Numerical solution of a quasilinear parabolic problem

P C JAIN and M K KADALBAJOO
Department of Mathematics, Indian Institute of Technology, Bombay 400 076

MS received 22 February 1979; revised 30 July 1979

Abstract. A combined approach of linearisation techniques and finite difference method is presented for obtaining the numerical solution of a quasilinear parabolic problem. The given problem is reduced to a sequence of linear problems by using the Picard or Newton methods. Each problem of this sequence is approximated by Crank-Nicolson difference scheme. The solutions of the resulting system of algebraic equations are obtained by using Block-Gaussian elimination method. Two numerical examples are solved by using both linearisation procedures to illustrate the method. For these examples, the Newton method is found to be more effective, especially when the given nonlinear problem has steep gradients.

Keywords. Parabolic equation; numerical method.

1. Introduction

Consider the quasilinear parabolic differential equation

\[ u_t = u_{xx} + u_{yy} + \phi(x, y, t, u, u_x) \quad (x, y) \in R, \ 0 < t \leq T \tag{1} \]

subject to the initial condition

\[ u(x, y, 0) = f(x, y), \ (x, y) \in R \tag{2} \]

and the boundary conditions

\[ u(x, y, t) = g(x, y, t), \ (x, y) \in \partial R, \ 0 < t \leq T, \tag{3} \]

where R is a bounded plane region with a smooth boundary \( \partial R \), and \( f(x, y) \), \( g(x, y, t) \) are given functions of their arguments defined in the respective domains. Existence and uniqueness theorems for the solution of this problem are known [5] under various assumptions on the functions \( \phi, f \) and \( g \). We assume that the conditions are satisfied so that the solution \( u(x, y, t) \) exists and is unique with suitable regularity properties in the domain. Many authors [4, 6] have tackled this problem by replacing the partial derivatives by finite differences and solving the resulting system of linear or nonlinear equations by iterative methods. An excellent presentation of such methods is given in a survey paper by Douglas [3].
We have used combined approach of linearisation and finite difference method for solving this problem. The problem is linearised by using Picard’s method. This introduces a sequence of functions \{u^{(n)}\} which satisfy the boundary conditions specified for \( u \) and linear partial differential equations

\[ u_t^{(n+1)} = u_{xx}^{(n+1)} + u_{yy}^{(n+1)} + \phi [x, y, t, u^{(n)}], \quad (n = 0, 1, 2, \ldots), \]

\( u^{(0)} \) being the initial guess.

If \( \phi \) is a differentiable function, we can use Newton-linearisation which again introduces a sequence of functions \{u^{(n)}\} that satisfy the boundary conditions specified for \( u \) and linear partial differential equations

\[ u_t^{(n+1)} = u_{xx}^{(n+1)} + u_{yy}^{(n+1)} + \phi [x, y, t, u^{(n)}], \quad (n = 0, 1, 2, \ldots), \]

\[ + \left[ u^{(n+1)} - u^{(n)} \right] \frac{\partial \phi}{\partial u} + \left[ u_{xx}^{(n+1)} - u_{xx}^{(n)} \right] \frac{\partial \phi}{\partial u}, \quad (n = 0, 1, 2, \ldots), \]

\( (u^{(0)} \) being the initial guess), where the partial derivatives of \( \phi \) are evaluated at the \( n \)-th step. When the sequence \{u^{(n)}\} converges, the convergence is linear in the first case and quadratic in the second case.

The above sequence of linear problem equation (4) or (5) is then approximated by Crank-Nicolson difference scheme and the resulting algebraic problem is solved by Block-Gaussian elimination method. By using such a method, one can avoid the nonlinearity as well as the occurrence of two point boundary value problem. To be specific, we use the linearized version [equation (4)] of the given problem for the analysis of our method. It may be mentioned that the similar type of procedure can be applied to equation (5).

2. Crank-Nicolson difference scheme

Applying the Crank-Nicolson difference scheme to equation (4), we have

\[ \left[ u(x, y, t + \Delta t) - u(x, y, t) \right]^{(n+1)} \]

\[ = \frac{1}{2} [u_{xx}(x, y, t + \Delta t) + u_{yy}(x, y, t + \Delta t) + u_{xx}(x, y, t) + u_{yy}(x, y, t)]^{(n+1)} + \frac{1}{2} \{ u(x, y, t + \Delta t) + u(x, y, t) \}^{(n)}, \]

\[ + \frac{1}{2} \{ u_{xx}(x, y, t + \Delta t) + u_{xx}(x, y, t) \}^{(n)} \].

(6)

For simplicity, we restrict our attention to the case where the given region \( R \) is a rectangle, \( 0 \leq x \leq a, 0 \leq y \leq b \). We take \( [a = Nh], \quad [b = Mh] \quad (N, M \text{ positive integers}) \) and space increments \( h = \Delta x = \Delta y \). We use the notations

\[ x_i = ih, \quad y_j = jh, \quad t_l = l \Delta, \quad u_{xt}(t_l) = u(ih, jh, l \Delta). \]

(7)

Replacing the partial derivatives in equation (6) by the standard finite difference approximations, we get the matrix-vector equation

\[ u_{i+1}^{(n+1)} (t_l+1) = u_{i+1}^{(n)} (t_l) + r [u_{i+1}^{(n+1)} (t_{l+1}) - 2u_{i+1}^{(n+1)} (t_{l+1}) + u_{i+1}^{(n+1)} (t_{l+1})]

\[ + u_{i-1}^{(n+1)} (t_{l+1}) - Q u_{i+1}^{(n+1)} (t_{l+1}) + u_{i-1}^{(n)} (t_l) \]
Quasilinear parabolic problem

\[-2u_t^{(n)}(t_i) + u_{xx}^{(n)}(t_i) - Qu_x^{(n)}(t_i) + 2S_i\]
\[+ \Delta \phi_t^{(n)}(t_{i+1}); \quad (i = 1, 2, \ldots, N - 1), \quad (8)\]

where the upper index \(n\) denotes the minimum number of iterations required to obtain an acceptable approximation to \(u_{t,i} (t_i)\). The criterion for acceptance of an approximation will be discussed later when we study some test examples.

Here we have used the following notations:

The vectors
\[u_i(t_{i+1}) = \begin{bmatrix} u_{i,1}(t_{i+1}) \\ u_{i,2}(t_{i+1}) \\ \vdots \\ u_{i,M-1}(t_{i+1}) \end{bmatrix}; \quad i = 1, 2, \ldots, N - 1 \]

\[S_i = [s_{i,j}]; \text{ where } s_{i,j} = \begin{cases} u_{i,0} & j = 1 \\ u_{i,j} & j = M - 1 \\ 0 & \text{otherwise} \end{cases} \]

\[\phi_i(t_{i+1}) = \begin{bmatrix} \phi_{i,1}(t_{i+1}) \\ \phi_{i,2}(t_{i+1}) \\ \vdots \\ \phi_{i,M-1}(t_{i+1}) \end{bmatrix}; \quad i = 1, 2, \ldots, N - 1 \]

with \(r = \Delta/2h^2\).

The matrix
\[Q = (q_{ij}); \text{ where } q_{ij} = \begin{cases} 2 & i = j \\ -1 & |i-j| = 1 \\ 0 & \text{otherwise} \end{cases} \]

Equation (8) is second order accurate in both space and time. It may be noted that the vector \(S_i\) remains the same for each time level due to the boundary conditions, and the vector \(\phi_i(t_{i+1})\) is a known vector.

The matrix-vector equation (8) is solved by the Block–Gaussian elimination method [1].

3. Test examples

To test our method, we solve the following examples:

Example 1:
\[u_t + uu_x = vu_x; \quad 0 < x < 1, \quad 0 < t \leq T, \quad (9)\]

subject to the initial and boundary conditions
\[u(x, 0) = \sin \pi x; \quad 0 \leq x \leq 1 \]
\[u(0, t) = u(1, t) = 0; \quad 0 < t \leq T. \quad (10)\]

Equation (9) is Burgers' equation well-known in the literature. We solve this problem by our method using both Picard as well as Newton linearisations. The linearised versions of equation (9) can be written as
\[u_t^{(n+1)} + u_x^{(n)}u_x^{(n)} = vu_x^{(n+1)}, \quad \text{(Picard)} \quad (11)\]
and

\[ u^{(n+1)} + u^{(n)} u_x^{(n)} + [u^{(n+1)} - u^{(n)}] u_{xx}^{(n)} + [u_{xx}^{(n+1)} - u_{xx}^{(n)}] u_x^{(n)} = -u_{xx}^{(n+1)}, \]  
\[ \quad \text{(Newton),} \quad (n = 0, 1, 2, \ldots) \]

subject to the initial and boundary conditions

\[ u^{(n+1)}(x, 0) = \sin \pi x; \quad 0 \leq x \leq 1 \]
\[ u^{(n+1)}(0, t) = u^{(n+1)}(1, t) = 0; \quad 0 < t \leq T. \] (13)

We apply the Crank–Nicolson scheme and replace the partial derivatives by standard finite differences and express the solution of the resulting equations in the form

\[ u^{(n+1)}(t_{i+1}) = a_i u^{(n+1)}(t_{i+1}) + b_i. \] (14)

The solutions of this problem at certain grid points for different values of \( \nu \) are given in tables 1 and 2. The criterion for the convergence was taken as

\[ \text{Max} \left| u^{(n+1)}(t_{i+1}) - u^{(n)}(t_{i+1}) \right| \leq 10^{-4}. \]

### Table 1. Numerical results for example 1 with \( \nu = 0.02 \) and \( \Delta = h = 0.05 \).

<table>
<thead>
<tr>
<th>Mesh point ((x,t))</th>
<th>Values of ( u )</th>
<th>No. of iterations by Picard method</th>
<th>No. of iterations by Newton method</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.05, 0.05)</td>
<td>0.134080</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>(0.45, 0.05)</td>
<td>0.935674</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.95, 0.05)</td>
<td>0.181590</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.15, 0.10)</td>
<td>0.308860</td>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>(0.55, 0.10)</td>
<td>0.913854</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.90, 0.10)</td>
<td>0.469373</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.15, 0.15)</td>
<td>0.233828</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>(0.60, 0.15)</td>
<td>0.825115</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.40, 0.20)</td>
<td>0.462650</td>
<td>43</td>
<td>4</td>
</tr>
<tr>
<td>(0.65, 0.20)</td>
<td>0.716397</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.25, 0.25)</td>
<td>0.225662</td>
<td>40</td>
<td>3</td>
</tr>
<tr>
<td>(0.35, 0.30)</td>
<td>0.247178</td>
<td>28</td>
<td>3</td>
</tr>
<tr>
<td>(0.50, 0.50)</td>
<td>0.158143</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>(0.65, 0.65)</td>
<td>0.112355</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>(0.65, 0.75)</td>
<td>0.075779</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>(0.45, 0.90)</td>
<td>0.037532</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>(0.70, 1.00)</td>
<td>0.024317</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>(0.60, 1.50)</td>
<td>0.001980</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>(0.50, 2.00)</td>
<td>0.000061</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Table 2. Numerical results for example 1 with $v = 0.005$ and $\Delta = h = 0.05$.

<table>
<thead>
<tr>
<th>Mesh point $(x, t)$</th>
<th>Values of $u$ by Picard method</th>
<th>No. of iterations</th>
<th>Values of $u$ by Newton method</th>
<th>No. of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(0.05, 0.05)$</td>
<td>0.134900</td>
<td></td>
<td>0.134900</td>
<td></td>
</tr>
<tr>
<td>$(0.50, 0.05)$</td>
<td>0.975170</td>
<td>11</td>
<td>0.975170</td>
<td>4</td>
</tr>
<tr>
<td>$(0.90, 0.05)$</td>
<td>0.358865</td>
<td></td>
<td>0.358865</td>
<td></td>
</tr>
<tr>
<td>$(0.10, 0.10)$</td>
<td>0.210120</td>
<td></td>
<td>0.210120</td>
<td></td>
</tr>
<tr>
<td>$(0.55, 0.10)$</td>
<td>0.930457</td>
<td>57</td>
<td>0.930457</td>
<td>4</td>
</tr>
<tr>
<td>$(0.85, 0.10)$</td>
<td>0.673192</td>
<td></td>
<td>0.673192</td>
<td></td>
</tr>
<tr>
<td>$(0.15, 0.15)$</td>
<td>*</td>
<td></td>
<td>0.237404</td>
<td></td>
</tr>
<tr>
<td>$(0.45, 0.15)$</td>
<td>*</td>
<td></td>
<td>0.675632</td>
<td></td>
</tr>
<tr>
<td>$(0.85, 0.15)$</td>
<td>*</td>
<td></td>
<td>0.866414</td>
<td></td>
</tr>
<tr>
<td>$(0.30, 0.30)$</td>
<td>*</td>
<td></td>
<td>0.214687</td>
<td></td>
</tr>
<tr>
<td>$(0.70, 0.30)$</td>
<td>*</td>
<td></td>
<td>0.493807</td>
<td>3</td>
</tr>
<tr>
<td>$(0.50, 0.50)$</td>
<td>*</td>
<td></td>
<td>0.160563</td>
<td>3</td>
</tr>
<tr>
<td>$(0.75, 0.75)$</td>
<td>*</td>
<td></td>
<td>0.115583</td>
<td>3</td>
</tr>
<tr>
<td>$(0.75, 0.95)$</td>
<td>*</td>
<td></td>
<td>0.079980</td>
<td>3</td>
</tr>
<tr>
<td>$(0.60, 1.50)$</td>
<td>*</td>
<td></td>
<td>0.021115</td>
<td>3</td>
</tr>
<tr>
<td>$(0.70, 1.75)$</td>
<td>*</td>
<td></td>
<td>0.012273</td>
<td>3</td>
</tr>
<tr>
<td>$(0.50, 1.95)$</td>
<td>*</td>
<td></td>
<td>0.008240</td>
<td>2</td>
</tr>
</tbody>
</table>

* Solution not converging

The number of iterations required for the convergence of solutions in both cases are also given in the tables. It may be noted that for $v = 1/200$ the scheme due to Picard linearization did not converge while the scheme using the Newton linearization yielded a solution in a few iterations.

Example 2:

$$u_t - u_{xx} = (1 + u^2) (1 - 2u).$$

We consider it over two different triangles:

$$0 \leq t \leq 1 - x; \quad 0 \leq x \leq 1,$$

$$0 \leq t \leq 1.5 - x; \quad 0 \leq x \leq 1.5.$$

In each case, the boundary conditions are so chosen as to give the unique exact solution $u = \tan (x + t)$. This problem has earlier been studied by Bellman et al [2] by using iterative methods. We solve this problem by the proposed method.

The solutions at certain grid point for both the triangles are given in table 3. The convergence criterion was taken to be the same as in the previous example. In the case of smaller triangle, it is found that the execution time is the same for both Picard and Newton linearisations for the same accuracy. However, the situa-
### Table 3. Numerical results for example 2 with \( \Delta = h = 0.05 \).

<table>
<thead>
<tr>
<th>Mesh point ((x, t))</th>
<th>Smaller triangle</th>
<th>Bigger triangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>((0.05, 0.05))</td>
<td>0.100367</td>
<td>(0.10, 0.05)</td>
</tr>
<tr>
<td>((0.45, 0.05))</td>
<td>0.546670</td>
<td>(0.45, 0.05)</td>
</tr>
<tr>
<td>((0.90, 0.05))</td>
<td>1.398947</td>
<td>(0.95, 0.05)</td>
</tr>
<tr>
<td>((0.10, 0.10))</td>
<td>0.221662</td>
<td>(0.10, 0.10)</td>
</tr>
<tr>
<td>((0.50, 0.10))</td>
<td>0.733277</td>
<td>(0.70, 0.10)</td>
</tr>
<tr>
<td>((0.85, 0.10))</td>
<td>1.290839</td>
<td>(1.15, 0.10)</td>
</tr>
<tr>
<td>((0.30, 0.15))</td>
<td>0.550513</td>
<td>(0.80, 0.15)</td>
</tr>
<tr>
<td>((0.65, 0.15))</td>
<td>1.092695</td>
<td>(1.20, 0.15)</td>
</tr>
<tr>
<td>((0.80, 0.15))</td>
<td>1.420196</td>
<td>(1.25, 0.15)</td>
</tr>
<tr>
<td>((0.25, 0.20))</td>
<td>0.477993</td>
<td>(0.50, 0.20)</td>
</tr>
<tr>
<td>((0.50, 0.20))</td>
<td>0.917943</td>
<td>(1.00, 0.20)</td>
</tr>
<tr>
<td>((0.75, 0.20))</td>
<td>1.421981</td>
<td>(0.55, 0.30)</td>
</tr>
<tr>
<td>((0.70, 0.25))</td>
<td>1.422444</td>
<td>(1.15, 0.30)</td>
</tr>
<tr>
<td>((0.40, 0.35))</td>
<td>0.997284</td>
<td>(0.50, 0.50)</td>
</tr>
<tr>
<td>((0.55, 0.35))</td>
<td>1.300877</td>
<td>(0.70, 0.75)</td>
</tr>
<tr>
<td>((0.35, 0.50))</td>
<td>1.181929</td>
<td>(0.40, 1.0)</td>
</tr>
<tr>
<td>((0.25, 0.70))</td>
<td>1.412587</td>
<td>(0.20, 1.25)</td>
</tr>
<tr>
<td>((0.15, 0.80))</td>
<td>1.408046</td>
<td>(0.05, 1.35)</td>
</tr>
<tr>
<td>((0.05, 0.90))</td>
<td>1.402138</td>
<td>(0.10, 1.5)</td>
</tr>
</tbody>
</table>

Notation is entirely different in the case of the bigger triangle, where only Newton linearisation is able to provide a solution. These observations are in agreement with those given by Bellman et al [2].

### 4. Discussion and conclusions

A study of the numerical results shows that the Newton method is more effective than the Picard method for obtaining the numerical solution of a quasilinear parabolic problem. The numerical results for the two examples considered have been given in tables 1–3.

Table 1 shows the results for the example 1 with \( v = 1/50 \) for both Picard and Newton linearisations. In this case, at the initial time levels the number of iterations taken by the Picard method are quite large as compared to the Newton method. However, this is not the case at the later time levels. This is due to the fact that...
after a certain stage in time, the solutions approach towards steady states and the gradients become less steep as the time level goes on increasing. It is due to this reason that both Picard and Newton methods take approximately the same number of iterations in the vicinity of the steady state solution. For this example, the steady state solution can be considered to be attained at $t = 2.00$. Table 2 gives the results for the same example, but with $v = 1/200$. In this case, Picard method provides solutions only up to the time level $t = 0.10$. After this, the solution given by Picard method does not stabilise. On the contrary, the Newton method gives the solutions at all time levels by using a few iterations only. In this case, as $v$ decreases further, the convective term $uv$ plays an important role.

The results for the second example are given in Table 3. It is found that the execution time is the same for both the methods, although Picard method takes almost double the number of iterations compared to the Newton method; it may be due to the reason that the expressions to be computed by the Picard method are simpler than for the Newton case. The Newton method appears to have no advantage over the Picard method as there are no steep gradients in this case. However, the situation is entirely different for the same example considered in the bigger triangle. In this case, Picard method fails to yield a solution. This may be due to the reason that steep gradients occur in this triangle in which the boundary condition on $t = 1.5 - x$ is $u = \tan 1.5$ which is quite large. Based on these numerical results, one may conclude that the Newton method is more effective than the Picard method especially when the given nonlinear problem has steep gradients.

References