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

Hassan Errami[1], Markus Eiswirth[2], Dima Grigoriev[3], Werner M. Seiler[1], Thomas Sturm[4], and Andreas Weber[5]

[1] Institut für Mathematik, Universität Kassel, Kassel, Germany;
errami@uni-kassel.de; seiler@mathematik.uni-kassel.de
[2] Fritz-Haber-Institut der Max-Planck-Gesellschaft, Berlin, Germany
and Ertl Center for Electrochemisty and Catalysis, GIST, South Korea;
eiswirth@fhi-berlin.mpg.de
[3] CNRS, Mathématiques, Université de Lille, Villeneuve d'Ascq, France;
Dmitry.Grigoryev@math.univ-lille1.fr
[4] Max-Planck-Institut für Informatik, Saarbrücken, Germany;
sturm@mpi-inf.mpg.de
[5] Institut für Informatik II, Universität Bonn, Bonn, Germany;
weber@cs.uni-bonn.de

**Abstract.** We build on our previous work to compute Hopf bifurcation fixed points for chemical reaction systems on the basis of reaction coordinates. For determining the existence of Hopf bifurcations the main algorithmic problem is to determine whether a single multivariate polynomial has a zero for positive coordinates. For this purpose we provide heuristics on the basis of the Newton polytope that ensure the existence of positive and negative values of the polynomial for positive coordinates. We apply our method to the example of the Methylene Blue Oscillator (MBO).

## 1 Introduction

In this paper we build on our previous work [1] to compute Hopf bifurcation fixed points for chemical reaction systems on the basis of reaction coordinates. In that previous work algorithmic ideas introduced by El Kahoui and Weber [2], which already had been used for mass action kinetics of small dimension [3], have been combined with methods of stoichiometric network analysis (SNA) introduced by Clarke in 1980 [4], which had been used in several "hand computations" in a semi-algorithmic way for parametric systems. The most elaborate of these computations haven been described in [5]. The algorithmic method presented in [1] uses and combines the ideas of these methods and extends them to a new approach for computing Hopf bifurcation in complex systems using reaction coordinates also allowing systems with linear constraints.

However, the criteria used for determining Hopf bifurcation fixed points with empty unstable manifold involving an equality condition on the principal minor

$\Delta_{n-1} = 0$ in conjunction with inequality conditions on $\Delta_{n-2} > 0 \wedge \cdots \wedge \Delta_1 > 0$ and positivity conditions on the variables and parameters still turned out to be hard problems for general real quantifier elimination procedures even for moderate dimensions.

In this paper we use the rather basic observation that when using convex coordinates, the condition for the existence of Hopf bifurcation fixed points is given by the single polynomial equation $\Delta_{n-1} = 0$ together with positivity conditions on the convex coordinates. This allows to delay or even drop a test for having empty unstable manifold on already determined witness points for Hopf bifurcations. Hence the main algorithmic problem is to determine whether a single multivariate polynomial has a zero for positive coordinates.

For this purpose we provide in Sect. 2 heuristics on the basis of the Newton polytope, which ensure the existence of positive and negative values of the polynomial for positive coordinates. In Sect. 3 we apply our method to the example of the Methylene Blue Oscillator (MBO).

As we are continuing here the work in [1] we allow ourselves to refer the reader to that publication for basic definitions as well as for an introduction of our software toolchains.

## 2 Condition for a Hopf bifurcation

Consider a parameterized autonomous ordinary differential equation of the form $\dot{x} = f(u, x)$ with a scalar parameter $u$. Due to a classical result by Hopf, this system exhibits at the point $(u_0, x_0)$ a Hopf bifurcation, i.e. an equilibrium transforms into a limit cycle, if $f(u_0, x_0) = 0$ and the Jacobian $D_x f(u_0, x_0)$ has a simple pair of pure imaginary eigenvalues and no other eigenvalues with zero real parts [6, Thm. 3.4.2].[6] The proof of this result is based on the center manifold theorem. From a physical point of view, the most interesting case is that the unstable manifold of the equilibrium $(u_0, x_0)$ is empty. However, for the mere existence of a Hopf bifurcation, this assumption is not necessary.

In [2] it is shown that for a parameterized vector field $f(u, x)$ and the autonomous ordinary differential system associated with it, there is a semi-algebraic description of the set of parameter values for which a Hopf bifurcation (with empty unstable manifold) occurs. Specifically, this semi-algebraic description can be expressed by the following first-order formula:

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

In this formula $a_n$ is $(-1)^n$ times the Jacobian determinant of the matrix $Df(u, x)$, and the $\Delta_i(u, x)$'s are the $i^{\text{th}}$ Hurwitz determinants of the characteristic polynomial of the same matrix $Df(u, x)$.

[6] We ignore here the non-degeneracy condition that this pair of eigenvalues crosses the imaginary axis transversally, as it is always satisfied in realistic models.

The proof uses a formula of Orlando [7], which is discussed also in several monographs, e.g., [8, 9]. However, a closer inspection of the two parts of the proof of [2, Theorem 3.5] shows even the following: for a fixed points (given in possibly parameterized form) the condition that there is a pair of purely complex eigenvalues is given by the condition $\Delta_{n-1}(u,x) = 0$ and the condition that all other eigenvalues have negative real part is given by $\Delta_{n-2}(u,x) > 0 \wedge \cdots \wedge \Delta_1(u,x) > 0$. This statement (without referring to parameters explicitly) is also contained in [10, Theorem 2], in which a different proof technique is used.

Hence if we drop the condition on Hopf bifurcation points that they have empty unstable manifold a semi-algebraic description of the set of parameter values for which a Hopf bifurcation occurs for the system is given by the following formula:

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

Notice that when the quantifier elimination procedure yields sample points for existentially quantified formulae, then the condition $\Delta_{n-2}(u,x) > 0 \wedge \cdots \wedge \Delta_1(u,x) > 0)$ can be tested for the sample points later on, i.e. one can then test whether this Hopf bifurcation fixed points has empty unstable manifold. Such sample points are yielded, e.g., by virtual substitution-based methods in Redlog [12–15].

As an example consider the famous *Lorenz System* named after Edward Lorenz at MIT, who first investigated this system as a simple model arising in connection with fluid convection [16, 6, 17]. It is given by the following system of ODEs:

$$\begin{aligned}
\dot{x}(t) &= \alpha\,(y(t) - x(t)) \\
\dot{y}(t) &= r\,x(t) - y(t) - x(t)\,z(t) \\
\dot{z}(t) &= x(t)\,y(t) - \beta\,z(t).
\end{aligned}$$

After imposing positivity conditions on the parameters the following answer is obtained using a combination of Redlog and SLFQ as described in [18] for the test of a Hopf bifurcation fixed points:

$$\begin{aligned}
(-\alpha^2 - \alpha\beta + \alpha r - 3\alpha - \beta r - r = 0 \vee -\alpha\beta + \alpha r - \alpha - \beta^2 - \beta = 0) \wedge \\
-\alpha^2 - \alpha\beta + \alpha r - 3\alpha - \beta r - r \leq 0 \wedge \\
\beta > 0 \wedge \alpha > 0 \wedge -\alpha\beta + \alpha r - \alpha - \beta^2 - \beta \geq 0.
\end{aligned}$$

When testing, in contrast, for Hopf bifurcation fixed points with empty unstable manifold we obtain the following formula, which is not equivalent to the one above:

$$\begin{aligned}
\alpha^2 + \alpha\beta - \alpha r + 3\alpha + \beta r + r = 0 \wedge \alpha r - \alpha - \beta^2 - \beta \geq 0 \wedge \\
2\alpha - 1 \geq 0 \wedge \beta > 0.
\end{aligned}$$

Hence for the case of the Lorenz system not all Hopf bifurcation fixed points have an unstable empty manifold.

This approach using concentration coordinates has been applied in [3].

### 2.1 Using Reaction Coordinates

In [1] a new approach for computing Hopf bifurcations in complex systems has been given, which uses reaction coordinates in contrast to concentration coordinates and also allows systems with linear constraints. The Jacobian matrix of a subsystem formed by $d$-faces is given by the following equation, where $\mathcal{S}$, $\mathcal{K}$ and $\mathcal{E}$ denote the stoichiometric matrix, kinetic matrix, and the set of extreme currents, respectively.

$$\mathrm{Jac}(x) = \mathcal{S}\mathrm{diag}(\sum_{i}^{d} j_i\mathcal{E}_i)\mathcal{K}^t\mathrm{diag}(1/x_1, ..., 1/x_n).$$

For checking for the existence of Hopf bifurcation fixed points without requiring empty unstable manifolds we have to decide the satisfiability of the following formula:

$$\Delta_{n-1}(j, x) = 0 \wedge j_1 \geq 0 \wedge \cdots j_d \geq 0 \wedge x_1 > 0 \wedge \cdots x_n > 0.$$

Hence the algorithmic task is to determine whether the single multivariate polynomial equation $\Delta_{n-1}(j, x) = 0$ has a solution subject to the given sign conditions on the variables. Note that whenever a satisfiability test provides sample points in the satisfiable case, then those points can be tested for having an empty unstable manifold by substituting them into the condition

$$\Delta_{n-2}(j, x) > 0 \wedge \cdots \wedge \Delta_1(j, x) > 0.$$

In the next section we are going to discuss sufficient conditions and their efficient algorithmic realizations.

### 2.2 Sufficient Conditions for a Positive Solution of a Single Multivariate Polynomial Equation

The method discussed in this section is summarized in an algorithmic way in Alg. 1, which uses Alg. 2 as a sub-algorithm.

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 it suffices by the intermediate value theorem to find $p \in ]0, \infty[^m$ such that $f(p) > 0$. Similarly, if $f_1 > 0$ it suffices to find $p \in ]0, \infty[^m$ such that $(-f)(p) > 0$. 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 are going to accompany the description of our method with the example $g_0 = -2x_1^6 + x_1^3x_2 - 3x_1^3 + 2x_1x_2^2 \in \mathbb{Z}[x_1, x_2]$. Fig. 1 shows an implicit plot of this polynomial. In addition to its variety, $g_0$ has three sign invariant regions, one bounded one and two unbounded ones. One of the unbounded regions contains

**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) $(\pi, \nu)$, 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)
```

**Algorithm 2:** $\text{pzerop}_1$

**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 ⊥
```

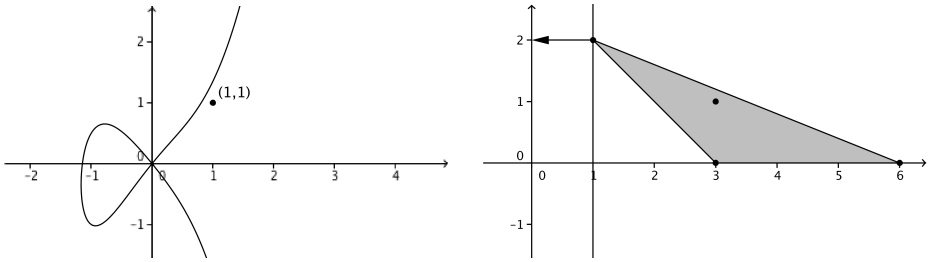


**Fig. 1.** We consider $g_0 = -2x_1^6 + x_1^3x_2 - 3x_1^3 + 2x_1x_2^2$. The left hand shows the variety $g_0 = 0$. The right hand side shows the frame, the Newton polytope, and a separating hyperplane for the positive monomial $2x_1x_2^2$ with its normal vector.

our initial test point $(1, 1)$, for which we find that $g_0(1, 1) = -2 < 0$. Thus our goal is to find one point $p \in ]0, \infty[^2$ such that $g_0(p) > 0$.

In the spirit of tropical geometry—and we refer to [19] as a standard reference with respect to its applications for polynomial system solving—we take an abstract view of

$$g = \sum_{d \in D} a_d x^d := \sum_{(d_1,\ldots,d_m) \in D} a_{d_1,\ldots,d_m} x_1^{d_1} \cdots x_m^{d_m}$$

as the set $\mathrm{frame}(g) = D \subseteq \mathbb{N}^m$ of all exponent vectors of the contained monomials. For each $d \in \mathrm{frame}(g)$ we are able to determine $\mathrm{sgn}(d) := \mathrm{sgn}(a_d) \in \{-1, 1\}$. The set of vertices of the convex hull of the frame is called the *Newton polytope* $\mathrm{newton}(g) \subseteq \mathrm{frame}(g)$. In fact, the existence of at least one point $d^* \in \mathrm{newton}(g)$ with $\mathrm{sgn}(d^*) = 1$ is sufficient for the existence of $p \in ]0, \infty[^m$ with $g(p) > 0$.

In our example we have $\mathrm{frame}(g_0) = \{(6, 0), (3, 1), (3, 0), (1, 2)\}$ and $\mathrm{newton}(g_0) = \{(6, 0), (3, 0), (1, 2)\} \subsetneq \mathrm{frame}(g_0)$. We are particularly interested in $d^* = (d_1^*, d_2^*) = (1, 2)$, which is the only point there with a positive sign as it corresponds to the monomial $2x_1x_2^2$.

In order to understand this sufficient condition, we are now going to compute from $d^*$ and $g$ a suitable point $p$. We construct a hyperplane $H : n^T x = c$ containing $d^*$ such that all other points of $\mathrm{newton}(g)$ are not contained in $H$ and lie on the same side of $H$. We choose the normal vector $n \in \mathbb{Q}^m$ such that it points into the half-space not containing the Newton polytope. The vector $c \in \mathbb{R}^m$ is such that $\frac{c}{|n|}$ is the offset of $H$ from the origin in the direction of $n$. We may assume w.l.o.g. that $n \in \mathbb{Z}^m$.

In our example $H$ is the line $x = 1$ given by $n = (-1, 0)$ and $c = -1$. Fig. 1 pictures the situation.

Considering the standard scalar product $\langle \cdot | \cdot \rangle$, it turns out that generally $\langle n | d^* \rangle = \max\{ \langle n | d \rangle \mid d \in \mathrm{newton}(g) \}$, and that this maximum is strict. For the monomials of the original polynomial $g = \sum_{d \in D} a_d x^d$ and a new variable $\omega$ this observation translates via the following identity:

$$\bar{g} = g[x \leftarrow \omega^n] = \sum_{d \in D} a_d \omega^{\langle n | d \rangle} \in \mathbb{Z}(\omega).$$

Hence plugging into $\bar{g}$ a number $\beta \in \mathbb{R}$ corresponds to plugging into $g$ the point $\beta^n \in \mathbb{R}^m$ and from our identity we see that in $\bar{g}$ the exponent $\langle n|d^*\rangle$ corresponding to our chosen point $d^* \in \text{newton}(g)$ dominates all other exponents so that for large $\beta$ the sign of $\bar{g}(\beta) = g(\beta^n)$ equals the positive sign of the coefficient $a_{d^*}$ of the corresponding monomial. To find a suitable $\beta$ we successively compute $\bar{g}(2^k)$ for increasing $k \in \mathbb{N}$.

In our example we obtain $\bar{g} = 2\omega^{-1} - 2\omega^{-3} - 2\omega^{-6}$, we obtain $\bar{g}(1) = -2 \leq 0$, but already $\bar{g}(2) = \frac{23}{32} > 0$. In terms of the original $g$ this corresponds to plugging in the point $p = 2^{(-1,0)} = \left(\frac{1}{2}, 1\right) \in ]0, \infty[^2$.

It remains to be clarified how to construct the hyperplane $H$. Consider $\text{frame}(g) = \{\, (d_{i1}, \ldots, d_{im}) \in \mathbb{N}^m \mid i \in \{1, \ldots, k\}\,\}$. If $\text{sgn}(d) = -1$ for all $d \in \text{frame}(g)$, then we know that $g$ is negative definite on $]0, \infty[^m$. Otherwise assume without loss of generality that $\text{sgn}(d_{11}, \ldots, d_{1m}) = 1$. We write down the following linear program:

$$\begin{pmatrix} d_{11} & \ldots & d_{1m} & -1 \end{pmatrix} \cdot \begin{pmatrix} n_1 \\ \vdots \\ n_m \\ c \end{pmatrix} = 0, \quad I = \begin{pmatrix} d_{21} & \ldots & d_{2m} & -1 \\ \vdots & \ddots & \vdots & \vdots \\ d_{k1} & \ldots & d_{km} & -1 \end{pmatrix} \cdot \begin{pmatrix} n_1 \\ \vdots \\ n_m \\ c \end{pmatrix} \leq -1.$$

Notice that in our system of inequalities we can use the LP-friendly conditions $I \leq -1$ in favor of the more natural conditions $I < 0$. Since the distance of the points $(d_{21}, \ldots, d_{2m})$, $\ldots$, $(d_{k1}, \ldots, d_{km})$ to the desired hyperplane $H$ is scaled by $|(n_1, \ldots, n_m)|$, there is a sufficient degree of freedom in the choice of $c$ in combination with $(n_1, \ldots, n_m)$ to achieve values smaller or equal to $-1$ in the feasible case. Our program is feasible if and only if $(d_{11}, \ldots, d_{1m}) \in \text{newton}(g)$. In the negative case, we know that $(d_{11}, \ldots, d_{1m}) \in \text{frame}(g) \setminus \text{newton}(g)$, and we iterate with another $d \in \text{frame}(g)$ with $\text{sgn}(d) = 1$. If we finally fail on all such $d$, then our incomplete algorithm has failed. In the positive case, the solution provides a normal vector $n = (n_1, \ldots, n_m)$ and the offset $c$ for a suitable hyperplane $H$. Our linear program can be solved using any standard LP solver. For our purposes here we have used Gurobi[7]; it turns out that the dual simplex of Glpsol[8] performs quite similarly on the input considered here.

For our example $g_0 = -2x_1^6 + x_1^3x_2 - 3x_1^3 + 2x_1x_2^2$, we generate the linear program

$$\begin{aligned} n_1 + 2n_2 - c &= 0 \\ 6n_1 - c &\leq -1 \\ 3n_1 + n_2 - c &\leq -1 \\ 3n_1 - c &\leq -1, \end{aligned}$$

for which Gurobi computes the solution $n = (n_1, n_2) = (-0.5, 0)$, $c = -0.5$. Notice that the solutions obtained from the LP solvers are typically floats, which we lift to integer vectors by suitable rounding and gcd computations.

[7] www.gurobi.com
[8] www.gnu.org/software/glpk

Note that we do not explicitly construct the convex hull newton($g$) of frame($g$) although there are advanced algorithms and implementations like QuickHull[9] available for this. Instead we favor a linear programming approach for several reasons. Firstly, we do not really need the quite comprehensive information, comprising, e.g. adjacency, obtained from such algorithms. For our purposes, it is rather sufficient to find one vertex with a positive sign the convex hull. Secondly, for the application discussed here it turns out that there typically exist only few (around 10%) such candidate points at all. Finally, it is known that for high dimensions the subset of frame($g$) establishing vertices of the convex hull gets comparatively large. Practical experiments using QuickHull on our data support these theoretical considerations.

### 2.3 Summarizing the Algorithm for Checking the Existence of Hopf Bifurcations

Computing Hopf bifurcation fixed points for high-dimensional systems and systems with conservation laws had turned out to be difficult in practice. To overcome this difficulty for systems arising from chemical reaction networks we introduced in our previous paper [1] an algorithm based on using reaction coordinates instead of concentration coordinates and applying real quantifier elimination for testing satisfiability. This enabled us to decide the occurrence of Hopf bifurcation in various chemical systems even with conservation laws. For some chemical networks with complex dynamics, however, it remained difficult to finally process the obtained quantified formulae with the currently available quantifier elimination packages; one hard example is the Methylene Blue Oscillator (MBO) discussed in the next section. To conclude this section, we are going to summarize our new efficient algorithmic approach for checking for Hopf bifurcation in complex chemical systems. Again using reaction coordinates, our approach here improves the previous one by simplifying the formulas expressing Hopf-existence conditions as shown in Subsection 2.1 and solving them by the method described in Subsection 2.2. The pre-processing step and the steps 2-6 presented in [1] remain the same. After computing the characteristic polynomial of each Jacobian matrix, we compute the $(n-1)^{\text{th}}$ Hurwitz determinant of the characteristic polynomial, and we apply Alg. 1 to check for positive solutions of the respective polynomial equations $\Delta_{n-1} = 0$. Alg. 3 outlines our new approach in an algorithmic fashion.

## 3 Algorithmic Determination of Hopf bifurcations in the Methylene Blue Oscillator System

As a complex example we consider the autocatalytic system *Methylen Blue Oscillator (MBO)*, which is defined by the following reaction equations:

[9] www.qhull.org

**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** $d = 1 \dots n$ **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 $\mathrm{Jac}_i$ of $\mathcal{N}_i$ in terms of convex coordinates $j_i$
    **if** $\mathrm{Jac}_i$ *is singular* **then**
      compute the reduced manifold of $\mathrm{Jac}_i$ calling the result also $\mathrm{Jac}_i$
    compute the characteristic polynomial $\chi_i$ of $\mathrm{Jac}_i$
    compute the $(n-1)^{\text{th}}$ Hurwitz determinant $\Delta_{n-1}$ of $\chi_i$
    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* $(\pi, \nu)$ **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** $(L_t, L_f, L_u)$

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

The MBO reaction system contains eleven species (not counting water) and fifteen reactions $O_2$, $O_2^-$, 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 pre-processing step of our algorithm yields the following two matrices describing the reaction laws: stoichiometric matrix $\mathcal{S}$ and kinetic matrix $\mathcal{K}$.

$$
\mathcal{S} = \begin{pmatrix}
1 & -1 & 1 & -2 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & 1 & 0\\
-1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0\\
1 & 1 & -1 & 0 & 1 & -1 & 1 & -1 & 2 & 0 & 1 & -2 & -1 & 0 & 0\\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0\\
0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & -1 & 0
\end{pmatrix}
$$

$$
\mathcal{K} = \begin{pmatrix}
0 & 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0\\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 2 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}.
$$

The flux cone of this Model is generated by 31 extreme currents. We tried to compute Hopf bifurcation in all subsystems involving 2-faces and 3-faces using our original approach described in [1], 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.

Using our new approach described here, in only 3% of the cases no definite answer could be obtained; in 67% of the cases it could be excluded that the resulting polynomial has a zero, whereas in 30% of the cases it could be verified that the resulting polynomial has a zero. Hence for at least 30% of the 2-faces there are Hopf bifurcations on these faces. Recall that the positive answer for at least one of the cases guarantees the existence of a Hopf bifurcation for the original system in spite of the fact that there are cases without definite answer.

The algorithmic test sketched in Section 2.2 can be parallelized easily for the different faces. Using 60 hyper-cores on a 2.4 GHz Intel Xeon E5-4640 running Debian Linux 64 bit the computation for all instances resulting from 2-faces could be completed in less than 90 seconds of wall clock time, which implies that also the worst-case computation time for the single instances has been at most 90 seconds.

## Acknowledgements

This research was supported in part by *Deutsche Forschungsgemeinschaft* within SPP 1489 and by the German Transregional Collaborative Research Center SFB/TR 14 AVACS. Thomas Sturm would like to thank B. Barber for his support with QuickHull and convex hull computation and W. Hagemann and M. Košta for helpful discussions on linear programming aspects.

## References

1. Errami, H., Seiler, W.M., Eiswirth, M., Weber, A.: Computing Hopf bifurcations in chemical reaction networks using reaction coordinates. In Gerdt, V.P., Koepf, W., Mayr, E.W., Vorozhtsov, E.V., eds.: Computer Algebra in Scientific Computing. Volume 7442 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 84–97
2. El Kahoui, M., Weber, A.: Deciding Hopf bifurcations by quantifier elimination in a software-component architecture. Journal of Symbolic Computation **30**(2) (August 2000) 161–179
3. Sturm, T., Weber, A., Abdel-Rahman, E.O., El Kahoui, M.: Investigating algebraic and logical algorithms to solve Hopf bifurcation problems in algebraic biology. Mathematics in Computer Science **2**(3) (March 2009) 493–515
4. Clarke, B.L.: Stability of Complex Reaction Networks. Volume XLIII of Advances in Chemical Physics. Wiley Online Library (1980)
5. Gatermann, K., Eiswirth, M., Sensse, A.: Toric ideals and graph theory to analyze Hopf bifurcations in mass action systems. Journal of Symbolic Computation **40**(6) (2005) 1361–1382
6. Guckenheimer, J., Holmes, P.: Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields. Volume 42 of Applied Mathematical Sciences. Springer-Verlag (1990)
7. Orlando, L.: Sul problema di hurwitz relativo alle parti reali delle radici di un'equazione algebrica. Mathematische Annalen **71**(2) (1911) 233–245
8. Gantmacher, F.R.: Application of the Theory of Matrices. Interscience Publishers, New York (1959)

9. Porter, B.: Stability Criteria for Linear Dynamical Systems. Academic Press, New York (1967)
10. Yu, P.: Closed-form conditions of bifurcation points for general differential equations. International Journal of Bifurcation and Chaos **15**(4) (April 2005) 1467–1483
11. Liu, W.M.: Criterion of Hopf bifurcations without using eigenvalues. Journal of Mathematical Analysis and Applications **182**(1) (1994) 250–256
12. Dolzmann, A., Sturm, T.: Redlog: Computer algebra meets computer logic. ACM SIGSAM Bulletin **31**(2) (June 1997) 2–9
13. Weispfenning, V.: Quantifier elimination for real algebra—the quadratic case and beyond. Applicable Algebra in Engineering Communication and Computing **8**(2) (February 1997) 85–101
14. Dolzmann, A., Sturm, T., Weispfenning, V.: Real quantifier elimination in practice. In Matzat, B.H., Greuel, G.M., Hiss, G., eds.: Algorithmic Algebra and Number Theory. Springer, Berlin (1998) 221–247
15. Dolzmann, A., Seidl, A., Sturm, T.: Efficient projection orders for CAD. In Gutierrez, J., ed.: Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation (ISSAC 2004). ACM Press, New York, NY (July 2004) 111–118
16. Lorenz, E.N.: Deterministic nonperiodic flow. Journal of the Atmospheric Sciences **20**(2) (1963) 130–141
17. Rand, R.H., Armbruster, D.: Perturbation Methods, Bifurcation Theory and Computer Algebra. Volume 65 of Applied Mathematical Sciences. Springer-Verlag (1987)
18. Brown, C.W., El Kahoui, M., Novotni, D., Weber, A.: Algorithmic methods for investigating equilibria in epidemic modeling. Journal of Symbolic Computation **41**(11) (2006) 1157–1173
19. Sturmfels, B.: Solving Systems of Polynomial Equations. AMS, Providence, RI (2002)