Int. J. Heat Mass Transfer. Vol. 22, pp. 1375–1379
Pergamon Press Ltd. 1979. Printed in Great Britain

NEW COMPUTATIONAL IMPLEMENTATION OF NON-LINEAR DERIVATIVE BOUNDARY CONDITIONS FOR A.D.I. METHODS

BO STRÖM

Department of Heat and Furnace Technology, The Royal Institute of Technology, Stockholm, Sweden

(*Received* 18 *September* 1978)

Abstract—Two new methods of introducing non-linear derivative boundary conditions for A.D.I. methods which solve the heat conduction equation in two space variables are suggested. The first method is fast, but less accurate than the second method with respect to the time variable. The second method has the same order of accuracy as the Crank–Nicolson method. The second method is most suited for recalculation of the previous time step with a new set of boundary conditions. The first method allows non-rectangular regions. The second method becomes less efficient if extended to non-rectangular regions.

NOMENCLATURE

a,b,c, weight factors;
$f, f', g, g', h_1, h'_1, h_2, h'_2, h_3, h'_3$, weight factors;
A,D, matrixes;
u, dependent variable;
$\mathbf{x}$, vector;
x,y, coordinates;
t, time variable;
$\mathbf{n}$, outward unit normal to surface;
h,k, step length in x and y directions respectively;
N,M, $N+1$ and $M+1$ are the number of gridpoints in the x and y directions respectively;
$r_1 = \frac{k\Delta t}{h^2}$, $r_2 = \frac{k\Delta t}{k^2}$, dimensionless quantities;
A,B,E_1,E_2,F_1,F_2, boundary values;
P, point at the boundary;
F, fictitious value.

Greek symbols

κ, thermal diffusivity;
λ, thermal conductivity;
α, convective heat transfer coefficient;
ε, emissivity;
σ, Stefan–Boltzmann constant;
Δt, time step.

Index

i,j, referring to nodpoints;
n, $(n+1)^*$, $(n+1)$, time levels;
k, referring to boundary point.

Operators

$\partial_x^2, \partial_y^2$, central difference operators in x and y directions respectively.

1. INTRODUCTION

THE PROBLEM of determining the temperature distribution in solids during heating or cooling is an important one. Two examples are the process control of reheating furnaces and the cooling of material during hot rolling. Mathematically the problem is given as an initial boundary value problem for the heat conduction equation:

$$\rho c_p \frac{\partial u}{\partial t} = \operatorname{div}(\lambda \operatorname{grad} u), \quad \text{in } \Omega$$

$$-\lambda \frac{\partial u}{\partial \mathbf{n}} = \alpha(u-u_0) + \sigma\varepsilon(u^4 - u_0^4), \quad \text{on } \partial\Omega \text{ for all } t,$$

$u(\mathbf{x}, 0)$ = initial temperature distribution, where $\partial\Omega$ is the boundary of the region Ω.

Even if we assume constant coefficients the problem stated above is not solvable analytically, due to the non-linearity in the boundary condition. We shall assume that $\mathbf{x}$ is a two dimensional vector and that Ω is a rectangular region unless otherwise stated. We also assume for notational reasons that the coefficients are constants. We note that in the algorithms non-linear coefficients are handled by letting for instance the thermal diffusivity κ assume the constant value $\kappa(u_{ij})$, which may differ for different nod points. Iterations are then possible.

2. THEORETICAL DEVELOPMENT

2.1. *First A.D.I. scheme*

Consider the heat conduction equation:

$$\frac{\partial u}{\partial t} = \kappa\left(\frac{\partial^2 u}{\partial_x^2} + \frac{\partial^2 u}{\partial_y^2}\right), \tag{1a}$$

in the region R given by Fig. 1 with the initial condition

$$u(x, y, 0) = \text{constant}, \quad (x, y) \in R, \tag{1b}$$

and the boundary condition

$$-\lambda \frac{\partial u}{\partial n} = \alpha(u-u_0) + \sigma\varepsilon(u^4 - u_0^4), \quad \text{on } \partial R \text{ for all } t, \tag{1c}$$

where ∂R is the boundary of R.

Let h,k denote the mesh size for x and y respectively;

$$x_i = ih, \; i = 0, \ldots, N \text{ and } y_j = jk, \; j = 0, \ldots, M.$$

Let Δt denote the mesh size for t; $t^n = n\Delta t$.

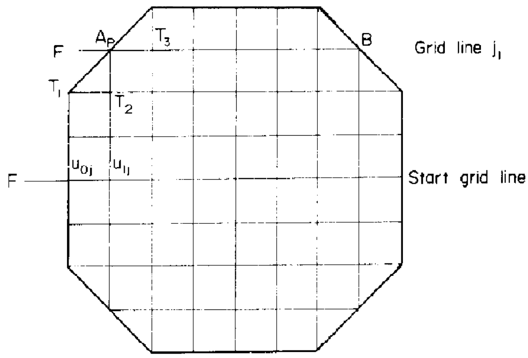

FIG. 1. Showing the region R used in describing the first A.D.I. Scheme $(k = h) A = u(P)$.

The original A.D.I. method of Peaceman and Rachford [1] takes the form:

$$(1-\tfrac{1}{2}r_1\partial_x^2)u_{ij}^{(n+1)^*} = (1+\tfrac{1}{2}r_2\partial_y^2)u_{ij}^n \tag{2a}$$

$$(1-\tfrac{1}{2}r_2\partial_y^2)u_{ij}^{n+1} = (1+\tfrac{1}{2}r_1\partial_x^2)u_{ij}^{(n+1)^*}, \tag{2b}$$

where $u_{ij}^n = u(x_i, y_j, t^n)$ are known for all $(x_i, y_j) \varepsilon R \cup \partial R$. The operators ∂_x^2 and ∂_y^2 are the usual central difference operators in the x and y directions respectively, and $r_1 = \kappa\Delta t/h^2$, $r_2 = \kappa\Delta t/k^2$. The quantities $u_{ij}^{(n+1)^*}$ can be considered as intermediate solutions. If we add equations (2a) and (2b) to eliminate $(1+\frac{1}{2}r_1\partial_x^2)u_{ij}^{(n+1)^*}$, the result is:

$$u_{ij}^{(n+1)^*} = \tfrac{1}{2}(1+\tfrac{1}{2}r_2\partial_y^2)u_{ij}^n + \tfrac{1}{2}(1-\tfrac{1}{2}r_2\partial_y^2)u_{ij}^{n-1}. \tag{3}$$

Thus we obtain by taking equations (2a) and (3) an A.D.I. scheme which requires less arithmetic operations as noted by Varga [2] and Fairweather and Mitchell [3]. The boundary initial value problem (1a–c) is equivalent to a Dirichlet problem where the boundary condition is:

$$u_{ij}(t) = g(x_i, y_j, t), \quad (x_i, y_j) \in \partial R \text{ for all } t, \tag{4}$$

and the $g_{ij}^n = g(x_i, y_j, t^n)$ are to be determined. We use equation (2a) at each interior grid point along a horizontal grid line which has a vertical boundary at each end. Let A and B denote the boundary values at the left and right boundary point respectively. We obtain a tridiagonal system of equations $Au^{(n+1)^*} = D$ where $D = (d_1, \ldots, d_{N-1})$. We add two more equations $u^{(n+1)^*} = A$ at the left boundary point and $u^{(n+1)^*} = B$ at the right boundary point. We solve this new tri-diagonal system of equations with three different right hand sides corresponding to the vectors $(1, \ldots, 0)$, $(0, \ldots, 1)$ and $(0, d_1, \ldots, d_{N-1}, 0)$. We obtain three solution vectors $\mathbf{a} = (a_i)$, $\mathbf{b} = (b_i)$ and $\mathbf{c} = (c_i)$, $i = 0, \ldots, N$. As all equations are linear we derive the solution as:

$$u_{ij}^{(n+1)^*} = a_{ij}A + b_{ij}B + c_{ij}, \tag{5}$$

where j is the number of the grid line. We use the boundary condition (1c) to find the values of A and B. Figure 1 shows the left boundary point with the fictitious value F. The boundary condition is approximated by a central difference operator. Thus we get:

$$-\lambda \frac{F - u_{1j}^{(n+1)^*}}{2h} = \alpha(A - u_0) + \sigma\varepsilon(A^4 - u_0^4), \tag{6}$$

where $u_{1j}^{(n+1)^*} = a_{1j}A + b_{1j}B + c_{1j}$ and u_0 is evaluated at the boundary point at time level $n+\frac{1}{2}$. This choice of time introduces an error as the solution $u^{(n+1)^*}$ is not equal to the solution $u^{n+1/2}$. By applying equation (2a) at the boundary point $(i = 0)$ and letting $F = u_{-1j}^{(n+1)^*}$ we derive an equation with the two unknowns A and B. A similar procedure at the right boundary point gives another equation in the two unknowns A and B. We solve this system of non-linear equations by, e.g., the method of Newton. Now we perform the sweeps in the x-direction up to and including the grid line j_1 in Fig. 1. The boundary condition at this grid line is approximated by, as the outward unit normal is $1/\sqrt{2}\,(-1, 1)$. $h = k$ in this case for simplicity

$$-\lambda \frac{F - T_2}{\sqrt{2}h} = \alpha\left(\frac{A + T_1}{2} - u_0\right) + \sigma\varepsilon\left(\left(\frac{A + T_1}{2}\right)^4 - u_0^4\right), \tag{7}$$

where

$$A = u(P), \quad T_3 = a_{1j_1}A + b_{1j_1}B + c_{1j_1},$$
$$T_1 = u_{0j_1-1}^{(n+1)^*} \quad \text{and} \quad T_2 = u_{1j_1-1}^{(n+1)^*}.$$

T_1 and T_2 are known. u_0 is evaluated at the point P. We apply equation (2a) in the same way as above. Thus we have one equation for the unknowns A and B. The same method is applied at the right boundary point. In a similar way we perform horizontal sweeps for grid lines below the starting grid line. The botton and top grid lines are swept by introducing fictitious values u_{i-1}^n and u_{iM+1}^n respectively. To obtain the solution u^{n+1} we use equation (3) along vertical grid lines and introduce the boundary conditions in a similar way as above. Note that the quantities $(1+\frac{1}{2}r_2\partial_y^2)u_{ij}^n$ are already calculated.

2.2. *Second A.D.I. scheme*

Consider a rectangular region with mesh size h, k in the x and y directions respectively. We perform sweeps in the x direction along each horizontal grid line with the aid of equation (2a). At the upper and lower boundary we introduce fictitious values u_{iM+1}^n and u_{i-1}^n respectively and use the boundary condition to eliminate them. Let A_j and B_j denote the unknown boundary values at the left and right boundary points at the time level $(n+1)^*$. For each horizontal grid line we obtain coefficients a_i, b_i, c_i as before. Hence we can write

$$u_{ij}^{(n+1)^*} = a_{ij}A_j + b_{ij}B_j + c_{ij}, \quad i = 0, \ldots, N,\ j = 0, \ldots, M.$$

Now we use equation (3) at the vertical grid lines corresponding to $i = 1$ and $i = N-1$. We derive the following system of equations:

$$\tfrac{1}{2}(1+\tfrac{1}{2}r_2\partial_y^2)u_{ij}^n+\tfrac{1}{2}(1-\tfrac{1}{2}r_2\partial_y^2)u_{ij}^{n+1} = (u_{ij}^{(n+1)^*}) = a_{ij}(\tfrac{1}{2}(1+\tfrac{1}{2}r_2\partial_y^2)u_{0j}^n+\tfrac{1}{2}(1-\tfrac{1}{2}r_2\partial_y^2)u_{0j}^{n+1}) + b_{ij}(\tfrac{1}{2}(1+\tfrac{1}{2}r_2\partial_y^2)u_{Nj}^n+\tfrac{1}{2}(1-\tfrac{1}{2}r_2\partial_y^2)u_{Nj}^{n+1})+c_{ij}, \quad j=1,\ldots,M-1. \quad (8)$$

We add two more equations $u_{10}^{n+1}=E_1$ and $u_{1M}^{n+1}=E_2$ to the system corresponding to $i=1$ and $u_{N-10}^{n+1}=F_1$, $u_{N-1M}^{n+1}=F_2$ to the system corresponding to $i=N-1$. By solving the system ($i=1$) with respect to u_{1j}^{n+1} for the $2\times(M+1)+3$ right hand vectors given by the $(M+1\times 2M+5)$ matrix D. We derive expressions for u_{1j}^{n+1}. The elements in the kth column of D for $0\leqslant k\leqslant M$ are given by:

$$d_{0k}=d_{Mk}=0,\quad d_{jk}=\tfrac{1}{2}a_{1j}(1-\tfrac{1}{2}r_2\partial_y^2)u_{0j}^{n+1},$$
$$j=1,\ldots,M-1 \quad \text{where} \quad u_{0i}^{n+1}=1 \quad \text{and}$$
$$u_{00}^{n+1}=\ldots=u_{0i-1}^{n+1}=u_{0i+1}^{n+1}=\ldots=u_{0M}^{n+1}=0$$
$$\text{for} \quad i=0,\ldots,M.$$

The elements d_{jk} for $M+1\leqslant k\leqslant 2M+2$ are given by $d_{0k}=d_{Mk}=0$,

$$d_{jk}=\tfrac{1}{2}b_{1j}(1-\tfrac{1}{2}r_2\partial_y^2)u_{Nj}^{n+1},\quad j=1,\ldots,M-1,$$

where

$$u_{Ni}^{n+1}=1u_{N0}^{n+1}=\ldots=u_{Ni-1}^{n+1}=u_{Ni+1}^{n+1}=\ldots=u_{NM}^{n+1}=0,$$

$i=0,\ldots,M$.

The $2M+3$ and $2M+4$ columns in D correspond to the vectors $(1,\ldots,0)$ and $(0,\ldots,1)$ respectively. The last column in D is given by the vector with elements:

$$d_1=d_M=0,$$
$$d_j=-\tfrac{1}{2}(1+\tfrac{1}{2}r_2\partial_y^2)u_{1j}^n+\tfrac{1}{2}a_{1j}(1+\tfrac{1}{2}r_2\partial_y^2)u_{0j}^n+\tfrac{1}{2}b_{1j}(1+\tfrac{1}{2}r_2\partial_y^2)u_{Nj}^n+c_{ij}.$$

Solving this system of equations enables us to write:

$$u_{1j}^{n+1}=\sum_{k=0}^{M}g_{jk}u_{0k}^{n+1}+\sum_{k=0}^{M}f_{kj}u_{Nk}^{n+1}+h_{1j}E_1+h_{2j}E_2+h_{3j}, \quad (9)$$

where g_{kj} and f_{kj} are the solution matrices corresponding to the first $M+1$ columns in D and the following $M+1$ columns in D respectively. h_{1j}, h_{2j} and h_{3j} are the solution vectors corresponding to the last three columns in D respectively. In the same way we solve the system of equations corresponding to the grid line $i=N-1$. Hence we are able to write:

$$u_{N-1j}^{n+1}=\sum_{k=0}^{M}g'_{kj}u_{0k}^{n+1}+\sum_{k=0}^{M}f'_{kj}u_{Nk}^{n+1}+h'_{1j}F_1+h'_{2j}F_2+h'_{3j}. \quad (10)$$

Note in this particular case, when the coefficients are constants, we would only have to solve the last system of equations for the last column in the matrix D. The remaining problem is to determine the boundary values u_{0j}^{n+1}, u_{Nj}^{n+1}, $j=0,\ldots,M$ and u_{10}^{n+1}, u_{1M}^{n+1}, u_{N-10}^{n+1} and u_{N-1M}^{n+1}. That is $2M+6$ unknowns. The equations are to be determined from the boundary conditions at the above mentioned points. We only describe the left part of the boundary. The right part is treated in a similar way. We distinguish between corner points, left vertical boundary and the upper and lower boundary points.

2.2.1. *Boundary point on the left vertical boundary except corner points.* We use the heat conduction equation at the boundary point approximated by

$$u_{0j}^{n+1}=\tfrac{1}{2}(r_1\partial_x^2u_{0j}^{n+1}+r_2\partial_y^2u_{0j}^{n+1})+\tfrac{1}{2}(r_1\partial_x^2u_{0j}^n+r_2\partial_y^2u_{0j}^n), \quad (11)$$

where the fictitious values u_{-1j}^{n+1} and u_{-1j}^n are determined from the boundary condition (1c) approximated by

$$-\lambda(u_{-1j}-u_{1j})=2h\alpha(u_{0j}-u_0)+2h\sigma\varepsilon(u_{0j}^4-u_0^4) \quad \text{for } t^n \text{ and } t^{n+1}.$$

As

$$u_{1j}^{n+1}=\sum_{k=0}^{M}g_{kj}u_{0k}^{n+1}+\sum_{k=0}^{M}f_{kj}u_{Nk}^{n+1}+h_{1j}E_1+h_{2j}E_2+h_{3j}$$

we have derived an equation only containing the unknown boundary values at time level $n+1$.

2.2.2. *Corner points.* Consider the upper left corner, the lower corner is treated similarly. As above we use the heat conduction equation at the corner point. The four fictitious values u_{-1M}^{n+1}, u_{-1M}^{u}, u_{0M+1}^{n+1} and u_{0M+1}^n are determined from the boundary conditions:

$$-\lambda(u_{-1M}-u_{1M})=2h\alpha(u_{0M}-u_0)+2h\sigma\varepsilon(u_{0M}^4-u_0^4)$$

and

$$-\lambda(u_{0M+1}-u_{0M-1})=2h\alpha(u_{0M}-u_0)+2h\sigma\varepsilon(u_{0M}^4-u_0^4) \quad \text{for } t^n \text{ and } t^{n+1}.$$

Hence we have an equation only containing the unknown boundary values at time level $n+1$.

2.2.3. *Boundary points on the upper and lower boundary.* We consider only the upper left boundary point; the lower is treated in the same way. We apply (3) at the boundary point (x_1, y_M) and rearrange to obtain $(1-\tfrac{1}{2}r_2\partial_y^2)u_{1M}^{n+1}=2u_{1M}^{(n+1)^*}-(1+\tfrac{1}{2}r_2\partial_y^2)u_{1M}^n$. The fictitious values u_{1M+1}^n and u_{1M+1}^{n+1} are determined from the boundary condition (1c) approximated by

$$-\lambda(u_{1M+1}-u_{1M-1})=2k\alpha(u_{1M}-u_0)+2k\sigma\varepsilon(u_{1M}^4-u_0^4)$$

for t^n and t^{n+1}. The boundary value

$$u_{1M}^{(n+1)^*}=a_{1M}u_{0M}^{(n+1)^*}+b_{1M}u_{NM}^{(n+1)^*}+c_{1M}$$

from equation (2a). Equation (3) applied at the two upper corner points (see above) gives

$$2u_{iM}^{(n+1)^*}=(1-\tfrac{1}{2}r_2\partial_y^2)u_{iM}^{n+1}+(1+\tfrac{1}{2}r_2\partial_y^2)u_{iM}^n \quad \text{for } i=0 \text{ and } N.$$

The fictitious values introduced are known from the equations at the corner points. We have another equation only containing the unknown boundary values at time level

$$n+1 \text{ as } u_{1M}^{n+1} = \sum_{k=0}^{M} g_{kM-1} u_{0k}^{n+1} + \sum_{k=0}^{M} f_{kM-1} u_{Nk}^{n+1} + h_{1M-1}E_1 + h_{2M-1}E_2 + h_{3M-1}.$$

Hence we have $2(M-1)+4+4$ equations in the $2M+6$ unknown boundary values. After solving this non-linear system of equations we use equation (3) at each corner point together with the boundary equations to obtain the boundary values u_{00}, u_{0M}, u_{N0} and u_{NM} at time level $(n+1)^*$. Equation (3) is then used to obtain the remaining vertical boundary values at time level $(n+1)^*$ for $i = 0$, N and $j = 1, \ldots, M-1$. Thus we know $u^{(n+1)^*}$ for every interior point as

$$u_{ij}^{(n+1)^*} = a_{ij} u_{0j}^{(n+1)^*} + b_{ij} u_{Nj}^{(n+1)^*} + c_{ij}.$$

The coefficients a_{ij}, b_{ij} and c_{ij} have already been determined by equation (5). The values $u_{ij}^{(n+1)}$ for $i = 2, \ldots, N-2$, $j = 0, \ldots, M$ are determined in the same way as the A.D.I. method 1.

Note that if we want to recalculate a time step with a different set of boundary conditions we only need to recalculate the solution to the $2M+6$ boundary conditions equations and so on.

3. NUMERICAL RESULTS

The two A.D.I. schemes together with an explicit scheme and the Crank–Nicolson scheme were implemented on an IBM 370/165 computer. The solutions of non-linear system of equations was done by Newton's method in the A.D.I.1 and the Crank–Nicolson schemes. (A.D.I.2 used for this purpose a standard IMSL-library subroutine ZSYSTEM 4.) It is most likely that this general routine is slower than the method of Newton in the test cases. On the other hand Newton's method requires a better starting value to converge. The following tests were run. All data are in the S.I. system.

A. *Accuracy test*

The input data were:
Thermal diffusivity = thermal conductivity = 1, surrounding media temperature = 0, initial temperature distribution = 1, width = height = 1. The boundary conditions are given by $\alpha = 1$, $\varepsilon = 0$. In the A case 11×11 nod points were used with a time step $= 0.23229 \times 10^{-2}$. In the B case 21×21 nod points were used with a time step $= 0.59488 \times 10^{-3}$. The solutions for the different methods were compared with the analytical solution. The results are shown in Table 1.

Table 1. Maximal error $\times 10^2$ for different methods compared with the analytical solution

Method of solution Time $\times 10^2$ (s)	Explicit A	Explicit B	A.D.I.1 A	A.D.I.1 B	Crank–Nic. A	Crank–Nic. B	A.D.I.2 A	A.D.I.2 B
0.0595	—	5.352	—	2.461	—	1.505	—	1.504
0.119	—	2.653	—	1.692	—	1.068	—	1.067
0.178	—	2.856	—	1.264		0.808	—	0.807
0.232	10.157	2.372	4.578	1.040	2.744	0.667	2.743	0.667
0.465	4.681	1.948	3.072	0.614	1.908	0.398	1.907	0.397
0.697	4.652	1.760	2.217	0.467	1.389	0.296	1.387	0.296
0.929	3.760	1.633	1.723	0.386	1.079	0.244	1.078	0.243
1.394	3.253	1.463	1.415	0.296	0.750	0.188	0.749	0.187
6.040	1.815	—	0.383	—	0.211	—	0.208	—
11.614	1.549	—	0.285	—	0.134	—	0.128	—

Table 2. Comparison of computing time for different methods

Method of solution	Explicit A	Explicit B	A.D.I.1. A	A.D.I.1. B	Crank–Nic. A	Crank–Nic. B	A.D.I.2 A	A.D.I.2 B
Number of iterations	540	2084	6	6	6	6	6	6
Computing time (s)	3.21	34.74	0.22	0.71	8.61	83.08	15.45	152.08

B. *Computational speed*

The input data were:
Thermal diffusivity $= 6 \times 10^{-6}$, thermal conductivity = 30, ambient temperature = 1600, initial temperature distribution = 300, width = height = 0.05. The boundary conditions are given by $\alpha = 20$, $\varepsilon = 0.7$. In the A case 11×11 nod points were used and in the B case 21×21 nod points. The computation stopped when the centre temperature was 1550. This heating time was compared between the different methods to ensure the same accuracy. The results are shown in Table 2.

Table 3. Comparison of computing time for different methods for the case with simulation of 100 different boundary

Method of solution	Explicit	A.D.I.1	Crank–Nic.	A.D.I.2
Computing time (s)	111.02	26.34	91.73	50.17

C. *Simulation of different boundary conditions*

The input data were:

Thermal diffusivity $= 6 \times 10^{-6}$, thermal conductivity $= 30$, ambient temperature $= 300$, initial temperature distribution $= 1550$, width $\times$ height $= 0.1 \times 0.025$. The boundary conditions are given by $\varepsilon = 0.7$ and $\alpha = 50$ on vertical boundaries and $\alpha = 20$, $\varepsilon = 0.7$, elsewhere. 21×6 grid points were used. This boundary condition was recalculated 100 times. The time step for A.D.I.2 and Crank–Nicolson was 240. The time step for A.D.I.1 was chosen to 40 i.e. 6 iterations. The explicit method chooses its own time step. The results are shown in Table 3.

4. CONCLUSIONS

The different tests show that the A.D.I.1 method seems to be the optimum of accuracy and computational speed in most cases. The slowness of A.D.I.2 method is presumably due to the non-linear equation solver. The A.D.I.2 method's advantage of recalculating boundary conditions is more useful to problems with many nod points.

Acknowledgement—This work was supported by the Swedish Board of Technical Development.

REFERENCES

1. D. W. Peaceman and H. H. Rachford, The numerical solution of parabolic and elliptic differential equations, *J. Soc. Indust. applic. Math.* **3**, 28–41 (1955).
2. R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey (1962).
3. G. Fairweather and A. R. Mitchell, A new computional procedure for A.D.I. methods, *SIAM J. Numer. Analysis* **4**, 163–170 (1967).
4. IMSL Library 1, Edition 6, (FORTRAN IV) IBMs/370–360 (1977).

NOUVEL OUTIL POUR LES CONDITIONS AUX LIMITES DIFFERENTIELLES NON LINEAIRES DANS LES METHODES A.D.I.

Résumé—On propose deux nouvelles méthodes pour introduire des conditions aux limites différentielles non linéaires dans des méthodes A.D.I., pour résoudre l'équation de conduction thermique à deux variables d'espace. La première méthode est rapide mais moins précise que la seconde, par rapport à la variables de temps. La seconde méthode est du même ordre de précision que la méthode de Crank–Nicolson et elle est plus indiquée pour le calcul du pas de temps avec un nouveau système de conditions aux limites. La première méthode convient aux domaines non rectangulaires tandis que la seconde est moins efficace dans ce cas.

NEUE RECHNERISCHE ERFÜLLUNG VON NICHTLINEAREN ABLEITUNGEN IN RANDBEDINGUNGEN BEI A.D.I.-METHODEN

Zusammenfassung—Zwei neue Methoden werden zur Einführung von nichtlinearen Ableitungen in Randbedingungen bei A.D.I.-Methoden vorgeschlagen, welche die Wärmeleitungsgleichung mit zwei Ortsvariablen lösen. Die erste Methode ist schnell, aber weniger genau als die zweite in bezug auf die Zeitvariable. Die zweite Methode hat denselben Genauigkeitsgrad wie die Crank-Nicolson-Methode. Die zweite Methode ist am geeignetsten zur Nachrechnung des vorausgegangenen Zeitschritts mit einem neuen Satz von Randbedingungen. Die erste Methode ist für nicht-rechteckige Gebiete anwendbar. Die zweite Methode wird weniger leistungsfähig, wenn sie auf nicht-rechteckige Gebiete ausgeweitet wird.

НОВЫЙ СПОСОБ ИСПОЛЬЗОВАНИЯ НЕЛИНЕЙНЫХ ГРАНИЧНЫХ УСЛОВИЙ, СОДЕРЖАЩИХ ПРОИЗВОДНЫЕ, В ADI МЕТОДЕ

Аннотация — Предложены два новых способа использования нелинейных граничных условий, содержащих производные, в ADI методе решения двухмерного уравнения теплопроводности. Первый является более быстрым, но менее точным, чем второй, при определении временной зависимости. По точности второй способ аналогичен методу Крэнка-Никольсона. Его лучше всего использовать для пересчёта предыдущего временного шага с новыми граничными условиями. Первый способ можно применять для непрямоугольных областей, для которых второй способ является менее эффективным.