

A New Integration Algorithm for Ordinary Differential Equations Based on Continued Fraction Approximations

I.M. Willers
CERN

A new integration algorithm is found, and an implementation is compared with other programmed algorithms. The new algorithm is a step-by-step procedure for solving the initial value problem in ordinary differential equations. It is designed to approximate poles of small integer order in the solutions of the differential equations by continued fractions obtained by manipulating the sums of truncated Taylor series expansions.

The new method is compared with the Gragg-Bulirsch-Stoer, and the Taylor series method. The Taylor series method and the new method are shown to be superior in speed and accuracy, while the new method is shown to be most superior when the solution is required near a singularity. The new method can finally be seen to pass automatically through singularities where all the other methods which are discussed will have failed.

Key Words and Phrases: ordinary differential equations, initial value problem, integration, Taylor series, singularities, continued fractions, program comparison.

CR Categories: 5.17

Copyright © 1974, Association for Computing Machinery, Inc. General permission to republish, but not for profit, all or part of this material is granted provided that ACM's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Association for Computing Machinery.

Author's address: CERN, Data Handling Division, 1211 Geneva 23, Switzerland.

1. Introduction

The Taylor series method has proved to be a very efficient procedure for solving the initial value problem in ordinary differential equations when the solutions are well approximated by polynomials in the independent variable (see Barton, Willers, and Zahar [1], and Willers [2]). In this paper a new algorithm is described that enables one to obtain a similar improvement over another class of problems, while still maintaining the superiority of the Taylor series method for the former class. The most commonly used algorithms which numerically solve ordinary differential equations are based on a polynomial or truncated Taylor series representation of the solution. The most common exception to this is the algorithm due to Bulirsch and Stoer [3], which was based on work done by Gragg [4]. It is well known that some functions are not well represented by polynomials and in any case a Taylor series is only valid within a circle whose radius is limited by the nearest singularity.

In this paper it is shown how a new step-by-step integration algorithm based on continued fraction expansions is able to overcome the problem of poles of small integer order. The algorithm arrives at continued fraction approximations by manipulating the sums of truncated Taylor series expansions. Since the algorithm is able to approximate poles of small integer order, the region of validity of the approximation is extended beyond such poles. Also singularities which are not poles of integer order may be more accurately approximated by continued fractions, so an appropriate gain is also possible here. It should be realized that it is possible to pass beyond integer order poles on the real axis via the complex plane, but it is mathematically impossible to do so uniquely for singularities which are not poles of integer order.

2. A Preliminary Comparison of Four Methods

Consider the differential equation $y' = -y^2$, with $y(1) = 1$. It has the solution $y = 1/t$, where t is the independent variable and will be integrated from $t = 1$ toward the value $t = 0$. The following routines were compared for this problem: Gragg-Bulirsch-Stoer (B.S.), the Taylor series method (T.S.), Adams predictor (A.P.) and Runge-Kutta-Gill (R.K.). Each routine had set an initial step length of 2^{-5} and an accuracy per step of 10^{-10} . The programs were stopped, either when the program gave up or when it had exhausted 30 sec of

Table I. Step Lengths Taken by Four Methods

t	Step for T.S.	Step for R.K.	Step for A.P.	Step for B.S.
10^{-1}	1.4×10^{-2}	4.8×10^{-4}	2.5×10^{-3}	5.8×10^{-2}
10^{-2}	9.2×10^{-4}		2.6×10^{-4}	9.0×10^{-3}
10^{-3}	5.2×10^{-6}		2.7×10^{-5}	6.0×10^{-4}
10^{-4}			2.8×10^{-6}	8.0×10^{-5}
10^{-5}			2.4×10^{-7}	
10^{-6}			2.5×10^{-8}	

computing time. Table I was then produced showing the modulus of the step lengths taken by each method for different values of t .

Each algorithm showed remarkably different behavior from the other three. The most efficient algorithm was Gragg-Bulirsch-Stoer; this result is quite expected if the algorithm is studied. The Taylor series method performed quite well until $t = 10^{-2}$, after which it became excessively slow. The Adams predictor method continued to produce good results with a step length which was a constant fraction of the theoretical limit, which was equal to the value of t . The Adams predictor method was able to progress to 10^{-6} in a reasonable amount of computing time. Runge-Kutta-Gill could make very little progress.

It was examples of this nature which motivated the author to find an integration algorithm which has good behavior near singularities. The new algorithm takes large step lengths when integrating the problem under consideration and does in fact integrate past the singularity (e.g. the step taken at $t = 10^{-1}$ was -6.7×10^{-1}).

3. The Algorithm

The algorithm is designed to take advantage of the Taylor series principle used in solving ordinary differential equations. Let it be assumed therefore that there exists a mechanism for calculating the coefficients of the Taylor series of functions that are the solutions of ordinary differential equations, at any point t , which may be defined by the new algorithm. Barton, Willers, and Zahar [5] describe how an efficient recurrence algorithm may be automatically generated from a set of ordinary differential equations and how it may be used to calculate the Taylor series for the dependent variables.

The new part of the algorithm is that, when given a truncated Taylor series, that series is manipulated as if it were the expansion of a rational function. If it is discovered that the original Taylor series is a better approximation to the function than a rational approximation, then it is to be expected that the work so done is an overhead on the ordinary Taylor series method.

However, equally accurate results will be obtained. When the rational approximation is superior, there will be an increase in accuracy and one may expect a corresponding reduction in computing time. In order to remain competitive with the Taylor series over the complete range of problems, an attempt was made to keep the overhead as small as possible.

Given the truncated Taylor series $A_0^\sigma = \sum_{\nu=0}^{\sigma} \eta^{\nu-1} t^\nu$, the series are manipulated using the quantities, A_μ^σ , C_μ^σ , and D_μ^σ , which are defined below. Equations (1) are used to define the quantities A_0^σ , C_0^σ and D_0^σ , and eqs. (4) are used to evaluate the A_μ^σ , C_μ^σ , and D_μ^σ , for $\mu > 0$. The A_μ^σ are a combination of polynomial and rational approximations to the value of the function at the point t . The following defines the A_μ^σ , C_μ^σ , and D_μ^σ , and indicates how the complete algorithm given by eqs. (1) and (4) may be deduced.

THEOREM. If A_0^σ , C_0^σ and D_0^σ are given by

$$C_0^\sigma = 0, \quad D_0^\sigma = 1, \quad A_0^\sigma = \sum_{\nu=0}^{\sigma} \eta_0^{\nu-1} t^\nu, \quad (1)$$

and A_μ^σ , C_μ^σ , and D_μ^σ are given by

$$A_\mu^\sigma = \sum_{\nu=0}^{\sigma} \eta_0^{\nu-1} t^\nu + t^{\sigma+1} \sum_{\lambda=0}^{2\mu-1} \eta_\lambda^\sigma, \quad \mu > 0,$$

$$C_\mu^\sigma = \frac{1}{t^{\sigma+1}} \left[\prod_{\nu=0}^{\mu-1} \eta_{2(\mu-\nu-1)}^{\sigma+\nu} t^{\sigma+\nu+1} \right] \sum_{\lambda=0}^{2(\mu+1)} \frac{1}{\eta_\lambda^\sigma}, \quad \mu > 0, \quad (2)$$

$$D_\mu^\sigma = \prod_{\nu=0}^{\mu-1} \eta_{2(\mu-\nu-1)}^{\sigma+\nu} t^{\sigma+\nu+1}, \quad \mu > 0. \quad (3)$$

where $\eta_0^{\nu-1}$ are coefficients of a truncated Taylor series expansion and η_λ^σ are functions of t where $\sum_{\lambda=0}^{2\mu-1} \eta_\lambda^\sigma$ is the Euler expansion and the terms of the expansion are defined by the recurrences (see Bauer [6])

$$t \left[\frac{1}{\eta_{2\mu}^\sigma} + \frac{1}{\eta_{2\mu+1}^\sigma} \right] = \frac{1}{\eta_{2\mu-1}^{\sigma+1}} + \frac{1}{\eta_{2\mu}^{\sigma+1}}, \quad (4)$$

$$\frac{1}{t} [\eta_{2\mu-1}^\sigma + \eta_{2\mu}^\sigma] = \eta_{2\mu-2}^{\sigma+1} + \eta_{2\mu-1}^{\sigma+1}, \quad \text{and} \quad (5)$$

$$1/\eta_{-1}^\sigma \equiv 0.$$

Then it can be shown that, for the quantities A_μ^σ , C_μ^σ , D_μ^σ , the following relationships exist

$$\begin{aligned} C_\mu^\sigma &= C_{\mu-1}^{\sigma+1} (A_{\mu-1}^{\sigma+1} - A_{\mu-1}^\sigma) + D_{\mu-1}^{\sigma+1}, \\ D_\mu^\sigma &= D_{\mu-1}^{\sigma+1} (A_{\mu-1}^{\sigma+1} - A_{\mu-1}^\sigma), \quad \mu = 1, 2, 3, \dots, \\ A_\mu^\sigma &= A_{\mu-1}^{\sigma+1} + D_\mu^\sigma D_{\mu-1}^{\sigma+1} (D_{\mu-1}^\sigma C_{\mu-1}^{\sigma+1} - D_{\mu-1}^{\sigma+1} C_{\mu-1}^\sigma)^{-1}. \end{aligned} \quad (6)$$

Note: the Euler expansion is in agreement with a continued fraction expansion up to the point of truncation. Therefore, the A_μ^σ represent an approximation which consists of a combination of a truncated Taylor series expansion and a truncated continued fraction represented by the Euler expansion. In the case of $\mu = 0$, A_0^σ represents the traditional truncated Taylor series expansion.

It may also be considered as a generalization of the

Taylor series method which permits consideration of the other elements in the lower half of the Padé table, in addition to the partial sums of the Taylor series itself.

The flow of the paper is not interrupted if the reader avoids the following proof.

PROOF.

$$\begin{aligned}
A_{\mu}^{\sigma+1} - A_{\mu}^{\sigma} &= \sum_{\nu=0}^{\sigma+1} \eta_{\sigma}^{\nu-1} t^{\nu} - \sum_{\nu=0}^{\sigma} \eta_{\sigma}^{\nu-1} t^{\nu} \\
&+ t^{\sigma+2} \sum_{\lambda=0}^{2\mu-1} \eta_{\lambda}^{\sigma+1} - t^{\sigma+1} \sum_{\lambda=0}^{2\mu-1} \eta_{\lambda}^{\sigma} \\
&= \eta_0^{\sigma} t^{\sigma+1} + t^{\sigma+2} \sum_{\lambda=0}^{2\mu-1} \left[\eta_{\lambda}^{\sigma+1} - \frac{1}{t} \eta_{\lambda}^{\sigma} \right] \\
&= \eta_0^{\sigma} t^{\sigma+1} + t^{\sigma+2} \sum_{\lambda=0}^{\mu-1} \\
&\quad \cdot \left[\eta_{2\lambda}^{\sigma+1} + \eta_{2\lambda+1}^{\sigma+1} - \frac{1}{t} (\eta_{2\lambda+1}^{\sigma} + \eta_{2\lambda+2}^{\sigma}) \right] \\
&\quad + t^{\sigma+1} [\eta_{2\mu}^{\sigma} - \eta_0^{\sigma}].
\end{aligned}$$

By eqs. (4) and (5) this gives

$$A_{\mu}^{\sigma+1} - A_{\mu}^{\sigma} = \eta_{2\mu}^{\sigma} t^{\sigma+1}. \quad (7)$$

Then from the definition of D_{μ}^{σ} given by eq. (3) $D_1^{\sigma} = \eta_0^{\sigma} t^{\sigma+1}$; by eqs. (3) and (7) $D_1^{\sigma} = D_0^{\sigma} (A_0^{\sigma+1} - A_0^{\sigma})$; and for $\mu > 0$

$$D_{\mu+1}^{\sigma} = \prod_{\nu=0}^{\mu} \eta_{2(\mu-\nu)}^{\sigma+\nu} t^{\sigma+\nu+1},$$

$$D_{\mu+1}^{\sigma} = \left[\prod_{\nu=0}^{\mu-1} \eta_{2(\mu-\nu-1)}^{\sigma+\nu+1} t^{\sigma+\nu+2} \right] [\eta_{2\mu}^{\sigma} t^{\sigma+1}];$$

and by eqs. (3) and (7)

$$D_{\mu+1}^{\sigma} = D_{\mu}^{\sigma+1} (A_{\mu}^{\sigma+1} - A_{\mu}^{\sigma}).$$

From the definition of C_{μ}^{σ} given by eq. (2)

$$C_1^{\sigma} = \frac{1}{t^{\sigma+1}} [\eta_0^{\sigma} t^{\sigma+1}] \frac{1}{\eta_0^{\sigma}}, \quad C_1^{\sigma} = 1, \quad \text{and}$$

$$C_2^{\sigma} = \frac{1}{t^{\sigma+1}} \left[\prod_{\nu=0}^1 \eta_{2(1-\nu)}^{\sigma+\nu} t^{\sigma+\nu+1} \right] \sum_{\nu=0}^2 \frac{1}{\eta_{\lambda}^{\sigma}}$$

$$C_2^{\sigma} = \frac{1}{t^{\sigma+1}} [\eta_2^{\sigma} t^{\sigma+1} \eta_0^{\sigma+1} t^{\sigma+2}] \left[\frac{1}{\eta_0^{\sigma}} + \frac{1}{\eta_1^{\sigma}} + \frac{1}{\eta_2^{\sigma}} \right],$$

and by eq. (4)

$$C_2^{\sigma} = t^{\sigma+2} \eta_0^{\sigma+1} \eta_2^{\sigma} \left[t \left(\frac{1}{\eta_{-1}^{\sigma+1}} + \frac{1}{\eta_0^{\sigma+1}} \right) + \frac{1}{\eta_2^{\sigma}} \right]$$

$$\begin{aligned}
C_2^{\sigma} &= \eta_2^{\sigma} t^{\sigma+3} + \eta_0^{\sigma+1} t^{\sigma+1} \\
&= C_1^{\sigma+1} (A_1^{\sigma+1} - A_1^{\sigma}) + D_1^{\sigma+1}.
\end{aligned}$$

Then for $\mu > 0$ by eq. (2)

$$\begin{aligned}
C_{\mu}^{\sigma} &= \frac{1}{t^{\sigma+1}} \left[\prod_{\nu=0}^{\mu-1} \eta_{2(\mu-\nu-1)}^{\sigma+\nu} t^{\sigma+\nu+1} \right] \sum_{\lambda=0}^{2(\mu-1)} \frac{1}{\eta_{\lambda}^{\sigma}} \\
&= \frac{1}{t^{\sigma+1}} \left[\prod_{\nu=0}^{\mu-2} \eta_{2(\mu-\nu)}^{\sigma+\nu+1} t^{\sigma+\nu+2} \right] [\eta_{2(\mu-1)}^{\sigma} t^{\sigma+1}] \\
&\quad \cdot \left[\sum_{\lambda=0}^{\mu-2} \left(\frac{1}{\eta_{2\lambda}^{\sigma}} + \frac{1}{\eta_{2\lambda+1}^{\sigma}} \right) + \frac{1}{\eta_{2(\mu-1)}^{\sigma}} \right];
\end{aligned}$$

by eq. (4)

$$\begin{aligned}
C_{\mu}^{\sigma} &= \eta_{2(\mu-1)}^{\sigma} \left[\prod_{\nu=0}^{\mu-2} \eta_{2(\mu-\nu)}^{\sigma+\nu+1} t^{\sigma+\nu+2} \right] \\
&\quad \cdot \left[\sum_{\lambda=0}^{\mu-1} t \left(\frac{1}{\eta_{2\lambda-1}^{\sigma+1}} + \frac{1}{\eta_{2\lambda}^{\sigma+1}} \right) + \frac{1}{\eta_{2(\mu-1)}^{\sigma}} \right] \\
&= t \eta_{2(\mu-1)}^{\sigma} \left[\prod_{\nu=0}^{\mu-2} \eta_{2(\mu-\nu)}^{\sigma+\nu+1} t^{\sigma+\nu+2} \right] \\
&\quad \cdot \left[\sum_{\lambda=0}^{2(\mu-1)} \frac{1}{\eta_{\lambda}^{\sigma+1}} - \frac{1}{\eta_{2(\mu-1)}^{\sigma}} \right];
\end{aligned}$$

by eqs. (2), (3), and (7)

$$C_{\mu}^{\sigma} = C_{\mu-1}^{\sigma+1} (A_{\mu-1}^{\sigma+1} - A_{\mu-1}^{\sigma}) + D_{\mu-1}^{\sigma+1}.$$

Now consider the quantity $A_{\mu-1}^{\sigma+1} + D_{\mu}^{\sigma} D_{\mu}^{\sigma+1} (D_{\mu}^{\sigma} C_{\mu}^{\sigma+1} - D_{\mu}^{\sigma+1} C_{\mu}^{\sigma})^{-1}$. Then, for $\mu = 1$, this gives

$$\begin{aligned}
A_0^{\sigma+1} + D_1^{\sigma} D_1^{\sigma+1} (D_1^{\sigma} C_1^{\sigma+1} - D_1^{\sigma+1} C_1^{\sigma})^{-1} \\
&= \sum_{\nu=0}^{\sigma+1} \eta_0^{\nu-1} t^{\nu} + (\eta_0^{\sigma} t^{\sigma+1}) (\eta_0^{\sigma+1} t^{\sigma+1}) [\eta_0^{\sigma-1} t^{\sigma+2} - \eta_0^{\sigma+1} t^{\sigma+2}]^{-1} \\
&= \sum_{\nu=0}^{\sigma+1} \eta_0^{\nu-1} t^{\nu} + t^{\sigma+2} \left[\frac{1}{\eta_0^{\sigma+1}} - t \frac{1}{\eta_0^{\sigma}} \right]^{-1} \\
&= \sum_{\nu=0}^{\sigma+1} \eta_0^{\nu-1} t^{\nu} + t^{\sigma+1} \eta_1^{\sigma},
\end{aligned}$$

and by eq. (4)

$$= \sum_{\nu=0}^{\sigma} \eta_0^{\nu-1} t^{\nu} + t^{\sigma+1} \sum_{\lambda=0}^1 \eta_{\lambda}^{\sigma} = A_1^{\sigma}.$$

In a similar way the following relationship may be established

$$A_{\mu}^{\sigma} = A_{\mu-1}^{\sigma+1} + D_{\mu}^{\sigma} D_{\mu}^{\sigma+1} (D_{\mu}^{\sigma} C_{\mu}^{\sigma+1} - D_{\mu}^{\sigma+1} C_{\mu}^{\sigma})^{-1}. \quad \square$$

Hence it is possible to establish the algorithm. The algorithm is algebraically equivalent to the epsilon algorithm (Shanks [7]). Computationally it has the advantage of only including one division per application unlike the epsilon or η algorithm. This makes control of the algorithm much simpler, and the avoidance of arithmetic overflow due to the division by zero or a small number easier. It is partly for this reason that in practice this algorithm has been proved to be superior to using the epsilon or η algorithm. The algorithm may be enclosed within one programmed loop.

To show the algorithm in operation, consider the series $A_0^{\sigma} = \sum_{\nu=0}^{\sigma} t^{\nu}$. Then

$$C_1^{\sigma} = 1, \quad D_1^{\sigma} = t^{\sigma+1},$$

$$A_1^{\sigma} = \sum_{\nu=0}^{\sigma+1} t^{\nu} + \frac{t^{2\sigma+3}}{t^{\sigma+1} - t^{\sigma+2}} = \sum_{\nu=0}^{\sigma+1} t^{\nu} + \frac{t^{\sigma+2}}{1-t} = \frac{1}{1-t}.$$

It is interesting to notice that while A_0^{σ} was not convergent outside $|t| = 1$ for increasing σ , the approximation A_1^{σ} is valid everywhere.

4. The Measurement of Error

It is now necessary to establish the error committed in truncating the series A_μ^σ .

$$A_\mu^\sigma = \sum_{\nu=0}^{\sigma} \eta_0^{\nu-1} t^\nu + t^{\sigma+1} \sum_{\lambda=0}^{2\mu-1} \eta_\lambda^\sigma, \quad \mu > 0.$$

The exact expression for the function being approximated is given by

$$A_\infty^\sigma = \sum_{\nu=0}^{\sigma} \eta_0^{\nu-1} t^\nu + t^{\sigma+1} \sum_{\lambda=0}^{\infty} \eta_\lambda^\sigma,$$

although the region in which this is valid may be local about $t = 0$ and may depend on the value of σ . However, locally it is possible to discuss the function $E_\mu^\sigma(t)$, which is defined as

$$E_\mu^\sigma(t) = A_\infty^\sigma - A_\mu^\sigma = t^{\sigma+1} \sum_{\lambda=2\mu}^{\infty} \eta_\lambda^\sigma, \quad \mu \neq 0.$$

In the case when $\mu = 0$, A_0^σ is given by

$$A_0^\sigma = \sum_{\nu=0}^{\sigma} \eta_0^{\nu-1} t^\nu,$$

and the exact expression locally is given by

$$A_0^\infty = \sum_{\nu=0}^{\infty} \eta_0^{\nu-1} t^\nu,$$

$$\text{giving } E_0^\sigma(t) = A_0^\infty - A_0^\sigma = \sum_{\nu=\sigma+1}^{\infty} \eta_0^{\nu-1} t^\nu.$$

From the theory for the Taylor series, it is known that there exists a point ξ and a point t for which $E_0^\sigma(t) = \eta_0^\sigma (\xi)(t - \xi)^{\sigma+1}$ in a region $|a, a + \lambda|$. So roughly $E_0^\sigma(t)$ may be estimated by $E_0^\sigma(t) \sim \eta_0^\sigma t^{\sigma+1}$, which is a good approximation if η_0^σ varies only slightly over the region of interest.

For $E_\mu^\sigma(t) = t^{\sigma+1} \sum_{\lambda=2\mu}^{\infty} \eta_\lambda^\sigma$, the sum represents a continued fraction expansion which locally is in agreement with the Taylor series expansion. A similar argument is deduced from this and $E_\mu^\sigma(t)$ is estimated by $E_\mu^\sigma(t) \sim \eta_{2\mu}^\sigma t^{\sigma+1}$, which is now also a valid expression for all $\mu \geq 0$. By eq. (7) this becomes

$$E_\mu^\sigma(t) = A_\mu^{\sigma+1} - A_\mu^\sigma. \quad (8)$$

If the original Taylor series is of length σ , it is possible for the quantities $E_\mu^1, E_\mu^2, \dots, E_\mu^{\sigma-1}$ to be estimated in this manner. The criterion used is to estimate an error per step $E(t)$, which is given by the relation

$$E(t) = \max_i \{ \max (|E_\mu^{\sigma-2}|, |E_\mu^{\sigma-1}|) / (1 + |y_i|), \alpha \},$$

where \max_i represents the maximum over the dependent variables y_i , and α is the machine accuracy relative to 1. The quantity α is introduced to avoid the erroneous setting of $E(t)$ to the value zero.

This measure leads to a very efficient method of estimating an upper bound to the possible error committed per step. It should also be noted that the measure is essentially relative for large $|y_i|$ and essentially absolute for small $|y_i|$. This has in practice been found to be the type of error measure usually required.

5. Step Length Control

The criterion for selecting a step is based on the error $E(t)$ committed by the previous step. Initially it is assumed that a step of $h_{-1} = 1$ has just been taken, and $E(t)$ is calculated on this basis. If ϵ is the local error per step requested, a step is accepted if $E(h_i) \leq \epsilon$ and rejected if $E(h_i) > \epsilon$. In both cases a new h is calculated using the formula $h_{\text{new}} = c_i (\delta \epsilon / E(h_i))^{1/(\sigma+1)}$.

Where $\delta = 0.2$ after a rejected step and 1 otherwise, $c_0 = 1$ and $c_{i+1} = c_i$ for an accepted step, and $(c_i)_{\text{new}} = (\delta \epsilon / E(h_i))^{1/(\sigma+1)} (c_i)_{\text{old}}$ after a rejected step. The error control is similar to the control used for the Taylor series system, and it is designed to control the Taylor series accuracy by the use of the $E(h)$.

6. Algorithm Control

Algorithm control is also accomplished by an error measure. After each application, as μ increases, the effectiveness can be estimated by measuring the values $E_\mu^1, E_\mu^2, \dots, E_\mu^{\sigma-1}$. If the value $\max |E_\mu^{\sigma-2}|, |E_\mu^{\sigma-1}|$ does not decrease as μ increases, the algorithm is halted. $A_{\mu-1}^{\sigma+1}$ is taken as the result of applying the algorithm to the Taylor series of length σ . If the algorithm has been completely applied, the result is taken as A_μ^σ . The error is then based on $E_{\mu-1}^{\sigma-2}$ and $E_{\mu-1}^{\sigma-1}$ since the appropriate values are not then available.

7. Performance

Returning to the example discussed at the beginning of this paper: $y' = -y^2$, $y(1) = 1$, which is integrated toward zero.

At $t = 10^{-1}$, the longest step length taken was 5.8×10^{-2} by the Gragg-Bulirsch-Stoer method. This step was much longer than that taken by other methods. The algorithm described in this paper took a step of 6.7×10^{-1} . With the default settings for accuracy of $\epsilon = 10^{-10}$ and the length of series set to 14, the method integrated up to $t = -1$ and produced an absolute error of 7×10^{-11} at this point (on the Cambridge Computing Laboratories Atlas machine, which has a 39-bit mantissa). This involved passing through the singularity at $t = 0$. This is accomplished with no extra user information concerning the position or order of the pole.

The new algorithm has been tested and found to be efficient and reliable over a large class of problems [see Willers [2]]. Comparisons have been made with other methods on the basis of observed final error versus computation time on the Cambridge Atlas computer. Three graphs are included here showing the performance of the Gragg-Bulirsch-Stoer method (B.S.), the Taylor series method (T.S.), and the new algorithm (N.A.)

on three different problems. The Gragg-Bulirsch-Stoer method, which is generally recognized as good, see [8], was written in Fortran specially for the Cambridge Atlas. The time for computation and code generation are not included, but for the Taylor series system and the new algorithm code, generation takes less than 0.1 sec of computing time for each problem. The error E was calculated using the L_1 norm, $E = \sum_{i=0}^n |E_i|$, where E_i is the error in the dependent variable y_i at the end of the range. Each point on the graph was produced by varying the minimum allowed accuracy per step.

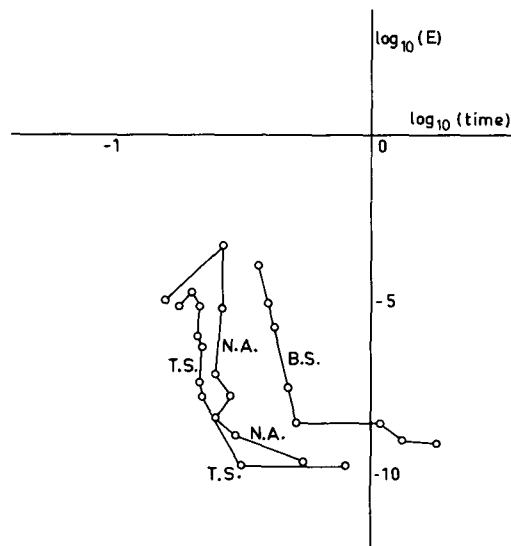
Problem 1

$$y' = z, \quad z' = -y(y^2 + z^2)^{-3/2}, \quad y = 0, \quad z = 1$$

when $t = 0$.

Range: $0 \leq t \leq 5$.

Solution: $y = \sin(t), z = \cos(t)$.

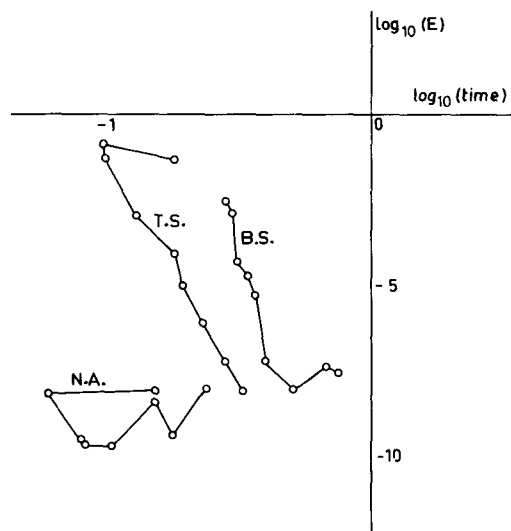


Problem 2

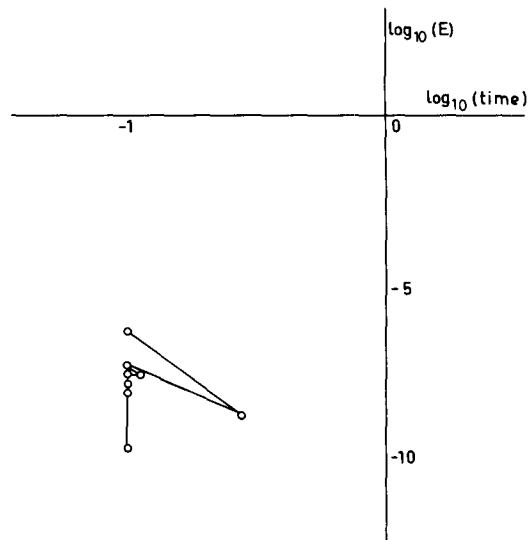
$$y' = 1 + y^2, \quad y = 1 \text{ when } t = 0.$$

Range: $0 \leq t \leq 0.75$.

Solution: $y = \tan[(\pi/4) + t]$.



Problem 3. This is the same case as problem 2 except that the range is extended to $0 \leq t \leq 1$. Only the new algorithm was capable of solving this problem owing to the singularity at $t = \pi/4 \approx 0.79$.



Acknowledgment. I would like to thank my supervisor at Cambridge, Dr. David Barton, for his continual help throughout this project.

Received July 1973; revised February 1974.

References

1. Barton, D., Willers, I.M., and Zahar, R.V.M. Taylor series methods for ordinary differential equations—an evaluation. Proc. Math. Software Symposium, Purdue U., Lafayette, Ind., 1970.
2. Willers, I.M. Mathematical software for the initial value problem. Ph.D. Th., U. of Cambridge, 1972.
3. Bulirsch, R., and Stoer, J. Numerical treatment of ordinary differential equations by extrapolation methods. *Num. Math.* 8 (1968), 1-13.
4. Gragg, W.B. On extrapolation algorithms for ordinary initial value problems. *SIAM J. Numer. Anal.* 2 (1965), 384-403.
5. Barton, D., Willers, I.M., and Zahar, R.V.M. An implementation of the Taylor series method for ordinary differential equations. *Compt. J.* 14, 3 (1971), 243-248; also in *The Best Computer Papers of 1971*, (Ed.) Petrocchi, Auerbach Philadelphia, 1971, pp. 147-163.
6. Bauer, F. L. The quotient difference and epsilon algorithms. On numerical approximation. U. of Wisconsin Press, Madison, Wis., 1959, pp. 361-370.
7. Shanks, D. Non-linear transformations of divergent and slowly convergent sequences. *J. Math. Phys.* 34 (1955), 1-42.
8. Hall, T.E., Enright, W.H., Fellen, B.M., Sedgwick, A.E. Comparing numerical methods for ordinary differential equations. *SIAM J. Numer. Anal.* 9 (1972), 603-637.