SIMULATION OF ION BEAM TRANSPORT INSIDE ELECTROSTATIC QUADRUPOLE LENS WITH ROTATIONALLY SYMMETRIC DESIGN

Stanislav MINÁRIK

doc. Ing. Stanislav Minárik, PhD., Slovak University of Technology in Bratislava, Faculty of Materials Science and Technology in Trnava, Advanced Technologies Research Institute, Bottova 25, 917 24 Trnava, Slovakia,
e-mail: stanislav.minarik@stuba.sk

Abstract

The exact analytical solution of Laplace`s equation and electrostatic scalar potential distribution inside the quadrupole lens with rotationally symmetric design were already found in our last work (1). In present contribution, we consider the simplest form of equations of motion established in the mentioned previous work concerning the simulation of ion beam transport. The solution of differential equations system governing the ion-beam transport was found numerically by time history integration. Using the integration scheme inherent in Mathematica Solver, we calculated the time history development of ion´s position. Data for the hydrogen ion-beam trajectory was shown while initial conditions and values of constants were chosen arbitrarily. This problem can be of considerable interest for any ion-beam technology and ion-beam analysis activities.

Key words

ion beam, electrostatic quadrupole, rotationally symmetric, differential system, Mathematica Solver

1. INTRODUCTION

In our previous work, we solved Laplace`s equation analytically for electrostatic quadrupole with special kind of symmetry shown in Fig.1 and established the equations of motion for ion inside this quadrupole. These system od differential equations (SDEs) describing the ion beam dynamics involves the terms corresponding to the specific geometry of electrode pairs. In this article, we document approximate solutions of those equations of motion by numerical techniques using suitable software tools. The main goal is to find the trajectory of ion-beam passing through the quadrupole and to evaluate basiccharacteristics enabling description of ion-beam behaviour.

The solution of a differential system can be found numerically by time history integration, using various integration schemes. Several programs exist for modelling and simulation

based on SDEs. We outline the design and implementation details of a single-step method for solving SDEs describing the ion dynamics in *Mathematica Solver*. It is possible to consult references about using *Mathematica* for differential equations (2, 3). In general, the most important features of the *NDSolve* framework in the *Mathematica* are object orientation and data hiding. Several methods are used in this framework (4,5), such as symmetry reduction techniques (6,7), integrating factor method (8) or Bocharov techniques (9). Each method has its own data object that contains information which is needed for invocation. This includes coefficients, workspaces, step-size control parameters, step-size acceptance and rejection information, Jacobian matrices etc. Highly accurate solutions can be obtained by making use of arbitrary precision software arithmetic (10, 11, 12). Issues relating to appropriate formulation and efficient implementation can also be discussed, together with the strategies for automatic method, order and parameter selection (13, 14).

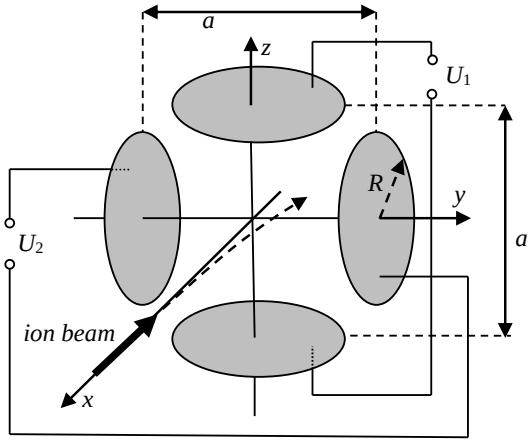

Fig. 1 *Layout of the electrostatic quadrupole system*

This paper will present the implementation of a general software tool *Mathematica* for simulation of ion beam transport through special electrostatic quadrupole (see Fig.1). The simulation is based on solution of SDEs describing the ion-beam dynamics. The implementation was made in *Mathematica Solver* in the frame of *NDSolve* command,what makes experimentation with this simulation approach readily available. The flexibility of the simulation approach will be demonstrated by one example of applications. In this example, the estimation of hydrogen ion-beam trajectory behavior is presented.

2. BRIEF DESCRIPTION OF SIMULATION PROCESS

Equations of motion for ion passing through electrostatic field in quadrupole are represented by second-order differential system. We found these governing equations in our previous work by analytical solution of Laplace`s equation. In the limit case for ion moving near the x - axis (see Fig.1), next simplified equations of motion can be written as:

$$M\frac{d^2x}{dt^2}=-qA\left(\frac{\pi}{a}\right)^2\left\{1+\left(\frac{\pi}{2a}\right)^2\left[\left(x^2+y^2\right)+\left(x^2+z^2\right)\right]\right\}x \qquad [1]$$

$$M\frac{d^2y}{dt^2}=-\frac{qU_2}{a}+qA\left(\frac{\pi}{a}\right)^2\left\{1-\left(\frac{\pi}{2a}\right)^2\left(x^2+z^2\right)\right\}y \qquad [2]$$

$$M\frac{d^2z}{dt^2}=-\frac{qU_1}{a}+qA\left(\frac{\pi}{a}\right)^2\left\{1-\left(\frac{\pi}{2a}\right)^2\left(x^2+y^2\right)\right\}z \qquad [3]$$

where: $A\approx\left(\frac{a}{\pi^2R}\right)\sqrt{U_1^2+U_2^2}$. [4]

M is mass of ion transporting through quadrupole, q is charge of the ion, U_1 and U_2 are applied voltages, R is electrodes radius and a is the spacing between the electrodes (see Fig.1). The task is to find position of the ion defined by coordinates [$x(t)$, $y(t)$, $z(t)$] as function af time t. It is difficult to find the analytical solution of the SDEs [1] –[3]. In

addition, it is not clear whether this SDEs have analytical solutions. However such equations can be solved using numerical techniques. As was shown the *NDSolve* command in *Mathematica* can be used to solve differential equations numerically. The problem [1-3] can be solved in *Mathematica Solver* using the following command:

dif=NDSolve[{x''[t] == -p(1 + b*((x[t]^2 + y[t]^2) + (x[t]^2 + z[t]^2)))* x[t], y''[t] == F + p*(1 - b*(x[t]^2 + z[t]^2))*y[t], z''[t] == G + p*(1 - b*(x[t]^2 + y[t]^2))*z[t], x[0] == -R, y[0] == c*Sin[2*(Pi/360)*s], z[0] == c*Cos[2*(Pi/360)*s], x'[0] == w, y'[0] == z'[0] == 0}, {x, y, z}, {t, zz, kk}, MaxStepSize* $\rightarrow$ 0.0000000005, *MaxSteps* $\rightarrow \infty$ *];*

where *p*, *b*, *F* and *G* are corresponding constants, while *zz*, *kk*, *w*, *s* and *c* are parameters. Parameter *c* represents radius of the ion-beam and *w* is initial velocity of ions. Size of interval $\langle zz, kk \rangle$ is proportional to time during which the ion is transporting through the quadrupole. So, parameters *zz* and *kk* has to be deduced intuitively. At first sight, the exact syntax of the above-mentioned command is perhaps a little confusing, but what single symbols in command-line signify is intuitively quite clear. A number of numerical methods referred in the literature rely on the fact that a closed-form solution of SDEs is available. But that is obviously quite limiting. In the *NDSolve* command showed above, it is possible to get very accurate approximations of SDEs solution using arbitrary-precision adaptive step size. These adaptive order methods are based on extrapolation procedure.
The *NDSolve* is defined as a block which allows us to combine *Mathematica* commands. These commands can be all evaluated together. *Mathematica* returns the numerical solution of SDEs and when adding boundary conditions into the differential equation solver function, the solver will return a function that is only valid as a solution of the boundary value problem on a particular time interval. Unfourtunately, the equation solver has no mechanism for reporting restriction of validity of the solution and some "non-physical" results can occur (e.g. in case of certain discontinuity properties of higher order derivatives). In many of applications where boundary value problems arise, there may be undetermined parameters in the problem itself and selection of these parameters is a part of the desired solution. Generally, by introducing the parameters as dependent variables, the problem can often be written as a boundary value problem in a standard form.
The result is an interpolating function "*InterpolatingFunction[{{x,y,z}},<>][t]*" which has been fitted to the numerical data. The function is only valid in the specified range of the independant variable *t*. We can use the solution "interpolating function" as a regular function but we can also define a function that is equal to the polynomial too.
Then in *Mathematica Solver* the ion-beam envelope can be plotted by next command:
ParametricPlot3D[Evaluate[{{x[t], y[t], z[t]} /. dif, {t, zz, kk}], PlotStyle $\rightarrow$ *Red]*
We carried out solution of the SDEs (1-3) by numerical algorithms mentioned above. Arbitrarily values of constants and parameters were chosen and trajectory of ion-beam was displayed. Output of numerical data allows to evaluate character of ion-beam behaviour (output beam energy, deflection of the beam, etc.). Ordinary two-core PC with Intel Celeron CPU, 1005M, 1.90GHz was used for the simulation.

3. RESULTS OF SIMULATION AND ITS POSSIBLE APPLICATION

First we investigated shape of the ion-beam trajectory for the beam entering to the middle of quadrupole along the *x*-axis (as is shown in Fig.1). $^{1}H^{+}$ ions (protons) with the mass $M = 1{,}6726.10^{-27}$ kg and charge $q = 1{,}602.10^{-19}$ C were considered. Radius of quadrupole electrodes was chosen $R = 0{,}5$ m and the spacing between the electrodes $a = 1{,}2$ m.

Ion-beam behaviour for various input parameters as a ion beam energy and applied voltages U_1 and U_2 was investigated for the simulation. Input beam radius was 1 mm. Some of results are summarized in Table 1.

EXAMPLE OF INPUT PARAMETERS FOR THE SIMULATION OF ION BEAM TRANSPORT THROUGH THE QUADRUPOLE AND OUTPUT ION-BEM PARAMETERS (OUTPUT BEAM ENERGY AND DEFLECTION OF THE BEAM) RESULTING FROM THE SIMULATION

Table 1

Voltage U_1 [kV]	Voltage U_2 [kV]	Initial beam energy [keV]	Output beam energy [keV]	Deflection of the beam [°]
10	8	30	31.9332	14.4098
30	20	100	104.7	12.2083
300	260	1200	1245.17	11.2958
310	280	1600	1645	8.79879
80	50	500	506.659	6.45175
11	8	50	51.4507	9.20414
15	10	80	81.7234	7.60959
50	30	120	130.622	16.0993
80	50	150	171.172	20.6499
90	60	200	221.589	17.8325

Here is the brief data output listing obtained from the simulation of the ion-beam trajectory for the input parameters shown in the firs row in Table 1, i.e. applied voltages U_1 = 10 kV, U_2 = 8 kV and initial energy of ions in the beam 30 keV:

{{*x*,*y*,*z*]} ={[-0.5,0,0] , [-0.479315,-0.0000425032,-0.0000340026] , [-0.458501,-0.000170017,-0.000136013] , [-0.437568,-0.000382554, -0.000306043] , [-0.416525,-0.000680136,-0.000544109] , [-0.395382,-0.0010628,-0.000850238] , [-0.374147,-0.00153058, -0.00122447] , [-0.352828,-0.00208355,-0.00166684] , [-0.331433,-0.00272178,-0.00217742] , [-0.309969,-0.00344534,-0.00275627] , [-0.288443,-0.00425435,-0.00340348] , [-0.26686,-0.00514891,-0.00411913] , [-0.245228,-0.00612917,-0.00490334] , [-0.22355, -0.00719527,-0.00575621] , [-0.201833,-0.00834736,-0.00667789] , [-0.180082,-0.00958564,-0.00766851] , [-0.1583,-0.0109103, -0.00872824] , [-0.136492,-0.0123215,-0.00985723] , [-0.114662,-0.0138196,-0.0110557] , [-0.0928134,-0.0154047,-0.0123237] , [-0.0709507,-0.0170771,-0.0136617] , [-0.0490772,-0.018837,-0.0150696] , [-0.0271961,-0.0206849,-0.0165479] , [-0.00531096, -0.0226208,-0.0180966] , [0.016575,-0.0246452,-0.0197161] , [0.0384585,-0.0267582,-0.0214066] , [0.0603361,-0.0289604, -0.0231683] , [0.0822044,-0.0312518,-0.0250014] , [0.10406,-0.033633,-0.0269064] , [0.125899,-0.0361041,-0.0288833] , [0.147718, -0.0386656,-0.0309324] , [0.169513,-0.0413177,-0.033054] , [0.19128,-0.0440607,-0.0352484] , [0.213014,-0.0468949,-0.0375158] , [0.23471,-0.0498207,-0.0398564] , [0.256364,-0.0528383,-0.0422704] , [0.27797,-0.0559479,-0.044758] , [0.299523,-0.0591499, -0.0473195] , [0.321016,-0.0624444,-0.049955] , [0.342443,-0.0658316,-0.0526647] , [0.363796,-0.0693118,-0.0554487] , [0.385069, -0.072885,-0.0583071] , [0.406253,-0.0765513,-0.06124] , [0.42734,-0.0803109,-0.0642475] , [0.448321,-0.0841638,-0.0673295] , [0.469186,-0.0881099,-0.0704861]}.

Coordinates are given in meters and this trajectory is shownn in Fig. 2. Next figures show trajectories of the ion-beam with input parameters referred in the second (Fig. 3) and third (Fig. 4) rows in Table 1.

It should be noted that the resulting coordinates mentioned above represent the position of ion at different times. Next, these coordinates should be analysed.

We can further check whether other solution schemes, possibly with finer integration steps, would reveal different solutions to the problem.

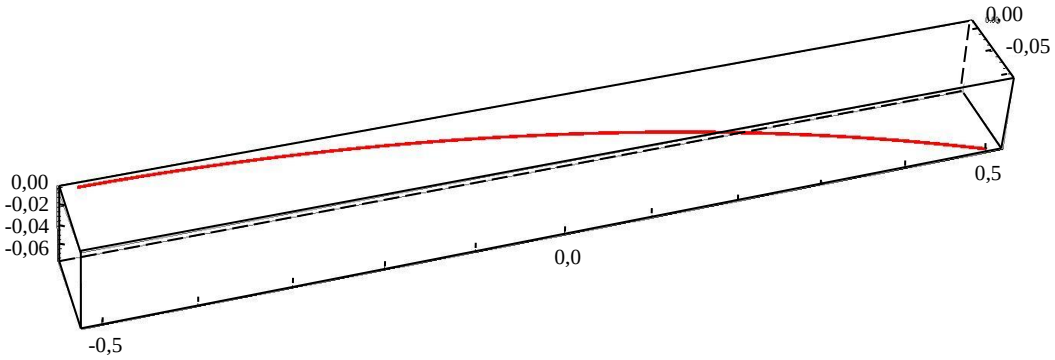

Fig. 2 30keV *Hydrogen ion beam trajectory inside the proposed electrostatic quadrupole at applied voltages* 10 kV *(horizontal) and* 8 kV*(vertical). Output energy of ions is* 31.9332 keV, *deflection angle* is 14.4098º. *Coordinates are given in meters.*

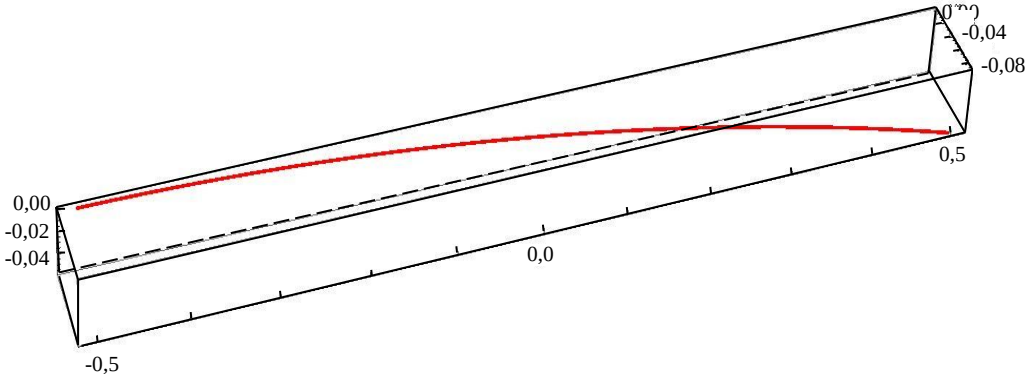

Fig. 3 *Calculated* 100keV *Hydrogen ion beam trajectory inside the proposed electrostatic quadrupole at applied voltages* 30 kV *(horizontal) and* 20 kV*(vertical). Output energy of ions is* 104.7 keV, *deflection angle* is 12.2083º. *Coordinates are given in meters.*

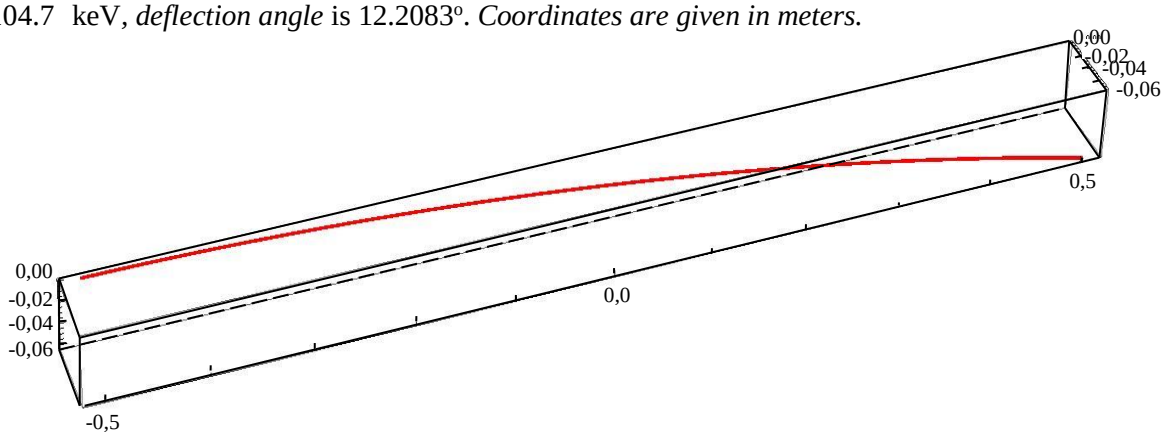

Fig. 4 *Calculated* 1,2MeV *Hydrogen ion beam trajectory inside the proposed electrostatic quadrupole at applied voltages* 300 kV *(horizontal) and* 260 kV*(vertical). Output energy of ions is* 1.24517 MeV, *deflection angle* is 11.2958º. *Coordinates are given in meters.*

Next, we investigated the ion-beam behaviour when passing through the quadrupole under different conditions. Results of simulation showed that the deflection angle of the ion-beam decreases with the increasing of the initial energy of ions at constant values of applied voltages.

The character of this decrease was calculated and two obtained results are shown in Fig. 5. It is also possible to simulate changes of the cross-sectional shape of ion-beam passing through the quadrupole. Result of such simulation is shown in Fig. 6. Ion-beam transport

calculations for the ion with radius 1 mm was performed and chnges of its shape was modeled.

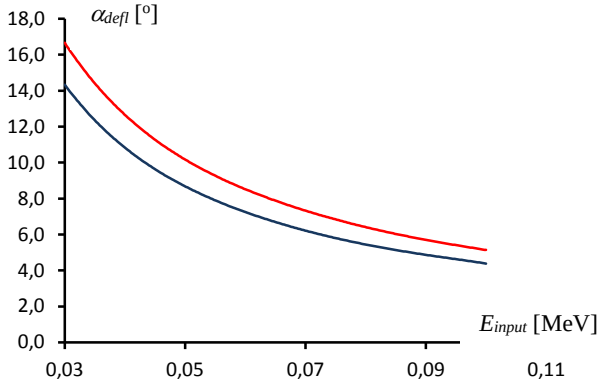

Fig. 5 *Calculated deflection angle as a function of initial energy of* $^1H^+$ *ions for next applied voltages:* ~~U_1~~ = 12 kV, U_2 = 9 kV, ~~U_1~~ = 10 kV, U_2 = 8 kV

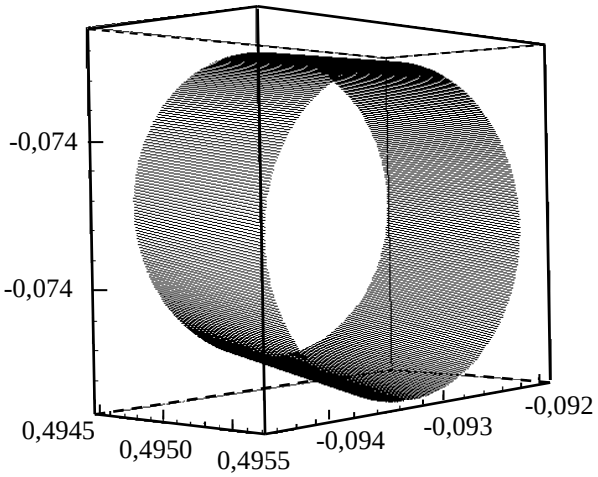

Fig. 6 *Calculated 3D envelope of the 30 keV* $^1H^+$ *ion- beam at the output of quadrupole for applied voltages* U_1 *=10 kV and* U_2 *= 8 kV*

4. CONCLUSION

In this paper, we evaluate numerical solutions to the simplified second order differential system governing the dynamics of ion in electrostatic quadrupole consisting of electrodes with rortational symmetry. We used the *Mahematica Solver* to obtain ion beam trajectories inside this quadrupole. Some spurious results that are non-physical can be expected when the input parameters are outside the area defined by assumptions of exact analytical solutions. Therefore, we had to introduce a matching technique to ensure that selected integration constants ensure corresponding these assumptions. *Mathematica Solver* was proven very useful for identification of solutions.

Nevertheless, we express a warning that the physics of the problem must be never forgotten when looking for the solutions to problems, that describe any event. Numerical results showed that deflection of ion-beam in quadrupole is negligible due to the resultant electrostatic force and it is very challenging for the beam transport through the beam-lines.

In our research, we focused mainly on the possibilities of numerical simulation of ion-beam trajectory using general available software tool. The present solution has encouraged us to continue the work and find possibly the ion-beam trajectories at a greater distance from the *x* axis, i.e. generally in the areas which do not meet assumptions for simplified differential system. In such case, unsimplified equations of motion must be solved. We concluded, that systematic analysis of ion beam transport by *Mathematica Solver* software is possible.

Acknowledgement

This research was funded by the ERDF - Research and Development Operational Programme under the project "University Scientific Park Campus MTF STU - CAMBO" ITMS: 26220220179.

References:

1. S. MINARIK at all., *Ion Beam Dynamics Inside Electrostatic Quadrupole Lens with Rotationally Symmetric Design* (in press).
2. M. SOFRONIOU and R. KNAPP. 2008. *Advanced Numerical Differential Equation Solving in Mathematica*, Champaign, IL: Wolfram Research, Inc.
3. V. G. GANZHA and E. V. VOROZHTSOV. 1996. *Numerical Solutions for Partial Differential Equations—Problem Solving* Using Mathematica, Boca Raton, FL: CRC Press, Inc.
4. BEREZIN I. S. and N. P. ZHIDKOV. 1965. *Computing Methods*, Volume 2. Pergamon.
5. NA T. Y. 1979. *Computational Methods in Engineering*: *Boundary Value Problems*. Academic Press.
6. CHEB-TERRAB, E.S., DUARTE, L.G.S., da MOTA, L.A.C.P. 1998. Computer algebra solving of second-order ODEs using symmetry methods. *Comput. Phys. Commun*., 108, 90.
7. STEPHANI, H. 1995. Differential equations: their solution using symmetries. In: *MacCallum, M.A.H., ed*. New York: Cambridge University Press.
8. CHEB-TERRAB, E.S., ROCHE A. D. 1999. Integrating Factors for Second-order ODEs. In: *J. Symbolic Computation*, 27, 501–519.
9. BOCHAROV, A.V., BRONSTEIN, M.L. 1989. Efficiency implementing two methods of the geometrical theory of differential equations: an experience in algorithm and software design. In: *Acta Appl. Math*., 16, 143.
10. WEIL, J.A. 2002. *Recent Algorithms for Solving Second-Order Differential Equations*. The Algorithm Project, http://pauillac.inria.fr/algo/seminars/sem01-02/weil.pdf; accessed 2002.
11. M. BRONSTEIN. 2001. Computer Algebra Algorithms for Linear Ordinary Differential and Difference Equations. *Progress in Mathematics*, Vol. 202, pp. 105–119.
12. WOLF, T., BRAND, A., MOHAMMADZADEH, M. 1999. Computer algebra algorithms and routines for the computation of conservation laws and fixing of gauge in differential expressions. In: *J. Symb. Comput.*, 27, 221–238.
13. M. F. SINGER and P. A. HENDRIKS. 1999. Solving Difference Equations in Finite Terms. *J. Symbolic Comput*. 27, 239– 259.
14. M. BRADY, B.K.P. HORN. 1981. Rotationally Symmetric Operators for Surface Interpolation. *Artifical Intelligence*, No. 654.

Reviewers:

doc. Ing. Ján Janík, CSc.
RNDr. Ľubomír Turňa, CSc.