Optimal Pruning in Parametric Differential Equations

Micha Janssen
UCL
10, Place Sainte Barbe
1348 Louvain-La Neuve

Pascal Van Hentenryck
Brown University
Box 1910
Providence, RI 02912

Yves Deville
UCL
10, Place Sainte Barbe
1348 Louvain-La Neuve

Abstract

Initial value problems for parametric ordinary differential equations (ODEs) arise in many areas of science and engineering. Since some of the data is uncertain, traditional numerical methods do not apply. This paper considers a constraint satisfaction approach that enhances traditional interval methods with a pruning component which uses a relaxation of the ODE and Hermite interpolation polynomials. It solves the main theoretical and practical open issue left in this approach: the choice of an optimal evaluation time for the relaxation. As a consequence, the constraint satisfaction approach is shown to provide a quadratic (asymptotical) improvement in accuracy over the best interval methods, while improving their running times. Experimental results on standard benchmarks confirm the theoretical results.

1 Introduction

Initial value problems (IVPs) for ordinary differential equations (ODEs) arise naturally in many applications in science and engineering, including chemistry, physics, molecular biology, and mechanics to name only a few. An *ordinary differential equation* $\mathcal{O}$ is a system of the form

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

often denoted in vector notation by $u'(t) = f(t, u(t))$ or $u' = f(t, u)$. An *initial value problem* is an ODE with an initial condition $u(t_0) = u_0$. In addition, in practice, it is often the case that the parameters and/or the initial values are not known with certainty but are given as intervals. Hence traditional methods do not apply to the resulting parametric ordinary differential equations since they would have to solve infinitely many systems. Interval methods, pioneered by Moore [Moo66], provide an approach to tackle parametric ODEs. These methods return enclosures of the exact solution at different points in time, i.e., they are guaranteed to return intervals containing the exact solution. In addition, they accommodate easily uncertainty in the parameters or initial values by using intervals instead of floating-point numbers. Interval methods typically apply a one-step Taylor interval method and make extensive use of automatic differentiation to obtain the Taylor coefficients [Eij81, Kru69, Moo66, Moo79]. Their major problem however is the explosion of the size of the boxes at successive points as they often accumulate errors from point to point and lose accuracy by enclosing the solution by a box (this is called the *wrapping effect*). Lohner's AWA system [Loh87] was an important step in interval methods which features efficient coordinate transformations to tackle the wrapping effect. More recently, Nedialkov and Jackson's IHO method [NJ99] improved on AWA by extending a Hermite-Obreschkoff's approach (which can be viewed as a generalized Taylor method) to intervals (see also

[Ber98]). Note that interval methods inherently accommodate uncertain data. Hence, in this paper, we talk about ODEs to denote both traditional and parametric ODEs.

This research takes a constraint satisfaction approach to ODEs. Its basic idea [DJVH98, JDVH99, JVHD01] is to view the solving of ODEs as the iteration of three processes: (1) a *bounding box* process that computes bounding boxes for the current step and proves (numerically) the existence and uniqueness of the solution, (2) a *predictor* process that computes initial enclosures at given times from enclosures at previous times and bounding boxes and (3) a *pruning* process that reduces the initial enclosures without removing solutions. The real novelty in our approach is the pruning component. Pruning in ODEs however generates significant challenges since ODEs contain unknown functions. The main contribution of our research is to show that an effective pruning technique can be derived from a relaxation of the ODE, importing a fundamental principle from constraint satisfaction into the field of differential equations. Four main steps are necessary to derive an effective pruning algorithm. The first step consists in obtaining a relaxation of the ODE by safely approximating its solution using, e.g., Hermite interpolation polynomials. The second step consists in using the mean-value form of this relaxation to solve the relaxation accurately and efficiently. Unfortunately, these two steps, which were skeched in [JDVH99], are not sufficient and the resulting pruning algorithm still suffers from traditional problems of interval methods. The third fundamental step, which was presented in [JVHD01], consists in globalizing the pruning by considering several successive relaxations together. This idea of generating a global constraint from a set of more primitive constraints is also at the heart of constraint satisfaction. It makes it possible, in this new context, to address the problem of dependencies and the wrapping effect simultaneously.[1] *The fourth step, which is the main contribution of this paper, consists of choosing an evaluation time for the relaxation that maximizes pruning.* Indeed, the global constraint generated in the third step, being a relaxation of the ODE itself, is parametrized by an evaluation time. In [JVHD01], the evaluation time was chosen heuristically and its choice was left as the main open issue in the constraint satisfaction approach to parametric ODEs.

The main contribution of this paper is to close this last open problem and to show that, for global filters based on Hermite interpolation polynomials, the optimal evaluation time is independent from the ODE itself and can be precomputed before starting the integration steps at negligible cost. This result has fundamental theoretical and practical consequences. From a theoretical standpoint, it can be shown that the constraint satisfaction approach provides a quadratic improvement in accuracy (asymptotically) over the best interval methods we know of while decreasing their computation costs as well. This result also implies that our approach should be significantly faster when the function f is very complex. Experimental results confirm the theory. They show that the constraint satisfaction approach often produces many orders of magnitude improvements in accuracy over existing methods while decreasing computation times. Alternatively, at similar accuracy, other approaches are significantly slower. The rest of the paper is organized as follows. Section 2 introduces the main definitions and notations. Section 3 gives a high-level overview of the constraint satisfaction approach to parametric ODEs. Section 4 is the core of the paper. It describes how to choose an evaluation time to maximize pruning. Sections 5 and 6 report the theoretical and experimental analyses and Section 7 concludes the paper. The appendix contains the proofs of the main results. A comprehensive presentation of all results, including detailed algorithms, is available in the technical report version of this paper.

2 Background and Definitions

Small letters denote real values, vectors and functions of real values. Capital letters denote matrices, sets, intervals, vectors and functions of intervals. $\mathbb{IR}$ denotes the set of all *closed* intervals $\subseteq \mathbb{R}$. A vector of intervals $D \in \mathbb{IR}^n$ is called a *box*. If $r \in \mathbb{R}$, then $\overline{r}$ denotes the smallest interval $I \in \mathbb{IR}$ such that $r \in I$.

[1] Global constraints in ordinary differential equations have also been found useful in [CB99]. The problem and the techniques in [CB99] are however fundamentally different.

If $r \in \mathbb{R}^n$, then $\overline{r} = (\overline{r_1}, \ldots, \overline{r_n})$. We often use r instead of $\overline{r}$ for simplicity. If $A \subseteq \mathbb{R}^n$, then $\Box A$ denotes the smallest box $D \in \mathbb{IR}^n$ such that $A \subseteq D$ and $g(A)$ denotes the set $\{g(x) \mid x \in A\}$. We also assume that a, b, t_i, t_e and t are reals, $I_i \in \mathbb{IR}$, u_i is in $\mathbb{R}^n$, and D_i and B_i are in $\mathbb{IR}^n$ $(i \in \mathbb{N})$. We use $m(D)$ to denote the midpoint of D and $s(D)$ to denote $D - m(D)$. Observe that $m(D) + s(D) = D$. We use $D_x g$ to denote the Jacobian of g wrt x and $\omega(D)$ to denote the width of a box. More precisely, $\omega([a, b]) = b - a$ and $\omega((I_1, \ldots, I_n)) = (\omega(I_1), \ldots, \omega(I_n))$.

Notation 1 *Let A be a set and $a_i \in A$ where $i \in \mathbb{N}$. We use the following bold face notations.*

$$\begin{array}{lcl} \mathbf{a} & = & (a_0, \ldots, a_k) \in A^{k+1} \\ \mathbf{a}_i & = & (a_{ik}, \ldots, a_{(i+1)k-1}) \in A^k \\ \mathbf{a}_{i..i+j} & = & (a_i, \ldots, a_{i+j}) \in A^{j+1} \end{array} \quad (1)$$

Observe that $\mathbf{a}_0 = (a_0, \ldots, a_{k-1})$, $\mathbf{a}_1 = (a_k, \ldots, a_{2k-1})$, and $\mathbf{a} = (a_0, \ldots, a_k)$. In the theoretical parts, we assume that the underlying interval arithmetic is exact. As traditional, we restrict attention to ODEs that have a unique solution for a given initial value and where $f \in C^\infty$. Techniques to verify this hypothesis numerically are well-known [Moo79, DJVH98]. Moreover, in practice, the objective is to produce (an approximation of) the values of the solution of $\mathcal{O}$ at different points $t_0, t_1, \ldots, t_m$. This motivates the following definition of solutions and its generalization to multistep solutions.

Definition 1 (Solution of an ODE) *The* solution *of an ODE $\mathcal{O}$ on $I \in \mathbb{IR}$ is the function $s : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ such that $\forall t \in I : \frac{\partial s}{\partial t}(t_0, u_0, t) = f(t, s(t_0, u_0, t))$ for an initial condition $s(t_0, u_0, t_0) = u_0$.*

Definition 2 (Multistep solution of an ODE) *The* multistep solution *of an ODE $\mathcal{O}$ 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 as $ms(\mathbf{t}, \mathbf{u}, t) = s(t_0, u_0, t)$* if *$u_i = s(t_0, u_0, t_i)$ $(1 \leq i \leq k)$ where s is the solution of $\mathcal{O}$ and is undefined otherwise.*

Since multistep solutions are partial functions, we generalize interval extensions to partial functions.

Definition 3 (Interval Extension of a Partial Function) *The interval function $G : \mathbb{IR}^n \to \mathbb{IR}^m$ is an* interval extension *of the partial function $g : E \subseteq \mathbb{R}^n \to \mathbb{R}^m$ if $\forall D \in \mathbb{IR}^n : g(E \cap D) \subseteq G(D)$.*

Finally, we generalize the concept of bounding boxes to multistep methods. Intuitively, a bounding box encloses all solutions of an ODE going through certain boxes at given times over a given time interval. Bounding boxes are often used to approximate error terms in ODEs.

Definition 4 (Bounding Box) *Let $\mathcal{O}$ be an ODE system, ms be the multistep solution of $\mathcal{O}$, and $\{t_0, \ldots, t_k\} \subseteq T \in \mathbb{IR}$. A box B is a* bounding box *of $\mathcal{O}$ over T wrt ($\mathbf{t}$,$\mathbf{D}$) if, for all $t \in T$, $ms(\mathbf{t}, \mathbf{D}, t) \subseteq B$.*

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 three processes: (1) a *bounding box* process that computes bounding boxes for the current step and proves (numerically) the existence and uniqueness of the solution, (2) a *predictor* process that computes initial enclosures at given times from enclosures at previous times and bounding boxes and (3) a *pruning* process that reduces the initial enclosures without removing solutions. The bounding box and predictor components are standard in interval methods for ODEs. This paper thus focuses on the pruning process, the main novelty of the approach. *Our pruning component is based on relaxations of the ODE.* To our knowledge, no other approach uses relaxations of the ODE to derive pruning operators and the only other approaches using a pruning component [NJ99, Rih98] were developed independently. Note also that, in the following, predicted boxes are generally superscripted with the symbol $-$ (e.g., D_1^-), while pruned boxes are generally superscripted with the symbol $*$ (e.g., D_1^*).

The pruning component uses *safe approximations* of the ODE to shrink the boxes computed by the predictor process. To understand this idea, it is useful to contrast the constraint satisfaction approach to nonlinear programming [VHMD97] and to ordinary differential 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 Cartesian product 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 $I'_1 \times \ldots \times I'_n$. The interval extension can be seen as a filter that can be used for pruning the search space in many ways. For instance, Numerica uses box(k)-consistency on these interval constraints [VHMD97]. Ordinary differential equations raise new challenges. In an ODE $\forall\, t:\ u' = f(t,u)$, functions *u* and *u'* are, of course, unknown. Hence it is not obvious how to obtain a filter to prune boxes.

One of the main contributions of our approach is to show how to derive effective pruning operators for parametric ODEs. The first step consists in rewriting the ODE in terms of its multistep solution *ms* to obtain

$$\forall\, t:\ \tfrac{\partial ms}{\partial t}(\mathbf{t}, \mathbf{u}, t) = f(t, ms(\mathbf{t}, \mathbf{u}, t)).$$

Let us denote this formula $\forall\, t: fl(\mathbf{t}, \mathbf{u}, t)$. This rewriting may not appear useful since *ms* is still an unknown function. However it suggests a way to approximate the ODE. Indeed, we show in Section 3.3 how to obtain interval extensions of *ms* and $\frac{\partial ms}{\partial t}$ by using polynomial interpolations together with their error terms. This simply requires a bounding box for the considered time interval and safe approximations of *ms* at successive times, both of which are available from the bounding box and predictor processes. Once these interval extensions are available, it is possible to obtain an interval formula of the form

$$\forall\, t: FL(\mathbf{t}, \mathbf{D}, t)$$

which approximates the original ODE. The above formula is still not ready to be used as a filter because t is universally quantified. The solution here is simpler and consists of restricting attention to a finite set T of times (possibly a singleton) to obtain the relation

$$\forall\, t\ \in\ T: FL(\mathbf{t}, \mathbf{D}, t)$$

which produces a computable filter. Indeed, if the relation $FL(\mathbf{t}, \mathbf{D}, t)$ does not hold for a time t, it follows that no solution of $u' = f(t,u)$ can go through boxes $D_0, \ldots, D_k$ at times $t_0, \ldots, t_k$. The following definition and proposition capture these concepts more formally.

Definition 5 (Multistep Filter) *Let $\mathcal{O}$ be an ODE and s its solution. A* multistep filter *for $\mathcal{O}$ is an interval relation $FL: \mathbb{R}^{k+1} \times (\mathbb{IR}^n)^{k+1} \times \mathbb{R} \rightarrow Bool$ satisfying*

$$\left.\begin{array}{c} u_i\ \in\ D_i \\ s(t_0, u_0, t_i) = u_i\ (0 \leq i \leq k) \end{array}\right\} \Rightarrow\ \ \forall t:\ FL(\mathbf{t}, \mathbf{D}, t).$$

How can we use this filter to obtain tighter enclosures of the solution? A simple technique consists of pruning the last box computed by the predictor process. Assume that D_i^* is a box enclosing the solution at time t_i ($0 \leq i < k$) and that we are interested in pruning the last predicted box D_k^-. A subbox $D \subseteq D_k^-$ can be pruned away if the condition $FL(\mathbf{t}, (D_0^*, \ldots, D_{k-1}^*, D), t_e)$ does not hold for some evaluation point t_e. Let us explain briefly the geometric intuition behind this formula by considering what we call *natural filters*. Given interval extensions *MS* and *DMS* of *ms* and $\frac{\partial ms}{\partial t}$, it is possible to approximate the ODE $u' = f(t,u)$ by the formula

$$DMS(\mathbf{t}, \mathbf{D}, t) = F(t, MS(\mathbf{t}, \mathbf{D}, t)).$$

In this formula, the left-hand side of the equation represents *the approximation of the slope of* u while the right-hand represents *the slope of the approximation of* u. Since the approximations are conservative, these two sides must intersect on boxes containing a solution. Hence an empty intersection means that the boxes used in the formula do not contain the solution to the ODE system. Traditional consistency techniques and algorithms based on this filter can now be applied. For instance, one may be interested in updating the last box computed by the predictor process using the operator $D_k^* = \Box\{r \in D_k^- \mid FL(\mathbf{t}, (D_0^*, \dots, D_{k-1}^*, r), t_e)\}$. *Observe that this operator uses an evaluation time* t_e *and the main result of this paper consists in showing that* t_e *can be chosen optimally to maximize pruning.* The following definition is a novel notion of consistency for ODEs to capture pruning of the last r boxes.

Definition 6 (Backward Consistency of Multistep Filters) *A multistep filter* $FL(\mathbf{t}, \mathbf{D}, e)$ *is backward-consistent in* $(\mathbf{t}, \mathbf{D})$ *for time* e *if* $\mathbf{D} = \Box\{u_k \in D_k \mid \exists \mathbf{u}_0 \in \mathbf{D}_0 : FL(\mathbf{t}, \mathbf{u}, e)\}$ *. A system of* r *successive multistep filters* $\{FL_i(\mathbf{t}_{i..k+i}, \mathbf{D}_{i..k+i}, e_i)\}_{0 \leq i < r}$ is backward(r)-consistent *in* $(\mathbf{t}_{0..k+r-1}, \mathbf{D}_{0..k+r-1})$ *for times* e_i $(0 \leq i < r)$ *if*

$$\mathbf{D}_{k..k+r-1} = \Box\{\mathbf{u}_{k..k+r-1} \in \mathbf{D}_{k..k+r-1} \mid \exists \mathbf{u}_0 \in \mathbf{D}_0 : \forall\, 0 \leq i < r : FL_i(\mathbf{t}_{i..k+i}, \mathbf{u}_{i..k+i}, e_i)\}.$$

3.1 Multistep Filters

Filters rely on interval extensions of the multistep solution and of its derivative wrt t. These extensions are, in general, based on decomposing the (unknown) multistep solution into the sum of a computable approximation p and an (unknown) error term e, i.e., $ms(\mathbf{t}, \mathbf{u}, t) = p(\mathbf{t}, \mathbf{u}, t) + e(\mathbf{t}, \mathbf{u}, t)$. There exist standard techniques to build p and $\frac{\partial p}{\partial t}$ and to bound e and $\frac{\partial e}{\partial t}$. Section 3.3 reviews how they can be derived from Hermite interpolation polynomials. Here we simply assume that they are available and we show how to use them to build filters. The presentation so far showed how natural multistep filters can be obtained by simply replacing the multistep solution and its derivative wrt t by their interval extensions to obtain $DMS(\mathbf{t}, \mathbf{D}, t) = F(t, MS(\mathbf{t}, \mathbf{D}, t))$. It is not easy however to enforce backward consistency on a natural filter since the variables may occur in complex nonlinear expressions. This problem is addressed by mean-value filters that we now briefly explain.

Mean-value forms (MVFs) play a fundamental role in interval computations and are derived from the mean-value theorem. They correspond to problem linearizations around a point and result in filters that are systems of linear equations with interval coefficients and whose solutions can be enclosed reasonably efficiently. Mean-value forms are effective when the sizes of the boxes are sufficiently small, which is the case in ODEs. In addition, being linear equations, they allow for an easier treatment of the so-called *wrapping effect*, a crucial problem in interval methods for ODEs. As a consequence, mean-value forms are especially appropriate in our context and will produce filters which are efficiently amenable to backward consistency. The rest of this section describes how to obtain mean-value filters. Mean-value filters are presented in detail in [JVHD01] and in the technical report version of this paper. For the purpose of this paper, it is sufficient to observe that they lead to a system of linear equations with interval coefficients. More precisely, let $\mathbf{D}^- \in \mathbb{IR}^{n(k+1)}$ be the predicted box of variable $\mathbf{u}$ and define $\mathbf{X}$ as $\mathbf{D} - m(\mathbf{D}^-)$. A mean-value filter is a system of equations of the form $\sum_{i=0}^{k} A_i(t) X_i = K(t)$ where $A_i(t) \in \mathbb{R}^{n \times n}$, $i = 0, \dots, k$ and $K(t) \in \mathbb{IR}^n$. In general, for initial value problems, we will be interested in pruning the last predicted box D_k^-. Hence it is convenient to derive a mean-value filter which is explicit in D_k by isolating X_k to obtain

$$X_k = A_k(t)^{-1} K(t) - \sum_{i=0}^{k-1} \left(A_k(t)^{-1} A_i(t)\right) X_i. \tag{2}$$

which is an explicit mean-value filter ($A_k(t)^{-1}$ denotes an enclosure of the inverse of $A_k(t)$). It is easy to use an explicit mean-value filter to prune the predicted box D_k^- at time t_k given the boxes $D_0^*, \dots, D_{k-1}^*$ from

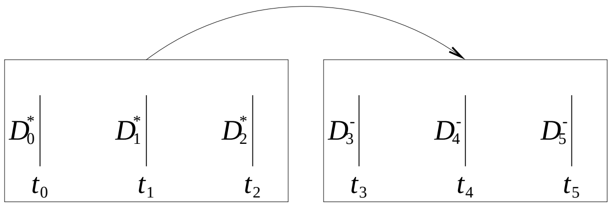

Figure 1: Intuition of the Globalization Process ($k = 3$).

the previous integration steps, since X_k (and thus D_k) has been isolated. The filter simply becomes

$$D_k = m(D_k^-) + A_k(t)^{-1}K(t) - \sum_{i=0}^{k-1} \left(A_k(t)^{-1}A_i(t)\right)(D_i^* - m(D_i^*))$$

and the pruned box D_k^* at time t_k is given by $D_k^* = D_k \cap D_k^-$. It follows directly that the explicit mean-value filter is backward-consistent in $\mathbf{D}^*$.

3.2 Global Filters

Mean-value filters may produce significant pruning of the boxes computed by the predictor process. However, they suffer from two limitations: the *wrapping effect* which is inherent in interval analysis and a *variable dependency* problem since the same boxes are used indirectly several times in a multistep method, possibly inducing a significant loss of precision. These two problems were addressed in [JVHD01] through global filters. The main idea underlying global filters is to cluster several mean-value filters together so that they do not overlap. The intuition is illustrated in Figure 1 for $k = 3$. It can be seen that the global filter prunes the 3 predicted boxes D_3^-, D_4^-, and D_5^- for times t_3, t_4, and t_5 using the boxes D_0^*, D_1^*, and D_2^* computed for times t_0, t_1, and t_2. Observe also that global filters do not overlap, i.e., the boxes D_0^*, D_1^*, and D_2^* will not be used in subsequent filters. More precisely, a global filter is a system of k successive explicit mean-value filters. It can be transformed into an explicit form $\mathbf{X}_1 = C(\mathbf{e_0})\mathbf{X}_0 + R(\mathbf{e_0})$ where $C(\mathbf{e_0}) \in \mathbb{IR}^{nk\times nk}$ and $R(\mathbf{e_0}) \in \mathbb{IR}^{nk}$. An interesting property of global filters is that each pruned box at times t_3, t_4, or t_5 can be computed only in terms of the predicted boxes and the boxes at times t_0, t_1, and t_2 by using Gaussian elimination. The resulting filter is backward(k)-consistent with respect to the resulting boxes. Finally, observe that global filters not only remove the variable dependency problem by globalizing the pruning process. They also produce square systems which makes it possible to apply standard techniques from one-step methods (e.g., local coordinate transformations and QR factorizations [Loh87]) to address the wrapping effect.

3.3 Hermite Filters

So far, we assumed the existence of interval extensions of p and $\partial p/\partial t$ and bounds on the error terms e and $\partial e/\partial t$. We now show how to use Hermite interpolation polynomials for this purpose. Informally speaking, a Hermite interpolation polynomial approximates a continuously differentiable function f which is known implicitly by its values and the values of its successive derivatives at various points. A Hermite interpolation polynomial is specified by imposing that its values and the values of its successive derivatives at some given points be equal to the values of f and of its derivatives at the same points. Note that the number of conditions (i.e., the number of successive derivatives that are considered) may vary at the different points.

Definition 7 (Hermite(σ) Interpolation Polynomial) *Consider the ODE* $u' = f(t,u)$ *and let* $\sigma = (\sigma_0, \dots, \sigma_k) \in \mathbb{N}^{k+1}$ *and* $\sigma_i \neq 0$ $(0 \le i \le k)$. *Let* $\sigma_s = \sum_{i=0}^{k} \sigma_i$, $u_i^{(0)} = u_i$, *and* $u_i^{(j)} = f^{(j-1)}(t_i, u_i)$ $(0 \le i \le k \;\&\; 0 \le j \le \sigma_i - 1)$. *The* Hermite($\sigma$) interpolation polynomial *wrt* f *and* $(\mathbf{t}, \mathbf{u})$ *is the unique polynomial* q *of degree* $\le \sigma_s - 1$ *satisfying* $q^{(j)}(t_i) = u_i^{(j)} \quad (0 \le j \le \sigma_i - 1 \;\&\; 0 \le i \le k)$.

Proposition 1 (Hermite Interpolation Polynomial) *The polynomial q satisfying conditions of definition 7 is given by*

$$q(t) = \sum_{i=0}^{k} \sum_{j=0}^{\sigma_i - 1} u_i^{(j)} L_{ij}(t) \tag{3}$$

where

$$\begin{array}{rcl} L_{i,\sigma_i-1}(t) & = & l_{i,\sigma_i-1}(t), \quad i = 0, \ldots, k \\ L_{ij}(t) & = & l_{ij}(t) - \sum_{\nu=j+1}^{\sigma_i-1} l_{ij}^{(\nu)}(t_i) L_{i\nu}(t), \quad i = 0, \ldots, k, \;\; j = 0, \ldots, \sigma_i - 2 \\ l_{ij}(t) & = & \frac{(t-t_i)^j}{j!} \prod_{\substack{\nu=0 \\ \nu \neq i}}^{k} \left(\frac{t - t_\nu}{t_i - t_\nu} \right)^{\sigma_\nu}, \quad i = 0, \ldots, k, \;\; j = 0, \ldots, \sigma_i - 1 \end{array} \tag{4}$$

It is easy to take interval extensions of a Hermite interpolation polynomial and of its derivatives. The only remaining issue is to bound the error terms. The following standard theorem (e.g., [SB80], [Atk88]) provides the necessary theoretical basis.

Proposition 2 (Hermite Error Term) *Let $p(\mathbf{t}, \mathbf{u}, t)$ be the Hermite(σ) interpolation polynomial in t wrt f and $(\mathbf{t}, \mathbf{u})$. Let $u(t) \equiv ms(\mathbf{t}, \mathbf{u}, t)$, $T = \Box\{t_0, \ldots, t_k, t\}$, $\sigma_s = \sum_{i=0}^{k} \sigma_i$ and $w(t) = \prod_{i=0}^{k} (t - t_i)^{\sigma_i}$. We have ($1 \leq i \leq n$)*
- $\exists\, \xi_i \in T : e_i(\mathbf{t}, \mathbf{u}, t) = \frac{1}{\sigma_s!} f_i^{(\sigma_s - 1)}(\xi_i, u(\xi_i)) w(t)$;
- $\exists\, \xi_{1,i}, \xi_{2,i} \in T : \frac{\partial e_i}{\partial t}(\mathbf{t}, \mathbf{u}, t) = \frac{1}{\sigma_s!} f_i^{(\sigma_s - 1)}(\xi_{1,i}, u(\xi_{1,i})) w'(t) + \frac{1}{(\sigma_s+1)!} f_i^{(\sigma_s)}(\xi_{2,i}, u(\xi_{2,i})) w(t)$.

How to use this proposition to bound the error terms? It suffices to take interval extensions of the formula given in the proposition and to replace $\xi_i, \xi_{1,i}, \xi_{2,i}$ by T and $u(\xi_i), u(\xi_{1,i}), u(\xi_{2,i})$ by a bounding box for the ODE over T. As a consequence, we can compute an effective relaxation of the ODE by specializing global filters with a Hermite interpolation and its error bound. In the following, filters based on Hermite(σ) interpolation are called *Hermite(σ) filters* and a global Hermite(σ) filter is denoted by GHF(σ).

4 Optimal Pruning in Hermite Filters

Let us summary what we have achieved so far. The basic idea of our approach is to approximate the ODE $\forall\, t : \; u' = f(t, u)$ by a filter $\forall\, t : FL(\mathbf{t}, \mathbf{D}, t)$. We have shown that a global filter prunes the last k boxes by using k successive mean-value filters and it addresses the wrapping effect and the variable dependency problem. We have also shown that a global filter can be obtained by using Hermite interpolation polynomials together with their error bounds. As a consequence, we obtain a filter $\forall\, \mathbf{e_0} : GHF(\sigma)(\mathbf{t}, \mathbf{D}, \mathbf{e_0})$ which can be used to prune the last k predicted boxes. The main remaining issue is to find an *evaluation time vector* $\mathbf{e_0}$ which maximizes pruning or, alternatively, which miminizes the sizes of the solution boxes in $GHF(\sigma)(\mathbf{t}, \mathbf{D}, \mathbf{e_0})$. More precisely, our main goal in choosing an evaluation time vector is to minimize the *local error* of the filter, i.e., the sizes of the boxes produced by the filter.

Definition 8 (Local Error of a Filter) *Let FL be a filter for ODE $u' = f(t, u)$. The local error of FL wrt $(\mathbf{t}_0, \mathbf{u}_0, t)$ is defined as $e_{loc}(FL, \mathbf{t}_0, \mathbf{u}_0, t) = \omega\left(\Box\{u_k \in \mathbb{R}^n \mid FL(\mathbf{t}, \mathbf{u}, t)\}\right)$.*

Observe that a global filter is obtained from several mean-value filters. Hence minimizing its local error amounts to minimizing the local error of individual mean-value filters. Moreover, since the local error is defined by evaluating the filter on real numbers, we can restrict attention, without loss of generality, to natural Hermite filters and do not need to consider their mean-value forms. To find an optimal evaluation time, we first derive the local error (Section 4.1). From the local error, we can then characterize the optimal evaluation time (Section 4.2). Two of the main results of this section are as follows:

1. For a sufficiently small stepsize $h = t_k - t_0$, the relative distance between the optimal evaluation time and the point t_k in a natural or mean-value Hermite filter depends only on the relative distances between the interpolation points $t_0, \ldots, t_k$ and on σ. It does not depend on the ODE itself.
2. From a practical standpoint, the optimal evaluation time can be precomputed once for all for a given step size and σ. This computation does not induce any overhead of the method.

The third main result is concerned with the order of a natural Hermite filter which is shown to be $O(h^{\sigma_s+1})$ where $\sigma_s = \sum_{i=0}^{k} \sigma_i$ when the evaluation point is chosen carefully (but not necessarily optimally!).

4.1 Local Error of a Natural Hermite Filter

To analyze the local error and determine the optimal evaluation time, we use standard asymptotical notations.

Notation 2 (Asymptotical Notations) *Consider two functions $f, g : \mathbb{R} \rightarrow \mathbb{R}$ and let $h > 0$. We use the following standard notations*

- $f(h) = O(g(h))$ *if* $\exists c > 0, \exists \varepsilon > 0 : h \leq \varepsilon \Rightarrow |f(h)| \leq c|g(h)|$;
- $f(h) = \Omega(g(h))$ *if* $\exists c > 0, \exists \varepsilon > 0 : h \leq \varepsilon \Rightarrow |f(h)| \geq c|g(h)|$;
- $f(h) = \Theta(g(h))$ *if* $f(h) = O(g(h))$ *and* $f(h) = \Omega(g(h))$.

The notations extend component-wise for vectors and matrices of functions.

Note that these notations characterize the behaviour of a function when h is sufficiently small. Asymptotic notations in computer science characterize, in general, the behaviour of algorithms when the size n of the problem becomes larger. These notations are simply obtained by substituting h by $1/n$. We also make a number of assumptions in this section. (Additional, more technical, assumptions are given in the appendix.) We assume that the step size h is given by $t_k - t_0$ and that the integration times are increasing, i.e., $t_0 < \ldots < t_k$. Moreover, we assume that the multistep solution ms is defined at $(\mathbf{t}_0, \mathbf{u}_0)$ or, in other words, that $\mathcal{O}$ has a solution going through $u_0, \ldots, u_{k-1}$ at times $t_0, \ldots, t_{k-1}$. We also use the notations $\sigma = (\sigma_0, \ldots, \sigma_k)$, $\sigma_s = \sum_{i=0}^{k} \sigma_i$, and $w(t) = \prod_{i=0}^{k} (t - t_i)^{\sigma_i}$. To characterize the local error of a natural Hermite filter, we first need a technical lemma which characterizes the behavior of the derivatives of the filter.

Lemma 1 *Consider an ODE $u' = f(t, u)$, let $p(\mathbf{t}, \mathbf{u}, t)$ be the Hermite(σ) interpolation polynomial in t wrt f and $(\mathbf{t}, \mathbf{u})$ and let $\Phi(t) = D_{u_k} \frac{\partial p}{\partial t}(\mathbf{t}, \mathbf{u}, t) - D_u f(t, p(\mathbf{t}, \mathbf{u}, t) + e) D_{u_k} p(\mathbf{t}, \mathbf{u}, t)$, $e \in \mathbb{R}^n$. Then, when $t - t_k = O(h)$ and h is sufficiently small, we have*

1. $\Phi(t) \approx I\lambda(t)$; 2. $\lambda(t) = \Theta(h^{-1})$ if $\lambda(t) \neq 0$; 3. $\lambda(t) \neq 0$ for $t_{k-1} < t < t_k$

where $\lambda(t)$ is defined by the formula

$$\begin{array}{l} \lambda(t) = \left(\left(\sum_{j=0}^{\sigma_k - 2} \beta_{j+1} \frac{(t-t_k)^j}{j!} \right) + \left(\sum_{j=0}^{\sigma_k - 1} \beta_j \frac{(t-t_k)^j}{j!} \right) \sum_{\nu=0}^{k-1} \frac{\sigma_\nu}{t_k - t_\nu} \right) \pi(t); \\ \beta_0 = 1, \beta_j = -\pi^{(j)}(t_k),\ j = 1, \ldots, \sigma_k - 1; \\ \pi(t) = \prod_{\nu=0}^{k-1} \left(\frac{t - t_\nu}{t_k - t_\nu} \right)^{\sigma_\nu}. \end{array} \qquad (5)$$

This lemma shows that $\Phi(t)$ is a $\Theta(h^{-1})$ almost diagonal matrix for $t_{k-1} < t < t_k$. Its proof is given in the appendix. We are now in position to characterize the local error of a natural Hermite filter.

Theorem 1 (Local Error of a Natural Hermite Filter) *Let FL be a natural Hermite(σ) filter for $u' = f(t, u)$ and assume that $t - t_k = O(h)$. With the notations of Lemma 1, we have*

1. *if $\Phi(t)$ is not singular, then $e_{loc}(FL, \mathbf{t}_0, \mathbf{u}_0, t) = |\Phi^{-1}(t)| \, (\Theta(h)|w(t)| + \Theta(h)|w'(t)|)$;*
2. *if $\Phi(t)$ is not singular, then $\Phi(t) = \Theta(h^{-1})$;*
3. *if $t_{k-1} < t < t_k$ and if h is sufficiently small, then $\Phi(t)$ is not singular;*

We are now ready to show how to find an optimal evaluation time for natural Hermite filters.

4.2 Optimal Evaluation Time for a Natural Hermite Filter

Our first result is fundamental and characterizes the order of a natural Hermite filter. It also hints on how to obtain an optimal evaluation time. Recall that the order of a method is the order of the local error minus 1.

Theorem 2 (Order of a Natural Hermite Filter) *Assume that $t - t_k = O(h)$ and let FL be a natural Hermite(σ) filter. With the notations of Lemma 1, we have*

1. *There exists t such that $t_{k-1} < t < t_k$ and $w'(t) = 0$;*
2. *If $t_{k-1} < t < t_k$, $w'(t) = 0$, and h is sufficiently small, then $e_{loc}(FL, \mathbf{t}_0, \mathbf{u}_0, t) = O(h^{\sigma_s+2})$;*
3. *If $w'(t) \neq 0$ and $\Phi(t)$ is not singular, then $e_{loc}(FL, \mathbf{t}_0, \mathbf{u}_0, t) = \Theta(h^{\sigma_s+1})$.*

Observe that the above theorem indicates that *the zeros of w' are evaluation times which lead to a method of a higher order for natural and mean-value Hermite filters* (provided that the matrix $\Phi(t)$ be non-singular at these points). This is the basis of our next result which describes a necessary condition for optimality.

Theorem 3 (Necessary Condition for Optimal Natural Hermite Filters) *Let FL be a natural Hermite(σ) filter and let $t_e \in \mathbb{R}$ be such that $e_{loc}(FL, \mathbf{t}_0, \mathbf{u}_0, t_e) = \min_{t-t_k=O(h)}\{e_{loc}(FL, \mathbf{t}_0, \mathbf{u}_0, t)\}$. We have that, for h sufficiently small, t_e is a zero of the function $\gamma(t) = \sum_{i=0}^{k} \frac{\sigma_i}{t-t_i}$*

Our next result specifies the number of zeros of the function γ as well as their locations.

Theorem 4 *The function γ in Theorem 3 has exactly k zeros $s_0, \ldots, s_{k-1}$ satisfying $t_i < s_i < t_{i+1}$.*

We are now ready to characterize precisely the optimal evaluation time for a natural Hermite filter.

Theorem 5 (Optimal Evaluation Time) *Let FL be a natural Hermite(σ) filter, let $t_e \in \mathbb{R}$ be such that $e_{loc}(FL, \mathbf{t}_0, \mathbf{u}_0, t_e) = \min_{t-t_k=O(h)}\{e_{loc}(FL, \mathbf{t}_0, \mathbf{u}_0, t)\}$, let λ and γ be the functions defined in Lemma 1 and Theorem 3 respectively, and let $s_0, \ldots, s_{k-1}$ be the zeros of γ. Then, for h sufficiently small,*

$$|(w/\lambda)(t_e)| = \min_{s \in \{s_0, \cdots, s_{k-1}\}} \{|(w/\lambda)(s)|\} \tag{6}$$

It is important to discuss the consequences of Theorem 3 in some detail. First observe that the *relative distance* between the optimal evaluation time t_e and the point t_k depends *only* on the *relative distances* between the interpolation points $t_0, \ldots, t_k$ and on the vector σ. In particular, it is independent from the ODE itself. For instance, for $k = 1$, we have $\gamma(t) = \frac{\sigma_0}{t-t_0} + \frac{\sigma_1}{t-t_1}$ and γ has a single zero given by $t_e = \frac{\sigma_1 t_0 + \sigma_0 t_1}{\sigma_0 + \sigma_1}$. In addition, if $\sigma_0 = \ldots = \sigma_k$, then the zeros of γ are independent from σ. In particular, for $k = 1$, we have $t_e = (t_0 + t_1)/2$. As a consequence, for a given σ and step size h, the relative distance between t_k and an optimal evaluation time t_e can be computed once at the beginning of the integration. In addition, since it does not depend on the ODE itself, this relative distance can be precomputed and stored for a variety of step sizes and vectors σ. The overhead of choosing an optimal evaluation time is thus negligible. Finally, it is worth

k	1	2	3	4	5	6
$(t_e - t_k)/h$	-0.5000	-0.2113	-0.1273	-0.0889	-0.0673	-0.0537

Table 1: Relative Distance between the Rightmost Zero t_e of γ and t_k when $\sigma_0 = \ldots = \sigma_k$.

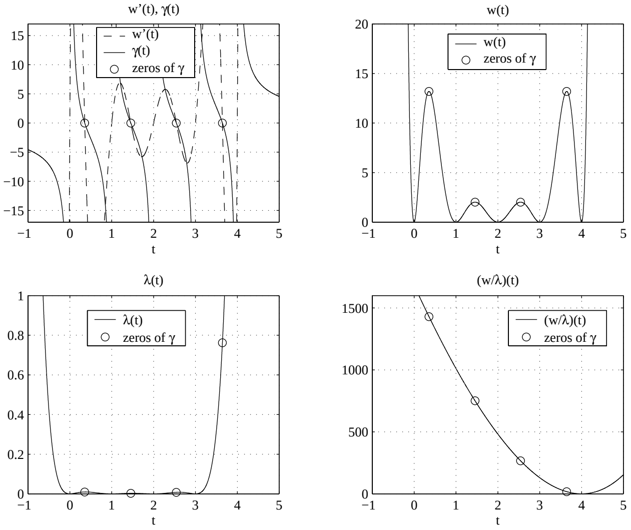

Figure 2: The functions γ, w, w', λ and w/λ for the case $k = 4, \sigma = (2, 2, 2, 2, 2)$.

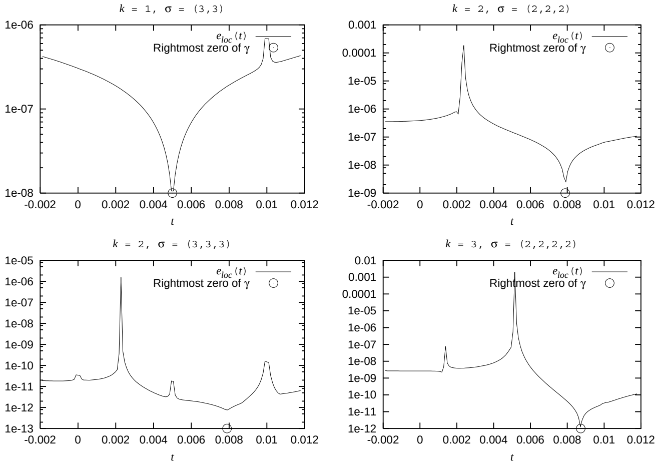

Figure 3: Local Error of Global Hermite Filters as a Function of the Evaluation Time for the Lorentz System.

stressing that any zero of function γ in Theorem 3 gives an $O(h^{\sigma_s+1})$ order for the Hermite filter (provided that the matrix $\Phi(t)$ be non-singular at that zero). Hence any such zero is in fact an appropriate evaluation time, although it is not necessarily optimal. In our experiments, the right-most zero was always the optimal evaluation time, although we have not been able to prove this result theoretically.

We now illustrate the theoretical results experimentally. Table 4.2 gives approximative values of the relative distance between the rightmost zero of the function γ and the point t_k $(1 \leq k \leq 6)$, for $\sigma_0 = \ldots = \sigma_k$, and $h = t_k - t_0$. Observe that, for two interpolation points, t_e is in the middle of t_0 and t_1. It then moves closer and closer to t_k for larger values of k. Figure 2 illustrates the functions γ, w, w', λ, and w/λ for $k = 4, \sigma = (2, 2, 2, 2, 2)$ and their sometimes complex interactions. The top-left figure shows the function w' and γ, as well as the zeros of γ. The top-right figure shows the function w with the zeros of γ in superposition. The bottom-left figure shows function λ with the zeros of γ in superposition. The bottom-right picture shows the function w/λ and the zeros of γ. It can be seen that the right-most zero minimizes the local error in this example. Figure 3 illustrates our theoretical results experimentally on a specific ODE. It plots the local error of several global Hermite filters (GHF) as a function of the evaluation time for the Lorenz system (e.g., [HNW87]). It is assumed that $t_{i+1} - t_i$ is constant $(0 \leq i \leq 2k - 2)$. In addition, we assume that, in each mean-value filter composing the GHF, the distance between the evaluation time and the rightmost interpolation point is constant. In the graphs, $[t_0, t_k] = [0, 0.01]$ and $h = t_k - t_0 = 0.01$. The figure also shows the rightmost zero of the function γ as obtained from Table 4.2. As we can see, the rightmost zero of γ is a very good approximation of the optimal evaluation time of the filter for all the cases displayed.

5 Theoretical Analysis

We analyze the cost of our algorithm based on the global Hermite filter method GHF(σ) and compare it to Nedialkov's IHO(p, q) method [NJ99], the best interval method we know of. Nedialkov shows that the IHO method outperforms interval Taylor series methods (e.g. Lohner's AWA [Loh87]). The step size is given by $h = t_k - t_0$ and we use the same step size in GHF(σ) and IHO(p, q). Let $\sigma_m = \max(\sigma)$ and $\sigma_s = \sigma_0 + \ldots + \sigma_k$. At each step i, we use the following assumptions when comparing GHF(σ) and IHO(p, q):

1. The bounding box process uses a Taylor series method ([CR96], [NJC99]) of order σ_s. Moreover, we assume that $B_{ik} = \ldots = B_{(i+1)k-1}$, i.e., the function computes a single bounding box over $[t_{ik-1}, t_{(i+1)k-1}]$;

2. The predictor process uses Moore's Taylor method [Moo66] of order $q + 1$ (same order as the predictor used in IHO(p, q)) to compute the boxes $\mathbf{D}_i^-$;

3. We choose the rightmost zero of function γ (see Section 4.2 and Table 4.2) as an evaluation time in the Hermite filters. Consequently, the GHF(σ) method is of order $\sigma_s + 1$.

For simplicity of the analysis, we assume that (the natural encoding of) function f contains only arithmetic operations. We denote by N_1 the number of $*, /$ operations in f, by N_2 the number of $\pm$ operations, and by N the sum $N_1 + N_2$. We also assume that the cost of evaluating $D_u f^{(r)}$ is n times the cost of evaluating $f^{(r)}$. We report separately interval arithmetic operations involved in (1) products of a real and an interval matrix which arise in the pruning step (Cost-1) and (2) the generation of Jacobians (Cost-2). Note that Cost-1 is a fixed cost in the sense that it is independent from the ODE. Cost-2 is a variable cost which increases as the expression of f becomes more complex.

Methods of the Same Order We first compare the costs of the GHF(σ) and IHO(p, q) methods when we assume that $p + q = \sigma_s$ and $q \in \{p, p + 1\}$. The methods GHF and IHO are thus of the same order $(\sigma_s + 1)$. Table 2 reports the *main* cost of a step in the IHO method and our GHF method. It also shows

	Cost-1	Cost-2
IHO	$-$	$2\lceil \frac{\sigma_s}{2} \rceil^2 nN_1 + O(\sigma_s nN_2)$
GHF	$7k^3n^3$	$((\sigma_m - 1)^2 + 1)knN_1 + \sigma_m knN_2$
GHF-1	$-$	$(\lfloor \frac{\sigma_s - 1}{2} \rfloor^2 + 1)nN_1 + O(\sigma_s nN_2)$
GHF-2	$(\frac{7}{8}\sigma_s - \frac{21}{4})\sigma_s^2 n^3$	$(\sigma_s - 2)nN$

Table 2: Cost Analysis : Methods of the Same Order.

	Cost-2
IHO	$2\lfloor \frac{\sigma_s - 1}{2} \rfloor^2 nN_1 + O(\sigma_s nN_2)$
GHF-1	$(\lfloor \frac{\sigma_s - 1}{2} \rfloor^2 + 1)nN_1 + O(\sigma_s nN_2)$

Table 3: Cost Analysis : Methods of Different Orders.

the complexity of two particular cases of GHF. The first case (GHF-1) corresponds to a polynomial with only two interpolation points ($k = 1$) and $|\sigma_1 - \sigma_0| \leq 1$, while the second case (GHF-2) corresponds to a polynomial imposing two conditions on every interpolation points ($\sigma_0 = \ldots = \sigma_k = 2$). The first main result is that GHF-1 *is always cheaper than* IHO, which means that our method can always be made to run faster by choosing only two interpolation points. (The next section will show that improvement in accuracy is also obtained in this case). GHF-2 is more expensive than GHF-1 and IHO when f is simple because in this case the Jacobians are cheap to compute and the fixed cost Cost-1 becomes large wrt Cost-2. However, *when f contains many operations (which is the case in many practical applications),* GHF-2 *can become substantially faster* because Cost-1 in GHF-2 is independent of f and Cost-2 is substantially smaller in GHF-2 than in GHF-1 and IHO. It also shows the versatility of the approach that can be taylored to the application at hand.

One-Step Methods of Different Orders We now show that our approach can be made both asymptotically more precise and faster. Consider the costs of the IHO(p, q) and GHF(σ_0, σ_1) methods when we assume that $|\sigma_1 - \sigma_0| \leq 1, p + q = \sigma_s - 2$ and $q \in \{p, p+1\}$. Under these conditions, IHO is a method of order $\sigma_s - 1$, while GHF is a method of order $\sigma_s + 1$. Table 3 reports the main cost of a step in IHO and GHF. As can be seen from the table, GHF is always cheaper than IHO. *The* GHF *method is thus both asymptotically more precise (by two orders of magnitude) and faster than the* IHO *method.*

6 Experimental Analysis

We now report experimental results of a C++ implementation of our approach on a Sun Ultra 10 workstation with a 333 MHz UltraSparc CPU. The underlying interval arithmetic and automatic differentiation packages are PROFIL/BIAS [Knu94] and FADBAD/TADIFF [BS96, BS97] respectively. Many of the tested examples are classical benchmarks for ODE solvers. These problems are taken from various domains, including chemistry, biology, mechanics, physics and electricity. The equation, initial conditions and interval of integration for each initial value problem are given in Appendix B. Note that, although we could use interval initial conditions, we consider only point initial conditions to compare the different methods. The “full Brusselator” (BRUS) and the “Oregonator” (OREG), a stiff problem, model famous chemical reactions, the Lorenz system (LOR) is an example of the so-called “strange attractors”, the Two-Body problem (2BP) comes from mechanics, and the van der Pol (VDP) equation describes an electrical circuit. All these problems are described in detail in [HNW87]. We also consider a problem from molecular biology (BIO), the Stiff DETEST problem D1 [Enr75], and another stiff problem (GRI) from [Gri72]. Finally, we consider four problems (LIEN, P1,

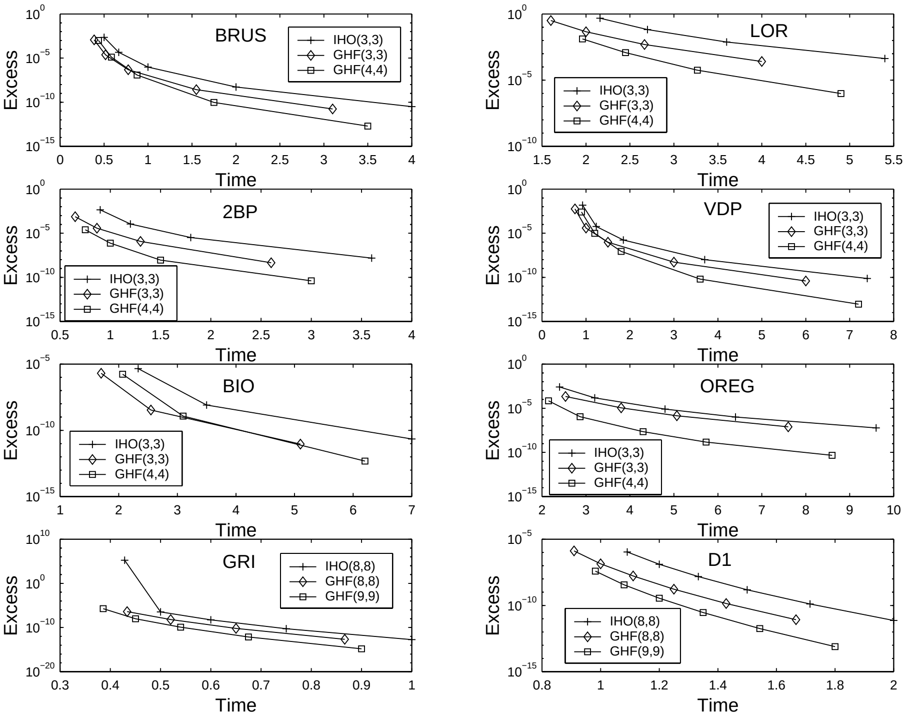

Figure 4: Experimental Comparison of the Methods IHO(p, p), GHF(p, p) and GHF($p + 1, p + 1$).

P2, P3) where the ODE has a more complex expression (i.e. the function f contains many operations). They are taken from [Per00]. The experimental results follow the same assumptions as in the theoretical analysis section and we make three types of comparisons: (1) one-step methods of the same order; (2) one-step methods of different orders, but of similar cost; and (3) multistep versus one-step methods of the same order. The figures report the global excess (where the global excess at point t_i is given by the infinite norm of the width of the enclosure D_i at t_i, i.e., the quantity $\|\omega(D_i)\|_\infty$) at the end of the interval of integration of the compared IHO and GHF methods. These figures are based on the tables given in Appendix C where we report the global excess, the excess ratio (an excess ratio higher than 1 means that GHF is more precise), the execution time of both methods (in seconds) and the time ratio (a time ratio higher than 1 means that GHF is faster).

One-Step Methods Figure 4 plots the excess as a function of the execution time in the methods IHO(p, p), GHF(p, p) and GHF($p + 1, p + 1$) for the problems of Tables 4 and 5. We take $p = 8$ for GRI and D1 and $p = 3$ for the other problems. As we can see from the figure, the curve of IHO is always above the curves of the GHF methods, showing that IHO is less precise than the GHF methods for a given execution time or, alternatively, IHO is slower than the GHF methods for a given precision. Thus, although GHF(p, p) may sometimes be less precise than IHO(p, p) for a given step size in the stiff problems OREG, GRI and D1, GHF(p, p) still performs better than IHO(p, p) because the cost of a step is less in GHF(p, p) than in IHO(p, p). The figure also shows that GHF($p + 1, p + 1$) performs better than GHF(p, p) in all cases. Furthermore, our results confirm that IHO(p, p) and GHF(p, p) are methods of the same order, and that GHF($p + 1, p + 1$) is a method of higher order.

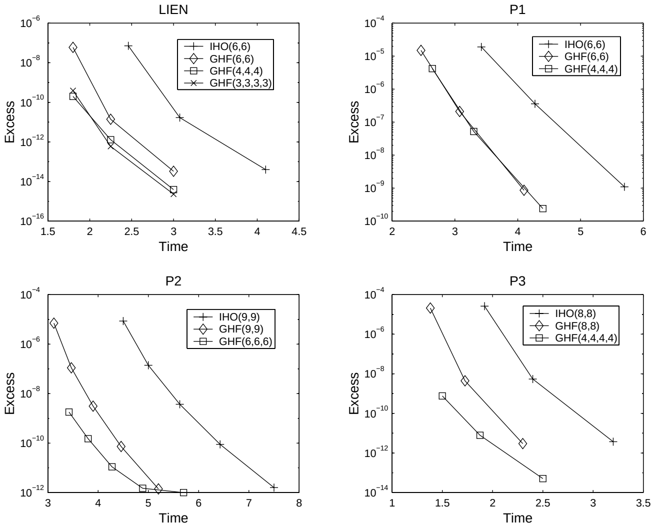

Figure 5: Experimental Comparison of Multistep and One-Step Methods.

Multistep Versus One-Step Methods We now compare multistep GHF methods versus IHO and one-step GHF methods of the same order in problems where the ODE has a more complex expression (i.e., f contains many operations). Figure 5 plots the excess as a function of the execution time in several methods for the problems of Table 6. Again, the curve of IHO is always above the curves of the GHF methods, showing that the latter perform better on these problems. Furthermore, we observe that the curves of the one-step GHF methods are above those of the multistep GHF methods. Multistep GHF methods perform thus better in these cases.

Summary *The results indicate that our method produces orders of magnitude improvements in accuracy and runs faster than the best known method.* The theoretical results are also confirmed by the experiments. When f contains many operations, using many interpolation points is particularly effective. For very complex functions, the gain in computation time could become substantial. When f is simple, using few interpolation points becomes more interesting.

7 Conclusion

This paper considered a constraint satisfaction approach to initial value problems for parametric ordinary differential equations introduced in [DJVH98, JDVH99, JVHD01]. It solved the main theoretical and practical open issue in this approach: the choice of an optimal evaluation time for the filters. In particular, it showed that the optimal evaluation time for a filter is independent of the ODE itself and can be precomputed, thus inducing no overhead for the method. This result has important theoretical and practical consequences. The theoretical results show that the constraint satisfaction approach provides a quadratic improvement in accuracy over the best interval methods, while improving their running times. The experimental results on standard benchmarks confirm the theoretical results.

References

[Atk88] K. E. Atkinson. *An introduction to Numerical Analysis*. John Wiley & Sons, New York, 1988.

[Ber98] Berz, M. and Makino, K. Verified Integration of ODEs and Flows Using Differential Algebraic Methods on High-Order Taylor Models. *Reliable Computing*, 4:361-369, 1998.

[BS96] C. Bendsten and O. Stauning. *FADBAD, a Flexible C++ Package for Automatic Differentiation Using the Forward and Backward Methods*. Technical Report 1996-x5-94, Technical University of Denmark, 1996.

[BS97] C. Bendsten and O. Stauning. *TADIFF, a Flexible C++ Package for Automatic Differentiation Using Taylor Series*. Technical Report 1997-x5-94, Technical University of Denmark, April 1997.

[CB99] J. Cruz, and P. Barahona. An Interval Constraint Approach to Handle Parametric Ordinary Differential Equations for Decision Support. *Proceedings of EKBD-99*, 93-108, 1999.

[CR96] G. F. Corliss and R. Rihm. Validating an a Priori Enclosure Using High-Order Taylor Series. In *Scientific Computing, Computer Arithmetic, and Validated Numerics*, pages 228-238, 1996.

[DJVH98] Y. Deville, M. Janssen, and P. Van Hentenryck. Consistency Techniques in Ordinary Differential Equations. In *CP'98*, Pisa, Italy, October 1998.

[Eij81] Eijgenraam, P. *The Solution of Initial Value Problems Using Interval Arithmetic*. Mathematical Centre Tracts No. 144. Stichting Mathematisch Centrum, Amsterdam, 1981.

[Enr75] W. H. Enright, T. E. Hull, and B. Lindberg. Comparing Numerical Methods for Stiff Systems of ODEs. *BIT*, 15:10-48, 1975.

[Gri72] R. D. Grigorieff. *Numerik gewöhnlicher Differentialgleichungen 1*. Teubner, Stuttgart, 1972.

[HNW87] E. Hairer, S.P. Nørsett, G. Wanner. *Solving Ordinary Differential Equations I*. Springer-Verlag, Berlin, 1987.

[JDVH99] M. Janssen, Y. Deville, and P. Van Hentenryck. Multistep Filtering Operators for Ordinary Differential Equations. In *CP'99*, Alexandria, VA, October 1999.

[JVHD01] M. Janssen, P. Van Hentenryck, and Y. Deville A Constraint Satisfaction Approach to Parametric Differential Equations In *Joint International Conference on Artificial Intelligence (IJCAI-2001)*, Seattle, WA, August 2001.

[Knu94] O. Knüppel. PROFIL/BIAS - A Fast Interval Library. *Computing*, 53(3-4), pp. 277-287, 1994.

[Kru69] Krueckeberg, F. Ordinary Differential Equations. In E. Hansen, editor, *Topics in Interval Analysis*, page 91-97. Clarendon Press, Oxford, 1969.

[Loh87] Lohner R. J. Enclosing the Solutions of Ordinary Initial and Boundary Value Problems. In *Computer Arithmetic: Scientific Computation and Programming Languages*, Wiley, 1987.

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

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

[Ned99] N. S. Nedialkov. Computing Rigorous Bounds on the Solution of an Initial Value Problem for an Ordinary Differential Equation, Ph.D. Thesis, Computer Science Dept, Univ. of Toronto, 1999.

[NJ99] N.S. Nedialkov and K.R. Jackson. An Interval Hermite-Obreschkoff Method for Computing Rigorous Bounds on the Solution of an Initial Value Problem for an ODE, *Developments in Reliable Computing*, Kluwer, 1999.

[NJC99] N. S. Nedialkov, K. R. Jackson and G. F. Corliss. Validated Solutions of Initial Value Problems for Ordinary Differential Equations. *Applied Mathematics and Computation*, 105, pp. 21-68, 1999.

[Per00] L. Perko. *Differential Equations and Dynamical Systems*. Springer-Verlag, New York, 2000.

[Rih98] R. Rihm. Implicit Methods for Enclosing Solutions of ODEs. *J. of Universal Computer Science*, 4(2), 1998.

[SB80] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag, New York, 1980.

[VHMD97] P. Van Hentenryck, L. Michel, and Y. Deville. *Numerica: a Modeling Language for Global Optimization*. The MIT Press, Cambridge, Mass., 1997.

A Proof of the Main Results.

The following assumptions are used in the proofs. We assume that the step size h is given by $t_k - t_0$ and that the integration times are increasing, i.e., $t_0 < \ldots < t_k$. We assume that the interval extension F of f is sufficiently precise, i.e.,

$$F(t, D) \approx f(t, D)$$

when $\omega(D)$ is sufficiently small. We also assume (see [Ned99]) that

$$\omega \left(\frac{1}{(j+1)!} F^{(j)}(T, B(T)) \right) = \Theta(h)$$

where $B(T)$ is a bounding box of $u' = f(t, u)$ over T with respect to a given $(\mathbf{t}, \mathbf{D})$. Finally, we assume that the multistep solution ms is defined at $(\mathbf{t}_0, \mathbf{u}_0)$ or, in other words, that $\mathcal{O}$ has a solution going through $u_0, \ldots, u_{k-1}$ at times $t_0, \ldots, t_{k-1}$. We also use the notations introduced in Theorem 2. In particular, we assume that $\sigma = (\sigma_0, \ldots, \sigma_k)$, $\sigma_s = \sum_{i=0}^{k} \sigma_i$, and $w(t) = \prod_{i=0}^{k} (t - t_i)^{\sigma_i}$. Finally, if $x = (x_1, \ldots, x_n)$ and $g = (g_1, \ldots, g_n)$, then $D_x g$ denotes the matrix

$$\begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \ldots & \frac{\partial g_1}{\partial x_n} \\ \ldots & \ldots & \ldots \\ \frac{\partial g_n}{\partial x_1} & \ldots & \frac{\partial g_n}{\partial x_n} \end{bmatrix}$$

We first prove the following lemma.

Lemma 2 *Consider an ODE $u' = f(t, u)$, let $p(\mathbf{t}, \mathbf{u}, t)$ be the Hermite(σ) interpolation polynomial in t wrt f and $(\mathbf{t}, \mathbf{u})$ and let $\Phi(t) = D_{u_k} \frac{\partial p}{\partial t}(u_k, t) - D_u f(t, p(u_k, t)) D_{u_k} p(u_k, t)$. If $t_{k-1} < t < t_k$, then $(D_{u_k} \frac{\partial p_i}{\partial t}(u_k, t))_i = \Theta(h^{-1})$ for $i = 1, \ldots, n$.*

Proof Assume that $t - t_k = O(h)$. Let $t_e \in]t_{k-1}, t_k[$, $i \in 1..n$ and $q(t) = D_{u_k} p_i(u_k, t)$. Observe that $q_i^{(j)}(t) = O(h^{-j})$ or $h^j q_i^{(j)}(t) = O(1)$, for all $j \geq 0$. By definition of p, for $l = 0, \ldots, k$ and $j = 1, \ldots, \sigma_i - 1$, we have $\Phi^{(j-1)}(t_l) = 0$ and thus $q_i^{(j)}(t_l) = O(h^{1-j})$ or $h^j q_i^{(j)}(t_l) = O(h)$. Furthermore, for $l = 0, \ldots, k-1$, $q_i(t_l) = 0 = O(h)$ and $q_i(t_k) = O(1)$.

By continuity and by Rolle's Theorem, for h sufficiently small, if $h^j q_i^{(j)}$ has at least n_j distinct $O(h)$ values at $\xi_1, \ldots, \xi_{n_j}$, then $h^{j+1} q_i^{(j+1)}$ has at least $n_j - 1$ distinct zeros $\zeta_1, \ldots, \zeta_{n_j - 1}$, with $\xi_1 < \zeta_1 < \xi_2 < \ldots < \xi_{n_j - 1} < \zeta_{n_j - 1} < \xi_{n_j}$. Thus, $\zeta_1, \ldots, \zeta_{n_j - 1}$ are distinct from t_l if $\Phi^{(j)}(t_l) = 0$ (since $\Phi^{(j)}(t_l) = 0 \Rightarrow \Phi^{(j-1)}(t_l) = 0$), $l = 0, \ldots, k$.

Let r_j be the minimum number of $O(h)$ values of $h^j q_i^{(j)}$ at distinct *interpolation* points and n_j the minimum number of $O(h)$ values of $h^j q_i^{(j)}$ at distinct points. We can write :

$$\begin{array}{rcl} n_j & = & n_{j-1} - 1 + r_j \\ & = & n_{j-2} - 2 + r_{j-1} + r_j \\ & = & n_{j-3} - 3 + r_{j-2} + r_{j-1} + r_j \\ & \vdots & \\ & = & n_1 - (j-1) + \sum_{\nu=2}^{j} r_\nu \end{array} \tag{7}$$

Let us assume that $hq_i'(t_e) = 0$. Since the $n_0 - 1$ distinct zeros of hq_i' arising from the n_0 distinct zeros of q_i are strictly smaller than t_{k-1}, they are distinct from $t_e > t_{k-1}$. Thus, we have $n_1 = n_0 - 1 + 1 + r_1 = r_0 + r_1$. In particular, $n_{\sigma_s - 1} = r_0 + r_1 - (\sigma_s - 2) + \sum_{\nu=2}^{\sigma_s - 1} r_\nu = 2 - \sigma_s + \sum_{\nu=0}^{\sigma_s - 1} r_\nu$. We can easily verify that $\sum_{\nu=0}^{\sigma_s - 1} r_\nu = \sigma_s - 1$. We obtain $n_{\sigma_s - 1} = 1$.

However, since $q_i^{(\sigma_s - 1)}(t) \neq 0$, i.e. $q_i^{(\sigma_s - 1)}$ has no zeros, and $n_{\sigma_s - 1} = 0$, we have a contradiction. As a consequence, for h sufficiently small, we must have $q_i'(t_e) \neq 0$. □

We are now in position to prove Lemma 1.

Lemma 1 *Consider an ODE $u' = f(t,u)$, let $p(\mathbf{t},\mathbf{u},t)$ be the Hermite(σ) interpolation polynomial in t wrt f and $(\mathbf{t},\mathbf{u})$ and let $\Phi(t) = D_{u_k}\frac{\partial p}{\partial t}(\mathbf{t},\mathbf{u},t) - D_u f(t,p(\mathbf{t},\mathbf{u},t)+e)D_{u_k}p(\mathbf{t},\mathbf{u},t)$, $e \in \mathbb{R}^n$. Then, when $t - t_k = O(h)$ and h is sufficiently small, we have*

1. $\Phi(t) \approx I\lambda(t)$; 2. $\lambda(t) = \Theta(h^{-1})$ if $\lambda(t) \neq 0$; 3. $\lambda(t) \neq 0$ for $t_{k-1} < t < t_k$

where $\lambda(t)$ is defined by the formula

$$\begin{array}{l}
\lambda(t) = \left(\left(\sum_{j=0}^{\sigma_k-2}\beta_{j+1}\frac{(t-t_k)^j}{j!}\right) + \left(\sum_{j=0}^{\sigma_k-1}\beta_j\frac{(t-t_k)^j}{j!}\right)\sum_{\nu=0}^{k-1}\frac{\sigma_\nu}{t_k-t_\nu}\right)\pi(t);\\
\beta_0 = 1, \beta_j = -\pi^{(j)}(t_k),\ j = 1,\ldots,\sigma_k - 1;\\
\pi(t) = \prod_{\nu=0}^{k-1}\left(\frac{t-t_\nu}{t_k-t_\nu}\right)^{\sigma_\nu}.
\end{array} \tag{8}$$

Proof Let $i \in 1..n$ and $q(t) = D_{u_k}p_i(\mathbf{t},\mathbf{u},t)$. From the definition of Hermite interpolation polynomials (see Proposition 1), we can rewrite $q(t)$ and $q'(t)$ as follows

$$\begin{array}{rcl}
q(t) & = & \sum_{j=0}^{\sigma_k-1} c_j L_{kj}(t)\\
& = & \sum_{j=0}^{\sigma_k-1} c_j \sum_{\nu=j}^{\sigma_k-1}\alpha_{\nu j} l_{k\nu}(t)\\
& = & \sum_{j=0}^{\sigma_k-1} d_j l_{kj}(t)\\
& = & \left(\sum_{j=0}^{\sigma_k-1} d_j\frac{(t-t_k)^j}{j!}\right)\pi(t)\\
\\
q'(t) & = & \left(\left(\sum_{j=0}^{\sigma_k-2} d_{j+1}\frac{(t-t_k)^j}{j!}\right) + \left(\sum_{j=0}^{\sigma_k-1} d_j\frac{(t-t_k)^j}{j!}\right)\sum_{\nu=0}^{k-1}\frac{\sigma_\nu}{t_k-t_\nu}\right)\pi(t)
\end{array} \tag{9}$$

where $\alpha_{\nu j} \in \mathbb{R}, c_j, d_j \in \mathbb{R}^n$ are independent of t. From Proposition 1 and by definition of $q(t)$, it follows that

$$d_j = c_0\beta_j + O(h^{1-j}) = e_i\beta_j + O(h^{1-j}) \tag{10}$$

where $\beta_j = \Theta(h^{-j})$ (since $t_0 < \ldots < t_k$) and e_i is the i-th canonical vector. Let $\Phi = (\phi_1^T,\ldots,\phi_n^T)^T$. As $t - t_k = O(h)$,

$$\begin{array}{rcl}
\phi_i(t) & = & q'(t) - D_u f_i(t,p(\mathbf{t},\mathbf{u},t)+e)\ D_{u_k}p(\mathbf{t},\mathbf{u},t)\\
& = & q'(t) - O(1)O(1)
\end{array} \tag{11}$$

For h sufficiently small, we can write

$$d_j \approx e_i\beta_j \tag{12}$$

and

$$\phi_i(t) \approx q'(t) \approx e_i\lambda(t) \tag{13}$$

We have $\lambda(t) = \Theta(h^{-1})$ if $\lambda(t) \neq 0$ and, by Lemma 2, for $t_{k-1} < t < t_k$ and h sufficiently small, $\lambda(t) \neq 0$. □

Theorem 1 (Local Error of a Natural Hermite Filter) *Let FL be a natural Hermite(σ) filter for $u' = f(t,u)$ and assume that $t - t_k = O(h)$. With the notations of Lemma 1, we have*

1. *if $\Phi(t)$ is not singular, then $e_{loc}(FL,\mathbf{t}_0,\mathbf{u}_0,t) = |\Phi^{-1}(t)|\left(\Theta(h)|w(t)| + \Theta(h)|w'(t)|\right)$;*
2. *if $\Phi(t)$ is not singular, then $\Phi(t) = \Theta(h^{-1})$;*
3. *if $t_{k-1} < t < t_k$ and if h is sufficiently small, then $\Phi(t)$ is not singular;*

Proof Consider the relation

$$r(\mathbf{t},(\mathbf{u}_0,v),t) \Leftrightarrow \exists e \in E(t), \exists de \in DE(t) : \delta(\mathbf{t},(\mathbf{u}_0,v),e,de,t) = 0 \tag{14}$$

where

$$\begin{array}{l}\delta(\mathbf{t},(\mathbf{u}_0,v),e,de,t) = \frac{\partial p}{\partial t}(\mathbf{t},(\mathbf{u}_0,v),t) + de - f(t,p(\mathbf{t},(\mathbf{u}_0,v),t)+e);\\ E(t) = B_1(t)w(t);\\ DE(t) = B_1(t)w'(t) + B_2(t)w(t);\\ B_1(t) = \frac{1}{\sigma_s!}F^{(\sigma_s-1)}(T,B(T));\\ B_2(t) = \frac{1}{(\sigma_s+1)!}F^{(\sigma_s)}(T,B(T));\\ B(T) \text{ is a bounding box of } u' = f(t,u) \text{ over } T = \Box\{t_0,\ldots,t_k,t\} \text{ wrt } (\mathbf{t}_0,\mathbf{u}_0);\\ p(\mathbf{t},(\mathbf{u}_0,v),t) \text{ is the Hermite}(\sigma) \text{ interpolation polynomial in } t \text{ wrt } f \text{ and } (\mathbf{t},(\mathbf{u}_0,v)).\end{array} \tag{15}$$

Since $t-t_k = O(h)$, it follows that, for h sufficiently small, $F(t,p(\mathbf{t},(\mathbf{u}_0,v),t)+E(t)) \approx f(t,p(\mathbf{t},(\mathbf{u}_0,v),t)+E(t))$. The set $\{v \in \mathbb{R}^n \mid r(\mathbf{t},(\mathbf{u}_0,v),t)\}$ is thus a good approximation of the set $\{v \in \mathbb{R}^n \mid FL(\mathbf{t},(\mathbf{u}_0,v),t)\}$ provided that $B(T)$ is the bounding box used in FL. Consider now two arbitrary vectors $v_1, v_2 \in \mathbb{R}^n$ such that

$$\delta(\mathbf{t},(\mathbf{u}_0,v_1),e_1,de_1,t) = \delta(\mathbf{t},(\mathbf{u}_0,v_2),e,de,t) = 0 \tag{16}$$

for some $e_1, e \in E(t)$ and $de_1, de \in DE(t)$. By Taylor's Theorem, $\delta(\mathbf{t},(\mathbf{u}_0,v_2),e,de,t)$ is equal to

$$\delta(\mathbf{t},(\mathbf{u}_0,v_1),e_1,de_1,t) + \Phi(t)(v_2-v_1) + \Psi(t)(e_1-e) + de - de_1 + O(\|v_2-v_1\|^2 + \|e-e_1\|^2) \tag{17}$$

where

$$\begin{array}{rcl}\Phi(t) & = & D_v\frac{\partial p}{\partial t}(\mathbf{t},(\mathbf{u}_0,v_1),t) - D_u f(t,p(\mathbf{t},(\mathbf{u}_0,v_1),t)+e_1)D_v p(\mathbf{t},(\mathbf{u}_0,v_1),t)\\ \Psi(t) & = & D_u f(t,p(\mathbf{t},(\mathbf{u}_0,v_1),t)+e_1)\end{array} \tag{18}$$

Since $t - t_k = O(h)$, we have, for h sufficiently small, that

$$\Phi(t)(v_2-v_1) + \Psi(t)(e_1-e) + de - de_1 \approx 0 \tag{19}$$

When the matrix $\Phi(t)$ is non singular, we can write

$$v_1 - v_2 \approx \Phi^{-1}(t)\left(\Psi(t)(e_1-e) + de - de_1\right). \tag{20}$$

Since the two vectors are chosen arbitrarily, it follows that

$$\begin{array}{rcl}e_{loc}(FL,\mathbf{t}_0,\mathbf{u}_0,t) & \approx & |\Phi^{-1}(t)|\left(|\Psi(t)|\omega(E(t)) + \omega(DE(t))\right)\\ & \approx & |\Phi^{-1}(t)|\left((|\Psi(t)|d_1(t) + d_2(t))|w(t)| + d_1(t)|w'(t)|\right)\end{array} \tag{21}$$

where $d_1(t) = \omega(B_1(t))$ and $d_2(t) = \omega(B_2(t))$. Since, by hypothesis, $d_1(t), d_2(t) = \Theta(h)$, $t - t_k = O(h)$, and $\Psi(t) = O(1)$, we have

$$e_{loc}(FL,\mathbf{t}_0,\mathbf{u}_0,t) = |\Phi^{-1}(t)|\left(\Theta(h)|w(t)| + \Theta(h)|w'(t)|\right) \tag{22}$$

which proves Point 1. Points 2 and 3 are now direct consequences of Point 1 and of Lemma 1. □

Theorem 2 (Order of a Natural Hermite Filter) *Assume that $t - t_k = O(h)$ and let FL be a natural Hermite(σ) filter. With the notations of Lemma 1, we have*

1. *There exists t such that $t_{k-1} < t < t_k$ and $w'(t) = 0$;*
2. *If $t_{k-1} < t < t_k$, $w'(t) = 0$, and h is sufficiently small, then $e_{loc}(FL,\mathbf{t}_0,\mathbf{u}_0,t) = O(h^{\sigma_s+2})$;*
3. *If $w'(t) \neq 0$ and $\Phi(t)$ is not singular, then $e_{loc}(FL,\mathbf{t}_0,\mathbf{u}_0,t) = \Theta(h^{\sigma_s+1})$.*

Proof Consider an evaluation time t such that $t - t_k = O(h)$ with h sufficiently small. We have $w(t) = O(h^{\sigma_s})$ and $w'(t) = O(h^{\sigma_s-1})$. First assume that $t_{k-1} < t < t_k$ and $w'(t) = 0$. By Rolle's Theorem, since $w(t_{k-1}) = w(t_k) = 0$, there exists such an evaluation time t. By Theorem 1, $e_{loc}(FL,\mathbf{t}_0,\mathbf{u}_0,t) = O(h^{\sigma_s+2})$. Now assume that $w'(t) \neq 0$ and $\Phi(t)$ is not singular. By Theorem 1, $e_{loc}(FL,\mathbf{t}_0,\mathbf{u}_0,t) = \Theta(h^{\sigma_s+1})$. □

Theorem 3 (Necessary Condition for Optimal Natural Hermite Filters) *Let FL be a natural Hermite(σ) filter and let $t_e \in \mathbb{R}$ be such that $e_{loc}(FL, \mathbf{t}_0, \mathbf{u}_0, t_e) = \min_{t-t_k=O(h)}\{e_{loc}(FL, \mathbf{t}_0, \mathbf{u}_0, t)\}$. We have that, for h sufficiently small, t_e is a zero of the function*

$$\gamma(t) = \sum_{i=0}^{k} \frac{\sigma_i}{t - t_i} \tag{23}$$

Proof Assume that $t - t_k = O(h)$ and that h is sufficiently small. By Theorem 2, $w'(t_e)$ must be zero to minimize the local error. Note that $\Phi(t_i)$ is singular if $w'(t_i) = 0$ $(0 \leq i \leq k)$. Thus $t_e \notin \{t_0, \ldots, t_k\}$ and $w(t_e) \neq 0$. Since $w'(t) = w(t)\gamma(t)$, we conclude that $\gamma(t_e) = 0$. □

Theorem 4 *The function γ in Theorem 3 has exactly k zeros $s_0, \ldots, s_{k-1}$ such that $t_i < s_i < t_{i+1}$ $(0 \leq i < k)$.*

Proof We have $w'(t) = w(t)\gamma(t)$. By Rolle's Theorem, as $w(t_i) = w(t_{i+1}) = 0$, w' has a root s_i with $t_i < s_i < t_{i+1}$ and $w(s_i) \neq 0$ $(0 \leq i < k)$. Futhermore, we can verify that γ has at most k zeros. □

Theorem 5 (Optimal Evaluation Time) *Let FL be a natural Hermite(σ) filter, let $t_e \in \mathbb{R}$ be such that $e_{loc}(FL, \mathbf{t}_0, \mathbf{u}_0, t_e) = \min_{t-t_k=O(h)}\{e_{loc}(FL, \mathbf{t}_0, \mathbf{u}_0, t)\}$, let*

$$\gamma(t) = \sum_{i=0}^{k} \frac{\sigma_i}{t - t_i}, \tag{24}$$

and let $s_0, \ldots, s_{k-1}$ be the zeros of γ. Then, for h sufficiently small,

$$|(w/\lambda)(t_e)| = \min_{s \in \{s_0, \ldots, s_{k-1}\}} \{|(w/\lambda)(s)|\} \tag{25}$$

where the function λ is defined in Lemma 1.

Proof By definition, $w'(s_i) = w(s_i)\gamma(s_i) = 0$, for $i = 0, \ldots, k-1$. From Theorem 1 and Lemma 1, $e_{loc}(FL, \mathbf{t}_0, \mathbf{u}_0, s_i) \approx |(w/\lambda)(s_i)|$. □

B The Benchmarks.

- The full Brusselator (BRUS)

$$\begin{aligned} u_1' &= 1 + u_1^2 u_2 - (u_3 + 1)u_1 \\ u_2' &= u_1 u_3 - u_1^2 u_2 \\ u_3' &= -u_1 u_3 + \alpha \end{aligned} \tag{26}$$

$$\alpha = 1, [t_0, t_f] = [0, 14], u(t_0) = (1, 2, 1)$$

- The Lorentz system (LOR)

$$\begin{aligned} u_1' &= \sigma(u_2 - u_1) \\ u_2' &= -u_1 u_3 + \rho u_1 - u_2 \\ u_3' &= u_1 u_2 - \beta u_3 \end{aligned} \tag{27}$$

$$\sigma = 10, \rho = 28, \beta = 8/3, [t_0, t_f] = [0, 10], u(t_0) = (15, 15, 36)$$

- The Two-Body problem (2BP)

$$\begin{aligned} & u_1' = u_3 \\ & u_2' = u_4 \\ & u_3' = -\tfrac{u_1}{(u_1^2+u_2^2)^{3/2}} \\ & u_4' = -\tfrac{u_2}{(u_1^2+u_2^2)^{3/2}} \\ & \\ & [t_0, t_f] = [0, 20], u(t_0) = (1, 0, 0, 1) \end{aligned} \tag{28}$$

- The van der Pol equation (VDP)

$$\begin{aligned} & u_1' = u_2 \\ & u_2' = \mu(1-u_1^2)u_2 - u_1 \\ & \\ & \mu = 5, [t_0, t_f] = [0, 20], u(t_0) = (2, 0) \end{aligned} \tag{29}$$

- Molecular biology problem (BIO)

$$\begin{aligned} & u_1' = vi - vd\; u_3 u_1/(kd + u_1) - kds\; u_1 \\ & u_2' = (v_1\; u_1/(kc+u_1))((1-u_2)/(k_1+1-u_2)) - v_2\; u_2(k_2+u_2) \\ & u_3' = u_2 v_3(1-u_3)/(k_3+1-u_3) - v_4\; u_3/(k_4+u_3) \\ & \\ & vi = 0.025, vd = 0.25, kd = 0.02, kds = 0.01, \\ & k_1 = 0.005, k_2 = 0.005, k_3 = 0.005, k_4 = 0.005, \\ & v_1 = 3, v_2 = 1.5, v_3 = 1, v_4 = 0.5, kc = 0.5 \\ & [t_0, t_f] = [0, 3], u(t_0) = (0.01, 0.01, 0.01) \end{aligned} \tag{30}$$

- The Oregonator (OREG)

$$\begin{aligned} & u_1' = 77.27(u_2 + u_1(1 - 8.375 \times 10^{-6} u_1 - u_2)) \\ & u_2' = \tfrac{1}{77.27}(u_3 - (1+u_1)u_2) \\ & u_3' = 0.161(u_1 - u_3) \\ & \\ & [t_0, t_f] = [0, 15], u(t_0) = (1, 2, 3) \end{aligned} \tag{31}$$

- Grigorieff stiff problem (GRI)

$$\begin{aligned} & u_1' = -50.5u_1 + 49.5u_2 \\ & u_2' = 49.5u_1 - 50.5u_2 \\ & \\ & [t_0, t_f] = [0, 10], u(t_0) = (3, 1) \end{aligned} \tag{32}$$

- The Stiff DETEST Problem D1

$$\begin{aligned} & u_1' = 0.2(u_2 - u_1) \\ & u_2' = 10u_1 - (60 - 0.125u_3)u_2 + 0.125u_3 \\ & u_3' = 1 \\ & \\ & [t_0, t_f] = [0, 20], u(t_0) = (0, 0, 0) \end{aligned} \tag{33}$$

- The Lienard system (LIEN)

$$\begin{aligned} & u_1' = u_2 - (a_1 u_0 + a_2 u_0^2 + \ldots + a_r u_0^r) \\ & u_2' = -u_1 \\ & \\ & r = 21, a_0 = \ldots = a_r = 1, [t_0, t_f] = [0, 20], u(t_0) = (0.2, 0.1) \end{aligned} \tag{34}$$

- Problem P1

$$\begin{aligned} u_1' &= -u_2 - u_1u_2^2 + u_3^2 - u_1^3 \\ u_2' &= u_1 + u_3^3 - u_2^3 \\ u_3' &= -u_1u_3 - u_3u_1^2 - u_2u_3^2 - u_3^5 \\ \\ [t_0, t_f] &= [0, 100], u(t_0) = (0.4, 0.1, 0.2) \end{aligned} \tag{35}$$

- Problem P2

$$\begin{aligned} u_1' &= -2u_2 + u_2u_3 - u_1^3 \\ u_2' &= u_1 - u_1u_3 - u_2^3 \\ u_3' &= u_1u_2 - u_3^3 \\ \\ [t_0, t_f] &= [0, 20], u(t_0) = (1.5, 1.2, 1) \end{aligned} \tag{36}$$

- Problem P3

$$\begin{aligned} u_1' &= u_1^2u_2^2 - u_1^4 + \alpha u_1u_2^2 + \beta u_1^3 - u_2^4 \\ u_2' &= u_1u_2^3 - u_1^3u_2 + \alpha u_2^3 + \beta u_1^2u_2 \\ \\ \alpha &= -1, \beta = 1, [t_0, t_f] = [0, 50], u(t_0) = (0.2, 0.6) \end{aligned} \tag{37}$$

C Experimental Results

IVP	IHO	GHF	h	Excess			Time		
	$p\,,\ q$	σ		IHO	GHF	Ratio	IHO	GHF	Ratio
BRUS	3,3	(3,3)	1E-1	2.3E-3	1.2E-3	1.9			
			7.5E-2	4.5E-5	2.4E-5	1.9			
			5E-2	9.7E-7	4.9E-7	2.0			
			2.5E-2	5.2E-9	2.7E-9	1.9			
			1.25E-2	3.2E-11	1.7E-11	1.9			
			1E-2	6.5E-12	3.5E-12	1.9	5.1	3.9	1.3
	4,4	(4,4)	1E-1	1.7E-4	9.9E-5	1.7			
			7.5E-2	2.0E-6	1.1E-6	1.8			
			5E-2	1.0E-8	5.0E-9	2.0			
			2.5E-2	7.4E-12	3.2E-12	2.3	2.8	2.0	1.4
	5,5	(5,5)	1E-1	2.4E-5	1.6E-5	1.5			
			7.5E-2	1.2E-7	7.6E-8	1.6			
			5E-2	1.6E-10	9.4E-11	1.7	1.9	1.3	1.5
	7,7	(7,7)	1E-1	7.6E-7	5.2E-7	1.5			
			7.5E-2	6.6E-10	4.7E-10	1.4	1.9	1.3	1.5
	8,8	(8,8)	1E-1	1.5E-7	1.1E-7	1.4			
			7.5E-2	5.4E-11	4.0E-11	1.4	2.2	1.5	1.5
LOR	3,3	(3,3)	1.25E-2	4.8E-1	3.2E-1	1.5			
			1E-2	6.7E-2	4.5E-2	1.5			
			7.5E-3	7.7E-3	4.9E-3	1.6			
			5E-3	4.3E-4	2.6E-4	1.7			
			2.5E-3	3.1E-6	2.0E-6	1.6	11	8	1.4
	4,4	(4,4)	2E-2	1.5E-1	1.0E-1	1.5			
			1.75E-2	2.7E-2	1.8E-2	1.5			
			1.5E-2	5.0E-3	3.0E-3	1.7			
			1.25E-2	8.0E-4	4.6E-4	1.7			
			1E-2	9.0E-5	5.0E-5	1.8			
			7.5E-3	6.0E-6	3.1E-6	1.9	4.7	3.6	1.3
	7,7	(7,7)	3E-2	3.0E-3	2.4E-3	1.2			
			2.75E-2	4.5E-4	3.6E-4	1.2			
			2.5E-2	6.6E-5	5.3E-5	1.2			
			2.25E-2	7.7E-6	6.2E-6	1.2	3.0	2.2	1.4
2BP	3,3	(3,3)	1E-1	4.5E-3	7.6E-4	6.0			
			7.5E-2	1.1E-4	3.7E-5	3.0			
			5E-2	3.3E-6	1.2E-6	2.7			
			2.5E-2	1.5E-8	4.5E-9	3.3	3.6	2.6	1.4
	4,4	(4,4)	1.25E-1	2.9E-4	7.4E-5	3.9			
			1E-1	1.2E-5	3.0E-6	4.0			
			7.5E-2	3.4E-7	8.5E-8	4.0			
			5E-2	3.4E-9	9.2E-10	3.7	2.5	1.7	1.5
	7,7	(7,7)	1.5E-1	1.1E-6	5.6E-7	2.0			
			1.25E-1	2.3E-9	9.7E-10	2.4	2.0	1.4	1.4
VDP	3,3	(3,3)	4E-2	1.5E-2	5.8E-3	2.6			
			3E-2	5.9E-5	3.8E-5	1.6			
			2E-2	1.7E-6	9.6E-7	1.8			
			1E-2	1.0E-8	5.3E-9	1.9			
			5E-3	7.4E-11	3.8E-11	1.9			
			2.5E-3	4.7E-13	2.6E-13	1.8	14	12	1.2
	4,4	(4,4)	4E-2	4.7E-5	4.0E-5	1.2			
			3E-2	8.4E-7	5.1E-7	1.6			
			2E-2	9.0E-9	4.5E-9	2.0			
			1E-2	1.1E-11	4.7E-12	2.3	4.5	3.7	1.2
	5,5	(5,5)	4E-2	2.6E-6	2.1E-6	1.2			
			3E-2	2.3E-8	1.6E-8	1.4			
			2E-2	6.7E-11	3.9E-11	1.7	2.9	2.3	1.3
BIO	3,3	(3,3)	7.5E-3	4.6E-6	2.0E-6	2.3			
			5E-3	8.2E-9	3.4E-9	2.4			
			2.5E-3	2.2E-11	9.2E-12	2.4	7.0	5.1	1.4
	4,4	(4,4)	7.5E-3	1.3E-6	7.6E-7	1.7			
			5E-3	2.9E-10	1.3E-10	2.2			
			2.5E-3	9.7E-14	3.3E-14	2.9	10	7.1	1.4
OREG	3,3	(3,3)	1.5E-2	1.5E-4	2.2E-4	0.7			
			1E-2	8.0E-6	1.1E-5	0.7			
			7.5E-3	1.0E-6	1.4E-6	0.7			
			5E-3	6.0E-8	7.9E-8	0.8	9.6	7.6	1.3
	4,4	(4,4)	2.5E-2	2.4E-4	3.4E-4	0.7			
			2E-2	1.2E-5	1.6E-5	0.7			
			1.5E-2	6.1E-7	7.6E-7	0.8			
			1E-2	1.5E-8	1.9E-8	0.8			
			7.5E-3	1.1E-9	1.4E-9	0.8	8.2	6.4	1.3
GRI	6,6	(6,6)	4E-2	1.8E-7	2.1E-7	0.9			
			3E-2	1.8E-9	2.2E-9	0.9			
			2E-2	4.6E-12	5.4E-12	0.9			
			1E-2	3.1E-16	3.8E-16	0.8	2.4	2.0	1.2
	8,8	(8,8)	6E-2	3.5E-7	4.0E-7	0.9			
			5E-2	5.5E-9	6.2E-9	0.9			
			4E-2	5.4E-11	6.1E-11	0.9			
			3E-2	1.9E-13	2.1E-13	0.9			
			2E-2	1.2E-16	1.5E-16	0.8	1.5	1.3	1.2
D1	8,8	(8,8)	1.1E-1	1.1E-6	1.3E-6	0.8			
			1E-1	1.3E-7	1.4E-7	0.9			
			9E-2	1.5E-8	1.7E-8	0.9			
			8E-2	1.5E-9	1.7E-9	0.9			
			7E-2	1.3E-10	1.4E-10	0.9			
			6E-2	7.3E-12	8.3E-12	0.9			
			5E-2	2.8E-13	3.1E-13	0.9			
			4E-2	7.1E-15	8.3E-15	0.9	3.0	2.5	1.2

Table 4: One-Step Methods of the Same Order.

IVP	IHO	GHF	h	Excess			Time		
	$p\,,\,q$	σ		IHO	GHF	Ratio	IHO	GHF	Ratio
BRUS	3,3	(4,4)	1E-1	2.3E-3	1.0E-3	2.3			
			7.5E-2	4.5E-5	1.3E-5	3.5			
			5E-2	9.7E-7	1.2E-7	8.1			
			2.5E-2	5.2E-9	9.5E-11	55			
			1.25E-2	3.2E-11	2.0E-13	160	4.0	3.6	1.1
	4,4	(5,5)	1E-1	1.7E-4	1.0E-4	1.7			
			7.5E-2	2.0E-6	9.9E-7	2.0			
			5E-2	1.0E-8	3.2E-9	3.1			
			2.5E-2	7.4E-12	6.4E-13	12	2.8	2.4	1.2
LOR	3,3	(4,4)	1.25E-2	4.8E-1	1.3E-2	1.5			
			1E-2	6.7E-2	1.2E-3	56			
			7.5E-3	7.7E-3	5.7E-5	135			
			5E-3	4.3E-4	9.7E-7	443	5.4	4.9	1.1
	4,4	(5,5)	2E-2	1.5E-1	6.2E-2	2.4			
			1.75E-2	2.7E-2	9.0E-3	3.0			
			1.5E-2	5.0E-3	1.2E-3	4.2			
			1.25E-2	8.0E-4	1.2E-4	6.7			
			1E-2	9.0E-5	7.2E-6	13			
			7.5E-3	6.0E-6	2.6E-7	23	4.7	4.1	1.1
2BP	3,3	(4,4)	1E-1	4.5E-3	2.5E-5	180			
			7.5E-2	1.1E-4	7.6E-7	145			
			5E-2	3.3E-6	8.9E-9	371			
			2.5E-2	1.5E-8	4.1E-11	366	3.6	3.0	1.2
	4,4	(5,5)	1.25E-1	2.9E-4	1.1E-5	26			
			1E-1	1.2E-5	3.6E-7	33			
			7.5E-2	3.4E-7	5.6E-9	61			
			5E-2	3.4E-9	5.5E-11	62	2.5	2.0	1.3
VDP	3,3	(4,4)	4E-2	1.5E-2	2.5E-3	6.0			
			3E-2	5.9E-5	9.7E-6	6.1			
			2E-2	1.7E-6	8.8E-8	19			
			1E-2	1.0E-8	6.2E-11	161			
			5E-3	7.4E-11	9.0E-14	822	7.4	7.2	1.0
	4,4	(5,5)	4E-2	4.7E-5	3.6E-5	1.3			
			3E-2	8.4E-7	3.6E-7	2.3			
			2E-2	9.0E-9	1.6E-9	5.6			
			1E-2	1.1E-11	2.8E-13	39	4.5	4.2	1.1
BIO	3,3	(4,4)	7.5E-3	4.6E-6	1.7E-6	2.7			
			5E-3	8.2E-9	1.2E-9	6.8			
			2.5E-3	2.2E-11	4.8E-13	46	7.0	6.2	1.1
	4,4	(5,5)	7.5E-3	1.3E-6	7.7E-7	1.7			
			5E-3	2.9E-10	9.3E-11	3.1			
			2.5E-3	9.7E-14	1.0E-14	9.7	10	8.4	1.2
OREG	3,3	(4,4)	2E-2	2.6E-3	7.0E-5	37			
			1.5E-2	1.5E-4	1.1E-6	136			
			1E-2	8.0E-6	2.2E-8	364			
			7.5E-3	1.0E-6	1.5E-9	667			
			5E-3	6.0E-8	4.6E-11	1304	9.6	8.6	1.1
	4,4	(5,5)	2.5E-2	2.4E-4	1.4E-4	1.7			
			2E-2	1.2E-5	3.9E-6	3.1			
			1.5E-2	6.1E-7	1.6E-8	38			
			1E-2	1.5E-8	6.3E-11	238	6.2	5.3	1.2
GRI	6,6	(7,7)	5E-2	2.1E-2	4.1E-7	51220			
			4E-2	1.8E-7	4.0E-9	45			
			3E-2	1.8E-9	2.4E-11	75			
			2E-2	4.6E-12	2.7E-14	170	1.2	1.1	1.1
	8,8	(9,9)	7E-2	1.8E+5	1.9E-6	9.5E+10			
			6E-2	3.5E-7	1.0E-8	35			
			5E-2	5.5E-9	1.2E-10	46			
			4E-2	5.4E-11	7.4E-13	73			
			3E-2	1.9E-13	1.5E-15	127	1.0	0.9	1.1
D1	8,8	(9,9)	1.1E-1	1.1E-6	3.9E-8	28			
			1E-1	1.3E-7	3.6E-9	36			
			9E-2	1.5E-8	3.5E-10	43			
			8E-2	1.5E-9	2.9E-11	53			
			7E-2	1.3E-10	1.8E-12	72			
			6E-2	7.3E-12	7.8E-14	94	2.0	1.8	1.1

Table 5: One-Step Methods of Different Orders.

IVP	IHO	GHF	h	Excess			Time		
	p, q	σ		IHO	GHF	Ratio	IHO	GHF	Ratio
LIEN	3,3	(3,3)	4E-1	1.4E-8	1.1E-8	1.3			
			3E-1	8.4E-10	4.7E-9	0.2			
			2E-1	2.3E-11	5.4E-11	0.4			
			1E-1	1.3E-13	1.3E-13	1.0			
			5E-2	8.7E-16	8.8E-16	1.0	9.2	7.2	1.3
	3,3	(2,2,2)	4E-1	1.4E-8	3.0E-9	4.7			
			3E-1	8.4E-10	1.9E-9	4.4			
			2E-1	2.3E-11	6.9E-12	3.3			
			1E-1	1.3E-13	3.7E-14	3.5			
			5E-2	8.7E-16	2.6E-16	3.3	9.2	7.2	1.3
	6,6	(6,6)	5E-1	7.2E-8	6.0E-8	1.2			
			4E-1	1.7E-11	1.4E-11	1.2			
			3E-1	4.0E-14	3.3E-14	1.2	4.1	3.0	1.4
	6,6	(4,4,4)	5E-1	7.2E-8	2.0E-10	360			
			4E-1	1.7E-11	1.3E-12	13			
			3E-1	4.0E-14	3.9E-15	10	4.1	3.0	1.4
	6,6	(3,3,3,3)	5E-1	7.2E-8	3.9E-10	185			
			4E-1	1.7E-11	6.0E-13	28			
			3E-1	4.0E-14	2.3E-15	17	4.1	3.0	1.4
	9,9	(9,9)	5E-1	1.5E-8	1.3E-8	1.2			
			4E-1	9.7E-14	8.3E-14	1.2	4.8	3.2	1.5
	9,9	(6,6,6)	5E-1	1.5E-8	5.3E-12	2830			
			4E-1	9.7E-14	4.0E-15	24	4.8	3.0	1.6
P1	6,6	(6,6)	5E-1	1.9E-5	1.5E-5	1.3			
			4E-1	3.6E-7	2.1E-7	1.7			
			3E-1	1.1E-9	8.6E-10	1.3	5.7	4.1	1.4
	6,6	(4,4,4)	5E-1	1.9E-5	4.2E-6	4.5			
			4E-1	3.6E-7	5.2E-8	6.9			
			3E-1	1.1E-9	2.4E-10	4.6	5.7	4.4	1.3
	8,8	(8,8)	6E-1	2.4E-3	2.9E-3	0.8			
			5E-1	1.0E-6	8.2E-7	1.2			
			4E-1	4.6E-9	3.7E-9	1.2	6.6	4.8	1.4
	8,8	(4,4,4,4)	6E-1	2.4E-3	2.0E-4	12			
			5E-1	1.0E-6	9.1E-8	11			
			4E-1	4.6E-9	3.9E-10	12	6.6	5.5	1.2
P2	6,6	(6,6)	1E-1	1.1E-4	8.3E-5	1.3			
			9E-2	5.0E-6	3.9E-6	1.3			
			8E-2	3.7E-7	2.9E-7	1.3			
			7E-2	2.5E-8	2.0E-8	1.3			
			6E-2	1.4E-9	1.1E-9	1.3			
			5E-2	5.3E-11	4.0E-11	1.3	4.4	3.1	1.4
	6,6	(4,4,4)	1E-1	1.1E-4	1.7E-7	647			
			9E-2	5.0E-6	2.7E-8	185			
			8E-2	3.7E-7	3.8E-9	97			
			7E-2	2.5E-8	4.4E-10	57			
			6E-2	1.4E-9	4.1E-11	34			
			5E-2	5.3E-11	2.9E-12	18	4.4	3.5	1.3
	6,6	(3,3,3,3)	1E-1	1.1E-4	5.4E-8	2037			
			9E-2	5.0E-6	9.6E-9	521			
			8E-2	3.7E-7	1.5E-9	247			
			7E-2	2.5E-8	1.9E-10	132			
			6E-2	1.4E-9	2.1E-11	67			
			5E-2	5.3E-11	4.3E-12	12	4.4	4.1	1.1
	8,8	(8,8)	1E-1	1.9E-5	1.6E-5	1.2			
			9E-2	4.4E-7	3.6E-7	1.2			
			8E-2	1.7E-8	1.4E-8	1.2			
			7E-2	5.7E-10	4.6E-10	1.2			
			6E-2	1.5E-11	1.2E-11	1.2	5.6	3.9	1.4
	8,8	(4,4,4,4)	1E-1	1.9E-5	1.1E-9	17273			
			9E-2	4.4E-7	1.3E-10	3385			
			8E-2	1.7E-8	1.7E-11	1000			
			7E-2	5.7E-10	5.5E-12	104			
			6E-2	1.5E-11	4.0E-12	3.7	5.6	4.9	1.1
	9,9	(9,9)	1E-1	8.5E-6	7.0E-6	1.2			
			9E-2	1.4E-7	1.1E-7	1.3			
			8E-2	3.7E-9	3.1E-9	1.2			
			7E-2	8.9E-11	7.3E-11	1.2			
			6E-2	1.6E-12	1.4E-12	1.1	7.5	5.2	1.4
	9,9	(6,6,6)	1E-1	8.5E-6	1.8E-9	4722			
			9E-2	1.4E-7	1.5E-10	933			
			8E-2	3.7E-9	1.1E-11	336			
			7E-2	8.9E-11	1.5E-12	59			
			6E-2	1.6E-12	1.0E-12	1.6	7.5	5.7	1.3
P3	6,6	(6,6)	5E-1	2.0E-4	1.6E-4	1.3			
			4E-1	1.4E-7	1.1E-7	1.3			
			3E-1	4.2E-10	3.2E-10	1.3			
			2E-1	2.4E-13	1.8E-13	1.3	3.1	2.2	1.4
	6,6	(4,4,4)	5E-1	2.0E-4	1.1E-7	1818			
			4E-1	1.4E-7	2.1E-9	67			
			3E-1	4.2E-10	1.6E-11	26			
			2E-1	2.4E-13	2.5E-14	9.6	3.1	2.4	1.3
	6,6	(3,3,3,3)	5E-1	2.0E-4	2.9E-8	6897			
			4E-1	1.4E-7	7.1E-10	197			
			3E-1	4.2E-10	7.0E-12	60			
			2E-1	2.4E-13	3.3E-14	7.3	3.1	2.6	1.2
	8,8	(8,8)	5E-1	2.6E-5	2.1E-5	1.2			
			4E-1	5.4E-9	4.4E-9	1.2			
			3E-1	3.7E-12	3.0E-12	1.2	3.2	2.3	1.4
	8,8	(4,4,4,4)	5E-1	2.6E-5	7.6E-10	34211			
			4E-1	5.4E-9	7.9E-12	684			
			3E-1	3.7E-12	5.1E-14	73	3.2	2.5	1.3
	9,9	(9,9)	5E-1	1.0E-5	8.2E-6	1.2			
			4E-1	1.1E-9	9.2E-10	1.2			
			3E-1	3.5E-13	2.9E-13	1.2	3.9	2.7	1.4
	9,9	(6,6,6)	5E-1	1.0E-5	1.6E-9	16250			
			4E-1	1.1E-9	7.1E-12	761			
			3E-1	3.5E-13	1.4E-14	264	3.9	2.9	1.3

Table 6: Multistep Versus One-Step Methods.