

An efficient predictor-corrector algorithm

D. Westreich

Department of Computer Engineering, MBT, Israel Aircraft Industries, Yahud, Israel

We develop a stable predictor-corrector algorithm for the solution of systems of first order differential equations. The algorithm requires two function evaluations at each step.

(Received July 1979)

Introduction

We give a new, efficient and stable predictor-corrector algorithm for the numerical solution of nonlinear systems of first order differential equations. The method is useful when the step size must be kept constant. Only two function evaluations are needed to compute the solution at each point.

Numerical experiments have shown it to converge much more rapidly than the traditional second order predictor-corrector methods. In general it converges only somewhat less slowly than the classical 4th order Runge-Kutta algorithm (which requires four function evaluations at each step).

The algorithm

Consider the system of first order differential equations

$$y' = F(t, y) \quad (1)$$

with initial value

$$y(t_0) = y_0$$

where y is an n vector to be determined and $F(t, y)$ is a continuous n vector function of t and y . To find the solution at t_f

1. Choose a step size h .
2. Find the solution y_1 to the equation at $t_1 = t_0 + h$ (say by a Runge-Kutta algorithm).

3. Set $i = 1$, $t_{i+1} = t_1 + h$, $y'_0 = F(t_0, y_0)$

$$y'_1 = F(t_1, y_1) \text{ and } I = 1$$

4. Let $I = -1$

5. Use the predictor

$$\bar{y}_{i+1} = y_{i-1} + 2h y'_i$$

to obtain an approximation to the solution at

$$t_{i+1} = t_0 + (i + 1)h.$$

6. Compute $\bar{y}'_{i+1} = F(t_{i+1}, \bar{y}_{i+1})$

7. If $I = -1$ go to step 12 otherwise go to step 8.

8. Use the corrector

$$\bar{y}_{i+1} = y_i + \cdot 5h(y'_i + \bar{y}'_{i+1})$$

to find an approximate solution at t_{i+1} .

9. Compute $y'_{i+1} = F(t_{i+1}, y_{i+1})$

10. Use the corrector of step 8 again to obtain

$$y_{i+1} = y_i + \cdot 5h(y'_i + y'_{i+1})$$

11. Go to step 14

12. Use the corrector

$$y_{i+1} = y_{i-1} + h(y'_{i-1} + 4y'_i + \bar{y}'_{i+1})/3$$

13. Compute

$$y'_{i+1} = F(t_{i+1}, y_{i+1})$$

14. If $t_{i+1} = t_f$ we stop, otherwise we let $i = i + 1$ and return to step 4.

Justification of the algorithm

Solving Equation (1) is equivalent to solving the Volterra integral equation

$$y(t) = y_0 + \int_{t_0}^t F(s, y(s)) ds. \quad (2)$$

Now if the interval $[t_0, t]$ is partitioned into subintervals of length h then the following numerical integration technique is appropriate for the evaluation of the right hand side of (2). When the number of subintervals are even we use the compound Simpson's quadrature rule and when it is odd we use the compound Simpson's rule until the last interval and then we use the trapezoid rule on the remaining interval. This method is numerically stable (Kershaw, 1974). In (Westreich and Cahlon, 1979) it is shown that this method is equivalent to the alternate corrector method used in our algorithm.

Numerical results

As an application of our algorithm we considered the equation

$$y'' - 2y^3 = 0$$

$$y(0) = 1.$$

$$y'(0) = -1.$$

which has the exact solution $y(t) = 1/(1 + t)$. We solved the equation on the interval $[0, 10]$ for step sizes $h = 0.1, 0.08, 0.05, 0.025, 0.01$ by converting it to a first order system. At each partition point we measured the deviation between the computed solution and the real solution. The largest of the absolute values of the deviations we called the maximum error. The solution at the second point was found by the classical 4th order Runge-Kutta algorithm. The results appear in the following table

h	maximum error
0.1	1.49 E-02
0.08	7.79 E-03
0.05	1.67 E-03
0.025	1.36 E-04
0.01	4.03 E-06

References

- KERSHAW, D. (1974). Volterra Equation of the Second Kind, in *Numerical Solution of Integral Equations*, L. M. Delves and J. Walsh, eds, Clarendon Press, Oxford, pp. 140-161.
- WESTREICH, D. and CAHLON, B. (1979). Solution of Volterra Integral Equations and 'Differential Equation' with Continuous or Discontinuous Terms.