Numer. Math. 58, 419–439 (1990)

Numerische
Mathematik

Dense output for extrapolation methods*

E. Hairer[1] **and A. Ostermann**[2]

[1] Université de Genève, Section de Mathématiques, 2–4 rue du Lièvre, CH-1211 Genève 24, Switzerland

[2] Universität Innsbruck, Institut für Mathematik und Geometrie, Technikerstrasse 13, A-6020 Innsbruck, Austria

Received September 6, 1989 / July 2, 1990

Summary. This paper is concerned with dense output formulas for extrapolation methods for ordinary differential equations. In particular, the extrapolated explicit Euler method, the GBS method (for non-stiff equations) and the extrapolated linearly implicit Euler method (for stiff and differential-algebraic equations) are considered. Existence and uniqueness questions for dense output formulas are discussed and an algorithmic description for their construction is given. Several numerical experiments illustrate the theoretical results.

Subject classifications AMS(MOS): 65L05; CR: G1.7.

1 Introduction

> "This is the weak point of extrapolation methods in general ... present extrapolation methods do not have a satisfactory *interpolation* procedure yet ..."
> (P. Deuflhard 1985)

For non-stiff and stiff differential equations, extrapolation methods are important means for getting an accurate numerical solution. The most popular extrapolation method for non-stiff problems is the Gragg-Bulirsch-Stoer method (GBS method) which is based on the explicit midpoint rule (Sect. 5). We also consider extrapolation of the explicit Euler method (Sect. 2), which is very simple and yet gives satisfactory results. For stiff problems one can use either fully implicit schemes (implicit Euler, trapezoidal rule with smoothing) or linearly implicit methods, such as those based on the linearly implicit Euler method (Sect. 7) or on the linearly implicit midpoint rule.

All extrapolation methods have in common that they use very large step sizes during the integration. To illustrate this, we have applied two extrapolation codes to an Arenstorf orbit of the well-known restricted three body problem. The equations and the initial values are given in [14, p. 127]. Figure 1 presents the discrete points of the numerical solution together with the continuous output

* This work has been supported by the "Fonds national suisse de la recherche scientifique"

Offprint requests to: E. Hairer

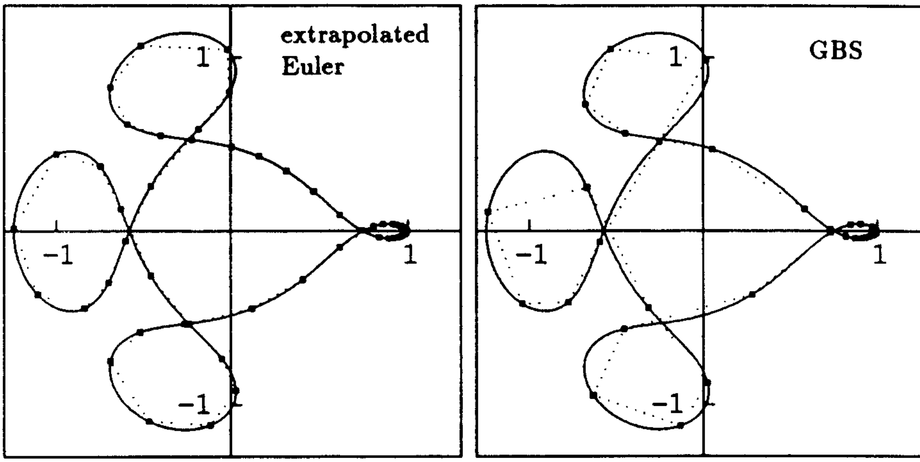

Fig. 1. Dense output solution for restricted three body problem

described in this article. The user supplied error tolerance was chosen $TOL = 10^{-6}$. The pictures nicely demonstrate the necessity of dense output formulas for extrapolation methods.

A continuous output is not only useful for graphical representation of the solution, but it may also be an important tool for delay differential equations [8], [14, Sect. II.5], and for event location problems [24].

The first attempt to get a dense output for extrapolation methods is due to Lindberg [18]. His approach, which is different from ours, imposes severe restrictions on the step number sequence. Shampine et al. [23] provided the GBS method with a 3rd order dense output formula.

We first investigate dense output formulas for non-stiff extrapolation integrators, namely the extrapolated explicit Euler method (Sects. 2, 3, and 4) and the GBS method (Sects. 5 and 6). Thereby we pursue two directions. On the one side we study the order conditions of very general interpolation procedures and their solution. This leads to existence and uniqueness results (Sects. 3 and 5) and provides insight into possible restrictions on the step size sequences. Independently from these investigations we present in Sects. 4 and 6 explicit dense output formulas which allow a simple implementation. Section 7 introduces a dense output formula for the linearly implicit Euler method, which is an attractive integrator for stiff differential equations. Finally, in Sect. 8 we study the order of this dense output, when the method is applied to a semi-explicit differential-algebraic system of index 1. Several numerical experiments illustrate the theoretical results.

2 Extrapolation of the explicit Euler method

We consider the system of differential equations

$$y' = f(x, y), \qquad y(x_0) = y_0, \tag{2.1}$$

where $f(x, y)$ and hence also the solution $y(x)$ is assumed to be sufficiently differentiable. The explicit Euler method applied to (2.1) reads

$$y_{i+1} = y_i + h f(x_i, y_i), \qquad i \geqq 0 \tag{2.2}$$

where h is the step size and $x_i = x_0 + ih$. We denote $y_h(x) = y_n$ for $x = x_0 + nh$, choose a sequence of integers $n_1 < n_2 < n_3 < \dots$ and define the corresponding step sizes $h_1 > h_2 > h_3 > \dots$ by $h_j = H/n_j$, where $H > 0$ is the basic step size. The h-extrapolation tableau is given by the formulas

$$Y_{j1} = y_{h_j}(x_0 + H) \tag{2.3a}$$

$$Y_{j,k+1} = Y_{j,k} + \frac{Y_{j,k} - Y_{j-1,k}}{(n_j/n_{j-k}) - 1}. \tag{2.3b}$$

Since the error $y_h(x_0 + H) - y(x_0 + H)$ has an asymptotic h-expansion, each extrapolation eliminates one power of h, so that

$$Y_{jk} - y(x_0 + H) = \mathbf{O}(H^{k+1}) \tag{2.4}$$

(see [14, Sects. II.8 and II.9]).

The subsequent analysis relies on the fact that each Y_{jk} can be interpreted as the numerical result of a Runge-Kutta scheme of order k. This can be seen as follows: for $h = H/n$ the numerical solution of the Euler method at $x_0 + H$ is given by

$$y_h(x_0 + H) = y_0 + H \sum_{i=1}^{n} b_i K_i$$

$$K_i = f\left(x_0 + c_i H, y_0 + H \sum_{j=1}^{i-1} a_{ij} K_j\right), \qquad i = 1, \dots, n \tag{2.5}$$

with coefficients

$$c_i = \frac{i-1}{n}, \qquad a_{ij} = \frac{1}{n}, \qquad b_i = \frac{1}{n}. \tag{2.6}$$

Formula (2.5) shows that for any j, Y_{j1} is the result of an explicit n_j-stage Runge-Kutta method of order 1. Since Y_{jk} is a linear combination of $Y_{j-k+1,1}, \dots, Y_{j1}$, it can also be written as an explicit Runge-Kutta method. By (2.4) its order is k.

The above algorithm yields a very accurate approximation to the solution at $x_0 + H$. In order to get a cheap continuous output we consider for $0 \leqq \theta \leqq 1$

$$Y_{j1}(\theta) = y_0 + H \sum_{i=1}^{n_j} b_i(\theta, n_j) K_i \tag{2.7}$$

with K_j given by (2.5). Our aim is to construct polynomials $b_i(\theta, n)$ in such a way that $b_i(0, n) = 0$, $b_i(1, n) = 1/n$ (this means $Y_{j1}(0) = y_0$, $Y_{j1}(1) = Y_{j1}$) and that the values $Y_{jk}(\theta)$, obtained by the recursion (2.3b) satisfy

$$Y_{jk}(\theta) - y(x_0 + \theta H) = \mathbf{O}(H^p) \qquad \text{for } 0 \leqq \theta \leqq 1 \tag{2.8}$$

with p as high as possible. This approach does not use additional function evaluations and does not change the Y_{jk}. Therefore the global behaviour of the method is not influenced. Dense output formulas of the type (2.7) are frequently used for explicit Runge-Kutta methods. From the vast literature on this topic we mention the articles [3, 7, 9, 14, 17, 20, 22, 26].

3 Existence and uniqueness of dense output for explicit Euler

Because of (2.4) the best possible p in (2.8) is $p=k+1$. The following theorem gives necessary and sufficient conditions to obtain this order for the dense output.

Theorem 1. *Consider a finite sequence $n_1<n_2<\ldots<n_\kappa$ and let $y(x)$ be the solution of* (2.1). *Then $Y_{jk}(\theta)$, given by* (2.7) *and the recursion* (2.3b), *satisfy*

$$Y_{jk}(\theta)-y(x_0+\theta H)=\mathbf{O}(H^{k+1}) \quad \text{for } 1\leqq k\leqq j\leqq \kappa \tag{3.1}$$

if and only if there exist functions $e_{21}(\theta)$; $e_{31}(\theta)$, $e_{32}(\theta)$; …; $e_{\kappa 1}(\theta), \ldots, e_{\kappa,\kappa-1}(\theta)$ such that

$$\sum_{i=1}^{n} b_i(\theta, n)\, c_i^{q-1}=\frac{\theta^q}{q}+\sum_{l=1}^{q-1} e_{ql}(\theta)\left(\frac{1}{n}\right)^l \tag{3.2}$$

for $q=1, \ldots, \kappa$ and $n\in\{n_1, \ldots, n_\kappa\}$ ($c_i=(i-1)/n$ as in (2.6)).

Before we give the proof of this theorem we discuss the solvability of the linear system (3.2). A solution can be found recursively as follows:

In a first step we choose $b_i(\theta, n_1)$ for $i=1, \ldots, n_1$ such that (3.2) holds with $q=1$ and $n=n_1$. In the general step we have to determine simultaneously $b_i(\theta, n_j)$ for $i=1, \ldots, n_j$ and $e_{j1}(\theta), \ldots, e_{j,j-1}(\theta)$ from the Eq. (3.2) with $q\in\{1, \ldots, j\}$, $n=n_j$ and $q=j$, $n\in\{n_1, \ldots, n_{j-1}\}$. These are $2j-1$ conditions for n_j+j-1 unknowns. Since $n_j\geqq j$ and the matrix of this linear system is Vandermonde-like, a solution always exists. Further we have uniqueness for $n_j=j$. Thus we have proven the following theorem:

Corollary 2. *For every step number sequence $\{n_1, n_2, \ldots, n_\kappa\}$ there exists a dense output formula* (2.7) *such that the extrapolated values satisfy* (3.1).
For the harmonic sequence $\{1, 2, 3, \ldots, \kappa\}$ this dense output formula is unique. □

For the proof of Theorem 1 we shall use the following lemma, which generalizes the idea of "simplifying assumptions" (see [14, p. 203]).

Lemma 3. *For the Runge-Kutta coefficients* (2.6) *there exist constants ε_{ql} such that*

$$\sum_{j=1}^{i-1} a_{ij}\, c_j^{q-1}=\frac{c_i^q}{q}+\sum_{l=1}^{q-1} \varepsilon_{ql}\, c_i^{q-l}\left(\frac{1}{n}\right)^l \quad \text{for all } q\geqq 1 \text{ and } i=1, \ldots, n. \tag{3.3}$$

Proof. We apply the Euler method (2.2) with stepsize $h=1/n$ to the problem $y'=x^{q-1}$, $y(0)=0$ whose exact solution is $y(x)=x^q/q$. An easy calculation shows that the numerical solution at $x=(i-1)\,h=c_i$ is

$$y_h(x)=\sum_{j=1}^{i-1} a_{ij}\, c_j^{q-1} \tag{3.4}$$

with a_{ij} given by (2.6). In order to prove (3.3) we make use of the asymptotic expansion

$$y_h(x)=y(x)+d_1(x)\,h+d_2(x)\,h^2+\dots, \tag{3.5}$$

where $d_l(0)=0$. Inserting (3.5) into (2.2), i.e. $y_h(x+h)=y_h(x)+h x^{q-1}$, and comparing powers of h, yields $d_l(x)=\varepsilon_{ql}\,x^{q-l}$ for $l<q$ and $d_l(x)=0$ for $l\geqq q$. Formula (3.5) with $x=c_i$ together with (3.4) proves the statement of the lemma. □

Proof of Theorem 1. Sufficiency. In order to prove (3.1) we consider $Y_{jk}(\theta)$ and $y(x_0+\theta H)$ as functions of H, expand them into Taylor series and verify that the first terms are equal. This can be done with the use of rooted trees, elementary differentials, etc., as explained in [14, Sect. II.2].

The q-th derivative of the exact solution $y(x_0+\theta H)$ satisfies (Theorem II.2.6 of [14])

$$\frac{d^q}{dH^q}\,y(x_0+\theta H)|_{H=0}=\sum_{t\in LT_q}\theta^q F(t)(x_0,y_0), \tag{3.6}$$

where $F(t)(x_0,y_0)$ is an expression composed of derivatives of $f(x,y)$ (elementary differentials). LT_q is the set of labelled trees with q vertices. Similarly, for the numerical solution $Y_{j1}(\theta)$ of (2.7) we have (Theorem II.2.11 of [14])

$$\frac{d^q}{dH^q}\,Y_{j1}(\theta)|_{H=0}=\sum_{t\in LT_q}\gamma(t)\sum_{i=1}^{n_j}b_i(\theta,n_j)\,\Phi_i(t)\,F(t)(x_0,y_0), \tag{3.7}$$

where $\Phi_i(t)$ depend on the coefficients (2.6) and are recursively defined by

$$\begin{aligned}&\Phi_i(\tau)=1\\ &\Phi_i(t)=\sum_{k_1,\dots,k_m}a_{ik_1}\dots a_{ik_m}\,\Phi_{k_1}(t_1)\dots\Phi_{k_m}(t_m)\qquad\text{if } t=[t_1,\dots,t_m].\end{aligned} \tag{3.8}$$

The number $\gamma(t)$ is given by $\gamma(\tau)=1$, $\gamma(t)=\rho(t)\;\gamma(t_1)\dots\gamma(t_m)$, and $\rho(t)$ denotes the number of vertices of the tree t. Suppose for the moment that condition (3.2) implies

$$\sum_{i=1}^{n}b_i(\theta,n)\,\Phi_i(t)=\frac{\theta^{\rho(t)}}{\gamma(t)}+\sum_{l=1}^{\rho(t)-1}e_l(\theta,t)\left(\frac{1}{n}\right)^l \tag{3.9}$$

for $n\in\{n_1,\dots,n_\kappa\}$ and trees t with $\rho(t)\leqq\kappa$. Then we can insert (3.9) into (3.7) and extrapolate $(k-1)$-times the q-th derivatives of $Y_{j-k+1,1}(\theta),\dots,Y_{j1}(\theta)$. This yields $\sum_{t\in LT_q}\theta^q F(t)(x_0,y_0)$ for $q\leqq k$, because the terms $e_l(\theta,t)\,(1/n)^l$ are eliminated for $l\leqq k-1$ by extrapolation. A comparison with (3.6) shows that the first k derivatives of $Y_{jk}(\theta)-y(x_0+\theta H)$ vanish. This proves (3.1).

We still have to justify formula (3.9). For this we shall prove by induction on $\rho(t)$ that

$$\Phi_i(t)=\frac{\rho(t)}{\gamma(t)}\,c_i^{\rho(t)-1}+\sum_{l=1}^{\rho(t)-2}\varepsilon_l(t)\,c_i^{\rho(t)-l-1}\left(\frac{1}{n}\right)^l. \tag{3.10}$$

By Lemma 3 this identity then implies

$$\sum_{j=1}^{i-1} a_{ij}\,\Phi_j(t)=\frac{c_i^{\rho(t)}}{\gamma(t)}+\sum_{l=1}^{\rho(t)-1}\hat{\varepsilon}_l(t)\,c_i^{\rho(t)-l}\left(\frac{1}{n}\right)^l. \tag{3.11}$$

Formula (3.10) is obviously true for $\rho(t)=1$ (i.e. $t=\tau$). Consider now a tree $t=[t_1, \ldots, t_m]$ and suppose that (3.10) and hence also (3.11) is valid for $t_1, \ldots, t_m$. Inserting (3.11) for $t_1, \ldots, t_m$ into (3.8) and using again Lemma 3 yields (3.10) for the tree t.

Next we multiply Formula (3.10) by $b_i(\theta, n)$ and sum up from $i=1$ to $i=n$. Substituting each occurrence of $\sum_i b_i(\theta, n)\,c_i^{q-1}$ by our assumption (3.2) we arrive at (3.9).

Necessity. Assume that (3.1) holds. Application of the Euler method with $H=1$ to $y'=x^{q-1}$, $y(0)=0$ yields by (2.7)

$$Y_{j1}(\theta)=\sum_{i=1}^{n_j} b_i(\theta, n_j)\,c_i^{q-1}. \tag{3.12}$$

Since every q-th order method integrates the above equation exactly, the extrapolated value $Y_{jq}(\theta)$ equals the exact solution at θ, i.e.

$$Y_{jq}(\theta)=\frac{\theta^q}{q} \quad \text{for all } j. \tag{3.13}$$

By definition of the extrapolation method, $Y_{jq}(\theta)$ can be interpreted as $Y_{jq}(\theta)=p_j(0)$ where $p_j(h)$ is the polynomial of degree $q-1$ which satisfies

$$p_j\left(\frac{1}{n_i}\right)=Y_{i1}(\theta) \quad \text{for } i=j-q+1, \ldots, j. \tag{3.14}$$

The coefficients of $p_j(h)$ seem to depend on j. However, by (3.13) the value $p_j(0)$ does not depent on j, so that two consecutive polynomials take the same values at q distinct points at least. Thus they are equal. This implies the existence of one polynomial

$$p(h)=\frac{\theta^q}{q}+\sum_{l=0}^{q-1} e_{ql}(\theta)\,h^l$$

which satisfies $Y_{j1}(\theta)=p(1/n_j)$ for all j. This, together with (3.12) and (3.14) proves the statement (3.2). □

4 Construction of dense output formulas for explicit Euler

Theorem 1 provides in principle a means for obtaining dense output formulas: one has to solve the linear system (3.2) and extrapolate the values $Y_{j1}(\theta)$ to obtain $Y_{jk}(\theta)$. If this is done for the harmonic sequence $\{1, 2, 3, 4, \ldots\}$, one obtains surprisingly simple formulas from apparently complicated equations. This suggests that an elegant idea should be able to lead to the same results.

This idea, which has been pointed out to the authors by Ch. Lubich, is the following: together with each Y_{j1} we compute sufficiently many finite differences at both ends of the integration interval, then we extrapolate them and use the so-obtained approximations to the derivatives for Hermite interpolation. A similar idea is used by Deuflhard and Nowak [6] to construct consistent initial values for differential-algebraic problems.

Suppose that the value $Y_{\kappa\kappa}$, obtained with the step size sequence $h_1 > h_2 > \ldots > h_\kappa$ has been accepted as numerical approximation to $y(x_0+H)$. The dense output formula is then given by the following algorithm:

Step 1. For each $j \in \{1, \ldots, \kappa\}$ we compute the approximations to the derivatives of $y(x)$ at the left and right endpoints of the interval $[x_0, x_0+H]$:

$$l_j^{(k)} = \frac{\Delta^{k-1} f_0^{(j)}}{h_j^{k-1}}, \qquad r_j^{(k)} = \frac{\nabla^{k-1} f_{n_j-1}^{(j)}}{h_j^{k-1}} \qquad \text{for } k=1, \ldots, j. \tag{4.1}$$

Here $f_i^{(j)} = f(x_0 + i h_j, y_i)$ is the function value, evaluated during the computation of Y_{j1}, and $\Delta f_i = f_{i+1} - f_i$, $\nabla f_i = f_i - f_{i-1}$ are the forward and backward differences. No additional function evaluation is necessary.

Step 2. In order to improve the accuracy of these approximations, we extrapolate the values $l_k^{(k)}, \ldots, l_\kappa^{(k)}$ $(\kappa - k)$-times to obtain $l^{(k)}$. More precisely, we put $T_{j1} = l_j^{(k)}$ for $j = k, \ldots, \kappa$, extrapolate these values according to (2.3b) and define $l^{(k)}$ by $l^{(k)} = T_{\kappa, \kappa-k+1}$. In exactly the same way we also compute $r^{(k)}$.

Step 3. For given λ and ρ $(-1 \leqq \lambda, \rho \leqq \kappa)$ we define the polynomial $P_{\lambda\rho}(\theta)$ of degree $\lambda + \rho + 1$ by

$$\begin{aligned} &P_{\lambda\rho}(0) = y_0 && \text{if } \lambda \geqq 0 \\ &P_{\lambda\rho}(1) = Y_{\kappa\kappa} && \text{if } \rho \geqq 0 \\ &P_{\lambda\rho}^{(k)}(0) = H^k l^{(k)} && \text{for } k = 1, \ldots, \lambda \\ &P_{\lambda\rho}^{(k)}(1) = H^k r^{(k)} && \text{for } k = 1, \ldots, \rho. \end{aligned} \tag{4.2}$$

The following theorem shows to which order these polynomials approximate the exact solution.

Theorem 4. *If $\lambda + \rho \geqq \kappa - 1$, then the error of the interpolation polynomial $P_{\lambda\rho}(\theta)$ satisfies*

$$P_{\lambda\rho}(\theta) - y(x_0 + \theta H) = \mathbf{O}(H^{\kappa+1}). \tag{4.3}$$

Proof. Since $P_{\lambda\rho}(\theta)$ is a polynomial of degree $\lambda + \rho + 1 \geqq \kappa$, the error due to interpolation is of size $\mathbf{O}(H^{\kappa+1})$. By (4.2) it therefore suffices to prove that the values $l^{(k)}$ and $r^{(k)}$ satisfy

$$\begin{aligned} &l^{(k)} = y^{(k)}(x_0) + \mathbf{O}(H^{\kappa-k+1}) && \text{for } k=1, \ldots, \lambda \\ &r^{(k)} = y^{(k)}(x_0+H) + \mathbf{O}(H^{\kappa-k+1}) && \text{for } k=1, \ldots, \rho. \end{aligned} \tag{4.4}$$

The approximations $l_j^{(k)}$ and $r_j^{(k)}$, defined in (4.1) have an asymptotic expansion of the form

$$l_j^{(k)} = y^{(k)}(x_0) + h_j a_1^{(k)} + h_j^2 a_2^{(k)} + \dots \tag{4.5}$$

$$r_j^{(k)} = y^{(k)}(x_0 + H) + h_j b_1^{(k)} + h_j^2 b_2^{(k)} + \dots . \tag{4.6}$$

Formula (4.5) follows immediately from Taylor series expansion, while Formula (4.6) needs also the asymptotic expansion of the global error of the Euler method. Since each extrapolation eliminates one power of h, this proves (4.4). □

The uniqueness result of Corollary 2 allows us to prove the following theorem.

Theorem 5. *For the harmonic sequence* $\{1, 2, \dots, \kappa\}$ *the polynomials* $P_{\lambda\rho}(\theta)$ *are identical as long as* $\lambda + \rho \geqq \kappa - 1$.

Proof. The proof relies on the fact that each interpolation polynomial $P_{\lambda\rho}(\theta)$ can be interpreted as the extrapolated value $Y_{\kappa\kappa}(\theta)$ of functions $Y_{j1}(\theta)$, which are of the form (2.7). This follows from the commutativity of the diagram

$$\begin{tikzcd}[column sep=huge, row sep=large]
y_0,\, Y_{j1},\, l_j^{(k)},\, r_j^{(k)} \arrow[r, "\text{Step 2}"] \arrow[d] & y_0,\, Y_{\kappa\kappa},\, l^{(k)},\, r^{(k)} \arrow[d, "\text{Step 3}"] \\
Y_{j1}(\theta) \arrow[r, "\text{extrapolation } \kappa \text{ times}"] & Y_{\kappa\kappa}(\theta) = P_{\lambda\rho}(\theta).
\end{tikzcd} \tag{4.7}$$

We still have to define $Y_{j1}(\theta)$. Recall that $l^{(k)} = p_l(0)$, where $p_l(h)$ is a polynomial of degree $\kappa - k$ which is defined by

$$p_l(H/n_j) = l_j^{(k)} \qquad \text{for } j = k, \dots, \kappa.$$

If we put

$$q_l(h) = p_l(h) \cdot \left(1 - \frac{h n_1}{H}\right) \dots \left(1 - \frac{h n_{k-1}}{H}\right)$$

then extrapolation of the values $q_l(H/n_j)$, $j = 1, \dots, \kappa$, again yields $l^{(k)} = q_l(0)$. We therefore set $Y_{j1}(\theta)$ as the polynomial of degree $\lambda + \rho + 1$ defined by (4.2) with $Y_{\kappa\kappa}$, $l^{(k)}$, $r^{(k)}$ replaced by Y_{j1}, $q_l(H/n_j)$, $q_r(H/n_j)$. Obviously, $q_r(h)$ and $p_r(h)$ denote the corresponding polynomials for the $r_j^{(k)}$. The statement is now an immediate consequence of Corollary 2. □

Remark. If λ and ρ are non-negative then, by the first two conditions of (4.2), this dense output gives a globally continuous approximation to the solution. A natural choice would be $\lambda = [\kappa/2]$ and $\rho = [(\kappa - 1)/2]$. A solution which is globally C^1 can easily be obtained for $\lambda \geqq 1$, $\rho \geqq 1$ if we replace $r^{(1)}$ in (4.2) by $f(x_0 + H, Y_{\kappa\kappa})$.

Numerical example. We modified the extrapolation code ODEX from [14] so that it is now based on the explicit Euler method, and equipped it with the above described dense output formulas. We call this new code ELEX. A first illustration of the performance of this code are the results already shown in Fig. 1. For a thorough study of the obtained errors of the dense output, we

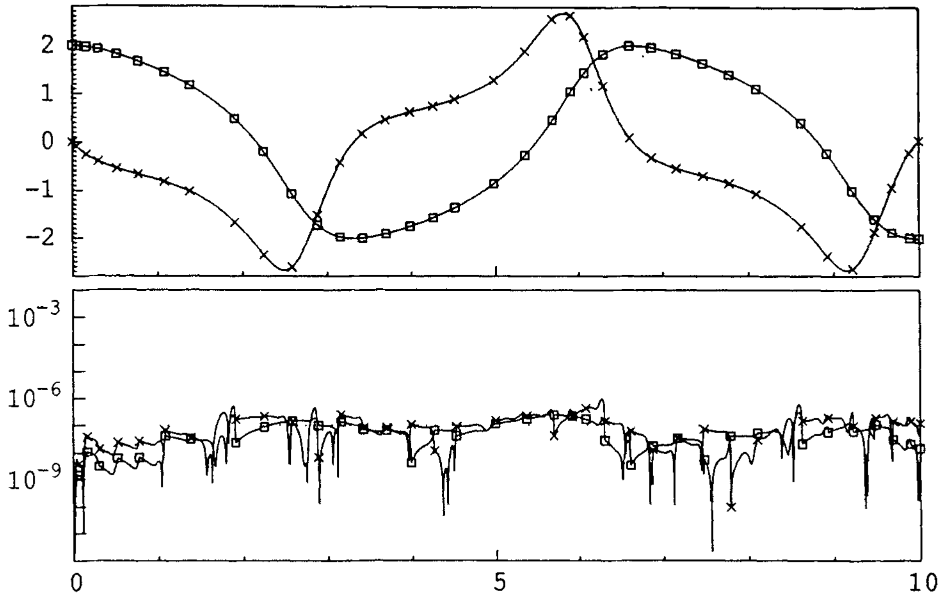

Fig. 2. Solution and error of extrapolated explicit Euler method

applied ELEX with step number sequence $\{1, 2, 3, 4, \ldots\}$ and with $\mathrm{TOL}=10^{-7}$ to the non-stiff Van der Pol equation

$$\begin{aligned} y_1' &= y_2, & y_1(0)&=2 \\ y_2' &= (1-y_1^2)\,y_2 - y_1, & y_2(0)&=0. \end{aligned} \tag{4.8}$$

The upper picture of Fig. 2 shows the dense output solution together with the natural output points. The global error of this solution is plotted in the lower picture.

5 Dense output for GBS method

The well known GBS method is defined by ($x = x_0 + nh$, with n even)

$$y_1 = y_0 + h f(x_0, y_0) \tag{5.1 a}$$

$$y_{i+1} = y_{i-1} + 2h f(x_i, y_i), \qquad i = 1, 2, \ldots, n \tag{5.1 b}$$

$$S_h(x) = \tfrac{1}{4}(y_{n-1} + 2y_n + y_{n+1}). \tag{5.1 c}$$

The value $S_h(x)$ is just the result of one Euler step (5.1 a), followed by n steps of the explicit midpoint rule (5.1 b), and by a smoothing step (5.1 c). For a sequence of even integers $n_1 < n_2 < \ldots$ we again denote the step sizes by $h_j = H/n_j$.

Since the error $S_h(x_0+H)-y(x_0+H)$ has an asymptotic h^2-expansion, the extrapolation tableau becomes

$$Y_{j1}=S_{h_j}(x_0+H) \tag{5.2a}$$

$$Y_{j,k+1}=Y_{j,k}+\frac{Y_{j,k}-Y_{j-1,k}}{(n_j/n_{j-k})^2-1} \tag{5.2b}$$

(see [25, Sect. 7.2.14] and [14, Sect. II.9]). As in Sect. 2 the value Y_{j1} is seen to be the result of an explicit (n_j+1)-stage Runge-Kutta method (2.5), whose coefficients are given by $(n=n_j)$

$$c_i=\frac{i-1}{n}, \qquad a_{ij}=\begin{cases}1/n & j=1, i \text{ even}\\ 2/n & 1<j<i, i+j \text{ odd}\\ 0 & \text{else}\end{cases} \tag{5.3}$$
$$b_i=\begin{cases}1/(2n) & i=1 \quad \text{or} \quad i=n+1\\ 1/n & 2\leqq i\leqq n\end{cases}$$

Each extrapolation in (5.2) eliminates two powers of h in the error. Therefore Y_{jk} represents a Runge-Kutta method of order $2k$. So as in Sect. 2 we are looking for a dense output formula

$$Y_{j1}(\theta)=y_0+H\sum_{i=1}^{n_j+1} b_i(\theta, n_j)\, K_i \qquad \text{for } 0\leqq\theta\leqq 1 \tag{5.4}$$

such that (2.8) is satisfied with $p=2k+1$.

With these preparations we are able to extend the results of Sect. 3 to the GBS method. Theorem 6 and Lemma 7 below are the analogues of Theorem 1 and Lemma 3, respectively. Since the proofs are similar, we omit details.

Theorem 6. *Consider a sequence $n_1<n_2<\ldots<n_\kappa$ of even integers. Then $Y_{jk}(\theta)$, given by* (5.4) *and* (5.2b), *satisfy*

$$Y_{jk}(\theta)-y(x_0+\theta H)=\mathbf{O}(H^{2k+1}) \quad \textit{for } 1\leqq k\leqq j\leqq\kappa \tag{5.5}$$

if and only if there exist functions $e_{ql}(\theta)$, $d_{ql}(\theta)$ such that for all $n\in\{n_1, \ldots, n_\kappa\}$

$$\sum_{i=1}^{n+1} b_i(\theta, n)\, c_i^{q-1}=\frac{\theta^q}{q}+\sum_{l=1}^{[(q-1)/2]} e_{ql}(\theta)\left(\frac{1}{n}\right)^{2l} \qquad \textit{for } q=1, \ldots, 2\kappa, \tag{5.6a}$$

$$\sum_{\substack{i=1\\ i \text{ even}}}^{n} b_i(\theta, n)\, c_i^{q-1}=\sum_{l=0}^{[(q-1)/2]} d_{ql}(\theta)\left(\frac{1}{n}\right)^{2l} \qquad \textit{for } q=1, \ldots, 2\kappa-2. \tag{5.6b}$$

The proof is based on the "simplifying assumptions":

Lemma 7. *For the Runge-Kutta coefficients* (5.3) *there exist constants* ε_{ql} *such that*

$$\sum_{j=1}^{i-1} a_{ij} c_j^{q-1} = \frac{c_i^q}{q} + \sum_{l=1}^{[(q-1)/2]} \varepsilon_{ql} c_i^{q-2l} \left(\frac{1}{n}\right)^{2l} \tag{5.7}$$
$$+ \begin{cases} \varepsilon_{q,q/2} \left(\frac{1}{n}\right)^q & \text{if } i \text{ even and } q \text{ even} \\ 0 & \text{else} \end{cases}$$

for all $q \geqq 1$ *and* $i = 1, \ldots, n+1$.

For the *proof* of Lemma 7 we apply the GBS method to $y' = x^{q-1}$, $y(0) = 0$ and make use of the asymptotic expansion of the global error (as in the proof of Lemma 3). Since this expansion is different at even and odd grid-points, it is not astonishing that we obtain two different formulas.

Theorem 6 differs from Theorem 1 mainly by the additional Formula (5.6b). The necessity of (5.6b) can be seen by applying the GBS method to

$$y_1' = x^{q-3} y_2, \qquad y_2' = x, \qquad y_1(0) = y_2(0) = 0.$$

In this case the numerical solution for y_1 is

$$Y_{j1}(\theta) = \sum_{i=1}^{n_j+1} b_i(\theta, n_j)\, c_i^{q-3} \sum_{m=1}^{i-1} a_{im} c_m$$

and has an expansion of the form (5.6a). Inserting (5.7) with $q = 2$ yields (5.6b).

Condition (5.7) allows to prove formulas similar to (3.10) and (3.11), whose coefficients depend on whether i is even or odd. Thus one needs both conditions in (5.6) to prove (5.5). □

We conclude this section with some remarks on the solvability of the linear system (5.6). Counting the number of conditions and the number of free parameters we get:

$$\text{number of conditions: } 4\kappa^2 - 2\kappa$$
$$\text{number of parameters: } n_1 + \ldots + n_\kappa + 2\kappa^2 - \kappa.$$

This suggests that the step number sequence should satisfy

$$n_1 + \ldots + n_\kappa \geqq 2\kappa^2 - \kappa \qquad \text{for all } \kappa \geqq 1. \tag{5.8}$$

For the standard sequence $\{2, 4, 6, 8, 10, \ldots\}$ condition (5.8) is satisfied only for $\kappa \leqq 2$. Indeed, the linear system (5.6) has a unique solution for $\kappa \leqq 2$, but is unsolvable for $\kappa > 2$ (even $\{2, 4, 8, \ldots\}$ leads to an unsolvable system (5.6) for $\kappa > 2$).

If the step number sequence satisfies $n_\kappa \geqq 4\kappa - 2$ and thus also (5.8), the system (5.6) is solvable for all $\kappa \geqq 1$ due to its special structure. This is seen by recursion, as steping from κ to $\kappa + 1$ yields more new parameters ($n_{\kappa+1} + 4\kappa$

$+1$) than new conditions ($8\kappa+2$). A natural step number sequence, which allows to construct dense output formulas satisfying (5.5) is thus

$$\{2, 6, 10, 14, 18, 22, \ldots\} \tag{5.9}$$

It is curious to see that a similar sequence has been found useful in the context of stiff extrapolation by Bader and Deuflhard [2].

Remark. (GBS method without smoothing). Instead of (5.2a) it is also possible to use $Y_{j1}=y_{h_j}(x_0+H)$, where $y_h(x)=y_n$ for $x=x_0+nh$. This saves one function evaluation in (5.1b) and thus reduces the number of free parameters to $n_1+\ldots+n_\kappa+2\kappa^2-2\kappa$. In this situation the sequence $\{2, 4, \ldots\}$ does not allow a dense output formula satisfying (5.5), even not for $\kappa=2$. This was already noticed by Shampine et al. [23]. The assumption $n_\kappa \geqq 4\kappa-2$ again implies the solvability of the system (5.6). Here the sequence (5.9) yields a unique dense output.

6 Construction of dense output formulas for GBS method

For the GBS method, the straightforward application of the ideas of Sect. 4 does not lead to satisfactory dense output formulas, as the one-sided differences (4.1) do not have an h^2-expansion (see (4.5), (4.6)). Therefore only about half of the order is achievable. We shall show here how to overcome this difficulty.

Suppose that the value $Y_{\kappa\kappa}$ has been computed with the step number sequence $\{n_1, \ldots, n_\kappa\}$. For the following construction we need that n_j is even and that the differences

$$n_{j+1}-n_j \quad \text{are multiples of 4,} \qquad j=1, 2, \ldots \tag{6.1}$$

a property, which is shared by (5.9).

Step 1. For each $j\in\{1, \ldots, \kappa\}$ we compute approximations to the derivatives of $y(x)$ at the midpoint $x_0+H/2$:

$$d_j^{(0)}=y_{n_j/2}^{(j)}, \qquad d_j^{(k)}=\frac{\delta^{k-1} f_{n_j/2}^{(j)}}{(2h_j)^{k-1}} \qquad \text{for } k=1, \ldots, 2j. \tag{6.2}$$

Again $f_i^{(j)}=f(x_0+ih_j, y_i^{(j)})$, and $y_i^{(j)}$ is the approximation obtained during the computation of Y_{j1}. Further, $\delta f_i=f_{i+1}-f_{i-1}$ denotes the central difference operator.

Step 2. We extrapolate $d_j^{(0)}$ $(\kappa-1)$ times and $d_j^{(2l-1)}$, $d_j^{(2l)}$ $(\kappa-l)$ times. This yields the improved approximations $d^{(k)}$.

Step 3. For given μ $(0\leqq\mu\leqq 2\kappa)$ we define the polynomial $P_\mu(\theta)$ of degree $\mu+4$ by

$$\begin{aligned} P_\mu(0)&=y_0, & P_\mu'(0)&=Hf(x_0, y_0),\\ P_\mu(1)&=Y_{\kappa\kappa}, & P_\mu'(1)&=Hf(x_0+H, Y_{\kappa\kappa}),\\ P_\mu^{(k)}(1/2)&=H^k d^{(k)} & &\text{for } k=0, \ldots, \mu. \end{aligned} \tag{6.3}$$

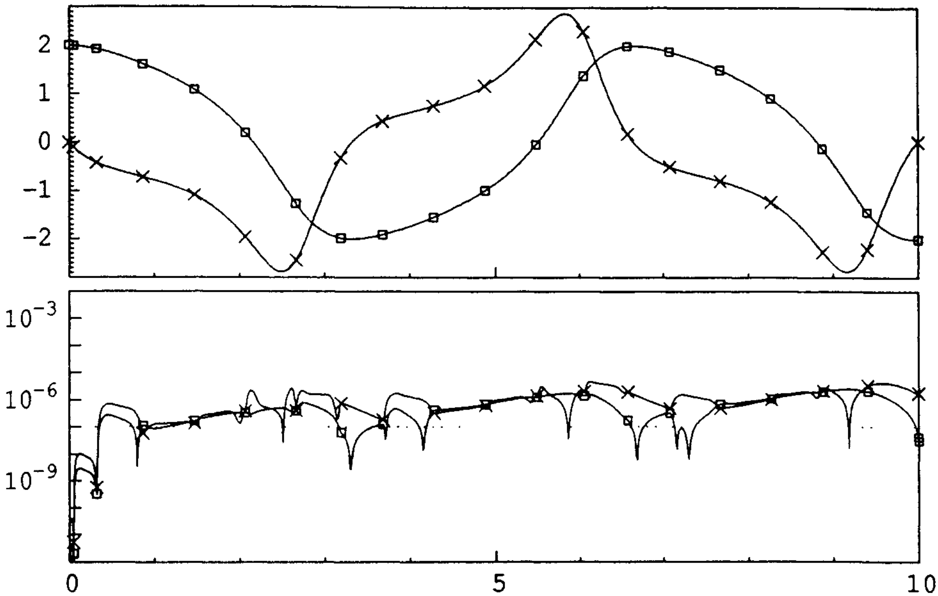

Fig. 3. Solution and error of GBS method

Since $Y_{\kappa\kappa}$ is the initial value of the next step, this dense output formula is globally C^1. We are now ready to discuss its accuracy.

Theorem 8. *If the step number sequence satisfies* (6.1), *then the error of* $P_\mu(\theta)$ *satisfies*

$$P_\mu(\theta)-y(x_0+\theta H)=\begin{cases}\mathbf{O}(H^{2\kappa+1}) & \textit{if } n_1=4 \textit{ and } \mu\geqq 2\kappa-4\\ \mathbf{O}(H^{2\kappa}) & \textit{if } n_1=2 \textit{ and } \mu\geqq 2\kappa-5.\end{cases} \tag{6.4}$$

Proof. Since $P_\mu(\theta)$ is a polynomial of degree $\mu+4$ the interpolation error is of size $\mathbf{O}(H^{\mu+5})$. This explains the restriction on μ in (6.4). We next study the error of $d^{(k)}$. It is well known that the error of y_i possesses an h^2-expansion with coefficients depending on whether the index i is even or odd. Since the symmetric differences in (6.2) use either only even or only odd indices, $d_j^{(k)}$ also has an h^2-expansion

$$d_j^{(k)}=y^{(k)}(x_0+H/2)+h_j^2\, a_{2,k}(x_0+H/2)+h_j^4\, a_{4,k}(x_0+H/2)+\ldots.$$

The extrapolated values therefore satisfy

$$H^k d^{(k)}-H^k y^{(k)}(x_0+H/2)=\begin{cases}\mathbf{O}(H^{2\kappa}) & \text{if } k=0\\ \mathbf{O}(H^{2\kappa+1}) & \text{if } k \text{ odd}\\ \mathbf{O}(H^{2\kappa+2}) & \text{if } k \text{ even and } k\geqq 2.\end{cases}$$

If $n_1/2$ is even, the functions $a_{i,0}(x)$ vanish at x_0 so that $d^{(0)}-y(x_0+H/2)=\mathbf{O}(H^{2\kappa+1})$ in this case. This proves the theorem. □

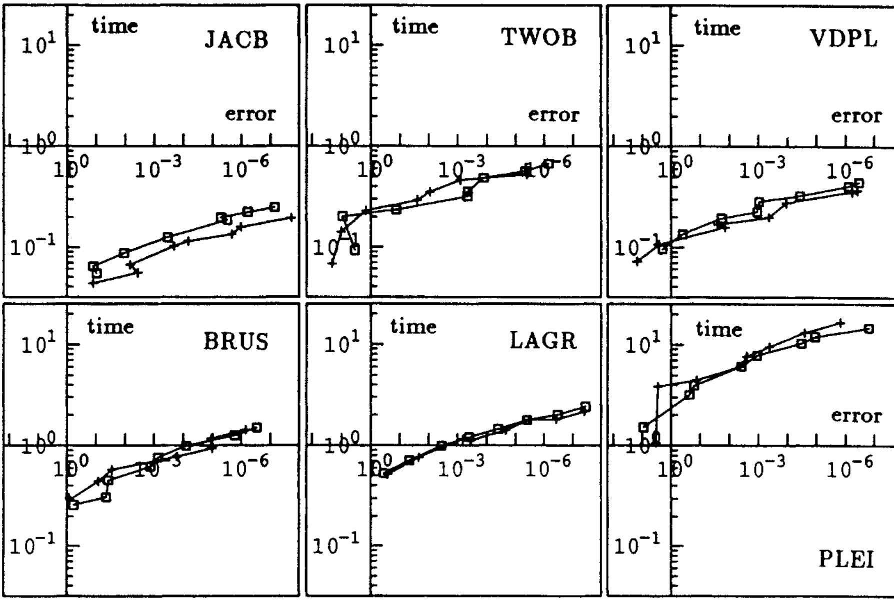

Fig. 4. Work-precision diagrams: □{2, 4, 6, 8, 10, ...}, + {2, 6, 10, 14, 18, ...}

Numerical results. We have provided the code ODEX of [14] with the above dense output formula choosing $\mu = 2\kappa - 3$ and the step number sequence (5.9). It turned out that, whenever a high order is selected in some step, one can observe rather large errors of the dense output in the interior of the corresponding interval. In [15] it has been shown that these errors are mainly due to interpolation. There a new step size strategy is proposed which keeps also the interpolation error comparable to the tolerance. We thus included this strategy from [15] into ODEX and applied it to the problem (4.8) with $\mathrm{TOL} = 10^{-7}$. The result is plotted in Fig. 3.

Standard implementations of the GBS method usually use step number sequences starting with $\{2, 4, 6, 8, \ldots\}$. The above dense output algorithm, however, requires condition (6.1) and leads to a sequence like (5.9). There may thus arise doubts whether this restriction deteriorates the performance of the code. For this reason we have compared the efficiency of both step number sequences in the same implementation on the six test problems of [14, Sect. II.10]. We have run these examples with tolerances $\mathrm{TOL} = 10^{-2}, 10^{-3}, \ldots, 10^{-9}$. Figure 4 shows the obtained work-precision diagrams (computer time on an Apollo workstation against the achieved accuracy at the end point). The results indicate that the efficiency of the code is nearly independent of the used step number sequences.

7 Dense output for the extrapolated linearly implicit Euler

We now turn out attention to a stiff differential equation $y' = f(x, y)$. Our main results of this and the subsequent section concern the linearly implicit Euler method

$$(7.1) \qquad (I - hf_y(x_0, y_0))(y_{i+1} - y_i) = hf(x_i, y_i) + h^2 f_x(x_0, y_0), \qquad i \geqq 0.$$

Here, I denotes the identity matrix, f_x and f_y are the partial derivatives of f. The approximations $y_h(x)=y_n$ (with $x=x_0+nh$) can be extrapolated according to (2.3).

The classical (non-stiff) analysis shows that the error of the extrapolated values Y_{jk} satisfy (2.4). The constant symbolized by the $\mathbf{O}(\ldots)$-term, however, may depend on the stiffness of the problem. For singularly perturbed problems the existence of a perturbed asymptotic expansion, justifying the extrapolation of (7.1), has been proven in [12].

In analogy to (4.1) it seems natural to use the extrapolated values of

$$l_j^{(k)}=\frac{\Delta^k y_0^{(j)}}{h_j^k}, \qquad r_j^{(k)}=\frac{\nabla^k y_{n_j}^{(j)}}{h_j^k} \tag{7.2}$$

for the construction of dense output formulas. To get a feeling to what extent the quality of these expressions is affected by the stiffness, we begin our study with the problem

$$\varepsilon y'=-y+g(x), \tag{7.3}$$

where $\varepsilon>0$ is a small parameter. The exact solution of (7.3) then has the asymptotic ε-expansion

$$y(x)=Ce^{-x/\varepsilon}+g(x)-\varepsilon g'(x)+\varepsilon^2 g''(x)-\varepsilon^3 g'''(x)+\ldots.$$

If we take the initial value on the smooth solution, i.e. such that the exponential term is not present, it becomes

$$y(x)=g(x)-\varepsilon g'(x)+\varepsilon^2 g''(x)-\varepsilon^3 g'''(x)+\ldots. \tag{7.4}$$

Application of (7.1) to (7.3) gives

$$\left(1+\frac{h}{\varepsilon}\right)(y_{i+1}-y_i)=-\frac{h}{\varepsilon}(y_i-g(x_i))+\frac{h^2}{\varepsilon}g'(x_0).$$

This yields the following recursion for the differences $y_i-g(x_i)$

$$y_{i+1}-g(x_{i+1})=\rho(y_i-g(x_i))-\Delta g(x_i)+(1-\rho)hg'(x_0), \tag{7.5}$$

where

$$\rho=\frac{\varepsilon}{h+\varepsilon}, \qquad 1-\rho=\frac{h}{h+\varepsilon}. \tag{7.6}$$

Again we have used $\Delta g(x_i)=g(x_{i+1})-g(x_i)$. Solving this recursion we obtain

$$y_i=g(x_i)-\sum_{j=0}^{i-1}\rho^{i-j-1}\Delta g(x_j)+\rho^i(y_0-g(x_0)-hg'(x_0))+hg'(x_0).$$

If we apply k-times the difference operator Δ to this equation and use the identity

(7.7) $$\Delta\left(\sum_{j=0}^{i-1}\rho^{i-j-1}\varphi(x_j)\right)=\rho^i\varphi(x_0)+\sum_{j=0}^{i-1}\rho^{i-j-1}\Delta\varphi(x_j),$$

which follows by induction on i, we get for $k\geqq 1$

$$\Delta^k y_i=\Delta^k g(x_i)-\sum_{j=0}^{i-1}\rho^{i-j-1}\Delta^{k+1}g(x_j)+\rho^i\Big((\rho-1)^k(y_0-g(x_0)-hg'(x_0))$$
$$-\sum_{m=1}^{k}(\rho-1)^{k-m}\Delta^m g(x_0)\Big).$$

Repeated use of the summation by parts formula

(7.8) $$(1-\rho)\sum_{j=0}^{i-1}\rho^{i-j-1}\varphi(x_j)=\varphi(x_i)-\rho^i\varphi(x_0)-\sum_{j=0}^{i-1}\rho^{i-j-1}\Delta\varphi(x_j)$$

gives for $k\geqq 1$ and $\varepsilon\leqq \mathrm{Const}\cdot h$

(7.9) $$\Delta^k y_i=\sum_{m=k}^{k+N}(\rho-1)^{k-m}\Delta^m g(x_i)-\rho^i\Big(\sum_{m=1}^{k+N}(\rho-1)^{k-m}\Delta^m g(x_0)$$
$$-(\rho-1)^k(y_0-g(x_0)-hg'(x_0))\Big)+\mathbf{O}(h^{k+N+1}).$$

Because of (7.4) the expression multiplied by ρ^i is $\mathbf{O}(h^2)$. Further we have

(7.10) $$(\rho-1)^{k-m}\frac{\Delta^m g(x_i)}{h^k}=(-\varepsilon)^{m-k}g^{(m)}(x_i)+a_1^{(m)}h+\ldots+a_N^{(m)}h^N+\mathbf{O}(h^{N+1})$$

where the functions $a_j^{(m)}$ may depend smoothly on ε and x_i. Inserting (7.10) into (7.9) and using again (7.4) one finally arrives at

(7.11) $$\frac{\Delta^k y_i}{h^k}=y^{(k)}(x_i)+a_1 h+\ldots+a_N h^N+\mathbf{O}(h^{N+1})+\mathbf{O}\left(\frac{\varepsilon^i}{h^{i+k-2}}\right).$$

The analysis in [12, Theorem 2] shows that for $\varepsilon\leqq\mathrm{Const}\cdot h$ the error of Y_{j1} has an h_j-expansion with leading perturbation term of size $\mathbf{O}(\varepsilon^{n_j}h^{2-n_j})$. The accuracy of the extrapolated solution Y_{kk} is thus $\mathbf{O}(H^{k+1}+\varepsilon^{n_1}H^{2-n_1})$. Consequently, for $\varepsilon\ll H$, it was suggested to take step number sequences starting with $n_1=2$. In order to get a comparable accuracy for the dense output formula, one should use the expression (7.11) only for $i\geqq n_1$. This suggests the following construction:

Step 1. For each $j\in\{1,\ldots,\kappa\}$ we compute $r_j^{(k)}$ of Formula (7.2) for $k=1,\ldots,j-\lambda$, where $\lambda\in\{0,1\}$.

Step 2. We extrapolate $r_j^{(k)}$ $(\kappa-k-\lambda)$ times. This yields the improved approximation $r^{(k)}$ to $y^{(k)}(x_0+H)$.

Step 3. We define the polynomial $P_\lambda(\theta)$ of degree κ by

$$(7.12)\qquad \begin{aligned} &P_\lambda(0)=y_0, && P_\lambda(1)=Y_{\kappa\kappa},\\ &P_\lambda^{(k)}(1)=H^k r^{(k)} && \text{for } k=1,\dots,\kappa-1 \end{aligned}$$

The classical (non-stiff) order result can be obtained along the lines of the proof of Theorem 4 and states as follows:

Theorem 9. *For $\lambda\in\{0,1\}$ the error of the interpolation polynomial $P_\lambda(\theta)$ satisfies*

$$P_\lambda(\theta)-y(x_0+\theta H)=\mathbf{O}(H^{\kappa+1-\lambda}). \quad \square$$

Remark. This is a purely non-stiff convergence result. The constant symbolized by the $\mathbf{O}(\dots)$-term usually depends on the stiffness of the problem. E.g., for the model (7.3) it follows from (7.11) that for $\varepsilon\leq \text{Const}\cdot h$

$$(7.13)\qquad P_\lambda(\theta)-y(x_0+\theta H)=\mathbf{O}(H^{\kappa+1-\lambda})+\mathbf{O}(\varepsilon^{n_1+\lambda-1}H^{3-n_1-\lambda}).$$

For $\lambda=1$ and $\varepsilon\ll H$ the second error-term in (7.13) is of the same size as that in the numerical solution Y_{kk} (compare the discussion after Formula (7.11)). However, one power of H is lost in the first term of (7.13). Ideally, λ should be chosen in such a way that both terms are of the same size.

It is now natural to investigate the error of the dense output formula for problems which are significantly more general than (7.3), e.g. singularly perturbed problems

$$(7.14)\qquad \begin{aligned} y'&=f(x,y,z)\\ \varepsilon z'&=g(x,y,z). \end{aligned}$$

For this we need an error estimate for the expressions $\Delta^k y_i/h^k$ which requires the knowledge of perturbed asymptotic expansions of $y_i-y(x_i)$, much more detailed than those provided by Theorem 2 of [12]. This seems to be a difficult task. We therefore study in the next section the limit case $\varepsilon=0$.

8 Differential-algebraic systems for linearly implicit Euler

We consider the differential-algebraic system

$$(8.1)\qquad \begin{aligned} y'&=f(x,y,z), && y(x_0)=y_0,\\ 0&=g(x,y,z), && z(x_0)=z_0, \end{aligned}$$

where we assume that the matrix g_z has a bounded inverse in a neighbourhood of the solution (index 1), and that the initial values are consistent ($g(x_0, y_0, z_0)=0$). The linearly implicit Euler method can be applied to (8.1) as follows:

$$(8.2)\qquad \begin{pmatrix} I-hf_y & -hf_z\\ -hg_y & -hg_z \end{pmatrix}\begin{pmatrix} y_{i+1}-y_i\\ z_{i+1}-z_i \end{pmatrix}=h\begin{pmatrix} f(x_i,y_i,z_i)\\ g(x_i,y_i,z_i) \end{pmatrix}+h^2\begin{pmatrix} f_x\\ g_x \end{pmatrix}$$

where the derivatives $f_x, f_y, \dots$ have to be evaluated at the initial value (x_0, y_0, z_0). These formulas can be derived by applying the method (7.1) to (7.14) and by considering the limit $\varepsilon\to 0$.

The asymptotic behaviour of the errors of y_i, z_i has been studied in [5]. Unfortunately, these errors do not have a pure asymptotic h-expansion (as for ordinary differential equations) but they can be written as

$$y_i - y(x_i) = \sum_{j=1}^{M} h^j (a_j(x_i) + \alpha_i^j) + \mathbf{O}(h^{M+1}) \tag{8.3}$$

$$z_i - z(x_i) = \sum_{j=1}^{M} h^j (b_j(x_i) + \beta_i^j) + \mathbf{O}(h^{M+1})$$

where $a_j(x)$, $b_j(x)$ are smooth functions and the perturbations satisfy

$$\begin{aligned} &\alpha_i^1 = 0, \quad \alpha_i^2 = 0, \quad \alpha_i^3 = 0, \quad \beta_i^1 = 0 \quad \text{for } i \geqq 0 \\ &\beta_i^2 = 0 \quad \text{for } i \geqq 1 \\ &\alpha_i^{j+1} = 0 \quad \text{for } i \geqq j-3 \text{ and } j \geqq 3 \\ &\beta_i^j = 0 \quad \text{for } i \geqq j-2 \text{ and } j \geqq 3 \end{aligned} \tag{8.4}$$

The proof of a slightly weaker assertion can be found in [5, Theorem 4].

Extrapolation of y_n, z_n only eliminates the smooth parts of the error expansion. The perturbations α_n^j, β_n^j (if non-zero) do not disappear. If we consider the harmonic sequence $\{1, 2, 3, 4, \ldots\}$, the extrapolated values Y_{jk}, Z_{jk} satisfy

$$Y_{jk} - y(x_0 + H) = \mathbf{O}(H^{r_{jk}+1}), \qquad Z_{jk} - z(x_0 + H) = \mathbf{O}(H^{s_{jk}}) \tag{8.5}$$

where the *differential-algebraic orders* r_{jk}, s_{jk} are given in Tables 1 and 2, respectively (see [5, 11]). For a step number sequence starting with $n_1 \geqq 2$, the estimates (8.5) hold with r_{jk}, s_{jk} given by the tables which are obtained from Tables 1 and 2 if the diagonals are omitted.

Table 1. Orders r_{jk}

1									
1	2								
1	2	3							
1	2	3	4						
1	2	3	4	4					
1	2	3	4	4	5				
1	2	3	4	4	5	5			
1	2	3	4	4	5	6	5		
1	2	3	4	4	5	6	6	5	
1	2	3	4	4	5	6	7	6	5

Table 2. Orders s_{jk}

2									
2	2								
2	2	3							
2	2	3	4						
2	2	3	4	4					
2	2	3	4	5	4				
2	2	3	4	5	5	4			
2	2	3	4	5	6	5	4		
2	2	3	4	5	6	6	5	4	
2	2	3	4	5	6	7	6	5	4

Condition (8.5) implies that the global error in both components is of size $\mathbf{O}(H^{p_{jk}})$ where $p_{jk} = \min(r_{jk}, s_{jk})$.

For the differential-algebraic system (8.1) one can define a dense output in exactly the same way as it has been done in Sect. 7, Formula (7.12), for ordinary differential equations. As the system (8.1) is partitioned into y- and z-components, it is convenient to denote the corresponding interpolation polynomials by $P_\lambda(\theta)$ and $Q_\lambda(\theta)$, respectively. Recall that these polynomials depend on κ, which is the number of lines in the extrapolation tableau used for their construction.

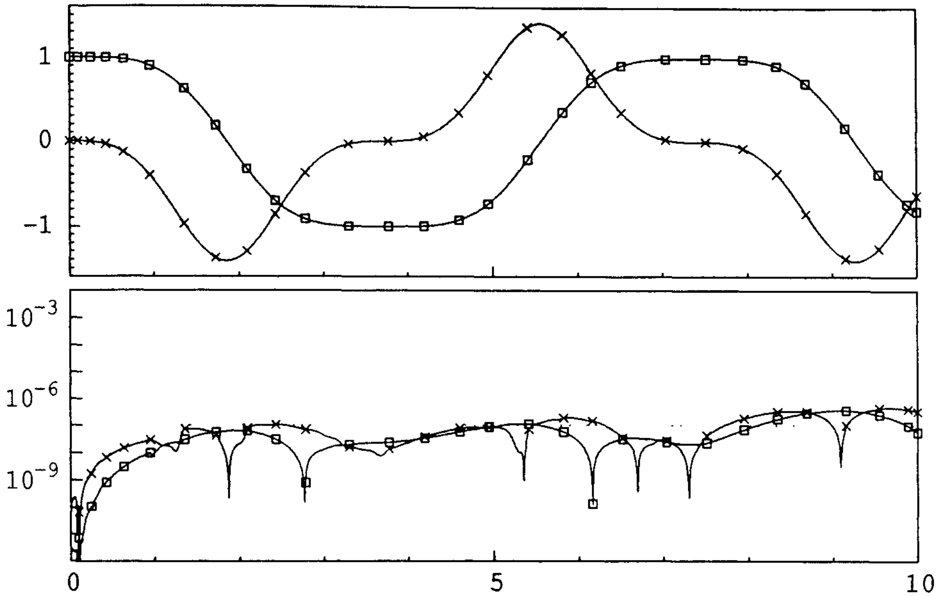

Fig. 5. Solution and error of extrapolated linearly implicit Euler method

Theorem 10. *Let $y(x)$, $z(x)$ be the solution of* (8.1). *Suppose that the step number sequence satisfies $n_1 \geqq 2$, then the interpolation polynomials satisfy*

$$P_\lambda(\theta) - y(x_0 + \theta H) = \mathbf{O}(H^{\kappa+1-\lambda}) + \mathbf{O}(H^{r+1}) \tag{8.6}$$
$$Q_\lambda(\theta) - z(x_0 + \theta H) = \mathbf{O}(H^{\kappa+1-\lambda}) + \mathbf{O}(H^{s})$$

where r and s are the $(\kappa + n_1 + \lambda - 2, \kappa)$-entries of Table 1 *and Table* 2, *respectively.*

Proof. Since $P_\lambda(\theta)$, $Q_\lambda(\theta)$ are polynomials of degree κ, the interpolation error is of size $\mathbf{O}(H^{\kappa+1})$. Inserting (8.3) into the definition of $r_j^{(k)}$ we obtain a perturbed asymptotic expansion of these expressions. Extrapolation of the smooth parts in these expansion yields an error of size $\mathbf{O}(H^{\kappa+1-\lambda})$. It therefore remains to study the influence of the perturbation term in (8.3). For the computation of $r_j^{(k)}$ only y_i with $i \geqq n_j - j + \lambda$ are used. Since $n_j - j \geqq n_1 - 1$ the values $y_0, \ldots, y_{n_1+\lambda-2}$ do not enter the formulas for $r_j^{(k)}$. The perturbation of $y_{n_1+\lambda-1}$ therefore leads to the $\mathbf{O}(H^{r+1})$ and $\mathbf{O}(H^s)$ terms. □

Numerical example. We have provided the code SEULEX of [16] with the dense output of this section. We applied it with the sequence $\{2, 3, 4, 5, 6, \ldots\}$, with $\lambda = 0$ and with TOL $= 10^{-7}$ to the pendulum problem in Index 1 formation

$$\begin{aligned} y_1' &= y_3, & y_1(0) &= 1 \\ y_2' &= y_4, & y_2(0) &= 0 \\ y_3' &= -y_1 y_5, & y_3(0) &= 0 \\ y_4' &= -y_2 y_5 - 1, & y_4(0) &= 0 \\ 0 &= y_3^2 + y_4^2 - y_2 - y_5, & y_5(0) &= 0. \end{aligned} \tag{8.7}$$

Figure 5 shows the solution and the errors of the components y_1 and y_3.

Remarks on other methods. Extrapolation of the *linearly implicit mid-point rule* is a further interesting method for the solution of stiff differential equations (Bader and Deuflhard [2]). It is an extension of the GBS method and the ideas of Sect. 5 can be used to construct dense output formulas. Unfortunately an expansion of the form (8.3) is not known for the numerical solution of the linearly implicit mid-point rule. Therefore, the above proof can not be extended. One way to obtain order results is the following: consider the extrapolated linearly implicit mid-point rule as a Rosenbrock method (see [10]) and verify the order conditions whose derivation can be found in [21]. Dense output for Rosenbrock methods has been considered recently in [19].

The global error of the *implicit Euler* discretisation, applied to (8.1), has an unperturbed asymptotic h-expansion (see [12], [13, Theorem 3.2], [1]). Therefore a dense output can be obtained exactly as in Sect. 4 for the explicit Euler method.

Acknowledgement. We like to thank Ch. Lubich for several hints which gave more insight in the results of this article. We are also grateful to G. Wanner for comments on an early version of this paper.

References

1. Auzinger, W., Frank, R., Macsek, F.: Asymptotic error expansions for stiff equations: The implicit Euler scheme. SIAM J. Numer. Anal. **27**, 67–104 (1990)
2. Bader, G., Deuflhard, P.: A semi-implicit mid-point rule for stiff systems of ordinary differential equations. Numer. Math. **41**, 373–398 (1983)
3. Calvo, M., Montijano, J.I., Randez, L.: New continuous extensions for the Dormand and Prince RK method. Report, Universidad de Zaragoza, Departamento de Matemática Aplicada, 50009 Zaragoza, Spain 1989
4. Deuflhard, P.: Recent progress in extrapolation methods for ordinary differential equations. SIAM Review **27**, 505–535 (1985)
5. Deuflhard, P., Hairer, E., Zugck, J.: One-step and extrapolation methods for differential-algebraic systems. Numer. Math. **51**, 501–516 (1985)
6. Deuflhard, P., Nowak, U.: Extrapolation integrators for quasilinear implicit ODEs. In: Deuflhard, P., Engquist, B. (eds.), Large-scale scientific computing, pp. 37–50, Boston: Birkhäuser 1987
7. Enright, W.H., Jackson, K.R., Nørsett, S.P., Thomson, P.G.: Interpolants for Runge-Kutta formulas. ACM Trans. Math. Software **12**, 193–218 (1986)
8. Enright, W.H., Jackson, K.R., Nørsett, S.P., Thomson, P.G.: Effective solution of discontinuous IVPs using a Runge-Kutta formula pair with interpolants. Applied Math. Comput. **27**, 313–335 (1988)
9. Gladwell, I., Shampine, L.F., Baca, L.S., Brankin, R.W.: Practical aspects of interpolation in Runge-Kutta codes. SIAM J. Sci. Stat. Comput. **8**, 322–341 (1987)
10. Hairer, E., Bader, G., Lubich, C.: On the stability of semi-implicit methods for ordinary differential equations. BIT **22**, 211–232 (1982)
11. Hairer, E., Lubich, C.: On extrapolation methods for stiff and differential-algebraic equations. In: Strehmel, K. (ed.), Numerical treatment of differential equations pp. 64–73, Teubner-Texte zur Mathematik, Band 104, 1988
12. Hairer, E. Lubich, C.: Extrapolation at stiff differential equations. Numer. Math. **52**, 377–400 (1988)
13. Hairer, E., Lubich, C., Roche, M.: The numerical solution of differential-algebraic systems by Runge-Kutta methods. Lecture Notes in Mathematics, Nr. 1409, Berlin Heidelberg New York: Springer 1989

14. Hairer, E., Nørsett, S.P., Wanner, G.: Solving ordinary differential equations I. Nonstiff problems. Berlin Heidelberg New York: Springer 1987
15. Hairer, E., Ostermann, A.: Dense output for the GBS extrapolation method. In the proceedings of the IMA Computational ODE-conference 1989, London (to appear)
16. Hairer, E., Wanner, G.: Solving ordinary differential equations II. Stiff and differential-algebraic problems. Berlin Heidelberg New York: Springer 1990 (to appear)
17. Horn, M.K.: Fourth and fifth-order scaled Runge-Kutta algorithms for treating dense output. SIAM J. Numer. Anal. **20**, 558–568 (1983)
18. Lindberg, B.: A simple interpolation algorithm for improvement of the numerical solution of a differential equation. SIAM J. Numer. Anal. **9**, 662–668 (1972)
19. Ostermann, A.: Continuous extensions of Rosenbrock-type methods. Computing **44**, 59–68 (1990)
20. Owren, B., Zennaro, M.: Order barriers for continuous explicit Runge-Kutta methods. Report, University of Trondheim, 7034 Trondheim, Norway 1989
21. Roche, M.: Rosenbrock methods for differential algebraic equations. Numer. Math. **52**, 45–63 (1988)
22. Shampine, L.F.: Interpolation for Runge-Kutta methods. SIAM J. Numer. Anal. **22**, 1014–1027 (1985)
23. Shampine, L.F., Baca, L.S., Bauer, H.-J.: Output in extrapolation codes. Comput. Math. Appls. **9**, 245–255 (1983)
24. Shampine, L.F., Gladwell, I., Brankin, R.W.: Reliable solution of special event location problems for ODEs. Report, Southern Methodist University, Dallas, Texas 75275 (1988)
25. Stoer, J., Bulirsch, R.: Introduction to Numerical Analysis, 1st Ed. New York Heidelberg Berlin: Springer 1980
26. Zennaro, M.: Natural continuous extensions of Runge-Kutta methods. Math. Comput. **46**, 119–133 (1986)