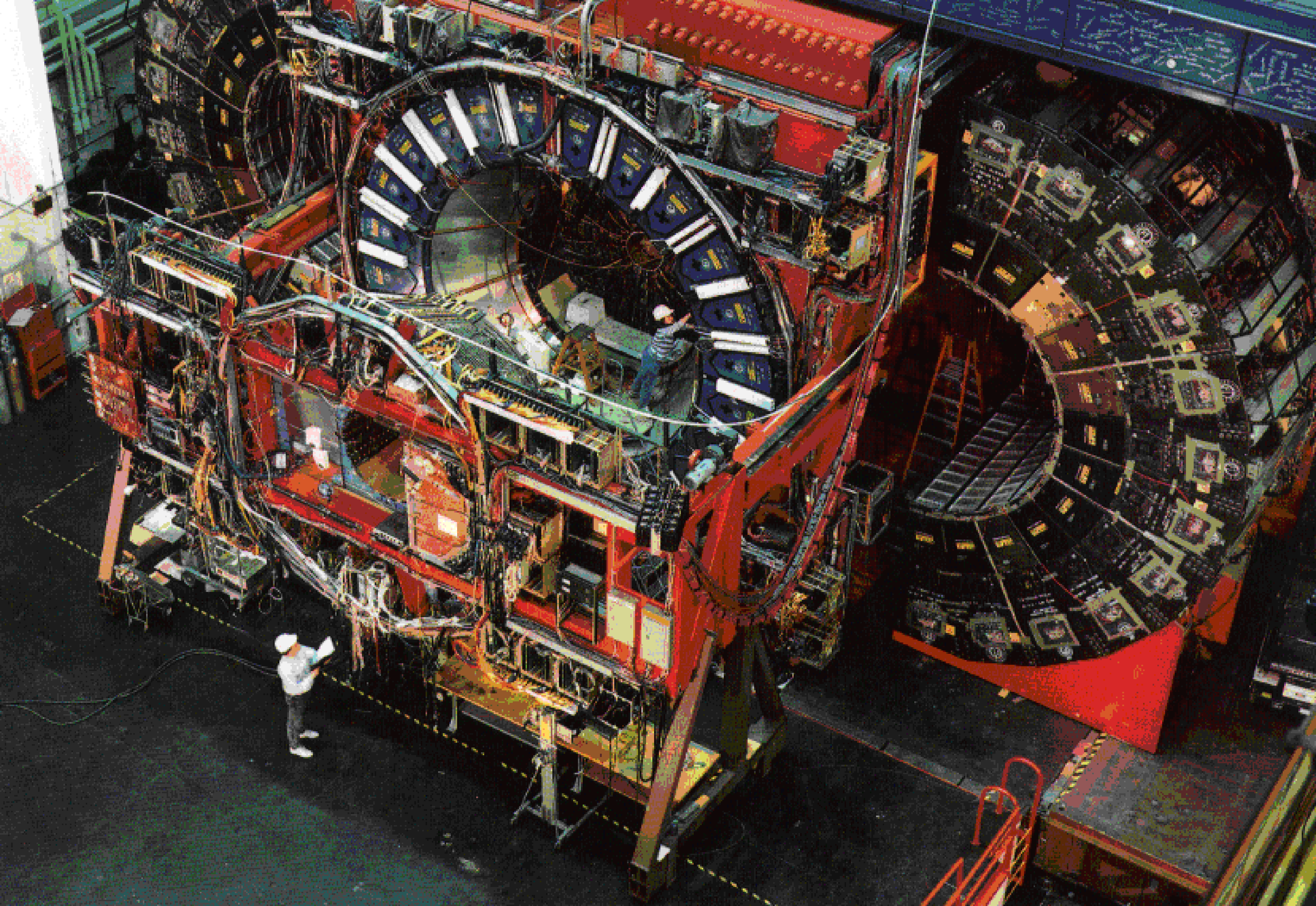
AS-FOUND
COMPLETE



The Henon Map

Henon Map: frequently used elementary example that exhibits many of the well-known effects of nonlinear dynamics, including chaos, periodic fixed points, islands and symplectic motion. The dynamics is two-dimensional, and given by

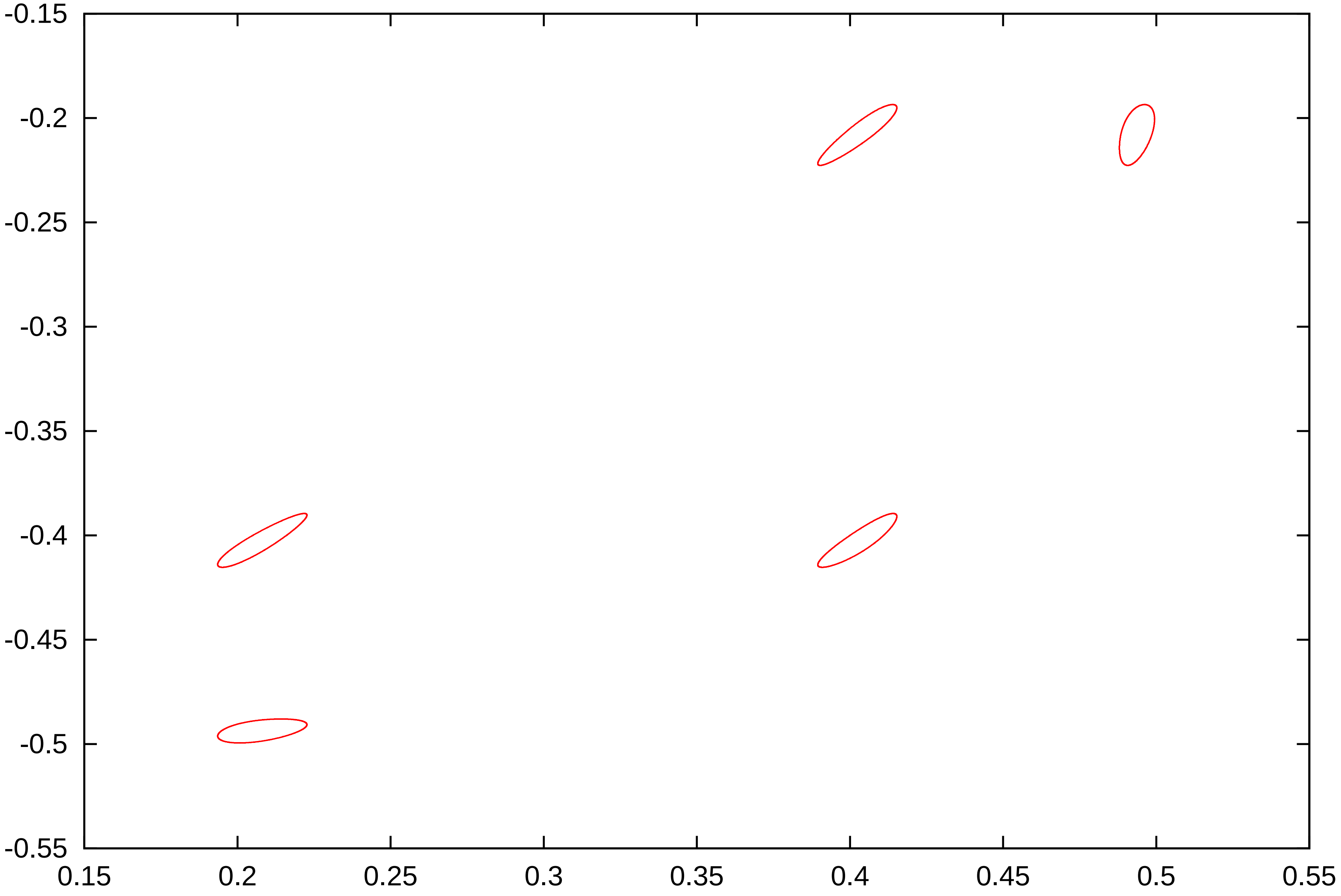
$$\begin{aligned} x_{n+1} &= 1 - \alpha x_n^2 + y_n \\ y_{n+1} &= \beta x_n. \end{aligned}$$

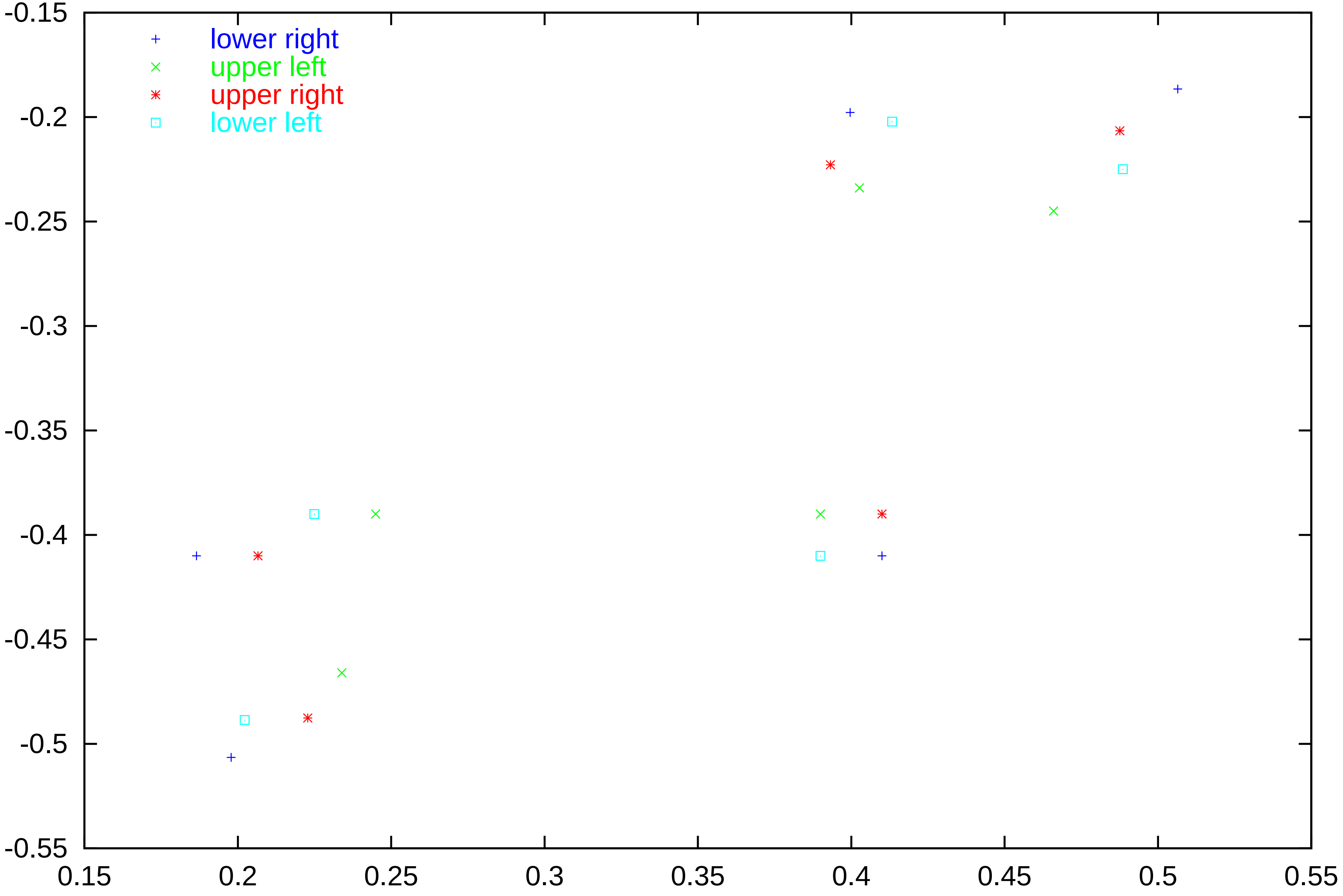
It can easily be seen that the motion is area preserving for $|\beta| = 1$.We consider

$$\alpha = 2.4 \text{ and } \beta = -1,$$

and concentrate on initial boxes of the from $(x_0, y_0) \in (0.4, \ -0.4) + [-d, d]^2$.

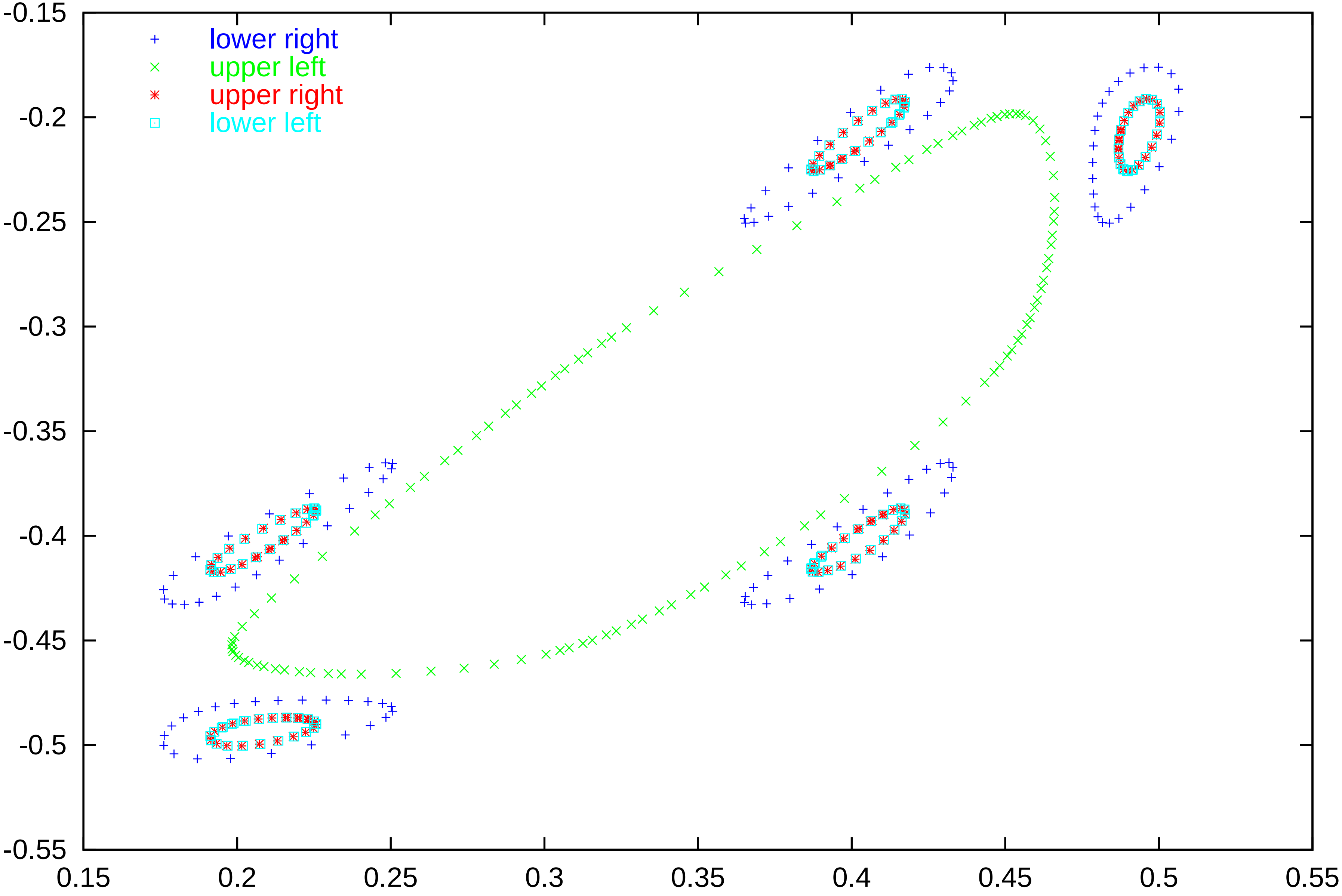
Henon system, xn = 1-2.4*x^2+y, yn = -x, the positions at each step

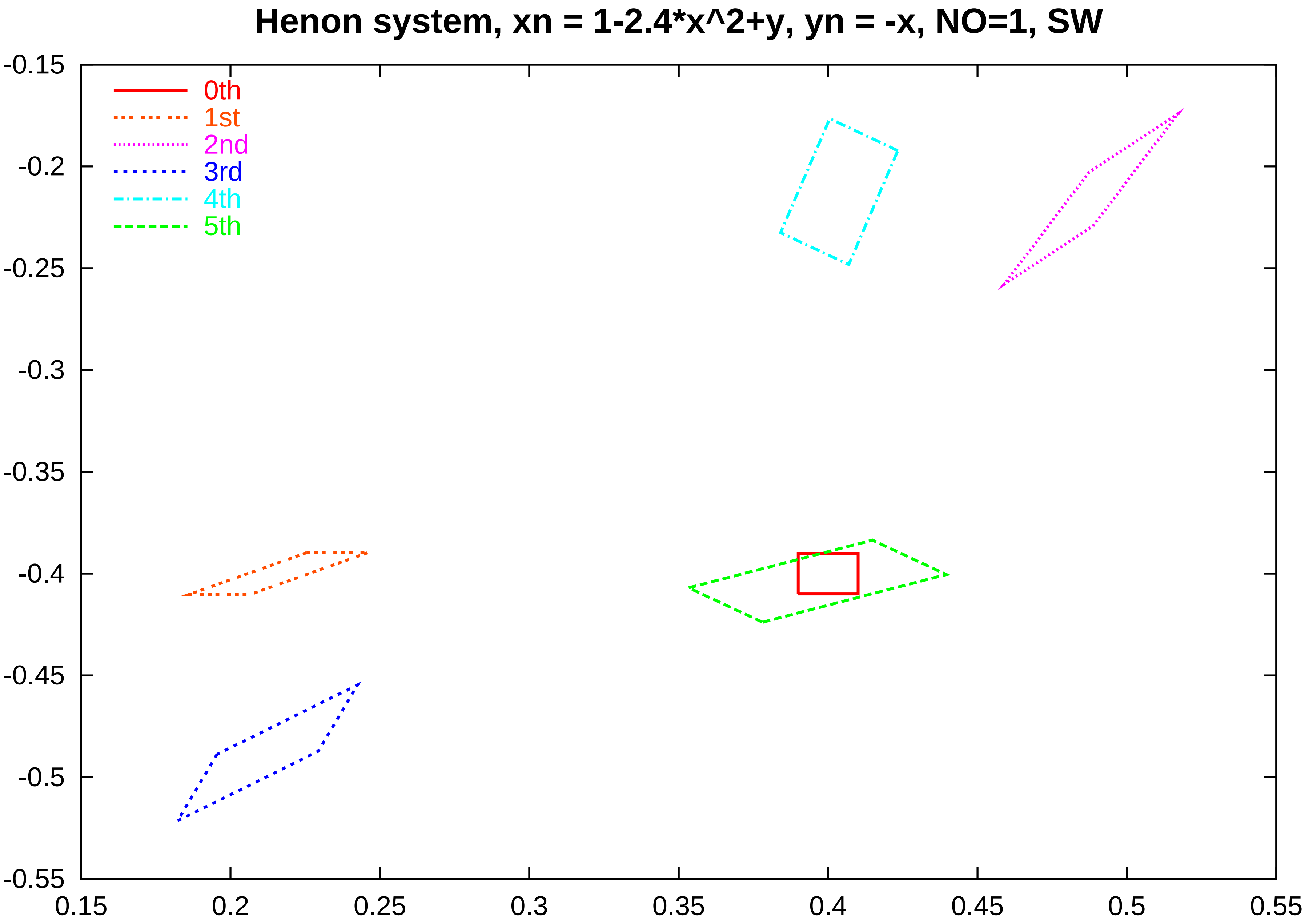


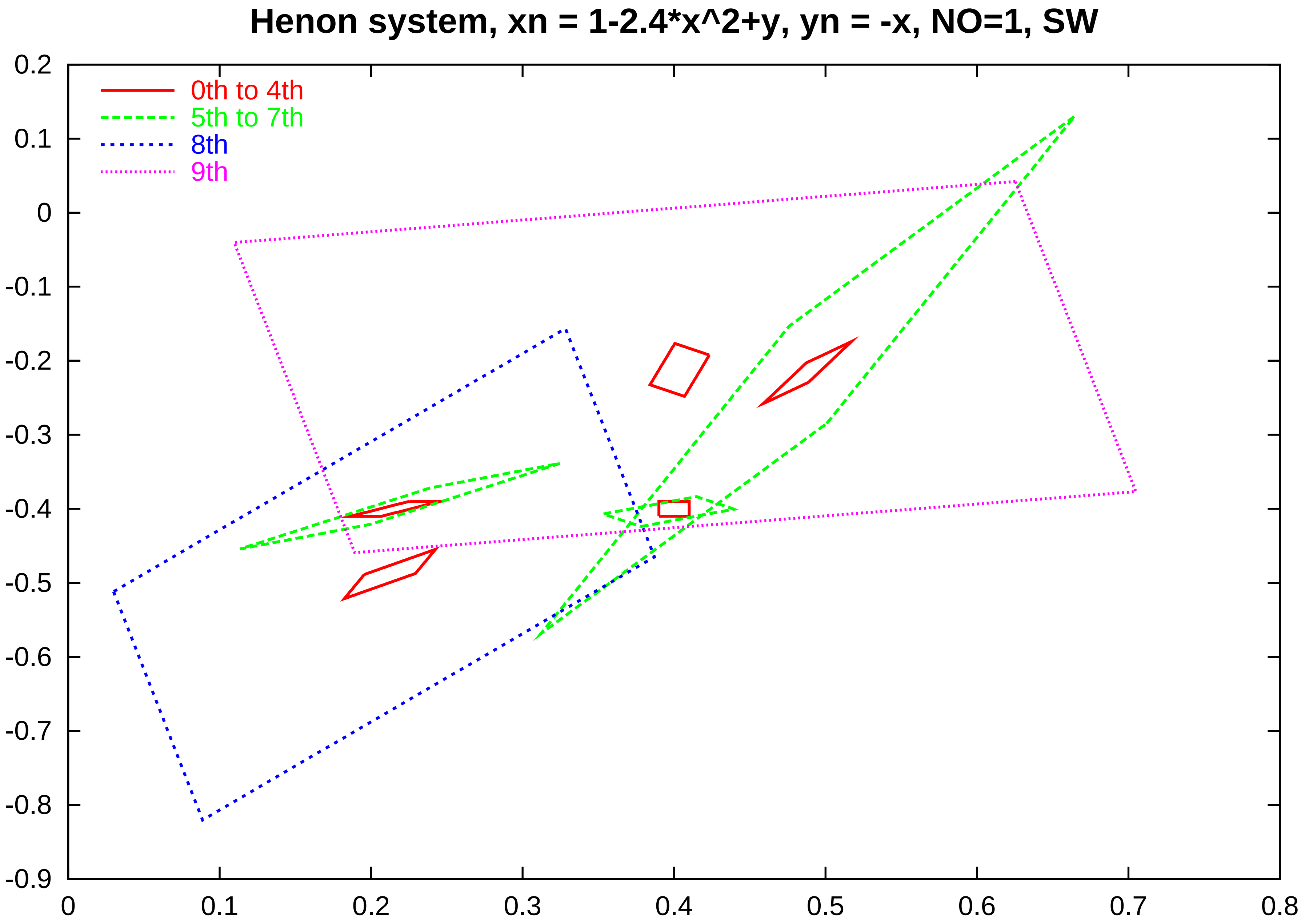


Henon system, xn = 1-2.4*x^2+y, yn = -x, corner points (+-0.01) the first 5 steps

Henon system, xn = 1-2.4*x^2+y, yn = -x, corner points (+-0.01) the first 120 steps




Henon system, xn = 1-2.4*x^2+y, yn = -x, NO=1, SW
0th
1st
2nd
3rd
4th
5th
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55


Henon system, xn = 1-2.4*x^2+y, yn = -x, NO=1, SW
0th to 4th
5th to 7th
8th
9th
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
-0.8
-0.9
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

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

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

where $0 < \theta < 1$.

Definitions - Taylor Model Exponential, cont.

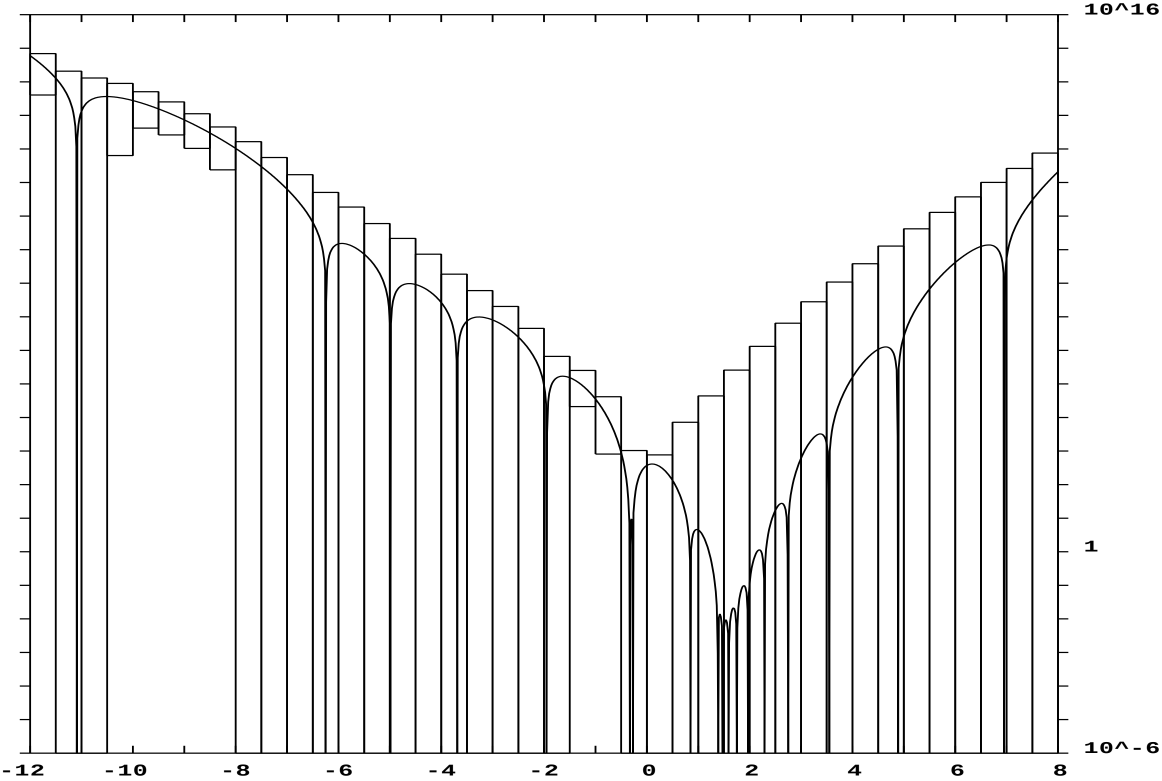
Taking $k \geq n$, the part

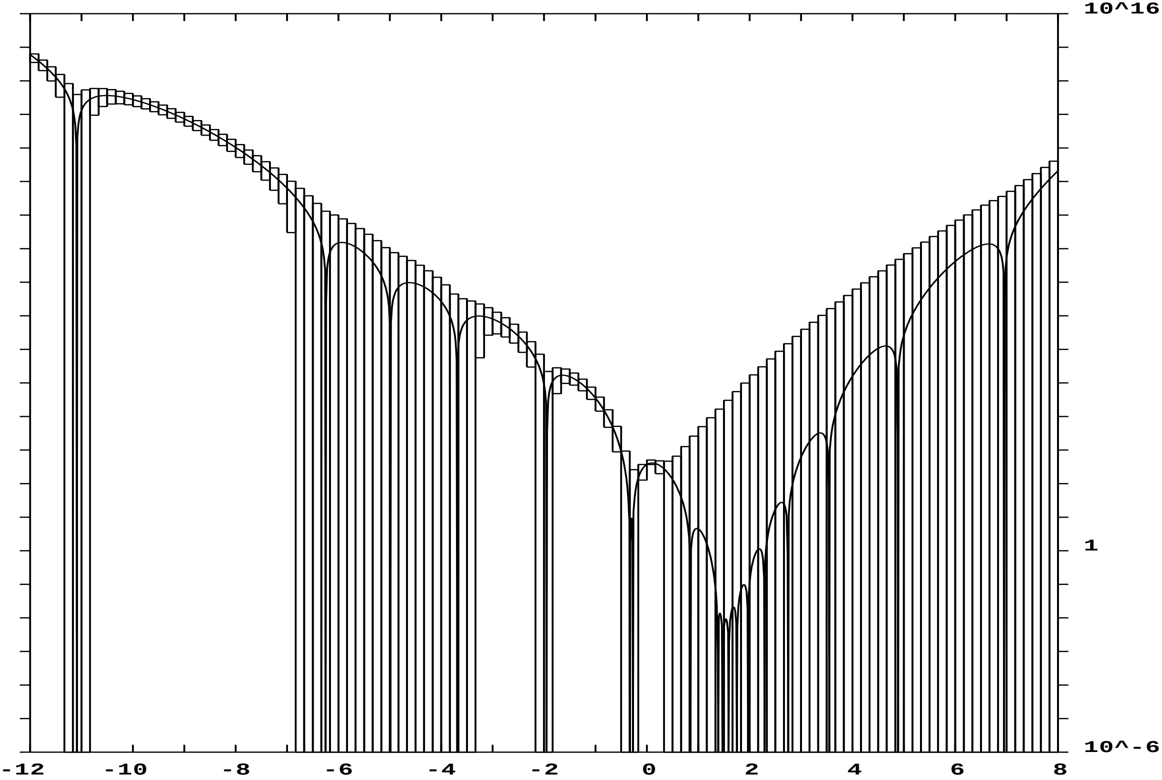
$$\exp(c_f) \cdot \left\{ 1 + \bar{f}(x) + \frac{1}{2!}(\bar{f}(x))^2 + \cdots + \frac{1}{n!}(\bar{f}(x))^n \right\}$$

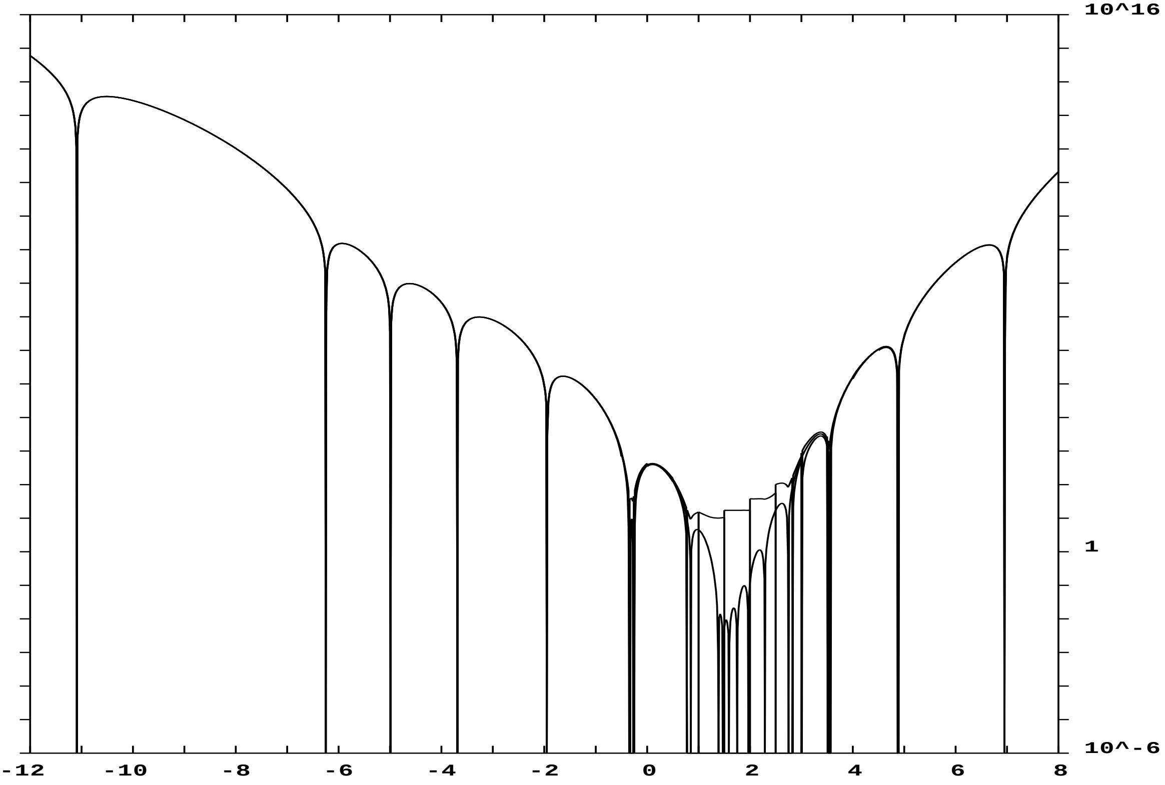
is merely a polynomial of $\bar{f}$, of which we can obtain the Taylor model via Taylor model addition and multiplication. The remainder part of $\exp(f(x))$, the expression

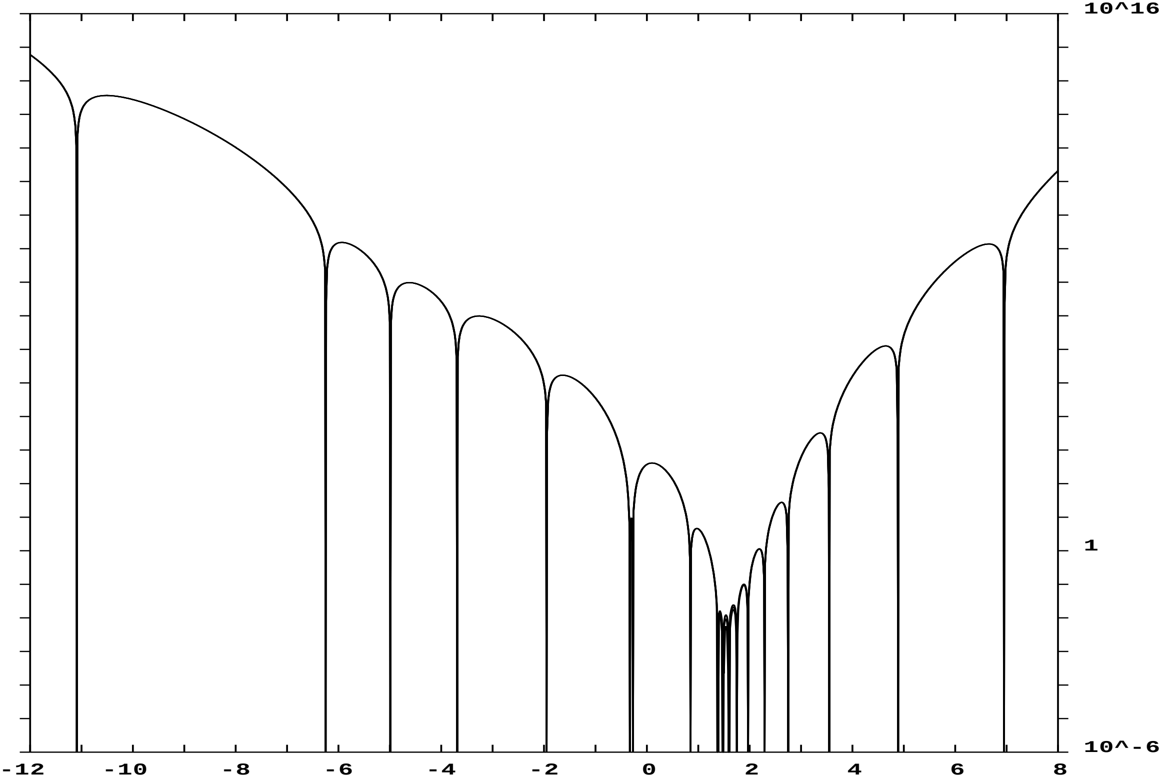
$$\exp(c_f) \cdot \left\{ \frac{1}{(n+1)!}(\bar{f}(x))^{n+1} \right. \\ \left. + \cdots + \frac{1}{(k+1)!}(\bar{f}(x))^{k+1} \exp\left(\theta \cdot \bar{f}(x)\right) \right\},$$

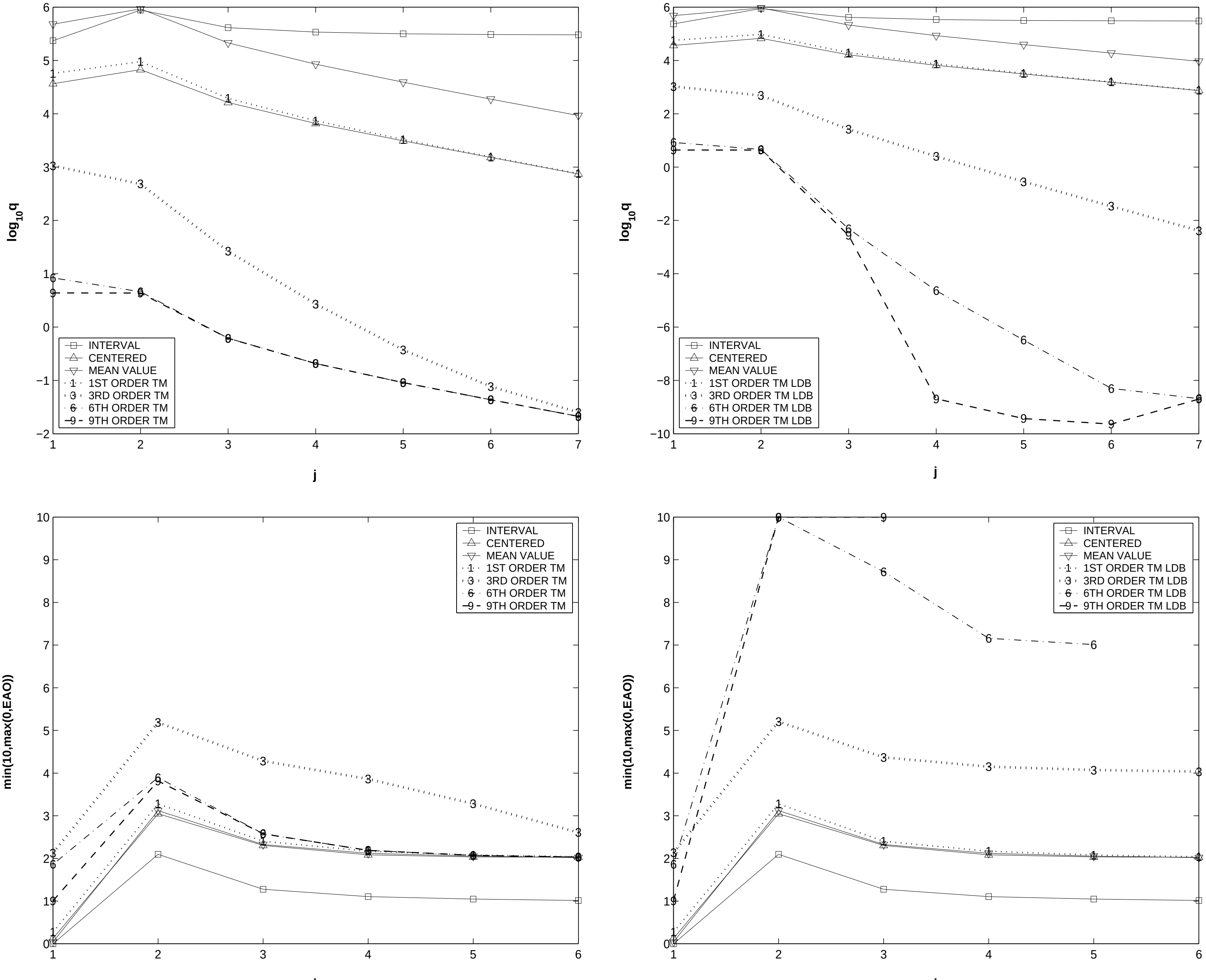
will be bounded by an interval. First observe that since the Taylor polynomial of $\bar{f}$ does not have a constant part, the $(n+1)$-st through $(k+1)$-st powers of the Taylor model $(\bar{P}, I)$ of $\bar{f}$ will have vanishing polynomial part, and thus so does the entire remainder part. The remainder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

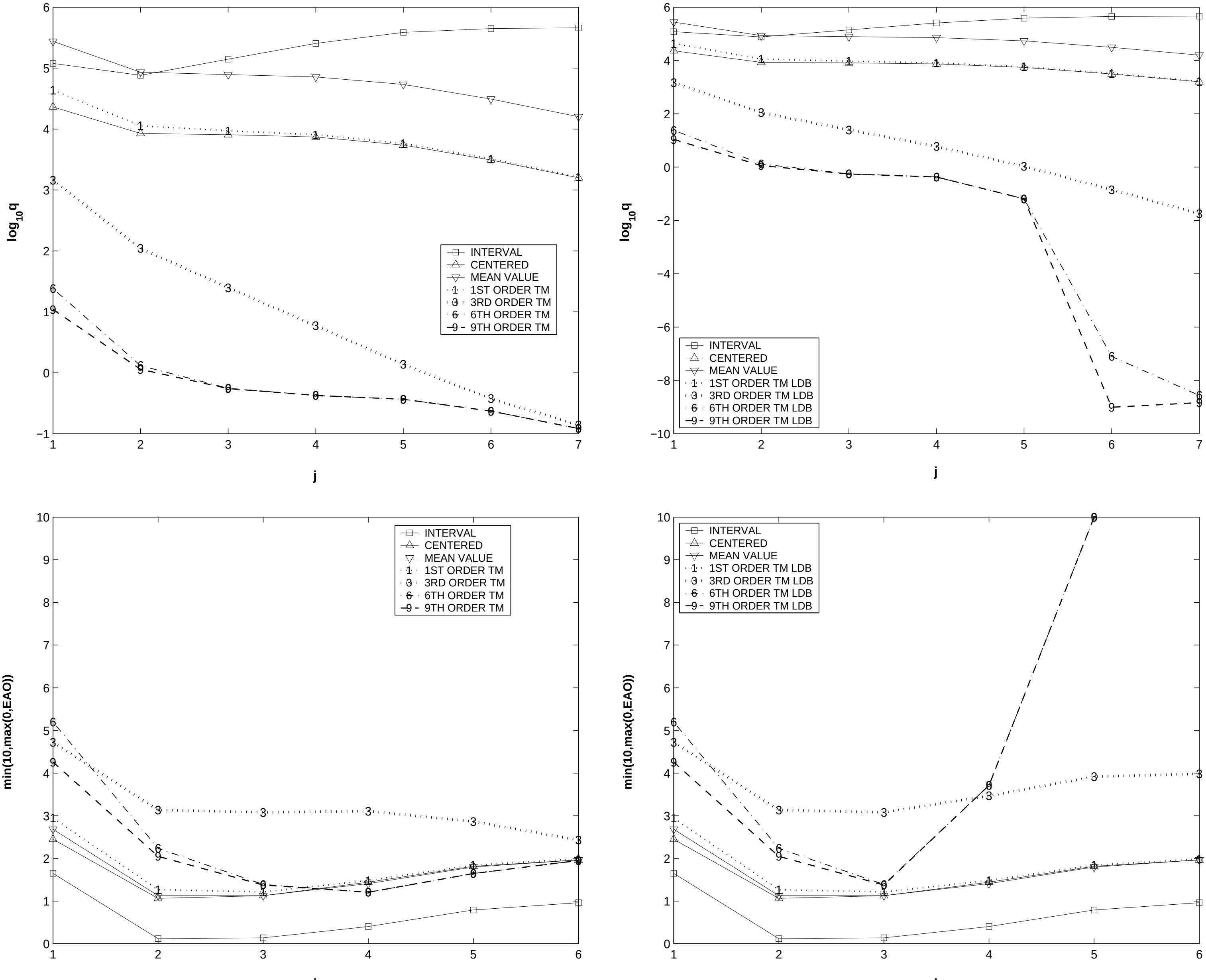

10^16
1
10^-6
-12
-10
-8
-6
-4
-2
0
2
4
6
8


10^16
1
10^-6
-12
-10
-8
-6
-4
-2
0
2
4
6
8


10^16
1
10^-6
-12
-10
-8
-6
-4
-2
0
2
4
6
8


10^16
1
10^-6
-12
-10
-8
-6
-4
-2
0
2
4
6
8


log10q
j
1
2
3
4
5
6
7
6
5
4
3
2
1
0
−1
−2
−4
−6
−8
−10
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB
min(10,max(0,EAO))
10
9
8
7


log10q
j
1
2
3
4
5
6
7
−1
0
1
2
3
4
5
6
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
log10q
j
−10
−8
−6
−4
−2
0
2
4
6
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB
min(10,max(0,EAO))
0
1
2
3
4
5
6
7
8
9
10
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
min(10,max(0,EAO))
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB

The Henon Map

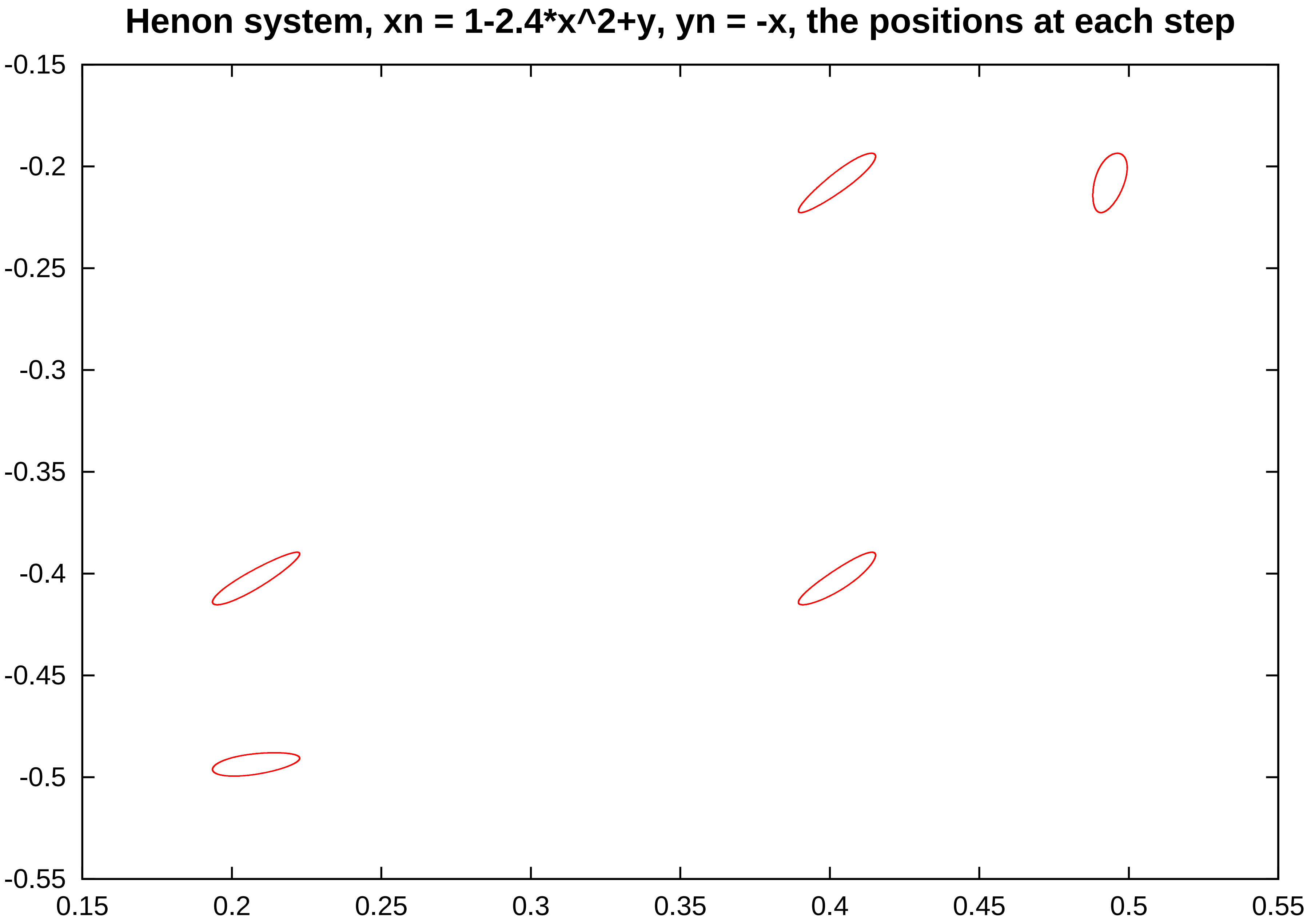
Henon Map: frequently used elementary example that exhibits many of the well-known effects of nonlinear dynamics, including chaos, periodic fixed points, islands and symplectic motion. The dynamics is two-dimensional, and given by

$$\begin{aligned} x_{n+1} &= 1 - \alpha x_n^2 + y_n \\ y_{n+1} &= \beta x_n. \end{aligned}$$

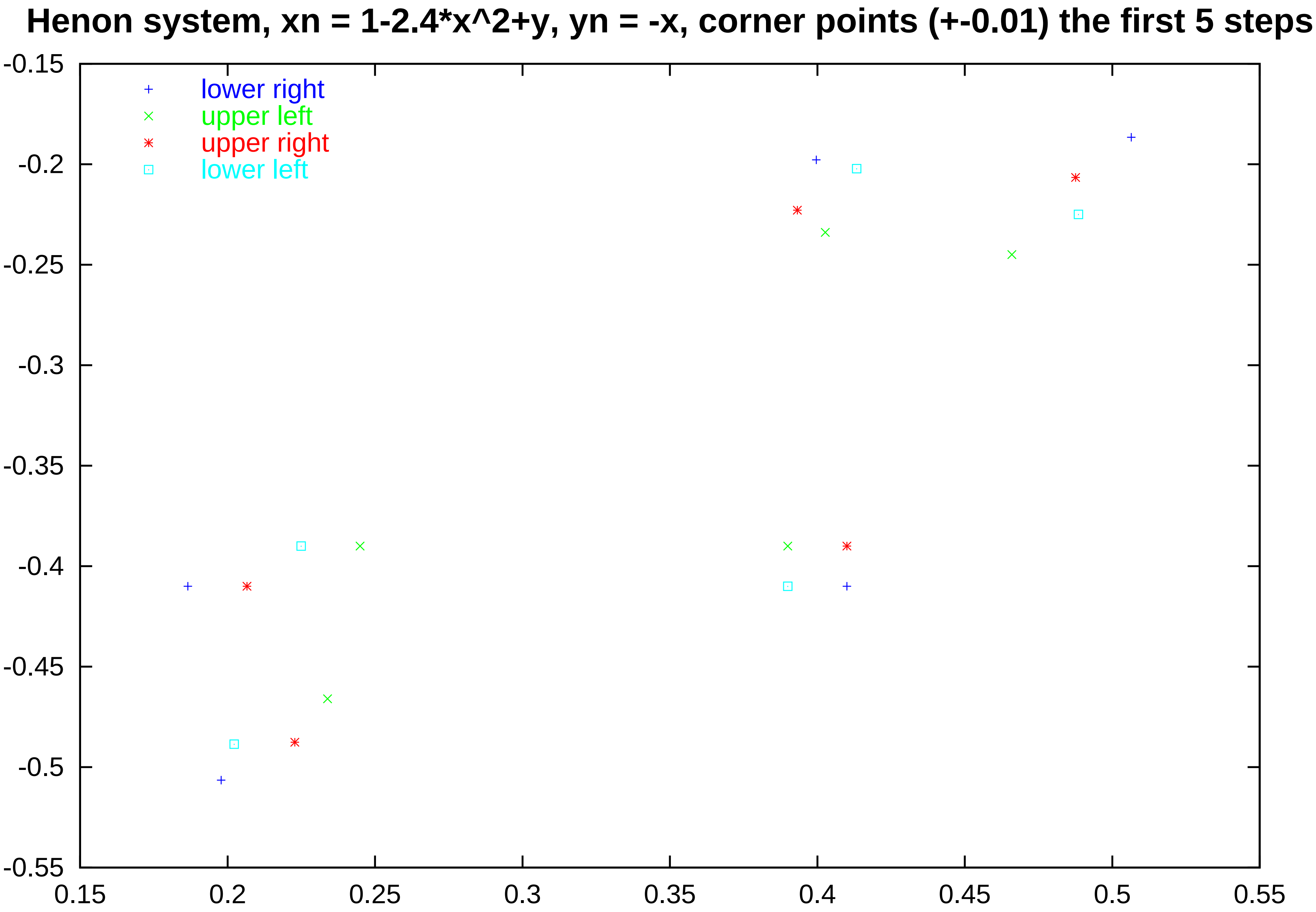
It can easily be seen that the motion is area preserving for $|\beta| = 1$.We consider

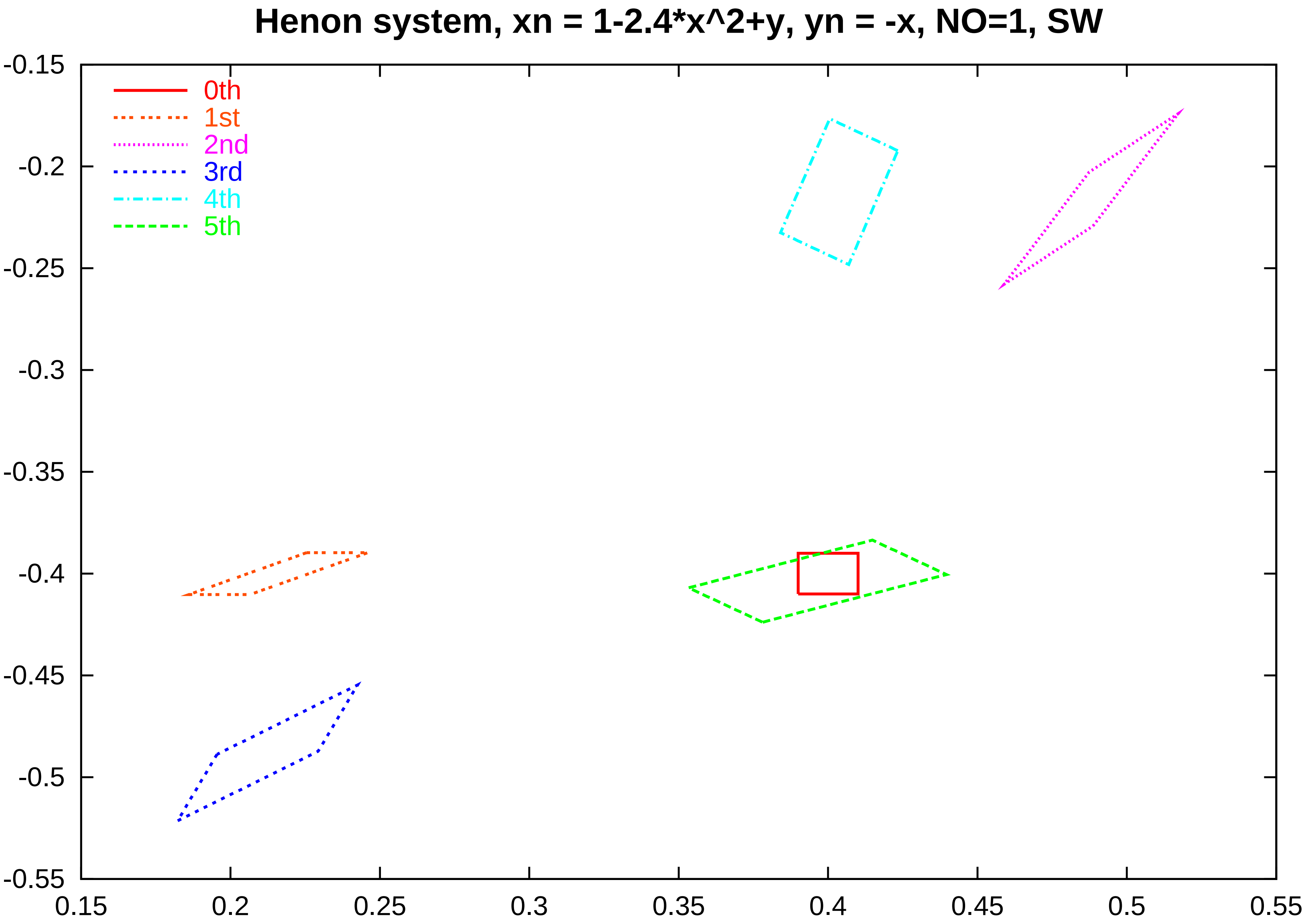
$$\alpha = 2.4 \text{ and } \beta = -1,$$

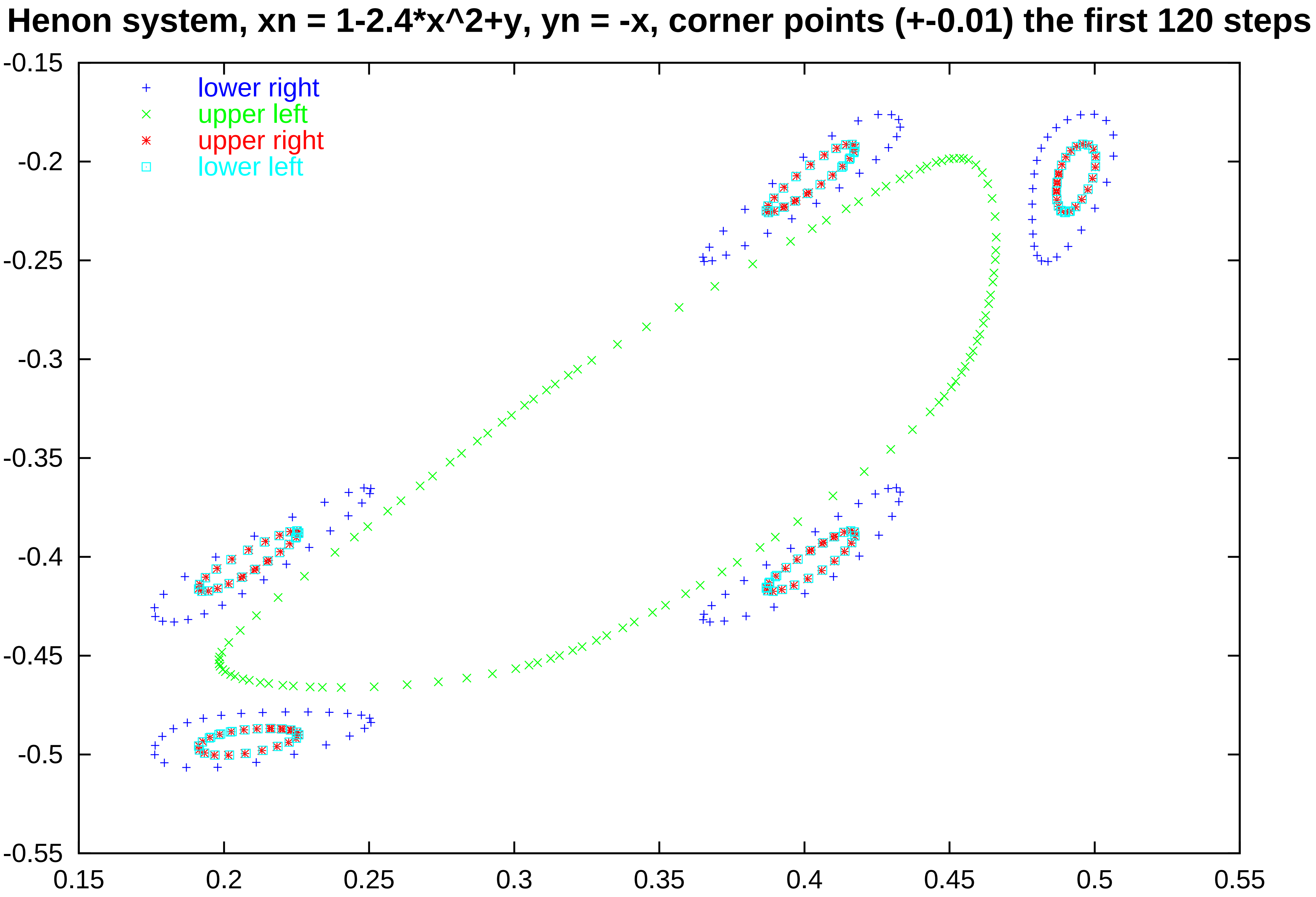
and concentrate on initial boxes of the from $(x_0, y_0) \in (0.4, -0.4) + [-d, d]^2$.

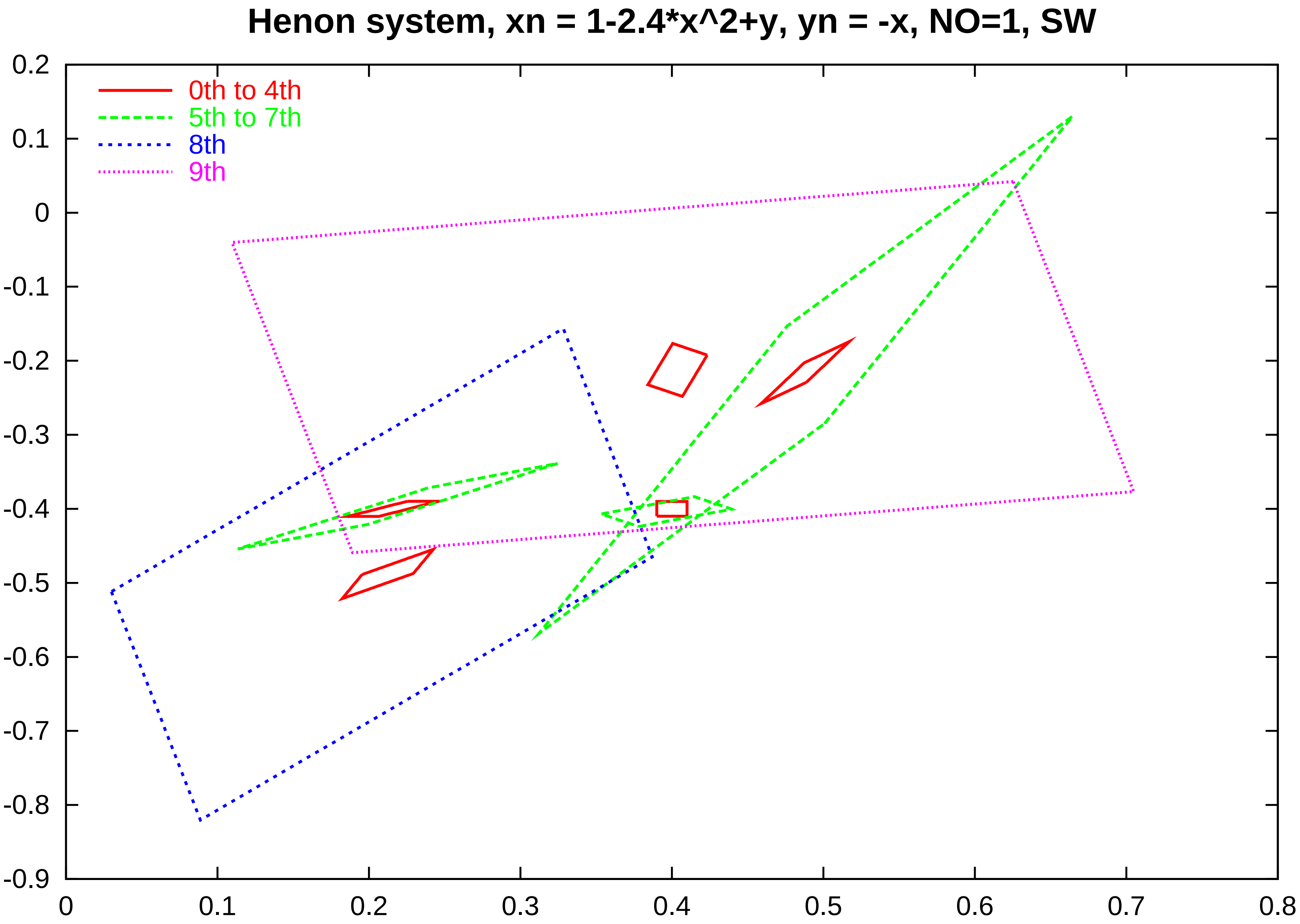


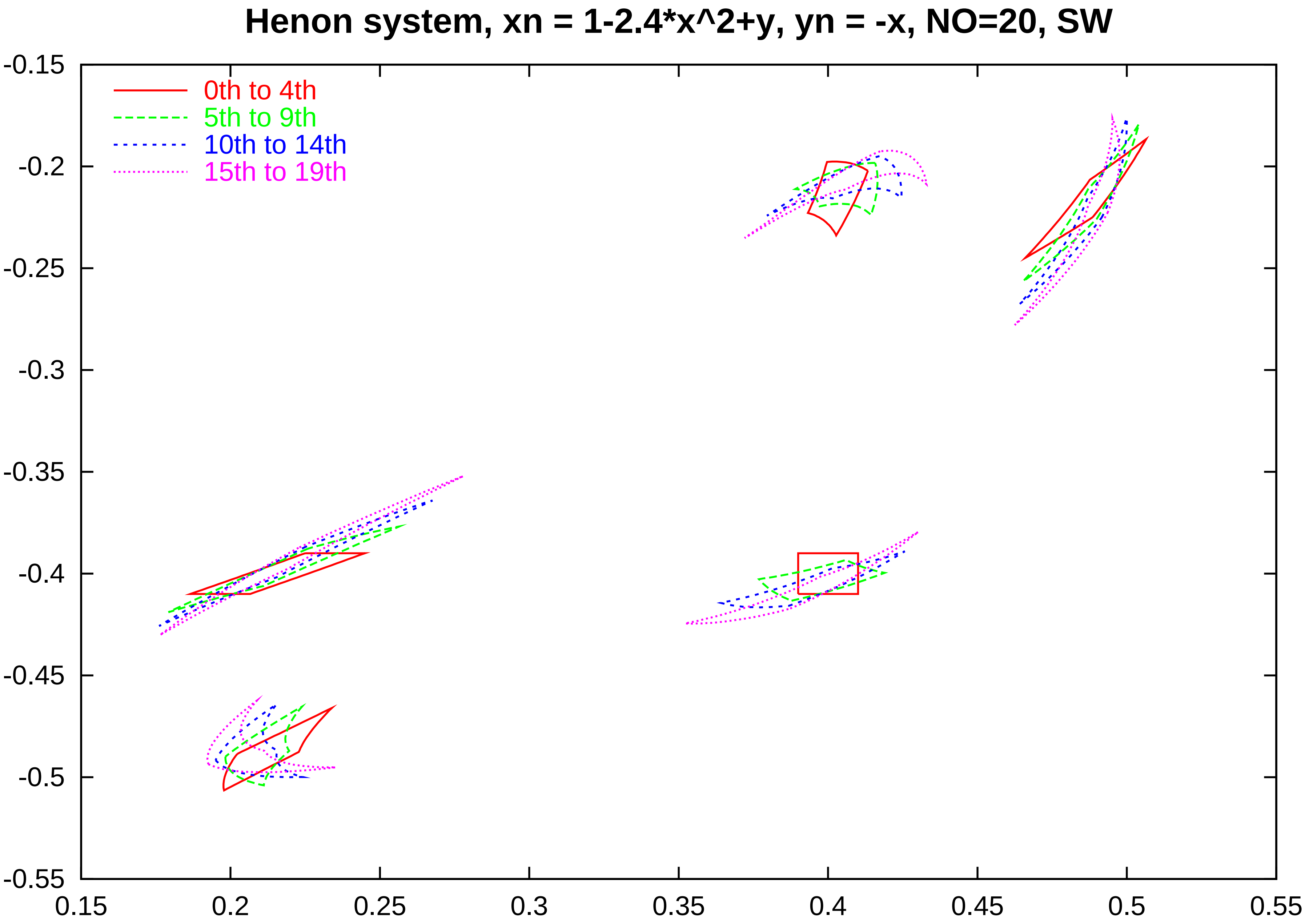
Henon system, xn = 1-2.4*x^2+y, yn = -x, the positions at each step

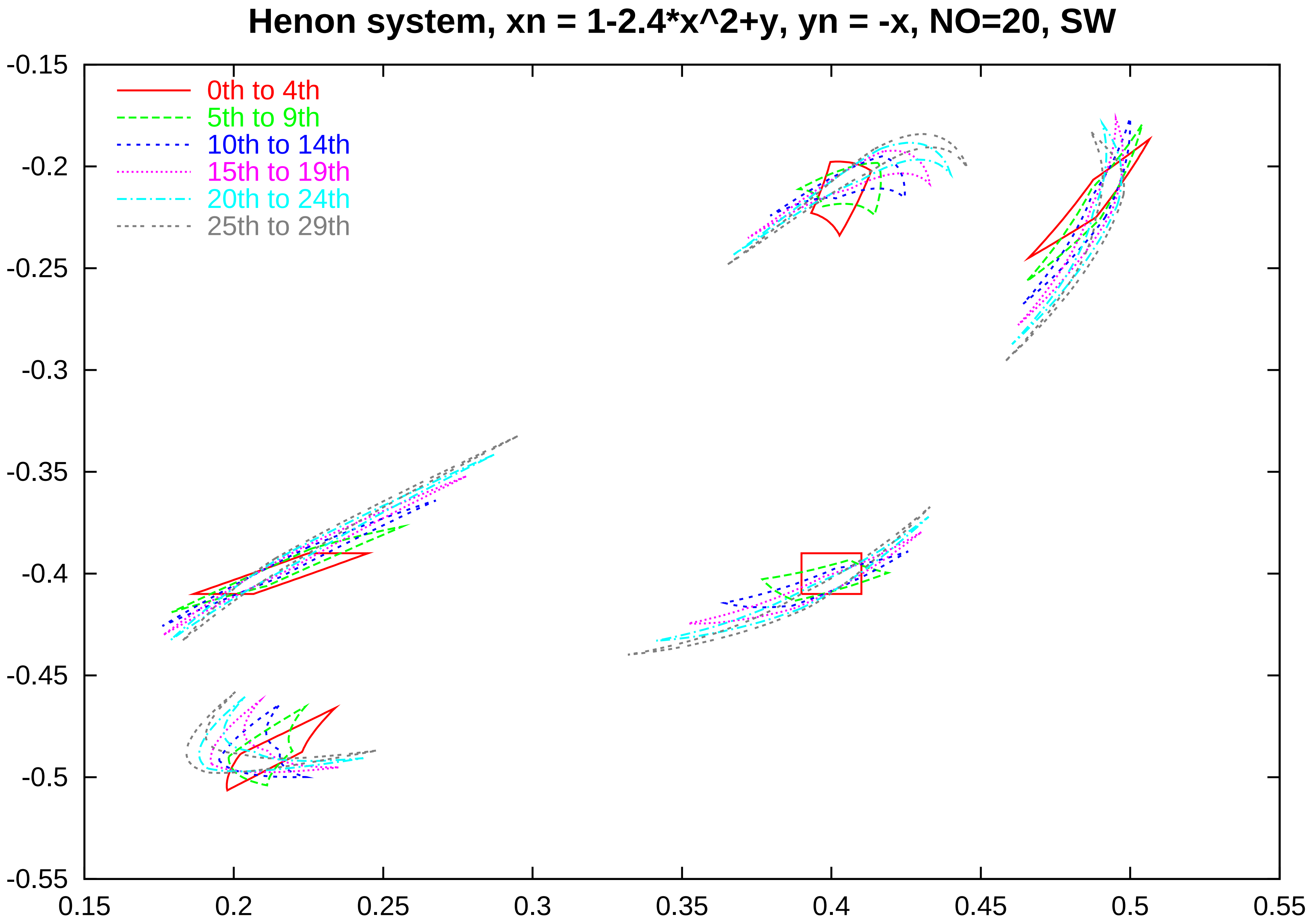

Henon system, xn = 1-2.4*x^2+y, yn = -x, corner points (+-0.01) the first 5 steps
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
+ lower right
× upper left
✳ upper right
□ lower left


Henon system, xn = 1-2.4*x^2+y, yn = -x, NO=1, SW
0th
1st
2nd
3rd
4th
5th
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55


Henon system, xn = 1-2.4*x^2+y, yn = -x, corner points (+-0.01) the first 120 steps
+ lower right
× upper left
✳ upper right
□ lower left
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55


Henon system, xn = 1-2.4*x^2+y, yn = -x, NO=1, SW
0th to 4th
5th to 7th
8th
9th
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
-0.8
-0.9
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


Henon system, xn = 1-2.4*x^2+y, yn = -x, NO=20, SW
0th to 4th
5th to 9th
10th to 14th
15th to 19th
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55


Henon system, xn = 1-2.4*x^2+y, yn = -x, NO=20, SW
0th to 4th
5th to 9th
10th to 14th
15th to 19th
20th to 24th
25th to 29th
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55

Taylor Model Integration

Definition

For an n-th order Taylor model $T = (P, I)$ and $k = 1, \ldots, v$, let

$$Q_{\xi_k} = \int_0^{x_k} P_{(n-1)}\left(x_1, \ldots, x_{k-1}, \xi_k, x_{k+1}, \ldots, x_v\right) d\xi_k .$$

The antiderivative $\partial_{\xi_k}^{-1}$ of T is defined by

$$\partial_{\xi_k}^{-1}(P, I) = \left(Q_{\xi_k}, \left(B\left(P_{(n)} - P_{(n-1)}\right) + I\right) \cdot 2\right).$$

- D is $[-1, 1]^v \subset \mathbb{R}^v$.

Taylor Model Finite Integration Scheme

Finite integration of a Taylor model,

$$J^{(v-1)} = \int_{-1}^{1} T^{v}\left(x_1, \ldots x_{k-1}, \xi_{k,} x_{k+1}, \ldots, x_v\right) d\xi_k,$$

Steps to compute the Taylor model $J^{(v-1)}$ using the antiderivative $\partial_{\xi_k}^{-1}$ operator.

1. Split the Taylor Model $T^v = (P, I)$ in to a polynomial P of order n and interval I.
2. Construct a new Taylor model $G^v = (P, I_\epsilon)$, where $I_\epsilon = [0, 0]$.
3. Apply the antiderivative operator $\partial_{\xi_k}^{-1}$ to the Taylor model G^v.
4. Evaluate the Taylor models $\partial_{\xi_k}^{-1} G^v$ (at $\xi_k = 1$) and $\partial_{\xi_k}^{-1} G^v$ (at $\xi_k = -1$) and subtract them to obtain a new Taylor model R^{v-1}.
5. Add the interval $2 \cdot I$ to the Taylor model R^{v-1} to give the Taylor model after integration with respect to the ξ_k variable.

The steps 1 through 5 can be repeated for the integration in more than one variable.

The Poisson Problem

Goal: Determine an accurate solution of the Poisson equation when derivatives of the solution are specified on the boundary

$$
\begin{aligned}
\nabla^2 \phi' (\vec{r}) &= \rho(\vec{r}) \text{ in the bounded volume } \Omega \subset \mathbb{E}^3 \\
\nabla \phi' (\vec{r}) &= \vec{g}(\vec{r}) \text{ on the surface } \partial\Omega
\end{aligned}
$$

- Motivation: Modelling space charge effects in accelerators, extraction of transfer maps

Splitting the problem

$$\phi^{'}\left(\vec{r}\right) = \underbrace{\phi\left(\vec{r}\right)}_{Laplace} + \underbrace{\psi\left(\vec{r}\right)}_{Poisson}$$

- The potential $\psi\left(\vec{r}\right)$ satisfies the Poisson equation with open boundary condition

$$\nabla^2\psi\left(\vec{r}\right) = \rho\left(\vec{r}\right) \text{ in the bounded volume } \Omega \subset \mathbb{E}^3$$

- The potential $\phi\left(\vec{r}\right)$ is the solution to the Laplace equation

$$\begin{aligned} \nabla^2\phi\left(\vec{r}\right) &= 0 \text{ in the bounded volume } \Omega \subset \mathbb{E}^3 \\ \nabla\phi\left(\vec{r}\right) &= \vec{g}\left(\vec{r}\right) - \nabla\psi\left(\vec{r}\right) \text{ on the surface } \partial\Omega \end{aligned}$$

- Poisson BVP translates to solving the Laplace BVP and the Poisson equation with open boundary condition
- The above splitting is good for electrostatic problems. However, similar treatment can be done for magnetostatic problems (the potential satisfies Ampere's equation). The method developed hold for both the electrostatic and magnetostatic case.

Poisson equation with open boundary condition

- The solution for ψ is given by

$$\psi\left(\vec{r}\right)=\int_{V}\frac{1}{4\pi\epsilon_{0}}\frac{\rho\left(\vec{r}^{\prime}\right)}{\left|\vec{r}-\vec{r}^{\prime}\right|}d\Omega^{\prime},$$

- For $\vec{r}\neq\vec{r}^{\prime}$ multipole expansion solution can be easily obtained by Taylor expanding the kernel and performing finite volume integration over the resulting Taylor model (example next slide)
- The integral is singular at $\vec{r}=\vec{r}^{\prime}$. Needs special technique to treat the case $\left|\vec{r}-\vec{r}^{\prime}\right|\rightarrow 0$.
  - Few schemes have been tried but the results only good at low orders. Problem needs further investigation.

The Laplace BVP

$$
\begin{aligned}
\nabla^2\phi(\vec{r}) &= 0 \text{ in the bounded volume } \Omega \subset \mathbb{E}^3 \\
\nabla\phi(\vec{r}) &= \vec{g}(\vec{r}) \text{ on the surface } \partial\Omega
\end{aligned}
$$

Goal:

- Provide solution as local expansion of the field ($\phi(\overrightarrow{r})$ and $\partial^n_{x_i}\phi(\overrightarrow{r})$)
- Highly accurate and work for case with large variation of field in the region of interest
- Computationally inexpensive
- Provide information about the field quality of measured data

Analytic closed form solution can only be found for few problems with certain regular geometries (separation of variables method, power series, finite Fourier transform)

Numerical Methods

- Finite Difference, Finite element methods
  - Numerical solution as data set in the region of interest
  - Relatively low approximation order
  - Often large number of mesh points and careful meshing required
  - Usually multipole expansion of the field can not be computed
- Methods using surface data
  - Boundary integral methods and source-based field models
    - ★ Require knowledge of Green's function for the problem
    - ★ Field inside of a source free volume due to a real sources outside of it can be exactly replicated by a distribution of fictitious sources on its surface. Error due to discretization of the source falls off rapidly as the field point moves away from the source.
  - Methods using the **Helmholtz theorem**

The Helmholtz Theorem

Any vector field $\overrightarrow{B}$ that vanishes at infinity can be written as the sum of two terms, one of which is called "irrotational" and the other "solenoidal" as

$$
\begin{aligned}
\overrightarrow{B}\left(\vec{x}\right) &= \vec{\nabla}\times\vec{A}_t\left(\vec{x}\right)+\vec{\nabla}\phi_n\left(\vec{x}\right) \text{ where} \\
\phi_n\left(\vec{x}\right) &= \frac{1}{4\pi}\int_{\partial\Omega}\frac{\vec{n}\left(\vec{x}_s\right)\cdot\overrightarrow{B}\left(\vec{x}_s\right)}{\left|\vec{x}-\vec{x}_s\right|}ds-\frac{1}{4\pi}\int_{\Omega}\frac{\vec{\nabla}\cdot\overrightarrow{B}\left(\vec{x}_v\right)}{\left|\vec{x}-\vec{x}_v\right|}dV \\
\vec{A}_t\left(\vec{x}\right) &= -\frac{1}{4\pi}\int_{\partial\Omega}\frac{\vec{n}\left(\vec{x}_s\right)\times\overrightarrow{B}\left(\vec{x}_s\right)}{\left|\vec{x}-\vec{x}_s\right|}ds+\frac{1}{4\pi}\int_{\Omega}\frac{\vec{\nabla}\times\overrightarrow{B}\left(\vec{x}_v\right)}{\left|\vec{x}-\vec{x}_v\right|}dV
\end{aligned}
$$

$\partial\Omega$ is a surface which bounds the volume Ω
$\vec{x}_s$ and $\vec{x}_v$ denote points on $\partial\Omega$ and within Ω
$\vec{\nabla}$ denotes the gradient with respect to $\vec{x}_v$
$\vec{n}$ is a unit normal vector pointing away from $\partial\Omega$

- If $\vec{B}$ is the magnetic/electric field in the source free region, we have $\vec{\nabla} \times \overrightarrow{B}(\vec{x}_v) = 0$ and $\vec{\nabla} \cdot \overrightarrow{B}(\vec{x}_v) = 0$, and the volume integral terms vanish
- $\phi_n(\vec{x})$ and $\vec{A}_t(\vec{x})$ are completely determined from the normal and the tangential field data on surface $\partial\Omega$ via

$$\phi_n(\vec{x}) = \frac{1}{4\pi} \int_{\partial\Omega} \frac{\vec{n}(\vec{x}_s) \cdot \overrightarrow{B}(\vec{x}_s)}{|\vec{x} - \vec{x}_s|} ds$$

$$\vec{A}_t(\vec{x}) = -\frac{1}{4\pi} \int_{\partial\Omega} \frac{\vec{n}(\vec{x}_s) \times \overrightarrow{B}(\vec{x}_s)}{|\vec{x} - \vec{x}_s|} ds$$

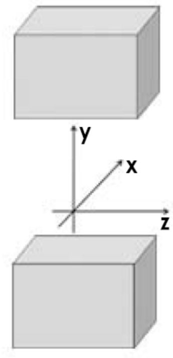
$$\overrightarrow{B}(\vec{x}) = \vec{\nabla} \times \vec{A}_t(\vec{x}) + \vec{\nabla}\phi_n(\vec{x})$$

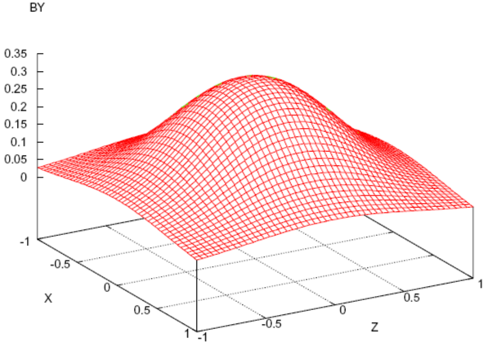
- The Helmholtz theorem can be used to find field directly from the surface field data
- Integral kernels that provides interior fields in terms of the boundary fields or source are smoothing
- Since the expressions are analytic, they can be expanded at least locally

Implementation

1. Discretize the surface $\partial\Omega$ into individual surface cells S_i with centers s_i and the volume Ω into volume cells V_j with centers v_j.
2. Pick a volume cell V_j.
3. For each surface cell S_i, evaluate the kernels for ϕ_n and A_t using Taylor model arithmetic to obtain a Taylor model representations in BOTH the surface variables of S_i AND the volume variables of V_j, i.e. in a total of five variables.
4. Use the Taylor model anti-derivation operation twice to perform integration over the surface variables of each cell S_i.
    - The dependence on the surface variables (x_s, y_s) are integrated over surface sub-cells Γ_i, which results in a highly accurate integration formula
    - The dependence on the volume variables (x, y, z) are retained, which leads to a high order finite element method
    - By using sufficiently high order, high accuracy can be achieved with a small number of surface elements
5. Add up all results to obtain a three dimensional Taylor model enclosing the field $\vec{B}$ over the volume cell V_j.

Analytic example: Bar magnet





- Interior of the magnet: $-0.5 \leq x \leq 0.5$, $|y| \leq 0.5$, and -0.5 $\leq z \leq$ 0.5
- Analytic solution for the magnetic field are know

Analytic solution

$$B_y\left(x,y,z\right) = \frac{B_0}{4\pi}\sum_{i,j=1}^{2}\left(-1\right)^{i+j}\left[\arctan\left(\frac{X_i \cdot Z_j}{Y_+ \cdot R_{ij}^{+}}\right) + \arctan\left(\frac{X_i \cdot Z_j}{Y_- \cdot R_{ij}^{-}}\right)\right]$$

$$B_x\left(x,y,z\right) = \frac{B_0}{4\pi}\sum_{i,j=1}^{2}\left(-1\right)^{i+j}\left[\ln\left(\frac{Z_j + R_{ij}^{-}}{Z_j + R_{ij}^{+}}\right)\right]$$

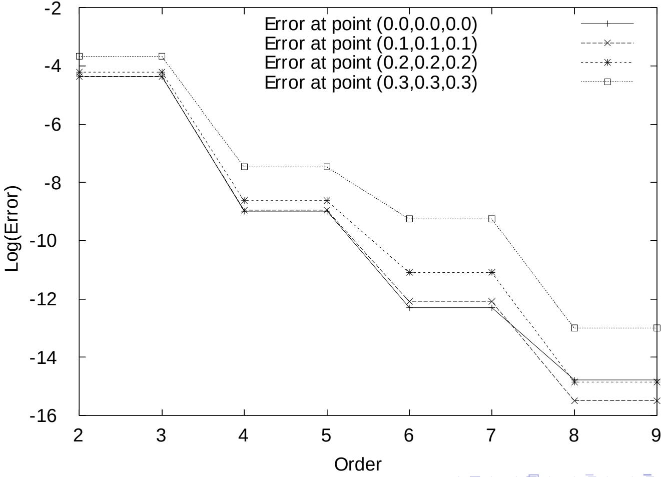
$$B_z\left(x,y,z\right) = \frac{B_0}{4\pi}\sum_{i,j=1}^{2}\left(-1\right)^{i+j}\left[\ln\left(\frac{X_j + R_{ij}^{-}}{X_j + R_{ij}^{+}}\right)\right]$$

where $X_i = x - x_i$, $Y_\pm = y_0 \pm y$, $Z_i = z - z_i$, and $R_{ij}^{\pm} = \left(X_i^2 + Y_j^2 + Z_\pm^2\right)^{\frac{1}{2}}$

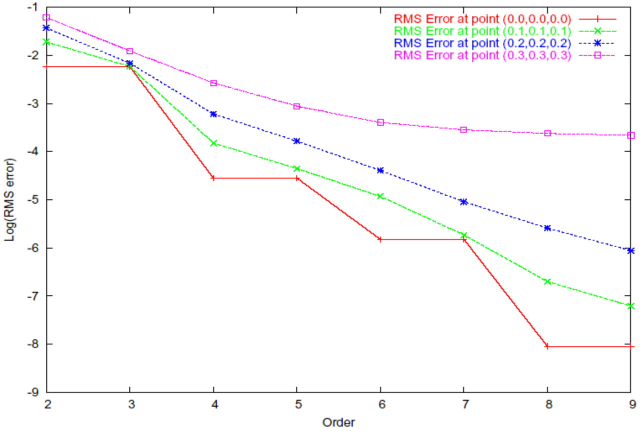
- Using the analytic formulas we specify magnetic field on the surface enclosing the volume of interest
- We use the Helmholtz method to compute the field inside
- We compare the results with the analytic formulas (three plots)

Performance of surface integration method

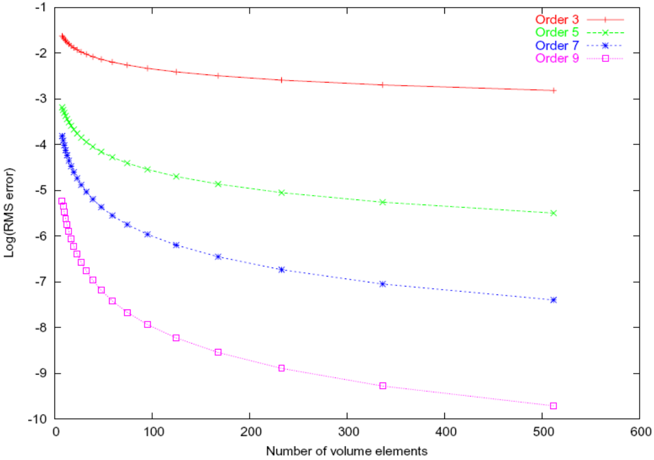
- Choose a cube of edge length 0.8 centered at origin
- each face is covered by 44 × 44 mesh (surface elements)
- Field data is specified on the surface mesh using analytic formulas



- Split the cube into $4 \times 4 \times 4$ volume elements of width 0.2
- Express magnetic field in each volume element by a local expansion about the center of the element
- The RMS average error for 1000 points



Dependency of the average error on the number of volume element.



Parallel implementation

- Contribution due to each surface element is independent of the other surface elements
- The large summation over all the surface elements can be parallelized
- NERSC (National Energy Research Scientific Computing Center) IBM RS6000 Seaborg Cluster consisting of 6080 processors
  - 380 computing nodes with each node having 16 processors (shared memory pool of 16 to 64 GBytes)
  - Communication between the processors within a node is much faster
- Implementation
  - (NPR processors) = ($N2$ groups)×($N1$ processors)
  - $N1 = INT\left(2 \cdot \sqrt{NPR}\right)$
  - Two parallel loop are used to make the summation efficient and also minimizes cross-communication

COSY High-Performance Computing

COSY has been ported to support **large scale computations** for beam dynamics simulation and design optimization, including verified global optimization. Most of the COSY tools have been adopted to **scale nearly linear to about 1000 nodes**. We currently work on

1. **NERSC** (National Energy Research Scientific Computing Center), LBNL

   Part of DOE-funded activities on ODE flow simulations and global optimization, now have a yearly allocation of **500,000 CPU-Hours** for

   - The NERSC 6080 node IBM RS6000 Seaborg Cluster
   - The NERSC 640 node Opteron high-speed Jacquard Cluster

2. **ANL** (Argonne National Laboratory)

   We are funded in connection with optimization and design efforts for next-generation accelerators at ANL. Currently no set time limit.

   - The MCS 2048 node IBM BlueGene/L Cluster

3. **MSU** (Michigan State University)

   - The 64 node SGI Altix 3700 Bx2 Cluster
   - The 512 node AMD Cluster

```
{Loop over N2 groups}
PLOOP JJ 1 N2;

    {Loop over N1 processors}
    PLOOP II 1 N1;

        {Evaluate the processor number PP}
        PP:=II+(JJ-1)*N1;

        [Code to identify the surface elements JBEG through JEND
        for which the processor PP will evaluate the partial sum]

        {Loop to compute the partial sum of the scalar and vector potential
        contributions over surface elements JBEG through JEND}
        LOOP IL JBEG JEND;
                ⋮
            [Code to compute the scalar and the vector potential
            contribution of a surface element IL.]
                ⋮
        ENDLOOP;

    {End the parallel loop over the group of N1 processors and
    send the results to sub-master processor using communication mode 4}
    ENDPLOOP 4 PN1_SCLPOT PN1_VECPOT;

    {Loop to evaluate group partial sum of N1 processors}
    LOOP II 1 N1;
        ⋮
        [Summation to get group partial sum GN2_SCLPOT and GN2_ VECPOT]
        ⋮
    ENDLOOP;

{End the parallel loop over the N2 groups and send the results to master processor}
 ENDPLOOP 4 GN2_SCLPOT GN2_VECPOT;

{Loop to evaluate sum over N2 groups}
LOOP JJ 1 N2;
    ⋮
    [Summation to get sum SCLPOT and VECPOT]
    ⋮
ENDLOOP;

[Code to evaluate the divergence of SCLPOT and the curl of VECPOT and sum them
to get the magnetic field]
```

Table 3.3.1. The code for the parallel algorithm.

Analysis of the Interval part

1. To study the dependency of the Interval part of the potentials and $\vec{B}$ field on the surface element length
    - All of the volume is considered as just one volume element
    - Examine contributions of each surface element towards the total integral
        - ★ Expansion is done at $\vec{r} = (.1, .1, .1)$ and
        - ★ Plot of interval width VS surface element length for scalar potential
    - Plot of interval width VS Order for different surface element length for x component of Magnetic field
2. Study the dependency of the Interval part of the B field on the volume element length
    - The surface element length is locked at $1/128$
    - Plot of interval width VS volume element length for y component of Magnetic field

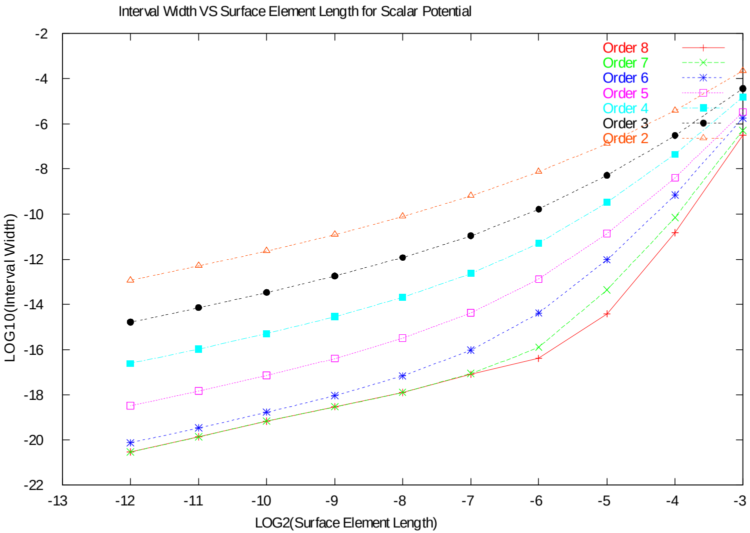


Figure: Integration over single surface element (for ϕ)

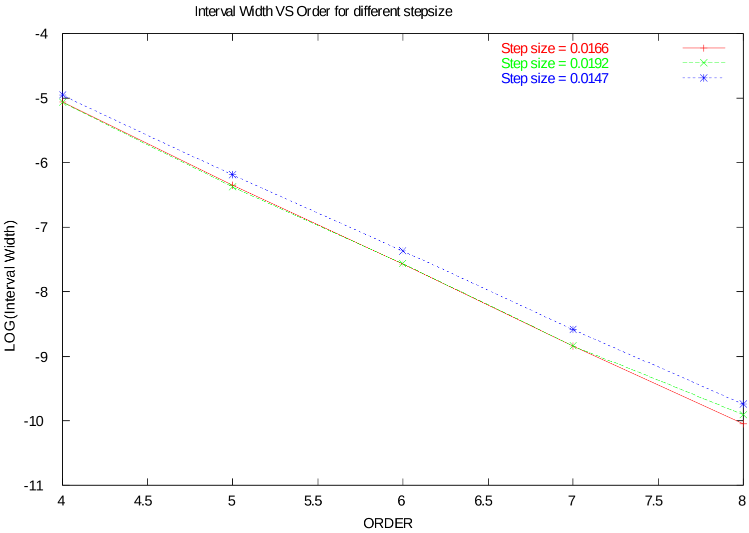


Figure: Interval width VS Order (for different step size)

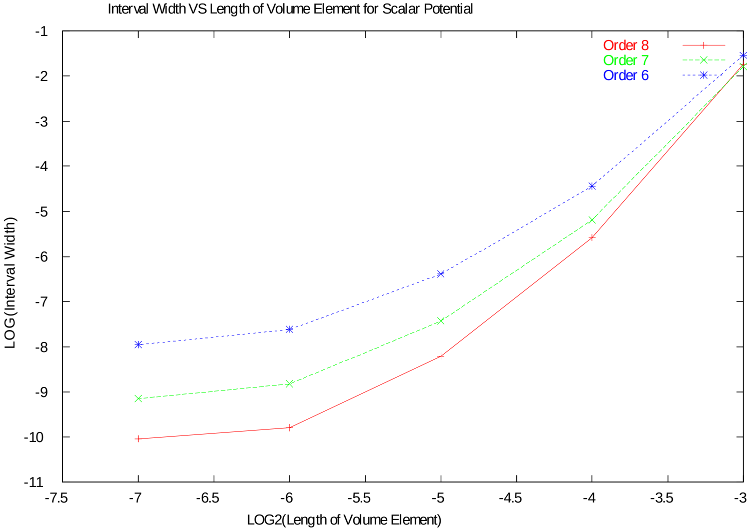
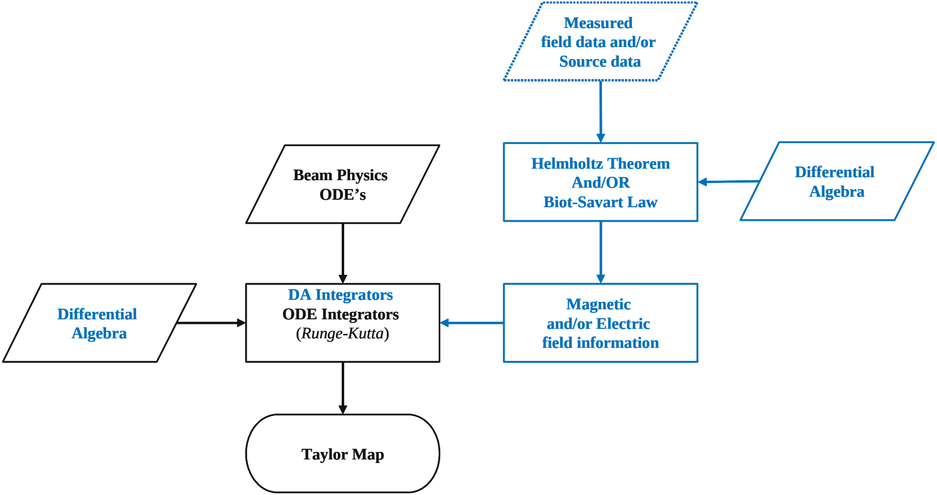


Figure: Interval width VS Volume element length (for ϕ)

Taylor transfer map from field/source information

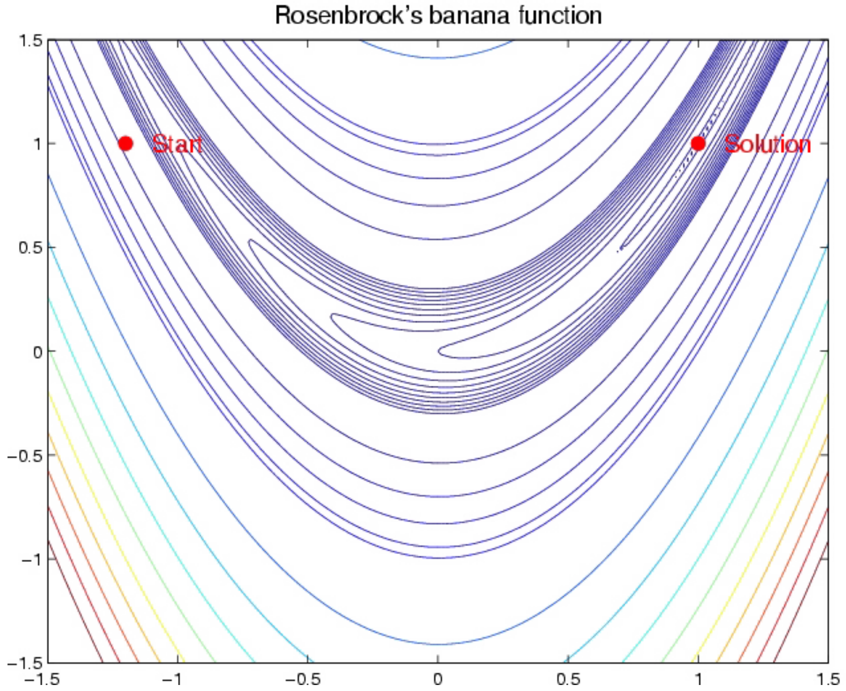


Rosenbrock’s “Banana” Function

$$f(x, y) = 100 \cdot (y - x^2)^2 - (1 - x)^2$$

Study on $[-2, 2] \times [-2, 2]$.

Assumes min 0 at $(1, 1)$, but it is very difficult for gradient methods.



Picture from http://www.math.wm.edu/~buckaroo/classes/csci638/homework/project2.html

Nonvalidated Results of Rosenbrock's Function

Using COSY's true and tested default optimizers:

```
Starting point (X,Y)> (-1.2,1.0)

Optimizer #> 1 : Simplex
Number of steps> 251
F> 0.1711168421282399E-16
(X,Y)> (1.000000004115731 ,1.000000008272989 )

Optimizer #> 2 : LMDIF
Number of steps> 70424
F> 0.9124815296170133E-10
(X,Y)> (0.9999904485839599,0.9999809108990141)

Optimizer #> 3 : Simulated Annealing
Number of steps> 100003
F> 0.5106520406572324E-05
(X,Y)> (0.9977499955081044,0.9954840773134492)
```

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, for each dimensional direction, ∇f is normal to the corresponding box surface, namely for determining x_{mk}, the k-th component of x_m,

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

Quadratic Pruning - The Algorithm I

WLOG, choose $k = n$, where n is the dimensionality of x. The condition for $i \neq n$ is

$$(H \cdot x)_i = 0 \text{ for } \forall i \neq n.$$

Denote the $(n-1)$ dimensional system of H and x, obtained by removing the n-th components from H and x, by $\tilde{H}$ and $\tilde{x}$. Using these, (can be expressed as

$$(H \cdot x)_i = (\tilde{H} \cdot \tilde{x})_i + h_{i,n} x_n = 0.$$

where

$$\tilde{H} = \begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n-1} \\ h_{1,2} & h_{2,2} & \cdots & h_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{1,n-1} & h_{2,n-1} & \cdots & h_{n-1,n-1} \end{pmatrix}, \quad \tilde{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{pmatrix}, \quad \tilde{h}_n = \begin{pmatrix} h_{1,n} \\ h_{2,n} \\ \vdots \\ h_{n-1,n} \end{pmatrix}$$

Combining all the components, we have

$$\tilde{H} \cdot \tilde{x} + x_n \tilde{h}_n = 0.$$

Thus

$$\tilde{x} = -\tilde{H}^{-1} \cdot \tilde{h}_n x_n.$$

Quadratic Pruning - The Algorithm II

Now, under this condition, the function P_2 is simplified as

$$\begin{aligned} P_2 &= \frac{1}{2} x^t \cdot H \cdot x = \frac{1}{2} x_n (H \cdot x)_n \\ &= \frac{1}{2} x_n^2 \left[h_{n,n} - \tilde{h}_n^t \cdot \tilde{H}^{-1} \cdot \tilde{h}_n \right], \end{aligned}$$

which contains only x_n among all the components of x. Here, the last expression is derived as

$$(H \cdot x)_n = \tilde{h}_n^t \cdot \tilde{x} + h_{n,n} x_n = \tilde{h}_n^t \cdot \left(-\tilde{H}^{-1} \cdot \tilde{h}_n x_n \right) + h_{n,n} x_n .$$

Demand the function value to be a when $x_n = x_{mn.}$ From above, such x_{mn} can be determined as

$$x_{mn} = \sqrt{\frac{2a}{h_{n,n} - \tilde{h}_n^t \cdot \tilde{H}^{-1} \cdot \tilde{h}_n}} .$$

Quadratic Pruning - Example I

Consider

$$f(x,y) = 2x^2 + y^2 = \frac{1}{2}(x,y)\cdot\begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}\cdot\begin{pmatrix} x \\ y \end{pmatrix}$$

Let us demand the pruning value $a = 1$. We have

$$x_m = \sqrt{\frac{2\cdot 1}{4 - 0\cdot\frac{1}{2}\cdot 0}} = \sqrt{\frac{1}{2}}, \quad y_m = \sqrt{\frac{2\cdot 1}{2 - 0\cdot\frac{1}{4}\cdot 0}} = 1.$$

Quadratic Pruning - Example II

The next example function is created by rotating the above $f(x,y)$ by $30°$. The function is now

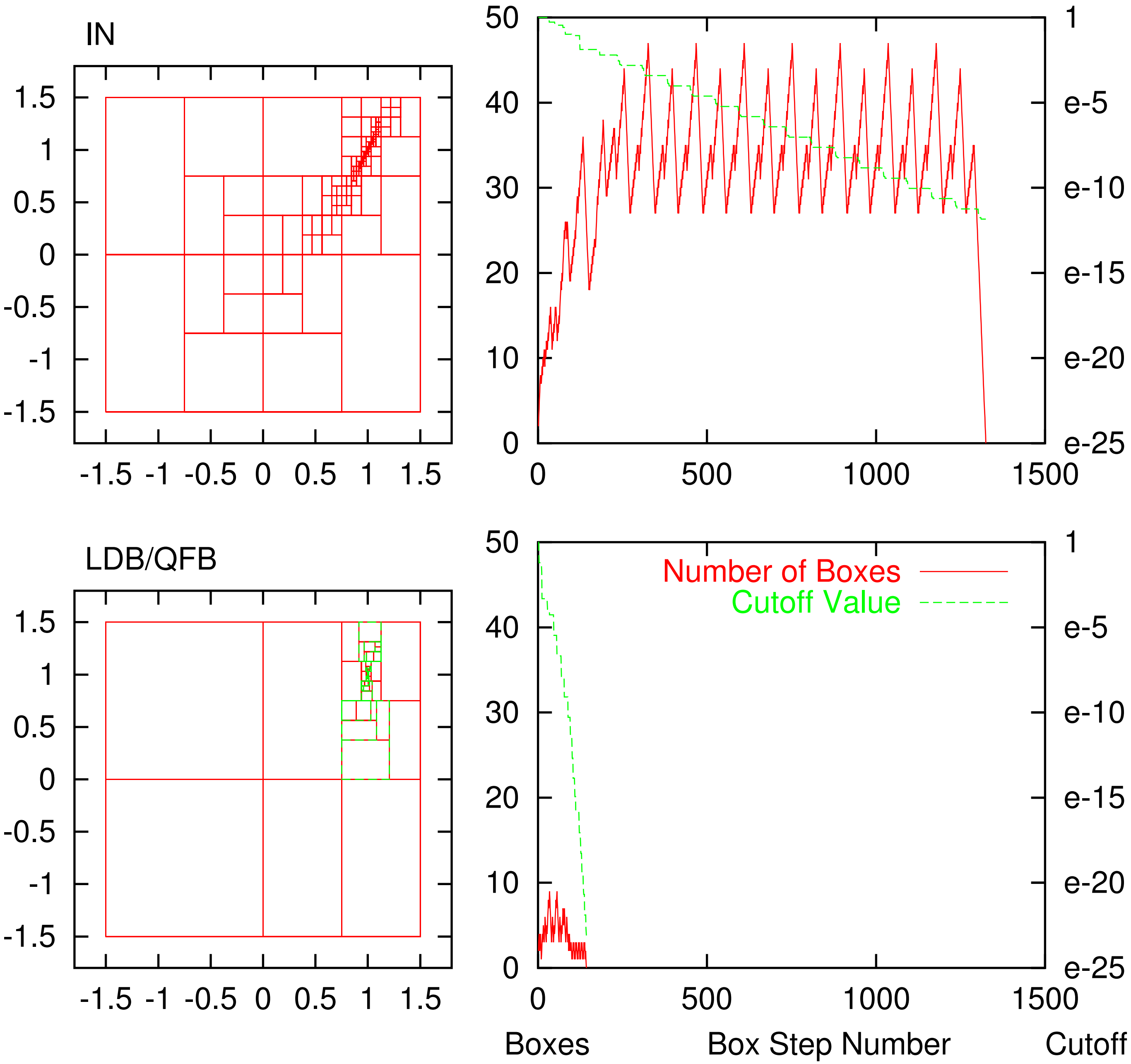
$$f(x,y) = \frac{7}{4}x^2 - \frac{\sqrt{3}}{2}xy + \frac{5}{4}y^2 = \frac{1}{2}(x,y)\cdot\begin{pmatrix} \frac{7}{2} & -\frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & \frac{5}{2} \end{pmatrix}\cdot\begin{pmatrix} x \\ y \end{pmatrix},$$

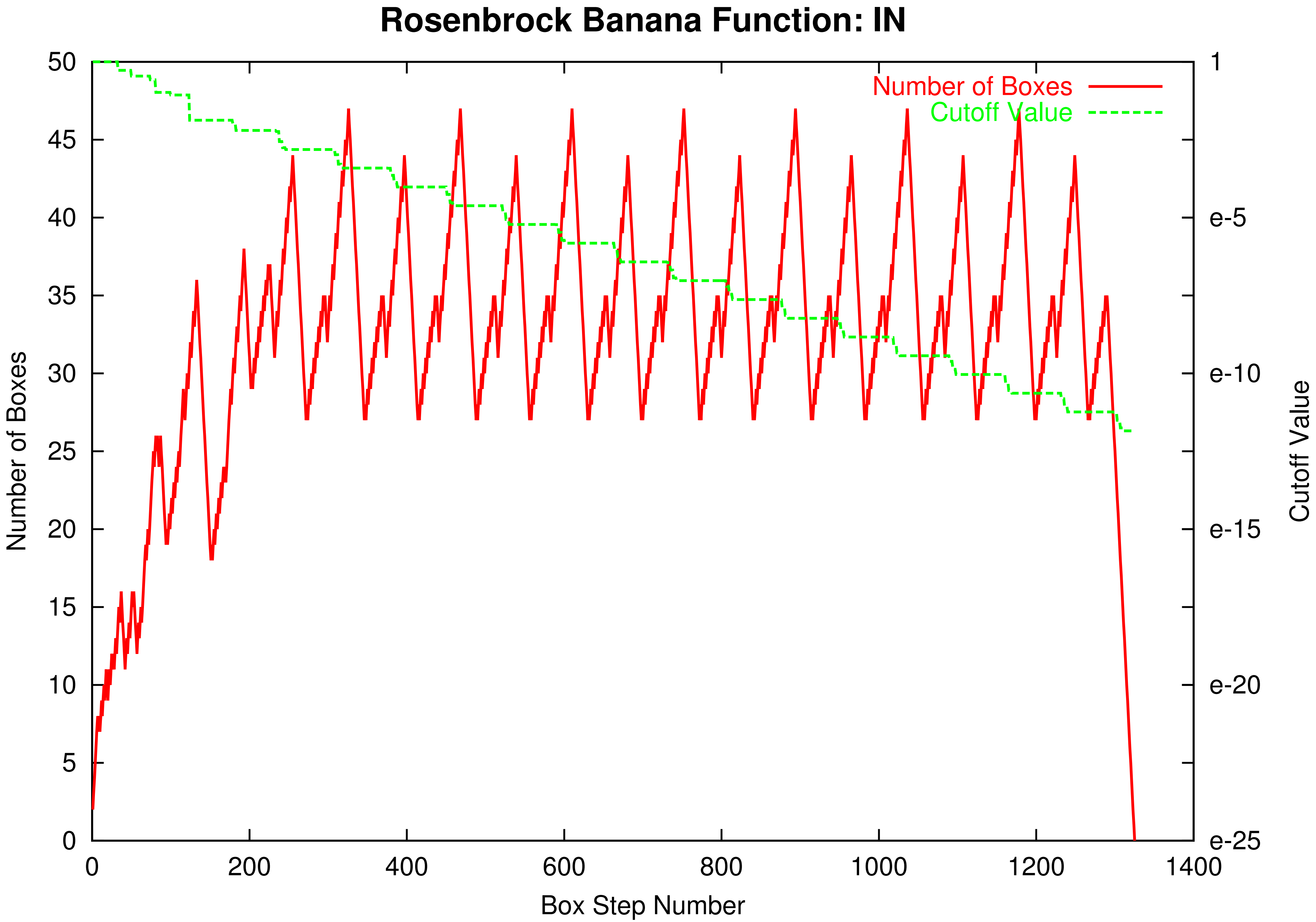
Again, we demand $a = 1$. Using the formula, at this time, we obtain x_m and y_m as

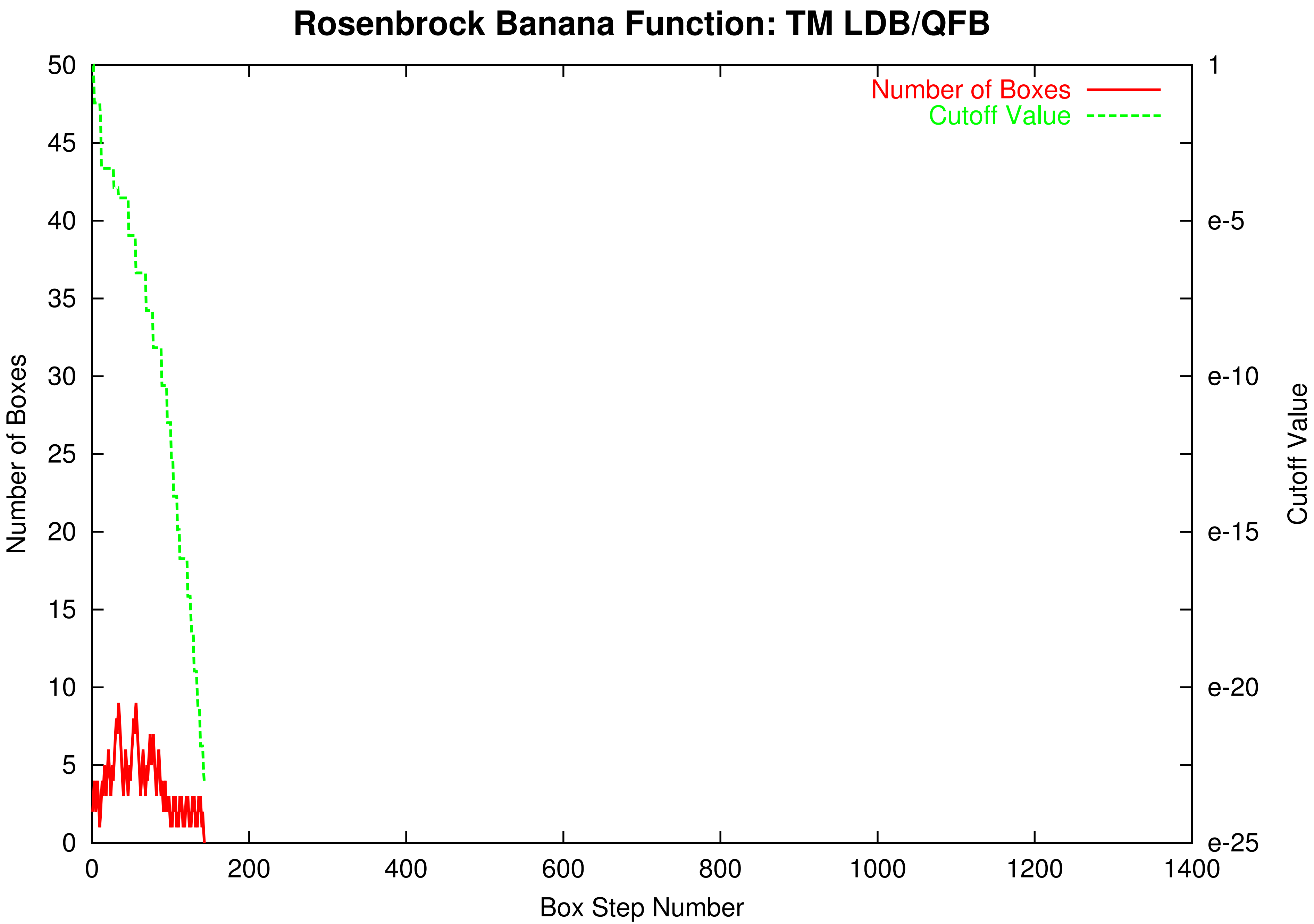
$$x_m = \sqrt{\frac{2\cdot 1}{\frac{7}{2} - \left(-\frac{\sqrt{3}}{2}\right)\cdot\frac{2}{5}\cdot\left(-\frac{\sqrt{3}}{2}\right)}} = \sqrt{\frac{5}{8}},$$

$$y_m = \sqrt{\frac{2\cdot 1}{\frac{5}{2} - \left(-\frac{\sqrt{3}}{2}\right)\cdot\frac{2}{7}\cdot\left(-\frac{\sqrt{3}}{2}\right)}} = \sqrt{\frac{7}{8}}.$$

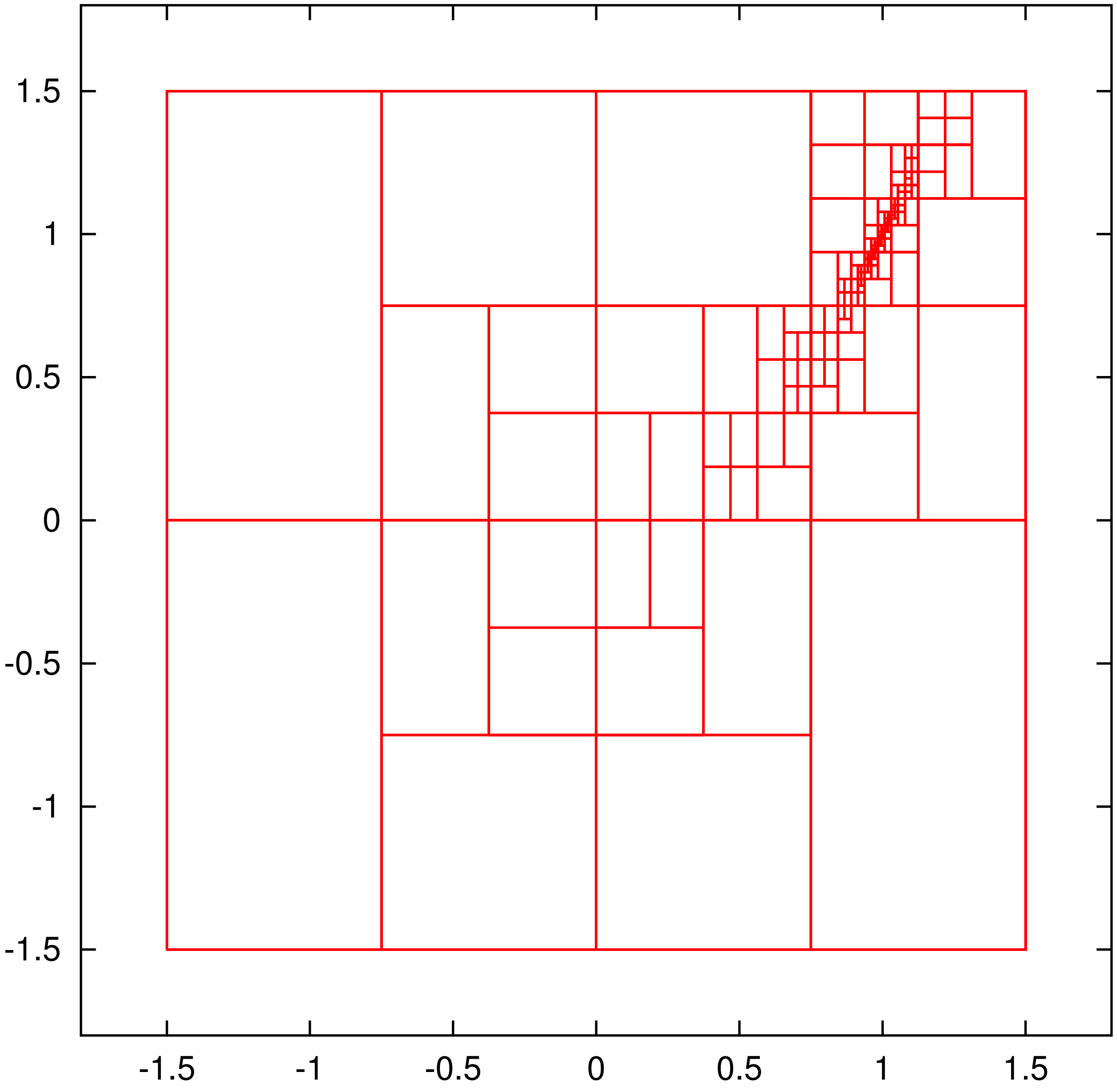
As expected, x_m is larger and y_m is smaller, and the area size of the interval box is larger.

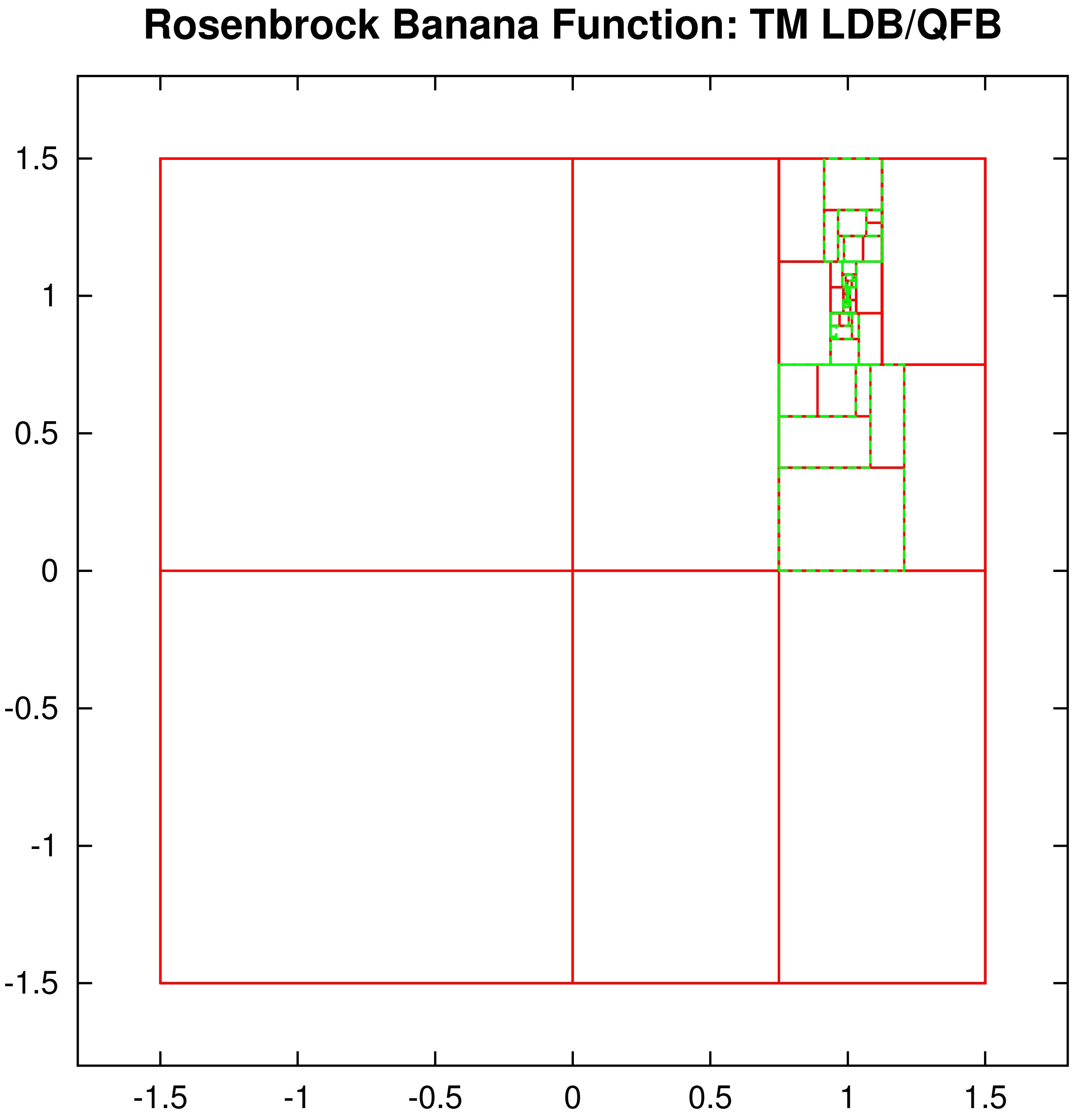

IN
LDB/QFB
-1.5
-1
-0.5
0
0.5
1
1.5
0
10
20
30
40
50
500
1000
1500
1
e-5
e-10
e-15
e-20
e-25
Number of Boxes
Cutoff Value
Boxes
Box Step Number
Cutoff


Rosenbrock Banana Function: IN
Number of Boxes
Cutoff Value
Number of Boxes
Box Step Number
Cutoff Value
0
5
10
15
20
25
30
35
40
45
50
0
200
400
600
800
1000
1200
1400
1
e-5
e-10
e-15
e-20
e-25


Rosenbrock Banana Function: TM LDB/QFB
Number of Boxes
Cutoff Value
Number of Boxes
Cutoff Value
Box Step Number
0
5
10
15
20
25
30
35
40
45
50
1
e-5
e-10
e-15
e-20
e-25
0
200
400
600
800
1000
1200
1400

Rosenbrock Banana Function: IN




Rosenbrock Banana Function: TM LDB/QFB
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

Some General Thoughts about Rigorous Parallel Optimization

1. Performance gains in modern computing are gained through multi-processor architectures, not increased clock speed and more efficient microcode.
2. While the global optimization task does not parallelize trivially, with due care it is manageable

Caveats:

1. Communication mode, in particular for large numbers of processors - point to point, master - slave, common meeting?
2. Load balancing, in particular with many processors and slow connections

Key Features and Algorithms of COSY-GO

- List management of boxes not yet determined to not contain the global minimizer. Loading a new box. Discarding a box with range above the current threshold value. Splitting a box with range not above the threshold value for further analysis. Storing a box smaller than the specified size.
- Application of a series of bounding schemes, starting from mere interval arithmetic to naive Taylor model bounding, LDB, then QFB. A higher bounding scheme is executed only if all the lower schemes fail.
- Update of the threshold cutoff value via various schemes. It includes upper bound estimates of the local minimum by corresponding bounding schemes, the mid point estimate, global estimates based on local behavior of function using gradient line search and convex quadratic form.
- Box size reduction using LDB QPB.
- Resulting data is available in various levels including graphics output.

COSY-GO in Parallel Environment

Design aspects of COSY-GO-P

1. Utilize MPI and be standard. This is done with a COSY language construct called **PLOOP**, a parallel loop with various types inter-processor updates upon conclusion. Can be **nested**.

2. Should scale from for **different numbers of processors**

   (a) multiple cores in a chip

   (b) large clusters with thousands of processors

3. Should scale for **different connection speeds**

   (a) extremely fast interconnect (multiple cores in one chip)

   (b) very fast (a few cores in a "node" with a nearly bus-like interconnect)

   (c) fast (specialized network for parallel use, at least Gigabit)

   (d) slow (grid-based systems - geographicly dispersed, relying on standard Internet)

Basic Ideas of the COSY-GO Parallel Environment

1. **List Management:** Each processor has two lists:
   (a) **Short List of large boxes**, shared with other processors
   (b) A section of Short List is pre-allocated to each processor.
   (c) **Long List of regular boxes** owned by each processor.
   (d) Long List is kept in moderately strict order of difficulty. Achieved by selection strategy favoring newer boxes

2. **Communication Concept**
   (a) Processors communicate in **sheduled meeting mode** after pre-determined fixed time interval T_m.
   (b) Time interval T_m is determined by trial and error for each environment under consideration. Single node: fraction of second, Berkeley NERSC cluster (~6000 processors): 1-2 minutes, Grid systems: fractions of hours.

What Happens in a Meeting

1. **Assess status**. Gather short data from each processor, scatter this information to all others. Cutoff updates, number of remaining large boxes and small boxes
2. **Processing of results.** Global cutoff is updated; it is determined if we can stop code
3. **Processing of status**. Each processor simultaneously identifies
   (a) how many boxes N_r are needed to replenish **Short List**
   (b) Let $N_p = N_r/N_{proc}$
4. **Load balancing.**
   (a) Each processor uploads its N_p largest boxes, if available, to the Short List
   (b) The Short List is randomized, so that the sections allocated to each processor are roughly of similar complexity

What Happens Between Meetings

1. Each processor splits its time between
   (a) working on its Long List of boxes. For each box, perform a sequence of tests: interval evaluation rejection test; Taylor model evaluation: LDB, QFB bounders, Gradient-based box rejection with Gradient Taylor models
   (b) performing non-rigorous global search (currently via genetic algorithm) in its assigned search space of global boxes, as well as neighboring global boxes
2. If Long List of boxes is exhausted, retrieve a box from the processor's section on the Short List
3. If processor's section on Short List is exhausted, continue to perform non-rigorous global search as in 1b.
4. After appropriate time, join next meeting.

Normal Form Methods

Iterative order-by-order coordinate transfomation to simplify dynamics around a fixed point.

Result: Except for resonances, up to order n,

- Elliptic case $\lambda_{i+1} = \bar{\lambda}_i$: spiral motion in $(\lambda_i, \lambda_{i+1})$ plane
- Elliptic unity case $\lambda_{i+1} = \bar{\lambda}_i$ and $|\bar{\lambda}_i| = 1$: circular motion, radius-dependent rotation frequency
- Hyperbolic case (λ_i real) motion along $\vec{e}_i$ axis, expanded or contracted by λ_i

Practial use:

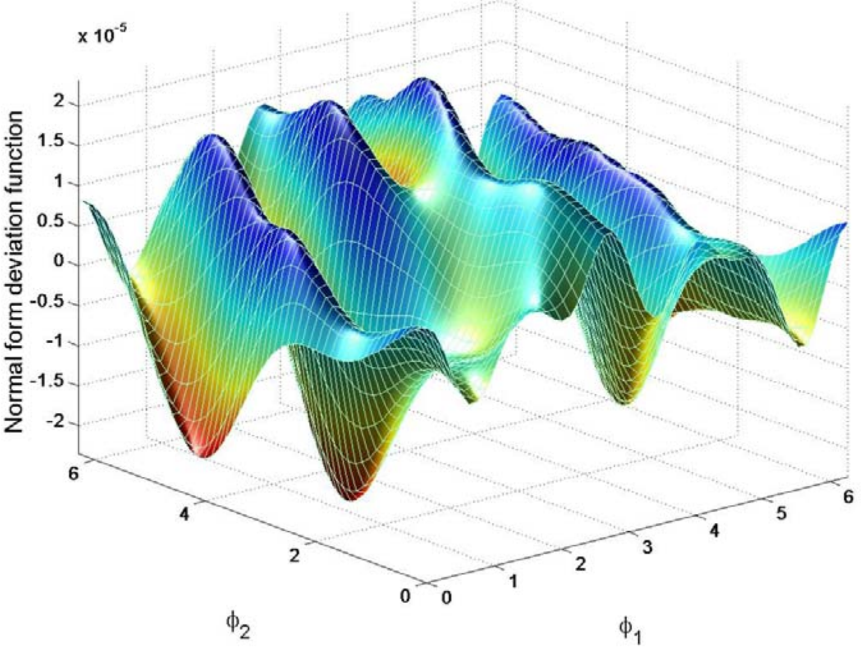
- Can be performed rigorously in Taylor model arithmetic
- Implemented to arbitrary order in arbitrarily many variables in COSY INFINITY

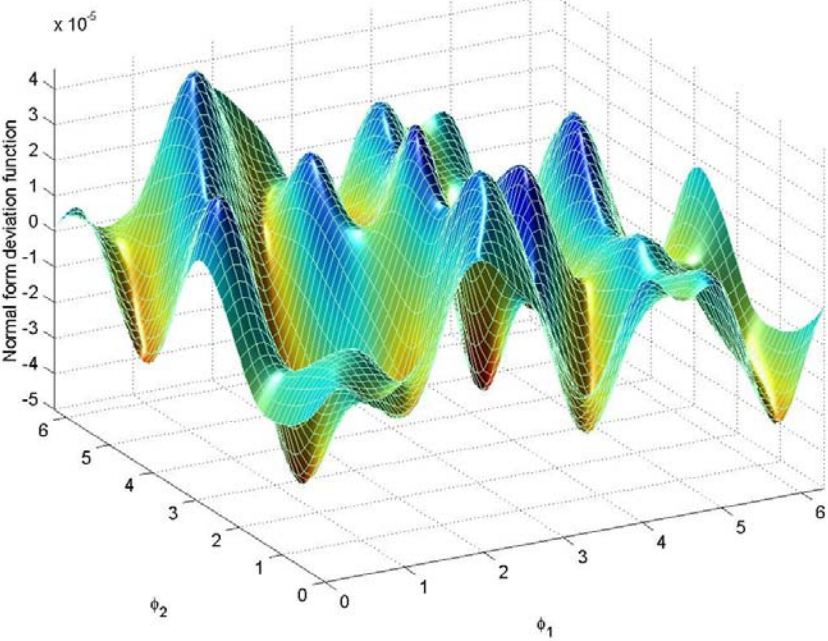
The Normal Form Defect Function

- **Extreme cancellation**; one of the reasons TM methods were invented
- Six-dimensional problem from dynamical systems theory
- Describes invariance defects of a particle accelerator
- Essentially composition of three tenth order polynomials
- The function vanishes identically to order ten
- Study for $a\cdot$ $(1,1,1,1,1,1)$ for $a = .1$ and $a = .2$
- Interesting **Speed observation**: on same machine,
  * one CF in INTLAB takes 45 minutes
  * one TM of order 7 takes 10 seconds

$$f_4(x_1, .., x_6) = \sum_{i=1}^{3} \left(\sqrt{y_{2i-1}^2 + y_{2i}^2} - \sqrt{x_{2i-1}^2 + x_{2i}^2} \right)^2$$

$$\text{where } \vec{y} = \vec{P}_1 \left(\vec{P}_2 \left(\vec{P}_3(\vec{x}) \right) \right)$$


x 10^-5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
Normal form deviation function
6
4
2
0
ϕ2
0
1
2
3
4
5
6
ϕ1


x 10^-5
Normal form deviation function
4
3
2
1
0
-1
-2
-3
-4
-5
6
5
4
3
2
1
0
φ2
0
1
2
3
4
5
6
φ1

GlobSol Results

For the computations, GlobSol's maximum list size was changed to 10^6, and the CPU limit was set to 10 days. All other parameters affecting the performance of GlobSol were left at their default values.

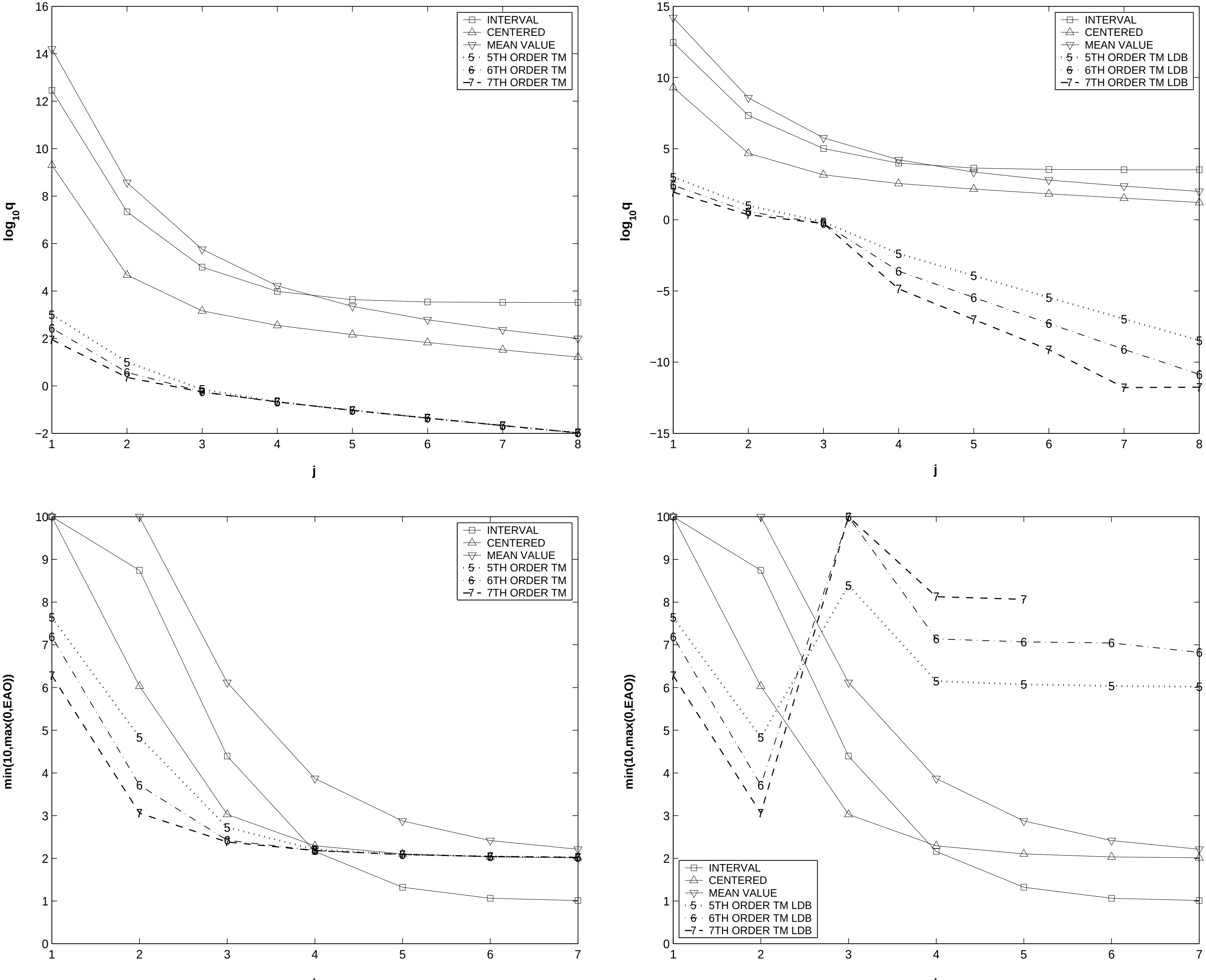
Dimension	CPU-time needed	Max list	Total # of Boxes
2	18810 sec		4733
3	>562896 sec (not finished yet)		
4	>259200 sec (could not finish)		63446 (remaining)
5	> 86400 sec (could not finish)		21306 (remaining)
6	not attempted		

We observe that in this example, COSY outperforms GlobSol by many orders of magnitude. However, we are not completely sure if a different choice of parameters for GlobSol could result in better performance.

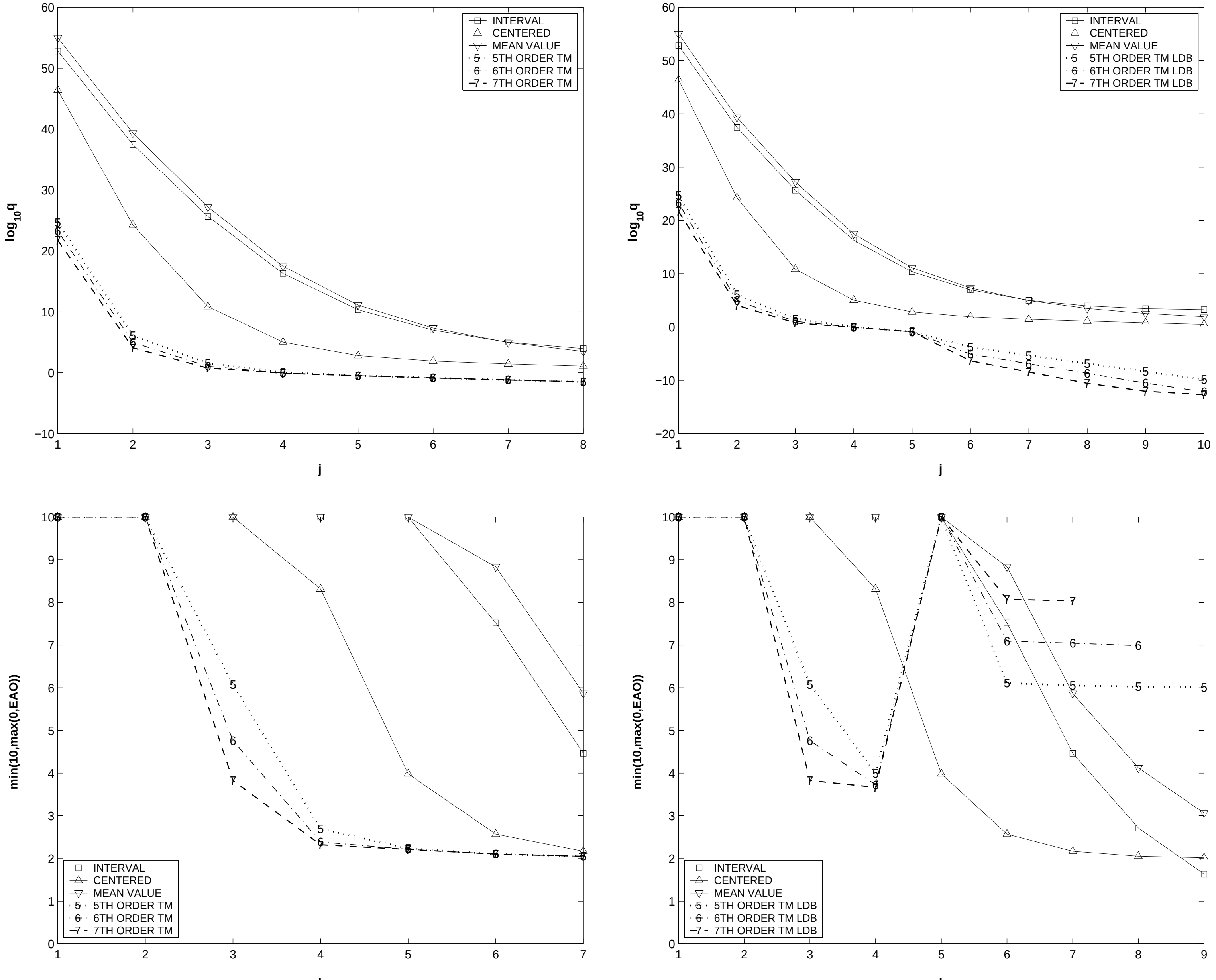
COSY-GO Results

Tolerance on the sharpness of the resulting minimum is 10^{-10}.For the evaluation of the objective function, Taylor models of order 5 were used. For the range bounding of the Taylor models, Makino's LDB with domain reduction was being used.

Dimension	CPU-time needed	Max list	Total # of Boxes
2	5.747071 sec	11	31
3	38.48828 sec	44	172
4	346.8604 sec	357	989
5	3970.746 sec	2248	6641
6	57841.94 sec	17241	49821



log10q
j
INTERVAL
CENTERED
MEAN VALUE
5TH ORDER TM
6TH ORDER TM
7TH ORDER TM
5TH ORDER TM LDB
6TH ORDER TM LDB
7TH ORDER TM LDB
min(10,max(0,EAO))
-2
0
2
4
6
8
10
12
14
16
-15
-10
-5
0
5
10
15
1
2
3
4
5
6
7
8
9
10



log10q
j
1
2
3
4
5
6
7
8
60
50
40
30
20
10
0
−10
INTERVAL
CENTERED
MEAN VALUE
5TH ORDER TM
6TH ORDER TM
7TH ORDER TM
log10q
j
1
2
3
4
5
6
7
8
9
10
60
50
40
30
20
10
0
−10
−20
INTERVAL
CENTERED
MEAN VALUE
5TH ORDER TM LDB
6TH ORDER TM LDB
7TH ORDER TM LDB
min(10,max(0,EAO))
1
2
3
4
5
6
7
10
9
8
7
6
5
4
3
2
1
0
INTERVAL
CENTERED
MEAN VALUE
5TH ORDER TM
6TH ORDER TM
7TH ORDER TM
min(10,max(0,EAO))
1
2
3
4
5
6
7
8
9
10
9
8
7
6
5
4
3
2
1
0
INTERVAL
CENTERED
MEAN VALUE
5TH ORDER TM LDB
6TH ORDER TM LDB
7TH ORDER TM LDB

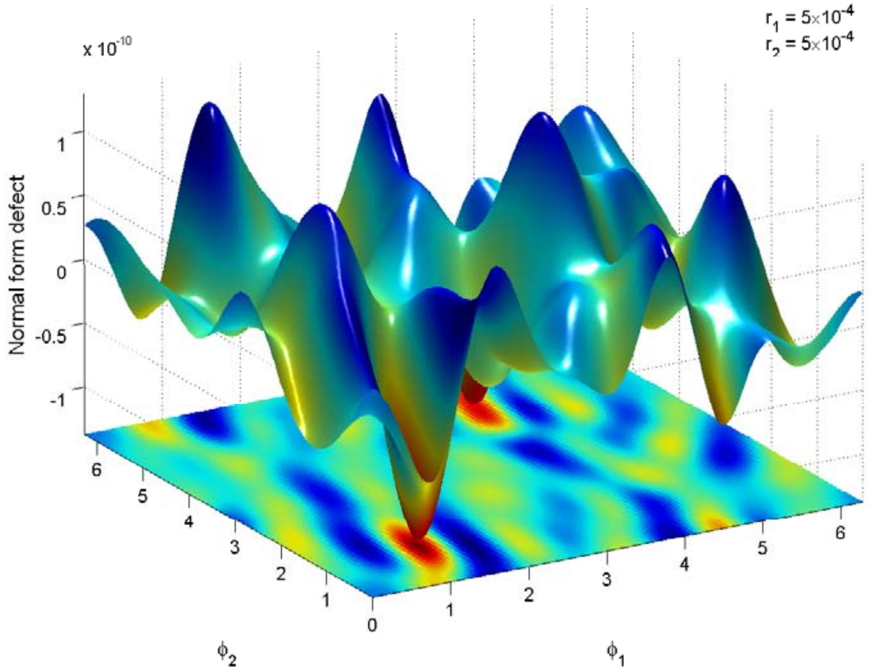


Fig. 9. Projection of the normal form defect function. Dependence on two angle variables for the fixed radii $r_1 = r_2 = 5 \cdot 10^{-4}$

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.