

A Method for Automating the Solution of Boundary Value Problems on High-Speed Computers

V. K. Saul'ev

1962

SovietRxiv

View the original and related papers at <https://sovietrxiv.org/items/ru-196201.99010>

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.

Abstract

Full Text

Reports of the Academy of Sciences of the USSR
1962. Volume 144, No. 3

MATHEMATICS

V. K. SAUL' EV

ON A METHOD FOR AUTOMATING THE SOLUTION OF BOUNDARY-VALUE PROBLEMS ON HIGH-SPEED COMPUTING MACHINES

(Presented by Academician A. A. Dorodnitsyn, 8 I 1962)

It is known that the preparation of programs for solving boundary-value problems on high-speed computing machines is a labor-intensive process. It is therefore advisable to construct programs not for particular problems (in particular, particular domains), but for a more or less broad class of problems (in particular, domains). Obviously, such "universal" programs are logically more complicated and require, to one degree or another, more machine time. However, this additional effort of the programmer, expended on "universality," is as a rule repaid many times over. The increase in machine time when universal programs are used, as compared with nonuniversal ones (arising from additional logical, and sometimes also idle arithmetic, operations), is not significant if one bears in mind the enormous speeds of modern electronic computing machines.

We shall now describe one principle for constructing universal programs (in the sense of their applicability to a broad class of domains) for solving certain boundary-value problems containing an elliptic operator on high-speed computing machines.

We shall explain the basic idea of the method on the simplest boundary-value problem

$$\Delta U \equiv \frac{1}{x^\delta} \frac{\partial}{\partial x} \left(x^\delta \frac{\partial U}{\partial x} \right) + \frac{\partial^2 U}{\partial y^2} = -f \quad \text{in } G, \quad (1)$$

$$U = 0 \quad \text{on } S, \quad (2)$$

where G is a certain finite domain bounded by a piecewise-smooth contour S , $f = \text{const} > 0$, $\delta = 0$ or 1 .

First of all, we approximate the boundary S by a broken line consisting of straight-line segments parallel to the coordinate axes; here we assume that the boundary S is such that the number of these segments is not too large. If, however, the boundary S , as often happens in practice, consists of straight-line segments parallel to the coordinate axes (for example, in cases where G is the cross-section of an I-beam or an angle), then this stage of approximating the boundary S (and correspondingly the domain G) is omitted. Denote the stepwise approximating boundary and the domain enclosed within it respectively by S_1 and G_1 . It is not difficult to show that as $S_1 \rightarrow S$ ($G_1 \rightarrow G$) the solution of the equation $\Delta U = -f$ for the domain G_1 with zero values on S_1 tends to the solution of the same equation for the domain G with zero values on S .

Now place the domain G_1 in the smallest rectangle R , and extend the straight-line segments of which the boundary S_1 is composed by straight lines up to their intersections with the boundary S_0 of the rectangle R . As a result, the rectangle R is represented in a regular manner as the sum of MN rectangles $\Pi^{(pq)}$ ($p = 1, 2, \dots, M$; $q = 1, 2, \dots, N$), where M (respectively N) is the number of rectangles in the direction of the x -axis (respectively the y -axis). We shall call the set of $M - 1$ vertical segments with abscissae $a_1, a_2, \dots, a_{M-1}$ and $N - 1$ horizontal segments with ordina-

the numbers $b_1, b_2, \dots, b_{N-1}$, by means of which the above-mentioned regular representation of the rectangle R as the sum of MN rectangles is carried out; let the internal boundary be denoted by $\bar{\Gamma}$. The boundary S_0 of the rectangle R , consisting of the segments $x = 0$, $x = a_M$ and $y = 0$, $y = b_N$ (it is assumed that the origin is placed at the lower left corner of the rectangle R), will be called the external boundary. Finally, the set of points obtained as a result of excluding from $\bar{\Gamma}$ the $(M - 1)(N - 1)$ "corner" points of intersection of the vertical $x = a_i$ ($i = 1, 2, \dots, M - 1$) and horizontal $y = b_i$ ($i = 1, 2, \dots, N - 1$) segments (at which the normal to $\bar{\Gamma}$ does not exist) will be denoted by Γ .

Instead of solving problem (1), (2) for the nonrectangular domain G , we shall solve the following problem for the rectangular domain R :

$$\Delta V = -f \quad \text{for } \Pi^{(pq)} \subset G_1; \quad (3)$$

$$\Delta V = 0 \quad \text{for } \Pi^{(pq)} \subset R \setminus G_1; \quad (4)$$

$$V = 0 \quad \text{on } S_0; \quad (5)$$

$$[V]_{p,q} = [V]_{p',q'} \quad (p = p' \text{ and } q \neq q' \text{ or } p \neq p' \text{ and } q = q') \quad \text{on } \bar{\Gamma}; \quad (6)$$

$$\left[\frac{\partial V}{\partial \nu} \right]_{p,q} = \lambda \left[\frac{\partial V}{\partial \nu} \right]_{p',q'} \quad (p = p' \text{ and } q \neq q' \text{ or } p \neq p' \text{ and } q = q') \quad \text{on } \Gamma, \quad (7)$$

where the symbol $[]_{p,q}$ (respectively $[]_{p',q'}$) means that the quantity enclosed in square brackets is taken on the internal boundary from the side of $\Pi^{(pq)}$ (respectively $\Pi^{(p'q')}$); the quantity λ differs from 1 only in the case when $\Pi^{(pq)} \subset G_1$, and $\Pi^{(p'q')} \subset R \setminus G_1$; ν is the direction of the normal to Γ .

Problem (3)–(7), in contrast to problem (1), (2), is cyclic in the programming sense and therefore admits convenient automation in the compilation of a universal program. For example, if one assigns the sign $\pi^{(pq)}$ to the rectangle $\Pi^{(pq)}$ ($p = 1, 2, \dots, M$; $q = 1, 2, \dots, N$), then it is sufficient to specify the initial information in the form of the following sequence of $MN + 2N + M + 2$ codes:

$$\begin{aligned} &\pi^{(11)}, \pi^{(21)}, \dots, \pi^{(M1)}, 0, \pi^{(12)}, \pi^{(22)}, \dots, \pi^{(M2)}, 0, \dots, \\ &\pi^{(1N)}, \pi^{(2N)}, \dots, \pi^{(MN)}, 0; 0, a_1, a_2, \dots, a_M, 0, b_1, b_2, \dots, b_N; f; \lambda, \end{aligned} \quad (8)$$

where the sign $\pi^{(pq)}$ is equal, for example, to 1 or 2, depending on whether the rectangle $\Pi^{(pq)}$ belongs to the domain G_1 or not, and the zeros play a separating role in the decoding by the machine of the initial information (8).

At the same time, one can prove, using methods of functional analysis (embedding theorems), that as $\lambda \rightarrow +\infty$ the solution V of problem (3)–(7) in the domain G_1 converges to the solution W of the problem

$$\Delta W = -f \text{ in } G_1; \quad W = 0 \text{ on } S_1 \quad (9)$$

with speed no less than $C/\sqrt{\lambda}$, where C is a certain constant depending only on the form of the domain G_1 . Moreover, the maximum principle makes it possible to establish easily that for $\lambda > 0$, V converges to W from above, i.e. $V \geq W$.

For $\lambda < 0$, a solution of problem (3)–(7) may fail to exist. However, for large negative values of λ in modulus, in many cases one may expect that the solution of problem (3)–(7) exists and is unique. In these cases, on the basis of the same maximum principle, one can show the validity of the reverse inequality, i.e. $V \leq W$.

Thus, the method of fictitious domains with large positive values of λ and large negative values of λ in modulus makes it possible (in the case of correctness of problem (3)–(7) when negative

values of λ) to enclose the solution W of problem (9) in a fork, i.e., to obtain a two-sided approximation. The convergence of W to U —the solution of the given problem (1)–(2)—has already been discussed.

For the numerical solution of problem (3)–(7), it is most expedient to use the mesh method. In this case, to the sequence of numbers (8) one should also add the sequence of numbers $m_1, m_2, \dots, m_M$; $n_1, n_2, \dots, n_N$, where m_i is the number of parts into which the interval $a_i - a_{i-1}$ is divided ($i = 1, 2, \dots, M$); n_i is the number of parts into which the interval $b_i - b_{i-1}$ is divided ($i = 1, 2, \dots, N$). Then, at the nodes lying on the internal boundary $\bar{\Gamma}$, one can use a five-point mesh equation of the form

$$\begin{aligned} \frac{(h_1 + h_3)(h_2 + h_4)}{4} L_h v_0 \equiv & \frac{h_2 \alpha_1 + h_4 \alpha_4}{2h_1} \left(1 + \frac{\delta h_3}{2x}\right) v_1 + \frac{h_2 \alpha_2 + h_4 \alpha_3}{2h_3} \left(1 - \frac{\delta h_1}{2x}\right) v_3 \\ & + \left[\frac{h_1 \alpha_1 + h_3 \alpha_2}{2h_2} + \frac{\delta h_1 h_3}{4x h_2} (\alpha_1 - \alpha_2) \right] v_2 + \left[\frac{h_3 \alpha_3 + h_1 \alpha_4}{2h_4} + \frac{\delta h_1 h_3}{4x h_4} (\alpha_4 - \alpha_3) \right] v_4 \\ & - s v_0 = \frac{1}{4} (h_1 h_2 \beta_1 + h_2 h_3 \beta_2 + h_3 h_4 \beta_3 + h_4 h_1 \beta_4) \\ & + \frac{\delta h_1 h_3}{8x} [h_2 (\beta_1 - \beta_2) + h_4 (\beta_4 - \beta_3)], \end{aligned} \quad (10)$$

where v_i ($i = 0, 1, \dots, 4$) are the values of the function v , respectively, at the given node $(x, y) \in \bar{\Gamma}$ and at the four neighboring nodes $(x + h_1, y)$, $(x, y + h_2)$, $(x - h_3, y)$, $(x, y - h_4)$; α_i (respectively β_i) is equal to 1 or λ (respectively $-f$ or 0) depending on whether the mesh cell D_i adjacent to the given node (x, y) and whose area is $h_i h_{i+1}$ ($i = 1, 2, 3, 4$; $h_5 = h_1$) belongs to the domain G_1 or not; s is the sum of the coefficients in (10) at v_i ($i = 1, 2, 3, 4$). At nodes lying inside the rectangles $\Pi^{(pq)}$ ($p = 1, 2, \dots, M$; $q = 1, 2, \dots, N$), one can use the usual five-point mesh equation, which is obtained from (10) if one sets $h_1 = h_3$, $h_2 = h_4$, $\alpha_i \equiv 1$, $\beta_i \equiv -f$ or 0.

For solving the system of mesh equations obtained in this way, it is very convenient (the domain of integration is a rectangle!) to apply the fastest-converging iterative method—the method of alternating directions. Let us note that for negative values of λ the iterative process may converge more slowly (or even diverge) than in the case of positive values of λ .

The asymptotic local error of the operator L_h (10) can be estimated by means of the inequality

$$\left| \sum_{i=1}^4 \omega_i \alpha_i (\Delta V)_i - L_h V_0 \right| \leq \frac{h_1 h_3 M_3}{6(h_1 + h_3)(h_2 + h_4)} \left\{ \left(\frac{2}{h_3} + \frac{\delta}{x} \right) [h_2 \alpha_1 (h_1 + h_2 + h_4 \alpha_4 (h_1 + h_4))] \right. \\ \left. + \left(\frac{2}{h_1} - \frac{\delta}{x} \right) [h_2 \alpha_2 (h_2 + h_3) + h_4 \alpha_3 (h_3 + h_4)] \right\} = O(h) \\ (h_1 \leq 2x/\delta),$$

where $(\Delta V)_i$ is the limiting (one-sided) value, from the side of D_i , of ΔV at the given node (x, y) ;

$$\omega_1 = \frac{h_1 h_2}{(h_1 + h_3)(h_2 + h_4)} \left(1 + \frac{\delta h_3}{2x} \right), \quad \omega_2 = \frac{h_2 h_3}{(h_1 + h_3)(h_2 + h_4)} \left(1 - \frac{\delta h_1}{2x} \right),$$

$$\omega_3 = \frac{h_3 h_4}{(h_1 + h_3)(h_2 + h_4)} \left(1 - \frac{\delta h_1}{2x} \right), \quad \omega_4 = \frac{h_1 h_4}{(h_1 + h_3)(h_2 + h_4)} \left(1 + \frac{\delta h_3}{2x} \right),$$

$$\sum_{i=1}^4 \omega_i = 1;$$

M_3 is the maximum in $D_1 \cup D_2 \cup D_3 \cup D_4$ of the moduli of the one-sided derivatives of the function V ; $h = \max_i h_i$.

The method considered above for constructing programs on the basis of a regular representation of the given nonrectangular domain, completed to a rectangle, in the form of a sum of “elementary” rectangular domains (some of which are “fictitious” ($\lambda \gg 1$)), proved—as shown by almost three years’ operation of the program we compiled according to this principle for the M-20 machine—to be quite effective. In fact, this program can be used to compute boundary-value problems considerably more complicated than (1), (2): mixed boundary conditions, systems of equations, equations with an eigenvalue parameter, and equations with discontinuous coefficients.

The method described above for constructing a program carries over directly to parabolic equations, as well as to equations with three spatial variables. Boundary-value problems for the biharmonic equation can also be solved by this method; in this case an appropriate choice of the matching condition on the boundary between the given and fictitious domains ensures, as $\lambda \rightarrow \infty$, convergence of V to W at a rate no less than $O(\lambda^{-1/2})$.

In conclusion we give the simplest example, computed by means of a universal program compiled according to the principle described, which illustrates the

formation of a bracket. Let $f = 2$; $\delta = 0$; G be the rectangle whose vertices are at the points $(0; 0)$, $(2; 0)$, $(2; 1)$, $(0; 1)$; and R the rectangle whose vertices are at the points $(0; 0)$, $(2.25; 0)$, $(2.25; 1)$, $(0; 1)$. Then

$x=y=$	1.00	1.25	1.50	1.75	2.00	2.125	2.25
$\lambda = -50$	0.22741	0.22006	0.19339	0.13074	-0.00325	-0.00153	0
$\lambda = -100$	0.22756	0.22026	0.19376	0.13152	-0.00162	-0.00076	0
Exact solution	0.22774	0.22052	0.19424	0.13239	0	—	—
$\lambda = 100$	0.22785	0.22065	0.19449	0.13305	0.00159	0.00075	0
$\lambda = 50$	0.22799	0.22084	0.19485	0.13380	0.00317	0.00149	0

As was to be expected, for $\lambda = -50$ and $\lambda = -100$ the solution was obtained with a deficit, and for $\lambda = 50$ and $\lambda = 100$, with an excess. Moreover, the larger λ , the narrower the bracket. However, for very large values of λ the bracket may not occur because the error of the mesh method will exceed the difference between the two computations with positive and negative values of λ .

All-Union Institute
of Scientific and Technical Information

Received
6 I 1962

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

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.