stichting
mathematisch
centrum

MC

AFDELING TOEGEPASTE WISKUNDE TW 120/70 APRIL

G.J.R. FÖRCH

NUMERICAL TREATMENT OF A TWO POINT BOUNDARY-VALUE PROBLEM

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.

1. Introduction

In this paper the following nonlinear ordinary-differential equation, arising in the theory of conduction of heat is discussed.

$$(1.1) \qquad -2y\,\frac{d\theta}{dy} = \frac{d}{dy}\left(D\,\frac{d\theta}{dy}\right) ,$$

where D is a function of θ.

Some methods for numerical treatment are given. The results are compared with those given by Crank [1], where the equation is reduced to the integral form

$$(1.2) \qquad \theta = 1 - \frac{\int_0^y (1/D)\ \exp\{-\int_0^y (2\tau/D)dt\}d_\xi}{\int_0^\infty (1/D)\ \exp\{-\int_0^y (2\ /D)d\tau\}d_\xi} .$$

which is then solved in an iterative manner.
Recently, Hays and Curds worked on the same problem. Their method employed variational calculus which, however, led to eleborate calculations.
Since Crank as well as Hays and Curds used numerical calculations only at the final stage of their approach, it seemed attractive to solve the problem by a direct numerical treatment.
In our numerical method we use second-order difference schemes as approximations of the differential equation (1.1).

In section 2 the problem is formulated with a brief indication of its background.
In section 3 the differential equation is subjected to some transformations, in order to facilitate the subsequent numerical treatment.
In sections 4, 5, 6, 7 and 8 various numerical methods are discussed.
Section 9 contains a discussion of the results. At the end of the paper the final numerical results are given and the ALGOL 60 programs used are reproduced.

2. Statement of the problem.

When, in diffusionproblems, one deals with variable diffusion-coefficients it is sometimes possible to transform the partial-differential equation

$$\frac{\partial\theta}{\partial t} = \frac{\partial}{\partial x} \left(D(\theta) \frac{\partial\theta}{\partial x}\right) , \tag{2.1}$$

into an ordinary-differential equation.
If the boundary conditions can be written as conditions involving the new variable

$$y = \frac{1}{2} x t^{-\frac{1}{2}} , \tag{2.2}$$

then it can be proved, that the solution of (2.1) is a function of y only.
The introduction of y transforms equation (2.1) into (see [1]):

$$2y \frac{d\theta}{dy} + \frac{d}{dy} \left(D(\theta) \frac{d\theta}{dy}\right) = 0. \tag{2.3}$$

Transformation (2.2) is known the Boltzmann transformation.

In this paper we consider the special case where D is a linear function of θ.
In particular, we consider the problem

$$2y \frac{d\theta}{dy} + \frac{d}{dy} \left((1+\sigma\theta) \frac{d\theta}{dy}\right) = 0 , \quad \sigma > 0, \tag{2.4}$$

with boundary conditions

$$\begin{aligned} \theta &= 1 \quad \text{for } y = 0 , \\ \theta &\to 0 \quad \text{for } y \to \infty . \end{aligned} \tag{2.5}$$

3. Transformations of the differential equation

By means of the transformation

$$U = \theta + \frac{1}{2} \sigma \theta^2 \tag{3.1}$$

(2.4) and (2.5) get the following, somewhat simpler, form

$$(3.2)\qquad \begin{cases} \dfrac{d^2U}{dy^2} + A(U,y)\,\dfrac{dU}{dy} = 0 \quad , \\ U = 1 + \dfrac{1}{2}\sigma \qquad \text{for } y = 0 \quad , \\ U \to 0 \qquad \text{for } y \to \infty \quad , \end{cases}$$

where $A(U,y) = 2y(1+2\sigma U)^{-\frac{1}{2}}$.

In order to facilitate the numerical treatment, it is convenient to transform the semi-infinite interval $0 \leq y < \infty$ into a finite interval, e.g. $0 \leq z < 1$. At the same time it is desirable to eliminate the singularity at $y = \infty$ of the equation (3.2).
Thus we are looking for a transformation

$$(3.3)\qquad z = f(y) \quad ,$$

which satisfies the following conditions
a. $f(0) = 0$ and $f(\infty) = 1$

b. The transformed version of equation (3.2)

$$(3.4)\qquad (f')^2 \frac{d^2v}{d\,z^2} + (f''+Af')\frac{dv}{dz} = 0 \quad ,$$

where $v(z) = U(y(z))$,
has no singularity for $0 \leq z \leq 1$.

A suitable transformation is furnished by

$$(3.5)\qquad z = \frac{2}{\sqrt{\pi}} \int_0^y \exp(-\tau^2)d\tau \equiv \operatorname{erf} y$$

It is obvious that condition a is satisfied.

By means of (3.5) the problem (3.2) is transformed into

$$(3.6)\qquad \begin{cases} \dfrac{d^2V}{d\,z^2} + B(U,y)\ \dfrac{dV}{dz} = 0 \quad , \\ V = 1 + \frac{1}{2}\sigma \qquad \text{for } z = 0 \quad , \\ V = 0 \qquad\qquad \text{for } z = 1 \quad , \end{cases}$$

where $B(U,y) = \sqrt{\pi}\ \ y\ \ \exp(y^2)((1+2\sigma U(y))^{-\frac{1}{2}} - 1)$,

and $V(z) = U(y(z))$.

We shall show that $B(U,y)$ is non singular at $y = \infty$, or $z = 1$, by using the asymptotic behaviour of $U(y)$ for $y \to \infty$.

When $y \to \infty$ the solution $U(y)$ of (3.2) will be very small. For these small values of $U(y)$ the equation (3.2) and the second boundary condition become, approximately,

$$(3.2a)\qquad \begin{cases} \dfrac{d^2U}{d\,y^2} + 2y\ \dfrac{dU}{dy} = 0 \quad , \\ U \to 0 \qquad \text{for } y \to \infty \ . \end{cases}$$

The general solution of this problem is $U(y) = a(1 - \operatorname{erf} y) = a \operatorname{erfc} y$, where a is an arbitrary constant. Substitution of this result into the expression for $B(U,y)$ yields with $y \to \infty$ the following limit.

$$\begin{aligned} \lim_{y\to\infty} B(U,y) &= \lim_{y\to\infty} \sqrt{\pi}\ y\ \exp(y^2)((1+2\sigma U(y))^{-\frac{1}{2}} - 1) \quad , \\ &= \lim_{y\to\infty} \ (-\sqrt{\pi}\ y\ \exp(y^2)\sigma\ U(y)) \\ &= \lim_{y\to\infty} \ (-\sqrt{\pi}\ a\ y\ \exp(y^2)\sigma\ \operatorname{erfc}(y)) \\ &= -a\ \sigma \end{aligned}$$

Hence,

$$(3.7)\qquad \lim_{y\to\infty} B(U,y) = -a\ \sigma$$

which shows that condition b is satisfied.

4. A numerical treatment

For the numerical treatment, the following three methods will be used.

A The two-point boundary-value problem (3.6) is solved by an iterative method.
A procedure for finding a sufficiently accurate initial approximation in order to save computation time, will be mentioned.

B The boundary-value problem (2.4), (2.5) is solved by an iterative step-by-step method.

C The boundary-value problem (3.2) is solved by the same iterative step-by-step method.

5. Method A

The boundary-value problem (3.6) will be solved by an iterative method.

A simple difference scheme for (3.6) is given by

$$(5.1)\quad \begin{cases} R\, v(j) \equiv \dfrac{v(j+1) - 2v(j) + v(j-1)}{h^2} + b(j)\, \dfrac{v(j+1) - v(j-1)}{2h} = 0, \\ \qquad j = 1, 2, \ldots, N-1\ ; \\ v(0) = 1 + \frac{1}{2}\sigma\ , \\ v(N) = 0\ , \end{cases}$$

where $v(j)$ and $b(j)$ are net functions defined in the points $z = jh$, $j = 0, 1, \ldots, N (h = \frac{1}{N})$, satisfying the relations $v(j) = V(jh)$ and $b(j) = B(jh)$.

N.B. We shall use small letters for net functions and capital letters for ordinary functions.

Let $\tilde{V}(z)$ be the analytical solution of boundary-value problem (3.6) then the error $e(j)$ of the numerical solution,

$$(5.2)\qquad e(j) = \tilde{v}(j) - v(j)\ ,$$

satisfies the following linear difference scheme:

$$(5.3)\quad \begin{cases} \dfrac{e(j+1) - 2e(j) + e(j-1)}{h^2} + b(j)\,\dfrac{e(j+1) - e(j-1)}{2h} \\ + \tilde{c}(j)\, e(j) = O(h^2)\ , \quad j = 1, \ldots, N-1\ ; \\ e(0) = e(N) = 0. \end{cases}$$

The net function $\tilde{c}(j)$ is defined by:

$$\tilde{c}(j) = -\sqrt{\pi}\ \sigma\ y(j)\ \exp(y(j))^2\ (1 + 2\sigma\ \tilde{v}(j))^{-\frac{3}{2}}\ \tilde{v}_z^2(j)$$

where $y(j)$ denotes the value of y that corresponds with $z = jh$ and where $\tilde{v}_z(j)$ represents the derivative of $\tilde{v}$ at the j^{th} net point.

Difference scheme (5.3) approximates the boundary-value problem

$$(5.4)\quad \begin{cases} \dfrac{d^2E}{d\,z^2} + B(V,y)\,\dfrac{dE}{dz} + \tilde{C}(V,y)\,E = 0\ , \\ E(0) = E(1) = 0\ , \end{cases}$$

within an error of order h^2.

We shall show that boundary-value problem (5.4) only allows the trivial solution $E(z) \equiv 0$.
It is easily checked that the function $\tilde{C}(V,y)$ is negative all over the interval $0 \leq z \leq 1$, for all positive values of V.
Suppose there is a solution $E_0(z)$ of (5.4) with $E_0(z) \not\equiv 0$, then there must be at least one region of the interval $0 \leq z \leq 1$ where $E_0(z)$ is either positive or negative.
If $E_0(z)$ assumes positive values, then there must be a point $z = z_0$, $0 < z_0 < 1$, with:

$$(5.5)\quad \begin{cases} E_0(z_0) > 0\ , \\ \left.\dfrac{dE_0}{dz}\right|_{z_0} = 0\ , \\ \left.\dfrac{d^2E_0}{d\,z^2}\right|_{z_0} < 0\ , \end{cases}$$

which contradicts (5.4).
Analogously, we can prove that $E_0(z)$ cannot assume negative values. Thus boundary-value problem (5.4) only allows the solution $E(z) \equiv 0$. From this we can conclude, that, if (5.3) is a stable difference scheme, then we have (see [2]) $e(j) = O(h^2)$.

A sufficient condition for the stability of (5.3) may be obtained when the roots λ_i, i = 1, 2, of the characteristic equation of (5.3) satisfy the inequality (see [2])

(5.6) $$|\lambda_i| \leq 1 + O(h) \quad , \quad i = 1, 2 .$$

The characteristic equation of (3.3) is given by:

(5.7) $$(2 + h\, b(j))\lambda^2 - (4 - 2\, h^2\, \tilde{c}(j))\lambda + (2 - h\, b(j)) = 0.$$

From (3.7) it follows that $h\, b(j) = O(h)$. One can proof that $h^2\, \tilde{c}(j) = O(h^2)$.
So the roots of equation (5.7) satisfy inequality (5.6).

The preceding considerations show that a function v(j), satisfying the difference boundary-value problem (5.1) is a second-order approximation to the analytical solution v(z) of boundary-value problem (3.6).

There remains the construction of the function v(j). We shall use the following iterative process:

(5.8) $$\begin{cases} v_{k+1}(j) = v_k(j) + \tau\, R_k\, v_k(j) & j = 1, 2, \ldots, N-1 \\ & k = 0, 1, \ldots \\ v_k(0) = 1 + \frac{1}{2}\sigma \quad , & \\ v_k(N) = 0 \quad , & \end{cases}$$

where $v_0(j)$ is an arbitrary initial approxi... paramter.

If this iterative process converges for $k \to \infty$, then the limit function $v_\infty(j)$ obviously satisfies (5.1).
We now investigate the stability of (5.8).
Let $v_0(j)$ satisfy the boundary conditions and let $v_k(j)$ be, for $j = 1, \ldots, N-1$, the components of a vector, then (5.8) may be written in matrixform:

(5.8a) $$v_{k+1} = M_k \, v_k \quad ,$$

where M_k is defined by

$$\begin{pmatrix} 1, & 0, & \cdots & & & & & & 0 \\ & \ddots & & & & & & & \\ 0 \cdots 0, & \frac{1}{2}\frac{\tau}{h^2}(2-h\,b_k(j)), & 1-\frac{2\tau}{h^2}, & \frac{1}{2}\frac{\tau}{h^2}(2+h\,b_k(j)), & 0, & \ldots & 0 \\ & & & & & & \ddots & & \\ 0, & 0, & \cdots & & & & & 0, & 1 \end{pmatrix}$$

For stability we require that

(5.9) $$||M_k|| \leq 1 \quad ,$$

for every k.
We shall use the maximumnorm for matrices,
i.e. $||A|| = \max\limits_j \sum\limits_i |a_{ij}|$.

Using this norm for M_k we obtain the following stability conditions

(5.10) $$\tau \leq \frac{1}{2} h^2$$

and

(5.11) $$h \leq 2/||b_k(j)||_{max} \quad ,$$

$b_k(j)||_{max}$ is defined as the maximum value of $b_k(j)$, $j = 1, 2, \ldots N-1$.

When we choose a positive approximation $v_0(j)$ then the functions $v_k(j)$ will also be positive, and then, from the definition of $b_k(j)$, we see that condition (5.11) is certainly satisfied when:

$$(5.11a) \qquad h \leq \frac{\exp(-y^2(N-1)}{y(N-1)} \quad .$$

N.B. Here y(N-1) denotes the value of y in the $(N-1)^{st}$ lattice point.

As h = erfc y(N-1), it is easy to see that condition (5.11a) is no real restriction for practical purposes.

The method, described above, is convergent when the stability conditions are satisfied; however, the rate of convergence is very small. Therefore, it is very important to choose a good initial approximation. We can find this good approximation with the following procedure. In (3.6) we replace the boundary condition v(N) = 0, by a second initial condition, $v(1) = \alpha$. The initial value problem, obtained in this way, can now be solved directly, using difference scheme (5.1). The stability of this scheme is related to the stability of difference-scheme (5.3) (see [2]), which is stable, as has been proved. Of course, the application of scheme (5.1), with initial conditions $v(0) = 1 + \frac{1}{2}\sigma$, $v(1) = \alpha$ $(0 < \alpha < 1+ \frac{1}{2}\sigma)$, will, in general, not lead to v(N) = 0. But by changing α, we can make $|v(N)|$ as small as desired, i.e. for some $\varepsilon > 0$, $|v(N)| < \varepsilon$.
From preceding considerations it follows, that the function v(j) obtained in this way, approximates within an error of order h^2, the solution $\tilde{v}(z)$ of equation (2.6), with boundary conditions $\tilde{v}(0) = 1 + \frac{1}{2}\sigma$, $\tilde{V}(1) = \varepsilon$.
We now define the initial approximation $v_0(j)$ in the following way:

$$\begin{cases} v_0(j) = v(j) & \text{for} \quad j = 0, 1, \ldots, N-1 \\ v_0(N) = 0 \end{cases}$$

with this initial approximation $v_0(j)$ the computation time is reduced considerably.

A shooting procedure, like this one, can only be used if all solutions of the differential equation in consideration are bounded in the relevant region.
In our case this means that all solutions of

$$(5.11) \qquad \frac{d^2v}{dz^2} + B(U,y)\,\frac{dv}{dz} = 0,$$

must be bounded in the interval $0 \leq z \leq 1$.
But $B(U,y)$ is non-singular in $0 \leq z \leq 1$, as was proved previously.
Thus (5.11) is an equation, defined on a bounded region, with bounded coefficients and with constant coefficient for the derivative of highest degree.
Hence all solutions of (5.11) are bounded.

6. Method B

We now discuss method B for solving the problem numerically.
Consider the original boundary-value problem (2.4), (2.5).
We will use the following consistent difference approximation

$$(6.1) \qquad \begin{cases} \dfrac{2y(j)\,(\theta(j+1) - \theta(j-1))}{2h} + \sigma\left(\dfrac{\theta(j+1) - \theta(j-1)}{2h}\right)^2 \\[2ex] \quad + \dfrac{(1+\sigma\,\theta(j))\,(\theta(j+1) - 2\theta(j) + \theta(j-1))}{h^2} = 0\,, \\[2ex] \theta(0) = 1\,, \\[1ex] \theta(\infty) = 0\,. \end{cases}$$

Now the boundary condition at infinity is replaced by an additional condition at $y = 0$ (analogously to the method used to reduce the computation time in section 5).
From analytical considerations we know that in a suitably chosen finite point, e.g. $y = c$, the value of $|\theta|$ and the value of the derivative $|\frac{d\theta}{dy}|$, for the anlytical solution are both very small, i.e. for some $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$, $|\theta| < \varepsilon_1$ and $|\frac{d\theta}{dy}| < \varepsilon_2$.

Now our computation procedure is as follows:
We start the computation with $\theta(0) = 1$ and $\theta(1) = \alpha$ $(0 < \alpha < 1)$; we calculate the solution over the interval $[0, c]$; then at $y = c$ we check the values of $|\theta|$ and $|\frac{d\theta}{dy}|$; if either $|\theta|$ or $|\frac{d\theta}{dy}|$ is not sufficiently small, we change α and repeat the computation. In practice we discontinue the computation and increase α, as soon as θ becomes negative (see the ALGOL 60 programm at the end of the paper).

In order to apply this method successfully, the difference scheme (6.1) has to be stable. This stability is related to the stability of the first variation of (6.1) (see [2]).
I.e., if we replace $\theta(j)$ in (6.1) by $\theta(j) + e(j)$, then we obtain the following linear difference scheme for $e(j)$.

$$(6.2)\quad \begin{cases} \dfrac{2\,y(j)\,(e(j+1) - e(j-1))}{2h} + \sigma\,\dfrac{e(j)\,(\theta(j+1) - 2\theta(j) + \theta(j-1))}{h^2} \\[2ex] + \dfrac{(1 + \sigma\,\theta(j))\,(e(j+1) - 2e(j) + e(j-1))}{h^2} \\[2ex] + 2\,\sigma\,\left(\dfrac{\theta(j+1) - \theta(j-1)}{2h}\right)\left(\dfrac{e(j+1) - e(j-1)}{2h}\right) = O(h^2) \end{cases}$$

stability of (6.2) is guaranteed if the roots λ_i, $i = 1, 2$, of the characteristic equation satisfy the inequality:

$$|\lambda_i| < 1 + O(h)\ , \quad i = 1, 2.$$

The characteristic equation is given by:

$$(y(j)h + 1 + \sigma\,\theta(j) + \tfrac{1}{2}\sigma\,\theta(j+1) - \tfrac{1}{2}\sigma\,\theta(j-1))\lambda^2$$
$$+ (\sigma\,\theta(j+1) - 2\,\sigma\,\theta(j) + \sigma\,\theta(j-1) - 2 - 2\,\sigma\,\theta(j))\lambda$$
$$+ (-\,y(j)h + 1 + \sigma\,\theta(j) - \tfrac{1}{2}\sigma\,\theta(j+1) + \tfrac{1}{2}\sigma\,\theta(j-1)) = 0$$

which can be written as

$$(1 + \sigma\,\theta(j) + O(h))\lambda^2 - 2(1 + \sigma\,\theta(j) + O(h))\lambda$$
$$+ (1 + \sigma\,\theta(j) + O(h)) = 0.$$

This implies that

$$|\lambda_i| < 1 + O(h) \quad , \quad i = 1, 2 \quad .$$

Hence, difference scheme (3.9) is stable.
Using a conjecture of J. von Neumann (see [2]). We conclude that (3.8) is also stable.

7. Method C

Exactly the same line of working as was used in Method B, may be applied in order to solve boundary-value problem (3.2).
Now the following difference approximation can be used:

$$(7.1) \qquad \frac{u(j+1) - 2u(j) + u(j-1)}{h^2} + a(u(j),j) \frac{u(j+1) - u(j-1)}{2h} = 0.$$

To investigate stability, again we replace u(j) by u(j) + e(j), obtaining a linear difference scheme for e(j)

$$(7.2) \qquad \frac{e(j+1) - 2e(j) + e(j-1)}{h^2} + \frac{a(u(j),j)\ (e(j+1) - e(j-1))}{2h} + p(j)\ e(j) = 0,$$

where

$$(7.3) \qquad p(j) = -2y(j)\sigma \left(\frac{u(j+1) - u(j-1)}{2h}\right) (1 + 2\sigma\ u(j))^{-\frac{3}{2}} .$$

The characteristic equation of (7.2) is

$$(1 + \tfrac{1}{2}\ a(u(j),j)h)\lambda^2 - 2(1 - p(j)h^2)\lambda + (1 - \tfrac{1}{2}\ a(u(j),j)h) = 0$$

or

$$(1 + O(h))\lambda^2 - 2(1 + O(h))\lambda + (1 + O(h)) = 0.$$

Thus $|\lambda_i| < 1 + O(h), \quad i = 1, 2.$

Therefore scheme (7.2) is stable.

8. Comparison of Method B and Method C

It is interesting to compare the difference schemes (6.1) and (7.1), which were used in Method B and Method C, respectively. For reasons of convenience we shall introduce the following notations. We denote the original differential equation (2.4) by

$$(8.1) \qquad P\,\tilde{\theta} = 0 \ ,$$

difference scheme (6.1) by

$$(8.2) \qquad R_1\,\theta(j) = 0,$$

and difference scheme (7.1) by

$$(8.3) \qquad R_2\,u(j) = 0.$$

According to formula (3.1) we replace in (7.1) u by $\theta + \frac{1}{2}\sigma\,\theta^2$, yielding

$$\frac{\theta(j+1) + \frac{1}{2}\sigma\,\theta^2(j+1) - 2\theta(j) - \sigma\,\theta^2(j) + \theta(j-1) + \frac{1}{2}\sigma\,\theta^2(j-1)}{h^2}$$

$$+ \frac{2y}{1+\sigma\,\theta(j)} \quad \frac{\theta(j+1) + \frac{1}{2}\sigma\,\theta^2(j+1) - \theta(j-1) - \frac{1}{2}\sigma\,\theta^2(j-1)}{2h} = 0 \ ,$$

or after some rearrangements

$$\frac{(1 + \sigma\,\theta(j))\,(\theta(j+1) - 2\theta(j) + \theta(j-1))}{h^2} + \frac{2y(j)\,(\theta(j+1) - \theta(j-1))}{2h}$$

$$+ \sigma\left(\frac{\theta(j+1) - \theta(j-1)}{2h}\right)^2 + \frac{1}{2}\sigma^2\,\theta(j)\,\frac{(\theta^2(j+1) - 2\theta^2(j) + \theta^2(j-1))}{h^2}$$

$$+ y(j)\,\frac{\sigma(\theta^2(j+1) - \theta^2(j-1))}{2h} + \frac{1}{4}\,\frac{\sigma(\theta(j+1) - \theta(j-1))^2 - 4\theta^2(j))}{h^2} = 0$$

which formula will be denoted by

$$(8.5) \qquad R_1\theta(j) + R_3\theta(j) = 0 \ .$$

Now, we investigate the operators R_1 and R_3 by expanding them into Taylor series.
R_1 then leads to

$$(8.6) \qquad R_1\theta = 2y \frac{d\theta}{dy} + \sigma\left(\frac{d\theta}{dy}\right)^2 + (1 + \sigma\,\theta)\frac{d^2\theta}{dy^2}$$

$$+ h^2\left\{ \frac{1}{3}\frac{d^3\theta}{dy^3} + \frac{1}{3}\frac{d\theta}{dy}\frac{d^3\theta}{dy^3} + \frac{1}{12}(1 + \sigma\,\theta)\frac{d^4\theta}{dy^4} \right\} + O(h^3) .$$

R_3 gives in the same manner,

$$(8.7) \qquad R_3\theta = \sigma\,\theta\left\{2y \frac{d\theta}{dy} + \sigma\left(\frac{d\theta}{dy}\right)^2 + (1 + \sigma\,\theta)\frac{d^2\theta}{dy^2} \right\}$$

$$+ \sigma\,h^2\left\{\frac{1}{4}(1 + \sigma\,\theta)\left(\frac{d^2\theta}{dy^2}\right)^2 + \frac{1}{3}\left(1 + \sigma\frac{d\theta}{dy}\right)\theta\frac{d^3\theta}{dy^3}\right.$$

$$\left. + y\frac{d\theta}{dy}\frac{d^2\theta}{dy^2} + \frac{1}{12}\,\theta\,\frac{d^4\theta}{dy^4} \right\} + O(h^3) .$$

From (8.6) we see that (8.2) is a second-order difference approximation to (8.1), with truncation error d_1, defined by

$$(8.8) \qquad d1 = h^2\left\{ \frac{1}{3}\frac{d^3\theta}{dy^3} + \frac{1}{3}\frac{d\theta}{dy}\frac{d^3\theta}{dy^3} + \frac{1}{12}(1 + \sigma\,\theta)\frac{d^4\theta}{dy^4} + O(h^3).\right.$$

From (8.6) and (8.7) it follows that (8.3) is a second-order approximation to

$$(8.9) \qquad (1 + \sigma\,\tilde{\theta})P\,\tilde{\theta} = 0 ,$$

with truncation error d2, defined by

$$d2 = h^2\{ \frac{1}{3}\frac{d^3\theta}{dy^3} + \frac{1}{3}\frac{d\theta}{dy}\frac{d^3\theta}{dy^3} + \frac{1}{12}(1 + \sigma\,\theta)\frac{d^4\theta}{dy^4}$$

$$(8.10) \qquad + \frac{\sigma}{4}(1 + \sigma\,\theta)\left(\frac{d^2\theta}{dy^2}\right)^2 + \frac{\sigma}{3}\left(y + \sigma\frac{d\theta}{dy}\right)\theta\,\frac{d^3\theta}{dy^3}$$

$$+ \sigma\, y\frac{d\theta}{dy}\frac{d^2\theta}{dy^2} + \frac{\sigma}{12}\,\theta\,\frac{d^4\theta}{dy^4}\} + O(h^3)\ .$$

From (8.9) we see that the accuracy of difference scheme (7.1) will decrease when $(1 + \sigma\,\theta)$ tends to zero. However, we are only interested in positive values of θ, and therefore $(1 + \sigma\,\theta) \geq 1$, σ being a positive parameter.

9. The results

For several values of the parameter σ, calculations have been performed with the three methods described in the preceding sections. At the end of the paper some results of the used methods are given, they are plotted by the CALCOMP plotter coupled to the Electrologica X1 Computer of the Mathematical Centre, Amsterdam.
The results obtained applying Method B and Method C, were in very good agreement with the values given by Crank.

Using Method A, the accuracy of the results was satisfactory for small values of σ, but got worse for increasing σ. The reason for this behaviour may be made clear by the following considerations.

The truncation error, d3, of difference scheme (5.1) which is used in Method A, is, as can easily be derived,

$$(9.1) \qquad d3 = h^2\{ \frac{1}{12}\frac{d^4v}{dz^4} + \frac{1}{6}\,b(j)\,\frac{d^3v}{dz^3}\} + O(h^3)$$

From this we see that difference approximation (5.1) will be inaccurate if the derivatives $\frac{d^3v}{dz^3}$ and $\frac{d^4v}{dz^4}$ are large.

Now, we consider transformation (3.5)

$$(3.5) \qquad z = \frac{2}{\sqrt{\pi}} \int_0^y \exp(-\tau^2)d\tau \equiv \operatorname{erf} y$$

If one chooses, in the interval $0 \leq z \leq 1$, a steplength $h = .01$ (the steplength used in actual computation), then the last interior point, $z = .99$, of the lattice corresponds to $y = 1.821$. Thus all values $1.821 < y < \infty$ are mapped by (3.5) on the z-interval $.99 < z < 1$. If the analytical solution $\tilde{v}$ of the problem, is not close to zero at $z = .99$, then, of course, in the interval $.99 < z < 1$ large values for the first derivative $\frac{d\tilde{v}}{dz}$, and therefore also for the higher derivatives, must occur. This results in a large truncation error d3. The lesser accuracy of the numerical solution with increasing σ, thus is a result of the fact that the analytical solution tends slower to zero when σ is large. For large values of σ transformation (3.5) is not a suitable tool for solving the problem.

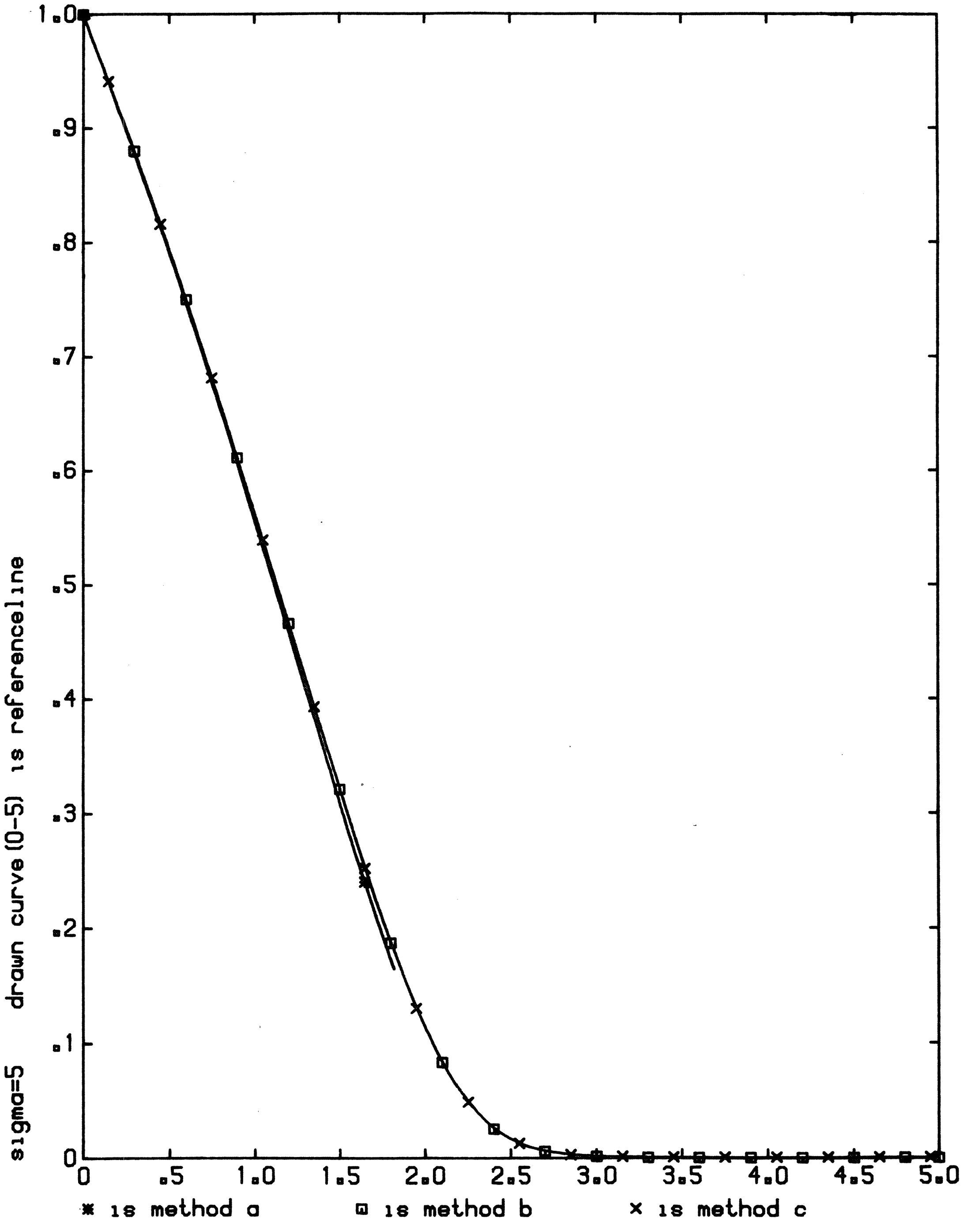
All difference schemes used in the preceding methods of calculation are only second-order approximations. In order to check the obtained results, we have solved boundary-value problem (2.4), (2.5) and boundary-value problem (3.2) numerically, applying a fifth-order Runge-Kutta procedure, due to Zonneveld (see [3]).
The boundary-value problems were solved with the iterative step-by-step method, described in section 5. In both cases the results were in very good agreement with the values given by Crank and those obtained applying Method B and C.

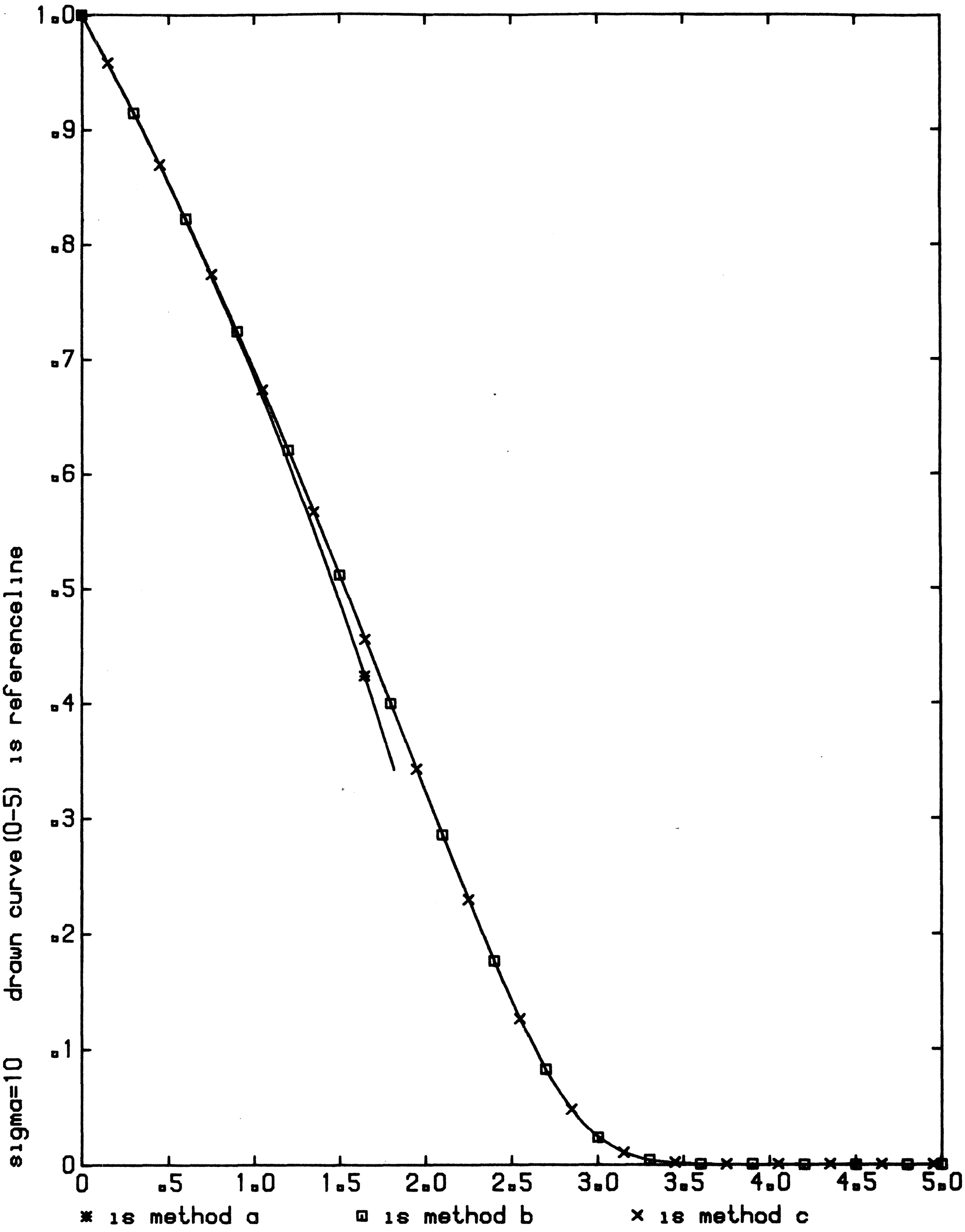
References.

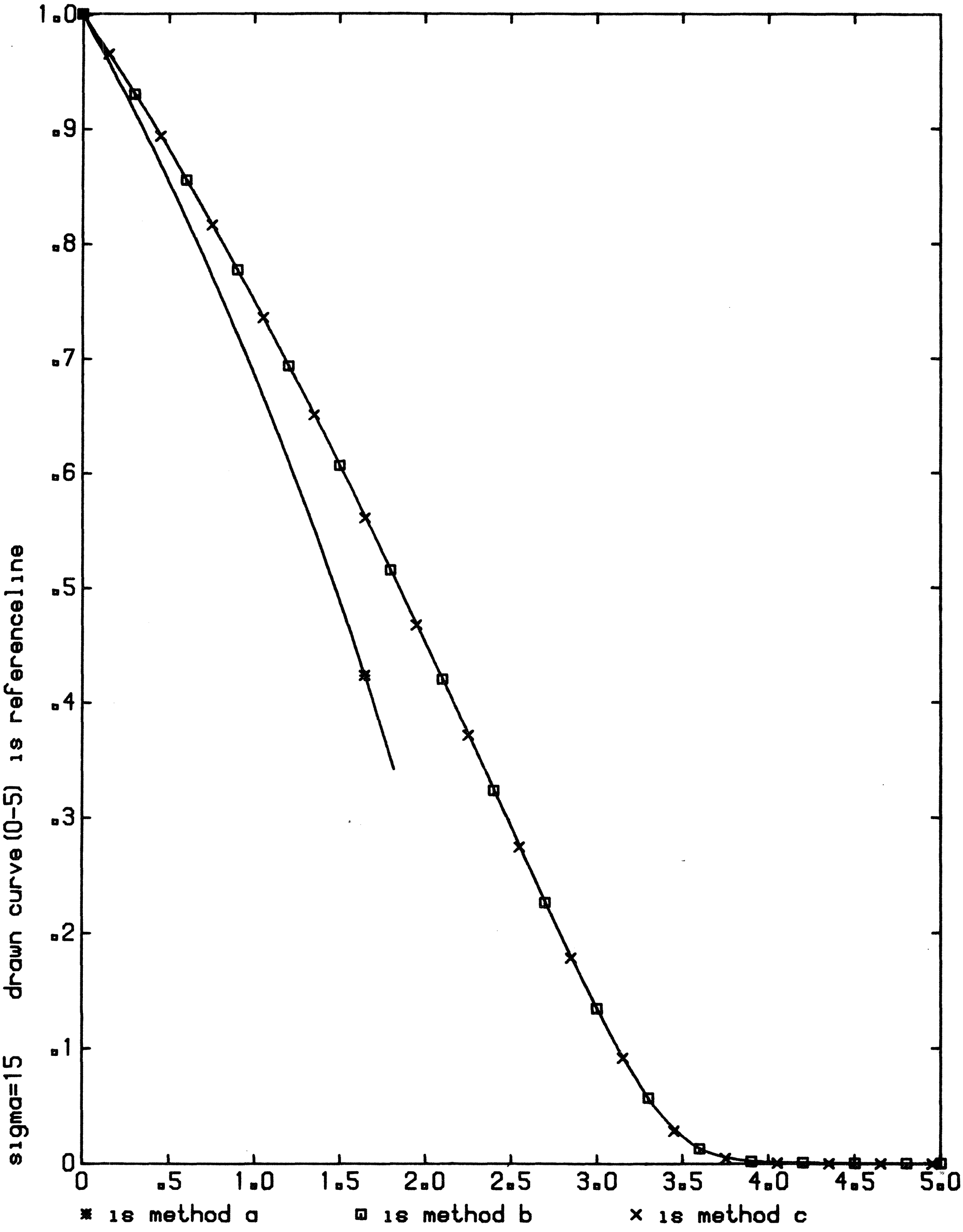
[1] J. Crank, Mathematics of diffusion.

[2] R.D. Richtmeyer and K.W. Morton, Difference methods for initial-value problems.

[3] J.A. Zonneveld, Automatic numerical integration.


sigma=5 drawn curve (0-5) is referenceline
1.0
.9
.8
.7
.6
.5
.4
.3
.2
.1
0
0 .5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
* is method a
□ is method b
× is method c


sigma=10 drawn curve(0-5) is referenceline
1.0
.9
.8
.7
.6
.5
.4
.3
.2
.1
0
0
.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
* is method a
□ is method b
× is method c


1.0
.9
.8
.7
.6
.5
.4
.3
.2
.1
0
0
.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
sigma=15   drawn curve (0-5) is referenceline
* is method a
□ is method b
× is method c

METHOD A

```
begin  comment        R1593 , GJF 010167/XIIIc .
                 Iterative solution of the differential equation :
                 U'' + 2×y/((1+2×sigma×U)↑-1/2)×U' = 0
                 as two point boundary value problem
                 including a procedure for finding an
                 initial approximation ;
       integer        N ;
BEGIN: N:= READ ;
       begin  integer        i , k , m ;
              real    a , y0 , sigma , eps , eps1 , dt , spi , max ,
                      z1 , z2 , h ;
              array   y , u[0:N] ;
              procedure      INPUT ;
              begin   h:= 1/N ; m:= READ ; sigma:= READ ; dt:= READ ;
                      z1:= READ ; z2:= READ ; eps:= READ ; eps1:= READ ;
                      spi:= sqrt(3.14159265359) ;
                      NLCR ; PRINTTEXT(≮ output gjf 010167/xiiic ≯) ;
                      NLCR ; PRINTTEXT(≮ number of netpoints N =≯) ;
                      ABSFIXT( 3,0,N ) ;
                      NLCR ; PRINTTEXT(≮ parameter sigma =≯) ;
                      ABSFIXT( 2,3,sigma ) ;
                      NLCR ; PRINTTEXT(≮ relaxation parameter dt =≯) ;
                      ABSFIXT( 2,4,dt ) ; NLCR ; NLCR ;
              end INPUT ;

       procedure      INITIAL APPROX ;
       begin  real    1 , u2 , u3 , zeta , y1 ;
AGAIN:        u1:= 1+.5×sigma ; zeta:= (z1+z2)/2 ; NLCR ; NLCR ;
              u2:= u1+h×zeta ;
              for i:= 1 step 1 until N-1 do
              begin   y0:= y[i] ;
                      a:= .5×spi×h×y0×exp(y0×y0)×(1/sqrt(1+2×sigma×u2)-1) ;
                      u3:= (2×u2+(a-1)×u1)/(a+1) ;
                      if u3 < 0 then
                      begin   z1:= (z1+z2)/2 ; goto AGAIN end ;
                      ABSFIXT( 2,3,y0 ) ; FLOT( 3,2,u3 ) ; SPACE(3) ;
                      u1:= u2 ; u2:= u3
              end ;
              if u3 > eps1 then
              begin z2:= (z2+z1)/2 ; goto AGAIN end ;
              NLCR ; NLCR ; PRINTTEXT( ≮ initialapproximation ≯ ) ; NLCR ;
              u[0]:= u1:= 1+.5×sigma ; u[1]:= u2:= u1+h×zeta ;
              SPACE(5) ; ABSFIXT( 2,3,0 ) ; SPACE(3) ;
              FLOT( 3,2,(-1+sqrt(1+2×sigma×u1))/sigma) ;
              SPACE(5) ; ABSFIXT( 2,3,y[1] ) ; SPACE(3) ;
              FLOT( 3,2,(-1+sqrt(1+2×sigma×u2))/sigma ) ;
              for i:= 1 step 1 until N-1 do
              begin   y0:= y[i] ; y1:= y[i+1] ;
                      a:= .5×h×spi×y0×exp(y0×y0)×(1/sqrt(1+2×sigma×u2)-1) ;
```

```
                    u[i+1]:= u3:= (2×u2+(a-1)×u1)/(a+1) ;
                    SPACE(5) ; ABSFIXT( 2,3,y1 ) ; SPACE(3) ;
                    FLOT( 3,2,(-1+sqrt(1+2×sigma×u3))/sigma ) ;
                    if (i:3)×3 = i then NLCR ;
                    u1:= u2 ; u2:= u3
               end ;
               u[N]:= 0 ; NLCR ; NLCR ; NLCR
          end INITIAL APPROX ;
          procedure       ITERATE ;
          begin   real    v0 , v1 , v2 , ui , Riv ;
ITER:             v0:= u[0] ; v1:= u[1] ; max:= 0 ; k:= k+1 ;
                  PRINTTEXT(≮ iteration started ≯) ; NLCR ; NLCR ;
                  for i:= 1 step 1 until N-1 do
                  begin   y0:= y[i] ; v2:= u[i+1] ;
                          a:= .5×h×spi×y0×exp(y0×y0)×(1/sqrt(1+2×sigma×v1)-1) ;
                          Riv:= (a+1)×v2-2×v1-(a-1)×v0 ;
                          if max < abs(Riv) then max:= abs(Riv) ;
                          u[i] := ui:= v1+dt×Riv ; v0:= v1 ; v1:= v2 ;
                          if (k:5)×5 = k then
                          begin   SPACE(4) ; ABSFIXT( 2,3,y0 ) ; SPACE(2) ;
                                  FLOT( 3,2,(-1+sqrt(1+2×sigma×ui))/sigma ) ;
                                  SPACE(2) ; FLOT( 3,2,Riv ) ;
                                  if (i:3)×3 = i then NLCR
                          end ; NLCR ; NLCR ;
                  end ;
                  if max > eps then goto ITER ;
                  PRINTTEXT(≮ final results gjf 010167/xiiic with  sigma =≯) ;
                  ABSFIXT( 2,3,sigma ) ; NLCR ; NLCR ;
                  for i:= 0 step 1 until N do
                  begin   SPACE(5) ; ABSFIXT( 2,3,y[i] ) ; SPACE(3) ;
                          FLOT( 3,2,(-1+sqrt(1+2×sigma×u[i]))/sigma ) ;
                          if (i:3)×3 = i then NLCR
                  end
          end ITERATE ;
                  for i:= 0 step 1 until N do y[i]:= READ ; k:= 0 ;
PROGRAM:          NEWPAGE ; INPUT ; INITIAL APPROX ; ITERATE ;
                  if m = 0 then goto BEGIN ;
                  if m = 1 then goto PROGRAM
          end
end
```

METHOD B

```
begin   comment   R1593 , GJF010167/XIIIe.
                  u'' = (-sigma×(u')↑2-2×u'×y)/(1+sigma×u) , as initial value
                  problem ;
        integer         n,m,Y ;
        real      sigma,h,y,theta1,theta2,theta3,z1,z2,zeta,eps1,eps2 ;
        procedure   SCHEME ;
                  begin   real    B , C ;
                          B:= (1+sigma×theta2-.5×sigma×theta1+y×h) ;
                          C:= -2×sigma×theta2×theta2+theta2×(sigma×theta1-2)
                                +(1-y×h+.25×sigma×theta1)×theta1 ;
                          y:= y+h ; theta3:= 2×(-B+sqrt(B↑2-sigma×C))/sigma ;
                          theta1:= theta2 ; theta2:= theta3
                  end ;
        NLCR ; PRINTTEXT (⊀ Results GJF010167/XIIIe ⊁) ; NLCR ;
NEW:    m:= READ ; sigma:= READ ; h:= READ ; z1:= READ ; z2:= READ ;
        Y:= READ ; eps1:= READ ; eps2:= READ ; ABSFIXT(2,3,sigma) ;
AGAIN:  NLCR ; NLCR ; y:= h ; theta1:= 1 ; zeta:= (z1+z2)/2 ;
        theta2:= theta1+h×zeta ;
        for n:= 1 step 1 until Y do
        begin   SCHEME ; ABSFIXT(2,3,y) ; FLOT(3,2,theta2) ;
                if theta2 < 0 then begin z1:= (z1+z2)/2 ; goto AGAIN end
        end ;
        zeta:= (theta2-theta1)/h ;
        if abs(zeta) < eps2 ∧ theta2 < eps1 then goto OUT ;
        z2:= (z1+z2)/2 ; goto AGAIN ;
OUT:    NLCR ; NLCR ; PRINTTEXT
        (⊀ final results GJF010167/XIIIe with sigma = ⊁) ;
        PRINT(sigma) ; y:= h ; theta1:= 1 ; zeta:= (z1+z2)/2 ;
        theta2:= theta1+zeta×h ;
        NLCR ; ABSFIXT(2,3,y) ; FLOT(3,2,theta2) ; FLOT(3,2,zeta) ;
        for n:= 1 step 1 until Y do
        begin SCHEME ; NLCR ; ABSFIXT(2,3,y) ; FLOT(3,2,theta2) end ;
        if m = 0 then begin NEWPAGE ; goto NEW end
end
```

METHOD C

```
begin   comment  R1593 ,  GJF010167/XIIIg.
                 U'' + 2y/((1+2sU)↑-1/2)U' = 0 , as initial value
                 problem ;
        integer        n,m,Y ;
        real     sigma,h,y,theta1,theta2,theta3,z1,z2,zeta,eps1,eps2 ;
        procedure    SCHEME ;
        begin    real    A ;
                 A:= h×y/sqrt(1+2×sigma×theta2) ;
                 theta3:= (2×theta2+(A-1)×theta1)/(A+1) ;
                 y:= y+h ; theta1:= theta2 ; theta2:= theta3
        end ;
        NLCR ; PRINTTEXT ( ≮ Results GJF010167/XIIIg ≯ ) ; NLCR ;
NEW:    m:= READ ; sigma:= READ ; h:= READ ; z1:= READ ; z2:= READ ;
        Y:= READ ; eps1:= READ ; eps2:= READ ; ABSFIXT( 2,3,sigma ) ;
AGAIN:  NLCR ; NLCR ; y:= h ; theta1:= 1+.5×sigma ; zeta:= (z1+z2)/2 ;
        theta2:= theta1+h×zeta ;
        for n:= 1 step 1 until Y do
        begin    SCHEME ; ABSFIXT(2,3,y) ; FLOT(3,2,theta2) ;
                 if theta2 < 0 then begin z1:= (z1+z2)/2 ; goto AGAIN end
        end ;
        zeta:= (theta2-theta1)/h ;
        if abs(zeta) < eps2 ∧ theta2 < eps1 then goto OUT ;
        z2:= (z1+z2)/2 ; goto AGAIN ;
OUT:    NLCR ; NLCR ; PRINTTEXT
         ( ≮ Final results  GJF010167/XIIIg  with sigma = ≯ ) ;
        PRINT ( sigma ) ; y:= h ; theta1:= 1+.5×sigma ;
        zeta:= (z1+z2)/2 ; theta2:= theta1+zeta×h ; NLCR ;
        ABSFIXT(2,3,y) ; FLOT(3,2,(-1+sqrt(1+2×sigma×theta2))/sigma) ;
        FLOT(3,2,zeta) ;
        for n:= 1 step 1 until Y do
        begin SCHEME ; NLCR ; ABSFIXT(2,3,y) ;
              FLOT(3,2,(-1+sqrt(1+2×sigma×theta3))/sigma)
        end ;
        if m = 0 then begin NEWPAGE ; goto NEW end
end
```

References.

[1] J.Crank, Mathematics of diffusion
Oxford, Clarendon Press 1956

[2] R.D.Richtmyer and K.W.Morton,
Difference methods for initial-value problems
New York, Interscience 1967

[3] J.A.Zonneveld, Automatic numerical integration
Amsterdam, Mathematical Centre Tracts 8