Reconstructing metabolic networks using interval analysis

Warwick Tucker

Department of Mathematics

Uppsala University

warwick@math.uu.se

In collaboration with

Vincent Moulton and Zoltán Kutalik

School of Computing Sciences

University of East Anglia, UK

Main topic: Parameter reconstruction for ODE:s.

- “Easy” case
  - Noise-free data
  - Dense data sets
  - Low-dimensional models
  - Fixed topology
- “Hard” case
  - Contaminated data
  - Sparse data sets
  - High-dimensional models
  - Unknown topology

Problem formulation

Given:

- A system of ODE:s that depend on a vector of parameters $p \in \mathcal{P} \subset \mathbb{R}^k$:

$$\dot{x} = f(x; p) \qquad (x \in \mathbb{R}^d, \quad p \in \mathcal{P}).$$

  [A more general setting: $\dot{x} = f(x, t, u; p)$.]

- A data set $\{t_j, x(t_j)\}_{j=0}^{N}$ of samples from the *target system* $\dot{x} = f(x; p^\star)$.

Wanted:

- The *best-fit* parameter $p^\sharp$ (hopefully $p^\sharp \approx p^\star$), or the set of all *consistent* parameters.
- The *topology* of the model (i.e., the non-zero components of p).

A small test-problem

Example: Take $\dot{x} = x^2 - pt$, with $\mathcal{P} = [1, 4]$.

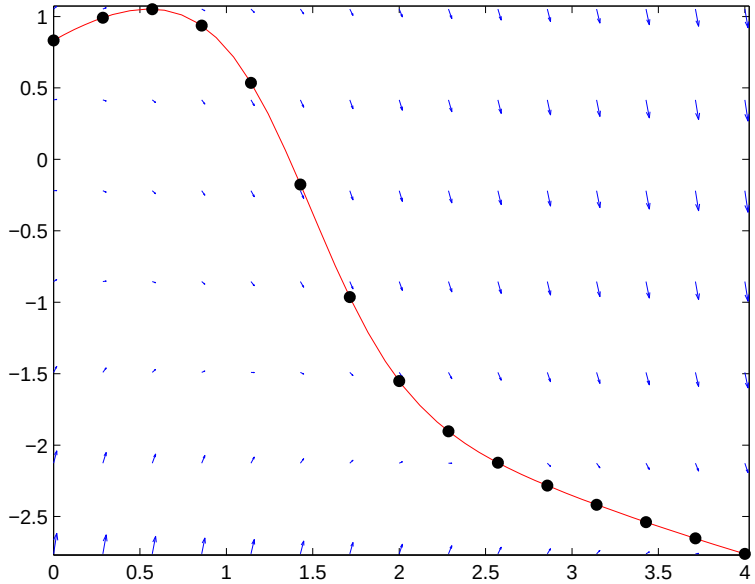
Here $p^\star = 2$ and $x(0) \approx 0.8318$.

The standard LS approach is to minimize some *data defect*:

$$\mathcal{E}_d^1 = \min_{p \in \mathcal{P}} \sum_{j=0}^{N} |\varphi(x(t_0), t_j; p) - x(t_j)|^2$$

$$\mathcal{E}_d^2 = \min_{p \in \mathcal{P}} \sum_{j=0}^{N-1} |\varphi(x(t_j), \Delta t_j; p) - x(t_{j+1})|^2$$

which recasts the problem as global optimization.



Problem: If we instead consider the (same) ODE, reformulated as $\dot{x} = x^2 - (p_1^2 + p_2^2)t$, with $\mathcal{P} = [1, 2]^2$, then there is no longer a single best fit. The set of consistent parameters is an entire circle. Set-valued solutions sets.

GMA- and S-systems

A *generalized mass action* model is a system of ODE:s on the form

$$\dot{x}_i = \sum_{j=1}^{M_i} a_{ij} \prod_{k=1}^{d} x_k^{g_{ijk}} \qquad (i = 1, \ldots, d). \tag{1}$$

$x_i \sim$ concentration of some reactant.

$a_{ij} \sim$ *rate constant*.

$g_{ijk} \sim$ *kinetic order*.

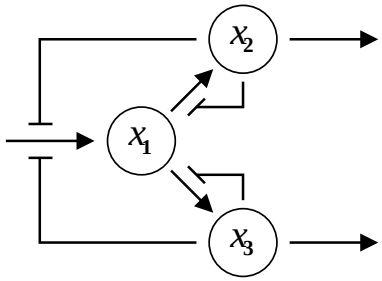
An *S-system* can be considered as a condensed version of a GMA-system, obtained by aggregating individual reactions into the net processes of synthesis and degradation:

$$\dot{x}_i = \alpha_i \prod_{k=1}^{d} x_k^{g_{ik}} - \beta_i \prod_{k=1}^{d} x_k^{h_{ik}} \qquad (i = 1, \ldots, d).$$

Here, α and β assume non-negative values, whereas g_{ik} and h_{ik} are real.

Two small(ish) examples

A 3-dimensional GMA-system:

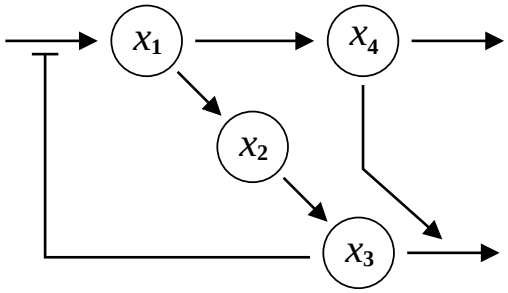


(a) A 3-d branched pathway.

$$\dot{x}_1 = 0.4x_2^{-1}x_3^{-1} - 3x_1^{0.5}x_2^{-0.1} - 2x_1^{0.75}x_3^{-0.2}$$
$$\dot{x}_2 = 3x_1^{0.5}x_2^{-0.1} - 1.5x_2^{0.5}$$
$$\dot{x}_3 = 2x_1^{0.75}x_3^{-0.2} - 5x_3^{0.5}$$

(b) The corresponding GMA-system

A 4-dimensional S-system



(a) A 4-d branched pathway.

$$\dot{x}_1 = 12x_3^{-0.8} - 10x_1^{0.5}$$
$$\dot{x}_2 = 8x_1^{0.5} - 3x_2^{0.75}$$
$$\dot{x}_3 = 3x_2^{0.75} - 5x_3^{0.5}x_4^{0.2}$$
$$\dot{x}_4 = 2x_1^{0.5} - 6x_4^{0.8}.$$

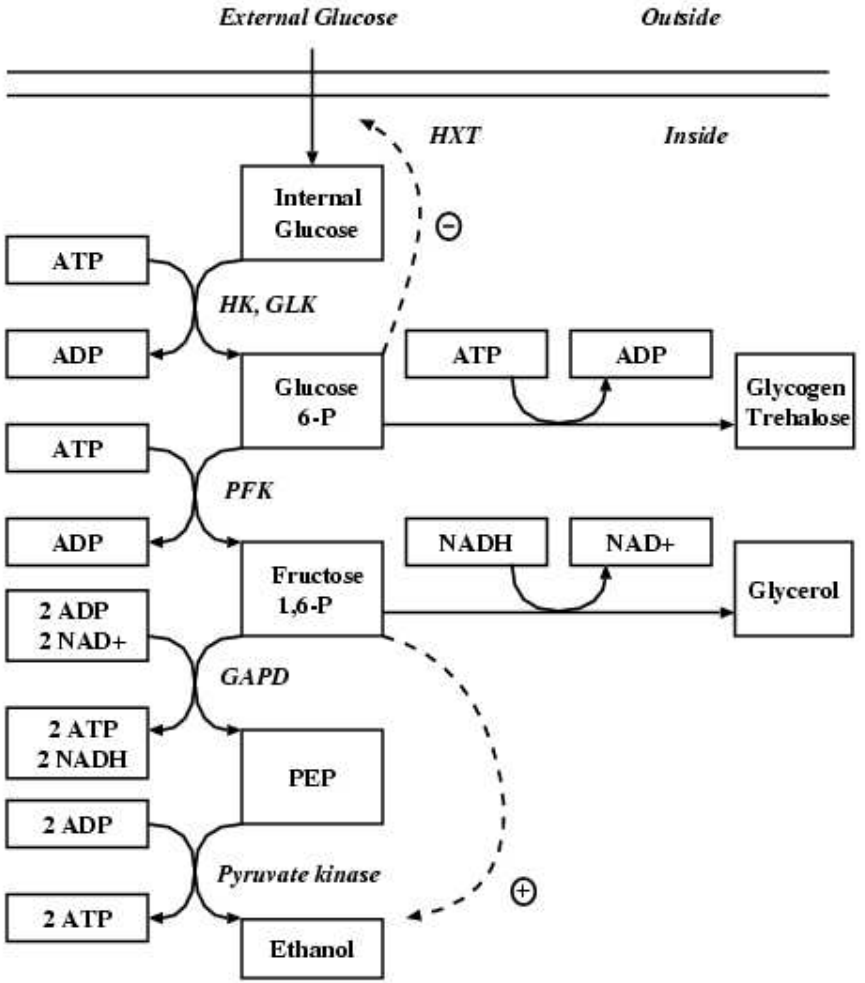
(b) The corresponding S-system.

A larger example

Example: (Metabolic engineering)

Consider the (simplified) model of anaerobic fermentation of glucose to ethanol, glycerol, and polysaccharides in *Saccharomyces cerevisiae* (common baking yeast).

By manipulating the external glucose concentration, changes in the dependent variables are forced as glucose is absorbed into the cell at different rates.



Using stoichiometric information, we can build the systems' corresponding GMA equations:

$$
\begin{aligned}
\dot{x}_1 &= 0.8122x_2^{-0.2344}x_6 - 2.8632x_1^{0.7464}x_5^{0.0243}x_7 \\
\dot{x}_2 &= 2.8632x_1^{0.7464}x_5^{0.0243}x_7 - 0.5232x_2^{0.7318}x_5^{-0.3941}x_8 - 0.0009x_2^{8.6107}x_{11} \\
\dot{x}_3 &= 0.5232x_2^{0.7318}x_5^{-0.3941}x_8 - 0.011x_3^{0.6159}x_5^{0.1308}x_9x_{14}^{-0.6088} - 0.04725x_3^{0.05}x_4^{0.533}x_{12} \\
\dot{x}_4 &= 0.022x_3^{0.6159}x_5^{0.1308}x_9x_{14}^{-0.6088} - 0.0945x_3^{0.05}x_4^{0.533}x_{12} \\
\dot{x}_5 &= 0.022x_3^{0.6159}x_5^{0.1308}x_9x_{14}^{-0.6088} + 0.0945x_3^{0.05}x_4^{0.533}x_{12} - 2.8632x_1^{0.7464}x_5^{0.0243}x_7 \\
&\quad -0.0009x_2^{8.6107}x_{11} - 0.5232x_2^{0.7318}x_5^{-0.3941}x_8 - x_5x_{13}
\end{aligned}
$$

Here, we *dependent* variables are $x_1 \sim \mathrm{G_{in}}$, $x_2 \sim \mathrm{G6P}$, $x_3 \sim \mathrm{FDP}$, $x_4 \sim \mathrm{PEP}$, and $x_5 \sim \mathrm{ATP}$. The *independent* variables are $x_6 \sim \mathrm{HXT}$, $x_7 \sim \mathrm{HK/GLK}$, $x_8 \sim \mathrm{PFK}$, $x_9 \sim \mathrm{GAPD}$, $x_{10} \sim \mathrm{pyruvate-kinase}$, $x_{11} \sim \mathrm{G/T}$,$x_{12} \sim \mathrm{G}$, $x_{13} \sim \mathrm{ATPase}$, and $x_{14} \sim \mathrm{NADH/NAD+}$.

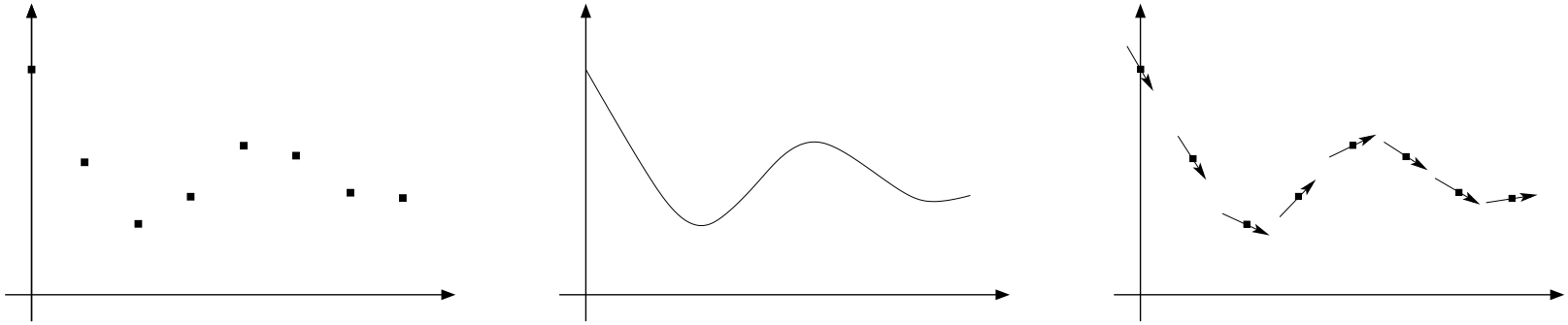
Using stoichiometric information, we can build the systems' corresponding GMA equations:

$$
\begin{aligned}
\dot{x}_1 &= 0.8122x_2^{-0.2344}x_6 - 2.8632x_1^{0.7464}x_5^{0.0243}x_7 \\
\dot{x}_2 &= 2.8632x_1^{0.7464}x_5^{0.0243}x_7 - 0.5232x_2^{0.7318}x_5^{-0.3941}x_8 - 0.0009x_2^{8.6107}x_{11} \\
\dot{x}_3 &= 0.5232x_2^{0.7318}x_5^{-0.3941}x_8 - 0.011x_3^{0.6159}x_5^{0.1308}x_9x_{14}^{-0.6088} - 0.04725x_3^{0.05}x_4^{0.533}x_{12} \\
\dot{x}_4 &= 0.022x_3^{0.6159}x_5^{0.1308}x_9x_{14}^{-0.6088} - 0.0945x_3^{0.05}x_4^{0.533}x_{12} \\
\dot{x}_5 &= 0.022x_3^{0.6159}x_5^{0.1308}x_9x_{14}^{-0.6088} + 0.0945x_3^{0.05}x_4^{0.533}x_{12} - 2.8632x_1^{0.7464}x_5^{0.0243}x_7 \\
&\quad -0.0009x_2^{8.6107}x_{11} - 0.5232x_2^{0.7318}x_5^{-0.3941}x_8 - x_5x_{13}
\end{aligned}
$$

Here, we *dependent* variables are $x_1 \sim \mathrm{G_{in}}$, $x_2 \sim \mathrm{G6P}$, $x_3 \sim \mathrm{FDP}$, $x_4 \sim \mathrm{PEP}$, and $x_5 \sim \mathrm{ATP}$. The *independent* variables are $x_6 \sim \mathrm{HXT}$, $x_7 \sim \mathrm{HK/GLK}$, $x_8 \sim \mathrm{PFK}$, $x_9 \sim \mathrm{GAPD}$, $x_{10} \sim \mathrm{pyruvate-kinase}$, $x_{11} \sim \mathrm{G/T}$,$x_{12} \sim \mathrm{G}$, $x_{13} \sim \mathrm{ATPase}$, and $x_{14} \sim \mathrm{NADH/NAD+}$.

Trajectory reconstruction - decoupling

Given a sufficiently dense data set, we can reconstruct good approximations of the full trajectories using e.g. piece-wise splines.



This enables us to approximate the slopes at each data point:

$$s_{ij} \approx f_i(x(t_j); p^\star) \qquad (i = 1, \ldots, d; \quad j = 0, \ldots, N).$$

With the *enhanced* data set $\{t_j, x(t_j), s(t_j)\}_{j=0}^{N}$, we can minimize the *slope defect*:

$$\mathcal{E}_s = \min_{p \in \mathcal{P}} \sum_{i=1}^{d} \sum_{j=0}^{N} |s_{ij} - f_i(x(t_j); p)|^2 = \sum_{i=1}^{d} \min_{p_i \in \mathcal{P}_i} \sum_{j=0}^{N} |s_{ij} - f_i(x(t_j); p_i)|^2.$$

For GMA-systems, the problem decouples: $\mathcal{E}_s = \mathcal{E}_s^{(1)} + \cdots + \mathcal{E}_s^{(d)}$.

Interval-valued slopes

Slope enclosures: Let F be an *interval extension* of the vector field f, and let $\mathbb{P}$ denote a box in $\mathcal{P}_i$, that is, each component of $\mathbb{P}$ is an interval. Then, for any point $p \in \mathbb{P}$, we have

$$f_i(x(t_j); p) \in F_i(x(t_j); \mathbb{P}),$$

Strategy: Adaptively partition $\mathcal{P}_i$ and discard all parameter boxes $\mathbb{P}$ for which an inclusion fails for some i and j:

$$s_{ij} \notin F_i(x(t_j); \mathbb{P})$$

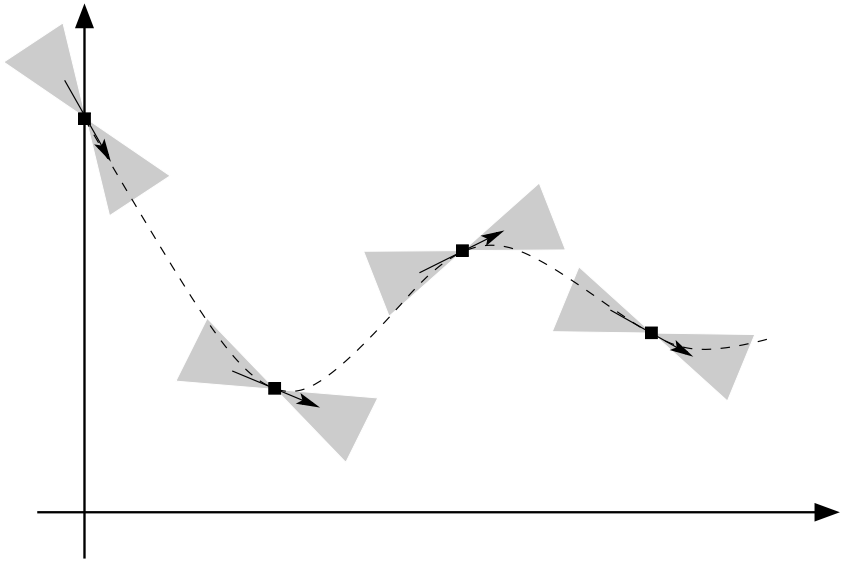
In other words, we only keep parameter boxes for which the boolean function

$$\mathcal{I}_i(\mathbb{P}) \stackrel{\text{def}}{=} \bigwedge_{j=0}^{N} \Big(s_{ij} \in F_i(x(t_j); \mathbb{P}) \Big)$$

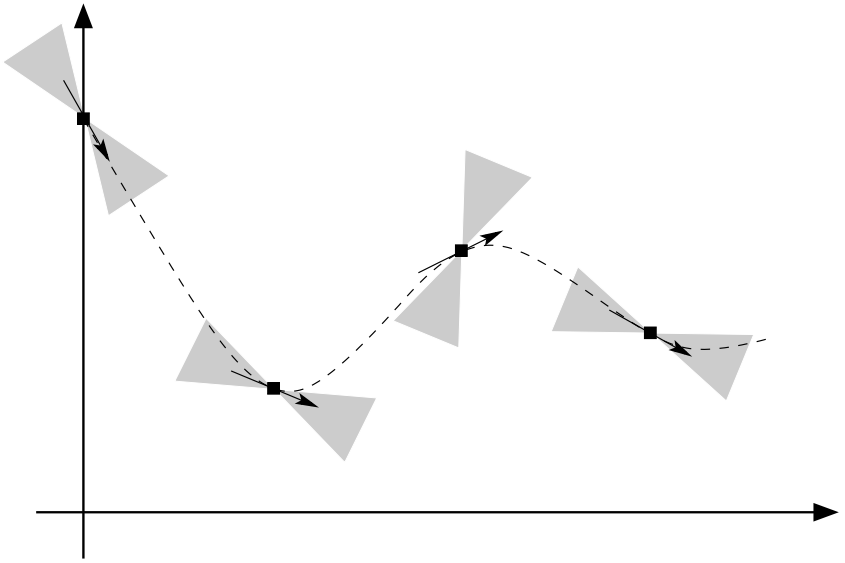
evaluates to true. This is our *cone condition*.

The cone condition

The cone condition ensures that the retained parameters are *consistent* with the reconstructed slopes.



(a) Cone condition satisfied at t_0, t_1, t_2, and t_3.



(b) Violated at time t_2.

Parameters that violate a cone condition are *inconsistent* with our data.

Interval constraint propagation

Basic idea: Turn the equations inside out.

$$\text{relations} \Longrightarrow \text{constraints} \Longrightarrow \text{contractors.}$$

Example: Consider the equation $y = x_1^3 + x_2$, which can be recast as $x_1 = \sqrt[3]{y - x_2}$ and $x_2 = y - x_1^3$. Given a search domain $\mathbb{X}_1 \times \mathbb{X}_2$, we can impose the constraints $x_1 \in \mathbb{X}_1 \cap \sqrt[3]{y - \mathbb{X}_2}$ and $x_2 \in \mathbb{X}_2 \cap (y - \mathbb{X}_1^3)$ on all solutions.

Taking $y = 2$, and looking for a solution $(x_1, x_2) \in [0,1] \times [0,1]$, the constraints produce

$$x_1 \in \mathbb{X}_1 \cap \sqrt[3]{y - \mathbb{X}_2} = [0,1] \cap \sqrt[3]{2 - [0,1]} = [0,1] \cap \sqrt[3]{[1,2]} = [0,1] \cap [1, \sqrt[3]{2}] = \{1\}$$

$$x_2 \in \mathbb{X}_2 \cap (y - \mathbb{X}_1^3) = [0,1] \cap (2 - [0,1]^3) = [0,1] \cap (2 - [0,1]) = [0,1] \cap [1,2] = \{1\}$$

which actually happens to give the (unique) solution within the domain.

Constraints for GMA:s

Recall that a component of a GMA system has the form

$$\dot{x} = \sum_{i=1}^{M} a_i \prod_{j=1}^{d} x_j^{g_{ij}}.$$

Given the search domains $a_i \in \mathbb{A}_i$ and $g_{ij} \in \mathbb{G}_{ij}$, the set-valued constraints for the rate constants become

$$a_k \in \mathcal{A}_k(\dot{x}, x, \mathbb{A}, \mathbb{G}) \stackrel{\text{def}}{=} \mathbb{A}_k \cap \left(\left(\dot{x} - \sum_{\substack{i=1 \\ i \neq k}}^{M} \mathbb{A}_i \prod_{j=1}^{d} x_j^{\mathbb{G}_{ij}} \right) \Big/ \prod_{j=1}^{d} x_j^{\mathbb{G}_{kj}} \right),$$

whereas, for the kinetic orders, we have

$$g_{k\ell} \in \mathcal{G}_{k\ell}(\dot{x}, x, \mathbb{A}, \mathbb{G}) \stackrel{\text{def}}{=} \mathbb{G}_{k\ell} \cap \log \left(\left(\dot{x} - \sum_{\substack{i=1 \\ i \neq k}}^{M} \mathbb{A}_i \prod_{j=1}^{d} x_j^{\mathbb{G}_{ij}} \right) \Big/ \left(\mathbb{A}_k \prod_{\substack{j=1 \\ j \neq \ell}}^{d} x_j^{\mathbb{G}_{kj}} \right) \right) \Big/ \log x_\ell.$$

Computational results

– Time-series data $\{t_j, x(t_j)\}_{j=0}^{N}$ generated via the MATLAB ode45 solver.

– Sample times $t_0, \ldots, t_N$ are non-uniformly distributed.

– Trajectories fitted by MATLABs piece-wise cubic splines.

– Data supplemented with slopes $\{s(t_j)\}_{j=0}^{N}$ obtained by differentiating the splines.

– Parameter estimation was carried out in C++, utilizing the PROFIL/BIAS package.

– Extended definitions of $\log$ and $\div$.

– Computed on a single 1200MHz Intel Pentium M processor with 384MB of RAM.

A 4-dimensional S-system

Assumptions: known topology, but unknown dependencies.

Search region: $\alpha_i, \beta_i \in [0, 20]$, and $g_{ij}, h_{ij} \in [-1, 1]$.

Data info: Samples = 30, Initial conditions = 5.

$$\begin{aligned}
\dot{x}_1 &= 12x_3^{-0.8} - 10x_1^{0.5} \\
\dot{x}_2 &= 8x_1^{0.5} - 3x_2^{0.75} \\
\dot{x}_3 &= 3x_2^{0.75} - 5x_3^{0.5}x_4^{0.2} \\
\dot{x}_4 &= 2x_1^{0.5} - 6x_4^{0.8}.
\end{aligned}$$

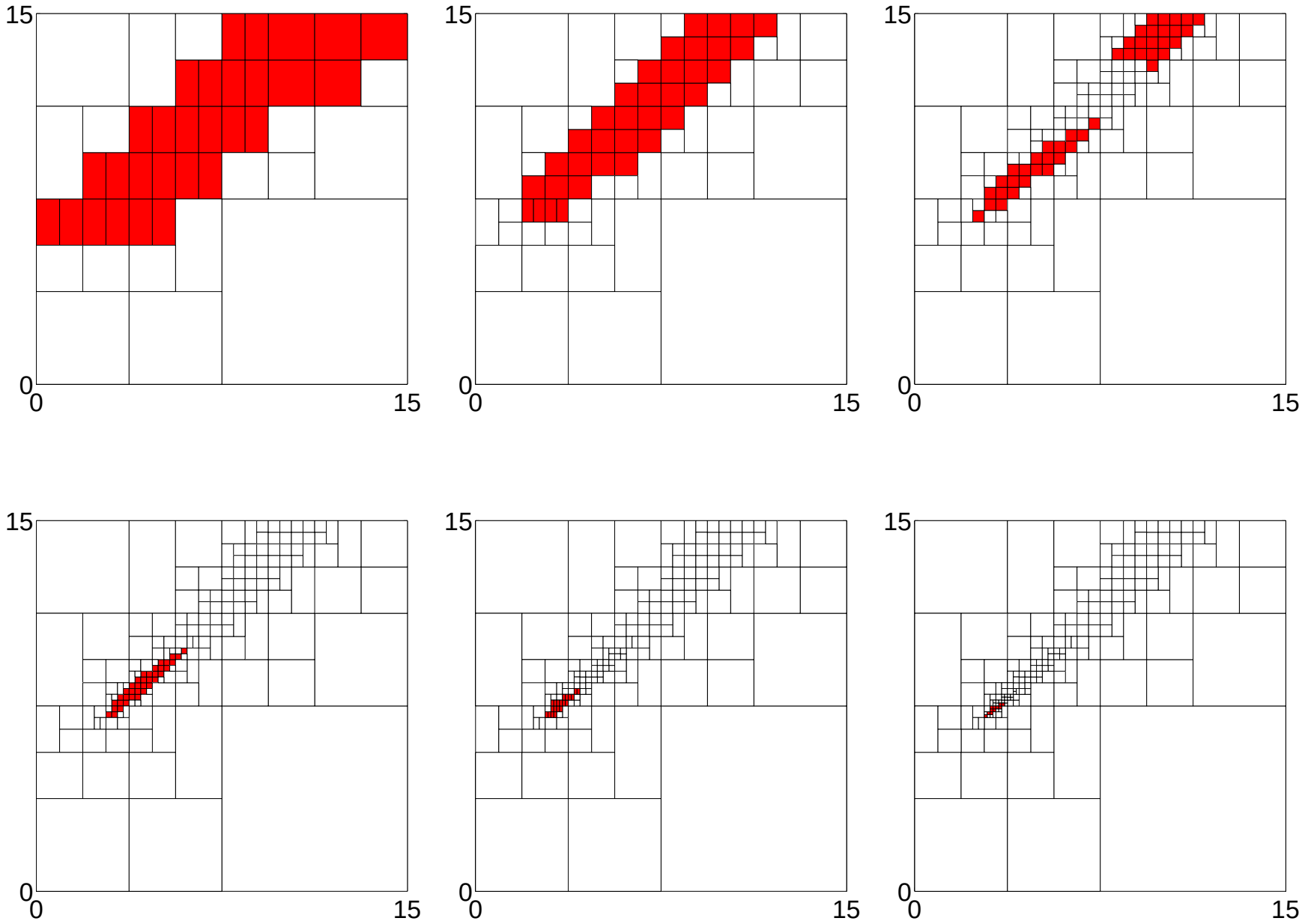
- Using the cone condition
  - Tolerance = 0.05
  - Examined boxes: 11547+ 4879 + 55335 + 5467 = 77228.
  - Elapsed run time: 11 + 4 + 52 + 5 = 1m 12 seconds.
- Using the constraints
  - Tolerance = 0.125
  - Examined boxes: 99 + 81 + 473 + 155 = 708.
  - Elapsed run time: 1 + 0 + 3 + 1 = 5 seconds.

Using the cone condition

$$
\begin{array}{rcll}
 & --\mathbf{comp1}-- & & \\
\mathbf{a}(1) & = & +1.213\mathbf{e}+01 & \mathbf{Diam} = +2.500\mathbf{e}+00 \\
\mathbf{g}(1,3) & = & -7.934\mathbf{e}-01 & \mathbf{Diam} = +1.562\mathbf{e}-01 \\
\mathbf{a}(2) & = & -1.012\mathbf{e}+01 & \mathbf{Diam} = +2.188\mathbf{e}-00 \\
\mathbf{g}(2,1) & = & +4.957\mathbf{e}-01 & \mathbf{Diam} = +1.250\mathbf{e}-01 \\
 & --\mathbf{comp2}-- & & \\
\mathbf{a}(1) & = & +8.010\mathbf{e}+00 & \mathbf{Diam} = +5.469\mathbf{e}-01 \\
\mathbf{g}(1,1) & = & +4.912\mathbf{e}-01 & \mathbf{Diam} = +6.250\mathbf{e}-02 \\
\mathbf{a}(2) & = & -3.017\mathbf{e}+00 & \mathbf{Diam} = +5.469\mathbf{e}-01 \\
\mathbf{g}(2,2) & = & +7.480\mathbf{e}-01 & \mathbf{Diam} = +1.250\mathbf{e}-01 \\
 & --\mathbf{comp3}-- & & \\
\mathbf{a}(1) & = & +3.074\mathbf{e}+00 & \mathbf{Diam} = +1.367\mathbf{e}+00 \\
\mathbf{g}(1,2) & = & +7.398\mathbf{e}-01 & \mathbf{Diam} = +2.812\mathbf{e}-01 \\
\mathbf{a}(2) & = & -5.076\mathbf{e}+01 & \mathbf{Diam} = +1.367\mathbf{e}+00 \\
\mathbf{g}(2,3) & = & +4.935\mathbf{e}-01 & \mathbf{Diam} = +1.562\mathbf{e}-01 \\
\mathbf{g}(2,4) & = & +1.981\mathbf{e}-01 & \mathbf{Diam} = +9.375\mathbf{e}-02 \\
 & --\mathbf{comp4}-- & & \\
\mathbf{a}(1) & = & +1.993\mathbf{e}+00 & \mathbf{Diam} = +7.812\mathbf{e}-01 \\
\mathbf{g}(1,1) & = & +5.017\mathbf{e}-01 & \mathbf{Diam} = +1.562\mathbf{e}-01 \\
\mathbf{a}(2) & = & -5.992\mathbf{e}+00 & \mathbf{Diam} = +7.812\mathbf{e}-01 \\
\mathbf{g}(2,4) & = & +8.019\mathbf{e}-01 & \mathbf{Diam} = +1.562\mathbf{e}-01
\end{array}
$$

Using the constraints

	$--\mathbf{comp1}--$		
$\mathbf{a}(1)$	$=$	$+1.210\mathbf{e}+01$	$\mathbf{Diam}=+1.123\mathbf{e}+00$
$\mathbf{g}(1,3)$	$=$	$-7.958\mathbf{e}-01$	$\mathbf{Diam}=+5.894\mathbf{e}-02$
$\mathbf{a}(2)$	$=$	$-1.008\mathbf{e}+01$	$\mathbf{Diam}=+9.078\mathbf{e}-01$
$\mathbf{g}(2,1)$	$=$	$+4.965\mathbf{e}-01$	$\mathbf{Diam}=+4.120\mathbf{e}-02$
	$--\mathbf{comp2}--$		
$\mathbf{a}(1)$	$=$	$+7.989\mathbf{e}+00$	$\mathbf{Diam}=+8.046\mathbf{e}-01$
$\mathbf{g}(1,1)$	$=$	$+5.004\mathbf{e}-01$	$\mathbf{Diam}=+6.193\mathbf{e}-02$
$\mathbf{a}(2)$	$=$	$-2.989\mathbf{e}+00$	$\mathbf{Diam}=+7.008\mathbf{e}-01$
$\mathbf{g}(2,2)$	$=$	$+7.522\mathbf{e}-01$	$\mathbf{Diam}=+1.339\mathbf{e}-01$
	$--\mathbf{comp3}--$		
$\mathbf{a}(1)$	$=$	$+3.046\mathbf{e}+00$	$\mathbf{Diam}=+1.248\mathbf{e}+00$
$\mathbf{g}(1,2)$	$=$	$+7.456\mathbf{e}-01$	$\mathbf{Diam}=+2.345\mathbf{e}-01$
$\mathbf{a}(2)$	$=$	$-5.048\mathbf{e}+00$	$\mathbf{Diam}=+1.198\mathbf{e}+00$
$\mathbf{g}(2,3)$	$=$	$+4.969\mathbf{e}-01$	$\mathbf{Diam}=+1.779\mathbf{e}-01$
$\mathbf{g}(2,4)$	$=$	$+1.990\mathbf{e}-01$	$\mathbf{Diam}=+1.231\mathbf{e}-01$
	$--\mathbf{comp4}--$		
$\mathbf{a}(1)$	$=$	$+2.031\mathbf{e}+00$	$\mathbf{Diam}=+8.492\mathbf{e}-01$
$\mathbf{g}(1,1)$	$=$	$+4.971\mathbf{e}-01$	$\mathbf{Diam}=+1.635\mathbf{e}-01$
$\mathbf{a}(2)$	$=$	$-6.030\mathbf{e}+00$	$\mathbf{Diam}=+8.972\mathbf{e}-01$
$\mathbf{g}(2,4)$	$=$	$+7.962\mathbf{e}-01$	$\mathbf{Diam}=+1.781\mathbf{e}-01$


15
0
0
15
15
0
0
15
15
0
0
15
15
0
0
15
15
0
0
15
15
0
0
15

Unknown topology

Now, the general component is:

$$\dot{x}_i = \alpha_i x_1^{g_{i1}} x_2^{g_{i2}} x_3^{g_{i3}} x_4^{g_{i4}} - \beta_i x_1^{h_{i1}} x_2^{h_{i2}} x_3^{h_{i3}} x_4^{h_{i4}} \qquad i = 1, \ldots, 4.$$

We start by examining sparse topologies, and work our way up.

Assumptions: As before, but now with tolerance = 0.05.

```
component: 1
|resultList| = 8
Hull of the resulting parameters:
a    = [+1.173e+01,+1.235e+01]  b    = [+9.775e+00,+1.028e+01]
g(1) = [-0.000e+00,+0.000e+00]  h(1) = [+4.873e-01,+5.100e-01]
g(2) = [-0.000e+00,+0.000e+00]  h(2) = [-0.000e+00,+0.000e+00]
g(3) = [-8.158e-01,-7.818e-01]  h(3) = [-0.000e+00,+0.000e+00]
g(4) = [-0.000e+00,+0.000e+00]  h(4) = [-0.000e+00,+0.000e+00]
Average of the resulting parameters:
a    = +1.202e+01       b    = +1.002e+01
g(1) = +0.000e+00       h(1) = +4.990e-01
g(2) = +0.000e+00       h(2) = +0.000e+00
g(3) = -7.990e-01       h(3) = +0.000e+00
g(4) = +0.000e+00       h(4) = +0.000e+00
```

```
component: 2
|resultList| = 3
Hull of the resulting parameters:
a    = [+8.033e+00,+8.385e+00]  b    = [+3.036e+00,+3.333e+00]
g(1) = [+4.735e-01,+4.977e-01]  h(1) = [-0.000e+00,+0.000e+00]
g(2) = [-0.000e+00,+0.000e+00]  h(2) = [+6.922e-01,+7.432e-01]
g(3) = [-0.000e+00,+0.000e+00]  h(3) = [-0.000e+00,+0.000e+00]
g(4) = [-0.000e+00,+0.000e+00]  h(4) = [-0.000e+00,+0.000e+00]
Average of the resulting parameters:
a    = +8.217e+00       b    = +3.196e+00
g(1) = +4.852e-01       h(1) = +0.000e+00
g(2) = +0.000e+00       h(2) = +7.164e-01
g(3) = +0.000e+00       h(3) = +0.000e+00
g(4) = +0.000e+00       h(4) = +0.000e+00

component: 4
|resultList| = 14
Hull of the resulting parameters:
a    = [+1.866e+00,+2.285e+00]  b    = [+5.862e+00,+6.294e+00]
g(1) = [+4.603e-01,+5.345e-01]  h(1) = [-0.000e+00,+0.000e+00]
g(2) = [-0.000e+00,+0.000e+00]  h(2) = [-0.000e+00,+0.000e+00]
g(3) = [-0.000e+00,+0.000e+00]  h(3) = [-0.000e+00,+0.000e+00]
g(4) = [-0.000e+00,+0.000e+00]  h(4) = [+7.441e-01,+8.330e-01]
Average of the resulting parameters:
a    = +2.058e+00       b    = +6.060e+00
g(1) = +4.930e-01       h(1) = +0.000e+00
g(2) = +0.000e+00       h(2) = +0.000e+00
g(3) = +0.000e+00       h(3) = +0.000e+00
g(4) = +0.000e+00       h(4) = +7.885e-01
```

Multiple solutions are possible

```
component: 3
|resultList| = 1
Average of the resulting parameters:
a    = +1.330e+01       b    = +1.550e+01
g(1) = +0.000e+00       h(1) = +0.000e+00
g(2) = +2.108e-01       h(2) = -1.738e-01
g(3) = -1.368e-01       h(3) = +0.000e+00
g(4) = +0.000e+00       h(4) = +0.000e+00
Elapsed run time: 24 seconds.
|resultList| = 1
Average of the resulting parameters:
a    = +3.214e+00       b    = +5.227e+00
g(1) = +0.000e+00       h(1) = +0.000e+00
g(2) = +6.882e-01       h(2) = +0.000e+00
g(3) = +0.000e+00       h(3) = +4.642e-01
g(4) = -2.025e-01       h(4) = +0.000e+00
Elapsed run time: 34 seconds.
resultList| = 24
Hull of the resulting parameters:
a    = [+2.774e+00,+3.212e+00]  b    = [+4.740e+00,+5.240e+00]
g(1) = [-0.000e+00,+0.000e+00]  h(1) = [-0.000e+00,+0.000e+00]
g(2) = [+7.109e-01,+7.930e-01]  h(2) = [-0.000e+00,+0.000e+00]
g(3) = [-0.000e+00,+0.000e+00]  h(3) = [+4.721e-01,+5.355e-01]
g(4) = [-0.000e+00,+0.000e+00]  h(4) = [+1.803e-01,+2.263e-01]
Average of the resulting parameters:
a    = +3.014e+00       b    = +5.018e+00
g(1) = +0.000e+00       h(1) = +0.000e+00
g(2) = +7.486e-01       h(2) = +0.000e+00
g(3) = +0.000e+00       h(3) = +4.993e-01
g(4) = +0.000e+00       h(4) = +2.001e-01
```

Problems

- The slopes are not perfectly reconstructed
- The constraint propagation notices this almost immediately
- Thus no solutions are reported
- The slower cone condition is more stable

Fixes

- Use constraint propagation initially only [when do we change to the cone condition?]
- Widen the reconstructed slopes and data [noise tolerant!]
- Do not use slopes at all [back to integrating the full system.]

Widened data

How does the set of *feasible* parameters change with respect to measurement errors?

Example: Consider the function Lotka-Volterra system

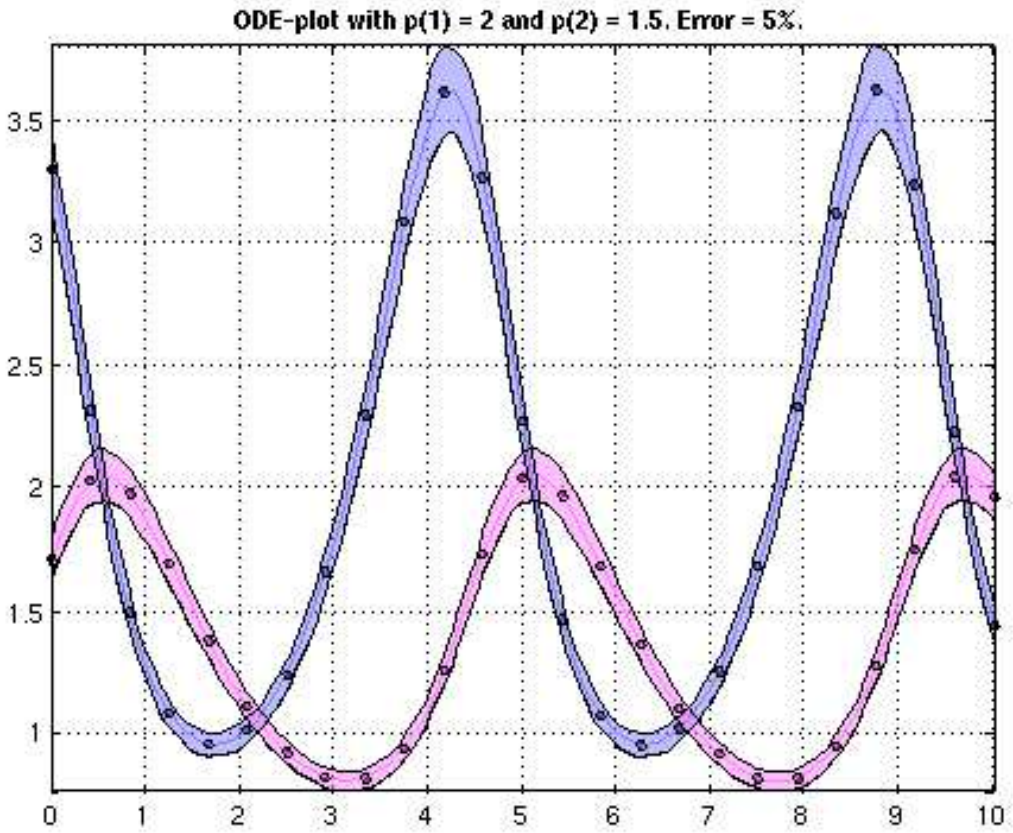
$$\dot{x}_1 = p_1 x_1 - p_2 x_1 x_2$$
$$\dot{x}_2 = -x_2 + 0.5 x_1 x_2$$

Generate data from the target parameter $(p_1, p_2) = (2, 1.5)$ and add 5% relative error.

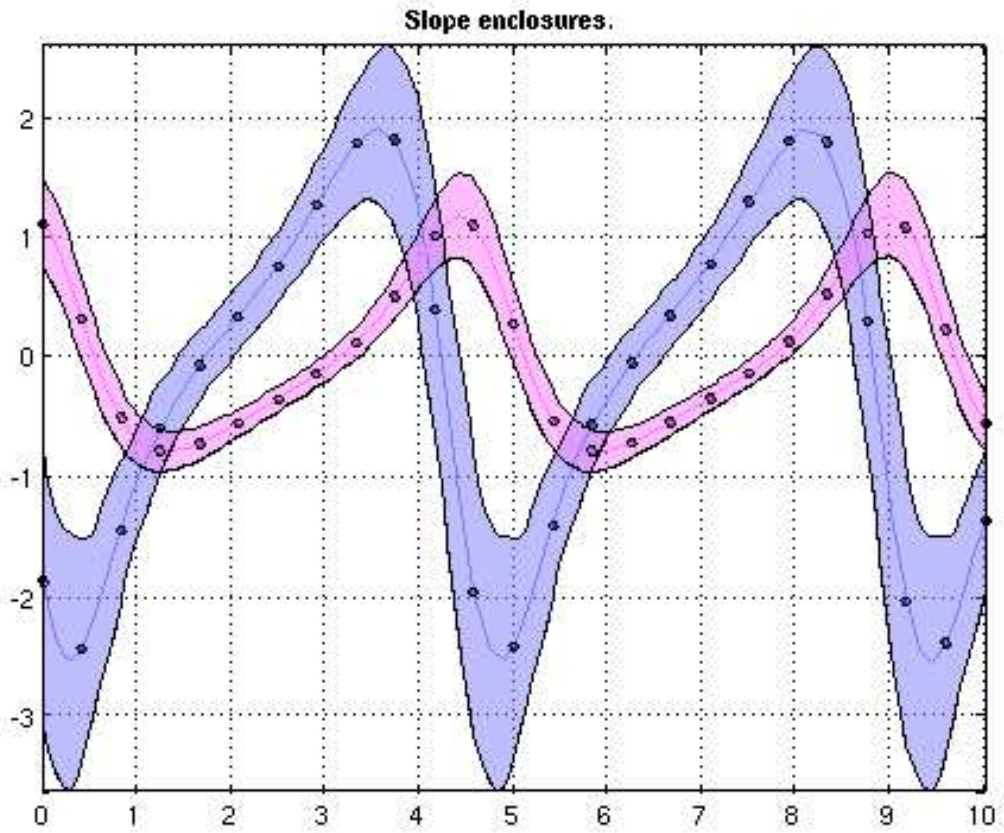
This produces *thick* trajectories, i.e., we consider all solutions that remain inside the enclosures. We also consider ranges of slopes.

The cone condition becomes:

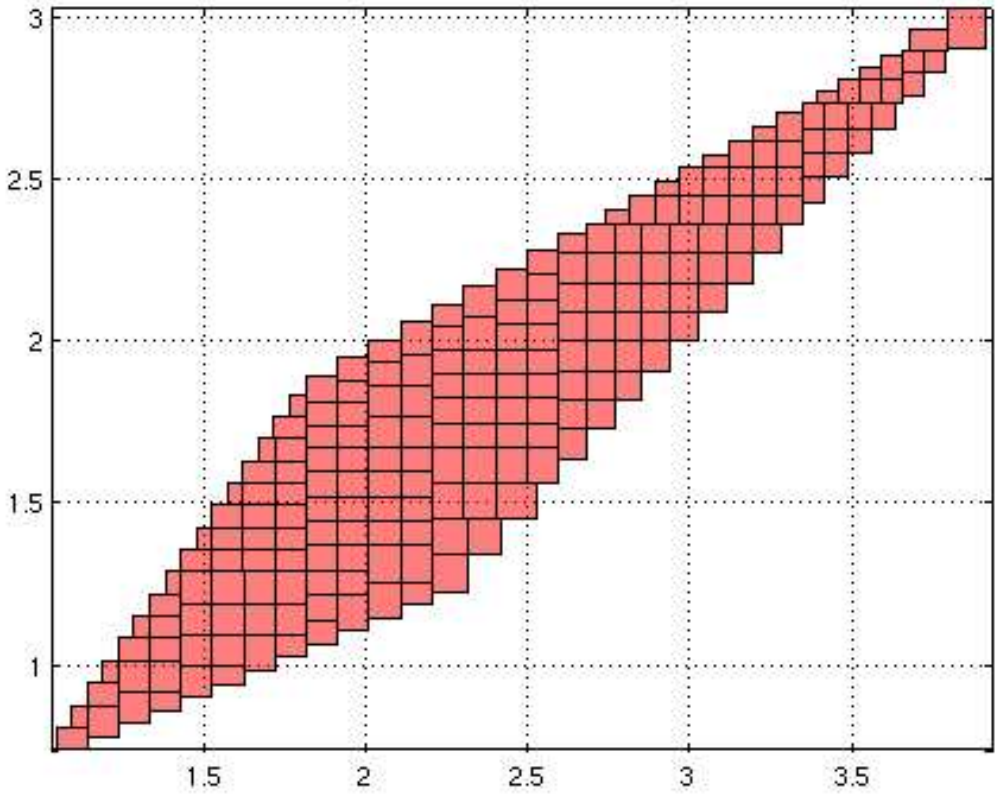
$$\mathcal{I}_i(\mathbb{P}) \stackrel{\text{def}}{=} \bigwedge_{j=0}^{N} \Big(\mathbb{S}_{ij} \cap F_i(\mathbb{X}(t_j); \mathbb{P}) \neq \emptyset \Big)$$



The set-valued trajectories.



The set-valued slopes.



The feasible set of parameters.