stichting
mathematisch
centrum

MC

AFDELING TOEGEPASTE WISKUNDE TN 58/70 NOVEMBER

P.J. VAN DER HOUWEN, C. DE VREUGD

A DIFFUSION PROBLEM WITH A DISCONTINUOUS INITIAL CONDITION

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

CONTENTS

	page
Introduction	1
1. Statement of the problem	3
2. The inital condition	3
3. Transformation of the space coördinate	4
4. Discretization of the space variable	7
5. Discretization of the time variable	8
6. Actual computation scheme	11
7. Numerical results	12
8. Description of the program	13
References	18

Introduction

A diffusion problem with a discontinuous initial condition is solved by means of an explicit difference scheme. The scheme has the following features.

Firstly, it uses a non-uniform grid in order to represent the discontinuity adequately.

Secondly, for reasons of stability, Chebyshev polynomials are used which reduce the computation time substantially.

The third characteristic of the scheme is that the boundary condition at infinity is replaced by a condition at a finite point in such a way that the differences between the corresponding solutions remain within the range of accuracy required.

Finally, when the discontinuity is smoothed out the non-uniform grid is replaced by a uniform one.

The calculations were performed on the EL-X8 computer of the Mathematical Centre, where the authors are members of the department of Applied Mathematics and the Computational department, respectively.

The work presented in this paper was carried out at the request of the F.O.M. laboratory.

1. Statement of the problem

In the study of heat diffusion one encounters the following initial boundary value problem:

$$(1.1)\qquad \begin{cases} \dfrac{\partial T}{\partial t} = (\alpha T+\beta)\ \left(\dfrac{\partial^2 T}{\partial r^2} + \dfrac{1}{r}\dfrac{\partial T}{\partial r}\right),\ 0 \le r \le \infty,\ 0 \le t \le \infty, \\ T(r,0) = 1, \quad 0 \le r \le 1, \\ T(r,0) = 0, \quad 1 < r \le \infty, \\ T_r(0,t) = T(\infty,t) = 0, \quad 0 \le t \le \infty. \end{cases}$$

Here, $-\alpha$ and β are given positive constants and T is the unknown function to be determined in a given domain $0 \le r \le r_0$, $0 \le t \le t_0$. The required order of accuracy is 1%.

2. The initial condition

For numerical calculations it is desirable to replace the discontinuous initial function T(r,0) by a continuous one. For instance, we may use the initial function

$$(2.1)\qquad T(r,0) = \begin{cases} 1 & , r \le 1-\Delta' r, \\ \frac{1}{2}\left(1+\cos\left(\dfrac{r-1+\Delta' r}{2\Delta' r}\,\pi\right)\right) & , 1-\Delta' r \le r \le 1+\Delta' r, \\ 0 & , r \ge 1+\Delta' r. \end{cases}$$

By taking $\Delta' r$ sufficiently small, this function approximates the physical initial function with any order of accuracy (see figure 2.1). We have chosen $\Delta' r = .1$.

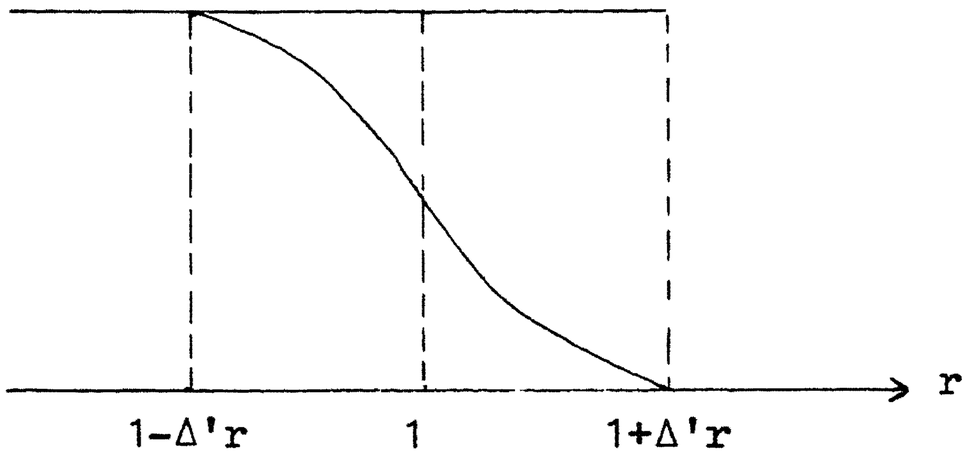

fig. 2.1 Discontinuous and continuous initial function.

3. Transformation of the space coordinate

In order to obtain an efficient difference scheme one should choose the grid points on the r-axis such that the mesh size in the neighbourhood of r = 1 is small compared with the mesh size further away. However, instead of choosing a non-uniform grid we rather introduce a new variable x = x(r) such that equally spaced points at the x-axis correspond with points on the r-axis which are distributed as mentioned above (see also figure 3.1).

Introduction of a transformation x = x(r) results in the equation

$$
(3.1)\qquad \begin{cases} \dfrac{\partial T}{\partial t} = A(x,T)\,\dfrac{\partial^2 T}{\partial x^2} + B(x,T)\,\dfrac{\partial T}{\partial x}\,, \\ \text{where} \\ A(x,T) = (\alpha T+\beta)\,x_r^2, \\ B(x,T) = (\alpha T+\beta)\,(x_{rr} + \dfrac{1}{r}\,x_r). \end{cases}
$$

For numerical calculations it is desirable that the coefficients A(x,T) and B(x,T) vary continuously. The following transformation satisfies this condition:

$$x = \begin{cases} r, & r-1 \leq -\Delta r, \\ r\,\dfrac{\Delta x}{\Delta r} - (1-\Delta r)\,\dfrac{\Delta x-\Delta r}{\Delta r} - \dfrac{1}{\pi}\,(\Delta x-\Delta r)\,\sin\!\left(\dfrac{r-1+\Delta r}{\Delta r}\,\pi\right), & |r-1| \leq \Delta r, \\ r + 2(\Delta x-\Delta r) + 1, & r-1 \geq \Delta r. \end{cases} \tag{3.2}$$

Furthermore, by choosing $\Delta x > \Delta r$ the function $x(r)$ behaves as illustrated in figure 3.1 (see also figure 3.2 where (3.2) is given).

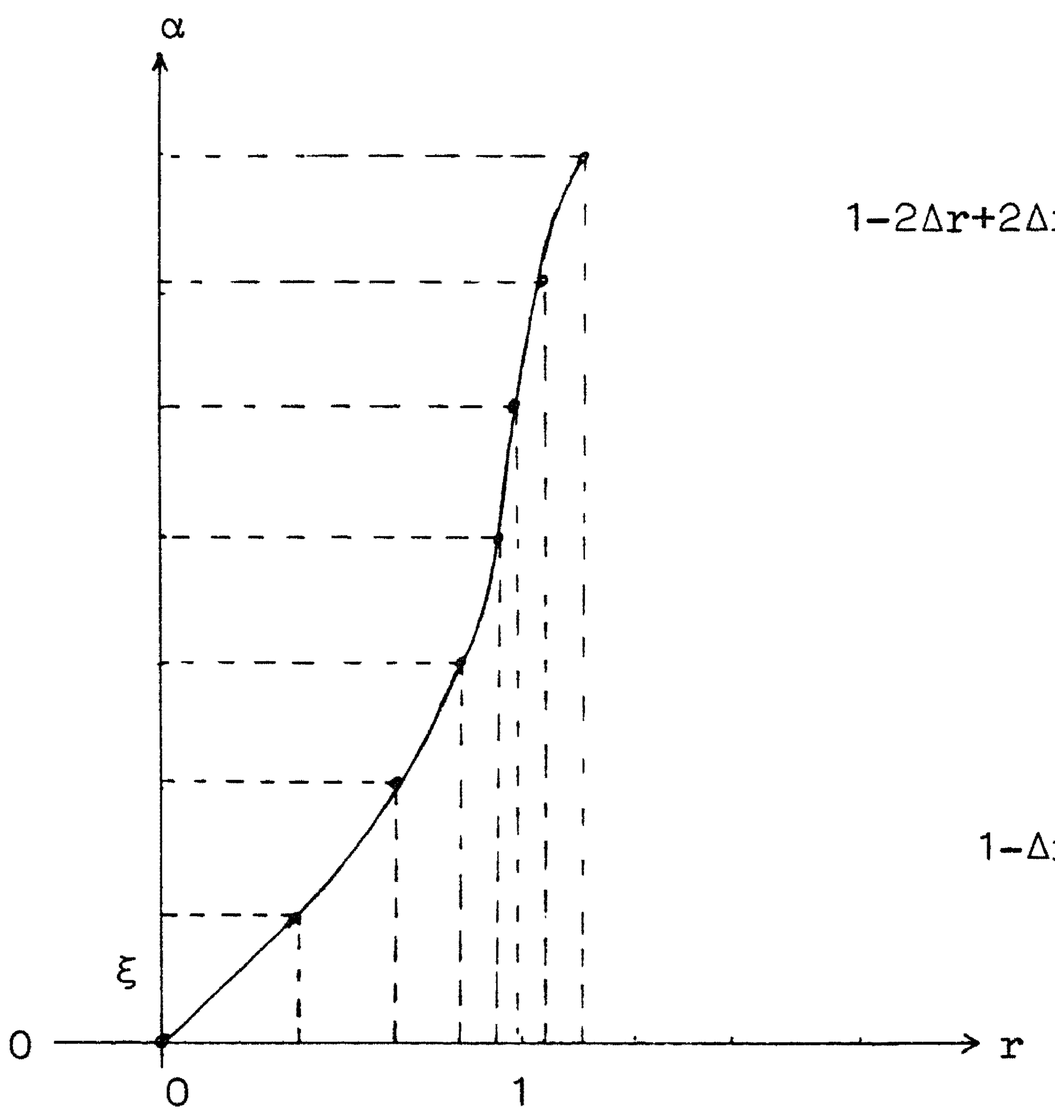

fig. 3.1 The transformation $x = x(r)$

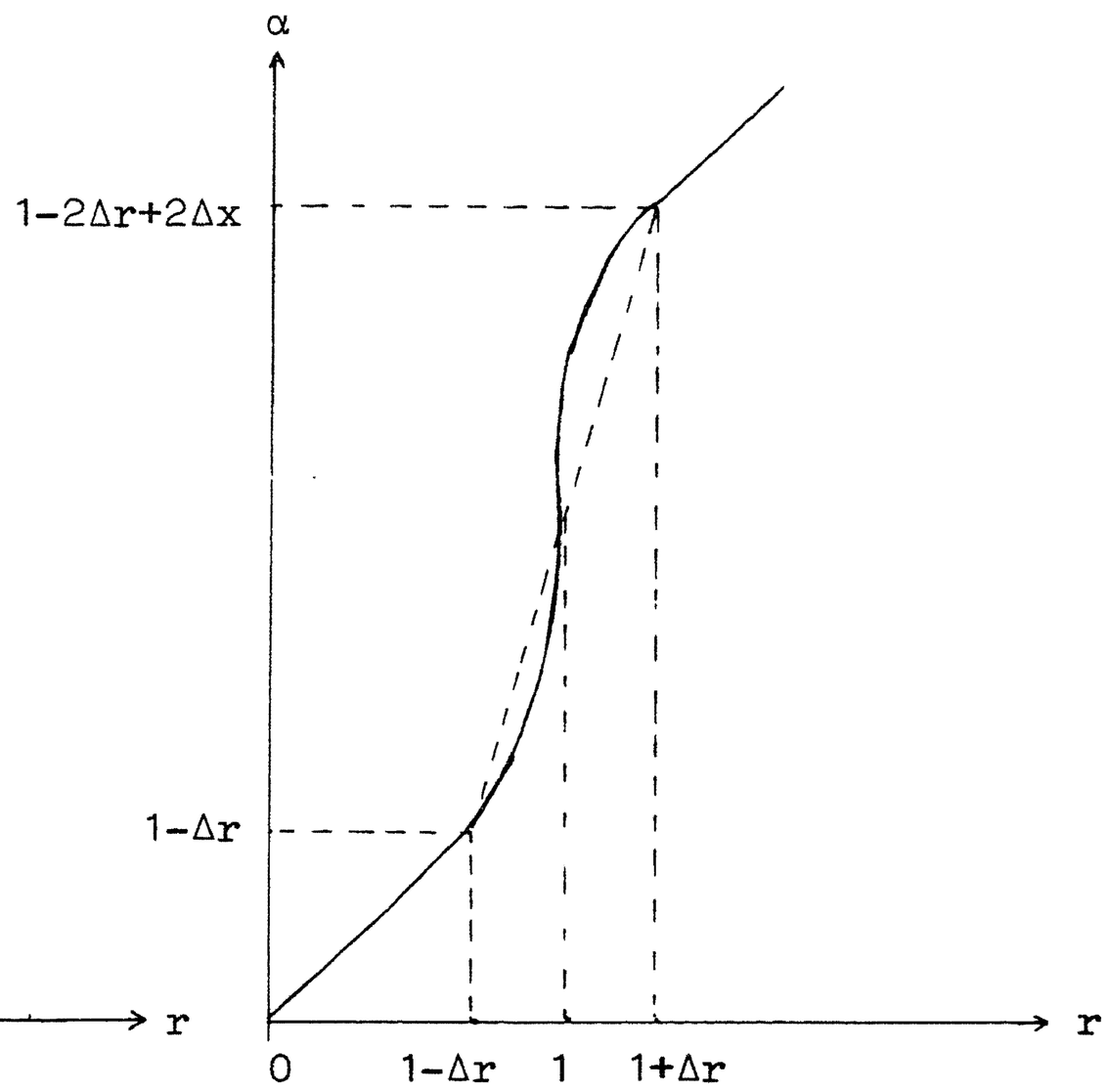

fig. 3.2 The transformation (3.1)

Next we consider the problem how Δr and Δx should be chosen. We shall use the criterion that the righthand side of (3.2) varies sufficiently slowly with x. There are many possibilities to derive conditions for Δx and Δr from this criterion. For example, we may require that

$$\left|\left|\frac{d}{dx}\left[x_r^2\, T_{xx}(r,0) + \left(x_{rr} + \frac{1}{r}\, x_r\right) T_x(r,0)\right]\right|\right| \leq a, \tag{3.3}$$

where $||\ ||$ denotes some norm in the space of functions defined at $1-\Delta'r \leq r \leq 1+\Delta'r$ and a measures the maximal allowed variation. Condition (3.3) can be written in the more simple form

$$(3.3') \qquad \left|\left|\frac{T_{rrr}+T_{rr}}{x_r}\right|\right| \leq a,$$

where we have neglected the coefficient $1/r$, i.e. we have put $1/r = 1$. In table 3.1 some values of

$$(3.4) \qquad \left|\left|\frac{T_{rrr}+T_{rr}}{x_r}\right|\right|_2 \equiv \int_{1-\Delta'r}^{1+\Delta'r} \left[\frac{T_{rrr}+T_{rr}}{x_r}\right]^2 dr$$

are given for the initial function (2.1) with $\Delta'r = .1$.

Table 3.1 Values of expression (3.4) divided by 10^3

Δx \ Δr	.1	.2	.3	.4	.5
.1	377				
.2	73	377			
.3	34	103	377		
.4	21	47	139	377	
.5	14	27	72	170	377
.6	11	17	44	96	194
.7	8	12	30	62	118
.8	7	9	21	43	79
.9	6	7	16	32	57
1.0	5	5	13	24	43

4. Discretization of the space variable

The next step is to replace the differential operator $A\partial^2/\partial x^2 + B\partial/\partial x$ in equation (3.2) by a difference operator defined at the grid points $j\xi$, $j = 0, 1, 2, \ldots$ the x-axis. For $j \geq 1$ we may use the operator

$$A_j \frac{X_+ - 2 + X_-}{\xi^2} + B_j \frac{X_+ - X_-}{2\xi}, \tag{4.1}$$

where A_j, B_j are the coefficients A, B evaluated at the point $x = x_j = j\xi$ and $X_\pm$ denote the shift operators over $\pm\xi$. At the point $x = 0$ we may use the operator (compare Saul'yev [1], p.77)

$$4 A_0 \frac{X_+ - 1}{\xi^2}. \tag{4.2}$$

Equation (3.2) reduces to an infinite set of ordinary, first order differential equations

$$\frac{dT}{dt} = DT, \tag{4.3}$$

where T now denotes a vector function of t, the components of which are the temperatures T at the gridpoints x_j. The matrix $D = (d_{i,j})$ is of tri-diagonal type, namely

$$D = \frac{1}{\xi^2} \begin{pmatrix} -4A_0 & 4A_0 & 0 & & \\ A_1 - \frac{1}{2}\xi B_1 & -2A_1 & A_1 + \frac{1}{2}\xi B_1 & 0 & \\ 0 & A_2 - \frac{1}{2}\xi B_2 & -2A_2 & A_2 + \frac{1}{2}\xi B_2 & 0 \\ & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \tag{4.4}$$

For stability considerations, some information about the eigenvalues of D is desirable. We have the following theorem.

Theorem 4.1

Let Δx, Δr and ξ be such that

$$A_j \geq \tfrac{1}{2}\xi|B_j|, \quad j = 1, 2, \ldots \tag{4.5}$$

Then the matrix D has its eigenvalues in the negative interval $[-\sigma(D),0]$, where the spectral radius $\sigma(D)$ satisfies the inequality

$$\sigma(D) \leq \frac{4}{\xi^2} \max_j \{2,(\alpha T_j+\beta)x_r^2\}. \tag{4.6}$$

Proof

If (4.5) is satisfied the off-diagonal elements of D are positive. From this property it follows that the tri-diagonal matrix has real eigenvalues (see Wilkinson [3], p. 335).

Furthermore, let $R_i = \sum_{\substack{j=0 \\ j\neq i}}^{\infty} |d_{i,j}|$. Then all the eigenvalues δ of D lie in the union of intervals

$$|\delta - d_{ii}| \leq R_i, \quad i = 0, 1, 2, \ldots .$$

From (4.4) and (4.5) it follows that

$$-8\frac{A_0}{\xi^2} \leq \delta \leq 0, \quad -4\frac{A_j}{\xi^2} \leq \delta \leq 0, \quad j = 1, 2, \ldots . \tag{4.7}$$

By using the definition of A_j inequality (4.6) easily follows.

5. Discretization of the time variable

There remains the problem of solving the set of ordinary differential equations (4.3). In Van der Houwen [2], p. 19 a method is given which is most appropriate for equations of this type, provided that the eigenvalues of D are non-positive. From theorem 4.1 it follows that D does have non-positive eigenvalues when condition (4.5) is satisfied. Let us assume that ξ is chosen so small that (4.5) is satisfied. Then we may use the difference scheme

$$T_{k+1} = C_n\left(1 + \frac{\tau D}{n^2}\right) T_k, \tag{5.1}$$

where τ is the time step, C_n is the Chebyshev polynomial of degree n, and T_k represents the temperature at $t = k\tau$.

To ensure stability the time step τ has to satisfy the condition (cf. [2], p. 22)

$$\tau \leq \frac{2n^2}{\sigma(D)} . \tag{5.2}$$

This condition is certainly satisfied when $\sigma(D)$ is replaced by the right hand side of relation (4.6), i.e.

$$\tau \leq \frac{n^2\xi^2}{2 \operatorname{Max}_j\{2,(\alpha T_j+\beta)x_r^2\}} \tag{5.2'}$$

We are now in a position to make a choice for Δx and Δr. On one hand, the right hand side of equation (3.2) is required to vary sufficiently slowly with x (see table 3.1), on the other hand, we wish to restrict the computation time. In order to get a rough idea of the computation time necessary to perform the integration in, let us say, a rectangle $[0,r_0] * [0,t_0]$, we compute the total number of gridpoints we have to consider (we assume that the computation time is roughly proportional to this number multiplied by the degree of the Chebyshev polynomial used). If we take a polynomial of degree n then at each new time step we have to add n points to the difference scheme. Therefore, the grid points necessary for the integration lie in a polygon as shown in figure 5.1. Here, the angle θ is defined by $\tan\theta = 1/n$.

For the time step τ we have taken the value

$$\tau = \frac{n^2\xi^2}{2\beta\left(\frac{2\Delta x-\Delta r}{\Delta r}\right)^2} , \tag{5.3}$$

which certainly satisfies relation (5.2'). Furthermore, we have chosen

$$t_0 = 1,\ r_0 = 2,\ \Delta r' = .1,\ \xi = .1,\ n = 10 \text{ and } \beta = 13.12\ ^{*)} \tag{5.4}$$

*) This value of β is of physical interest.

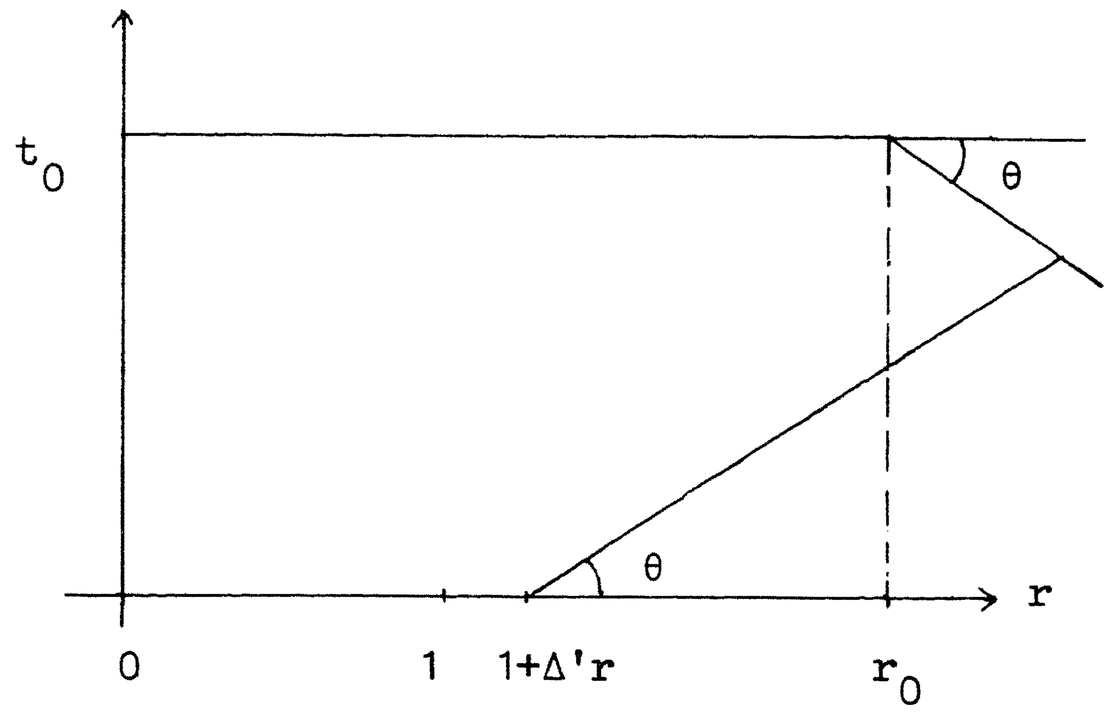

fig. 5.1 Grid points necessary for integration

For some values of Δr and Δx an estimate is given in table 5.1 of the number of grid points multiplied by $n \times 10^{-3}$.

Table 5.1 Estimates of computational labor

Δx \ Δr	.1	.2	.3	.4	.5
.2	10				
.3	29	5			
.4	60	11	3		
.5	103	20	6	3	
.6	160	32	11	5	2
.7	232	48	17	8	4
.8	321	67	25	11	6
.9	427	91	34	16	8
1.0	552	119	45	21	12

A comparison of table 3.1 and 5.1 reveals that in the domain of interest the lines of equal variation approximately correspond with lines of equal computation time. A series of numerical experiments has shown that

(5.5) $\Delta r = .2,\ \Delta x = .6$

yield satisfactory results for the special case (5.3), (5.4).

6. Actual computation scheme

In our actual calculations we have applied the difference scheme described in the preceding sections. The parameters have been chosen according to (5.4) and (5.5). We have used a variable time step given by

$$(6.1) \qquad \tau = \frac{n^2\xi^2}{2(\alpha T(0)+\beta)\left(\frac{2\Delta x-\Delta r}{\Delta r}\right)^2} = \frac{1}{50(\alpha T(0)+\beta)} ,$$

where α was chosen to be $-.263$.
This step is slightly larger than the one given by (5.3).
In addition, we have made two modifications, which turned out to save a considerable amount of computing time.

Firstly, instead of adding 10 points at each new level only those points have been added in which the temperature exceeded the value 10^{-12}. Numerical experiments showed that the results did not change much and that the effect of the above criterion resulted in only one or two new points at each level.

The second modification was the transition to a uniform grid on the r-axis as soon as the discontinuity introduced by the initial temperature has been smoothed out. The discontinuity was said to be smoothed out when

$$T(1-\Delta r) - T(1+\Delta r) \leq .2.$$

The transition to the new grid on the r-axis required the interpolation of the temperature at the new points in the interval $(1-\Delta r,\ 1+\Delta r)$, after which the original differential equation was integrated with a variable timestep given by

$$\tau = \frac{n^2\xi^2}{4(\alpha T(0)+\beta)} = \frac{1}{4(\alpha T(0)+\beta)} .$$

7. Numerical results

In the following tables the results are given, obtained by applying the method described in the preceding sections where the parameters r_0, $\Delta r'$, ξ, n, Δr, Δx and τ have the values specified in (5.4), (5.5) and (6.1). α and β have the values -.263 and .291 respectively.

r \ t	63	1.71	2.91	4.01	5.07	6.70
0	.99	.99	.91	.76	.62	.48
.1	.99	.99	.90	.75	.62	.48
.2	.99	.98	.88	.74	.62	.48
.3	.99	.97	.86	.72	.61	.47
.4	.99	.94	.82	.70	.59	.47
.5	.99	.90	.78	.67	.57	.45
.6	.98	.85	.74	.64	.56	.45
.7	.94	.79	.70	.62	.54	.43
.8	.87	.74	.64	.57	.50	.41
.872	.78	.66	.62	.56	.50	.41
.909	.72	.63	.60	.54	.49	.40
.936	.66	.60	.58	.53	.47	.40
.959	.63	.59	.56	.51	.47	.39
.980	.61	.58	.54	.50	.46	.39
1	.60	.57	.53	.49	.45	.38
1.020	.58	.55	.51	.48	.44	.38
1.041	.58	.54	.49	.46	.43	.37
1.064	.57	.53	.47	.45	.42	.36
1.091	.54	.51	.46	.43	.40	.35
1.128	.48	.48	.44	.42	.39	.34
1.2	.37	.41	.41	.40	.38	.33
1.3	.27	.35	.39	.38	.37	.33
1.4	.26	.32	.34	.34	.33	.31
1.5	.21	.30	.30	.31	.30	.29
1.6	.14	.24	.29	.30	.30	.28
1.7	.09	.20	.26	.28	.28	.27
1.8	.08	.18	.22	.24	.25	.25
1.9	.06	.16	.19	.22	.23	.23
2	.04	.13	.18	.20	.22	.22

r \ t	7.54	8.60	9.83	10.70	12.37
0	.43	.36	.31	.29	.25
.1	.42	.36	.31	.29	.25
.2	.42	.36	.31	.29	.24
.3	.42	.36	.31	.28	.24
.4	.41	.35	.31	.28	.24
.5	.40	.35	.31	.28	.24
.6	.40	.35	.30	.28	.24
.7	.38	.34	.30	.27	.23
.8	.37	.33	.29	.27	.23
.9	.36	.32	.28	.26	.23
1	.35	.31	.28	.26	.22
1.1	.33	.30	.27	.25	.22
1.2	.32	.29	.26	.24	.21
1.3	.31	.28	.25	.24	.21
1.4	.29	.27	.25	.23	.20
1.5	.27	.26	.24	.22	.20
1.6	.27	.25	.23	.22	.19
1.7	.26	.24	.22	.21	.19
1.8	.24	.23	.21	.20	.18
1.9	.22	.21	.20	.19	.18
2	.21	.20	.19	.19	.17

8. Description of the program

In procedure matrixcoeff the coefficients occurring in (4.4) are computed. This procedure also determines the points of the non-uniform grid with the corresponding initial values of u in the interval $(1-\Delta r, 1+\Delta r)$.

Procedures D and D2 transform a vector $\vec{c}$ into $D\vec{c}$ where D is the matrix given in (4.4).
D is used for a non-uniform grid and D2 for a uniform grid.
We now give the complete ALGOL 60 program.

```
begin comment fom-diffusion-problem;
    integer deg, g, I, i, j, k, k1, k2, l, m, n, o, p, q, s, t, t1;
    real a, alfa, b, beta, delta, dr, dr1, dr2, dx, dx1, inx, irho, pi,
    r, rho, rho2, sigma, T, tau, tau1, itau, tyd, tyd1, u0, u1, u2,
    v, x, y, z;

    procedure matrixcoeff(k, l, p, r, ri, xr2, xrr, u); value k, l;
    integer k, l; array ri, xr2, xrr, u;
    begin integer i, m;
        real b, e, f, x, y;
        boolean yes;
        f:= 0; m:= k + l + l; b:= pi / (2 × dr1); e:= pi / dr;
        yes:= false;
        for i:= 1 step 1 until l do
        begin f:= f + rho; r:= 1 - dr; x:= 1 + dr;
            zeroin(r, x, - sin((r - 1 + dr) × e) × v / pi + (dx × r -
            (1 - dr) × v) / dr - 1 + dr - f, 8 - 11);
            x:= - cos((r - 1 + dr) × e) × v + dx;
            y:= sin((r - 1 + dr) × e) × v × e; ri[k + i]:= r;
            ri[m - i]:= 2 - r; xr2[k + i]:= xr2[m - i]:= x × x;
            xrr[k + i]:= x / r + y; xrr[m - i]:= x / (2 - r) - y;
            if r > 1 - dr1 ∧ ¬yes then
            begin yes:= true; p:= k + i - 1 end;
            if yes then
            begin u[k + i]:= a:= (cos((r - 1 + dr1) × b) + 1) × .5;
                u[m - i]:= 1 - a
            end;
            xr2[k]:= xr2[m]:= dr × dr; xrr[k]:= dr / (1 - dr);
            xrr[m]:= dr / (1 + dr); ri[k]:= 1 - dr; ri[m]:= 1 + dr
    end matrixcoeff;

    procedure D(k, l, m, c, u, xr2, xrr); value k, l, m; integer k, l, m;
    array c, u, xr2, xrr;
    begin integer i, l1, m1;
        real cimin1, ci, ciplus1;
        ci:= c[0]; ciplus1:= c[1];
        c[0]:= ( - ci + ciplus1) × (alfa × u[0] + beta) × 4 / rho2;
        for i:= 1 step 1 until k do
        begin cimin1:= ci; ci:= ciplus1; ciplus1:= c[i + 1];
            c[i]:= ((ciplus1 - cimin1) / (i + i) + ciplus1 - 2 × ci +
            cimin1) × (alfa × u[i] + beta) / rho2
        end;
        l1:= k + l - 1;
        for i:= k + 1 step 1 until l1 do
        begin cimin1:= ci; ci:= ciplus1; ciplus1:= c[i + 1];
            c[i]:= ((ciplus1 - 2 × ci + cimin1) × xr2[i] / dr2 +
            (ciplus1 - cimin1) × xrr[i] / 2) × (alfa × u[i] + beta) /
            dr2
        end;
        m1:= k + l + m - 1;
        for i:= k + l step 1 until m1 do
        begin cimin1:= ci; ci:= ciplus1; ciplus1:= c[i + 1];
```

```
        c[i]:= ((ciplus1 - cimin1) / (i + i) + ciplus1 - 2 × ci +
        cimin1) × (alfa × u[i] + beta) / rho2
    end
end D;

procedure D2(k, c, u); value k; integer k; array c, u;
begin integer i;
    real cimin1, ci, ciplus1;
    ci:= c[0]; ciplus1:= c[1];
    c[0]:= ( - ci + ciplus1) × (alfa × u[0] + beta) × 4 / rho2;
    for i:= 1 step 1 until k do
    begin cimin1:= ci; ci:= ciplus1; ciplus1:= c[i + 1];
        c[i]:= ((ciplus1 - cimin1) / (i + i) + ciplus1 - 2 × ci +
        cimin1) × (alfa × u[i] + beta) / rho2
    end
end;

deg:= READ; PRINTTEXT(ϟ deg =ϟ); ABSFIXT(3, 0, deg); NLCR;
begin array B[1:deg];
    for i:= 1 step 1 until deg do B[i]:= READ;
    pi:= 3.14159 26535 89793; tyd:= time;
    for alfa:= READ while alfa ≠ ₁₀5 do
    begin PRINTTEXT(ϟalfa =ϟ); FIXT(3, 6, alfa); beta:= READ;
        PRINTTEXT(ϟ    beta =ϟ); FIXT(3, 6, beta); NLCR; rho:= READ;
        PRINTTEXT(ϟ rho =ϟ); ABSFIXT(3, 6, rho); NLCR; dr:= READ;
        PRINTTEXT(ϟ  dr =ϟ); ABSFIXT(3, 6, dr); dx:= READ;
        PRINTTEXT(ϟ      dx =ϟ); ABSFIXT(3, 6, dx); NLCR;
        inx:= READ; PRINTTEXT(ϟ inx =ϟ); ABSFIXT(3, 6, inx); NLCR;
        dr1:= READ; PRINTTEXT(ϟ dr1 =ϟ); ABSFIXT(3, 6, dr1);
        T:= READ; PRINTTEXT(ϟ       T =ϟ); ABSFIXT(3, 2, T);
        CARRIAGE(3); rho2:= rho × rho; v:= dx - dr; dr2:= dr × rho;
        k:= entier((1 - dr + ₁₀ - 12) / rho);
        o:= entier((dx + ₁₀ - 12) / rho); l:= o + o;
        m:= entier((inx - dr - 1 + ₁₀ - 12) / rho); n:= k + l;
        s:= (available - 6 × l - 200) : 6;
        begin array u, U, C[0:s], ri, xr2, xrr[k:n];
            matrixcoeff(k, o, k1, r, ri, xr2, xrr, u);
            k2:= n + k - k1;
            for i:= 0 step 1 until k1 do u[i]:= 1;
            for i:= k2 step 1 until s do u[i]:= 0;
            for i:= 0 step 1 until l do
            begin SPACE(4); FIXT(2, 12, ri[k + i]);
                FIXT(2, 12, u[k + i]); FIXT(2, 12, xr2[k + i]);
                FIXT(2, 12, xrr[k + i])
            end;
            CARRIAGE(6); t:= n; x:= deg × deg × rho × rho;
            z:= ((2 × dx - dr) / dr) ↑ 2;
            y:= x / ((if z > 2 then z else 2) × 2); tau:= y / beta;
            itau:= tau; s:= m + n;
            for i:= 1 step 1 until i + 1 do
            begin t1:= t; t:= t + deg; tau1:= 1;
                for j:= 0 step 1 until t do C[j]:= U[j]:= u[j];
                for I:= 1 step 1 until deg do
                begin D(k, l, t - n, C, U, xr2, xrr);
```

```
    tau1:= tau1 × tau;
    for j:= t step - 1 until 0 do u[j]:= tau1 × B[I] ×
    C[j] + u[j]
end PI(tauD)Uk;
PRINTTEXT(‹t =›); ABSFIXT(2, 5, itau); CARRIAGE(2);
u0:= u[0]; FIXT(3, 12, u0); NLCR;
for j:= 1 step 1 until s do FIXT(3, 12, u[j]);
for j:= t1 step 1 until t do
begin a:= u[j]; if abs(a) < 10 - 12 then
    begin t:= j; goto fin end
    end;
fin: CARRIAGE(2); if u0 < T then goto end; a:= u[k];
    b:= u[n]; if (a - b) / (dr + dr) < .2 then goto uss;
    tau:= y / (alfa × b + beta); itau:= itau + tau
end;
uss: PRINTTEXT(‹uniform space step›); CARRIAGE(2);
    p:= entier((2 × dr) / rho + 10 - 12) - 1; g:= 1;
    irho:= 1 - dr + rho; x:= 1 - dr; u1:= u[k];
    for i:= 1 step 1 until p do
    begin for j:= g step 1 until 1 do
        begin y:= ri[k + j]; u2:= u[k + j]; if y > irho then
            begin u[k + i]:= (u2 - u1) × (irho - x) / (y - x)
                + u1; x:= y; g:= j + 1; irho:= irho + rho;
                u1:= u2; goto again
            end;
            x:= y; u1:= u2
        end;
again:
end;
m:= 1 - p - 1;
for i:= n step 1 until t do u[i - m]:= u[i];
for i:= t - m + 1 step 1 until t do u[i]:= 0; t:= t - m;
delta:= - beta / (alfa + alfa);
x:= deg × deg × rho × rho;
tau:= if u0 < delta then x / ((alfa × u0 + beta) × 4)
else x / (beta + beta); FIXT(3, 12, u0); NLCR;
s:= entier((inx + 10 - 12) / rho) - 1;
for i:= 1 step 1 until s do FIXT(3, 12, u[i]);
u2:= u[s + 1]; FIXT(3, 12, u2); CARRIAGE(2);
sigma:= .025 / (u0 - u2);
tau:= if tau < sigma then tau else sigma;
itau:= itau + tau; PRINTTEXT(‹ taunew =›);
ABSFIXT(2, 5, tau); CARRIAGE(3);
for i:= 1 step 1 until 1000 do
begin t1:= t; t:= t + deg; tau1:= 1;
    for j:= 0 step 1 until t do C[j]:= U[j]:= u[j];
    for I:= 1 step 1 until deg do
    begin D2(t - 1, C, U); tau1:= tau1 × tau;
        for j:= t step - 1 until 0 do u[j]:= tau1 × B[I] ×
        C[j] + u[j]
    end PI(taunewD2)Uk;
    PRINTTEXT(‹t =›); ABSFIXT(2, 5, itau); CARRIAGE(2);
    u0:= u[0]; FIXT(3, 12, u0); NLCR;
    for j:= 1 step 1 until s do FIXT(3, 12, u[j]);
    u2:= u[s + 1]; FIXT(3, 12, u2);
```

```
    for j:= t1 + 1 step 1 until t do
    begin a:= u[j]; if abs(a) < 10 - 12 then
        begin t:= j; goto fin2 end
    end;
fin2: CARRIAGE(2); if u0 < T then goto end;
    tau:= if u0 < delta then x / ((alfa × u0 + beta) × 4)
    else x / (beta + beta); sigma:= .025 / (u0 - u2);
    tau:= if tau < sigma then tau else sigma;
                itau:= itau + tau
            end;
        end: tyd1:= time; CARRIAGE(2); PRINTTEXT(‹rekentyd =›);
            ABSFIXT(3, 2, tyd1 - tyd); PRINTTEXT(‹sec.›); tyd:= tyd1;
            NEW PAGE
        end
    end
  end
end
```

References

[1] Saul'yev, V.K., Integration of equations of parabolic type by the method of nets, Pergamon Press, Oxford, 1964.

[2] Van der Houwen, P.J., One step methods for linear initial value problems I, Report TW 119, Mathematical Centre, Amsterdam, 1970.

[3] Wilkinson, J.H., The algebraic eigenvalue problem, Clarendon Press, Oxford, 1965.