

SOME PROPERTIES OF SOLUTIONS TO POLYNOMIAL SYSTEMS OF DIFFERENTIAL EQUATIONS

DAVID C. CAROTHERS , G. EDGAR PARKER,
JAMES S. SOCHACKI, PAUL G. WARNE

ABSTRACT. In [7] and [8], Parker and Sochacki considered iterative methods for computing the power series solution to $\mathbf{y}' = \mathbf{G} \circ \mathbf{y}$ where \mathbf{G} is a polynomial from \mathbb{R}^n to \mathbb{R}^n , including truncations of Picard iteration. The authors demonstrated that many ODE's may be transformed into computationally feasible polynomial problems of this type, and the methods generalize to a broad class of initial value PDE's. In this paper we show that the subset of the real analytic functions \mathcal{A} consisting of functions that are components of the solution to polynomial differential equations is a proper subset of \mathcal{A} and that it shares the field and near-field structure of \mathcal{A} , thus making it a proper sub-algebra. Consequences of the algebraic structure are investigated. Using these results we show that the Maclaurin or Taylor series can be generated algebraically for a large class of functions. This finding can be used to generate efficient numerical methods of arbitrary order (accuracy) for initial value ordinary differential equations. Examples to indicate these techniques are presented. Future advances in numerical solutions to initial value ordinary differential equations are indicated.

1. INTRODUCTION

$\mathbf{G} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is an \mathbb{R}^n -*polynomial* if each component of \mathbf{G} is a polynomial functional on \mathbb{R}^n . A *polynomial system* is an autonomous initial value system of differential equations of the form

$$\mathbf{y}' = \mathbf{G} \circ \mathbf{y}, \quad \mathbf{y}(a) = \mathbf{b}$$

for some \mathbb{R}^n -polynomial \mathbf{G} , $a \in \mathbb{R}$, and $\mathbf{b} \in \mathbb{R}^n$. Let \mathcal{A} be the set of real-analytic functions. An element f of \mathcal{A} is said to be *projectively polynomial* if f is a component of the solution to a polynomial system. We denote the collection of all projectively polynomial functions by \mathcal{P} . \mathcal{P}^a will denote those elements of \mathcal{P} with initial condition at $a \in \mathbb{R}$.

In [7] and [8], Parker and Sochacki showed that many differential equations may be re-written as polynomial systems, and also gave efficient algorithms for the numerical solution of these systems. In this paper, we further explore the properties of solutions to such systems, and we begin by classifying elements of \mathcal{P}

2000 *Mathematics Subject Classification*. 34A05, 34A25, 34A34, 65L05.

Key words and phrases. Analytic functions; inverse functions; Maclaurin polynomials; Pade expansions; Gröbner bases.

©2005 Texas State University - San Marcos.

Submitted August 26, 2004. Published April 5, 2005.

according to the dimension of the polynomial system and degree of the underlying polynomial for which they are components of the solution. We show in Theorem 1 that every element of \mathcal{P} is the solution to a system of at most second-degree. In Theorem 2, we show that there are real-analytic functions that are not elements of \mathcal{P} and establish that \mathcal{P} shares the field and near-field structure of the real-analytic functions, making it a proper subalgebra in both senses. In Theorem 3, we use Gröbner basis techniques to show that membership of a particular function in \mathcal{P} is equivalent to that function being a solution to a single higher order polynomial differential equation in a single variable. Example 5 will use this result to show how a wide range of systems of differential equations may be “de-coupled” to produce equivalent equations in a single variable.

Note that elementary facts about differential equations show that \mathcal{P} contains the functions sine, cosine, exponential, and polynomials. In [7] and [8], it was shown that \mathcal{P} is closed under addition, multiplication, and function composition. Theorem 4 below shows that local inverses of elements of \mathcal{P} are also in \mathcal{P} . Function r defined by $r(t) = \frac{1}{t}$ is the solution to

$$r' = -r^2; \quad r(1) = 1$$

and thus $r \in \mathcal{P}$. If $h(t) = t^p$ for $p \in \mathbb{R}$, we note that h and r are the solutions to the system

$$\begin{aligned} h' &= phr \\ r' &= -r^2 \\ h(1) &= 1, \quad r(1) = 1 \end{aligned}$$

so that $h \in \mathcal{P}$. Hence, \mathcal{P} includes all algebraic combinations and local inverses of the elementary functions. As an illustration of how large is the class of differential equations that may be transformed into polynomial form, Theorem 5 will demonstrate that if $f \in \mathcal{P}$ and y is a solution to the differential equation $y' = f \circ y$, then $y \in \mathcal{P}$.

Finally, in Theorem 6 we demonstrate cosets for \mathcal{P} by $*$ and \circ can intersect only in elements of \mathcal{P} except when represented by the respective identity elements. Applications that are consequences of the algebraic structure are suggested.

We use the development discussed in the last paragraph to show that we can use Cauchy’s theory for the analytic solution to an initial value ordinary differential equation with analytic generator and Cauchy products to improve on the methods presented in [7] and [8]. We give examples to show how the theory presented can be used in a numerical or symbolic environment. As a matter of fact, one can build a computing environment that does strictly polynomial arithmetic to any order of accuracy desired. The need to have Maclaurin or Taylor polynomials stored in a computer library is only necessary for the initial conditions since we can generate the Maclaurin and Taylor polynomials for the solution to an initial value ordinary differential equation with polynomial right hand side to any degree desired with one algorithm.

2. \mathcal{P} AND POLYNOMIAL DEGREE

For $f \in \mathcal{P}$, if f is a component of the solution to $\mathbf{y}' = \mathbf{G} \circ \mathbf{y}$, $\mathbf{y}(a) = \mathbf{b}$, then \mathbf{G} will be called a *generator* for f . $\mathcal{P}_{i,k} \subseteq \mathcal{P}$ will denote those elements having a polynomial generator whose domain has dimension that is less than or equal to i

and whose (polynomial) degree is less than or equal to k . \mathcal{P}_k will denote $\cup_{i \in \mathbb{N}} \mathcal{P}_{i,k}$. For $f \in \mathcal{P}$, a particular system of differential equations for which f is a component of the solution will be called a *projection* for f .

In [7] and [8] examples were given that indicate that lower degree, higher dimension projections may be more numerically robust than related lower dimension, higher degree projections. In [7], a first degree projection was shown for each polynomial, and \mathcal{P}_1 is thus dense in \mathcal{P} . The following theorem shows that every element of \mathcal{P} has a second-degree projection.

Theorem 1. $\mathcal{P} = \mathcal{P}_2$.

Proof. Suppose $f \in \mathcal{P}$, and that f is the first component of a solution to

$$\mathbf{y}' = \mathbf{G} \circ \mathbf{y} = \mathbf{G}(y_1, \dots, y_n), \quad \mathbf{y}(a) = \mathbf{b}.$$

If $\{G_1, \dots, G_n\}$ is the set of components of \mathbf{G} , let N_j denote the largest power of y_j that appears in any of the polynomials $\{G_1, \dots, G_n\}$. We make the substitution

$$v_{i_1, \dots, i_n} = y_1^{i_1} y_2^{i_2} \cdots y_n^{i_n}$$

with each $0 \leq i_j \leq N_j$ and at least one of $i_j > 0$.

Note that under this substitution $y_1 = v_{1,0,\dots,0}$, $y_2 = v_{0,1,0,\dots,0}$, and so on. Substituting into each of the original equations for y_j' yields a first degree equation in the variables $\{v_{i_0, \dots, i_n}\}$, in which the left side is $v'_{i_0, i_1, \dots, i_n}$ with only one $i_j = 1$ and the remaining $i_j = 0$.

This set of equations may be augmented with additional equations of at most second degree to form a new polynomial system by adding an additional equation for each of the remaining v_{i_1, \dots, i_n} variables that does not correspond to one of the original y_j . For these variables, we have formally

$$v'_{i_1, \dots, i_n} = \sum_{k=0}^n i_k v_{i_1, \dots, i_{k-1}, \dots, i_n} y_k'.$$

Since each y_k' may be replaced under the substitution by a linear combination of the variables $\{v_{i_0, \dots, i_n}\}$, each equation in the new system has degree at most two. f is the first component (corresponding to $v_{1,0,\dots,0}$) of a solution to this system. \square

3. CHARACTERIZATION OF \mathcal{P}

Recall that \mathcal{P}^a denotes those elements of \mathcal{P} with initial condition at $a \in \mathbb{R}$. In [7], \mathcal{P}^a is shown to be closed under addition and multiplication, and that $f \circ g \in \mathcal{P}^a$ for $g \in \mathcal{P}^a$ and $f \in \mathcal{P}^{g(a)}$. For any fixed real number α , suppose that h is defined by

$$h(t) = \frac{1}{t - \alpha}, \quad t > \alpha.$$

Then $h \in \mathcal{P}^a$ for any $a > \alpha$. This guarantees the local invertibility of any non-zero element of \mathcal{P} under multiplication. Thus \mathcal{P} shares with \mathcal{A} both its local field properties under addition and multiplication and its local near-ring properties under addition and function composition. First we show that \mathcal{P} is a proper subset of \mathcal{A} .

Theorem 2. $\mathcal{A} \neq \mathcal{P}$.

Proof. We will find an analytic function $u_1 \notin \mathcal{P}$ by specifying values for $u_1^{(j)}(0)$, $|u_1^{(j)}(0)| \leq 1$ for the series expansion of u_1 in such a way that $u_1 \notin \mathcal{P}_{m,2}$ for any m .

It is possible to choose values for $u_1^{(j)}(0)$, $0 \leq j \leq 4$, $|u_1^{(j)}(0)| \leq 1$, so that any function whose 4-th degree Maclaurin polynomial is

$$\sum_{j \leq 4} \frac{u_1^{(j)}(0)}{j!} t^j$$

is not in $\mathcal{P}_{1,2}$. To show this, note that $u_1 \in \mathcal{P}_{1,2}$ implies that there are numbers a_0, a_1, a_2 so that

$$u_1' = a_0 + a_1 u_1 + a_2 u_1^2$$

Differentiate three times:

$$\begin{aligned} u_1'' &= a_1 u_1' + 2a_2 u_1 u_1' \\ u_1''' &= a_1 u_1'' + 2a_2 (u_1 u_1'' + u_1'^2) \\ u_1^{(4)} &= a_1 u_1''' + 2a_2 (u_1 u_1''' + 3u_1' u_1'') \end{aligned}$$

Any choice of values for $u_1^{(j)}(0)$, $0 \leq j \leq 4$ yields a system of 4 (linear) equations in the unknowns a_0, a_1, a_2 . It is of course possible to make this choice so that the system of equations has no solution (e.g. $u_1^{(j)}(0) = 1$ for $j = 0, 1, 2, 3$ and $u_1^{(4)}(0) = \frac{1}{2}$). For any such function u_1 , there is no appropriate differential equation which u_1 satisfies, and we conclude $u_1 \notin \mathcal{P}_{1,2}$.

Now assume that for a particular value of K , values for $u_1^{(j)}(0)$, $0 \leq j \leq n_K$, $|u_1^{(j)}(0)| \leq 1$, have been determined so that any function whose n_K degree Maclaurin polynomial is

$$\sum_{r \leq n_K} \frac{u_1^{(r)}(0)}{r!} t^r$$

is not in $\mathcal{P}_{K,2}$. In order that $u_1 \in \mathcal{P}_{K+1,2}$, it would be necessary that there exist real analytic functions u_2, \dots, u_{K+1} and numbers a_{ijk}, b_{ij}, c_i with $0 \leq i, j, k \leq K + 1$, so that u_i , $1 \leq i \leq K + 1$ satisfy the system

$$u_i' = \sum_{j=1}^{K+1} \sum_{k=1}^{K+1} a_{ijk} u_j u_k + \sum_{j=1}^{K+1} b_{ij} u_j + c_i.$$

Evaluate u_i' at 0 to obtain

$$u_i'(0) = Q_{1,i}$$

where each $Q_{1,i}$ is a polynomial in the unknowns $\{a_{ijk}, b_{ij}, c_i, u_2(0), \dots, u_{K+1}(0)\}$ (let M represent the number of these unknowns).

Differentiate each of the u_i' , evaluate at 0, and replace each $u_i'(0)$, $2 \leq i \leq K + 1$ with the corresponding $Q_{1,i}$ to obtain

$$u_i''(0) = Q_{2,i}$$

where each $Q_{2,i}$ is a polynomial in the same set of M unknowns. Proceeding in this way, we obtain a set of polynomial equations

$$u_i^{(r)}(0) = Q_{r,i}, \quad 1 \leq r \leq n_K$$

for $1 \leq i \leq K + 1$ (note that the left side of the equations with $i = 1$ are already defined). We will consider solutions in the M unknowns to the equations whose left

side is $u_1^{(j)}(0)$. These equations determine a (possibly empty) solution set $S \subseteq \mathbb{C}^M$. Let $\dim(S)$ denote the dimension (as an algebraic variety) of S in \mathbb{C}^M . Note that $\dim(S) < M$.

Differentiate $u_1^{(n_K)}$. Evaluate at 0 and replace each $u_i^{(n_K)}(0)$, $2 \leq i \leq K + 1$ with $Q_{n_K,i}$ as before to obtain a polynomial equation in the M unknowns

$$u_1^{(n_K+1)}(0) = Q_{n_K+1,1}$$

with $u_1^{(n_K+1)}(0)$ not yet determined. For any choice of $u_1^{(n_K+1)}(0)$, the system of equations

$$u_1^{(r)}(0) = Q_{r,1}, \quad 1 \leq j \leq n_K + 1$$

determines a solution set $S^* \subseteq S$. $u_1^{(n_K+1)}(0)$ may be chosen so that $\dim(S^*) < \dim(S)$ as follows: Divide S into (a finite number) of irreducible components (as in [6]). Choose a point from each component, and find $u_1^{(n_K+1)}(0)$ with $|u_1^{(n_K+1)}(0)| \leq 1$ different from the values of $Q_{n_K+1,1}$ at these points. For this $u_1^{(n_K+1)}(0)$, the intersection of S^* with each component of S is strictly contained in each component, and by [6], Theorem 2.15, we have $\dim(S^*) < \dim(S)$.

Continue as above, adding a sufficient number of additional polynomial equations

$$u_1^{(r)}(0) = Q_{j,1}, \quad j > n_K + 1$$

where each new polynomial reduces the dimension of the solution set, to produce choices for $u_1^{(n_K+1)}(0), \dots, u_1^{(n_{(K+1)})}(0)$ so that the resulting solution set in \mathbb{C}^M is empty. Thus, for any function whose $n_{(K+1)}$ degree Maclaurin polynomial is

$$\sum_{r \leq n_{(K+1)}} \frac{u_1^{(r)}(0)}{r!} t^r$$

it is impossible that $u_1 \in \mathcal{P}_{K+1,2}$. We may proceed as above to inductively construct u_1 so that $u_1 \notin \mathcal{P}_{n,2}$ for any n , noting that $|u_1^{(r)}(0)| \leq 1$ for all r guarantees the convergence of the series. \square

Next, we provide an alternative characterization of the elements of \mathcal{P} .¹ One consequence of the following theorem is the possibility of “de-coupling” any system of differential equations that may be re-cast as a polynomial system, and writing a single differential equation for each of the original variables. (See Example 5.)

Theorem 3. $u \in \mathcal{P}$ if and only if for some n there is a polynomial Q in $n + 1$ variables so that $Q(u, u', \dots, u^{(n)}) = 0$.

Proof. (\Rightarrow) If u is projectively polynomial, then there are functions v_1, \dots, v_{n-1} and polynomials P_0, \dots, P_{n-1} so that

$$\begin{aligned} u' &= P_0(u, v_1, \dots, v_{n-1}), \\ v_j' &= P_j(u, v_1, \dots, v_{n-1}) \end{aligned}$$

for $j = 1, \dots, n - 1$. Formally differentiate both sides of the first equation to find u'' and substitute for u', \dots, v_{n-1}' in the right side to write u'' as a polynomial in terms of u, \dots, v_{n-1} . Repeat this procedure several times on the result to find $u''', \dots, u^{(n)}$, obtaining:

$$u^{(j)} = P_j^*(u, v_1, \dots, v_{n-1})$$

¹The authors thank Professor Carter Lyons for his contributions to this theorem.

for $j = 2, \dots, n$. For convenience we will rename $P_0 = P_1^*$ and let $P_0^*(u, v_1, \dots, v_{n-1}) = u$ so that $u^{(j)} = P_j^*(u, v_1, \dots, v_{n-1})$ for $j = 0, \dots, n$.

For the moment we will treat the “ u ” and “ v ” functions as strictly algebraic objects, and the above as polynomial equations in $2n + 1$ variables:

$$\{v_1, \dots, v_{n-1}, u, u^{(0)}, u', \dots, u^{(n)}\}$$

We distinguish between u and $u^{(0)}$ (as algebraic objects), and will assume this order of the variables in what follows. We seek a polynomial differential equation in the last $n + 1$ variables

$$\{u^{(0)}, u', \dots, u^{(n)}\}$$

in which the equation is satisfied by all points given by a solution to the original system of differential equations. To this end, we wish to algebraically manipulate the above set of equations to eliminate the variables $\{v_1, \dots, v_{n-1}, u\}$. Let I denote the ideal in the set of all polynomials in $\{v_1, \dots, v_{n-1}, u, u^{(0)}, u', \dots, u^{(n)}\}$ generated by the collection

$$\{u^{(j)} - P_j^*, j = 0, \dots, n\},$$

and let I_n be those elements of I that involve only the last $n + 1$ variables $\{u^{(0)}, u', \dots, u^{(n)}\}$. I_n is referred to as the the n th *elimination ideal*. The equations $\{u^{(j)} = P_j^*(u, v_1, \dots, v_{n-1}), j = 0, \dots, n\}$ are a *parameterization* for an *algebraic variety* V in \mathbb{C}^{n+1} . Here the term *variety* denotes a collection of points in \mathbb{C}^{n+1} determined by a set of polynomial equations in $\{u^{(j)}\}$. The points in \mathbb{C}^{n+1} with coordinates $u^{(j)}$ determined by allowing the *parameters* $\{v_1, \dots, v_{n-1}, u\}$ to vary in \mathbb{C}^n generate (in an alternate way) a variety for some set of polynomial equations in $\{u^{(j)}\}$, technically the smallest variety containing the points given by the parameterization, in case these points do not fill up the smallest variety containing them. In the language of algebraic geometry, we seek an *implicitization* that will provide the polynomial equations in $\{u^{(j)}\}$ alone defining V , with the parameters eliminated. Applying Gröbner basis elimination techniques with a lexicographic ordering determined by our order of the variables above, we conclude that I_n contains non-zero elements. Specifically, by Chapter 3, Section 3, Theorem 1 of [3], if \mathbf{G} is a Gröbner basis for I , then $I_n \cap \mathbf{G}$ is a Gröbner basis for I_n . The comments in [3] following this theorem describe the specifics of the algorithm for eliminating the variables $\{v_1, \dots, v_{n-1}, u\}$. A basis element $Q \in I_n \cap \mathbf{G}$ gives us the appropriate differential equation $Q(u^{(0)}, u', \dots, u^{(n)}) = 0$ (and at this point we may for convenience identify $u^{(0)}$ with u). In this way, the resulting Gröbner basis provides an equation in $\{u^{(0)}, u', \dots, u^{(n)}\}$ that includes in its solution set any solution to the original (parameterized) equations $\{u^{(j)} = P_j^*(u, v_1, \dots, v_{n-1})\}$

To prove (\Leftarrow), suppose that u is a solution to

$$Q(u, u', \dots, u^{(n)}) = 0$$

for some polynomial Q . We introduce new variables by letting

$$v_j = u^{(j)}$$

for $j = 0, \dots, n$, noting that $u = v_0$. The above polynomial equation becomes

$$Q(v_0, v_1, \dots, v_n) = 0.$$

Note that for each i , $\frac{\partial Q}{\partial v_i}$ is a polynomial in the collection $\{v_j\}$, as are the second partials. We introduce one additional variable w corresponding to $1/\frac{\partial Q}{\partial v_n}$. The

desired polynomial system is then

$$\begin{aligned}
 v'_j &= v_{j+1}, \quad j = 0, \dots, n-1 \\
 v'_n &= -w \sum_{j=0}^{n-1} \frac{\partial Q}{\partial v_j} v_{j+1} \\
 w' &= -w^2 \left(\sum_{j=0}^{n-1} \frac{\partial^2 Q}{\partial v_j \partial v_n} v_{j+1} + \frac{\partial^2 Q}{\partial v_n^2} \left(-w \sum_{j=0}^{n-1} \frac{\partial Q}{\partial v_j} v_{j+1} \right) \right).
 \end{aligned}$$

□

The following theorem shows that \mathcal{P} is a local sub-near-field of \mathcal{A} by showing that local inverses under composition of elements of \mathcal{P} are elements of \mathcal{P} .

Theorem 4. *Suppose $f \in \mathcal{P}^a$, $f'(a) \neq 0$ and f^{-1} is a local inverse for f in a neighborhood of a . Then $f^{-1} \in \mathcal{P}^{f(a)}$.*

Proof. Let $f \in \mathcal{P}^a$ with local inverse f^{-1} in a neighborhood of a . There is an open interval U over which f has a non-zero derivative. Since $f \in \mathcal{P}^a$, we have $f \in \mathcal{P}_{n,2}^a$ for some n by Theorem 1. For convenience, we let $f_1 = f$, and note that there are functions f_i , $2 \leq i \leq n$ and numbers a_{ijk} , b_{ij} , c_i with $1 \leq i, j, k \leq n$, so that

$$f'_i = \sum_{j=1}^n \sum_{k=1}^n a_{ijk} f_j f_k + \sum_{j=1}^n b_{ij} f_j + c_i.$$

Note that if we differentiate the above and replace every resulting occurrence of f'_j , f'_k in the right side with a corresponding polynomial in f_j , f_k , we may note that f''_i is a polynomial in the original functions. If t is an element of the range of f restricted to U , then

$$f' \circ f^{-1}(t)(f^{-1})'(t) = 1$$

or equivalently

$$(f^{-1})'(t) = \frac{1}{f' \circ f^{-1}(t)}.$$

Also note that

$$\left(\frac{1}{f' \circ f^{-1}} \right)' = - \left(\frac{1}{f' \circ f^{-1}} \right)^2 ((f'' \circ f^{-1})(f^{-1})') = - \left(\frac{1}{f' \circ f^{-1}} \right)^3 (f'' \circ f^{-1}).$$

In the last expression, it is possible to replace $(f'' \circ f^{-1})$ with a polynomial in the collection $\{f_i \circ f^{-1}\}$.

We now consider a system of polynomial equations in f_1^{-1} , $\frac{1}{f' \circ f_1^{-1}}$, and $\{f_i \circ f^{-1}, 1 \leq i \leq n\}$. From the above, we see that $(f^{-1})'$ and $\frac{1}{f' \circ f_1^{-1}}$ may be expressed as polynomials in this collection of functions. In addition,

$$(f_i \circ f^{-1})' = (f'_i \circ f^{-1})(f^{-1})'$$

may be expressed as a polynomial in this collection of functions by replacing f'_i with $\sum_{j=1}^n \sum_{k=1}^n a_{ijk} f_j f_k + \sum_{j=1}^n b_{ij} f_j + c_i$ □

The above theorem guarantees that \mathcal{P} , like \mathcal{A} , is a local near-field for $+$ and \circ . In Section 5, we will see that reasonably tractable projections can occur when considering certain inverse problems.

A result of Cauchy ([2], p. 169, Corollary 2) shows that if G is real-analytic then the solution to

$$y' = G \circ y$$

must also be real-analytic on some open interval containing its initial condition. Note that a computation similar to that in the proof of the preceding theorem shows that if f is real-analytic, f' is not 0 on an open interval and g is the local inverse of f then

$$f'(t) = \frac{1}{g'(f(t))}.$$

That is, the converse of Cauchy’s theorem is true. ²

From classical analysis if f is a real-analytic function, y_0 is interior to interval of convergence of f , and y is a solution to

$$y' = f \circ y; \quad y(0) = y_0$$

then y is also real analytic. The following theorem shows that an analogous statement is also true with “real analytic” replaced by “projectively polynomial,” demonstrating something of the range of non-polynomial differential equations that may be transformed into polynomial systems.

Theorem 5. *Suppose that $f \in \mathcal{P}^{y_0}$. If y is a solution to*

$$y' = f \circ y; \quad y(0) = y_0$$

then $y \in \mathcal{P}^0$.

Proof. Since $f \in \mathcal{P}^{y_0}$ we may assume there are n \mathbb{R}^n -polynomials Q_j so that that f is the first component (setting $f_1 = f$) of the solution to the system

$$f'_j = Q_j(f_1, \dots, f_n); \quad f_j(y_0) = b_j, \quad j = 1, \dots, n.$$

Thus

$$\begin{aligned} (f_j \circ y)' &= (f'_j \circ y)y' \\ &= (Q_j(f_1, \dots, f_n) \circ y)(f_1 \circ y) \\ &= Q_j(f_1 \circ y, \dots, f_n \circ y)(f_1 \circ y). \end{aligned}$$

The above along with the original equation $y' = f_1 \circ y$ provides a polynomial system for which the components of the solution are y and $f_j \circ y$, $j = 1, \dots, n$. Hence $y \in \mathcal{P}^0$. □

4. THE INTERSECTING COSET PROPERTY

The previous section establishes that \mathcal{P} is a proper sub-algebra of \mathcal{A} both as a local field by $+$ and $*$ and as a local near-field by $+$ and \circ . In this section we examine consequences of the fact that $\mathcal{P} - \{0\}$ under $*$ and the non-constant elements of \mathcal{P} under \circ are local groups.

Theorem 6. *Suppose that $f \in C^1$ is defined in a neighborhood of $a \in \mathbb{R}$ and that $\mathcal{P}^{f(a)} \circ f \cap f * \mathcal{P}^a \neq \{kf | k \in \mathbb{R}\}$. Then $f \in \mathcal{P}$.*

²The authors thank Professor Peter Lax for his insight on this observation.

Proof. Since constant functions are elements of \mathcal{P} , suppose that f is not a constant function and that $g \in \mathcal{P}^{f(a)}$ and $h \in \mathcal{P}^a$ satisfy the following equation

$$g \circ f = h * f,$$

with h not a constant function. Since $g \in \mathcal{P}^{f(a)}$, there are functions g_j , $2 \leq j \leq n$ defined in a neighborhood of $f(a)$ (and we let $g_1 = g$ for convenience) so that each g'_i may be written as a polynomial expression in the collection $\{g_j\}$. Since $h \in \mathcal{P}^a$, we may similarly find a collection h_j , $2 \leq j \leq m$ defined in a neighborhood of a (letting $h = h_1$) satisfying a similar system of polynomial differential equations.

We restrict attention to an interval containing a that is in the common domain for f and the collections $\{g_j \circ f\}$ and $\{h_j\}$. Because $g \circ f = h * f$, we have

$$(g' \circ f)f' = h'f + hf'.$$

If $g' \circ f - h$ is not identically zero, let U be an open interval on which $g' \circ f - h$ is non-zero. Then for $t \in U$ we have

$$f'(t) = \frac{h'(t)f(t)}{g' \circ f(t) - h(t)}.$$

Consider a system of polynomial equations involving f , $\frac{1}{g' \circ f - h}$, $\{g_j \circ f, 1 \leq j \leq n\}$, and $\{h_j, 1 \leq j \leq m\}$. The above shows that

$$f' = h'_1 f \frac{1}{g' \circ f - h}$$

and we may express f' as a polynomial in this collection by replacing h'_1 with the corresponding polynomial in the h_j . Also

$$\left(\frac{1}{g' \circ f - h}\right)' = -\left(\frac{1}{g' \circ f - h}\right)^2(((g'' \circ f) * f') - h').$$

Since $g' = g'_1$, g' may be written as a polynomial in the collection $\{g_j\}$. Differentiating both sides and replacing each g'_j on the right side with a polynomial in the collection $\{g_j\}$, we see that g'' may be written as a polynomial in the collection $\{g_j\}$. $h' = h'_1$ may be written as a polynomial in the collection of $\{h_j\}$. Thus, the above equation shows that $\left(\frac{1}{g' \circ f - h}\right)'$ may be written as a polynomial in f , $\frac{1}{g' \circ f - h}$, $\{g_j \circ f | 1 \leq j \leq n\}$, and $\{h_j | 1 \leq j \leq m\}$. Finally, since

$$(g_i \circ f)' = (g'_i \circ f) * f'$$

we note again that g'_i may be replaced by a polynomial in the collection $\{g_j\}$, so that g'_i may also be written as a polynomial in f , $\frac{1}{g' \circ f - h}$, $\{g_j \circ f, 1 \leq j \leq n\}$, and $\{h_j, 1 \leq j \leq m\}$.

The above shows that $f \in \mathcal{P}$ as long as $(g' \circ f) - h$ is not identically 0 (and note that $f \in \mathcal{P}^a$ in case the interval U contains a). Otherwise, since in this case $g' \circ f = h$ we have $f = (g')^{-1} \circ h$ and thus $f \in \mathcal{P}^a$, as long g' is not constant. But g' constant would imply that h is constant in the original relation $g \circ f = h * f$. \square

5. EXAMPLES

Example 1. Consider the question of finding a solution to the polynomial equation $\sum_{k=0}^n a_k t^k = 0$, $a_n \neq 0$. Let f be the polynomial function defined by

$$f(t) = \sum_{k=0}^n a_k t^k.$$

From the proof of Theorem 4 we conclude

$$(f^{-1})' = \frac{1}{f' \circ f^{-1}}$$

and

$$\left(\frac{1}{f' \circ f^{-1}}\right)' = -\left(\frac{1}{f' \circ f^{-1}}\right)^3 (f'' \circ f^{-1}).$$

Thus, any local inverse for f is a solution to the system

$$\begin{aligned} x' &= y \\ y' &= -y^3 \sum_{k=2}^n k(k-1)x^{k-2} \end{aligned}$$

with the particular local inverse determined by the initial conditions. If $f'(p) \neq 0$ and we use the initial conditions $x(f(p)) = 0$ and $y(f(p)) = \frac{1}{f'(p)}$, then computation of $x(0)$ solves the polynomial equation. Notice that the application of the methods of [7] to this system provides an alternative to the classical power series methods discussed in [1]. We have a qualitative guarantee associated with this computation; we will find the zero of f in the portion of the domain containing p over which f is invertible, if there is such a point, or else the second component will increase or decrease without limit. One may compare this with Newton’s method, where conditions on the geometry of the graph of the function modelling the equation are necessary to guarantee the convergence to a particular zero of the equation and failure of the iteration is no guarantee of non-existence of a zero within the component of the initial guess.

According to Theorem 1 of this paper, if we have an algorithm to generate the Maclaurin polynomial for the solution to the initial value problem

$$\mathbf{y}' = \mathbf{Q} \circ \mathbf{y}, \mathbf{y}(0) = \mathbf{a}$$

where \mathbf{Q} is linear or quadratic on \mathbb{R}^n , and $\mathbf{a} \in \mathbb{R}^n$. Then we can generate the Maclaurin polynomial for the solution to any initial value ordinary differential equation (IVODE) with a polynomial right hand side. From Theorem 1 of [7] we know that the Picard iterates contain the Maclaurin polynomial for the solution to the initial value ordinary differential equation, and the proof in that paper shows that only that part of the iterate which is the Maclaurin polynomial need be carried forward to continue the process. In what follows, we show that we can further modify the Picard iteration process so that the Picard iterates generate the terms of the Maclaurin polynomial algebraically.

We illustrate the process in \mathbb{R}^1 , taking $Q(y) = A + By + Cy^2$ and $y(0) = \alpha$. Start the Picard process with

$$P_1(t) = \alpha = M_1(t)$$

and then define

$$P_{k+1}(t) = \alpha + \int_0^t Q(P_k(s))ds = \alpha + \int_0^t (A + BP_k(s) + CP_k(s)^2)ds.$$

This integral equation is equivalent to

$$P'_{k+1}(t) = A + BP_k(t) + CP_k(t)^2 = Q(P_k(t)) \quad y(0) = \alpha.$$

In [7] we showed that

$$P_k(t) = M_k(t) + \sum_{n=k}^{2^k-1} b_n t^n,$$

where

$$M_{k+1}(t) = \sum_{n=0}^k a_n t^n$$

is the degree k Maclaurin polynomial for the solution y to the initial value ordinary differential equation. One can use the integral equation or differential equation for $P_{k+1}(t)$ to solve for a_k and b_k . Since $a_0 = \alpha$ and $a_1 = Q(\alpha)$, this only needs to be done for $k \geq 2$. Using Cauchy products this leads to

$$\begin{aligned} P'_{k+1} &= \sum_{n=1}^k n a_n t^{n-1} + \sum_{n=k+1}^{2^{k+1}-1} n b_n t^{n-1} \\ &= \sum_{n=0}^{k-1} (n+1) a_{n+1} t^n + \sum_{n=k+1}^{2^{k+1}-1} n b_n t^{n-1} \\ &= A + B \sum_{n=0}^{2^k-1} b_n t^n + C \sum_{n=0}^{2^k-1} b_n t^n \sum_{n=0}^{2^k-1} b_n t^n \\ &= A + B \sum_{n=0}^{2^k-1} b_n t^n + C \sum_{j=0}^{2^k-1} \sum_{n=0}^{2^k-1} b_j b_n t^{n+j} \\ &= A + B \sum_{n=0}^{2^k-1} b_n t^n + C \sum_{n=0}^{2(2^k-1)} d_n t^n, \end{aligned}$$

where $b_n = a_n$ for $n \leq k$ and $d_n = \sum_{j=0}^n b_j b_{n-j}$ for $0 \leq n \leq 2^k - 1$ and $d_n = \sum_{j=n-2^k+1}^{2^k-1} b_j b_{n-j}$ for $2^k \leq n \leq 2(2^k - 1)$. By equating like powers it is straightforward to show that

$$k a_k = B a_{k-1} + C \sum_{j=0}^{k-1} a_j a_{k-1-j}.$$

That is,

$$M_{k+1}(t) = M_k(t) + \frac{(B a_{k-1} + C \sum_{j=0}^{k-1} a_j a_{k-1-j})}{k} t^k.$$

That is, we can obtain the Maclaurin polynomial algebraically. This is easily extended to any polynomial ODE or system of polynomial ODE's. Therefore, for ODE's with polynomial right hand side the Maclaurin coefficients of the solution can be obtained algebraically with the same algorithm. This algorithm we call the Algebraic-Maclaurin algorithm. The Cauchy result [2, p. 169, Corollary 2] mentioned after the proof of Theorem 4 shows that this algorithm gives the analytic solution to the initial value ordinary differential equation. One can also generate a formula for the b'_n s for $n > k$ using the above results. In a future paper, we exploit this to give convergence rates and error estimates for polynomial differential equations. It is also easy to see that one can modify the above algorithm to work

for any polynomial system. In the examples below we do this in our numerical environments.

We now present three examples using the Algebraic-Maclaurin algorithm. Since this algorithm is totally algebraic, we can generate as many Maclaurin coefficients as wanted at any time step in the algorithm. Therefore, we have a method that can change the order (degree) of accuracy on the 'fly'. We compare numerical results from this method with numerical results from the fourth order Runge-Kutta method.

Example 2. Consider the initial value ordinary differential equation (I):

$$u' = -u^r; u(0) = \alpha. \quad (5.1)$$

If we let $x_1 = u$, $x_2 = u^r$ and $x_3 = x_1^{-1}$, we obtain (II):

$$\begin{aligned} x_1' &= -x_2; & x_1(0) &= \alpha \\ x_2' &= -rx_2^2x_3; & x_2(0) &= \alpha^r \\ x_3' &= x_2x_3^2; & x_3(0) &= \alpha^{-1}. \end{aligned} \quad (5.2)$$

If we let $x_4 = x_2x_3$, we obtain (III):

$$\begin{aligned} x_1' &= -x_2; & x_1(0) &= \alpha \\ x_2' &= -rx_2x_4; & x_2(0) &= \alpha^r \\ x_3' &= x_3x_4; & x_3(0) &= \alpha^{-1} \\ x_4' &= (1-r)x_4^2; & x_4(0) &= \alpha^{r-1}. \end{aligned} \quad (5.3)$$

Noticing from this that we do not need the x_3 equation, we obtain (IV):

$$\begin{aligned} x_1' &= -x_1x_4; & x_1(0) &= \alpha \\ x_4' &= (1-r)x_4^2; & x_4(0) &= \alpha^{r-1}. \end{aligned} \quad (5.4)$$

Since the solution for the original initial value ordinary differential equation can be obtained from this last IVIDE, x_1 is in $\mathcal{P}_{2,2}$. We compare the solutions to (5.1)–(5.4) using the fourth order Runge-Kutta scheme and the Algebraic-Maclaurin algorithm to generate the fourth, fifth and sixth order (degree) Maclaurin polynomials for (5.2)–(5.4). Of course, the Maclaurin polynomial for u is the same in (5.1)–(5.4). However, in a numeric environment we update in t using numerical values which leads to different numerical values for each method.

In the table below is the error of the results of a simulation with a time step of 0.0625 using fourth order Runge-Kutta on the systems (5.1)–(5.4) and the fourth, fifth and sixth order Maclaurin polynomial on (5.2)–(5.4) using the algorithm above with $r = 3/2$ and $x(0) = 1.25$. The computing times were essentially the same. (See Example 4 below where computing times are relevant.) The errors are in comparison with the exact solution given by Maple using 30 digits of accuracy.

Time	R-K on I	R-K on II	R-K on III	R-K on IV
0.125	7.03864E-08	6.44809E-07	2.6543E-07	2.36975E-08
		Maclaurin on II	Maclaurin on III	Maclaurin on IV
	Fourth Order	5.8797E-07	5.83925E-07	6.42737E-07
	Fifth Order	2.30942E-08	2.29035E-08	2.58348E-08
	Sixth Order	8.87643E-10	8.78999E-10	1.01714E-09
Time	R-K on I	R-K on II	R-K on III	R-K on IV
1.25	7.57602E-08	1.54213E-07	4.93023E-07	3.19551E-08
		Maclaurin on II	Maclaurin on III	Maclaurin on IV
	Fourth Order	1.95583E-08	1.51518E-06	8.53911E-07
	Fifth Order	1.03188E-08	7.762E-08	3.02237E-08
	Sixth Order	7.6610E-10	3.61642E-09	1.06356E-09

It is interesting to note that the fourth and fifth order Maclaurin polynomials are similar to the fourth order Runge-Kutta results, but that the sixth order Maclaurin polynomials give noticeable improvement without much labor. It is also interesting to note that the fourth order Runge-Kutta method performs best on (IV) while the same is not true for the Maclaurin polynomials. One of the reasons is that different Maclaurin polynomials are used to update the Maclaurin polynomial for u in (5.2), (5.3) and (5.4). In a future paper we analyze which polynomial projections give the 'best' numerical results.

In the next example we show a projection that gives an interesting way to obtain an analytic solution and then we use that system to generate numerical results.

Example 3. Consider (I):

$$x' = \sin x; \quad x(0) = \alpha. \quad (5.5)$$

We let $x_2 = \sin x$ and $x_3 = \cos x$ to obtain (II):

$$\begin{aligned} x' &= x_2; & x(0) &= \alpha \\ x_2' &= x_2 x_3; & x_2(0) &= \sin(\alpha) \\ x_3' &= -x_2^2; & x_3(0) &= \cos(\alpha). \end{aligned} \quad (5.6)$$

Note that in this system one does not have to use the Maclaurin polynomial for the sine function and that the above algorithm will generate the coefficients of the Maclaurin polynomial for x without knowing the one for sine. We give numerical results for these two systems using fourth order Runge-Kutta and fourth, fifth and sixth order Maclaurin polynomials, but first we show that using the polynomial system it is easy to generate the solution to the original initial value ordinary differential equation.

Using the equation for x_2' and x_3' it is seen that $x_2^2 + x_3^2 = 1$ so that $x_3' = x_3^2 - 1$ whose solution is

$$x_3 = \frac{1 - e^{2t+2B}}{1 + e^{2t+2B}}.$$

From this we obtain

$$x_2 = \frac{2e^{t+B}}{1 + e^{2t+2B}}$$

and since $x' = x_2$, we finally have

$$x = 2 \arctan e^{t+B} - 2 \arctan B + \alpha,$$

where $B = \sqrt{\frac{1-\cos(\alpha)}{1+\cos(\alpha)}}$. We use this exact solution to compare to our numerical results.

In the table below is the error of the results of a simulation with a time step of 0.0625 using fourth order Runge-Kutta on the systems (5.5)–(5.6) and the fifth order Maclaurin polynomial on (5.6) using the algorithm above for the initial condition $x(0) = \frac{31\pi}{32}$. The computing times were essentially the same. The errors are in comparison with the exact solution given by Maple using 30 digits of accuracy.

Time	Order	R-K on I	R-K on II	Maclaurin on II
0.125	4	1.443658E-09	1.35448E-09	1.21177E-09
	5			7.6911E-12
	6			6.14E-14
2	4	3.6696E-09	3.56877E-09	4.52047E-09
	5			6.74697E-11
	6			1.1585E-12

In these results it is seen that the polynomial system gives the best results and that increasing the degree improves the results significantly.

It is well documented that (rational polynomial) Pade’ approximants are valuable in approximating analytic functions. Pade’ approximants are particularly robust when approximating a wide class of singular functions. However, very little is found in the literature regarding the use of Pade approximants in the numerical solution of differential systems. This is especially true regarding algorithms that operate in a non-symbolic environment. One contributing factor to this is likely the upfront cost of generating the Taylor coefficients of the solution of the system, which are required to create the Pade’ approximants. For a given degree, the Taylor coefficients would typically be generated by symbolically differentiating the right hand side of the system multiple times, which characteristically results in computationally complex expressions that need to be evaluated numerous times throughout the solution procedure. Also, if higher degree approximations are desired, the numerical algorithm must be recoded to include the additional expressions first derived symbolically from computing higher order derivatives of the right hand side. As shown in the examples above and documented in [7] and [8], the Algebraic-Maclaurin algorithm generates the Taylor coefficients of the solution of the system in a remarkably computationally friendly manner. With this method, generating higher (or lower) degree approximations requires nothing more than a reassignment of a parameter in a running program, which as mentioned above, gives the user “on the fly” control over the order of the numerical algorithm. The Algebraic-Maclaurin method serves as an excellent vehicle to numerically generate Pade’ approximants to the solution of differential systems, and we refer to this method as the Algebraic-Maclaurin-Pade (AMP) method.

Consider the following second order nonlinear initial value problem, which can occur in physical examples. A similar equation arises in the N -body problem, which has been recently studied in [9] using the results of [7] and [8].

Example 4. (I):

$$x''(t) = x^{3/2} \quad x(0) = 1, \quad x'(0) = 2/\sqrt{5}. \tag{5.7}$$

For this equation, the explicit solution is

$$x(t) = \frac{1}{\left(1 - \frac{1}{2\sqrt{5}}t\right)^4}.$$

Note the singularity in the solution at $t = 2\sqrt{5}$. On letting $x_1 = x$, $x_2 = x'$, $x_3 = x^{1/2}$, and $x_4 = \frac{1}{x_3}$, we obtain (II):

$$\begin{aligned} x_1' &= x_2; & x_1(0) &= 1 \\ x_2' &= x_3^3; & x_2(0) &= 2/\sqrt{5} \\ x_3' &= \frac{1}{2}x_4x_2; & x_3(0) &= 1 \\ x_4' &= -\frac{1}{2}x_4^3x_2; & x_4(0) &= 1. \end{aligned} \tag{5.8}$$

We compare numerical solutions of (5.7) and (5.8) using the fourth order Runge-Kutta scheme on (5.7), and both the above outlined Algebraic-Maclaurin and AMP methods to generate various order Maclaurin and Rational polynomial approximations for (II). The fourth order Runge-Kutta approximation for (I) and the Maclaurin and Rational polynomial approximations of the first component of (II) are contrasted against the explicit solution. Of course, the Algebraic-Maclaurin approximation matches results from employing a standard Taylor/Maclaurin series numerical method. We present results for relative error and approximate CPU cost for t “close” to the singularity. If $t = 4.4721$ the exact solution gives $x(4.4721) = 2.393\dots E + 20$. The table below gives the relative error at several orders and number of steps on the interval $[0, 4.4721]$.

Method	Order	Steps	Relative Error	CPU (<i>secs</i>)
R-K	4	$1.E + 05$	5.4751E-01	3.00E-01
R-K	4	$1.E + 07$	2.0309E-07	3.02E+01
R-K	4	$1.E + 08$	2.3338E-08	3.03E+02
Algebraic-Maclaurin	4	$1.E + 05$	6.4479E-01	1.15E+00
Algebraic-Maclaurin	32	$1.E + 05$	2.4705E-06	3.58E+01
Algebraic-Maclaurin	64	$1.E + 05$	3.7209E-09	1.36E+02
AMP	16	$1.E + 03$	2.2709E-08	2.80E-01
AMP	36	$1.E + 03$	4.1217E-10	1.51E+00

Here R-K represents the fourth order Runge-Kutta method, and AMP represents results from the Algebraic-Maclaurin-Pade method. Notice the only parameter that can be changed in the fourth order Runge-Kutta method is the number of steps, but both the Maclaurin, generated by the Algebraic-Maclaurin algorithm, and the AMP method have parameters for both the order and the number of steps. In the results above we have kept the number of steps constant merely to emphasize the impact of the change of order. The results indicate that even though the AMP algorithm is computationally more expensive per step, the increased accuracy, with a considerably longer step size, results in significant improvement in overall CPU costs. It is worth noting that the AMP results above are from an algorithm that uses full Gaussian elimination with scaled partial pivoting to generate Pade’ coefficients from known Maclaurin coefficients. There will be a further significant reduction in CPU costs when a numerically stable algorithm which uses a continued fraction technique is developed and replaces the less efficient full Gaussian elimination

portion of the AMP algorithm. We could also have posed this IODE in $P_{9,2}$ by making the projection $x_5 = x_3^2, x_6 = x_4^2, x_7 = x_2x_4, x_8 = x_2x_6$ and $x_9 = x_3x_5$.

Example 5. The proof of Theorem 3 provides an algorithm for “de-coupling” any system of differential equations that may be recast as a polynomial system. From Theorem 5 and previous examples, this includes a very wide class of systems of differential equations constructed from elementary functions. E.g. consider the system in x and y

$$\begin{aligned} x' &= y + e^{xy} \\ y' &= x + 2e^{xy}. \end{aligned}$$

To rewrite this in polynomial form, we introduce a new variable u , identifying u with e^{xy} . The system then becomes

$$\begin{aligned} x' &= y + u \\ y' &= x + 2u \\ u' &= u(x^2 + 2xu + y^2 + yu) \end{aligned}$$

We compute x'' and x''' in terms of x, y , and u as in the proof of Theorem 3, and write corresponding polynomials $x^{(j)} - P_j^*(x, y, u)$ for $j = 0, 1, 2, 3$.

Using the polynomial computer algebra system SINGULAR ([5]), we apply elimination techniques to obtain the appropriate Gröbner basis polynomial for the following single (polynomial) differential equation in x alone:

$$\begin{aligned} &4x^7x' + x^6x'^2 + 4x^5x'^3 + 2x^4x'^4 - 4x^3x'^5 + x^2x'^6 - 4xx'^7 + 5x^7x'' \\ &- 4x^6x'x'' + 9x^5x'^2x'' - 4x^4x'^3x'' + 3x^3x'^4x'' + 4x^2x'^5x'' - xx'^6x'' \\ &+ 4x'^7x'' - 5x^6x''^2 - 10x^4x'^2x''^2 - 5x^2x'^4x''^2 - 20x^5x' + 16x^4x'^2 \\ &+ 26x^3x'^3 - 12x^2x'^4 - 18xx'^5 + 4x'^6 - 20x^5x'' + 69x^4x'x'' + 4x^3x'^2x'' \\ &- 74x^2x'^3x'' + 24xx'^4x'' + 17x'^5x'' + 60x^4x''^2 - 72x^3x'x''^2 - 16x^2x'^2x''^2 \\ &+ 40xx'^3x''^2 - 12x'^4x''^2 - 40x^3x''^3 + 18x^2x'x''^3 - 8xx'^2x''^3 + 2x'^3x''^3 \\ &+ 4x^5x''' + 6x^3x'^2x''' - 4x^2x'^3x''' + 2xx'^4x''' - 4x'^5x''' + x^4x''x''' \\ &+ 4x^3x'x''x''' + 4xx'^3x''x''' - x'^4x''x''' + 32x^3x' - 28x^2x'^2 - 20xx'^3 \\ &+ 17x'^4 + 20x^3x'' - 48x^2x'x'' + 18xx'^2x'' + 4x'^3x'' - 19x^2x''^2 + 4xx'x''^2 \\ &+ 14x'^2x''^2 - 2xx''^3 - 8x'x''^3 + x''^4 - 16x^3x''' + 24x^2x'x''' + 8xx'^2x''' \\ &- 18x'^3x''' + 32x^2x''x''' - 30xx'x''x''' + 12x'^2x''x''' + 4xx''^2x''' \\ &- 2x'x''^2x''' + 4x^2x'''^2 - 4xx'x'''^2 + x'^2x'''^2 - 16xx' + 16x'^2 - 4x'x'' \\ &+ 16xx''' - 16x'x''' + 4x''x''' = 0 \end{aligned}$$

In an upcoming paper we consider error estimates, convergence rates, and other implementation issues.

Acknowledgments. The authors want to thank Professor Darin Stephenson for his helpful conversations

REFERENCES

- [1] Apostol, Tom M. *Calculating Higher Derivatives of Inverses*, The American Mathematical Monthly, 107 : 8 (2000), 738-741.
- [2] Birkhoff, Garrett and Rota, Gian-Carlo. *Ordinary Differential Equations, 2nd ed.*, (1969) Blaisdell Publishing Co., Waltham, MA.
- [3] Cox, David; Little, John; and O'Shea, Donal. *Ideals, Varieties, and Algorithms, 2nd ed.* (1996) Springer-Verlag, New York.
- [4] Gerald, Curtis F. and Wheatley, Patrick O. *Applied Numerical Analysis, 6th ed.*, (1999) Addison-Wesley-Longman, Inc., Reading, MA.
- [5] Greuel, G.-M.; Pfister, G.; and Schönemann, H. SINGULAR 2.0. *A Computer Algebra System for Polynomial Computations*. Centre for Computer Algebra, University of Kaiserslautern (2001). <http://www.singular.uni-kl.de>.
- [6] Kendig, Keith. *Elementary Algebraic Geometry* (1977) Springer-Verlag, New York.
- [7] Parker, G. Edgar, and Sochacki, James S. *Implementing the Picard Iteration*, Neural, Parallel, and Scientific Computation, 4 (1996), 97-112.
- [8] Parker, G. Edgar, and Sochacki, James S. *A Picard-McLaurin Theorem for Initial Value PDE's*, Abstract and Applied Analysis 5 : 1 (2000), 47-63.
- [9] Pruett, C.D.; Rudmin, J.W.; and Lacy, J.M. *An adaptive N-body algorithm of optimal order*, Journal of Computational Physics, 187 (2003), pp. 298-317.

DEPARTMENT OF MATHEMATICS AND STATISTICS, JAMES MADISON UNIVERSITY, HARRISONBURG,
VA 22807, USA

E-mail address: carothdc@jmu.edu

E-mail address: parkerge@jmu.edu

E-mail address: jim@math.jmu.edu

E-mail address: warnepg@math.jmu.edu