Algebraic Biochemistry: a Framework for On-line Analog Computation in Cells

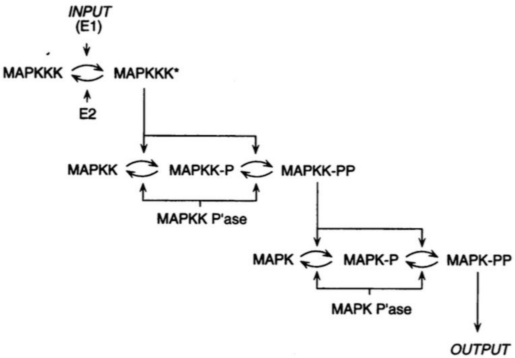
Mathieu Hemery & François Fages

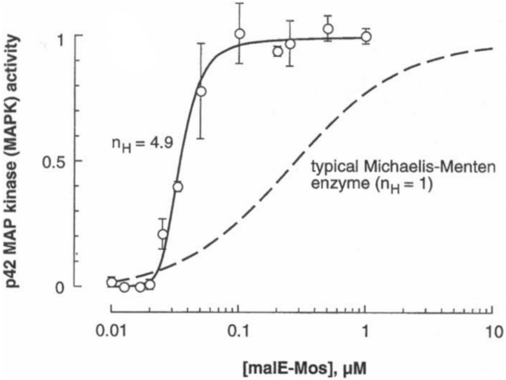
EPI Lifeware, Centre Inria Saclay Île de France

September, CMSB 2022

Mitogen-Activated Protein Kinase (MAPK)

An “on-line” analog digital converter: $f(x) \simeq \dfrac{x^{4.9}}{1+x^{4.9}}$!





Huang, C.-Y. et Ferrell J. Ultrasensitivity in the MAPK cascade. PNAS, 1996, **93**,19.

Chemical Reactions Networks 101

A Chemical Reaction Network (CRN) is:

- a finite set of species X_i
- a set of reactions:
  - a multiset of reactants: R
  - a multiset of products: P
  - a kinetic rate function: $f(R)$

  and denoted:

$$R \xrightarrow{f(R)} P.$$

Chemical Reactions Networks 101

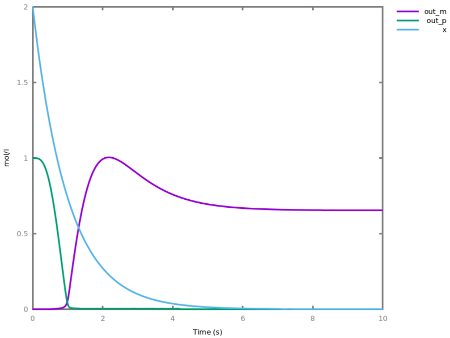
In the *differential semantics*, a CRN defines an Ordinary Differential Equations (ODE), $A \xrightarrow{f(A)} 2 \cdot B$:

$$\begin{aligned} \frac{dA}{dt} &= -f(A), \\ \frac{dB}{dt} &= 2 \cdot f(A). \end{aligned} \tag{1}$$

With *Mass Action Law*, f is a monomial: $f(R) = k \prod_{X \in R} X$,
and we have Polynomial ODE (PODE).

Chemical Reactions Networks are Turing Complete

$$\lim_{t\to\infty} \text{out} = \cos x^2$$



$f : I \subset \mathbb{R}_+ \to \mathbb{R}_+$ is CRN-computable if there exist a CRN over species $\overline{y}$, with ODE semantics $\overline{p}$, and a polynomial $q \in \mathbb{R}_+^n[\mathbb{R}_+]$, such that for all $x \in I$:

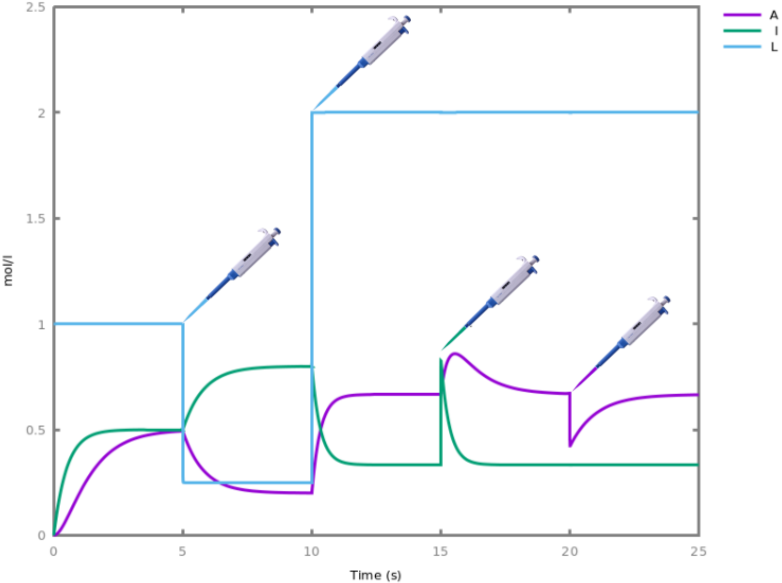
- $\overline{y}(0) = \overline{q}(x)$
- $\overline{y}'(t) = \overline{p}(\overline{y}(t))$
- $\boxed{\forall t > 1, \quad |y_1(t) - f(x)| \leqslant y_2(t)}$,
- $y_2(t) \geqslant 0$ and $\boxed{\lim_{t\to\infty} y_2(t) = 0}$.

Fages F., Le Guludec G., et al. Strong turing completeness of continuous CRN..., CMSB 2017.

Computing "on-line" with perturbations

A toy model:
the "1-level MAPK"

$$
\begin{aligned}
L + I &\rightarrow L + A \\
\emptyset &\leftrightarrow I \\
A &\rightarrow \emptyset
\end{aligned}
\qquad (2)
$$



Battle plan

What is the class of functions that can be computed On-line by a CRN?

- Definition
- Proof & Theorem
- Compilation pipeline

Def – F_S: functions stabilized by a CRN

A CRN over m inputs X, 1 output y and n auxiliary Z, stabilizes $f : I \subset \mathbb{R}_+^m \mapsto \mathbb{R}_+$, over the domain $D \subset \mathbb{R}_+^{m+1+n}$ if:

1. $\forall X^0 \in I, \{(X, y, Z) \in D | X = X^0\}$ is of plain dimension $n + 1$,
2. In the differential semantic with pinned input species X and initial conditions $X^0, y^0, Z^0 \in D$:

$$\boxed{\lim_{t \to \infty} y(t) = f(X^0)}.$$

It can be extended to functions $f : \mathbb{R}^m \to \mathbb{R}$ by dual-rail encoding:

$$\lim_{t \to \infty}(y^+ - y^-)(t) = f(X^+ - X^-).$$

Example: 1-level MAPK

We use the differential semantics with Mass Action Law.

At steady state, we have:

$$\frac{dI}{dt} = 1 - I - LI = 0, \quad \frac{dA}{dt} = LI - A = 0, \quad \frac{dL}{dt} = 0. \tag{3}$$

Thus by elimination:

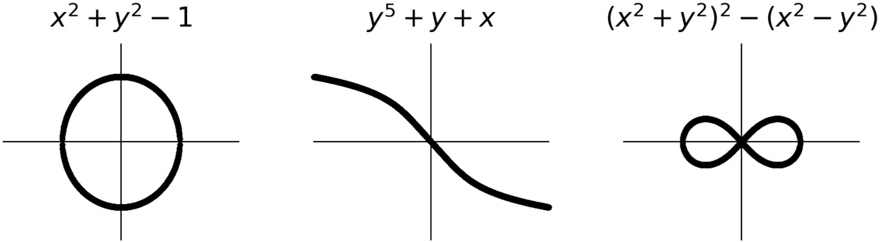
$$P(L, A) = \boxed{L - A - LA = 0}. \tag{4}$$

That is: $A(L) = \frac{L}{1+L}$.

Def – F_A: real algebraic functions

A function $f : I \subset \mathbb{R}^m \mapsto \mathbb{R}$ is algebraic iff there exists a polynomial P_f of $m+1$ variables such that:

$$\forall X \in I, \quad \boxed{P_f(X, f(X)) = 0}. \tag{5}$$



$F_S \subset F_A$ (sketch of the proof)

As $y(t)$ is projectively polynomial [1]:

$$P(X, y, y^{(1)}, \dots, y^{(n)}) = 0 \tag{6}$$

with X^0, y^0 as initial condition we have: $\forall k,\ y^{(k)}(t) = 0$, thus:

$$P^{\star}(X^0, y^0) = 0. \tag{7}$$

Can we stabilize any algebraic functions?

[1] Carothers D., Parker G., et al. Some properties of solutions to PODE. EJDE 2005, vol. 2005, No. 40, 17.

$F_A \subset F_S$ (sketch of the proof)

Take $f : I \mapsto \mathbb{R}$ a real algebraic function
$\exists P_f$ in **reduced form**, $\forall X \in I, P_f(X, f(X)) = 0$,

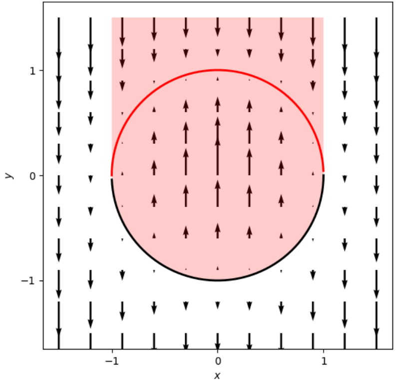
$$\begin{aligned} \frac{dy}{dt} &= \pm P_f(X, y), \\ \frac{dX}{dt} &= 0, \end{aligned} \tag{8}$$

$Y = f(X)$ is a fixed point.

Choose $+$ or $-$ to stabilized the desired branch.

Example: Unit circle

$$P(x, y) = 1 - x^2 - y^2 \tag{9}$$



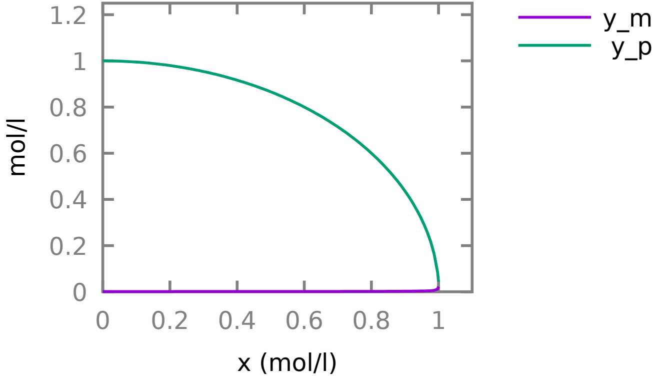
Theorem

$$F_S = F_A$$

Example: Unit circle in Biocham

```
stabilize_expression(x^2 + y^2 - 1, y, [x = 0, y=1]).
```

$$
\begin{aligned}
y_m + y_p &\xrightarrow{\text{fast}} \varnothing, \\
\varnothing &\rightarrow y_p, \\
2 \cdot y_p &\rightarrow y_p, \\
2 \cdot x &\rightarrow y_m + 2 \cdot x, \\
2 \cdot y_m &\rightarrow 3 \cdot y_m.
\end{aligned} \tag{10}
$$



Compilation pipeline in Biocham

```
stabilize_expression(polynomial, output_name, point)
```

Polynomialization preprocessing to ease the user interface

Stabilization choose the $\pm$ to stabilize the selected branch

Quadratization reduce to a quadratic PODE to enforce elementary reactions

Dual-rail encoding split variable that may become negative into their positive and negative part

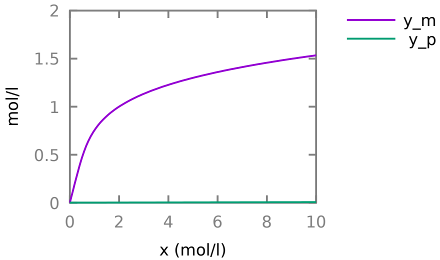
CRN generation insert a reaction for each monomial

Hemery M., Fages F. et Soliman S., On the complexity of quadratization for PODE. CMSB 2020.

Example: Bring radical

```
stabilize_expression(y^5 + y + x, y, [x = 2, y=1]).
```

$$y_m + y_p \xrightarrow{\text{fast}} \varnothing, \qquad y2_m + y2_p \xrightarrow{\text{fast}} \varnothing,$$
$$y3_m + y3_p \xrightarrow{\text{fast}} \varnothing, \qquad y_p \to \varnothing,$$
$$y2_m + y3_p \to y2_m + y3_p + y_p, \qquad y2_p + y3_m \to y2_p + y3_m + y_p,$$
$$x \to x + y_m, \qquad y_m \to \varnothing,$$
$$y2_p + y3_p \to y2_p + y3_p + y_m, \qquad y2_m + y3_m \to y2_m + y3_m + y_m,$$
$$2 \cdot y_p \to y2_p + 2 \cdot y_p, \qquad 2 \cdot y_m \to y2_p + 2 \cdot y_m,$$
$$y2_p \to \varnothing, \qquad y2_m \to \varnothing,$$
$$y2_p + y_p \to y2_p + y3_p + y_p, \qquad y2_m + y_m \to y2_m + y3_p + y_m,$$
$$y3_p \to \varnothing, \qquad y2_p + y_m \to y2_p + y3_m + y_m,$$
$$y2_m + y_p \to y2_m + y3_m + y_p, \qquad y3_m \to \varnothing.$$



Hermite, C. Sur la résolution de l'équation du cinquème degré. Comptes Rendus de l'Académie des Sciences, 1858.

A two inputs example: Computing the norm

$$norm(x,y) = \sqrt{x^2 + y^2} \quad \Rightarrow \quad P(x,y,norm) = norm^2 - x^2 - y^2$$

```
stabilize_expression(norm^2-x^2-y^2, norm,
[norm = 1.4, x = 1, y=1]).
```

$$\begin{aligned} 2 \cdot x &\longrightarrow norm + 2 \cdot x, \\ 2 \cdot y &\longrightarrow norm + 2 \cdot y, \\ 2 \cdot norm &\longrightarrow norm. \end{aligned}$$

Conclusions & Perspectives

- On-line analog computation is restricted to the set of algebraic functions,
- We can compile any algebraic function into an abstract stabilizing CRN.

- Towards a concrete compiler with real reactions from BRENDA,
- and its *in-vitro* implementation [1].

[1] Courbet A., Amar P., Fages F., et al. Computer-aided biochemical programming of synthetic microreactors as diagnostic devices. Mol. sys. bio., 2018.