Fourth International Workshop on Taylor Methods

Boca Raton, Florida
December 16-19, 2006

http://bt.pa.msu.edu/TM/BocaRaton2006/

Topics:
High-Order Methods
Verification & Taylor Models
Automatic Differentiation
Differential Algebraic Tools

and their use for:
ODE and PDE Solvers
Global Optimization
Constraint Satisfaction
Dynamical Systems
Beam Physics

Support: Department of Energy
Michigan State University

Recent Advances in Taylor Model-based ODE Integration

Martin Berz and Kyoko Makino

Michigan State University

Outline

1. The Reference Trajectory
2. The Flow Operator
3. Defect-Based Verification
4. Step Size Control
5. Example: The Double Pendulum
6. Example / Outlook: Manifolds of the Henon Map

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

Differential Algebras on Taylor Polynomial Spaces

Consider space ${}_nD_v$ of Taylor polynomials in v variables and order n with truncation multiplication. Formally: introduce **equivalence relation** on space of smooth functions

$$f =_n g$$

if all derivatives from 0 to n agree at 0. **Class** of f is denoted $[f]$. This induces addition, multiplication and scalar multiplication on classes. The resulting structure forms an algebra.

An algebra is a **Differential Algebra** if there is an operation ∂, called a derivation, that satisfies

$$\partial(s \cdot a + t \cdot b) = s \cdot \partial a + t \cdot \partial b \text{ and}$$
$$\partial(a \cdot b) = a \cdot (\partial b) + (\partial a) \cdot b$$

for any vectors a and b and scalars s and t. **Unfortunately**, the **natural partial derivative** operations $[f] \to [\partial_i f]$ does **not** introduce a differential algebra, because of loss of highest order.

Differential Algebras on Taylor Polynomial Spaces

However, consider the modified operation

$$\partial_f \text{ with } \partial_f g = f \cdot \nabla g$$

If f is origin preserving, i.e. $f(0) = 0$, then ∂_f is a derivation on the space ${}_nD_v$. Why?

- Each derivative operation in the gradient ∇g looses the highest order;
- but since $f(0) = 0$, the missing order in ∇g **does not matter** since it does not contribute to the product $f \cdot \nabla g$.

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

Error Analysis via Interval Defect

Third step of rigorous method: provide rigorous error estimate. We now try to introduce a set of variables $\tilde{e}$, the error variables, such that the flow rigorously satisfies

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

ODE for $\tilde{e}(t)$:

$$\tilde{e}' = f(c(t) + A \cdot \tilde{r}(t) + \tilde{e}) - c'(t) - A \cdot \tilde{r}'(t)$$

Now again evaluating ODE for $\tilde{e}'$ in Taylor arithmetic. Obtain a Taylor expansion of the ODE:

$$\tilde{e}' = 0$$

up to order n in time and k in initial conditions(!)

Of course this is not the real ODE: we are missing the remainder errors. However, evaluating the ODE for $\tilde{e}'$ in **Taylor Model** arithmetic, we obtain a (very small) interval term R, the Taylor model remainder, such that

$$\tilde{e}' \in R$$

Error Analysis via Defect - Implementation

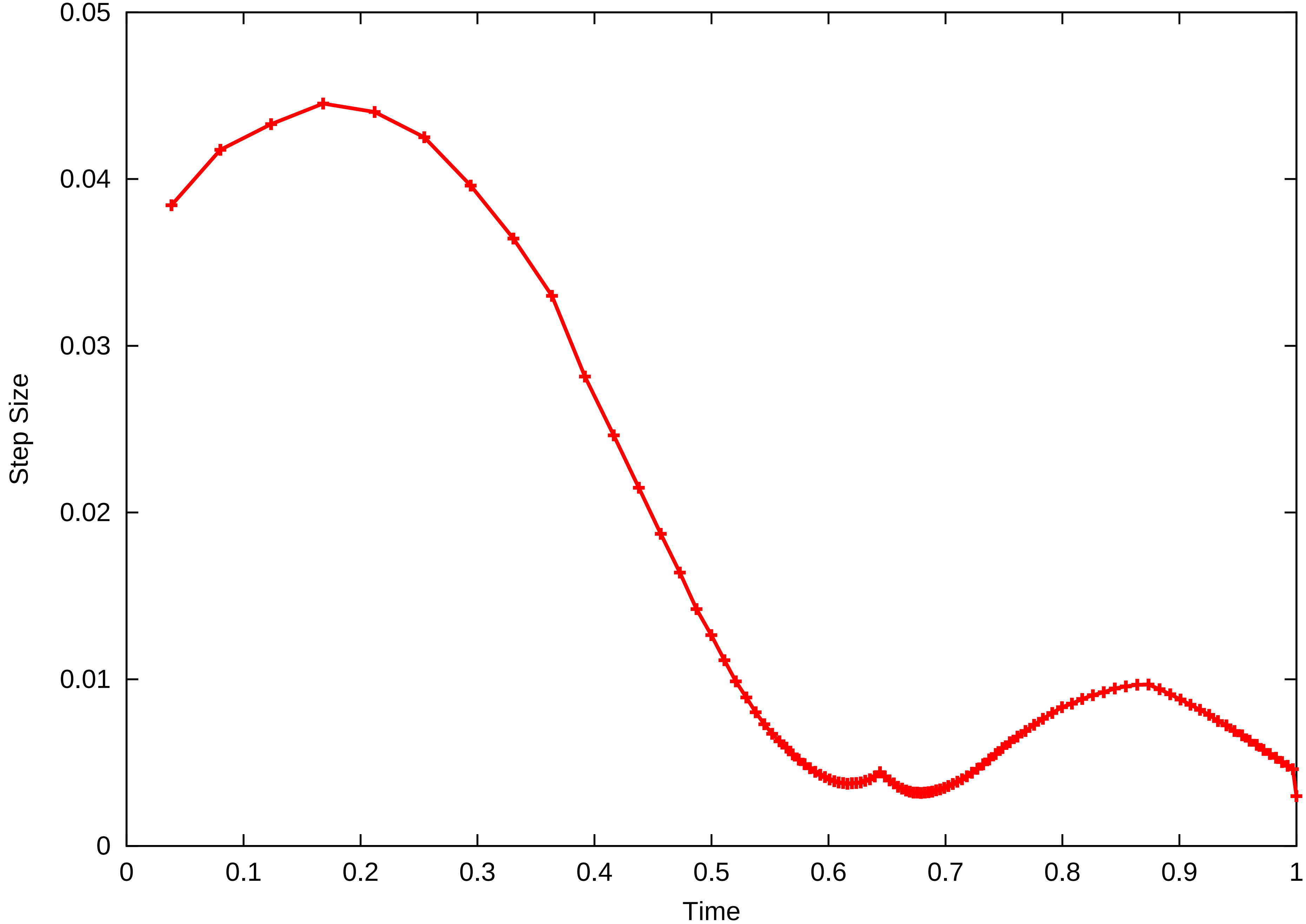
For practical implementation, the following aspects are critical:

1. Make sure $\tilde{r}'(t)$ numerically fits with $\tilde{r}(t)$. Solution: obtain approximate value of $\tilde{r}'(t)$, and then obtain a Taylor model for $\int \tilde{r}'(t)$ to represent $\tilde{r}$. Can be done
2. Defect ODE can be solved with very simple Euler-type integrator.
3. Simplest possible case: treat $\tilde{e}$ as intervals. Leads to a cone-type flow enclosure.
4. Next more sophisticated case: treat $\tilde{e}$ as additional variables (to very low order). Leads to linear inhomogeneous ODE.

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 $\Delta t_3 = c \cdot \Delta t_2$



Step Size
0.05
0.04
0.03
0.02
0.01
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time

Dynamic Domain Decomposition

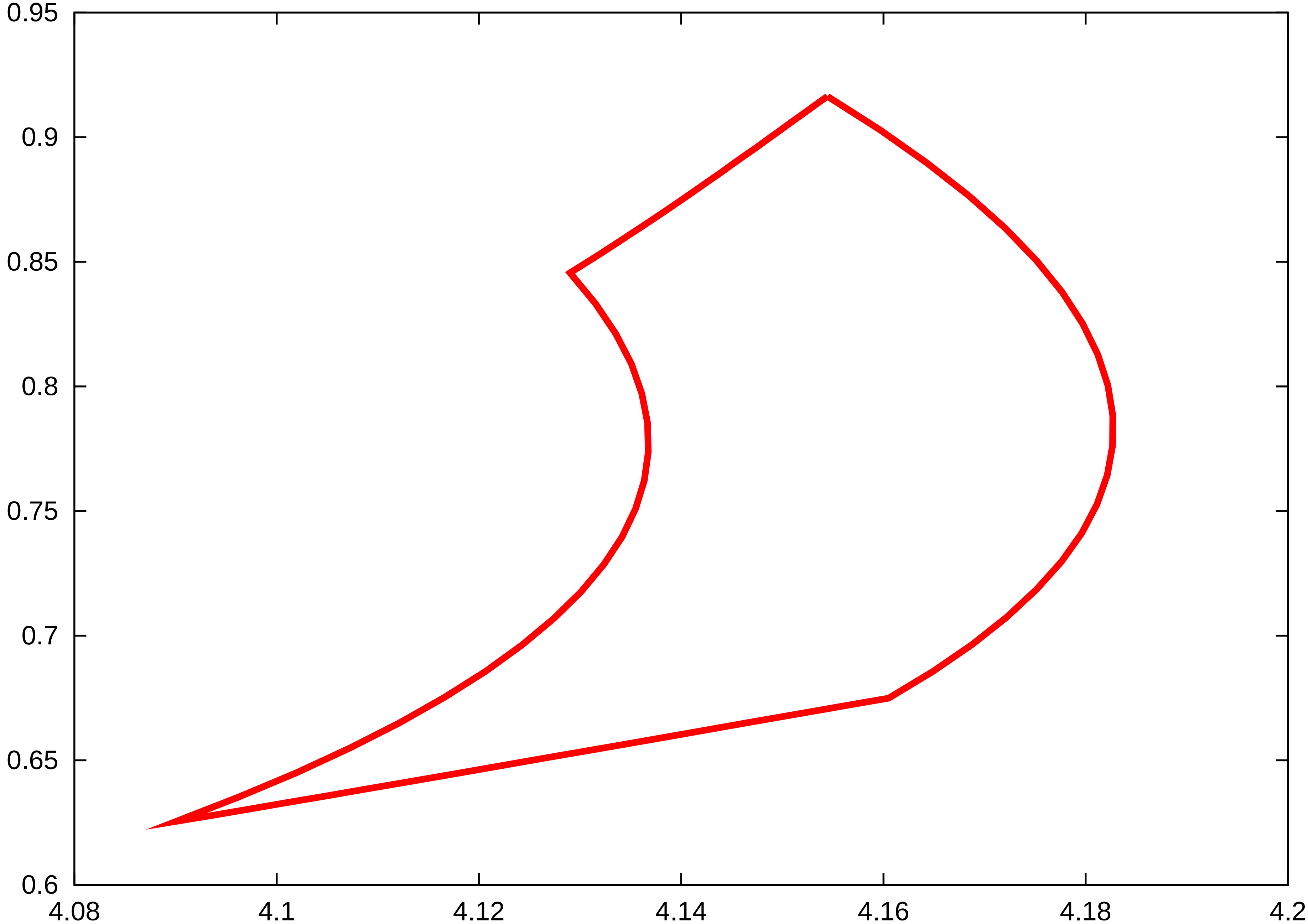
For extended domains (i.e. not only point solutions), this is **natural equivalent** to step size control. Similarity to what's done in global optimization.

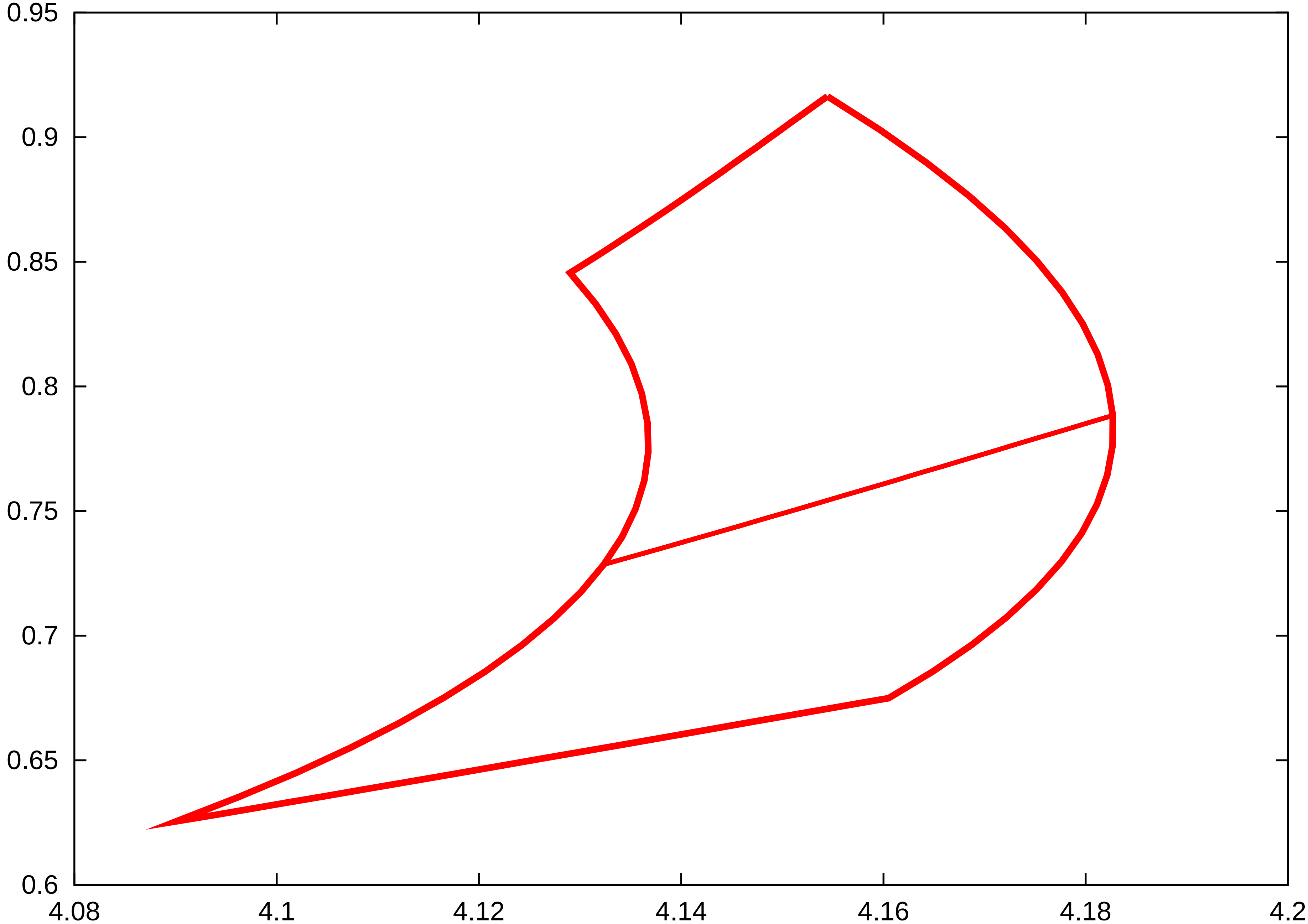
1. Evaluate ODE for $\Delta t = 0$ for current flow.
2. If remainder resulting remainder bound R greater than say $\varepsilon/10$, split domain along variable leading to longest axis.
3. Put one half of the box on stack for future work.

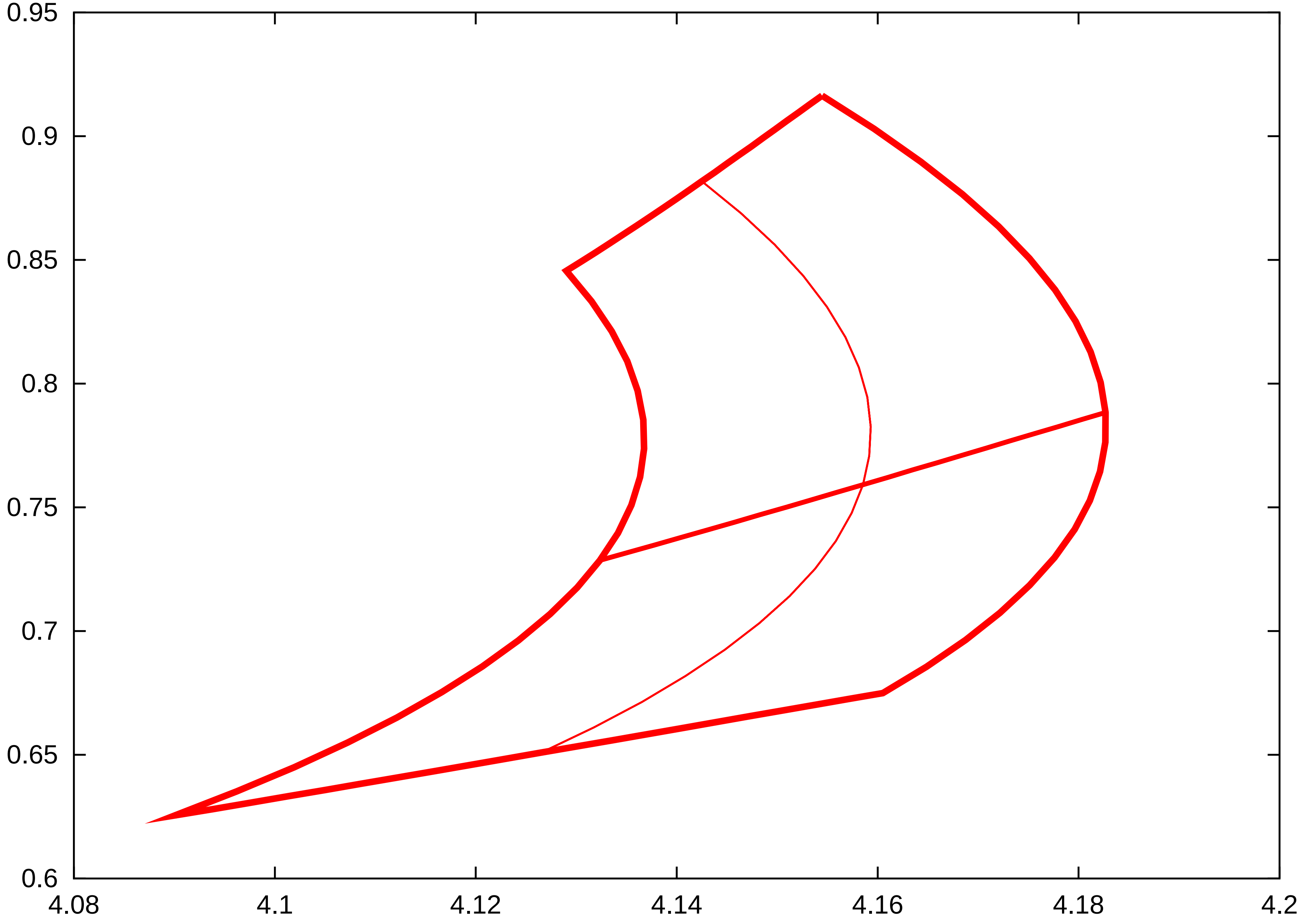
Things to consider:

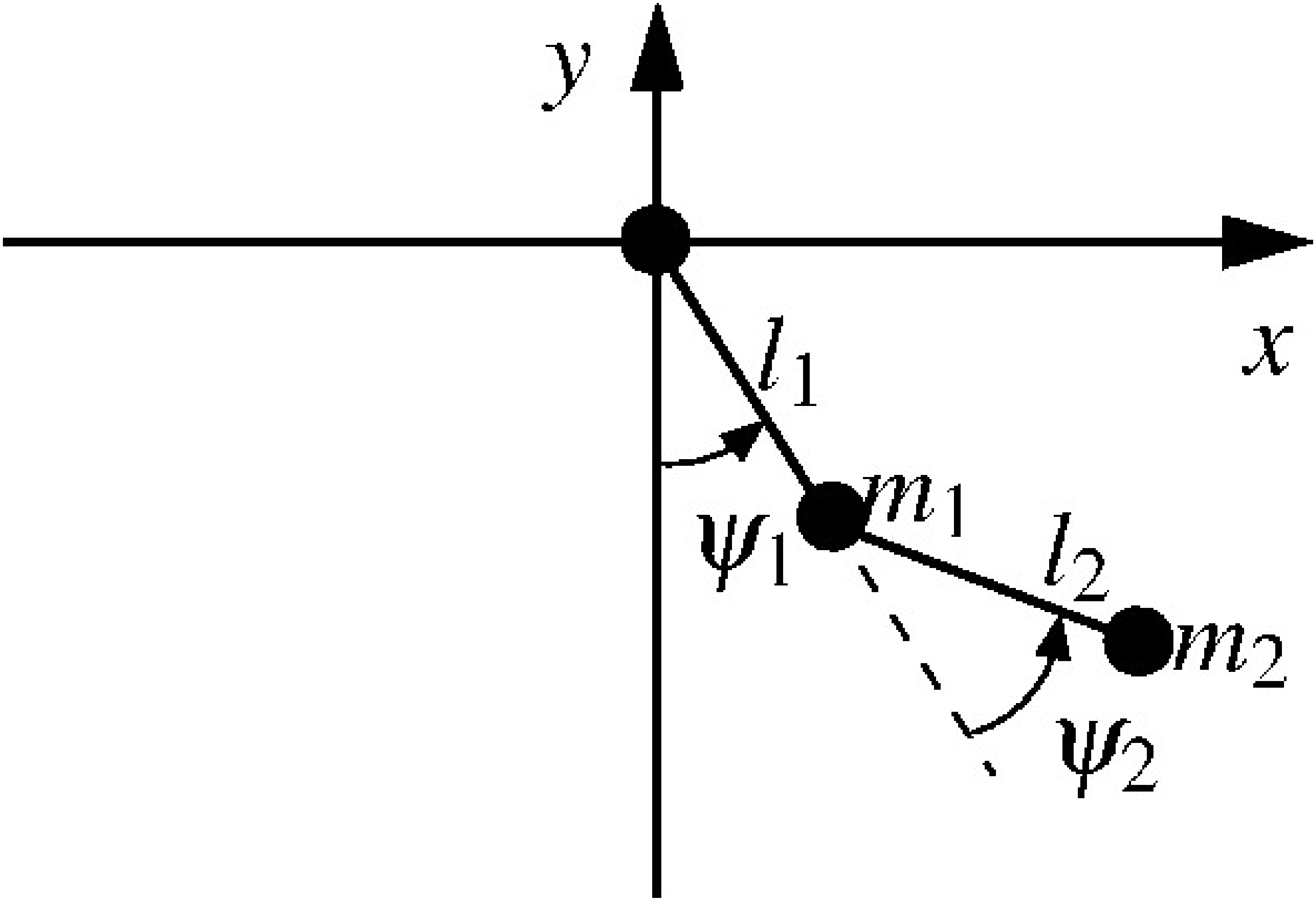
- Since TM provides inner and outer estimate, R is very convenient measure for actual overstimation
- Utilize "First-in-last-out" stack; minimizes stack length. Special adjustments for stack management in a parallel environment, including load balancing.

- When using QR preconditioning, make sure longest side stays longest. (Not a problem for CV preconditioning)
- Outlook: also dynamic order control for dependence on initial conditions


0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
4.08
4.1
4.12
4.14
4.16
4.18
4.2


0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
4.08
4.1
4.12
4.14
4.16
4.18
4.2


0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
4.08
4.1
4.12
4.14
4.16
4.18
4.2



y
x
l1
m1
ψ1
l2
m2
ψ2

The Double Pendulum - a Chaotic System

$$\frac{d^2}{dt^2}\psi_1 = \frac{l_1 m_2 \left[l_2(\dot\psi_1 + \dot\psi_2)^2 + l_1 \dot\psi_1^2 \cos\psi_2\right]}{l_1^2\left[m_1 + m_2 \sin^2\psi_2\right]} \sin\psi_2 + g \cdot \frac{-l_1(m_1+m_2)\sin\psi_1 + l_1 m_2 \cos\psi_2 \sin(\psi_1+\psi_2)}{l_1^2\left[m_1 + m_2\sin^2\psi_2\right]}$$

$$\begin{aligned}\frac{d^2}{dt^2}\psi_2 = &-\frac{(l_1(m_1+m_2) + l_2 m_2 \cos\psi_2) l_1 \dot\psi_1^2}{l_1 l_2 (m_1 + m_2\sin^2\psi_2)}\sin\psi_2 \\ &- \frac{l_2 m_2 (l_2 + l_1\cos\psi_2)(\dot\psi_1+\dot\psi_2)^2}{l_1 l_2(m_1+m_2\sin^2\psi_2)}\sin\psi_2 \\ &+ g\cdot\frac{(m_1+m_2)(l_2+l_1\cos\psi_2)\sin\psi_1}{l_1 l_2(m_1+m_2\sin^2\psi_2)} \\ &- g\cdot\frac{(l_1(m_1+m_2)+l_2 m_2\cos\psi_2)\sin(\psi_1+\psi_2)}{l_1 l_2(m_1+m_2\sin^2\psi_2)}\end{aligned}$$

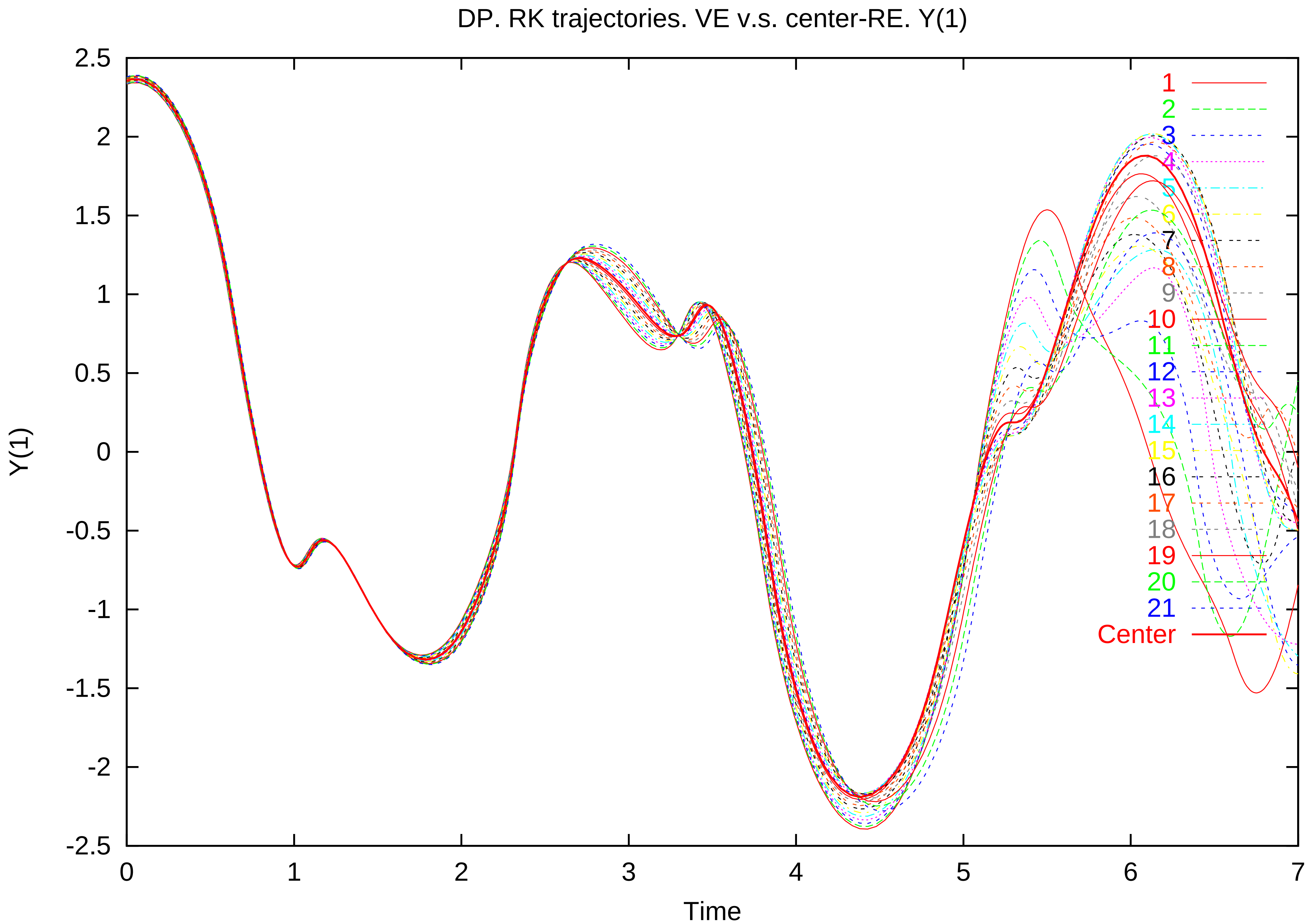
The Double Pendulum - Initial Conditions

In agreement with recent work of Rauh et al. (SCAN2006), we consider the parameter values $(l_1, l_2, m_1, m_2, g) = (1, 1, 1, 1, 9.81)$.

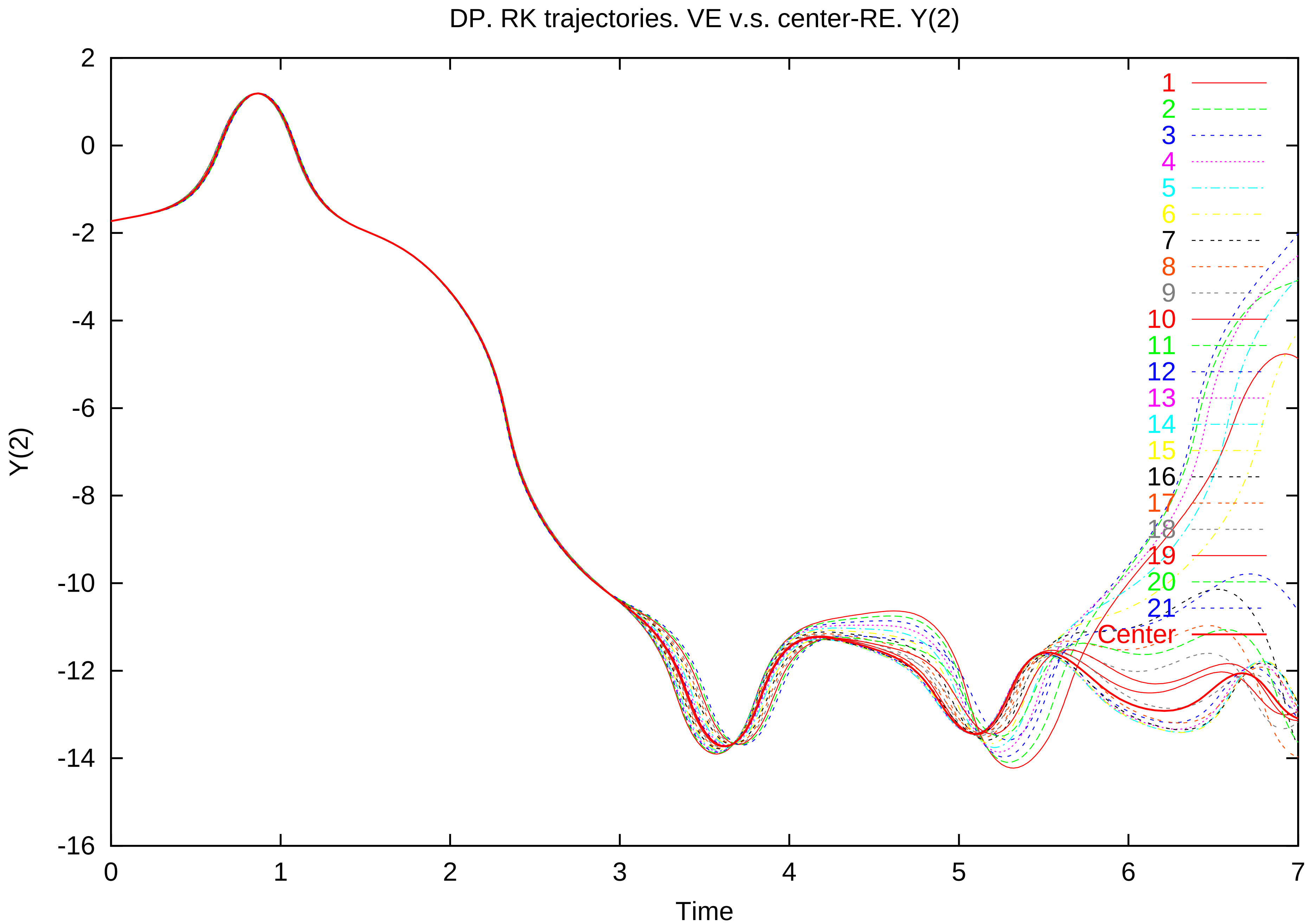
$$\psi_1(t=0) \in \frac{3\pi}{4} + \frac{1}{100}\frac{3\pi}{4}[-1,+1]$$
$$\psi_2(t=0) = -1.726533538$$
$$\dot{\psi}_1(t=0) = 0.4138843714$$
$$\dot{\psi}_2(t=0) = 0.6724072960$$

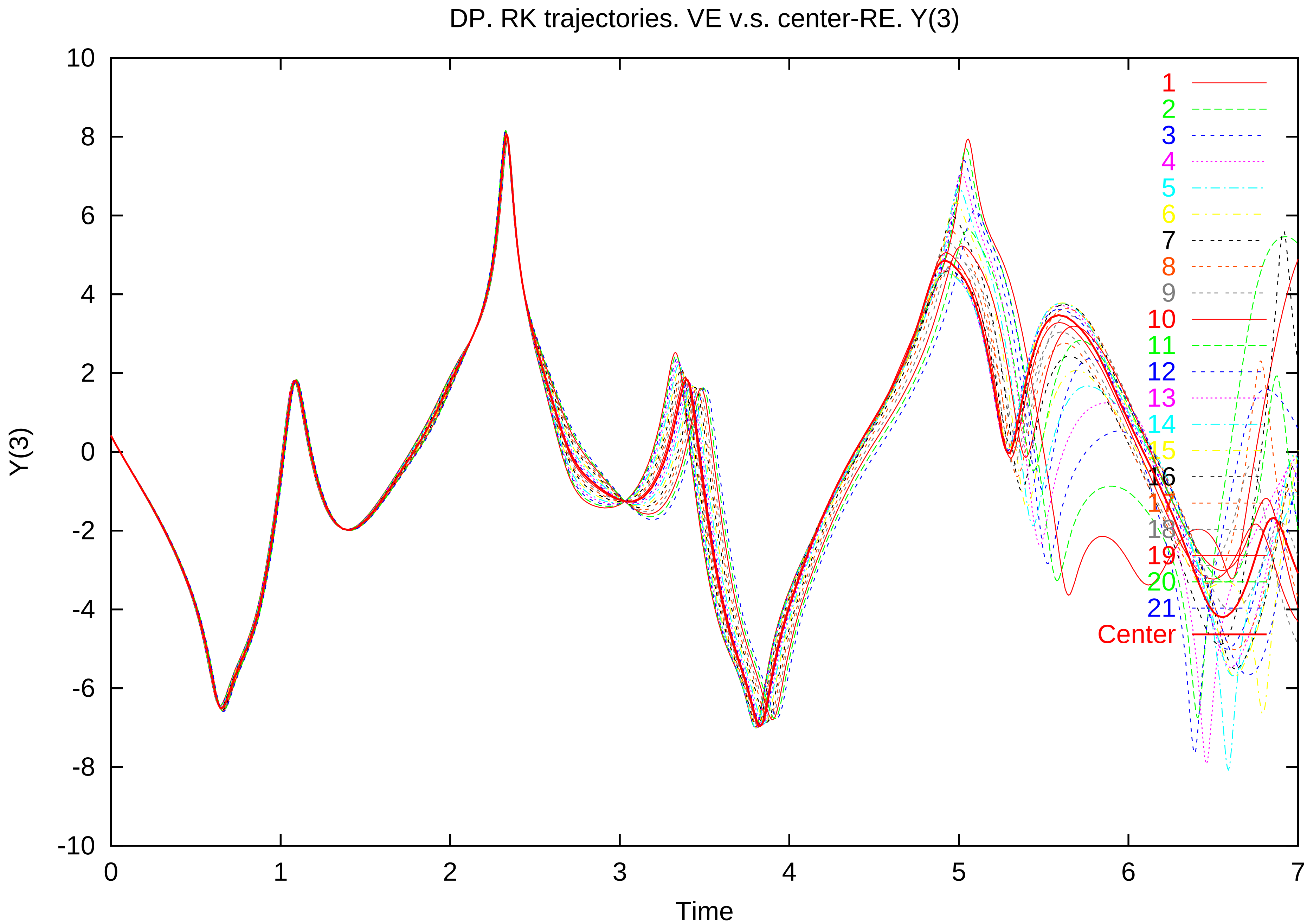
These initial conditions are in the **chaotic regime**. Illustration of motion (for similar, but not identical initial conditions):

`http://www.vis.uni-stuttgart.de/~kraus/`
`LiveGraphics3D/examples/parametrized/pendulum.html`

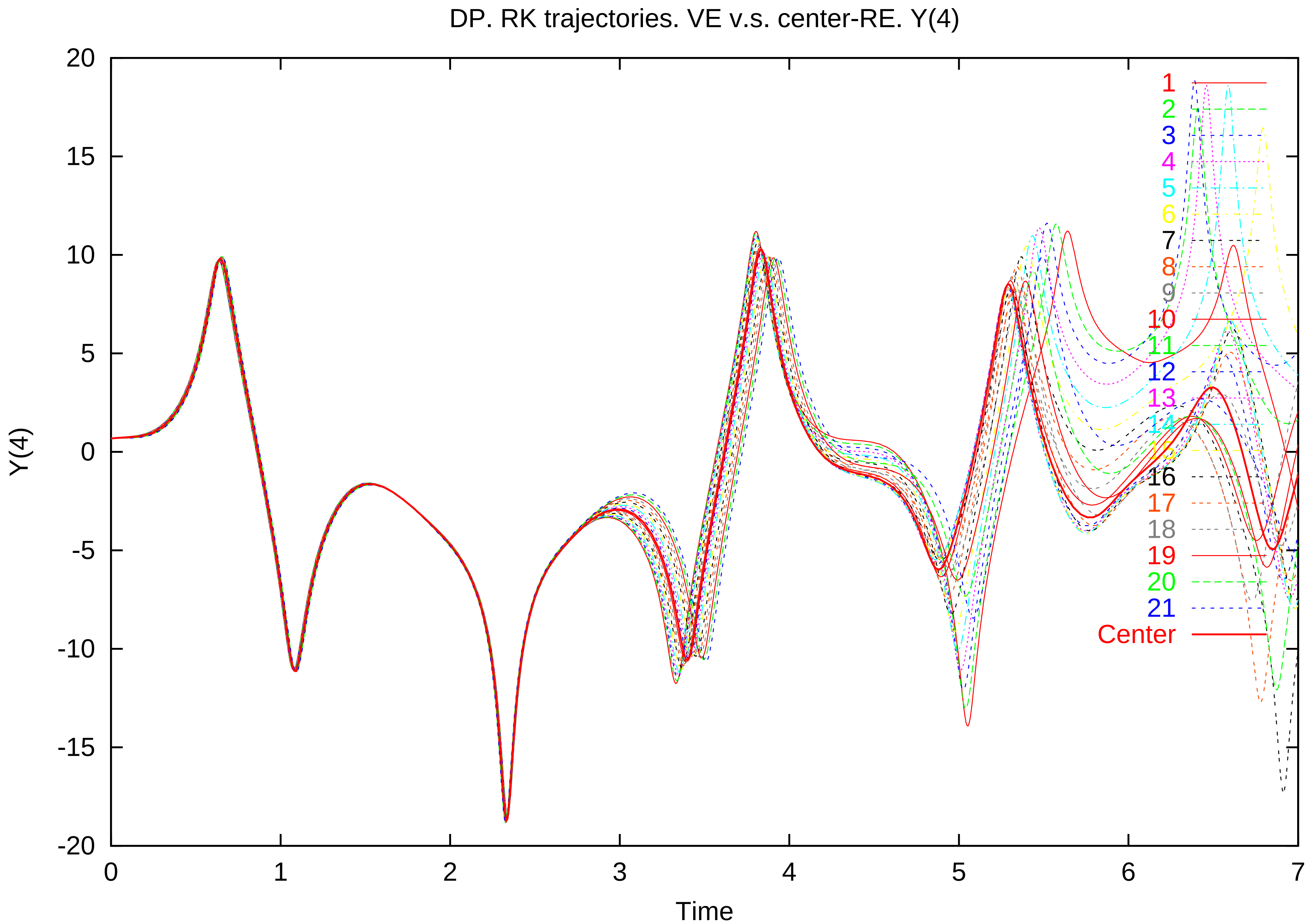


DP. RK trajectories. VE v.s. center-RE. Y(1)
Y(1)
Time
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
1
2
3
4
5
6
7
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
Center

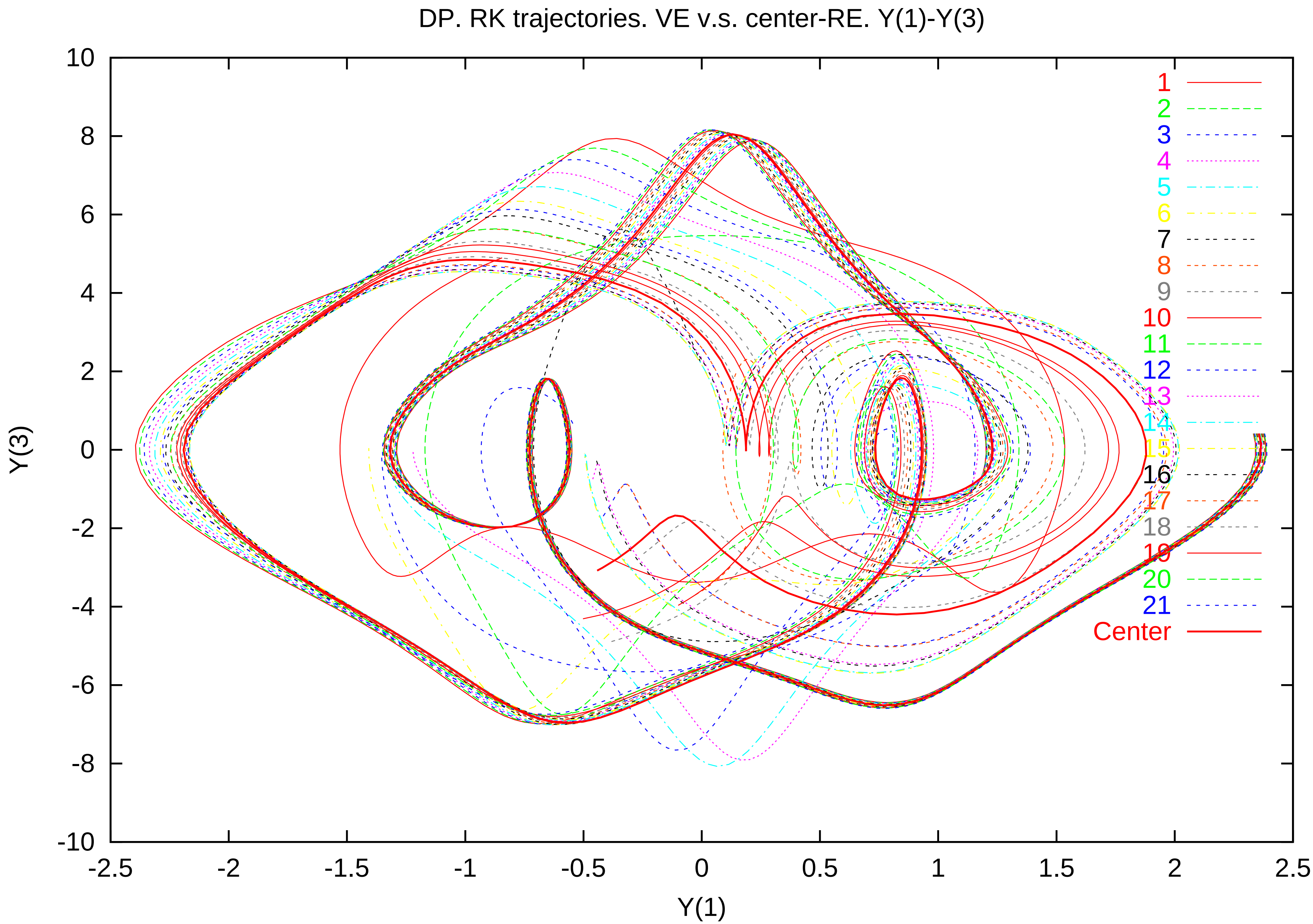

DP. RK trajectories. VE v.s. center-RE. Y(2)
Y(2)
Time
2
0
-2
-4
-6
-8
-10
-12
-14
-16
0
1
2
3
4
5
6
7
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
Center

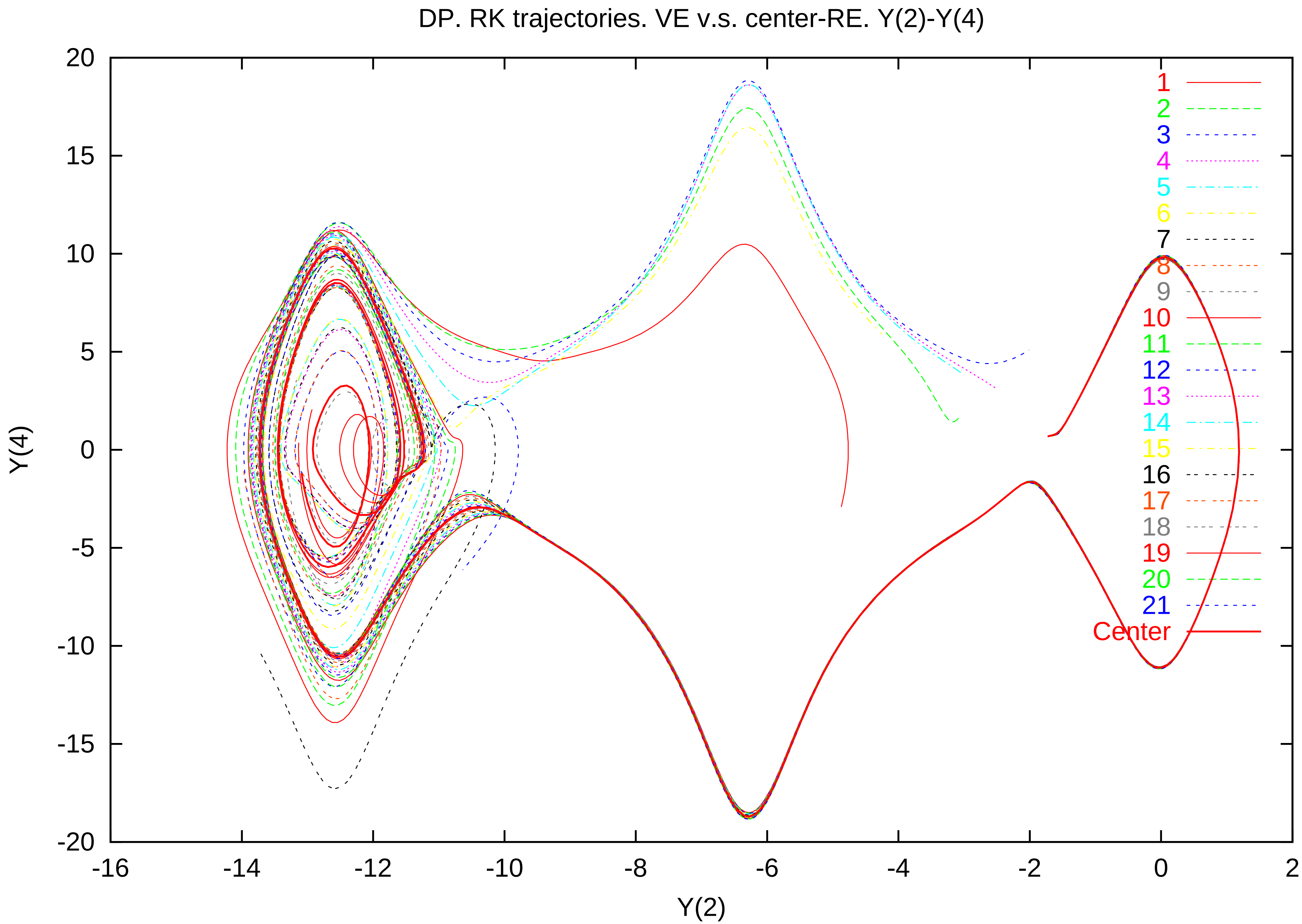


DP. RK trajectories. VE v.s. center-RE. Y(3)
Y(3)
Time
10
8
6
4
2
0
-2
-4
-6
-8
-10
0
1
2
3
4
5
6
7
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
Center



DP. RK trajectories. VE v.s. center-RE. Y(4)
Y(4)
Time
20
15
10
5
0
-5
-10
-15
-20
0
1
2
3
4
5
6
7
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
Center


DP. RK trajectories. VE v.s. center-RE. Y(1)-Y(3)
Y(3)
Y(1)
10
8
6
4
2
0
-2
-4
-6
-8
-10
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
Center


DP. RK trajectories. VE v.s. center-RE. Y(2)-Y(4)
Y(4)
Y(2)
20
15
10
5
0
-5
-10
-15
-20
-16
-14
-12
-10
-8
-6
-4
-2
0
2
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
Center

The Double Pendulum - Code Performance

Integration was carried out from $t = 0$ until $t = 0.5$ sec.
VNODE (Ned Nedialkov), QR method
ValEncIA-IVP (Rauh and Auer), Domain decomposition by lots of intervals, forward/backward integration for pruning
COSY-VI Taylor models, no domain decomposition at $t = 1.0$.
(The data reported for VNODE and ValEncIA-IVP are quoted from Rauh et al., SCAN2006).

Time t	**CPU VNODE**	**CPU ValEncIA**	**CPU COSY**
0.5	15.4 sec	5880 / 94 sec *	0.51 sec
1.0	(breakdown $t < 0.6$)	(breakdown $t < 0.6$)	2.04 sec

* First number: Implementation using Matlab-Intlab,
second number: using C++ interval library

The Double Pendulum - Check of COSY-VI

The doule pendulum **preservers energy**. Evaluating energy in Taylor model arithmetic over the entire flow at any two points in the integration, and subtracting the results, must result in a **tight enclosure of zero**.

Total Energy E is given as

$$E = m_1 \cdot g \cdot y_1 + m_2 \cdot g \cdot y_2 + \frac{1}{2} m_1 \left(\dot{x}_1^2 + \dot{y}_1^2 \right) + \frac{1}{2} m_2 \left(\dot{x}_2^2 + \dot{y}_2^2 \right).$$

Elementary arithmetic shows that

$$\begin{aligned}
&x_1 = l_1 \cdot \sin \psi_1,\ x_2 = x_1 + l_2 \cdot \sin(\psi_1 + \psi_2) \\
&y_1 = -l_1 \cdot \cos \psi_1,\ \ y_2 = y_1 - l_2 \cdot \cos(\psi_1 + \psi_1) \\
&\dot{x}_1 = \dot{\psi}_1 \cdot l_1 \cdot \cos \psi_1,\ \ \dot{x}_2 = \dot{x}_1 + (\dot{\psi}_1 + \dot{\psi}_2) \cdot l_2 \cdot \cos(\psi_1 + \psi_2) \\
&\dot{y}_1 = \dot{\psi}_1 \cdot l_1 \cdot \sin \psi_1,\ \ \dot{y}_2 = \dot{y}_1 + (\dot{\psi}_1 + \dot{\psi}_2) \cdot l_2 \cdot \sin(\psi_1 + \psi_2)
\end{aligned}$$

The Double Pendulum - Energy at t=0

```
I  COEFFICIENT              ORDER EXPONENTS
1   6.636304564436251         0    0 0  0 0  0
2  0.4629982784681443         1    1 0  0 0  0
3  -.1650152672869661E-02    2    2 0  0 0  0
4  -.4284009231437226E-04    3    3 0  0 0  0
5  0.7634228476230531E-07    4    4 0  0 0  0
6  0.1189166522762920E-08    5    5 0  0 0  0
7  -.1412752780648741E-11    6    6 0  0 0  0
8  -.1571866492860271E-13    7    7 0  0 0  0
R [-.1538109061161243E-012,0.1517608952722424E-012]
```

The Double Pendulum - Energy at t=0.5

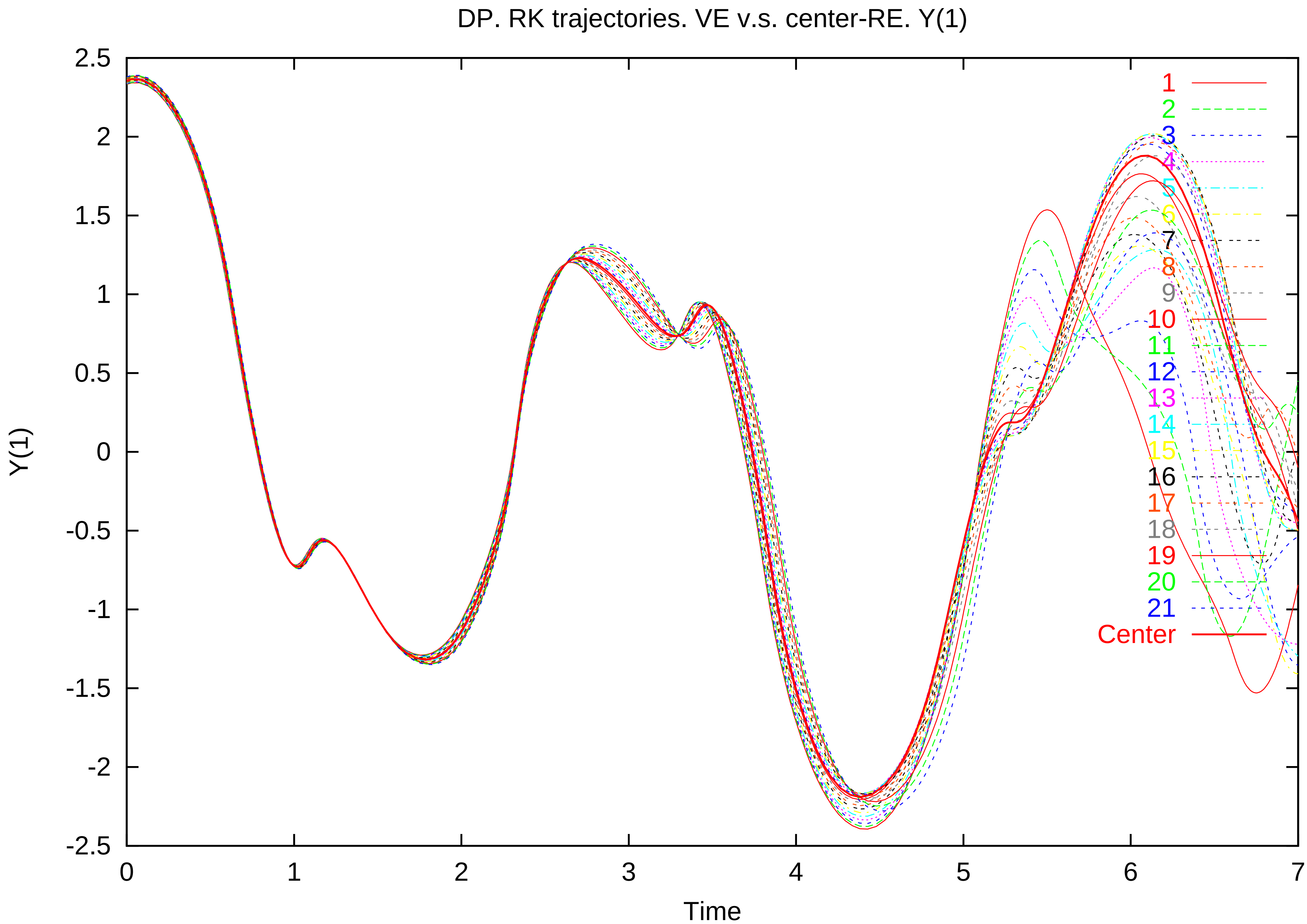
I	COEFFICIENT	ORDER	EXPONENTS				
1	6.636304564436253	0	0	0	0	0	0
2	0.4629982784681632	1	1	0	0	0	0
3	-.1650152672942219E-02	2	2	0	0	0	0
4	-.4284009217517837E-04	3	3	0	0	0	0
5	0.7634212049420934E-07	4	4	0	0	0	0
6	0.1189297979605227E-08	5	5	0	0	0	0
7	-.1487493064301731E-11	6	6	0	0	0	0
8	0.1498746352978318E-13	7	7	0	0	0	0
9	-.8978311500296960E-14	8	8	0	0	0	0
10	0.1732136627097570E-14	9	9	0	0	0	0
11	-.1410358744591400E-15	10	10	0	0	0	0
12	-.3488804283416099E-16	11	11	0	0	0	0
13	0.1647113913603616E-16	12	12	0	0	0	0

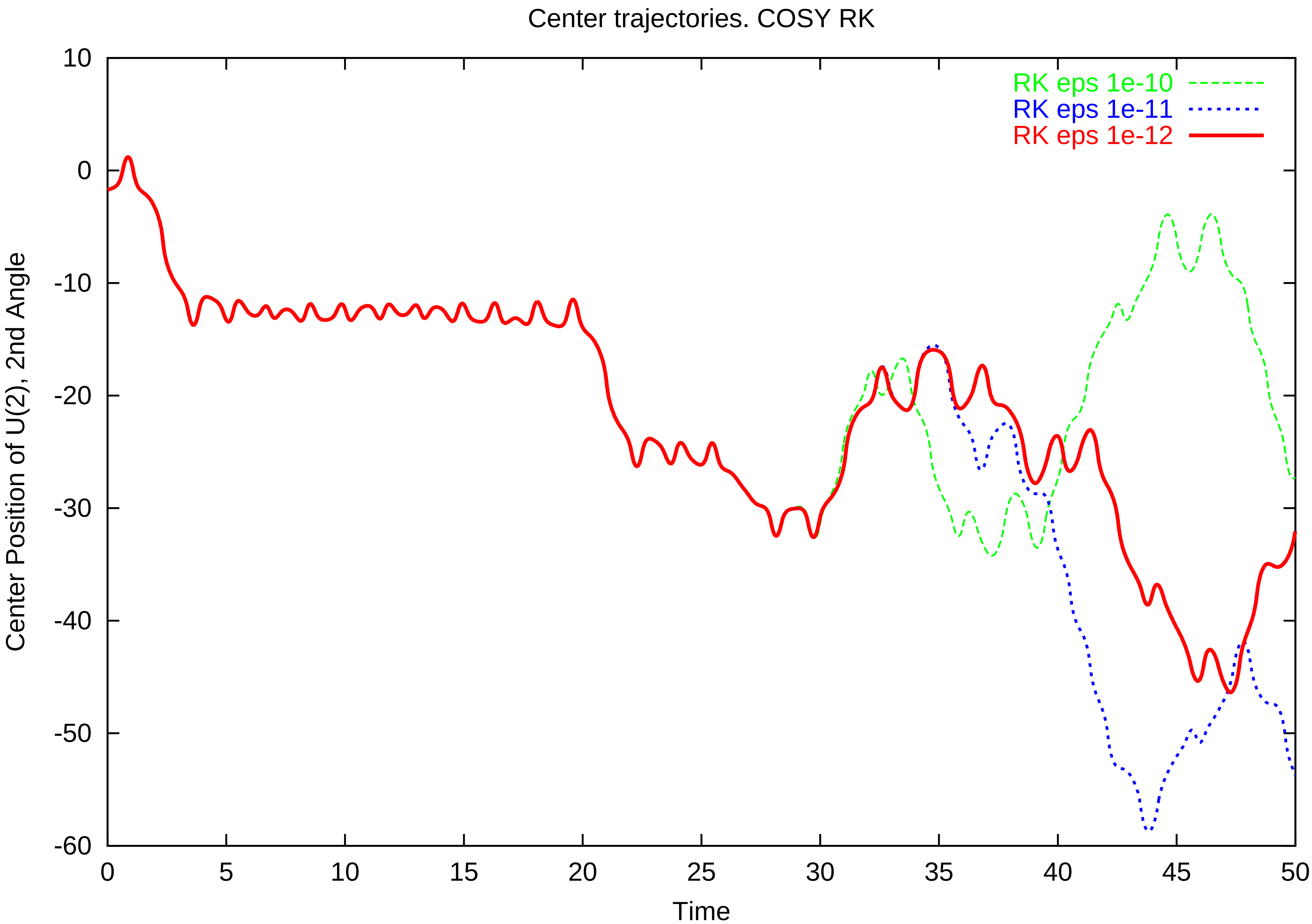
R [-.6845903858358710E-010,0.7016561210090576E-010]

The Double Pendulum - Energy Difference

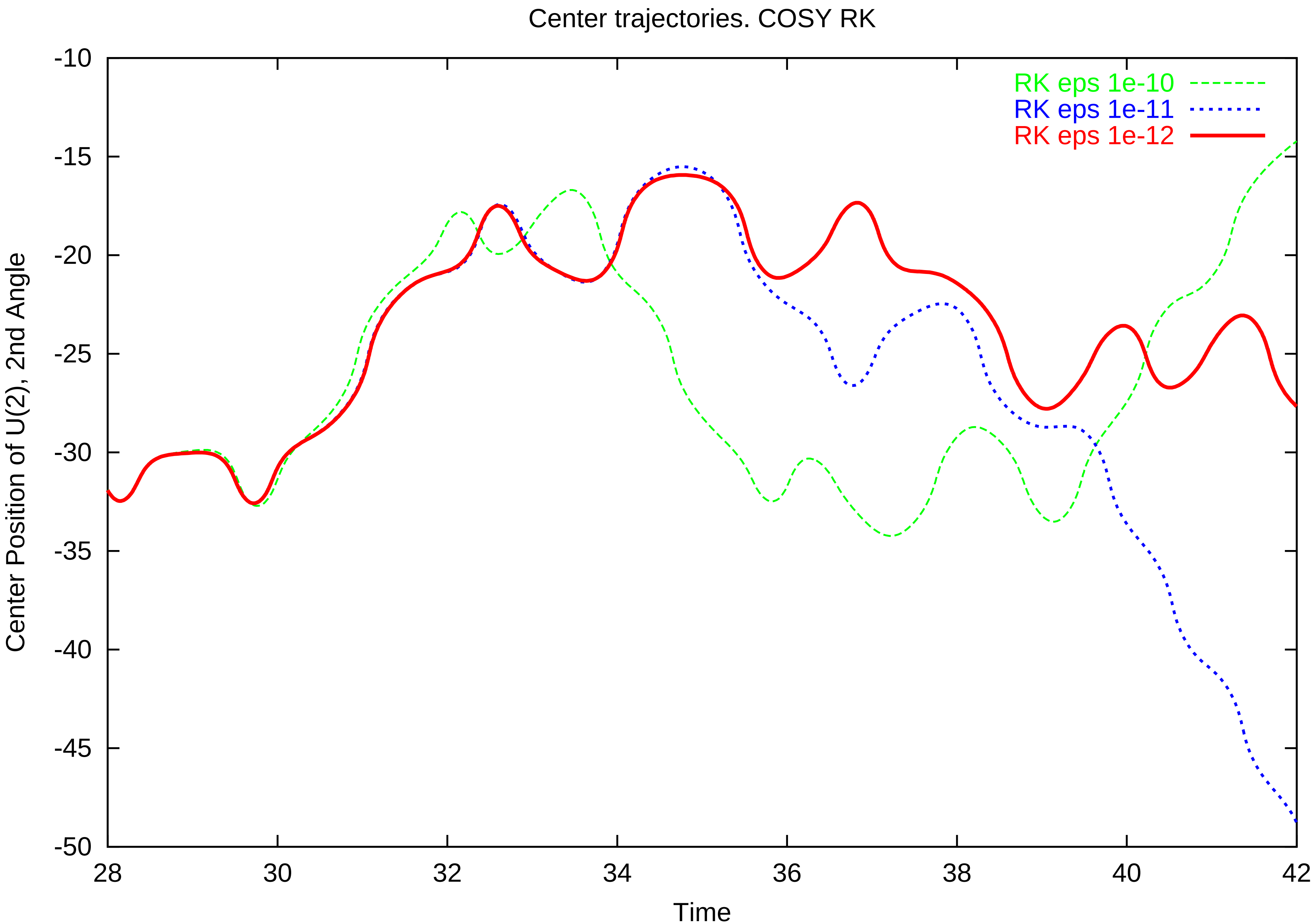
I	COEFFICIENT	ORDER	EXPONENTS				
1	0.266453525910037E-14	0	0	0	0	0	0
2	0.188182802673964E-13	1	1	0	0	0	0
3	-.725584956701164E-13	2	2	0	0	0	0
4	0.139193893227990E-12	3	3	0	0	0	0
5	-.164268095972426E-12	4	4	0	0	0	0
6	0.131456842307669E-12	5	5	0	0	0	0
7	-.747402836529906E-13	6	6	0	0	0	0
8	0.307061284583859E-13	7	7	0	0	0	0
9	-.899231705828288E-14	8	8	0	0	0	0
10	0.173213662709757E-14	9	9	0	0	0	0
11	-.141035874459140E-15	10	10	0	0	0	0
12	-.348880428341609E-16	11	11	0	0	0	0
13	0.164711391360361E-16	12	12	0	0	0	0

R [-.686171064301878E-010,0.703257299583504E-010]


DP. RK trajectories. VE v.s. center-RE. Y(1)
Y(1)
Time
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
1
2
3
4
5
6
7
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
Center



Center trajectories. COSY RK
RK eps 1e-10
RK eps 1e-11
RK eps 1e-12
10
0
-10
-20
-30
-40
-50
-60
0
5
10
15
20
25
30
35
40
45
50
Time
Center Position of U(2), 2nd Angle

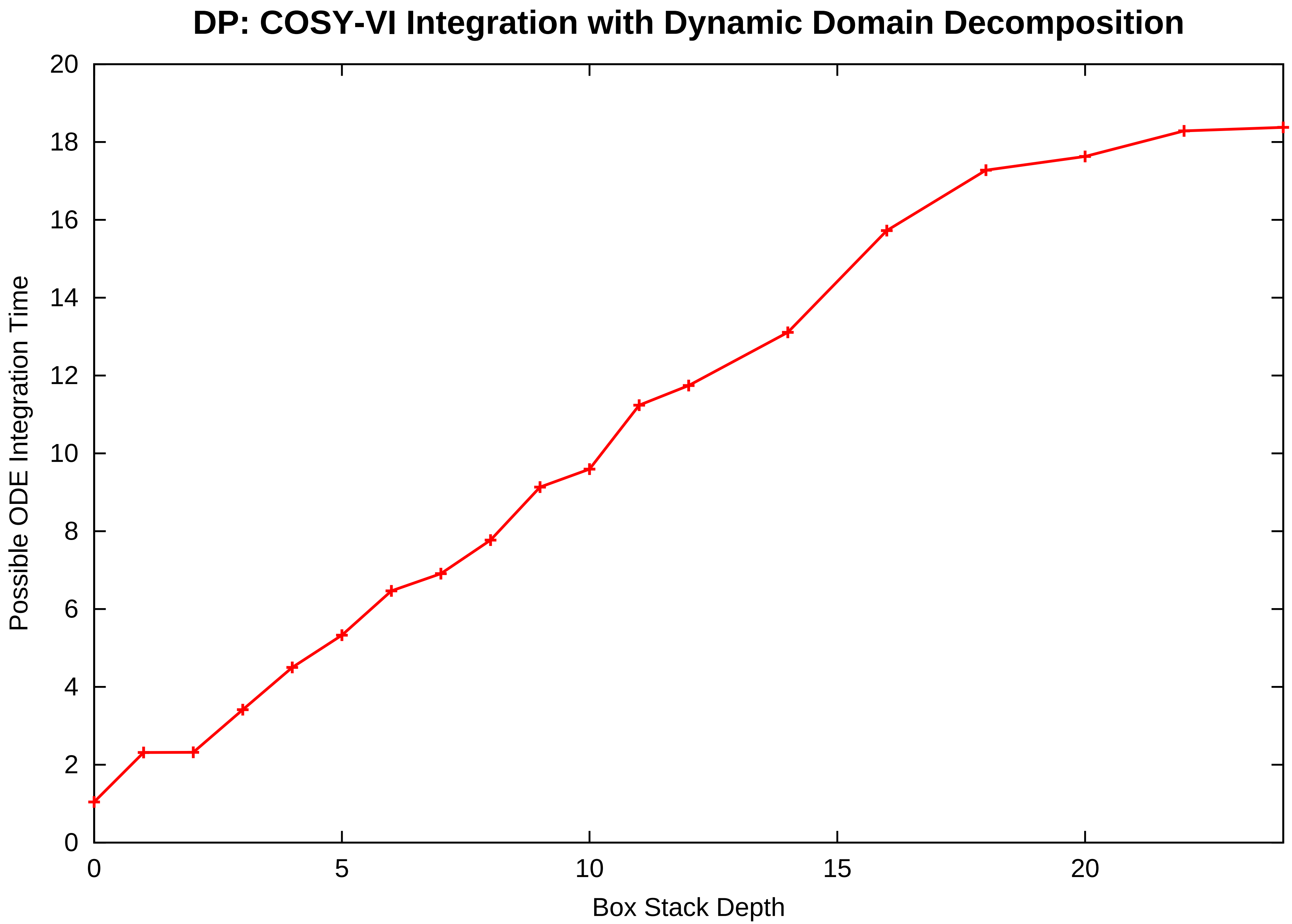

Center trajectories. COSY RK
RK eps 1e-10
RK eps 1e-11
RK eps 1e-12
Center Position of U(2), 2nd Angle
Time
-10
-15
-20
-25
-30
-35
-40
-45
-50
28
30
32
34
36
38
40
42

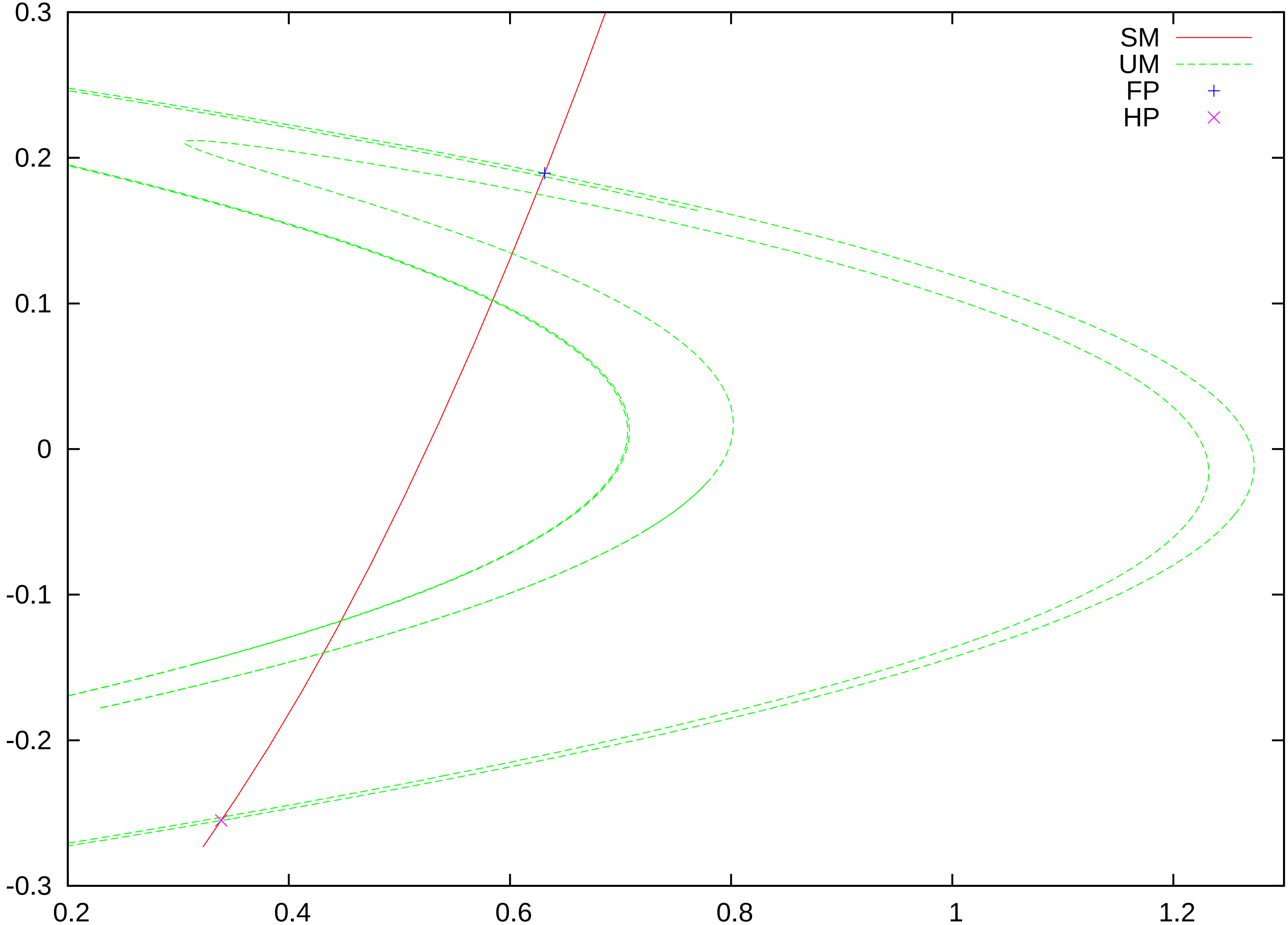
Long Term Behavior

We observe the following:

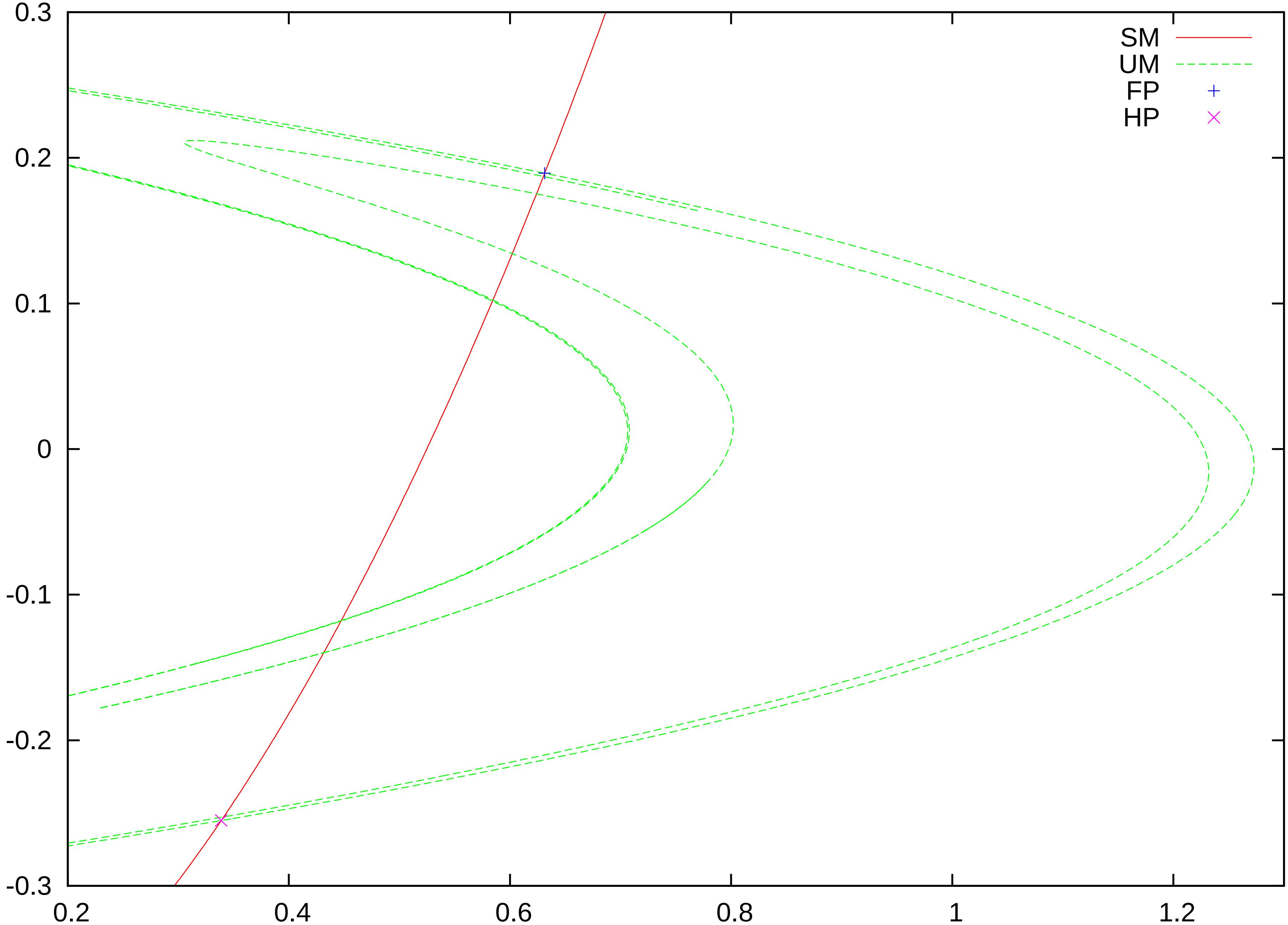
1. Around $t = 2$, initial condition range leads to noticeable broadening of ranges
2. Around $t = 5$, initial condition range leads to angle spread by $> 2\pi$, i.e. different full revolutions
3. Around $t = 30$, conventional non-verified integrators reach their accuracy limit

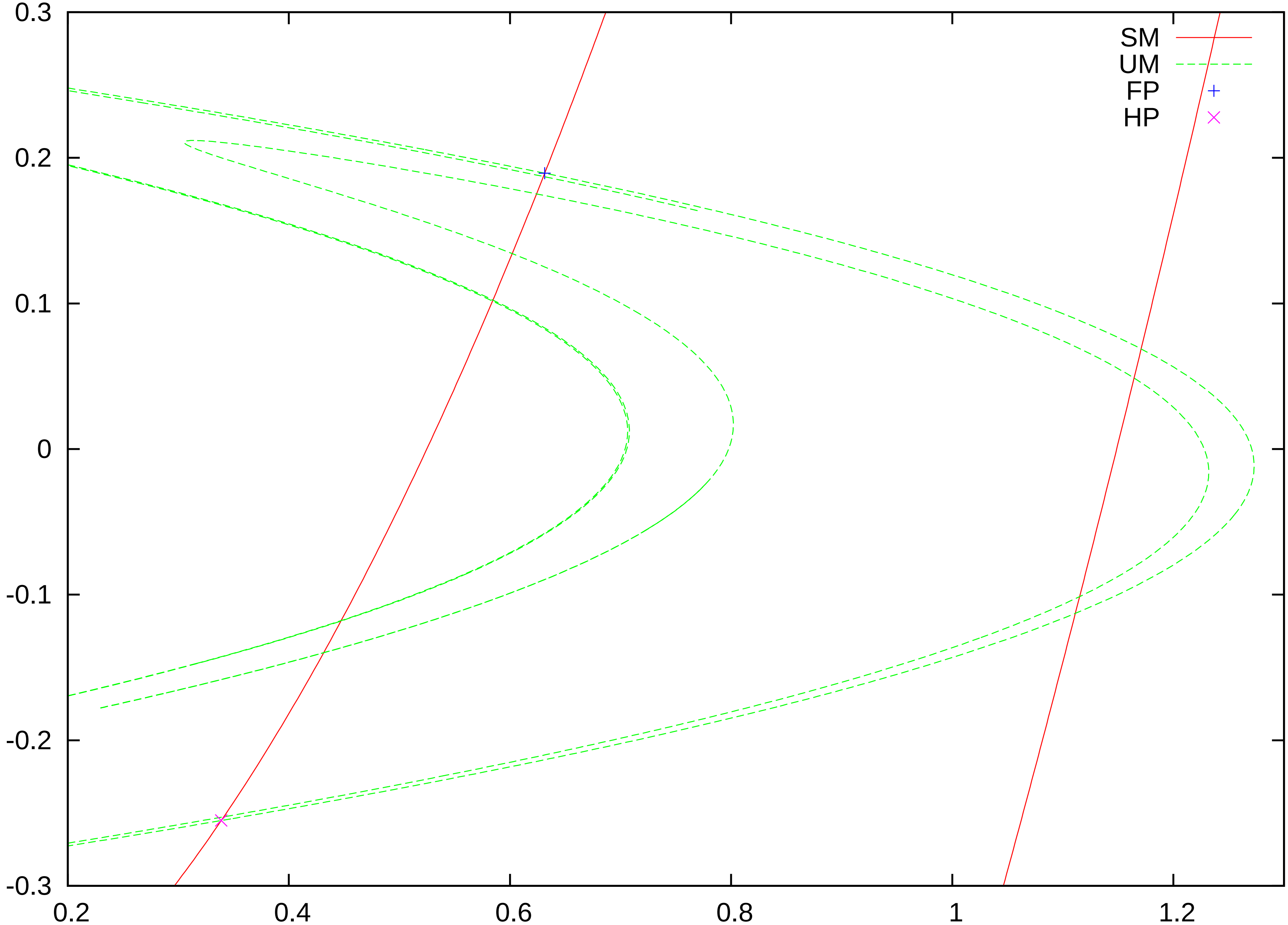
Question: How long can COSY integrate with dynamic domain decomposition?

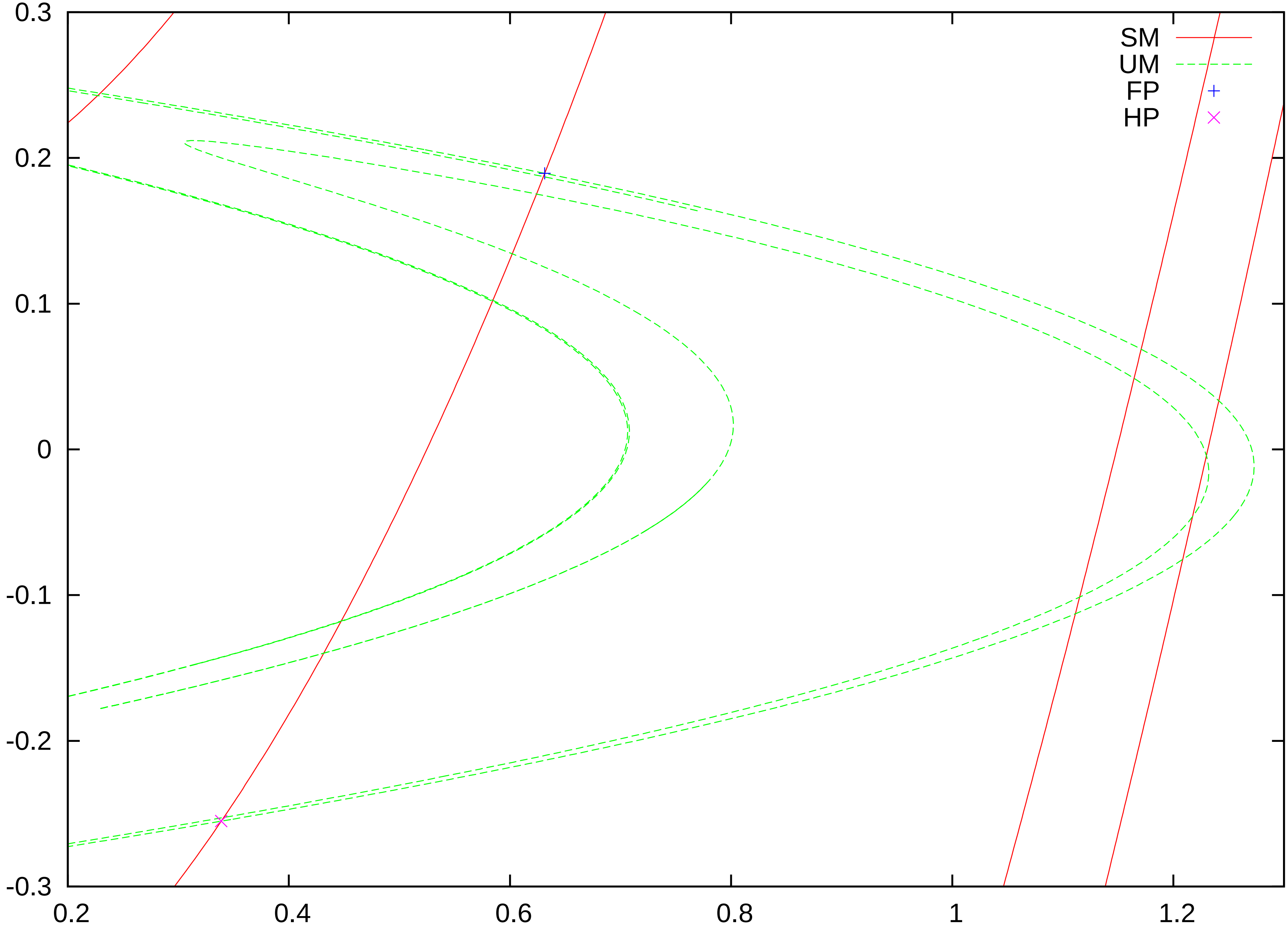

DP: COSY-VI Integration with Dynamic Domain Decomposition
Possible ODE Integration Time
Box Stack Depth
0
2
4
6
8
10
12
14
16
18
20
0
5
10
15
20



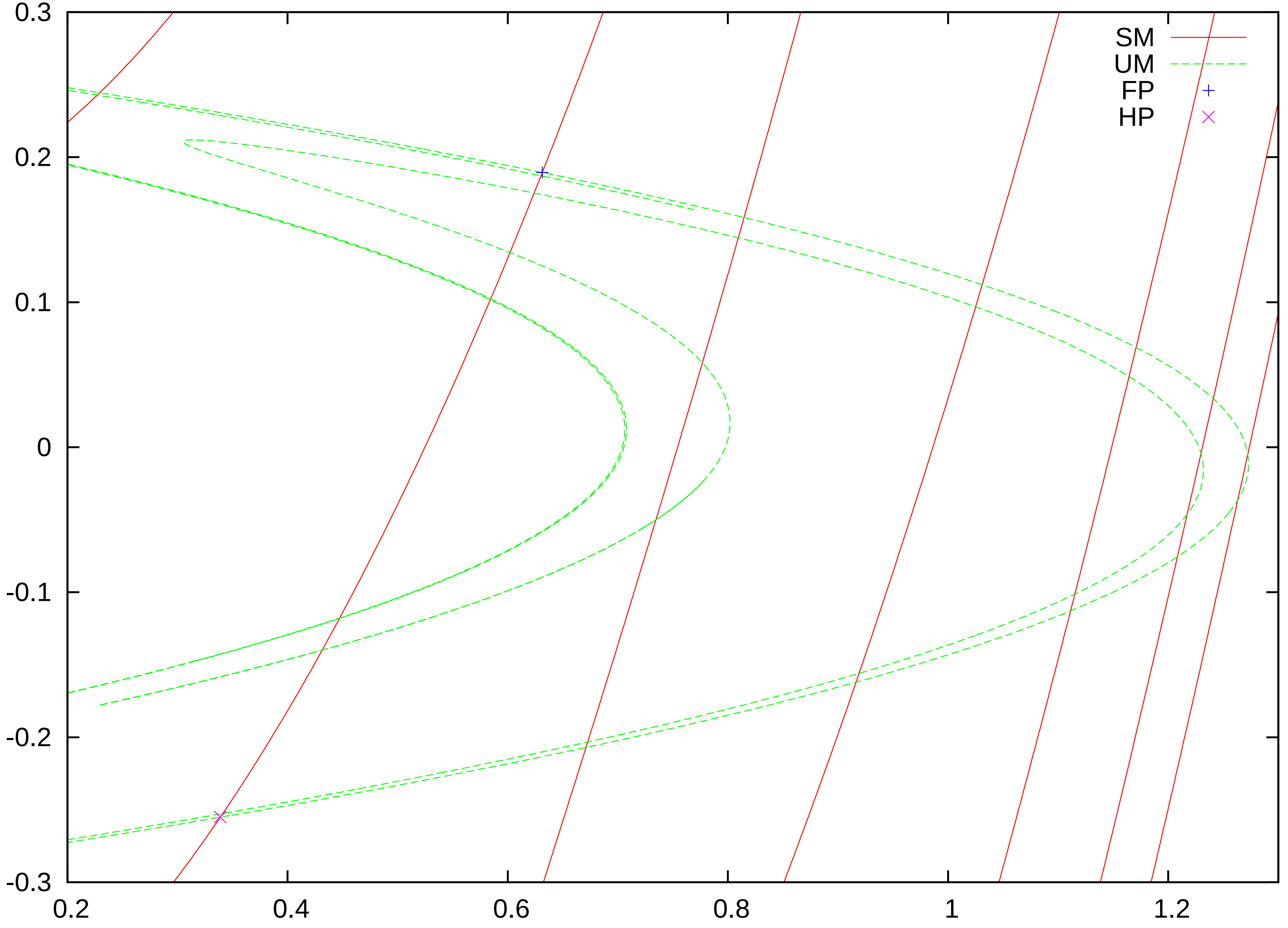
SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2


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

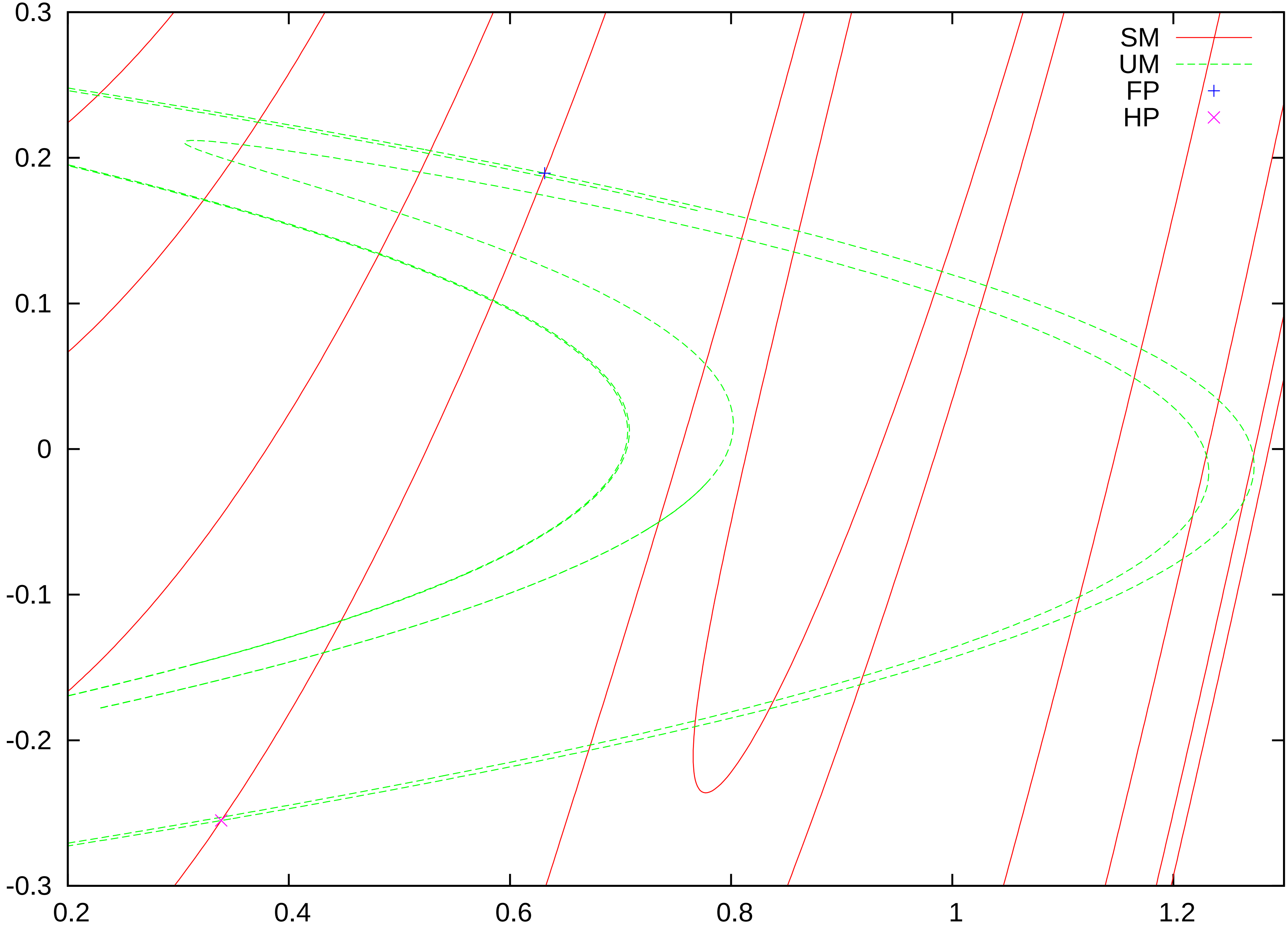

SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2



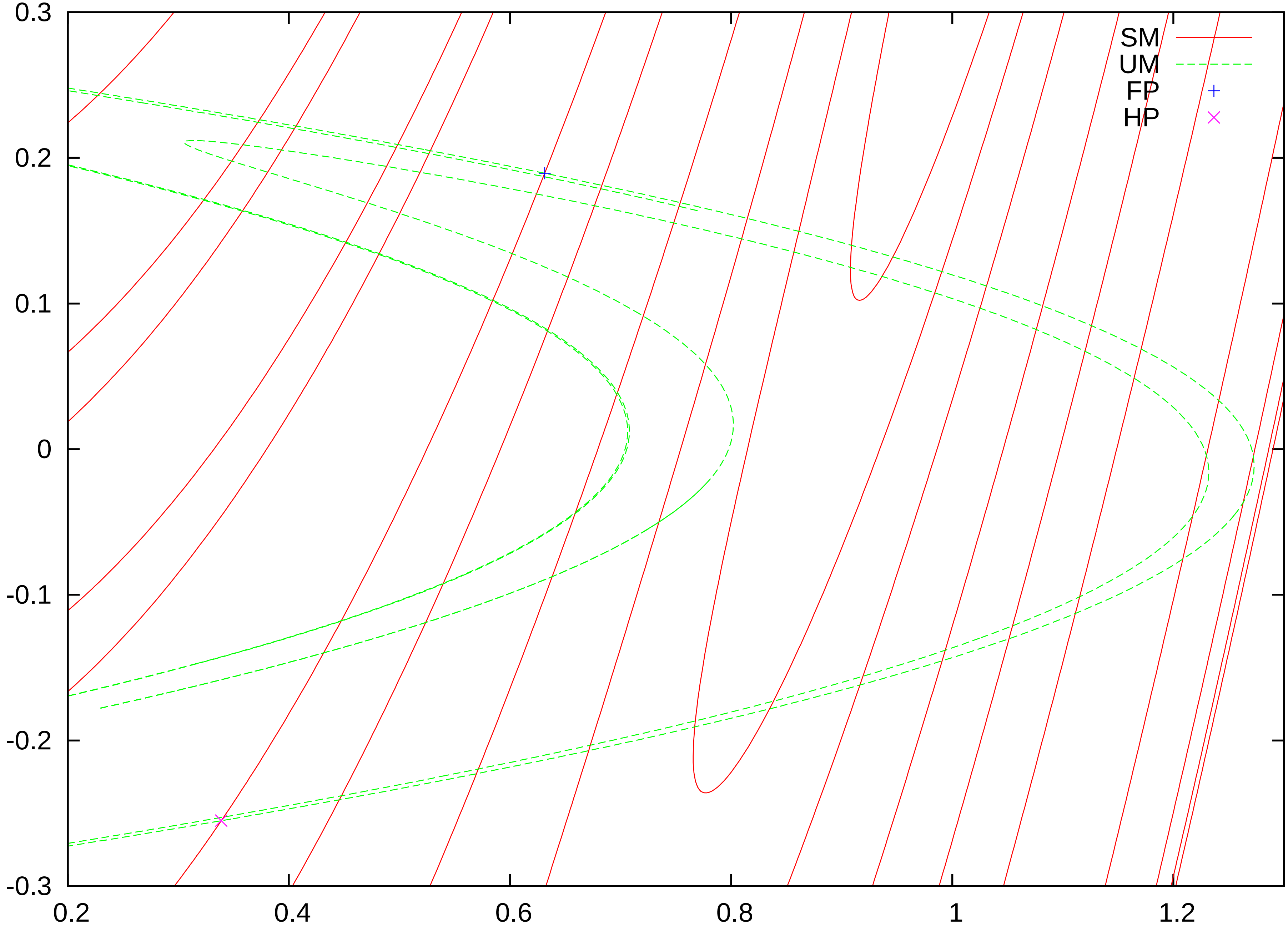
SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2

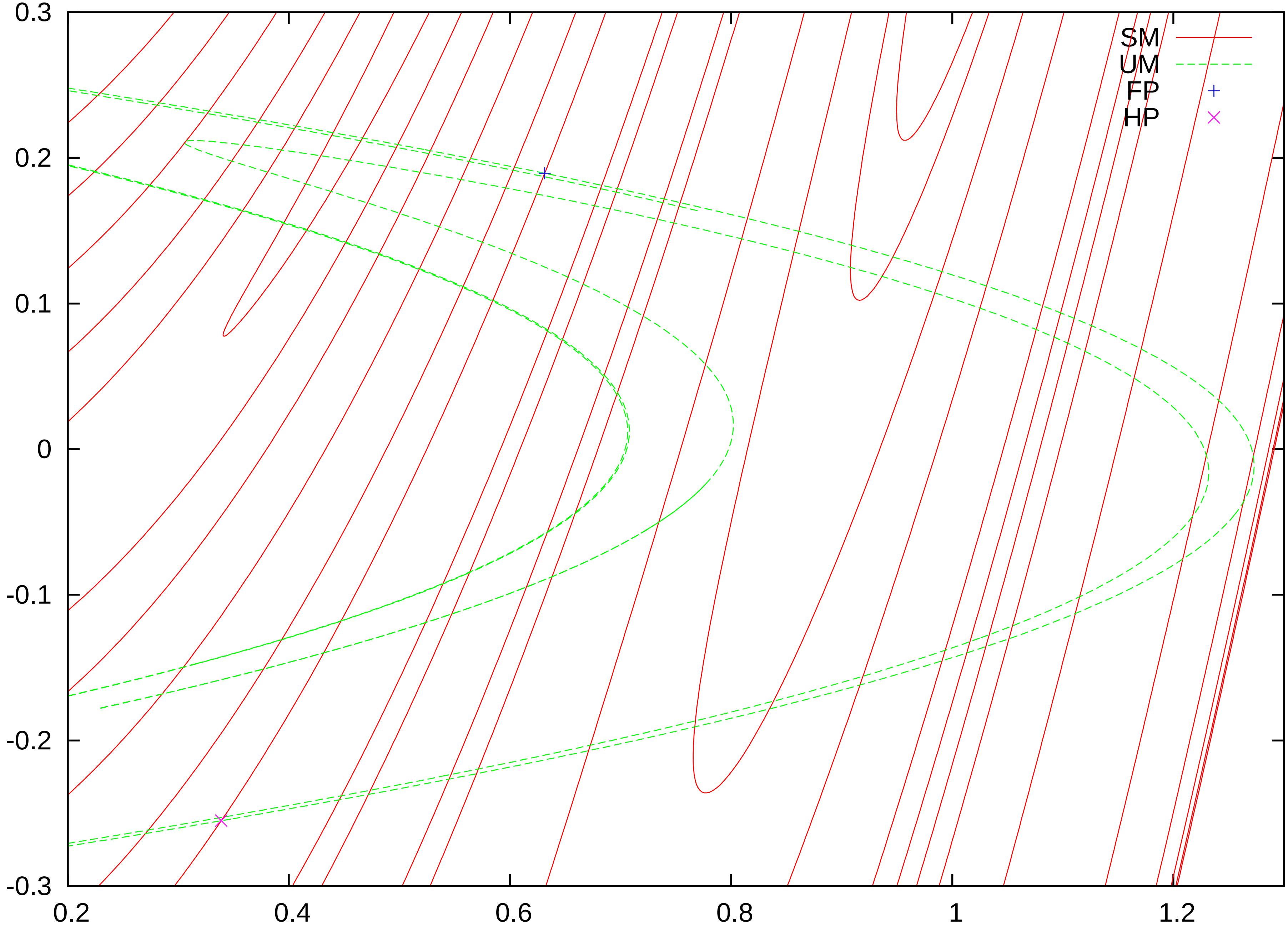


0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2
SM
UM
FP
HP

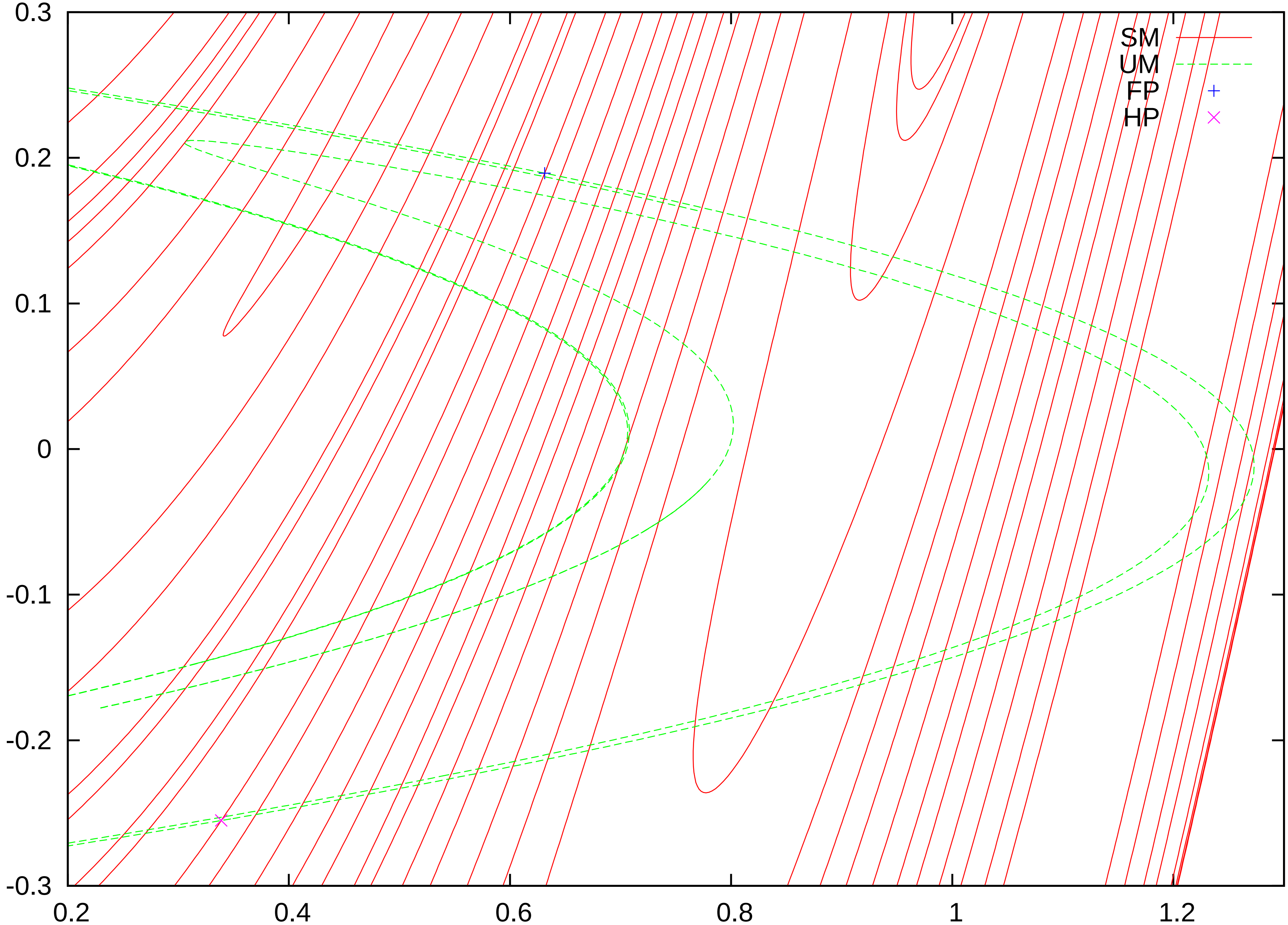


SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2

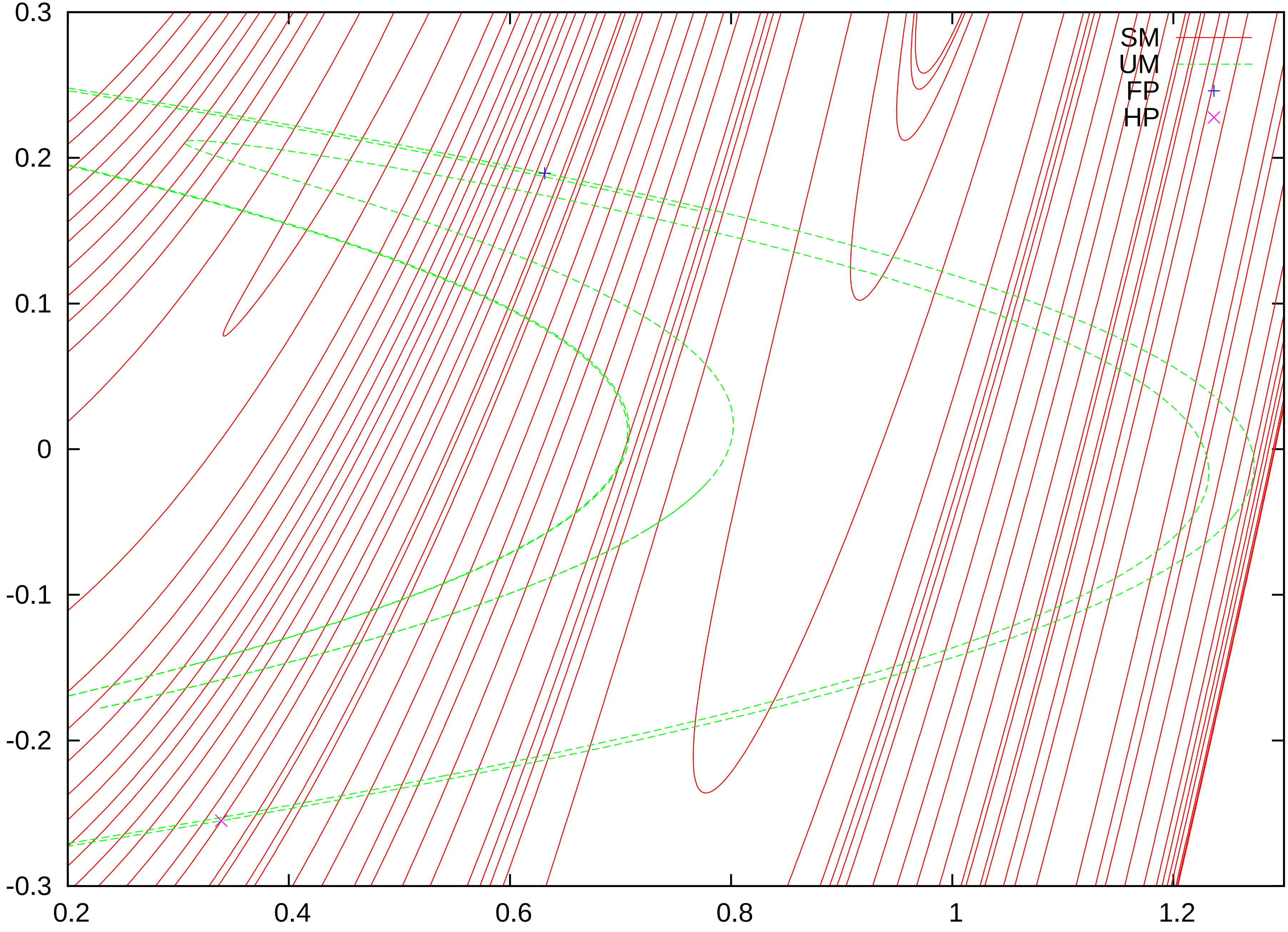

SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2



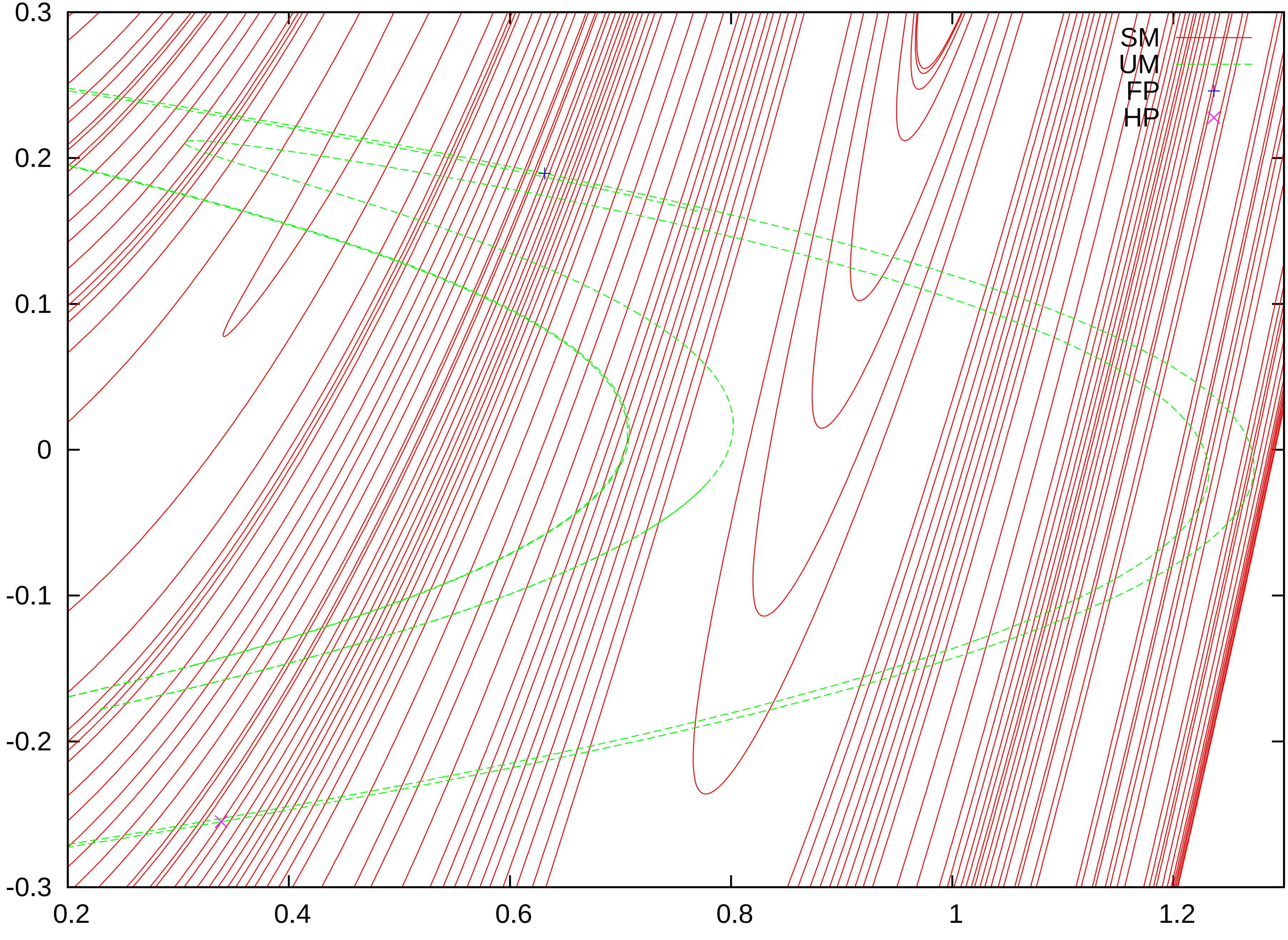
SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2



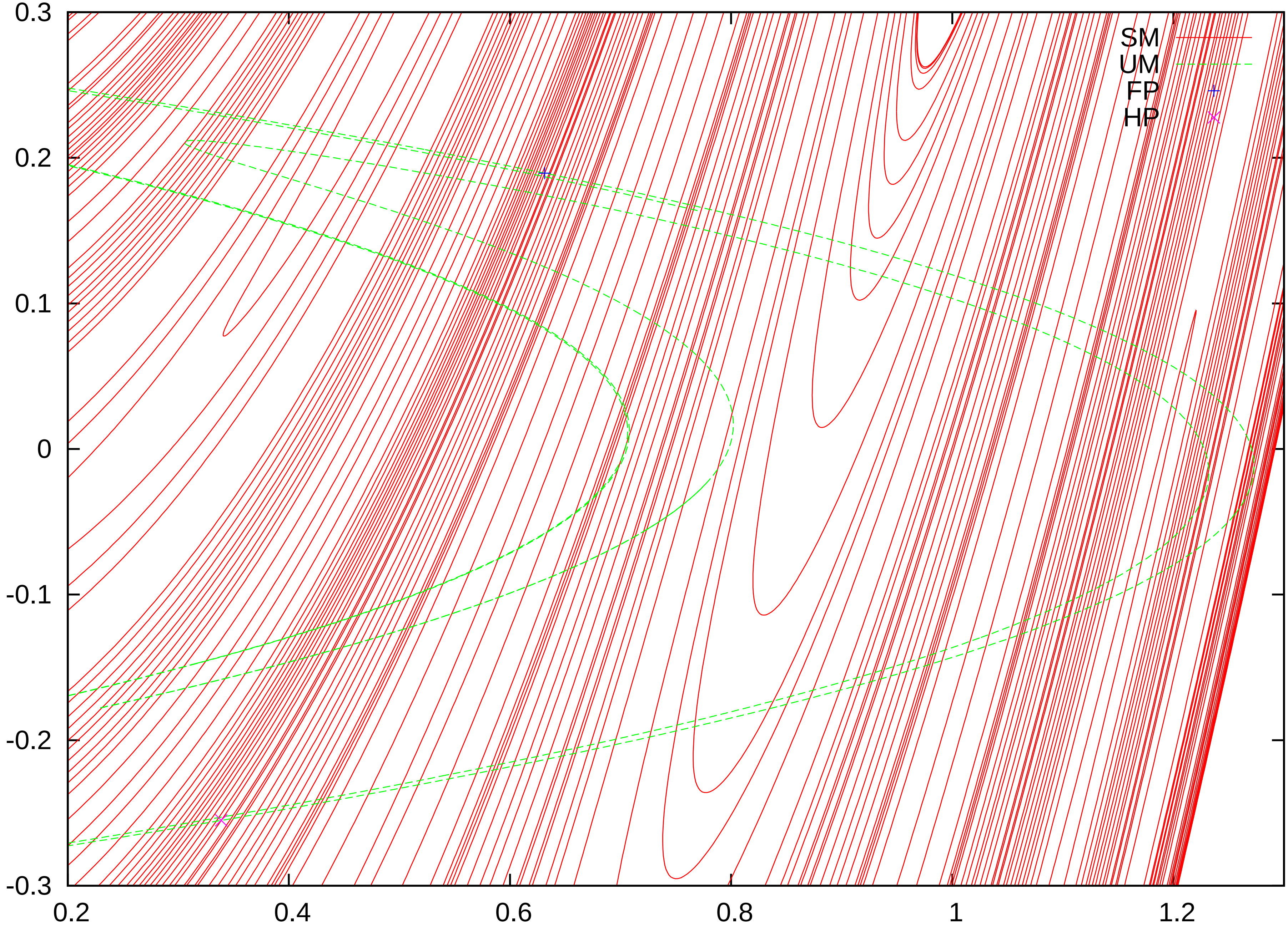
SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2



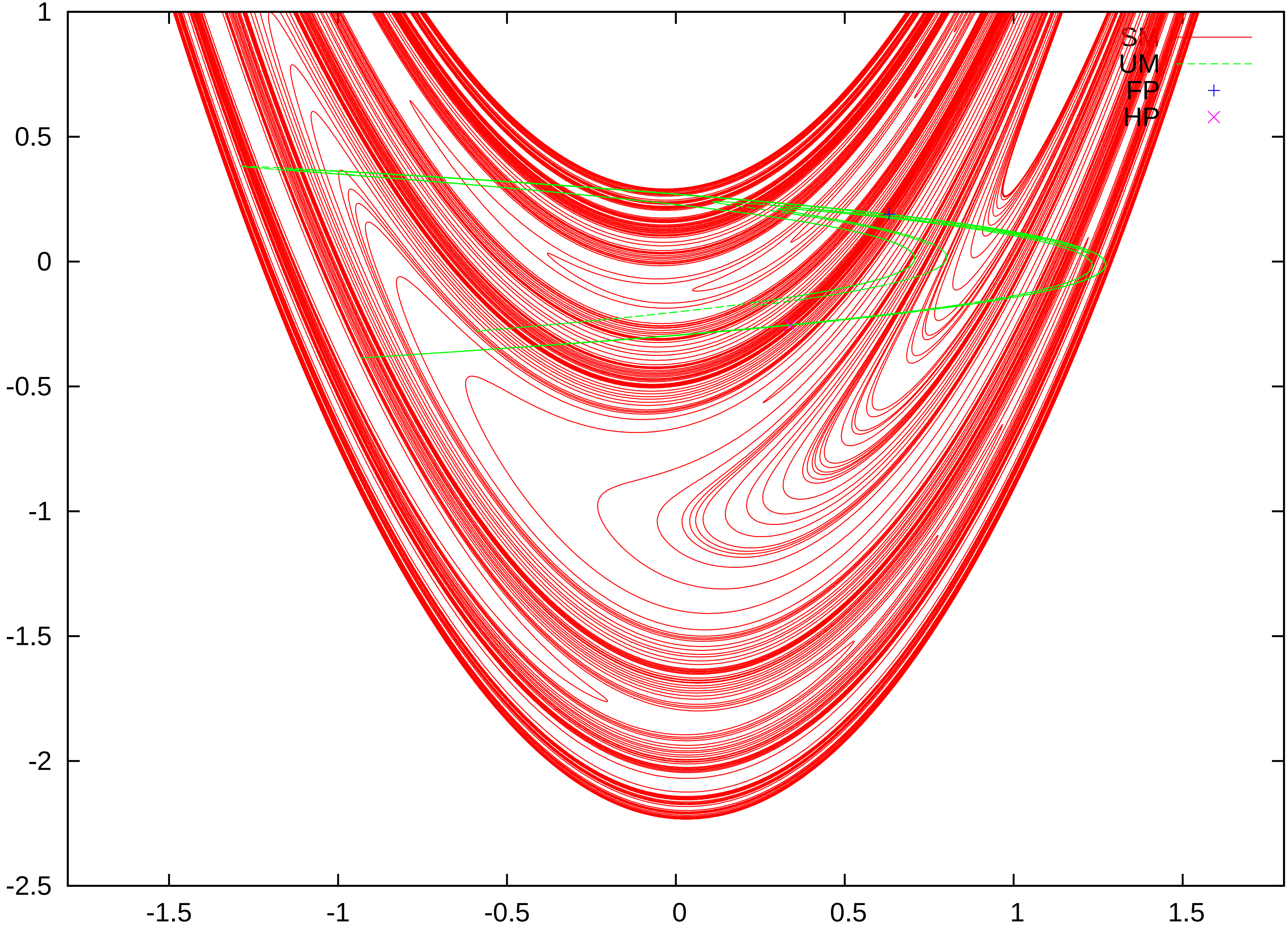
SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2



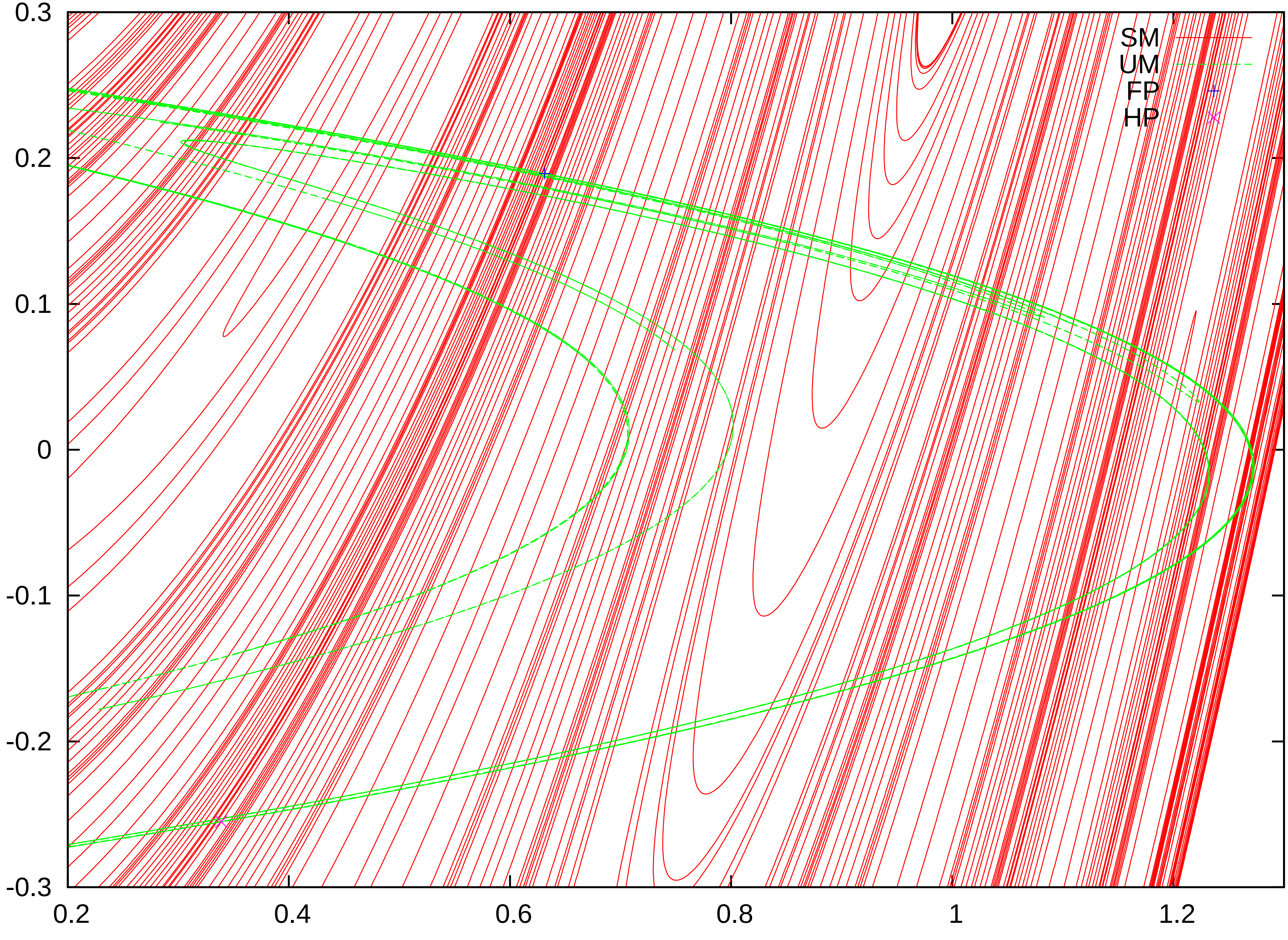
SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2



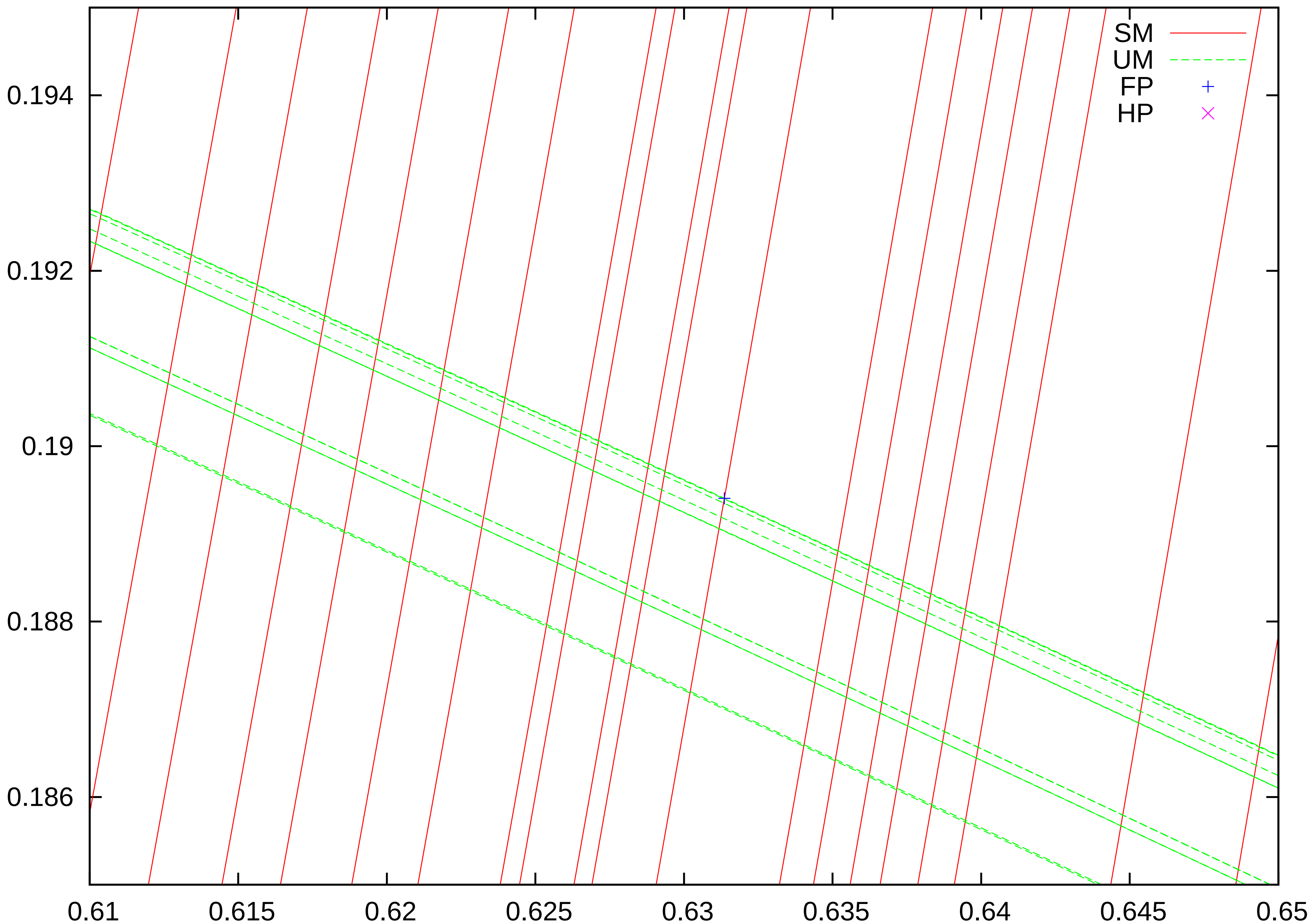
SM
UM
FP
HP
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.2
0.4
0.6
0.8
1
1.2



SM
UM
FP
HP
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


Fourth International Workshop
on Taylor Methods
Boca Raton, Florida
December 16-19, 2006
http://bt.pa.msu.edu/TM/BocaRaton2006/
Topics:
High-Order Methods
Verification & Taylor Models
Automatic Differentiation
Differential Algebraic Tools
and their use for:
ODE and PDE Solvers
Global Optimization
Constraint Satisfaction
Dynamical Systems
Beam Physics
Support: Department of Energy
Michigan State University