ELSEVIER

Journal of Computational and Applied Mathematics 106 (1999) 255–269

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.nl/locate/cam

Fully implicit finite differences methods for two-dimensional diffusion with a non-local boundary condition

Mehdi Dehghan

Department of Applied Mathematics, Amirkabir University of Technology, Hafez Avenue, No. 424, Tehran, Iran

Received 17 August 1998

Abstract

Three new fully implicit methods which are based on the (5,5) Crank–Nicolson method, the (5,5) N-H (Noye–Hayman) implicit method and the (9,9) N-H implicit method are developed for solving the heat equation in two dimensional space with non-local boundary conditions. The latter is fourth-order while the others are second-order. While the implicit methods developed here, like the scheme based on the standard implicit backward time centered space (BTCS) method, use a large amount of central processor (CPU) time, the high accuracy of the new fourth-order fully implicit scheme is significant. Like the BTCS method, the new methods are also unconditionally stable. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Two-dimensional diffusion; Numerical integration technique; Non-local boundary value problem; Finite differences scheme; Fully implicit method; Partial differential equation

1. Introduction

Three new fully implicit methods which are based on the (5,5) Crank–Nicolson method, the (5,5) N-H implicit method and the (9,9) N-H implicit method are developed. The problem to which the three methods are applied is the two dimensional time dependent diffusion

$$\frac{\partial u}{\partial t} = \alpha_x \frac{\partial^2 u}{\partial x^2} + \alpha_y \frac{\partial^2 u}{\partial y^2} \tag{1}$$

with initial condition given by

$$u(x, y, 0) = f(x, y), \quad 0 \leqslant x, \ y \leqslant 1 \tag{2}$$

and boundary conditions

$$u(0, y, t) = g_0(y, t), \quad 0 \leqslant t \leqslant T, \quad 0 \leqslant y \leqslant 1, \tag{3}$$

$$u(1, y, t) = g_1(y, t), \quad 0 \leqslant t \leqslant T, \quad 0 \leqslant y \leqslant 1, \tag{4}$$

0377-0427/99/$ - see front matter
PII: S0377-0427(99)00065-5

$$u(x,1,t) = h_1(x,t), \quad 0 \leqslant t \leqslant T, \quad 0 \leqslant x \leqslant 1, \tag{5}$$

$$u(x,0,t) = h_0(x)\mu(t), \quad 0 \leqslant t \leqslant T, \quad 0 \leqslant x \leqslant 1 \tag{6}$$

with the nonlocal boundary condition

$$\int_0^1 \int_0^{d(x)} u(x,y,t)\,\mathrm{d}x\,\mathrm{d}y = m(t), \quad 0 \leqslant x, \ y \leqslant 1, \tag{7}$$

where f, g_0, g_1, h_0, h_1, d and m are known functions, while the functions u and μ are unknown.

This kind of problem arises in many important applications in heat transfer, control theory, thermoelasticity and medical science [2–6,8,14].

Numerical schemes for the solution of Eqs. (1)–(7) are described in Section 2. The iterative procedure which is used to incorporate (7) with μ unknown is described in Section 3.

The results produced by using these methods for a test are described in Section 4. In each case errors are tabulated. Section 5 summarizes the findings of this article.

2. Finite-difference methods

The domain $[0,1]^2 \times [0,T]$ will be divided into an $M^2 \times N$ mesh with spatial step size $h = 1/M$ in both x and y directions and the time step size $k = T/N$, respectively.

Grid points (x_i, y_j, t_n) are given by

$$x_i = ih, \quad i = 0,1,2,\ldots,M, \tag{8}$$

$$y_j = jh, \quad j = 0,1,2,\ldots,M, \tag{9}$$

$$t_n = nk, \quad n = 0,1,2,\ldots,N, \tag{10}$$

in which M is an even integer. We use $u_{i,j}^n$ and μ^n to denote the finite difference approximations of $u(ih,jh,nk)$ and $\mu(nk)$, respectively.

The numerical methods suggested here are based on 3 approaches: Firstly, the standard fully implicit second-order BTCS method [10], or the (5,5) Crank–Nicolson fully implicit method [7], or the (5,5) N-H fully implicit method [12], or the (9,9) N-H fully implicit method [12], is used to approximate the solution of the two-dimensional diffusion equation at interior grid points. Secondly, the Simpson's numerical integration scheme [9] is used to approximate the integral in Eq. (7). Thirdly an iteration procedure is employed to handle the non-local boundary condition [1].

The problem (1)–(6) is solved numerically at the spatial points (x_i, y_j), commencing with initial values $u_{i,j}^0 = f(x_i, y_j)$, $i,j = 0,1,2,\ldots,M$, and boundary values (2)–(6) where $\mu(t)$ is computed using an iterative procedure.

Given numerical solutions of u and μ at time level n, $n = 0,1,2\ldots$, an appropriate initial guess for μ is made at the time level $n+1$, say μ at the time level n, then (1)–(6) by using any of those methods which are mentioned already to find the value of u at the time level $n+1$. If the solution satisfies the nonlocal condition (7) within a prescribed tolerance, then the present values of u and μ are accepted as the approximate solution for u and μ at the $n+1$ level. Otherwise, a prediction for μ will be found from Eq. (7). Computations are then repeated with this new prediction until Eq. (7) is satisfied within the given tolerance, and this is repeated for higher levels.

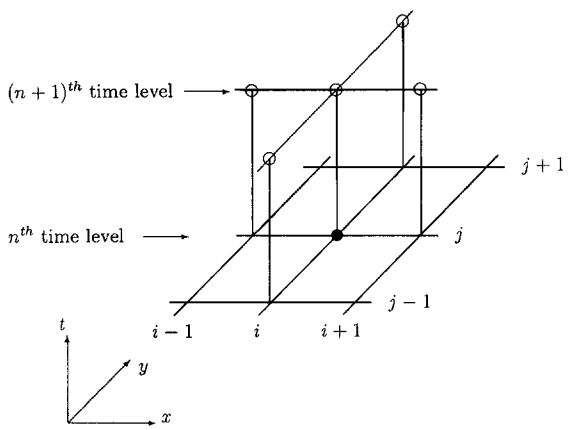

Fig. 1. (5,1) BTCS computational stencil.

2.1. The standard (5,1) BTCS method

The five point BTCS [10] (backward Euler) for solving the two-dimensional partial differential equation (1) uses the following formula:

$$s_x(u^{n+1}_{i-1,j} + u^{n+1}_{i+1,j}) + s_y(u^{n+1}_{i,j-1} + u^{n+1}_{i,j+1}) - (1 + 2s_x + 2s_y)u^{n+1}_{i,j} = -u^n_{i,j}, \tag{11}$$

for $i,j = 1,2,\ldots,M-1$, where

$$s_x = \alpha_x k/h^2, \tag{12}$$

$$s_y = \alpha_y k/h^2. \tag{13}$$

In the case $\alpha_x = \alpha_y = \alpha$, we have

$$s_x = s_y = s = k\alpha/h^2, \tag{14}$$

and Eq. (11) becomes

$$s(u^{n+1}_{i-1,j} + u^{n+1}_{i+1,j} + u^{n+1}_{i,j-1} + u^{n+1}_{i,j+1}) - (1 + 4s)u^{n+1}_{i,j} = -u^n_{i,j}. \tag{15}$$

For the classical boundary value problem values of $u^{n+1}_{i,j}$ on the boundaries $x = 0,1$ and $y = 0,1$ are provided by the boundary conditions (3)–(6) at the appropriate grid points.

The computational molecule of this method is given in Fig. 1. In the following this will be referred to as the (5,1) method, because the computational molecule involves 5 gridpoints at the new time level and 1 at the old level.

The modified equivalent equation for this method is as follows [15]:

$$\frac{\partial u}{\partial t} - \alpha_x \frac{\partial^2 u}{\partial x^2} - \alpha_y \frac{\partial^2 u}{\partial y^2} - \frac{\alpha_x(\Delta x)^2}{12}(1 + 6s_x)\frac{\partial^4 u}{\partial x^4} - \frac{\alpha_y(\Delta y)^2}{12}(1 + 6s_y)\frac{\partial^4 u}{\partial y^4} + O\{4\} = 0, \tag{16}$$

so the scheme is second-order accurate with respect to h.

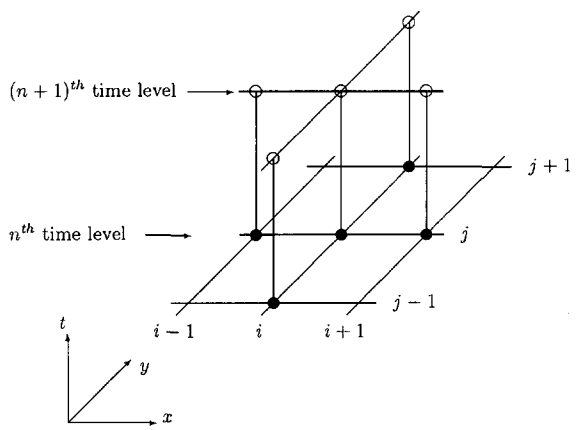

Fig. 2. The computational molecule for the (5,5) Crank–Nicolson and (5,5) N-H implicit methods.

2.2. The Crank–Nicolson (5, 5) method

If we replace all spatial derivatives with the average of their values at the n and $n+1$ time levels and then substitute centred-difference forms for all derivatives, we get the Crank–Nicolson (5,5) formula [7]

$$s_x(u^{n+1}_{i+1,j} + u^{n+1}_{i-1,j}) - 2(1 + s_x + s_y)u^{n+1}_{i,j} + s_y(u^{n+1}_{i,j+1} + u^{n+1}_{i,j-1})$$
$$= -s_y(u^{n}_{i,j-1} + u^{n}_{i,j+1}) - 2(1 - s_x - s_y)u^{n}_{i,j} - s_x(u^{n}_{i-1,j} + u^{n}_{i+1,j}). \tag{17}$$

In the case $\alpha_x = \alpha_y = \alpha$ we have $s_x = s_y = s$, and the new finite-difference equation is

$$s(u^{n+1}_{i,j-1} + u^{n+1}_{i-1,j}) - 2(1 + 2s)u^{n+1}_{i,j} + s(u^{n+1}_{i+1,j} + u^{n+1}_{i,j+1})$$
$$= -s(u^{n}_{i,j-1} + u^{n}_{i,j+1}) - 2(1 - 2s)u^{n}_{i,j} - s(u^{n}_{i-1,j} + u^{n}_{i+1,j}). \tag{18}$$

This scheme has the computational molecule which is shown in Fig. 2. In the following this will be referred to as the (5,5) CN method, because the computational molecule involves 5 gridpoints at the new time level and 5 at the old level.

The modified equivalent equation of the Crank–Nicolson formula (18) is as follows [15]:

$$\frac{\partial u}{\partial t} - \alpha_x \frac{\partial^2 u}{\partial x^2} - \alpha_y \frac{\partial^2 u}{\partial y^2} - \frac{\alpha_x (\Delta x)^2}{12}\frac{\partial^4 u}{\partial x^4} - \frac{\alpha_y (\Delta y)^2}{12}\frac{\partial^4 u}{\partial y^4} + \mathrm{O}\{4\} = 0. \tag{19}$$

It is second-order accurate in the spatial grid size with no second-order cross-derivative terms. However, there is no set of values of s for which the method will be fourth-order accurate.

2.3. The (5, 5) N-H implicit method

This method uses the following finite-difference formula [12]

$$(1 - 6s_x)(u^{n+1}_{i-1,j} + u^{n+1}_{i+1,j}) + (1 - 6s_y)(u^{n+1}_{i,j-1} + u^{n+1}_{i,j+1}) + 4(2 + 3s_x + 3s_y)u^{n+1}_{i,j}$$
$$= (1 + 6s_y)(u^{n}_{i,j-1} + u^{n}_{i,j+1}) + (1 + 6s_x)(u^{n}_{i-1,j} + u^{n}_{i+1,j}) + 4(2 - 3s_x - 3s_y)u^{n}_{i,j}. \tag{20}$$

In the case where $s_x = s_y = s$ the above scheme uses the following simplified form:

$$(1-6s)(u^{n+1}_{i-1,j}+u^{n+1}_{i,j-1}+u^{n+1}_{i+1,j}+u^{n+1}_{i,j+1})+8(1+3s)u^{n+1}_{i,j}$$
$$=(1+6s)(u^{n}_{i,j-1}+u^{n}_{i,j+1}+u^{n}_{i-1,j}+u^{n}_{i+1,j})+8(1-3s)u^{n}_{i,j}. \tag{21}$$

The computational molecule of this scheme is the same as the Crank–Nicolson formula shown in Fig. 2.

The modified equivalent equation of this (5,5) N-H implicit formula is [12]

$$\frac{\partial u}{\partial t}-\alpha_x\frac{\partial^2 u}{\partial x^2}-\alpha_y\frac{\partial^2 u}{\partial y^2}+\frac{\sqrt{(\alpha_x\alpha_y)}(\Delta x)(\Delta y)}{12}(s_x+s_y)\frac{\partial^4 u}{\partial x^2\partial y^2}+\mathrm{O}\{4\}=0. \tag{22}$$

It contains only the second-order cross-derivative error term in its modified equivalent equation.

2.4. The fourth-order (9, 9) N-H implicit method

This scheme uses the finite difference formula [12]

$$-(s_x+s_y)(u^{n+1}_{i-1,j-1}+u^{n+1}_{i+1,j-1}+u^{n+1}_{i-1,j+1}+u^{n+1}_{i+1,j+1})$$
$$+2(1+s_x-5s_y)(u^{n+1}_{i,j-1}+u^{n+1}_{i,j+1})+2(1+s_y-5s_x)(u^{n+1}_{i-1,j}+u^{n+1}_{i+1,j})$$
$$+4(4+5s_x+5s_y)u^{n+1}_{i,j}$$
$$=(s_x+s_y)(u^{n}_{i-1,j-1}+u^{n}_{i-1,j+1}+u^{n}_{i+1,j-1}+u^{n}_{i+1,j+1})$$
$$+2(1-s_x+5s_y)(u^{n}_{i,j-1}+u^{n}_{i,j+1})+2(1-s_y+5s_x)(u^{n}_{i-1,j}+u^{n}_{i+1,j})$$
$$+4(4-5s_x-5s_y)u^{n}_{i,j}. \tag{23}$$

In the case where $s_x = s_y = s$ we have

$$-s(u^{n+1}_{i-1,j-1}+u^{n+1}_{i+1,j-1})+(1-4s)(u^{n+1}_{i,j-1}+u^{n+1}_{i-1,j})$$
$$+4(2+5s)u^{n+1}_{i,j}+(1-4s)(u^{n+1}_{i+1,j}+u^{n+1}_{i,j+1})-s(u^{n+1}_{i-1,j+1}+u^{n+1}_{i+1,j+1})$$
$$=s(u^{n}_{i-1,j-1}+u^{n}_{i-1,j+1}+u^{n}_{i+1,j-1}+u^{n}_{i+1,j+1})$$
$$+(1+4s)(u^{n}_{i-1,j}+u^{n}_{i+1,j}+u^{n}_{i,j-1}+u^{n}_{i,j+1})+4(2-5s)u^{n}_{i,j}. \tag{24}$$

This scheme has the computational molecule which is shown in Fig. 3. In the following this will be referred to as the (9,9) N-H method, because the computational molecule involves 9 gridpoints at the new time level and 9 at the old level.

The modified equivalent equation of this implicit formula is as follows [12]:

$$\frac{\partial u}{\partial t}-\alpha_x\frac{\partial^2 u}{\partial x^2}-\alpha_y\frac{\partial^2 u}{\partial y^2}+\frac{\alpha_x(\Delta x)^4}{240}(1-20(s_x)^2)\frac{\partial^6 u}{\partial x^6}$$
$$-\frac{\alpha_x(\Delta x)^2(\Delta y)^2}{144}(1+36s_xs_y)\frac{\partial^6 u}{\partial x^4\partial y^2}-\frac{\alpha_y(\Delta x)^2(\Delta y)^2}{144}(1+36s_xs_y)\frac{\partial^6 u}{\partial x^2\partial y^4}$$
$$+\frac{\alpha_y(\Delta y)^4}{240}(1-20(s_y)^2)\frac{\partial^6 u}{\partial y^6}+\mathrm{O}\{6\}=0, \tag{25}$$

which verifies its fourth-order accuracy with respect to h.

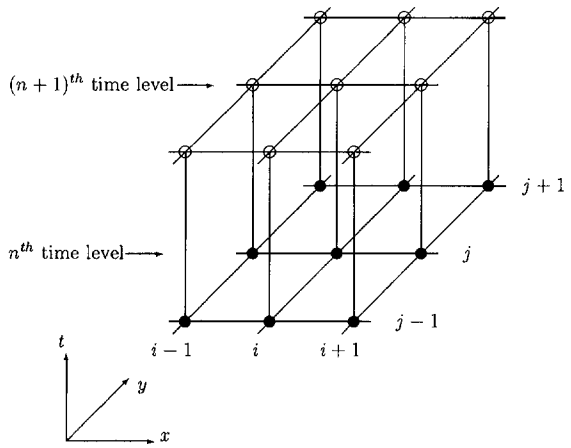

Fig. 3. (9,9) N-H implicit computational molecule.

3. Simpson's numerical integration procedure

Given computed values of u and μ at time level n, $n=0,1,2,\ldots$, first a suitable initial estimate for μ is made at the n time level, then, (1)–(6) is solved by using any of methods mentioned previously to find the value of u at the time level $n+1$. If the solution satisfies the nonlocal condition (7) within a chosen tolerance, then the current values of u and μ are accepted as the solution for u and μ at the time level $n+1$. Otherwise, a new estimate for μ will be found from (7). Computations are then repeated with this new guess until (7) is satisfied with the given tolerance and then repeat this for higher levels [1].

Consider the integral

$$H(x,t^{n+1}) = \int_0^{d(x)} u(x,y,t^{n+1})\,\mathrm{d}y. \tag{26}$$

Application of Simpson's composite 'one-third' rule [9] gives

$$\int_0^1 H(x,t^{n+1})\,\mathrm{d}x \simeq \frac{h}{3}\left(H_0^{n+1} + 4\sum_{i=1}^{M/2} H_{2i-1}^{n+1} + 2\sum_{i=1}^{(M/2)-1} H_{2i}^{n+1} + H_M^{n+1}\right), \tag{27}$$

$$H_i^{n+1} = \int_0^{d(ih)} u(x_i,y,t^{n+1})\,\mathrm{d}y, \tag{28}$$

in which

$$H_i^{n+1} = \int_0^{2l_ih} u(x_i,y,t^{n+1})\,\mathrm{d}y + \int_{2l_ih}^{d(ih)} u(x_i,y,t^{n+1})\,\mathrm{d}y, \tag{29}$$

where

$$l_i = [d(ih)/2h], \tag{30}$$

and $[\cdot]$ represents the integer part of the argument. Substituting in the second integral of Eq. (29)

$$z_i = y/h - 2l_i, \tag{31}$$

yields

$$\int_{2l_i h}^{d(ih)} u(x_i, y, t^{n+1})\,\mathrm{d}y = h\int_0^{\delta_i} u(x_i, z_i, t^{n+1})\,\mathrm{d}z_i, \tag{32}$$

where

$$\delta_i = (d(ih)/h) - 2l_i. \tag{33}$$

Replacement of u in the integral with a quadratic interpolating polynomial (the Newton's forward-difference formula) [13] through the grid values concerned, gives

$$\int_0^{\delta_i} u(x_i, z_i, t^{n+1})\,\mathrm{d}z_i = \int_0^{\delta_i} [u(x_i, y_{2l_i}, t^{n+1}) + z_i \Delta u(x_i, y_{2l_i}, t^{n+1}) + \frac{1}{2} z_i(z_i - 1)\Delta^2 u(x_i, y_{2l_i}, t^{n+1})]\,\mathrm{d}z_i + \mathrm{O}(h^4). \tag{34}$$

where

$$\Delta u(x_i, y_{2l_i}, t^{n+1}) = u(x_i, y_{2l_{i+1}}, t^{n+1}) - u_(x_i, y_{2l_i}, t^{n+1}), \tag{35}$$

$$\Delta^2 u(x_i, y_{2l_i}, t^{n+1}) = u(x_i, y_{2l_i+1}, t^{n+1}) - 2u(x_i, y_{2l_i+1}, t^{n+1}) + u(x_i, y_{2l_i}, t^{n+1}). \tag{36}$$

Integrating Eq. (34) and collecting like terms and then substituting in Eq. (29) means that

$$\begin{aligned} H_i^{n+1} = \frac{h}{3}\Bigg[& u(x_i, 0, t^{n+1}) + 4\sum_{j=1}^{l_i} u(x_i, y_{2j-1}, t^{n+1}) + 2\sum_{j=1}^{l_i-1} u(x_i, y_{2j}, t^{n+1}) \\ & + u(x_i, y_{2l_i}, t^{n+1}) + 3\delta_i(1 - 3\delta_i/4 + \delta_i^2/6)u(x_i, y_{2l_i}, t^{n+1}) \\ & + 3\delta_i^2(1 - \delta_i/3)u(x_i, y_{2l_i+1}, t^{n+1}) \\ & + (\delta_i^2/4)(2\delta_i - 3)u(x_i, y_{2l_i+2}, t^{n+1})\Bigg] + \mathrm{O}(h^4). \end{aligned} \tag{37}$$

However, at all interior points we have computed $u(x_i, y_j, t^{n+1})$ to say, rth-order, where

$$u(x_i, y_{2l_i}, t^{n+1}) = u_{i,j}^{n+1} + \mathrm{O}(h^r). \tag{38}$$

Substituting these approximations in Eq. (37) gives

$$\begin{aligned} H_i^{n+1} = \frac{h}{3}[& u_{i,0}^{n+1} + 4\sum_{j=1}^{l_i} u_{i,2j-1}^{n+1} + 2\sum_{j=1}^{l_i-1} u_{i,2j}^{n+1} + u_{i,2l_i}^{n+1} \\ & + 3\delta_i(1 - 3\delta_i/4 + \delta_i^2/6)u_{i,2l_i}^{n+1} + 3\delta_i^2(1 - \delta_i/3)u_{i,2l_i+1}^{n+1} \end{aligned} \tag{39}$$

$$+ (\delta_i^2/4)(2\delta_i - 3)u_{i,2l_i+2}^{n+1}] + \mathrm{O}(h^q), \tag{40}$$

where

$$q = \min\{r, 4\}. \tag{41}$$

Putting

$$V(t^{n+1}) = \int_0^1 H(x, t^{n+1})\,dx, \tag{42}$$

and using the approximation

$$v^{n+1} = \frac{h}{3}\left(H_0^{n+1} + 4\sum_{i=1}^{M/2} H_{2i-1}^{n+1} + 2\sum_{i=1}^{(M/2)-1} H_{2i}^{n+1} + H_M^{n+1}\right) + O(h^q), \tag{43}$$

then gives

$$v^{n+1} = \frac{h^2}{9}\left(u_{0,0}^{n+1} + 4\sum_{i=1}^{M} u_{2i-1,0}^{n+1} + 2\sum_{i=1}^{(M/2)-1} u_{2i,0}^{n+1} + u_{M,0}^{n+1}\right) + R^{n+1} + O(h^q). \tag{44}$$

Note that

$$v^{n+1} = \frac{h}{3}\mu^{n+1}\int_0^1 h_0(x)\,dx + R^{n+1} + O(h^q), \tag{45}$$

where R^{n+1} is the summation in v^{n+1} excluding the values at the boundary $y = 0$.

Since v^{n+1} is an approximation to the left-hand side of Eq. (7) it follows that

$$\mu^{n+1} = \frac{m^{n+1} - R^{n+1}}{\frac{h}{3}\int_0^1 h_0(x)\,dx} + O(h^{q-1}), \qquad \int_0^1 h_0(x)\,dx \neq 0. \tag{46}$$

As seen above, the order of convergence of μ depends on two things: firstly, the order of the finite-difference formula used at interior gridpoints and, secondly, the order of the numerical quadrature used to approximately evaluate (7). For example, if u^{n+1} is evaluated using a fourth-order formula, when $q = 4$, μ^{n+1} is only third-order convergent. If u^{n+1} is found at interior points by a second-order formula when $q = 2$ then μ^{n+1} is only first-order convergent.

Here as a new estimate for $\mu^{n+1,p+1}$ we use the following:

$$\mu^{n+1,p+1} = \frac{m^{n+1} - R^{n+1,p}}{\frac{h}{3}\int_0^1 h_0(x)\,dx} + O(h^{q-1}), \tag{47}$$

If u^{n+1} is evaluated approximately using the (9,9) N-H implicit formula, when $q = 4$, μ^{n+1} is only third-order convergent. If u^{n+1} is found at interior points by any of the BTCS formula or Crank–Nicolson formula or the (5,5) N-H implicit formula, when $q = 2$ then μ^{n+1} is only first-order convergent.

4. Numerical test

A problem for which exact nonlocal boundary solutions are known is now used to test the methods described. Firstly, these methods are applied to solve Eqs. (1)–(6) with $\mu(t)$ given, in order to test the methods used to compute values of $u_{i,j}^{n+1}$ from $u_{i,j}^{n}$ in the interior of the solution domain.

Consider Eqs. (1)–(7) with $\alpha_x = \alpha_y = 1$, and

$$f(x, y) = \exp(x + y), \tag{48}$$

Table 1
Results for u with $T = 1.0$, $h = 0.05$, $s = 1/2$

x	y	Exact u	BTCS Error	Crank–Nicolson Error	(5,5) N-H implicit Error	(9,9) N-H implicit Error
0.1	0.1	9.025013	-0.5×10^{-3}	-0.7×10^{-4}	0.7×10^{-4}	-0.3×10^{-7}
0.2	0.2	11.023176	-0.1×10^{-2}	-0.2×10^{-3}	0.2×10^{-3}	-0.1×10^{-6}
0.3	0.3	13.463738	-0.2×10^{-2}	-0.3×10^{-3}	0.3×10^{-3}	-0.2×10^{-6}
0.4	0.4	16.444647	-0.3×10^{-2}	-0.5×10^{-3}	0.5×10^{-3}	-0.2×10^{-6}
0.5	0.5	20.905243	-0.4×10^{-2}	-0.6×10^{-3}	0.6×10^{-3}	-0.3×10^{-6}
0.6	0.6	24.532530	-0.4×10^{-2}	-0.6×10^{-3}	0.6×10^{-3}	-0.3×10^{-6}
0.7	0.7	29.964100	-0.3×10^{-2}	-0.5×10^{-3}	0.5×10^{-3}	-0.3×10^{-6}
0.8	0.8	36.598234	-0.3×10^{-2}	-0.4×10^{-3}	0.4×10^{-3}	-0.2×10^{-6}
0.9	0.9	40.447304	-0.1×10^{-2}	-0.2×10^{-3}	0.2×10^{-3}	-0.9×10^{-7}

$$g_0(y,t) = \exp(y + 2t), \tag{49}$$

$$g_1(y,t) = \exp(1 + y + 2t), \tag{50}$$

$$h_0(x) = \exp(x), \tag{51}$$

$$h_1(x) = \exp(1 + x + 2t), \tag{52}$$

$$\mu(t) = \exp(2t), \tag{53}$$

$$m(t) = (4\exp(\exp(1)/4) - 4\exp(1/4) - \exp(1) + 1)\exp(2t), \tag{54}$$

$$d(x) = \exp(x)/4, \tag{55}$$

for which the exact solution is

$$u(x,y,t) = \exp(x + y + 2t). \tag{56}$$

The results for $u^N_{i,j}$ with $h=0.05$, $s=1/2$ at $T=1.0$, using the four fully implicit methods discussed in Section 2 and defining $\mu(t)$ as in Eq. (53) and excluding Eq. (54), are shown in Table 1. Note that the errors obtained when using the Crank–Nicolson scheme or the (5,5) N-H implicit method are generally more than a thousand times larger than those obtained using the (9,9) N-H implicit method. The errors with the BTCS method are generally 10 000 times larger than those obtained using the (9,9) N-H implicit method.

When the absolute value of the error

$$e^n_{i,j} = u(ih, jh, nk) - u^n_{i,j}, \tag{57}$$

at the point (0.5,0.5) at time $T = 1.0$ was graphed against h on a logarithmic scale for various values of s, it was found that the slopes of lines were always close to 2 for the BTCS formula, the (5,5) N-H implicit formula and the Crank–Nicolson formula, and were close to 4 for the (9,9) N-H implicit formula (see Figs. 4–7). These results reflect the orders of convergence referred to in Section 2.

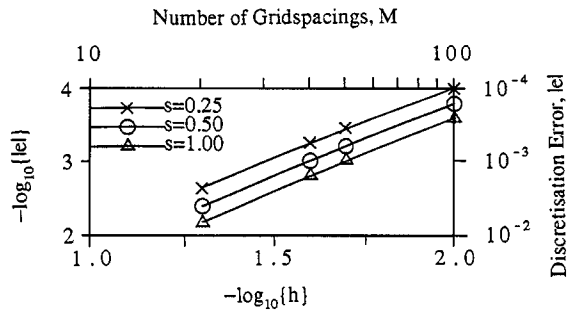

Fig. 4. Relation between error in u and grid spacing for the (5,1) BTCS method.

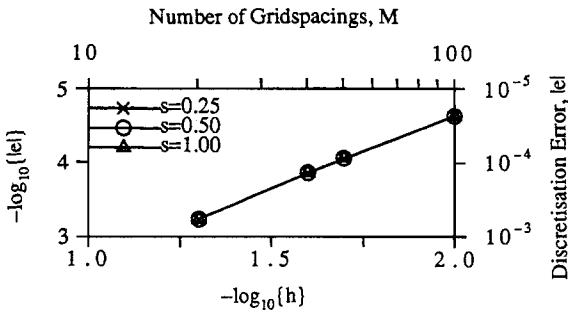

Fig. 5. Relation between error in u and grid spacing for the (5,5) Crank–Nicolson method.

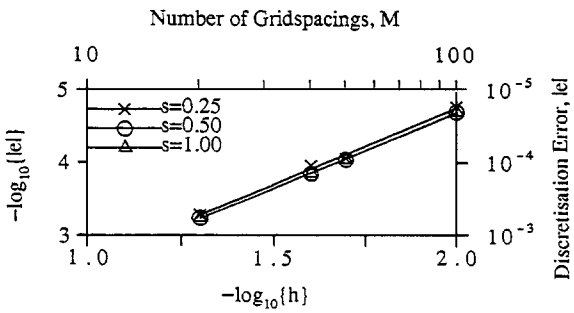

Fig. 6. Relation between error in u and grid spacing for the (5,5) N-H implicit method.

Fig. 5 shows that the accuracy of the Crank–Nicolson method is not changed as s increases. This is because of the fact that the leading error term in (19) does not depend on the value of s for the same value of h. It is clear from Fig. 7 that the worst results obtained when using the (9,9) N-H implicit method are better than the best results obtained when using the BTCS scheme, the Crank–Nicolson method or the (5,5) N-H implicit scheme.

Secondly, the fully implicit methods described in Section 2 are applied to solve Eqs. (1)–(7). The results obtained for μ with $h = 0.05$, $s = 1/2$, using the BTCS method, the Crank–Nicolson method, the (5,5) N-H implicit method and the (9,9) N-H implicit method, with $m(t)$ defined as in (54), and $\mu(t)$ considered to be unknown and found by (46), are shown in Table 2. Note that the

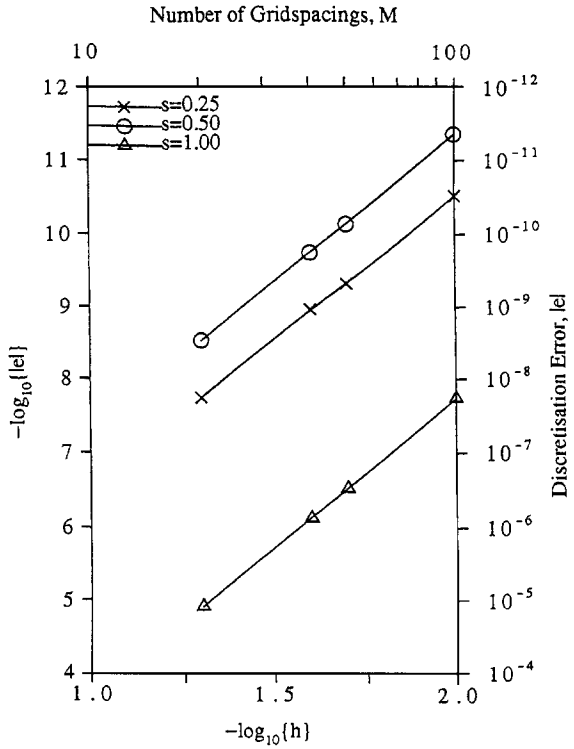

Fig. 7. Relation between error in u and grid spacing for the (9,9) N-H implicit method.

Table 2
Results for μ with $h = 0.05$, $s = 1/2$

t	Exact μ	BTCS Error	Crank–Nicolson Error	(5,5) N-H implicit Error	(9,9) N-H implicit Error
0.1	1.221403	-0.3×10^{-2}	-0.1×10^{-2}	0.1×10^{-2}	-0.5×10^{-5}
0.2	1.491825	-0.4×10^{-2}	-0.2×10^{-2}	0.2×10^{-2}	-0.6×10^{-5}
0.3	1.822119	-0.4×10^{-2}	-0.2×10^{-2}	0.2×10^{-2}	-0.7×10^{-5}
0.4	2.225541	-0.5×10^{-2}	-0.2×10^{-2}	0.2×10^{-2}	-0.8×10^{-5}
0.5	2.718282	-0.7×10^{-2}	-0.3×10^{-2}	0.3×10^{-2}	-0.9×10^{-5}
0.6	3.320117	-0.8×10^{-2}	-0.4×10^{-2}	0.4×10^{-2}	-0.1×10^{-4}
0.7	4.055200	-0.1×10^{-1}	-0.4×10^{-2}	0.4×10^{-2}	-0.2×10^{-4}
0.8	4.953032	-0.1×10^{-1}	-0.5×10^{-2}	0.5×10^{-2}	-0.2×10^{-4}
0.9	6.049647	-0.1×10^{-1}	-0.7×10^{-2}	0.7×10^{-2}	-0.3×10^{-4}
1.0	7.389056	-0.2×10^{-1}	-0.8×10^{-2}	0.8×10^{-2}	-0.4×10^{-4}

errors with the (9,9) N-H implicit method are less than one-thousandth of the errors obtained using the other methods. Also note that the tolerance was chosen to be 0.005 for the BTCS method, the Crank–Nicolson and the (5,5) N-H method, and 0.000005 for the (9,9) N-H method.

When the absolute value of the error

$$e^n = \mu(nk) - \mu^n, \tag{58}$$

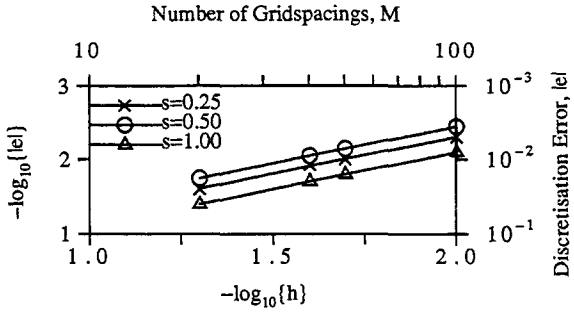

Fig. 8. Relation between error in μ and grid spacing for the (5,1) BTCS method.

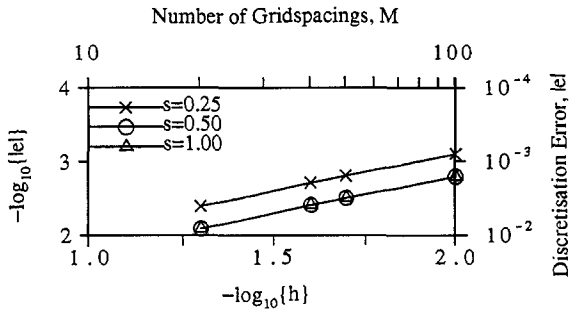

Fig. 9. Relation between error in μ and grid spacing for the (5,5) Crank–Nicolson method.

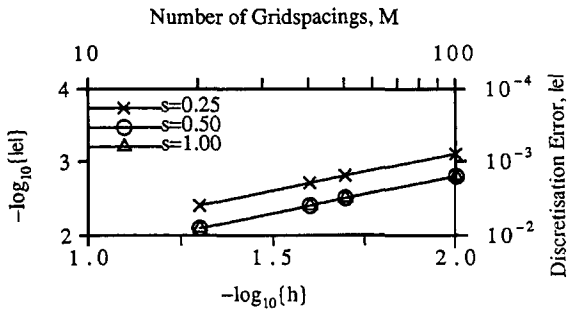

Fig. 10. Relation between error in μ and grid spacing for the (5,5) N-H implicit method.

at the point (0.5,0.5) at time $T = 1.0$ was graphed against h on a logarithmic scale for various values of s, it was found that the slopes of lines were always close to 1 for the BTCS formula, the Crank–Nicolson formula, and the (5,5) N-H implicit formula, but was close to 3 for the (9,9) N-H implicit formula (see Figs. 8–11). These results reflect the orders of convergence referred to earlier in Section 2. The interesting feature of these figures is that the minimum discretisation error produced by the (9,9) N-H implicit scheme is smaller than the maximum discretisation error obtain when using the BTCS method, the Crank–Nicolson method or the (5,5) N-H implicit scheme.

The absolute value of the discretization error at the point (0.5,0.5) at time $T = 1.0$ is graphed against the CPU time on a logarithmic scale for various values of s (see Figs. 12–15).

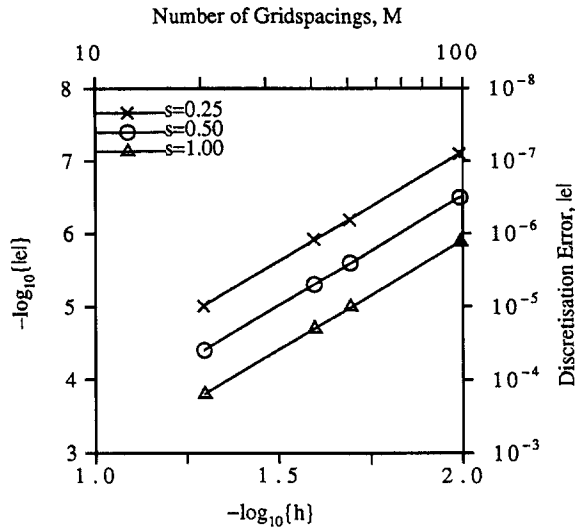

Fig. 11. Relation between error in μ and grid spacing for the (9,9) N-H implicit method.

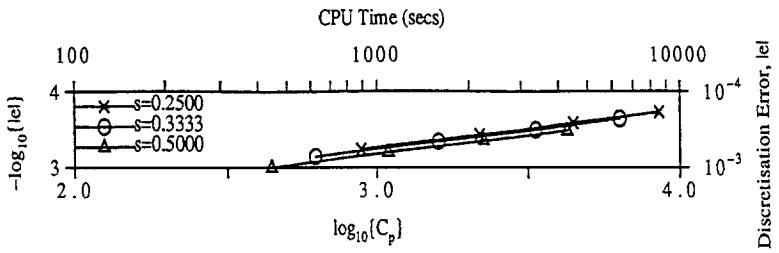

Fig. 12. Relation between the CPU times and the error for the (5,1) BTCS method.

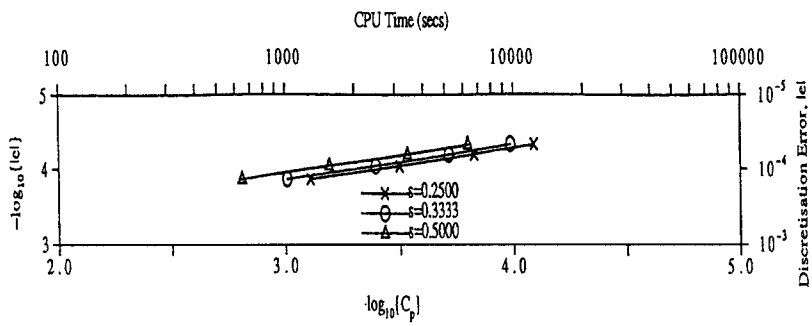

Fig. 13. Relation between the CPU times and the error for the (5,5) Crank–Nicolson method.

5. Conclusion

In this article three fully implicit methods, the Crank–Nicolson method, the (5,5) N-H implicit method and the (9,9) N-H implicit method, were applied to the two-dimensional diffusion equa-

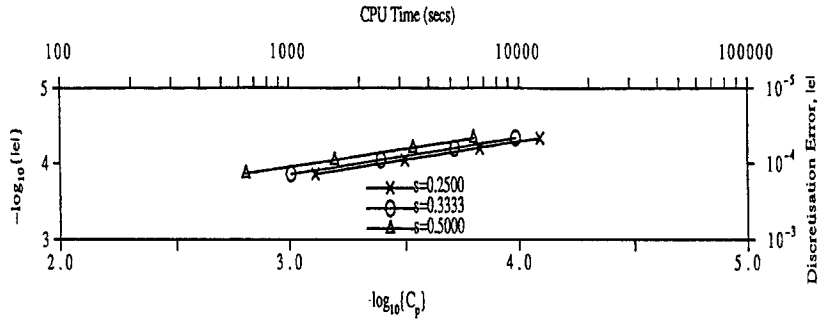

Fig. 14. Relation between the CPU times and the error for the (5,5) N-H implicit method.

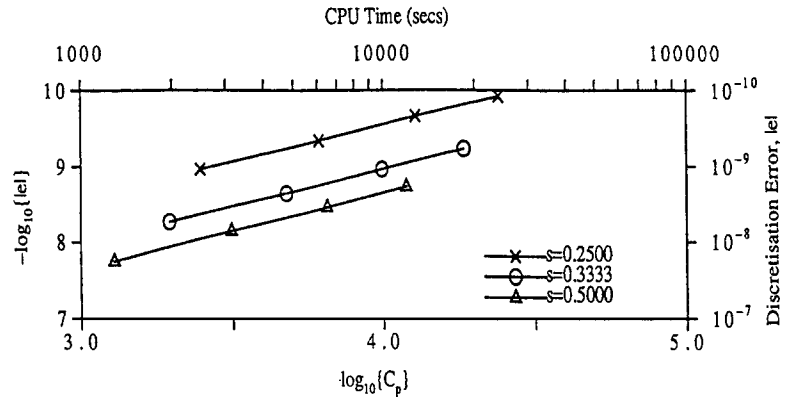

Fig. 15. Relation between the CPU times and the error for the (9,9) N-H implicit method.

tion. The latter worked very well for two dimensional nonlocal diffusion problem because of its fourth-order accuracy. This method seems particularly suited for parabolic partial differential equations with continuous boundary conditions. A comparison with the backward Euler scheme (BTCS) of [1] for the model problem clearly demonstrates the very high accuracy of the (9,9) N-H implicit scheme. The fully implicit methods developed in this report are unconditionally von Neumann stable. Note that the fully explicit schemes developed in [11] have greater restriction on stability, and are only useful over small time steps. The fully implicit (9,9) N-H scheme is slower than the others, but its fourth-order accuracy for every diffusion number is significant. As in the implicit schemes the values at interior grid points at new time levels cannot be obtained before computing the values at boundaries, an iteration procedure is employed to handle the nonlocal boundary condition.

The numerical test applied to these methods gives acceptable results and suggests convergence to exact solution when h goes to zero.

References

[1] J.R. Cannon, Y. Lin, A. Matheson, The solution of the diffusion equation subject to specification of mass, Appl. Anal. 50 (2) (1993) 1–11.

[2] J.R. Cannon, Y. Lin, S. Wang, An implicit finite difference scheme for the diffusion equation subject to mass specification, Internat. J. Eng. Sci. 28 (1990) 573–578.
[3] J.R. Cannon, A. Matheson, On the numerical solution of diffusion subject to the specification of mass, Internat. J. Eng. Sci. 31 (1993) 347–355.
[4] J.R. Cannon, S. Prez-Esteva, J. van der Hoek, A Galerkin procedure for the diffusion equation subject to the specification of mass, SIAM J. Numer. Anal. 24 (1987) 499–515.
[5] J.R. Cannon, J. van der Hoek, Implicit finite difference schemes for the diffusion of mass in porous media, Numerical Solutions of Partial Differential Equations, B.J. Noye (Ed.), North-Holland, Amsterdam, 1982, pp. 527–539.
[6] J.R. Cannon, J. van der Hoek, Diffusion subject to the specification of mass, J. Math. Anal. Appl. 115 (1986) 517–529.
[7] J. Crank, P. Nicolson, A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type, Proc. Cambridge Philos. Soc. 43 (1) (1947) 50–67.
[8] W.A. Day, A decreasing property of solutions of a parabolic equation with applications in thermoelasticity and other theories, Q. Appl. Math. 41 (1983) 475–486.
[9] C.F. Gerald, Applied Numerical Analysis, 5th ed., Reading, MA, Addison-Wesley, 1995.
[10] A.R. Mitchell, D.F. Griffiths, The Finite Difference Methods in Partial Differential Equations, Wiley, New York, 1980.
[11] B.J. Noye, M. Dehghan, J. van der Hoek, Explicit finite difference methods for two-dimensional diffusion equation with a non-local boundary condition, Int. J. Eng. Sci. 32 (1994) 1829–1834.
[12] B.J. Noye, K.J. Hayman, Implicit two-level finite-differences methods for the two-dimensional diffusion equation, Int. J. Comput. Math. 48 (1993) 219–228.
[13] S. Wang, A numerical method for the heat conduction subject to moving boundary energy specification, Numer. Heat Transfer 130 (1990) 35–38.
[14] S. Wang, Y. Lin, A finite difference solution to an inverse problem determining a control function in a parabolic partial differential equation, Inverse Problems 5 (1989) 631–640.
[15] R.F. Warming, B.J. Hyett, The modified equation approach to the stability and accuracy analysis of finite difference methods, J. Comput. Phys. 14 (2) (1974) 159–179.