

Probabilistic 2D Point Interpolation and Extrapolation via Data Modeling

Dariusz Jacek Jakóbczak

Department of Electronics and Computer Science, Technical University of Koszalin,

Sniadeckich 2, 75-453 Koszalin, Poland

E-mail: dariusz.jakobczak@tu.koszalin.pl

Keywords: 2D data interpolation, Hurwitz-Radon matrices, MHR method, probabilistic modeling, curve extrapolation

Received: June 25, 2014

Mathematics and computer science are interested in methods of 2D curve interpolation and extrapolation using the set of key points (knots). A proposed method of Hurwitz-Radon Matrices (MHR) is such a method. This novel method is based on the family of Hurwitz-Radon (HR) matrices which possess columns composed of orthogonal vectors. Two-dimensional curve is interpolated via different functions as probability distribution functions: polynomial, sinus, cosine, tangent, cotangent, logarithm, exponent, arcsin, arccos, arctan, arcctg or power function, also inverse functions. It is shown how to build the orthogonal matrix operator and how to use it in a process of curve reconstruction.

Povzetek: Opisana je nova metoda 2D interpolacije in ekstrapolacije krivulj.

1 Introduction

Curve interpolation and extrapolation [1] represents one of the most important problems in mathematics: how to model the curve [2] via discrete set of two-dimensional points [3]? Also the matter of curve representation and parameterization is still opened in mathematics and computer sciences [4]. The author wants to approach a problem of curve modeling by characteristic points. Proposed method relies on functional modeling of curve points situated between the basic set of the nodes or outside the nodes. The functions that are used in calculations represent whole family of elementary functions with inverse functions: polynomials, trigonometric, cyclometric, logarithmic, exponential and power function. These functions are treated as probability distribution functions in the range [0,1]. Nowadays methods apply mainly polynomial functions, for example Bernstein polynomials in Bezier curves, splines and NURBS [5]. Numerical methods for data interpolation or extrapolation are based on polynomial or trigonometric functions, for example Lagrange, Newton, Aitken and Hermite methods. These methods have some weak sides [6] and are not sufficient for curve interpolation and extrapolation in the situations when the curve cannot be build by polynomials or trigonometric functions. Proposed 2D curve interpolation and extrapolation is the functional modeling via any elementary functions and it helps us to fit the curve during the computations.

The main contributions of the paper are dealing with presentation the method that connects such problems as: interpolation, extrapolation, modeling, numerical methods and probabilistic methods. This is new approach to these problems. Differences from the previous papers of the author are connected with calculations without

matrices ($N = 1$), new probabilistic distribution functions and novel look on shape modeling and curve reconstruction.

The method of Hurwitz-Radon Matrices (MHR) requires minimal assumptions: the only information about a curve is the set of at least two nodes. Proposed method of Hurwitz-Radon Matrices (MHR) is applied in curve modeling via different coefficients: polynomial, sinusoidal, cosinusoidal, tangent, cotangent, logarithmic, exponential, arcsin, arccos, arctan, arcctg or power. Function for MHR calculations is chosen individually at each interpolation and it represents probability distribution function of parameter $\in [0,1]$ for every point situated between two interpolation knots. MHR method uses two-dimensional vectors (x,y) for curve modeling - knots $(x_i,y_i) \in \mathbf{R}^2$ in MHR method:

1. MHR version with no matrices ($N = 1$) needs 2 knots or more;
2. At least five knots (x_1,y_1) , (x_2,y_2) , (x_3,y_3) , (x_4,y_4) and (x_5,y_5) if MHR method is implemented with matrices of dimension $N = 2$;
3. For more precise modeling knots ought to be settled at key points of the curve, for example local minimum or maximum and at least one node between two successive local extrema.

Condition 2 is connected with important features of MHR method: MHR version with matrices of dimension $N = 2$ (MHR-2) requires at least five nodes, MHR version with matrices of dimension $N = 4$ (MHR-4) needs at least nine nodes and MHR version with matrices of dimension $N = 8$ (MHR-8) requires at least 17 nodes. Condition 3 means for example the highest point of the curve in a particular orientation, convexity changing or curvature extrema. So this paper wants to answer the

question: how to interpolate end extrapolate the curve by a set of knots?

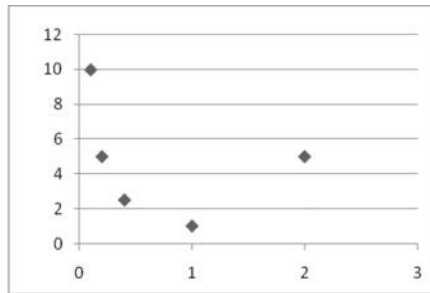


Figure 1: Knots of the curve before modelling.

Coefficients for curve modeling are computed using probability distribution functions: polynomials, power functions, sinus, cosine, tangent, cotangent, logarithm, exponent or arcsin, arccos, arctan, arcctg.

2 Probabilistic 2D interpolation and extrapolation

The method of Hurwitz – Radon Matrices (MHR) is computing points between two successive nodes of the curve: calculated points are interpolated and parameterized for real number $\alpha \in [0,1]$ in the range of two successive nodes. Data extrapolation is possible for $\alpha < 0$ or $\alpha > 1$. MHR calculations are dealing with square matrices of dimension $N = 1, 2, 4$ or 8 . Matrices $A_i, i=1,2,...m$ satisfying

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

are called a family of Hurwitz - Radon matrices. They were discussed by Adolf Hurwitz and Johann Radon separately in 1923. A family of Hurwitz-Radon (HR) matrices [7] are skew-symmetric: $A_i^T = -A_i$ and $A_i^{-1} = -A_i$. Only for dimensions $N = 1, 2, 4$ or 8 the family of HR matrices consists of $N - 1$ matrices. For $N = 1$ there is no matrices but only calculations with real numbers. For $N=2$:

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

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

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

For $N = 8$ we have seven HR matrices with elements $0, \pm 1$. So far HR matrices have found applications in Space-Time Block Coding (STBC) [8] and orthogonal design [9], in signal processing [10] and Hamiltonian Neural Nets [11].

How coordinates of knots are applied for interpolation and extrapolation? If knots are represented by the following points $\{(x_i, y_i), i = 1, 2, \dots, n\}$ then HR matrices combined with the identity matrix I_N are used to build the orthogonal Hurwitz - Radon Operator (OHR).

For point $p_1=(x_1, y_1)$ and $x_1 \neq 0$ OHR of dimension $N = 1$ is the matrix (real number) M_1 :

$$M_1(p_1) = \frac{1}{x_1^2} [x_1 \cdot y_1] = \frac{y_1}{x_1} \tag{0}$$

For points $p_1=(x_1, y_1)$ and $p_2=(x_2, y_2)$ OHR of dimension $N=2$ is build via matrix M_2 :

$$M_2(p_1, p_2) = \frac{1}{x_1^2 + x_2^2} \begin{bmatrix} x_1 y_1 + x_2 y_2 & x_2 y_1 - x_1 y_2 \\ x_1 y_2 - x_2 y_1 & x_1 y_1 + x_2 y_2 \end{bmatrix} \tag{1}$$

For points $p_1=(x_1, y_1), p_2=(x_2, y_2), p_3=(x_3, y_3)$ and $p_4=(x_4, y_4)$ OHR M_4 of dimension $N = 4$ is introduced:

$$M_4(p_1, p_2, p_3, p_4) = \frac{1}{x_1^2 + x_2^2 + x_3^2 + x_4^2} \begin{bmatrix} u_0 & u_1 & u_2 & u_3 \\ -u_1 & u_0 & -u_3 & u_2 \\ -u_2 & u_3 & u_0 & -u_1 \\ -u_3 & -u_2 & u_1 & u_0 \end{bmatrix} \tag{2}$$

where

$$\begin{aligned} u_0 &= x_1 y_1 + x_2 y_2 + x_3 y_3 + x_4 y_4, \\ u_1 &= -x_1 y_2 + x_2 y_1 + x_3 y_4 - x_4 y_3, \\ u_2 &= -x_1 y_3 - x_2 y_4 + x_3 y_1 + x_4 y_2, \\ u_3 &= -x_1 y_4 + x_2 y_3 - x_3 y_2 + x_4 y_1. \end{aligned}$$

For knots $p_1=(x_1, y_1), p_2=(x_2, y_2), \dots$ and $p_8=(x_8, y_8)$ OHR M_8 of dimension $N = 8$ is constructed [12] similarly as (1) and (2):

$$M_8(p_1, p_2, \dots, p_8) = \frac{1}{\sum_{i=1}^8 x_i^2} \begin{bmatrix} u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \\ -u_1 & u_0 & u_3 & -u_2 & u_5 & -u_4 & -u_7 & u_6 \\ -u_2 & -u_3 & u_0 & u_1 & u_6 & u_7 & -u_4 & -u_5 \\ -u_3 & u_2 & -u_1 & u_0 & u_7 & -u_6 & u_5 & -u_4 \\ -u_4 & -u_5 & -u_6 & -u_7 & u_0 & u_1 & u_2 & u_3 \\ -u_5 & u_4 & -u_7 & u_6 & -u_1 & u_0 & -u_3 & u_2 \\ -u_6 & u_7 & u_4 & -u_5 & -u_2 & u_3 & u_0 & -u_1 \\ -u_7 & -u_6 & u_5 & u_4 & -u_3 & -u_2 & u_1 & u_0 \end{bmatrix} \tag{3}$$

where

$$\underline{u} = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ -y_2 & y_1 & -y_4 & y_3 & -y_6 & y_5 & y_8 & -y_7 \\ -y_3 & y_4 & y_1 & -y_2 & -y_7 & -y_8 & y_5 & y_6 \\ -y_4 & -y_3 & y_2 & y_1 & -y_8 & y_7 & -y_6 & y_5 \\ -y_5 & y_6 & y_7 & y_8 & y_1 & -y_2 & -y_3 & -y_4 \\ -y_6 & -y_5 & y_8 & -y_7 & y_2 & y_1 & y_4 & -y_3 \\ -y_7 & -y_8 & -y_5 & y_6 & y_3 & -y_4 & y_1 & y_2 \\ -y_8 & y_7 & -y_6 & -y_5 & y_4 & y_3 & -y_2 & y_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} \tag{4}$$

and $\underline{u} = (u_0, u_1, \dots, u_7)^T$ (4). OHR operators M_N (0)-(3) satisfy the condition of interpolation

$$M_N \mathbf{x} = \mathbf{y} \tag{5}$$

for $\mathbf{x} = (x_1, x_2, \dots, x_N)^T \in \mathbf{R}^N, \mathbf{x} \neq \mathbf{0}, \mathbf{y} = (y_1, y_2, \dots, y_N)^T \in \mathbf{R}^N$ and $N = 1, 2, 4$ or 8 .

2.1 Distribution functions in MHR interpolation and extrapolation

Points settled between the nodes are computed [13] using MHR method [14]. Each real number $c \in [a;b]$ is calculated by a convex combination

$$c = \alpha \cdot a + (1 - \alpha) \cdot b \quad \text{for} \\ r = \frac{b - c}{b - a} \in [0,1]. \quad (6)$$

The weighted average OHR operator M of dimension $N = 1, 2, 4$ or 8 is build:

$$M = x \cdot A + (1 - x) \cdot B. \quad (7)$$

The OHR matrix A is constructed (1)-(3) by every second knot $p_1=(x_1=a, y_1), p_3=(x_3, y_3), \dots$ and $p_{2N-1}=(x_{2N-1}, y_{2N-1})$:

$$A = M_N(p_1, p_3, \dots, p_{2N-1}).$$

The OHR matrix B is computed (1)-(3) by knots $p_2=(x_2=b, y_2), p_4=(x_4, y_4), \dots$ and $p_{2N}=(x_{2N}, y_{2N})$:

$$B = M_N(p_2, p_4, \dots, p_{2N}).$$

Vector of first coordinates C is defined for

$$c_i = \alpha \cdot x_{2i-1} + (1 - \alpha) \cdot x_{2i}, \quad i = 1, 2, \dots, N \quad (8)$$

and $C = [c_1, c_2, \dots, c_N]^T$. The formula to calculate second coordinates $y(c_i)$ is similar to the interpolation formula (5):

$$Y(C) = M \cdot C \quad (9)$$

where $Y(C) = [y(c_1), y(c_2), \dots, y(c_N)]^T$. So interpolated value $y(c_i)$ from (9) depends on two, four, eight or sixteen ($2N$) successive nodes. For example $N=1$ results in computations without matrices:

$$A = M_1(p_1) = \frac{y_1}{x_1}, \quad B = M_1(p_2) = \frac{y_2}{x_2},$$

$$C = c_1 = \alpha \cdot x_1 + (1 - \alpha) \cdot x_2,$$

$$Y(C) = y(c_1) = (x \frac{y_1}{x_1} + (1 - x) \frac{y_2}{x_2}) \cdot c_1,$$

$$y(c_1) = r \cdot x \cdot y_1 + (1 - r)(1 - x)y_2 + x(1 - r) \frac{y_1}{x_1} x_2 + r(1 - x) \frac{y_2}{x_2} x_1. \quad (10)$$

Formula (10) shows a clear calculation for interpolation of any point between two successive nodes (x_1, y_1) and (x_2, y_2) . Key question is dealing with coefficient r in (7). Basic MHR version means $r = x$ and then (10):

$$y(c_1) = r^2 \cdot y_1 + (1 - r)^2 y_2 + r(1 - r) (\frac{y_1}{x_1} x_2 + \frac{y_2}{x_2} x_1). \quad (11)$$

Formula (11) represents the simplest way of MHR calculations ($N = 1, r = x$) and it differs from linear interpolation $y(c) = r \cdot y_1 + (1 - r)y_2$. MHR is not a linear interpolation.

Each interpolation requires specific distribution of parameter r (7) and depends on parameter $s \in [0,1]$:

$$r = F(s), \quad F: [0,1] \rightarrow [0,1], \quad F(0) = 0, \quad F(1) = 1$$

and F is strictly monotonic.

Coefficient r is calculated using different functions (polynomials, power functions, sinus, cosine, tangent, cotangent, logarithm, exponent, arcsin, arccos, arctan or arcctg, also inverse functions) and choice of function is connected with initial requirements and curve specifications. Different values of coefficient r are

connected with applied functions $F(\cdot)$. These functions (12)-(41) represent the probability distribution functions for random variable $s \in [0,1]$ and real number $s > 0$:

$$1. \text{ power function} \\ r = s^s \quad \text{with} \quad s > 0. \quad (12)$$

For $s = 1$: basic version of MHR method when $r = x$.

$$2. \text{ sinus} \\ r = \sin(s \cdot \pi / 2), \quad s > 0 \quad (13)$$

$$\text{or} \\ r = \sin^s(s \cdot \pi / 2), \quad s > 0. \quad (14)$$

$$\text{For } s = 1: \quad r = \sin(\pi / 2). \quad (15)$$

$$3. \text{ cosine} \\ r = 1 - \cos(s \cdot \pi / 2), \quad s > 0 \quad (16)$$

$$\text{or} \\ r = 1 - \cos^s(s \cdot \pi / 2), \quad s > 0. \quad (17)$$

$$\text{For } s = 1: \quad r = 1 - \cos(\pi / 2). \quad (18)$$

$$4. \text{ tangent} \\ r = \tan(s \cdot \pi / 4), \quad s > 0 \quad (19)$$

$$\text{or} \\ r = \tan^s(s \cdot \pi / 4), \quad s > 0. \quad (20)$$

$$\text{For } s = 1: \quad r = \tan(\pi / 4). \quad (21)$$

$$5. \text{ logarithm} \\ r = \log_2(s + 1), \quad s > 0 \quad (22)$$

$$\text{or} \\ r = \log_2^s(s + 1), \quad s > 0. \quad (23)$$

$$\text{For } s = 1: \quad r = \log_2(2 + 1). \quad (24)$$

$$6. \text{ exponent} \\ r = \left(\frac{a^r - 1}{a - 1}\right)^s, \quad s > 0 \text{ and } a > 0 \text{ and } a \neq 1. \quad (25)$$

$$\text{For } s = 1 \text{ and } a = 2: \quad r = 2 - 1. \quad (26)$$

$$7. \text{ arc sine} \\ r = 2/\pi \cdot \arcsin(s), \quad s > 0 \quad (27)$$

$$\text{or} \\ r = (2/\pi \cdot \arcsin(s))^s, \quad s > 0. \quad (28)$$

$$\text{For } s = 1: \quad r = 2/\pi \cdot \arcsin(1). \quad (29)$$

$$8. \text{ arc cosine} \\ r = 1 - 2/\pi \cdot \arccos(s), \quad s > 0 \quad (30)$$

$$\text{or} \\ r = 1 - (2/\pi \cdot \arccos(s))^s, \quad s > 0. \quad (31)$$

$$\text{For } s = 1: \quad r = 1 - 2/\pi \cdot \arccos(0). \quad (32)$$

$$9. \text{ arc tangent} \\ r = 4/\pi \cdot \arctan(s), \quad s > 0 \quad (33)$$

$$\text{or} \\ r = (4/\pi \cdot \arctan(s))^s, \quad s > 0. \quad (34)$$

$$\text{For } s = 1: \quad r = 4/\pi \cdot \arctan(1). \quad (35)$$

$$10. \text{ cotangent} \\ r = \text{ctg}(\pi / 2 - s \cdot \pi / 4), \quad s > 0 \quad (36)$$

$$\text{or} \\ r = \text{ctg}^s(\pi / 2 - s \cdot \pi / 4), \quad s > 0. \quad (37)$$

$$\text{For } s = 1: \quad r = \text{ctg}(\pi / 2 - \pi / 4). \quad (38)$$

11. arc cotangent
 $= 2 - 4/ \cdot \text{arccotg}(\text{ }^s), s > 0$ (39)

or
 $= (2 - 4/ \cdot \text{arccotg}(\text{ }^s), s > 0.$ (40)

For $s = 1$:
 $= 2 - 4/ \cdot \text{arccotg}(\text{ }).$ (41)

Functions used in calculations (12)-(41) are strictly monotonic for random variable $\in [0,1]$ as $= F()$ is probability distribution function. Also inverse function $F^{-1}()$ is appropriate for calculations. Choice of function and value s depends on curve specifications and individual requirements. Interpolating of coordinates for curve points using (6)-(9) is called by author the method of Hurwitz - Radon Matrices (MHR) [15]. So here are five steps of MHR interpolation:

Step 1: Choice of knots at key points.

Step 2: Fixing the dimension of matrices $N = 1, 2, 4$ or 8 : $N = 1$ is the most universal for calculations (it needs only two nodes to compute unknown points between them) and it has the lowest computational costs (10); MHR with $N = 2$ uses four successive nodes to compute unknown coordinate; MHR version for $N = 4$ applies eight successive nodes to get unknown point and MHR with $N = 8$ needs sixteen successive nodes to calculate unknown coordinate (it has the biggest computational costs).

Step 3: Choice of distribution $= F()$: basic distribution for $=$.

Step 4: Determining values of $: = 0.1, 0.2...0.9$ (nine points) or $0.01, 0.02...0.99$ (99 points) or others.

Step 5: The computations (9).

These five steps can be treated as the algorithm of MHR method of curve modeling and interpolation (6)-(9).

Considering nowadays used probability distribution functions for random variable $\in [0,1]$ - one distribution is dealing with the range $[0,1]$: beta distribution. Probability density function f for random variable $\in [0,1]$ is:

$$f(r) = c \cdot r^s \cdot (1-r)^r, s \geq 0, r \geq 0$$
 (42)

When $r = 0$ probability density function (42) represents $f(r) = c \cdot r^s$ and then probability distribution function F is like (12), for example $f(r) = 3r^2$ and $= 3$. If s and r are positive integer numbers then is the polynomial, for example $f(r) = 6r(1-r)$ and $= 3^2 \cdot 2^3$. So beta distribution gives us coefficient in (7) as polynomial because of interdependence between probability density f and distribution F functions:

$$f(r) = F'(r), F(r) = \int_0^r f(t) dt$$
 (43)

For example (43): $f(r) = r \cdot e^r$ and $x = F(r) = (r - 1)e^r + 1$.

What is very important: two curves may have the same set of nodes but different N or results in different interpolations (Fig.6-13). Here are some applications of MHR method with basic version ($=$): MHR-2 is

MHR version with matrices of dimension $N = 2$ and MHR-4 means MHR version with matrices of dimension $N = 4$.

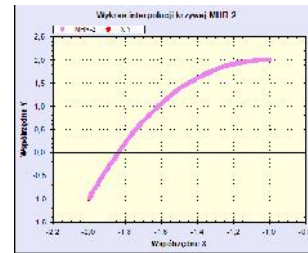


Figure 2: Function $f(x) = x^3 + x^2 - x + 1$ with 396 interpolated points using basic MHR-2 with 5 nodes.

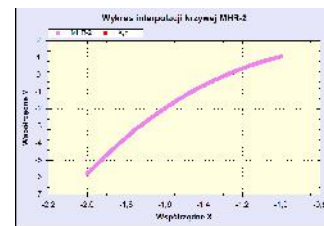


Figure 3: Function $f(x) = x^3 + \ln(7-x)$ with 396 interpolated points using basic MHR-2 with 5 nodes.

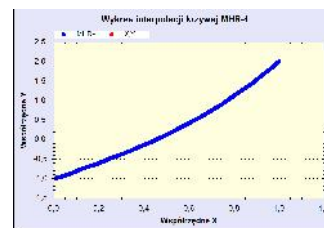


Figure 4: Function $f(x) = x^3 + 2x - 1$ with 792 interpolated points using basic MHR-4 with 9 nodes.

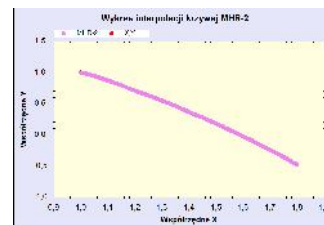


Figure 5: Function $f(x) = 3 - 2^x$ with 396 interpolated points using basic MHR-2 with 5 nodes.

Figures 2-5 show interpolation of continues functions connected with determined formula. So these functions are interpolated and modeled. Without knowledge about the formula, curve interpolation has to implement the coefficients (12)-(43), but MHR is not limited only to these coefficients. Each strictly monotonic function F between points (0;0) and (1;1) can be used in MHR interpolation. MHR 2D data extrapolation is possible for $\alpha < 0$ or $\alpha > 1$.

3 Implementations of 2D probabilistic interpolation

Curve knots (0.1;10), (0.2;5), (0.4;2.5), (1;1) and (2;5) from Fig.1 are used in some examples of MHR interpolation with different . Points of the curve are calculated for $N = 1$ and $=$ (11) in example 1 and with

matrices of dimension $N = 2$ in examples 2 - 8 for $\alpha = 0.1, 0.2, \dots, 0.9$.

Example 1

Curve interpolation for $N = 1$ and $\alpha = 0.1$.

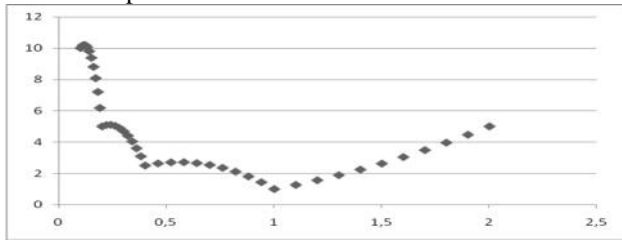


Figure 6: Modeling without matrices ($N = 1$) for nine reconstructed points between nodes.

Example 2

Sinusoidal interpolation with $\alpha = \sin(\pi \cdot x / 2)$.

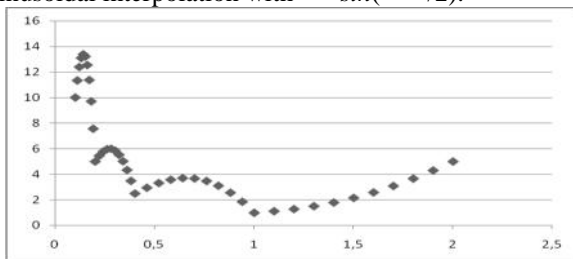


Figure 7: Sinusoidal modeling with nine reconstructed curve points between nodes.

Example 3

Tangent interpolation for $\alpha = \tan(\pi \cdot x / 4)$.

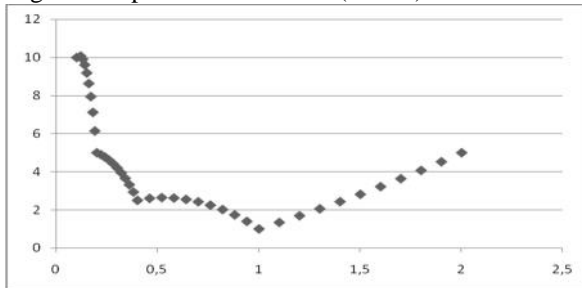


Figure 8: Tangent curve modeling with nine interpolated points between nodes.

Example 4

Tangent interpolation with $\alpha = \tan(\pi \cdot x / 4)$ and $s = 1.5$.

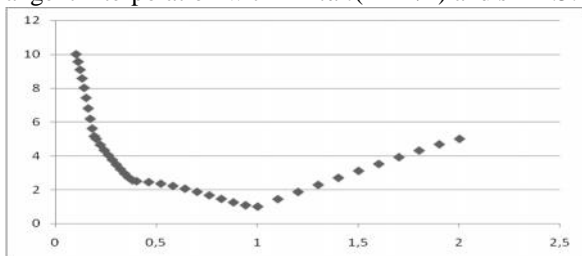


Figure 9: Tangent modeling with nine recovered points between nodes.

Example 5

Tangent curve interpolation for $\alpha = \tan(\pi \cdot x / 4)$ and $s = 1.797$.

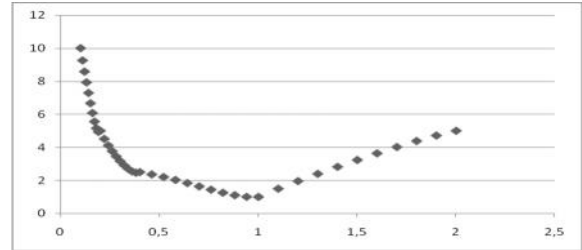


Figure 10: Tangent modeling with nine reconstructed points between nodes.

Example 6

Sinusoidal interpolation with $\alpha = \sin(\pi \cdot x / 2)$ and $s = 2.759$.

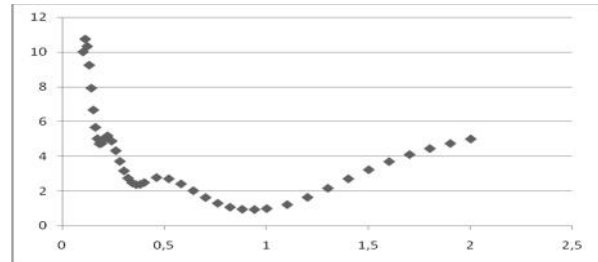


Figure 11: Sinusoidal modeling with nine interpolated curve points between nodes.

Example 7

Power function modeling for $\alpha = x^s$ and $s = 2.1205$.

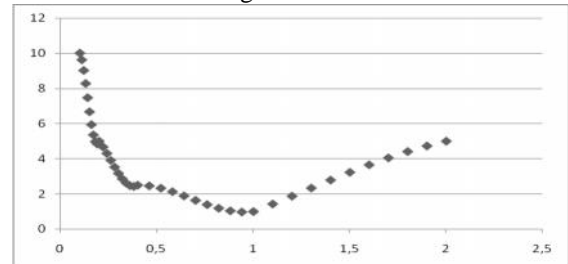


Figure 12: Power function curve modeling with nine recovered points between nodes.

Example 8

Logarithmic curve modeling with $\alpha = \log_2(x + 1)$ and $s = 2.533$.

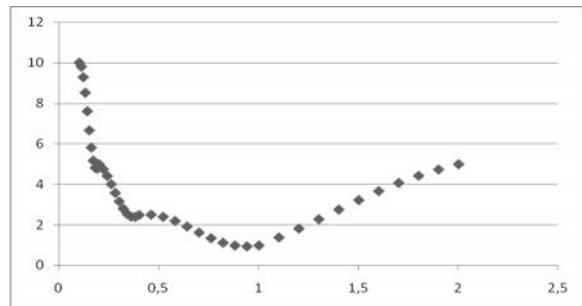


Figure 13: Logarithmic modeling with nine reconstructed points between nodes.

These eight examples demonstrate possibilities of curve interpolation for key nodes. Reconstructed values and interpolated points, calculated by MHR method, are applied in the process of curve modeling. Every curve can be interpolated by some distribution function as

parameter α . This parameter is treated as probability distribution function for each curve.

4 MHR 2D modeling versus polynomial interpolation

4.1 Example 4.1

Let us consider a graph of function $f(x) = 2/x$ in range $[0.4, 1.6]$. There are given five interpolation nodes for $x = 0.4, 0.7, 1.0, 1.3, 1.6$. The curve $y = 2/x$ reconstructed by basic MHR method (12) with $N = 2$ for $s = 1$ looks not precisely:

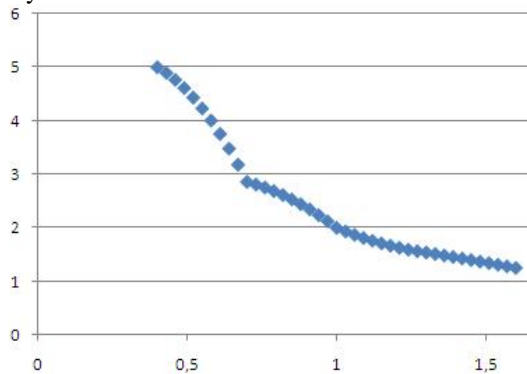


Figure 14: The curve $y = 2/x$ reconstructed by MHR method for $\alpha = 0.5$ (7) and five nodes together with 36 computed points.

Lagrange interpolation polynomial is not to be accepted:

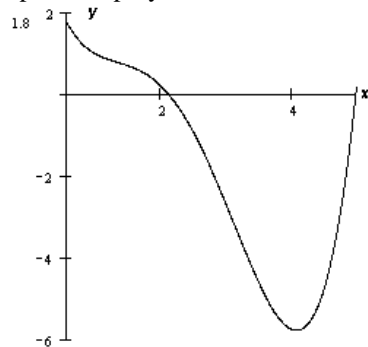


Figure 15: Polynomial interpolation of function $y = 1/x$ is out of acceptance.

For better reconstruction of the curve, appropriate parameter s in MHR method (12) is calculated. Choice of parameter s is connected with comparison of accurate values w_i for function $f(x) = 2/x$ in control points p_i , situated in the middle between interpolation nodes ($\alpha=0.5$), and values in control points p_i computed by MHR method. Control points are settled in the middle between interpolation nodes, because interpolation error of MHR method is the biggest. Choice of coefficient s is done by criterion: difference between precise values w_i and values reconstructed by MHR method is the smallest. Control points p_i in this example are established for $x_i = 0.55, 0.85, 1.15, 1.45$. Four values of the curve are compared for parameter $s = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6,$

$0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0$. The smallest difference is calculated for $s = 1.5$:

$$|w_1 - 3.694| + |w_3 - 1.632| + |w_2 - 2.302| + |w_4 - 1.329| = 0.266 \quad (44)$$

Calculations for average OHR operator $M(7)$ are done:

$$M = \begin{bmatrix} 2.405 & 2.062 \\ -2.062 & 2.405 \end{bmatrix}, \quad M \cdot \begin{bmatrix} p_1 \\ p_3 \end{bmatrix} = \begin{bmatrix} 3.694 \\ 1.632 \end{bmatrix},$$

$$M = \begin{bmatrix} 1.375 & 0.782 \\ -0.782 & 1.375 \end{bmatrix}, \quad M \cdot \begin{bmatrix} p_2 \\ p_4 \end{bmatrix} = \begin{bmatrix} 2.302 \\ 1.329 \end{bmatrix}.$$

Computed values appear in (44). Other results:

- a) $s=0.1$
 $|w_1 - 5.815| + |w_3 - 1.939| + |w_2 - 3.209| + |w_4 - 1.601| = 3.456$
- b) $s=0.2$:
 $|w_1 - 5.587| + |w_3 - 1.906| + |w_2 - 3.111| + |w_4 - 1.572| = 3.068$
- c) $s=0.3$:
 $|w_1 - 5.373| + |w_3 - 1.875| + |w_2 - 3.02| + |w_4 - 1.544| = 2.704$
- d) $s=0.4$:
 $|w_1 - 5.174| + |w_3 - 1.846| + |w_2 - 2.935| + |w_4 - 1.519| = 2.366$
- e) $s=0.5$:
 $|w_1 - 4.988| + |w_3 - 1.819| + |w_2 - 2.856| + |w_4 - 1.495| = 2.05$
- f) $s=0.6$:
 $|w_1 - 4.815| + |w_3 - 1.794| + |w_2 - 2.782| + |w_4 - 1.473| = 1.756$
- g) $s=0.7$:
 $|w_1 - 4.653| + |w_3 - 1.771| + |w_2 - 2.712| + |w_4 - 1.452| = 1.48$
- h) $s=0.8$:
 $|w_1 - 4.503| + |w_3 - 1.749| + |w_2 - 2.648| + |w_4 - 1.433| = 1.225$
- i) $s=0.9$:
 $|w_1 - 4.362| + |w_3 - 1.728| + |w_2 - 2.588| + |w_4 - 1.415| = 1.008$
- j) $s=1$:
 $|w_1 - 4.23| + |w_3 - 1.709| + |w_2 - 2.532| + |w_4 - 1.398| = 0.822$
- k) $s=1.1$:
 $|w_1 - 4.108| + |w_3 - 1.692| + |w_2 - 2.479| + |w_4 - 1.382| = 0.648$
- l) $s=1.2$:
 $|w_1 - 3.994| + |w_3 - 1.675| + |w_2 - 2.43| + |w_4 - 1.368| = 0.51$
- m) $s=1.3$:
 $|w_1 - 3.887| + |w_3 - 1.66| + |w_2 - 2.385| + |w_4 - 1.354| = 0.387$
- n) $s=1.4$:
 $|w_1 - 3.787| + |w_3 - 1.645| + |w_2 - 2.342| + |w_4 - 1.341| = 0.294$
- o) $s=1.6$:
 $|w_1 - 3.608| + |w_3 - 1.619| + |w_2 - 2.265| + |w_4 - 1.318| = 0.298$
- p) $s=1.7$:
 $|w_1 - 3.527| + |w_3 - 1.608| + |w_2 - 2.231| + |w_4 - 1.308| = 0.434$
- q) $s=1.8$:
 $|w_1 - 3.451| + |w_3 - 1.597| + |w_2 - 2.199| + |w_4 - 1.298| = 0.563$
- r) $s=1.9$:
 $|w_1 - 3.381| + |w_3 - 1.587| + |w_2 - 2.169| + |w_4 - 1.289| = 0.682$
- s) $s=2.0$:
 $|w_1 - 3.315| + |w_3 - 1.577| + |w_2 - 2.14| + |w_4 - 1.281| = 0.795$

One can see the changes of interpolation error. Reconstruction of the curve $y=2/x$ by MHR method for $N = 2$ and $\alpha = 1.5$ looks as follows:

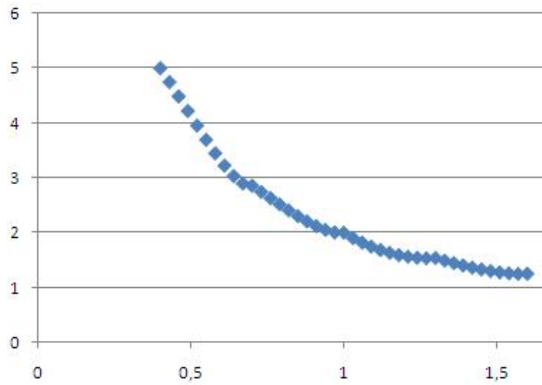


Figure 16: The curve $y = 2/x$ modeled via MHR method (12) for $s = 1.5$ and five nodes together with 36 reconstructed points.

Figure 16 represents the curve $y = 2/x$ more precisely than classical interpolation and Figure 14. If we would like to have better parameter s (with two digits after coma), calculations are done:

- a) $s=1.49$:
 $|w_1 - 3.703| + |w_3 - 1.633| + |w_2 - 2.306| + |w_4 - 1.33| = 0.269$
- b) $s=1.51$:
 $|w_1 - 3.685| + |w_3 - 1.631| + |w_2 - 2.299| + |w_4 - 1.328| = 0.262$
- c) $s=1.52$:
 $|w_1 - 3.677| + |w_3 - 1.629| + |w_2 - 2.295| + |w_4 - 1.327| = 0.261$
- d) $s=1.53$:
 $|w_1 - 3.668| + |w_3 - 1.628| + |w_2 - 2.291| + |w_4 - 1.326| = 0.258$
- e) $s=1.54$:
 $|w_1 - 3.659| + |w_3 - 1.627| + |w_2 - 2.287| + |w_4 - 1.325| = 0.255$
- f) $s=1.55$:
 $|w_1 - 3.65| + |w_3 - 1.626| + |w_2 - 2.284| + |w_4 - 1.324| = 0.251$
- g) $s=1.56$:
 $|w_1 - 3.642| + |w_3 - 1.624| + |w_2 - 2.28| + |w_4 - 1.323| = 0.25$
- h) $s=1.57$:
 $|w_1 - 3.633| + |w_3 - 1.623| + |w_2 - 2.276| + |w_4 - 1.321| = 0.255$
- i) $s=1.58$:
 $|w_1 - 3.625| + |w_3 - 1.622| + |w_2 - 2.273| + |w_4 - 1.32| = 0.268$
- j) $s=1.59$:
 $|w_1 - 3.616| + |w_3 - 1.621| + |w_2 - 2.269| + |w_4 - 1.319| = 0.283$

Model of the curve for $s = 1.56$ in (12) is more accurate than with $s = 1.5$.

Convexity of reconstructed curve is very important factor in MHR method with parameter s . Appropriate choice of parameter s is connected with regulation and controlling of convexity: model of the curve (Fig. 16) preserves monotonicity and convexity.

4.2 Example 4.2

This example considers a graph of function $f(x) = 1/(1 + 5x^2)$ in the range $[-1,1]$:

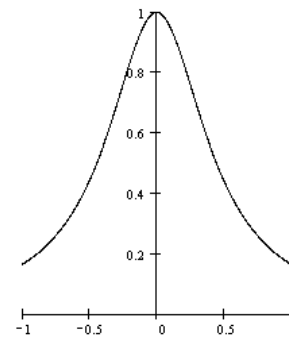


Figure 17: A graph of function $f(x) = 1/(1 + 5x^2)$ in range $[-1,1]$.

There are given five interpolation nodes for $x = -1.0, -0.5, 0, 0.5, 1.0$. It is an example of function with useless Lagrange interpolation polynomial: Runge phenomenon and unpleasant two minima.

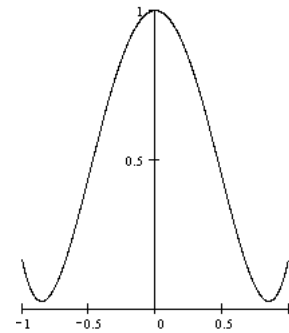


Figure 18: Lagrange interpolation polynomial differs extremely from a graph of function $f(x) = 1/(1 + 5x^2)$.

Model of the curve $y = 1/(1 + 5x^2)$ in basic MHR method (12) for $N = 2$ and $s = 1$:

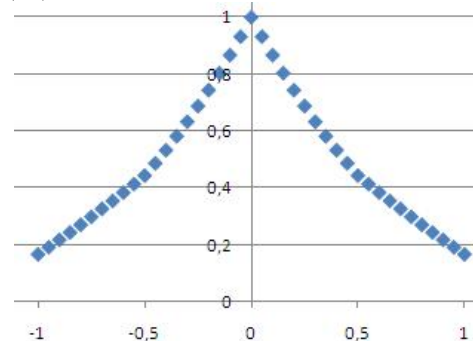


Figure 19: The curve $y = 1/(1 + 5x^2)$ modeled via MHR method for $s = 1$ and five nodes together with 36 reconstructed points.

Reconstructed curve (Fig. 19) preserves monotonicity and symmetry for $s = 1$. Comparing precise values w_i with values computed by MHR method in control points p_i , fixed for $x_i = -0.75, -0.25, 0.25, 0.75$, identical the best results appear for $s = 1$ (12):

$$|w_1 - 0.299| + |w_3 - 0.688| + |w_2 - 0.688| + |w_4 - 0.299| = 0.221 \tag{45}$$

and for $s = 0.5, 0.6, 0.7, 0.8, 0.9, 1.1, 1.2, 1.3, 1.4, 1.5, 1.7, 1.8, 1.9$. But only for $s = 1$ reconstructed curve is

symmetric- look at values in (45). So if a case $s = 1$ appears among the best results, then model ought to be built for $s = 1$. And if a case $s = 1$ does not appear among the best results, then model ought to be built for s near by 1. Here is an example of reconstructed curve for $s=1.5$ (without feature of symmetry):

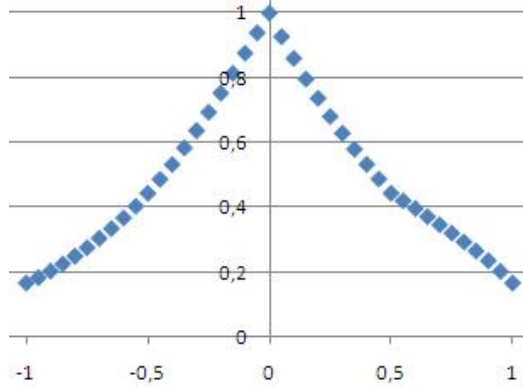


Figure 20: The curve $y = 1/(1 + 5x^2)$ modeled via MHR method for $s = 1.5$ (12) and five nodes together with 36 reconstructed points.

The best model of the curve $y = 1/(1 + 5x^2)$, built by MHR method for $s = 1$ (Fig. 19), preserves monotonicity and symmetry. Convexity of the function (Fig. 17) can make some troubles.

5 Future research directions

Future works with MHR method are connected with object recognition after geometrical transformations of curve (translations, rotations, scaling)- only nodes are transformed and new curve (for example contour of the object) for new nodes is reconstructed. Also estimation of object area in the plane, using nodes of object contour, will be possible by MHR interpolation. Object area is significant feature for object recognition. Future works are dealing with smoothing the curve, parameterization of whole curve and possibility to apply MHR method to three-dimensional curves. Also case of equidistance nodes must be studied with all details. Another future research direction is to apply MHR method in artificial intelligence and computer vision, for example identification of a specific person's face or fingerprint, optical character recognition (OCR), image restoration, content-based image retrieval and pose estimation. Future works are connected with object recognition for any set of contour points. There are several specialized tasks based on recognition to consider and it is important to use the shape of whole contour for identification and detection of persons, vehicles or other objects. Other applications of MHR method will be directed to computer graphics, modeling and image processing.

6 Conclusion

The method of Hurwitz-Radon Matrices (MHR) enables interpolation of two-dimensional curves using different coefficients: polynomial, sinusoidal, cosinusoidal, tangent, cotangent, logarithmic, exponential, arcsin,

arccos, arctan, arctg or power function [16], also inverse functions. Function for calculations is chosen individually at each curve modeling and it is treated as probability distribution function: depends on initial requirements and curve specifications. MHR method leads to curve interpolation via discrete set of fixed knots. So MHR makes possible the combination of two important problems: interpolation and modeling. Main features of MHR method are:

- the smaller distance between knots the better;
- calculations for coordinate x close to zero and near by extremum require more attention;
- MHR interpolation of the function is more precise than linear interpolation;
- minimum two interpolation knots for calculations without matrices when $N=1$, but MHR is not a linear interpolation;
- interpolation of L points is connected with the computational cost of rank $O(L)$;
- MHR is well-conditioned method (orthogonal matrices)[17];
- coefficient is crucial in the process of curve probabilistic interpolation and it is computed individually for a single curve;
- MHR 2D data extrapolation is possible for $\alpha < 0$ or $\alpha > 1$.

Future works are going to: choice and features of coefficient, implementation of MHR in object recognition [18], shape geometry, contour modeling and parameterization [19].

References

- Collins II, G.W.: Fundamental Numerical Methods and Data Analysis. Case Western Reserve University (2003)
- Chapra, S.C.: Applied Numerical Methods. McGraw-Hill (2012)
- Ralston, A., Rabinowitz, P.: A First Course in Numerical Analysis – Second Edition. Dover Publications, New York (2001)
- Zhang, D., Lu, G.: Review of Shape Representation and Description Techniques. Pattern Recognition 1(37), 1-19 (2004)
- Schumaker, L.L.: Spline Functions: Basic Theory. Cambridge Mathematical Library (2007)
- Dahlquist, G., Björck, A.: Numerical Methods. Prentice Hall, New York (1974)
- Eckmann, B.: Topology, Algebra, Analysis-Relations and Missing Links. Notices of the American Mathematical Society 5(46), 520-527 (1999)
- Citko, W., Jakóbczak, D., Sieko, W.: On Hurwitz - Radon Matrices Based Signal Processing. Workshop Signal Processing at Poznan University of Technology (2005)
- Tarokh, V., Jafarkhani, H., Calderbank, R.: Space-Time Block Codes from Orthogonal Designs. IEEE Transactions on Information Theory 5(45), 1456-1467 (1999)

- [10] Sieko, W., Citko, W., Wilamowski, B.: Hamiltonian Neural Nets as a Universal Signal Processor. 28th Annual Conference of the IEEE Industrial Electronics Society IECON (2002)
- [11] Sieko, W., Citko, W.: Hamiltonian Neural Net Based Signal Processing. The International Conference on Signal and Electronic System ICSES (2002)
- [12] Jakóbczak, D.: 2D and 3D Image Modeling Using Hurwitz-Radon Matrices. Polish Journal of Environmental Studies 4A(16), 104-107 (2007)
- [13] Jakóbczak, D.: Shape Representation and Shape Coefficients via Method of Hurwitz-Radon Matrices. Lecture Notes in Computer Science 6374 (Computer Vision and Graphics: Proc. ICCVG 2010, Part I), Springer-Verlag Berlin Heidelberg, 411-419 (2010)
- [14] Jakóbczak, D.: Curve Interpolation Using Hurwitz-Radon Matrices. Polish Journal of Environmental Studies 3B(18), 126-130 (2009)
- [15] Jakóbczak, D.: Application of Hurwitz-Radon Matrices in Shape Representation. In: Banaszak, Z., Wiśniewski, A. (eds.) Applied Computer Science: Modelling of Production Processes 1(6), pp. 63-74. Lublin University of Technology Press, Lublin (2010)
- [16] Jakóbczak, D.: Object Modeling Using Method of Hurwitz-Radon Matrices of Rank k . In: Wolski, W., Borawski, M. (eds.) Computer Graphics: Selected Issues, pp. 79-90. University of Szczecin Press, Szczecin (2010)
- [17] Jakóbczak, D.: Implementation of Hurwitz-Radon Matrices in Shape Representation. In: Chora, R.S. (ed.) Advances in Intelligent and Soft Computing 84, Image Processing and Communications: Challenges 2, pp. 39-50. Springer-Verlag, Berlin Heidelberg (2010)
- [18] 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, pp. 87-107. IGI Global, Hershey PA, USA (2011)
- [19] Jakóbczak, D.: Curve Parameterization and Curvature via Method of Hurwitz-Radon Matrices. Image Processing & Communications- An International Journal 1-2(16),49-56,(2011)