INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 4, Number 1, Pages 63–73

SOLUTION OF A SINGULAR INTEGRAL EQUATION BY A SPLIT-INTERVAL METHOD

TERESA DIOGO, NEVILLE J. FORD, PEDRO M. LIMA, AND SOPHY M. THOMAS

Abstract. In this paper we give details of a new numerical method for the solution of a singular integral equation of Volterra type that has an infinite class of solutions. The split-interval method we have adopted utilises a simple robust numerical method over an initial time interval (which includes the singularity) combined with extrapolation. We describe the method and give details of its order of convergence together with examples that show its effectiveness.

Key Words. Numerical methods, extrapolation, singular integral equation, Volterra equation.

1. Introduction

This paper is concerned with the development of efficient numerical methods for the solution of Volterra integral equations which have a singularity in the kernel that leads to there being more than one solution. Therefore we are not here concerned with equations of Abel type where, despite a singularity in the kernel there is nevertheless a unique solution. More details of numerical methods for Abel type equations may be found, for example, in [1], [2]. As prototype for the type of problems we are interested in we shall investigate the equation

$$u(t) = g(t) + \int_0^t \frac{s^{\mu-1}}{t^{\mu}} u(s)\, ds \tag{1}$$

which has just the kind of properties that present real challenges to numerical analysts.

The precise analysis of equation (1) depends critically on the value of μ. For the case when $\mu < 1$, it has been shown in [8] that there is an infinite set of solutions, one of which is continuously differentiable whenever g is continuously differentiable, and all the other solutions have infinite gradient at the origin. This infinity of solutions arises because for $\mu < 1$ the singularity in the integrand persists at $s = 0$ for all $t > 0$. Of course when $\mu > 1$ there is still a singularity in the integrand for $t = 0$ but this does not persist when $t > 0$ and it turns out that the equation then has a unique solution.

Thus we shall concentrate on the case $0 < \mu < 1$. We are aided in our work by the fact that an explicit formula for the set of solutions has been derived. The solution formula is

$$u(t) = ct^{1-\mu} + g(t) + \frac{g(0)}{\mu - 1} + t^{1-\mu} \int_0^t s^{\mu-2}(g(t) - g(0))\, ds \tag{2}$$

Received by the editors January 25, 2005 and revised January 10, 2006.
2000 *Mathematics Subject Classification.* 65R20.
Teresa Diogo and Pedro Lima are pleased to acknowledge the support from FCT, project POCTI/MAT/45700/2002.

where $g \in C^1$, and c is an arbitrary constant , which determines the particular trajectory to be followed. When $c = 0$, we have the smooth solution (see [8]).

All this makes our choice of prototype equation ideal. The underlying equation has the complicated properties we wish to investigate in our numerical schemes but we have the luxury of an exact solution with which we can make comparisons of our numerical approximations.

1.1. Previous numerical approaches. The derivation and theoretical background of equation (1) are given in [9] and the references cited therein. In this context, the first use of product integration formulae (see [12]) was applied to the related equation

$$y(t) + \int_0^t p(t,s)y(s)ds = f(t) \tag{3}$$

$$p(t,s) := \frac{1}{\sqrt{\pi}} \frac{1}{\sqrt{\log(t/s)}} \left(\frac{s}{t}\right)^{\mu} \frac{1}{s}\, ds.$$

The product Euler and product trapezoidal methods were applied to equation (3) for the case when $\mu > 1$, with the restriction on the forcing function that $f(t) \in C^m$, where $m = 1$ for the product Euler and $m = 2$ for the product trapezoidal scheme, with convergence orders 1 and 2 respectively.

Equation (3) may be transformed into (1), where the forcing functions are related by the formula

$$g(t) = -\int_0^t p(t,s)f(s)\, ds + f(t).$$

This process is used in [7], and Hermite-type collocation is applied to the simplified equation, again for $\mu > 1$, and the convergence is of order 4, but subject to the more stringent restriction that $g \in C^5$.

Extrapolation procedures were introduced in conjunction with the product Euler scheme in [10] for the solution of (1), still with $\mu > 1$, but now allowing the g to have a singularity at the origin. Richardson's extrapolation is applied to the results, but we find that where the error expansion contains non-integer powers of h, the extrapolation is not always effective.

The case where $0 < \mu \leq 1$, and the implication of multiple solutions, is first considered in [9]. The product Euler method is now shown to converge to the single smooth solution, where the input function $g(t) \in C^1[0,t]$. If $\mu = 1$, there is the additional constraint that $g(0) = 0$. For $g \in C^2[0,T]$ and $\mu < 1$, the error expansion is now shown to be of the form

$$e_k^h = Ah^{\mu} + O(h).$$

If g is not sufficiently smooth, the techniques of [10] may be employed, and if $g(t) = t^{\tilde{\gamma}}\overline{g}(t)$, $\overline{g} \in C^2$, $\tilde{\gamma} \in R$, such that $\mu + \tilde{\gamma} < 1$, the order will depend on $\mu + \tilde{\gamma}$, not μ alone. We further note that if $\mu = 1$ or $\mu = 2$, there is a logarithmic term in the error expansion. The numerical results reflect this weak order of convergence, so extrapolation is again used to improve the accuracy. The scheme used in the paper is the E-algorithm of Brezinski (see [3] or [9]), which is a very general scheme, applicable to any situation for which the error expansion is known.

Up to this point, attention has been focused on approximating the smooth solution, and numerical methods for its trajectory which commence at the origin. In [4], approximation of the non-smooth solutions is considered. A specific member of the non-smooth family may be uniquely identified at a point $t = \alpha$ ($\alpha = kh, k \in Z$),