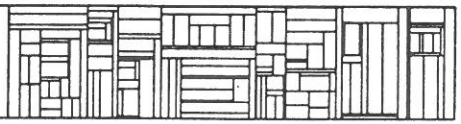
ISSN 0105-8517

Step Change Strategies for Multistep Methods

Ole Østerby

DAIMI PB - 196
August 1985

DATALOGISK AFDELING
Bygning 540 - Ny Munkegade - 8000 Aarhus C
tlf. (06) 12 83 55, telex 64767 aausci dk
Matematisk Institut Aarhus Universitet



Step Change Strategies for Multistep Methods

Ole Østerby
Datalogisk Afdeling
Århus Universitet
Ny Munkegade
8000 Århus C
Danmark

Abstract

When a system of ordinary differential equations is solved using a step-by-step method it is often desirable to change the step size during the course of the integration. We show that the commonly used formulas for calculating the new step sizes are not correct for multistep methods and we derive correct formulas for Adams methods.

1. Introduction

The two main reasons for adjusting the step size of an ODE-solver are stability and error control. We shall focus our attention on the latter. The main objective is to keep the local error (or an estimate of it) below a certain tolerance, ε, but not too much below. The local truncation error for a multistep formula of order p is usually written in the form

$$\text{local truncation error} = C_{p+1}\, h^{p+1}\, y^{(p+1)}(x_n) + o(h^{p+1}) \tag{1.1}$$

If the norm of the local error estimate, est, is larger than, or much smaller than ε then based on formula (1.1) a new step size is calculated as

$$h_{new} = h_{old} \cdot (\varepsilon/est)^{1/p+1} \tag{1.2}$$

Formulas of type (1.2) are often used in existing ODE software but they are not correct for multistep methods. Formula (1.1) is based on the assumption that the step size, h, is the same throughout the last p steps and can therefore not be used when we wish to change the step size.

The most serious effect of this is that when the step size must be reduced it is not reduced enough when (1.2) is used. Or in other words: the next step will give rise to a local error estimate which is bigger than expected and the step must be rejected.

Why hasn't this been noticed before?
Well it has, and this has led some practitioners to relax the error criterion for a couple of steps after a step change, while others have circumvented the problem by introducing a "safety factor" into (1.2) reducing the new step size by a certain fraction. Such safety factors are needed anyway because of variations in $y^{(p+1)}$ and this might have obscured the true nature of things.

2. Definitions

The task is to solve numerically the initial value problem

(2.1) $\quad y' = f(x,y), \qquad a \leq x \leq b,$

(2.2) $\quad y(a) = y_o$

We shall assume that f is sufficiently smooth and that the problem is non-stiff such that step size selection is governed by accuracy rather than stability.

The problem will be solved by a multistep method, more specifically by a predictor-corrector method based on Adams formulae. In this way we produce a set of approximate solution values $\{y_n\}$ corresponding to values of the independent variable $\{x_n\}$ satisfying $a = x_o < x_1 < \cdots < x_n < \cdots < x_N = b$. The corresponding values of the true solution to (2.1)-(2.2) are denoted $\{y(x_n)\}$.

The <u>global error</u> at $x = x_n$ is defined as

(2.3) $\quad$ global error $= y(x_n) - y_n.$

This is the quantity which the user presumably is interested in keeping track of. There exist various methods for estimating the global error [9,14,15] but global error estimates are not easy to utilize for step size selection [1].

Ordinary differential eguations have no "memory" in the sense that if we have drifted away from the true solution curve because of errors due to rounding or truncation etc. then we have no way of telling that we are not on the right track or how far away we are from it. The accuracy of ODE solvers is therefore usually measured in terms of the local behaviour.

We define the <u>local solution</u> at x_{n-1} to be the solution of the initial value problem

(2.4) $\quad u' = f(x,u),$

(2.5) $\quad u(x_{n-1}) = y_{n-1},$

i.e. u (which could be given the subscript n-1) is that solution of the differential equation which passes through the previous computed point (x_{n-1}, y_{n-1}).

We can now define the <u>local error</u> at $x = x_n$ as

(2.6) $\quad$ local error $= u(x_n) - y_n.$

A third concept is the local truncation error which is defined as the discrepancy which appears when the true solution is plugged into the numerical formula. For a fixed-step multistep formula such as

$$(2.7)\qquad \sum_{j=o}^{k} \alpha_j y_{n+j} = h \sum_{j=o}^{k} \beta_j f(x_{n+j}, y_{n+j}),\ \alpha_k = 1,$$

We therefore have

$$(2.8)\qquad \text{local truncation error} = \sum_{j=o}^{k} \{\alpha_j y(x_{n+j}) - h\beta_j y'(x_{n+j})\}.$$

Using Taylor expansions we can write the local truncation error on the form ([6])

$$(2.9)\qquad \text{local truncation error} = C_{p+1} h^{p+1} y^{(p+1)}(x_n) + C_{p+2} h^{p+2} y^{(p+2)}(x_n) + \ldots$$

or

$$(2.10)\qquad \text{local truncation error} = C_{p+1} h^{p+1} \{y_i^{(p+1)}(x_n + \theta_i h)\},\quad 0 < \theta < k.$$

The local truncation error is not the same thing as the local error but if the local error is expanded in a way similar to (2.9) then it can be shown that for Runge-Kutta methods and multistep methods of Adams type the leading terms are identical. We say that the local error and the local truncation error are asymptotically equal where asymptotical refers to the limit $h \to 0$.

3. Errors and step sizes

Neither the local error nor the local truncation error can be computed but there are various ways of estimating their size. When using a pair of multistep formulas of the same order as a predictor-corrector pair (e.g. an Adams-Bashforth-Moulton method) then the local error can be estimated using Milne's device. If a class of Runge-Kutta or multistep methods of different orders is used then the local error of a lower order formula can be estimated by comparing it with a higher order formula.

In either case we can obtain an asymptotically correct estimate of the local truncation error (and for Adams and Runge-Kutta methods therefore also of the local error), i.e. an estimate which when expanded in a Taylor series has the same leading term as (2.9) and tends to (2.9) as $h \to 0$.

In practical computations the step size will usually not tend to zero, but the underlying assumption is that h is small enough for the leading term to dominate such that the local error estimate gives a fair picture of the size of the local (truncation) error. This will usually be the case when operating with strict error tolerances whereas the error estimates are often noted to be poor when the accuracy required is low.

Whenever the numerical solution of the ODE is taken one step further the local error estimate (est) is computed and compared to a tolerance (ε). If est > ε then the step is rejected and we must try again with a smaller step size. On the other hand if est << ε then we ought to increase the step size in order to reduce the amount of computation.

When using a multistep method we shall not want to change the step size too often, i.e. we should like to keep the same step size for 5-10 steps or maybe even more. But more important is to avoid large variations such as halving or doubling the step.

A special problem is how to choose the initial step size. This has been considered in [5,12,17].

Once the integration has been started in a reasonable manner and the local error can be expected to satisfy (2.10) then any changes in the step size must be due to variations in $y^{(p+1)}$ and if f is smooth then so are these variations.

A value of the local error estimate which would result in (more than) halving or doubling the step size indicates a failure of our assumptions about f and special measures should be taken. A safe assumption might be that (some component of) f has a discontinuity and this should be treated in an appropriate way [4].

Early techniques in connection with step doubling amounted to remembering and using information at every other point backwards in time, and a similar strategy with step halving was to interpolate to the missing half-way points. Since we are considering less dramatic step-variations we shall not use such techniques but rather use the information from the last k steps whether it be in the form of modified divided differences [13] or in a Nordsieck representation [2,8] or anything equivalent, and whether the last k steps have the same size or not.

4. Changing the step size

When calculating a new step size most existing codes have been based on formulas such as (2.10) which suggest that the local error varies with h roughly as h^{p+1} for a method of order p. The formula for the new step size becomes

$$(4.1) \qquad h_{new} = h_{old} \cdot (\varepsilon/est)^{1/p+1}$$

Formula (4.1) does not take into account variations in $y^{(p+1)}$ and would therefore lead to many step failures if used as it stands.

Normal practice is therefore to introduce a "safety factor" into (4.1) either as

$$(4.2) \qquad h_{new} = h_{old} \cdot (\gamma \cdot \varepsilon/est)^{1/p+1}, \qquad 0 < \gamma < 1,$$

or as

$$(4.3) \qquad h_{new} = \delta \cdot h_{old} \cdot (\varepsilon/est)^{1/p+1}, \qquad 0 < \delta < 1.$$

For fixed order methods these two approaches are equivalent (with $\gamma = \delta^{p+1}$). For variable order methods (4.2) aims consistently at a local error estimate of magnitude $\gamma \cdot \varepsilon$ (rather than ε) whereas (4.3) aims at lower and lower values as the order increases. This latter effect has been quite beneficial since, as we shall see in the following, the error in (4.1) increases with the order.

It should be mentioned here that we have in mind an error-per-step (EPS) strategy [10,12] requiring the local error estimate to be less than ε at each step. Straightforward modifications in the following formulas will enable the reader to derive formulas for the error-per-unit-step (EPUS) strategy where the local error estimate is required to be less than ε times the step size.

Formula (4.1) is based on (2.10) which is derived under the assumption of a fixed step size. Since we have in mind changing the step from h_{old} to h_{new} in the next step we are not in the fixed step situation any longer and (2.10) does not hold for the next step(s). The new step size as calculated from (4.1) will therefore not lead to a local error (estimate) of size ε as intended.

What we need is therefore formulas for the local truncation error to replace (2.10) and based on which a sounder step size calculation can be made.

5. Variable step formulas

In order to derive the Adams formulas with variable step size we integrate Newton's interpolation formula

$$(5.1)\qquad f(x) = f(x_n) + (x-x_n)f[x_n,x_{n-1}] + (x-x_n)(x-x_{n-1})f[x_n,x_{n-1},x_{n-2}] + \cdots$$

$f[x_n,x_{n-1},\cdots,x_{n-j}]$ denotes the j-th divided difference of f at the points $x_n,x_{n-1},\cdots,x_{n-j}$. The remainder term for a k-step formula is the k+1-st term of (5.1) with x_{n-k-1} in the divided difference replaced by x:

$$(5.2)\qquad R_k = (x-x_n)\cdots(x-x_{n-k})\cdot f[x_n,\cdots,x_{n-k},x]$$

We now get an integration formula for the interval [a,b] by integrating (5.1) and (5.2):

$$(5.3)\qquad \int_a^b f(x)dx = f(x_n)\cdot(b-a) + f[x_n,x_{n-1}]\cdot\int_a^b (x-x_n)dx + \cdots$$

and the error term is

$$(5.4)\qquad E_k = \int_a^b (x-x_n)\cdots(x-x_{n-k})\cdot f[x_n,\cdots,x_{n-k},x]\,dx.$$

For an Adams-Bashforth explicit formula we should choose $[a,b] = [x_n,x_{n+1}]$ and for an Adams-Moulton implicit formula $[a,b] = [x_{n-1},x_n]$. In both cases the product $(x-x_n)\cdots(x-x_{n-k})$ is of constant sign in (a,b) and we can use the mean value theorem on (5.4):

$$(5.5)\qquad E_k = f[x_n,\cdots,x_{n-k},\xi]\cdot\int_a^b (x-x_n)\cdots(x-x_{n-k})\,dx$$

where $\xi \in (x_{n-k},b)$. Again there is a close correspondence between the error term and the k+1-st term in (5.3).

In order to find formulas for the integrals in (5.3) and (5.5), we introduce the following

Definition Let $P_i(x;x_{n-1},x_{n-2},\cdots,x_{n-k})$ be the sum of all products of i factors out of $(x-x_{n-1}),(x-x_{n-2}),\cdots,(x-x_{n-k})$.

Examples ($k \geq 1$):

$$(5.6)\qquad P_k(x;x_{n-1},x_{n-2},\cdots,x_{n-k}) = \prod_{i=1}^{k}(x-x_{n-i})$$

$$(5.7)\qquad P_{k-1}(x;x_{n-1},x_{n-2},\cdots,x_{n-k}) = \sum_{i=1}^{k}\prod_{j\neq i}(x-x_{n-j})$$

$$(5.8)\qquad P_1(x;x_{n-1},x_{n-2},\cdots,x_{n-k}) = \sum_{i=1}^{k}(x-x_{n-i})$$

$$(5.9)\qquad P_0(x;x_{n-1},x_{n-2},\cdots,x_{n-k}) = 1 \qquad (k\geq 0)$$

The following properties of P_i will be useful:

$$(5.10)\qquad P_i(x_{n-1};x_{n-1},x_{n-2},\cdots,x_{n-k}) = \begin{cases} 1 & \text{if } i=k=0\\ 0 & \text{if } i=k\neq 0\\ P_i(x_{n-1};x_{n-2},\cdots,x_{n-k}), & \text{if } i<k\end{cases}$$

$$(5.11)\qquad P_i(x;x_{n-2},\cdots,x_{n-k}) = (x-x_{n-k})\cdot P_{i-1}(x;x_{n-2},\cdots,x_{n-k+1}) + P_i(x;x_{n-2},\cdots,x_{n-k+1}) \qquad (i>0)$$

$$(5.12)\qquad \frac{d}{dx}P_i(x;x_{n-1},\cdots,x_{n-k}) = (k-i+1)\cdot P_{i-1}(x;x_{n-1},\cdots,x_{n-k}) \qquad (i>0)$$

$$(5.13)\qquad \int_a^b (x-x_n)\cdots(x-x_{n-k})\,dx = \sum_{j=0}^{k}\left[(-1)^j\frac{(x-x_n)^{j+2}}{(j+2)(j+1)}P_{k-j}(x;x_{n-1},\cdots,x_{n-k})\right]_a^b$$

The last relation is proved by differentiating the expression on the right-hand side, using (5.12) and (5.9) and observing that all but the first term in the summations cancel.

Using (5.13) with $(a,b) = (x_n, x_{n+1})$ we get

$$(5.14)\qquad \int_{x_n}^{x_{n+1}} (x-x_n)\cdots(x-x_{n-k})\,dx =$$

$$\sum_{j=0}^{k} (-1)^j \frac{(x_{n+1}-x_n)^{j+2}}{(j+2)(j+1)}\, P_{k-j}(x_{n+1};x_{n-1},\cdots,x_{n-k})$$

Similarly with $(a,b) = (x_{n-1}, x_n)$ and using (5.10):

$$(5.15)\qquad \int_{x_{n-1}}^{x_n} (x-x_n)\cdots(x-x_{n-k})\,dx =$$

$$-\sum_{j=0}^{k} \frac{(x_n-x_{n-1})^{j+2}}{(j+2)(j+1)}\, P_{k-j}(x_{n-1};x_{n-2},\cdots,x_{n-k})$$

Combining (5.3) and (5.14) and introducing the last step size $h = x_{n+1}-x_n$ we have the following variable step form of the Adams-Bashforth formula:

$$(5.16)\qquad \int_{x_n}^{x_{n+1}} f(x)\,dx = f(x_n)\cdot(x_{n+1}-x_n) +$$

$$+ \sum_{k\geq 0} f[x_n, x_{n-1}, \cdots, x_{n-k-1}]\cdot$$

$$\sum_{j=0}^{k} (-1)^j \frac{(x_{n+1}-x_n)^{j+2}}{(j+2)(j+1)}\, P_{k-j}(x_{n+1};x_{n-1},\cdots,x_{n-k})$$

$$= f(x_n)\cdot h$$

$$+ f[x_n, x_{n-1}]\cdot \tfrac{1}{2}h^2\cdot P_0$$

$$+ f[x_n, x_{n-1}, x_{n-2}]\cdot \{\tfrac{1}{2}h^2 P_1 - \tfrac{1}{6}h^3 P_0\}$$

$$+ f[x_n, \cdots, x_{n-3}]\cdot \{\tfrac{1}{2}h^2 P_2 - \tfrac{1}{6}h^3 P_1 + \tfrac{1}{12}h^4 P_0\}$$

$$+ f[x_n, \cdots, x_{n-4}]\cdot \{\tfrac{1}{2}h^2 P_3 - \tfrac{1}{6}h^3 P_2 + \tfrac{1}{12}h^4 P_1 - \tfrac{1}{20}h^5 P_0\}$$

$$+ \cdots$$

The parameters of P_i are $(x_{n+1}; x_{n-1}, \cdots, x_{n-k})$ in the term containing a divided difference of order k+1. If we further assume that the k previous steps have the same size, i.e.

$$(5.17)\qquad x_n - x_{n-1} = x_{n-1} - x_{n-2} = \cdots = c$$

then

$$(5.18)\qquad \int_{x_n}^{x_{n+1}} f(x)\,dx = f(x_n)\cdot h$$

$$+ f[x_n, x_{n-1}]\cdot \tfrac{1}{2}h^2$$

$$+ f[x_n, x_{n-1}, x_{n-2}]\cdot \tfrac{1}{6}h^2(3c+2h)$$

$$+ f[x_n, \cdots, x_{n-3}]\cdot \tfrac{1}{12}h^2\, 3(2c+h)^2$$

$$+ f[x_n, \cdots, x_{n-4}]\cdot \tfrac{1}{60}h^2\cdot 2\cdot\{90c^3+110hc^2+45h^2c+6h^3\}$$

$$+ \cdots$$

We have similar expressions for the integrals in the implicit Adams-Moulton formula. In this case $(a,b) = (x_{n-1}, x_n)$ and we set

$$(5.19)\qquad h = x_n - x_{n-1} \quad \text{and} \quad c = x_{n-1}-x_{n-2} = x_{n-2}-x_{n-3} = \cdots$$

$$\int_{x_{n-1}}^{x_n} f(x)dx = f(x_n)\cdot(x_n-x_{n-1})$$

$$- \sum_{k\geq 0} f[x_n,\cdots,x_{n-k-1}]\cdot \sum_{j=0}^{k} \frac{(x_n-x_{n-1})^{j+2}}{(j+2)(j+1)} P_{k-j}(x_{n-1};x_{n-2},\cdots,x_{n-k})$$

$$= f(x_n)\cdot h - f[x_n,x_{n-1}]\cdot\tfrac{1}{2}h^2$$

$$- f[x_n,x_{n-1},x_{n-2}]\cdot \tfrac{1}{6}h^3P_0$$

$$- f[x_n,\cdots,x_{n-3}]\cdot\{\tfrac{1}{6}h^3P_1+\tfrac{1}{12}h^4P_0\}$$

$$- f[x_n,\cdots,x_{n-4}]\cdot\{\tfrac{1}{6}h^3P_2+\tfrac{1}{12}h^4P_1+\tfrac{1}{20}h^5P_0\}$$

$$- f[x_n,\cdots,x_{n-5}]\cdot\{\tfrac{1}{6}h^3P_3+\tfrac{1}{12}h^4P_2+\tfrac{1}{20}h^5P_1+\tfrac{1}{30}h^6P_0\}$$

$$- \cdots$$

$$= \tfrac{1}{2}h\{f(x_n)+f(x_{n-1})\} \qquad (5.20)$$

$$- f[x_n,x_{n-1},x_{n-2}]\cdot\tfrac{1}{6}h^3$$

$$- f[x_n,\cdots,x_{n-3}]\cdot\tfrac{1}{12}h^3\{2c+h\}$$

$$- f[x_n,\cdots,x_{n-4}]\cdot\tfrac{1}{60}h^3\cdot\{20c^2+15hc+3h^2\}$$

$$- f[x_n,\cdots,x_{n-5}]\cdot\tfrac{1}{60}h^3\cdot\{60c^3+55hc^2+18h^2c+2h^3\}$$

$$- \cdots$$

Because of formula (5.5) we can directly read the componentwise error terms after the new step for the Adams formulas from formulas (5.16), (5.18) and (5.20). For a formula or order p we just have to replace x_{n-p} in the p-th order divided difference by ξ and disregard the other terms.

The traditional error terms for the fixed step size case are obtained by setting c = h and are perhaps easier recognized when we note that a p-th order divided difference can be written as $f^{(p)}(\eta)/p!$ where η is an intermediate point.

The formulas in this chapter were first derived in [3].

If h is much smaller than c then formulas (5.18) and (5.20) show that the local truncation error for Adams-Bashforth formulas is $O(h^2)$ and for Adams-Moulton formulas is $O(h^3)$ irrespective of the order as pointed out by Shampine [11]. But as already mentioned we are more interested in the case where h and c are comparable and we would therefore like to investigate the behaviour more closely in this region.

If a predictor-corrector method is composed of an Adams-Bashforth predictor and an Adams-Moulton corrector of the same order in say PECE mode then in the fixed step case the leading term of the local truncation error is equal to that of the corrector alone. In the variable step case this is not true if we define "leading term" as the term containing the lowest power of h; but since h and c are supposed to be comparable in magnitude it is more reasonable to define "leading term" as the sum of the terms containing the lowest collective power of h and c (which is p+1) and with this definition the statement retains its truth.

Our results for the Adams-Moulton formulas thus carry over to Adams predictor-corrector methods (in any mode) and it is therefore important to be able to extend (5.20) to arbitrary orders. Noting that $P_{k-j}(x_{n-1};x_{n-2},\cdots,x_{n-k})$ is a constant times c^{k-j} and using (5.11) we can write (5.20) as

$$(5.21)\qquad \int_{x_{n-1}}^{x_n} f(x)\,dx = \frac{1}{2}h\,\{f(x_n) + f(x_{n-1})\} - \sum_{k\geq 1} f[x_n,\cdots,x_{n-k-1}]\cdot \sum_{j=1}^{k} \frac{h^{j+2}\,c^{k-j}}{(j+2)(j+1)}\cdot d_{jk}$$

where

$$(5.22)\qquad d_{11} = 1,\quad d_{ok} = d_{k+1,k} = 0 \quad (1 \leq k)$$
$$d_{jk} = (k-1)\,d_{j,k-1} + d_{j-1,k-1},\ (1 \leq j \leq k).$$

The error term for a p-th order Adams-Moulton method can now be written as

$$(5.23)\qquad E_k = -\frac{y^{(p+1)}(\eta)}{p!} \sum_{j=1}^{k} \frac{h^{j+2}\,c^{k-j}}{(j+2)(j+1)}\,d_{jk}\,, \qquad (k = p-1)$$

A table of d_{jk} is given on the next page:

Table of d_{jk}

k \ j	1	2	3	4	5	6	7	8	9	10	11
1	1										
2	1	1									
3	2	3	1								
4	6	11	6	1							
5	24	50	35	10	1						
6	120	274	225	85	15	1					
7	720	1764	1624	735	175	21	1				
8	5040	13068	13132	6769	1960	322	28	1			
9	40320	109584	118124	67284	22449	4536	546	36	1		
10	362880	1026576	1172700	723680	269325	63273	9450	870	45	1	
11	3628800	10628640	12753576	8409500	3416930	902055	157773	18150	1320	55	1
(j+1)(j+2)	6	12	20	30	42	56	72	90	110	132	156

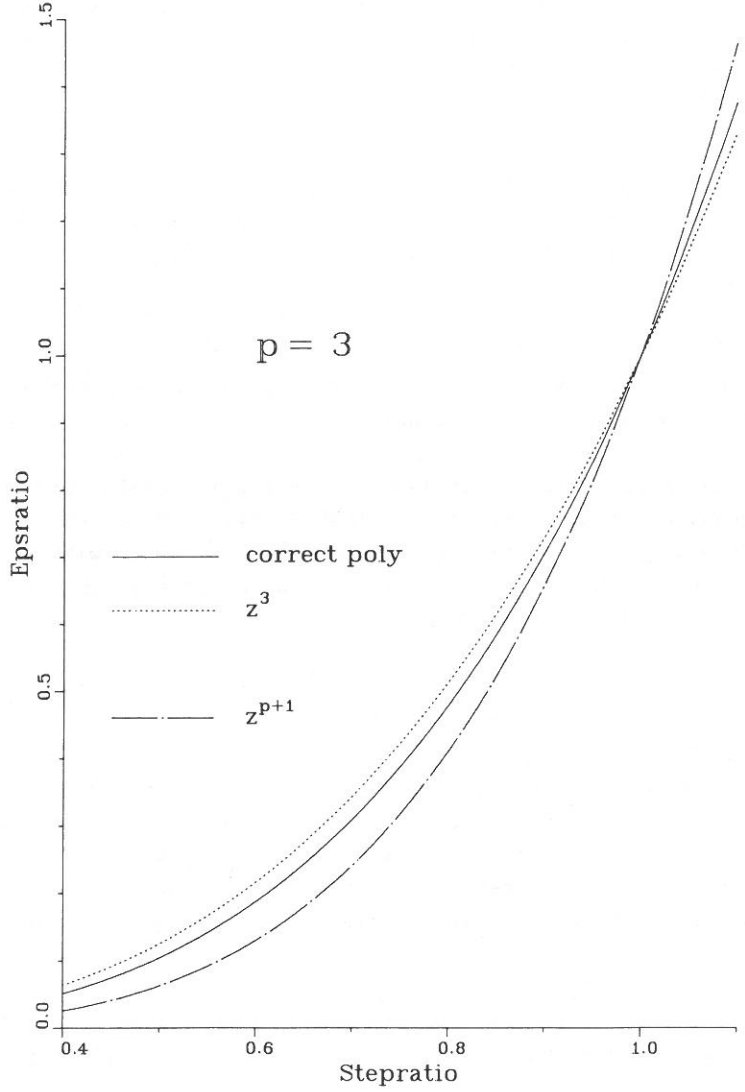


Fig.1. Graph of $Q_3(z), z^3$ and z^4.

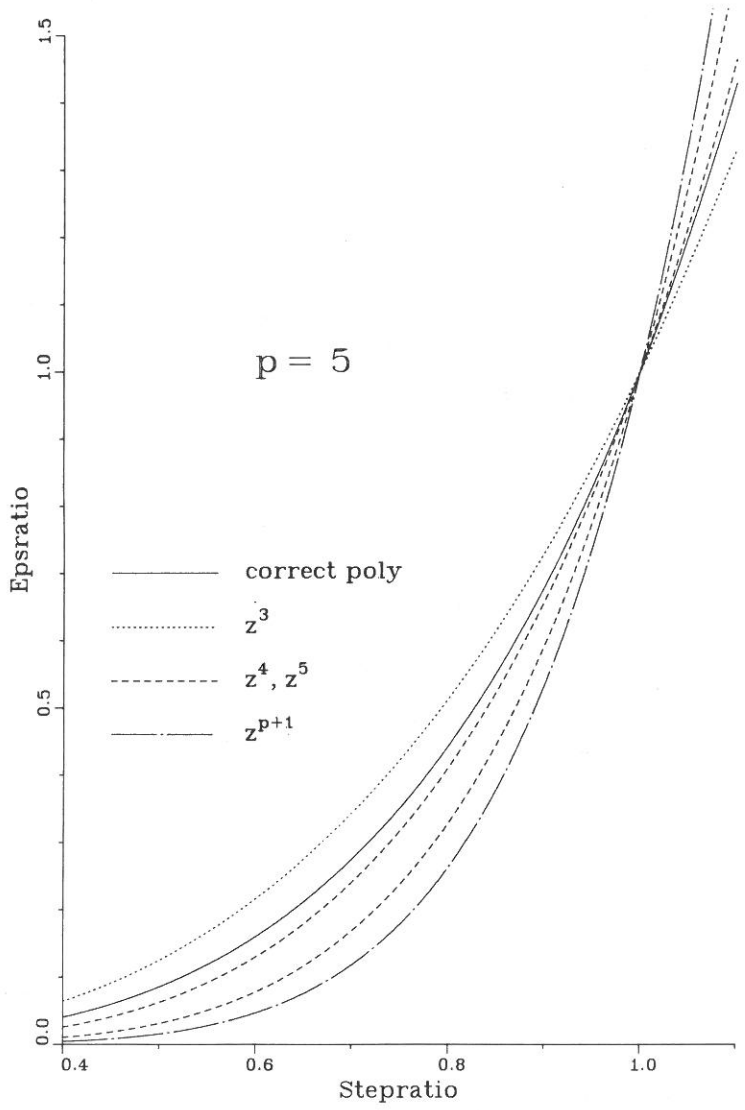


Fig.2. Graph of $Q_5(z), z^3, z^4, z^5$ and z^6.

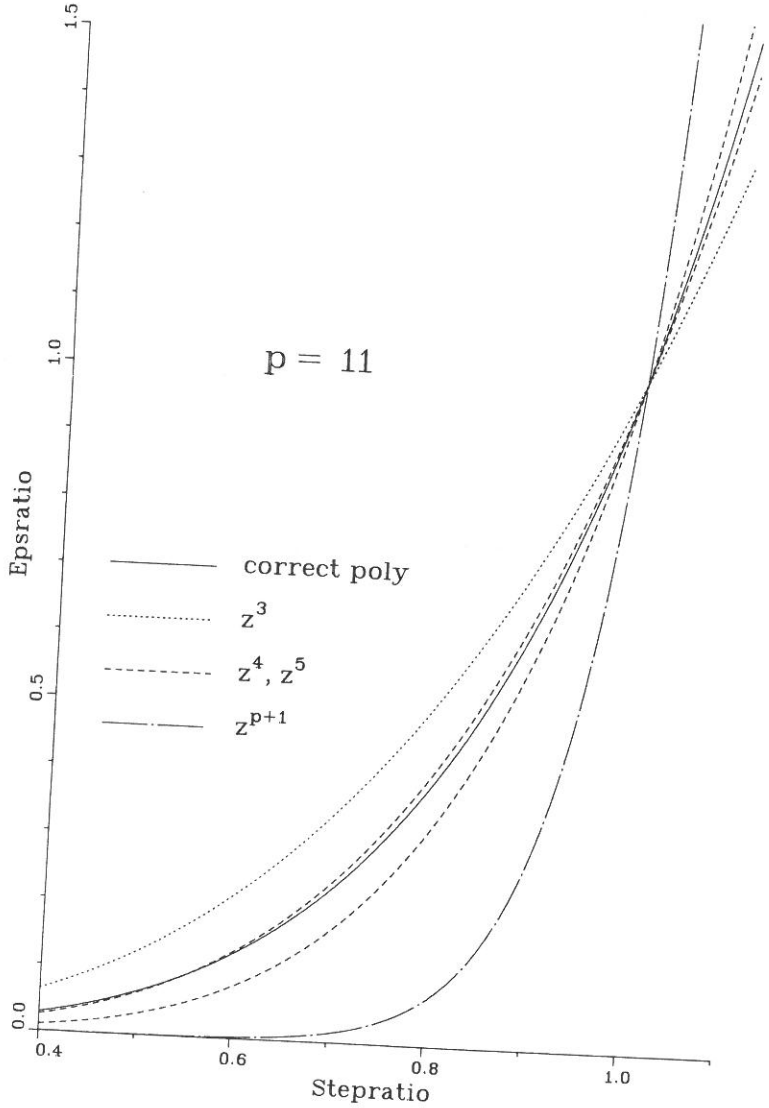


Fig.3. Graph of $Q_{11}(z), z^3, z^4, z^5$ and z^{12}.

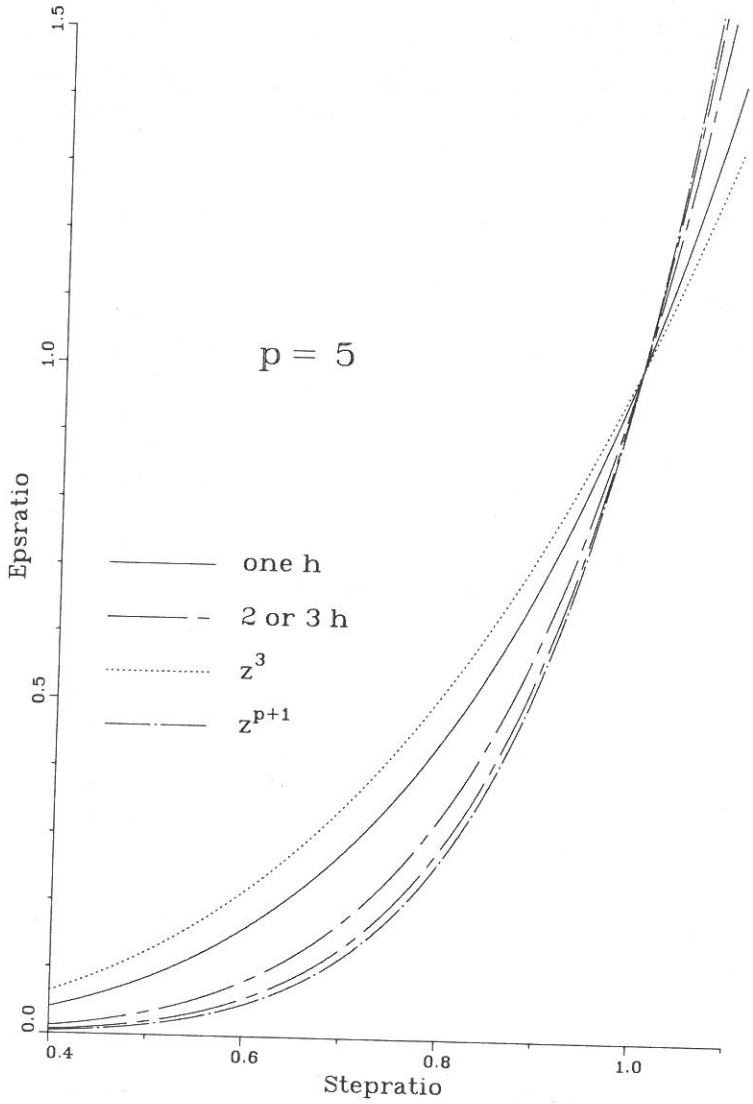


Fig.4. Graph of $Q_5(z)$ ("one h") and the polynomials corresponding to the last 2 and 3 step sizes being h, together with z^3 and z^6.

7. Practical suggestions

Based on the behaviour of $Q_p(z)$ as illustrated in Fig. 1-4 and the preceding example we can now offer the following practical suggestions:

a. If est << ε then compute a new step size by

(7.1) $$h_{new} = h_{old} \cdot (\gamma_1 \cdot \varepsilon/est)^{1/p+1}$$

γ_1 can be chosen close to 1, e.g. 0.8 or 0.9, and the << should mean that the step increase should be at least 10%. If the increase, however, is too large (say more than 100%) then this might be due to spurious variations in the local error (estimate) and we should limit the increase to 100%.

b. If est > ε then the last step must be rejected and another step should be computed as the positive root of

(7.2) $$Q_p(z) = \gamma_2 \cdot \varepsilon/est, \quad z = h_{new}/h_{old}.$$

γ_2 should be smaller than γ_1 since there is reason to believe that $\|y^{(p+1)}\|$ is increasing, e.g. $\gamma_2 = 0.6$ or 0.7.
If $h_{new} < h_{old}/2$ then f is probably not as smooth as we thought. Possibly we are in the neighbourhood of a discontinuity which must be found by a special technique [4].

Finding the root of (7.2) is not much worse than the traditional problem of finding a root of

(7.3) $$z^{p+1} = \gamma_2 \cdot \varepsilon/est.$$

In practice the difficulty lies in determining $Q_p(z)$ e.g. using the recurrence relations (5.22) or storing the various polynomials in question. The first alternative takes time and the second one takes computer memory.

If one is not willing to pay either of these prices a third alternative can be mentioned. As shown in the following lemma the positive root of

(7.4) $$z^3 = \gamma_2 \cdot \varepsilon/est$$

will always provide an underestimate of $h_{new} = z \cdot h_{old}$ when $\varepsilon < est$ such that using (7.4) will bring us on the safe side of the error tolerance. This will result in slightly smaller local errors and slightly more work but might be considered a reasonable alternative to (7.2). For $p \geq 9$ (7.4) can even be replaced by

(7.5) $$z^4 = \gamma_2 \cdot \varepsilon/est$$

but this is more a practical observation than a mathematically provable statement.

Lemma If $0 < \lambda < 1$ then $\lambda^{1/3}$ is smaller than the positive root of $Q_p(z) = \lambda$ for $p \geq 3$.

Proof Since $Q_p(z)$ is increasing for $z > 0$ the statement is equivalent to

$$Q_p(\lambda^{1/3}) \leq \lambda, \quad (p \geq 3).$$

but

$$Q_p(z) = \sum_{i=3}^{p+1} a_i z^i, \quad a_i > 0, \quad \sum_{i=3}^{p+1} a_i = 1$$

so for $0 < \lambda < 1$

$$Q_p(\lambda^{1/3}) = \lambda \cdot \sum_{i=3}^{p+1} a_i \lambda^{(i-3)/3} \leq \lambda \sum_{i=3}^{p+1} a_i = \lambda$$

□

8. Concluding remarks

The preceding ideas and suggestions have been tried on a few simple test equations and using an experimental fixed order code. These experiments have confirmed the relevance of the theory and we believe that implementation in existing ODE software will make this more efficient.

We believe that the present techniques will enable ODE solvers to stick rather closely to a given local error tolerance thereby possibly improving on the proportionality between the error tolerance and the global error which has been emphasized as a desirable property [16].

9. Acknowledgements

Part of this work has been carried out during a stay at University of Illinois at Urbana-Champaign supported in part by the Danish Science Research Council.

The author wishes to thank his colleagues, in particular C.W. Gear at Illinois and J. Sand at Aarhus for valuable discussions and constructive criticism.

References

[1] G. Dahlquist: On the control of the global error in stiff initial value problems, TRITA-NA-8106, Stockholm, 1981.

[2] C.W. Gear: Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, 1971.

[3] C.W. Gear and O. Østerby: Solving ordinary differential equations with discontinuities, UIUCDCS-R-81-1064, Urbana, 1981.

[4] C.W. Gear and O. Østerby: Solving ordinary differential equations with discontinuities, ACM Trans. Math. Software 10 (1984) 23-44.

[5] I. Gladwell: Initial value routines in the NAG library, ACM Trans. Math. Software 5 (1979) 386-400.

[6] J.D. Lambert: Computational Methods in Ordinary Differential Equations, John Wiley, 1973.

[7] B. Lindberg: Characterization of optimal stepsize sequences for methods for stiff differential equations, SIAM J. Numer. Anal. 14 (1977) 859-887.

[8] A. Nordsieck: On numerical integration of ordinary differential equations, Math. Comp. 16 (1962) 22-49.

[9] A. Prothero: Estimating the accuracy of numerical solutions to ordinary differential equations. I. Gladwell and D.K. Sayers (eds) Computational Techniques for Ordinary Differential Equations, Academic Press (1980) 103-128.

[10] L.F. Shampine: Local error control in codes for ordinary differential equations, Appl. Math. Comput. 3 (1977) 189-210.

[11] L.F. Shampine: The effect of changing step size on the accuracy of multistep formulas, SAND82-1584, Albuquerque, 1982.

[12] L.F. Shampine: The step sizes used by one-step codes for ODEs, Appl. Numer. Math. 1 (1985) 95-106.

[13] L.F. Shampine and M.K. Gordon: Computer Solution of Ordinary Differential Equations, W.H. Freeman, 1975.

[14] L.F. Shampine and H.A. Watts: Global error estimation for ordinary differential equations, ACM Trans. Math. Software 2 (1976) 172-186.

[15] H.J. Stetter: Global error estimation in ODE-solvers. G.A. Watson (ed) Numerical Analysis Dundee 1977, Springer Lecture Notes in Mathematics 630 (1978) 179-189.

[16] H.J. Stetter: Tolerance proportionality in ODE codes. R.D. Skeel (ed) Numerical Ordinary Differential Equations, SIGNUM, UIUCDCS-R-79-963, Urbana (1979) 10. 1-6.

[17] H.A. Watts: Starting step size for an ODE-solver, J. Comput. Appl. Math. 9 (1983) 177-191.