SPECTRAL METHOD FOR SEMILINEAR PARABOLIC INTEGRODIFFERENTIAL EQUATIONS

Liu Xiao-qing (刘小清) Wu Sheng-chang (吴声昌)
(Institute of Applied Mathematics, Academia Sinica, Beijing)
(Received Feb. 7, 1994; Communicated by Ding Xie-ping)

Abstract

Based on the discussion of the semidiscretization of a parabolic equation with a semilinear memory term, an error estimate is derived for the fully discrete scheme with spectral method in space and the backward Euler method in time. The trapezoidal rule is adopted for the quadrature of the memory term and the quadrature error is estimated.

Key words parabolic, integrodifferential equation, spectral method, backward Euler method

I. Problem and Algorithm

The semilinear parabolic integrodifferential equation

\[ \frac{\partial u(x, t)}{\partial t} + \sigma u(x, t) \frac{\partial^2 u}{\partial x^2} = \int_0^t f(t, s, u(x, s)) ds \quad (1.1) \]

is the mathematical model for heat conduction in material with memory. It also arises in many other applications in the fields of physics, dynamics, and biology, such as the compression of poro-viscoelastic media, reactor dynamics, epidemic phenomena etc. Several cases of equations in the form of (1.1) have been examined as to the existence, uniqueness, and asymptotic behavior of their solutions.\[1\]

For convenience of the application of Fourier approximation, we assume that \( u_0 \) and \( f(t, s, \cdot, u) \) are of period \( 2\pi \), and consider the following weak solution form of (1.1):

\[ \left\{ \begin{array}{l}
(\frac{\partial u}{\partial t}, v) + A(u, v) = \int_0^t (f(t, s, u), v) ds, \\
u(x, 0) = u_0(x),
\end{array} \right. \quad (1.2a)
\]

where \( H^1(0, 2\pi) \) is the subspace of \( H^1(0, 2\pi) \) consisting of all functions with periodic derivatives of every order on the boundary, and \( A(\cdot, \cdot) \) is the bilinear form on \( H^1(0, 2\pi) \times H^1(0, 2\pi) \) defined by

\[ A(u, v) = (d^2u/dx^2, v) \]

Let

\[ F_N = \{ e^{ikx}, x \in (0, 2\pi) : |k| \leq N \} \]

and denote by \( P_N \) the orthogonal projection from \( L^2(0, 2\pi) \) onto \( F_N \). We pose the
semidiscrete scheme as: to find \( u_N \in F_N \) such that

\[
\begin{aligned}
\left\{ \begin{array}{l}
\left( \frac{\partial u_N}{\partial t}, \, v \right) + A(u_N, v) &= \int_0^t \left( f(t, \, s, \, u_N), \, v \right) ds \\
\quad &= \left( \tilde{f}(t, \, u_N), \, v \right), \quad \forall v \in F_N \\
u_N(0) &= P_N u_0
\end{array} \right.
\end{aligned}
\]  

(1.3a)

If we introduce the discrete version \( A_N : F_N \to F_N \) of the operator \( d^2/dx^2 \) defined by

\[
(A_N u, \, v) = A(u, \, v), \quad \forall u, \, v \in F_N
\]

(1.4)

then (1.3a) can be rewritten as

\[
\frac{\partial u_N}{\partial t} + A_N u_N = P_N \tilde{f}(t, \, u_N)
\]

Next, we use the backward Euler method in time to construct a fully discrete scheme as: to find \( u_N^* \in F_N \) such that

\[
\begin{aligned}
\left\{ \begin{array}{l}
\partial_{\tau} u_N^* + A_N u_N^* &= P_N \tilde{f}^*(u_N) \\
u_N^0 &= P_N u_0
\end{array} \right.
\end{aligned}
\]

(1.5a)

where \( \tau \) is the time step, and \( \partial_{\tau} \) denote the backward difference, i.e.

\[
\partial_{\tau} u_N^* = \frac{1}{\tau} (u_N^* - u_N^{*-1})
\]

\( \tilde{f}^*(u_N) \) is an approximation \( \tilde{f}(t_N, \, u_N) \) based on some quadrature rule. The simplest quadrature rule is the rectangular rule, which requires storage of \( u_N^i, \, i=0, \cdots, n-1 \), in order to compute \( u_N^N \).

To avoid this disadvantage in the implementation, we consider employing the trapezoidal rule instead, so that the truncation error is of second-order. Therefore, a larger quadrature stepsize can be adopted, and fewer quadrature points \( O(\tau^{-1/2}) \) need to be stored.

Let \( h = k \tau \) be the quadrature stepsize, where \( k = [\tau^{-1/2}] \), and \( T_n \) be the unique nonnegative integer satisfying \( I_n h \leq t_{n-1} \), and \( (I_n + 1) h > t_{n-1} \), and let \( S_i = i h, \, i=0, \cdots, n \). We give a partition of the interval of integration as follows:

\[
[0, \, t_n] = \bigcup_{i=0}^{I_n-1} [S_i, \, S_{i+1}] \cup [S_{I_n}, \, t_{n+1}] \cup [t_{n+1}, \, t_n]
\]

To ensure that (1.5) leads to a system of linear algebraic equations, we use the rectangular rule on the subinterval \([t_{n-1}, \, t_n]\) with \( t_{n-1} \) as the node. On the other subintervals, we use the trapezoidal rule. Thus the quadrature formula is

\[
\tilde{f}^*(u_N) = \frac{h}{2} \sum_{i=0}^{I_n-1} \left[ f(t_n, \, S_i, \, u_N^i) + f(t_n, \, S_{i+1}, \, u_N^{i+1}) \right] + \frac{1}{2} (t_{n-1} - S_I) \left[ f(t_n, \, S_I, \, u_N^{I-1}) \right]
\]

\[
\quad + \tau f(t_n, \, t_{n-1}, \, u_N^{n-1})
\]