RIGOROUS NUMERICS FOR MAPS AND ODEs

P. Zgliczyński

Jagiellonian University, Kraków, Poland

http://www.ii.uj.edu.pl/~zgliczyn

Interval arithmetics
a cure for round-off errors

Arithmetics on closed intervals. For example:

- $[1, 3] \langle + \rangle [3, 17] = [4, 20]$

- $1 \langle / \rangle 3 = [0.33333, 0.33334]$

diam$[a_-, a^+] = a^+ - a^-$, m$([a_-, a^+]) = (a^+ + a^-)/2$

Rigorous interval arithmetics can be realized on the computer i.e. for each arithmetic operator $\diamondsuit \in \{+, -, \cdot, /\}$ the following is true

$$[a_-, a^+] \diamondsuit [b_-, b_+] \subset [a_-, a^+] \langle \diamondsuit \rangle [b_-, b_+]$$

For any elementary function $f : \mathbf{R}^n \to \mathbf{R}^s$ and any set $Z \subset \mathbf{R}^n$

$$f(Z) \subset \langle f \rangle (\langle Z \rangle)$$

Interval arithmetics - problems

• wrapping
the result of evaluation of multidimensional map is product of intervals, disastrous results when considering f^n for n-large, ODEs

• dependency:
for $x = [-1, 1]$ holds

$$x \langle - \rangle x = [-2, 2]$$

Another example:

$$e^{-x} = \sum_{n=0}^{\infty} \frac{(-1)^n x^n}{n!}$$

$$[1 - \sinh(t), \cosh(t)] \subset \langle e^{-[0,t]} \rangle$$

$$\mathrm{diam}(\langle e^{-[0,t]} \rangle) \geq e^t - 1, \quad \mathrm{diam}(e^{-[0,t]}) = 1 - e^{-t}$$

Interval arithmetics - fighting the dependency problem

Let $[X] \subset \mathbf{R}^n$ - convex, $x_0 \in [X]$, $f : \mathbf{R}^n \to \mathbf{R}^k$ C^1-function, then

$$f([X]) \subset f(x_0) + [df([X])]_I \cdot ([X] - x_0) \quad (1)$$

where $[df([X])]_I$ is *the interval enclosure* of $df([X])$

$$[df(Z)]_I = \Big\{ M \in \mathbf{R}^{k \times n}, \\ M_{ij} \in \left[\inf_{z \in Z} \frac{\partial f_i}{\partial x_j}(z), \sup_{z \in Z} \frac{\partial f_i}{\partial x_j}(z) \right] \Big\}$$

Mean value form: Let $x_0 = m([X_0])$, then we set

$$\langle f \rangle ([X]) = \langle f \rangle (x_0) + \langle [df([X])]_I \cdot ([X] - x_0) \rangle$$

Interval arithmetics - fighting the dependency problem, Examples

- evaluation of $f(x) = x - x$, will give zero

- evaluation of $f(x) = x^2 - x^2$ on $[-1, 1]$

$< f([-1,1]) >= 0 + (2x - 2x)[-1,1] = [-4,4]$

rather bad result

- evaluation of $f(x) = e^{-x}$

$$< e^{-[0,t]} >= e^{-t/2} + \left(-e^{-[0,t]}\right) \cdot [-\frac{t}{2}, \frac{t}{2}] = e^{-t/2} + t/2 \cdot e^{-[0,t]} \cdot [-1,1] = e^{-t/2} + t \cosh(t)/2 \cdot [-1,1]$$

Hence

$$\text{diam}(< e^{-[0,t]} >) = t \cosh t = t + \frac{t^3}{2!} + \frac{t^5}{4!} + \dots,$$

for small t we have improvement, for t large it is worse.

Some methods for the reduction of the error growth

• *set division*: Let $S_t = \varphi(t, S)$. When S_t becomes too large, one should divide it into smaller pieces and compute further the evolution of each each piece separately

• *Lohner algorithm:* in order to avoid wrapping effect one should choose good coordinate frame in each step

• *Taylor models (Berz, Makino):*

Interval arithmetics - fighting the dependency problem, Taylor models(Berz, Makino)

All sets represented as image of polynomial maps plus small remainder term (Taylor models) - similar to symbolic computations

application of map to such that such - recomputation of the coefficient in the Taylor model

very general, flexible, virtually no dependency and wrapping problems

very slow and hard to program

One step of the Lohner algorithm

$x' = f(x)$ induces $\varphi(t, x_0)$ - t-time, x_0 - initial condition,

$\Phi(h, x)$ - numerical method, Taylor method of order p

Input:

- t_k - time,h_k - time step
- $[x_k] \subset \mathbf{R}^n$, such that $\varphi(t_k, [x_0]) \subset [x_k]$

Output:

- $t_{k+1} = t_k + h_k$
- $[x_{k+1}] \subset \mathbf{R}^n$, such that $\varphi(t_{k+1}, [x_0]) \subset [x_{k+1}]$

1. Rough enclosure of $\varphi([0,h_k],[x_k])$

$[W_1] \subset \mathbf{R}^n$ compact and convex

$$\varphi([0,h_k],[x_k]) \subset [W_1]$$

2. $[A_k] = \frac{\partial \Phi}{\partial x}(h_k,[x_k])$

3. $[x_{k+1}]$ ($m([x_k])$ - midpoint of $[x_k]$)

$$[x_{k+1}] = \Phi(h_k, m([x_k])) + [A_k]([x_k] - m([x_k])) + \text{Rem}([W_1])$$

Taylor method - for rigorous integration of ODEs

$x' = f(x)$, $x(0) = x_0$ give rise to $\varphi(t, x_0)$
$[X], [Y]$ - interval sets - products of intervals

If $[Y] = [X] + [0, h]f([Z]) \subset \text{int}[Z]$,
then $\varphi([0, h], [X]) \subset [Y]$.

Taylor expansion for $\varphi(h, x)$ for $x \in \mathbf{R}^n$, with respect to h (below $n = 1$), can be generated from ODE (automatic differentiation algorithm)

$$\frac{\partial}{\partial t}\varphi(0, x_0) = x^{(1)}(x_0) = f(x_0)$$

$$\frac{\partial^2}{\partial t^2}\varphi(0, x_0) = x^{(2)}(x_0) = f'(x_0)x^{(1)}$$

$$\frac{\partial^3}{\partial t^3}\varphi(0, x_0) = x^{(3)}(x_0) = f^{(2)}(x_0)(x')^2 + f'(x_0)x^{(2)}$$

$$\dots$$

Error term (r - the order of the Taylor method)

$$\frac{\partial^{r+1}}{\partial t^{r+1}}\varphi(\theta h, x_0) = \frac{\partial^{r+1}}{\partial t^{r+1}}\varphi(0, \varphi(\theta h, x_0)) \subset x^{r+1}([Y]).$$

$$\varphi(h, [X_0]) \subset [X_0] + \sum_{k=1}^{r} x^{(k)}([X_0])\frac{h^k}{k!} + x^{r+1}([Y])\frac{h^{r+1}}{(r+1)!}.$$

Reduction of the wrapping effect

$[x_k] = x_k + [r_k], \qquad x_k = m([x_k]), [r_k] = [x_k] - x_k$

The equation to evaluate:

$$[r_{k+1}] = [A_k][r_k] + [z_{k+1}]$$

Eventual reduction of the wrapping effect depends on how we will represent $[r_k]$

- interval set $[r_k] = \Pi I_j$, I_j -interval
- parallelepiped $[r_k] = B_k[\tilde{r}_k]$, B_k - matrix, $\tilde{r}_k$ -interval set
- cuboid $[r_k] = Q_k[\tilde{r}_k]$, $\tilde{r}_k$ -interval set, Q_k orthogonal matrix
- doubleton $[r_k] = C_k[r_0] + [\tilde{r}_k]$, C_k matrix, $[\tilde{r}_k]$ is either cuboid, parallelepiped or interval set

using interval sets == wrapping effect

other approaches try to minimize wrapping through choosing good coordinate frame

We choose different coordinate frame: $[r_k] = B_k[\widehat{r}_k]$,

$$[r_{k+1}] = [A_k][r_k] + [z_{k+1}] = B_{k+1}\left(B_{k+1}^{-1}[A_k]B_k[\widehat{r}_k] + B_{k+1}^{-1}[z_{k+1}]\right)$$

$$[r_0] = [B_0][\widehat{r}_0], \quad [B_0] = \{Id\}$$
$$[\widehat{r}_{k+1}] = \left([B_{k+1}^{-1}][A_k][B_k]\right)[\widehat{r}_k] + [B_{k+1}^{-1}][z_{k+1}]$$
$$[r_{k+1}] = [B_{k+1}][\widehat{r}_{k+1}]$$

Usually B_{k+1} is a Q-factor from QR decomposition of $U \in [A_k][B_k]$, but first we permute columns of U, so that their norms are decreasing

Even better:

$$[r_{k+1}] = C_{k+1}[r_0] + [\tilde{r}_{k+1}]$$
$$[\tilde{r}_{k+1}] = [A_k][\tilde{r}_k] + [z_{k+1}] + ([A_k]C_k - C_{k+1})[r_0],$$
$$[\tilde{r}_0] = 0$$
$$\text{and } C_0 = Id, \quad C_{k+1} \in [A_k]C_k$$

$[\tilde{r}_k]$ is evaluated using previous method

When Lohner algorithm can fail?

Only the first step - the generation of the rough enclosure - is heuristic. It can happen that a solution does not exists on $[0, h_k]$.

Easy first order rough enclosure

$$x' = f(x), \quad x \in \mathbf{R}^n, \quad f \in C^1 \tag{2}$$

$\varphi(t,x)$ the flow induced by (2)

Theorem: Let $h > 0$. Let X, Z be interval sets, $X \subset \text{int}Z$. Suppose that

$$Y := \text{interval hull}(X + [0,h]f(Z)) \subset \text{int}Z \tag{3}$$

then

$$\varphi([0,h],X) \subset Y \tag{4}$$

Problem: For $x' = -Lx$ we have a bound for the time step

$$h < \frac{1}{L}$$

Insert here an example for $n = 2$ with one dissipative coordinate

Improved rough enclosure for dissipative ODE

$$x_i' = f_i(x) = \lambda_i x_i + N_i(x), \quad i = 1, \dots, n \quad (5)$$

Theorem: $h > 0$, $X \subset Z \subset \mathbf{R}^n$ - interval sets. Let $D \subset \{1, \dots, n\}$ (*dissipative*(damped) directions), if $k \in D$, then

$$\begin{aligned} \lambda_k &< 0 \\ \lambda_k x_k + N_k^- &< \dot{x}_k < \lambda_k x_k + N_k^+ \end{aligned}$$

where $N_k(Z) \subset (N_k^-, N_k^+)$.

For $k \in D$ we set

$$\begin{aligned} b_k^\pm &= \frac{N_k^\pm}{-\lambda_k} \\ g_k^\pm &= \left(X_k^\pm - b_k^\pm\right) e^{\lambda_k h} + b_k^\pm. \end{aligned}$$

Let $Y = \Pi_{i=1}^{n} Y_i$ be such that

$$\begin{aligned} Y_i &= X_i + [0,h] f_i(Z), \qquad i \notin D \\ Y_i &= Z_i, \qquad i \in D. \end{aligned}$$

Then

$$\varphi([0,h], X) \subset Y,$$

provided the following conditions are satisfied for $i = 1, \ldots, n$

1. if $i \notin D$, then

$$Y_i \subset \text{int} Z_i$$

2. *upper bounds* for $i \in D$

$$\text{if} \quad Z_i^+ < b_i^+, \quad \text{then} \quad Z_i^+ \geq g_k^+$$

3. *lower bounds* for $i \in D$

$$\text{if} \quad Z_i^- > b_i^-, \quad \text{then} \quad Z_i^- \leq g_k^-$$

For

$$x' = -Lx, \quad L > 0$$

and $X = [-1, 1]$ one obtains $Y = [-1, 1]$ and no bound on $h > 0$.

Computation of the Poincaré map

- One needs a procedure which gives a rigorous estimates between time steps for x-variable for ODE - rough enclosure ok

- we need to come very close to section

- the section error is minimized for sections perpendicular to the flow

Lohner algorithm

$[x_k] = x_k + [r_k]$, where $[r_k] = Q_k[\tilde{r}_k]$ or $[r_k] = C_k[r_0] + Q_k[\tilde{r}_k]$

1. Finding rough enclosure $[W]$ of $\varphi([0, h_k], [x_k])$

2. $[A_k] = \frac{\partial \Phi}{\partial x}(h, [x_k])$

3. $x_{k+1} = m(\Phi(h_k, x_k))$,
$[z_{k+1}] = Rem([W]) + \Phi(h_k, x_k) - x_{k+1}$

4. $[r_{k+1}] = [A_k] \cdot [r_{k+1}] + [z_{k+1}]$ - the rearangement computations

Step 2 - the most time consuming part (we practically solve variational equation)

$Cost(Step2) \approx n^{?} Cost(Step3)$

$Cost(Step3) >> Cost(Step1 + Step4)$

Very slow compared to nonrigorous computations by factor of order $10n$.

Reasons:

the interval arithmetics is at least two times slower than nonrigorous one

The Lohner algorithm is a C^0-algorithm, but internally is in fact C^1

Question: Can one do better with C^0 algorithm?

C^0- algorithm

$[x_k] = x_k + B(0, r_k)$, where $r \in \mathbf{R}$, $B(0, r)$ - the ball of radius r. The choice of the good norm is an important parameter of algorithm.

1. Find rough enclosures $\varphi([0, h_k], [x_k]) \subset [W]$ of and $\varphi([0, h_k], x_k) \subset [W_1]$

2. compute $l = l(df([W]))$

3. $x_{k+1} = m(\Phi(h_k, x_k) + Rem([W_1]))$, $z_{k+1} = \Phi(h_k, x_k) + Rem([W_1]) - x_{k+1}$

4. $r_{k+1} = r_k e^{lh_k} + \|z_{k+1}\|$

Question: what is $l(df([W]))$?

Propagation of errors according to the typical numerical analysis textbook:

$$x' = f(x) \tag{6}$$

$|f(x) - f(y)| \leq L|x - y|$.

Let $\varphi(t, x_0)$ be a solution of (6) with an initial condition $x(0) = x_0$. Then

$$|\varphi(t, x) - \varphi(t, y)| \leq e^{Lt}|x - y|, \qquad t \geq 0$$

This is very bad estimate

Examples:

• $x' = -10x$, predicts error-growth: e^{10t}

• for the Lorenz attractor (from the proof by Galias and P. Z.), gives an estimate for Lipschitz constant for the Poincare map $L > 10^9$, while from simulations it is clear that $L \approx 5-6$

• in the proof for Rössler system (P.Z.), gives an estimate for the Lipschitz constant of Poincare map $L > 5 \cdot 10^{41}$, while from simulations $L \approx 2-3$ **cosmic computation time**

Logarithmic norms

Logarithmic norm: $Q \in R^{n \times n}$

$$\mu(Q) = \lim_{h>0, h \to 0} \frac{\|I + hQ\| - 1}{h}$$

can be negative !!!

- for Euclidean norm

  $\mu(Q) =$ the largest eigenvalue of $1/2(Q+Q^T)$.

- for max norm $\|x\| = \max_k |x_k|$

$$\mu(Q) = \max_k (q_{kk} + \sum_{i \neq k} |q_{ki}|)$$

- for norm $\|x\| = \sum_k |x_k|$

$$\mu(Q) = \max_i (q_{ii} + \sum_{k \neq i} |q_{ki}|)$$

Logarithmic norms - Fundamental lemma

Lemma: Let $\phi(t, x)$ be a flow induced by

$$x' = f(x).$$

Assume that Z is a convex set,

$$\begin{aligned} y([0,T]), \varphi([0,T], x_0) &\in Z \\ \mu\left(\frac{\partial f}{\partial x}(\eta)\right) \leq l, \qquad \text{for } \eta &\in Z \\ \left\|\frac{dy}{dt}(t) - f(y(t))\right\| &\leq \delta. \end{aligned}$$

Then for $0 \leq t \leq T$ we have

$$\|\varphi(t, x_0) - y(t)\| \leq e^{lt}\|y(0) - x_0\| + \delta\frac{e^{lt} - 1}{l}, \quad \text{if } l \neq 0.$$

For $l = 0$ we have

$$\|\varphi(t, x_0) - y(t)\| \leq \rho + \delta t.$$

In particular: e^{lT} is a Lipschitz constant for $\phi(t, \cdot)$ in Z (if Z is forward invariant).

Examples:

• $x' = -10x$, predicts error-growth: e^{-10t} very good

• in the proof for Rössler system (P.Z.) logaritmic norm based on the euclidian norm was used, the estimate for the Lipschitz constant of Poincare map in some region was $L > 2{\cdot}10^4$, while from simulations $L \approx 2-3$ this is doable. Using Lohner algorithm with cuboids one get Lipschitz constant around 60 and using doubletons something like $6-10$.

Lohner-type algorithm for differential inclusion

$$x'(t) \in f(x(t)) + [\delta] \tag{7}$$

$x \in \mathbf{R}^n$, $[\delta] \subset \mathbf{R}^n$

Find a rigorous enclosure for $x(t)$.

We compare the solutions of two ODEs

$$x_1' = f(x_1), \tag{8}$$

$$x_2' = f(x_2) + y(t) \tag{9}$$

$$x_1(t_0) = x_2(t_0) = x_0 \tag{10}$$

where $y(t) \in [\delta]$ is given (but unknown) function.

Lohner-type algorithm for differential inclusion - Fundamental Lemma

Lemma: Let:
$[W_1] \subset [W_2] \subset \mathbf{R}^n$ - convex and compact.
$x_1([t_0, t_0+h]) \subset [W_1]$, $x_2([t_0, t_0+h]) \subset [W_2]$ for any continuous function $y : [t_0, t_0 + h] \to [\delta]$.

Then the following inequality holds for $t \in [t_0, t_0+h]$ and for $i = 1, \ldots, n_1$

$$|x_{1,i}(t) - x_{2,i}(t)| \leq \left(\int_{t_0}^{t} e^{J(t-s)} C \, ds \right)_i , \quad (11)$$

where

$$\begin{aligned} C_i &\geq \sup |[\delta_i]| , \quad i = 1, \ldots, n_1 \\ J_{ij} &\geq \sup \frac{\partial f_i}{\partial x_j}([W_2]) \text{if } i = j, \\ J_{ij} &\geq \sup \left| \frac{\partial f_i}{\partial x_j}([W_2]) \right| \text{if } i \neq j. \end{aligned}$$

Lohner-type algorithm for differential inclusion - one step

$\varphi(t, x_0, [\delta])$ - a solution of $x' \in f(x) + [\delta]$, $x(0) = x_0$.
$\overline{\varphi}(t, x_0)$ - a solution of $x' = f(x)$, $x(0) = x_0$.

Input data:
t_k, h_k - a time step,
$[x_k] \subset \mathbf{R}^n$, such that $\varphi(t_k, [x_0], [\delta]) \subset [x_k]$.

Output data:
$t_{k+1} = t_k + h_k$,
$[x_{k+1}] \subset \mathbf{R}^{n_1}$, such that $\varphi(t_{k+1}, [x_0], [\delta]) \subset [x_{k+1}]$.

Lohner-type algorithm for differential inclusion - one step - details

1. Generation of a priori bounds for φ.
Find a convex and compact set $[W_2] \subset \mathbf{R}^n$, such that

$$\varphi([0,h_k],[x_k],[\delta]) \subset [W_2]. \tag{12}$$

2. Computation of $\overline{\varphi}$. We use Lohner algorithm to obtain $[\overline{x}_{k+1}] \subset \mathbf{R}^n$ and a convex and compact set $[W_1] \subset \mathbf{R}^n$, such that

$$\begin{aligned} \overline{\varphi}(h_k,[x_k]) &\subset [\overline{x}_{k+1}] \\ \overline{\varphi}([0,h_k],[x_k]) &\subset [W_1] \end{aligned}$$

Lohner-type algorithm for differential inclusion - one step - details continued

3. Computation of perturbation. Using Fundamental Lemma we find a set $[\Delta] \subset \mathbf{R}^n$, such that

$$\varphi(t_{k+1}, [x_0], [\delta]) \subset \overline{\varphi}(h_k, [x_k]) + [\Delta]. \qquad (13)$$

Hence

$$\varphi(t_{k+1}, [x_0], [y_0]) \subset [x_{k+1}] = [\overline{x}_{k+1}] + [\Delta] \qquad (14)$$

Lohner-type .. - details and comments

Part 3 - details

1. We set

$$
\begin{aligned}
C_i &= \text{right}\left(|[\delta_i]|\right), \quad i = 1, \ldots, n_1 \\
J_{ij} &= \text{right}\left(\frac{\partial f_i}{\partial x_i}([W_2])\right) \text{if } i = j, \\
J_{ij} &= \text{right}\left(\left|\frac{\partial f_i}{\partial x_j}([W_2])\right|\right), \text{if } i \neq j.
\end{aligned}
$$

2.$D = \int_0^h e^{J(h-s)} C \, ds$

3. $[\Delta_i] = [-D_i, D_i]$, for $i = 1, \ldots, n_1$

Lohner-type .. - Computation of $\int_0^t e^{A(t-s)}C\,ds$.

$$\int_0^t e^{A(t-s)}C\,ds = t\left(\sum_{n=0}^{\infty}\frac{(At)^n}{(n+1)!}\right)\cdot C. \qquad (15)$$

We fix any norm $\|\cdot\|$, preferably the L^{∞}-norm, ($\|x\|_{\infty} = \max_i |x_i|$).

$$\tilde{A} = At, \qquad A_n = \frac{\tilde{A}^n}{(n+1)!},$$

$$\sum_{n=0}^{\infty}\frac{(At)^n}{(n+1)!} = \sum_{n=0}^{\infty} A_n$$

$$A_0 = \mathrm{Id}, \qquad A_{n+1} = A_n \cdot \frac{\tilde{A}}{n+2}$$

Remainder: $\|A_{N+k}\| \leq \|A_N\|\cdot\left\|\frac{\tilde{A}}{N+2}\right\|^k$. If $\left\|\frac{\tilde{A}}{N+2}\right\| < 1$, then

$$\left\|\sum_{n>N} A_n\right\| \leq \|A_N\|\cdot\left\|\frac{\tilde{A}}{N+2}\right\|\cdot\left(1-\left\|\frac{\tilde{A}}{N+2}\right\|\right)^{-1}$$

Lohner-type .. - Representation of sets and rearrangement.

Lohner's approach.

In part 3:

$$[x_{k+1}] = [\overline{x}_{k+1}] + [\Delta] \qquad (16)$$

Evaluations 2 and 3. In this representation

$$[x_k] = x_k + [B_k][\tilde{r}_k]. \qquad (17)$$

In the context of our algorithm in part 3 we obtain

$$[\overline{x}_{k+1}] = \overline{x}_{k+1} + [B_{k+1}][\overline{r}_{k+1}]. \qquad (18)$$

We set

$$\begin{aligned} x_{k+1} &= \mathrm{m}(\overline{x}_{k+1} + [\Delta]) \\ [\tilde{r}_{k+1}] &= [\overline{r}_{k+1}] + [B_{k+1}^{-1}]\left(\overline{x}_{k+1} + [\Delta] - x_{k+1}\right). \end{aligned}$$

Lohner-type .. - Representation of sets and rearrangement II

Evaluation 4. In this representation

$$[x_k] = x_k + C_k[r_0] + [B_k][\tilde{r}_k]. \qquad (19)$$

In the context of our algorithm in part 3 we obtain

$$[\overline{x}_{k+1}] = \overline{x}_{k+1} + C_{k+1}[r_0] + [B_{k+1}][\overline{r}_{k+1}]. \qquad (20)$$

Equation (16) is taken into account exactly in the same way as in previous evaluations, i.e. we use equations (19) and (19).

Variational equations, C^n-computations

Let

$$\begin{aligned} \frac{\partial \varphi_i}{\partial x_j}(t, x_0) &= V_{i,j}(t), \\ \frac{\partial^2 \varphi_i}{\partial x_j \partial x_k}(t, x_0) &= H_{ijk}(t). \end{aligned}$$

It is well known that

$$x' = f(x), \tag{21}$$

$$\frac{d}{dt}V_{ij}(t) = \sum_{s=1}^{n} \frac{\partial f_i}{\partial x_s}(x) V_{sj}(t) \tag{22}$$

$$\frac{d}{dt}H_{ijk}(t) = \sum_{s,r=1}^{n} \frac{\partial^2 f_i}{\partial x_s \partial x_r}(x) V_{rk}(t) V_{sj}(t) + \sum_{s=1}^{n} \frac{\partial f_i}{\partial x_s}(x) H_{sjk}(x), \tag{23}$$

with the initial conditions

$$x(0) = x_0, \quad V(0) = Id,$$
$$H_{ijk}(0) = 0, \quad i, j, k = 1, \ldots, n.$$

An algorithm for C^n-computations

Simple approach: Apply C^0-Lohner algorithm to the system of variational equations, this works rather badly

• the control of wrapping effect may for x variables may not work

• computationally ineffective because it totally ignores the structure of the system,

Let $\Phi(h,x)$ be a Taylor expansion for $\varphi(h,x)$ of order p, then $V(h,x) = \frac{\partial \Phi}{\partial x}(h,x) + h^{p+1}$

Observe that $\frac{\partial \Phi}{\partial x}(h,[W])$ is already computed in step 2 of C^0 algorithm

An effective C^n-algorithm

- takes into account the structure of the system variational equations

- the rearrangement is done separately which partial derivatives of given order

- implemented in CAPD library, we did some computer assisted proofs involving C^5 computations for ODE $n = 2, 3$