Novel Computer Software for Interpolation and Approximation of Ravine and Stiff Digital Dependencies Using Root-Polynomoal and Root-Fractional-Rational Functions

Igor Melnyk[1], Mykhailo Skrypka[1], Alina Pochynok[2] and Olga Demyanchenko[3]
[1]*Department of Electronic Devices and Systems, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Beresteiska Avenue 37, 03056 Kyiv, Ukraine*
[2]*Research Institute of Electronics and Microsystems Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Beresteiska Avenue 37, 03056 Kyiv, Ukraine*
[3]*Institute of Applied Mathematics and Fundamental Sciences of the National University "Lviv Polytechnic", Mytropolyta Andreia Str. 5, Building 4, 79016 Lviv, Ukraine*
imelnik@phbme.kpi.ua, scientetik@gmail.com, alina_pochynok@yahoo.com, olgademyanchenko@gmail.com

Keywords: Interpolation, Approximation, Ravine Dependence, Stiff Dependence, Root-Polynomial Function, Root-Fractional-Rational Function.

Abstract: The article considers the possibilities of solving interpolation and approximation problems using special types of functions, such as root polynomials and root fractional rational, and provides relevant examples. It is prove, that the use of root polynomial functions is especially effective for interpolation and approximation of numerical dependencies with a ravine data set, and the use of root fractional rational functions gives the best results for various data sets with a more rigid functional dependence. To solve the approximation problem, a new approximation by reference points is proposed and tested. With a small number of points in the data set for the approximation problem, equal to twenty or less, the convergence of the proposed method is usually guaranteed. In general, the proposed algorithms are very universal and can be easily adapted to any complex problems. All the proposed methods are implemented and tested in the newly developed computer software created in the Python programming language.

1 INTRODUCTION

In general, the problem of interpolation and approximation of digital data is well known and described in various manual books on numerical methods and computational technologies [1 – 7]. But, although classical approaches in the theory of interpolation and approximation are still effectively used today, their application is often difficult or even impossible to solve complex modeling problems with minimization of data exchange. Furthermore, elaboration of novel approaches for solving interpolation and approximation tasks is very important today for organizing effective data exchange in modern network technologies, including cloud and fog computing [8].

Polynomial functions, especially when increasing the order of interpolation and approximation, always give unwanted outliers that are not typical for the behaviour of the function, and interpolation using splines requires a large number of reference points, which increases the set of necessary numerical data. There is often no compromise solution to achieve high interpolation accuracy with a small number of points, especially when using high-order functions [1 – 7]. For this reason, since 2019, a new approach to interpolation and approximation of ravine, vertex, and other types of rigid functions with local and extrema, based on the use of root-polynomial and root-fractional-rational functions, has been proposed and implemented. Basic principles of this approach have been formulated in the papers [9 – 11].

In the papers [9, 10], it has been shown and proven that for symmetric ravine data sets, the level of relative error of interpolation using root-polynomial functions from second to fifth order is in the range of a fraction of a percent.

Further research on root-polynomial functions has been provided in the papers [12 – 14]. Namely, in papers [12, 14], the algorithm of approximation of ravine data sets by the tangent to the linear branch of the ravine set has been proposed and described. Generally, such an approach is based on the well-known standard least-squares method of approximation [15, 16], but the necessity of obtaining the derivatives for root-polynomial functions leads to sophisticated calculations, especially for root-polynomial functions of higher order [12]. Using the advantages of the symbolic processor of the MATLAB system for scientific and technical calculation [3] has been proposed in the paper [13] for solving this sophisticated problem. A general description of this method of approximation has also been provided in the work [18].

In [13], a point-based approximation method was also proposed. It is simpler from a computational point of view than tangent approximation, therefore, if the convergence of this method is ensured, its use is preferable in terms of the use of computer resources. The conducted studies have shown that for 15–20 points of the approximated data set, the approximation error by points usually does not exceed several percent and approaches the error of the least squares method. Corresponding examples are given in [13].

And, finally, in the paper [14], generalized formulas for root-polynomial functions from second to sixth order are given, and conditions of convergence for the proposed methods for solving the tasks of interpolation and approximation are considered. In [14] are generally systemized all provided research, connected with root-polynomial functions and study its properties.

On the basis of the provided research, advanced computer software has been created using the programming language Python [17 – 20]. Firstly, this software has been created for solving the tasks of electron optics [21 – 27], but it became clear that the possibilities of using root-polynomial functions are significantly greater. Therefore, the aim of this paper is to describe elaborated computer software from the point of view of interpolation and approximation of ravine data sets by root-polynomial functions of different orders. Corresponding examples are given. Applying root-fraction-rational functions for solving interpolation tasks for stiff functions is also proposed.

2 BASIC CONCEPTION

As noted in the previous section, root polynomial functions were first proposed to be used to solve electron optics problems [21 – 27]. Namely, it was proven that the boundary trajectory of short-focus electron beams propagating in ionized gases corresponds to a ravine function with one global minimum and quasi-linear dependencies outside the minimum region [9, 11, 14]. It was also proven in [14] that the root polynomial functions correspond to the differential equation for the boundary trajectory of an electron beam propagating in an ionized gas. In general, this equation is based on the laws of electron optics [21 – 27] and the basic concepts of plasma physics [14, 29 – 31]. In general, these set of algebra-differential equations is written as follows [14]:

$$f=\frac{n_e}{n_{i0}-n_e};C=\frac{I_b\left(1-f-\beta^2\right)}{4\pi\varepsilon_0\sqrt{\frac{2e}{m_e}}U_{ac}^{1.5}};\frac{d^2r_b}{dz^2}=\frac{C}{r_b};\theta=\frac{dr_b}{dz}+\theta_s;$$

$$n_e=\frac{I_b}{\pi r_b^2};\ v_e=\sqrt{\frac{2eU_{ac}}{m_e}};$$

$$n_{i0}=r_b^2B_ipn_e\sqrt{\frac{\pi M\varepsilon_0n_e}{m_eU_{ac}}}\exp\left(-\frac{U_{ac}}{\varepsilon_0n_er_b^2}\right);\qquad(1)$$

$$\gamma=\sqrt{1-\beta^2};\ \tan\left(\frac{\theta_{min}}{2}\right)=\frac{10^{-4}Z_a^{4/3}}{2\gamma\beta^2};\ \tan\left(\frac{\theta_{max}}{2}\right)=\frac{Z_a^{3/2}}{2\gamma\beta^2};$$

$$\overline{\theta^2}=\frac{8\pi\left(r_bZ_a\right)^2dz}{n_e}\ln\left(\frac{\theta_{min}}{\theta_{max}}\right),\ \beta=\frac{v_e}{c},$$

where z is the longitudinal coordinate, r_b is the radius of the boundary trajectory of the electron beam, I_b is the electron beam current, U_{ac} is the accelerating voltage, p is the residual gas pressure, n_{i0} is the concentration of residual gas ions on the beam symmetry axis, n_e is the beam electrons' concentration, f is the residual gas ionization level, B_i is the gas ionization level, θ_{min} and θ_{max} are the minimum and maximum scattering angles of the beam electrons, corresponding to Rutherford model, $\bar{\theta}$ is the average scattering angle of the beam electrons, ε_0 is the dielectric constant, v_e is the average velocity of the beam electrons, m_e is the electron mass, c is the light velocity, γ is the relativistic factor, Z_a is the nuclear charge of the residual gas atoms, dz is the length of the electron path in the longitudinal direction at the current iteration.

The numerical solving of the set of equations (1) can be described with high accuracy by root-polynomial functions, which in generalized form is written as follows [9, 10]:

$$r_b\left(z\right)=\sqrt[n]{C_nz^n+C_{n-1}z^{n-1}+\ldots+C_1z+C_0},\qquad(2)$$

where n is the degree of the polynomial and the order of the root-polynomial function, and $C_0 - C_n$ are the polynomial coefficients.

Finding of coefficient $C_0 - C_n$ is the simple task, and with the known set of basic points $\{P_1(z_1, r_1), P_2(z_2, r_2), \ldots, P_{n+1}(z_{n+1}, r_{n+1})\}$ it led to analytical solving the set of linear equations:

$$r_j - C_0 = \sum_{i=1}^{n} C_i z_j^i, \forall j\left(j = 1\ldots n, i \in N\right), \quad (3)$$

where N is natural number.

Interpolation error ε(z) is estimated relative to numerical data $r_{num}(z)$, which has been obtained by solving the set of algebra-differential equations (1). This numerical data is considered as etalon values. Corresponded analytical relation for error estimation is follows [9, 10]:

$$\varepsilon(z) = \frac{\left|r_{num}(z) - r_{INT}(z)\right|}{r_{num}(z)} \cdot 100\%, \quad (4)$$

where r_{INT} is the result of interpolation using relation (2) and set of equations (3).

The analytical method of solving the set of linear equations (3) as well as the obtained sets of analytical relations for defining polynomial coefficients for root-polynomial functions (2) from second to sixth order are complexly presented in works [18, 20]. The general theory of root-polynomial functions is also given in [14]. Examples of solving such interpolation tasks are presented in papers [13].

It should be pointed out that the simulation of electron beam propagation in a soft vacuum of ionized gas is also an important industrial task. Solving this task is especially important for the elaboration and application in industry of high-voltage glow discharge electron guns [14]. Such guns are considered today as an advanced tools for providing in the soft vacuum such technological operations, as welding the contacts of electron devices [32, 33], deposition of ceramic coatings [33, 34], as well as refining refractory materials [35, 36].

Another concept for simulation relativistic electron beams propagating in plasma for microwave devices is based on the accurate processing and systematization of experimental data. This approach was proposed in [37, 38].

3 EXAMPLES OF SOLVING INTERPOLATION TASK

Example 1. Using root-polynomial function (2) of forth order interpolate the boundary trajectory of electron beam for the following parameters of simulation task: U_{ac} = 20 kV, I_b = 5 A, p = 3 Pa, operation gas nitrogen, start beam radius r_{b0} = 6 mm, start angle of beam convergence $\theta_0 = 12^0$, start point on z coordinate z_0 = 0.1 m.

Corresponded results of interpolation, obtained using elaborated software, is presented at Fig. 1.

For this task maximal error of interpolation is 2.9 % and average error in the range of z coordinate [0.1 m; 0.16 m] is 1,28 %. Should been pointed out, that for sixth-order root-polynomial functions values of errors are smaller: maximal value is 1.73 % and average value is 0.635 %. As proven in the [18, 20], generally even high-order functions give the smallest error of interpolation. Odd-order functions usually give the larger value of error. It has also been found that at large values of the accelerating voltage and small values of the current, the interpolation algorithm often diverges due to the small values of the minimum radius of the electron beam [18, 20].

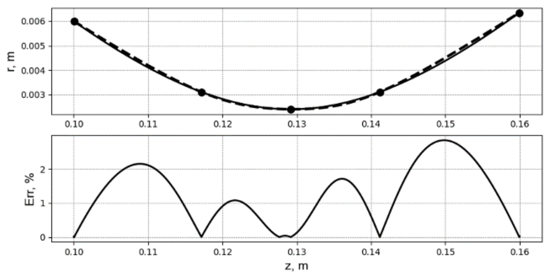

Figure 1: Solving of interpolation task for Example 1.

In the developed software, the set of equation (1) is first numerically solved using the fourth-order Runge-Kutt method [1, 2], and at the second stage, the obtained numerical data is transferred to the interpolation problem. To provide such access to a large amount of numerical data, the well-known tool for describing global variables in the Python programming language was used. Data transfer is implemented by analyzing the button click event using the functional programming paradigm [17 – 20]. Corresponding forth-order root-polynomial function for this example is as follows:

$$r(z) = \sqrt[4]{\begin{array}{c}1.534 \cdot 10^{-3} z^4 - 7.9 \cdot 10^{-3} z^3 + 1.5343 \cdot 10^{-4} z^2 - \\ -1.32326 \cdot 10^{-4} z + 4.2854 \cdot 10^{-7}.\end{array}}$$

Example 2. Interpolate manually using root-polynomial function ravine data set, presented in Table 1. It is clear that since the presented data set contains 7 values, only a sixth-order root-polynomial function is suitable for solving this task.

Table 1: Data set for ravine function of Example 2.

z, m	0.1	0.2	0.3	0.4
r, mm	7.0	5.0	4.0	3.5
z, m	0.5	0.6	0.7	–
r, mm	4.0	5.0	7.0	–

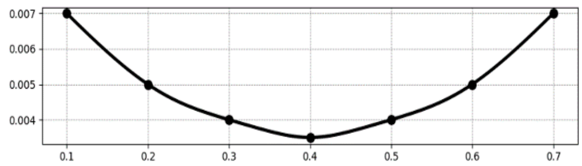

Figure 2: Solving of interpolation task for Example 2.

The root-polynomial function, obtained in this example, is written as follows:

$$r(z)=\sqrt[6]{\begin{array}{c}1.86\cdot 10^{-10}z^{6}-4.464\cdot 10^{-10}z^{5}+\\ +4.4104\cdot 10^{-10}z^{4}-2.2953\cdot 10^{-10}z^{3}+\\ +6.6558\cdot 10^{-11}z^{2}-1.027\cdot 10^{-11}z+6.69\cdot 10^{-13}.\end{array}} \qquad (5)$$

The coefficient C_0 – C_6 in relation (5) has been calculated by solving the set of equations (3) analytically. Corresponding analytical relations for defining the values of these coefficients by the coordinates of the points P_1 – P_7 are given in [13, 14].

4 METHOD OF SOLVING THE APPROXIMATION TASK BY THE REFERENCE POINTS

Firstly, it should be pointed out that the task of approximation, in contrast to the interpolation task, is usually solved for a much larger number of numerical data points, which describes the ravine function, than the value of polynomial order *n*. This presumption is very important because only if this rule is followed, the error of approximation is relatively small.

Taking into account this presumption, the task of approximation is simplified by its transforming to interpolation task. Interpolation is provided between n +1 selected reference points, where n is the order of root-polynomial function. Corresponded method of interpolation has been described in the pervious section of this paper. On the next iteration interpolation is provided again between following n +1 reference points, which give the maximal error in the pervious step. And finally, the two best curves with minimal integrated error are selected and the optimal solution is located between them with using well-known dichotomy method [1 – 5] for the values of polynomial coefficients [13].

Given this assumption, the approximation problem is simplified by converting it into an interpolation problem. Interpolation is carried out between n + 1 selected reference points, where n is the order of the root polynomial function. The corresponding interpolation method was described in the previous section of the article. At the next iteration, interpolation is again carried out between the next n + 1 reference points for which the maximum approximation error was obtained in the previous step. And finally, using the well-known dichotomy method [1 – 5], the two best curves with the minimum integral error are selected for the values of the polynomial coefficients, and the optimal solution is found between them [13]. The last step of solving the approximation problem can also be carried out repeatedly with a decrease in the error value. As test experiments have shown, with the correct implementation of this method for 10-20 reference points, the approximation error is small and approaches the corresponding value for the well-known least squares method [15, 16].

But there are many particularities in realizing this approach that are connected with the mathematical properties of root-polynomial functions. Complexly, this approach is based on dividing the root-polynomial function into linear branches and the region of the local minimum. A corresponding flowchart of this algorithm is presented in the paper [12].

Since the number of reference points n_a in the approximation task is significant value, its coordinate z and r are formed in one vector and given in square bracket, for example: $\mathbf{Z} = [z_1, z_2, \ldots, z_{na}]$. Such approach of presentation structured data is fully corresponding to basic conception of MATLAB system for scientific and technical calculations.

Considering corresponding example.

Example 3. Solving manually the task of approximation using root-polynomial functions from fourth to sixth order ravine data set, presented in Table 2.

Solving of this task using proposed algorithm of approximation is presented in Figure 3.

Table 2: Data set for ravine function of Example 3.

z, m	0.1	0.2	0.3	0.4	0.5
r, mm	4.394	4.02	3.594	3.201	2.792
z, m	0.6	0.7	0.8	0.9	1.0
r, mm	2.501	2.294	2.203	2.305	2.495
z, m	1.1	1.2	1.3	1.4	1.5
r, mm	2.801	3.193	3.607	3.96	4.405

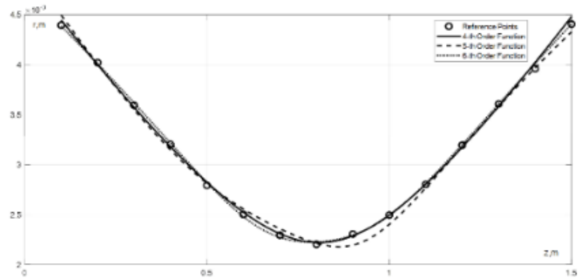

Figure 3: Solving of approximation task for Example 3.

It is clear from the obtained results of manual approximation that only the fifth-order root-polynomial function, as a function of odd order, gives a relatively large value of error. The total error of the approximation δ has been estimated as follows:

$$\delta = \sum_{i=1}^{n_a} \frac{\left| r_{ref}\left(z_i\right) - r_{APPR}\left(z_i\right) \right|}{r_{ref}\left(z_i\right)} \cdot 100\%, \quad (6)$$

where z_i are the z coordinate of the reference points, $r_{ref}(z_i)$ – r coordinate of the reference points, $r_{APPR}(z_i)$ – the corresponding values of the root polynomial-function at the reference points, $i = 1 \ldots n_a$. For solving approximation task value of δ was $\delta_4 = 1.693$ % for fourth-order root-polynomial function and $\delta_6 = 3.938$ % for sixth-order function. The fourth-order root-polynomial function has been obtained by solving this task is written as follows:

$$r\left(z\right) = \sqrt[4]{\begin{gathered}8.369 \cdot 10^{-10} z^4 - 2.65 \cdot 10^{-9} z^3 + 3.495 \cdot 10^{-9} z^2 - \\ -2.224 \cdot 10^{-9} z + 5.8068 \cdot 10^{-10}.\end{gathered}}$$

Also, provided research has shown that the proposed approximation algorithm always gives a smaller error if the reference points are not located exactly uniformly but with a slight deviation along the radius coordinate [12 – 14].

5 ADVANCED POSSIBILITIES OF USING ROOT-FRACTIONAL RATIONAL FUHCTIONS

Unfortunately, the convergence of interpolation methods based on the use of relations (2, 4) significantly depends on the maximum value of the derivative for the set of interpolated data and is not always guaranteed [10, 14]. Therefore, the use of root-factional-rational functions has been proposed. In a generalized, form such a function is written as follows:

$$r\left(z\right) = \sqrt[n]{\frac{z^n + C_1 z^{n-1} + \ldots + C_{n-1} z + C_n}{C_{n+1} z^{n-1} + \ldots + C_{2n} z + C_{2n+1}}}. \quad (7)$$

Generally, the task of interpolation by using function (7) leads to the analytical solution of such a system of linear equations:

$$\sum_{i=1}^{n} C_i z_j^i - \left(r_j\left(z_j\right)\right)^n \sum_{i=n+1}^{2n+1} C_i z_j^i = -z_j^n; \; j \in N, \; j \le 2n+1. \quad (8)$$

The developed computer software makes it possible to study the properties of interpolation of various analytical dependencies using fractional rational functions. The mathematical function interpolated over the specified interval is entered into the appropriate text field. To ensure that the corresponding calculations are carried out at the programming level, the well-known concept of lambda-functions in the Python programming language is used [17 – 20]. Let's consider a relevant example.

Example 4. Describe the analytical function $r(z) = \exp\left(\frac{-z^4}{2}\right) + 1.5$ by a fourth-order root-fractional-rational function. Providing interpolation using the following reference points: $z_1 = 0.045$; $z_2 = 0.225$; $z_3 = 0.405$; $z_4 = 0.8775$; $z_5 = 1.3545$, $z_6 = 1.8$, $z_7 = 2.295$, $z_8 = 2.79$, $z_9 = 3.195$.

The result of interpolation for a given set of numerical data is presented in Fig. 4. Clear, that the interpolation error for this example does not exceed $5 \cdot 10^{-3}$ %.

Generally, the provided research shows, that root-fractional-rational functions written in the form (7) can be used to solve a wide range of practical scientific and engineering problems. Some of these problems are as follows.

1) To interpolate the magnetic field of focusing lenses in tasks of electron optics [28 – 33].
2) Interpolation of the values of probability distribution and probability density functions in problems of probability theory and mathematical statistics. The main feature here is the high degree of stiffness of such functions and the impossibility of obtaining an analytical expression for the inverse function in a simple algebraic form [15, 16].
3) Interpolation of membership function values in fuzzy logic problems.

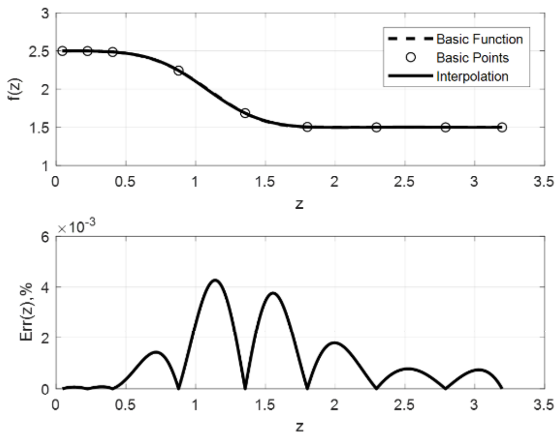

Figure 4: The result of interpolation by a fourth-order root-fractional-rational function for example 4

Generally, the main aspects of using root-fraction-rational functions (7) as an advanced method of interpolation and approximation for the stiff function are under study today.

6 OBTAINED RESULTS AND RECOMMENDATIONS

As it is clear from the examples presented in previous sections, elaborated computer software is a very effective tool for solving the tasks of interpolation and approximation. Root-polynomial functions are effective for solving such problems for ravine data sets, which can be obtained both by numerical calculations and experimentally. Obtained experience with the simulation of the boundary trajectory of an electron beam confirms that changing an enormous amount of numerical data to root-polynomial interpolation is the best way to solve the sophisticated tasks of electron-beam technologies. Such data transformation leads to a significant reduction in processor time and memory resources while maintaining the same accuracy in calculations. Therefore, such an approach is very effective in solving practical tasks, which lead to analyzing the ravine functions, including different physical tasks, tasks of probability theory, of economics, social science, etc. The advanced possibilities of using proposed approaches to solving different interpolation and approximation tasks are the subjects of future research.

7 CONCLUSIONS

The created computer software and the results of the provided research show that the use of root-polynomial functions is a very effective tool for interpolation and approximation of the data sets of ravine functions. Such an approach leads to a significant reduction of computer resources in the case of solving sophisticated mathematical tasks. The use of root-fractional-rational functions can also be considered an advanced possibility for solving the tasks of computation physics, probability theory, and fuzzy logic. The use of root-fractional-rational functions is especially effective for the interpolation of data sets with stiff functional dependences. Generally, the obtained results may be interesting to a wide range of specialists in the field of computational mathematics.

REFERENCES

[1] S.C. Chapra and R.P. Canale, Numerical Methods for Engineers, 7th ed. New York: McGraw Hill, 2014, 992 p.

[2] M.K. Jain, S.R.K. Iengar, and R.K. Jain, Numerical Methods for Scientific and Engineering Computation. New Delhi: New Age International Pvt. Ltd., 2010, 733 p.

[3] J.H. Mathews and K.D. Fink, Numerical Methods: Using Matlab. Amazon, 1998, 680 p.

[4] I.N. Bronshtein, K.A. Semendyayev, G. Musiol, and H. Mühlig, Handbook of Mathematics, 5th ed. Berlin: Springer, 2007, 1164 p.

[5] M. Abramowitz and I.A. Stegun, Eds., Handbook on Mathematical Functions with Formulas, Graphs, and Mathematical Tables. Washington: National Bureau of Standards, 1964, 1046 p.

[6] G.M. Phillips, Interpolation and Approximation by Polynomials. Cham: Springer, 2023, 312 p. [Online]. Available: http://bayanbox.ir/view/251880397 4255898294/George-M.-Phillips-Interpolation-and-Approximation-by-Polynomials-Springer-2003.pdf.

[7] N. Draper and H. Smith, Applied Regression Analysis, 3rd ed. New York: Wiley Series, 1998, 706 p. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9628935.

[8] I.V. Melnyk and A.O. Luntovskyy, "Estimation of Energy Efficiency and Quality of Service in Cloud Realizations of Parallel Computing Algorithms for IBN," in Future Intent-Based Networking: On the QoS Robust and Energy Efficient Heterogeneous Software Defined Networks, Lecture Notes in Electrical Engineering, vol. 831, M. Klymash, M. Beshley, and A. Luntovskyy, Eds. Cham: Springer, 2022, pp. 339–379. ISBN 978-3-030-92433-1. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-92435-5_20.

[9] I. Melnyk, S. Tuhai, and A. Pochynok, "Interpolation of the Boundary Trajectories of Electron Beams by the Roots from Polynomic Functions of Corresponded Order," in Proc. 2020 IEEE 40th Int. Conf. Electronics and Nanotechnology (ELNANO), 2020, pp. 28–33. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=9085228.

[10] I.V. Melnyk and A.V. Pochynok, "Study of a Class of Algebraic Functions for Interpolation of Boundary Trajectories of Short-Focus Electron Beams," System Researches and Information Technologies, no. 3, pp. 23–39, 2020. [Online]. Available: http://journal.iasa.kpi.ua/issue/view/13416/7188.

[11] I.V. Melnyk, S.B. Tuhai, and A.V. Pochynok, "Interpolation Functions for Describing the Boundary Trajectories of Electron Beams Propagated in Ionised Gas," in Proc. 15th Int. Conf. Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET–2020), 2020, pp. 79–83. [Online]. Available: https://ieeexplore.ieee.org/document/9088609.

[12] I. Melnyk and A. Pochynok, "Basic Algorithm for Approximation of the Boundary Trajectory of Short-Focus Electron Beam Using The Root-Polynomial Functions of the Fourth and Fifth Order," System Research and Information Technologies, no. 3, pp.

127–148, 2023. [Online]. Available: http://journal.iasa.kpi.ua/article/view/290474/284062.

[13] I. Melnyk, S. Tuhai, M. Skrypka, T. Khyzhniak, and A. Pochynok, "A New Approach to Interpolation and Approximation of Boundary Trajectories of Electron Beams for Realizing Cloud Computing Using Root-Polynomial Functions," in Information and Communication Technologies and Sustainable Development (ICT&SD 2022), Lecture Notes in Networks and Systems, vol. 809, S. Dovgyi, O. Trofymchuk, V. Ustimenko, and L. Globa, Eds. Cham: Springer, 2023, pp. 395–427. ISBN 978-3-031-46879-7. [Online]. Available: https://doi.org/10.1007/978-3-031-46880-3_24.

[14] I. Melnyk and A. Pochynok, "Theoretical Justification of Application Possibility of Different Order Root-Polynomial Functions for Interpolation and Approximation of Boundary Trajectory of Electron Beam," Radioelectronics and Communications Systems, vol. 66, no. 1, pp. 53–73, 2023. [Online]. Available: https://radioelektronika.org/issue/view/2023-02.

[15] E. Wentzel and L. Ovcharov, Applied Problems of Probability Theory. Mir, 2002, p. 432. [Online]. Available: https://mirtitles.org/2022/06/03/applied-problems-in-probability-theory-wentzel-ovcharov/.

[16] J.A. Gubner, Probability and Random Processes for Electrical and Computer Engineers. Cambridge, UK: Cambridge University Press, 2006. [Online]. Available: http://www.cambridge.org/gb/academic/subjects/engineering/communications-and-signal-processing/probability-and-random-processes-electrical-and-computer-engineers.

[17] P. Robbins, Python Programming for Beginners. Independently published, 2022, p. 114. [Online]. Available: https://www.amazon.com/Hweryho-Python-Programming-Beginners-Hands/dp/B0BTRRLCYZ/ref=zg_bs_285856_sccl_6/144-8112897-9282521?psc=1.

[18] W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter, 3rd ed. O'Reilly Media, 2023, p. 579.

[19] F. Chollet, Deep Learning with Python, 2nd ed. Manning, 2022, p. 509. [Online]. Available: https://www.amazon.com/Learning-Python-Second-François-Chollet/dp/1617296864/ref=zg_bs_285856_sccl_17/144-8112897-9282521?psc=1.

[20] M. Lutz, Learning Python, 5th ed. O'Reilly, 2013, p. 1643..

[21] S. Schiller, U. Heisig, and S. Panzer, Electron Beam Technology. New York: John Wiley & Sons, 1982, p. 509.

[22] M. Szilagyi, Electron and Ion Optics. Springer Science & Business Media, 2012, p. 539. [Online]. Available: https://www.amazon.com/Electron-Optics-Microdevices-Miklos-Szilagyi/dp/1461282470.

[23] S.J.R. Humphries, Charged Particle Beams. Courier Corporation, 2013, p. 834. [Online]. Available: https://library.uoh.edu.iq/admin/ebooks/76728-charged-particle-beams---s.-humphries.pdf.

[24] E. Kasper and P. Hawkes, Principles of Electron Optics: Applied Geometrical Optics. Elsevier Science, 1989, p. 1188. [Online]. Available: https://books.google.com.ua/books?id=XHEDruIilgEC&hl=uk&source=gbs_book_similarbooks.

[25] P. Grivet, P.W. Hawkes, and A. Septier, Electron Optics. Elsevier, 2013, p. 412. [Online]. Available: https://books.google.com.ua/books?id=jDgvBQAAQBAJ&hl=uk&source=gbs_book_similarbooks.

[26] J. Orloff, Handbook of Charged Particle Optics. CRC Press, 1997, p. 528. [Online]. Available: https://books.google.com.ua/books?id=_rYbAyf-iAkC&hl=uk&source=gbs_similarbooks.

[27] A.B. El-Kareh, Ed., Electron Beams, Lenses, and Optics. Academic Press, 2012, p. 428.

[28] B.M. Smirnov, Theory of Gas Discharge Plasma. Springer, 2015, p. 433. [Online]. Available: https://www.amazon.com/Theory-Discharge-Springer-Optical-Physics/dp/3319110640.

[29] M.A. Lieberman and A.J. Lichtenberg, Principles of Plasma Discharges for Materials Processing. New York: Wiley Interscience, 1994, p. 572. [Online]. Available: https://people.physics.anu.edu.au/~jnh112/AIIM/c17/Plasma_discharge_fundamentals.pdf.

[30] Yu.P. Raizer, Gas Discharge Physics. New York: Springer, 1991, p. 449. [Online]. Available: https://d-nb.info/910692815/04.

[31] A.A. Druzhinin, I.P. Ostrovskii, Y.N. Khoverko, N.S. Liakh-Kaguy, and A.M. Vuytsyk, "Low temperature characteristics of germanium whiskers," Functional Materials, vol. 21, no. 2, pp. 130–136, 2014.

[32] A.A. Druzhinin, I.A. Bolshakova, I.P. Ostrovskii, Y.N. Khoverko, and N.S. Liakh-Kaguy, "Low temperature magnetoresistance of InSb whiskers," Materials Science in Semiconductor Processing, vol. 40, pp. 550–555, 2015..

[33] T.O. Prikhna et al., "Electron-Beam and Plasma Oxidation-Resistant and Thermal-Barrier Coatings Deposited on Turbine Blades Using Cast and Powder Ni(Co)CrALY(Si) Alloys I. Fundamentals of the Production Technology, Structure, and Phase Composition of Cast NiCrAlY Alloys," Powder Metallurgy and Metal Ceramics, vol. 61, no. 1–2, pp. 70–76, 2022. [Online]. Available: https://www.springer.com/journal/11106.

[34] T.O. Prikhna et al., "Electron-Beam and Plasma Oxidation-Resistant and Thermal-Barrier Coatings Deposited on Turbine Blades Using Cast and Powder Ni(Co)CrAlY(Si) Alloys Produced by Electron-Beam Melting II. Structure and Chemical and Phase Composition of Cast CoCrAlY Alloys," Powder Metallurgy and Metal Ceramics, vol. 61, no. 3–4, pp. 230–237, 2022.

[35] S.V. Akhonin, B.A. Berezos et al., "Production of high-strength titanium alloy VT22 by electron beam melting method," Advanced in Electrometallurgy, no. 3(131), pp. 8–14, 2018.

[36] T. Kemmotsu, T. Nagai, and M. Maeda, "Removal Rate of Phosphorous from Melting Silicon," High Temperature Materials and Processes, vol. 30, no. 1–2, pp. 17–22, 2011.

[37] V.M. Lazurik, V.T. Lazurik, G. Popov, and Z. Zimek, "Two-parametric model of electron beam in computational dosimetry for radiation processing," Radiation Physics and Chemistry, vol. 124, pp. 230–234, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0969806X1530133X.

[38] V. Lazurik and V. Moskvin, “Monte Carlo calculation of charge-deposition depth profile in slabs irradiated by electrons,” Nuclear Instruments and Methods in Physics Research, Section B: Beam Interactions with Materials and Atoms, vol. 108, no. 3, pp. 276–281, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/0168583X95010521.