Taylor Methods and Computer Assisted Proofs - CAP08 -

Lecture Notes of Course at
Universitat de Barcelona
June 3-7, 2008
MSU Technical Report
MSUHEP-080609, June 2008

Martin Berz and Kyoko Makino

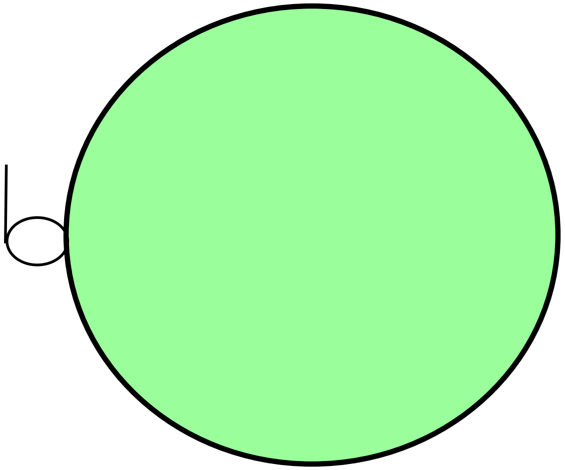
Schedule of CAP08 - Foundations

- Foundations of Rigorous Computing: Interval methods, floating point requirements and standards, dependency problem, Taylor methods and related approaches, rigorous higher order bounds, paths to rigorous arbitrary precision and practical realization.
- Rigorous Integration of ODEs and Flows. Interval-based integration, error estimation, wrapping effect, differential algebraic structures, Taylor integration of flows, rigorous error bounds, automatic step size control
- Constraint Satisfaction Taylor models for inverses, point solutions, constraint manifolds, differential algebraic equations (DAEs), rigorous high-order control theory
- Divide and Conquer Methods. Taylor manifolds and automatic domain decomposition, applications for rigorous global optimization, constraint satisfaction, and flow integration.

Schedule of CAP08 - Computer Assisted Proofs

- Enclosures of attractors of discrete and continuous systems in two and higher dimensions,
- High order normal forms,
- Enclosures of hyperbolic manifolds in various dimensions
- Enclosure of homoclinic points and determination of symbolic dynamics,
- Sharp estimates of topological entropy, center manifolds and nonlinear
- Lyapunov and pseudo-Lyapunov functions, rigorous Nekhoroshev-type long-term stability estimates.

Motion in the Tevatron



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

⇒ 4x10^4 revs/sec.

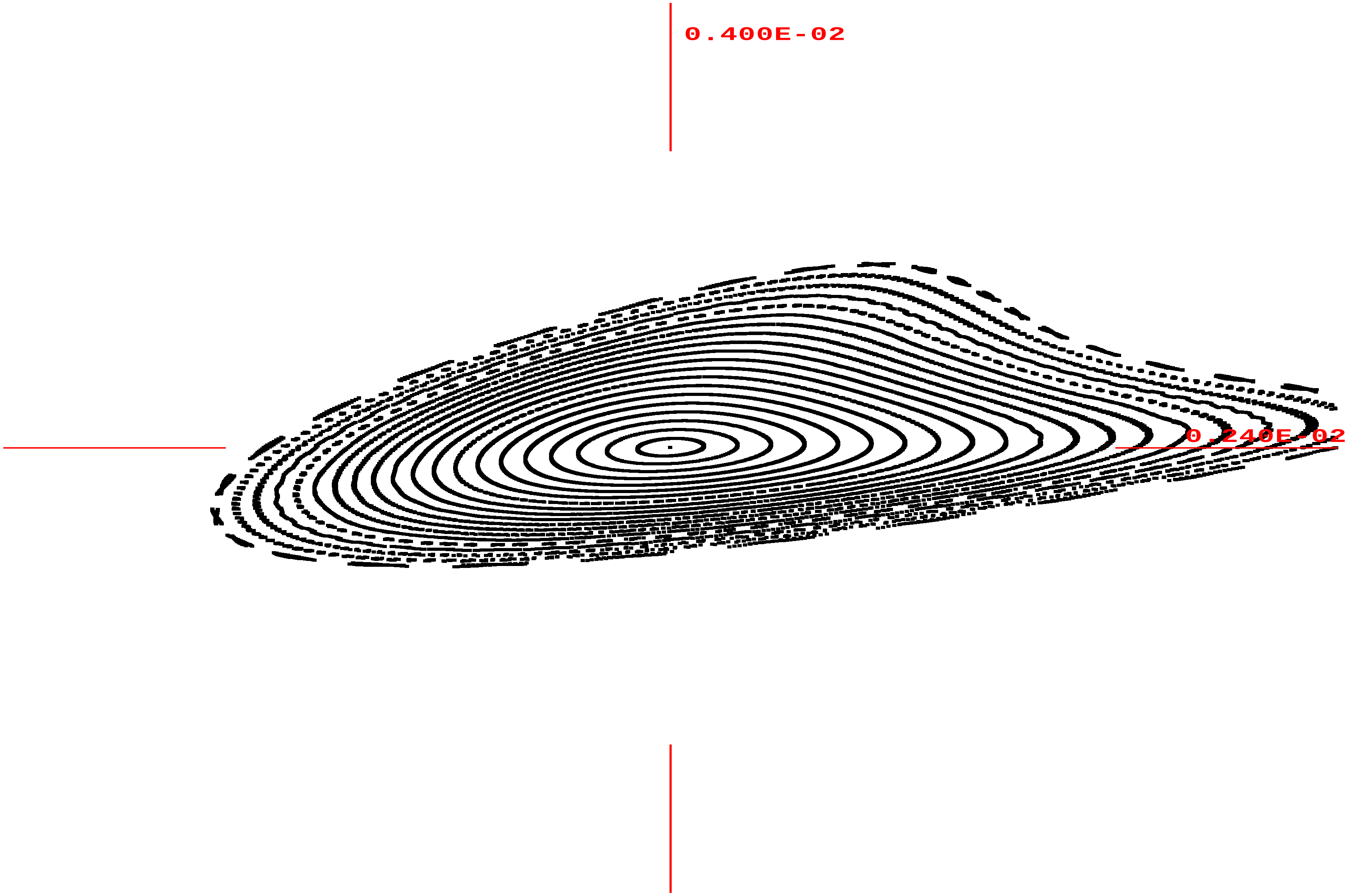
- Need to store about 10 hours, or 4x10^5 sec

⇒ 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


0.400E-02
0.240E-02

The Particle Optical Equations of Motion

$$
\begin{aligned}
x' &= a \cdot (1 + hx) \cdot \frac{p_0}{p_z} \\
y' &= b \cdot (1 + hx) \cdot \frac{p_0}{p_z} \\
l' &= (1 + \delta_m) \cdot (1 + hx) \cdot \frac{1 + \eta}{1 + \eta_0} \cdot \frac{p_0}{p_z} \\
a' &= \left((1 + \delta_m) \cdot \frac{1 + \eta}{1 + \eta_0} \cdot \frac{p_0}{p_z} \cdot \frac{E_x}{\chi_{E0}} \;-\; \frac{B_y}{\chi_{M0}} \;+\; b \cdot \frac{p_0}{p_z} \cdot \frac{B_z}{\chi_{M0}} \right) \\
&\quad \cdot (1 + hx) \cdot (1 + \delta_z) + h \cdot \frac{p_z}{p_0} \\
b' &= \left((1 + \delta_m) \cdot \frac{1 + \eta}{1 + \eta_0} \cdot \frac{p_0}{p_z} \cdot \frac{E_y}{\chi_{E0}} \;+\; \frac{B_x}{\chi_{M0}} \;-\; a \cdot \frac{p_0}{p_z} \cdot \frac{B_z}{\chi_{M0}} \right) \\
&\quad \cdot (1 + hx) \cdot (1 + \delta_z)
\end{aligned}
$$

Here the following abbreviations are used:

$$
\begin{aligned}
\chi_{E0} &= \frac{p_0 \cdot v_0}{z_0 e} \;,\;\; \chi_{M0} = \frac{p_0}{z_0 e} \\
\eta &= \left(\frac{K_0 \cdot (1+\delta_k) - z_0 \cdot e \cdot (1+\delta_z) \cdot V(x,y,s)}{m_0 c^2 \cdot (1+\delta_m)} \right) \\
\frac{p_z}{p_0} &= \sqrt{(1+\delta_m)^2 \cdot \frac{\eta(2+\eta)}{\eta_0(2+\eta_0)} - a^2 - b^2}
\end{aligned}
$$

Equations are expressed in **curvilinear coordinates,** an orthogonal system attached to a **reference orbit.** From earliest times, these have proven to be advantageous in practice for numerical stability.

Description and Analysis of Optical Systems

Easiest way to study optical system: trace rays through system. Light Optics: Snell's Law. Particle Optics: numerical integration through electromagnetic fields.

Very easy to do. **BUT:** does not provide much insight. Hard to see what to change if a system has problems.

Better to look at the **MAP** of the system:

$$\begin{pmatrix} x_i \\ a_i = p_{xi}/p_o \\ y_i \\ b_i = p_{yi}/p_o \\ \Delta E/E_0 \\ \Delta t/t_0 \end{pmatrix} \longrightarrow \begin{pmatrix} x_f \\ a_f = p_{xf}/p_o \\ y_f \\ b_f = p_{yf}/p_o \\ \Delta E/E_0 \\ \Delta t/t_0 \end{pmatrix}$$

In terms of conventional ODE integration, this is merely the local flow.

Matrix Formulation of Gaussian Optics

Gaussian Optics is equivalent to considering only **linear** part of Map. Can be represented by matrix:

$$\begin{pmatrix} x_f \\ a_f \end{pmatrix} = \begin{pmatrix} (x,x) & (x,a) \\ (a,x) & (a,a) \end{pmatrix} \begin{pmatrix} x_i \\ a_i \end{pmatrix}$$

- Elements of Matrices tell us about the system:
  $(x,a) = 0$: Image. (x,x): Magnification
- Matrices for systems can be obtained by multiplication of Matrices of sub-systems (convenient!)

$$\begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -1/f & 1 \end{pmatrix} \begin{pmatrix} 1 & g \\ 0 & g \end{pmatrix} \begin{pmatrix} 1 - b/f & (g - bg/f + b) \\ -1/f & 1 - g/f \end{pmatrix}$$

Image: $1/b + 1/g - 1/f = 0$. One solution is $b = g = 2f$, in this case magnification is -1.

Nonlinear Terms

Principle can be extended to higher derivatives of map. These describe ”aberrations”.
Example: centered system with cylindrical symmetry
(Seidel Aberrations)

(x,aaa)	Spherical Aberration
(x,xaa)	Coma
(x,xxa)	Curvature of Image
(x,xxx)	Distortion (Barrel or Pincushion)

In addition, there are chromatic effects:

(x,xd)
(x,ad)

The Limitations

1. Higher orders are needed
   - Very high resolution (electron microscopes, spectrographs)
   - Large phase spaces (muon accelerators)
   - Long systems where errors can build up (circular accelerators)
2. Special elements
   - Fringe fields of particle optical elements
   - Off-axis effects
   - Electric and magnetic round lenses
   - General field elements based on measured data

As before, in all these cases numerical integration can provide individual rays, but not maps.

History of Higher Order Optics

	Light Optics	**Electron Optics**	**Particle Optics**
	(Round Lenses)	(Round Lenses)	(Non-Round Lenses)
1	Gauss 1841		?
2	(Gauss 1841)		Brown 1959
3	Petzval 1840 Seidel 1856	Scherzer 1936	Matsuda, Wollnik 1965
4			
5	Kohlschütter, Schwarzschild 1905 Rabinovich 1946		M.B. 1985

Maps as Taylor Series

> ”The determination of terms of order higher than fourth is very laborious in all but the simplest cases. For this reason, algebraic calculations are usually restricted to the domain of the Seidel theory, supplemented where necessary by ray tracing”.

Born-Wolf, *Principles of Optics, Pergamon 1989*

Some Power Series Particle Optics Codes:

- TRANSPORT (2nd order, thick elements, early 60s)
- GIOS (3rd order, thick elements, fringe fields, late 60s)
- MaryLie (3rd order, thick elements, fringe fields, late 70s)
- COSY 5.0 (5th order, thick elements, fringe fields, 1985)
- COSY INFINITY (arbitrary order, thick elements, fringe fields, 1987) (More than 1000 users by 2004)

```
      SUBROUTINE elmm(L,Z,K01,K02,K03,K04,K05,K27,K32,NORDER,NG,ND)
*     *************************************************************
*
*     Subroutine to Compute Aberration Equations Equations
*     Magnetic Multipole to Fifth Order
*     Computer Generated by Program HAMILTON (C) M. Berz 1985
*
      IMPLICIT DOUBLE PRECISION (A - Z)
*
      INTEGER NORDER, NG, ND
*
      DOUBLE PRECISION L(0:461,7)
*
      K30    = 1./(1+K32)
      K31    = 1./(1+K32/2.)
*
      FX2 = -K01*K27
*
      FY2 = +K01*K27
*
      IF(FX2.LT.-1.D-8) THEN
         AFX = SQRT(-FX2)
         CX  = COS(AFX*Z)
         SX  = SIN(AFX*Z)/AFX
      ELSEIF(FX2.GT.1.D-8) THEN
         AFX = SQRT(FX2)
         EX  = EXP(AFX*Z)
         EEX = 1.D0/EX
         CX  = (EX + EEX)/2.D0
         SX  = (EX - EEX)/2.D0/AFX
      ELSE
         CX  = 1.D0
         SX  = Z
         FX2 = 1.D-8
      ENDIF
*
      IF(FY2.LT.-1.D-8) THEN
         AFY = SQRT(-FY2)
         CY  = COS(AFY*Z)
         SY  = SIN(AFY*Z)/AFY
      ELSEIF(FY2.GT.1.D-8) THEN
         AFY = SQRT(FY2)
         EY  = EXP(AFY*Z)
         EEY = 1.D0/EY
         CY  = (EY + EEY)/2.D0
         SY  = (EY - EEY)/2.D0/AFY
      ELSE
         CY  = 1.D0
         SY  = Z
         FY2 = 1.D-8
      ENDIF
*
*
      CS2 = CX
      CS3 = SX
      CS4 = CY
      CS5 = SY
      CS6 = Z
      KK2 = K31*K32
      KK3 = K30*K32
      FF2 = FX2
      TT2 = CS2
      TT3 = CS3
      TT4 = CS3*FF2
      TT5 = CS4
      TT6 = CS5
      TT7 = CS5*FF2
      TT8 = CS6
      TT9 = CS6*KK2
      TT10 = CS6*KK3
      L(1,1) = (+TT2)
      L(2,1) = (+TT3)
      L(1,2) = (+TT4)
      L(2,2) = (+TT2)
      L(3,3) = (+TT5)
      L(4,3) = (+TT6)
      L(3,4) = (-TT7)
      L(4,4) = (+TT5)
*
      IF(ND.EQ.0.AND.NG.EQ.0) GOTO 100
*
      L(6,6) = (+1)
      L(6,7) = (-0.5D+00*TT8-0.25D+00*TT9+TT10)
*
      IF(NG.EQ.0) GOTO 100
*
      L(5,5) = (+1)
      L(5,7) = (+0.5D+00*TT8+0.25D+00*TT9-TT10)
*
 100  IF(NORDER.EQ.1) GOTO 1000
*
      CS7 = CS3*CX
```

```
CS8 = CS3*SX
CS9 = CS4*CX
CS10 = CS4*SX
CS11 = CS5*CX
CS12 = CS5*SX
CS13 = CS5*CY
CS14 = CS5*SY
CS15 = CS6*CX
CS16 = CS6*SX
CS17 = CS6*CY
CS18 = CS6*SY
KK4 = KK2*K31*K32
KK5 = KK3*K31*K32
KK6 = K02*K27
FF3 = 1/FX2/FX2
FF4 = FF3*FX2
TT11 = KK6*FF4
TT12 = CS2*KK6*FF4
TT13 = CS8*KK6
TT14 = CS3*KK6*FF4
TT15 = CS7*KK6*FF4
TT16 = KK6*FF3
TT17 = CS2*KK6*FF3
TT18 = CS8*KK6*FF4
TT19 = CS14*KK6
TT20 = CS13*KK6*FF4
TT21 = CS14*KK6*FF4
TT22 = CS16*FF2
TT23 = CS16*KK2*FF2
TT24 = CS3*KK2
TT25 = CS15
TT26 = CS15*KK2
TT27 = CS3*KK6
TT28 = CS7*KK6
TT29 = CS13*KK6
TT30 = CS3*KK2*FF2
TT31 = CS15*FF2
TT32 = CS15*KK2*FF2
TT33 = CS4*KK6*FF4
TT34 = CS9*KK6*FF4
TT35 = CS12*KK6
TT36 = CS10*KK6*FF4
TT37 = CS5*KK6*FF4
TT38 = CS11*KK6*FF4
TT39 = CS4*KK6*FF3
TT40 = CS9*KK6*FF3
TT41 = CS12*KK6*FF4
```

```
      TT42 = CS18*FF2
      TT43 = CS18*KK2*FF2
      TT44 = CS5*KK2
      TT45 = CS17
      TT46 = CS17*KK2
      TT47 = CS10*KK6
      TT48 = CS5*KK6
      TT49 = CS11*KK6
      TT50 = CS5*KK2*FF2
      TT51 = CS17*FF2
      TT52 = CS17*KK2*FF2
      TT53 = CS7*FF2
      TT54 = CS6*FF2
      TT55 = CS8*FF2
      TT56 = CS7
      TT57 = CS13*FF2
      TT58 = CS14*FF2
      TT59 = CS13
      TT60 = CS6*KK4
      TT61 = CS6*KK5
      L(7,1) = (+0.33333334327D+00*(+TT11-TT12-TT13))
      L(8,1) = (+0.66666668654D+00*(+TT14-TT15))
      L(13,1) = (+0.66666668654D+00*(-TT16+TT17)-0.33333333333D+00*TT
     *  18)
      L(18,1) = (+0.60000002384D+00*(-TT11+TT12)+0.2D+00*TT19)
      L(19,1) = (+0.40000000596D+00*(+TT14-TT20))
      L(22,1) = (+0.40000000596D+00*(-TT16+TT17)-0.2D+00*TT21)
      L(7,2) = (-0.33333333333D+00*TT27-0.66666666667D+00*TT28)
      L(8,2) = (+0.66666668654D+00*(-TT11+TT12)-0.13333333333D+01*TT13)
      L(13,2) = (+0.66666668654D+00*(+TT14-TT15))
      L(18,2) = (+0.6D+00*TT27+0.4D+00*TT29)
      L(19,2) = (+0.40000000596D+00*(-TT11+TT12)+0.8D+00*TT19)
      L(22,2) = (+0.40000000596D+00*(+TT14-TT20))
      L(9,3) = (+0.40000000596D+00*(-TT33+TT34)+0.8D+00*TT35)
      L(14,3) = (+0.4D+00*TT36-0.12D+01*TT37+0.8D+00*TT38)
      L(10,3) = (-0.8D+00*TT36+0.40000000596D+00*(+TT37+TT38))
      L(15,3) = (+0.80000001192D+00*(+TT39-TT40)+0.4D+00*TT41)
      L(9,4) = (+0.12D+01*TT47+0.40000000596D+00*(+TT48+TT49))
      L(14,4) = (+0.12000000477D+01*(-TT33+TT34)+0.4D+00*TT35)
      L(10,4) = (+0.40000000596D+00*(+TT33-TT34)+0.12D+01*TT35)
      L(15,4) = (-0.4D+00*TT36-0.8D+00*TT37+0.12D+01*TT38)
      L(7,7) = (+0.25D+00*(+TT53-TT54))
      L(8,7) = (+0.5D+00*TT55)
      L(13,7) = (+0.25D+00*(+TT56+TT8))
      L(18,7) = (+0.25D+00*(-TT57+TT54))
      L(19,7) = (-0.5D+00*TT58)
      L(22,7) = (+0.25D+00*(+TT59+TT8))
```

27,000 lines further down:

```
      L(449,7) = (-0.87890625D-01*TT59-0.244140625D-01*TT347
     *  +0.2197265625D-01*TT1723+0.14282226563D+00*TT6924+0.390625D-01*(
     *  +TT348-TT1724)-0.263671875D+00*TT6925-0.380859375D+00*TT8
     *  -0.1162109375D+00*TT9+0.9521484375D-01*TT60+0.26733398438D+00*TT
     *  351+0.1484375D+00*(+TT10-TT61)-0.439453125D+00*TT352
     *  +0.29296875D+00*TT349+0.91796875D-01*TT350-0.732421875D-01*TT
     *  1725-0.12451171875D+00*TT6926+0.109375D+00*(-TT1726+TT1727)
     *  +0.17578125D+00*TT6927+0.140625D+00*TT1728+0.546875D-01*TT1729
     *  -0.3515625D-01*TT1730-0.29296875D-01*TT6928+0.3125D-01*(-TT
     *  6929+TT6930)+0.234375D-01*TT6931+0.78125D-02*(+TT6892-TT6933)
     *  +0.390625D-02*(+TT6893+TT6934)+0.1953125D-02*(-TT6894+TT6935)
     *  -0.9765625D-03*TT6895-0.15625D-01*TT6932)
      L(458,7) = (+0.234375D-01*TT8-0.390625D-02*TT9-0.8203125D-01*TT
     *  60+0.380859375D+00*TT351-0.32470703125D+00*TT1731
     *  +0.76904296875D-01*TT6936+0.15625D-01*TT10+0.21875D+00*TT61
     *  -0.9140625D+00*TT352+0.7421875D+00*TT1732-0.1708984375D+00*TT
     *  6937)
      L(459,7) = (+0.390625D-01*TT8+0.390625D-02*TT9-0.1171875D-01*TT
     *  60-0.68359375D-01*TT351+0.18798828125D+00*TT1731
     *  -0.76904296875D-01*TT6936-0.15625D-01*TT10+0.3125D-01*TT61
     *  +0.1640625D+00*TT352-0.4296875D+00*TT1732+0.1708984375D+00*TT
     *  6937)
      L(460,7) = (+0.13671875D+00*TT8+0.29296875D-01*TT9+0.5859375D-02
     *  *TT60-0.48828125D-02*TT351-0.25634765625D-01*TT1731
     *  +0.38452148438D-01*TT6936-0.1171875D+00*TT10-0.15625D-01*TT61
     *  +0.1171875D-01*TT352+0.5859375D-01*TT1732-0.8544921875D-01*TT
     *  6937)
*
  500 IF(NORDER.EQ.5) GOTO 1000
*
 1000 CONTINUE
*
      RETURN
      END
```

History of Higher Order Optics

	Light Optics	**Electron Optics**	**Particle Optics**
	(Round Lenses)	(Round Lenses)	(Non-Round Lenses)
1	Gauss 1841		?
2	(Gauss 1841)		Brown 1959
3	Petzval 1840 Seidel 1856	Scherzer 1936	Matsuda, Wollnik 1965
4			
5	Kohlschütter, Schwarzschild 1905 Rabinovich 1946		M.B. 1985

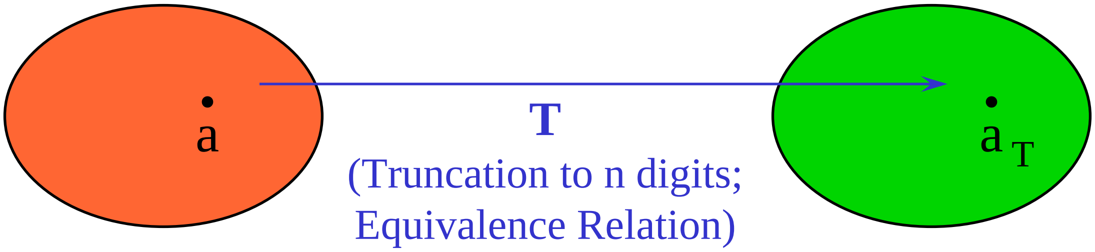
Maps as Taylor Series

> ”The determination of terms of order higher than fourth is very laborious in all but the simplest cases. For this reason, algebraic calculations are usually restricted to the domain of the Seidel theory, supplemented where necessary by ray tracing”.

Born-Wolf, *Principles of Optics, Pergamon 1989*
Some Power Series Particle Optics Codes:

- TRANSPORT (2nd order, thick elements, early 60s)
- GIOS (3rd order, thick elements, fringe fields, late 60s)
- MaryLie (3rd order, thick elements, fringe fields, late 70s)
- COSY 5.0 (5th order, thick elements, fringe fields, 1985)
- COSY INFINITY (arbitrary order, thick elements, fringe fields, 1987) (More than 1000 users by 2004)

Number Fields and Floating Point Numbers



a

a_T

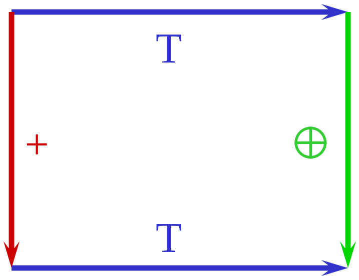
T

(Truncation to n digits;
Equivalence Relation)

Real Numbers

Floating Point
"Numbers"

$c = a + b$



T

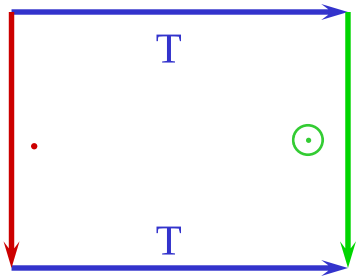
$+$

$\oplus$

T

$c_T = a_T \oplus b_T$

$c = a \cdot b$



T

$\cdot$

$\odot$

T

$c_T = a_T \odot b_T$

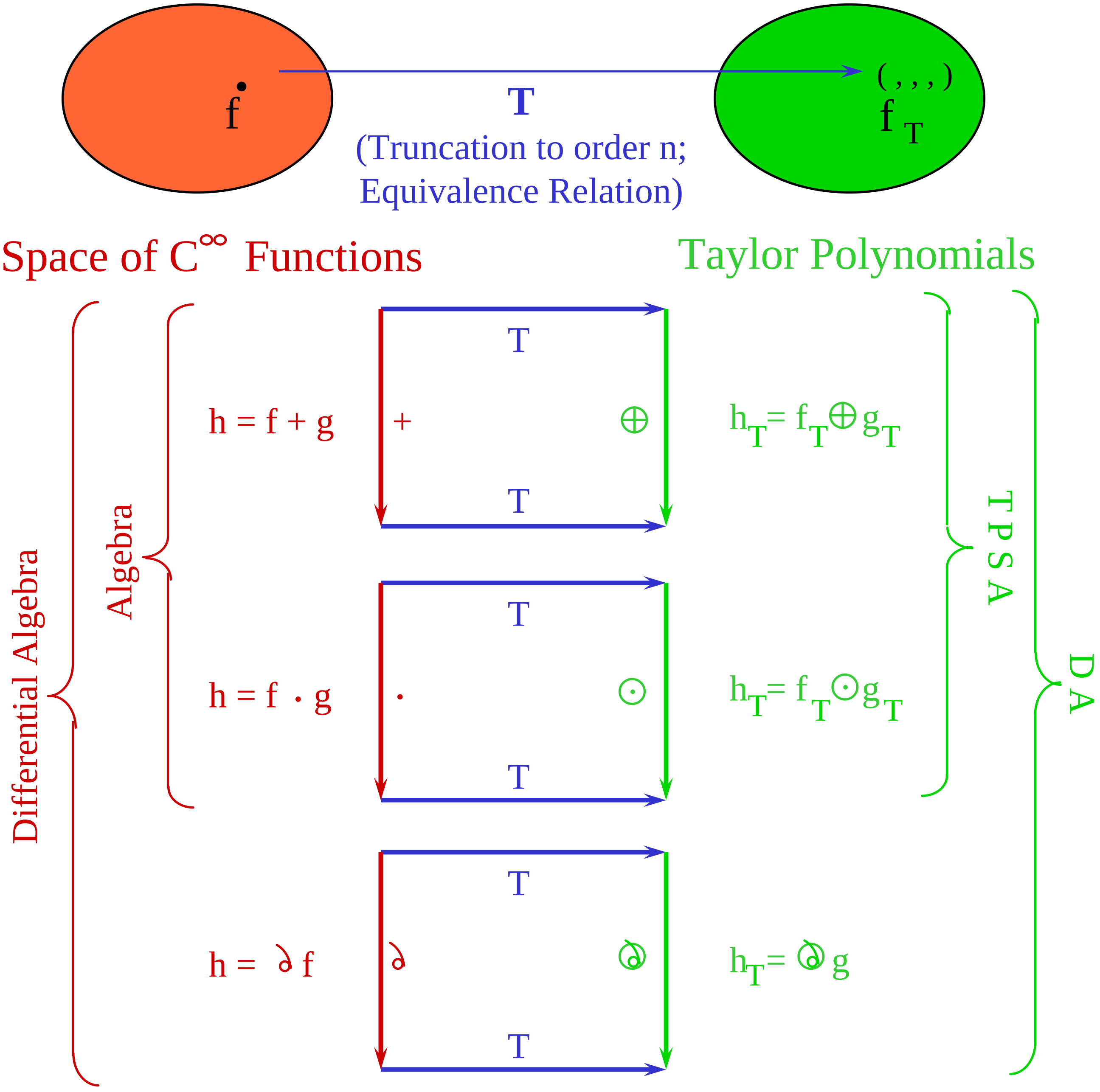
Field
(Also want "exp", "sin"
etc: Banach Field)

Diagrams
commute
"approximately"

Field
("approximately")

T: Extracts Information
considered relevant

Function Algebras



f

$(\,,\,,\,,\,)$ f_T

T

(Truncation to order n;
Equivalence Relation)

Space of C^∞ Functions

Taylor Polynomials

Differential Algebra

Algebra

$h = f + g$ $\quad +$ $\quad T$ $\quad \oplus$ $\quad h_T = f_T \oplus g_T$

$h = f \cdot g$ $\quad \cdot$ $\quad T$ $\quad \odot$ $\quad h_T = f_T \odot g_T$

$h = \partial f$ $\quad \partial$ $\quad T$ $\quad \textcircled{\partial}$ $\quad h_T = \textcircled{\partial}\, g$

TPSA

DA

Differential Algebra
(also want “exp”, “sin”
etc: Banach DA)

Diagrams
commute
exactly

Differential Algebra
(even Banach DA)

T: Extracts Information
considered relevant

```
 DX       1, NO =  11, NV =   6, INA = 27
 *****************************************

    I  COEFFICIENT           ORDER   EXPONENTS
    1  -.11235361224856E-27   0     0 0  0 0  0 0
    2  -2.4577764619843       1     1 0  0 0  0 0
    3   382.75567438967       1     0 1  0 0  0 0
    4  -2.9058143083356       1     0 0  0 0  1 0
    5  -.18527560925181E-01   2     2 0  0 0  0 0
    6  -2.0449053795051       2     1 1  0 0  0 0
    7   230.43212330444       2     0 2  0 0  0 0
    8  0.10745263897241E-01   2     0 0  2 0  0 0
    9  -.40104008916025       2     0 0  1 1  0 0
   10  0.10748538944562E-01   2     1 0  0 0  1 0
   11  -2.3077398243793       2     0 1  0 0  1 0
   12   4.4690872366630       2     0 0  0 2  0 0
   13  0.61271339425922E-02   2     0 0  0 0  2 0
   14  0.42422562884505E-03   3     3 0  0 0  0 0
   15  -.25018158222129       3     2 1  0 0  0 0
   16   34.164428767158       3     1 2  0 0  0 0
   17  -2131.5807609885       3     0 3  0 0  0 0
   18  -.42666140335568E-03   3     1 0  2 0  0 0
   19  0.61867753199362       3     0 1  2 0  0 0
   20  -.11349802973152       3     1 0  1 1  0 0
   21   56.549952804501       3     0 1  1 1  0 0
   22  0.15244016913989E-02   3     2 0  0 0  1 0
   23  -.46319864237315       3     1 1  0 0  1 0
   24   40.581998425553       3     0 2  0 0  1 0
   25  -.25450130254498E-02   3     0 0  2 0  1 0
   26  -4.6823358538539       3     1 0  0 2  0 0
   27   1456.5235333461       3     0 1  0 2  0 0
   28  -.25643847152791       3     0 0  1 1  1 0
   29  0.15479910051681E-02   3     1 0  0 0  2 0
   30  -.26430052248601       3     0 1  0 0  2 0
   31  -7.1898343490889       3     0 0  0 2  1 0
   32  0.58340280013060E-03   3     0 0  0 0  3 0
   33  -.54394288125262E-04   4     4 0  0 0  0 0
   34  0.38915493503025E-01   4     3 1  0 0  0 0
   35  -10.856955574871       4     2 2  0 0  0 0
   36   1254.1599009949       4     1 3  0 0  0 0
   37  -54750.339715548       4     0 4  0 0  0 0
   38  0.42865921904244E-03   4     2 0  2 0  0 0
   39  -.10129872906961       4     1 1  2 0  0 0
   40   16.600177401064       4     0 2  2 0  0 0
   41  -.83438533353867E-03   4     0 0  4 0  0 0
   42  0.44656225936067E-01   4     2 0  1 1  0 0
```

```
43  -14.242733931933       4     1 1  1 1  0 0
44   1812.0210497771       4     0 2  1 1  0 0
45  -.14401693756858       4     0 0  3 1  0 0
46  -.26812285838556E-03   4     3 0  0 0  1 0
47  0.14433601146177       4     2 1  0 0  1 0
48  -25.002121384519       4     1 2  0 0  1 0
49   1422.4392191357       4     0 3  0 0  1 0
50  0.67315613953893E-03   4     1 0  2 0  1 0
51  -.16340868932716       4     0 1  2 0  1 0
52   1.3257841503873       4     2 0  0 2  0 0
53  -472.13503057908       4     1 1  0 2  0 0
54   51681.544792611       4     0 2  0 2  0 0
55  -9.7160870866478       4     0 0  2 2  0 0
56  0.88334103087622E-01   4     1 0  1 1  1 0
57  -19.022409240455       4     0 1  1 1  1 0
58  -.48689687612391E-03   4     2 0  0 0  2 0
59  0.16755301265834       4     1 1  0 0  2 0
60  -14.106243085175       4     0 2  0 0  2 0
61  0.43660754939067E-03   4     0 0  2 0  2 0
62  -300.75757764658       4     0 0  1 3  0 0
63   2.9012366747428       4     1 0  0 2  1 0
64  -572.73159967434       4     0 1  0 2  1 0
65  0.53258960203668E-01   4     0 0  1 1  2 0
66  -.37768280924720E-03   4     1 0  0 0  3 0
67  0.62989087030710E-01   4     0 1  0 0  3 0
68  -3573.0798614067       4     0 0  0 4  0 0
69   1.6716112876871       4     0 0  0 2  2 0
70  -.10658118377282E-03   4     0 0  0 0  4 0
71  -.25414137178589E-05   5     5 0  0 0  0 0
72  0.14017294168757E-02   5     4 1  0 0  0 0
73  -.22915066598219       5     3 2  0 0  0 0
74   9.3046379608457       5     2 3  0 0  0 0
75   711.33599214748       5     1 4  0 0  0 0
76  -36864.079818068       5     0 5  0 0  0 0
77  0.76846052404647E-05   5     3 0  2 0  0 0
78  -.24398280565609E-02   5     2 1  2 0  0 0
79  0.49036157764730E-01   5     1 2  2 0  0 0
80   10.611563244730       5     0 3  2 0  0 0
81  -.69006293687322E-05   5     1 0  4 0  0 0
82  -.58227683391493E-03   5     0 1  4 0  0 0
83  0.62878426765274E-03   5     3 0  1 1  0 0
84  -.93993449228250E-01   5     2 1  1 1  0 0
```

1100 lines further down ...

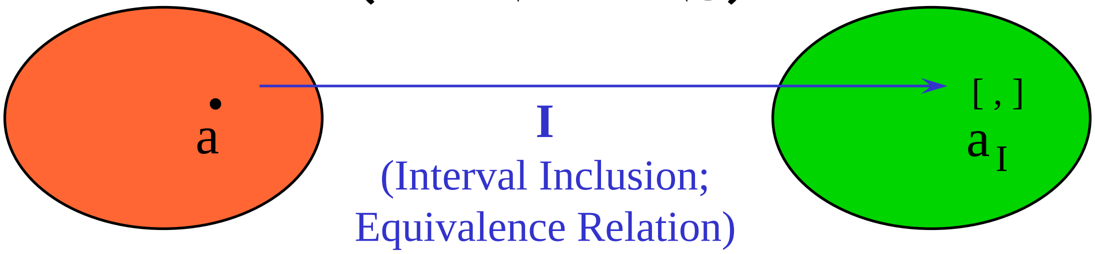
```
 2198  -14060.942901990      11     1 1  0 6  3 0
 2199   1653837.7385130      11     0 2  0 6  3 0
 2200  -257.59556830741      11     0 0  2 6  3 0
 2201  -.35544757510218      11     1 0  1 5  4 0
 2202  -79.173081098476      11     0 1  1 5  4 0
 2203  0.51087261295152E-01  11     2 0  0 4  5 0
 2204  -18.716238122641      11     1 1  0 4  5 0
 2205   1542.8309211899      11     0 2  0 4  5 0
 2206  -.28054439599319E-01  11     0 0  2 4  5 0
 2207  0.22645955750085E-02  11     1 0  1 3  6 0
 2208  -.38909056207237      11     0 1  1 3  6 0
 2209  -.24095385693697E-04  11     2 0  0 2  7 0
 2210  0.88753236469477E-02  11     1 1  0 2  7 0
 2211  -.78021595715990      11     0 2  0 2  7 0
 2212  0.12444200845638E-04  11     0 0  2 2  7 0
 2213  -.25672923569253E-06  11     1 0  1 1  8 0
 2214  0.46190659792613E-04  11     0 1  1 1  8 0
 2215  0.85194029561067E-09  11     2 0  0 0  9 0
 2216  -.30147170356411E-06  11     1 1  0 0  9 0
 2217  0.25626592777412E-04  11     0 2  0 0  9 0
 2218  -.27294471690391E-09  11     0 0  2 0  9 0
 2219  -116911999.30448      11     1 0  010  0 0
 2220   20702000792.760      11     0 1  010  0 0
 2221  -19548517.253335      11     0 0  1 9  1 0
 2222  -48961.786425583      11     1 0  0 8  2 0
 2223   8224946.4473064      11     0 1  0 8  2 0
 2224  -3422.0920883930      11     0 0  1 7  3 0
 2225   17.260507499664      11     1 0  0 6  4 0
 2226  -4159.1942930668      11     0 1  0 6  4 0
 2227  -.53695898362788E-01  11     0 0  1 5  5 0
 2228  0.21124533035838E-01  11     1 0  0 4  6 0
 2229  -3.4758111999671      11     0 1  0 4  6 0
 2230  0.37040986757764E-03  11     0 0  1 3  7 0
 2231  -.72548521359208E-05  11     1 0  0 2  8 0
 2232  0.12676479473598E-02  11     0 1  0 2  8 0
 2233  -.33276602342639E-07  11     0 0  1 1  9 0
 2234  0.20264295213287E-09  11     1 0  0 0 10 0
 2235  -.34321706228840E-07  11     0 1  0 0 10 0
 2236  -97662091.940232      11     0 0  010  1 0
 2237  -17547.398707127      11     0 0  0 8  3 0
 2238   4.1188522337990      11     0 0  0 6  5 0
 2239  0.33673939625236E-02  11     0 0  0 4  7 0
 2240  -.91990677263444E-06  11     0 0  0 2  9 0
 2241  0.20981553537075E-10  11     0 0  0 0 11 0
```

Validated Maps as Taylor Series

In validated methods, use of first order is absolutely essential because of **wrapping effect.**

- Zeroth Order: Moore, Kaucher-Miranker, many others
- First Order: Moore early 1960s, Eijgenraam
- First Order, "good" asymptotics: Lohner 1986 "AWA"
- Arbitrary Order by Taylor Models: Makino and Berz 1997 "COSY-VI". Shrink Wrapping, Blunting, Curvlinear Preconditioning
- Theoretical Foundation of "good" aysmptotics of AWA by Nedialkov and Jackson: 1999(?)

Number Field Inclusions (Intervals)



I
(Interval Inclusion;
Equivalence Relation)

Real Numbers

Floating Point Intervals

$c = a + b$

$$\begin{tikzcd} {} \arrow[r, "I"'] \arrow[d, "+"'] & {} \arrow[d, "\oplus"] \\ {} \arrow[r, "I"] & {} \end{tikzcd}$$

$c_I = a_I \oplus b_I$

$c = a \cdot b$

$$\begin{tikzcd} {} \arrow[r, "I"'] \arrow[d, "\cdot"'] & {} \arrow[d, "\odot"] \\ {} \arrow[r, "I"] & {} \end{tikzcd}$$

$c_I = a_I \odot b_I$

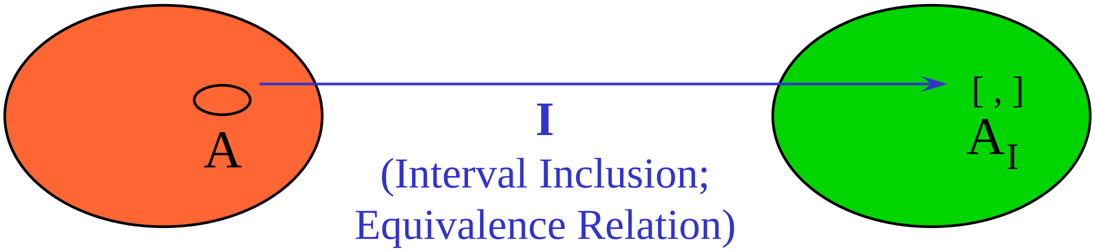
Field
(Also want “exp”, “sin”
etc: Banach Field)

Diagrams
commute
exactly!

Little algebraic
structure

I: Extracts Information
considered relevant

Set Inclusions (Intervals)



A

[,]

A_I

I

(Interval Inclusion; Equivalence Relation)

Real Number Set

Floating Point Intervals

$C = A + B$

$$\begin{tikzcd} {} \arrow[r, "I"] \arrow[d, "+"'] & {} \arrow[d, "\oplus"] \\ {} \arrow[r, "I"] & {} \end{tikzcd}$$

$C_I = A_I \oplus B_I$

$C = A \cdot B$

$$\begin{tikzcd} {} \arrow[r, "I"] \arrow[d, "\cdot"'] & {} \arrow[d, "\odot"] \\ {} \arrow[r, "I"] & {} \end{tikzcd}$$

$C_I = A_I \odot B_I$

Little algebraic structure

Diagrams commute exactly!

Little algebraic structure

I: Extracts Information considered relevant

Introduction

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

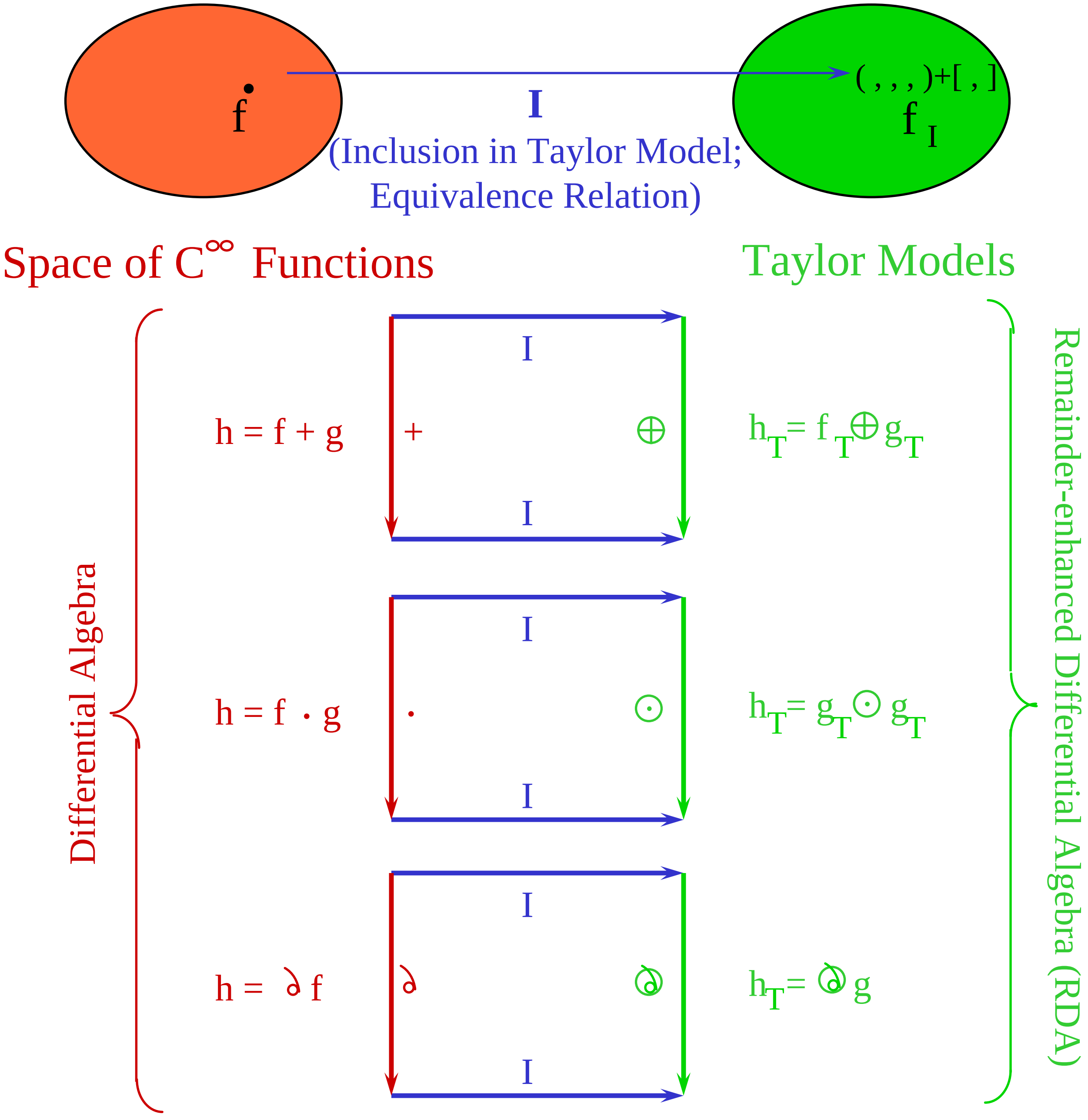
- Functions consist of code lists of 10^4 to 10^5 terms
- Have about the worst imaginable cancellation problem
- Are obtained via validated integration of large initial condition boxes.

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.

Function Algebra Inclusions



Differential Algebra
(also want “exp”, “sin”
etc: Banach DA)

Diagrams
commute
exactly

Differential Algebra
with Remainder

I: Extracts Information
considered relevant

Validated Maps as Taylor Series

In validated methods, use of first order is absolutely essential because of **wrapping effect.**

- Zeroth Order: Moore, Kaucher-Miranker, many others
- First Order: Moore early 1960s, Eijgenraam
- First Order, "good" asymptotics: Lohner 1986 "AWA"
- Arbitrary Order by Taylor Models: Makino and Berz 1997 "COSY-VI". Shrink Wrapping, Blunting, Curvlinear Preconditioning
- Theoretical Foundation of "good" aysmptotics of AWA by Nedialkov and Jackson: 1999(?)

The Henon Map

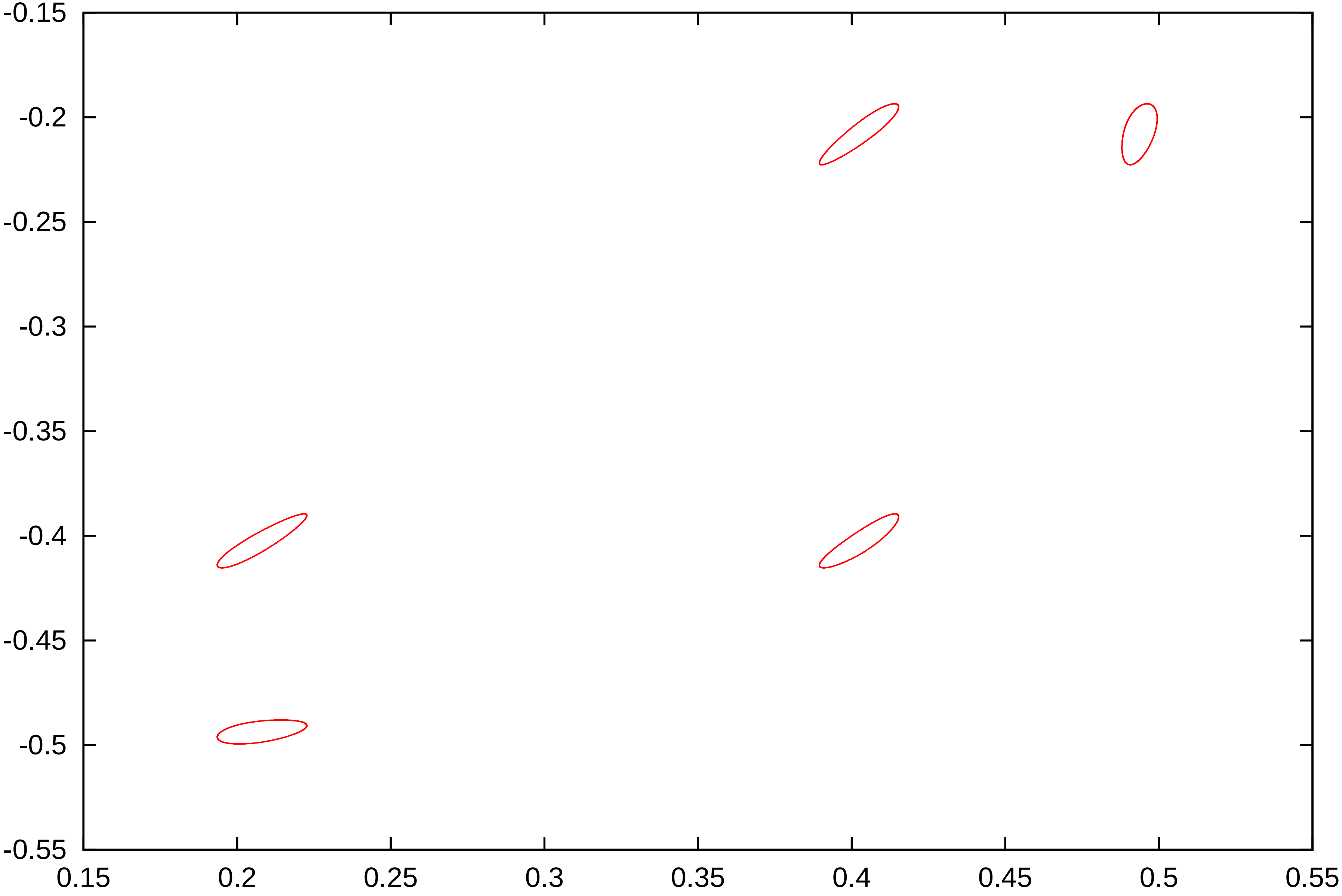
Henon Map: frequently used elementary example that exhibits many of the well-known effects of nonlinear dynamics, including chaos, periodic fixed points, islands and symplectic motion. The dynamics is two-dimensional, and given by

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

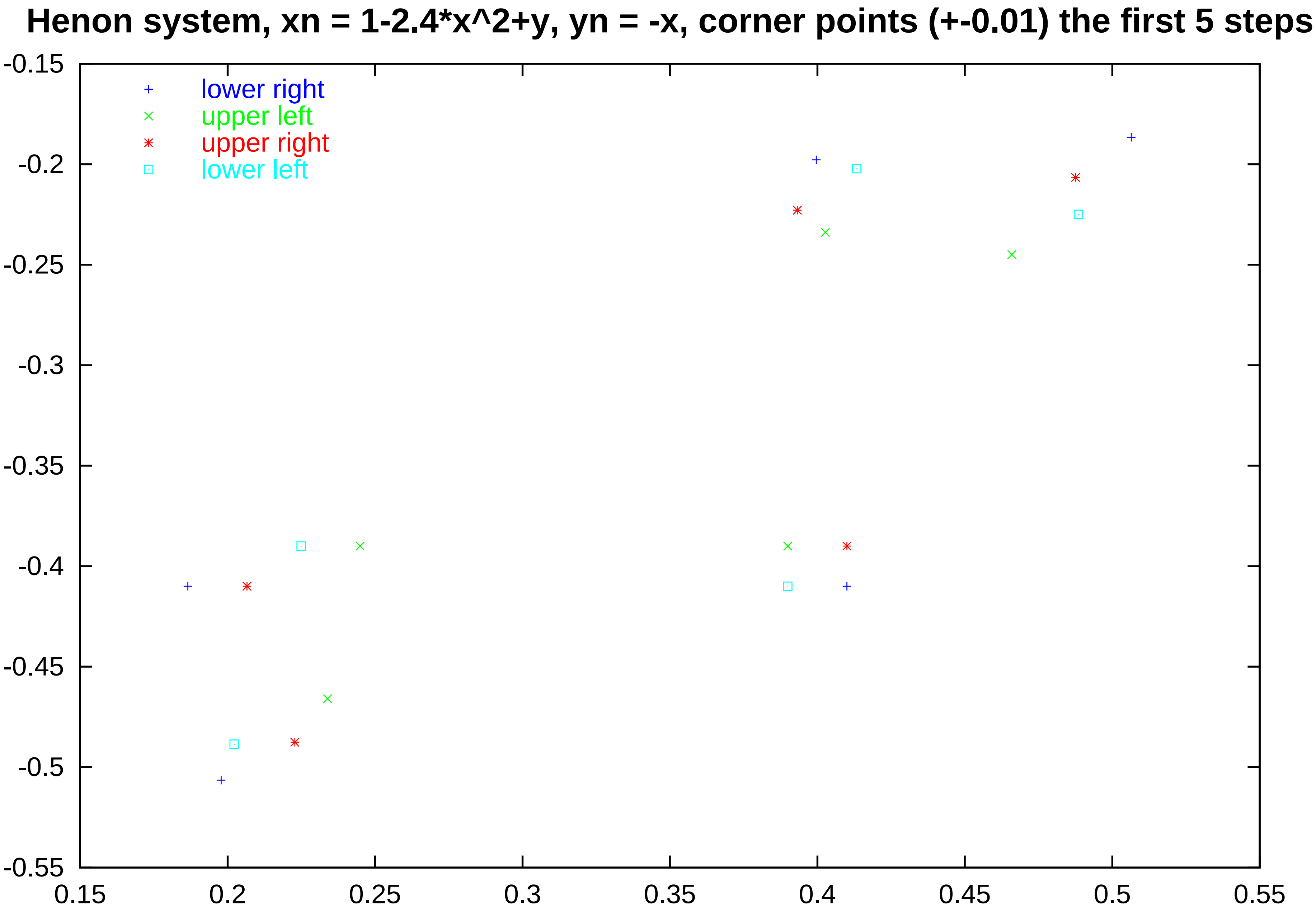
It can easily be seen that the motion is area preserving for $|\beta| = 1$.We consider

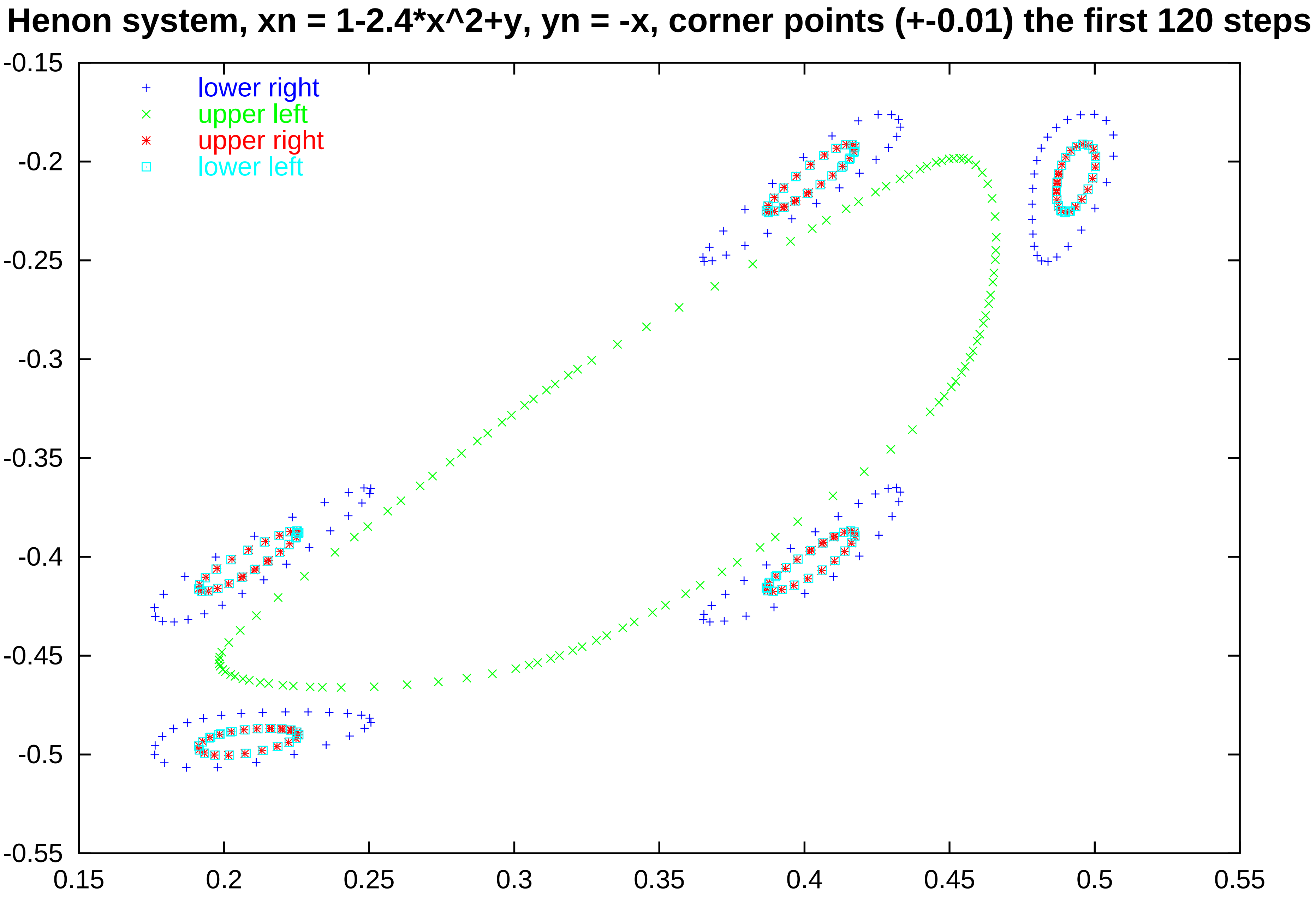
$$\alpha = 2.4 \text{ and } \beta = -1,$$

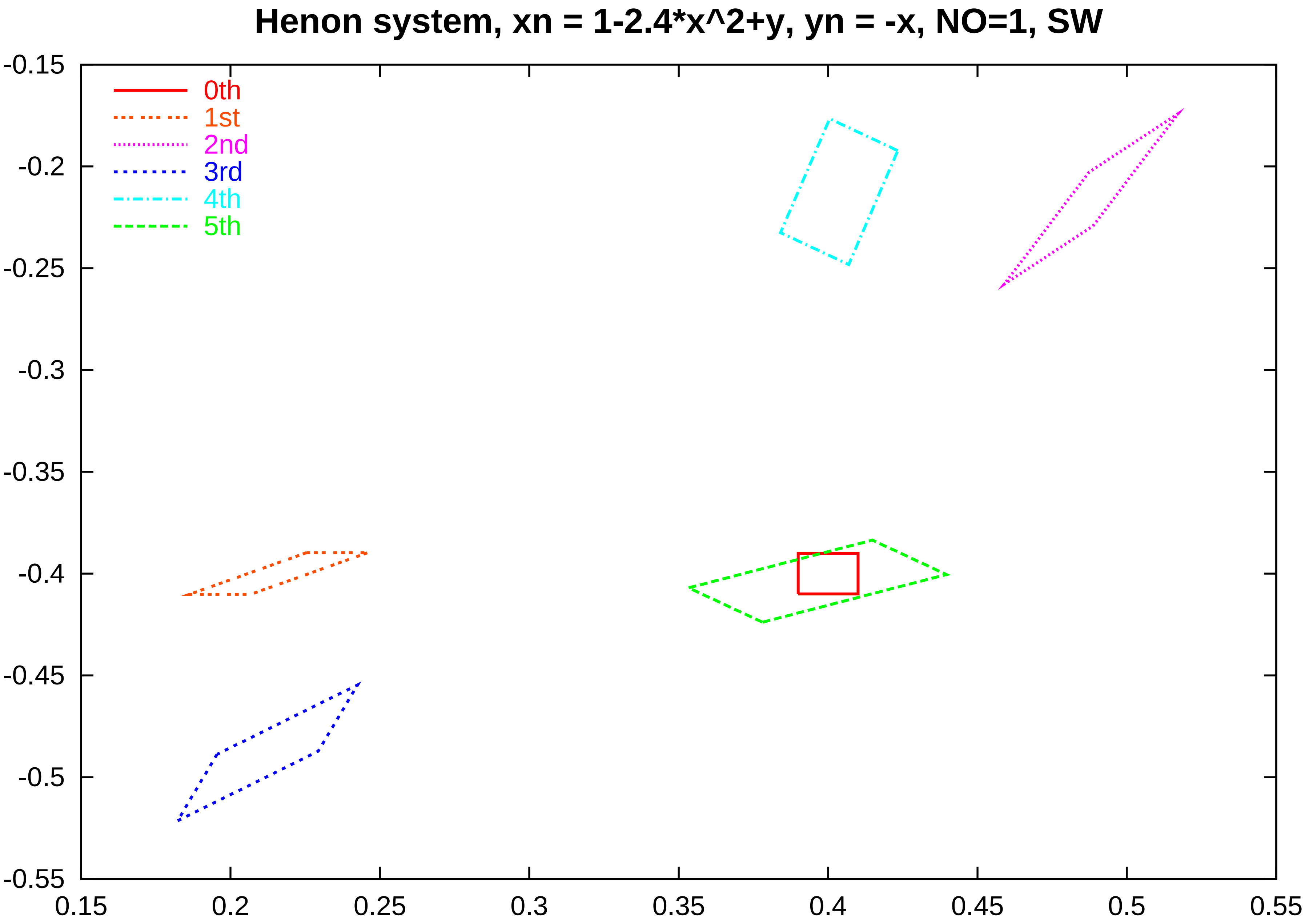
and concentrate on initial boxes of the from $(x_0, y_0) \in (0.4, -0.4) + [-d, d]^2$.

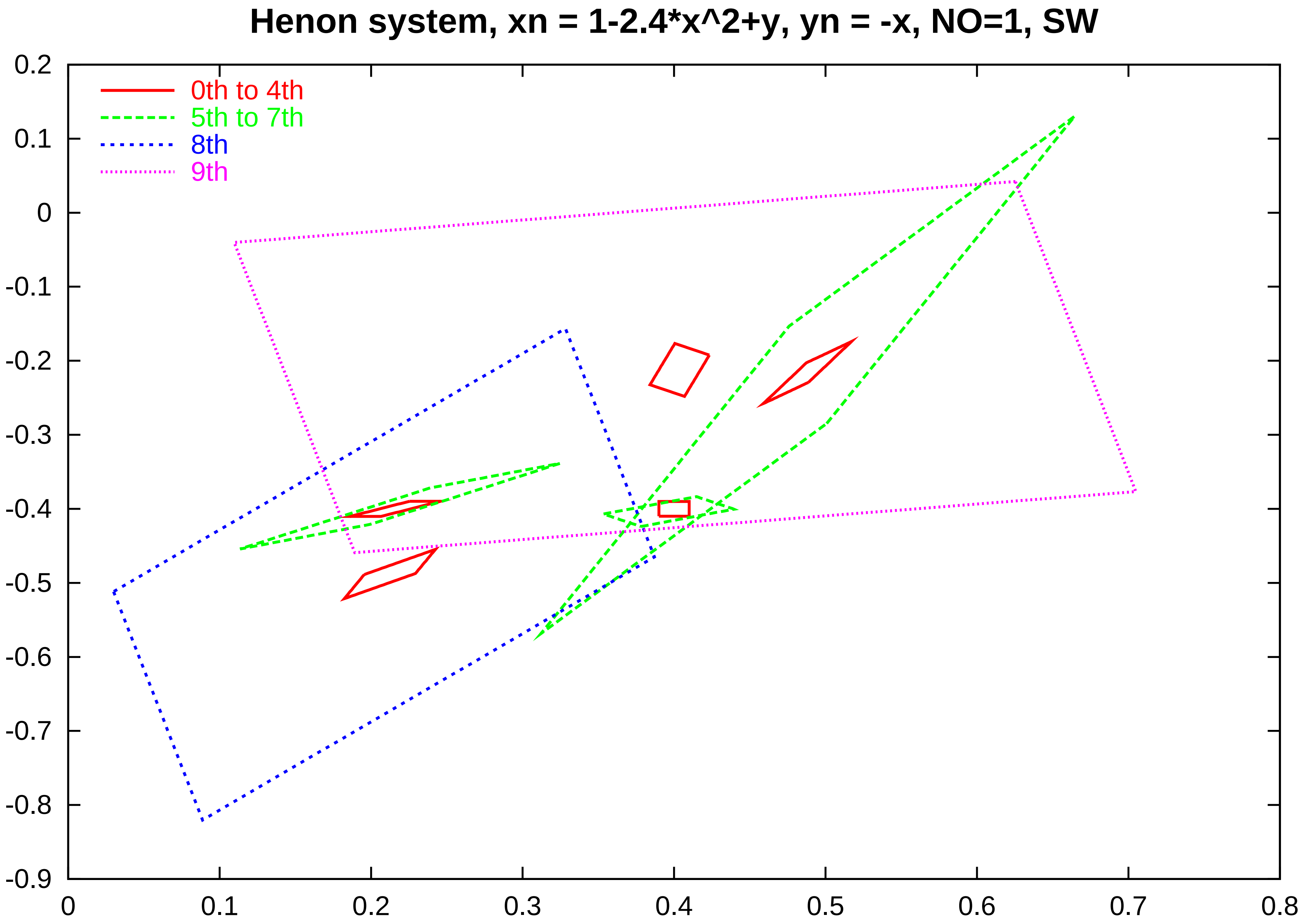


Henon system, xn = 1-2.4*x^2+y, yn = -x, the positions at each step


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

Definitions - Taylor Model Exponential, cont.

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

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

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

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

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

where

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

$$\arcsin^{(3)}(a) = (1+2a^2)/(1-a^2)^{5/2}, ...$$

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++

TM Enclosure Theorem

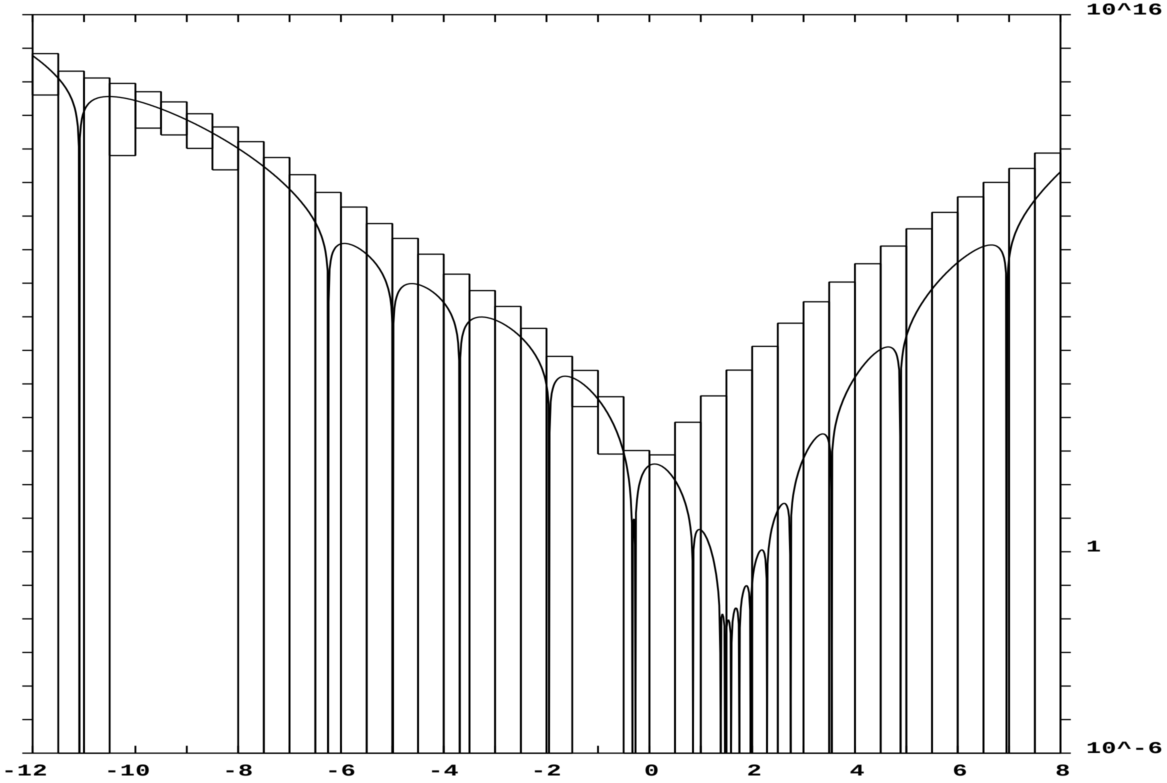
Theorem (Taylor Model Enclosure Theorem) Let the function $f : R^v \to R^v$ be contained within $P_f + I_f$ over the domain $D \subset R^v$. Let the function $g : R^v \to R$ be given by a code list comprised of finitely many elementary operations and intrinsic functions, and let g be defined over the range of an enclosure of $P_f, +I_f$. Let $P + I$ be the result obtained by executing the code list for g in admissible FP Taylor model arithmetic, beginning with the Taylor model $P_f + I_f$. Then $P + I$ is an enclosure for $g \circ f$ over D.

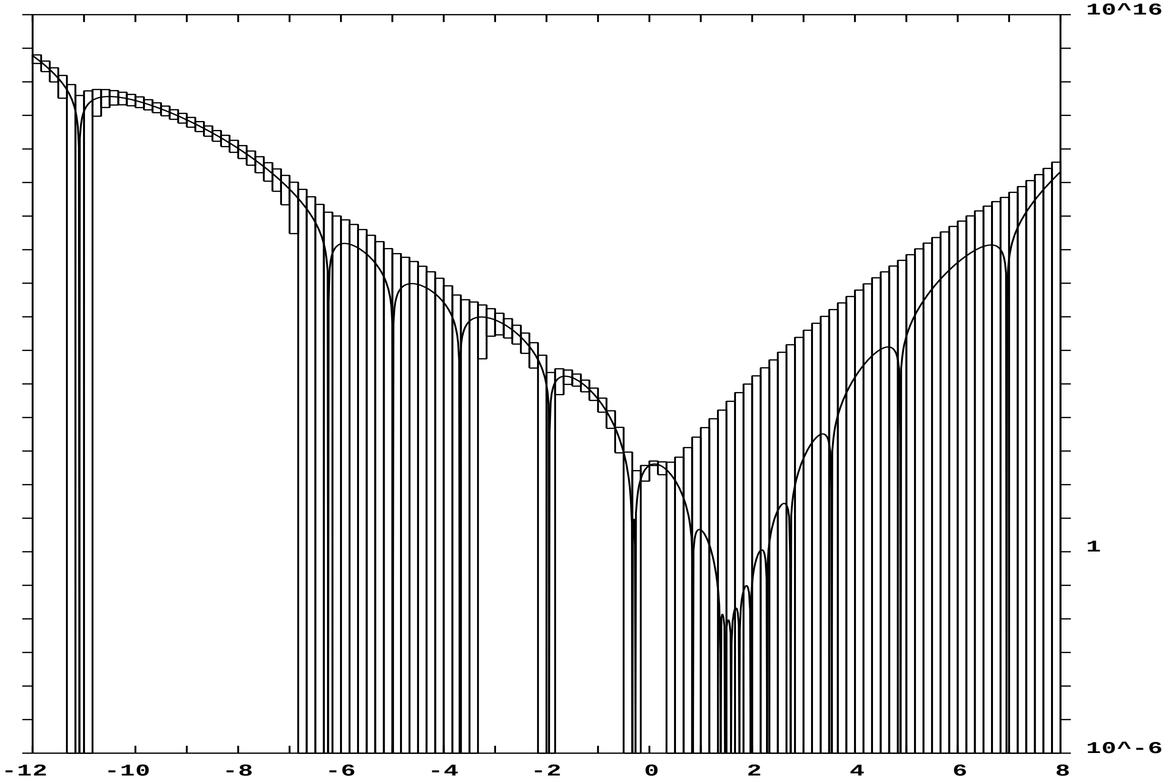
Proof The proof follows by induction over the code list of g from the elementary properties of the Taylor model arithmetic.

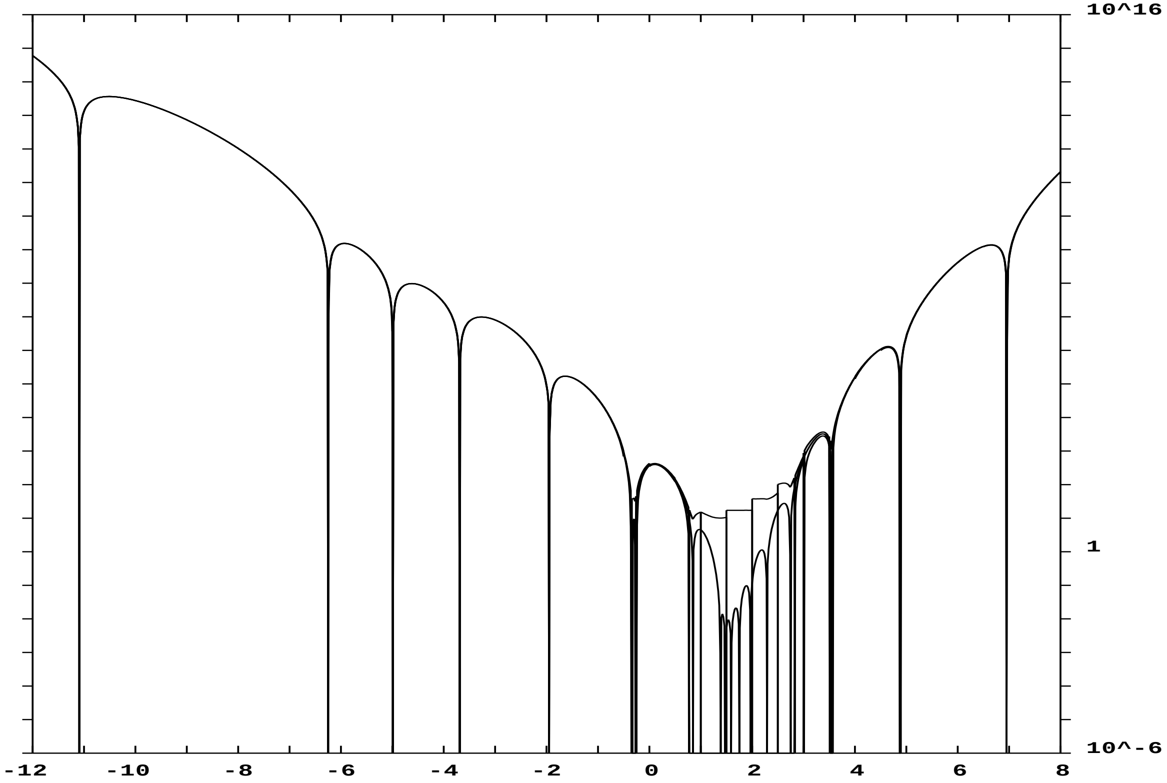
Apparently the presence of the floating point errors entails that P is not precisely the **Taylor polynomial**. In a similar fashion, also the **scaling property** of the remainder bound in a rigorous sense is lost. However, these properties of Taylor models are **retained in an approximate fashion**.

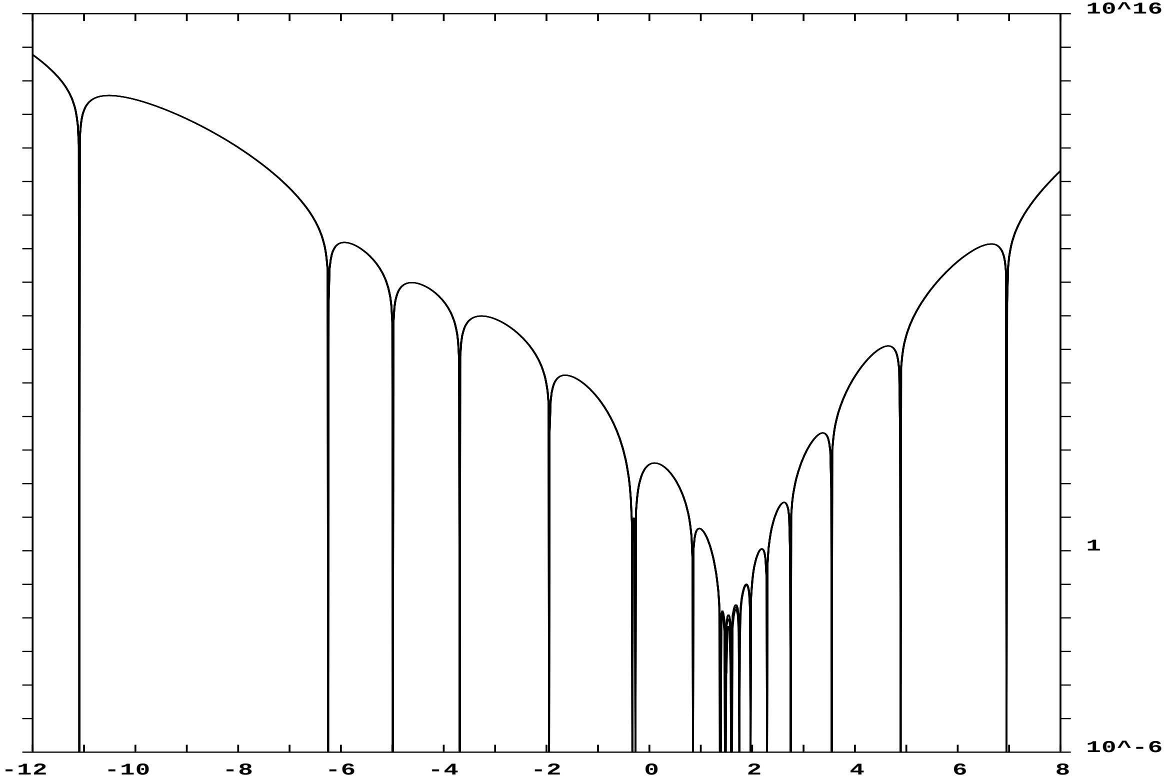
Important TM Algorithms

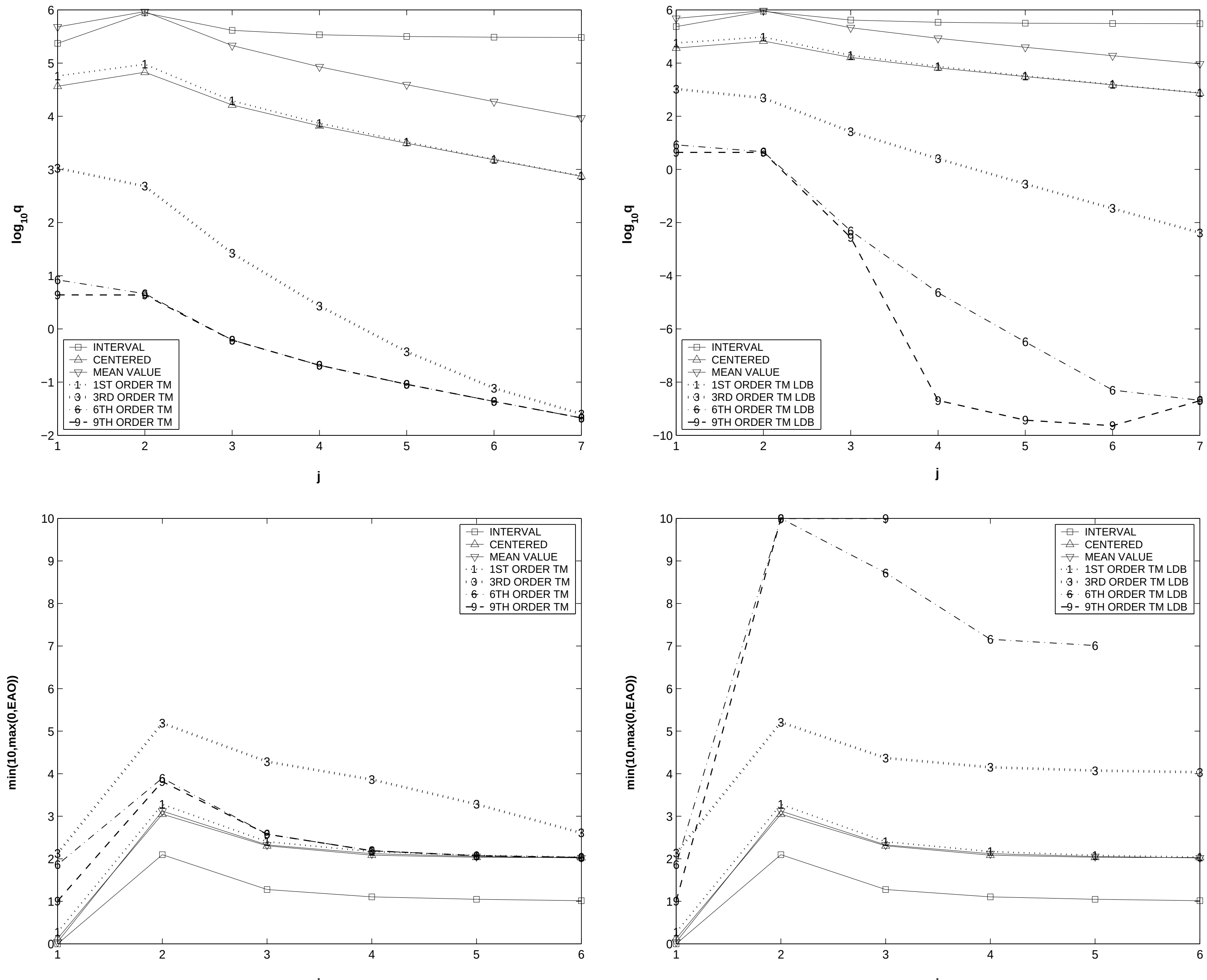
- **Range Bounding** (Evaluate f as TM, bound polynomial, add remainder bound)
- **Quadrature** (Evaluate f as TM, integrate polynomial and remainder bound)
- **Implicit Equations** (Obtain TMs for implicit solutions of TM equations)
- **Superconvergent** Interval Newton Method (Application of Implicit Equations)
- **ODEs** (Obtain TMs describing dependence of final coordinates on initial coordinates)
- **Implicit ODEs and DAEs**
- **Complex Arithmetic** (Describe complex ranges as two-dimensional TMs)

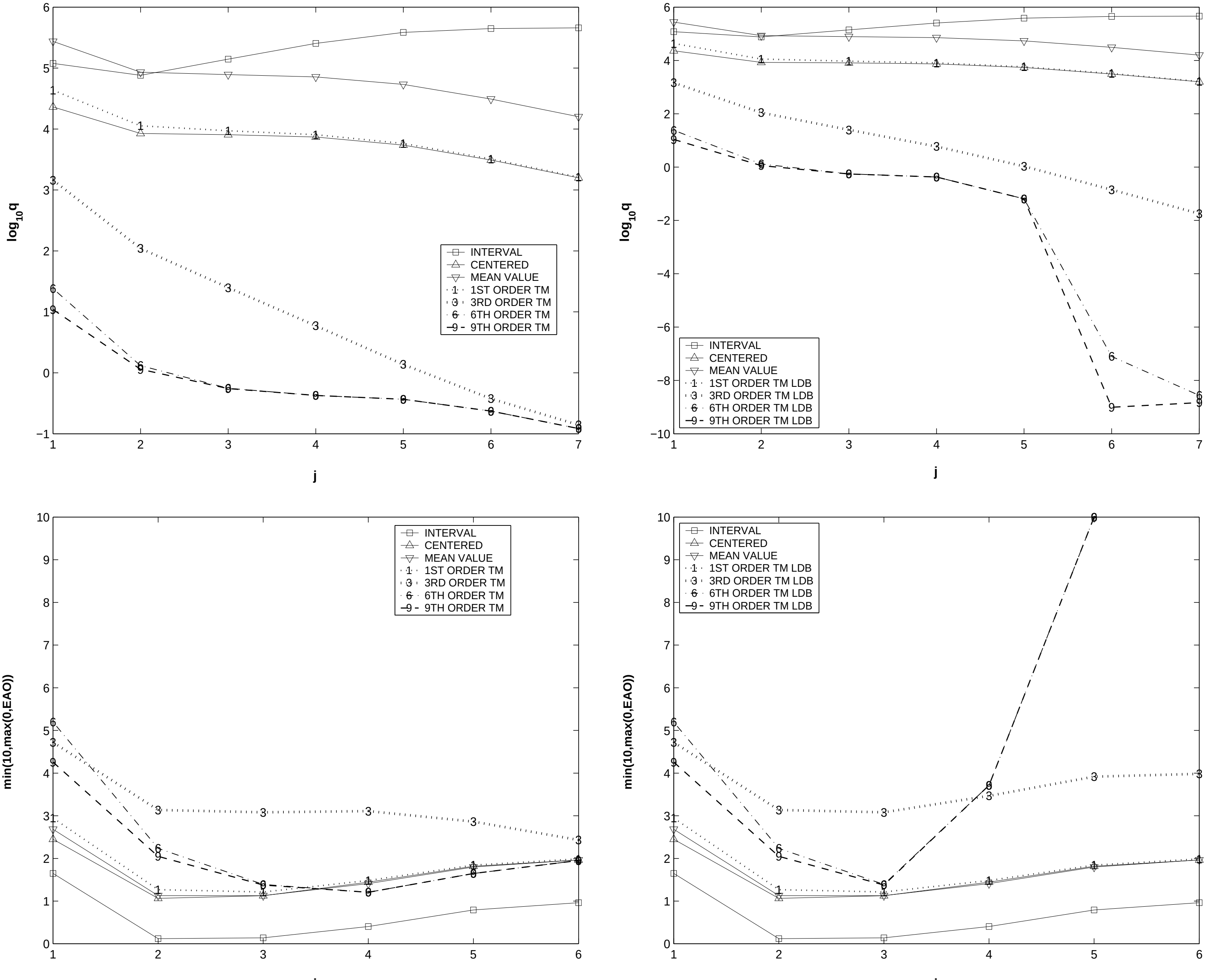

10^16
1
10^-6
-12
-10
-8
-6
-4
-2
0
2
4
6
8


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


log$_{10}$q
j
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))
6
5
4
3
2
1
0
−1
−2
−4
−6
−8
−10
1
2
3
4
5
6
7
8
9
10



log10q
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
log10q
j
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))
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

A Simple 1D Example

Approximate the cos function by its power series to order 60:

$$f(x) = \sum_{i=0}^{30} (-1)^i \frac{x^{2i}}{(2i)!}.$$

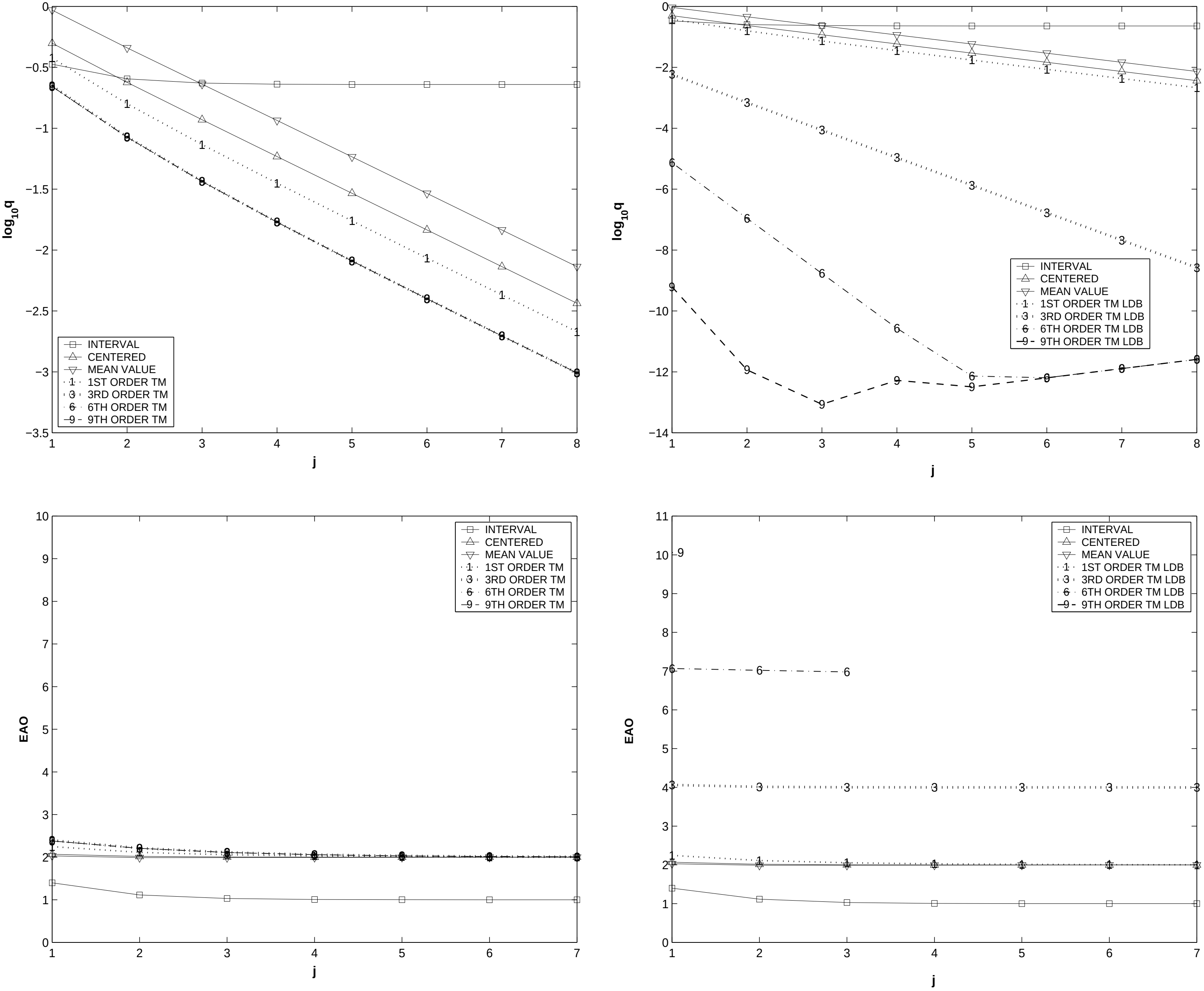
Several nice properties:

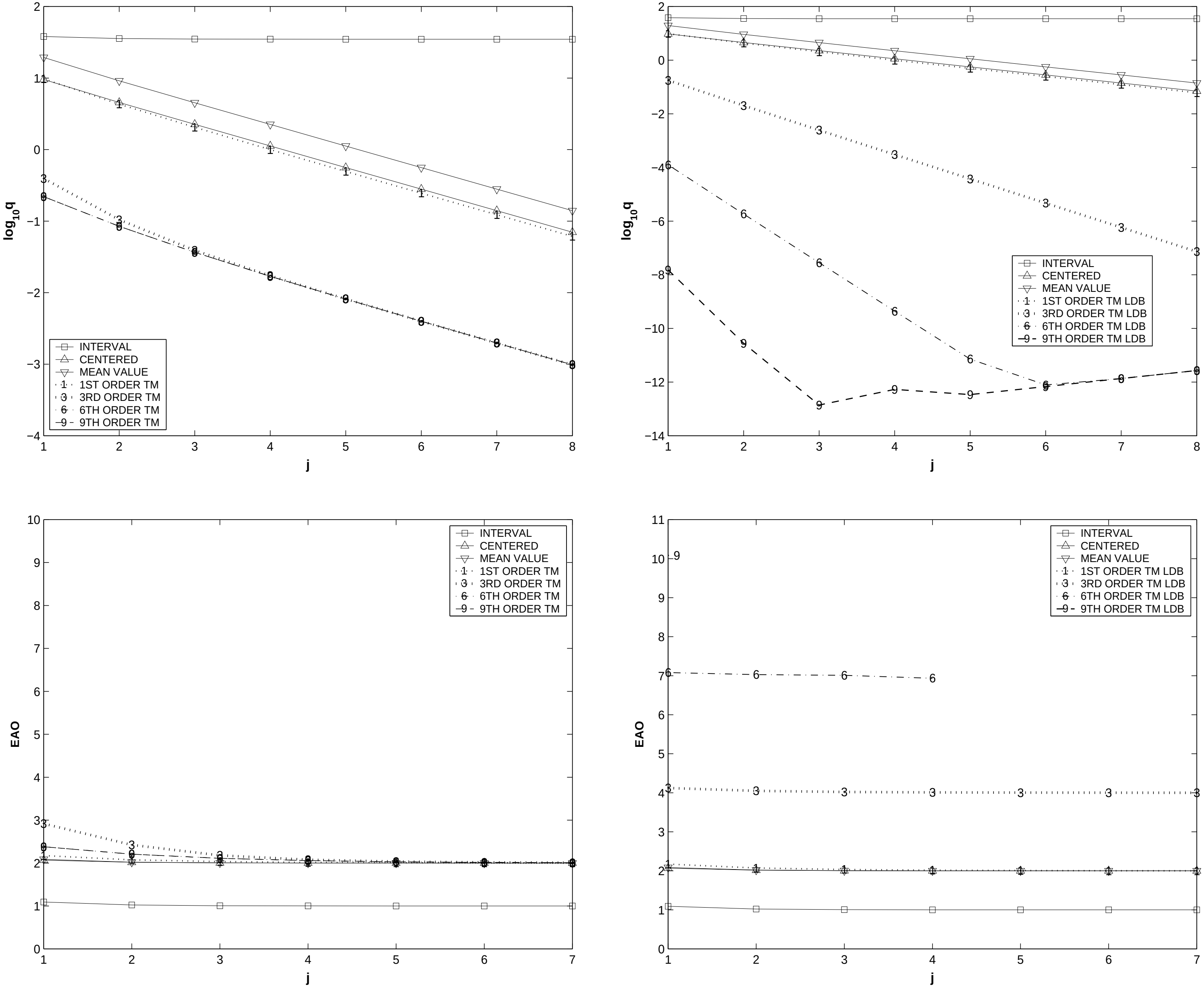
1. Properties of the function are well known
2. Dependency increases with x from very small to very large
3. Periodicity allows the study of the same functional behavior with varying amounts of dependency
4. Study at points with both non-stationary and stationary points is possible

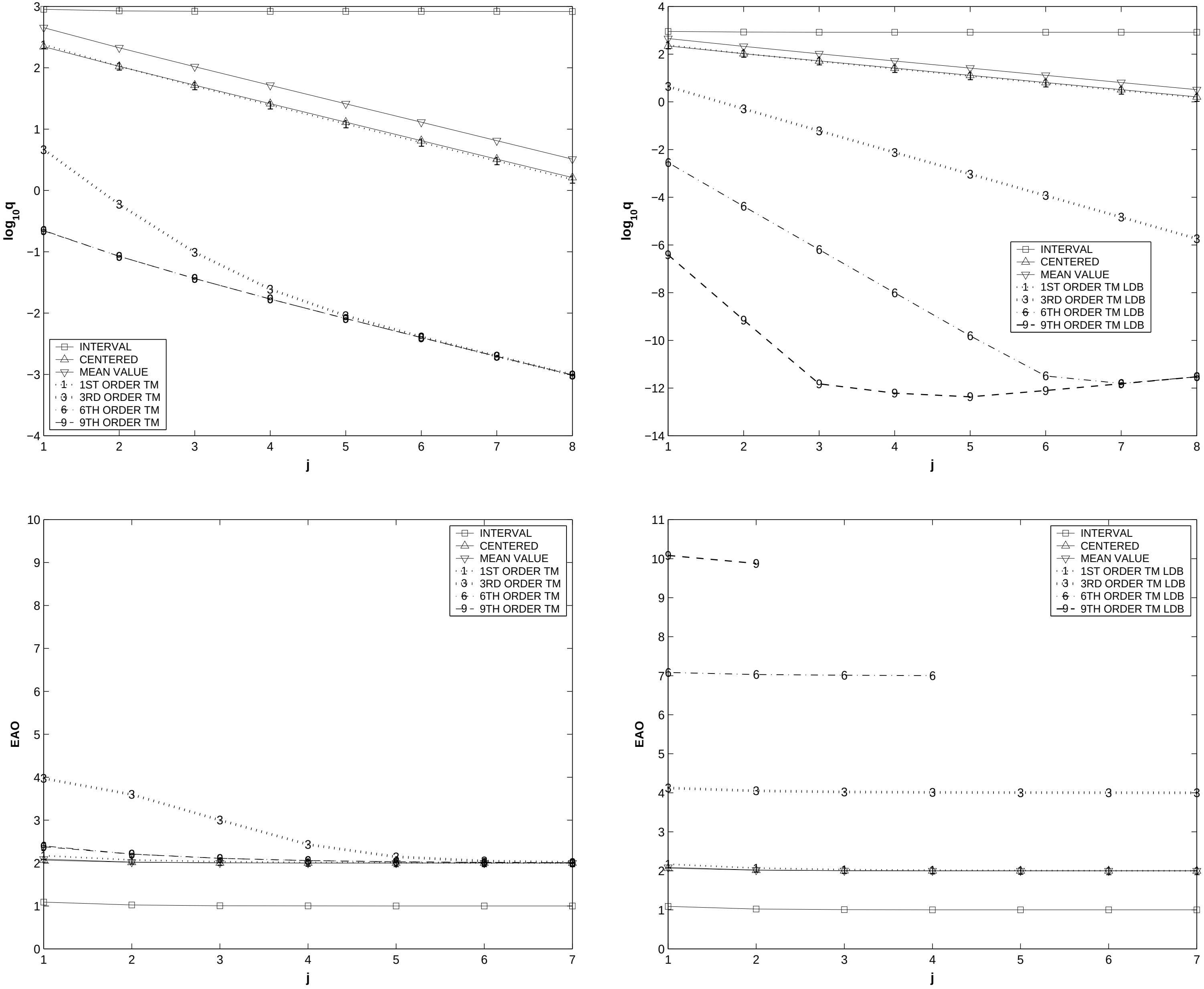
Study results for expansion points $x_0 = n \cdot \pi/4$ for

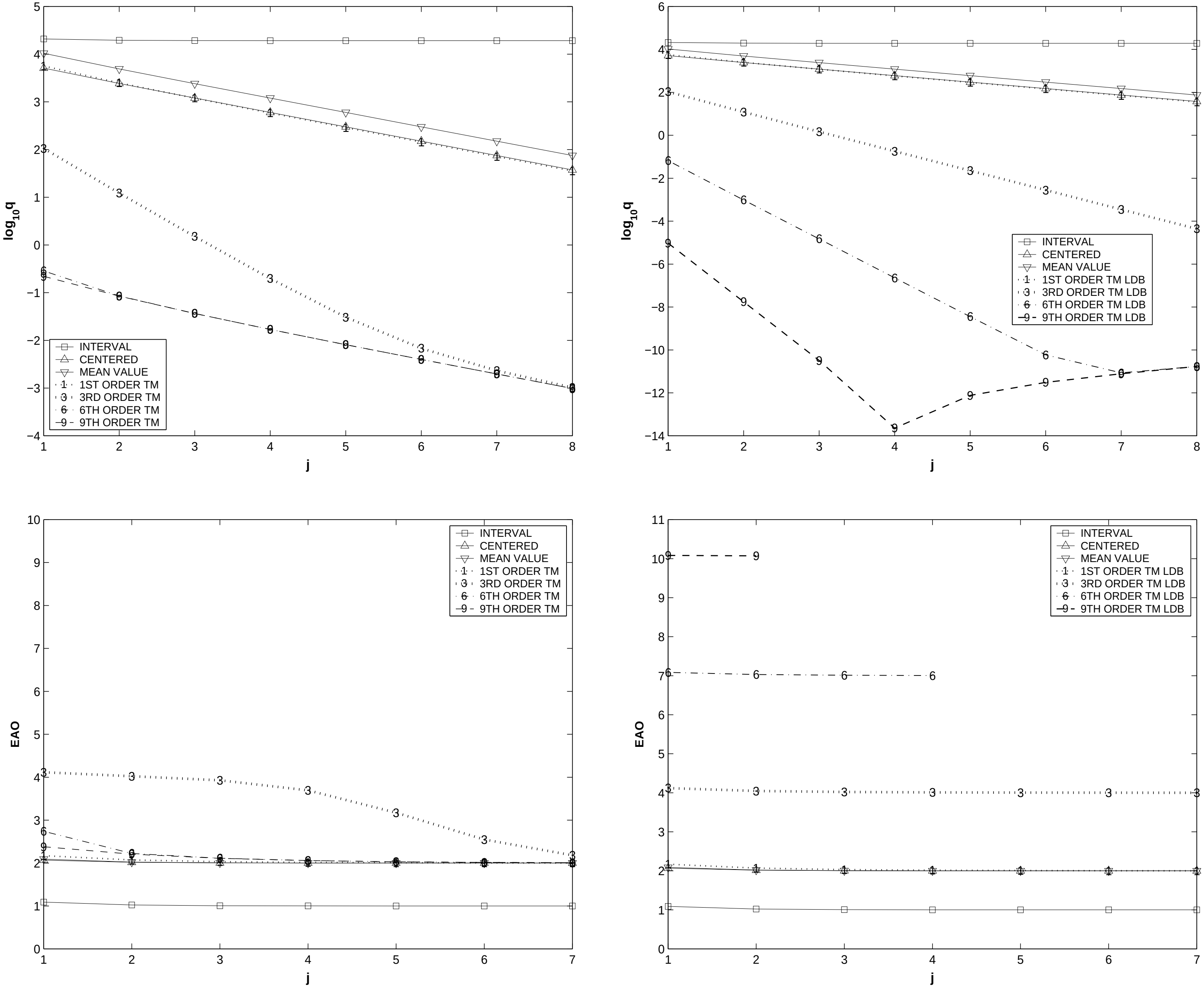
$$n = 1, 5, 9, 13 \text{ and } n = 0, 4, 8, 12.$$

For each of these points, domains are $x_0 + [-2^{-j}, 2^{-j}]$ for $j = 1, ..., 8$.

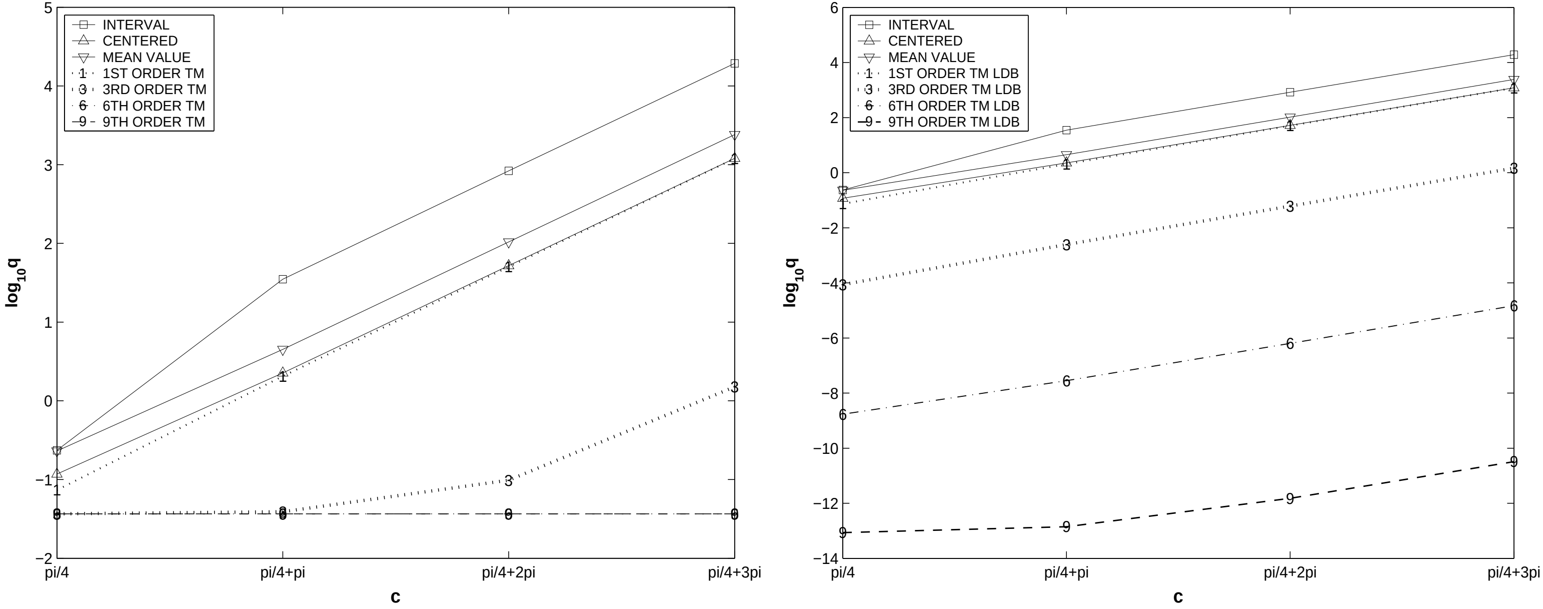

$\log_{10}q$
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
j
1
2
3
4
5
6
7
8
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
$\log_{10}q$
0
-2
-4
-6
-8
-10
-12
-14
j
1
2
3
4
5
6
7
8
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB
EAO
0
1
2
3
4
5
6
7
8
9
10
j
1
2
3
4
5
6
7
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
EAO
0
1
2
3
4
5
6
7
8
9
10
11
j
1
2
3
4
5
6
7
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB

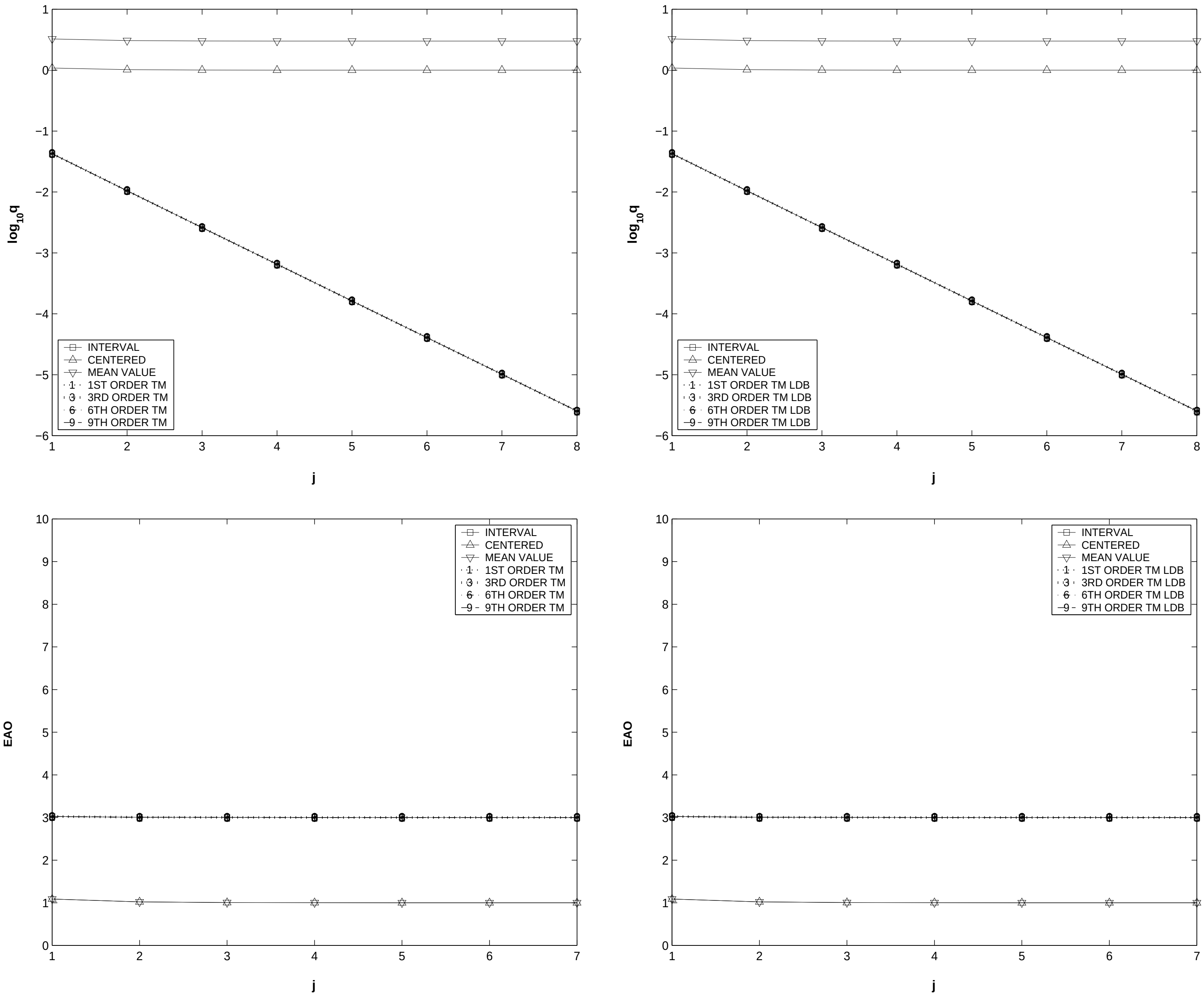

$\log_{10} q$
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
$\log_{10} q$
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB
EAO
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
EAO
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB

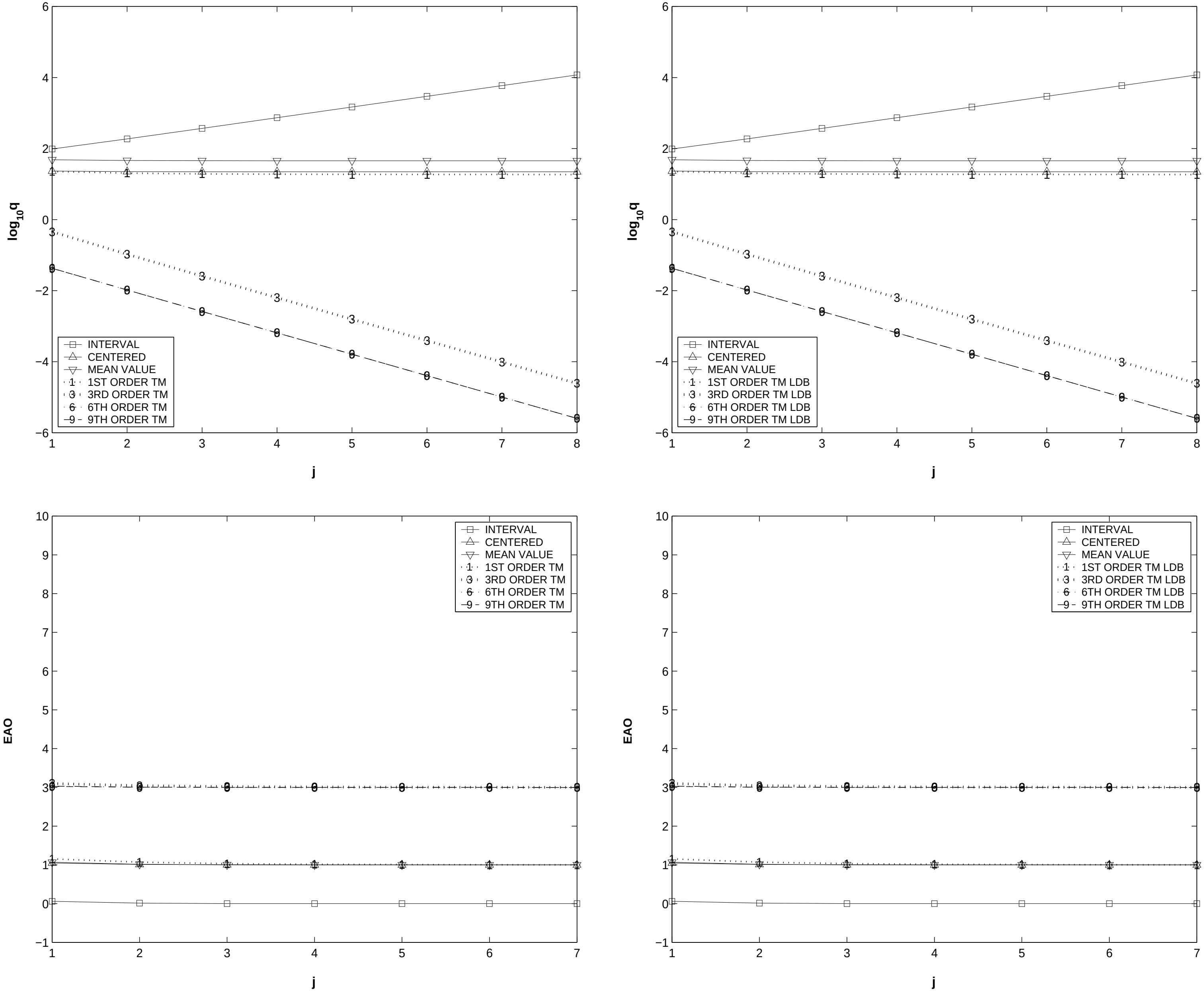

log10q
j
1
2
3
4
5
6
7
8
3
2
1
0
−1
−2
−3
−4
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
log10q
j
4
2
0
−2
−4
−6
−8
−10
−12
−14
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB
EAO
j
1
2
3
4
5
6
7
10
9
8
7
6
5
4
3
2
1
0
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
EAO
j
11
10
9
8
7
6
5
4
3
2
1
0
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB

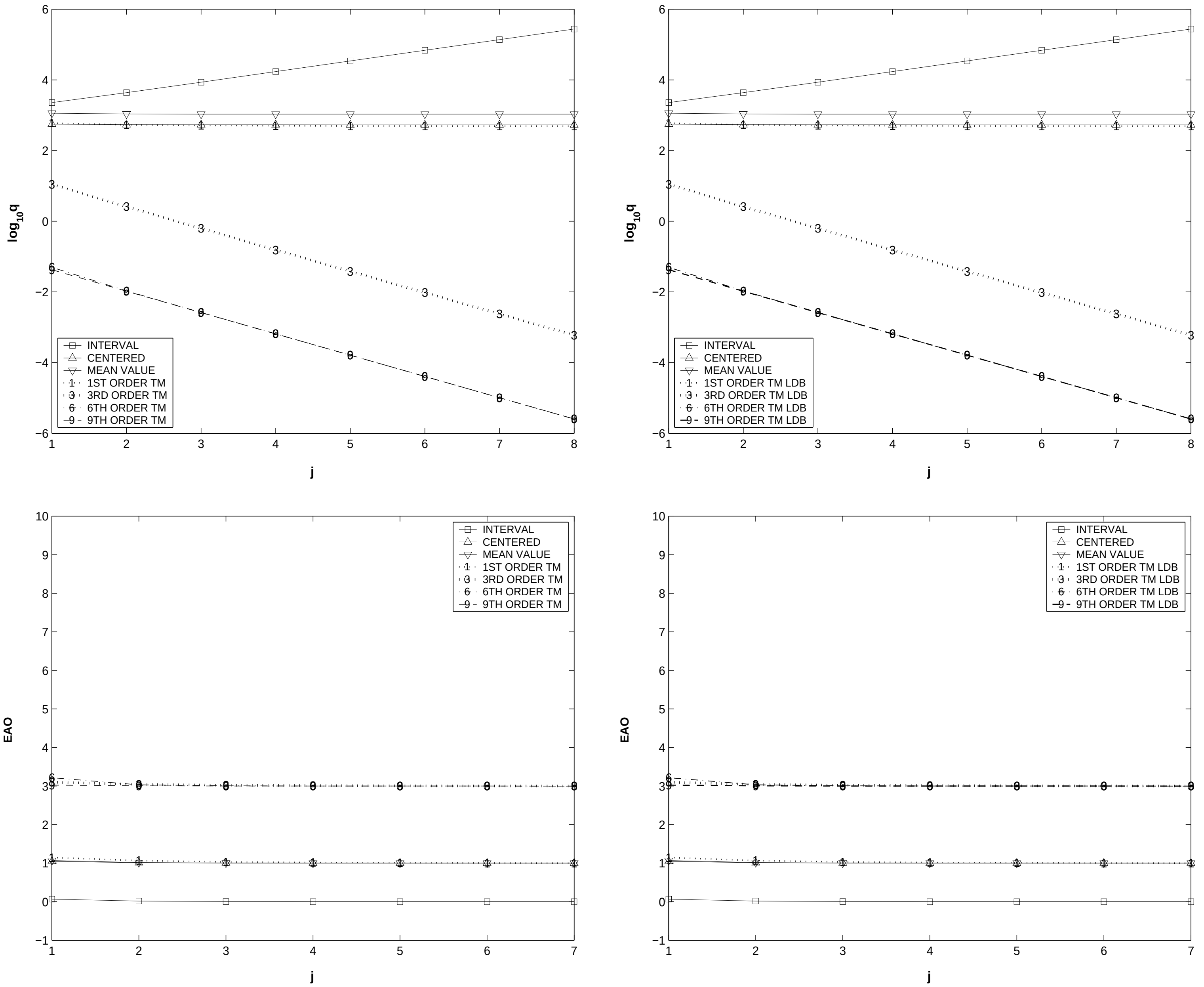


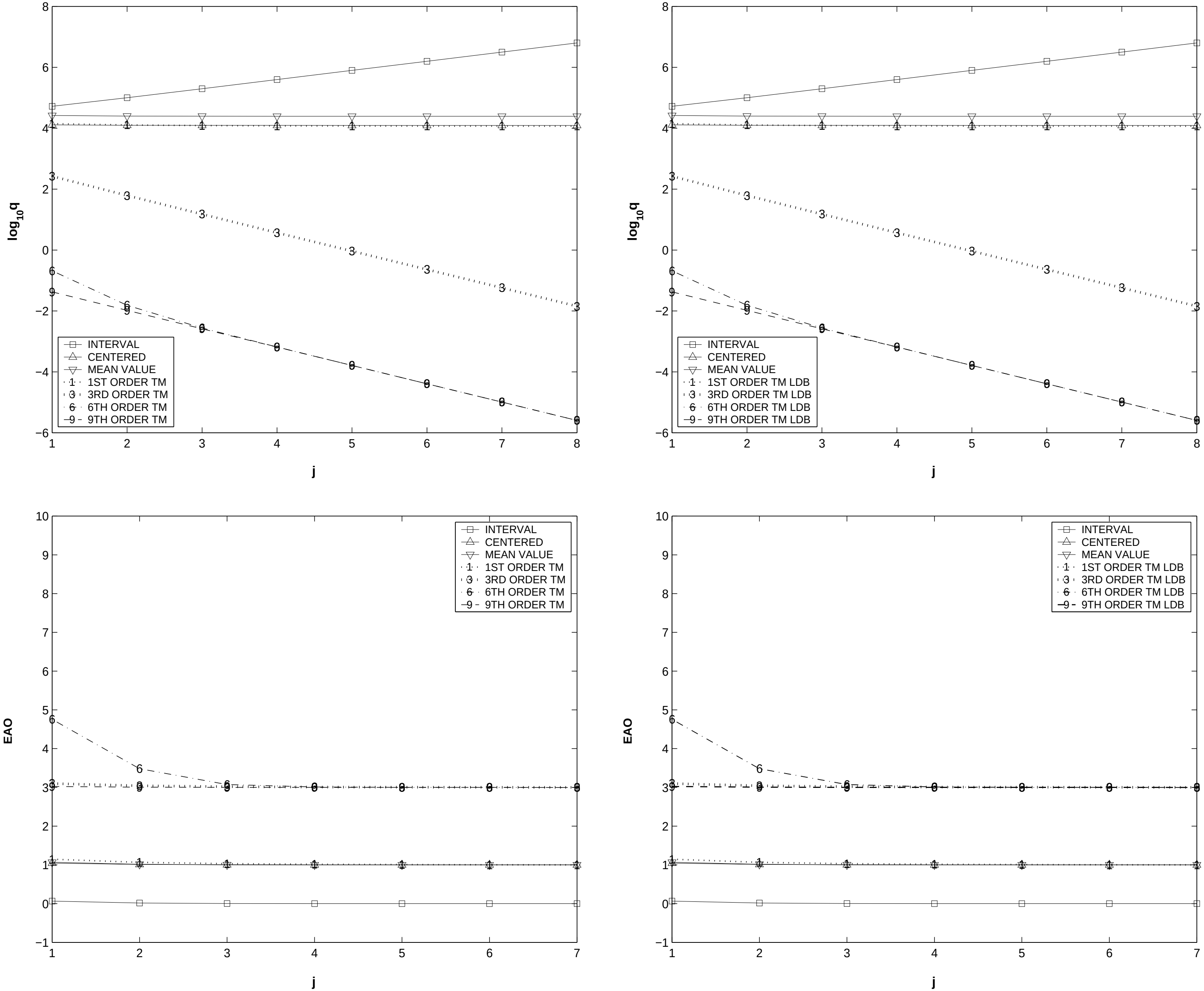
log₁₀q
j
1 2 3 4 5 6 7 8
5 4 3 2 1 0 −1 −2 −3 −4
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
log₁₀q
j
6 4 2 0 −2 −4 −6 −8 −10 −12 −14
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB
EAO
j
1 2 3 4 5 6 7
10 9 8 7 6 5 4 3 2 1 0
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
EAO
j
11 10 9 8 7 6 5 4 3 2 1 0
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB

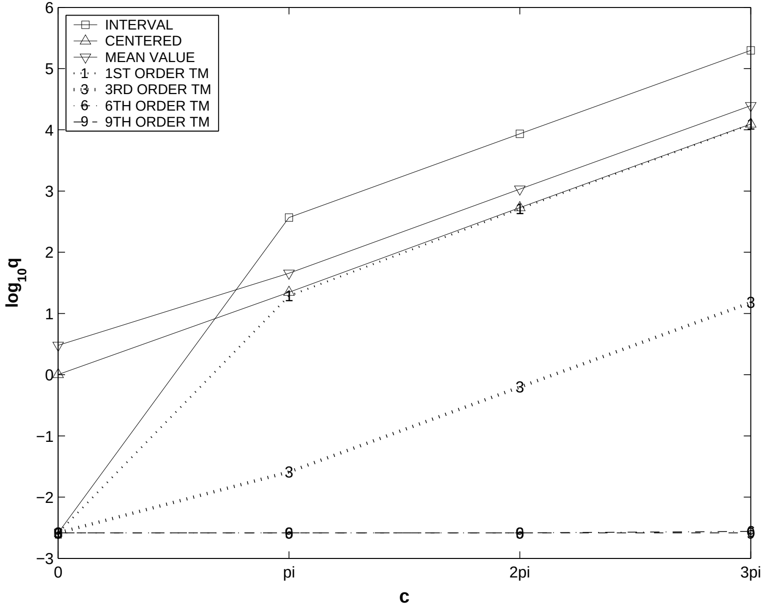

INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
$\log_{10} q$
5
4
3
2
1
0
−1
−2
pi/4
pi/4+pi
pi/4+2pi
pi/4+3pi
c
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB
$\log_{10} q$
6
4
2
0
−2
−4
−6
−8
−10
−12
−14
pi/4
pi/4+pi
pi/4+2pi
pi/4+3pi
c

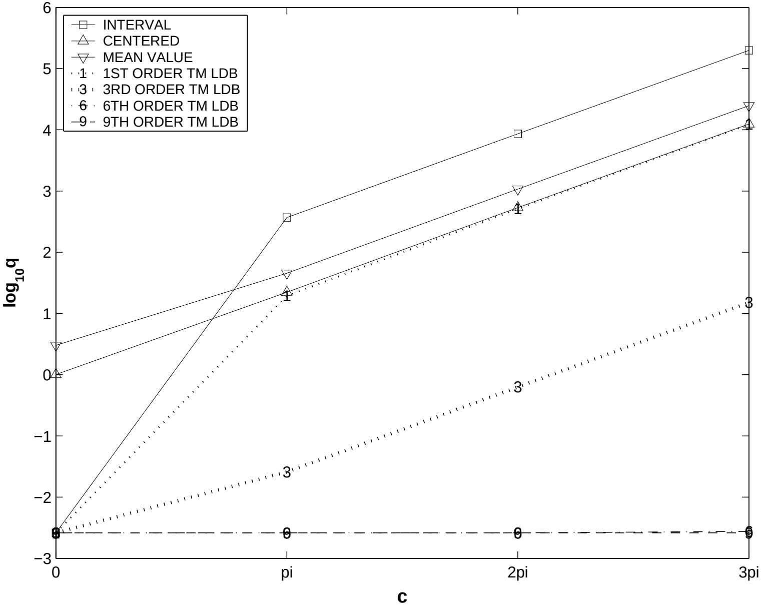

1
0
−1
−2
−3
−4
−5
−6
log10q
1
2
3
4
5
6
7
8
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
log10q
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB
10
9
8
7
6
5
4
3
2
1
0
EAO
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER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


log$_{10}$q
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
log$_{10}$q
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB
EAO
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
EAO
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB


log10q
j
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
EAO
6
4
2
0
−2
−4
−6
1
2
3
4
5
6
7
8
10
9


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


INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM
6
5
4
3
2
1
0
−1
−2
−3
$\log_{10}q$
0
pi
2pi
3pi
c


INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB
6
5
4
3
2
1
0
−1
−2
−3
$\log_{10}q$
0
pi
2pi
3pi
c

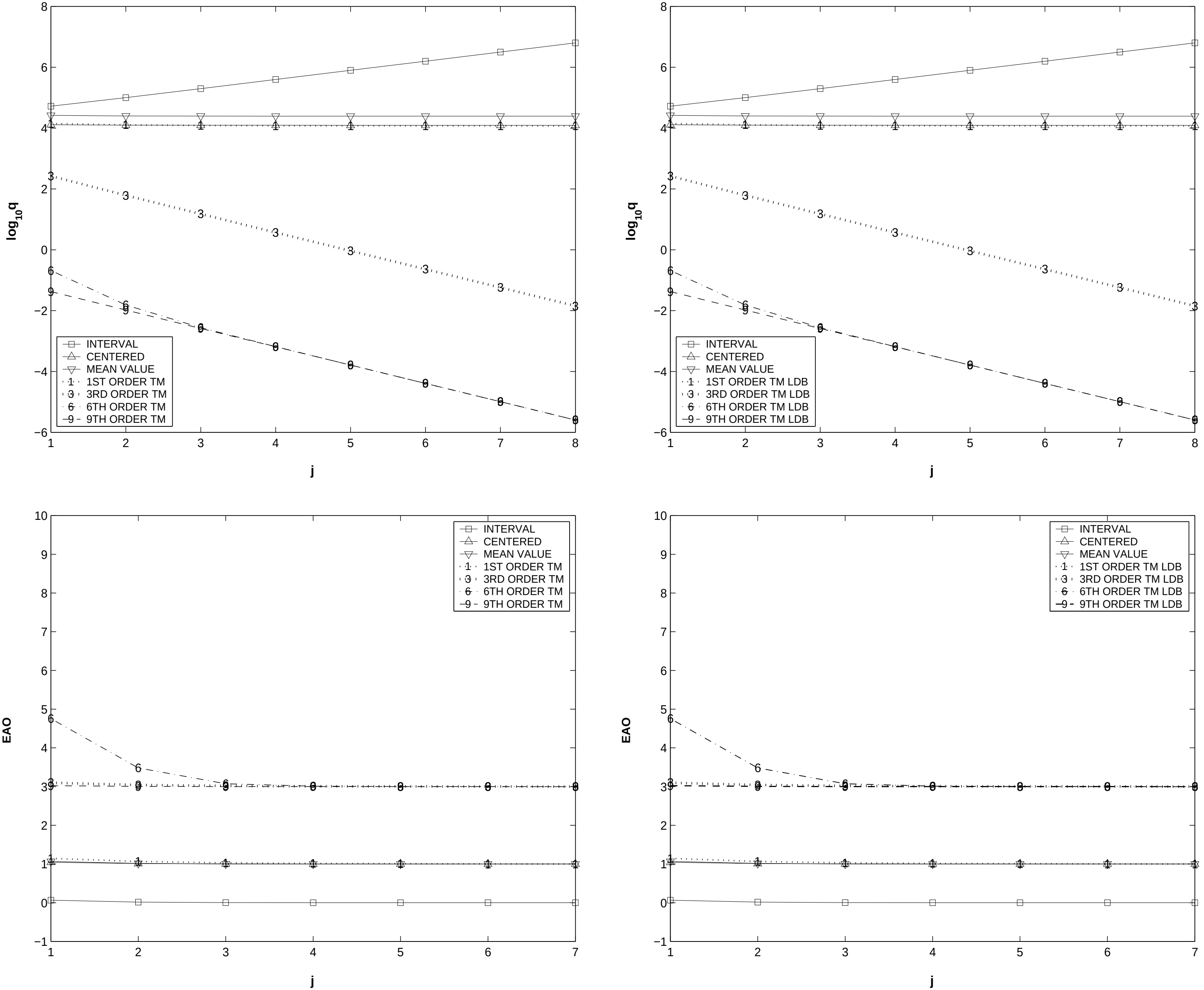
Implementation of TM Arithmetic

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++



log₁₀q
j
8
6
4
2
0
−2
−4
−6
1
2
3
4
5
6
7
8
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
EAO
10
9
7
5
3
1
−1

The Henon Map

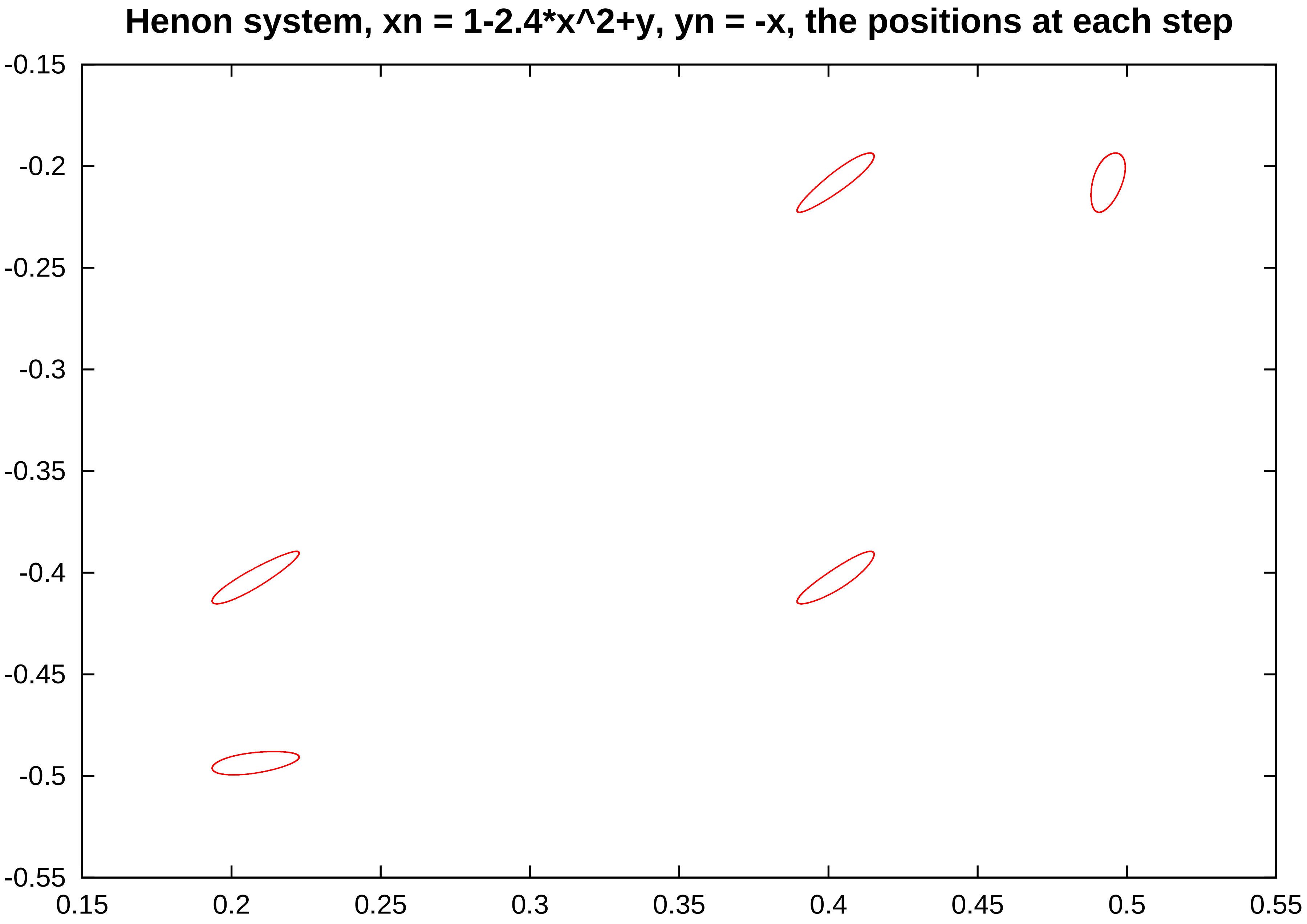
Henon Map: frequently used elementary example that exhibits many of the well-known effects of nonlinear dynamics, including chaos, periodic fixed points, islands and symplectic motion. The dynamics is two-dimensional, and given by

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

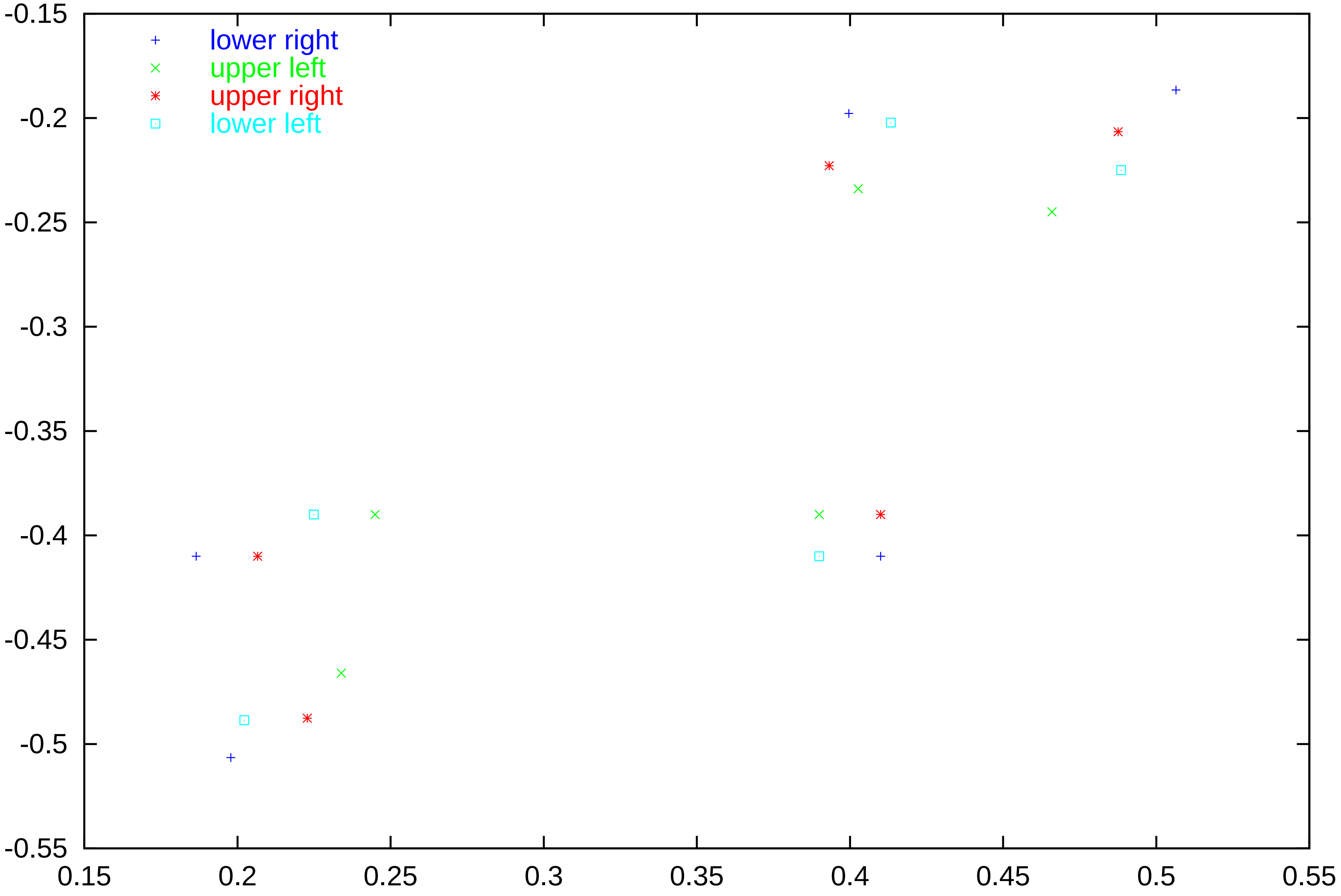
It can easily be seen that the motion is area preserving for $|\beta| = 1$.We consider

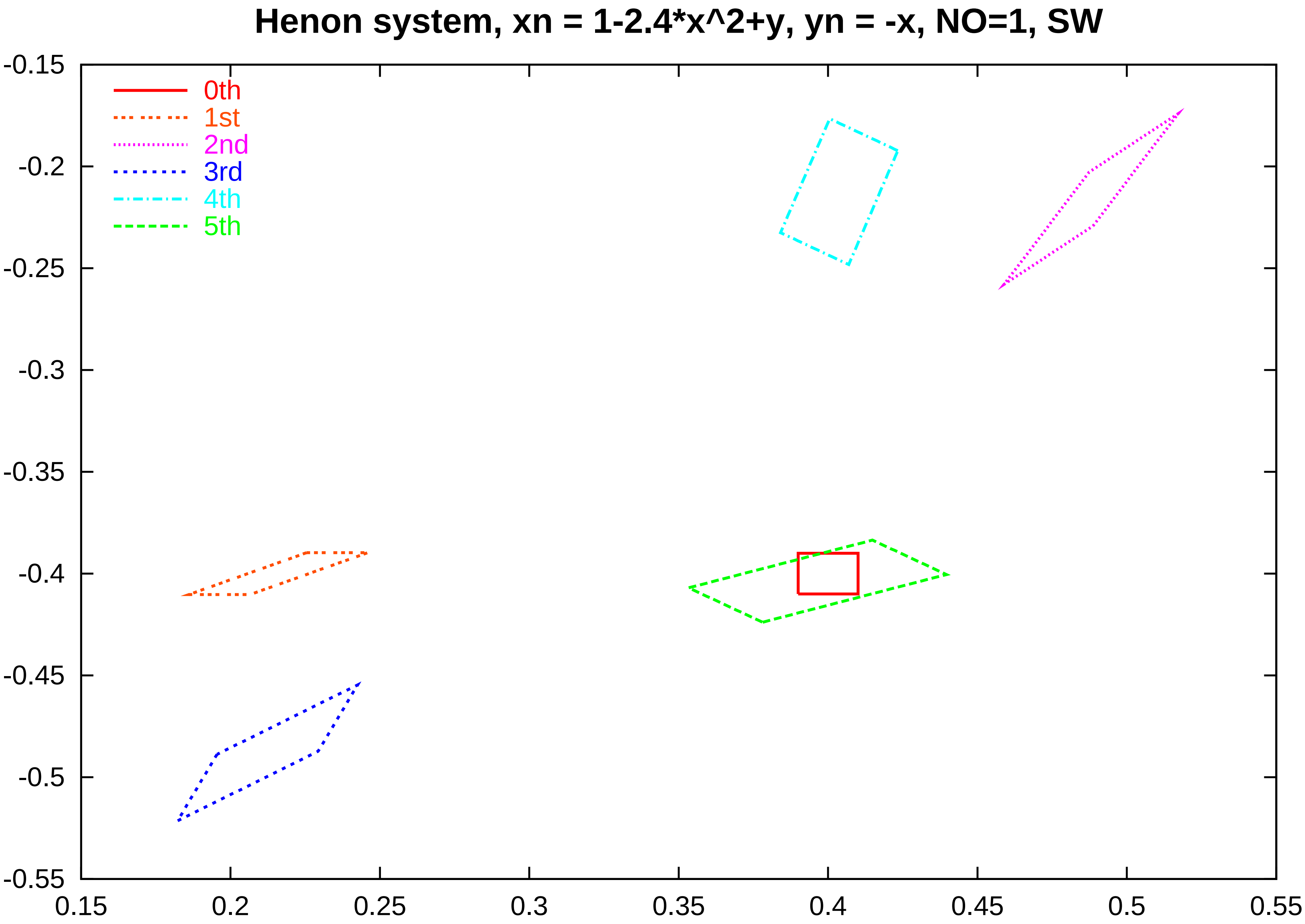
$$\alpha = 2.4 \text{ and } \beta = -1,$$

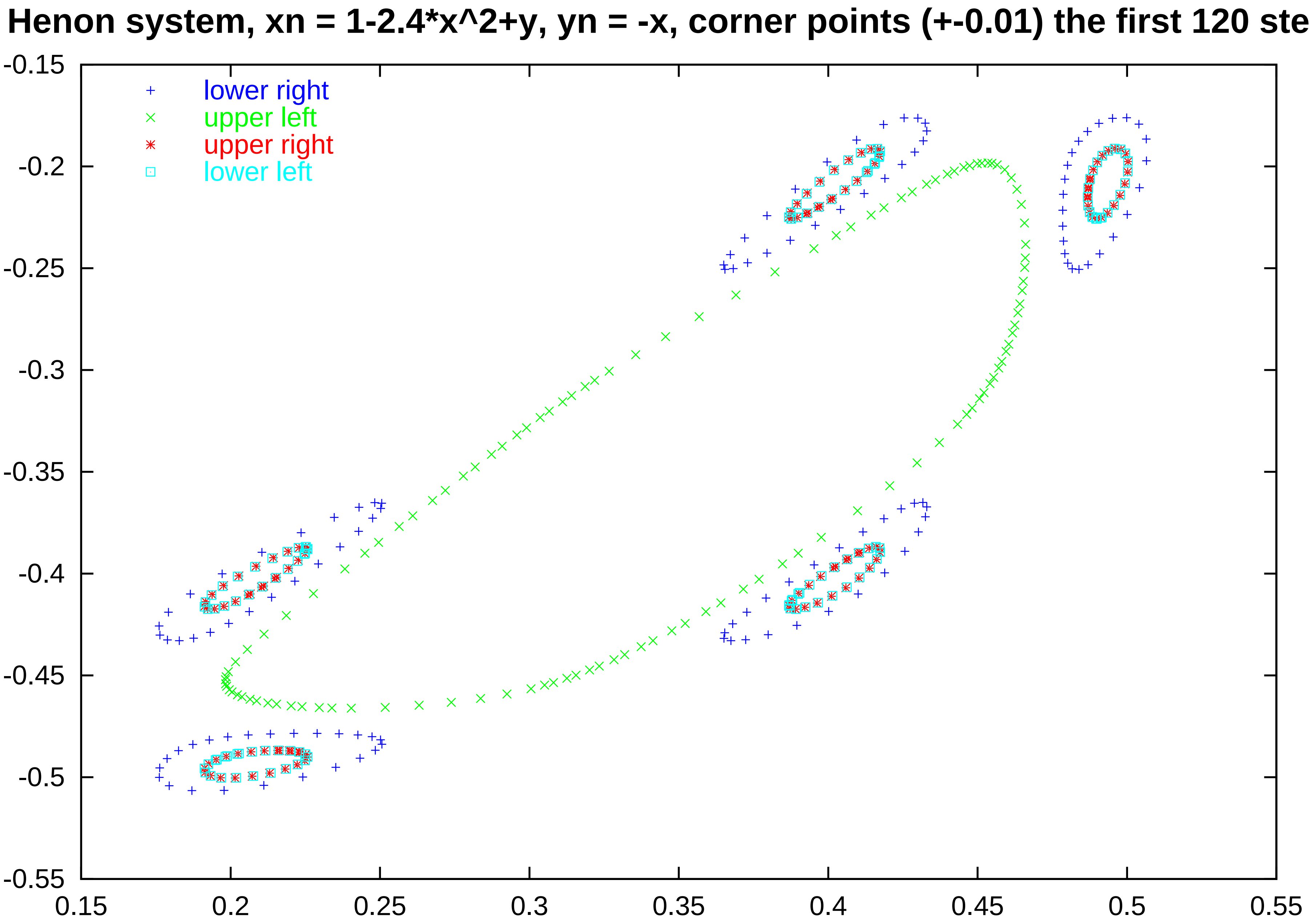
and concentrate on initial boxes of the from $(x_0, y_0) \in (0.4, \, -0.4) + [-d, d]^2$.


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

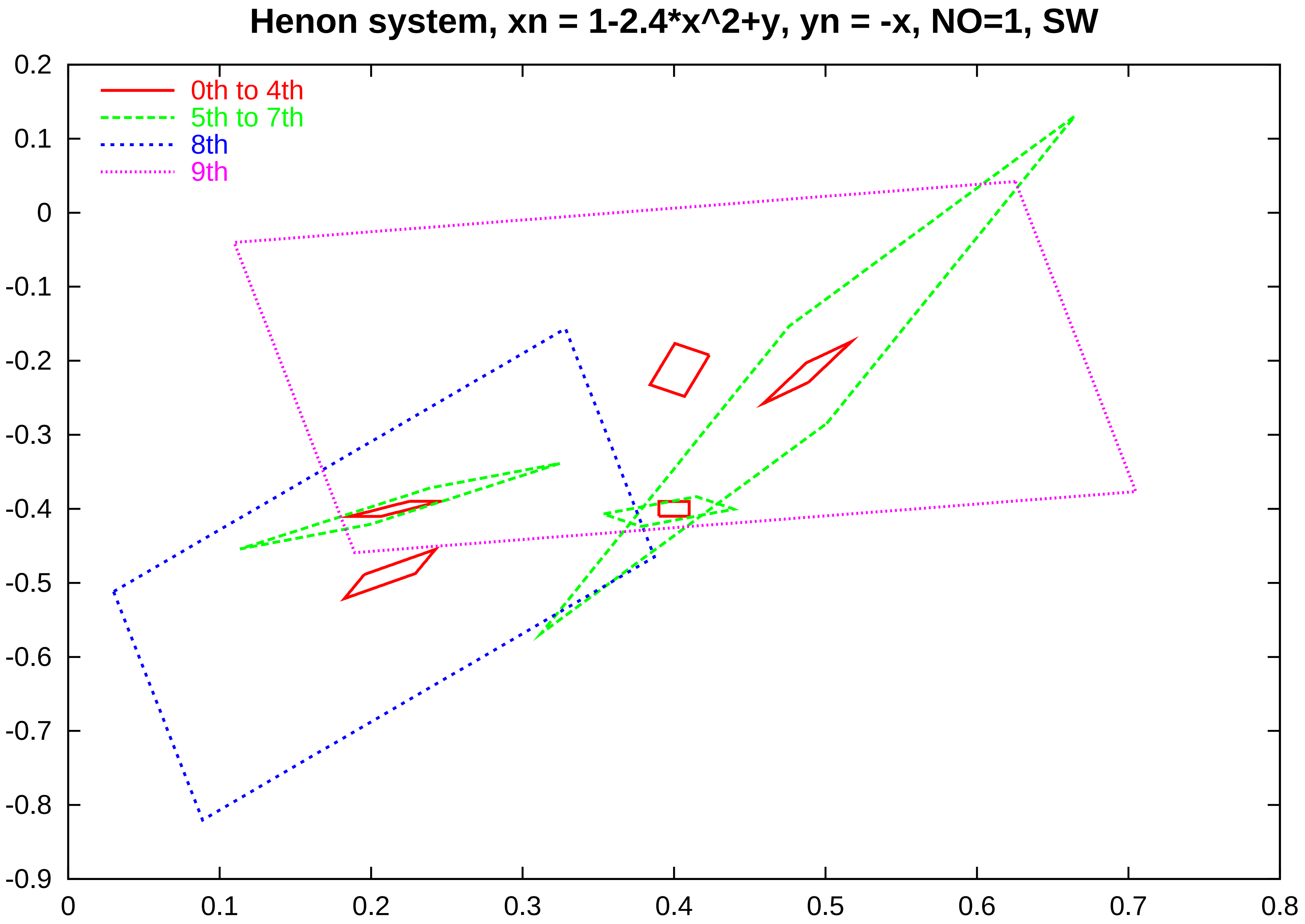
Henon system, xn = 1-2.4*x^2+y, yn = -x, corner points (+-0.01) the first 5 steps

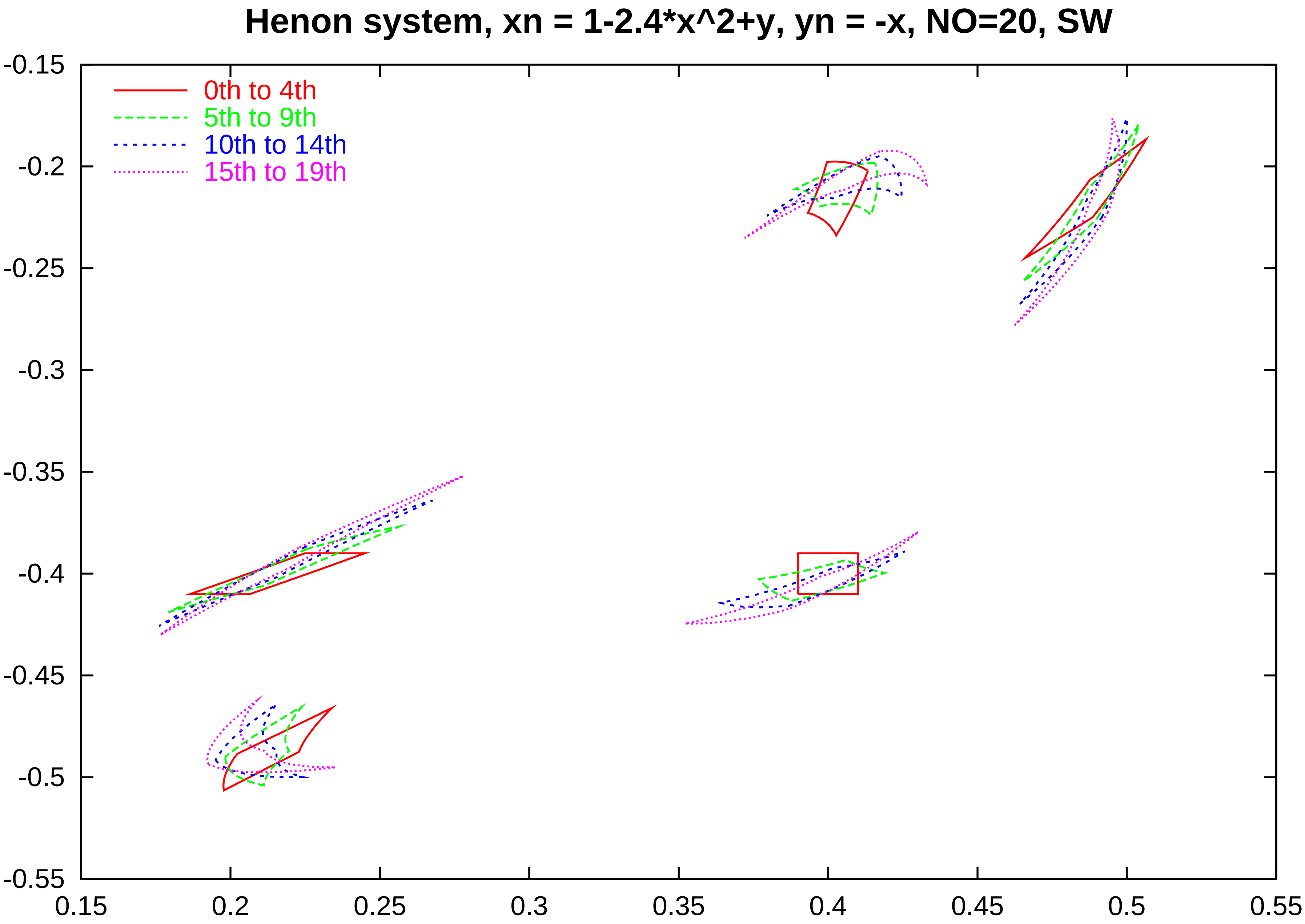


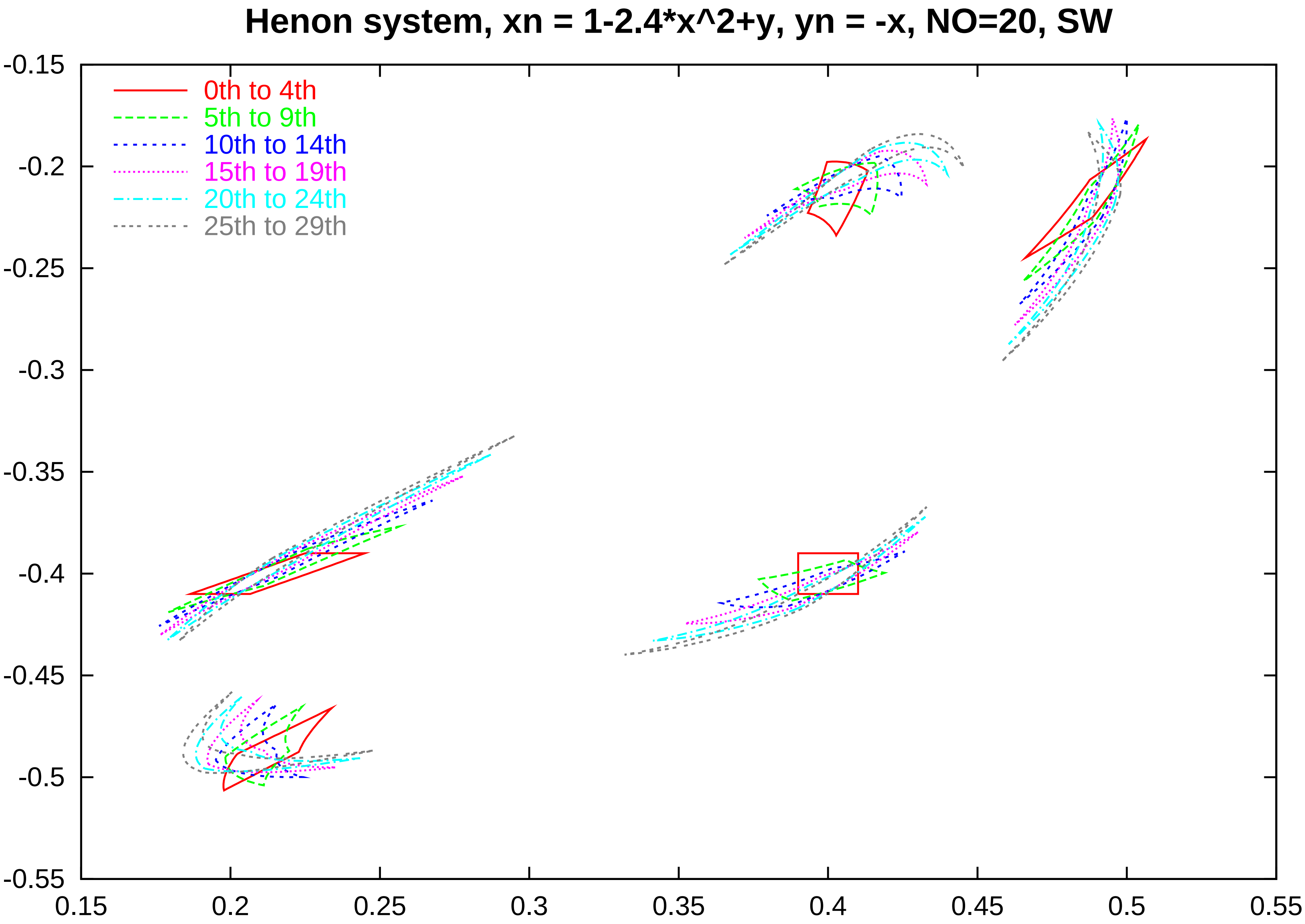

Henon system, xn = 1-2.4*x^2+y, yn = -x, NO=1, SW
0th
1st
2nd
3rd
4th
5th
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55



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


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

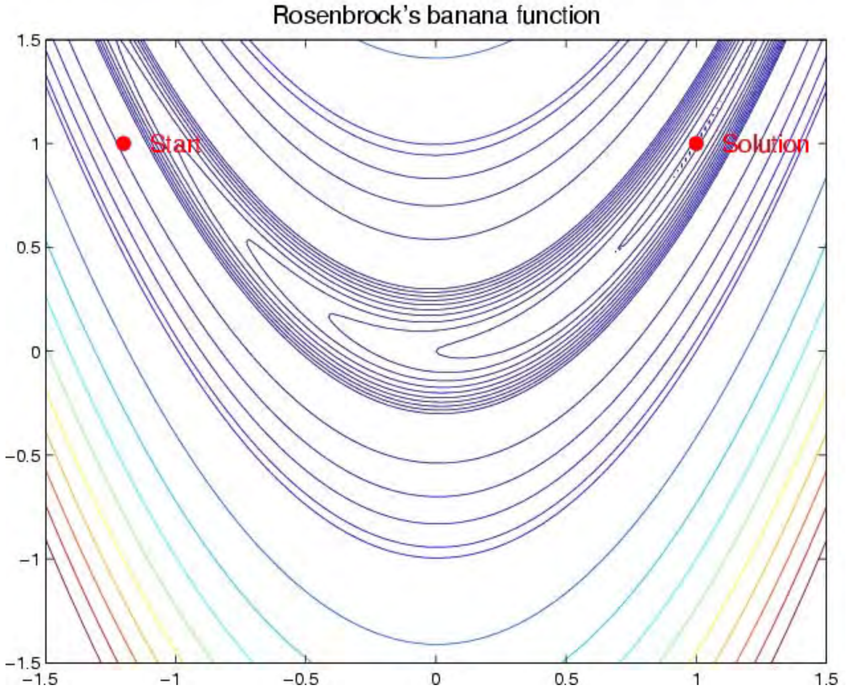

Henon system, xn = 1-2.4*x^2+y, yn = -x, NO=20, SW
0th to 4th
5th to 9th
10th to 14th
15th to 19th
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55


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

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)
```

Ordered LDL (Extended Cholesky) Decomposition

Given Quadratic Form with symmetric H

$$Q(x) = \frac{1}{2} x^t \cdot H \cdot x + a \cdot x + b$$

We determine Ordered LDL Decomposition (L: lower diagonal with unit diagonal, D: diagonal) as follows

1. Pre-sort rows and columns by the size of their diagonal elements
2. Successively execute conventional $L^t DL$ decomposition step in interval arithmetic, beginning by representing every element of H by a thin interval; in step i:
   (a) If $l(D(i,i)) > 0$ proceed to the next row and column.
   (b) If $u(D(i,i)) < 0$ exchange row and column i with row and column $i+1$, $i+2$, ... If a positive element is found, increment i and repeat. If none is found, stop.

Note: Correction Matrix In case $0 \in D(i,i)$, apply small correction C to H, i.e. study $H + C$ instead of H, such that all elements of D are clearly positive or negative. $|C|$ is lumped into the remainder bound of the original problem.

Ordered LDL Decomposition - Result

Have obtained representation of H as LDL composition

$$P^t H P = L^t D L$$

- First p elements of D satisfy $l(D(i,i)) > 0$
- Remaining $(n-p)$ elements of D will satisfy $u(D(i,i)) < 0$

Proposition: Sufficiently near a local minimizer, D will contain only positive elements. Furthermore, in the wider vicinity of the local minimizer, the number of negative elements in D will decrease as the minimizer is approached.

Simply follows from continuity of the matrix D as a function of position

The QDB (Quadratic Dominated Bounder) Algorithm

1. Let u be an external cutoff. Initialize $u = \min(u, Q(C))$. Initialize list with all 3^n surfaces for study.
2. If no boxes are remaining, terminate. Otherwise select one surface S of highest dimension.
3. On S, apply LDB. If a complete rejection is possible, strike S from the list and proceed to step 2. If a partial rejection is possible, strike the respective surfaces of S from the list and proceed to step 2.
4. Determine the definiteness of the Hessian of Q when restricted to S
5. If the Hessian is not p.d. strike S from the list and proceed to step 2.
6. If the Hessian is p.d., determine the corresponding critical point c.
7. If c is fully inside S, strike S and all surfaces of S from the list, update u $= \min(u, Q(c))$,and proceed to step 2
8. If c not inside S, strike S. If certain components of c lie between -1 and $+1$, strike the corresponding surfaces and proceed to step 2

The QDB Algorithm - Properties

The QDB algorithm has the following properties.

1. The quadratic bounder QDB has the third order approximation property.
2. The effort of finding the minimum requires the study of at most 3^n surfaces.
3. In the p.d. case, the computational effort requires at most the study of 2^n surfaces
4. Because of extensive box striking, in practice, the numbers of boxes to study is usually much much less.

The QDB Algorithm - Properties

The QDB algorithm has the following properties.

1. The quadratic bounder QDB has the third order approximation property.
2. The effort of finding the minimum requires the study of at most 3^n surfaces.
3. In the p.d. case, the computational effort requires at most the study of 2^n surfaces
4. Because of extensive box striking, in practice, the numbers of boxes to study is usually much much less.

But still, it is desirable to have something **FASTER**.

The QFB (Quadratic Fast Bounder) Algorithm

Let $P+I$ be a given Taylor model. Idea. Decompose into two parts

$$P+I=(P-Q)+I+Q \text{ and observe}$$

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

Choose Q such that

1. Q can be easily bounded from below
2. $P-Q$ is sufficiently simplified to allow bounding above given cutoff.

First possibility: Let H be p.d. part of P, set

$$Q=x^tHx$$

Then $l(Q)=0$. Removes all second order parts of P (!) Better yet:

$$Q_{x_0}=(x-x_0)^tH(x-x_0)$$

Allows to manipulate linear part. Works for ANY x_0 in domain. Still $l(Q_{x_0})=0$.

Which choices for x_0 are good?

The QFB Algorithm - Properties

Most critical case: near local minimizer, so H is the entire purely quadratic part of P.

Theorem: If x_0 is the (unique) minimizer of quadratic part of P on the domain of $P + I$, then the lower bound of the linear part of $(P - Q_{x_0})$ is zero. 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.

Proof: Follows readily from Kuhn-Tucker conditions. If x_0 inside, linear part vanishes completely. Otherwise, wlog if i-th component of x_0 is at left end, i-th partial there must be non-negative, so that we get non-negative contribution.

Remark: The closer x_0 is to the minimizer, the closer we are to order 3 cutoff.

Algorithm: (Third Order Cutoff Test). Let $x^{(n)}$ be a sequence of points that converges to the minimum x_0 of the convex quadratic part P_2 In step n, determine a bound of $(P - Q_{x_n})$ by interval evaluation, and assess whether the bound exceeds the cutoff threshold. If it does, reject the box and terminate; if it does not, proceed to the next point x_{n+1}.

The QMLoc Algorithm

Tool to generate efficient sequence $x^{(n)}$. Determine "feasible descent direction"

$$g_i^{(n)} = \begin{cases} -\frac{\partial Q}{\partial x_i} & \text{if } x_i^{(n)} \text{ inside} \\ \min\left(-\frac{\partial Q}{\partial x_i}, 0\right) & \text{if } x_i^{(n)} \text{ on right} \\ \max\left(-\frac{\partial Q}{\partial x_i}, 0\right) & \text{if } x_i^{(n)} \text{ on left} \end{cases}$$

Now move in direction of $g^{(n)}$ until we hit box or quadratic minimum along line. Very fast to do, can change set of active constraints very quickly.

Result: Cheap iterative third order cutoff.

Use of QFB - Example

Let $f_1(x) = \frac{1}{2}x^t \cdot A_v \cdot x - A_v \cdot (a \cdot x) + \frac{1}{2}a^t \cdot A_v \cdot a$ with

$$A_v = \begin{pmatrix} 2 & 3 & \dots & 3 \\ -1 & 2 & \dots & 3 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & 2 \end{pmatrix}$$

known to be p.d. with minimum a. Choose a random vector a, and 5^v boxes around it. Check box rejection with Interval evaluation, Centered Form, QFB. Output average number of QFB iterations.

Use of QFB - Example

Let $f_1(x) = \frac{1}{2}x^t \cdot A_v \cdot x - A_v \cdot (a \cdot x) + \frac{1}{2}a^t \cdot A_v \cdot a$ with

$$A_v = \begin{pmatrix} 2 & 3 & \dots & 3 \\ -1 & 2 & \dots & 3 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & 2 \end{pmatrix}$$

known to be p.d. with minimum a. Choose a random vector a, and 5^v boxes around it. Check box rejection with Interval evaluation, Centered Form, QFB. Output average number of QFB iterations.

```
v     N=5^v     NI         NC    NQFB Avg. Iter
2        25        25          8   1       1.1
4       625       625        308   1      0.31
```

Use of QFB - Example

Let $f_1(x) = \frac{1}{2}x^t \cdot A_v \cdot x - A_v \cdot (a \cdot x) + \frac{1}{2}a^t \cdot A_v \cdot a$ with

$$A_v = \begin{pmatrix} 2 & 3 & \dots & 3 \\ -1 & 2 & \dots & 3 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & 2 \end{pmatrix}$$

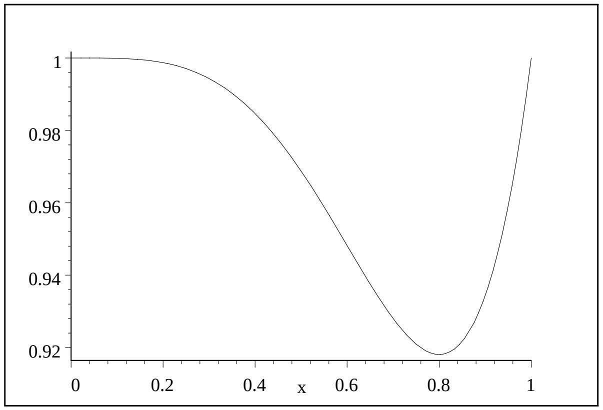
known to be p.d. with minimum a. Choose a random vector a, and 5^v boxes around it. Check box rejection with Interval evaluation, Centered Form, QFB. Output average number of QFB iterations.

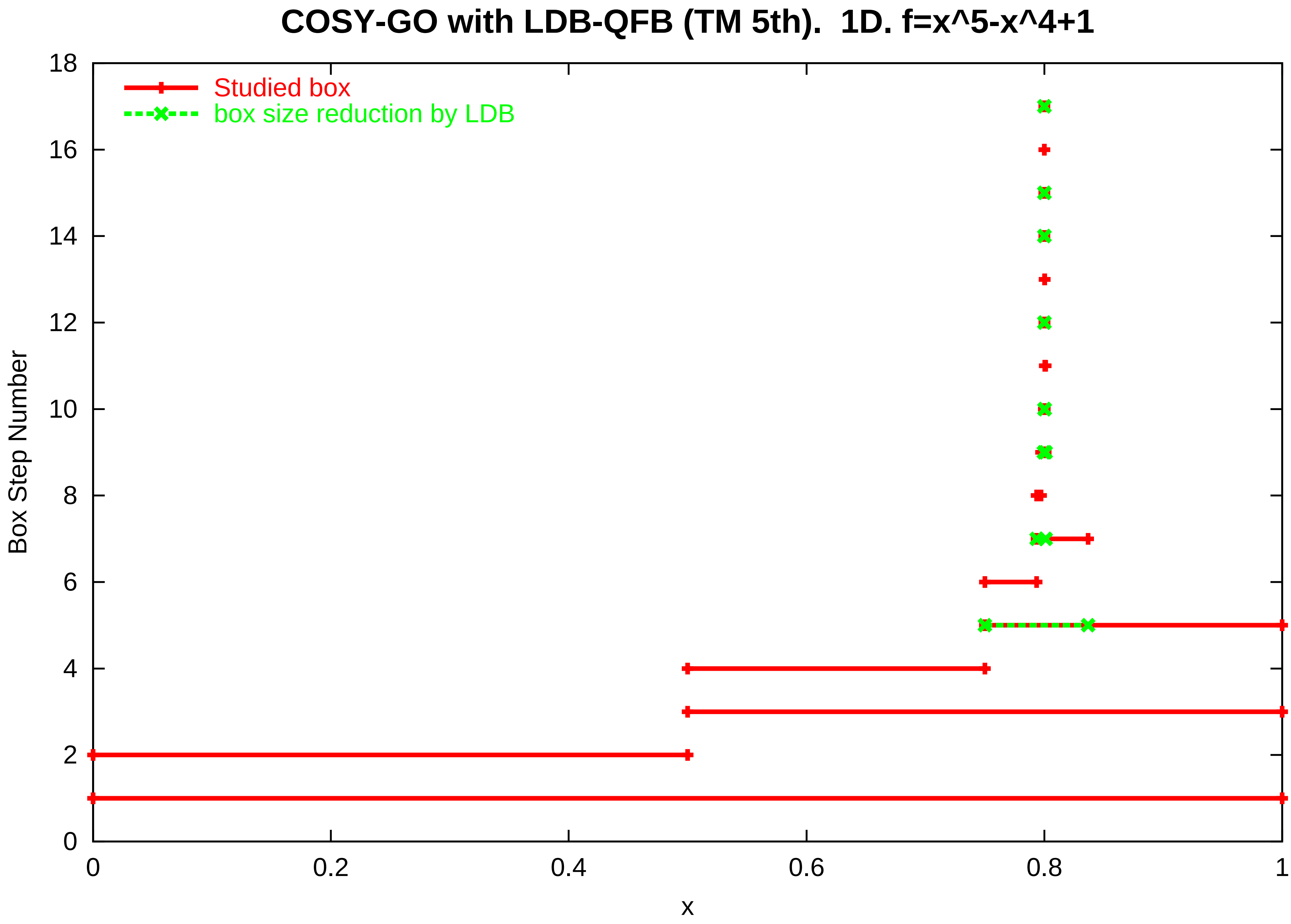
```
 v     N=5^v      NI          NC    NQFB Avg. Iter
 2        25        25           8   1       1.1
 4       625       625         308   1      0.31
 6    15,625    15,625      12,434   1      0.31
 8   390,625   390,625     372,376   1      0.43
10 9,765,625 9,765,625 9,622,750   1      0.55
```

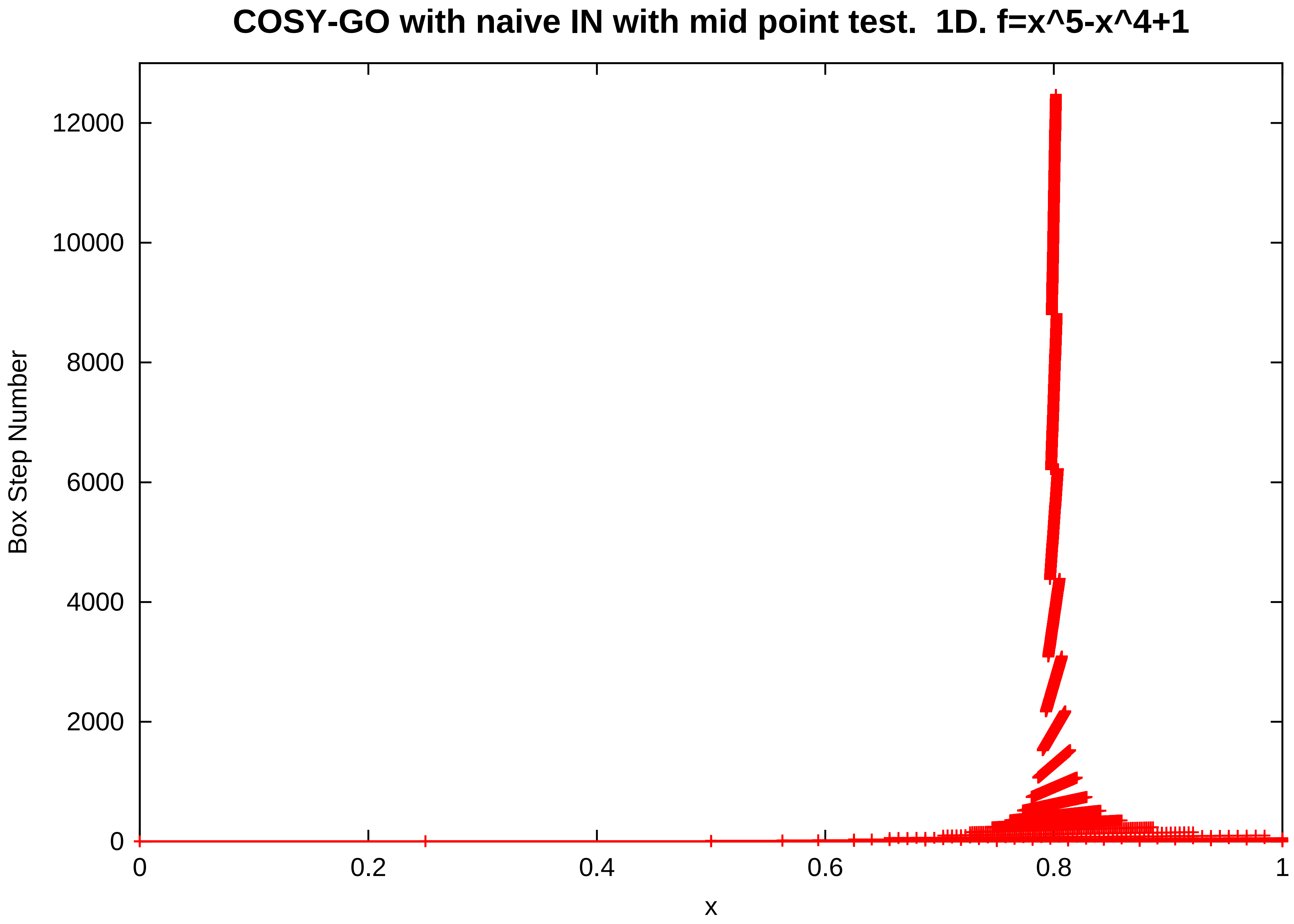
Moore's Simple 1D Function

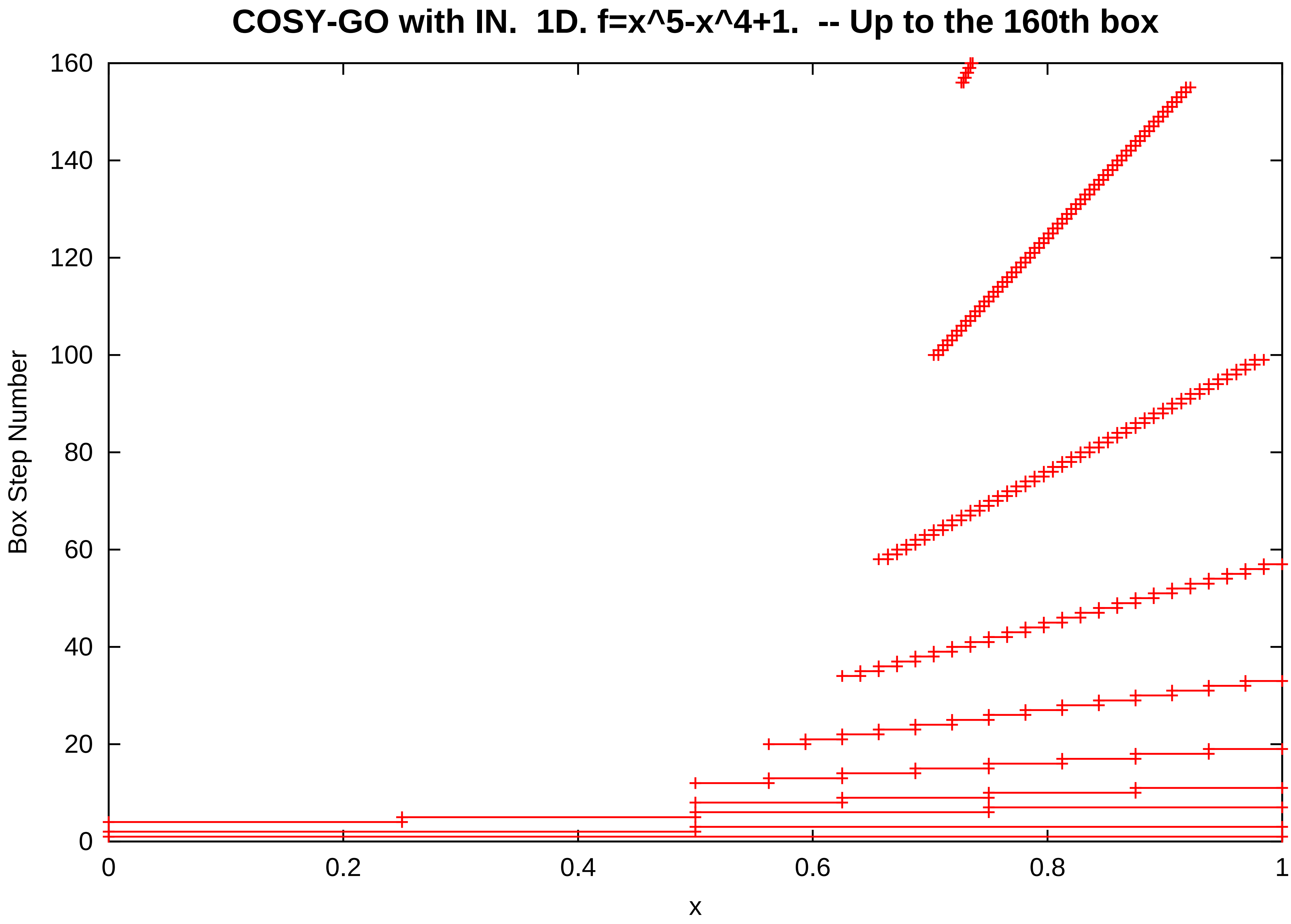
$$f(x) = 1 + x^5 - x^4.$$

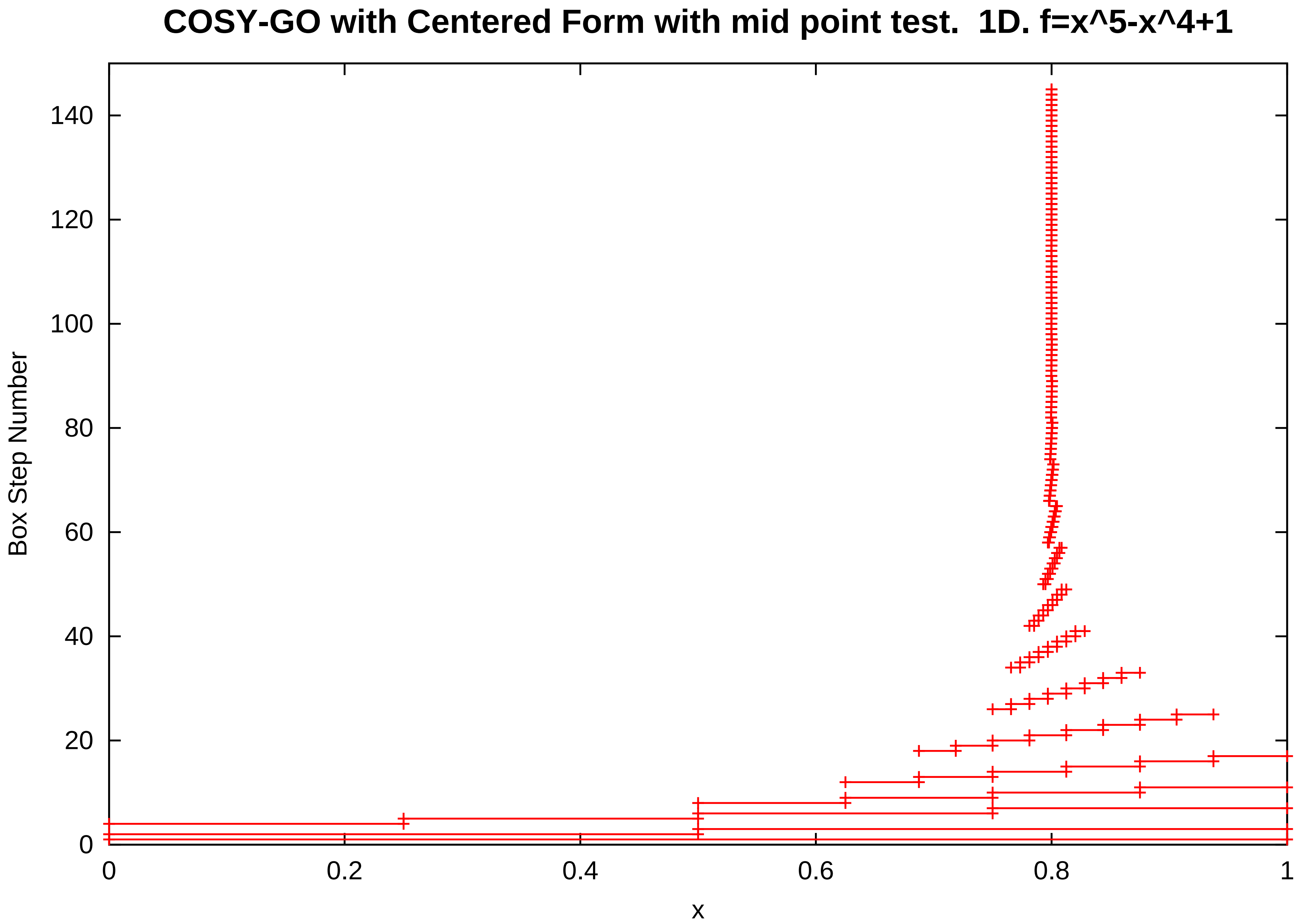
Study on $[0,1]$. Trivial-looking, but dependency and high order. Assumes shallow min at 0.8.




COSY-GO with LDB-QFB (TM 5th). 1D. f=x^5-x^4+1
Studied box
box size reduction by LDB
Box Step Number
x
0
2
4
6
8
10
12
14
16
18
0
0.2
0.4
0.6
0.8
1


COSY-GO with naive IN with mid point test. 1D. f=x^5-x^4+1
Box Step Number
12000
10000
8000
6000
4000
2000
0
0
0.2
0.4
0.6
0.8
1
x


COSY-GO with IN. 1D. f=x^5-x^4+1. -- Up to the 160th box
Box Step Number
160
140
120
100
80
60
40
20
0
0
0.2
0.4
0.6
0.8
1
x


COSY-GO with Centered Form with mid point test. 1D. f=x^5-x^4+1
Box Step Number
0
20
40
60
80
100
120
140
0
0.2
0.4
0.6
0.8
1
x

Beale's 2D and 4D Function

$$f(x_1, x_2) = (1.5 - x_1(1 - x_2))^2 + \left(2.25 - x_1(1 - x_2^2)\right)^2 + \left(2.625 - x_1(1 - x_2^3)\right)^2$$

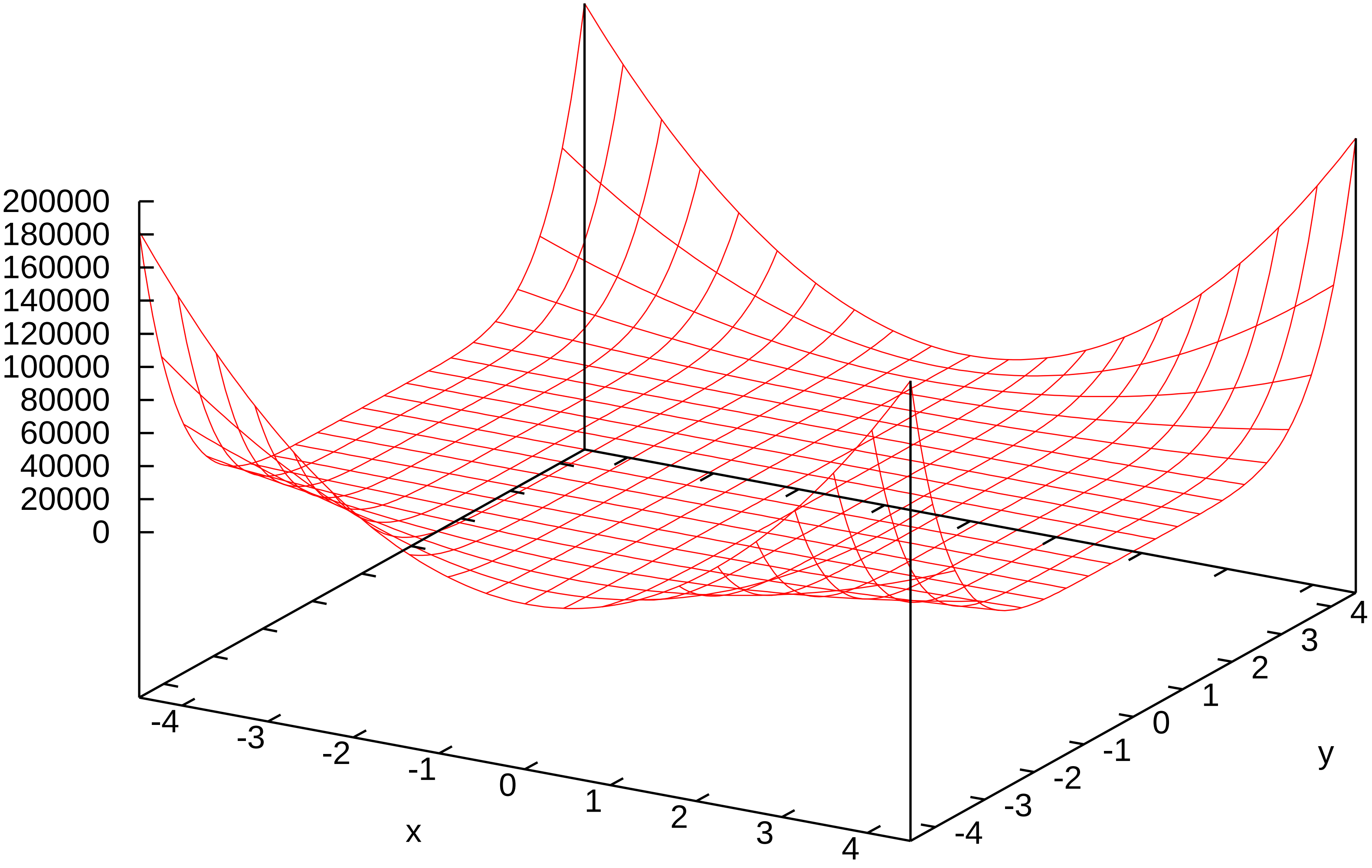
Domain $[-4.5, 4.5]^2$. Minimum value 0 at $(3, 0.5)$.

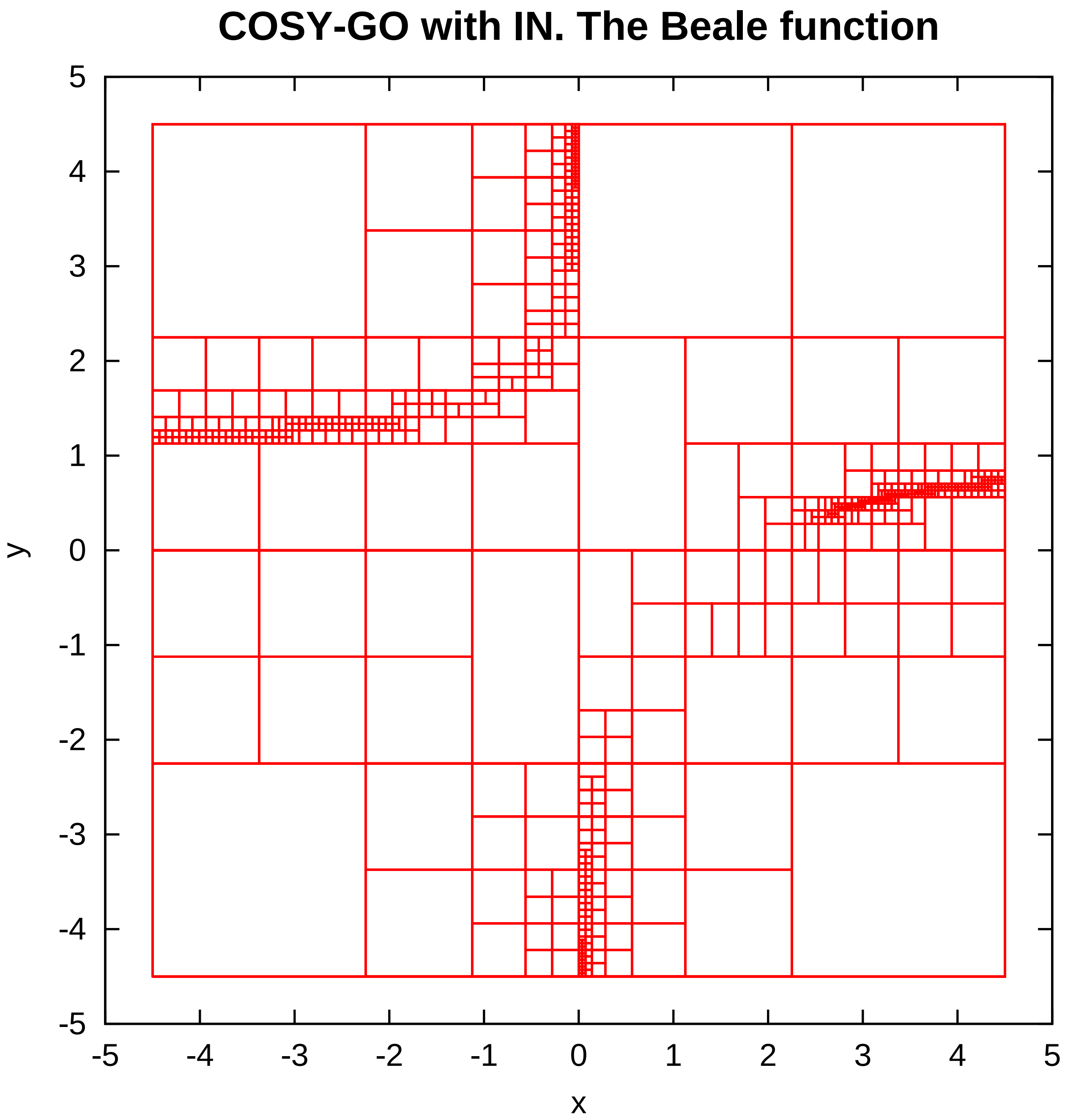
Little dependency, but tricky very shallow behavior.
Generalization to 4D:

$$\begin{aligned} f(x_1, x_2, x_3, x_4) = {} & (1.5 - x_1(1 - x_2))^2 + \left(2.25 - x_1(1 - x_2^2)\right)^2 + \left(2.625 - x_1(1 - x_2^3)\right) \\ & + (1 + x_3(1 - x_4))^2 + \left(3 + x_3(1 - x_4^2)\right)^2 + \left(7 + x_3(1 - x_4^3)\right)^2 \\ & + (3 + x_1(1 - x_4))^2 + \left(9 + x_1(1 - x_4^2)\right)^2 + \left(21 + x_1(1 - x_4^3)\right)^2 \\ & + (0.5 - x_3(1 - x_2))^2 + \left(0.75 - x_3(1 - x_2^2)\right)^2 + \left(0.875 - x_3(1 - x_2^3)\right)^2 \end{aligned}$$

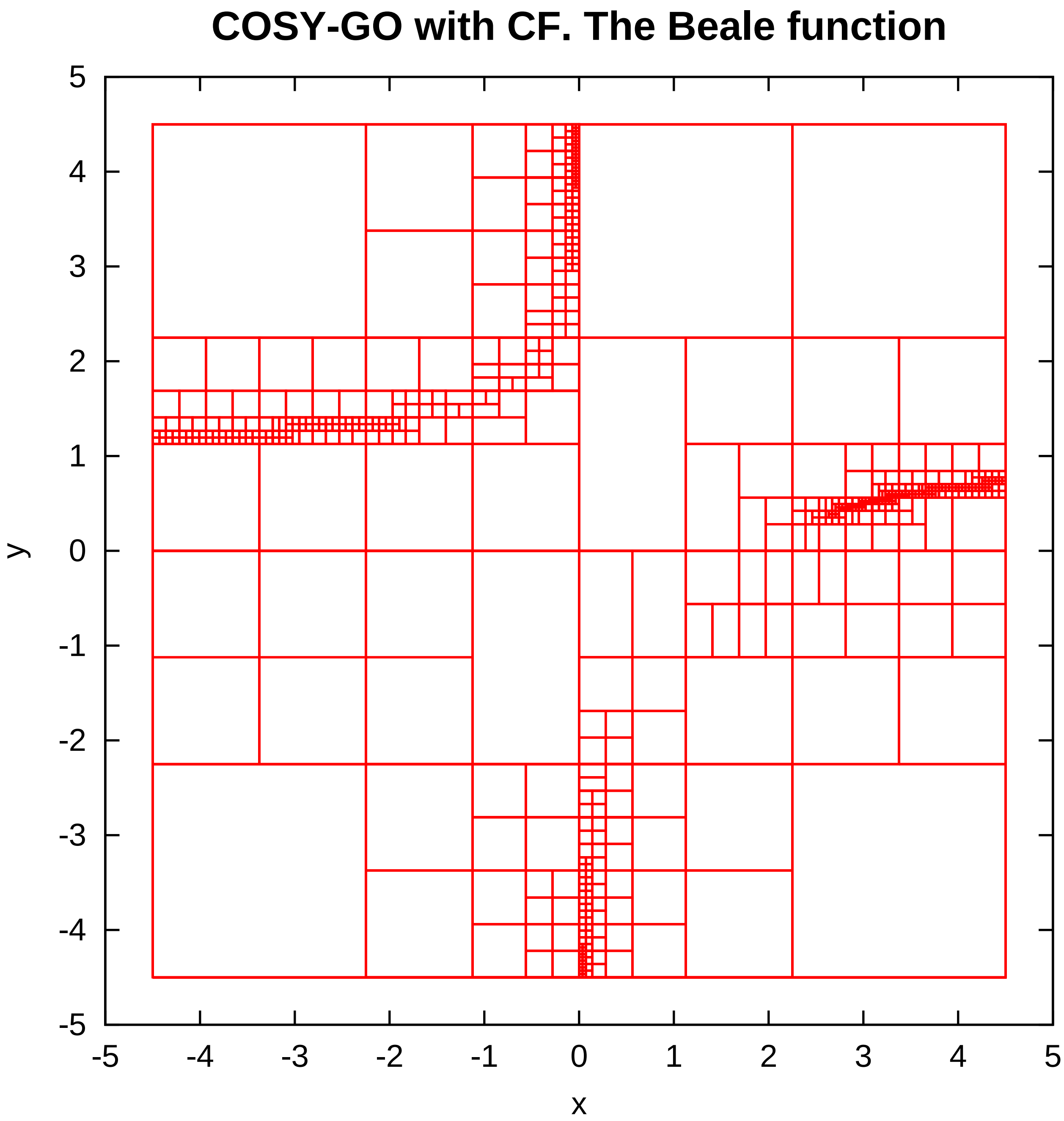
Domain $[0, 4]^4$. Minimum value 0 at $(3, 0.5, 1, 2)$

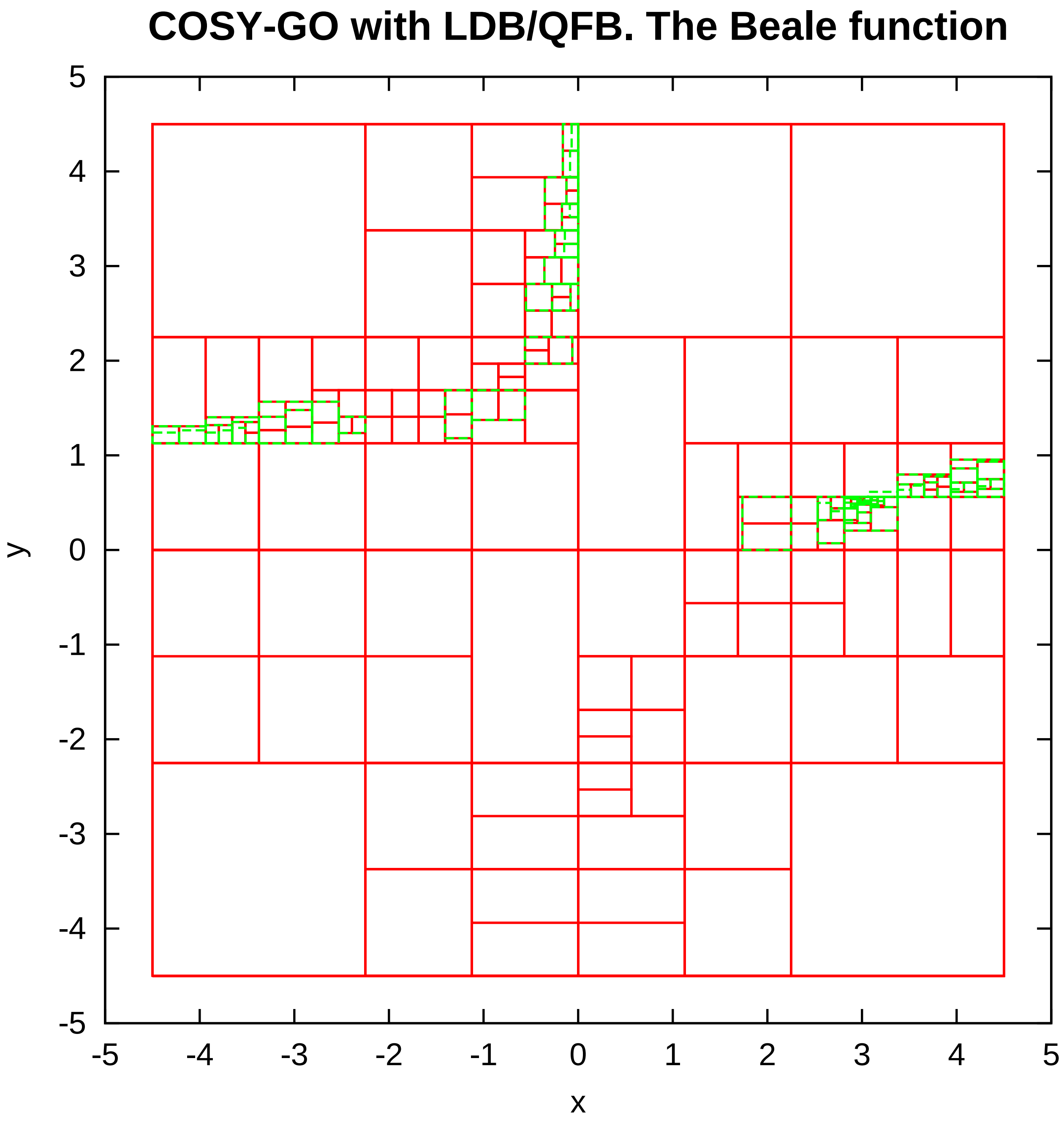
The Beale function. f = [1.5-x(1-y)]^2 + [2.25-x(1-y^2)]^2 + [2.625-x(1-y^3)]^2

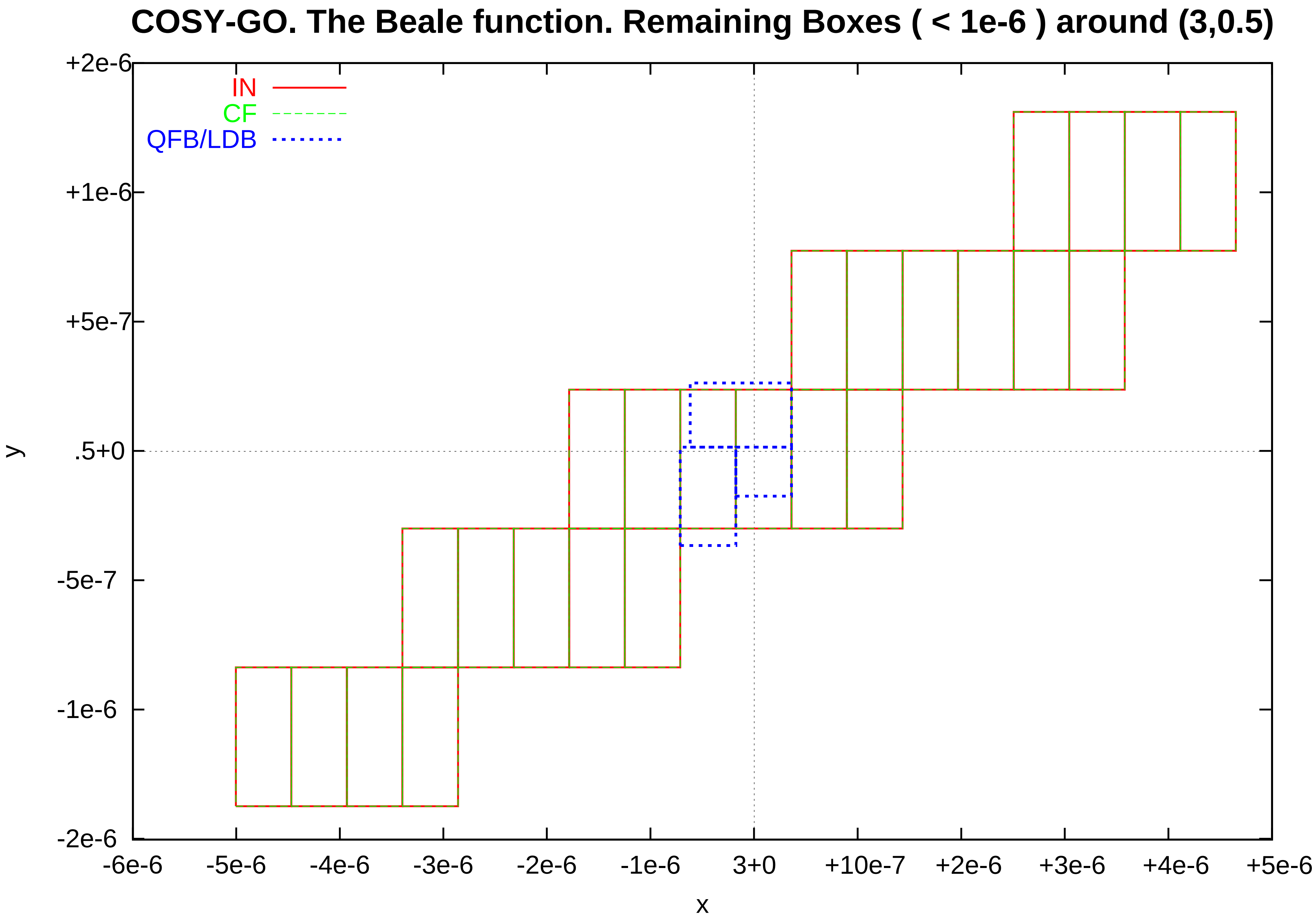




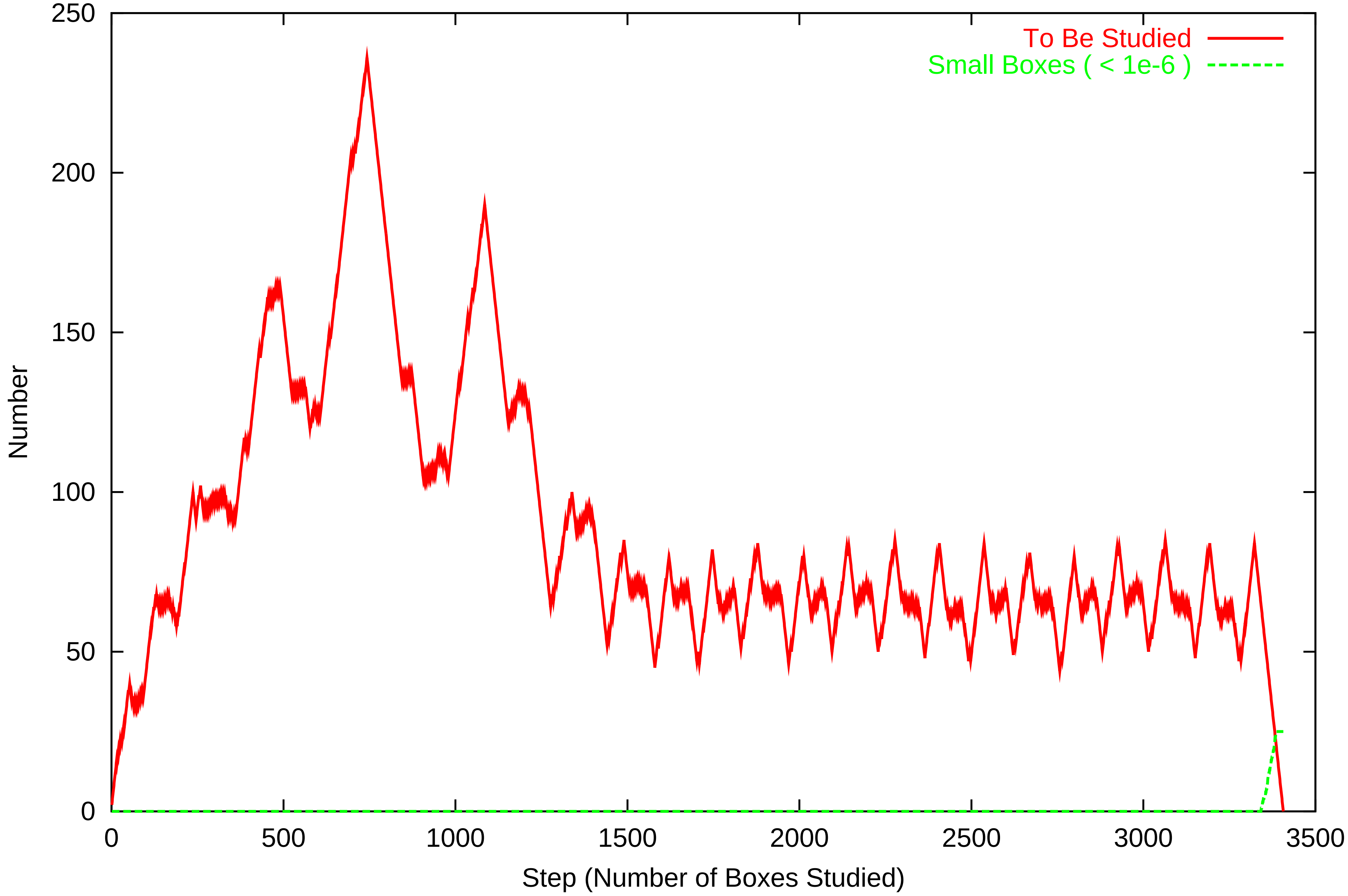
COSY-GO with IN. The Beale function

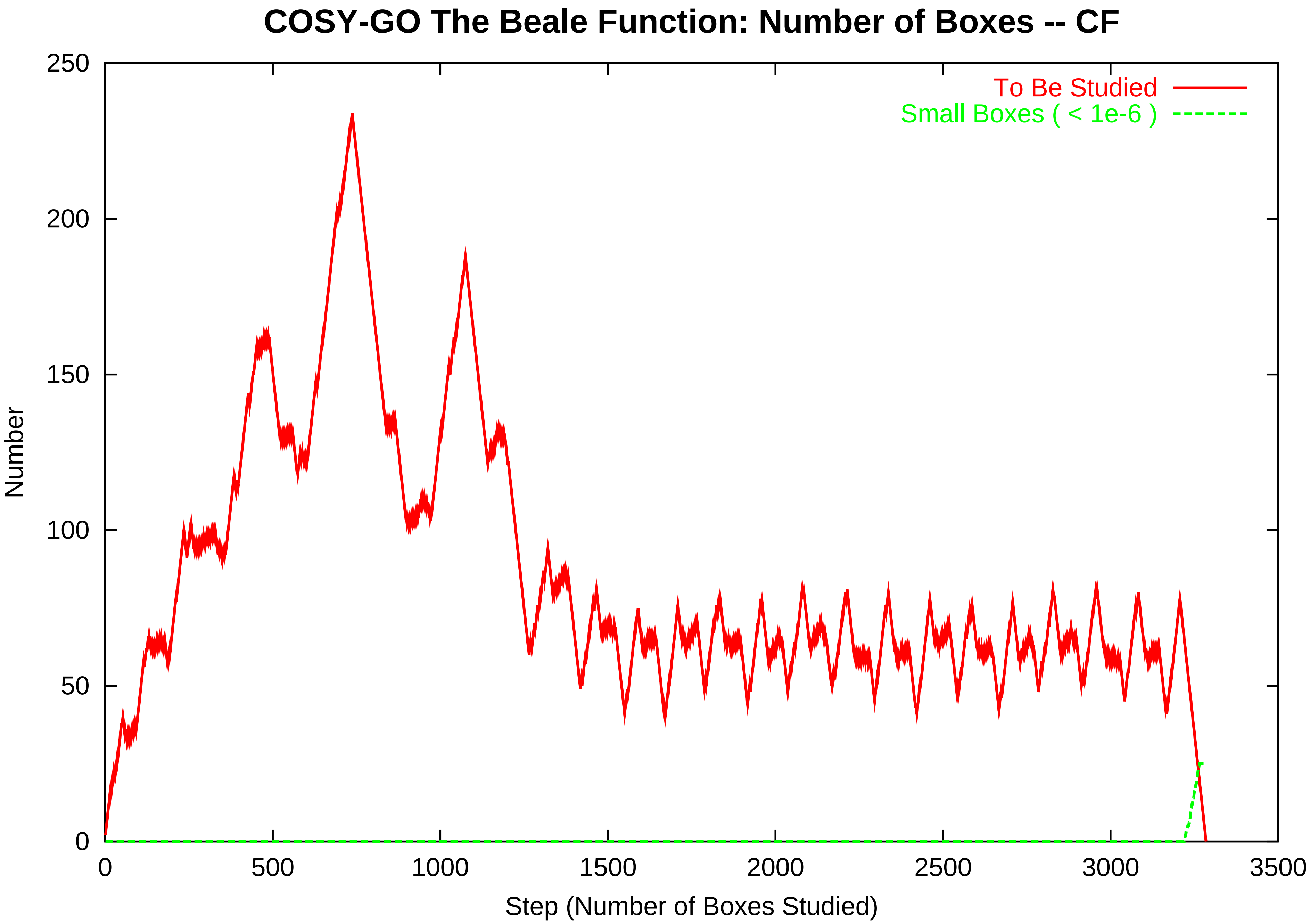

COSY-GO with CF. The Beale function
5
4
3
2
1
0
-1
-2
-3
-4
-5
y
-5
-4
-3
-2
-1
0
1
2
3
4
5
x

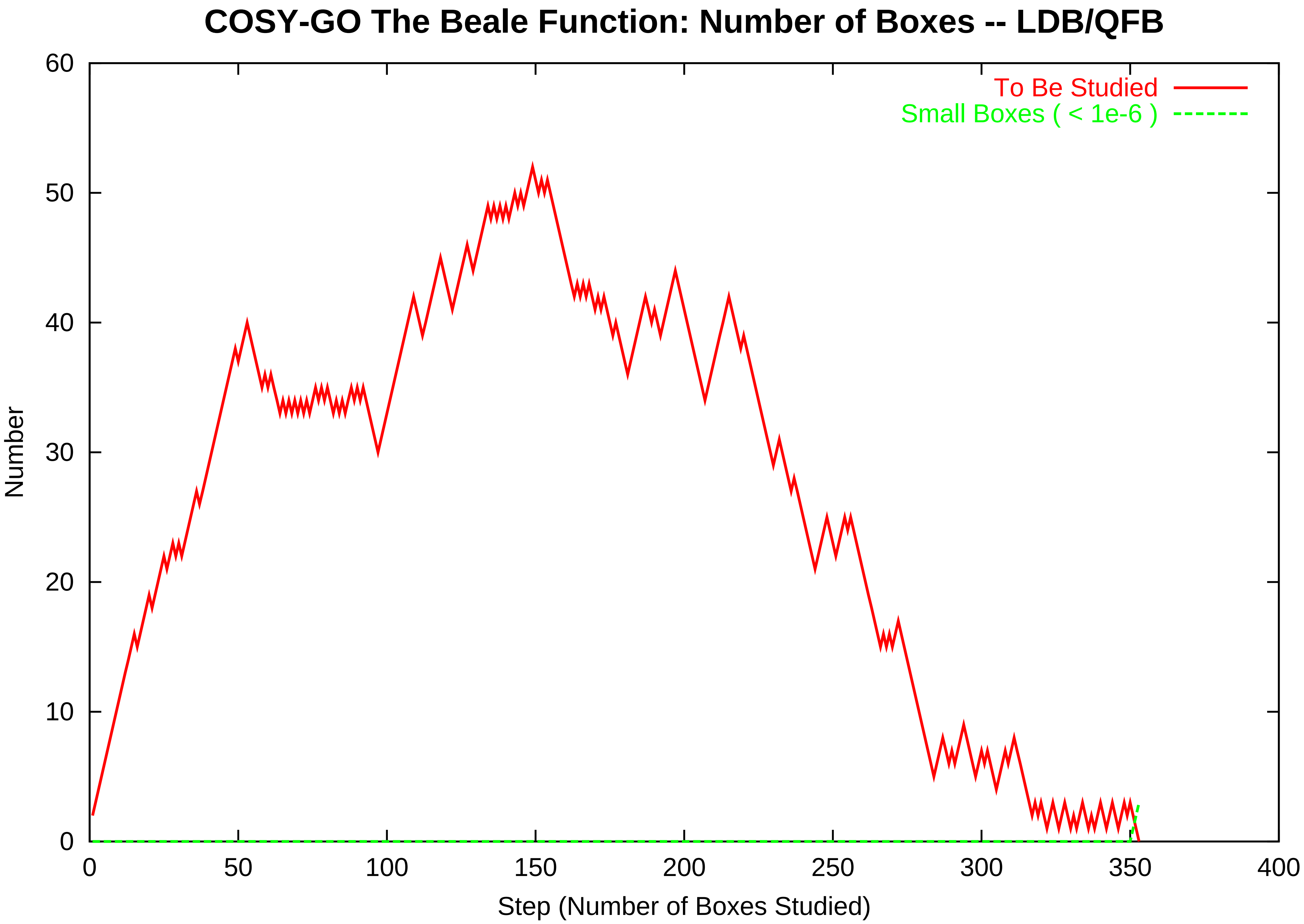

COSY-GO with LDB/QFB. The Beale function
y
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
-4
-3
-2
-1
0
1
2
3
4
5
x

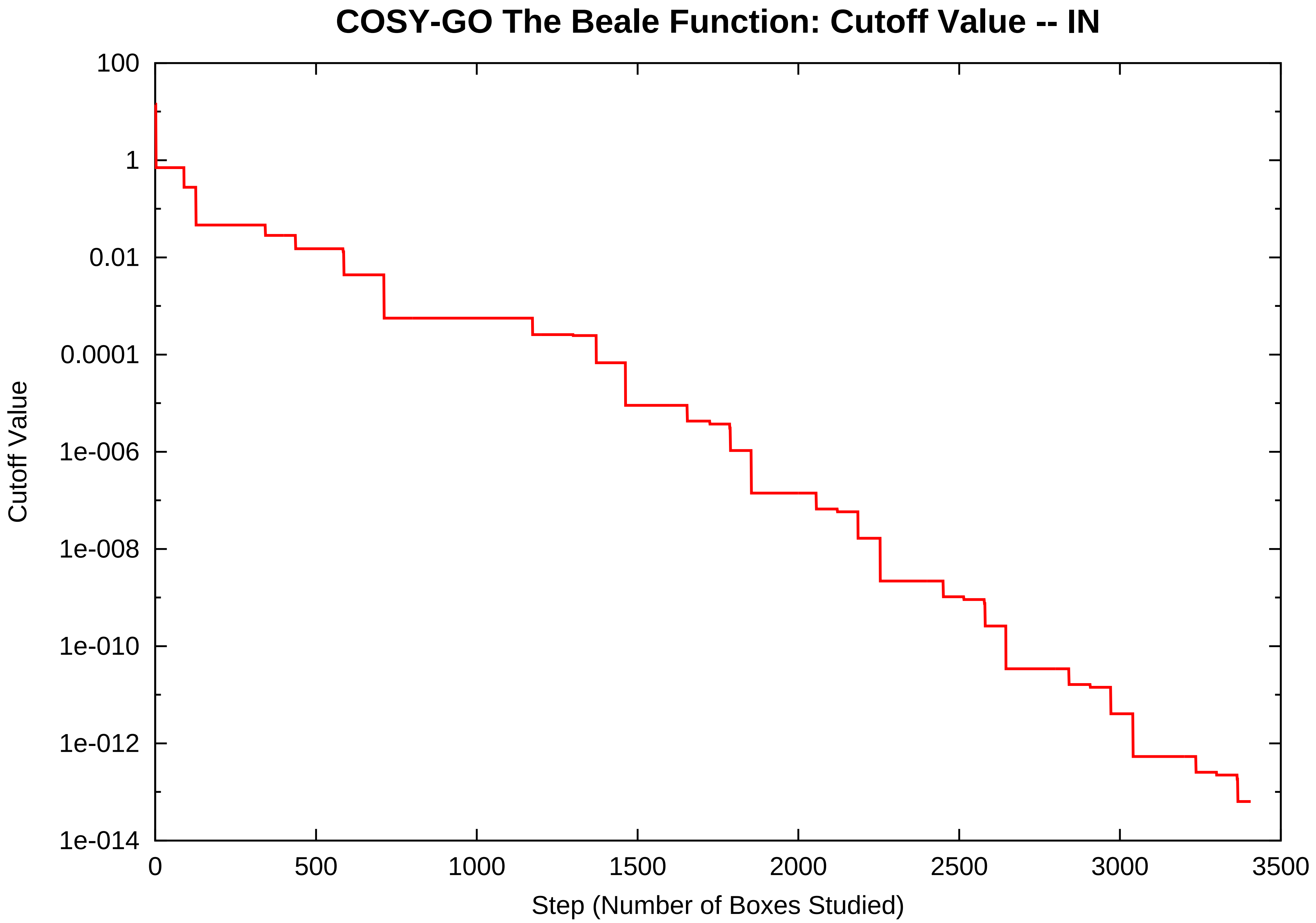

COSY-GO. The Beale function. Remaining Boxes (< 1e-6) around (3,0.5)
IN
CF
QFB/LDB
+2e-6
+1e-6
+5e-7
.5+0
-5e-7
-1e-6
-2e-6
y
-6e-6
-5e-6
-4e-6
-3e-6
-2e-6
-1e-6
3+0
+10e-7
+2e-6
+3e-6
+4e-6
+5e-6
x

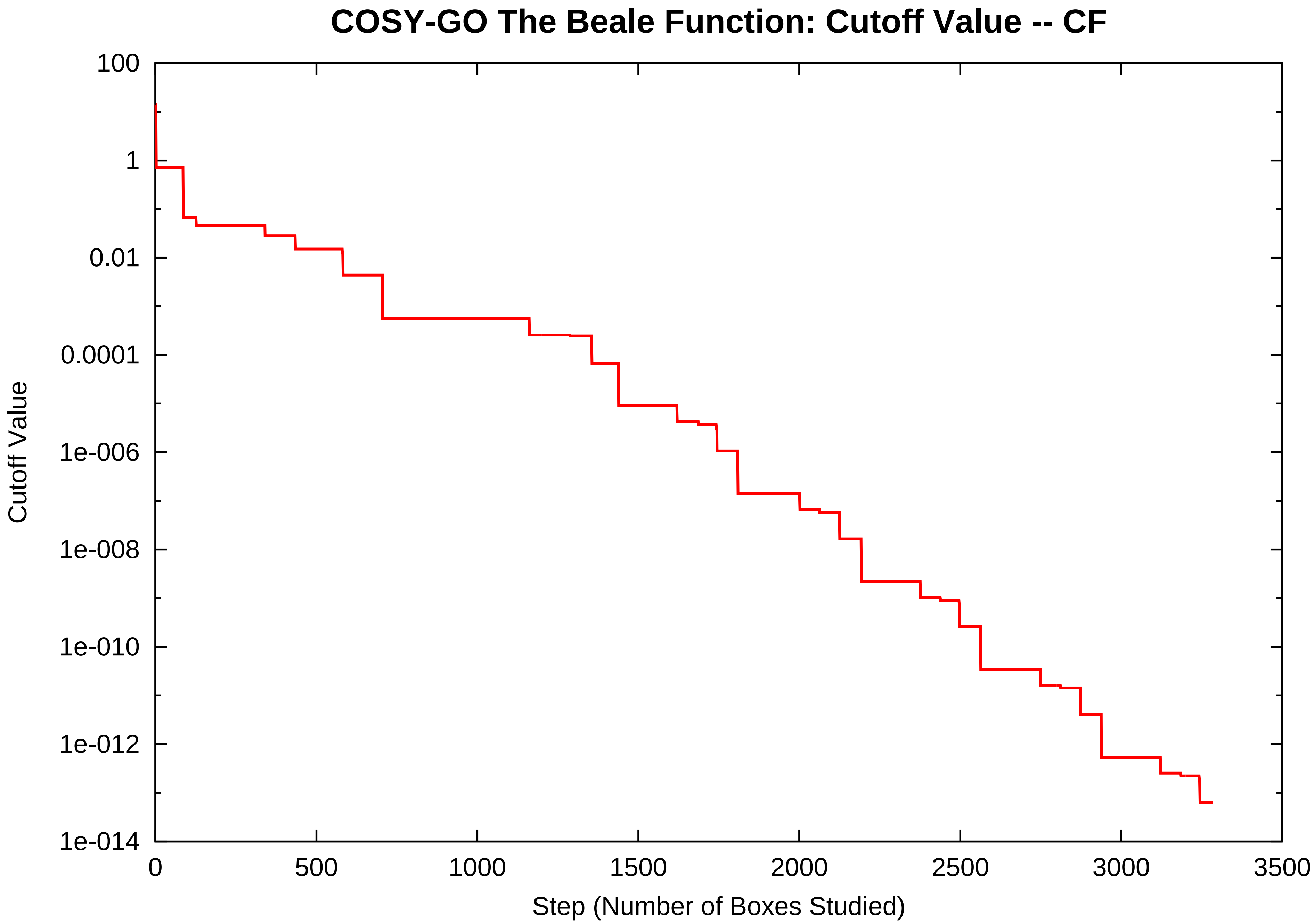
COSY-GO The Beale Function: Number of Boxes -- IN

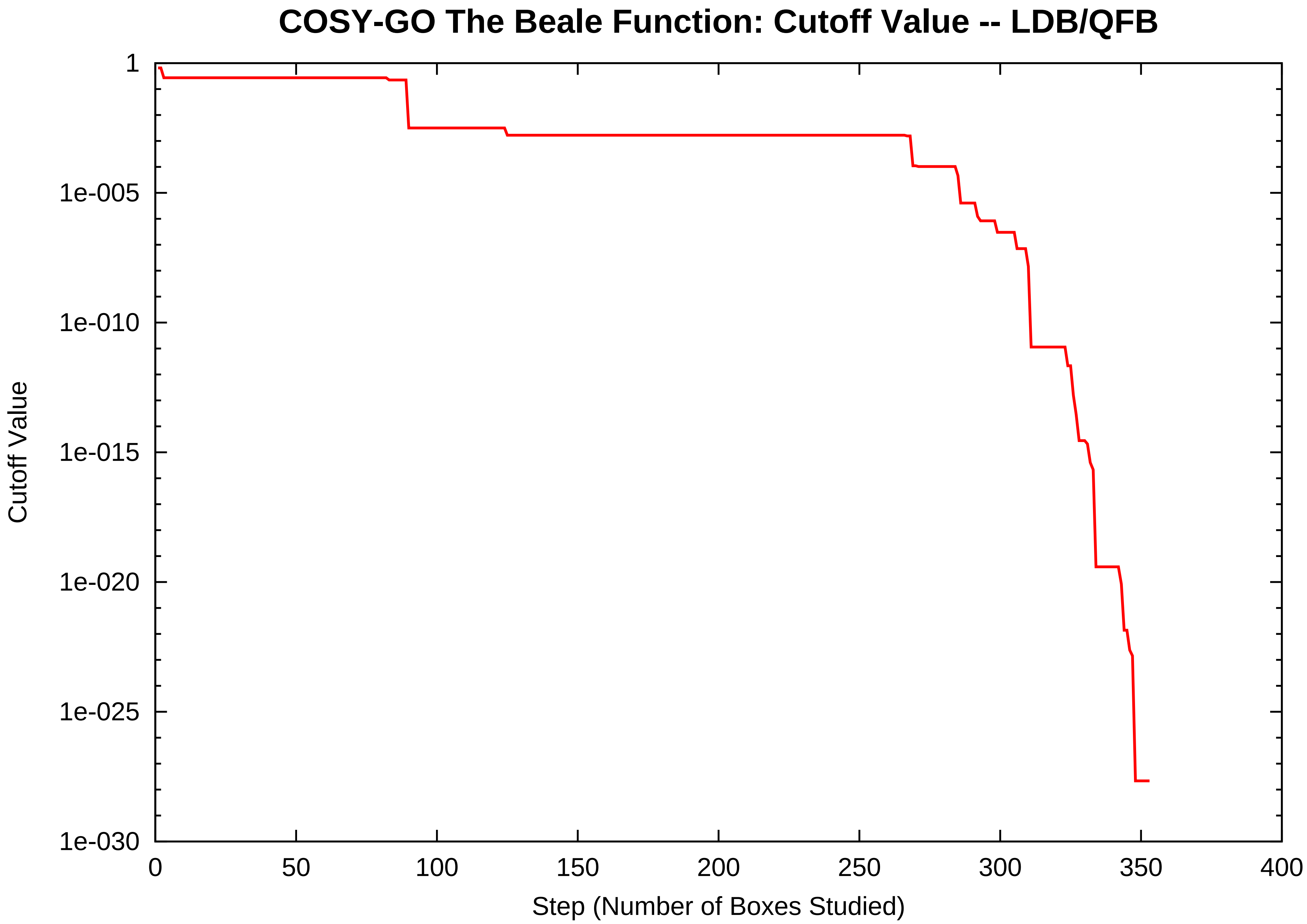


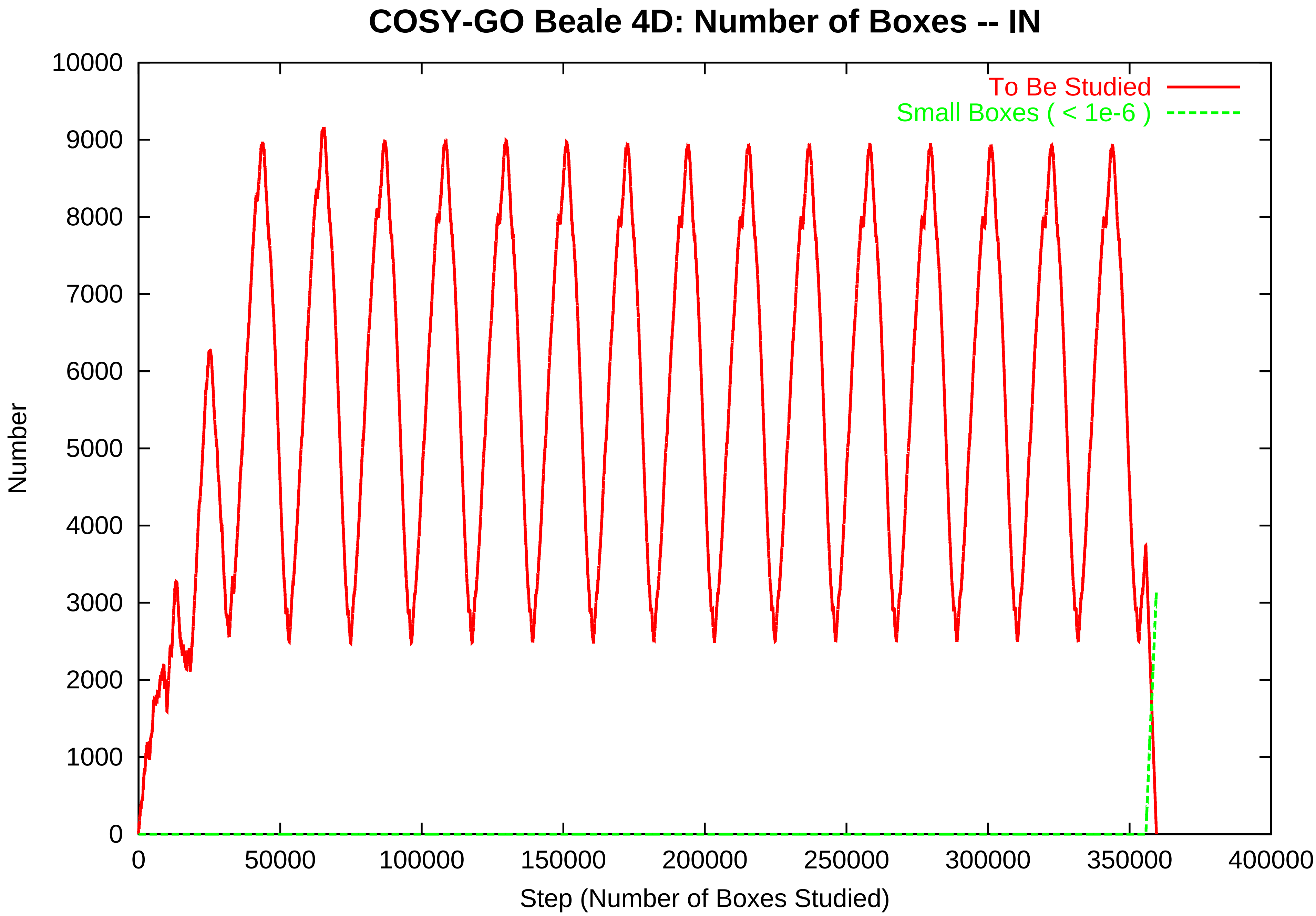

COSY-GO The Beale Function: Number of Boxes -- CF
To Be Studied
Small Boxes (< 1e-6)
Number
Step (Number of Boxes Studied)
0
50
100
150
200
250
0
500
1000
1500
2000
2500
3000
3500

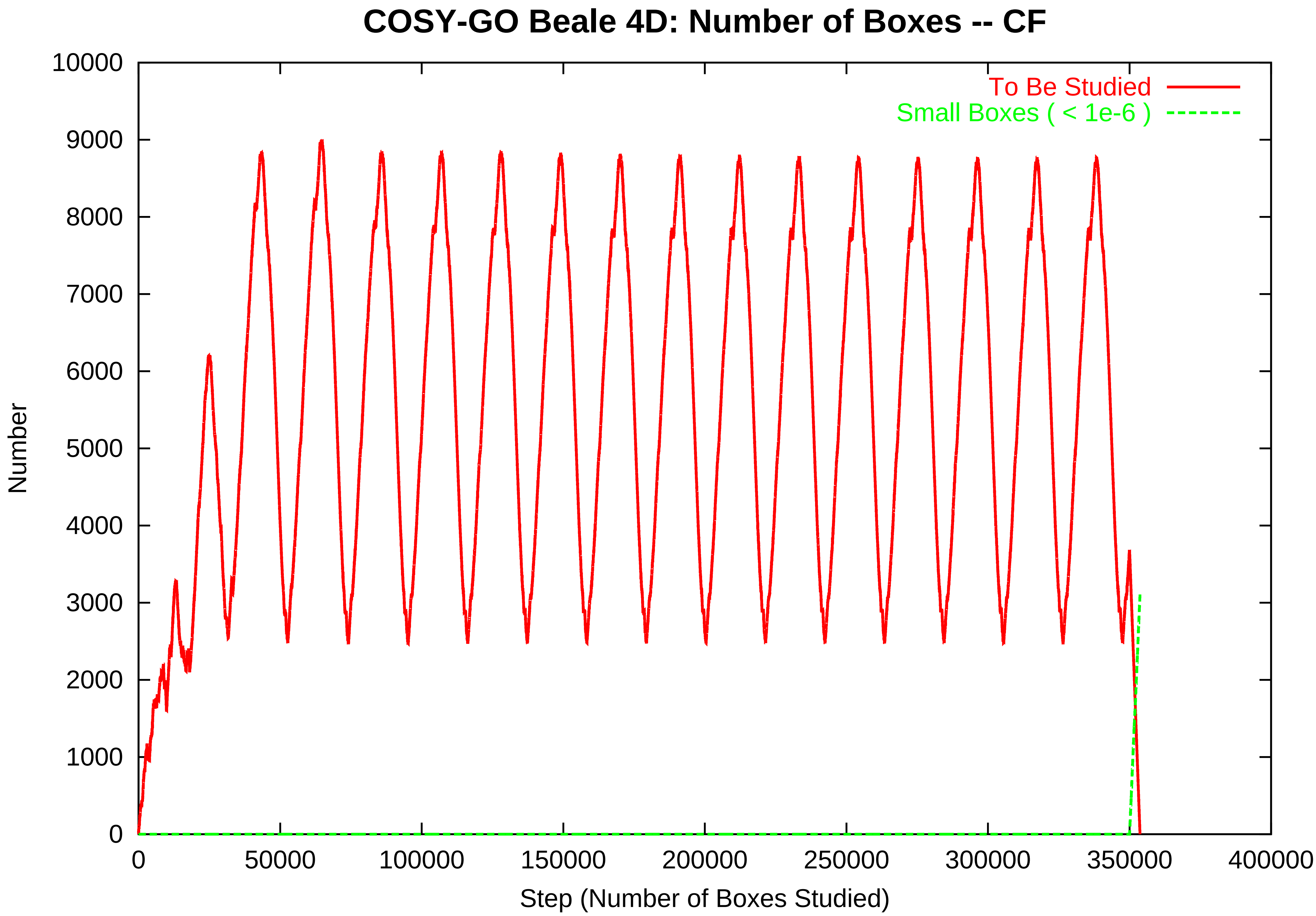

COSY-GO The Beale Function: Number of Boxes -- LDB/QFB
To Be Studied
Small Boxes (< 1e-6)
Number
Step (Number of Boxes Studied)
0
10
20
30
40
50
60
0
50
100
150
200
250
300
350
400

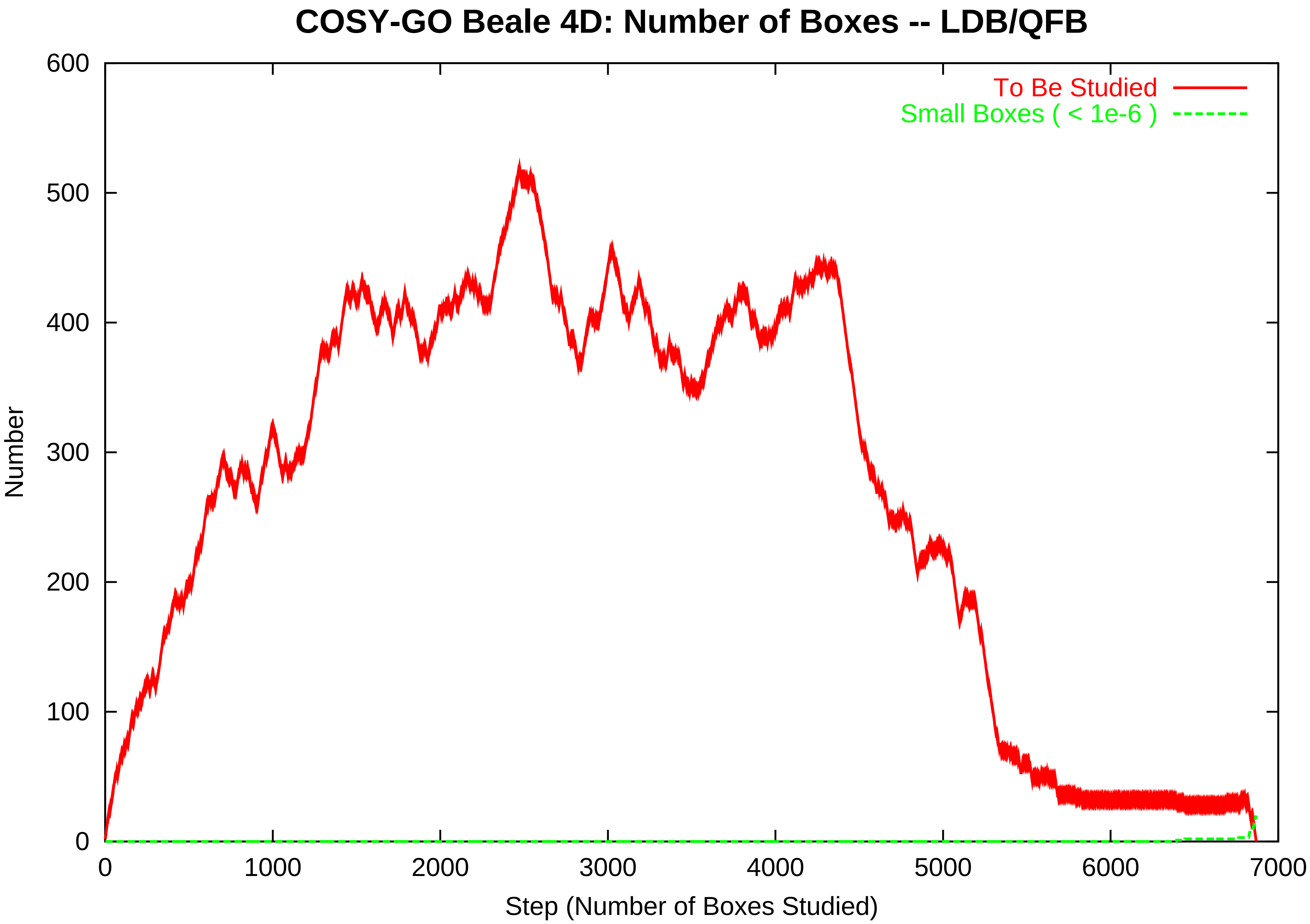

COSY-GO The Beale Function: Cutoff Value -- IN
Cutoff Value
100
1
0.01
0.0001
1e-006
1e-008
1e-010
1e-012
1e-014
0
500
1000
1500
2000
2500
3000
3500
Step (Number of Boxes Studied)

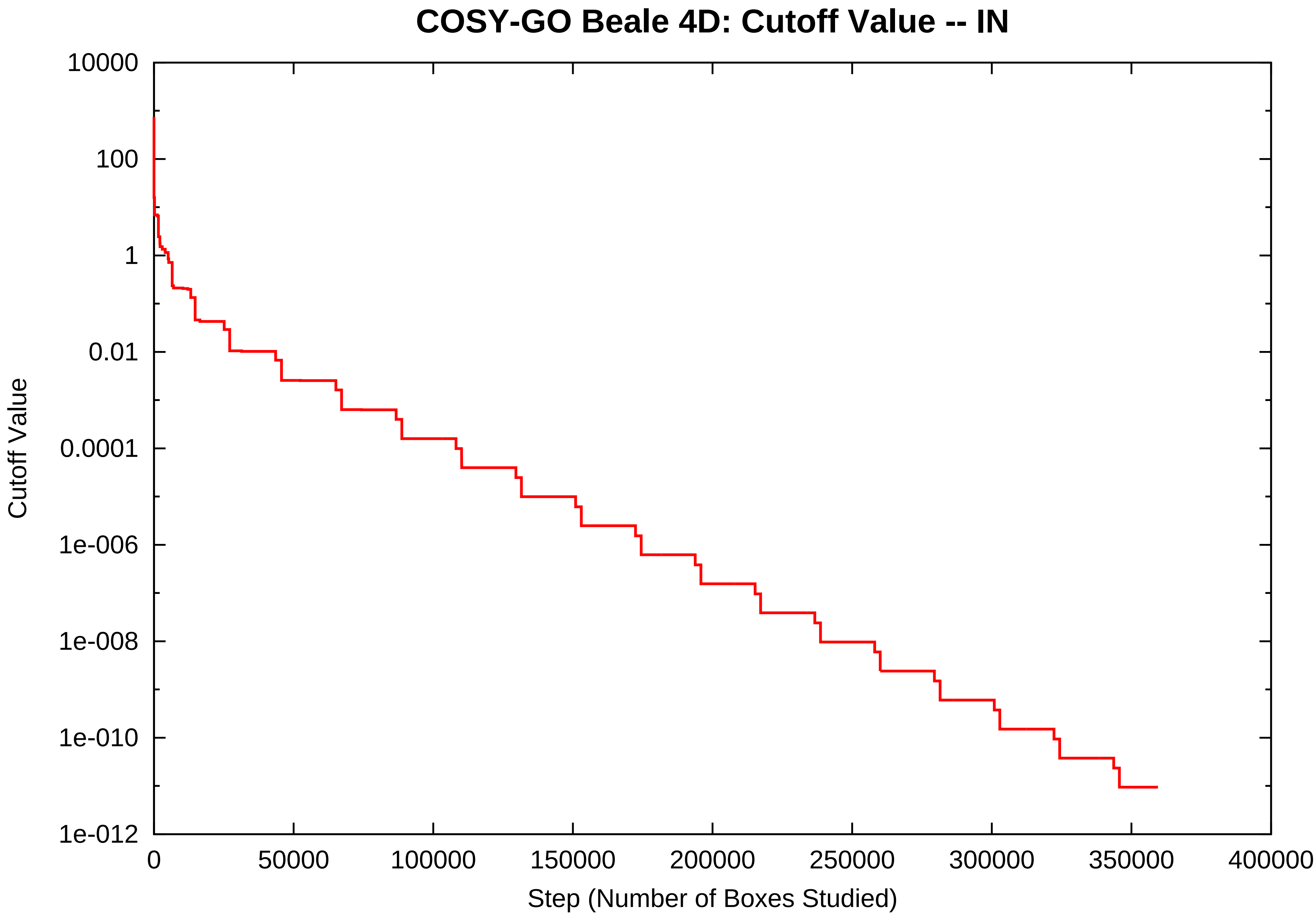

COSY-GO The Beale Function: Cutoff Value -- CF
Cutoff Value
100
1
0.01
0.0001
1e-006
1e-008
1e-010
1e-012
1e-014
0
500
1000
1500
2000
2500
3000
3500
Step (Number of Boxes Studied)

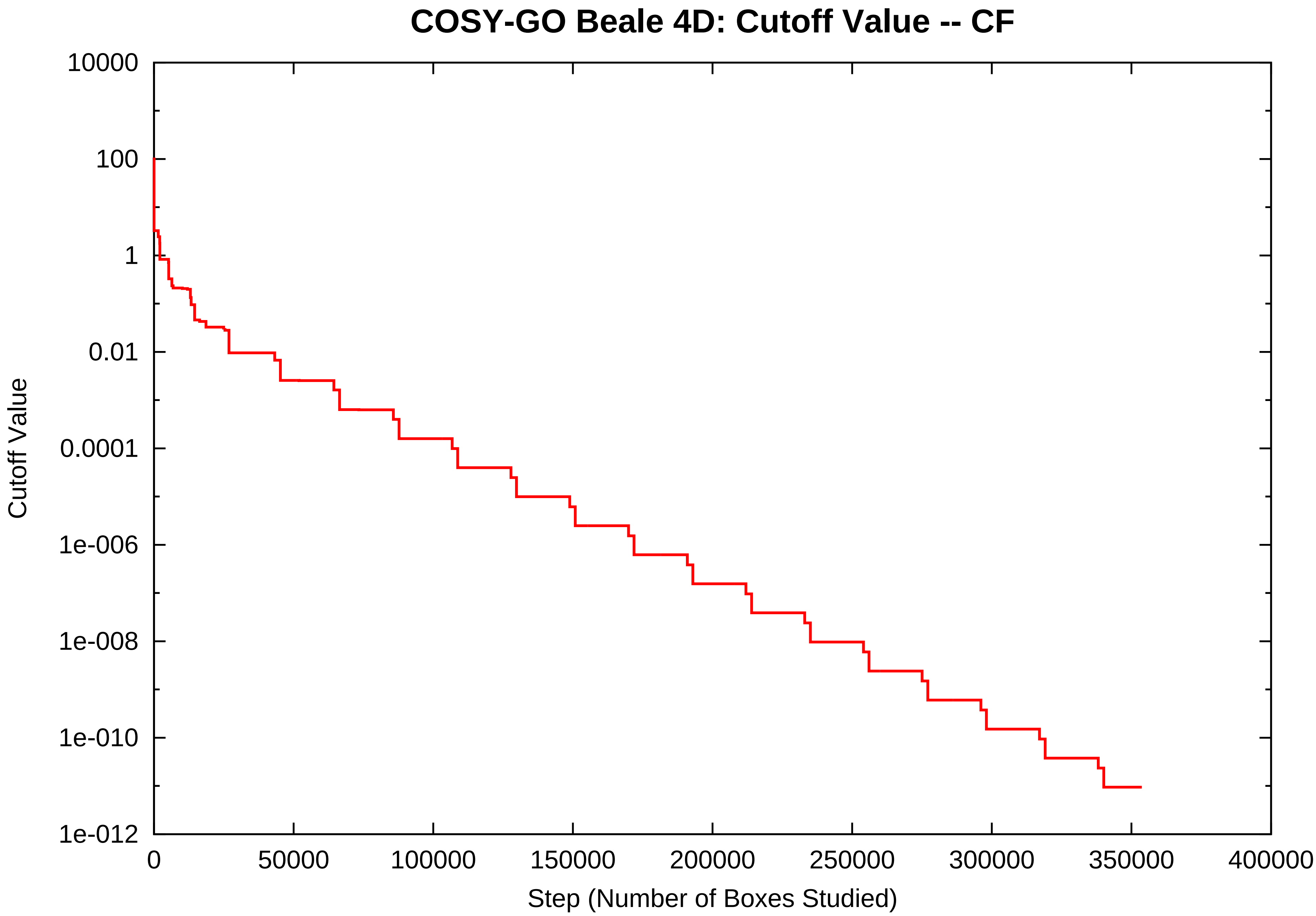

COSY-GO The Beale Function: Cutoff Value -- LDB/QFB
Cutoff Value
1
1e-005
1e-010
1e-015
1e-020
1e-025
1e-030
0
50
100
150
200
250
300
350
400
Step (Number of Boxes Studied)

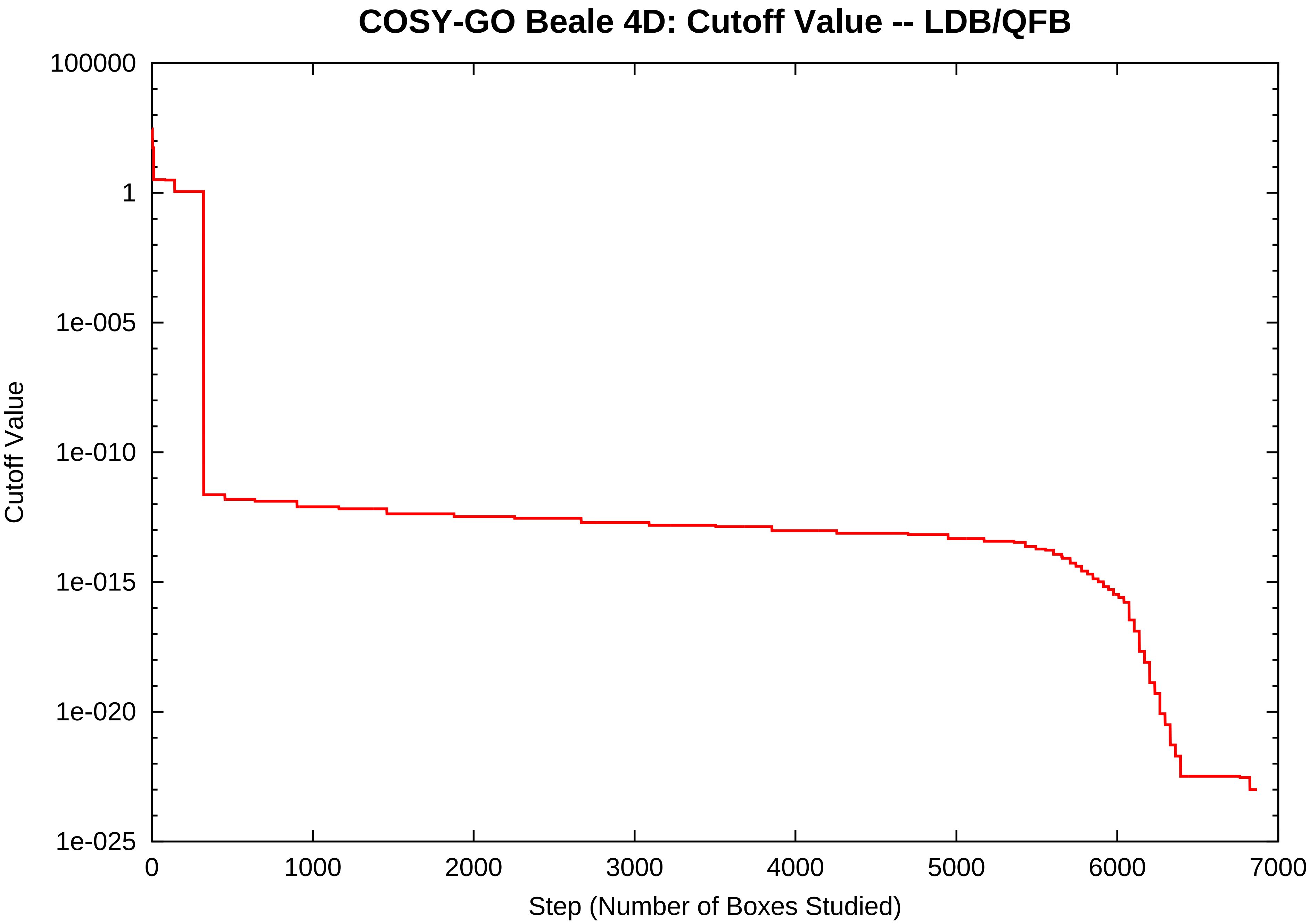

COSY-GO Beale 4D: Number of Boxes -- IN
To Be Studied
Small Boxes (< 1e-6)
Number
Step (Number of Boxes Studied)
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
0
50000
100000
150000
200000
250000
300000
350000
400000


COSY-GO Beale 4D: Number of Boxes -- CF
To Be Studied
Small Boxes (< 1e-6)
Number
Step (Number of Boxes Studied)
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
0
50000
100000
150000
200000
250000
300000
350000
400000


COSY-GO Beale 4D: Number of Boxes -- LDB/QFB
To Be Studied
Small Boxes (< 1e-6)
Number
Step (Number of Boxes Studied)
0
100
200
300
400
500
600
0
1000
2000
3000
4000
5000
6000
7000


COSY-GO Beale 4D: Cutoff Value -- IN
Cutoff Value
10000
100
1
0.01
0.0001
1e-006
1e-008
1e-010
1e-012
0
50000
100000
150000
200000
250000
300000
350000
400000
Step (Number of Boxes Studied)


COSY-GO Beale 4D: Cutoff Value -- CF
Cutoff Value
10000
100
1
0.01
0.0001
1e-006
1e-008
1e-010
1e-012
0
50000
100000
150000
200000
250000
300000
350000
400000
Step (Number of Boxes Studied)

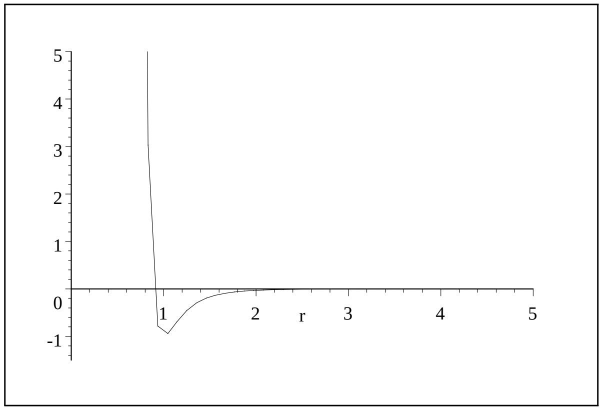


COSY-GO Beale 4D: Cutoff Value -- LDB/QFB

Lennard-Jones Potentials

Ensemble of n particles interacting pointwise with potentials

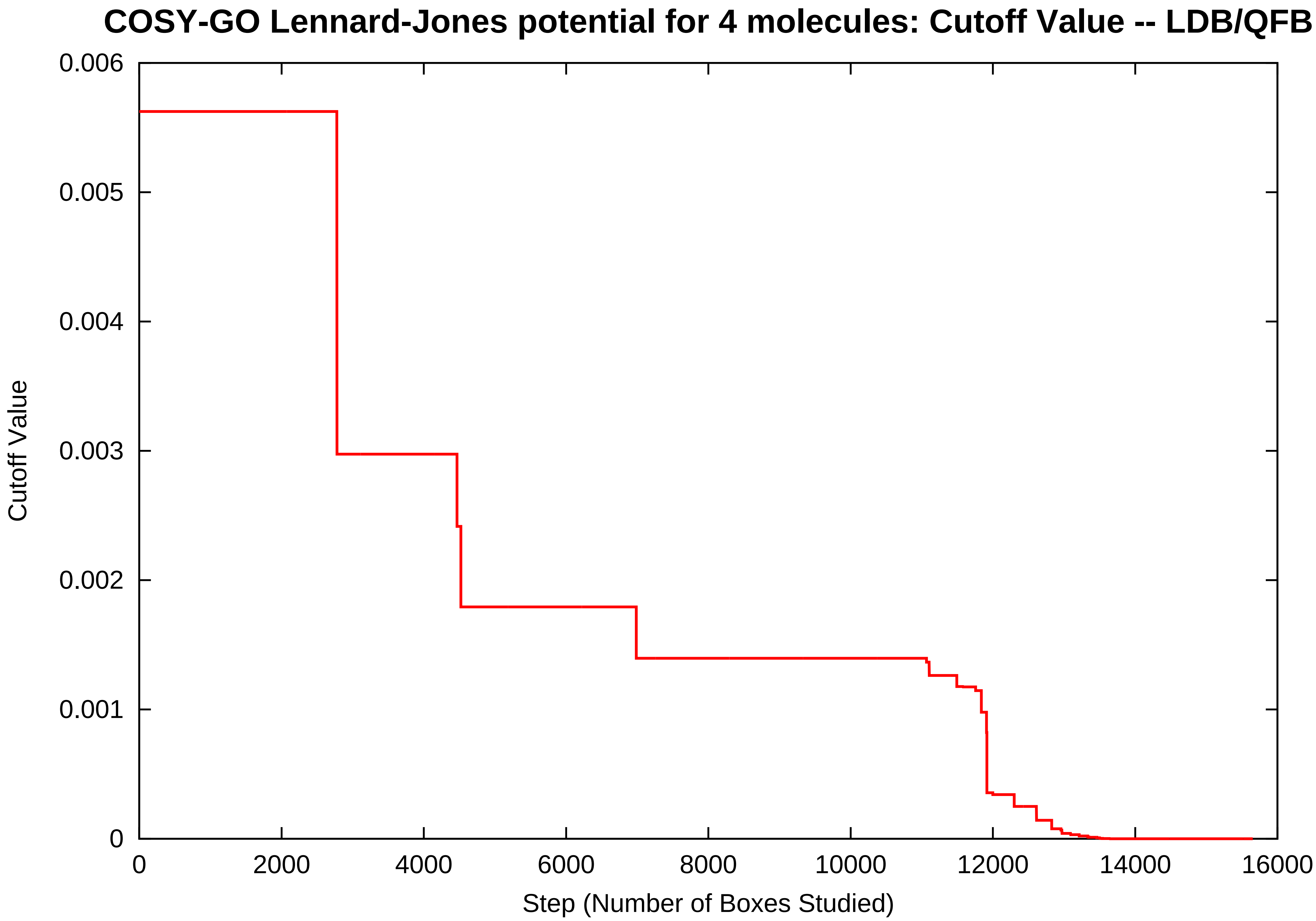
$$V_{LJ}(r) = \frac{1}{r^{12}} - 2 \cdot \frac{1}{r^6}$$



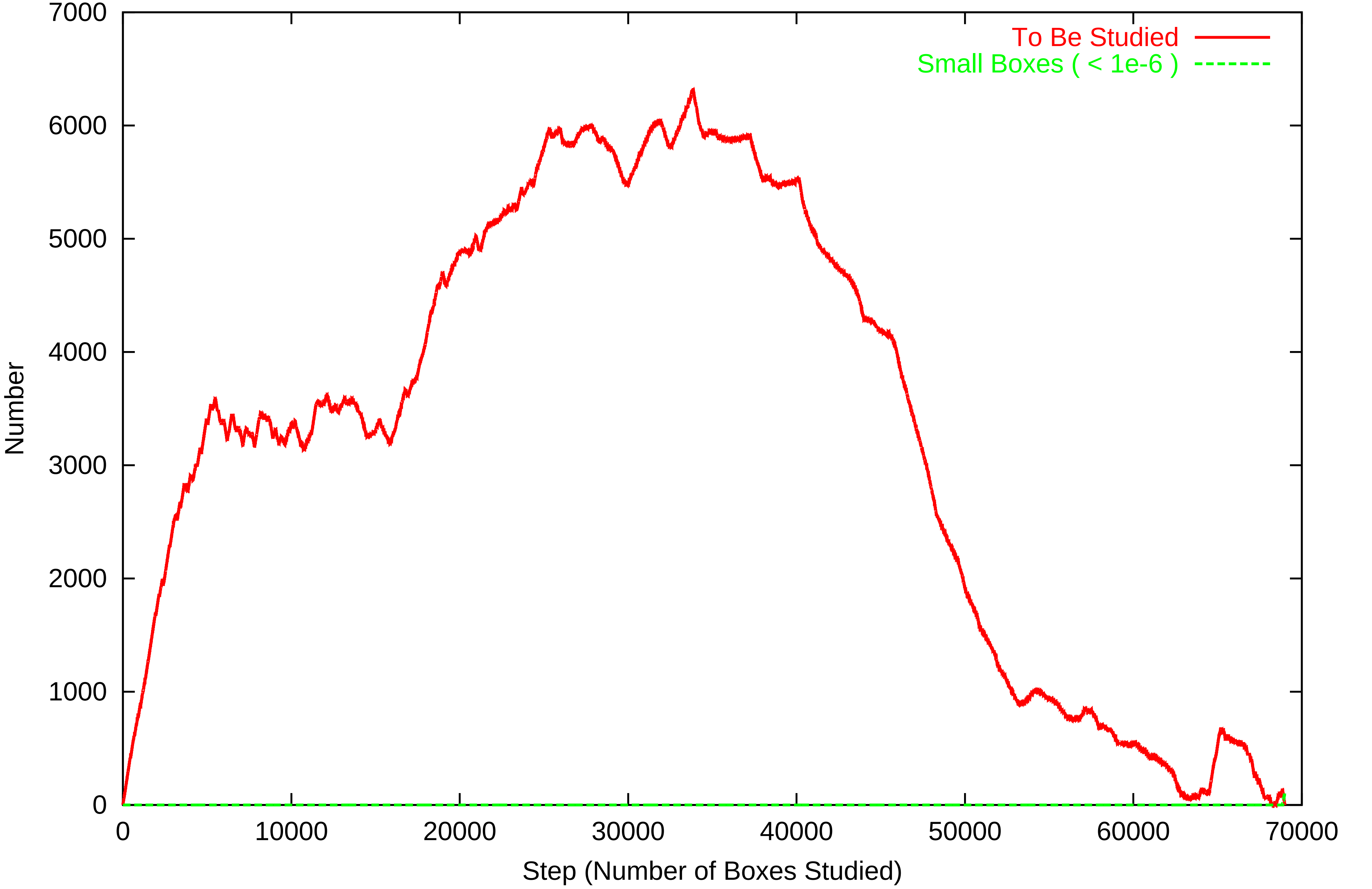
Has very shallow minimum of -1 at $r = 0$. Very hard to Taylor expand. Extremely wide range of function values: $V_{LJ}(0.5) \approx 4000$, $V_{LJ}(2) \approx 0.03$

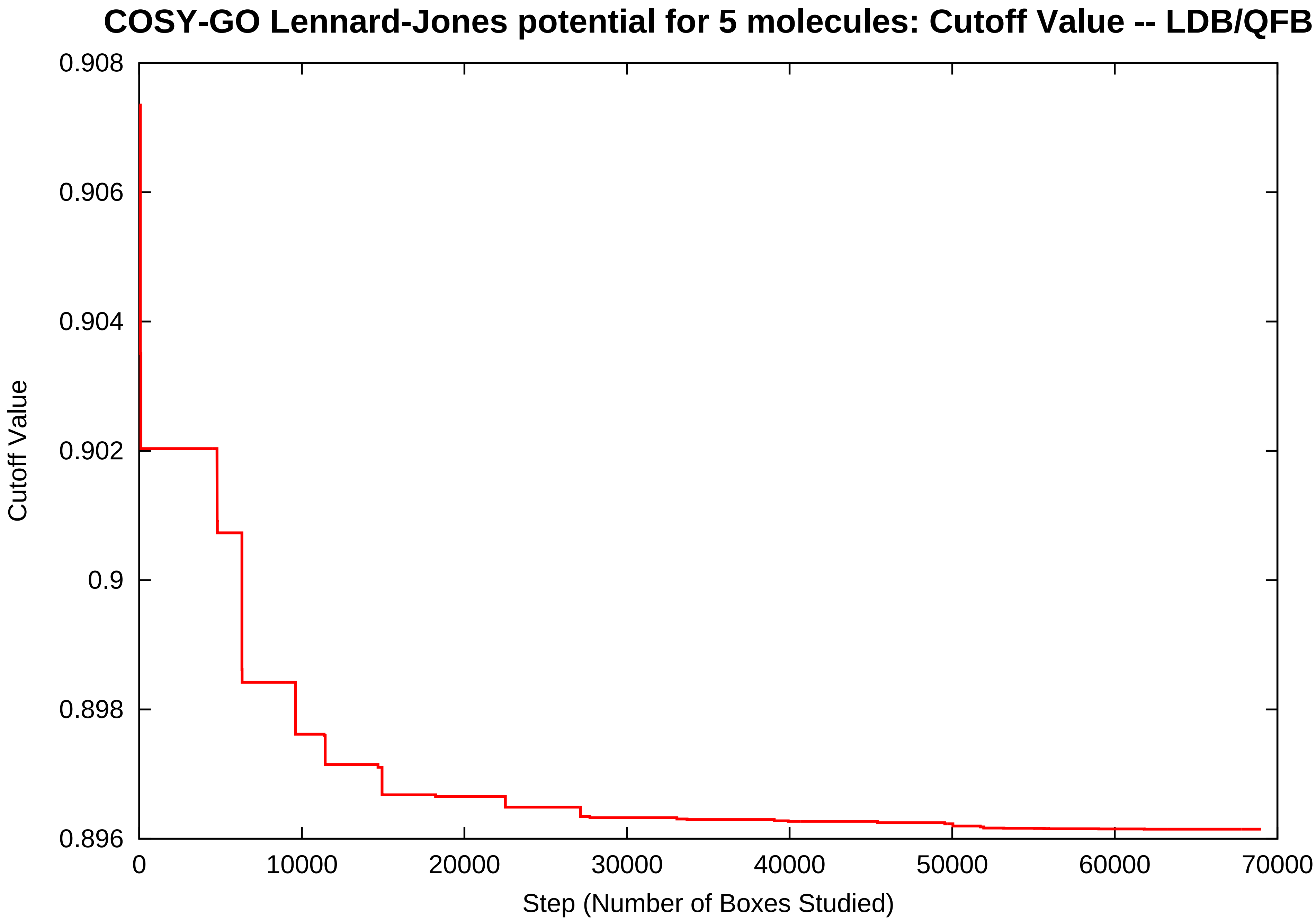
$$V = \sum_{i<j}^{n} V_{LJ}\left(r_i - r_j\right)$$

Study $n = 3, 4, 5$. Pop quiz: What do resulting molecules look like?


COSY-GO Lennard-Jones potential for 4 molecules: Cutoff Value -- LDB/QFB
Cutoff Value
0.006
0.005
0.004
0.003
0.002
0.001
0
0
2000
4000
6000
8000
10000
12000
14000
16000
Step (Number of Boxes Studied)

COSY-GO Lennard-Jones potential for 5 molecules: Number of Boxes -- LDB/QFB

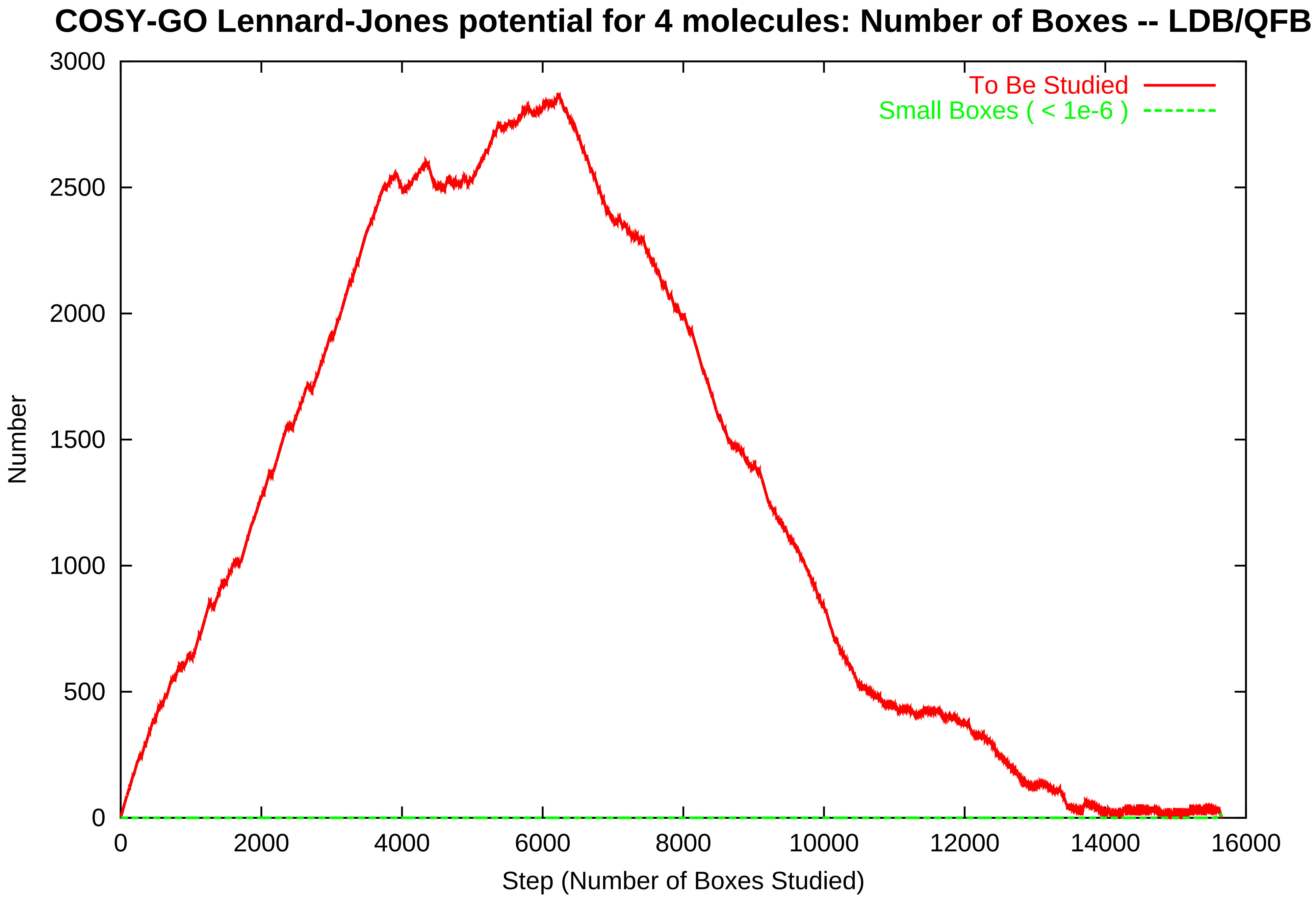



COSY-GO Lennard-Jones potential for 5 molecules: Cutoff Value -- LDB/QFB
Cutoff Value
0.908
0.906
0.904
0.902
0.9
0.898
0.896
0
10000
20000
30000
40000
50000
60000
70000
Step (Number of Boxes Studied)

Lennard-Jones Potentials - Results

Find minimum with COSY-GO and Globsol.
Use TMs of Order 5, QFB&LFB.
Use Globsol in default mode.

Problem	CPU-time needed	Max list	Total # of Boxes
n=4, COSY	89 sec	2,866	15,655
n=5, COSY	1,550 sec	6,321	69,001


COSY-GO Lennard-Jones potential for 4 molecules: Number of Boxes -- LDB/QFB
To Be Studied
Small Boxes (< 1e-6)
Number
Step (Number of Boxes Studied)
0
500
1000
1500
2000
2500
3000
0
2000
4000
6000
8000
10000
12000
14000
16000

Lennard-Jones Potentials - Results

Find minimum with COSY-GO and Globsol.
Use TMs of Order 5, QFB&LFB.
Use Globsol in default mode.

Problem	CPU-time needed	Max list	Total # of Boxes
n=4, COSY	89 sec	2,866	15,655
n=5, COSY	1,550 sec	6,321	69,001
n=4, Globsol	5,833 sec		243,911
n=5,□Globsol□	>60,530□sec (not finished yet)		

The Higher Order Bounder

After removing first and second order part of polynomial, we have

$$\begin{aligned} P\left(\vec{x}-\vec{x}_0\right) &= \tilde{P}\left(\vec{x}-\vec{x}_c\right) \\ &= b+\frac{1}{2}\left(\vec{x}-\vec{x}_c\right)^T H\left(\vec{x}-\vec{x}_c\right)+\tilde{P}_{>2}\left(\vec{x}-\vec{x}_c\right), \end{aligned}$$

Goal: want to find *nonlinear* polynomial $\vec{\mathcal{T}}:\mathbb{R}^v \rightarrow \mathbb{R}^v$ such that with $\vec{y}=\left(\vec{x}-\vec{x}_0\right)$, we have

$$\frac{1}{2}\vec{\mathcal{T}}\left(\vec{y}\right)^T H\vec{\mathcal{T}}\left(\vec{y}\right) =_n \frac{1}{2}\vec{y}^T H\vec{y}+\tilde{P}_{>2}\left(\vec{y}\right),$$

The Higher Order Bounder Algorithm

Will do this to arbitrary order, in an order-by-order fashion. Let $\vec{\mathcal{T}}_m(\vec{y})$ denote the part of $\vec{\mathcal{T}}(\vec{y})$ consisting of the terms of the m-th order, so that

$$\vec{\mathcal{T}}(\vec{y}) = \sum_{m=0}^{n-1} \vec{\mathcal{T}}_m(\vec{y}) \text{. Let } \vec{\mathcal{T}}_{\leq m}(\vec{y}) = \sum_{l=0}^{m} \vec{\mathcal{T}}_l(\vec{y}) \text{.}$$

Note $\vec{\mathcal{T}}_1(\vec{y}) = \vec{y}$. Let us now define a sequence of real-valued polynomial functions $\mathcal{S}_m(\vec{y})$ by

$$\mathcal{S}_m(\vec{y}) = \tilde{P}_{\geq 2}(\vec{y}) - \frac{1}{2}\vec{\mathcal{T}}_{\leq m-1}(\vec{y})^T H \vec{\mathcal{T}}_{\leq m-1}(\vec{y}) \text{ for } m = 1, 2, \ldots, n.$$

The Higher Order Bounder II

Assume we have determined $\vec{\mathcal{T}}_{\leq m-1}$. We want to determine $\vec{\mathcal{T}}_m$. Note that then, $\mathcal{S}_m\left(\vec{y}\right)$ has only terms of order $m+1$ and higher.

We demand

$$\begin{aligned}
0 &=_{m+1} \tilde{P}_{\geq 2}\left(\vec{y}\right)-\frac{1}{2}\left(\vec{\mathcal{T}}_{\leq m-1}\left(\vec{y}\right)+\vec{\mathcal{T}}_m\left(\vec{y}\right)\right)^T H\left(\vec{\mathcal{T}}_{\leq m-1}\left(\vec{y}\right)+\vec{\mathcal{T}}_m\left(\vec{y}\right)\right) \\
&=_{m+1} \tilde{P}_{\geq 2}\left(\vec{y}\right)-\frac{1}{2}\vec{\mathcal{T}}_{\leq m-1}\left(\vec{y}\right)^T H\vec{\mathcal{T}}_{\leq m-1}\left(\vec{y}\right) \\
&-\vec{\mathcal{T}}_{\leq m-1}\left(\vec{y}\right)^T H\vec{\mathcal{T}}_m\left(\vec{y}\right)-\frac{1}{2}\vec{\mathcal{T}}_m\left(\vec{y}\right)^T H\vec{\mathcal{T}}_m\left(\vec{y}\right) \\
&=_{m+1} \mathcal{S}_{m-1}\left(\vec{y}\right)-\vec{\mathcal{T}}_{\leq m-1}\left(\vec{y}\right)^T H\vec{\mathcal{T}}_m\left(\vec{y}\right) \\
&=_{m+1} \mathcal{S}_{m-1}\left(\vec{y}\right)-\vec{y}^T H\vec{\mathcal{T}}_m\left(\vec{y}\right).
\end{aligned}$$

This establishes a requirement for the sought $\vec{\mathcal{T}}_m\left(\vec{y}\right)$ .Now note that each term in $\mathcal{S}_{m-1}$ contains at least one of the variables $y_1, ..., y_n$ comprising $\vec{y}=(y_1, ..., y_n)$.

The Higher Order Bounder III

Now factor out one such term in term in $\mathcal{S}_{m-1}$, and write

$$\mathcal{S}_{m-1} = \vec{y}^t \cdot I \cdot \tilde{\mathcal{S}}_{m-1}$$

Then we can satisfy condition on $\vec{\mathcal{T}}_m\left(\vec{y}\right)$ by picking

$$\vec{\mathcal{T}}_m\left(\vec{y}\right) = H^{-1} \cdot \tilde{\mathcal{S}}_{m-1}$$

Example: Smooth Function in 6 Dimensions

Let

$$f\left(\vec{x}\right) = -\exp\left(-\frac{1}{2} g\left(\vec{x}\right)\right) + \frac{1}{4}\exp\left(-g\left(\vec{x}\right)\right) \text{ for } \vec{x} \in B_j, \text{ where}$$

$$g\left(\vec{x}\right) = \left(\sum_{i=1}^{v} \left(R\vec{x}\right)_i^2\right) + \left(\exp\left(\frac{1}{2}\sum_{i=1}^{v}\left(R\vec{x}\right)_i\right) - 1\right)^2$$

with a $v \times v$ rotation matrix R. Has resemblance to a linear combination of two Gaussian functions.

Choose boxes

$$B_j = a + 2^{-j-1} \cdot [-1, 1]$$

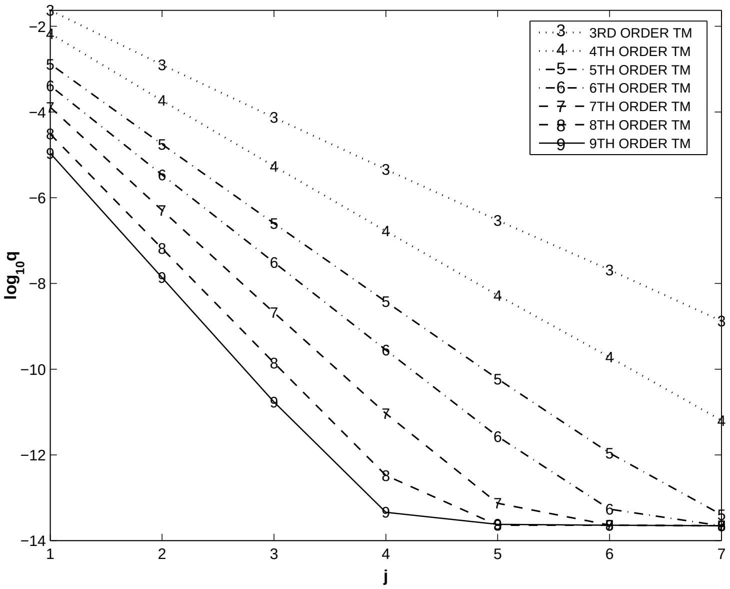


Figure 1: Logarithmic plot of the measurements of an upper bound q of the overestimation in $l(f)$ with different orders $n = 3, \ldots, 9$ of Taylor models.

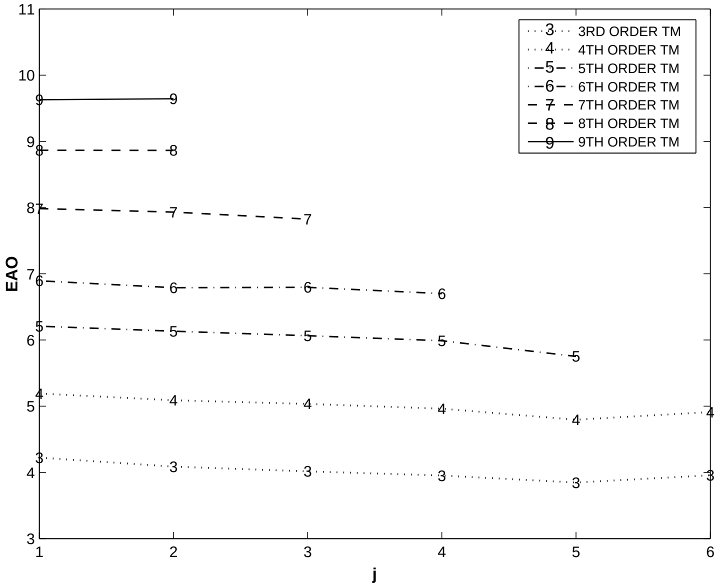


Figure 2: Plot of the empirical approximation order (EAO) for different orders $n = 3, \ldots, 9$ of Taylor model representations.

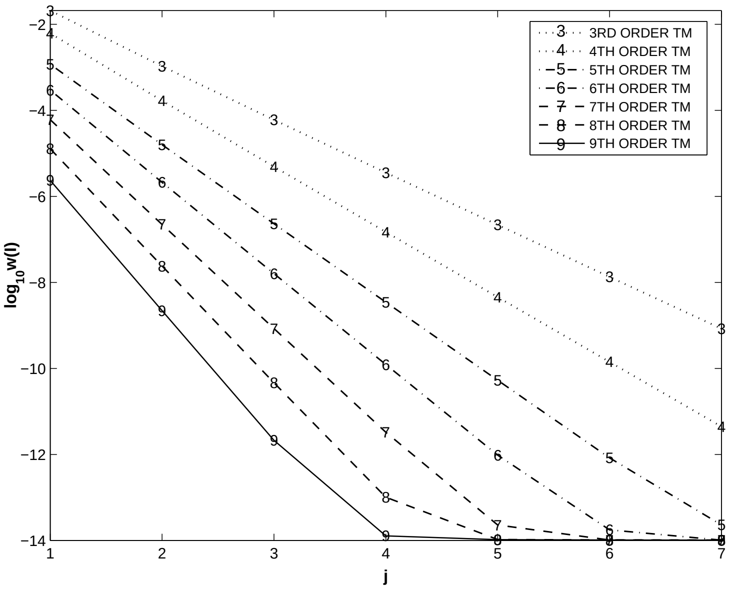


Figure 3: Logarithmic plot of the size $w(I)$ of the remainder bounds of Taylor models of different orders $n = 3, \ldots, 9$.

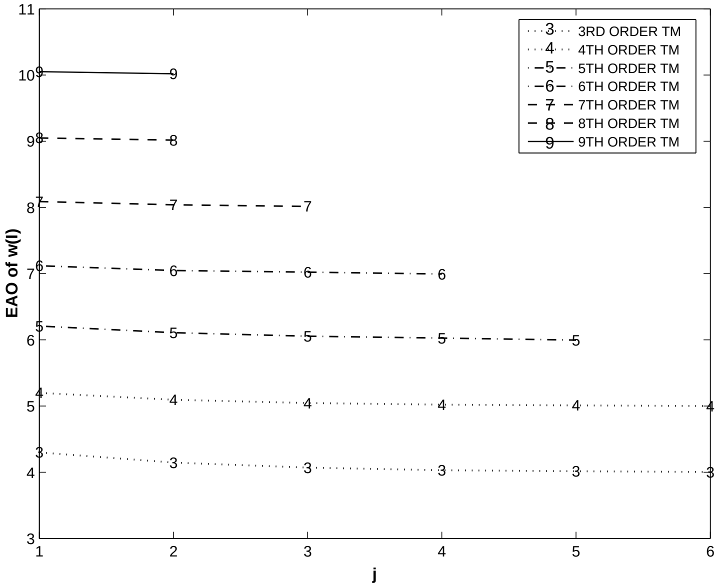


Figure 4: Plot of the empirical approximation order (EAO) of $w(I)$ for different orders $n = 3, \ldots, 9$ of Taylor model representations.

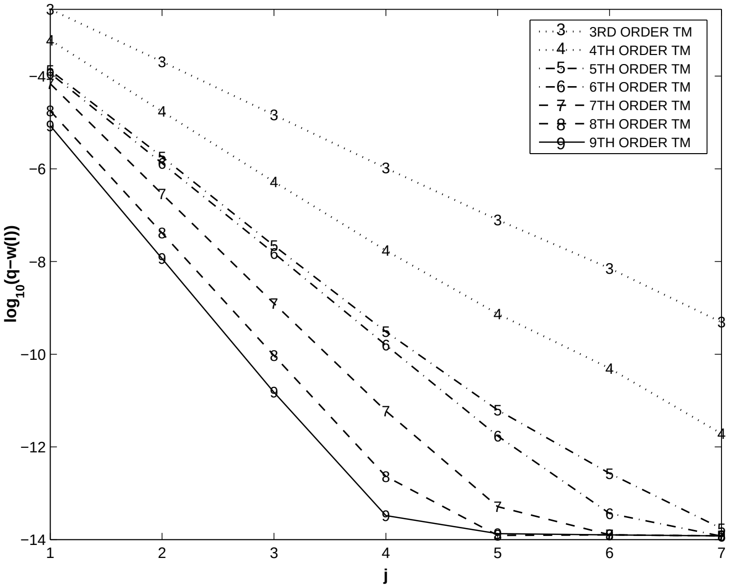


Figure 5: Logarithmic plot of an upper bound $q - w(I)$ of the overestimation in $l(P)$ of Taylor models of orders $n = 3, \ldots, 9$.

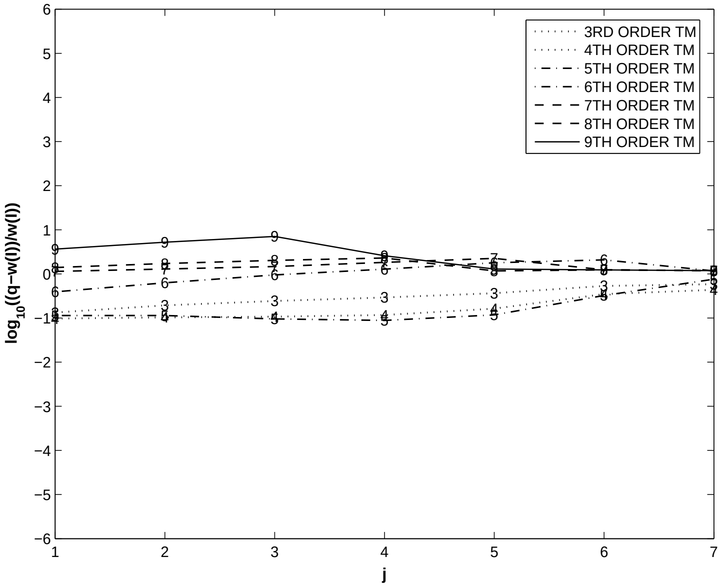


Figure 6: Logarithmic plot of the ratio of $q - w(I)$ to the size $w(I)$ of the remainder bounds of Taylor models of orders $n = 3, \ldots, 9$.

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

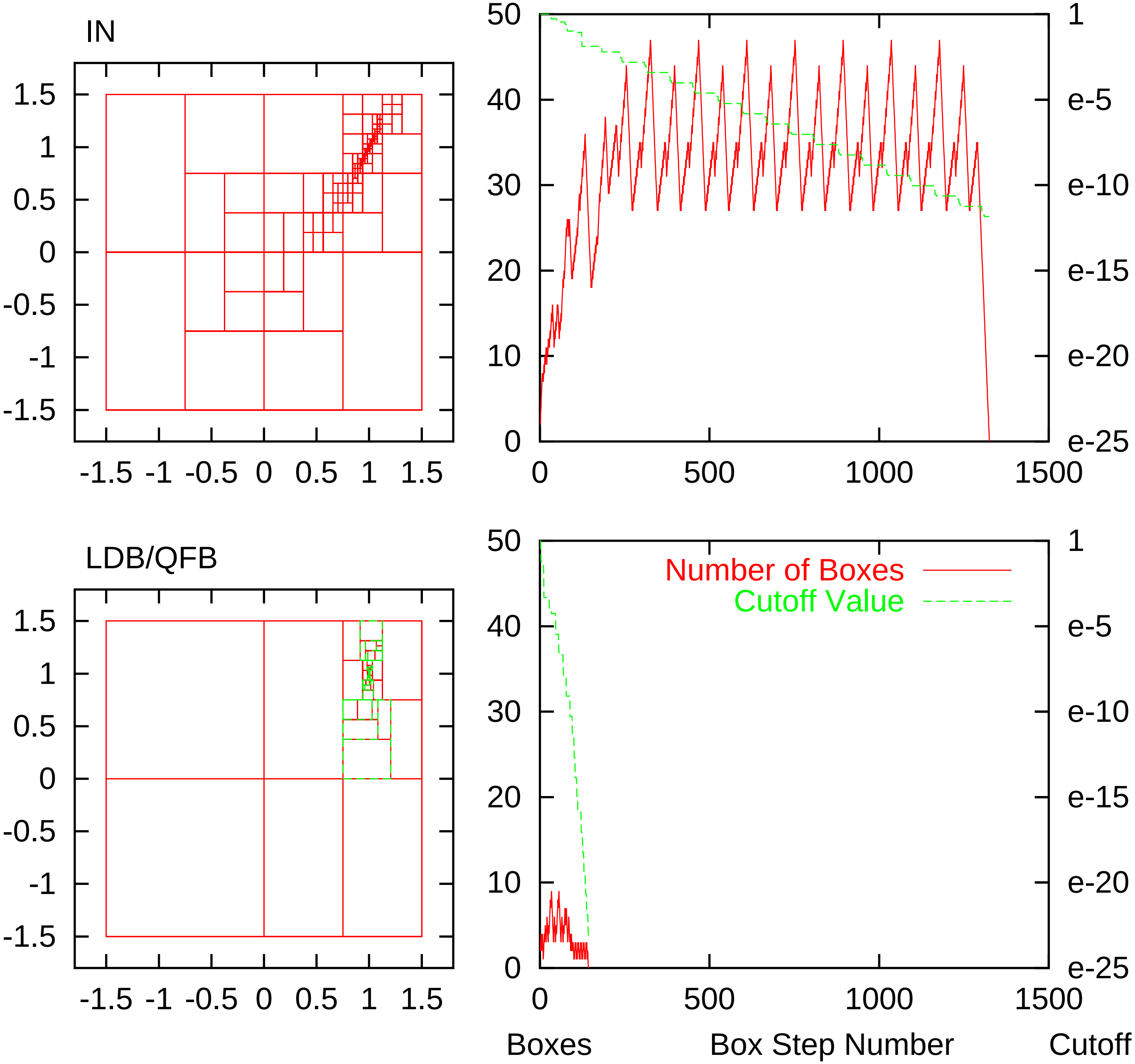
The next example function is created by rotating the above $f(x, y)$ by $30°$. The function is now

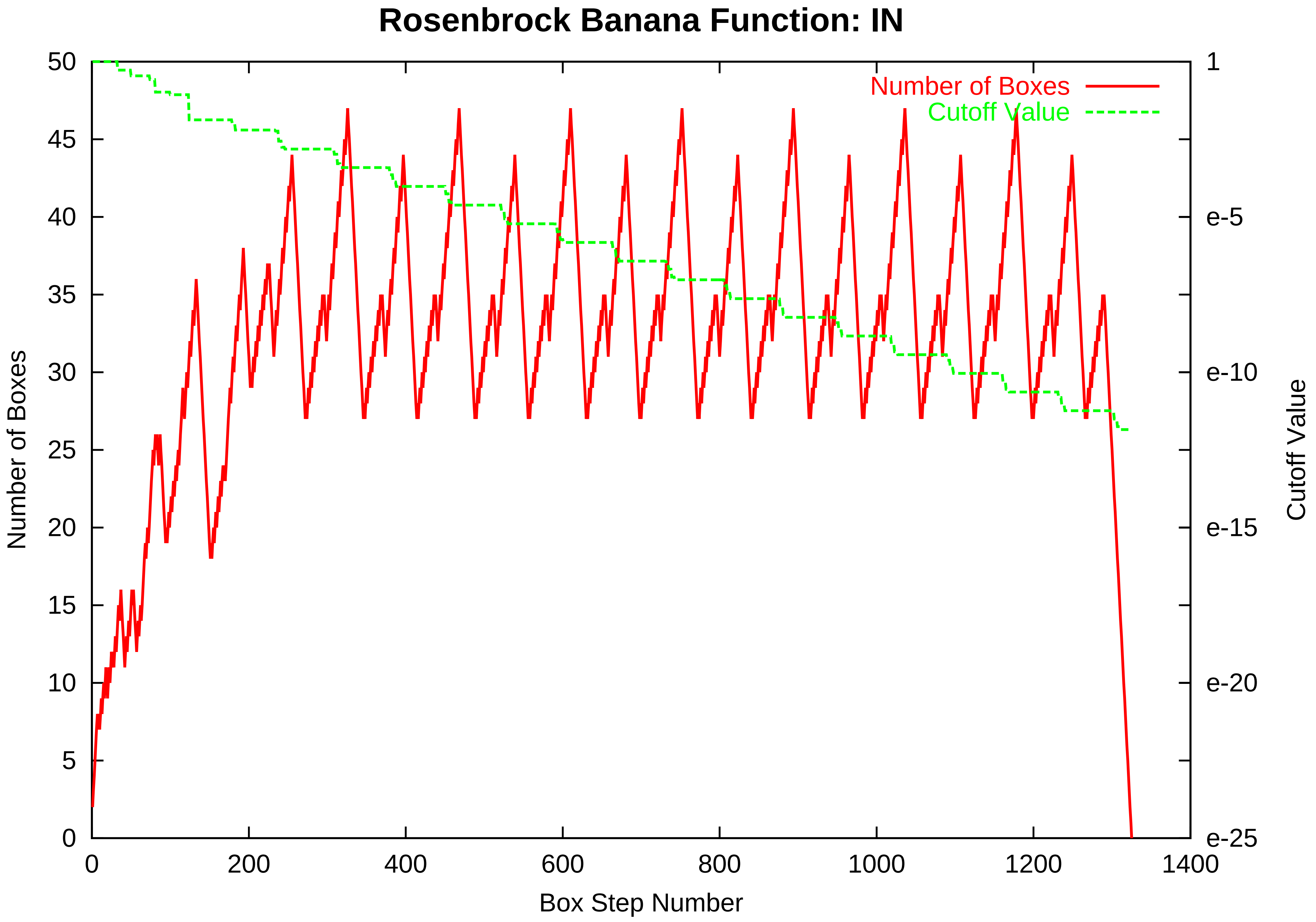
$$f(x,y) = \frac{7}{4}x^2 - \frac{\sqrt{3}}{2}xy + \frac{5}{4}y^2 = \frac{1}{2}(x,y) \cdot \begin{pmatrix} \frac{7}{2} & -\frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & \frac{5}{2} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix},$$

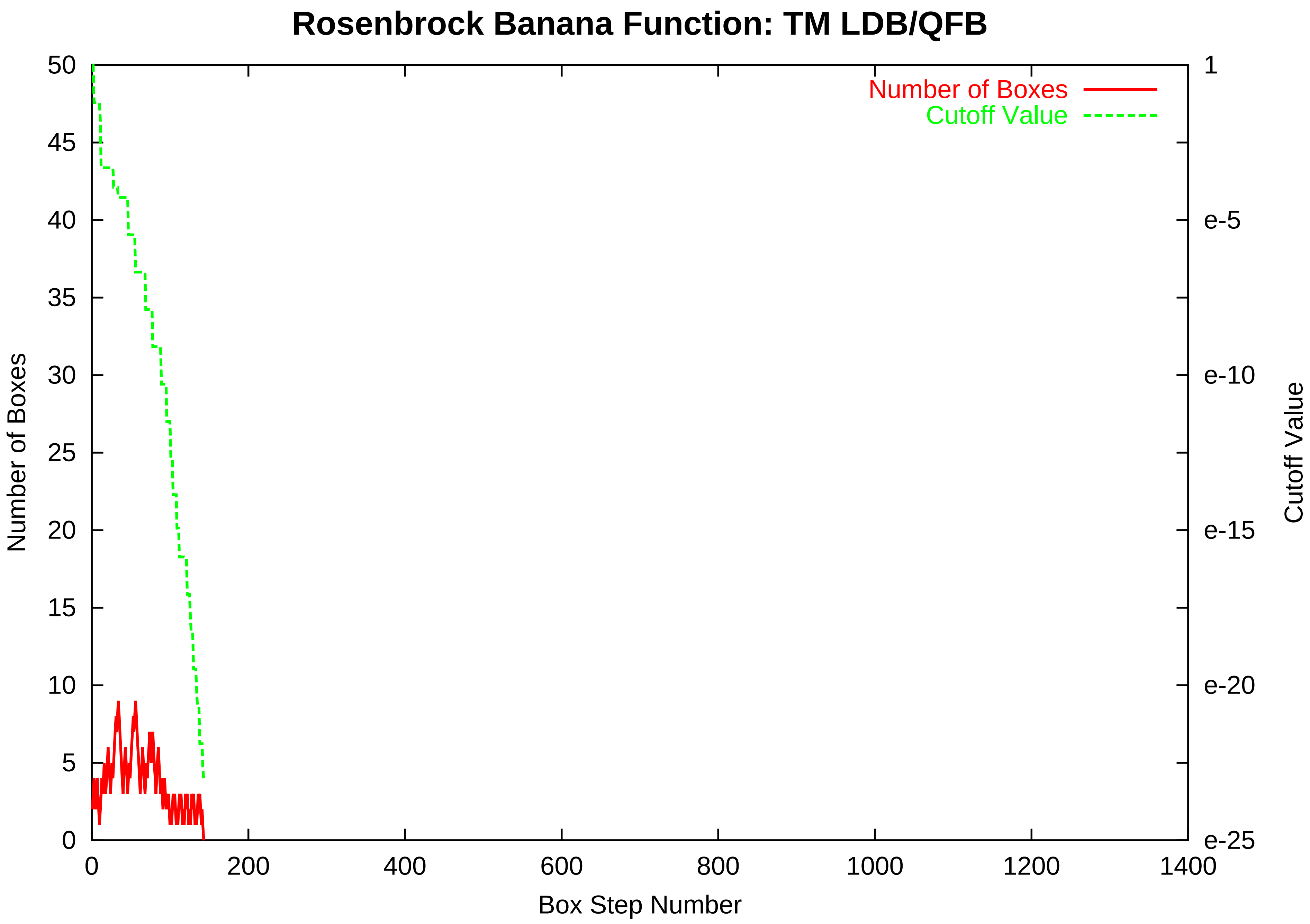
Again, we demand $a = 1$. Using the formula, at this time, we obtain x_m and y_m as

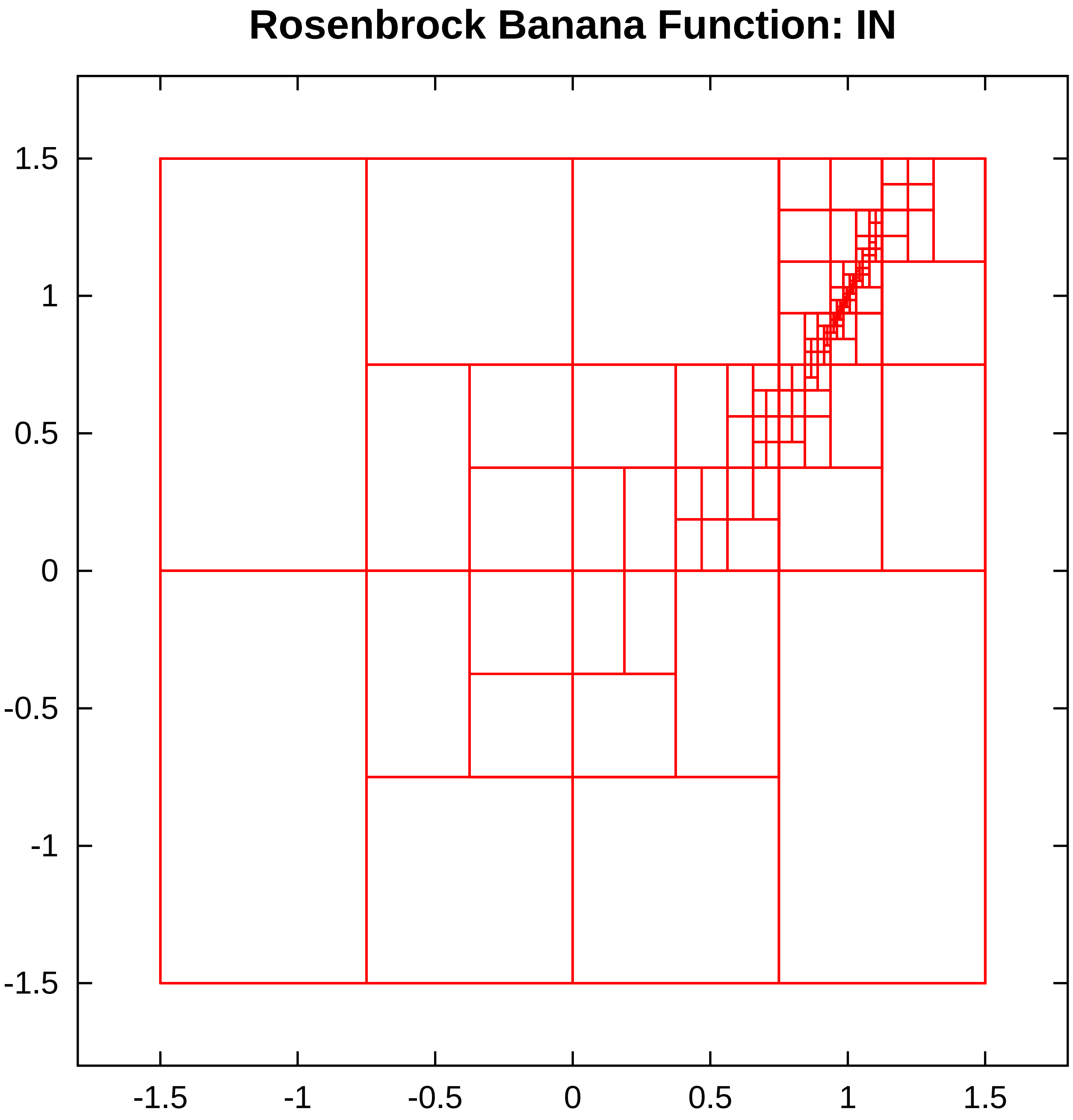
$$x_m = \sqrt{\frac{2 \cdot 1}{\frac{7}{2} - \left(-\frac{\sqrt{3}}{2}\right) \cdot \frac{2}{5} \cdot \left(-\frac{\sqrt{3}}{2}\right)}} = \sqrt{\frac{5}{8}},$$
$$y_m = \sqrt{\frac{2 \cdot 1}{\frac{5}{2} - \left(-\frac{\sqrt{3}}{2}\right) \cdot \frac{2}{7} \cdot \left(-\frac{\sqrt{3}}{2}\right)}} = \sqrt{\frac{7}{8}}.$$

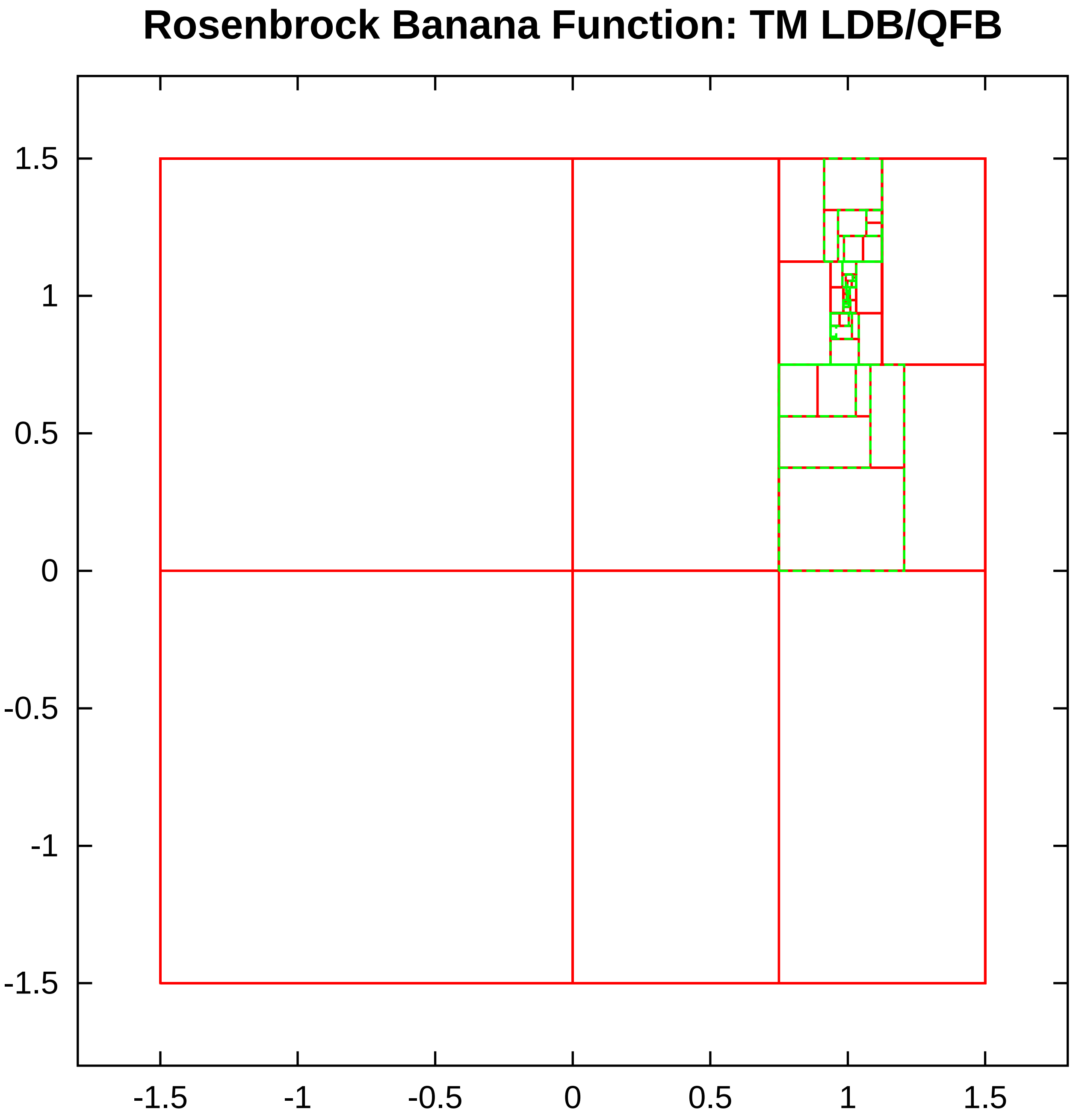
As expected, x_m is larger and y_m is smaller, and the area size of the interval box is larger.


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


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


Rosenbrock Banana Function: IN
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5


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

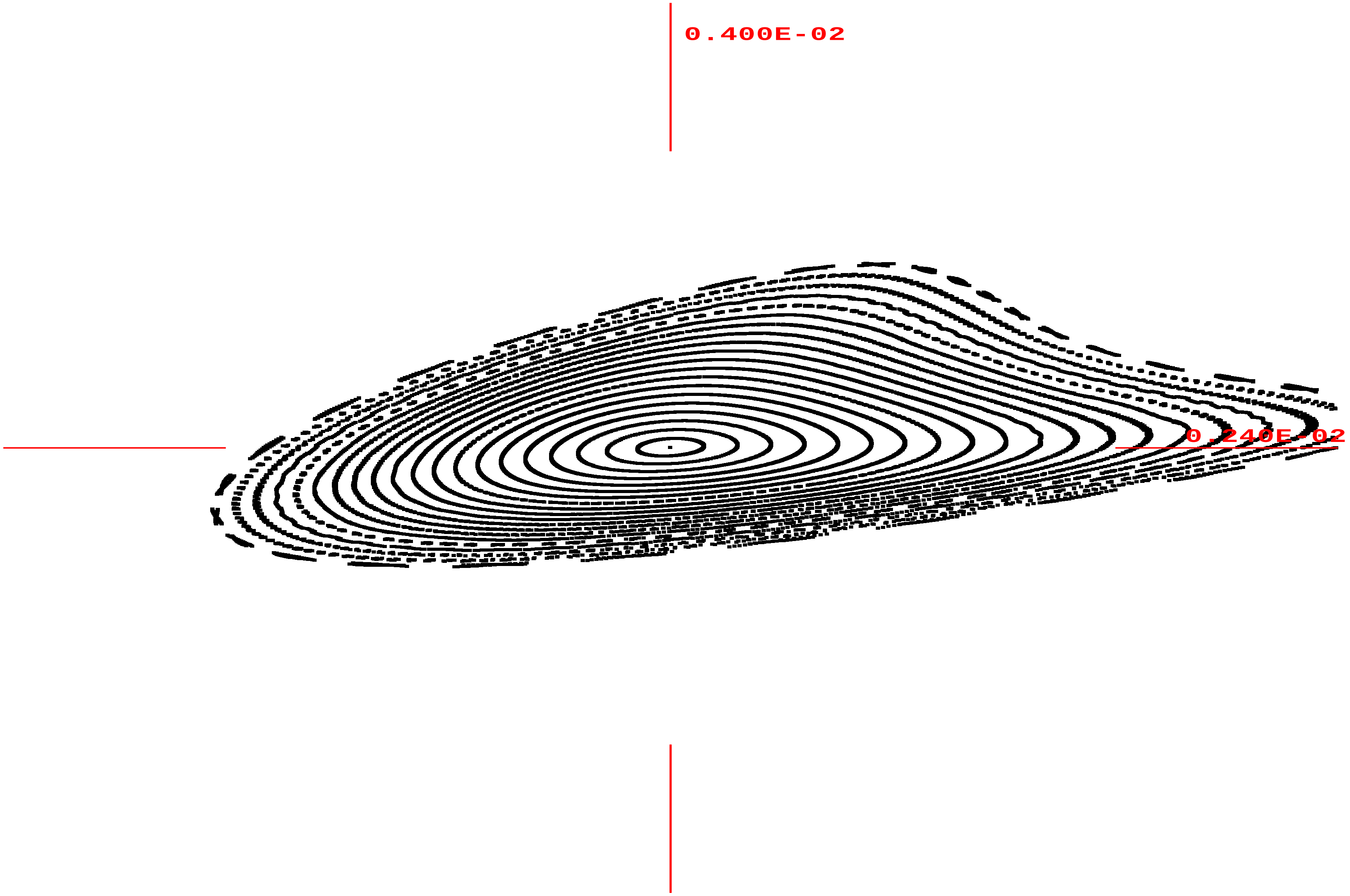
1. **List Management:** Each processor has two lists:
   (a) **Short List of large boxes**, shared with other processors
   (b) A section of Short List is pre-allocated to each processor.
   (c) **Long List of regular boxes** owned by each processor.
   (d) Long List is kept in moderately strict order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


0.400E-02
0.240E-02

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

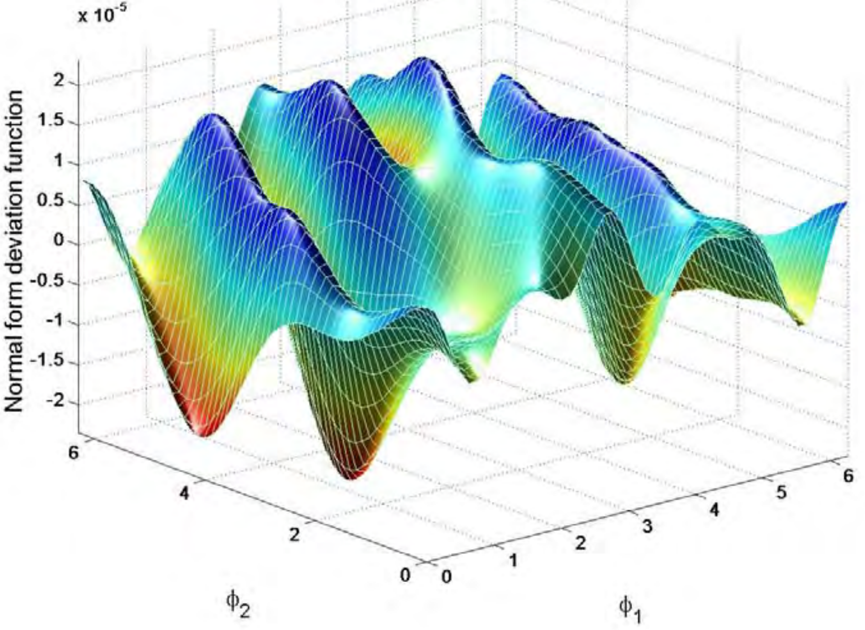
- **Extreme cancellation**; one of the reasons TM methods were invented
- Six-dimensional problem from dynamical systems theory
- Describes invariance defects of a particle accelerator
- Essentially composition of three tenth order polynomials
- The function vanishes identically to order ten
- Study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

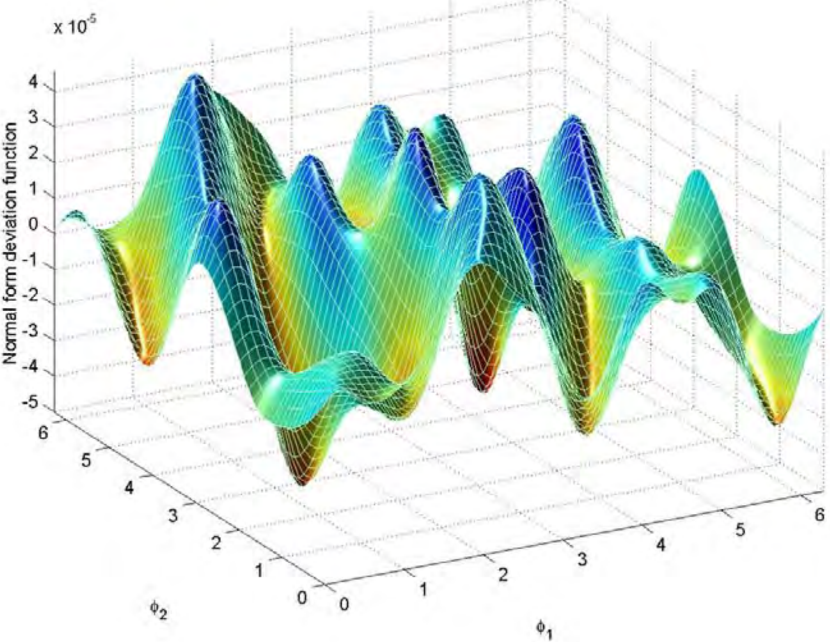
  * one CF in INTLAB takes 45 minutes

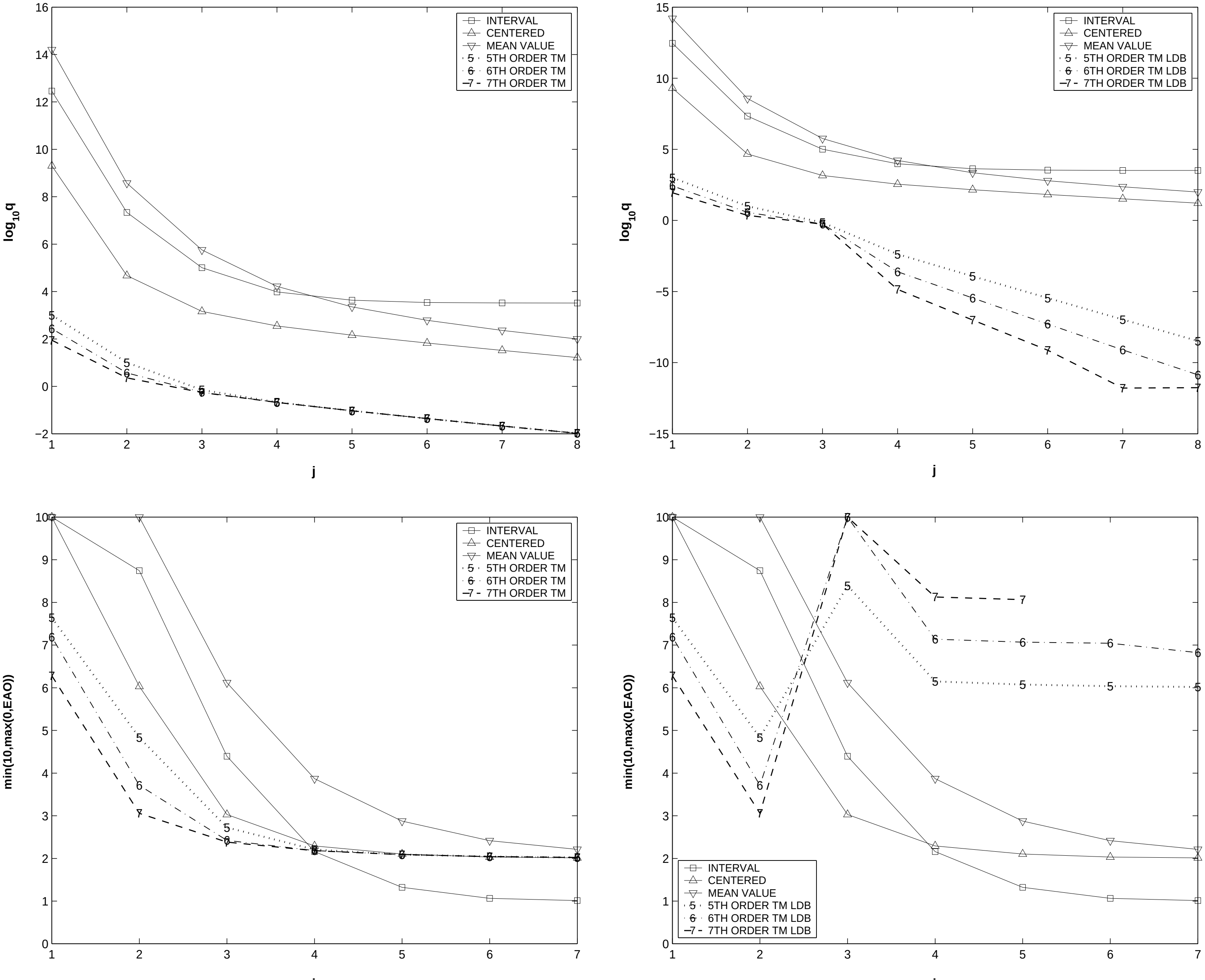
  * one TM of order 7 takes 10 seconds

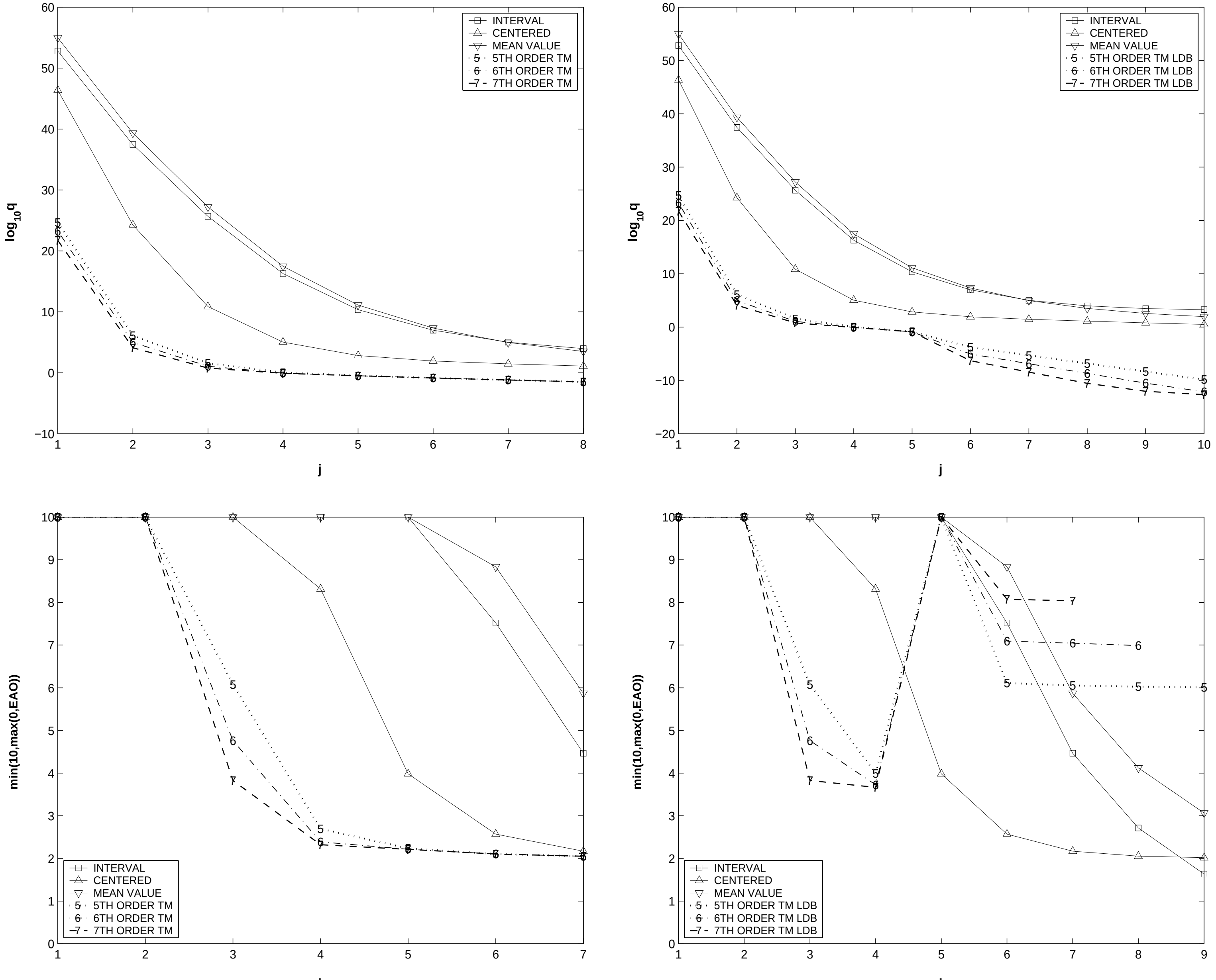
$$f_4(x_1, .., x_6) = \sum_{i=1}^{3} \left(\sqrt{y_{2i-1}^2 + y_{2i}^2} - \sqrt{x_{2i-1}^2 + x_{2i}^2} \right)^2$$

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


x 10^-5
Normal form deviation function
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
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


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))
16
14
12
10
8
6
4
2
0
−2
15
10
5
0
−5
−10
−15
1
2
3
4
5
6
7
8
9



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

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

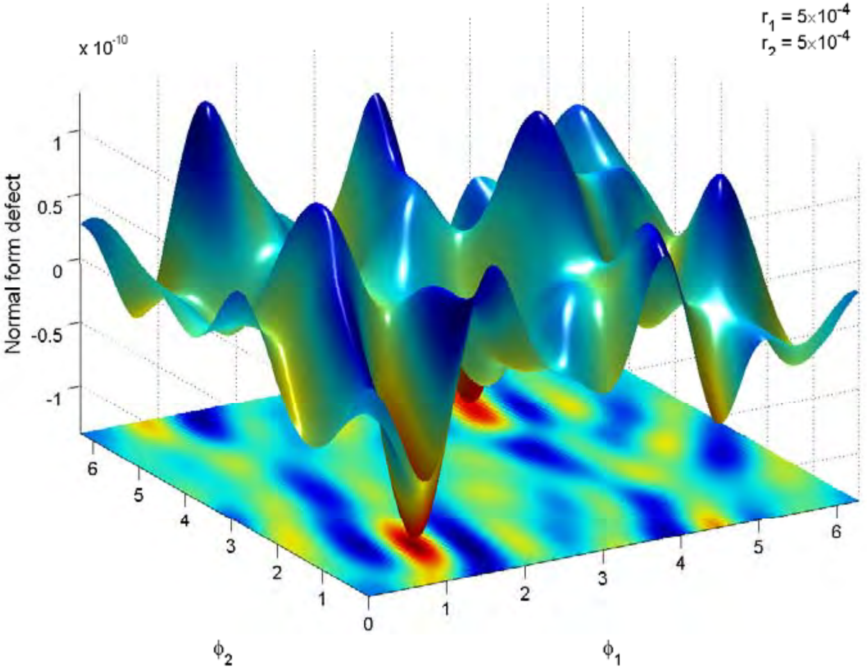
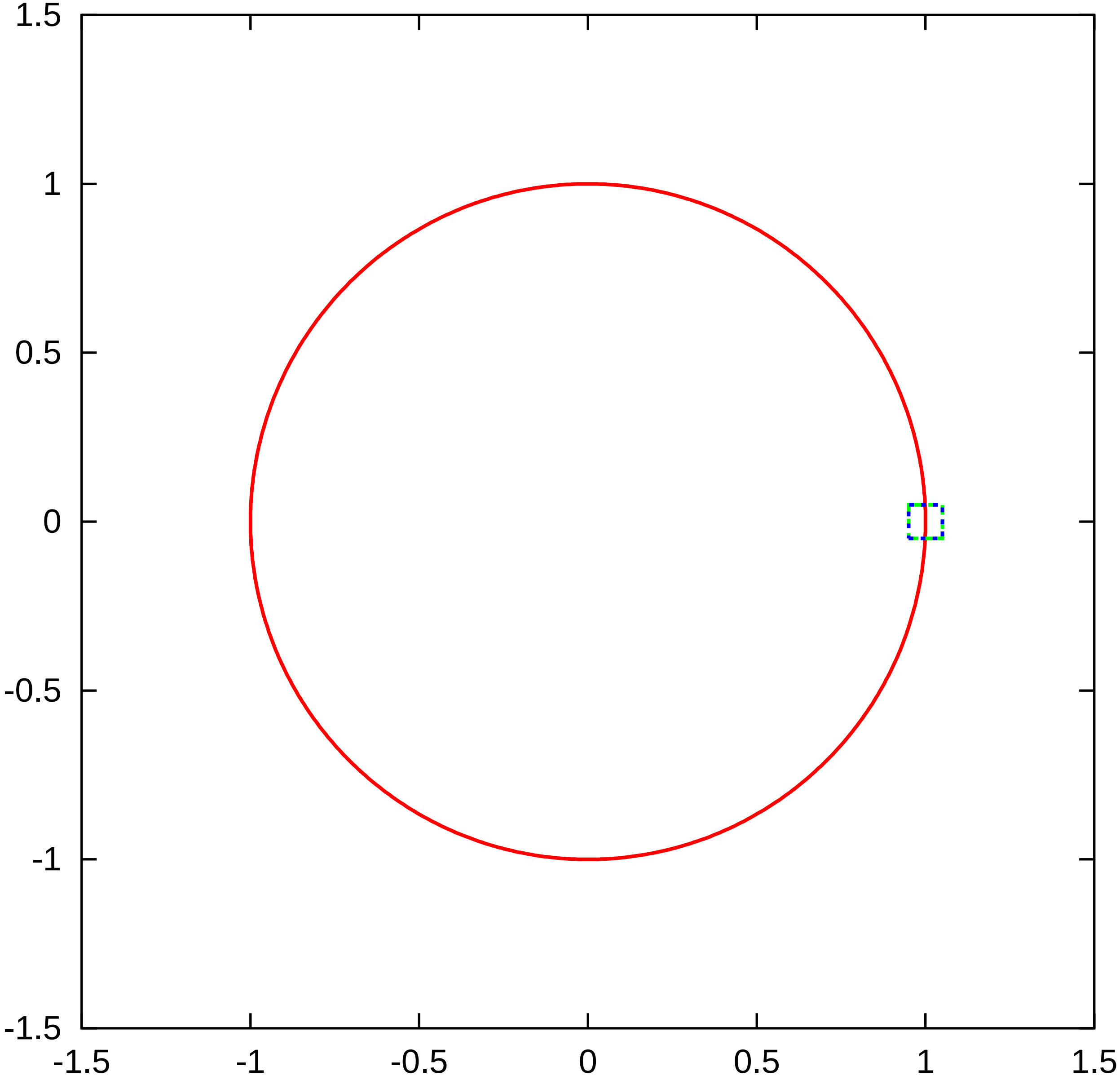


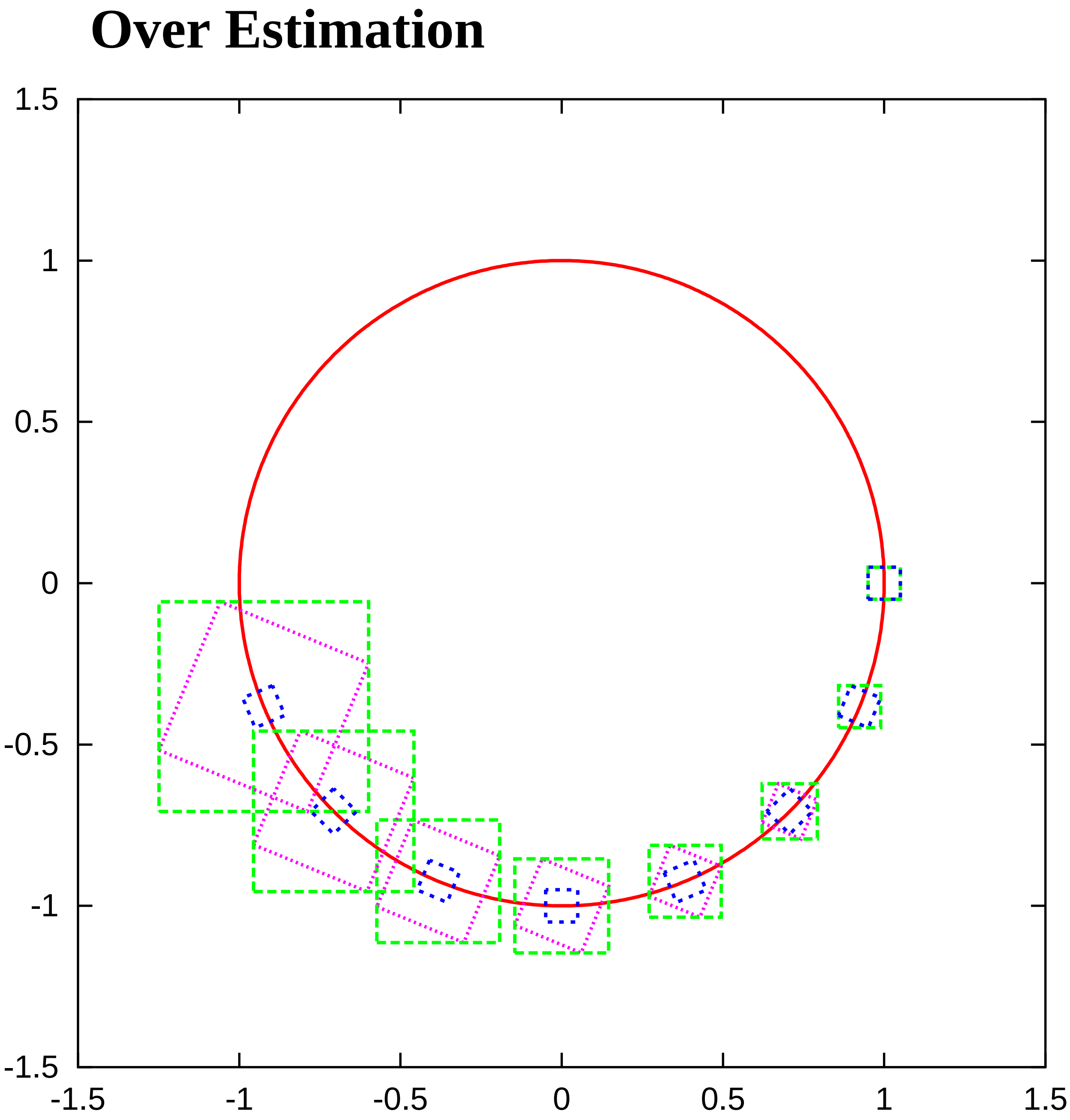
Fig. 9. Projection of the normal form defect function. Dependence on two angle variables for the fixed radii $r_1 = r_2 = 5 \cdot 10^{-4}$

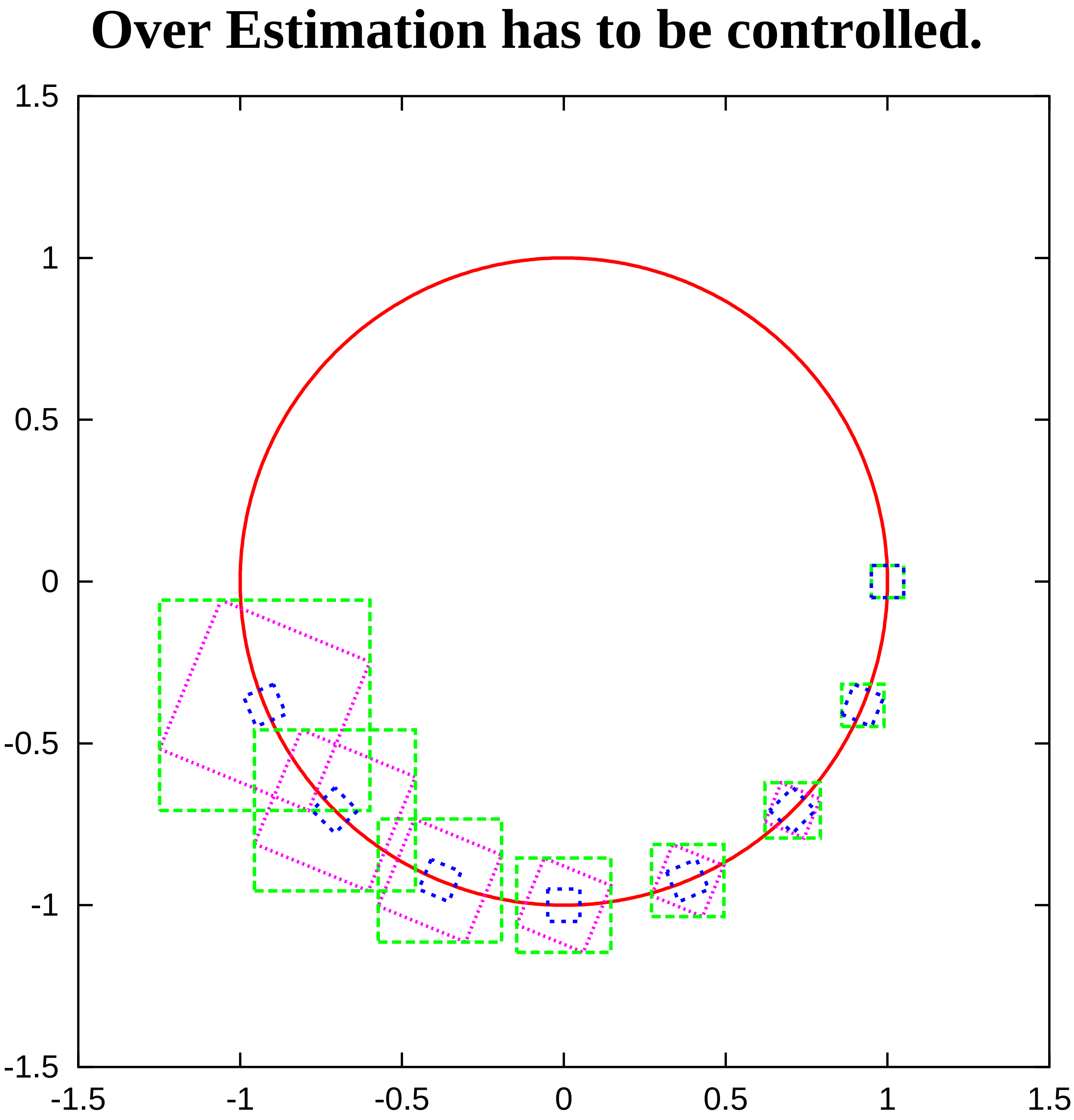
Region	Boxes studied	CPU-time	Bound	Transversal Iterations
$[0.2, 0.4] \cdot 10^{-4}$	$82,930$	$30,603$ sec	$0.859 \cdot 10^{-13}$	$2.3283 \cdot 10^8$
$[0.4, 0.6] \cdot 10^{-4}$	$82,626$	$30,603$ sec	$0.587 \cdot 10^{-12}$	$3.4072 \cdot 10^7$
$[0.6, 0.9] \cdot 10^{-4}$	$64,131$	$14,441$ sec	$0.616 \cdot 10^{-11}$	$4.8701 \cdot 10^6$
$[0.9, 1.2] \cdot 10^{-4}$	$73,701$	$13,501$ sec	$0.372 \cdot 10^{-10}$	$8.0645 \cdot 10^5$
$[1.2, 1.5] \cdot 10^{-4}$	$106,929$	$24,304$ sec	$0.144 \cdot 10^{-9}$	$2.0833 \cdot 10^5$
$[1.5, 1.8] \cdot 10^{-4}$	$111,391$	$26,103$ 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.

To transport a large phase space volume with validation,




Over Estimation
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5


Over Estimation has to be controlled.
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

The Use of Schauder's Theorem

Re-write differential equation as integral equation

$$\vec{r}(t) = \vec{r}_0 + \int_{t_0}^{t} \vec{F}(\vec{r}(t'), t')\, dt'.$$

Now introduce the operator

$$A : \vec{C}^0[t_0, t_1] \to \vec{C}^0[t_0, t_1]$$

on space of continuous functions via

$$A\left(\vec{f}\right)(t) = \vec{r}_0 + \int_{t_0}^{t} \vec{F}\left(\vec{f}(t'), t'\right)\, dt'.$$

Then the solution of ODE is transformed to a fixed-point problem on space of continuous functions

$$\vec{r} = A(\vec{r}).$$

Theorem (Schauder): *Let A be a continuous operator on the Banach Space X. Let $M \subset X$ be compact and convex, and let $A(M) \subset M$. Then A has a fixed point in M, i.e. there is an $\vec{r} \in M$ such that $A(\vec{r}) = \vec{r}$.*

Satisfying Requirements of the Schauder Theorem

Here, $X = \vec{C}^0[t_0, t_1]$, Banach space of continuous functions on $[t_0, t_1]$, equipped with maximum norm. The integral operator A is continuous. The strategy to apply Schauder's Fixed Point Theorem consists of the following steps:

1. Determine family Y of subsets of X, the Schauder Candidate Sets. Each set in Y should be compact and convex, it should be contained in suitable Taylor model, and its image under A should be in Y.
2. Using RDA, determine initial set $M_0 \in Y$ that satisfies $A(M_0) \subset M_0$. Then last requirement of Schauder is satisfied, and M_0 contains solution.
3. Iteratively generate $M_i = A(M_{i-1})$. Each M_i also satisfies $A(M_i) \subset M_{i,}$ and we have $M_1 \supset M_2 \supset$... Continue until size stabilizes sufficiently.

Schauder Candidate Sets

As first step, it is necessary to establish a family of sets Y from which to draw candidates for M_0. Let $(\vec{P}+\vec{I})$ be a Taylor model depending on time as well as the initial condition $\vec{r}_0$. Then define the associated set $M_{\vec{P}+\vec{I}}$ as follows:

$$\begin{aligned} M_{\vec{P}+\vec{I}} \subset \vec{C}^0[t_0,t_1]; \text{ and for } \vec{r} \in M_{\vec{P}+\vec{I}} :\\ \vec{r}(t_0) = \vec{r}_0 \\ \vec{r}(t) \in \vec{P}+\vec{I}\; \forall t \in [t_0,t_1]\; \forall \vec{r}_0 \\ |\vec{r}(t') - \vec{r}(t'')| \leq k|t'-t''|\, \forall t',t'' \in [t_0,t_1]\; \forall \vec{r}_0 \end{aligned}$$

In the last condition, k is a bound for $\vec{F}$, which exists because $\vec{F}$ is continuous and the solutions can cover only finite range over interval $[t_0,t_1]$. The last condition means that all $\vec{r} \in M_{\vec{P}+\vec{I}}$ are uniformly Lipschitz with constant k. Define the candidate set Y as

$$Y = \bigcup_{\vec{P}+\vec{I}} M_{\vec{P}+\vec{I}}$$

Convexity, Compactness, Invariance of Candidate Sets

Let $M \in Y$. Then M is convex, because

$$\vec{x}_1, \vec{x}_2 \in M \Rightarrow$$
$$\alpha \vec{x}_1 + (1-\alpha)\vec{x}_2 \in M \ \forall \alpha \in [0,1]$$

Furthermore, M is compact, i.e. any sequence in M has a clusterpoint in M. To see this, let $(\vec{x}_n)$ be a sequence of functions in M.Then by definition of M, $(\vec{x}_n)$ is uniformly Lipschitz, and thus uniformly equicontinuous. $(\vec{x}_n)$ is also uniformly bounded, and hence according to the Ascoli-Arzela Theorem, has a uniformly convergent subsequence. Since the $\vec{x}_n$ are continuous, so is the limit $\vec{x}^*$ of this subsequence, and since M is closed, the limit $\vec{x}^*$ is in M.

Finally, A maps Y into itself, and the uniform Lipschitzness follows because $\vec{F}$ is bounded by k.

Satisfying Inclusion with Taylor Models

The only remaining requirements for Schauder's theorem is to find a Taylor model $\vec{P}+\vec{I}$ such that

$$A(\vec{P}+\vec{I}) \subset \vec{P}+\vec{I}.$$

But this condition can be checked with Taylor Models.

To succeed with inclusion requirement depends on finding suitable choice for $\vec{P}$ and $\vec{I}$. Furthermore, it is desirable that $\vec{I}$ be tight.

Both benefit from the choice of a polynomial $\vec{P}$ that is already "close" to the true solution of the ODE.

The Polynomial of the Self-Including Set

Attempt sets M^* of the form

$$M^* = M_{\vec{P}^* + \vec{I}^*} \text{ where}$$
$$\vec{P}^* = \mathcal{M}_n(\vec{r}_0, t),$$

the n-th order Taylor expansion of the flow of the ODE. It is to be expected that $\vec{I}^*$ can be chosen smaller and smaller as order n of $\vec{P}^*$ increases.

This requires knowledge of nth order flow $\mathcal{M}_n(\vec{r}_0, t)$, including time dependence. It can be obtained by iterating in polynomial arithmetic, or Taylor models without treatment of a remainder. To this end, one chooses an initial function $\mathcal{M}_n^{(0)}(\vec{r}, t) = \mathcal{I}$, where $\mathcal{I}$ is the identity function, and then iteratively determines

$$\mathcal{M}_n^{(k+1)} =_n A(\mathcal{M}_n^{(k)}).$$

This process converges to the exact result $\mathcal{M}_n$ in exactly n steps.

The Remainder of the Self-Including Set

Now try to find $\vec{I}^*$ such that

$$A(\mathcal{M}_n + \vec{I}^*) \subset \mathcal{M}_n + \vec{I}^*,$$

the Schauder inclusion requirement. Suitable choice for $\vec{I}^*$ requires experimenting, but is greatly simplified by the observation

$$\vec{I}^* \supset \vec{I}^{(0)} = A(\mathcal{M}_n(\vec{r}, t) + [\vec{0}, \vec{0}]) - \mathcal{M}_n(\vec{r}, t).$$

Evaluating the right hand side in RDA yields a lower bound for $\vec{I}^*$, and a benchmark for the size to be expected. Now iteratively try

$$\vec{I}^{(k)} = 2^k \cdot \vec{I}^{(0)},$$

until computational inclusion is found, i.e.

$$A(\mathcal{M}_n(\vec{r}, t) + \vec{I}^{(k)}) \subset \mathcal{M}_n(\vec{r}, t) + \vec{I}^{(k)}.$$

Iterative Refinement of the Self-Including Set

Once computational inclusion has been determined, solution of ODE is known to be contained in the Taylor model $\mathcal{M}_n(\vec{r},t) + \vec{I}^{(k)}$. Set $\vec{I}_{(1)} = \vec{I}^{(k)}$; since the solution is a fixed point of A, it is even contained in

$$A^k(\mathcal{M}_n(\vec{r},t) + \vec{I}_{(1)}) \text{ for all } k.$$

Furthermore, the iterates of A are shrinking in size, i.e.

$$A^k(\mathcal{M}_n(\vec{r},t) + \vec{I}_{(1)}) \subset A^{k-1}(\mathcal{M}_n(\vec{r},t) + \vec{I}_{(1)}) \ \forall k$$

So the width of the remainder bound of the flow can be decreased by iteratively determining

$$\mathcal{M}_n(\vec{r},t) + \vec{I}_{(k)} = A(\mathcal{M}_n(\vec{r},t) + \vec{I}_{(k-1)}),$$

until no further significant decrease in size is achieved. As a result,

$$\mathcal{M}_n(\vec{r},t) + \vec{I}_{(k)}$$

is the desired sharp inclusion of the flow of the original ODE.

The Volterra Equation

Describe dynamics of two conflicting populations

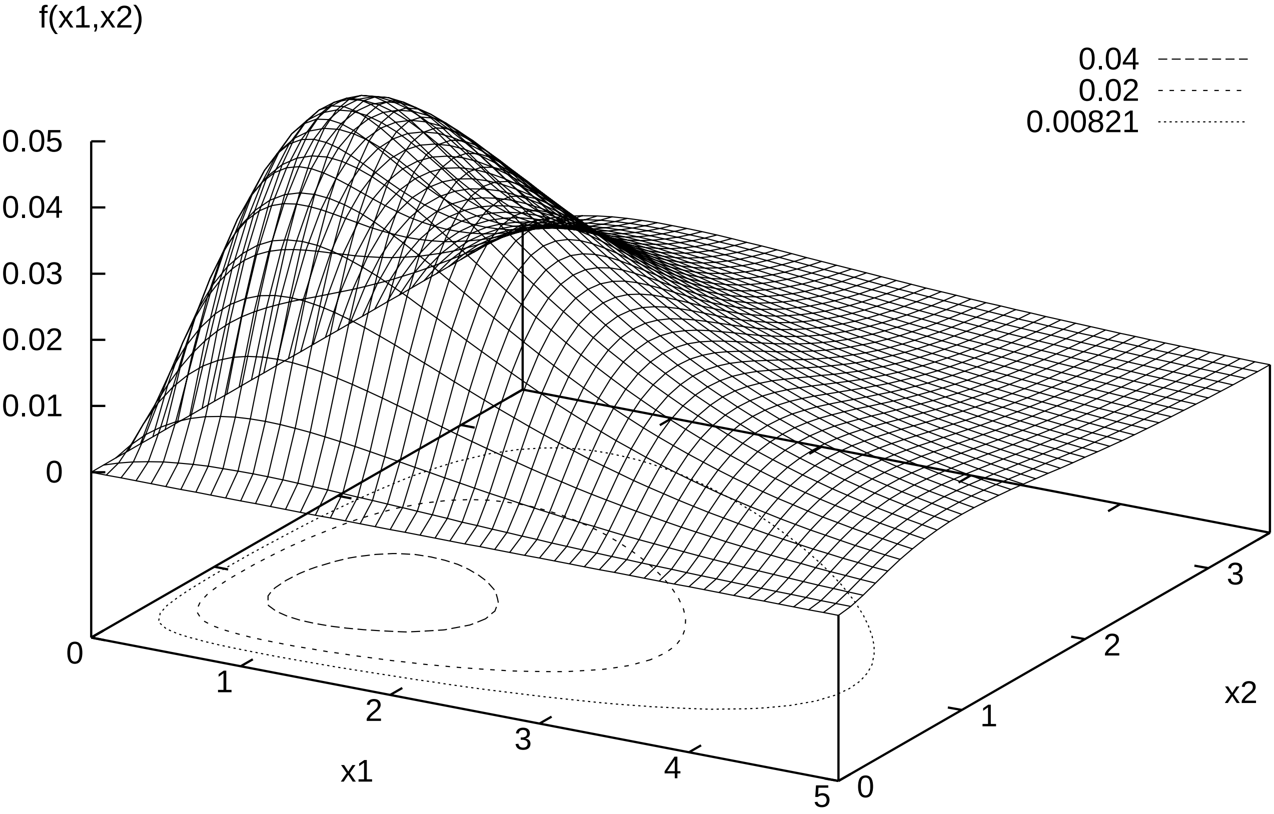
$$\frac{dx_1}{dt} = 2x_1(1 - x_2), \quad \frac{dx_2}{dt} = -x_2(1 - x_1)$$

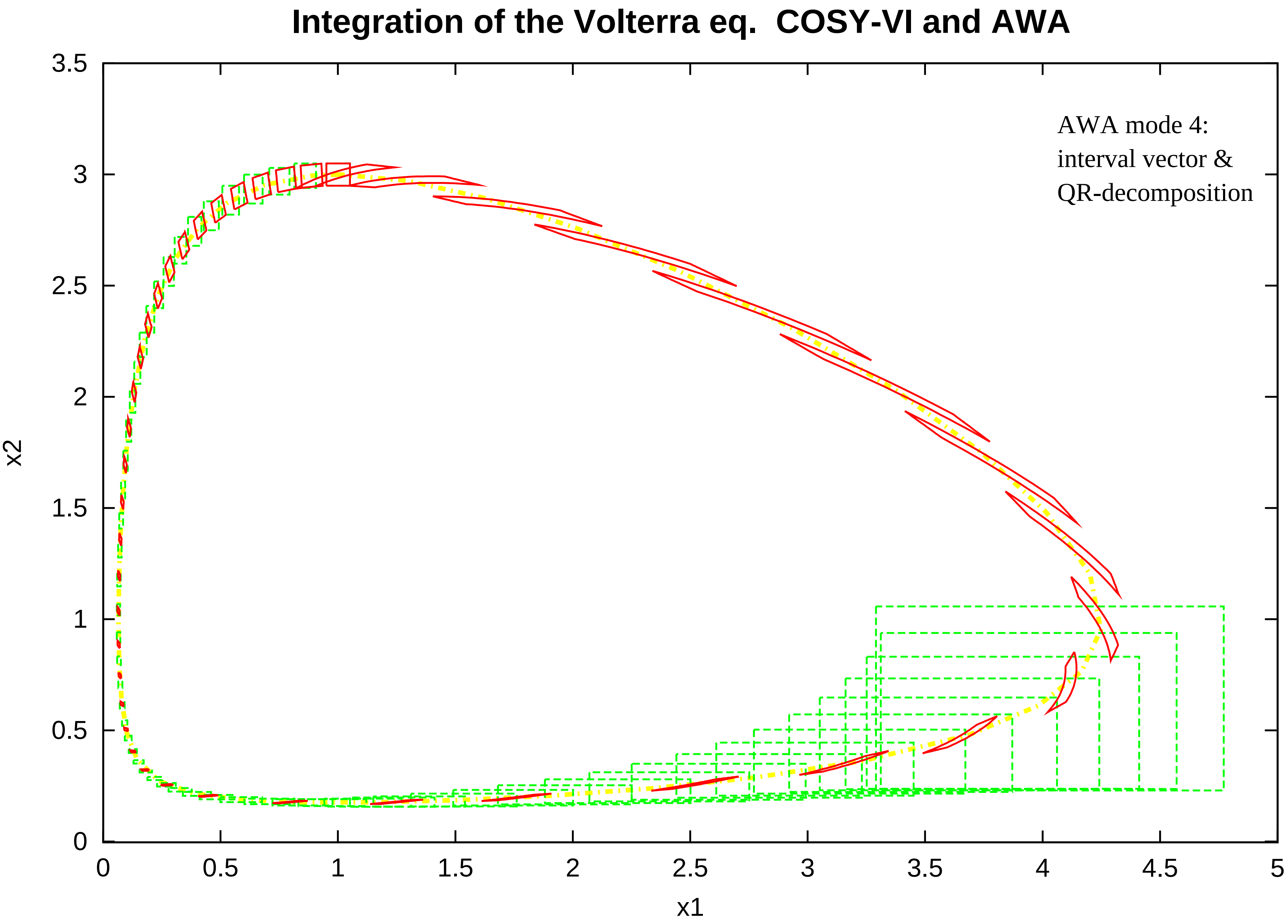
Interested in initial condition

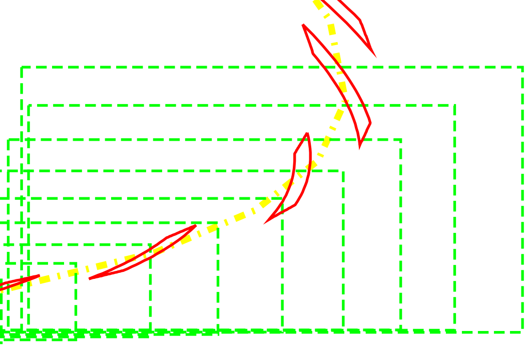
$$x_{01} \in 1 + [-0.05, 0.05], \quad x_{02} \in 3 + [-0.05, 0.05] \quad \text{at } t = 0.$$

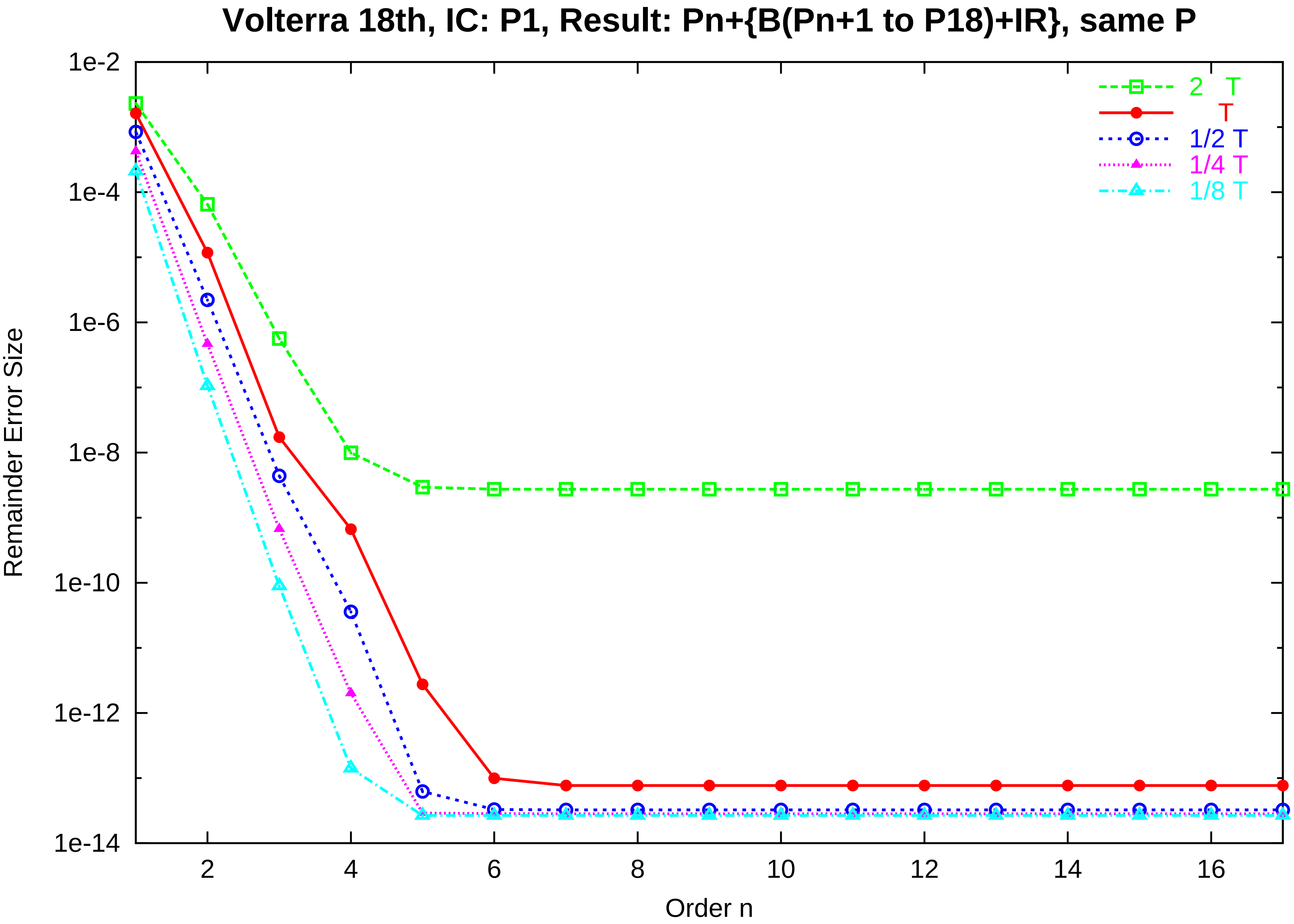
Satisfies constraint condition

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

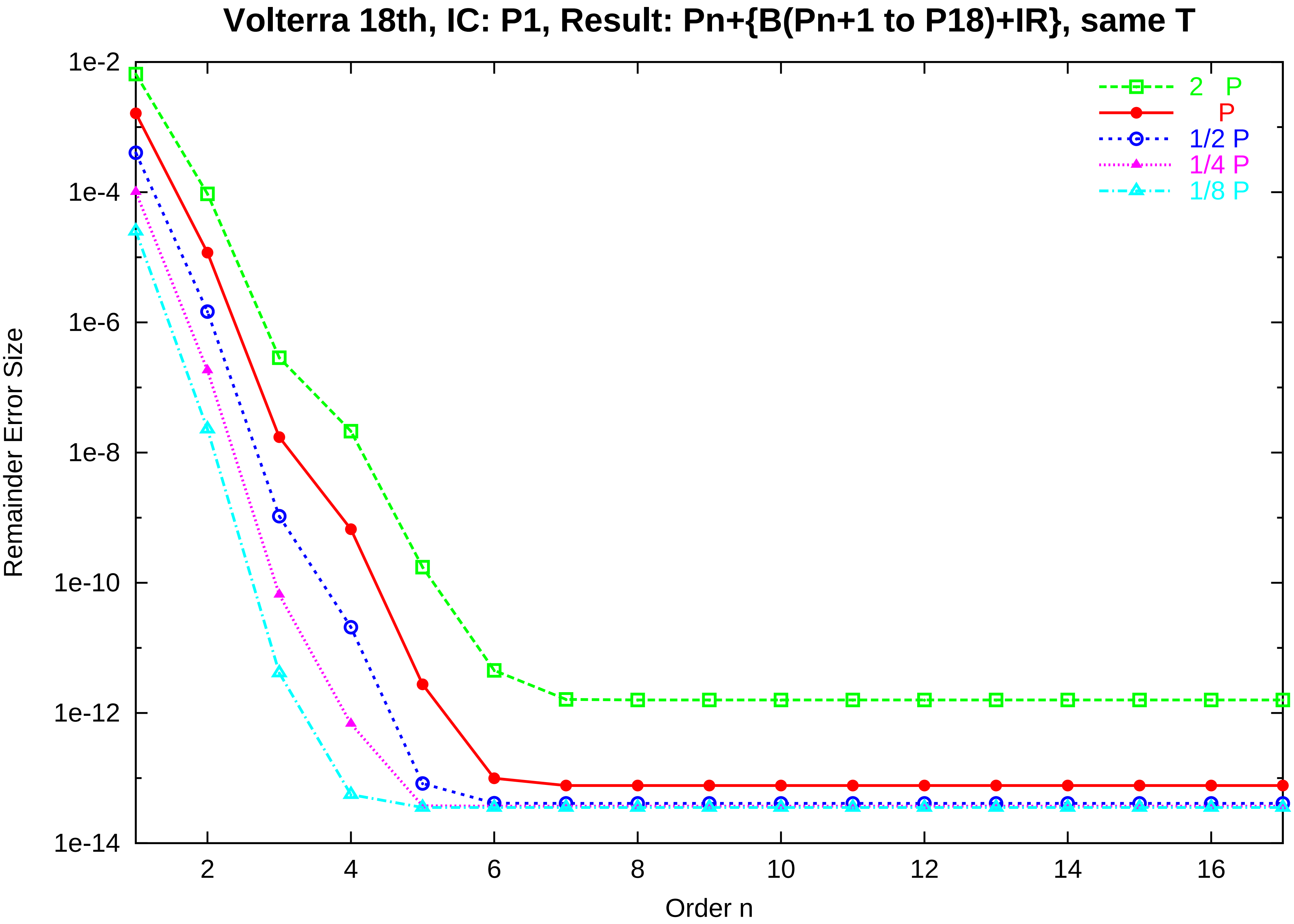

f(x1,x2)
0.04
0.02
0.00821
0.05
0.04
0.03
0.02
0.01
0
0
1
2
3
4
5
x1
0
1
2
3
x2

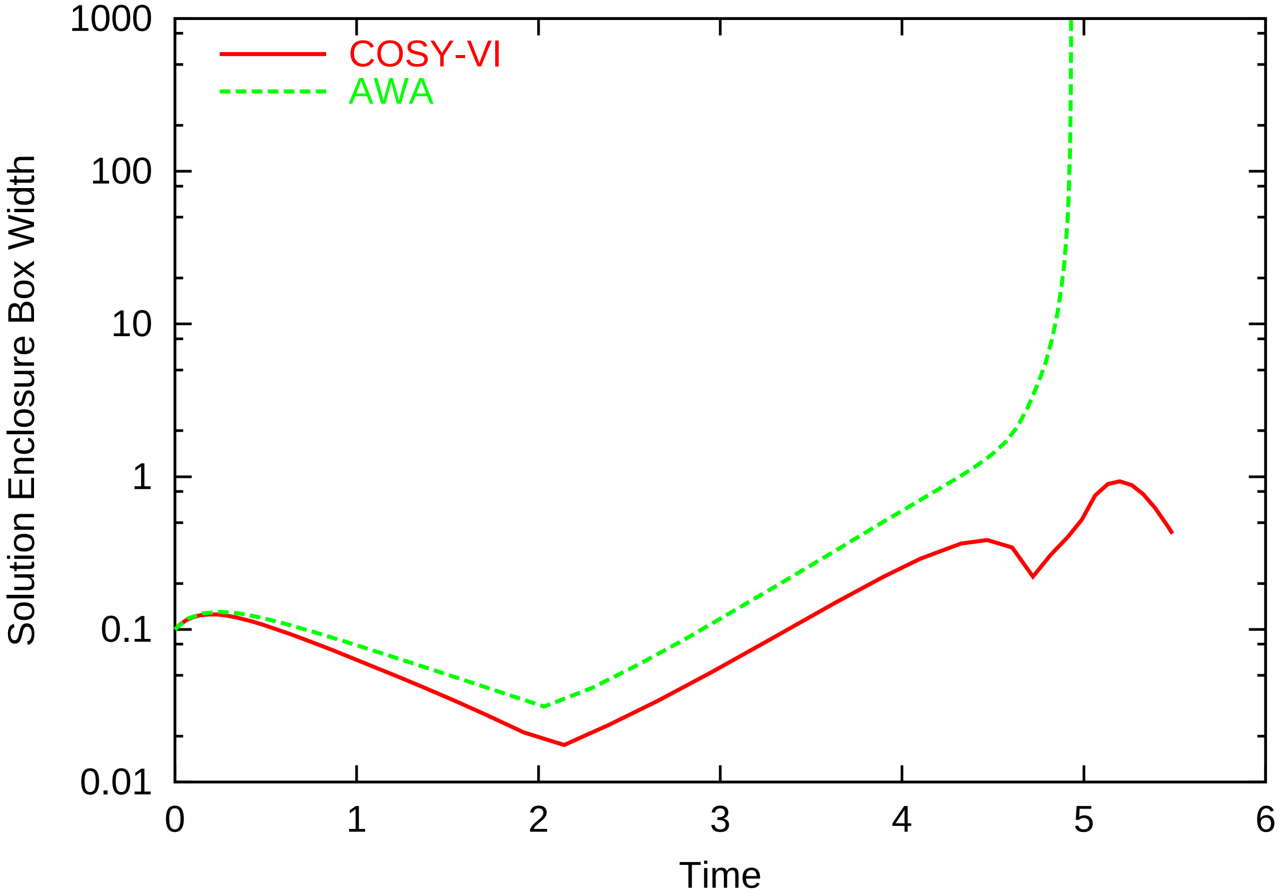

Integration of the Volterra eq. COSY-VI and AWA
AWA mode 4:
interval vector &
QR-decomposition
x2
x1
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
0.5
1
1.5
2
2.5
3
3.5



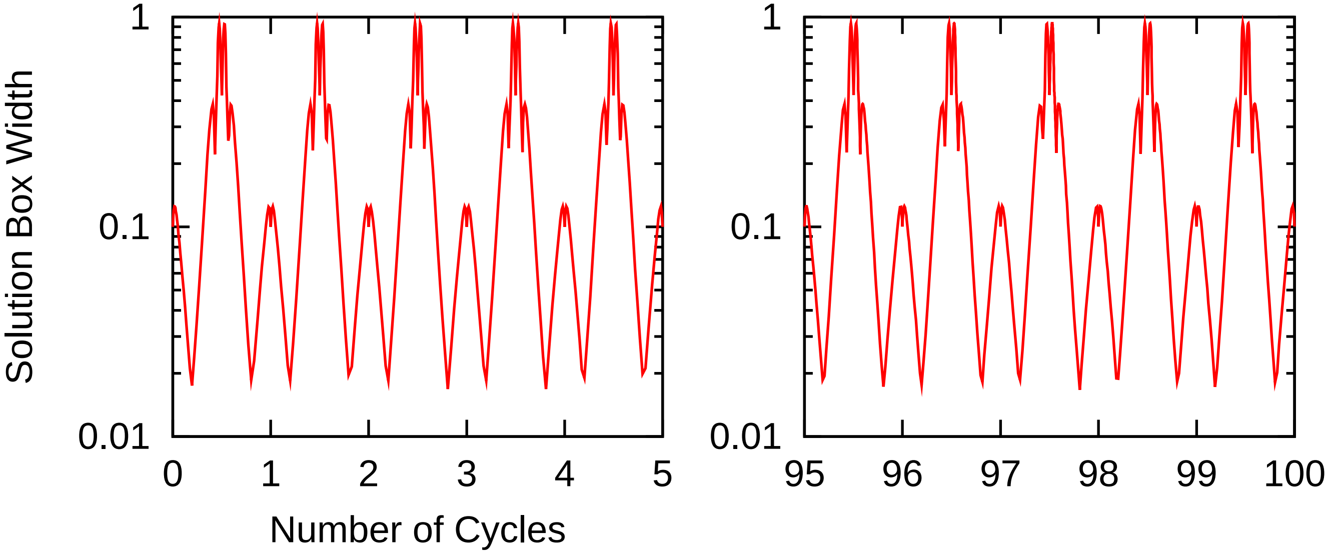


Volterra 18th, IC: P1, Result: Pn+{B(Pn+1 to P18)+IR}, same P


Volterra 18th, IC: P1, Result: Pn+{B(Pn+1 to P18)+IR}, same T
Remainder Error Size
1e-2
1e-4
1e-6
1e-8
1e-10
1e-12
1e-14
Order n
2
4
6
8
10
12
14
16
2 P
P
1/2 P
1/4 P
1/8 P


COSY-VI
AWA
Solution Enclosure Box Width
1000
100
10
1
0.1
0.01
0
1
2
3
4
5
6
Time

Solution Enclosing Box Width of Preconditioned TM Integration
during Forward-Backward Cycles of the Volterra Eq.
with Shrink Wrapping



Long-Term Behavior - Floating Point Case

Consider very simple two-state dynamical system:

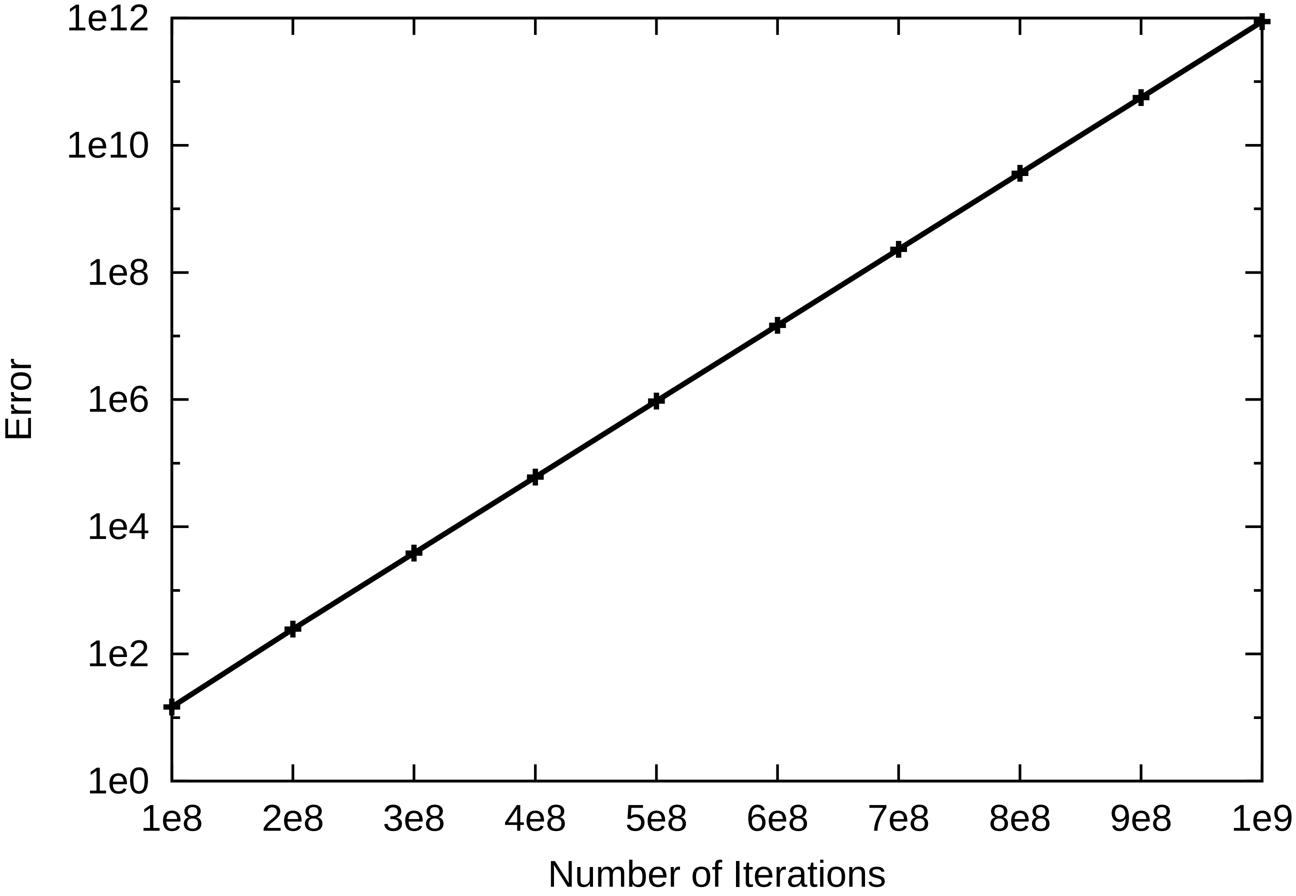
$$x_{n+1} = a \cdot x_n$$
$$x_{n+2} = (1/a) \cdot x_{n+1}$$

with initial condition $x_0 = 1$. Study the behavior for specific choices of a in both single and double precision arithmetic on

- F77 compiler by DEC, now distributed as f77 Digital Visual Fortran Version 5.0 as part of Microsoft Fortran PowerStation
- G77 compiler distributed by GNU; we specifically tested Version V0.5.24.

Choose $a_1 = 3$ for single precision, $a_2 = 0.9999999901608054$ for double precision

In both cases, we observe exponential growth of the error!


1e12
1e10
1e8
1e6
1e4
1e2
1e0
Error
1e8
2e8
3e8
4e8
5e8
6e8
7e8
8e8
9e8
1e9
Number of Iterations

Long-Term Behavior - Validated Case

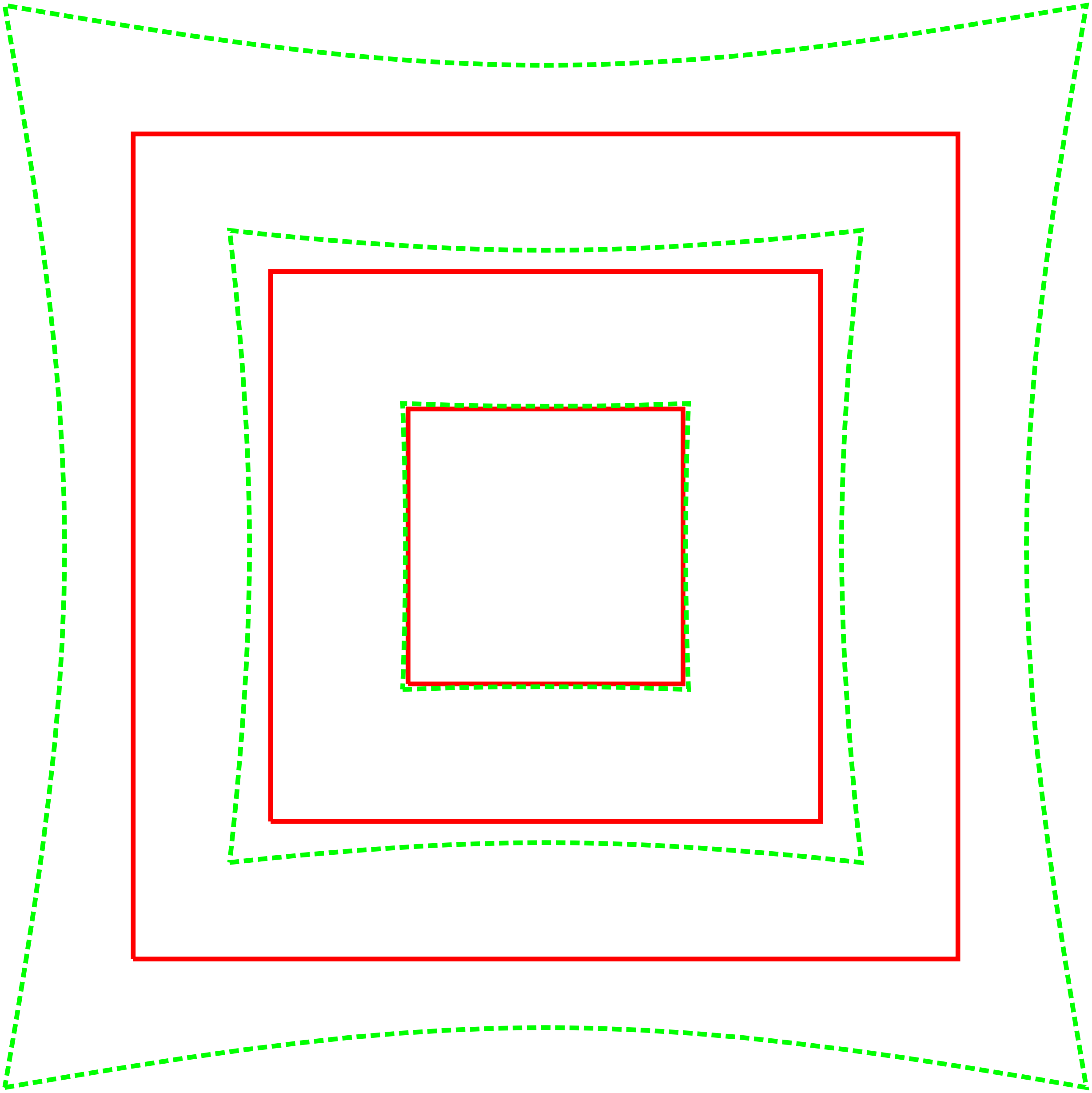
Consider very simple two-state dynamical system:

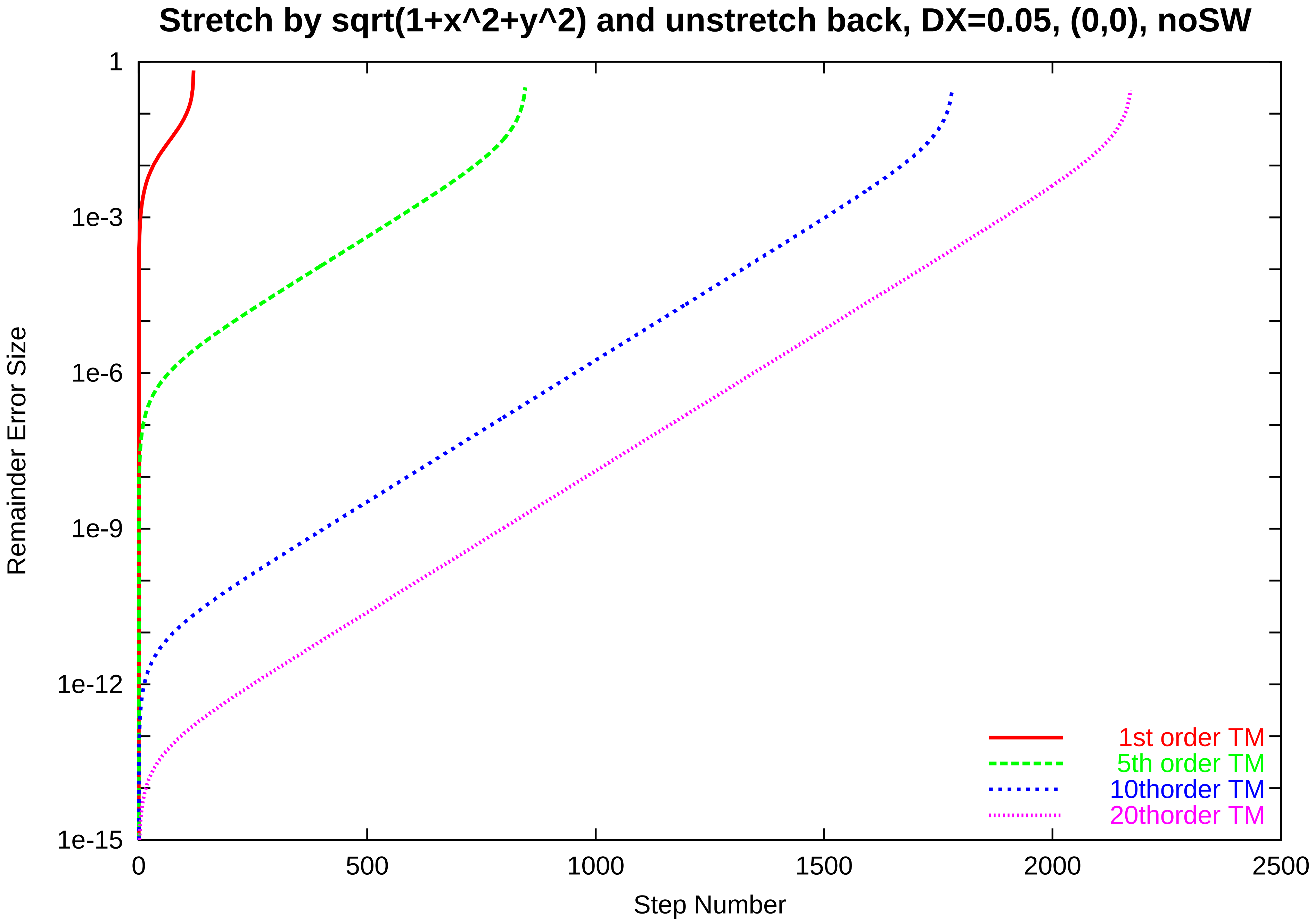
$$x_{n+1} = x_n \cdot \sqrt{1 + x_n^2 + y_n^2} \text{ and } y_{n+1} = y_n \cdot \sqrt{1 + x_n^2 + y_n^2}$$

$$x_{n+2} = x_{n+1} \cdot \sqrt{\frac{2}{1 + \sqrt{1 + 4(x_{n+1}^2 + y_{n+1}^2)}}} \text{ and}$$

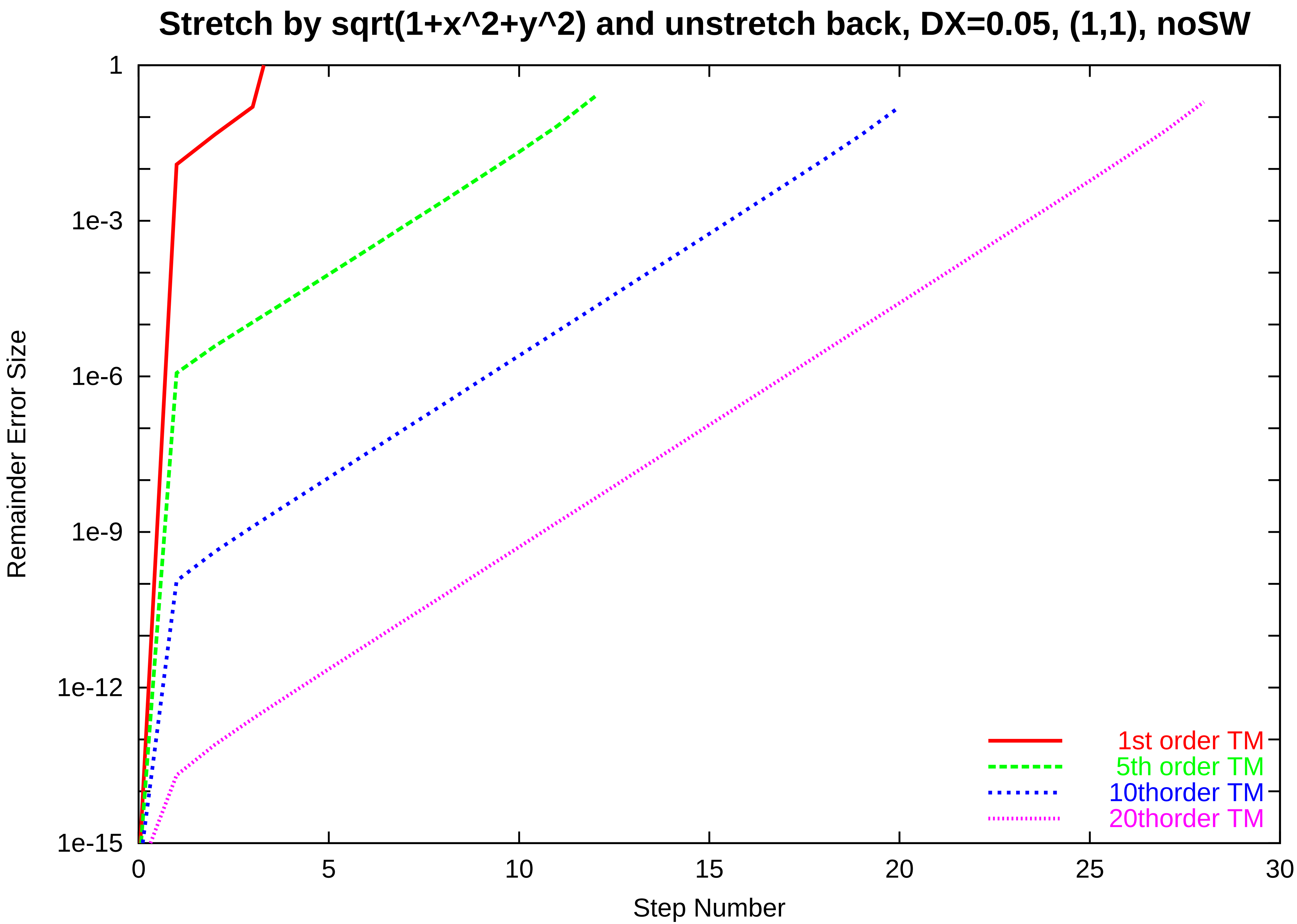
$$y_{n+2} = y_{n+1} \cdot \sqrt{\frac{2}{1 + \sqrt{1 + 4(x_{n+1}^2 + y_{n+1}^2)}}}.$$

Simple arithmetic shows that, also here we have $(x_{n+2}, y_{n+2}) = (x_n, y_n)$.





Stretch by sqrt(1+x^2+y^2) and unstretch back, DX=0.05, (0,0), noSW


Stretch by sqrt(1+x^2+y^2) and unstretch back, DX=0.05, (1,1), noSW
Remainder Error Size
1
1e-3
1e-6
1e-9
1e-12
1e-15
0
5
10
15
20
25
30
Step Number
1st order TM
5th order TM
10thorder TM
20thorder TM

Shrink Wrapping I

A method to remove the remainder bound of a Taylor model by increasing the polynomial part.

After the kth step of the integration, the region occupied by the final variables is given by

$$A = \vec{I}_0 + \bigcup_{\vec{x}_0 \in \vec{B}} \mathcal{M}_0(\vec{x}_0),$$

where $\vec{x}_0$ are the initial variables, $\vec{B}$ is the original box of initial conditions, $\mathcal{M}_0$ is the polynomial part of the Taylor model, and $\vec{I}_0$ is the remainder bound interval. $\mathcal{M}_0$ is scaled such that the original box $\vec{B}$ is unity, i.e. $\vec{B} = [-1, 1]^v$. $\vec{I}_0$ accounts for the local approximation error of the expansion in time carried out in the kth step as well as floating point errors and potentially other accumulated errors from previous steps; it is usually very small. Try to "absorb" the small remainder interval into a set very similar to the first part via

$$A \subset A^* = \vec{I}_0^* + \bigcup_{\vec{x}_0 \in \vec{B}} \mathcal{M}_0^*(\vec{x}_0),$$

where $\mathcal{M}_0^*$ is a slightly modified polynomial, and $\vec{I}_0^*$ is significantly reduced

Shrink Wrapping II

First, extract the constant part $\vec{a}_0$ and linear part $\hat{M}_0 \cdot \vec{x}$ of $\mathcal{M}_0$ and determine a floating point approximation $\bar{M}_0^{-1}$ of $\hat{M}_0$. If ODEs admits unique solutions, attempting to invert the linear transformation $\hat{M}_0$ in a floating point environement will very likely succeed.

After approximate inverse $\bar{M}_0^{-1}$ has been determined, apply linear transformation $\bar{M}_0^{-1} \cdot (\vec{x} - \vec{a}_0)$ from the left to the Taylor model $\mathcal{M}_0(\vec{x}_0) + \vec{I}_0$ that describes the current flow. As a result, the constant part of the resulting Taylor model now vanishes, and its linear part is near identity. We write the resulting Taylor model as

$$\mathcal{M} + \vec{I} = \mathcal{I} + \mathcal{S} + \vec{I},$$

where $\mathcal{I}$ is the identity, and the function $\mathcal{S}$ contains the nonlinear parts of the resulting Taylor model as well as some small linear corrections due to the error in inversion. We include $\vec{I}$ into the interval box $d \cdot [-1, 1]^v$, where d is a small number.

Shrink Wrapping III

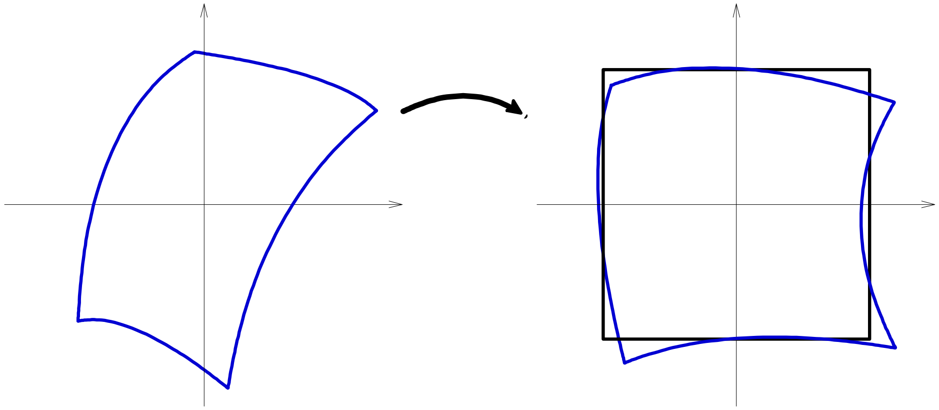


Figure 1: The region described by the Taylor model $\mathcal{M}_0 + \vec{I}_0$ is transformed to be normalized as $\mathcal{I} + \mathcal{S} + \vec{I}$, where $\mathcal{I}$ is the identity.

Definition (Shrinkability) Let $\mathcal{M} = \mathcal{I} + \mathcal{S} + \vec{I}$, where $\mathcal{S}$ is a polynomial and $\vec{I}$ is a small interval. We include $\vec{I}$ into the interval box $d \cdot [-1,1]^v$. We pick numbers s and t satisfying

$$s \geq |\mathcal{S}_i(\vec{x})| \ \forall\ \vec{x} \in B,\ 1 \leq i \leq v,$$
$$t \geq \left|\frac{\partial \mathcal{S}_i}{\partial x_j}\right| \ \forall\ \vec{x} \in B,\ 1 \leq i, j \leq v.$$

We call a map $\mathcal{M}$ shrinkable if $(1 - vt) > 0$ and $(1 - s) > 0$;

Shrink Wrapping IV

Then we define q, the so-called shrink wrap factor, as

$$q = 1 + d \cdot \frac{1}{(1-(v-1)t)\cdot(1-s)}.$$

The bounds s and t for the polynomials $\mathcal{S}_i$ and $\partial\mathcal{S}_i/\partial x_j$ can be computed by interval evaluation. The factor q will prove to be a factor by which the Taylor polynomial $\mathcal{I}+\mathcal{S}$ has to be multiplied in order to absorb the remainder bound interval.

Remark (Typical values for q) To put the various numbers in perspective, in the case of the verified integration of the Asteroid 1997 XF11, we typically have $d = 10^{-7}$, $s = 10^{-4}$, $t = 10^{-4}$, and thus $q \approx 1 + 10^{-7}$. It is interesting to note that the values for s and t are determined by the nonlinearity in the problem at hand, while in the absence of "noise" terms in the ODEs described by intervals, the value of d is determined mostly by the accuracy of the arithmetic. Rough estimates of the expected performance in quadruple precision arithmetic indicate that with an accompanying decrease in step size, if desired d can be decreased below 10^{-12}, resulting in $q \approx 1 + 10^{-12}$.

Shrink Wrapping V

In order to proceed, we need some estimates relating image distances to origin distances.

Lemma. Let $\mathcal{M}$ be a map as above, let $\|\cdot\|$ denote the max norm, and let $(1 - vt) > 0$. Then we have

$$\begin{aligned} \left|\mathcal{M}_i(\vec{\bar{x}}) - \mathcal{M}_i(\vec{x})\right| &\leq \sum_j |\delta_{i,j} + t| \;\; |\bar{x}_j - x_j| \,, \\ \left\|\mathcal{M}(\vec{\bar{x}}) - \mathcal{M}(\vec{x})\right\| &\leq (1 + vt) \cdot \left\|\vec{\bar{x}} - \vec{x}\right\| \,, \text{ and} \\ \left\|\mathcal{M}(\vec{\bar{x}}) - \mathcal{M}(\vec{x})\right\| &\geq (1 - vt) \cdot \left\|\vec{\bar{x}} - \vec{x}\right\| \,. \end{aligned}$$

Proof. For the proof of the first assertion, we observe that all $(v - 1)$ partials of $\partial\mathcal{M}_i/\partial x_j$ for $j \neq i$ are bounded in magnitude by t, while $\partial\mathcal{M}_i/\partial x_i$ is bounded in magnitude by $1+t$; thus the first statement follows from the intermediate value theorem. For the second assertion, we trivially observe

$$\begin{aligned}\left\|\mathcal{M}(\vec{\bar{x}}) - \mathcal{M}(\vec{x})\right\| &= \max_i \left|\mathcal{M}_i(\vec{\bar{x}}) - \mathcal{M}_i(\vec{x})\right| \\ &\leq \max_i \sum_j |\delta_{i,j} + t| \;\; |\bar{x}_j - x_j| \\ &\leq (1 + vt) \;\; \left\|\vec{\bar{x}} - \vec{x}\right\|.\end{aligned}$$

For the proof of the third assertion, which is more involved, let k be such that $\left\|\vec{\bar{x}} - \vec{x}\right\| = |\bar{x}_k - x_k|$, and wlog let $\bar{x}_k - x_k > 0$. Then we have

$$\begin{aligned}\left\|\mathcal{M}(\vec{\bar{x}}) - \mathcal{M}(\vec{x})\right\| &= \max_i \left|\mathcal{M}_i(\vec{\bar{x}}) - \mathcal{M}_i(\vec{x})\right| \\ &\geq \left|\mathcal{M}_k(\vec{\bar{x}}) - \mathcal{M}_k(\vec{x})\right| \\ &= \left|(1 + c_k)(\bar{x}_k - x_k) + \sum_{j \neq k} c_j(\bar{x}_j - x_j)\right| \qquad (1)\end{aligned}$$

for some set of c_j with $|c_j| \leq t \; \forall j = 1, ..., v$, according to the mean value theorem. Now observe that for any such set of c_j,

$$\begin{aligned}\left|\sum_{j\neq k} c_j(\bar{x}_j - x_j)\right| &\leq \sum_{j\neq k} |c_j| \;\; |\bar{x}_j - x_j| \leq \left(\sum_{j\neq k} |c_j|\right) \;\; |\bar{x}_k - x_k| \\ &\leq (v-1)\; t\; |\bar{x}_k - x_k| \\ &\leq (1-t)\; |\bar{x}_k - x_k| \leq (1+c_k)\,(\bar{x}_k - x_k)\,.\end{aligned}$$

Hence the left term in the right hand absolute value in (1) dominates the right term for any set of c_j, and we thus have

$$\begin{aligned}&\left|(1+c_k)(\bar{x}_k - x_k) + \sum_{j\neq k} c_j(\bar{x}_j - x_j)\right| \\ &\geq (1-t)(\bar{x}_k - x_k) - \sum_{j\neq k} t \;\; |\bar{x}_j - x_j| \\ &\geq (1-t)(\bar{x}_k - x_k) - (v-1)\; t\; (\bar{x}_k - x_k) \\ &= (1-vt)(\bar{x}_k - x_k) = (1-vt)\;\; \left\|\vec{\bar{x}} - \vec{x}\right\|,\end{aligned}$$

which completes the proof.

Shrink Wrapping VI

Theorem (Shrink Wrapping) Let $\mathcal{M} = \mathcal{I} + \mathcal{S}(\vec{x})$, where $\mathcal{I}$ is the identity. Let $\vec{I} = d \cdot [-1, 1]^v$, and

$$R = \vec{I} + \bigcup_{\vec{x} \in \vec{B}} \mathcal{M}(\vec{x})$$

be the set sum of the interval $\vec{I} = [-d, d]^v$ and the range of $\mathcal{M}$ over the original domain box $\vec{B}$. So R is the range enclosure of the flow of the ODE over the interval $\vec{B}$ provided by the Taylor model. Let q be the shrink wrap factor of $\mathcal{M}$; then we have

$$R \subset \bigcup_{\vec{x} \in \vec{B}} (q\mathcal{M})(\vec{x}),$$

and hence multiplying $\mathcal{M}$ with the number q allows to set the remainder bound to zero.

Proof. Let $1 \leq i \leq v$ be given. We note that because $\partial \mathcal{M}_i / \partial x_i > 1 - t > 0$, $\mathcal{M}_i$ increases monotonically with x_i. Consider now the $(v-1)$ dimensional surface set $(x_1, ..., x_v)$ with $x_i = 1$ fixed. Pick a set of $x_j \in [-1, 1]$, $j \neq i$. We want to study how far the set $R = \vec{I} + \bigcup_{\vec{x} \in \vec{B}} \mathcal{M}(\vec{x})$ can extend beyond the surface in direction i at the surface point $\vec{y} = \mathcal{M}(x_1, ..., x_{i-1}, 1, x_{i+1}, ..., x_v)$.

Let y_i be the i-th component of $\vec{y}$. The i-th components of the set $\vec{y}+\vec{I}$ apparently extend beyond y_i by d. However, it is obvious that R can extend further than that beyond y_i. In fact, for any other $\vec{\bar{y}}$ with $|\bar{y}_j - y_j| \leq d$ for $j \neq i$, there are points in $\vec{\bar{y}}+\vec{I}$ with all but the i-th component equal to those of $\vec{y}$. On the other hand, any $\vec{\bar{y}}$ with $|\bar{y}_j - y_j| > d$ for some $j \neq i$ can not have a point in $\vec{\bar{y}}+\vec{I}$ with all but the i-th component matching those of $\vec{y}$. So at the point y_i, the set R can extend to

$$r_i(\vec{y}) = d + \sup_{\{\vec{\bar{y}}|\ |\bar{y}_j - y_j| \leq d\ \ (j \neq i)\}} \bar{y}_i.$$

We shall now find a bound for $r_i(\vec{y})$. First we observe that because of the monotonicity of $\mathcal{M}_i$, we can restrict the search to the case with $x_i = 1$. We now project to an $(v-1)$ dimensional subspace by fixing $x_i = 1$ and by removing the i-th component $\mathcal{M}_i$. We denote the resulting map by $\mathcal{M}^{(i)}$, and similarly denote all $(v-1)$ dimensional variables with the superscript "(i)".

We observe that with the function $\mathcal{M}$, also the function $\mathcal{M}^{(i)}$ is shrinkable according to the definition, with factors s and t inherited from $\mathcal{M}$. Apparently the condition on $\vec{\bar{y}}$ in the definition of $r_i(\vec{y})$ entails that in the $(v-1)$ dimensional subspace, $\left\|\vec{\bar{y}}^{(i)} - \vec{y}^{(i)}\right\| \leq d$. Let $\vec{\bar{x}}^{(i)}$ and $\vec{x}^{(i)}$ be the $(v-1)$ di-

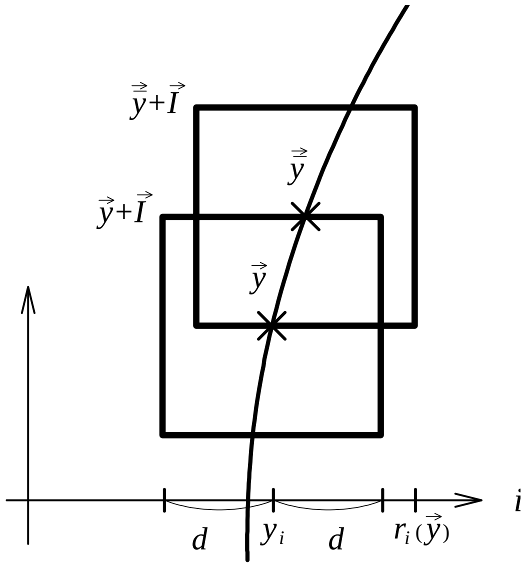


Figure 2: At the point y_i, the set $R = \vec{I} + \bigcup_{\vec{x}\in\vec{B}} \mathcal{M}(\vec{x})$ can extend to $r_i(\vec{y})$.

mensional pre-images of $\vec{\bar{y}}^{(i)}$ and $\vec{y}^{(i)}$, respectively; because $\left\|\vec{\bar{y}}^{(i)} - \vec{y}^{(i)}\right\| \leq d$, we have according to the above lemma that

$$\left\|\vec{\bar{x}}^{(i)} - \vec{x}^{(i)}\right\| \leq \frac{d}{1-(v-1)t},$$

which entails that also in the original space we have $|\bar{x}_j - x_j| \leq d/(1-(v-1)t)$ for $j \neq i$. Hence we can bound $r_i(\vec{y})$ via

$$r_i(\vec{y}) \leq d + \sup_{\substack{\{\vec{\bar{x}} |\ |\bar{x}_j - x_j| \leq d/(1-(v-1)t) \\ (j \neq i),\ x_i = \bar{x}_i = 1\}}} \mathcal{M}_i(\vec{\bar{x}}).$$

We now invoke the first statement of the lemma for the case of $\vec{\bar{x}}$, $\vec{x}$ satisfying $|\bar{x}_j - x_j| \leq d/(1-(v-1)t)$ $(j \neq i)$, $x_i = \bar{x}_i = 1$. The last condition implies that the term involving $(\delta_{i,j} + t)$ does not contribute, and we thus have $|\mathcal{M}_i(\vec{\bar{x}}) - \mathcal{M}_i(\vec{x})| \leq (v-1)t \cdot d/(1-(v-1)t)$, and altogether

$$\begin{aligned} r_i(\vec{y}) &\leq y_i + d + \frac{d \cdot (v-1)t}{1-(v-1)t} \\ &= y_i + d \cdot \frac{1}{1-(v-1)t}. \end{aligned}$$

We observe that the second term in the last expression is independent of i. Hence we have shown that the "band" around $\bigcup_{\vec{x} \in \vec{B}} \mathcal{M}(\vec{x})$ generated by the addition of $\vec{I}$ never extends more than $d/(1-(v-1)t)$ in any direction.

To complete the proof, we observe that because of the bound s on $\mathcal{S}$, the box $(1-s)[-1,1]^v$ lies entirely in the range of $\mathcal{M}$. Thus multiplying the map $\mathcal{M}$ with any factor $q > 1$ entails that the edges of the box $(1-s)[-1,1]^v$ move out by the amount $(1-s)(q-1)$ in all directions. Since the box is entirely inside the range of $\mathcal{M}$, this also means that the border of the range of $\mathcal{M}$ moves out by at least the same amount in any direction i. Thus choosing q as

$$q = 1 + d \cdot \frac{1}{(1-(v-1)t)\cdot(1-s)}$$

assures that

$$\bigcup_{\vec{x}\in\vec{B}} (q\mathcal{M}) \supset R$$

as advertised.

Shrink Wrapping VII

Let us consider the practical limiations of the method; apparently the measures of the nonlinearities s and t must not become too large

Remark (Limitations of shrink wrapping) Apparently the shrink wrap method discussed above has the following limitations

Remark 1 *1. The measures of nonlinearities s and t must not become too large*

2. The application of the inverse of the linear part should not lead to large increases in the size of remainder bounds.

Apparently the first requirement limits the domain size that can be covered by the Taylor model, and it will thus happen only in extreme cases. Furthermore, in practice the case of s and t becoming large is connected to also having accumulated a large remainder bound, since the remainder bounds are calculated from the bounds of the various orders of s. In the light of this, not much additional harm is done by removing the offending s into the remainder bound and create a linearized Taylor model.

Definition (Blunting of an Ill-Conditioned Matrix)

Let $\hat{A}$ be a regular matrix that is potentially ill-conditioned and $\vec{q} = (q_1, ...q_n)$ a vector with $q_i > 0$. Arrange the column vectors $\vec{a}_i$ of $\hat{A}$ by size.

Let $\vec{e}_i$ be the familiar orthonormal vectors obtained through the Gram-Schmidt procedure, i.e.

$$\vec{e}_i = \frac{\vec{a}_i - \sum_{k=1}^{i-1} \vec{e}_k \ \left(\vec{a}_i \cdot \vec{e}_k\right)}{\left|\vec{a}_i - \sum_{k=1}^{i-1} \vec{e}_k \ \left(\vec{a}_i \cdot \vec{e}_k\right)\right|}.$$

We form vectors $\vec{b}_i$ via

$$\vec{b}_i = \vec{a}_i + q_i \vec{e}_i$$

and assemble them columnwise into the matrix $\hat{B}$. We call $\hat{B}$ the $\vec{q}$-blunted matrix belonging to $\hat{A}$

Proposition (Regularity of the Blunted Matrix) The $\vec{b}_i$ are linearly independent and thus $\hat{B}$ is regular.

Proof. By induction. Apparently $\vec{b}_1$ is linearly independent. Assume now that $\vec{b}_1, ..., \vec{b}_{i-1}$ are linearly independent. We first observe that for each i, the vector $\vec{b}_i$ is a linear combination of the $\vec{a}_k$ for $k = 1, ..., i$ and thus also of the $\vec{e}_k$ for $k = 1, ..., i$, since both the $\vec{a}_k$ and the $\vec{e}_k$ span the same space. Now assume $\vec{b}_i$ is linearly dependent on $\vec{b}_1, ..., \vec{b}_{i-1}$; then it is also linearly dependent on $\vec{e}_1, ..., \vec{e}_{i-1}$, i.e. there are $\lambda_1, ..., \lambda_{i-1}$ such that

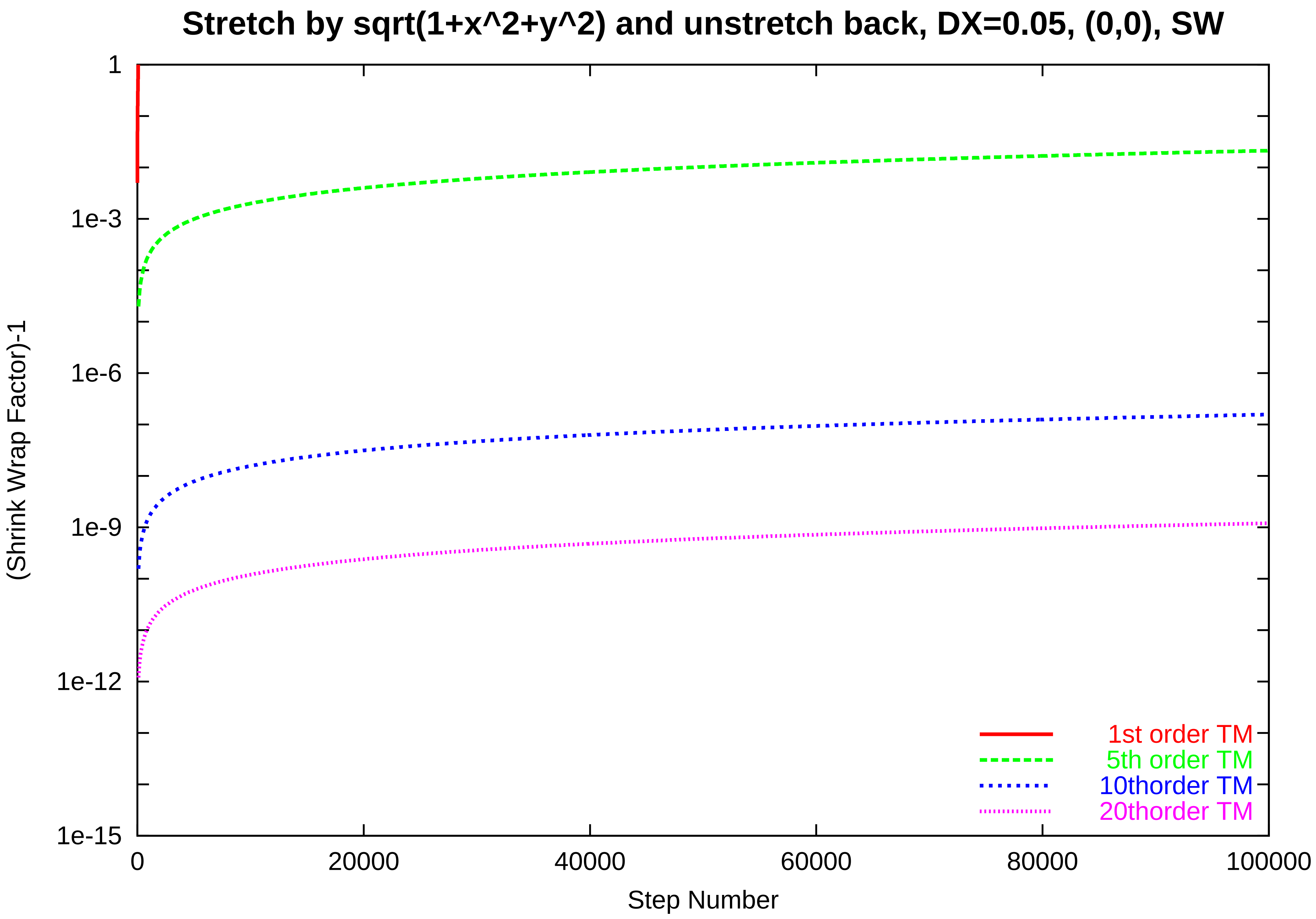
$$\vec{b}_i = \sum_{k=1}^{i-1} \lambda_k \vec{e}_k.$$

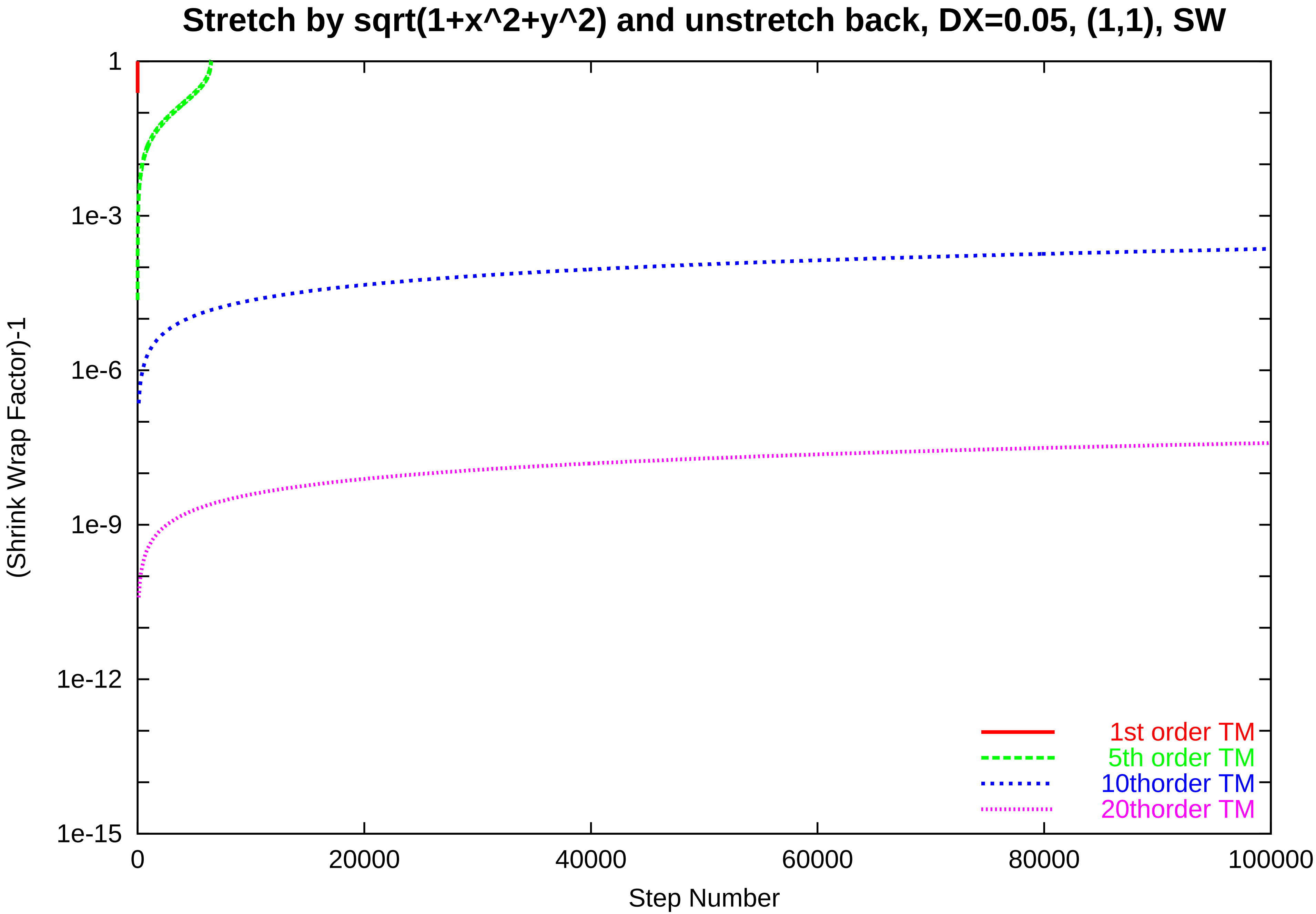
But because $\vec{b}_i = \vec{a}_i + q_i \vec{e}_i$, we have

$$\vec{a}_i \left(1 + \frac{q_i}{\left| \vec{a}_i - \sum_{k=1}^{i-1} \vec{e}_k \left(\vec{a}_i \cdot \vec{e}_k \right) \right|} \right) = \sum_{k=1}^{i-1} \left(\lambda_k + \vec{a}_i \cdot \vec{e}_k \right) \; \vec{e}_k$$

Since by requirement, $q_i > 0$, the factor of $\vec{a}_i$ is nonzero, and we have a contradiction to the linear independence of $\vec{a}_i$ from $\vec{e}_1, ..., \vec{e}_{i-1}$. Thus $\vec{b}_1, ..., \vec{b}_i$ are linearly independent.

Intuitively, of course, the effect of blunting is that each vector $\vec{b}_i$ is being "pulled away" from the space spanned by the previous vectors $\vec{b}_1, ..., \vec{b}_{i-1}$, and more strongly so if q_i becomes bigger and bigger. In fact, we have the following result: .


Stretch by sqrt(1+x^2+y^2) and unstretch back, DX=0.05, (0,0), SW
(Shrink Wrap Factor)-1
1
1e-3
1e-6
1e-9
1e-12
1e-15
0
20000
40000
60000
80000
100000
Step Number
1st order TM
5th order TM
10thorder TM
20thorder TM


Stretch by sqrt(1+x^2+y^2) and unstretch back, DX=0.05, (1,1), SW
(Shrink Wrap Factor)-1
1
1e-3
1e-6
1e-9
1e-12
1e-15
0
20000
40000
60000
80000
100000
Step Number
1st order TM
5th order TM
10thorder TM
20thorder TM

Preconditioning the Flow

It can be viewed as a coordinate transformation.

Definition (Preconditioning the Flow) Let $(P + I)$ be a Taylor model. We say that $(P_l + I_l), (P_r + I_r)$ is a factorization of $(P + I)$ if $B(P_r + I_r) \in [-1, 1]$ and

$$(P + I) \in (P_l + I_l) \circ (P_r + I_r) \text{ for all } x \in D$$

where D is the domain of the Taylor model $(P_r + I_r)$.

Proposition Let $(P_{l,n} + I_{l,n}) \circ (P_{r.n} + I_{r,n})$ be a factored Taylor model that encloses the flow of the ODE at time t_n. Let $(P^*_{l,n+1}, I^*_{l,n+1})$ be the result of integrating $(P_{l,n} + I_{l,n})$ from t_n to t_{n+1}. Then

$$(P^*_{l,n+1}, I^*_{l,n+1}) \circ (P_{r.n} + I_{r,n})$$

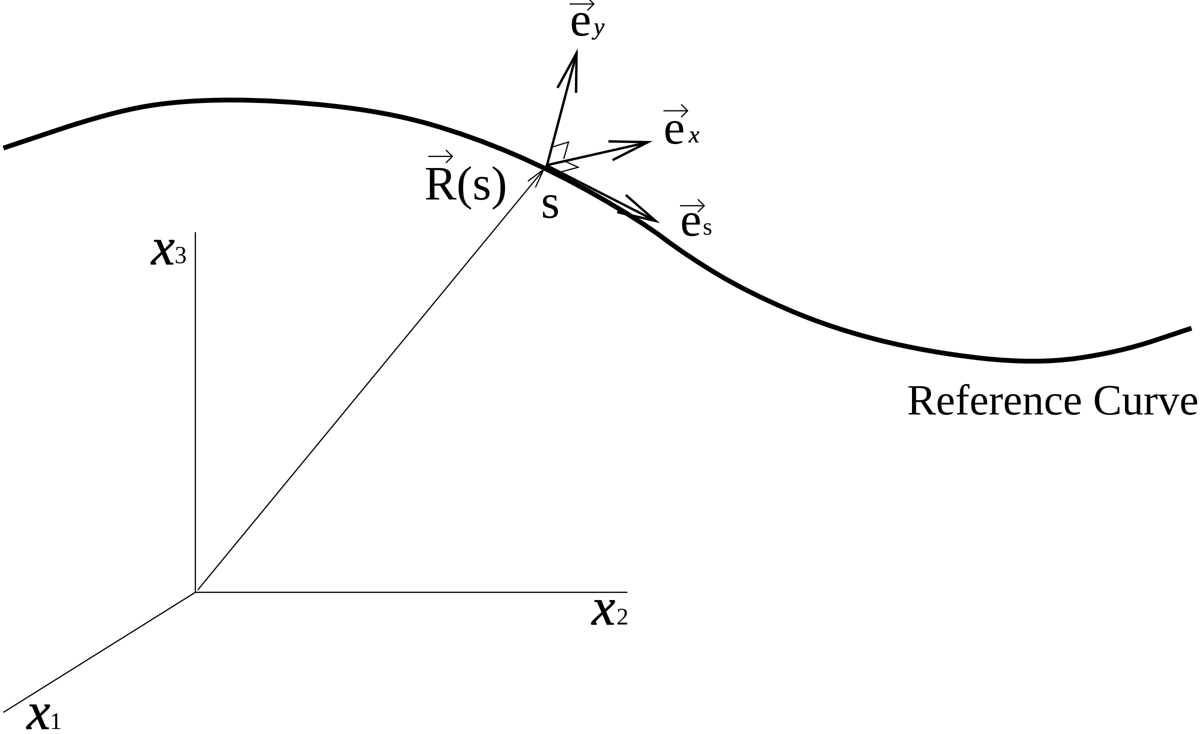
is a factorization of the flow at time t_{n+1}.

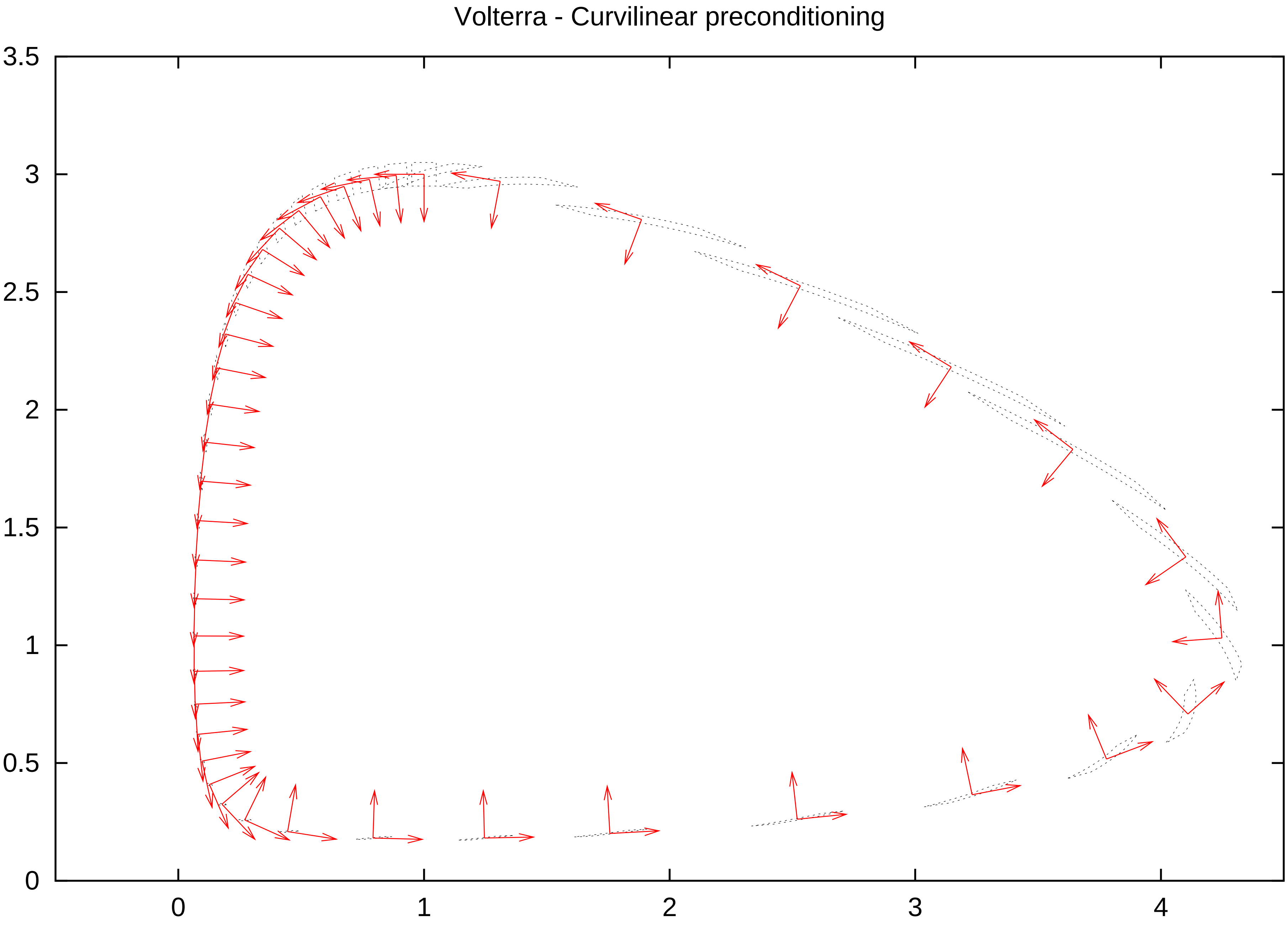
Example Preconditionings: QR, Blunted, Curvilinear.

Curvilinear Preconditioning

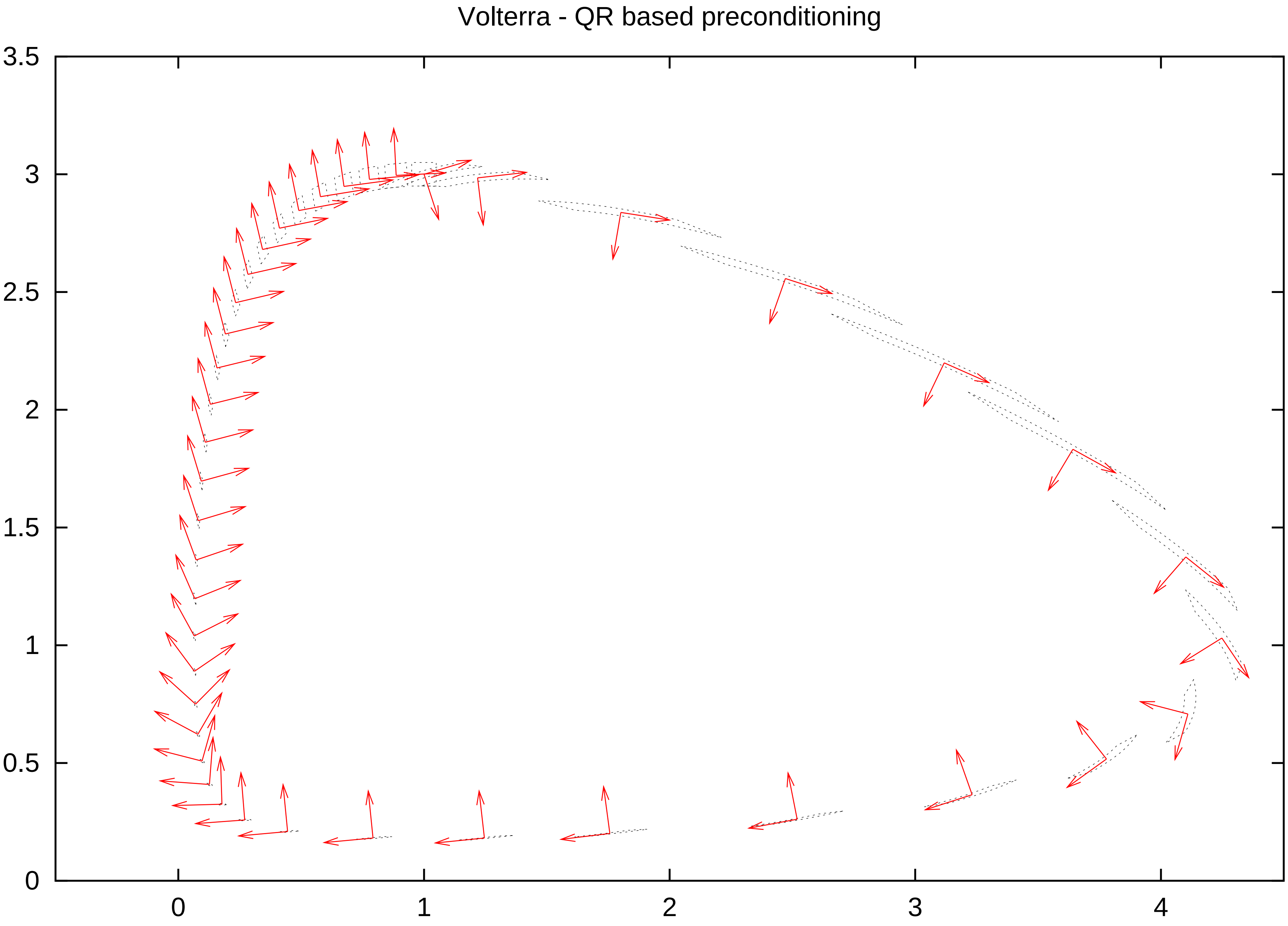
Definition (Curvilinear Preconditioning) Let $x^{(m)} = f(x, x', ...x^{(m-1)}, t)$ be an m-th order ODE in n variables. Let $x_r(t)$ be a solution of the ODE and $x'_r(t)$, ..., $x_r^{(k)}(t)$ its first k time derivatives. Let $\vec{e}_1, ..., \vec{e}_l$ be the l unit vectors not in the span of $x'_r(t)$, ..., $x_r^{(k)}(t)$, sorted by distance from the span. Then we call the Gram-Schmidt orthonormalization of the set ($x'_r(t)$, ..., $x_r^{(k)}(t), \vec{e}_1, ..., \vec{e}_l$) the curvilinear basis of depth k.

Example (Solar System and Particle Accelerators)
In this case, $n = 3$, and one usually chooses $k = 2$.
(1) The first basis vector points in the direction of reference orbit motion.
(2) The second is perpendicular to it and points approximately to the sun or the center of the accelerator.
(3) The third is chosen perpendicular to the plane of the previous two.


$\vec{e}_y$
$\vec{e}_x$
$\vec{R}(s)$
s
$\vec{e}_s$
x_3
Reference Curve
x_2
x_1



Volterra - Curvilinear preconditioning
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4


Volterra - QR based preconditioning
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4

Blunting

Definition (Blunting of an Ill-Conditioned Matrix)
Let $\hat{A}$ be a regular matrix that is potentially ill-conditioned and $\vec{q} = (q_1, ...q_n)$ with $q_i > 0$. Arrange the column vectors $\vec{a}_i$ of $\hat{A}$ by size. Let $\vec{e}_i$ be the orthonormal vectors obtained through the Gram-Schmidt procedure;

$$\vec{e}_i = \frac{\vec{a}_i - \sum\limits_{k=1}^{i-1} \vec{e}_k \ \left(\vec{a}_i \cdot \vec{e}_k\right)}{\left|\vec{a}_i - \sum\limits_{k=1}^{i-1} \vec{e}_k \ \left(\vec{a}_i \cdot \vec{e}_k\right)\right|}.$$

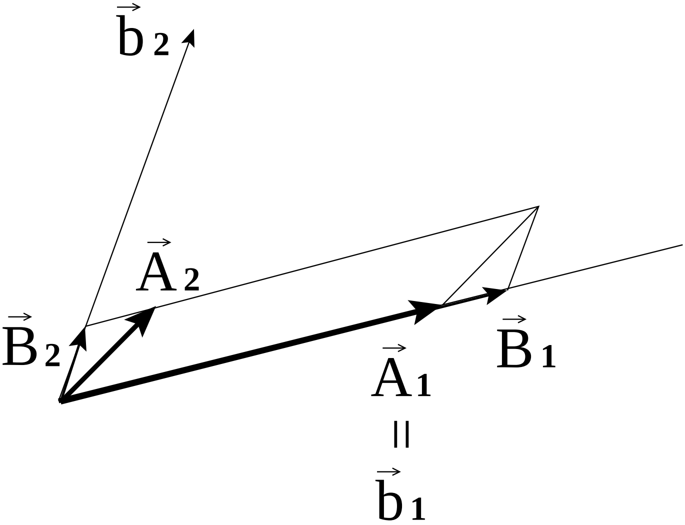
We form vectors $\vec{b}_i$ via

$$\vec{b}_i = \vec{a}_i + q_i \vec{e}_i$$

and assemble them columnwise into the matrix $\hat{B}$. We call $\hat{B}$ the $\vec{q}$-blunted matrix belonging to $\hat{A}$

Proposition (Regularity of the Blunted Matrix) The $\vec{b}_i$ are linearly independent and thus $\hat{B}$ is regular.

The effect of blunting is that each vector $\vec{b}_i$ is being "pulled away" from the space spanned by the previous vectors $\vec{b}_1, ..., \vec{b}_{i-1}$, and more strongly so if q_i becomes bigger and bigger.


$\vec{b}_2$
$\vec{A}_2$
$\vec{B}_2$
$\vec{B}_1$
$\vec{A}_1$
||
$\vec{b}_1$

Random Matrices - Discrete

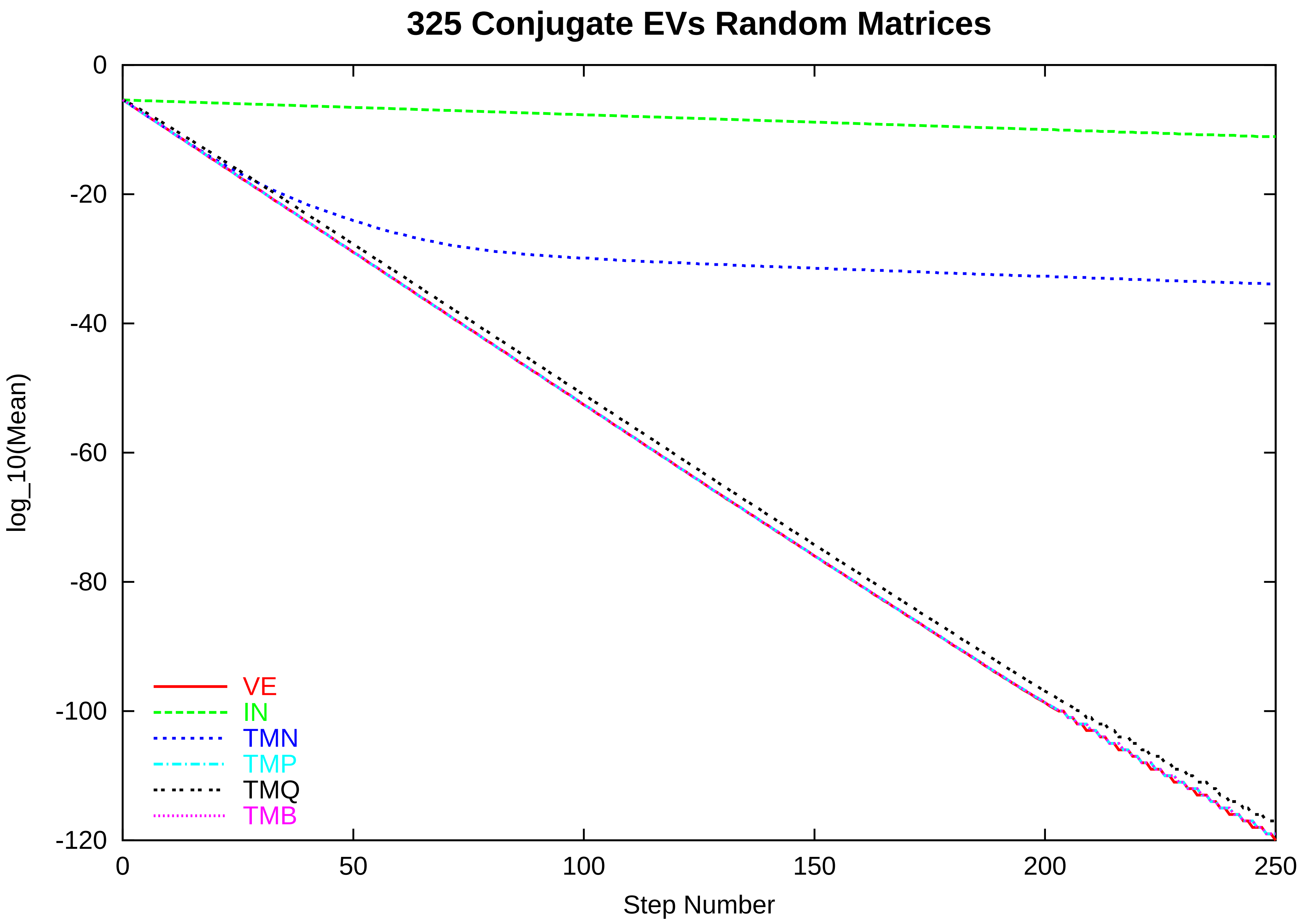
Select 1000 twodimensional random matrices with coefficients in $[-1, 1]$.Sort according to eigenvalues into seven sub-cases.

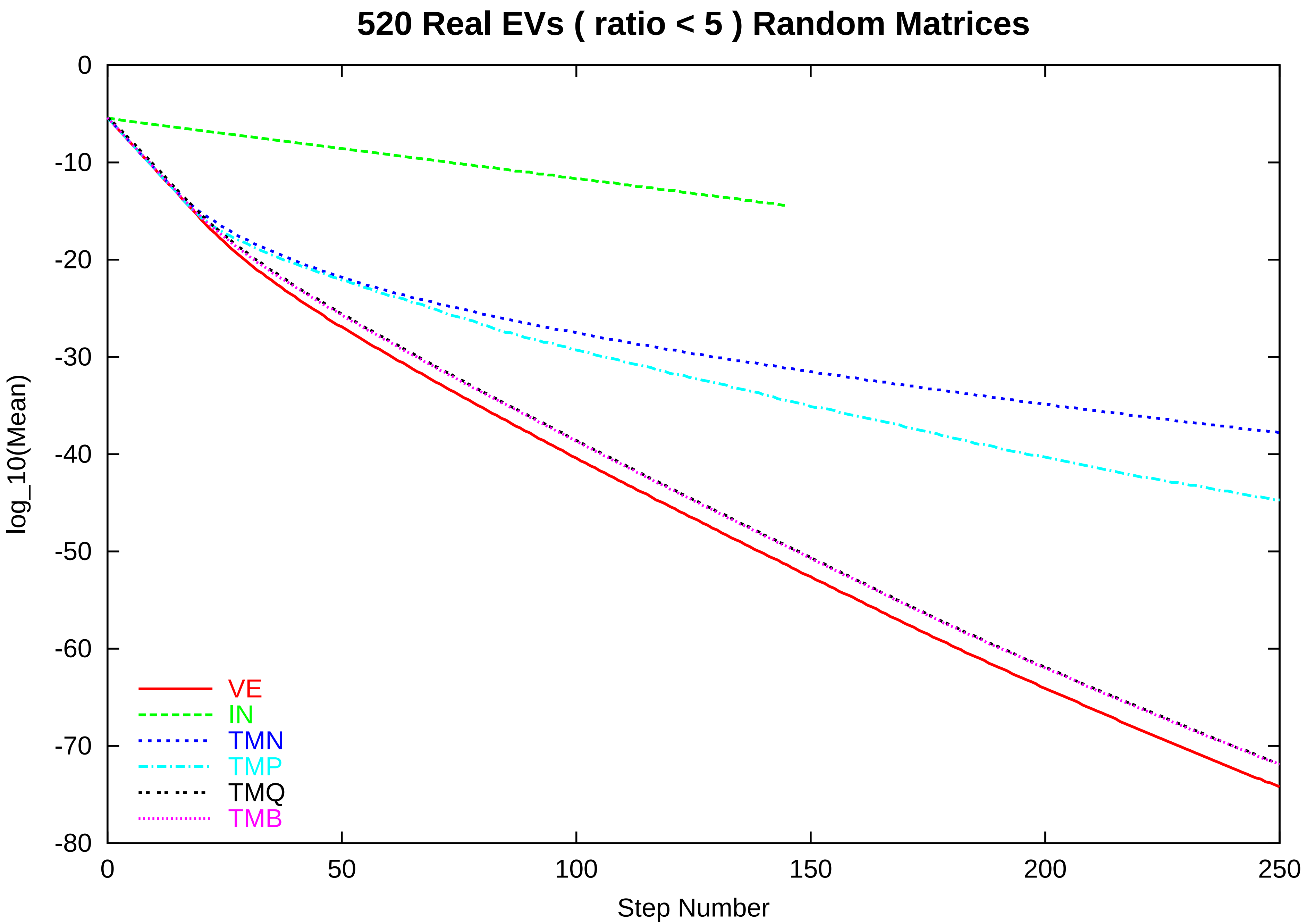
Perform iteration in the following ways:

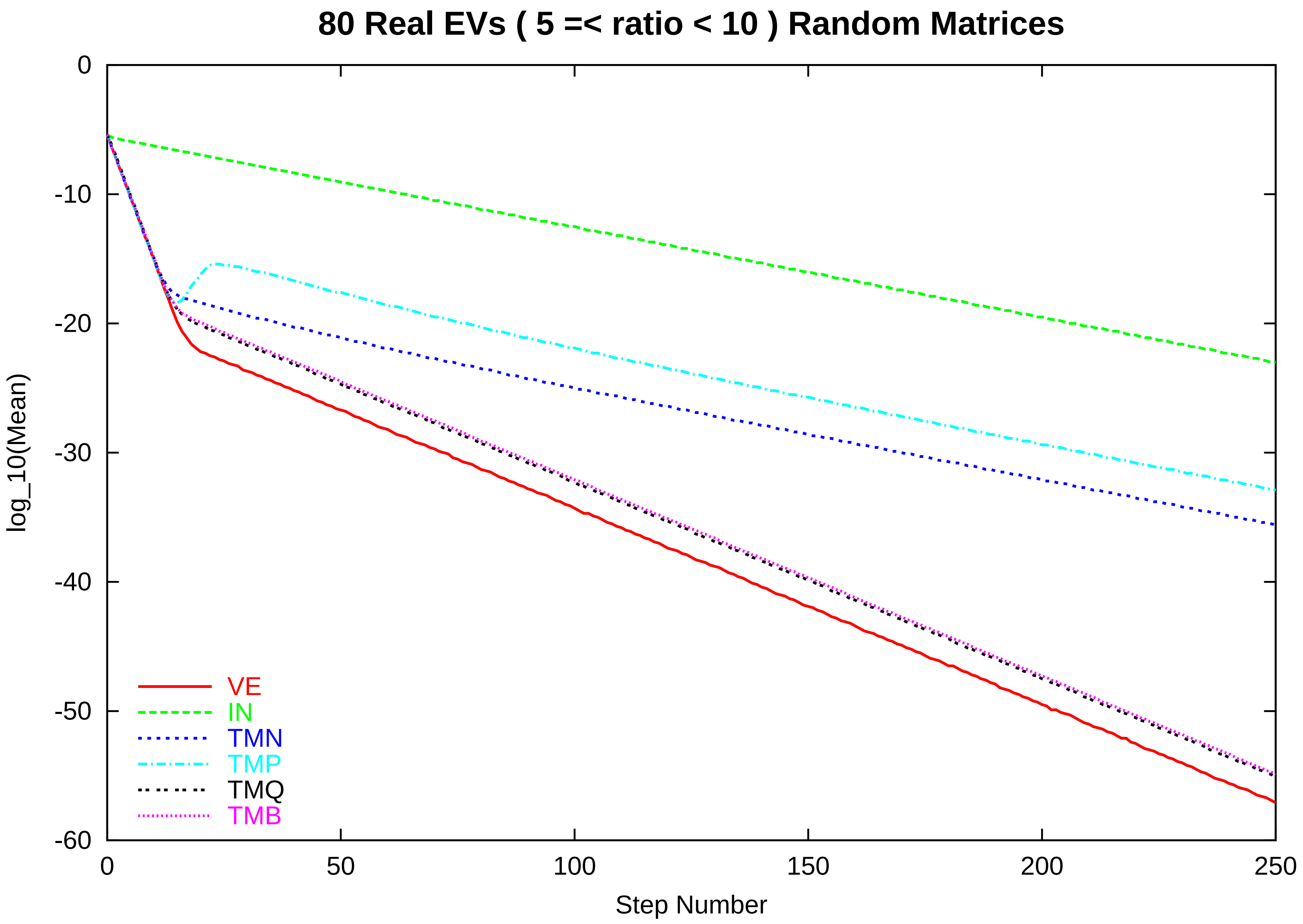
- Naive Interval
- Naive Taylormodel
- Parallelepiped-preconditioned Taylormodel
- QR-preconditioned Taylormodel
- Blunted preconditioned TM, various blunting factors
- Set of four floating point corner points for volume estimation

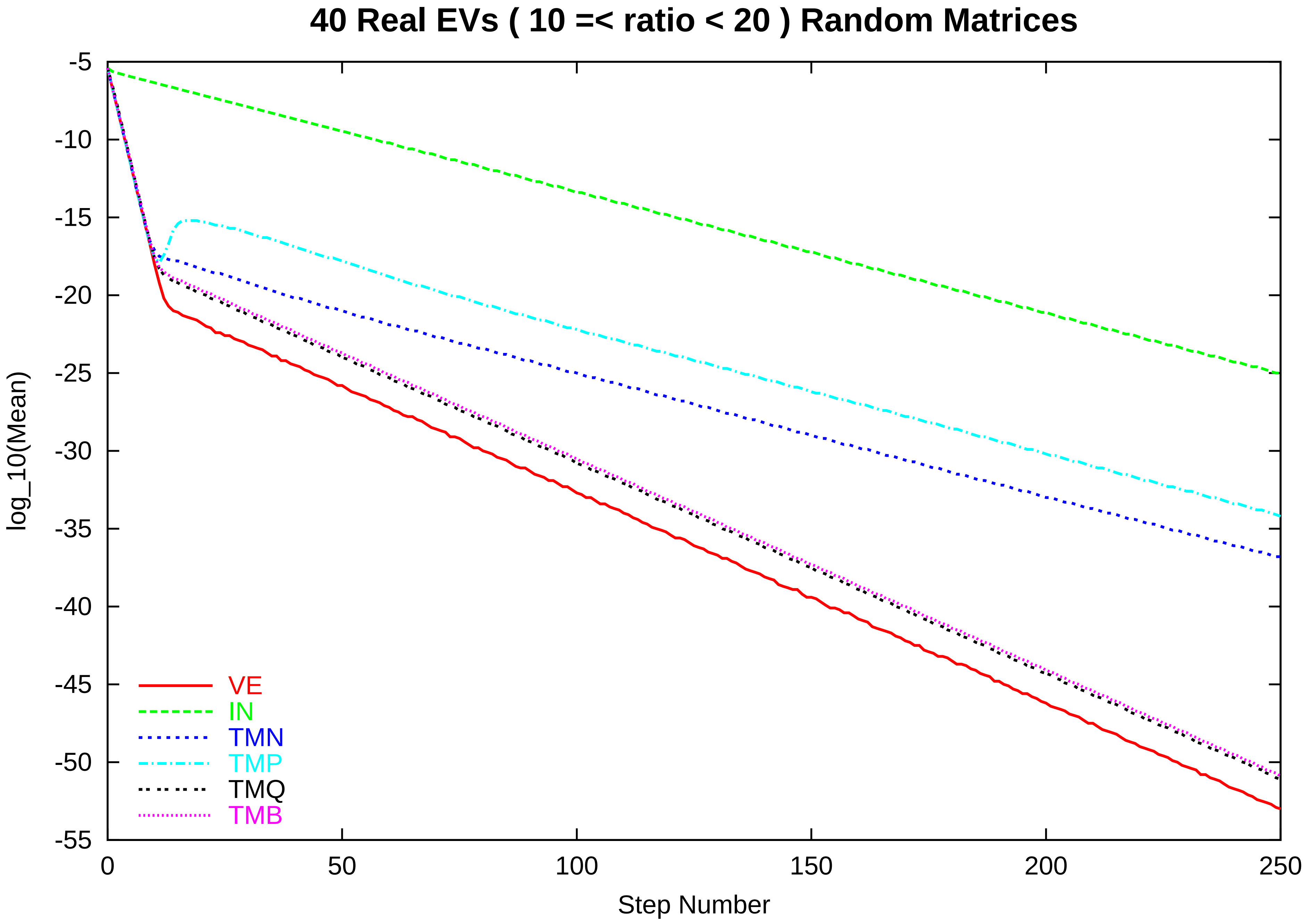
Perform the following tasks:

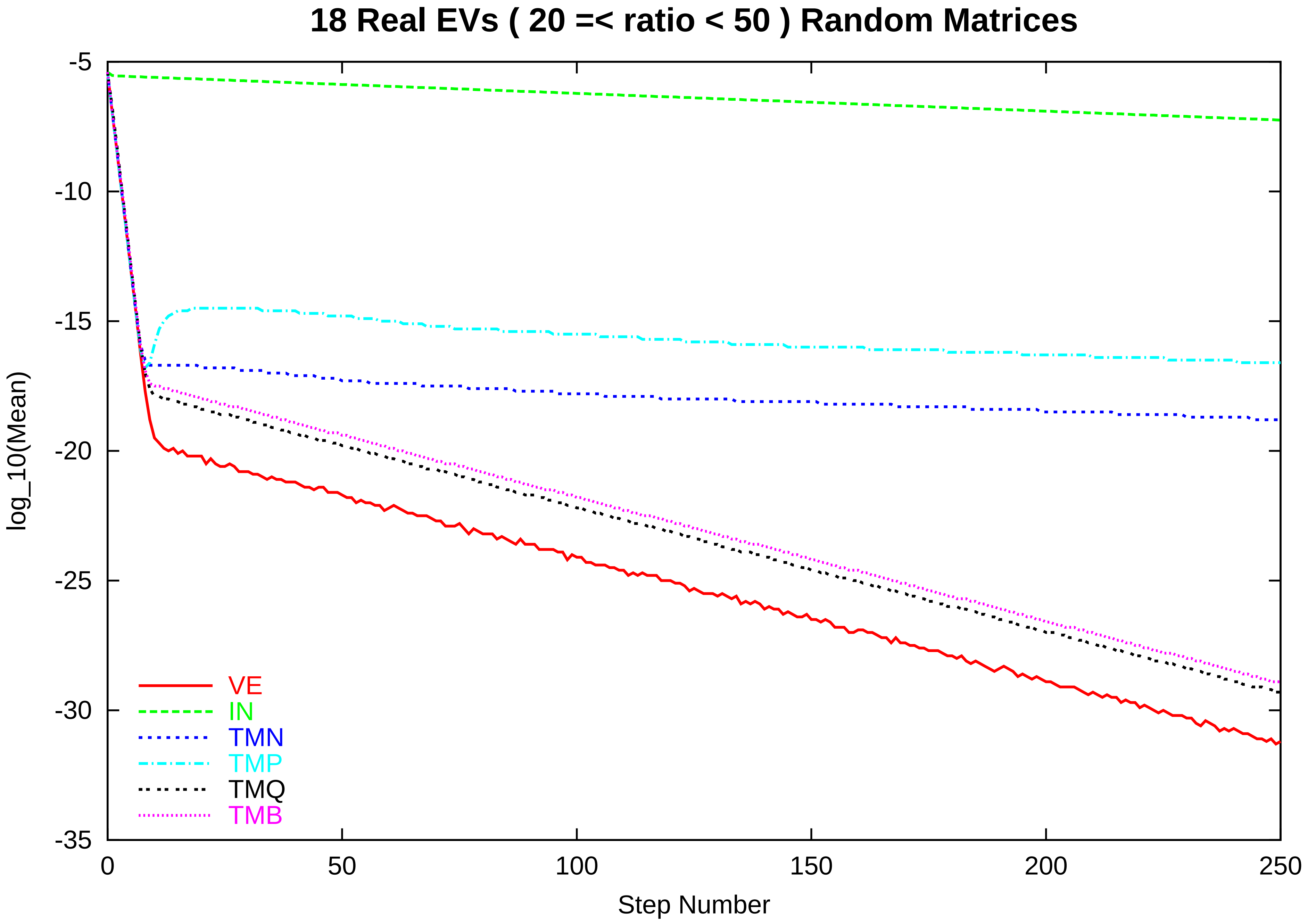
- Iterations through matrix
- Sets of iterations through matrix and its inverse


325 Conjugate EVs Random Matrices
0
-20
-40
-60
-80
-100
-120
log_10(Mean)
0
50
100
150
200
250
Step Number
VE
IN
TMN
TMP
TMQ
TMB


520 Real EVs (ratio < 5) Random Matrices
log_10(Mean)
Step Number
0
-10
-20
-30
-40
-50
-60
-70
-80
0
50
100
150
200
250
VE
IN
TMN
TMP
TMQ
TMB


80 Real EVs (5 =< ratio < 10) Random Matrices
log_10(Mean)
0
-10
-20
-30
-40
-50
-60
0
50
100
150
200
250
Step Number
VE
IN
TMN
TMP
TMQ
TMB


40 Real EVs (10 =< ratio < 20) Random Matrices
log_10(Mean)
Step Number
-5
-10
-15
-20
-25
-30
-35
-40
-45
-50
-55
0
50
100
150
200
250
VE
IN
TMN
TMP
TMQ
TMB


18 Real EVs (20 =< ratio < 50) Random Matrices
log_10(Mean)
-5
-10
-15
-20
-25
-30
-35
0
50
100
150
200
250
Step Number
VE
IN
TMN
TMP
TMQ
TMB

Random Matrices - Discrete

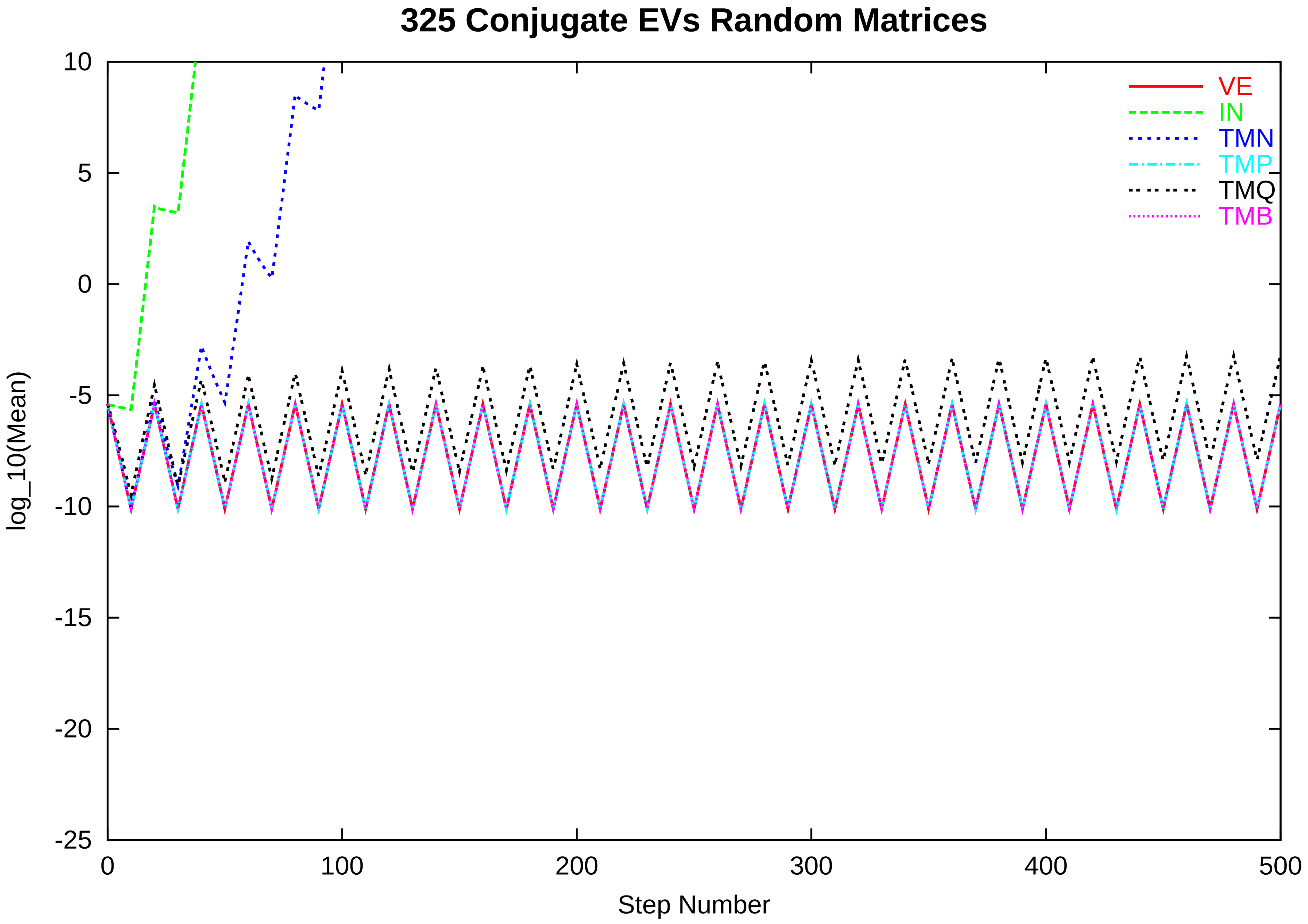
Select 1000 twodimensional random matrices with coefficients in $[-1, 1]$.Sort according to eigenvalues into seven sub-cases.

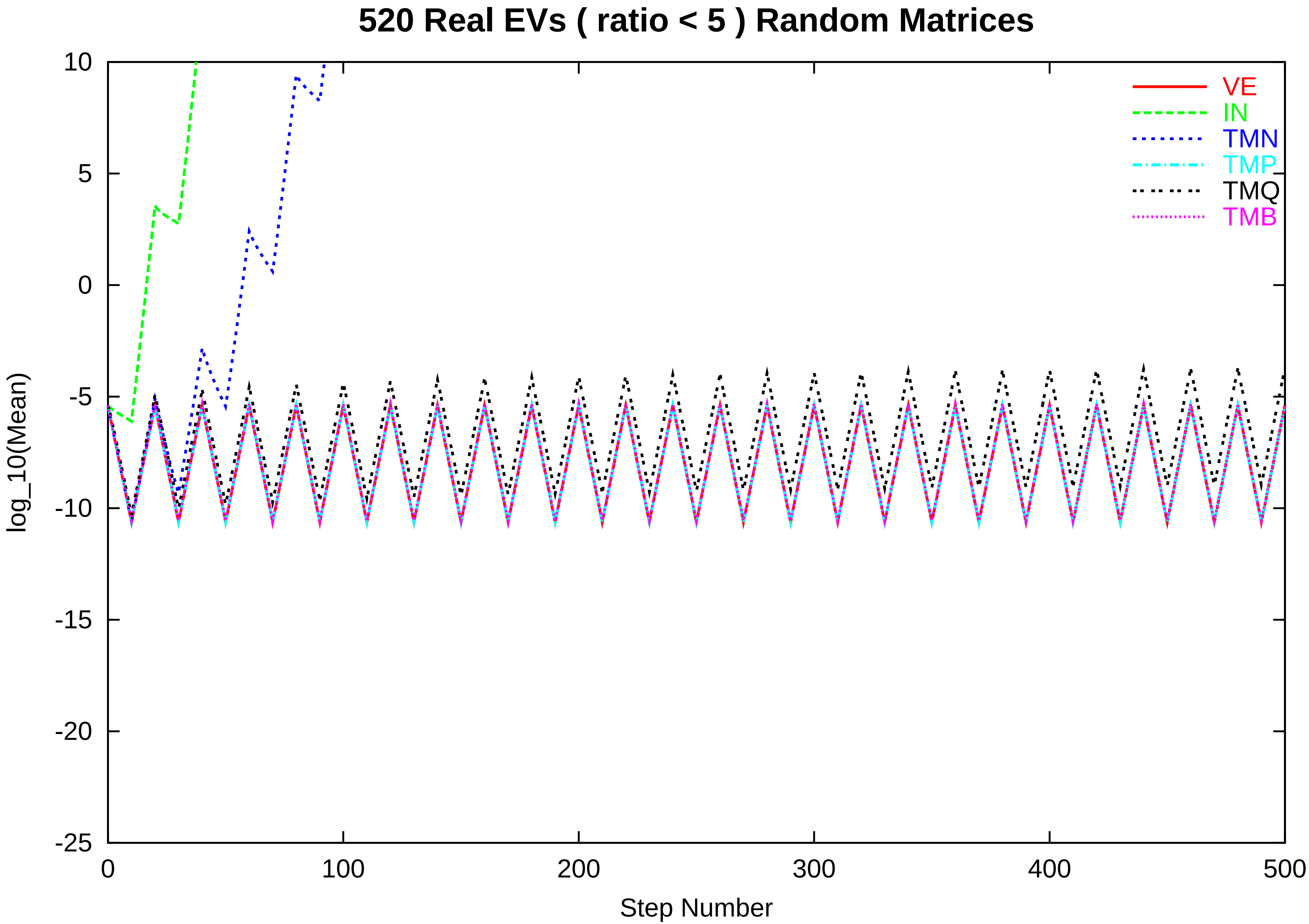
Perform iteration in the following ways:

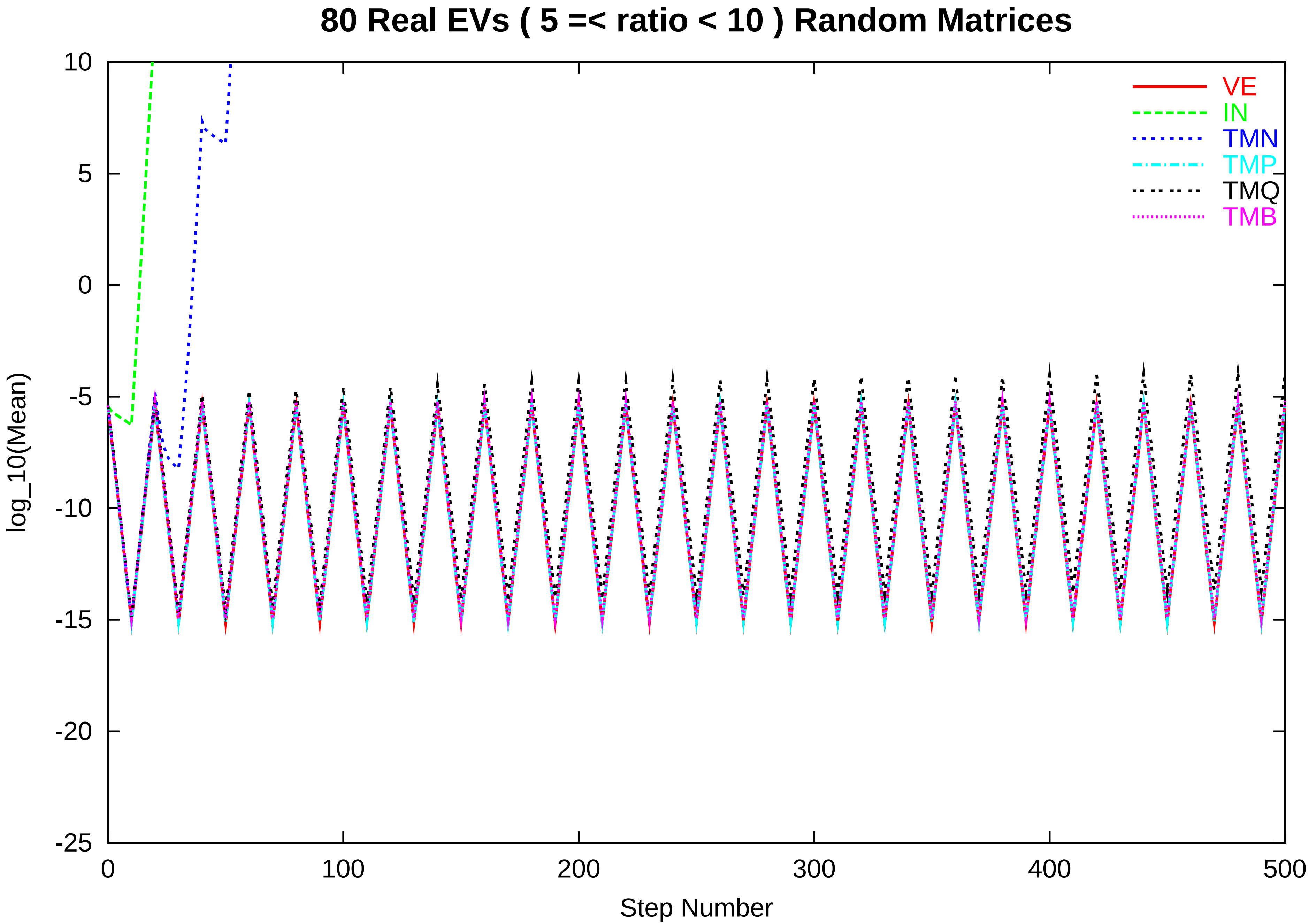
- Naive Interval
- Naive Taylormodel
- Parallelepiped-preconditioned Taylormodel
- QR-preconditioned Taylormodel
- Blunted preconditioned TM, various blunting factors
- Set of four floating point corner points for volume estimation

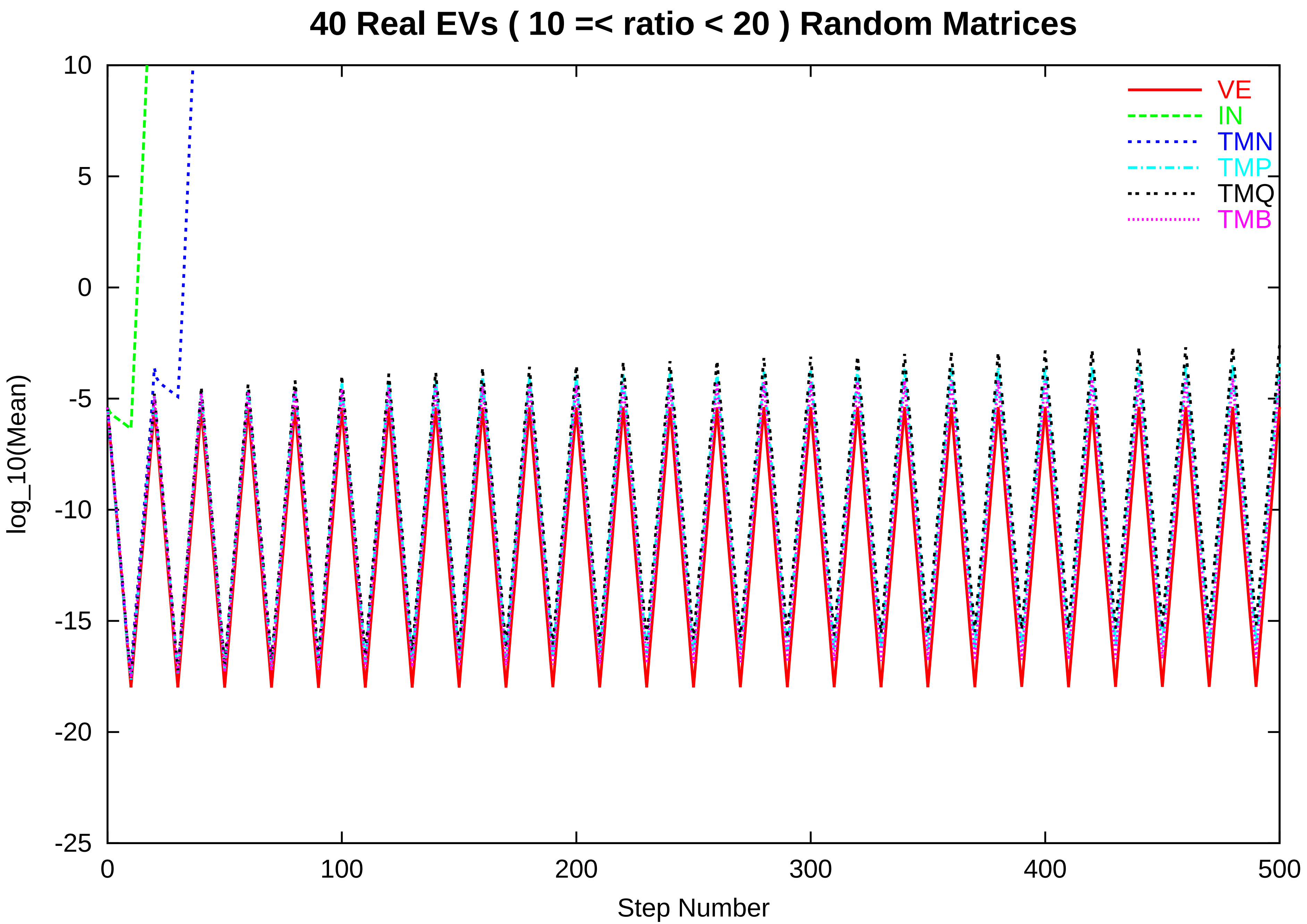
Perform the following tasks:

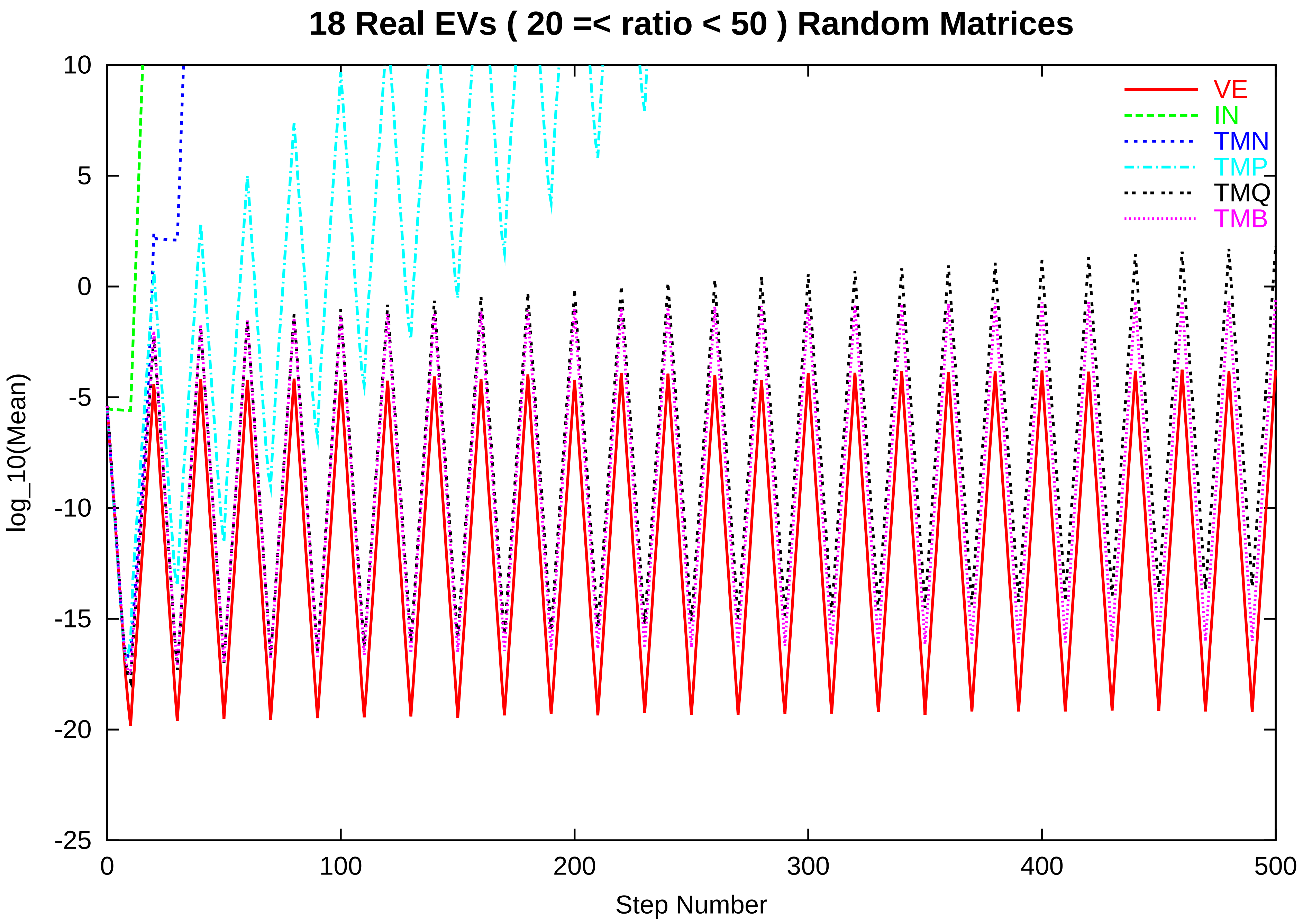
- Iterations through matrix
- Sets of iterations through matrix and its inverse

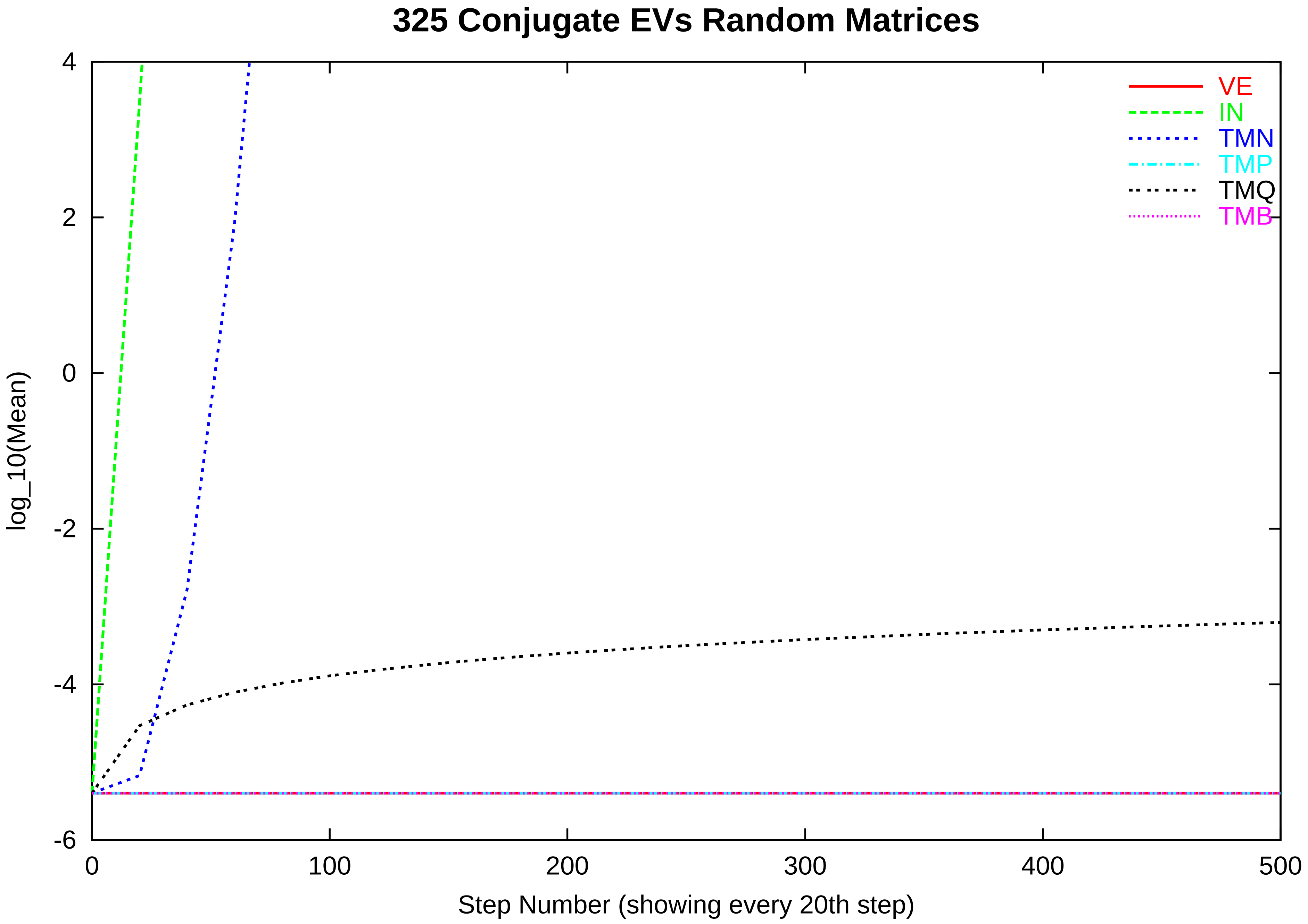

325 Conjugate EVs Random Matrices
log_10(Mean)
10
5
0
-5
-10
-15
-20
-25
0
100
200
300
400
500
Step Number
VE
IN
TMN
TMP
TMQ
TMB

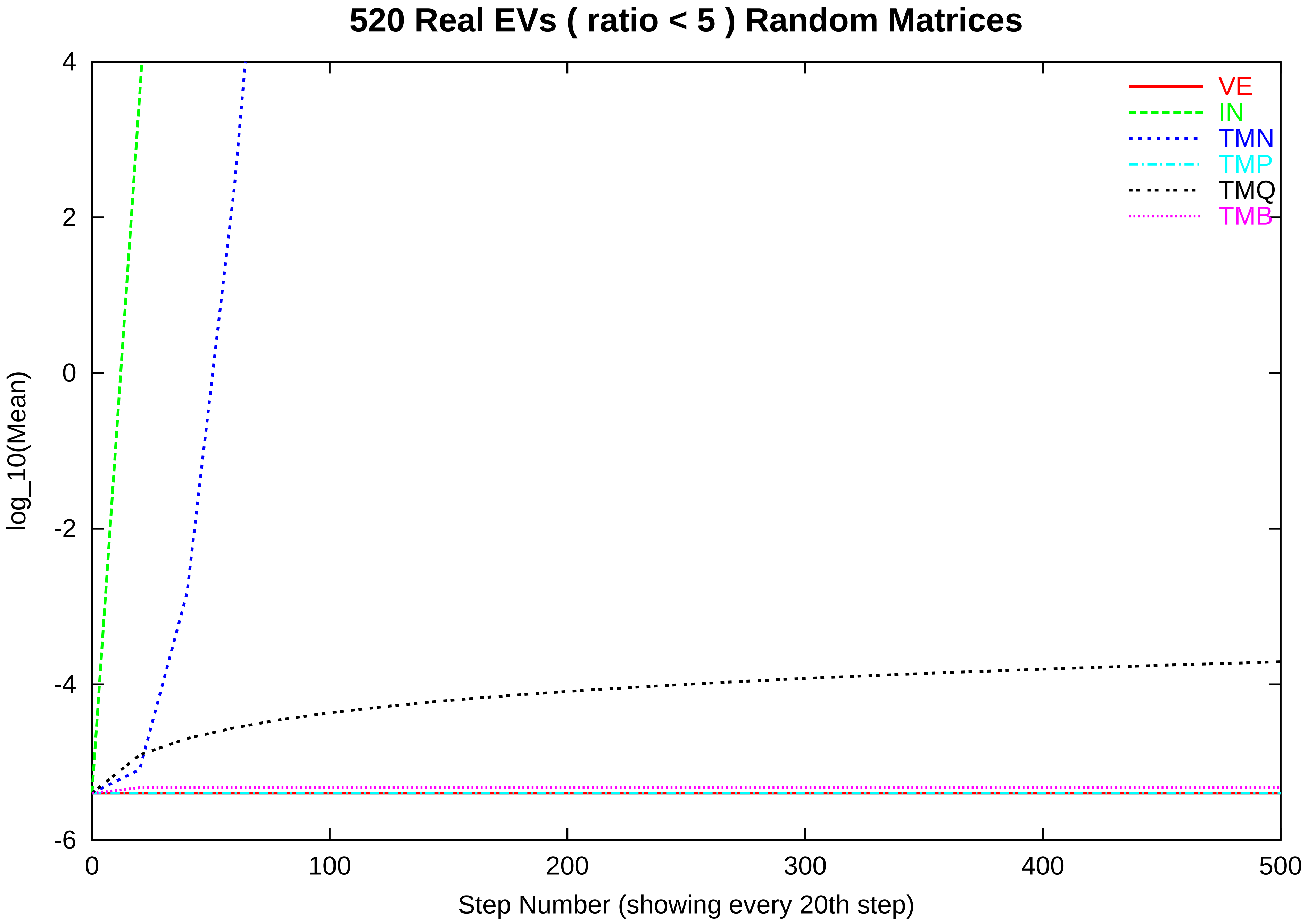

520 Real EVs (ratio < 5) Random Matrices
10
5
0
-5
-10
-15
-20
-25
log_10(Mean)
0
100
200
300
400
500
Step Number
VE
IN
TMN
TMP
TMQ
TMB

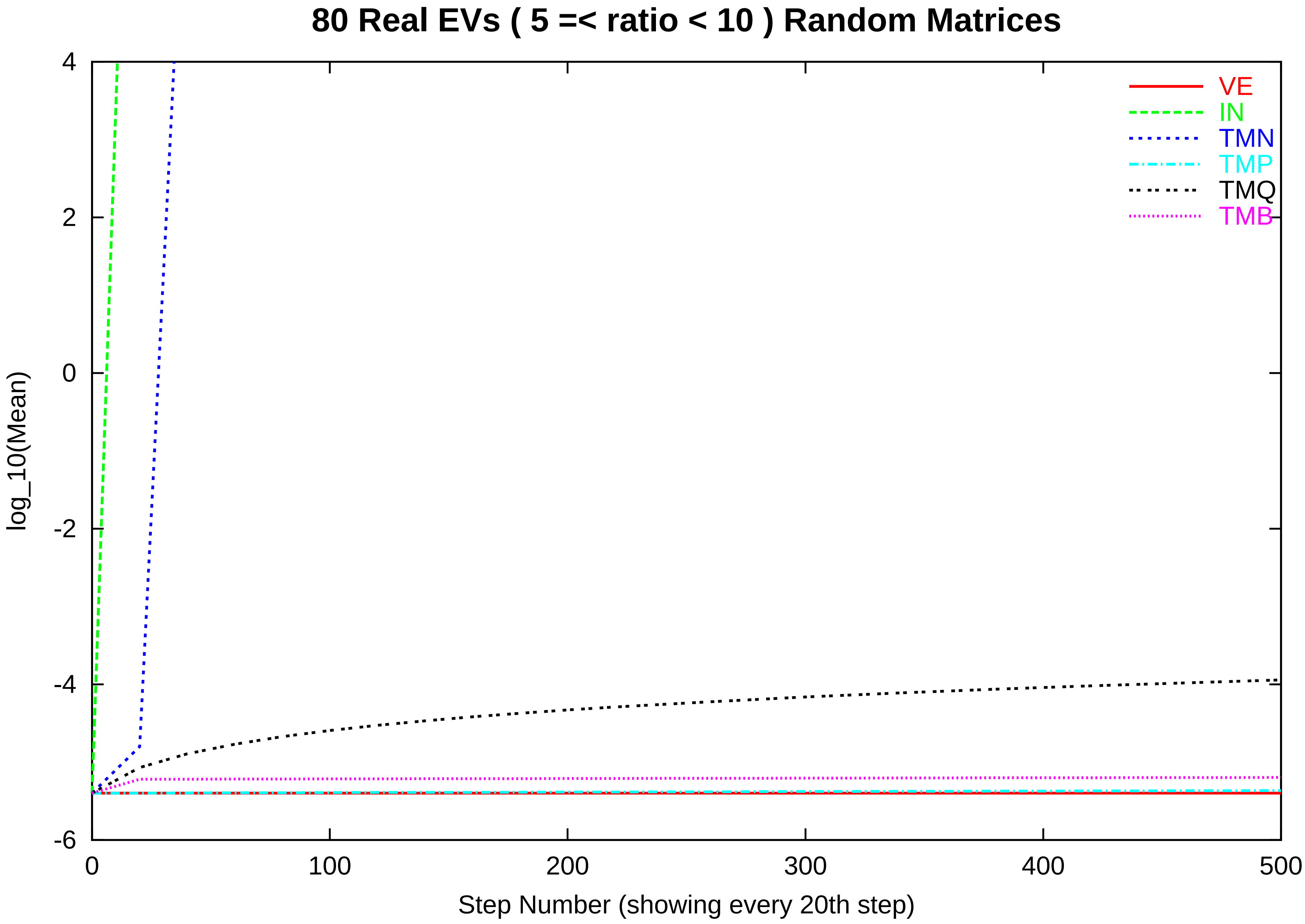

80 Real EVs (5 =< ratio < 10) Random Matrices
log_10(Mean)
Step Number
10
5
0
-5
-10
-15
-20
-25
0
100
200
300
400
500
VE
IN
TMN
TMP
TMQ
TMB

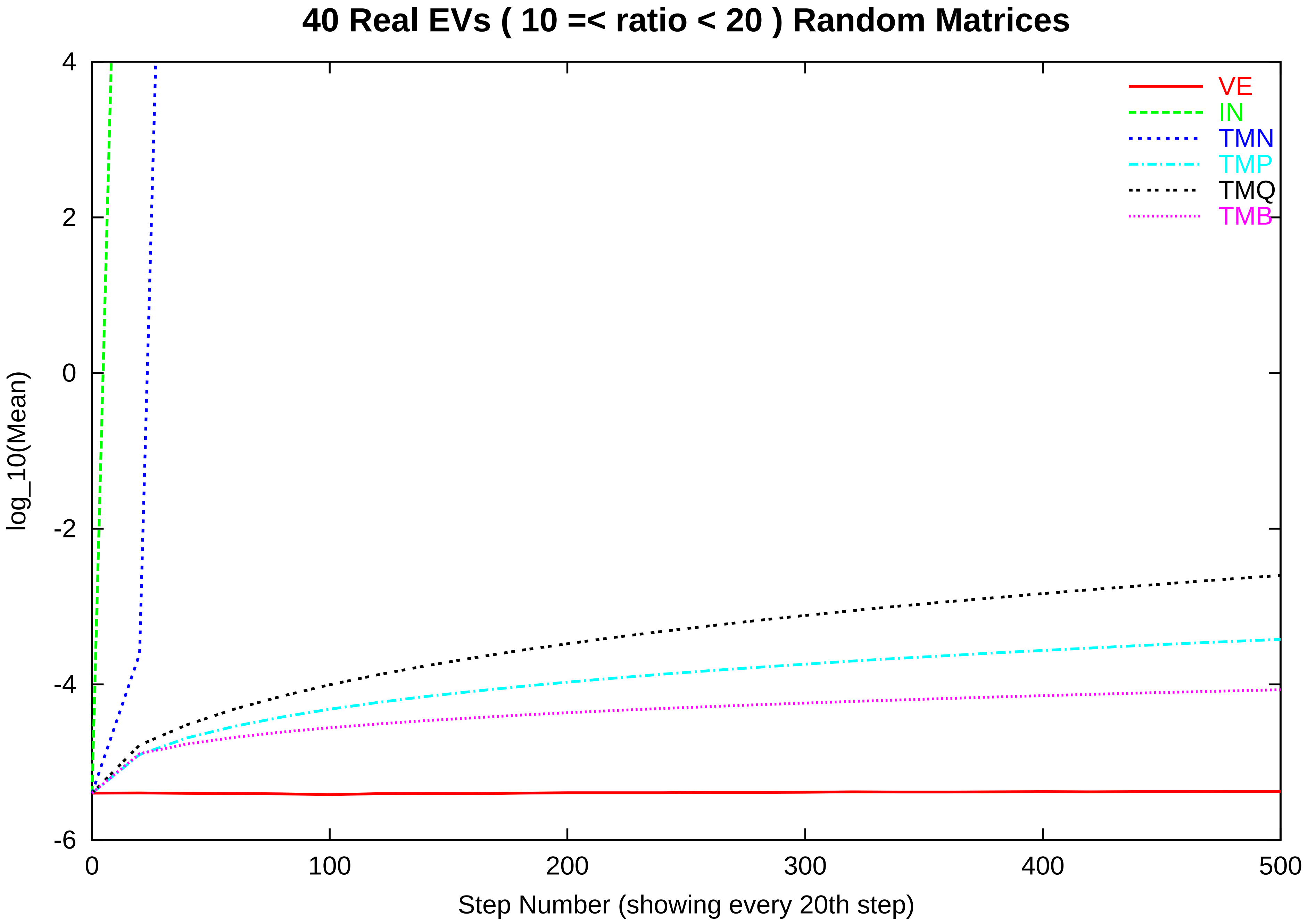

40 Real EVs (10 =< ratio < 20) Random Matrices
VE
IN
TMN
TMP
TMQ
TMB
10
5
0
-5
-10
-15
-20
-25
log_10(Mean)
0
100
200
300
400
500
Step Number


18 Real EVs (20 =< ratio < 50) Random Matrices
log_10(Mean)
Step Number
10
5
0
-5
-10
-15
-20
-25
0
100
200
300
400
500
VE
IN
TMN
TMP
TMQ
TMB

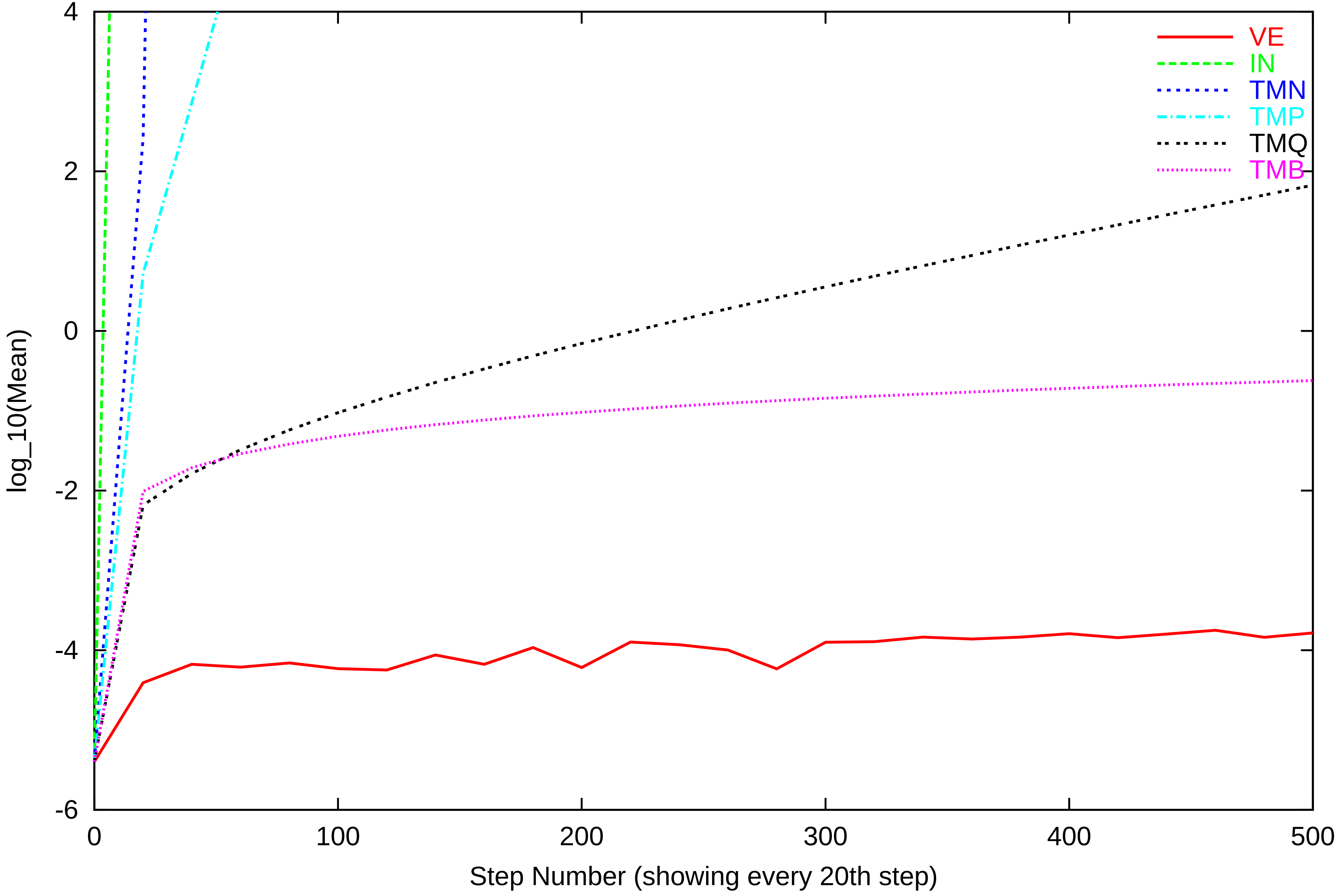

325 Conjugate EVs Random Matrices
log_10(Mean)
4
2
0
-2
-4
-6
0
100
200
300
400
500
Step Number (showing every 20th step)
VE
IN
TMN
TMP
TMQ
TMB


520 Real EVs (ratio < 5) Random Matrices
log_10(Mean)
4
2
0
-2
-4
-6
0
100
200
300
400
500
Step Number (showing every 20th step)
VE
IN
TMN
TMP
TMQ
TMB


80 Real EVs (5 =< ratio < 10) Random Matrices
log_10(Mean)
4
2
0
-2
-4
-6
0
100
200
300
400
500
Step Number (showing every 20th step)
VE
IN
TMN
TMP
TMQ
TMB


40 Real EVs (10 =< ratio < 20) Random Matrices
log_10(Mean)
4
2
0
-2
-4
-6
0
100
200
300
400
500
Step Number (showing every 20th step)
VE
IN
TMN
TMP
TMQ
TMB

18 Real EVs (20 =< ratio < 50) Random Matrices



Random Matrices - Continuous

Select 10 fourdimensional random matrices A with coefficients in $[-1, 1]$. Solve ODE

$$\frac{d}{dt} r = A \cdot r$$

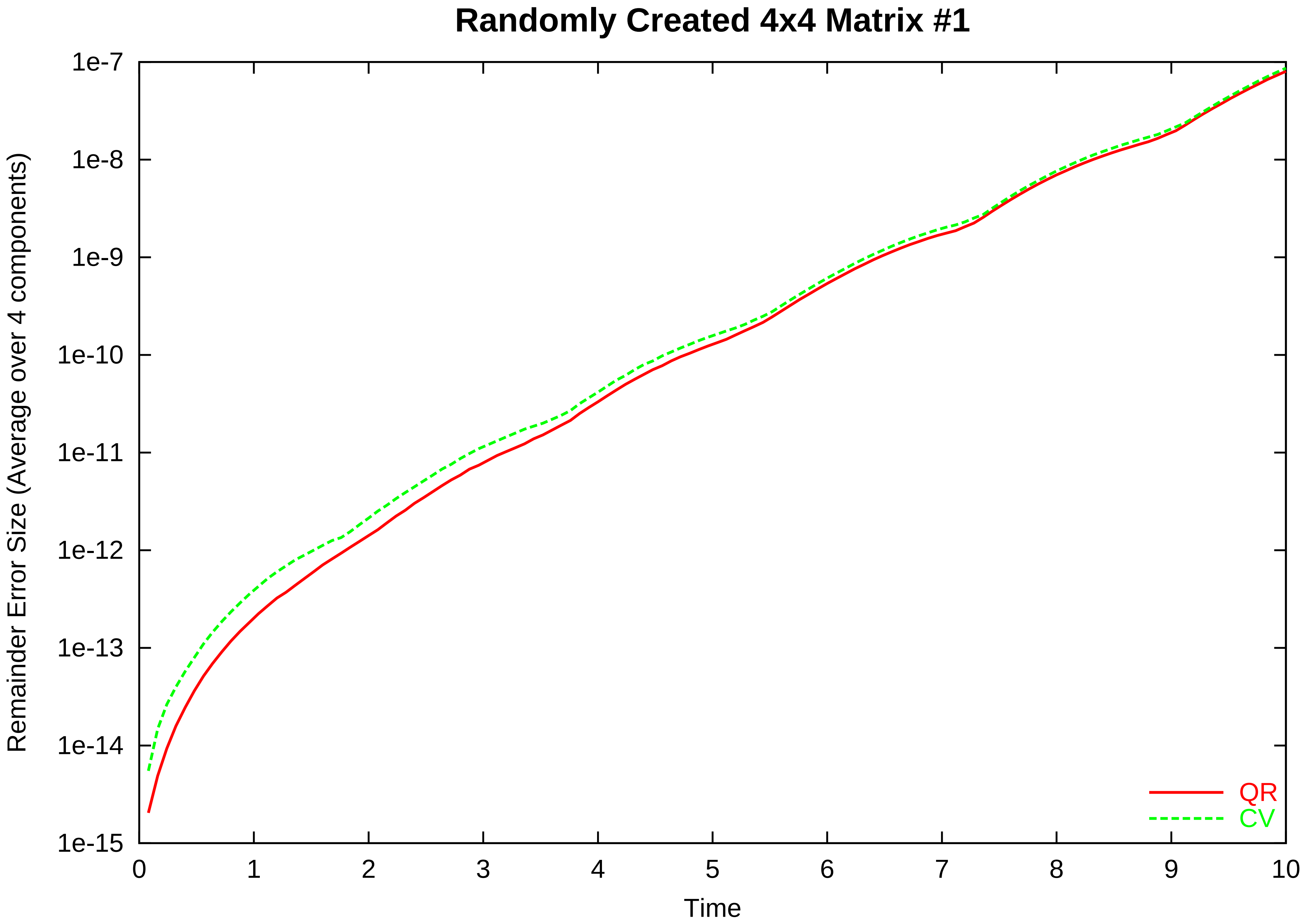
with random initial conditions.

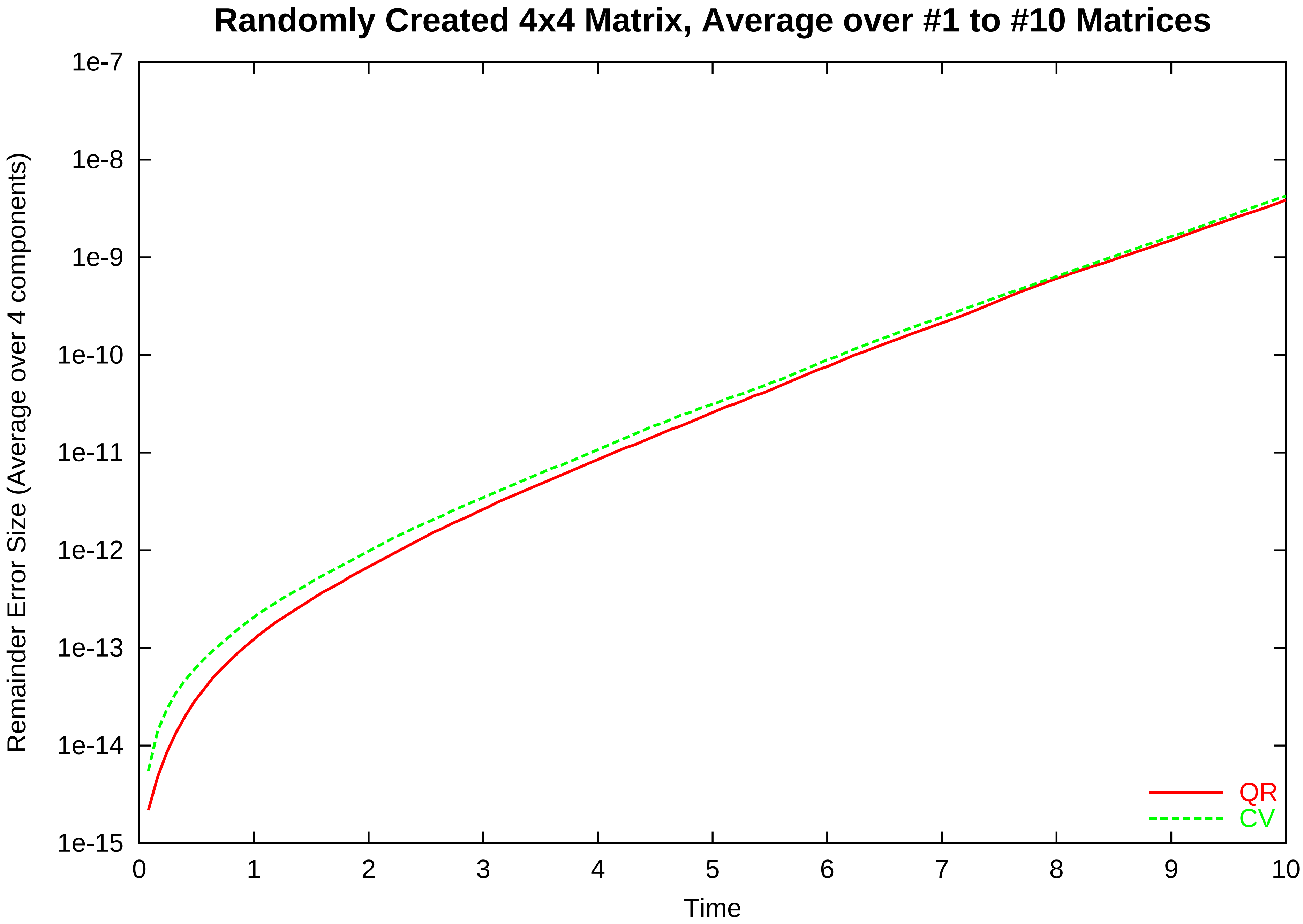
Perform integration in the following ways:

- Curvilinear-Preconditioned Taylormodel
- QR-Preconditioned Taylormodel

Observe that

- CV and QR preconditiong have the same asymptotic behavior
- Both lead to error growth agreeing with growth along longest EV up to 1%.
- Thus, same error growth as in non-validated case.


Randomly Created 4x4 Matrix #1
Remainder Error Size (Average over 4 components)
1e-7
1e-8
1e-9
1e-10
1e-11
1e-12
1e-13
1e-14
1e-15
0
1
2
3
4
5
6
7
8
9
10
Time
QR
CV


Randomly Created 4x4 Matrix, Average over #1 to #10 Matrices
Remainder Error Size (Average over 4 components)
1e-7
1e-8
1e-9
1e-10
1e-11
1e-12
1e-13
1e-14
1e-15
0
1
2
3
4
5
6
7
8
9
10
Time
QR
CV

A Muon Cooling Ring

Example from Beam Physics: Simple model of muon cooling ring, using curvilinear preconditioning.

Simultaneous damping via matter, and azimuthal accelerations. Equations of motion:

$$\begin{aligned}
\dot{x} &= p_x \\
\dot{y} &= p_y \\
\dot{p}_x &= p_y - \frac{\alpha}{\sqrt{p_x^2 + p_y^2}} \cdot p_x + \frac{\alpha}{\sqrt{x^2 + y^2}} \cdot y \\
\dot{p}_y &= -p_x - \frac{\alpha}{\sqrt{p_x^2 + p_y^2}} \cdot p_y - \frac{\alpha}{\sqrt{x^2 + y^2}} \cdot x
\end{aligned}$$

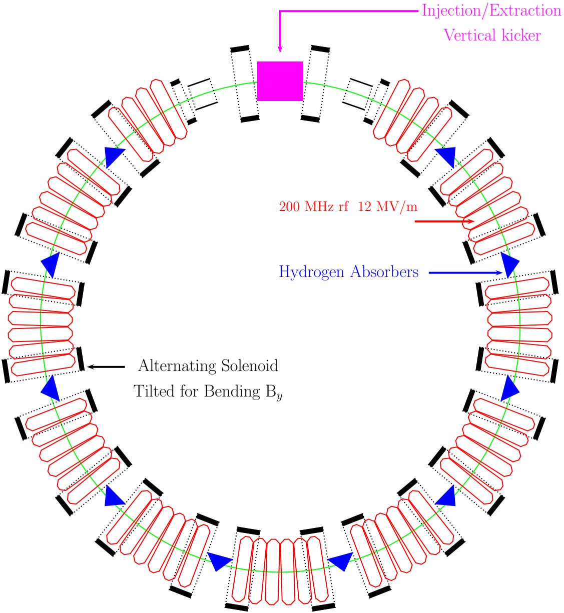
Has invariant solution

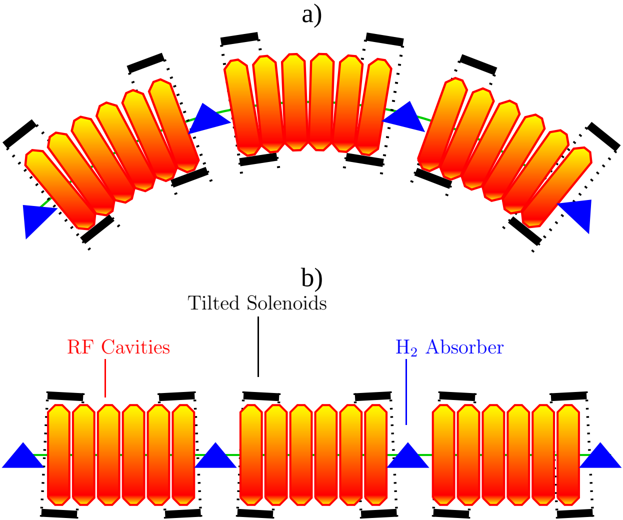
$$(x, y, p_x, p_y)_I(t) = (\cos t, -\sin t, -\sin t, -\cos t),$$

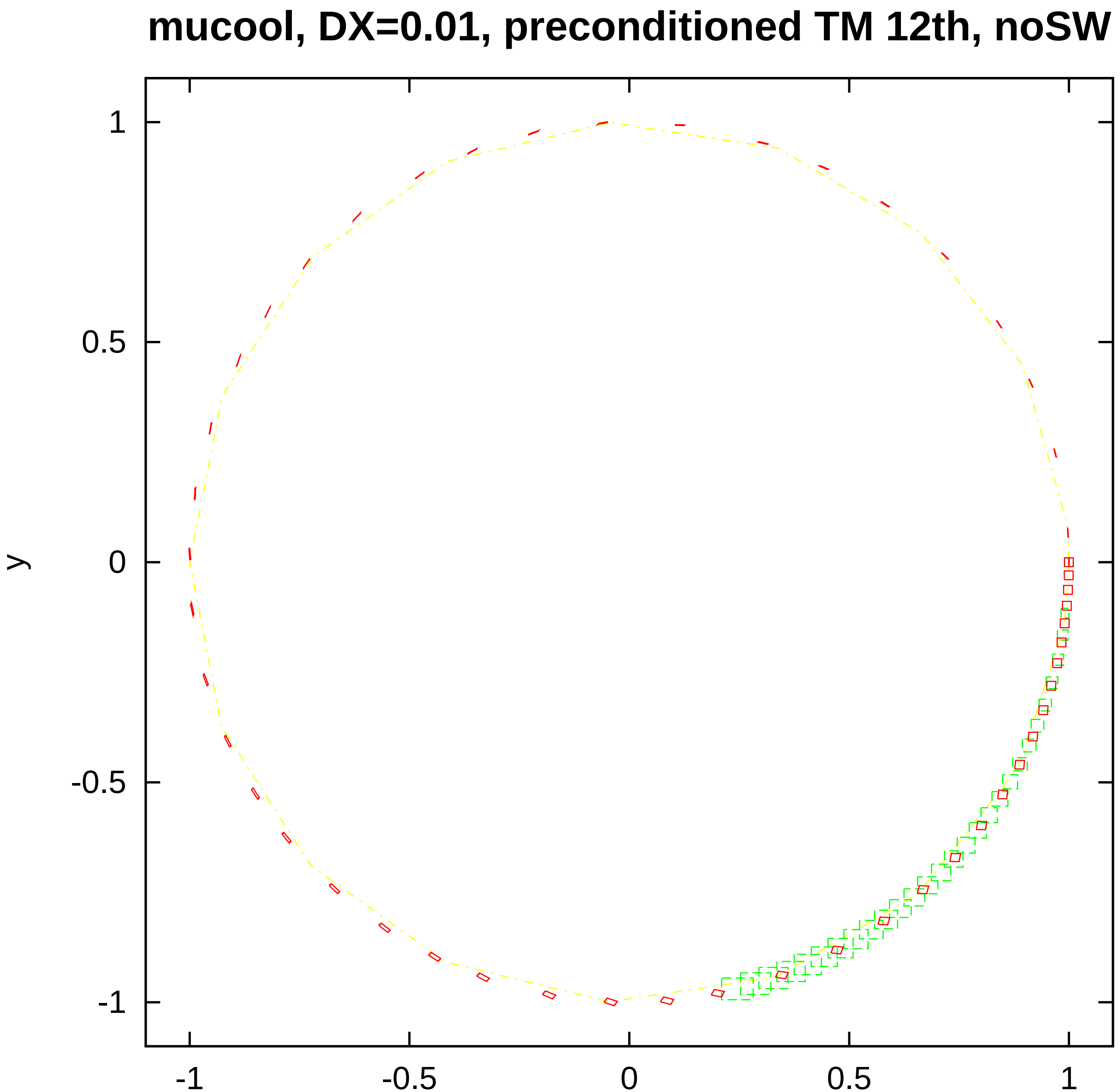
ODE asymptotically approach circular motion of the form

$$(x, y, p_x, p_y)_a(t) = (\cos(t - \phi), -\sin(t - \phi), -\sin(t - \phi), -\cos(t - \phi)),$$

where ϕ is a characteristic angle for each particle.


Injection/Extraction
Vertical kicker
200 MHz rf 12 MV/m
Hydrogen Absorbers
Alternating Solenoid
Tilted for Bending B$_y$


a)
b)
Tilted Solenoids
RF Cavities
H_2 Absorber


mucool, DX=0.01, preconditioned TM 12th, noSW
y
x
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

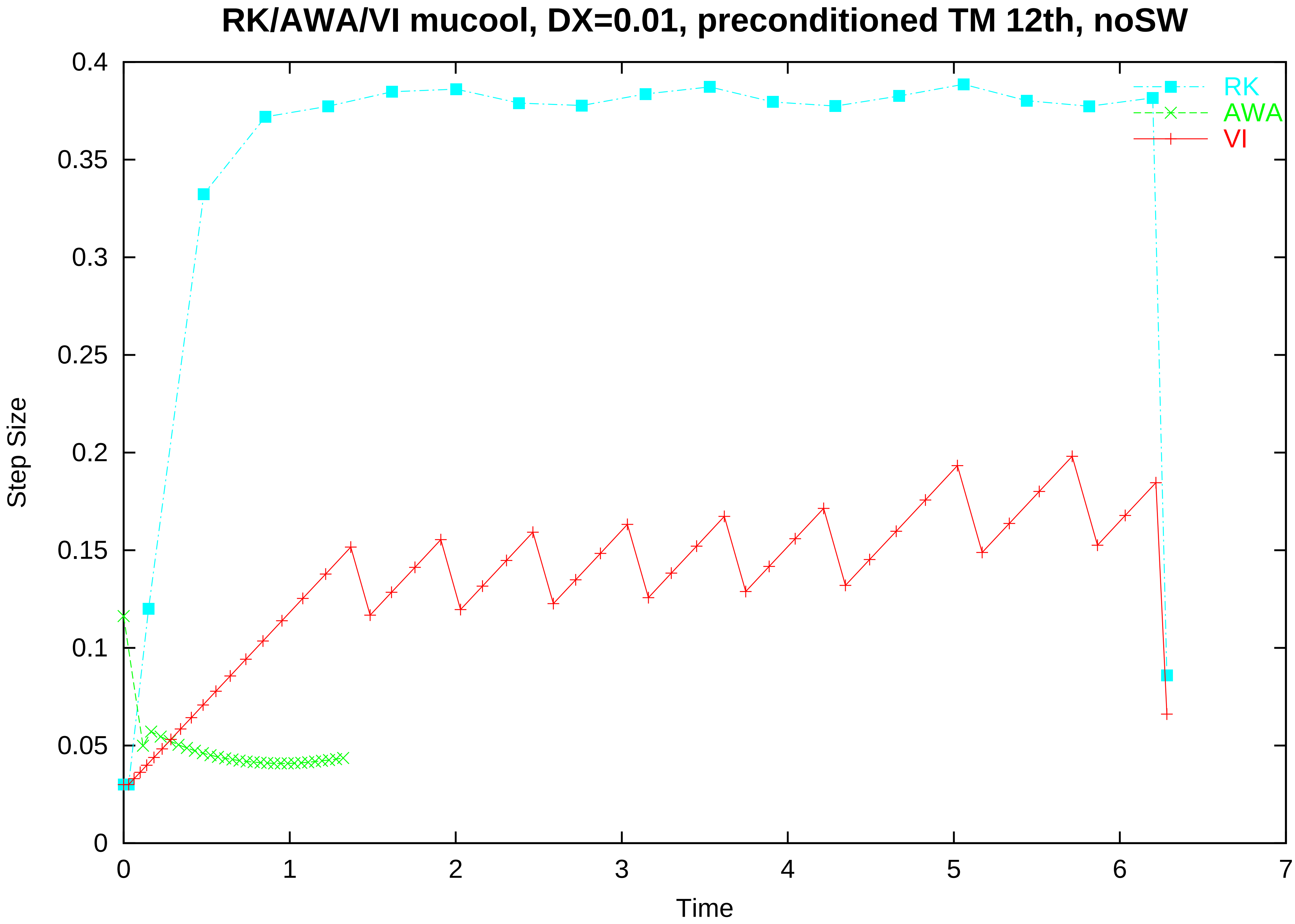
A Muon Cooling Ring - Results

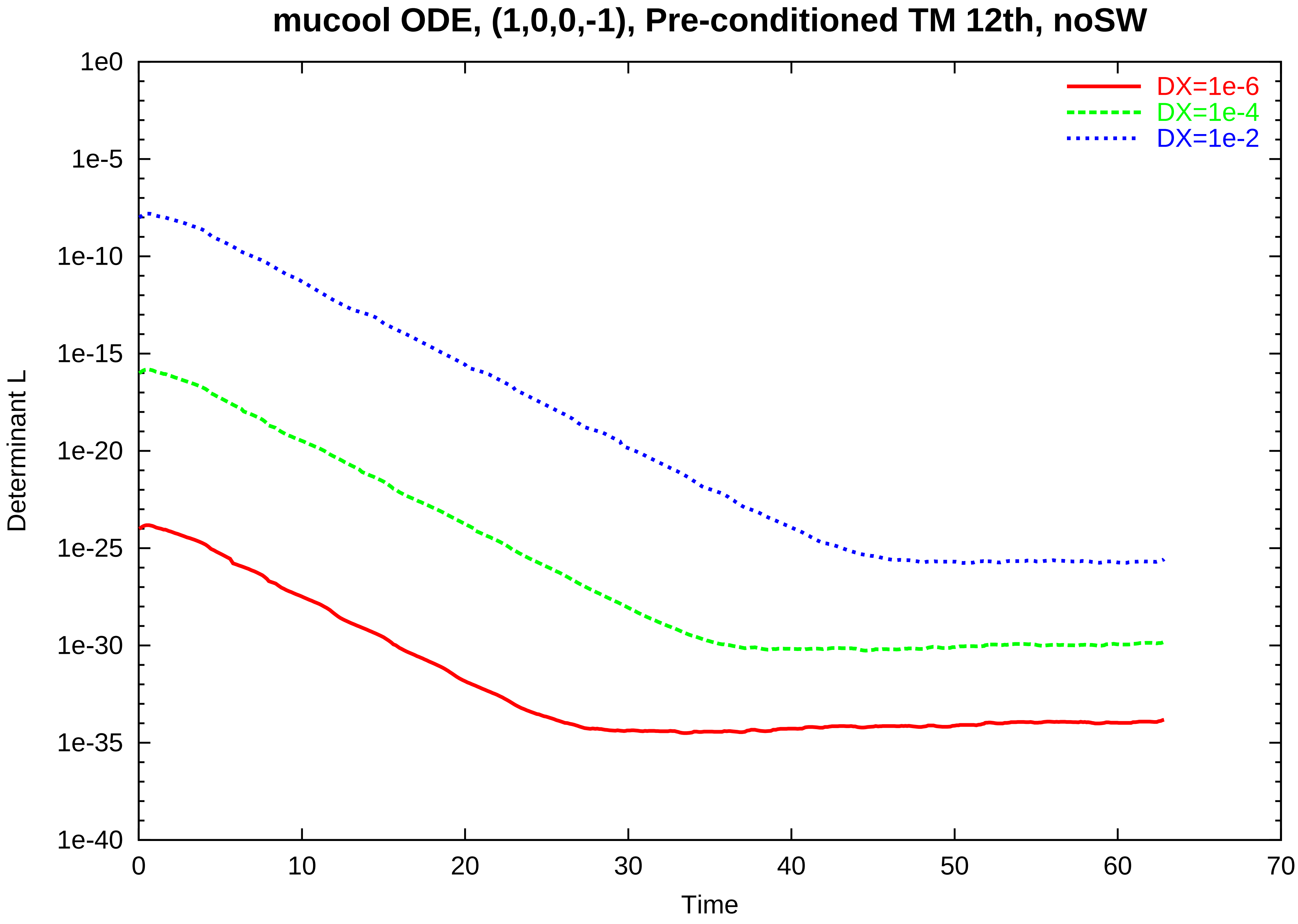
1. Need to treat a large box of $[-10^{-2}, 10^{-2}]^4$
2. Because of damping action towards the invariant limit cycle, the linear part of the motion is more and more ill-conditioned.

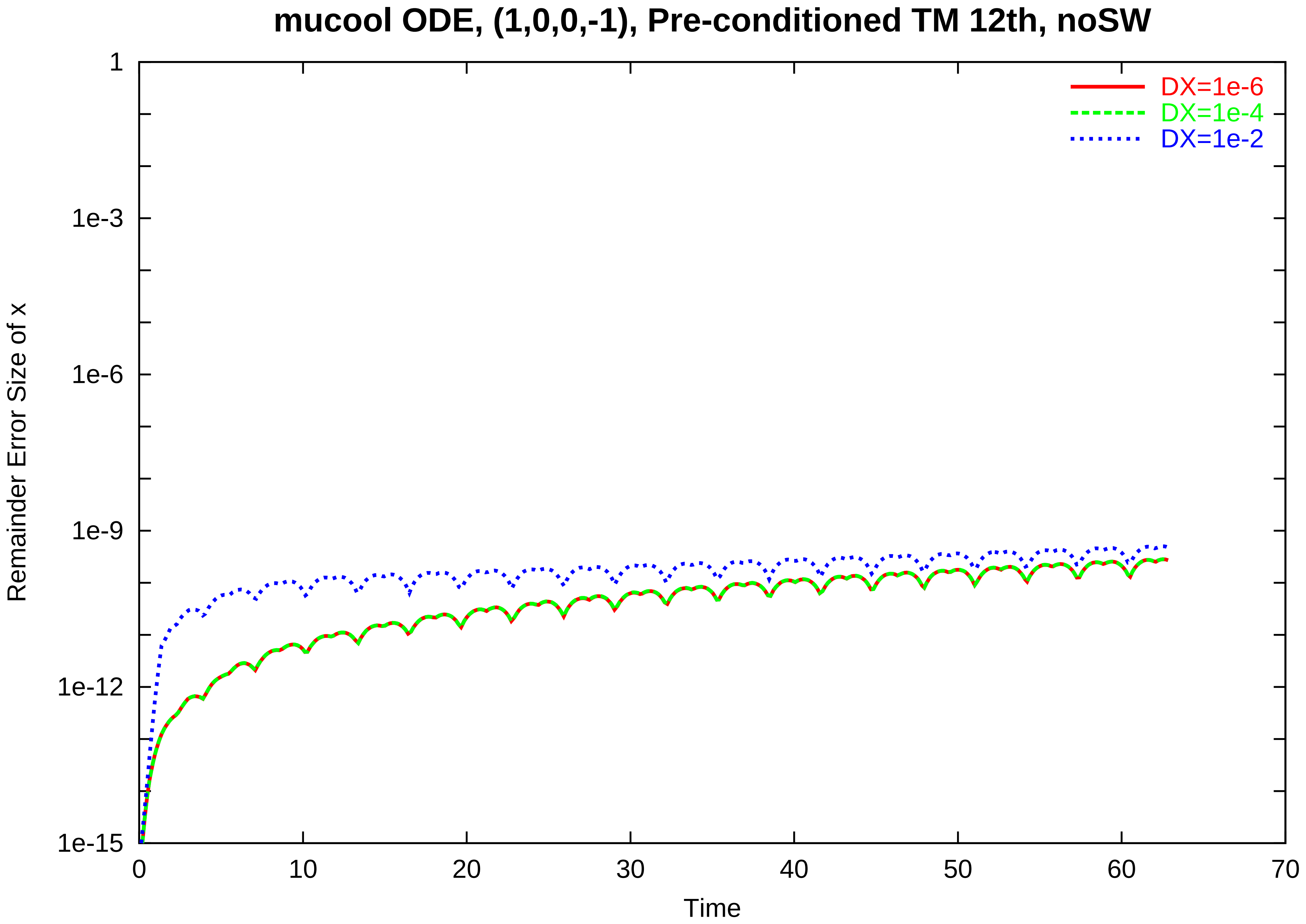
COSY easily integrates 10 cycles for $d = 10^{-2}$ with curvilinear preconditioning and QR preconditioning. AWA (method 4) behaves as follows:

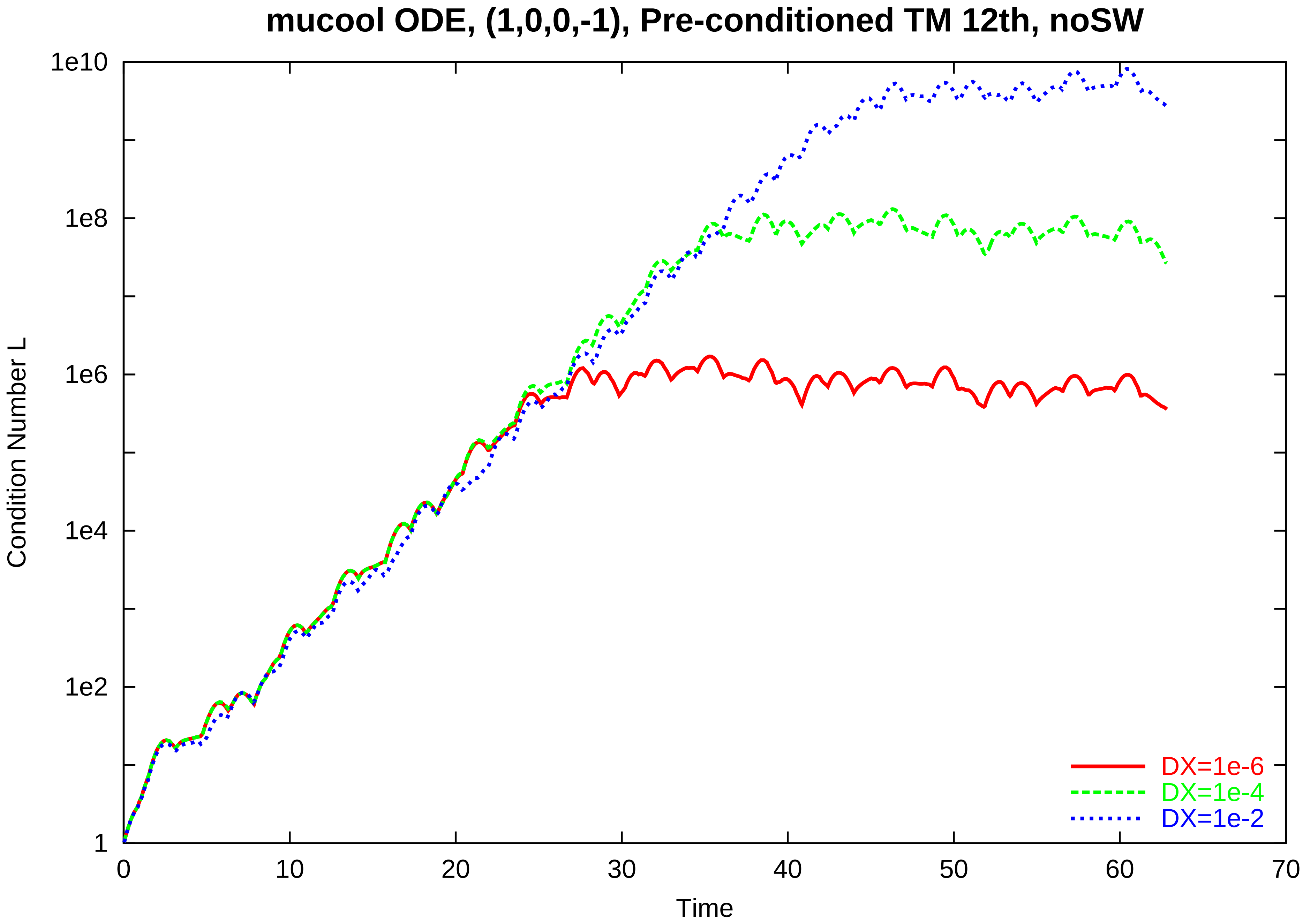
d	**Cycles**
10^{-2}	0.22
10^{-3}	1.25
10^{-4}	9.5

Thus, trying to simulate the system with AWA requires $> (10^2)^4 = 10^8$ subdivisions of the box that COSY can transport in one piece.


RK/AWA/VI mucool, DX=0.01, preconditioned TM 12th, noSW
Step Size
Time
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0
1
2
3
4
5
6
7
RK
AWA
VI


mucool ODE, (1,0,0,-1), Pre-conditioned TM 12th, noSW
DX=1e-6
DX=1e-4
DX=1e-2
Determinant L
1e0
1e-5
1e-10
1e-15
1e-20
1e-25
1e-30
1e-35
1e-40
0
10
20
30
40
50
60
70
Time


mucool ODE, (1,0,0,-1), Pre-conditioned TM 12th, noSW
Remainder Error Size of x
Time
1
1e-3
1e-6
1e-9
1e-12
1e-15
0
10
20
30
40
50
60
70
DX=1e-6
DX=1e-4
DX=1e-2


mucool ODE, (1,0,0,-1), Pre-conditioned TM 12th, noSW
Condition Number L
Time
1e10
1e8
1e6
1e4
1e2
1
0
10
20
30
40
50
60
70
DX=1e-6
DX=1e-4
DX=1e-2

--- Conclusion ---

Key Features and Algorithms of COSY-VI

- 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.
- Resulting data is available in various levels including graphics output.

TM Integrator: Pushing Further...

- The Reference Trajectory and the Flow Operator
- Step Size Control
- Error Parametrization of Taylor Models
- Dynamic Domain Decomposition

The Reference Trajectory

First Step: Obtain Taylor expansion in time of solution of ODE of center point c, i.e. obtain

$$c(t) = c_0 + c_1 \cdot (t - t_0) + c_2 \cdot (t - t_0)^2 + ... + c_n \cdot (t - t_0)^n$$

Very well known from day one how to do this with automatic differentiation. Rather convenient way: can be done by n iterations of the Picard Operator

$$c(t) = c_0 + \int_0^t f(r(t'), t) dt'$$

in one-dimensional Taylor arithmetic. Each iteration raises the order by one; so in each iteration i, only need to do Taylor arithmetic in order i. In either way, this step is **cheap** since it involves only **one-dimensional** operations.

The Nonlinear Flow

Second Step: The goal is to obtain Taylor expansion in time to order n **and** initial conditions to order k. Note:

1. This is usually the most **expensive** step. In the original Taylor model-based algorithm, it is done by n **iterations** of the Picard Operator in multi-dimensional Taylor arithmetic, where c_0 is now a polynomial in initial conditions.
2. The case $k = 1$ has been known for a long time. Traditionally solved by setting up **ODEs for sensitivities** and solving these as before.
3. The case of higher k goes back to Beam Physics (M. Berz, Particle Accelerators 1988)
4. Newest Taylor model arithmetic naturally supports different expansions orders k for initial conditions and n for time.

Goal: Obtain flow with one **single evaluation** of right hand side.

The Nonlinear Relative ODE

We now develop a better way for second step.
First: introduce new "perturbation" variables $\tilde{r}$ such that

$$r(t) = c(t) + A \cdot \tilde{r}(t).$$

The matrix A provides **preconditioning**. ODE for $\tilde{r}(t)$:

$$\tilde{r}' = A^{-1} \left[f(c(t) + A \cdot \tilde{r}(t)) - c'(t) \right]$$

Second: evaluate ODE for $\tilde{r}'$ in Taylor arithmetic. Obtain a Taylor expansion of the ODE, i.e.

$$\tilde{r}' = P(\tilde{r}, t)$$

up to order n in time and k in $\tilde{r}$. **Very important** for later use: the polynomial P will have no constant part, i.e.

$$P(0, t) = 0.$$

Reminder: The Lie Derivative

Let

$$r' = f(r,t)$$

be a dynamical system. Let g be a variable in state space, and let us study $g(r(t))$, i.e. along a solution of the ODE. We have

$$\frac{d}{dt}g(t) = f \cdot \nabla g + \frac{\partial g}{\partial t}$$

Introducing the **Lie Derivative** $L_f = f \cdot \nabla + \partial/\partial t$, we have

$$\frac{d^n}{dt^n}g = L_f^n g \text{ and } g(t) \approx \sum_{i=0}^{n} \frac{(t-t_0)^i}{i!}\, L_f^i g\Big/_{t=t_0}$$

Polynomial Flow from Lie Derivative

Remember the ODE for $\tilde{r}'$:

$$\tilde{r}' = P(\tilde{r}, t)$$

up to order n in time and k in $\tilde{r}$. And remember $P(0,t) = 0$. Thus we can obtain the n-th order expansion of the flow as

$$\tilde{r}(t) = \sum_{i=0}^{n} \frac{(t-t_0)^i}{i!} \cdot \left(P \cdot \nabla + \frac{\partial}{\partial t} \right)^i \tilde{r}_0 \Bigg/_{t=t_0}$$

- The fact that $P(0,t) = 0$ restores the derivatives lost in ∇
- The fact that $\partial/\partial t$ appears without origin-preserving factor limits the expansion to order n.

Performance of Lie Derivative Flow Methods

Apparently we have the following:

- Each term in the Lie derivative sum requires $v + 1$ derivations (very cheap, just re-shuffling of coefficients)
- Each term requires v multiplications
- We need **one** evaluation of f in ${}_nD_v$ (to set up ODE)

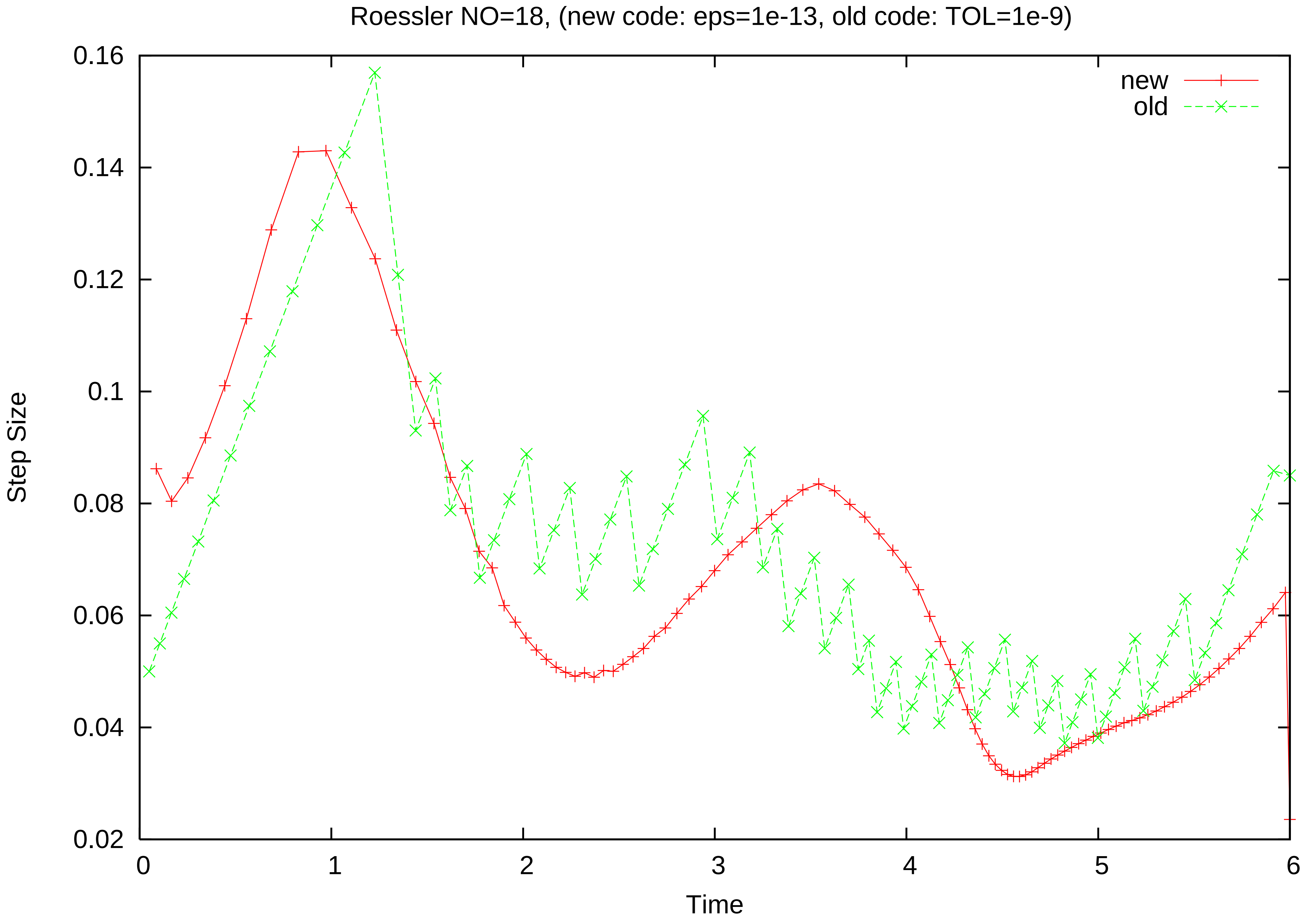
Compare this with the conventional algorithm, which requires n evaluations of the function f of the right hand side. Thus, roughly, if the evaluation of f requires more than v multiplications, the new method is more efficient.

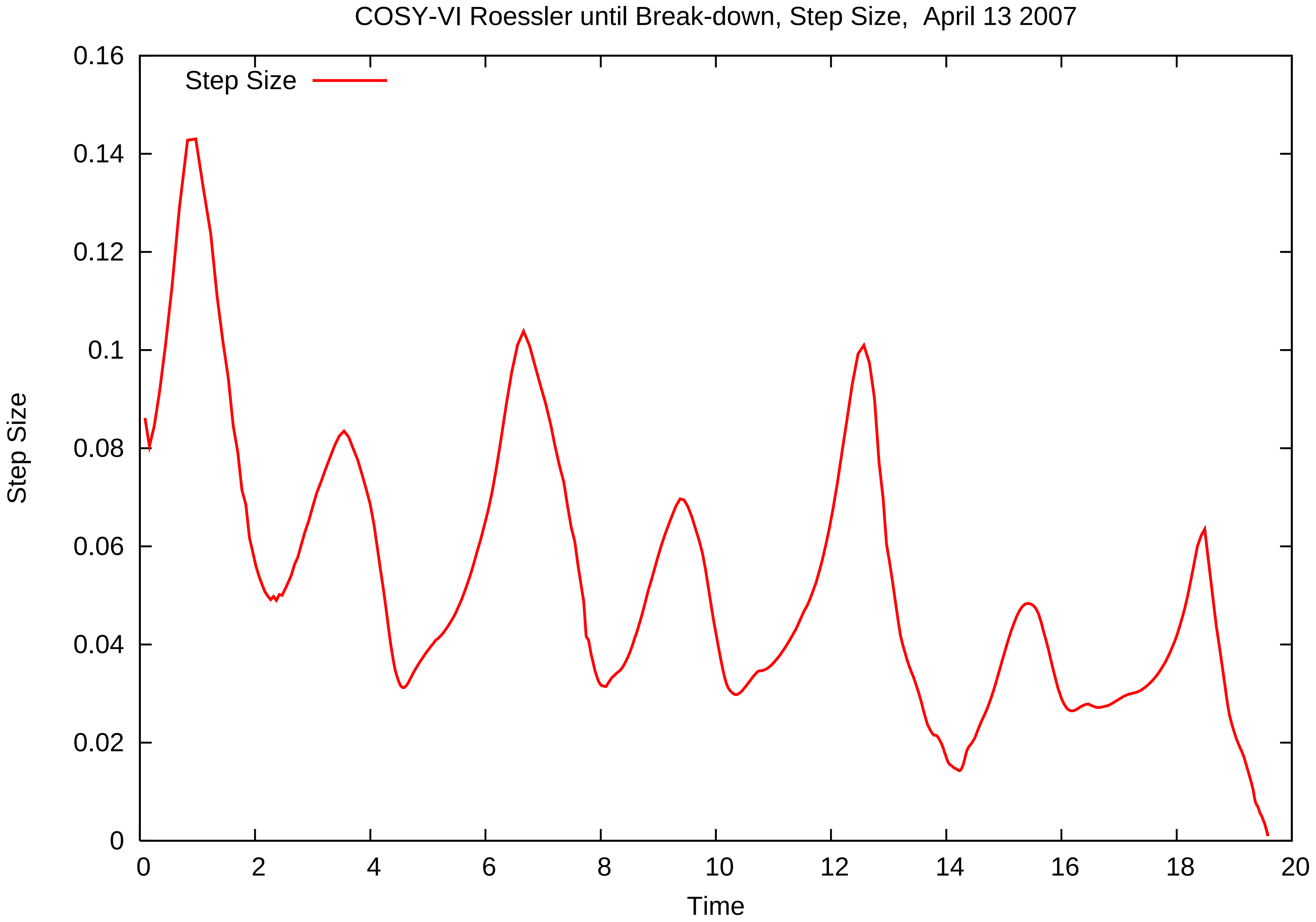
- Many practically appearing right hand sides f satisfy this.
- But on the other hand, if the function f does not satisfy this (for example for the linear case), then also P will be simple (in the linear case: P will be linear), and thus less operations appear

Step Size Control

Step size control to maintain approximate error ε in each step. Based on a suite of tests:

1. Utilize the **Reference Orbit.** Extrapolate the size of coefficients for estimate of remainder error, scale so that it reaches and get Δt_1. Goes back to Moore in 1960s. This is one of conveniences when using Taylor integrators.
2. Utilize the **Flow.** Compute flow time step with Δt_1.Extrapolate the contributions of each order of flow for estimate of remainder error to get update Δt_2.
3. Utilize a **Correction factor** c to account for overestimation in TM arithmetic as $c = \sqrt[n+1]{|R|/\varepsilon}$. Largely a measure of complexity of ODE. Dynamically update the correction factor.
4. Perform verification attempt for $\Delta t_3 = c \cdot \Delta t_2$


Roessler NO=18, (new code: eps=1e-13, old code: TOL=1e-9)
Step Size
Time
new
old
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1
2
3
4
5
6


COSY-VI Roessler until Break-down, Step Size, April 13 2007
Step Size
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
2
4
6
8
10
12
14
16
18
20
Time
Step Size

Error Parametrization of Taylor models

Motivation: Is it possible to absorb the remainder error bound intervals of Taylor models into the polynomial parts using additional parameters?

Phrase the question as the following problem:

1. Have Taylor models with 0 remainder error interval, which depend on the independent variables $\vec{x}$ and the parameters $\vec{\alpha}$.
$$\vec{T}_0 = \vec{P}_0(\vec{x}, \vec{\alpha}) + \overrightarrow{[0,0]}.$$
2. Perform Taylor model arithmetic on $\vec{T}_0$, namely $\vec{F}(\vec{T}_0)$
$$\vec{F}(\vec{T}_0) = \vec{P}(\vec{x}, \vec{\alpha}) + \vec{I}_F, \text{ where } \vec{I}_F \neq \overrightarrow{[0,0]}.$$
3. Try to absorb $\vec{I}_F$ into the polynomial part that depends on $\vec{\alpha}$
$$\vec{P}(\vec{x}, \vec{\alpha}) + \vec{I}_F \subseteq \vec{P}'(\vec{x}, \vec{\alpha}) + \overrightarrow{[0,0]}. \tag{A}$$

Observe

$$\vec{P}(\vec{x},\vec{\alpha}) = \underbrace{\vec{P}(\vec{x},0)}_{\vec{\alpha}\text{-indep.}} + \underbrace{\vec{P}(\vec{x},\vec{\alpha}) - \vec{P}(\vec{x},0)}_{\vec{\alpha}\text{-dependent}} = \vec{P}(\vec{x},0) + \vec{P}_\alpha(\vec{x},\vec{\alpha})$$

The size of $\vec{P}(\vec{x},0)$ is much larger than the rest, because the rest is essentially errors. The process of (A) does not alter $\vec{P}(\vec{x},0)$, so set the $\vec{\alpha}$-independent part $\vec{P}(\vec{x},0)$ aside from the whole process, which helps the numerical stability of the process.

The task is now

$$\vec{P}_\alpha(\vec{x},\vec{\alpha}) + \vec{I}_F \subseteq \vec{P}'_\alpha(\vec{x},\vec{\alpha}) + \overrightarrow{[0,0]}.$$

We limit $\vec{P}_\alpha(\vec{x},\vec{\alpha})$ to be only **linearly** dependent on $\vec{\alpha}$.

$$\vec{P}_\alpha(\vec{x},\vec{\alpha}) + \vec{I}_F = \left(\widehat{M} + \widehat{\bar{M}}(\vec{x})\right) \cdot \vec{\alpha} + \vec{I}_F.$$

Express $\vec{I}_F$ by the matrix form using additional parameters $\vec{\beta}$

$$\vec{I}_F \subseteq \left(\widehat{I}_F + \widehat{\bar{I}}_F(\vec{x})\right) \cdot \vec{\beta}.$$

where $\widehat{\bar{I}}_F(\vec{x}) = 0$ and $\left(\widehat{I}_F\right)_{ii} = \left|I_{Fi}\right|.$

$$\vec{P}_\alpha(\vec{x}, \vec{\alpha}) + \vec{I}_F \subseteq \left(\widehat{M} + \widehat{\bar{M}}(\vec{x})\right) \cdot \vec{\alpha} + \left(\widehat{I}_F + \widehat{\bar{I}}_F(\vec{x})\right) \cdot \vec{\beta}.$$

View this as a collection of $2 \cdot v$ column vectors associated to $2 \cdot v$ parameters $\vec{\alpha}$ and $\vec{\beta}$. Recall a matrix, or a collection of v column vectors, represent a parallelepiped. The problem is now to find a **set sum of two parallelepipeds**.

Psum Algorithm for choosing column vectors

Task: Choose v vectors out of n vectors $\vec{s}_i$, $i = 1, ..., n$, $n \geq v$.

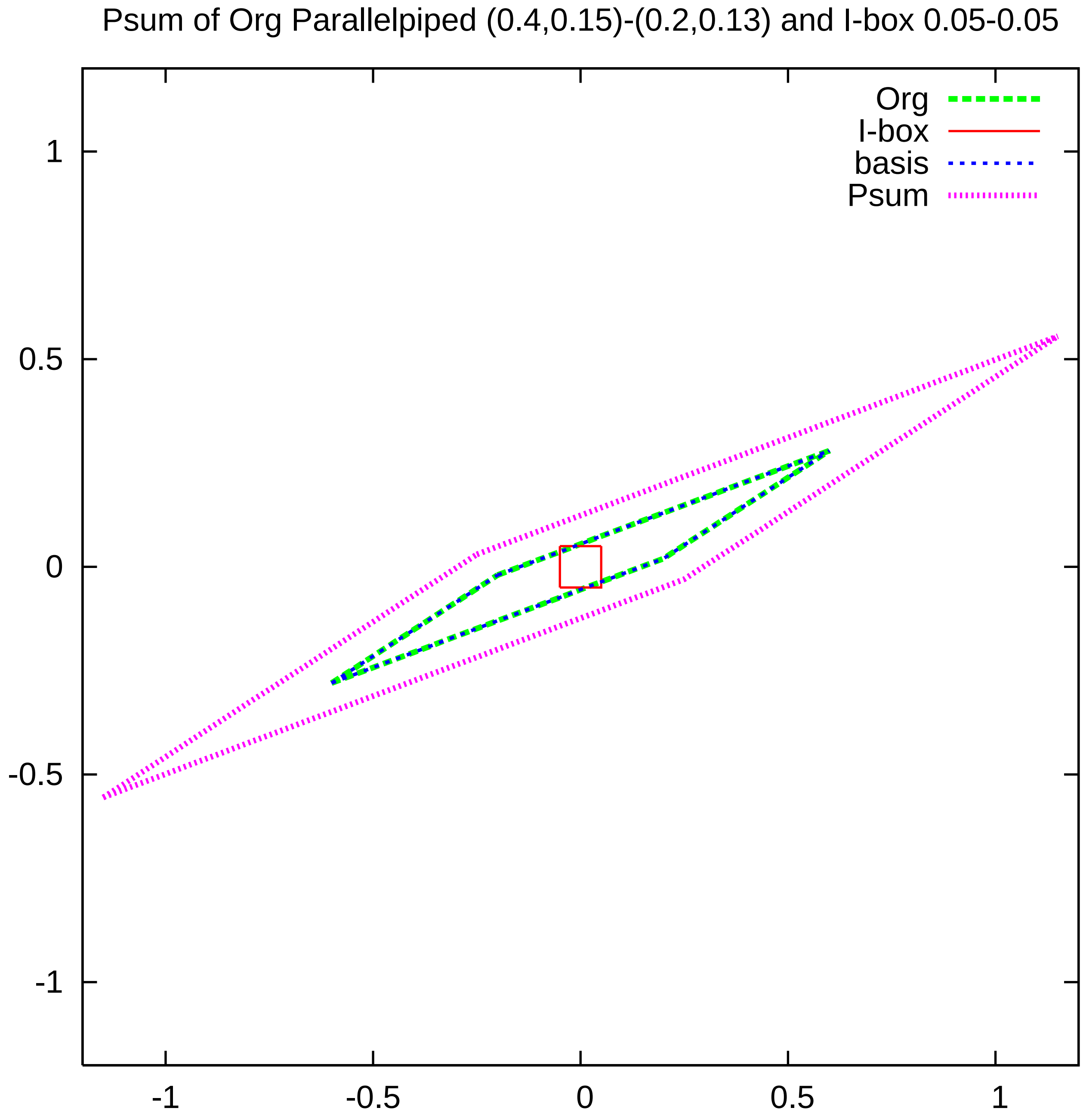
1. Choose the longest vector $\vec{s}_k$, and assign it as $\vec{t}_1$. Normalize it as $\vec{e}_1 = \vec{t}_1 / \left| \vec{t}_1 \right|$.
2. Out of the remaining vectors $\vec{s}_i$, choose the j-th vector $\vec{t}_j = \vec{s}_k$ such that
$$\frac{\left|\vec{s}_k\right|^2 - \sum_{m=1}^{j-1} \left|\vec{s}_k \cdot \vec{e}_m\right|^2}{\left|\vec{s}_k\right|^{2p}}$$
is largest. Compute $\vec{e}_j$, the orthonormalized vector of $\vec{t}_j$ to $\vec{e}_1, ..., \vec{e}_{j-1}$. (Gram-Schmidt)
3. Repeat the process 2 until $j = v$.

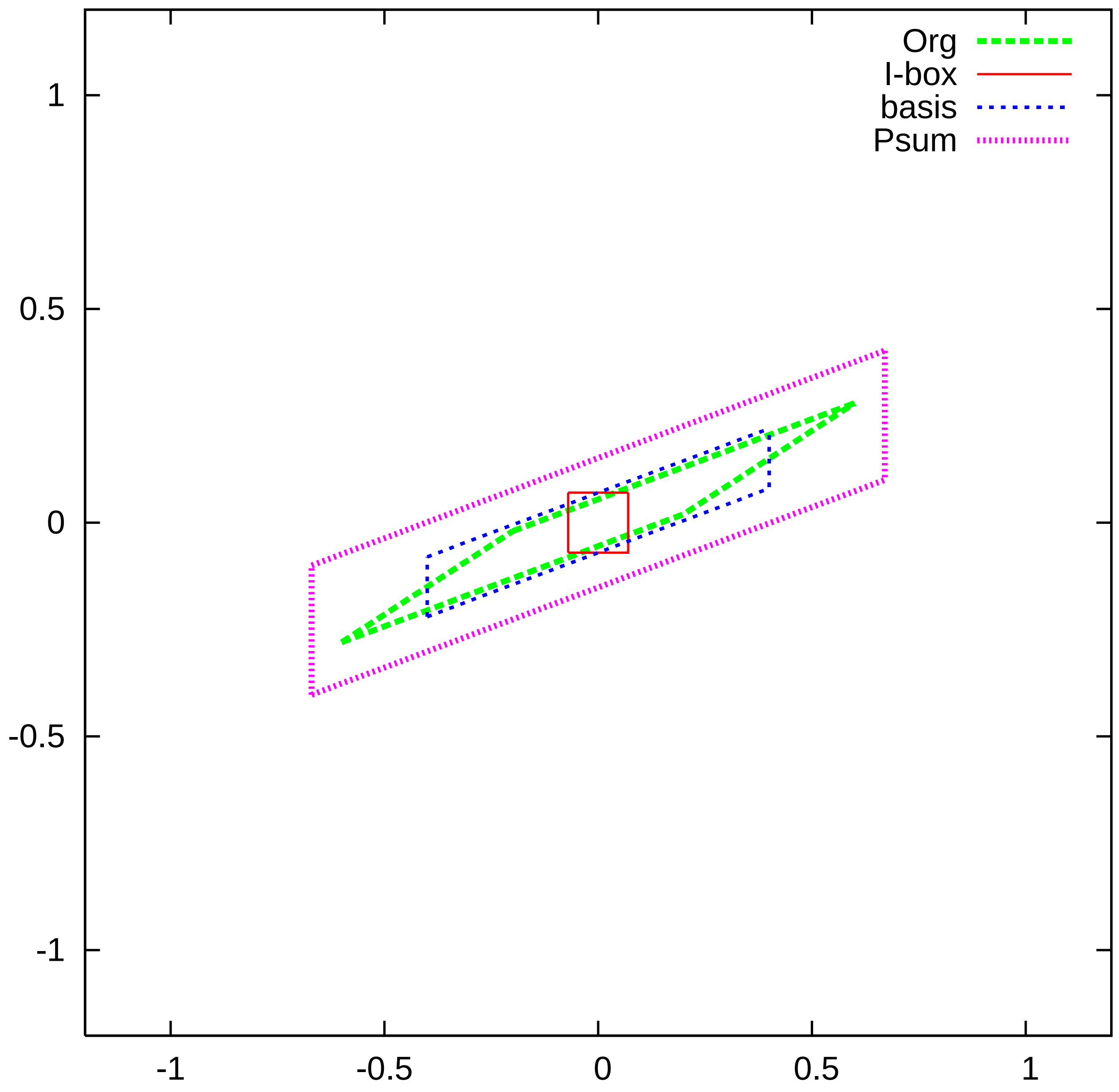
Experimentally, $p = 0.5$ is found to be efficient and robust for obtaining a set sum of two parallelepipeds

Psum Algorithm for two parallelepipeds

Task: Obtain a set sum of two parallelepipeds $\widehat{M_1}$ and $\widehat{M_2}$.

1. Prepare the basis $\widehat{M_b}$ using the Psum algorithm for choosing v column vectors out of $2 \cdot v$ column vectors from $\widehat{M_1}$ and $\widehat{M_2}$.
2. Compute conditioned parallelepipeds $\widehat{M_b}^{-1} \cdot \widehat{M_1}$ and $\widehat{M_b}^{-1} \cdot \widehat{M_2}$.
3. Confine the conditioned parallelepipeds by bounding them.
$$\vec{B}_1 = \text{bound}\left(\widehat{M_b}^{-1} \cdot \widehat{M_1}\right) \text{ and } \vec{B}_2 = \text{bound}\left(\widehat{M_b}^{-1} \cdot \widehat{M_2}\right).$$
4. Compute the interval sum $\vec{B} = \vec{B}_1 + \vec{B}_2$. $\vec{B}$ confines the conditioned set sum of the conditioned parallelepipeds.
5. From $\vec{B}$, set up a parallelepiped as a box $\widehat{B} = \begin{pmatrix} |B_1| & & 0 \\ & \ddots & \\ 0 & & |B_v| \end{pmatrix}.$
6. Compute $\widehat{M_b} \cdot \widehat{B}$, which is a set sum of $\widehat{M_1}$ and $\widehat{M_2}$ under $\widehat{M_b}$.


Psum of Org Parallelpiped (0.4,0.15)-(0.2,0.13) and I-box 0.05-0.05
Org
I-box
basis
Psum
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1



Psum of Org Parallelpiped (0.4,0.15)-(0.2,0.13) and I-box 0.07-0.07

Error Absorption

We now chose a favoured collection of v column vectors $\widehat{L}+\widehat{\bar{L}}(\vec{x})$ using the Psum algorithm. Collect the left over v column vectors to $\widehat{E}+\widehat{\bar{E}}(\vec{x})$. Associate them to $2 \cdot v$ parameters $\vec{\alpha}'$ and $\vec{\beta}'$.

$$\vec{P}_{\alpha}(\vec{x},\vec{\alpha})+\vec{I}_F \subseteq \left(\widehat{L}+\widehat{\bar{L}}(\vec{x})\right)\cdot\vec{\alpha}'+\left(\widehat{E}+\widehat{\bar{E}}(\vec{x})\right)\cdot\vec{\beta}'.$$

Since $\vec{\alpha}$ and $\vec{\beta}$ do not appear anymore, we can rename $\vec{\alpha}'$ and $\vec{\beta}'$ as $\vec{\alpha}$ and $\vec{\beta}$ for the simplicity.

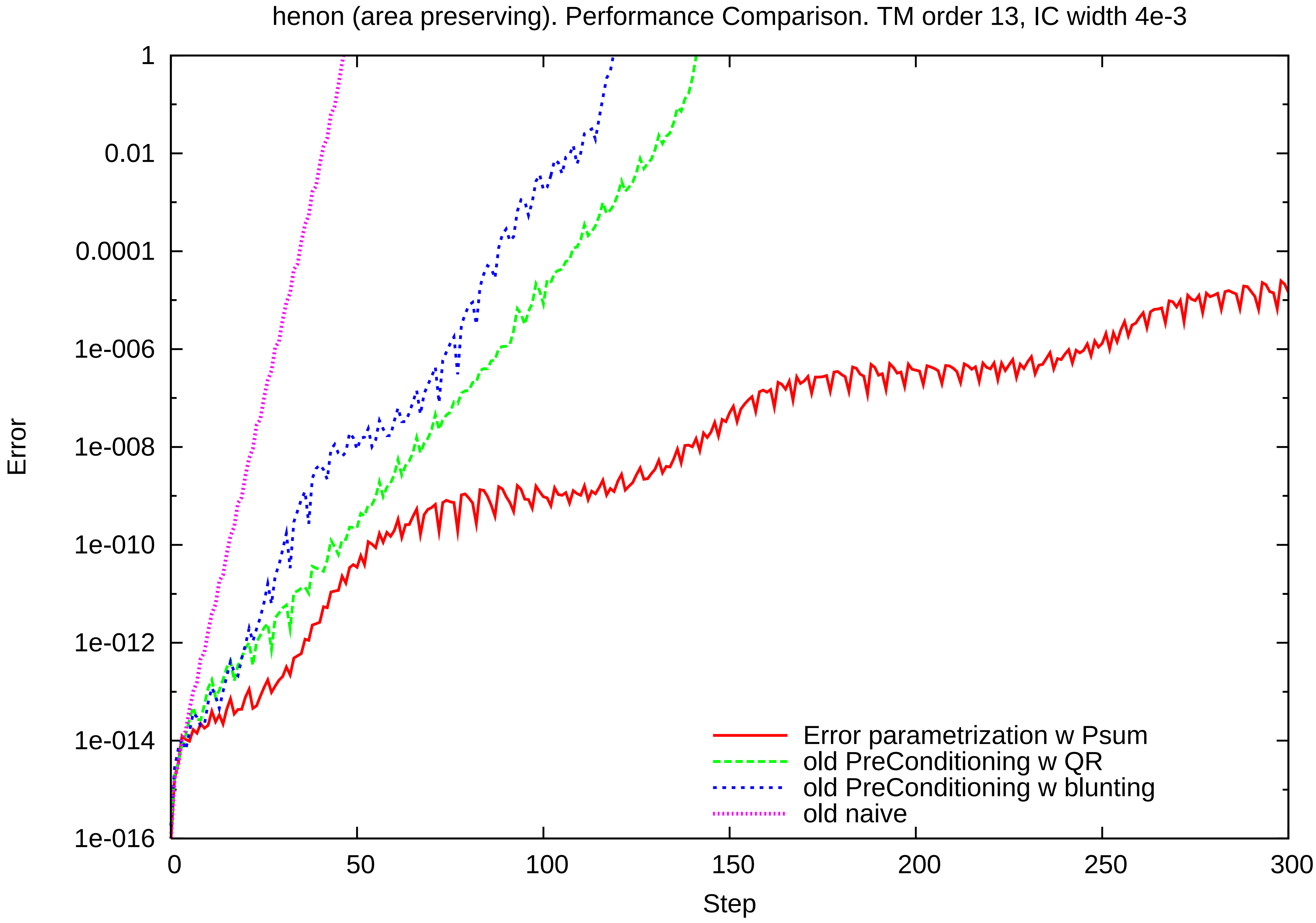
$$\begin{aligned}
\vec{P}_{\alpha}(\vec{x},\vec{\alpha})+\vec{I}_F &\subseteq \left(\widehat{L}+\widehat{\bar{L}}(\vec{x})\right)\cdot\vec{\alpha}+\left(\widehat{E}+\widehat{\bar{E}}(\vec{x})\right)\cdot\vec{\beta} \\
&= \widehat{L}\circ\left[\widehat{L}^{-1}\circ\left(\widehat{L}+\widehat{\bar{L}}(\vec{x})\right)\cdot\vec{\alpha}+\widehat{L}^{-1}\circ\left(\widehat{E}+\widehat{\bar{E}}(\vec{x})\right)\cdot\vec{\beta}\right] \\
&\subseteq \widehat{L}\circ\left[\left(\widehat{I}+\widehat{L}^{-1}\circ\widehat{\bar{L}}(\vec{x})\right)\cdot\vec{\alpha}+\widehat{B}\cdot\vec{\beta}\right]
\end{aligned}$$

where $\widehat{B}$ is a diagonal matrix with the i-th element is $|B_i|$ and $\vec{B}$ =bound$\left(\widehat{L}^{-1}\circ\left(\widehat{E}+\widehat{\bar{E}}(\vec{x})\right)\cdot\vec{\beta}\right)$.

If the **diagonal terms** of $\left(\widehat{I}+\widehat{L}^{-1}\circ\widehat{\bar{\bar{L}}}(\vec{x})\right)$ are **positive**,

$$\begin{aligned}
\vec{P}_\alpha(\vec{x},\vec{\alpha})+\vec{I}_F &\subseteq \widehat{L}\circ\left[\left(\widehat{I}+\widehat{L}^{-1}\circ\widehat{\bar{\bar{L}}}(\vec{x})\right)\cdot\vec{\alpha}+\widehat{B}\cdot\vec{\alpha}\right]\\
&=\widehat{L}\circ\left(\widehat{I}+\widehat{L}^{-1}\circ\widehat{\bar{\bar{L}}}(\vec{x})\right)\cdot\vec{\alpha}+\widehat{L}\circ\widehat{B}\cdot\vec{\alpha}\\
&=\left(\widehat{L}+\widehat{\bar{\bar{L}}}(\vec{x})+\widehat{L}\circ\widehat{B}\right)\cdot\vec{\alpha}.
\end{aligned}$$

Note: A modification to use $\widehat{A}$ instead of $\widehat{L}$, when $\widehat{A}\approx\widehat{L}$, is done easily. This involves bounding of $\widehat{A}^{-1}\circ\left(\widehat{L}-\widehat{A}\right)\cdot\vec{\alpha}$ and the diagonal terms to be checked positive are those of $\left(\widehat{I}+\widehat{A}^{-1}\circ\widehat{\bar{\bar{L}}}(\vec{x})\right)$.


henon (area preserving). Performance Comparison. TM order 13, IC width 4e-3
Error
Step
1
0.01
0.0001
1e-006
1e-008
1e-010
1e-012
1e-014
1e-016
0
50
100
150
200
250
300
Error parametrization w Psum
old PreConditioning w QR
old PreConditioning w blunting
old naive

Cost of Additional Parameters

For a v dimensional system, we need v parameters $\vec{\alpha}$ to absorb Taylor model remainder error bound intervals. The dependence on $\vec{\alpha}$ is limited to **linear**. So, we use weighted DA. Choose an appropriate weight order w for $\vec{\alpha}$.

- The dependence on $\vec{\alpha}$ has to be kept linear. Namely $2 \cdot w > n$, where n is the computational order of Taylor models. Choose

$$w = \text{Int}\left(\frac{n}{2}\right) + 1.$$

Maximum size necessary for DA and TM for $v = 2$.

n	v	DA	TM	v	DA	TM		w	v_w	DA	TM
13	2	105	140	$2+2$	2380	2419	$\Rightarrow$	7	$2+2_w$	161	200
21	2	253	304	$2+2$	12650	12705		11	$2+2_w$	385	440
33	2	595	670	$2+2$	66045	66124		17	$2+2_w$	901	980

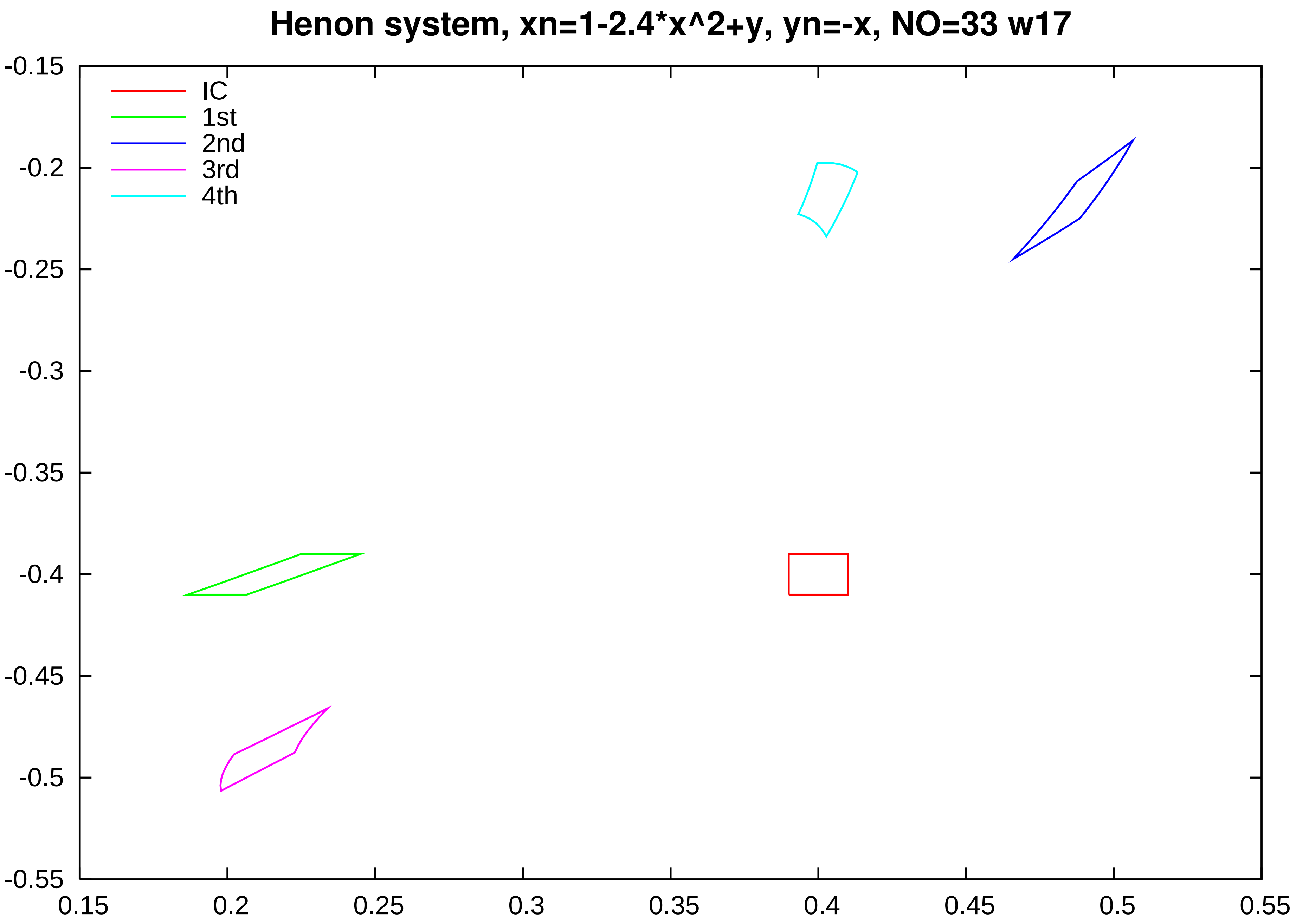
Dynamic Domain Decomposition

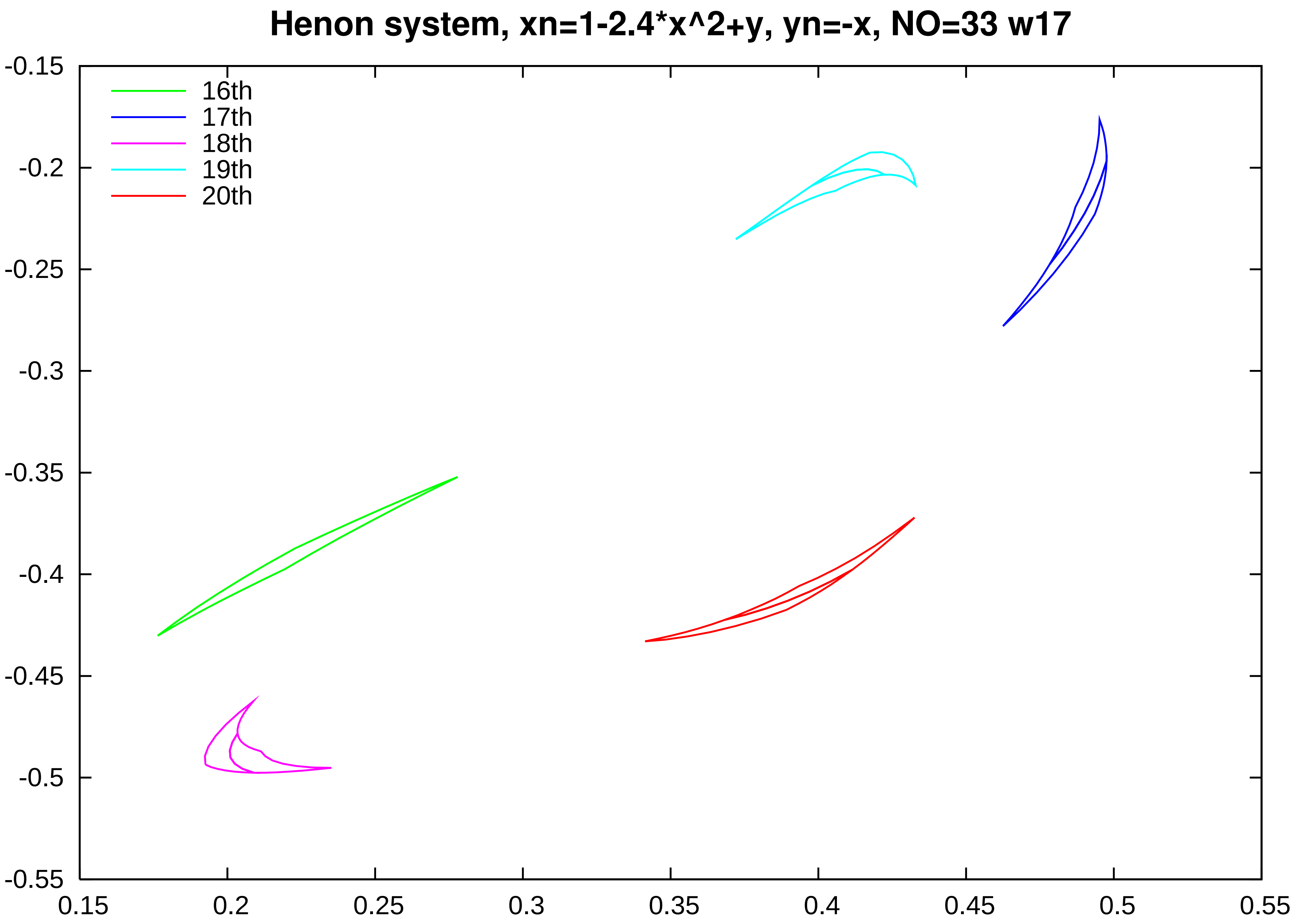
For extended domains, this is **natural equivalent** to step size control. Similarity to what's done in global optimization.

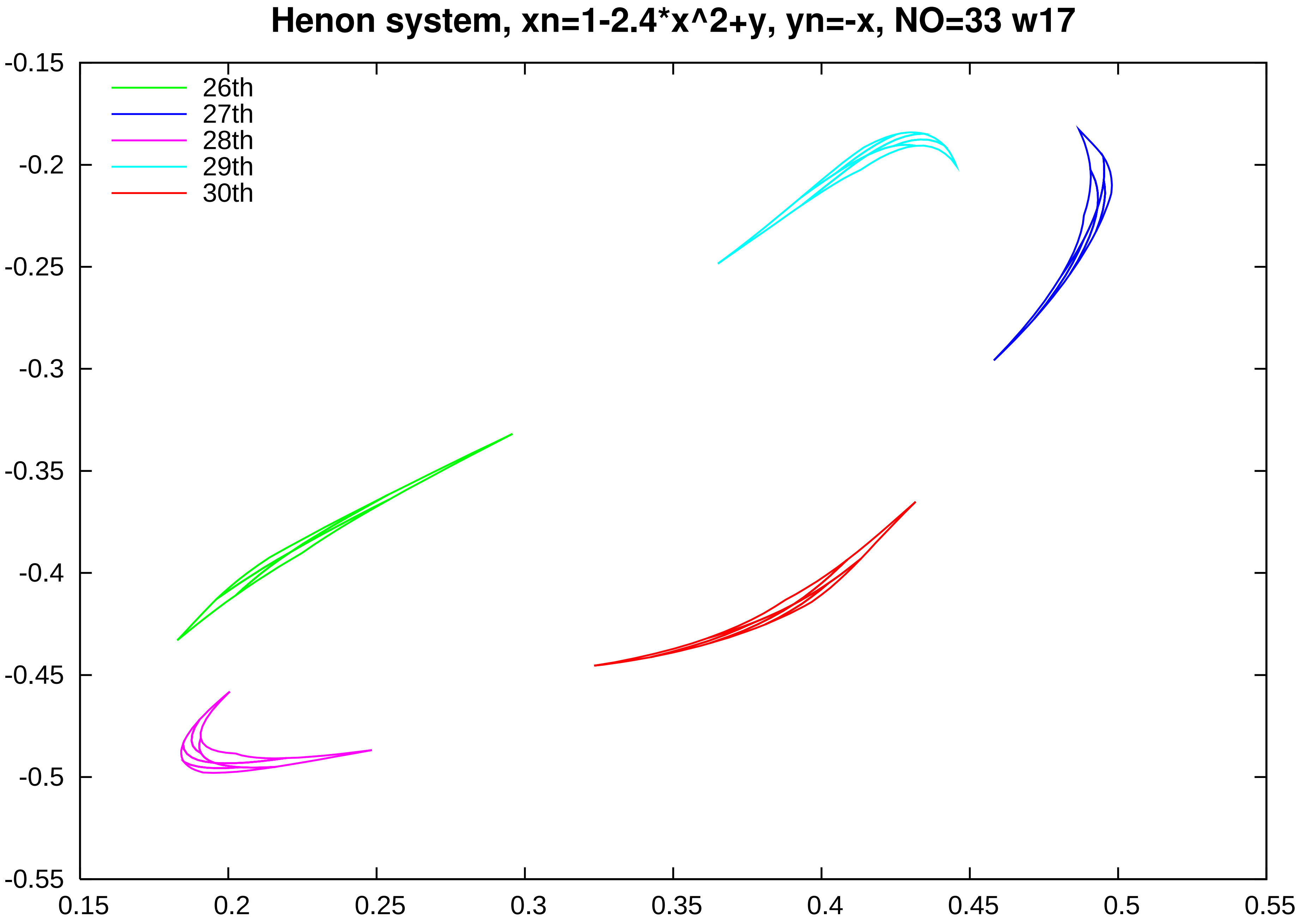
1. Evaluate ODE for $\Delta t = 0$ for current flow.
2. If resulting remainder bound R greater than ε, split the domain along variable leading to longest axis.
3. Absorb R in the TM polynomial part using the error parametrization method. If it fails, split the domain along variable leading to largest x dependence of the error.
4. Put one half of the box on stack for future work.

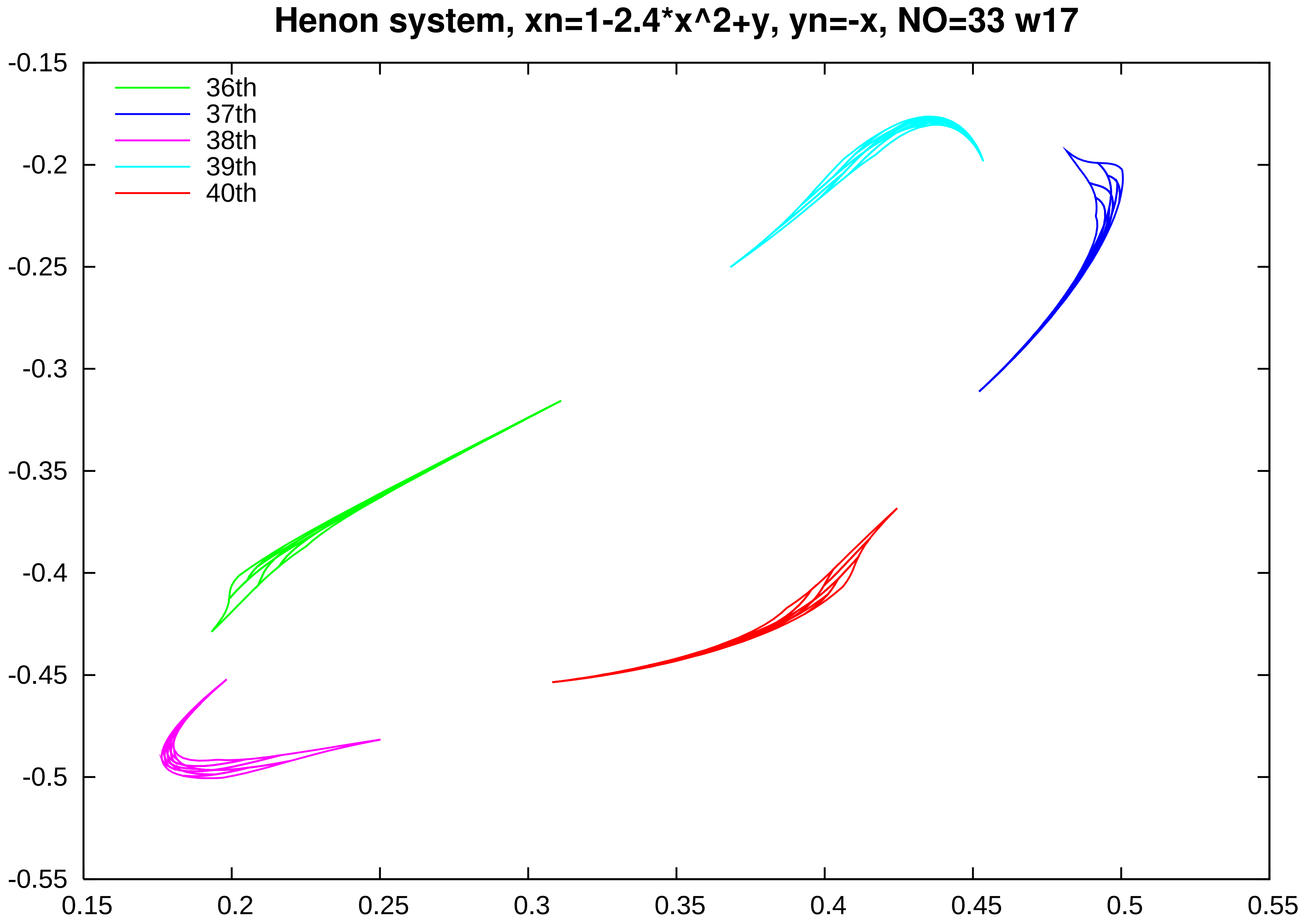
Things to consider:

- Utilize "First-in-last-out" stack; minimizes stack length. Special adjustments for stack management in a parallel environment, including load balancing.
- Outlook: also dynamic order control for dependence on initial conditions

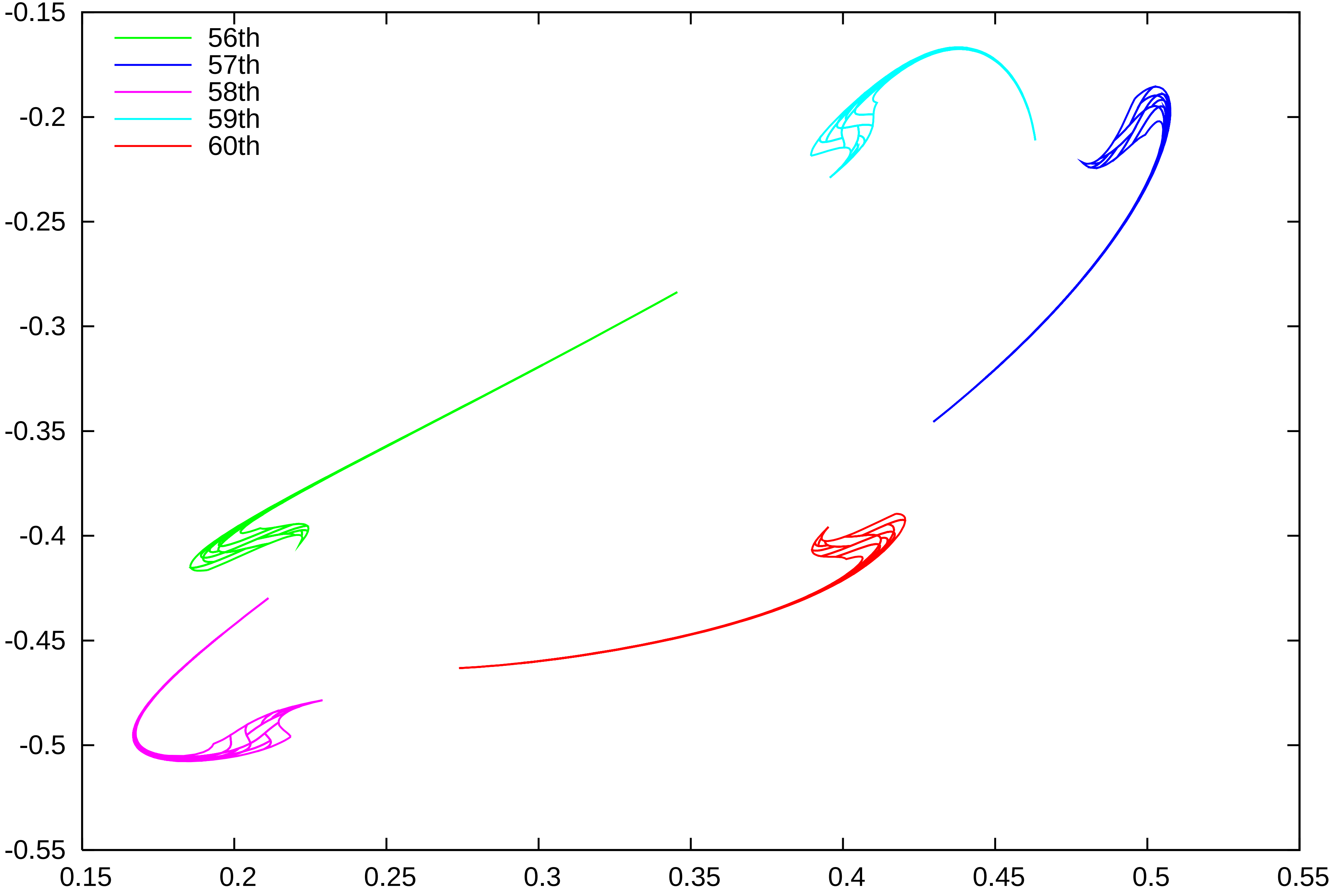

Henon system, xn=1-2.4*x^2+y, yn=-x, NO=33 w17
IC
1st
2nd
3rd
4th
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55

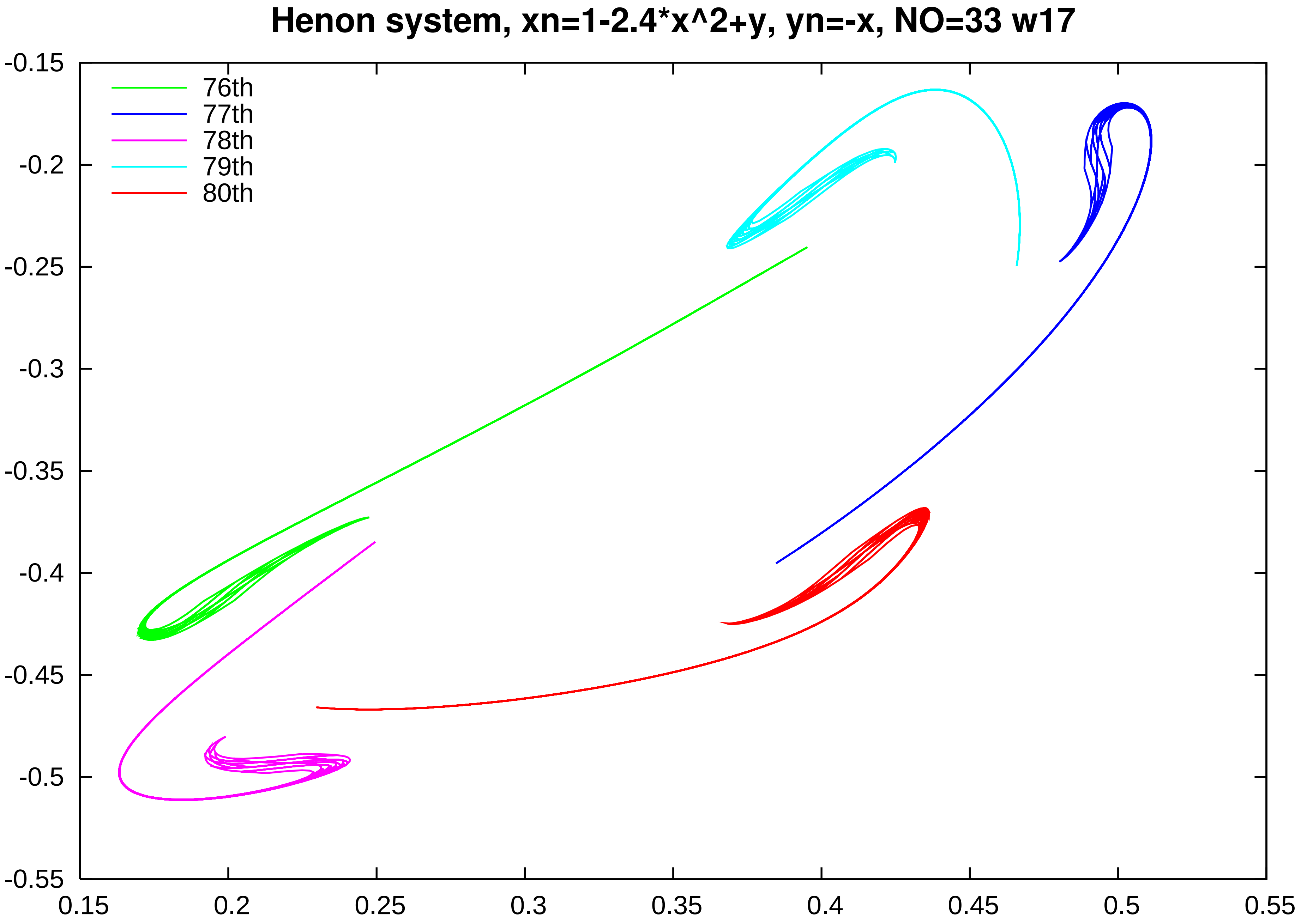

Henon system, xn=1-2.4*x^2+y, yn=-x, NO=33 w17
16th
17th
18th
19th
20th
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55


Henon system, xn=1-2.4*x^2+y, yn=-x, NO=33 w17
26th
27th
28th
29th
30th
-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

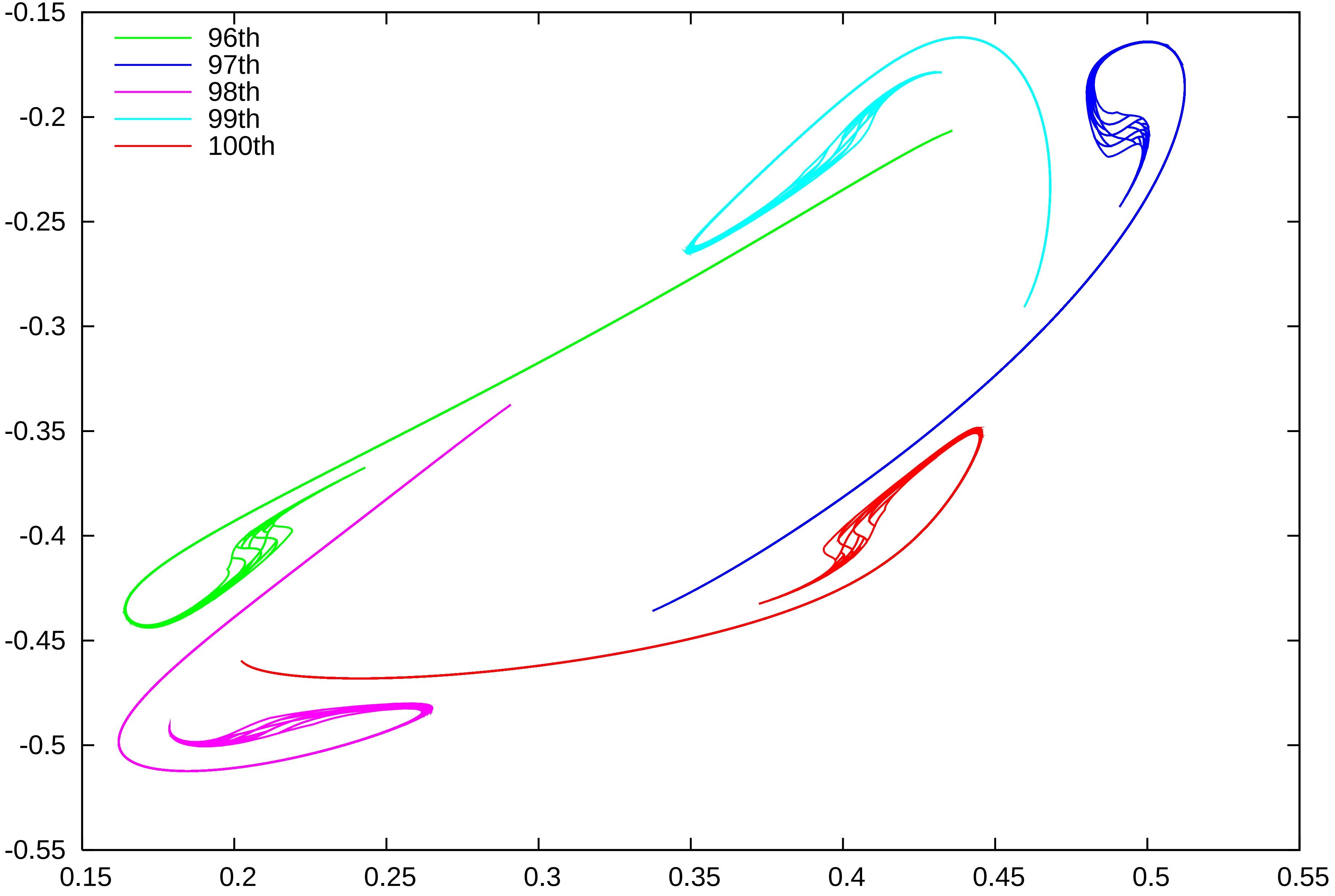

Henon system, xn=1-2.4*x^2+y, yn=-x, NO=33 w17
36th
37th
38th
39th
40th
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55

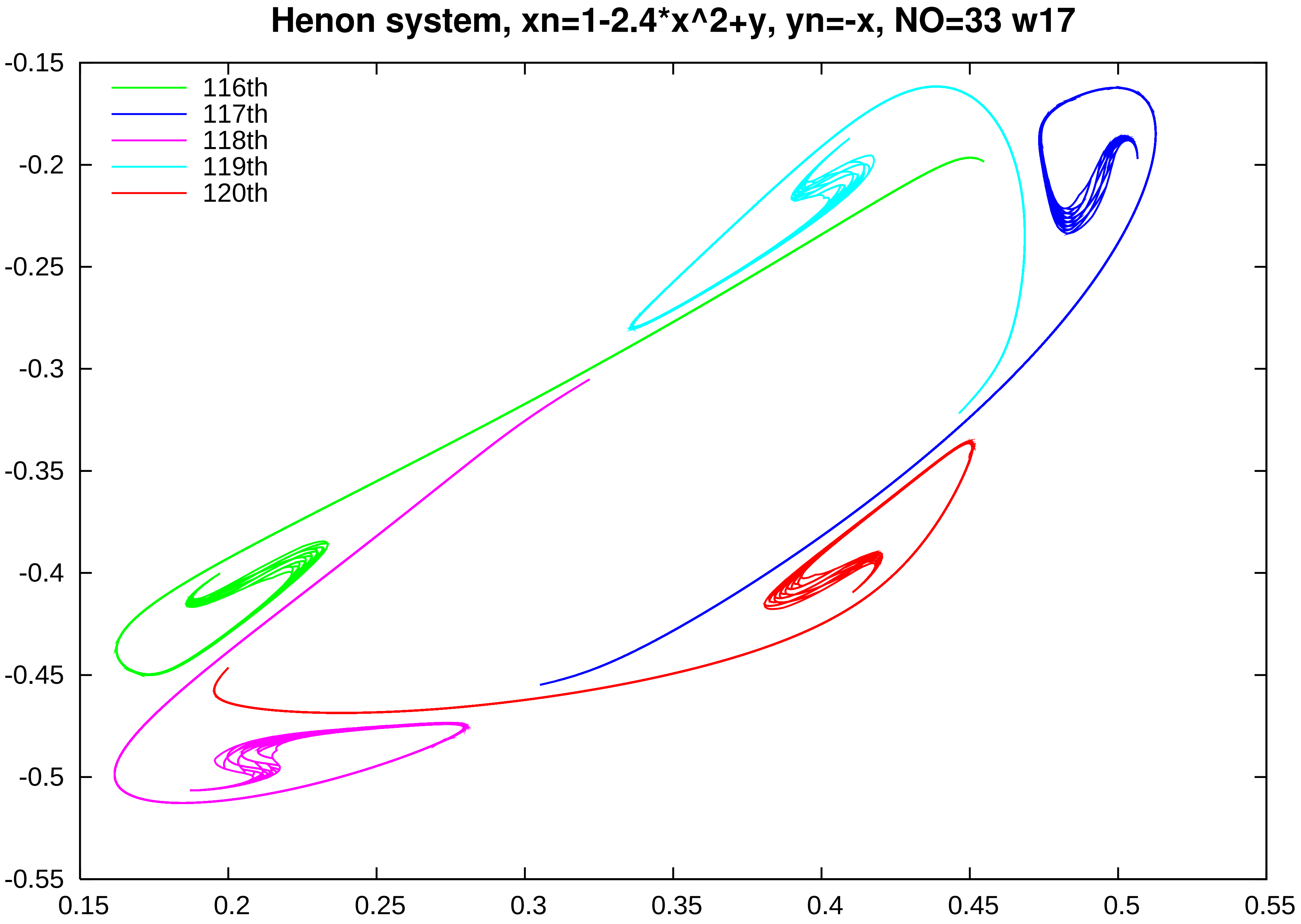
Henon system, xn=1-2.4*x^2+y, yn=-x, NO=33 w17

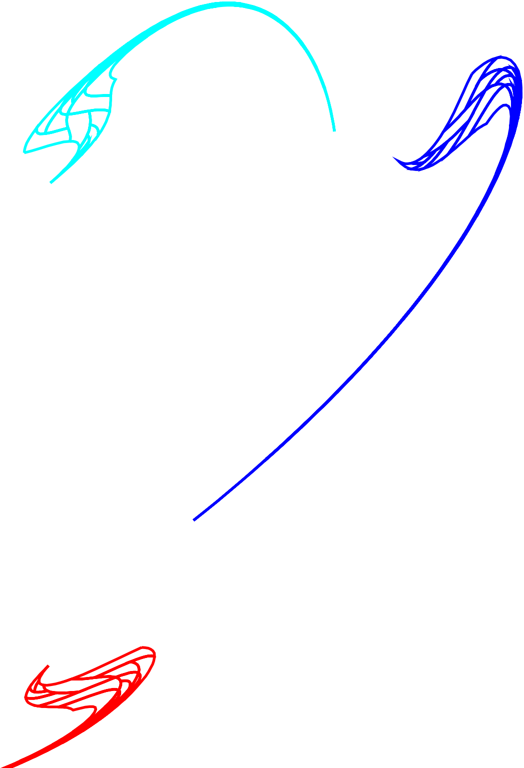


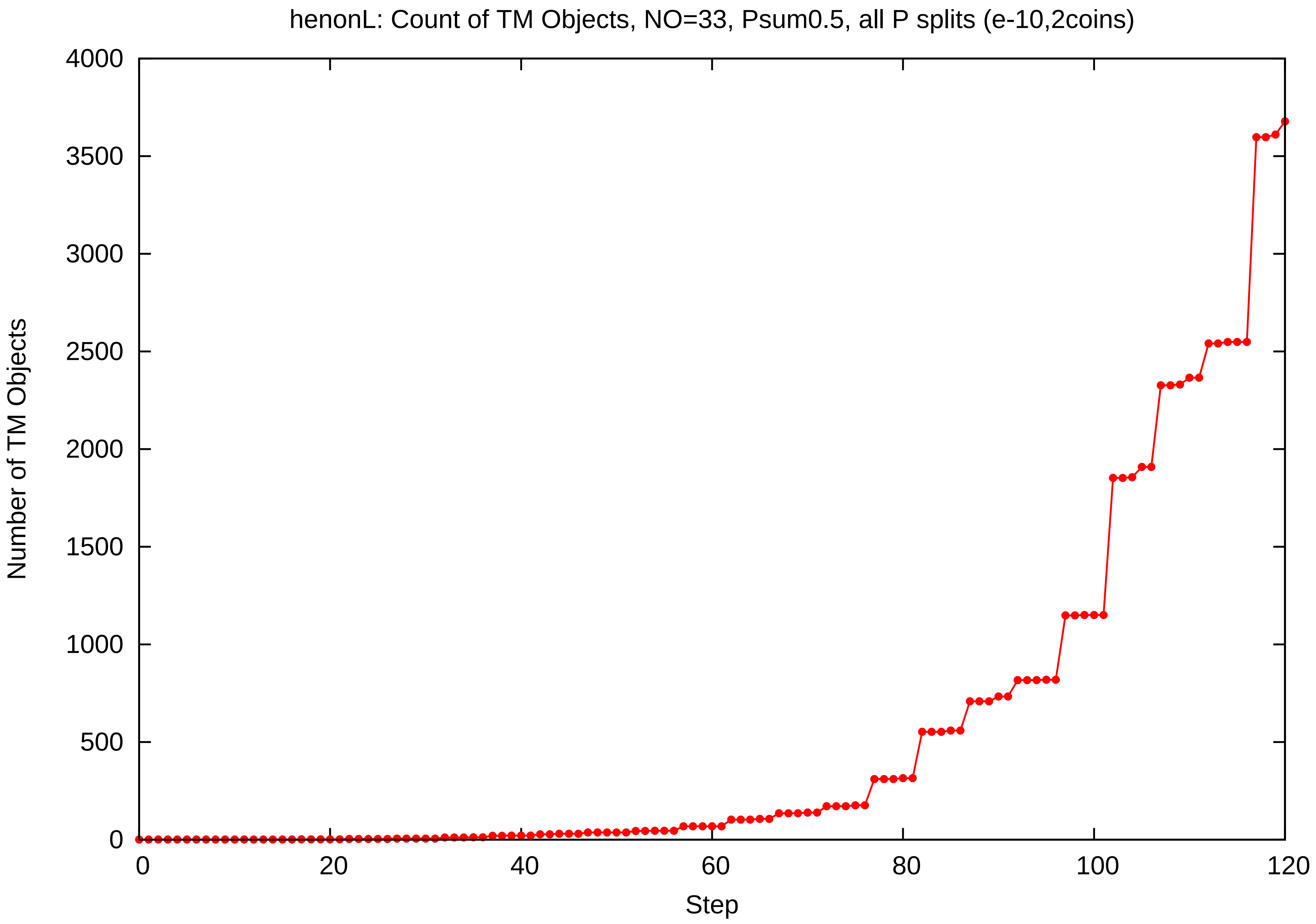

Henon system, xn=1-2.4*x^2+y, yn=-x, NO=33 w17
76th
77th
78th
79th
80th
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55

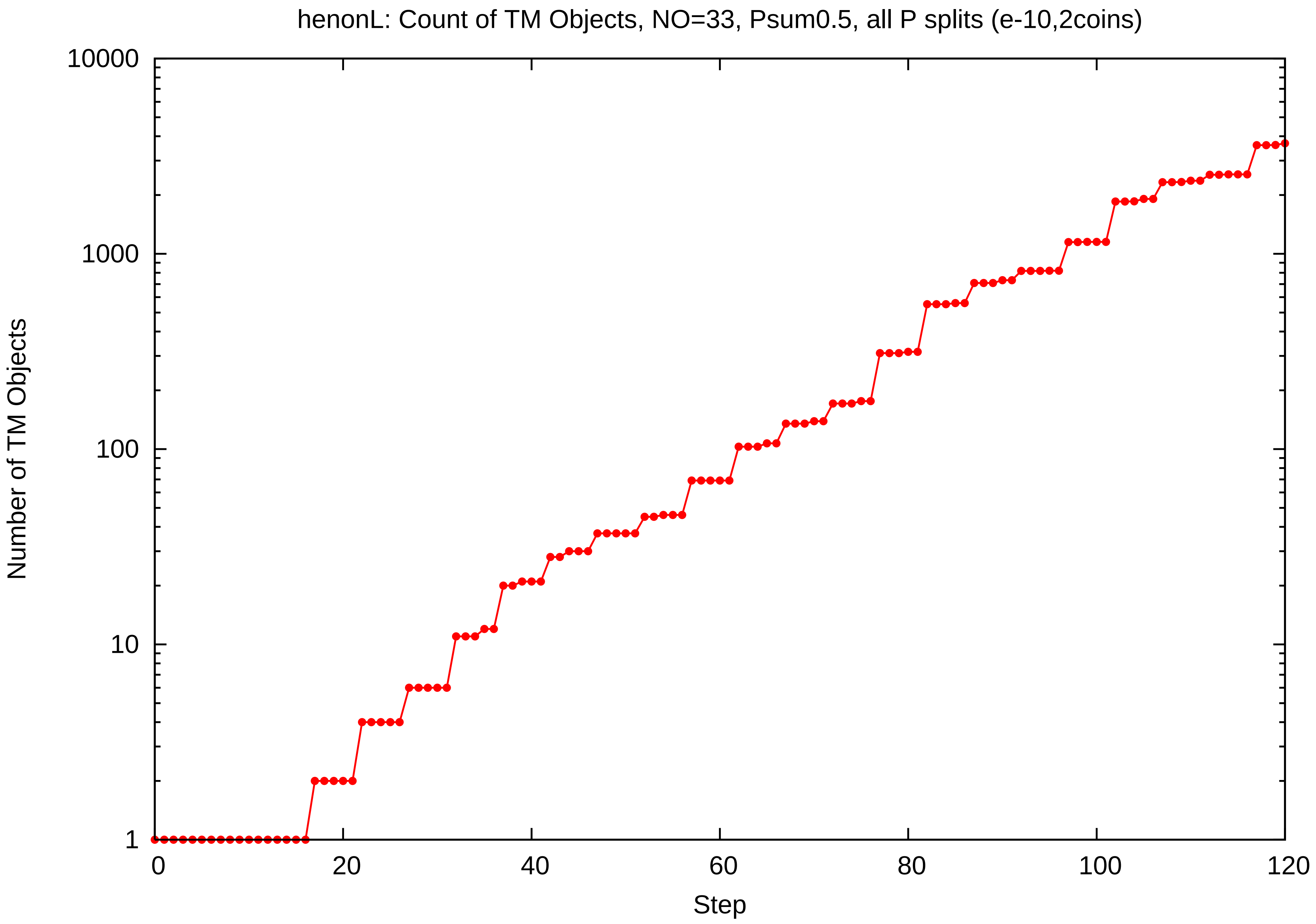
Henon system, xn=1-2.4*x^2+y, yn=-x, NO=33 w17



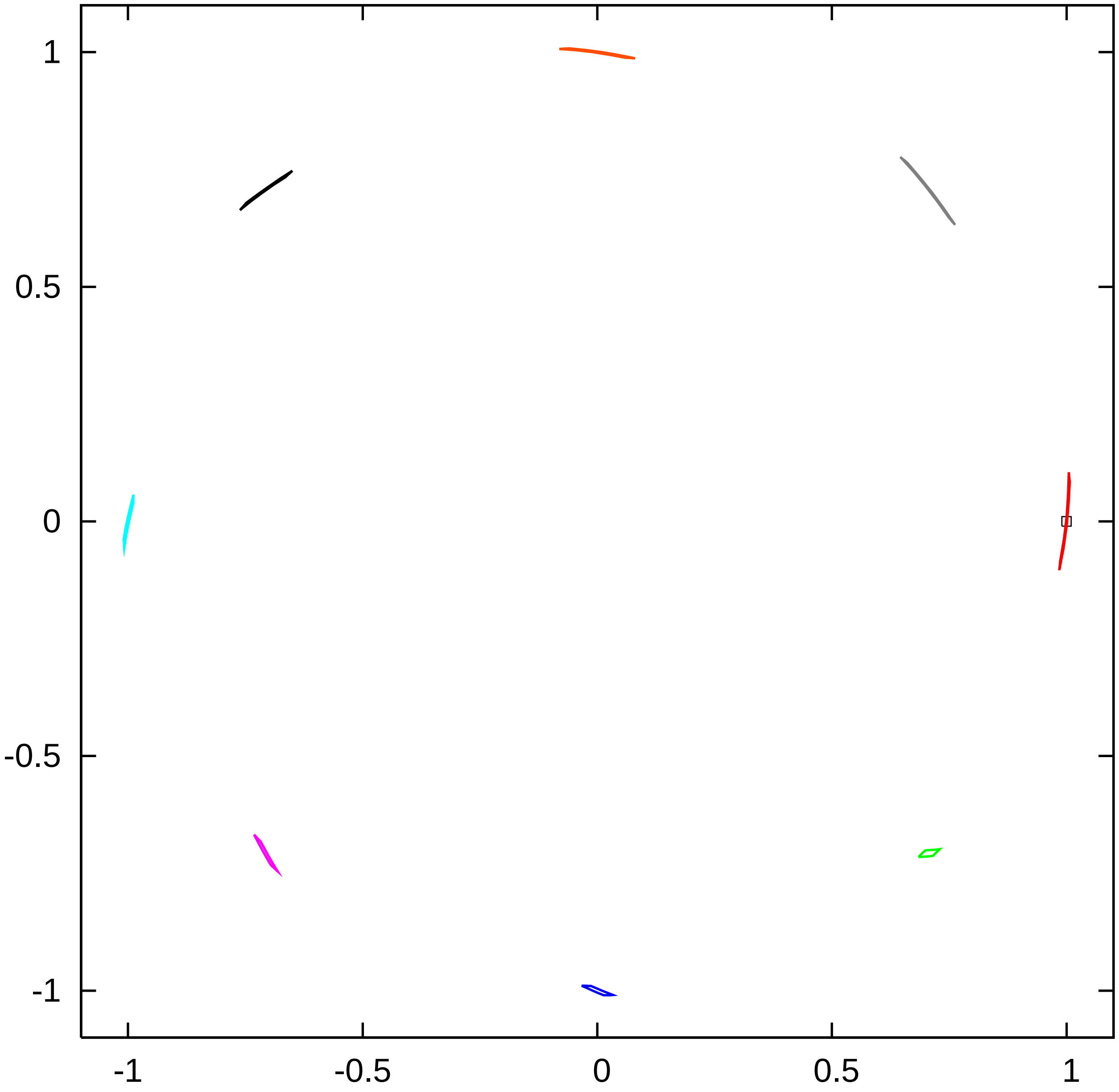

Henon system, xn=1-2.4*x^2+y, yn=-x, NO=33 w17
116th
117th
118th
119th
120th
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55



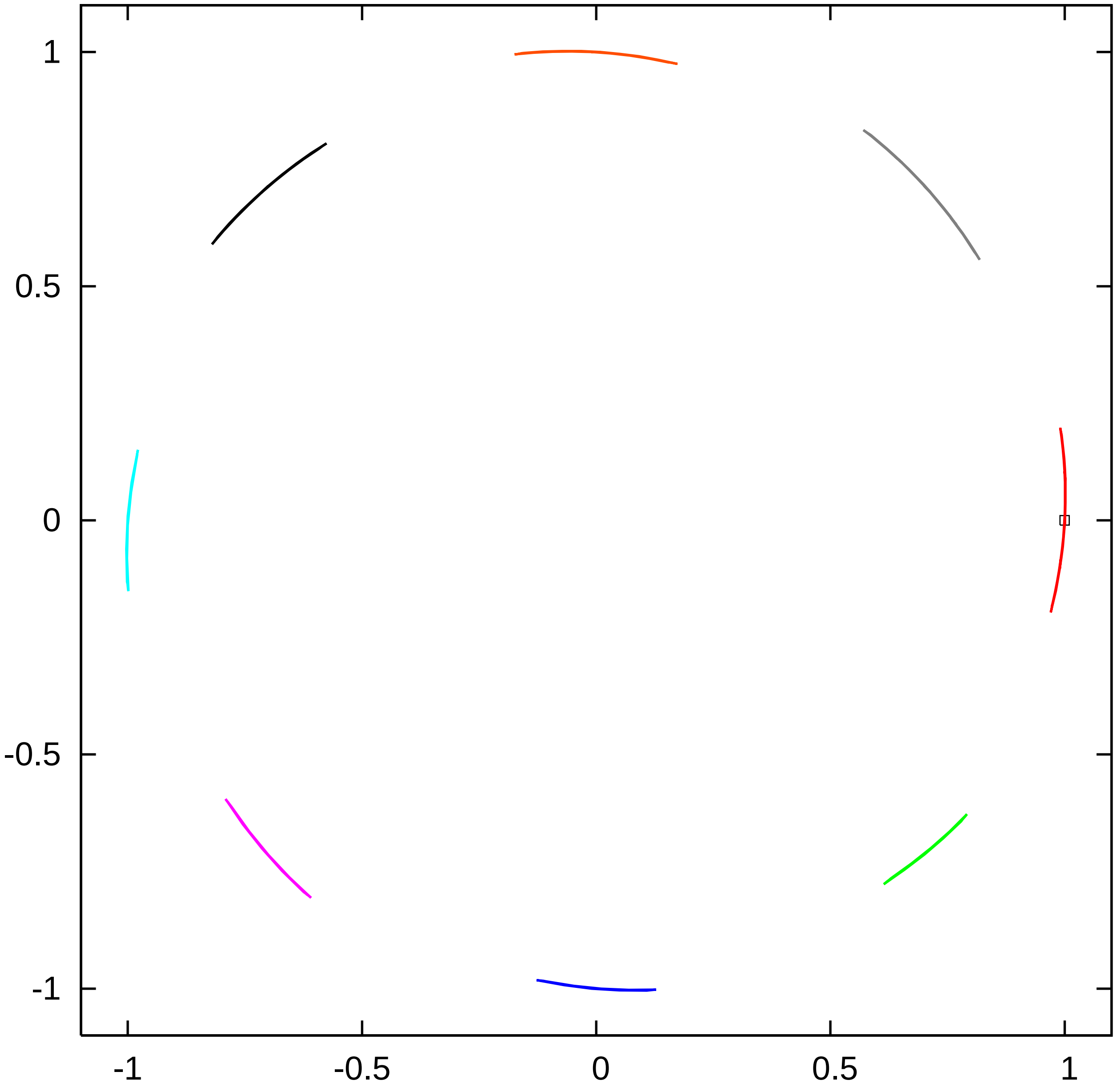

henonL: Count of TM Objects, NO=33, Psum0.5, all P splits (e-10,2coins)
Number of TM Objects
4000
3500
3000
2500
2000
1500
1000
500
0
0
20
40
60
80
100
120
Step


henonL: Count of TM Objects, NO=33, Psum0.5, all P splits (e-10,2coins)
Number of TM Objects
Step
10000
1000
100
10
1
0
20
40
60
80
100
120

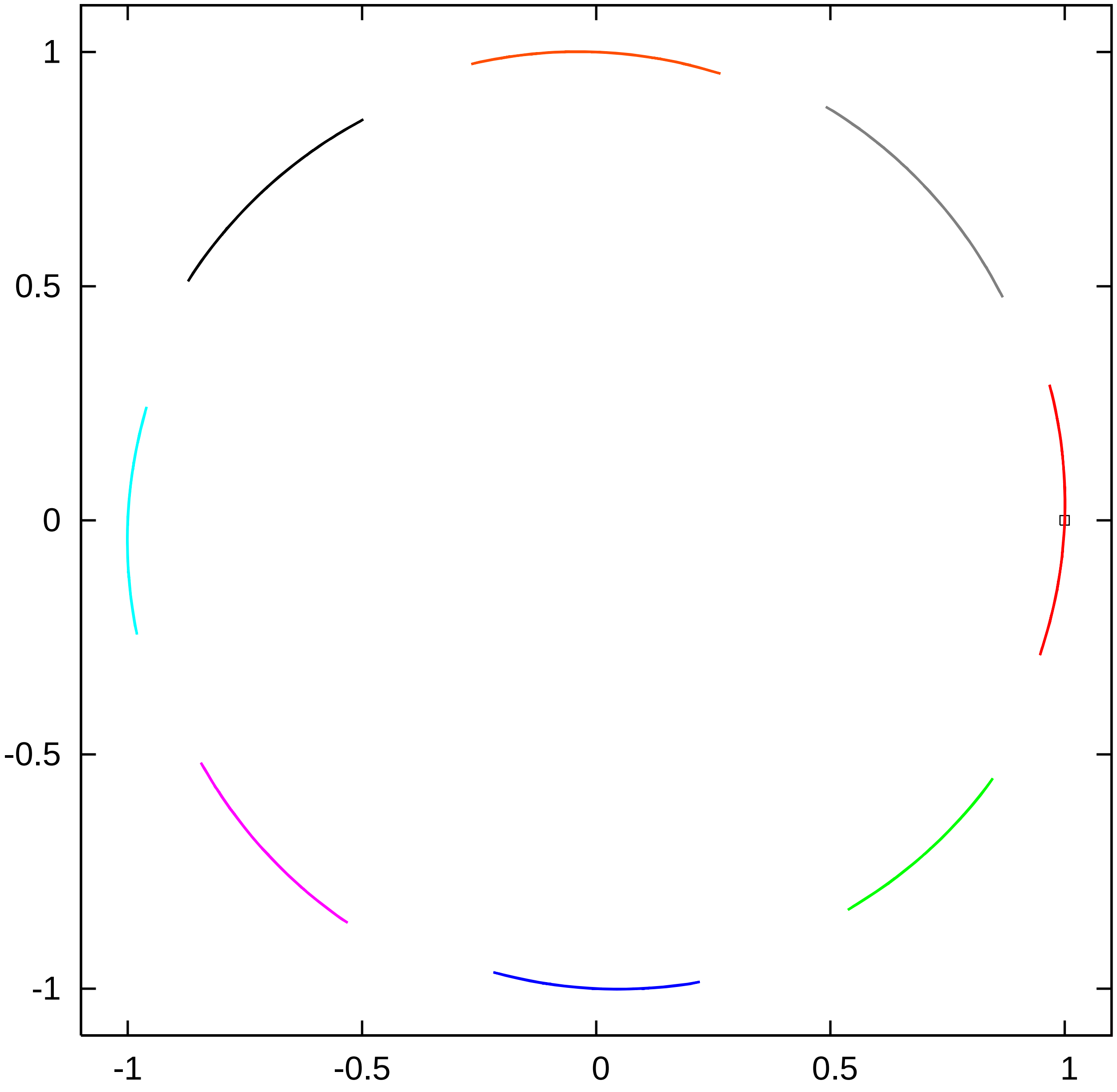
discrete kepler. 1st revolution, ICw 0.02, NO=13 w7

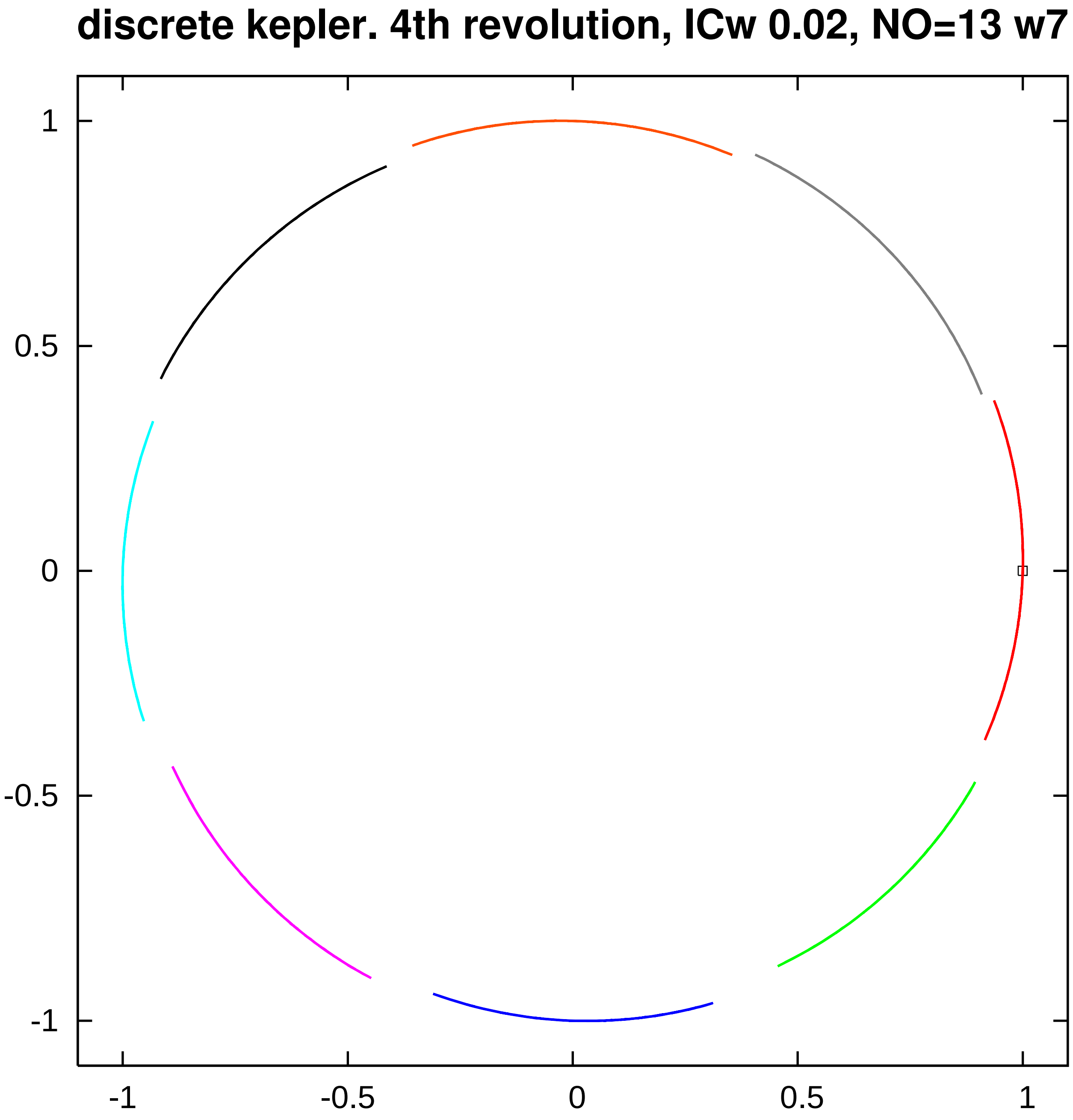


discrete kepler. 2nd revolution, ICw 0.02, NO=13 w7

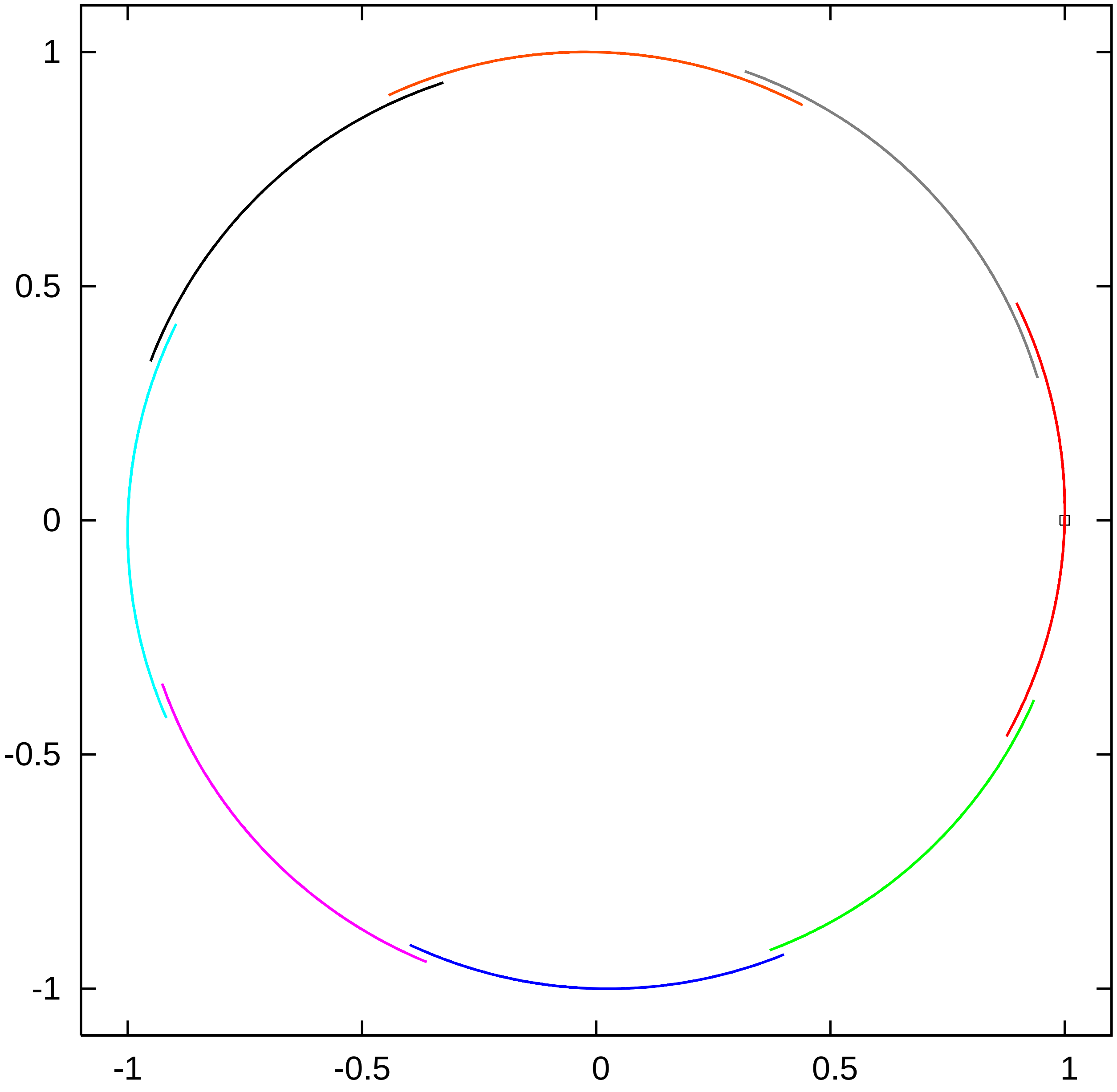


discrete kepler. 3rd revolution, ICw 0.02, NO=13 w7

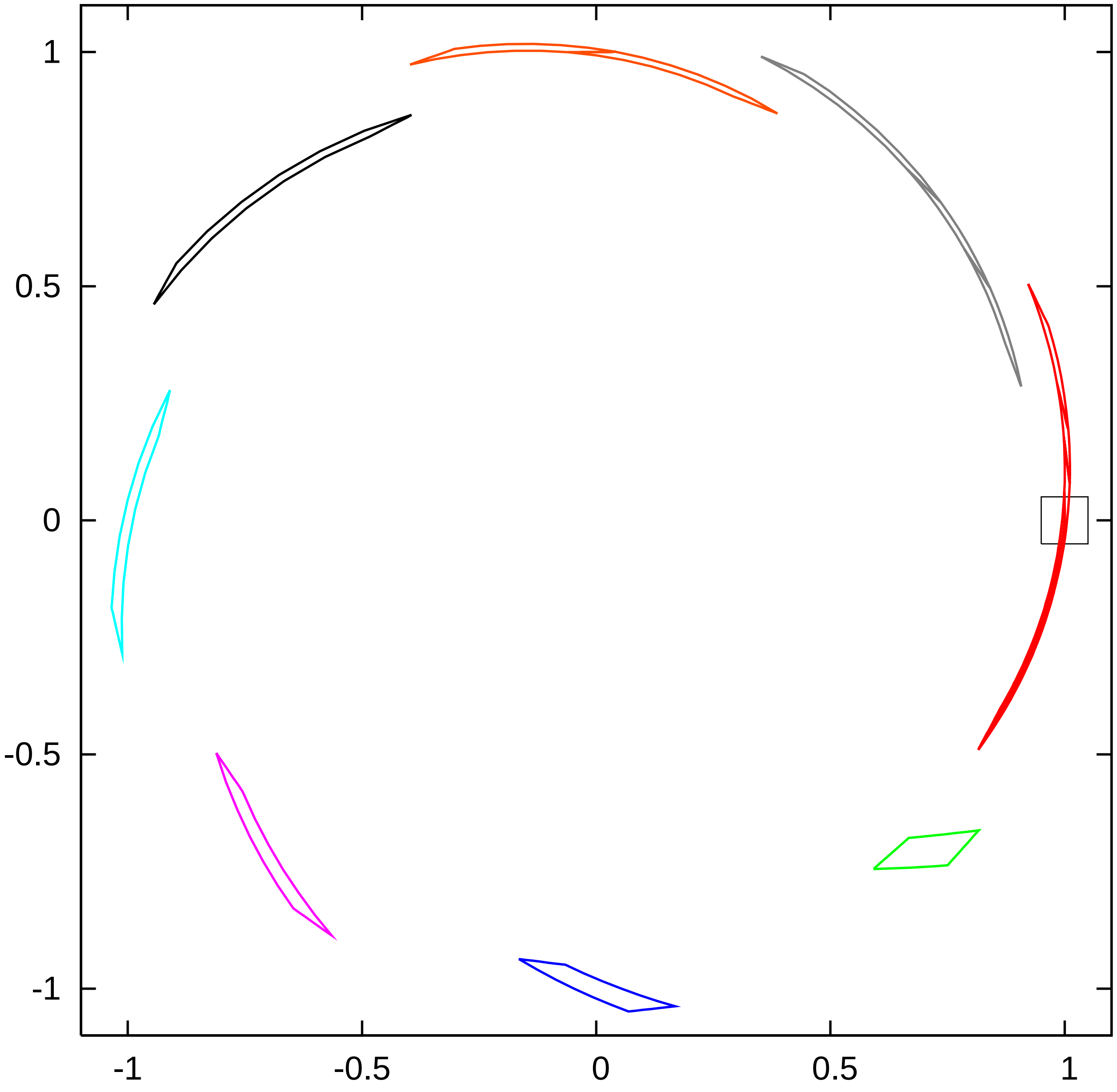


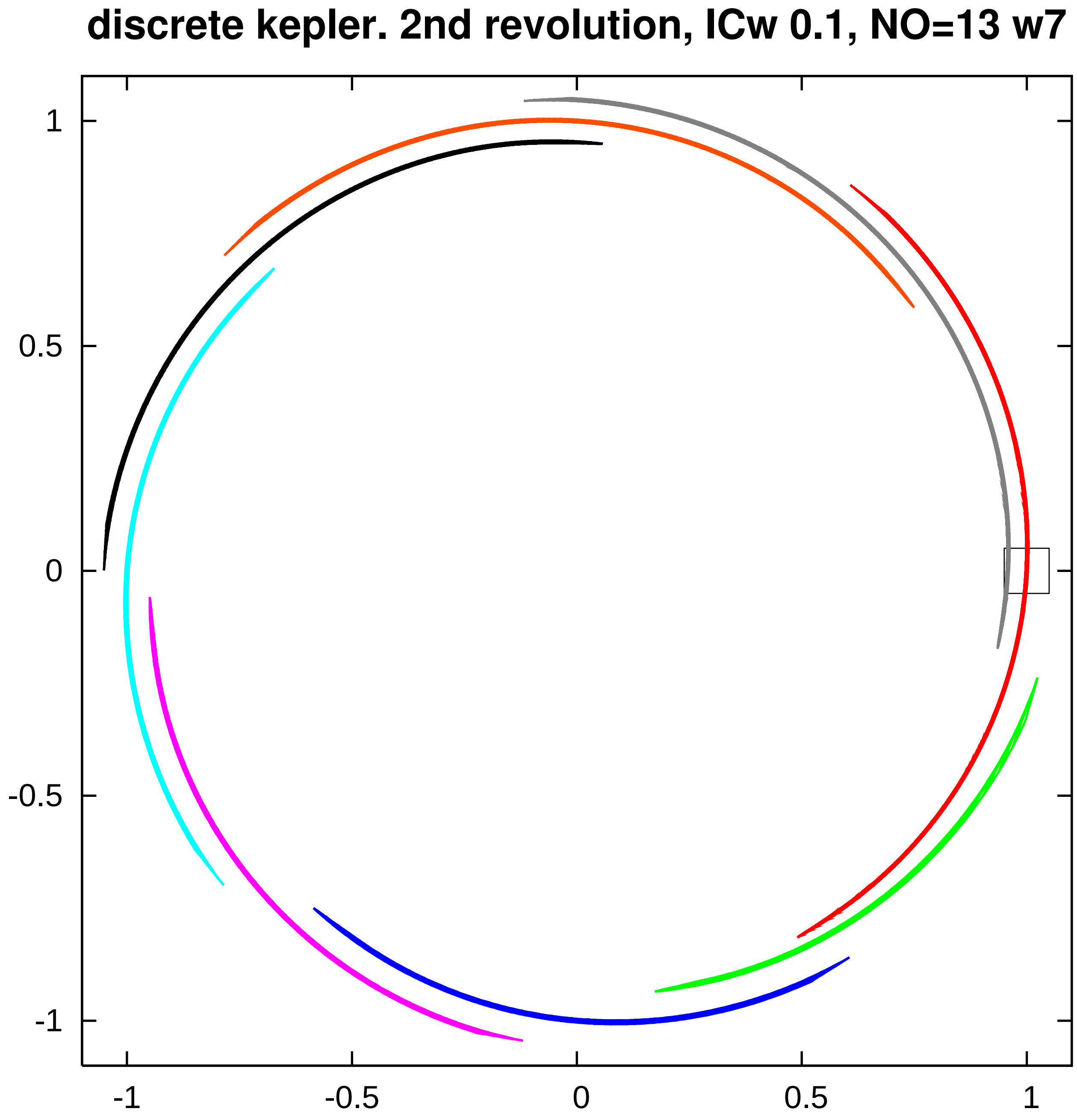

discrete kepler. 4th revolution, ICw 0.02, NO=13 w7
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

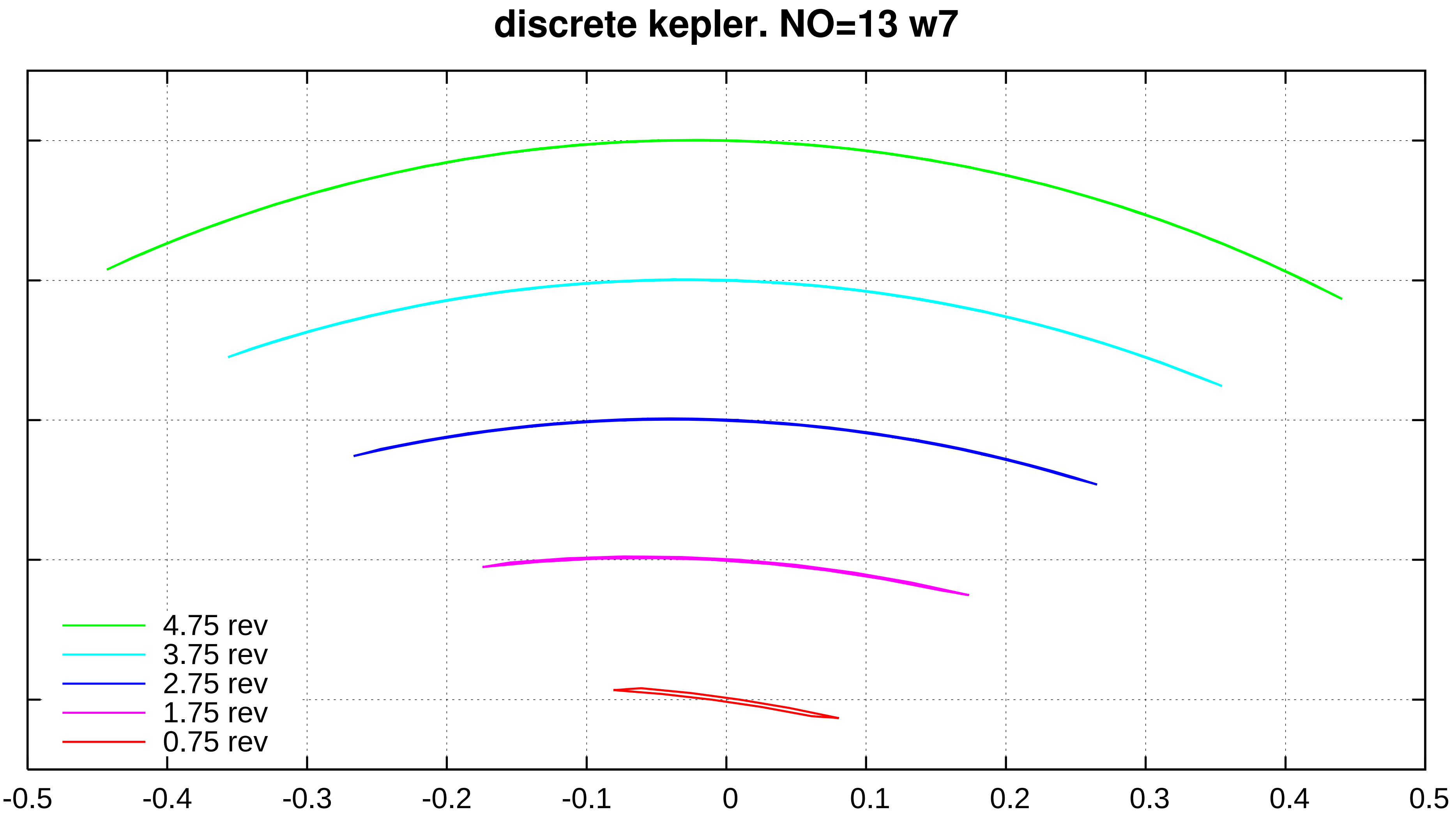
discrete kepler. 5th revolution, ICw 0.02, NO=13 w7



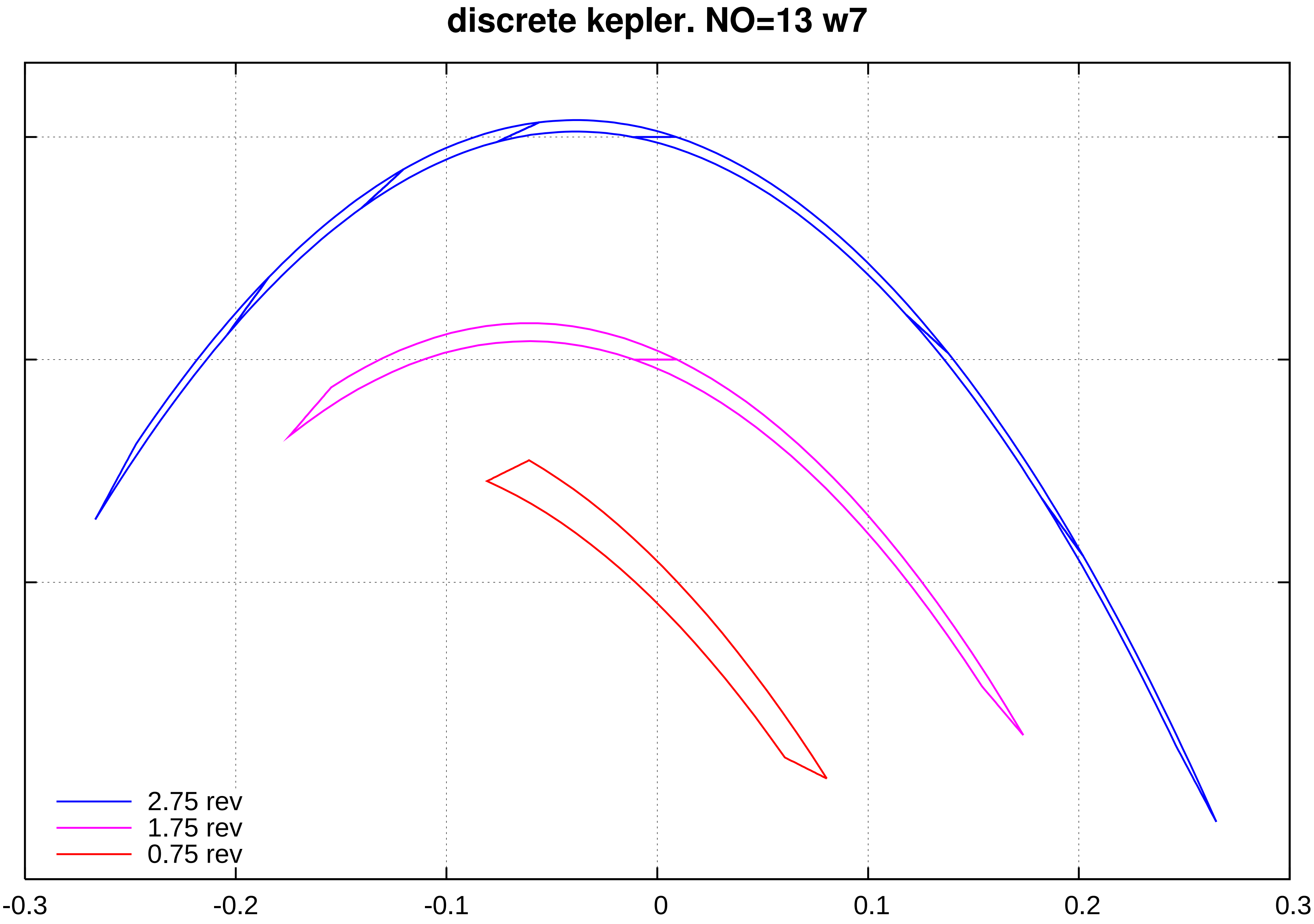
discrete kepler. 1st revolution, ICw 0.1, NO=13 w7

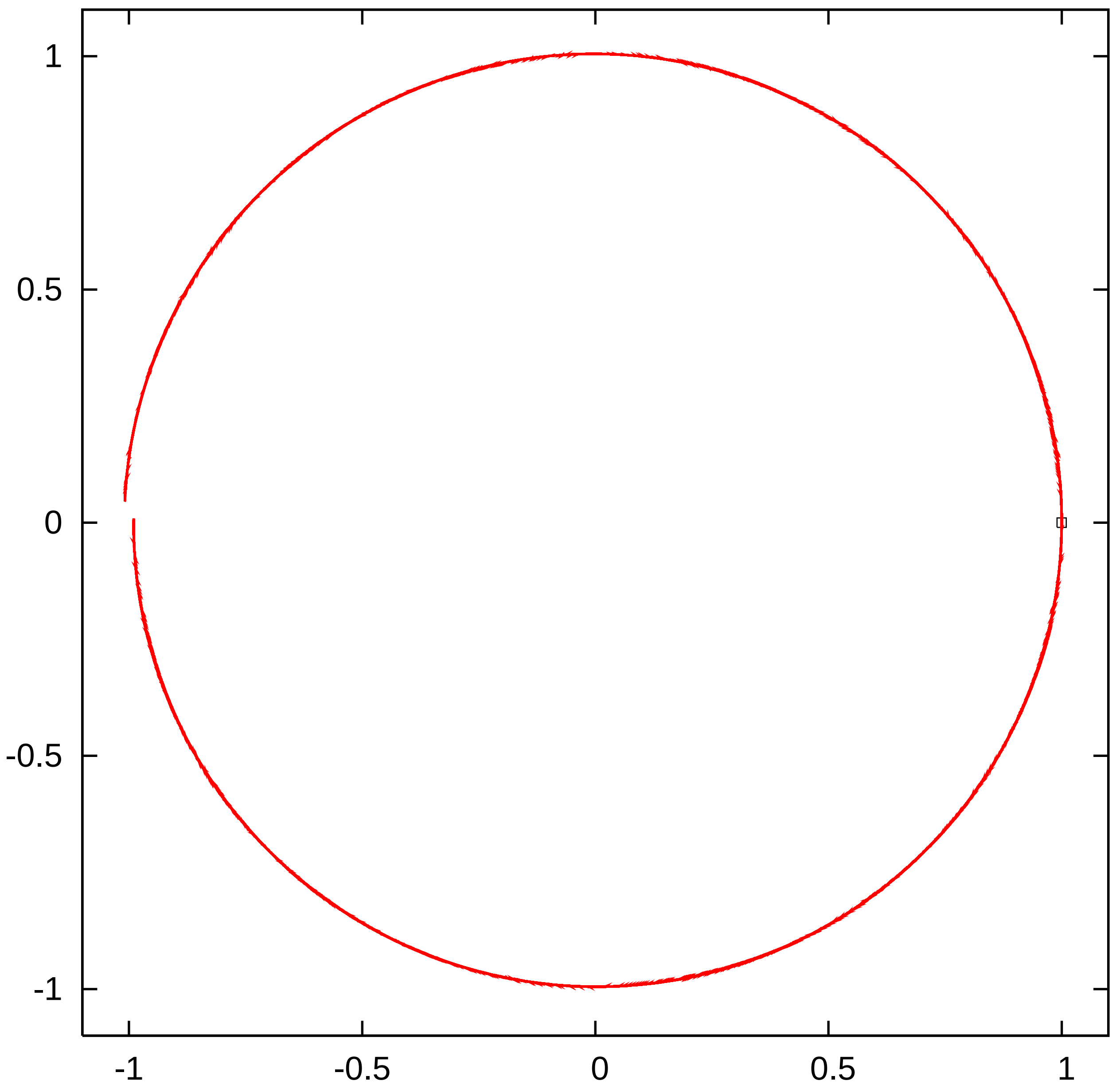



discrete kepler. 2nd revolution, ICw 0.1, NO=13 w7
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

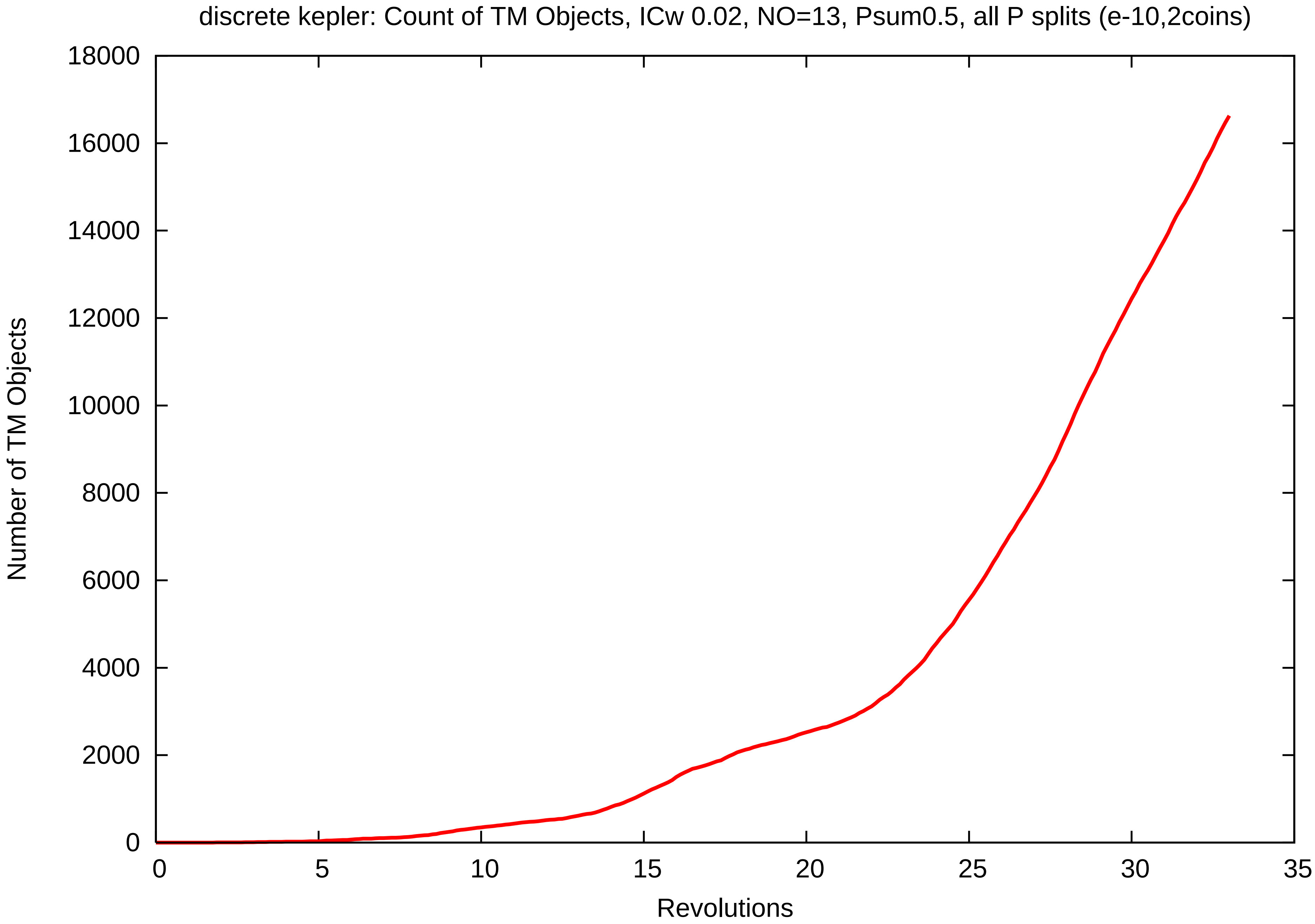


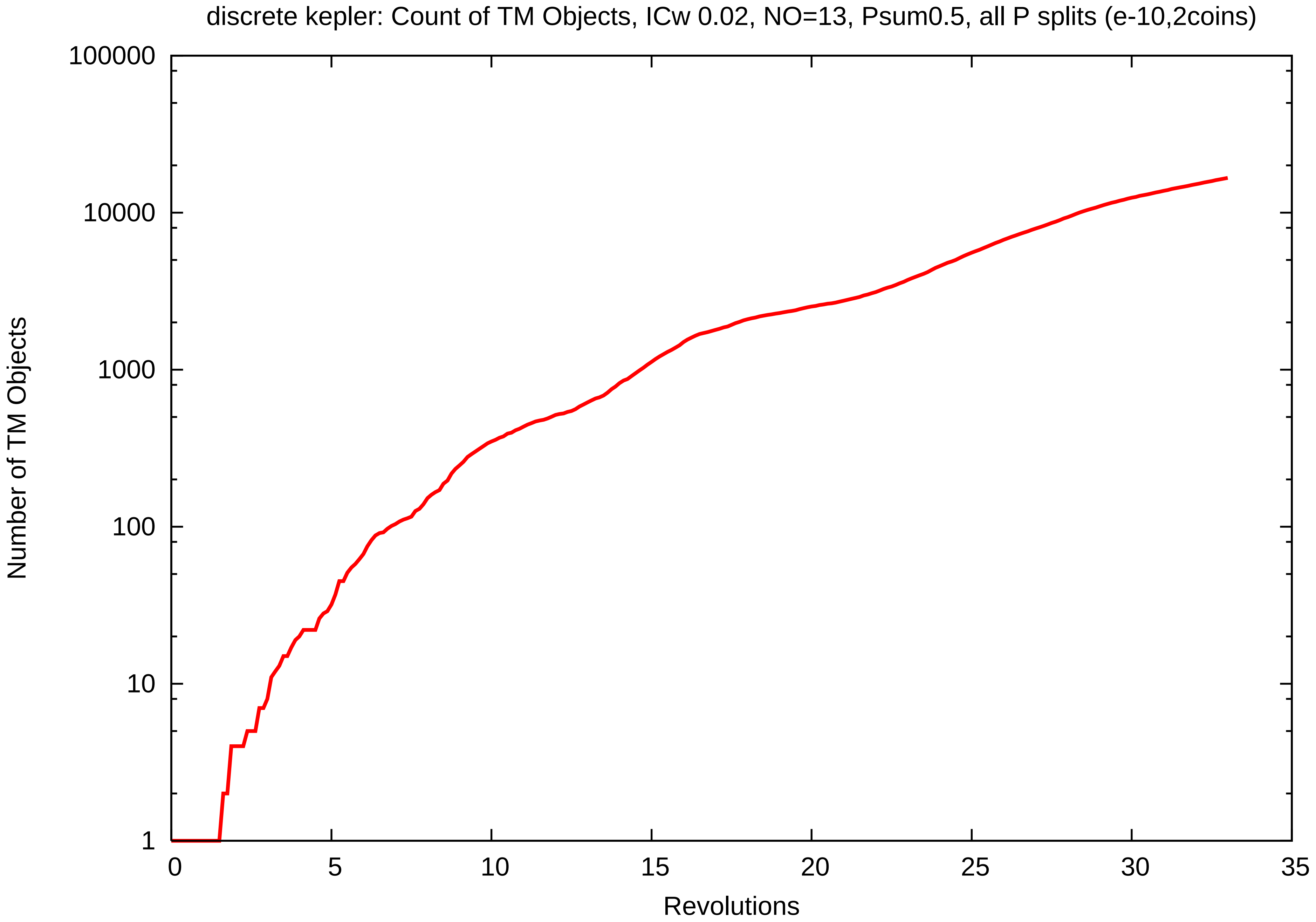
discrete kepler. NO=13 w7
4.75 rev
3.75 rev
2.75 rev
1.75 rev
0.75 rev
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5


discrete kepler. NO=13 w7
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
2.75 rev
1.75 rev
0.75 rev



discrete kepler. 33rd revolution, ICw 0.02, NO=13 w7


discrete kepler: Count of TM Objects, ICw 0.02, NO=13, Psum0.5, all P splits (e-10,2coins)
Number of TM Objects
Revolutions
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
0
5
10
15
20
25
30
35

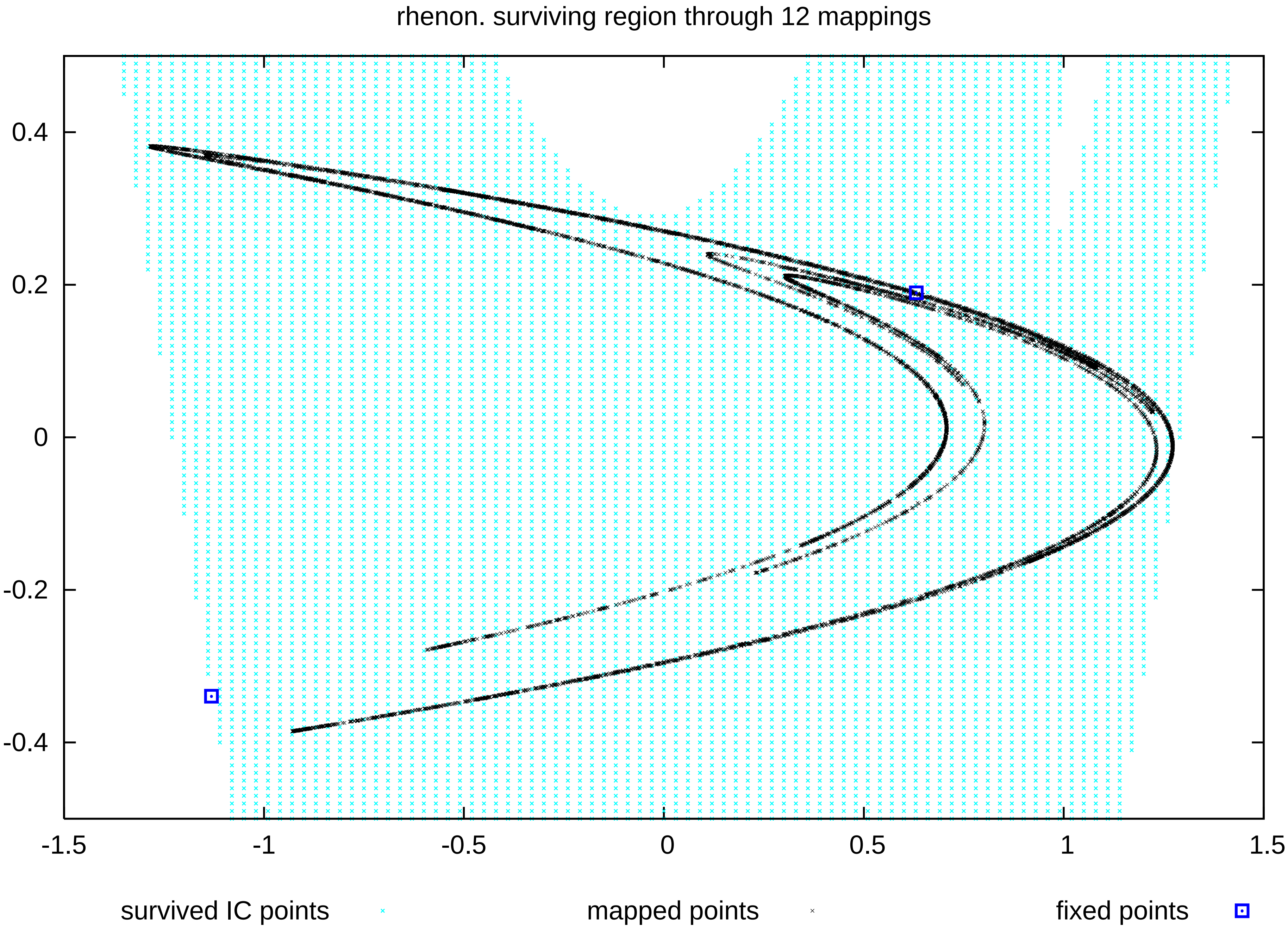

discrete kepler: Count of TM Objects, ICw 0.02, NO=13, Psum0.5, all P splits (e-10,2coins)
Number of TM Objects
Revolutions
100000
10000
1000
100
10
1
0
5
10
15
20
25
30
35

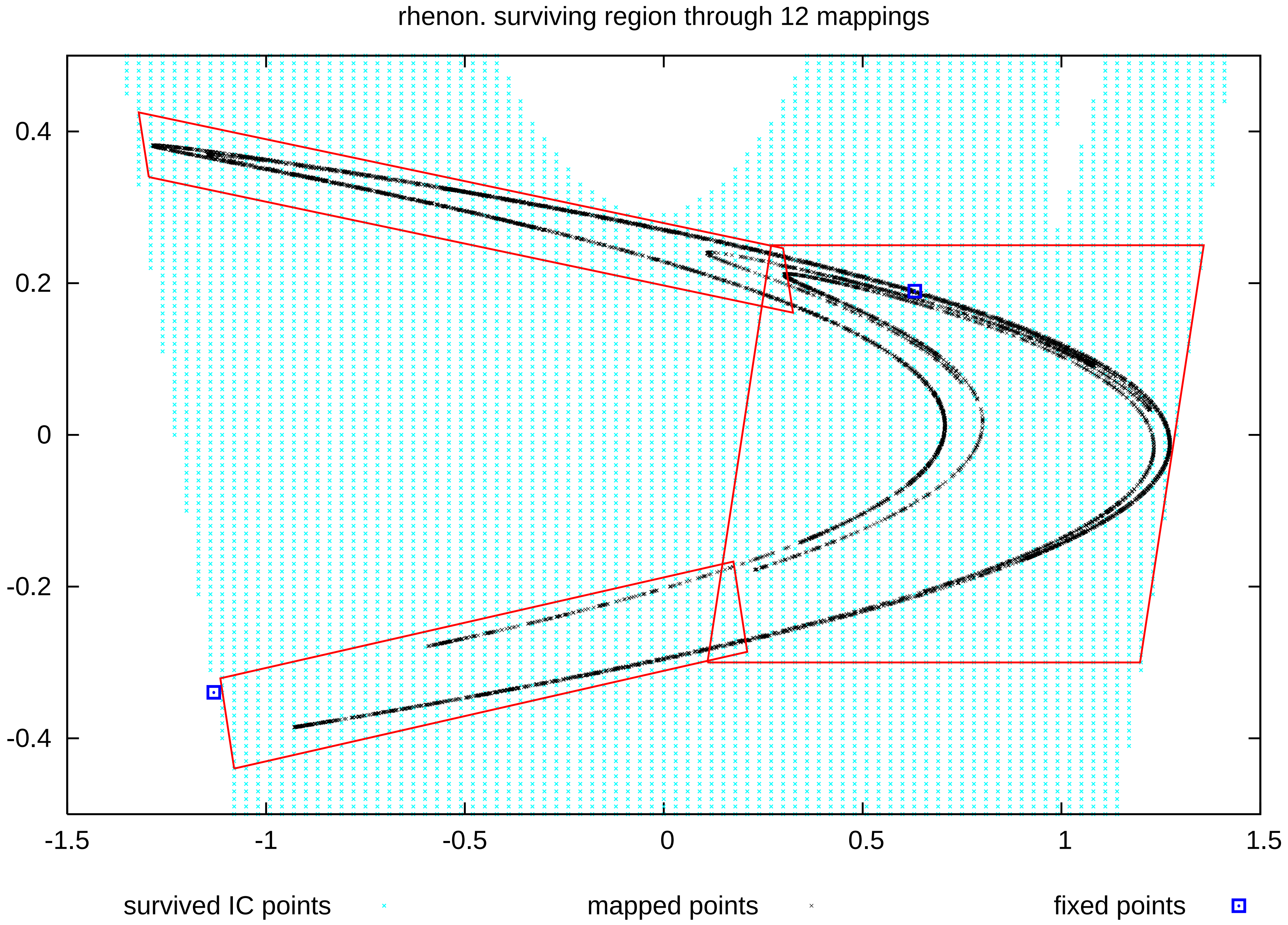
The Henon Map

$$H(x, y) = (1 - ax^2 + y, bx).$$

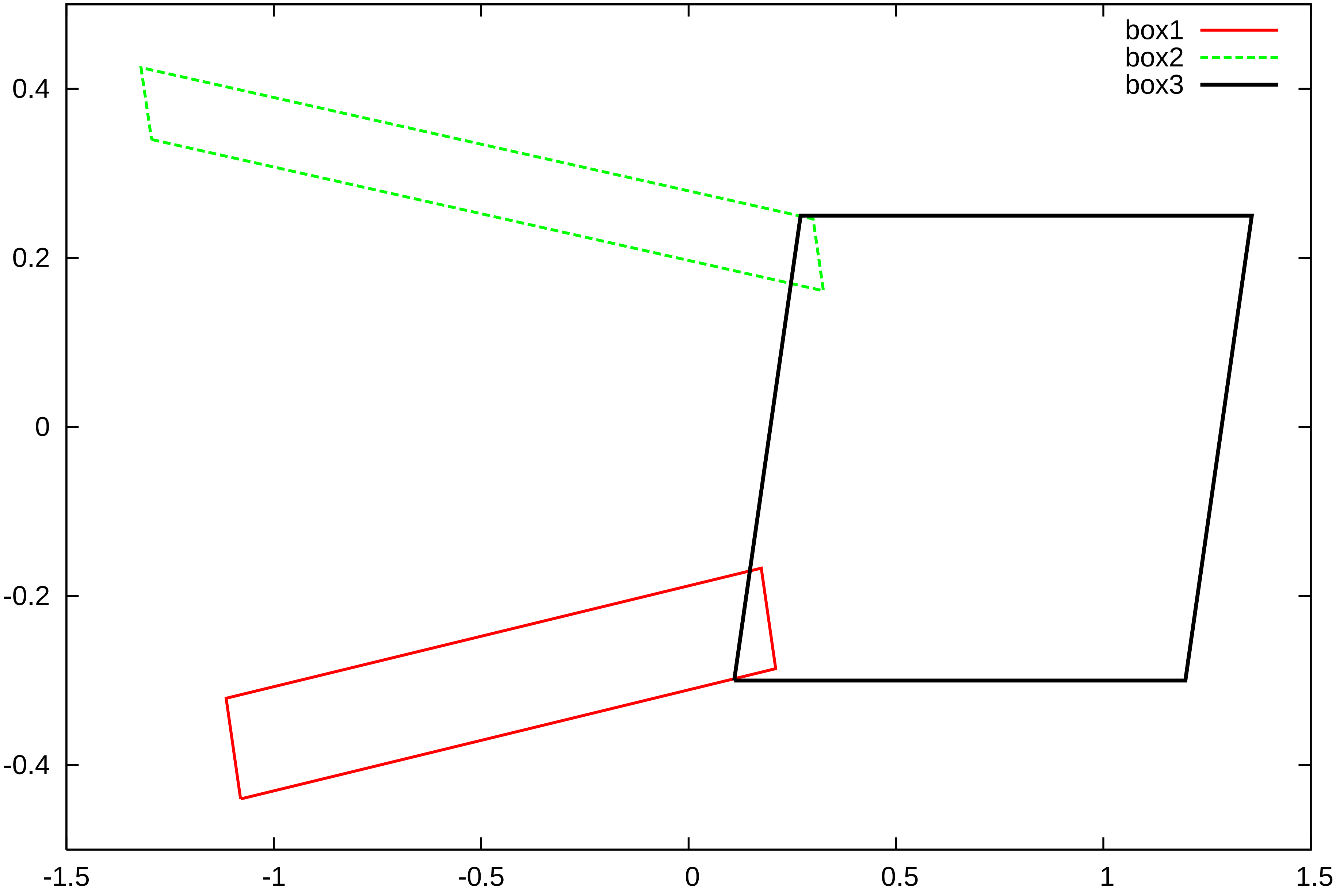
We set the parameters $a = 1.4$ and $b = 0.3$, which are originally considered by Henon. The map H has two fixed points.

$$\vec{p}_1 = (0.63135, 0.18940) \quad \text{and} \quad \vec{p}_2 = (-1.13135, -0.33941).$$

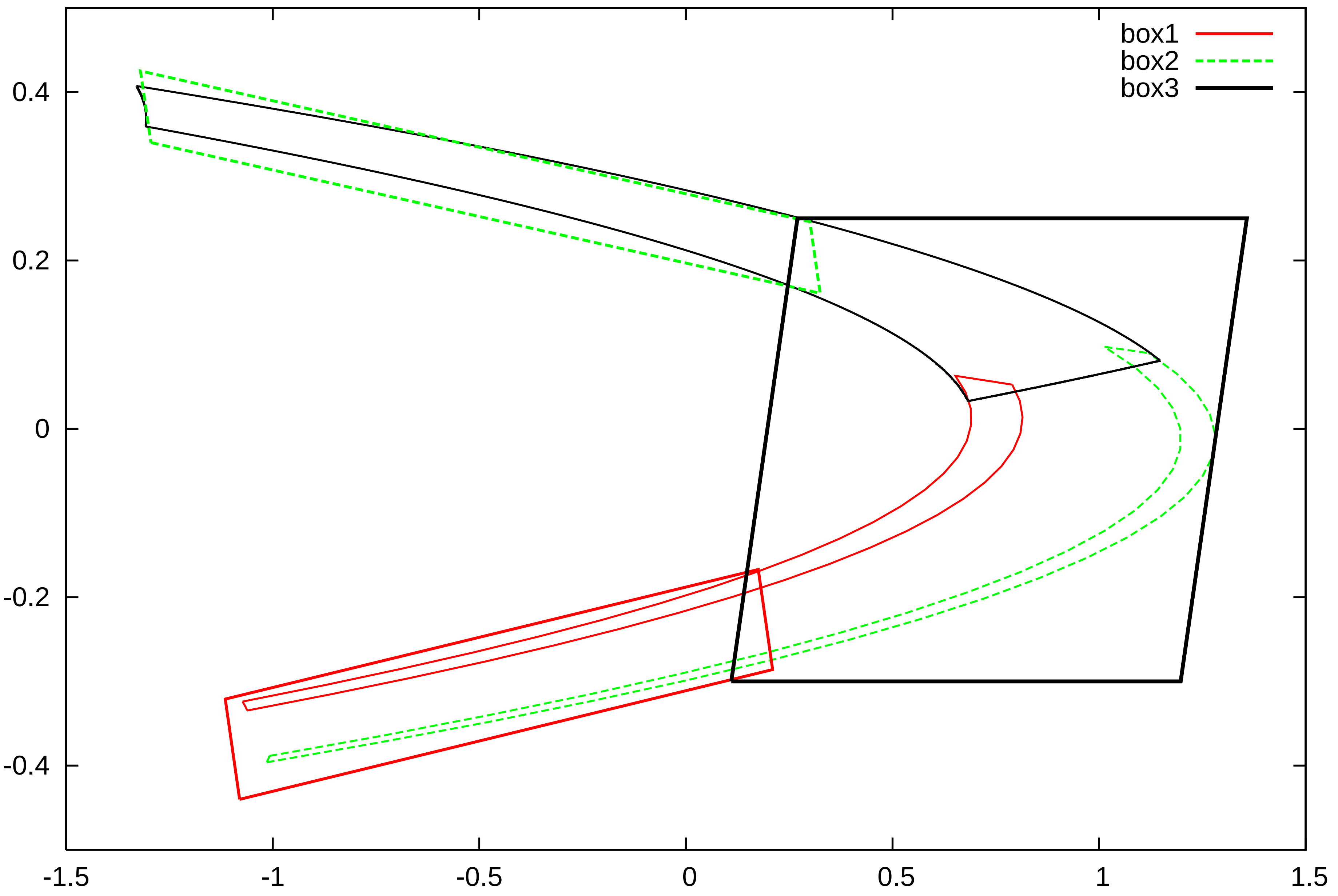

rhenon. surviving region through 12 mappings
0.4
0.2
0
-0.2
-0.4
-1.5
-1
-0.5
0
0.5
1
1.5
survived IC points
mapped points
fixed points


rhenon. surviving region through 12 mappings
0.4
0.2
0
-0.2
-0.4
-1.5
-1
-0.5
0
0.5
1
1.5
survived IC points
mapped points
fixed points

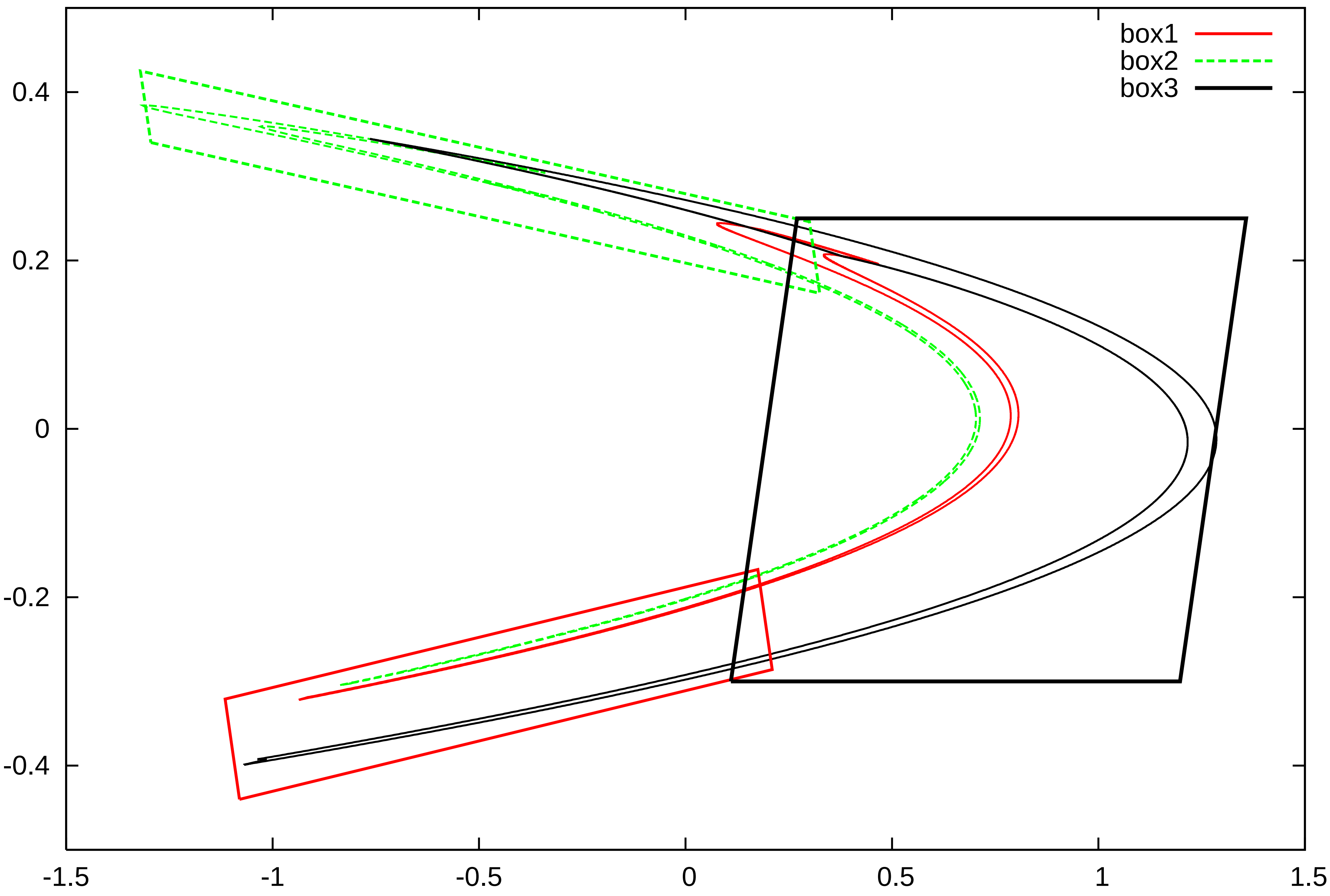
rhenon. IC boxes 3/3/08

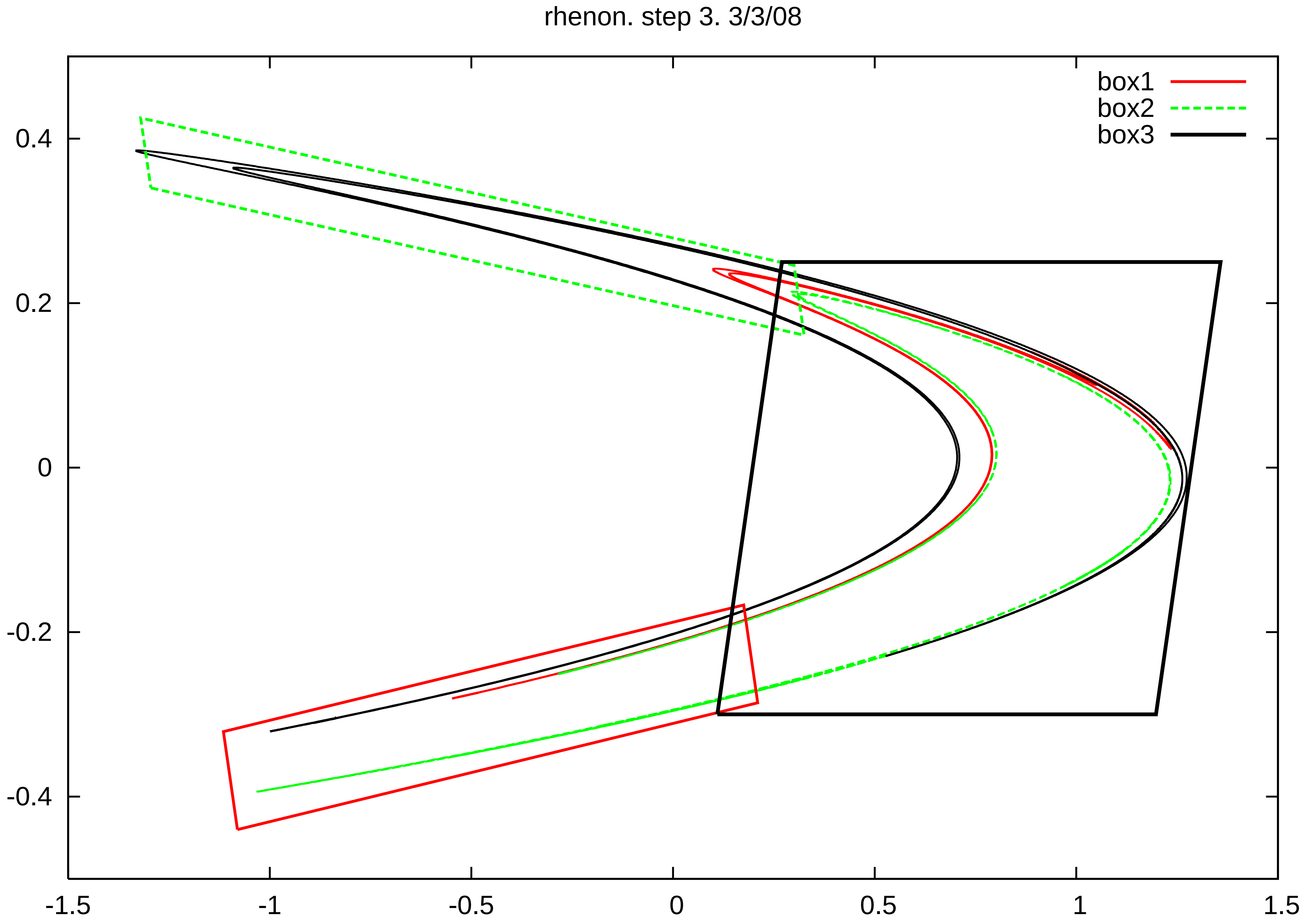


rhenon. step 1. 3/3/08

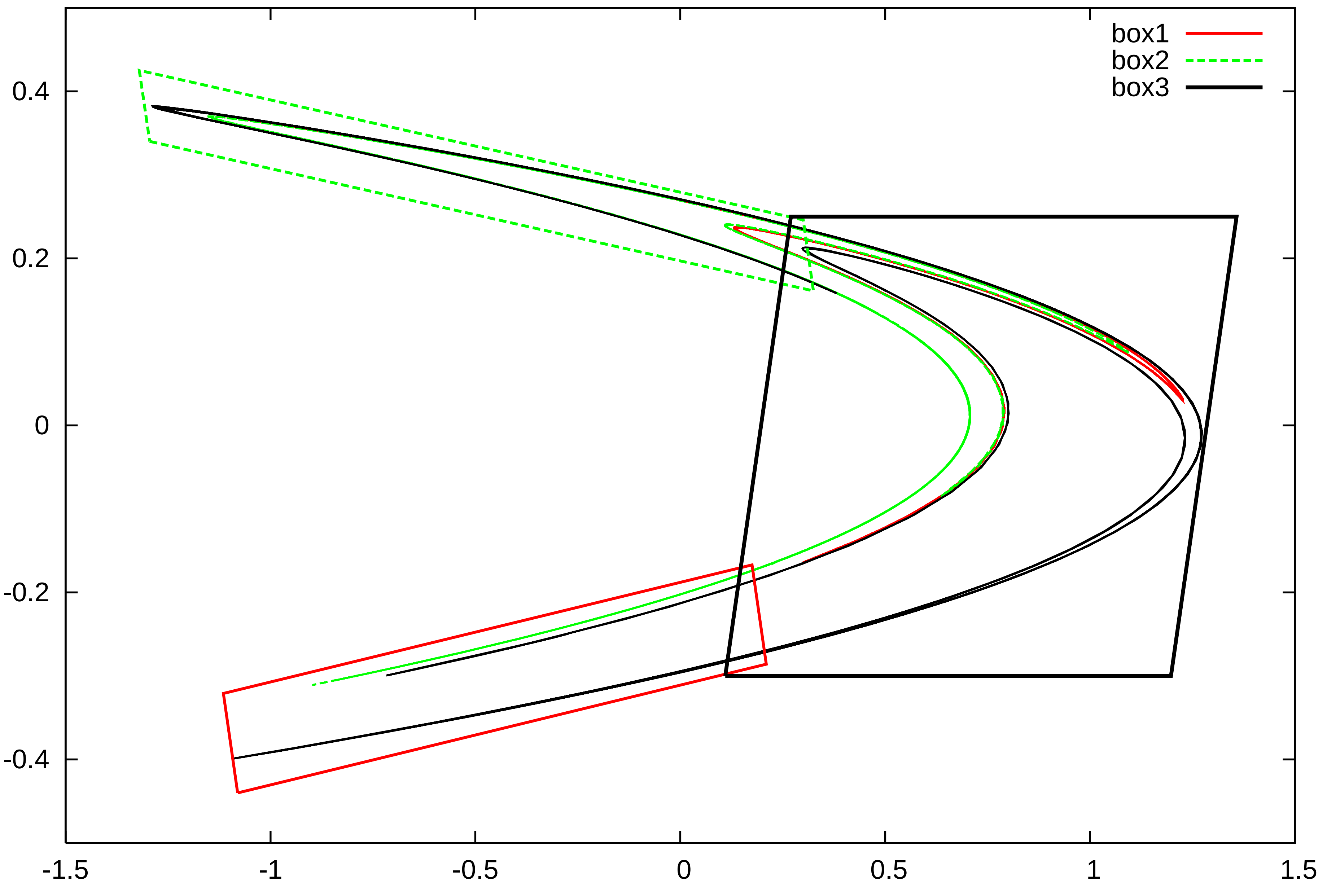


rhenon. step 2. 3/3/08

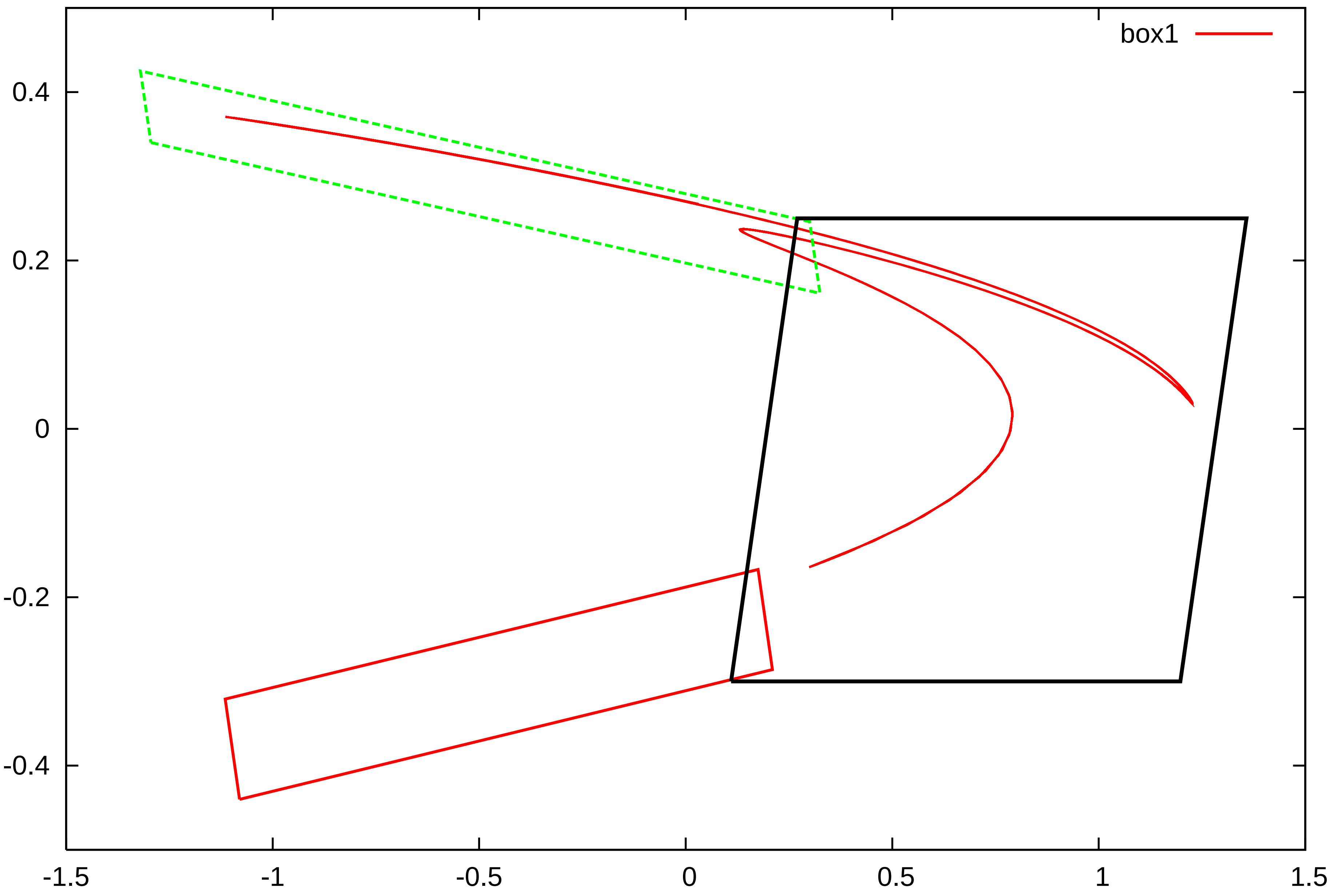


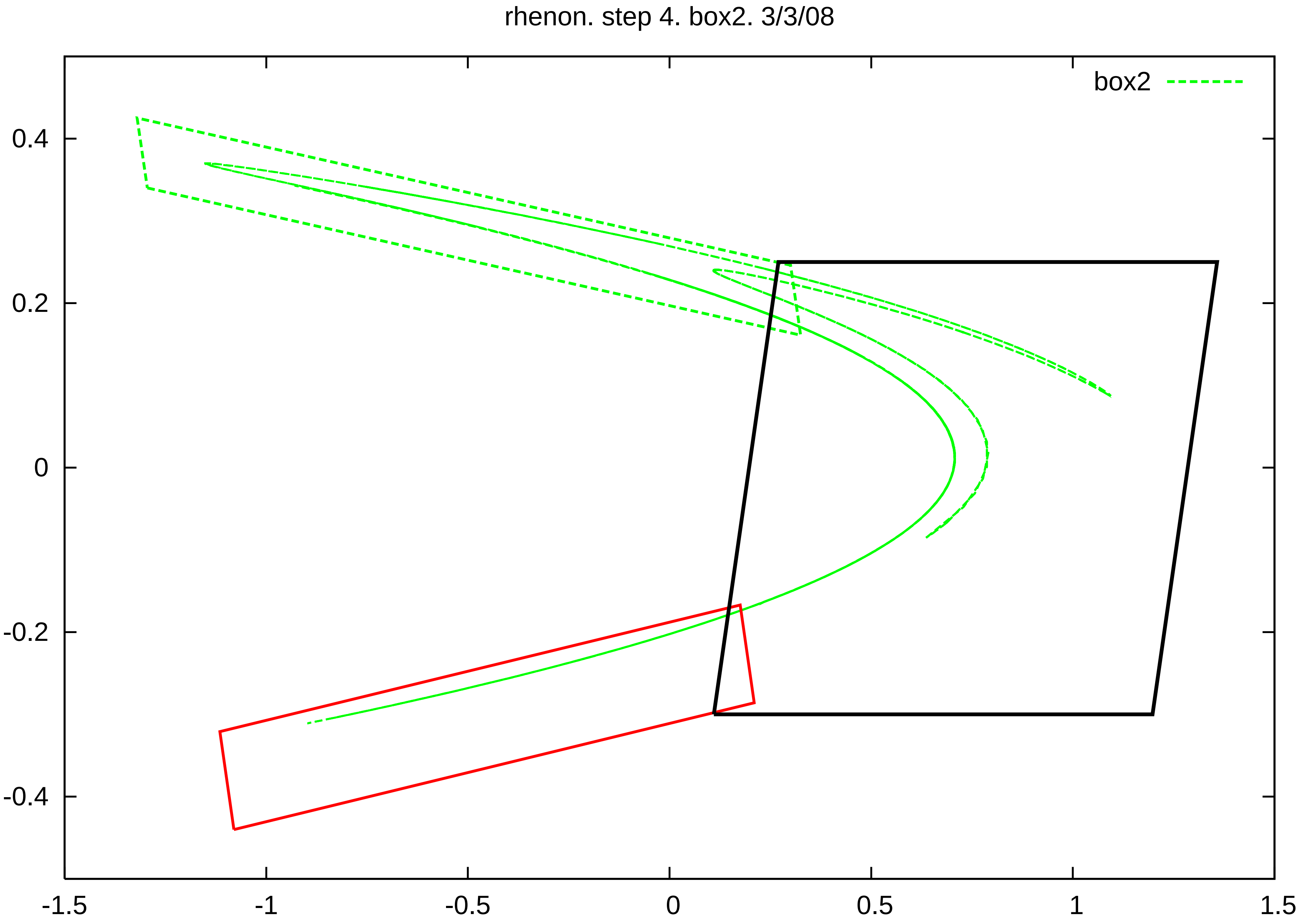

rhenon. step 3. 3/3/08
box1
box2
box3
0.4
0.2
0
-0.2
-0.4
-1.5
-1
-0.5
0
0.5
1
1.5

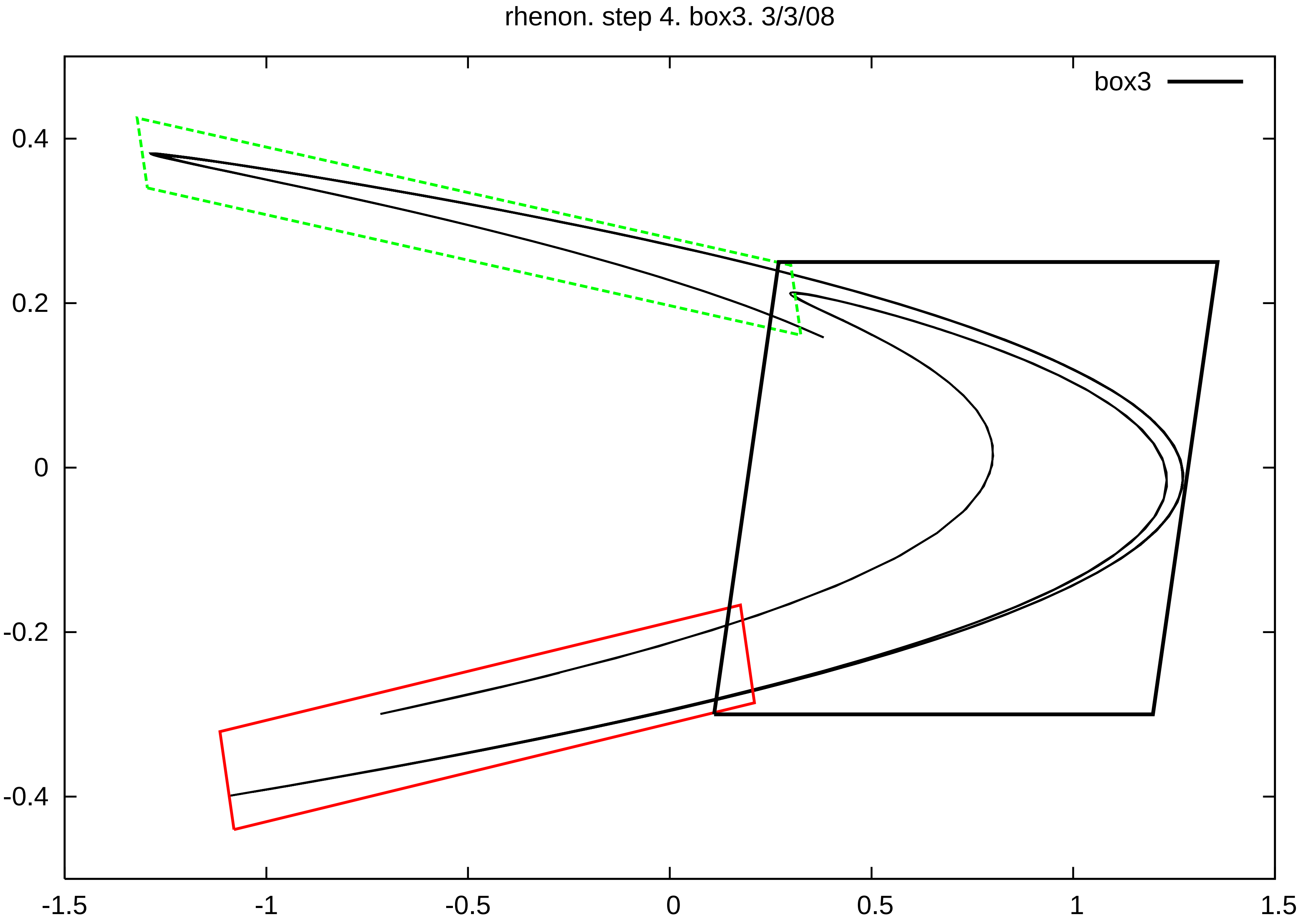
rhenon. step 4. 3/3/08



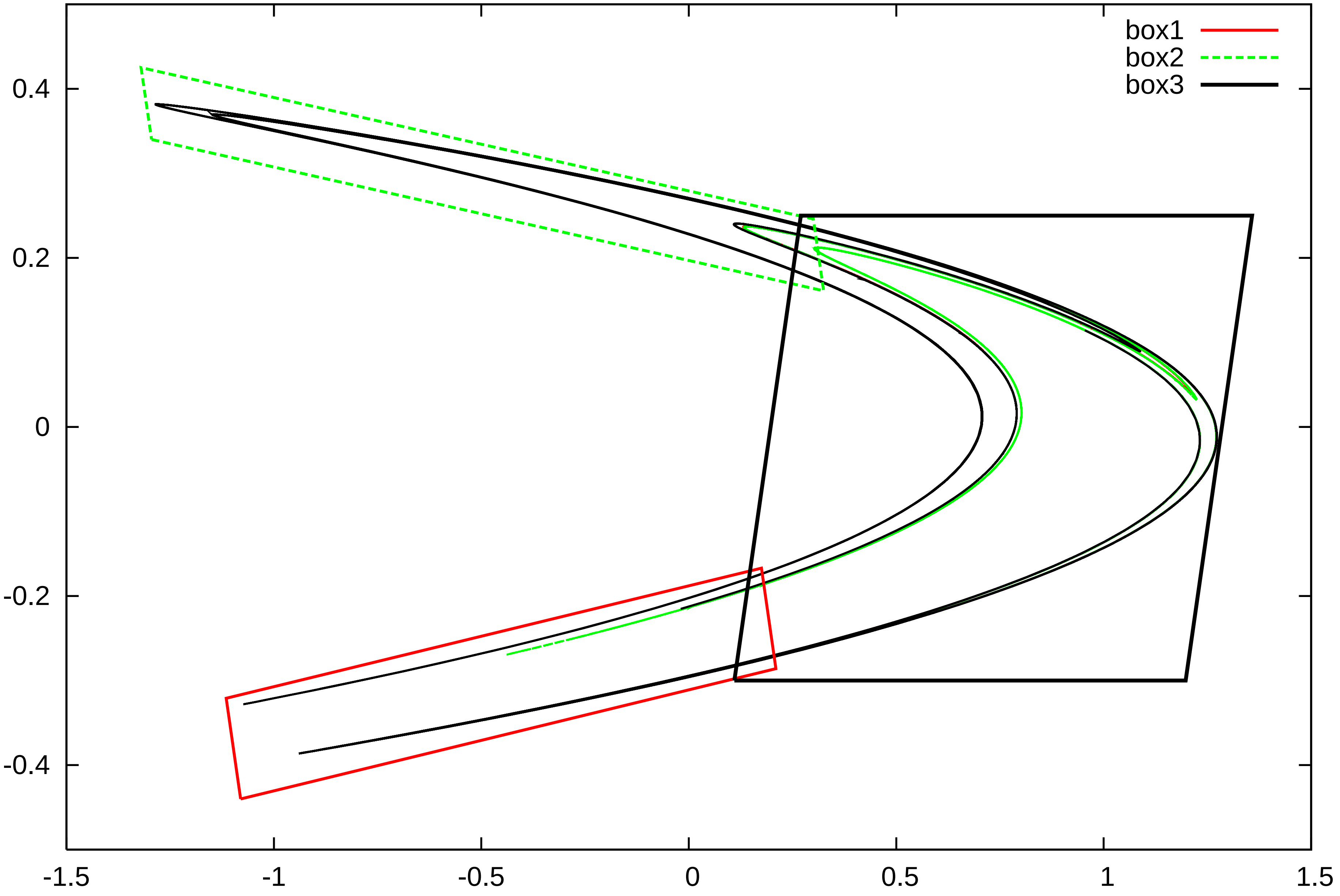
rhenon. step 4. box1. 3/3/08

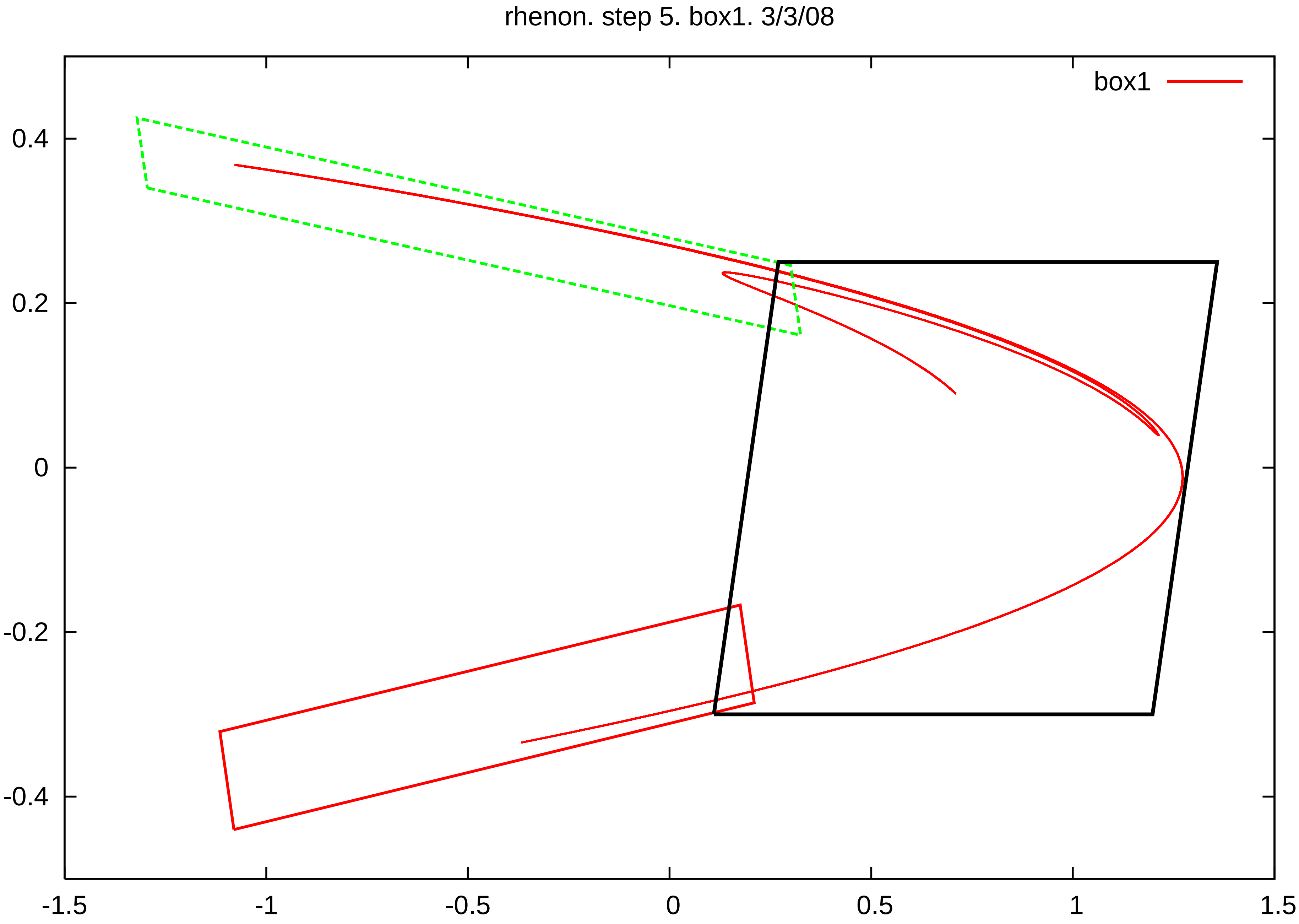


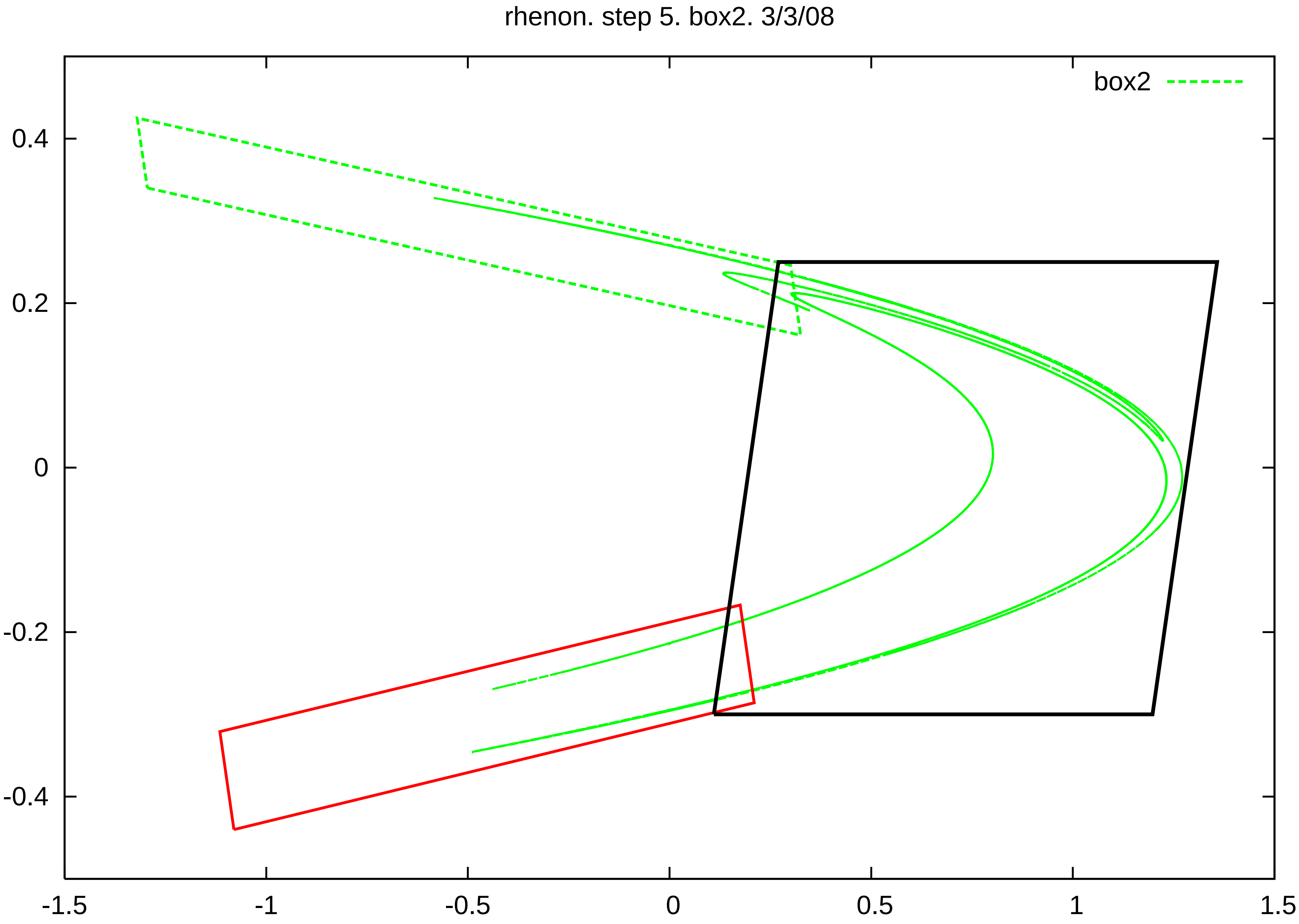

rhenon. step 4. box2. 3/3/08
box2
0.4
0.2
0
-0.2
-0.4
-1.5
-1
-0.5
0
0.5
1
1.5


rhenon. step 4. box3. 3/3/08
box3
0.4
0.2
0
-0.2
-0.4
-1.5
-1
-0.5
0
0.5
1
1.5

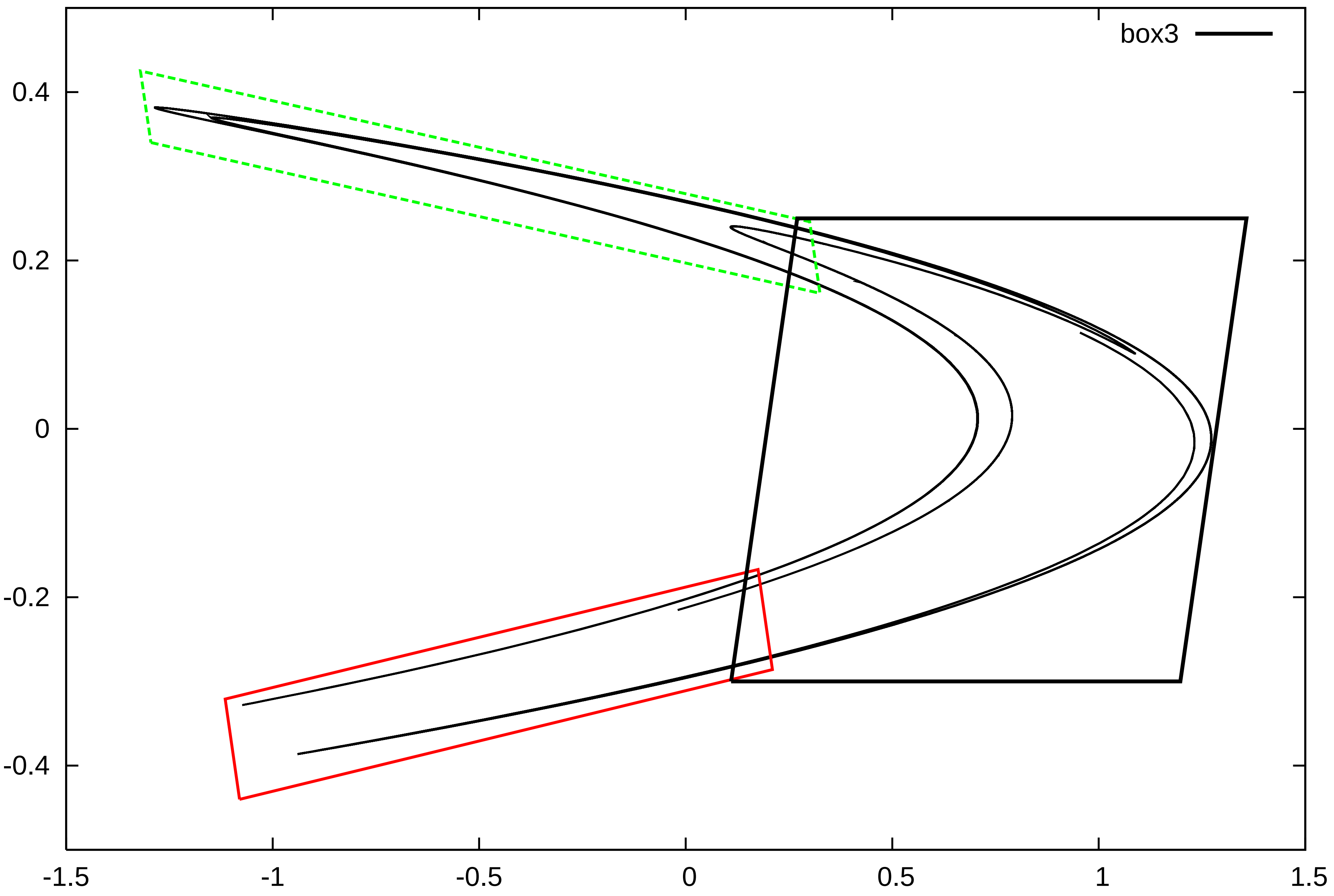
rhenon. step 5. 3/3/08




rhenon. step 5. box1. 3/3/08
box1
0.4
0.2
0
-0.2
-0.4
-1.5
-1
-0.5
0
0.5
1
1.5


rhenon. step 5. box2. 3/3/08
box2
0.4
0.2
0
-0.2
-0.4
-1.5
-1
-0.5
0
0.5
1
1.5

rhenon. step 5. box3. 3/3/08



rhenon: Number of Objects

To carry out multiple mappings of the Henon map, Taylor model objects underwent the domain decomposition.

Number of Taylor model objects used for multiple mappings:

	n	w	for 5 steps	for 7 steps
box1	33	17	3	1386
box2	21	11	148	1691
box3	33	17	8	2839

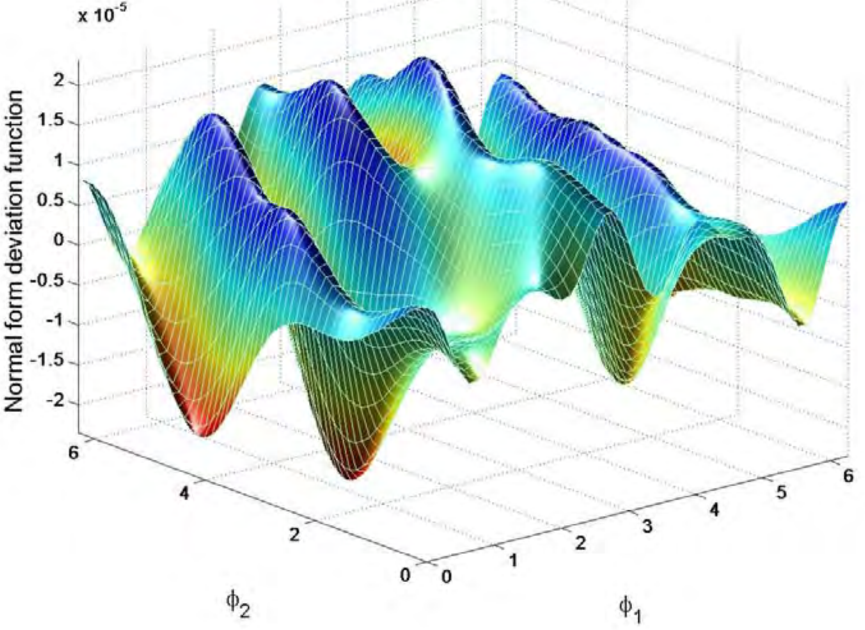
Normal Form Methods

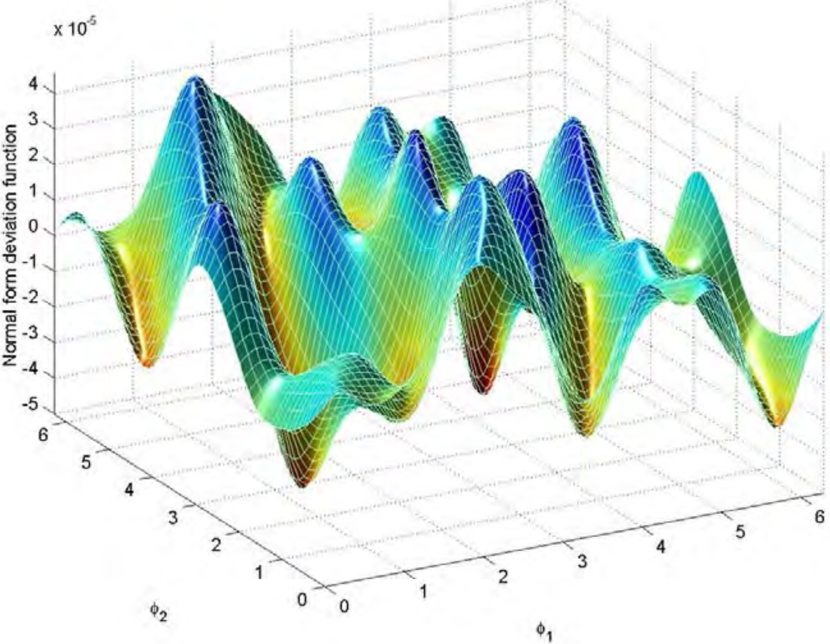
Iterative order-by-order coordinate transfomation to simplify dynamics around a fixed point. Assume we have TM representation of

1. Discrete Systems: One rigorous iteration of nonlinear map
2. Continuous Systems: Rigorous Flow representation of suitable time step Δt

Result: Except for resonances, obtain a coordinate transformation that up to order n **linearizes** the motion

- 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, i.e. $\lambda_i > 1$ real for $i = 1, ..., k$, $\lambda_i < 1$ for $i = 1, ..., v$: motion along hyperpolae, $\vec{e}_i$ axis expanded or contracted by λ_i


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

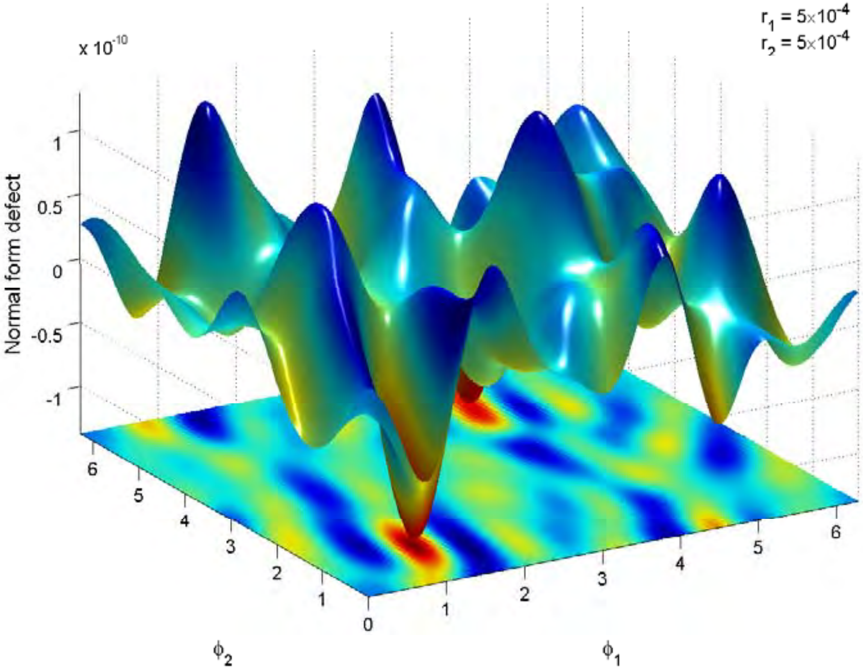


Fig. 9. Projection of the normal form defect function. Dependence on two angle variables for the fixed radii $r_1 = r_2 = 5 \cdot 10^{-4}$

Region	Boxes studied	CPU-time	Bound	Transversal Iterations
$[0.2, 0.4] \cdot 10^{-4}$	$82,930$	$30,603$ sec	$0.859 \cdot 10^{-13}$	$2.3283 \cdot 10^8$
$[0.4, 0.6] \cdot 10^{-4}$	$82,626$	$30,603$ sec	$0.587 \cdot 10^{-12}$	$3.4072 \cdot 10^7$
$[0.6, 0.9] \cdot 10^{-4}$	$64,131$	$14,441$ sec	$0.616 \cdot 10^{-11}$	$4.8701 \cdot 10^6$
$[0.9, 1.2] \cdot 10^{-4}$	$73,701$	$13,501$ sec	$0.372 \cdot 10^{-10}$	$8.0645 \cdot 10^5$
$[1.2, 1.5] \cdot 10^{-4}$	$106,929$	$24,304$ sec	$0.144 \cdot 10^{-9}$	$2.0833 \cdot 10^5$
$[1.5, 1.8] \cdot 10^{-4}$	$111,391$	$26,103$ 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.

Normal Form Methods

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

Result: Except for resonances, up to order n,

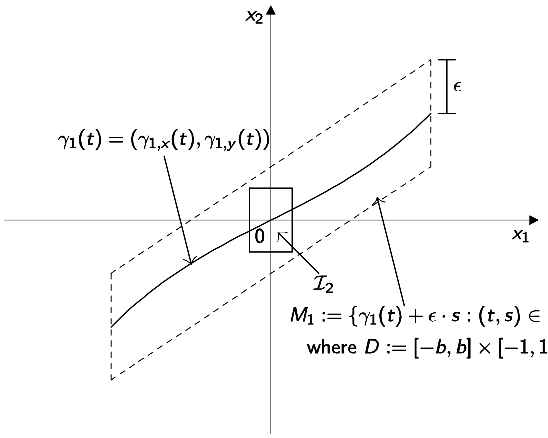
- Elliptic case $\lambda_{i+1} = \bar{\lambda}_i$: spiral motion in $(\lambda_i, \lambda_{i+1})$ plane
- Elliptic unity case $\lambda_{i+1} = \bar{\lambda}_i$ and $|\bar{\lambda}_i| = 1$: circular motion, radius-dependent rotation frequency
- Hyperbolic case (λ_i real) motion along $\vec{e}_i$ axis,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

Rigorous Unstable Manifold Enclosures I

Goal: Find collection of hopefully very narrow Taylor models that contain a hopefully long stretch of unstable manifold.


x_2
ϵ
$\gamma_1(t) = (\gamma_{1,x}(t), \gamma_{1,y}(t))$
0
x_1
$\mathcal{I}_2$
$M_1 := \{\gamma_1(t) + \epsilon \cdot s : (t, s) \in D\}$
where $D := [-b, b] \times [-1, 1]$

Begin with unstable manifold near fixed point:

- Obtain approximate polynomial path $\gamma(t)$ as image of normal form $\vec{e}_1$ axis
- Put "test tube" around $\gamma(t)$ to get $\gamma(t) + \varepsilon \cdot s \cdot \vec{e}_2$. Practical choice: $\varepsilon = 10^{-14}$

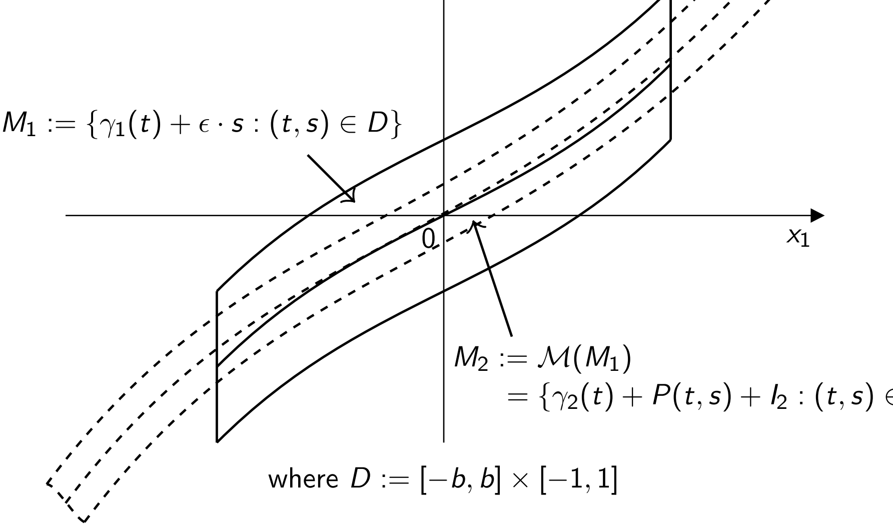
Rigorous Unstable Manifold Enclosures II

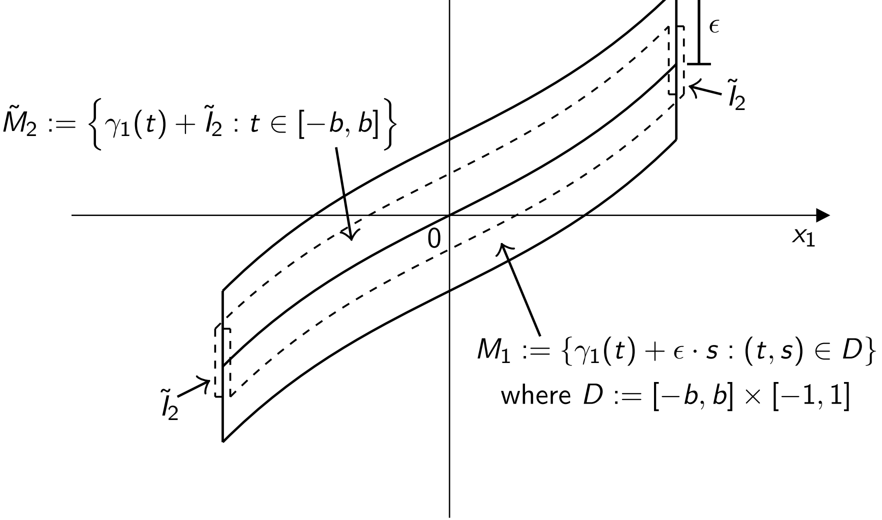
- Verify that $M(\gamma(t) + \varepsilon \cdot s \cdot \vec{e}_2)$ leaves "test tube" only at ends. Very useful for that:

  1. $M(\gamma(t)) =_n \gamma(\lambda_1 \cdot t)$, so orbit of γ is reproduced to order n
  2. M is contracting with λ_2 perpendicular to γ
  3. $\gamma(t)+\varepsilon \cdot s \cdot \vec{e}_2$ and its image under M can be treated rigorously in Taylor model arithmetic

After these steps, it is assured that

- The unstable manifold does NOT leave $\gamma(t) + \varepsilon \cdot s \cdot \vec{e}_2$ at top or bottom
- The unstable manifold DOES leave $\gamma(t) + \varepsilon \cdot s \cdot \vec{e}_2$ at the sides (easy to show)


$M_1 := \{\gamma_1(t) + \epsilon \cdot s : (t,s) \in D\}$
0
x_1
$M_2 := \mathcal{M}(M_1)$
$= \{\gamma_2(t) + P(t,s) + l_2 : (t,s) \in$
where $D := [-b,b] \times [-1,1]$


ϵ
$\tilde{I}_2$
$\tilde{M}_2 := \left\{ \gamma_1(t) + \tilde{I}_2 : t \in [-b, b] \right\}$
0
x_1
$M_1 := \{ \gamma_1(t) + \epsilon \cdot s : (t, s) \in D \}$
where $D := [-b, b] \times [-1, 1]$
$\tilde{I}_2$

Rigorous Unstable Manifold Enclosures III

Unstable manifold can be drawn as far as desired by

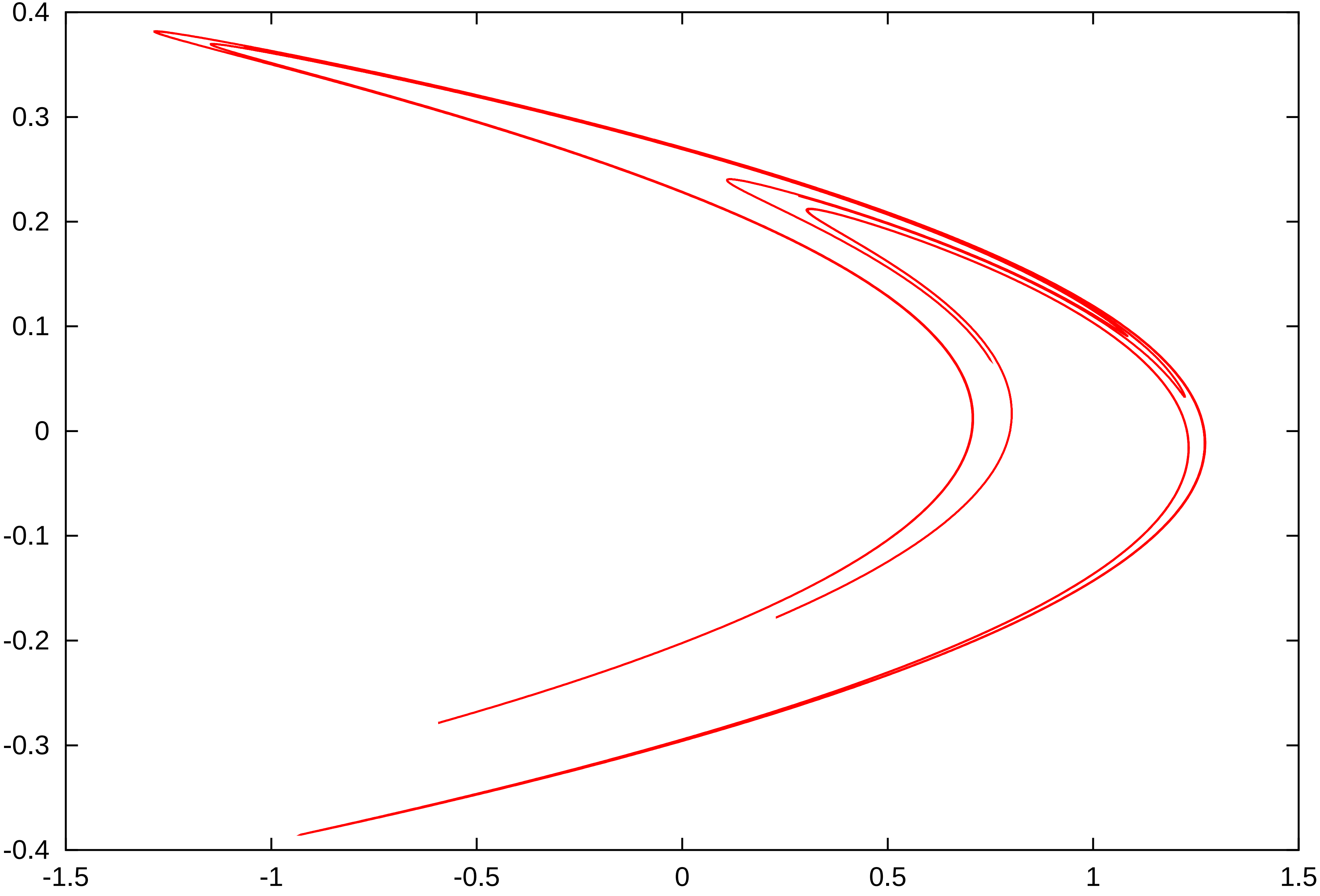
- Mapping $\gamma(t) + \varepsilon \cdot s \cdot \vec{e}_2$ through M repeatedly
- Splitting result if length > tolerance

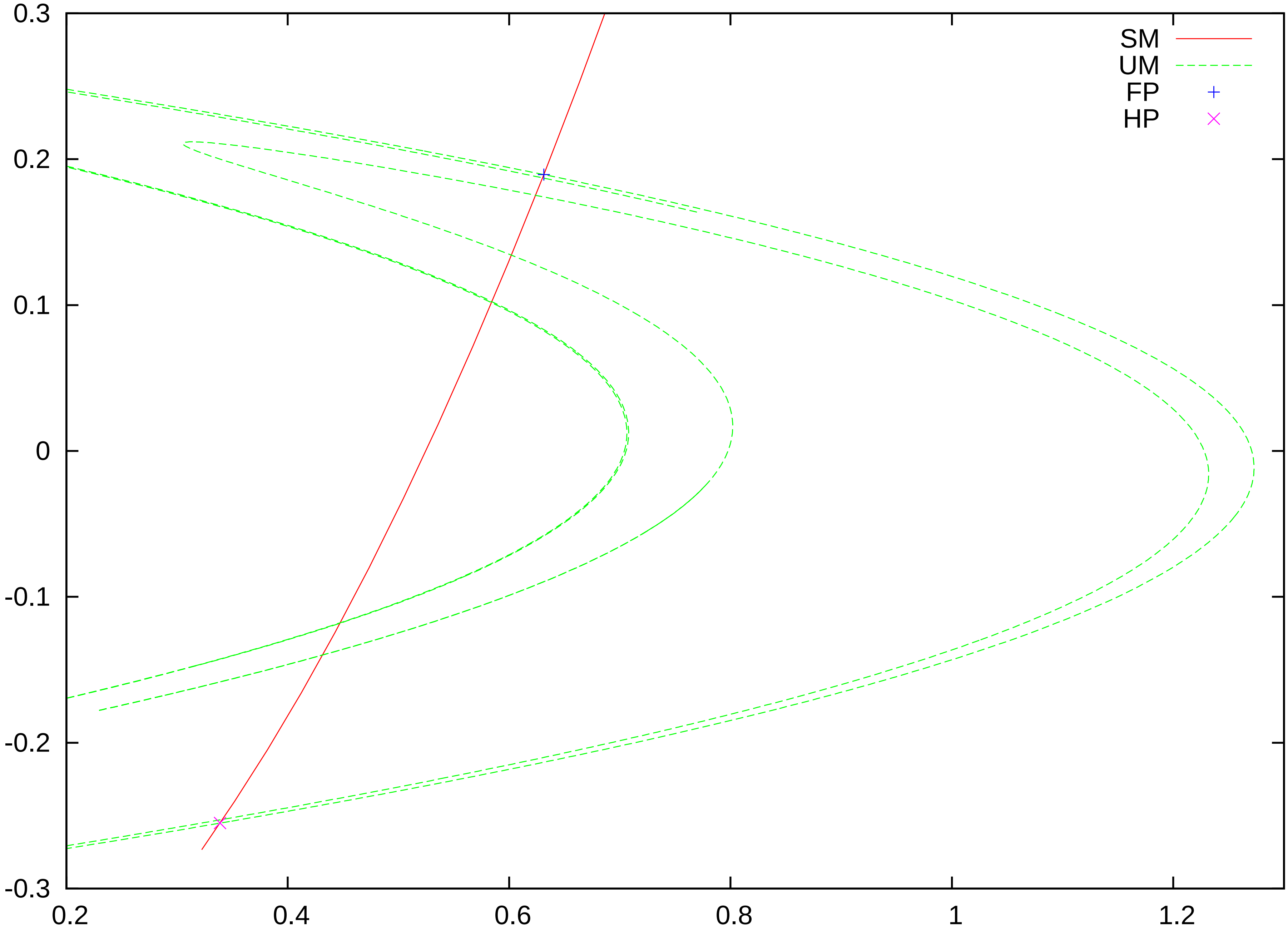
As a result, we obtain a collection of as many Taylor model as we wish, each of which

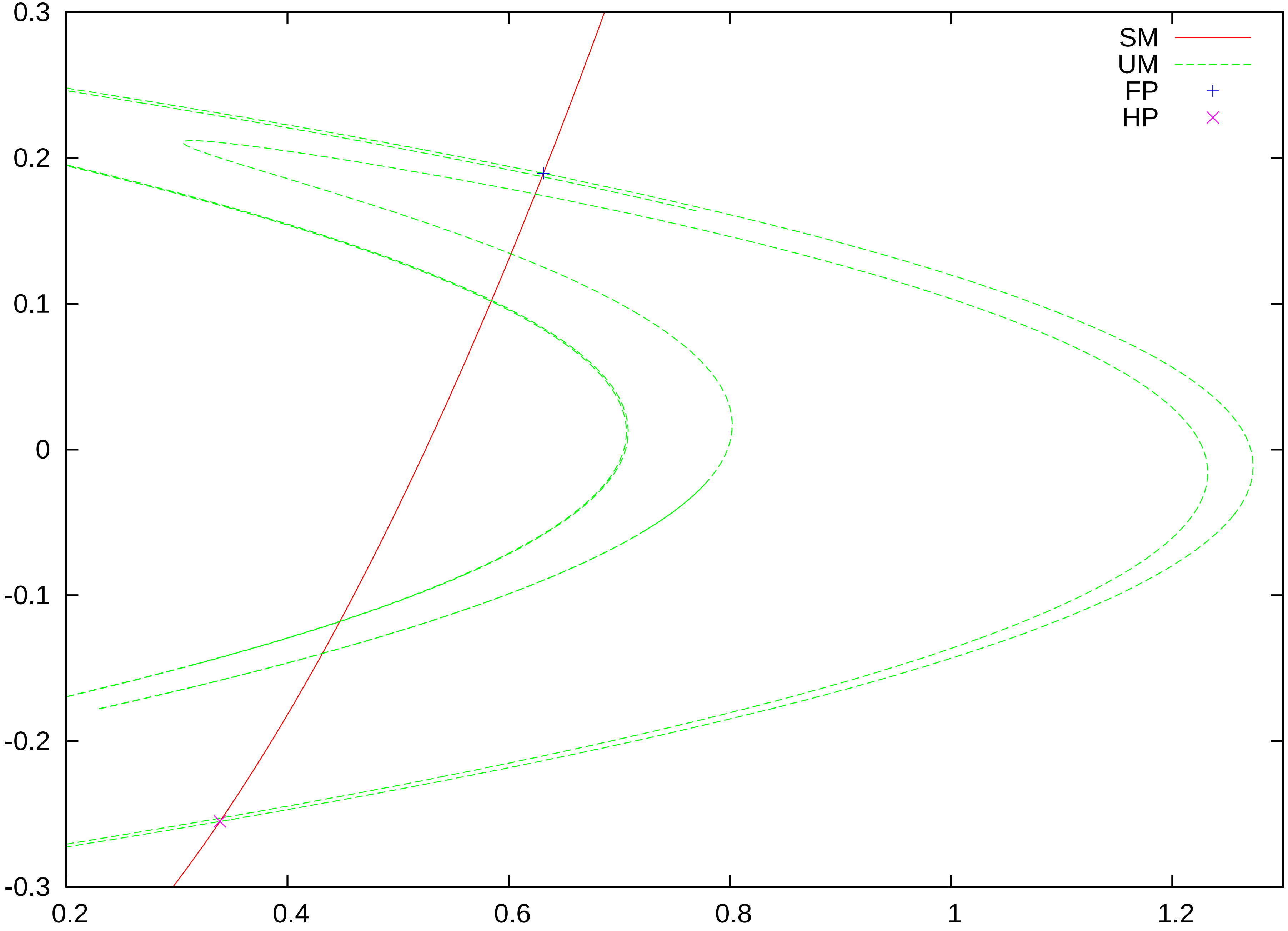
- Contains a piece of the unstable manifold
- The unstable manifold leaves through the "narrow sides"
- The unstable manifold does not leave through the "long sides"

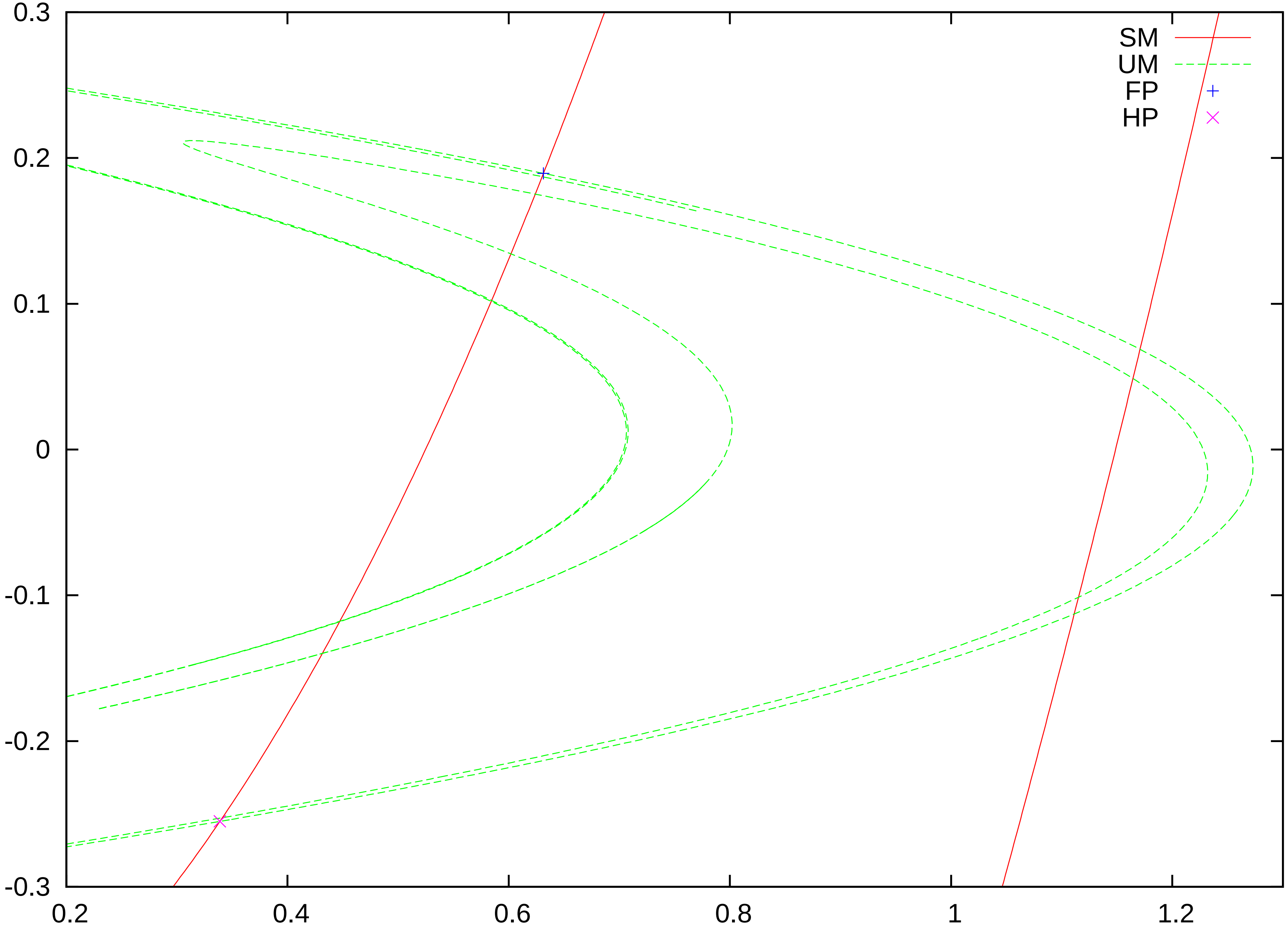
By considering the inverse map, we can analogously obtain rigorous enclosures of the stable manifolds.

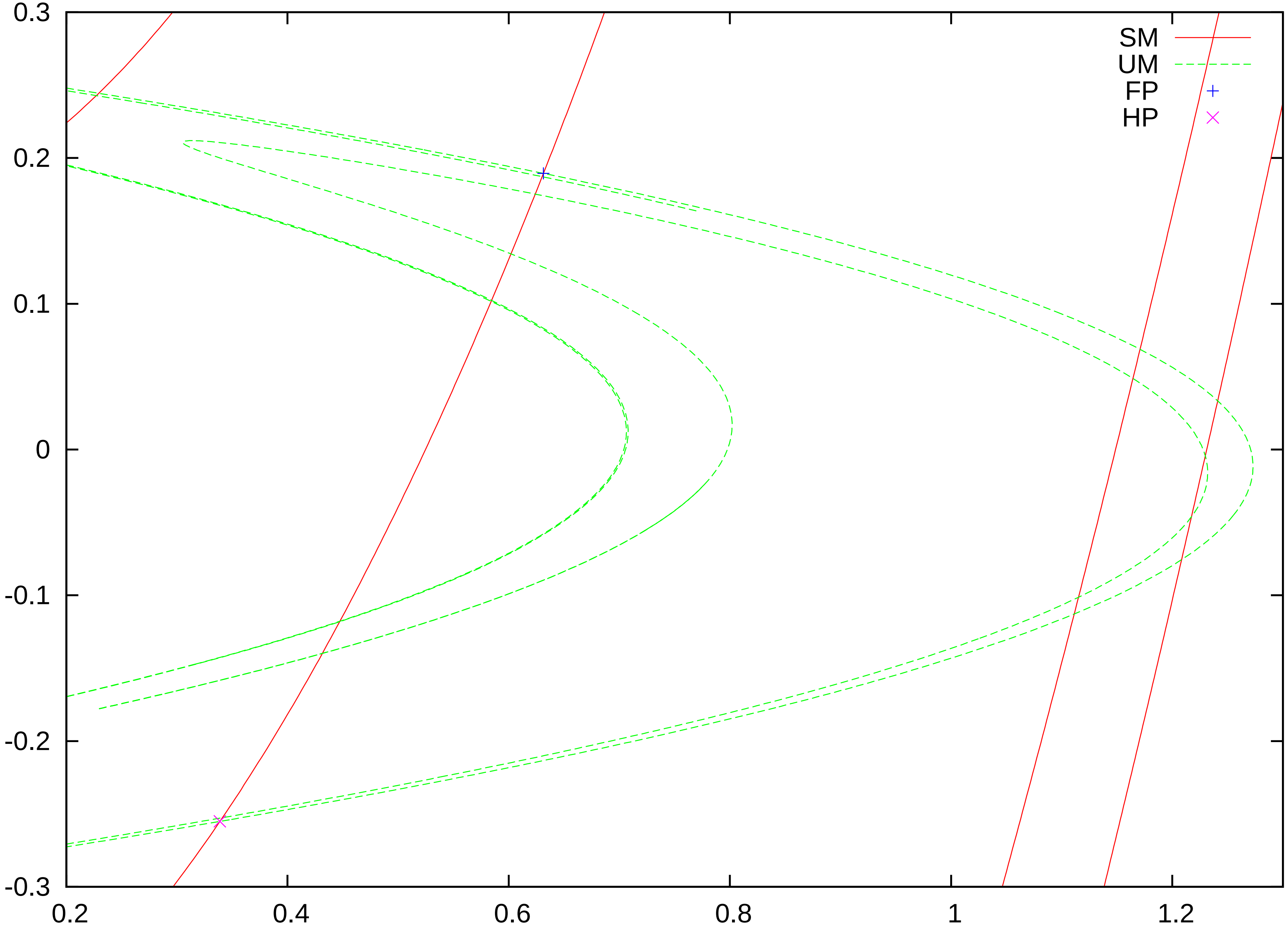
Unstable Manifold of a Henon map (a=1.4, b=0.3) represented by 450 pieces of TMs

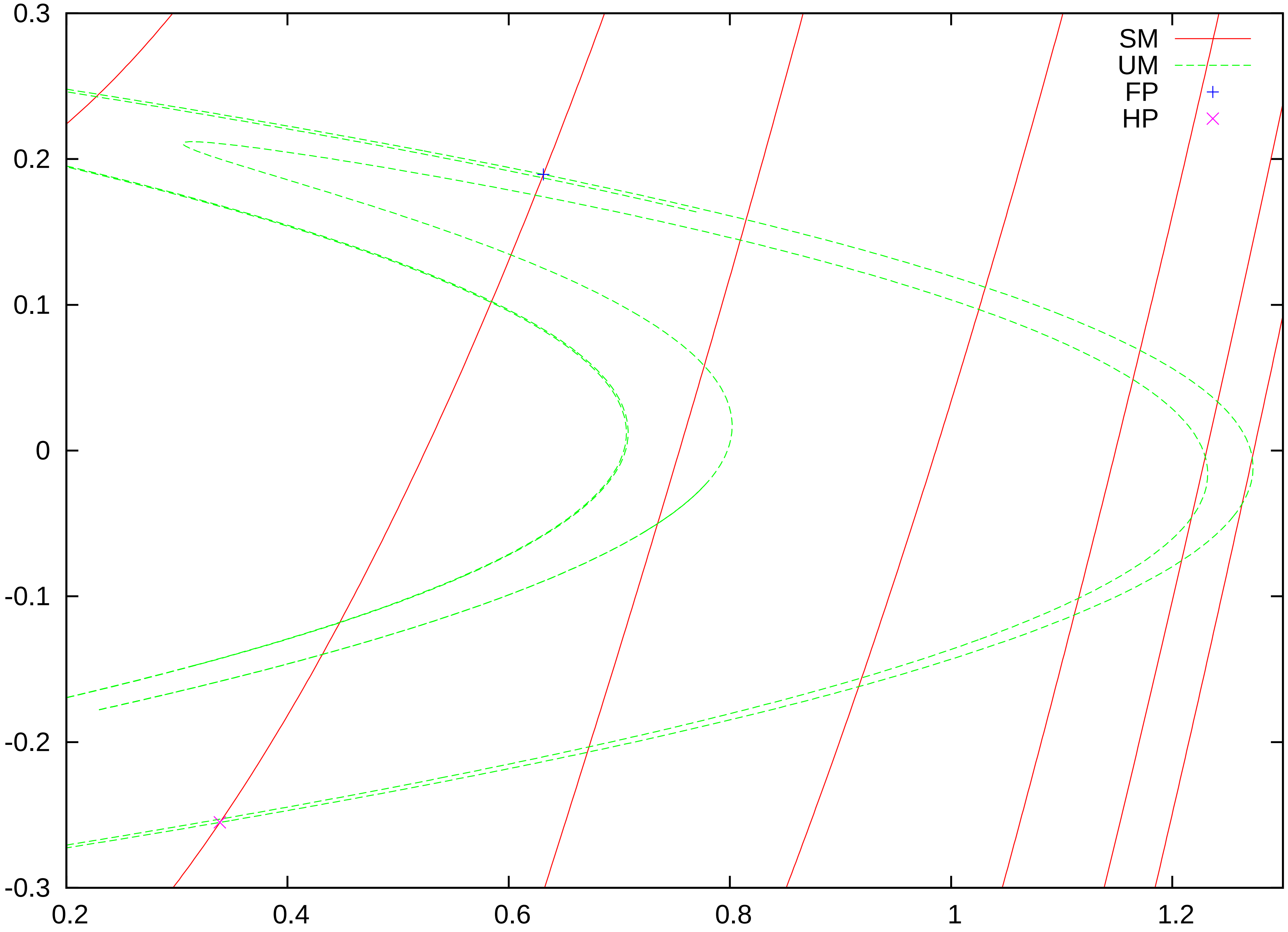


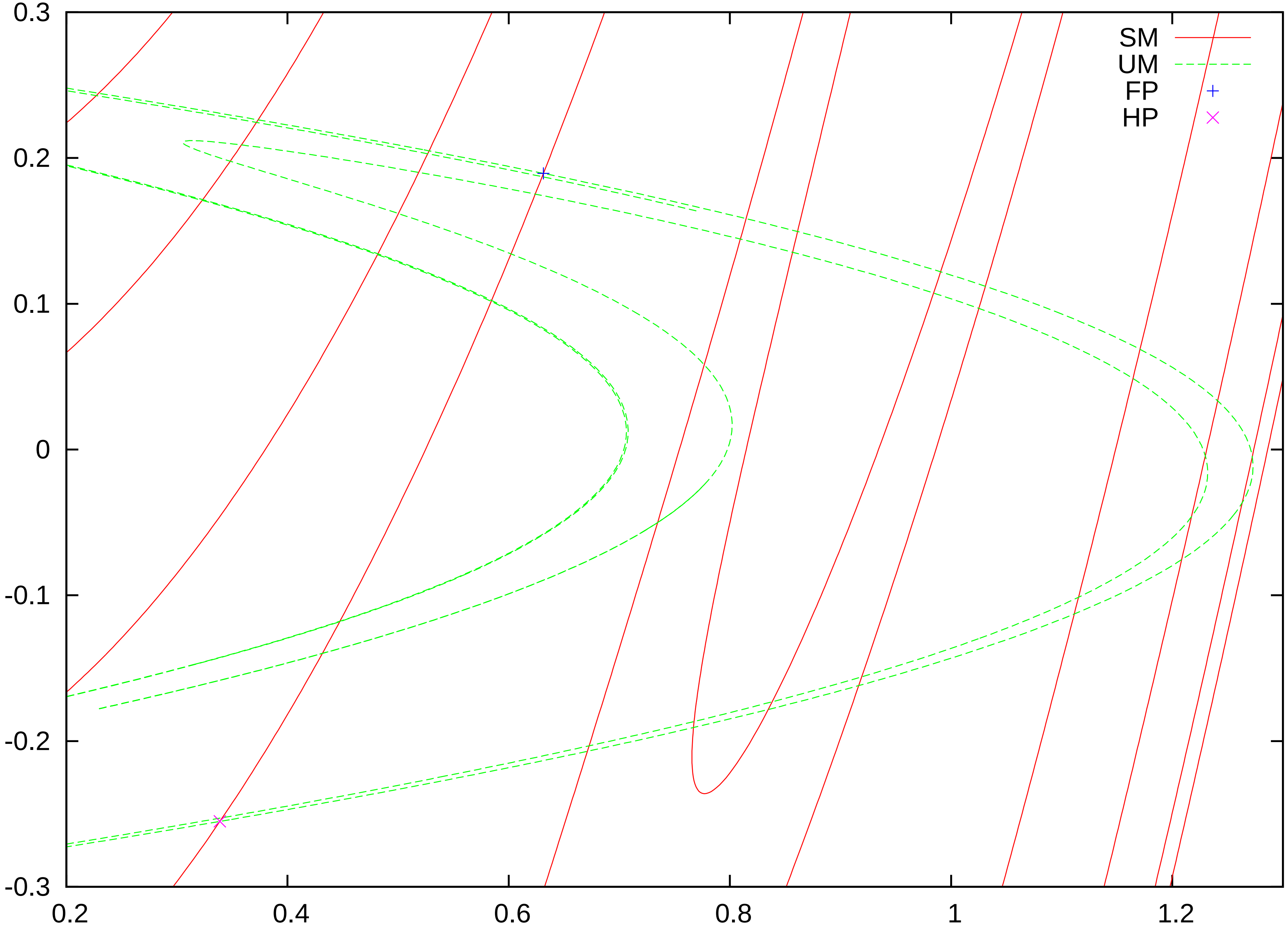

SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2

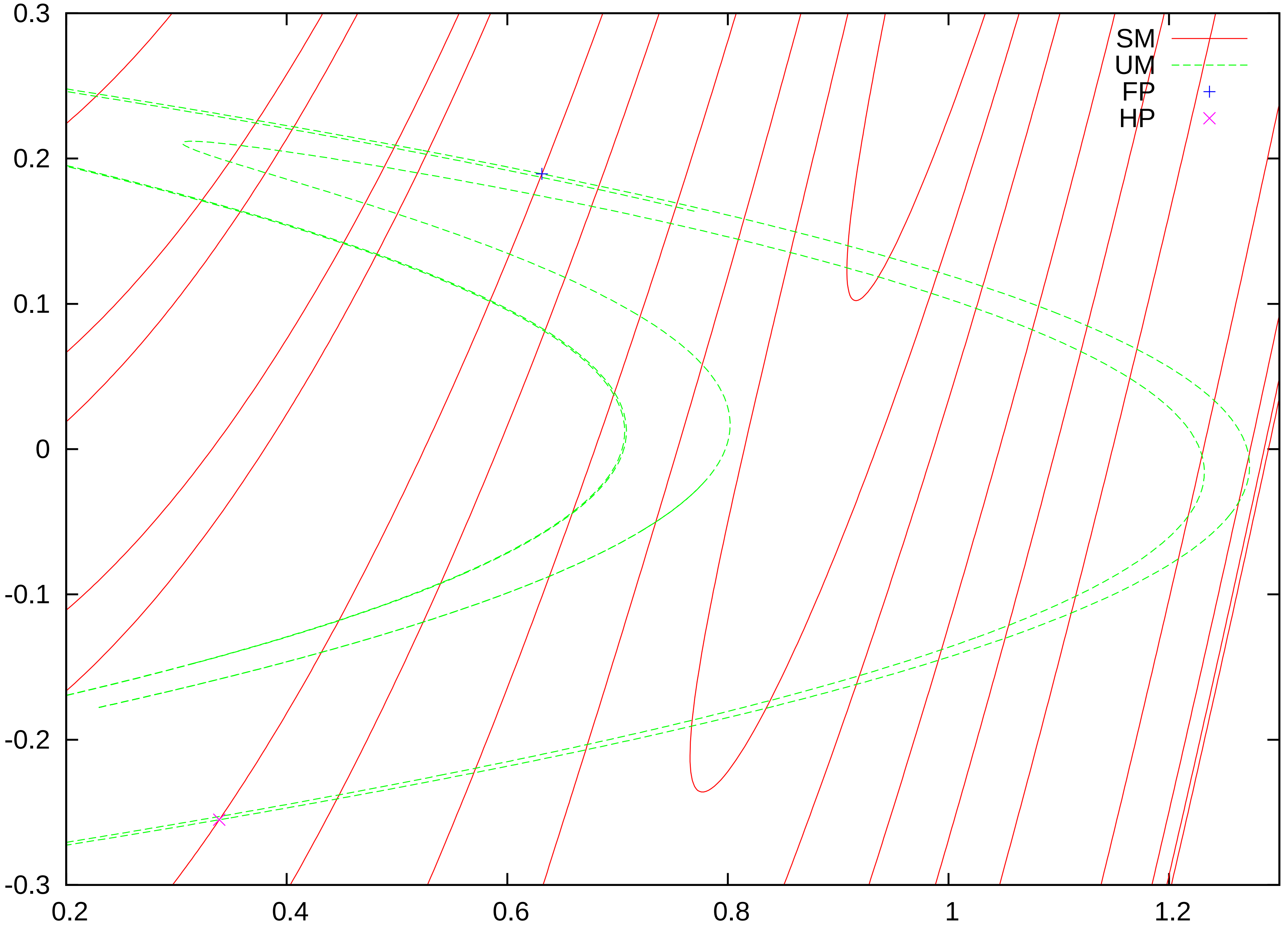

0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2
SM
UM
FP +
HP ×

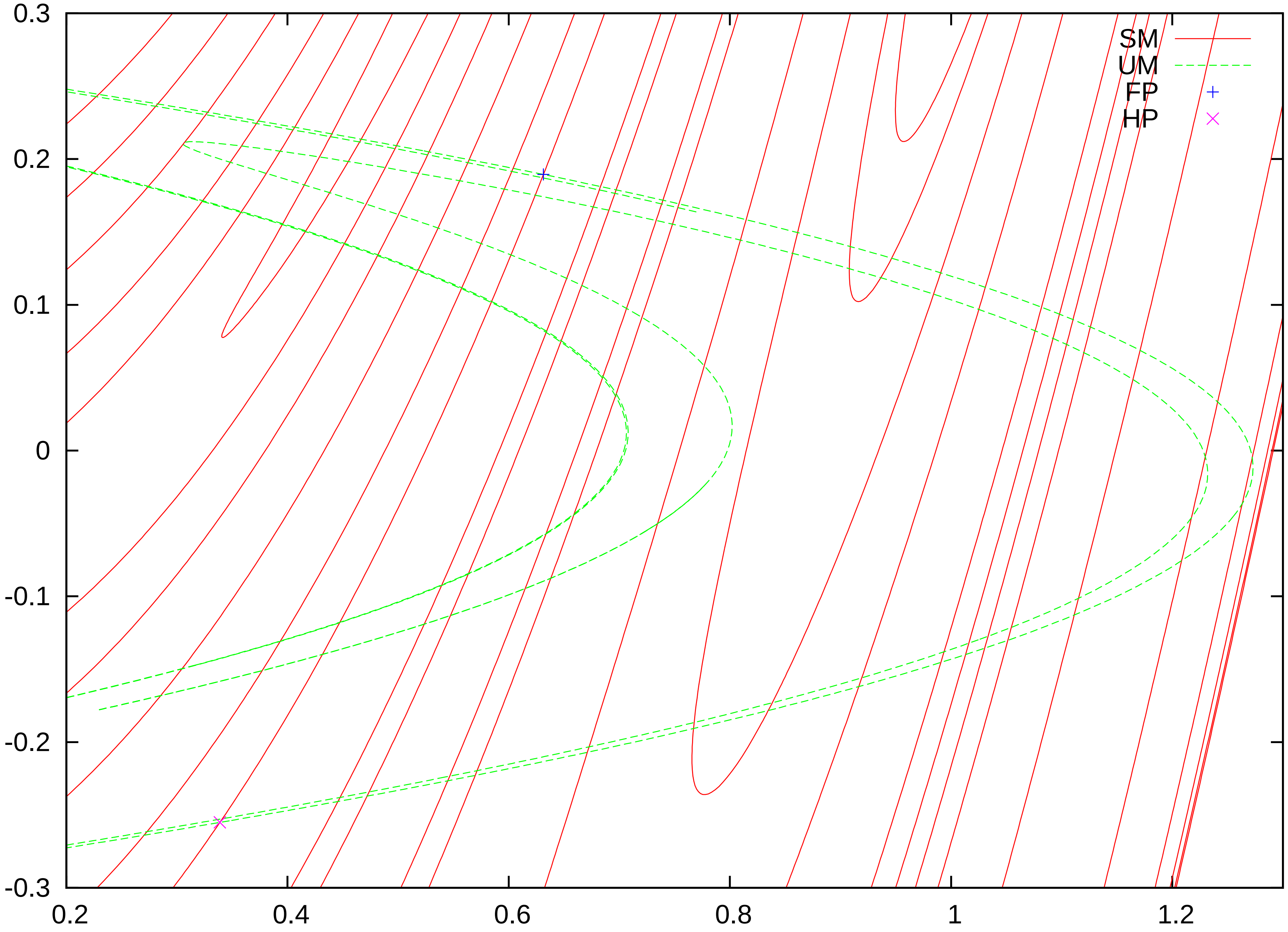

SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2

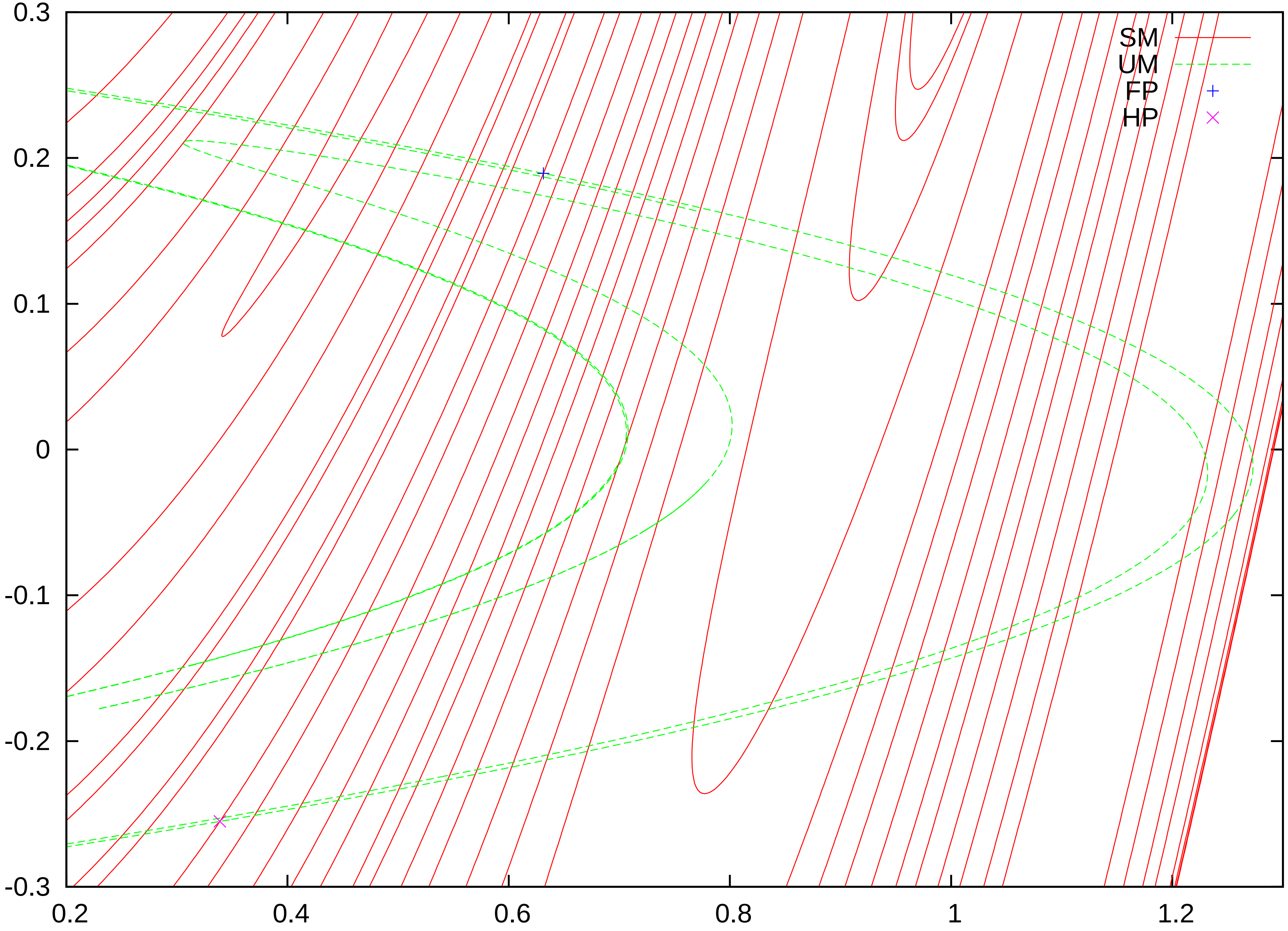

0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2
SM
UM
FP
HP

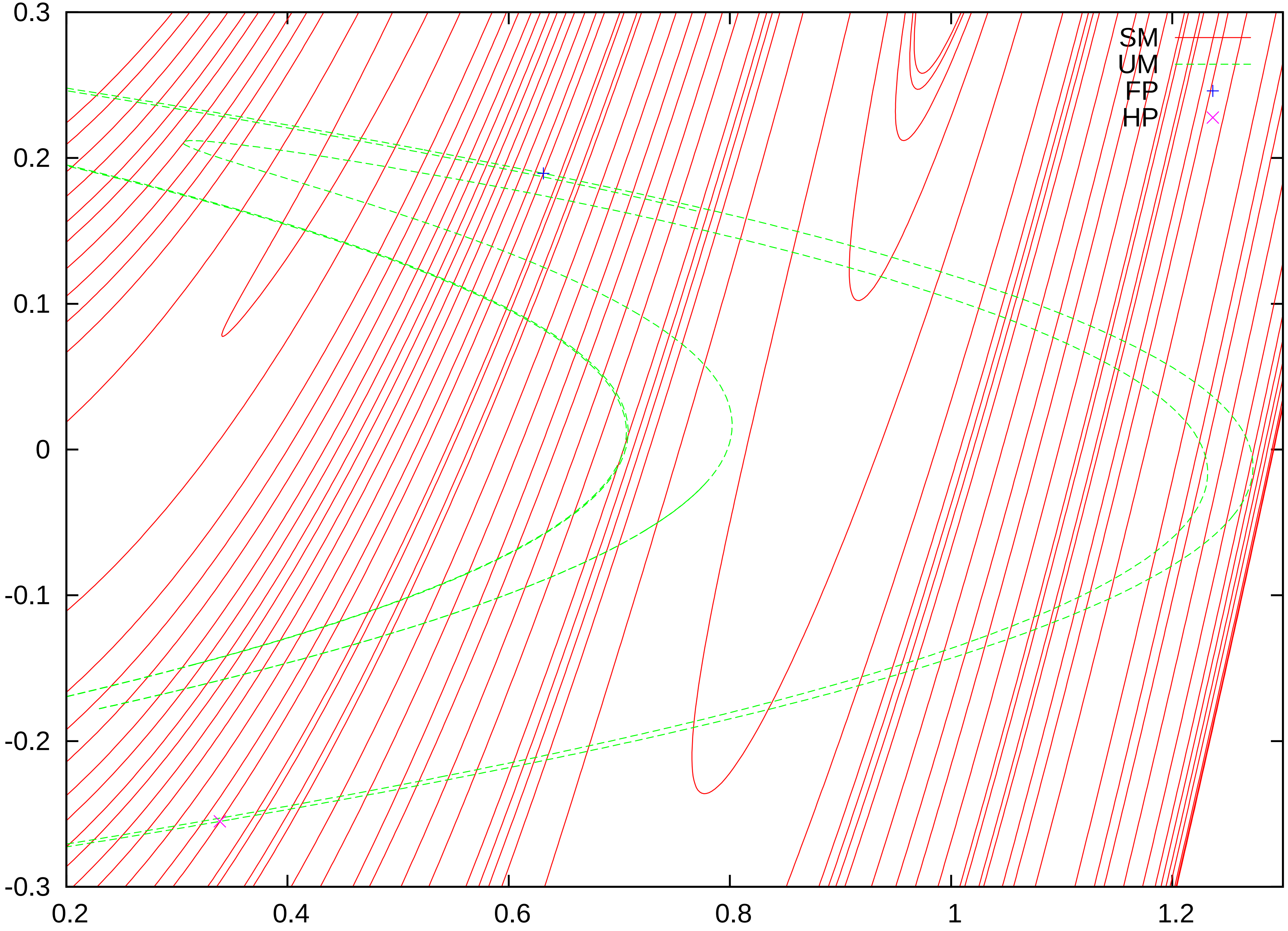

0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2
SM
UM
FP
HP

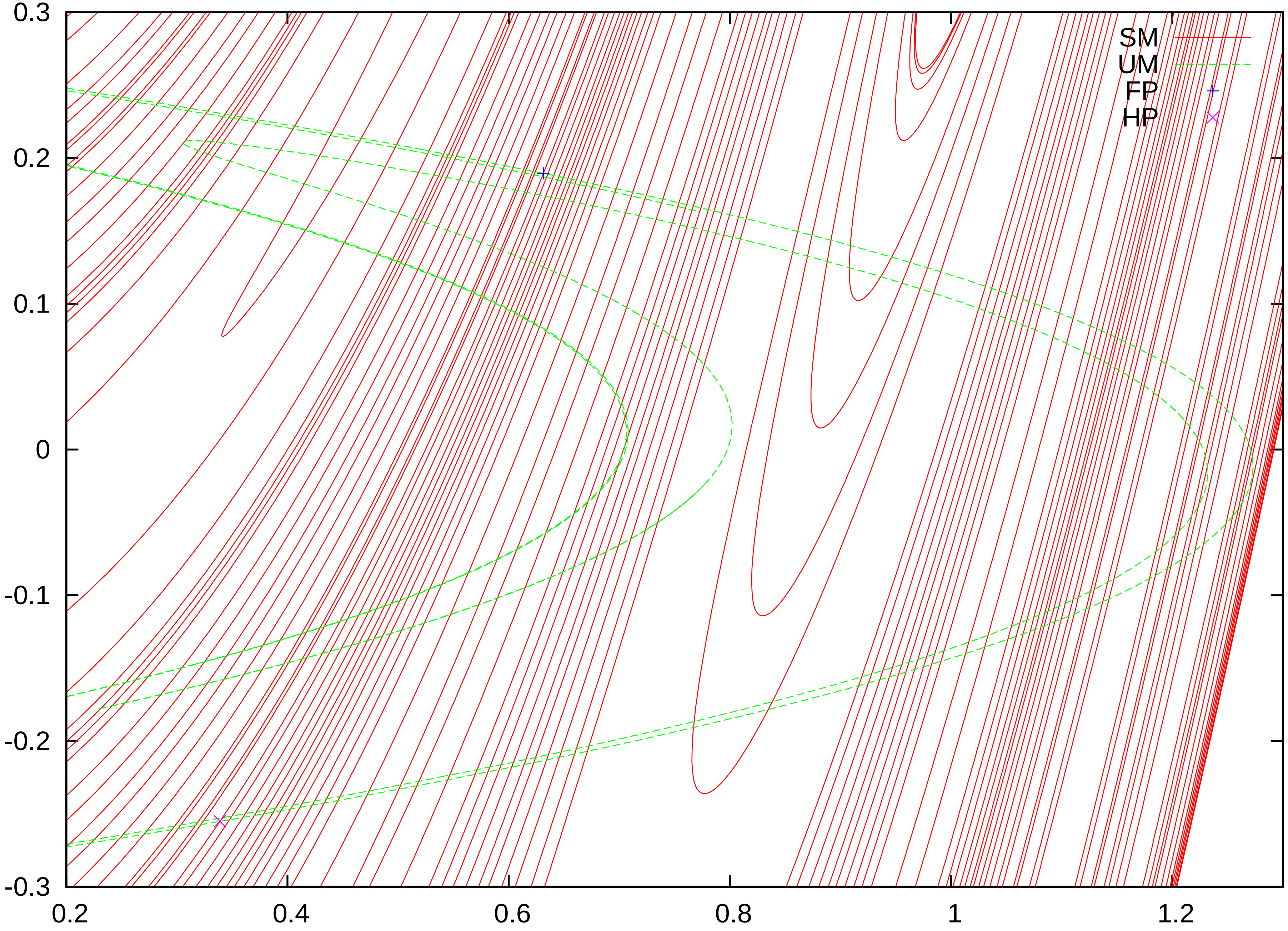

SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2

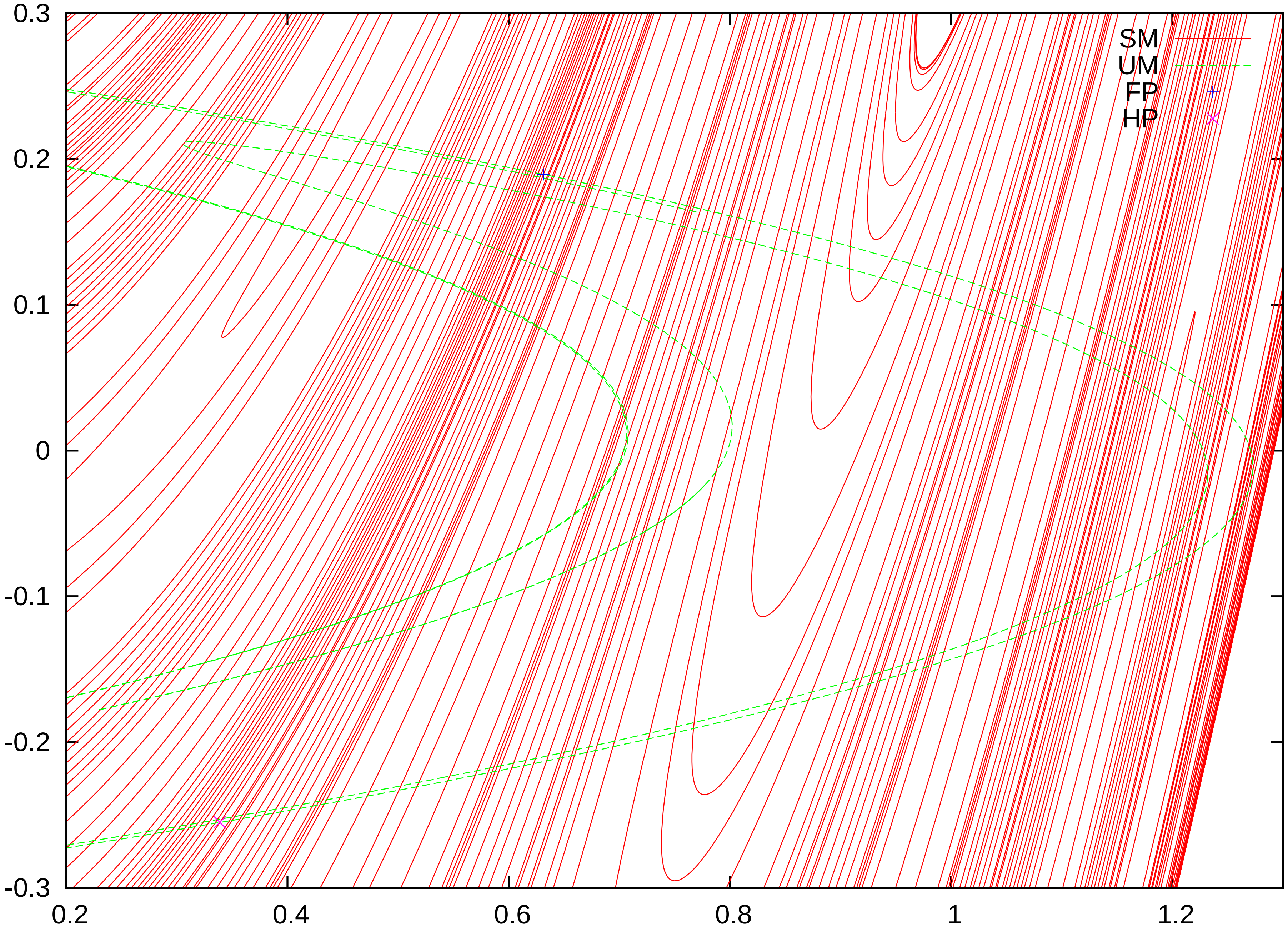

SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2

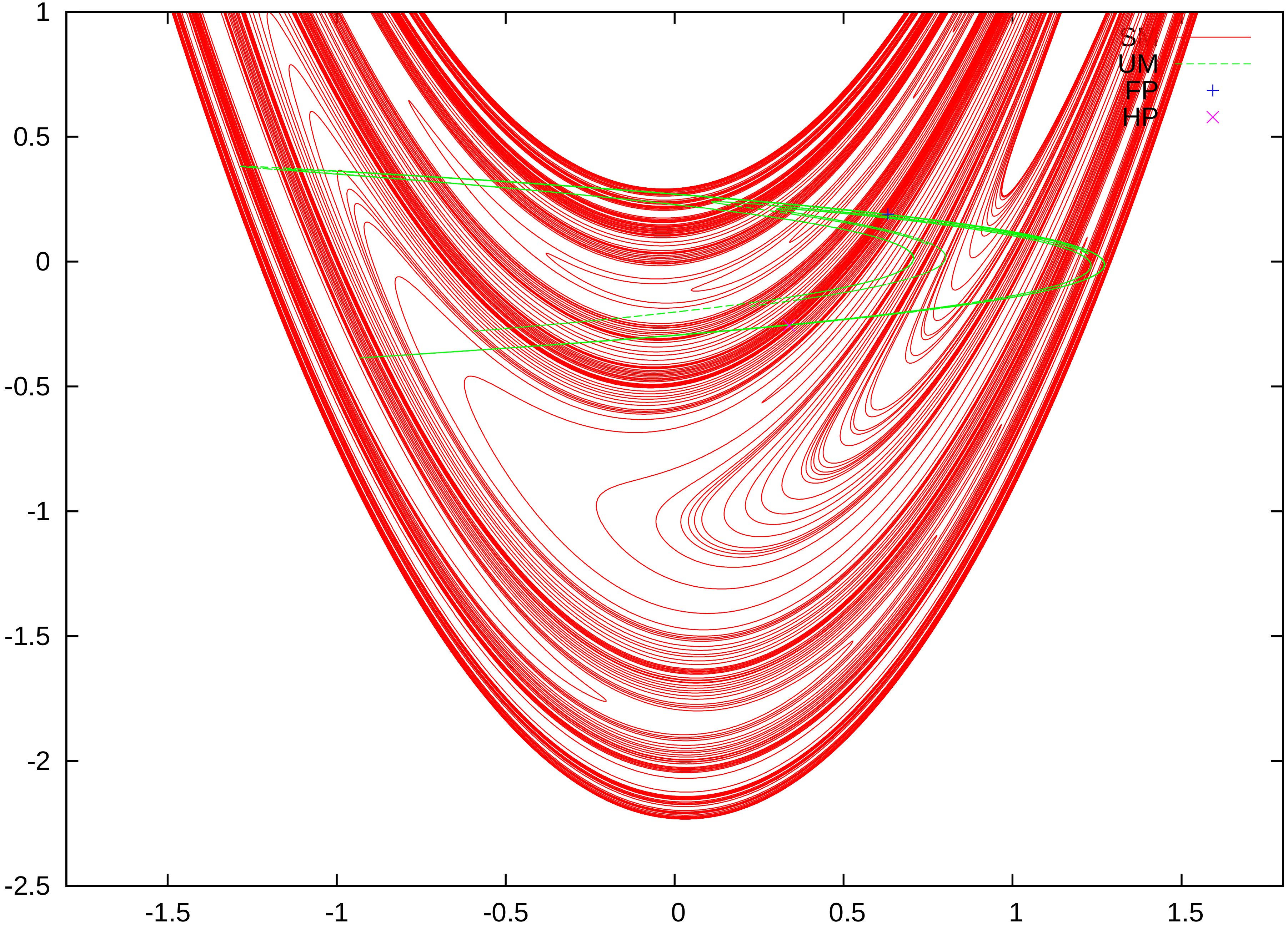

0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2
SM
UM
FP +
HP ×

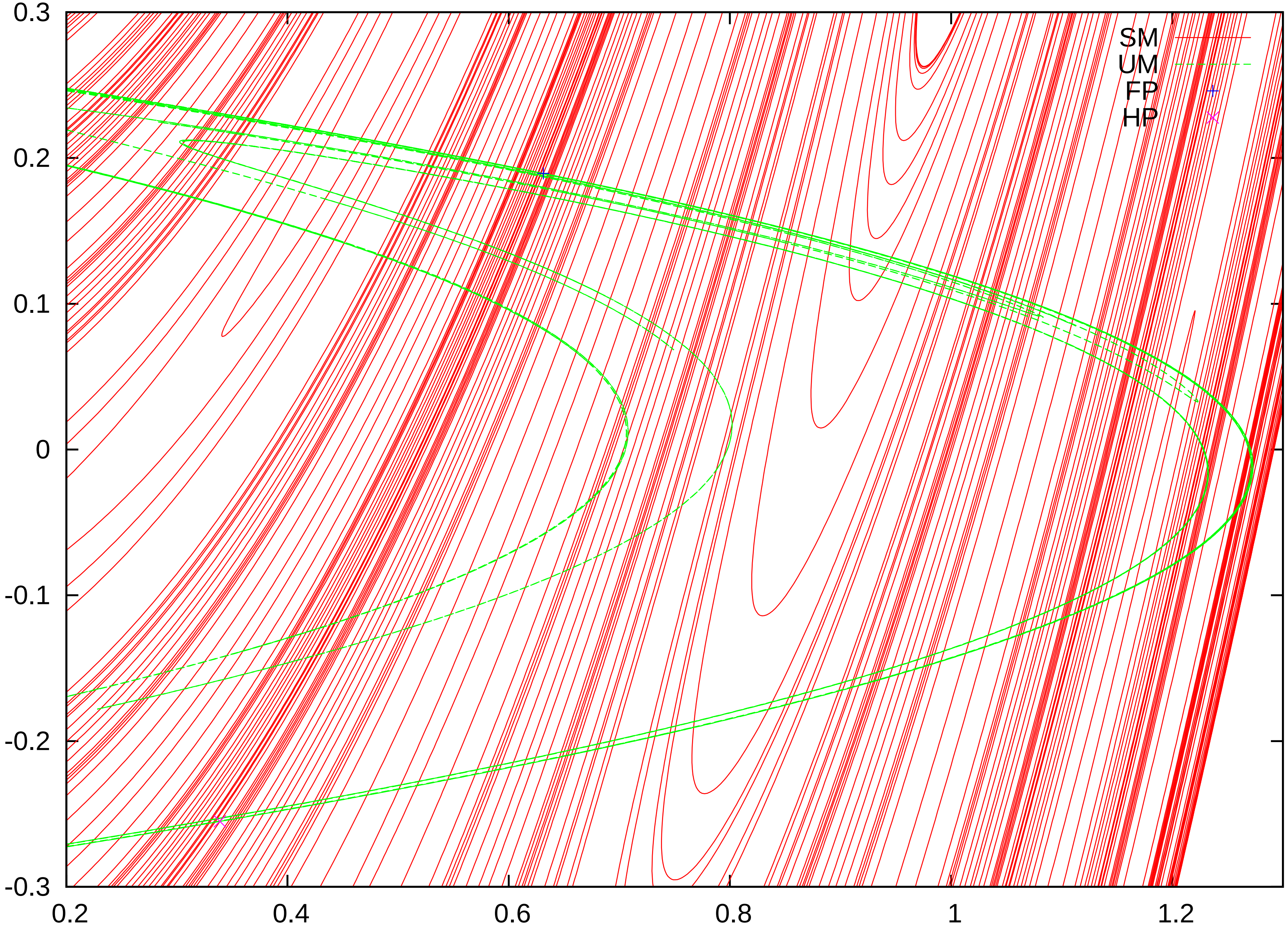

0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2
SM
UM
FP
HP

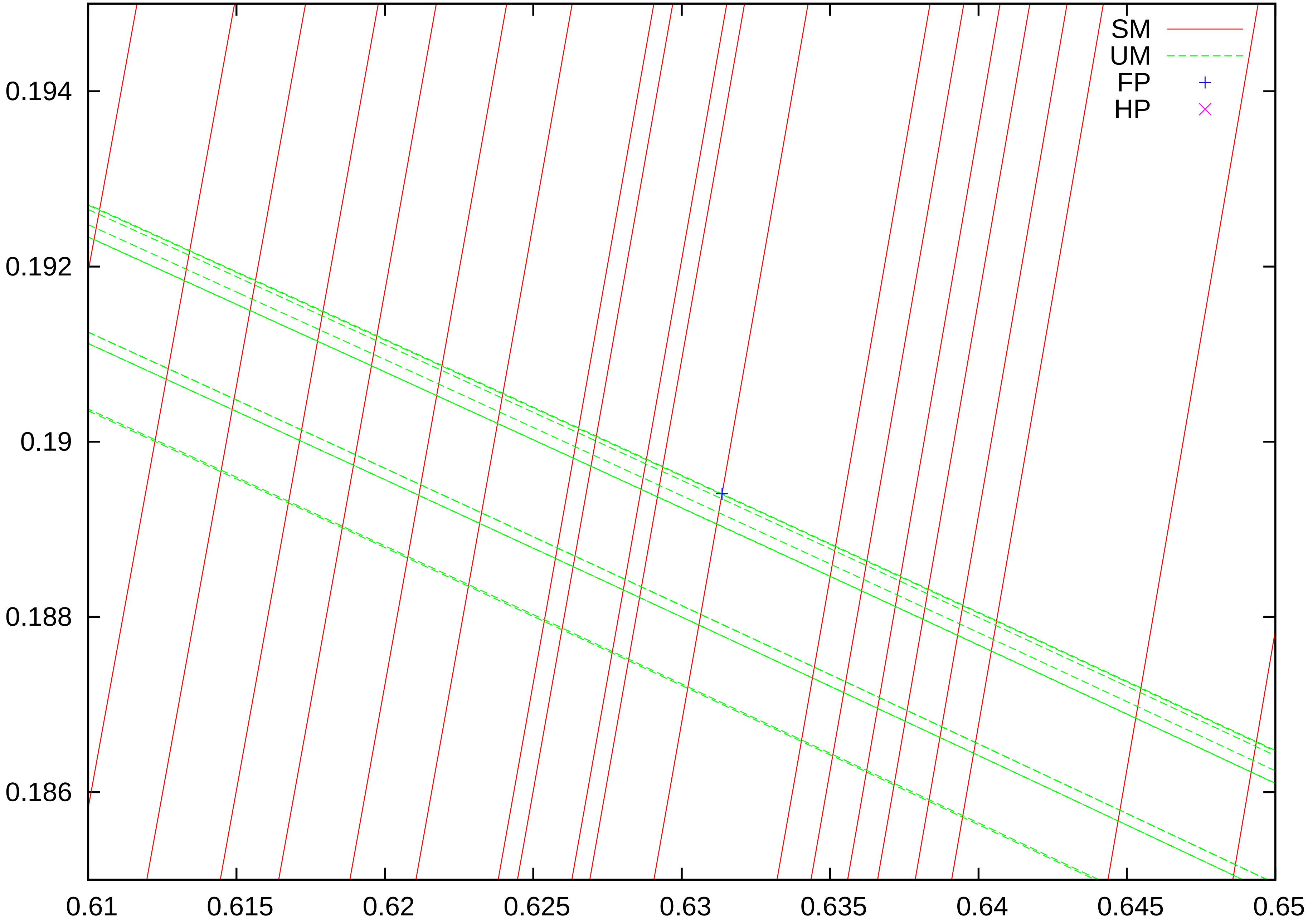

SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2


0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2
SM
UM
FP
HP


0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2
SM
UM
FP
HP


SM
UM
FP
HP
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-1.5
-1
-0.5
0
0.5
1
1.5


SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2


SM
UM
FP
HP
0.194
0.192
0.19
0.188
0.186
0.61
0.615
0.62
0.625
0.63
0.635
0.64
0.645
0.65

Homoclinic and Heteroclinic Points

Rigorous enclosures of the manifolds up to a certain arc length allows:

1. Rigorous enclosures of **homoclinic points** (intersections of stable and unstable manifolds of same fixed points). For example, the "Fundamental" homoclinic point H of the standard Henon map is guaranteed to satisfy

$$H \in (0.3388525493^{912819}_{878994}, -0.2551126297 8^{29170}_{31221}).$$

2. Rigorous enclosures of **heteroclinic points** (intersections of stable and unstable manifolds of separate fixed points). These have practical applications, for example the design of low-energy transfers in restricted three body problem.

Symbolic Dynamics

Rigorous insight into the behavior of a dynamical system can be obtained by studying symbolic dynamics. This refers to a **projection** of the dynamics into finite sets of "symbols", and study of how these evolve under map. Prime example: determine suitable subsets of variables and study their mapping properties rigorously.

Ideal candidates:

Curvilinear Rectangles: having homoclinic points in their corners, pieces of unstable and stable manifold, respectively, as their sides.

Advantages: Their mapping properties can be rigorously understood by the knowledge of the location of **all** homoclinic points up to a certain arc length of stable and unstable manifold, as well as the **mapping properties** of these homoclinic points.

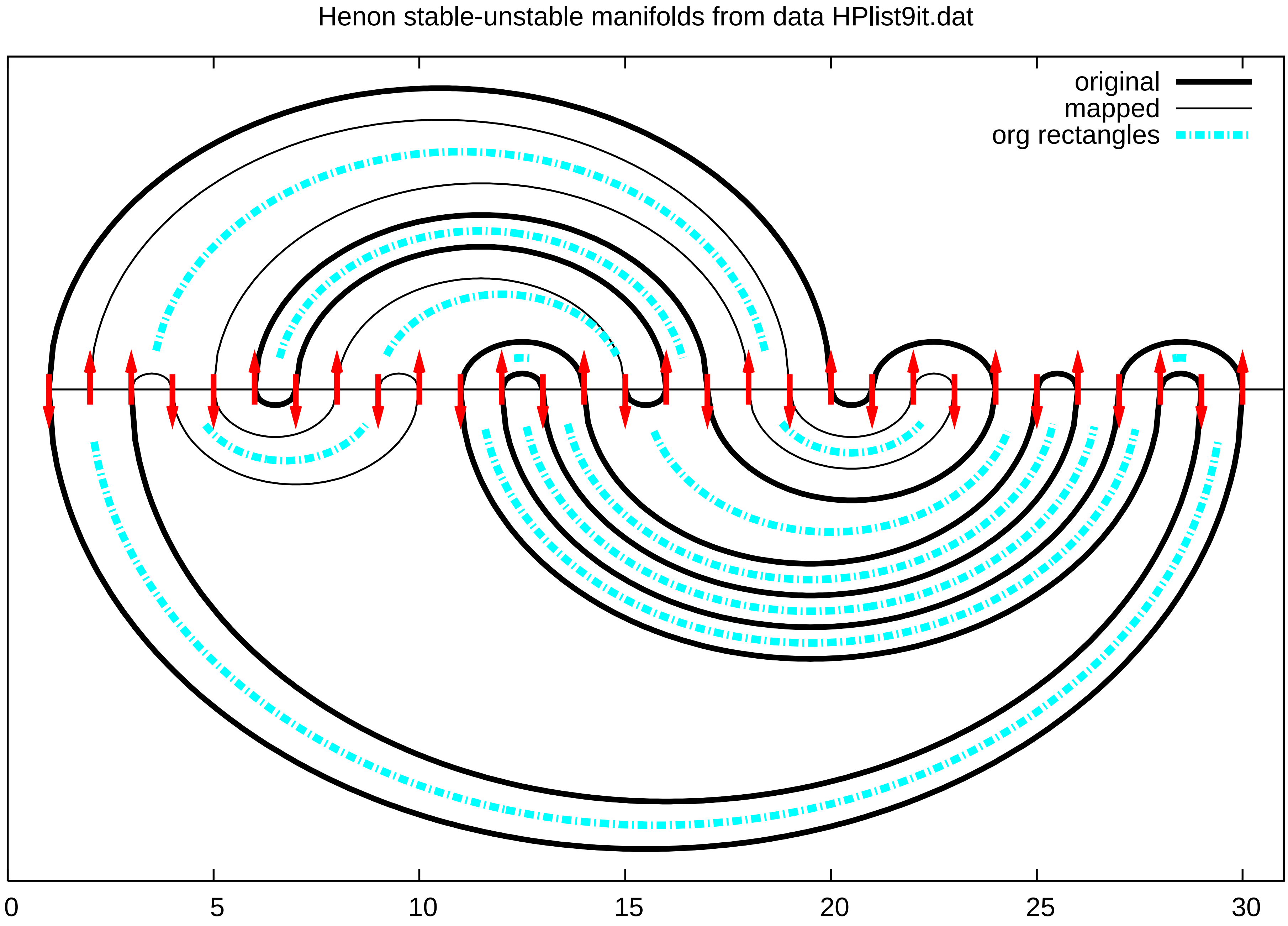
Rigorous Computational Symbolic Dynamics

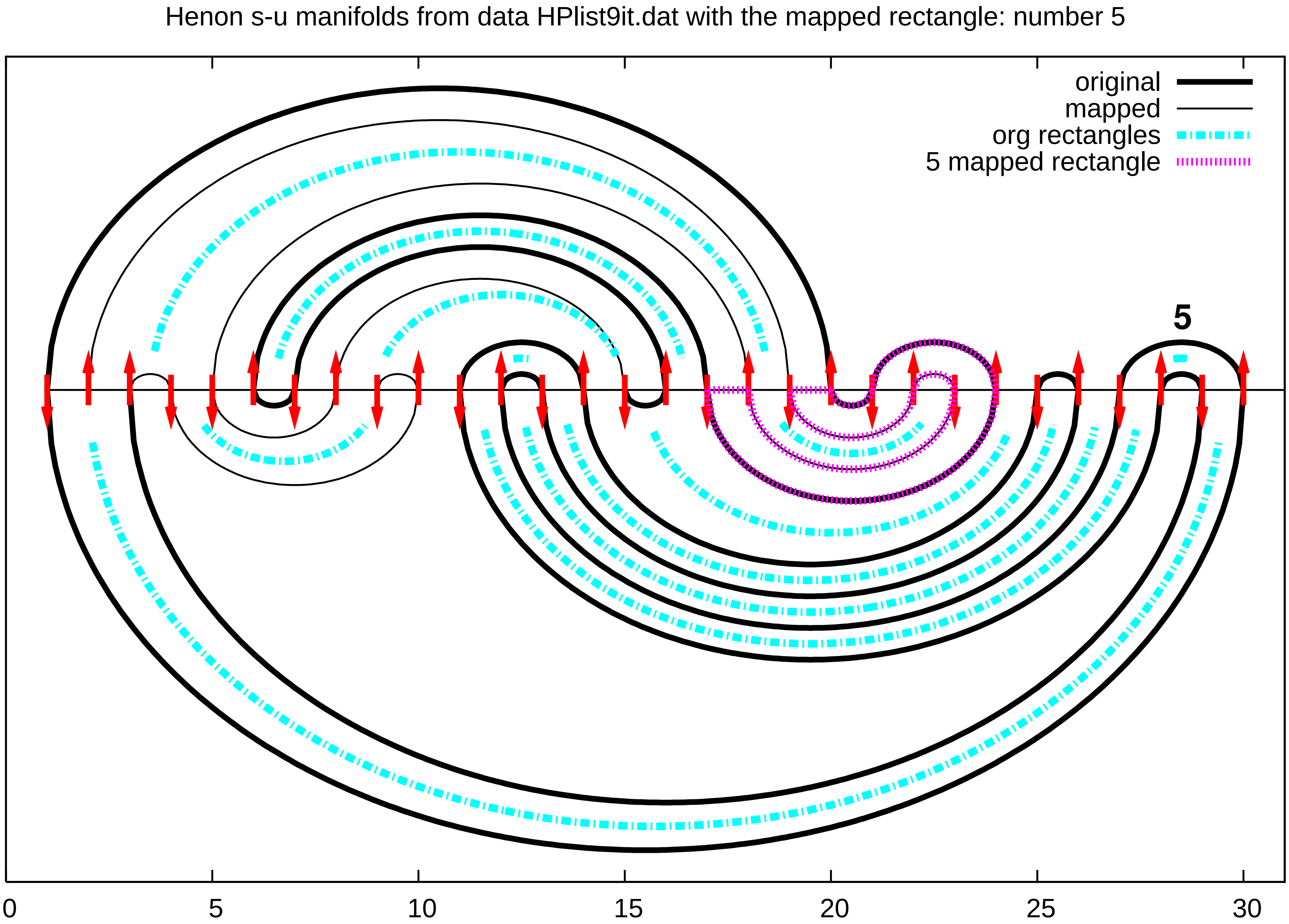
Using Taylor model based flow integrators and normal form methods, can set up even very complicated symbolic dynamics. Let two initial pieces of stable and unstable manifold be given.

1. Rigorously enclose **ALL** homoclinic points of using the rigorous global optimizer COSY-GO.
2. Determine rigorous **parent-child relationships** of these homoclinic points.

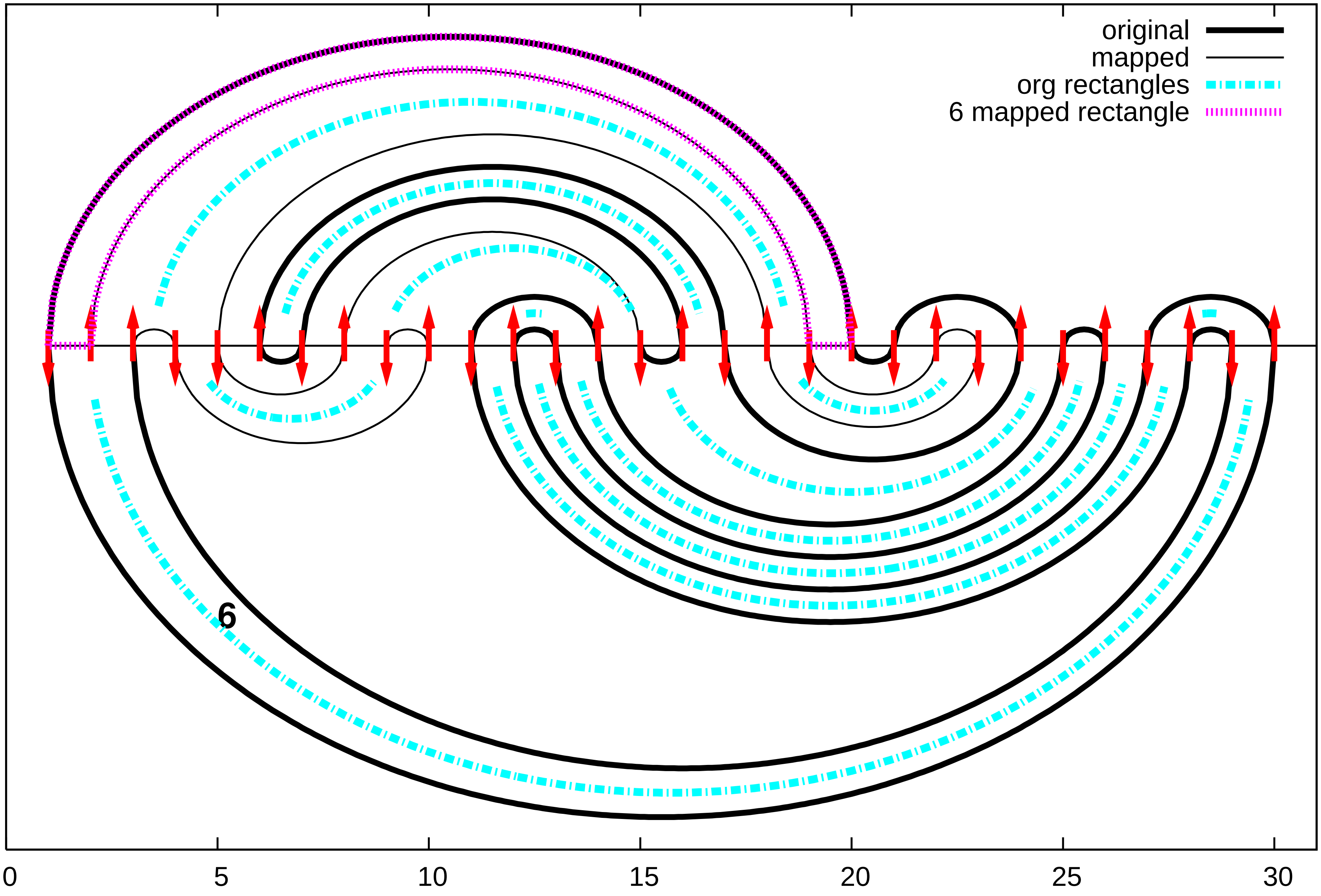
This allows the rigorous determination the mapping properties of curvilinear rectangles, which can be described by the so-called incidence matrix. The largest eigenvalue of it is a lower bound of the topological entropy.

Note: probably the first such attempt at a rigorous dynamics was done by Piotr Zgliczynski for the Henon map, proving that it follows a horseshoe dynamics, with

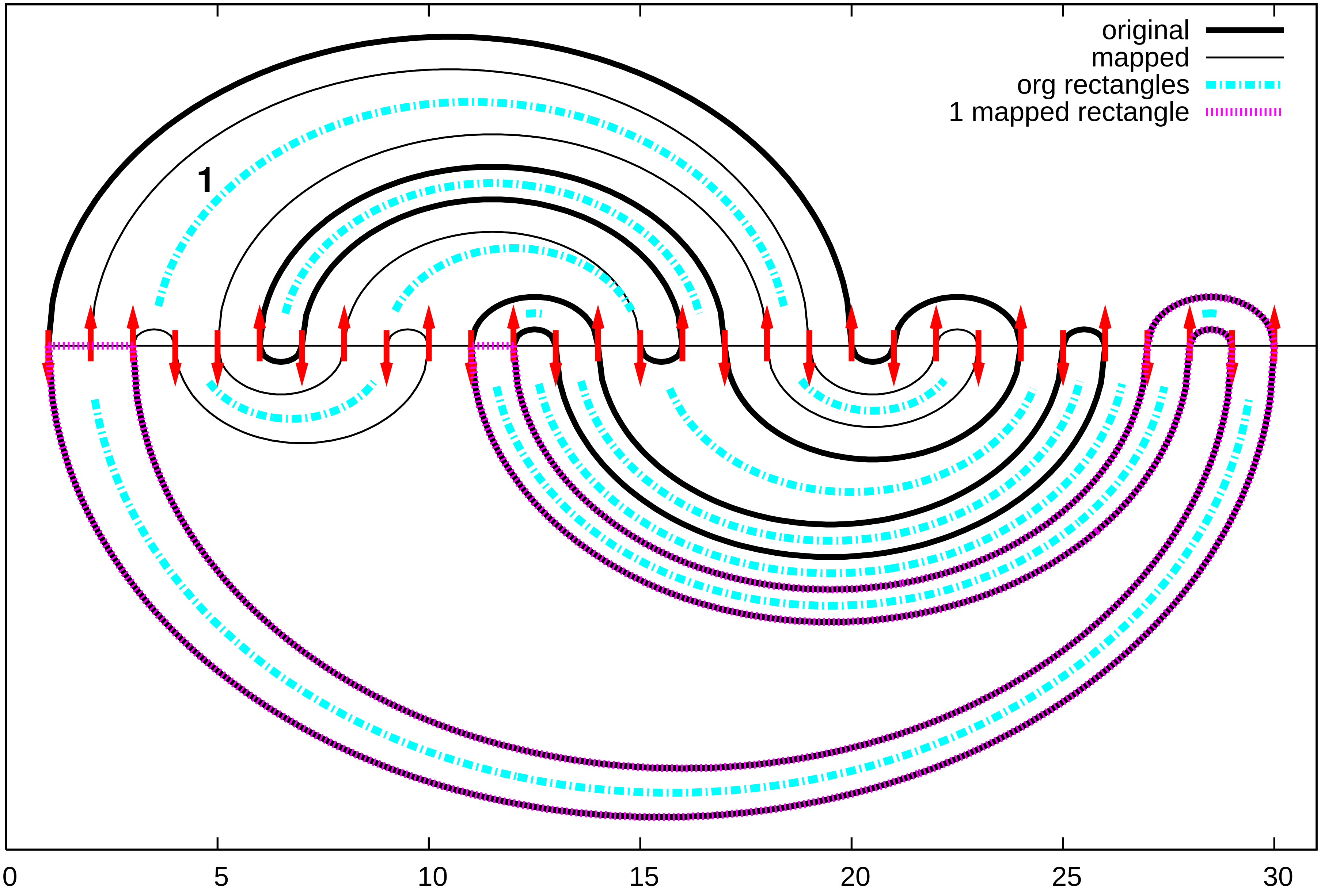

Henon stable-unstable manifolds from data HPlist9it.dat
original
mapped
org rectangles
0
5
10
15
20
25
30

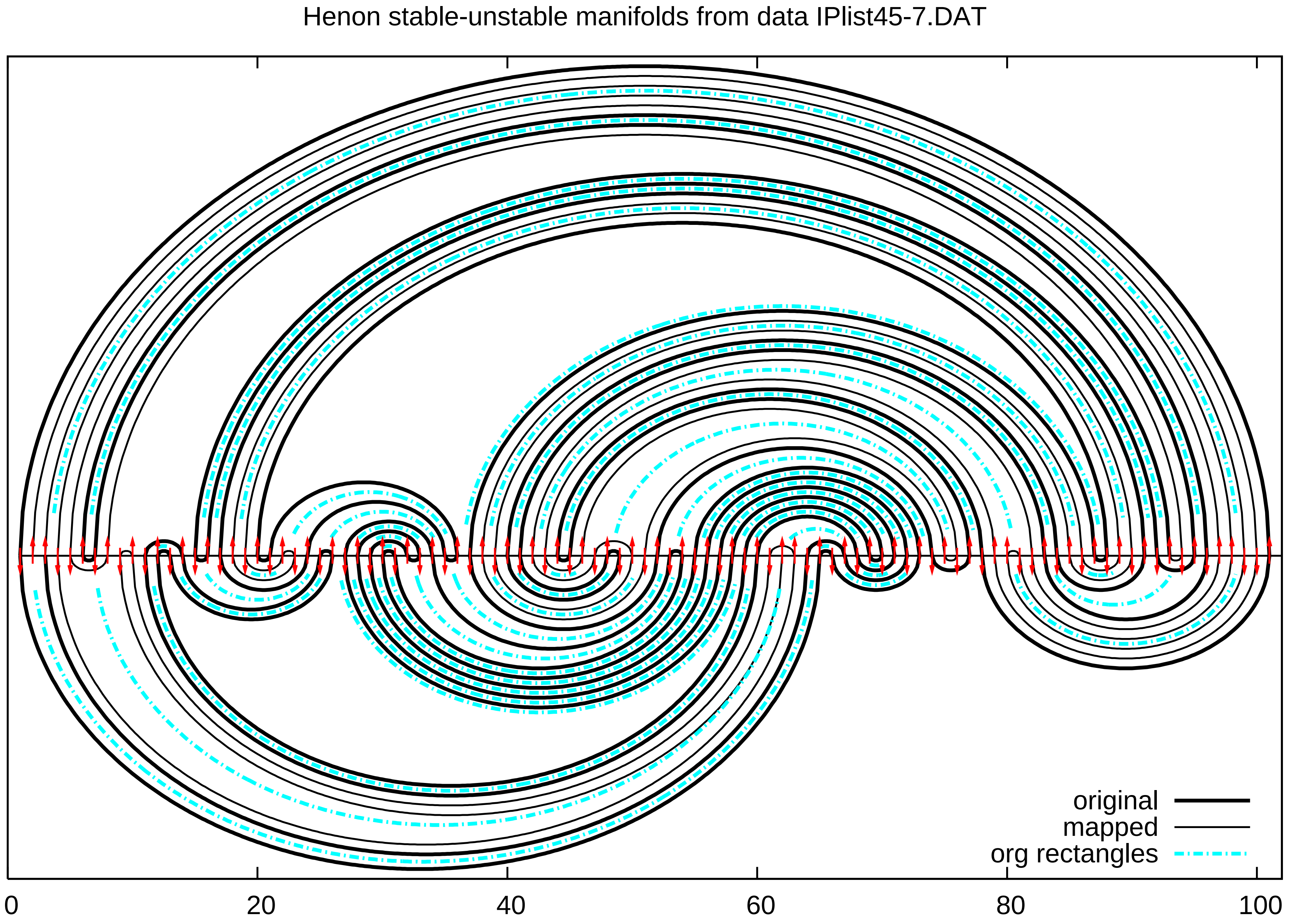

Henon s-u manifolds from data HPlist9it.dat with the mapped rectangle: number 5
original
mapped
org rectangles
5 mapped rectangle
5
0
5
10
15
20
25
30

Henon s-u manifolds from data HPlist9it.dat with the mapped rectangle: number 6

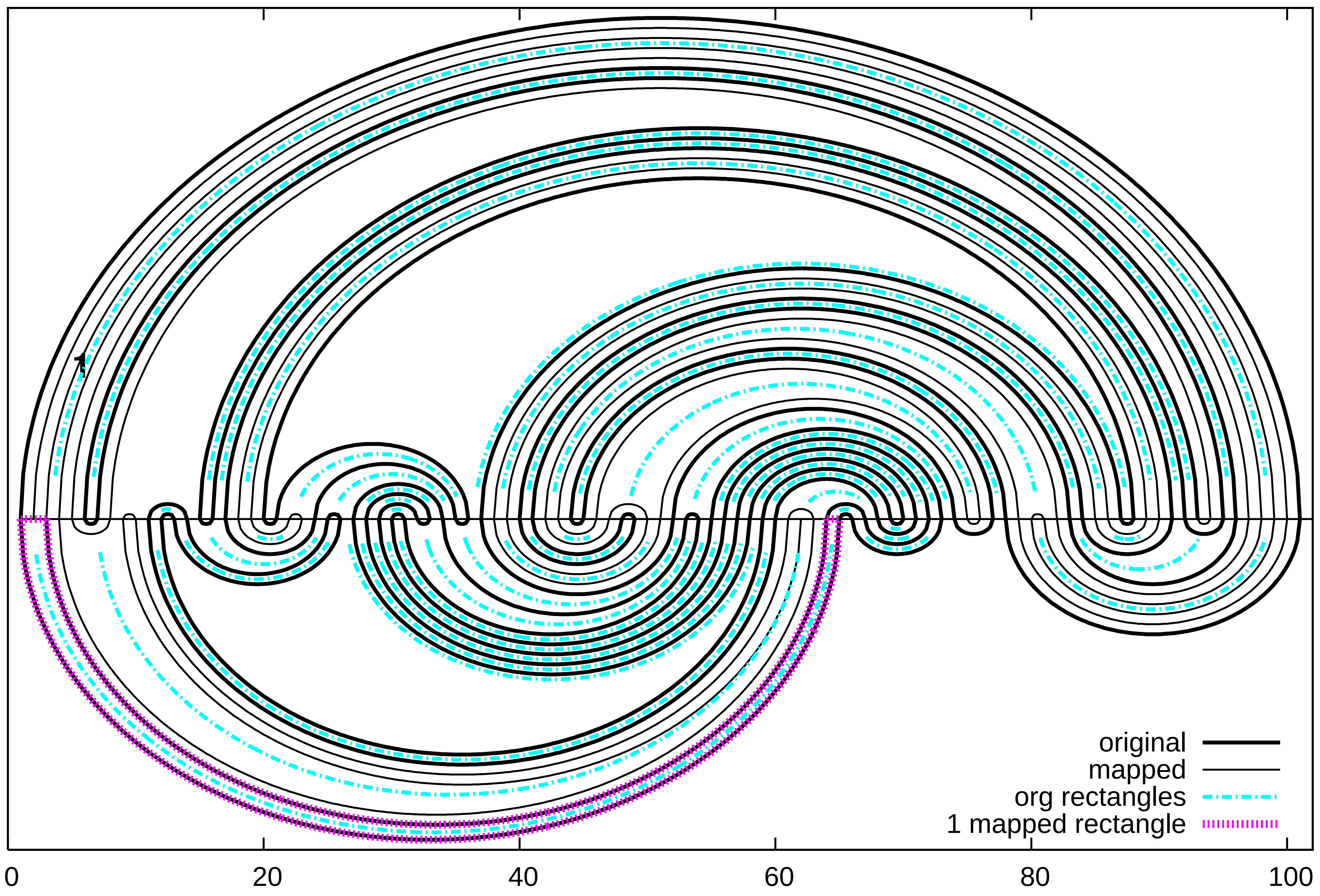


Henon s-u manifolds from data HPlist9it.dat with the mapped rectangle: number 1

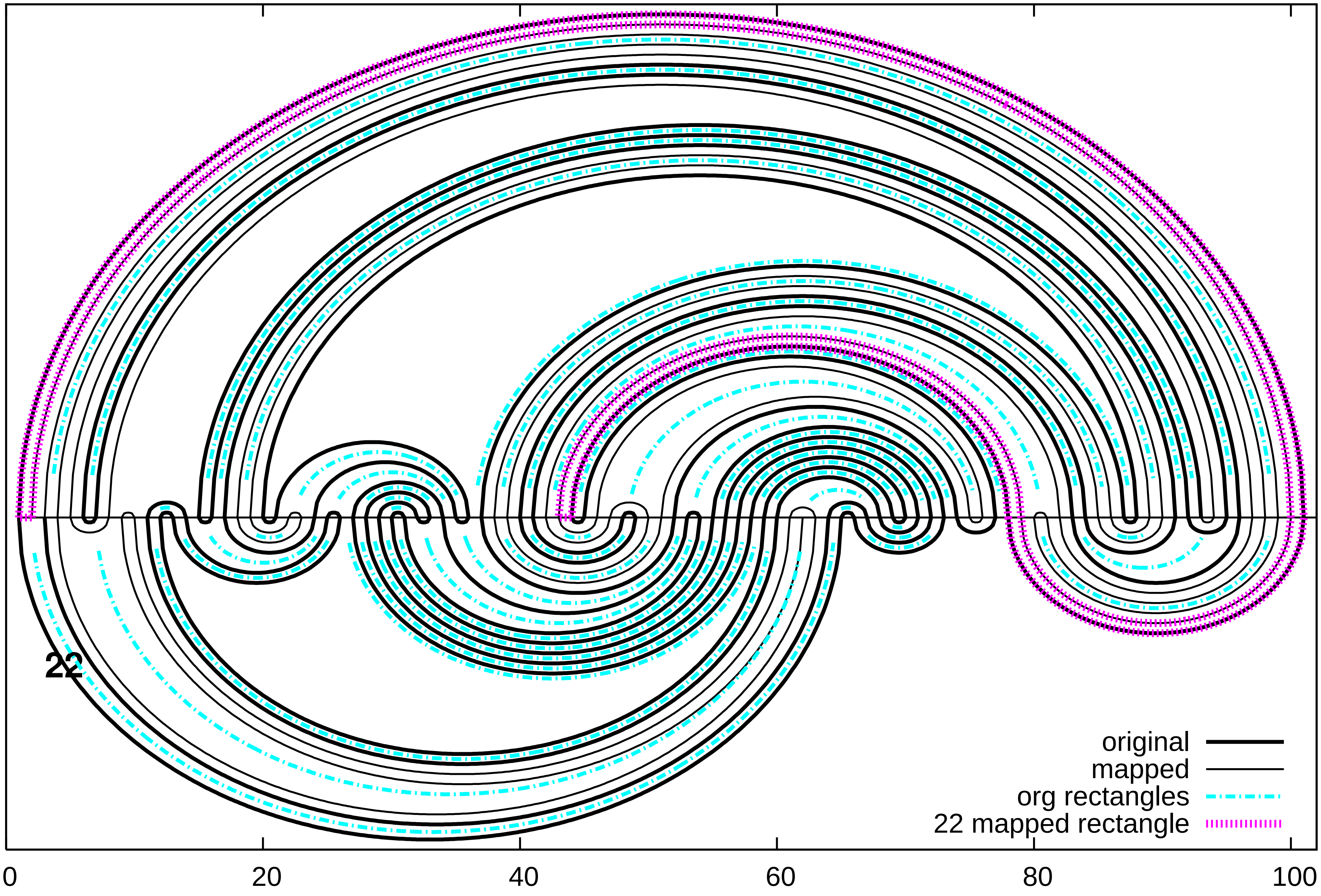



Henon stable-unstable manifolds from data IPlist45-7.DAT
original
mapped
org rectangles
0
20
40
60
80
100

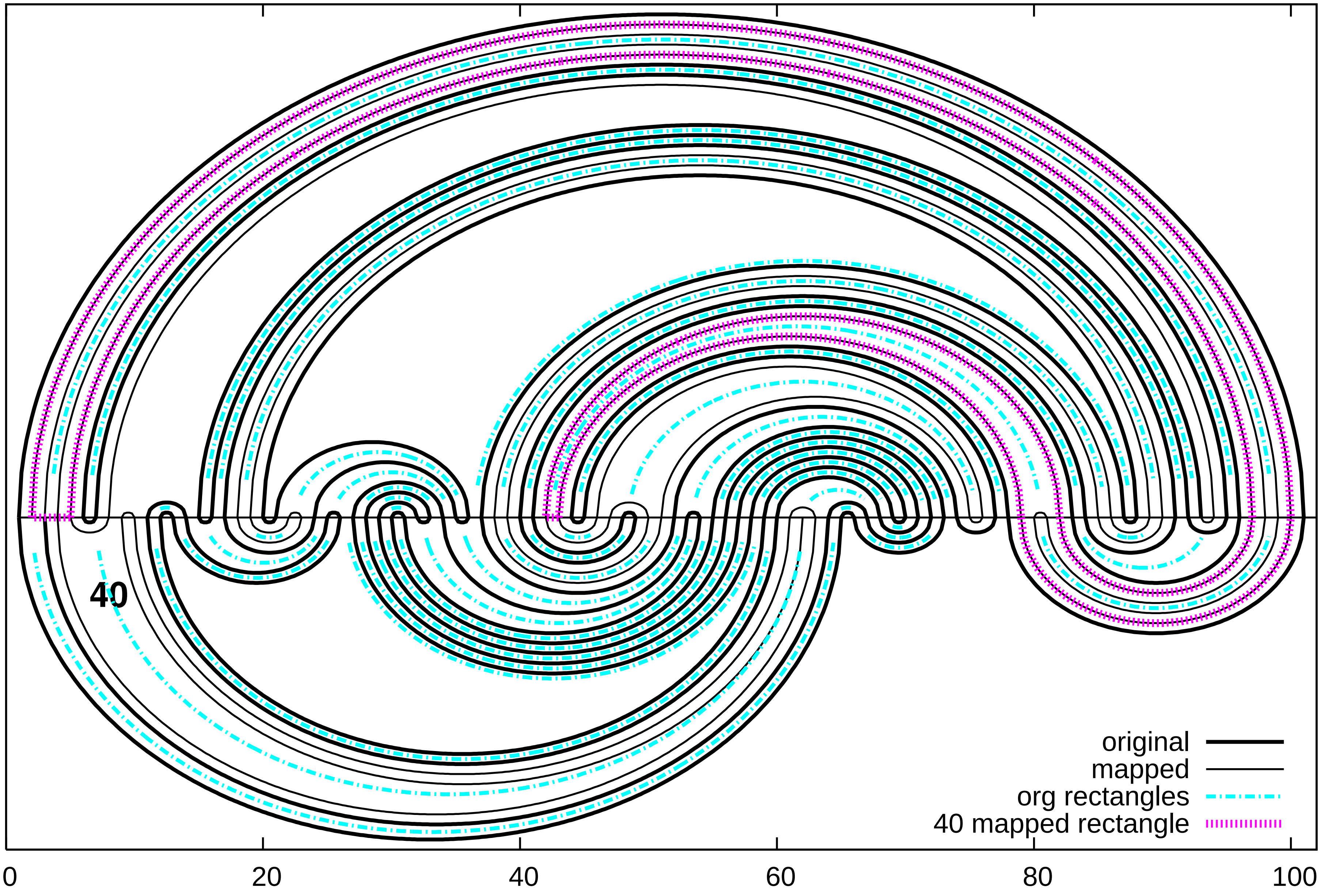
Henon s-u manifolds from data IPlist45-7.DAT with the mapped rectangle: number 1



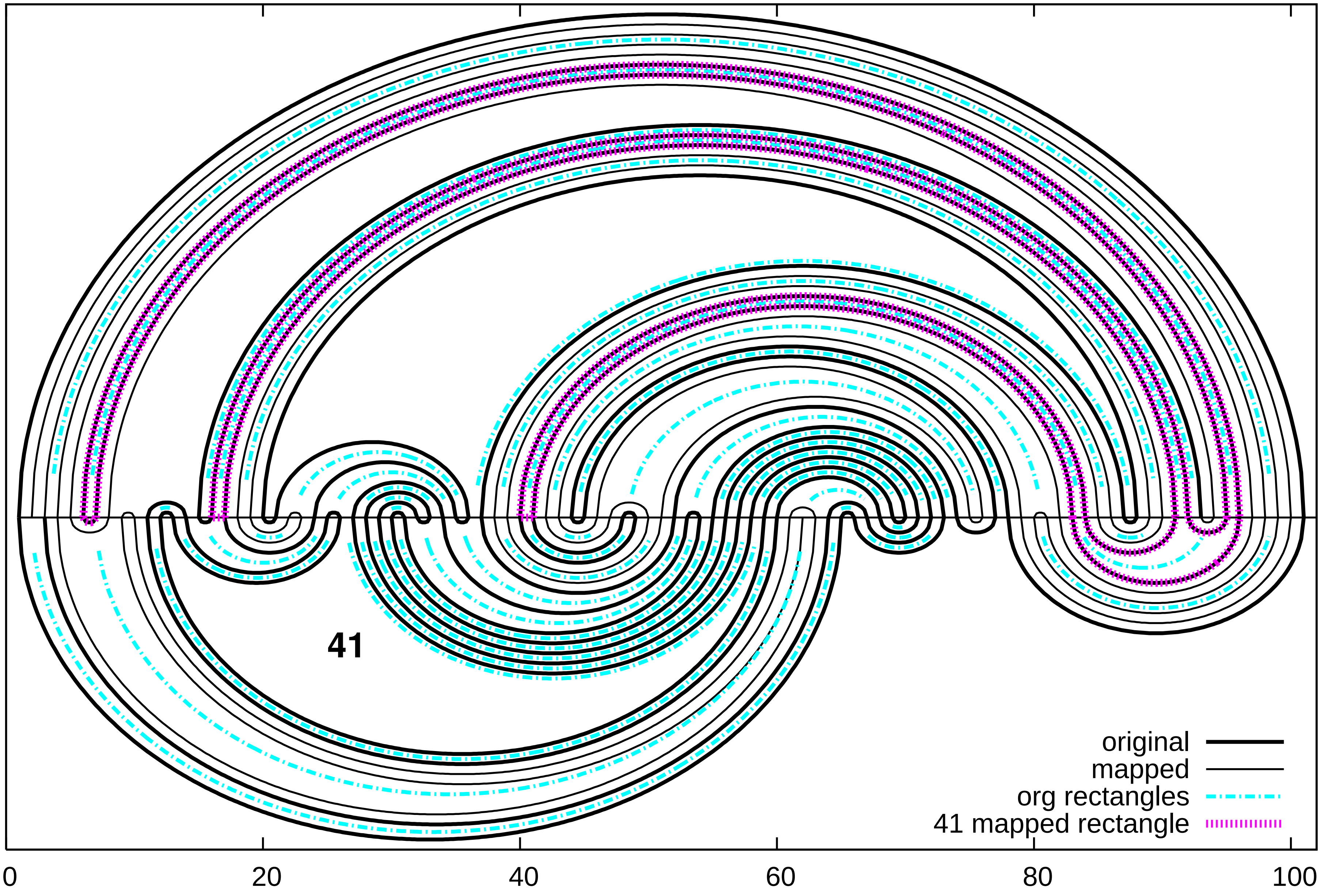
Henon s-u manifolds from data IPlist45-7.DAT with the mapped rectangle: number 22

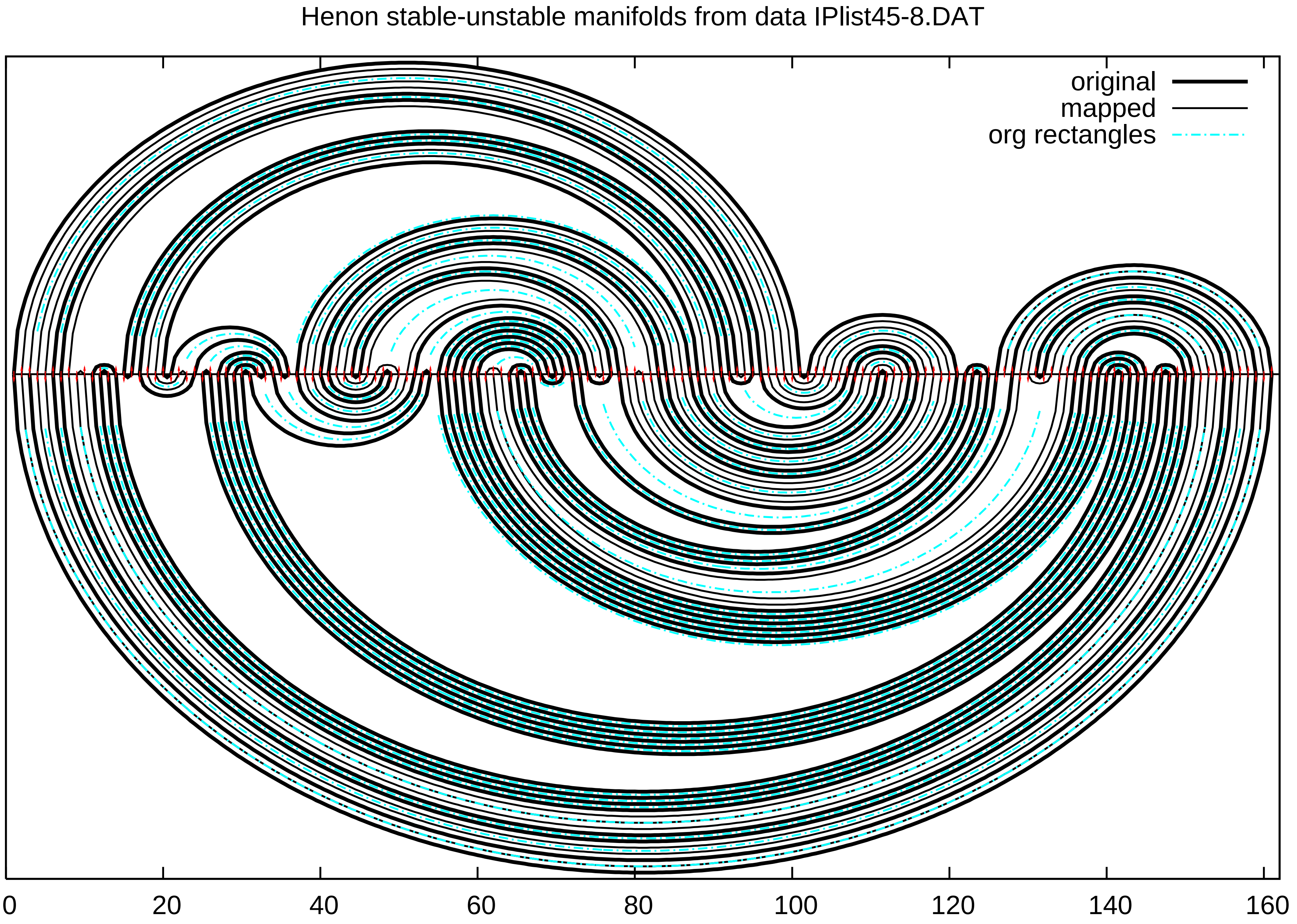


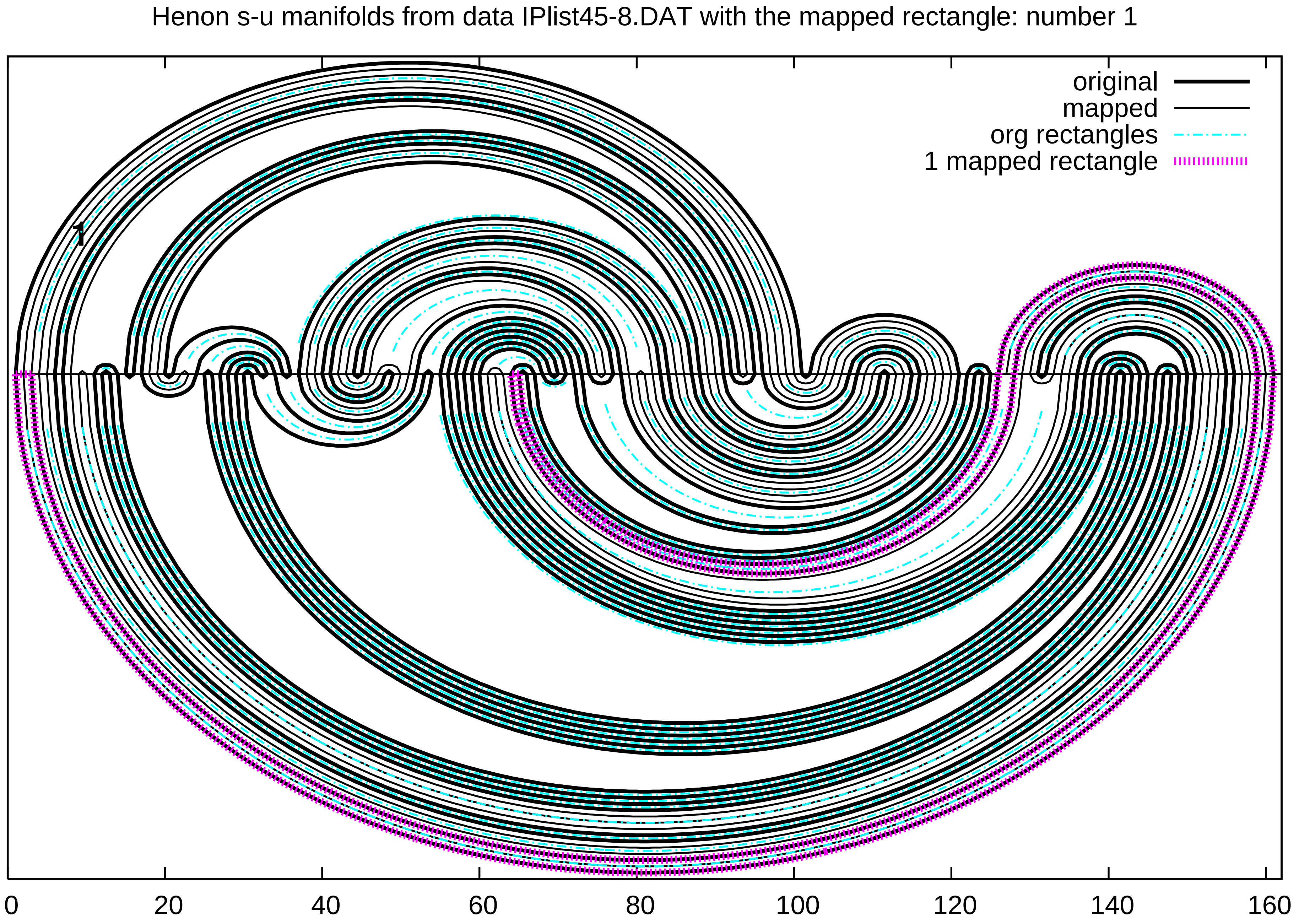
Henon s-u manifolds from data IPlist45-7.DAT with the mapped rectangle: number 40



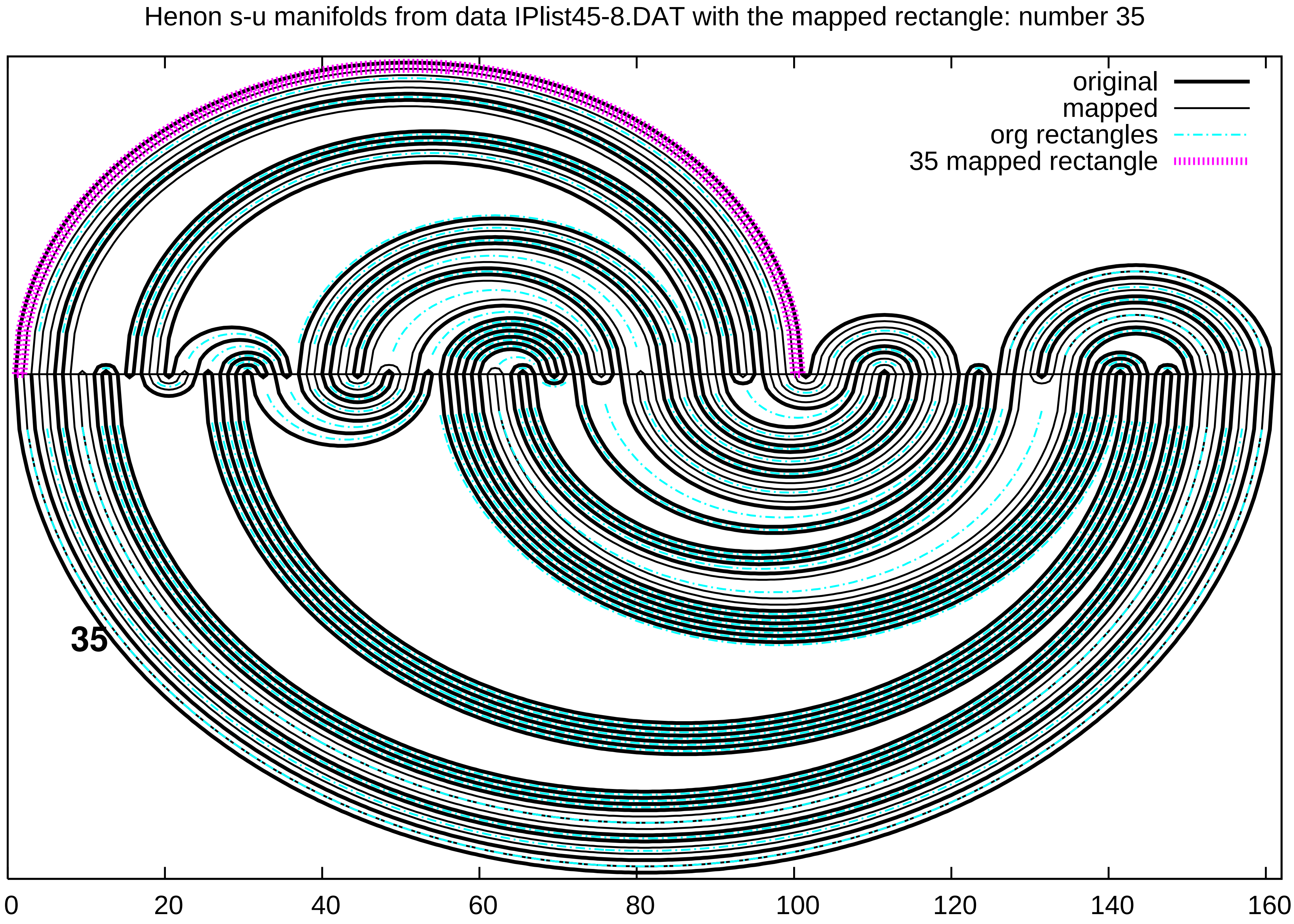
Henon s-u manifolds from data IPlist45-7.DAT with the mapped rectangle: number 41



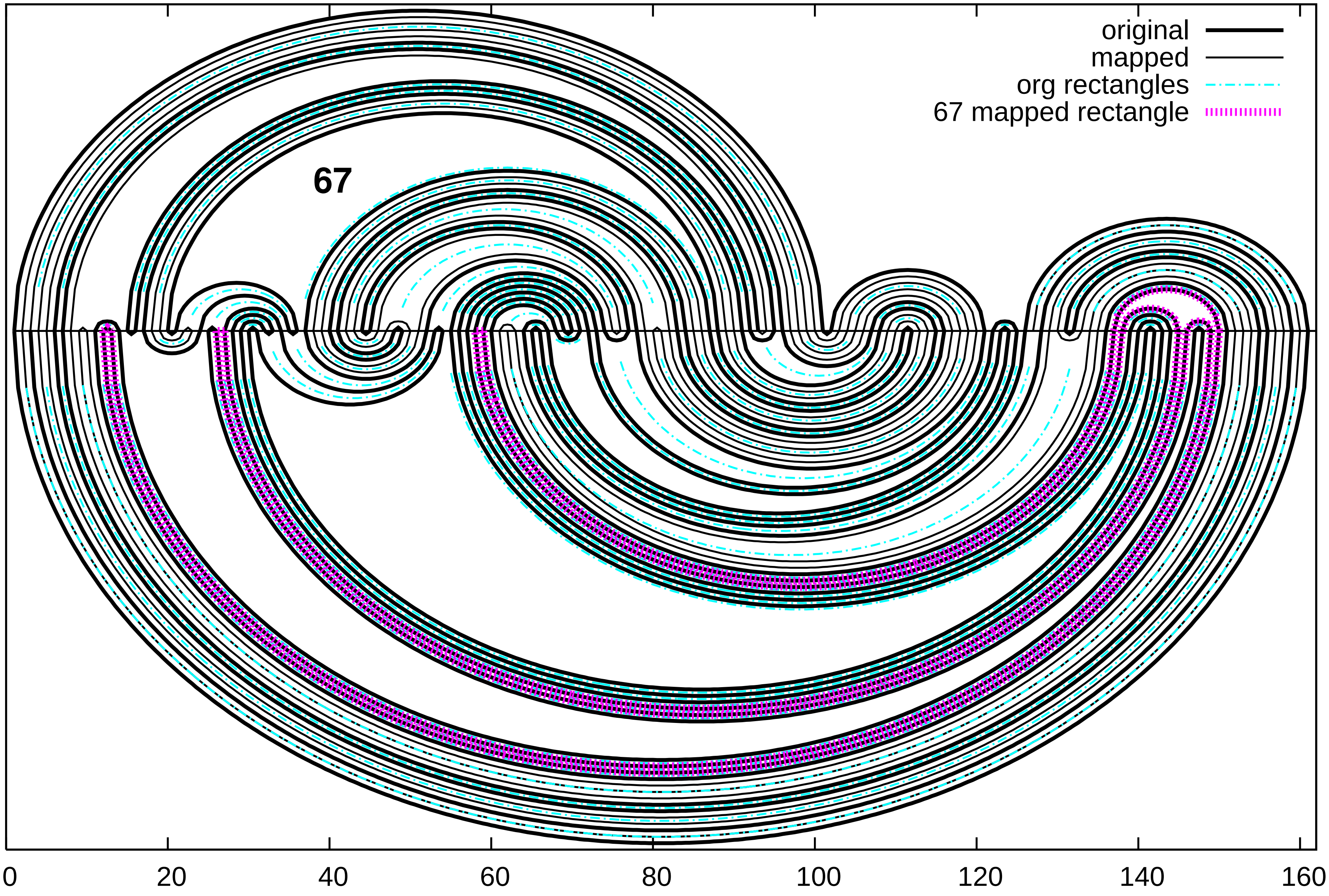

Henon stable-unstable manifolds from data IPlist45-8.DAT
original
mapped
org rectangles
0
20
40
60
80
100
120
140
160



Henon s-u manifolds from data IPlist45-8.DAT with the mapped rectangle: number 1


Henon s-u manifolds from data IPlist45-8.DAT with the mapped rectangle: number 35
original
mapped
org rectangles
35 mapped rectangle
35
0
20
40
60
80
100
120
140
160

Henon s-u manifolds from data IPlist45-8.DAT with the mapped rectangle: number 67



s with the following

Galias' Subshift:

$$1 \quad 1 \Bigg) . \qquad (7)$$

th the transition ma-
pological entropy of
).382. This is better
$h) > 0.338$, see [3]).
npts to find complex
The largest bound
30 was obtained for
is close to the non-
y based on the num-
see [2]).

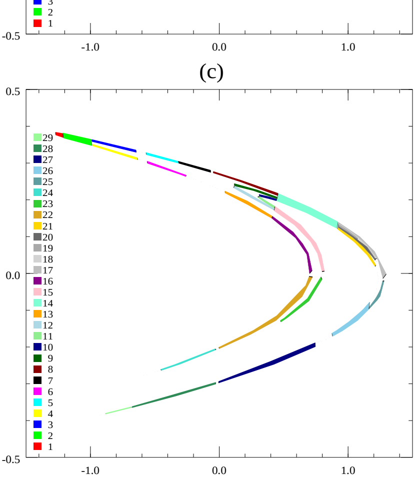


Figure 3: (a) Symbolic dynamics on 8 symbols, initial quadrangles, (b) Symbolic dynamics on 8 symbols, improved quadrangles, (c) Symbolic dynamics on 29 symbols

Figure: Galias Subshift with $h(H) > 0.430$, 29 symbols

Galias-Zgliczynski periodic table:

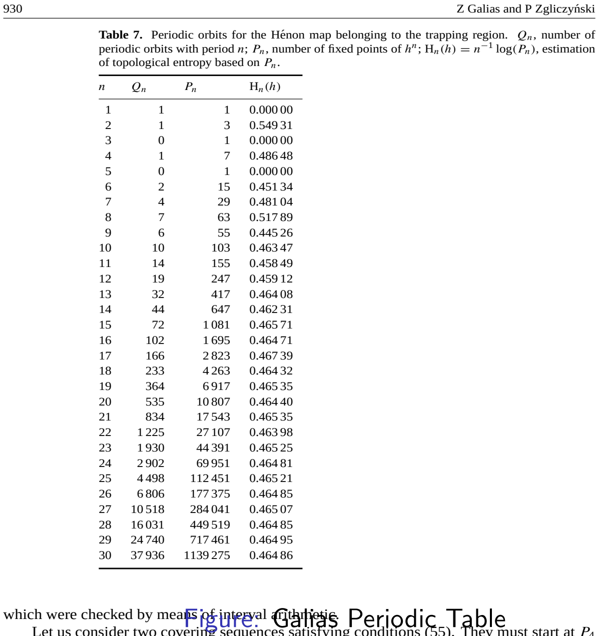


Table 7. Periodic orbits for the Hénon map belonging to the trapping region. Q_n, number of periodic orbits with period n; P_n, number of fixed points of h^n; $\mathrm{H}_n(h) = n^{-1}\log(P_n)$, estimation of topological entropy based on P_n.

n	Q_n	P_n	$\mathrm{H}_n(h)$
1	1	1	0.000 00
2	1	3	0.549 31
3	0	1	0.000 00
4	1	7	0.486 48
5	0	1	0.000 00
6	2	15	0.451 34
7	4	29	0.481 04
8	7	63	0.517 89
9	6	55	0.445 26
10	10	103	0.463 47
11	14	155	0.458 49
12	19	247	0.459 12
13	32	417	0.464 08
14	44	647	0.462 31
15	72	1 081	0.465 71
16	102	1 695	0.464 71
17	166	2 823	0.467 39
18	233	4 263	0.464 32
19	364	6 917	0.465 35
20	535	10 807	0.464 40
21	834	17 543	0.465 35
22	1 225	27 107	0.463 98
23	1 930	44 391	0.465 25
24	2 902	69 951	0.464 81
25	4 498	112 451	0.465 21
26	6 806	177 375	0.464 85
27	10 518	284 041	0.465 07
28	16 031	449 519	0.464 85
29	24 740	717 461	0.464 95
30	37 936	1139 275	0.464 86

which were checked by means of interval arithmetic.

Let us consider two covering sequences satisfying conditions (55). They must start at P_4

Figure: Galias Periodic Table

Entropy Estimates from Trellis Untangling

1. 161 HP's, Pure Rectangles, 66 Symbols, 94 Crossings: 0.4131

Entropy Estimates from Trellis Untangling

1. 161 HP's, Pure Rectangles, 66 Symbols, 94 Crossings: 0.4131
2. 161 HP's, Rect +Hexagons, 77 Symbols, 110 Crossings: 0.4309

Entropy Estimates from Trellis Untangling

1. 161 HP's, Pure Rectangles, 66 Symbols, 94 Crossings: 0.4131
2. 161 HP's, Rect +Hexagons, 77 Symbols, 110 Crossings: 0.4309
3. 267 HP's, Pure Rectangles, 119 Symbols, 185 Crossings: 0.4131
4. 267 HP's, Rect +Hexagons, 130 Symbols, 205 Crossings: 0.4402

Entropy Estimates from Trellis Untangling

1. 161 HP's, Pure Rectangles, 66 Symbols, 94 Crossings: 0.4131
2. 161 HP's, Rect +Hexagons, 77 Symbols, 110 Crossings: 0.4309
3. 267 HP's, Pure Rectangles, 119 Symbols, 185 Crossings: 0.4131
4. 267 HP's, Rect +Hexagons, 130 Symbols, 205 Crossings: 0.4402
5. 437 HP's, Pure Rectangles, 218 Symbols, 346 Crossings: 0.4282
6. 437 HP's, Rect +Hexagons, 229 Symbols, 366 Crossings: 0.4499

Entropy Estimates from Trellis Untangling

1. 161 HP's, Pure Rectangles, 66 Symbols, 94 Crossings: 0.4131
2. 161 HP's, Rect +Hexagons, 77 Symbols, 110 Crossings: 0.4309
3. 267 HP's, Pure Rectangles, 119 Symbols, 185 Crossings: 0.4131
4. 267 HP's, Rect +Hexagons, 130 Symbols, 205 Crossings: 0.4402
5. 437 HP's, Pure Rectangles, 218 Symbols, 346 Crossings: 0.4282
6. 437 HP's, Rect +Hexagons, 229 Symbols, 366 Crossings: 0.4499
7. 707 HP's, Pure Rectangles, 381 Symbols, 603 Crossings: 0.4417
8. 707 HP's, Rect +Hexagons, 392 Symbols, 621 Crossings: 0.4536

Galias' Subshift:

s with the following

$$\begin{matrix} 1 & 1 \end{matrix} \Bigg) . \qquad (7)$$

th the transition ma-
pological entropy of
).382. This is better
$h) > 0.338$, see [3]).
npts to find complex
The largest bound
30 was obtained for
is close to the non-
y based on the num-
see [2]).

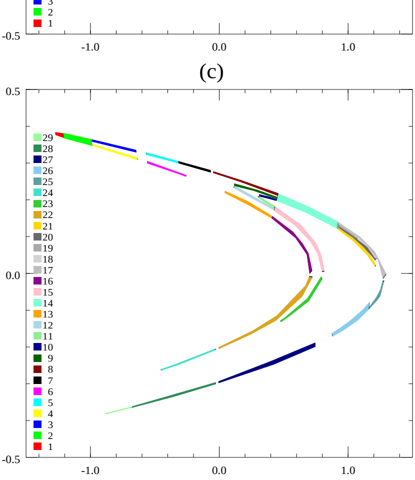
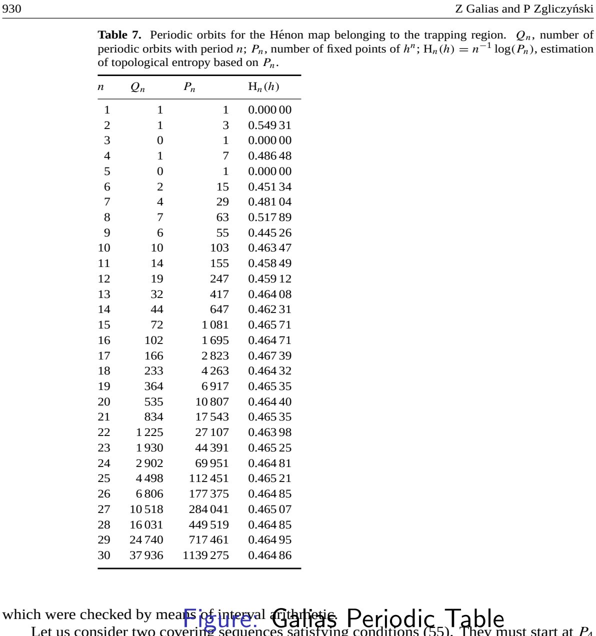


Figure 3: (a) Symbolic dynamics on 8 symbols, initial quadrangles, (b) Symbolic dynamics on 8 symbols, improved quadrangles, (c) Symbolic dynamics on 29 symbols

Figure: Galias Subshift with $h(H) > 0.430$, 29 symbols

Galias-Zgliczynski periodic table:



930 Z Galias and P Zgliczyński

Table 7. Periodic orbits for the Hénon map belonging to the trapping region. Q_n, number of periodic orbits with period n; P_n, number of fixed points of h^n; $\mathrm{H}_n(h) = n^{-1}\log(P_n)$, estimation of topological entropy based on P_n.

n	Q_n	P_n	$\mathrm{H}_n(h)$
1	1	1	0.000 00
2	1	3	0.549 31
3	0	1	0.000 00
4	1	7	0.486 48
5	0	1	0.000 00
6	2	15	0.451 34
7	4	29	0.481 04
8	7	63	0.517 89
9	6	55	0.445 26
10	10	103	0.463 47
11	14	155	0.458 49
12	19	247	0.459 12
13	32	417	0.464 08
14	44	647	0.462 31
15	72	1 081	0.465 71
16	102	1 695	0.464 71
17	166	2 823	0.467 39
18	233	4 263	0.464 32
19	364	6 917	0.465 35
20	535	10 807	0.464 40
21	834	17 543	0.465 35
22	1 225	27 107	0.463 98
23	1 930	44 391	0.465 25
24	2 902	69 951	0.464 81
25	4 498	112 451	0.465 21
26	6 806	177 375	0.464 85
27	10 518	284 041	0.465 07
28	16 031	449 519	0.464 85
29	24 740	717 461	0.464 95
30	37 936	1139 275	0.464 86

which were checked by means of interval arithmetic.

Let us consider two covering sequences satisfying conditions (55). They must start at P_4

Figure: Galias Periodic Table

Entropy Estimates from Trellis Untangling

1. 161 HP's, Pure Rectangles, 66 Symbols, 94 Crossings: 0.4131
2. 161 HP's, Rect +Hexagons, 77 Symbols, 110 Crossings: 0.4309
3. 267 HP's, Pure Rectangles, 119 Symbols, 185 Crossings: 0.4131
4. 267 HP's, Rect +Hexagons, 130 Symbols, 205 Crossings: 0.4402
5. 437 HP's, Pure Rectangles, 218 Symbols, 346 Crossings: 0.4282
6. 437 HP's, Rect +Hexagons, 229 Symbols, 366 Crossings: 0.4499
7. 707 HP's, Pure Rectangles, 381 Symbols, 603 Crossings: 0.4417
8. 707 HP's, Rect +Hexagons, 392 Symbols, 621 Crossings: 0.4536

Outlook

1. Current Computations take a few minutes for HP's, and a few seconds for symbolic dynamics
2. Expect we can go to 100,000 HP's
3. Other Symbolic Dynamics with Taylor Model Symbols

High-Order Constraint Satisfaction

Problem: often need to efficiently satisfy various constraint conditions.
For example, minimize f over box $B \subset R^v$ subject to

$$\begin{gathered} g_i \le 0, \;\; i = 1, ..., c_{\le} \\ h_i = 0, \;\; i = 1, ..., c_{=} \end{gathered}$$

Frequently used approach: subdivide box B into smaller boxes, reject those that can be shown to violate the constraints, and perform minimization only on "active" boxes.

Difficulty: this is expensive - for good results need small boxes.

Goal: "Prune" box B by resolving the constraints with Taylor models; evaluate f over the resulting constraint Taylor model.

As a first step, consider only the surfaces $g_i = 0$, $i = 1, ..., c_{\le}$, and $h_i = 0$, $i = 1, ..., c_{=}$.

Poincare Sections

A very useful tool for the study of long-term dynamics.

- Select a plane, the Poincare section, through which the motion passes repeatedly
- Each time the motion passes trough the plane, project it onto the plane

Determine the time the "center" meets the plane, to floating point error (Newton etc). Then flow has the form

$$x_f = P(x_i, t) + I$$

Problem:

Find the time $t(x_i)$ such that for each x_i, $P(x_i, t(x_i))$ lies on the Poincare section.

Viewed in general terms, this is a **constraint satisfaction problem.**

Outer and Inner Enclosures using Taylor Models

Example: A crescent mapping (A. Goldsztejn and L. Jaulin, 2005)
The function f on $R = \{(x,y)|x^2+y^2 \in [1,2]\}$ given by

$$f(x,y) = \begin{cases} xy \\ x+y \end{cases}$$

Using Taylor models:

- Representing the domain R using Taylor models as

$$(x,y) = (r\cos(\phi), r\sin(\phi)) \text{ where } (r,\phi) \in [1,\sqrt{2}] \times [0,2\pi] = D.$$

Outer and Inner Enclosures using Taylor Models

Example: A crescent mapping (A. Goldsztejn and L. Jaulin, 2005)
The function f on $R = \{(x,y) | x^2 + y^2 \in [1,2]\}$ given by

$$f(x,y) = \begin{cases} xy \\ x+y \end{cases}$$

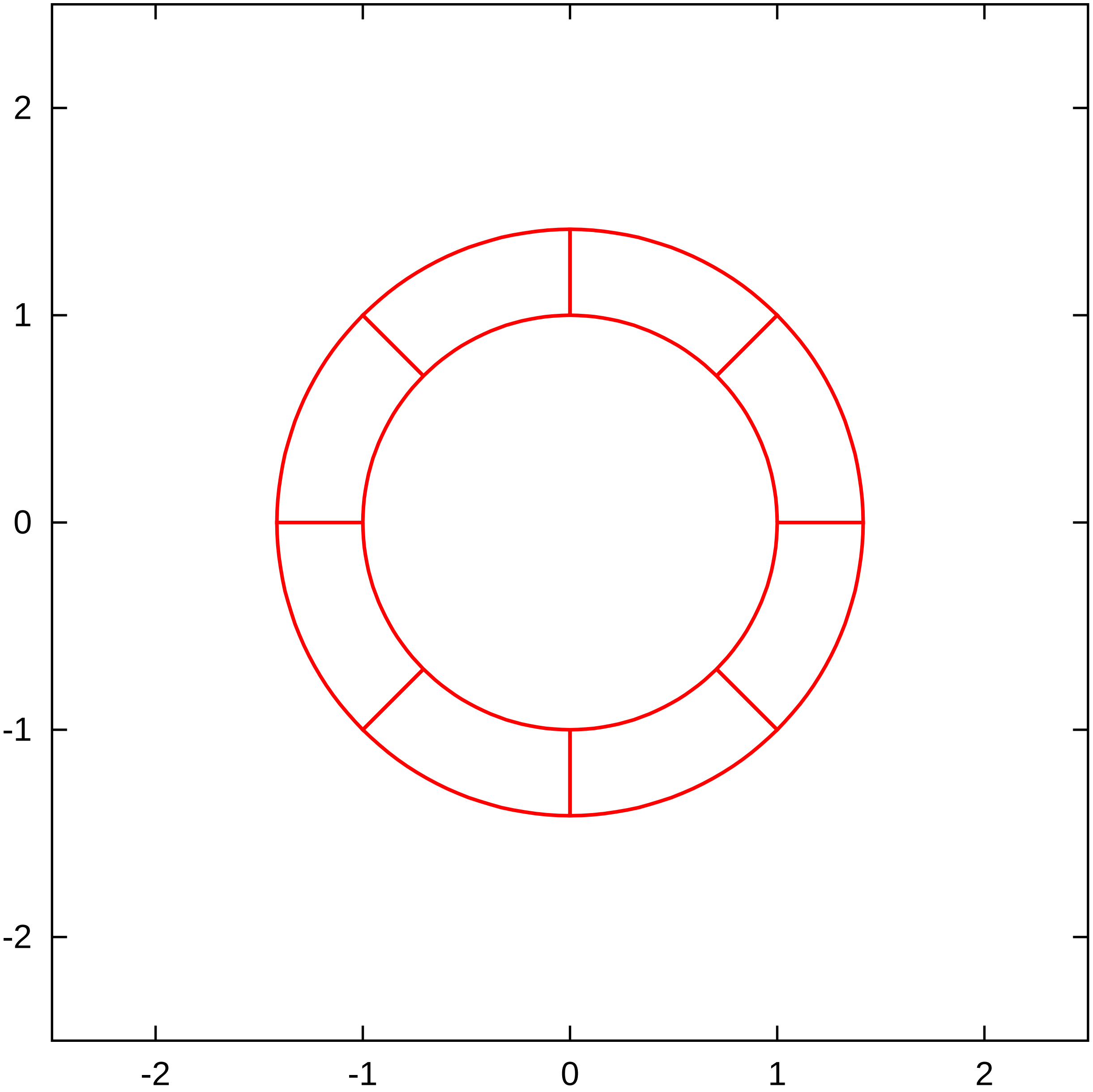
Using Taylor models:

- Representing the domain R using Taylor models as

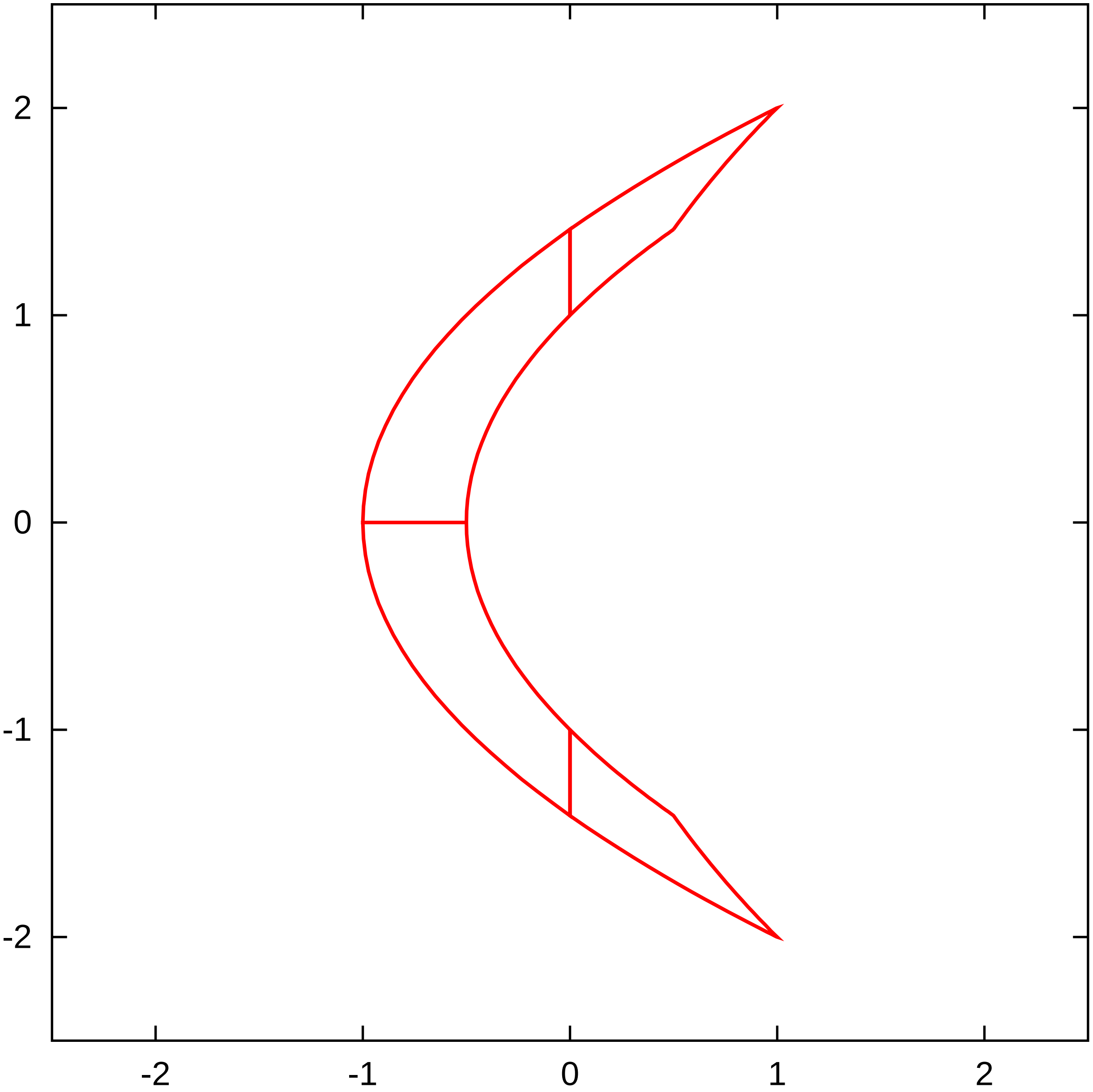
$$(x,y) = (r\cos(\phi), r\sin(\phi)) \text{ where } (r,\phi) \in [1,\sqrt{2}] \times [0,2\pi] = D.$$

- Split D in 8 subdomains to represent f by 8 Taylor models of order 5.
  → width(I) < 0.03; the outer and inner enclosures are indistinguishable.

Region 1=< r^2 =< 2 expressed by TMs (5th order)



fx=x*y, fy=x+y, mapped by TMs in 1=< r^2 =< 2 (5th order)



Outer and Inner Enclosures using Taylor Models

Example: A crescent mapping (A. Goldsztejn and L. Jaulin, 2005)
The function f on $R = \{(x, y) | x^2 + y^2 \in [1, 2]\}$ given by

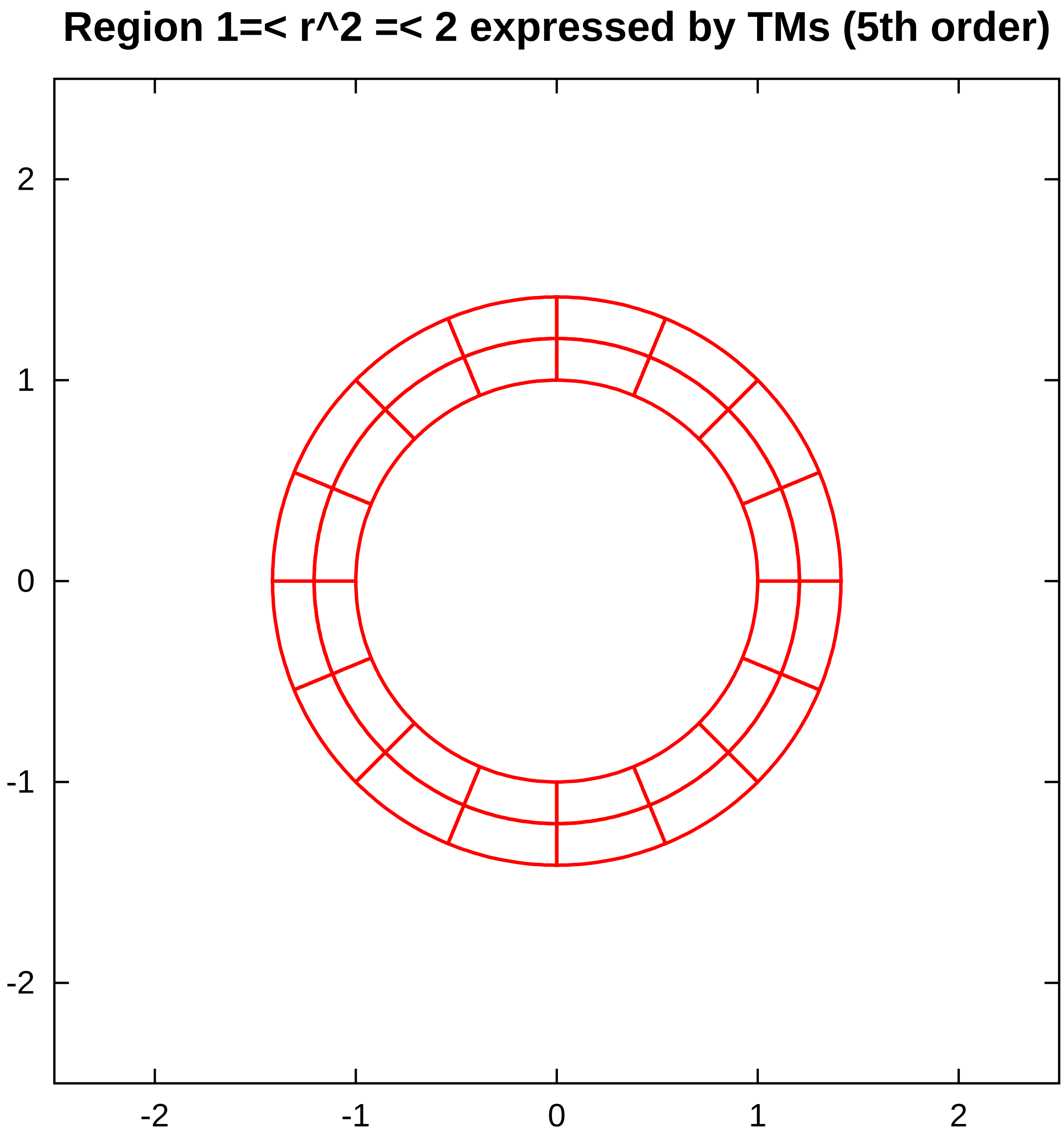
$$f(x, y) = \begin{cases} xy \\ x + y \end{cases}$$

Using Taylor models:

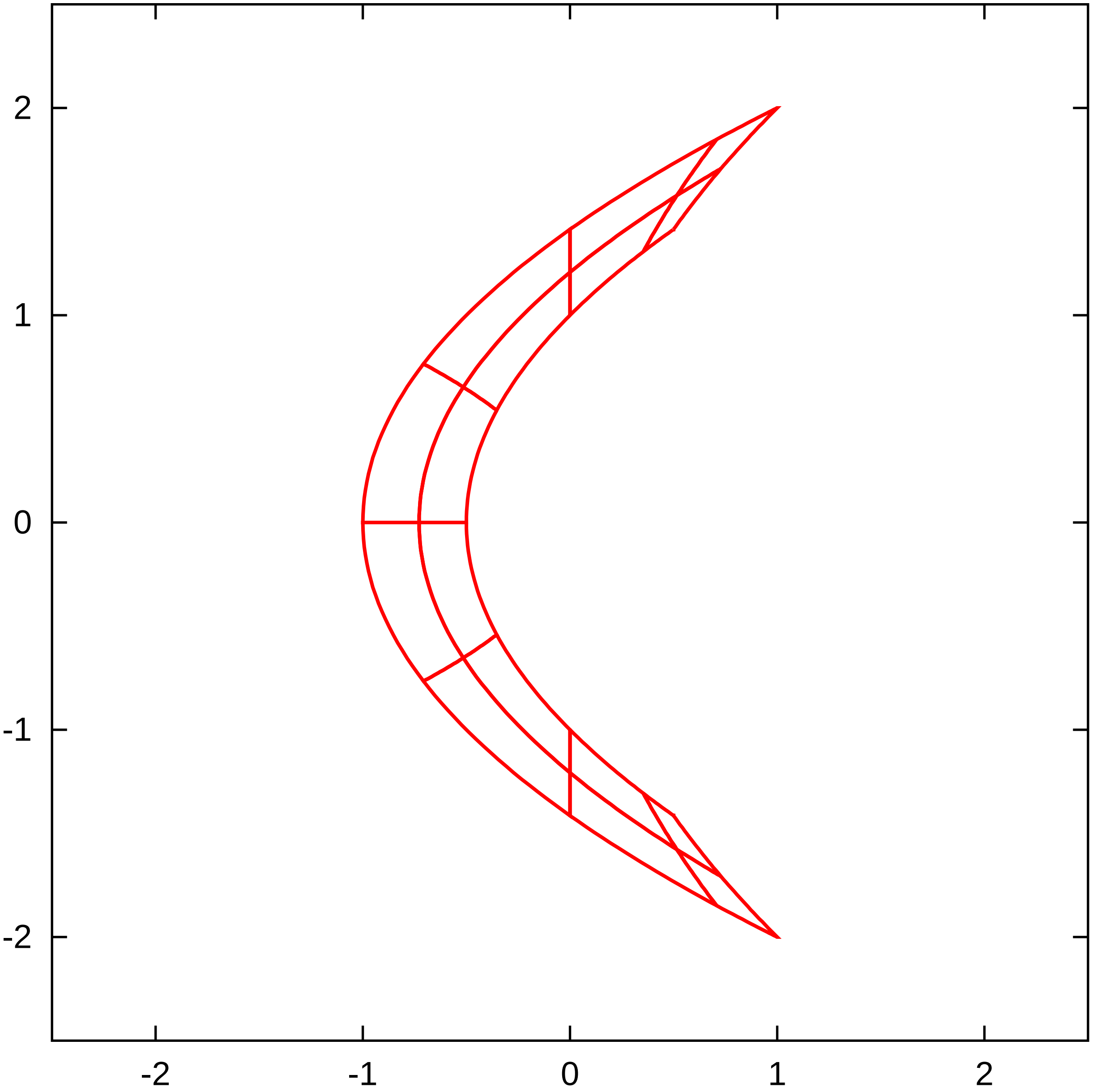
- Representing the domain R using Taylor models as

$$(x, y) = (r\cos(\phi), r\sin(\phi)) \text{ where } (r, \phi) \in [1, \sqrt{2}] \times [0, 2\pi] = D.$$

- Split D in 8 subdomains to represent f by 8 Taylor models of order 5.
  → width(I) < 0.03; the outer and inner enclosures are indistinguishable.
- ... 16 × 2 subdomains, ... by 32 TMs of order 5.


Region 1=< r^2 =< 2 expressed by TMs (5th order)
2
1
0
-1
-2
-2
-1
0
1
2

fx=x*y, fy=x+y, mapped by TMs in 1=< r^2 =< 2 (5th order)



Outer and Inner Enclosures using Taylor Models

Example: A crescent mapping (A. Goldsztejn and L. Jaulin, 2005)
The function f on $R = \{(x,y) | x^2 + y^2 \in [1,2]\}$ given by

$$f(x,y) = \begin{cases} xy \\ x+y \end{cases}$$

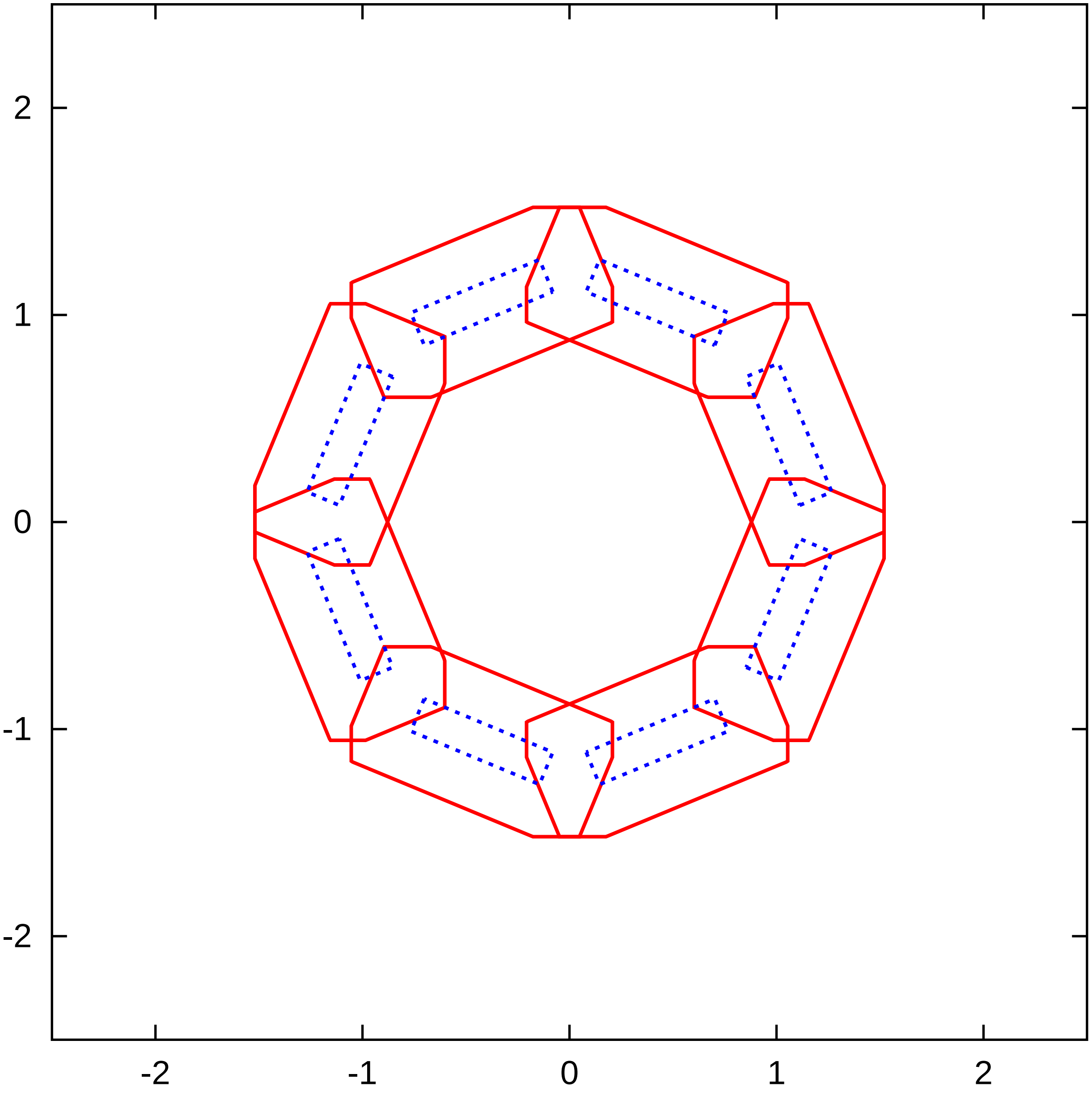
Using Taylor models:

- Representing the domain R using Taylor models as

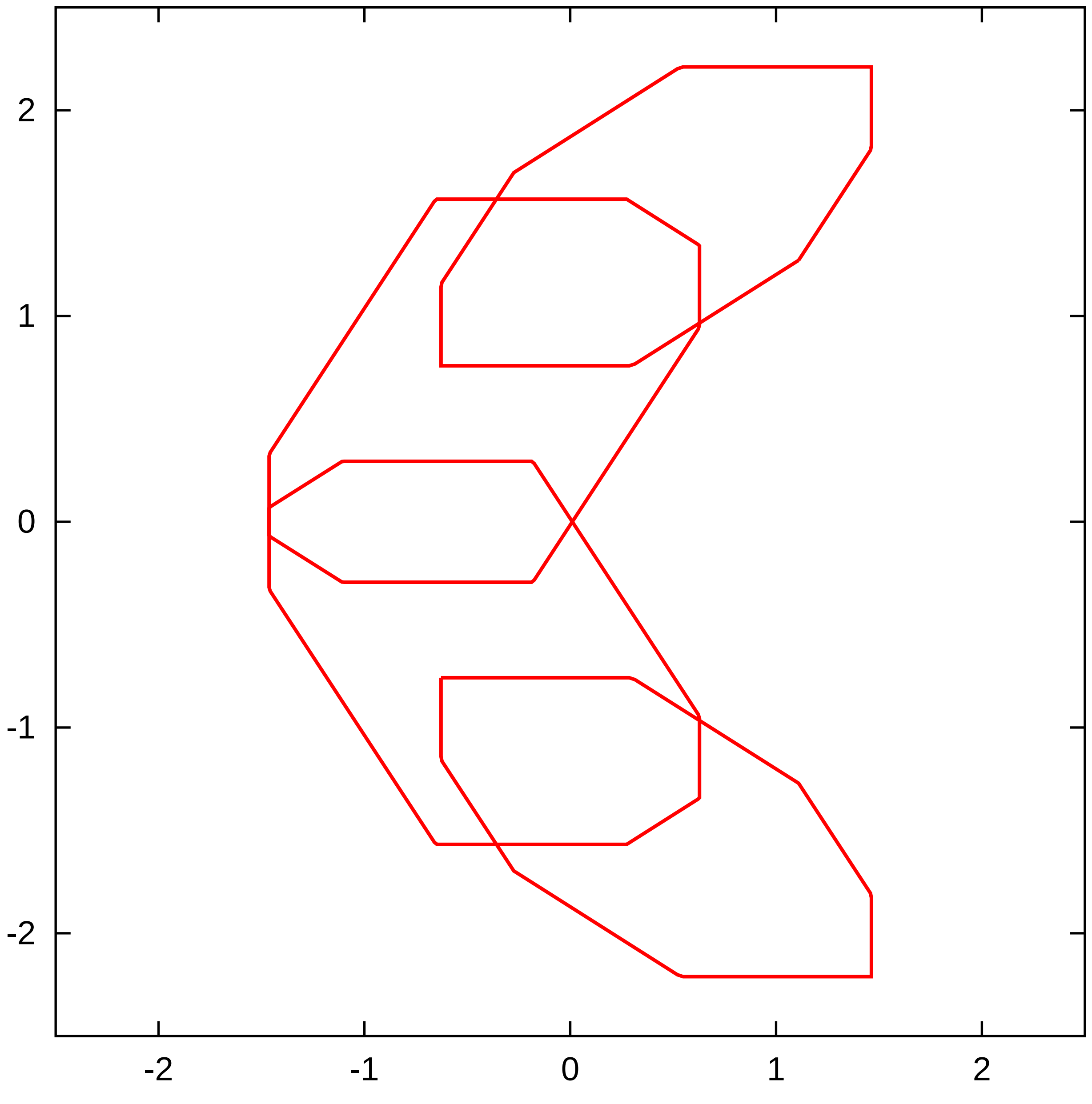
$$(x,y) = (r\cos(\phi), r\sin(\phi)) \text{ where } (r,\phi) \in [1,\sqrt{2}] \times [0,2\pi] = D.$$

- Split D in 8 subdomains to represent f by 8 Taylor models of order 5.
→ width$(I) < 0.03$; the outer and inner enclosures are indistinguishable.
- ... 16×2 subdomains, ... by 32 TMs of order 5.
- ... 8 subdomains, ... by 8 TMs of order 1.
→ width$(I) \simeq 0.9$; the inner representations of f are empty sets.

Region 1=< r^2 =< 2 expressed by TMs (1st order)



fx=x*y, fy=x+y, mapped by TMs in 1=< r^2 =< 2 (1st order)



Outer and Inner Enclosures using Taylor Models

Example: A crescent mapping (A. Goldsztejn and L. Jaulin, 2005)
The function f on $R = \{(x, y)|x^2 + y^2 \in [1, 2]\}$ given by

$$f(x,y) = \begin{cases} xy \\ x+y \end{cases}$$

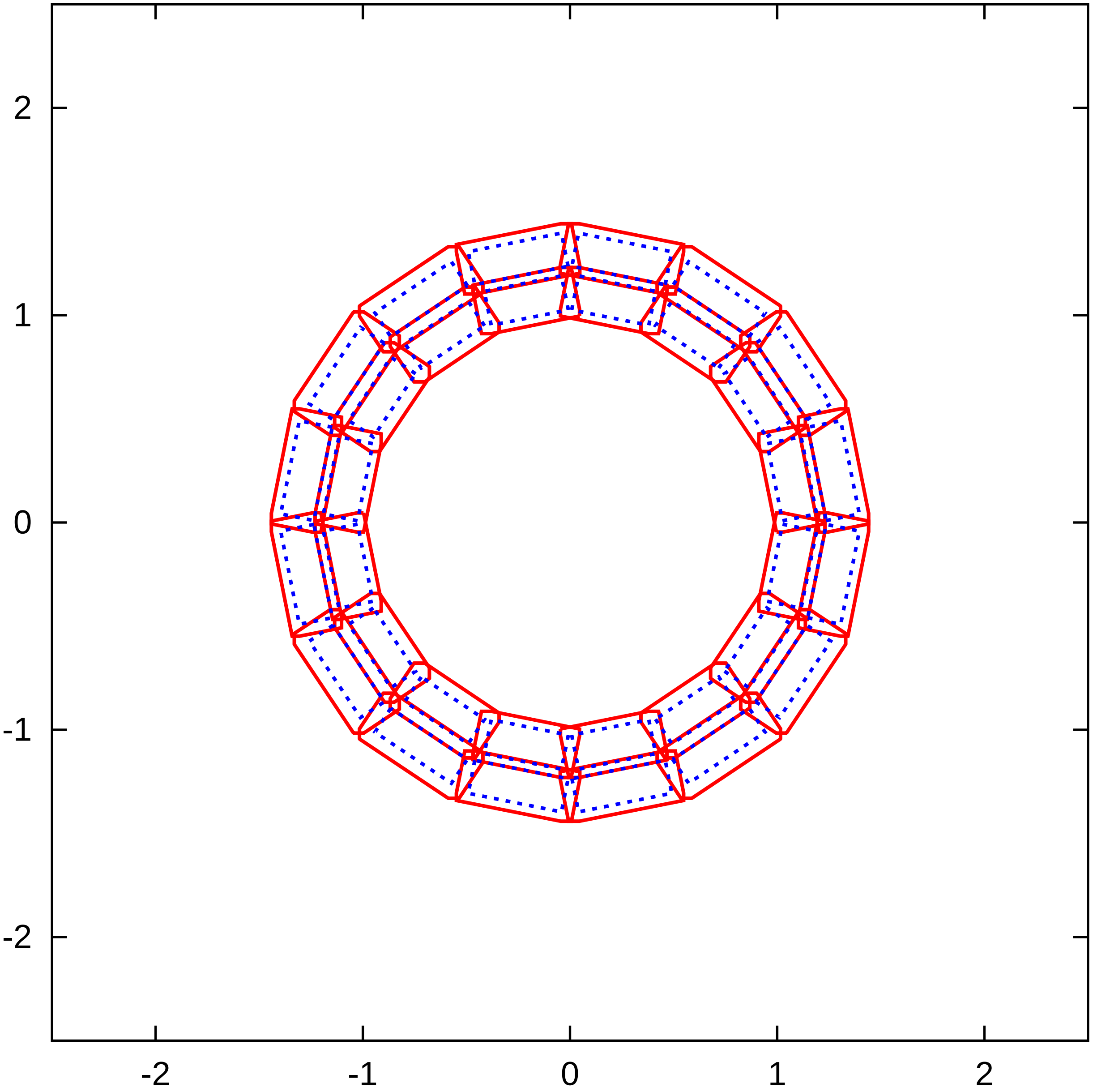
Using Taylor models:

- Representing the domain R using Taylor models as

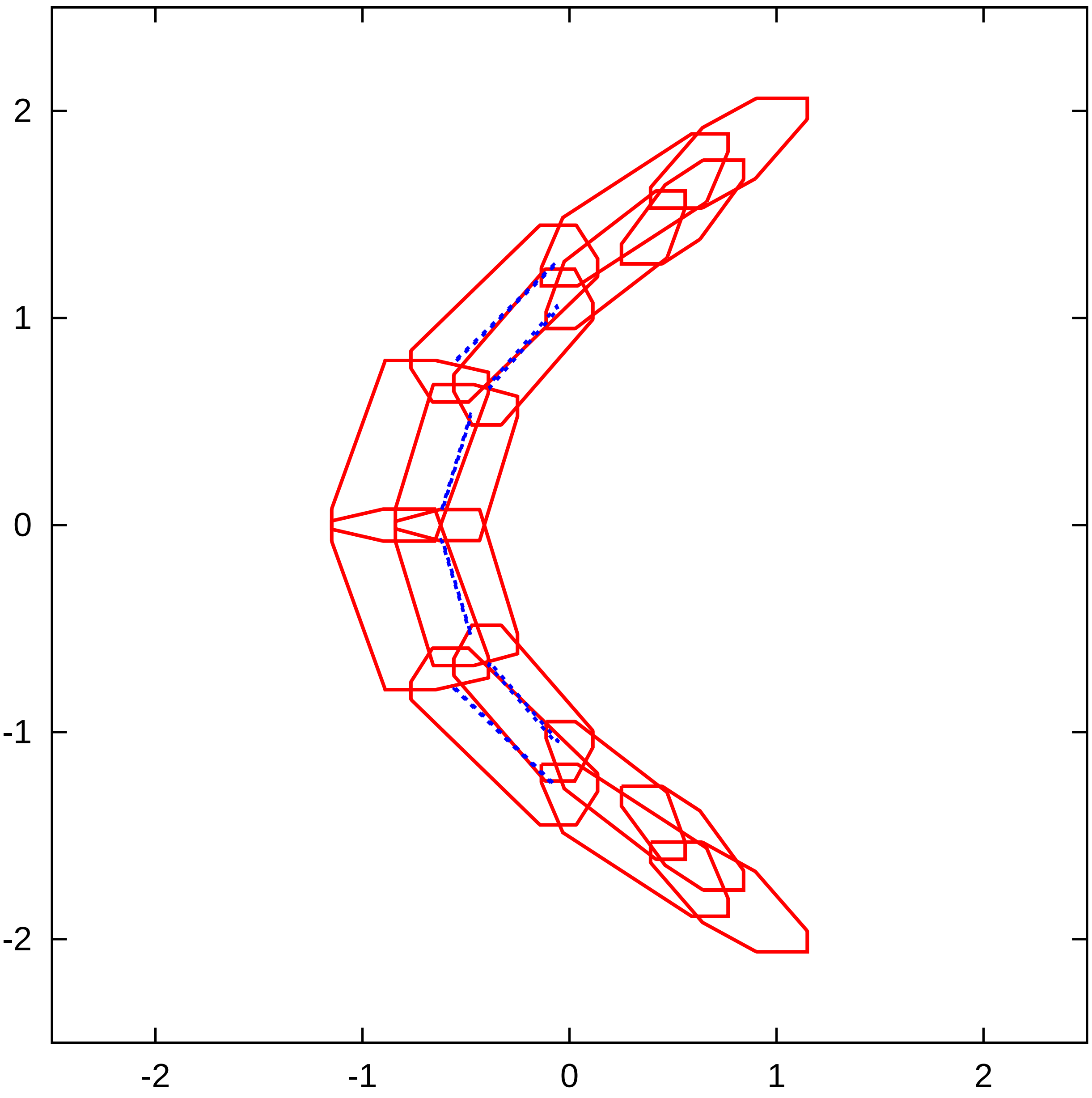
$$(x, y) = (r\cos(\phi), r\sin(\phi)) \text{ where } (r, \phi) \in [1, \sqrt{2}] \times [0, 2\pi] = D.$$

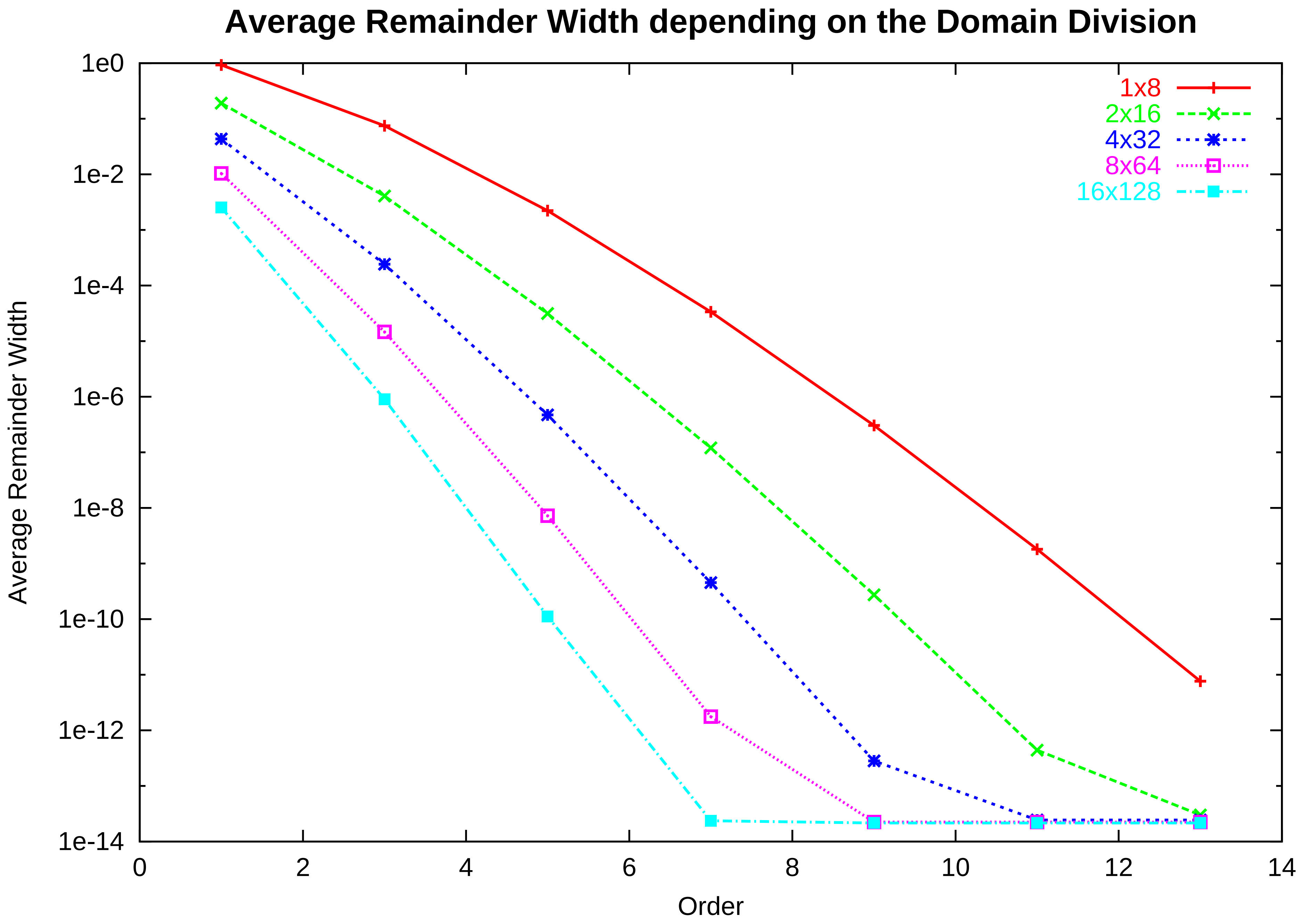
- Split D in 8 subdomains to represent f by 8 Taylor models of order 5.
  → width(I) < 0.03; the outer and inner enclosures are indistinguishable.
- ... 16 × 2 subdomains, ... by 32 TMs of order 5.
- ... 8 subdomains, ... by 8 TMs of order 1.
  → width(I) $\simeq 0.9$; the inner representations of f are empty sets.
- ... 16 × 2 subdomains, ... by 32 TMs of order 1.

Region 1=< r^2 =< 2 expressed by TMs (1st order)



fx=x*y, fy=x+y, mapped by TMs in 1=< r^2 =< 2 (1st order)




Average Remainder Width depending on the Domain Division
1e0
1e-2
1e-4
1e-6
1e-8
1e-10
1e-12
1e-14
Average Remainder Width
0
2
4
6
8
10
12
14
Order
1x8
2x16
4x32
8x64
16x128

High-Order Verification of Exteriority of Points

Goal: Decide whether a point c is outside the range of $f(D)$;

$$c \notin f(D).$$

Scheme:

Have a Taylor model representation $f(x) \in P(x) + I$ for all $x \in D$.
Let $C = P(0)$ and $P^{\geq 1}(x) = P(x) - C$.
Then we want to assert that there is no $x \in D$ with $c - C \in P^{\geq 1}(x) + I$.
Determine the order n inverse $P^{\geq 1*}$ to $P^{\geq 1}$, and apply it to both sides:

$$P^{\geq 1*}(c - C) \in P^{\geq 1*}(P^{\geq 1}(x) + I) \subset \mathcal{I} + I^* \subset D + I^*.$$

So, if it can be shown that

$$P^{\geq 1*}(c - C) \notin D + I^*,$$

then $c \notin f(D)$.

The Taylor Polynomial of the Inverse

In practice it is usually not possible to determine the inverse of M. However, we develop a method that allows the computation of the Taylor polynomial of M^{-1} from the Taylor polynomial of M. We split M into its linear and nonlinear parts

$$M = L + N$$

and note that there is no constant part since the constraint conditions satisfy $f_i = 0$ for $x = 0$ because of the shift of origin. Composing with the inverse map M^{-1}, we have

$$\begin{aligned} M \circ M^{-1} &= I \\ L \circ M^{-1} &= I - N \circ M^{-1} \\ M^{-1} &= L^{-1} \circ (I - N \circ M^{-1}). \end{aligned}$$

The latter relationship now allows to iteratively compute the Taylor polynomial of the inverse. Because note that N is purely nonlinear; so if M^{-1} is known to order k, $N \circ M^{-1}$ is automatically know to order $k+1$. Thus also $L^{-1} \circ (I - N \circ M^{-1})$ is known to order k +1, and hence M^{-1} is obtained to the next higher order. Thus by sufficient iteration one can determine the Taylor polynomial of the inverse to any desired order.

High-Order Verification of Interiority of Points

Goal: Decide whether a point c is inside the range of $f(D)$;

$$c \in f(D).$$

Scheme:

Very similar to the previous scheme for an exteriority test.

If it can be shown that

$$P^{\geq 1*}(c - C) \in D + I^*,$$

then $c \in f(D)$. In this case, however, $P^{\geq 1*}$ has to be shown to be injective (for example by factoring out linear part, and showing that resulting identity plus higher orders has strictly positive slopes).

High-Order Verification of Interiority of Points

Goal: Decide whether a point c is inside the range of $f(D)$;

$$c \in f(D).$$

Scheme 2:

Have a Taylor model representation $f(x) \in P(x)+I$ for all $x \in D$, and let $r(x) = f(x)-P(x)$, where $r(x) \in I$. Let $C = P(0)$ and $P^{\geq 1}(x) = P(x)-C$.

We attempt to assure the existence of a point $s \in D : f(s) = c$, which is equivalent to

$$c - C = P^{\geq 1}(s) + r(s).$$

Determine the order n inverse $P^{\geq 1*}$ to $P^{\geq 1}$. Let

$$s_0 = P^{\geq 1*}(c - C),$$

an approximation for a solution s, and it is likely $s_0 \in D$. If not, move s_0 towards the center of D.

Now we introduce a change of variables $\bar{s} = s - s_0$.

Let $\bar{L}$ denote the linear part of $P^{\geq 1}(s_0 + \bar{s})$ in $\bar{s}$, and let $\bar{L}^*$ be an approximate non-singular inverse of $\bar{L}$. Then the problem is equivalent to

$$\begin{aligned} c - C &= P^{\geq 1}(s_0 + \bar{s}) + r(s_0 + \bar{s}), \text{ or} \\ \mathcal{I}(\bar{s}) &= \bar{L}^* \left[c - C - \left(P^{\geq 1}(s_0 + \bar{s}) + r(s_0 + \bar{s})\right)\right] + \mathcal{I}(\bar{s}). \end{aligned}$$

$$\mathcal{I}(\bar{s}) = \bar{L}^* \left[c - C - (P^{\geq 1}(s_0 + \bar{s}) + r(s_0 + \bar{s}))\right] + \mathcal{I}(\bar{s}).$$

Compute the RHS by TM arithmetic ($r(s_0 + \bar{s})$ by $0 + I$). We obtain a fixed point problem in $\bar{s}$:

$$\mathcal{I}(\bar{s}) = R(\bar{s}) + I_f.$$

Observe that the zeroth order and the linear parts of $R(\bar{s})$ have very small coefficients. By bounding them by an interval I_R,

$$\mathcal{I}(\bar{s}) = R^{\geq 2}(\bar{s}) + I_R + I_f.$$

Now we attempt to find an interval $\bar{S}$:

$$\bar{S} \supset A(\bar{S}) = R^{\geq 2}(\bar{S}) + I_R + I_f.$$

If such $\bar{S}$ is found, $\bar{S}$ contains a fixed point (via Brower FP theorem). Then we have found a solution of the original problem as long as $s_0 + \bar{S} \subseteq D$.

$$c \in f(s_0 + \bar{S}), \text{ thus } c \in f(D).$$

Note: For a small interval $\bar{S}$, $R^{\geq 2}(\bar{S})$ is a much smaller interval. Choosing $\bar{S}$ as a small multiple of the interval $(I_f + I_R)$ will likely lead to an immediate self enclosure, which will scale with order $(n + 1)$ of the domain size.

Properties:

1. Given a TM $P+I$ describing the function f over D, the problem of deciding interiority is transformed to a FP problem. The occurring intervals are comparable in size to the small remainder I. Since only quadratic and higher order terms appear in the FP problem, the method is very likely to succeed as long as I is sufficiently small.
2. The resulting enclosure of the FP scales with the width of I, and hence with order $(n+1)$ of the domain D.
3. The setup of this FP problem requires only limited inexpensive additional TM arithmetic beyond the computation of the TM representing the original function f. In particular, using an intrinsic tool for Horner shifts, no TM multiplications are necessary.
4. Unlike conventional interval-Newton methods, the approach does not require the inversion of any interval matrices, but only floating point approximations to inverses of floating point matrices.

Example: Exteriority and Interiority of Points

Example: A crescent mapping (A. Goldsztejn and L. Jaulin, 2005)
The function f on $R = \{(x, y)|x^2 + y^2 \in [1, 2]\}$ given by

$$f(x, y) = \begin{cases} xy \\ x + y \end{cases}$$

Using the discussed schemes:

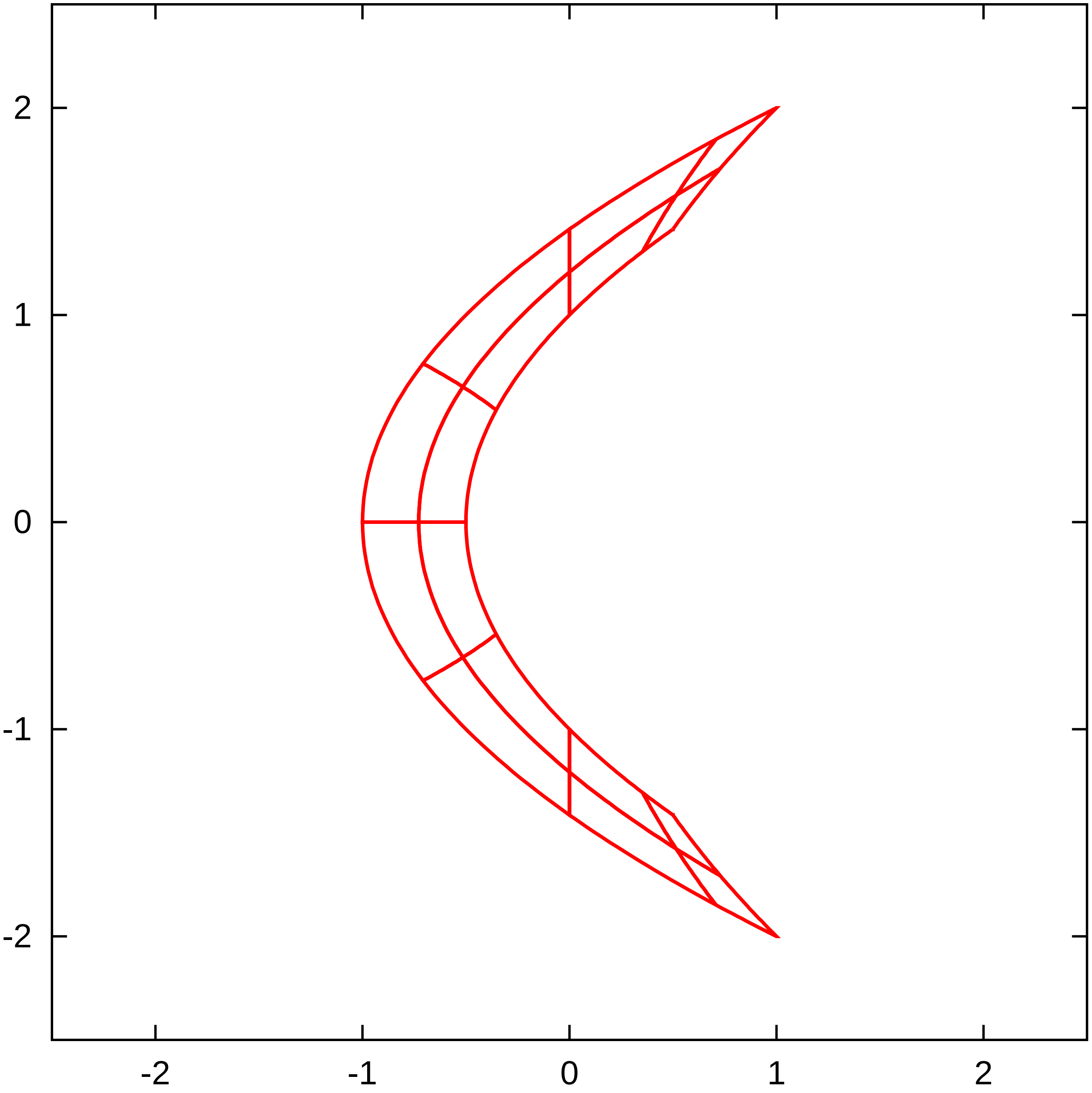
- Study the distance of points verified barely outside and inside the range of the function $f(D)$.

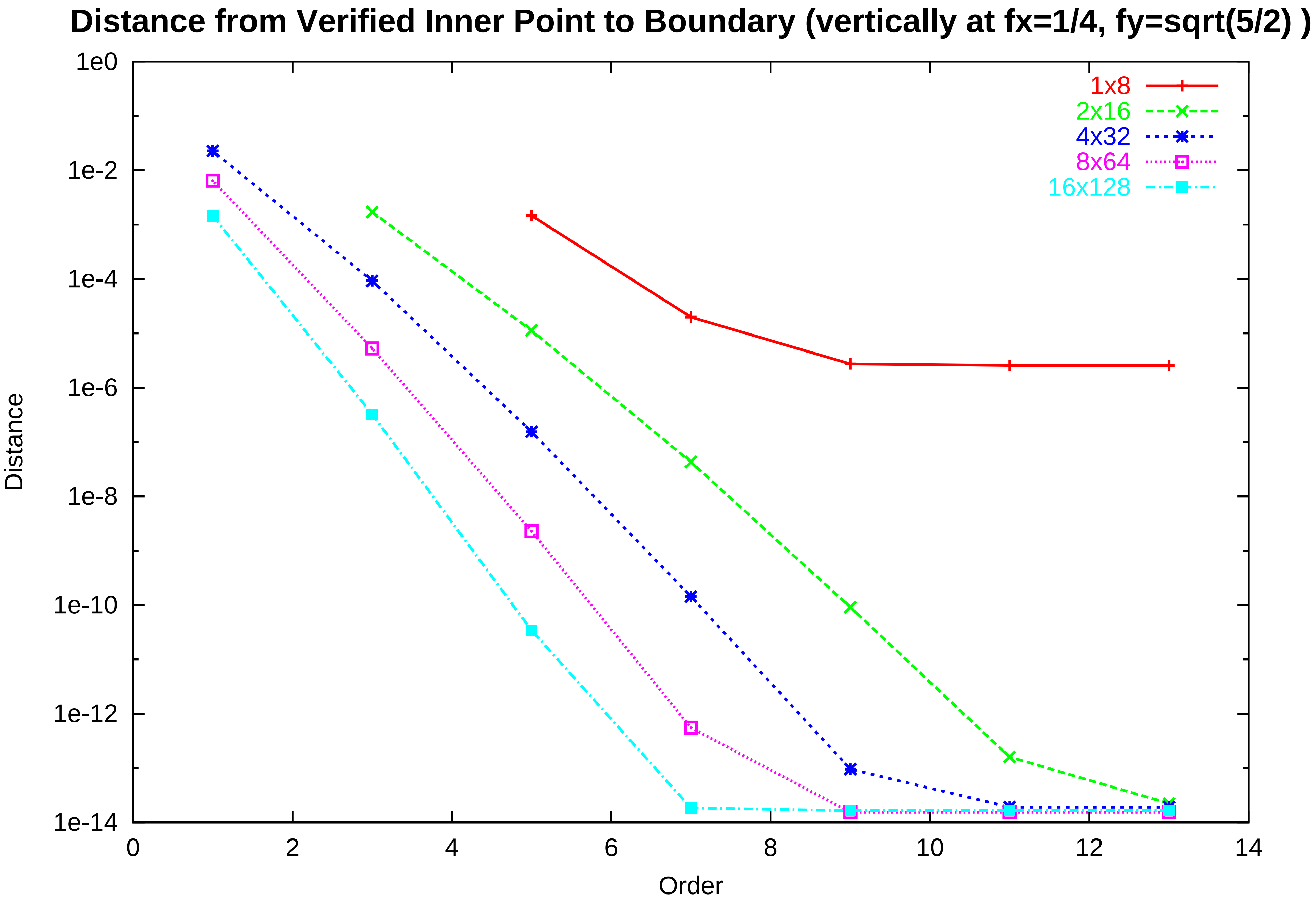
  Measure the distance along the vertical line through the point

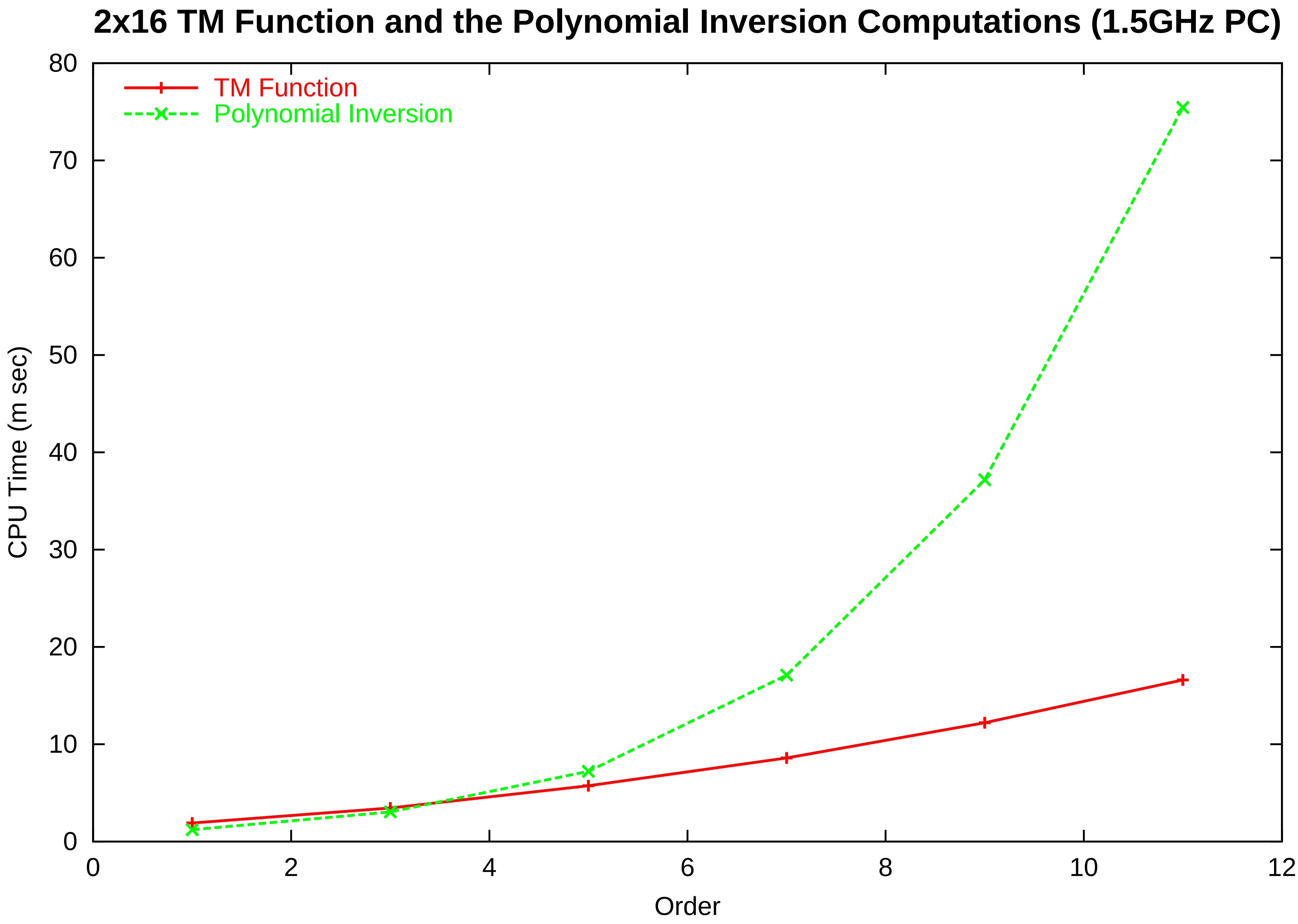
  $$(x, y) = \left(\frac{1}{4}, \sqrt{\frac{5}{2}}\right),$$

  which is known to lie on the boundary.

fx=x*y, fy=x+y, mapped by TMs in 1=< r^2 =< 2 (5th order)




Distance from Verified Inner Point to Boundary (vertically at fx=1/4, fy=sqrt(5/2))
1e0
1e-2
1e-4
1e-6
1e-8
1e-10
1e-12
1e-14
Distance
0
2
4
6
8
10
12
14
Order
1x8
2x16
4x32
8x64
16x128


2x16 TM Function and the Polynomial Inversion Computations (1.5GHz PC)
TM Function
Polynomial Inversion
CPU Time (m sec)
Order
0
10
20
30
40
50
60
70
80
0
2
4
6
8
10
12

Gradient Taylor Models - Motivation

Conventional Taylor Model for f on D: Pair of Taylor polynomial P of f and remainder I such that

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

Many verified tools benefit from enclosures for derivatives of functions; for example **Global Optimization**.

Frequently used: **Automatic Differentiation** with interval arguments yields interval enclosure of derivatives

Similarly, can use Automatic Differentiation **with Taylor Model** arguments to get Taylor model enclosures of derivatives.

But: the Taylor polynomial of $\partial f/\partial x_i$ is just $\partial P/\partial x_i$ (to one order less)

Thus: Knowing P from TM of f, we already know the bulk of the information for the TMs of the gradient (to one order less)!

Idea: Save time by calculating only P, and **not** the Taylor polynomials of the derivative.

The Gradient Taylor Model

Definition: Gradient Taylor model

For a function $f(x)$ on D, we call the vector

$$(P, I, I_1, \dots, I_v)$$

a gradient Taylor model of f, if $\forall x \in D$,

$$\begin{aligned} f(x) &\in P(x) + I \text{ and} \\ \frac{\partial f(x)}{\partial x_i} &\in \frac{\partial P(x)}{\partial x_i} + I_i \text{ for all } i = 1, ..., v \end{aligned}$$

Thus a gradient Taylor model for the function f consists of its Taylor polynomial, a remainder bound, as well as remainder bounds for its partial derivatives based on their Taylor polynomials of one order less.

Addition of Gradient Taylor Models

Assume we know gradient Taylor models for f and g, and thus on D, we have

$$\begin{aligned}(f,\nabla f) &\in \left(P_f + I_f, \frac{\partial P_f}{\partial x_{f1}} + I_{f1}, \ldots, \frac{\partial P_f}{\partial x_{fv}} + I_{fv}\right)\\ (g,\nabla g) &\in \left(P_g + I_g, \frac{\partial P_g}{\partial x_{g1}} + I_{g1}, \ldots, \frac{\partial P_g}{\partial x_{gv}} + I_{gv}\right).\end{aligned}$$

Then we apparently also have

$$\begin{gathered}(f+g,\nabla f+\nabla g) \in \left(P_f + P_g + I_f + I_g, \frac{\partial P_f}{\partial x_{f1}} + \frac{\partial P_g}{\partial x_{g1}} + I_{f1} + I_{g1},\right.\\ \left. ..., \frac{\partial P_f}{\partial x_{fv}} + \frac{\partial P_g}{\partial x_{gv}} + I_{fv} + I_{gv}\right).\end{gathered}$$

and thus $(P_f + P_g, I_f + I_g, I_{f1} + I_{g1}, ..., I_{fv} + I_{gv})$ is a gradient Taylor model for $f + g$.

Multiplication of Gradient Taylor Models

We compute the regular Taylor model part $P_{(f \cdot g)}$ and $I_{(f \cdot g)}$ as we do it for regular Taylor models. For the remainder bounds, there are several ways based on the product rule

$$\frac{\partial}{\partial x_i}(f \cdot g) = \left(\frac{\partial}{\partial x_i} f \right) \cdot g + f \cdot \left(\frac{\partial}{\partial x_i} g \right) . \tag{1}$$

Method 1: Use

$$\frac{\partial f}{\partial x_i} \in \frac{\partial P_f}{\partial x_i} + I_{fi}$$

and perform the computation of (1) in regular Taylor model arithmetic, then get $I_{(f \cdot g)i}$ as the remainder of the arithmetic for (1).

This is simple, but wasteful as it always re-computes the polynomial parts of the derivatives, which is unnecessary as they are merely derivatives of $P_{f \cdot g}$.

Multiplication of Gradient TM - Remainders

Studying the details of Taylor Model multiplication, we see: To obtain the remainder bound of the product, all we need are **bounds for each order** $I_{f,i}^{(j)}, I_{g,i}^{(j)}$ of the factors. Indeed, derivative remainder bounds $I_{(f\cdot g),i}$ are then given by

$$I_{(f\cdot g),i} = \sum_{j=0}^{n} I_{f,i}^{(j)} \cdot I_g^{(\geq n-j+1)} + I_f^{(j)} \cdot I_{g,i}^{(\geq n-j+1)}$$

Method 2a: Use the polynomial derivation operation to calculate each of the gradient polynomials

$$\frac{\partial P_f}{\partial x_i}, \frac{\partial P_g}{\partial x_i}$$

and use the already existing tool for order bounding.

Relatively fast (polynomial derivation is essentially coefficient re-shuffling) and **optimally precise**, given available information.

Multiplication of Gradient TM - Remainders

Method 2b: Observe that

$$I_f^{(j+1)} = \sum_{i_1+\cdots+i_v=j+1} |a_{i_1,\ldots,i_i,\ldots,i_v}| \text{ and we also have}$$

$$I_{f,i}^{(j)} = \sum_{i_1+\cdots+i_v=j+1} |i_i \cdot a_{i_1,\ldots,i_i,\ldots,i_v}| = \sum_{i_1+\cdots+i_v=j+1} i_i \cdot |a_{i_1,\ldots,i_i,\ldots,i_v}|$$

So, the order bounds of the derivative polynomials satisfy

$$I_{f,i}^{(j)} \subset I_f^{(j+1)} \cdot (j+1) \tag{2}$$

Extremely fast (only re-use existing order bounds), but **less precise**.

Intrinsic Functions of Gradient Taylor Models

Problem: given gradient TM of f, want a gradient TM for $int(f)$, where $int \in \{\sin, \cos, \exp, \log\text{,sqrt,...}\}$.

Example: sin. Note

$$\frac{d}{dx_i}\left(\sin(f)\right) = \cos(f) \cdot \frac{d}{dx_i} f$$

1. First factor: perform one additional intrinsic evaluation, compute order bounds.
2. First factor: bulk of effort lies in computing powers of f, which are already available from $\sin(f)$ evaluation!
3. Second factor: Obtain order bounds directly, as above.
4. Determine derivative remainder bound as above from $\sum_{j=0}^{n} I_{f,i}^{(j)} \cdot I_{\cos(f)}^{(\geq n-j+1)}$

Generalizations of Gradient Taylor Models

1. **Extremely Cheap:** do not store $I_1, \ldots, I_v$, but only **a single** I' such that
$$I_1 \subset I', \ldots, I_v \subset I'$$
Particularly useful if combined with method 2b based on $I_{f,i}^{(j)} \subset I_f^{(j+1)} \cdot (j+1)$, since all $I_{f,i}^{(j)}$ will be the same already.

2. **Higher Orders:** Apparently the approach readily generalizes to Hessian Taylor models, or higher derivatives yet.

3. **Cheap Higher Orders:** Can be done economically, i.e. without large numbers of remainder bounds, by using the "extremely cheap" storage way.

Gradient Taylor Models - Summary

- We have shown how to **simultaneously** obtain Taylor models for the **function** and its entire **gradient.**
- There are **various ways** of obtaining remainders, differing in sharpness, speed, and storage
- Even the least sharp way yields good results, since TM remainder bounds are usually so small.
- In all cases, the extra computational **effort is minor**, usually less than 10% beyond the normal cost of a Taylor model.
- Can be generalized to **higher orders.**

High-Order Constraint Satisfaction

Problem: often need to efficiently satisfy various constraint conditions. For example, minimize f over box $B \subset R^v$ subject to

$$g_i \leq 0,\ i = 1, ..., c_{\leq}$$
$$h_i = 0,\ i = 1, ..., c_{=}$$

Frequently used approach: subdivide box B into smaller boxes, reject those that can be shown to violate the constraints, and perform minimization only on "active" boxes.

Difficulty: this is expensive - for good results need small boxes.

Goal: "Prune" box B by resolving the constraints with Taylor models; evaluate f over the resulting constraint Taylor model.

As a first step, consider only the surfaces $g_i = 0$, $i = 1, ..., c_{\leq}$, and $h_i = 0$, $i = 1, ..., c_{=}$.

Poincare Sections

A very useful tool for the study of long-term dynamics.

- Select a plane, the Poincare section, through which the motion passes repeatedly
- Each time the motion passes trough the plane, project it onto the plane

Determine the time the "center" meets the plane, to floating point error (Newton etc). Then flow has the form

$$x_f = P(x_i, t) + I$$

Problem:

Find the time $t(x_i)$ such that for each x_i, $P(x_i, t(x_i))$ lies on the Poincare section.

Viewed in general terms, this is a **constraint satisfaction problem.**

Constraint Satisfaction - General Concept I

Given m smooth constraint functions $f_1, ..., f_m$ in n variables $x_1, ...,\ x_n$ satisfying

$$f_i(x_1, ..., x_n) = 0 \text{ for } i = 1, ..., m.$$

Assume the point $(x_1^{(0)}, ..., x_n^{(0)})$ satisfies the constraints. Wlog $(x_1^{(0)}, ..., x_n^{(0)})$ $= 0$, otherwise shift origin.

Also assume that the first $(n - m)$ variables can be used to parameterize the constraint; can usually be achieved by rearranging the variables. Construct the function M as follows:

$$M = \begin{pmatrix} I_{(n-m)} \\ F \end{pmatrix}.$$

More specifically, the first $(n-m)$ components of M are simply the identity, i.e. $M_i(x) = x_i$ for $i = 1, ..., (n - m)$, and the last m components are the constraint functions $f_1, ..., f_m$.

Constraint Satisfaction - General Concept II

If M is invertible, then M^{-1} satisfies

$$M_i^{-1}(x_1, ..., x_{(n-m)}, 0, ..., 0) = x_i.$$

The first $(n - m)$ rows are merely the identity and are thus uninteresting. However, the lower m rows contain the information on how the m constrained variables x_i depend on the unconstrained variables x_j for $j = 1, ..., (n - m)$, which is what is desired.

Thus, as long as the inversion of M can be carried out, the constraints can be resolved.

The Taylor Polynomial of the Inverse

In practice it is usually not possible to determine the inverse of M. However, we develop a method that allows the computation of the Taylor polynomial of M^{-1} from the Taylor polynomial of M. We split M into its linear and nonlinear parts

$$M = L + N$$

and note that there is no constant part since the constraint conditions satisfy $f_i = 0$ for $x = 0$ because of the shift of origin. Composing with the inverse map M^{-1}, we have

$$\begin{aligned} M \circ M^{-1} &= I \\ L \circ M^{-1} &= I - N \circ M^{-1} \\ M^{-1} &= L^{-1} \circ (I - N \circ M^{-1}). \end{aligned}$$

The latter relationship now allows to iteratively compute the Taylor polynomial of the inverse. Because note that N is purely nonlinear; so if M^{-1} is known to order k, $N \circ M^{-1}$ is automatically know to order $k+1$. Thus also $L^{-1} \circ (I - N \circ M^{-1})$ is known to order k +1, and hence M^{-1} is obtained to the next higher order. Thus by sufficient iteration one can determine the Taylor polynomial of the inverse to any desired order.

A Taylor Model for the Constraints

Previous algorithm allows to compute Taylor polynomial for constraint functions f_i in terms of the parametrizing variables.

Now want to obtain validated remainder bounds. Let P_i be the Taylor polynomial of the dependence of constraint i on the unconstrained variables.

Try to guess values $\varepsilon_{i,l}$ and $\varepsilon_{i,u}$ such that constraint i is violated for all

$$(x_1, ..., x_{n-m}, y_1, ..., y_i > P_i + \varepsilon_{i,u}, ..., y_n)$$
$$(x_1, ..., x_{n-m}, y_1, ..., y_i < P_i - \varepsilon_{i,l}, ..., y_n).$$

Describe ith component by auxiliary variable $\tilde{y}_i \in [-1, 1]$ via

$$y_i = (P_i + \varepsilon_{i,u})(1 + \tilde{y}_i)/2 + y_{i,\max}(1 - \tilde{y}_i)/2 \text{ and}$$
$$y_i = (P_i - \varepsilon_{i,l})(1 + \tilde{y}_i)/2 + y_{i,\min}(1 - \tilde{y}_i)/2, \text{ respectively}$$

Lower bound for educated guess can be obtained from evaluating $f_i(x_1 + [0, 0], ..., P_1(x_i) + [0, 0], ..., P_n(x_i) + [0, 0])$

Example: A Robotic Arm

Two rods in plane, attached at origin, lengths l_1 and l_2. Variables (x_1, x_2, x_3, x_4) :

$$\begin{cases} (x_1, y_1) : \text{ position of elbow} \\ (x_2, y_2) : \text{ location of hand} \end{cases}$$

Constraint conditions:

$$0 = f_1(x_1, x_2, y_1, y_2) = \sqrt{x_1^2 + y_1^2} - l_1$$
$$0 = f_2(x_1, x_2, y_1, y_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} - l_2$$

Consider special case of domain D and lengths l_1 and l_2 as

$$D = [0, 5]^4$$
$$l_1 = 1.0, l_2 = 1.2$$

Example: Box

Let us consider the constraint resolution problem for the box

$$B = (0.2, 0.5, 1.0, 2.1) + [-.1, +.1]^4$$

This box contains feasible points, and can thus not be rejected outright by any method. Try to "prune" the region of the box containing the constraint using Taylor models.

- **First step:** Choice of variables - x_1 and x_2 are parametrizing variables.
- **Second step:** Approximate feasible point for center value. Using standard non-validated tool, we find an approximate solution as

$$y_1^{(0)} = 0.979795897113271$$
$$y_2^{(0)} = 2.141690900975497$$

Example: Approximate Polynomial P_1

- **Third step:** Find approximate polynomial. Using algorithm outlined above, up to fourth order, we find $P_1(x_1, x_2)$ for y_1 as

```
I  COEFFICIENT            ORDER EXPONENTS
1   0.979795897113271       0   0 0  0 0
2  -.2041241452319316       1   1 0  0 0
3  -.5315732948748217       2   2 0  0 0
4  -.1107444364322545       3   3 0  0 0
5  -.1672702425278845       4   4 0  0 0
----------------------------------------
```

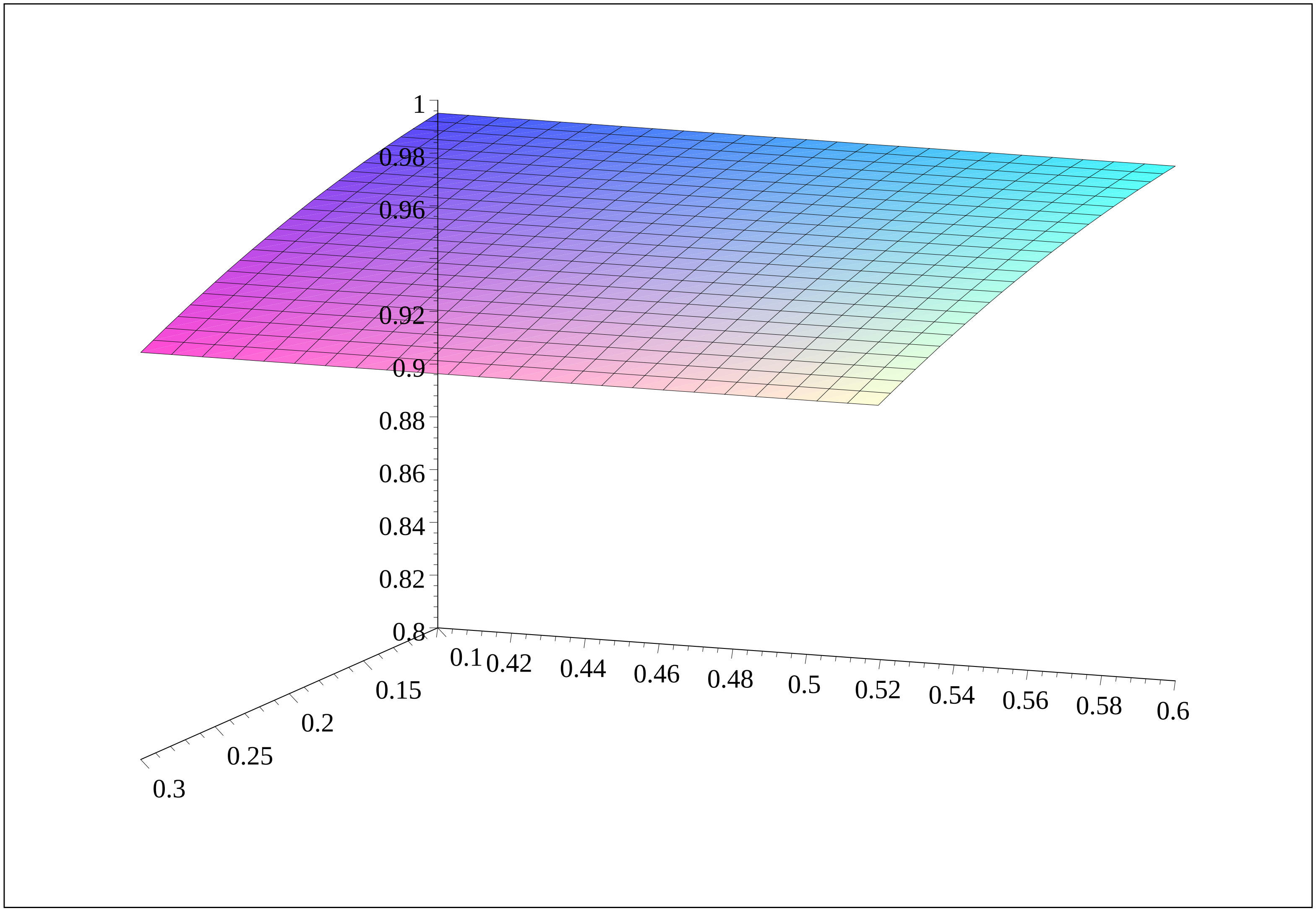
Note that because of the specific form of the constraint, y_1 depends only on x_1, and not on x_2.

Example: Approximate Polynomial P_2

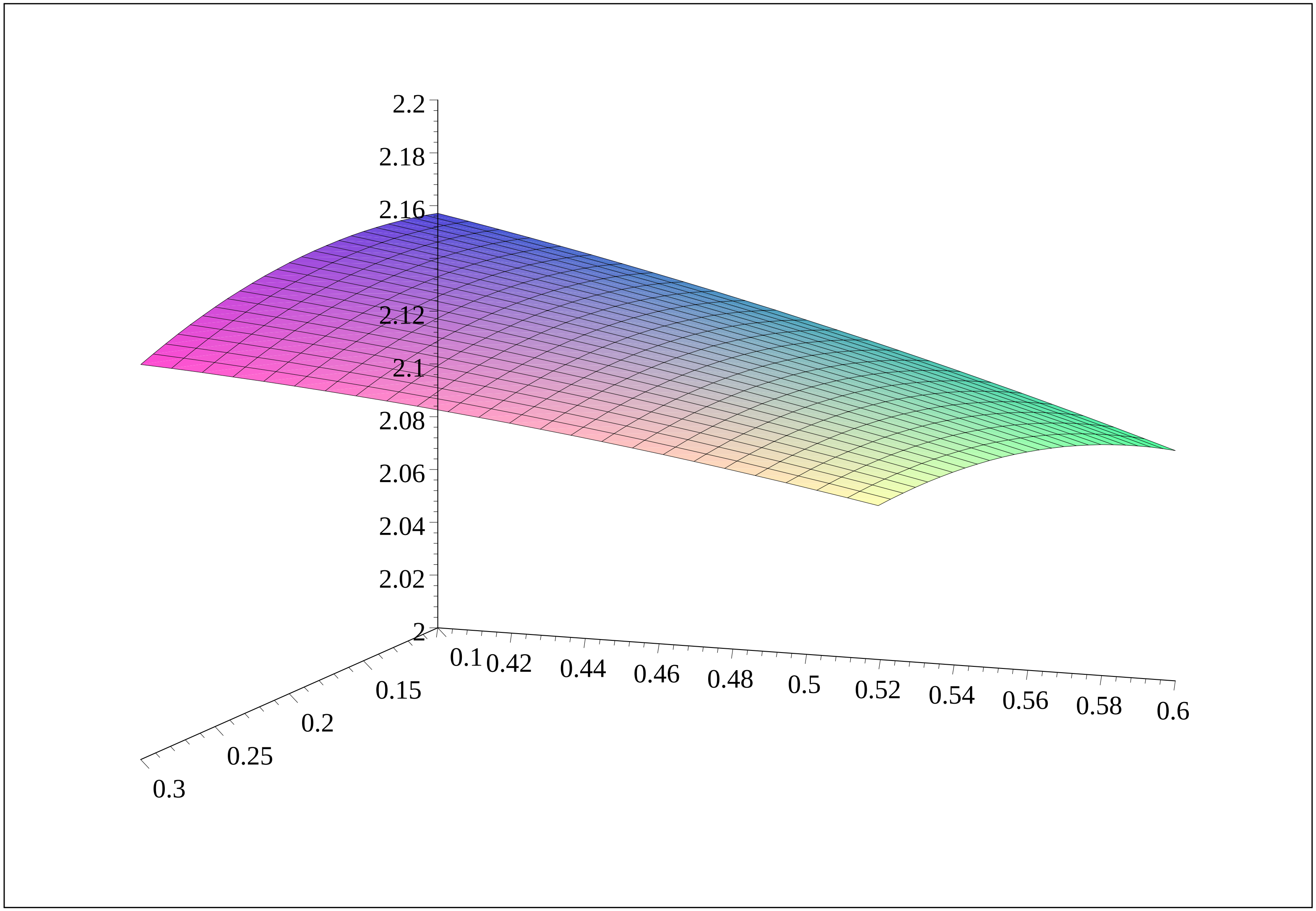
Similarly, we find $P_2(x_1, x_2)$ as

```
  I  COEFFICIENT            ORDER EXPONENTS
  1  2.141690900975497        0   0 0  0 0
  2  0.5407474451522962E-01   1   1 0  0 0
  3  -.2581988897471612       1   0 1  0 0
  4  -.9905935433142194       2   2 0  0 0
  5  0.9180404968787954       2   1 1  0 0
  6  -.4590202484393976       2   0 2  0 0
  7  -.8739936779055102E-02   3   3 0  0 0
  8  -.3060134989595984       3   2 1  0 0
  9  0.3060134989595984       3   1 2  0 0
 10  -.1020044996531995       3   0 3  0 0
 11  -.2806085754758839       4   4 0  0 0
 12  0.4533533317919972       4   3 1  0 0
 13  -.6800299976879959       4   2 2  0 0
 14  0.4533533317919973       4   1 3  0 0
 15  -.1133383329479993       4   0 4  0 0
```

Example: Approximate Polynomial P_1



Example: Approximate Polynomial P_2



Example: Checking Polynomial Part

To check whether indeed the polyomials $P_1(x_1, x_2)$ and $P_2(x_1, x_2)$ resolve the constraints up to order 4, we insert their Taylor models into the contraint conditions. Determine the resulting Taylor model. Their polynomial parts should be zero. We indeed obtain for constraint 1:

```
I  COEFFICIENT            ORDER EXPONENTS
1  -.1110223024625157E-15   1   1 0  0 0
2  0.1387778780781446E-15   3   3 0  0 0
----------------------------------------
```

Note that terms below the cutoff value of 10^{-16} are not shown. Similarly we obtain for constraint 2:

Example: Checking Polynomial Part

```
 I  COEFFICIENT            ORDER EXPONENTS
 1  0.666133814775094E-15   0    0 0  0 0
 2  0.333066907387547E-15   1    1 0  0 0
 3  -.333066907387547E-15   1    0 1  0 0
 4  -.666133814775094E-15   2    2 0  0 0
 5  0.133226762955019E-14   2    1 1  0 0
 6  -.444089209850063E-15   2    0 2  0 0
 7  -.194289029309402E-15   3    3 0  0 0
 8  0.166533453693773E-15   3    2 1  0 0
 9  -.166533453693773E-15   3    1 2  0 0
10  0.555111512312578E-16   3    0 3  0 0
11  0.111022302462516E-15   4    4 0  0 0
12  -.144328993201270E-14   4    3 1  0 0
13  0.166533453693773E-14   4    2 2  0 0
14  -.888178419700125E-15   4    1 3  0 0
15  0.194289029309402E-15   4    0 4  0 0
-----------------------------------------
```

Indeed constraint conditions have been resolved.

Example: Remainder Interval Construction

As the last step, we we now try to find small correction values

$$\varepsilon_{1,l}, \varepsilon_{1,u},$$
$$\varepsilon_{2,l}, \varepsilon_{2,u}$$

such that all values of (x_1, x_2, y_1, y_2) that satisfy the constraint conditions also satisfy

$$y_1 \in P_1(x_1, x_2) + [\varepsilon_{1,l}, \varepsilon_{1,u}]$$
$$y_2 \in P_2(x_1, x_2) + [\varepsilon_{2,l}, \varepsilon_{2,u}]$$

For the guess of possible values, we let ourselves be guided by the widths of the remainder bounds obtained when evaluating the constraint conditons on the Taylor model $(x_1, x_2, P_1(x_1, x_2) + [0, 0], P_2(x_1, x_2) + [0, 0])$. Those values are about $8 \cdot 10^{-5}$. So we attempt

$$\varepsilon_{1,l} = -2 \cdot 10^{-4},\ \varepsilon_{1,u} = 2 \cdot 10^{-4}$$
$$\varepsilon_{2,l} = -2 \cdot 10^{-4},\ \varepsilon_{2,u} = 2 \cdot 10^{-4}$$

Note that using higher order Taylor models, even tighter bounds can be obtained.

Example: Remainder Interval Validation

We do this by showing that the outside of these sets violate some of the constraints. To verify that it is necessary to have $y_1 \in P_1(x_1, x_2) + [\varepsilon_{1,l}, \varepsilon_{1,u}]$, we evaluate constraint 1 with the Taylor model

$$\left(\begin{array}{c} 0.2 + 0.1 \cdot x_1, 0.5 + 0.1 \cdot x_2, \\ 0.9 \cdot (1 - \tilde{y}_1)/2 + (P_1(x_1, x_2) - \varepsilon_{1,l}) \cdot (1 + \tilde{y}_1)/2, 2.1 + 0.1 \cdot y_2 \end{array} \right)$$

with domain box $[-1, 1]^4$. Over this box, the range of this Taylor model contains all those parts of original domain of interest that satisfy $y_1 < P_1(x_1, x_2) + [\varepsilon_{1,l}, \varepsilon_{1,u}]$.

Numerical evaluation shows that for this seed Taylor model, the upper bound of constraint function 1 is always negative, and so constraint 1 is always violated. Indeed, the polynomial part is very close to vanishing altogether because the polynomial P_1 resolves the constraint, and bounding is simple.

Likewise, we use the seed

$$\left(\begin{array}{c} 0.2 + 0.1 \cdot x_1, 0.5 + 0.1 \cdot x_2, \\ 2.2 \cdot (1 + \tilde{y}_1)/2 + (P_1(x_1, x_2) + \varepsilon_{1,u}) \cdot (1 - \tilde{y}_1)/2, 2.1 + 0.1 \cdot y_2 \end{array} \right)$$

over $[-1, 1]^4$ and obtain that constraint function 1 is always positive.

Example: Constraint Enclosure Theorem

Altogether, we have shown:

Theorem: Over the box

$$B = (0.2, 0.5, 1.0, 2.1) + [-.1, +.1]^4$$

only points (x_1, x_2, y_1, y_2) that satisfy

$$\begin{aligned} y_1 &\in P_1 + [-2 \cdot 10^{-4}, 2 \cdot 10^{-4}] \\ y_2 &\in P_2 + [-2 \cdot 10^{-4}, 2 \cdot 10^{-4}] \end{aligned}$$

are compatible with the constraint conditions.

For subsequent minimization of an objective function g, it is sufficient to evaluate g with the seed Taylor model

$$\left(\begin{array}{c} 0.2 + 0.1 \cdot x_1, 0.5 + 0.1 \cdot x_2, \\ P_1(x_1, x_2) + [\varepsilon_{1,l}, \varepsilon_{1,u}], P_1(x_1, x_2) + [\varepsilon_{1,l}, \varepsilon_{1,u}] \end{array} \right).$$

Thus, for Taylor model-based global optimization, the constraint has been resolved.