On the Blunting Method in Verified Integration of ODEs*

Nedialko S. Nedialkov† Kenneth R. Jackson‡ Markus Neher§

Abstract

Verified methods for the integration of initial value problems (IVPs) for ODEs aim at computing guaranteed error bounds for the flow of an ODE while maintaining a low level of overestimation. This paper is concerned with one of the sources of overestimation: a matrix-vector product describing a parallelepiped in phase space.

We analyze the blunting method developed by Berz and Makino, which consists of a special choice of the matrix in this product. For the linear model problem

$$u' = Au, \quad u(0) = u_0 \in \boldsymbol{u}_0,$$

where $u \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times m}$, $m \geq 2$, and $\boldsymbol{u}_0$ is a given interval vector, we compare the convergence behavior of the blunting method with that of the well-known QR method. In both methods, the amount of overestimation of the flow of the initial set depends on the spectral radius of some well-defined matrix. We show that under certain conditions, the spectral radii of the matrices that describe the excess propagation in the QR method and in the blunting method have the same limits, and the excess propagation in both methods is similar.

1 Verified Integration of IVPs

Consider the set of autonomous IVPs

$$u' = f(u), \quad u(t_0) = u_0 \in \boldsymbol{u}_0, \quad t \in \boldsymbol{t} = [t_0, t_{\text{end}}], \tag{1}$$

on some domain $D \subset \mathbb{R}^m$, where $f \in C^n(D)$, $f : D \to \mathbb{R}^m$; $\boldsymbol{u}_0$ is a given interval vector in the space variables; and $t_{\text{end}} > t_0$ is a given endpoint of the time interval. The ODE is defined in the traditional way, but the initial value is allowed to vary. In applications, this variability is used for modeling uncertainties in initial conditions.

*This research was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

†Department of Computing and Software, McMaster University, Hamilton, ON, L8S 4K1, Canada (nedialk@mcmaster.ca).

‡Computer Science Department, University of Toronto, 10 King's College Rd, Toronto, ON, M5S 3G4, Canada (krj@cs.toronto.edu).

§Institute for Applied and Numerical Mathematics, Karlsruhe Institute of Technology (KIT), 76049 Karlsruhe, Germany (markus.neher@kit.edu).

For each $u_0 \in \boldsymbol{u}_0$, the point IVP

$$u' = f(u), \quad u(t_0) \;=\; u_0$$

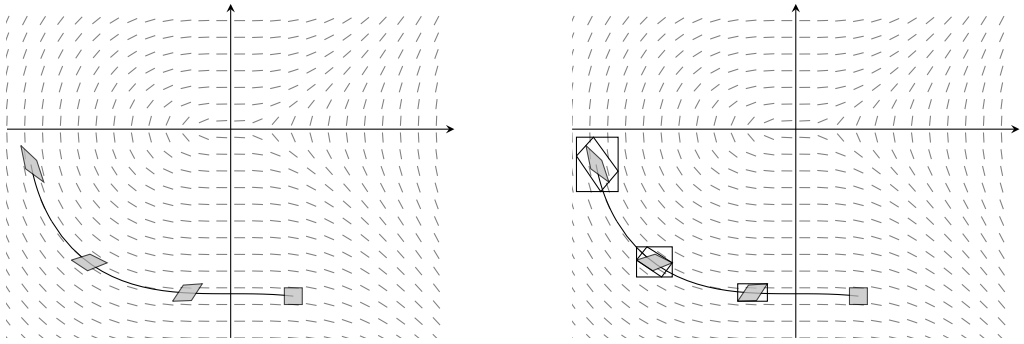
has a classical solution, denoted by $u(t; u_0)$. In the following, we assume that $u(t; u_0)$ exists and is bounded for all $t \in \boldsymbol{t}$ and for all $u_0 \in \boldsymbol{u}_0$.

Our goal, when solving (1), is to calculate bounds on the flow of the interval IVP. For each $t \in \boldsymbol{t}$, we wish to calculate an interval vector $\boldsymbol{u}(t)$ such that

$$u(t; u_0) \in \boldsymbol{u}(t) \qquad \text{for all } u_0 \in \boldsymbol{u}_0.$$

The tube $\boldsymbol{u}(t)$, $t \in \boldsymbol{t}$, then contains all solutions of $u' = f(u)$ that emerge from $\boldsymbol{u}_0$.

We illustrate in Figure 1(a) a flow emanating from an initial set $\boldsymbol{u}_0 \subset \mathbb{R}^2$ in the phase plane. The direction field of the ODE is described by the arrows. The initial set $\boldsymbol{u}_0 \subset \mathbb{R}^2$ at $t = 0$ is defined by the rightmost square. As time passes, this set moves to the left in the phase plane and is deformed along its path. At some time $t_1 > 0$, the transported flow has changed to the shape of the set in the middle of the picture, whereas the leftmost set shows the flow at some time $t_2 > t_1$.



(a) Flow of an initial set. (b) Interval enclosures for the flow.

Figure 1: Flow of an ODE and its enclosures.

A verified integration method for ordinary IVPs computes enclosure sets for the flow at distinct points t_j, $j = 0, \ldots, N$ [6, 8, 9, 10, 11, 13, 15, 18, 21, 23, 24, 26, 31, 32, 33, 34, 35, 38, 39, 40, 41, 42, 43, 44, 47, 48, 49, 50]. In Figure 1(b), the first three time steps of such an enclosure method are illustrated.

At $t = t_1$, the rectangle $\boldsymbol{u}_1$ is an enclosure set for the flow of $\boldsymbol{u}_0$ at $t = t_1$. Typically, $\boldsymbol{u}_1$ overestimates this flow. This wrapping of the flow by $\boldsymbol{u}_1$ introduces overestimation in any verified integration method. Instead of the exact flow at $t = t_1$, the enclosure set $\boldsymbol{u}_1$ is used as the initial set in the second integration step,

$$u' = f(u), \quad t \in [t_1, t_2], \quad u(t_1) \in \boldsymbol{u}_1.$$

Then, at $t = t_2$, the flow of $\boldsymbol{u}_1$ is enclosed by some set $\boldsymbol{u}_2$, depicted by the leftmost square in Figure 1(b). We observe that the flow of the original initial set $\boldsymbol{u}_0$ at $t = t_2$ is contained in $\boldsymbol{u}_2$, but there is already substantial overestimation in this picture.

We assume that the reader is familiar with real interval arithmetic [1, 19, 35, 36, 45]. In this paper, the set of compact real intervals is denoted by

$$\mathbb{IR} = \{\, \boldsymbol{x} = [\underline{x}, \overline{x}] \mid \underline{x}, \overline{x} \in \mathbb{R},\ \underline{x} \le \overline{x} \,\}.$$

Intervals are denoted by boldface. Lower-case letters are used to denote scalars and vectors. Matrices are denoted by upper-case letters. The *midpoint* and the *width* of an interval $\boldsymbol{x}$ are denoted by $\mathrm{m}(\boldsymbol{x}) := (\overline{x} + \underline{x})/2$ and $\mathrm{w}(\boldsymbol{x}) := \overline{x} - \underline{x}$, respectively. The set of all m-dimensional interval vectors is denoted by $\mathbb{IR}^m$.

This paper is organized as follows. The general interval Taylor method for ordinary IVPs is outlined in Section 2. In section 3, several interval methods for a linear model problem are compared. The blunting method for this model problem is analyzed in Section 4. Numerical experiments are reported in Section 5, and conclusions are in Section 6.

2 Interval Taylor Series Methods for IVPs

In this section, we review interval Taylor series methods for the IVP (1). The basic idea is to approximate each solution of (1) by its Taylor polynomial:

$$u(t) \approx u(t_0) + \sum_{k=1}^{n-1} \frac{u^{(k)}(t_0)}{k!}(t - t_0)^k. \tag{2}$$

The Taylor coefficients in (2) are computed by differentiating $u' = f(u)$ with respect to t. From the chain rule, we obtain

$$\frac{u''}{2} = \frac{1}{2}\left(\frac{\partial}{\partial u} f(u)\right) u' = \frac{1}{2}\left(\frac{\partial}{\partial u} f(u)\right) f(u).$$

Higher-order Taylor coefficients of u are computed similarly. Letting

$$f^{[0]}(u) = u, \quad f^{[k]}(u) = \frac{1}{k}\left(\frac{\partial f^{[k-1]}}{\partial u} f\right)(u) \quad \text{for} \quad k \geq 1,$$

values of these coefficients can be computed recursively by automatic differentiation from initial values u_0 or $\boldsymbol{u}_0$, and the well-known formulas for the Taylor coefficients of sums, products, quotients, and the standard functions [16, 35, 46].

In verified integration, it is not sufficient to compute an approximation to the solution. Taylor's theorem, however, allows us to compute not only the approximate solution in (2), but also a rigorous error bound for it. According to Taylor's theorem,

$$u(t) = u(t_0) + \sum_{k=1}^{n-1} f^{[k]}\big(u(t_0)\big)(t - t_0)^k + f^{[n]}(u_\xi)(t - t_0)^n, \tag{3}$$

where $u_\xi = \big(u_1(\xi_1), u_2(\xi_2), \ldots, u_m(\xi_m)\big)$, and each ξ_i is between t_0 and t_1. Now suppose that for some $t_1 > t_0$, componentwise bounds $\underline{u}, \overline{u}$ fulfilling

$$\underline{u} \leq u(t) \leq \overline{u} \quad \text{for all } t \in [t_0, t_1]$$

are known. Then the interval arithmetic evaluation of $f^{[n]}([\underline{u}, \overline{u}])$ yields rigorous lower and upper bounds for the nth Taylor coefficient, and thus also lower and upper bounds for the solutions of (1), for given $u(t_0) \in \boldsymbol{u}_0$ and for all $t \in [t_0, t_1]$.

Suitable values $\underline{u}, \overline{u}$ can be computed by Picard's iteration [35]. Using the Picard-Lindelöff operator and the Banach fixed-point theorem, one can show that, if h_0 and $\boldsymbol{v}_0 \supseteq \boldsymbol{u}_0$ satisfy

$$\widetilde{\boldsymbol{u}}_0 = \boldsymbol{u}_0 + [0, h_0] f(\boldsymbol{v}_0) \subseteq \boldsymbol{v}_0,$$

then (1) has a unique solution $u(t; u_0)$ for $t \in [t_0, t_0 + h_0]$ and each $u_0 \in \boldsymbol{u}_0$. Moreover, $u(t; u_0) \in \widetilde{\boldsymbol{u}}_0$ holds for all $t \in [t_0, t_0 + h_0]$ and all $u_0 \in \boldsymbol{u}_0$. The interval bound $\widetilde{\boldsymbol{u}}_0$ is called a coarse enclosure of the flow of $\boldsymbol{u}_0$, for $t \in [t_0, t_1]$, where $t_1 = t_0 + h_0$.

Picard's iteration is easily performed, but it has the disadvantage that if often restricts the step size of verified integration methods. More advanced schemes for computing coarse enclosures are discussed in [12, 26].

Once a coarse enclosure is available, it can be used to obtain tighter bounds on the flow by a refinement procedure. Inserting $\widetilde{\boldsymbol{u}}_0$ in the formula for the truncation error of the Taylor series in (3), the inclusion

$$u(t_1; u_0) \in \boldsymbol{u}_1 = \boldsymbol{u}_0 + \sum_{k=1}^{n-1} h_0^k f^{[k]}(\boldsymbol{u}_0) + h_0^n f^{[n]}(\widetilde{\boldsymbol{u}}_0) \tag{4}$$

is obtained. This formula is not useful for practical computations, however, because it is width increasing in time. That is,

$$\mathrm{w}(\boldsymbol{u}_1) = \mathrm{w}(\boldsymbol{u}_0) + \mathrm{w}\left(\sum_{k=1}^{n-1} h_0^k f^{[k]}(\boldsymbol{u}_0) + h_0^n f^{[n]}(\widetilde{\boldsymbol{u}}_0) \right) \geq \mathrm{w}(\boldsymbol{u}_0),$$

and typically $\mathrm{w}(\boldsymbol{u}_1) > \mathrm{w}(\boldsymbol{u}_0)$.

Tighter enclosures can be achieved by applying the mean-value form to $f^{[k]}$ in (4). For any $\widehat{u}_0 \in \boldsymbol{u}_0$, it holds that

$$u(t_1; u_0) \in \widehat{u}_0 + \sum_{k=1}^{n-1} h_0^k f^{[k]}(\widehat{u}_0) + h_0^n f^{[n]}(\widetilde{\boldsymbol{u}}_0) + \left(I + \sum_{k=1}^{n-1} h_0^k \frac{\partial f^{[k]}}{\partial u}(\boldsymbol{u}_0) \right) (\boldsymbol{u}_0 - \widehat{u}_0), \tag{5}$$

where I is the identity matrix, and $\partial f^{[k]} / \partial u$ is the Jacobian of $f^{[k]}$. These Jacobians can be computed via automatic differentiation (see e.g. [23, 38]).

Letting

$$\boldsymbol{S}_0 = I + \sum_{k=1}^{n-1} h_0^k \frac{\partial f^{[k]}}{\partial u}(\boldsymbol{u}_0) \quad \text{and} \quad \boldsymbol{z}_1 = h_0^n \, f^{[n]}(\widetilde{\boldsymbol{u}}_0),$$

we write (5) as

$$u(t_1; u_0) \in \boldsymbol{u}_1 = \widehat{u}_0 + \sum_{k=1}^{n-1} h_0^k f^{[k]}(\widehat{u}_0) + \boldsymbol{z}_1 + \boldsymbol{S}_0(\boldsymbol{u}_0 - \widehat{u}_0). \tag{6}$$

This approach was introduced by Moore [33, 34, 35]. Its main disadvantage lies in the computation of the interval vector $\boldsymbol{S}_0(\boldsymbol{u}_0 - \widehat{u}_0)$. To obtain a valid enclosure method for the flow of (1), it would be sufficient to compute a bound on the set

$$\mathcal{S} = \{ S_0(u_0 - \widehat{u}_0) \mid S_0 \in \boldsymbol{S}_0, \ u_0 \in \boldsymbol{u}_0 \}.$$

Wrapping $\mathcal{S}$ by $\boldsymbol{S}_0(\boldsymbol{u}_0 - \widehat{u}_0)$ may result in significant overestimation of $\mathcal{S}$, which also causes overestimation in the initial set for the next integration step (cf. Section 1).

This so-called *wrapping effect* was first observed by Moore in 1965 [34]; recent analyses of it are in [25, 39]. To reduce wrapping in interval methods for the verified integration of IVPs, several schemes have been proposed in the literature [15, 21, 23, 24, 34]. On a general scale, interval arithmetic has also been extended with symbolic computations [14, 20, 27] to diminish the wrapping effect. Starting in the 1990s, Berz and his group developed a rigorous multivariate Taylor arithmetic [2, 27, 30]. Taylor model arithmetic has been defined in [2, 5, 27, 29, 30]. A software implementation of Taylor model arithmetic has been developed by Berz and Makino [3, 28] in the COSY Infinity package [4]. Using COSY Infinity, Taylor models have been applied with success to a variety of problems, including global optimization [37], verified multi-dimensional integration [7], and the verified solution of ODEs and DAEs [6, 17].

The major source of overestimation in (6) is in the matrix-vector product $\boldsymbol{S}_0(\boldsymbol{u}_0-\widehat{u}_0)$. The amount of overestimation is effectively reduced if this product is not performed explicitly, but if $\mathcal{S}$ is propagated in the integration scheme as a parallelepiped instead of an interval. This is achieved by introducing local coordinate systems described by matrices, denoted here by B_j.

Letting $\widehat{u}_0 = \mathrm{m}(\boldsymbol{u}_0)$, $\boldsymbol{r}_0 = \boldsymbol{u}_0 - \widehat{u}_0$, $B_0 = I$, choosing nonsingular matrices B_j, and performing a sequence of integration steps, formula (6) is extended to the interval iteration

$$\left.\begin{aligned} \widehat{u}_j &= \widehat{u}_{j-1} + \sum_{k=1}^{n-1} h_{j-1}^k f^{[k]}(\widehat{u}_{j-1}) + \mathrm{m}(\boldsymbol{z}_j), \\ \boldsymbol{u}_j &= \widehat{u}_{j-1} + \sum_{k=1}^{n-1} h_{j-1}^k f^{[k]}(\widehat{u}_{j-1}) + \boldsymbol{z}_j + (\boldsymbol{S}_{j-1}B_{j-1})\boldsymbol{r}_{j-1}, \end{aligned}\right\} \quad j = 1, 2, \ldots, \tag{7}$$

in which $\widehat{u}_j$ is an approximate point solution, $\boldsymbol{z}_j$ is an enclosure of the (usually small) local error of the jth integration step, and the global error $\boldsymbol{r}_j$ is propagated according to

$$\boldsymbol{r}_j = \big(B_j^{-1}(\boldsymbol{S}_{j-1}B_{j-1})\big)\,\boldsymbol{r}_{j-1} + B_j^{-1}\big(\boldsymbol{z}_j - \mathrm{m}(\boldsymbol{z}_j)\big). \tag{8}$$

We are free to choose the matrices B_j to optimize the error propagation properties of the method (7–8). In Moore's method, $B_j = I$ is used in all steps, and the wrapping effect generally appears with this choice. The *parallelepiped* method [15, 23] is obtained by choosing $B_j = \mathrm{m}(\boldsymbol{S}_{j-1}B_{j-1})$. This method, for reducing the wrapping effect, is suitable for pure rotations, but otherwise the matrices B_j tend to become close to singular (and frequently singular in floating-point arithmetic) after a few steps.

The hitherto most successful general scheme for reducing the wrapping effect is Lohner's QR method [23, 24]. In this method, B_j is chosen as the orthogonal matrix Q in the QR factorization of $\mathrm{m}(\boldsymbol{S}_{j-1}B_{j-1})$. Thus, the matrices B_j are always well conditioned. In [39], it has been shown that, for an important subclass of linear autonomous constant-coefficient systems, the bound on the global error of the QR method is not much bigger than the global error of the corresponding approximate Taylor method.

As an alternative to QR factorization, one could modify the matrices B_j of the parallelepiped method such that their condition number does not exceed some suitable bound.

This variant of the parallelepiped method was already mentioned by Lohner in [23], but not implemented in his thesis. Berz and Makino [8, 31] developed this idea into a practical algorithm, which they named the *blunting method*.

The purpose of this paper is the analysis of the blunting method and a comparison of its asymptotic behavior with that of the QR method. We have not been successful in developing an analysis for nonlinear or time-dependent linear systems. Our analysis applies only to the model problem of a linear system with constant coefficients. This, of course, does not cover the full story, but it does provide useful information. In particular, methods that perform poorly for the model problem are likely to perform poorly for other problems as well.

3 Taylor Methods for a Linear Model Problem

We now consider the linear constant-coefficient model problem

$$\begin{aligned} u' &= Au, \qquad (A \in \mathbb{R}^{m\times m},\ m \geq 2) \\ u(0) &= u_0 \in \boldsymbol{u}_0. \end{aligned} \tag{9}$$

The interval Taylor method with constant order n and stepsize h for solving (9) consists of the iteration

$$u_j := Tu_{j-1} + z_j, \quad j = 1, 2, \ldots,$$

where

$$T = T_{n-1}(hA) = \sum_{k=0}^{n-1} \frac{(hA)^k}{k!}, \tag{10}$$

and z_j denotes the local error.

We enclose the flow of (9) by

$$\begin{aligned} u(t_j; \boldsymbol{u}_0) &= \{u(t_j; u_0) \mid u_0 \in \boldsymbol{u}_0\} \\ &\subseteq \{u_j + S_j w + B_j r \mid w \in \boldsymbol{u}_0 - \mathrm{m}(\boldsymbol{u}_0),\ r \in \boldsymbol{r}_j\}, \end{aligned}$$

where $u_j, w, r \in \mathbb{R}^m$, $\boldsymbol{r}_j \in \mathbb{IR}^m$; $S_j, B_j \in \mathbb{R}^{m\times m}$; and B_j must be nonsingular. The set

$$\{u_j + S_j w \mid w \in \boldsymbol{u}_0 - \mathrm{m}(\boldsymbol{u}_0)\}$$

approximates the flow $\{u(t_j; u_0) \mid u_0 \in \boldsymbol{u}_0\}$ at $t = t_j$. The global error is guaranteed to be contained in the set

$$\{B_j r \mid r \in \boldsymbol{r}_j\}.$$

We first formulate the interval methods mentioned in the previous section for the model problem and then develop an error analysis for the blunting method.

3.1 Propagation of the Global Error

In all interval methods that we study in this paper, the iteration is started with $u_0 = \mathrm{m}(\boldsymbol{u}_0)$, $\boldsymbol{r}_0 = 0$, $S_0 = B_0 = I$. The propagation of the global error is described by the iteration (8), which for our model problem is

$$\boldsymbol{r}_j = (B_j^{-1} T B_{j-1})\boldsymbol{r}_{j-1} + B_j^{-1}\big(\boldsymbol{z}_j - \mathrm{m}(\boldsymbol{z}_j)\big). \tag{11}$$

For minimal overestimation, B_j should be chosen such that the inclusion

$$\{TB_{j-1}r + z \mid r \in \boldsymbol{r}_{j-1},\, z \in \boldsymbol{z}_j - \mathrm{m}(\boldsymbol{z}_j)\} \subseteq \{B_j r \mid r \in \boldsymbol{r}_j\} \tag{12}$$

is as a tight as possible.

There is a straightforward geometric interpretation of the inclusion (12). The sum of a parallelepiped (containing the accumulated global error of the previous integration steps) and an interval box (containing the local error of the current integration step) is enclosed into a new parallelepiped, as illustrated by Figure 2.

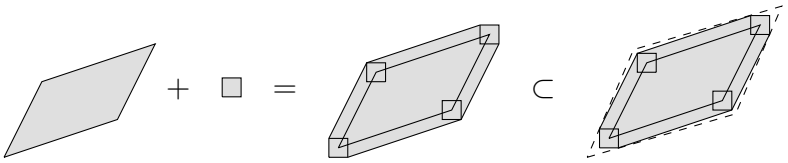


Figure 2: Propagation of the global error.

3.2 Wrapping Effect

In Moore's method [33, 34, 35], $B_j = I$ is used for all j, so that the global error is propagated according to

$$\boldsymbol{r}_j = T\boldsymbol{r}_{j-1} + \boldsymbol{z}_j - \mathrm{m}(\boldsymbol{z}_j).$$

The chosen coordinates are optimal for the local error, which is simply added to the global error. However, if T describes a rotation in phase space, then the volume of $T\boldsymbol{r}_{j-1}$ can be much larger than the volume of $\boldsymbol{r}_{j-1}$.

The parallelepiped method [15, 23] uses $B_j = TB_{j-1}$. Thus,

$$\boldsymbol{r}_j = \boldsymbol{r}_{j-1} + T^{-j}\big(\boldsymbol{z}_j - \mathrm{m}(\boldsymbol{z}_j)\big).$$

The chosen coordinates are optimal for the global error. The local errors are usually small compared to the global error, so that a certain amount of overestimation of the local error may be tolerable. In the parallelepiped method, the local coordinate system is only suitable for the local error if $\mathrm{cond}(T^j)$ is not too big. However, in the presence of shear, T^j becomes singular for $j \to \infty$. In practice, the matrices B_j often become ill-conditioned, resulting in large overestimations and ultimately in the breakdown of the method.

Lohner [23, 24] proposed the following modification of the parallelepiped method. The QR factorization of TB_{j-1} is computed, $Q_jR_j = TB_{j-1}$, and B_j is chosen as the orthogonal matrix Q_j:

$$\boldsymbol{r}_j = R_j\boldsymbol{r}_{j-1} + Q_j^T\big(\boldsymbol{z}_j - \mathrm{m}(\boldsymbol{z}_j)\big).$$

The coordinate system used in the QR method is not optimal for the global error, but rotations occurring in T are accounted for. To minimize overestimation in the QR method, the columns of TB_{j-1} must be ordered according to decreasing length in the orthogonalization process [23, 24].

Since Q_j is an orthogonal matrix, B_j is well-conditioned for all j. So far, the QR method has been the most successful method for simultaneously treating rotation, contraction, and shear. However, there is an example by Kühn [21] showing that even the QR method may fail badly.

In the next section, we study the blunting method of Berz and Makino [8, 31]. In numerical experiments, it has been observed that for certain choices of *blunting factors*, the local coordinates are sometimes better suited for the global error than the coordinates used in the QR method. Like the QR method, the blunting method can simultaneously handle rotation, contraction, and shear, but the actual amount of overestimation depends on these factors.

4 The Blunting Method

As noted above, in the QR method, we perform a QR factorization $TB_{j-1} = Q_jR_j$ and select $B_j = Q_j$. For the model problem, this choice leads to the simultaneous iteration $TQ_{j-1} = Q_jR_j$ [39]. In the blunting method, we select B_j from

$$\begin{aligned} TB_{j-1} &= Q_j^*R_j^* \qquad\qquad (\text{QR factorization of } TB_{j-1}),\\ \widehat{B}_j &= TB_{j-1}D_j + Q_j^*G_j, \\ B_j &= \widehat{B}_jF_j. \end{aligned} \tag{13}$$

We use Q_j^* and R_j^* in (13) to distinguish these matrices from the matrices Q_j and R_j that occur in the QR method. D_j is a diagonal matrix such that $TB_{j-1}D_j$ is normalized, with each column of length one in the Euclidean norm; G_j is a diagonal matrix with heuristically chosen positive blunting factors [8, 31]; and F_j is a diagonal matrix such that the columns of $B_j = \widehat{B}_jF_j$ are of Euclidean norm one.

First, we compare the asymptotic behavior of the blunting and QR methods and then investigate the condition numbers of the matrices B_j in the blunting method.

4.1 Error Propagation: Blunting versus QR

Let

$$V_j = (R_j^*D_j + G_j)F_j. \tag{14}$$

Then $B_j = Q_j^* V_j$. In the QR factorization in (13), it is desirable to ensure that all diagonal entries of R_j^* are positive. Otherwise, it may happen that a diagonal entry of $R_j^* D_j + G_j$ becomes zero, and hence B_j becomes singular. In our analysis, we assume that the QR factorizations in the QR and blunting methods are such that the diagonal of each R_j and R_j^* is positive.

Using (14), we write the blunting iteration (13) as

$$TQ_{j-1}^* = Q_j^*(R_j^* V_{j-1}^{-1}).$$

Choosing $Q_0 = Q_0^* = I$ (where I is the identity matrix), the relations between the respective matrices in the QR and in the blunting methods are

$$Q_j = Q_j^* \quad \text{and} \quad R_j = R_j^* V_{j-1}^{-1}.$$

We are interested (cf. (11)) in the excess propagation in

$$(B_j^{-1} T B_{j-1})\boldsymbol{r}_{j-1}.$$

In the QR method, we have $B_j^{-1} T B_{j-1} = Q_j^T T Q_{j-1} = R_j$, whereas, in the the blunting method, we have

$$B_j^{-1} T B_{j-1} = V_j^{-1} Q_j^T Q_j R_j^* = V_j^{-1} R_j^* = V_j^{-1} R_j V_{j-1}.$$

Since the width of $\boldsymbol{r}_j$ is

$$\mathrm{w}(\boldsymbol{r}_j) = |B_j^{-1} T B_{j-1}| \mathrm{w}(\boldsymbol{r}_{j-1}) + |B_j^{-1}| \mathrm{w}(\boldsymbol{z}_j) \tag{15}$$

(absolute values are taken componentwise), the excess propagation depends on the spectral radii of the matrices specified below.

In the QR method, this matrix is [39]

$$H_{j,i} = |Q_j^{-1} T Q_{j-1}|\, |Q_{j-1}^{-1} T Q_{j-2}| \cdots |Q_{i+1}^{-1} T Q_i| = |R_j|\, |R_{j-1}| \cdots |R_{i+1}|,$$

whereas, in the blunting method, it is

$$\begin{aligned} P_{j,i} &= |B_j^{-1} T B_{j-1}|\, |B_{j-1}^{-1} T B_{j-2}| \cdots |B_{i+1}^{-1} T B_i| \\ &= |V_j^{-1} R_j V_{j-1}|\, |V_{j-1}^{-1} R_{j-1} V_{j-2}| \cdots |V_{i+1}^{-1} R_{i+1} V_i|. \end{aligned}$$

Now we consider the case that T has eigenvalues λ_i of distinct magnitudes, that is, $|\lambda_1| > |\lambda_2| > \cdots |\lambda_m| > 0$. In the QR method, the diagonal of $|R_j|$ converges to $(|\lambda_1|, |\lambda_2|, \ldots, |\lambda_m|)$, as $j \to \infty$. For i and j sufficiently large, the diagonal of $H_{j,i}$ converges to

$$\big(|\lambda_1|^{j-i+1}, |\lambda_2|^{j-i+1}, \ldots, |\lambda_m|^{j-i+1} \big).$$

Denote

$$\alpha_{i,j}^{(k)} = \frac{(V_i)_{k,k}}{(V_j)_{k,k}}. \tag{16}$$

The (k,k) entry of $P_{j,i}$ is

$$(P_{j,i})_{k,k} = \frac{(V_i)_{k,k}}{(V_j)_{k,k}} \prod_{l=i+1}^{j} |(R_l)_{k,k}| = \alpha_{i,j}^{(k)} \prod_{l=i+1}^{j} |(R_l)_{k,k}|.$$

(for all $k = 1, \ldots, m$). Since $|(V_l)_{1,1}| = 1$ for all l, $\alpha_{i,j}^{(1)} = 1$ for all i, j with $i < j$. Hence, the diagonal of $P_{j,i}$ behaves like

$$\big(|\lambda_1|^{j-i+1}, \alpha_{i,j}^{(2)} |\lambda_2|^{j-i+1}, \ldots, \alpha_{i,j}^{(m)} |\lambda_m|^{j-i+1} \big).$$

Now, we derive bounds for the $\alpha_{i,j}^{(k)}$, for $k = 2, \ldots, m$. Since the (diagonal) matrix F_l normalizes each column of $R_l^* D_l + G_l$ in the Euclidean norm, and each column of $R_l^* D_l$ is of length one,

$$\begin{aligned} 1/(F_l)_{k,k} &= \big\| (R_l^* D_l)_{1,k}, ..., (R_l^* D_l)_{k-1,k}, (R_l^* D_l)_{k,k} + (G_l)_{k,k} \big\|_2 \\ &= \sqrt{1 + 2(R_l^* D_l)_{k,k} (G_l)_{k,k} + (G_l)_{k,k}^2} \\ &\leq 1 + (G_l)_{k,k} \end{aligned}$$

$(0 < (R_l^* D_l)_{k,k} \leq 1)$. Then,

$$(V_l)_{k,k} = (R_l^* D_l + G_l)_{k,k} (F_l)_{k,k} \geq (G_l)_{k,k} (F_l)_{k,k} \geq \frac{(G_l)_{k,k}}{1 + (G_l)_{k,k}}. \tag{17}$$

Taking into account that $(V_l)_{k,k} \leq 1$, since the length of the k column of V_l is one (cf. (14)), and using (17) in (16), we obtain

$$\frac{(G_i)_{k,k}}{1 + (G_i)_{k,k}} \leq \alpha_{i,j}^{(k)} \leq 1 + \frac{1}{(G_j)_{k,k}}, \quad \text{for all } k = 2, \ldots, m.$$

Consider the case when all blunting factors are the same, for example $G_j = \epsilon I$. Then

$$\frac{\epsilon}{1+\epsilon} \leq \alpha_{i,j}^{(k)} \leq 1 + \frac{1}{\epsilon}.$$

If, for example, $\epsilon = 10^{-3}$,

$$10^{-3} \approx \frac{10^{-3}}{1 + 10^{-3}} \leq \alpha_{i,j}^{(k)} \leq 1001,$$

whereas for $\epsilon = 1$, the estimation is

$$0.5 \leq \alpha_{i,j}^{(k)} \leq 2.$$

Obviously, as ϵ becomes large, the $\alpha_{i,j}^{(k)}$ become closer and closer to 1, which implies that the QR and the blunting method should exhibit similar behavior.

The case that T has eigenvalues of the same magnitude appears much more difficult to analyze (see also [39]). For this case, we report numerical experiments in §5.3 on two problems that have eigenvalues of the same magnitude.

4.2 Condition numbers of B_j

To obtain insights into the condition numbers of the $B_j = Q_j^* V_j$ matrices, we employ the following result from [22] on condition numbers of triangular matrices. Let U be upper triangular and denote

$$\beta = \max_{l<p} \frac{|U_{l,p}|}{|U_{l,l}|} \quad \text{and} \quad \gamma = \min_l |U_{l,l}|.$$

Then

$$\left\|U^{-1}\right\|_{1,\infty} \leq \frac{(\beta+1)^{m-1}}{\gamma},$$

where equality can be achieved in this bound [22].

For the upper triangular V_j (with positive diagonal entries), we have

$$\beta = \max_{l<p} \frac{|(V_j)_{l,p}|}{(V_j)_{l,l}}, \qquad \gamma = \min_l (V_j)_{l,l}.$$

Since $|(V_j)_{l,p}| < 1$ for all l, p, using (17) we obtain

$$\beta \leq \frac{1}{\min_l (V_j)_{l,l}} \leq \frac{1}{\min_l \frac{(G_j)_{l,l}}{1+(G_j)_{l,l}}} \quad \text{and} \quad \gamma \geq \min_l \frac{(G_j)_{l,l}}{1+(G_j)_{l,l}}.$$

When $G_j = \epsilon I$,

$$\beta \leq 1/\epsilon + 1, \qquad \gamma \geq \epsilon/(1+\epsilon),$$

and

$$\left\|V_j^{-1}\right\|_{1,\infty} \leq \frac{(1+\epsilon)\,(1/\epsilon+1)^{m-1}}{\epsilon}.$$

For $\epsilon = 10^{-3}$, this bound reads

$$\left\|V_j^{-1}\right\|_{1,\infty} \leq (1+10^{-3})10^3(10^3+1)^{m-1},$$

which is huge even for small values of m. In §5.4, we give an example for which the condition number $\mathrm{cond}(B_j)$ is large, although not as large as the above bound.

For $\epsilon = 1$, the more practical bound

$$\left\|V_j^{-1}\right\|_{1,\infty} \leq 2^m$$

is obtained, which ensures that the matrices B_j are well conditioned for $\epsilon \approx 1$, and m not very large.

5 Numerical experiments

In §5.1, we use a simple example to illustrate how the sets $\left\{B_j r \mid r \in \boldsymbol{r}_j\right\}$ propagate in the blunting, parallelepiped, and QR methods. In §5.2, we show these sets for various blunting factors and also for the QR method. In §5.3, we compare the error propagation in the blunting method, for various blunting factors, with the error propagation in the QR method. In §5.4, we show an example for which the B_j's that occur in the blunting method have large condition numbers.

In all the examples that follow, the blunting factor matrix $G_j = \epsilon I$, for all j, $\epsilon > 0$. We perform all experiments with constant stepsize, h.

The computations and plots are produced with MATLAB.

5.1 Blunting versus parallelepiped and QR methods

Consider the problem

$$u' = Au = \begin{pmatrix} 1 & -2 \\ 3 & -4 \end{pmatrix} u.$$

Here and in §5.2, we assume for simplicity that

$$\boldsymbol{z}_j - \mathrm{m}(\boldsymbol{z}_j) = 10^{-12}\,([-10, 10], [-1, 1])^T, \quad \text{for all } j = 0, 1, \ldots;$$

use $h = 0.1$ and set $T = e^{hA}$.

In Figure 3, we show the sets $\{B_j r \mid r \in \boldsymbol{r}_j\}$ computed with the parallelepiped, QR, and blunting methods for $j = 1, \ldots, 9$, where $\epsilon = 0.3$ in the blunting method. (The matrices B_j and the vectors $\boldsymbol{r}_j$ are computed in each of these methods as described earlier.) Although the parallelepiped method gives reasonable enclosures during the first few steps, it produces large overestimations as j increases. On the other hand, both the blunting and QR methods produce much tighter enclosures, with the enclosures associated with the blunting method being a little smaller than those associated with the QR method, for all $j = 1, \ldots, 9$.

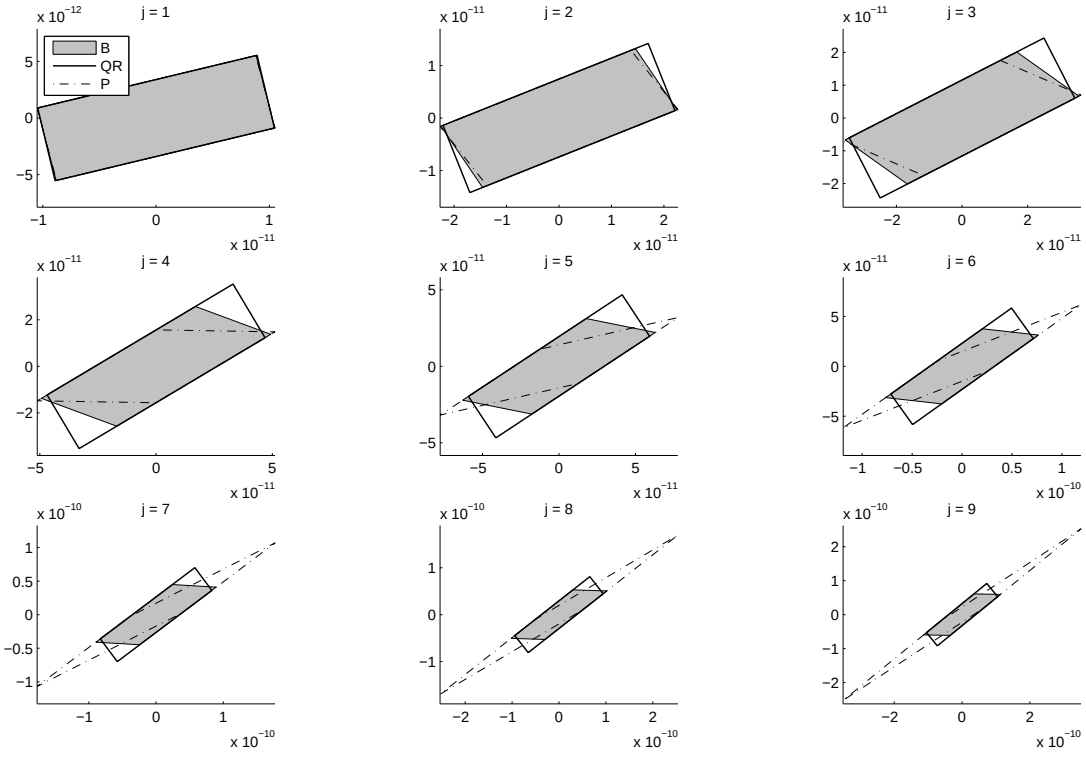


Figure 3: Propagation of the sets $\{B_j r \mid r \in \boldsymbol{r}_j\}$ in the parallelepiped (P), QR (QR), and blunting (B) methods; the blunting factor is 0.3.

5.2 Influence of blunting factors

Now we illustrate how the sets $\{B_j r \mid r \in \boldsymbol{r}_j\}$ associated with the blunting method behave depending on the blunting factor, and compare them with the corresponding enclosure

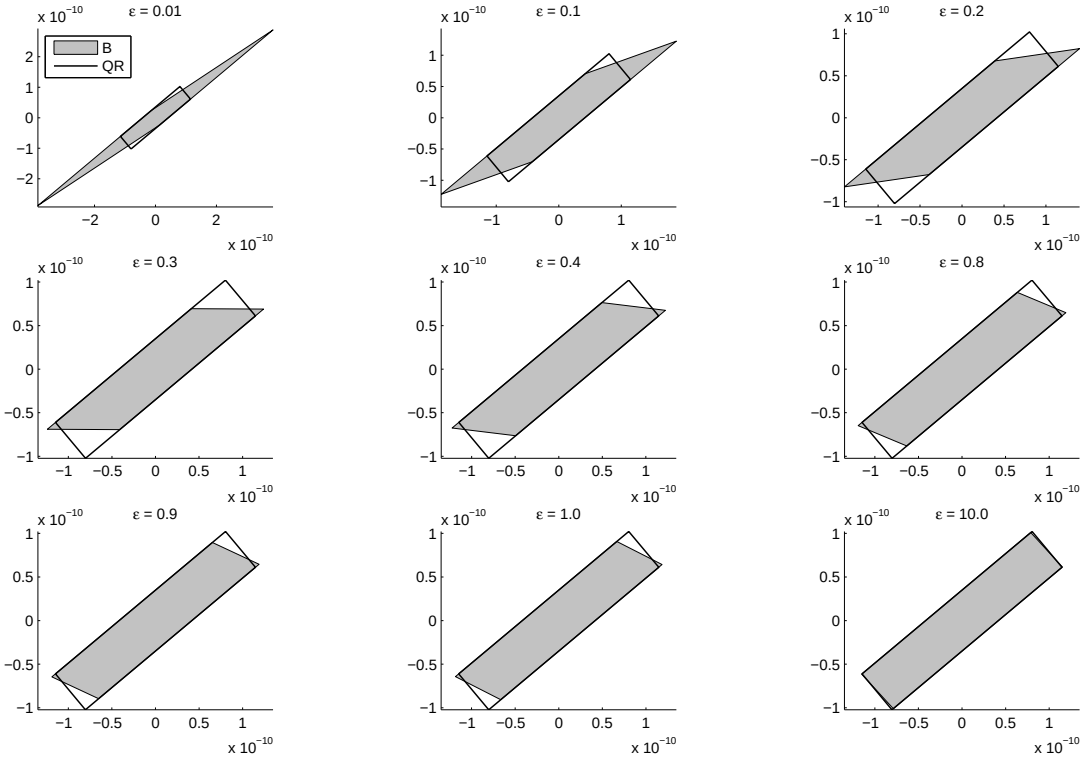


Figure 4: The sets $\left\{B_{100}r \mid r \in \boldsymbol{r}_{100}\right\}$ in the QR (QR) and blunting (B) methods.

sets produced by the QR method. More specifically, for $j = 100$, we show in Figure 4 the sets $\left\{B_{100}r \mid r \in \boldsymbol{r}_{100}\right\}$ for the blunting method with $\epsilon = 0.01, 0.1, 0.2, 0.3, 0.4, 0.8, 0.9, 1,$ and 10, and for the QR method. As expected, for this example, the blunting method behaves more like the parallelepiped method for small ϵ, and more like the QR method for large ϵ.

5.3 Error propagation: blunting versus QR

We compare empirically the error propagation in the QR and blunting methods. To simplify our study, we follow the approach in [39] and compare how the local error on the first step is propagated by these methods for the same order and stepsize of the Taylor series. From (15), this error propagates to the jth step as

$$|B_j^{-1}TB_{j-1}| \cdots |B_2^{-1}TB_1|\,|B_1^{-1}|\,\mathrm{w}(\boldsymbol{z}_1).$$

We investigate the ratio

$$\kappa(j) = \frac{\left\| |B_j^{-1}TB_{j-1}| \cdots |B_2^{-1}TB_1|\,|B_1^{-1}| \right\|_2}{\left\| |R_j|\,|R_{j-1}| \cdots |R_2|\,|Q_1| \right\|_2}$$

($j \geq 1$), where the B_j (for the blunting method) are computed as in (13), and the R_i and Q_1 are from the QR method. We plot this ratio versus j for blunting factors $\epsilon = 0.001, 0.1, 0.5,$ and 1, and four choices for the matrix A of our model problem.

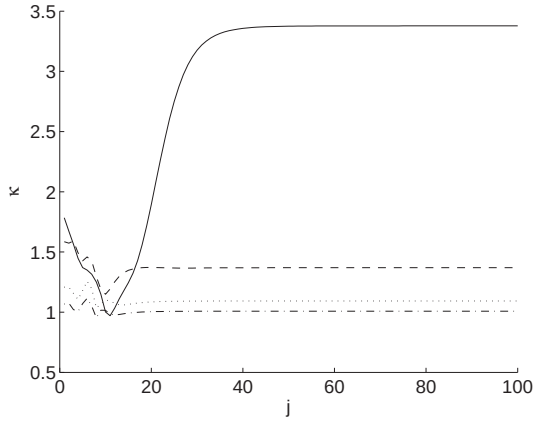
5.3.1 Non-normal A with real negative eigenvalues

In this and the next subsection, we construct A by computing an orthogonal matrix Q from the QR-factorization of a 5×5 Hilbert matrix, choose an R as described below and set $A = QRQ^T$. Then $T = T_{16}(hA)$ is computed from the Taylor polynomial (10), with $n = 17$ and stepsizes as indicated below.

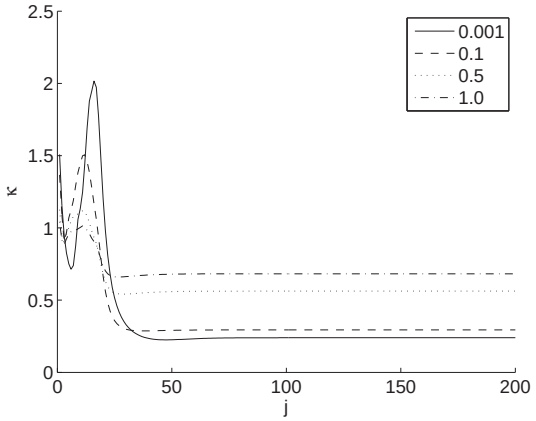
Distinct eigenvalues. In this example,

$$R = \begin{pmatrix} -1 & 6 & 0 & 0 & 0 \\ 0 & -3 & 6 & 0 & 0 \\ 0 & 0 & -5 & 6 & 0 \\ 0 & 0 & 0 & -7 & 6 \\ 0 & 0 & 0 & 0 & -9 \end{pmatrix}$$

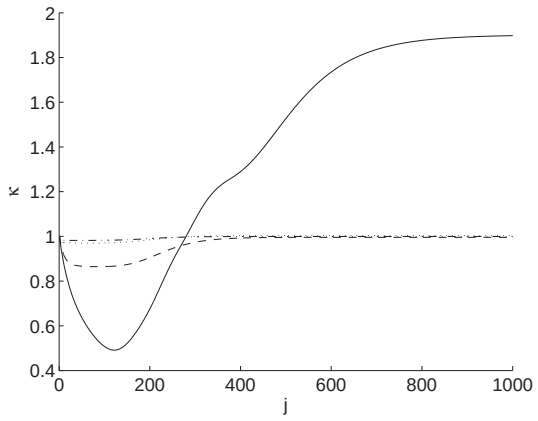
and $h = 0.1$. The plots of $\kappa(j)$ are in Figure 5(a). Here the error propagation in the



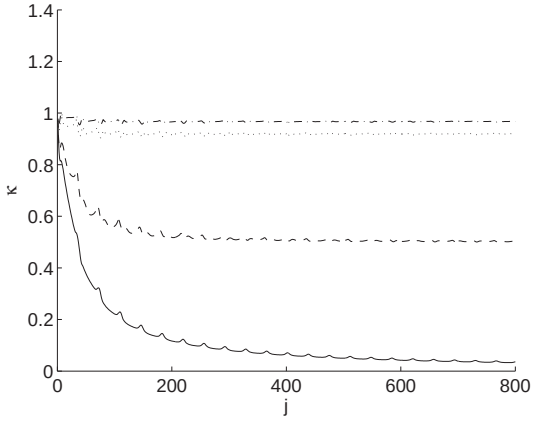
(a) A is non-normal with $\lambda_i = -2i + 1$, $i = 1, \ldots, 5$; $h = 0.1$, $\rho(T) \approx 0.905$.



(b) A is non-normal with $\lambda_i = -3$, $i = 1, \ldots, 4$, $\lambda_5 = -1$; $h = 0.1$, $\rho(T) \approx 0.905$.



(c) A is symmetric with $\lambda_i = i$, $i = 1, \ldots, 5$; $h = 0.01$, $\rho(T) \approx 1.051$.



(d) A has a dominating complex eigenvalue, $h = 0.1$, $\rho(T) \approx 0.971$.

Figure 5: The ratios $\kappa(j)$ for blunting factors $\epsilon = 0.001, 0.1, 0.5$, and 10.

blunting method is consistently worse than the error propagation in the QR method, with small blunting factors giving worse results than large ones.

Multiple same eigenvalues. However, if we change R to

$$R = \begin{pmatrix} -3 & 10 & 0 & 0 & 0 \\ 0 & -3 & 10 & 0 & 0 \\ 0 & 0 & -3 & 10 & 0 \\ 0 & 0 & 0 & -3 & 10 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

and retain $h = 0.1$, we obtain the plots in Figure 5(b). Initially, the blunting method performs worse than the QR method, but, after $j \approx 40$, the former has more favorable error propagation than the latter, with small blunting factors giving better results than large ones.

5.3.2 Symmetric A with positive eigenvalues.

For

$$R = \mathrm{diag}(1, 2, 3, 4, 5)$$

and $h = 0.01$, the corresponding plots are in Figure 5(c). Up to $j \approx 280$ the error propagation for the blunting method for $\epsilon = 0.001, 0.1, 0.5$ is more favorable than in the QR method. However, after $j \approx 280$, the blunting method with all ϵ considered here, except $\epsilon = 0.001$, performs nearly the same as the QR method. For $\epsilon = 0.001$, the performance of the blunting method is worse than that of the QR method.

5.3.3 Dominating complex eigenvalue.

We chose

$$A = \begin{pmatrix} -0.5 & 0.5 & 0 \\ -2.0 & -0.5 & 1 \\ 0 & 1 & -2 \end{pmatrix}.$$

For $n = 17$ and $h = 0.1$, the eigenvalues of T are $\gamma_{1,2} \approx 0.9674 \pm 0.0836i$ and $\gamma_3 \approx 0.7858$. Here $|\gamma_1| = |\gamma_2| \approx 0.9710 > |\gamma_3| \approx 0.7858$. On this example, for $0 < \epsilon \ll 1$, the error propagation for the blunting method is substantially more favorable than for the QR method; see Figure 5(d).

5.4 Poorly conditioned B_j

The bounds in §5.4 suggest that, if the V_j are such that the magnitude of a diagonal entry is much smaller than the magnitude of an entry to the right of it, $\mathrm{cond}(B_j)$ may be large. Such matrices are not difficult to obtain. For example, we can chose a matrix A with diagonal

$$(-0.1, -100, -101, \ldots, -100 - m - 1),$$

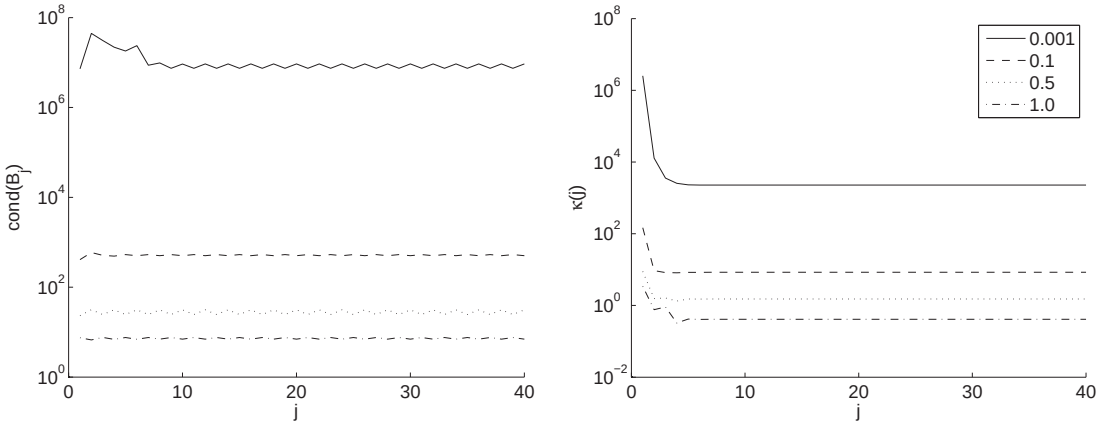
super diagonal with all entries equal to 100, and remaining entries equal to 0. For $m = 9$, $h = 0.1$, and $n = 30$, the T computed from (10) has largest eigenvalue ≈ 0.99, and the remaining eigenvalues are negative and of magnitude 10^{-2} to 10^{-3}. The condition number of T (using the Euclidean norm) is $\approx 5.13 \times 10^{14}$.

Note that, T is upper triangular, and all $Q_j^* = I$, in the blunting method. So,

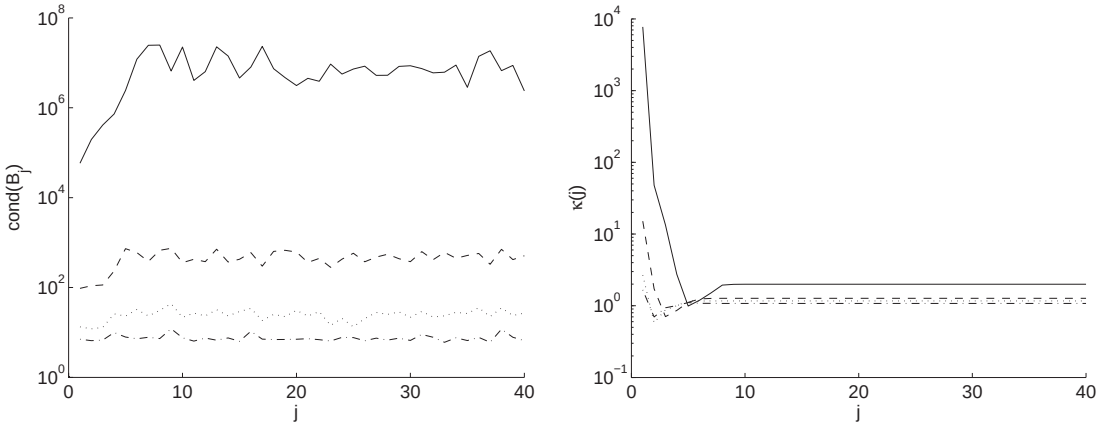
$$B_1 = V_1 = (R_1^* D_1 + G_1)F_1 = (TD_1 + G_1)F_1.$$

$T = R_1^*$ has the eigenvalues of T on its diagonal, in decreasing order of magnitude, and larger entries above it. Scaling by D_1 and F_1 does not change these properties of the matrix, if the blunting factors are small for all $i = 2, \ldots, m$.

In Figure 6(a), we plot on a semi-log scale $\text{cond}(B_j)$ versus j for blunting factors $0.001, 0.1, 0.5$, and 1.0, and in Figure 6(b), we plot the ratios $\kappa(j)$ for the same blunting factors. For this particular A, small blunting factors result in large condition numbers



(a) $\text{cond}(B_j)$ vs. j (b) $\kappa(j)$ vs. j



(c) $\text{cond}(B_j)$ vs. j (d) $\kappa(j)$ vs. j

Figure 6: Plots from ill-conditioned matrices B_j in the blunting method.

and large $\kappa(j)$, and as ϵ decreases in size, both the condition numbers and the $\kappa(j)$ increase in magnitude.

The previous example may seem "artificial." In the last experiment, we chose $A_1 = QAQ^T$, where A is the matrix described in the previous example, and Q is an orthogonal matrix from the QR factorization of a 10×10 Hilbert matrix. We perform the same experiments as above and plot $\text{cond}(B_j)$ and $\kappa(j)$ versus j in Figure 6(c) and (d). Although the condition numbers behave similarly, with larger "wiggles" in the latter example, we observe much smaller $\kappa(j)$ for the blunting method with A_1 than for the blunting method with A, cf. Figure 6(b) and (d).

Since theoretically $T_{29}(hA)$ and $T_{29}(hA_1)$ have the same eigenvalues, we do not have a sound explanation for this different behavior. One reason could be that, due to the floating-point computations, complex eigenvalues appear in $T = T_{29}(hA_1)$, while they do not occur in $T_{29}(hA)$, when evaluated in floating-point arithmetic.

6 Conclusion

We have studied the blunting method in the verified integration of IVPs for ODEs. For our linear model problem, we showed that the blunting method and the QR methods have similar asymptotic performance, if the matrix T in the Taylor method has eigenvalues of distinct magnitude. An extension of our analysis to linear problems with eigenvalues of the same magnitude (this includes the important case that T has a pair of complex conjugate eigenvalues) and to linear, time-dependent problems is desirable. Also, we do not know how to accommodate in our analysis permutations in the QR and blunting methods.

The numerical experiments we have performed show that the blunting method can have more favorable error propagation than the QR method, for some problems and some choices of blunting factors, and less favorable for others. That is, there is no clear winner between these two methods. Moreover, the choice of blunting factor can make a significant difference in the performance of the blunting method. However, how to choose optimal blunting factors does not appear to be easy. In particular, we note that, the suggested blunting factor 10^{-3} [8, 31] may not always be a good choice. Developing a good heuristic for selecting the "right" blunting factors is an interesting challenge for future study.

References

[1] G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press, New York, 1983.

[2] M. Berz. From Taylor series to Taylor models. In *AIP Conference Proceedings 405*, pages 1–23, 1997.

[3] M. Berz. Cosy Infinity Version 8 reference manual. NSCL Technical Report MSUCL-1088, Michigan State University, 1998.

[4] M. Berz. COSY INFINITY. `http://www.bt.pa.msu.edu/index_cosy.htm`, July 2015.

[5] M. Berz and G. Hoffstätter. Computation and application of Taylor polynomials with interval remainder bounds. *Reliable Computing*, 4:83–97, 1998.

[6] M. Berz and K. Makino. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4:361–369, 1998.

[7] M. Berz and K. Makino. New methods for high-dimensional verified quadrature. *Reliable Computing*, 5:13–22, 1999.

[8] M. Berz and K. Makino. Suppression of the wrapping effect by Taylor model-based verified integrators: Long-term stabilization by shrink wrapping. *Int. J. Diff. Eq. Appl.*, 10:385–403, 2005.

[9] M. Berz and K. Makino. Performance of Taylor model methods for validated integration of ODEs. *Lecture Notes in Computer Science*, 3732:65–73, 2006.

[10] G. F. Corliss. Survey of interval algorithms for ordinary differential equations. *Appl. Math. Comput.*, 31:112–120, 1989.

[11] G. F. Corliss. Guaranteed error bounds for ordinary differential equations. In M. Ainsworth, J. Levesley, W. A. Light, and M. Marletta, editors, *Theory and Numerics of Ordinary and Partial Differential Equations*. Clarendon Press, Oxford, 1995.

[12] G. F. Corliss and R. Rihm. Validating an a priori enclosure using high-order Taylor series. In G. Alefeld, A. Frommer, and B. Lang, editors, *Scientific Computing and Validated Numerics*, pages 228–238. Akademie-Verlag, Berlin, 1996.

[13] T. Dzetkulič. Rigorous integration of non-linear ordinary differential equations in Chebyshev basis. *Numer. Algorithms*, 69:183–205, 2015.

[14] J.-P. Eckmann, H. Koch, and P. Wittwer. A computer-assisted proof of universality in area-preserving maps. *Memoirs of the AMS*, 47(289), 1984.

[15] P. Eijgenraam. *The Solution of Initial Value Problems Using Interval Arithmetic*. Mathematical Centre Tracts No. 144, Stichting Mathematisch Centrum, 1981.

[16] A. Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, Philadelphia, 2000.

[17] J. Hoefkens, M. Berz, and K. Makino. Verified high-order integration of DAEs and higher-order ODEs. In W. Krämer and J. Wolff von Gudenberg, editors, *Scientific Computing, Validated Numerics and Interval Methods*, pages 281–292. Kluwer, Dordrecht, Netherlands, 2001.

[18] B. Houska, M.E. Villanueva, and B. Chachuat. A validated integration algorithm for nonlinear ODEs using Taylor models and ellipsoidal calculus. In *IEEE 52nd Annual Conference on Decision and Control (CDC)*, pages 484–489, 2013.

[19] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer, London, 2001.

[20] E. W. Kaucher and W. L. Miranker. *Self-Validating Numerics for Function Space Problems*. Academic Press, New York, 1984.

[21] W. Kühn. Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing*, 61:47–67, 1998.

[22] F. Lemeire. Bounds for condition numbers of triangular and trapezoid matrices. *BIT*, 15(1):58–64, 1975.

[23] R. Lohner. *Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen*. PhD thesis, Universität Karlsruhe, 1988.

[24] R. Lohner. Computation of guaranteed solutions of ordinary initial and boundary value problems. In J. R. Cash and I. Gladwell, editors, *Computational Ordinary Differential Equations*, pages 425–435. Clarendon Press, Oxford, 1992.

[25] R. Lohner. On the ubiquity of the wrapping effect in the computation of error bounds. In U. Kulisch, R. Lohner, and A. Facius, editors, *Perspectives of Enclosure Methods*, pages 201–217. Springer, Wien, 2001.

[26] K. Makino. *Rigorous analysis of nonlinear motion in particle accelerators*. PhD thesis, Michigan State University, 1998.

[27] K. Makino and M. Berz. Remainder differential algebras and their applications. In M. Berz, C. Bischof, G. Corliss, and A. Griewank, editors, *Computational Differentiation: Techniques, Applications and Tools*, pages 63–74. SIAM, Philadelphia, 1996.

[28] K. Makino and M. Berz. COSY INFINITY version 8. *Nuclear Instruments & Methods in Physics Research A*, 427:338–343, 1999.

[29] K. Makino and M. Berz. Efficient control of the dependency problem based on Taylor model methods. *Reliable Computing*, 5:3–12, 1999.

[30] K. Makino and M. Berz. Taylor models and other validated functional inclusion methods. *Int. J. Pure Appl. Math.*, 4:379–456, 2003.

[31] K. Makino and M. Berz. Suppression of the wrapping effect by Taylor model-based verified integrators: Long-term stabilization by preconditioning. *Int. J. Diff. Eq. Appl.*, 10:353–384, 2005.

[32] K. Makino and M. Berz. Suppression of the wrapping effect by Taylor model-based verified integrators: The single step. *Int. J. Pure Appl. Math.*, 36:175–197, 2006.

[33] R. E. Moore. The automatic analysis and control of error in digital computation based on the use of interval numbers. In L. B. Rall, editor, *Error in Digital Computation, Vol. I*, pages 61–130. John Wiley and Sons, New York, 1965.

[34] R. E. Moore. Automatic local coordinate transformations to reduce the growth of error bounds in interval computation of solutions of ordinary differential equations. In L. B. Rall, editor, *Error in Digital Computation, Vol. II*, pages 103–140. John Wiley and Sons, New York, 1965.

[35] R. E. Moore. *Interval Analysis*. Prentice Hall, Englewood Cliffs, N.J., 1966.

[36] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis*. SIAM, Philadelphia, 2009.

[37] P. S. V. Nataraj and K. Kotecha. Global optimization with higher order inclusion function forms. Part 1: A combined Taylor-Bernstein form. *Reliable Computing*, 10:27–44, 2004.

[38] N. S. Nedialkov. *Computing rigorous bounds on the solution of an IVP for an ODE*. PhD thesis, University of Toronto, 1999.

[39] N. S. Nedialkov and K. R. Jackson. A new perspective on the wrapping effect in interval methods for initial value problems for ordinary differential equations. In U. Kulisch, R. Lohner, and A. Facius, editors, *Perspectives of Enclosure Methods*, pages 219–264. Springer, Wien, 2001.

[40] N. S. Nedialkov and K. R. Jackson. Some recent advances in validated methods for IVPs for ODEs. *Appl. Numer. Math.*, 42:269–284, 2003.

[41] N. S. Nedialkov, K. R. Jackson, and G. F. Corliss. Validated solutions of initial value problems for ordinary differential equations. *Appl. Math. Comput.*, 105:21–68, 1999.

[42] N. S. Nedialkov, K. R. Jackson, and J. Pryce. An effective high-order interval method for validating existence and uniqueness of the solution of an IVP for an ODE. *Reliable Computing*, 7:449–465, 2001.

[43] M. Neher. Geometric series bounds for the local errors of Taylor methods for linear nth order ODEs. In G. Alefeld, J. Rohn, S. Rump, and T. Yamamoto, editors, *Symbolic Algebraic Methods and Verification Methods*, pages 183–193. Springer, Wien, 2001.

[44] M. Neher, K. R. Jackson, and N. S. Nedialkov. On Taylor model based integration of ODEs. *SIAM J. Numer. Anal.*, 45:236–262, 2007.

[45] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge, 1990.

[46] L. B. Rall. *Automatic Differentiation: Techniques and Applications, Lecture Notes in Computer Science, Vol. 120*. Springer, Berlin, 1981.

[47] R. Rihm. *Über Einschließungsverfahren für gewöhnliche Anfangswertprobleme und ihre Anwendung auf Differentialgleichungen mit unstetiger rechter Seite*. PhD thesis, Universität Karlsruhe, Karlsruhe, Germany, 1993.

[48] R. Rihm. Interval methods for initial value problems in ODEs. In J. Herzberger, editor, *Topics in Validated Computations*, pages 173–207. Elsevier, Amsterdam, 1994.

[49] H. J. Stetter. Validated solution of initial value problems for ODE. *Notes Rep. Math. Sci. Eng.*, 7:171–193, 1990.

[50] M.E. Villanueva, B. Houska, and B. Chachuat. Unified framework for the propagation of continuous-time enclosures for parametric nonlinear ODEs. *J. Glob. Optim.*, 62(3):575–613, 2015.