Efficient Methods to Compute Hopf Bifurcations in Chemical Reaction Networks Using Reaction Coordinates

Hassan Errami
Markus Eiswirth
Dima Grigoriev
Werner Seiler
Thomas Sturm
Andreas Weber

New method: HoCaT method

• We build on our previous work already presented in CASC 2012: "*Computing Hopf Bifurcations in Chemical Reaction Networks Using Reaction Coordinates*".

• The new method „HoCaT“:

- extends the previous algorithmic approach „HoCoQ“.
- uses a simplified condition for Hopf bifurcation.
- is based on tropical geometry.

Previous Work: HoCoQ method

- The previous method „HoCoQ“:

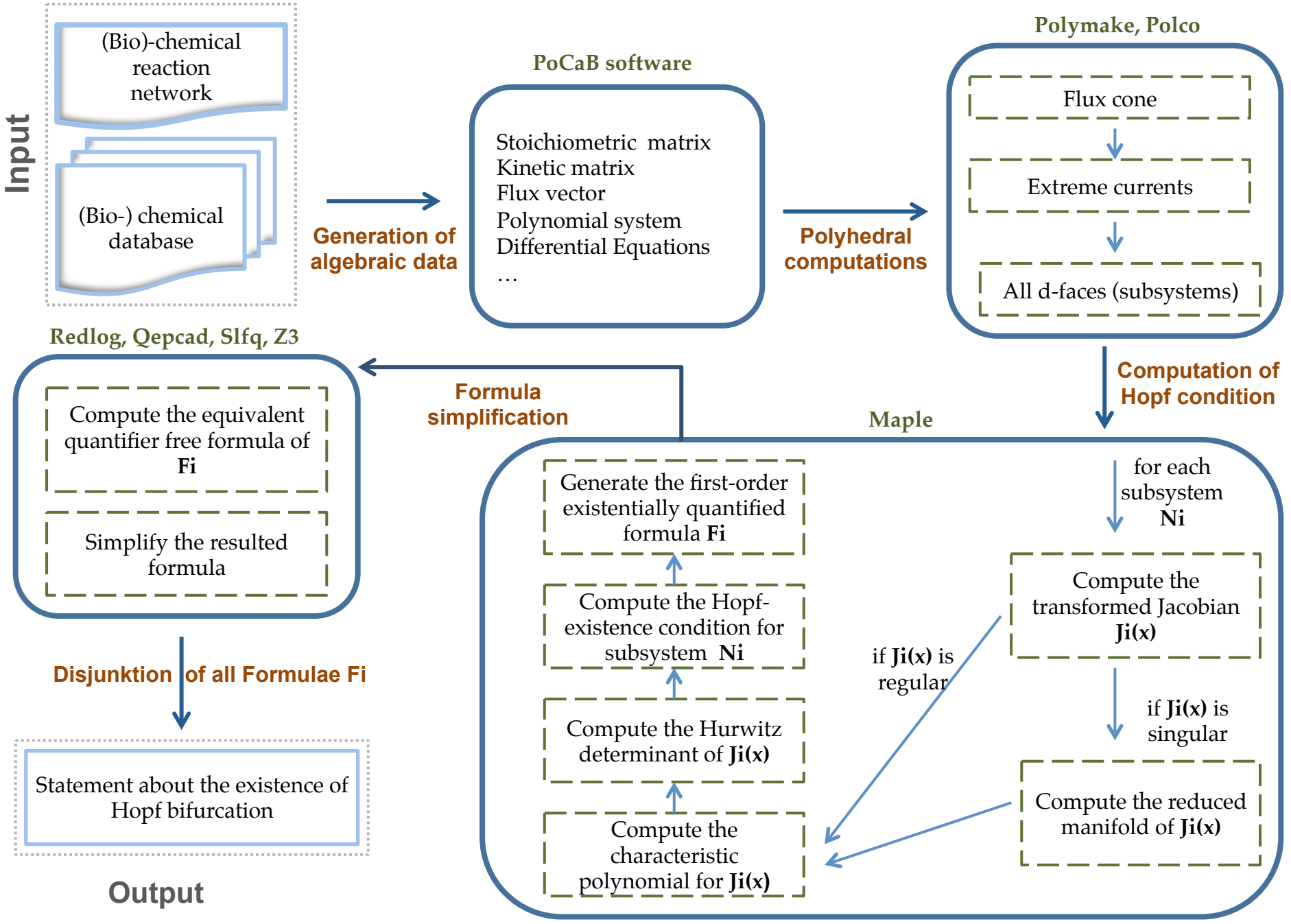
➢ combines the ideas of stoichiometric network analysis (SNA) and quantifier elimination.

➢ allows also the analysis of chemical systems with linear constraint.

➢ uses the semi algebraic condition for Hopf bifurcation:

$$\exists x(f_1(u,x)=0 \wedge f_2(u,x)=0 \wedge \cdots \wedge f_n(u,x)=0 \wedge$$
$$a_n > 0 \wedge \Delta_{n-1}(u,x)=0 \wedge \Delta_{n-2}(u,x)>0 \wedge \cdots \wedge \Delta_1(u,x)>0).$$

Previous Work: HoCoQ method



Previous Work: Results

- Using HoCoQ method we could compute Hopf bifurcation in chemical reaction systems, for which the existing symbolic methods fails.

- The criteria for determining Hopf bifurcation involving equality and inequality conditions still turned out to be hard problems for general real quantifier elimination procedures even for moderate dimensions.

The new method HoCaT: simplified condition for Hopf bifurcation

- By dropping condition of empty unstable manifold computational problem becomes easier.

- Criterion: pair of pure imaginary eigenvalues of Jacobian; dropping that all others have negative real part.
  Criterion is a consequence from (Orlando, 1911); see also (Liu, 1994) or (El Kahoui and Weber, 2000), . . . for alternate proofs and generalized criteria for n-pairs of pure imaginary eigenvalues.

- In convex coordinates this leads to a dramatic simplification of the problem.

HoCaT method: condition of pair of pure imaginary eigenvalues

- The new condition for the existence of Hopf bifurcations is given by $\Delta_{n-1}(j, x) = 0$ only.

- Solving such single equations enables us to refrain from utilizing quantifier elimination techniques.

- Instead, the main algorithmic problem is to determine whether a single multivariate polynomial has a zero for positive coordinates.

HoCaT method: Sufficient Conditions for a Positive Solution of a Single Multivariate Polynomial Equation

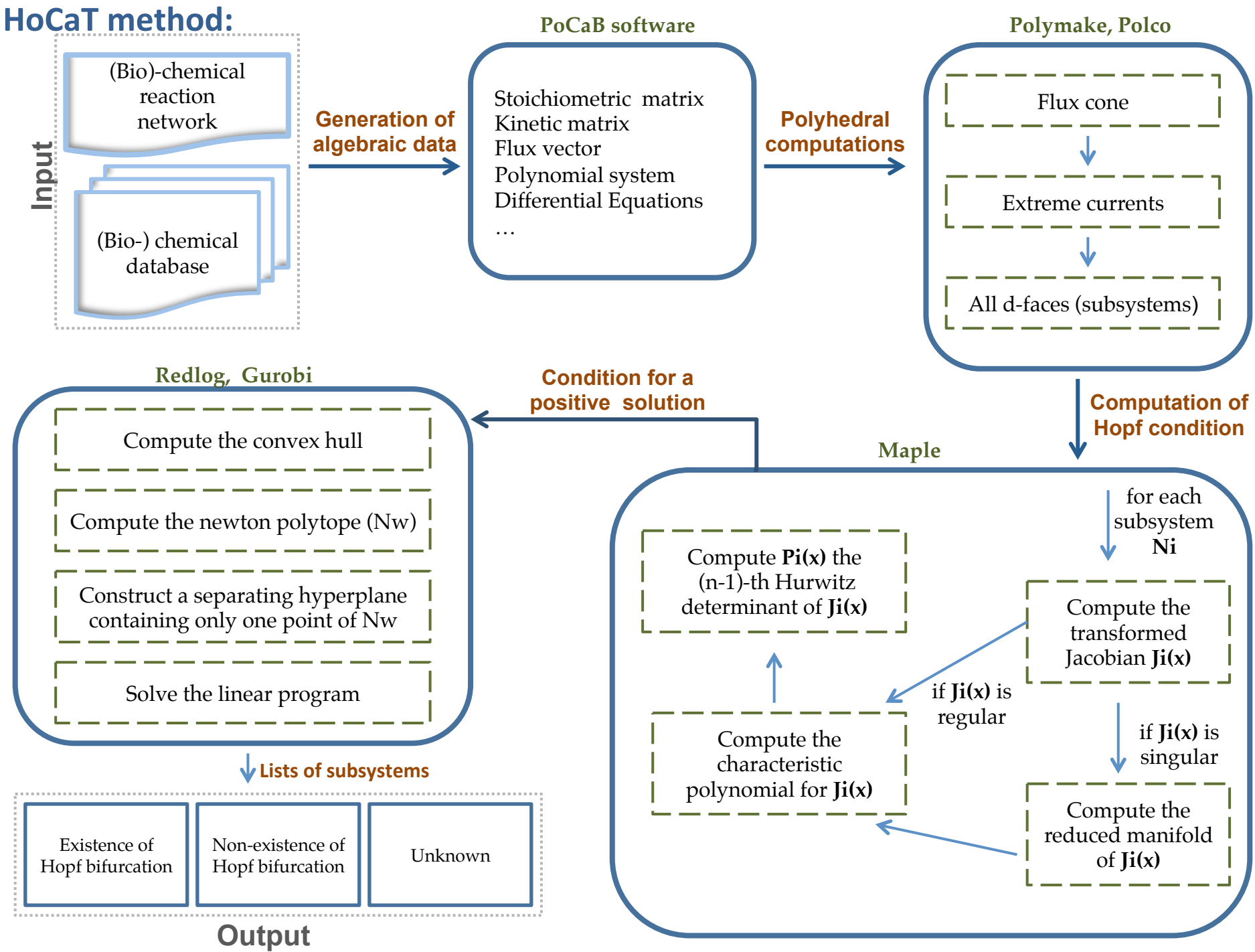
- Given $f \in \mathbb{Z}[x_1, \ldots, x_m]$, our goal is to heuristically certify the existence of at least one zero $(z_1, \ldots, z_m) \in]0, \infty[^m$ for which all coordinates are strictly positive.
- To start with, we evaluate $f(1, \ldots, 1) = f_1 \in \mathbb{R}$. If $f_1 = 0$, then we are done.
- If $f_1 < 0$, then by the intermediate value theorem, it is sufficient to find $p \in]0, \infty[^m$ such that $f(p) > 0$.
- Similarly, if $f_1 > 0$ it is sufficient to find $p \in]0, \infty[^m$ such that $(-f)(p) > 0$.

HoCaT method: Sufficient Conditions for a Positive Solution of a Single Multivariate Polynomial Equation

- This algorithmically reduces our original problem to finding, for given $g \in \mathbb{Z}[x_1, \ldots, x_m]$, at least one $p \in]0, \infty[^m$ such that $g(p) = f_2 > 0$.

- We have implemented a heuristics on the basis of the Newton polytope that ensure the existence of positive and negative values of the polynomial for positive coordinates.

- Requires linear programming on the Newton polytope of the single multivariate polynomial.
  The implementation (by. T. Sturm) has been made available in REDLOG with external calls to Gurobi.


HoCaT method:
Input
(Bio)-chemical reaction network
(Bio-) chemical database
Generation of algebraic data
PoCaB software
Stoichiometric matrix
Kinetic matrix
Flux vector
Polynomial system
Differential Equations
…
Polyhedral computations
Polymake, Polco
Flux cone
Extreme currents
All d-faces (subsystems)
Computation of Hopf condition
Maple
for each subsystem Ni
Compute the transformed Jacobian Ji(x)
if Ji(x) is singular
Compute the reduced manifold of Ji(x)
if Ji(x) is regular
Compute the characteristic polynomial for Ji(x)
Compute Pi(x) the (n-1)-th Hurwitz determinant of Ji(x)
Condition for a positive solution
Redlog, Gurobi
Compute the convex hull
Compute the newton polytope (Nw)
Construct a separating hyperplane containing only one point of Nw
Solve the linear program
Lists of subsystems
Existence of Hopf bifurcation
Non-existence of Hopf bifurcation
Unknown
Output

HoCaT vs HoCoQ: computation of Hopf bifurcations in the phosphofructokinase reaction using HoCaT method

Subsystem	Result	Time
$\mathcal{E}_1$	unsat	< 1
$\mathcal{E}_2$	unsat	< 1
$\mathcal{E}_3$	unsat	< 1
$\mathcal{E}_4$	sat	< 1
$\mathcal{E}_1\mathcal{E}_2$	unsat	< 1
$\mathcal{E}_1\mathcal{E}_3$	unsat	< 1
$\mathcal{E}_1\mathcal{E}_4$	sat	< 1
$\mathcal{E}_2\mathcal{E}_3$	unsat	< 1
$\mathcal{E}_2\mathcal{E}_4$	sat	< 1
$\mathcal{E}_3\mathcal{E}_4$	sat	< 1
$\mathcal{E}_1\mathcal{E}_2\mathcal{E}_3$	unsat	< 1
$\mathcal{E}_1\mathcal{E}_2\mathcal{E}_4$	sat	< 1
$\mathcal{E}_1\mathcal{E}_3\mathcal{E}_4$	sat	< 1
$\mathcal{E}_2\mathcal{E}_3\mathcal{E}_4$	sat	< 1
$\mathcal{E}_1\mathcal{E}_2\mathcal{E}_3\mathcal{E}_4$	sat	< 1

HoCaT vs HoCoQ: Computation of Hopf bifurcations in the phosphofructokinase reaction using HoCoQ method

Subsystem	Redlog		Z3	
	Result	Time(s)	Result	Time(s)
$\mathcal{E}_1$	false	< 1	unsat	< 1
$\mathcal{E}_2$	false	< 1	unsat	< 1
$\mathcal{E}_3$	false	< 1	unsat	< 1
$\mathcal{E}_4$	true	< 1	sat	< 1
$\mathcal{E}_1\mathcal{E}_2$	false	< 1	unsat	< 1
$\mathcal{E}_1\mathcal{E}_3$	false	< 1	unsat	< 1
$\mathcal{E}_1\mathcal{E}_4$	true	< 1	sat	< 1
$\mathcal{E}_2\mathcal{E}_3$	false	< 1	unsat	< 1
$\mathcal{E}_2\mathcal{E}_4$	true	< 1	sat	< 1
$\mathcal{E}_3\mathcal{E}_4$	true	< 1	sat	< 1
$\mathcal{E}_1\mathcal{E}_2\mathcal{E}_3$	false	< 1	unsat	< 1
$\mathcal{E}_1\mathcal{E}_2\mathcal{E}_4$	true	< 1	sat	< 1
$\mathcal{E}_1\mathcal{E}_3\mathcal{E}_4$	true	1	sat	< 1
$\mathcal{E}_2\mathcal{E}_3\mathcal{E}_4$	true	2.5	sat	< 1
$\mathcal{E}_1\mathcal{E}_2\mathcal{E}_3\mathcal{E}_4$	true	6	no result	> 10000

Computing Hopf bifurcation in Methylene Blue Oscillator System:

$$
\begin{aligned}
MB^+ + HS^- &\longrightarrow MB + HS \\
H_2O + MB + HS^- &\longrightarrow MBH + HS + OH^- \\
HS + OH^- + MB^+ &\longrightarrow MB + S + H_2O \\
H_2O + 2MB &\longrightarrow MB^+ + MBH + OH^- \\
HS^- + O_2 &\longrightarrow HS + O_2^- \\
HS + O_2 + OH^- &\longrightarrow O_2^- + S + H_2O \\
H_2O + HS^- + O_2^- &\nrightarrow HO_2^- + HS + OH^- \\
O_2^- + HS &\longrightarrow HO_2^- + S \\
H_2O_2 + 2HS^- &\longrightarrow 2HS + 2OH^- \\
MB + O_2 &\longrightarrow MB^+ + O_2^- \\
HS^- + MB + H_2O_2 &\longrightarrow MB^+ + HS + 2OH^- \\
OH^- + 2HS &\longrightarrow HS^- + S + H_2O \\
MB + HS &\longrightarrow MBH + S \\
H_2O + MBH + O_2^- &\longrightarrow MB + H_2O_2 + OH^- \\
&\longrightarrow O_2
\end{aligned}
$$

Computing Hopf bifurcation in Methylene Blue Oscillator System:

- The MBO reaction system contains 15 reactions and 11 species (not counting water)

O_2, O_2^-, HS, MB^+, MB, MBH, HS^-, OH^-, S, H_2O_2 and HO_2^-.

- It may be reduced to a six dimensional system by considering only the essential species:

O_2, O_2^-, HS, MB^+, MB and MBH.

- The flux cone of this model is spanned by 31 extreme currents

Computing Hopf bifurcation in Methylene Blue Oscillator System using HoCoQ:

We tried to compute Hopf bifurcation in all subsystems involving 2-faces and 3-faces using our original approach HoCoQ, but the generated quantified formulae could not be solved by quantifier elimination, even with main memory up to 500 GB and computation times up to one week.

Computing Hopf bifurcation in Methylene Blue Oscillator System using HoCaT method:

- Using HoCaT method we could find Hopf bifurcation in 30% of the cases involving 2- faces.

- For 67% of the cases it could be excluded that the resulting polynomial has a zero.

- In only 3% of the cases no definite answer could be obtained.

Thank you for your attention !

errami@uni-kassel.de

Algorithm 3: Computing Hopf Bifurcations in Chemical Reaction Networks Using Reaction Coordinates

Input: A chemical reaction network $\mathcal{N}$ with $\dim(\mathcal{N}) = n$.

Output: (L_t, L_f, L_u) as follows: L_t is a list of subsystems containing a Hopf bifurcation, L_f is a list of subsystems in which its occurrence is excluded, and L_u is a list of subsystems for which the incomplete sub-procedure pzerop fails.

begin
  $L_t = \emptyset$
  $L_f = \emptyset$
  $L_u = \emptyset$
  generate the stoichiometric matrix $\mathcal{S}$ and the kinetic matrix $\mathcal{K}$ of $\mathcal{N}$
  compute the minimal set $\mathcal{E}$ of the vectors generating the flux cone
  for $d = 1 \ldots n$ **do**
    compute all d-faces (subsystems) $\{\mathcal{N}_i\}_i$ of the flux cone
  for *each subsystem* $\mathcal{N}_i$ **do**
    compute from $\mathcal{K}$, $\mathcal{S}$ the transformed Jacobian Jac_i of $\mathcal{N}_i$ in terms of convex coordinates j_i
    if Jac_i *is singular* **then**
      compute the reduced manifold of Jac_i calling the result also Jac_i
    compute the characteristic polynomial χ_i of Jac_i
    compute the $(n-1)^{\text{th}}$ Hurwitz determinant Δ_{n-1} of χ_i
    compute $\mathcal{F}_i := \mathrm{pzerop}(\Delta_{n-1}(j, x))$ using Algorithm 1
    if $\mathcal{F}_i = 1$ **or** $\mathcal{F}_i$ *is of the form* (π, ν) **then**
      $L_t := L_t \cup \{\mathcal{N}_i\}$
    else if $\mathcal{F}_i = +$ **or** $\mathcal{F}_i = -$ **then**
      $L_f := L_f \cup \{\mathcal{N}_i\}$
    else if $\mathcal{F}_i = \perp$ **then**
      $L_u := L_u \cup \{\mathcal{N}_i\}$
  return (L_t, L_f, L_u)

Algorithm 1: pzerop

Input: $f \in \mathbb{Z}[x_1, \ldots, x_m]$

Output: One of the following:

(A) 1, which means that $f(1, \ldots, 1) = 0$.

(B) (π, ν), where $\nu = (p, f(p))$ and $\pi = (q, f(q))$ for $p, q \in]0, \infty[^m$, which means that $f(p) < 0 < f(q)$. Then there is a zero on $]0, \infty[^m$ by the intermediate value theorem.

(C) $+$, which means that f has been identified as positive definite on $]0, \infty[^m$. Then there is no zero on $]0, \infty[^m$.

(D) $-$, which means that f has been identified as negative definite on $]0, \infty[^m$. Then there is no zero on $]0, \infty[^m$.

(E) $\perp$, which means that this incomplete procedure failed.

```
1  begin
2      f_1 := f(1, ..., 1)
3      if f_1 = 0 then
4          return 1
5      else if f_1 < 0 then
6          π := pzerop_1(f)
7          ν := ((1, ..., 1), f_1)
8          if π ∈ {⊥, −} then
9              return π
10         else
11             return (ν, π)
12     else
13         π := ((1, ..., 1), f_1)
14         ν' := pzerop_1(−f)
15         if ν' = ⊥ then
16             return ⊥
17         else if ν' = − then
18             return +
19         else
20             (p, f(p)) := ν'
21             ν := (p, −f(p))
22             return (ν, π)
```

BI

Algorithm 2: pzerop_1

Input: $g \in \mathbb{Z}[x_1, \ldots, x_m]$

Output: One of the following:

(A) $\pi = (q, g(q))$, where $q \in]0, \infty[^m$ with $0 < g(q)$.

(B) $-$, which means that g has been identified as negative definite on $]0, \infty[^m$. Then there is no zero on $]0, \infty[^m$.

(C) $\perp$, which means that this incomplete procedure failed.

```
1  begin
2      F+ := { d ∈ frame(g) | sgn(d) = 1 }
3      if F+ = ∅ then
4          return −
5      foreach (d_1, ..., d_m) ∈ F+ do
6          L := {d_1 n_1 + ··· + d_m n_m − c = 0}
7          foreach (e_1, ..., e_m) ∈ frame(g) \ F+ do
8              L := L ∪ {e_1 n_1 + ··· + e_m n_m − c ≤ −1}
9          if L is feasible with solution (n_1, ..., n_m, c) ∈ Q^{m+1} then
10             n := the principal denominator of n_1, ..., n_m
11             (N_1, ..., N_m) := (n n_1, ..., n n_m) ∈ Z^m
12             ḡ := g[x_1 ← ω^{N_1}, ..., x_m ← ω^{N_m}] ∈ Z(ω)
13             assert lc(ḡ) > 0 when using non-exact arithmetic in the LP solver
14             k := min{ k ∈ N | ḡ(2^k) > 0 }
15             return ((2^{kN1}, ..., 2^{kN_m}), ḡ(2^k))
16     return ⊥
```