Cent. Eur. J. Comp. Sci. • 2(4) • 2012 • 440-458
DOI: 10.2478/s13537-012-0029-1

VERSITA

Central European Journal of **Computer Science**

Application of Hurwitz-Radon matrices in curve interpolation and almost-smoothing

Research Article

Dariusz Jakóbczak*

Koszalin University of Technology,
Chair of Computer Science and Management,
Ul. Śniadeckich 2, 75-543 Koszalin, Poland

Received 11 April 2011; accepted 22 September 2012

Abstract: Dedicated methods for dealing with curve interpolation and curve smoothing have been developed. One such method, Hurwitz-Radon Matrices (MHR), can be used in reconstruction and interpolation of curves in the plane. The method is based on a family of Hurwitz-Radon (HR) matrices. The matrices are skew-symmetric and possess columns composed of orthogonal vectors. The of Hurwitz-Radon Operator (OHR), built from these matrices, is described. It is shown how to create the orthogonal and discrete OHR and how to use it in a process of curve interpolation and modelling. The method needs suitable choice of nodes, i.e. points of the curve to be reconstructed: nodes should be settled at each local extremum and nodes should be monotonic in one of coordinates (for example equidistance). Application of MHR gives a very good interpolation accuracy in the process of modeling and reconstruction of the curve. Created from the family of *N-1* HR matrices and completed with the identity matrix, the system of matrices is orthogonal only for vector spaces of dimensions N = 2, 4 or 8. Orthogonality of columns and rows is very important and significant for stability and high precision of calculations. The MHR method models the curve point by point without using any formula or function. Main features of the MHR method are: the accuracy of curve reconstruction depends on the number of nodes and method of choosing nodes, interpolation of L points of the curve has a computational cost of rank $O(L)$, and the smoothing of the curve depends on the number of OHR operators used to build the average matrix operator. The problem of curve length estimation is also considered. Algorithms and numerical results are presented.

Keywords: curve interpolation • curve smoothing • length estimation • points reconstruction • curve modelling • Hurwitz-Radon matrices • MHR method

© *Versita Sp. z o.o.*

1. Introduction

The following question is important in mathematics and computer science: is it possible to find a method of curve interpolation in the plane without building the interpolation polynomials or other functions? This paper aims at giving the positive answer to this question [18]. **This paper defines the problem statement as: given at least five key points of the curve (interpolation nodes) how this curve can be interpolated avoiding the limitations of classical interpolation**

* *E-mail: Dariusz.Jakobczak@ie.tu.koszalin.pl*

methods? Current methods of curve interpolation [39] are based on classical polynomial interpolation: Newton, Lagrange or Hermite polynomials and spline curves which are piecewise polynomials [5, 7]. Classical methods cannot be used to interpolate functions are not cotinuously differentiable [1] such as the absolute value function $f(x) = |x|$ at $x = 0$. If point (0,0) is one of the interpolation nodes, then precise polynomial interpolation of the absolute value function is very hard [11]. Also, when the graph of the interpolated function differs considerably from the shape of the interpolation polynomial, for example $f(x)=1/x$, interpolation is very difficult because of existing local extrema of the polynomial [33]. We cannot forget about the Runge's phenomenon: when interpolation nodes are equidistance then high-order polynomials oscillate toward at the ends of the interval, for example close to -1 and 1 with functions $f(x) = 1/(1 + 25x^2)$ [30, 37, 38] or $f(x) = 1/(1 + 5x^2)$. Of course there exist several well established methods: spline functions [2, 12, 34], shape-preserving techniques [6, 28], subdivision algorithms [3, 8, 31, 40], Bezier curves, B-splines, NURBS [27, 32] and others [25] to overcome the above difficulties of polynomial interpolation, but matrix interpolation MHR (based on simple matrix calculations with low computational costs) seems to be quite novel in the area of interpolation.
This paper deals with the problem of interpolation without computing the polynomials or any fixed functions. Values of nodes are used for building the orthogonal matrix operators and a linear (convex) combination of Hurwitz-Radon Operators (OHR) leads to the calculation of curve points. The main idea of the MHR method is that the curve is interpolated point by point by computing the unknown coordinates of the points. The only significant factors in the MHR method are: choosing the interpolation nodes, fixing the dimension of the HR matrix (N = 2, 4 or 8), and setting the number of OHR operators in the process of building the average OHR operator. Other characteristic features of the function or curve, such as the shape or similarity to polynomials, derivative or Runge's phenomenon, are not important in the process of MHR interpolation. The curve or function in the MHR method is parameterized for value $\alpha \in [0,1]$ in the range of two or more successive interpolation nodes [28]. Estimation of the curve length with high precision is possible increasing the number of curve points used.
In this paper, computational algorithms are considered as well as the computational time. Complexity of calculations for one unknown point in Lagrange or Newton interpolation based on n nodes is connected with the computational cost of $O(n^2)$. Complexity of calculations for L unknown points in MHR interpolation based on n nodes is connected with the computational cost of rank $O(L)$. This is a very important feature of the MHR method.
The paper is organized as follows: Section 2 presents properties of Hurwitz-Radon matrices and the method of curve interpolation based on the average OHR operator. Section 3 deals with the numerical complexity of MHR calculations. Section 4 pays attention to fixing the interpolation nodes. Section 5 presents the way of curve smoothing in the MHR method. Section 6 deals with the problem of curve length estimation and describes numerical results using the MHR method. Section 7 presents conclusions and future work.

2. The method of Hurwitz-Radon Matrices

This section deals with the Hurwitz-Radon matrices, the OHR operators and the novel method of curve interpolation called MHR.
Adolf Hurwitz (1859-1919) and Johann Radon (1887-1956) published papers [13, 14, 29] about specific class of matrices in 1923.

Definition 2.1.

Matrices A_i, $i = 1, 2 \ldots m$ satisfying

$$A_j A_k + A_k A_j = 0, \quad Aj^2 = -I \quad \text{for} \quad j \neq k; \quad j,\ k\ = 1, 2 \ldots m \tag{1}$$

are called *a family of Hurwitz-Radon matrices*.

A family of HR matrices (1) has important features: HR matrices are skew-symmetric ($A_i^T = -A_i$) and invertable: $A_i^{-1} = -A_i$. For dimensions N = 2, 4 or 8 the family of Hurwitz-Radon matrices consists of N-1 matrices.
For $N = 2$ we have one matrix

$$A_1 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

For $N = 4$ there are three matrices with integer entries:

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, A_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}.$$

For $N = 8$ we have seven matrices with elements 0, ±1 [9].

2.1. The Hurwitz-Radon Operator (OHR)

Let us define the matrix $W(\underline{w})$ (also named W) and $W(w_0, \underline{w})$ for $N = 2$, 4 or 8 and $\underline{w} = (w_1, \ldots, w_{N-1}) \in \boldsymbol{R}^{N-1}$:

$$W = W(\underline{w}) = \sum_{i=1}^{N-1} w_i \cdot W_i, \qquad \sum_{i=1}^{N-1} w_i^2 = 1, \qquad W(w_0, \underline{w}) = W + w_0 I_N, \qquad w_0 \neq 0, \tag{2}$$

where $W_1, \ldots, W_{N-1}$ belong to the HR family of matrices (1) with elements 0, ±1 and dimension N, $w_0 \neq 0, w_1, \ldots, w_{N-1}$ are real numbers, and I_N represents the identity matrix. The (OHR) operator M is built using $W(w_0, \underline{w})$.
For $N = 2$ the matrices are (2):

$$W = \begin{bmatrix} 0 & w_1 \\ -w_1 & 0 \end{bmatrix}, \quad W(w_0, \underline{w}) = \begin{bmatrix} w_0 & w_1 \\ -w_1 & w_0 \end{bmatrix}.$$

For $N = 4$ the matrix W is a linear combination of three HR matrices and $W(w_0, \underline{w}) = W + w_0 I_4$:

$$W = \begin{bmatrix} 0 & w_1 & w_2 & w_3 \\ -w_1 & 0 & -w_3 & w_2 \\ -w_2 & w_3 & 0 & -w_1 \\ -w_3 & -w_2 & w_1 & 0 \end{bmatrix}, \quad W(w_0, \underline{w}) = \begin{bmatrix} w_0 & w_1 & w_2 & w_3 \\ -w_1 & w_0 & -w_3 & w_2 \\ -w_2 & w_3 & w_0 & -w_1 \\ -w_3 & -w_2 & w_1 & w_0 \end{bmatrix}.$$

For $N = 8$ the matrix W is a linear combination of seven HR matrices and $W(w_0, \underline{w}) = W + w_0 I_8$:

$$W = \begin{bmatrix} 0 & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \\ -w_1 & 0 & w_3 & -w_2 & w_5 & -w_4 & -w_7 & w_6 \\ -w_2 & -w_3 & 0 & w_1 & w_6 & w_7 & -w_4 & -w_5 \\ -w_3 & w_2 & -w_1 & 0 & w_7 & -w_6 & w_5 & -w_4 \\ -w_4 & -w_5 & -w_6 & -w_7 & 0 & w_1 & w_2 & w_3 \\ -w_5 & w_4 & -w_7 & w_6 & -w_1 & 0 & -w_3 & w_2 \\ -w_6 & w_7 & w_4 & -w_5 & -w_2 & w_3 & 0 & -w_1 \\ -w_7 & -w_6 & w_5 & w_4 & -w_3 & -w_2 & w_1 & 0 \end{bmatrix}, W(w_0, \underline{w}) = \begin{bmatrix} w_0 & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \\ -w_1 & w_0 & w_3 & -w_2 & w_5 & -w_4 & -w_7 & w_6 \\ -w_2 & -w_3 & w_0 & w_1 & w_6 & w_7 & -w_4 & -w_5 \\ -w_3 & w_2 & -w_1 & w_0 & w_7 & -w_6 & w_5 & -w_4 \\ -w_4 & -w_5 & -w_6 & -w_7 & w_0 & w_1 & w_2 & w_3 \\ -w_5 & w_4 & -w_7 & w_6 & -w_1 & w_0 & -w_3 & w_2 \\ -w_6 & w_7 & w_4 & -w_5 & -w_2 & w_3 & w_0 & -w_1 \\ -w_7 & -w_6 & w_5 & w_4 & -w_3 & -w_2 & w_1 & w_0 \end{bmatrix}.$$

Let $\mathbf{x} = (x_1, x_2 \ldots, x_N) \in \boldsymbol{R}^N$, $\mathbf{x} \neq \mathbf{0}$, $\mathbf{y} = (y_1, y_2 \ldots, y_N) \in \boldsymbol{R}^N$, $\mathbf{y} \neq \mathbf{0}$ for $N = 2$, 4 or 8. Assume a new vector of first coordinates $\mathbf{v} = (v_1, v_2 \ldots, v_N) \in \boldsymbol{R}^N$ satisfying $\mathbf{v} = \mathbf{x} - w_0 \cdot \mathbf{y}$ for $w_0 \neq 0$ and $\mathbf{y} = W \cdot \mathbf{v}$ [4, 36]. The following equations (3):

$$\begin{aligned} \mathbf{x} &= \mathbf{v} + w_0 \cdot \mathbf{y}, \\ \mathbf{x} &= -W \cdot \mathbf{y} + w_0 \cdot \mathbf{y}, \\ \mathbf{x} &= (-W + w_0 I_N)\mathbf{y} \end{aligned} \tag{3}$$

determine the system of linear equations with solutions $w_0, w_1, \ldots, w_{N-1}$ for a given vector of first coordinates $\mathbf{x}$ and a vector of second coordinates $\mathbf{y}$. We can write the solution for (3):

1) for $N = 2$:

$$\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \frac{1}{y_1^2 + y_2^2} \begin{bmatrix} y_1 & y_2 \\ -y_2 & y_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

2) for $N = 4$:

$$\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} = \frac{1}{\sum_{i=1}^{4} y_i^2} \begin{bmatrix} y_1 & y_2 & y_3 & y_4 \\ -y_2 & y_1 & y_4 & -y_3 \\ -y_3 & -y_4 & y_1 & y_2 \\ -y_4 & y_3 & -y_2 & y_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix},$$

3) for $N = 8$:

$$\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{bmatrix} = \frac{1}{\sum_{i=1}^{8} y_i^2} \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ -y_2 & y_1 & -y_4 & y_3 & -y_6 & y_5 & y_8 & -y_7 \\ -y_3 & y_4 & y_1 & -y_2 & -y_7 & -y_8 & y_5 & y_6 \\ -y_4 & -y_3 & y_2 & y_1 & -y_8 & y_7 & -y_6 & y_5 \\ -y_5 & y_6 & y_7 & y_8 & y_1 & -y_2 & -y_3 & -y_4 \\ -y_6 & -y_5 & y_8 & -y_7 & y_2 & y_1 & y_4 & -y_3 \\ -y_7 & -y_8 & -y_5 & y_6 & y_3 & -y_4 & y_1 & y_2 \\ -y_8 & y_7 & -y_6 & -y_5 & y_4 & y_3 & -y_2 & y_1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}.$$

According to (3):

$$\mathbf{y} = (\text{-}W + w_0 \cdot I_N)^{-1} \cdot \mathbf{x}.$$

The Operator of Hurwitz-Radon M satisfies

$$\mathbf{y} \ = M \cdot \mathbf{x} \quad \text{for} \quad M = (-W + w_0 \cdot I_N)^{-1}$$

which means that

$$M = \frac{1}{\sum_{i=0}^{N-1} w_i^2} (W + w_0 I), M^{-1} = w_0 I_N - W. \tag{4}$$

After some reductions in (4) we can construct OHR as follows (5)-(11). Assume there is given a finite set of points of the curve, called further nodes $(x_i, y_i) \in \boldsymbol{R}^2$ such that:

1. nodes (key points) are settled at each local extremum (maximum or minimum of the coordinate) and at least one point between two successive local extrema;
2. each node (x_i, y_i) is monotonic in coordinates x_i or y_i;
3. there are at least five nodes for the curve.

Assume that the nodes belong to a curve in the plane. How can the whole curve be reconstructed using this discrete set of nodes? Proposed method [15, 16] is based on local, orthogonal matrix operators. Values of nodes' coordinates (x_i, y_i) are connected with HR matrices [31] built in N dimensional vector space. It is important that the HR matrices are skew-symmetric and only for dimensions $N = 2$, 4 or 8 columns and rows the HR matrices are orthogonal [26].
If one curve is described by a set of nodes $\{(x_i, y_i), i = 1, 2, \ldots, n\}$ monotonic in coordinates x_i, then HR matrices combined with identity matrix are used to build an orthogonal and discrete Hurwitz-Radon Operator (OHR). For nodes (x_1, y_1) and (x_2, y_2) OHR of dimension $N = 2$ is constructed:

$$M = \frac{1}{x_1^2 + x_2^2}\begin{bmatrix} x_1 & x_2 \\ -x_2 & x_1 \end{bmatrix}\begin{bmatrix} y_1 & -y_2 \\ y_2 & y_1 \end{bmatrix} = \frac{1}{x_1^2 + x_2^2}\begin{bmatrix} x_1y_1 + x_2y_2 & x_2y_1 - x_1y_2 \\ x_1y_2 - x_2y_1 & x_1y_1 + x_2y_2 \end{bmatrix}. \tag{5}$$

For nodes (x_1,y_1), (x_2,y_2), (x_3,y_3), (x_4,y_4) monotonic in x_i OHR of dimension $N = 4$ is constructed:

$$M = \frac{1}{x_1^2 + x_2^2 + x_3^2 + x_4^2}\begin{bmatrix} u_0 & u_1 & u_2 & u_3 \\ -u_1 & u_0 & -u_3 & u_2 \\ -u_2 & u_3 & u_0 & -u_1 \\ -u_3 & -u_2 & u_1 & u_0 \end{bmatrix}, \tag{6}$$

where

$$u_0 = x_1y_1 + x_2y_2 + x_3y_3 + x_4y_4, \qquad u_1 = -x_1y_2 + x_2y_1 + x_3y_4 - x_4y_3,$$
$$u_2 = -x_1y_3 - x_2y_4 + x_3y_1 + x_4y_2, \qquad u_3 = -x_1y_4 + x_2y_3 - x_3y_2 + x_4y_1.$$

For nodes (x_1,y_1), (x_2,y_2), ..., (x_8,y_8) monotonic in x_i OHR of dimension $N = 8$ is equal to

$$M = \frac{1}{\sum_{i=1}^{8} x_i^2}\begin{bmatrix} u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \\ -u_1 & u_0 & u_3 & -u_2 & u_5 & -u_4 & -u_7 & u_6 \\ -u_2 & -u_3 & u_0 & u_1 & u_6 & u_7 & -u_4 & -u_5 \\ -u_3 & u_2 & -u_1 & u_0 & u_7 & -u_6 & u_5 & -u_4 \\ -u_4 & -u_5 & -u_6 & -u_7 & u_0 & u_1 & u_2 & u_3 \\ -u_5 & u_4 & -u_7 & u_6 & -u_1 & u_0 & -u_3 & u_2 \\ -u_6 & u_7 & u_4 & -u_5 & -u_2 & u_3 & u_0 & -u_1 \\ -u_7 & -u_6 & u_5 & u_4 & -u_3 & -u_2 & u_1 & u_0 \end{bmatrix} \tag{7}$$

where $\underline{u} = (u_0, u_1,..., u_7)^T$:

$$\underline{u} = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ -y_2 & y_1 & -y_4 & y_3 & -y_6 & y_5 & y_8 & -y_7 \\ -y_3 & y_4 & y_1 & -y_2 & -y_7 & -y_8 & y_5 & y_6 \\ -y_4 & -y_3 & y_2 & y_1 & -y_8 & y_7 & -y_6 & y_5 \\ -y_5 & y_6 & y_7 & y_8 & y_1 & -y_2 & -y_3 & -y_4 \\ -y_6 & -y_5 & y_8 & -y_7 & y_2 & y_1 & y_4 & -y_3 \\ -y_7 & -y_8 & -y_5 & y_6 & y_3 & -y_4 & y_1 & y_2 \\ -y_8 & y_7 & -y_6 & -y_5 & y_4 & y_3 & -y_2 & y_1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}. \tag{8}$$

We can see that the components of vector $\underline{u} = (u_0, u_1,..., u_7)^T$, appearing in matrix M (7) are defined by (8) in the similar way to (6) but in terms of the coordinates of the above 8 nodes. Note that OHR operators (5)-(7) satisfy the condition of interpolation:

$$M \cdot \mathbf{x} = \mathbf{y}$$

for $\mathbf{x} = (x_1,x_2...,x_N) \in \boldsymbol{R}^N$, $\mathbf{x} \neq \mathbf{0}$, $\mathbf{y} = (y_1,y_2...,y_N) \in \boldsymbol{R}^N$, $N = 2$, 4 or 8.
If one curve is described by a set of nodes $\{(x_i,y_i),\ i = 1,2,...,n\}$ monotonic in coordinates y_i, all calculations (5)-(8) and others are completely symmetric exchanging x_i with y_i. For nodes (x_1,y_1), (x_2,y_2) reverse OHR of dimension $N = 2$ is constructed:

$$M^{-1} = \frac{1}{y_1^2 + y_2^2} \begin{bmatrix} x_1 & -x_2 \\ x_2 & x_1 \end{bmatrix} \begin{bmatrix} y_1 & y_2 \\ -y_2 & y_1 \end{bmatrix} = \frac{1}{y_1^2 + y_2^2} \begin{bmatrix} x_1 y_1 + x_2 y_2 & -x_2 y_1 + x_1 y_2 \\ -x_1 y_2 + x_2 y_1 & x_1 y_1 + x_2 y_2 \end{bmatrix}. \tag{9}$$

For nodes (x_1,y_1), (x_2,y_2), (x_3,y_3), (x_4,y_4) monotonic in y_i the reverse OHR of dimension $N = 4$ is constructed for (6):

$$M^{-1} = \frac{1}{y_1^2 + y_2^2 + y_3^2 + y_4^2} \begin{bmatrix} u_0 & -u_1 & -u_2 & -u_3 \\ u_1 & u_0 & u_3 & -u_2 \\ u_2 & -u_3 & u_0 & u_1 \\ u_3 & u_2 & -u_1 & u_0 \end{bmatrix}. \tag{10}$$

For nodes (x_1,y_1), (x_2,y_2), ..., (x_8,y_8) monotonic in y_i the reverse OHR of dimension $N = 8$ is equal with

$$M^{-1} = \frac{1}{\sum_{i=1}^{8} y_i^2} \begin{bmatrix} u_0 & -u_1 & -u_2 & -u_3 & -u_4 & -u_5 & -u_6 & -u_7 \\ u_1 & u_0 & -u_3 & u_2 & -u_5 & u_4 & u_7 & -u_6 \\ u_2 & u_3 & u_0 & -u_1 & -u_6 & -u_7 & u_4 & u_5 \\ u_3 & -u_2 & u_1 & u_0 & -u_7 & u_6 & -u_5 & u_4 \\ u_4 & u_5 & u_6 & u_7 & u_0 & -u_1 & -u_2 & -u_3 \\ u_5 & -u_4 & u_7 & -u_6 & u_1 & u_0 & u_3 & -u_2 \\ u_6 & -u_7 & -u_4 & u_5 & u_2 & -u_3 & u_0 & u_1 \\ u_7 & u_6 & -u_5 & -u_4 & u_3 & u_2 & -u_1 & u_0 \end{bmatrix}, \tag{11}$$

where the components of the vector $\underline{u} = (u_0, u_1, \ldots, u_7)^T$ are defined in terms of (8).
Note that reverse OHR operators (9)-(11) satisfy the condition of interpolation

$$M^{-1} \cdot \mathbf{y} = \mathbf{x}$$

for $\mathbf{x} = (x_1,x_2\ldots,x_N) \in \boldsymbol{R}^N$, $\mathbf{y} = (y_1,y_2\ldots,y_N) \in \boldsymbol{R}^N$, $\mathbf{y} \neq \mathbf{0}$, $N = 2$, 4 or 8.

2.2. The method of Hurwitz-Radon Matrices

How can we compute coordinates of points settled between interpolation nodes? On a segment of a line every number "c" situated between "a" and "b" is described by a linear (convex) combination $c = \alpha \cdot a + (1-\alpha) \cdot b$ (Fig. 1)

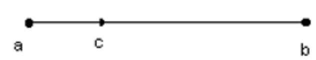


Figure 1. Point "*c*" between "*a*" and "*b*".

for

$$\alpha = \frac{b-c}{b-a} \in [0; 1]. \tag{12}$$

When the nodes are monotonic in coordinates x_i, average OHR operator M_2 of dimension $N = 2$, 4 or 8 is constructed as follows:

$$M_2 = \alpha \cdot M_0 + (1-\alpha) \cdot M_1 \tag{13}$$

with the operator M_0 built (5)-(7) by "odd" nodes (x_1=a,y_1), (x_3,y_3), ..., (x_{2N-1},y_{2N-1}) and M_1 built (5)-(7) by "even" nodes (x_2=b,y_2), (x_4,y_4), ..., (x_{2N},y_{2N}). Having the operator M_2 for coordinates $x_i < x_{i+1}$ it is possible to reconstruct the second coordinates of points (x,y) in terms of the vector C defined with

$$c_i = \alpha \cdot x_{2i-1} + (1-\alpha) \cdot x_{2i} \quad \text{for} \quad i = 1, 2, \ldots, N \tag{14}$$

as $C = [c_1, c_2, \ldots, c_N]^T$. The required formula is

$$Y(C) = M_2 \cdot C \tag{15}$$

in which components of vector $Y(C)$ give the second coordinate of the points (x,y) corresponding to the first coordinate, given in terms of components of vector C.
On the other hand, for coordinates $y_i < y_{i+1}$ it is possible to reconstruct the first coordinates of points (x,y):

$$c_i = \alpha \cdot y_{2i-1} + (1-\alpha) \cdot y_{2i} \quad \text{for} \quad i = 1, 2, \ldots, N \tag{16}$$

$$C = [c_1, c_2, \ldots, c_N]^T,$$

$$M_2^{-1} = \alpha \cdot M_0^{-1} + (1-\alpha) \cdot M_1^{-1}, \tag{17}$$

$$X(C) = M_2^{-1} \cdot C. \tag{18}$$

After computing (15) or (18) for any $\alpha \in [0;1]$, we have a half of reconstructed points ($j = 1$ in Algorithm 1). Now it is necessary to find second half of unknown coordinates ($j = 2$ in Algorithm 1) for

$$c_i = \alpha \cdot x_{2i} + (1-\alpha) \cdot x_{2i+1}, \quad i = 1, 2, \ldots, N \tag{19}$$

or

$$c_i = \alpha \cdot y_{2i} + (1-\alpha) \cdot y_{2i+1}, \quad i = 1, 2, \ldots, N \tag{20}$$

depending on whether x_i or y_i is monotonic. There is no need to build the OHR for nodes (x_2,y_2), (x_4,y_4), ..., (x_{2N},y_{2N}), because we found M_1 or M_1^{-1}. This operator will play the role of M_0 or M_0^{-1} in (13) or (17). New M_1 or M_1^{-1} must be computed for nodes (x_3,y_3),..., (x_{2N-1},y_{2N-1}), (x_{2N+1},y_{2N+1}).
As we see the minimum number of interpolation nodes $n = 2N+1 = 5$, 9 or 17 using OHR operators of dimension $N = 2$, 4 or 8 respectively. If there are more nodes than $2N+1$, the same calculations (12)-(20) have to be done for next range(s) or last range of $2N+1$ nodes. For example, if $n = 9$ then we can use OHR operators of dimension $N = 4$ or OHR operators of dimension $N = 2$ for two subsets of nodes: {(x_1,y_1), ...,(x_5,y_5)} and {(x_5,y_5), ...,(x_9,y_9)}.
We summarize this section in the following algorithm of points reconstruction for $2N+1 = 5$, 9 or 17 successive nodes.

Algorithm 1

Let $j = 1$.

Input: Set of interpolation nodes (x_i,y_i), $i = 1,2,...,n$; $n = 5$, 9 or 17 such as:

a) nodes are settled at each local extremum and at least one point between two successive local extrema

b) nodes (x_i,y_i) are monotonic in coordinates x_i or y_i.

Step 1. Determine the dimension N of OHR operators: $N = 2$ if $n = 5$, $N = 4$ if $n = 9$, $N = 8$ if $n = 17$.

Step 2. If nodes are monotonic in x_i then build M_0 for nodes $(x_1=a,y_1)$, (x_3,y_3), ..., (x_{2N-1},y_{2N-1}) and M_1 for nodes $(x_2=b,y_2)$, (x_4,y_4), ..., (x_{2N},y_{2N}) from (5)-(7). If nodes are monotonic in y_i then build M_0^{-1} for nodes $(x_1,y_1=a)$, (x_3,y_3), ..., (x_{2N-1},y_{2N-1}) and M_1^{-1} for nodes $(x_2,y_2=b)$, (x_4,y_4), ..., (x_{2N},y_{2N}) from (9)-(11).

Step 3. Determine the number of points to be reconstructed $K_j > 0$ between two successive nodes, let $k = 1$.

Step 4. Compute $\alpha \in [0;1]$ from (12) for $c_1 = c = \alpha\cdot a + (1-\alpha)\cdot b$.

Step 5. Build M_2 from (13) or M_2^{-1} from (14).

Step 6. Compute vector $C = [c_1, c_2,...,c_N]^T$ from (15) or (17).

Step 7. Compute unknown coordinates $Y(C)$ from (16) or $X(C)$ from (18).

Step 8. If $k < K_j$, set $k = k+1$ and go to Step 4. Otherwise if $j = 1$, set $M_0 = M_1$, $a = x_2$, $b = x_3$ (if nodes monotonic in x_i) or $M_0^{-1} = M_1^{-1}$, $a = y_2$, $b = y_3$ (if nodes monotonic in y_i), build new M_1 or M_1^{-1} for nodes (x_3,y_3), (x_5,y_5), ..., (x_{2N+1},y_{2N+1}), let $j = 2$ and go to Step 3. Otherwise, stop.

Remark 1. The MHR method defined by Algorithm 1 can not be used to interpolate an arbitrary number of points (Algorithm 1 is valid only in ranges of 5, 9 or 17 successive nodes), but the MHR method as defined by Algorithm 2 (Section (5) can be used to interpolate an arbitrary odd number of points in case of proportional distribution of nodes at each half, quarter or one eigth of nodes (for example nodes equidistance in one coordinate).

The number of reconstructed points in Algorithm 1 is $K = N(K_1+K_2)$. If there are more nodes than $2N+1 = 5$, 9 or 17, Algorithm 1 has to be done for next range(s) or last range of $2N+1$ nodes. Reconstruction of curve points using Algorithm 1 is called the method of Hurwitz-Radon Matrices (MHR). The next section deals with complexity of MHR calculations.

3. Complexity of MHR calculations

In this section we consider the number of multiplications and divisions for the MHR method during reconstruction of $K = L - n$ points having n interpolation nodes of the curve consists of L points.

First we present a formula for computing one unknown coordinate of a single point. Assume there are given four nodes (x_1,y_1), (x_2,y_2), (x_3,y_3) and (x_4,y_4) monotonic in x_i. OHR operators of dimension $N = 2$ are built (5) as follows:

$$M_0 = \frac{1}{x_1^2+x_3^2}\begin{bmatrix} x_1y_1+x_3y_3 & x_3y_1-x_1y_3 \\ x_1y_3-x_3y_1 & x_1y_1+x_3y_3 \end{bmatrix}, \quad M_1 = \frac{1}{x_2^2+x_4^2}\begin{bmatrix} x_2y_2+x_4y_4 & x_4y_2-x_2y_4 \\ x_2y_4-x_4y_2 & x_2y_2+x_4y_4 \end{bmatrix}.$$

Let first coordinate c_1 of reconstructed point is situated between x_1 and x_2:

$$c_1 = \alpha\cdot x_1 + \beta\cdot x_2 \quad \text{for} \quad 0 \le \beta = 1-\alpha \le 1. \tag{21}$$

Compute second coordinate of reconstructed point $y(c_1)$ for $Y(C) = [y(c_1), y(c_2)]^T$ from (16):

$$\begin{bmatrix} y(c_1) \\ y(c_2) \end{bmatrix} = (\alpha\cdot M_0 + \beta\cdot M_1)\cdot\begin{bmatrix} \alpha\cdot x_1+\beta\cdot x_2 \\ \alpha\cdot x_3+\beta\cdot x_4 \end{bmatrix}. \tag{22}$$

After calculation (22):

$$y(c_1) = \alpha^2\cdot y_1 + \beta^2\cdot y_2 + \frac{\alpha\cdot\beta}{x_1^2+x_3^2}(x_1x_2y_1 + x_2x_3y_3 + x_3x_4y_1 - x_1x_4y_3) + \frac{\alpha\cdot\beta}{x_2^2+x_4^2}(x_1x_2y_2 + x_1x_4y_4 + x_3x_4y_2 - x_2x_3y_4). \tag{23}$$

So each point of the curve $P = (c_1, y(c_1))$ settled between nodes (x_1,y_1) and (x_2,y_2) is parameterized by $P(\alpha)$ of rank α^2 for (21), (23) and $\alpha \in [0;1]$. Similar calculations could be done for nodes (x_1,y_1), (x_2,y_2), (x_3,y_3) and (x_4,y_4) monotonic in y_i to compute $x(c_1)$:

$$M_0^{-1} = \frac{1}{y_1^2 + y_3^2}\begin{bmatrix} x_1y_1 + x_3y_3 & -x_3y_1 + x_1y_3 \\ -x_1y_3 + x_3y_1 & x_1y_1 + x_3y_3 \end{bmatrix},$$
$$M_1^{-1} = \frac{1}{y_2^2 + y_4^2}\begin{bmatrix} x_2y_2 + x_4y_4 & -x_4y_2 + x_2y_4 \\ -x_2y_4 + x_4y_2 & x_2y_2 + x_4y_4 \end{bmatrix},$$
$$c_1 = \alpha \cdot y_1 + \beta \cdot y_2 \quad for \quad 0 \le \beta = 1 - \alpha \le 1,$$
$$\begin{bmatrix} x(c_1) \\ x(c_2) \end{bmatrix} = (\alpha \cdot M_0^{-1} + \beta \cdot M_1^{-1}) \cdot \begin{bmatrix} \alpha \cdot y_1 + \beta \cdot y_2 \\ \alpha \cdot y_3 + \beta \cdot y_4 \end{bmatrix},$$
$$x(c_1) = \alpha^2 \cdot x_1 + \beta^2 \cdot x_2 + \frac{\alpha \cdot \beta}{y_1^2 + y_3^2} r_1 + \frac{\alpha \cdot \beta}{y_2^2 + y_4^2} r_2 \tag{24}$$

for $r_1 = const.$, $r_2 = const.$ depending on nodes' coordinates: see (23). If nodes are monotonic in y_i, there is parameterization of curve points P settled between nodes (x_1,y_1) and (x_2,y_2): $P(\alpha) = (x(c_1), c_1)$ of rank α^2 for $\alpha \in [0;1]$.
The goal of this paper is not a reconstruction of single point, like for example (21) and (23), but interpolation of curve consists of L points. If we have n interpolation nodes, then there is $K = L - n$ points to find using Algorithm 1 and MHR method. Now we consider the complexity of MHR calculations.

Lemma 3.1.
Let n = 5, 9 or 17 is the number of interpolation nodes, let MHR method (Algorithm 1) is done for reconstruction of the curve consists of L points. Then MHR method is connected with the computational cost of rank $O(L)$.

Proof. Using Algorithm 1 we have to reconstruct $K = L - n$ points of unknown curve. Counting the number of multiplications and divisions D in Algorithm 1 here are the results:

1. $D = 4L+7$ for $n = 5$ and $L = 2i + 5$;
2. $D = 6L+21$ for $n = 9$ and $L = 4i + 9$;
3. $D = 10L+73$ for $n = 17$ and $L = 8i + 17$; i = 2,3,4...

□

The lowest computational cost appears in MHR method with five nodes and OHR operators of dimension $N = 2$. Therefore whole set of n nodes can be divided into subsets of five nodes. Then whole curve is to be reconstructed by Algorithm 1 with all subsets of five nodes: $\{(x_1,y_1), ..., (x_5,y_5)\}$, $\{(x_5,y_5), ..., (x_9,y_9)\}$,$\{(x_9,y_9), ..., (x_{13},y_{13})\}$....If the last node (x_n,y_n) is indexed $n \neq 4i+1$ then we have to use last five nodes $\{(x_{n-4},y_{n-4}), ..., (x_n,y_n)\}$ in Algorithm 1.
Advantage of using OHR operators of dimension $N = 2$ is described in Section 5, but first Section 4 deals with fixing the nodes, for example case of equidistance nodes and calculations (parameterization, interpolation error) connected with equidistance nodes.

4. Fixing the interpolation nodes

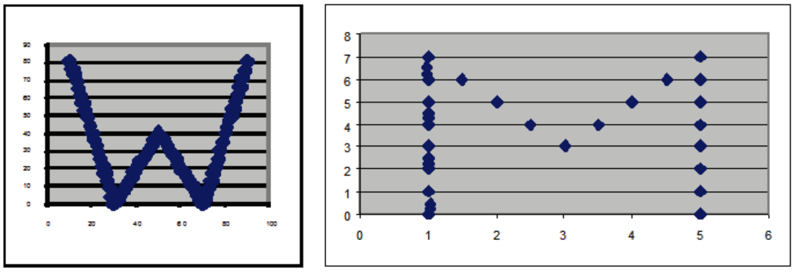
If nodes (x_i,y_i) are equidistance in coordinate x_i or y_i, then parameterization of unknown coordinate (23)-(24) is simpler. Let four successive nodes (x_1,y_1), (x_2,y_2), (x_3,y_3) and (x_4,y_4) are equidistance in coordinate x_i and $a = x_1$, $h/2 = x_{i+1} - x_i = const.$ Calculations (22)-(23) are done for c_1 (21):

$$y(c_1) = \alpha y_1 + \beta y_2 + \alpha\beta s \tag{25}$$

and

$$s = h\left(\frac{2ay_1 + hy_1 + hy_3}{4a^2 + 4ah + 2h^2} - \frac{2ay_2 + 2hy_2 + hy_4}{4a^2 + 8ah + 5h^2}\right). \tag{26}$$

As we can see in (25)-(26), MHR interpolation is not a linear interpolation. So the scheme does not reproduce exactly constant and linear functions, but graphical reconstructions of constant or linear functions are optimistic:



(a) (b)

Figure 2. Linear functions (a) and constant curve $x = 1$ (b) reconstructed by MHR method.

Examples of calculations in case of constant curve $x = 1$ (Fig. 2b) and average reverse OHR operator A_2 of dimension $N = 4$:

a) $y^T = \begin{bmatrix} 0.5 & 2.5 & 4.5 & 6.5 \end{bmatrix}$,

$$A_2 = \begin{bmatrix} 0.202381 & 0 & 0.119048 & 0.059524 \\ 0 & 0.202381 & 0.059524 & 0.119048 \\ -0.119048 & 0.059524 & 0.202381 & 0 \\ -0.059524 & -0.119048 & 0 & 0.202381 \end{bmatrix}, \quad A_2 \cdot y = \begin{bmatrix} 1.02381 \\ 1.011903 \\ 1.000001 \\ 0.988096 \end{bmatrix};$$

b) $y^T = \begin{bmatrix} 0.25 & 2.25 & 4.25 & 6.25 \end{bmatrix}$,

$$A_2 = \begin{bmatrix} 0.208334 & 0 & 0.130952 & 0.065477 \\ 0 & 0.208334 & -0.065477 & 0.130952 \\ -0.130952 & 0.065477 & 0.208334 & 0 \\ -0.065477 & -0.130952 & 0 & 0.208334 \end{bmatrix}, \quad A_2 \cdot y = \begin{bmatrix} 1.017859 \\ 1.008927 \\ 1.000001 \\ 0.991073 \end{bmatrix};$$

In case of constant function we can always use simple linear interpolation. It is possible to estimate the interpolation error of MHR method (Algorithm 1) for the class of linear function f:

$$|f(c_1) - y(c_1)| = |\alpha y_1 + \beta y_2 - y(c_1)| = \alpha\beta\,|s|\,. \tag{27}$$

Notice that estimation (27) has the biggest value $0.25|s|$ for $\beta = \alpha = 0.5$, when c_1 is situated in the middle between x_1 and x_2. In the case of constant function y coefficient s (26) is calculated:

$$s = \frac{h^2 y(x_1 h + 2x_3^2)}{(x_1^2 + x_3^2)(h^2 + 4x_3^2)}.$$

So $s \to 0$ when $h \to 0$: for "very small" h MHR method is similar to linear interpolation.

Having four successive nodes (x_1,y_1), (x_2,y_2), (x_3,y_3) and (x_4,y_4) equidistance in coordinate x_i ($a = x_1$, $h/2 = x_{i+1} - x_i = const.$) we can compute polynomial $W_3(x) = m_3x^3 + m_2x^2 + m_1x + t$ for these nodes and estimate the interpolation error of MHR method (Algorithm 1) for the class of order three polynomials. After solving the system of equations:

$$
\begin{aligned}
y_1 &= m_3 a^3 + m_2 a^2 + m_1 a + t,\\
y_2 &= m_3\left(a+\frac{h}{2}\right)^3 + m_2\left(a+\frac{h}{2}\right)^2 + m_1\left(a+\frac{h}{2}\right) + t,\\
y_3 &= m_3(a+h)^3 + m_2(a+h)^2 + m_1(a+h) + t,\\
y_4 &= m_3\left(a+\frac{3h}{2}\right)^3 + m_2\left(a+\frac{3h}{2}\right)^2 + m_1\left(a+\frac{3h}{2}\right) + t,
\end{aligned}
$$

it is possible to compute m_3, m_2, m_1, t and the estimation for point c_1 (21):

$$
|W_3(c_1) - y(c_1)| = \beta\left|\frac{1}{2}h(y_2-y_1)\frac{12\alpha\beta a + 12a\beta^2 + 2h\beta^2 - \alpha\beta a h^2 - 12a - 2h}{6ha + 12a^2 + h^2} - \alpha\cdot s\right|. \tag{28}
$$

Notice that estimations (27)-(28) are equal with zero for $\alpha = 0$ or $\beta = 0$.
For eight successive nodes (x_i,y_i), $i = 1,2,...,8$ equidistance in coordinate x_i, $a = x_1$, $h/2 = x_{i+1} - x_i = const.$, using OHR of dimension $N = 4$ in (16) here is a formula of second coordinate reconstruction with first coordinate c_1 (21):

$$
\begin{aligned}
y(c_1) &= \alpha y_1 + \beta y_2 + \frac{\beta h^2}{4a^2+16ah+21h^2}[(\alpha y_3 + \beta y_4) + 2(\alpha y_7 + \beta y_8)]\\
&+ \frac{\beta h}{(4a^2+16ah+21h^2)(2a^2+6ah+7h^2)}[2(\alpha y_1+\beta y_2)(2a^3+10a^2h+19ah^2+14h^3)\\
&+ \alpha h(a y_1 + h y_7)(2a+7h) + 3.5\alpha h^3(y_1+y_3)\\
&+ 2ah(\alpha h y_3 + \alpha h y_7 + 17 a y_2 - a y_4 - 2a y_8 - 3h y_4 - 6h y_8)\\
&+ -7h^3(21y_2 + y_4 + 2y_8) - 2ay_2(2a^2+27ah+19h^2)].
\end{aligned} \tag{29}
$$

So we have another parameterization (29) of the point $P(\alpha) = (c_1, y(c_1))$ for $N = 4$ and $\beta = 1 - \alpha$. Formula (29) doesn't include values y_5 and y_6: Algorithm 1 with nine successive nodes (x_i,y_i), $i = 1,2,...,9$ equidistance in coordinate x_i is free of using y_5 and y_6 for computing second coordinate of the point settled between first and second node.
Algorithm 1 deals with average OHR operators (13)-(14) built with two OHR. This situation leads to parameterization of reconstructed point $P(\alpha) = (c_1, y(c_1))$ or $P(\alpha) = (x(c_1), c_1)$ settled between two successive nodes, where $\alpha \in [0;1]$ is order two in (23)-(25) or (29). It means that MHR method via Algorithm 1 gives curve interpolation with "sharp corner" in each node. So the reconstructed curve is not smooth in all interpolation nodes. Section 5 deals with the problem of curve smoothing.

5. Curve almost-smoothing

Assume that there is odd number of interpolation nodes (x_1,y_1), (x_2,y_2),..., (x_{2k+1},y_{2k+1}) in MHR method ($k = 2,3,4...$, $k = const.$) and all coordinates x_i or all coordinates y_i are settled proportionally in the first and second half of the nodes, including the case of equidistance nodes. For example dealing with coordinate x_i we have the condition of proportion for first and second half nodes ($N = 2$):

$$
\forall i = 2, ..., k : \frac{x_{k+1} - x_i}{x_{k+1} - x_1} = \frac{x_{2k+1} - x_{k+i}}{x_{2k+1} - x_{k+1}} = p_{i-1}. \tag{30}
$$

Values $p_1 > \ldots > p_{k-1} \in (0;1)$ with $p_0 = 1$ and $p_k = 0$ are crucial in the process of curve smoothing. In particular nodes equidistance in x_i satisfy (30).

Let M_i (i = 0,1,2,...,k) by an OHR operator of dimension $N = 2$ constructed (5) for nodes (x_{i+1},y_{i+1}) and (x_{k+i+1}, y_{k+i+1}). The average OHR operator M_{k+1} is built as follows:

$$M_{k+1} = \sum_{i=0}^{k} s_i \cdot M_i. \tag{31}$$

Average OHR operator M_2 in (13) is calculated using (31) for $k = 1$ and $p_1 = 0$.
Coefficients *si* are computed:

$$s_i = \frac{(\alpha - p_0)(\alpha - p_1)...(\alpha - p_{i-1})(\alpha - p_{i+1})...(\alpha - p_k)}{(p_i - p_0)(p_i - p_1)...(p_i - p_{i-1})(p_i - p_{i+1})...(p_i - p_k)}, \tag{32}$$

$$s_i = \frac{\prod_{j=0,j\neq i}^{k}(\alpha - p_j)}{\prod_{j=0,j\neq i}^{k}(p_i - p_j)}, \quad \sum_{i=0}^{k} s_i = 1$$

for any coordinate c_1 situated between x_1 and x_{k+1} (first half of nodes) as follows:

$$c_1 = \alpha \cdot x_1 + \beta \cdot x_{k+1} \quad \text{for} \quad 0 \leq \beta = 1 - \alpha \leq 1, \\ \alpha = \frac{x_{k+1} - c_1}{x_{k+1} - x_1} \in [0; 1]. \tag{33}$$

Notice that coefficients s_i (32) are similar to coefficients appearing in the Lagrange interpolation polynomial. Vector of second coordinates $Y(C) = [y(c_1), y(c_2)]^T$ is calculated:

$$Y(C) = M_{k+1} \cdot \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = M_{k+1} \cdot \left(\alpha \begin{bmatrix} x_1 \\ x_{k+1} \end{bmatrix} + (1 - \alpha) \begin{bmatrix} x_{k+1} \\ x_{2k+1} \end{bmatrix} \right), \tag{34}$$

where

$$c_2 = \alpha \cdot x_{k+1} + \beta \cdot x_{2k+1}.$$

We can say that (34) shows some kind of matrix version of Lagrange interpolation (looking at coefficients s_i) in case $N = 2$ separately for first half of nodes (computing coordinate y for c_1) and second half of nodes (value y for c_2).
Formula (34) gives a parameterization of any point $P(\alpha) = (c_1, y(c_1))$ or $(c_2, y(c_2))$ of order α^{k+1}. It means that reconstructed curve is smooth between nodes (x_1,y_1) and (x_{k+1},y_{k+1}), also between nodes (x_{k+1},y_{k+1}) and (x_{2k+1},y_{2k+1}). Whole interpolated curve is not smooth at one point: a central node (x_{k+1}, y_{k+1}). That's why the author wrote "almost-smoothing" in the title. Suitable selection of nodes is able to minimize this fact, especially on the screen with discrete set of pixels.
So using the interpolation nodes (x_1,y_1), (x_2,y_2),..., (x_{2k+1},y_{2k+1}) satisfying (30) we can build OHR operators M_0,...,M_{k+1} of dimension $N = 2$ (lower computational costs than operators of dimension $N = 4$ or 8) and the reconstructed curve has a "sharp corner" only in the central node (x_{k+1},y_{k+1}). OHR operators of dimension $N = 4$ have to be used if nodes are not proportionally distributed at each half, but proportionally distributed at each quarter of nodes (minimum nine nodes). Then there are three "sharp corners". OHR operators of dimension $N = 8$ have to be used if nodes are proportionally distributed at each one eigth of nodes (minimum 17 nodes). Then there are seven "sharp corners".

Here are the examples of average operators (31) for five, seven and nine nodes equidistance in coordinate x_i:

1) five nodes (x_1,y_1), (x_2,y_2),..., (x_5,y_5), $k = 2$, $p_2 = 0$, $p_1 = 1/2$, $p_0 = 1$:

$$M_0 = \frac{1}{x_1^2 + x_3^2} \begin{bmatrix} x_1 y_1 + x_3 y_3 & x_3 y_1 - x_1 y_3 \\ x_1 y_3 - x_3 y_1 & x_1 y_1 + x_3 y_3 \end{bmatrix},$$

$$M_1 = \frac{1}{x_2^2 + x_4^2} \begin{bmatrix} x_2 y_2 + x_4 y_4 & x_4 y_2 - x_2 y_4 \\ x_2 y_4 - x_4 y_2 & x_2 y_2 + x_4 y_4 \end{bmatrix},$$

$$M_2 = \frac{1}{x_3^2 + x_5^2} \begin{bmatrix} x_3 y_3 + x_5 y_5 & x_5 y_3 - x_3 y_5 \\ x_3 y_5 - x_5 y_3 & x_3 y_3 + x_5 y_5 \end{bmatrix},$$

$$s_0 = \frac{(\alpha-0)(\alpha-0.5)}{(1-0)(1-0.5)}, \quad s_1 = \frac{(\alpha-0)(\alpha-1)}{(0.5-0)(0.5-1)}, \quad s_2 = \frac{(\alpha-1)(\alpha-0.5)}{(0-1)(0-0.5)},$$

$$M_3 = 2\alpha\left(\alpha - \frac{1}{2}\right) M_0 - 4\alpha(\alpha-1)M_1 + 2(\alpha-1)\left(\alpha - \frac{1}{2}\right) M_2,$$

$$\sum_{i=0}^{2} s_i = 2\alpha\left(\alpha - \frac{1}{2}\right) - 4\alpha(\alpha-1) + 2(\alpha-1)\left(\alpha - \frac{1}{2}\right) = 1.$$

Calculations in nodes:

$\alpha = p_0 = 1$: $M_3 \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = M_0 \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_3 \end{bmatrix}$;

$\alpha = p_1 = 1/2$: $M_3 \begin{bmatrix} x_2 \\ x_4 \end{bmatrix} = M_1 \begin{bmatrix} x_2 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_2 \\ y_4 \end{bmatrix}$;

$\alpha = p_2 = 0$: $M_3 \begin{bmatrix} x_3 \\ x_5 \end{bmatrix} = M_2 \begin{bmatrix} x_3 \\ x_5 \end{bmatrix} = \begin{bmatrix} y_3 \\ y_5 \end{bmatrix}$.

2) seven nodes (x_1,y_1), (x_2,y_2),..., (x_7,y_7), $k = 3$, $p_3 = 0$, $p_2 = 1/3$, $p_1 = 2/3$, $p_0 = 1$:

$$M_0 = \frac{1}{x_1^2 + x_4^2} \begin{bmatrix} x_1 y_1 + x_4 y_4 & x_4 y_1 - x_1 y_4 \\ x_1 y_4 - x_4 y_1 & x_1 y_1 + x_4 y_4 \end{bmatrix},$$

$$M_1 = \frac{1}{x_2^2 + x_5^2} \begin{bmatrix} x_2 y_2 + x_5 y_5 & x_5 y_2 - x_2 y_5 \\ x_2 y_5 - x_5 y_2 & x_2 y_2 + x_5 y_5 \end{bmatrix},$$

$$M_2 = \frac{1}{x_3^2 + x_6^2} \begin{bmatrix} x_3 y_3 + x_6 y_6 & x_6 y_3 - x_3 y_6 \\ x_3 y_6 - x_6 y_3 & x_3 y_3 + x_6 y_6 \end{bmatrix},$$

$$M_3 = \frac{1}{x_4^2 + x_7^2} \begin{bmatrix} x_4 y_4 + x_7 y_7 & x_7 y_4 - x_4 y_7 \\ x_4 y_7 - x_7 y_4 & x_4 y_4 + x_7 y_7 \end{bmatrix},$$

$$s_0 = \frac{(\alpha-0)(\alpha-1/3)(\alpha-2/3)}{(1-0)(1-1/3)(1-2/3)}, \quad s_1 = \frac{(\alpha-0)(\alpha-1/3)(\alpha-1)}{(2/3-0)(2/3-1/3)(2/3-1)},$$

$$s_2 = \frac{(\alpha-0)(\alpha-2/3)(\alpha-1)}{(1/3-0)(1/3-2/3)(1/3-1)}, \quad s_3 = \frac{(\alpha-1)(\alpha-1/3)(\alpha-2/3)}{(0-1)(0-1/3)(0-2/3)}, \sum_{i=0}^{3} s_i = 1,$$

$$M_4 = \tfrac{9}{2}\alpha(\alpha-\tfrac{1}{3})(\alpha-\tfrac{2}{3})M_0 - \tfrac{27}{2}\alpha(\alpha-1)(\alpha-\tfrac{1}{3})M_1 \\ + \tfrac{27}{2}\alpha(\alpha-1)(\alpha-\tfrac{2}{3})M_2 - \tfrac{9}{2}(\alpha-1)(\alpha-\tfrac{1}{3})(\alpha-\tfrac{2}{3})M_3;$$

3) nine nodes (x_1,y_1), (x_2,y_2),..., (x_9,y_9), $k = 4$, $p_4 = 0$, $p_3 = 1/4$, $p_2 = 2/4$, $p_1 = 3/4$, $p_0 = 1$; M_0 built for (x_1,y_1) and (x_5,y_5), M_1 for (x_2,y_2) and (x_6,y_6), M_2 for (x_3,y_3) and (x_7,y_7), M_3 for (x_4,y_4) and (x_8,y_8), M_4 for (x_5,y_5) and (x_9,y_9):

$$M_5 = \tfrac{32}{3}\alpha(\alpha-\tfrac{1}{4})(\alpha-\tfrac{1}{2})(\alpha-\tfrac{3}{4})M_0 - \tfrac{128}{3}\alpha(\alpha-1)(\alpha-\tfrac{1}{4})(\alpha-\tfrac{1}{2})M_1$$
$$+64\alpha(\alpha-1)(\alpha-\tfrac{1}{4})(\alpha-\tfrac{3}{4})M_2 - \tfrac{128}{3}\alpha(\alpha-1)(\alpha-\tfrac{1}{2})(\alpha-\tfrac{3}{4})M_3$$
$$+\tfrac{32}{3}(\alpha-1)(\alpha-\tfrac{1}{4})(\alpha-\tfrac{1}{2})(\alpha-\tfrac{3}{4})M_4.$$

If nodes (x_1,y_1), (x_2,y_2),..., (x_{2k+1},y_{2k+1}) are monotonic in coordinate y_i and these coordinates are settled proportionally at first and second half of the nodes, formula (30) is exchanged:

$$\forall i = 2, ..., k : \frac{y_{k+1} - y_i}{y_{k+1} - y_1} = \frac{y_{2k+1} - y_{k+i}}{y_{2k+1} - y_{k+1}} = p_{i-1}. \tag{35}$$

All calculations (31)-(34) are the same for reverse OHR operators M_i^{-1}:

$$M_{k+1}{}^{-1} = \sum_{i=0}^{k} s_i \cdot M_i^{-1}, \tag{36}$$

$$c_1 = \alpha \cdot y_1 + \beta \cdot y_{k+1}, c_2 = \alpha \cdot y_{k+1} + \beta \cdot y_{2k+1}, \beta = 1 - \alpha,$$

$$\alpha = \frac{y_{k+1} - c_1}{y_{k+1} - y_1} \in [0; 1]. \tag{37}$$

Vector of first coordinates $X(C) = [x(c_1), x(c_2)]^T$ is computed as follows:

$$X(C) = M_{k+1}{}^{-1} \cdot \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = M_{k+1}{}^{-1} \cdot \left(\alpha \begin{bmatrix} y_1 \\ y_{k+1} \end{bmatrix} + \beta \begin{bmatrix} y_{k+1} \\ y_{2k+1} \end{bmatrix} \right). \tag{38}$$

Here is the example for five nodes (x_1,y_1), (x_2,y_2),..., (x_5,y_5), $k = 2$ according to (35) when:

$$p_1 = \frac{y_3 - y_2}{y_3 - y_1} = \frac{y_5 - y_4}{y_5 - y_3} = \frac{1}{4}, \; p_0 = 1, p_2 = 0,$$

$$M_0^{-1} = \frac{1}{y_1^2 + y_3^2} \begin{bmatrix} x_1 y_1 + x_3 y_3 & -x_3 y_1 + x_1 y_3 \\ -x_1 y_3 + x_3 y_1 & x_1 y_1 + x_3 y_3 \end{bmatrix},$$

$$M_1^{-1} = \frac{1}{y_2^2 + y_4^2} \begin{bmatrix} x_2 y_2 + x_4 y_4 & -x_4 y_2 + x_2 y_4 \\ -x_2 y_4 + x_4 y_2 & x_2 y_2 + x_4 y_4 \end{bmatrix},$$

$$M_2^{-1} = \frac{1}{y_3^2 + y_5^2} \begin{bmatrix} x_3 y_3 + x_5 y_5 & -x_5 y_3 + x_3 y_5 \\ -x_3 y_5 + x_5 y_3 & x_3 y_3 + x_5 y_5 \end{bmatrix},$$

$$s_0 = \frac{(\alpha-0)(\alpha-0.25)}{(1-0)(1-0.25)}, \quad s_1 = \frac{(\alpha-0)(\alpha-1)}{(0.25-0)(0.25-1)}, \quad s_2 = \frac{(\alpha-1)(\alpha-0.25)}{(0-1)(0-0.25)}, \sum_{i=0}^{2} s_i = 1,$$

$$M_3^{-1} = \frac{4}{3}\alpha\left(\alpha - \frac{1}{4}\right) M_0^{-1} - \frac{16}{3}\alpha(\alpha-1)M_1^{-1} + 4(\alpha-1)\left(\alpha - \frac{1}{4}\right) M_2^{-1}.$$

We summarize this section in the following algorithm.

Algorithm 2

Input: Set of interpolation nodes $\{(x_j,y_j), j = 1,2,...,n; n = 2k+1, k \geq 2\}$ *such as:*

a) nodes are settled at each local extremum and at least one point between two successive local extrema;

b) nodes (x_j,y_j) *satisfy (30) or (35).*

Step 1. If nodes satisfy (30) then build operators M_i *for nodes* (x_{i+1},y_{i+1}), (x_{k+i+1},y_{k+i+1}) *from (5),* $i = 0,1,..., k$.

If nodes satisfy (35) then build operators M_i^{-1} *for nodes* (x_{i+1},y_{i+1}), (x_{k+i+1},y_{k+i+1}) *from (9),* $i = 0,1,..., k$.

Step 2. Calculate $p_1,..., p_{k-1}$ *from (30) or (35) with* $p_0 = 1$ and $p_k = 0$.

Step 3. Determine the number of points to be reconstructed $K > 0$ *between nodes* (x_1,y_1) *and* (x_{k+1},y_{k+1}), *let* $m = 1$.

Step 4. Compute $\alpha \in [0;1]$ *from (33) or (37) for* c_1, c_2.

Step 5. Calculate si from (32), $i = 0,1,..., k$.

Step 6. Build M_{k+1} *from (31) or* $M_{k+1}{}^{-1}$ *from (36).*

Step 7. Compute unknown coordinates Y(C) from (34) or X(C) from (38).

Step 8. If $m < K$, *set* $m = m+1$ *and go to Step 4. Otherwise, stop.*

The number of reconstructed points in Algorithm 2 is $2K$. Section 5 showed that the MHR method defined by Algorithm 2 gives the possibility of using OHR operators of dimension $N = 2$, with lower computational costs than operators of dimension $N = 4$ or 8, and that the reconstructed curve is "almost-smooth": not smooth only in the central node (x_{k+1},y_{k+1}). Section 6 deals with the problem of curve length estimation and describes some numerical results using MHR method.

6. Numerical results

Selection of the nodes is a key factor in the process of interpolation. Also the length estimation depends on the nodes. Having nodes (x_1,y_1), (x_2,y_2),..., (x_n,y_n) in the MHR method (Algorithm 1 or Algorithm 2), it is possible to compute as many curve points as we want for any parameter $\alpha \in [0;1]$.

Assume that L is the number of reconstructed points plus n nodes. So the curve consists of L points that can be indexed (x_1',y_1'), (x_2',y_2'),..., (x_L',y_L'), where $(x_1',y_1') = (x_1,y_1)$ and $(x_L',y_L') = (x_n,y_n)$. The length of the curve consisting of L points is estimated:

$$d(L) = \sum_{i=1}^{L-1} \sqrt{(x_{i+1}' - x_i')^2 + (y_{i+1}' - y_i')^2}. \tag{39}$$

For any accuracy of length estimation $\epsilon > 0$, it is possible to use the MHR method (Algorithm 1 or Algorithm 2) with a suitable number and location of n nodes and the reconstruct curve consists of L and L_1 points, where

$$|d(L) - d(L_1)| < \varepsilon.$$

So there is no need to compute

$$\int_{x_1}^{x_n} \sqrt{1 + [f'(x)]^2} dx \tag{40}$$

as the length of curve *f*. Formula (39) has lower computational cost then (40).

This section deals with application of MHR method for curves: $f(x) = |x|$, $f(x) = 1/x$, $f(x) = 1/(1+25x^2)$. As we know, these functions cause problems for classical (polynomial) interpolation. The function $f(x) = |x|$ fails to be differentiable at point (0;0). If point (0;0) is one of the interpolation nodes, then precise polynomial (Newton or Lagrange) interpolation of the absolute value function is very hard. MHR method via Algorithm 2 gives optimistic results for this function. Figure 3 shows reconstructed points for 9 nodes: x= -1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75,1.

After reconstruction of 22 points (Fig. 3), together with $n = 9$ nodes, we can estimate the length of curve (39): $d(L) =$ 2.859 for $n = 9$, whereas precise length (40) is $2\sqrt{2} = 2.828$.

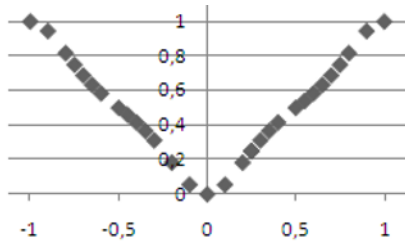


Figure 3. Twenty two interpolated points of function $f(x) = |x|$ using MHR method (Algorithm 2) together with 9 nodes.

The function $f(x) = 1/(1+25x^2)$ is connected with the Runge's phenomenon during classical polynomial interpolation:

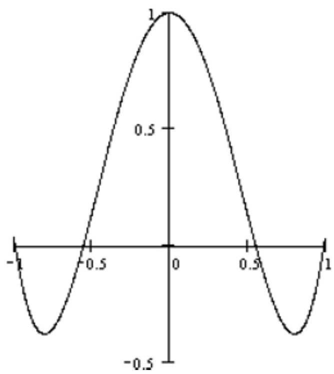


Figure 4. Lagrange interpolation polynomial differs extremely from the shape of function $f(x)=1/(1+25x^2)$.

Algorithm 2 for function $f(x) = 1/(1+25x_2)$ and nodes equidistance in [-1;1] gives us reconstructed curve without oscillations close to $x = 1$ or $x = -1$. Figure 5 shows reconstructed points for 5 nodes (x = -1, -1/2, 0, 1/2, 1), 7 nodes (x = -1, -2/3, -1/3, 0, 1/3, 2/3, 1) and 9 (x = -1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1) interpolation nodes respectively.

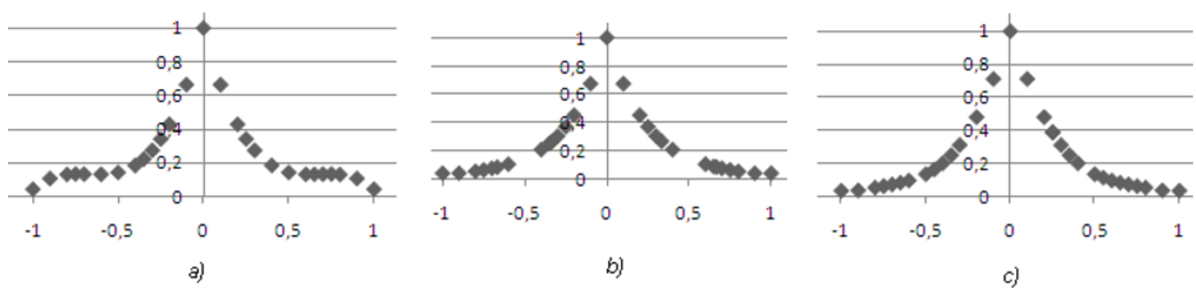


Figure 5. Twenty two interpolated points of function $f(x) = 1/(1+25x^2)$ using MHR method (Algorithm 2) together with 5 nodes (a), 7 nodes (b) and 9 nodes (c).

Having 22 reconstructed points (Fig. 5) together with n = 5 (a), 7 (b) and 9 (c) nodes, we can estimate the length of curve (39):

$$d(L) = 3.120 \text{ for } n = 5;$$
$$d(L) = 3.073 \text{ for } n = 7;$$
$$d(L) = 3.066 \text{ for } n = 9.$$

Precise length (40) is 3.084.
The function *f(x)* = *1*/*x* is an example when the graph of the interpolated function differs considerably from the shape of the interpolating polynomials. Classical interpolation is very hard because of the existing local extrema of the polynomial:

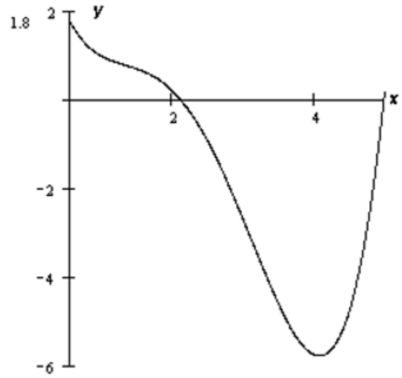


Figure 6. Lagrange interpolation polynomial for nodes (5;0.2), (5/3;0.6), (1;1), (5/7;1.4), (5/9;1.8) differs extremely from the shape of function *f(x) = 1/x*.

Here is the application of Algorithm 2 for function *f(x)* = *1*/*x* and nine nodes equidistance in second coordinate: $y = 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8$.

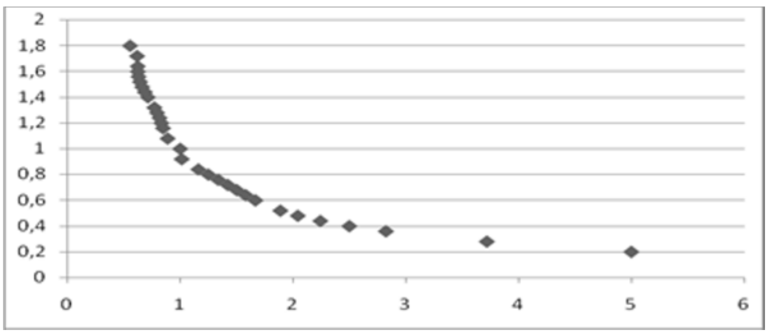


Figure 7. Twenty two interpolated points of function *f(x) = 1/x* using MHR method (Algorithm 2) together with 9 nodes.

Comparing MHR method with Bezier curves one important fact is worth mentioning: a small change of one characteristic point can result a big change of Bezier curves. This feature does not occur in MHR method.

7. Conclusions and future extensions

This paper confirms the expected benefits following from the proposed approach in comparison with classical interpolation methods. The results of the novel MHR method leads to the possibility of direct applications of the proposed method, for example in numerical methods: length of the curve, solving of nonlinear equations and curve extrapolation; computer graphics: object transformation and graph of the function; image processing: shape representation [19, 21], shape coefficients [17] and contour parameterization [23]; artificial intelligence: object recognition [22]. The method of Hurwitz-Radon Matrices in the first version (Algorithm 1 for non-proportional distributed nodes) or te second version (Algorithm 2 for proportional distribution of the nodes) leads to curve interpolation depending on the number of nodes and location of the nodes. No characteristic features of the curve are important in the MHR method: failing to be differentiable at any point, the Runge's phenomenon or differences from the shape of polynomials. These features are very significant for classical polynomial interpolations.

The MHR method gives the possibility of reconstruction a a curve consisting of several parts, for example a closed curve (contour). The only condition is to have a set of nodes for each part of a curve according to assumptions in Algorithm 1 or Algorithm 2. These algorithms have difficulties with smooth curves: a curve interpolated by Algorithm 1 is not smooth in all nodes, whereas curve interpolated by Algorithm 2 is "almost-smooth": not smooth in the central node (also not smooth at the first node and last node during interpolation of a curve consisting of several parts).

The MHR method is useful in discrete mathematics and computer sciences: a graph of a function or curve is never done precisely on the screen. Having a finite set of pixels, the interpolated curve is seen appropriately with points where it is not smooth. A suitable set of nodes is responsible for a graph in discrete mathematics and for good estimation of curve length.

Curve modelling [20] by the MHR method is connected with the possibility of changing the node coordinates and reconstruction of a new curve or contour for a new set of nodes, no matter what the shape of the curve or contour to be reconstructed is. Main features of the MHR method are listed below:

1) accuracy of curve reconstruction is dependent on number of nodes and the method of choosing nodes;

2) reconstruction of a curve consisting of L points has a computational cost of rank $O(L)$;

3) Algorithm 1 is dealing with local operators: average OHR operators M_2 or M_2^{-1} (13)-(14) are built by successive 4, 8 or 16 nodes, has smaller computational costs than using all nodes; significant is the fact that changing node coordinates (x_i,y_i) with for example index $i = 2$, the calculated values of points' coordinates between nodes with index, for example $i = 25$ and $i = 26$, will be unchanged.

Future works are connected with: geometrical transformations of curves (translations, rotations, scaling) - only nodes are transformed and new curves (for example contour of the object) for new nodes can be reconstructed; estimation of an object area in the plane using nodes of the object's contour; possibility of applying te MHR method to three-dimensional curves [15, 16]. Future works will also address smoothing the curve in all nodes and parameterization of whole curve.

References

[1] Atkinson K.A., An Introduction to Numerical Analysis, 2nd ed.(John Wiley and Sons, 1988)
[2] Catmull E., Rom R., A class of local interpolating splines, Comput.-Aided Geom. Des., 317-326, 1974
[3] Cavaretta A. et al., Subdivision Algorithms, Math. Meth. CAGD, 543-581, 1989
[4] Citko W., Jakóbczak D., Sieńko W., On Hurwitz-Radon Matrices Based Signal Processing, In: Proceedings of Signal Processing Conference, 19-24 (Poznań, Poland, 2005)
[5] Dahlquist G., Bjoerck A., Numerical Methods (Englewood Cliffs N.J., Prentice-Hall, 1974)
[6] Dejdumrong N., A shape preserving verification techniques for parametric curves, Comput. Graphics, Imaging Visualization, 163-168, 2007
[7] Dryja M., Jankowska J., Jankowski M., Survey of Numerical Methods and Algorithms. Part II. (WNT, Warsaw, 1982) (in Polish)
[8] Dyn N., Levin D., Gregory J. A., A 4-point interpolatory subdivision scheme for curve design, Comput.-Aided Geom. Des., 4, 257-268, 1987
[9] Eckmann B., GruppentheoretischerBeweis des Satzes von Hurwitz-Radon uber die Komposition der quadratischen Formen, Comment. Math. Helv., 15, 358-366, 1942/43
[10] Eckmann B., Topology, algebra, analysis- relations and missing links, Notices Amer. Math. Soc., 46, 520-527, 1999
[11] Fortuna Z., Macukow B., Wąsowski J., Numerical methods (Warsaw, Poland, WNT, 1982) (in Polish)
[12] Fritsch F. N., Carlson R. E., Monotone piecewise cubic interpolation, SIAM J. Numer. Anal., 17, 238-246, 1980
[13] Hurwitz A., Über die Komposition quadratischer Formen von beliebig vielen Variabeln, Nachr. Ges. d. Wiss. Göttingen, 309-316, 1898
[14] Hurwitz A., ÃIJber die Komposition der quadratischen Formen, Math. Ann. Bd., 88, 1-25, 1923
[15] Jakóbczak D., 2D and 3D image modeling using Hurwitz-Radon matrices, Pol. J. Env. Stud., 16, 104-107, 2007
[16] Jakóbczak D., Kosiński W., Hurwitz-Radon Operator in monochromatic medical image reconstruction, J. Med. Informatics Technol., 11, 69-78, 2007

[17] Jakóbczak, D., Shape representation and shape coefficients via method of Hurwitz-Radon Matrices, Lect. Notes Comput. Sci., 6374, 411-419, 2010

[18] Jakóbczak, D., Curve interpolation using Hurwitz-Radon matrices, Pol. J. Env. Stud. 18, 126-130, 2009

[19] Jakóbczak, D.: Application of Hurwitz-Radon matrices in shape representation, In: Banaszak, Z., Świć, A. (eds.) Applied Computer Science: Modelling of Production Processes 1, 63-74 (Lublin University of Technology Press, Lublin, 2010)

[20] Jakóbczak, D., Object modeling using method of Hurwitz-Radon Matrices of rank k, In: Wolski, W., Borawski, M. (eds.) Computer Graphics: Selected Issues, 79-90 (University of Szczecin Press, Szczecin, 2010)

[21] Jakóbczak, D., Implementation of Hurwitz-Radon matrices in shape representation, In: Choraś, R.S. (ed.) Advances in Intelligent and Soft Computing 84, Image Processing and Communications: Challenges 2, 39-50 (Springer-Verlag, Berlin Heidelberg, 2010)

[22] Jakóbczak D., Object Recognition via Contour Points Reconstruction Using Hurwitz-Radon Matrices, In: Józefczyk J., Orski D. (eds.) Knowledge-Based Intelligent System Advancements: Systemic and Cybernetic Approaches, 87-107 (IGI Global, Hershey PA, USA, 2011)

[23] Jakóbczak, D., Curve parameterization and curvature via method of Hurwitz-Radon Matrices, Image Process. Commun. Int. J. 1-2, 49-56, 2011

[24] Jankowska J., Jankowski M., Survey of Numerical Methods and Algorithms. Part I. (WNT, Warsaw, 1981) (in Polish)

[25] Kozera R., Curve modeling via interpolation based on multidimensional reduced data, (Gliwice, Poland, Silesian University of Technology Press, 2004)

[26] Lang S., Algebra (Addison-Wesley Publishing Company, 1970)

[27] Piegl L., Tiller W., The NURBS Book (Springer-Verlag, Berlin Heidelberg 1995-1997)

[28] Poggio T., Smale S., The mathematics of learning: dealing with data, Notices Amer. Math. Soc., 50, 537-544, 2003

[29] Radon J., Lineare Scharen orthogonaler Matrizen, Abh. Sem. Hamburg I, 1-14, 1923

[30] Ralston A., A First Course in Numerical Analysis (McGraw-Hill Book Company, 1965)

[31] Reif U., A unified approach to subdivision algorithms near extraordinary vertices, Comput.-Aided Geom. Des., 12, 153-174, 1995

[32] Rogers D. F., An Introduction to NURBS with Historical Perspective (Morgan Kaufmann Publishers, 2001)

[33] Schatzman M. , Numerical Analysis: A Mathematical Introduction (Clarendon Press, Oxford, 2002)

[34] Schumaker L.L., Spline Functions: Basic Theory (Cambridge Mathematical Library, 2007)

[35] Seokjeong Hyun, Cheol Kim, An efficient shape-preserving curve fitting technique for complex structure, Commun. Numer. Methods Eng., 20, 737-746, 2004

[36] Sieńko W., Citko W., Jakóbczak D., Learning and system modeling via Hamiltonian Neural Networks, Lect. Notes Artif. Intell., 3070, 266-271, 2004

[37] Stoer J., Einfuehrung in die Numerische Mathematik I (Berlin, Springer, 1972)

[38] Stoer J., Bulirsch R., Einfuehrung in die Numerische Mathematik II (Berlin, Springer, 1973)

[39] Süli E., Mayers D., An Introduction to Numerical Analysis (Cambridge University Press, 2003)

[40] Wu Zongmin, Ye Qing, Liu Jianpin, The subdivision algorithm for generating curves and its properties, Appl. Math. J. Chin. Univ., 9B, 169-176, 1994