Journal of Computational Mathematics, Vol.12, No.3, 1994, 259–264.

TIME DISCRETIZATION SCHEMES FOR AN INTEGRO-DIFFERENTIAL EQUATION OF PARABOLIC TYPE*1)

Huang Yun-qing
(*Department of Mathematics, Xiangtan University, Xiangtan, Hunan, China*)

Abstract

In this paper a new approach for time discretization of an integro-differential equation of parabolic type is proposed. The methods are based on the backward-Euler and Crank-Nicolson Schemes but the memory and computational requirements are greatly reduced without assuming more regularities on the solution u.

1. Introduction

We consider the time discretization of the equation

$$\begin{cases} u_t + Au = \int_0^T b(t,s)Bu(s)ds + f(t), \quad 0 < t < T, \\ u(0) = v, \end{cases} \tag{1.1}$$

where A is an unbounded positive definite self-adjoint operator with dense domain $D(A)$ in a Hilbert space H and B is another operator with domain $D(B) \supset D(A)$. The kernel $b(t,s)$ is assumed to be a smooth real-valued function of both t and s for $0 \le s \le t$ and $f(t) \in H$ is a smooth function.

This type of problem occurs in applications such as heat conduction in material with memory, compression of poro-viscoelastic media, nuclear reactor dynamics, etc. The numerical solution by means of spatial discretization by finite differences and finite element methods has been studied by several authors; see V. Thomee [2] and the references cited there.

In this paper, we shall restrict our attention to the time discretization of such problems. A standard way of time discretization is to employ the quadrature formula

$$\int_0^{t_n} g(s)ds \approx \sum_{j=0}^{n-1} \omega_{nj} g(jk), \tag{1.2}$$

* Received July 7, 1993.
1) The Project Supported by National Natural Science Foundation of China.

where k denotes the time step, e.g., the left rectangle rule and the trapezoidal rule are simple quadrature rules which are consist with $O(k)$ accuracy of the backward-Euler scheme and with $O(k^2)$ accuracy of the Crank-Nicolson scheme , respectively.

Let $t_n = nk$ and U^n be the approximation of $u(t_n)$ and $f^n = f(t_n)$. Also we define the backward difference operator by

$$\bar{\partial}U^n = \frac{U^n - U^{n-1}}{k}. \tag{1.3}$$

Let $\sigma_1^n(g) = k\sum_{j=0}^{n-1} g(t_j)$ and $\sigma_2^n(g) = \frac{1}{2}kg(0) + \sum_{j=0}^{n-1} g(t_j)$ be the left rectangle rule and the trapezoidal rule respectively. Then, the standard backward Euler and Crank-Nicolson schemes are

$$BE: \begin{cases} \bar{\partial}U^n + AU^n = \sigma_1^n(b(t_n, s)BU) + f^n, & n = 1, 2, \cdots, \\ U^0 = V; \end{cases} \tag{1.4}$$

$$CN: \begin{cases} \bar{\partial}U^n + A\Big(\dfrac{U^n + U^{n-1}}{2}\Big) = \sigma_2^n(b(t_{n-\frac{1}{2}}, s)BU) + f^{n-\frac{1}{2}}, & n = 1, 2, \cdots, \\ U^0 = V, \end{cases} \tag{1.5}$$

where $\sigma_1^n(b(t_n, s)BU) = k\sum_{j=0}^{n-1} b(t_n, t_j)BU^j$ and σ_2^n is similar.

A practical difficulty of these methods is that all U^j need to be stored as they all enter the subsequent equations; hence the number of U^j which have to be stored is of order $O(\frac{1}{k})$ per unit time.

In order to reduce the memory requirement, Sloan and Thomee[1] proposed more economical schemes by using quadrature rules with higher order truncation errors. For example, in order to retain the accuracy of the backward Euler scheme, they used the trapezoidal rule with mesh size $k_1 = O(\sqrt{k})$ on $[0, \bar{t_{jn}}]$ and the rectangle rule with mesh size k on the remaining small part $[\bar{t_{jn}}, t_n]$, where $\bar{t_{jn}} = \max\{jk_1\}$ $(jk_1 \leq t_{n-1})$. For this scheme, the storage requirements are reduced from $O(\frac{1}{k})$ to $O(\frac{1}{\sqrt{k}})$ per unit time. Likewise, a combination of Simpson's rule and the trapezoidal rule preserves the accuracy of the Crank-Nicolson scheme. Because of using higher order quadratures, the regularity requirement of the solution u is very severe.

The results here are based on the following iterative relations for the quadrature:

$$\sigma_1^n(g) = k\sum_{j=0}^{n-1} g(t_j) = \sigma_1^{n-1}(g) + kg(t_{n-1}) \left(\approx \int_0^{t_n} g(s)ds\right), \tag{1.6}$$

$$\begin{cases} \sigma_2^n(g) = \dfrac{1}{2}kg(0) + k\displaystyle\sum_{j=0}^{n-1} g(t_j) = \sigma_2^{n-1}(g) + kg(t_{n-1}) \\ \qquad \left(\approx \displaystyle\int_0^{t_{n-\frac{1}{2}}} g(s)ds\right), \\ \sigma_2^0(g) = -\dfrac{1}{2}kg(0). \end{cases} \tag{1.7}$$