# Consistency Techniques in Ordinary Differential Equations*

Yves Deville[1], Micha Janssen[1], Pascal Van Hentenryck[2]

[1] Université catholique de Louvain,
Pl. Ste Barbe 2,
B-1348 Louvain-la-Neuve, Belgium
{yde,mja}@info.ucl.ac.be
[2] Brown University
Box 1910
Providence, RI 02912, USA
pvh@cs.brown.edu

**Abstract.** This paper studies the application of interval analysis and consistency techniques to ordinary differential equations. It presents a unifying framework to extend traditional numerical techniques to intervals. In particular, it shows how to extend explicit and implicit, one-step and multi-step, methods to intervals. The paper also took a fresh look at the traditional problems encountered by interval techniques and studied how consistency techniques may help. It proposes to generalize interval techniques into a two-step process: a forward process that computes an enclosure and a backward process that reduces this enclosures. In addition, the paper studies how consistency techniques may help in improving the forward process and the wrapping effect.

## 1 Introduction

Differential equations (DE) are important in many scientific applications in areas such as physics, chemistry, and mechanics to name only a few. In addition, computers play a fundamental role in obtaining solutions to these systems.

THE PROBLEM A (first-order) *ordinary differential equation* (ODE) system $\mathcal{O}$ is a system of the form

$$\begin{aligned} {u_1}'(t) &= f_1(t, u_1(t), \ldots, u_n(t)) \\ {u_2}'(t) &= f_2(t, u_1(t), \ldots, u_n(t)) \\ &\vdots \\ {u_n}'(t) &= f_n(t, u_1(t), \ldots, u_n(t)) \end{aligned}$$

In the following, we use the vector representation $u'(t) = f(t, u(t))$ or, more simply, $u' = f(t, u)$ Given an initial condition $u(t_0) = u_0$ and assuming existence and uniqueness of solution, the solution of $\mathcal{O}$ is a function $s^* : \Re \rightarrow \Re^n$ satisfying $\mathcal{O}$ and the initial condition $s^*(t_0) = u_0$. Note that differential equations of order $p$ (i.e. $f(t, u, u', u'', \ldots, u^p) = 0$ ) can always be transformed into an ODE by introduction of new variables. Although an ODE system can potentially be transformed into autonomous ODE ($u' = f(u)$) by the addition of a new function $u_{n+1}(t)$ (with $u'_{n+1}(t) = 1$ and $u_{n+1}(t_0) = t_0$), we prefer to keep the time variable explicit for a clearer presentation of some of our novel techniques. However, the autonomous form is more appropriate for some treatment such as automatic differentiation.

There exist different mathematical methods for proving the existence and uniqueness of a solution of an ODE system with initial value. But, in practice, a system is generally required, not only to prove existence, but also to produce numerical values of the solution $s^*(t)$ for different values of variable $t$. If, for some classes of ODE systems, the solution can be represented in closed form (i.e. combination of elementary functions), it is safe to say that most ODE systems cannot be solve explicitly [Hen62]. For instance, the innocent-looking equation $u' = t^2 + u^2$ cannot be solved in terms of elementary functions!

*Discrete variable methods* aim to approximate the solution $s^*(t)$ of *any* ODE system, not over a continuous range of $t$, but only at some points $t_0, t_1, \ldots, t_m$. Discrete variable methods include *one-step methods* (where $s^*(t_j)$ is approximated from the approximation $u_{j-1}$ of $s^*(t_{j-1})$) and *multistep methods* (where $s^*(t_j)$ is

* Proceedings of the Fourth International Conference on Principles and Practice of Constraint Programming, LNCS, Springer-Verlag, 1998.

approximated from the approximation $u_{j-1}, \ldots, u_{j-p}$ of $s^*(t_{j-1}), \ldots, s^*(t_{j-p})$). In general, these methods do not guarantee the existence of a solution within a given bound and may suffer from traditional numerical problems of floating-point systems.

INTERVAL ANALYSIS IN ODE Interval techniques for ODE systems were introduced by Moore [Moo66]. (See [BBCG96] for a description and a bibliography of the application of interval analysis to ODE systems.) These methods provide numerically reliable enclosures of the exact solution at points $t_0, t_1, \ldots, t_m$. To achieve this result, they typically apply a one-step Taylor interval method and make extensive use of automatic differentiation to obtain the Taylor coefficients [Moo79, Ral80, Ral81, Cor88, Abe88]. The major problem of interval methods on ODE systems is the explosion of the size of resulting boxes at points $t_0, t_1, \ldots, t_m$. There are mainly two reasons for this explosion. On the one hand, step methods have a tendency to accummulate errors from point to point. On the other, the approximation of an aribitrary region by a box, called the wrapping effect, may introduce considerable imprecision after a number of steps. One of the best systems in this area is Lohner's AWA [Loh87, Sta96]. It uses the Picard iteration to prove existence and uniqueness and to find a rough enclosure of the solution. This rough enclosure is then used to compute correct enclosures using a mean value method and the Taylor expansion on a variational equation on global errors. It also applies coordinate transformations to reduce the wrapping effect.

GOAL OF THE PAPER This paper mainly serves two purposes. First, it provides a unifying framework to extend traditional numerical techniques to intervals. In particular, the paper shows how to extend explicit and implicit, one-step and multi-step, methods to intervals. Second, the paper attempts to take a fresh look at the traditional problems encountered by interval techniques and to study how consistency techniques may help. It proposes to generalize interval techniques into a two-step process: a forward process that computes an enclosure and a backward process that reduces this enclosure. In addition, the paper studies how consistency techniques may help in improving the forward process and the wrapping effect.

The new techniques proposed in this paper should be viewed as defining an experimental agenda to be carried out in the coming years. The techniques are reasonably simple mathematically and algorithmically and were motivated by the same intuitions as the techniques at the core of the NUMERICA system [VHLD97]. In this respect, they should complement well existing methods. But, as it was the case for NUMERICA, only extensive experimental evaluation will determine which combinations of these techniques is useful in practice and which application areas they are best suited for. Very preliminary experimental results illustrate the potential benefits.

The rest of this paper is organized as follows. Section 2 provides the necessary background and notations. Section 3 presents the generic algorithm that can be instantiated to produce the various methods. Section 4 describes how to find bounding box. Section 5 describes the step methods used in the forward phase. Section 6 describes the backward pruning based on box-consistency. Section 7 discusses the wrapping effect. Section 8 presents some experimental results. Section 9 concludes the paper.

## 2 Background and Definitions

This paper uses rather standard notations of interval programming. $\mathcal{F}$ denotes the set of $\mathcal{F}$-numbers, $\mathcal{D}$ the set of boxes $\subseteq \Re^n$ whose bounds are in $\mathcal{F}$, $\mathcal{I}$ the set of intervals $\subseteq \Re$ whose bounds are in $\mathcal{F}$, and $D$ (possibly subscripted) denotes a box in $\mathcal{D}$. Given a real $r$ and a subset $A$ of $\Re^n$, $\bar{r}$ denotes the smallest interval in $\mathcal{I}$ containing $r$ and $\Box A$ the smallest box in $\mathcal{D}$ containing $A$. If $g$ is a function, $\hat{g}$ and $G$ denote interval extensions of $g$. We also use $g_i(x)$ and $G_i(D)$ to denote the $i^{th}$ component of $g(x)$ and $G(D)$.

The solution of an ODE system can be formalized mathematically as follows.

**Definition 1 Solution of an ODE System with Initial Value.** A *solution* of an ODE system $\mathcal{O}$ with initial conditions $u(t_0) = u_0$ is a function $s^*(t) : \Re \rightarrow \Re^n$ satisfying $\mathcal{O}$ and the initial conditions $s^*(t_0) = u_0$.

In this paper, we restrict attention to ODE systems that have a unique solution for a given initial value. Techniques to verify this hypothesis numerically are given in the paper. Moreover, in practice, as mentioned, the objective is to produce (an approximation of) the values of the solution function $s^*$ of the system $\mathcal{O}$ at different points $t_0, t_1, \ldots, t_m$. It is thus useful to adapt the definition of a solution to account for this practical motivation.

**Definition 2 Solution of an ODE System.** The *solution* of an ODE system $\mathcal{O}$ is a function

$$s(t_0, u_0, t_1) : \Re \times \Re^n \times \Re \rightarrow \Re^n$$

such that $s(t_0, u_0, t_1) = s^*(t_1)$, where $s^*$ is the solution of $\mathcal{O}$ with initial conditions $u(t_0) = u_0$.

The solution of an ODE system $\mathcal{O}$ can be used to obtain the solution of $\mathcal{O}$ at *any* point for *any* initial value. It is useful to extend our definition to sets of values.

**Definition 3 Set Solution of an ODE System.** Let $s$ be the solution of an ODE system $\mathcal{O}$. The *set solution* $\mathcal{O}$ at $t_1$ wrt $t_0$ and $D$ is the set $s(t_0, D, t_1) = \{s(t_0, u, t_1) \mid u \in D\}$.

The interval techniques presented in this paper aim at approximating set solutions as tightly as possible. The next definition introduces the concept of bounding box that is fundamental to prove the existence and the uniqueness of a solution to an ODE system over a box and to bound the errors.

**Definition 4 Bounding Box.** Let $s$ be the solution of an ODE system $\mathcal{O}$. A box $B$ is a *bounding box* of $s$ in $[t_0, t_1]$ wrt $D$ if, forall $t \in [t_0, t_1]$, $s(t_0, D, t) \subseteq B$.

Informally speaking, a bounding box is thus an enclosure of the solution on the *whole interval* $[t_0, t_1]$. The following theorem is an interesting topological property of solutions.

**Theorem 5.** *Let $\mathcal{O}$ be an ODE system $u' = f(t, u)$ with $f \in \mathcal{C}$, let $s$ be the solution of $\mathcal{O}$ (i.e. existence and uniqueness), and let $Fr$ be the frontier of $D$. Then,*
1. *$s(t_0, D, t_1)$ is a closed set;*
2. *$s(t_0, Fr, t_1)$ is the frontier of $s(t_0, D, t_1)$.*

*Proof.* (Sketch) Under the given hypothesis, $s \in \mathcal{C}$ [Har64]. It can then be shown that a point in $D \setminus Fr$ cannot belong to the frontier of $s(t_0, D, t_1)$ [DJVH98].

As a consequence, $s(t_j, D_j, t_{j+1})$ can be computed by considering the frontier of $D_j$ only.

## 3 The Generic Algorithm

The interval methods described in this paper can be viewed as instantiations of a generic algorithm. It is useful to present the generic algorithm first and to describe its components in detail in the rest of the paper. The generic algorithm is parametrized by three procedures: a procedure to compute a bounding box, since bounding-boxes are fundamental in obtaining enclosures, a step procedure to compute forward, and a procedure to prune by using step procedures backwards. Procedure BoundingBox computes a bounding box of an ODE system in an interval for a given box. Procedure Step computes a box approximating the value of $s^*(t_j)$ given the approximations of $s^*(t_k)$ $(1 \leq k \leq j-1)$ and the bounding boxes $B_1, \ldots, B_{j-1}$. Procedure Prune prunes the boxes $D_j$ at $t_j$ using the box $D_{j-1}$ at $t_{j-1}$. The intuition underlying the basic step of the generic algorithm is illustrated in Figure 1. The next three sections review these three components. Note however that it is possible to use several step procedures, in which case the intersection of their results is also an enclosure.

```
solve(O, D0, < t0, ..., tm >)
begin
   forall(j in 1..n)
   begin
      Bj := BoundingBox(O, Dj-1, tj-1, tj);
      Dj := Step(O, < D0, ..., Dj-1 >, < t0, ..., tj >, < B0, ..., Bj >);
      Dj := Prune(O, Dj-1, tj-1, Dj, tj, Bj);
   end;
end;
```

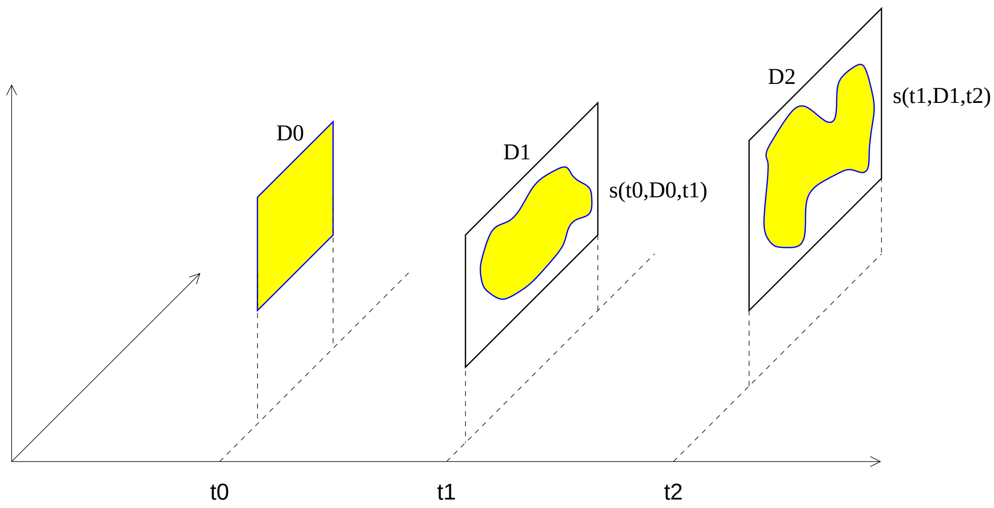


**Fig. 1.** Computing correct enclosures of the solution

# 4 The Bounding Box

This section considers how to obtain a bounding box for an ODE system. As will become clear later on, bounding boxes are fundamental to obtain reliable solutions to ODE systems. The traditional interval techniques to obtain bounding boxes are based on Picard operator [Har64, Moo79].

**Theorem 6 (Picard Operator).** *Let $D_0$ and $B$ be two boxes such that $D_0 \subseteq B$, let $[t_0, t_1] \in \mathcal{I}$, and let $h = t_1 - t_0$. Let $\mathcal{O}$ be an ODE system $u' = f(t, u)$, where $f$ is continuous and has a Jacobian (i.e first-order partial derivatives) over $[t_0, t_1]$. Let $\Phi$ be the transformation (Picard Operator)*

$$\Phi(B) = D_0 + [0, h]F([t_0, t_1], B).$$

*where $F$ be an interval extension of $f$.*

*If $\Phi(B) \subseteq B$, then 1. The $\mathcal{O}$ system with initial value $u(t_0) \in D_0$ has a unique solution $s$;*
*2. $\Phi(B)$ is a bounding box of $s$ in $[t_0, t_1]$ wrt $D_0$.*

Theorem 6 can be used for proving existence and uniqueness of a solution and for providing a bounding box [Loh87, Cor95]. A typical algorithm starts from an approximation $B^0 = D_0$ and applies Picard operator. If $\Phi(B^0) \not\subseteq B$, the algorithm widens $B^0$ into $B^1$ (e.g., by doubling its size) and iterates the process. The algorithm can also narrow the step size. Note that the existence of Jacobian(f) can be checked numerically by evaluating its interval extension over the box. Note also that the Picard operator uses a Taylor expansion of order 1. It can be generalized for higher orders, which is interesting to increase the step sizes.

# 5 The Step Methods

This section describes the step methods. The step methods are presented in isolation. However, as mentioned previously, they can be used together, since the intersection of their results is also a step method.

## 5.1 Explicit One-Step Methods

This section considers one-step methods: It first describes traditional numerical methods, moves to traditional interval methods, and proposes improvements which can be obtained from consistency techniques.

TRADITIONAL NUMERICAL METHODS To understand traditional interval methods, it is useful to review traditional numerical methods. In explicit one-step methods, the solution $s$ of an ODE system $\mathcal{O}$ is viewed as the summation of two functions:

$$s(t_0, u_0, t_1) = sc(t_0, u_0, t_1) + e(t_0, u_0, t_1). \tag{1}$$

where the function $sc$ can always be computed while the function $e$ cannot. As a consequence, a traditional numerical method based on an explicit one-step method is an algorithm of the form

```
forall(i in 1..n)
   u_i := sc(t_{i-1}, u_{i-1}, t_i);
```

This algorithm tries to approximate the solution $s^*(t)$ for an initial value $u(t_0) = u_0$.

*Example 1 Taylor Method.* The Taylor method is one of the best known explicit one-step methods where the function $sc$ is given by the Taylor expansion of a given order $p$, i.e.,

$$sc_T(t_0, u_0, t_1) = u_0 + hf^{(0)}(t_0, u_0) + \frac{h^2}{2} f^{(1)}(t_0, u_0) + \ldots + \frac{h^p}{p!} f^{(p-1)}(t_0, u_0)$$

INTERVAL METHODS The key idea underlying (explicit or implicit) one-step interval methods is to define an extension of the interval solution $s$.[3]

**Definition 7 Interval Solution of an ODE System.** Let $s$ be the solution of an ODE system $\mathcal{O}$. An *interval solution* of $\mathcal{O}$ is an interval extension $S$ of $s$, i.e.

$$\forall t_0, t_1 \in \mathcal{F}\ ,\ D_0 \in \mathcal{D} : s(t_0, D_0, t_1) \subseteq S(t_0, D_0, t_1)$$

As a consequence, a traditional interval method based on an one-step method is an algorithm of the form

```
D_0 = ū_0;
forall(i in 1..n)
   D_i := S(t_{i-1}, D_{i-1}, t_i);
```

This algorithm provides safe intervals for $s^*(t_1), \ldots, s^*(t_m)$, i.e.,

$$s^*(t_i) \in D_i \ \ (1 \leq i \leq m).$$

DIRECT INTERVAL EXTENSIONS Traditionally, interval solutions are often constructed by considering an explicit one-step function $s(t_0, u_0, t_1) = sc(t_0, u_0, t_1) + e(t_0, u_0, t_1)$, by taking an interval extension $SC$ of $sc$ and by using a bounding box to bound the error function $e$ to obtain a function of the form $S(t_0, D_0, t_1) = SC(t_0, D_0, t_1) + E(t_0, D_0, t_1)$.

**Definition 8 Direct Explicit One-Step Interval Extension.** Let $s$ be an explicit one-step solution of an ODE system $\mathcal{O}$ of the form

$$s(t_0, u_0, t_1) = sc(t_0, u_0, t_1) + e(t_0, u_0, t_1)$$

A direct explicit one-step interval extension of $s$ is an interval solution $S$ of the form

$$S(t_0, D_0, t_1) = SC(t_0, D_0, t_1) + E(t_0, B_0, t_1).$$

where $SC$ is an interval extension of $sc$, $B_0$ is a bounding box of $s$ in $[t_0, t_1]$ wrt $D_0$, and $E$ is an interval extension of $e$.

[3] As usual, interval solutions could also be defined on particular subsets of $\mathcal{F}$ and $\mathcal{D}$.

*Example 2 Taylor Interval Solution.* The Taylor Interval Solution of order $p$ of an ODE system $\mathcal{O}$ is defined as

$$S_T(t_0, D_0, t_1) = D_0 + hF^{(0)}(t_0, D_0) + \frac{h^2}{2}F^{(1)}(t_0, D_0) + \ldots + \frac{h^p}{p!}F^{(p-1)}(t_0, D_0) + \frac{h^{p+1}}{(p+1)!}F^{(p)}([t_0, t_1], B)$$

where $h = t_1 - t_0$, $B$ is a bounding box of $s$ in $[t_0, t_1]$ wrt $D_0$, and the interval functions $F^{(j)}$ are interval extensions of functions $f^{(j)}$ inductively defined as follows [Moo66] ($f^{(j)}$ is the "total $j^{th}$ derivative of $f$ wrt $t$")

$$f_i{}^{(0)}(t, u_1(t), \ldots, u_n(t)) = f_i(t, u_1(t), \ldots, u_n(t))$$
$$f_i{}^{(j)}(t, u_1(t), \ldots, u_n(t)) = \frac{\partial f_i{}^{(j-1)}}{\partial t} + \sum_{1 \leq m \leq n} \frac{\partial f_i{}^{(j-1)}}{\partial u_m} f_m{}^{(0)}$$

More information on automatic generation of the value of these functions can be found in [Moo79, Ral80, Ral81, Cor88, Abe88].

Piecewise Interval Extensions Direct interval techniques propagate entire boxes through interval solutions. As a consequence, errors may tend to accummulate as computations proceed. This section investigates a variety of techniques inspired by, and using, consistency techniques that can be proposed to reduce the accummulation of errors. The main idea, which is used several times in this paper and was inspired by box-consistency, is to propagate small boxes as illustrated in Figure 2.

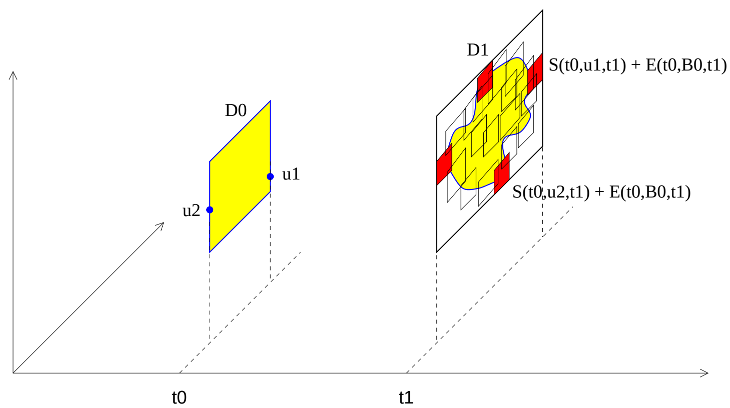


**Fig. 2.** A Piecewise Interval Solution

**Definition 9 Piecewise Explicit One-Step Interval Extension.** Let $s$ be a solution to an ODE system $\mathcal{O}$ of the form

$$s(t_0, u_0, t_1) = sc(t_0, u_0, t_1) + e(t_0, u_0, t_1).$$

A piecewise explicit one-step interval extension of $s$ is a function $S(t_0, D, t_1)$ defined as

$$S(t_0, D_0, t_1) = \Box\{SC(t_0, \bar{u_0}, t_1) \mid u_0 \in D_0\} + E(t_0, B_0, t_1)$$

where $SC$ is an interval extension of $sc$, $B_0$ is a bounding box of $s$ in $[t_0, t_1]$ wrt $D_0$, and $E$ is an interval extension of $e$.

Piecewise interval extensions of an ODE system are not only a theoretical concept: they can in fact also be computed. The basic idea here is to express piecewise interval extension as unconstrained optimization problems.

**Proposition 10.** *Let $s$ be a solution to an ODE system $\mathcal{O}$ of the form*

$$s(t_0, u_0, t_1) = sc(t_0, u_0, t_1) + e(t_0, u_0, t_1).$$

*A piecewise explicit one-step interval extension of $s$ is a function $S(t_0, D, t_1)$ defined as*

$$S_i(t_0, D_0, t_1) = [min_{u \in D_0} SC_i(t_0, u, t_1), max_{u \in D_0} SC_i(t_0, u, t_1)] \; + E_i(t_0, B_0, t_1) \quad (1 \leq i \leq n)$$

*where $SC$ is an interval extension of $sc$, $B_0$ is a bounding box of $s$ in $[t_0, t_1]$ wrt $D_0$, and $E$ is an interval extension of $e$.*

Note that these minimization problems must be solved globally to guarantee reliable solutions. The implementation section discusses how a system like NUMERICA may be generalized to solve these problems. The efficiency of the system of course depends on the step size, on the size of $D_0$, and on the desired accuracy. It is interesting to observe that the function $SC$ does not depend on the error term and hence methods that are not normally considered in the interval community (e.g., Runge-Kutta method) may turn beneficial from a computational standpoint. It is of course possible to sacrifice accuracy for computation time by using projections, the fundamental idea behind consistency techniques. For instance, interval methods are generally very fast on one-dimensional problems, which partly explains why consistency techniques have been successful to solve systems of nonlinear equations.

**Definition 11 Box-Piecewise Explicit One-Step Interval Extension.** Let $s$ be a solution to an ODE system $\mathcal{O}$ of the form

$$s(t_0, u_0, t_1) = sc(t_0, u_0, t_1) + e(t_0, u_0, t_1).$$

A box-piecewise explicit one-step interval extension of $s$ wrt dimension $i$ is a function $S_i(t_0, D, t_1)$ defined as

$$S_i(t_0, < I_1, \ldots, I_n >, t_1) = \Box\{SC(t_0, < I_1, \ldots, I_{i-1}, \bar{r}, I_{i+1}, \ldots, I_n >, t_1) \mid r \in I_i\} + E(t_0, B_0, t_1)$$

where $SC$ is an interval extension of $sc$, $B_0$ is a bounding box of $s$ in $[t_0, t_1]$ wrt $D_0$, and $E$ is an interval extension of $e$. The box-piecewise explicit one-step interval extension of $s$ wrt $E$ and $B$ is the function

$$S(t_0, D_0, t_1) = \cap_{i \in 1..n} S_i(t_0, D_0, t_1)$$

Each of the interval solutions reduces to a one-dimensional (interval) unconstrained optimization problem. The following property is a direct consequence of interval extensions.

**Proposition 12 (Box-)Piecewise Explicit One-Step Interval Solution.** *The piecewise and box-piecewise one-step interval extensions are interval solutions.*

In essence, box-optimal solutions safely approximate a multi-dimensional problem by the intersection of many one-dimensional problems. Of course, it is possible, and probably desirable, to define notions such as box(k)-piecewise interval solutions where projections are performed on several variables. Finally, notice that optimal interval solutions were defined with respect to a given bounding box. More precise interval solutions could be obtained if local bounding boxes were considered in the above definitions. It is easy to generalize our definitions to integrate this idea.

### 5.2 Implicit One-Step Methods

This section considers implicit one-step method. It first reviews traditional numerical methods and shows how they can be generalized to obtain interval methods. The presentation essentially follows the same lines as the previous section.

Traditional Numerical Methods In implicit one-step methods, the solution of ODE $\mathcal{O}$ is viewed as the solution of an equation.

**Definition 13 Implicit One-Step Solution.** An implicit one-step solution to an ODE system $\mathcal{O}$ is a function of the form $s(t_0, u_0, t_1) = u_1$ where $u_1$ is the solution of an equation $u_1 = sc(t_0, u_0, t_1, u_1) + e(t_0, u_0, t_1)$.

Since the error term cannot be computed in general, the above equation is replaced in practice by its approximation $u_1 = sc(t_0, u_0, t_1, u_1)$. As a result, an implicit one-step method is an algorithm of the form

```
forall(i in 1..n)
   u_i := solve(u_i =sc(t_{i-1}, u_{i-1}, t_i, u_i));
```

where *solve(S)* returns an element $x$ in *Solution(S)*, the set of solutions of $S$.

*Example 3 Trapezoid Method.* The trapezoid method is an implicit-one step method that consists of solving, at each step, an equation of the form $u_1 = u_0 + \frac{h}{2}(f(t_0, u_0) + f(t_1, u_1))$.

Interval Methods We now show how to generalize implicit one-step methods to intervals. The basic idea is to replace the search for a solution to a system of equations by a search for the solutions of a set of interval equations. The resulting interval solution can then be used as in explicit methods.

**Definition 14 Direct Implicit One-Step Interval Extension.** Let $s(t_0, u_0, t_1) = u_1$ where $u_1$ is the solution of the equation $u_1 = sc(t_0, u_0, t_1, u_1) + e(t_0, u_0, t_1)$ be an implicit one-step interval solution of an ODE system $\mathcal{O}$. Let $SC$ be an interval extension of $sc$ and $E$ be an interval extension of $e$. A direct implicit one-step interval extension of $s$ is an interval function $S(t_0, D_0, t_1) = D_1$ where

$$D_1 = \Box\{D \in B \mid D \text{ is canonical } \& \ D \approx SC(t_0, D_0, t_1, D) + E(t_0, B_0, t_1)\}$$

and $B_0$ is a bounding box of $s$ in $[t_0, t_1]$ wrt $D_0$.[4]

Note that this definition amounts to finding all solutions of an "interval equation" in a box. The definition uses the bounding box as the initial search space. However, any step method can be used instead to provide a smaller search space.

*Example 4 Trapezoid Interval Method.* The trapezoid interval extension of the trapezoid method requires the solving of the system of interval-valued equations

$$u_1 = u_0 + \frac{h}{2}(F(t_0, D_0) + F(t_1, u_1)) + \frac{h^3}{12}F^{(2)}([t_0, t_1], B_0)$$

where $h = t_1 - t_0$, $F$ is an interval extension of $f$, and $B_0$ is a bounding box of $s$ in $[t_0, t_1]$ wrt $D_0$.

It is possible to improve this result by incorporating the idea of piecewise interval extension proposed earlier. Coarser extensions can be defined in a similar way as well.

**Definition 15 Piecewise Implicit One-Step Interval Extension.** Let $s(t_0, u_0, t_1) = u_1$ where $u_1$ is the solution of the equation $u_1 = sc(t_0, u_0, t_1, u_1) + e(t_0, u_0, t_1)$ be an implicit one-step interval solution of an ODE system $\mathcal{O}$. Let $SC$ be an interval extension of $sc$ and $E$ be an interval extension of $e$. A piecewise implicit one-step interval extension of $s$ is an interval function $S(t_0, D_0, t_1) = D_1$ where

$$D_1 = \Box\{D \in B \mid D \approx SC(t_0, D_c, t_1, D) + E(t_0, B_0, t_1) \ \& \ D_c \subseteq D_0 \ \& \ D, D_c \text{ are canonical }\}$$

and $B_0$ is a bounding box of $s$ in $[t_0, t_1]$ wrt $D_0$.

[4] The interval relation $\approx$ is an interval extension of equality, i.e., $D_1 \approx D_2$ if $D_1 \cap D_2 \neq \emptyset$.

### 5.3 Multi-Step Methods

Some methods for solving ODE are multistep methods that compute the value at point from values at point $t_{j-k}, \ldots, t_{j-1}$ (for some $k > 1$). Obviously, the value at points $t_1, \ldots, t_{k-1}$ must be computed by some other method. In order to define interval extensions of such methods, we extend our definition of solution and interval solution. We use the following notations : $\vec{u} = \langle u_0, \ldots, u_{k-1} \rangle$, $\vec{D} = \langle D_0, \ldots, D_{k-1} \rangle$, $D_k = \langle I_1, \ldots, I_n \rangle$, and $\vec{t} = \langle t_0, \ldots, t_{k-1} \rangle$.

**Definition 16 Multistep Solution of an ODE System.** Let $s$ be the solution of an ODE system $\mathcal{O}$. The *multistep solution* of $\mathcal{O}$ at $t_k$ wrt $\vec{t}, \vec{D}$ is the set

$$ms(\vec{t}, \vec{D}, t_k) = \{s(t_0, u_0, t_k) \mid s(t_0, u_0, t_i) \in D_i\} \quad \text{for } 0 \leq i < k \ \}$$

**Definition 17 Multistep Interval Solution of an ODE System.** Let $ms$ be the multistep solution of an ODE system $\mathcal{O}$. A *multistep interval solution* of $\mathcal{O}$ is an interval extension $S$ of $ms$, i.e.

$$\forall \vec{t}, t_k, \vec{D} \quad ms(\vec{t}, \vec{D}, t_k) \subseteq S(\vec{t}, \vec{D}, t_k)$$

Explicit Multi-Step Methods In explicit multi-step methods, the solution $s$ of ODE $\mathcal{O}$ is decomposed as follows :

$$ms(\vec{t}, \vec{u}, t_k) = msc(\vec{t}, \vec{u}, t_k) + e(\vec{t}, \vec{u}, t_k) \tag{2}$$

These methods can be generalized to intervals in a way similar to one-step methods. For brevity, we only give an example of such an interval method.

*Example 5 Adams-Bashforth Interval Solution (Order 4).* Let $h$ be $t_{i+1} - t_i$ for $1 \leq i \leq 4$. The Adams-Bashforth interval solution of order 4 is the multi-step interval solution

$$S_{AB}(\langle t_0, t_1, t_2, t_3 \rangle, \langle D_0, D_1, D_2, D_3 \rangle, t_4) = D_4$$

where

$$D_4 = D_3 + \frac{h}{24}(55F(t_3, D_3) - 59F(t_2, D_2) + 37F(t_1, D_1) - 9F(t_0, D_0)) + \frac{251h^5}{720} F^{(4)}([t_0, t_4], B)$$

and $B$ is a bounding box of $s$ in $[t_0, t_4]$ wrt $D_0$. Notice that $F^{(4)}([t_0, t_4], B)$ can be approximated by $F^{(4)}([t_0, t_4], B) = \Box(\cup_{1 \leq i < 4} F^{(4)}([t_i, t_{i+1}], B_i))$ where $B_i$ is a bounding box of $s$ in $[t_i, t_{i+1}]$ wrt $D_i$.

Implicit Multistep Methods Implicit multi-step methods can be defined in a similar fashion. Let $u_k = ms(\vec{t}, \vec{u}, t_k)$. The value of $u_k$ is the solution of the equation :

$$u_k = msc(\vec{t}, \vec{u}, t_k, u_k) + e(\vec{t}, \vec{u}, t_k)$$

*Example 6 Adams-Moulton Implicit Multi-Step Interval Solution.* Let $h = t_{i+1} - t_i$ for $1 \leq i \leq 3$. The Adams-Moulton implicit multi-step interval solution is the function defined as

$$S_{AM}(\langle t_0, t_1, t_2 \rangle, \langle D_0, D_1, D_2 \rangle, t_3) = D_3$$

where

$$\begin{aligned} D_3 = \Box\{D \in B \mid \ & D \approx D_2 + \tfrac{h}{24}(9F(t_3, D) + 19F(t_2, D_2) - 5F(t_1, D_1) + F(t_0, D_0)) + \tfrac{-19h^5}{720} F^{(4)}([t_0, t_3], B) \\ & \& \ D \text{ is canonical } \ \& \ B \text{ is a bounding box of } s \text{ in } [t_0, t_3] \text{ wrt } D_0.\} \end{aligned}$$

### 5.4 Mean Value Form Step Methods

It can be observed that, in all our explicit interval solutions $S$,

$$D_0 \subseteq S(t_0, D_0, t_1) \text{ or } D_{k-1} \subseteq S(\vec{t}, \vec{D}, t_k)$$

indicating that the intervals are thus growing. Mean value forms have been proposed to use contraction characteristics of functions and may return smaller intervals. From Equation 1, we may apply the mean value theorem on $sc(t_0, u, t_1)$ (on variable $u$) to obtain

$$s(t_0, u, t_1) = sc(t_0, m, t_1) + \sum_{i=1}^{n} \left( \frac{\partial sc}{\partial (u)_i} \right) (t_0, \xi, t_1)\ (u_i - m_i) + e(t_0, u, t_1)$$

for some $\xi$ between $u$ and $m$. As a consequence, any interval solution of $s$, may serve as a basis to define a new interval solution.

**Definition 18.** Let $D$ be a box $\langle I_1, \ldots, I_n \rangle$, $m_i$ be the center of $I_i$, and $S_M = SC_M + E_M$ be an interval solution of an ODE system $\mathcal{O}$. The MVF solution of $\mathcal{O}$ in $D$ wrt $SC_M$, denoted by $\tau_M(t_0, D, t_1)$, is the interval function

$$SC_M(t_0, \langle \overline{m_1}, \ldots, \overline{m_n} \rangle, t_1) + \sum_{i=1}^{n} \left( \widehat{\frac{\partial sc}{\partial (u)_i}} \right) (I_i)\ (I_i - \overline{m_i}) + E_M(t_0, D_0, t_1)$$

In the above definiton, the interval function $\left( \widehat{\frac{\partial sc}{\partial (u)_i}} \right)(I_i)$ can be evaluated by automatic differentiation, during the evaluation of $SC(t_0, I_i, t_1)$. The definition also generalizes to multistep interval extensions.

### 5.5 Implementation Issues

Several of the novel techniques proposed in this section can be reduced to unconstrained optimization problems. In general, interval techniques for unconstrained optimization problems require the function to satisfy a stability requirement (i.e., the optimum is not on the frontier of the box defining the search space). This requirement is not guaranteed in this context since, by Theorem 5, we know that the minimum of function $s$ is on the frontier of $D_0$, and we minimize function $sc$, an approximation of $s$.

**Definition 19 min-stability.** A function $g$ is *min-stable* for box $K = \langle K_1, \ldots, K_n \rangle \subseteq \Re^n$ if there exists some $\epsilon > 0$ such that

$$min(g(K)) = min(g(K'))$$

with $K' = \langle K_1 + [-\epsilon, \epsilon], \ldots, K_n + [-\epsilon, \epsilon] \rangle$.

Let $K = \langle K_1, \ldots, K_n \rangle \subseteq \Re^n$ be a box and $g : \Re^n \rightarrow \Re$ be a function to minimize in $K$. Here are some necessary conditions for a point $d$ in $K$ to be a minimum when the function is not min-stable.

$$\begin{array}{ll} \frac{\partial g}{\partial x_i}(d) = 0 & \text{if } d_i \text{ is in the interior of } K_i\ (left(K_i) < d_i < right(K_i)) \\ \frac{\partial g}{\partial x_i}(d) \geq 0 & \text{if } d_i = left(K_i) \\ \frac{\partial g}{\partial x_i}(d) \leq 0 & \text{if } d_i = right(K_i) \end{array}$$

Traditional interval algorithms for unconstrained minimization can be generalized to include the interval meta-constraints

$$left(I_i) \neq left(K_i) \wedge right(I_i) \neq right(K_i) \Rightarrow \left( \widehat{\frac{\partial g}{\partial x_i}} \right)(D) = 0$$

$$left(I_i) = left(K_i) \wedge right(I_i) \neq right(K_i) \Rightarrow \left( \widehat{\frac{\partial g}{\partial x_i}} \right)(D) \geq 0$$

$$left(I_i) \neq left(K_i) \wedge right(I_i) = right(K_i) \Rightarrow \left( \widehat{\frac{\partial g}{\partial x_i}} \right)(D) \leq 0$$

with $\widehat{\left(\frac{\partial g}{\partial x_i}\right)}$ an interval extension of $\left(\frac{\partial g}{\partial x_i}\right)$, and $D = \langle I_1, \ldots, I_n \rangle$. The search can also be restricted to the frontier by adding the redundant constraint

$$\bigvee_{1 \leq i \leq n} left(I_i) = left(K_i) \vee right(I_i) = right(K_i)$$

and applying techniques such as constructive disjunction [VHSDar]. Note that combining these two necessary conditions require some extra care to preserve correctness.

## 6 Backwards Pruning: Box-Consistency for ODE

This section proposes another technique to address the growth of intervals in the step methods. The fundamental intuition here is illustrated in Figure 3. We know that all the solutions at $t_{j-1}$ are in $D_{j-1}$. If, in $D_j$, there is some box $H$ such that $S(t_j, H, t_{j-1}) \cap D_{j-1} = \emptyset$, then we know that the box $H$ is *not* part of the solution at $t_j$. In other words, it is possible to use the step methods backwards to determine whether pieces of the box can be pruned away. This section formalizes this idea in terms of box-consistency.

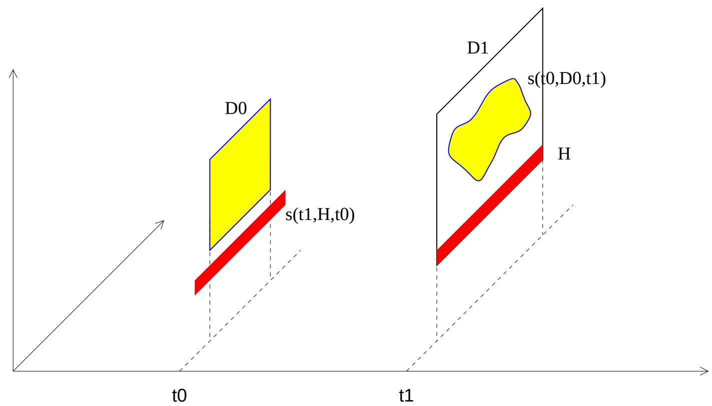


**Fig. 3.** Pruning

Box consistency aims at reducing a box $D_j$ at $t_j$ given that the solution at $t_{j-1}$ are known to be in $D_{j-1}$.

**Definition 20 Interval Projection of an ODE System.** An *interval projection ODE* $\langle S, i \rangle$ is the association of an interval solution $S$ and of an index $i$ $(1 \leq i \leq n)$.

**Definition 21 Box Consistency of an ODE System.** Let $S$ be an interval solution of an ODE system $\mathcal{O}$. An interval projection ODE $\langle S, i \rangle$ is *box-consistent* at $t_1, D_1$ wrt $t_0, D_0$ if

$$I_i = \square\{\ r_i \ \in \ I_i \mid \emptyset \neq D_0 \cap S(t_1, \langle I_1, ..., I_{i-1}, \overline{r_i}, I_{i+1}, ..., I_n \rangle, t_0)\}$$

where $D_1 = \langle I_1, \ldots, I_n \rangle$. An interval solution is *box-consistent* at $t_1, D_1$ wrt $t_0, D_0$ if its projections are box-consistent at $t_1, D_1$ wrt $t_0, D_0$.

**Proposition 22.** *Let $D_1 = \langle I_1, \ldots, I_n \rangle$, and $I_i = [l_i, r_i]$. An interval projection ODE $\langle S, i \rangle$ is* box-consistent *at $t_1, D_1$ wrt $t_0, D_0$ iff, when $l_i \neq r_i$,*

$$\emptyset \neq D_0 \cap S(t_1, \langle I_1, ..., I_{i-1}, [l_i, l_i^+], I_{i+1}, ..., I_n \rangle, t_0) \ \wedge \ \emptyset \neq D_0 \cap S(t_1, \langle I_1, ..., I_{i-1}, [r_i^-, r_i], I_{i+1}, ..., I_n \rangle, t_0)$$

*and, when $l_i = r_i$,*

$$\emptyset \neq D_0 \cap S(t_1, \langle I_1, ..., I_{i-1}, [l_i, l_i], I_{i+1}, ..., I_n \rangle, t_1).$$

Traditional propagation algorithms can now be defined to enforce box-consistency of ODE systems.

## 7 The Wrapping Effect

The wrapping effect is the name given to the error resulting from the enclosure of a region (which is not a box) by a box. It only occurs for multidimensional function. In one dimension, a perfect interval extension of a continuous function $g$ always yields the correct interval. However, a perfect interval extension of a multidimentional function $g$ introduces overestimations in the resulting box, because the set $g(D) = \{g(d) | d \in D\}$ is not necessarily a box. This effect is especially important when the enclosure is used for finding a new region which is also enclosed by a box. The wrapping effect is thus central in interval methods for ODE. The following classical example, due to Moore [Moo66] and explained in [Cor95] , illustrates this problem :

$$u' = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} u \quad \text{with} \quad u_0 \in \begin{pmatrix} -0.1 & 0.1 \\ 0.9 & 1.1 \end{pmatrix}$$

The trajectories of individual point-valued solutions of this ODE are circles in the $((u)_1, (u)_2)$-phase space. The set of solution values is a rotated rectangle. Figure 4 shows that the resulting boxes at $t_{j-1}$, $t_j$, $t_{j+1}$. Moore shows that the width of the enclosures grow exponentially even if the stepwise $(t_j - t_{j-1})$ converges to zero. The wrapping effect can be reduced by changing the coordinate system at each step of the computation process. The idea is to choose a coordinate system more appropriate to the shape of $s(t_{j-1}, D_{j-1}, t_j)$, hence reducing the overestimation of the box representation of this set, as illustrated in Figure 5.

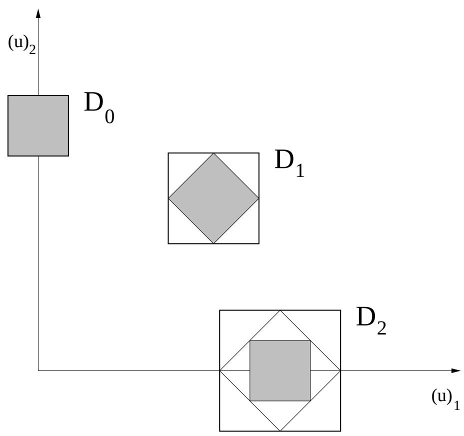


**Fig. 4.** The Wrapping Effect

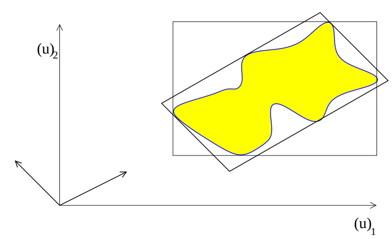


**Fig. 5.** Reducing the Overestimation by Coordinate Transformation

An appropriate coordinate system has to be chosen at each step. Assuming that such coordinate systems are given by mean of (invertible) matrices $M_j$, a naive approach, based on an explicit one-step method, would consist of computing

$$D_j := S(t_{j-1}, M_{j-1}.D'_{j-1}, t_j) \; ;$$
$$D'_j := M_j^{-1} D_j$$

where $D'_j$ and $D'_{j-1}$ are the boxes at $t_j$ and $t_{j-1}$ in their local coordinate system. This approach is naive since it introduces three wrapping effects: in $M_{j-1}D'_{j-1}$ to restore the original coordinate system needed to compute $S$, in the computation of $S$, and in the computation of $M_j^{-1}D_j$ to produce the result in the new coordinate system. To remedy this limitation, more advanced techniques (see, for instance, [Loh87, Ste71, DS76]) have been proposed but they are all bound to a specific step procedure. For instance, Lohner merges the two naive steps together using a mean value form and use associativity in the matrix products to try eliminating the wrapping effect. More precisely, the key term to be evaluated in his step method is of the form $(M_j^{-1} J M_{j-1}) D'_{j-1}$ and the goal is to choose $M_j^{-1}$ so that $M_j^{-1} J M_{j-1}$ is close to an identity matrix.

Piecewise interval extensions, however, reduce the wrapping effect in the naive method substantially, as illustrated in Figure 6. The overestimations of $M_{j-1}.D'_{j-1}$ and $M_j^{-1}D_j$ on $\epsilon$-boxes introduce wrapping

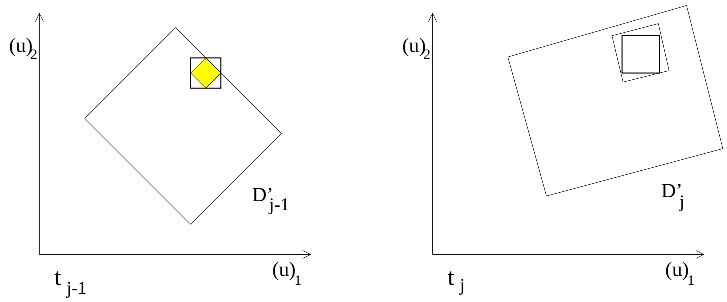


**Fig. 6.** Coordinate transformation on $\epsilon$-boxes

| t | Taylor | | Piecewise Taylor | | Exact solution |
|---|---|---|---|---|---|
| | *Result* | *Error* | *Result* | *Error* | |
| 0.0 | [-1.000000 , 1.000000] | 0.00% | [-1.000000 , 1.000000] | 0.00% | [-1.000000 , 1.000000] |
| 0.5 | [-1.648698 , 1.648698] | 171.82% | [-0.607031 , 0.607031] | 0.08% | [-0.606531 , 0.606531] |
| 1.0 | [-2.718205 , 2.718205] | 638.88% | [-0.368487 , 0.368487] | 0.17% | [-0.367879 , 0.367879] |
| 1.5 | [-4.481499 , 4.481499] | 1908.47% | [-0.223683 , 0.223683] | 0.25% | [-0.223130 , 0.223130] |
| 2.0 | [-7.388637 , 7.388637] | 5359.51% | [-0.135783 , 0.135783] | 0.33% | [-0.135335 , 0.135335] |
| 2.5 | [-12.181631 , 12.181631] | 14740.26% | [-0.082424 , 0.082424] | 0.41% | [-0.082085 , 0.082085] |
| 3.0 | [-20.083830 , 20.083830] | 40239.45% | [-0.050034 , 0.050034] | 0.50% | [-0.049787 , 0.049787] |
| 3.5 | [-33.112169 , 33.112169] | 109552.44% | [-0.030372 , 0.030372] | 0.58% | [-0.030197 , 0.030197] |
| 4.0 | [-54.591963 , 54.591963] | 297962.02% | [-0.018437 , 0.018437] | 0.66% | [-0.018316 , 0.018316] |

**Table 1.** ODE $u'(t) = -u(t)$

effects that are small compared to the overall size of the box and to the benefits of using piecewise interval extensions. In addition, this reduction of the wrapping effect is not tailored to a specific step method. The basic idea is thus (1) to find a linear approximation of $s(t_{j-1}, M_{j-1}.D'_{j-1}, t_j)$; (2) to compute the matrix $M_j^{-1}$ from the linear relaxation; (3) to apply the naive method on $\epsilon$-boxes. Step (1) can be obtained by using, for instance, a Taylor extension, while Step (2) can use Lohner's method that consists of obtaining a QR factorization of the linear relaxation. Lohner's method has the benefit of being numerically stable.

## 8 Experimental Results

This section compares some standard techniques with piecewise interval extensions. This goal is to show that consitency techniques can bring substantial gain in precision. The results were computed with Numerica with a precision of `1e-8`, using optimal bounding boxes.

Consider the ODE $u'(t) = -u(t)$ for an initial box [-1,1] at $t_0 = 0$. Table 1 compares the results obtained by an interval Taylor method of order 4 with step size 0.5, the results obtained by the piecewise interval extension of the same method, and the exact solutions. Relative errors on the size of the boxes are also given. As can be seen, the intervals of the traditional Taylor method grow quickly, although this function is actually contracting. The piecewise interval extension, on the other hand, is close to the exact solutions and is able to exploit the contraction characteristics of the function.

Consider now the ODE $u'(t) = -u^2(t)$ for an initial box $[0.1, 0.4]$ at $t_0 = 0$. Table 2 compares the results obtained by a mean value form of a Taylor method of order 4, the results obtained by the piecewise interval extension of the Taylor method of order 4, and the exact solutions. Once again, it can be seen that the standard method leads to an explosion of the size of the intervals, while the piecewise interval extension is close to the exact results. Note that the Taylor method of order 4 also behaves badly on this ODE.

| t | Taylor MVF | | Piecewise Taylor | | Exact solution |
|---|---|---|---|---|---|
| | *Result* | *Error* | *Result* | *Error* | |
| 0.0 | [0.10000 , 0.40000] | 0.00% | [0.10000 , 0.40000] | 0.00% | [0.10000 , 0.40000] |
| 0.5 | [0.06798 , 0.37635] | 29.52% | [0.09511 , 0.33344] | 0.10% | [0.09524 , 0.33333] |
| 1.0 | [0.03884 , 0.36099] | 65.37% | [0.09075 , 0.28583] | 0.14% | [0.09091 , 0.28571] |
| 1.5 | [0.01027 , 0.35316] | 110.31% | [0.08679 , 0.25010] | 0.16% | [0.08696 , 0.25000] |
| 2.0 | [-0.02004 , 0.35314] | 168.68% | [0.08318 , 0.22231] | 0.18% | [0.08333 , 0.22222] |
| 2.5 | $[-\infty , +\infty]$ | | [0.07985 , 0.20007] | 0.19% | [0.08000 , 0.20000] |
| 3.0 | | | [0.07678 , 0.18188] | 0.19% | [0.07692 , 0.18182] |
| 3.5 | | | [0.07394 , 0.16672] | 0.20% | [0.07407 , 0.16667] |
| 4.0 | | | [0.07131 , 0.15389] | 0.21% | [0.07143 , 0.15385] |
| 4.5 | | | [0.06885 , 0.14290] | 0.21% | [0.06897 , 0.14286] |
| 5.0 | | | [0.06656 , 0.13337] | 0.21% | [0.06667 , 0.13333] |

**Table 2.** ODE $u'(t) = -u^2(t)$

| t | dim | Taylor | | Piecewise Taylor | | Exact solution |
|---|---|---|---|---|---|---|
| | | *Result* | *Error* | *Result* | *Error* | |
| 0.0 | u1 | [5.900000 , 6.100000] | 0.00% | [5.900000 , 6.100000] | 0.00% | [5.900000 , 6.100000] |
| | u2 | [3.900000 , 4.100000] | 0.00% | [3.900000 , 4.100000] | 0.00% | [3.900000 , 4.100000] |
| 0.1 | u1 | [4.754144 , 5.029167] | 24.43% | [4.781062 , 5.002249] | 0.07% | [4.781105 , 5.002139] |
| | u2 | [1.700997 , 1.922260] | 0.10% | [1.700997 , 1.922260] | 0.10% | [1.701061 , 1.922096] |
| 0.2 | u1 | [4.057424 , 4.422907] | 49.62% | [4.117875 , 4.362457] | 0.12% | [4.117981 , 4.362262] |
| | u2 | [0.065658 , 0.440812] | 53.58% | [0.130888 , 0.375582] | 0.17% | [0.131028 , 0.375309] |
| 0.3 | u1 | [3.652193 , 4.156882] | 86.94% | [3.769327 , 4.039751] | 0.17% | [3.769509 , 4.039480] |
| | u2 | [-1.192704 , -0.592271] | 122.41% | [-1.027760 , -0.757212] | 0.21% | [-1.027538 , -0.757566] |
| 0.4 | u1 | [3.433161 , 4.149383] | 140.05% | [3.641787 , 3.940761] | 0.20% | [3.642053 , 3.940418] |
| | u2 | [-2.238533 , -1.305898] | 212.58% | [-1.921764 , -1.622661] | 0.25% | [-1.921453 , -1.623088] |
| 0.5 | u1 | [3.321163 , 4.356458] | 213.97% | [3.673553 , 4.004076] | 0.24% | [3.673912 , 4.003656] |
| | u2 | [-3.197756 , -1.773026] | 332.07% | [-2.650709 , -2.320059] | 0.27% | [-2.650303 , -2.320559] |
| 0.6 | u1 | [3.249885 , 4.764365] | 315.58% | [3.824439 , 4.189827] | 0.26% | [3.824898 , 4.189321] |
| | u2 | [-4.177761 , -2.022052] | 491.54% | [-3.282651 , -2.917138] | 0.30% | [-3.282142 , -2.917718] |
| 0.7 | u1 | [3.154632 , 5.386877] | 454.25% | [4.068802 , 4.472723] | 0.29% | [4.069370 , 4.472121] |
| | u2 | [-5.284330 , -2.040935] | 705.31% | [-3.864643 , -3.460596] | 0.32% | [-3.864020 , -3.461270] |
| 0.8 | u1 | [2.961297 , 6.266973] | 642.67% | [4.390886 , 4.837398] | 0.32% | [4.391577 , 4.836685] |
| | u2 | [-6.638268 , -1.774736] | 992.66% | [-4.429812 , -3.983172] | 0.34% | [-4.429063 , -3.983955] |
| 0.9 | u1 | [2.573762 , 7.483245] | 898.02% | [4.781714 , 5.275302] | 0.34% | [4.782541 , 5.274461] |
| | u2 | [-8.394401 , -1.116121] | 1379.56% | [-5.002115 , -4.508393] | 0.37% | [-5.001224 , -4.509304] |
| 1.0 | u1 | [1.857660 , 9.161995] | 1243.56% | [5.237017 , 5.782644] | 0.36% | [5.237998 , 5.781654] |
| | u2 | [-10.766038 , 0.112700] | 1901.03% | [-5.599549 , -5.053780] | 0.39% | [-5.598498 , -5.054842] |

**Table 3.** ODE $u_1'(t) = -u_1(t) - 2u_2(t)$ and $u_2'(t) = -3u_1(t) - 2u_2(t)$

Consider now the system

$$\begin{cases} u_1'(t) = -u_1(t) - 2u_2(t) \\ u_2'(t) = -3u_1(t) - 2u_2(t) \end{cases}$$

for an initial box ([5.9,6.1],[3.9,4.1]) at $t_0 = 0$. Table 3 compares the results obtained by an interval Taylor method of order 4, the results obtained by the piecewise interval extension of the same method, and the exact solutions. Once again, similar results can be observed.

## 9 Conclusion

This paper studied the application of interval analysis and consistency techniques to ordinary differential equations. It presented a unifying framework to extend traditional numerical techniques to intervals showing, in particular, how to extend explicit and implicit, one-step and multi-step, methods to intervals. The paper also took a fresh look at the traditional problems encountered by interval techniques and studied how consistency techniques may help. It proposed to generalize interval techniques into a two-step process: a forward process that computes an enclosure and a backward process that reduces this enclosures. In addition, the paper studied how consistency techniques may help in improving the forward process and the wrapping effect. Very preliminary results indicate the potential benefits of the approach. Our current work focuses on the full implementation and experimental evaluation of the techniques proposed in this paper in order to determine which combinations of these techniques will be effective in practice. Future work will also be devoted to the application of consistency techniques to ODE systems with boundary values, since interval analysis and consistency techniques are particularly well-adapted when compared to traditional methods (as observed by several members of the community).

**Acknowledgment**

Many thanks to Philippe Delsarte and Jean Mawhin for fruitfull discussions. This research is partially supported by the *Actions de recherche concertées (ARC/95/00-187)* of the Direction générale de la Recherche Scientifique – Communauté Française de Belgique and by an NSF NYI award.

## References

[Abe88] Oliver Aberth. *Precise Numerical Analysis.* William Brown, Dubuque, Iowa, 1988.

[BBCG96] Martin Berz, Christian Bischof, George Corliss, and Andreas Griewank, editors. *Computational Differentiation: Techniques, Applications, and Tools.* SIAM, Philadelphia, Penn., 1996.

[Cor88] George F. Corliss. Applications of differentiation arithmetic. In Ramon E. Moore, editor, *Reliability in Computing*, pages 127–148. Academic Press, London, 1988.

[Cor95] G.F. Corliss. *Theory of Numerics in Ordinary and Partial Diffential Equations (W.A. Light, M. Machetta (Eds)*, volume Vol IV, chapter Guaranteed Error Bounds for Ordinary Differential Equations, pages 1–75. Oxford University Press, 1995.

[DJVH98] Y. Deville, M. Janssen, and P. Van Hentenryck. Consistency Techniques in Ordinary Differential Equations. Technical Report TR98-06, Université catholique de Louvain, July 1998.

[DS76] D.P. Davey and N.F. Stewart. Guaranteed Error Boubds for the Initial Value problem Using Polytope Arithmetic. *BIT*, 16:257–268, 1976.

[Har64] Ph. Hartman. *Ordinary Differential Equations.* Wiley, New York, 1964.

[Hen62] P. Henrici. *Discrete Variable Methods in Ordinary Differential Equations.* John Wiley & Sons, New York, 1962.

[Loh87] Rudolf J. Lohner. Enclosing the solutions of ordinary initial and boundary value problems. In Edgar W. Kaucher, Ulrich W. Kulisch, and Christian Ullrich, editors, *Computer Arithmetic: Scientific Computation and Programming Languages*, pages 255–286. Wiley-Teubner Series in Computer Science, Stuttgart, 1987.

[Moo66] R.E. Moore. *Interval Analysis.* Prentice-Hall, Englewood Cliffs, NJ, 1966.

[Moo79] R.E. Moore. *Methods and Applications of Interval Analysis.* SIAM Publ., 1979.

[Ral80] Louis B. Rall. Applications of software for automatic differentiation in numerical computation. In Götz Alefeld and R. D. Grigorieff, editors, *Fundamentals of Numerical Computation (Computer Oriented Numerical Analysis)*, Computing Supplement No. 2, pages 141–156. Springer-Verlag, Berlin, 1980.

[Ral81] Louis B. Rall. *Automatic Differentiation: Techniques and Applications*, volume 120 of *Lecture Notes in Computer Science.* Springer-Verlag, Berlin, 1981.

[Sta96] O. Stauning. Enclosing Solutions of Ordinary Differential Equations. Technical Report Tech. Report IMM-REP-1996-18, Technical University Of Denmark, 1996.

[Ste71] N.F. Stewart. A Heuristic to Reduce the Wrapping Effect in the Numerical Solution of ODE. *BIT*, 11:328–337, 1971.

[VHLD97] P. Van Hentenryck, M. Laurent, and Y. Deville. *Numerica, A Modeling Language for Global Optimization.* MIT Press, 1997.

[VHSDar] P. Van Hentenryck, V. Saraswat, and Y. Deville. The Design, Implementation, and Evaluation of the Constraint Language `cc(FD)`. *Journal of Logic Programming (Special Issue on Constraint Logic Programming)*, 1998 (to appear).

This article was processed using the LaTeX macro package with LLNCS style