# Certifying trajectories of dynamical systems

JORIS VAN DER HOEVEN

Laboratoire d'informatique, UMR 7161 CNRS
Campus de l'École polytechnique
1, rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing, CS35003
91120 Palaiseau

*Email:* `vdhoeven@lix.polytechnique.fr`

*August 29, 2015 with latest corrections on October 6, 2018*

This paper concerns the reliable integration of dynamical systems with a focus on the computation of one specific trajectory for a given initial condition at high precision. We describe several algorithmic tricks which allow for faster parallel computations and better error estimates. We also introduce "Lagrange models". These serve a similar purpose as the more classical Taylor models, but we will show that they allow for larger step sizes, especially when the truncation orders get large.

KEYWORDS: reliable computation, dynamical systems, certified integration, ball arithmetic, Taylor models, multiple precision computations

A.M.S. SUBJECT CLASSIFICATION: 65G20, 37-04

## 1. INTRODUCTION

### Description of the problem and background

Let $\Phi \in \mathbb{C}[F_1, \ldots, F_d]^d$ be a polynomial vector field and consider the dynamical system

$$f' = \Phi(f). \tag{1}$$

Given an initial condition $f(u) = I \in \mathbb{C}^d$ at $u \in \mathbb{R}$, a target point $z > u$ such that $f$ is analytic on $[u, z]$, the topic of this paper is to compute $f(z)$.

On actual computers, this problem can only be solved at finite precisions, although the user might request the precision to be as large as needed. One high level way to formalize this is to assume that numbers in the input (i.e. the coefficients of $\Phi$ and $I$, as well as $u$ and $z$) are *computable* [27, 28] and to request $f(z)$ again to be a vector of computable complex numbers.

From a practical point of view, it is customary to perform the computations using *interval arithmetic* [18, 1, 23, 9, 11, 19, 25]. In our complex setting, we prefer to use a variant, called *ball arithmetic* or *midpoint-radius arithmetic*. In our main problem, this means that we replace our input coefficients by complex balls, and that the output again to be a vector of balls. Throughout this paper, we assume that the reader is familiar with interval and ball arithmetic. We refer to [5, 8] for basic details on ball arithmetic.

It will be convenient to denote balls using a bold font, e.g. $\boldsymbol{f}(z) \in \boldsymbol{\mathbb{C}}^d$. The explicit compact ball with center $c$ and radius $r$ will be denoted by $\mathcal{B}(c, r)$. Vector notation will also be used systematically. For instance, if $c \in \mathbb{C}^d$ and $r \in (\mathbb{R}^>)^d$ with $\mathbb{R}^> = \{x \in \mathbb{R}: x > 0\}$, then $\mathcal{B}(c, r) = (\mathcal{B}(c_1, r_1), \ldots, \mathcal{B}(c_d, r_d))$.

Sometimes, it is useful to obtain further information about the dependence of the value $f(z)$ on the initial conditions; this means that we are interested in the *flow* $f(z, I)$, which satisfies the same differential equation (1) and the initial condition $f(u, I) = I$. In particular, the *first variation* $V = \partial f / \partial I$ is an important quantity, since it measures the sensitivity of the output on the initial conditions. If $\kappa$ denotes the condition number of $V$, then it will typically be necessary to compute with a precision of at least $\log_2 \kappa$ bits in order to obtain any useful output.

There is a vast literature on the reliable integration of dynamical systems [17, 18, 22, 13, 3, 20, 15, 12, 14, 21, 16]. Until recently, most work focused on low precision, allowing for efficient implementations using machine arithmetic. For practical purposes, it is also useful to have higher order information about the flow. *Taylor models* are currently the most efficient device for performing this kind of computations [15, 16].

## Main strategy for certification of high precision trajectories

In this paper, we are interested in the time complexity of reliable integration of dynamical systems. We take a more theoretical perspective in which the working precision might become high. We are interested in certifying one particular trajectory, so we do not request any information about the flow beyond the first variation.

From this complexity point of view it is important to stress that there is a tradeoff between efficiency and quality: faster algorithms can be designed if we allow for larger radii in the output. Whenever one of these radii becomes infinite, then we say that the integration method *breaks down*: a ball with infinite radius no longer provides any useful information. Now some "radius swell" occurs structurally, as soon as the condition number of $V$ becomes large. But high quality integration methods should limit all other sources of precision loss.

The outline of this paper is as follows:

1. For problems from reliable analysis it is usually best to perform certifications at the outermost level. In our case, this means that we first compute the entire numeric trajectory with a sufficient precision, and only perform the certification at a second stage. We will see that this numeric computation is the only part of the method which is essentially sequential.

2. The problem of certifying a complete trajectory contains a global and a local part. From the global point of view, we need to cut the trajectory in smaller pieces that can be certified by local means, and then devise a method to recombine the local certificates into a global one.

3. For the local certification, we will introduce *Lagrange models*. As in the case of Taylor models, this approach is based on Taylor series expansions, but the more precise error estimates allow for larger time steps.

The first idea has been applied to many problems from reliable computation (it is for instance known as Hansen's method in the case of matrix inversion). Nevertheless, we think that progress is often possible by applying this simple idea even more systematically.

## Outline of the paper

In Section 2, we start with a quick survey of the most significant facts about numerical integration schemes for the equation (1). We also present a new parallel and dichotomic algorithm for increasing the precision of an already computed trajectory. In Section 3, we will see that a similar strategy can be used for certifying trajectories.

In Section 3 we present a method for reducing the problem of certifying a global trajectory to the local problem of certifying sufficiently short trajectories. It is interesting to compare this approach with the more classical stepwise certification scheme, along with the numerical integration itself. The problem with stepwise schemes is that they are essentially sequential and thereby give rise to a linear precision loss in the number of steps. The global approach reduces this to a logarithmic precision loss only. The global strategy already pays off in the linear case [3] and it is possible to reorganize the computations in such a way that they can be re-incorporated into an iterative scheme. The global approach was generalized to the non linear case in [5]. Our current presentation has the merit of conceptual simplicity and ease of implementation.

The main contribution of this paper concerns the introduction of "Lagrange models" and the way that such models allow for larger time steps. This material is presented in Section 4; an earlier (non refereed) version appeared in the lecture notes [8]. Classical Taylor models approximate analytic functions $f$ on the compact unit disk (say) by a polynomial $P \in \mathbb{C}[z]$ of degree $<n$ and an error $\varepsilon \geqslant 0$ with the property that $|f(z) - P(z)| \leqslant \varepsilon$ for all $|z| \leqslant 1$. The idea behind Lagrange models is to give a more precise meaning to the "big Oh" in $f(z) = f_0 + \cdots + f_{n-1} z^{n-1} + O(z^n)$. More precisely, it consists of a polynomial $\boldsymbol{P} \in \mathbb{C}[z]$ of degree $<n$ with ball coefficients and an

$\varepsilon \geqslant 0$ such that $f(z) \in \boldsymbol{P}(z) + \mathcal{B}(0, \varepsilon)\, z^n$ for all $|z| \leqslant 1$. The advantage comes from the fact that the integration operator has norm 1 for general analytic functions on the unit disk but only norm $1/(n+1)$ for analytic functions that are divisible by $z^n$. Although Lagrange model arithmetic can be a constant times more expensive than Taylor model arithmetic, the more precise error estimates allow us to increase the time step.

## 2. Fast numerical integration

### 2.1. Classical algorithms

From a high level perspective, the integration problem between two times $u < z$ can be decomposed into two parts: finding suitable intermediate points $u = z_0 < z_1 < \cdots < z_{s-1} < z_s = z$ and the actual computation of $f(z_k)$ as a function of $f(z_{k-1})$ for $k = 1, \ldots, s$ (or as a function of $f(z_{k-1}), \ldots, f(z_{k-i})$ for some schemes).

The optimal choice of intermediate points is determined by the distance to the closest singularities in the complex plane as well as the efficiency of the scheme that computes $f(z_k)$ as a function of $f(z_{k-1})$. For $t \in [u, z]$, let $\varrho(t)$ be the convergence radius of the function $f$ at $t$. High order integration schemes will enable us to take $|z_k - z_{k-1}| \geqslant c\, \varrho(z_{k-1})$ for some fixed constant $c > 0$. Lower order schemes may force us to take smaller steps, but perform individual steps more efficiently. In some cases (e.g. when $f$ admits many singularities just above the real axis, but none below), it may also be interesting to allow for intermediate points $z_1, \ldots, z_{s-1}$ in $\mathbb{C}$ that keep a larger distance with the singularities of $f$.

For small working precisions, Runge-Kutta methods [24] provide the most efficient schemes for numerical integration. For instance, the best Runge-Kutta method of order 8 requires 11 evaluations of $f$ for each step. For somewhat larger working precisions (e.g. quadruple precision), higher order methods may be required in order to produce accurate results. One first alternative is to use relaxed power series computations [4, 7] which involve an overhead $n^2$ for small orders $n$ and $n \log^2 n$ for large orders. For very large orders, a power series analogue of Newton's method provides the most efficient numerical integration method [2, 26, 6]. This method actually computes the first variation of the solution along with the solution itself, which is very useful for the purpose of this paper.

### 2.2. Parallelism

Another interesting question concerns the amount of computations that can be done in parallel. In principle, the integration process is essentially sequential (apart from some parallelism which may be possible at each individual step). Nevertheless, given a full numeric solution at a sufficiently large precision $p$, we claim that a solution at a roughly doubled precision can be computed in parallel.

More precisely, for each $z$, $u$ and $I$, let $f(z, u, I)$ be the solution of (1) with $f(u, u, I) = I$, and denote $V(z, u, I) = (\partial f / \partial I)(z, u, I)$. We regard $f(z, u, I)$ as the "transitional flow" between $u$ and $z$, assuming the initial condition $I$ at $u$. Notice that $V(u, u, I) = \mathrm{Id}$ and, for $u < v < z$,

$$\begin{aligned} f(z, u, I) &= f(z, v, f(v, u, I)) \\ V(z, u, I) &= V(z, v, f(v, u, I))\, V(v, u, I). \end{aligned}$$

Now assume that we are given $f_{k,0;p} \approx f(z_k)$ for $k = 1, \ldots, s$ and at precision $p$. Then we may compute $V_{k,k-1;p} \approx V(z_k, z_{k-1}, f(z_{k-1}))$ in parallel at precision $p$. Using a dichotomic procedure of depth $\lceil \log_2 s \rceil$, we will compute $f_{k,0;2p} \approx f(z_k)$ at precision $2\,p$ in parallel for $k = 1, \ldots, s$, together with $V_{k,0;p} \approx V(z_k, z_0, f(z_0))$ at precision $p$.

More precisely, assume that $s \geqslant 2$ and let $m = \lceil s/2 \rceil$. We start with the recursive computation of $f_{k,0;2p} \approx f(z_k)$ and $f_{m+k,m;2p} \approx f(z_{m+k}, z_m, f_{m;p})$ at precision $2\,p$ for $k = 1, \ldots, m$ (resp. $k = 1, \ldots, s - m$), together with $V_{k,0;p} \approx V(z_k, z_0, f(z_0))$ and $V_{m+k,m;p} \approx V(z_{m+k}, z_m, f(z_m))$ at precision $p$. Setting $\delta = f_{m,0;2p} - f_{m,0;p}$, we take

$$\begin{aligned} f_{m+k,0;2p} &:= f_{m+k,m;2p} + V_{m+k,m;p}\, \delta \\ V_{m+k,0;p} &:= V_{m+k,m;p}\, V_{m,0;p} \end{aligned}$$

for $k=1,\ldots,s-m$. These formulas are justified by the facts that

$$\begin{aligned} f(z_{m+k}) &\approx f(z_{m+k}, z_m, f_{m,0;p}+\delta) \\ &\approx f_{m+k,m;2p}+V_{m+k,m;p}\,\delta \end{aligned}$$

at precision $2\,p$ and $V(z_{m+k}, z_0, f(z_0)) \approx V_{m+k,m;p}\,V_{m,0;p}$ at precision $p$.

The above algorithm suggests an interesting practical strategy for the integration of dynamical systems on massively parallel computers: the fastest processor(s) in the system plays the rôle of a "spearhead" and performs a low precision integration at top speed. The remaining processors are used for enhancing the precision as soon as a rough initial guess of the trajectory is known. The spearhead occasionally may have to redo some computations whenever the initial guess drifts too far away from the actual solution. The remaining processors might also compute other types of "enhancements", such as the first and higher order variations, or certifications of the trajectory. Nevertheless, the main bottleneck on a massively parallel computer seems to be the spearhead.

## 3. Global certification

### 3.1. From local to global certification

A *certified integrator* of the dynamical system (1) can be defined to be a ball function

$$\boldsymbol{f}\colon (z,u,\boldsymbol{I}) \mapsto \boldsymbol{f}(z,u,\boldsymbol{I})$$

with the property that $f(z,u,I)\in\boldsymbol{f}(z,u,\boldsymbol{I})$ for any $u<z$ and $I\in\boldsymbol{I}$. An *extended certified integrator* additionally requires a ball function

$$\boldsymbol{V}\colon (z,u,\boldsymbol{I}) \mapsto \boldsymbol{V}(z,u,\boldsymbol{I})$$

with the property that $V(z,u,I)\in\boldsymbol{V}(z,u,\boldsymbol{I})$ for any $u<z$ and $I\in\boldsymbol{I}$.

A *local certified integrator* of (1) is a special kind of certified integrator which only produces meaningful results if $z$ and $u$ are sufficiently close (and in particular $|z-u|<\varrho(u)$). In other words, we allow the radii of the entries of $\boldsymbol{f}(z,u,\boldsymbol{I})$ to become infinite whenever this is not the case. Extended local certified integrators are defined similarly.

One interesting problem is how to produce global (extended) certified integrators out of local ones. The most naive strategy for doing this goes as follows. Assume that we are given a local certified integrator $\boldsymbol{f}^{\text{loc}}$, as well as $u<z$ and $\boldsymbol{I}$. If the radii of the entries of $\boldsymbol{f}^{\text{loc}}(z,u,\boldsymbol{I})$ are "sufficiently small" (finite, for instance, but we might require more precise answers), then we define $\boldsymbol{f}^{\text{glob}}(z,u,\boldsymbol{I}) := \boldsymbol{f}^{\text{loc}}(z,u,\boldsymbol{I})$. Otherwise, we take $v=(z+u)/2$ and define $\boldsymbol{f}^{\text{glob}}(z,u,\boldsymbol{I}) := \boldsymbol{f}^{\text{glob}}(z,v,\boldsymbol{f}^{\text{glob}}(v,u,\boldsymbol{I}))$. One may refine the strategy by including additional exception handling for breakdown situations. It is well known that, unfortunately, this naive strategy produces error estimates of extremely poor quality (due to the wrapping effect, and for several other reasons).

### 3.2. Certifying a numerical trajectory

A better strategy is to first compute a numerical solution to (1) together with its first variation and to certify this "extended solution" at a second stage. So assume that we are given a subdivision $u=z_0<\cdots<z_s=z$ and approximate values $f_0\approx f(z_0),\ldots,f_s\approx f(z_s)$, as well as $V_0\approx V(z_0),\ldots,V_s\approx V(z_s)$. We proceed as follows:

**Stage 1.** We first produce reasonable candidate enclosures $\boldsymbol{f}_1=\boldsymbol{f}(z_1,z_0,\boldsymbol{I}),\ldots,\boldsymbol{f}_s=\boldsymbol{f}(z_s,z_0,\boldsymbol{I})$ with $f(z_k,z_0,I)\in\boldsymbol{f}_k$ for all $k=1,\ldots,s$ and $I\in\boldsymbol{I}$. Let $f_0$ denote the center of $\boldsymbol{f}_0=\boldsymbol{I}$, $\rho$ its radius, and let $p$ be the current working precision. For some large constant $K\gg 1$, a good typical ansatz would be to take

$$\boldsymbol{f}_k \;=\; f_k+2\,V_k\,\boldsymbol{\delta},$$

where

$$\boldsymbol{\delta} \;=\; \mathcal{B}(0,\rho+K\,2^{-p}\,|f_0|).$$

At the very end, we will have to prove the correctness of the ansatz, thereby producing a certificate for the numerical trajectory.

**Stage 2.** We compute $\boldsymbol{V}_{k,k-1} = \boldsymbol{V}^{\text{loc}}(z_k, z_{k-1}, \boldsymbol{f}_{k-1})$ for $k = 1, \ldots, s$ using an extended local integrator. Given $0 \leqslant j < k \leqslant s$, and assuming correctness of the ansatz enclosures $\boldsymbol{f}_j, \ldots, \boldsymbol{f}_{k-1}$, this provides us with a certified enclosure

$$\boldsymbol{V}_{k,j} = \boldsymbol{V}_{k,k-1} \cdots \boldsymbol{V}_{j+1,j} \tag{2}$$

for $V(z_k, z_j, \boldsymbol{f}_j)$.

**Stage 3.** We compute $\boldsymbol{\varphi}_{k,k-1} = \boldsymbol{f}^{\text{loc}}(z_k, z_{k-1}, \mathcal{B}(f_{k-1}, 0))$ for $k = 1, \ldots, s$ using a local integrator. Given $0 \leqslant j < k \leqslant s$, and assuming correctness of the ansatz enclosures $\boldsymbol{f}_j, \ldots, \boldsymbol{f}_{k-1}$, this provides us with certified enclosures

$$\boldsymbol{\varphi}_{k,j} = f_k + \sum_{i=j+1}^{k} \boldsymbol{V}_{k,i} (\boldsymbol{f}_{i,i-1} - f_i) \tag{3}$$

$$\boldsymbol{f}_{k,j} = \boldsymbol{\varphi}_{k,j} + \boldsymbol{V}_{k,j} (\boldsymbol{f}_j - f_j) \tag{4}$$

for $f(z_k, z_j, \mathcal{B}(f_j, 0))$ and $f(z_k, z_j, \boldsymbol{f}_j)$.

**Stage 4.** We finally check whether $\boldsymbol{f}_{k,0} \subseteq \boldsymbol{f}_k$ for $k = 1, \ldots, s$. If this is the case, then the correctness of the ansatz $\boldsymbol{f}_k$ follows by induction over $k$. Otherwise, for each index $k$ with $\boldsymbol{f}_{k,0} \nsubseteq \boldsymbol{f}_k$ we replace our ansatz $\boldsymbol{f}_k$ by a new one $\tilde{\boldsymbol{f}}_k$ as follows: we consider $\boldsymbol{\delta}_{k,0} := \boldsymbol{f}_{k,0} - f_k$, $\boldsymbol{\delta}_k := \boldsymbol{f}_k - f_k$, and take $\tilde{\boldsymbol{f}}_k := f_k + 2 \sup (\boldsymbol{\delta}_{k,0}, \boldsymbol{\delta}_k)$. We next return to Stage 2 with this new ansatz. We return an error if no certificate is obtained after a suitable and fixed number of such iterations.

**Remark 1.** If we want to certify our trajectory with a high precision $p$, then we clearly have to compute the enclosures $\boldsymbol{f}_k$ with precision $p$. On the other hand, the enclosures $\boldsymbol{V}_{k,j}$ are only needed during the auxiliary computations (3) and (4) and it actually suffices to compute them with a much lower precision (which remains bounded if we let $p \to \infty$). For the initial ansatz, we essentially need this precision to be sufficiently large such that $\boldsymbol{V}_k \boldsymbol{\delta} \subseteq 2 V_k \boldsymbol{\delta}$ for $k = 1, \ldots, s$. In general, we rather must have $f_k + \boldsymbol{V}_k \boldsymbol{\delta} \subseteq \boldsymbol{f}_k$.

## 3.3. Algorithmic considerations and parallelism

The next issue concerns the efficient and high quality evaluation of the formulas (2) and (3). The main potential problem already occurs in the case when $\Phi$ is constant, and (2) essentially reduces to the computation of the $k$-th power $\boldsymbol{V}^k$ of a ball matrix $\boldsymbol{V}$. Assuming standard ball matrix arithmetic, the naive iterative method

$$\boldsymbol{V}^k = \boldsymbol{V}\boldsymbol{V}^{k-1}$$

may lead to an exponential growth of the relative error as a function of $k$. Here we understand the relative error of a ball matrix to be the norm of the matrix of radii divided by the norm of the matrix of centers. The bad exponential growth occurs for instance for the matrix

$$\boldsymbol{V} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix},$$

which corresponds to the complex number $1 + \mathrm{i}$. The growth remains polynomial in $k$ in the case of triangular matrices $\boldsymbol{V}$. When using binary powering

$$\boldsymbol{V}^{2k} = \boldsymbol{V}^k \boldsymbol{V}^k$$
$$\boldsymbol{V}^{2k+1} = \boldsymbol{V}\boldsymbol{V}^k \boldsymbol{V}^k,$$

the growth of the relative error is systematically kept down to a polynomial in $k$.

For this reason, it is recommended to evaluate (2) and (3) using a similar dichotomic algorithm as in Section 2.2. More precisely, we will compute $\boldsymbol{\varphi}_{k,0}$ and $\boldsymbol{V}_{k,0}$ using a parallel dichotomic algorithm for $k = 1, \ldots, s$. Assuming that $s \geqslant 2$, let $m = \lceil s / 2 \rceil$. We start with the recursive computation of $\boldsymbol{\varphi}_{k,0}$ and $\boldsymbol{V}_{k,0}$ for $k = 1, \ldots, m$, as well as $\boldsymbol{\varphi}_{m+k,m}$ and $\boldsymbol{V}_{m+k,m}$ for $k = 1, \ldots, s - m$. Then we compute

$$\boldsymbol{V}_{m+k,0} := \boldsymbol{V}_{m+k,m} \boldsymbol{V}_{m,0}$$
$$\boldsymbol{\varphi}_{m+k,0} := \boldsymbol{\varphi}_{m+k,m} + \boldsymbol{V}_{m+k,m} (\boldsymbol{\varphi}_{m,0} - f_m)$$

for $k=1,\ldots,s-m$.

The depth of this dichotomic method is $O(\log s)$. Given the initial numerical trajectory, it follows that the cost of the certification grows only with $\log s$ on sufficiently parallel computers. It is also interesting to notice that the parallel dichotomic technique that we used to double the precision uses very similar ingredients as the above way to certify trajectories. We found this analogy to apply on several other occasions, such as the computation of eigenvalues of matrices. This is probably due to the similarity between ball arithmetic and arithmetic in jet spaces of order one.

## 4. Lagrange models

### 4.1. Taylor models

Let $\mathcal{D}=\mathcal{B}(0,r)$ be the compact disk of center zero and radius $r$. A *Taylor model* of order $n\in\mathbb{N}$ on $\mathcal{D}$ consists of a polynomial $P\in\mathbb{C}[z]$ of degree $<n$ together with an error $\varepsilon\in\mathbb{R}^{>}$. We will denote such a Taylor model by $P+\mathcal{B}_{\mathcal{D}}(\varepsilon)$ and consider it as a ball of functions: given an analytic function $f$ on $\mathcal{D}$ and $\boldsymbol{f}=P+\mathcal{B}_{\mathcal{D}}(\varepsilon)$, we write $f\in\boldsymbol{f}$ if $\|f-P\|_{\mathcal{D}}=\sup_{z\in\mathcal{D}}|f(z)-P(z)|\leqslant\varepsilon$.

Basic arithmetic on Taylor models works in a similar way as ball arithmetic. The ring operations are defined as follows:

$$\begin{aligned}(P+\mathcal{B}_{\mathcal{D}}(\delta))\pm(Q+\mathcal{B}_{\mathcal{D}}(\varepsilon)) &= (P\pm Q)+\mathcal{B}_{\mathcal{D}}(\delta+\varepsilon)\\ (P+\mathcal{B}_{\mathcal{D}}(\delta))\cdot(Q+\mathcal{B}_{\mathcal{D}}(\varepsilon)) &= (P\cdot Q)_{<n}+\mathcal{B}_{\mathcal{D}}(\lceil\!\lceil P\rceil\!\rceil_{\mathcal{D}}\,\varepsilon+\lceil\!\lceil Q\rceil\!\rceil_{\mathcal{D}}\,\delta+\varepsilon\,\delta+\lceil\!\lceil (P\cdot Q)_{\geqslant n}\rceil\!\rceil_{\mathcal{D}}).\end{aligned}$$

Given a polynomial $A\in\mathbb{C}[z]$ (or an analytic function $A$), the product formula uses the notations

$$\begin{aligned}A_{<n} &= A_0+\cdots+A_{n-1}\,z^{n-1}\\ A_{\geqslant n} &= A_n\,z^n+A_{n+1}\,z^{n+1}+\cdots,\end{aligned}$$

and $\lceil\!\lceil A\rceil\!\rceil_{\mathcal{D}}$ denotes any upper bound for $\|A\|_{\mathcal{D}}$ that is easy to compute. One may for instance take

$$\lceil\!\lceil A\rceil\!\rceil_{\mathcal{D}} = |A_0|+\cdots+|A_{\deg A}|.$$

Now consider the operation $\int$ of integration from zero $(\int f)(z)=\int_0^z f(u)\,\mathrm{d}u$. The integral of a Taylor model may computed using

$$\textstyle\int(P+\mathcal{B}_{\mathcal{D}}(\delta)) = \int P_{<n-1}+\mathcal{B}_{\mathcal{D}}(r\,|P_{n-1}|/n+r\,\delta).$$

This formula is justified by the mean value theorem.

In practice, the numerical computations at a given working precision involve additional rounding errors. Bounds for these rounding errors have to be added to the errors in the above formulas. It is also easy to define Taylor models on disks $\mathcal{B}(c,r)$ with general centers as being given by a Taylor model on $\mathcal{D}$ in the variable $z'=z-c$. For more details, we refer to [15, 16].

### 4.2. Lagrange models

A *Lagrange model* of order $n\in\mathbb{N}$ on $\mathcal{D}$ is a functional "ball" of the form $\boldsymbol{P}+\mathcal{B}_{\mathcal{D}}(\varepsilon)\,z^n$, where $\boldsymbol{P}\in\mathbb{C}[z]$ is a ball polynomial of degree $<n$ and $\varepsilon\in\mathbb{R}^{>}$ the so called *tail bound*. Given an analytic function $f$ on $\mathcal{D}$ and $\boldsymbol{f}=\boldsymbol{P}+\mathcal{B}_{\mathcal{D}}(\varepsilon)\,z^n$, we write $f\in\boldsymbol{f}$ if $f_k\in\boldsymbol{P}_k$ for all $k<n$ and $\|f_{\geqslant n}\,z^{-n}\|_{\mathcal{D}}\leqslant\varepsilon$. The name "Lagrange model" is motivated by Taylor–Lagrange's formula, which provides a careful estimate for the truncation error of a Taylor series expansion. We may also regard Lagrange models as a way to substantiate the "big Oh" term in the expansion $f=f_0+\cdots+f_{n-1}\,z^{n-1}+O(z^n)$.

Basic arithmetic on Lagrange models works in a similar way as in the case of Taylor models:

$$\begin{aligned}(\boldsymbol{P}+\mathcal{B}_{\mathcal{D}}(\delta)\,z^n)\pm(\boldsymbol{Q}+\mathcal{B}_{\mathcal{D}}(\varepsilon)\,z^n) &= (\boldsymbol{P}\pm\boldsymbol{Q})+\mathcal{B}_{\mathcal{D}}(\delta+\varepsilon)\,z^n\\ (\boldsymbol{P}+\mathcal{B}_{\mathcal{D}}(\delta)\,z^n)\cdot(\boldsymbol{Q}+\mathcal{B}_{\mathcal{D}}(\varepsilon)\,z^n) &= (\boldsymbol{P}\cdot\boldsymbol{Q})_{<n}+\mathcal{B}_{\mathcal{D}}(\lceil\!\lceil\boldsymbol{P}\rceil\!\rceil_{\mathcal{D}}\,\varepsilon+\lceil\!\lceil\boldsymbol{Q}\rceil\!\rceil_{\mathcal{D}}\,\delta+\varepsilon\,\delta+\lceil\!\lceil(\boldsymbol{P}\cdot\boldsymbol{Q})_{\geqslant n}\rceil\!\rceil_{\mathcal{D}})\,z^n.\end{aligned}$$

This time, we may take

$$\begin{aligned}\lceil\!\lceil\boldsymbol{A}\rceil\!\rceil_{\mathcal{D}} &= \lceil\boldsymbol{A}_0\rceil+\cdots+\lceil\boldsymbol{A}_{\deg\boldsymbol{A}}\rceil.\\ \lceil\mathcal{B}(c,\rho)\rceil &= |c|+\rho\end{aligned}$$

as the “easy to compute” upper bound of a ball polynomial $\boldsymbol{A} \in \mathbb{C}[z]$. The main advantage of Lagrange models with respect to Taylor models is that they allow for more precise tail bounds for integrals:

$$\textstyle\int(\boldsymbol{P}+\mathcal{B}_{\mathcal{D}}(\delta)\, z^n) \;=\; \int\boldsymbol{P}_{<n-1}+\mathcal{B}_{\mathcal{D}}(r\,\lceil\boldsymbol{P}_{n-1}\rceil/n+r\,\delta/(n+1))\, z^n.$$

Indeed, for any function $f$ on $\mathcal{D}$, integration on a straight line segment from 0 to any $z\in\mathcal{D}$ yields

$$\left|\int_0^z f(u)\, u^n\, \mathrm{d}u\right| = \left|\int_0^{z^{n+1}} \frac{f(\sqrt[n+1]{v})}{n+1}\, \mathrm{d}v\right| \leqslant \frac{r\,\|f\|_{\mathcal{D}}}{n+1},$$

whence $\|\int(f\, z^n)\|_{\mathcal{D}} \leqslant r\,\|f\|_{\mathcal{D}}/(n+1)$.

The main disadvantage of Lagrange models with respect to Taylor models is that they require more data (individual error bounds for the coefficients of the polynomial) and that basic arithmetic is slightly more expensive. Indeed, arithmetic on ball polynomials is a constant time more expensive than ordinary polynomial arithmetic. Nevertheless, this constant tends to one if either $n$ or the working precision $p$ gets large. This makes Lagrange models particularly well suited for high precision computations, where they cause negligible overhead, while improving the quality of tail bounds for integrals. For efficient implementations of basic arithmetic on ball polynomials, we refer to [5].

## 4.3. Reliable integration of dynamical systems

Let us now return to the dynamical system (1). We already mentioned relaxed power series computations and Newton's method as two efficient techniques for the numerical computation of power series solutions to differential equations. These methods can still be used for ball coefficients, modulo some preconditioning or suitable tweaking of the basic arithmetic on ball polynomials; see [5] for more details. In order to obtain a local certified integrator in the sense of Section 3.1, it remains to be shown how to compute tail bounds for truncated power solutions at order $n$.

From now on, we will be interested in finding local certified solutions of (1) at the origin. We may rewrite the system (1) together with the initial condition $f(0)=I$ as a fixed point equation

$$f \;=\; I+\textstyle\int\Phi(f). \tag{5}$$

Now assume that a Lagrange model $\boldsymbol{f}$ satisfies

$$\boldsymbol{I}+\textstyle\int\Phi(\boldsymbol{f}) \;\subseteq\; \boldsymbol{f}. \tag{6}$$

Then we claim that for any $I\in\boldsymbol{I}$, there is an analytic function $f\in\boldsymbol{f}$ that satisfies (5). The initial value problem (5) indeed admits a solution $f_{\text{fix}}$ on some small disk around the origin. Now the operator $\Xi\colon f\in\boldsymbol{f}\mapsto I+\int\Phi(f)\in\boldsymbol{f}$ has the property that $f_{\text{fix}}-\Xi^k(0)=O(z^k)$ for all $k$, whereas these differences $f_{\text{fix}}-\Xi^k(0)$ are all bounded on $\mathcal{D}$. It follows that $\Xi^k(0)$ converges to an analytic function on the interior of $\mathcal{D}$, which must be the analytic continuation of $f_{\text{fix}}$. This function is also bounded, so it extends by continuity to $\mathcal{D}$. Since the initial value problems on the border of $\mathcal{D}$ all admit analytic solutions on small disks, $f_{\text{fix}}$ admits an analytic extension to the whole disk $\mathcal{D}$.

Using ball power series computations we already know how to compute a ball polynomial $\boldsymbol{P}$ of degree $<n$ such that

$$\boldsymbol{I}+\textstyle\int\Phi(\boldsymbol{P}) \;\subseteq\; \boldsymbol{P}+O(z^n).$$

Taking $\boldsymbol{f}=\boldsymbol{P}+\mathcal{B}_{\mathcal{D}}(\varepsilon)\, z^n$, it remains to be shown how to compute $\varepsilon\in(\mathbb{R}^>)^d$ in such a way that (6) is satisfied. Now denoting by $J$ the Jacobian matrix of $\Phi$, and putting $\boldsymbol{\varepsilon}=\mathcal{B}_{\mathcal{D}}(\varepsilon)\, z^n$, we have

$$\Phi(\boldsymbol{f}) \;\subseteq\; \Phi(\boldsymbol{P})+J(\boldsymbol{f})\,\boldsymbol{\varepsilon}.$$

Writing $\boldsymbol{Q}+\mathcal{B}_{\mathcal{D}}(\delta)\, z^n$ for $\boldsymbol{I}+\int\Phi(\boldsymbol{P})$ and $\boldsymbol{\delta}=\mathcal{B}_{\mathcal{D}}(\delta)\, z^n$, we thus have $\boldsymbol{Q}\subseteq\boldsymbol{P}$ and it suffices to find $\boldsymbol{\varepsilon}$ such that

$$\boldsymbol{\delta}+\textstyle\int J(\boldsymbol{f})\,\boldsymbol{\varepsilon} \;\subseteq\; \boldsymbol{\varepsilon}. \tag{7}$$

Assuming that all eigenvalues of $J(\boldsymbol{f})$ are strictly bounded by $(n+1)/r$, it suffices to “take”

$$\varepsilon \;=\; \left\lceil\!\left\lceil \left(1-\tfrac{r}{n+1}\,J(\boldsymbol{f})\right)^{-1} \right\rceil\!\right\rceil_{\mathcal{D}} \delta. \tag{8}$$

We have to be a little bit careful here, since $J(\boldsymbol{f})$ depends on $\varepsilon$. Nevertheless, the formula (8) provides a good ansatz: starting with $\varepsilon^{[0]}=0$, we may define

$$\varepsilon^{[i+1]} \;:=\; \left\lceil\!\left\lceil \left(1-\tfrac{r}{n+1}\,J(\boldsymbol{P}+\mathcal{B}_{\mathcal{D}}(\varepsilon^{[i]})\,z^n)\right)^{-1} \right\rceil\!\right\rceil_{\mathcal{D}} \delta \tag{9}$$

for all $i$. If $r$ was chosen small enough, then this sequence quickly stabilizes. Assuming that $\varepsilon^{[l+1]}\approx\varepsilon^{[l]}$, we set $\varepsilon=\varepsilon^{[l]}+2\,(\varepsilon^{[l+1]}-\varepsilon^{[l]})$, and check whether (7) holds. If so, then we have obtained the required Lagrange model solution of (6). Otherwise, we will need to decrease $r$, or increase $n$ and the working precision.

## 4.4. Discussion

Several remarks are in place about the method from the previous subsection. Let us first consider the important case when $\boldsymbol{I}=\mathcal{B}(I,0)$ is given exactly, and let $R$ denote the convergence radius of the unique solution $f$ of (5). For large working precisions $p$ and expansion orders $n$, we can make $\delta$ arbitrarily small. Assuming that the eigenvalues of $J(f)$ are strictly bounded by $(n+1)/r$, this also implies that $\varepsilon^{[1]},\varepsilon^{[2]},\ldots$ become arbitrarily small, and that $\varepsilon=\varepsilon^{[1]}+2\,(\varepsilon^{[2]}-\varepsilon^{[1]})$ satisfies (7). In other words, for any $r<R$, there exists a sufficiently large $n$ (and working precision $p$) for which the method succeeds.

Let us now investigate what happens if we apply the same method with Taylor models instead of Lagrange models. In that case, the equation (8) becomes

$$\varepsilon \;=\; \lceil\!\lceil (1-r\,J(\boldsymbol{f}))^{-1} \rceil\!\rceil_{\mathcal{D}}\,\delta.$$

On the one hand this implies that the method will break down as soon as $1/r$ reaches the largest eigenvalue of $J(f)$, which may happen for $r\ll R$. Even if $J$ is constant (i.e. $f'=\Phi(f)$ reduces to the differential equation $f'=J f$ for a constant matrix $J$), the step size cannot exceed the inverse of the maximal eigenvalue of $J$. On the other hand, and still in the case when $J$ is constant, we see that Lagrange models allow us to take a step size which is $n+1$ times as large. In general, the gain will be smaller since $J$ usually admits larger eigenvalues on larger disks. Nevertheless, Lagrange models will systematically allow for larger step sizes.

Notice that the matrices that we need to invert in (8) and (9) admit Lagrange model entries, which should really be regarded as functions. Ideally speaking, we would like to compute a uniform bound for the inverses of the evaluations of these matrices at all points in $\mathcal{D}$. However, this may be computationally expensive. Usually, it is preferable to replace each Taylor model entry $\boldsymbol{g}+\mathcal{B}_{\mathcal{D}}(\eta)\,z^n$ of the matrix to be inverted by a ball enclosure $\boldsymbol{g}_0+\cdots+\boldsymbol{g}_{n-1}\,\mathcal{B}(0,r^{n-1})+\mathcal{B}(0,\eta\,r^n)$. The resulting ball matrix can be inverted much faster, although the resulting error bounds may be of inferior quality.

We finally stress (once more) that bound computations usually require a far smaller accuracy than the working precision. This makes it interesting to consider the problem of computing tail bounds for Lagrange (and Taylor) models independently from the problem of evaluating them: for accurate evaluations, we need the working precision $p$ and the expansion order $n$ to be approximately proportional. But the tail bound computations could be done at a much smaller precision $p'\ll n$. In particular, certifying the convergence of a solution to (1) on a compact disk can often be done using low precision computations only.

## Bibliography

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

**[2]** R.P. Brent and H.T. Kung. Fast algorithms for manipulating formal power series. *Journal of the ACM*, 25:581–595, 1978.

**[3]** T.N. Gambill and R.D. Skeel. Logarithmic reduction of the wrapping effect with application to ordinary differential equations. *SIAM J. Numer. Anal.*, 25(1):153–162, 1988.

**[4]** J. van der Hoeven. Relax, but don’t be too lazy. *JSC*, 34:479–542, 2002.

**[5]** J. van der Hoeven. Ball arithmetic. In Arnold Beckmann, Christine Gaßner, and Bededikt Löwe, editors, *Logical approaches to Barriers in Computing and Complexity*, number 6 in Preprint-Reihe Mathematik, pages 179–208. Ernst-Moritz-Arndt-Universität Greifswald, February 2010. International Workshop.

**[6]** J. van der Hoeven. Newton's method and FFT trading. *JSC*, 45(8):857–878, 2010.

**[7]** J. van der Hoeven. Faster relaxed multiplication. In *Proc. ISSAC '14*, pages 405–412. Kobe, Japan, July 2014.

**[8]** J. van der Hoeven. *Journées Nationales de Calcul Formel (2011)*, volume 2 of *Les cours du CIRM*, chapter Calcul analytique. CEDRAM, 2011. Exp. No. 4, 85 pages, `http://ccirm.cedram.org/ccirm-bin/fitem?id=CCIRM_2011__2_1_A4_0`.

**[9]** L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied interval analysis*. Springer, London, 2001.

**[10]** V. Kreinovich. Interval computations. `http://www.cs.utep.edu/interval-comp/`. Useful information and references on interval computations.

**[11]** U.W. Kulisch. *Computer Arithmetic and Validity. Theory, Implementation, and Applications*. Number 33 in Studies in Mathematics. De Gruyter, 2008.

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

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

**[14]** 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 on enclosure methods*, pages 201–217. Wien, New York, 2001. Springer.

**[15]** 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.

**[16]** K. Makino and M. Berz. Suppression of the wrapping effect by Taylor model-based validated integrators. Technical Report MSU Report MSUHEP 40910, Michigan State University, 2004.

**[17]** R.E. Moore. Automatic local coordinate transformations to reduce the growth of error bounds in interval computation of solutions to ordinary differential equation. In L.B. Rall, editor, *Error in Digital Computation*, volume 2, pages 103–140. John Wiley, 1965.

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

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

**[20]** A. Neumaier. The wrapping effect, ellipsoid arithmetic, stability and confedence regions. *Computing Supplementum*, 9:175–190, 1993.

**[21]** A. Neumaier. Taylor forms - use and limits. *Reliable Computing*, 9:43–79, 2002.

**[22]** K. Nickel. How to fight the wrapping effect. In Springer-Verlag, editor, *Proc. of the Intern. Symp. on interval mathematics*, pages 121–132. 1985.

**[23]** A. Neumaier. *Interval methods for systems of equations*. Cambridge university press, Cambridge, 1990.

**[24]** W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes, the art of scientific computing*. Cambridge University Press, 3rd edition, 2007.

**[25]** S.M. Rump. Verification methods: rigorous results using floating-point arithmetic. *Acta Numerica*, 19:287–449, 2010.

**[26]** A. Sedoglavic. *Méthodes seminumériques en algèbre différentielle ; applications à l'étude des propriétés structurelles de systèmes différentiels algébriques en automatique*. PhD thesis, École polytechnique, 2001.

**[27]** A. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Maths. Soc.*, 2(42):230–265, 1936.

**[28]** K. Weihrauch. *Computable analysis*. Springer-Verlag, Berlin/Heidelberg, 2000.