

# Solving Inverse Problems for ODEs Using the Picard Contraction Mapping

H.E. Kunze<sup>1</sup> and E.R. Vrscay  
Department of Applied Mathematics  
Faculty of Mathematics  
University of Waterloo  
Waterloo, Ontario, Canada N2L 3G1

(January 29, 1999)

**Abstract:** We consider the use of Banach’s Fixed Point Theorem for contraction maps to solve the following class of inverse problems for ordinary differential equations: Given a function  $x(t)$  (which may be an interpolation of a set of experimental data points  $(x_i, t_i)$ ), find an ODE  $\dot{x}(t) = f(x, t)$  that admits  $x(t)$  as an exact or approximate solution, where  $f$  is restricted to a class of functional forms, e.g. affine, quadratic. Borrowing from “fractal-based” methods of approximation, the method described here represents a systematic procedure for finding optimal vector fields  $f$  such that the associated contractive Picard operators  $T$  map the target solution  $x$  as close as possible to itself.

**Key Words:** inverse problem, ordinary differential equations, contraction mapping, collage theorem

## 1 Introduction

In this paper we formulate a class of inverse problems for ordinary differential equations and provide a mathematical basis for “solving” them within the framework of Banach’s Fixed Point Theorem for contraction mappings. The motivation for our treatment comes from the use of contraction maps in “fractal-based” approximation methods such as fractal interpolation [1] and fractal image compression [6, 8]. The essence of these methods, including the technique presented here, is *the approximation of elements of a complete metric space by fixed points of contractive operators on that space*, phrased as follows [7]:

Let  $(Y, d_Y)$  denote a complete metric space and  $Con(Y)$  the set of contraction maps on  $Y$ . Now let  $y \in Y$  be the “target” element we wish to approximate. Given an  $\epsilon > 0$  can we find a mapping  $T_\epsilon \in Con(Y)$  with fixed point  $\bar{y}_\epsilon \in Y$  such that  $d_Y(y, \bar{y}_\epsilon) < \epsilon$ ?

In fractal image compression, the mapping  $T_\epsilon$  is used to represent the target image  $y$ , often requiring much less computer storage than  $y$ . The approximation  $\bar{y}_\epsilon$  to  $y$  is then generated by iteration: Pick a  $y_0 \in Y$  (e.g. a blank screen) and form the iteration sequence  $y_{n+1} = T_\epsilon y_n$  so that  $d_Y(y_n, \bar{y}_\epsilon) \rightarrow 0$  as  $n \rightarrow \infty$ .

---

<sup>1</sup>**Present address:** Department of Mathematics and Statistics, University of Guelph, Guelph, Ontario, Canada N1G 2W1.

Here we consider this approximation method to study inverse problems for ODEs such as the following:

Given a solution curve  $x(t)$  for  $t \in [0, 1]$ , find an ODE  $\dot{x} = f(x, t)$  that admits  $x(t)$  as a solution as closely as desired, where  $f$  may be restricted to a prescribed class of functions, e.g. affine, quadratic in  $x$ .

Note the phrase “as closely as desired”. In many cases, differentiation of  $x(t)$  followed by some manipulation will lead to the ODE it satisfies. However, we wish to consider the more general cases for which an *exact* solution is improbable, for example:

1. when  $x(t)$  is not given in closed form but rather in the form of data points  $(x_i, t_i)$  which may then be interpolated by a smooth function,
2. when it is desired to restrict  $f(x, t)$  to a specific class of functions, e.g. polynomial in  $x$  and/or  $t$ , possibly only of first or second degree.

Both of these situations may arise when mechanical or biological systems are modelled by ODEs having particular functional forms. Experimental data points may then be used to estimate the various parameters (e.g. natural frequency, damping) that define the ODE.

Solutions to initial value problems are fixed points of contractive Picard integral operators. In this framework, the inverse problem becomes one of finding optimal vector fields that define Picard operators whose fixed point solutions  $\bar{x}(t)$  lie close to the target solutions  $x(t)$ . The end result, an algorithm that determines these vector fields by a kind of “least squares” polynomial fitting procedure, is incredibly simple both in concept and in form. Nevertheless, we emphasize that our methods are rigorously justified within the mathematical framework of Banach’s fixed point theorem along with several important corollaries that may not be so well known, at least in terms of applications. Indeed, we present these results in hope of stimulating research into the use of these methods for more complicated inverse problems.

The layout of this paper is as follows. In Section 2, we review the celebrated contraction mapping theorem along with the important corollaries that provide the basis for fractal image compression. In Section 3, we consider initial value problems for ODEs and the associated integral Picard operators, briefly reviewing the conditions that guarantee the contractivity of these operators on an appropriate space of functions with respect to the *sup* norm. We then show how the classical continuity properties of solutions with respect to (1) initial conditions and (2) the vector fields follows from the contractivity of the Picard operator. This permits a reformulation of the inverse problem in terms of finding an optimal contractive Picard operator  $T$  such that the “collage distance”  $d_Y(y, Ty)$  is minimized. A formal solution to the inverse problem follows since arbitrarily small collage distances may be achieved with the use of polynomial approximations to the vector fields  $f(x, t)$ .

Note that from a practical point of view, the use of polynomial vector fields is not a restriction since many models employed in biomathematics (population, disease), chemistry (reaction kinetics) and physics (damped, forced anharmonic oscillators) are formulated in terms of systems of polynomial ODEs.

We show that the Picard integral operators are also contractive with respect to the  $\mathcal{L}^2$  metric. This yields a convenient “least-squares”-type algorithm for the inverse problem. If we assume that the unknown vector fields are polynomial, then the minimization of the squared  $\mathcal{L}^2$  collage distance yields a set of linear equations in the unknown polynomial expansion coefficients. Some simple examples in one and two dimensions are then presented. In special cases where the target solutions are known in closed form, the optimal vector fields may be obtained algebraically with the use of a computer algebra package. In general, however, the target solutions are known as discrete sets of data points so that some kind of numerical interpolation must be performed. As expected, the accuracy with which the discretized target solution is approximated affects the accuracy of the vector field yielded by the inverse problem algorithm.

In Section 4, various strategies to improve the numerical accuracy of our algorithm are explored, including the partitioning of both time and space intervals. In applications, it is quite probable that a number of data curves or sets will be available, all of which are to be considered as solution curves of a single ODE or system of ODEs. As shown in Section 5, our algorithm is easily applied to such “multiple data sets”. The algorithm is also shown to be quite successful in determining vector fields for “partial data sets,” e.g. portions of a solution curve  $(x(t), y(t))$  that is suspected to be a periodic orbit of a 2D system of ODEs.

Finally, in Section 6, we briefly consider the application of our algorithm to noisy data sets.

## 2 Contraction Mappings and the “Collage Theorem”

In this section we briefly recall the main ideas behind fractal-based approximation, beginning with Banach’s celebrated fixed point theorem. Unless otherwise stated,  $(Y, d_Y)$  denotes a complete metric space. As well,  $Con(Y)$  denotes the set of contraction maps on  $Y$ , that is,

$$Con(Y) = \{T : Y \rightarrow Y \mid d_Y(Ty_1, Ty_2) \leq c_T d_Y(y_1, y_2), \forall y_1, y_2 \in Y, \text{ for some } c_T \in [0, 1)\}.$$

**Theorem 1** (*Banach*) *For each  $T \in Con(Y)$  there exists a unique  $\bar{y} \in Y$  such that  $T\bar{y} = \bar{y}$ . Furthermore  $d_Y(T^n y, \bar{y}) \rightarrow 0$  as  $n \rightarrow \infty$  for any  $y \in Y$ .*

A consequence of Banach’s Fixed Point Theorem is the following result, often referred to as the “Collage Theorem” in the fractals literature [2, 1]. We include a very simple proof only to emphasize its significance to the inverse problems studied here.

**Theorem 2** *Let  $y \in Y$  and  $T \in Con(Y)$  with contraction factor  $c_T \in [0, 1)$  and fixed point  $\bar{y} \in Y$ . Then*

$$d_Y(y, \bar{y}) \leq \frac{1}{1 - c_T} d_Y(y, Ty).$$

**Proof:**

$$\begin{aligned} d_Y(y, \bar{y}) &\leq d_Y(y, Ty) + d_Y(Ty, T\bar{y}) \\ &\leq d_Y(y, Ty) + c_T d_Y(y, \bar{y}). \end{aligned}$$

A rearrangement yields the desired result. ■

In fractal-based applications, the operator  $T$  typically produces a union of shrunken copies of  $y$ , i.e. a tiling or “collage” of  $y$  with distorted copies of itself. Therefore the term  $d_Y(y, Ty)$  is commonly referred to as the “collage distance.” The Collage Theorem leads to the following reformulation of the inverse problem stated in Section 1:

Let  $y \in Y$  be the “target” element we wish to approximate. Given a  $\delta > 0$  find a mapping  $T_\delta \in \text{Con}(Y)$  such that  $d_Y(y, T_\delta y) < \delta$ .

In other words, instead of searching for contraction maps whose fixed points lie close to  $y$ , we search for contraction maps that send  $y$  close to itself. Fractal-based methods of approximation then focus on the minimization of the collage distance  $d_Y(y, Ty)$ , where  $(Y, d_Y)$  is typically the space  $\mathcal{L}^2$  of square integrable functions on  $[0, 1]^2$  with associated metric (the root mean square (RMS) distance in the engineering literature). Although not often explicitly stated, these methods are based on the following continuity property of contractive operators and their associated fixed points [3].

**Theorem 3** *Let  $(Y, d_Y)$  be a compact metric space and define the following metric on  $\text{Con}(Y)$ : For  $T_1, T_2 \in \text{Con}(Y)$ ,*

$$d_{\text{Con}(Y)}(T_1, T_2) = \sup_{y \in Y} d_Y(T_1 y, T_2 y).$$

*If  $\bar{y}_1$  and  $\bar{y}_2$  denote the respective fixed points of  $T_1$  and  $T_2$ , then*

$$d_Y(\bar{y}_1, \bar{y}_2) \leq \frac{1}{1 - \min(c_1, c_2)} d_{\text{Con}(Y)}(T_1, T_2),$$

*where  $c_1$  and  $c_2$  denote the respective contraction factors of  $T_1$  and  $T_2$ .*

As expected intuitively, the “closer” the operators  $T_1$  and  $T_2$  are, the closer their respective fixed points are expected to be.

### 3 The Picard Operator and the Inverse Problem for ODEs

#### 3.1 The Picard Operator as a Contraction Mapping

We first review the existence and uniqueness theorem for solutions of ODEs in terms of contraction mappings. For a more detailed treatment, the reader is referred to [4].

Consider the following initial value problem

$$\frac{dx}{dt} = f(x, t), \quad x(t_0) = x_0, \tag{1}$$

where  $x \in \mathbb{R}$  and  $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  is, for the moment, continuous. (The extension to systems of ODEs, where  $x \in \mathbb{R}^n$  and  $f : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ , is straightforward.) A solution to this problem satisfies the equivalent integral equation

$$x(t) = x_0 + \int_{t_0}^t f(x(s), s) ds. \tag{2}$$

The Picard operator  $T$  associated with Eq. (1) is defined as

$$v(t) = (Tu)(t) = x_0 + \int_{t_0}^t f(u(s), s) ds. \quad (3)$$

Let  $I = [t_0 - a, t_0 + a]$  for  $a > 0$  and  $\mathcal{C}(I)$  be the Banach space of continuous functions  $x(t)$  on  $I$  with norm  $\|x\|_\infty = \max_{t \in I} |x(t)|$ . Then  $T : \mathcal{C}(I) \rightarrow \mathcal{C}(I)$ . Furthermore, a solution  $x(t)$  to Eq. (1) is a fixed point of  $T$ , i.e.  $Tx = x$ . As is well known, and as we briefly summarize below, the solution to Eq. (1) is unique under additional assumptions on  $f$ .

In what follows, without loss of generality, we let  $x_0 = t_0 = 0$  so that  $I = [-a, a]$ . (Nonzero values may be accommodated by appropriate shifting and scaling of parameters introduced below.) Define  $D = \{(x, t) \mid |x| \leq b, |t| \leq a\}$  and  $\bar{\mathcal{C}}(I) = \{u \in \mathcal{C}(I) \mid \|u\|_\infty \leq b\}$ . Also assume that

1.  $\max_{(x,t) \in D} |f(x, t)| < b/a$  and
2.  $f(x, t)$  satisfies the following Lipschitz condition on  $D$ :

$$|f(x_1, t) - f(x_2, t)| \leq K|x_1 - x_2|, \quad \forall (x_i, t) \in D,$$

such that  $c = Ka < 1$ .

Now define a metric on  $\bar{\mathcal{C}}(I)$  in the usual way, i.e.

$$\begin{aligned} d_\infty(u, v) &= \|u - v\|_\infty \\ &= \sup_{t \in I} |u(t) - v(t)|, \quad \forall u, v \in \bar{\mathcal{C}}(I). \end{aligned} \quad (4)$$

Then  $(\bar{\mathcal{C}}(I), d_\infty)$  is a complete metric space. By construction  $T$  maps  $\bar{\mathcal{C}}(I)$  to itself and [4]

$$d_\infty(Tu, Tv) \leq cd_\infty(u, v), \quad \forall u, v \in \bar{\mathcal{C}}(I). \quad (5)$$

The contractivity of  $T$  implies the existence of a unique element  $\bar{x} \in \bar{\mathcal{C}}(I)$  such that  $\bar{x} = T\bar{x}$ , i.e. the unique solution to the IVP in Eq. (1).

Now let  $S = [-\delta, \delta]$ , for some  $0 < \delta \ll 1$  and redefine  $D$  as  $D = \{(x, t) \mid |x| \leq b + \delta, |t| \leq a\}$ . Let  $\mathcal{F}(D)$  denote the set of all functions  $f(x, t)$  on  $D$  that satisfy properties 1 and 2 above and define

$$\|f_1 - f_2\|_{\mathcal{F}(D)} = \max_{(x,t) \in D} |f_1(x, t) - f_2(x, t)|. \quad (6)$$

We let  $\Pi(I)$  denote the set of all Picard operators  $T : \mathcal{C}(I) \rightarrow \mathcal{C}(I)$  having the form in Eq. (3) where  $x_0 \in S$  and  $f \in \mathcal{F}(D)$ . Eq. (5) is satisfied by all  $T \in \Pi(I)$ . The following result establishes the continuity of fixed points  $\bar{x}$  of operators  $T \in \Pi(I)$  - hence solutions to the initial value problem in Eq. (1) - with respect to the initial condition  $x_0$  and the vector field  $f(x, t)$ .

**Proposition 1** *Let  $T_1, T_2 \in \Pi(I)$ , as defined above, i.e. for  $u \in \bar{\mathcal{C}}(I)$ ,*

$$(T_i u)(t) = x_i + \int_0^t f_i(u(s), s) ds, \quad t \in I, \quad i = 1, 2. \quad (7)$$

Also let  $\bar{u}_i(t)$  denote the fixed point functions of the  $T_i$ . Then

$$d_\infty(\bar{u}_1, \bar{u}_2) \leq \frac{1}{1-c} \left[ |x_1 - x_2| + a \|f_1 - f_2\|_{\mathcal{F}(D)} \right], \quad (8)$$

where  $c = Ka < 1$ .

**Proof:** We examine the distance between the contractive operators  $T_1$  and  $T_2$  over  $\bar{\mathcal{C}}(I)$  as defined in Theorem 3:

$$\sup_{u \in \bar{\mathcal{C}}(I)} \|T_1 u - T_2 u\|_\infty \leq |x_1 - x_2| + \sup_{u \in \bar{\mathcal{C}}(I)} \left\| \int_0^t [f_1(u(s), s) - f_2(u(s), s)] ds \right\|_\infty. \quad (9)$$

However,

$$\begin{aligned} \sup_{u \in \bar{\mathcal{C}}(I)} \left\| \int_0^t [f_1(u(s), s) - f_2(u(s), s)] ds \right\|_\infty &\leq \sup_{u \in \bar{\mathcal{C}}(I)} \max_{t \in I} \left| \int_0^t [f_1(u(s), s) - f_2(u(s), s)] ds \right| \\ &\leq \sup_{u \in \bar{\mathcal{C}}(I)} \max_{t \in I} \left| \int_0^t |f_1(u(s), s) - f_2(u(s), s)| ds \right| \\ &\leq a \|f_1 - f_2\|_{\mathcal{F}(D)} \end{aligned} \quad (10)$$

The desired result now follows from Theorem 3. ■

Intuitively, closeness of the vector fields  $f_1, f_2$  and initial conditions  $x_1, x_2$  ensures closeness of the solutions  $\bar{x}_1(t), \bar{x}_2(t)$  to the corresponding IVPs. The above result is a recasting of the classical results on continuous dependence (see, for example, [4]) in terms of Picard contractive maps.

From a computational point of view, however, it is not convenient to work with the  $d_\infty$  metric. As in other applications, including fractal-based compression, it will be more convenient to work with the  $\mathcal{L}^2$  metric. Note that

1.  $\bar{\mathcal{C}}(I) \subset \mathcal{C}(I) \subset \mathcal{L}^2(I)$  and
2.  $T : \bar{\mathcal{C}}(I) \rightarrow \bar{\mathcal{C}}(I)$ .

The following result establishes that the operator  $T \in \Pi(I)$  is also contractive in the  $\mathcal{L}^2$  metric, which we denote by  $d_2$ .

**Proposition 2** *Let  $T \in \Pi(I)$  as defined in the previous section. Then*

$$d_2(Tu, Tv) \leq \frac{c}{\sqrt{2}} d_2(u, v), \quad \forall u, v \in \bar{\mathcal{C}}(I), \quad (11)$$

where  $c = Ka < 1$  and  $d_2(u, v) = \|u - v\|_2$ .

**Proof:** For  $u, v \in \bar{\mathcal{C}}(I) \subset \mathcal{L}^2(I)$ ,

$$\begin{aligned} \|Tu - Tv\|_2^2 &= \int_I \left[ \int_0^t (f(u(s), s) - f(v(s), s)) ds \right]^2 dt \\ &\leq \int_I \left[ \int_0^t |f(u(s), s) - f(v(s), s)| ds \right]^2 dt \\ &\leq K^2 \int_I \left[ \int_0^t |u(s) - v(s)| ds \right]^2 dt \end{aligned} \quad (12)$$

For  $t > 0$ , and from the Cauchy-Schwartz inequality,

$$\begin{aligned} \int_0^t |u(s) - v(s)| ds &\leq \left[ \int_0^t ds \right]^{1/2} \left[ \int_0^t |u(s) - v(s)|^2 ds \right]^{1/2} \\ &\leq t^{1/2} \left[ \int_0^t |u(s) - v(s)|^2 ds \right]^{1/2}. \end{aligned}$$

A similar result follows for  $t < 0$ . Note that

$$\begin{aligned} \int_0^a \left[ \int_0^t |u(s) - v(s)| ds \right]^2 dt &\leq \int_0^a \int_0^t t |u(s) - v(s)|^2 ds dt \\ &= \int_0^a \int_s^a t |u(s) - v(s)|^2 dt ds \\ &= \frac{1}{2} \int_0^a (a^2 - s^2) |u(s) - v(s)|^2 ds \\ &\leq \frac{a^2}{2} \int_0^a |u(s) - v(s)|^2 ds. \end{aligned}$$

A similar result is obtained for the integral over  $[-a, 0]$ . Substitution into Eq. (12) completes the proof. ■

Note that the space  $\bar{\mathcal{C}}(I)$  is not complete with respect to the  $\mathcal{L}^2$  metric  $d_2$ . This is not a problem, however, since the fixed point  $\bar{x}(t)$  of  $T$  lies in  $\bar{\mathcal{C}}(I)$ . The Picard operator  $T \in \Pi(I)$  is also contractive in  $\mathcal{L}^1$  metric. The following result is obtained by a change in the order of integration as above.

**Proposition 3** *Let  $T \in \Pi(I)$  as defined in the previous section. Then*

$$d_1(Tu, Tv) \leq cd_1(u, v), \quad \forall u, v \in \bar{\mathcal{C}}(I) \ (\subset \mathcal{L}^1(I)), \quad (13)$$

where  $c = Ka < 1$  and  $d_1(u, v) = \|u - v\|_1$ .

### 3.2 Inverse Problem for ODE's via Collage Theorem

We now present a more formal definition of the inverse problem for ODEs:

Let  $x(t)$  be a solution to the IVP  $\dot{x}(t) = f(x, t)$ ,  $x(t_0) = x_0$  for  $t \in I$ , where  $I$  is some interval centered at  $x_0$ . Given an  $\epsilon > 0$ , find a vector field  $g(x, t)$  (subject to appropriate conditions) such that the (unique) solution  $y(t)$  to the IVP  $\dot{x}(t) = g(x, t)$ ,  $x(t_0) = x_0$  satisfies  $\|x - y\| < \epsilon$ .

Intuitively, the vector field  $g(x, t)$  must be “close” to  $f(x, t)$ . However, the latter is generally *unknown*. The Collage Theorem provides a systematic method of finding such approximations from a knowledge of the target solution  $x(t)$ : Find a vector field  $g(x, t)$  with associated Picard operator  $T$  such that the collage distance  $d_Y(x, Tx)$  is as small as desired.

As mentioned earlier, in practical applications, we consider the  $\mathcal{L}^2$  collage distance  $\|x - Tx\|_2$ . The minimization of the squared  $\mathcal{L}^2$  collage distance conveniently becomes a

least-squares problem in the parameters that define  $f$ , hence  $T$ . In principle, the use of polynomial approximations to vector fields is sufficient for a formal solution to the following inverse problem for ODEs. (For simplicity of notation, we consider only the one-dimensional case below: The extension to systems in  $\mathbb{R}^n$  is straightforward.)

**Theorem 4** *Suppose that  $x(t)$  is a solution to the initial value value problem*

$$\dot{x} = f(x), \quad x(0) = 0, \quad (14)$$

for  $t \in I = [-a, a]$  and  $f \in \mathcal{F}(D)$ , defined earlier. Then given an  $\epsilon > 0$ , there exists an interval  $\bar{I} \subseteq I$  and a Picard operator  $T_\epsilon \in \Pi(\bar{I})$  such that  $\|x - T_\epsilon x\|_\infty < \epsilon$  where the norm is computed over  $\bar{I}$ ,

**Proof:** From the Weierstrass approximation theorem, for any  $\eta > 0$ , there exists a polynomial  $P_N(x)$  such that  $\|P_N - f(x)\|_\infty < \eta$ . Define a subinterval  $\bar{I} \subseteq I$ ,  $\bar{I} = [-\bar{a}, \bar{a}]$ , such that  $c_N = K_N \bar{a} < 1/2$ , where  $K_N$  is the Lipschitz constant of  $P_N$  on  $\bar{I}$ . (The value  $1/2$  above is chosen without loss of generality.) Now let  $T_N$  be the Picard operator associated with  $P_N(x)$  and  $x_0 = 0$ . By construction,  $T_N$  is contractive on  $\bar{C}(\bar{I})$  with contraction factor  $c_N$ . Let  $\bar{x}_N \in C(\bar{I})$  denote the fixed point of  $T_N$ . From Proposition 1,

$$\|x - \bar{x}_N\|_\infty \leq \frac{\bar{a}\eta}{1 - c_N} < 2\bar{a}\eta.$$

Then

$$\begin{aligned} \|x - T_N x\|_\infty &\leq \|x - \bar{x}_N\|_\infty + \|\bar{x}_N - T_N x\|_\infty \\ &= \|x - \bar{x}_N\|_\infty + \|T_N \bar{x}_N - T_N x\|_\infty \\ &\leq (1 + c_N) \|x - \bar{x}_N\|_\infty \\ &< 2 \|x - \bar{x}_N\|_\infty. \end{aligned}$$

Since  $|\bar{a}| < |a|$ , given an  $\epsilon > 0$ , there exists an  $N$  sufficiently large so that  $4\bar{a}\eta < \epsilon$ , yielding the result  $\|x - T_N x\|_\infty < \epsilon$ . We may simply rename  $T_N$  as  $T_\epsilon$ , acknowledging the dependence of  $N$  upon  $\epsilon$ , to obtain the desired result. ■

We now have, in terms of the Collage Theorem, the basis for a systematic algorithm to provide polynomial approximations to an unknown vector field  $f(x, t)$  that will admit a solution  $x(t)$  as closely as desired.

### 3.2.1 Algorithm for the Inverse Problem

For the moment, we consider only the one-dimensional case, i.e. a target solution  $x(t) \in \mathbb{R}$  and *autonomous* vector fields that are polynomial in  $x$ , i.e.

$$f(x) = \sum_{n=0}^N c_n x^n \quad (15)$$

for some  $N > 0$ . Without loss of generality, we let  $t_0 = 0$  but, for reasons to be made clear below, we leave  $x_0$  as a variable. Then

$$(Tx)(t) = x_0 + \int_0^t \left[ \sum_{k=0}^N c_k (x(s))^k \right] ds. \quad (16)$$

The squared  $\mathcal{L}^2$  collage distance is given by

$$\begin{aligned} \Delta^2 &= \int_I [x(t) - (Tx)(t)]^2 dt \\ &= \int_I \left[ x(t) - x_0 - \sum_{k=0}^N c_k g_k(t) \right]^2 dt. \end{aligned} \quad (17)$$

where

$$g_k(t) = \int_0^t (x(s))^k ds, \quad k = 0, 1, \dots \quad (18)$$

(Clearly,  $g_0(t) = t$ .)  $\Delta^2$  is a quadratic form in the parameters  $c_k$ ,  $0 \leq k \leq N$ , as well as  $x_0$ .

We now minimize  $\Delta^2$  with respect to the variational parameters  $c_k$ ,  $0 \leq k \leq N$ , and possibly  $x_0$  as well. In other words, we may not necessarily impose the condition that  $x_0 = x(0)$ . (One justification is that there may be errors associated with the data  $x(t)$ .) The stationarity conditions  $\partial\Delta^2/\partial x_0 = 0$  and  $\partial\Delta^2/\partial c_k = 0$  yield the following set of linear equations:

$$\begin{bmatrix} 1 & \langle g_0 \rangle & \langle g_1 \rangle & \dots & \langle g_N \rangle \\ \langle g_0 \rangle & \langle g_0 g_0 \rangle & \langle g_0 g_1 \rangle & \dots & \langle g_0 g_N \rangle \\ \langle g_1 \rangle & \langle g_1 g_0 \rangle & \langle g_1 g_1 \rangle & \dots & \langle g_1 g_N \rangle \\ \dots & \dots & \dots & \dots & \dots \\ \langle g_N \rangle & \langle g_N g_0 \rangle & \langle g_N g_1 \rangle & \dots & \langle g_N g_N \rangle \end{bmatrix} \begin{bmatrix} x_0 \\ c_0 \\ c_1 \\ \dots \\ c_N \end{bmatrix} = \begin{bmatrix} \langle x \rangle \\ \langle x g_0 \rangle \\ \langle x g_1 \rangle \\ \dots \\ \langle x g_N \rangle \end{bmatrix} \quad (19)$$

where  $\langle f \rangle = \int_I f(t) dt$ . If  $x_0$  is not considered to be a variational parameter then the first row and column of the above matrix (as well as the first element of each column vector) are simply removed. It is not in general guaranteed that the matrix of this linear system is nonsingular. For example, if  $x(t) = C$ , the  $g_k(t) = C^k t$  and for  $i > 1$  the  $i^{\text{th}}$  column of the matrix in Eq. (19) is  $C^{i-2} t$  times the first column; the matrix has rank two. In such situations, however, the collage distance can trivially be made equal to zero.

From the theorem of the previous section, the collage distance  $\Delta$  in Eq. (17) may be made as small as desired by making  $N$  sufficiently large. These ideas extend naturally to higher dimensions. Let  $x = (x_1, \dots, x_n)$  and  $f = (f_1, \dots, f_n)$ . The  $i^{\text{th}}$  component of the  $\mathcal{L}^2$  collage distance is

$$\Delta_i^2 = \int_I \left[ x_i(t) - x_i(0) - \int_0^t f_i(x_1(s), \dots, x_n(s)) ds \right]^2 dt.$$

Presupposing a particular form for the vector field components  $f_i$  determines the set of variational parameters;  $x_i(0)$  may be included. Imposing the usual stationarity conditions will yield a system of equations for these parameters. If  $f_i$  is assumed to be polynomial in  $x_j$ ,  $j = 1, \dots, n$ , the process yields a linear system similar to Eq. (19).

### 3.2.2 Practical Considerations: “Collaging” an Approximate Solution

In practical applications, the target solution  $x(t)$  may not be known in exact or closed form but rather in the form of data points, e.g.  $(x_i, t_i) = x(t_i)$ ,  $1 \leq i \leq n$  in one dimension,  $(x_i, y_i) = (x(t_i), y(t_i))$  in two dimensions. One may perform some kind of smooth interpolation or optimal fitting of the data points to produce an approximate target solution which shall be denoted as  $\tilde{x}(t)$ . In this paper, we consider approximations having the following form,

$$\tilde{x}(t) = \sum_{l=0}^p a_l \phi_l(t),$$

where the  $\{\phi_l\}$  comprise a suitable basis. As we show below, the most convenient form of approximation is the best  $\mathcal{L}^2$  or “least squares” polynomial approximation, i.e.  $\phi_l(t) = t^l$ .

Given a target solution  $x(t)$ , its approximation  $\tilde{x}(t)$  and a Picard operator  $T$ , the collage distance satisfies the inequality

$$\begin{aligned} \|x - Tx\| &\leq \|x - \tilde{x}\| + \|\tilde{x} - T\tilde{x}\| + \|T\tilde{x} - Tx\| \\ &\leq (1 + c) \|x - \tilde{x}\| + \|\tilde{x} - T\tilde{x}\|. \end{aligned} \quad (20)$$

(The norm and corresponding metric are unspecified for the moment.) Let us define:

1.  $\delta_1 = \|x - \tilde{x}\|$ , the error in approximation of  $x(t)$  by  $\tilde{x}$ ,
2.  $\delta_2 = \|\tilde{x} - T\tilde{x}\|$ , the collage distance of  $\tilde{x}$ .

Each of these terms is independent of the other. Once a satisfactory approximation  $\tilde{x}(t)$  is constructed, we then apply the algorithm of Section 3.2.1 to it, seeking to find an optimal Picard operator for which the collage distance  $\delta_2$  is sufficiently small. The condition  $(1 + c)\delta_1 + \delta_2 < \epsilon$  guarantees that the “true” collage distance satisfies  $\|x - Tx\| < \epsilon$ .

### 3.2.3 Some One-Dimensional Examples

We first apply the inverse algorithm to cases where the exact solution  $x(t)$  is known in closed form. This permits all integrals, etc. in Eq. (19) to be calculated exactly. The computer algebra system *Maple V* was used in all the calculations reported below.

**Example 1:** Let  $x(t) = Ae^{Bt} + C$  be the target solution, where  $A, B, C \in \mathbb{R}$ . This is the exact solution of the linear ODE

$$\frac{dx}{dt} = -BC + Bx$$

for which  $A$  plays the role of the arbitrary constant. If we choose  $N = 1$  in Eq. (5), i.e. a linear vector field, then the solution of the linear system in (19) is given by

$$x_0 = A + C, \quad c_0 = -BC, \quad c_1 = B,$$

which agrees with the above ODE. (This solution, obtained analytically using *Maple V*, is independent of the choice of the interval  $I$ , as well as  $N$ .)

**Example 2:** Let  $x(t) = t^2$  be the target solution on the half-interval  $I = [0, 1]$ . If we choose  $N = 1$  and  $x_0$  variable, the solution to the  $3 \times 3$  system in Eq. (19) defines the IVP:

$$\frac{dx}{dt} = \frac{5}{12} + \frac{35}{18}x, \quad x(0) = -\frac{1}{27},$$

with corresponding (minimized) collage distance  $\|x - Tx\|_2 = 0.0124$ . The solution to this IVP is

$$\bar{x}(t) = \frac{67}{378} \exp\left(\frac{35}{18}t\right) - \frac{3}{14}.$$

Note that  $\bar{x}(0) \neq x(0)$ . The  $\mathcal{L}^2$  distance between the two functions is  $\|x - \bar{x}\|_2 = 0.0123$ .

If, however, we impose the condition that  $x_0 = x(0) = 0$ , then the solution to the  $2 \times 2$  system in Eq. (19) yields the IVP

$$\frac{dx}{dt} = \frac{5}{12} + \frac{35}{16}x, \quad x(0) = 0,$$

with corresponding collage distance  $\|x - Tx\|_2 = 0.0186$  and solution

$$\bar{x}(t) = \frac{1}{7} \exp\left(\frac{35}{16}t\right) - \frac{1}{7}.$$

As expected, the distance  $\|x - \bar{x}\|_2 = 0.0463$  is larger than in the previous case where  $x_0$  was not constrained.

Setting  $N = 2$ , i.e. allowing  $f$  to be quadratic, and solving Eq. (19) leads to IVPs with smaller collage distances, as one would expect. Table 1 summarizes some results for this example. As expected, the error  $\|x - \bar{x}\|_2$  decreases as  $N$  increases.

	$f$	$x_0$	$\ x - Tx\ _2$	$\ x - \bar{x}\ _2$
$f$ linear, $x_0$ constrained	$\frac{5}{16} + \frac{35}{16}x$	0	0.0186	0.0463
$f$ linear, $x_0$ variable	$\frac{5}{12} + \frac{35}{18}x$	$-\frac{1}{27}$	0.0124	0.0123
$f$ quadratic, $x_0$ constrained	$\frac{105}{512} + \frac{945}{256}x - \frac{1155}{512}x^2$	0	0.0070	0.0300
$f$ quadratic, $x_0$ variable	$\frac{35}{128} + \frac{105}{32}x - \frac{231}{128}x^2$	$-\frac{1}{60}$	0.0047	0.0049

Table 1: Inverse problem results for Example 2,  $x(t) = t^2$ .

**Example 3:** Consider  $x(t) = \frac{-125}{8}t^4 + \frac{1225}{36}t^3 - \frac{625}{24}t^2 + \frac{25}{3}t$ , with  $I = [0, 1]$ , and repeat the calculations of the previous example. This quartic has a local maximum at  $t = \frac{1}{3}$  and  $t = \frac{4}{5}$  and a local minimum  $t = \frac{1}{2}$ . The coefficients were chosen to scale  $x(t)$  for graphing on  $[0, 1]^2$ . Results are summarized in Table 2; in the quadratic  $f$  case, decimal coefficients are presented here to avoid writing the cumbersome rational expressions.

In this example, the two measures of distance over  $[0, 1]$ ,  $\|x - Tx\|_2$  and  $\|x - \bar{x}\|_2$ , appear to face impassable lower bounds. Increasing  $N$  (increasing the degree of  $f$ ) does not shrink either distance to zero, at least for moderate values of  $N$ . Graphically, all four cases look similar. Figure 1 presents two graphs to illustrate the two distance measures for the case  $f$  quadratic and  $x_0$  variable. It is the nonmonotonicity of  $x(t)$  that causes difficulty. In Section 4, we tackle this example with a modified strategy that allows the collage distance (and hence the actual  $\mathcal{L}^2$  error) to be made arbitrarily small.

	$f$	IC	$\ x - Tx\ _2$	$\ x - \bar{x}\ _2$
$f$ linear $x_0$ constrained	$\frac{158165}{15844} - \frac{40887}{3961}x$	0	0.0608	0.0504
$f$ linear $x_0$ variable	$\frac{192815}{18444} - \frac{16632}{1537}x$	$-\frac{8425}{221328}$	0.0604	0.0497
$f$ quadratic $x_0$ constrained	$9.5235 - 8.6938x - 1.2020x^2$	0	0.0607	0.0497
$f$ quadratic $x_0$ variable	$11.0343 - 12.4563x + 1.0744x^2$	-0.0518	0.0603	0.0501

Table 2: Inverse problem results for Example 3,  $x(t) = \frac{-125}{8}t^4 + \frac{1225}{36}t^3 - \frac{625}{24}t^2 + \frac{25}{3}t$ .

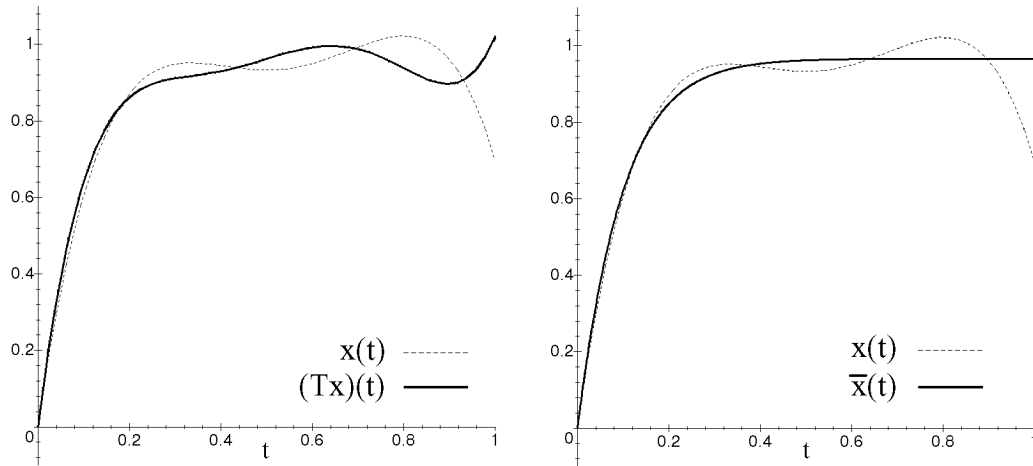


Figure 1: Graphical results for Example 3,  $x(t) = \frac{-125}{8}t^4 + \frac{1225}{36}t^3 - \frac{625}{24}t^2 + \frac{25}{3}t$ .

### 3.2.4 Examples in Two Dimensions

Given the parametric representation of a curve  $x = x(t)$ ,  $y = y(t)$ ,  $t \geq 0$ , we look for a two-dimensional system of ODEs of the form

$$\dot{x}(t) = f(x, y), \quad x(0) = x_0, \quad (21)$$

$$\dot{y}(t) = g(x, y), \quad y(0) = y_0, \quad (22)$$

with conditions on  $f$  and  $g$  to be specified below.

**Example 4:** Applying this method to the parabola  $x(t) = t$  and  $y(t) = t^2$  with  $f$  and  $g$  restricted to be linear functions of  $x$  and  $y$ , we obtain, as expected, the system

$$\dot{x}(t) = 1, \quad x(0) = 0$$

$$\dot{y}(t) = 2x, \quad y(0) = 0.$$

When  $f$  and  $g$  are allowed to be at most quadratic in  $x$  and  $y$ , we obtain the system

$$\dot{x}(t) = 1 + c_1(x^2 - y), \quad x(0) = 0,$$

$$\dot{y}(t) = 2x + c_2(x^2 - y), \quad y(0) = 0,$$

where  $c_1$  and  $c_2$  are arbitrary constants.

**Example 5:** For the target solution  $x(t) = \cos(t)$  and  $y(t) = \sin(t)$  (unit circle), with  $f$  and  $g$  at most quadratic in  $x$  and  $y$ , we obtain

$$\begin{aligned}\dot{x}(t) &= -y + c_1(x^2 + y^2 - 1), \quad x(0) = 1, \\ \dot{y}(t) &= x + c_2(x^2 + y^2 - 1), \quad y(0) = 0,\end{aligned}$$

where  $c_1$  and  $c_2$  are arbitrary constants. (These constants disappear when  $f$  and  $g$  are constrained to be at most linear in  $x$  and  $y$ .)

We now consider target solutions that are not known in closed form. Instead, they exist in the form of data series  $(x_i, y_i)$  from which are constructed approximate target solutions  $(\tilde{x}(t), \tilde{y}(t))$ . In the following two examples, the data series are obtained by numerical integration of a known two-dimensional system of ODEs.

**Example 6:** We first examine the Lotka-Volterra system

$$\dot{x}(t) = x - 2xy, \quad x(0) = \frac{1}{3}, \tag{23}$$

$$\dot{y}(t) = 2xy - y, \quad y(0) = \frac{1}{3}, \tag{24}$$

the solution of which is a periodic cycle. For  $\tilde{x}(t)$  and  $\tilde{y}(t)$  polynomials of degree 30, we obtain:

$$\|x(t) - \tilde{x}(t)\|_2 \approx 0.00007 \text{ and } \|y(t) - \tilde{y}(t)\|_2 \approx 0.0001.$$

Given such a periodic cycle there are a number of possibilities that can be explored, including:

1. Restricting the functional forms of the admissible vector fields  $f(x, y)$  and  $g(x, y)$  (e.g. Lotka-Volterra form) *vs.* allowing the fields to range over a wider class of functions (e.g. quadratic higher degree polynomials). For example, an experimentalist may wish to find the “best” Lotka-Volterra system for a given data series.
2. Constraining the initial value  $(x_0, y_0)$  or allowing it to be a variable.

In Table 3 are presented numerical results for a few such possibilities applied to this example. In order to save space, we have defined  $Ex = \|\tilde{x} - T\tilde{x}\|_2$ ,  $Ey = \|\tilde{y} - T\tilde{y}\|_2$ ,  $\Delta x = \|x - \bar{x}\|_2$  and  $\Delta y = \|y - \bar{y}\|_2$ , where  $(\bar{x}, \bar{y})$  denotes the fixed point of the Picard operator  $T$ .

From the Table, it is clear that when the vector fields are constrained to be of Lotka-Volterra type (first two entries), the algorithm locates the original system in Eqs. (23,24) to many digits of accuracy. Relaxing the constraint on the initial values  $(x_0, y_0)$  yields no significant change in the approximations. In the case that the vector fields are allowed to be quadratic, the “true” Lotka-Volterra system is located to two digits accuracy.

As well, as evidenced in the examples, it should come as no surprise that increasing the accuracy of the basis representation of the target function(s) leads to a resulting DE (or system) that is closer to the original. As an example, we present Table 4, which shows the effect of poorer polynomial approximations to  $x(t)$  and  $y(t)$  in the simplest case of the earlier Lotka-Volterra example, where we seek an  $f$  and  $g$  of the correct form.

form of $f$ and $g$	$f$ and $g$	initial conditions	collage distances	actual $\mathcal{L}^2$ error
$f = c_1x + c_2xy$ $g = c_3y + c_4xy$ $x_0$ constrained $y_0$ constrained	$f = 1.0000x - 2.0000xy$ $g = -1.0000y + 2.0000xy$	$x_0 = \frac{1}{3}$ $y_0 = \frac{1}{3}$	$Ex = 0.0002$ $Ey = 0.0002$	$\Delta x = 0.000002$ $\Delta y = 0.000002$
$f = c_1x + c_2xy$ $g = c_3y + c_4xy$ $x_0$ variable $y_0$ variable	$f = 1.0000x - 2.0000xy$ $g = -1.0000y + 2.0000xy$	$x_0 = 0.3333$ $y_0 = 0.3333$	$Ex = 0.0002$ $Ey = 0.0002$	$\Delta x = 0.000002$ $\Delta y = 0.000002$
$f$ quadratic $g$ quadratic $x_0$ constrained $y_0$ constrained	$f = -0.0008 + 1.0017x$ $+ 0.0016y - 0.0017x^2$ $- 1.9999xy - 0.0015y^2$ $g = 0.0008 - 0.0018x$ $- 1.0018y + 0.0017x^2$ $+ 2.0000xy + 0.0017y^2$	$x_0 = \frac{1}{3}$ $y_0 = \frac{1}{3}$	$Ex = 0.0002$ $Ey = 0.0002$	$\Delta x = 0.000004$ $\Delta y = 0.000004$
$f$ quadratic $g$ quadratic $x_0$ variable $y_0$ variable	$f = -0.0009 + 1.0020x$ $+ 0.0019y - 0.0019x^2$ $- 2.0000xy - 0.0018y^2$ $g = 0.0008 - 0.0017x$ $- 1.0016y + 0.0016x^2$ $+ 2.0000xy + 0.0015y^2$	$x_0 = 0.3333$ $y_0 = 0.3333$	$Ex = 0.0002$ $Ey = 0.0002$	$\Delta x = 0.000004$ $\Delta y = 0.000004$

Table 3: Inverse problem results for Lotka-Volterra system in Example 6.

polynomial basis degree	$\ x(t) - \tilde{x}(t)\ _2$	$\ y(t) - \tilde{y}(t)\ _2$	$f$	$g$
10	0.0022853	0.0023162	$0.998918x - 1.997938xy$	$-0.999749y + 1.999531xy$
15	0.0001278	0.0001234	$1.000121x - 2.000197xy$	$-0.999974y + 1.999943xy$
20	0.0000162	0.0000164	$1.000022x - 2.000036xy$	$-0.999996y + 1.999987xy$
25	0.0000009	0.0000008	$0.999996x - 1.999994xy$	$-0.999998y + 1.999997xy$
30	0.0000002	0.0000002	$1.000003x - 2.000005xy$	$-0.999999y + 1.999999xy$

Table 4: Effect of different quality basis representations for the Lotka-Volterra example.

**Note:** The target functions  $x(t)$  and  $y(t)$  can also be approximated using trigonometric function series. Indeed, for the case of periodic orbits, one would expect better approximations. This is found numerically - the same accuracy of approximation is achieved with fewer trigonometric terms. However, there is a dramatic increase in computational time and memory requirements. This is due to the moment-type integrals,  $g_k(t)$ , that must be computed in Eq. (20) because of the polynomial nature of the vector fields. Our *Maple V* algorithm computes these integrals using symbolic algebra. Symbolically, it is much easier to integrate products of polynomials in  $t$  than it is to integrate products of trigonometric series. As an illustration, in Table 5 we present the CPU time and memory required to compute the results (on a 266MHz Pentium II with 128Mb RAM running *Maple V Release 4*) shown in the first (and least complex) row entry of Table 3

Basis	Computation Time (in seconds)		Memory (in Mb)	
	$f = c_1x + c_2xy$ $g = c_3x + c_4xy$	$f, g$ quadratic	$f = c_1x + c_2xy$ $g = c_3x + c_4xy$	$f, g$ quadratic
polynomial, to degree 20	1678	1850	16.2	29.7
polynomial, to degree 30	2023	2608	18.5	60.3
trigonometric, to $\cos(5t/T)$ , $\sin(5t/T)$	4047	program runs out of memory	49.3	> 99.6
trigonometric, to $\cos(10t/T)$ , $\sin(10t/T)$	program runs out of memory	program runs out of memory	> 118	> 120

Table 5: Illustration of CPU usage for various runs of the Lotka-Volterra problem.

**Example 7:** The van der Pol equation

$$\dot{x}(t) = y, \quad x(0) = 2, \quad (25)$$

$$\dot{y}(t) = -x - 0.8(x^2 - 1)y, \quad y(0) = 0, \quad (26)$$

has a periodic cycle as solution. Using 46th-degree polynomials to approximate the cycle, we find

$$\|x(t) - \tilde{x}(t)\|_2 \approx 0.0141 \text{ and } \|y(t) - \tilde{y}(t)\|_2 \approx 0.0894.$$

The results for this example are given in Table 6.

### 3.3 Improving the Numerical Accuracy by Partitioning

As shown in Table 4, the accuracy achieved by numerical methods increases with the accuracy to which the target solutions  $x(t)$  and  $y(t)$  are approximated in terms of basis expansions. A significant additional increase in accuracy may be achieved if the inverse problem is partitioned in the time domain so that the target solutions are better approximated over smaller time intervals.

First, divide the time interval  $I$  into  $m$  subintervals,  $I_k = [t_{k-1}, t_k]$ ,  $k = 1, \dots, m$ . The squared  $\mathcal{L}^2$  collage distance in Eq. (17) then becomes a sum of squared collage distances over the subintervals  $I_k$ :

$$\Delta^2 = \sum_{k=1}^m \Delta_k^2.$$

All integrals in Eq. (19) also become partitioned into sums of component integrals over the  $I_k$ . Over each subinterval  $I_k$ , assume a separate expansion of the target solutions, which we write formally as

$$\tilde{x}(t) = \sum_{l=0}^{p_k} c_{k,l} \phi_l(t), \quad t \in I_k, \quad 1 \leq k \leq m,$$

assuming, once again, that the  $\{\phi_k\}$  comprise a suitable basis, typically  $\phi_l(t) = t^l$ .

form of $f$ and $g$	$f$ and $g$	initial conditions	collage distances	actual $\mathcal{L}^2$ error
$f = c_1y$ $g = c_2x + c_3y + c_4x^2y$ $x_0$ constrained $y_0$ constrained	$f = 1.0001y$ $g = -1.0007x + 0.8003y$ $-0.8012x^2y$	$x_0 = 2$ $y_0 = 0$	$Ex = 0.0110$ $Ey = 0.0546$	$\Delta x = 0.0011$ $\Delta y = 0.0016$
$f = c_1y$ $g = c_2x + c_3y + c_4x^2y$ $x_0$ variable $y_0$ variable	$f = 1.0000y$ $g = -1.0007x + 0.7993y$ $-0.7999x^2y$	$x_0 = 2.0000$ $y_0 = 0.0023$	$Ex = 0.0002$ $Ey = 0.0002$	$\Delta x = 0.0012$ $\Delta y = 0.0018$
$f$ quadratic $g$ quadratic+ $c_1x^2y$ $x_0$ constrained $y_0$ constrained	$f = 0.000176 - 0.000014x$ $+1.000002y - 0.000052x^2$ $+0.000038xy - 0.000041y^2$ $g = -0.000968 - 0.999941x$ $+0.799944y + 0.000277x^2$ $-0.000179xy + 0.000220y^2$ $-0.799920x^2y$	$x_0 = 2$ $y_0 = 0$	$Ex = 0.000427$ $Ey = 0.000427$	$\Delta x = 0.000261$ $\Delta y = 0.002394$
$f$ quadratic $g$ quadratic+ $c_1x^2y$ $x_0$ variable $y_0$ variable	$f = -0.000004 + 0.000004x$ $+1.000002y + 0.000007x^2$ $+0.000002xy - 0.000003y^2$ $g = -0.000591 - 0.999979x$ $+0.799921y + 0.000151x^2$ $-0.000105xy + 0.000142y^2$ $-0.799898x^2y$	$x_0 = 1.999853$ $y_0 = 0.000323$	$Ex = 0.000459$ $Ey = 0.000459$	$\Delta x = 0.000351$ $\Delta y = 0.002447$

Table 6: Inverse problem results for van der Pol system in Example 7.

Note that in this method, we employ the *same* vector field  $f(x)$ , hence the same unknown coefficients  $c_k$ , cf. Eq. (16), over all subintervals  $I_k$ . Minimization of the squared  $\mathcal{L}^2$  collage distance will lead to a linear system of equations in the  $c_k$  having the form in Eq. (19).

From a practical programming standpoint it is simplest to partition at the midpoints of subintervals, letting  $t_i = [t_0 + 2^{1-m}(t_m - t_0)(i - 1), t_0 + 2^{1-m}(t_m - t_0)i]$ ,  $i = 0, \dots, m$ . We also consider partitioning at the turning points of the target function(s), i.e. those points  $t_k$  at which one of the component solutions  $x(t)$ ,  $y(t)$ , etc., has a local maximum or minimum. In higher dimensions, the integral in the Picard operator is split into component integrals at the union of the sets of turning points of the component functions.

To illustrate the gains of this approach, we consider once again the Lotka-Volterra system from Example 6, employing the technique outlined above. We find polynomial basis representations of  $x(t)$  and  $y(t)$ , in each case partitioning at the turning points of the function under consideration. The turning points are approximated by stepping through the numerical data. The results are presented in Table 7, which can be compared to those of Table 4.

polynomial basis degree	$\ x(t) - \tilde{x}(t)\ _2$	$\ y(t) - \tilde{y}(t)\ _2$	$f$	$g$
10	0.0000029210	0.0000057568	$0.999906x - 1.999606xy$	$-0.999968y + 1.999874xy$
15	0.0000000087	0.0000000176	$0.999998x - 1.999991xy$	$-1.000000y + 2.000001xy$
20	0.0000000008	0.0000000002	$1.000003x - 2.000007xy$	$-0.999998y + 1.999996xy$
25	0.0000000002	0.0000000002	$1.000003x - 2.000006xy$	$-0.999996y + 1.999991xy$
30	0.0000000002	0.0000000003	$1.000003x - 2.000005xy$	$-0.999999y + 2.000001xy$

Table 7: Results for the Lotka-Volterra problem with partitioning of the basis representations of  $x(t)$  and  $y(t)$ .

## 4 Improving the “Collaging” by Partitioning: Changing the Vector Field

The accuracy of numerical solutions may also be increased by partitioning the space and/or time domains, employing different approximations of the vector field over the various subdomains, followed by “patching” of the solutions and/or vector fields. Such partitioning may seem rather artificial and motivated only by numerical concerns. However, as we show below, in particular cases, the partitioning of a vector field is necessary due to the existence of equilibrium points in the target solution. As before, we consider only autonomous initial value problems, i.e.  $\dot{x} = f(x)$ ,  $x(0) = x_0$ , and employ the notation introduced in Section 3.1.

### 4.1 Partitioning the Vector Field in the Spatial Domain

One may consider the partitioning of the spatial region  $D$  into subregions  $D_i$ ,  $1 \leq i \leq m$ , and solve the inverse problem over each region  $D_i$ . Over each subregion  $D_i$ , this is equivalent to finding the vector field  $f_i$  that best approximates the true vector field  $f(x)$  supported on  $D_i$ . These problems are not independent, however, since the vector field  $f(x)$  must be continuous on  $D$ , implying “patching” conditions of the  $f_i$  at the boundaries of the regions  $D_i$ .

This procedure is analogous to that of “charting” of vector fields discussed in Ref. [5], namely, the approximation of the vector field  $f(x)$  by a continuous, piecewise polynomial vector field. (In order to satisfy continuity and differentiability conditions at the boundary points, the polynomial vector fields must be at least quadratic.) An application of the Weierstrass approximation theorem yields a result similar to Theorem 4.

In principle, this partitioning method is quite straightforward. However, since the target solutions  $x(t)$  are generally presented as time series, and the Picard operator itself involves integrations over the time variable, it is more practical to consider partitioning schemes in the time domain.

### 4.2 Partitioning in the Time Domain

In this case, as in Section 3.2.5, we partition the time interval  $I$  into  $m$  subintervals,  $I_k = [t_{k-1}, t_k]$ ,  $k = 1, \dots, m$ , and consider the following set of  $m$  inverse problems over the respective subintervals  $I_k$ : Given a target function  $x(t)$ ,  $t \in I$ , find an ODE of the form

(once again, for simplicity, we discuss only the one-dimensional case)

$$\dot{x}(t) = f_k(x), \quad t \in I_k, \quad k = 1, \dots, m, \quad x(t_0) = x_0, \quad (27)$$

with solution approximating the target  $x(t)$ .

For each subinterval  $I_k$ , the algorithm of Section 3.2 can be employed to find a Picard map  $T_k$  that minimizes the collage distance  $\|x - T_k x\|_2$  on  $I_k$ . The Picard operator  $T : C(I) \rightarrow C(I)$  is considered as a vector of operators  $T = (T_1, T_2, \dots, T_m)$  where  $T_k : C(I_k) \rightarrow C(I_k)$ ,  $1 \leq k \leq m$ . The partitioning method essentially produces a “patching” of the vector field  $f(x)$  by the vector fields  $f_k(x)$  acting over different time intervals  $I_k$ . In order to guarantee that each  $T_k$  maps  $C(I_k)$  into itself, the vector fields  $f_k(x)$  are assumed to obey Lipschitz conditions given in Section 1 over appropriate rectangular (sub)regions in  $(x, t)$  space. Note that this “patching” necessitates a number of additional conditions, implying that the  $m$  inverse problems over the subintervals  $I_k$  are *not* independent:

1. In order to guarantee that  $T : C(I) \rightarrow C(I)$ , i.e. that  $Tx$  is continuous on  $I$ , the following conditions must be satisfied:

$$T_k(t_k) = T_{k+1}(t_k), \quad k = 1, \dots, m - 1.$$

2. In order to guarantee that  $x(t)$  is differentiable at the partition points  $t_k$ , the vector field  $f(x)$  must be continuous, i.e.

$$f_k(x(t_k)) = f_{k+1}(t_k), \quad k = 1, \dots, m - 1.$$

(In some applications, where a piecewise differentiable solution  $x(t)$  is sufficient, it may be desirable to relax this constraint.)

The modified inverse problem algorithm now assumes the following form:

1. Find  $T_1$  that minimizes  $\|x - T_1 x\|_2$  on  $I_1$  (possibly imposing the constraint that  $x(0) = x_0$ ),
2. For  $k = 2, 3, \dots, m$ , find  $T_k$  that minimizes  $\|x - T_k x\|_2$  on  $I_k$  and satisfies  $T_k(t_{k-1}) = T_{k-1}(t_{k-1})$ .

Once again, a Weierstrass polynomial approximation of  $f$  over each interval  $I_k$ , etc., leads to a result similar to Theorem 4 of Section 3.2.

**Remarks:**

1. It is not necessary to begin the above algorithm with a “collaging” over the first subinterval  $I_1$ . One could begin with another subinterval, shooting forward and backward using the appropriate matching conditions. In general, depending on the choice of starting subinterval we produce a different Picard vector operator  $T$  since the optimization process in each of the subsequent subintervals must satisfy matching conditions.

2. The procedures outlined above are easily extended to higher dimensions, as before.

**Motivation for this partitioning method:** Let us return to the target solution of Example 2,  $x(t) = t^2$ , but over the interval  $I = [-1, 1]$ . This curve is a “solution curve” to the ODE

$$\dot{x} = 2\sqrt{x}.$$

Obviously, the vector field  $f(x) = 2\sqrt{x}$  does not satisfy a Lipschitz condition at  $x = 0$ , so that the initial value problem  $x(0) = 0$  has, in fact, an infinity of solutions. This technicality, however, does not preclude the determination of an optimal vector field  $f(x)$  from a prescribed class of functions, given the particular solution curve  $x(t) = t^2$ .

From a dynamical systems point of view,  $x = 0$  is an equilibrium point of the ODE so that  $x(t) = 0$ ,  $t \in \mathbb{R}$ , is a solution of the ODE. From this viewpoint, the target curve  $x(t) = t^2$  is composed of *three* solutions. However, if we wish to interpret the target curve  $x(t) = t^2$ ,  $t \in [-a, a]$ , for some  $a > 0$ , as a single dynamical solution in time, the vector field  $f(x)$  will have to undergo a change at  $t = 0$ , i.e.

$$\dot{x} = f(x) = \begin{cases} -2\sqrt{x}, & t < 0, \\ 0, & t = 0, \\ 2\sqrt{x}, & t > 0. \end{cases}$$

In the one-dimensional case, such problems are clearly encountered whenever the solution curve is nonmonotonic in time  $t$ . An alternative to such cases, which may be quite relevant to practical physical problems, is to employ nonautonomous vector fields  $f(x, t)$ .

In the examples presented below, we restrict our attention to polynomial vector fields  $f_i$  that depend only upon  $x$ , that is,

$$f_i(x) = \sum_{n=0}^N c_{i,n} x^n, \quad i = 1, \dots, m.$$

**Example 8:** We reconsider the quartic  $x(t) = \frac{-125}{8}t^4 + \frac{1225}{36}t^3 - \frac{625}{24}t^2 + \frac{25}{3}t$ , with  $I = [0, 1]$ . Tables 8 and 9 list the results achieved using various partitioning schemes for  $x_0$  constrained and variable, respectively. In each case, we look for a piecewise linear  $f$ . Although the collage distance decreases as the number of partitions increases, the actual  $\mathcal{L}^2$  error  $\|x - \bar{x}\|_2$  increases. Figure 2 illustrates collage distances and actual errors for some of the cases in Table 9. It is of minor interest to note, for example, that when partitioning  $[0, 1]$  into fourths, eighths and sixteenths, the best starting subintervals for our algorithm are, respectively,  $[\frac{1}{4}, \frac{1}{2}]$ ,  $[\frac{1}{8}, \frac{1}{4}]$  and  $[\frac{5}{8}, \frac{11}{16}]$ .

**Example 9:** We reconsider the Lotka-Volterra system of Eqs. (23)–(24), applying our modified algorithm by partitioning at the turning points of  $x(t)$  and  $y(t)$ , i.e. at  $t$ -values of 0, 0.997, 2.589, 3.850, 5.443, and 6.349. Regardless of whether the initial conditions are constrained or variable, if we seek an  $f(x, y)$  and  $g(x, y)$  of the correct form (i.e.  $f(x, y) = c_1x + c_2xy$  and  $g(x, y) = c_3y + c_4xy$ ), we find the correct system coefficients to four decimal places. The sum of the two minimal collage distances is 0.000053 and the actual  $\mathcal{L}^2$  errors are  $\|x - \bar{x}\|_2 \approx 0.000017$  and  $\|y - \bar{y}\|_2 \approx 0.000016$  in either case.

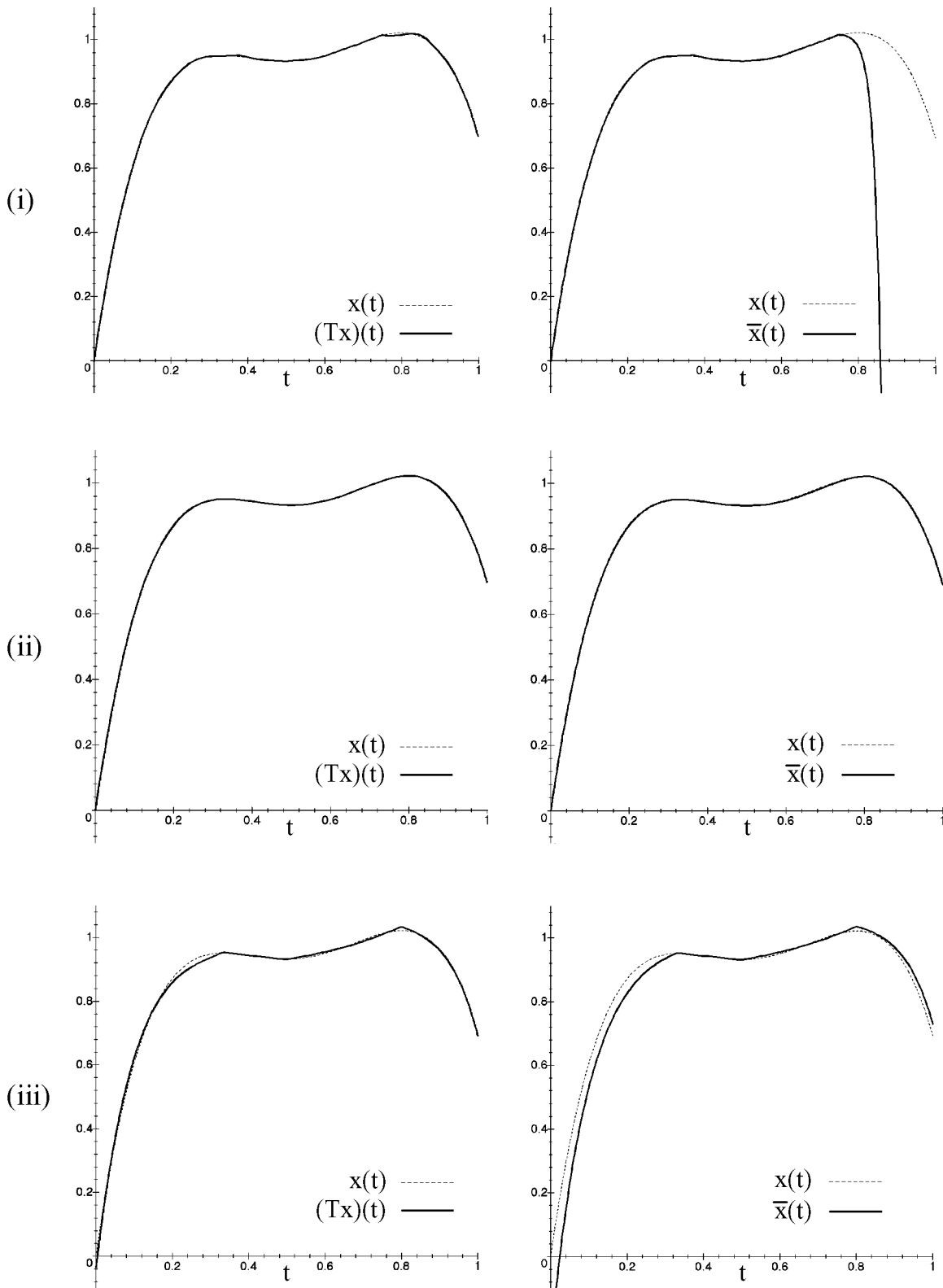


Figure 2: Results for  $x(t) = -\frac{125}{8}t^4 + \frac{1225}{36}t^3 - \frac{625}{24}t^2 + \frac{25}{3}t$ , partitioning at (i) the eighths, (ii) the sixteenths, and (iii) the three turning points.

	$f$	IC	$\ x - Tx\ _2$	$\ x - \bar{x}\ _2$
[0, 1] partitioned into halves	$\begin{cases} 9.7972 - 10.0253x, & 0 \leq t < \frac{1}{2} \\ -8.6121 + 8.7439x, & \frac{1}{2} \leq t \leq 1 \end{cases}$	0	0.0615	0.2286
[0, 1] partitioned into quarters	$\begin{cases} 8.7049 - 7.9211x, & 0 \leq t < \frac{1}{4} \\ -11.8923 - 12.5423x, & \frac{1}{4} \leq t < \frac{1}{2} \\ -5.2579 - 5.8172x, & \frac{1}{2} \leq t < \frac{3}{4} \\ -16.7268 + 16.2277x, & \frac{3}{4} \leq t \leq 1 \end{cases}$	0	0.0120	0.2064
[0, 1] partitioned into eighths	$\begin{cases} 8.4869 - 7.2825x, & 0 \leq t < \frac{1}{8} \\ -11.4444 - 11.4311x, & \frac{1}{8} \leq t < \frac{1}{4} \\ 29.5751 - 31.0530x, & \frac{1}{4} \leq t < \frac{3}{8} \\ 17.5673 - 18.8690x, & \frac{3}{8} \leq t < \frac{1}{2} \\ -16.0799 - 17.3036x, & \frac{1}{2} \leq t < \frac{5}{8} \\ -0.4913 - 0.9387x, & \frac{5}{8} \leq t < \frac{3}{4} \\ -56.6195 - 55.5788x, & \frac{3}{4} \leq t < \frac{7}{8} \\ -12.6412 + 11.7137x, & \frac{7}{8} \leq t \leq 1 \end{cases}$	0	0.0046	2.3353
[0, 1] partitioned into sixteenths	$\begin{cases} 8.3760 - 6.7740x, & 0 \leq t < \frac{1}{16} \\ 9.0679 - 8.2700x, & \frac{1}{16} \leq t < \frac{1}{8} \\ 10.6178 - 10.4032x, & \frac{1}{8} \leq t < \frac{3}{16} \\ 13.8282 - 14.1219x, & \frac{3}{16} \leq t < \frac{1}{4} \\ 20.3023 - 21.1350x, & \frac{1}{4} \leq t < \frac{5}{16} \\ -52.4732 + 55.1399x, & \frac{5}{16} \leq t < \frac{3}{8} \\ 1.4107 - 1.66574x, & \frac{3}{8} \leq t < \frac{7}{16} \\ 23.3108 - 25.0283x, & \frac{7}{16} \leq t < \frac{1}{2} \\ -24.8660 + 26.7004x, & \frac{1}{2} \leq t < \frac{9}{16} \\ -7.8283 + 8.58339x, & \frac{9}{16} \leq t < \frac{5}{8} \\ -1.9745 + 2.49173x, & \frac{5}{8} \leq t < \frac{11}{16} \\ 4.6966 - 4.27301x, & \frac{11}{16} \leq t < \frac{3}{4} \\ 37.1538 - 36.2923x, & \frac{3}{4} \leq t < \frac{13}{16} \\ -25.3855 + 24.6116x, & \frac{13}{16} \leq t < \frac{7}{8} \\ -14.9304 + 14.0873x, & \frac{7}{8} \leq t < \frac{15}{16} \\ -11.2059 + 10.0062x, & \frac{15}{16} \leq t \leq 1 \end{cases}$	0	0.0008	0.0013
[0, 1] partitioned at turning points	$\begin{cases} 9.2179 - 9.0701x, & 0 \leq t < \frac{1}{3} \\ 1.3907 - 1.6179x, & \frac{1}{3} \leq t < \frac{1}{2} \\ -2.7872 + 3.2159x, & \frac{1}{2} \leq t < \frac{4}{5} \\ -13.7093 + 12.9148x, & \frac{4}{5} \leq t \leq 1 \end{cases}$	0	0.0147	0.0783

Table 8: Inverse problem results for  $x(t) = \frac{-125}{8}t^4 + \frac{1225}{36}t^3 - \frac{625}{24}t^2 + \frac{25}{3}t$ , with  $x_0$  constrained.

If we look for a quadratic  $f(x, y)$  and  $g(x, y)$ , the collage distances decrease and the actual errors increase. For example, with the initial conditions variable, the resulting system is (correct to five decimal places)

$$\begin{aligned} \dot{x}(t) &= \begin{cases} 0.70604 - 0.06165x - 1.63697y + 1.84186x^2 - 4.82187xy + 2.72230y^2, & 0 \leq t < 0.997 \\ +0.01152 + 0.98342x - 0.03714y + 0.01686x^2 - 2.00054xy + 0.03837y^2, & 0.997 \leq t < 2.589 \\ -3.29172 + 6.62263x + 5.62499y - 3.15640x^2 - 4.97878xy - 3.15814y^2, & 2.589 \leq t < 3.850 \\ 0.09040 + 0.69522x - 0.12357y + 0.28138x^2 - 1.96479xy + 0.11719y^2, & 3.850 \leq t < 5.443 \\ -0.99662 - 3.14791x + 1.49264y - 3.74238x^2 + 2.53170xy - 2.74219y^2, & 5.443 \leq t < 6.349 \end{cases} \\ \dot{y}(t) &= \begin{cases} 0.02642 - 0.00990x - 1.05750y + 0.07206x^2 + 1.88356xy + 0.09884y^2, & 0 \leq t < 0.997 \\ +0.32796 - 0.43588x - 2.13089y + 0.40276x^2 + 2.18538xy + 0.99570y^2, & 0.997 \leq t < 2.589 \\ -5.58746 + 9.60207x + 8.61690y - 5.45901x^2 - 3.02357xy - 5.46959y^2, & 2.589 \leq t < 3.850 \\ 0.05207 + 0.17442x - 0.92828y - 0.17531x^2 + 1.99471xy + 0.07165y^2, & 3.850 \leq t < 5.443 \\ 0.17517 - 0.34236x - 1.26010y + 0.63842x^2 + 1.07931xy + 0.51321y^2, & 5.443 \leq t < 6.349 \end{cases} \end{aligned}$$

	$f$	IC	$\ x - Tx\ _2$	$\ x - \bar{x}\ _2$
[0, 1] partitioned into halves	$\begin{cases} 10.7048 - 11.0513x, & 0 \leq t < \frac{1}{2} \\ -7.9443 + 8.0786x, & \frac{1}{2} \leq t \leq 1 \end{cases}$	-0.602	0.0592	0.2283
[0, 1] partitioned into quarters	$\begin{cases} 8.9176 - 8.2062x, & 0 \leq t < \frac{1}{4} \\ -11.8923 - 12.5423x, & \frac{1}{4} \leq t < \frac{1}{2} \\ -5.2579 - 5.8172x, & \frac{1}{2} \leq t < \frac{3}{4} \\ -16.7268 + 16.2277x, & \frac{3}{4} \leq t \leq 1 \end{cases}$	0.0135	0.0116	1.2480
[0, 1] partitioned into eighths	$\begin{cases} 8.6130 - 7.5355x, & 0 \leq t < \frac{1}{8} \\ -11.4444 - 11.4311x, & \frac{1}{8} \leq t < \frac{1}{4} \\ 29.5751 - 31.0530x, & \frac{1}{4} \leq t < \frac{3}{8} \\ 17.5673 - 18.8690x, & \frac{3}{8} \leq t < \frac{1}{2} \\ -16.0799 - 17.3036x, & \frac{1}{2} \leq t < \frac{5}{8} \\ -0.4913 - 0.9387x, & \frac{5}{8} \leq t < \frac{3}{4} \\ -56.6195 - 55.5788x, & \frac{3}{4} \leq t < \frac{7}{8} \\ -12.6412 + 11.7137x, & \frac{7}{8} \leq t \leq 1 \end{cases}$	-0.0041	0.0044	2.3380
[0, 1] partitioned into sixteenths	$\begin{cases} 8.4124 - 6.9003x, & 0 \leq t < \frac{1}{16} \\ 9.0679 - 8.2700x, & \frac{1}{16} \leq t < \frac{1}{8} \\ 10.6178 - 10.4032x, & \frac{1}{8} \leq t < \frac{3}{16} \\ 13.8282 - 14.1219x, & \frac{3}{16} \leq t < \frac{1}{4} \\ 20.3023 - 21.1350x, & \frac{1}{4} \leq t < \frac{5}{16} \\ -52.4732 + 55.1399x, & \frac{5}{16} \leq t < \frac{3}{8} \\ 1.4107 - 1.66574x, & \frac{3}{8} \leq t < \frac{7}{16} \\ 23.3108 - 25.0283x, & \frac{7}{16} \leq t < \frac{1}{2} \\ -24.8660 + 26.7004x, & \frac{1}{2} \leq t < \frac{9}{16} \\ -7.8283 + 8.58339x, & \frac{9}{16} \leq t < \frac{5}{8} \\ -1.9745 + 2.49173x, & \frac{5}{8} \leq t < \frac{11}{16} \\ 4.6966 - 4.27301x, & \frac{11}{16} \leq t < \frac{3}{4} \\ 37.1538 - 36.2923x, & \frac{3}{4} \leq t < \frac{13}{16} \\ -25.3855 + 24.6116x, & \frac{13}{16} \leq t < \frac{7}{8} \\ -14.9304 + 14.0873x, & \frac{7}{8} \leq t < \frac{15}{16} \\ -11.2059 + 10.0062x, & \frac{15}{16} \leq t \leq 1 \end{cases}$	-0.0005	0.0008	0.0013
[0, 1] partitioned at turning points	$\begin{cases} 9.9388 - 9.9429x, & 0 \leq t < \frac{1}{3} \\ 1.3907 - 1.6179x, & \frac{1}{3} \leq t < \frac{1}{2} \\ -2.7872 + 3.2159x, & \frac{1}{2} \leq t < \frac{4}{5} \\ -13.7093 + 12.9148x, & \frac{4}{5} \leq t \leq 1 \end{cases}$	-0.2641	0.0122	0.0528

Table 9: Inverse problem results for  $x(t) = \frac{-125}{8}t^4 + \frac{1225}{36}t^3 - \frac{625}{24}t^2 + \frac{25}{3}t$ , with  $x_0$  variable.

with initial conditions  $x_0 = y_0 = 0.33333$ . The minimal collage distances add to 0.000040 and the actual errors are  $\|x - \bar{x}\|_2 \approx 0.000029$  and  $\|y - \bar{y}\|_2 \approx 0.000042$ .

**Example 10:** Returning to the van der Pol system of Eqs. (25)–(26), we apply our revised algorithm, partitioning at the turning points of  $x(t)$  and  $y(t)$ . In this case, the partition points are 0, 2.365, 3.261, 5.630, and 6.519. Searching for  $f(x, y)$  and  $g(x, y)$  of the correct form (i.e.  $f(x, y) = c_1y$  and  $g(x, y) = c_2x + c_3y + c_4x^2y$ ) leads to a system that is correct to three decimal places, regardless of whether we allow  $x_0$  and  $y_0$  to vary. Adding the minimal collage distances yields 0.000030 (IC constrained) and 0.000025 (IC variable). The actual  $\mathcal{L}^2$  errors are  $\|x - \bar{x}\|_2 \approx 0.000047$  and  $\|y - \bar{y}\|_2 \approx 0.00036$  (IC constrained) and  $\|x - \bar{x}\|_2 \approx 0.000007$  and  $\|y - \bar{y}\|_2 \approx 0.00011$  (IC variable).

As before, if we allow  $f$  and  $g$  to have quadratic terms, the collage distances decrease,

although the actual errors increase. In this case, with the initial conditions variable we arrive at the system (to five decimal places)

$$\dot{x}(t) = \begin{cases} 0.00173 - 0.00165x + 1.00158y + 0.00037x^2 - 0.00077xy + 0.00036y^2, & 0 \leq t < 2.365 \\ -0.00022 + 0.00004x - 0.99990y + 0.00008x^2 - 0.00002xy, & 2.365 \leq t < 3.261 \\ 0.00133 + 0.00111x + 0.99887y + 0.00023x^2 - 0.00048xy + 0.00023y^2, & 3.261 \leq t < 5.630 \\ -0.00281 - 0.00132x + 1.00066y + 0.00134x^2 - 0.00015xy + 0.00025y^2, & 5.630 \leq t < 6.519 \end{cases}$$

$$\dot{y}(t) = \begin{cases} 0.00088 - 1.00063x + 0.80088y + 0.00011x^2 - 0.00032xy + 0.00022y^2 - 0.79995x^2y, & 0 \leq t < 2.365 \\ 0.29959 - 0.73273x + 0.89845y + 0.05872x^2 - 0.11933xy - 0.00966y^2 - 0.76533x^2y, & 2.365 \leq t < 3.261 \\ -0.00228 - 1.00195x + 0.80211y - 0.00042x^2 - 0.00091xy - 0.00049y^2 - 0.79999x^2y, & 3.261 \leq t < 5.630 \\ 0.20160 - 1.14177x + 0.73799y + 0.02059x^2 - 0.08445xy - 0.01169y^2 - 0.82689x^2y, & 5.630 \leq t < 6.519 \end{cases}$$

with initial conditions  $x_0 = 1.99995$  and  $y_0 = 0.00021$ . The sum of the minimal collage distances is 0.000015, and the corresponding actual errors are  $\|x - \bar{x}\|_2 \approx 0.00001$  and  $\|y - \bar{y}\|_2 \approx 0.00173$ . Similar results are obtained when the initial conditions are constrained.

## 5 Inverse Problem For Multiple or Partial Data Sets

In practical applications, it is most probable that not one but a number of (experimental) data curves or sets will be available, all of which are to be considered as solution curves of a single ODE or system of ODEs. It is relatively straightforward to accomodate multiple data sets in our inverse problem algorithm of Section 3.2.1.

For each target function (each possibly a basis representation of some numerical data), we can calculate the squared collage distance of Eq. (17). Each integral involves the polynomial coefficients  $c_k$  of  $f$  since we assume that the data correspond to different solutions to the same DE. We then minimize the sum of the squared collage distances with respect to the variational parameters  $c_k$ ,  $0 \leq k \leq N$ . Notice that  $x_0$  can no longer be treated as a parameter since the different solutions will correspond to different  $x_0$  values. (It is also possible to minimize weighted sums of the squared collage distances if some of the solutions are presumed to be more/less accurate or perhaps more/less important.)

Minimizing the sum of the *squared*  $\mathcal{L}^2$  collage distances is much easier to perform computationally than minimizing the sum of the collage distances. Certainly, from a theoretical perspective, the sum of the squared collage distances is *not* a metric. Nevertheless, squeezing this sum towards zero guarantees a squeezing of the individual collage distances, which is sufficient for applications.

**Example 11:** Consider the anharmonic oscillator system

$$\dot{x}(t) = y(t), \tag{28}$$

$$\dot{y}(t) = -x(t) - 0.1x^3(t). \tag{29}$$

The period of the resulting cycle depends on the initial conditions that are imposed. In what follows, we consider the solutions that correspond to the two sets of initial conditions:  $\{x(0) = 1, y(0) = 0\}$  and  $\{x(0) = 2, y(0) = 0\}$ . Solving each system numerically and fitting each component of the numerical solution with a polynomial gives us our input data. The results of minimizing the combined collage distances for various polynomial basis degrees are given in Table 10.

polynomial basis degree	$f$	$g$
5	$0.99894640y$	$-1.02120674x - 0.09696297x^3$
10	$0.99962727y$	$-1.00059610x - 0.09976199x^3$
15	$1.00001325y$	$-0.99999964x - 0.10000014x^3$
20	$0.99999417y$	$-0.99999985x - 0.10000008x^3$
25	$1.00000322y$	$-1.00000006x - 0.09999995x^3$
30	$1.00000322y$	$-1.00000006x - 0.09999997x^3$

Table 10: Results for the anharmonic oscillator problem in Example 11 with two data sets.

It is also conceivable that only “partial data sets” are available; for example, only a portion of a solution curve  $(x(t), y(t))$  that is suspected to be a periodic orbit of a two-dimensional system of ODEs. In this situation, we simply apply our algorithm to the available data.

**Example 12:** We again solve numerically the Lotka-Volterra system of Example 6. This time, however, we only find a polynomial representation for a portion of the solution cycle and seek a system of ODEs which has this curve as a solution. Table 11 presents the results for both degree 10 and degree 30 polynomial fits and variously-sized partial data sets when Lotka-Volterra type systems are sought. Table 12 presents similar results in the case of quadratic systems of ODEs. All coefficients and initial conditions are presented to six decimal places. At first glance, the tables paradoxically seem to suggest that using less data yields better results. This artifact is actually due to the improved basis representation of the numerical data as the partial data set is made smaller: In all cases, the Riemann sum which approximates the  $\mathcal{L}^2$  distance between the numerical solution and the basis representation is computed using 80 partitions of the time interval under consideration. Therefore, as the time intervals are made smaller, the basis approximation is substantially improved. Comparing the results for degree 10 and degree 30 polynomial bases further confirms the observation that improved basis representations yield improved results. Of course, it bears mentioning that any of the earlier techniques for improving results can be used here. For example, partitioning the time domain at the turning points of  $x(t)$  and  $y(t)$  improves the basis representation and yields a better result, as expected.

## 6 Inverse Problem For Noisy Data Sets

The numerical examples in the earlier sections are highly idealized: the input to our algorithm is a polynomial representation of the exact numerical solution to a DE or system. In a practical application, it is far more likely that the interpolation of data points leads to a much rougher representation of the solution to the underlying DE or system. To simulate this imprecision, we can add Gaussian noise to our numerical solution before performing the basis fit. More precisely, the Riemann sum that approximates the  $\mathcal{L}^2$  distance between our solution and the basis representation is constructed by adding a small amount of noise

polynomial basis degree	percentage of cycle used	resulting Lotka-Volterra type system	IC
10	80	$\dot{x}(t) = 1.000283x - 2.000482xy$ $\dot{y}(t) = -1.000078y + 2.000064xy$	$x_0 = 0.333311$ $y_0 = 0.333540$
10	50	$\dot{x}(t) = 0.999999x - 1.999998xy$ $\dot{y}(t) = -1.000002y + 2.000004xy$	$x_0 = 0.333333$ $y_0 = 0.333333$
10	20	$\dot{x}(t) = 1.000000x - 2.000000xy$ $\dot{y}(t) = -1.000000y + 2.000000xy$	$x_0 = 0.333333$ $y_0 = 0.333333$
30	80	$\dot{x}(t) = 0.999996x - 1.999993xy$ $\dot{y}(t) = -1.000001y + 2.000004xy$	$x_0 = 0.333334$ $y_0 = 0.333332$
30	50	$\dot{x}(t) = 0.999998x - 1.999995xy$ $\dot{y}(t) = -1.000001y + 2.000003xy$	$x_0 = 0.333333$ $y_0 = 0.333333$
30	20	$\dot{x}(t) = 1.000000x - 2.000000xy$ $\dot{y}(t) = -1.000000y + 2.000000xy$	$x_0 = 0.333333$ $y_0 = 0.333333$

Table 11: Results for partial data sets and the Lotka-Volterra system.

to each sample of the solution. We then look for a DE or system that generates this noisy solution. The earlier partitioning ideas could also be used here. The effect of noise is in itself a large subject for inquiry. We limit our discussion to one illustrative example.

**Example 13:** Consider once again the Lotka-Volterra system of Example 6. The noise has been generated by using a normal, Gaussian distribution with unit standard deviation. The effects of noise addition are illustrated in Figure 3, in which are presented the solution components and their noisy basis representations, along with the periodic cycle in the  $xy$ -plane and its basis representation. The left column of the graphs corresponds to a degree 10 polynomial basis; the right column corresponds to a degree 30 polynomial basis. In the latter case, the periodic orbit is obliterated by the noise since the degree 30 polynomials can better represent the noisy oscillations in the data. Nevertheless, our algorithm yields the following results:

Degree 10 polynomial basis:

$$\begin{aligned}\dot{x}(t) &= 1.072174x - 2.134525xy, & x(0) &= 0.316367 \\ \dot{y}(t) &= -0.985326y + 1.960385xy, & y(0) &= 0.348851.\end{aligned}$$

Degree 30 polynomial basis:

$$\begin{aligned}\dot{x}(t) &= 1.013603x - 2.028574xy, & x(0) &= 0.332764 \\ \dot{y}(t) &= -1.021983y + 2.051016xy, & y(0) &= 0.352637.\end{aligned}$$

In both cases, the vector fields and initial conditions are quite close to those of the original Lotka-Volterra system in Example 6, Eqs. (23)-(24). However, they are noticeably worse than the results obtained from partial data sets in the previous section. Clearly, there is much room for further exploration on the effects of noise on our inverse algorithm.

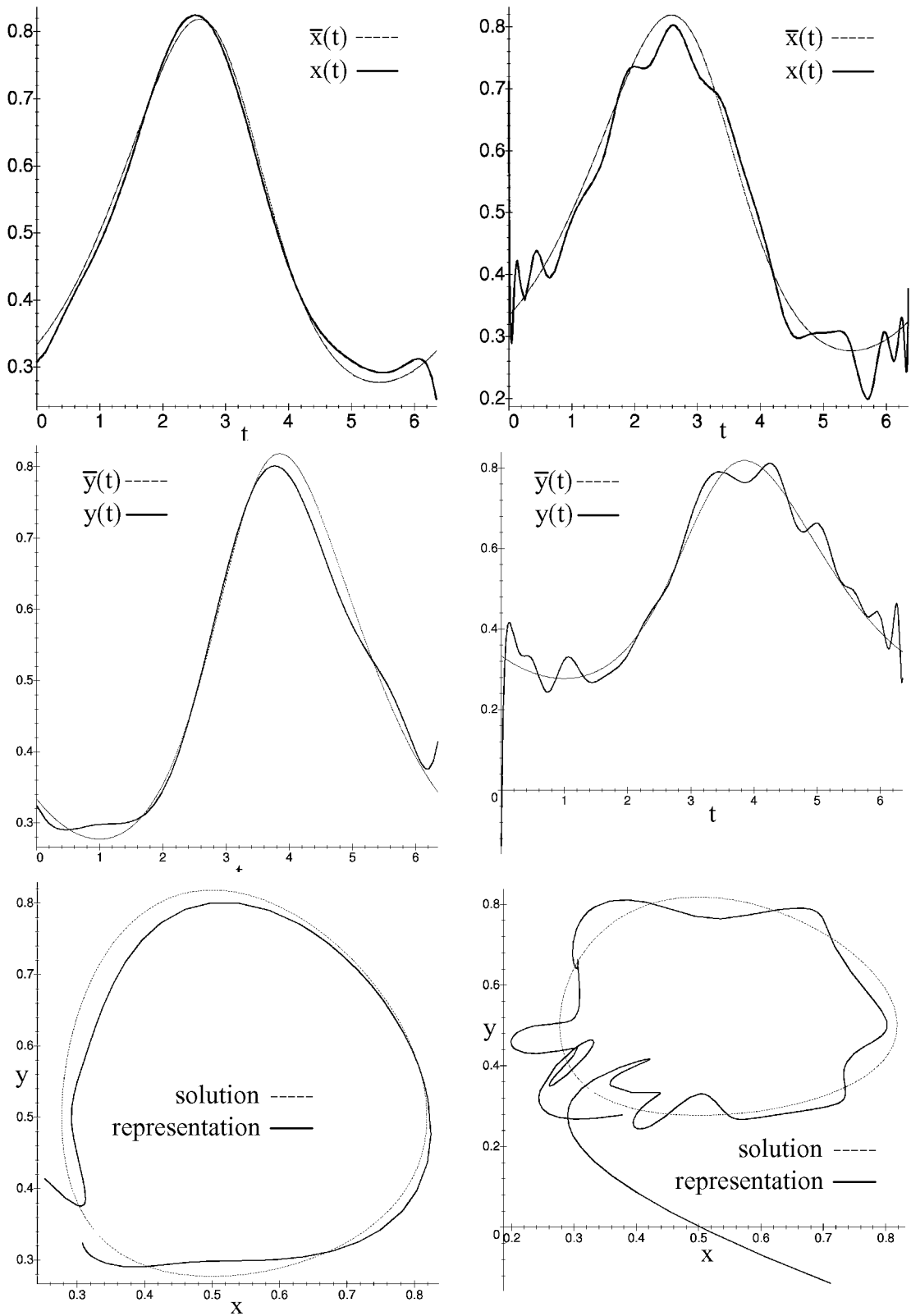


Figure 3: Lotka-Volterra solution components and orbits with noise added (left column: degree 10 polynomial basis; right column: degree 30 polynomial basis)

polynomial basis degree	percentage of cycle used	resulting quadratic system
10	80	$\dot{x}(t) = -0.018442 + 1.037874x + 0.044724y - 0.040676x^2 - 1.991041xy - 0.046363y^2$ $\dot{y}(t) = 0.010396 - 0.020771x - 1.027131y + 0.023885x^2 + 1.992760xy + 0.029110y^2$ $x_0 = 0.333071$ $y_0 = 0.333846$
10	50	$\dot{x}(t) = -0.000675 + 1.001050x + 0.002024y - 0.000937x^2 - 2.000340xy - 0.001716y^2$ $\dot{y}(t) = -0.000070 - 0.000121x - 0.999811y - 0.000178x^2 + 2.000219xy - 0.000367y^2$ $x_0 = 0.333330$ $y_0 = 0.333334$
10	20	$\dot{x}(t) = -0.000001 + 1.000002x + 0.000003y - 0.000003x^2 - 1.999996xy - 0.000006y^2$ $\dot{y}(t) = 0.000001 - 0.000001x - 1.000002y + 0.000002x^2 + 1.999997xy + 0.000003y^2$ $x_0 = 0.333333$ $y_0 = 0.333333$
30	80	$\dot{x}(t) = 0.000019 + 0.999963x - 0.000040y + 0.000026x^2 - 1.999992xy + 0.000036y^2$ $\dot{y}(t) = -0.000020 + 0.000042x - 0.999966y - 0.000039x^2 + 2.000010xy - 0.000032y^2$ $x_0 = 0.333333$ $y_0 = 0.333334$
30	50	$\dot{x}(t) = -0.000001 + 1.000001x + 0.000002y - 0.000003x^2 - 1.999994xy - 0.000002y^2$ $\dot{y}(t) = -0.000001 - 0.000000x - 0.999999y - 0.000000x^2 + 2.000005xy - 0.000001y^2$ $x_0 = 0.333333$ $y_0 = 0.333333$
30	20	$\dot{x}(t) = 0.000000 + 1.000000x + 0.000001y - 0.000001x^2 - 2.000000xy - 0.000001y^2$ $\dot{y}(t) = 0.000000 - 0.000001x - 1.000001y + 0.000001x^2 + 1.999999xy + 0.000001y^2$ $x_0 = 0.333333$ $y_0 = 0.333333$

Table 12: Results for partial data sets and the Lotka-Volterra system.

## 7 Concluding Remarks

We have introduced a systematic algorithm to provide solutions to the following inverse problem for ODEs: Given a function  $x(t)$  (which may be an interpolation of data points), find an ODE  $\dot{x}(t) = f(x, t)$  that admits  $x(t)$  as either an exact or, as closely as desired, an approximate solution, where the vector field  $f$  is restricted to a specific class of functional forms, e.g. polynomial. In particular applications, the exact functional form of the vector field may be known, e.g. Lotka-Volterra, and the algorithm may then be used to determine optimal values of parameters.

Our algorithm is based on “fractal-based” approximation methods that find vector fields  $f$  whose associated Picard operators  $T$  map the target solution  $x(t)$  as close as possible to itself. As with fractal approximation, however, the minimization of the collage distance does *not*, in general, imply optimality, that is, it is not guaranteed that we find the Picard operator  $T$  with the closest fixed point  $\bar{x}$  to  $x$ . In general, this is a difficult problem that has not been solved in the fractal compression literature. Some quite specialized studies have shown that rather modest improvements in approximation over the collage method are achieved, sometimes at significant computational expense. The examples presented in this paper show, as expected, that if a vector field for which the target curve is an exact solution exists in our prescribed class of functions, the collage method will find it. Nevertheless, for the more general cases where such exact solutions do not exist, we intend to investigate procedures to determine fixed points  $\bar{x}(t)$  that lie closer to the target than those obtained by collage distance minimization.

Finally, we are interested in exploring extensions and other applications of this method, for example, the determination of ODEs for chaotic systems (three and higher dimensions) as well as nonautonomous systems (e.g. external forcing). In principle, the method is straightforward, although computationally much more complicated.

## 8 Acknowledgements

This research was supported by grants from the Natural Sciences and Engineering Research Council of Canada (ERV) as well as by funding from the Department of Applied Mathematics and Faculty of Mathematics, University of Waterloo, all of which are most gratefully acknowledged.

## References

- [1] M.F. Barnsley, *Fractals Everywhere*, Academic Press, New York, 1988.
- [2] M.F. Barnsley, V. Ervin, D. Hardin and J. Lancaster, Solution of an inverse problem for fractals and other sets, *Proc. Nat. Acad. Sci. USA* **83** (1985), pp. 1975-1977.
- [3] P. Centore and E.R. Vrscay, Continuity of attractors and invariant measures for Iterated Function Systems, *Canad. Math. Bull.*, **37**(3) (1994), pp. 315-329.
- [4] E.A. Coddington and N. Levinson, *Theory of Ordinary Differential Equations*, McGraw-Hill, New York, 1955.
- [5] J. Crutchfield and B. McNamara, Equations of Motion from a Data Series, *Complex Systems*, **1** (1987), pp. 417-452.
- [6] Y. Fisher, *Fractal Image Compression, Theory and Application*, Springer-Verlag, New York, 1995.
- [7] B. Forte and E.R. Vrscay, Inverse Problem Methods for Generalized Fractal Transforms, in *Fractal Image Encoding and Analysis*, NATO ASI Series F, Vol. 159, ed. Y. Fisher, Springer Verlag, Heidelberg, 1998.
- [8] N. Lu, *Fractal Imaging*, Academic Press, New York, 1997.