

# MODELLING ITERATIVE ROOTS OF MAPPINGS IN MULTIDIMENSIONAL SPACES

Lars Kindermann

Brain Science Institute, RIKEN  
Lab. for Mathematical Neuroscience  
2-1, Hirosawa, Wako-shi  
Saitama, 351-0198, Japan  
kindermann@brain.riken.go.jp

Pando Georgiev

Brain Science Institute, RIKEN  
Lab. for Advanced Brain  
Signal Processing  
2-1, Hirosawa, Wako-shi  
Saitama, 351-0198, Japan  
georgiev@bsp.brain.riken.go.jp

## ABSTRACT

Solutions  $\varphi(x)$  of the functional equation  $\varphi(\varphi(x)) = f(x)$  are called iterative roots of the given function  $f(x)$ . They are of interest in dynamical systems, chaos and complexity theory and also in the modelling of certain industrial and financial processes. The problem of computing this "square root" in function (or operator) spaces remains a hard task and is, for the general case, still unsolved. While the theory of functional equations provides some insight for real and complex valued functions, iterative roots of mappings from  $\mathbf{R}^n$  to  $\mathbf{R}^n$  are not well understood by theory and there exists no published numerical algorithm for their computation. Here we prove existence of iterative roots of a certain class of monotonic mappings in  $\mathbf{R}^n$  spaces and demonstrate how a method based on neural networks can find solutions to some examples that arise from simple physical systems.

## 1. INTRODUCTION

### 1.1. Iterative roots and fractional iterations

Given some arbitrary self mapping of a set  $X$ , the solutions of the functional equation

$$\varphi(\varphi(x)) = f(x), \quad x \in X$$

(if they exist) are called iterative roots of  $f$ . Formally they can also be introduced as generalization of the operation of iteration towards non-integer iteration counts. In addition to the well known notations  $f^1 = f, f^0 = id, f^{-1}$  for the inverse of  $f$  and  $f^n = f \circ f \circ \dots \circ f$  ( $n$  times) for the  $n$ -th iterate of  $f$ , the iterative root of  $f$  can be symbolized by  $f^{1/2}$ . Furthermore  $f^{m/n}, m, n \in \mathbf{N}$  ( $\mathbf{N}$  being the set of all natural numbers) defines a fractional iterate of  $f$ . Finally  $f^t, t \in \mathbf{R}, t \geq 0$  is called continuous iterate of  $f$ , if the limits  $\lim_{q \rightarrow t} f^q(x)$  exist for any  $t \geq 0$  and any  $x \in X$  ( $X$  being in this case a topological space) and are equal to  $f^p(x)$ , if  $t = p$  (a rational number). In this case the one

parameter family of operators  $f^t : X \rightarrow X, t \geq 0$  is called a *continuous iteration semigroup* (see [9], page 398).

This problem was first considered by Charles Babbage, the inventor of the digital programmable computer, in his "Essay on the calculus of functions" from 1815 [2], where he extended the concept of algebra to functional spaces. The function combination  $f \circ \varphi$  is the most natural operation in this context and the iterates  $f^2$  arise naturally. Most famous is the so called Babbage equation  $\varphi(\varphi(x)) = x$ , whose solutions are denoted as "roots of identity".

The main field of applications of this formalism is in dynamical systems [1] and in the theory of complexity, where iterative roots of the exponential function play an important role [4]. But there are also industrial applications which require to find iterative roots of some order [5].

The problem of existence, uniqueness and analytical description of iterative roots for given mappings is far from solved, a survey paper from 2001 on functional equations states: "...one should not expect results on iterative roots in a general situation. In fact, even roots of polynomials are not described. Even worse: we do not know whether every complex cubic polynomial has a square root..." [3]. Furthermore, most of the theoretical work stays in the real or complex domain, while applications get more interesting in higher dimensional spaces.

### 1.2. Application to data from dynamical systems

Let  $x$  be a (complete) state vector of a deterministic dynamical system, that evolves continuously in time according to some unknown law.

Assume, that we can measure this state only in fixed time intervals of length  $\Delta t$  (sampling rate). From this measurements we obtain a set of pairs of states  $(x_0, x_1)$ .

Let  $f$  be the (initially unknown) function that describes how  $x$  changes in  $\Delta t$ , i.e.  $x_1 = f(x_0)$  (the finite flow).

Let us assume the ideal case that we can derive  $f$  from the training data arbitrarily well, e.g. by a neural network

that is trained on the data set. This is simple function approximation, one of the standard applications of neural networks.

Without loss of generality we can set  $\Delta t = 1$  and  $t = 0$  when measuring the initial state. Then iterating  $f$  on  $x_0$  will give the discrete time evolution of this state:

$$x_n = f^n(x_0), \quad n \in \mathbf{N}.$$

The task is to derive, if possible, from this *discrete time* mapping another mappings with smaller time steps or even “*continuous*” time model  $x(x_0, t)$ , described by a continuous iterative semigroup of  $f$

$$x(x_0, t) = f^t(x_0), \quad t \in \mathbf{R},$$

defined by the following three properties:

- 1)  $f^0(x) = x$  for any  $x \in X$  ( $X$  being a state space);
- 2)  $f^{t+s}(x) = f^t(f^s(x))$  for any  $x \in X$  and any  $t, s \geq 0$  (translation equation, or semigroup property).
- 3)  $\lim_{t \rightarrow 0+} f^t(x) = x$  for any  $x \in X$  (continuity property).

If, in addition,  $f^t(x_0)$  is non-expansive for fixed  $t$  (i.e.  $\|f^t(x) - f^t(y)\| \leq \|x - y\|$  for any  $x, y \in X$ ) there exists a unique (possibly multi-valued) m-accretive operator  $A : X \rightarrow X$  (see [9], page 387) such that  $f^t(x_0)$  is a strong solution of the differential inclusion

$$\dot{x}(t) \in -Ax(t), \quad x(0) = x_0.$$

called “evolution equation” (see [9], Theorem 8.19).

As solutions are generally hard to find, we attempt to calculate them approximately by the fractional iterate  $f^{m/n}$  with  $m, n \in \mathbf{N}$  and  $m/n \cong t$ , for which we have developed a neural network based method [5].

In short: under the given assumptions this method allows to reconstruct true continuous time trajectories from discrete time data. It is the natural interpolation method in a sense that it does not introduce new assumptions like linear or spline interpolation and is error free under ideal conditions.

However, in many practical situations the non-expansive property is not satisfied (even for the discrete mappings), but nevertheless, there exist iterative roots of such mappings. In the next section we shall see such examples.

## 2. ITERATIVE ROOTS OF MONOTONE MAPPINGS

It is well known that continuous monotone iterative roots of all orders exist for continuous strictly monotone function on the real line (see for instance [8]). In this section we generalize this result for the multidimensional case of strictly monotone mappings defined in  $\mathbf{R}^n$ .

Let  $C$  be a closed, convex, pointed cone ( $C \cap (-C) = \{0\}$ ) in  $\mathbf{R}^n$  with  $\text{int}C \neq \emptyset$ . We shall write  $x \geq_C y$  if and only if  $x \in y + C$ , and  $x >_C y$  if and only if  $x \in y + \text{int}C$ .

**Definition 2.1** *The mapping  $A : \mathbf{R}^n \rightarrow \mathbf{R}^n$  is said to be monotone (resp. strictly monotone) with respect to  $C$ , iff  $x \geq_C y$  (resp.  $x >_C y$ ) implies  $A(x) \geq_C A(y)$  (resp.  $A(x) >_C A(y)$ ) for any  $x, y \in H$ .*

**Theorem 2.1** *Suppose that  $A : \mathbf{R}^n \rightarrow \mathbf{R}^n$  is a continuous strictly monotone mapping with respect to the cone  $C$ , which is coercive (i.e.  $\lim_{\|x\| \rightarrow \infty} \|A(x)\| \rightarrow \infty$ ) and the inverse mapping  $A^{-1}$  is strictly monotone with respect to the cone  $-C$ . Then for any initial point  $x_0$  satisfying  $A(x_0) >_C x_0$  there exists a continuous strictly monotone iterative root of  $A|_D$  ( $A$  restricted to  $D$ ) where  $D = x_0 + C$ , i.e. there exists a continuous strictly monotone mapping  $B : D \rightarrow \mathbf{R}^n$  such that  $B(B(x)) = A(x)$  for any  $x \in D$ .*

**Proof.** Define the vector  $d = (A(x_0) + x_0)/2$  and the mapping  $B : D \rightarrow H$  by the following way. Put  $B(x_0) = d$  and define the mapping  $B$  on the set  $D_0 := (x_0 + C) \setminus (d + C)$  as

$$B(x) := B(x_0) + x - x_0.$$

Next, putting  $D_1 := (B(x_0) + C) \setminus (A(x_0) + C)$ ,  $D_{2(n+1)} = A(D_{2n})$  and  $D_{2n+1} = A(D_{2n-1})$  for  $n = 0, 1, 2, \dots$  we define by induction the mapping  $B$  on  $D$  as

$$B(x) = A(B^{-1}(x)) \quad (1)$$

starting from  $x \in D_1$ . We shall prove that

$$B(D_n) = D_{n+1}. \quad (2)$$

For  $n = 0$ , (2) is true by definition. We have also that  $B^{-1}$  is single-valued on  $D_1$ . Assume that (2) is true for some  $n$  and  $B^{-1}$  is single-valued on  $D_{n+1}$ . For  $n + 1$  we have: if  $y \in D_{n+1}$ , then  $y = B(x)$  for some  $x \in D_n$ , so  $x = B^{-1}(y)$  by induction hypothesis. By (1),  $B(y) = A(x) \in A(D_n) = D_{n+2}$ , so,  $B(D_{n+1}) \subset D_{n+2}$ . Let  $z \in D_{n+2}$ . Now, using the strict monotonicity, continuity and coercivity assumptions on  $A$ , and the condition that  $A^{-1}$  is strictly monotone with respect to the cone  $-C$ , one can easily derive that there exists  $x_z$  such that  $A(x_z) = z$ . Therefore  $z = B(B(x_z))$  and  $B(x_z)$  necessarily must belong to  $D_n$ . So, (2) is proved by induction.

Now, by (2) it follows that  $B$  is well defined on  $D$  and that  $B(B(x)) = A(x)$  for any  $x \in D$ . It is straightforward to prove that  $B$  is continuous on  $D$  (proceeding by induction on  $D_n$ ).

We shall prove that  $B$  is strictly monotone. We consider two cases:

- 1)  $B$  is restricted to  $\overline{D}_n$  (the closed hull of  $D_n$ ).

We proceed by induction. For  $n = 0$  the assertion is true by definition of  $B$  on  $D_0$ . Assume that for some  $n$ ,  $B$  restricted to  $D_n$  is strictly monotone. Let  $x_1, x_2 \in D_{n+1}$  and  $x_1 >_C x_2$ . By (2),  $x_i = B(y_i)$  for some  $y_i \in D_n, i =$

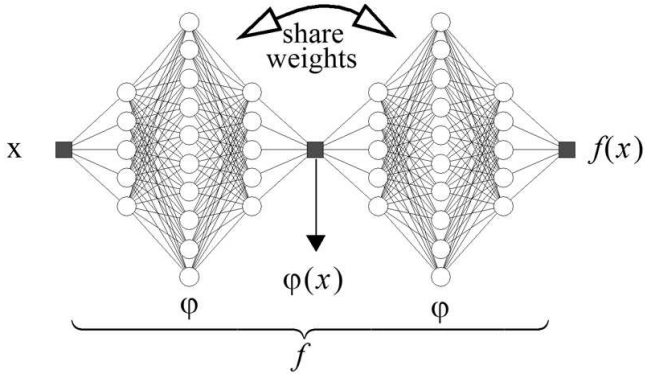


Figure 1: The whole network approximates  $f(x)$ , and - if they are kept identical for example by weight sharing methods - each subnet approximates the iterative root of  $f$

1, 2. Then, by definition of  $B$  and by induction hypothesis,  $B(x_1) = A(y_1) >_C A(y_2) = B(x_2)$ . So,  $B$  restricted to  $D_n$  is strictly monotone. Now by continuity it follows that  $B$  restricted to  $\overline{D}_n$  is also strictly monotone.

2) General case. Let  $x_1, x_2 \in x_0 + C$  and  $x_1 >_C x_2$ . Assume that the segment

$$[x_1, x_2] = \{x : x = \lambda x_1 + (1 - \lambda)x_2, \lambda \in [0, 1]\}$$

intersects the union of the boundaries of  $D_n, \cup_{i=0}^{\infty} bdD_i$  at points  $z_j, j \in J, J$  is an index set of natural numbers. We may assume without loss of generality that  $z_{j_1} >_C z_{j_2}$  if  $j_1 > j_2$ . Now applying case 1), we finish the proof. ■

**Remark 2.1** The construction of  $B$  is not unique: we have freedom to define  $B(x_0)$  and  $B$  on  $D_0$ .

### 3. APPROXIMATING ITERATIVE ROOTS WITH NEURAL NETWORKS

#### 3.1. Topology of the network

The idea how to map the problem of finding iterative roots of a given function  $f(x)$  to a neural network is straightforward: A network of the topology shown in figure 1, with the additional constraint that all the weights in both subnets are pairwise identical, is trained to approximate  $f$ . If this succeeds, each of the two identical subnets approximates it's iterative root  $\varphi$ [5].

This scheme can easily be extended to compute fractional iterates  $f^{m/n}$  by simply introducing more layers.

### 4. LEARNING RULES

The problem is to find a training algorithm for these networks, that incorporates the constraints of equal weights in

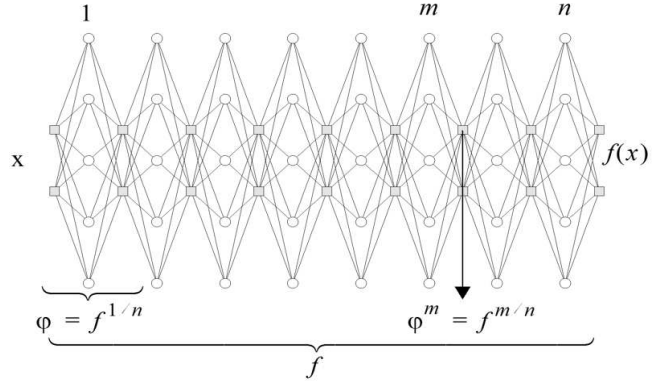


Figure 2: A multilayer network for computing fractional iterates of a two dimensional mapping.

all subnets and still has a fast convergence behavior. We derived two methods that give good results. One starts with different subnets and uses a penalty term, that assigns an error to the sum of all squared differences of corresponding weights in the subnetworks. Furthermore we calculated the exact derivatives  $\frac{\partial E}{\partial w}$  in the case of initially identical subnetworks (or a finitely recurrent network) which can be used in a slightly modified pseudo Newton algorithm [6].

Apart from the case where explicit functions are given as a formula, practical applications often rely on measurements or sampled data. If the function  $f$  is given only implicitly as input - output data pairs defined by a table of  $(x, f(x))$  values, the *training* set in neural network terminology, finding the iterative root becomes part of a regression problem: There may be noise or errors in this data and one has to deal with problems like generalization and overfitting. Neural networks have the proven ability to be highly successful in this context and lots of methods are available, which then naturally combine with the fractional iteration capability described here to provide practical solutions for specific applications.

Previously we have shown the capability of this method to find the iterative roots of several one-dimensional functions. This allowed to solve some interesting problems from physics with a purely data based approach [7].

#### 4.1. Improving convergence by initialization

The more complex the function  $f$  gets, the more difficult it is for the network to find a solution for it's iterative root. Local minima become a serious problem for the used network topology. Therefore it becomes important to start somewhere near the desired solution if possible. While it is difficult to calculate an exact solution of the iterative root, it can often be guessed how it looks like. This guess can be used to pre-train all subnetworks before the whole network is used

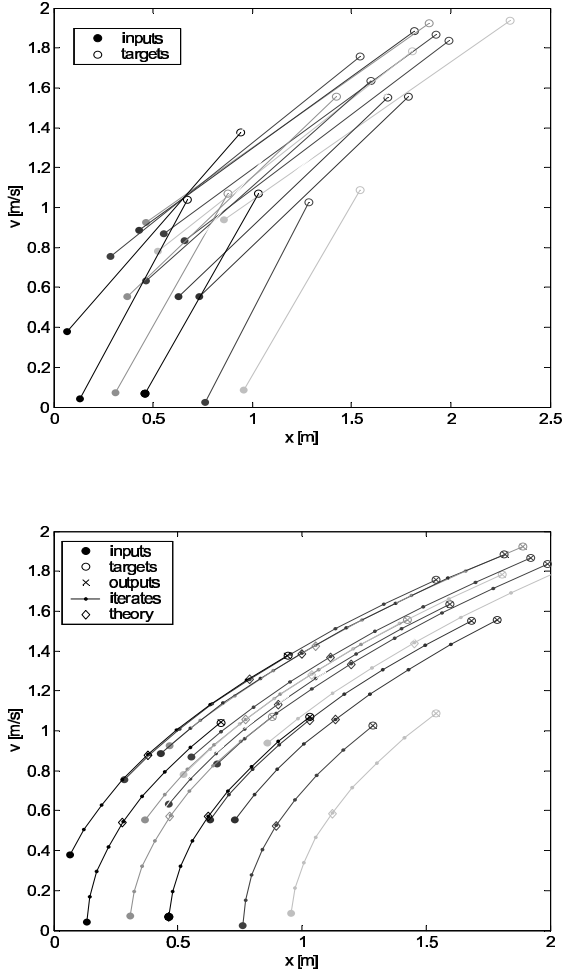


Figure 3: a) Data samples calculated from the free fall equations. They are used as training data. b) The reconstructed trajectories (8 steps) compared to the expected solution.

to find the exact solution. For strictly monotonic mappings there is always a solution that is located somewhere between identity and the mapping itself. A linear interpolation can be used to find a starting point for the search for  $\varphi$ .

For each pair of vectors from the training set  $(x, y)$  with  $y = f(x)$  create another set of  $n$  vector pairs  $(y_0, y_1), (y_1, y_2), \dots, (y_{n-1}, y)$  in the following manner

$$y_0 = x, \quad y_{i+1} = y_i + \frac{1}{n}(y - y_i)$$

and use this as the training data set for a single sublayer. For the given examples this reduced the number of training epochs by about a factor of ten in those cases the net found an solution and reduced the number of failures to find a solution due to local minima from about 90 % to about 5 %. This means, without initialization only one in ten attempts

to find a solution succeeds while with previous interpolation the method works almost immediately.

## 5. EXAMPLES

### 5.1. Free fall

The state of falling body is completely defined by its position and speed  $v$ . For the free fall with gravitational acceleration  $a$  we know how an initial state  $(x_0, y_0)$  evolves in time:

$$x(t) = x_0 + v_0 t + \frac{1}{2} a t^2, \quad v(t) = v_0 + a t.$$

From this we generated a set of data pairs  $(x_0, v_0), x(\Delta(t)), v(\Delta(t))$  as examples with a fixed time interval  $\Delta(t)$ . Figure 3a shows this mapping with closed circles marking initial states and open circles final positions. Of course this mapping should have iterative roots of all order because of it's construction from a continuous time system. But existence can also be guaranteed without this knowledge because the mapping is monotone with respect to the first octant, according to chapter 2. In order to reconstruct the complete trajectories from this data, we trained the network from figure 2 with this data and computed the fractional iterates up to order 8. The  $(2 - 5 - 2)^8$  network was able to reproduce the targets with an accuracy (relative mean square error) of  $10^{-8}$  and the iterative root with MSE of  $10^{-6}$ . Figure 3b shows the trajectories compared to the expected values (diamonds) at  $\Delta t/2$ .

### 5.2. Damped harmonic oscillator

The state of a harmonic damped oscillator  $x(t) = A \sin(\omega t - \varphi) e^{-ct}, v(t) = \dot{x}(t)$ , is sampled "stroboscopically" at fixed time intervals  $\Delta(t)$ . Figure 4a graphs this data closed circles mark a state  $(x, v)$  at  $t_0$ , open circles at  $t_0 + \Delta(t)$ .

While this is not a monotone mapping in the sense of chapter 2 the existence of solutions is also guaranteed because this is a non expansive mapping so there exist a continuous iterative semigroup that defines a continuous flow of this system.

We used the network from figure 2 again to compute the 8th fractional iterate on this data set to calculate trajectories with an 8-fold increased time resolution. The diamonds show the expected positions at times  $\Delta(t)/2$  which are modelled perfectly by the network. The accuracy is about the same as in the last example.

## 6. CONCLUSION

We presented a neural network model for finding rational roots of multidimensional mappings. We considered a special class of mappings, called monotone (with respect to a cone) and prove that they have iterative roots.

## 7. REFERENCES

- [1] R. Aldrovandi, L.P. Freitas, "Continuous Iteration of Dynamical Maps", *J. Math. Phys.* 39, pp. 5324-5336, 1998.
- [2] C. Babbage, "Essay towards the Calculus of functions", *Phil. trans. Royal Soc. London*, 105, pp. 389-424, 1915.
- [3] K. Baron, W. Jarczyk, "Recent results on functional equations in a single variable, perspectives and open problems", *Aequationes Math.*, 61, pp. 1-48, 2001.
- [4] G.H. Hardy, "Orders of infinity", *Cambridge Tracts in Mathematics*, 12, pp. 31, 1924.
- [5] L. Kindermann, "Computing Iterative Roots with Neural Networks", *Proc. Fifth Int. Conf. on Neural Information Processing - ICONIP'98*, Vol. 2, Kiyotakyushu, pp. 713-715, 1998.
- [6] L. Kindermann, P. Protzel, "Computing iterative roots with second order training methods", *Proceedings of the International Joint Conference on Neural Networks (IJCNN'2001)*, Washington DC, 2001.
- [7] L. Kindermann, P. Protzel, "Physics without laws - Making exact predictions with data based methods", *Proceedings of the International Joint Conference on Neural Networks (IJCNN'2002)*, Honolulu, pp. 1673-1677, 2002.
- [8] M. Kuczma, B. Choczewski, R. Ger, "Iterative Functional Equations", *Cambridge University Press*, Cambridge, 1990.
- [9] S. Hu, N.S. Papageorgiou, "Nonlinear semigroups, in: Handbook of Multivalued Analysis", *Math. & Its Applic.*, V. 419, Kluwer Academic, 1997.

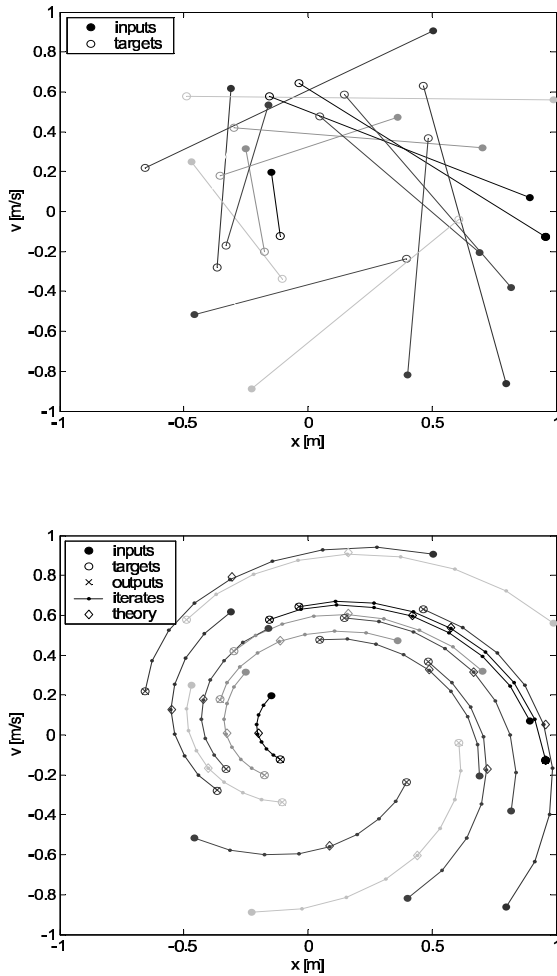


Figure 4: a) Data samples from a damped harmonic oscillator, sampled at fixed time differences. b) shows the reconstructed trajectories in phase space.