MATHEMATICS

M. A. ALEKSIDZE

1958

SovietRxiv

Abstract

Full Text

MATHEMATICS

M. A. ALEKSIDZE

ON AN ALGORITHM FOR AUTOMATING THE NUMERICAL SOLUTION OF THE PLANE DIRICHLET PROBLEM FOR THE LAPLACE EQUATION

(Presented by Academician S. L. Sobolev, 27 XI 1957)

1. For the successful operation of general-purpose electronic digital machines it is necessary to have universal programs (UP) for an entire class of problems for which general numerical methods of solution have been well developed. The advantage of a UP over a programming program is the complete automation of both stages of programming and a substantial reduction in the volume of initial information. The initial information for a UP for solving the first boundary-value problem will be the encoded equations for the boundary of the domain and the function prescribed on the boundary. In the algorithm described here it is assumed that the equations are given in parametric form and that, as the parameter increases, the domain is traversed on the right.

The extraordinary bulkiness and monotony of the final computations in solving boundary-value problems by the grid method makes the use of electronic digital machines highly effective. Complete automation of the solution of a boundary-value problem also presupposes the automation of the following processes: a) replacement of the boundary of the domain by a polygonal line consisting of sides and diagonals of the grid; b) transfer of the grid domain into the internal memory device (IMD) of the machine–linearization of the domain; c) finding all boundary points and the information necessary for compiling at these points irregular difference equations; d) the direct computation and output of the final results in a form convenient for decoding.

2. Let us consider a finite-difference analogue of the Dirichlet problem for the Laplace equation

$$\Delta_h u = 0 \quad \text{in } G_h, \qquad u|_{\Gamma_h} = \psi,$$

where Δ_h is one of the grid approximations of the Laplace operator $\Delta_h u(A) = \sum_B C(A,B)u(B)$. We shall call those points B for which $C(A,B) \neq 0$ neighboring to A in the sense of the operator Δ_h. The nodal points of the grid lying

in the domain G_h or on the contour Γ_h will be called computational points. A computational point will be a boundary point if the distance from it to the boundary of the domain along the coordinate lines is less than the mesh size. Otherwise the computational point will be an interior point.

For purposes of automation, the division of computational points into interior and boundary points is insufficient. The number of types of points depends on the kind of boundary-value problem, on the geometry of the domain, and on the formulas approximately replacing the differential operators. In the case of the simplest approximation of the Laplace operator and of correcting the values of the function at boundary points by Collatz' s method, the following additional subdivision of the set of boundary points is necessary. Double boundary points are points lying simultaneously both on the contour and on the polygonal line approximating the contour. The remaining boundary points, depending on the direction in which for them the support-

value are divided into groups: boundary-right, boundary-left, boundary-upper, and boundary-lower. We shall call **reference** values the values of the function at grid points used to correct the values of the function at boundary points.

With respect to the domains under consideration, let us assume that they are bounded by a finite number of curves given in parametric form and satisfying Lipschitz conditions with exponent 1. The Lipschitz constants on the separate pieces of the boundary must be given in the initial information. The domains may have holes, narrow necks, and may form a disconnected set of grid points.

To linearize a domain with a fixed column height, we find the smallest rectangle containing the given domain in its entirety, and then transfer the columns of this rectangle into the machine' s external storage device. In this case neighboring points will be located in strictly definite cells.

For marking the memory array one may use either local marking—storing in each cell an indication of the type of point to which the given cell belongs—or integral marking—compiling and storing separately information on the alternation of all types of points in each column. The choice of local or integral marking depends on the geometry of the domain, on the volume of the machine' s external storage device, and on the number of digits in a memory cell. In the case of an arbitrary domain, local marking is more expedient for the BESM.

We shall call the numbers of the straight lines forming a node its **integer coordinates**. We shall likewise call the integer parts

$$\left[\frac{y-(y_0-512h)}{h}\right], \quad \left[\frac{x-(x_0-512h)}{h}\right],$$

where x, y are the coordinates of the given point; x_0, y_0 are the coordinates of the initial point of traversal of the contour. It is not difficult to verify that the

integer coordinates of a boundary point and of the lower corner of the square in which the given point lies coincide.

We shall say that a **boundary point has been found** if, in addition to the integer coordinates, all the information necessary for its processing has been obtained. To find boundary points, one may move along the straight lines forming the grid domain and find intersections with the boundary of the domain, or else move along the boundary and find intersections with the straight lines. Because of the inconvenience of specifying the initial information in the first case, we chose the second method.

Let us divide each square of the grid into 9 parts (Fig. 1) and form the matchings E_1, E_2 and comparisons E_3, E_4, E_5, E_6:

E_1	$,<$	$\langle [x]_{n+1} \rangle$	$\langle [x]_n \rangle$
E_2	$,<$	$\langle [y]_{n+1} \rangle$	$\langle [y]_n \rangle$
E_3	$<$	$\langle \mu \rangle$	$\langle \varepsilon \rangle$
E_4	$<$	$\langle \mu_1 \rangle$	$\langle \varepsilon \rangle$
E_5	$<$	$\langle \nu \rangle$	$\langle \varepsilon \rangle$
E_6	$<$	$\langle \nu_1 \rangle$	$\langle \varepsilon \rangle$

where $\langle a \rangle$ denotes the address of the cell in which a is stored; $[x]_{n+1}, [y]_{n+1}$ and $[x]_n, [y]_n$ are the integer coordinates of the points $f(s)$ and $f(s + \Delta s)$; Δs satisfies the inequalities

$$|x(s) - x(s + \Delta s)| < h, \quad |y(s) - y(s + \Delta s)| < h. \tag{1}$$

We shall regard the matchings and comparisons given above as logical propositions. A matching (comparison) is true ($E_i = 1$) if, after it

the next command in number is executed; otherwise the merge (comparison) is false ($E_i = 0$).

Let us consider all possible cases generated by the binary variables E_i. The 64 possible cases can be grouped as follows: 55, 59, 61, 62, 63—after these cases the next step along the contour is made; 7, 11, 13, 14, 15, 29, 30, 31, 39, 43, 47—a reverse pass along the contour is made with step $\Delta s/2$; 5, 6, 9, 10, 21, 22, 23, 25, 26, 27, 37, 38, 41, 42, 45, 46, 53, 54, 57, 58—in accordance with the right-hand traversal of the region, the integer coordinates of boundary points and all the information necessary for correcting the value of the function at them are stored. The remaining cases are impossible because the inequalities $\mu < \varepsilon$, $\mu_1 < \varepsilon$, as well as $\nu < \varepsilon$, $\nu_1 < \varepsilon$, cannot hold simultaneously ($\varepsilon \ll h$).

Fig. 1

Consider, for example, case 46 (101110). The falsity of the merge E_2 indicates that the boundary exits either upward or downward, while the falsity of E_6

Fig. 1

Figure 1: Fig. 1

Fig. 2

Figure 2: Fig. 2

indicates that we are in part V. Because of (1), this is possible only when the boundary exits downward. In accordance with the right-hand traversal of the region, node A is stored as a boundary-left point, along with ν and ψ.

Simultaneously with the intersections, the maximum and minimum integer coordinates are stored; these make it possible to linearize

Fig. 2

the region with a constant column height, and also $\psi_{\max}$, whose order is taken as the scale factor. The maximum principle makes it possible to carry out the computations with a fixed decimal point, while storing signs in the orders. In the lower 10 digits of a boundary point there will be stored the address of the cell containing the encoded information for correcting the values of the function at this point.

3. After all points of intersection have been found, they are checked. In Fig. 2 all special cases for a boundary-right point are given. Case a: when moving from 4 to 1, the intersections remain untouched; otherwise they are destroyed; to determine the direction, the condition $\delta_2 + \delta_3 > h$ is checked (δ_i is the distance from point i to the boundary). Case b: the value of the function at point B is interpolated and the point is marked as a double boundary point; the intersections are destroyed. Cases v, g: the intersection farther from the boundary is destroyed. Case d: intersection 1 is destroyed.

Then the boundary values are arranged. The true address N for the boundary point $([x], [y])$ will be

$$N = ([x] - [x]_{\min})([y]_{\max} - [y]_{\min} + 1) + [y] - [y]_{\min}.$$

The initial values for the interior points are computed by the formula

$$\left(\frac{1}{ac} + \frac{1}{bd}\right) u_0 = \frac{u_a}{a(a+c)} + \frac{u_b}{b(b+d)} + \frac{u_c}{c(a+c)} + \frac{u_d}{d(b+d)},$$

where u_0, u_a, u_b, u_c, u_d are the values of the function at the points $(0;0)$, $(a;0)$, $(b;0)$, $(0;-c)$, $(0;-d)$.

In the case of the Liebmann iteration, a boundary (non-double) point requires 27 commands, and an interior point 24. The machine itself determines, by Gershgorin' s well-known formula, the moment at which to terminate the iteration, using the residuals, in order to ensure the required accuracy of the solution of the difference equations.

In conclusion, I express my gratitude to E. A. Volkov for his guidance in compiling the UP.

Institute of Precision Mechanics and Computer Engineering
of the Academy of Sciences of the USSR

A. M. Razmadze Mathematical Institute
of the Academy of Sciences of the Georgian SSR

Received
25 XI 1957

Note: Figure translations are in progress. See original paper for figures.