Verified Computation of High-Order Poincaré Maps

JOHANNES GROTE, KYOKO MAKINO and MARTIN BERZ
Department of Physics and Astronomy
Michigan State University
East Lansing, MI 48824
USA
grotejoh@msu.edu, makino@msu.edu, berz@msu.edu http://bt.pa.msu.edu/

Abstract: Poincaré maps often prove to be invaluable tools in the study of long-term behaviour and qualitative properties of a given dynamical system. While the analytic theory of these maps is fully explored, finding numerical algorithms that allow the computation of Poincaré maps in concrete problems is far from trivial. For the verification it is desirable to approximate the Poincaré map over as large a domain as possible. Knowledge of the flow of the system is a prerequisite for any computation of Poincaré maps. Taylor model based verified integrators compute final coordinates as high-order polynomials in terms of initial coordinates, with a small remainder error interval which typically is many orders of magnitude smaller than the initial domain. We present a method to obtain a Taylor model representation of the Poincaré map from the original Taylor model flow representation. First a high-order polynomial approximation of the time necessary to reach the Poincaré section is determined as a function of the initial conditions. This is achieved by reducing the problem to a non-verified polynomial inversion. This approximate crossing time is inserted into the Taylor model of the time-dependent flow, leading to an approximate Poincaré map. A verified correction is performed heuristically which provides a rigorous enclosure of the Poincaré map.

Key-Words: Poincaré map, verified computation, differential algebra, Taylor model.

1 Introduction

Poincaré maps are a standard tool in general dynamical systems theory to study qualitative properties of a continuous dynamical system under consideration (e.g. the flow generated by an ordinary differential equation), most prominently the asymptotic stability of periodic or almost periodic orbits. The traditional notion of a Poincaré map is that it describes how on a plane S (the Poincaré section) transversed by a periodic orbit $\mathcal{O}$ (the reference orbit) points which are sufficiently close to $\mathcal{O} \cap S$ get mapped back onto S by the flow. The two key benefits in this approach are that long-term behaviour of the flow close to $\mathcal{O}$ can be analyzed through the derivative of the Poincaré map at the intersection point of S and $\mathcal{O}$, which is available after just one revolution of $\mathcal{O}$, and that the dimensionality of the problem has been reduced by one since the Poincaré map is defined on S and neglects the 'trivial' direction of the flow perpendicular to the surface.

We will actually consider a somewhat generalized notion of Poincaré maps, by dropping the restriction that the flow exhibits a periodic reference orbit. Assuming we are given a smooth surface in phase space, the Poincaré section S, and an initial condition X_0 for the flow such that the orbit originating at X_0 actually transverses the section S at some crossing time, then by a Poincaré map we understand a map that projects all flows originating in a neighborhood of X_0 directly to the section S.

One is faced with the question which numerical representations of a flow are particularly favorable in the sense that they easily allow the computation of corresponding Poincaré maps for a given reference orbit and Poincaré section. In this paper we present a method that yields both polynomial

approximations of Poincaré maps of a certain type as well as their verification, i.e. a rigorous interval enclosure of the true map. The method assumes that a (verified) high-order polynomial approximation of the flow has been obtained previously using differential algebraic (DA) and Taylor model (TM) methods as described in [1, 4]. We focus on the case where the flow under consideration has been generated by an ODE.

The key question that will be discussed is how to project a domain box exactly to a given surface. If we recall that the flow representation in DA- or TM-arithmetic yields final coordinates in terms of initial coordinates, the key step will be to express the crossing time as a function depending on said initial conditions and insert it into the flow.

2 Review: DA- and TM-Tools

The DA- and TM-tools which are necessary to appreciate the method are described in detail in [1, 4]. Here we review briefly the most important applications of DA/TM-methods as far as they relate to the problem discussed here: the DA-integration method employed to obtain high-order polynomial approximations of the flow $\varphi(x_0, t)$, the verification technique which finds rigorous interval enclosures for $\varphi(x_0, t)$ and the functional inversion tools which are necessary in later steps of the algorithm.

2.1 DA-integration of ODEs

First we tackle the problem of obtaining a polynomial approximation of the solution of the initial value problem

$$\dot{x}(t) = f(x(t), t), \quad x(0) = X_0 + x_0 \qquad (1)$$

where $f : \mathbb{R}^{\nu+1} \supset U^{open} \longrightarrow \mathbb{R}^{\nu}$ is defined for DA-vectors (see below), i.e. f is a composition of intrinsic functions which have been defined in DA-arithmetic. This also entails that f exhibits sufficient smoothness to guarantee existence and uniqueness of solutions for all initial conditions. The vector $X_0 \in \mathbb{R}^{\nu}$ is constant and the midpoint of the domain box $D = X_0 + [-d_1, d_1] \times ... \times [-d_\nu, d_\nu] \subset \mathbb{R}^{\nu}$ for the small relative initial conditions $x_0 \in D$. Typical box widths d_i are of the order 10^{-2} to 10^{-8}. We remark that for the sake of simplicity, we will refer to the flow of the system as $\varphi(x_0, t)$ instead of $\varphi(X_0 + x_0, t)$, since X_0 is just the constant part of the initial value $x(0)$ and the actual interesting aspect is the dependence of the flow on the initial displacement x_0. Also, we will use the name $\varphi(x_0, t)$ for both the flow and its numerical representation (in DA- and TM-arithmetic respectively), it will be clear from the context which notion is meant. The desired polynomial approximation $\varphi(x_0, t)$ of the flow of eq.(1) is an expansion in terms of the independent time coordinate t and the relative initial conditions x_0, and the representation of this approximation is a so-called DA-vector which stores the expansion coefficients up to a prespecified order n in a structured fashion.

To achieve the aforementioned goal, we recall that the standard procedure of a Picard iteration yields a polynomial approximation of the solution of (1) after repeated application of a Picard operator on the initial conditions. The iteration in general increases the order of the expansion by at least one in every step, and since a DA-vector can store coefficients up to order n, we see that the iteration converges after finitely many steps in the DA-case (for details see [1]).

The Picard-operator in the DA-computation is defined by

$$\mathcal{C}(.) := (X_0 + x_0) + \partial_{\nu+1}^{-1} f(.) \qquad (2)$$

where f is computed in DA-arithmetic and $\partial_{\nu+1}^{-1}$is the *antiderivation operator*, essentially the integration with respect to the $(\nu + 1)$st variable t. It can now be shown that $\mathcal{C}$ is a contracting operator (with a suitable definition of a contraction, see [1]) and fixed-point theorems exist which guarantee that repeated application of $\mathcal{C}$ on the initial condition $x(0) = X_0 + x_0$ will converge to the DA-vector representation of the solution $\varphi(x_0, t)$ of (1) in $n + 1$ steps, where n is the order of computation. Subsequently the step-size of the integration step is inserted as the final time to eliminate the time-dependence and yield final coordinates only in terms of initial coordinates.

2.2 Taylor model verified integration of ODEs

Verification of the previously obtained DA-vector representation of the ODE-flow is performed with the full time-dependence of the solution, prior to the insertion of the stepsize. Assuming we have obtained such a DA-vector representation of the flow for one time step, it is now necessary for the verification to outfit this polynomial in time t and initial conditions x_0 with a rigorous remainder interval vector which encloses the true flow in a verified fashion.

The procedure to achieve this goal is explained in detail in [4]. Apart from being based on educated heuristics, it rests fundamentally on Schauder's fixed point theorem, which states that for a continuous operator A on a Banach space X a unique fixed point for A exists in any compact convex set $M \subset X$ if only $A(M) \subset M$. Moreover, the sequence $A^n(m)$ converges to the fixed point as $n \to \infty$ for arbitrary $m \in M$. Applied to the problem at hand, this means that for an interval vector I the true flow of the ODE is contained in the Taylor model $\varphi(x_0,t) + I$ exactly if

$$\mathcal{C}(\varphi(x_0,t) + I) \subset \varphi(x_0,t) + I,$$

where $\mathcal{C}$ is again the Picard operator, but evaluated in TM-arithmetic. Note also that the polynomial part stored in the DA-vector $\varphi(x_0,t)$ stays invariant. We proceed iteratively to find such an I. A first guess about the interval vector size needed to satisfy the inclusion property is obtained by evaluating

$$\mathcal{C}(\varphi(x_0,t) + [0,0]) = \varphi(x_0,t) + I^*$$

and setting $I^{(1)} := I^*$. If

$$\mathcal{C}(\varphi(x_0,t) + I^{(1)}) \subset \varphi(x_0,t) + I^{(1)},$$

we are done. If not, choose $I^{(k)} \supset I^{(k-1)}$, say $I^{(k)} := 2^k \cdot I^{(1)}$, until

$$\mathcal{C}(\varphi(x_0,t) + I^{(k)}) \subset \varphi(x_0,t) + I^{(k)}.$$

Once this is satisfied for some k, we have established the existence of a fixed point of $\mathcal{C}$ in $\varphi(x_0,t) + I^{(k)}$ according to Schauder and can proceed to sharpen the remainder bound $I^{(k)}$ by repeated application of $\mathcal{C}$ until the remainder bound has converged sufficiently. Again subsequently the stepsize is inserted for the time dependence.

2.3 Functional inversion using DA-arithmetic

Next we review the actual functional inversion employed to obtain the inverse $\mathcal{M}^{-1}$ of a function $\mathcal{M}$, or rather a DA-vector which stores the expansion coefficients of $\mathcal{M}^{-1}$ up to order n. Assume we are given a smooth map $\mathcal{M} : \mathbb{R}^\nu \longrightarrow \mathbb{R}^\nu$ s.t. $\mathcal{M}(0) = 0$ and its linearization M is invertible at the origin. This assures the existence of a smooth inverse $\mathcal{M}^{-1}$ in a neighborhood of the origin. If we write $\mathcal{M} = M + \mathcal{N}$, where $\mathcal{N}$ is the nonlinear part, and insert this into the fundamental condition $\mathcal{M} \circ \mathcal{M}^{-1} = \mathcal{I}$, we easily obtain the relation

$$\mathcal{M}^{-1} = M^{-1} \circ (\mathcal{I} - \mathcal{N} \circ \mathcal{M}^{-1})$$

and see that the desired inverse $\mathcal{M}^{-1}$ is a fixed point of the operator $\mathcal{D}(.) := M^{-1} \circ (\mathcal{I} - \mathcal{N} \circ .)$, which proves to be a contracting operator in the DA-picture (see [2]). Hence the existence of the fixed point $\mathcal{M}^{-1}$ of $\mathcal{D}$ is verified and $\mathcal{M}^{-1}$can be obtained through repeated iteration of $\mathcal{D}$, beginning with the identity $\mathcal{I}$. Also in this case the iteration converges to $\mathcal{M}^{-1}$ in at most $n+1$ steps (n is the computation order).

3 Computation of the Taylor Approximation

The first step we perform in order to find a Taylor model for the Poincaré map is to compute its Taylor polynomial part. This computation will be performed entirely in the DA-framework, i.e. without verification. It will prove possible to outfit the resulting polynomial with a rigorous error bound in a post-correction step.

3.1 Preliminary remarks

3.1.1 Reduction of the problem

We begin our discussion with the assumption that the ODE under consideration exhibits a periodic or almost periodic solution $\varphi(X_0, t)$ which starts on a suitable Poincaré section S and returns after a period T, which has been determined approximately e.g. by a high-order Runge-Kutta-integration.

We assume that we are given a suitable Poincaré section S (see next section for details) and that the flow $\varphi(0, t)$ originating at the initial condition X_0 crosses this section at the crossing time T. This crossing time needs to be known to high accuracy (however, a non-verified result for T is sufficient) and we assume that it is known. Typically the computation of T can be formed as a scalar constraint satisfaction problem and can be solved by standard algorithms for the solution of nonlinear systems.

Since tools for the verified integration of ODEs are available (as described in the previous section), the propagation of a domain box $X_0 + D$, where $D := [-d_1, d_1] \times ... \times [-d_\nu, d_\nu]$, through the time interval $[0, T]$ is merely a technicality and we assume that it has been performed until the last time step.

The interesting question of how this transported box can be projected to S only becomes apparent in this last step. Since it is intuitively clear that the time dependence of the TM-solution for the flow is necessary to perform this projection, we do not perform the usual insertion of the time-stepsize into the time dependence, but instead keep the full time expansion for this last time step at T.

3.1.2 Treatable types of sections

We want to consider as large a class of surfaces as Poincaré sections as possible. A suitable assumption is that the Poincaré section $S \subset \mathbb{R}^\nu$ is given implicitly in terms of a function $\sigma : \mathbb{R}^\nu \longrightarrow \mathbb{R}$ as $S := \{x \in \mathbb{R}^\nu : \sigma(x) = 0\}$. Since the function σ also needs to be expressed in terms of elementary functions available in the computer environment for DA/TM-arithmetic, it is necessarily smooth, and hence also the surface S. This contains most surfaces which are of practical interest, in particular the most common case where S is an affine plane of the form $S := \{x \in \mathbb{R}^\nu : x_1 = c\}$ where without loss of generality the first component x_1 of the vector x is set to the fixed value $c \in \mathbb{R}$; here $\sigma(x) = x_1 - c$.

3.1.3 Transversality of the flow

Another condition which needs to be met by S is that the flow is transversal to it for all possible initial conditions $x_0 \in D$, i.e. that

$$0 \notin (\nabla\sigma(\varphi(x_0, t)), f(\varphi(x_0, t))) \, \forall \, x_0 \in D.$$

Without this assumption a Poincaré map cannot be defined meaningfully, since for its definition the existence of a uniquely defined crossing time for each initial condition is required. However, for our method this question can be neglected. It is because in the 'pathological' case that the vectorfield is in the tangent space of the surface at any point, the functional inversion step described in the following will fail anyway.

3.2 Projection of the domain box to the section

The starting point of our discussion is that we have a Taylor model enclosure $\varphi(x_0, t) + I_\varphi$ of the flow at the time T, the crossing time of the reference orbit, where I_φ is a TM remainder error interval of the flow φ. Furthermore $\varphi(x_0, t) + I_\varphi$ still contains the dependence of the time-expansion around T.

Let t_l and $t_u > t_l$ be times such that at t_l the transported domain box has not yet crossed S and that at t_u the entire transported box has crossed the section. These times should be roughly of the dimension of the stepsize at the final step and the crossing condition can be checked using various range bounding tools as in [5, 6]. Then the interval $[t_l, t_u]$ contains a unique crossing time $t_c(x_0) \, \forall \, x_0 \in D$ determined by the geometry of S and the flow (or the vectorfield of the ODE, respectively). From a purely analytic standpoint, the existence of such a crossing time near the reference orbit is only guaranteed locally, however in practice usually D is small and both the flow and

S are somewhat well-behaved, and then $t_c(x_0)$ exists globally on D. This is again related to the question of transversality, if the crossing time does indeed not exist globally the inversion explained in the subsequent paragraphs will fail.

The construction of the projection (and thus Poincaré map) enclosure $\mathcal{P}(x_0)+I_{\mathcal{P}}$ can be reduced to the construction of a Taylor model for $t_c(x_0) + I_{t_c}$ for the crossing time. If we succeed in doing this, then $\mathcal{P}(x_0)+I_{\mathcal{P}}$ can be easily found simply by insertion of the crossing time into the flow

$$\mathcal{P}(x_0)+I_{\mathcal{P}} := \varphi(x_0 + [0,0], t_c(x_0) + I_{t_c}) + I_{\varphi}.$$

We note that from now on in this section all computations are performed in the nonverified DA-framework and proceed by constructing an artificial DA-valued function $\psi(x_0, t)$ with components $\psi_k(x_0, t)$ from the function x_0 with components $x_{0,k}$ via

$$\begin{aligned} \psi_k(x_0, t) &:= x_{0,k} \,\forall\, k \in \{1, ..., \nu\} \\ \psi_{\nu+1}(x_0, t) &:= \sigma(\varphi(x_0, t)) \end{aligned}$$

where the indices k denote components of the respective vectors.

Note that only the polynomial part $\varphi(x_0, t)$ of the Taylor model solution is used. To get an idea how the construction of ψ comes about, we remark that a functional inversion step is expected because of the implicit occurrence of the $t_c(x_0)$ in the problem, and hence ψ needs to map between spaces of equal dimension. Furthermore $t_c(x_0)$ depends on the variables x_0 and is determined by the constraint condition

$$\sigma(\varphi(x_0, t_c(x_0))) = 0 \tag{3}$$

and ψ contains both the constraint condition (3) and the independent variables x_0 as simple identities. Because of (3) $t_c(x_0)$ satisfies

$$\psi(x_0, t_c(x_0)) = (x_0, 0)$$

and suppose ψ is invertible then we can evaluate

$$\psi^{-1}(x_0, 0) = \psi^{-1}(\psi(x_0, t_c(x_0))) = (x_0, t_c(x_0))^T$$

and immediately extract the DA-vector representation of $t_c(x_0)$ in terms of the x_0 in the last component. In this case the invertibility of ψ at the point $(x_0, t_c(x_0))$ is actually guaranteed (at least in an analytic sense) by the condition of transversality.

We can now employ the previously described DA inversion tools to manipulate ψ and obtain the inverse ψ^{-1}. Naturally, because of the identities in ψ, also ψ^{-1} will preserve these identities and hence only the component $\psi^{-1}_{\nu+1}(x_0, 0)$ is nontrivial.

3.3 Summary of the algorithm for the Taylor approximation of the Poincaré map

We conclude the first part of the method by summarizing the algorithmic steps:

(1) Perform a verified transport of the initial domain box $X_0 + D$ for one cycle using polynomial expansion in time and initial conditions.
(2) Keep the time-expansion in the last time step at T.
(3) Find interval enclosure $I_t := [t_l, t_u]$ for all crossing times $t_c(x_0)$, $x_0 \in D$.
(4) Set up and invert the auxiliary function ψ using DA functional inversion to obtain a DA-vector representation of ψ^{-1}.
(5) Obtain $t_c(x_0) := \psi^{-1}_{\nu+1}(x_0, 0)$.
(6) Obtain the projection $\mathcal{P}(x_0) := \varphi(x_0, t_c(x_0))$.

4 Verification

4.1 Preliminary remarks

In the following we present a technique of finding a rigorous remainder bound for the truncation error of the polynomial approximation of the Poincaré map, which we have obtained in the previous section.

We remark that methods have been developed that allow to find Taylor model enclosures of the inverse function directly from a Taylor model for a function (see [2]), which would directly lead to a Taylor model enclosure of the crossing time if applied to a Taylor model extension of $\psi(x_0, t)$

in the previous section. However, these inversion techniques are much harder to implement than the method presented here.

Furthermore, for many applications the verification is actually not required and in those cases the nonverified method described above is satisfactory. The computation can be sped up by performing all TM-operations simply in the DA-framework and completely ignoring the remainder bounds.

As already noted, the verification of the Poincaré map automatically follows if we are able to find a verification for the crossing time $t_c(x_0)$. Recall that in the box $D \times I_t$ the polynomial approximation of the crossing time models the solution manifold of points $(x_0, t) \in D \times I_t$ which satisfy $\sigma(\varphi(x_0, t)) = 0$. The verification consists in finding a remainder bound which, if added to the polynomial $t_c(x_0)$, rigorously encloses the full part of the manifold which intersects $D \times I_t$.

For the purpose of verification we now also need a verified Taylor model representation of the constraint condition, i.e. in this case the parameterization of the section S. In other words, the function σ by the means of which the section is given implicitly needs to be enclosed in a Taylor model

$$\sigma(x_0) + I_\sigma := \sigma(x_{0,1} + [0,0], ..., x_{0,\nu} + [0,0])$$

where the right hand side of the last equation is evaluated in Taylor model arithmetic.

4.2 Heuristic enclosure of $t_c(x_0)$

We attempt to verify $t_c(x_0)$ in a heuristic manner by simply making an educated guess about the form of this remainder bound, based on the fact that its magnitude can roughly be estimated by evaluating $\sigma(\varphi(x_0, t_c + [0,0])) + I_\sigma$ over all $x_0 \in D$ using Taylor model range bounding techniques.

Pick random numbers $\epsilon_u^{(1)}, \epsilon_l^{(1)} \geq 0$ and construct a remainder bound $[-\epsilon_l^{(1)}, \epsilon_u^{(1)}]$ for t_c. The verification argument hinges on the fact that if we scan the complement $C := C^+ \cup C^- := (D \times I_t) \backslash \{t_c(x_0) + [-\epsilon_l^{(1)}, \epsilon_u^{(1)}] : x_0 \in D\}$ which lies 'above' and 'below' the set $\{t_c(x_0) + [-\epsilon_l^{(1)}, \epsilon_u^{(1)}] : x_0 \in D\}$ and can verify that the constraint $\sigma(\varphi(x,t)) = 0$ is violated there, then the feasible set of all crossing times which satisfy the constraint must be contained in the Taylor model $t_c(x_0) + [-\epsilon_l^{(1)}, \epsilon_u^{(1)}]$ and thus $[-\epsilon_l^{(1)}, \epsilon_u^{(1)}]$ is indeed a rigorous enclosure of the remainder error. If this is not the case, choose $\epsilon_u^{(2)} > \epsilon_u^{(1)}$ and $\epsilon_l^{(2)} > \epsilon_l^{(1)}$ and repeat the construction. So the only condition that needs to be checked is

$$0 \notin \sigma(\varphi(C)) + I_\sigma,$$

compare also figure 1.

This test can be performed by parameterizing the two disjoint parts $C^\pm$ of C in TM-arithmetic by the variables $(x_0, t^\pm(x_0, y))$ where

$$t^+(x_0, y) = (t_c(x_0) + \epsilon_u^{(1)}) \frac{1+y}{2} + t_u \frac{1-y}{2} \quad (4)$$
$$t^-(x_0, y) = (t_c(x_0) - \epsilon_l^{(1)}) \frac{1+y}{2} + t_l \frac{1-y}{2}$$

and then perform a range bounding of the two Taylor models

$$\sigma(\varphi(x_{0,1} + [0,0], ..., x_{0,\nu} + [0,0], t^+(x_0, y))) + I_\sigma$$
$$\sigma(\varphi(x_{0,1} + [0,0], ..., x_{0,\nu} + [0,0], t^-(x_0, y))) + I_\sigma$$

over the set $(x_{0,1}, ..., x_{0,\nu}, y) \in D \times [-1, 1]$ for the scanning of $C^\pm$ respectively.

One might ask why we choose such a somewhat awkward parameterization (4) for the $C^\pm$ in terms of the variables $x_{0,1}, ..., x_{0,\nu}, y$. This is a peculiarity of our computing environment. In principle it is arbitrary by which means we perform the range bounding in expression (4). However, we use the package COSY INFINITY [3] for our computations, which fully supports DA-arithmetic, interval arithmetic, and TM-arithmetic as well as the rigorous accounting for round-off and threshold errors. The range bounding necessary in (4) can be easily performed using Taylor model arithmetic (see [5, 6]) in COSY INFINITY, however fully rigorous error-handling in Taylor model range bounding is performed most efficiently if all involved Taylor models have the domain $[-1, 1]$ for each of their expansion variables. COSY INFINITY also can automatically rescale all Taylor models internally in such a way that they are defined on the domain $[-1, 1]^{\nu+1}$, but for the parameterization (4) we can

easily satisfy this requirement by hand calculation and prefer to do so.

References:

[1] M. Berz. *Modern Map Methods in Particle Beam Physics*. Academic Press, San Diego, 1999. Also available at http://bt.pa.msu.edu/pub.

[2] M. Berz and J. Hoefkens. Verified high-order inversion of functional dependencies and superconvergent interval Newton methods. *Reliable Computing*, 7(5):379–398, 2001.

[3] M. Berz, J. Hoefkens, and K. Makino. COSY INFINITY Version 8.1 - programming manual. Technical Report MSUHEP-20703, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, 2001. See also http://cosy.pa.msu.edu.

[4] M. Berz and K. Makino. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4(4):361–369, 1998.

[5] K. Makino. *Rigorous Analysis of Nonlinear Motion in Particle Accelerators*. PhD thesis, Michigan State University, East Lansing, Michigan, USA, 1998. Also MSUCL-1093.

[6] K. Makino and M. Berz. The LDB, QDB, and QFB Bounders. Technical Report MSUHEP-40617, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, 2004.

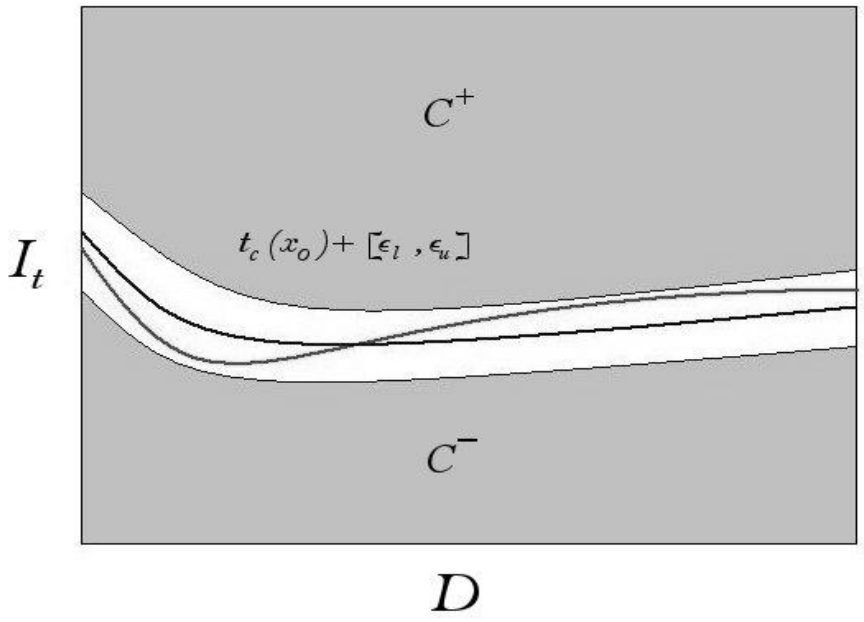

Figure 1: Domain box $D \times I_t$, solution manifold (grey line), Taylor model $t_c(x_0)+ [-\epsilon_l, \epsilon_u]$ (white), and complements $C^{\pm}$ (grey region).