Mathematical Problems in Engineering, 2002, Vol. 8(2), pp. 111–122 Taylor & Francis Taylor & Francis Group

Determination of an Unknown Parameter in a Semi-linear Parabolic Equation

MEHDI DEHGHAN

Department of Applied Mathematics, Faculty of Mathematics and Computer Science, Amirkabir University of Technology, No. 424, Hafez Avenue, Tehran, Iran

(Received 1 December 2001)

Some finite difference approximations are developed for an inverse problem of determining an unknown parameter $p(t)$ which is a coefficient of the solution u in a semi-linear parabolic partial differential equation subject to a boundary integral overspecification. The accuracy and efficiency of the procedures are discussed. Some computational results using the newly proposed numerical techniques are presented. CPU times needed for this problem are reported.

Key words: Three-dimensional diffusion-second-order accuracy, explicit scheme, implicit technique, numerical integration procedure, cubic spline function, numerical differentiation

AMS Subject Classification: 65N06, Mxx, D25

1 INTRODUCTION

Parabolic partial differential equations in two or three space dimensions with over-specified boundary data feature in the mathematical modelling of many important phenomena. While a significant body of knowledge about the theory and numerical methods for parabolic partial differential equations with classical boundary conditions has been accumulated, not much has been extended to parabolic partial differential equations with overspecified boundary data.

It is the purpose of the present paper to develop three different finite difference techniques for the numerical solution of the following three-dimensional semi-linear time dependent diffusion equation:

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} + p(t)w + \phi(x, y, z, t), \quad 0 \le x, y, z \le 1, \quad 0 < t \le T, \tag{1}$$

with initial condition

$$w(x, y, z, 0) = f(x, y, z), \quad 0 < x, y, z < 1, \tag{2}$$

ISSN 1024-123X print; ISSN 1563-5147 online

and boundary conditions

$$w(0, y, z, t) = g_0(y, z, t), \quad 0 < t \leq T, \quad 0 \leq y, z, \leq 1, \tag{3}$$

$$w(1, y, z, t) = g_1(y, z, t), \quad 0 < t \leq T, \quad 0 \leq y, z \leq 1, \tag{4}$$

$$w(x, 0, z, t) = h_0(x, z, t), \quad 0 < t \leq T, \quad 0 \leq x, z \leq 1, \tag{5}$$

$$w(x, 1, z, t) = h_1(x, z, t), \quad 0 < t \leq T, \quad 0 \leq x, z \leq 1, \tag{6}$$

$$w(x, y, 0, t) = k_0(x, y, t), \quad 0 < t \leq T, \quad 0 \leq x, y \leq 1, \tag{7}$$

$$w(x, y, 1, t) = k_1(x, y, t), \quad 0 < t \leq T, \quad 0 \leq x, y \leq 1, \tag{8}$$

subject to the integral identity

$$\int_0^1 \int_0^1 \int_0^1 w(x, y, z, t)\, \mathrm{d}x\, \mathrm{d}y\, \mathrm{d}z = E(t), \quad 0 < t \leq T, \tag{9}$$

where f, $g_0, g_1, h_0, h_1, k_1, k_2, \phi$ and E are known functions, while the functions w and p are unknown.

Parabolic partial differential equations with an unknown parameter have many important applications in science and engineering [1, 2, 4, 5, 7, 10–13].

Only in the last decade it has attracted some attention to the developement, analysis and implementation of the numerical techniques for solving the diffusion equation with non-classical boundary conditions.

The existence and uniqueness and continuous dependence on data for the solution pair (u, p) to the problem (1)–(9) and other similar problems have been studied in [3, 6, 14].

The three-dimensional inverse problem (1)–(9) can model certain type of physical problems. Equation (1) can be used to describe a heat transfer process with an unknown source parameter present. Equation (9) can be interpreted as a weighted thermal energy contained in a portion of the spatial domain, at time t. Thus the purpose of solving this three-dimensional parabolic inverse problem with energy overspecification is to identify the unknown parameter $p(t)$ that will produce at each time t a desired energy distribution in a portion of the spatial domain.

An overview of this paper is as follows:

The employed transformations and the method of evaluating the unknown parameter $p(t)$, is given in Section 2. Section 3 is devoted to the numerical finite difference schemes for the solution of (1)–(9). The numerical integration procedure used to overcome the over-specification boundary condition, is described in Section 4. This procedure utilizes the trapezoidal numerical integration rule, which is a simple second-order accurate scheme. In Section 4 we demonstrate how to construct a predicting-correcting mechanism and how to use it to advance our computation step by step. The numerical differentiation formula or the natural cubic spline functions employed to compute the values of the unknown parameter

$p(t)$, are described in Section 5. The numerical results produced by these schemes are presented in Section 6. The comparison of both accuracy and efficiency between the methods developed are also made in Section 6. Section 7 concludes this paper with a brief summary. Also some directions for further study are pointed out in this section.

2 THE NEW INVERSE PROBLEM

In this section we want to eliminate the term $p(t)w(x, y, z, t)$ in (1) by introducing the following transformations:

$$u(x, y, z, t) = w(x, y, z, t) \exp\left(\int_0^t p(s)\, ds \right), \tag{10}$$

$$r(t) = \exp\left(- \int_0^t p(s)\, ds \right). \tag{11}$$

So we have

$$w(x, y, z, t) = u(x, y, z, t) \exp\left(- \int_0^t p(s)\, ds \right), \tag{12}$$

$$p(t) = -\frac{r'(t)}{r(t)}. \tag{13}$$

where $r'(t) = dr(t)/dt$.

Transformations (10) and (11) allow us to eliminate the unknown term $p(t)$ from the equation (1) and to obtain a new nonclassic parabolic partial differential equation which is equivalent to the original inverse problem provided that some compatibility conditions are satisfied. Employing the above pair of transformations (10)–(11), we can write (1)–(9) as follows:

$$u_t = u_{xx} + u_{yy} + u_{zz} + r(t)\phi(x, y, z, t), \quad 0 \le x, y, z \le 1, \quad 0 < t \le T, \tag{14}$$

subject to

$$u(x, y, z, 0) = f(x, y, z), \quad 0 < x, y, z < 1, \tag{15}$$

$$u(0, y, z, t) = q_0(y, z, t), \quad 0 < t \le T, \quad 0 \le y, z \le 1, \tag{16}$$

$$u(1, y, z, t) = q_1(y, z, t), \quad 0 < t \le T, \quad 0 \le y, z \le 1, \tag{17}$$

$$u(x, 0, z, t) = z_0(x, z, t), \quad 0 < t \le T, \quad 0 \le x, z \le 1, \tag{18}$$

$$u(x, 1, z, t) = z_1(x, z, t), \quad 0 < t \le T, \quad 0 \le x, z \le 1, \tag{19}$$

$$u(x, y, 0, t) = d_0(x, y, t), \quad 0 < t \le T, \quad 0 \le x, y \le 1, \tag{20}$$

$$u(x, y, 1, t) = d_1(x, y, t), \quad 0 < t \le T, \quad 0 \le x, y \le 1, \tag{21}$$

and

$$\int_0^1 \int_0^1 \int_0^1 u(x, y, z, t)\, dx\, dy\, dz = r(t)E(t), \quad 0 < t \le T. \tag{22}$$

The latter is equivalent to

$$r(t) = \frac{\int_0^1 \int_0^1 \int_0^1 u(x, y, z, t)\, dx\, dy\, dz}{E(t)} \tag{23}$$

where $E(t) \neq 0$ and also we have:

$$q_0(y, z, t) = r(t)g_0(y, z, t), \quad 0 < t \leq T, \quad 0 \leq y, z \leq 1, \tag{24}$$

$$q_1(y, z, t) = r(t)g_1(y, z, t), \quad 0 < t \leq T, \quad 0 \leq y, z \leq 1, \tag{25}$$

$$z_0(x, z, t) = r(t)h_0(x, z, t), \quad 0 < t \leq T, \quad 0 \leq x, z \leq 1, \tag{26}$$

$$z_1(x, z, t) = r(t)h_1(x, z, t), \quad 0 < t \leq T, \quad 0 \leq x, z \leq 1, \tag{27}$$

$$d_0(x, y, 0, t) = r(t)k_0(x, y, t), \quad 0 < t \leq T, \quad 0 \leq x, y \leq 1, \tag{28}$$

$$d_1(x, y, 1, t) = r(t)k_1(x, y, t), \quad 0 < t \leq T, \quad 0 \leq x, y \leq 1. \tag{29}$$

Note that with the transformations used, $p(t)$ disappears and its role is represented implicitly by $r(t)$. So, we overcome the difficulties in handling with $p(t)$. This means that a predicting-correcting mechanism can be constructed easily.

It is worth mentioning that: if $r(t)$ can be made known, then we can choose any appropriate finite difference scheme to solve (14)–(21) numerically. So the key to the solution of our inverse problem lies in retrieving the information about the boundary values from the overspecification data (22), and this is the very place where the main difficulty exists [6].

3 THE DEVELOPED FINITE DIFFERENCE TECHNIQUES

The domain $[0, 1]^3 \times [0, T]$ is divided into an $M^3 \times N$ mesh with the spatial step size $h = 1/M$ in x, y and z directions and the time step size $l = T/N$, respectively.

Grid points (x_i, y_j, z_k, t_n) are defined by

$$x_i = ih, \quad i = 0, 1, 2, \ldots, M, \tag{30}$$

$$y_j = jh, \quad j = 0, 1, 2, \ldots, M, \tag{31}$$

$$z_k = kh, \quad k = 0, 1, 2, \ldots, M, \tag{32}$$

$$t_n = nl, \quad n = 0, 1, 2, \ldots, N, \tag{33}$$

in which M and N are integers. The notations $u^n_{i,j,k}, p^n$ and r^n are used for the finite difference approximations of $u(ih, jh, kh, nl), p(nl)$ and $r(nl)$, respectively.

The finite difference formula described in this section and to be applied at interior grid points in the solution domain are the (7,1) second-order implicit finite difference scheme or the (7,7) second-order implicit finite difference method or the (1,7) second-order explicit finite difference technique, which approximate the solution of the three-dimensional semi-linear diffusion equation.

Using the initial condition $u(x, y, z, 0) = f(x, y, z)$, $0 \le x, y, z \le 1$, equation (1) is solved approximately at the spatial points (x_i, y_j, z_k), commencing with initial values $u^0_{i,j,k} = f(x_i, y_j, z_k)$, $i, j, k = 0, 1, 2, \ldots, M$, and boundary values:

$$u^{n+1}_{0,j,k} = q_0(y_j, z_k, t_{n+1}), \tag{34}$$

$$u^{n+1}_{M,j,k} = q_1(y_j, z_k, t_{n+1}), \tag{35}$$

$$u^{n+1}_{i,0,k} = z_0(x_i, z_k, t_{n+1}), \tag{36}$$

$$u^{n+1}_{i,M,k} = z_1(x_i, z_k, t_{n+1}), \tag{37}$$

$$u^{n+1}_{i,j,0} = d_0(x_i, y_j, t_{n+1}), \tag{38}$$

$$u^{n+1}_{i,j,M} = d_1(x_i, y_j, t_{n+1}), \tag{39}$$

for $n = 0, 1, 2, \ldots, N-1$, where $q_0(x, z, t), q_1(x, z, t), z_0(y, z, t), z_1(y, z, t), d_0(x, y, t)$ and $d_1(x, y, t)$ are given by boundary conditions (24)–(29) and $r(t)$ will be found by the procedure described in Section 4.

3.1 The (7,1) Implicit Finite Difference Method

The seven-point implicit method for solving the three-dimensional partial differential equation (14) uses the following formula

$$\begin{aligned} &s_x\left(u^{n+1}_{i-1,j,k} + u^{n+1}_{i+1,j,k}\right) + s_y\left(u^{n+1}_{i,j-1,k} + u^{n+1}_{i,j+1,k}\right) \\ &\quad + s_z\left(u^{n+1}_{i,j,k-1} + u^{n+1}_{i,j,k+1}\right) - (1 + 2s_x + 2s_y + 2s_z)u^{n+1}_{i,j,k} \\ &\quad = -u^n_{i,j,k} - lr^n\phi^n_{i,j,k}, \end{aligned} \tag{40}$$

for $i, j, k = 1, 2, \ldots, M-1$, where $s_x = l/(\Delta x)^2$, $s_y = l/(\Delta y)^2$, $s_z = l/(\Delta z)^2$.

In the case $\Delta x = \Delta y = \Delta z = h$, we have

$$s_x = s_y = s_z = s = l/h^2, \tag{41}$$

and (40) becomes

$$s\left(u_{i-1,j,k}^{n+1}+u_{i+1,j,k}^{n+1}+u_{i,j-1,k}^{n+1}+u_{i,j+1,k}^{n+1}+u_{i,j,k-1}^{n+1}+u_{i,j,k+1}^{n+1}\right)-(1+6s)u_{i,j,k}^{n+1}=-u_{i,j,k}^{n}-lr^{n}\phi_{i,j,k}^{n}. \tag{42}$$

It is very easy to see that this procedure is unconditionally stable [9].

The resulting system of linear equations is strictly diagonally dominant, which guarantees that it is solvable.

The computational molecule of this method involves 7 grid points at the new time level and 1 at the old level. So in the following this will be referred to as the (7,1) formula.

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

$$\frac{\partial u}{\partial t}-\frac{\partial^2 u}{\partial x^2}-\frac{\partial^2 u}{\partial y^2}-\frac{\partial^2 u}{\partial z^2}-r\phi-\frac{(\Delta x)^2}{12}(1+6s_x)\frac{\partial^4 u}{\partial x^4}-\frac{(\Delta y)^2}{12}(1+6s_y)\frac{\partial^2 u}{\partial y^4}-\frac{(\Delta z)^2}{12}(1+6s_z)\frac{\partial^4 u}{\partial z^4}+O\left\{(\Delta x)^4,(\Delta y)^4,(\Delta z)^4\right\}=0. \tag{43}$$

So this scheme is second-order accurate in the spatial grid size. However there is no set of values for which the method will be fourth-order accurate.

3.2 The (1,7) Explicit Finite Difference Method

The (1,7) explicit method uses the forward difference approximation for the time derivative and the centred-difference approximation for the space derivatives applied at the point (ih, jh, kh, nl).

So it follows that:

$$u_{i,j,k}^{n+1}=s_x\left(u_{i-1,j,k}^{n}+u_{i+1,j,k}^{n}\right)+s_y\left(u_{i,j-1,k}^{n}+u_{i,j+1,k}^{n}\right)+s_z\left(u_{i,j,k-1}^{n}+u_{i,j,k+1}^{n}\right)+(1-2s_x-2s_y-2s_z)u_{i,j,k}^{n}+lr^{n}\phi_{i,j,k}^{n} \tag{44}$$

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

In the special case $\Delta x = \Delta y = \Delta z = h$, we will have:

$$u_{i,j,k}^{n+1}=s\left(u_{i-1,j,k}^{n}+u_{i+1,j,k}^{n}+u_{i,j-1,k}^{n}+u_{i,j+1,k}^{n}+u_{i,j,k-1}+u_{i,j,k+1}^{n}\right)+(1-6s)u_{i,j,k}^{n}+lr^{n}\phi_{i,j,k}^{n}. \tag{45}$$

Values of $u_{i,j,k}^{n+1}$ on the boundaries $x = 0, 1$, $y = 0, 1$ and $z = 0, 1$, are provided by the boundary conditions (34)–(39), at the appropriate grid points.

The range of stability for this procedure is

$$0 < s \leq \frac{1}{6} \tag{46}$$

It can be seen that the modified equivalent partial differential equation of the (1,7) formula (45) is [15]:

$$\begin{aligned}&\frac{\partial u}{\partial t}-\frac{\partial^2 u}{\partial x^2}-\frac{\partial^2 u}{\partial y^2}-\frac{\partial^2 u}{\partial z^2}-r\phi+\frac{(\Delta x)^2}{12}(6s_x-1)\frac{\partial^4 u}{\partial x^4}\\&\quad+\frac{(\Delta y)^2}{12}(6s_y-1)\frac{\partial^4 u}{\partial y^4}+\frac{(\Delta z)^2}{12}(6s_z-1)\frac{\partial^4 u}{\partial z^4}\\&\quad+\frac{(\Delta x)(\Delta y)}{4}s_x s_y\frac{\partial^4 u}{\partial x^2\partial y^2}\\&\quad+\frac{(\Delta x)(\Delta z)}{4}s_x s_z\frac{\partial^4 u}{\partial x^2\partial z^2}+\frac{(\Delta y)(\Delta z)}{4}s_y s_z\frac{\partial^4 u}{\partial y^2\partial z^2}\\&\quad+O\{(\Delta x)^4,(\Delta y)^4,(\Delta z)^4\}=0.\end{aligned}\tag{47}$$

The formula (47) is second-order accurate for all $s>0$ as it can be seen by the modified equivalent partial differential equation analysis.

In following this scheme will be referred to as the (1,7) explicit technique, because the computational molecule of this method involves 1 grid point at the new time level and 7 at the old level.

3.3 The Implicit (7,7) Finite Difference Scheme

We replace all spatial derivatives with the average of their values at the n and $n+1$ time levels and then substitute the centred-difference forms for all derivatives. So we get the implicit (7,7) formula:

$$\begin{aligned}&s_x\left(u^{n+1}_{i-1,j,k}+u^{n+1}_{i+1,j,k}\right)-2(1+s_x+s_y+s_z)u^{n+1}_{i,j,k}\\&\quad+s_y\left(u^{n+1}_{i,j-1,k}+u^{n+1}_{i,j+1,k}\right)+s_z\left(u^{n+1}_{i,j,k-1}+u^{n+1}_{i,j,k+1}\right)\\&\quad=-s_x\left(u^{n}_{i-1,j,k}+u^{n}_{i+1,j,k}\right)-2(1-s_x-s_y-s_z)u^{n}_{i,j,k}\\&\quad\quad-s_y\left(u^{n}_{i,j-1,k}+u^{n}_{i,j+1,k}\right)-s_z\left(u^{n}_{i,j,k-1}+u^{n}_{i,j,k+1}\right)\\&\quad\quad+\frac{l}{4}(r^n+r^{n+1})\left(\phi^n_{i,j,k}+\phi^{n+1}_{i,j,k}\right).\end{aligned}\tag{48}$$

In the case $\Delta x=\Delta y=\Delta z=h$, the new finite-difference equation is

$$\begin{aligned}&s\left(u^{n+1}_{i-1,j,k}+u^{n+1}_{i+1,j,k}+u^{n+1}_{i,j-1,k}+u^{n+1}_{i,j+1,k}+u^{n+1}_{i,j,k+1}+u^{n+1}_{i,j,k+1}\right)\\&\quad-2(1+3s)u^{n+1}_{i,j,k}=-s\left(u^{n}_{i,j-1,k}+u^{n}_{i,j+1,k}+u^{n}_{i-1,j,k}+u^{n}_{i+1,j,k}+u^{n}_{i,j,k-1}+u^{n}_{i,j,k+1}\right)\\&\quad-2(1-3s)u^{n}_{i,j,k}-\frac{l}{4}(r^n+r^{n+1})\left(\phi^n_{i,j,k}+\phi^{n+1}_{i,j,k}\right).\end{aligned}\tag{49}$$

This scheme is stable for every diffusion number $s>0$.

In following this will be referred to as the (7,7) implicit technique, because the computational molecule of this technique involves 7 grid points at the new time level and 7 at the old level.

The resulting system of linear equations is strictly diagonally dominant, which guarantees that it is solvable.

The modified equivalent equation of the (7,7) implicit formula (49) is as follows [15]

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} - \frac{\partial^2 u}{\partial z^2} - r\phi - \frac{(\Delta x)^2}{12}\frac{\partial^4 u}{\partial x^4} - \frac{(\Delta y)^2}{12}\frac{\partial^4 u}{\partial y^4} - \frac{(\Delta z)^2}{12}\frac{\partial^4 u}{\partial z^4} + O\{(\Delta x)^4, (\Delta y)^4, (\Delta z)^4\} = 0. \tag{50}$$

This scheme 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. Also, note that it has a leading error term which is independent of s.

4 TREATMENT OF THE NON-CLASSICAL BOUNDARY CONDITION

The treatment of the integral term in the parabolic partial differential equations with non-classical boundary conditions is not an easy task. The presence of an integral term in a boundary condition can greatly complicate the application of standard numerical techniques such as finite differences, finite elements, spectral methods, etc. It is therefore important to be able to convert the overspecification boundary value problems to a more desirable form, to make them more widely applicable to problems of practical interest.

The accuracy of the quadrature must be compatible with that of the discretization of the differential equation.

For this purpose, we approximate the integral in (9) numerically by the trapezoidal rule [16]. The reason we choose the trapezoidal numerical integration procedure is because it is very simple and has a truncation error of order 2. Note that the accuracy of the solution will not reduce under this method.

In order to use (23) we evaluate the integral $\int_0^1 \int_0^1 \int_0^1 u(x, y, z, t)\,dx\,dy\,dz$ in a discrete form:

Firstly, the triple integral in (22) is approximated using the trapezoidal numerical integration rule [8] to give

$$\int_0^1 \int_0^1 \int_0^1 u(x, y, z, t_n)\,dx\,dy\,dz \simeq \frac{h^3}{8} \sum_{i=0}^{M} \sum_{j=0}^{M} \sum_{k=0}^{M} d_i d_j d_k u_{i,j,k}^n, \tag{51}$$

where

$$d_0 = d_M = 1, \quad d_i = 2, \quad i = 1, 2, \ldots, M-1. \tag{52}$$

As we mentioned already, given an initial guess of $r(t)$, several numerical schemes can be used to solve (14)–(21).

We employ three second-order finite difference schemes which are based on the (1,7) explicit finite difference scheme, the (7,7) implicit technique and the (7,1) implicit finite difference formula.

The numerical treatment we propose to overcome the non-linear boundary over-specification, is a predictor–corrector type approach. We will use one of the finite difference schemes discussed in Section 3, to carry out the numerical solutions level by level in the time direction. Given the finite difference approximation values of $u(x, y, z, t)$ and $r(t)$ at the nth

level $n = 0, 1, 2, \ldots$, we first assign an appropriate initial guess to $r(t)$ for the $(n+1)$st level. With this initial guess, (14) can be solved to yield the corresponding finite difference approximation values for $u(x, y, z, t)$ at the $(n+1)$st level. If the solution satisfies the over-specified condition (22) within a prescribed tolerance, then the corresponding values for $u(x, y, z, t)$ and $r(t)$ are accepted as correct numerical solutions for the two functions at the $(n+1)$st level. Otherwise, a new guess for $r(t)$ is generated from (23). Computations are then repeated with the new guess of $r(t)$. This predicting–correcting process is repeated until finally (22) is satisfied with the desired accuracy and, thus, we advance to the next higher level [13].

Using (51), we can obtain the numerical approximation for $\int_0^1 \int_0^1 \int_0^1 u(x, y, z, t^n)\, dx\, dy\, dz$. Let us denote the result by $v^{n(m)}$, where (m) means the numerical integration is done with the approximation of $v^{(n+1)(m)}$ corresponding to $r^{n+1(m)}$ over $[0, 1]^3$.

Denoting

$$v^{n(m)} \simeq \frac{h^3}{8} \sum_{i=0}^{M} \sum_{j=0}^{M} \sum_{k=0}^{M} d_i d_j d_k u_{i,j,k}^{n(m)}, \tag{53}$$

and using this notation, the finite difference form of the corrector (23) can be written as:

$$r^{n+1(m+1)} = \frac{v^{n+1(m)}}{E^{(n+1)}}. \tag{54}$$

5 IDENTIFYING THE MAIN UNKNOWN PARAMETER

After we obtained the values of $r^n s$, we can convert them into the corresponding values of $p^n s$ through the inverse transformations (10)–(11).

This can be done in two different ways, numerical differentiation or interpolating the data of $r^n s$ with smooth functions and then taking derivative.

The finite difference form of (13) is

$$p^n = -\frac{(r')^n}{r^n}. \tag{55}$$

For the direct numerical differentiation, the following formula is used:

$$(r')^n = \frac{r^{n+1} - r^{n-1}}{2l}, \quad n = 2, 3, \ldots, N-1, \tag{56}$$

$$(r')^1 = \frac{-3r^1 + 4r^2 - r^3}{2l}, \tag{57}$$

$$(r')^N = \frac{3r^N - 4r^{N-1} + r^{N-2}}{2l}. \tag{58}$$

For the natural cubic spline function interpolation, in each subinterval $[t^{j-1}, t^j]$, we use a cubic polynomial

$$\begin{aligned} c^j(t) = \frac{1}{6\Delta t}\left[(t^j - t)^3 m^{j-1} + (t - t^{j-1})^3 m^j\right] + \frac{1}{\Delta t}\left[(t^j - t) r^{j-1}\right. \\ \left. + (t - t^{j-1}) r^j\right] - \frac{\Delta t}{6}\left[(t^j - t) m^{j-1} + (t - t^{j-1}) m^j\right], \end{aligned} \tag{59}$$

where $m^1 = m^N = 0$ and m^{j-1}, m^j, m^{j+1} satisfy the following tridiagonal system of linear equations:

$$\frac{h}{6}m^{j-1} + \frac{2h}{3}m^j + \frac{h}{6}m^{j+1} = \frac{1}{h}\left(r^{j+1} - 2r^j + r^{j+1}\right), \tag{60}$$

for $j = 2, 3, \ldots, N-1$.

Finally, we use the least squares data fitting method to find a smooth functional approximation for each of the data sets [8,14].

6 NUMERICAL RESULTS

In this section we present one example to support our theoretical results. A problem for which the exact solution is known is now used to test the different methods described in this report for solving the three-dimensional parabolic partial differential equation with an unknown source parameter. This is applied to solve (1)–(9) with $p(t)$ and w unknown.

Consider (1)–(9) with

$$\phi(x, y, z, t) = \left(\frac{7\pi^2}{8} - 10t\right)\exp(t)\sin\frac{\pi}{4}(x + 2y + 3z), \tag{61}$$

$$g_0(0, y, z, t) = \exp(t)\sin\frac{\pi}{4}(2y + 3z), \tag{62}$$

$$g_1(1, y, z, t) = \exp(t)\sin\frac{\pi}{4}(1 + 2y + 3z), \tag{63}$$

$$h_0(x, 0, z, t) = \exp(t)\sin\frac{\pi}{4}(x + 3z), \tag{64}$$

$$h_1(x, 1, z, t) = \exp(t)\sin\frac{\pi}{4}(x + 2 + 3z), \tag{65}$$

$$k_0(x, y, 0, t) = \exp(t)\sin\frac{\pi}{4}(x + 2y), \tag{66}$$

$$k_1(x, y, 1, t) = \exp(t)\sin\frac{\pi}{4}(x + 2y + 3), \tag{67}$$

$$E(t) = \frac{32\sqrt{2}}{3\pi^3}\exp(t), \tag{68}$$

$$f(x, y, z) = \sin\frac{\pi}{4}(x + 2y + 3z), \tag{69}$$

for which the exact solution is

$$w(x, y, z, t) = \exp(t)\sin\frac{\pi}{4}(x + 2y + 3z), \tag{70}$$

and

$$p(t) = 1 + 10t. \tag{71}$$

TABLE I Results for w from Test 1 with $h = \frac{1}{50}$, $s = \frac{1}{6}$.

x	y	z	*Exact u*	*(1,7) Explicit error*	*(7,1) Implicit error*	*(7,7) Implicit error*
0.1	0.1	0.1	1.2340741	5.2×10^{-3}	5.4×10^{-3}	5.7×10^{-3}
0.2	0.2	0.2	2.1991362	5.2×10^{-3}	5.5×10^{-3}	5.8×10^{-3}
0.3	0.3	0.3	2.6848153	5.3×10^{-3}	5.5×10^{-3}	5.9×10^{-3}
0.4	0.4	0.4	2.5852397	5.2×10^{-3}	5.3×10^{-3}	5.9×10^{-3}
0.5	0.5	0.5	1.9221155	5.4×10^{-3}	5.5×10^{-3}	5.8×10^{-3}
0.6	0.6	0.6	0.8399953	5.3×10^{-3}	5.4×10^{-3}	5.7×10^{-3}
0.7	0.7	0.7	−0.4252330	5.5×10^{-3}	5.4×10^{-3}	5.7×10^{-3}
0.8	0.8	0.8	−1.5977660	5.2×10^{-3}	5.3×10^{-3}	5.7×10^{-3}
0.9	0.9	0.9	−2.4220068	5.1×10^{-3}	5.6×10^{-3}	5.5×10^{-3}

TABLE II Results for p from Test 1 with $h = \frac{1}{50}$, $s = \frac{1}{6}$.

t	*Exact p*	*(1,7) Explicit error*	*(7,1) Implicit error*	*(7,7) Implicit error*
0.1	1.500000	9.0×10^{-3}	9.4×10^{-3}	9.2×10^{-3}
0.2	2.000000	9.1×10^{-3}	9.3×10^{-3}	9.0×10^{-3}
0.3	2.500000	9.2×10^{-3}	9.4×10^{-3}	9.1×10^{-3}
0.4	3.000000	9.1×10^{-3}	9.2×10^{-3}	9.2×10^{-3}
0.5	3.500000	9.1×10^{-3}	9.2×10^{-3}	9.3×10^{-3}
0.6	4.000000	8.9×10^{-3}	9.1×10^{-3}	9.2×10^{-3}
0.7	4.500000	8.9×10^{-3}	9.0×10^{-3}	9.2×10^{-3}
0.8	5.000000	9.1×10^{-3}	9.1×10^{-3}	9.1×10^{-3}
0.9	5.500000	9.2×10^{-3}	9.2×10^{-3}	9.0×10^{-3}
1.0	6.000000	9.1×10^{-3}	9.4×10^{-3}	9.0×10^{-3}

The results obtained for $w_{i,j,k}^N$ at $T = 1.0$, computed for $h = 0.02, s = \frac{1}{6}$ using the (1,7) explicit finite difference method, the (7,7) implicit technique and the (7,1) implicit finite difference scheme, are listed in Table I.

Although the (7,7) implicit method provides almost the same answer as the (1,7) explicit scheme, it involves more computational work.

The results obtained for p^n with $h = 0.02, s = \frac{1}{6}$, using the finite difference methods developed, with $p(t)$ defined as in (71) and it was considered to be unknown and found by (55) are shown in Table II.

The CPU time for the (1,7) explicit method was 91.57 s and for the (7,7) implicit scheme was 2005.6 s while the time for the (7,1) implicit formula was 1917.8 s.

The reason that we chose $s = \frac{1}{6}$ for the test problem used, is that this value is in the range of stability of the (1,7) explicit scheme discussed in Section 3. This forces the two unconditionally stable finite difference methods *i.e.* the (7,7) implicit technique and the (7,1) implicit scheme to use a much smaller time step than is necessary. But, overall the numerical results showed the computational superiority of the (1,7) explicit finite difference scheme for the three-dimensional diffusion equation with an unknown source parameter.

7 CONCLUDING REMARKS

In this paper three numerical methods were applied to a three-dimensional diffusion equation with an unknown parameter. The proposed numerical schemes solved this model quite

satisfactory. Using the (1,7) explicit finite difference method or the (7,7) implicit technique or the (7,1) implicit finite difference method for the three-dimensional semi-linear diffusion problem with appropriate treatment on the unknown source parameter describe our model well. The (1,7) explicit procedure is very simple to implement and economical to use. It is very efficient and needs less CPU time than the implicit methods. In order to overcome the integral overspecification boundary condition the trapezoidal numerical integration rule, which is a second-order accurate scheme, was employed. Note that the (1,7) explicit finite difference scheme discussed have greater restriction on stability, and is only useful over small time steps. The implicit (7,7) finite difference scheme employed is slower than the (7,1) implicit scheme. Note that the stability of the (7,1) scheme and the (7,7) implicit finite difference method for every diffusion number is significant. Overall, a useful comparison between the different finite difference techniques discussed in this article, showed the efficiency of the (1,7) explicit technique compared to tedious work required by the (7,7) finite difference method or the (7,1) implicit technique. The numerical test applied to these methods gives acceptable results and suggests convergence to exact solution when h goes to zero. Also we observed second-order accuracy, which is the best we can expect from the schemes and the formula we used. In closing we mention the question of whether these results can be generalized to cover other three-dimensional diffusion problems with an unknown source parameter, such as the case in which the inhomogenous term $F = F(x, y, z, t, w, w_x, w_y, w_z, p)$, has more complicated structure. In this case, the soultions will be more difficult to compute.

References

[1] Cannon, J. R. and Yin, H. M. (1989) On a class of non-classical parabolic problems, *J. Differential Equations*, **79**, 266–288.

[2] Cannon, J. R. and Lin, Y. (1988) Determination of parameter $p(t)$ in Holder classes for some semilinear parabolic equations, *Inverse Problems*, **4**, 595–606.

[3] Cannon, J. R. and Lin, Y. (1990) An inverse problem of finding a parameter in a semi-linear heat equation, *J. Math. Anal. Appl.*, **145**(2), 470–484.

[4] Cannon, J. R. and Yin, H. M. (1990) Numerical solution of some parabolic inverse problems, *Numerical Methods for Partial Differential Equations*, **2**, 177–191.

[5] Cannon, J. R. and Lin, Y. (1988) Determination of parameter $p(t)$ in some quasi-linear parabolic differential equations, *Inverse Problems*, **4**, 35–45.

[6] Cannon, J. R., Lin, Y. and Wang, S. (1992) Determination of source parameter in parabolic equations, *Meccanica*, **27**, 85–94.

[7] Dehghan, M. (2001) Implicit solution of a two-dimensional parabolic inverse problem with temperature overspecification, *J. Comput. Anal. Appl.*, **3**(4), 383–398.

[8] Gerald, C. F. and Wheatley, P. O. (1994) *Applied Numerical Analysis* (5th ed). Addison-Wesley.

[9] Mitchell, A. R. and Griffiths, D. F. (1980) *The Finite Difference Methods in Partial Differential Equations*. T. Wiley.

[10] Prilepko, A. I. and Orlovskii, D. G. (1985) Determination of the evolution parameter of an equation and inverse problems of mathematical physics, I, *Differential Equations*, **21**, 119–129, and part II, **21**, 649–701.

[11] Prilepko, A. I. and Soloev, V. V. (1987) Solvability of the inverse boundary value problem of finding a coefficient of a lower order term in a parabolic equation, *Differential Equations*, **23**(1), 136–143.

[12] Rundell, W. (1980) Determination of an unknown non-homogenous term in a linear partial differential equation from overspecified boundary data, *Appl. Anal.*, **10**, 231–242.

[13] Wang, S. (1992) Numerical solutions of two inverse problems for identifying control parameters in 2-dimensional parabolic partial differential equations, *Modern Developments in Numerical Simulation of Flow and Heat Transfer*, HTP-Vol.194, 11–16.

[14] Wang, S. and Lin, Y. (1989) A finite difference solution to an inverse problem determining a control function in parabolic partial differential equations, *Inverse Problems*, **5**, 631–640.

[15] Warming, R. F. and Hyett, B. J. (1974) The modified equation approach to the stability and accuracy analysis of finite-difference methods, *Journal of Computational Physics*, **14**(2), 159–179.

[16] Dehghan, M. (1999) Implicit solution locally one-dimensional methods for two-dimensional diffusion with a non-local boundary condition, *Math. Comput. Simulation*, **49**, 331–349.