SUPPRESSION OF THE WRAPPING EFFECT BY TAYLOR MODEL- BASED VERIFIED INTEGRATORS: LONG-TERM STABILIZATION BY PRECONDITIONING

Kyoko Makino, Martin Berz

Department of Physics and Astronomy
Michigan State University
East Lansing, MI 48824, USA
e-mails: makino@msu.edu, berz@msu.edu

Abstract: The verified integration of ranges of initial conditons through ODEs faces two major challenges, namely the precise representation of the flow over the short term, and the avoidance of unfavorable buildup of errors in the long term. The Taylor model approach is very well suited to overcome the problems of short term flow representation. In this paper we discuss a method based on preconditioning that stabilizes the long-term evolution of the flow. Examples of the performance of the method and comparisons to other approaches are given.

AMS Subject Classification: 65L05, 65G20, 34-04, 41A58
Key Words: Differential equations, ODE, Initial value problem, IVP, Taylor integration, Verification, Rigorous computation, Wrapping effect, Preconditioning, QR method, Taylor model, Interval method

1. Introduction

In this paper we describe a method to stabilize the long-term behavior of Taylor model-based verified integrators based on preconditioning. As discussed elsewhere in detail [2], [17], [4], the following advantages have been

observed when executing the single step by a Taylor model- based integrator. First, the explicit dependency on initial variables is carried through the whole integration process. This controls the bulk of the dependency problem very efficiently and hence the main source of the wrapping effect is eliminated to order $n + 1$ for the single step.

On the practical side, the inclusion requirement asserting existence of a solution reduces to a mere inclusion of the remainder intervals, and different from conventional methods based on two separate algorithms for initial validation by an Euler step and subsequent higher order execution, the entire steps is performed in one algorithm. There is also no need to utilize additional ODEs for derivatives. Finally, the direct availability of the antiderivation on Taylor models allows to treat the Picard operator like any other function, avoiding the need to explicitly bound error terms of integration formulas and leading to a rather straightforward verified fixed point problem.

The results of the methods developed in [17], [14], [2] can be summarized in the following theorem.

Theorem 1. *(**Continuous Dynamical System with Taylor Models**) Let $P + I$ be an n-dimensional Taylor model describing the flow of the ODE at the time t; i.e. for all initial conditions x_0 in the original domain region $B \subset R^n$, we have*

$$x(x_0, t) \in I + \bigcup_{x_0 \in B} P(x_0).$$

Let $P^*(x_0, t)$ be the invariant polynomial depending on x_0 and t obtained in [2], and assume that the self-inclusion step of the Picard Operator mapping described there is satisfied over the interval $[t, t+\Delta t]$ by the remainder bound I^*. Then for all $x_0 \in B$, we have

$$x(x_0, t + \Delta t) \in I^* + \bigcup_{x_0 \in B} P^*(x_0, t + \Delta t).$$

Furthermore, if even $x(x_0, t) \in P(x_0) + I$, then $x(x_0, t + \Delta t) \in P^*(x_0, t + \Delta t) + I^*$.

By induction over the individual steps, we obtain a relationship between initial conditions and final conditions at time t. Thus formally, the continuous case is made equivalent to the discrete case, for which the respective property follows immediately from the respective enclosure properties of Taylor models, as described for example in [16].

Theorem 2. *(Discrete Dynamical System with Taylor Models)*
Let $P+I$ be an n-dimensional Taylor model describing the flow of the discrete dynamical system $x_{n+1} = f(x_n, n)$, i.e. for all initial conditions x_0 in the original domain region $B \subset R^n$, we have

$$x_n(x_0) \in I + \bigcup_{x_0 \in B} P(x_0).$$

Let $P^* + I^*$ be the Taylor model evaluation of $f\,(P+I, n)$. Then for all $x_0 \in B$, we have

$$x_{n+1}(x_0) \in I^* + \bigcup_{x_0 \in B} P^*(x_0).$$

Furthermore, if even $x_n\,(x_0, t) \in P\,(x_0) + I$, then $x_{n+1}\,(x_0) \in P^*(x_0) + I^*$.

The two theorems thus allows the verified study of continuous and discrete dynamical systems, provided that the Taylor model arithmetic is performed in a verified manner. In the case of the implementation in COSY INFINITY, all errors in the floating point coefficients are fully accounted for [16], [19].

For the purpose of practical efficiency, it is important that the treatment of the coefficients arithmetic supports sparsity, i.e. only coefficients that are nonzero (or more specifically, above a pre-specified accuracy threshold [16], [19]) contribute to computational effort, and different variables can be carried to different orders [5].

In the following sections we will study another method for the faithful representation of the verified flow of the ODE, the method of preconditioning. We will illustrate the behavior with a large number of examples.

2. Preconditioning the Flow

In this section we will discuss another method to affect the behavior of the remainder bounds of the solutions of ODEs. The idea is to write the Taylor model of the solution as a composition of two Taylor models $(P_l + I_l)$ and $(P_r + I_r)$, and then choose $P_l + I_l$ in such a way that I_l is zero up to roundoff, and the operations appearing on I_r are minimized so as not to increase the size of I_r significantly. In a wider context, the Taylor model $(P_l + I_l)$ can be viewed as a specific coordinate system in which the motion is studied. For practical purposes, in the factorization we impose that $(P_r + I_r)$ is normalized such that each of its components has a range in $[-1, 1]$; for purposes of

numerical stability, it is advantageous that the range is in fact near $[-1, 1]$. This is achieved by factoring out a linear diagonal transformation containing scaling factors.

Definition 3. Let $(P + I)$ be a Taylor model. We say that $(P_l + I_l)$, S, and $(P_r + I_r)$ form a factorization of $(P + I)$ if the components of the range $B(P_r + I_r)$ of $P_r + I_r$ lie in $[-1, 1]$, S is a diagonal linear scaling transformation, and

$$(P + I) \in (P_l + I_l) \circ S \circ (P_r + I_r) \text{ for all } x \in D.$$

Here D is the domain of the Taylor model $(P+I)$. In this case, we call $P_l + I_l$ the preconditioner, S the scaling, and $P_r + I_r$ the conditioned Taylor model.

The composition $(P_1 + I_1) \circ (P_2 + I_2)$ of the Taylor models $(P_1 + I_1)$ and $(P_2 + I_2)$ is here to be understood as insertion of the Taylor model $(P_2 + I_2)$ into the polynomial P_1 via Taylor model addition and multiplication, and subsequent addition of the remainder bound I_1. For the study of the solutions of ODEs, the following result is important

Proposition 4. Let $(P_{l,n} + I_{l,n}) \circ S_n \circ (P_{r.n} + I_{r,n})$ be a factored Taylor model that encloses the flow of the ODE at time t_n. Let $(P^*_{l,n+1}, I^*_{l,n+1})$ be the result of integrating $(P_{l,n} + I_{l,n})$ from t_n to t_{n+1}. Then

$$(P^*_{l,n+1}, I^*_{l,n+1}) \circ S_n \circ (P_{r,n} + I_{r,n})$$

is a factorization of the flow at time t_{n+1}.

Thus the right factor remains unchanged. Considering that in the beginning of the integration, the flow of the initial condition box can be represented as the composition of two identity Taylor models, this immediately leads to the obvious but uninteresting case of leaving the right factor as the identity throughout the integration, which apparently reduces to the naive Taylor model integration. However, the key to the beneficial use of the method, and in particular its use in reducing the growth of remainder terms, lies in moving terms between the left and right factors.

To actually achieve the factorization, the following steps are necessary. First, observe that according to proposition 4, an inclusion of the flow in a Taylor model is given by $(P^*_{l,n+1} + I^*_{l,n+1}) \circ S_n \circ (P_{r.n} + I_{r,n})$. Let c^*_{n+1}, C^*_{n+1} be the constant and linear parts of $P^*_{l,n+1}$ and $N^*_{l,n+1}$ the nonlinear part and the remainder, so that $P^*_{l,n+1} = c^*_{n+1} + C^*_{n+1} + N^*_{n+1}$. We set $c_{l,n+1} = c^*_{n+1}$ and assume that $C_{l,n+1}$ is the desired linear part of the left factor; more on useful choices for $C_{l,n+1}$ below. We then insert the identity transformation

$(C_{l,n+1} \circ C_{l,n+1}^{-1})$ in front of the parentheses, and thus have an inclusion of the flow as follows:

$$
\begin{aligned}
&\left(c_{n+1}^* + C_{n+1}^* + N_{n+1}^*\right) \circ S_n \circ (P_{r,n} + I_{r,n}) \\
&= c_{n+1}^* + \left(C_{n+1}^* + N_{n+1}^*\right) \circ S_n \circ (P_{r,n} + I_{r,n}) \\
&= c_{l,n+1} + (C_{l,n+1} + [0,0]) \circ \left(C_{l,n+1}^{-1} \circ \left(C_{n+1}^* + N_{n+1}^*\right) \circ S_n \circ (P_{r,n} + I_{r,n})\right) \\
&= (c_{l,n+1} + C_{l,n+1} + [0,0]) \circ \\
&\quad \left\{\left[C_{l,n+1}^{-1} \circ C_{n+1}^* + C_{l,n+1}^{-1} \circ N_{n+1}^*\right] \circ S_n \circ (P_{r,n} + I_{r,n})\right\} \qquad (1)
\end{aligned}
$$

We now denote the expression in the curly brackets by $\left(P_{r,n+1}' + I_{r,n+1}'\right)$ and determine its component bounds, which produces the scaling matrix S_{n+1}. Denoting $(P_{r,n+1} + I_{r,n+1}) = S_{n+1}^{-1} \circ \left(P_{r,n+1}' + I_{r,n+1}'\right)$, we thus have an enclosure of the flow at t_{n+1} as

$$
(c_{l,n+1} + C_{l,n+1} + [0,0]) \circ S_{n+1} \circ (P_{r,n+1} + I_{r,n+1}).
$$

To analyze the effects of this procedure, the following observations are crucial:

1. The polynomial part of $C_{l,n+1}^{-1} \circ N_{n+1}^*$ is purely nonlinear, so its action on $S_n \circ (P_{r,n} + I_{r,n})$ via composition only introduces small contributions to the remainder bound which scale at least quadratically with the components of S_{n+1}. Thus for sufficiently small S_{n+1}, this effect will be small.

2. The remainder part of $C_{l,n+1}^{-1} \circ N_{n+1}^*$, which contains as one important contribution the action of $C_{l,n+1}^{-1}$ on the remainder interval of N_{n+1}^*, will be added to $I_{r,n}$. The magnification of the remainder bound of N_{n+1}^* by the action of $C_{l,n+1}^{-1}$ is proportional to the condition number of $C_{l,n+1}$.

3. Contributions of a similar magnitude as $I_{r,n}$ come from application of the linear term $C_{l,n+1}^{-1} \circ C_{n+1}^*$ to $I_{r,n}$. If this term is not chosen properly, over time, exponential growth of the remainder bound can occur.

We now are ready to consider several choices for the determination of $C_{l,n+1}$. As a first nearly trivial but nevertheless interesting example, we assume that the polynomial $P_{l,n}$ represents the identity:

Definition 5.   (Identity Preconditioning) We choose $C_{l,n+1}$ as the identity:

$$C_{l,n+1} = \mathcal{I}$$

This form of preconditioning amounts merely to moving the remainder error to the right. In the subsequent step, the flow is then computed on an identity without the presence of a remainder bound, which can lead to improved performance. This is somewhat reminiscent of the common distinction between "algorithm 1" and "algorithm 2" of integration approaches such as those in the code AWA, where "algorithm 1" provides a first enclosure over an interval box enclosing the current flow.

As the first nontrivial but nevertheless quite obvious example, we assume that the polynomial $P_{l,n}$ represents the linear flow of the motion.

Definition 6.   (Parallelepiped Preconditioning) We choose

$$C_{l,n+1} = C_{n+1}^*$$

The parallelepiped preconditioning thus has the interesting effect that the entire constant and linear parts of the flow are described by the left factor alone; and the nonlinear parts of the motion and remainder bounds will by accumulated in the right factor. Analyzing the arithmetic more carefully we see that the term $C_{l,n+1}^{-1} \circ C_{n+1}^* + C_{l,n+1}^{-1} \circ N_{n+1}^*$ appearing in the square brackets in eq. 1 plays a crucial role. Its linear part amounts to identity up to floating point error which leads to very favorable numerics in the subsequent composition with $S_n \circ (P_{r,n} + I_{r,n})$.

On the other hand, $C_{l,n+1}^{-1}$ is also acting on the nonlinear part and the remainder bound. However, it is known that in various practical cases of interest, over long periods of time, $C_{l,n+1}$ can become more and more ill-conditioned; this is for example the case in linear problems where the matrix of the ODE has distinct real eigenvalues. Since the multiplication of a matrix with an interval vector leads to an overestimation that scales with the condition number, this effect may lead to a rapid growth of the remainder bound of the term, and thus in cases of ill-conditioned flow is of limited value.

The method can be much improved by the following choice of preconditioner:

Definition 7.   (Blunted Parallelepiped Preconditioning) We choose $C_{l,n+1}$ to be the $q-$blunting of C_{n+1}^*, where q is a suitable blunting factor.

As seen above, the q-blunting provides an upper bound for the condition number of the matrix $C_{l,n+1}$, and thus a strict upper limit to the overestimation obtained when sending the remainder bound interval of N_{n+1}^* through $C_{l,n+1}^{-1} \circ \left(C_{n+1}^* + N_{n+1}^*\right)$. On the other hand, since a sufficiently small choice of q only modifies $C_{l,n+1}$ in a minor amount, we still have that the linear part of $C_{l,n+1}^{-1} \circ \left(C_{n+1}^* + N_{n+1}^*\right)$ is nearly identity, which still favorably affects the subsequent application to $S_n \circ (P_{r,n} + I_{r,n})$. So a suitable choice of q may lead to an acceptable overestimation due to the condition number of $C_{l,n+1}$ while still providing only limited overestimation in the last step. Examples of the effect of blunted parallelepiped preconditioning will be given in the next section.

As another example of preconditioning with a linear transformation, we consider the following choice

Definition 8. (**Curvilinear Preconditioning**) Let $x^{(m)} = f(x, x', ..., x^{(m-1)}, t)$ be an m-th order ODE in n variables. Let $x_r(t)$ be a solution of the ODE and $x_r'(t), ..., x_r^{(k)}(t)$ its first k time derivatives. Let $e_1(t), ..., e_l(t)$ be l unit vectors not in the span of $x_r'(t), ..., x_r^{(k)}(t)$ such that $X = (x_r'(t), ..., x_r^{(k)}(t), e_1(t), ..., e_l(t))$ have maximal rank. Then we call the Gram-Schmidt orthonormalization of X a curvilinear basis of depth k, and we refer to its use for preconditioning as curvilinear preconditioning.

The basis introduced in the last definition is a generalization heuristically found specific choices of coordinates for particular cases that have been used in various disciplines. A special case of curvilinear coordinates is used in the study of the six dimensional dynamics of an object in the solar system, and for the last 50 years in the six-dimensional dynamics in large particle accelerators. For a treatment of their properties in the latter case, see [1], and [14] as well as [15]. As an aside, we note that it is possible to even preserve Hamiltonian structure in the transformation to curvilinear coordinates[14], [3], which is important for long term integration using symplectic methods as in [6] and [7].

Example 9. (**Curvilinear Coordinates for the Solar System and Particle Accelerators**) In this case, $m = 2$, $n = 3$, and one usually chooses $k = 2$. The first basis vector points in the direction of motion of the reference orbit. The second vector perpendicular to it points approximately to the sun or the center of the accelerator. The third vector is chosen perpendicular to the plane of the previous two.

Of particular interest for our study of long-term error growth is the

following case:

Theorem 10.   (Curvilinear Coordinates for Autonomous Linear Systems) *Let $x' = A \cdot x$ be an n-dimensional linear system that has n distinct nonzero eigenvalues λ_i with eigenvectors a_i. Let B be a box with nonzero volume, and $x_r = \sum_{i=1}^{n} X_i a_i \in B$ such that $X_i \neq 0$ for all $i = 1, ..., n$. Then the derivatives of $x_r^{(i)}$, $i = 1, ..., n$, are linearly independent, and hence curvilinear coordinates of depth n can be obtained by applying the Gram-Schmidt procedure to the derivatives $x_r^{(i)}$, $i = 1, ..., n$.*

Proof. The motion of the reference point x_r as a function of time is apparently given by

$$x_r(t) = \sum_{i=1}^{n} X_i \cdot a_i \cdot \exp(\lambda_i t)$$

so that the jth derivative assumes the form

$$x_r^{(j)}(t) = \sum_{i=1}^{n} X_i \cdot a_i \cdot \lambda_i^j \exp(\lambda_i t).$$

We now consider the determinant of the matrix of coefficients in the basis a_i, and observe

$$\det \begin{pmatrix} X_1\lambda_1 & X_1\lambda_1^2 & & X_1\lambda_1^n \\ X_2\lambda_2 & X_2\lambda_2^2 & & X_2\lambda_2^n \\ & & \ddots & \\ X_n\lambda_n & X_n\lambda_n^2 & & X_n\lambda_n^n \end{pmatrix}$$

$$= \prod_{i=1}^{n} (\lambda_i X_i) \cdot \det \begin{pmatrix} 1 & \lambda_1^1 & & \lambda_1^{n-1} \\ 1 & \lambda_2^1 & & \lambda_2^{n-1} \\ & & \ddots & \\ 1 & \lambda_n^1 & & \lambda_n^{n-1} \end{pmatrix} = \prod_{i=1}^{n} (\lambda_i X_i) \prod_{i>j} (\lambda_i - \lambda_j) \neq 0$$

because of the well-known property of the Vandermonde matrix.   □

Definition 11.   (Natural Coordinate System for Linear System) Let $x' = A \cdot x$ be an n-dimensional linear system that has n distinct real eigenvalues $\lambda_1 > \lambda_2 > ... > \lambda_n$ with eigenvectors $a_1, ..., a_n$. We define the normal basis (b_i) of the system to be the result of applying the Gram-Schmidt

orthonormalization procedure to the vectors $a_1, ..., a_n$, i.e. the result of the recursive computation

$$b_i = \frac{a_i - \sum_{j=1}^{i-1} b_j \cdot (a_i \cdot b_j)}{\left| a_i - \sum_{j=1}^{i-1} b_j \cdot (a_i \cdot b_j) \right|}.$$

The Natural Coordinate System has the property that as time progresses, the motion is pulled most towards the vector b_1, and then towards b_2, and so on.

Proposition 12.   (Curvilinear Coordinates for Autonomous Linear Systems) Let $x' = A \cdot x$ be an n-dimensional linear system that has n distinct real eigenvalues λ_i with eigenvectors a_i. Let b_i be the natural coordinate system of the linear system. Let B be a box with nonzero volume, and $x_r = \sum_{i=1}^{n} X_i a_i \in B$ such that $X_i \neq 0$. If x_r is used as the reference orbit to define the curvilinear coordinates c_i, then the curvilinear coordinates converge to the natural coordinates, i.e. we have

$$c_i \rightarrow b_i \text{ for all } i \text{ as } t \rightarrow \infty.$$

Proof. The derivatives of the motion of the reference point x_r as a function of time of order 0 and higher are apparently given by

$$x_r^{(j)}(t) = \sum_{i=1}^{n} X_i \cdot a_i \cdot \lambda_i^j \exp(\lambda_i t).$$

Because of the ordering of the eigenvectors by size, we clearly have $c_1 = x_r'(t)/|x_r'(t)| \rightarrow b_1$ as $t \rightarrow \infty$. Since c_2 is perpendicular to c_1, we thus also have that $c_2 \cdot b_1 \rightarrow 0$ as $t \rightarrow \infty$, and so $\lim_{t \rightarrow \infty} c_2$ is in the span of $b_2, ..., b_n$. Because in this subspace, the coefficient $\exp(\lambda_2 t)$ is dominating, we even have $c_2 \rightarrow b_2$ as $t \rightarrow \infty$. In a similar fashion we obtain iteratively that $c_j \rightarrow b_j$ as $t \rightarrow \infty$. $\square$

Remark 13.   Variations of these arguments are obviously possible to treat the case of complex eigenvalues. In this case, the "natural" generalization of the natural coordinate system has two non-uniquely defined vectors in the subspace belonging to the conjugate pair of eigenvalues.

Remark 14. (Depth of Curvilinear Coordinates) One may wonder about the significance of the depth of curvilinear coordinates chosen, i.e. the number of derivatives employed. As long as the first k eigenvectors are

of larger magnitude than the subsequent ones, then the subspace spanned by the first k derivatives will be asymptotically dominating over the remaining subspace, and thus the detailed choices of subsequent basis elements are insignificant as long as the basis matrix remains well-conditioned.

Definition 15. (QR Preconditioning) We choose $C_{l,n+1}$ to be the matrix Q of the QR factorization of the matrix obtained by sorting the columns of C_{n+1}^* by size in descending order.

So the matrix $C_{l,n+1}$ is chosen in the same fashion as originally proposed by Lohner [9], [10], [8], [11], [12], [13]. Different from his algorithm, also the Taylor model describing the linear and nonlinear parts of the motion is expressed in this coordinate system. This entails that the coefficients of this polynomial are subjected to smaller coordinate transformations, which leads to reduced roundoff errors. And of course, the transformations relating initial and final conditions are not merely linear, but nonlinear.

Like the curvilinear preconditioning method, the QR preconditioning leads to a coordinate system that is orthogonal, and thus the transformation in and out of this system is computationally benign because of the favorable condition number of the system. However, there are more similarities between curvilinear preconditioning and QR preconditioning:

Proposition 16. (QR Coordinates for Autonomous Linear Systems) Let $x' = A \cdot x$ be an n-dimensional linear system that has n distinct nonzero eigenvalues λ_i with eigenvectors a_i. Let b_i be the natural coordinate system of the linear system and c_i the basis vectors of the QR coordinate system. Then we have

$$c_i \to b_i \text{ for all } i \text{ as } t \to \infty.$$

The proof follows from the arguments developed in the work of Nedialkov and Jackson [18]. As a consequence, we obtain that for the important case of linear autonomous systems, the asymptotic behavior of the QR method and the curvilinear method are identical.

To illustrate the performance of the curvilinear (CV) and QR preconditioning, both of which provide orthogonal coordinate systems in which the motion is studied, let us consider the example of the simple linear ODE $x_1' = x_1$, $x_2' = x_1$. It has distinct eigenvalues ± 1 , and the eigenvector belonging to the larger eigenvalue $+1$ is $(1, 1)$, thus asymptotically, the motion is "pulled" towards this eigenvector. Figure 1 shows that in the CV preconditioning, one of the coordinate axes is attached to the direction of motion, and thus the axis will eventually line up with the vector $(1, 1)$. In the case of
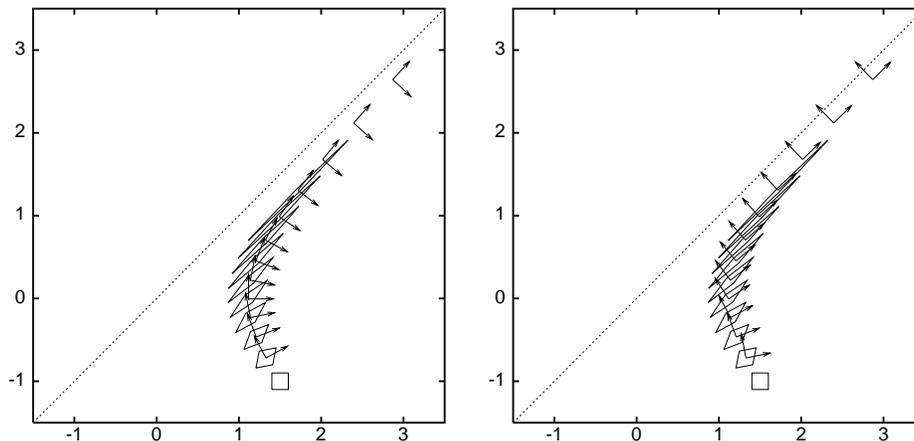
Figure 1: Preconditioning coordinate systems for the ODE $x'_1 = x_2$, $x'_2 = x_1$. Left: curvilinear, right: QR

the QR preconditioning, where one of the vectors is always attached to the longer domain box, the motion is less regular but leads to the same asymptotic behavior, since eventually also the direction of main elongation of the solution set aligns itself with the direction of motion.

For the purpose of a nonlinear example, we use the Volterra ODE and initial conditions studied in [17]. Figures 2 and 3 shows the coordinate systems for the case of the curvilinear preconditioning and the QR preconditioning, respectively. The curvilinear coordinate system performs a full rotation by 2π upon return to the initial condition by virtue of the fact that after one full period, necessarily also the direction of the tangent to the orbit is reproduced exactly. The coordinate system used by the QR method is less regular, and it can be seen that after one revolution of the center point, the coordinate system is not rotated by 2π. The long-term success of the QR method rests on the ability to asymptotically produce rotations by 2π for each revolution of the reference point, since any persistent lag in angle will produce linear wrapping. In nonlinear systems, it is not a priori clear that this condition must always be satisfied.
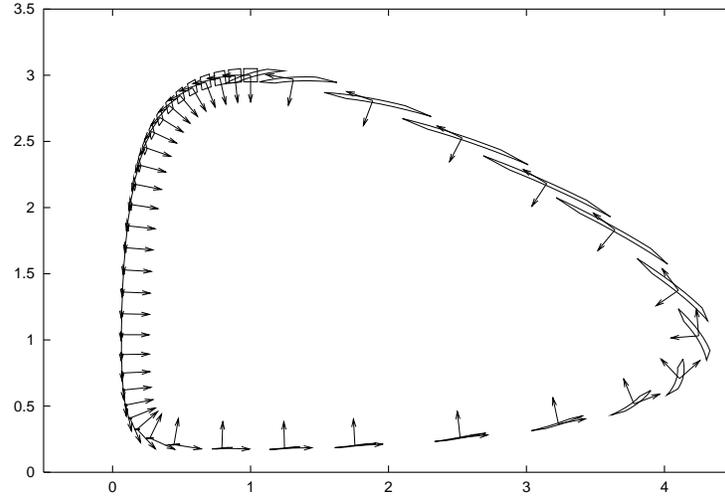
Figure 2: Preconditioning coordinate systems for the Volterra equations by the curvilinear method.
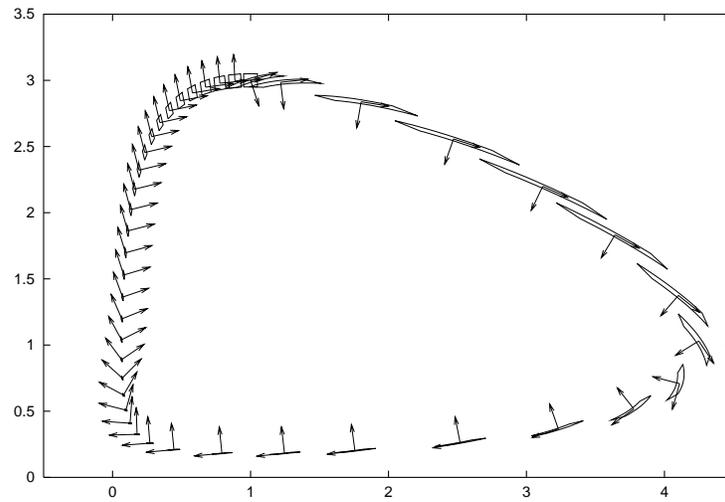
Figure 3: Preconditioning coordinate systems for the Volterra equations by the QR method.

3. Example: The Performance of Preconditioning

In this section we will study various aspects of the performance of the preconditioning method for the Roessler equation

$$x' = -(y + z)$$
$$y' = x + 0.2 \cdot y$$
$$z' = 0.2 + z \cdot (x - a)$$

for the specific value $a = 5.7$, for which numerical simulation suggests the existence of a strange attractor. We consider the integration of the relatively large cube

$$(x, y, z) = (0, -8.38095, 0.0295902) + [-0.2, 0.2]^3$$

for various choices of the integration order for approximately one full revolution, which is sufficient to assess the dynamical behavior of the system and corresponds to integration from $t = 0$ to about $t = 6$. We employ automatic step size control to minimize the total error. Figure 4 illustrates the enclosures of the $y - z$ projections of the flow obtained by Lohner's code AWA (top) and by COSY-VI (bottom). The top picture shows rapid inflation of the box enclosure, resulting in breakdown of the method in a short time. The bottom picture shows center cuts in x direction through the three dimensional Taylor models, as well as box enclosure of the full three dimensional Taylor model. It can be seen that AWA produces overestimation rather quickly, but COSY-VI can successfully integrate even rather initial large boxes. Specifically, COSY-VI can transport boxes of a volume more than 1000 times larger than what AWA is able to handle.

Figure 5 shows the total number of steps for the integration of the cube $(x, y, z) = (0, -8.38095, 0.0295902) + [-0.1, 0.1]^3$. For orders beyond ten, the integration can be carried out in about 100 steps. Next we study the required CPU time. Figure 6 shows the CPU time required as a function of order. To avoid very small step sizes, the minimum step size was set to be 0.0001. For low orders the CPU time is large, but it is even suppressed because of the minimum step size requirement, then the CPU time reaches a minimum of about 3 sec for order 8. For higher orders, the CPU time increases because of the increased complexity of the underlying Taylor model operations. Finally we address the resulting accuracy, measured in terms of the width of the resulting remainder bound at $t = 6$. Figure 7 shows the width of the remainder bound interval of the Taylor model for the flow of
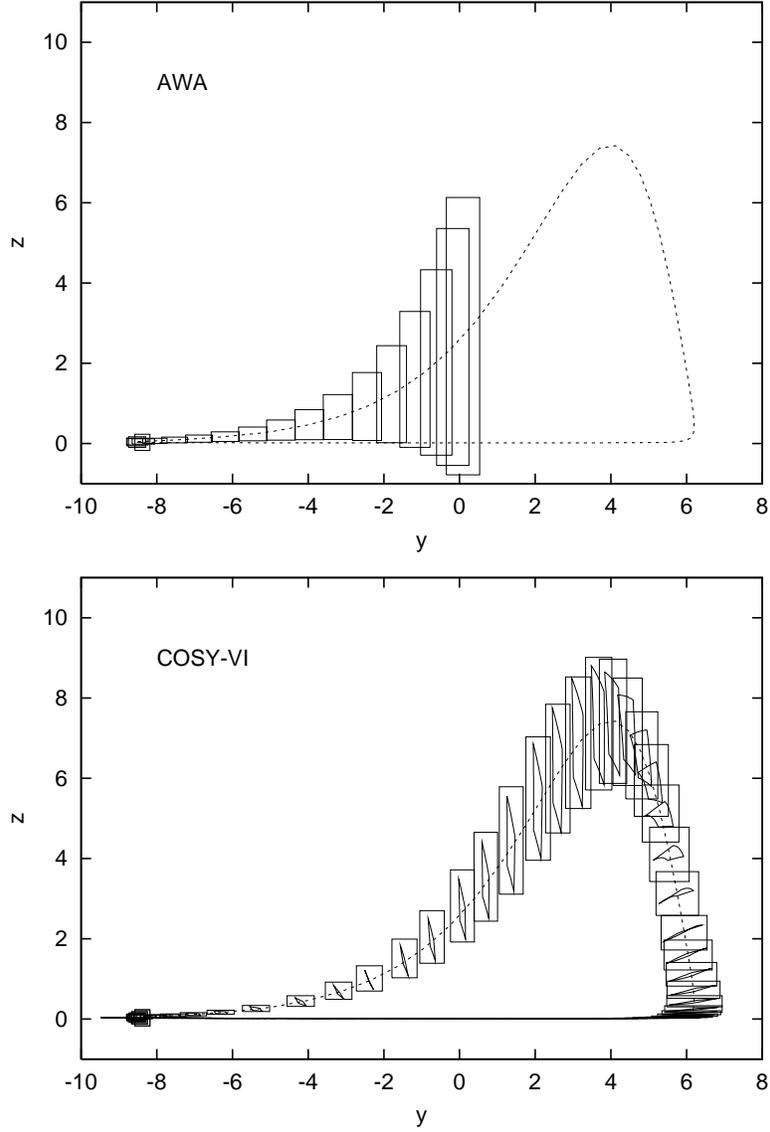
Figure 4: Projection into the $y - z$ plane for enclosures for the flow of an initial box of width 0.4 by AWA (top) and COSY-VI (bottom).
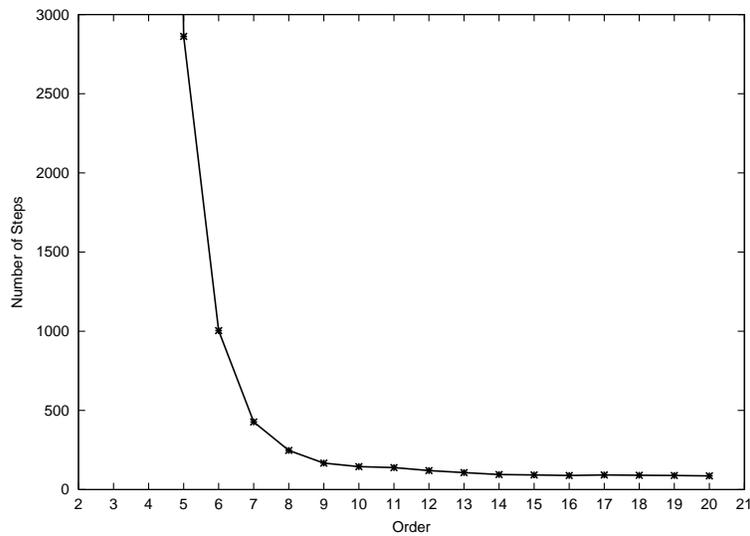
Figure 5: Total number of steps chosen by the automatic step size controller of COSY-VI as a function of order for the verified integration of a cube of width 0.2 through the Roessler equations.

the cube of width 0.2 at $t = 0.6$. The error decreases with order, and from about order 15 reaches a level below 10^{-10}, a very small fraction of the size of the cube. The comparison of figures 6 and 7 allows balancing speed versus accuracy for the specific requirements of the integration.
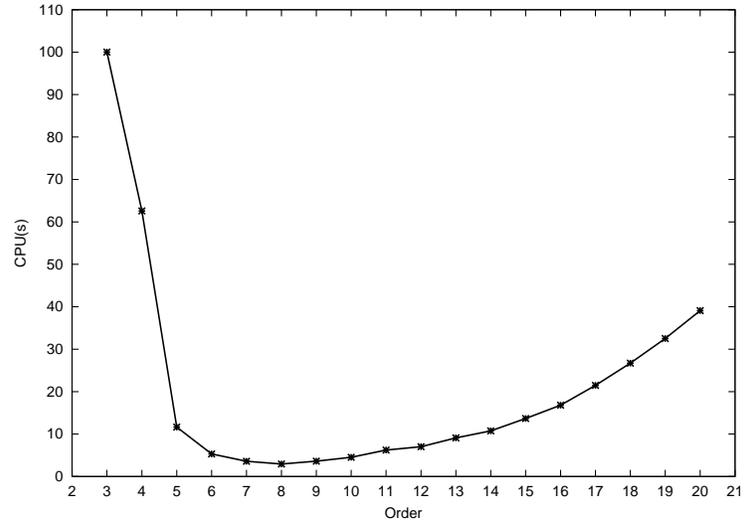
Figure 6: Total CPU time required on a 2 GHz Pentium III processor for integration of a cube of width 0.2 for one revolution through the Roessler equations.
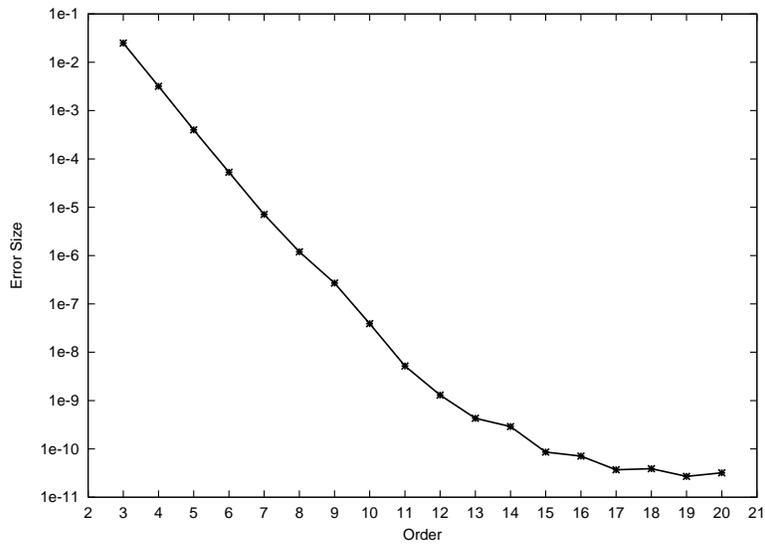
Figure 7: Error after one revolution of the verified integration of a cube of width 0.2 through the Roessler equation.

4. Linear Autonomous Examples

In this section, we will address the behavior of linear problems that may become ill-conditioned and forgo the study of nonlinear effects. Because linear problems lead to a merely linear dependence on initial conditions, they thus allow a clear separation of the effects of the Taylor model methods that are due to the expansion in initial conditions and those of their asymptotic behavior. We consider both autonomous problems, the asymptotic behavior of which can apparently also be studied more efficiently with verified eigenvalue/eigenvector tools, as well as a specific case of a non-autonomous problems. Both of these cases allow to devise certain challenges for verified integrators, and thus represent a *sine qua non.*

We begin the analysis of the behavior of the various methods by studying discrete dynamics of iteration through two-dimensional matrices. To minimize the influence of particular choice, we consider a collection of 1000 matrices with coefficients randomly chosen in the interval $[-1, 1]$. The initial condition under study is chosen to be $(1, 1) + d \cdot [-1, 1]$ with a value of $d = 10^{-3}$. Apparently the choice of the center point of the domain box is rather immaterial due to the randomness of the matrices; and because of linearity, the value of d is of importance only relative to the floor of precision of the floating precision environment.

In all cases, we study the development of the area of enclosure as a measure of the sharpness of the method. We compare preconditioning the Taylor models by the blunted method (TMB), the parallelepiped method (TMP), and the QR method (TMQ). In this linear scenario, the TMB method also describes the effects of the blunted shrink wrapping method discussed in [5], which in this case also reduces to sending the remainder term through the blunted linear matrix. We chose the blunting factors q_i to be 10^{-3} times the length of the longest column vector of the linear matrix. In order to provide a frame of reference, we also study the performance of naive interval (IN) method as well as the naive Taylor model method (TMN); in the latter case, the area is estimated as the sum of the determinant of the linear part plus the area of the remainder bound interval box. In addition, in order to provide an assessment of the influence of the effects of the underlying floating point arithmetic, we also perform a non-verified tracking of the vectors of the four corner points $(1, 1) + d \cdot (\pm 1, \pm 1)$ and determine the area of the linear structure spanned by the vectors; this method is referred to as the vector method (VE). Since this method is naturally inaccurate in particular for strongly elongated structures, we average over a large number of matrices

to control statistical fluctuations.

In the first test, we study an autonomous problem for 500 iterations. Apparently in this case, the true solution of the problem shows an exponential shrinkage of the area by the product $|\lambda_1| \cdot |\lambda_2|$ of the magnitudes of the eigenvalues. For the purpose of analysis, we group the matrices in six categories; the category C_1 contains all matrices in which the eigenvalues form conjugate pairs. The other matrices are sorted into categories based on the ratios $r = |\lambda_1|/|\lambda_2|$ of the eigenvalue λ_1 of larger magnitude to the one of smaller magnitude. Specifically we consider the categories C_2 with $1 \leq r < 5$, C_3 with $5 \leq r < 10$, C_4 with $10 \leq r < 20$, C_5 with $20 \leq r < 50$, and C_6 with $50 \leq r$. The numbers of matrices in categories C_1 through C_6 are 325, 520, 80, 40, 18, and 17. Within each category, we calculate the average of the logarithm of the areas enclosed by the various methods as a function of the iteration number, which for the true dynamics would lead to a decrease along a straight line, the slope of which is given by the value $\log\left(|\lambda_1| \cdot |\lambda_2|\right)$.

Figure 8 shows the results of the situation for categories C_1 and C_2. It is clearly visible that in the dynamics of C_1, the behavior is characterized by the expected linear decrease, and the blunted (TMB), parallelepiped (TMP), and QR method (TMQ) all show this behavior. All three of these methods very closely follow the non-verified result (VE), with a closer inspection showing that the TMB and TMP methods provide enclosures about 1 to 2 orders of magnitude sharper than the TMQ method. The behavior of the methods is in agreement with the theoretical results and practical examples found in [18]. On the other hand, the naive interval method (IN) as well as the naive Taylor model method (TMN) show a qualitatively different behavior; the interval method leads to a different slope, while over the short term the naive Taylor model method performs similar to the other methods until the size of the remainder bound becomes the dominating contribution, at which time its slope becomes similar to that of the interval method.

Studying the behavior of the class C_2 shows a similar pattern, except that now the TMB and TMQ methods provide indistinguishable sharpness, while the parallelepiped method now performs markedly worse. This is due to the unfavorable conditioning of the TMP approach that does not appear in the TMQ approach.

Studying the behavior in the classes C_3 and C_4 shown in figure 9 reveals again that the TMB and TMQ methods perform virtually indistinguishable, and both of them follow the non-verified result VE very closely. Furthermore, the naive Taylor model method TMN and the parallelepiped method TMP

both perform quite similar to each other, but substantially worse than the TMB and TMQ methods.

However, another interesting effect appears. We notice that there is a marked change in the slope of the curve after somewhere around $n = 20$ iterations for the C_3 case and $n = 15$ iterations for the C_4 case. This is attributed to the fact that after this number of iterations, the quantity r^n reaches around 10^{17}, and thus the ratio of the elongations of the solution domain in the directions of the eigenvectors v_1 and v_2 reaches the limit of what can be represented in a double precision floating point environment. Before this value of n, the computed volume decreased by $\lambda_1 \cdot \lambda_2$ at each iteration, but after this n, the apparent "thickness" of the needle-like structure will be determined by the floating point accuracy ε times the length of the needle. Thus any decrease in volume is merely due to the decrease of the needle's length, which is governed by the eigenvalue of smaller magnitude λ_2, and so the subsequent volume decrease is given by λ_2^2. Thus one is bound to observe a jump in slope of about the factor r.

Thus we observe that in the process of floating point errors, the long-term behavior of the area is predicted qualitatively wrong, and thus does not follow the predictions of [18] for the infinite precision case anymore. However, it is most noticeable that this effect does not only appear within the verified setting, but just in the same way in the non-verified case. In the latter case, the perceived "thickness" of the needle is merely given by floating point rounding errors that prevent the four corner points from being collinear, where again the deviation from collinearity being given by the ε times the length of the respective vectors, which leads to a perceived area very similar to that in the verified case. This observation appears most important, since it stresses that the spurious exponential growth observed compared to the true result is an unavoidable consequence of the floating point environment per se and has nothing to do with the attempt to do verified computation.

The situation for the cases of C_5 and C_6 are similar to those of the C_3 and C_4 cases, except that as expected the change in slopes appears earlier and is more pronounced.
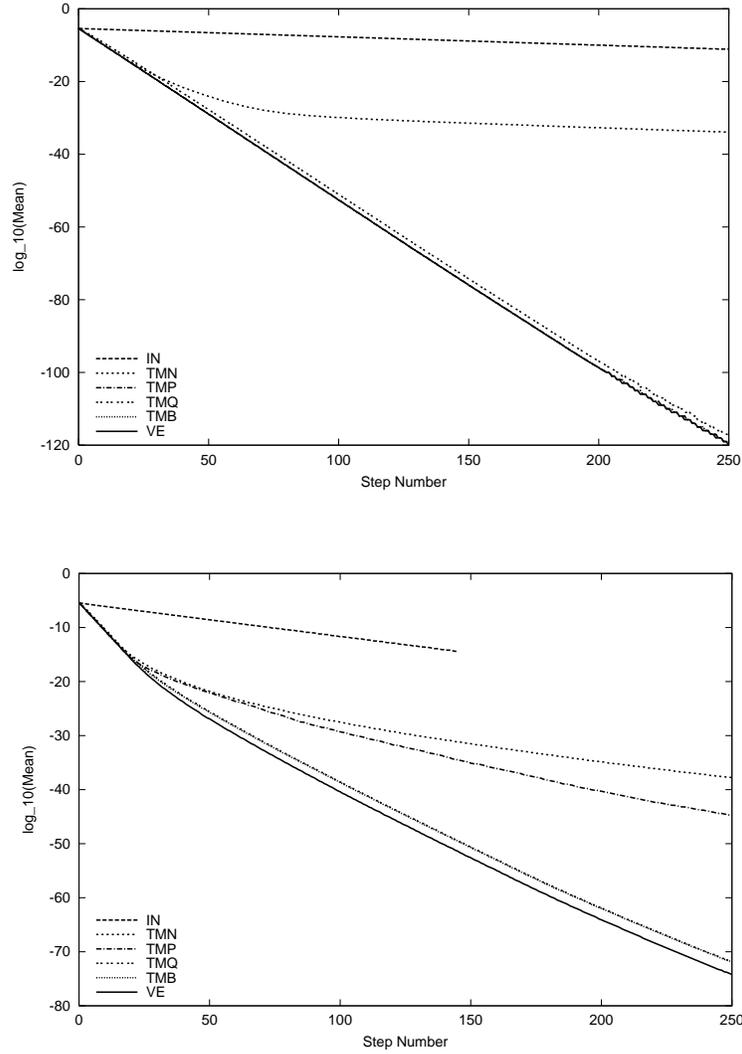
Figure 8: Areas predicted in the iteration through random 2×2 matrices with conjugate eigenvalues (top) and eigenvalues differing in magnitude by a factor of 1 to 5 (bottom) for various enclosure methods.
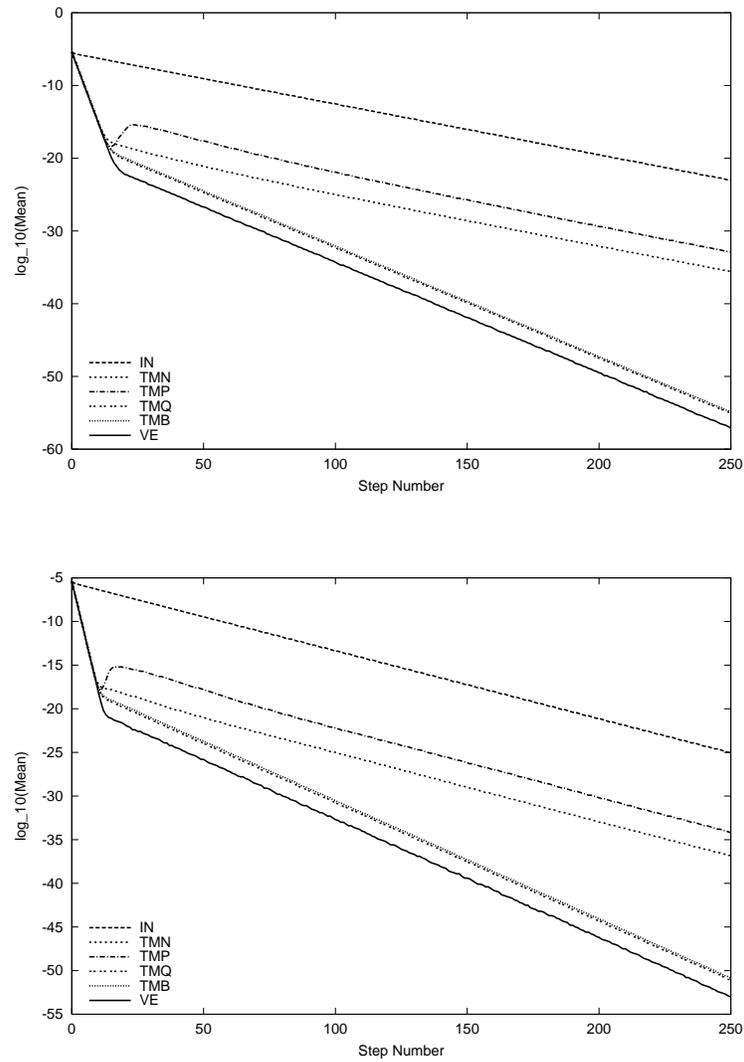
Figure 9: Areas predicted in the iteration through random 2×2 matrices with eigenvalues differing in magnitude by a factor between 5 and 10 (top) and between 10 and 20 (bottom) for various enclosure methods.

5. Linear Non-Autonomous Examples

As another set of test cases, we want to perform a limited study of non-autonomous linear systems. This case is interesting because the quantitative analysis of the behavior of the QR methods undertaken in [18] does not hold in this case, and as already observed by Kühn, spurious exponential error growth is possible; thus a comparison to the TMB method is worthwhile.

For the purposes of our non-autonomous study, we merely iterate through the 1000 random matrices for 10 iterations, and follow these by iterating through the approximate floating point inverses of the respective matrices for the next 10 iterations, repeating this procedure a total of 25 times. The overall transformation reaches the identity after each 20 steps, and thus the analysis of the performance is straightforward. Figure 10 shows the behavior of the various methods for the case of conjugate eigenvalues and the case $1 \leq q < 5$; it is clearly seen that the naive interval (IN) and the naive Taylor model (TMN) methods lead to overestimation rather quickly. For the purpose of better readability, in figure 11 we show the enclosure area only after every 20 steps, at which point the overall transformation reaches identity. It can be seen that the TMB (and the TMP) methods reproduce the correct result to printer resolution, while the TMQ method reaches an overestimation of two orders of magnitude. For the case $1 \leq r < 5$, which is more favorable to the QR approach, again the TMB (and the TMP) method produce very little overestimation, while the TMQ method has about one order of magnitude of overestimation. For larger values of r, the advantage of the TMB method becomes less pronounced but is still one order of magnitude, while the TMP method begins to produce larger overestimations, as can be seen in figures 12 for the cases of $5 \leq r < 10$ and $10 \leq r < 20$.
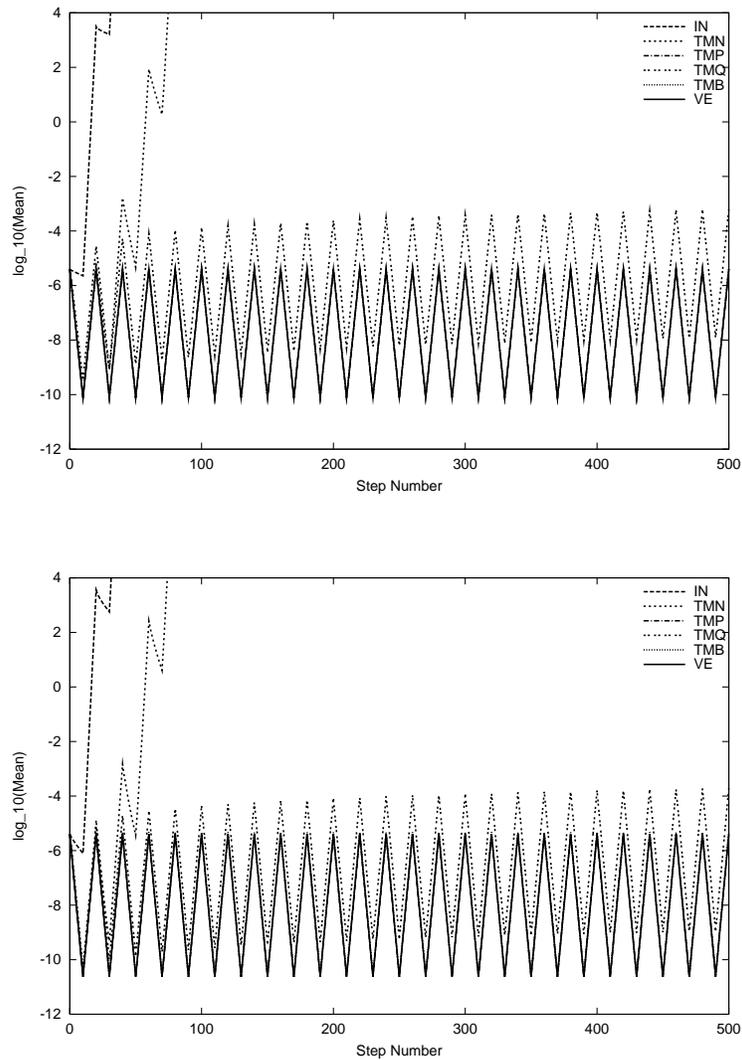
Figure 10: Predicted areas for 10 forward and 10 backward iterations through random 2×2 matrices with conjugate eigenvalues (top) and eigenvalues differing in magnitude by a factor of 1 to 5 (bottom) for various enclosure methods.
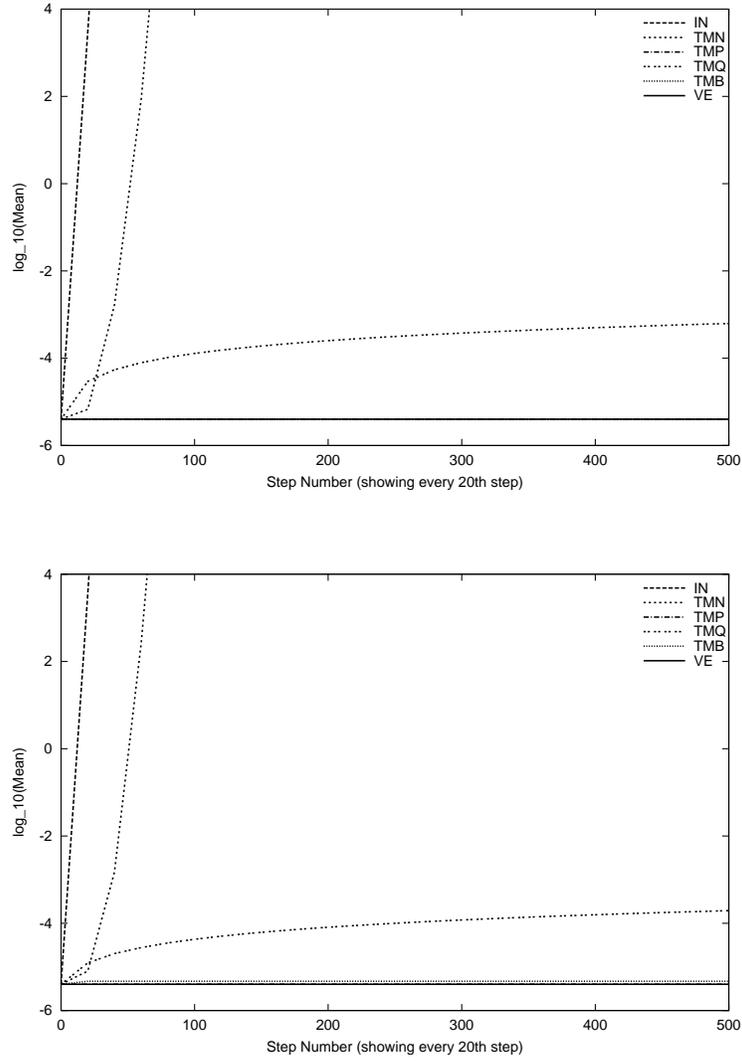
Figure 11: Predicted areas for groups of 10 forward and 10 backward iterations through random 2×2 matrices with conjugate eigenvalues (top) and eigenvalues differing in magnitude by a factor between 1 and 5 (bottom) for various enclosure methods. Results shown after each set of 20 steps.
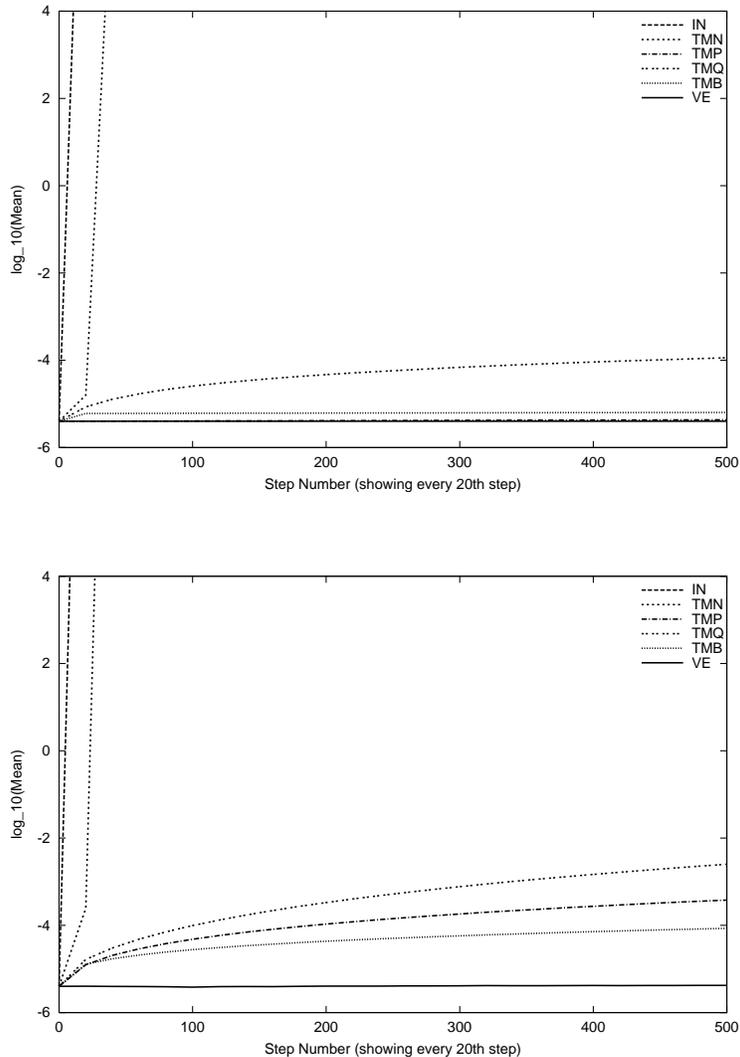
Figure 12: Predicted areas for groups of 10 forward and 10 backward iterations through random 2×2 matrices with eigenvalues differing in magnitude by a factor between 5 and 10 (top) and 20 and 50 (bottom) for various enclosure methods. Results shown after each set of 20 steps. The blunted method (TMB) outperforms the QR method.

6. Linear Continuous Examples

As another example for the use of preconditioning tools for linear problems, we study some continuous problems and compare the behavior of the QR preconditioning method with the curvilinear (CV) preconditioning methods. We study an ensemble of 4x4 matrices with random elements in $[-1, 1]$, and determine verified solutions of the linear homogeneous ODE

$$r' = A \cdot r$$

over the time domain $[0, 10]$ for initial domain box $r + [-.1, .1]^4$ where r is a vector with random number entries.

We study the details of the situation for one particular matrix A_1 of the form

$$A_1 = \begin{pmatrix} +0.9564 & +0.2004 & +0.4826 & +0.8871 \\ -0.4922 & +0.5651 & -0.1474 & -0.7678 \\ -0.0269 & -0.8587 & -0.3785 & -0.6168 \\ -0.8271 & +0.2661 & -0.9380 & +0.5289 \end{pmatrix}$$

and approximate eigenvalues 0.3928, -0.3911, $1.005 \pm 0.8669i$ as well as for a set of 10 random matrices. The matrix A_1 was selected because it has positive, negative, and complex conjugate eigenvalues, and the complex conjugate pair is even dominating in magnitude. The random center point of the initial domain box was approximately $(0.6446, 0, 0050, -0.2394, 0.4581)$. The other matrices were studied to give confidence that what is observed is not an isolated case.

The top picture in figure 13 shows the effects of QR and CV preconditioning for the specific case A_1. Note that both remainder estimates are increasing exponentially, which is due to the fact that the magnitude of the leading eigenvalues, those that form the complex conjugate pair, exceeds unity. Apparently the two methods behave very similarly, where in the very beginning the QR preconditioning provides results that are sharper by about a factor of 2. Note that there is an oscillatory pattern visible, which is due to the fact that two of the four eigenvalues of the matrix form a complex conjugate pair, resulting in some oscillatory motion in one of the invariant subspaces of the matrix.

An attempt of a quantitative analysis of the figure shows that after the initial period of rapid error growth, which is due to the proximity of the floating point accuracy floor, the function rises exponentially from 10^{-11} at $t = 3$ to 10^{-7} at $t = 10$, which corresponds to a gain of $10^{4/7} \approx 10^{.5715}$ per time unit. On the other hand, the magnitude of the complex eigenvalue is

approximately 1.327, leading to a gain of $\exp(1.327) \approx 3.769 \approx 10^{0.5763}$ per time unit. So we see that to very good approximation, the growth in the remainder error matches the growth of the parallelogram enclosing the flow of the initial domain box or the corner points thereof, which is the behavior observed in a non-verified integrator.

To conclude the discussion of linear the study of linear problems with preconditioning, we summarize the observed behavior of the methods:

1. For iteration through identical matrices, which corresponds to study of autonomous systems, the Blunted Method and the QR method have the same asymptotic behavior and error growth as the non-verified method. On the other hand, the naive interval method, the parallelepiped method, and the naive Taylor model produce overestimations that grow exponentially.

2. For iteration through sets of matrices and their inverses, which corresponds to a periodic non-autonomous system, the blunted and the parallelepiped methods perform superior to the QR method, which in turn is superior to the naive interval and naive Taylor model methods.
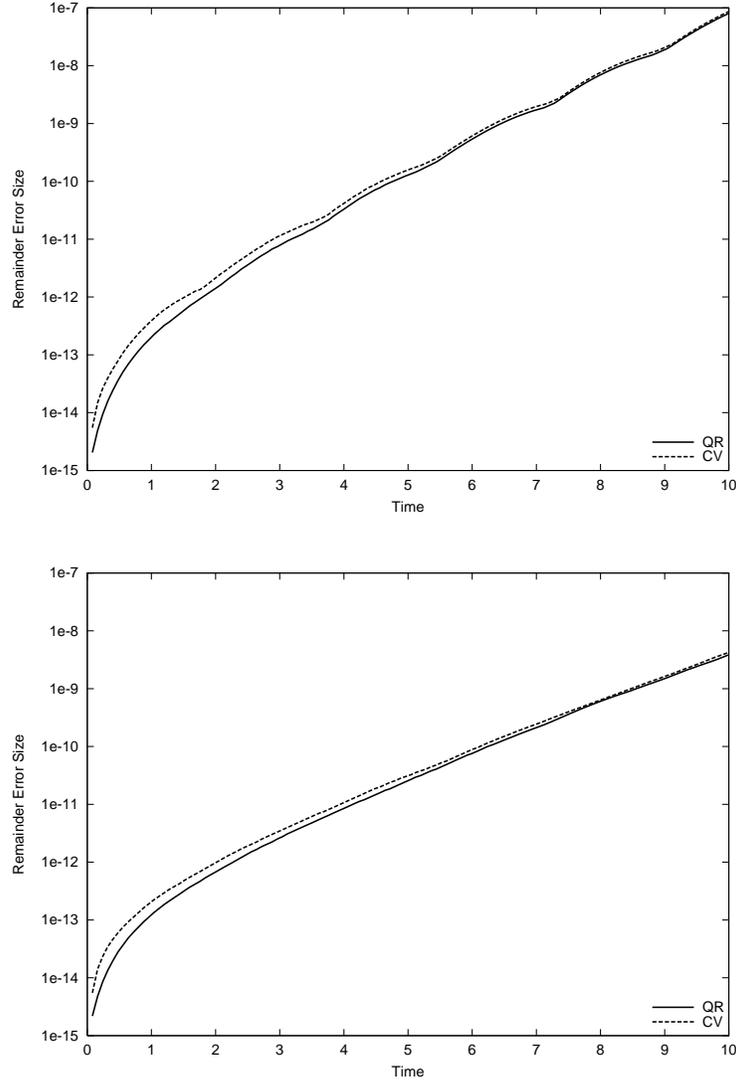
Figure 13: The size of the interval remainder errors for 4×4 linear systems as determined using QR and CV preconditioning. Averages of the errors of the four components for the matrix A_1 (top) and averages of those of 10 random matrices (bottom).

7. Example: A Muon Cooling Ring

In this section we study a problem from the field of beam physics and illustrate the use of curvilinear coordinates. We use a simple model of a muon cooling ring, the purpose of which is to reduce the size of a beam by passing it through material and simultaneously accelerating it. Specifically, the particles are held in a confined orbit by a homogenous magnetic field in vertical direction; for reasons of simplicity, we restrict the dynamics to lie only in the plane. The coordinates describing the motion are the Euclidean x and y, and the corresponding momenta p_x and p_y.

The particles are moving in homogenous matter, which provides a deceleration force of magnitude α along their direction of motion; the direction of motion is given by $(p_x, p_y)/\sqrt{p_x^2 + p_y^2}$. Furthermore, there is an azimuthal acceleration force of equal magnitude α and opposite direction. For particles at coordinates (x, y), the azimuthal direction is given by $(y, -x)/\sqrt{x^2 + y^2}$. Altogether, the equations of motion are

$$
\begin{aligned}
\dot{x} &= p_x \\
\dot{y} &= p_y \\
\dot{p}_x &= p_y - \frac{\alpha}{\sqrt{p_x^2 + p_y^2}} \cdot p_x + \frac{\alpha}{\sqrt{x^2 + y^2}} \cdot y \\
\dot{p}_y &= -p_x - \frac{\alpha}{\sqrt{p_x^2 + p_y^2}} \cdot p_y - \frac{\alpha}{\sqrt{x^2 + y^2}} \cdot x
\end{aligned}
\tag{2}
$$

It can be easily verified that the system has an invariant solution

$$
(x, y, p_x, p_y)_I(t) = (\cos t, -\sin t, -\sin t, -\cos t),
$$

which represents a clockwise rotation in the horizontal plane with constant radius 1 and constant momentum 1. The practically significant property of the system is that acceleration always happens azimuthally, while deceleration happens in the direction of motion; this leads to a decrease of the radial component of the momentum, and mathematically to the fact that solutions of the ODE asymptotically approach circular motion of the form

$$
(x, y, p_x, p_y)_a(t) = (\cos(t - \phi), -\sin(t - \phi), -\sin(t - \phi), -\cos(t - \phi)),
$$

where ϕ is a characteristic angle of the particle in question. For practical applications, this is eminently useful, as it reduces the volume of the four

dimensional space of values (x, y, p_x, p_y), an effect known as cooling. While in practice, many technical details have to be considered, the simple ODE (2) represents the essence of this process.

For the purpose of using the problem as a test case for verified integration, the following aspects are important

1. It is important to treat a large initial domain box of a range of $[-10^{-2}, 10^{-2}]^4$. This will entail the presence of rather strong nonlinearities.

2. Because of the transversal damping action towards the invariant limit cycle, the linear part of the motion will be more and more ill-conditioned.

We study the dynamics using COSY-VI using curvilinear preconditioning, which is standard in beam physics simulations (see for example [14], [1] and references therein). We perform the integration until no further reduction in phase space can be performed due to the proximity of the floating point floor. Figure 14 shows the effects of cooling for domain boxes $(0, 1, 1, 0) + [-d, d]^4$ for $d = 10^{-2}$, 10^{-4}, and 10^{-6}; the value of 10^{-2} approximately corresponds to the practical needs.

Studying the magnitude of the determinants of the linear part, which roughly correspond to the volume, we see that cooling happens exponentially and with nearly the same speed in all three cases. The final volume that is attained is larger for the larger initial volumes, which is due to the fact that volume gets compressed only transversely to the motion of the beam, while nothing affects the particle's longitudinally motion. Thus for larger initial boxes, the final box in the direction of motion will be larger.

As a result, we obtain a very narrow elongated structure with nearly vanishing radial thickness that rotates around a circle. As a consequence, the condition number of the linear part becomes larger and larger, as shown on the right of figure 14. If not treated properly, computationally this may represent a significant challenge, but as expected, the curvilinear preconditioning can overcome this difficulty. To study the motion in detail, we look at the remainder bounds of the dynamics, which are shown in figure 15. Overall, we see that COSY-VI has no difficulty performing the integration of the muon damping system with $d = 10^{-2}$ for ten revolutions, which is sufficient to perform the required damping task. On the other hand, the linear code AWA can only succeed with this task for $d = 10^{-4}$. Thus a full simulation of the necessary space of initial conditions, which can be performed with one run of COSY-VI, would require approximately $\left(10^2\right)^4 = 10^8$ runs of AWA.
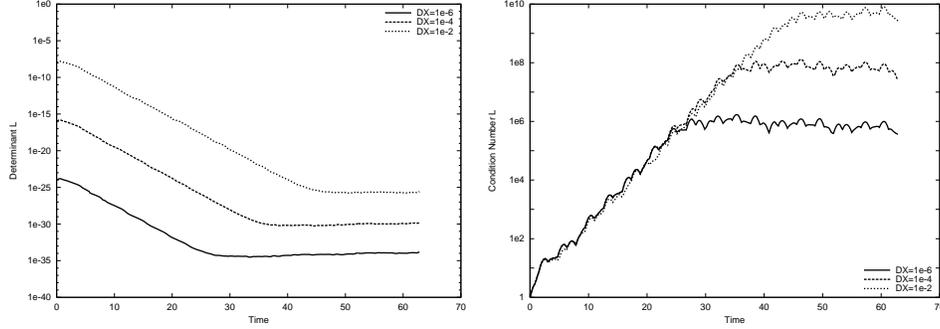
Figure 14: Simulation of the Muon Cooling Ring for initial condition boxes $(0, 1, 1, 0) + [-d, d]^4$ located at the upper top for $d = 10^{-2}$, 10^{-4}, 10^{-6}. Shown are the determinants of the linear part (left) indicating progress in cooling, and the condition numbers of the linear part (right).
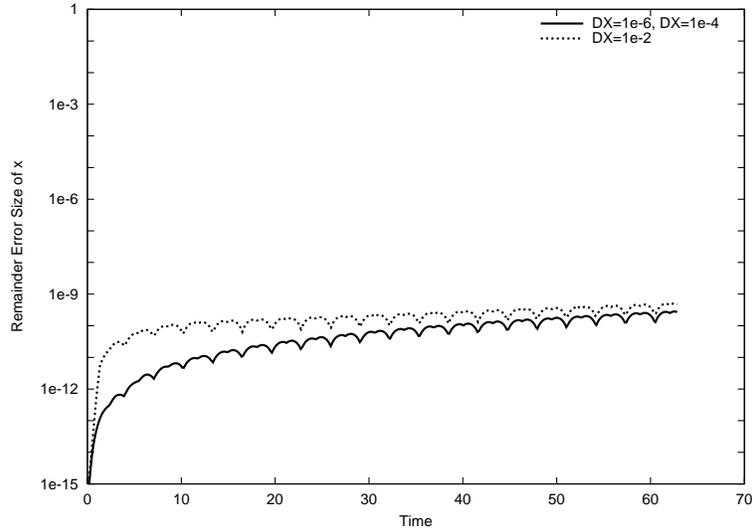
Figure 15: Remainder bound sizes for the simulation of the Muon Cooling Ring for initial condition boxes $(0, 1, 1, 0) + [-d, d]^4$ for $d = 10^{-2}$, 10^{-4}, 10^{-6}.

8. Example: The Discrete 2D Circular Kepler Problem

This system describes the dynamics of circular Kepler orbits around a central mass in terms of the variables (x, y) in the plane of the motion. It is well known from Kepler's third law that the periods T and large semi-major axes a of a Kepler ellipse are related via $T^2 = k \cdot a^3$, where k is determined by the mass of the central object. For circular orbits of radius r, for $k = 1$ this entails an angular velocity of $\omega = 2\pi/T = 2\pi \cdot r^{-3/2}$, which means that the transformation by a fixed time step Δt is given by the two-dimensional transformation

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} \cos \Delta\phi & \sin \Delta\phi \\ -\sin \Delta\phi & \cos \Delta\phi \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix}$$
$$\text{where } \Delta\phi = \frac{2\pi \Delta t}{(x^2 + y^2)^{3/4}}.$$

While addressing only circular motion, the dynamics is also quite characteristic of the general motion of Kepler ellipses because it captures one of the main effects: as time progresses, there is a larger and larger lag between the circles of different radii r. This lag makes Taylor expansion of final condition in terms of initial conditions impossible for sufficiently large times, and thus represents a challenge for all Taylor-based methods that will necessarily lead to their eventual failure. The interest in the problem now lies in the attempt to delay failure.

Figure 16 show the remainder bounds of the study of the dynamics without shrink wrapping for repeated application of the discrete transformation with $\phi_0 = \pi/4$, in which case one full revolution, or one cycle, consists of eight applications of the individual map.

We study three cases: as a reference we use first order Taylor models preconditioned by curvilinear coordinates, which behave similar to the PEQR method. We compare with tenth order Taylor models preconditioned by curvilinear coordinates, and tenth order Taylor models preconditioned by the QR method. The growth of the remainder bounds is shown for four different initial domain widths of 10^{-6}, 10^{-8}, 10^{-10}, and 10^{-12} as a function of full cycles of 2π. It can be seen that for each case, the tenth order Taylor model method survives between 7 and 10 times longer than the first order method. Furthermore, the preconditioning by curvilinear coordinates leads to a slightly better performance, which is attributed to the fact that the movement of the coordinate system is smoother since it follows the reference
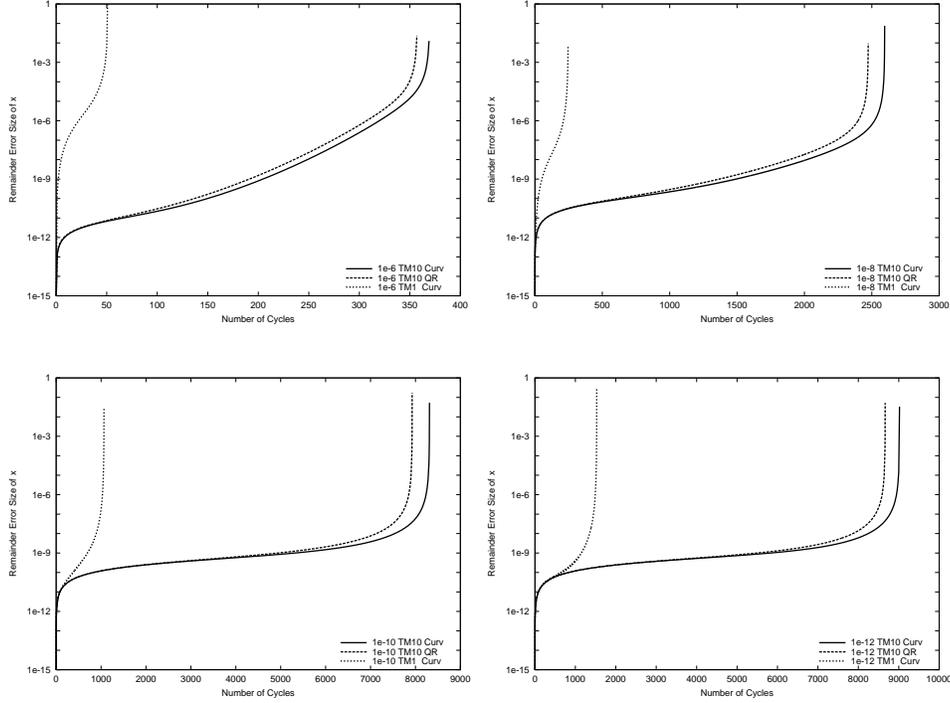
Figure 16: Dynamics in the discrete 2D Kepler system for initial box sizes widths of 10^{-6} (top left), 10^{-8} (top right), 10^{-10} (bottom left) and 10^{-12} (bottom right). Shown are the remainders obtained by the first and tenth order Taylor method using Curvilinear Preconditioning and QR preconditioning.

orbit instead of the somewhat more random orientation of the longest edge.

It is also interesting to estimate the growth rate of the remainder bounds in the high-order TM methods. An inspection of the bottom right picture in figure 16 reveals that during revolutions 1000 and 6000, the remainder width increases from about 10^{-10} to about 10^{-9}, for a total increase of $9 \cdot 10^{-9}$ over $5,000$ revolutions or $40,000$ iterations. This corresponds to about $2 \cdot 10^{-13}$ per map iteration; considering that each iteration requires several function evaluations, and that in our current implementation, intrinsic functions carry an overestimation of around 10 ulps, this number is very close to the unavoidable consequences of accounting for the mere floating point errors of the arithmetic involving the constant part of the Taylor model.

Acknowledgement

References

[1] M. Berz. *Modern Map Methods in Particle Beam Physics*. Academic Press, San Diego, 1999. Also available at http://bt.pa.msu.edu/pub.

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

[3] M. Berz and K. Makino. Preservation of canonical structure in non-planar curvilinear coordinates. *International Journal of Applied Mathematics*, 3,4:401–419, 2000.

[4] M. Berz and K. Makino. Performance of Taylor model methods for validated integration of ODEs. In *Lecture Notes in Computer Science*, volume 3732, pages 65–74, 2005.

[5] M. Berz and K. Makino. Suppression of the wrapping effect by Taylor model- based verified integrators: Long-term stabilization by shrink wrapping. *International Journal of Differential Equations and Applications*, 2006.

[6] B. Erdélyi and M. Berz. Optimal symplectic approximation of Hamiltonian flows. *Physical Review Letters*, 87,11:114302, 2001.

[7] B. Erdélyi and M. Berz. Local theory and applications of extended generating functions. *International Journal of Pure and Applied Mathematics*, 11,3:241–282, 2004.

[8] R. J. Lohner. AWA - Software for the computation of guaranteed bounds for solutions of ordinary initial value problems.

[9] R. J. Lohner. Enclosing the solutions of ordinary initial and boundary value problems. In E. Kaucher, U. Kulisch, and C. Ullrich, editors, *Computer Arithmetic: Scientific Computation and Programming Languages*, pages 255–286. Teubner, Stuttgart, 1987.

[10] R. J. Lohner. *Einschliessung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen.* Dissertation, Fakultät für Mathematik, Universität Karlsruhe, 1988.

[11] R. J. Lohner. *Einschliessungen bei Anfangs- und Randwertaufgaben gewohnlicher Differentialgleichungen; and: Praktikum 'Einschliessungen bei Differentialgleichungen',*, pages 183–207 and 209–223. Akademie-Verlag, Berlin, 1989.

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

[13] R. J. Lohner. Step size and order control in the verifed solution of ivp with ode's. In *SciCADE '95 International Conference on Scientific Computation and Differential Equations*, pages 28.3.–1.4., Stanford, 1995.

[14] K. Makino. *Rigorous Analysis of Nonlinear Motion in Particle Accelerators.* PhD thesis, Michigan State University, East Lansing, Michigan, USA, 1998. Also MSUCL-1093.

[15] K. Makino and M. Berz. Perturbative equations of motion and differential operators in nonplanar curvilinear coordinates. *International Journal of Applied Mathematics*, 3,4:421–440, 2000.

[16] K. Makino and M. Berz. Taylor models and other validated functional inclusion methods. *International Journal of Pure and Applied Mathematics*, 6,3:239–316, 2003.

[17] K. Makino and M. Berz. Suppression of the wrapping effect by Taylor model- based verified integrators: The single step. *International Journal of Pure and Applied Mathematics*, 2006.

[18] N. S. Nedialkov and K. R. Jackson. A new perspective on the wrapping effect in interval methods for IVPs for ODEs. *Proc. SCAN2000, Kluwer*, 2001.

[19] N. Revol, K. Makino, and M. Berz. Taylor models and floating-point arithmetic: Proof that arithmetic operations are validated in COSY. *Journal of Logic and Algebraic Programming*, 64/1:135–154, 2004.