NUMERICAL DIFFERENTIATION VIA THE INTERPOLATION METHOD OF HURWITZ-RADON MATRICES

Dariusz Jakobczak, PhD
Department of Computer Science and Management,
Technical University of Koszalin , Poland

Abstract

Mathematics and computer science need suitable method for numerical calculation of derivative. Classical methods, based on polynomial interpolation, have some negative features: they are useless to interpolate the function that fails to be differentiable at one point or differs from the shape of polynomial considerably, also the Runge's phenomenon cannot be forgotten. To deal with numerical interpolation and differentiation dedicated methods should be constructed. One of them, called by author the method of Hurwitz-Radon Matrices (MHR), can be used in reconstruction and interpolation of curves in the plane. This novel method is based on a family of Hurwitz-Radon (HR) matrices. The matrices are skew-symmetric and possess columns composed of orthogonal vectors. The operator of Hurwitz-Radon (OHR), built from that matrices, is described. It is shown how to create the orthogonal and discrete OHR and how to use it in a process of function interpolation and numerical differentiation. Created from the family of N-1 HR matrices and completed with the identical matrix, system of matrices is orthogonal only for dimensions N = 2, 4 or 8. Orthogonality of columns and rows is very significant for stability and high precision of calculations. MHR method is interpolating the function point by point without using any formula of function. Main features of MHR method are: accuracy of curve reconstruction depending on number of nodes and method of choosing nodes, interpolation of L points of the curve is connected with the computational cost of rank $O(L)$, MHR interpolation is not a linear interpolation.

Keywords: Curve interpolation, numerical differentiation, Hurwitz-Radon matrices, MHR method

Introduction

Many applications of numerical methods don’t use the formula of function, but only finite set of the points (nodes). The following question is important in mathematics and computer sciences: is it possible to find a method of function interpolation and numerical differentiation without building the interpolation polynomials or other functions? This paper aims at giving the positive answer to the question. Current methods for numerical calculation of derivatives are mainly based on classical polynomial interpolation: Newton, Lagrange or Hermite polynomials and spline curves which are piecewise polynomials (Dahlquist et al. 1974; Jankowska et al. 1981). Classical methods are useless to interpolate the function that fails to be differentiable at one point, for example 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 impossible. Also when the graph of interpolated function differs from the shape of polynomial considerably, for example $f(x)=1/x$, interpolation is very hard because of existing local extrema and the roots of polynomial. We cannot forget about the Runge’s phenomenon: when nodes are equidistance then high-order polynomial oscillates toward the end of the interval, for example close to -1 and 1 with function $f(x)=1/(1+25x^2)$ (Ralston 1965).

This paper deals with the problem of interpolation (Kozera 2004; Jakóbczak 2009) and numerical differentiation without computing the polynomial or any fixed function. Coordinates of the nodes are used to build the orthogonal Hurwitz-Radon matrix operators (OHR) and a linear (convex) combinations of OHR operators lead to calculation of curve points. Main idea of MHR method is that the curve is interpolated point by point and computing the unknown coordinates of the points. The only significant factors in MHR method are choosing the interpolation nodes and fixing the dimension of HR matrices (N = 2, 4 or 8). Other characteristic features of function, such as shape or similarity to polynomials, derivative or Runge’s phenomenon, are not important in the process of MHR interpolation. The curve is parameterized for value $\alpha \in [0;1]$ in the range of two successive interpolation nodes.

In this paper computational algorithm is considered and then we have to talk about time. Complexity of calculations for one unknown point in Lagrange or Newton interpolation based on n nodes is connected with the computational cost of rank $O(n^2)$. 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 very important feature of MHR method.

The interpolation method of Hurwitz-Radon Matrices (MHR)

Adolf Hurwitz (1859-1919) and Johann Radon (1887-1956) published the papers about specific class of matrices in 1923. Matrices A_i**,** $i = 1,2\ldots m$ satisfying

$$A_jA_k+A_kA_j = 0, \quad A_j^2 = -I \quad \text{for} \quad j \neq k; \quad j, k = 1,2...m \quad (1)$$

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 reverse matrix $A_i^{-1} = - A_i$. Only for dimension N = 2, 4 or 8 the family of Hurwitz-Radon matrices consists of N-1 matrices (Citko et al. 2005).

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:

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

For N = 8 we have seven matrices with elements 0, ±1 (Sieńko et al. 2004).

Let’s assume there is given a finite set of points of the function, called further nodes $(x_i,y_i)\in\boldsymbol{R}^2$ such as:

1. nodes are settled at key points (for example local extrema: maximum or minimum) and at least one point between two successive key points;
2. there are five nodes or more.

Assume that the nodes belong to a curve in the plane. How the whole curve could be reconstructed using this discrete set of nodes? Proposed method (Jakóbczak 2007; Jakóbczak et al. 2007) is based on local and orthogonal matrix operators. Values of nodes’ coordinates (x_i,y_i) are connected with HR matrices (Eckmann 1999) build on N dimensional vector space. It is important that HR matrices are skew-symmetric and only for dimension N = 2, 4 or 8 columns and rows of HR matrices are orthogonal (Lang 1970).

If the function is described by the set of nodes $\{(x_i,y_i), i = 1,2,\ldots,n\}$ then HR matrices combined with identity matrix are used to build an orthogonal Hurwitz-Radon Operator (OHR). For nodes (x_1,y_1), (x_2,y_2) OHR of dimension N = 2 is constructed:

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

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

$$M = \frac{1}{x_1^{\ 2} + x_2^{\ 2} + x_3^{\ 2} + x_4^{\ 2}} \begin{bmatrix} u_0 & u_1 & u_2 & u_3 \\ -u_1 & u_0 & -u_3 & u_2 \\ -u_2 & u_3 & u_0 & -u_1 \\ -u_3 & -u_2 & u_1 & u_0 \end{bmatrix} \qquad (3)$$

where

$u_0 = x_1 y_1 + x_2 y_2 + x_3 y_3 + x_4 y_4$, $u_1 = -x_1 y_2 + x_2 y_1 + x_3 y_4 - x_4 y_3$,

$u_2 = -x_1 y_3 - x_2 y_4 + x_3 y_1 + x_4 y_2$, $u_3 = -x_1 y_4 + x_2 y_3 - x_3 y_2 + x_4 y_1$.

For nodes (x_1,y_1), (x_2,y_2), …, (x_8,y_8) OHR M of dimension N = 8 is built (Jakóbczak 2007) similarly as (3):

$$M = \frac{1}{\sum_{i=1}^{8} x_i^{\ 2}} \begin{bmatrix} u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \\ -u_1 & u_0 & u_3 & -u_2 & u_5 & -u_4 & -u_7 & u_6 \\ -u_2 & -u_3 & u_0 & u_1 & u_6 & u_7 & -u_4 & -u_5 \\ -u_3 & u_2 & -u_1 & u_0 & u_7 & -u_6 & u_5 & -u_4 \\ -u_4 & -u_5 & -u_6 & -u_7 & u_0 & u_1 & u_2 & u_3 \\ -u_5 & u_4 & -u_7 & u_6 & -u_1 & u_0 & -u_3 & u_2 \\ -u_6 & u_7 & u_4 & -u_5 & -u_2 & u_3 & u_0 & -u_1 \\ -u_7 & -u_6 & u_5 & u_4 & -u_3 & -u_2 & u_1 & u_0 \end{bmatrix} \qquad (4)$$

where

$$u = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ -y_2 & y_1 & -y_4 & y_3 & -y_6 & y_5 & y_8 & -y_7 \\ -y_3 & y_4 & y_1 & -y_2 & -y_7 & -y_8 & y_5 & y_6 \\ -y_4 & -y_3 & y_2 & y_1 & -y_8 & y_7 & -y_6 & y_5 \\ -y_5 & y_6 & y_7 & y_8 & y_1 & -y_2 & -y_3 & -y_4 \\ -y_6 & -y_5 & y_8 & -y_7 & y_2 & y_1 & y_4 & -y_3 \\ -y_7 & -y_8 & -y_5 & y_6 & y_3 & -y_4 & y_1 & y_2 \\ -y_8 & y_7 & -y_6 & -y_5 & y_4 & y_3 & -y_2 & y_1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}. \qquad (5)$$

The components of the vector $u = (u_0, u_1,\ldots, u_7)^T$, appearing in the matrix M (4), are defined by (5) in the similar way to (2)-(3) but in terms of the coordinates of the above 8 nodes. Note that OHR operators (2)-(4) satisfy the condition of interpolation

$$M\cdot\mathbf{x} = \mathbf{y} \qquad (6)$$

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

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

$$\alpha = \frac{b-c}{b-a} \in [0;1]. \qquad (7)$$

Average OHR operator M_2 of dimension N = 2, 4 or 8 is constructed as follows:

$$M_2 = \alpha \cdot M_0 + (1-\alpha) \cdot M_1 \qquad (8)$$

with the operator M_0 built (2)-(4) by "odd" nodes (x_1=a,y_1), (x_3,y_3), …, ($x_{2N\text{-}1}$,$y_{2N\text{-}1}$) and M_1 built (2)-(4) by "even" nodes (x_2=b,y_2), (x_4,y_4), …, (x_{2N},y_{2N}). Notice that 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

$$c_i = \alpha \cdot x_{2i\text{-}1} + (1\text{-}\alpha) \cdot x_{2i} \quad \text{for} \quad i = 1,2,\ldots,N \qquad (9)$$

as $C = [c_1, c_2, \ldots, c_N]^{\mathrm{T}}$. The required formula is adequate to (6):

$$Y(C) = M_2 \cdot C \qquad (10)$$

in which components of vector Y(C) give the second coordinates of the points (x,y) corresponding to the first coordinates, given in terms of components (9) of the vector C.

After computing (7)-(10) 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

$$c_i = \alpha \cdot x_{2i} + (1\text{-}\alpha) \cdot x_{2i+1} \quad , \quad i = 1,2,\ldots,N. \qquad (11)$$

There is no need to build the OHR for nodes (x_2=a,y_2), (x_4,y_4), …, (x_{2N},y_{2N}) because we just find M_1. This operator will play as role as M_0 in (8). New M_1 must be computed for nodes (x_3=b,y_3), …, ($x_{2N\text{-}1}$,$y_{2N\text{-}1}$), (x_{2N+1},y_{2N+1}). As we see the minimum number of interpolation nodes is n = 2N+1 = 5, 9 or 17 using OHR operators of dimension N = 2, 4 or 8 respectively. If there is more nodes than 2N+1, the same calculations (7)-(11) 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.

<u>Algorithm 1</u>: let j = 1.

Input: Set of interpolation nodes {(x_i,y_i), i = 1,2,…,n; n = 5, 9 *or* 17}.

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. Build M_0 *for nodes* (x_1=a,y_1), (x_3,y_3), …, ($x_{2N\text{-}1}$,$y_{2N\text{-}1}$) *and* M_1 *for nodes* (x_2=b,y_2),
(x_4,y_4), …, (x_{2N},y_{2N}) *from (2)-(4).*

Step 3. Determine the number of points to be reconstructed $K_j > 0$ *between two successive nodes (for example 9 or 99), let* k = 1.

Step 4. Compute $\alpha \in [0;1]$ *from (7) for* $c_1 = c = \alpha \cdot a + (1\text{-}\alpha) \cdot b$.

Step 5. Build M_2 *from (8).*

Step 6. Compute vector $C = [c_1, c_2,\ldots, c_N]^T$ *from* (9).
Step 7. Compute unknown coordinates Y(C) from (10).
Step 8. If $k < K_j$ *, set* $k = k+1$ *and go to Step 4. Otherwise if* $j = 1$*, set* $M_0 = M_1$,
$a = x_2$, $b = x_3$, *build new* M_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.*

The number of reconstructed points in Algorithm 1 is $K = N(K_1+K_2)$. If there is 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 by author the method of Hurwitz-Radon Matrices (MHR).

MHR application and graphical examples

In this section we consider the number of multiplications and divisions for 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). OHR operators of dimension $N = 2$ are built (2) as follows:

$$M_0 = \frac{1}{x_1^2 + x_3^2}\begin{bmatrix} x_1y_1 + x_3y_3 & x_3y_1 - x_1y_3 \\ x_1y_3 - x_3y_1 & x_1y_1 + x_3y_3 \end{bmatrix},$$

$$M_1 = \frac{1}{x_2^2 + x_4^2}\begin{bmatrix} x_2y_2 + x_4y_4 & x_4y_2 - x_2y_4 \\ x_2y_4 - x_4y_2 & x_2y_2 + x_4y_4 \end{bmatrix}.$$

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

$$c_1 = \alpha\cdot x_1+\beta\cdot x_2 \qquad \text{for} \qquad 0 \le \beta = 1 - \alpha \le 1. \qquad (12)$$

Compute second coordinate of reconstructed point $y(c_1)$ for $Y(C) = [y(c_1), y(c_2)]^T$ from (10):

$$\begin{bmatrix} y(c_1) \\ y(c_2) \end{bmatrix} = (\alpha\cdot M_0 + \beta\cdot M_1)\cdot\begin{bmatrix} \alpha\cdot x_1 + \beta\cdot x_2 \\ \alpha\cdot x_3 + \beta\cdot x_4 \end{bmatrix}. \qquad (13)$$

After calculation (13):

$$y(c_1) = \alpha^2\cdot y_1 + \beta^2\cdot y_2 + \frac{\alpha\cdot\beta}{x_1^2 + x_3^2}(x_1x_2y_1 + x_2x_3y_3 + x_3x_4y_1 - x_1x_4y_3) +$$
$$+ \frac{\alpha\cdot\beta}{x_2^2 + x_4^2}(x_1x_2y_2 + x_1x_4y_4 + x_3x_4y_2 - x_2x_3y_4). \qquad (14)$$

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 $P(\alpha)$ for (12), (14) and $\alpha \in [0;1]$.

If nodes (x_i,y_i) are equidistance in coordinate x_i, then parameterization of unknown coordinate (14) 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 (13) and (14) are done for c_1 (12):

$$y(c_1) = \alpha y_1 + \beta y_2 + \alpha\beta s \qquad (15)$$

and

$$s = h\left(\frac{2ay_1 + hy_1 + hy_3}{4a^2 + 4ah + 2h^2} - \frac{2ay_2 + 2hy_2 + hy_4}{4a^2 + 8ah + 5h^2}\right). \qquad (16)$$

As we can see in (15) and (16), MHR interpolation is not a linear interpolation. It is possible to estimate the interpolation error of MHR method (Algorithm 1) for the class of linear function *f*:

$$\left| f(c_1) - y(c_1) \right| = \left| \alpha y_1 + \beta y_2 - y(c_1) \right| = \alpha\beta |s|. \qquad (17)$$

Notice that estimation (17) has the biggest value ¼$|s|$ for $\beta = \alpha = ½$, when c_1 is situated in the middle between x_1 and x_2.

The goal of this paper is not a reconstruction of single point, like for example (14) and (15), 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 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 costs appear 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: $\{(x_1,y_1),\ldots,(x_5,y_5)\},\{(x_5,y_5),\ldots,(x_9,y_9)\},\{(x_9,y_9), \ldots, (x_{13},y_{13})\}\ldots$ If the last node (x_n,y_n) is indexed $n \neq 4i+1$ then we have to use last five nodes $\{(x_{n-4},y_{n-4}), \ldots, (x_n,y_n)\}$ in Algorithm 1.

Function $f(x) = 1/x$ is an example when the graph of interpolated function differs from the shape of polynomials considerably. Then classical interpolation is very hard because of existing local extrema and the roots of polynomial (Fig.2). Here is the application of Algorithm 1 for this function and five nodes.

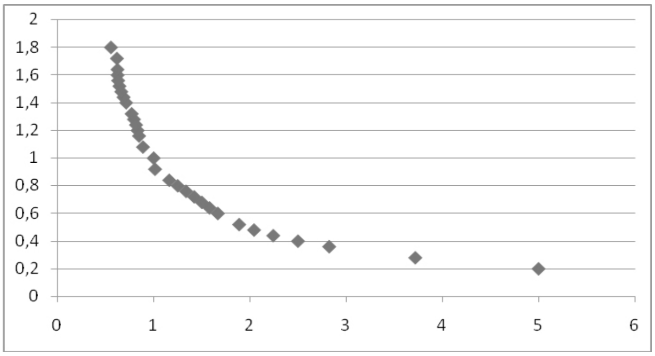

Figure 1- Twenty six interpolated points of function *f*(*x*) = 1/*x* using MHR method (Algorithm 1) together with 5 nodes: (5;0.2), (5/3;0.6), (1;1), (5/7;1.4), (5/9;1.8)

Figure 1 contains not too many (twenty six) interpolated points (x_i,y_i) and minimal number of nodes (five). As the example, numerical calculations of derivative $f'(x) = -1/x^2$ for $x_i = 1.421$ look as follows:

1. precise value $f'(1.421) = -0.495$;
2. two-point estimation $f'(x_i) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = -0.5$;
3. three-point estimation $f'(x_i) = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} = -0.488$.

Second example - numerical calculations of derivative for $x_i = 3.719$ (Fig.1):

1. precise value $f'(3.719) = -0.0723$;
2. two-point estimation $f'(x_i) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = -0.06245$;
3. three-point estimation $f'(x_i) = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} = -0.07353$.

Greater number of nodes and interpolated points gives us more accurate value of differentiation. Lagrange interpolation polynomial for function *f*(*x*) = 1/*x* and nodes (5;0.2), (5/3;0.6), (1;1), (5/7;1.4), (5/9;1.8) has one local minimum and two roots.

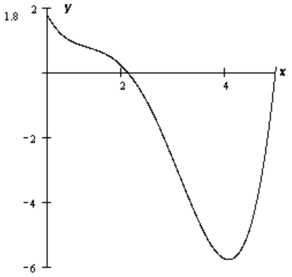

Figure 2 – Lagrange interpolation polynomial for nodes (5;0.2), (5/3;0.6), (1;1), (5/7;1.4), (5/9;1.8) differs extremely from the shape of function *f*(*x*) = 1/*x*

Other examples of MHR interpolation and numerical differentiation:

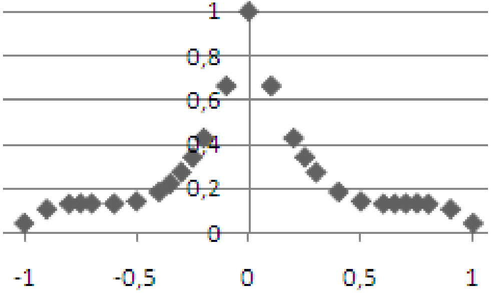

Figure 3 - Twenty two interpolated points of functions *f*(*x*) = 1/(1+25*x*^{2}) using MHR method with 5 nodes for x_i = -1; -0.5; 0; 0.5 and 1: no Runge's phenomenon

Figure 3 contains minimal number of nodes (five) and only twenty two interpolated points. As the example, numerical calculations of derivative for x_i = 0.0 look as follows:

1. precise value *f'*(0.0) = 0.0;
2. three-point estimation $f'(x_i)=\dfrac{y_{i+1}-y_{i-1}}{x_{i+1}-x_{i-1}}=0.0$.

Second example- numerical calculations of derivative for x_i = -0.35 (Fig.3):

1. precise value *f'*(-0.35) = 1.06;
2. two-point estimation $f'(x_i)=\dfrac{y_{i+1}-y_i}{x_{i+1}-x_i}=1.04$;
3. three-point estimation $f'(x_i)=\dfrac{y_{i+1}-y_{i-1}}{x_{i+1}-x_{i-1}}=0.9$.

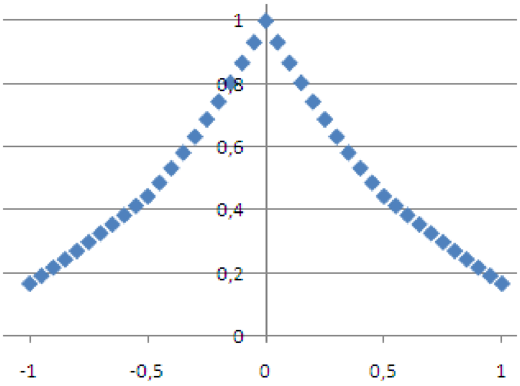

Figure 4 - Thirty six interpolated points of functions *f*(*x*) = 1/(1+5*x*^{2}) using MHR method with 5 nodes for x_i = -1; -0.5; 0; 0.5 and 1: no Runge's phenomenon

Figure 4 contains minimal number of nodes (five) and not too many interpolated points (thirty six). As the example, numerical calculations of derivative for x_i = 0.0 look as follows:

1. precise value *f'*(0.0) = 0.0;
2. three-point estimation $f'(x_i)=\dfrac{y_{i+1}-y_{i-1}}{x_{i+1}-x_{i-1}}=0.0$.

Second example - numerical calculations of derivative for x_i = 0.15 (Fig.4):

1. precise value *f'*(0.15) = -1.212;
2. two-point estimation $f'(x_i)=\dfrac{y_{i+1}-y_i}{x_{i+1}-x_i}=-1.2$;
3. three-point estimation $f'(x_i)=\dfrac{y_{i+1}-y_{i-1}}{x_{i+1}-x_{i-1}}=-1.23$.

Here are the graphs of functions interpolated by MHR method with 5 nodes (Fig.5,6,8) and 9 nodes (Fig.7):

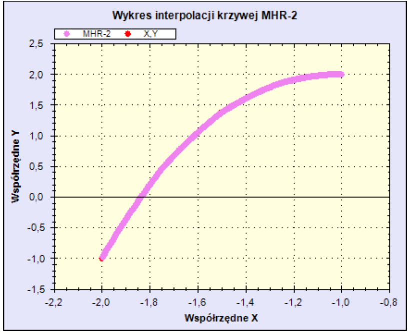

Figure 5 - Function *f*(*x*) = x^3+x^2-x+1 with 396 interpolated points using MHR method with 5 nodes: (-2;-1), (-1.75; 0.453125), (-1.5;1.375), (-1.25;1.859375) and (-1;2).

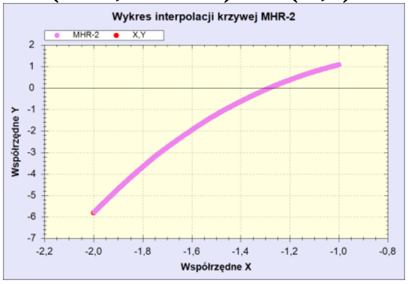

Figure 6 - Function *f*(*x*) = x^3+ln(7-*x*) with 396 interpolated points using MHR method with 5 nodes: (-2;-5.803), (-1.75;-3.190), (-1.5;-1.235), (-1.25;0.1571) and (-1;1.0794).

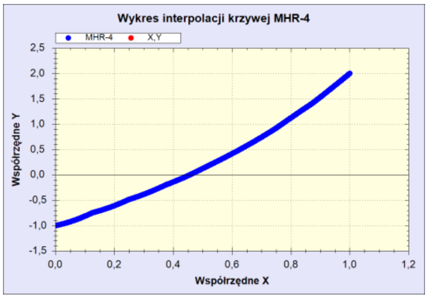

Figure 7 - Function $f(x) = x^3+2x-1$ with 792 interpolated points using MHR method with 9 nodes for x_i= 0; 0.125; 0.25; 0.375; 0.5; 0.625; 0.75; 0.875 and 1.

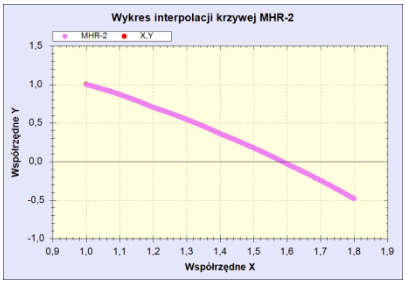

Figure 8 - Function $f(x) = 3-2^x$ with 396 interpolated points using MHR method with 5 nodes: (1;1), (1.2;0.7026), (1.4;0.361), (1.6;-0.031) and (1.8;-0.482).

After computing of K points for interpolated function (algorithm 1), it is possible to calculate the derivative via two-point or three-point estimation. Greater number of nodes and interpolated points gives us more accurate value of differentiation.

Conclusion

The method of Hurwitz-Radon Matrices (MHR - Algorithm 1) leads to curve interpolation depending on the number of nodes and location of nodes. No characteristic features of curve are important in 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. MHR method gives the possibility of curve reconstruction and then numerical calculations of derivatives for interpolated function via two-point or three-point estimation for example. The only condition is to have a set of nodes according to assumptions in Algorithm 1. Curve modeling (Jakóbczak 2010) by MHR method is connected with possibility of changing the nodes coordinates and reconstruction of new curve for new set of nodes, no matter what shape of curve or function is to be reconstructed. Main features of MHR method are:

1) accuracy of curve reconstruction depending on number of nodes and method of choosing nodes;

2) reconstruction of curve consists of *L* points is connected with the computational cost of rank *O(L*);

3) Algorithm 1 is dealing with local operators: average OHR operator M_2 (8) is built by successive 4, 8 or 16 nodes, what is connected with smaller computational costs then using all nodes.

Future works are connected with: 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; estimation of curve length (Jakóbczak 2010); possibility to apply MHR method to three-dimensional curves; object recognition (Jakóbczak 2011), shape representation (Jakóbczak 2010) and parameterization in image processing; curve extrapolation (Jakóbczak 2011).

References:

Citko W., Jakóbczak D., Sieńko W.: *On Hurwitz-Radon Matrices Based Signal Processing*, Signal Processing' 2005, Poznań, 19-24, 2005.

Dahlquist G., Bjoerck A.: *Numerical methods*, Englewood Cliffs N. J., Prentice-Hall, 1974.

Eckmann B.: *Topology, algebra, analysis- relations and missing links,* Notices of AMS, vol. 46(5), 520-527, 1999.

Jakóbczak D.: *2D and 3D Image Modeling Using Hurwitz-Radon Matrices*, Polish Journal of Environmental Studies, vol. 16 (4A), 104-107, 2007.

Jakóbczak D., Kosiński W.: *Hurwitz-Radon Operator in Monochromatic Medical Image Reconstruction,* Journal of Medical Informatics & Technologies, vol.11, 69-78, 2007.

Jankowska J., Jankowski M.*: Survey of numerical methods and algorithms (Part I),* Wydawnictwa Naukowo-Techniczne, Warsaw, 1981 [in Polish].

Lang S:. *Algebra,* Addison-Wesley Publishing Company, 1970.

Ralston A.: *A first course in numerical analysis,* McGraw-Hill Book Company, 1965.

Sieńko W., Citko W., Jakóbczak D.: *Learning and System Modeling via Hamiltonian Neural Networks*, Lecture Notes on Artificial Intelligence, Springer-Verlag: Berlin Heidelberg, vol. 3070, 266-271, 2004.

Kozera, R.: Curve Modeling via Interpolation Based on Multidimensional Reduced Data. Silesian University of Technology Press, Gliwice (2004)

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)

Jakóbczak, D.: Curve Interpolation Using Hurwitz-Radon Matrices. Polish Journal of Environmental Studies 3B(18), 126-130 (2009)

Jakóbczak, D.: Application of Hurwitz-Radon Matrices in Shape Representation. In: Banaszak, Z., Świć, A. (eds.) Applied Computer Science: Modelling of Production Processes 1(6), pp. 63-74. Lublin University of Technology Press, Lublin (2010)

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)

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, pp. 39-50. Springer-Verlag, Berlin Heidelberg (2010)

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)

Jakóbczak, D.: Data Extrapolation and Decision Making via Method of Hurwitz-Radon Matrices. Lecture Notes in Computer Science/LNAI 6922 (Computational Collective Intelligence, Technologies and Applications: Proc. ICCCI 2011, Part 1), Springer-Verlag Berlin Heidelberg, 173-182 (2011)

Jakóbczak, D.: Curve Parameterization and Curvature via Method of Hurwitz-Radon Matrices. Image Processing & Communications- An International Journal 1-2(16), 49-56 (2011)