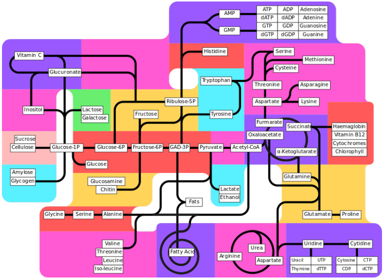
SYMBIONT Project
LPHI, UMR 5235, CNRS, University of Montpellier

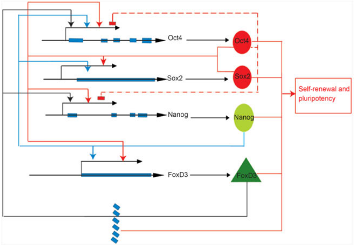
Reduction of Chemical Reaction Networks with Approximate Conservation Laws

Aurélien Desoeuvres, Alexandru Iosif, Ovidiu Radulescu,
Hamid Rahkooy, Matthias Seiß, Thomas Sturm

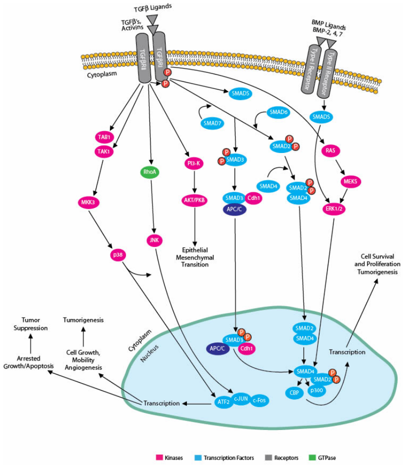
Systems biology : how the cell works



Metabolic networks: synthesis, growth

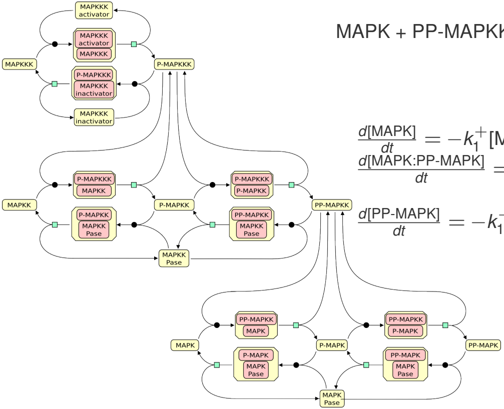


Gene networks: Decision making



Signalling networks: information transfer

Chemical reactions networks



Reactions:

$$\text{MAPK} + \text{PP-MAPKK} \underset{k_1^-}{\overset{k_1^+}{\rightleftharpoons}} \text{MAPK:PP-MAPK} \overset{k_2}{\rightarrow} \text{P-MAPK} + \text{PP-MAPKK}$$

Differential equations:

$$\frac{d[\text{MAPK}]}{dt} = -k_1^+[\text{MAPK}]\,[\text{PP-MAPKK}] + k_1^-[\text{MAPK:PP-MAPK}]$$

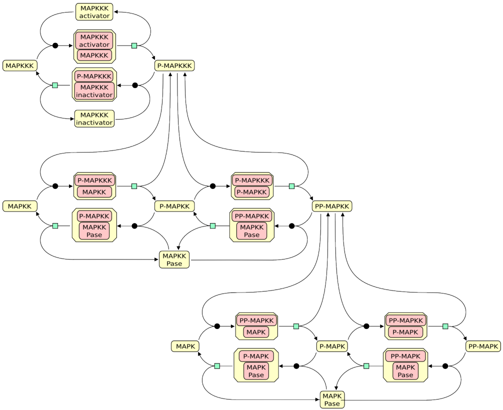
$$\frac{d[\text{MAPK:PP-MAPK}]}{dt} = k_1^+[\text{MAPK}]\,[\text{PP-MAPKK}] - (k_1^- + k_2)[\text{MAPK:PP-MAPK}]$$

$$\frac{d[\text{PP-MAPK}]}{dt} = -k_1^+[\text{MAPK}]\,[\text{PP-MAPKK}] + (k_1^- + k_2)[\text{MAPK:PP-MAPK}]$$

Michaelis-Menten equations

Chemical reactions networks

Chemical kinetics: polynomial ODEs



$x1' = k3*x13 + k4*x13 - k2*x1*x3$

$x2' = k6*x14 + k7*x14 - k5*x2*x4$

$x3' = k3*x13 + k7*x14 - k2*x1*x3$

$x4' = k4*x13 + k6*x14 + k9*x15 + k10*x15 + k15*x16 + k16*x16 - k5*x2*x4 - k8*x4*x5 - k14*x4*x6$

$x5' = k9*x15 + k13*x20 - k8*x4*x5$

$x6' = k10*x15 + k15*x16 + k12*x20 + k19*x19 - k14*x4*x6 - k11*x6*x12$

$x7' = k16*x16 + k18*x19 + k21*x17 + k22*x17 + k27*x18 + k28*x18 - k20*x7*x8 - k17*x7*x12 - k26*x7*x9$

$x8' = k21*x17 + k25*x22 - k20*x7*x8$

$x9' = k22*x17 + k27*x18 + k24*x22 + k31*x21 - k26*x7*x9 - k23*x9*x11$

$x10' = k28*x18 + k30*x21 - k29*x10*x11$

$x11' = k24*x22 + k25*x22 + k30*x21 + k31*x21 - k23*x9*x11 - k29*x10*x11$

$x12' = k12*x20 + k13*x20 + k18*x19 + k19*x19 - k11*x6*x12 - k17*x7*x12$

$x13' = k2*x1*x3 - k4*x13 - k3*x13$

$x14' = k5*x2*x4 - k7*x14 - k6*x14$

$x15' = k8*x4*x5 - k10*x15 - k9*x15$

$x16' = k14*x4*x6 - k16*x16 - k15*x16$

$x17' = k20*x7*x8 - k22*x17 - k21*x17$

$x18' = k26*x7*x9 - k28*x18 - k27*x18$

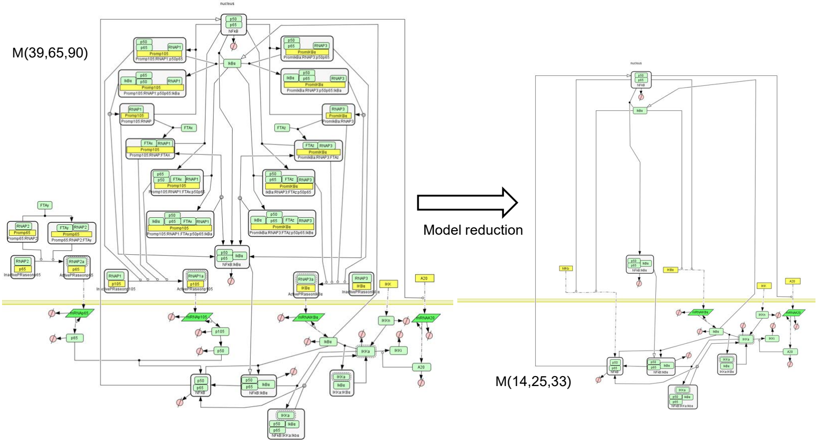
$x19' = k17*x7*x12 - k19*x19 - k18*x19$

$x20' = k11*x6*x12 - k13*x20 - k12*x20$

$x21' = k29*x10*x11 - k31*x21 - k30*x21$

$x22' = k23*x9*x11 - k25*x22 - k24*x22$

Model order reduction



From Radulescu et al 2008

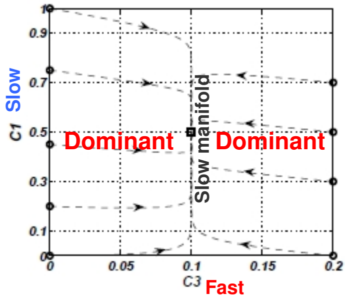
produce models with less variables, parameters and differential equations;

keep only minimum details

Model reduction methods

- Quasi-steady state: Briggs,Haldane (1925); Bodenstein(1913); Semenov, Frank-Kamenetskii(1939); Segel and Slemrod (1989); Radulescu et al (2008); Goeke and Walcher (2013);
- Quasi-equilibrium: Michaelis-Menten (1913); Gorban et al (2001); Radulescu et al (2012);
- Singular perturbations, slow/fast systems: Tikhonov (1952); Hoppenstaedt (1967); Fenichel(1979);
- Singular perturbations, multiple timescales: Hoppenstaedt (1967); O'Malley (1971); Cardin and Teixera (2017);
- Linear networks with totally separated constants: Gorban and Radulescu (2008);
- Singular perturbations and tropical scaling: Noel et al (2012); Samal et al (2015); Radulescu,Grigoriev,Vakulenko (2015); Kruff et al (2021);

Slow/fast systems: quasi-steady state approximation



from Chiavazzo et al Comm.Comp.Phys. 2007

$$\frac{dX}{dt} = \varepsilon^{-1}(F_1(X,Y) + \varepsilon^a F_2(X,Y))$$

$$\frac{dY}{dt} = G_1(X,Y) + \varepsilon^b G_2(X,Y)$$

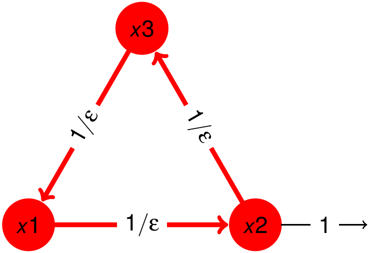
$$\underbrace{\overline{\text{quasi-steady states are hyperbolic}} \quad \Re(Spec(D_X F_1(X,Y))) < 0, F_1(X,Y) = 0}$$

$$\Big\Downarrow \varepsilon \to 0$$

$$\frac{dY}{dt} = G_1(\psi(Y), Y)$$

$X = \psi(Y)$ solution of $F_1(X,Y) = 0$

Approximate conservation laws: quasi-equilibrium



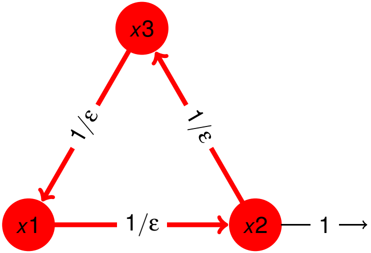
$x_1 + x_2 + x_3$ is an approximate conservation law

$$\frac{dX}{dt} = \varepsilon^{-1}(F_1(X,Y) + \varepsilon^a F_2(X,Y))$$
$$\frac{dY}{dt} = G_1(X,Y) + \varepsilon^b G_2(X,Y)$$

The fast dynamics $\frac{dX}{dt} = \varepsilon^{-1} F_1(X,Y)$ has a conservation law.

$D_X\Phi(X)F_1(X,Y) = 0 \Rightarrow \frac{d\Phi(X)}{dt} = \varepsilon^{-1} D_X\Phi(X)F_1(X,Y) = 0.$

Approximate conservation laws: quasi-equilibrium



$x_1 + x_2 + x_3$ is an approximate conservation law

$$\frac{dX}{dt} = \varepsilon^{-1}(F_1(X,Y) + \varepsilon^a F_2(X,Y))$$
$$\frac{dY}{dt} = G_1(X,Y) + \varepsilon^b G_2(X,Y)$$

The fast dynamics $\frac{dX}{dt} = \varepsilon^{-1} F_1(X,Y)$ has a conservation law.

$D_X\Phi(X)F_1(X,Y) = 0 \Rightarrow \frac{d\Phi(X)}{dt} = \varepsilon^{-1} D_X\Phi(X)F_1(X,Y) = 0.$

$D_{XX}\Phi(X)F_1(X,Y) + D_X\Phi(X)D_X F_1(X,Y) = 0$

$D_X\Phi(X)D_X F_1(X,Y) = 0$ when $F_1(X,Y) = 0$

$|D_X F_1(X,Y)| = 0, F_1(X,Y) = 0$

quasi-steady states are degenerated.

Earlier results on degenerated QSS

- Critical singular perturbations, Boundary series, Two time-scale expansions: Vasil'eva and Butuzov (1973).
- Formal solution using nonlinear conservations: Schneider and Wilhelm (2000).
- Formal solution using linear conservations, heuristic for timescales : Gorban, Radulescu, Zinovyev (2010).

Formal scaling

The model is $S_* := \{\dot{x}_1 = f_1(\boldsymbol{k},\boldsymbol{x}), \ldots, \dot{x}_n = f_n(\boldsymbol{k},\boldsymbol{x})\}$, where

$$f_i(\boldsymbol{k},\boldsymbol{x}) = \sum_{j=1}^{r} S_{ij} k_j \boldsymbol{x}^{\alpha_j} \in \mathbb{Z}[\boldsymbol{k},\boldsymbol{x}] = \mathbb{Z}[k_1,\ldots,k_r,x_1,\ldots,x_n].$$

Rescale $k_i = \bar{k}_i \varepsilon^{e_i}$, $x_k = y_k \varepsilon^{d_k}$,

Obtain $S_\varepsilon = \left\{\dot{y}_i = \sum_{j=1}^{r} \varepsilon^{\psi_{ij}} S_{ij} \bar{k}_j \boldsymbol{y}^{\alpha_j} \mid 1 \leq i \leq n\right\}$, where

$$\psi_{i,j} = e_j + \langle \boldsymbol{d}, \alpha_j \rangle - d_i.$$

Formal scaling

change time $\tau = \varepsilon^{\mu} t$, $\mu = \min\{\psi_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq r, S_{ij} \neq 0\}$
Let $a_{ij} = \psi_{ij} - \mu \geq 0$, $a_i = \min\{a_{ij} \mid 1 \leq j \leq r, S_{ij} \neq 0\} \geq 0$ and $a'_{ij} = a_{ij} - a_i > 0$. We get

$$y'_i = \varepsilon^{a_i} \Big(\sum_{a_{ij} = a_i} S_{ij} \bar{k}_j \boldsymbol{y}^{\alpha_j} + \sum_{a_{ij} \neq a_i} S_{ij} \bar{k}_j \varepsilon^{a'_{ij}} \boldsymbol{y}^{\alpha_j} \Big),$$

Timescale orders: ε^{a_i}, y_i (thus x_i) is fast if a_i is small.
Tropical equilibration condition: for all $i \in \{1, \ldots, n\}$ the set $a_{ij} = a_i$ contains at least one positive and one negative monomial.

Formal scaling

Change parameter $\delta = \varepsilon^{1/o}$ where o is the least common multiple of all denominators of the rational orders.

$$y_i' = \delta^{b_i}(\bar{f}_i^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{y}) + \delta^{b_i'}\bar{f}_i^{(2)}(\bar{\boldsymbol{k}},\boldsymbol{y},\delta)),\ b_i \in \mathbb{Z}_{\geq 0}, b_i' \in \mathbb{Z}_{>0}$$

$$\bar{f}_i^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{y}) = \sum_{a_{ij}=a_i} S_{ij}\bar{k}_j\boldsymbol{y}^{\alpha_j} = \sum_j S_{ij}^{(1)}\bar{k}_j\boldsymbol{y}^{\alpha_j},$$

$$\bar{f}_i^{(2)}(\bar{\boldsymbol{k}},\boldsymbol{y},\delta) = \sum_{a_{ij}\neq a_i} S_{ij}\bar{k}_j\delta^{b_{ij}'}\boldsymbol{y}^{\alpha_j} = \sum_j S_{ij}^{(2)}\bar{k}_j\boldsymbol{y}^{\alpha_j},$$

$$\dot{x}_i = f_i^{(1)}(\boldsymbol{k},\boldsymbol{x}) + f_i^{(2)}(\boldsymbol{k},\boldsymbol{x}),\ f_i^{(1,2)}(\boldsymbol{k},\boldsymbol{x}) = \sum_j S_{ij}^{(1,2)}k_j\boldsymbol{x}^{\alpha_j}$$

Truncated system

$$\dot{\boldsymbol{x}} = \boldsymbol{F}^{(1)}(\boldsymbol{k},\boldsymbol{x}),\ \boldsymbol{F}^{(1)} = (f_1^{(1)},\dots,f_n^{(1)})^{\mathsf{T}}.$$

Approximate conservation laws

- The **truncated system** is $\dot{\boldsymbol{x}} = \boldsymbol{F}^{(1)}(\boldsymbol{k}, \boldsymbol{x})$.
An **approximated conservation law** is a function $\phi(\boldsymbol{x})$ satisfying $D_{\boldsymbol{x}}\phi(\boldsymbol{x})\boldsymbol{F}^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0$, for any $\boldsymbol{x} \in \mathbb{R}^n_{>0}$. If this is true also for any $\boldsymbol{k} \in \mathbb{R}^r_{>0}$ the conservation law is **unconditional on the parameters**.

Approximate conservation laws

• The **truncated system** is $\dot{\boldsymbol{x}} = \boldsymbol{F}^{(1)}(\boldsymbol{k}, \boldsymbol{x})$.
An **approximated conservation law** is a function $\phi(\boldsymbol{x})$ satisfying $D_{\boldsymbol{x}}\phi(\boldsymbol{x})\boldsymbol{F}^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0$, for any $\boldsymbol{x} \in \mathbb{R}^n_{>0}$. If this is true also for any $\boldsymbol{k} \in \mathbb{R}^r_{>0}$ the conservation law is **unconditional on the parameters**.

• A set

$$\Phi(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_s(\boldsymbol{x}))^{\mathsf{T}}$$

of approximate conservation laws is called **complete** if the Jacobian matrix

$$\boldsymbol{J}_{\boldsymbol{F}^{(1)},\phi}(\boldsymbol{k}, \boldsymbol{x}) = D_{\boldsymbol{x}}(\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}), \Phi(\boldsymbol{x}))^{\mathsf{T}}$$

has rank n for any $\boldsymbol{k} \in \mathbb{R}^r_{>0}, \boldsymbol{x} \in \mathbb{R}^n_{>0}$ satisfying $\boldsymbol{F}^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0$.

Approximate conservation laws

- The **truncated system** is $\dot{\boldsymbol{x}} = \boldsymbol{F}^{(1)}(\boldsymbol{k}, \boldsymbol{x})$.

An **approximated conservation law** is a function $\phi(\boldsymbol{x})$ satisfying $D_{\boldsymbol{x}}\phi(\boldsymbol{x})\boldsymbol{F}^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0$, for any $\boldsymbol{x} \in \mathbb{R}^n_{>0}$. If this is true also for any $\boldsymbol{k} \in \mathbb{R}^r_{>0}$ the conservation law is **unconditional on the parameters**.

- A set

$$\Phi(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_s(\boldsymbol{x}))^{\mathsf{T}}$$

of approximate conservation laws is called **complete** if the Jacobian matrix

$$\boldsymbol{J}_{\boldsymbol{F}^{(1)},\phi}(\boldsymbol{k}, \boldsymbol{x}) = D_{\boldsymbol{x}}(\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}), \Phi(\boldsymbol{x}))^{\mathsf{T}}$$

has rank n for any $\boldsymbol{k} \in \mathbb{R}^r_{>0}, \boldsymbol{x} \in \mathbb{R}^n_{>0}$ satisfying $\boldsymbol{F}^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0$.

- The set is called **independent** if the Jacobian matrix of $\Phi(\boldsymbol{x})^{\mathsf{T}}$ with respect to $\boldsymbol{x}$ has rank s for any $\boldsymbol{k} \in \mathbb{R}^r_{>0}, \boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\boldsymbol{F}^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0$.

Approximate conservation laws

- The **truncated system** is $\dot{\boldsymbol{x}} = \boldsymbol{F}^{(1)}(\boldsymbol{k}, \boldsymbol{x})$.

An **approximated conservation law** is a function $\phi(\boldsymbol{x})$ satisfying $D_{\boldsymbol{x}}\phi(\boldsymbol{x})\boldsymbol{F}^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0$, for any $\boldsymbol{x} \in \mathbb{R}^n_{>0}$. If this is true also for any $\boldsymbol{k} \in \mathbb{R}^r_{>0}$ the conservation law is **unconditional on the parameters**.

- A set

$$\Phi(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_s(\boldsymbol{x}))^{\mathsf{T}}$$

of approximate conservation laws is called **complete** if the Jacobian matrix

$$\boldsymbol{J}_{\boldsymbol{F}^{(1)},\phi}(\boldsymbol{k}, \boldsymbol{x}) = D_{\boldsymbol{x}}(\boldsymbol{F}(\boldsymbol{k}, \boldsymbol{x}), \Phi(\boldsymbol{x}))^{\mathsf{T}}$$

has rank n for any $\boldsymbol{k} \in \mathbb{R}^r_{>0}, \boldsymbol{x} \in \mathbb{R}^n_{>0}$ satisfying $\boldsymbol{F}^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0$.

- The set is called **independent** if the Jacobian matrix of $\Phi(\boldsymbol{x})^{\mathsf{T}}$ with respect to $\boldsymbol{x}$ has rank s for any $\boldsymbol{k} \in \mathbb{R}^r_{>0}, \boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\boldsymbol{F}^{(1)}(\boldsymbol{k}, \boldsymbol{x}) = 0$.
- Properties tested using parametric rank computation.

Approximate conservation laws

• The **truncated system** is $\dot{\boldsymbol{x}} = \boldsymbol{F}^{(1)}(\boldsymbol{k},\boldsymbol{x})$.
An **approximated conservation law** is a function $\phi(\boldsymbol{x})$ satisfying $D_{\boldsymbol{x}}\phi(\boldsymbol{x})\boldsymbol{F}^{(1)}(\boldsymbol{k},\boldsymbol{x}) = 0$, for any $\boldsymbol{x} \in \mathbb{R}^n_{>0}$. If this is true also for any $\boldsymbol{k} \in \mathbb{R}^r_{>0}$ the conservation law is **unconditional on the parameters**.

• A set

$$\Phi(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}),\dots,\phi_s(\boldsymbol{x}))^{\mathsf{T}}$$

of approximate conservation laws is called **complete** if the Jacobian matrix

$$\boldsymbol{J}_{\boldsymbol{F}^{(1)},\phi}(\boldsymbol{k},\boldsymbol{x}) = D_{\boldsymbol{x}}(\boldsymbol{F}(\boldsymbol{k},\boldsymbol{x}),\Phi(\boldsymbol{x}))^{\mathsf{T}}$$

has rank n for any $\boldsymbol{k} \in \mathbb{R}^r_{>0}, \boldsymbol{x} \in \mathbb{R}^n_{>0}$ satisfying $\boldsymbol{F}^{(1)}(\boldsymbol{k},\boldsymbol{x}) = 0$.

• The set is called **independent** if the Jacobian matrix of $\Phi(\boldsymbol{x})^{\mathsf{T}}$ with respect to $\boldsymbol{x}$ has rank s for any $\boldsymbol{k} \in \mathbb{R}^r_{>0}, \boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\boldsymbol{F}^{(1)}(\boldsymbol{k},\boldsymbol{x}) = 0$.

• Properties tested using parametric rank computation.

Proposition: *If a system has a complete set of conservation laws, then the intersection of the steady state variety* $\mathcal{S}_{\boldsymbol{k}} = \{\boldsymbol{F}^{(1)}(\boldsymbol{x},\boldsymbol{k}) = 0\}$ *with* $\{\Phi(\boldsymbol{x}) = \boldsymbol{c}_0\} \cap \mathbb{R}^n_{>0}$*, where* $\boldsymbol{c}_0 \in \mathbb{R}^r$ *is finite.*

Approximate conservation laws as slow variables

- Let $\Phi(x_1, \dots, x_n)$ a linear, monomial or polynomial approximate conservation law.
- Φ **depends** on x_i if $\frac{\partial \Phi}{\partial x_i} \neq 0$.
- Φ is irreducible if there are no Φ_1, Φ_2 depending on distinct sets of variables such that $\Phi = \Phi_1 + \Phi_2$ or $\Phi = \Phi_1 \Phi_2$.
- Let $q = \Phi(\boldsymbol{x})$ a new variable.

Theorem: *If Φ is an irreducible linear, monomial or polynomial approximate conservation law, then q is slower than all the variables x_i on which Φ is dependent.*

For the proof one needs to consider the dominant term (valuation) of $\frac{\dot{q}}{q}$.

Michaelis-Menten example

$$S + E \underset{k_2}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_3\delta} E + P$$

$$\begin{aligned}
\dot{x_1} &= -k_1 x_1 x_3 + k_2 x_2 \\
\dot{x_2} &= k_1 x_1 x_3 - k_2 x_2 - \delta k_3 x_2 \\
\dot{x_3} &= -k_1 x_1 x_3 + k_2 x_2 + \delta k_3 x_2.
\end{aligned}$$

The truncated system

$$\begin{aligned}
\dot{x_1} &= -k_1 x_1 x_3 + k_2 x_2 \\
\dot{x_2} &= k_1 x_1 x_3 - k_2 x_2 \\
\dot{x_3} &= -k_1 x_1 x_3 + k_2 x_2
\end{aligned}$$

has two irreducible linear conservation laws
$x_4 = x_1 + x_2$ and $x_5 = x_2 + x_3$.
They are independent and can be used to eliminate two fast variables
$x_2 = x_4 - x_1, x_3 = x_5 - x_4 + x_1$.

Michaelis-Menten example

In the new variables

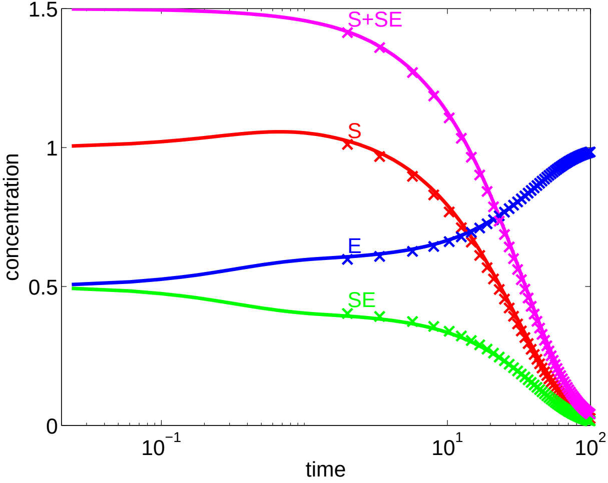
$$\begin{aligned}\dot{x}_1 &= -k_1 x_1 (x_5 - x_4 + x_1) + k_2 (x_4 - x_1),\\ \dot{x}_4 &= -\delta k_3 (x_4 - x_1),\\ \dot{x}_5 &= 0\end{aligned}$$

x_4 is slow, x_5 is conserved.
Turn x_5 into a parameter $x_5 = k_4$

$$\begin{aligned}\dot{x}_1 &= -k_1 x_1 (k_4 - x_4 + x_1) + k_2 (x_4 - x_1)\\ \dot{x}_4 &= \delta k_3 (x_4 - x_1)\end{aligned}$$

which is a standard slow/fast system with hyperbolic quasi-steady state.

Numerical test of the reduction



Schur complement

Let $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ be a block matrix such that A is invertible.

$$AX + BY = 0$$

$$CX + DY = (D - CA^{-1}B)Y$$

$M/A = D - CA^{-1}B$ is called Schur complement of the block A of M. It satisfies:

$$\begin{aligned} |M| &= |M/A||A|, \text{ Schur formula,} \\ rk(M) &= rk(M/A) + rk(A), \text{ Guttman rank additivity formula.} \end{aligned}$$

$$F_X(X,Y) = 0$$

$$F_{red}(Y) = F_Y(X,Y)$$

$$D_Y F_{red}(Y) = D_{X,Y} \begin{bmatrix} F_X \\ F_Y \end{bmatrix} / D_X F_X$$

Nested reduction

Set $\tau' = \tau\delta^{b_l}$, $\delta_k = \delta^{b_{k+1}-b_k}$

$$\underbrace{\begin{aligned}
\delta_1\delta_2\ldots\delta_{l-1}\boldsymbol{z}_1' &= (\bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_1(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})),\\
&\vdots\\
\delta_{l-1}\boldsymbol{z}_{l-1}' &= (\bar{\boldsymbol{f}}_{l-1}^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_{l-1}(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})),\\
\boldsymbol{z}_l' &= (\bar{\boldsymbol{f}}_l^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_l(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})),\\
\boldsymbol{z}_{l+1}' &= \delta_l(\bar{\boldsymbol{f}}_{l+1}^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_{l+1}(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})),\\
&\vdots\\
\boldsymbol{z}_m' &= \delta_l\delta_{l+1}\ldots\delta_{m-1}(\bar{\boldsymbol{f}}_m^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_m(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})).
\end{aligned}}_{S_\delta^{(l)}}$$

Set $\delta_1 = \delta_2 = \ldots = \delta_{l-1} = 0$

$$\underbrace{\begin{aligned}
0 &= \bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}),\\
&\vdots\\
0 &= \bar{\boldsymbol{f}}_{l-1}^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}),\\
\boldsymbol{z}_l' &= \bar{\boldsymbol{f}}_l^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}),\\
\boldsymbol{z}_{l+1}' &= \delta_l(\bar{\boldsymbol{f}}_{l+1}^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_{l+1}(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})),\\
&\vdots\\
\boldsymbol{z}_m' &= \delta_l\delta_{l+1}\ldots\delta_{m-1}(\bar{\boldsymbol{f}}_m^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_m(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})).
\end{aligned}}_{S_0^{(l)}}$$

[Cardin and Teixera 2017] When $\delta_{l-1} \to 0$, the solutions of $S_\delta^{(l)}$ converge uniformly on $[0, T]$ to solutions of $S_0^{(l)}$, provided that $\Re(Spec(D_{\boldsymbol{z}_k}\bar{\boldsymbol{F}}_k^{(1)}/D_{\boldsymbol{z}_{k-1}}\bar{\boldsymbol{F}}_{k-1}^{(1)})) < 0$ on $\mathcal{M}_k := \{\bar{\boldsymbol{F}}_k^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) = 0\}$ for $1 \le k \le l$, where $\bar{\boldsymbol{F}}_k^{(1)} = (\bar{\boldsymbol{f}}_1^{(1)},\ldots,\bar{\boldsymbol{f}}_k^{(1)})^\mathsf{T}$.

Nested reduction

Set $\tau' = \tau\delta^{b_l}$, $\delta_k = \delta^{b_{k+1}-b_k}$

$$\underbrace{\begin{aligned}
\delta_1\delta_2\ldots\delta_{l-1}\boldsymbol{z}_1' &= (\bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_1(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})),\\
&\vdots\\
\delta_{l-1}\boldsymbol{z}_{l-1}' &= (\bar{\boldsymbol{f}}_{l-1}^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_{l-1}(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})),\\
\boldsymbol{z}_l' &= (\bar{\boldsymbol{f}}_l^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_l(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})),\\
\boldsymbol{z}_{l+1}' &= \delta_l(\bar{\boldsymbol{f}}_{l+1}^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_{l+1}(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})),\\
&\vdots\\
\boldsymbol{z}_m' &= \delta_l\delta_{l+1}\ldots\delta_{m-1}(\bar{\boldsymbol{f}}_m^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_m(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})).
\end{aligned}}_{S_\delta^{(l)}}$$

Set $\delta_1 = \delta_2 = \ldots = \delta_{l-1} = 0$

$$\underbrace{\begin{aligned}
0 &= \bar{\boldsymbol{f}}_1^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}),\\
&\vdots\\
0 &= \bar{\boldsymbol{f}}_{l-1}^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}),\\
\boldsymbol{z}_l' &= \bar{\boldsymbol{f}}_l^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}),\\
\boldsymbol{z}_{l+1}' &= \delta_l(\bar{\boldsymbol{f}}_{l+1}^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_{l+1}(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})),\\
&\vdots\\
\boldsymbol{z}_m' &= \delta_l\delta_{l+1}\ldots\delta_{m-1}(\bar{\boldsymbol{f}}_m^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) + \bar{\boldsymbol{g}}_m(\bar{\boldsymbol{k}},\boldsymbol{z},\bar{\delta})).
\end{aligned}}_{S_0^{(l)}}$$

[Cardin and Teixera 2017] When $\delta_{l-1} \to 0$, the solutions of $S_\delta^{(l)}$ converge uniformly on $[0, T]$ to solutions of $S_0^{(l)}$, provided that $\Re(Spec(D_{\boldsymbol{z}_k}\bar{\boldsymbol{F}}_k^{(1)}/D_{\boldsymbol{z}_{k-1}}\bar{\boldsymbol{F}}_{k-1}^{(1)})) < 0$ on $\mathcal{M}_k := \{\bar{\boldsymbol{F}}_k^{(1)}(\bar{\boldsymbol{k}},\boldsymbol{z}) = 0\}$ for $1 \le k \le l$, where $\bar{\boldsymbol{F}}_k^{(1)} = (\bar{\boldsymbol{f}}_1^{(1)},\ldots,\bar{\boldsymbol{f}}_k^{(1)})^{\mathsf{T}}$.

Not applicable if $\bar{\boldsymbol{F}}_l^{(1)}$ has approximate conservation laws.

Reduction with approximate conservation laws

The following conditions are satisfied:

1. There is $\boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\boldsymbol{F}_l^{(1)}(\boldsymbol{x}) = 0$. For all $\boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\boldsymbol{F}_l^{(1)}(\boldsymbol{x}) = 0$, $|D_{\boldsymbol{X}_k} \boldsymbol{F}_k^{(1)}| \neq 0$ for all $k \in \{1, \ldots, l-1\}$, and $|D_{\boldsymbol{X}_l} \boldsymbol{F}_l^{(1)}| = 0$.
2. There are complete, irreducible, independent approximate conservation laws $\Phi_l(\boldsymbol{x}) = (\phi_{1l}(\boldsymbol{x}), \ldots, \phi_{s_l l}(\boldsymbol{x}))^{\mathsf{T}}$ depending only on $\boldsymbol{X}_l$, such that

   $$(D_{\boldsymbol{X}_l} \Phi_l) \boldsymbol{F}_l^{(1)} = 0,\ rk\left(D_{\boldsymbol{X}_l} \begin{pmatrix} \boldsymbol{F}_l^{(1)} \\ \Phi_l \end{pmatrix} \right) = n_1 + \ldots + n_l,\ rk(D_{\boldsymbol{X}_l} \Phi_l) = s_l,$$

   for $\boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\boldsymbol{F}_l^{(1)}(\boldsymbol{x}) = 0$.
3. Furthermore, $\begin{pmatrix} \boldsymbol{F}_k^{(1)} \\ \Phi_l \end{pmatrix}$ are independent as functions of $(\boldsymbol{X}_k, \boldsymbol{x}_l)$, namely

   $$rk\left(D_{(\boldsymbol{X}_k, \boldsymbol{x}_l)} \begin{pmatrix} \boldsymbol{F}_k^{(1)} \\ \Phi_l \end{pmatrix} \right) = n_1 + \ldots + n_k + s_l,$$

   for all $1 \leq k \leq l-1$ and $\boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\boldsymbol{F}_l^{(1)}(\boldsymbol{x}) = 0$.

Reduction with approximate conservation laws

- Elimination: Start with

$$\boldsymbol{x}_l^c = \Phi_l(\boldsymbol{X}_l).$$

We have $\boldsymbol{X}_l = (\boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \check{\boldsymbol{x}}_l)$, where $\boldsymbol{X}_{l-1} \in \mathbb{R}^{n_1+\ldots+n_{l-1}}$, $\hat{\boldsymbol{x}}_l \in \mathbb{R}^{n_l - s_l}$, $\check{\boldsymbol{x}}_l \in \mathbb{R}^{s_l}$ and $|D_{\check{\boldsymbol{x}}_l}\Phi_l| \neq 0$.

$$\check{\boldsymbol{x}}_l = \Psi_l(\boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \boldsymbol{x}_l^c)$$

- Reduction: Define

$$\boldsymbol{F}_k^{red,i}(\boldsymbol{k}, \boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \boldsymbol{x}_l^c, \boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, \ldots, \boldsymbol{x}_m)$$

$$\hat{\boldsymbol{F}}_k^{(i)}(\boldsymbol{k}, \boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \Psi_l(\boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \boldsymbol{x}_l^c), \boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, \ldots, \boldsymbol{x}_m),$$

where $\hat{\boldsymbol{F}}_k^{(i)} = \boldsymbol{F}_k^{(i)}$, for $k \in \{1, \ldots, l-1\}$, and $\hat{\boldsymbol{F}}_l^{(i)} = (\boldsymbol{F}_{l-1}^{(i)}, \hat{\boldsymbol{f}}_l^{(i)})$, for $i \in \{1, 2\}$.

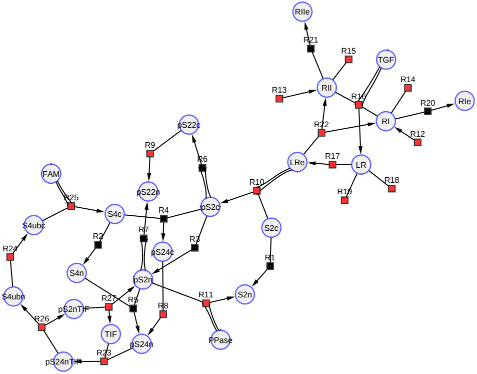
Theorem: *If the Conditions are fulfilled and $\boldsymbol{F}_l^{red,1}(\boldsymbol{k}, \boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l, \boldsymbol{x}_l^c, \boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, \ldots, \boldsymbol{x}_m) = 0$, then $|D_{\hat{\boldsymbol{X}}_k}\boldsymbol{F}_k^{(red,1)}| \neq 0$, where $\hat{\boldsymbol{X}}_k = \boldsymbol{X}_k$, for $1 \leq k \leq l-1$, and $\hat{\boldsymbol{X}}_l = (\boldsymbol{X}_{l-1}, \hat{\boldsymbol{x}}_l)$.*

Algorithm

INPUT A CRN given by a polynomial vector field $\boldsymbol{F}(\boldsymbol{k},\boldsymbol{x})$.
OUTPUT: A transformed CRN given by a modified polynomial vector field.

ScaleAndTruncate.
l:=0
While $l < m$
  l:=l+1
  While $|D_{\boldsymbol{X}_l} F_l^{(1)}| \neq 0$
    l:=l+1
  end while
  Find a complete set Φ_l of independent conservation laws for $\boldsymbol{F}_l^{(1)}$ satisfying rank conditions.
  Compute $\psi_{il}(\boldsymbol{X}_{l-1},\hat{\boldsymbol{x}}_l,\boldsymbol{x}_l^c)$ the solution of the equation $\boldsymbol{x}_l^c = \Phi_{\boldsymbol{l}}(\boldsymbol{x})$.
  For $i := 1$ to s_l
    If Φ_{il} is not exact
      Replace the ODE satisfied by $\check{x}_{il}$ by $\dot{x}_{il}^c = (D_{\boldsymbol{x}}\Phi_{il})\boldsymbol{F}(\boldsymbol{k},\boldsymbol{x})$
      Substitute $\check{x}_{il} \leftarrow \psi_{il}(\boldsymbol{X}_{l-1},\hat{\boldsymbol{x}}_l,\boldsymbol{x}_l^c)$
    else
      Delete the ODE satisfied by $\check{x}_{il}$
      Define new constants k_{il}^c and concatenate them to $\boldsymbol{k}$.
      Substitute $\check{x}_{il} \leftarrow \psi_{il}(\boldsymbol{X}_{l-1},\hat{\boldsymbol{x}}_l,\boldsymbol{k}_l^c)$
    end if
  end for
  ScaleAndTruncate.
end while

Case study, TGFβ model (21 variables, 4 timescales)



$$\dot{y}_1 = \varepsilon^2(\bar{k}_2 y_2 - \bar{k}_1 y_1 - \varepsilon^2 \bar{k}_{16} y_1 y_{11})$$
$$\dot{y}_2 = \varepsilon^1(\bar{k}_1 y_1 + \varepsilon^2 \bar{k}_{17}\bar{k}_{36} y_6 - \bar{k}_2 y_2)$$
$$\dot{y}_3 = \varepsilon^2(\bar{k}_3 y_4 + \varepsilon\bar{k}_7 y_7 + \varepsilon\bar{k}_{33}\bar{k}_{38} y_{20} - \bar{k}_3 y_3 - \varepsilon\bar{k}_6 y_3 y_5)$$
$$\dot{y}_4 = \varepsilon^2(\bar{k}_3 y_3 + \varepsilon\bar{k}_9 y_8 - \bar{k}_3 y_4 - \varepsilon\bar{k}_8 y_4 y_6)$$

$$\dot{y}_5 = \varepsilon^1(\bar{k}_5 y_6 + \bar{k}_7 y_7 + 2\varepsilon^2 \bar{k}_{11} y_9 + \varepsilon\bar{k}_{16} y_1 y_{11} - 2$$
$$\dot{y}_6 = \varepsilon^1(\bar{k}_9 y_8 + \bar{k}_{35} y_{21} + 2\varepsilon^2 \bar{k}_{13} y_{10} + \varepsilon\bar{k}_4 y_5 - \bar{k}_5$$
$$\dot{y}_7 = \varepsilon^2(\bar{k}_6 y_3 y_5 - \bar{k}_7 y_7 - \bar{k}_{14} y_7)$$
$$\dot{y}_8 = \varepsilon^2(\bar{k}_{14} y_7 + \bar{k}_8 y_4 y_6 - \bar{k}_9 y_8 - \bar{k}_{31} y_8 y_{17})$$
$$\dot{y}_9 = \varepsilon^2(\bar{k}_{10} y_5^2 - \bar{k}_{11} y_9 - \bar{k}_{15} y_9)$$
$$\dot{y}_{10} = \varepsilon^2(\bar{k}_{15} y_9 + \bar{k}_{12} y_6^2 - \bar{k}_{13} y_{10})$$
$$\dot{y}_{11} = \varepsilon^3(\bar{k}_{23} y_{14} - \bar{k}_{30} y_{11})$$
$$\dot{y}_{12} = \varepsilon^2(\varepsilon\bar{k}_{18} + \bar{k}_{27} y_{15} + \varepsilon\bar{k}_{30} y_{11} - \bar{k}_{26} y_{12} - \varepsilon\bar{k}_{20}$$
$$\dot{y}_{13} = \varepsilon^0(\bar{k}_{19} + \bar{k}_{30} y_{11} + \varepsilon^2 \bar{k}_{29} y_{16} - \varepsilon^3 \bar{k}_{21} y_{13} - \varepsilon^2$$
$$\dot{y}_{14} = \varepsilon^2(\bar{k}_{22}\bar{k}_{37} y_{12} y_{13} - \bar{k}_{23} y_{14} - \bar{k}_{25} y_{14} - \varepsilon\bar{k}_{24} y_1$$
$$\dot{y}_{15} = \varepsilon^3(\bar{k}_{26} y_{12} - \bar{k}_{27} y_{15})$$
$$\dot{y}_{16} = \varepsilon^3(\bar{k}_{28} y_{13} - \bar{k}_{29} y_{16})$$
$$\dot{y}_{17} = \varepsilon^2(\bar{k}_{35} y_{21} - \bar{k}_{31} y_8 y_{17})$$
$$\dot{y}_{18} = \varepsilon^2(\bar{k}_{31} y_8 y_{17} - \bar{k}_{34} y_{18})$$
$$\dot{y}_{19} = \varepsilon^2(\bar{k}_{34} y_{18} - \bar{k}_{32} y_{19})$$
$$\dot{y}_{20} = \varepsilon^1(\bar{k}_{32} y_{19} - \bar{k}_{33}\bar{k}_{38} y_{20})$$
$$\dot{y}_{21} = \varepsilon^2(\bar{k}_{34} y_{18} - \bar{k}_{35} y_{21}),$$

Transformed model (18 variables, 5 timescales)

$$
\begin{aligned}
\dot{y}_2 &= \varepsilon^1(k_1 k_{40} - k_2 y_2),\\
\dot{y}_4 &= \varepsilon^2(k_3 k_{41} - 2k_3 y_4),\\
\dot{y}_5 &= \varepsilon^1(k_5 y_6 + k_7 y_7 + k_6 y_4 y_5 - k_6 k_{41} y_5),\\
\dot{y}_6 &= \varepsilon^1(k_{35} k_{39} + k_9 y_8 + k_9 y_{17} - k_9 k_{39} - k_5 y_6 - k_9 y_7 - k_{35} y_{17} - k_{35} y_{18} - k_8 y_4 y_6),\\
\dot{y}_7 &= \varepsilon^2(k_6 k_{41} y_5 - k_7 y_7 - k_1 4 y_7 - k_6 y_4 y_5),\\
\dot{y}_8 &= \varepsilon^3(k_{16} k_{40} y_{11} - k_{17} k_{36} y_6),\\
\dot{y}_9 &= \varepsilon^2(k_{10} y_5^2 - k_{11} y_9 - k_{15} y_9),\\
\dot{y}_{10} &= \varepsilon^2(k_{15} y_9 + k_{12} y_6^2 - k_{13} y_{10}),\\
\dot{y}_{11} &= \varepsilon^3(k_{23} y_{14} - k_{30} y_{11}),\\
\dot{y}_{12} &= \varepsilon^2(k_{27} y_{15} - k_{26} y_{12}),\\
\dot{y}_{13} &= \varepsilon^0(k_{19} + k_{30} y_{11} - k_{22} k_{37} y_{12} y_{13}),\\
\dot{y}_{14} &= \varepsilon^2(k_{22} k_{37} y_{12} y_{13} - k_{23} y_{14} - k_{25} y_{14}),\\
\dot{y}_{15} &= \varepsilon^4(k_{18} + k_{30} y_{11} - k_{20} y_{12} - k_{22} k_{37} y_{12} y_{13}),\\
\dot{y}_{16} &= \varepsilon^3(k_{28} y_{13} - k_{29} y_{16}),\\
\dot{y}_{17} &= \varepsilon^2(k_{35} k_{39} + k_{31} k_{39} y_{17} + k_{31} y_7 y_{17} - k_{35} y_{17} - k_{35} y_{18} - k_{31} y_{17}^2 - k_{31} y_8 y_{17}),\\
\dot{y}_{18} &= \varepsilon^2(k_{31} y_{17}^2 + k_{31} y_8 y_{17} - k_{34} y_{18} - k_{31} k_{39} y_{17} - k_{31} y_7 y_{17}),\\
\dot{y}_{19} &= \varepsilon^2(k_{34} y_{18} - k_{32} y_{19}),\\
\dot{y}_{20} &= \varepsilon^1(k_{32} y_{19} - k_{33} k_{38} y_{20}).
\end{aligned}
$$

Numerical test of the reduction