KYBERNETIKA — VOLUME 9 (1973), NUMBER 5

Iterative Solving of Partial Differential Equations

Ružena Apalovičová, Jozef Babirád

The paper deals with solving parabolic partial differential equations by the method of Continuous Space - Discret Time (CSDT) on an analog computer.

INTRODUCTION

When using analog computers to various purposes we often face the problem of solving partial differential equations of the type

$$\frac{\partial u(x, y)}{\partial x} = a(y) \frac{\partial^2 u(x, y)}{\partial y^2}, \tag{1}$$

with initial condition

$$u(0, y) = f(y), \quad 0 \leqq y \leqq L,$$

and boundary conditions

$$u(x, 0) = \mu_1(x), \quad x > 0,$$
$$u(x, L) = \mu_2(x).$$

Such equations are most often solved by the use of difference methods, which are based on the approximation of partial derivatives by finite difference expansions. Equations of the parabolic type, such as equation (1), and of a hyperbolic type, are solved by the method of straight lines. One of the variables, say x, is given as independent machine variable t, while the other variable, in this case y, is assumed to be discrete. With respect to the independent machine variable the task has now the character of an initial problem while it represents a boundary problem for the other variable. By using the first approximation, equation (1) obtaing the form

$$\frac{\mathrm{d}u_j(x)}{\mathrm{d}x} \doteq a_j(y)\,\frac{u_{j+1}(x) - 2u_j(x) + u_{j-1}(x)}{(\Delta y)^2}\,, \tag{2}$$

$$j = 1, 2, \ldots, n - 1\,;$$

where the interval $\langle 0, L\rangle$ of variable y is divided into n parts $\Delta y = L/n$. An analog computer can be used for solving a set of $(n - 1)$ ordinary difference-differential equations of the type (2); Fig. 1.

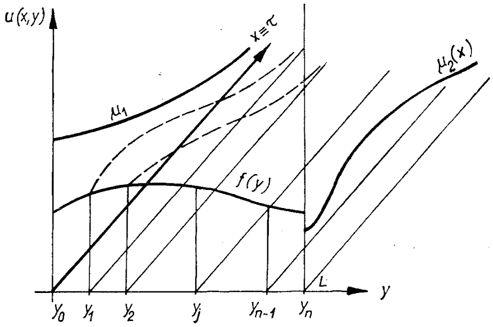

Fig. 1.

The method of straight lines may be used inversely, i.e., when variable y changes continuously and variable x is discrete. Equation (1) then takes the form

$$\frac{u_i(y) - u_{i-1}(y)}{\Delta x} \doteq a_i(y)\,\frac{\mathrm{d}^2 u_i(y)}{\mathrm{d}y^2}\,, \tag{3}$$

$$i = 1, 2, \ldots, m\,.$$

Such an approach requires the knowledge of asymptotic solution of the dependent variable $u(x, y)$ for $x \to \infty$; Fig. 2.

The boundary conditions cannot be fulfilled directly, instead we have to use methods of solving boundary problems of ordinary differential equations. Unlike the previous method, it is necessary to solve equation (3) m-times in succession. However, in technical practice it is often suitable to know the solution in individual sections for $i = 1, 2, \ldots, m$. Besides, this method appears most suitable for solving partial differential equations [6], and therefore we shall solve equation (1) in this manner.

For an m-fold solution of equation (3) we have to accomplish the following partial tasks:

1. Solving the boundary problem of function $u(x_i, y)$ for $y = L$. This task is solved by means automatic iterative method.

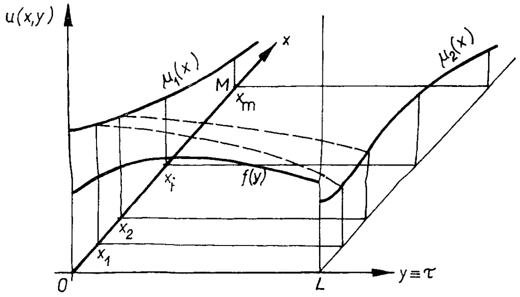

Fig. 2.

2. Storaging function $u_{i-1}(y)$, which is obtained as a solution in step $i - 1$, and needed for the solution of $u_i(y)$ in the i-th step – use of the analog continuous memory.

3. Constructing the sequence of control signals in order that the entire computation, including the plotting of results, may proceed automatically, without any intervention on the part of the operator.

1. SOLVING THE BOUNDARY PROBLEM OF THE FUNCTION $u(x_i, y)$ AT POINT $y = L$

Solving the boundary problem consists in looking for a function, which is the solution of a given partial differential equation inside a given region, and which suits given boundary conditions on the boundary of this region.

In solving parabolic partial differential equations we shall proceed as follows:

$$(1.1) \qquad \frac{\partial u(x, y)}{\partial x} = a \frac{\partial^2 u(x, y)}{\partial y^2};$$

initial condition

$$u(0, y) = f(y), \quad y \in \langle 0, L \rangle;$$

boundary conditions

$$u(x, 0) = \mu_1(x), \quad x > 0,$$

$$u(x, L) = \mu_2(x).$$

Replacing $\partial u/\partial x$ by finite difference expansion we obtain the rearranged equation in the form

$$(1.2) \qquad \frac{\mathrm{d}^2 u_i(y)}{\mathrm{d}y^2} \doteq \frac{1}{a} \frac{u_i(y) - u_{i-1}(y)}{\Delta x}\,, \qquad i = 1, 2, \ldots, m\,; \quad x \in (0, M\rangle\,.$$

For $i = 1$, the value of function $u_{i-1}(y)$ is given, as a matter of fact, by the initial condition $u(0, y) = f(y)$. The first boundary condition $u(x_i, 0) = \mu_1(x_i)$ will be the initial condition in the solution of equation (1.2). It remains to fulfil the second boundary condition $u(x_i, L) = \mu_2(x_i)$ for all values of $i = 1, 2, \ldots, m$. This condition will be fulfilled by a suitable choice of the second initial condition $\mathrm{d}u_i(y)/\mathrm{d}y|_{x=0}$, which will be achieved by means of the automatic iterative method.

From the point of view of the analog computer technique, it is most advantageous to apply the iterative process based on the modified Newton-Raphson iteration used by Brunner [3].

We shall choose Q, the quantity expressing the fulfilment of all the boundary conditions, called the error criterion or the criterion function, in the following form:

$$(1.3) \qquad Q = \tfrac{1}{2} \sum_{i=1}^{m} g_i(t) \left[\varepsilon_i(t)\right]^2\,, \quad i = 1, 2, \ldots, m\,,$$

where $g_i(t)$ are weight functions, $g_i(t) > 0$, and $\varepsilon_i(t)$ are error functions which have zero values, when boundary conditions in point $y = L$ are fulfilled, while $y \equiv t$.

We shall express the error function in the form:

$$(1.4) \qquad \varepsilon_i(t) = u_i(L) - u_i^*(L)$$

where $u_i(L)$ is the value of function $u_i(y)$ in point $y = L$ and $u_i^*(L)$ is the required value of function $u_i(y)$ in point $y = L$.

Satisfying boundary conditions demands the minimisation of the criterion function Q. We shall obtain conditions for minimal Q by differentiating equation (1.3) with respect to $[\mathrm{d}u_i(y)/\mathrm{d}y]_{x=0} = \dot{u}_{i0}$, and by requiring that the obtained derivations be zero:

$$(1.5) \qquad \frac{\partial Q}{\partial \dot{u}_{i0}} = \sum_{i=1}^{m} g_i(t)\, \varepsilon_i(t) \frac{\partial \varepsilon_i(t)}{\partial \dot{u}_{i0}} = 0\,, \qquad i = 1, 2, \ldots, m\,.$$

For solving these equations by an analog computer we may implement the iterative method, which will allow to use the same circuit for all equations of system (1.5). The solution will occur cyclically, only one parameter, $\dot{u}_{i0}$,will be set within one step. The iterative method, which will be used in the minimisation of Q, is derived from the generalised Newton-Raphson method of the steepest descent [4]. To determine the direction of the steepest descent of errors $\varepsilon_i(\dot{u}_{i0})$, we try to obtain that the error vector after the rearrangement of $\dot{u}_{i0}$, i.e., $\varepsilon_i(\dot{u}_{i0} + \delta\dot{u}_{i0})$ be perpendicular to $\varepsilon_i(\dot{u}_{i0})$. Moreover, condition (1.5) must be satisfied; it has, after using current symbols, the form:

$$\sum_{i=1}^{m} g_i \frac{\partial \varepsilon_i}{\partial \dot{u}_{i0}} \varepsilon_i(\dot{u}_{i0} + \delta\dot{u}_{i0}) = 0 \,. \tag{1.6}$$

The increase $\delta\dot{u}_{i0}$ is determined from this condition by which $\dot{u}_{i0}$ is to be increased to reach the minimum Q. For a sufficiently small $\delta\dot{u}_{i0}$ we get, after re-arrangement:

$$\delta\dot{u}_{i0} = \frac{\sum_{i=1}^{m} g_i(t)\, \varepsilon_i(t) \dfrac{\partial \varepsilon_i(t)}{\partial \dot{u}_{i0}}}{\sum_{i=1}^{m} g_i(t) \left(\dfrac{\partial \varepsilon_i(t)}{\partial \dot{u}_{i0}}\right)^2} \,. \tag{1.7}$$

The iterative process for the determination of parameters $\dot{u}_{i0}$ satisfying the required conditions is derived from the equation

$$\dot{u}_{i0}^{(k+1)} = \dot{u}_{i0}^{(k)} + \delta\dot{u}_{i0}^{(k)} \tag{1.8}$$

where k is the number of iterations.

The derived iterative process converges; we could prove it by showing that the criterion function Q decreases. To determine the partial derivation $\partial\varepsilon_i/\partial\dot{u}_{i0}$ we shall apply the method of parameter influence functions as elaborated by Meissinger [5].

Partial derivatives $\partial u_i/\partial\dot{u}_{i0}$ are called parameter influence functions, and they express the impact of parameter changes $\dot{u}_{i0}$ on the solution of $u_i(y)$. Their solution gives sets of so called sensitivity equations, together with the solutions of the sets of original partial differential equations. Sensitivity equations are obtained, if the original systems of partial differential equations is differentiated with respect to $\dot{u}_{i0}$.

For parameter influence functions we use the following expressions:

$$\frac{\partial u_i(y, \dot{u}_{i0})}{\partial \dot{u}_{i0}} = w_i(y, \dot{u}_{i0}) \,, \tag{1.9}$$

$$\frac{\partial \dot{u}_i}{\partial \dot{u}_{i0}} = \frac{\partial}{\partial y} \frac{\partial u_i}{\partial \dot{u}_{i0}} = \dot{w}_i(y, \dot{u}_{i0}) \,,$$

$$\frac{\partial \ddot{u}_i}{\partial \dot{u}_{i0}} = \frac{\partial^2}{\partial y^2} \frac{\partial u_i}{\partial \dot{u}_{i0}} = \ddot{w}_i(y, \dot{u}_{i0}) \,.$$

The final form of the sensitivity functions is:

$$(1.10) \qquad \ddot{w}_i \doteq \frac{1}{a} \frac{w_i - w_{i-1}}{\Delta x}$$

$$w_{i0} = \frac{\partial u_{i0}}{\partial \dot{u}_{i0}} = 0\,,$$

$$\dot{w}_{i0} = \frac{\partial \dot{u}_{i0}}{\partial \dot{u}_{i0}} = 1\,,$$

$$i = 1, 2, \ldots, m\,.$$

This equation is solved for $w_i(y, \dot{u}_{i0})$.

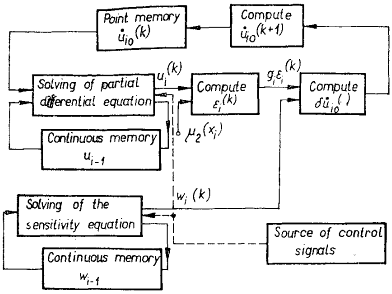
Point memory $\dot{u}_{i0}(k)$
Compute $\dot{u}_{i0}(k+1)$
Solving of partial differential equation
$u_i^{(k)}$
Compute $\varepsilon_i^{(k)}$
$g_i\varepsilon_i^{(k)}$
Compute $\delta\ddot{u}_{i0}(\,)$
Continuous memory u_{i-1}
$\mu_2(x_i)$
$w_i\ (k)$
Solving of the sensitivity equation
Continuous memory w_{i-1}
Source of control signals

Fig. 3.

The block diagram of the solution is given in Fig. 3. The source of control signals of the block diagram is described in section 3.

2. CONTINUOUS ANALOG MEMORY

The continuous memory consists, in general, of $n - 1$ point analog memories. The selection of n depends on the form of the function we wish to remember, and on the equipment of the computer at our disposal. In the case given below, we have selected $n = 7$. The continuous analog memory is made of two parts. One is the input part in which function u_i is remembered, and the other is the output part in which we read the function u_{i-1}. A 20-branch potentiometer of the ARIPOT type is also a part of the output portion of the continuous analog memory. It is built in the servomultiplier, on which the linear interpolation of the function u_{i-1} is performed, according to the relation:

$$(2.1) \qquad u(y) = u(y_j) + [y - y_{(j)}] \frac{u(y_{j+1}) - u(y_j)}{y_{(j+1)} - y_{(j)}}\,,$$

$$j = 0, 1, \ldots, 6\,.$$

The overall scheme of the continuous memory is in Fig. 4, where the control circuits are also shown for the continuous analog memory. The transfer of information

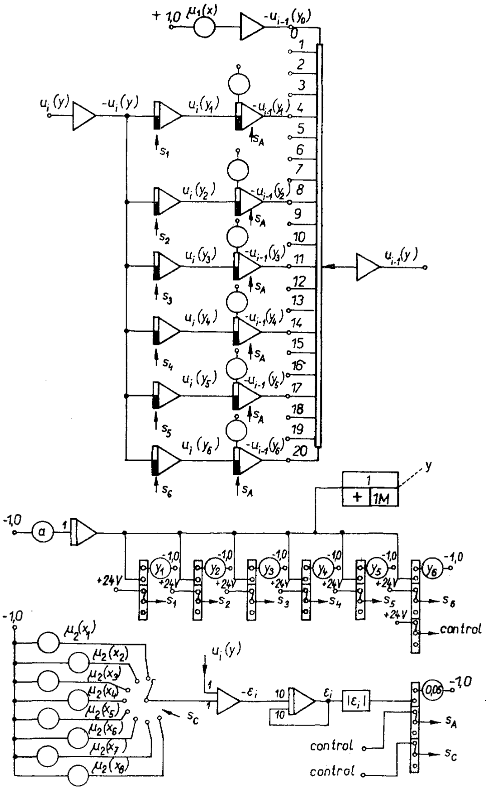

Fig. 4.

from the input part of the continuous analog memory into its output part is dependent on the condition:

$$|u_i(y_6) - \mu_2(x_i)| \leqq 0.5 \text{ V}, \tag{2.2}$$

i.e., on the satisfaction of the other boundary condition with the given precision.

3. SEQUENCE OF CONTROL SIGNALS

When the sequence of control signals is formed, the following operations must be performed:

1. Control of the iterative computing, which consists of the computer modes "Initial conditions", "Hold", "Compute", for all the units in the computer network (in Fig. 5 all the units with the exception of the unit controlled by signal S_B) and of the modes "Hold" and "Compute" for the control of point memory and accumulator.

2. Control of the input part of the continuous analog memory, which consists of the sequence of modes "Compute" and "Hold".

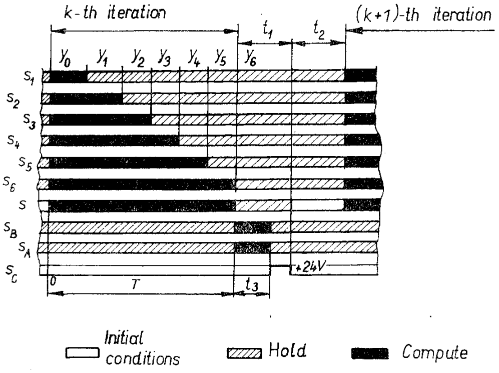

Fig. 5.

3. Control of the output part of the continuous analog memory, which consists of the sequence of modes "Compute" and "Hold" in relation to the satisfaction of the other boundary condition

$$\varepsilon_i \geqq |u_i(y_6) - \mu_2(x_i)| \, . \tag{3.1}$$

4. Control of the step-by-step selector, which sets new initial and new boundary conditions.

Figure 5 shows the time control diagram described in points 1 to 4.

The algorithm for solving is on Figure 6.

Example

As an example we shall give the solution of a one-dimensional heat-conduction equation in an idealised rod according to equation (1)

$$\frac{\partial u(x, y)}{\partial x} = a\frac{\partial^2 u(x, y)}{\partial y^2}\,!$$

The program diagram is on Figure 7.

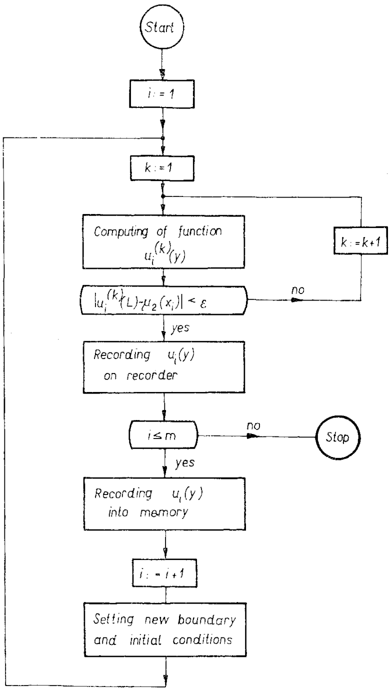
Start
$i:=1$
$k:=1$
Computing of function $u_i^{(k)}(y)$
$k:=k+1$
$|u_i^{(k)}(L)-\mu_2(x_i)| < \varepsilon$
no
yes
Recording $u_i(y)$ on recorder
$i\leq m$
no
Stop
yes
Recording $u_i(y)$ into memory
$i:=i+1$
Setting new boundary and initial conditions

Fig. 6.

The value of the weight coefficient was chosen experimentally as $g_i = 0.29$, the satisfaction of the boundary condition $\mu_2(x_i)$ was achieved with an accuracy of ± 0.5 V, with machine unit 100 V. The number of iterations necessary for reaching the boundary condition with the required accuracy was $2 \div 3$. Axonometrically plotted solution is in Figure 8. The network for the axonometric plotting the solution of a partial differential equation is described in [7].

Autors have tried to explain how to solve partial differential equations by the method CSDT using the sensitivity equations. It is possible to solve also this kind of hybrid type problems using only analog computer (e.g. AP3-M) by the given implementation of continuous analog memory.

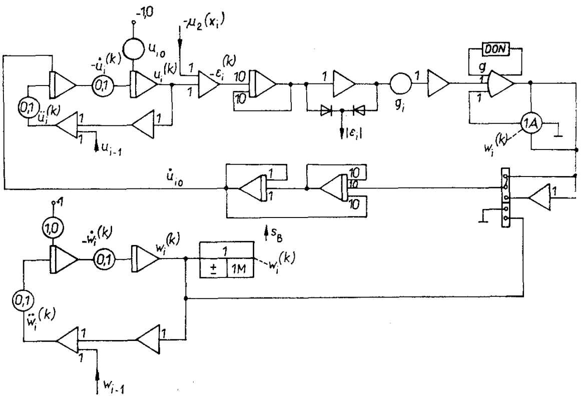

Fig. 7.

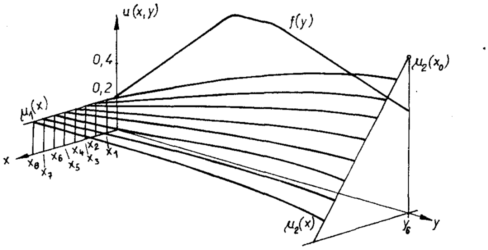

Fig. 8.

The paper was presented on the International Conference — Application of Hybrid Computation in Gliwice, Poland, June 22—24, 1972.

(Received August 1, 1972.)

REFERENCES

[1] I. Plander: Matematické metódy a programovanie analógových počítačov. Vydav. SAV, Bratislava 1969.

[2] L. Gvozdjak: Analógové počítače. Lecture notes SVŠT. Bratislava 1967.

[3] W. Brunner: An iteration procedure for parametric model building and boundary value problem. Proc. WJCC 1961, pp. 517—533.

[4] G. N. Lange: Numerical methods for high-speed computers. London 1960.

[5] H. F. Meissinger: The use of parameter influence coefficients in computer analysis of dynamic systems. Proc. WJCC 1960, pp. 181—192.

[6] G. A. Bekey, W. J. Karplus: Hybrid computation. John Wiley Inc., New York 1968.

[7] V. Kanich: Trojrozmerný axonometrický zápis na súradnicovom zapisovači. Automatizace *XIV* (1971), 2, 39—40.

Ing. Ružena Apalovičová, Ing. Jozef Babirád; Katedra matematických strojov EF SVŠT (Department of Computer Science, Slovak Technical University), Vazovova 1/b, 800 00 Bratislava. Czechoslovakia.