data

MDPI

Article

Geometrical Platform of Big Database Computing for Modeling of Complex Physical Phenomena in Electric Current Treatment of Liquid Metals

Yuriy Zaporozhets [1], Artem Ivanov [1,*] and Yuriy Kondratenko [2]

[1] Institute of Pulse Processes and Technologies of National Academy of Science of Ukraine, Mykolaiv 54018, Ukraine; yurii.zaporozhets@nuos.edu.ua

[2] Petro Mohyla Black Sea National University, Mykolaiv ~~0~~54003, Ukraine; y_kondrat2002@yahoo.com

* Correspondence: artiomsan@gmail.com; Tel.: +38-051-258-7147

Received: 29 June 2019; Accepted: 16 August 2019; Published: 5 October 2019

Abstract: According to the principles of multiphysical, multiscale simulations of phenomena and processes which take place during the electric current treatment of liquid metals, the need to create an adjustable and concise geometrical platform for the big database computing of mathematical models and simulation~~s~~ is justified. In this article, a geometrical platform was developed based on approximation~~s~~ of boundary contours using arcs for application of the integral equations method and matrix transformations. This method achieves regular procedures using multidimensional scale matri~~x~~ces for big data transfer and computing. The efficiency of this method was verified by computer simulation and used for different model contours, which are parts of real contours. The obtained results showed that the numerical algorithm was highly accurate based on the presented geometrical platform of big database computing and that it possesses a potential ability for use in the organization of computational processes regarding the modeling and simulation of electromagnetic, thermal, hydrodynamic, wave, and mechanical fields (as a practical case in metal melts treated by electric current). The efficiency of this developed approach for big data matrices computing and equation system formation was displayed, as the number of numerical procedures, as well as the time taken to perform them, were much smaller when compared to the finite element method used for the same model contours.

Keywords: geometrical platform; big databases; modeling; approximating arcs; numerical procedure; integral equations; kernels; electric current treatment

1. Introduction

The study of complex processes and phenomena by mathematical modeling is a natural way to improve and enhance the efficiency of technical methods and technologies. Today, efficiency upgrades in every industry and scientific field involve data processing and mathematical modeling. In some areas of human activity, information technology for data processing is an integral part of production process management. Mathematical models are also often used alongside data obtained from nature experiments, and physical model construction is firmly established in the practice of engineering design.

There are also technical spheres where the experimental study of the internal content of current processes and real measurements of local physical parameters of reagents or interacting media are not currently possible. In these cases, the only way to identify the influence of setting exposures on controlled parameters, intermediate data, and final results is by using a computational method that utilizes generalized data of empirical dependencies or numerical simulations based on existing theories of the corresponding phenomena.

An illustrative example of this kind of situation is the development of empirical equations which are used to determine the concentration of an aqueous solution of lithium bromide in a refrigerating chamber system by measuring the mediated physical parameters of the solution outside the work space, which is not directly accessible [1]. Naturally, to obtain sufficiently reliable results using these calculations, it is necessary to accumulate a significant amount of experimental data.

A different approach but under similar circumstances within the field of robotics was applied in [2–8], where the problem of ensuring the reliable retention of massive technological tools by a mobile robot on an irregular surface was considered. Under these conditions, it was necessary to build a special computational model to track the forces on the patch of the robot's magnetic clutch with the working surface using the online processing of continuous data flow from magnetic sensors. The basis of this model is an algorithm for the flexible, adaptive formation of geometric data, that is, parameters of a point grid on the control lines of the analytical continuation elliptic equation solutions.

Similar situations, with real data processing, also occur in other areas. In particular, various sets of measures are used in the foundry industry to solve the fundamental problem of ensuring cast metal product quality. Energy treatment methods have been proven to work well when the melt is exposed to external fields with different physical nature. As a rule, these treatment methods initiate and generate a whole complex of physical processes, which need to be controlled in order to obtain an adequate effect. As an example, conductive electric current treatment (CECT) used with liquids, crystalizing metals, and alloys is an effective modern method of forming favorable casting structures and properties. This was confirmed by many publications, especially over the past decade [9–16]. However, with CECT, a number of interrelated phenomena arise that need to be controlled (electromagnetic, thermal, hydrodynamic, wave, mechanical, etc.), and the solution to this problem remains unclear [17].

Therefore, various methods regarding physical and numerical experiments are widely used. Physical experimental methods are expensive, difficult, and are not as universal as numerical simulation methods, due to the necessary applying the so-called "black box" method. This means that the parameters of the impact on the melt are compared with the results already established by metallography techniques and mechanical tests. For example, the results of a reliability analysis of a fleet of mechanisms performed in [18] using "white box" and "black box" analytical methods alongside a simulation method showed definite similarity with the model, especially in the case of the "white box" method. Hence, adequate modeling could at least be used to check results.

There remain a number of problems associated with the application of numerical modeling to coupled electromagnetic, hydrodynamic, thermal, and wave fields and diffusion processes. Although numerical exploration is cheaper and more variable in its controlled and control parameters [19–22], as the experiment shows, solving these problems is difficult and often impossible. Therefore, simplification of these tasks in various ways is necessary. For example, 650,000 elements are required in order to calculate hydrodynamic processes at the volume of the melt in small experimental containers 63 mm high and 50 mm in diameter using the currently popular finite element method (FEM) [23]. However, millions of elements would be required if one needed to solve this problem at a large volume, considering grid refinement features of micro-inclusions and calculating coupled electromagnetic, thermal, hydrodynamic, and other processes in the dynamics. Moreover, the estimation of any numerical value in the volume under study requires that the calculation procedure be performed for the whole object, taking into account variations in the initial and boundary conditions. This affects the convenience, speed, and accuracy of the calculations, as well as the requirements for any hardware and software used for the mathematical modeling.

As a rule, it is possible to simplify the algorithms used to solve these tasks, but this is to the detriment of the accuracy and adequacy of the described situations. Therefore, searching for algorithms that optimize the processing and preparation of large arrays and continuous data flow for numerical method implementation is often performed as a scientific and technical task in data processing and computational mathematics.

In this article, special attention is paid to the geometric platform of big database computing to model complex physical phenomena in the electric current treatment of liquid metals.

2. Problem Statement

In light of the noted specificity of the problems of numerical simulation of CECT processes, a promising approach to solving a set of mathematical physics equations seems to be the application of a more concise and compact method instead of (or together with) FEM, based on an integral representation of the characteristics of physical fields in the area where they interact with the medium. This especially concerns the situation when the medium, besides all, changes its properties.

In general, this approach is called "the method of integral equations (IEs)". Its advantages become apparent in situations where the spatial interaction region (i.e., the set where a process or phenomenon is localized) can be represented equivalently on a set of smaller dimensionality: from the volume of the body (space) to its surface area; and in two-dimensional problems, from the figure area to the boundary contour [24].

In fact, such a transition is completely analogous to the representation of solutions of the Dirichlet and Neumann problems for the differential Laplace and Poisson equations by Green's integral formulas, in which the main properties of physical fields are displayed by simple and double-layer potentials [25]. These potentials are expressed by integral operators (IOs) with the kernels $K_1(P,M)$ and $K_2(P,M)$ of the form:

$$K_1(P,M)=\frac{1}{r_{P,M}};\quad K_2(P,M)=\frac{\partial}{\partial n}\left(\frac{1}{r_{P,M}}\right)\ , \tag{1}$$

where $r_{PM}=\sqrt{\left(x_P-x_M\right)^2+\left(y_P-y_M\right)^2+\left(z_P-z_M\right)^2}$ is the radius vector drawn from the integration point M on the surface (or contour) to some observation point P. That is, kernels are functions of two points' coordinates; the kernel K_2 is actually a partial derivative of K_1 in the direction normal to the surface of integration.

The derivative above is calculated by the following formulae.

For the 3D model:

$$K_2(P,M)=\frac{\cos(\overline{r}_{PM},\overline{n}_P)}{r^2{}_{PM}}\ ; \tag{2}$$

and for the 2D model:

$$K_2(P,M)=\frac{\cos(\overline{r}_{PM},\overline{n}_P)}{r_{PM}}, \tag{3}$$

where $\cos(\overline{r}_{PM},\overline{n}_P)=\cos(\overline{r}_{PM},\vec{e}_x)\cdot\cos(\overline{n}_P,\vec{e}_x)+\sin(\overline{r}_{PM},\vec{e}_y)\cdot\sin(\overline{n}_P,\vec{e}_y)$ is the cosine of the angle between radius-vector $\overline{r}_{PM}$ and normal $\overline{n}_P$ at point P ($\vec{e}_x$ is the unit vector of axis x, and $\vec{e}_y$ is the unit vector of axis y).

In most cases, the numerical solution of an IE with kernels of the specified type is carried out by replacing the integrals with finite sums having a sufficiently high degree of discretization of the integration surface (contour) and reducing them to systems of linear algebraic equations (SLAEs). However, it should be noted that, depending on the boundary conditions and parameters connected with the characteristics of the media, the SLAEs may be ill-conditioned—the task becomes incorrect [26], and its solution requires the application of special methods such as singular value decomposition (SVD).The effectiveness of this procedure has been repeatedly noted [27,28] (particularly in [29]), but its implementation is connected with multiple matrix transformations, the computation of which requires the creation of a sufficiently large database related to the configuration of the integration surfaces.

From the above, it is clear that in the transition from IE to SLAE and the success of their solution, the primary point of methodological importance is the procedure of the rational presentation of the geometrical parameters of the corresponding sets, accounting for the need to use them explicitly in the algorithmic structure of mathematical operations of differentiation and integration on curvilinear surfaces, as well as calculating two-point operators in integral equations.

Thus, in order to ensure the computational process, first of all, for the formation of matrices of coefficients of the SLAE, which approximates the set of IEs of the complex of physical processes of CECT, it is necessary to create a ramified scheme for calculating the geometric parameters of the elements of curvilinear surfaces and contour lines, their curvatures, tangents and normal vectors to them, and so on—that is, a kind of geometrical platform for the prompt update of large databases of numerical models.

The basis for the implementation of the above approach should be a rational method of setting the configuration of computational areas (bodies) and the optimal procedures for simulating the transduction of their influence into local zones of field. Medium interactions should serve as the basis for the implementation of the stated approach.

Despite the existence of a number of rather good algorithmic procedures [30], the composition of adequate geometric models of complex technical objects for the numerical analysis of various physical processes is still an urgent problem, since the well-known and accessible software from this arsenal is not always suitable due to its profile specialization for some classes of tasks [31].

Particularly, traditional combinational or generational approaches impose definite limitations: firstly, they practically do not allow any variations; secondly, being a locally deterministic procedure, they do not guarantee the convergence of the grid to a given area on a complicated surface [32].

It is precisely this situation that we encounter when considering a complex of interconnected physical phenomena with a high degree of uncertainty of the external and internal boundaries of their localization areas (patches, bubbles) in the CECT process.

The purpose of this work was to create a geometrical platform for the operative formation and processing of large databases of numerical models of physical processes and phenomena in the technological operations of a foundry, which are inaccessible to direct observation.

3. Method for Specifying Geometric Objects of Complicated Shape

The main idea of the procedure for forming a geometrical platform which serves as the basis for the computational operations of the outlined problems is similar to that presented in [2]. In this paper, this concept has undergone further development and acquired a complete form. Unlike the three conceptually different approaches specified in [30], it is *synthetic*, uniting the elements of object modeling according to drawings of their projections, the parametric description of surfaces, and the composition of basic topological primitives. Its essence is as follows.

Virtually any complicated detail or construction can be described (depicted) by a set of its projections and sections by parallel (or non-parallel) planes, so that the borders of the computational areas will represent closed contours formed by curves or pieces of straight lines. Examples include wing profiles, ship hull frames, magnetohydrodynamic (MHD) flow channels, melt ladle profiles, etc. A set of such two-dimensional (flat) contours allows a 3D-model of objects to be obtained from them rather simply from slices. In [32], this technique was called the "method of frame sections". A similar idea was implemented with the help of topological primitives and R-functions in [33], but this approach seems rather cumbersome for tasks with multiple runs of varying modes of CECT processes.

So, the problem comes down to the task of assigning a set of closed curvilinear contours with different curvature segments in the simplest way. It is almost always possible to approximate such a contour with sufficient accuracy by a sequence of joined arcs of circles of suitable radius. The agnetoarsenal of approximation methods is unlimited: from the visual splitting of the contour into segments and the assignment of coordinates of the docking points, to the fully automated conversion of scanned drawings into digital arrays.

Thus, in our algorithm an arc of a circle is taken as a basic primitive; its limiting forms are the corner point with a zero (conditionally) radius, and a straight line segment with an infinitely large radius.

For a representation of the preparatory stage of the procedure for forming the geometrical platform, let us consider the configuration of the ladle section with the molten metal, into which a pulsing current is fed through the electrodes (Figure 1). For simplicity, we consider that the ladle is extended along the *Z* axis (beyond the drawing plane), and by following literally all the bends of the border surfaces, we obtain a shape whose contour needs to be approximated by the mentioned primitives. According to Figure 1, we visually select 11 segments of straight lines and 2 arcs of the contour on its curvilinear part. Thus, we define the total number of approximating segments (arcs): $M = 13$.

Then, we perform the operation of specifying the spatial position and key geometric parameters of the arcs.

Firstly, all arcs and contour segments, starting at any point, are numbered one by one counterclockwise according to the standard positive direction at planimetric integration. The initial (*IP*) and the final (*EP*) points of each arc are designated by the corresponding indices, so that the end points of the previous arcs coincide with the initial points of the subsequent arcs: $EP_k \equiv IP_{k+1}$ (Figure 2).

In the next step we specify the coordinates ($X_{ip,j}$, $Y_{ip,j}$) of the first and every subsequent initial points of arcs, which coincide with the coordinates of the end points of previous arcs ($X_{ep,j-1}$, $Y_{ep,j-1}$), where *j* is the arc number from the common quantity of arcs *M*.

After that we define the curvature radii of the arcs $R_{c,j}$, but as for their value there exists the restriction that $R_{c,j}$ must be not less than half of the distance $l_{a,j}$ from the beginning to the end of the arc (i.e., of the chord). If $R_{c,j} = 0.5\ l_{a,j}$, it means the arc is a semicircle; if $R_{c,j} > 100\ l_{a,j}$, then the arc becomes a piece of the straight line.

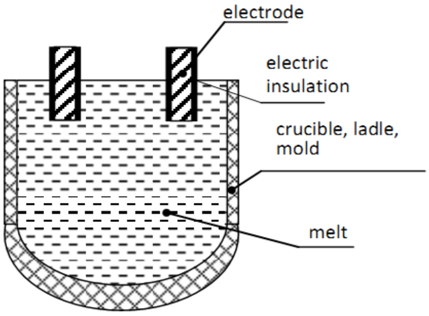


Figure 1. Principal constructive scheme of the conductive electric current treatment (CECT).

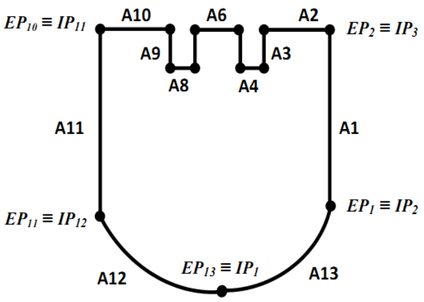


Figure 2. Ladle contour approximated by straight lines and arcs.

Moreover, we assign to the radius the sign: "+" or "−". A positive value of the *Rc,j* indicates that the center of arc curvature is situated on the right side of chord direction, passing the arc from the initial point to the end point. Accordingly, a negative value of *Rc,j* indicates that the center of the arc curvature is situated on the left side of the chord direction.

The position of the center of arc curvature relative to the chord, which is oriented from the beginning to the end along the contour, characterizes the bending of the arc—that is, its concavity or convexity. If the radius is positive, the arc will be concave, will pass to the left of the chord, penetrating into the area covered by the contour, and reducing its area; with a negative radius, the arc will be convex, located to the right of the chord, respectively, expanding the area.

Therefore, we have a set of parameters for each approximating arc: *X*, *Y* coordinates for initial and end points, as well as the radius of curvature with the sign ±*R*. For convenience of use and the further computer processing of these data by the software of widespread versions of MATLAB or other languages, it is expedient to present them as a compact file in the form of a text file. Into this file, we enter in the usual way a two-dimensional matrix with the size of 5 × *M*, where *M* is the total number of approximating arcs. The structure of the matrix is as follows:

1st line: *Xip,j*—coordinates of the initial points of the arcs;
2nd line: *Yip,j*—coordinates of the initial points of the arcs;
3rd line: *Xep,j*—coordinates of the end points of the arcs;
4th line: *Yep,j*—coordinates of the end points of the arcs;
5th line: *Rc,j*—radii of curvature of arcs with a sign.
The data are entered in MATLAB matrix format by column:

```
entArray = [[Xi1; Yi1; Xe1; Ye1; Rc1], ..., [XiM; YiM; XeM; YeM; RcM]].
```

An example listing (*listing 1*) of such a file is given below (see Table 1); the listing data represent a contour in the form of a circle of unit radius $R_c = -1$ with the center at the beginning of the coordinates (0,0, 0), which is composed of six arcs of 60 degrees each with the countdown from the abscissa axis.

Table 1. The contour input data matrix.

Arc Coordinates	**1st arc**	**2nd arc**	**3rd arc**	**4th arc**	**5th arc**	**6th arc**
Variables	**MATLAB code** (listing 1)					
Xip,j	1.0000	0.5000	−0.5000	−1.000	−0.5000	0.5000
Yip,j	0.0	0.8660	0.8660	0.0	−0.8660	−0.8660
Xep,j	0.5000	−0.5000	−1.000	−0.5000	0.5000	1.000
Yep,j	0.8660	0.8660	0.0	−0.8660	−0.8660	0.0
Rc,j	−1.000	−1.000	−1.000	−1.000	−1.000	−1.000

4. Calculation of the Geometrical Parameters of Approximating Arcs

To perform the numerical integration of a vector function along a curvilinear contour, it is necessary, to divide it into elementary sections (i.e., arcs), which involves calculating the coordinates of the nodal points (i.e., the points of the conjugation of elementary arcs and their lengths, as well as the direction cosines of the tangents to them). In addition, IO kernels of the form (1) require the operation of the form (2), (3) by means of which from the scalar product of vectors $\overline{r}_{PM}$ and $\overline{n}_P$ the cosine of the angle between them is found.

The accuracy of integration is ensured by setting the smallest possible size of elementary arcs; however, their total number is limited by the capabilities of computing means.

Now, on the basis of item 3 data, it is possible to calculate every geometrical parameter necessary for the computation of the kernels of integral operators, which is executed in a few stages.

First of all, we need to calculate the coordinates of arcs' curvature centers, sector angles, directions of chords, and normals to the corresponding arcs.

To get the formulae for calculating the coordinates (Xcc,j, Ycc,j) of the arcs' curvature centers and the sector angles γ_j that correspond to each arc, let us consider Figure 3, which shows the coordinate plane with one arc as a separate part of the entire contour.

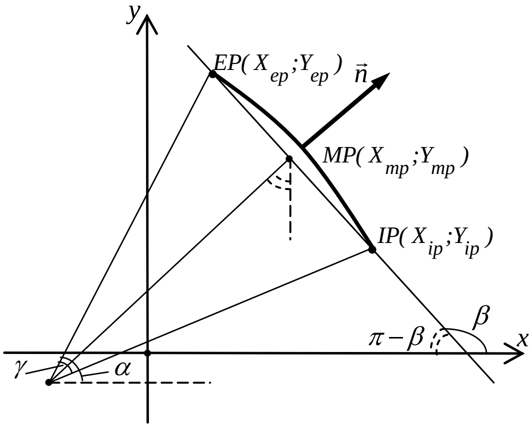


Figure 3. For determination of the main geometrical parameters of a separate arc.

The main designations in Figure 3 are *IP*—initial point, *EP*—end point, *CC*—center of the circle, *MP*—middle point of the arc's chord, α—angle of the arc chord inclination to the vertical, β—angle of the arc chord inclination to the horizontal.

First, we perform some elementary auxiliary counting:

- The coordinates of the *CP* point are calculated as the half-sum of corresponding coordinates of the initial and end points of a given arc
  $X_J = \frac{1}{2}\left(X_{ip} + X_{ep}\right); \mathrm{Y}_J = \frac{1}{2}\left(Y_{ip} + Y_{ep}\right)$; half of the length of *IP–EP* arc chord denoted by *lc:*

$$l_c = JH = \frac{1}{2}\sqrt{\left(X_{ip} - X_{ep}\right)^2 + \left(Y_{ip} - Y_{ep}\right)^2}\,. \tag{4}$$

- The height h_c of an isosceles triangle, formed by the chord and two radii $R_{c,l}$ and $R_{c,r}$ (left and right), drawn from the common curvature center of the arc to its ends IP and EP, which is the bisectrix of the sector angle γ as well:

$$h_c = \sqrt{R_c^2 - l_c^2}\,, \tag{5}$$

$$\gamma = 2arctg(l_c/h_c)\,. \tag{6}$$

- The length of each of the arcs and the total length of the contour:

$$l_a = \gamma\left|R_c\right| \quad \Rightarrow \quad L = \sum_j l_{\grave{a},j}\,. \tag{7}$$

For the convergence of subsequent calculations, it is very important to identify correctly the position of the chord and the direction of the normal with respect to the axes of coordinates by means of the angles α and β (Figure 3). In [2], this situation was resolved by the combinatorial method—by searching all possible variants of the arc, chord, and normal in the quadrants of the coordinate system, taking into account the convexity or concavity of the arc. Although the result thus obtained is reliable, the logic of the computational algorithm was rather cumbersome.

Note that it is possible to simplify the procedure for determining these angles: the normal to the arc is the prolongation of the radius drawn from the center to the midpoint of the arc, and its direction coincides with the direction of this radius. In this case, as previously indicated, the chord perpendicular to this radius has an orientation from the beginning of the arc to its end. That is, it can be considered as a vector which is rotated with respect to the normal at an angle of 90° counterclockwise. If such a vector is presented in complex form by the number

$$z_{ch} = (X_{ep} - X_{ip}) + i(Y_{ep} - Y_{ip}), \tag{8}$$

then its module will give the length of the chord,

$$|z_{ch}| = \sqrt{(X_{ep} - X_{ip})^2 + (Y_{ep} - Y_{ip})^2} = 2l_c, \tag{9}$$

where *lc* is a half-chord according to (4), and the argument will determine the required angle:

$$\beta = \arg z_{ch} \Rightarrow -\pi < \beta \le \pi. \tag{10}$$

Operations (8)–(10) are directly performed by MATLAB functions, and the result is transmitted without any additional manipulations to the corresponding executable block of the calculation program. We give a concise listing of these calculations in Table 2, where the number z_{ch} is given the name "comHord".

Table 2. Calculation of the chord inclination angle.

Variables	MATLAB Code (Listing 2)
comHord	= complex((xK − xH),(yK − yH));
Beta	= angle(comHord);

The next parameter related to the angle β is the angle α, which is the angle of inclination of the normal to the arc at its midpoint to the horizontal, coinciding with the angle of inclination of the chord to the vertical. Thus, the normal in the complex plane is 90° behind the chord, and therefore the angle α is determined quite simply:

$$\alpha = e^{-i\frac{\pi}{2}} \cdot \arg z_{ch} = (-i) \cdot \arg z_{ch},$$

or in a scalar form

$$\alpha = \beta - \pi/2. \tag{11}$$

Finally, we determine the coordinates of the point CC (i.e., the center of arc curvature):

$$X_{cc} = X_{mp} - h_c \cos\alpha; \quad Y_{cc} = Y_{mp} - h_c \sin\alpha. \tag{12}$$

The last item of the set of approximating arcs parameters that will be required for further calculations are the angles α_{IP} and α_{EP}, which determine the slope of the normals at the initial and end points of a given arc, which serve as reference points:

$$\alpha_{IP} = \alpha - \gamma/2; \quad \alpha_{EP} = \alpha + \gamma/2\ . \tag{13}$$

5. Procedure of Approximating Arcs Partition into Elementary Sections and Determination of their Geometric Parameters

To obtain a numerical solution of integral equations (IEs) which describes the physical phenomena in the CECT process, it is necessary to reduce from their continuous (integral) formulation to a discrete one as a sum of values of operators like (1) on elementary parts of the integration contour breakdown. At the same time, the number of such parts, that is, arc finite elements (AFEs), should be large enough to ensure the necessary accuracy of the finite element approximation of contour integrals.

Experience with similar calculations has shown that acceptable (engineering) accuracy can be achieved by setting the size (length) of the SLAE to at least less than 1% of the length of the integration contour or, in other words, more than 100 AFEs per contour. This means that the square matrix of the SLAE coefficients, which corresponds to the IEs, will contain more than 10^4 elements. If we take for example the contour in the form of a full circle of 360°, represented by Table 1, and divide it into AFEs by 1°, then we obtain a matrix of 130,000 elements.

Therefore, the first thing required to define the set of AFEs is to appoint the rational number of partitions of each arc into elementary sections and total the number of them.

The number of elementary sections m_j into which every arc is parted is preset in such a way that the ratio $\Delta l_j = l_j/m_j \approx \Sigma l_j/\Sigma\ m_j = L/M$ is approximately maintained. Here *j* is the number of the approximating arc along the contour.

The values of m_j may be set initially together with the array entArray when entering the input data of arcs approximating the contour, through the program interface (if provided), or manually using the command input ('arcNDel = '), where arcNDel is a vector-string of input values.

Then, we calculate:

- The sector angle of each elementary arc $\Delta\gamma_j = \gamma j/mj$;
- The corresponding lengths of each AFE as $\Delta lj = lj/mj$.

On the basis of these data, using the values of angles (13), we calculate the coordinates of the nodal points, that is, points of AFEs linking:

$$x_{j,i} = x_{cc,j} + |R_c| cos\left(\alpha_{IP,j} + i\Delta\gamma_j\right);\ \ y_{j,i} = y_{cc,j} + |R_c| sin\left(\alpha_{IP,j} + i\Delta\gamma_j\right), \tag{14}$$

where the index *i* denotes the number of a nodal point on the *j*-th approximating arc.

Now we can find the most important parameters of each AFE—the angle $\beta_{j,i}$ of inclination of its chord to the horizontal, and then the angle $\alpha_{j,i}$ of inclination of the normal $\overline{n}_{j,i}$ at the midpoint of the *i*-th elementary arc to the *x*-axis. To do this, using the coordinates of the nodal points $(x_{j,i}, y_{j,i})$, we again turn to the complex form of representing the vectors of chords and normals and apply the same procedure to each elementary arc, which is described by Formulas (8), (10), and (11):

$$\Delta z_{j,i} = (x_{j,i+1} - x_{j,i}) + i(y_{j,i+1} - y_{j,i});\ \beta_{j,i} = \arg\Delta z_{j,i}\ ;\ \alpha_{j,i} = \beta_{j,i} - \pi/2\,. \tag{15}$$

Finally, we form a system of collocation points $(\xi_{j,i}, \eta_{j,i})$ at which the values of integral functionals will be calculated and solutions of IEs will be built. Additionally, the convergence of the calculation process and accuracy of the obtained results will be checked at these points. It is convenient to place these points in the middle between the node points, and then their coordinates are determined similarly to Formula (13):

$$\xi_{j,i} = x_{cc,j} + |R_c| cos\,\alpha_{j,i};\ \ \eta_{j,i} = y_{cc,j} + |R_c| sin\,\alpha_{j,i}\ \ , \tag{16}$$

and with a sufficiently large number of the contour breakdowns on AFEs, it may be counted even simpler:

$$\xi_{j,i} = (x_{j,i} + x_{j,i+1})/2 \text{ and } \eta_{j,i} = (y_{j,i} + y_{j,i+1})/2\ . \tag{17}$$

6. Organizing the Computation of Integral Operator Kernels

The calculation of the kernels of integral operators is performed by Formulae (2) and (3). All of the variable values necessary for their use are determined according to procedures outlined in the previous section. Here we point out some features of their applications and a form of the database in which calculation results are arranged.

All variables in the given expressions contain two indices: the number of the approximating arc *j* and the number of a point on this arc *i*. Therefore, in the process of computing, the values of each variable are grouped into *M* linear matrices (row vectors) generally of different lengths (with a different number of elements).

At the same time, the result of calculating Formulae (2) and (3) is a value related to any two points *P* and *M* selected on the contour, which means that it is naturally supplied with two indexes—the numbers of these points *kP* and *kM* according to numbering on the contour. Therefore, the indexation of variable values must be transformed from two-digit to single-digit—that is, one vector must be built from M vectors by combining them in the required sequence. This procedure is called *concatenation*, and in the MATLAB environment it is performed as a usual matrix operation [34].

7. Pattern of the Procedure for the Formation of One-Dimensional (Linear) Arrays of Nodal Points Coordinates

To clarify the approach to the implementation of this procedure, we give a fragment of the program code with arrays named XNODEJ and YNODEJ (see Table 3):

Table 3. Formation of arrays of nodal points coordinates with names XNODEJ and YNODEJ.

Variables	MATLAB Code (Listing 3)
XNODEJ	= [];
YNODEJ	= [];
	for in = 1 : arcNumber
jar	= arcNDel(in)+1;
	for jm = 1 : jar
xNode(jm)	= XCent+abs(RK)*cos(AlfaH+(jm-1)*delgamma);
yNode(jm)	= YCent+abs(RK)*sin(AlfaH+(jm-1)*delgamma);
	end
XNODEJ	**= [XNODEJ,xNode];**
YNODEJ	**= [YNODEJ,yNode];**
	end

Recall that "arcNumber" is the total number of approximating arcs (M), and "arcNDel(in)" is the number of partitions of the arc number "in" on AFE; the calculations in the internal cycle correspond to Formula (14).

Linear matrices of the coordinates of collocation points (**XCOLLJ** and **YCOLLJ**—Formula (16)) and complex values of the normals (**UNITNORM**—Formula (15)) are formed in a completely analogous way.

To ensure the flexibility (adaptability) of integral operators' computing to different classes of physical phenomena, in addition to the specified arrays we also enter the matrices **RCURVE** and **DELARC** to the database of the geometrical platform, which contain the values of the curvature radius and the length of the elementary arc (AFE) for each point according to numbering along the contour.

Thus, the geometric platform includes seven linear matrices:
(**XNODEJ,YNODEJ**), (**XCOLLJ,YCOLLJ**), **UNITNORM, RCURVE, DELARC.**

The size of the matrices corresponds to the total number of node points on the contour N.

8. Pattern of Procedure for the Formation of the Database for Solving Integral Equations

The last stage of forming of the database for the solution of integral equations is computing a square matrix of SLAE coefficients, by which the IEs are replaced. It is most convenient to carry out this stage of preparation of the simulation of complex physical phenomena in CECT by a separate procedure, which in the MATLAB environment is made out in the form of the user function. Its listing is given in Table 4 below.

Table 4. Computation of a square matrix of system of linear algebraic equations (SLAE) coefficients.

Variables	MATLAB Code (Listing 4)
function [Kernel]	= [];
	global UNITNORM; global COLLOC;
	global RCURVE;

COSnX	= real(UNITNORM(kP));
COSnY	= imag(UNITNORM(kP));
delX	= COLLOC(1,kP) − COLLOC(1,kM);
delY	= COLLOC(2,kP) − COLLOC(2,kM);
RadPM	= sqrt(delX^2 + delY^2);
	if RadPM == 0
Kernel	= −1/(2 × RCURVE(kP));
	else
COSrX	= delX/RadPM;
COSrY	= delY/RadPM;
COSpsi	= COSnX × COSrX + COSnY × COSrY;
Kernel	= COSpsi/RadPM;
	end

This function has two input arguments: the numbers *kP* and *kM* of points *P* and *M* on the contour, and one output argument "Kernel", which transmits the calculated kernel value for these points to the main program. The filling of the matrix of SLAE coefficients of *N* × *N* size is performed by calculating the function "yadro" in a double cycle by the numbers of points *P* and *M*.

9. Examples and Estimates of Efficiency

Testing of the developed procedures for the formation of the geometrical platform and corresponding databases is done on a control contour in the form of a circumference with radius R = 1, divided into six even arcs, which is presented in Table 1. The number of arc partitions was varied from 5 to 50. Graphically, the results of the calculation are shown in Figure 4a (i.e., the distribution of nodal and collocation points (120)) and 4b (i.e., the distribution of normal vectors). In all variants of partitioning, the sum of AFEs was equal to circumference length (2π) with accuracy not less than 10 characters.

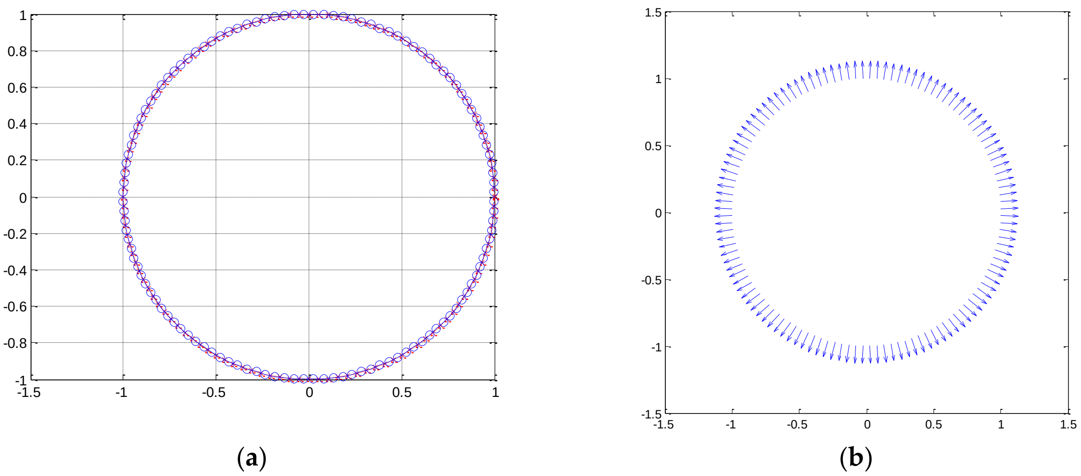
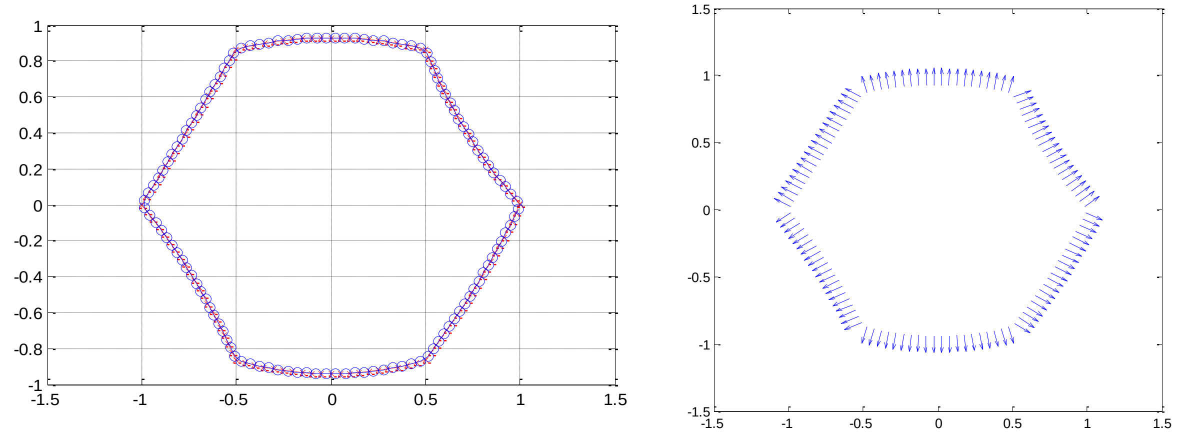


Figure 4. Distribution of points and normals on a circumference: (**a**) distribution of nodal and collocation points; (**b**) distribution of normal vectors.

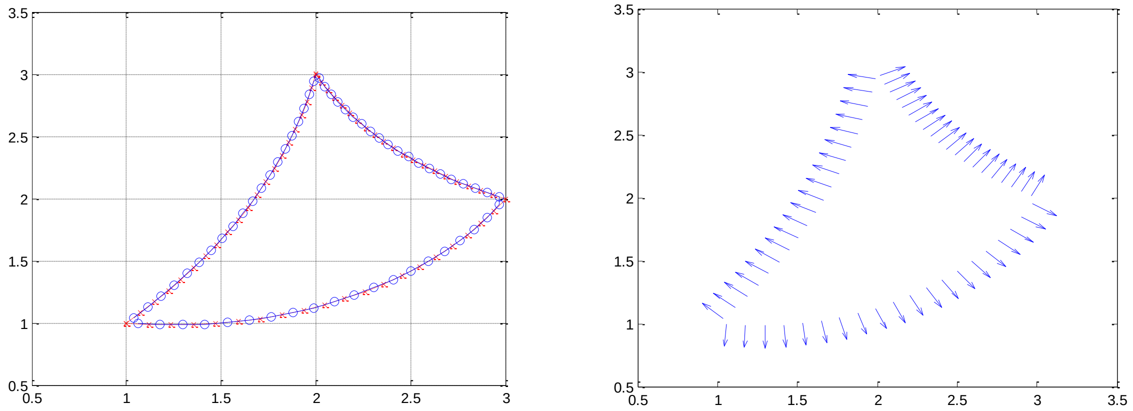
The functionality of the submitted geometrical platform was also evaluated on more difficult configurations: a hexagonal figure with arches of different curvature and camber (Figure 5), as well as a trigonous figure (Figure 6). These results indicate full suitability of the developed tool for the solution of tasks with a multilink boundary configuration.

For the circumference represented by six equal arcs, all components of the geometrical platform specified in the text were calculated, including the coordinates of the points, the directing cosines, and the arc finite elements (AFEs). The figures show the *absolute* (with the 0th error) coincidence of the normals to the arcs with the directions of the radii of arcs' curvature for all contours.



(**a**) (**b**)

Figure 5. Distribution of points and normals on a hexagonal contour: (**a**) distribution of nodal and collocation points on a contour; (**b**) distribution of normal vectors on a contour.



(**a**) (**b**)

Figure 6. Distribution of points and normals on a trigonous contour: (**a**) distribution of nodal and collocation points on a contour, (**b**) distribution of normal vectors on a contour.

As another criterion for the quality of approximation of different figures by means of the geometrical platform, authors accepted the extent of coincidence of the sum of all AFEs with the theoretically exact length of a contour. In particular, for a circumference with different amounts of AFEs (from 30 to 300) even at the roughest discretization (30 AFEs), the values coincided up to the tenth digit inclusively. Table 5 demonstrates the same results for other forms of contour. In fact, this testing was carried out for a dozen different configurations, close to the outlines of real objects, but are not presented here for the sake of brevity. In all cases, the accuracy of reproducing objects' geometrical parameters was more than sufficient.

Table 5. Comparison of calculated contour length with theoretical value.

		Length of Contour		
Kind of contour		Figure 4	Figure 5	Figure 6
Theoretical value		6.28318530718	6.12463518473	6.08452492271
Calculated with number of arcs partitions (arc finite element—AFE)	30–300	6.28318530716	6.12463518473	6.08452492271

The functional validity of the geometrical platform (GP) for execution of the main task—calculating the kernel of integral operators—was tested on a number of contours, ranging from a circle to curved acute-angled triangles. In particular, we obtained data characterizing the accuracy of kernel value calculation and the duration of computation for a circle, with respect to which the exact theoretical value is known: K_2 *(P, M)* = 1/2R_c. Accordingly, for R_c = 1 the kernel K_2(*P*, *M*) = 0.5. The results of calculating kernel values, accuracy, and duration of computing for different numbers of AFEs are presented in Table 6.

Table 6. Kernel values and duration of computing.

	Theoretical	**Calculated with Number of Arcs Partitions (AFE)**				
		30	60	120	180	300
Value	0.5	0.502754	0.500686	0.500171	0.500076	0.500027
Error (%)		5.48	0.137	0.034	0.015	0.005
Duration (s)		0.05	0.21	0.85	1.89	5.22

Now, it is appropriate to evaluate the general options of the obtained database and to compare them with those that characterize the base for solving similar problems by means of differential equations (DEs) using FEM. As a reference point for comparison, the parameters of test contour may be taken from six even arcs, which is presented in *Listing 1*—a circle of radius R = 1, broken down into 360 AFEs (through 1°). Then, the number of sought-for values of variables at the collocation points will be N = 360, and the SLAE matrix will have 360 × 360 = 129,600 elements.

To confirm and illustrate the full functional validity of GP processing for kernel calculation, a visual representation of a kernel values array is shown in Figure 7 for two contours (i.e., Figures 5 and 6).

Therefore, the obtained results give the authors all grounds to claim that the submitted geometrical platform is a reliable software tool of procedures that are entirely suitable for solving integral equations of the considered types.

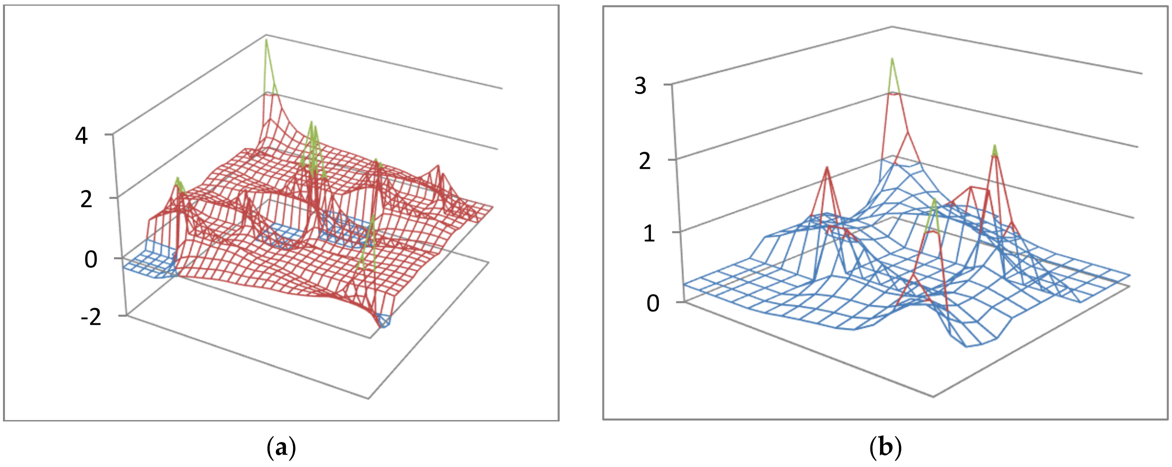
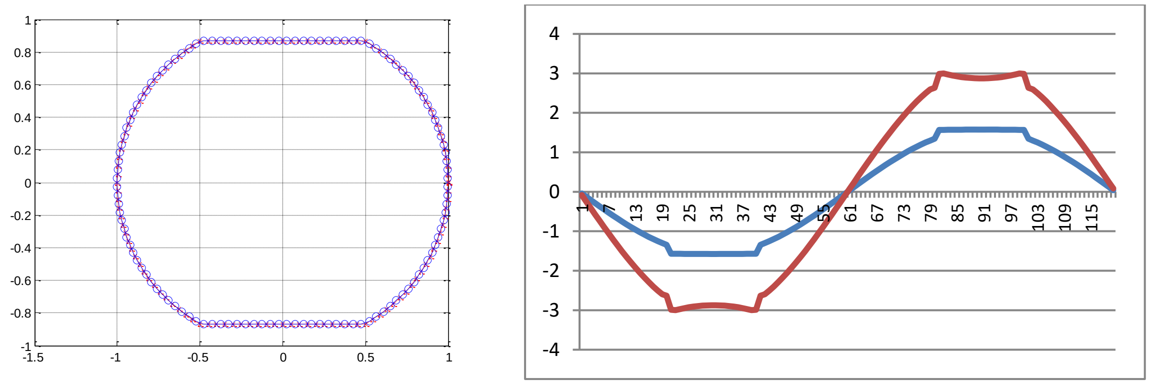


Figure 7. Plots of kernel values array: (**a**) of the contour in Figure 5 with 30 AFEs; (**b**) of the contour in Figure 6 with 15 AFEs.

If for a task with differential equations we consider the computational domain in the form of a square equivalent in area to the mentioned circle with a side $\sqrt{\pi} = 1772$, breaking its perimeter into the same AFEs as the circle, we get 406 contour elements, and the points of collocations where it is necessary to find the variables' values (i.e., the order of the equations system) will be approximately $(406/2)^2 \approx 41{,}000$. The order of the coefficient matrix for such a system will reach more than 1 billion elements. Certainly, such systems are not solved directly, but are reduced to sparse matrices. However, the counting duration and the accumulation of errors (residuals) of their solution increase very significantly. Computing such systems of equations using a computer with a 2–4-core processor can take more than 5–6 h.

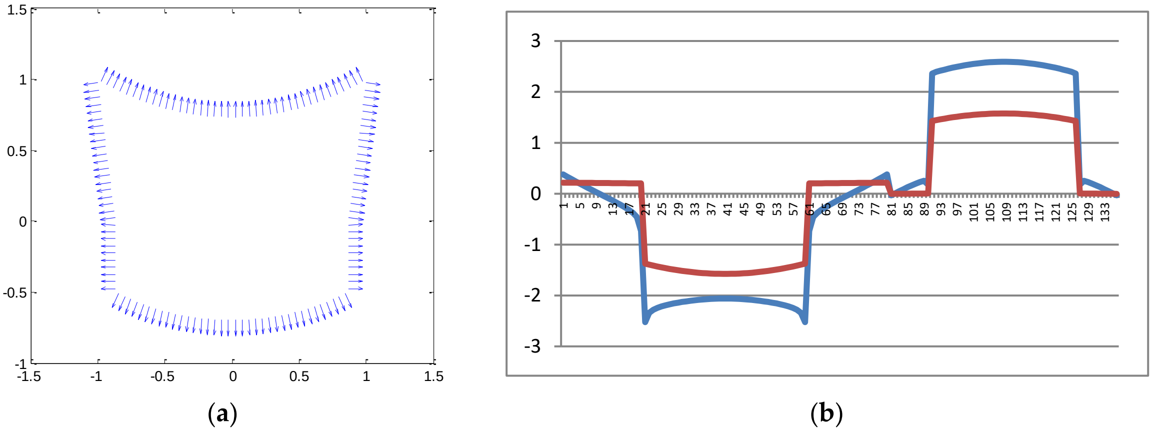
To estimate the temporal characteristics of the presented geometric platform, a timer was built into the program code, which captures the computing time. As a result of the test, the above-mentioned matrix of SLAE coefficients of order 360 was filled for 5 s, and its 300-fold repetition lasted 4 min and 49 s. Based on the quadratic trend of time growth from the order of the matrix (i.e., SLAE), it can be expected that computing a system of order of about 1000 would take 50–60 s.

Moreover, the authors executed an extended test of the entire package of GP program blocks, applying it directly to solve the integral equations on the contours, the outlines of which are close to actual cross-sections of metallurgical ladles. In particular, two examples of field problem solving by IEM on the basis of submitted GP were considered: section of type "keg" (8a) and section of type "jug" (9a), placed in an external uniform exciting field. The solutions of equations were found as the distribution of the normal component of the resulting field along the contours of the conditional ladle section. In Figures 8 and 9, the plots of mentioned field distribution are shown in comparison with the exciting field. Authors must remark that the presented examples are only illustrative, but there are no principal complications to expanding this approach for other CECT tasks.



(**a**) (**b**)

Figure 8. (**a**) Section of type "keg"; (**b**) (-------) exciting field; (-------) resulting field.



(**a**) (**b**)

Figure 9. (**a**) Section of type "jug"; (**b**) (-------) exciting field; (-------) resulting field.

Based on the results, the authors may declare with full responsibility that the submitted geometric platform as an ordered set of program codes is completely ready to be used for solving manifold tasks connected with CECT.

Therefore, this approach will be applied in future research dealing with the practical solution, using the method of integral equations (MIE), of real problems during the CECT of liquid metal in genuine ladles based on the submitted geometric platform for big data computing.

10. Conclusions

The management of casting quality in modern foundry production has paramount importance for which, in particular, the electric current treatment of liquid metal is used. In this process, the interaction of electric current with the melt is accompanied by the emergence of a complex of interrelated physical fields of different nature—electromagnetic, thermal, hydrodynamic, wave and mechanical fields, which eventually predetermine the quality of castings. However, the totality of physical phenomena in this process is largely unavailable for direct observation and measurement.

Therefore, a key role in the choice of the most appropriate treatment parameters for the directed quality control of castings is played by the mathematical modeling and simulation of specified complexes of physical fields—primarily electromagnetic. The theoretical basis for studying the interaction with the medium (liquid and solidifying metal) of the listed fields are the known differential equations in various interpretations. At present, traditional approaches to their solution in the context of CECT problems are based mainly on use of the finite element method and its modifications.

However, the complicated configuration of computational areas and the uncertainty of boundary conditions in boundary value problems often lead to the fact that a very significant degree of discretization of the DE solution area is required in order to ensure the reliability of results. The number of FEs and, accordingly, the number of sought-for variable values is a very large, and may be more than one hundred thousand. At the same time, the database for preparing and executing convergent decision-making procedures according to FEM must have at least a quadratic trend relative to dimensionality of the problem (i.e., it will exceed 1 billion components). In another case, the algorithmic optimization of the database dimension with the use of disperse matrices causes an increase in the volume of computational operations and in the duration of counting in geometrical progression. A standard computer with a 4-core processor spends 5–6 h to solve such tasks.

The main contribution of this work is the development of an alternative methodological approach to formulating problems of mathematical modeling and complex physical phenomenon simulation during the CECT of liquid metals. The approach developed by the authors is guided by the use of integral formulations of fundamental solutions of field theory problems in the form of Green’s formulae and connected relationships, which are generalized in the method of integral equations (MIE).

The solution of tasks in this case is built on the boundary set, and as a result the spatial dimensionality of the original (primary) problem is reduced by at least one order of magnitude and the number of sought unknowns is reduced by almost by two orders.

For transition of the task formulation from DE to IE and the equivalent SLAE, as well as the success of their solution, the procedure of rationally representing the geometric parameters of computational area boundaries—accounting for the need of their explicit use in the algorithmic structure of the mathematical operations of differentiation and integration on curvilinear surfaces as well as the calculation of two-point operators in the integral equations—has great methodological importance.

In other words, in these circumstances, necessity appeared to form an adaptive and concise geometric platform for computing big databases in problems of studying complex physical phenomena during the CECT of liquid metals. The authors obtained the fundamental results in solving this problem by: (a) developing a mathematical model for representing intricate configurations of computational areas by using fairly simple procedures, (b) developing a scheme for the organization of computational processes, and (c) implementing the geometrical platform in the form of program code that may provide big-database computing for the whole variety of problems arising from CECT tasks.

Finally, the computational efficiency of the created platform can be characterized by the following comparison: for the boundary contour representing a ladle section with melt shown in Figure 2, the order of the equations system (i.e., the number of unknown values) will be (depending on the required accuracy) 500–600, compared to 100,000 in the solution by FE method. Naturally, the computing time is reduced to several minutes instead of 5 h.

The functionality of the proposed geometrical platform was evaluated on complicated configurations: a hexagonal figure with arches of different curvature and camber, and a trigonous configuration. These results show the full suitability of the developed tool for the solution of tasks with a multilink configuration of boundaries.

Future research will deal with the use of MIE for the practical solution of real problems during the CECT of liquid metal in genuine ladles based on the submitted geometric platform for big data computing.

Author Contributions: Conceptualization, mathematical description of the geometrical platform of big databases computing, methodology, and software, Y.Z.; formal analysis, data curation, and binding of the obtained numerical procedures to the modelling of physical fields in foundry, A.I.; structuring of the article, supervision, and validation of the work, Y.K. All authors contributed equally to interpreting the results and to writing this article.

Funding: This research received no external funding.

Acknowledgments: Authors cordially thank the Institute of Pulse Processes and Technologies, National Academy of Science of Ukraine for the support of this research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Osta-Omar, S.M.; Micallef, C. Determination of Concentration of the Aqueous Lithium–Bromide Solution in a Vapour Absorption Refrigeration System by Measurement of Electrical Conductivity and Temperature. *Data* **2017**, *2*, 6, doi:10.3390/data2010006.
2. Gerasin, O.; Zaporozhets, Y.; Kondratenko, Y. Models of Magnetic Driver Inter-action with Ferromagnetic Surface and Geometric Data Computing for Clamping Force Localization Patches. In Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP, Lviv, Ukraine, 21–25 August 2018; pp. 44–48.
3. Zaporozhets, Y.M.; Kondratenko, Y.P.; Kondratenko, V.Y. Mathematical Model of Magnetic Field Penetration for Applied Tasks of Electromagnetic Driver and Ferromagnetic Layer Interaction. In *Applied Mathematics and Computational Intelligence. FIM 2015. Advances in Intelligent Systems and Computing*; Gil-Lafuente, A., Merigó, J., Dass, B., Verma, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; Volume 730, pp. 40–53, doi:10.1007/978-3-319-75792-6_4.
4. Kondratenko, Y.; Zaporozhets, Y.; Rudolph, J.; Gerasin, O.; Topalov, A.; Kozlov, O. Modeling of clamping magnets interaction with ferromagnetic surface for wheel mobile robots. *Int. J. Comput*. **2018**, *17*, 33–46.
5. Kondratenko, Y.; Zaporozhets, Y.; Rudolph, J.; Gerasin, O.; Topalov, A.; Kozlov, O. Features of clamping electromagnets using in wheel mobile robots and modeling of their interaction with ferromagnetic plate. In Proceedings of the 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Bucharest, Romania, 21–23 September 2017; Volume 1, pp. 453–458, doi:10.1109/IDAACS.2017.8095122.
6. Kondratenko, Y.P.; Rudolph, J.; Kozlov, O.V.; Zaporozhets, Y.M.; Gerasin, O.S. Neuro-fuzzy observers of clamping force for magnetically operated movers of mobile robots. *Tech. Electrodyn*. **2017**, *5*, 53–61. (In Ukrainian)
7. Kondratenko, Y.P.; Kozlov, O.V.; Gerasin, O.S.; Zaporozhets, Y.M. Synthesis and research of neuro-fuzzy observer of clamping force for mobile robot automatic control system. In Proceedings of the 2016 IEEE First International Conference on Data Stream Mining & Processing, DSMP, Lviv, Ukraine, 23–27 August 2016; pp. 90–95, doi:10.1109/DSMP.2016.7583514.
8. Zaporozhets, Y.M.; Kondratenko, Y.P.; Shyshkin, O.S. Mathematical model of slip displacement sensor with registration of transversal constituents of magnetic field of sensing element. *Tech. Electrodyn*. **2012**, *4*, 67–72. (In Ukrainian)
9. Tsurkin, V.N.; Sinchuk, A.V.; Ivanov; A.V. Electric current treatment of liquid and crystallizing alloys in casting technologies. *Surf. Eng. Appl. Electrochem*. **2011**, *46*, 456–464.
10. Tsurkin, V.N. Cast metal quality management concepts. *Met. Cast. Ukr*. **2008**, *9*, 25–28. (In Russian)
11. Tsurkin, V.N. The principles of a systematic approach to the choice of furnace melt treatment method. *Met. Cast. Ukr*. **2009**, *7–8*, 12–16. (In Russian)

12. Ban, C.; Han, Y.; Ba, Q.; Cui, J. Influence of pulse electric current on solidification structure of Al-Sn alloy. *Electromagn. Process. Mater*. **2007**, *8*, 34–37.
13. Zhang, Y.; Song, C.; Zhu, L.; Zheng, H.; Zhong, H.; Han, Q.; Zhai, Q. Influence of Electric-Current Pulse Treatment on the Formation of Regular Eutectic Morphology in an Al-Si Eutectic Alloy. *Metall. Mater. Trans. B* **2011**, *42*, 604–611.
14. Xu, G.; Zheng, J.; Liu, Y.; Cui, J. Effect of electric current on the cast micro-structure of Al-Si alloy. *China Foundry* **2005**, *2*, 171–175.
15. Tsurkin, V.N.; Ivanov, A.V.; Cherepovskii, S.S.; Vasyanovich, N.A. Comparative analysis of functional possibilities of methods of pulse treatment of a melt. *Surf. Eng. Appl. Electrochem*. **2016**, *52*, 56–61.
16. Nakada, M.; Shiohara, Y.; Flemings, M.C. Modification of solidification structures by pulse electric discharging. *ISIJ Int*. **1990**, *30*, 27–33.
17. Eskin, D.G.; Mi, J. *Solidification Processing of Metallic Alloys Under External Fields. Springer Series in Materials Science*; Springer: Berlin/Heidelberg, Germany, 2019; p. 328.
18. Hoseinie, S.; Al-Chalabi, H.; Ghodrati, B. Comparison between Simulation and Analytical Methods in Reliability Data Analysis: A Case Study on Face Drilling Rigs. *Data* **2018**, *3*, 12, doi:10.3390/data3020012.
19. Ivanov, A.V.; Sinchuk, A.V.; Bogoslavskaya A.S. A Study of the Electromagnetic and Hydrodynamic Processes in a Liquid-Metal Conductor Exposed to Current Pulses. *Surf. Eng. Appl. Electrochem*. **2011**, *47*, 28–34.
20. Ivanov, A.V.; Sinchuk, A.V.; Ruban, A.S. Effect of the Technological Parameters of the Melt Treatment by a Electric Pulse Current on the Mixing Process. *Surf. Eng. Appl. Electrochem*. **2012**, *48*, 180–186.
21. Tsurkin, V.N.; Ivanov, A.V. Peculiarities of Redistribution of Electric and Thermal Fields at the Interface When Passing the Electric Current through the Melt. *Surf. Eng. Appl. Electrochem*. **2018**, *54*, 577–584.
22. Ivanov, A.V.; Tsurkin, V.N. Peculiarities of Distribution of Electromagnetic and Hydrodynamic Fields for Conductive Electric Current Treatment of Melts in Different Modes. *Surf. Eng. Appl. Electrochem*. **2018**, *55*, 53–64.
23. Räbiger, D.; Zhang, Y.; Galindo, V.; Franke, S.; Willers, B.; Eckert, S. Experimental study on directional solidification of Al-Si alloysunder the influence of electric currents. *IOP Conf. Ser. Mater. Sci. Eng*. **2016**, *143*, 012021.
24. Demirchian, K.S.; Chechurin, V.L. *Computer Calculations of Electromagnetic Fields*; Vyisshayashkola: Moscow, Russia, 1986. (In Russian)
25. Tihonov, A.N.; Samarskiy, A.A. *Equations of Mathematical Physics, Textbook for High Schools*, 5th ed.; Nauka: Moscow, Russia, 1977. (In Russian).
26. Tikhonov, A.N.; *Arsenin, V.Y. Methods of Solving the Incorrect Problems;* Nauka: Moscow, Russia, 1979. (In Russian)
27. Malcolm, M.; Forsite, J.; Mouler, C. *Mathematical Calculations Machine Methods*; Mir: Moscow, Russia, 1980. (In Russian)
28. Lawson, C.L.; Hanson, R.J. *Solving Least Squares Problems;* Nauka: Moscow, Russia, 1986. (In Russian)
29. Wang, D. Adjustable Robust Singular Value Decomposition: Design, Analysis and Application to Finance. *Data* **2017**, *2*, 29, doi:10.3390/data2030029.
30. Lisin, D.A.; Maksimenko-Sheiko, K.V.; Sheiko, T.I. Mathematical modeling of the automobile bodies with the help of R-functions. *J. Mech. Eng*. **2013**, *13*, 51–60.
31. *Lisnyak, A.A.; Gomenyk, S.I. Application of R-functions for geometric modeling of objects of complex shape. Radio Electron. Comput. Sci. Control* **2009**, *2*, 76–81.
32. Zaporozhets, Y.M.; Chudaykin, I.I.; Amirshadov, E.G. Combination-generation algorithm of triangulation of the ship’s surface in deviation tasks. In Proceedings of the Jubilee Scientific and Technical Conference Dedicated to the 50th Anniversary of the Shipprotection Service, St. Petersburg, Russia, 1994. (In Russian)
33. Maksymenko-Sheyko, K.V.; Sheyko, T.I. R-functions in mathematical modeling of geometrical objects in 3D under the information in 2D. *Visnyk Zaporizhzhya Natl. Univ. Phys. Math. Sci*. **2010**, *1*, 98–104.
34. Anufriev, I.E.; Smirnov, L.B.; Smirnova, E.N. MATLAB 7; BHV-Petersburg: St. Petersburg, Russia, 2005. (In Russian).