stichting
mathematisch
centrum

MC

AFDELING NUMERIEKE WISKUNDE (DEPARTMENT OF NUMERICAL MATHEMATICS) NN 11/77 MAART

P.J. VAN DER HOUWEN

A COMPARISON OF NYSTRÖM-RUNGE-KUTTA AND LINEAR MULTISTEP METHODS FOR SECOND ORDER DIFFERENTIAL EQUATIONS WITH SLOWLY AND RAPIDLY OSCILLATING SOLUTIONS

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O).

AMS(MOS) subject classification scheme (1970): 65L05, M10, M20

A comparison of Nyström-Runge-Kutta and linear multistep methods for second order differential equations with slowly and rapidly oscillating solutions

by

P.J. van der Houwen

ABSTRACT

Numerical experiments are reported of the application of a Nyström-Runge-Kutta method and two linear three-step methods to a class of second order differential equations without first derivatives, the solution of which is assumed to consist of slowly and rapidly oscillating functions, the latter being small in magnitude.

KEY WORDS & PHRASES: *Nyström formulas, linear multisteps methods, special second order differential equations, hyperbolic equations*

CONTENTS

1. INTRODUCTION

This note studies numerical aspects of several algorithms designed for special second order differential equations of the form

$$\vec{y}'' = \vec{f}(x,\vec{y}), \tag{1.1}$$

where $\partial\vec{f}/\partial\vec{y}$ is assumed to have negative eigenvalues with a very large spread. Such equations arise from the partial discretization of *hyperbolic* differential equations (e.g. the wave equation) and are characterized by the occurrence of slowly and rapidly oscillating components in the solution, the rapidly oscillating functions being small in magnitude with respect to the slowly varying functions. We will respectively consider a third order, explicit three-step method, a second order, implicit three-step method and a second order, explicit Nyström-Runge-Kutta method. These formulas are designed for the integration of *hyperbolic* equations, that is they possess a relatively large interval of stability. In our numerical experiments we did not choose systems originating from hyperbolic partial differential equations, but we simulated such a situation by selecting *scalar* differential equations and initial values whose solutions are something like

$$y(x) = s(x) + \sin(\omega x) + \cos(\omega x), \tag{1.2}$$

where $s(x)$ is slowly varying function, $|s(x)| >> 1$ and $\omega >> 1$. Thus, we try to compute a function which behaves as shown in figure 1.1.

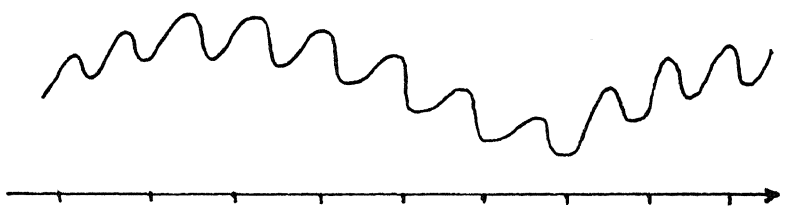

Fig. 1.1. Slowly varying solutions with rapidly oscillating perturbations

Since we are not interested in the rapid oscillations which are superposed on the drift function s(x), we would like to integrate with integration steps which are not prescribed by the rapid oscillations but rather by the function s(x).

From the numerical experiments, it may be concluded that the explicit three-step formula is the most accurate one, but requires relatively small integration steps; the implicit formula when applied with a single Newton step for the solution of the implicit equations yields a modest accuracy but remains stable for large steps and is therefore suitable for obtaining quickly a rough approximation to the solution. The Nyström-Runge-Kutta method is more accurate than the implicit one and more stable (larger steps) than the explicit one. This method is to be preferred in situations where the Jacobian matrix is not available or when its LU-decomposition is too expensive.

2. AN EXPLICIT THREE-STEP FORMULA OF THIRD ORDER

In reference [2] an *explicit* three-step formula of *third order* is given characterized by a relatively large interval of strong stability. This formula reads

$$\vec{y}_{n+1} = \frac{5}{2}\vec{y}_n - 2\vec{y}_{n-1} + \frac{1}{2}\vec{y}_{n-2} + \frac{1}{24}h^2[25\vec{f}_n - 14\vec{f}_{n-1} + \vec{f}_{n-2}] \tag{2.1}$$

h being the step length $x_{n+1} - x_n$ and $\vec{y}_n$ presenting the numerical approximation to the exact solution $\vec{y}(x_n)$. When the Jacobian matrix $\partial\vec{f}/\partial\vec{y}$ has negative eigenvalues δ, this formula is stable when

$$h < \sqrt{\frac{37}{10|\delta|_{max}}} . \tag{2.2}$$

3. AN IMPLICIT THREE-STEP FORMULA

In [2] also a family of *implicit* formulas is derived which are unconditionally stable provided that $\partial\vec{f}/\partial\vec{y}$ has a negative spectrum. This

family is given by

$$(3.1)\qquad \vec{y}_{n+1} = \frac{4+\varepsilon}{2}\vec{y}_n - (1+\varepsilon)\vec{y}_{n-1} + \tfrac{1}{2}\varepsilon\vec{y}_{n-2} +$$

$$+ \tfrac{1}{4}h^2[(1+\varepsilon)\vec{f}_{n+1} + 2(1-\varepsilon)\vec{f}_n + (1-\varepsilon)\vec{f}_{n-1}] ,$$

where $0 < \varepsilon < 2$. This implicit system has to be solved for $\vec{y}_{n+1}$. In order to maintain the unconditional stability, we use modified Newton-Raphson iteration to obtain after one interation step the approximation

$$(3.1')\qquad \vec{y}_{n+1} = \vec{y}_n + \tfrac{1}{2}[I-\tfrac{1}{4}(1+\varepsilon)h^2\frac{\partial\vec{f}^*}{\partial\vec{y}}]^{-1}$$

$$\cdot[(2+\varepsilon)\vec{y}_n - 2(1+\varepsilon)\vec{y}_{n-1} + \varepsilon\vec{y}_{n-2} + \tfrac{1}{2}(3-\varepsilon)h^2\vec{f}_n + \tfrac{1}{2}(1-\varepsilon)h^2\vec{f}_{n-1}],$$

where $\partial\vec{f}^*/\partial\vec{y}$ is some approximation to $\partial\vec{f}/\partial\vec{y}$ at the point $(x_n, \vec{y}_n)$. This formula requires one $\vec{f}$-evaluation per integration step and, assuming that $\partial\vec{f}/\partial\vec{y}$ is a slowly varying function, occasionally the evaluation of $\partial\vec{f}/\partial\vec{y}$ plus an LU-decomposition. In our experiments we have used formula (3.1'). It should be observed, however, that (3.1') is of *first order*, whereas formula (3.1) is of *second order* (cf. [2]). This is easily seen by deriving the truncation error of (3.1'). An elementary calculation yields that the residu left by substituting a sufficiently differentiable function $\vec{y}(x)$ into (3-1') is given by

$$(3.2)\qquad \tfrac{1}{4}(1+\varepsilon)h^3\left\{\frac{\partial\vec{f}}{\partial x}\Big|_n + \left[\frac{\partial\vec{f}}{\partial\vec{y}}\Big|_n - \frac{\partial\vec{f}^*}{\partial\vec{y}}\right]\vec{y}'(x_n)\right\} + O(h^4).$$

Thus, only for autonomous equations and exact Jacobian evaluations we have second order accuracy. Of course, by performing a second interation step we always have second order accuracy, but also an additional function evaluation and more storage is needed when implemented on a computer.

In view of our class of hyperbolic problems we have chosen $\varepsilon = 1$. Storage is then limited and the damping effect on the higher harmonics ($\delta << 0$) is of order $10/|h^2\delta|$, δ being the eigenvalue of $\partial\vec{f}/\partial\vec{y}$ of the corresponding harmonic.

4. A TWO-POINT NYSTRÖM-RUNGE-KUTTA FORMULA

In reference [1], Nyström-Runge-Kutta formulas are proposed with an extended real stability interval. A typical example is the two-point formula of *second order*

$$\vec{y}_{n+1} = \vec{y}_n + h\vec{y}'_n + \tfrac{1}{2}h^2\vec{f}(x_n+\tfrac{1}{2}h, \vec{y}_n+\tfrac{1}{2}h\vec{y}'_n+\lambda h^2\vec{f}^*(x_n+\mu h, \vec{y}_n+\mu h\vec{y}'_n)),$$

$$\vec{y}'_{n+1} = 2\,\frac{\vec{y}_{n+1}-\vec{y}_n}{h} - \vec{y}'_n ,$$

where $\lambda = .06373440810$ and $\mu = .4935439997$, and where $\vec{f}^*(x,y)$ is a function satisfying the relation

$$\frac{\partial \vec{f}}{\partial \vec{y}} \cong \frac{\partial \vec{f}^*}{\partial \vec{y}} \quad . \tag{4.2}$$

When this relation is approximately true, we have stability when

$$h < \sqrt{\frac{15.6}{|\delta|_{max}}} \quad . \tag{4.3}$$

Evidently, relation is satisfied when we choose $\vec{f}^* = \vec{f}$. In practice, however, it may be advantageous to replace $\vec{f}$ by a "cheaper" function $\vec{f}^*$. The effect of introducing $\vec{f}^*$ is restricted to the h^4-term and higher order terms. This may be concluded from the following. Replace $\vec{f}^*$ by $\vec{f}$ in formula (4.1) and denote the differences with the resulting $\vec{y}_{n+1}$ and $\vec{y}'_{n+1}$ vectors by $\Delta\vec{y}_{n+1}$ and $\Delta\vec{y}'_{n+1}$, respectively. We then find

$$\Delta\vec{y}_{n+1} \cong \tfrac{1}{2}\lambda h^4 \frac{\partial \vec{f}}{\partial \vec{y}}\Big|_n (\vec{f}^*-\vec{f}) + O(h^5),$$

$$\Delta\vec{y}'_{n+1} \cong \lambda h^3 \frac{\partial \vec{f}}{\partial \vec{y}}\Big|_n (\vec{f}^*-\vec{f}) + O(h^4). \tag{4.4}$$

Hence, we may expect that the use of "cheap" f^*-functions will be profitable when $\Delta\vec{y}_{n+1}$ and $\Delta\vec{y}'_{n+1}$ are of the order of magnitude of the tolerances

prescribed for $\vec{y}_{n+1}$ and $\vec{y}'_{n+1}$.

5. NUMERICAL EXPERIMENTS

The algorithms (2.1), (3.1') with $\varepsilon = 1$ and (4.1) were designed to integrate the large systems of differential equations with a large negative spectrum which originate from the semi-discretization of hyperbolic initial value problems. When the data are smooth, slowly varying functions, the solution of such systems consists of a large number of oscillating components, the rapid oscillations, however, being small in magnitude when compared with the slowly oscillating components. These rapidly oscillating functions correspond to the large negative eigenvalues of $\partial\vec{f}/\partial\vec{y}$ and they are responsible for the development of instabilities when the negative interval of stability of the numerical algorithm is too small with respect to the integration step used. Such is nearly always the case when the integration step is chosen by only considering slowly varying components and ignoring the unwanted rapid oscillations. Only when the algorithm used has a strong damping effect on the higher harmonics and when the length of the stability interval in terms of the integration step desired, has the order of magnitude of the maximal frequency of the unwanted oscillations, only then the integration process will be stable. It may be remarked that a similar situation arises in the integration of stiff differential equations of first order where the solution consists of slowly and rapidly decaying components, the latter being small in magnitude.

The algorithms mentioned above do have relatively large stability regions and a more or less strong damping effect on the higher harmonics; it is the purpose of this note to compare their mutual efficiency. As test problems we chose single equations of which the solutions consist of a slowly varying function and a very rapidly oscillating function. The initial values were chosen such that the oscillating component is very small in magnitude. In this way we simulate in a simple manner what is going on in a hyperbolic problem where one has to cope with hundred or thousand coupled differential equations. All experiments were carried out on a Hewlett-Packard 67.

The results of our experiments are indicated by listing the accuracy

obtained and the computational effort involved. As a measure of accuracy we took the number sd of significant digits, i.e.

$$(5.1) \qquad sd = -^{10}\log\left| \frac{y_n - y(x_n)}{y_n} \right|;$$

the computational effort of each experiment was expressed in numbers of right hand side evaluations. In (3.1') we did not count the evaluation of $\partial f^*/\partial y$ neither did we count the computational effort to perform the LU-decomposition. In many cases this is justified by the observation that in actual computation the Jacobian matrix is only occasionaly reevaluated.

In all test problems the integration step h was chosen in such a way that the quantity $h^2|\partial f/\partial y|$ assumes the values 56, 14, 3.5 and .875, respectively. Algorithm (3.1') is expected to be stable for all values of h, algorithm (4.1) is unstable for the largest h-value and stable for the other ones, and algorithm (2.1) is expected to be stable for the two smallest h-values and unstable for the larger step sizes.

Finally, in case of the three-step formulas the starting values were provided either by the analytic solution or, when the analytic solution is unknown, by applying algorithm (4.1) with $f^* = f$ and a sufficiently small integration step (in the following (4.1) will always denote the formula with $f^* = f$, whereas (4.1*) will be used when $f^* \neq f$ is meant).

5.1. A linear equation

Consider the equation

$$(5.2) \qquad y'' = -\omega^2(y-10-\sin x)-\sin x.$$

The general solution is given by

$$(5.3) \qquad y = 10 + \sin x + a \sin(\omega x) + b \cos(\omega x),$$

where a and b are the integration constants. For large values of ω this solution consists of a slowly varying component 10 + sin x and two rapidly

oscillating components a sin(ωx) and b cos(ωx). Let us choose a = b = 0, i.e.

(5.4) $\qquad y(0) = 10, \qquad y'(0) = 1.$

Analytically, the solution is slowly varying and, as can be seen from (5.2), the value of y" is bounded by 1. Numerically, however, y_n does not exactly equal 10 + sin(x_n) so that $y_n'' = f(x_n, y_n)$ may become very large for large values of ω, that is rapidly oscillating components will be introduced into the numerical solution.

In table 5.1 the number of correct digits sd and the number N of right hand side evaluations are given for integrating until the point

(5.5) $$x_e = 10\sqrt{\frac{56}{|\frac{\partial f}{\partial y}|_{\max}}} .$$

Table 5.1 Results for problem (5.2)-(5.4)
at $x_e = 2.366$ for $\omega^2 = 1000$

N	(2.1)	(3.1')	(4.1)	(4.1*)
10	<<0	1.9	<<0	<<0
20	<<0	2.1	<<0	.5
40	8.5	2.4	3.4	1.8
80	>10	2.7	5.0	1.4

As expected the implicit formula (3.1') is to be used when a rough picture of the solution is desired for less computation time. As soon as high accuracy is wanted, formula (2.1) is superior. The Nyström-Runge-Kutta formula (4.1) with $f^* = f$ is inferior in this example. Also the "economized" version (4.1*) with

$$f^* = \omega^2(10-y),$$

saving the evaluation of sin x is not of help which may be explained by applying estimate (4.4); for the smallest integration step $h = [.875/|\partial f/\partial y|]^{\frac{1}{2}} \cong .03$ we find

$$\Delta y_{n+1} \cong .025 \sin x_n, \qquad \Delta y'_{n+1} \cong 1.7 \sin x_n$$

thus, at some points ($\sin x_n \cong \pm 1$) the solutions obtained by (4.1) and (4.1*) have less than one significant digit in common ($-^{10}\log(\Delta y'_{n+1}/y_{n+1})$).

5.2. A non-linear problem

Consider the initial value problem

$$y'' = -\omega^2[(y-\sin x)^3 - 10^3] - \sin x$$

(5.6)

$$y(0) = 10, \qquad y'(0) = 1.$$

Evidently, the solution is given by

(5.7) $$y(x) = 10 + \sin x.$$

All neighbouring integral curves, however, will contain oscillating components with frequencies of order $300\omega^2$. The same series of experiments as done in the preceding section yields at the point x_e defined by (5.5) the results listed in table 5.2 (in applying (3.1') we took $\partial f^*/\partial y = -300\omega^2$)

Table 5.2 Results for problem (5.6) at $x_e = .43$ for $\omega^2 = 100$

N	(2.1)	(3.1')	(4.1)
10	<<0	2.4	<<0
20	<<0	2.7	<<0
40	8.2	3.0	5.1
80	9.0	3.4	6.7

A comparison of table 5.1 and 5.2 reveals that the three algorithms show a similar behaviour for the linear problem (5.2) and the non-linear problem (5.6).

5.3. A problem where "cheap" right hand side evaluations are allowed

Consider the initial value problem

$$y'' = -\omega^2(y^3-10^3) + c(x)$$

$$y(0) = 10, \quad y'(0) = 0 \tag{5.8}$$

where $c(x)$ is some complicated function of x. By choosing in (4.1)

$$f^*(x,y) = -\omega^2(y^3-10^3)$$

this formula becomes effectively a "one-function-evalutaion" method and by virtue of its large stability interval is able to integrate with twice as large integration steps as formula (2.1). In the following tables a few results are presented.

Table 5.3 Results for problem (5.8) at the point
$x_e = .432...$ with $\omega^2 = 100$ and $c(x) = \exp(-x)$;
$y(x_e) \cong 9.99998840$

N	(2.1)	(3.1')	(4.1)	(4.1*)
10	<<0	5.5	<<0	<<0
20	<<0	5.5	<<0	4.6
40	6.5	5.5	4.7	5.2
80	7.4	5.5	5.3	7.8

Table 5.4 Results for problem (5.8) at the point x_e = .432... with ω^2 = 100 and c(x) = 10 exp(-x); $y(x_e) \cong 9.9998838$

N	(2.1)	(3.1')	(4.1)	(4.1*)
10	0	5.3	<< 0	0
20	0	5.3	<< 0	4.4
40	5.4	5.3	4.8	4.3
80	6.1	5.3	4.8	6.3

5.4. Damping of higher harmonics

Finally, we study the damping effect of the three algorithms on the rapidly oscillating components. We shall do this by integrating the initial value problem

$$y'' = -\omega^2(y^3-10^3)$$

(5.9) ,

$$y(0) = 10, \quad y'(0) = 0$$

with perturbed initial values, that is we take y(0) = 11, y'(0) = 1. Integration is performed using the maximal integration step allowed by the stability condition of the formula. In case of the unconditionally stable formula (3.1'), integration is performed with the steps used in the other formulas

Table 5.5 Propagation of initial errors at x_e = .4 for ω = 100

Algorithm	h	x_e/h	$y_n-y(x_n)$	reduction/step
(2.1)	.01	40	.0058	.88
(3.1')	.01	40	.000027	.77
(3.1')	.02	20	.000002	.52
(4.1)	.02	20	.0767	.88

Table 5.5 shows that the implicit algorithm (3.1') has the strongest damping and algorithm (2.1) and (4.1) are comparable in this respect. Furthermore, we see that algorithm (3.1') reduces errors more as h is larger; this is in agreement with the asymptotic behaviour of the amplification factor (compare the discussion in section 3).

REFERENCES

[1] HOUWEN VAN DER, P.J., *Stabilized Runge-Kutta methods for second order differential equations without first derivatives*, Report NW 26/75, Mathematisch Centrum, Amsterdam (1975).

[2] _____ : *Linear multistep methods for a class of hyperbolic differential equations*, Report NW 36/77, Mathematisch Centrum, Amsterdam (1977).