Multistep Filtering Operators for Ordinary Differential Equations

Micha Janssen, Yves Deville, and Pascal Van Hentenryck
Université catholique de Louvain,
Pl. Ste Barbe 2,
B-1348 Louvain-la-Neuve, Belgium
{mja,yde,pvh}@info.ucl.ac.be

Abstract

Interval methods for ordinary differential equations (ODEs) provide guaranteed enclosures of the solutions and numerical proofs of existence and unicity of the solution. Unfortunately, they may result in large over-approximations of the solution because of the loss of precision in interval computations and the wrapping effect. The main open issue in this area is to find tighter enclosures of the solution, while not sacrificing efficiency too much.

This paper takes a constraint satisfaction approach to this problem, whose basic idea is to iterate a forward step to produce an initial enclosure with a pruning step that tightens it. The paper focuses on the pruning step and proposes novel multistep filtering operators for ODEs. These operators are based on interval extensions of a multistep solution that are obtained by using (Lagrange and Hermite) interpolation polynomials and their error terms. The paper also shows how traditional techniques (such as mean-value forms and coordinate transformations) can be adapted to this new context. Preliminary experimental results illustrate the potential of the approach, especially on stiff problems, well-known to be very difficult to solve.

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

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

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 $u(t_{init}) = u_{init}$ and assuming existence and uniqueness of the solution, the solution of $\mathcal{O}$ is a function $s^* : \mathbb{R} \rightarrow \mathbb{R}^n$ satisfying $\mathcal{O}$ and the initial condition $s^*(t_{init}) = u_{init}$. 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.

Discrete variable methods aim to approximate the solution $s^*(t)$ of an ODE system at some points $t_0, t_1, \ldots, t_m$. They 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 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 can only return approximations since they ignore error terms.

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 $t_0, t_1, \ldots, t_m$. 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 $t_0, t_1, \ldots, t_m$ that is due to two reasons. On the one hand, step methods have a tendency to accumulate errors from point to point. On the other, the approximation of an arbitrary region by a box, called the wrapping effect, may introduce considerable imprecision after a number of steps. Much research has been devoted to address this problem. 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.

A Constraint Satisfaction Approach Our research takes a constraint satisfaction approach to the problem of producing tighter enclosures. The basic idea [DJVH98] is to view the solving of ODEs as the iteration of two steps: a forward process that produces an initial enclosure of the solution at a given time (given enclosures at previous times) and a pruning process that tightens this first enclosure. Our previous results, as most research in interval methods, mostly focused on the forward process. Our current research, in contrast, concentrates on the pruning step, where constraint satisfaction techniques seem particularly well adapted.

It is important to mention that taking a constraint satisfaction approach gives a fundamentally new perspective on this problem. Instead of trying to adapt traditional numerical techniques to intervals, the constraint satisfaction approach looks at the problem in a more global way and makes it possible to exploit a wealth of mathematical results. In this context, the basic methodology consists of finding necessary conditions on the solution that can be used for pruning. This paper will also show experimentally that the forward and backward steps are in fact orthogonal, clearly showing the interest of the approach. We thus may hope that constraint satisfaction will be as fruitful for ODEs as for combinatorial optimization and nonlinear programming.

Goal of the Paper As mentioned, the main goal of this paper is to design filtering algorithms to produce tighter enclosures of the solution. The problem is difficult because, contrary to traditional discrete or continuous problems, the constraints cannot be used directly since they involve unknown functions (and their derivatives). The key idea of the paper is to show that effective multistep filtering operators can be obtained by using conservative approximations of these unknown functions. These approximations can be obtained by using polynomial interpolations and their error terms. Once these multistep filtering operators are available, traditional constraint satisfaction techniques (e.g., box(k)-consistency [VHLD97]) can be applied to prune the initial enclosure.

Contributions This paper contains three main contributions. First, it proposes a generic filtering operator based on interval extensions of a multistep solution function and its derivatives. Second, it shows how these interval extensions can be obtained using Lagrange and Hermite interpolation polynomials. Third, it shows how the filtering operator can accommodate standard techniques such as mean-value forms and coordinate transformations to address the wrapping effect during the pruning step as well. The paper also contains some preliminary experimental evidence to show that the techniques are effective in tightening the initial enclosures.

Organization The rest of this paper is organized as follows. Sections 2 and 3 set up the background and recall the constraint satisfaction approach from [DJVH98]. Sections 4 and 5 are the core of the paper: Section 4 contains the novel generic multistep pruning operator, while Section 5 shows how to build the interval approximations it needs using Lagrange and Hermite polynomials. Section 6 describes advanced techniques, i.e., how to adapt the multistep filtering operator to the mean-value form and to local coordinate transformations. Section 7 discusses the implementation issues. Section 8 contains the experimental results and Section 9 concludes the paper. The proof of all results are given in the appendix.

2 Background and Definitions

This section briefly reviews the notational conventions and the main definitions used in this paper. Additional information can be found in [AH83, Neu90, Moo79, VHLD97, DJVH98]. Note that Section 2.2 contains generalizations of the standard definitions needed for this paper.

2.1 Basic Notational Conventions

The following conventions are adopted in this paper. (Sequences of) small letters denote real values, vectors and functions of real values. (Sequences of) capital letters denote real matrices, sets, intervals, vectors and functions of intervals. Capital letters between square brackets denote interval matrices. Bold face small letters denote sequences (delimited by "$\langle$" and "$\rangle$") of real values. Bold face capital letters denote sequences (delimited by "$\langle$" and "$\rangle$") of intervals. All these (sequences of) letters may be subscripted.

We use traditional conventions for abstracting floating-point numbers. If $\mathcal{F}$ is a floating-point system, the elements of $\mathcal{F}$ are called $\mathcal{F}$-numbers. If $a \in \mathcal{F}$, then a^+ denotes the smallest $\mathcal{F}$-number strictly greater than a and a^- the largest $\mathcal{F}$-number strictly smaller than a. $\mathcal{I}$ denotes the set of all *closed* intervals $\subseteq \mathbb{R}$ whose bounds are in $\mathcal{F}$. A vector of intervals $D \in \mathcal{I}^n$ is called a *box*. If $r \in \mathbb{R}$, then $\overline{r}$ denotes the smallest interval $I \in \mathcal{I}$ such that $r \in I$. If $r \in \mathbb{R}^n$, then $\overline{r} = (\overline{r_1}, ..., \overline{r_n})$. If $A \subseteq \mathbb{R}^n$, then $\Box A$ denotes the smallest box $D \in \mathcal{I}^n$ such that $A \subseteq D$. We also assume that $t_0, \ldots, t_k$, t_e and t are reals, $u_0, \ldots, u_k$ are in $\mathbb{R}^n$, and $D_0, \ldots, D_k$ are in $\mathcal{I}^n$. Finally, we use $\mathbf{t}_k$ to denote $\langle t_0, ..., t_k \rangle$, $\mathbf{u}_k$ to denote $\langle u_0, ..., u_k \rangle$, $T_{\mathbf{t}_k}$ to denote the interval $[\min(t_0, ..., t_k), \max(t_0, ..., t_k)]$, $T_{\mathbf{t}_k,t}$ to denote the interval $[\min(t_0, ..., t_k, t), \max(t_0, ..., t_k, t)]$ and $\mathbf{D}_k$ to denote $\langle D_0, ..., D_k \rangle$. The following definitions are standard.

Definition 1 Let A, B be sets, $a \in A$, g a function and r a relation defined on A, and op $\in \{+, -, \cdot, /\}$. Then, $g(A) = \{g(x) \mid x \in A\}$, $r(A) = \bigvee_{x \in A} r(x)$, A op $B = \{x \text{ op } y \mid x \in A, y \in B\}$ and a op $A = \{a \text{ op } x \mid x \in A\}$.

2.2 Interval Extensions of Partial Functions

We assume traditional definitions of interval extensions for functions and relations. In addition, if f (resp. c) is a function (resp. relation), then F (resp. C) denotes an interval extension of f (resp. C). We also overload traditional real operators (e.g., +, *, =, ...) and use them for their interval extensions. Because the techniques proposed in this paper use multistep solutions (which are partial functions), it is necessary to define interval extensions of partial functions and relations.

Definition 2 (Interval Extension of a Partial Function) The interval function $G : \mathcal{I}^n \to \mathcal{I}^m$ is *an interval extension* of the partial function $g : E \subseteq \mathbb{R}^n \to \mathbb{R}^m$ if

$$\forall D \in \mathcal{I}^n : g(E \cap D) \subseteq G(D).$$

Definition 3 (Interval Extension of a Partial Relation) The interval relation $R \subseteq \mathcal{I}^n$ is an *interval extension* of the partial relation $r \subseteq E \subseteq \mathbb{R}^n$ if

$$\forall D \in \mathcal{I}^n : r(E \cap D) \Rightarrow R(D).$$

In the context of ODEs, it is also useful to define interval extensions with respect to some variables only.

Notation 1 Let $g : (x, y) \mapsto g(x, y)$. Then, $g(a, \bullet)$ denotes the unary function

$$g(a, \bullet) : y \mapsto g(a, y).$$

A similar definition holds for $g(\bullet, a)$. The generalization to n-ary functions is straightforward.

Definition 4 (Interval Extension wrt a Subset of the Variables) The function $G : \mathcal{I} \times \mathbb{R} \to \mathcal{I}$ is *an interval extension* of the function $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ *wrt the* 1^{st} *variable* if the function $G(\bullet, a)$ is an interval extension of $g(\bullet, a)$ for all $a \in \mathbb{R}$. A similar definition holds for an interval extension wrt the 2^{nd} variable. The generalization to $\mathbb{R}^n \to \mathbb{R}^m$ (partial) functions is straightforward.

2.3 ODE Systems

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

Definition 5 (Solution of an ODE System with Initial Value) The *solution* of an ODE system $\mathcal{O}$ with initial conditions $u(t_{init}) = u_{init}$ is the function $s^*(t) : \mathbb{R} \to \mathbb{R}^n$ satisfying $\mathcal{O}$ and the initial conditions $s^*(t_{init}) = u_{init}$.

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 well-known [Moo79, DJVH98]. 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 6 (Solution of an ODE System) The *solution* of an ODE system $\mathcal{O}$ is the function $s(t_0, u_0, t) : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ such that $s(t_0, u_0, t) = s^*(t)$, where s^* is the solution of $\mathcal{O}$ with initial conditions $u(t_0) = u_0$.

This definition can be generalized to multistep functions.

Definition 7 (Multistep solution of an ODE) *The multistep solution* of an ODE system $\mathcal{O}$ is the *partial* function $ms : A \subseteq (\mathbb{R}^{k+1} \times (\mathbb{R}^n)^{k+1} \times \mathbb{R}) \to \mathbb{R}^n$ defined by

$$\begin{array}{rcll} ms(\mathbf{t}_k, \mathbf{u}_k, t) & = & s(t_0, u_0, t) & \text{if } \ u_i = s(t_0, u_0, t_i) \text{ for } \ 1 \le i \le k, \\ & & \text{undefined} & \text{otherwise} \end{array}$$

where s is the solution of $\mathcal{O}$.

It is important to stress that the multistep function is a partial function. Hence, interval extensions of multistep functions may behave very differently outside the domain of definition of the functions. This fact is exploited by the novel filtering operators proposed in this paper.

Finally, we generalize the concept of bounding boxes, a fundamental concept in interval methods for ODEs, to multistep methods. Intuitively, a bounding box encloses all solutions of an ODE going through certain boxes at given times.

Definition 8 (Bounding box) Let $\mathcal{O}$ be an ODE system, ms be the multistep solution of $\mathcal{O}$, and $\{t_0, ..., t_k\} \subseteq T \in \mathcal{I}$. A box B is a *bounding box* of ms over T wrt $\mathbf{D}_k$ and $\mathbf{t}_k$, if, for all $t \in T$, $ms(\mathbf{t}_k, \mathbf{D}_k, t) \subseteq B$.

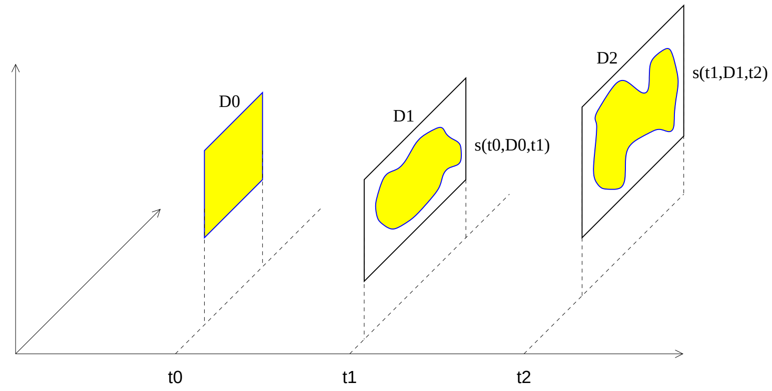

Figure 1: Intuition of the STEP procedure

3 The Constraint Satisfaction Approach

The constraint satisfaction approach followed in this paper was first presented in [DJVH98]. It consists of a generic algorithm for ODEs that iterates two steps: (1) a *forward* step that computes an initial enclosure at a given time from enclosures at previous times and bounding boxes and (2) a *pruning* step that reduces the initial enclosures without removing solutions. The forward process also provides numerical proofs of existence and unicity of the solution. The intuition underlying the basic step of the generic algorithm is illustrated in Figure 1.

Various techniques were presented in [DJVH98] for the forward step and these are not discussed here. In contrast, this paper focuses on the pruning step, which is, in fact, the main novelty of the approach. The pruning step prunes the last box D_k at t_k produced by the forward step, using, say, the last k boxes $D_0, \ldots, D_{k-1}$ obtained at times $t_0, \ldots, t_{k-1}$.[1] To our knowledge, no research has been devoted to pruning techniques for ODEs, except for the proposal in [DJVH98] to use the forward step backwards. These techniques however are promising since they open new directions to tackle the traditional problems of interval methods for ODEs.

4 A Multistep Filtering Operator for ODEs

This section presents a multistep filtering operator for ODEs to tighten the initial enclosure of the solutions effectively. It starts with an informal presentation to convey the main ideas and intuitions before formalizing the concepts.

4.1 Overview

To understand the main contribution of this paper, it is useful to contrast the techniques proposed herein with interval techniques for nonlinear equations. In nonlinear programming, a constraint $c(x_1, \ldots, x_n)$ can be used almost directly for pruning the search space (i.e., the carthesian products of the intervals I_i associated with the variables x_i). It suffices to take an interval extension $C(X_1, \ldots, X_n)$ of the constraint. Now if $C(I'_1, \ldots, I'_n)$ does not hold, it follows, by definition of interval extensions, that no solution of c lies in $I'_1 \times \ldots \times I'_n$. This basic property can be seen as a filtering operator that can be used for pruning the search space in many ways, including box(k)-consistency as in Numerica [VHLD97, VH98b]. Recall that a constraint C is box(1)-consistent wrt

[1] Note that the time t_0 is not, in general, the time t_{init} of the initial condition.

$I_1, \ldots, I_n$ and x_i if the condition

$$C(I_1, \ldots, I_{i-1}, [l_i, l_i^+], I_{i+1}, \ldots, I_n) \wedge C(I_1, \ldots, I_{i-1}, [u_i^-, u_i], I_{i+1}, \ldots, I_n)$$

holds where $I_i = [l_i, u_i]$. The filtering algorithm based on box(1)-consistency reduces the interval of the variables without removing any solution until the constraint is box(1)-consistent wrt the intervals and all variables. Stronger consistency notions, e.g., box(2)-consistency, are also useful for especially difficult problems [VH98a]. It is interesting here to distinguish the filter or pruning operator, i.e., the technique used to determine if a box cannot contain a solution, from the filtering algorithm that uses the pruning operator in a specific way to prune the search space.

The goal of the research described in this paper is to device similar techniques for ODEs. The main difficulty is that there is no obvious filter in this context. Indeed, the equation $u' = f(t, u)$ cannot be used directly since u and u' are unknown functions. We now discuss how to overcome this problem and, in a first step, restrict attention to one-dimensional problems for simplicity.

Assume first that we have at our disposal the multistep solution ms of the equation. In this case, the equation $u' = f(t, u)$ can be rewritten into

$$\frac{\partial ms}{\partial t}(\langle t_0, \ldots, t_k \rangle, \langle v_0, \ldots, v_k \rangle, t) = f(t, ms(\langle t_0, \ldots, t_k \rangle, \langle v_0, \ldots, v_k \rangle, t)).$$

Let us denote this equation

$$fl(\langle t_0, \ldots, t_k \rangle, \langle v_0, \ldots, v_k \rangle, t).$$

At first sight, of course, this equation may not appear useful since ms is still an unknown function. However, as Section 5 shows, it is possible to obtain interval extensions of ms and $\frac{\partial ms}{\partial t}$ by using, say, polynomial interpolations together with their error terms. If MS and DMS are such interval extensions, then we obtain an interval equation

$$DMS(\langle t_0, \ldots, t_k \rangle, \langle X_0, \ldots, X_k \rangle, t) = F(\bar{t}, MS(\langle t_0, \ldots, t_k \rangle, \langle X_0, \ldots, X_k \rangle, t))$$

that can be used as a filtering operator. Let us denote this operator by

$$FL(\langle t_0, \ldots, t_k \rangle, \langle X_0, \ldots, X_k \rangle, t)$$

and illustrate how it can prune the search space. If the condition

$$FL(\langle t_0, \ldots, t_k \rangle, \langle I_0, \ldots, I_k \rangle, t)$$

does not hold, then it follows that no solution of $u' = f(t, u)$ can go through intervals $I_0, \ldots, I_k$ at times $t_0, \ldots, t_k$.

How can we use this filter to obtain tighter enclosures of the solution? A simple technique consists of pruning the last interval produced by the forward process. Assume that I_i is an interval enclosing the solution at time t_i ($0 \leq i \leq k$) and that we are interested in pruning the last interval I_k. A subinterval $I \subseteq I_k$ can be pruned away if the condition

$$FL(\langle t_0, \ldots, t_k \rangle, \langle I_0, \ldots, I_{k-1}, I \rangle, t_e)$$

does not hold for some evaluation point t_e.

Let us explain briefly the geometric intuition behind this formula. Figure 2 is generated from an actual ordinary differential equation, considers only points instead of intervals, and ignores error terms for simplicity. It illustrates how this technique can prune away a value as a potential solution at a given time. In the figure, we consider the solution to the equation that evaluates to u_0 and u_1 at t_0 and t_1 respectively. Two possible points u_2 and u_2' are then considered as possible values at t_2. The curve marked `KO` describes an interpolation polynomial going through u_0, u_1, u_2' at times t_0, t_1, t_2. To determine if u_2' is the value of the solution at time t_2, the idea is to test if the equation

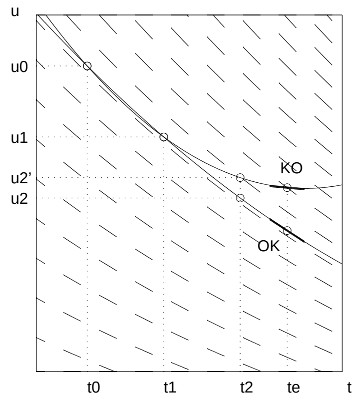

Figure 2: Geometric Intuition of the Multistep Filtering Operator

is satisfied at times t_e. (We will say more about how to choose t_e later in this paper). As can be seen easily, the slope of the interpolation polynomial is different from the slope specified by f at time t_e and hence u_2' cannot be the value of the solution at t_2. The curve marked `OK` describes an interpolation polynomial going through u_0, u_1, u_2 at times t_0, t_1, t_2. In this case, the equation is satisfied at time t_e, which means that u_2 cannot be pruned away.

Of course, the filter proposed earlier generalizes this intuition to intervals. The interval function DMS is an interval extension of $\frac{\partial ms}{\partial t}$ obtained, say, by taking an interval extension of the derivative of an interpolation polynomial and a bound on its error term. The interval function MS is an interval extension of an interpolation polynomial and a bound on its error term. These interval functions are evaluated over intervals produced by the forward process. The filtering operator thus tests whether a solution can go through interval I by *testing* this interval equation at time t_e. If a solution goes through I, then the filter must hold because the left- and the righ-hand sides of the filter are both interval extensions of $\frac{\partial ms}{\partial t}$. If I does not contain a solution, by definition of partial interval extension (See Definition 3), no constraints are imposed on MS and DMS and there is no reason to believe that the filter will hold. It may hold because of a loss of precision in the computation or because we are unlucky but a careful choice of the interpolation polynomials will minimize these risks.

It is important to stress that traditional consistency techniques and filtering algorithms based on this filtering operator can now be applied. For instance, one may be interested in computing the set

$$I_k' = \Box\{r \in I_k \mid FL(\langle t_0, \ldots, t_k \rangle, \langle I_0, \ldots, I_{k-1}, \overline{r} \rangle, t_e)\}.$$

For multi-dimensional problems, one may be interested in obtaining box(k)-approximations of the multi-dimensional sets defined in a similar fashion.

It is also important to mention that the filtering operator can be used in many different ways, even if only the last interval (or box) is considered for pruning. For instance, once an interval $I \subseteq I_k$ is selected, it is possible to prune the intervals $I_0, \ldots, I_{k-1}$ using, say, the forward process run backwards as already suggested in [DJVH98]. This makes it possible to obtain tighter enclosures of ms and $\frac{\partial ms}{\partial t}$, thus obtaining a more effective filtering algorithm for I.

Finally, it is useful to stress that the filtering operator suggested here shares some interesting connections with Gear's method, a traditional implicit multistep procedure that is particularly useful for stiff problems. We may thus hope that the filtering operator will be particularly well adapted for stiff problems as well (as our prelimininary results show). We will say more about these connections once some more technical details have been given.

4.2 Formalization

We now formalize the intuition given in the previous subsection. A multistep filtering operator is defined as an interval extension of the original equation rewritten to make the multistep solution explicit.

Definition 9 (Multistep Filtering Operator) Let $\mathcal{O}$ be an ODE $u' = f(t,u)$ and let *ms* be the multistep solution of $\mathcal{O}$. A *multistep filtering* operator for $\mathcal{O}$ is an interval extension of the constraint

$$\frac{\partial ms}{\partial t}(\mathbf{t}_k, \mathbf{u}_k, t_e) = f(t_e, ms(\mathbf{t}_k, \mathbf{u}_k, t_e))$$

wrt the variables in $\mathbf{u}_k$.

It can be shown that a multistep filtering operator never prunes solutions away.

Proposition 1 (Soundness of the Multistep Filtering Operator) Let $\mathcal{O}$ be an ODE $u' = f(t,u)$, let FL be a multistep filtering operator for $\mathcal{O}$. If $FL(\mathbf{t}_k, \mathbf{D}_k, t_e)$ does not hold, then there exists no solution of $\mathcal{O}$ going through $\mathbf{D}_k$ at times $\mathbf{t}_k$.

The intuition given previously was based on a natural multistep filtering operator.

Definition 10 (Natural Multistep Filtering Operator) Let $\mathcal{O}$ be an ODE $u' = f(t,u)$, let *ms* be the multistep solution of $\mathcal{O}$, let F be an interval extension of f, and let *MS* and *DMS* be interval extensions of *ms* and $\frac{\partial ms}{\partial t}$ wrt to their second argument. A *natural multistep filtering* operator for $\mathcal{O}$ is an interval equation

$$DMS(\langle t_0, \ldots, t_k \rangle, \langle X_0, \ldots, X_k \rangle, t_e) = F(\overline{t_e}, MS(\langle t_0, \ldots, t_k \rangle, \langle X_0, \ldots, X_k \rangle, t_e)).$$

There are other interesting multistep filtering operators, e.g., the mean-value form of the natural multistep filtering operator (see section 6). Different multistep filtering operators may be more appropriate when close or far from a solution as was already the case for nonlinear equations [VHLD97].

It remains to show how to obtain interval extensions of the solution function *ms* and its derivative $\frac{\partial ms}{\partial t}$.

5 Interval Extensions of the Solution Function

This section is devoted to interval extensions of the multistep solution function and its derivative. These extensions are, in general, based on decomposing the (unknown) multistep function into the summation of a computable approximation p and an (unknown) error term e, i.e.,

$$ms(\mathbf{t}_k, \mathbf{u}_k, t) = p(\mathbf{t}_k, \mathbf{u}_k, t) + e(\mathbf{t}_k, \mathbf{u}_k, t). \tag{1}$$

There exist standard techniques to build p and to bound e. In the rest of this section, two such approximations are presented. We also show how to bound the error term of the derivative of the multistep solution functions, since these are critical to obtain multistep filtering operators.

5.1 A Lagrange Polynomial Interval Extension

Our first interval extension is based on Lagrange polynomial interpolation.

Definition 11 (Lagrange Polynomial [KC96]) Let $t_0, ..., t_k$ be distinct points. The *Lagrange polynomial* that interpolates points $(t_0, u_0), ..., (t_k, u_k)$ is the *unique* polynomial $p_L : \mathbb{R} \to \mathbb{R}^n$ of degree $\leq k$ satisfying $p_L(t_i) = u_i$ $(0 \leq i \leq k)$. It is defined by

$$p_L(t) = \sum_{i=0}^{k} u_i \varphi_i(t),$$

where

$$\varphi_i(t) = \frac{\prod_{j=0, j\neq i}^{k}(t - t_j)}{\prod_{j=0, j\neq i}^{k}(t_i - t_j)}, \quad 0 \leq i \leq k.$$

It is possible to bound the errors made by a Lagrange polynomial when interpolating a function.

Theorem 1 (Lagrange Error Term) *Let $a, b \in \mathbb{R}$, g be a function in $C^{k+1}([a, b], \mathbb{R}^n)$, let p_L be the Lagrange polynomial of degree $\leq k$ that interpolates g at $k + 1$ distinct points $t_0, ..., t_k$ in the interval $[a, b]$, and $t \in ([a, b] \setminus T_{\mathbf{t}_k}) \cup \{t_0, ..., t_k\}$. Then,*

1. $\exists \zeta_t \in]a, b[: g(t) - p_L(t) = \frac{1}{(k+1)!} g^{(k+1)}(\zeta_t) w(t);$
2. $\exists \xi_t \in]a, b[: g'(t) - p'_L(t) = \frac{1}{(k+1)!} g^{(k+1)}(\xi_t) w'(t).$

where $w(t) = \prod_{i=0}^{k}(t - t_i)$.

This result makes it possible to obtain interval extensions of ms and its derivative. Consider, for instance, function ms. The key idea to obtain an interval extension of ms consists of considering $f^{(k)}(\zeta_t, ms(\mathbf{t}_k, \mathbf{u}_k, \zeta_t))$ in the error term obtained from Theorem 1 and of

1. replacing the unknown time ζ_t by the interval $[a, b]$ in which it takes its value;
2. replacing the unknown function ms by one of its bounding boxes.

Together, these ideas gives conservative approximations of the error terms and thus interval extensions of the multistep solution function and of its derivative.

Definition 12 (Lagrange Interval Polynomial) Let $\mathcal{O}$ be an ODE $u' = f(t, u)$ and ms be the multistep solution of $\mathcal{O}$. A *Lagrange Interval Polynomial* and its derivative for $\mathcal{O}$ are the functions MS_L and DMS_L respectively defined by

$$\begin{aligned} MS_L(\mathbf{t}_k, \mathbf{D}_k, t) &= P_L(\mathbf{t}_k, \mathbf{D}_k, t) + E_L(\mathbf{t}_k, \mathbf{D}_k, t), \\ DMS_L(\mathbf{t}_k, \mathbf{D}_k, t) &= DP_L(\mathbf{t}_k, \mathbf{D}_k, t) + DE_L(\mathbf{t}_k, \mathbf{D}_k, t), \end{aligned} \tag{2}$$

where

$$\begin{aligned} P_L(\mathbf{t}_k, \mathbf{D}_k, t) &= \textstyle\sum_{i=0}^{k} D_i \varphi_i(t), \\ DP_L(\mathbf{t}_k, \mathbf{D}_k, t) &= \textstyle\sum_{i=0}^{k} D_i \varphi'_i(t), \\ E_L(\mathbf{t}_k, \mathbf{D}_k, t) &= \frac{1}{(k+1)!} F^{(k)}(T_{\mathbf{t}_k,t}, B_{\mathbf{t}_k,t}) w(t), \\ DE_L(\mathbf{t}_k, \mathbf{D}_k, t) &= \frac{1}{(k+1)!} F^{(k)}(T_{\mathbf{t}_k,t}, B_{\mathbf{t}_k,t}) w'(t), \\ w(t) &= \textstyle\prod_{i=0}^{k}(t - t_i). \end{aligned} \tag{3}$$

and where F is an interval extension of f and $B_{\mathbf{t}_k,t}$ is a bounding box of ms over $T_{\mathbf{t}_k,t}$ wrt $\mathbf{D}_k$ and $\mathbf{t}_k$.

We now show that, under certain restrictions on t, MS_L and DMS_L are interval extensions respectively of ms and $\frac{\partial ms}{\partial t}$ wrt the variables in $\mathbf{u}_k$.

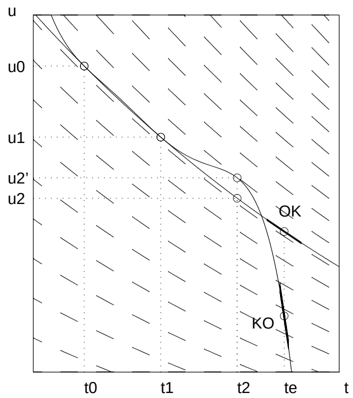

Figure 3: Geometric Intuition for the Hermite Polynomials

Proposition 2 (Correctness of Lagrange Interval Polynomials) *Let $\mathcal{O}$ be an ODE $u' = f(t,u)$ whose solutions are in $C^{k+1}(T_{\mathbf{t}_k,t}, \mathbb{R}^n)$, ms be the multistep solution of $\mathcal{O}$, and $\frac{\partial ms}{\partial t}$ be its derivative. Let MS_L and DMS_L be a Lagrange interval polynomial and its derivative for $\mathcal{O}$. Then, MS_L and DMS_L are interval extensions of ms and $\frac{\partial ms}{\partial t}$ wrt their second arguments forall $t \in (T_{\mathbf{t}_k,t} \setminus T_{\mathbf{t}_k}) \cup \{t_0, ..., t_k\}$.*

It is interesting at this point to make the connection between a natural multistep filtering operator based on Lagrange polynomials and Gear's method. Gear's method is a (traditional) implicit multistep method for solving ODEs that consists of solving (locally) a system of nonlinear equations based on Lagrange polynomial to find an (approximate) value at t_k given the (approximate) values at $t_0, \ldots, t_{k-1}$. The nonlinear equations in Gear's method specify implicitly the value of the solution at time t_k (i.e., there is no evaluation point t_e as in our case). The multistep filtering operator defined here uses Lagrange polynomials in a global way to prune the search space. As a consequence, at a very high level, the multistep filtering operator based on Lagrange polynomials is to Gear's method for ODEs what the interval Newton method is to Newton method for nonlinear equations.

5.2 An Hermite Polynomial Interval Extension

Lagrange polynomial interpolations are simple to compute but they only exploit a subset of the information available. For instance, they do not exploit the derivative information available at each evaluation point. This section presents an interpolation based on Hermite polynomials using this information. The intuition, depicted in Figure 3, is to constrain the polynomials to have acceptable slopes at the evaluation times. The figure, that uses the same differential equation as previously, shows that the interpolation polynomial must now have the correct slope at the various times. As can be seen, the slope at time t_e differs even much more from the solution than with Lagrange polynomials. As confirmed by our preliminary experimental results, the use of Hermite polynomials in the filtering operator should produce tighter enclosures of the multistep solution and its derivatives since the additional constraints tend to produce interpolation polynomials whose slopes are more similar (thus reducing the approximations due to interval computations).

Definition 13 (Hermite Polynomial [Atk88]) Let $u'_0, ..., u'_k \in \mathbb{R}^n$. Assume that $t_0, ..., t_k$ are distinct points. The *Hermite polynomial* that interpolates points $(t_0, u_0), ..., (t_k, u_k)$ and whose derivative interpolates points $(t_0, u'_0), ..., (t_k, u'_k)$ is the *unique* polynomial $p_H : \mathbb{R} \to \mathbb{R}^n$ of degree $\leq 2k+1$ satisfying

$$\begin{aligned} p_H(t_i) &= u_i, \quad 0 \leq i \leq k, \\ \frac{\partial p_H}{\partial t}(t_i) &= u'_i, \quad 0 \leq i \leq k. \end{aligned}$$

It is given by

$$p_H(t) = \sum_{i=0}^{k} u_i \varphi_i(t) + \sum_{i=0}^{k} u'_i \psi_i(t),$$

where

$$\begin{aligned} l_i(t) &= \tfrac{w(t)}{(t-t_i)w'(t_i)}, \\ w(t) &= \textstyle\prod_{i=0}^{k}(t-t_i), \\ \psi_i(t) &= (t-t_i)[l_i(t)]^2, \\ \varphi_i(t) &= [1-2l'_i(t_i)(t-t_i)][l_i(t)]^2, \quad 0 \le i \le k. \end{aligned}$$

It is possible to bound the errors made by a Hermite polynomial when interpolating a function.

Theorem 2 (Hermite Error Term) *Let $a, b \in \mathbb{R}$, g be a function in $C^{2k+2}([a,b],\mathbb{R}^n)$, p_H be the Hermite polynomial of degree $\le 2k+1$ that interpolates g at $k+1$ distinct points $t_0, ..., t_k$ in the interval $[a,b]$ and whose derivative interpolates g' at $t_0, ..., t_k$, and $t \in [a,b] \setminus T_{\mathbf{t}_k}$, Then,*

1. *$\exists \zeta_t \in]a,b[$: $g(t) - p_H(t) = \frac{1}{(2k+2)!} g^{(2k+2)}(\zeta_t) w^2(t)$;*

2. *$\exists \xi_t \in]a,b[$: $g'(t) - p'_H(t) = \frac{1}{(2k+2)!} g^{(2k+2)}(\xi_t)(w^2)'(t)$*

This result makes it possible to obtain interval extensions of *ms* and its derivative.

Definition 14 (Hermite Interval Polynomial) Let $\mathcal{O}$ be an ODE $u' = f(t,u)$ and *ms* be the multistep solution of $\mathcal{O}$. An *Hermite interval polynomial* and its derivative for $\mathcal{O}$ are respectively the functions MS_H and DMS_H defined by

$$\begin{aligned} MS_H(\mathbf{t}_k, \mathbf{D}_k, t) &= P_H(\mathbf{t}_k, \mathbf{D}_k, t) + E_H(\mathbf{t}_k, \mathbf{D}_k, t), \\ DMS_H(\mathbf{t}_k, \mathbf{D}_k, t) &= DP_H(\mathbf{t}_k, \mathbf{D}_k, t) + DE_H(\mathbf{t}_k, \mathbf{D}_k, t), \end{aligned} \tag{4}$$

where

$$\begin{aligned} P_H(\mathbf{t}_k, \mathbf{D}_k, t) &= \textstyle\sum_{i=0}^{k} D_i \varphi_i(t) + \sum_{i=0}^{k} F(t_i, D_i)\psi_i(t), \\ DP_H(\mathbf{t}_k, \mathbf{D}_k, t) &= \textstyle\sum_{i=0}^{k} D_i \varphi'_i(t) + \sum_{i=0}^{k} F(t_i, D_i)\psi'_i(t), \\ E_H(\mathbf{t}_k, \mathbf{D}_k, t) &= \tfrac{1}{(2k+2)!} F^{(2k+1)}(T_{\mathbf{t}_k,t}, B_{\mathbf{t}_k,t}) w^2(t), \\ DE_H(\mathbf{t}_k, \mathbf{D}_k, t) &= \tfrac{1}{(2k+2)!} F^{(2k+1)}(T_{\mathbf{t}_k,t}, B_{\mathbf{t}_k,t}) (w^2)'(t), \end{aligned} \tag{5}$$

and where F is an interval extension of f and $B_{\mathbf{t}_k,t}$ is a bounding box of *ms* over $T_{\mathbf{t}_k,t}$ wrt $\mathbf{D}_k$ and $\mathbf{t}_k$.

We are now in a position of proving that MS_H and DMS_H are interval extensions respectively of ms and $\frac{\partial ms}{\partial t}$ wrt the variables in $\mathbf{u}_k$, under the conditions of Theorem 2.

Proposition 3 (Correctness of Hermite Interval Polynomials) *Let $\mathcal{O}$ be an ODE $u' = f(t,u)$ whose solutions are in $C^{2k+2}(T_{\mathbf{t}_k,t}, \mathbb{R}^n)$, ms be the multistep solution of $\mathcal{O}$, and $\frac{\partial ms}{\partial t}$ be its derivative. Let MS_H and DMS_H be a Lagrange interval polynomial and its derivative for $\mathcal{O}$. Then, MS_H and DMS_H are interval extensions of ms and $\frac{\partial ms}{\partial t}$ wrt their second arguments forall $t \in (T_{\mathbf{t}_k,t} \setminus T_{\mathbf{t}_k})$.*

It is also important to discuss the choice of the evaluation point t_e in the filters using Hermite polynomials. On the one hand, because of the derivative constraints, choosing t_e too close from t_k produces too weak a constraint, since the filter is trivially satisfies at t_k. On the other hand, choosing it too far from t_k increases the sizes of the time intervals, of the bounding box, and, possibly, the polynomial evaluation itself. Hence, a reasonable choice of t_e should be a compromise between these two extremes. Of course, it is always possible to use several evaluation times.

6 Advanced Techniques

In this section, we consider some more advanced multistep filtering operators.

6.1 A Mean-Value Multistep Filtering Operator

In solving nonlinear equations, it is often useful to use several interval extensions (e.g., the natural extension and the Taylor extension) since they complement each other well. The natural extension is in general more appropriate far from a solution, while the Taylor extension is better suited when the search is closer to a solution [VHLD97]. This idea has also been used in solving systems of differential equations [Loh87]. In this section, we present a mean-value form (MVF) of the multistep pruning operator.

To understand the main intuition, recall that the multistep filtering operator is a constraint of the form

$$\frac{\partial ms}{\partial t} - f(t_e, ms) = 0$$

The idea is to replace the left-hand side of this equation by its mean-value form, while assuming that the multistep solution is of the form $ms = p + e$. Note that a direct application of the mean-value form would require to approximate a term of the form $\frac{\partial e}{\partial u}$, which is impossible since e is unknown. As a consequence, it is necessary to consider an interval extension E of function e that is independant from the variable u.

Definition 15 (Mean-Value Multistep Filtering Operator) Let $\mathcal{O}$ be an ODE $u' = f(t,u)$, let ms be the multistep solution of $\mathcal{O}$ expressed in the form

$$ms(\mathbf{t}_k, \mathbf{u}_k, t) = p(\mathbf{t}_k, \mathbf{u}_k, t) + e(\mathbf{t}_k, \mathbf{u}_k, t).$$

A *mean-value multistep filtering* operator for $\mathcal{O}$ is an interval equation

$$K - [A]R^T = 0 \tag{6}$$

where $K \in \mathcal{I}^n$, $[A] \in \mathcal{I}^{n \times n(k+1)}$ and $R \in \mathcal{I}^{n(k+1)}$ are defined as

$$\begin{aligned} K &= DP(\mathbf{t}_k, \overline{\mathbf{m}_k}, t_e) + DE(\mathbf{t}_k, \mathbf{D}_k, t_e) - F(\overline{t_e}, P(\mathbf{t}_k, \overline{\mathbf{m}_k}, t_e) + E(\mathbf{t}_k, \mathbf{D}_k, t_e)), \\ [A] &= DF(\overline{t_e}, P(\mathbf{t}_k, \mathbf{D}_k, t_e) + E(\mathbf{t}_k, \mathbf{D}_k, t_e)) \cdot\ DUP(\mathbf{t}_k, \mathbf{D}_k, t_e) \\ &\quad - DDP(\mathbf{t}_k, \mathbf{D}_k, t_e), \\ R^T &= \begin{pmatrix} D_0^T - \overline{m_0}^T \\ \vdots \\ D_k^T - \overline{m_k}^T \end{pmatrix}, \end{aligned} \tag{7}$$

where P, DP, E, DE, DF, DUP, and DDP are interval extensions of $p, \frac{\partial p}{\partial t}, e, \frac{\partial e}{\partial t}, \frac{\partial f}{\partial u}$ (Jacobian of f), $\frac{\partial p}{\partial \mathbf{u}_k}$ and $\frac{\partial}{\partial \mathbf{u}_k}\frac{\partial p}{\partial t}$, wrt the variables in $\mathbf{u}_k$ and where $m_i \in D_i$ $(0 \leq i \leq k)$ and $\mathbf{m}_k = \langle m_0, \ldots, m_k \rangle$.

The following lemma is the core of the correctness proof of the mean-value multistep filtering operator.

Lemma 1 *Let $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p : (x,y) \mapsto g(x,y)$ be differentiable wrt variable x. Let $D \in \mathcal{I}^n$, $m \in D$ and $S \subseteq \mathbb{R}^m$. Then, for all $x \in D$, we have:*

$$g(x,S) \subseteq g(m,S) + \tfrac{\partial g}{\partial x}(D,S) \cdot (x - m).$$

Proposition 4 (Soundness of the Mean-Value Multistep Filtering Operator) *A mean-value multistep pruning operator is a multistep pruning operator.*

6.2 The Wrapping Effect

The wrapping effect is a fundamental problem faced by interval methods. It comes from the fact that the solution at each evaluation time must be enclosed by a box. The approximations, that accumulate at each step, may significantly reduce the precision. This problem has been well-studied (see, for instance, [Loh87]) and a typical solution consists of choosing, at each evaluation time, a local coordinate system that fits the solution set as best as possible (which is equivalent to using parallelepipeds (a generalization of boxes) as enclosures of the solution sets). The local coordinate system for each t_i $0 \leq i \leq k$ amounts to defining a linear transformation

$$u_i = M_i w_i, \quad 0 \leq i \leq k$$

and there are standard techniques to choose appropriate matrices M_i (see, e.g., [Loh87]). With this idea in mind, it is possible to define a multistep filtering operator incorporating distinct coordinate systems.

Definition 16 (Preconditioned Multistep Filtering Operator) Let $\mathcal{O}$ be an ODE $u' = f(t, u)$, let *ms* be the multistep solution of $\mathcal{O}$, and let $M_0, ..., M_k \in \mathbb{R}^{n \times n}$. A *preconditioned multistep filtering* operator for $\mathcal{O}$ is an interval extension of the constraint

$$\frac{\partial ms}{\partial t}(\mathbf{t}_k, \langle M_0 w_o, \ldots, M_k w_k \rangle, t_e) = f(t_e, ms(\mathbf{t}_k, \langle M_0 w_o, \ldots, M_k w_k \rangle, t_e))$$

wrt the variables in $w_0, \ldots, w_k$.

Preconditioned multistep filtering operators are particularly adapted to the mean-value form because they enable us to control the wrapping by using associativity.

Definition 17 (Preconditioned Mean-Value Multistep Pruning Operator) Let $\mathcal{O}$ be an ODE $u' = f(t, u)$, let *ms* be the multistep solution of $\mathcal{O}$ of the form

$$ms(\mathbf{t}_k, \mathbf{u}_k, t) = p(\mathbf{t}_k, \mathbf{u}_k, t) + e(\mathbf{t}_k, \mathbf{u}_k, t).$$

and let $M_0, ..., M_k \in \mathbb{R}^{n \times n}$. A *preconditioned mean-value multistep filtering operator* for $\mathcal{O}$ is an interval equation of the form

$$K - ([A]M)R'^T = 0 \tag{8}$$

where $R' \in \mathcal{I}^{n(k+1)}$ and $M \in \mathbb{R}^{n(k+1) \times n(k+1)}$ are defined as

$$R'^T = \begin{pmatrix} D_0'^T - \overline{m_0'}^T \\ \vdots \\ D_k'^T - \overline{m_k'}^T \end{pmatrix}, \quad M = \begin{pmatrix} M_0 & 0 & \cdots & 0 \\ 0 & M_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & M_k \end{pmatrix}, \tag{9}$$

and where $K \in \mathcal{I}^n$ and $[A] \in \mathcal{I}^{n \times n(k+1)}$ are as in Definition 15 with

$$D_i = M_i D_i', \; m_i = M_i m_i' \; (0 \leq i \leq k), \; \mathbf{m}_k = \langle m_0, ..., m_k \rangle. \tag{10}$$

Note that the key in controlling the wrapping effect is the left-associaticity in $([A]M)R'^T$, since the right-associativity $[A](MR'^T)$ would reintroduce the wrapping effect. The correctness of the preconditioned mean-value multistep filtering operator is a direct consequence of the previous results.

t	Taylor	With Natural Pruning	Ratio
0.0	[0.99900 , 1.00000]	[0.99900 , 1.00000]	1.0
0.1	[0.36248 , 0.37574]	[0.36347 , 0.37379]	1.3
0.2	[0.12450 , 0.14811]	[0.13030 , 0.14051]	2.3
0.3	[0.01792 , 0.08270]	[0.04408 , 0.05581]	5.5
0.4	[-0.06954 , 0.10652]	[0.01196 , 0.02473]	13.8
0.5	[-0.23236 , 0.24592]	[-0.00021 , 0.01375]	34.2
0.6	[-0.64717 , 0.65214]	[-0.00512 , 0.01009]	85.4
0.7	[-1.76397 , 1.76579]	[-0.00736 , 0.00920]	213.1
0.8	[-4.79425 , 4.79491]	[-0.00868 , 0.00936]	531.5
0.9	[-13.02514 , 13.02539]	[-0.00970 , 0.00995]	1325.7
1.0	[-35.38525 , 35.38534]	[-0.01066 , 0.01075]	3305.4
1.1	[-96.13003 , 96.13006]	[-0.01164 , 0.01167]	8248.0
1.2	[-261.15328 , 261.15329]	[-0.01269 , 0.01270]	20571.3
1.3	[-709.46641 , 709.46641]	[-0.01383 , 0.01383]	51299.1
1.4	[-1927.38374 , 1927.38374]	[-0.01506 , 0.01507]	127937.8
1.5	[-5236.05915 , 5236.05915]	[-0.01641 , 0.01641]	319077.3

Table 1: ODE $u'(t) = -10u(t)$

7 Implementation Issues

Let us briefly discuss implementation issues to indicate that the approach is reasonable from a computational standpoint. First, recall that any interval method should compute bounding boxes and Taylor coefficients over these boxes to produce the initial enclosures (forward step). As a consequence, the error terms in the interpolation polynomials can be computed during this forward step, without introducing any significant overhead. Second, observe that the polynomials themselves are trivial to construct and evaluate. The construction takes place only once and is about $O(k^2)$, where k is the number of evaluation times considered. An evaluation of the natural multistep filtering operator based on these polynomials takes $O(kn)$, where n is the dimension of the ODE system, which is close to optimality. As a consequence, the main complexity will be associated with the filtering algorithm itself. Finally, the mean-value and preconditioned operator can be constructed efficiently since they only require information (e.g. the Jacobian) that is needed in the forward step based on these techniques. An evaluation is slightly more costly but remains perfectly reasonable. Note also that the cost of the filter is substantially less than the backwards pruning technique proposed in [DJVH98] which involves computing the Taylor coefficients for each evaluation.

8 Experimental results

This section reports some preliminary evidence that the filtering operator is an effective way to tighten the enclosures produced by the forward process. The results are only given for stiff problems although the filtering operator is also effective on simpler problems. The experimental results are obtained by applying box-consistency on the filtering operator based on Hermite polynomials of degree 5 (i.e. $k = 2$).

Table 1 presents the results on a simple problem. It shows the substantial gain produced by the pruning step over a traditional interval Taylor method of order 4, using a natural filtering operator. Note that even with higher order interval Taylor series, the gain remains substantial (e.g. with a Taylor series of order 8, the gain is bigger than 10^5 at time 1.5). Table 3 considers a quadratic ODE and compares a Taylor MVF (mean-value form) method (of order 4) as in Lohner's method [Loh87]

t	Piecewise Taylor	Taylor with Natural Pruning	Ratio
0.0	[0.00000 , 0.00000]	[0.00000 , 0.00000]	1.0
0.3	[-0.30291 , 0.89395]	[-0.07389 , 0.41865]	2.4
0.6	[-2.07810 , 3.20739]	[0.23293 , 0.87991]	8.2
0.9	[-8.65903 , 10.22568]	[0.15993 , 1.19334]	18.3
1.2	[-31.47707 , 33.34114]	[0.22960 , 1.62460]	46.5
1.5	[-109.32716 , 111.32215]	[0.09149 , 1.95039]	118.7
1.8	[-374.18327 , 376.13097]	[-0.06153 , 2.33463]	313.1
2.1	[-1274.89513 , 1276.62155]	[-0.48267 , 2.66253]	811.2
2.4	[-4337.28310 , 4338.63402]	[-1.10181 , 3.05072]	2089.3
2.7	[-14749.13410 , 14749.98886]	[-2.04553 , 3.44376]	5373.9
3.0	[-50148.94757 , 50149.22981]	[-3.27441 , 3.96133]	13861.5

Table 2: ODE $u'(t) = -10(u(t) - \sin(t)) + \cos(t)$

t	Taylor MVF	With Natural Pruning	With Mean-Value Pruning
0.0	[0.99900 , 1.00000]	[0.99900 , 1.00000]	[0.99900 , 1.00000]
0.5	[0.46864 , 0.70655]	[0.55967 , 0.70655]	[0.55996 , 0.70655]
1.0	[0.23241 , 0.56479]	[0.32344 , 0.52051]	[0.39500 , 0.52037]
1.5	[0.07896 , 0.52853]	[0.20427 , 0.42314]	[0.30245 , 0.39728]
2.0	[-0.08931 , 0.57922]	[0.13409 , 0.36708]	[0.24612 , 0.30683]
2.5	[-0.43481 , 0.83518]	[0.08495 , 0.32876]	[0.20736 , 0.25151]
3.0	[-3.05917 , 2.54748]	[0.04613 , 0.29898]	[0.17931 , 0.21180]
3.5	...	[0.01298 , 0.27534]	[0.15795 , 0.18317]
4.0	...	[-0.01759 , 0.25548]	[0.14116 , 0.16119]
4.5	...	[-0.04783 , 0.23822]	[0.12761 , 0.14393]
5.0	...	[-0.07463 , 0.22312]	[0.11644 , 0.12998]

Table 3: ODE $u'(t) = -1.5u^2(t)$

with a natural filtering operator and a mean-value filtering operator. The Taylor MVF method deteriorates quickly and explodes after time 3. The natural filtering operator does much better as can easily be seen. The mean-value filtering operator is even better and, in fact, converges towards the interval [0,0] when t grows. Table 2 considers another problem that leads to an explosion of the piecewise Taylor method (of order 4), i.e., the best forward method possible. The pruning step, using a natural filtering operator, substantially reduces the explosion in this case, although the step size is large (0.3). This clearly shows that the pruning step is orthogonal to the forward step (since it improves the best possible forward step) and is thus a promising research direction. Note also that a smaller step size (e.g., 0.2) would produce a tighter enclosure (e.g., [-0.14803,0.46879] at time 3) and smaller steps further improve the precision.

9 Conclusion

This paper was concerned with the solving of ODE systems using interval methods that provide guaranteed enclosures of the solutions and numerical proofs of existence and unicity of the solution. Unfortunately, they may result in large over-approximations of the solution because of the loss of precision in interval computations and the wrapping effect.

This paper took a constraint satisfaction approach to find tighter enclosures of the solution, while not sacrificing efficiency too much. The basic idea underlying this approach is to iterate a forward step to produce an initial enclosure with a pruning step that tightens it. The paper focused on the pruning step and proposed novel multistep filtering operators for ODEs. These operators overcome the fact that the constraints cannot be used directly by approximating the multistep solution through interval extensions based on (Lagrange and Hermite) interpolation polynomials and their error terms. The paper also showed how traditional techniques (such as mean-value forms and coordinate transformations) can be adapted to this new context. Finally, the paper also indicated that the filtering operators can be implemented effectively and preliminary experimental results were given to illustrate the potential of the approach.

The constraint satisfaction approach seems to open novel directions and future work will aim at validating its potential through a complete implementation and the investigation of new pruning operators, since it should be possible to derive a pruning operator from the mathematical properties underlying any implicit method.

Acknowledgment

This research is partially supported by the *Actions de recherche concertées (ARC/95/00-187)* of the Direction générale de la Recherche Scientifique – Communauté Franaise de Belgique.

References

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

[AH83] G. Alefeld and J. Herzberger. *Introduction to Interval Computations.* Academic Press, New York, NY, 1983.

[Atk88] K.E. Atkinson. *An Introduction to Numerical Analysis.* John Wiley & Sons, 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.

[DJVH98] Y. Deville, M. Janssen, and P. Van Hentenryck. Consistency Techniques in Ordinary Differential Equations. In *International Conference on Principles and Practice on Constraint Programming, LNCS 1520*, pages 162–176, 1998.

[KC96] D. Kincaid and W. Cheney. *Numerical Analysis.* Brooks/Cole, 1996.

[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.

[Neu90] A. Neumaier. *Interval Methods for Systems of Equations.* PHI Series in Computer Science. Cambridge University Press, Cambridge, 1990.

[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.

[VH98a] P. Van Hentenryck. A Constraint Satisfaction Approach to a Circuit Design Problem. *Journal of Global Optimization*, 13:75–93, 1998.

[VH98b] P. Van Hentenryck. A Gentle Introduction to Numerica. *Artificial Intelligence*, 103(1-2):209–235, 1998.

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

A Proofs of the Results

Theorem 1 (Lagrange Error Term) *Let $a, b \in \mathbb{R}$, g be a function in $C^{k+1}([a,b], \mathbb{R}^n)$, let p_L be the Lagrange polynomial of degree $\leq k$ that interpolates g at $k+1$ distinct points $t_0, ..., t_k$ in the interval $[a,b]$, and $t \in ([a,b] \setminus T_{\mathbf{t}_k}) \cup \{t_0, ..., t_k\}$. Then,*

1. $\exists \zeta_t \in]a,b[$: $g(t) - p_L(t) = \frac{1}{(k+1)!} g^{(k+1)}(\zeta_t) w(t)$;
2. $\exists \xi_t \in]a,b[$: $g'(t) - p'_L(t) = \frac{1}{(k+1)!} g^{(k+1)}(\xi_t) w'(t)$.

where $w(t) = \prod_{i=0}^{k}(t - t_i)$.

Proof
1. See [KC96].
2. Let t be a point in $([a,b] \setminus T_{\mathbf{t}_k}) \cup \{t_0, ..., t_k\}$. Put $\phi(x) = g(x) - p_L(x) - \lambda w(x)$, where λ is given by $\lambda = \frac{g'(t) - p'_L(t)}{w'(t)}$. The function ϕ is in $C^{k+1}([a,b], \mathbb{R}^n)$ and it takes the value 0 at the $k+1$ points $t_0, ..., t_k$. By Rolle's Theorem, ϕ' has at least k distinct zeros in $T_{\mathbf{t}_k}$ that are not in $\{t_0, ..., t_k\}$ and that are distinct from t. Moreover, $\phi'(t) = 0$. Thus, ϕ' has at least $k+1$ distinct zeros in $[a,b]$. By Rolle's Theorem, ϕ'' has at least k distinct zeros in $]a,b[$. Similarly, ϕ''' has at least $k-1$ distinct zeros in $]a,b[$, etc. Finally, we see that $\phi^{(k+1)}$ has at least one zero, say ξ_t, in $]a,b[$. We have:

$$\phi^{(k+1)} = g^{(k+1)} - p_L^{(k+1)} - \lambda w^{(k+1)} = g^{(k+1)} - (k+1)!\lambda.$$

Hence,

$$0 = \phi^{(k+1)}(\xi_t) = g^{(k+1)}(\xi_t) - (k+1)!\lambda = g^{(k+1)}(\xi_t) - (k+1)! \frac{g'(t) - p'_L(t)}{w'(t)}.$$

□

Proposition 1 (Correctness of Lagrange Interval Polynomials) *Let $\mathcal{O}$ be an ODE $u' = f(t,u)$ whose solutions are in $C^{k+1}(T_{\mathbf{t}_k,t}, \mathbb{R}^n)$, ms be the multistep solution of $\mathcal{O}$, and $\frac{\partial ms}{\partial t}$ be its derivative. Let MS_L and DMS_L be a Lagrange interval polynomial and its derivative for $\mathcal{O}$. Then, MS_L and DMS_L are interval extensions of ms and $\frac{\partial ms}{\partial t}$ wrt their second arguments forall $t \in (T_{\mathbf{t}_k,t} \setminus T_{\mathbf{t}_k}) \cup \{t_0, ..., t_k\}$.*

Proof Let $t \in (T_{\mathbf{t}_k,t} \setminus T_{\mathbf{t}_k}) \cup \{t_0, ..., t_k\}$. Let $p_L(\mathbf{t}_k, \mathbf{u}_k, \bullet)$ be the Lagrange polynomial interpolating points $(t_0, u_0), ..., (t_k, u_k)$. Let $e_L(\mathbf{t}_k, \mathbf{u}_k, t) = ms(\mathbf{t}_k, \mathbf{u}_k, t) - p_L(\mathbf{t}_k, \mathbf{u}_k, t)$. Let E be the domain of definition of function $ms(\mathbf{t}_k, \bullet, t)$. If $\mathbf{u}_k \in E$ and $u(t) = ms(\mathbf{t}_k, \mathbf{u}_k, t)$, then, by Theorem 1, there exists $\zeta_t, \xi_t \in T_{\mathbf{t}_k,t}$ such that

$$\begin{aligned} e_L(\mathbf{t}_k, \mathbf{u}_k, t) &= \frac{1}{(k+1)!} u^{(k+1)}(\zeta_t) w(t) = \frac{1}{(k+1)!} f^{(k)}(\zeta_t, u(\zeta_t)) w(t); \\ \frac{\partial e_L}{\partial t}(\mathbf{t}_k, \mathbf{u}_k, t) &= \frac{1}{(k+1)!} u^{(k+1)}(\xi_t) w'(t) = \frac{1}{(k+1)!} f^{(k)}(\xi_t, u(\xi_t)) w'(t). \end{aligned}$$

Clearly, we have:

$$\begin{aligned} p_L(\mathbf{t}_k, E \cap \mathbf{D}_k, t) &\subseteq P_L(\mathbf{t}_k, \mathbf{D}_k, t), \\ \frac{\partial p_L}{\partial t}(\mathbf{t}_k, E \cap \mathbf{D}_k, t) &\subseteq DP_L(\mathbf{t}_k, \mathbf{D}_k, t), \\ e_L(\mathbf{t}_k, E \cap \mathbf{D}_k, t) &\subseteq E_L(\mathbf{t}_k, \mathbf{D}_k, t), \\ \frac{\partial e_L}{\partial t}(\mathbf{t}_k, E \cap \mathbf{D}_k, t) &\subseteq DE_L(\mathbf{t}_k, \mathbf{D}_k, t), \end{aligned}$$

and thus

$$\begin{aligned} ms(\mathbf{t}_k, E \cap \mathbf{D}_k, t) &\subseteq MS_L(\mathbf{t}_k, \mathbf{D}_k, t), \\ \frac{\partial ms}{\partial t}(\mathbf{t}_k, E \cap \mathbf{D}_k, t) &\subseteq DMS_L(\mathbf{t}_k, \mathbf{D}_k, t), \end{aligned}$$

which means that MS_L and DMS_L are interval extensions respectively of ms and $\frac{\partial ms}{\partial t}$ wrt the variables in $\mathbf{u}_k$. □

Theorem 2 (Hermite Error Term) *Let $a, b \in \mathbb{R}$, g be a function in $C^{2k+2}([a,b], \mathbb{R}^n)$, p_H be the Hermite polynomial of degree $\leq 2k+1$ that interpolates g at $k+1$ distinct points $t_0, ..., t_k$ in the interval $[a,b]$ and whose derivative interpolates g' at $t_0, ..., t_k$, and $t \in [a,b] \setminus T_{\mathbf{t}_k}$, Then,*

1. $\exists \zeta_t \in]a,b[: g(t) - p_H(t) = \frac{1}{(2k+2)!} g^{(2k+2)}(\zeta_t) w^2(t)$;
2. $\exists \xi_t \in]a,b[: g'(t) - p'_H(t) = \frac{1}{(2k+2)!} g^{(2k+2)}(\xi_t)(w^2)'(t)$

Proof
1. See [Atk88].
2. Let t be a point in $[a,b] \setminus T_{\mathbf{t}_k}$. Put $\phi(x) = g(x) - p_H(x) - \lambda w^2(x)$, where λ is given by $\lambda = \frac{g'(t) - p'_H(t)}{(w^2)'(t)}$. Function $\phi \in C^{2k+2}([a,b], \mathbb{R}^n)$, and ϕ takes the value 0 at the $k+1$ points $t_0, ..., t_k$. By Rolle's Theorem, ϕ' has at least k distinct zeros in $T_{\mathbf{t}_k}$, and distinct from $t_0, ..., t_k$. Moreover, $\phi'(t) = 0$ and $t \notin T_{\mathbf{t}_k}$. We also have $\phi'(t_i) = 0$ for $0 \leq i \leq k$. Thus, ϕ' has at least $2k+2$ distinct zeros in $[a,b]$. By Rolle's Theorem, ϕ'' has at least $2k+1$ distinct zeros in $]a,b[$. Similarly, ϕ''' has at least $2k$ distinct zeros in $]a,b[$, etc. Finally, we see that $\phi^{(2k+2)}$ has at least one zero, say ξ_t, in $]a,b[$. We have:

$$\phi^{(2k+2)} = g^{(2k+2)} - p_H^{(2k+2)} - \lambda (w^2)^{(2k+2)} = g^{(2k+2)} - (2k+2)!\lambda.$$

Hence,

$$0 = \phi^{(2k+2)}(\xi_t) = g^{(2k+2)}(\xi_t) - (2k+2)!\lambda = g^{(2k+2)}(\xi_t) - (2k+2)! \frac{g'(t) - p'_H(t)}{(w^2)'(t)}.$$

□

Proposition 2 (Correctness of Hermite Interval Polynomials) *Let $\mathcal{O}$ be an ODE $u' = f(t,u)$ whose solutions are in $C^{2k+2}(T_{\mathbf{t}_k,t}, \mathbb{R}^n)$, ms be the multistep solution of $\mathcal{O}$, and $\frac{\partial ms}{\partial t}$ be its derivative. Let MS_H and DMS_H be a Lagrange interval polynomial and its derivative for $\mathcal{O}$. Then, MS_H and DMS_H are interval extensions of ms and $\frac{\partial ms}{\partial t}$ wrt their second arguments forall $t \in (T_{\mathbf{t}_k,t} \setminus T_{\mathbf{t}_k})$.*

Proof Let $t \in T_{\mathbf{t}_k,t} \setminus T_{\mathbf{t}_k}$. Let $p_H(\mathbf{t}_k, \mathbf{u}_k, \bullet)$ be the Hermite polynomial that interpolates points $(t_0, u_0), ..., (t_k, u_k)$ and whose derivative interpolates points $(t_0, f(t_0, u_0)), ..., (t_k, f(t_k, u_k))$. Let $e_H(\mathbf{t}_k, \mathbf{u}_k, t) = ms(\mathbf{t}_k, \mathbf{u}_k, t) - p_H(\mathbf{t}_k, \mathbf{u}_k, t)$. Let E be the domain of definition of function $ms(\mathbf{t}_k, \bullet, t)$. If $\mathbf{u}_k \in E$ and $u(t) = ms(\mathbf{t}_k, \mathbf{u}_k, t)$, then, by Theorem 2, there exists $\zeta_t, \xi_t \in T_{\mathbf{t}_k,t}$ such that

$$\begin{aligned} e_H(\mathbf{t}_k, \mathbf{u}_k, t) &= \frac{1}{(2k+2)!} u^{(2k+2)}(\zeta_t) w^2(t) &&= \frac{1}{(2k+2)!} f^{(2k+1)}(\zeta_t, u(\zeta_t)) w^2(t); \\ \frac{\partial e_H}{\partial t}(\mathbf{t}_k, \mathbf{u}_k, t) &= \frac{1}{(2k+2)!} u^{(2k+2)}(\xi_t)(w^2)'(t) &&= \frac{1}{(2k+2)!} f^{(2k+1)}(\xi_t, u(\xi_t))(w^2)'(t). \end{aligned}$$

Clearly, we have:

$$\begin{aligned} p_H(\mathbf{t}_k, E \cap \mathbf{D}_k, t) &\subseteq P_H(\mathbf{t}_k, \mathbf{D}_k, t), \\ \frac{\partial p_H}{\partial t}(\mathbf{t}_k, E \cap \mathbf{D}_k, t) &\subseteq DP_H(\mathbf{t}_k, \mathbf{D}_k, t), \\ e_H(\mathbf{t}_k, E \cap \mathbf{D}_k, t) &\subseteq E_H(\mathbf{t}_k, \mathbf{D}_k, t), \\ \frac{\partial e_H}{\partial t}(\mathbf{t}_k, E \cap \mathbf{D}_k, t) &\subseteq DE_H(\mathbf{t}_k, \mathbf{D}_k, t), \end{aligned}$$

and thus

$$\begin{aligned} ms(\mathbf{t}_k, E \cap \mathbf{D}_k, t) &\subseteq MS_H(\mathbf{t}_k, \mathbf{D}_k, t), \\ \frac{\partial ms}{\partial t}(\mathbf{t}_k, E \cap \mathbf{D}_k, t) &\subseteq DMS_H(\mathbf{t}_k, \mathbf{D}_k, t), \end{aligned}$$

which means that MS_H and DMS_H are interval extensions respectively of ms and $\frac{\partial ms}{\partial t}$ wrt the variables in $\mathbf{u}_k$. □

Lemma 1 *Let $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p : (x,y) \mapsto g(x,y)$ be differentiable wrt variable x. Let $D \in \mathcal{I}^n$, $m \in D$ and $S \subseteq \mathbb{R}^m$. Then, for all $x \in D$, we have:*

$$g(x,S) \subseteq g(m,S) + \tfrac{\partial g}{\partial x}(D,S) \cdot (x-m).$$

Proof Let $1 \leq i \leq p$. Let $x \in D$ and $e \in S$. By the mean-value theorem, we can write:

$$g_i(x,e) = g_i(m,e) + \tfrac{\partial g_i}{\partial x}(\xi_i, e) \cdot (x-m),$$

for some ξ_i on the straight line between x and m. As D is a convex set, the straight line between x and m must lie in D and we have

$$g_i(x,e) \in g_i(m,e) + \tfrac{\partial g_i}{\partial x}(D, e) \cdot (x-m),$$

for each $1 \leq i \leq p$. Thus,

$$g(x,e) \in g(m,e) + \tfrac{\partial g}{\partial x}(D, e) \cdot (x-m).$$

Furthermore,

$$\begin{aligned} g(m,S) + \tfrac{\partial g}{\partial x}(D,S) \cdot (x-m) &\supseteq \textstyle\bigcup_{e \in S}(g(m,e) + \tfrac{\partial g}{\partial x}(D,e) \cdot (x-m)) \\ &\supseteq \textstyle\bigcup_{e \in S}\{g(x,e)\} = \{g(x,e) \mid e \in S\} = g(x,S). \end{aligned}$$

□

Proposition 3 (Soundness of the Mean-Value Multistep Filtering Operator) *A mean-value multistep pruning operator is a multistep pruning operator.*

Proof Assuming the notations of Definition 15, let $\mathbf{u}_k \in \mathbf{D}_k$. Then the multistep filtering operator can be written as ,

$$\begin{array}{c} \frac{\partial p}{\partial t}(\mathbf{t}_k, \mathbf{u}_k, t_e) + \frac{\partial e}{\partial t}(\mathbf{t}_k, \mathbf{u}_k, t_e) - f(t_e, p(\mathbf{t}_k, \mathbf{u}_k, t_e) + e(\mathbf{t}_k, \mathbf{u}_k, t_e)) = 0 \\ \Downarrow \\ \frac{\partial p}{\partial t}(\mathbf{t}_k, \mathbf{u}_k, t_e) + DE(\mathbf{t}_k, \mathbf{D}_k, t_e) - f(t_e, p(\mathbf{t}_k, \mathbf{u}_k, t_e) + E(\mathbf{t}_k, \mathbf{D}_k, t_e)) = 0 \\ \Downarrow \text{(by Lemma 1)} \\ \mathcal{K} - \mathcal{A}\mathcal{R}^T \end{array}$$

$$\begin{array}{ll} \text{with} & \mathcal{K} = \frac{\partial p}{\partial t}(\mathbf{t}_k, \mathbf{m}_k, t_e) + DE(\mathbf{t}_k, \mathbf{D}_k, t_e) - f(t_e, p(\mathbf{t}_k, \mathbf{m}_k, t_e) + E(\mathbf{t}_k, \mathbf{D}_k, t_e)) \\ & \mathcal{A} = \frac{\partial f}{\partial u}(t_e, p(\mathbf{t}_k, \mathbf{D}_k, t_e) + E(\mathbf{t}_k, \mathbf{D}_k, t_e)) \cdot \frac{\partial p}{\partial \mathbf{u}_k}(\mathbf{t}_k, \mathbf{D}_k, t_e) - \frac{\partial}{\partial \mathbf{u}_k}\frac{\partial p}{\partial t}(\mathbf{t}_k, \mathbf{D}_k, t_e) \\ & \mathcal{R}^T = \begin{pmatrix} u_0^T - m_0^T \\ \vdots \\ u_k^T - m_k^T \end{pmatrix} \end{array}$$

It follows that the mean-value multistep filtering operator is a multistep filtering operator. □