Peter Deuflhard

A Note on Extrapolation Methods for Second Order ODE Systems

Preprint SC 87-5 (September 1987)

Konrad-Zuse-Zentrum für Informationstechnik;
Heilbronner Straße 10; D-1000 Berlin 31

Peter Deuflhard

A Note on Extrapolation Methods for Second Order ODE Systems

Abstract

A slight modification of the extended Stoermer discretization for non-selfadjoint second order ODE systems is derived on the basis of a simple stability analysis. This discretization easily extends to implicit ODE systems, which are known to arise e.g. in mechanical engineering. In addition, a special variant of the semi-implicit Euler discretization is proposed, which essentially treats the state variables explicitly, but their derivatives implicitly. Numerical tests over critical parameter values of the van der Pol oscillator illustrate the domain of efficiency of the suggested discretizations.

Keywords: numerical integration, extrapolation methods.

Subject Classification: AMS(MOS): 65 L 05; CR: 5.17

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Heilbronner Strasse 10
1000 Berlin 31
Verantwortlich: Dr. Klaus André
Umschlagsatz und Druck: Verwaltungsdruckerei Berlin

Contents

The author wishes to thank S. Wacker for her quick and careful typing of the manuscript.

0. Introduction

Direct discretization of second order ODE systems is known to pay off in cases where only second derivatives appear in the system. In such cases the so-called Stoermer discretization is the basis of a rather efficient extrapolation method - see e.g. the survey article [3] or the recent textbook [8], p. 271-274. The associated quadratic asymptotic expansion was first shown to hold by Gragg [6].

An extension of the Stoermer discretization for the case, when the first derivatives are present, was given by the author together with a proof of the quadratic asymptotic expansion (cf. [3] and references therein). For a special subclass of such problems an extrapolation code (DIFEXM) has been implemented and tested by Bauer in [1].

The present note first reports about progress made beyond [1] in the development of the extended Stoermer discretization (section 1). The theoretical basis is a rather elementary stability model in the spirit of Dahlquist [2]. In section 2, the generalization to *implicit* ODE systems, which typically arise in mechanical engineering, is worked out. A special semi-implicit Euler discretization is derived as well. Finally, in section 3, numerical tests over critical parameter values of the van der Pol oscillator are included.

1. Extended Stoermer Discretization

Consider the initial value problem (IVP) for the second order system (of dimension n):

$$\begin{aligned} u'' &= f(u) + D(u)u' \\ u(0) &:= u_0\,,\ \ u'(0) := v_0\,. \end{aligned} \tag{1.1}$$

Herein typically D is a diagonal (n,n)-matrix representing the dissipativity of the system, in which case the entries of D are negative. This feature nicely shows up in the *model problem.*

$$u'' - \lambda u' + \omega^2 u = f(t) \quad , \tag{1.2.a}$$

which has

$$u_{\text{hom}}(t) := e^{\lambda t}\cos(\omega t + \alpha) \tag{1.2.b}$$

as homogeneous part of the solution. For $\lambda < 0$ *asymptotic stability* is guaranteed, whereas $\lambda > 0$ characterizes *inherent instability.* For large negative λ, the second order system may be regarded as "stiff".

Of course, (1.1) can be reformulated as a first-order system and then discretized using any method for stiff ODE's. However, in the present case, the "stiffness effect" is only caused by the u'-term. Hence, an efficient discretization of (1.1) needs only be *implicit* for non-vanishing u'-term. This property holds for the *extended Stoermer discretization* (stepsize h):

$$\begin{aligned} &u_0, v_0 \text{ given} \\ &u_1 := u_0 + h\left(v_0 + \frac{h}{2}\Big(f(u_0) + D(u_0)v_0\Big)\right) \end{aligned} \tag{1.3.a}$$

$$\begin{aligned} &k = 1,\ldots,l-1: \\ &\left(I - \frac{h}{2}D(u_k)\right)v_k = \frac{1}{h}\Big(u_k - u_{k-1}\Big) + \frac{h}{2}f(u_k) \\ &u_{k+1} := 2u_k - u_{k-1} + h^2\Big(f(u_k) + D(u_k)v_k\Big) \end{aligned} \tag{1.3.b}$$

$$\Big(I - \frac{h}{2}D(u_l)\Big)v_l = \frac{1}{h}(u_l - u_{l-1}) + \frac{h}{2}f(u_l)\,. \tag{1.3.c}$$

The existence of a *quadratic* asymptotic expansion of the discretization error is known. Let

$$\begin{aligned} D &:= \text{diag}\,(d_1,\ldots,d_n) \\ \mu &:= \max_i d_i(u)\ . \end{aligned} \tag{1.4}$$

Then a natural condition for the interval, within which the asymptotic expansion holds, will be

$$\mu\frac{h}{2} < 1\ . \tag{1.5}$$

For $\mu < 0$ (purely dissipative case), condition (1.5) holds for all $h \geq 0$.

Stability considerations. In order to analyze any discretization of (1.1), the model (1.2) suggests a further simplification in the spirit of [2] :

$$\begin{aligned} &u'' - \lambda u' = 0 \\ &u(0) = u_0, \;\; u'(0) = v_0 \end{aligned} \tag{1.6.a}$$

The associated analytic solution is

$$\begin{aligned} u(t) &= u_0 + \frac{v_0}{\lambda}(\exp(\lambda t) - 1) \\ u'(t) &= v_0 \exp(\lambda t) \end{aligned} \tag{1.6.b}$$

The asymptotic behavior for $\lambda t \to -\infty$ appears to be

$$u(t) \to u_0 \;, \;\; u'(t) \to 0 \;. \tag{1.6.c}$$

Of course, a similar behavior would be desirable for the discrete solution. For the analysis of discretization (1.3) applied to (1.6) one conveniently introduces the notation

$$\begin{aligned} \Delta_k &:= (u_{k+1} - u_k)/h \\ z &:= \lambda h \\ q &:= \left(1 + \frac{z}{2}\right) / \left(1 - \frac{z}{2}\right) \;. \end{aligned} \tag{1.7}$$

Straightforward calculation leads to

$$\begin{aligned} v_k &= q^k v_0 \\ \Delta_k &= \left(1 + \frac{z}{2}\right) q^k v_0 \\ u_l - u_0 &= h v_0 \left(1 + \frac{z}{2}\right) \frac{1 - q^l}{1 - q} \end{aligned} \tag{1.8}$$

In order to study the case $z \to -\infty$, first note that asymptotically

$$q \doteq (-1)\left(1 + \frac{4}{z}\right) \;.$$

Insertion into (1.8) then yields

$$u_l - u_0 \doteq h v_0 \left[\left(1 - (-1)^l\right)\frac{z}{4} - l(-1)^l\right] \;. \tag{1.9}$$

Let $lh = H$ and $u''(0) = \lambda v_0$. Then (1.9) implies

$$\begin{aligned} &\text{for } l = 2m > 0: \\ &u_l - u_0 \doteq -v_0 H \;, \;\; v_l \doteq v_0 \end{aligned} \tag{1.10.a}$$

for $l = 2m+1 > 0$:

$$u_l - u_0 \doteq v_0 H + \tfrac{1}{2}u''(0)h^2 \ , \quad v_l \doteq -v_0 \ . \tag{1.10.b}$$

Upon examining the implementation of (1.3), a cheaply computable symmetric final step appears to be (compare (2.2.c))

$$\hat{u}_l := \tfrac{1}{4}(u_{l-1} + 2u_l + u_{l+1}) \ , \tag{1.11.a}$$

which exhibits the asymptotic behavior

$$\hat{u}_l - u_0 \doteq \tfrac{1}{2}u''(0) \cdot h^2 \ , \ m > 0 \ . \tag{1.11.b}$$

Note that one step of extrapolation removes the λ-dependence - at least in the linear model problem.
The associated symmetric ending for the derivative variable

$$\hat{v}_l := \tfrac{1}{4}(v_{l-1} + 2v_l + v_{l+1}) \tag{1.12.a}$$

would exhibit the most desirable property

$$\hat{v}_l \doteq 0 \ , \tag{1.12.b}$$

but require an additional evaluation of the right-hand side for $t_{l+1} > H$.
Summarizing, the above analysis shows that a mixture of values l even and l odd should be avoided. Numerical experiments showed that the extrapolation method with double harmonic refinement sequence

$$\mathcal{F}_{2H} := \{2, 4, 6, 8, 10, \ldots\}$$

is slightly more accurate than a version with sequence

$$\mathcal{F}_{odd} := \{1, 3, 5, 7, 9, \ldots\} \ .$$

For this reason, numerical results only for $\mathcal{F}_{2H}$ are presented in section 3.

Reversibility. For some kinds of applications, especially in large scale boundary value problems, a desirable property is the symmetry of the discretization with respect to the transformation

$$(k, h) \to (l - k, -h) \ , \tag{1.13}$$

which is often called reversibility. Reformulation of (1.3.c) yields

$$u_{l-1} = u_l - h\left(v_l - \frac{h}{2}\Big(f(u_l) + D(u_l)v_l\Big)\right) \ , \tag{1.3'.c}$$

which by transformation (1.13) is just (1.3.a). Next, the second line of (1.3.b) is obviously reversible. Insertion of this line into the first line of (1.3.b) yields

$$\left(I - \frac{h}{2}D(u_{k+1})\right) v_{k+1} - \left(I + \frac{h}{2}D(u_k)\right) v_k = \frac{h}{2}\Big(f(u_{k+1}) + f(u_k)\Big) \quad , \qquad (1.3'.\mathrm{b})$$

which is now also seen to be reversible (implicit trapezoidal rule for u'). Of course, extrapolation or any "final step" at one interval end will destroy this property.

2. Treatment of Implicit Second Order ODE Systems

In mechanical engineering, the following class of problems arises typically:

$$\begin{aligned} M(u)u'' &= f(t,u) + D(u)u' \\ u(0) &= u_0\ , \quad u'(0) = v_0 \end{aligned} \tag{2.1}$$

Herein M is positive definite symmetric. Hence, there exists the Cholesky decomposition (let M=const.).

$$M = LL^T \quad , \quad L \text{ nonsingular} \ .$$

Now, introduce

$$\bar{u} := L^T u, \quad \bar{D} := L^{-1} D L^{-T} \ .$$

Then (2.1) may be rewritten as

$$\bar{u}'' - \bar{D}\bar{u}' = \bar{f}(t, \bar{u})$$

with properly defined $\bar{f}$. Herein $\bar{D}$ and D are *congruent*, which means that the sign structure of the eigenvalues is the same (Sylvester's law of inertia [5]). As a consequence, the stability considerations of the preceding section based on the model problem (1.6), should apply here as well.
In order to use discretization (1.3), multiply (2.1) by M^{-1} (note that M is assumed to be nonsingular). Remultiplication by M and introduction of the notation

$$\Delta_k := \frac{1}{h}(u_{k+1} - u_k)\ , \quad t_k := t_0 + k \cdot h, k = 0, 1, \ldots$$

then yields (including final step (1.11)):

u_0, v_0 given

$$\begin{aligned} M(u_0)\bar{\Delta}_0 &:= \frac{h}{2}\Big(f(t_0,u_0) + D(u_0)v_0\Big) \\ \Delta_0 &:= \bar{\Delta}_0 + v_0 \end{aligned} \tag{2.2.a}$$

$k = 1, \ldots, l$:

$$\begin{aligned} u_k &:= u_{k-1} + h\Delta_{k-1} \\ \left(M(u_k) - \frac{h}{2}D(u_k)\right)\bar{v}_k &:= \frac{h}{2}\Big(f(t_k,u_k) + D(u_k)\Delta_{k-1}\Big) \\ v_k &:= \Delta_{k-1} + \bar{v}_k \\ M(u_k)\bar{\Delta}_k &:= h(f(t_k,u_k) + D(u_k)v_k) \\ \Delta_k &:= \Delta_{k-1} + \bar{\Delta}_k \end{aligned} \tag{2.2.b}$$

$$\hat{u}_l := u_l + \frac{h}{4}\bar{\Delta}_l \tag{2.2.c}$$

Obviously, the discretization requires the solution of two different linear n-systems per step, which may be prohibitive when the computing costs are dominated by the linear algebra part or when storage is restricted.

An alternative treatment of (2.1) would be to rewrite it as a first order system (with $v = u'$) and apply the semi-implicit Euler discretization with h-extrapolation - see [3]. If one exploits the special structure of the thus arising linear systems, then (n, n)-matrices of the form

$$M - hD - h^2(f_u + D_u u' - M_u u'')$$

have to be decomposed. In the light of the stability considerations of the preceding section, however, one may as well apply an *implicit Euler scheme* just to the variable v - which leads to $(k = 0, 1, \ldots)$:

$$\begin{aligned} M(u_k)\frac{v_{k+1} - v_k}{h} - D(u_k)v_{k+1} &= f(t_k, u_k) \\ \frac{u_{k+1} - u_k}{h} &= v_{k+1} \ , \end{aligned} \tag{2.3}$$

or, equivalently:

$$\begin{aligned} \big(M(u_k) - hD(u_k)\big)\Delta v_k &:= h\big(f(t_k, u_k) + D(u_k)v_k\big) \\ v_{k+1} &:= v_k + \Delta v_k \\ u_{k+1} &:= u_k + h \cdot v_{k+1} \end{aligned} \tag{2.3'}$$

Compared with (2.2), only one type of linear n-system arises. On the other hand, h^2-extrapolation for (2.2) is replaced by h-extrapolation for (2.3).

Stability considerations. Application of discretization (2.3) to the model equation (1.6) yields $(z := \lambda h)$:

$$v_k = v_0/(1 - z)^k \tag{2.4}$$

This implies the asymptotic behavior (for $Re(z) \to -\infty$):

$$v_l \doteq 0 \ , \quad u_l - u_0 \doteq 0 \ , \tag{2.5}$$

which is just the desirable property (1.6.c) independent of any restriction on the refinement sequence. Hence, the harmonic sequence $\mathcal{F}_H$ can be used.

Remark 1. Both discretizations can be generalized to the case when u' appears nonlinearly in the right-hand side of (2.1). The proper generalizations might be efficient only, when special structure of the problem class can be exploited as, for instance, in regular celestial mechanics or mechanical engineering.

Remark 2. Note that discretization (2.3) can be extended to the case of singular M (differential-algebraic systems), if only the matrix pencil

$$\{M - \gamma D\}$$

remains regular. This property does not hold for discretization (2.2).

3. Numerical Comparisons

A small but nevertheless both typical and challenging test problem is the *van der Pol oscillator*. This example is either given as

$$\begin{aligned} u'' &= \alpha(1-u^2)u' - u \\ u(0) &= 2, \quad u'(0) = 0 \end{aligned} \tag{3.1}$$

or, after rescaling $t \to t/\alpha$, as

$$\begin{aligned} \varepsilon u'' &= (1-u^2)u' - u \\ \varepsilon &= 1/\alpha^2 \,. \end{aligned} \tag{3.1'}$$

For test purposes, (3.1) was solved over the interval $[0, T]$ with

$$T := 2(3 - \ln 2)\alpha \ ,$$

which is equivalent to solving (3.1') over $[0, T/\alpha]$. In this problem, the stiffness effect can be nicely studied by variation of α for $\alpha \gg 1$.

The subsequent test covers results from the following extrapolation codes:

E1: based on the semi-implicit Euler discretization [3], newest EULSIM version from [4],

DIFEXM: based on the extended Stoermer discretization [3],

E2: based on the second order semi-implicit Euler discretization, section 2 herein, code EUSIM2.

Of course, EULSIM is applied to the first order equivalent system (with $v = u'$). Three versions of DIFEXM are compared:

D-B: Code DIFEXM due to Bauer [1], harmonic refinement sequence $\mathcal{F}_H$,

D-N0: new DIFEXM version (this paper), double harmonic sequence $\mathcal{F}_{2H}$, no final step,

D-N1: as above, but with final step (1.11).

All computations were done in FORTRAN double precision on the Siemens 7.865 of the Konrad Zuse Center, Berlin. The following characteristic numbers will be used:

NF - number of function evaluations (f, D)

TOL - user prescribed local error tolerance

ERR - actually obtained global error at T (using same scaling as internally)

The "exact" reference solution needed to determine ERR was computed by EUSIM2 and DIFEXM with TOL=10^{-12} and TOL=10^{-13}.

Figure 1.a presents a comparison NF over ERR of the 5 different codes (E1, E2, D-B, D-N0, D-N1) over a range of prescribed local error tolerances from TOL=1.D-4 to TOL=1.D-10 for test problem (3.1) with $\alpha = 10^2$. Among the DIFEXM codes, the old version D-B is clearly superceded by the two new versions: obviously, the restriction to even refinement sequence $\mathcal{F}_{2H}$ pays off. The effect of the final step (1.11), however, is less marked but visible as an increased global accuracy. Among the semi-implicit Euler codes, E2 is fastest and more accurate. The parameter $\alpha = 10^2$ has been selected for presentation, since at this value the code D-B still performed in a comparable way (compare also Figure 2).

For $\alpha = 10^4$, see Figure 1.b, the code D-B failed to solve the problem within any comparable scale of NF and over all TOL. For this parameter, E2 is the clear winner over all tolerances. For low tolerances, all other codes fail to supply the correct solution and show a rather irregular NF/ERR pattern - which is even inverse to the expected pattern in the DIFEXM codes. For high tolerances, say TOL=1.D-10, the computing times of the satisfactory runs are

D-N1: 3 sec E2: 4 sec E1: 8 sec

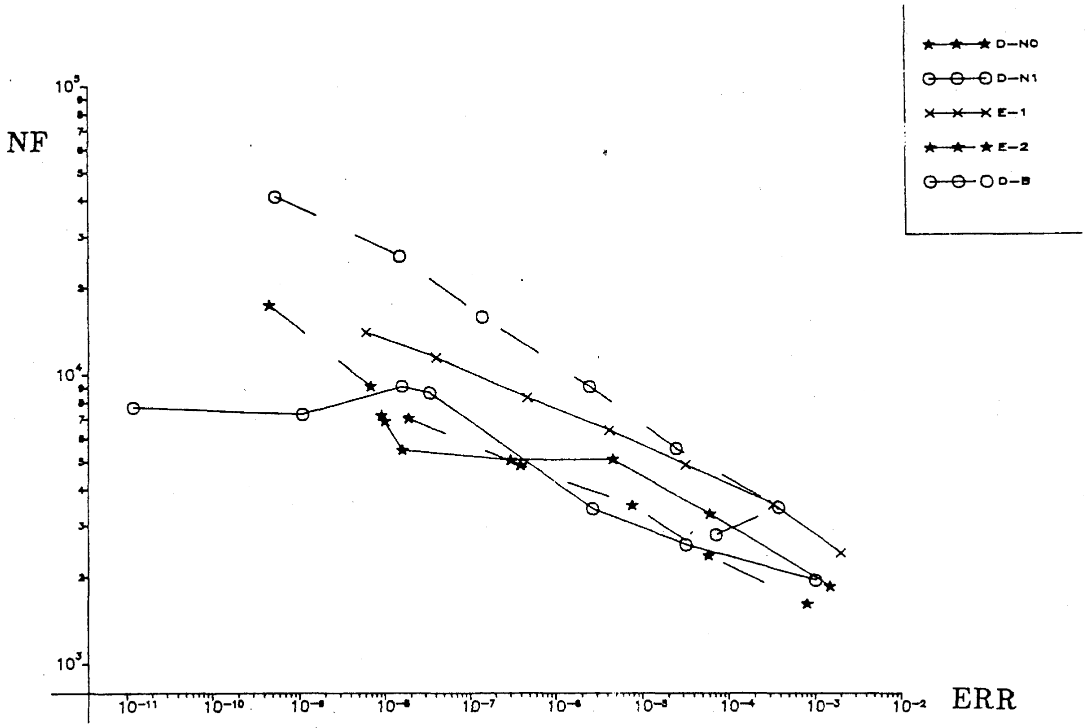

Figure 1.a Example (3.1) with $\alpha = 10^2$

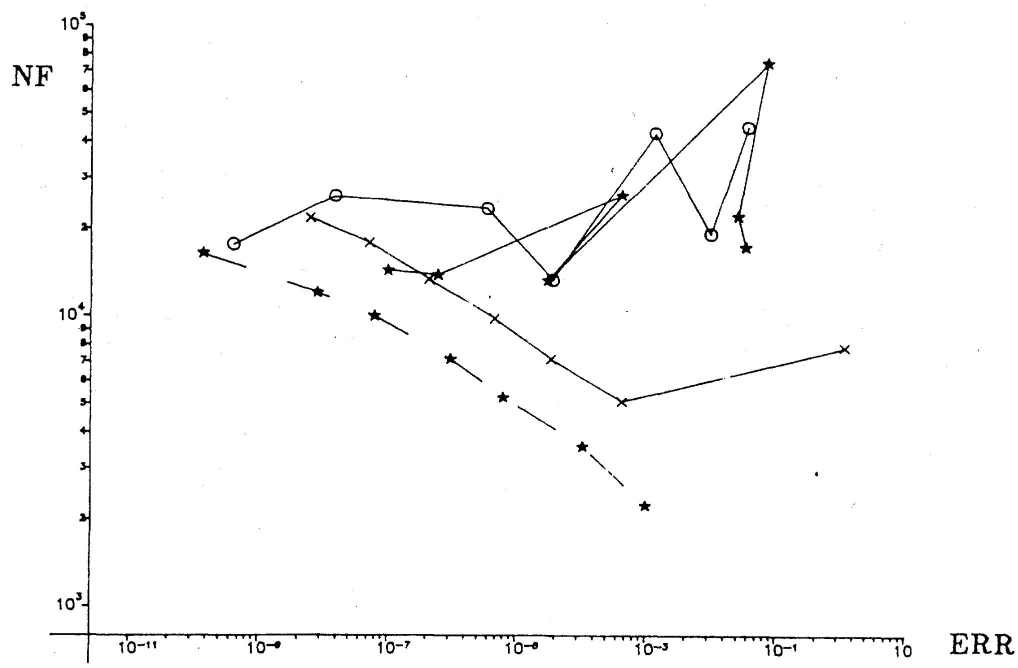

Figure 1.b Example (3.1) with $\alpha = 10^4$

The following three Figures vary α and keep TOL. Runs with ERR>TOL*100 are regarded as inaccurate and therefore omitted throughout Fig. 2. For low precision (Figure 2.a, TOL=1.D-4), the new code E2 is the clear winner, both in terms of speed and of accuracy (E2 produces ERR≤TOL*10). This situation is typical roughly up to precision TOL=1.D-7 (Figure 2.b). For very high precision (Figure 2.c, TOL=1.D-10) the new code D-N1 is the winner, since E2 spends too much effort in the mildly stiff case (lower values of α). This effect indicates the existence of a perturbed asymptotic expansion for discretization (2.3) – similar as shown for the semi-implicit Euler method in [7]. Finally, in Figure 3.a,b, a graph of the solution (Figure 3.a: u, Figure 3.b: u') for $\alpha = 10^4$ is presented, obtained from E2 with TOL=1.D-4. The both flexible and robust behavior of the code E2 is nicely illustrated by the sequence of automatically selected integration points in Figure 3.a.

Finally, note that in large scale problems the computing time will essentially depend on NF. For implicit systems, which are typical in real life applications, the extension of EUSIM2 will have an additional advantage: as can be seen from section 2, the linear algebra work of E2 is lower than for the extension of D-N1, the new DIFEXM version.

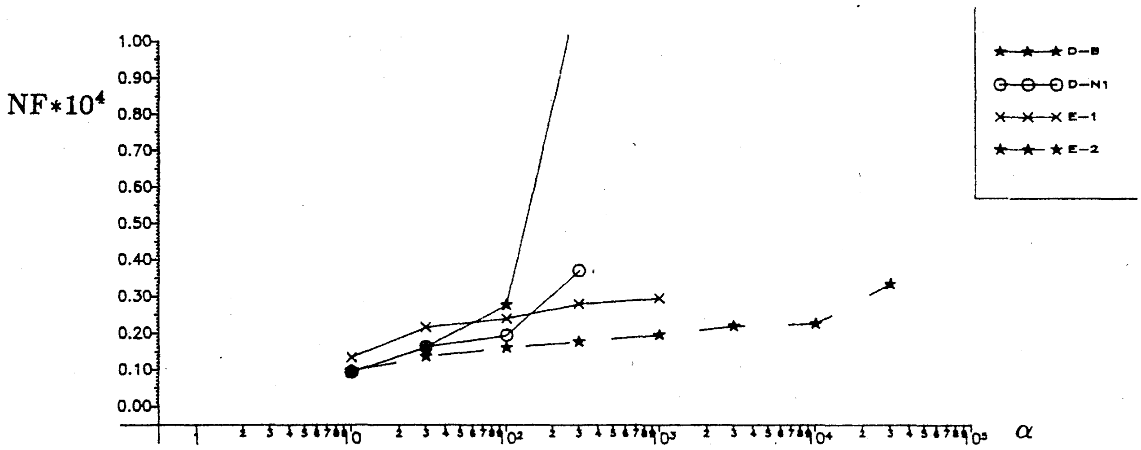

Figure 2.a Number of function evaluations (NF) for increasing α (TOL=10^{-4})

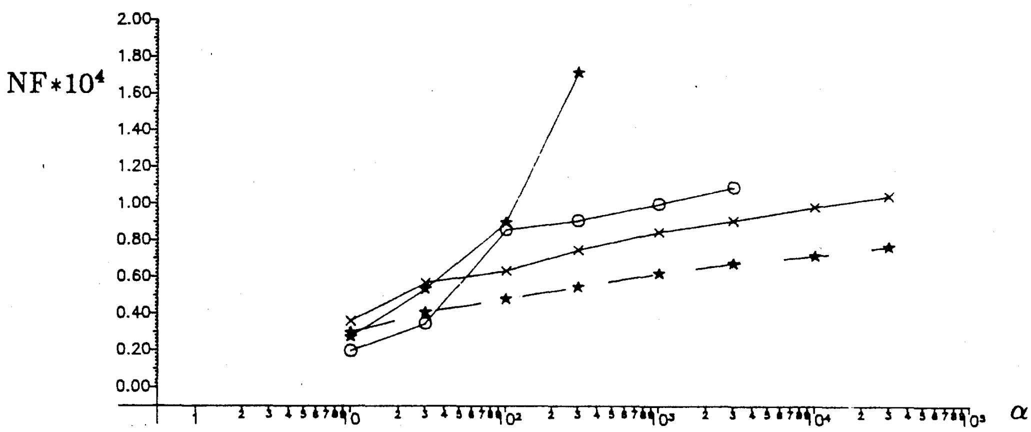

Figure 2.b Number of function evaluations (NF) for increasing α (TOL=10^{-7})

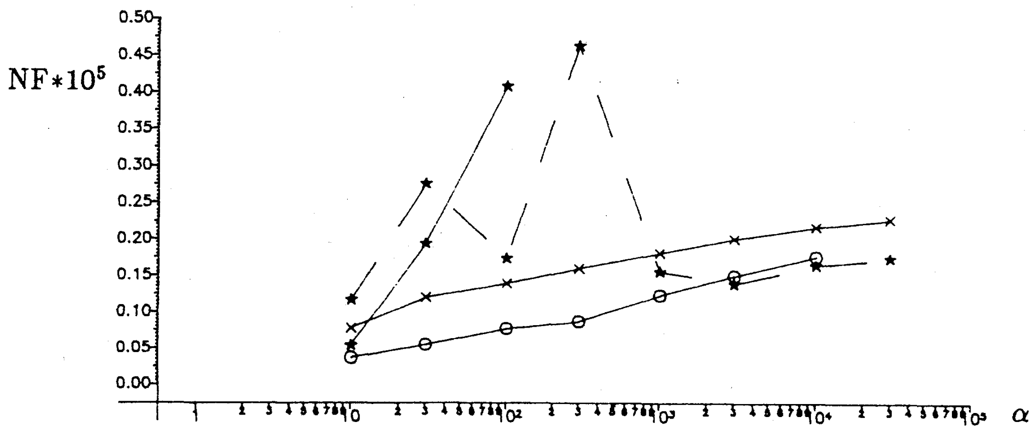

Figure 2.c Number of function evaluations (NF) for increasing α (TOL=10^{-10})

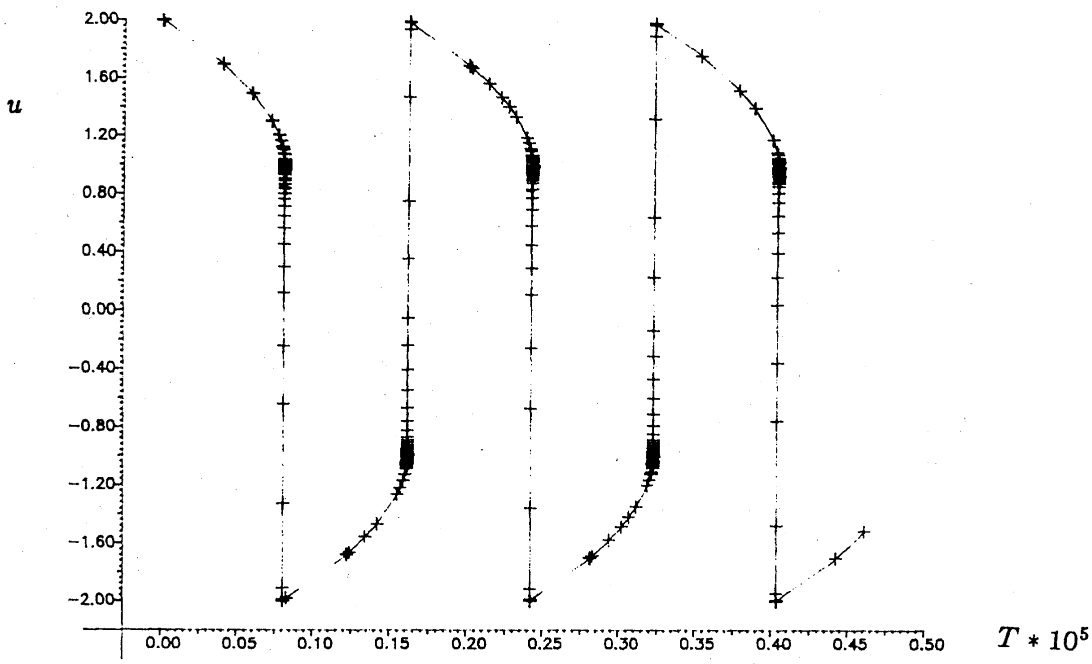

Figure 3.a Solution of example (3.1) for $\alpha = 10^4$ (integration points of EUSIM2, TOL=10^{-4})

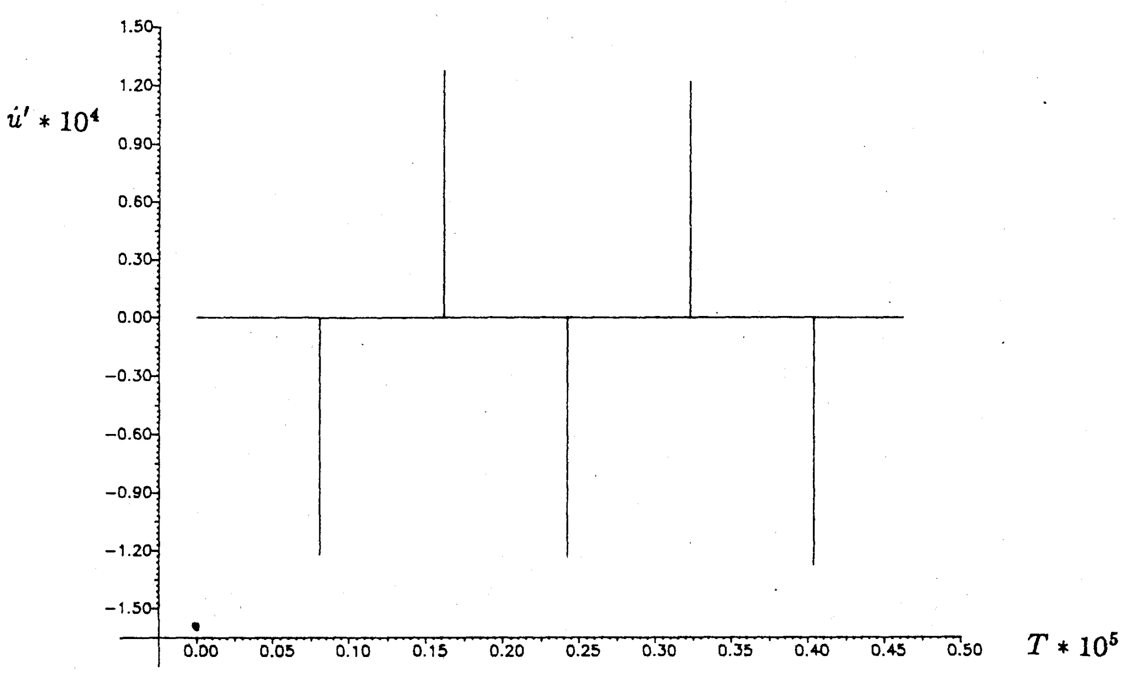

Figure 3.b Derivative of example (3.1) for $\alpha = 10^4$ (EUSIM2, TOL=10^{-4})

Conclusion

The above though restricted numerical experiments seem to indicate that the extended Stoermer discretization with h^2-extrapolation (as modified herein) may be efficient for non-stiff and mildly stiff non-selfadjoint second order ODE systems - with the rule of thumb "the stiffer the more precision needed". The special semi-implicit Euler method suggested herein seems to be the extrapolation method of choice for low precision computations (typical in engineering problems) or for stiff systems up to differential-algebraic second order systems. On the present basis numerical experiments with real life and large scale problems seem to be promising.

Acknowledgement. The author wishes to thank U. Pöhle for his extensive and efficient computational assistance.

References

[1] H.J. Bauer: *Entwicklung leistungsfähiger Extrapolationscodes.* Univ. Heidelberg, Inst. Angew. Math: Diplomarbeit (1983).

[2] G. Dahlquist: *A special stability problem for linear multistep methods.* BIT 3 (1963) 27-43.

[3] P. Deuflhard: *Recent Progress in Extrapolation Methods of Ordinary Differential Equations.* SIAM Rev. 27 (1985) 505-535.

[4] P. Deuflhard: *Uniqueness Theorems for Stiff ODE Initial Value Problems.* Konrad Zuse Center, Berlin: Preprint SC-87-3 (1987).

[5] G.H. Golub, Ch.F. v. Loan: *Matrix Computation.* The Johns Hopkins University Press. Baltimore, USA (1983).

[6] W.B. Gragg: *On extrapolation algorithms for ordinary initial value problems.* SIAM J. Numer. Anal. 2 (1965) 384-404.

[7] E. Hairer, Ch. Lubich: *Extrapolation at stiff differential equations.* Univ. Genève, Dept. mathématiques, Tech. Rep. (Feb. 1987).

[8] E. Hairer, S.P. Nørsett, G. Wanner: *Solving Ordinary Differential Equations I. Nonstiff Problems.* Springer Series in Computational Mathematics Vol. 8 (1987).