Taylor Models and Their Applications

Martin Berz and Kyoko Makino

Department of Physics and Astronomy
Michigan State University

Contributions: Alex Wittig (MSU)
R. Armellin, P. Di Lizia (Politecnico di Milano)

Outline

- Motivation
- Introduction to Taylor models
  Basics. Arithmetic. Comparisons with interval computations.
- Applications — Demonstrate with examples
  - Range bounding
    Moore's simple 1D function
  - Global optimizations
    Moore's simple 1D function. Normal form defect functions.
  - ODE integrations
    The Volterra equations. Near earth asteroid Apophis.
    The Lorenz equations.[Long time integrations, Poincare projections, covering huge size initial conditions, manifolds]
- Work in progress

Introduction

Taylor model (TM) methods were originally developed for a practical problem from nonlinear dynamics, range bounding of normal form defect functions.

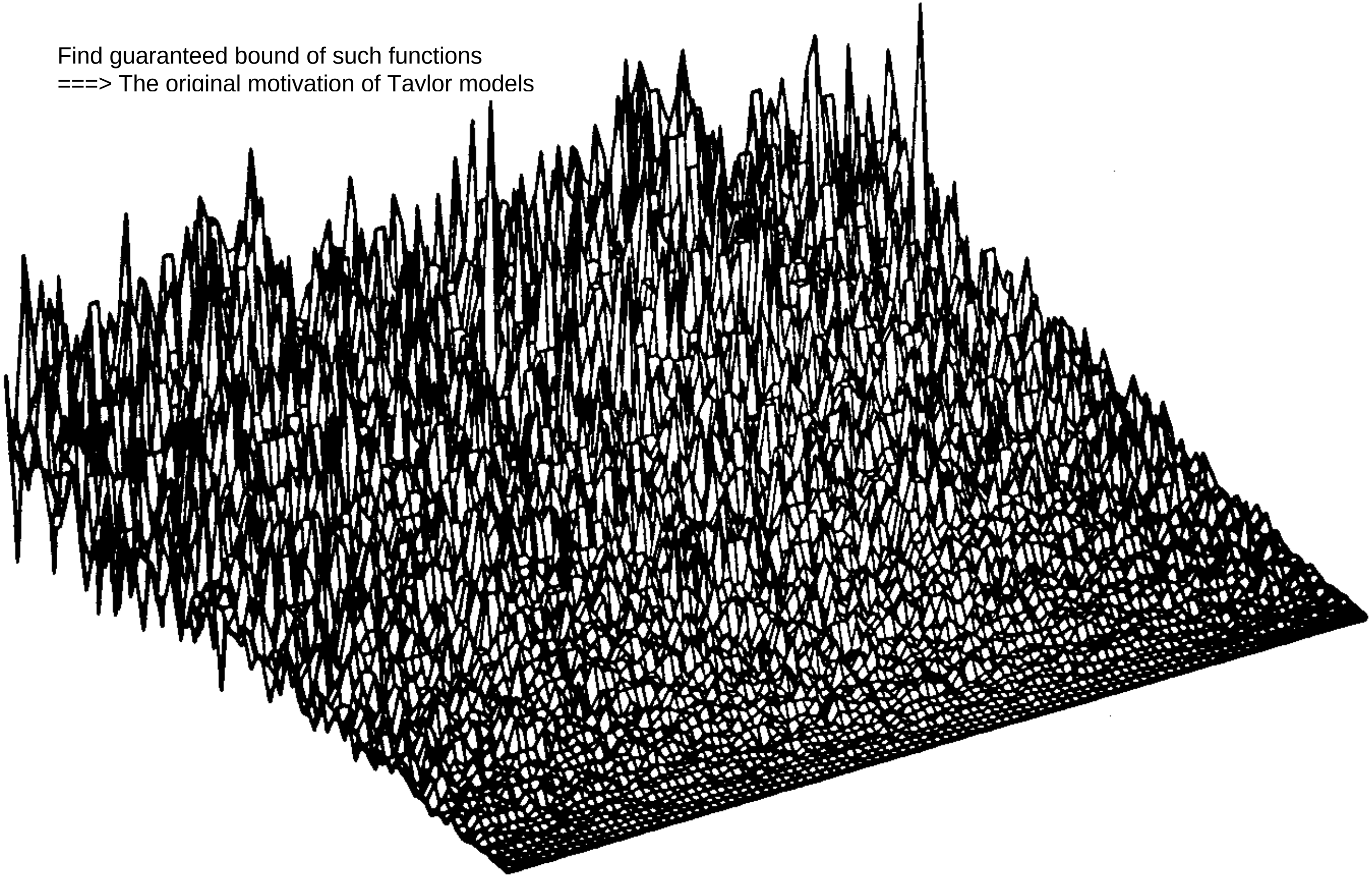
- Functions consist of code lists of 10^4 to 10^5 terms
- Have about the worst imaginable cancellation problem
- Are obtained via validated integration of large initial condition boxes.

Originally nearly universally considered intractable by the community. But ... a small challenge goes a long way towards generating new ideas!

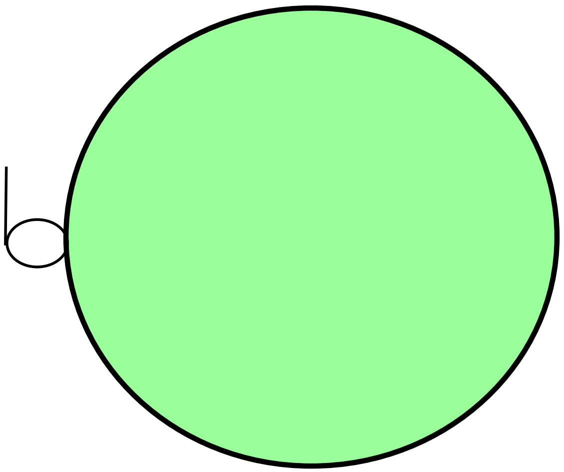
Idea: represent all functional dependencies as a pair of a polynomial P and a remainder bound I, introduce arithmetic, and a new ODE solver. Obtain the following properties:

- The ability to provide enclosures of any function given by a finite computer code list by a Taylor polynomial and a remainder bound with a sharpness that scales with order $(n + 1)$ of the width of the domain.
- The ability to alleviate the dependency problem in the calculation.
- The ability to scale favorable to higher dimensional problems.

Find guaranteed bound of such functions
===> The original motivation of Taylor models



Motion in the Tevatron



- Speed of Light: $3x10^8$ m/sec
- Circumference: $6.28x10^3$ m

→ $4x10^4$ revs/sec.

- Need to store about 10 hours, or $4x10^5$ sec

→ $2x10^{10}$ revolutions total.

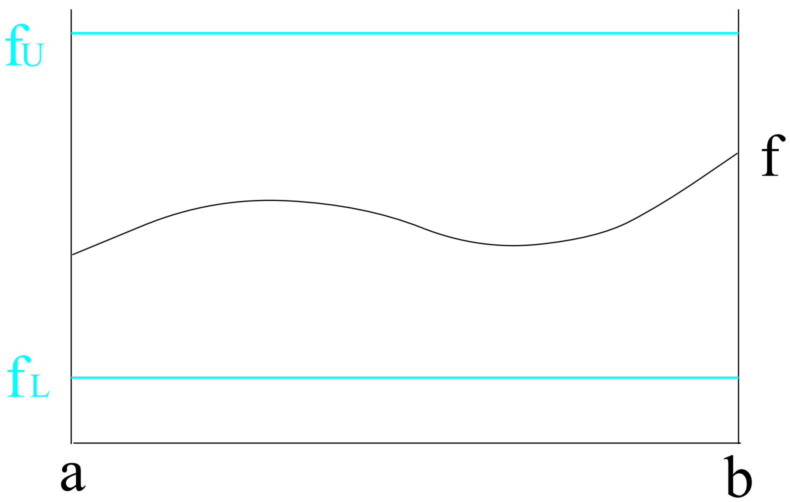
- 10,000 magnets in ring

→ $2x10^{14}$ contacts with fields!

- Extremely challenging computationally
- Need for several State-Of-The-Art Methods:
  - Phase Space Maps
  - Perturbation Theory
  - Lyapunov- and other Stability Theories
  - High-Performance Verified Methods

Bounds Estimates

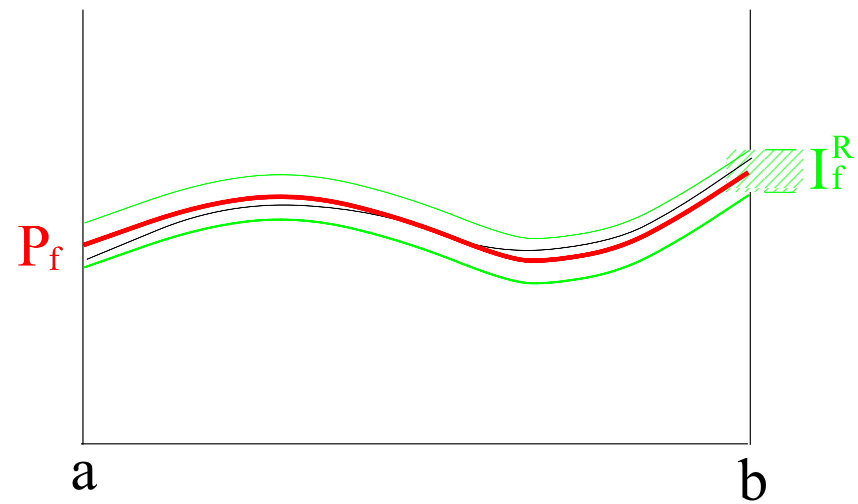
Interval Method



Bounds: $[f_L, f_U]$

RDA

Taylor Model Method



Bounds: $P_f + I_f^R$

P_f : Polynomial

I_f^R : Remainder Bounds

Interval Arithmetic

A method to perform guaranteed calculations on computer by presenting all numbers by intervals.

$$[a, b] + [c, d] = [a + c, b + d]$$
$$[a, b] - [c, d] = [a - d, b - c]$$
$$[a, b] \cdot [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$$
$$[a, b]/[c, d] = [\min(a/c, a/d, b/c, b/d), \max(a/c, a/d, b/c, b/d)]$$

Not a group because $[a, b] - [c, d] \neq [0, 0]$ unless $a = b, c = d$.
In particular,

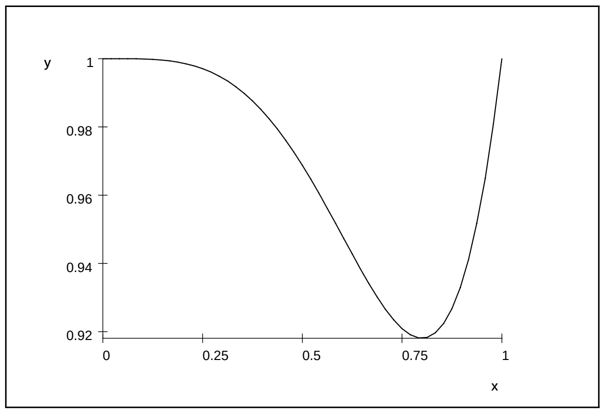
$$[a, b] - [a, b] = [a - b, b - a]$$
$$[a, b]/[a, b] = [\min(1, a/b, b/a), \max(1, a/b, b/a)]$$

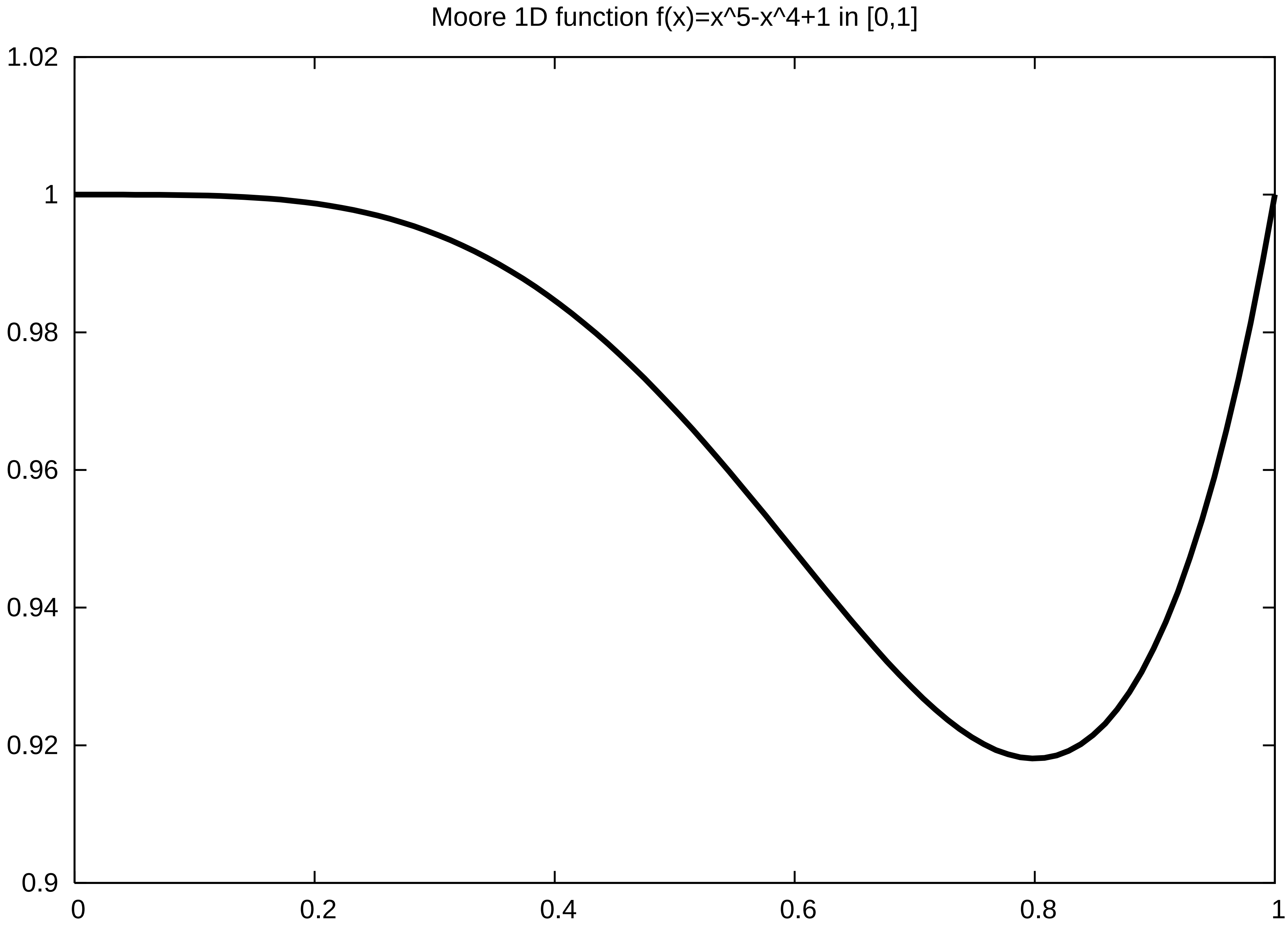
Thus, operations lead to over estimation, which can become large with time to blow up.

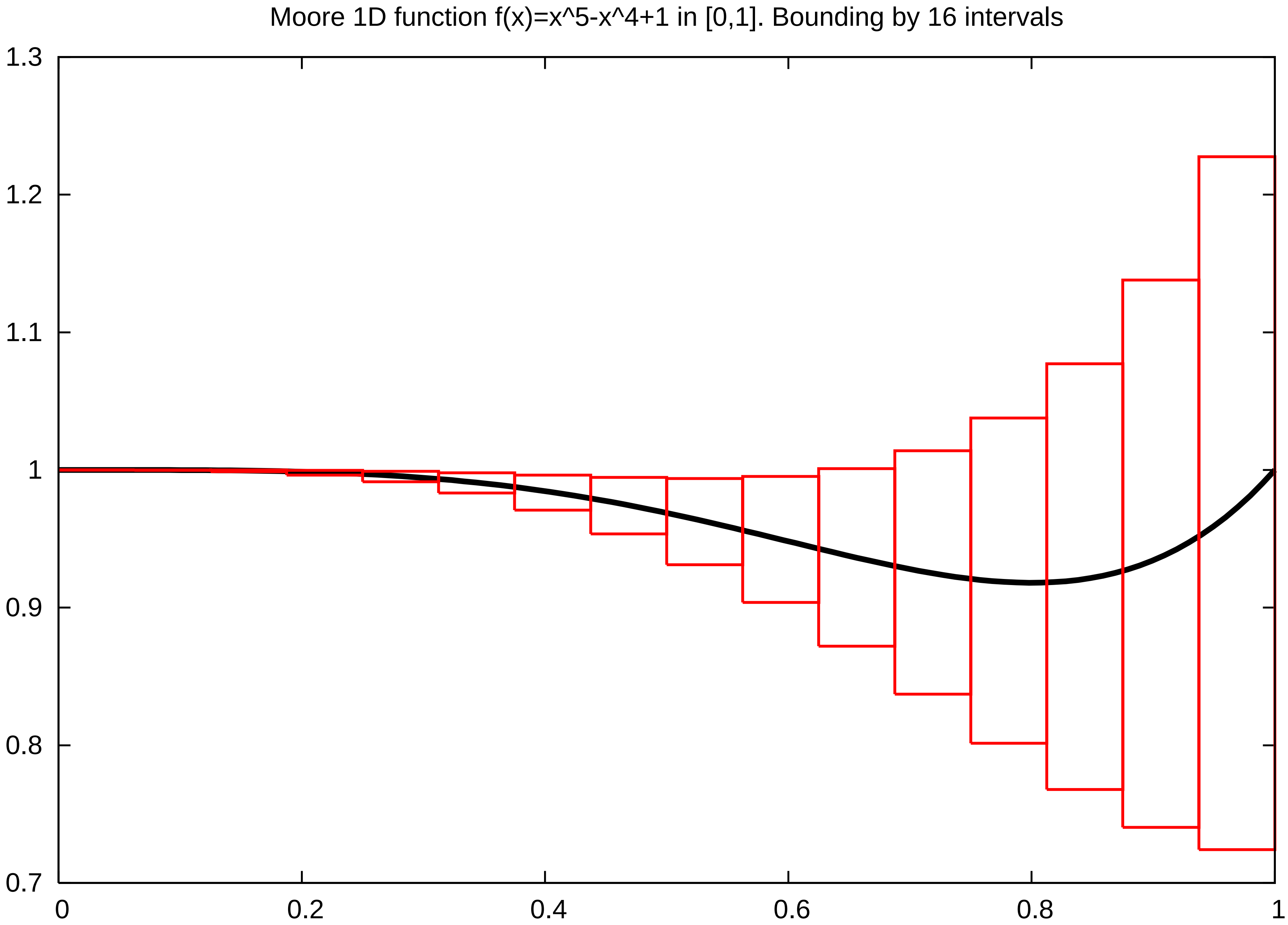
Moore's Simple 1D Function

$$f(x) = 1 + x^5 - x^4.$$

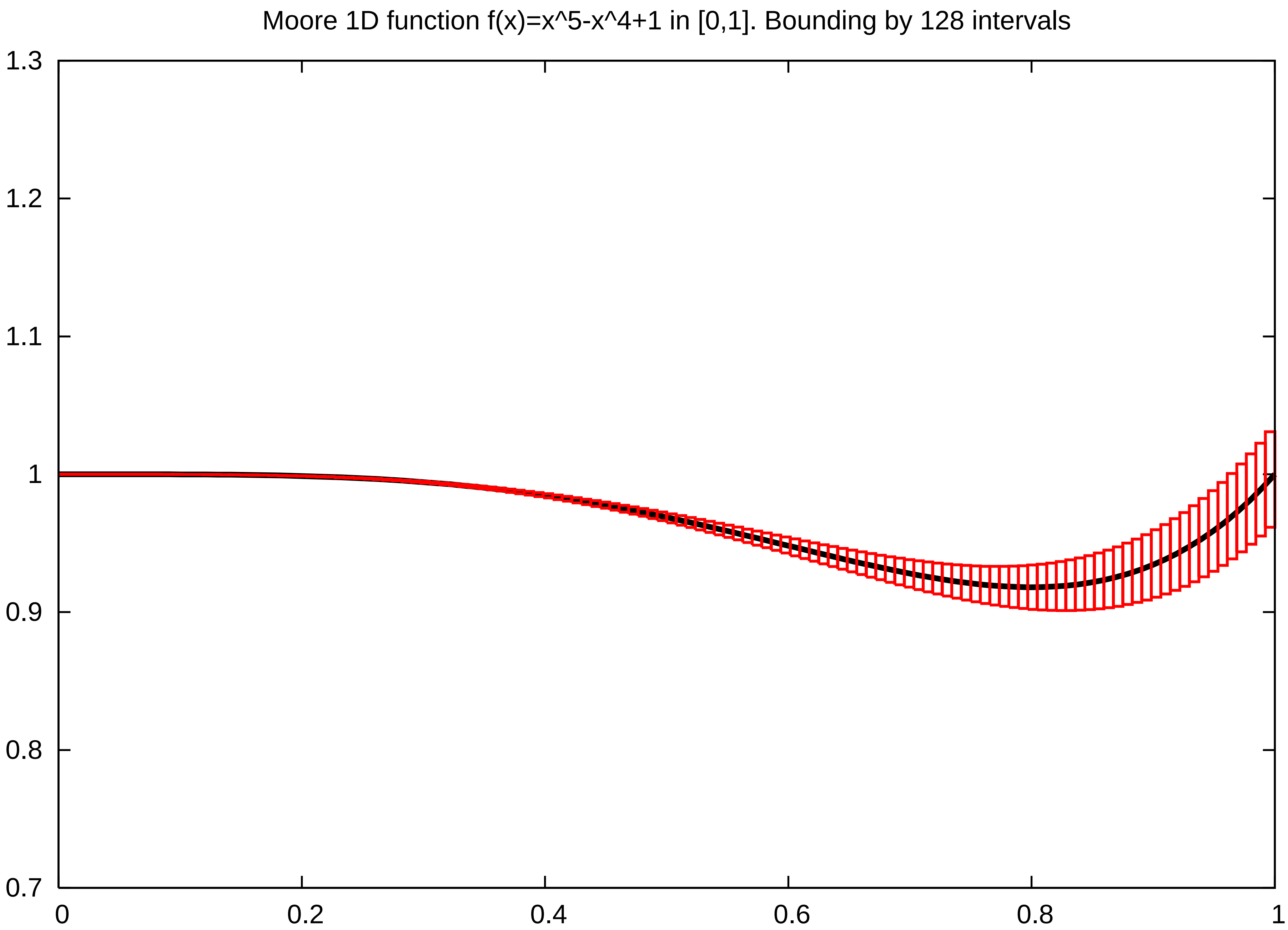
Study on $[0, 1]$. Trivial-looking, but dependency and high order. Assumes shallow min at 0.8.



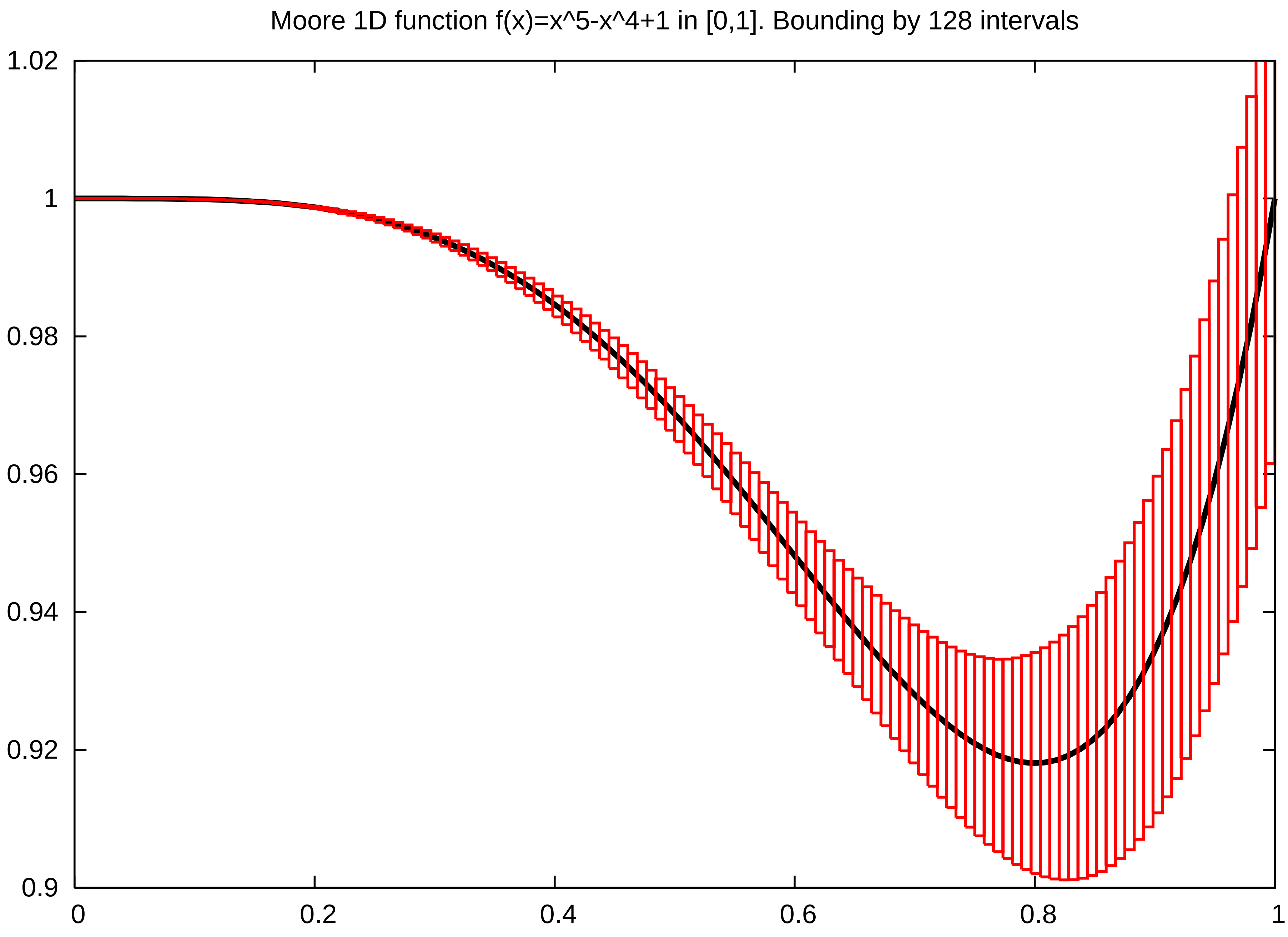

Moore 1D function f(x)=x^5-x^4+1 in [0,1]
1.02
1
0.98
0.96
0.94
0.92
0.9
0
0.2
0.4
0.6
0.8
1

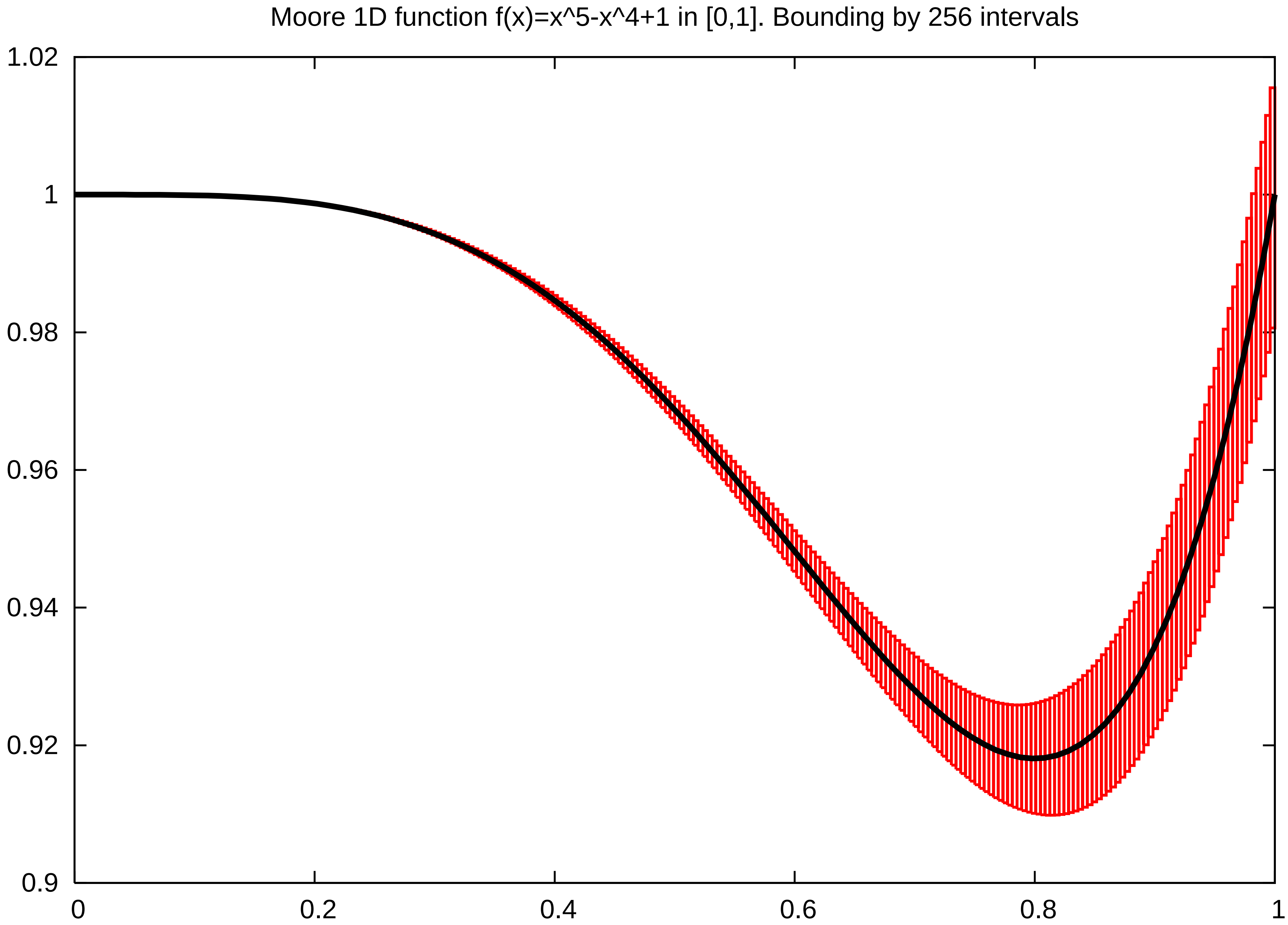


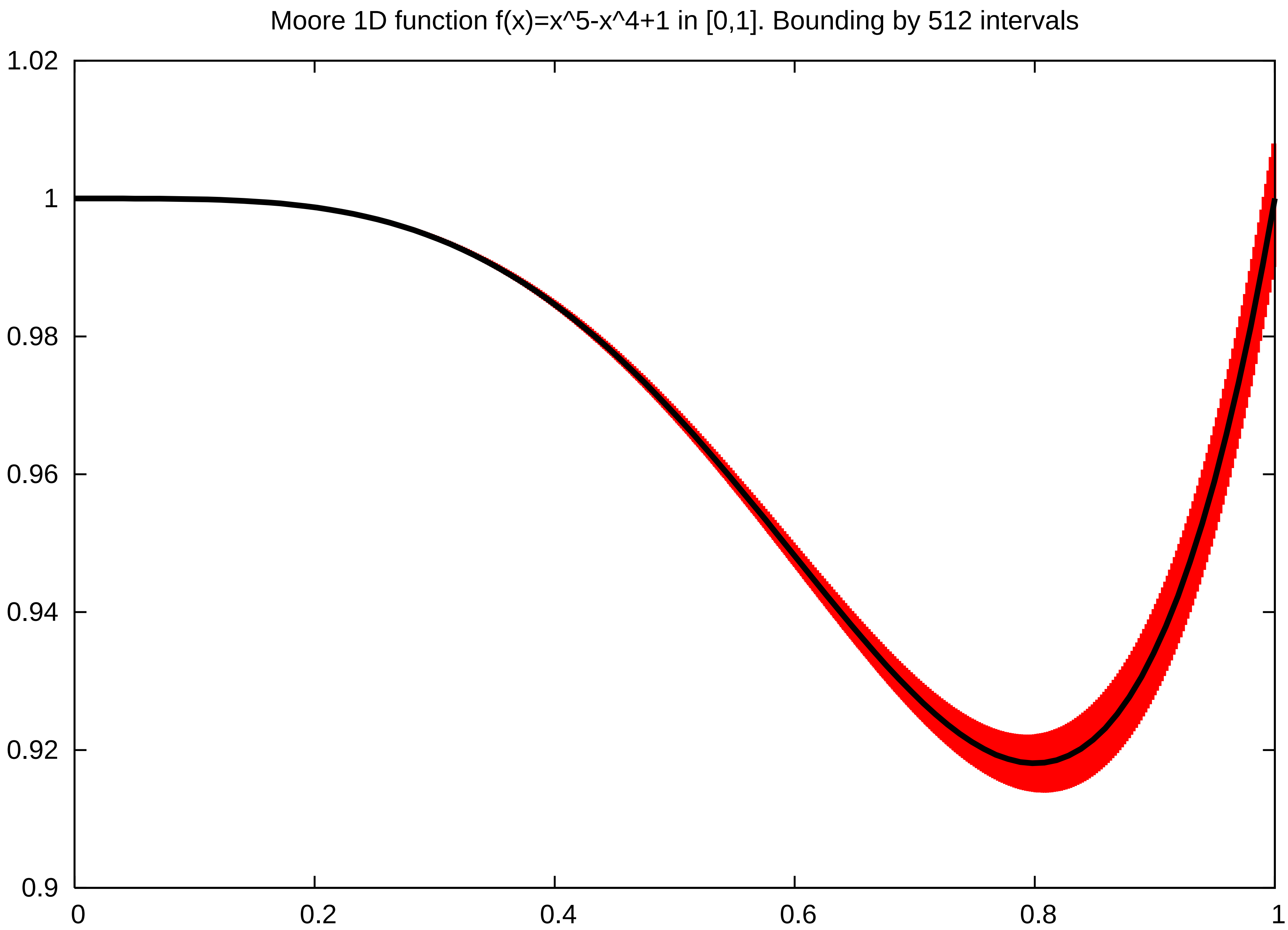
Moore 1D function f(x)=x^5-x^4+1 in [0,1]. Bounding by 16 intervals
1.3
1.2
1.1
1
0.9
0.8
0.7
0
0.2
0.4
0.6
0.8
1



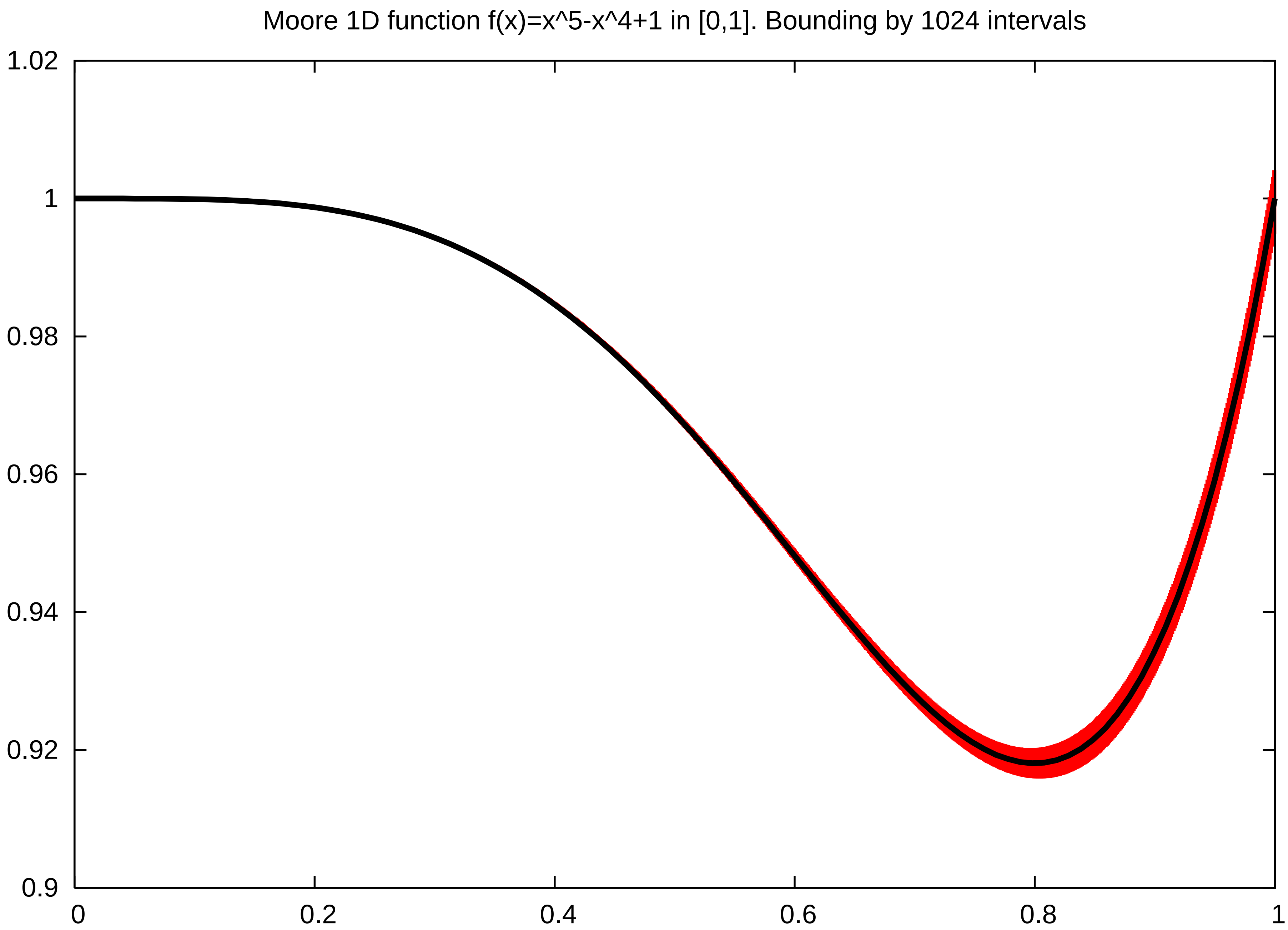
Moore 1D function f(x)=x^5-x^4+1 in [0,1]. Bounding by 128 intervals
1.3
1.2
1.1
1
0.9
0.8
0.7
0
0.2
0.4
0.6
0.8
1


Moore 1D function f(x)=x^5-x^4+1 in [0,1]. Bounding by 128 intervals
1.02
1
0.98
0.96
0.94
0.92
0.9
0
0.2
0.4
0.6
0.8
1


Moore 1D function f(x)=x^5-x^4+1 in [0,1]. Bounding by 256 intervals
1.02
1
0.98
0.96
0.94
0.92
0.9
0
0.2
0.4
0.6
0.8
1



Moore 1D function f(x)=x^5-x^4+1 in [0,1]. Bounding by 512 intervals
1.02
1
0.98
0.96
0.94
0.92
0.9
0
0.2
0.4
0.6
0.8
1


Moore 1D function f(x)=x^5-x^4+1 in [0,1]. Bounding by 1024 intervals
1.02
1
0.98
0.96
0.94
0.92
0.9
0
0.2
0.4
0.6
0.8
1

Definitions - Taylor Models and Operations

We begin with a review of the definitions of the basic operations.

Definition (Taylor Model) Let $f : D \subset R^v \to R$ be a function that is $(n+1)$ times continuously partially differentiable on an open set containing the domain v-dimensional domain D. Let x_0 be a point in D and P the n-th order Taylor polynomial of f around x_0. Let I be an interval such that

$$f(x) \in P(x - x_0) + I \text{ for all } x \in D.$$

Then we call the pair (P, I) an n-th order Taylor model of f around x_0 on D.

Definition (Addition and Multiplication) Let $T_{1,2} = (P_{1,2}, I_{1,2})$ be n-th order Taylor models around x_0 over the domain D. We define

$$T_1 + T_2 = (P_1 + P_2, I_1 + I_2)$$
$$T_1 \cdot T_2 = (P_{1\cdot 2}, I_{1\cdot 2})$$

where $P_{1\cdot 2}$ is the part of the polynomial $P_1 \cdot P_2$ up to order n and

$$I_{1\cdot 2} = B(P_e) + B(P_1) \cdot I_2 + B(P_2) \cdot I_1 + I_1 \cdot I_2$$

where P_e is the part of the polynomial $P_1 \cdot P_2$ of orders $(n + 1)$ to $2n$, and $B(P)$ denotes a bound of P on the domain D. We demand that $B(P)$ is at least as sharp as direct interval evaluation of $P(x - x_0)$ on D.

Definitions - Taylor Model Intrinsics

Definition (Intrinsic Functions of Taylor Models) Let $T = (P, I)$ be a Taylor model of order n over the v-dimensional domain $D = [a, b]$ around the point x_0. We define intrinsic functions for the Taylor models by performing various manipulations that will allow the computation of Taylor models for the intrinsics from those of the arguments. In the following, let $f(x) \in P(x - x_0) + I$ be any function in the Taylor model, and let $c_f = f(x_0)$, and $\bar{f}$ be defined by $\bar{f}(x) = f(x) - c_f$. Likewise we define $\bar{P}$ by $\bar{P}(x - x_0) = P(x - x_0) - c_f$, so that $(\bar{P}, I)$ is a Taylor model for $\bar{f}$. For the various intrinsics, we proceed as follows.

Exponential. We first write

$$\begin{aligned}\exp(f(x)) &= \exp\left(c_f + \bar{f}(x)\right) = \exp(c_f) \cdot \exp\left(\bar{f}(x)\right) \\ &= \exp(c_f) \cdot \Big\{1 + \bar{f}(x) + \frac{1}{2!}(\bar{f}(x))^2 + \cdots + \frac{1}{k!}(\bar{f}(x))^k \\ &\qquad + \frac{1}{(k+1)!}(\bar{f}(x))^{k+1} \exp\left(\theta \cdot \bar{f}(x)\right)\Big\},\end{aligned}$$

where $0 < \theta < 1$.

Definitions - Taylor Model Exponential, cont.

Taking $k \geq n$, the part

$$\exp(c_f) \cdot \left\{ 1 + \bar{f}(x) + \frac{1}{2!}(\bar{f}(x))^2 + \cdots + \frac{1}{n!}(\bar{f}(x))^n \right\}$$

is merely a polynomial of $\bar{f}$, of which we can obtain the Taylor model via Taylor model addition and multiplication. The remainder part of $\exp(f(x))$, the expression

$$\exp(c_f) \cdot \left\{ \frac{1}{(n+1)!}(\bar{f}(x))^{n+1} \right.$$
$$\left. + \cdots + \frac{1}{(k+1)!}(\bar{f}(x))^{k+1} \exp\left(\theta \cdot \bar{f}(x)\right) \right\},$$

will be bounded by an interval. First observe that since the Taylor polynomial of $\bar{f}$ does not have a constant part, the $(n+1)$-st through $(k+1)$-st powers of the Taylor model $(\bar{P}, I)$ of $\bar{f}$ will have vanishing polynomial part, and thus so does the entire remainder part. The remainder bound interval for the Lagrange remainder term

Definitions - Taylor Model Exponential, cont.

$$\exp(c_f)\frac{1}{(k+1)!}(\bar{f}(x))^{k+1}\exp\left(\theta\cdot\bar{f}(x)\right)$$

can be estimated because, for any $x \in D$, $\bar{P}(x-x_0) \in B(\bar{P})$, and $0 < \theta < 1$, and so

$$\begin{aligned}(\bar{f}(x))^{k+1}\exp\left(\theta\cdot\bar{f}(x)\right) \in \left(B(\bar{P})+I\right)^{k+1} \\ \times\exp\left([0,1]\cdot(B(\bar{P})+I)\right).\end{aligned}$$

The evaluation of the "exp" term is mere standard interval arithmetic. In the actual implementation, one may choose $k=n$ for simplicity, but it is not a priori clear which value of k would yield the sharpest enclosures.

Definitions - Taylor Model Arc Sine

Arcsine. Under the condition $\forall x \in D,\ B(P(x-x_0)+I) \subset (-1,1)$, using an addition formula for the arcsine, we re-write

$$\arcsin(f(x)) = \arcsin(c_f) + \arcsin\left(f(x)\cdot\sqrt{1-c_f^2} - c_f\cdot\sqrt{1-(f(x))^2}\right).$$

Utilizing that

$$g(x) \equiv f(x)\cdot\sqrt{1-c_f^2} - c_f\cdot\sqrt{1-(f(x))^2}$$

does not have a constant part, we have

$$\arcsin(g(x)) = g(x) + \frac{1}{3!}(g(x))^3 + \frac{3^2}{5!}(g(x))^5 + \frac{3^2\cdot 5^2}{7!}(g(x))^7$$
$$+\cdots+\frac{1}{(k+1)!}(g(x))^{k+1}\cdot\arcsin^{(k+1)}(\theta\cdot g(x)),$$

where

$$\arcsin'(a) = 1/\sqrt{1-a^2}, \qquad \arcsin''(a) = a/(1-a^2)^{3/2},$$
$$\arcsin^{(3)}(a) = (1+2a^2)/(1-a^2)^{5/2}, \ldots$$

Definitions - Taylor Model Arc Sine, Antiderivation

A recursive formula for the higher order derivatives of arcsin

$$\arcsin^{(k+2)}(a) = \frac{1}{1-a^2}\{(2k+1)a\arcsin^{(k+1)}(a) + k^2\arcsin^{(k)}(a)\}$$

is useful. Then, evaluating in Taylor model arithmetic yields the desired result, where again the terms involving θ only produce interval contributions.

Antiderivation. We note that a Taylor model for the integral with respect to variable i of a function f can be obtained from the Taylor model (P, I) of the function by merely integrating the part P_{n-1} of order up to $n-1$ of the polynomial, and bounding the n-th order into the new remainder bound. Specifically, we have

$$\partial_i^{-1}(P, I) = \left(\int_0^{x_i} P_{n-1}(x)dx_i\,,\ (B(P - P_{n-1}) + I)\cdot(b_i - a_i)\right).$$

Thus, given a Taylor model for a function f, the Taylor model intrinsic functions produce a Taylor models for the composition of the respective intrinsic with f. Furthermore, we have the following result.

TM Scaling Theorem

Theorem (Scaling Theorem) Let $f,\ g \in C^{n+1}(D)$ and $(P_{f,h}, I_{f,h})$ and $(P_{g,h}, I_{g,h})$ be n-th order Taylor models for f and g around x_h on $x_h + [-h, h]^v \subset D$. Let the remainder bounds $I_{f,h}$ and $I_{g,h}$ satisfy $I_{f,h} = O(h^{n+1})$ and $I_{g,h} = O(h^{n+1})$. Then the Taylor models $(P_{f+g}, I_{f+g,h})$ and $(P_{f\cdot g}, I_{f\cdot g,h})$ for the sum and products of f and g obtained via addition and multiplication of Taylor models satisfy

$$I_{f+g,h} = O(h^{n+1}),\ \text{and}\ I_{f\cdot g,h} = O(h^{n+1}).$$

Furthermore, let s be any of the intrinsic functions defined above, then the Taylor model $(P_{s(f)}, I_{s(f),h})$ for $s(f)$ obtained by the above definition satisfies

$$I_{s(f),h} = O(h^{n+1}).$$

We say the Taylor model arithmetic has the $(n+1)$-st order scaling property.

Proof. The proof for the binary operations follows directly from the definition of the remainder bounds for the binaries. Similarly, the proof for the intrinsics follows because all intrinsics are composed of binary operations as well as an additional interval, the width of which scales at least with the $(n+1)$-st power of a bound B of a function that scales at least linearly with h.

Fundamental Theorem of TM Arithmetic

The scaling theorem states that a given function f can be approximated by P with an error that scales with order $(n+1)$. Common mathematical jargon. But in interval community, a related but different meaning of scaling exists, namely the behavior of the overestimation of a given method to determine the range of a function.

Theorem (FTTMA, Fundamental Theorem of TM Arithmetic) Let the function $f: R^v \rightarrow R^v$ be described by a multivariate Taylor model $P_f + I_f$ over the domain $D \subset R^v$. Let the function $g: R^v \rightarrow R$ be given by a code list comprised of finitely many elementary operations and intrinsic functions, and let g be defined over the range of the Taylor model $P_f, +I_f$. Let $P+I$ be the Taylor model obtained by executing the code list for g, beginning with the Taylor model $P_f + I_f$. Then $P+I$ is a Taylor model for $g \circ f$.

Furthermore, if the Taylor model of f has the $(n+1)$-st order scaling property, so does the resulting Taylor model for g.

Proof. Induction over code list.

Example: Consider f with $f(x) = \sin^2(\exp(x+1)) + \cos^2(\exp(x+1))$. We know $f(x) = 1$, but validated methods don't.

Implementation of TM Arithmetic

Validated Implementation of TM Arithmetic exists. The following points are important

- Strict requirements for **underlying FP arithmetic**
- Taylor models require cutoff threshold (**garbage collection**)
- Coefficients remain FP, not intervals
- Package quite **extensively tested** by Corliss et al.

For practical considerations, the following is important:

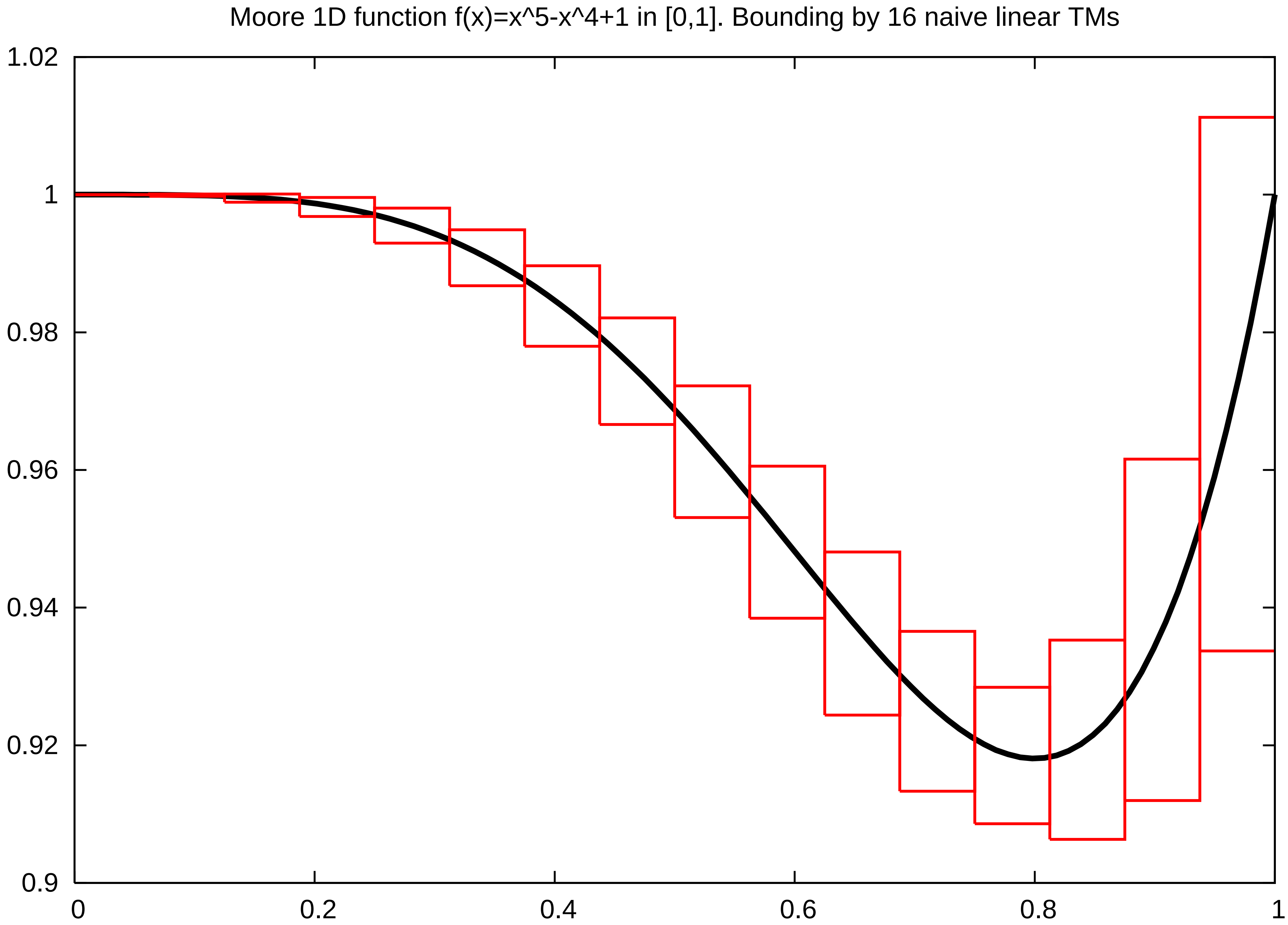
- Need **sparsity** support
- Need efficient coefficient **addressing** scheme
- About $50,000$ lines of code
- **Language Independent** Platform, coexistence in F77, C, F90, C++

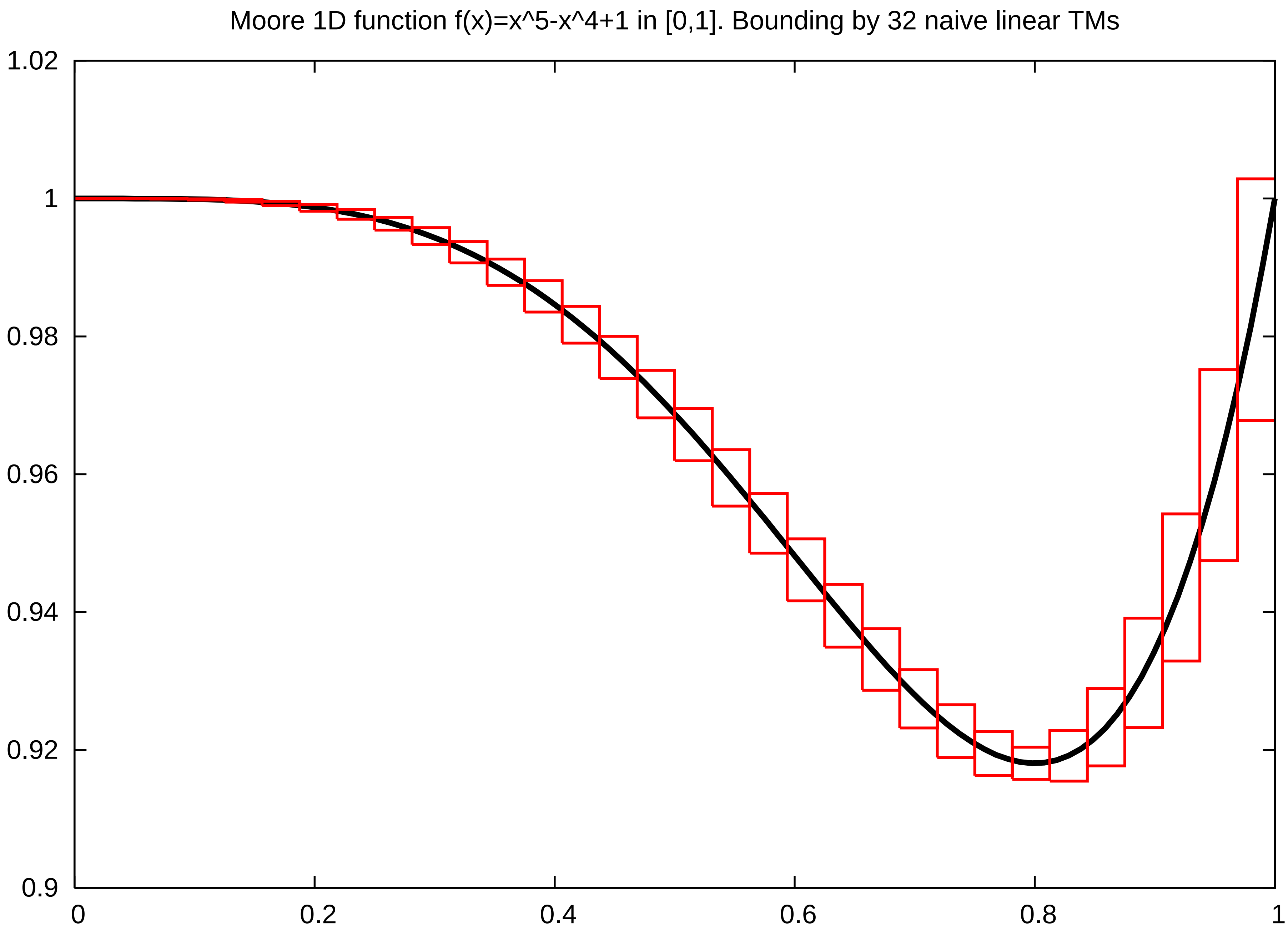
Important TM Algorithms

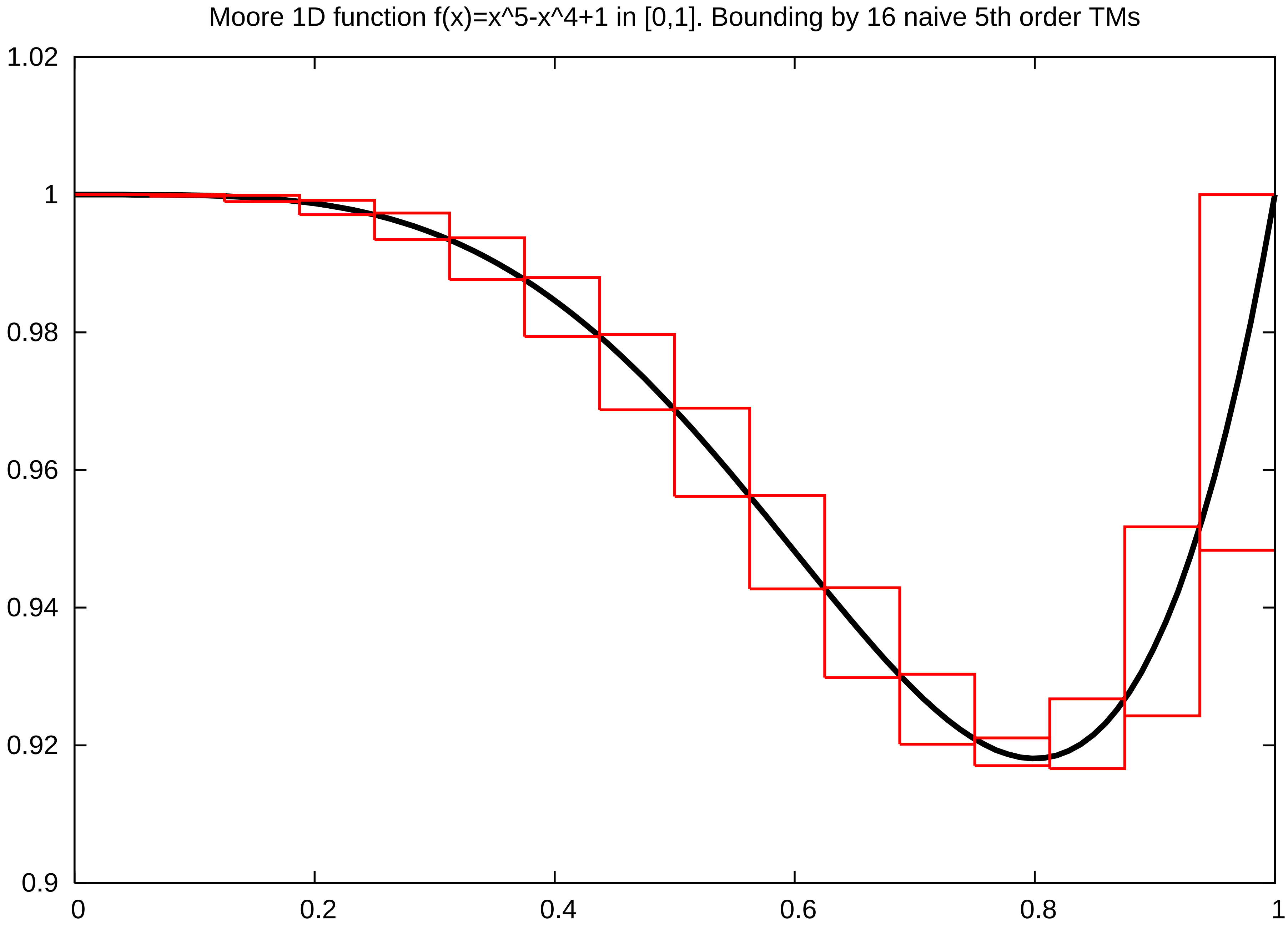
- **Range Bounding** (Evaluate f as TM, bound polynomial, add remainder bound. LDB, QFB etc.)
- **Global Optimization** (Use TM bounding schemes, obtain good cutoff values quickly by using non-verified schemes)
- **Quadrature** (Evaluate f as TM, integrate polynomial and remainder bound)
- **Implicit Equations** (Obtain TMs for implicit solutions of TM equations)
- **Superconvergent** Interval Newton Method (Application of Implicit Equations)
- **Implicit ODEs and DAEs**
- **Complex Arithmetic**
- **ODEs** (Obtain TMs describing dependence of final coordinates on initial coordinates)

Important TM Algorithms

- **Range Bounding** (Evaluate f as TM, bound polynomial, add remainder bound. LDB, QFB etc.)
- **Global Optimization** (Use TM bounding schemes, obtain good cutoff values quickly by using non-verified schemes)
- **Quadrature** (Evaluate f as TM, integrate polynomial and remainder bound)
- **Implicit Equations** (Obtain TMs for implicit solutions of TM equations)
- **Superconvergent** Interval Newton Method (Application of Implicit Equations)
- **Implicit ODEs and DAEs**
- **Complex Arithmetic**
- **ODEs** (Obtain TMs describing dependence of final coordinates on initial coordinates)


Moore 1D function f(x)=x^5-x^4+1 in [0,1]. Bounding by 16 naive linear TMs
1.02
1
0.98
0.96
0.94
0.92
0.9
0
0.2
0.4
0.6
0.8
1


Moore 1D function f(x)=x^5-x^4+1 in [0,1]. Bounding by 32 naive linear TMs
1.02
1
0.98
0.96
0.94
0.92
0.9
0
0.2
0.4
0.6
0.8
1

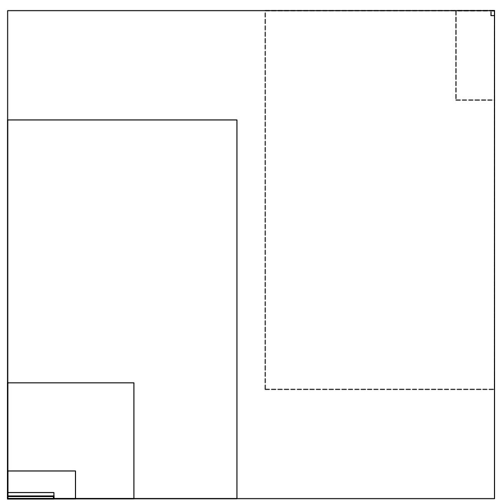

Moore 1D function f(x)=x^5-x^4+1 in [0,1]. Bounding by 16 naive 5th order TMs
1.02
1
0.98
0.96
0.94
0.92
0.9
0
0.2
0.4
0.6
0.8
1

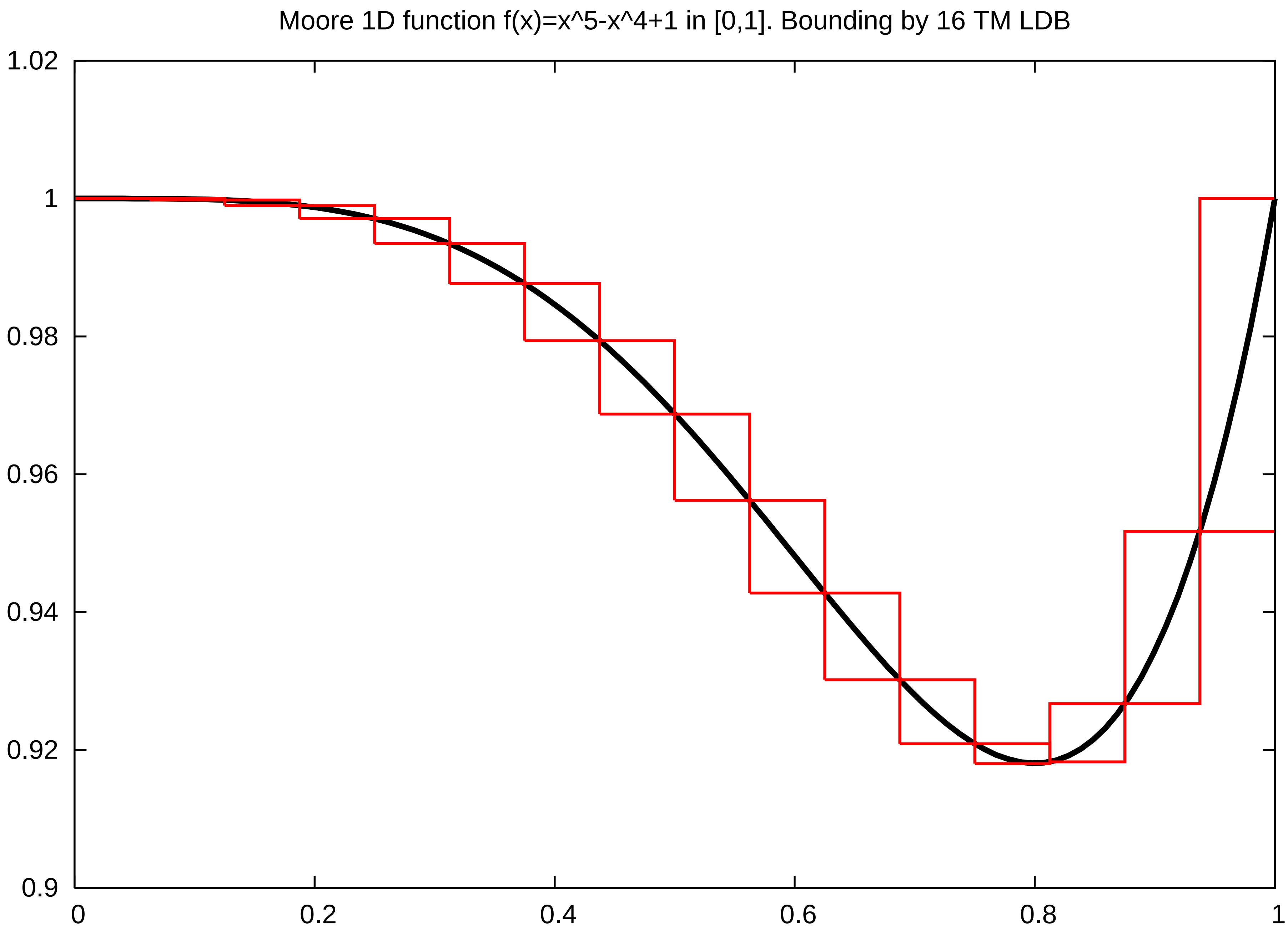
Important TM Algorithms

- **Range Bounding** (Evaluate f as TM, bound polynomial, add remainder bound. LDB, QFB etc.)
- **Global Optimization** (Use TM bounding schemes, obtain good cutoff values quickly by using non-verified schemes)
- **Quadrature** (Evaluate f as TM, integrate polynomial and remainder bound)
- **Implicit Equations** (Obtain TMs for implicit solutions of TM equations)
- **Superconvergent** Interval Newton Method (Application of Implicit Equations)
- **Implicit ODEs and DAEs**
- **Complex Arithmetic**
- **ODEs** (Obtain TMs describing dependence of final coordinates on initial coordinates)

The Linear Dominated Bounder (LDB)

- The linear part of TM polynomial is the leading part, also for range bounding.
- The idea is easily extended to the multi-dimensional case.
- Use the linear part as a guideline for domain splitting and elimination.
- The reduction of the size of interested box works multi-dimensionally and automatically. Thus, the reduction rate is fast.
- Even there is no linear part in the original TM, by shifting the expansion point, normally the linear part is introduced.
- Exact bound (with rounding) is obtained if monotonic.





Moore 1D function f(x)=x^5-x^4+1 in [0,1]. Bounding by 16 TM LDB
1.02
1
0.98
0.96
0.94
0.92
0.9
0
0.2
0.4
0.6
0.8
1

Important TM Algorithms

- **Range Bounding** (Evaluate f as TM, bound polynomial, add remainder bound. LDB, QFB etc.)
- **Global Optimization** (Use TM bounding schemes, obtain good cutoff values quickly by using non-verified schemes)
- **Quadrature** (Evaluate f as TM, integrate polynomial and remainder bound)
- **Implicit Equations** (Obtain TMs for implicit solutions of TM equations)
- **Superconvergent** Interval Newton Method (Application of Implicit Equations)
- **Implicit ODEs and DAEs**
- **Complex Arithmetic**
- **ODEs** (Obtain TMs describing dependence of final coordinates on initial coordinates)

The QFB (Quadratic Fast Bounder) Algorithm

The most critical case in global optimization is to obtain a good lower bound near local minimizer.

Let $P + I$ be a given Taylor model in the domain D, and let P have a positive definite Hessian H. Decompose the Taylor model as

$$P + I = (P - Q) + I + Q$$

and observe

$$l(P + I) \geq l(P - Q) + l(Q) + l(I).$$

Choose Q quadratic such that

$$Q_{x_0} = \frac{1}{2}(x - x_0)^t \cdot H \cdot (x - x_0) \text{ with } x_0 \in D.$$

Then $l(Q_{x_0}) = 0$, and $(P - Q)$ does not contain pure quadratic terms.

If x_0 is the minimizer of quadratic part of P on D, then x_0 is also the minimizer of linear part of $(P - Q_{x_0})$, due to the Kuhn-Tucker conditions. Furthermore, the lower bound of $(P - Q_{x_0})$, when evaluated with plain interval evaluation, is accurate to order 3 of the original domain box.

Remark: The closer x_0 is to the minimizer, the closer there is order 3 cutoff. $\rightarrow$ Determine a sequence $x^{(n)}$ of candidates for x_0 in a "feasible descent direction."

Quadratic Pruning - The Idea

Extract a convex quadratic part P_2 of Taylor model, write

$$f(x) \in P_2(x) + R(x) + I \text{ where}$$
$$P_2(x) = \frac{1}{2} x^t \cdot H \cdot x$$

Want to confine the region $P_2(x) \leq a$ with $a > 0$, by an interval box $[-x_m, x_m]$ with $x_m > 0$.

Because of positive definiteness and convexity, this region is inside a **closed ellipsoidal contour surface** $P_2(x) = a$. The optimal confining interval box touches such a region at each box side surface tangentially, so the condition to find x_m is ∇f is normal to the corresponding box surface; namely for determining the k-th component x_{mk},

$$(\nabla P)_i = 0 \text{ for } \forall i \neq k.$$

Important TM Algorithms

- **Range Bounding** (Evaluate f as TM, bound polynomial, add remainder bound. LDB, QFB etc.)
- **Global Optimization** (Use TM bounding schemes, obtain good cutoff values quickly by using non-verified schemes)
- **Quadrature** (Evaluate f as TM, integrate polynomial and remainder bound)
- **Implicit Equations** (Obtain TMs for implicit solutions of TM equations)
- **Superconvergent** Interval Newton Method (Application of Implicit Equations)
- **Implicit ODEs and DAEs**
- **Complex Arithmetic**
- **ODEs** (Obtain TMs describing dependence of final coordinates on initial coordinates)

The TM based Global Optimizer, COSY-GO

has utilized various algorothms based on Taylor models.

- LDB (Linear Dominated Bounding) bounding and domain reduction
- QFB (Quadratic Fast Bounding) bounding and domain reduction for positive definate cases (Quadratic pruning)
- Various cutoff value update schemes

And, we have completed

- Adjustment to pallarel environments with low inter-processoer communication rate
- Restart capability
- Continuation of computations while the underlying arithmetic fails
- COSY INFINITY Version 9.0 has been released

And, what we are doing further...

- High-order derivative based box rejection and the domain reduction
- Supporting high multiple precision computations for TMs

Moore's Simple 1D Function

$$f(x) = 1 + x^5 - x^4$$

Study on $[0, 1]$. Trivial-looking, but dependency and high order. Assumes shallow min at 0.8.

Result of COSY-GO

Mode	Min Enclosure	Steps	Remained Volume	CPU s
LDB/QFB	$0.9180^{800000000021}_{799999999953}$	8	1.43e-7	0.004
LDB	$0.9180^{800000000048}_{799999999801}$	19	8.41e-7	0.010
naiveTM	$0.9180^{800000000284}_{799999997508}$	77	1.91e-6	0.013
IN	$0.9180^{800000000487}_{780468610746}$	13767	2.47e-3	1.702

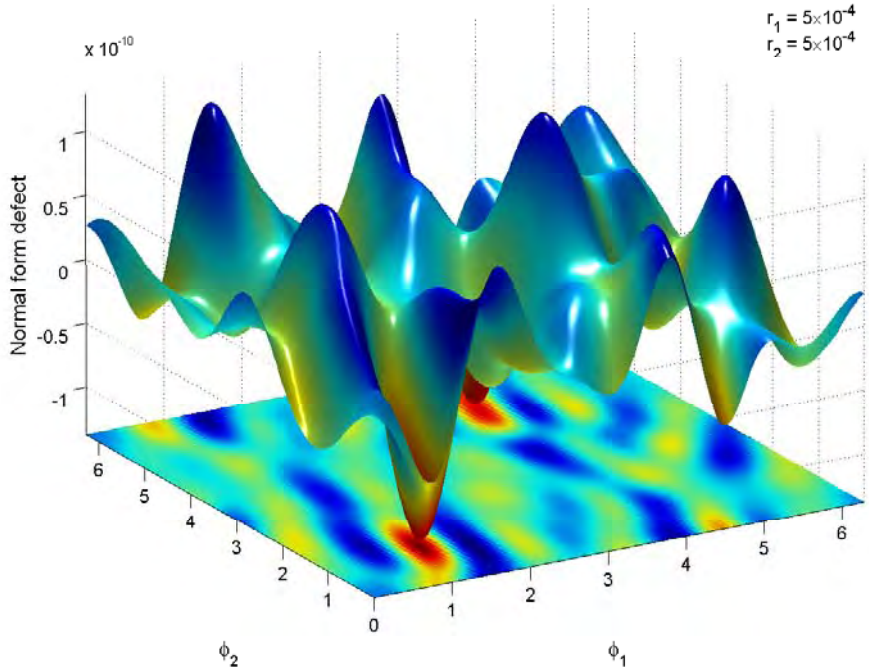


Fig. 9. Projection of the normal form defect function. Dependence on two angle variables for the fixed radii $r_1 = r_2 = 5 \cdot 10^{-4}$

Region	Boxes studied	CPU-time	Bound	Transversal Iterations
$[0.2, 0.4] \cdot 10^{-4}$	$82,930$	$30,603$ sec	$0.859 \cdot 10^{-13}$	$2.328\,3 \cdot 10^{8}$
$[0.4, 0.6] \cdot 10^{-4}$	$82,626$	$30,603$ sec	$0.587 \cdot 10^{-12}$	$3.\,407\,2 \cdot 10^{7}$
$[0.6, 0.9] \cdot 10^{-4}$	$64,131$	$14,441$ sec	$0.616 \cdot 10^{-11}$	$4.870\,1 \cdot 10^{6}$
$[0.9, 1.2] \cdot 10^{-4}$	$73,701$	$13,501$ sec	$0.372 \cdot 10^{-10}$	$8.064\,5 \cdot 10^{5}$
$[1.2, 1.5] \cdot 10^{-4}$	$106,929$	$24,304$ sec	$0.144 \cdot 10^{-9}$	$2.083\,3 \cdot 10^{5}$
$[1.5, 1.8] \cdot 10^{-4}$	$111,391$	$26,103$ sec	$0.314 \cdot 10^{-9}$	$0.95541 \cdot 10^{5}$

Table 8
Global bounds obtained for six radial regions in normal form space for the Tevatron. Also computed are the guaranteed minimum transversal iterations.

Applications

There are so many problems requiring optimizations.
Using COSY-GO, we have worked on

- Numerous challenging benchmark tests
- Design parameter optimizations
- Rump's Toeplits problems
- Entropy estimates for chaotic dynamical systems
- Long-term stability estimates of the Tevatron
- Molecule packing problems
- Gravity assist interplanetary spacecraft trajectory designs

And more are, and will be, coming.

- Edge curvature design for FFAG magnets
- Complicated field computaions for beam transfer maps
- ... Any problem you can imagine...

Important TM Algorithms

- **Range Bounding** (Evaluate f as TM, bound polynomial, add remainder bound. LDB, QFB etc.)
- **Global Optimization** (Use TM bounding schemes, obtain good cutoff values quickly by using non-verified schemes)
- **Quadrature** (Evaluate f as TM, integrate polynomial and remainder bound)
- **Implicit Equations** (Obtain TMs for implicit solutions of TM equations)
- **Superconvergent** Interval Newton Method (Application of Implicit Equations)
- **Implicit ODEs and DAEs**
- **Complex Arithmetic**
- **ODEs** (Obtain TMs describing dependence of final coordinates on initial coordinates)

ODE Integration with Taylor Models

Idea: retain full **dependence on initial conditions** as Taylor model (Non-verified version: big breakthrough in particle optics and beam physics, 1984 - allows to calculate "aberrations" to any order, from earlier order three)

1. Different from other validated methods, the approach is **single step** - no need for a separate coarse enclosure and subsequent verification step
2. Error due to **time stepping** is $O(n_t + 1)$
3. Error due to **initial variables** is $O(n_v + 1)$, **not** $O(2)$ as in other methods
4. By choosing n_t and n_v appropriately, the error due to finite domain and time stepping can be made **arbitrarily small.**
5. Overall, **never** leave the TM represenation until possibly the very end. Doing so may remove higher order dependence.

Old Taylor Model based Integrators (–2004)

- High order expansion not only in time t but also in transversal variables $\vec{x}$.
- Capability of weighted order computation, allowing to suppress the expansion order in transversal variables $\vec{x}$.
- Shrink wrapping algorithm including blunting to control ill-conditioned cases.
- Pre-conditioning algorithms based on the Curvilinear, QR decomposition, and blunting pre-conditioners.

Taylor Model based Integrator COSY-VI version 3 (2007-)

- More economical one time step integration using the reference trajectory and the Lie derivative based flow operator on the deviation equations.
- Non aborting mechanism when prohibited arithmetic happens such as $1/f$ for $0 \in f$.
- Improvement of step size control.
- Error parametrization of Taylor models.
- Dynamic domain decomposition.

The Volterra Equation

Describe dynamics of two conflicting populations

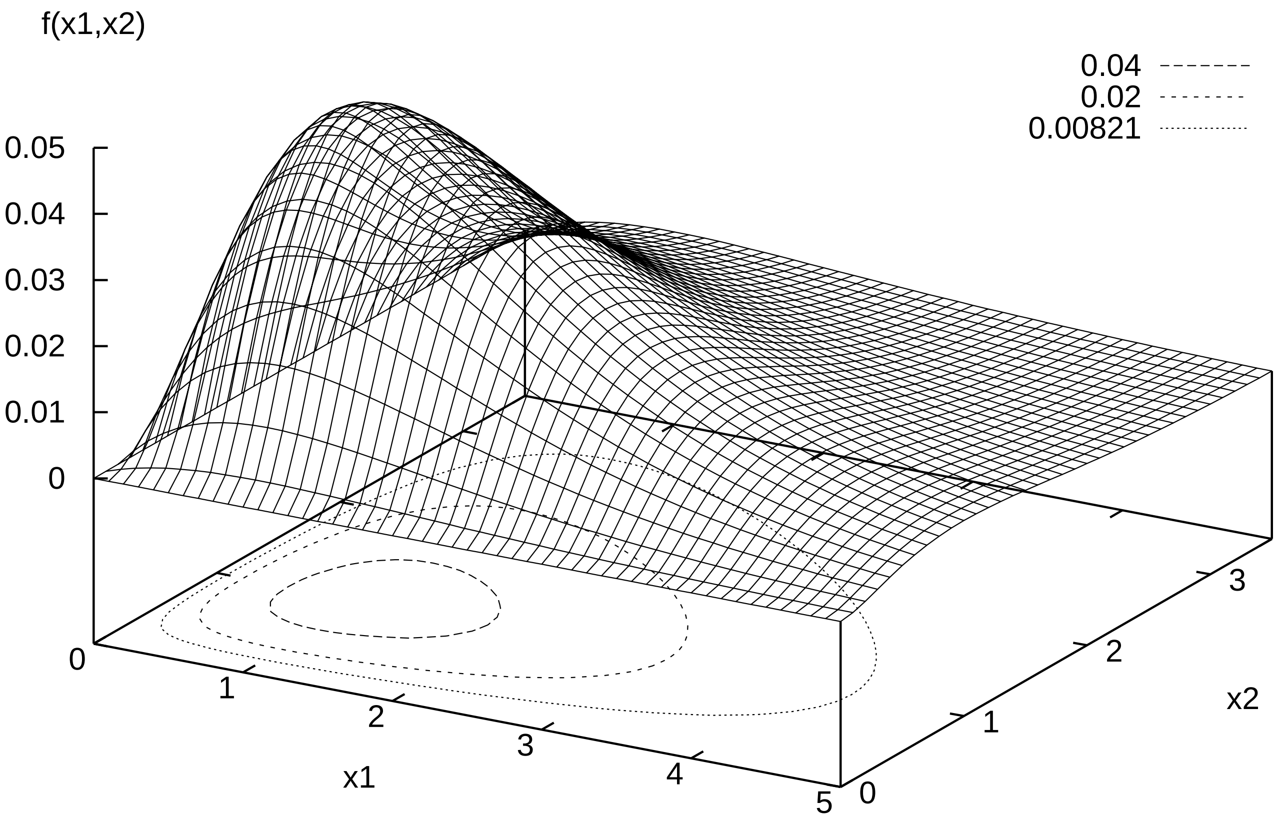
$$\frac{dx_1}{dt} = 2x_1(1 - x_2), \quad \frac{dx_2}{dt} = -x_2(1 - x_1)$$

Interested in initial condition

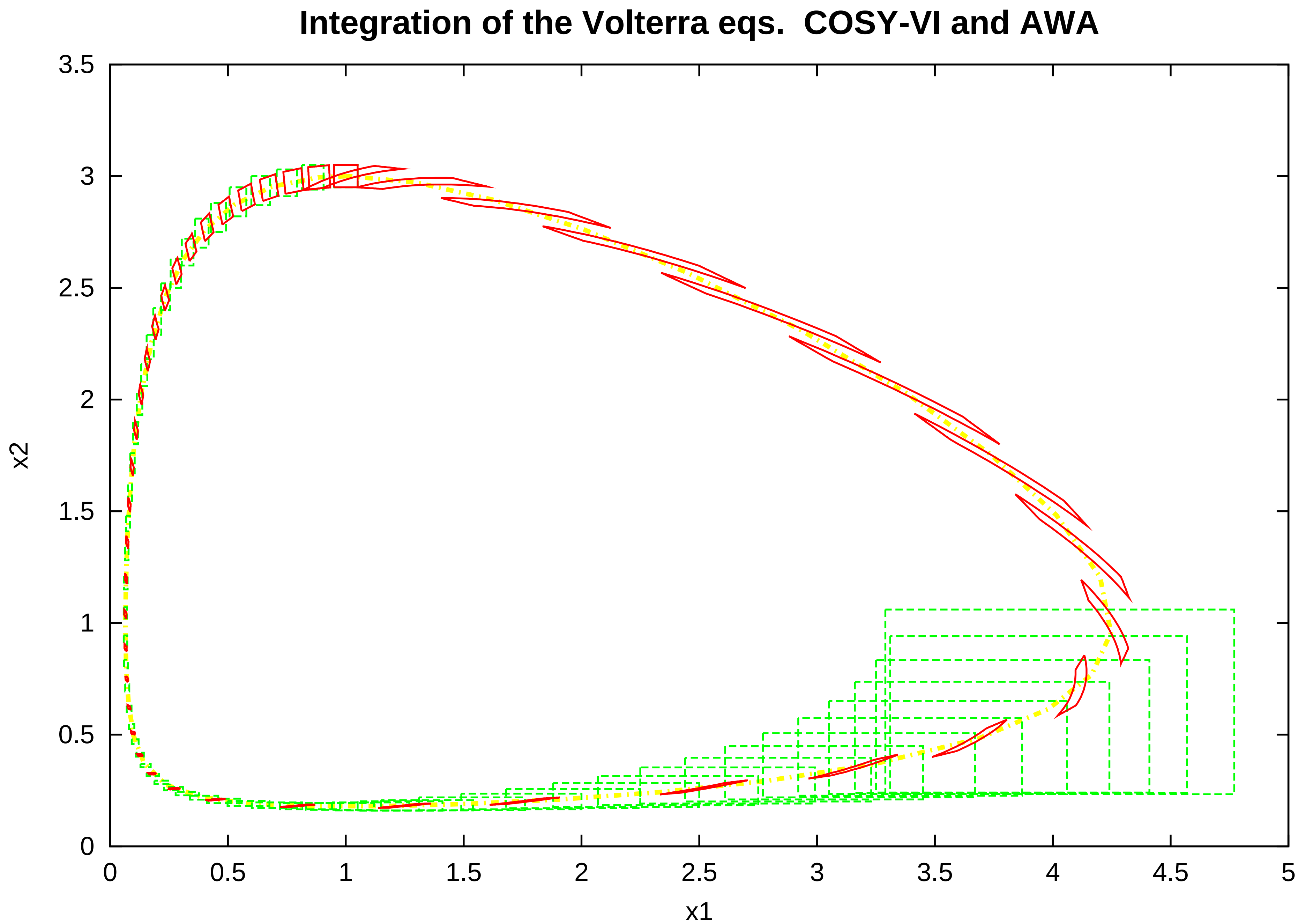
$$x_{01} \in 1 + [-0.05, 0.05], \quad x_{02} \in 3 + [-0.05, 0.05] \quad \text{at } t = 0.$$

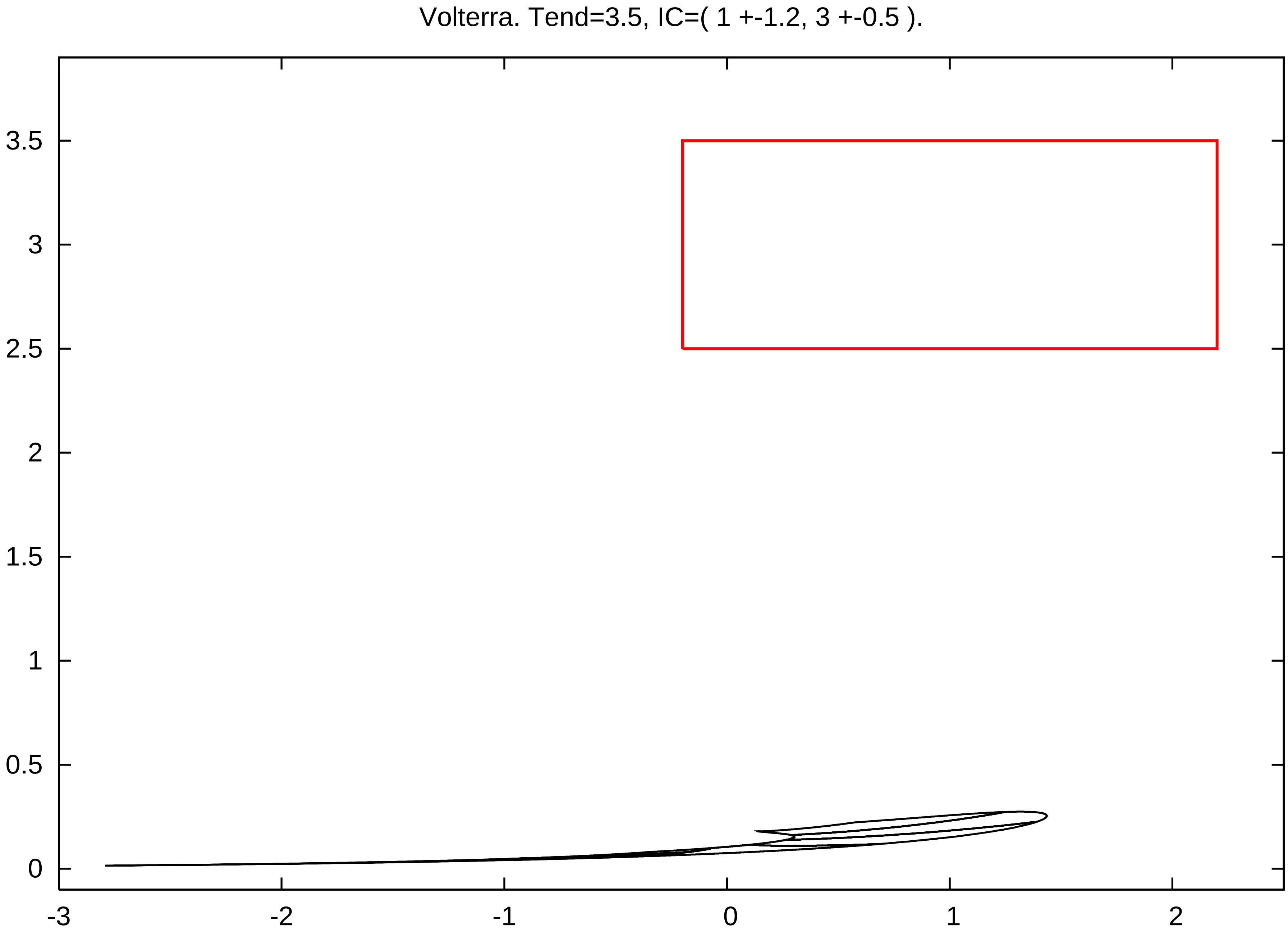
Satisfies constraint condition

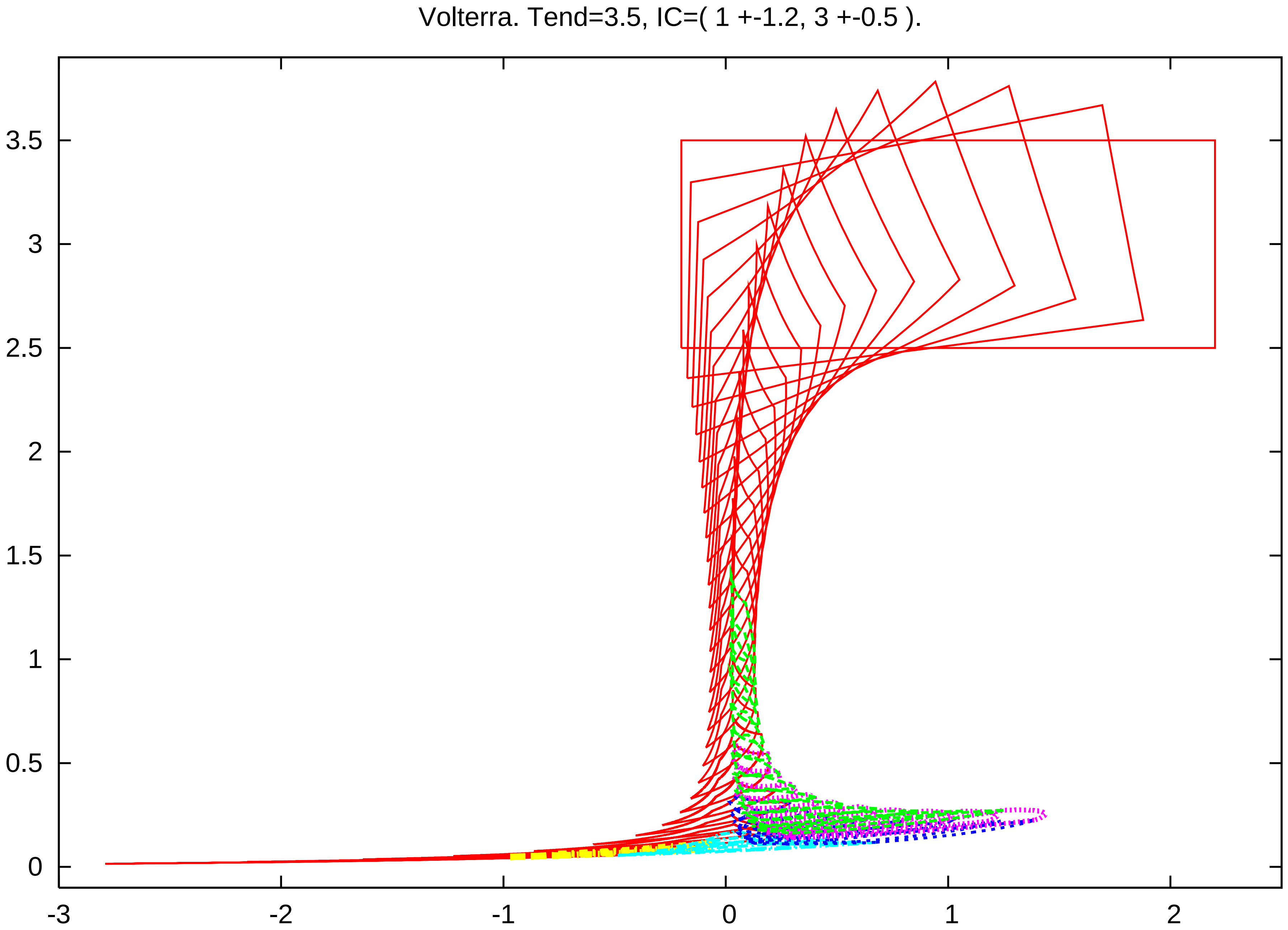
$$C(x_1, x_2) = x_1 x_2^2 e^{-x_1 - 2x_2} = \text{Constant}$$



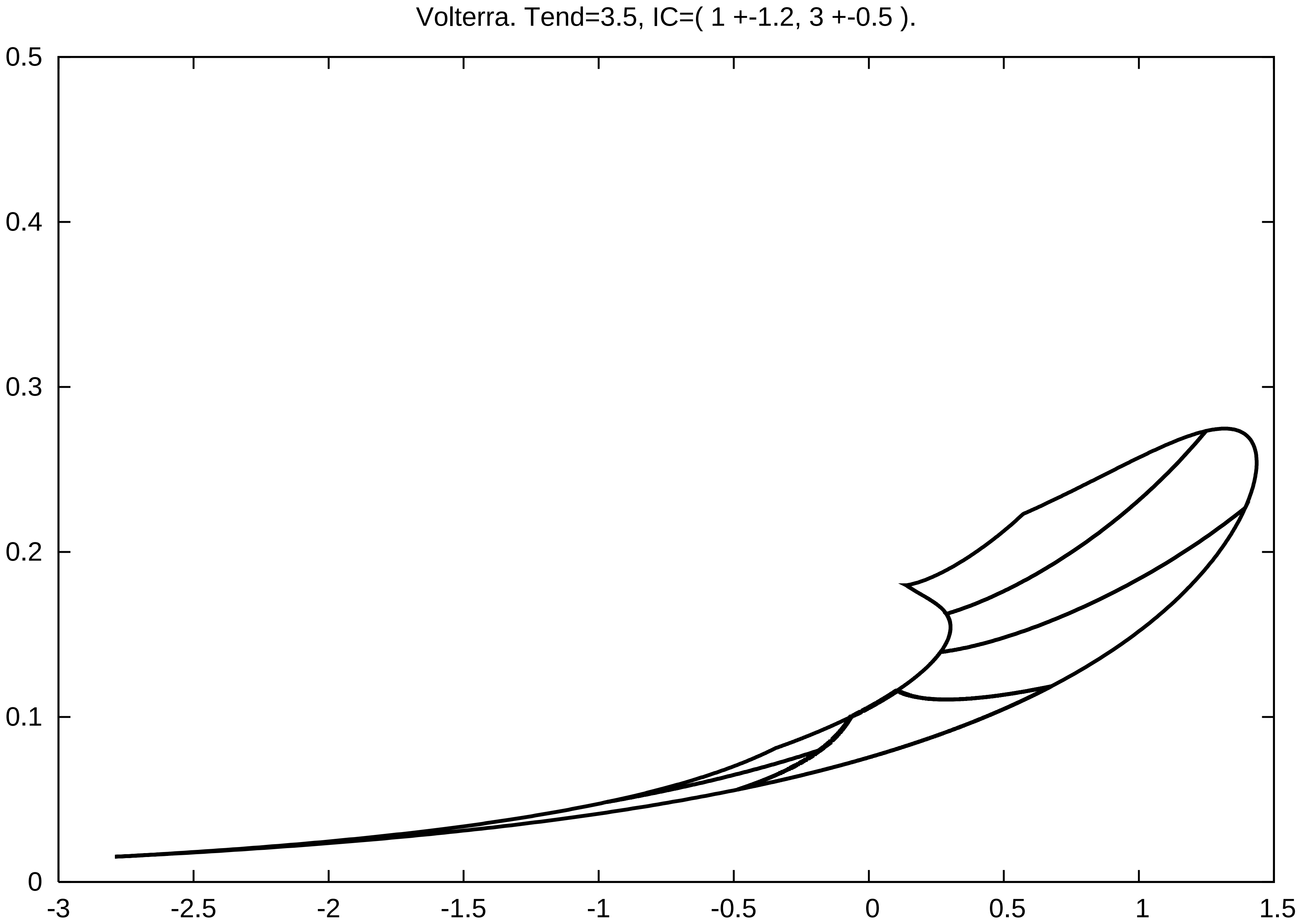
f(x1,x2)
0.04
0.02
0.00821
0.05
0.04
0.03
0.02
0.01
0
0
1
2
3
4
5
x1
0
1
2
3
x2

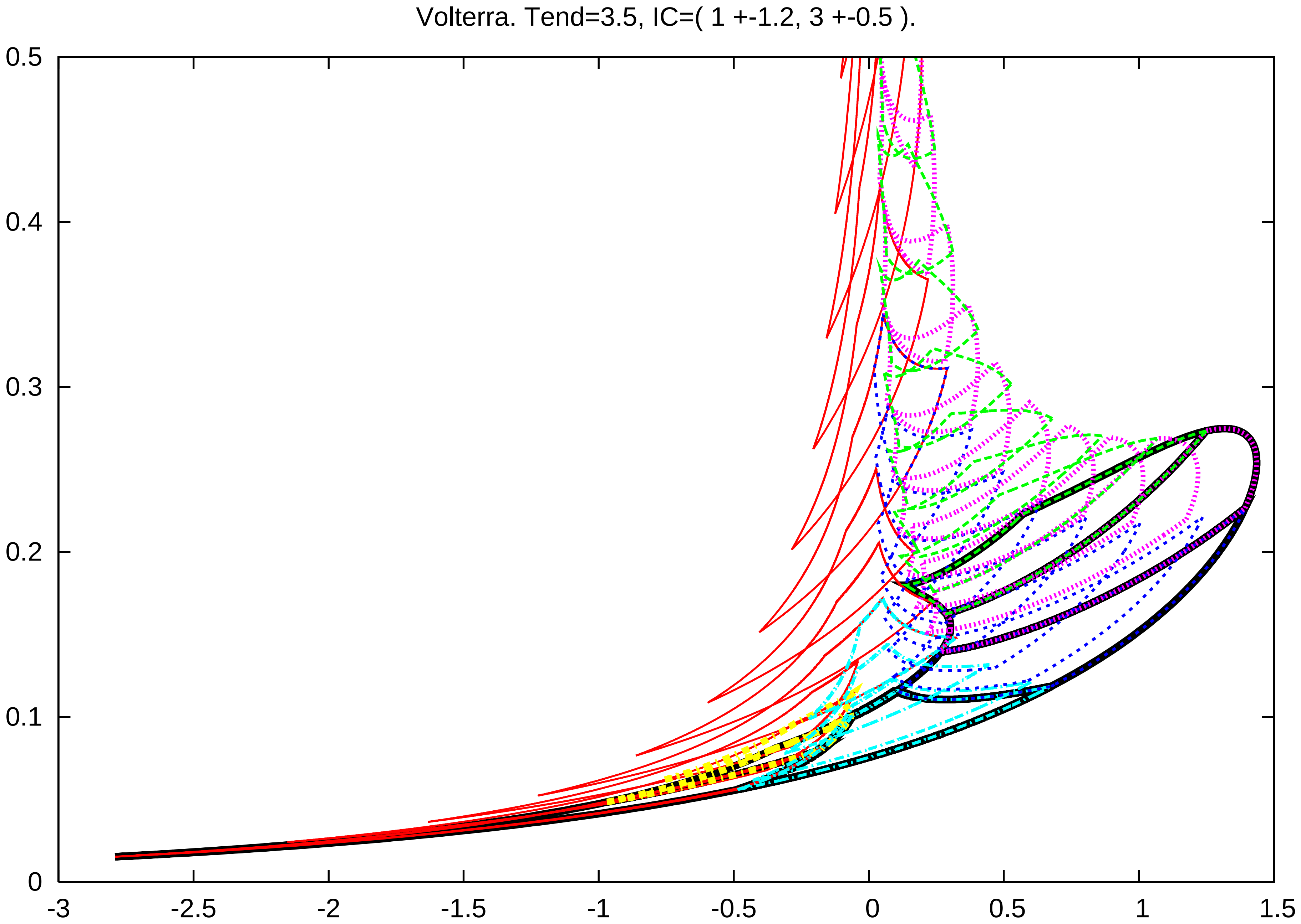

Integration of the Volterra eqs. COSY-VI and AWA
x1
x2
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
0.5
1
1.5
2
2.5
3
3.5

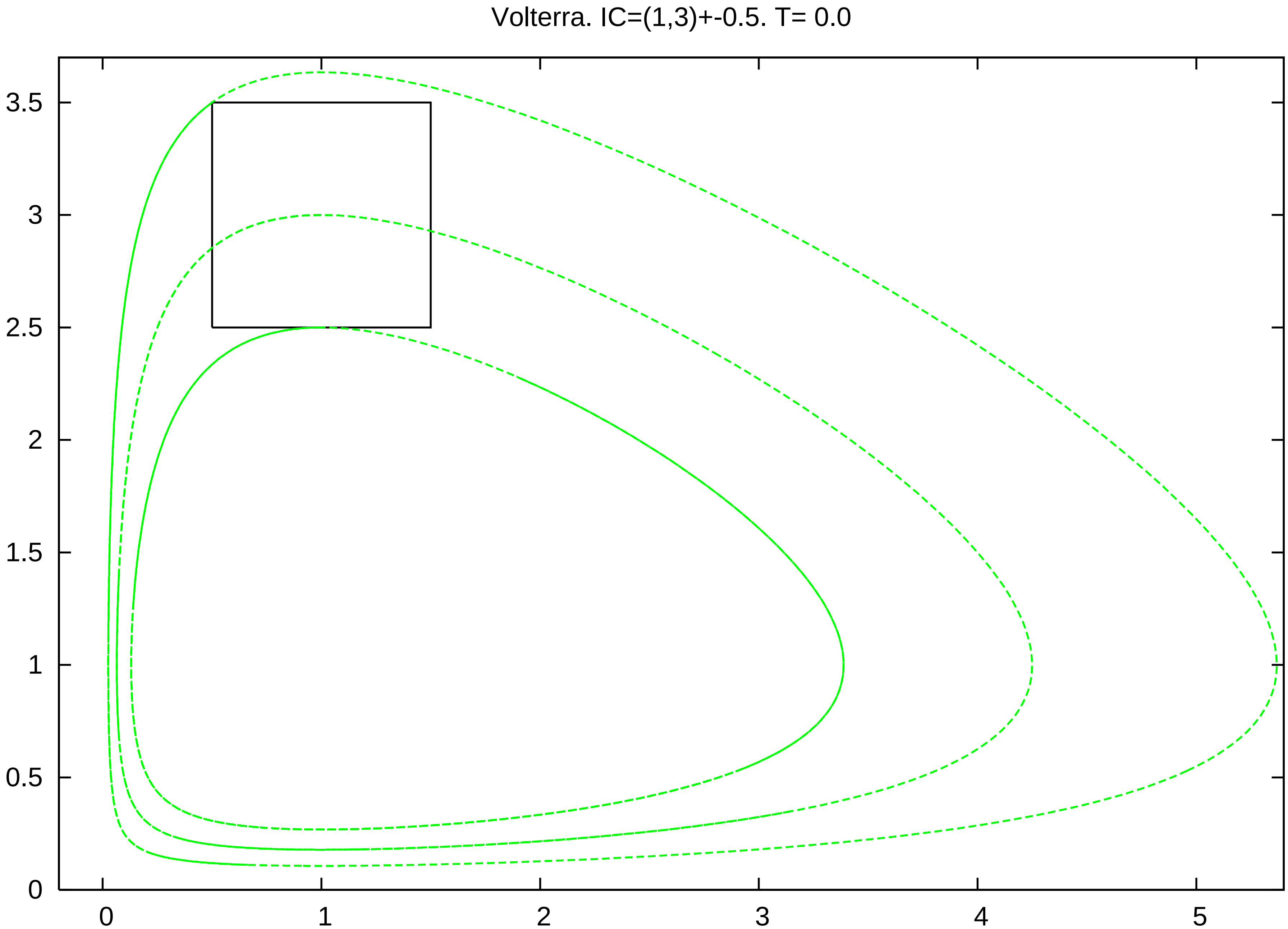

Volterra. Tend=3.5, IC=(1 +-1.2, 3 +-0.5).
3.5
3
2.5
2
1.5
1
0.5
0
-3
-2
-1
0
1
2

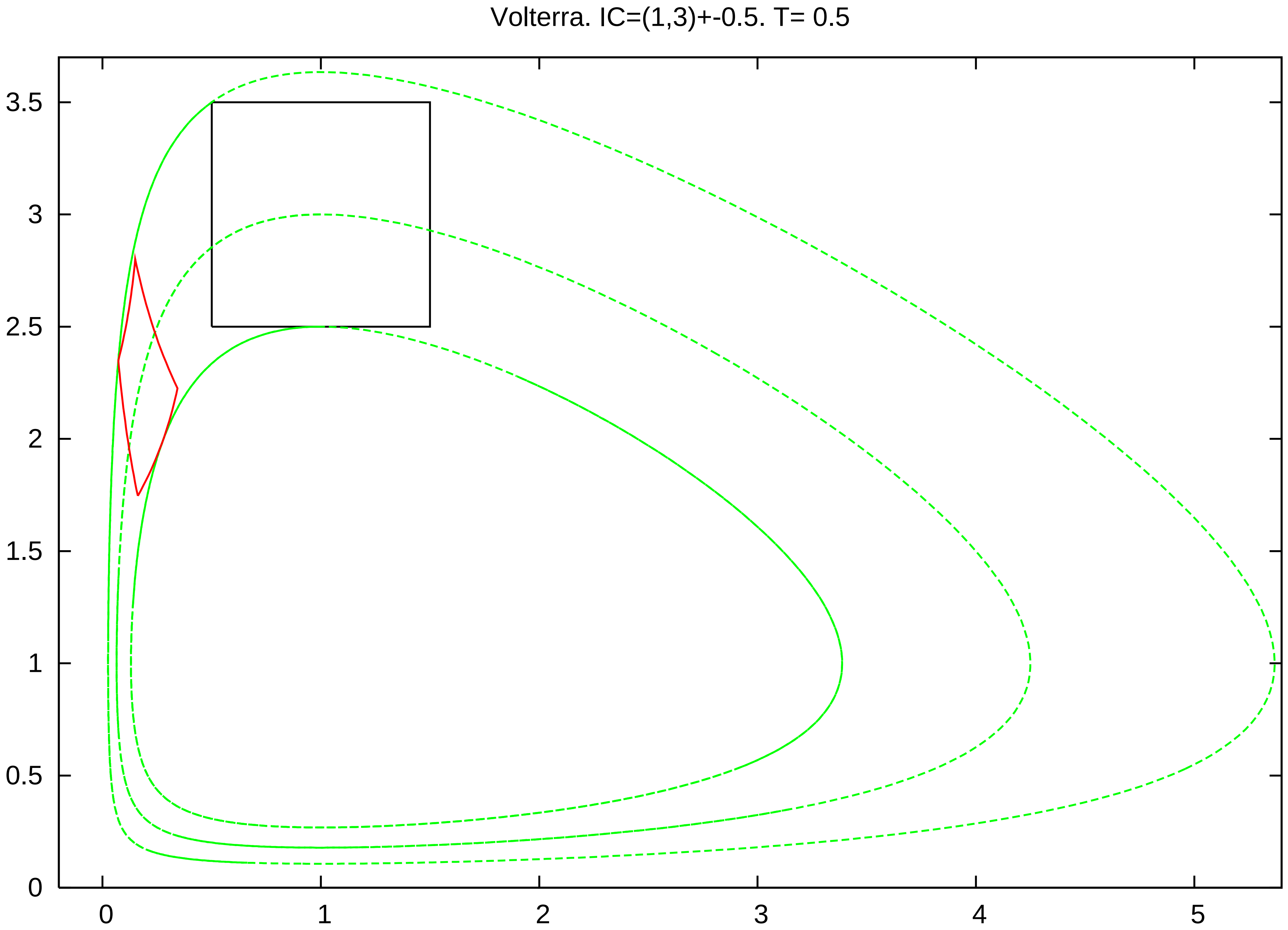


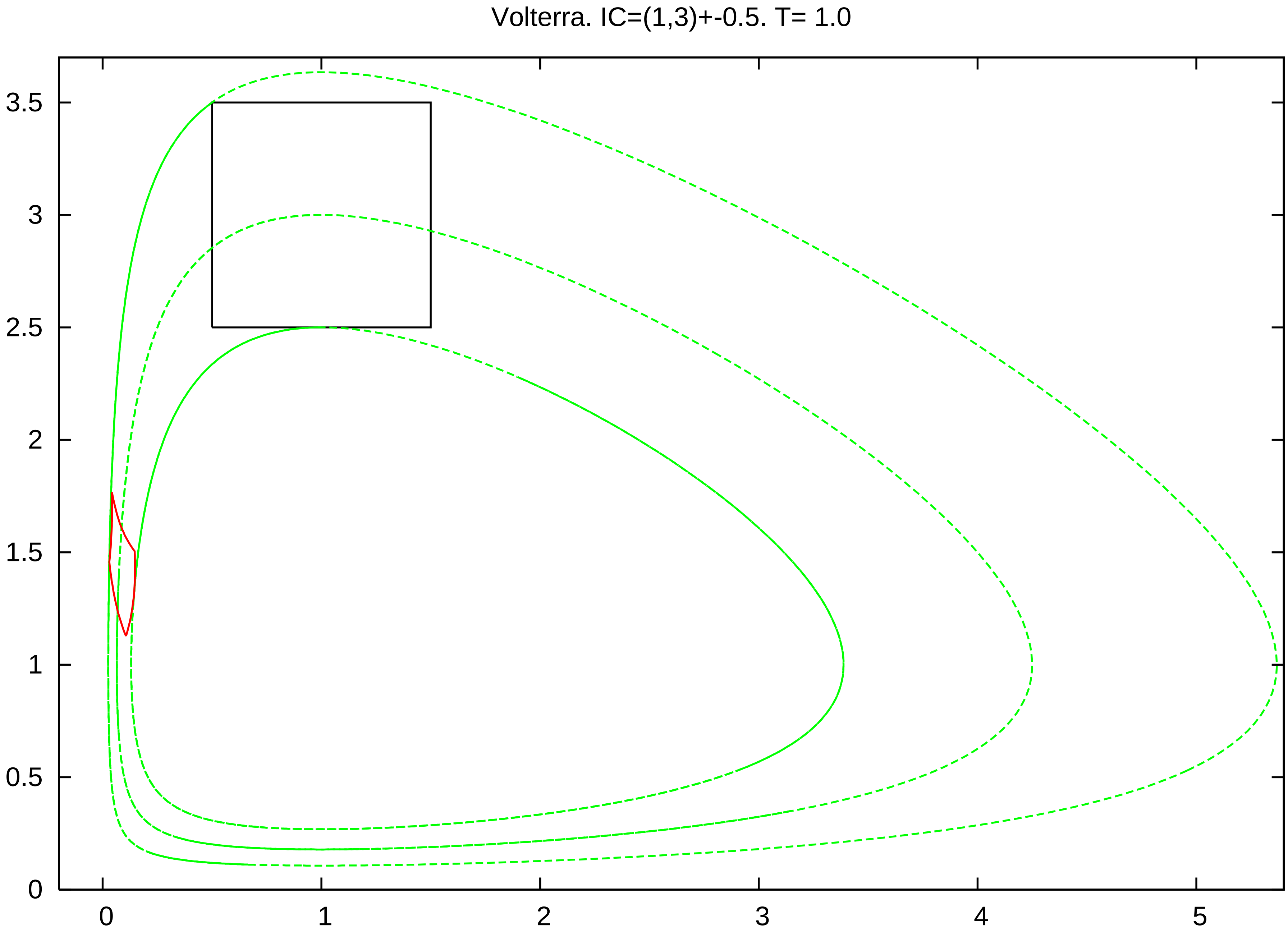
Volterra. Tend=3.5, IC=(1 +-1.2, 3 +-0.5).
3.5
3
2.5
2
1.5
1
0.5
0
-3
-2
-1
0
1
2


Volterra. Tend=3.5, IC=(1 +-1.2, 3 +-0.5).
0.5
0.4
0.3
0.2
0.1
0
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5

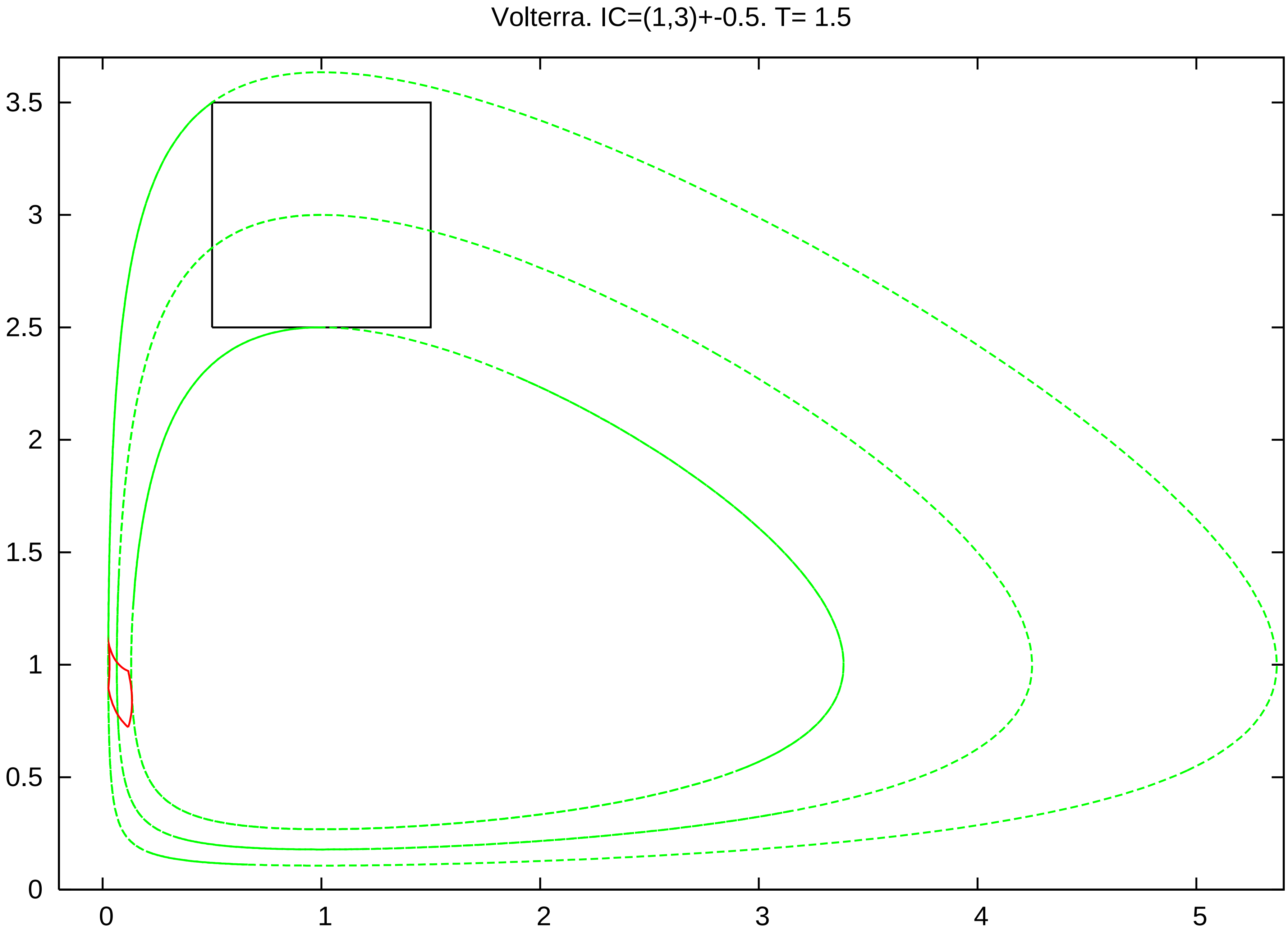

Volterra. Tend=3.5, IC=(1 +-1.2, 3 +-0.5).
0.5
0.4
0.3
0.2
0.1
0
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5


Volterra. IC=(1,3)+-0.5. T= 0.0
0
0.5
1
1.5
2
2.5
3
3.5
0
1
2
3
4
5

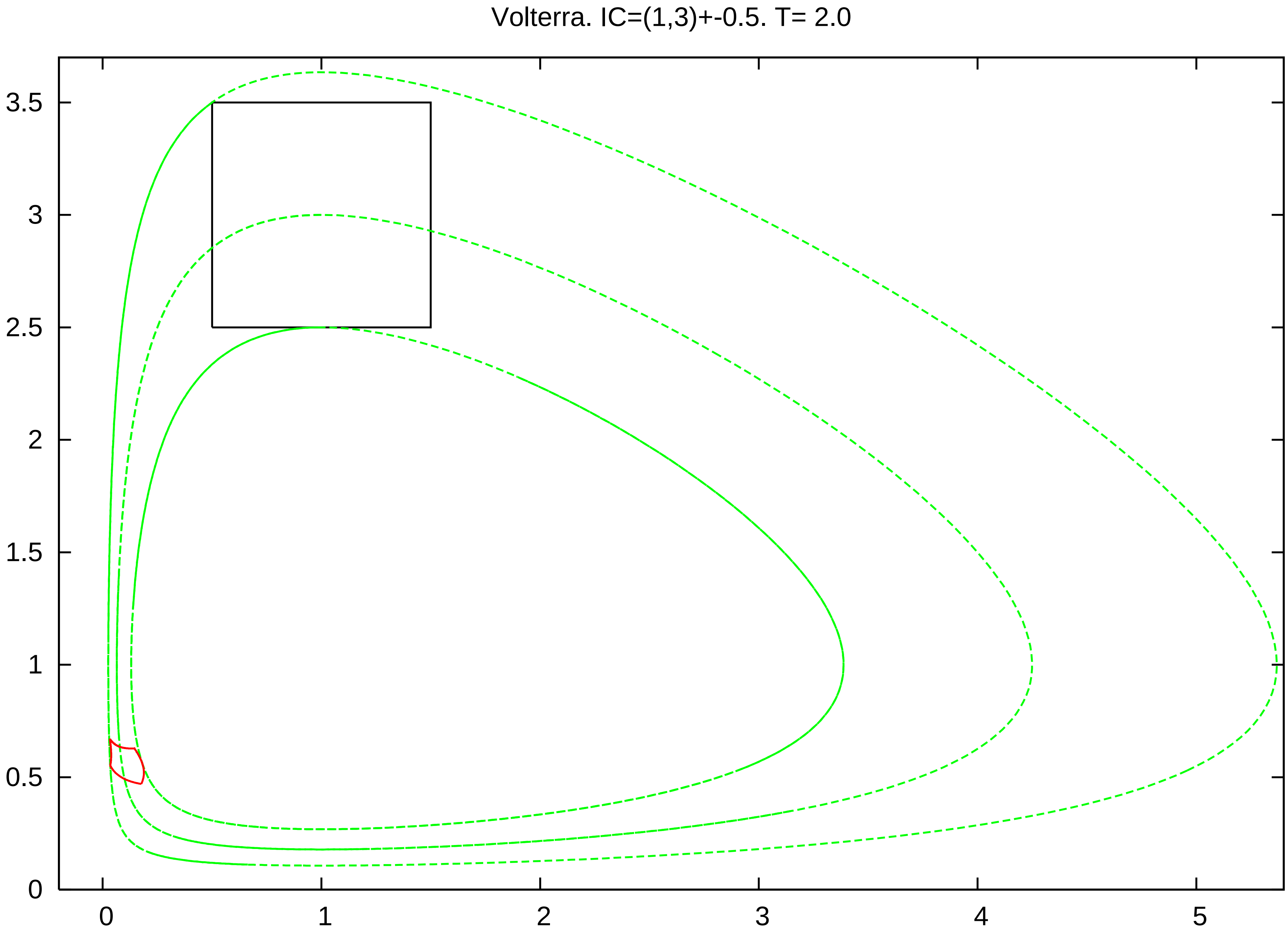

Volterra. IC=(1,3)+-0.5. T= 0.5
0
0.5
1
1.5
2
2.5
3
3.5
0
1
2
3
4
5

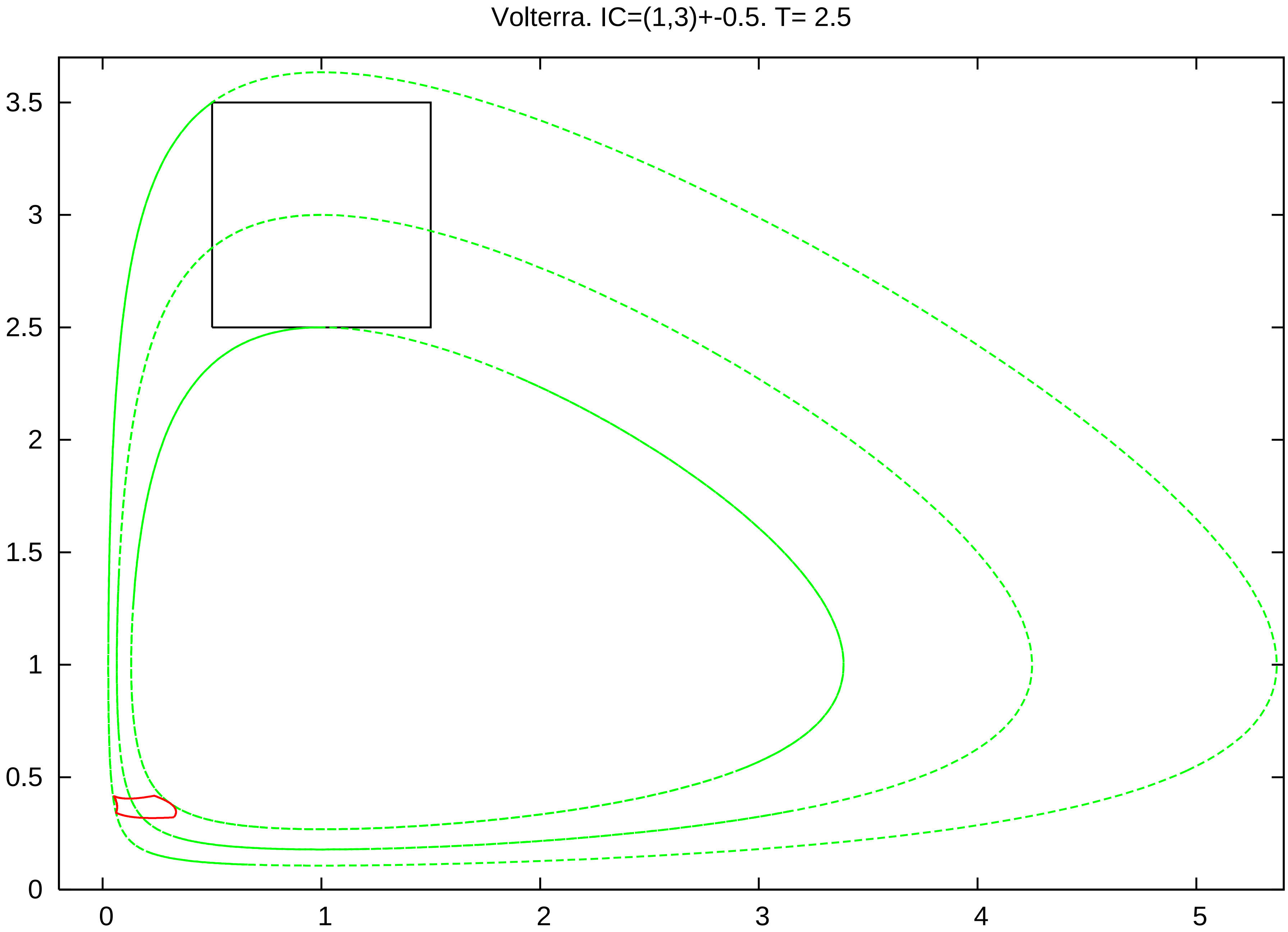


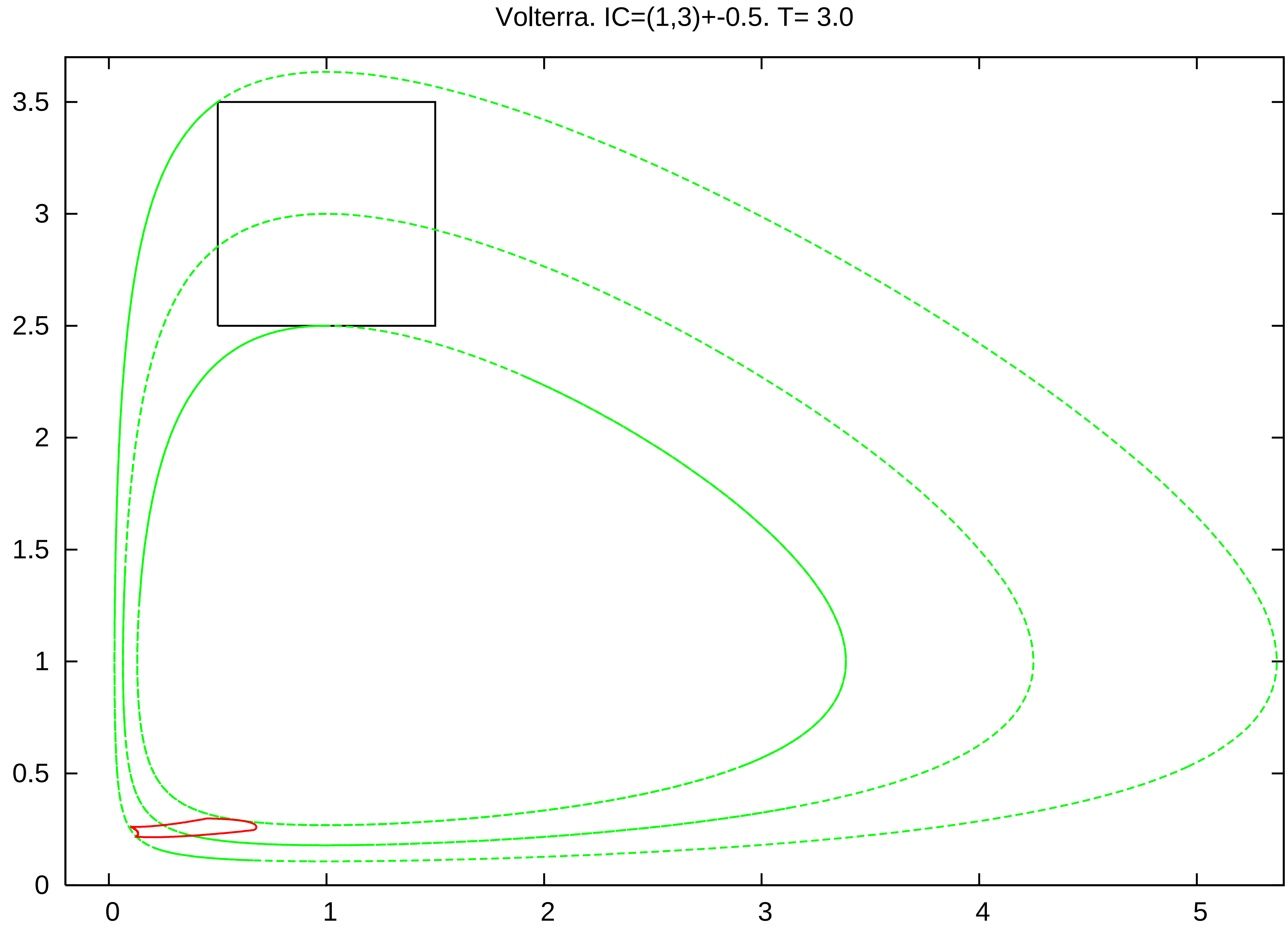
Volterra. IC=(1,3)+-0.5. T= 1.0
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5

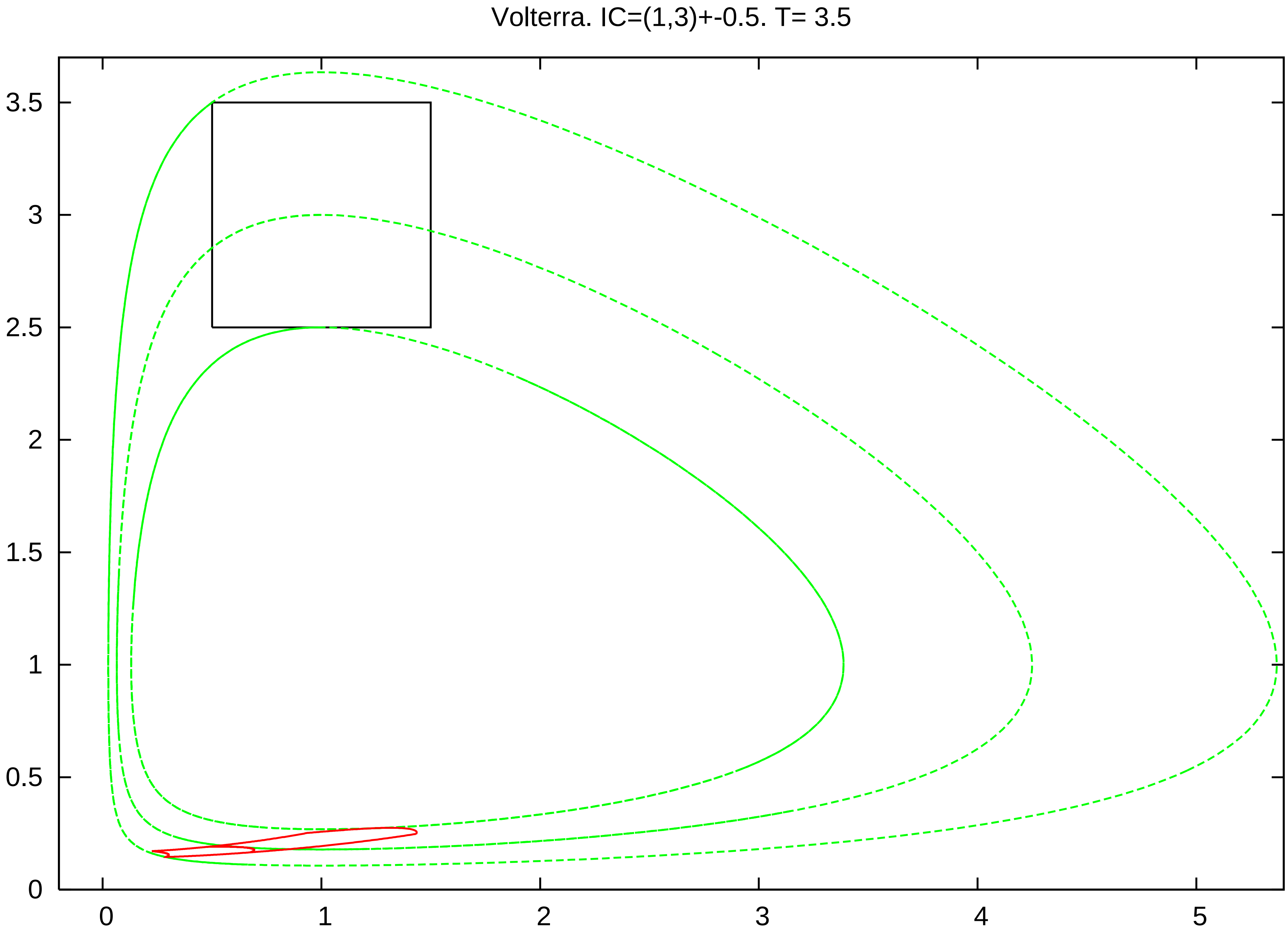


Volterra. IC=(1,3)+-0.5. T= 1.5
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5

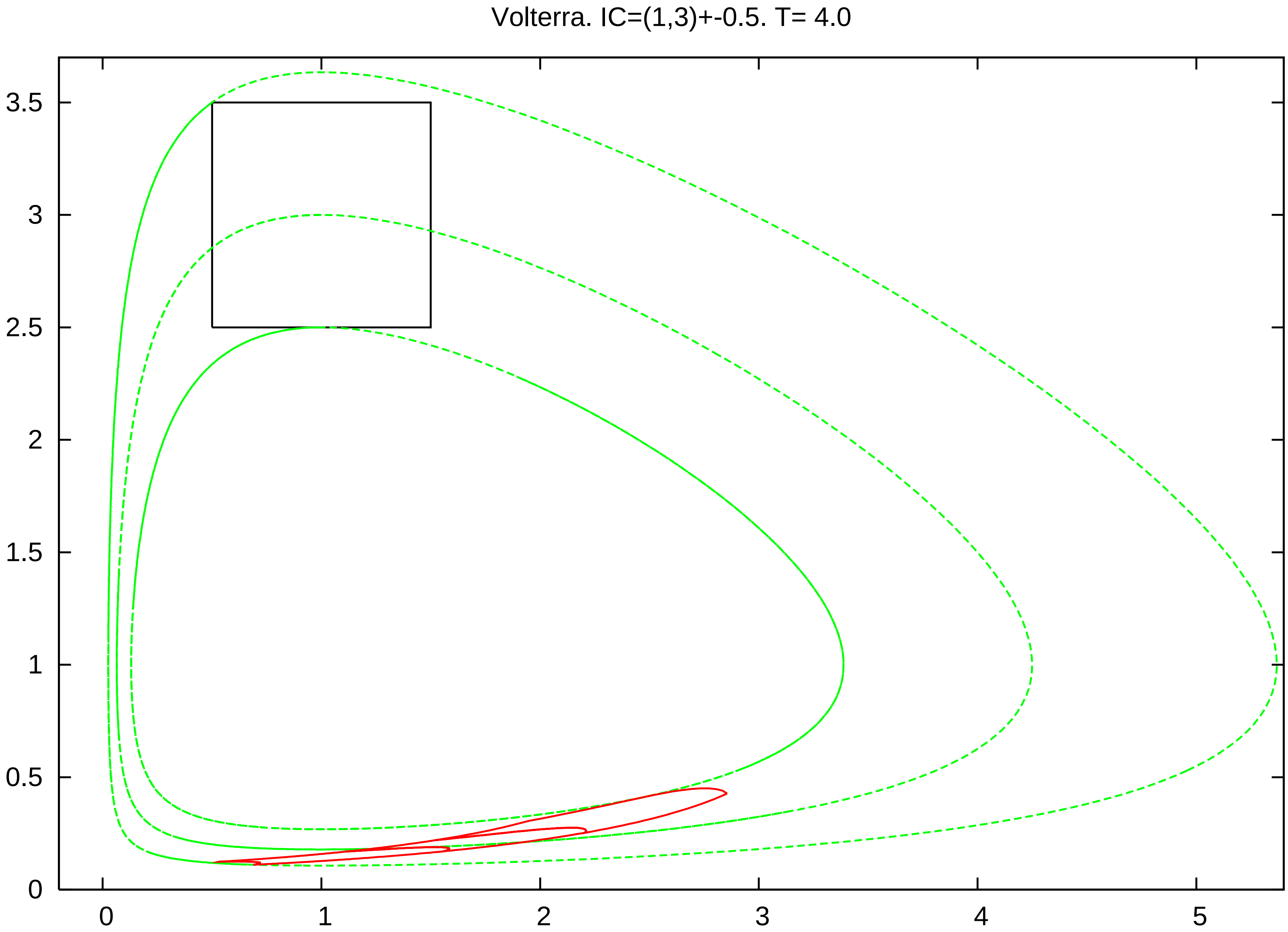

Volterra. IC=(1,3)+-0.5. T= 2.0
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5


Volterra. IC=(1,3)+-0.5. T= 2.5
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5

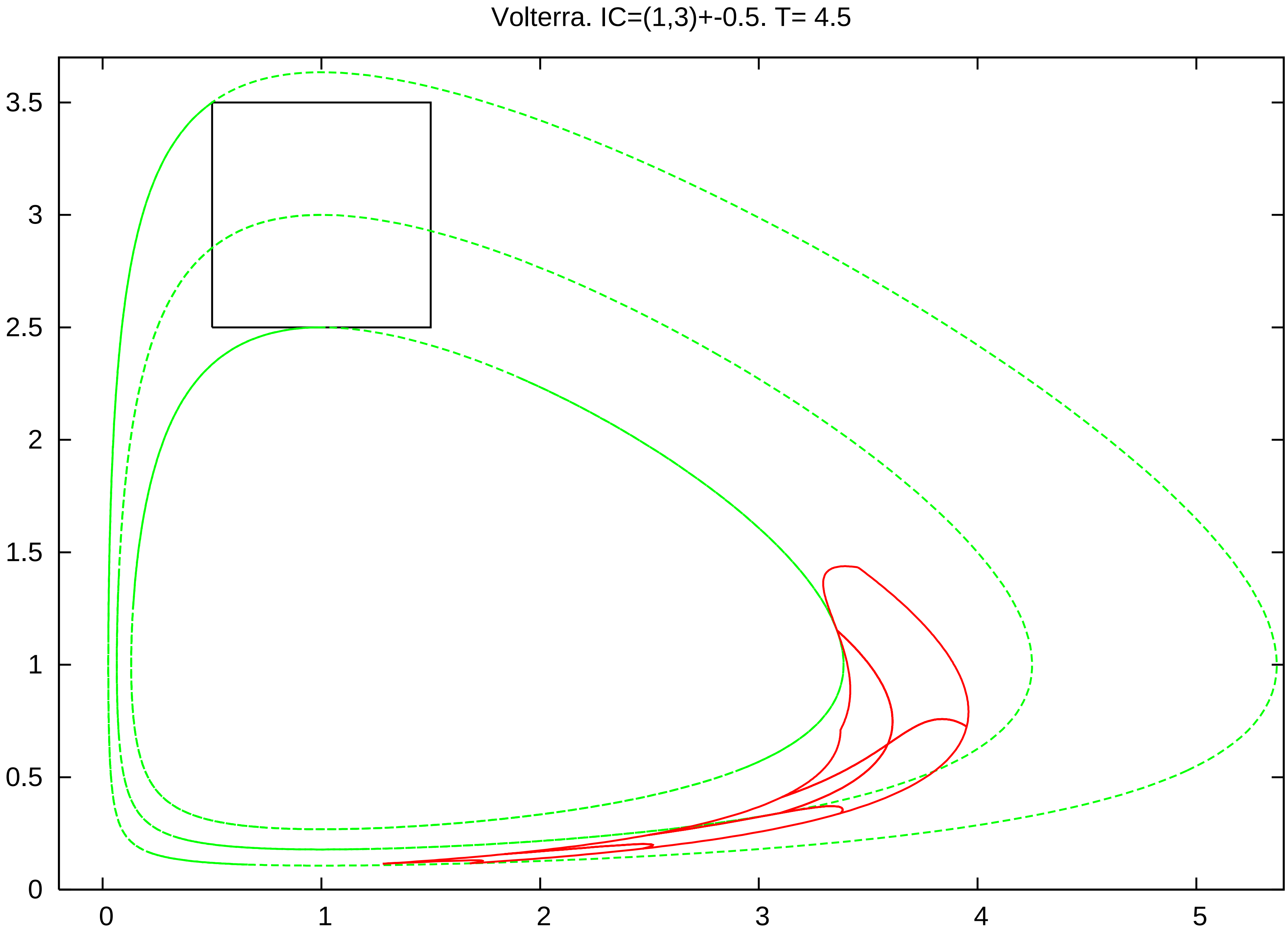

Volterra. IC=(1,3)+-0.5. T= 3.0
0
0.5
1
1.5
2
2.5
3
3.5
0
1
2
3
4
5

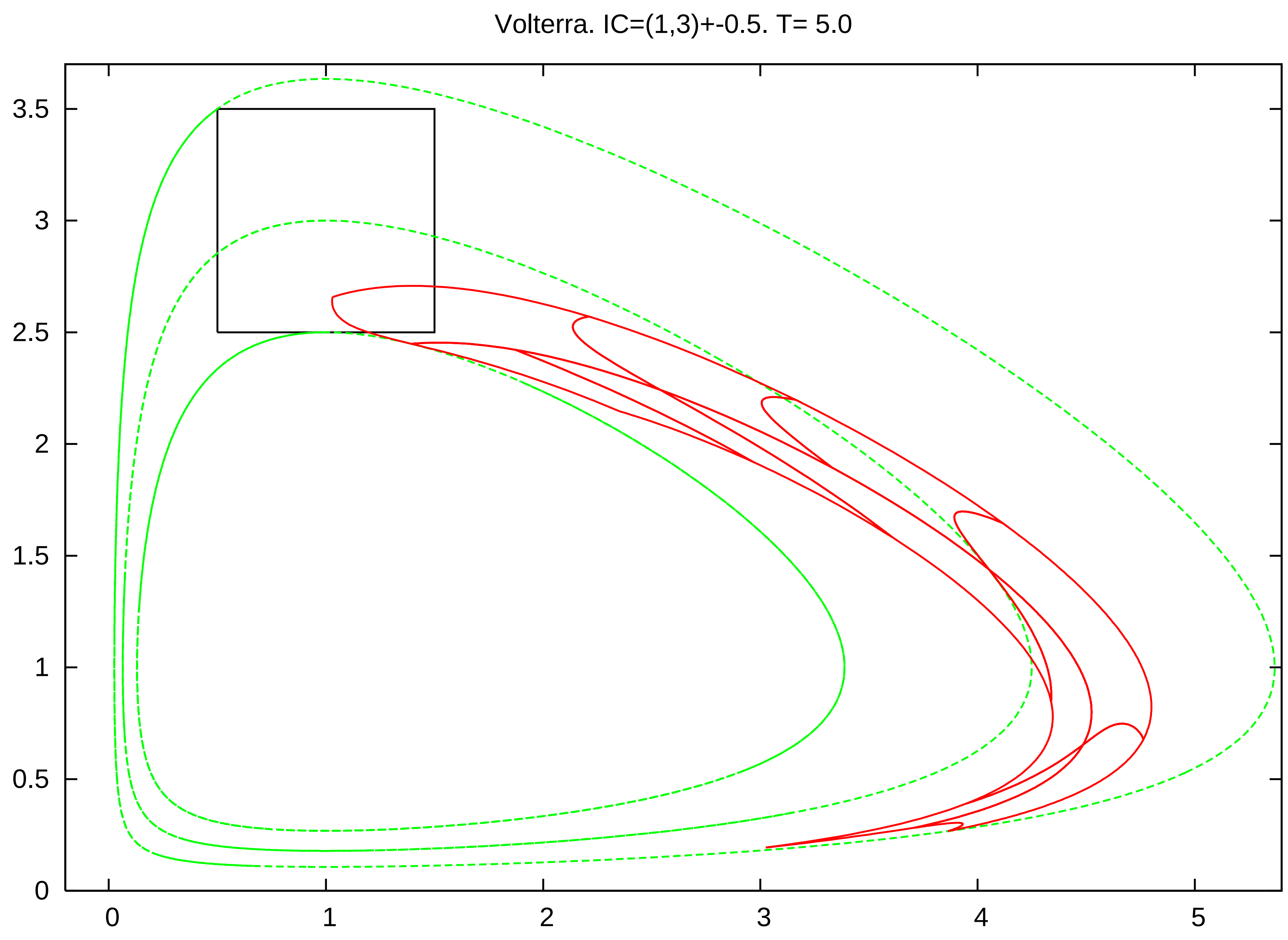


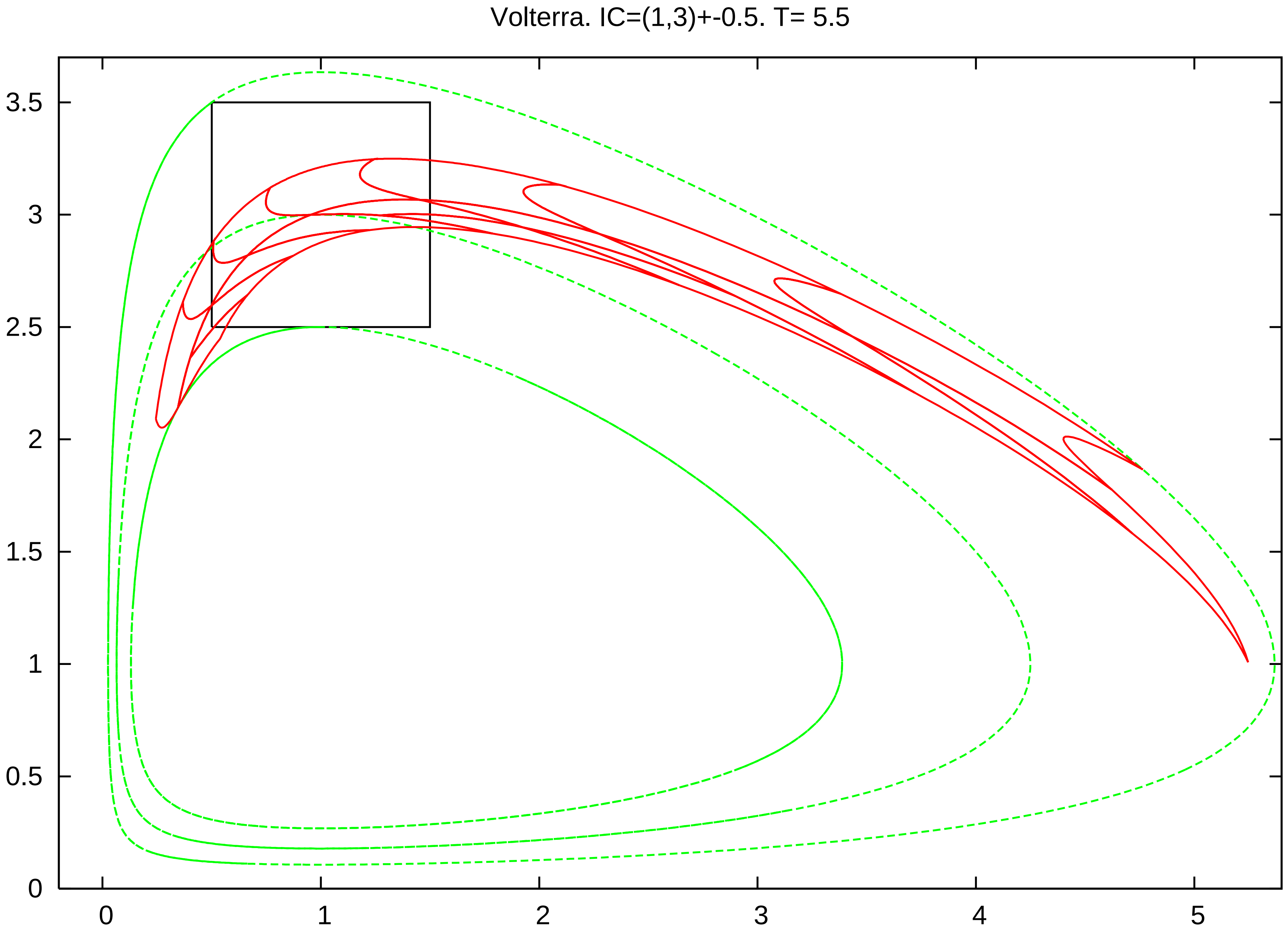
Volterra. IC=(1,3)+-0.5. T= 3.5
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5

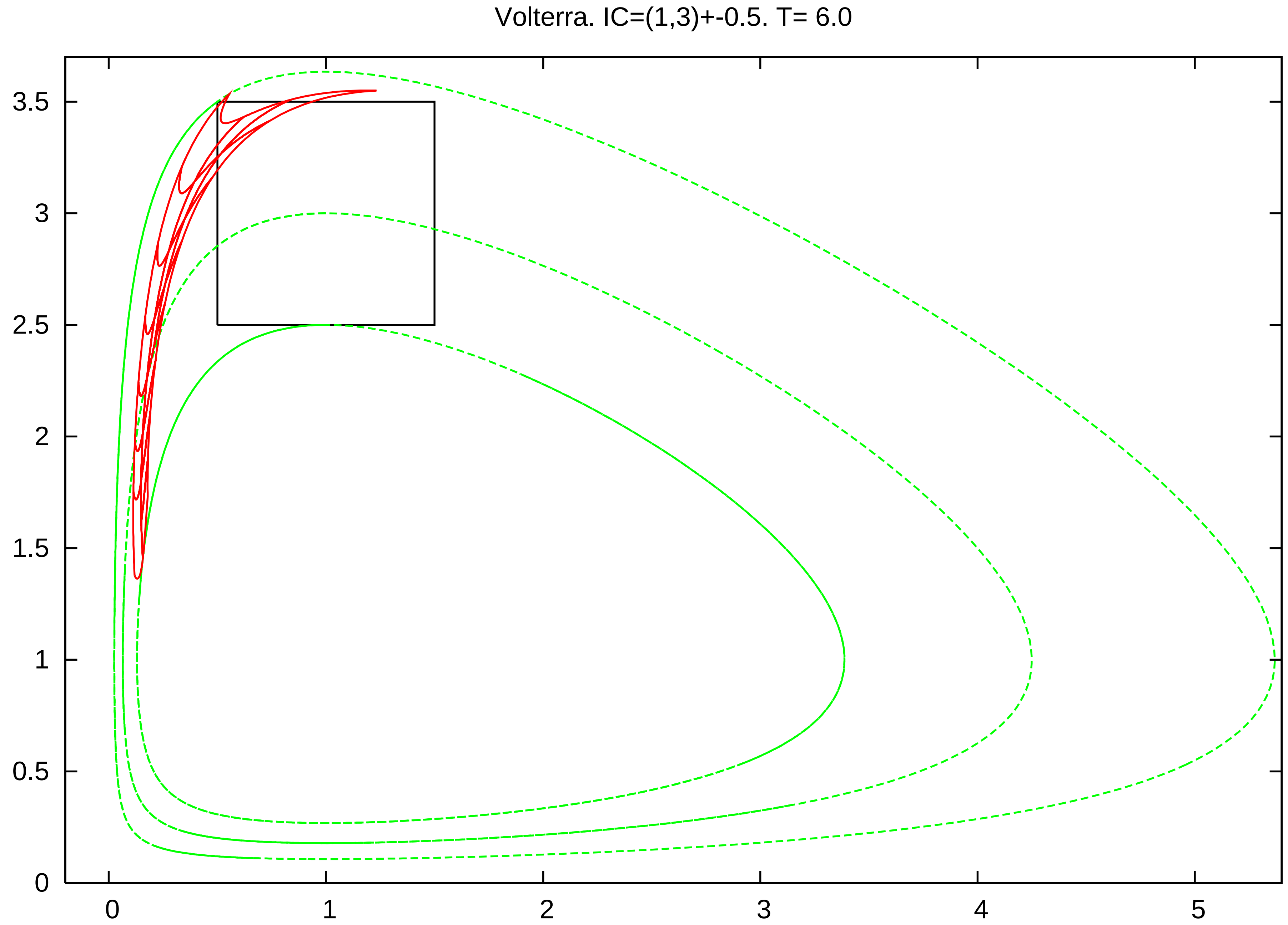


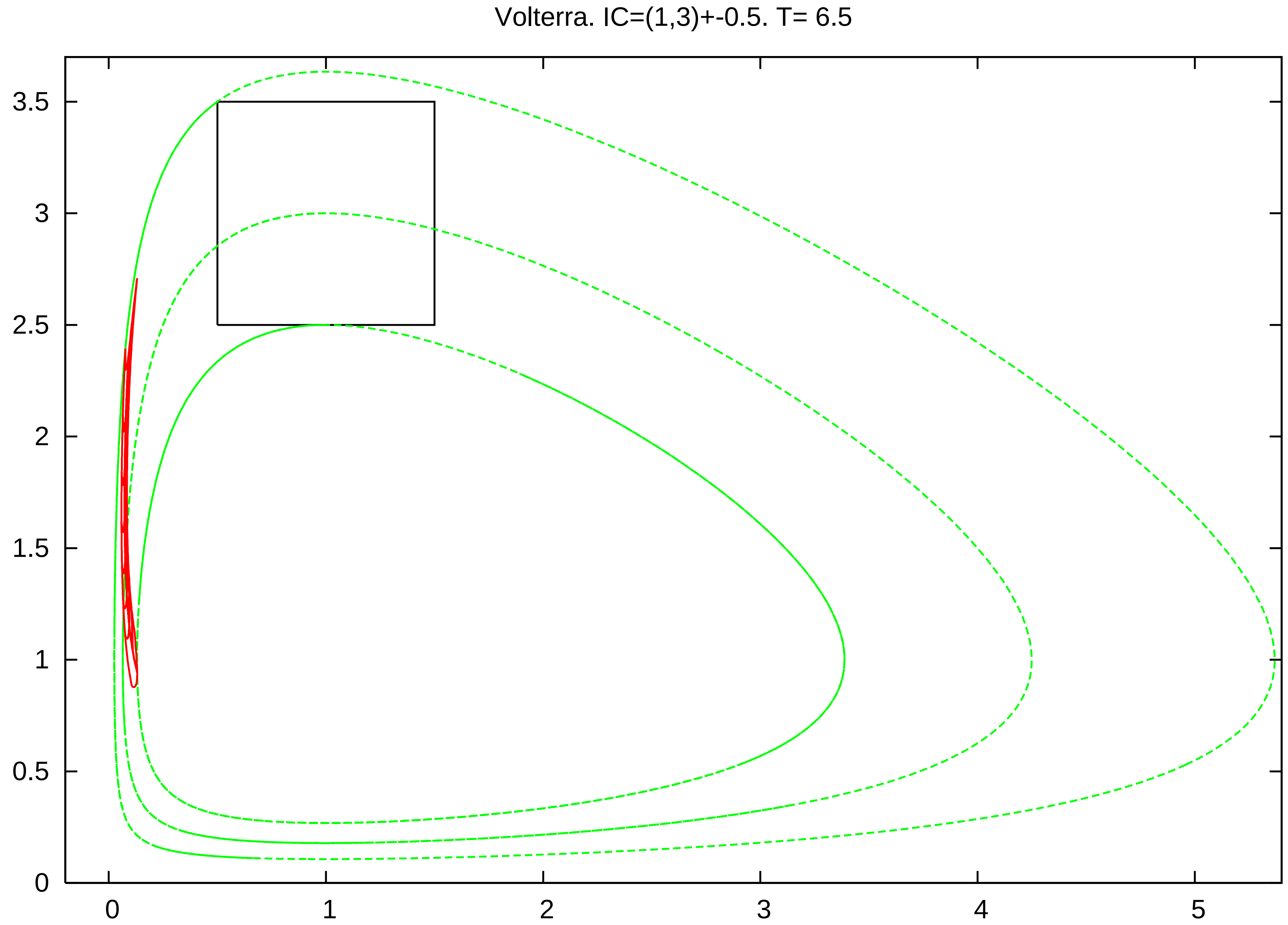
Volterra. IC=(1,3)+-0.5. T= 4.0
0
0.5
1
1.5
2
2.5
3
3.5
0
1
2
3
4
5


Volterra. IC=(1,3)+-0.5. T= 4.5
0
0.5
1
1.5
2
2.5
3
3.5
0
1
2
3
4
5

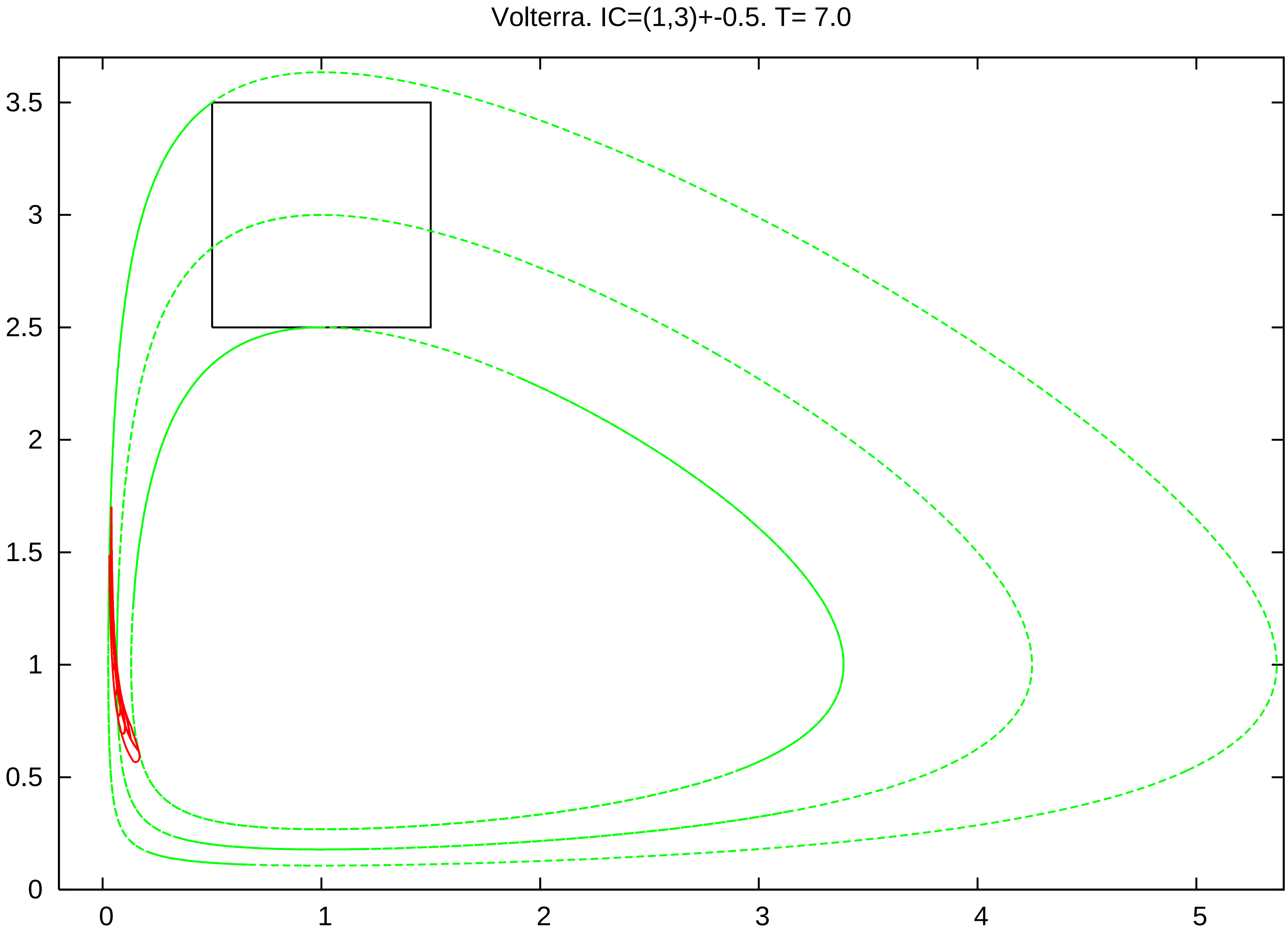

Volterra. IC=(1,3)+-0.5. T= 5.0
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5

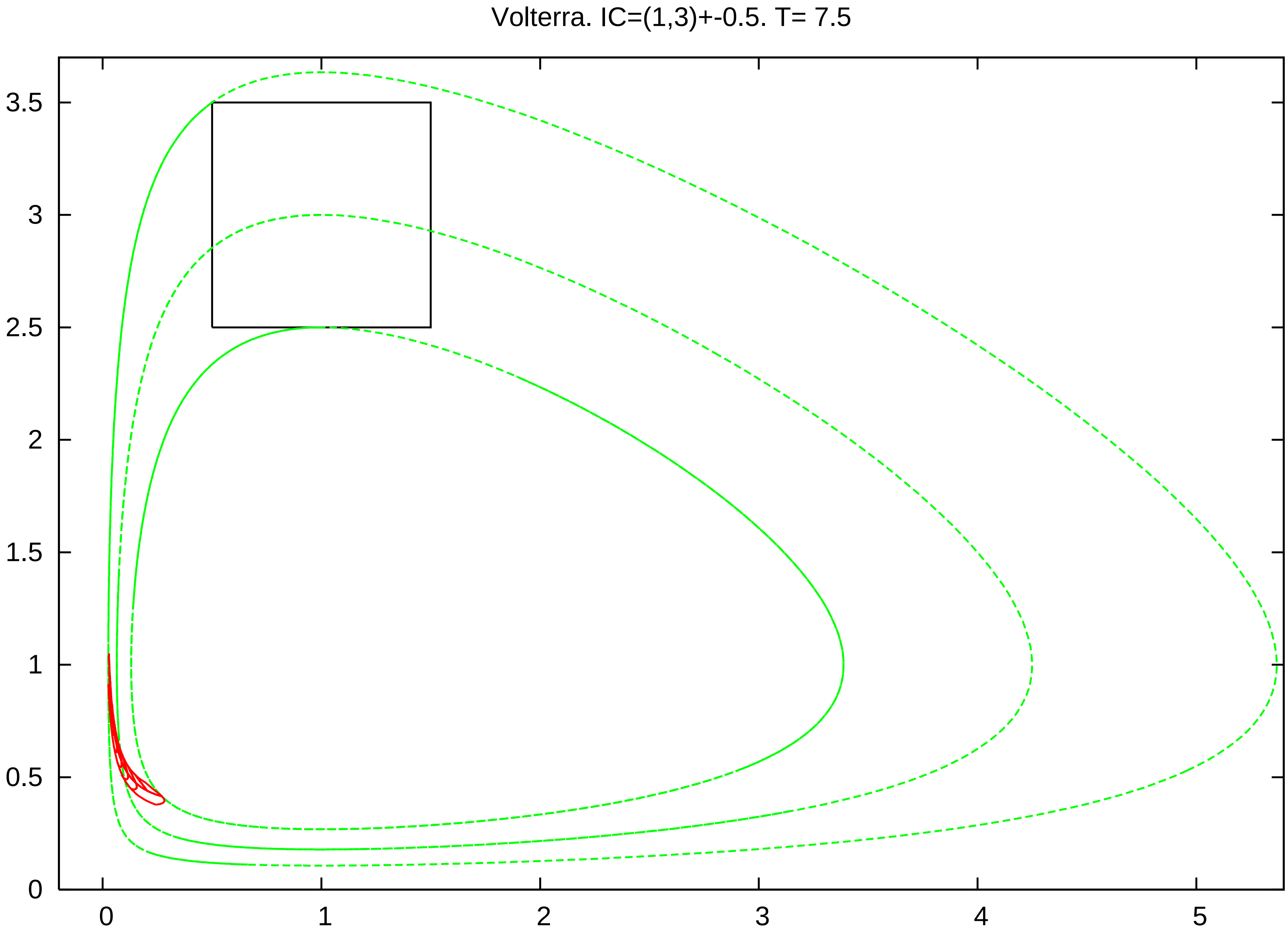

Volterra. IC=(1,3)+-0.5. T= 5.5
0
0.5
1
1.5
2
2.5
3
3.5
0
1
2
3
4
5

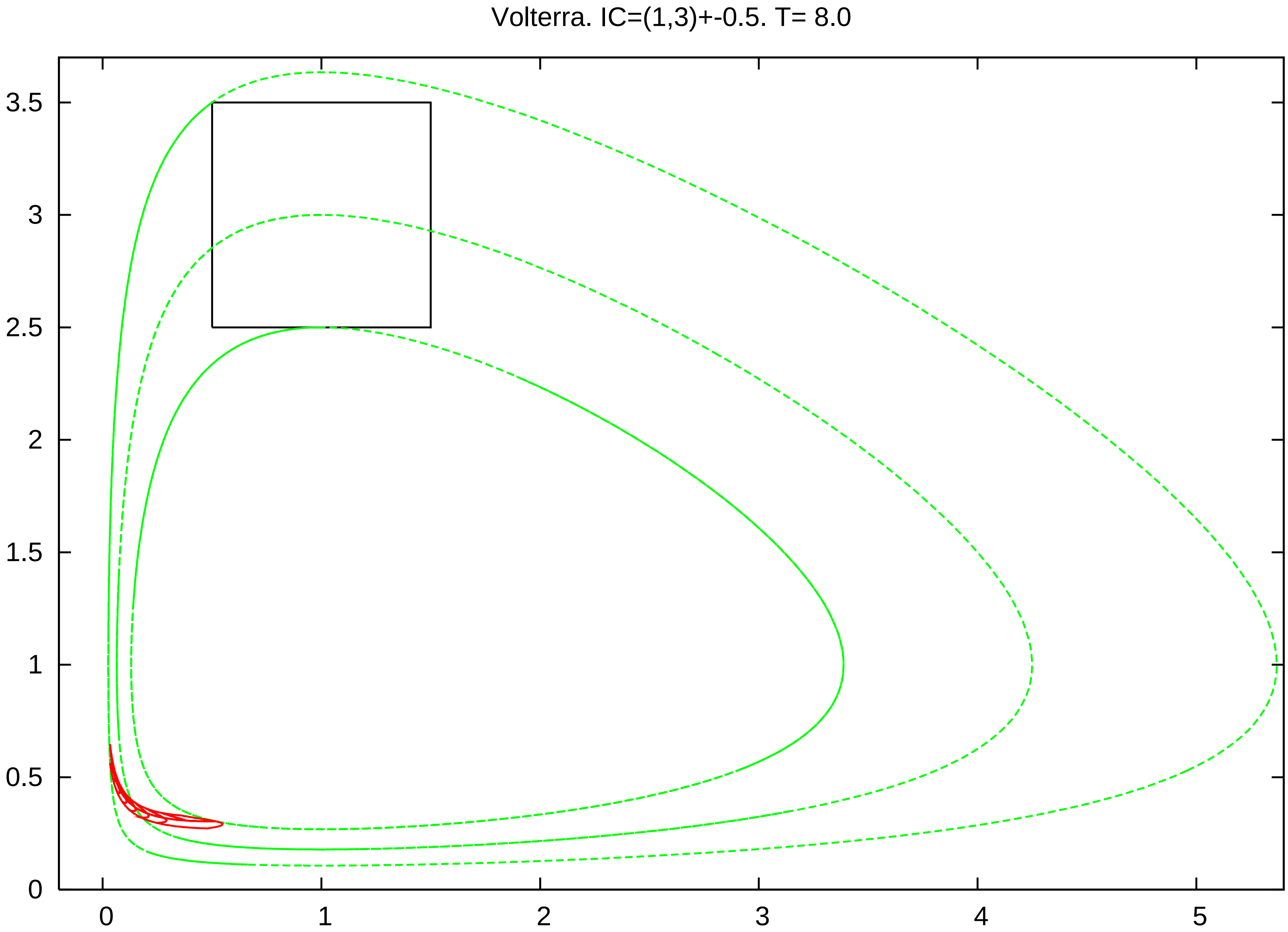

Volterra. IC=(1,3)+-0.5. T= 6.0
0
0.5
1
1.5
2
2.5
3
3.5
0
1
2
3
4
5

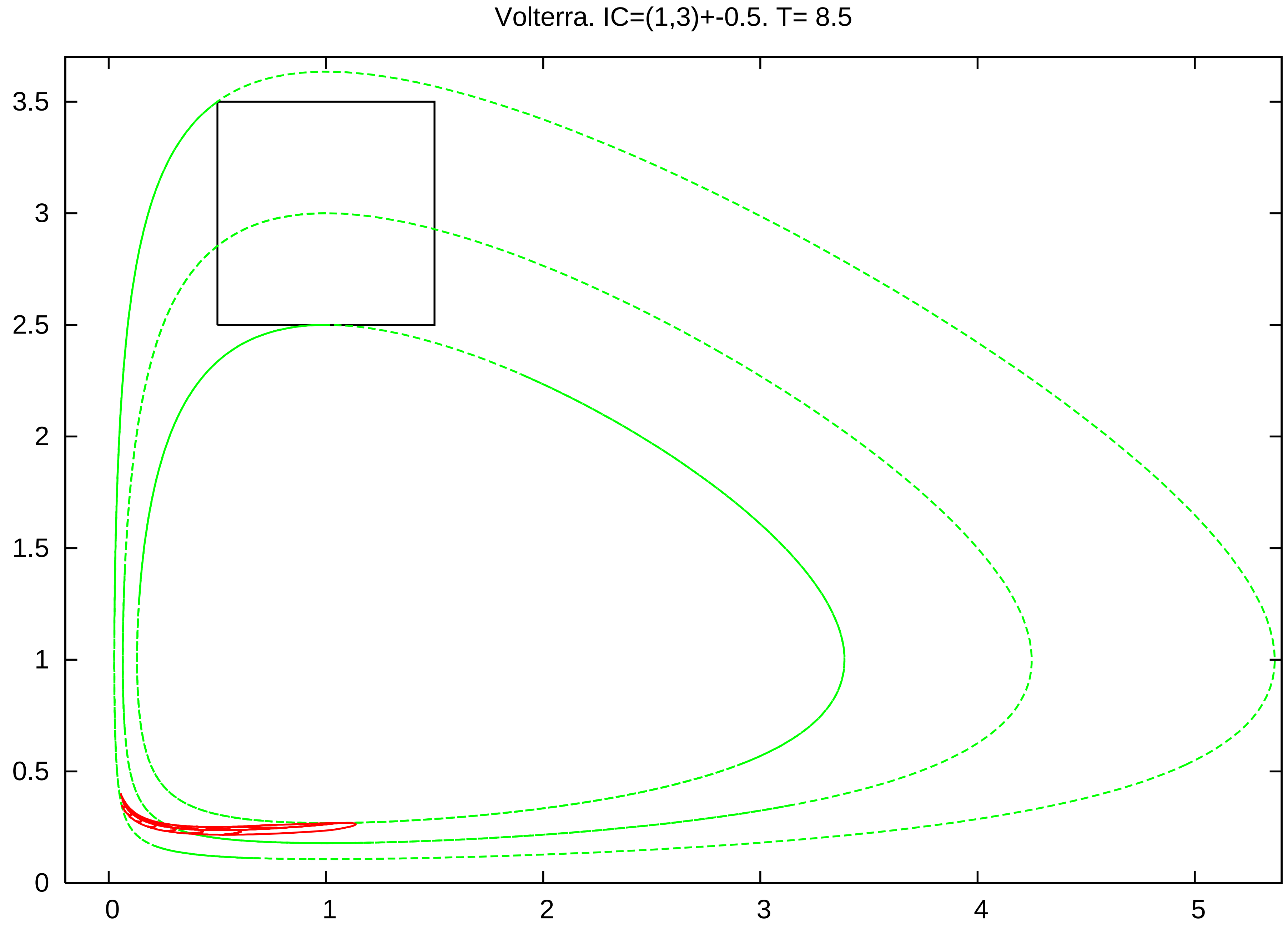


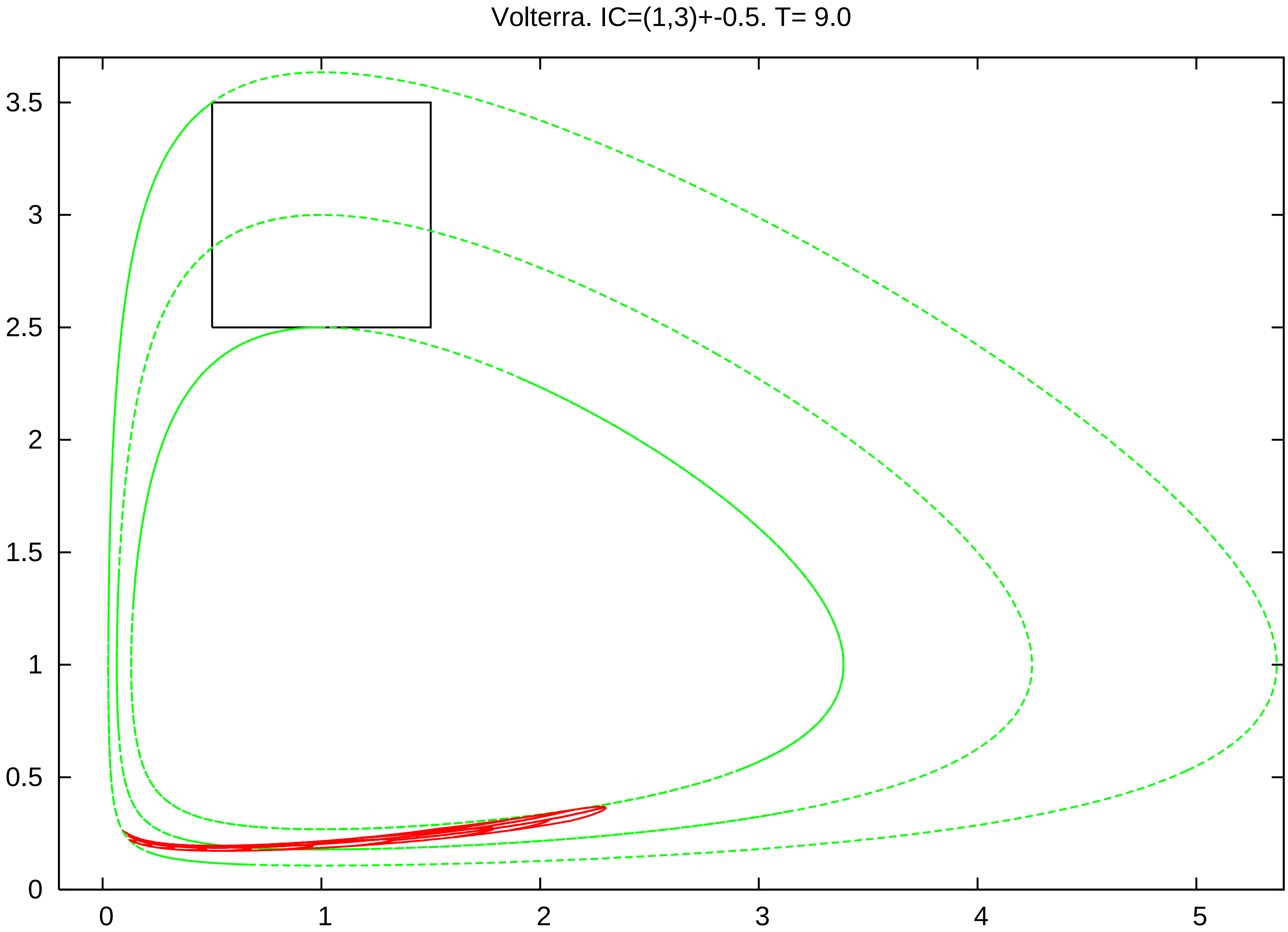
Volterra. IC=(1,3)+-0.5. T= 6.5
0
0.5
1
1.5
2
2.5
3
3.5
0
1
2
3
4
5

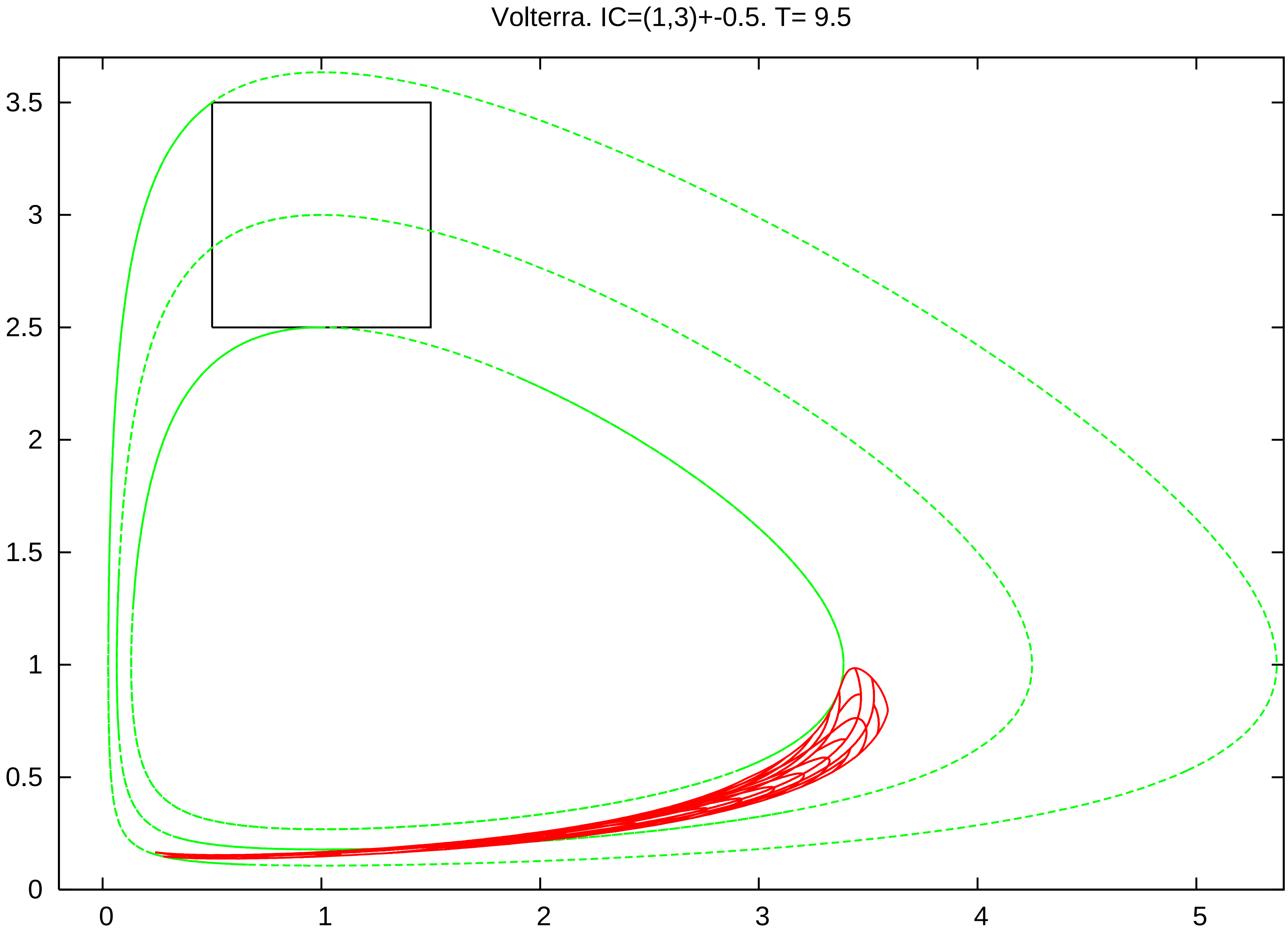

Volterra. IC=(1,3)+-0.5. T= 7.0
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5


Volterra. IC=(1,3)+-0.5. T= 7.5
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5

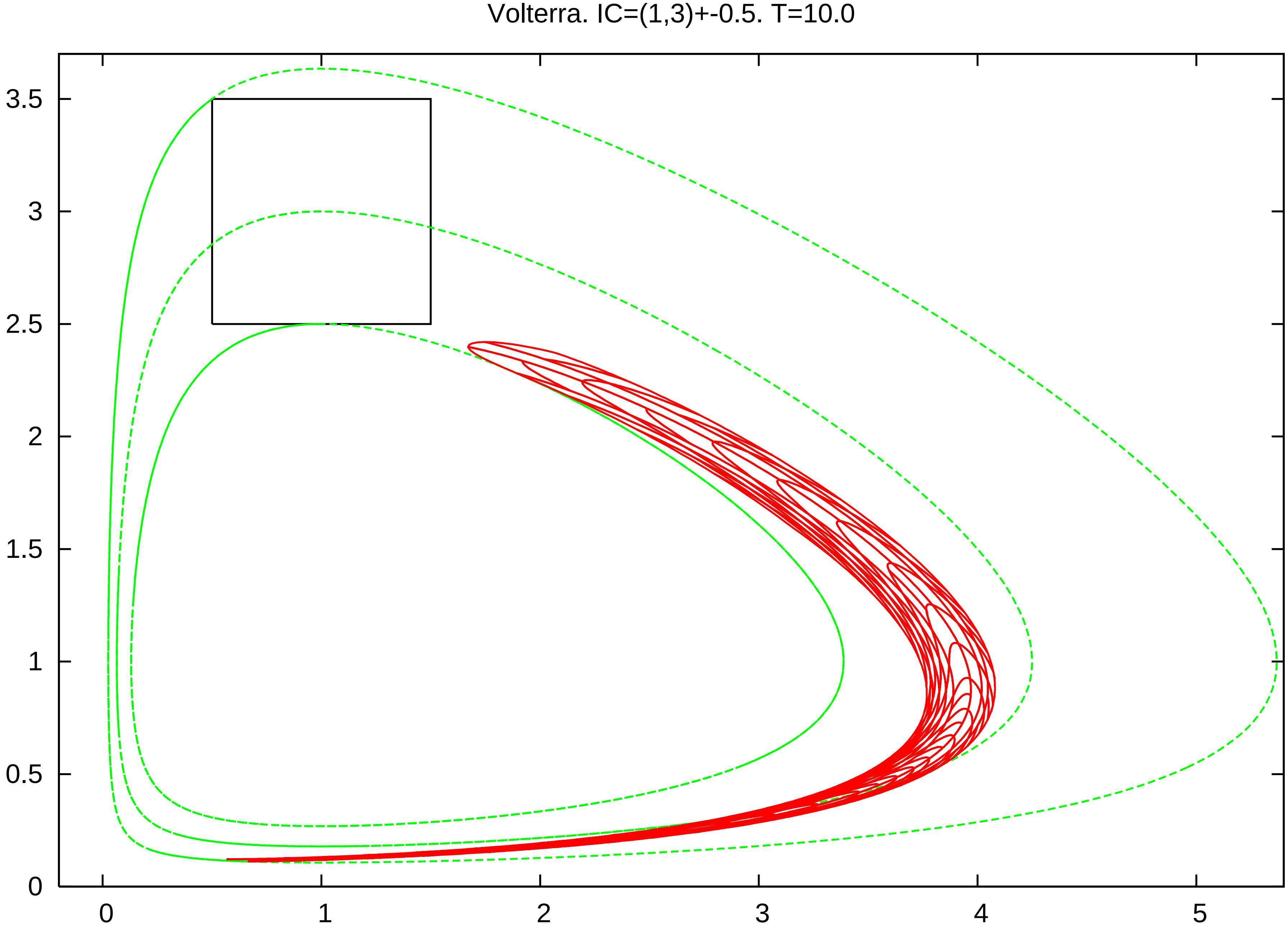

Volterra. IC=(1,3)+-0.5. T= 8.0
0
0.5
1
1.5
2
2.5
3
3.5
0
1
2
3
4
5

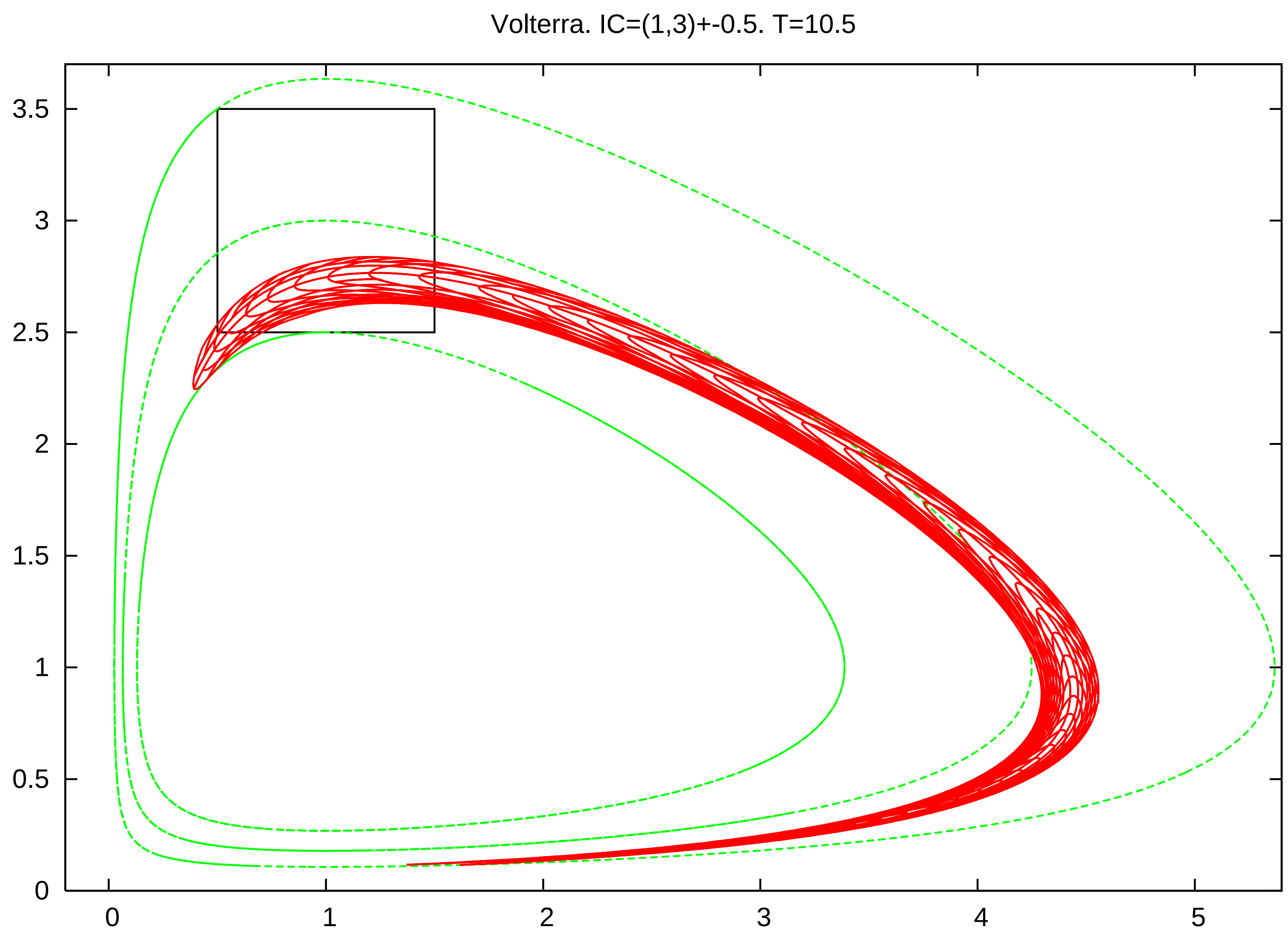

Volterra. IC=(1,3)+-0.5. T= 8.5
0
0.5
1
1.5
2
2.5
3
3.5
0
1
2
3
4
5


Volterra. IC=(1,3)+-0.5. T= 9.0
0
0.5
1
1.5
2
2.5
3
3.5
0
1
2
3
4
5



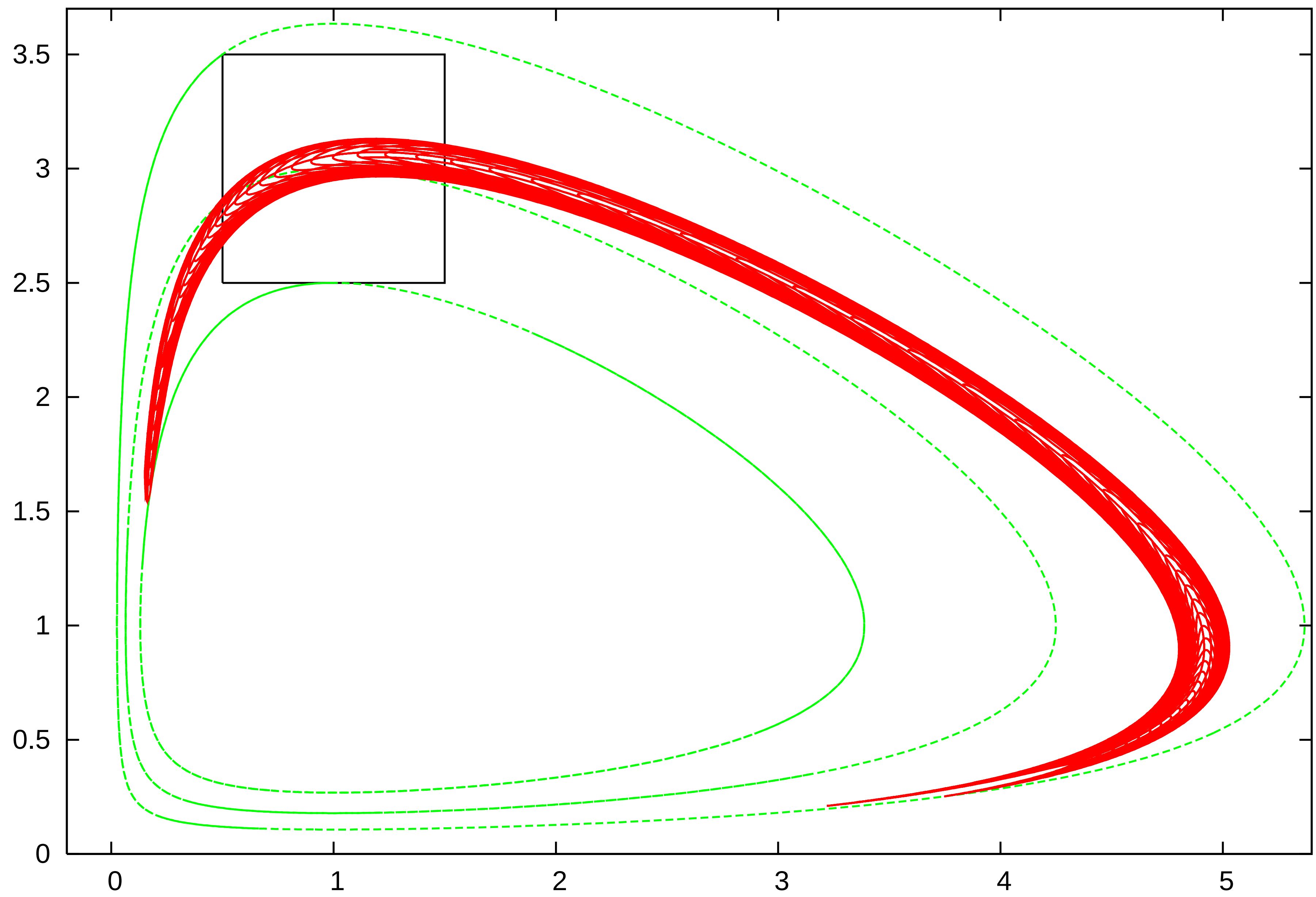
Volterra. IC=(1,3)+-0.5. T= 9.5
0
0.5
1
1.5
2
2.5
3
3.5
0
1
2
3
4
5


Volterra. IC=(1,3)+-0.5. T=10.0
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5



Volterra. IC=(1,3)+-0.5. T=10.5
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5

Volterra. IC=(1,3)+-0.5. T=11.0



The Milano-Michigan ESA Project

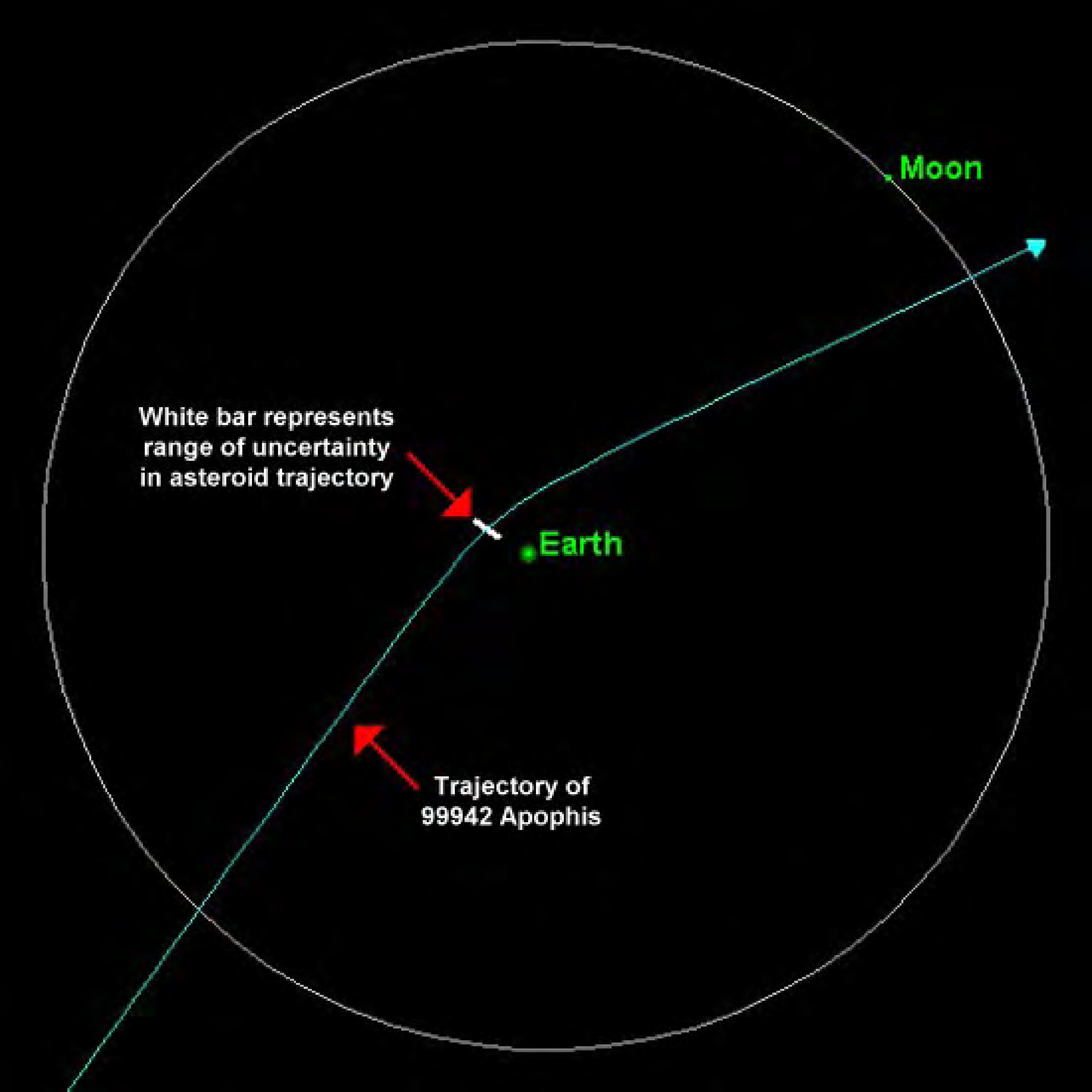
A Collaboration of the Instituto Aerospaziale at Politecnico di Milano and Michigan State University. Currently funded by the European Space Agency to

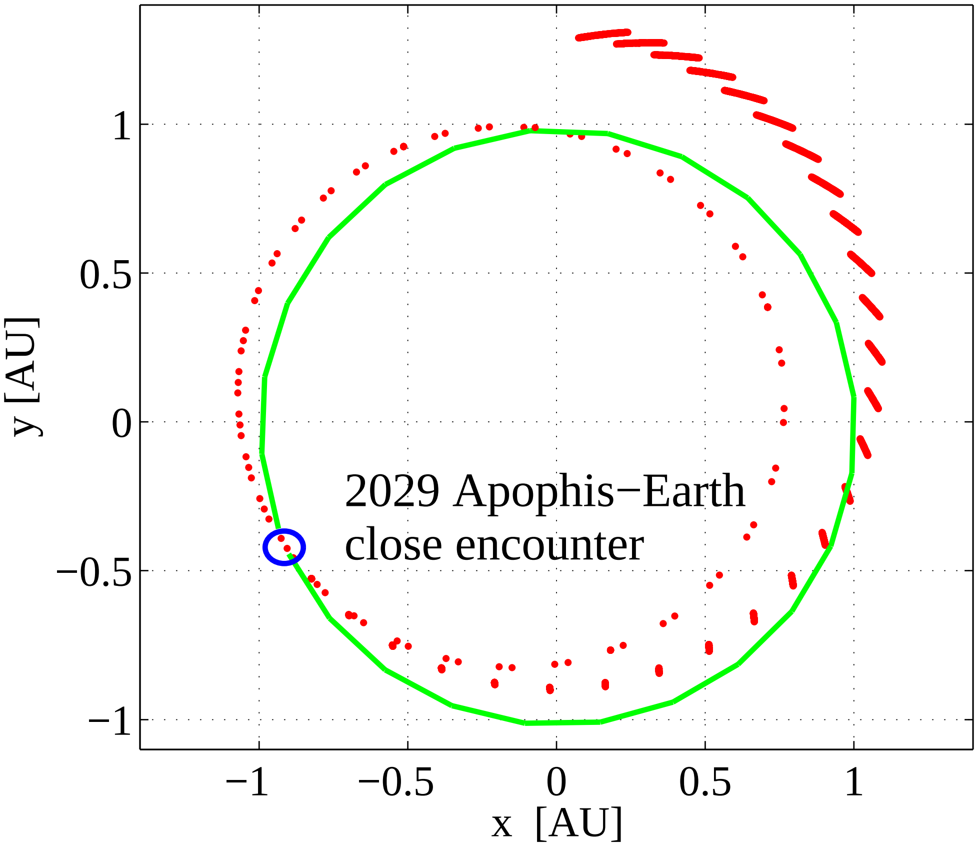
- Develop a verified integrator for solar system dynamics in a complete model of the solar system
- Includes influences of all planets, major asteroids, general relativity, etc
- Analyze its behavior and abilities
- Apply the integrator to study the dynamics of the Near-Earth Asteroid (99942) Apophis

Near Earth Asteroid (99942) Apophis

- A Near-Earth Asteroid discovered in 2004
- Eccentric orbit between the orbits of Venus and Mars
- Apophis will have a first near collision with Earth on **Friday, April 13, 2029**
- Apophis will have another near **(???)** collision with Earth on (Monday), **April 13, 2036**
- The near collision in 2029 very significantly alters Apophis' orbit

The small uncertainties of Apophis' current orbit parameters, amplified by the influence of the near collision in 2029, makes predictions for 2036 **very difficult.**


Moon
White bar represents
range of uncertainty
in asteroid trajectory
Earth
Trajectory of
99942 Apophis



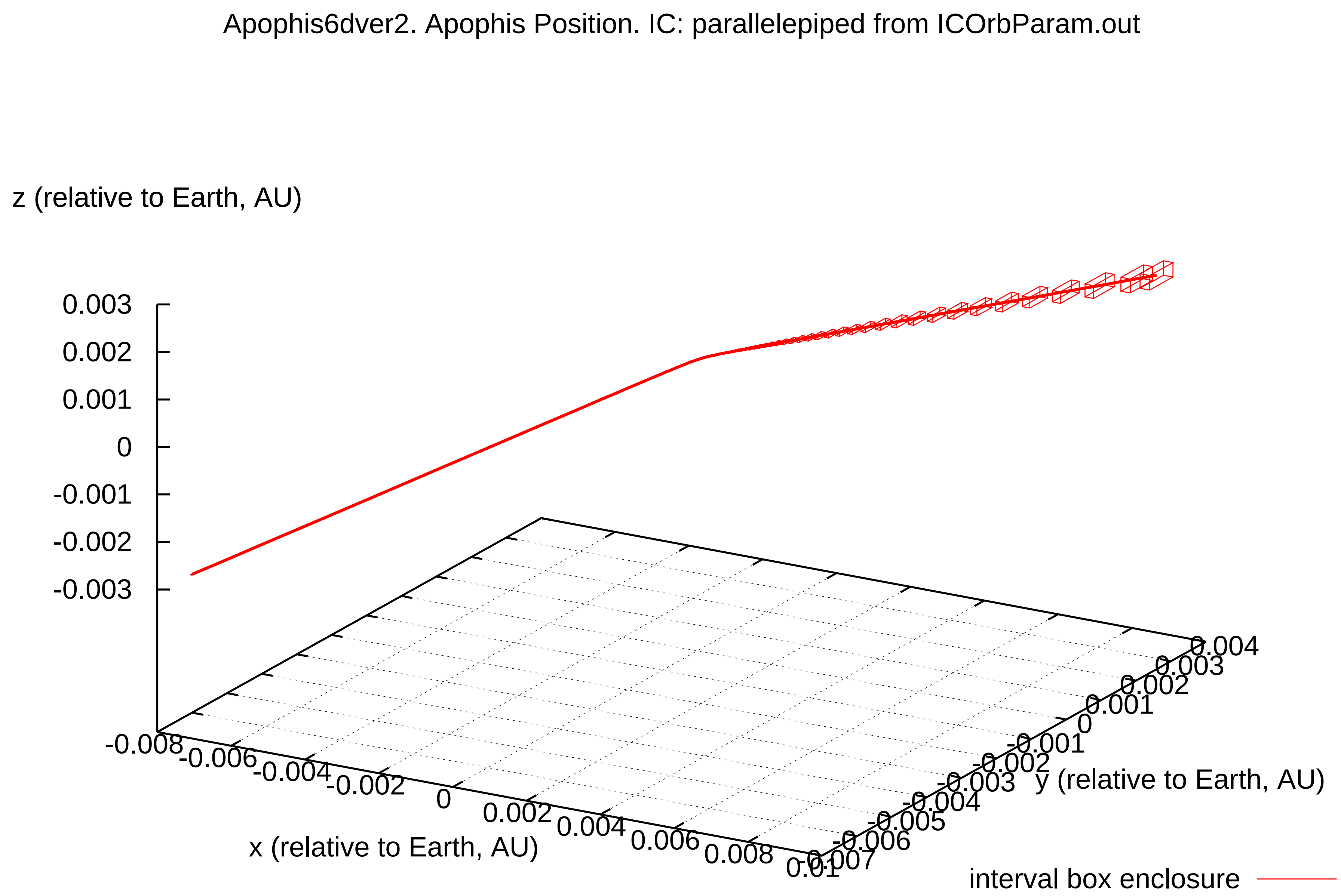
2029 Apophis−Earth
close encounter
y [AU]
x [AU]
1
0.5
0
−0.5
−1
−1
−0.5
0
0.5
1

(99942) Apophis - Encounter 2036

Prediction of motion of Apophis is very difficult. Its orbit is significantly affected by tiny perturbations:

- Detailed shape of Earth's gravitational field (oblateness, mountains)
- Gravitational pull of other asteroids
- Radiation pressure from Sun (even a small reflective shield being applied can deflect the asteroid)
- 64 bit accuracy of numerical integrators (regardless of verification)

All these influences affect the final position to the size of more than one Earth diameter


Apophis6dver2. Apophis Position. IC: parallelepiped from ICOrbParam.out
z (relative to Earth, AU)
0.003
0.002
0.001
0
-0.001
-0.002
-0.003
-0.008
-0.006
-0.004
-0.002
0
0.002
0.004
0.006
0.008
0.01
x (relative to Earth, AU)
0.004
0.003
0.002
0.001
0
-0.001
-0.002
-0.003
-0.004
-0.005
-0.006
-0.007
y (relative to Earth, AU)
interval box enclosure

The Lorenz Equations

The equations describe a simplified model of unpredictable turbulent flows in fluid dynamics.
Exhibits sensitive dependence on initial conditions and chaoticity.

$$\begin{aligned}\dot{x} &= \sigma(y - x)\\ \dot{y} &= x(\rho - z) - y\\ \dot{z} &= xy - \beta z\end{aligned}$$

The standard parameter values are

$$\sigma = 10, \beta = \frac{8}{3}, \rho = 28$$

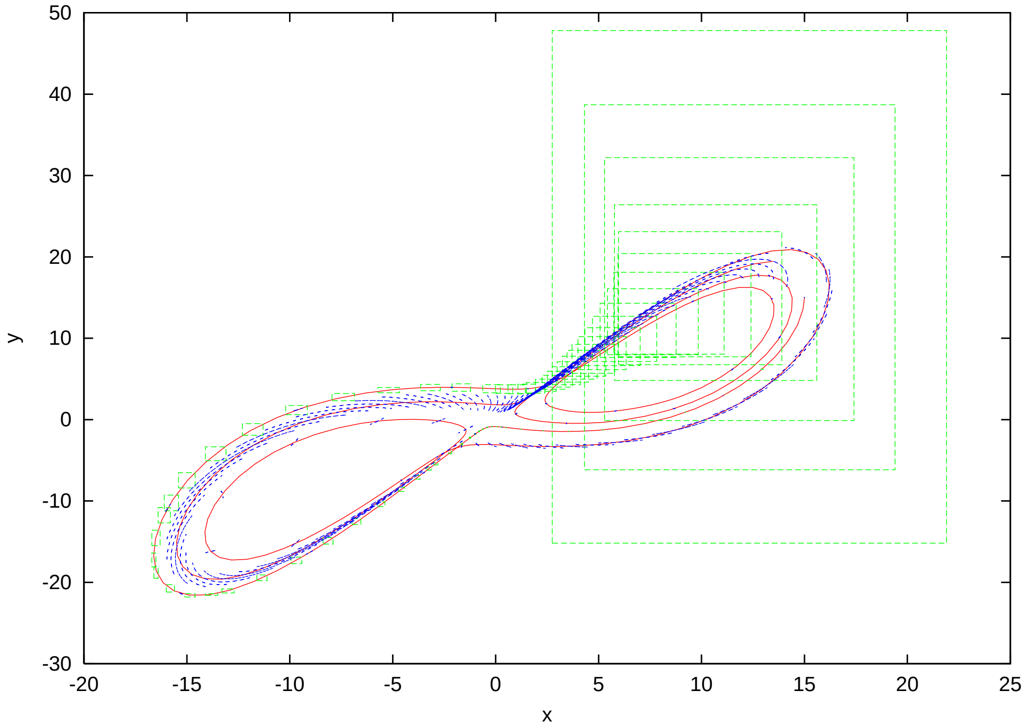
and ρ is often varied. The fixed points are

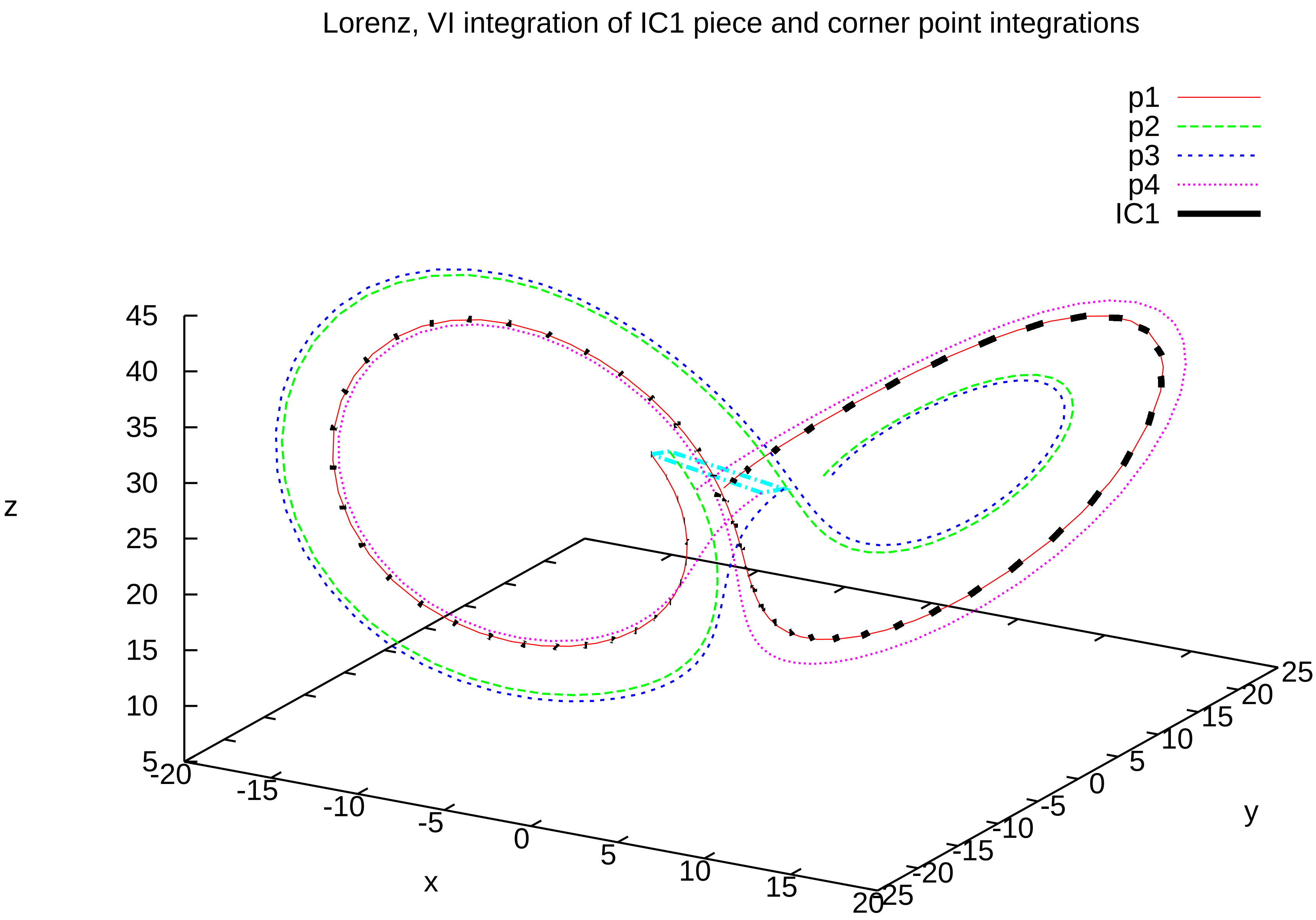
$$(0,0,0), \quad (\pm\sqrt{\beta(\rho - 1)}, \pm\sqrt{\beta(\rho - 1)}, \rho - 1).$$

Verified Integration of the Lorenz System

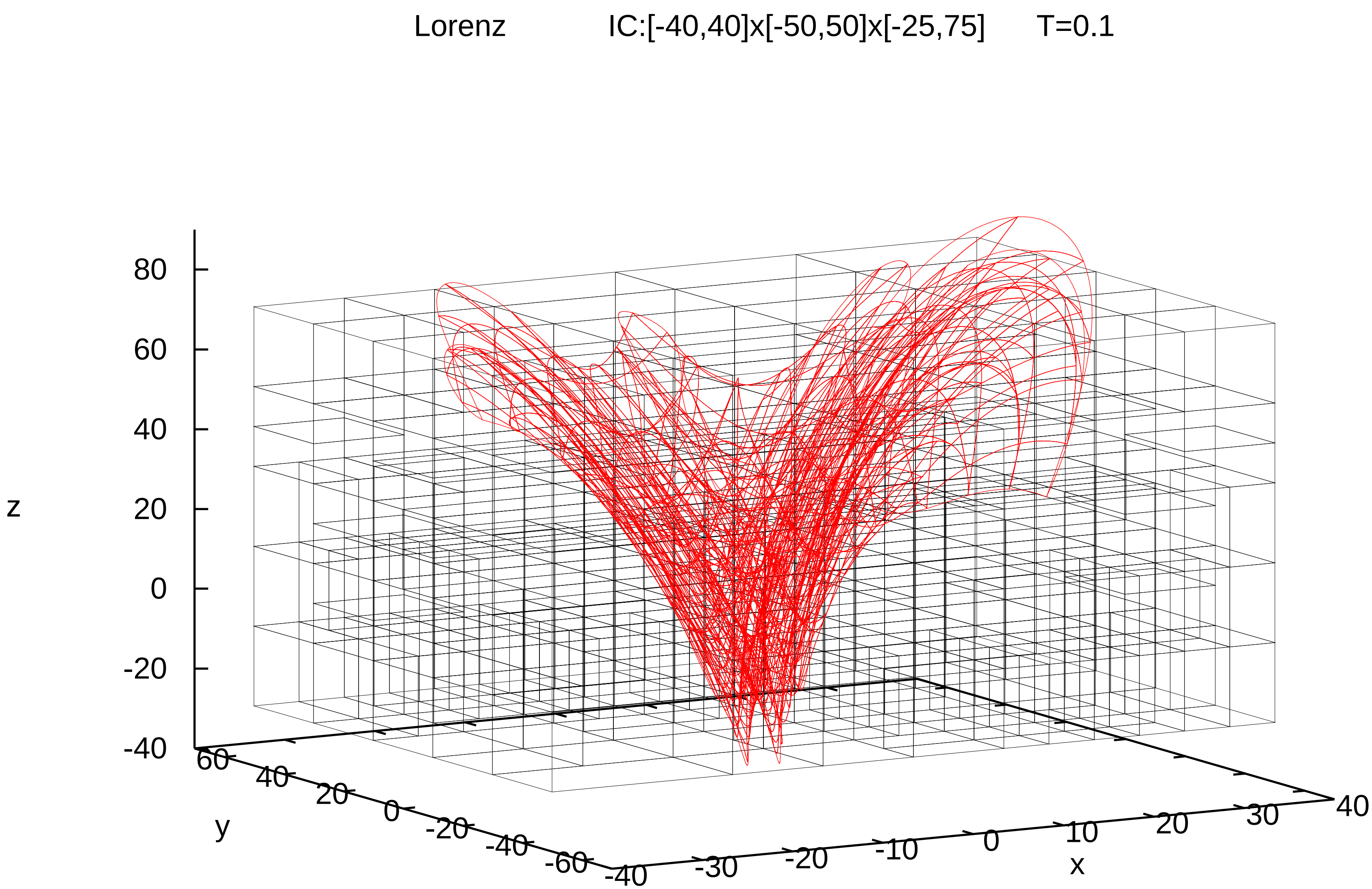
$$x' = 10(y - x), \quad y' = x(28 - z) - y, \quad z' = xy - \frac{8}{3}z$$
$$x_0 = 15 \pm 0.01, \; y_0 = 15 \pm 0.01, \; z_0 = 36 \pm 0.01$$

Solution Ranges by COSY and AWA (18th Order)

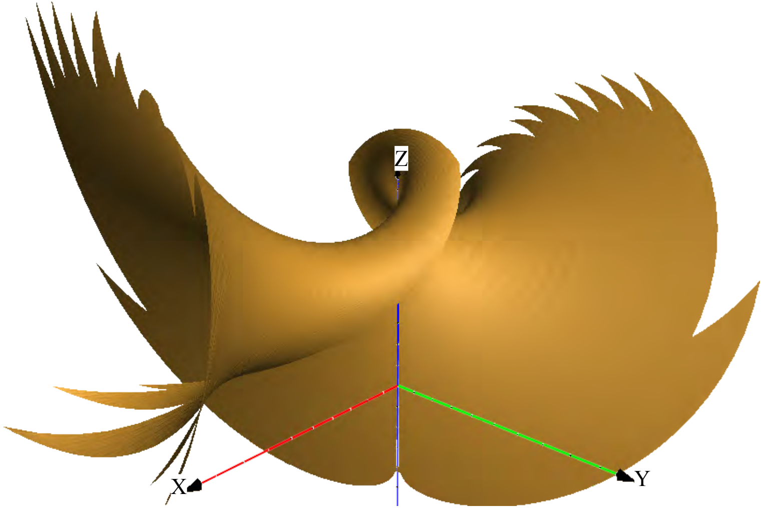




Lorenz, VI integration of IC1 piece and corner point integrations
p1
p2
p3
p4
IC1
z
45
40
35
30
25
20
15
10
5
x
-20
-15
-10
-5
0
5
10
15
20
y
-25
-20
-15
-10
-5
0
5
10
15
20
25


Lorenz IC:[-40,40]x[-50,50]x[-25,75] T=0.1
z
80
60
40
20
0
-20
-40
y
60
40
20
0
-20
-40
-60
x
-40
-30
-20
-10
0
10
20
30
40

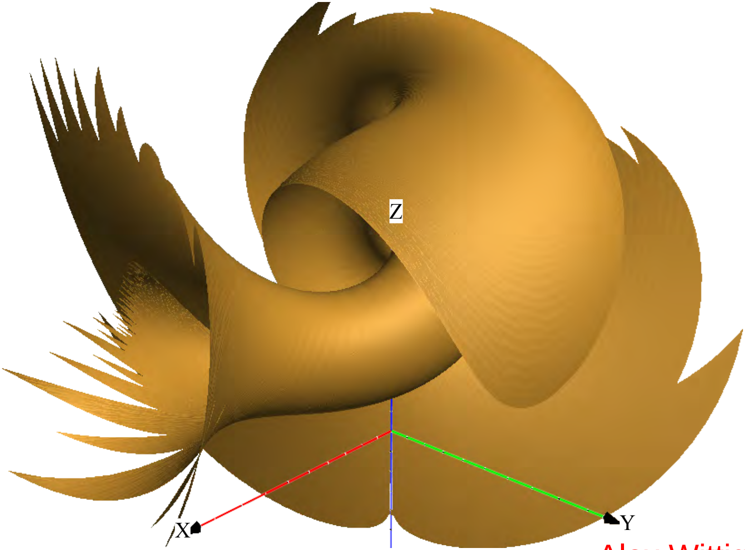
Stable Manifold



Time $t = 0.4$

MICHIGAN STATE UNIVERSITY

Stable Manifold



Time $t = 0.6$

Work in Progress

- Improvement of the Taylor model arithmetic package in COSY to allow arbitrarily high precision Taylor model computations.
  - All the preparation work has been completed.
  - The final system integration work is in progress.
  - Upon the completion, COSY-VI and COSY-GO will be adjusted for utilizing it.
- Improvement of COSY-VI
  - Various schemes to conduct Poincare projections
  - Computations in parallel environment
- Improvement of COSY-GO
  - Utilizing Genetic Algorithm based non-rigorous global optimizers for better cut-off tests
    - Such an optimizer has been implemented in COSY.
      The system integration work has to be done.

High Precision Taylor Models

Interval world Center point with error term:

$$A = \sum_{i=1}^{n} a_i \pm a_{err}$$

Taylor Model world Polynomial with "precision" variable:

$$P(x;p) = (a_{0,0} + a_{0,1} \cdot p + a_{0,2} \cdot p^2) + (a_{1,0} + a_{1,1} \cdot p + a_{1,2} \cdot p^2) \cdot x + \dots$$

- Fits directly into existing code for sparse polynomial storage
- Dynamic precision adjustment for higher order terms

Attractive Fixed Point

$$\mathcal{H} : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} 1 + y - Ax^2 \\ Bx \end{pmatrix}$$

$A = 1.4$ and $B = 0.3$ in standard Hénon map, we use $A = 1.422$ and $B = 0.3$.

Order of attractive fixed point

$f^{(00)}$:	$x = -0.0869282203452\underline{939}$	$y = 0.239153675071\underline{6747}$
$f^{(15)}$:	$x = -0.0869282203454\underline{442}$	$y = 0.239153675071\underline{6964}$
$f^{(30)}$:	$x = -0.0869282203452\underline{939}$	$y = 0.239153675071\underline{6747}$
$f^{(45)}$:	$x = -0.0869282203454\underline{442}$	$y = 0.239153675071\underline{6964}$
$f^{(60)}$:	$x = -0.0869282203452\underline{939}$	$y = 0.239153675071\underline{6747}$
⋮	⋮	⋮

Attractive Fixed Point: Results

Taylor Model Enclosure

$$x = 1.1957^{80721557596}_{58008577504}$$

$$y = 0.050^{52194963414509}_{49328335698421}$$

	Taylor Models	HP Taylor Models	Intervals
Halfwidth	10^{-5}	10^{-60}	10^{-70}
Precision	16	75	75
Boxes	1	1	70,000,000
Time	$<$ 1 sec.	$\sim$ 1 sec.	130 min

Attractive Fixed Point: Results

High Precision Taylor Model Enclosure

$$x = 1.19576936506755033604110098396\,55489352337235594806801053003^{7188}_{6812}$$

$$y = 0.05050761649556464888828848 01\,75616101684142680828370628141 05^{797}_{403}$$

	Taylor Models	HP Taylor Models	Intervals
Halfwidth	10^{-5}	10^{-60}	10^{-70}
Precision	16	75	75
Boxes	1	1	$70,000,000$
Time	$<$ 1 sec.	$\sim$ 1 sec.	130 min.

Attractive Fixed Point: Results

High Precision Interval Enclosure

$$x = 1.19576936506755033604110098396554893523372355948068010530037073508396 8^{32853}_{10139}$$

$$y = 0.05050761649556464888828848017561610168414268082837062814105551657822 9^{4397960}_{1531331}$$

	Taylor Models	HP Taylor Models	Intervals
Halfwidth	10^{-5}	10^{-60}	10^{-70}
Precision	16	75	75
Boxes	1	1	70,000,000
Time	<1 sec.	~ 1 sec.	130 min.

Important TM Algorithms

- **Range Bounding** (Evaluate f as TM, bound polynomial, add remainder bound. LDB, QFB etc.)
- **Global Optimization** (Use TM bounding schemes, obtain good cutoff values quickly by using non-verified schemes)
- **Quadrature** (Evaluate f as TM, integrate polynomial and remainder bound)
- **Implicit Equations** (Obtain TMs for implicit solutions of TM equations)
- **Superconvergent** Interval Newton Method (Application of Implicit Equations)
- **Implicit ODEs and DAEs**
- **Complex Arithmetic**
- **ODEs** (Obtain TMs describing dependence of final coordinates on initial coordinates)