



UNIVERSIDAD DE BUENOS AIRES
Facultad de Ciencias Exactas y Naturales
Departamento de Matemática

Tesis de Licenciatura

**Estabilidad y oscilaciones en la red ERK: un paseo por la matemática
para estudiar la regulación celular**

Raigorodsky Pedro

Directora: Mercedes Pérez Millán

Marzo de 2023

Gracias a mis viejos por bancarme en todo. A mis hermanos, Katu, Vicky y Simón. A mis abuelos, tíos, primos y toda mi familia que siempre están para mí.

Gracias a Lucas, Chano, Julián, Huanca, Rodri. No tengo palabras para decir los buenos amigos que son, y también fueron parte de este proceso. Ustedes también me llevaron hasta acá.

Gracias a Dante y Cami. Mi carrera no hubiera sido lo mismo sin ustedes para hablar, colaborar, discutir, y divertirnos. Lo mismo con Juli, Manu, Sol, Juja, Sol, Martín, Guido, amigos y compañeros con los que he aprendido una barbaridad, muchas veces más que en las clases. Flashearla es, creo, la mejor parte de estudiar esta carrera. También incluyo al equipo del remando, a compañeros de otras carreras, y mucha otra gente que me ha dado mucha alegría durante estos años.

Gracias a todos los increíbles docentes que tuve a lo largo de la carrera, desde el CBC hasta el último año de cursada, no los nombro porque sería más largo que mi tesis. Muchos estuvieron ahí en los momentos más duros, listos para ayudar con lo que pudieran. Ustedes son los que hacen que la carrera y el departamento sea lo que sea, y me alegra poder haber sido colega y participar del ciclo docente; y espero poder seguir siéndolo a lo largo de mi vida.

Gracias a Mercedes por dirigirme. Me ayudaste muchísimo y me tuviste muchísima paciencia. Siempre sentí que estuviste presente a lo largo de todo el proceso, dispuesta a ayudarme con lo que sea. Me alegra mucho haber compartido este momento y me llevo muchas enseñanzas. Sin ninguna duda, este trabajo también es fruto de todo el esfuerzo, dedicación y acompañamiento que le pusiste. Realmente, muchas gracias.

Gracias a Alicia y Tico por aceptar ser jurados y dedicarle su tiempo a mi trabajo.

Por último, gracias a la UBA por darme la oportunidad de cursar esta carrera.

Índice general

1. Introducción	1
1.1. Redes de reacciones químicas	2
1.1.1. Estados Estacionarios	6
1.2. La red de la proteína ERK	9
1.2.1. La cascada de señalización de ERK	9
1.2.2. Modelado de ERK	11
2. Herramientas Teóricas	19
2.1. Herramientas para estudiar estabilidad	19
2.1.1. Multiestacionariedad y Funciones Críticas	19
2.2. Bifurcaciones de Hopf	29
2.3. Introducción a los polítopos	41
2.3.1. Polítopos de Newton	44
2.4. Resultantes	46
3. Estudio de la Dinámica de ERK.	49
3.1. Biestabilidad	49
3.2. Bifurcaciones de Hopf y Oscilaciones.	51
3.3. Máxima Cantidad de Estados Estacionarios	54
3.3.1. Método por Resultantes	56
4. Programas	59
4.1. Método de Polítopos de Newton.	59
4.1.1. Implementación	61
4.1.2. Otra Aplicación	62
4.2. Otros programas	63
A. Códigos	65
A.1. Código del Algoritmo de Polítopo de Newton	65
A.1.1. Primera Versión del Algoritmo	67
A.1.2. Segunda Versión del Algoritmo	69
A.1.3. Aplicación a Otro Sistema	75
A.2. Código de Otros Programas	78

Capítulo 1

Introducción

El objetivo de este trabajo es estudiar la dinámica de la red de la proteína ERK (*extracellular signal-regulated kinase*, que se traduce como quinasa regulada por señal extracelular) por medio de distintas herramientas matemáticas. Estas proteínas abundan en las células y están involucradas en varias funciones como por ejemplo la regulación de la división celular, y su camino se activa por muchos estímulos diferentes como infecciones virales, factores de crecimiento, carcinógenos, etc.

Nos basaremos mayormente en los trabajos de Conradi et al. [5] y Obatake et al. [20]. Estudiaremos distintos modelos matemáticos de la red que pueden dar lugar a oscilaciones o biestabilidad.

En el Capítulo 1 introduciremos los conceptos básicos de redes de reacciones bioquímicas bajo cinética de acción de masas a partir de un modelo sencillo de interacciones entre proteínas. Hablaremos de estados estacionarios de las mismas, la estabilidad de estos estados, y la noción de multiestacionariedad y biestabilidad de una red. En la segunda sección introduciremos la red ERK, que será el objeto de estudio central de este trabajo. Daremos su contexto biológico, y presentaremos el modelo matemático que estudiaremos. También presentaremos subredes importantes, que son más simples que la red completa y son de interés. Mostraremos parametrizaciones de sus estados estacionarios y probaremos que no tienen estados estacionarios en el borde del ortante positivo.

En el Capítulo 2, presentaremos las herramientas matemáticas que utilizaremos para estudiar el modelo dado en el capítulo anterior. Usando teoría de grado [23], daremos un criterio para estudiar la multiestacionariedad de una red y un criterio útil para llevar propiedades de una subred a la red completa. Introduciremos la noción de Bifurcación de Hopf [17], un tipo de cambio de estabilidad en un sistema, que nos interesará estudiar en el contexto de la red de ERK por sus consecuencias biológicas. Para esto presentaremos el criterio de Routh-Hurwitz. Luego, daremos una introducción elemental a la teoría de los polítopos (basada en [26]) que nos será de utilidad para la sección computacional. Introducimos también las nociones de polítopo de Newton y volumen mixto de un polítopo. Finalmente, introducimos nociones de resultantes, que servirán para acotar el número de estados de una red.

En el Capítulo 3, usaremos las herramientas del capítulo anterior para estudiar la dinámica de la red ERK. Mostramos que para ciertas subredes de ERK tenemos biestabilidad y, con un

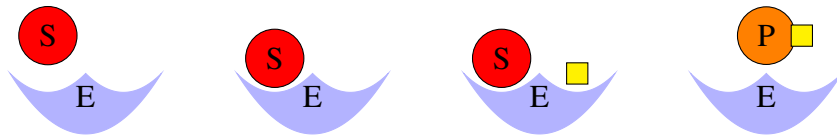
resultado previo podemos levantar la biestabilidad y multiestacionariedad a la red completa. También probaremos la existencia de multiestacionariedad para distintos niveles de procesatividad de la red, la existencia de bifurcaciones de Hopf y la posibilidad de coexistencia de bifurcaciones y biestabilidad dentro de una misma clase de compatibilidad. Acabamos el capítulo usando herramientas de volumen mixto y resultantes para acotar la máxima cantidad de estados estacionarios de una red.

El Capítulo 4 nos centraremos en los aspectos computacionales del trabajo. Hacemos una implementación completa del algoritmo presente en el Apéndice de Obatake et al. [20] en Python. Explicaremos el Algoritmo de Polítopo de Newton, demostraremos su correctitud, y haremos comentarios sobre la implementación del mismo. Además, usamos parte del algoritmo para mostrar la existencia de bifurcaciones de Hopf para una red mínima de [2]. Finalmente, presentaremos otros programas varios relacionados al Capítulo 3.

Por último, incluimos un Apéndice con el código realizado a lo largo de la tesis. Por motivos de formato, no todas las indentaciones son correctas para Python, en caso que se desee correr el código recomendamos abrir los archivos correspondientes en los cuadernos (formato .ipynb) adjuntos a esta tesis.

1.1. Redes de reacciones químicas

Nos centraremos en reacciones que se dan entre proteínas adentro de la célula, aunque la teoría que desarrollamos en esta sección es más general. El ejemplo que usaremos para ilustrar considera la interacción entre dos proteínas: una enzima (E) y un sustrato (S). La enzima se une al sustrato, produciendo una especie intermediaria (ES) y así se cataliza la reacción que transforma a S en el producto P . Podemos pensar en un proceso de fosforilación, donde E es una quinasa que agrega un grupo fosfato a la proteína S . La proteína P es, entonces, la forma fosforilada de la proteína S [1, Capítulo 3].



Las redes de reacciones químicas consisten en *especies*, *complejos* y *reacciones*. En nuestro ejemplo tenemos que las especies son $\{E, S, ES, P\}$ (que representan los compuestos químicos), los complejos son $\{E + S, ES, E + P\}$ y las reacciones son $\{E + S \rightarrow ES, ES \rightarrow E + S, ES \rightarrow E + P\}$. Es decir, que nuestras redes de reacciones químicas consisten en grafos dirigidos donde los vértices son los complejos y las aristas son las reacciones. El grafo que le corresponde a nuestro ejemplo es:



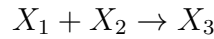
Más en general:

Definición 1.1.1 (Red de reacciones químicas). Sea $\mathcal{X} = \{X_1, \dots, X_s\}$ un conjunto de **especies**. Una red de reacciones químicas $G = (\mathcal{C}, \mathcal{R})$ es un grafo dirigido donde:

- Los **complejos** $c \in \mathcal{C}$ consisten en combinaciones lineales enteras no negativas de elementos de \mathcal{X} (es decir, $c = \sum_{i=1}^s \alpha_i X_i$ donde $\alpha_i \in \mathbb{N}_0$).
- El conjunto de **reacciones** $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$ es tal que no contiene pares de la forma (c, c) , y para todo complejo $c_1 \in \mathcal{C}$, existe $c_2 \in \mathcal{C}$ de modo que $(c_1, c_2) \in \mathcal{R}$ o $(c_2, c_1) \in \mathcal{R}$.

Utilizaremos x_1, \dots, x_s para denotar las concentraciones de las especies X_1, \dots, X_s respectivamente. Esto se suele notar $x_i = [X_i]$, $i = 1, \dots, s$.

Las concentraciones de las especies varían con el tiempo y modelaremos la dinámica de la red de reacciones químicas a través de la ley de cinética de acción de masas. Esta nos dice que la velocidad de reacción es proporcional al producto de las concentraciones de las especies reactivas. Por ejemplo, si nuestra red solo consiste en la reacción:

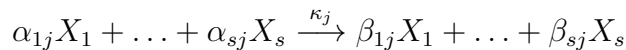


Entonces $\dot{x}_3(t) = \frac{dx_3}{dt}(t) = \kappa x_1(t)x_2(t)$ para algún $\kappa > 0$ (y de hecho $\dot{x}_1(t) = -\kappa x_1(t)x_2(t)$ y $\dot{x}_2(t) = -\kappa x_1(t)x_2(t)$). A este κ se lo denomina *constante de reacción*. Etiquetaremos las aristas del grafo con las constantes de reacción, obteniendo así una noción de la velocidad de las reacciones.

En lo que sigue generalizaremos este concepto a redes de varias especies y reacciones.

Definición 1.1.2 (Cinética). Sea $G = (\mathcal{C}, \mathcal{R})$ una red de reacciones químicas con $m \in \mathbb{N}$ reacciones r_1, r_2, \dots, r_m . A cada r_j le asignaremos $\kappa_j > 0$ denominada **constante de reacción**.

Cada reacción r_j se escribe de la forma:



donde $\alpha_{ij}, \beta_{ij} \in \mathbb{N}_0$. Si denotamos $\alpha_j = (\alpha_{1j}, \dots, \alpha_{sj})$ y $\beta_j = (\beta_{1j}, \dots, \beta_{sj})$, se suele notar a la reacción $\alpha_j \xrightarrow{\kappa_j} \beta_j$.

Definimos $N_{ij} = \beta_{ij} - \alpha_{ij}$ y tomamos a la matriz $N = (N_{ij})_{i,j} \in \mathbb{Z}^{s \times m}$, que se conoce como la **matriz de estequiometría**. Luego, por la ley de acción de masas, obtenemos el siguiente sistema de ecuaciones diferenciales autónomo:

$$\dot{x} = f(x) = N \begin{pmatrix} \kappa_1 x_1^{\alpha_{11}} \dots x_s^{\alpha_{s1}} \\ \vdots \\ \kappa_m x_1^{\alpha_{1m}} \dots x_s^{\alpha_{sm}} \end{pmatrix}, \quad (1.1)$$

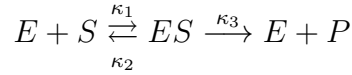
donde $x = (x_1, \dots, x_s)$.

A partir de la notación estándar, definimos $x^{\alpha_j} = x_1^{\alpha_{1j}} \dots x_s^{\alpha_{sj}}$, podemos reescribir lo anterior de la siguiente manera:

$$\dot{x} = N \begin{pmatrix} \kappa_1 x^{\alpha_1} \\ \vdots \\ \kappa_m x^{\alpha_m} \end{pmatrix}$$

Notemos que al definir f como $\dot{x} = f(x)$, obtenemos que cada f_i es un polinomio en $\mathbb{Q}[\kappa][x]$, para $i = 1, \dots, s$.

Ejemplo 1.1.1. Apliquemos estas definiciones a la red con la que comenzamos:



Denotemos $x_1 = [E]$, $x_2 = [S]$, $x_3 = [ES]$, $x_4 = [P]$. El sistema queda:

$$\begin{cases} \dot{x}_1 = -\kappa_1 x_1 x_2 + (\kappa_2 + \kappa_3) x_3 \\ \dot{x}_2 = -\kappa_1 x_1 x_2 + \kappa_2 x_3 \\ \dot{x}_3 = \kappa_1 x_1 x_2 - (\kappa_2 + \kappa_3) x_3 \\ \dot{x}_4 = \kappa_3 x_3 \end{cases}.$$

O escrito como en (1.1):

$$\dot{x} = \begin{pmatrix} -1 & 1 & 1 \\ -1 & 1 & 0 \\ 1 & -1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \kappa_1 x_1 x_2 \\ \kappa_2 x_3 \\ \kappa_3 x_3 \end{pmatrix}.$$

Una propiedad importante de los sistemas dinámicos bajo la ley de acción de masas es que tanto el ortante positivo, $\mathbb{R}_{>0}^s$, como su clausura, $\mathbb{R}_{\geq 0}^s$, son *positivamente invariantes*. Es decir, que si $x(0) \in \mathbb{R}_{>0}^s$ (respectivamente $x(0) \in \mathbb{R}_{\geq 0}^s$), entonces $x(t) \in \mathbb{R}_{>0}^s$ (respectivamente $x(t) \in \mathbb{R}_{\geq 0}^s$) para todo $t \geq 0$ donde x esté definida. Adaptamos una prueba de [18].

En primer lugar, observemos que en nuestro modelo, fijando una cordenada f_i , todos los coeficientes negativos provienen de reacciones que salen de un nodo con la especie X_i , y de hecho es un si y solo si. Por ende, cada ecuación f_i se puede escribir de la forma $p_i - x_i q_i$, donde p_i, q_i son polinomios con coeficientes no negativos. Esta caracterización de los sistemas que provienen de cinética de acción de masas se denomina *el lema Húngaro*.

Lema 1.1.1 (Lema Húngaro). *Dado $f = (f_1, \dots, f_s)$ con $f_i \in \mathbb{R}[x_1, \dots, x_s]$, luego $\dot{x} = f(x)$ es un sistema asociado a una red de reacciones bioquímicas modelado bajo cinética de acción de masas si y solo si $f_i = p_i - x_i q_i$ con p_i, q_i polinomios reales y con coeficientes no negativos.*

Usamos esta caracterización para demostrar que los sistemas son positivamente invariantes.

Proposición 1.1.1. *El sistema asociado a una red modelada bajo cinética de acción de masas es positivamente invariante.*

Demostración. Para cada $1 \leq i \leq s$, sabemos que $\dot{x}_i = p_i(x(t)) - x_i(t)q_i(x(t))$. Supongamos que $x(0) \in \mathbb{R}_{>0}^n$ pero que existe $T > 0$ y un i para el cual $x_i(T) < 0$. Por Bolzano, deben existir para cada una de estas coordenadas un $\tau_i > 0$ tal que $x_i(\tau_i) = 0$ y $x_i(t) > 0$ si $t < \tau_i$. Sea τ el menor de estos valores e i el índice correspondiente.

En tal caso, como todas las coordenadas son positivas, es claro que $p_i(t) \geq 0$ si $t < \tau$, por lo que $\dot{x}_i(t) \geq -x_i(t)q_i(x(t))$ para todo $t \in [0, \tau)$. Además, por un argumento de compacidad sabemos que existe $C > 0$ tal que $q_i(x(t)) \leq C$ para todo $t \in [0, \tau]$ y tenemos

$$\frac{\dot{x}_i(t)}{x_i(t)} \geq -C.$$

Lo que implica que

$$\int_0^\tau \frac{\dot{x}_i(t)}{x_i(t)} dt \geq -C\tau$$

$$\log(x_i(\tau^-)) - \log(x_i(0)) \geq -C\tau$$

como $x_i(\tau^-) = 0$, del lado izquierdo tenemos $-\infty$, claramente llegando a un absurdo. \square

La invariancia positiva del ortante no negativo es inmediata del resultado anterior (ver, por ejemplo [3, 16]).

Además de la dinámica dada por la ley de acción de masas, las condiciones del sistema están limitadas por lo que se denominan las leyes de conservación.

Definición 1.1.3 (Subespacio estequiométrico. Matriz de conservación). Sea G una red de reacciones químicas, con matriz de estequiometría $N \in \mathbb{Z}^{s \times m}$. Sea $d = s - \text{rango}(N)$. Definimos \mathcal{S} , el **subespacio estequiométrico** de G , como el subespacio generado por las columnas de N . Diremos que $W \in \mathbb{R}^{d \times s}$ es una **matriz de conservación de G** si vale lo siguiente:

- W está triangulada.
- El subespacio generado por las filas de W es el subespacio ortogonal de \mathcal{S} .

Si existe una elección de W de modo que sus coeficientes sean no negativos y que cada columna tenga una entrada positiva (es decir, que cada especie aparezca en al menos una ley de conservación), diremos que G es **conservativa**. (O, equivalentemente, si existe una elección de W donde la primera fila tenga todas sus entradas positivas.)

Volviendo al Ejemplo 1.1.1, notemos que $\text{rango}(N) = 2$, por lo que tendremos dos leyes de conservación independientes. Una matriz posible es:

$$W = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

La idea es que $Wx(t)$ se mantendrá constante para todo tiempo t donde x esté definida. Reinterpretando, esto nos diría que $x_1 + x_3 = [E] + [ES]$ y $x_2 + x_3 + x_4 = [S] + [ES] + [P]$ se mantienen constantes (solo dependen de las condiciones iniciales).

Recordamos que dada una función $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ la matriz Jacobiana es la matriz que posee las derivadas parciales de f , es decir, si $f = (f_1, \dots, f_m)$, entonces $\text{Jac}(f)_{ij} = \partial_j f_i$.

Definición 1.1.4 (Clase de compatibilidad estequiométrica - Sistema ampliado). Sea $x_0 \in \mathbb{R}_{\geq 0}^s$. Sea $c = Wx_0$. Definimos:

$$\mathcal{S}_c = \{x \in \mathbb{R}_{\geq 0}^s : Wx = c\} \quad (1.2)$$

como la **clase de compatibilidad estequiométrica** de x_0 en G .

Sea $I = \{i_1 < i_2 < \dots < i_d\}$ el conjunto de los primeros índices no nulos de cada fila de W . Definimos la **función ampliada** $f_{c,\kappa} : \mathbb{R}_{\geq 0}^s \rightarrow \mathbb{R}^s$ de la siguiente manera

$$f_{c,\kappa}(x)_i = \begin{cases} f_i(x) & \text{si } i \notin I \\ (Wx - c)_k & \text{si } i = i_k \end{cases}. \quad (1.3)$$

Cada fila del sistema $Wx = c$ es luego de la forma $\sum_{j=1}^{\ell} w_{ik_j} x_{k_j} = c_i$. A cada una de estas ecuaciones se la denomina **ley de conservación** asociada al sistema.

Los sistemas conservativos son aquellos en los que se preservan las proteínas a lo largo del proceso. Es decir, si bien estas se pueden “unir” o fosforilar, la cantidad total de proteínas es la misma. En particular, las concentraciones totales de las mismas son constantes. Y por ende, las órbitas del sistema son acotadas, ya que cada coordenada lo está. En efecto,

Proposición 1.1.2. *Sea G una red conservativa. Entonces las clases de compatibilidad \mathcal{S}_c son compactos. En particular, como las soluciones viven dentro de alguna clase de compatibilidad estequiométrica, estas deben ser acotadas.*

Demostración. Son cerradas por ser intersección de cerrados: $\mathbb{R}_{\geq 0}^s$ y $\{Wx = c\}$. Para ver que cada \mathcal{S}_c es acotada, recordamos que cada variable x_i aparece en alguna ley de conservación $\phi(x) = c$. Como los coeficientes de ϕ son positivos, tenemos que cada coordenada luego debe ser acotada por lo que el conjunto también lo es. \square

Retomando el ejemplo 1.1.1, dados $c_1, c_2 > 0$, nuestra función quedaría

$$f_{c,\kappa}(x) = (x_1 + x_3 - c_1, x_2 + x_3 + x_4 - c_2, \kappa_1 x_1 x_2 - (\kappa_2 + \kappa_3) x_3, \kappa_3 x_3)$$

Queda entonces definida la matriz Jacobiana de $f_{c,\kappa}$ con respecto a x como:

$$\text{Jac} f_{c,\kappa}(x)_i = \begin{cases} J_{f_i}(x) & \text{si } i \notin I, \\ W_i & \text{si } i \in I, \end{cases} \quad (1.4)$$

donde W_i es la i -ésima fila de W . Es decir, es la matriz Jacobiana de f con respecto a x , donde la i_k -ésima fila, para $k = 1, \dots, d$, se reemplaza por la k -ésima fila de W .

1.1.1. Estados Estacionarios

Como es natural, será de gran interés estudiar los equilibrios (no negativos) de nuestro sistema dinámico. Estos claramente provienen de la ecuación $\dot{x} = 0$.

Definición 1.1.5 (Estado estacionario - Variedad de estados estacionarios). Un **estado estacionario** del sistema (1.1) es un vector de concentraciones no negativo $x^* \in \mathbb{R}_{\geq 0}^s$ para el cual $f(x^*) = 0$. Llamaremos $V_f = \{x \in \mathbb{R}_{\geq 0}^s : f(x) = 0\}$ a la variedad no negativa de estados estacionarios del sistema.

Veamos primero formas alternativas de referirse a los estados estacionarios dentro de una misma clase de compatibilidad estequiométrica.

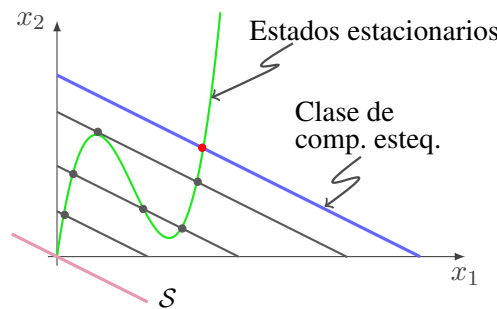
Proposición 1.1.3. Sea G una red de reacciones químicas conservativa, con matriz de reacción $N \in \mathbb{Z}^{s \times m}$. Sean $d = s - \text{rango}(N)$, $W \in \mathbb{R}^{d \times s}$ una matriz de conservación de G , $x_0 \in \mathbb{R}_{\geq 0}^s$ fijo y $c = Wx_0$. Sea la función ampliada $f_{c,\kappa}$ que surge de G y W como en (1.3) y \mathcal{S}_c la clase de compatibilidad estequiométrica como en (1.2). Son equivalentes:

1. $f_{c,\kappa}(x) = 0$
2. $N(\kappa_1 x^{\alpha_1}, \dots, \kappa_m x^{\alpha_m})^T = 0$ y $x \in \mathcal{S}_c$

Demostración. Es evidente que $(2 \Rightarrow 1)$. Para ver la otra implicación, que $x \in \mathcal{S}_c$ es también inmediato. Veamos entonces que $N(\kappa_1 x^{\alpha_1}, \dots, \kappa_m x^{\alpha_m})^T = 0$. Nos falta ver que para cada $i \in I = \{i_1, \dots, i_d\}$, vale que $f_i(x) = 0$. Primero notemos que $WN = 0$ y por lo tanto $Wf(x) = 0$. Además, por construcción, la i -ésima fila de W es de la forma $(0, 0, \dots, 0, w_i, w_{i+1}, \dots, w_s)$ con $w_i \neq 0$ para cada $i \in I$. De esta forma obtenemos $\sum_{j=i}^s w_j f_j(x) = 0$. Mirando la última de estas filas, tenemos que $f_{i_d}(x) = \frac{1}{w_{i_d}} (\sum_{j=i_d+1}^s w_j f_j(x))$, pero como $f_{c,\kappa}(x) = 0$, todas estos sumandos son 0 por lo que f_{i_d} da 0. Repitiendo este argumento, agregando que además $f_{i_d} = 0$, obtenemos que $f_{i_{d-1}} = 0$ y así sucesivamente. \square

Notemos que el sistema $f_{c,\kappa}(x) = 0$ es un sistema de s ecuaciones en s incógnitas. Sería esperable que tenga finitas soluciones. ¿Pero cuántas? ¿Una, más de una? Para esto definimos el concepto de multiestacionariedad.

Definición 1.1.6 (Multiestacionariedad). Diremos que una red G es **multiestacionaria** si existen constantes de reacción κ y una clase de compatibilidad estequiométrica \mathcal{S}_c de modo que $\#(V_f \cap \mathbb{R}_{>0}^s \cap \mathcal{S}_c) \geq 2$. Es decir, si existe una elección de constantes de reacción tal que en alguna clase de compatibilidad estequiométrica nuestro sistema tiene al menos dos estados estacionarios positivos.



Clasificaremos también a los puntos de equilibrio mediante su estabilidad como es usual.

Definición 1.1.7. Sea G una red de reacciones químicas con V_f la variedad no negativa de estados estacionarios de la misma y $x^* \in V_f$.

- Si $x^* \in \mathbb{R}_{>0}^s$ (es decir, $x_i^* > 0$ para $1 \leq i \leq s$) diremos que x^* es un estado estacionario **positivo**. Caso contrario, diremos que es un estado estacionario **en el borde**.
- Si $\text{Im}(\text{Jac}f(x^*)|_S) = S$, diremos que x^* es **no degenerado**.
- Si x^* es no degenerado y los autovalores no nulos de $\text{Jac}f(x^*)$ tienen parte real negativa, diremos que es **exponencialmente estable**.
- Diremos que G es **biestable** si existe $\kappa \in \mathbb{R}_{>0}^m$ para el cual existe una clase S_c de compatibilidad estequiométrica donde $f_{c,\kappa}$ tiene dos o más estados estacionarios positivos exponencialmente estables.

Observemos que biestabilidad implica multiestacionariedad. Veremos más adelante un criterio para detectar multiestacionariedad, pero en general es difícil determinar la biestabilidad.

Definición 1.1.8 (Parametrización de estados estacionarios - Función crítica). Dada G una red de reacciones químicas con s especies, m reacciones y W matriz de conservación. Una **parametrización de estados estacionarios** consiste en una función $\phi : \mathbb{R}_{>0}^{m'} \times \mathbb{R}_{>0}^{s'} \rightarrow \mathbb{R}_{>0}^m \times \mathbb{R}_{>0}^s$, donde $m' \leq m$ y $s' \leq s$, que verifica:

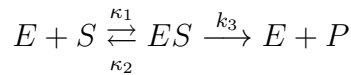
- $\phi(\hat{\kappa}, \hat{x})$ extiende a $(\hat{\kappa}, \hat{x})$ en las coordenadas correspondientes. Más precisamente, la proyección natural $\pi : \mathbb{R}^m \times \mathbb{R}^s \rightarrow \mathbb{R}^{m'} \times \mathbb{R}^{s'}$ verifica que $\pi \circ \phi = id_{\mathbb{R}_{>0}^{m'} \times \mathbb{R}_{>0}^{s'}}$.
- La imagen de ϕ está dada por

$$\text{Im}(\phi) = \{(\kappa^*, x^*) \in \mathbb{R}_{>0}^m \times \mathbb{R}_{>0}^s : x^* \text{ es un estado estacionario del sistema definido por } G \text{ y } \kappa = \kappa^*\}$$

Dada una parametrización de estados estacionarios ϕ , definimos la **función crítica** $C : \mathbb{R}^{m'} \times \mathbb{R}^{s'} \rightarrow \mathbb{R}$ vía:

$$C(\hat{\kappa}, \hat{x}) = (\det(\text{Jac}f_{c,\kappa}))|_{(\kappa,x)=\phi(\hat{\kappa},\hat{x})}. \quad (1.5)$$

Apliquemos estas ideas al Ejemplo 1.1.1. Supongamos que $c_1, c_2 > 0$ (lo que es razonable por la no negatividad de las concentraciones). Igualando $f_{c,\kappa}(x) = 0$ se puede ver que podemos despejar $x_1 = -x_3 + c_1$ y $x_4 = -x_3 - x_2 + c_2$. Y de la última ecuación obtenemos que $\kappa_3 x_3 = 0$ así que $x_3 = 0$ y $x_1 = c_1$. Usando la tercera ecuación, tenemos que $x_2 = 0$ y así obtenemos que $(c_1, 0, 0, c_2)$ es el único estado estacionario del sistema en la clase de compatibilidad estequiométrica definida por c_1 y c_2 , independientemente de los valores de $\kappa_1, \kappa_2, \kappa_3$.



Recordando de dónde proviene nuestro sistema, este es un resultado esperable. Eventualmente, la reacción irá de izquierda a derecha, las especies S y ES desaparecerán hasta que solo

queden E y P , en las concentraciones apropiadas para respetar las leyes de conservación, y este comportamiento es independiente del valor de las constantes de reacción.

Estudemos la estabilidad en $(c_1, 0, 0, c_2)$ de nuestra red. En este caso, tenemos que

$$\text{Jac}f(c_1, 0, 0, c_2) = \begin{pmatrix} 0 & -\kappa_1 c_1 & \kappa_2 + \kappa_3 & 0 \\ 0 & -\kappa_1 c_1 & \kappa_2 & 0 \\ 0 & \kappa_1 c_1 & -\kappa_2 - \kappa_3 & 0 \\ 0 & 0 & \kappa_3 & 0 \end{pmatrix}$$

Recordemos que $\mathcal{S} = \langle (-1, -1, 1, 0), (1, 1, -1, 0), (1, 0, -1, 1) \rangle$, y es sencillo ver que $\text{Im}(\text{Jac}f) \subseteq \mathcal{S}$ y que $\text{Jac}(f)|_{\mathcal{S}}$ es monomorfismo por lo que nuestro estado es no degenerado. Para estudiar la estabilidad, miramos los autovalores no nulos de la matriz. Observar que

$$p(\lambda) = \det(\lambda I - \text{Jac}f(c_1, 0, 0, c_2)) = \lambda^2(\lambda^2 + (\kappa_1 c_1 + k_2 + \kappa_3)\lambda + k_1 k_3 c_1)$$

Nos interesa la parte real de las raíces de $q(\lambda) = \lambda^2 + (\kappa_1 c_1 + k_2 + \kappa_3)\lambda + k_1 k_3 c_1 = a\lambda^2 + b\lambda + c$. Primero veremos que son reales.

$$\begin{aligned} b^2 - 4ac &= \kappa_1^2 c_1^2 + \kappa_2^2 + \kappa_3^2 + 2\kappa_1 \kappa_2 c_1 + 2\kappa_1 \kappa_3 c_1 + 2\kappa_2 \kappa_3 - 4\kappa_1 \kappa_3 c_1 \\ &= \kappa_1^2 c_1^2 + \kappa_2^2 + \kappa_3^2 + 2\kappa_1 \kappa_2 c_1 - 2\kappa_1 \kappa_3 c_1 + 2\kappa_2 \kappa_3 \\ &= (\kappa_1 c_1 - \kappa_3)^2 + 2\kappa_1 \kappa_2 c_1 + 2\kappa_2 \kappa_3 > 0 \end{aligned}$$

Por ende las raíces son reales. Por otro lado, observar que $a > 0$, $c > 0$ y que $-b/2a < 0$, por lo que necesariamente ambas raíces deben ser negativas, por lo que el equilibrio es siempre estable (lo cual es intuitivo si uno recuerda la estructura de la red).

Observar que si bien pudimos verificar la estabilidad para todo c_1, c_2 , incluso en una red sencilla, no fue obvio. Esto se reflejará más en general cuando veamos que la biestabilidad es más difícil de verificar como otras propiedades como la multiestacionariedad, entre otros motivos por la elevada dificultad computacional. Aun así, en la Sección 2.2 introduciremos un criterio útil.

1.2. La red de la proteína ERK

En esta sección presentaremos la red en la que nos enfocaremos en el resto de la tesis: Quinasa Regulada por Señal Extracelular (ERK por sus siglas en inglés).

1.2.1. La cascada de señalización de ERK

A pesar de que las células tienen una membrana, son capaces de reaccionar a estímulos exteriores (“señales”). Esto se logra mediante receptores que se encuentran en la membrana y

que desencadenan una serie de reacciones mediante una compleja red de moléculas que les permiten realizar acciones de crecimiento y regulación, entre otras. Para una explicación detallada de la señalización celular, ver [1, Cap.15].

El objetivo de este trabajo es estudiar un modelo matemático de la red ERK. Esta red cumple un rol importante en el ciclo celular. Dicho de manera muy simplificada, luego de la división celular (mitosis), la célula recibe una señal exterior que a través de una cadena de reacciones llega de la membrana celular al núcleo y luego se procede a producir una proteína que es la que hace que la célula crezca.

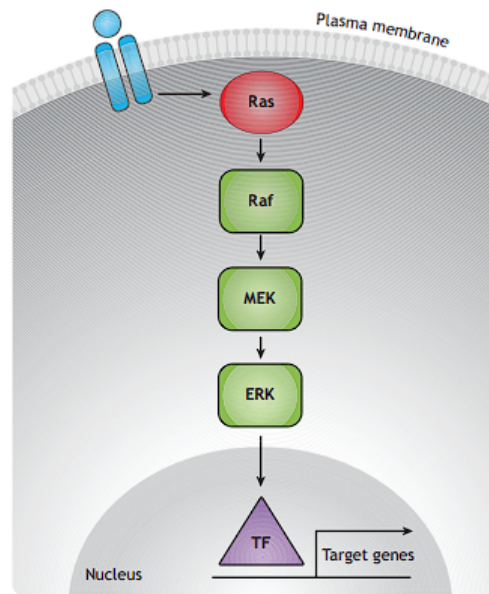


Figura 1.1: **Versión simplificada del camino de la señal desde la membrana celular hasta el núcleo.** Los factores de crecimiento (representados por el círculo celeste) se unen a los receptores transmembrana, quienes pasan la señal a la proteína Ras. Luego esta activa la principal cascada de fosforilación desde la quinasa Raf a la quinasa MEK, y esta última a la quinasa regulada por señal extracelular ERK. La quinasa ERK luego se transloca al núcleo, donde interactúa con factores de transcripción para regular la expresión de genes objetivo [21].

La última de estas reacciones antes de llegar al núcleo es la que estudiaremos con más detalle. En esta participan tres enzimas: la enzima ERK (tiene el mismo nombre que la red), y las enzimas MEK y MPK3 que tienen la función de fosforilar y desfosforilar a ERK, respectivamente (fosforilar se refiere a agregar un grupo fosfato a la enzima). La fosforilación produce un cambio en la distribución de los niveles de energía de la enzima, cambiando sus propiedades biológicas [12].

Trabajaremos con un modelo de doble-fosforilación, es decir, que a nuestra enzima ERK se le pueden colocar a lo sumo dos grupos fosfatos. Entonces denotamos S_{00} a ERK si no posee ningún grupo fosfato adicionado, S_{01} o S_{10} cuando tiene exactamente uno (consideramos que cada índice es un sitio donde se puede colocar el grupo fosfato) y S_{11} si la proteína está completamente fosforilada. Cuando ambas fosforilaciones pueden darse sucesivas sin que la enzima se separe del sustrato, se dice que la fosforilación es *procesativa*. Si, en cambio, la enzima debe

separarse del sustrato para que se dé la segunda fosforilación se dice que la fosforilación es *distributiva*.

Luego, si denotamos a la enzima MEK como E y a la enzima MPK3 como F tenemos la red de reacciones químicas que se puede ver en la Figura 1.2.

1.2.2. Modelado de ERK

Detallaremos aquí dos modelos de la red de ERK, como se presentan en [5]. El primero de ellos lo representamos en la Figura 1.2. Lo que interesa estudiar en esta red es la pregunta planteada por primera vez en [22]:

Pregunta 1.2.1. *¿Es verdad que para todos los valores $p_k := k_{cat}/(k_{cat} + k_{off})$ y $p_\ell = \ell_{cat}/(\ell_{cat} + \ell_{off})$ cercanos a 1 vale que la red de ERK de la Figura 1.2 es biestable y oscilatoria?*

Cuando p_k y p_ℓ se acercan a 1, la red “tiende” a una completamente procesativa que se sabe no tiene biestabilidad ni oscilaciones [6]. Como la biestabilidad y las oscilaciones son fenómenos dinámicos que permiten que la célula realice funciones biológicamente significativas, es interesante saber “hasta qué punto” la red mantiene la capacidad de presentar estas características dinámicas importantes.

En la Sección 3.1 mostramos (a partir de [20]) que la biestabilidad sí se preserva cuando $k_2 = m_1 = \ell_2 = n_2 = 0$ y, o bien $k_{on} = 0$ y $\ell_{on} \neq 0$, o bien $k_{on} \neq 0$ y $\ell_{on} = 0$, o ambas son no nulas. De esta forma, se define la *mínima red ERK biestable* (que está representada en la Figura 1.2 al eliminar las reacciones con constantes pintadas de azul en negrita) y esto da lugar a la siguiente pregunta:

Pregunta 1.2.2. *Para valores de p_k y p_ℓ cercanos a 1, ¿es biestable la mínima red ERK biestable?*

Por resultados que levantan la biestabilidad desde una subred a una red que la contiene (ver Teorema 2.1.4), sabremos entonces que si la mínima red ERK biestable es realmente biestable, entonces la red ERK también lo será.

Establezcamos ahora algunas definiciones que nos serán de utilidad más adelante.

Definición 1.2.1. Para la red ERK definida a partir de la Figura 1.2:

- Diremos que una red es una **subred de ERK** si proviene de quitar algunas reacciones de la misma, o lo que es lo mismo, de fijar alguna constante de reacción igual a 0.
- Decimos que una subred de ERK es **irreversible** si alguna constante $k_2, k_{on}, m_1, \ell_2, \ell_{on}, n_2$ es 0 (en la Figura 1.2 están marcadas con azul). Decimos que es completamente irreversible si todas son 0.
- Definimos los **niveles de procesatividad** como $p_k = \frac{k_{cat}}{k_{cat} + k_{off}}$ y $p_\ell = \frac{\ell_{cat}}{\ell_{cat} + \ell_{off}}$. Cuanto más cercanos a uno, las reacciones verticales k_{off} y ℓ_{off} son más despreciables y por ende “la señal avanza más cómodamente hacia el final de la red”.

Subrayamos en azul los términos correspondientes a reacciones que no están en la mínima red ERK biestable. Al borrarlos obtenemos las ecuaciones de esta red más chica.

Tenemos además tres leyes de conservación, que se corresponden con la concentración total de ERK (S), de la quinasa MEK (E) y de la fosfatasa MPK3 (F):

$$\begin{aligned} x_1 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} &= S_{tot} =: c_1 \\ x_2 + x_7 + x_8 + x_{11} &= E_{tot} =: c_2 \\ x_3 + x_4 + x_5 + x_6 &= F_{tot} =: c_3 \end{aligned} \quad (1.7)$$

Como la matriz de conservación que surge de considerar los coeficientes del sistema lineal (1.7) tiene entradas no negativas y cada columna tiene al menos una entrada positiva, deducimos que la red ERK es conservativa. Además, como la dimensión de \mathcal{S}^\perp es 3, tenemos que $\dim(\mathcal{S}) = 12 - 3 = 9$.

Notemos que al borrar los términos que están subrayados en azul obtenemos las mismas relaciones de conservación y por lo tanto la mínima red ERK biestable (y cualquier subred de ERK irreversible) es conservativa. Más aún, los subespacios estequiométricos de la red ERK y sus subredes irreversibles coinciden.

Daremos entonces una parametrización de estados estacionarios para la ERK completa y sus subredes irreversibles.

Proposición 1.2.1. *Sea G la red ERK completa o una subred irreversible como en la Definición 1.2.1. Dada una constante de reacción, κ , definimos $\chi_\kappa = 1$ si κ está en la red y 0 si no. Luego tenemos sistema de ecuaciones polinomiales para los estados estacionarios dentro de la clase de compatibilidad estequiométrica dado por $h_{c,a} : \mathbb{R}_{>0}^{12} \rightarrow \mathbb{R}^{12}$:*

$$\begin{cases} h_{c,a,1} = x_1 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} - c_1, \\ h_{c,a,2} = x_2 + x_7 + x_8 + x_{11} - c_2, \\ h_{c,a,3} = x_3 + x_4 + x_5 + x_6 - c_3, \\ h_{c,a,4} = a_{12}x_3x_{12} - x_4, \\ h_{c,a,5} = a_3x_4 - x_5 - a_2x_8, \\ h_{c,a,6} = a_{13}x_3x_9 - x_6, \\ h_{c,a,7} = a_5x_{11} - a_4x_6 - x_7, \\ h_{c,a,8} = a_{11}x_2x_{10} - x_8, \\ h_{c,a,9} = a_9x_7 - \chi_{k_{on}}a_8x_2x_9 - x_6, \\ h_{c,a,10} = a_7x_5 - \chi_{\ell_{on}}a_6x_3x_{10} - x_8, \\ h_{c,a,11} = a_{10}x_1x_2 - x_{11}, \\ h_{c,a,12} = x_7 - a_1x_5. \end{cases} \quad (1.8)$$

Más aún, si $\hat{a} = (a_2, a_4, a_6, a_8)$ si k_{on} y ℓ_{on} ambas están, $\hat{a} = (a_2, a_4, a_6)$ si está k_{on} pero no ℓ_{on} y $\hat{a} = (a_2, a_4, a_8)$ si ℓ_{on} está y k_{on} no, y $\hat{a} = (a_2, a_4)$ si ninguna está. Luego tenemos la siguiente parametrización de estados estacionarios positivos:

$$\phi(\hat{a}, x_1, \dots, x_{12}) = (a_1, \dots, a_{13}, x_1, \dots, x_{12}) \quad (1.9)$$

con

$$\begin{aligned} a_1 &:= \frac{x_7}{x_5} & a_3 &:= \frac{a_2 x_8 + x_5}{x_4} & a_5 &:= \frac{a_4 x_6 + x_7}{x_{11}} \\ a_7 &:= \frac{\chi_{\ell_{on}} a_6 x_3 x_{10} + x_8}{x_5} & a_{11} &:= \frac{x_8}{x_2 x_{10}} & a_{10} &:= \frac{x_{11}}{x_1 x_2} \\ a_9 &:= \frac{\chi_{k_{on}} a_8 x_2 x_9 + x_6}{x_7} & a_{12} &:= \frac{x_4}{x_3 x_{12}} & a_{13} &:= \frac{x_6}{x_3 x_9} \end{aligned}$$

Los polinomios $h_{c,a}$ que no se obtienen de las relaciones de conservación del sistema se consiguen operando linealmente con los polinomios f_1, \dots, f_{12} , con $f(x) = \dot{x}$. La parametrización se obtiene luego despejando.

Para la mínima red ERK biestable (es decir, cuando $k_2 = m_1 = \ell_2 = \ell_{on} = n_2 = 0$) daremos otra parametrización de estados estacionarios que también usaremos. Efectivamente, si denotamos por $\kappa := (k_1, k_3, k_{cat}, k_{on}, k_{off}, \ell_1, \ell_3, \ell_{cat}, \ell_{off}, m_2, m_3, n_1, n_3)$, tenemos:

$$\begin{aligned} \phi : \mathbb{R}_{>0}^{13} \times \mathbb{R}_{>0}^3 &\rightarrow \mathbb{R}_{>0}^{13} \times \mathbb{R}_{>0}^{12} \\ (\kappa; x_1, x_2, x_3) &\mapsto (\kappa; x_1, x_2, \dots, x_{12}), \end{aligned} \quad (1.10)$$

donde

$$\begin{aligned} x_4 &= \frac{k_1 k_{cat} (\ell_{cat} + \ell_{off}) (k_{on} x_2 + n_1 x_3) x_1 x_2}{\ell_3 \ell_{cat} (k_{cat} k_{on} x_2 + k_{cat} n_1 x_3 + k_{off} n_1 x_3)}, & x_5 &= \frac{k_1 k_{cat} (k_{on} x_2 + n_1 x_3) x_1 x_2}{\ell_{cat} (k_{cat} k_{on} x_2 + k_{cat} n_1 x_3 + k_{off} n_1 x_3)}, \\ x_6 &= \frac{n_1 k_1 k_{off} x_1 x_2 x_3}{n_3 (k_{cat} k_{on} x_2 + k_{cat} n_1 x_3 + k_{off} n_1 x_3)}, & x_7 &= \frac{k_1 (k_{on} x_2 + n_1 x_3) x_1 x_2}{k_{cat} k_{on} x_2 + k_{cat} n_1 x_3 + k_{off} n_1 x_3}, \\ x_8 &= \frac{k_1 k_{cat} \ell_{off} (k_{on} x_2 + n_1 x_3) x_1 x_2}{\ell_{cat} m_3 (k_{cat} k_{on} x_2 + k_{cat} n_1 x_3 + k_{off} n_1 x_3)}, & x_9 &= \frac{k_1 k_{off} x_1 x_2}{k_{cat} k_{on} x_2 + k_{cat} n_1 x_3 + k_{off} n_1 x_3}, \\ x_{10} &= \frac{\ell_{cat} m_2 (k_{cat} k_{on} x_2 + k_{cat} n_1 x_3 + k_{off} n_1 x_3)}{k_1 k_{cat} \ell_{off} (k_{on} x_2 + n_1 x_3) x_1}, & x_{11} &= \frac{k_1 x_2 x_2}{k_3}, \\ x_{12} &= \frac{k_1 k_{cat} (\ell_{cat} + \ell_{off}) (k_{on} x_2 + n_1 x_3) x_1 x_2}{\ell_{cat} \ell_1 (k_{cat} k_{on} x_2 + k_{cat} n_1 x_3 + k_{off} n_1 x_3) x_3}. \end{aligned}$$

Será importante para estudiar multiestacionariedad y biestabilidad (Sección 3.1) verificar que nuestras redes no poseen estados estacionarios en el borde $\partial \mathbb{R}_{\geq 0}^n$.

Proposición 1.2.2. *La red ERK completa, las subredes irreversibles y la red reducida no poseen estados estacionarios con alguna coordenada nula, es decir, si $f_{c,\kappa}(x) = 0$, entonces $x \in \mathbb{R}_{>0}^n$.*

Demostración. Lo probaremos para la red completa, el resto son argumentos similares. Re-

agrupamos las ecuaciones de la siguiente manera:

$$\left\{ \begin{array}{l} f_1 = -k_1 x_1 x_2 + (\underline{k_2 x_{11}} + \ell_{cat} x_5 + n_3 x_6) \\ f_2 = -(k_1 x_1 + k_{on} x_9 + m_2 x_{10}) x_2 + (\underline{k_2 x_{11}} + k_{cat} x_7 + k_{off} x_7 + \underline{m_1 x_8} + m_3 x_8) \\ f_3 = -(\ell_1 x_{12} + \underline{\ell_{on} x_{10}} + n_1 x_9) x_3 + (\underline{\ell_2 x_4} + \ell_{cat} x_5 + \ell_{off} x_5 + \underline{n_2 x_6} + n_3 x_6) \\ f_4 = \ell_1 x_3 x_{12} - (\underline{\ell_2} + \ell_3) x_4 \\ f_5 = \underline{\ell_{on} x_3 x_{10}} + \ell_3 x_4 - (\ell_{cat} + \ell_{off}) x_5 \\ f_6 = n_1 x_3 x_9 - (\underline{n_2} + n_3) x_6 \\ f_7 = k_{on} x_2 x_9 + k_3 x_{11} - (k_{cat} + k_{off}) x_7 \\ f_8 = m_2 x_2 x_{10} - (\underline{m_1} + m_3) x_8 \\ f_9 = -(k_{on} x_2 + n_1 x_3) x_9 + (k_{off} x_7 + \underline{n_2 x_6}) \\ f_{10} = -(\underline{\ell_{on} x_3} + m_2 x_2) x_{10} + (\ell_{off} x_5 + \underline{m_1 x_8}) \\ f_{11} = k_1 x_1 x_2 - (\underline{k_2} + k_3) x_{11} \\ f_{12} = -\ell_1 x_3 x_{12} + (k_{cat} x_7 + \underline{\ell_2 x_4} + m_3 x_8) \end{array} \right.$$

Y recordemos las leyes de conservación en (1.7).

- Si $x_2 = 0$, por f_2 , como todas las variables son no negativas, y $\dot{x}_2 = f_2(x) = 0$ por hipótesis, tenemos que x_{11}, x_7, x_8 son 0. Pero recordemos que $c_1 > 0$, por lo que E_{tot} no se cumple y tenemos un absurdo.
- Similarmente, si $x_3 = 0$. por f_3 tenemos que x_4, x_5, x_6 son 0, pero entonces no se respeta la ley asociada a F_{tot}
- Si $x_1 = 0$, por f_1 , x_{11}, x_5, x_6 son 0. Pero entonces por f_5 , x_4 y $x_{10} = 0$ (ya sabemos que $x_3 \neq 0$). Por f_4 , $x_{12} = 0$, y por f_6 , $x_9 = 0$. Así contradicimos la ley asociada a S_{tot} .
- Luego podemos suponer que x_1, x_2, x_3 son no nulos. Por f_{11} , x_{11} debe ser no nula (ya que debe compensar el signo). Por f_7, f_9 y f_{12} podemos deducir que $x_7 \neq 0$, por lo que $x_9 \neq 0$ y $x_{12} \neq 0$. Usando f_6, f_4 podemos deducir que x_4, x_6 son no nulos respectivamente. Usando f_5 , deducimos que x_5 es no nulo, usando f_{10} que x_{10} es no nulo y usando f_8 que x_8 es no nulo.

□

Presentamos ahora otra red ERK más chica, a la que llamaremos la *red reducida de ERK* [20] (ver Figura 1.3).

Para esta red planteamos la siguiente pregunta:

Pregunta 1.2.3. Para valores de p_k y p_ℓ cercanos a 1, ¿es oscilatoria la red reducida de ERK?

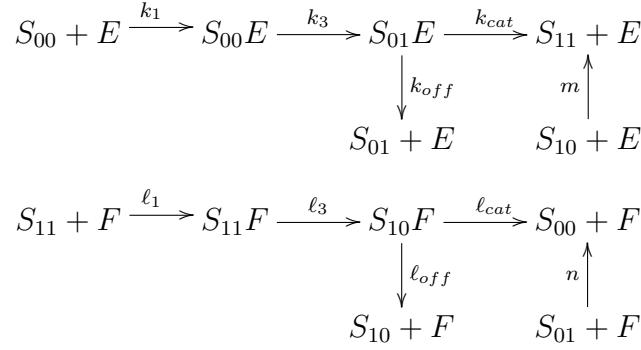


Figura 1.3: La red reducida de ERK. [20]

Las ecuaciones de esta red son

$$\begin{cases}
\dot{x}_1 = -k_1 x_1 x_2 + n x_6 x_8 + \ell_{cat} x_{10} \\
\dot{x}_2 = -k_1 x_1 x_2 + k_{cat} x_4 + k_{off} x_4 \\
\dot{x}_3 = k_1 x_1 x_2 - k_3 x_3 \\
\dot{x}_4 = k_3 x_3 - k_{cat} x_4 - k_{off} x_4 \\
\dot{x}_5 = m x_2 x_7 - \ell_1 x_5 x_8 + k_{cat} x_4 \\
\dot{x}_6 = -n x_6 x_8 + k_{off} x_4 \\
\dot{x}_7 = -m x_2 x_7 + \ell_{off} x_{10} \\
\dot{x}_8 = -\ell_1 x_5 x_8 + \ell_{off} x_{10} + \ell_{cat} x_{10} \\
\dot{x}_9 = \ell_1 x_5 x_8 - \ell_3 x_9 \\
\dot{x}_{10} = -\ell_{off} x_{10} + \ell_3 x_9 - \ell_{cat} x_{10}
\end{cases} \quad (1.11)$$

Tiene además tres leyes de conservación:

$$\begin{aligned}
x_1 + x_3 + x_4 + x_5 + x_6 + x_7 + x_9 + x_{10} &= S_{tot} =: c_1 \\
x_2 + x_3 + x_4 &= E_{tot} =: c_2 \\
x_8 + x_9 + x_{10} &= F_{tot} =: c_3
\end{aligned} \quad (1.12)$$

Análogamente, tenemos una parametrización para la red reducida.

Proposición 1.2.3. *Para la red ERK reducida (ver Figura 1.3), se tiene un sistema de ecuaciones polinomiales para los estados estacionarios dentro de la clase de compatibilidad estequiométrica dado por $h_{c,a} : \mathbb{R}_{>0}^{10} \rightarrow \mathbb{R}^{10}$:*

$$\begin{aligned}
h_{c,a,1} &= x_1 + x_3 + x_4 + x_5 + x_6 + x_7 + x_9 + x_{10} - c_1, \\
h_{c,a,2} &= x_2 + x_3 + x_4 - c_2, \\
h_{c,a,3} &= -(k_{cat} + k_{off})\ell_{cat}x_{10} + k_1k_{cat}x_1x_2, \\
h_{c,a,4} &= k_3x_3 - (k_{cat} + k_{off})x_4, \\
h_{c,a,5} &= \ell_{off}x_{10} - m_2x_2x_7 \\
h_{c,a,6} &= \ell_1x_5x_8 - (\ell_{cat} + \ell_{off})x_{10} \\
h_{c,a,7} &= \ell_3x_9 - (\ell_{cat} + \ell_{off})x_{10} \\
h_{c,a,8} &= x_8 + x_9 + x_{10} - c_3 \\
h_{c,a,9} &= k_{cat}x_4 - \ell_{cat}x_{10}, \\
h_{c,a,10} &= k_{off}\ell_{cat}x_{10} - k_{cat}nx_6x_8.
\end{aligned}$$

Despejando de $h_{c,a} = 0$ se obtiene la siguiente parametrización positiva de los estados estacionarios:

$$\begin{aligned}
\phi : \mathbb{R}_{>0}^{3+10} &\rightarrow \mathbb{R}_{>0}^{10+10} \\
(k_{cat}, k_{off}, \ell_{off}, x_1, x_2, \dots, x_{10}) &\mapsto (\kappa_1, \kappa_3, k_{cat}, k_{off}, m, \ell_1, \ell_3, \lambda_{cat}, \ell_{off}, n, x_1, x_2, \dots, x_{10}),
\end{aligned} \tag{1.13}$$

dada por

$$\begin{aligned}
k_1 &:= \frac{(k_{cat} + k_{off})x_4}{x_1x_2} & k_3 &:= \frac{(k_{cat} + k_{off})x_4}{x_3} & m &:= \frac{\ell_{off}x_{10}}{x_2x_7} & \ell_1 &:= \frac{\ell_{off}x_{10} + k_{cat}x_4}{x_5x_8} \\
\ell_3 &:= \frac{\ell_{off}x_{10} + k_{cat}x_4}{x_9} & \ell_{cat} &:= \frac{k_{cat}x_4}{x_{10}} & n &:= \frac{k_{off}x_4}{x_6x_8}.
\end{aligned}$$

En particular, la imagen de ϕ es el siguiente conjunto de pares de estados estacionarios positivos y constantes de reacción:

$$\{(k^*, x^*) \in \mathbb{R}_{>0}^{10+10} \mid x^* \text{ es un estado estacionario de (1.11) cuando } k = k^*\}.$$

Donde k denota el vector $(k_1, k_3, k_{cat}, k_{off}, m, \ell_1, \ell_3, \ell_{cat}, \ell_{off}, n)$.

Los polinomios $h_{c,a}$ que no se obtienen de las relaciones de conservación del sistema se consiguen operando linealmente con los polinomios f_1, \dots, f_{10} , con $f(x) = \dot{x}$.

Desarrollaremos en el Capítulo 3 los resultados que han surgido al intentar responder estas preguntas. Primero presentaremos en el Capítulo 2 la teoría necesaria para abordar los problemas.

Capítulo 2

Herramientas Teóricas

El objetivo de este capítulo es presentar distintas herramientas teóricas que nos serán de utilidad a la hora de estudiar la dinámica de la red ERK.

2.1. Herramientas para estudiar estabilidad

En esta sección presentaremos dos herramientas que se utilizan para estudiar los estados estacionarios de las redes de reacciones químicas. La primera herramienta la introducimos en el Teorema 2.1.1 (adaptada de [5, Proposición 2.2]) y determina condiciones suficientes y necesarias para que un sistema sea multiestacionario. Su demostración se basa en Teoría de Grado. Y la segunda herramienta, que presentamos en el Teorema 2.2.4 (ver [19]) es una generalización del criterio de Routh-Hurwitz, que se utiliza para detectar bifurcaciones de Hopf simples.

2.1.1. Multiestacionariedad y Funciones Críticas

El siguiente teorema nos permite determinar si el sistema de reacciones químicas dado por una red G conservativa y sin estados estacionarios en el borde es multiestacionario o no, a través del análisis del signo de la función crítica definida en (1.5). Más aún, en caso de confirmar multiestacionariedad, nos permite construir un *testigo* de dicha condición.

Teorema 2.1.1. *Sea G una red de reacciones químicas conservativa con matriz de estequiometría N y matriz de conservación W . Sean ϕ una parametrización de estados estacionarios y C la función crítica como en la Definición 1.1.8. Si todos los estados estacionarios de G son positivos (es decir, no tiene estados estacionarios con alguna coordenada nula) entonces:*

1. *G es multiestacionaria si y solo si existe $(\hat{\kappa}^*, \hat{x}^*)$ de modo que $\text{signo}(C(\hat{\kappa}^*, \hat{x}^*)) = (-1)^{\text{rango}(N)+1}$.*
2. *Si $(\hat{\kappa}^*, \hat{x}^*)$ verifica lo anterior y $(\kappa^*, x^*) = \phi(\hat{\kappa}^*, \hat{x}^*)$, entonces si $c^* = Wx^*$, el sistema ampliado vía f_{c^*, κ^*} tiene al menos dos estados estacionarios positivos. Es decir, a partir de un punto que verifica la condición del ítem anterior, se pueden obtener constantes de*

reacción y una clase de compatibilidad estequiométrica para las cuales verificamos la multiestacionariedad.

La demostración de este teorema está basada en el Teorema 4.1 de la Información Complementaria de [4]. Probaremos aquí una versión de ese teorema, adaptada a nuestro contexto.

Teorema 2.1.2. *Sea G una red de reacciones químicas y $\dot{x} = f_{\kappa}(x)$ el sistema de cinética de acción de masas resultante. Asumimos que G es conservativa y sin estados estacionarios en el borde del ortante positivo $\mathbb{R}_{>0}^s$. Sea N la matriz de estequiometría de la red y \mathcal{S}_c una clase de compatibilidad estequiométrica tal que $\mathcal{S}_c \cap \mathbb{R}_{>0}^s \neq \emptyset$, y consideramos $\text{Jac}f_{c,\kappa}$ como en (1.4). Entonces vale lo siguiente:*

- (A) **Unicidad de equilibrio:** *Si $\text{signo}(\det(\text{Jac}f_{c,\kappa}(x))) = (-1)^{\text{rango}(N)}$ para todo estado estacionario $x \in \mathcal{S}_c \cap \mathbb{R}_{>0}^s$, entonces existe exactamente un estado estacionario positivo en \mathcal{S}_c . Más aún, este equilibrio es no degenerado.*
- (B) **Múltiples equilibrios:** *Si $\text{signo}(\det(\text{Jac}f_{c,\kappa}(x))) = (-1)^{\text{rango}(N)+1}$ para algún estado estacionario $x \in \mathcal{S}_c \cap \mathbb{R}_{>0}^s$, entonces hay al menos dos estados estacionarios positivos en \mathcal{S}_c , al menos uno de ellos es no degenerado. Si todos los posibles estados estacionarios en \mathcal{S}_c son no degenerados, entonces hay al menos tres y siempre una cantidad impar.*

Veamos cómo, a partir de este resultado, podemos deducir el Teorema 2.1.1. Consideremos W , la matriz de conservación de G y para $(\hat{\kappa}^*, \hat{x}^*)$ como en el ítem 1 del Teorema 2.1.1 llamemos $(\kappa^*, x^*) = \phi(\hat{\kappa}^*, \hat{x}^*)$, $c^* = Wx^*$ y f_{c^*,κ^*} como en (1.3). Como ϕ es una parametrización de estados estacionarios, tenemos que x^* es estado estacionario para el sistema definido por G para las constantes de reacción κ^* , y entonces los ítems (A) y (B) del Teorema 2.1.2 nos dan las dos implicaciones para multiestacionariedad. El ítem 2 del Teorema 2.1.1 es inmediato de lo anterior.

Veamos ahora la prueba del Teorema 2.1.2. Para esto necesitamos introducir algunos conceptos y resultados de Teoría de Grado [4, 23] adaptados a nuestro contexto.

Sean $B \subset \mathbb{R}^s$ un conjunto abierto y acotado, y $f \in \mathcal{C}^1(\overline{B}, \mathbb{R}^s)$ una función continua y derivable, con derivada continua, de la clausura de B , \overline{B} , en \mathbb{R}^s . Notamos con ∂B a $\overline{B} \setminus B$, el borde de B . Diremos que $y \in \mathbb{R}^s \setminus f(\partial B)$ es un *valor regular* de f si $\det(\text{Jac}f(x)) \neq 0$ para todo $x \in B$ tal que $f(x) = y$.

Lema 2.1.1. *Si $B \subset \mathbb{R}^s$ es abierto y acotado, $f \in \mathcal{C}^1(\overline{B}, \mathbb{R}^s)$ e $y \in \mathbb{R}^s \setminus f(\partial B)$ es un valor regular de f , entonces el conjunto $\{x \in B \mid f(x) = y\}$ es finito.*

Demostración. Supongamos que no fuera así, es decir, $f^{-1}(y)$ es infinito. Como $B \subseteq \mathbb{R}^s$ es acotado, sabemos que $f^{-1}(y)$ tiene punto de acumulación en \overline{B} . Sea luego $\{x_n\} \subseteq f^{-1}(y)$ sucesión de elementos distintos que converge a $x \in \overline{B}$.

En primer lugar, $x \notin \partial B$. En efecto, de ser así, por continuidad de f tendríamos que $f(x) = \lim f(x_n) = y$, pero $y \notin f(\partial B)$. Ahora, como y es un valor regular, tenemos que $\det(\text{Jac}f(x)) \neq 0$. Por el Teorema de la Función Inversa, debe existir un entorno $U \subseteq B$ de x para la cual $f|_U : U \rightarrow \mathbb{R}^s$ es inyectiva. Pero esto es absurdo ya que $\lim x_n = x$ y $f(x_n) = y$ para todo $n \in \mathbb{N}$. \square

Si $B \subset \mathbb{R}^s$ es abierto y acotado, $f \in C^1(\overline{B}, \mathbb{R}^s)$ e $y \in \mathbb{R}^s \setminus f(\partial B)$ es un valor regular de f , definimos su grado de Brouwer como

$$\deg(f, B, y) := \sum_{\{x \in B \mid f(x) = y\}} \text{signo}(\det(\text{Jac}f(x))). \quad (2.1)$$

Observación 2.1.1. Si $\deg(f, B, y) = \pm 1$, entonces la ecuación $f(x) = y$ tiene al menos una solución $x \in B$ y la cantidad de soluciones en B es impar (porque si llamamos $s_+ = \#\{x \in B \mid f(x) = y, \text{signo}(\det(\text{Jac}f(x))) = 1\}$ y $s_- = \#\{x \in B \mid f(x) = y, \text{signo}(\det(\text{Jac}f(x))) = -1\}$, tenemos $s_+ - s_- = \pm 1$ y entonces $s_+ + s_- = 2s_- \pm 1$).

Ahora veamos que el grado de Brouwer se preserve por pequeñas variaciones.

Lema 2.1.2. Sea $f : \overline{B} \subseteq \mathbb{R}^s \rightarrow \mathbb{R}^s$ de clase $C^1(B, \mathbb{R}^s)$ donde B es acotado. Si y es un valor regular de f y existe $\delta > 0$ de modo que si $\|f - g\|_{C^1} < \delta$ entonces $\deg(f, B, y) = \deg(g, B, y)$.

Demostración. Como \det es una función continua y $f^{-1}(y)$ es finito, existen entornos U_x que podemos tomar disjuntos tales que

$$\text{signo}(\det(\text{Jac}f(x))) = \text{signo}(\det(\text{Jac}f(z))) \quad \text{para todo } z \in U_x$$

Podemos tomar δ_1 de modo que $\text{signo}(\det(\text{Jac}g(z))) = \text{signo}(\det(\text{Jac}f(x)))$ para todo $z \in U_x$ para toda $g \in C^1(B, \mathbb{R}^s)$ con $\|f - g\|_{C^1} < \delta_1$. En particular, el Teorema de la Función Inversa vale para g y f en todo punto de U_x .

Ahora debemos demostrar que el si g esta suficientemente cerca de f , entonces existe un único $z_y \in U_x$ de modo que $g(z_y) = y$. Para esto, damos una versión general del Teorema de Lagrange o Valor Medio. Si $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ de clase C^1 , definiendo $U_i(t) = f(tz + (1 - t)x)$ tenemos que:

$$f_i(x) - f_i(z) = U_i(0) - U_i(1) = - \int_0^1 U_i'(t) dt = \int_0^1 \langle \nabla f_i(tz + (1 - t)x), (x - z) \rangle$$

Podemos reescribir esto para el caso $f = (f_1, \dots, f_n)$ de la siguiente manera

$$f(x) - f(z) = \left(\int_0^1 \text{Jac}f((1 - t)x + tz) dt \right) (x - z),$$

donde $\left(\int_0^1 \text{Jac}f((1 - t)x + tz) dt \right)$ corresponde a integrar cada coordenada de la matriz jacobiana.

Con esto presente, realizamos un cambio de variable para centrar nuestro problema en el origen. Sean $h = z - x$ y $D = y - g(x)$. Definimos las funciones

$$R(h) = g(x + h) - g(x) - \text{Jac}g(x)h \quad \text{y} \quad T(h) = (\text{Jac}g(x))^{-1}(D - R(h))$$

Notar que si $T(h) = h$, entonces $\text{Jac}g(x)h = (D - R(h)) = g(x) - y - (g(x+h) - g(x) - \text{Jac}g(x)h)$. Así obtenemos que $y = g(x+h) = g(z)$. Luego, nuestro problema se reduce a demostrar que T tiene un único punto fijo en un entorno $B_\delta(0)$. Pero en efecto,

$$\begin{aligned} R(h) - R(h') &= g(x+h) - g(x+h') - \text{Jac}g(x)(h-h') = \\ &= \left(\int_0^1 \text{Jac}g((x+th+(1-t)h'))(h-h')dt \right) - \text{Jac}g(x)(h-h') = \\ &= \left(\int_0^1 (\text{Jac}g((x+th+(1-t)h')) - \text{Jac}g(x))(h-h')dt \right) \end{aligned}$$

Notemos que además:

$$\begin{aligned} \|\text{Jac}g((x+th+(1-t)h')) - \text{Jac}g(x)\| &\leq \|\text{Jac}g((x+th+(1-t)h')) - \text{Jac}f(x+th+(1-t)h')\| + \\ &\quad \|\text{Jac}f(x+th+(1-t)h') - \text{Jac}f(x)\| + \|\text{Jac}f(x) - \text{Jac}g(x)\| \end{aligned}$$

Así que tomando un entorno suficientemente cercano de 0 para h y una función suficientemente cercana g , podemos acotar

$$\|\text{Jac}g((x+th+(1-t)h')) - \text{Jac}g(x)\| \leq \mu$$

Sea $M = \max\{\|\text{Jac}g(x)\|^{-1}\}$ y $M_f = \max\{\|\text{Jac}f(x)\|^{-1}\}$, luego tenemos que $|T(h) - T(h')| \leq \|\text{Jac}g(x)\|^{-1} \|R(h) - R(h')\| \leq M\mu \|h - h'\|$. Tomando, g suficientemente cercana a f , podemos hacer que $M \leq M_f + 1$. Más aún, tomando un entorno del 0 suficientemente chico y una g suficientemente cercana a f , podemos hacer que $\mu \leq \frac{1}{2}(M_f + 1)$. Así, T es una contracción y por Teorema de Banach admite un único punto fijo. Tomando un δ suficientemente chico, $B_\delta(x) \subseteq U_x$ para cada $x \in f^{-1}(y)$ como queríamos probar. \square

Una propiedad importante del grado de Brouwer es la invariancia por homotopía. Nosotros probaremos la invariancia cuando la homotopía es suave, pero vale también para el caso en el que sea solamente continua.

Lema 2.1.3. Sean $B \subset \mathbb{R}^s$ abierto y acotado, $f, g : \overline{B} \rightarrow \mathbb{R}^s$ dos funciones C^1 homotópicamente equivalentes por una homotopía suave $H : \overline{B} \times [0, 1] \rightarrow \mathbb{R}^s$ tal que $H(x, 0) = f(x)$ y $H(x, 1) = g(x)$. Si $y \notin H(\partial B \times [0, 1])$, entonces

$$\deg(f, B, y) = \deg(g, B, y).$$

Demostración. Supongamos que $\deg(f, B, y) \neq \deg(g, B, y)$. Definimos:

$$\lambda = \inf\{t \in [0, 1] : \deg(f, B, y) \neq \deg(H_t, B, y)\}.$$

Como H_λ es C^1 e $y \notin H_\lambda(\partial B)$, existe un $\varepsilon > 0$ de modo que si $\|G - H_\lambda\|_{C^1} < \varepsilon$, entonces $\deg(G, B, y) = \deg(H_t, B, y)$. Pero por lo dicho antes, existe un δ de modo que si $|\lambda - s| < \delta$ entonces $\deg(H_s, B, y) = \deg(H_\lambda, B, y)$. En particular, si $\lambda \neq 0$, podemos tomar $s = \lambda - \delta/2$ y obtenemos que $\deg(f, B, y) \neq \deg(H_s, B, y)$ y $s < \lambda$. Pero esto es absurdo. El absurdo provino de suponer que los grados de f y g en y eran distintos. Si $\lambda = 0$, tomando $s = \lambda + \delta/2$, para $t \leq s$ vale que $\deg(f, B, y) = \deg(H_t, B, y)$ contradiciendo que 0 es ínfimo. \square

Aunque los resultados que siguen en esta sección valen para un conjunto $B_c \subset \mathbb{R}_{>0}^s$ abierto, acotado y compacto más general, por conveniencia definiremos el siguiente conjunto B_c .

Lema 2.1.4. *Sea G una red de reacciones químicas y $\dot{x} = f_\kappa(x)$ el sistema de cinética de acción de masas resultante. Asumimos que G es conservativa y sin estados estacionarios en el borde de $\mathbb{R}_{\geq 0}^s$. Sean W una matriz de conservación de G reducida por filas, \mathcal{S}_c como en (1.2) tal que $\mathcal{S}_c \cap \mathbb{R}_{>0}^s \neq \emptyset$ y $\tilde{w} \in \mathbb{R}_{>0}^s$, $\tilde{c} > 0$ tales que $\tilde{w}x^T = \tilde{c}$ para todo $x \in \mathcal{S}_c$. Se define el conjunto*

$$B_c = \{x \in \mathbb{R}_{>0}^s : \tilde{w}x^T < 2\tilde{c}\}. \quad (2.2)$$

Vale que

1. B_c es abierto, acotado y convexo;
2. $\overline{B}_c = \{x \in \mathbb{R}_{\geq 0}^s : \tilde{w}x^T \leq 2\tilde{c}\}$;
3. $B_c \cap \mathcal{S}_c = \mathcal{S}_c \cap \mathbb{R}_{>0}^s$ y $\partial B_c \cap \mathcal{S}_c = \mathcal{S}_c \cap \partial \mathbb{R}_{\geq 0}^s$;
4. para todo $x \in \mathcal{S}_c \cap \partial \mathbb{R}_{\geq 0}^s$, vale que $x + \varepsilon f(x) \in \overline{B}_c$ para todo ε suficientemente chico.

Demostración. 1. Es evidente que B_c es abierto y convexo (es intersección de abiertos y convexos). Para observar que es acotado, notar que tanto \tilde{w} como x deben tener sus coordenadas positivas, luego estará contenido en una bola suficientemente grande.

2. La contención $\overline{B}_c \subseteq \{x \in \mathbb{R}_{\geq 0}^s : \tilde{w}x^T \leq 2\tilde{c}\}$ es clara. Para la otra contención, supongamos que $\tilde{w}z^T = 2\tilde{c}$. Notar que el punto $z_\varepsilon = z - \varepsilon \tilde{w}$ está arbitrariamente cerca de z y está contenido en B_c . Por último, supongamos sin perder generalidad que $z = (0, z_2, \dots, z_s)$ con $z_i \neq 0$ tal que $\tilde{w}z^T \leq 2\tilde{c}$. Como las coordenadas de w son positivas, podemos achicar la segunda coordenada y aumentar la primera para tener un punto sin coordenadas nulas que siga cumpliendo $\tilde{w}z^T \leq 2\tilde{c}$. Repitiendo el argumento anterior, obtenemos un punto arbitrariamente cerca de z dentro de B_c .

3. La primera igualdad es inmediata. Para la segunda, observar que el borde de \overline{B}_c esta compuesto por puntos donde alguna coordenada es nula y puntos que verifican $\tilde{w}z^T = 2\tilde{c}$. Pero es claro que estos últimos no pertenecen a \mathcal{S}_c . La otra contención es inmediata.

4. Efectivamente, basta tomar $\varepsilon < c/(\tilde{w}f(x)^T)$, donde observamos que si $\tilde{w}f(x) = 0$, entonces para cualquier ε vale la contención. \square

Teorema 2.1.3. *Sea G una red de reacciones químicas y $\dot{x} = f_\kappa(x)$ el sistema de cinética de acción de masas resultante. Asumimos que G es conservativa y sin estados estacionarios en el borde del ortante $\mathbb{R}_{\geq 0}^s$. Sean, N la matriz de estequiometría de G y W una matriz de conservación de G reducida por filas de rango $d = s - \text{rango}(N)$. Sea $c \in \mathbb{R}^d$, \mathcal{S}_c como en (1.2) tal que $\mathcal{S}_c \cap \mathbb{R}_{>0}^s \neq \emptyset$, B_c como en (2.2), y $f_{c,\kappa}$ como en (1.3), entonces*

$$\deg(f_{c,\kappa}, B_c, 0) = (-1)^{\text{rango}(N)}.$$

Demostración. Sin pérdida de generalidad, podemos asumir que W se puede escribir en bloques como

$$W = (I_d \quad \widehat{W}),$$

donde $\widehat{W} \in \mathbb{R}^{d \times (s-d)}$ e I_d es la matriz identidad de tamaño d . Notemos que si $Wx = Wy$, tenemos $(x_1, \dots, x_d)^T + \widehat{W}(x_{d+1}, \dots, x_s)^T = (y_1, \dots, y_d)^T + \widehat{W}(y_{d+1}, \dots, y_s)^T$. Llamemos π es la proyección a las últimas $s - d$ coordenadas de \mathbb{R}^s . De esta forma, si $Wx = Wy$ vale que

$$x = y \quad \text{si y solo si} \quad \pi(x) = \pi(y) \quad (2.3)$$

Elegimos ahora un punto arbitrario $\bar{x} \in \mathcal{S}_c \cap \mathbb{R}_{>0}^s$ y consideramos la función continua $G : \overline{B}_c \rightarrow \mathbb{R}^s$ definida por

$$G(x) = (Wx - c, \pi(\bar{x} - x)) \in \mathbb{R}^d \times \mathbb{R}^{\text{rango}(N)} \cong \mathbb{R}^s.$$

La matriz jacobiana de G tiene la forma

$$\text{Jac}G(x) = \begin{pmatrix} I_d & \widehat{W} \\ 0 & -I_{\text{rango}(N)} \end{pmatrix}.$$

Y entonces $\det(\text{Jac}G(x)) = (-1)^{\text{rango}(N)}$ para todo x . En particular, 0 es un valor regular de G . Más aún, si $G(x) = 0$, entonces $x \in \mathcal{S}_c$, porque $Wx = c$, y $\pi(\bar{x}) = \pi(x)$. Como $W\bar{x} = c$, de (2.3) concluimos que $x = \bar{x}$. Como $\bar{x} \notin \partial B_c$, se sigue que G no se anula en el borde de B_c . De esta forma,

$$\deg(G, B_c, 0) := \text{signo}(\det(\text{Jac}G(\bar{x}))) = (-1)^{\text{rango}(N)}.$$

Consideremos ahora la siguiente homotopía entre las funciones $f_{c,\kappa}$ y G :

$$\begin{aligned} H : \overline{B}_c \times [0, 1] &\rightarrow \mathbb{R}^s \\ (x, t) &\mapsto tf_{c,\kappa}(x) + (1 - t)G(x). \end{aligned}$$

Claramente H es continua. Para poder aplicar el Lema 2.1.3 para hallar el grado de $f_{c,\kappa}$, debemos mostrar que $H(\partial B_c \times [0, 1]) \neq 0$. Pero

$$H(x, t) = (Wx - c, t\pi(f_{c,\kappa}(x)) + (1 - t)\pi(\bar{x} - x)),$$

y entonces $H(x, t) = 0$ implica $Wx = c$ y por lo tanto $x \in \mathcal{S}_c$. De esta forma, lo que debemos mostrar es que

$$t\pi(f_{c,\kappa}(x)) + (1 - t)\pi(\bar{x} - x) \neq 0 \quad \text{para todo } x \in \mathcal{S}_c \cap \partial \mathbb{R}_{\geq 0}^s. \quad (2.4)$$

Para $t = 1$, (2.4) se sigue de (2.3) usando que $Wf_{c,\kappa}(x) = 0$ y $f_{c,\kappa}(x) \neq 0$ para $x \in \partial \mathbb{R}_{\geq 0}^s$ por hipótesis. Para $t = 0$, ya mostramos que G no se anula en ∂B_c y en particular no se anula en $\mathcal{S}_c \cap \partial \mathbb{R}_{\geq 0}^s$. Supongamos ahora que (2.4) no vale para $t \in (0, 1)$. Es decir, existe $x' \in \mathcal{S}_c \cap \partial \mathbb{R}_{\geq 0}^s$ tal que

$$\pi(f_{c,\kappa}(x')) = \frac{t-1}{t}\pi(\bar{x} - x').$$

Como $x' \in \mathcal{S}_c$, tenemos $Wx' = W\bar{x}$ y por lo tanto $W(\bar{x} - x') = 0$, y también $Wf_{c,\kappa}(x') = 0$. A partir de (2.3) deducimos que

$$f_{c,\kappa}(x') = \frac{t-1}{t}(\bar{x} - x'). \quad (2.5)$$

Por el Lema 2.1.4 tenemos que para ε suficientemente chico, $x' + \varepsilon f_{c,\kappa}(x') \in \overline{B}_c$, y en particular $x' + \varepsilon \frac{t-1}{t}(\bar{x} - x') \in \mathbb{R}_{>0}^s$. Pero para $i \in \{1 \dots, s\}$ tal que $x'_i = 0$ tenemos $(x' + \varepsilon \frac{t-1}{t}(\bar{x} - x'))_i = \varepsilon \frac{t-1}{t} \bar{x}_i < 0$ porque $\frac{t-1}{t} < 0$ y $\bar{x}_i > 0$ y esto es una contradicción. Luego, $H(x, t) \neq 0$ para todo $x \in \partial B_c$ y $t \in [0, 1]$. Como consecuencia, la invariancia por homotopía del grado de Brouwer nos da el resultado deseado

$$\deg(f_{c,\kappa}, B_c, 0) = \deg(G, B_c, 0) = (-1)^{\text{rango}(N)}.$$

□

Ahora podemos demostrar el Teorema 2.1.2.

Demostración (Teorema 2.1.2). Las hipótesis aseguran que podemos usar el Teorema 2.1.3. Si tomamos B_c como en (2.2), todos los estados estacionarios positivos se encuentran allí porque $B_c \cap \mathcal{S}_c = \mathcal{S}_c \cap \mathbb{R}_{>0}^s$ (ver Lema 2.1.4), y vale que

$$\deg(f_{c,\kappa}, B_c, 0) = (-1)^{\text{rango}(N)}.$$

Sea V_c el conjunto de estacionarios positivos en la clase de compatibilidad estequiométrica \mathcal{S}_c . Notemos que

$$V_c = \{x \in B_c : f_{c,\kappa}(x) = 0\}.$$

(A) Como $\text{signo}(\det(\text{Jac}f(x))) = (-1)^{\text{rango}(N)} \neq 0$ para todo estado estacionario en \mathcal{S}_c , 0 es un valor regular para $f_{c,\kappa}$. Podemos entonces aplicar el Teorema 2.1.3 y obtenemos

$$(-1)^{\text{rango}(N)} = \sum_{x \in V_c} \text{signo}(\det(\text{Jac}f(x))) = (-1)^{\text{rango}(N)} (\#V_c),$$

donde $\#V_c$ es el cardinal de V_c . Concluimos entonces que $\#V_c = 1$ y por lo tanto existe un único estado estacionario en la clase de compatibilidad estequiométrica. Más aún, como $\text{signo}(\det(\text{Jac}f(x))) \neq 0$ para todo estado estacionario, el estado estacionario que obtenemos es no degenerado.

(B) Sea $x^* \in V_c$ tal que $\text{signo}(\det(\text{Jac}f(x))) = (-1)^{\text{rango}(N)+1}$. Si 0 es un valor regular para $f_{c,\kappa}$, entonces la igualdad

$$(-1)^{\text{rango}(N)} = \sum_{x \in V_c} \text{signo}(\det(\text{Jac}f(x))) = (-1)^{\text{rango}(N)+1} + \sum_{x \in V_c, x \neq x^*} \text{signo}(\det(\text{Jac}f(x)))$$

implica que debe existir al menos otros dos puntos $x', x'' \in V_c$ tales que

$$\text{signo}(\det(\text{Jac}f(x'))) = \text{signo}(\det(\text{Jac}f(x''))) = (-1)^{\text{rango}(N)},$$

es decir, que hay al menos tres estados estacionarios positivos en \mathcal{S}_c , todos no degenerados. En este caso, por la Observación 2.1.1, hay un número impar de estados estacionarios y son todos no degenerados.

Si 0 no es un valor regular para $f_{c,\kappa}$, entonces debe existir otro estado estacionario positivo x' en \mathcal{S}_c para el cual la matriz jacobiana de $f_{c,\kappa}(x')$ es singular. Esto implica que hay al menos dos estados estacionarios positivos en \mathcal{S}_c , x^* y x' , uno de los cuales es no degenerado. □

Ahora que contamos con estas herramientas de Teoría de Grado, vamos a presentar un resultado que relaciona $\text{Jac}f$ con $\text{Jac}f_{c,\kappa}$ (del sistema ampliado). Introducimos primero un lema de Álgebra Lineal.

Lema 2.1.5. *Sea M una matriz de $n \times n$. Dado $I \subseteq \{1, \dots, n\}$, denotamos $M_{I,I}$ a la submatriz formada por los elementos M_{ij} con $i, j \in I$. Si $p = \lambda^n + \dots + p_1\lambda + p_0$ es el polinomio característico de M entonces:*

$$p_{n-s} = (-1)^s \sum_{I \subseteq \{1, \dots, n\}, \#I=s} \det(M_{I,I})$$

Demostración. Sea $q(x) = \det(xI + M)$, notar que $p(x) = (-1)^n q(-x)$, por lo que basta probar que:

$$q_{n-s} = \sum_{I \subseteq \{1, \dots, n\}, \#I=s} \det(M_{I,I})$$

Por Taylor sabemos que $q_{n-s} = \frac{q^{(n-s)}(0)}{(n-s)!}$.

- $q_0 = q(0) = \det(M)$.
- $q_1 = q'(0) = \lim_{x \rightarrow 0} \frac{\det(M + xI) - \det(M)}{x}$. Sea M_1 la matriz $M + xE_{11}$ (es decir, sumamos un x a la primera coordenada). Entonces, si notamos con $M[i, i]$ a la submatriz de M que se obtiene al eliminar las i -ésimas fila y columna, vemos que $\det(M_1) - \det(M) = x \det(M[1, 1])$ por multilinealidad y desarrollar por primera columna. Si consideramos $M_2 = M_1 + xE_{22}$, vale que $\det(M_2) - \det(M_1) = x \det(M[2, 2])$, y así sucesivamente. Luego:

$$\frac{\det(M + xI) - \det(M)}{x} = \sum_{i=1}^n \det(M_{n-i}) - \det(M_{n-i-1}) = \sum_{i=1}^n \det(M[i, i])$$

que es lo que queríamos ver.

- Notamos que con el razonamiento anterior podemos obtener $q'(x) = \sum_{i=1}^n \det((M + xI)[i, i])$. Derivando luego la expresión y repitiendo el mismo argumento para las submatrices $(M + xI)[i, i]$, y evaluando en $x = 0$ tenemos la fórmula para $q''(0)$, donde notemos que cada término aparece dos veces (pues $M[i, j][k, l] = M[k, l][i, j]$). Por ende $q''(0) = 2! \sum_{I \subseteq \{1, \dots, n\}, \#I=2} M_{I,I}$. Se puede ver que el caso general proviene de repetir este argumento.

□

Dada M una matriz de $n \times n$, sea S el espacio generado por sus columnas. Sea $s = \dim(S)$ y $d = n - s$. Notar que podemos colocar mediante un cambio de filas, s filas linealmente independientes en la parte inferior de la matriz.

Construimos una base de S^\perp de la forma $\{w_1, \dots, w_d\}$ de modo que $(w_i)_j = \delta_{ij}$ si $1 \leq i, j \leq d$, donde δ_{ij} es la delta de Kronecker. Definimos A como la matriz que posee en las primeras filas a los vectores w_i , y las últimas s filas coinciden con las últimas filas de M . Es decir

$$A = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \\ M_{\{d+1, \dots, n\}\{1, \dots, n\}} \end{pmatrix} \quad (2.6)$$

Luego tenemos el siguiente resultado de [25, Prop. 5.3]:

Proposición 2.1.1. *Sean A, M matrices definidas como arriba. Entonces tenemos que:*

$$\det(A) = \sum_{I \subseteq \{1, \dots, n\}, \#I=s} \det(M_{I,I})$$

Demostración. Por el lema anterior, basta probar que $\det(A)$ es $(-1)^d p_d$ donde p_d es el d -ésimo coeficiente del característico.

Definimos P como la matriz cuyas primeras d -filas coinciden con A y cuyas últimas filas coinciden con la identidad. Notar que $\det(P) = 1$, por lo que:

$$p = \det(\lambda I - M) = \det(\lambda P - PM)$$

Observemos que las primeras d filas de PM son nulas (ya que las primeras filas de P son elementos del ortogonal del espacio columna de M). Las últimas filas de PM coinciden con M (pues las últimas filas de P son los canónicos). Entonces:

$$\lambda P - MP = \begin{pmatrix} \lambda w_1 \\ \lambda w_2 \\ \vdots \\ \lambda w_d \\ \lambda I_{s \times s} - M_{\{d+1, \dots, n\}\{1, \dots, n\}} \end{pmatrix}$$

Usando multilinealidad para las primeras d filas, tenemos que $p = \det(\lambda P - MP) = \lambda^d \det(P'_\lambda)$, donde

$$P'_\lambda = \lambda I_{s \times s} - M_{\{d+1, \dots, d\}}$$

Notar que luego necesitamos el término de grado 0 de $\det(P'_\lambda)$, que ya sabemos que es $(-1)^s \det(P'_0)$ pero es claro que $\det(P'_0) = \det(A)$ por lo que demostramos lo buscado. \square

Notar que este resultado permite conectar $\text{Jac}(f)$ con $\text{Jac}(f_{c,\kappa})$, ya que este proviene de agregar las leyes de conservación a $\text{Jac}(f)$, que es el espacio ortogonal al subespacio estequiométrico.

Por último, agregaremos un resultado que nos permita ver cómo preservar la biestabilidad y cantidad de estados estacionarios al cambiar la red agregando reacciones. Para esto enunciamos el siguiente lema [10]:

Lema 2.1.6. *Sea $\Omega \subseteq \mathbb{R}^s$ un dominio acotado y $g_\lambda : \overline{\Omega} \rightarrow \mathbb{R}^s$ una familia de funciones suaves que varían suavemente con respecto a λ (de hecho, basta con pedir que la variación sea continua). Supongamos que:*

- $\det(\text{Jac}(g_0))|_{x=x^*} \neq 0 \neq \det(\text{Jac}(g_1))|_{x=x^*}$ para todo $x^* \in \mathbb{R}^s$
- g_λ no posee ceros en $\partial\Omega$.

Entonces g_0 y g_1 tienen la misma cantidad de ceros en Ω .

Demostración. Sea $a(x^*) = \det(\text{Jac}(g_0)|_{x=x^*})$. Por hipótesis, a no posee raíces, por lo que su signo es constante. Por lo tanto, el grado de Brouwer $\deg(g, 0, \Omega)$ coincide en módulo con la cantidad de ceros de g_0 . Análogamente vale para g_1 . Como ya habíamos demostrado la invarianza del grado de Brouwer, tenemos que g_0 y g_1 tienen la misma cantidad de ceros. \square

Enunciamos ahora un teorema que nos permite preservar la cantidad de estados estacionarios y su estabilidad al agregar reacciones [15]:

Teorema 2.1.4. *Sea N una subred de una red dada G de modo que el subespacio estequiométrico de ambas coinciden. Si κ corresponde a las constantes de reacción de G entonces:*

- *Si N admite múltiples estados estacionarios en una clase de compatibilidad estequiométrica, entonces G también. De hecho, si N tiene finitos estados estacionarios, la cantidad de estados de G es mayor igual a la cantidad de estados de N .*
- *Si N es biestable, es decir que admite múltiples estados estacionarios estables en una clase de compatibilidad estequiométrica, entonces G también. Análogamente si N tiene finitos estados estacionarios exponencialmente estables, la cantidad de estados estacionarios exponencialmente estables de G es mayor o igual a la de N .*

Demostración. Sea \mathcal{S} el espacio estequiométrico de N y G . Sean f_N y f_G los sistemas asociados a las redes N y G respectivamente. Luego notar que podemos separar los términos de f_G entre los que corresponden con reacciones que aparecen en N y los que no. Más precisamente:

$$f_G(x) = f_N(x) + \sum_{\alpha \xrightarrow{\kappa} \beta \notin \mathcal{R}_N} (\beta - \alpha) \kappa \mathbf{x}^\alpha$$

donde \mathcal{R}_N son las reacciones que están en N . Definimos entonces:

$$g_\lambda(x) := f_N(x) + \lambda \sum_{\alpha \xrightarrow{\kappa} \beta \notin \mathcal{R}_N} (\beta - \alpha) \kappa \mathbf{x}^\alpha$$

Sea x^* un estado estacionario no degenerado de N , es decir, para ciertas constantes de reaccion (τ_1, \dots, τ_j) (que corresponden a las reacciones que aparecen en N), tenemos que $f_{N,\tau}(x^*) = 0$. Sabemos por continuidad que existe una bola $\Omega = B_r(x^*)$ de modo que $\text{Jac}(f_N)$ sea inversible para todo punto en la bola. Más aún, tomando un radio suficientemente chico, podemos asegurar que x^* es el único estado estacionario no degenerado en $\overline{B_r(x^*)}$. En particular tenemos que $f_N(y) \neq 0$ para todo $y \in \partial\Omega$.

Ahora, notar que podemos interpretar $g_\lambda(x) = T(x, \lambda, \mathbf{k}')$ donde $\mathbf{k} = (\kappa_1, \dots, \kappa_m)$ corresponden a las reacciones en G pero no en N , y así vemos que es una función continua con respecto a \mathbf{k}' y que $T(x, \lambda, (0, \dots, 0))$ es no nula en $\partial\Omega \times [0, 1]$. Como este conjunto es compacto, existe $\varepsilon > 0$ de modo que para cualquier elección de constantes $\mathbf{k}' = (\kappa_1, \dots, \kappa_m) \in B_\varepsilon(0, \dots, 0)$, tenemos que $T(x, \lambda, \mathbf{k}') \neq 0$. Y observemos que siempre podemos tomar una elección donde sean todas positivas. Así verificaríamos que $g_\lambda(x)$ no posee ceros en $\partial\Omega$ para todo $\lambda \in [0, 1]$.

Usando un argumento análogo, observando que $\text{Jac}(g_\lambda) = R(x, \lambda, \mathbf{k}')$, podemos tomar una elección de constantes no nulas de modo que la matriz jacobiana sea no degenerada para todo $y \in \overline{\Omega}$ y todo $\lambda \in [0, 1]$. Luego aplicando el lema anterior, tenemos que $g_1 = f_G$ posee un cero $x_G^* \in \Omega$ estado estacionario para constantes de reacción $(\tau_1, \dots, \tau_j, \kappa_1, \dots, \kappa_m)$.

Para acabar la primera parte de la prueba, si N tiene finitos estados estacionarios, podemos luego tomar radios suficientemente chicos de modo que las bolas sean disjuntas y \mathbf{k}' sea suficientemente chico de modo que el argumento anterior valga para cada uno de los dominios.

Finalmente, para verificar que se preserva la estabilidad exponencial, realizamos un argumento idéntico en el que observamos que los autovalores varían continuamente con respecto a λ y \mathbf{k}' , y por ende al partir de un estado estable, podemos tomar un radio y variables \mathbf{k}' de modo que podamos preservar el signo de la parte real de los mismos. \square

2.2. Bifurcaciones de Hopf

Dado un sistema de ecuaciones diferenciales que depende de un parámetro, nos interesa encontrar valores del parámetro para los cuales algún punto de equilibrio cambia su condición de estabilidad. Desarrollemos algo de teoría de bifurcaciones a partir de [17].

Consideremos un sistema autónomo definido por

$$\dot{x} = f(x), \quad x \in \mathbb{R}^s, \quad (2.7)$$

donde f es suave (es decir, f es de clase $C^1(\mathbb{R}^s)$). Sea x_0 un equilibrio del sistema (i.e., $f(x_0) = 0$) y sea $\text{Jac}(f)(x_0)$ la matriz jacobiana de f evaluada en $x = x_0$. Se dice que un equilibrio es hiperbólico si no hay autovalores de $\text{Jac}(f)(x_0)$ sobre el eje imaginario.

La Figura 2.1 muestra resultados clásicos. Hay tres clases topológicas de equilibrios hiperbólicos en el plano: los nodos (focos) estables, los puntos silla, y los nodos (focos) inestables. Los nodos y los focos (con la correspondiente estabilidad) son topológicamente equivalentes pero se pueden identificar al mirar los autovalores.



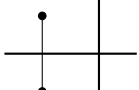

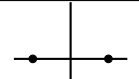

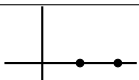
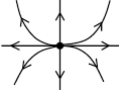
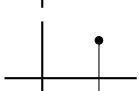

(n_+, n_-)	Autovalores	Diagrama de fases	Estabilidad
$(0, 2)$		 nodo	estable
		 foco	
$(1, 1)$		 silla	inestable
$(2, 0)$		 nodo	inestable
		 foco	

Figura 2.1: Clasificación topológica de los equilibrios en el plano [17, Figura 2.5]

Consideremos ahora el siguiente sistema autónomo que depende de un parámetro:

$$\dot{x} = f_\mu(x) = f(x, \mu), \quad x \in \mathbb{R}^s, \quad \mu \in \mathbb{R}, \quad (2.8)$$

donde f es suave con respecto a x y a μ . La aparición de un diagrama de fases topológicamente no equivalente cuando μ varía se llama una *bifurcación*.

Ejemplo 2.2.1. Consideremos el siguiente sistema en el plano:

$$\begin{cases} \dot{x}_1 &= \mu x_1 - x_2 - x_1(x_1^2 + x_2^2), \\ \dot{x}_2 &= x_1 + \mu x_2 - x_2(x_1^2 + x_2^2). \end{cases} \quad (2.9)$$

Evidentemente, en $x_1 = x_2 = 0$, tenemos un equilibrio. Se verifica además que la matriz Jacobiana de f_μ en $(0, 0)$ es:

$$\text{Jac}(f)|_{(0,0)} = \begin{pmatrix} \mu & -1 \\ 1 & \mu \end{pmatrix}$$

por lo que el característico es $p(\lambda) = (\lambda - \mu)^2 + 1$, o sea que los autovalores son $\mu + i$ y $\mu - i$. Observemos entonces que:

- si $\mu > 0$, en $(0, 0)$ tenemos un equilibrio inestable. En un diagrama de fases, esto se vería como que las trayectorias se alejan del punto.
- si $\mu < 0$, en $(0, 0)$ tenemos un equilibrio estable. En un diagrama de fases, esto se vería como que las trayectorias apuntan hacia el punto.

Así vemos que en $\mu = 0$ tenemos una bifurcación.

Definición 2.2.1. Consideremos el sistema autónomo que depende de un parámetro:

$$\dot{x} = f_\mu(x) = f(x, \mu), \quad x \in \mathbb{R}^s, \quad \mu \in \mathbb{R},$$

donde f es suave con respecto a x y a μ . Sea x_0 un equilibrio de f_{μ_0} . La aparición de una bifurcación para $\mu = \mu_0$ por la presencia de dos autovalores $\pm i\omega_0$ de $\text{Jac}(f_{\mu_0})(x_0)$ con $\omega_0 > 0$ se llama una *bifurcación de Hopf*.

Si $\text{Jac}(f_{\mu_0})(x_0)$ tiene un único par de autovalores imaginarios puros y los demás tienen parte real negativa, entonces hay una curva suave de equilibrios $x(\mu)$ con $x(\mu_0) = x_0$ y los autovalores $\lambda(\mu), \bar{\lambda}(\mu)$ de $\text{Jac}(f_\mu)(x(\mu))$ que son puramente imaginarios en $\mu = \mu_0$ varían suavemente con μ . Si además $d(\text{Re}(\lambda))/d\mu|_{\mu=\mu_0} \neq 0$, entonces hay una *bifurcación de Hopf simple*. Es decir, una bifurcación de Hopf simple se da cuando un solo par de autovalores de la matriz jacobiana cruza el eje imaginario mientras todos los demás permanecen con parte real negativa.

Fundamentalmente, esto nos habla de un cambio de tipo de estabilidad en x_μ , lo que a su vez implica que el sistema presenta comportamiento oscilatorio en alguno de los dos “lados” de la bifurcación. Esto es relevante para nosotros ya que la aparición de soluciones periódicas tiene interpretaciones biológicas.

Ejemplo 2.2.2. Retomando el Ejemplo 2.2.1, observemos que en $\mu = 0$, los autovalores son i y $-i$. Además, $\partial_\mu \text{Re}(\lambda(\mu)) = 1 \neq 0$, así que podemos decir que tenemos una bifurcación de Hopf simple.

En $\mu = 0$, es conocido que tenemos soluciones oscilatorias (linealizando, obtenemos órbitas circulares). Pero nos interesa entender esto para otros valores de μ . Con esto en mente, consideramos la variable compleja $z = x_1 + ix_2$ y usando que $z' = x_1' + ix_2'$ tenemos que el sistema 2.9 es equivalente a:

$$z' = (\mu + i)z - z|z|^2$$

Usando la representación polar, $z = re^{i\varphi}$ tenemos que por regla del producto, $z' = r'e^{i\varphi} + re^{i\varphi}i\varphi'$. Remplazando a ambos lados de la igualdad y despejando obtenemos la versión polar del sistema:

$$\begin{cases} r' = r(\mu - r^2) \\ \varphi' = 1 \end{cases} \quad (2.10)$$

De aquí podemos tener una mejor intuición geométrica. Si $\mu < 0$, el sistema no tiene soluciones fuera de $r' = 0$, es decir, de $P = (0, 0)$, lo cual se puede explicar con el hecho de que las trayectorias en este sistema vienen del infinito y llegan hasta el origen.

Pero en el caso $\mu > 0$, notar que si $r = \sqrt{\mu}$, tenemos que $r' = 0$ por lo que tenemos una solución que se mueve (con velocidad angular constante) por esa circunferencia, es decir, una solución periódica del sistema. Más aún, el radio de esa circunferencia varía continuamente con respecto a μ , por lo que si bien hay una pérdida de la estabilidad, para valores de μ positivos y cercanos al 0, las soluciones se mantienen cerca del origen. Este comportamiento se puede ver en la Figura 2.2.2.

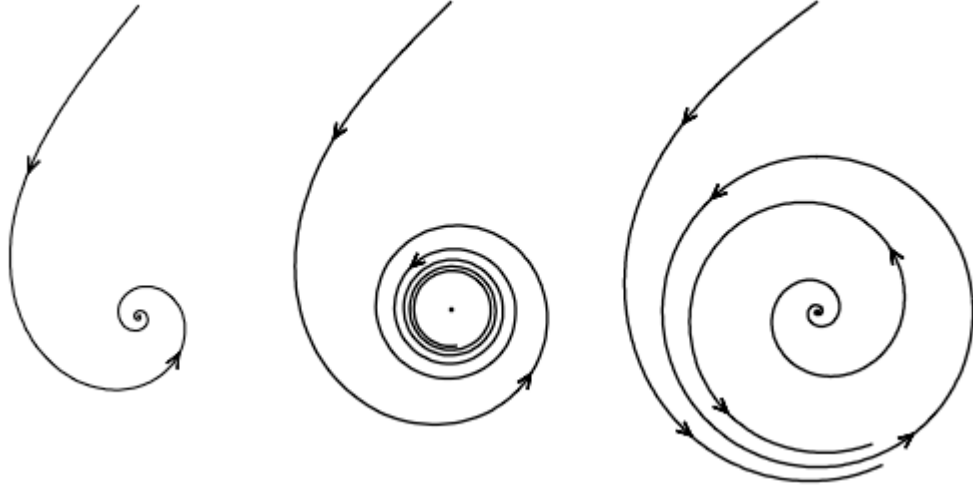


Figura 2.2: La figura de la izquierda muestra el caso $\mu < 0$, donde el equilibrio es estable. Cuando $\mu = 0$ tenemos un diagrama como el del centro. Si $\mu > 0$, tenemos un equilibrio inestable y una solución cíclica. [17]

En este caso, decimos que la bifurcación es supercrítica.

Afortunadamente, este ejemplo es representativo de cualquier bifurcación de Hopf en el siguiente sentido.

Teorema 2.2.1. *Para dimensión $n = 2$, si $\dot{x} = f_\mu(x)$ admite una bifurcación de Hopf supercrítica para $\mu = \mu_0$ en un cierto equilibrio $x = x_0$, entonces localmente el diagrama de fases es topológicamente equivalente al de (2.9). En particular, tomando un valor de μ cercano a μ_0 y del lado correcto de la bifurcación, tenemos una solución oscilatoria al sistema.*

La demostración de este teorema consiste en demostrar en primer lugar que la existencia de bifurcaciones de Hopf implica que el sistema es equivalente al sistema 2.9 más términos de orden 4 o más. Luego se puede demostrar que estos términos se pueden despreciar.

Ahora presentaremos un criterio que nos sirve para detectar bifurcaciones de Hopf simples sin tener que calcular autovalores. Primero definimos:

Definición 2.2.2 (Matriz de Hurwitz). Sea el polinomio

$$p = X^n + b_1 X^{n-1} + \dots + b_{n-1} X + b_n \in \mathbb{R}[x] \quad (2.11)$$

(consideramos $b_0 = 1$). Definimos la matriz de Hurwitz $H = (H_{ij})_{ij}$ donde $H_{ij} = b_{2i-j}$, con $b_k = 0$ si $k < 0$ o si $k > n$. Es decir,

$$H = \begin{pmatrix} b_1 & b_0 & 0 & 0 & 0 & \dots & 0 \\ b_3 & b_2 & b_1 & b_0 & 0 & \dots & 0 \\ & & \vdots & \vdots & & & \\ b_{2n-1} & b_{2n-2} & b_{2n-3} & b_{2n-4} & b_{2n-5} & \dots & b_n \end{pmatrix} \quad (2.12)$$

Definimos además D_j como el determinante del menor principal j -ésimo de la matriz H .

Definición 2.2.3. Dado un sistema autónomo como en (2.7), si todos los autovalores del polinomio característico de su matriz jacobiana en el equilibrio x_0 ,

$$p(\lambda) = \det(\lambda I_s - \text{Jac}f(x_0)) \quad (2.13)$$

tienen parte real negativa, entonces x_0 es estable y se dice que el *polinomio es estable*.

Criterio 2.2.1 (Routh-Hurwitz). *El polinomio p en (2.13) es estable si y solo si todos los determinantes de los menores principales de su matriz de Hurwitz son positivos:*

$$D_1 > 0, D_2 > 0, \dots, D_n = \det(H) = b_n D_{n-1} > 0.$$

O, lo que es equivalente,

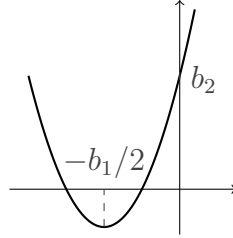
$$b_n > 0, D_1 > 0, D_2 > 0, \dots, D_{n-1} > 0.$$

Ejemplo 2.2.3. Consideremos $p(\lambda) = \lambda^2 + b_1\lambda + b_2$. La matriz de Hurwitz de p es:

$$H = \begin{pmatrix} b_1 & 1 \\ 0 & b_2 \end{pmatrix}$$

El criterio nos dice que p es estable si y solo si $b_2 > 0$ y $D_1 = b_1 > 0$.

En efecto, si las dos raíces de p son reales, serán ambas negativas si y solo si $b_1 > 0$ y $b_2 > 0$:



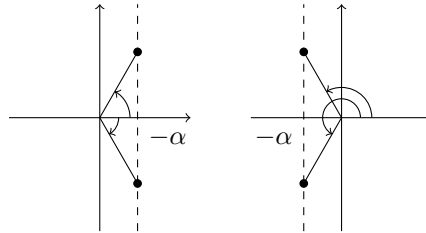
Si $\alpha + i\beta$ es raíz de p con $\beta \neq 0$, entonces

$$p(\lambda) = (\lambda - \alpha - i\beta)(\lambda - \alpha + i\beta) = \lambda^2 - 2\alpha\lambda + (\alpha^2 + \beta^2).$$

Necesariamente $b_2 > 0$ y vale que $\alpha < 0$ si y solo si $b_1 > 0$.

Presentamos aquí una idea de la demostración del Criterio de Routh-Hurwitz, basada en [7]. Consideremos el argumento de un número complejo en el intervalo $(-\frac{\pi}{2}, \frac{3\pi}{2}]$. Sean $\alpha, \beta, \omega \in \mathbb{R}$. El argumento de $\omega i - (\alpha + \beta i) = -\alpha + (\omega - \beta)i$ va a variar

- de $-\frac{\pi}{2}$ a $\frac{\pi}{2}$ si $\alpha < 0$ o
- de $\frac{3\pi}{2}$ a $\frac{\pi}{2}$ si $\alpha > 0$



cuando movemos ω de $-\infty$ a $+\infty$. Dicho de otra forma, el argumento de $\omega i - (\alpha + \beta i)$

- aumenta en π si $\alpha < 0$ o
- disminuye en π si $\alpha > 0$

cuando ω va de $-\infty$ a $+\infty$.

Si p no tiene raíces imaginarias puras y llamamos n_+ y n_- a la cantidad de raíces con parte real positiva y negativa respectivamente (contadas con multiplicidad), entonces $n = \text{gr}(p) = n_+ + n_-$. Si $\alpha_1 + \beta_1 i, \dots, \alpha_n + \beta_n i$ son las n raíces de p , tenemos

$$p(i\omega) = (i\omega - \alpha_1 - \beta_1 i) \dots (i\omega - \alpha_n - \beta_n i),$$

y el aumento en el argumento de $p(i\omega)$ cuando ω varía entre $-\infty$ y $+\infty$ está dado por

$$\Delta = (n_- - n_+)\pi. \quad (2.14)$$

Como $n_+ = n - n_-$, podemos despejar

$$n_- = \frac{1}{2}\left(n + \frac{\Delta}{\pi}\right). \quad (2.15)$$

Escribamos ahora

$$i^n p(-iX) = p_0(X) + ip_1(X), \quad (2.16)$$

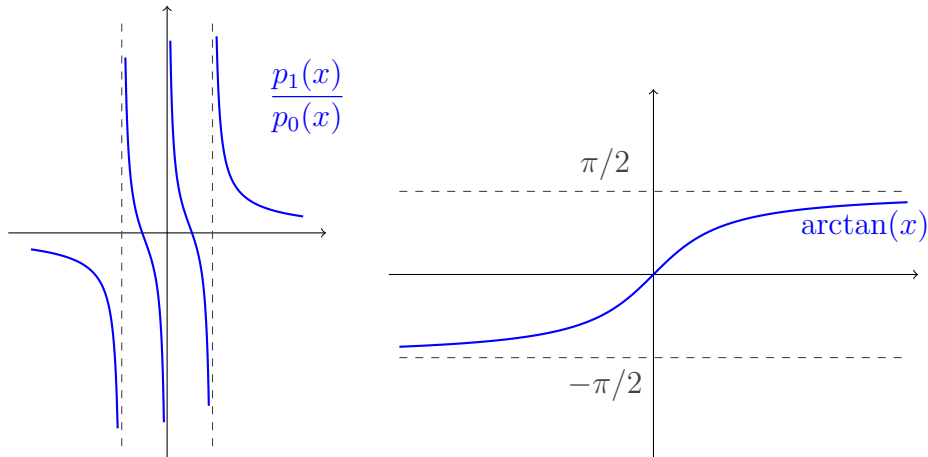
donde p_0 y p_1 son polinomios en $\mathbb{R}[X]$ unívocamente determinados, p_0 tiene grado n y es mónico, y el grado de p_1 es menor o igual a $n - 1$. Por ejemplo, cuando $n = 3$ tenemos

$$p(X) = X^3 + b_1 X^2 + b_2 X + b_3,$$

$$i^3 p(-iX) = -i(iX^3 - b_1 X^2 - ib_2 X + b_3) = p_0(X) + ip_1(X),$$

con $p_0(X) = X^3 - b_2 X = X(X^2 - b_2)$ y $p_1(X) = b_1 X^2 - b_3$.

Notemos que si p no tiene raíces imaginarias puras, entonces p_0 y p_1 no tienen raíces reales en común. Entonces Δ es igual a la disminución neta del argumento de $i^n p(-i\omega)$ cuando ω varía entre $-\infty$ y $+\infty$ que, a su vez, es la disminución neta de $\arctan(p_1(\omega)/p_0(\omega))$ cuando ω varía entre $-\infty$ y $+\infty$.



Sean $x_1 < x_2 < \dots < x_m$ los ceros reales distintos de p_0 , y sea Δ_k el decrecimiento de $\arctan(p_1(\omega)/p_0(\omega))$ cuando ω aumenta de x_k a x_{k+1} ($k = 0, 1, \dots, m$; $x_0 = -\infty$, $x_{m+1} = +\infty$). Como p_0 y p_1 no se anulan simultáneamente en valores reales:

$$\Delta_k = \frac{1}{2}\pi \left[\text{signo} \left(\lim_{x \rightarrow x_k^+} \frac{p_1(x)}{p_0(x)} \right) - \text{signo} \left(\lim_{x \rightarrow x_{k+1}^-} \frac{p_1(x)}{p_0(x)} \right) \right]$$

para $k = 1, \dots, m-1$. Además, como p_1/p_0 tiende a cero cuando $x \rightarrow \pm\infty$,

$$\Delta_0 + \Delta_m = \frac{1}{2}\pi \left[\text{signo} \left(\lim_{x \rightarrow x_m^+} \frac{p_1(x)}{p_0(x)} \right) - \text{signo} \left(\lim_{x \rightarrow x_1^-} \frac{p_1(x)}{p_0(x)} \right) \right].$$

De esta forma,

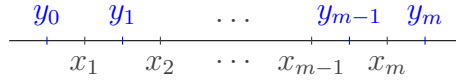
$$\Delta = \sum_{k=0}^m \Delta_k = \frac{1}{2}\pi \sum_{k=1}^m \left[\text{signo} \left(\lim_{x \rightarrow x_k^+} \frac{p_1(x)}{p_0(x)} \right) - \text{signo} \left(\lim_{x \rightarrow x_k^-} \frac{p_1(x)}{p_0(x)} \right) \right].$$

Introducimos el concepto de *índice* de una función racional $R(x)$ en un intervalo abierto (a, b) , y denotado por $I_a^b(R)$ como la diferencia entre la cantidad de veces que R salta de $-\infty$ a $+\infty$ y la cantidad de veces que salta de $+\infty$ a $-\infty$ a medida que x aumenta de a a b .

Es fácil ver que si $c \in (a, b)$ es tal que R no da un salto en c , entonces

$$I_a^b(R) = I_a^c(R) + I_c^b(R).$$

Volviendo a nuestro razonamiento, consideremos y_0, y_1, \dots, y_m tales que $y_0 < x_1 < y_1 < x_2 < \dots < x_{m-1} < y_{m-1} < x_m < y_m$.



Tenemos que

$$\begin{aligned} I_{y_{k-1}}^{y_k}(p_1/p_0) &= \begin{cases} 1 & \text{si } p_1/p_0 \text{ salta de } -\infty \text{ a } +\infty \text{ en } x_k \\ -1 & \text{si } p_1/p_0 \text{ salta de } +\infty \text{ a } -\infty \text{ en } x_k \\ 0 & \text{si } \lim_{x \rightarrow x_k^+} p_1(x)/p_0(x) = \lim_{x \rightarrow x_k^-} p_1(x)/p_0(x) \end{cases} \\ &= \frac{1}{2} \left(\text{signo} \left(\lim_{x \rightarrow x_k^+} \frac{p_1(x)}{p_0(x)} \right) - \text{signo} \left(\lim_{x \rightarrow x_k^-} \frac{p_1(x)}{p_0(x)} \right) \right). \end{aligned}$$

Y por lo tanto

$$\Delta = \pi \sum_{k=1}^m I_{y_{k-1}}^{y_k}(p_1/p_0) = \pi I_{y_0}^{y_m}(p_1/p_0),$$

lo que nos da, a partir de (2.15) y del hecho de que $I_{y_0}^{y_m}(p_1/p_0) = I_{-\infty}^{+\infty}(p_1/p_0)$:

$$n_- = \frac{1}{2}(n + I_{-\infty}^{+\infty}(p_1/p_0)). \quad (2.17)$$

Ahora, si f posee raíces imaginarias puras, denotemosla iw_1, \dots, iw_j . Denotamos $d(z) = (z + w_1) \dots (z + w_j)$. Luego $f(z) = d(z)g(z)$, por lo que g no posee raíces imaginarias puras. Luego

$$i^n f(-iz) = (p_0(z) + ip_1(z))/d(z)$$

Por lo visto, tenemos luego que f posee $\frac{1}{2}(n - j - I_{-\infty}^{+\infty}(p_1/p_0))$ ceros con parte real negativa, por lo que para tener estabilidad necesitamos que $j = 0$ y que $I_{-\infty}^{+\infty}(p_1/p_0) = \pm n$. La vuelta también vale.

Veamos entonces cómo calcular $I_{-\infty}^{+\infty}(p_1/p_0)$.

Consideramos la siguiente sucesión de polinomios de forma recursiva:

- p_0, p_1 como ya definidos.
- Para $1 \leq k \leq n$, escribimos mediante Euclides, $p_{k-1} = qp_k + r$. Definimos en tal caso $p_{k+1} := -r = -p_{k-1} + qp_k$. Si $p_{k-1} - qp_k = 0$ en su lugar definimos $p_{k+1} = p_k$ (notar que en tal caso el algoritmo se estabiliza).

Notar que esto verifica que:

- $p_0(c) = 0$ entonces $p_1(c) \neq 0$ ya que p no tiene raíces imaginarias puras.
- Si $d = (p_0 : p_1)$, entonces $d|p_j$ para todo $0 \leq j \leq n$. Más aún, de manera idéntica al algoritmo de Euclides, si $-p_{k-1} + qp_k = 0$, podemos ir para atrás y demostrar que $p_k|p_0$ y $p_k|p_1$. Así, $p_k = (p_0 : p_1)$.
- Como $\deg(p_k) < \deg(p_{k-1})$ si $p_k \neq p_{k+1}$, debe suceder que $\deg(p_n) = \deg(p_{n-1})$ y por lo anterior $p_n = (p_0 : p_1)$. En particular p_n no posee raíces reales ya que p no posee raíces imaginarias puras.
- Si $p_k(c) = 0$, entonces $p_{k-1}(c)p_{k+1}(c) < 0$. Observar que uno es el opuesto del otro. Lo que habría que ver es que no son 0. En caso de serlo, tendríamos que c es raíz común de los tres polinomios. En particular, c es raíz real de $(p_k : p_{k-1})$. Pero es sencillo verificar que, análogamente al algoritmo de Euclides, $(p_k : p_{k+1}) = (p_0 : p_1)$ que ya dijimos que no tenía raíces reales.

Denotamos a esta sucesión una *serie de Sturm* asociada a $\{p_0, p_1\}$. Denotaremos $v(c)$ a la cantidad de cambios de signos dentro de la secuencia $\{p_0(c) \dots p_n(c)\}$. Afirmamos que v es una función localmente constante en todo un intervalo (a, b) salvo para finitos puntos.

Proposición 2.2.1. *Sea p_0, p_1 y v como ya definidas. Si c no es cero con multiplicidad impar de p_0 , entonces $v(c^+) = v(c^-)$. En particular si p_0 no posee ceros en un intervalo (a, b) , entonces V es constante en ese intervalo.*

Demostración. Si c no es cero de ningún polinomio, entonces es evidente. Si c es cero de p_k notar que $k \neq n$ (ya que sino p_0, p_1 tendrían raíz real común). Supogamos que $k \neq 0$. Por lo visto antes, $p_{k-1}(c), p_{k+1}(c)$ tienen signos distintos y son no nulos. Por ende en un intervalo, $V[\{p_1(x), \dots, p_n(x)\}]$ se mantiene.

Veamos el caso $k = 0$, es decir, que c es un cero de multiplicidad par de p_0 . Claramente el signo de p_0 se preserva en un intervalo. Pero por lo ya mencionado, $p_1(c) \neq 0$, así que en un intervalo también se preserva el signo. Luego $v(c)$ se preserva. \square

Veamos ahora como esto se conecta con el índice de p_1/p_0 :

Proposición 2.2.2. *Si c es un cero de multiplicidad impar de p_0 entonces existe $\varepsilon > 0$ tal que:*

$$v(c^+) - v(c^-) = -I_{c-\varepsilon}^{c+\varepsilon}(p_1/p_0)$$

Demostración. Elegimos $\varepsilon > 0$ de modo que p_0 no tenga otras raíces en $(c - \varepsilon, c + \varepsilon)$. Si el índice es $+1$, es decir, se mueve de $-\infty$ a $+\infty$, entonces es porque de un lado p_0 y p_1 tienen signos distintos y del otro lado tienen el mismo signo. Observar que “perdemos” un cambio de signo en la secuencia de Strum. Como antes, achicando ε si es necesario, no tenemos variaciones en la cantidad de cambios de signos de p_1 en adelante. En particular, $v(c^+) = v(c^-) - 1$. El caso en el que el índice es negativo es análogo. \square

Teorema 2.2.2 (Strum). *Para p_0, p_1, v como ya definidas, vale que:*

$$v(b) - v(a) = I_a^b(p_1/p_0)$$

Demostración. Como v es una función localmente constante salvo para finitos puntos, tenemos que:

$$v(b) - v(a) = \sum_{c \text{ discontinuidad de } v} v(c^+) - v(c^-) = \sum_c I_{c-\varepsilon}^{c+\varepsilon} p_1/p_0 = I_a^b p_1/p_0$$

ya que si c no es raíz de p_0 o es raíz de índice par, entonces el índice no varía. \square

Por último, veamos qué pasa cuando $a = -\infty$ y $b = +\infty$. Si $c_k x^{n_k}$ es el monomio de mayor grado de p_k , evidentemente el signo de p_k en $-\infty$ será el signo de $c_k(-1)^{n_k}$, y en $+\infty$ será el signo de c_k .

Debido a que el índice racional no varía al extender el intervalo sin agregar raíces de p_0 , consideramos un valor de a y b suficientemente grande para los cuales las raíces estén en ese intervalo. En tal caso tenemos por Strum que $v(a) - v(b) = I_a^b p_1/p_0 = I_{-\infty}^{+\infty} p_1/p_0$. Tomando a y b suficientemente grande, tenemos que:

$$I_{-\infty}^{+\infty} p_1/p_0 = V[(-1)^{n_0} c_0, \dots, (-1)^{n_s} c_s] - V[c_0, \dots, c_s]$$

donde $V[\lambda_1, \dots, \lambda_s]$ es la cantidad de cambios de signos en la secuencia $(\lambda_1, \dots, \lambda_s)$.

Finalmente, lo que conseguimos es que p es estable, es decir tiene $n_- = n$, si y solo si $I_{-\infty}^{+\infty}(p_1/p_0) = n$. Equivalentemente, p es estable si y solo si

$$V[(-1)^{n_0} c_0, \dots, (-1)^{n_s} c_s] - V[c_0, \dots, c_s] = n.$$

Si $n_s = n$, esto solo pasa si $V[(-1)^{n_0} c_0, \dots, (-1)^{n_s} c_s] = n$ y $V[c_0, \dots, c_s] = 0$. Es decir:

Teorema 2.2.3. *El polinomio p es estable si y solo si*

$$p_k(X) = c_k X^{n-k} + \dots \quad (c_k \neq 0; k = 0, 1, \dots, n)$$

y los coeficientes principales c_k ($k = 0, 1, \dots, n$) todos tienen el mismo signo.

Para vincular esta cuenta con los determinantes de la matriz de Hurwitz que presenta el Criterio de Routh-Hurwitz, lo veremos para el caso $n = 3$. Tenemos $p(X) = X^3 + b_1 X^2 + b_2 X + b_3$ y suponemos $b_3 \neq 0$ (si $b_3 = 0$, sabremos que $X = 0$ es raíz y nos reducimos al estudio de un polinomio de grado 2). Notemos también que b_1 es menos la suma de las raíces de p y deberá ser positivo en caso de que todas las raíces de p tengan parte real negativa. Como vimos antes, $i^3 p(-iX) = p_0(X) + i p_1(X)$ con $p_0(X) = X^3 - b_2 X = X(X^2 - b_2)$, $p_1(X) = b_1 X^2 - b_3$. Calculemos ahora los polinomios p_2, \dots . Como asumimos $b_1 \neq 0$, $p_0(X) = \frac{1}{b_1} X p_1(X) - (b_2 - \frac{b_3}{b_1}) X$ y por lo tanto $p_2(X) = (b_2 - \frac{b_3}{b_1}) X$.

- si $b_2 = \frac{b_3}{b_1}$, entonces $p(X) = X^3 + b_1 X^2 + \frac{b_3}{b_1} X + b_3 = X^2(X + b_1) + \frac{b_3}{b_1}(X + b_1) = (X^2 + \frac{b_3}{b_1})(X + b_1)$. Pero las raíces de $X^2 + \frac{b_3}{b_1}$ o bien están sobre el eje imaginario, o bien son una positiva y una negativa. Para que p sea estable, necesariamente $b_2 \neq \frac{b_3}{b_1}$.
- si $b_2 \neq \frac{b_3}{b_1}$, $p_1(X) = \frac{b_1}{b_2 - \frac{b_3}{b_1}} p_2(X) - b_3$ y entonces $p_3(X) = b_3$. Tenemos así $c_0 = 1$, $c_1 = b_1$, $c_2 = b_2 - \frac{b_3}{b_1}$ y $c_3 = b_3$. Y por lo tanto, por el Teorema 2.2.3 y como $c_0 = 1 > 0$, p es estable si y solo si

$$b_1 > 0, \quad b_2 + \frac{b_3}{b_1} > 0, \quad \text{y} \quad b_3 > 0.$$

La matriz de Hurwitz de p es:

$$H = \begin{pmatrix} b_1 & 1 & 0 \\ b_3 & b_2 & b_1 \\ 0 & 0 & b_3 \end{pmatrix}$$

y por lo tanto $D_1 = b_1$, $D_2 = b_1 b_2 - b_3$ y $D_3 = b_3(b_1 b_2 - b_3) = b_3 D_2$. El criterio de Routh-Hurwitz nos dice entonces que p es estable si y solo si

$$b_1 > 0, \quad b_2 - \frac{b_3}{b_1} > 0 \quad \text{y} \quad b_3 > 0.$$

Nosotros lo que vamos a usar para estudiar la dinámica de la red ERK es un criterio que presentamos en el siguiente teorema. Está relacionado con el criterio de Routh-Hurwitz y permite detectar bifurcaciones de Hopf simples a partir de los coeficientes del polinomio característico de la matriz jacobiana del sistema en el equilibrio correspondiente [19].

Teorema 2.2.4. *Sea $\dot{x} = f_\mu(x)$ un sistema como en (2.8), y $\mu_0 \in \mathbb{R}$. Sea x_0 un estado estacionario de $\dot{x} = f_{\mu_0}(x)$. Sea*

$$p = X^n + b_1 X^{n-1} + \dots + b_{n-1} X + b_n \in \mathbb{R}[X]$$

el polinomio característico de $\text{Jac} f_{\mu_0}(x_0)$, y H la matriz de Hurwitz de p . Son equivalentes:

- $b_n > 0$, $D_1 > 0$, \dots , $D_{n-2} > 0$, $D_{n-1} = 0$ y $dD_{n-1}/d\mu|_{\mu=\mu_0} \neq 0$.
- $\dot{x} = f_\mu(x)$ tiene una bifurcación de Hopf simple en x_0 para $\mu = \mu_0$.

Antes de demostrar el resultado, demostraremos unos lemas previos.

Lema 2.2.1. *Sea $A_\mu = A(\mu)$ una familia de matrices de rango $n - 1$ que varía continuamente con respecto a μ . Definimos $p(\mu, x) = p_\mu(x) = \det(xI - A_\mu)$. Supongamos que $x = x(\mu)$ es una función suave con respecto a μ tal que $x(\mu_0) = 0$. Luego existe $C \neq 0$ tal que:*

$$\partial_\mu p(\mu_0, x(\mu_0)) = C \partial_\mu x(\mu_0).$$

Demostración. Como A tiene rango $n - 1$, notar que $p(\mu, x(\mu)) = x(\mu)q(\mu, x(\mu))$, donde q tiene grado $n-1$ en $x(\mu)$ y $q(0, x(0)) \neq 0$. Tomando $C = q(0, \mu(0))$ llegamos a lo buscado. \square

Lema 2.2.2. *Sean $\{A_\mu\}$ y $x = x(\mu)$ igual que en lema anterior. Consideremos $B(\mu) = B_\mu$ familia de matrices de $n \times n$ inversibles para todo μ y que varía suavemente con respecto a μ . Definimos:*

$$f(\mu) = \det(x(\mu)B(\mu) + A(\mu)).$$

Entonces existe $C \neq 0$ tal que $\partial_\mu f(\mu_0) = C \partial_\mu x(\mu_0)$.

Demostración. Usando que B_μ es inversible, notar que:

$$f(\mu) = \det(B(\mu))(x(\mu)I - (-B^{-1}(\mu)A(\mu))).$$

Si $E(\mu) := -B^{-1}(\mu)A(\mu)$, notar que estamos en las hipótesis del resultado anterior, y que $f(\mu_0) = \det(0.I + B^{-1}(\mu_0)A(\mu_0)) = 0$. Luego usando C' como en el lema anterior para $E(\mu)$, tenemos que $C = C' \det(B(\mu))$ cumple lo pedido. \square

Ahora sí, estamos en condiciones de demostrar el teorema.

Demostración del Teorema 2.2.4. Primero demostraremos que si tenemos una bifurcación simple, entonces se verifica la condición de los menores.

Usando que tenemos una Bifurcación de Hopf en $\mu = \mu_0$ y la suavidad de los autovalores, sabemos que en un entorno de μ_0 , existen funciones $\alpha, \beta(\mu)$ suaves de modo que:

$$p_\mu(\lambda) = (\lambda^2 + \alpha(\mu)\lambda + \beta(\mu))q_\mu(\lambda) \quad (2.18)$$

donde $\beta(\mu) > 0$, $\alpha^2(\mu) - 4\beta(\mu) < 0$. Y más aún $\alpha(\mu_0) = 0$ y $\partial_\mu \alpha(\mu_0) \neq 0$.

Por otro lado, las raíces de q_μ tienen todas parte real negativa. Por ende, verifican la condición de Routh-Hurwitz. Es decir, los determinantes Q_i de los menores de la matriz de Hurwitz son positivos.

Ahora escribimos $p = p_0(\mu) + p_1(\mu)\lambda + \dots + p_n(\mu)\lambda^n$ y $q = q_0(\mu) + \dots + q_{n-2}(\mu)\lambda$. Entonces, por (2.18), tenemos que:

$$p_i = \beta q_i + \alpha q_{i-1} + q_{i-2}$$

para $i = 0, \dots, n$ (recordar que en caso de tener un índice inválido, definimos estos coeficientes como 0). De aquí notar que

$$L_{n-1}(\mu) = M_{n-1}(\mu) + \alpha(\mu)N_{n-1}(\mu) \quad (2.19)$$

con

$$L_{n-1}(\mu) = \begin{pmatrix} p_1(\mu) & p_0(\mu) & 0 & 0 & 0 & \dots & 0 \\ p_3(\mu) & p_2(\mu) & p_1(\mu) & p_0(\mu) & 0 & \dots & 0 \\ & & \vdots & \vdots & & & \\ p_{2n-3}(\mu) & p_{2n-4}(\mu) & p_{2n-5}(\mu) & p_{2n-6}(\mu) & p_{2n-7}(\mu) & \dots & p_{n-1}(\mu) \end{pmatrix}$$

$$N_{n-1}(\mu) = \begin{pmatrix} q_0(\mu) & 0 & 0 & 0 & 0 & \dots & 0 \\ q_2(\mu) & q_1(\mu) & 0 & 0 & 0 & \dots & 0 \\ & & \vdots & \vdots & & & \\ q_{2n-4}(\mu) & q_{2n-5}(\mu) & q_{2n-6}(\mu) & q_{2n-7}(\mu) & q_{2n-8}(\mu) & \dots & q_{n-2}(\mu) \end{pmatrix}$$

$$M_{n-1}(\mu) = \begin{pmatrix} \beta(\mu)q_1(\mu) & \beta(\mu)q_0(\mu) & \dots & 0 \\ \beta(\mu)q_3(\mu) + q_1(\mu) & \beta(\mu)q_2(\mu) + q_0(\mu) & \dots & 0 \\ & \vdots & & \\ \beta(\mu)q_{2n-3}(\mu) + q_{2n-5}(\mu) & \beta(\mu)q_{2n-4}(\mu) + q_{2n-6}(\mu) & \dots & \beta(\mu)q_{n-1}(\mu) + q_{n-3}(\mu) \end{pmatrix}$$

Observamos que $\det(N_{n-1}(\mu)) = q_0(\mu)Q_{n-2}$ que es distinto de cero al ser ambos distintos de cero (recordar que q_μ no tiene a 0 como raíz), por lo que N_{n-1} es inversible.

Ahora, denotando M_i a la submatriz principal de $i \times i$ de M_{n-1} , al calcularle el determinante, podemos sacar el β en la primera fila y simplificar la segunda. Luego sacar el β en la segunda y simplificar la tercera. Así:

- $\det(M_1(\mu)) = \beta(\mu)q_1(\mu) = \beta(\mu)Q_1(\mu)$. Por el criterio de Hurwitz, Q_1 debe ser positivo, así el determinante queda positivo.

- $\det(M_2(\mu)) = \beta^2 Q_2(\mu)$ que es positivo por los mismos argumentos.
- En general, si $k \neq n - 1$ $\det(M_k(\mu)) = \beta^k Q_k$ que siempre es positivo.
- Pero para $k = n - 1$, observar que $q_{2n-i} = 0$ si $3 \leq i \leq n - 1$ por lo que al realizar este proceso, la última fila nos queda 0, así que $\det(M_{n-1}(\mu)) = 0$.

Es un resultado clásico de álgebra lineal luego que $M_{n-1}(\mu)$ tiene rango exactamente $n - 2$. Por ende, debido al lema y usando que $0 \neq \partial_\mu \operatorname{Re}(\lambda(\mu_0)) = -1/2 \partial_\mu \alpha(\alpha_0)$, verificamos que la derivada del determinante del último menor no se anula.

Usando que $\alpha(\mu_0) = 0$, tenemos que $D_i = \det(L_i(\mu_0)) = \det(M_i(\mu_0)) = 0$ si $0 \leq i \leq n - 2$, y que $D_{n-1} = \det(L_{n-1}(\mu_0)) = 0$.

Supongamos ahora que valen las condiciones sobre D_i y la derivada de α . Queremos ver que el sistema tiene una bifurcación de Hopf simple en $\mu = \mu_0$.

Como la derivada de $D_{n-1}(\mu)$ no se anula en $\mu = \mu_0$ y $D_{n-1}(\mu_0) = 0$, en algún de los dos lados de μ_0 , este determinante debe ser positivo, y del otro lado debe ser negativo. Sin pérdida de generalidad, asumimos que $D_{n-1}(\mu) < 0$ si $\mu_0 - \delta \leq \mu \leq \mu_0$. Entonces para tales valores de μ , el polinomio $p_\lambda(\mu)$ tiene todas sus raíces con parte real negativa por el criterio de Hurwitz (tomando un entorno en el cual los otros determinantes sigan siendo negativos, claro está). Más aún, del otro lado, es decir si $\mu_0 \leq \mu \leq \mu_0 + \delta$, esta propiedad no puede valer, es decir que existe al menos una raíz con parte real positiva.

Pero notar que $p_0(\mu) > 0$ para todo $\mu \in (\mu - \delta, \mu + \delta)$, así que en $\mu = \mu_0$ las raíces deben cruzar el eje imaginario con parte real distinta de 0. De aquí podemos deducir que estas raíces deben ser complejas no reales (y que por ende son al menos dos). Veamos que son exactamente dos. Para esto, usamos que tenemos una factorización como (2.18). ¿Podría q tener raíces con parte real positiva en $(\mu_0, \mu_0 + \delta)$? En tal caso, tendríamos que algún Q_i no podría ser positivo, y por exactamente los argumentos ya hechos, D_i tampoco lo sería.

Lo último que queda probar es la condición sobre la derivada de la parte real. En efecto, el polinomio q tiene todas sus raíces con parte real negativa por lo que usando exactamente los mismos argumentos con las matrices, llegamos a que $\partial_\mu \operatorname{Re}(\lambda(\mu_0)) = -1/2 \partial_\mu \alpha(\alpha_0) \neq 0$ por el Lema 2.2.2. \square

2.3. Introducción a los polítopos

En esta sección introduciremos nociones elementales de polítopos y programación lineal que serán de utilidad en la sección computacional. Nos basaremos en el texto de Ziegler [26].

Definición 2.3.1. Dado un conjunto finito $\{x_1, \dots, x_k\} \subseteq \mathbb{R}^n$, definimos su *cápsula convexa* como $\operatorname{Conv}(x_1, \dots, x_k) = \{\lambda_1 x_1 + \dots + \lambda_k x_k : \lambda_i \geq 0, \sum_{i=1}^k \lambda_i = 1\}$. Diremos que $P \subseteq \mathbb{R}^n$ es un *polítopo* si es la cápsula convexa de un conjunto finito.

Definimos el *cono poliedral* de la siguiente manera: $\operatorname{Cono}(x_1, \dots, x_k) = \{\lambda_1 x_1 + \dots + \lambda_k x_k : \lambda_i \geq 0\}$. A veces consideramos el cono trasladado, que se define para un dado $c \in \mathbb{R}^n$ como $\operatorname{Cono}_c(x_1, \dots, x_k) = \{\lambda_1 x_1 + \dots + \lambda_k x_k + c : \lambda_i \geq 0\}$.

Observemos que tanto la cápsula convexa como el cono poliedral de un conjunto de puntos son conjuntos convexos. De hecho, se puede ver fácilmente que $\text{Conv}(\{x_1, \dots, x_k\})$ es el conjunto convexo más chico que contiene a $\{x_1, \dots, x_k\}$. Por otro lado, un polígono siempre es compacto, ya que es imagen de una función continua sobre el compacto $\{\lambda : \lambda_1 + \dots + \lambda_n = 1\}$, pero los conos claramente no.

Ahora enunciaremos una caracterización de estos conjuntos, es decir que pueden ser identificados como intersecciones de semi-hiperplanos en \mathbb{R}^n .

Definición 2.3.2. Sean $x, y \in \mathbb{R}^n$. Decimos que $x \leq y$ si y solo si $x_i \leq y_i$ para todo $1 \leq i \leq n$.

Teorema 2.3.1 (Teorema Principal). Sea $P \subseteq \mathbb{R}^n$. Luego:

- Si P es acotado, entonces P es la cápsula de un conjunto convexo finito si y solo si existe una matriz $A \in \mathbb{R}^{m \times n}$ y $z \in \mathbb{R}^m$ de modo que $P = \{x : Ax \leq z\}$.
- P es un cono si y solo si existe una matriz $A \in \mathbb{R}^{m \times n}$ de modo que $P = \{x : Ax \leq 0\}$.

De aquí tenemos varios corolarios. Se puede ver que intersecar polítopos da un polígono, y que intersecar un polígono con un subespacio afín da un polígono, ya que en ambos casos estamos agregando “más filas” al sistema de desigualdades.

Definición 2.3.3. (Caras y Vértices) Sea $P \subseteq \mathbb{R}^n$. Si una inecuación $cx \leq c_0$ es válida para todo $x \in P$, entonces decimos que $F = P \cap \{cx = c_0\}$ es una *cara* de P . En caso de F ser un solo punto, decimos que F es un *vértice*.

Proposición 2.3.1. Sea $P \subseteq \mathbb{R}^n$ un polígono y V su conjunto de vértices. Luego:

- $P = \text{Conv}(V)$
- Si $P = \text{Conv}(X)$ para algún conjunto finito X , entonces $V \subseteq X$.

La demostración de este resultado utiliza Lemas de Farkas para caracterizar los puntos de la cápsula convexa. Fundamentalmente, este resultado nos dice que siempre podemos obtener un sistema minimal de generadores (los vértices) a partir de cualquier generador de la cápsula convexa. Pero la inclusión del segundo ítem efectivamente puede ser estricta, es sencillo verificar que $\text{Conv}((1, 0)(0, 0)(\frac{1}{2}, 0))$ nos devuelve el segmento de vértices $(0, 0)$ y $(1, 0)$.

Definición 2.3.4. Sea $P \subseteq \mathbb{R}^n$ un polígono y $c \in \mathbb{R}^n$. Definimos el *cono normal* de P en c como:

$$N_c(P) = \{y \in \mathbb{R}^n : y(c - x) \leq 0 \text{ para todo } x \in P\}$$

Geométricamente, el cono normal devuelve todos los vectores que forman un ángulo obtuso o recto con el vector $c - x$ para todo $x \in P$. A nosotros nos interesará el caso en el que c es un vértice del cono. El problema de la definición es que podría incluir infinitas desigualdades. Pero veremos que se puede pasar fácilmente al caso finito.

Proposición 2.3.2. Sea $P \subseteq \mathbb{R}^n$ un polígono y V sus vértices. Luego

$$N_c(P) = \{y \in \mathbb{R}^n : y(c - v) \leq 0 \text{ para todo } v \in V\}$$

Demostración. Una de las inclusiones es evidente. Tomemos y de modo que $y(c - v) \leq 0$ para todo $v \in V$. Sea $x \in P$, ya sabemos luego que como $P = \text{Conv}(V)$, existen $\lambda_1, \dots, \lambda_k$ no negativos de modo que $x = \lambda_1 v_1 + \dots + \lambda_k v_k$ y que $\sum_{i=1}^n \lambda_i = 1$. Entonces:

$$y(c - x) = y(c - \sum_{i=1}^n \lambda_i v_i) = \sum_{i=1}^n y(\lambda_i c - \lambda_i v_i) = \sum_{i=1}^n \lambda_i y(c - v_i) \leq 0$$

y así se verifica que $y \in N_c(P)$. \square

Esto nos da el siguiente corolario.

Corolario 2.3.1. Si P es un polígono, $P = \text{Conv}(X)$ con $X \subseteq \mathbb{R}^n$ finito. Entonces $N_c(P)$ es convexo y cerrado.

Demostración. Como $P = \text{Conv}(V)$ y V es finito (ya que X es finito y $V \subseteq X$), $N_c(P)$ es cerrado al ser preimagen de un cerrado por una función continua. Para ver que es convexo, basta con notar que es un cono. \square

Veamos ahora cómo podemos calcular los interiores de conos. El siguiente resultado también tiene un análogo para polítopos.

Proposición 2.3.3. Sea $P \subseteq \mathbb{R}^n$ un cono que pasa por el origen. Luego $P = \{x : Ax \leq 0\}$. Entonces el interior de P es $P^\circ = \{x : Ax < 0\}$.

Demostración. Si $Ax < 0$, usando que $Q = \{y : (Ay)_i = 0 \text{ para alguna coordenada } i\}$ es un cerrado, sabemos que existe ε positivo de modo que $B_\varepsilon(x) \cap Q = \emptyset$. En particular, $B_\varepsilon(x) \subseteq P$, ya que si no para alguna coordenada $(Az)_i > 0$ y por Bolzano algún punto del segmento de x a z tendría dicha coordenada nula.

Finalmente, notar que si $(Ax)_i = 0$ para alguna coordenada, tomando vectores de la forma $x + \text{signo}(a_{i1})\varepsilon e_1$, tenemos que $A(x + \text{signo}(a_{i1})\varepsilon e_i) > 0$, pero $\|x - x - \varepsilon e_1\| = \varepsilon$, así que toda bola se sale de P . \square

Finalizamos esta sección dando una introducción a la programación lineal.

Definición 2.3.5. Sea $\varphi \in (\mathbb{R}^n)^*$ (es decir, un funcional lineal de \mathbb{R}^n), $A \in \mathbb{R}^{m \times n}$ y $b \in \mathbb{R}^m$. Un problema II de programación lineal consiste en hallar $z \in \mathbb{R}^n$ que verifica:

- $Az \leq b$ (es decir, z verifica las restricciones dadas).
- Si $Aw \leq b$ entonces $\varphi(w) \leq \varphi(z)$ (es decir, z maximiza φ respecto a las restricciones dadas).

En caso de existir tal z , diremos que el problema es factible.

Estos problemas son extremadamente comunes y muchísimo se ha estudiado al respecto [14]. Múltiples algoritmos existen que resuelven este problema (el método Simplex el más conocido). El problema de factibilidad, es decir que $Az \leq b$ sea no vacío, es el que a nosotros nos interesa. No obstante, como nos interesa el interior de los conos, tendremos que trabajar con desigualdades estrictas. Pero lo que podemos hacer es lo siguiente:

Proposición 2.3.4. Sea $x \in P^\circ$, y $P = \{z : Az \leq b\}$. Entonces existe ε para el cual $Ax \leq b - \varepsilon(1, \dots, 1)$. En particular, P° es no vacío si y solo $Ax \leq b$ es no vacío y existe un ε positivo tal que $Ax \leq b - \varepsilon(1, \dots, 1)$ es no vacío.

Por ende, lo que podemos hacer para determinar si el interior de un cono es vacío, es primero verificar que $Ax \leq b$ sea no vacío vía algún resolvente de programación lineal (esto no es estrictamente necesario, pero servirá sacarnos de encima los casos que no están en el borde), y luego exhibir un ε para el cual $Ax \leq b - \varepsilon$ sea factible. Optimizando cualquier funcional lineal, obtendremos así un punto en el interior del cono. No obstante, vamos a probar una versión más fuerte para conos, mostrando que podemos tomar $\varepsilon = 1$.

Proposición 2.3.5. Sea el cono $C = \{z : Az \leq 0\}$. Entonces $C^\circ \neq \emptyset$ si y solo si existe $y \in \mathbb{R}^n$ tal que $Ay \leq (-1, \dots, -1)$

Demostración. La vuelta es inmediata de la proposición anterior. Sea entonces $x \in C^\circ$. Por lo visto antes, existe $\varepsilon > 0$ tal que $Ax \leq -\varepsilon(1, \dots, 1)$. Notar que $\lambda x \in C$ para todo $\lambda > 0$, por lo que escogiendo $\lambda = 1/\varepsilon$, obtenemos $y = \lambda x$ que cumple que $Ay = \lambda Ax \leq (-1, \dots, -1)$. \square

2.3.1. Polítopos de Newton

En esta sección, presentaremos una herramienta útil para acotar la cantidad de ceros aislados de un sistema polinomial.

Definición 2.3.6 (Polítopo de Newton asociado a un polinomio.). Sea $f \in \mathbb{R}[x_1, \dots, x_s]$ escrito de la forma $f = a_1x^{\sigma_1} + \dots + a_\ell x^{\sigma_\ell}$, donde $\sigma_i \in \mathbb{Z}^s$ son distintos y $a_i \neq 0$. Definimos el *polítopo de Netwon asociado a f* , $\text{Newt}(f)$, como la cápsula convexa generada por $\{\sigma_1, \dots, \sigma_\ell\} \subseteq \mathbb{R}^s$.

Por ejemplo, si $f(x, y) = x^2y + xy^2 + xy$, el polítopo de Netwon asociado a f será la cápsula convexa de $\{(2, 1)(1, 2)(1, 1)\}$, es decir el triángulo con esos vértices.

Definición 2.3.7 (Volumen de Minkowski. Volumen Mixto.). Sean $P_1, \dots, P_s \subseteq \mathbb{R}^s$ polítopos. El volumen del conjunto $\lambda_1 P_1 + \dots + \lambda_s P_s$ es polinomial y homogéneo de grado s en $\lambda_1, \dots, \lambda_s$. Definimos el *volumen mixto* de $\{P_1, \dots, P_s\}$ como el coeficiente asociado al monomio $\lambda_1 \dots \lambda_s$.

Trataremos de dar una idea de por qué este conjunto es un polinomio homogéneo basándonos en [9]. Notar que al trabajar con polítopos de Newton, nuestras coordenadas son enteras. La idea es notar que para el caso en que $s = 1$ es sencillo al estar trabajando con intervalos en \mathbb{R} y que el volumen de un polítopo se puede relacionar con el volumen de sus caras maximales mediante

$$\text{Vol}(P) = \frac{1}{n} \sum_F a_F \frac{\text{Vol}(F)}{\|v_f\|}$$

donde $v_f \in \mathbb{Z}^s$ es la normal interior entera primitiva (que se puede demostrar que existe y es única) y a_F de modo tal que $\{z : zv_f = -a_f\}$ es el hiperplano que contiene a F . La prueba luego restringe los valores de λ_i a valores positivos para verificar que la suma de Minkowski

siempre tiene la misma normal interior de modo que puede usar la fórmula de arriba y bajar la dimensión.

Enunciaremos ahora una fórmula para computar el volumen mixto.

Proposición 2.3.6. *Si $MV(P_1, \dots, P_s)$ es el volumen mixto de P_1, \dots, P_s entonces:*

$$MV(P_1, \dots, P_s) = \sum_{k=1}^n (-1)^{n-k} \sum_{I \subseteq \{1, \dots, n\}, \#I=k} \text{Vol} \left(\sum_{i \in I} P_i \right).$$

Por ejemplo, si tenemos tres polítopos P_1, P_2, P_3 el volumen mixto es:

$$MV(P_1, P_2, P_3) = \text{Vol}(P_1) + \text{Vol}(P_2) + \text{Vol}(P_3) - \text{Vol}(P_1 + P_2) - \text{Vol}(P_1 + P_3) - \text{Vol}(P_2 + P_3) + \text{Vol}(P_1 + P_2 + P_3)$$

Nuestra idea es usar la noción de volumen mixto para dar una generalización del Teorema de Bezout. Notar que, por ejemplo, dados dos polinomios de grado 5, sabemos el teorema de Bezout nos da una cota de 25 intersecciones en plano proyectivo complejo. Se puede verificar que en el caso en el que f_1, \dots, f_s sean polinomios *completos* de grados d_1, \dots, d_s respectivamente (es decir, todos los coeficientes de grado total j de f_i sean no nulos para todo $j \leq d_i$) vale que

$$MV(\text{Newt}(f_1), \dots, \text{Newt}(f_s)) = d_1 \dots d_s.$$

Por ende, el volumen mixto da igual a la cota del teorema de Bezout.

No obstante, este número podría dar menor si no pedimos que los polinomios sean completos, es decir, que los polítopos tengan menos vértices. Y esto efectivamente sigue siendo una cota para la cantidad de ceros no aislados de un sistema polinomial. Es decir

Teorema 2.3.2 (Teorema de Bernstein.). *Sean $g_1, \dots, g_s \in \mathbb{C}[x_1, \dots, x_s]$. Luego la cantidad de soluciones aisladas en \mathbb{C}^* contadas con multiplicidad de $g_1(x) = \dots = g_s(x) = 0$ es menor o igual al volumen mixto de $\{\text{Newt}(g_1), \dots, \text{Newt}(g_s)\}$.*

Otra aplicación útil de los polítopos de Newton es para encontrar puntos donde el polinomio tenga un signo. Efectivamente, llevamos el problema a una variable buscando puntos de la forma $t^\alpha = (t^{\alpha_1}, \dots, t^{\alpha_n})$ donde $f(t^\alpha)$ sea positivo o negativo. La idea es buscar coeficientes positivos o negativos de la función y buscar exponentes α de modo que ese monomio domine a la expresión. Primero incorporamos algunas definiciones.

Definición 2.3.8. Sea $f \in \mathbb{R}[x_1, \dots, x_s]$.

- Si $f = \sum_{i=1}^s b_i x^{\sigma_i} \in \mathbb{R}[x]$, decimos que σ_i es un vértice positivo (respectivamente negativo) de $\text{Newt}(f)$ si $b_i > 0$ (resp. $b_i < 0$).
- Sea σ un vértice de $\text{Newt}(f)$, definimos el *cono normal exterior* a $\text{Newt}(f)$ en σ como el cono generado por los vectores normales exteriores a los hiperplanos correspondientes a las caras maximales que contienen a σ , y lo denotamos $N_f(\sigma)$.

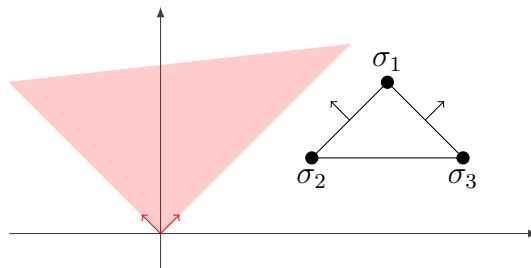


Figura 2.3: Si $f(x) = b_1x_1^3x_2^2 + b_2x_1^2x_2 + b_3x_1^4x_2$, tenemos $\sigma_1 = (3, 2)$, $\sigma_2 = (2, 1)$, $\sigma_3 = (4, 1)$ y representamos $\text{Newt}(f)$ junto con $N_f(\sigma_1)$ (en rojo).

Notemos que $N_f(\sigma) = \{y \in \mathbb{R}^n : y(\sigma - \tau) \geq 0 \text{ para todo } \tau \in \text{Newt}(f)\}$, es decir, es análogo a la noción de cono normal salvo por un signo. Ver, por ejemplo la Figura 2.3

Proposición 2.3.7. Sea $f \in \mathbb{R}[x_1, \dots, x_s]$ y σ_i un vértice positivo (respectivamente negativo) de $\text{Newt}(f)$. Luego existe un $x^* \in \mathbb{R}_{>0}^s$ tal que $f(x^*) > 0$ (resp. $f(x^*) < 0$).

Demostración. Sea $w = (w_1, \dots, w_s)$ en $N_f(\sigma_i)^\circ$, es decir, en el interior del cono normal exterior a $\text{Newt}(f)$ con vértice σ_i .

Consideramos $\varphi(v) = \langle v, w \rangle$. Por definición de cono normal exterior, sabemos que $\varphi(\sigma_i) > \varphi(\sigma_j)$ si $\sigma_i \neq \sigma_j$. Luego, si consideramos la función para $t > 0$:

$$f(t^{w_1}, \dots, t^{w_s}) = b_1 t^{\varphi(\sigma_1)} + \dots + b_s t^{\varphi(\sigma_s)},$$

tenemos que el signo de f coincide con el signo de b_i si t es suficientemente grande. Luego basta tomar $x^* = (t^{w_1}, \dots, t^{w_s})$. \square

2.4. Resultantes

En esta sección introducimos la noción de resultante entre dos polinomios, que será de utilidad para acotar la cantidad de estados estacionarios de ERK. Nos basaremos en el libro [8].

Supongamos que $f, g \in \mathbb{C}[x]$ comparten un factor común. Luego $f = hf_1$, $g = hg_1$ entonces tenemos que $(-g_1)f + f_1g = 0$. Es decir, tenemos dos polinomios A, B de grado menor que g, f respectivamente de modo que $Af + Bg = 0$.

Si f, g no comparten factores en común, por Bezout, existen C, D tales que $Cf + gD = 1$. Supongamos que existieran A, B de grados menores a g, f respectivamente de modo que $Af + Bg = 0$. En tal caso, como $Bg = -Af$ vale que

$$B = B(Cf + Dg) = CBf + DBg = (CB - AD)f$$

lo que contradice que el grado de B sea menor que el de f . Por ende tenemos una caracterización para cuándo dos polinomios comparten un factor común:

Proposición 2.4.1. Sean $f, g \in \mathbb{C}[x]$. Luego f, g tienen un factor común no trivial si y solo si existen $A, B \in \mathbb{C}[x]$ de grados menores que g, f respectivamente de modo que $Ag + Bf = 0$.

Buscamos ahora A, B . Si $A = u_0x^{m-1} + \dots + u_{m-1}$ y $B = v_0x^{\ell-1} + \dots + v_{\ell-1}$, y $f = c_0x^\ell + \dots + c_\ell$, $g = d_0x^m + \dots + d_m$, y $Af + Bg = 0$, entonces obtenemos un sistema lineal a partir de la matriz de Sylvester de tamaño $(m + \ell) \times (m + \ell)$:

$$\text{Syl}(f, g, x) = \begin{pmatrix} c_0 & & & d_0 & & \\ c_1 & \ddots & & \vdots & \ddots & \\ \vdots & & & & & d_0 \\ & & c_0 & d_{m-1} & & \\ \vdots & & c_1 & d_m & & \\ c_\ell & & \vdots & & \ddots & \\ & \ddots & & & & \\ & & c_\ell & & & d_m \end{pmatrix}$$

$\underbrace{\hspace{10em}}_{m \text{ columnas}} \quad \underbrace{\hspace{10em}}_{\ell \text{ columnas}}$

Observar que el sistema tiene solución no trivial si y solo si el determinante de $\text{Syl}(f, g, x)$ da 0. Definimos entonces $\text{Res}(f, g)(x) = \det(\text{Syl}(f, g, x))$ y tenemos el siguiente resultado.

Teorema 2.4.1. Dados $f, g \in \mathbb{C}[x]$ no nulos, f, g tienen un factor común no trivial si y solo si $\text{Res}(f, g, x) = 0$.

Una propiedad útil es que podemos escribir a la resultante como combinación lineal entera de los polinomios. Es decir,

Proposición 2.4.2. Dados $f, g \in \mathbb{C}[x]$ no constantes. Entonces $\text{Res}(f, g) = Af + Bg$ con $A, B \in \mathbb{C}[x]$ o equivalentemente, $\text{Res}(f, g) \in \langle f, g \rangle$. Más aún, A, B son enteros en los coeficientes de f, g (es decir, si $f, g \in \mathcal{A}[x]$ con $\mathcal{A} \subseteq \mathbb{C}$ anillo entonces $A, B \in \mathcal{A}[x]$).

La demostración del resultado se basa en la regla de Crámer, notando que el caso en el que f, g comparten un factor común es trivial, por lo que basta estudiar el caso en el que son coprimos. Para esto, se estudia el sistema $\hat{A}f + \hat{B}g = 1$ que también utiliza la matriz de Sylvester pero buscando solución para $b = (0, \dots, 0, 1)$.

Recordando la regla de Cramer, obtenemos una fórmula de la forma $\hat{u}_i = \frac{1}{\det(\text{Syl}(f, g))} \det(M)$ donde M proviene de remplazar la columna correspondiente de la matriz con b . Así tenemos que cada \hat{u}_i es el cociente de un polinomio entero en los elementos de la matriz de Sylvester y la resultante $\text{Res}(f, g)$. Luego $A = \hat{A}\text{Res}(f, g)$ y $B = \hat{B}\text{Res}(f, g)$ cumplen lo pedido.

Acabamos esta sección dando una introducción muy elemental a la teoría de eliminación basandonos en [8].

Definición 2.4.1. Sea $I = \langle f_1, \dots, f_s \rangle$ el ideal generado por los polinomios $f_1, \dots, f_s \in k[x_1, \dots, x_s]$.

- Definimos el primer ideal de eliminación de $I_1 = I \cap k[x_2, \dots, x_s]$.

- La variedad asociada a I consiste en $V(I) = \{x \in \mathbb{C}^n : f_i(x) = 0 \text{ si } 1 \leq i \leq s\}$.
- El ideal asociado a un conjunto $S \subseteq \mathbb{C}^n$ se define como $I(S) = \{f \in \mathbb{C}[x_1, \dots, x_n] : f(s) = 0 \text{ para todo } s \in S\}$.
- Definimos la clausura Zariski de un conjunto $S \subseteq \mathbb{C}^n$ como

$$\overline{S} = \bigcap_{f \in I(S)} f^{-1}(0).$$

Este es el conjunto más chico que contiene a S y que es anulado por un conjunto de polinomios, es decir, la variedad afín más chica que contiene a S . En particular $V(I(S)) = \overline{S}$.

Finalizamos la sección con un teorema que nos permite relacionar la proyección de una variedad con el primer ideal de eliminación.

Teorema 2.4.2 (Teorema de la Clausura). *Sea $\pi : \mathbb{C}^n \rightarrow \mathbb{C}^{n-1}$ la proyección a las últimas $n - 1$ coordenadas. Entonces $\overline{\pi(V)} = V(I_1)$.*

Demostración. Veamos que $V(I_1) = V(I(\pi(V)))$. Que $V(I_1) \supseteq V(I(\pi(V)))$ es evidente.

Sea $f \in I(\pi(V))$, es decir, $f(a_2, \dots, a_n) = 0$ para todo $(a_2, \dots, a_n) \in \pi(V)$. Claramente tenemos que $f(a_1, \dots, a_n) = 0$ para todo (a_1, \dots, a_n) al pensar a f como elemento de $\mathbb{C}[x_1, \dots, x_n]$. Por el Nullstellensatz, $f^N \in \langle f_1, \dots, f_s \rangle$. Como f no depende de x_1 , f^N tampoco. En particular $f^N \in \sqrt{I_1}$. Finalmente, $V(I_1) = V(\sqrt{I_1}) \subseteq V(I(\pi(V)))$. \square

Capítulo 3

Estudio de la Dinámica de ERK.

En este capítulo estudiaremos la dinámica de la red ERK a partir de los resultados principales de los trabajos Conradi et al. (2021) [5] y Obatake et al.(2019) [20].

3.1. Biestabilidad

Recordemos las Preguntas 1.2.1 y 1.2.2, que hacían referencia a la biestabilidad de la red ERK y de la mínima red ERK biestable. El siguiente resultado nos dirá que, efectivamente, la mínima red ERK biestable es biestable y luego, por el Teorema 2.1.4 podremos concluir que la red ERK es biestable. Cabe notar que esto nos indica que van a existir constantes de reacción y alguna clase de compatibilidad estequiométrica (para cada sistema) donde hayan dos estados estacionarios exponencialmente estables. Las preguntas apuntan a si es posible que las constantes de reacción que permiten biestabilidad en alguna clase cumplen tener p_k y p_ℓ tan cercanos a 1 como se desee.

Teorema 3.1.1. *Sea G una subred de ERK irreversible (ver Definición 1.2.1). Luego son equivalentes:*

1. G es multiestacionaria.
2. G es biestable.
3. G contiene alguna de las reacciones con constantes k_{on} y ℓ_{on} .

Demostración. Es claro que $2 \Rightarrow 1$, así que probaremos $1 \Rightarrow 3 \Rightarrow 2$.

Veamos $1 \Rightarrow 3$. Supongamos que G no contiene ninguna de las reacciones asociadas a k_{on} y ℓ_{on} . Consideramos la función crítica $C(\kappa, x)$ hallada para esta red como en (1.5), pero tomando $k_{on} = \ell_{on} = 0$. Hicimos la cuenta (ver 4.2) como los autores y obtuvimos el siguiente denominador: $(k_{cat}\ell_{cat}x_1x_{10}x_{12}x_2x_3x_5x_9)$ El denominador es positivo (recordemos que siempre asumimos que nuestras constantes de reacción son positivas), por lo que basta con estudiar el signo del numerador. También podemos ver que este es un polinomio de grado 5 y que todos

sus coeficientes son negativos. Debido a que asumimos que $x \geq 0$, deducimos que el signo es negativo, es decir, para todo $x \geq 0$,

$$\text{signo}(C(\kappa, x)) = -1 = (-1)^9$$

que, como $9 = \dim(\mathcal{S})$, usando que G no presenta estados estacionarios en el borde y que es conservativa, tenemos que nuestro sistema es monoestacionario.

Veamos ahora $3 \Rightarrow 2$. En primer lugar, por un argumento de simetría (cambiando $E \leftrightarrow F$ y $S_{00} \leftrightarrow S_{11}$ y $S_{01} \leftrightarrow S_{01}$) podemos suponer que $k_{on} \neq 0$. Sea G' la red que se obtiene de remover todas las otras reacciones “azules” (ver Figura 1.2). En tal caso, en la clase $(c_1, c_2, c_3) = (46, 13, 13)$ y vector de reacciones:

$$(k_1, k_3, k_{cat}, k_{on}, k_{off}, \ell_1, \ell_3, \ell_{cat}, \ell_{off}, m_2, m_3, n_1, n_3) = (2, 1, 1, 1, 5, 15, 2, 1, 1, 1, 10, 20, 10, 20, 10)$$

los autores encuentran dos estados estables, que nosotros verificamos computacionalmente (ver el capítulo de programas, sección 4.2). Por el resultado anterior, como G proviene de hacer reversibles algunas de las reacciones de G' y este es biestable, por el resultado 2.1.4, tenemos que G es biestable. \square

Además, por el Teorema 2.1.4, deducimos el siguiente resultado.

Teorema 3.1.2. *La red ERK completa es biestable.*

Con esto tenemos un estudio bastante completo acerca de la biestabilidad al estudiar reversibilidad. ¿Qué sucede luego al estudiar distintos niveles de procesatividad? Veremos que la red mínima biestable de ERK es multiestacionaria para cualquier nivel de procesatividad.

Teorema 3.1.3. *Consideremos la red ERK mínima biestable de la Figura 1.2. Para cualesquiera $p_k \in (0, 1)$ y $p_\ell \in (0, 1)$ existe un vector de constantes de reacción*

$$\kappa^* = (k_1^*, k_3^*, k_{cat}^*, k_{on}^*, k_{off}^*, \ell_1^*, \ell_3^*, \ell_{cat}^*, \ell_{off}^*, m_2^*, m_3^*, n_1^*, n_3^*)$$

de modo que:

- $p_k = \frac{k_{cat}^*}{k_{cat}^* + k_{off}^*}$ y $p_\ell = \frac{\ell_{cat}^*}{\ell_{cat}^* + \ell_{off}^*}$
- *el sistema asociado es multiestacionario, es decir, presenta más de un estado estacionario positivo en alguna clase de compatibilidad estequiométrica.*

Demostración. Para la parametrización que mostramos en (1.10), consideramos la función crítica $C(\kappa, x_1, x_2, x_3) = C(\kappa, \hat{x})$. Tomemos $\ell_{off}^* = k_{off}^* = 1$ y dejemos ℓ_{cat}^*, k_{cat}^* positivos libres, así obtendríamos todos los posibles valores de $p_k, p_\ell \in (0, 1)$ (pues la imagen de $f(x) = x/(x+1)$ cuando $x > 0$ es $(0, 1)$).

Como vimos en §1.2.2, $\dim(\mathcal{S}) = \text{rango}(N) = 9$ (con 3 leyes de conservación), luego $(-1)^{\text{rango}(N)+1} = 1$. Los autores calcularon para $k_{off} = 1 = \ell_{off}$ y obtuvieron entonces el denominador de nuestra función crítica:

$$(k_{cat}k_{on}x_2 + k_{cat}n_1x_3 + n_1x_3)^2\ell_{cat}x_3$$

que es siempre positivo. Por ende, basta buscar constantes de modo que el numerador sea positivo. Dejamos libres k_{cat}, ℓ_{cat} y tomamos para $t > 0$, $x_1 = x_2 = t$, $k_1 = k_3 = \ell_3 = t^{-1}$ y el resto de las constantes igualadas a 1. El numerador, por otro lado, es de la forma:

$$(2k_{cat}\ell_{cat}^2 + 2k_{cat}^2\ell_{cat})t^5 + p(t),$$

donde $p(t)$ es un polinomio de grado a lo sumo 4. Como el coeficiente principal es positivo (recordar que asumimos $\ell_{cat} > 0$), tomando t suficientemente grande, encontramos un valor para el cual la función crítica es positiva, como queríamos. \square

Para encontrar la parametrización en la demostración, la idea es usar la propiedad de polítopos de Newton 2.3.7 para encontrar un vector de exponentes sobre el cual la función crítica como polinomios con coeficientes en k_{on} , k_{cat} y ℓ_{cat} es eventualmente positiva. Una idea similar está en el capítulo 4 pero más general, y una aplicación de estas ideas en la subsección 4.1.2 del mismo capítulo.

De aquí surge la pregunta, ¿para todos niveles procesativos la red presenta biestabilidad? Esta pregunta se encuentra abierta.

Conjetura 3.1.1. *Bajo las hipótesis del teorema anterior, para cualesquiera $p_k \in (0, 1)$, $p_\ell \in (0, 1)$ existe un vector de constantes de reacción κ^* para el cual $p_k = \frac{k_{cat}^*}{k_{cat}^* + k_{off}^*}$, $p_\ell = \frac{\ell_{cat}^*}{\ell_{cat}^* + \ell_{off}^*}$ y el sistema asociado es **biestable**.*

Veremos que el resultado es cierto para algunos valores de procesatividad. En el trabajo [5] también se realiza el mismo análisis para niveles de procesatividad encontrados al azar.

Proposición 3.1.1 (Biestabilidad para varios niveles de procesatividad). *Dados*

$$p_k = p_\ell \in \{0,1; 0,2; \dots; 0,9; 0,91; \dots; 0,99\}$$

Existe un vector κ^ para el cual el sistema de la red mínima biestable tiene múltiples estados estacionarios exponencialmente estables en alguna clase de compatibilidad y que además $p_k =$*

$$\frac{k_{cat}^*}{k_{cat}^* + k_{off}^*} \text{ y } p_\ell = \frac{\ell_{cat}^*}{\ell_{cat}^* + \ell_{off}^*}.$$

Para esta demostración recordemos la del Teorema 3.1.3, en la que la función crítica, al evaluar en κ_t obtenemos un polinomio de grado 5 en t con coeficiente principal positivo. Por ende, lo que se hace numéricamente es encontrar un valor de t para el cual dicho polinomio resulte positivo y luego analizar el signo de la parte real de los autovalores de la matriz jacobiana.

3.2. Bifurcaciones de Hopf y Oscilaciones.

En la siguiente sección buscaremos estudiar la existencia de bifurcaciones de Hopf y oscilaciones para la red ERK **reducida**. Estudiamos esta en lugar de la red completa o la mínima biestable por cuestiones de complejidad computacional.

Un primer resultado de [5] nos indica que para niveles de procesatividad cercanos a 1, podemos tener bifurcaciones de Hopf. De hecho, el resultado lo demuestra para casi todos los posibles valores de procesatividad.

Teorema 3.2.1. *Sea ε tal que $0,002295 < \varepsilon < 1$. Entonces la red ERK reducida presenta bifurcaciones de Hopf para cualesquiera niveles de procesatividad $p_k, p_\ell > \varepsilon$. Más formalmente, existe un vector $\kappa = (k_1, k_3, k_{cat}, k_{off}, m, \ell_1, \ell_3, \ell_{cat}, \ell_{off})$. De modo que:*

- $p_k = \frac{k_{cat}}{k_{cat} + k_{off}} > \varepsilon$ y $p_\ell = \frac{\ell_{cat}}{\ell_{cat} + \ell_{off}} > \varepsilon$;
- el sistema asociado tiene una bifurcación de Hopf con respecto a k_{cat} .

Demostración. En primer lugar vamos a tomar las siguientes restricciones para el vector de constantes κ :

- $k_{cat} > \frac{\varepsilon}{1 - \varepsilon} > \frac{0,002295}{1 - 0,002295} \approx 0,0023$.
- $k_{off} = \ell_{off} = 1$.
- Sea $t > 1$, $\ell_{cat} = t^2 k_{cat}$.

De esta manera, tenemos que $\varepsilon < \frac{k_{cat}}{k_{cat} + k_{off}} < \frac{\ell_{cat}}{\ell_{cat} + \ell_{off}}$.

Verificaremos la existencia de bifurcaciones estudiando h_4, h_5, h_6 . Los autores verifican simbólicamente que h_4 es suma de términos positivos para $\hat{\kappa} = (k_{cat}, k_{off}, k_{on})$.

Sea $(\hat{\kappa}; x) = (k_{cat}, 1, 1; 1, 1, t^2, 1, t^2, 1/t, 1, t^2, 1)$. Es decir, $x_3 = x_5 = x_8 = t^2$, $x_6 = 1/t$ y k_{cat} libre (lo que es natural pues buscamos estudiar bifurcaciones con respecto a k_{cat}). Utilizando Mathematica, los autores verifican que si $k_{cat} > 0,0023$, entonces $h_5(\hat{\kappa}; x) > 0$. Evaluando en $t = 1$, se puede ver que tenemos un polinomio positivo si $k_{cat} > 0$. Pero, si $t \rightarrow \infty$, entonces $h_6(\hat{\kappa}; x) \rightarrow -\infty$. Por Bolzano, existe $t_0 > 1$ para el cual $h(\hat{\kappa}, x) = 0$, donde $x = (1, 1, t_0^2, 1, t_0^2, 1/t_0, 1, t_0^2, 1)$. Verificamos numéricamente que la derivada es no nula y así estamos en condiciones de aplicar el criterio de Routh-Hurwitz 2.2.1. \square

Análogamente a las demostraciones anteriores, la parametrización hallada en la demostración se basa en la Proposición 2.3.7. Exploraremos la misma pregunta y daremos un algoritmo general para este problema en el capítulo 4.

En la Figura 3.1 podemos observar la existencia de oscilaciones en la red ERK reducida a partir de simulaciones numéricas. El programa utilizado está presentado capítulo 4.

Una pregunta razonable luego es, ¿puede el sistema ser biestable y presentar bifurcaciones de Hopf en una misma clase de compatibilidad estequiométrica? Veamos una respuesta parcial a esta pregunta.

Teorema 3.2.2. *Sea (G, κ) un sistema de reacciones químicas conservativo y \mathcal{S}_c una clase de compatibilidad estequiométrica tal que tiene a lo sumo tres estados estacionarios, entonces \mathcal{S}_c no presenta bifurcaciones de Hopf y dos estados estacionarios estables simultáneamente.*

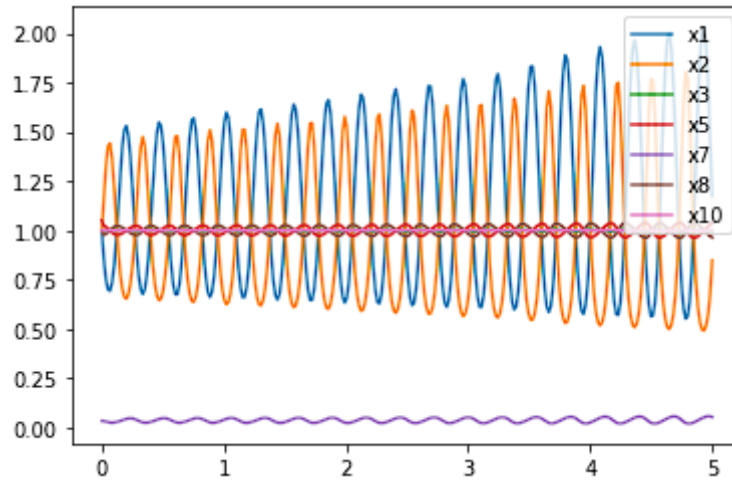


Figura 3.1: Observamos el comportamiento oscilatorio para la evaluación $k_{cat} = 3$ y $t = 28,7459$ de acuerdo con la demostración del Teorema 3.2.1 para las variables $x_1, x_2, x_3, x_5, x_7, x_{10}$. Tomamos como valor inicial x^* ligeramente modificado para observar las oscilaciones

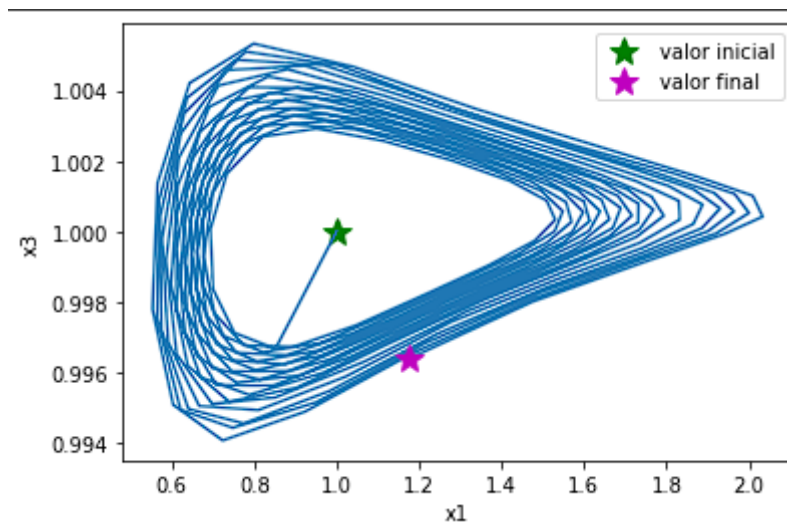


Figura 3.2: Vemos la oscilación en el plano x_1x_3 , bajo las mismas condiciones que en la Figura 3.1.

Demostración. Asumamos, sin pérdida de generalidad, que las especies están ordenadas de forma tal que podemos considerar una matriz de conservación de G, W , que se puede escribir en bloques como $W = (I_d \quad \widehat{W})$, donde $\widehat{W} \in \mathbb{R}^{d \times (s-d)}$ e I_d es la matriz identidad de tamaño d . Sea $f_{c,\kappa}$ el sistema ampliado como en (1.3). Sea $x^* \in \mathcal{S}_c$, y consideremos $p(\lambda) = \det(\lambda I - \text{Jac}f)|_{x=x^*}$ el polinomio característico de $\text{Jac}f$ evaluado en $x = x^*$. Como hay d relaciones de conservación, $\lambda = 0$ es una raíz de p de multiplicidad d . Consideremos entonces b_{s-d} el coeficiente de λ^d del polinomio p . Entonces:

- si en x^* se presenta una bifurcación de Hopf, por el Teorema 2.2.4 aplicado a $q(\lambda) = \frac{1}{\lambda^d} p(\lambda)$, tenemos que $b_{s-d} > 0$;
- si x^* es un estado estacionario estable, por el criterio de Routh-Hurwitz 2.2.1, tenemos $b_{s-d} > 0$;
- sabemos que $b_{s-d} = (-1)^{s-d} \sum_{|J|=s-d} \det(\text{Jac}f|_{x=x^*})_{J,J}$, es decir, salvo signo, la suma sobre los conjuntos de índices de cardinal $s - d$ del determinante de la matriz Jacobiana de f restringida a $x = x^*$ 2.1.5. Pero por la Proposición 5.3 de [25], tenemos entonces que $b_{s-d} = (-1)^{s-d} \det \text{Jac}f_{c,\kappa}|_{x=x^*}$.

Si el sistema tuviera una bifurcación de Hopf en $x^{(1)}$ y dos estados estacionarios estables $x^{(2)}$ y $x^{(3)}$ en \mathcal{S}_c (y no más estados estacionarios en \mathcal{S}_c por hipótesis), entonces el grado de Brouwer de $f_{c,\kappa}$ con respecto a \mathcal{S}_c es, por (2.1):

$$\sum_{i=1}^3 \text{signo}(\det \text{Jac}f_{c,\kappa}|_{x=x^{(i)}}) = 3(-1)^{(s-d)},$$

pero esto es una contradicción porque vimos en el Teorema 2.1.3 que este número debería ser ± 1 .

□

En la próxima sección estudiaremos la máxima cantidad de estados estacionarios positivos que puede tener nuestro sistema en una misma clase de compatibilidad, lo cual evidentemente nos puede servir para responder la pregunta anterior.

Otra pregunta que puede surgir es si en ERK pueden coexistir bifurcaciones de Hopf y biestabilidad pero no necesariamente en la misma clase de compatibilidad estequiométrica (en la original o en la mínima biestable, ya que la irreversible no presenta biestabilidad). En [20] se realizaron investigaciones numéricas pero no se encontró este suceso.

3.3. Máxima Cantidad de Estados Estacionarios

Tanto [5] como [20] presentan métodos para acotar la máxima cantidad de estados estacionarios.

Definición 3.3.1. Sea G una red conservativa con s especies, m reacciones y $W \in \mathbb{R}^{d \times s}$ matriz de conservación. Decimos que G **admite k estados estacionarios** si existe $\kappa \in \mathbb{R}_{>0}^m$ constantes de reacción tales que (G, κ) tiene k soluciones en alguna clase de compatibilidad estequiométrica.

Al mayor número que verifique esta propiedad lo denominamos **la máxima cantidad de estados estacionarios de G** .

En primer lugar, utilizaremos polítopos de Newton como en § 2.3.1.

Definición 3.3.2. Sea G una red conservativa con s especies y matriz de conservación W . Sea $c \in \mathbb{R}_{>0}^m$ y κ^* genérico. Definimos el volumen mixto de G como el volumen mixto de $\{\text{Newt}(f_{c,\kappa,1}), \text{Newt}(f_{c,\kappa,2}), \dots, \text{Newt}(f_{c,\kappa,s})\}$.

Luego, por el Teorema 2.3.2, tenemos que:

Proposición 3.3.1. *Sea G una red conservativa. La cantidad de estados estacionarios de G es menor o igual que el volumen mixto de G .*

Aplicamos este resultado a ERK.

Proposición 3.3.2. *Para las siguientes versiones de ERK (ver § 1.2), tenemos las siguientes cotas para la máxima cantidad de estados estacionarios.*

Red ERK	Máy estados positivos	Máy estados sobre \mathbb{C}^*	Volumen Mixto
Red Completa	≥ 3	7	7
Red con $k_{on} = 0$	≥ 3	5	5
Irreversible	1	3	3
Reducida	1	3	3

Demostración. Los resultados computacionales para calcular el volumen mixto se encuentran en [20]. Como el volumen mixto da una cota, basta probar las otras desigualdades. Para esto, para cada una de las redes, buscaremos vectores κ para los cuales encontramos esa cantidad de estados estacionarios.

Para ERK completa, [11] proporciona 7 estados estacionarios. Para el caso $k_{on} = 0$, consideramos $c = (1, 2, 3)$ y el vector:

$$(k_1, k_2, k_3, k_{cat}, k_{off}, k_{on}, \ell_1, \ell_2, \ell_3, \ell_{cat}, \ell_{on}, \ell_{off}, m_1, m_2, m_3, n_1, n_2, n_3) = (3, 25, 1, 5, 0, 6, 5, 23, 11, 13, 43, 41, 12, 7, 8, 12, 31, 21).$$

Se obtienen cinco estados estacionarios, 3 de los cuales son reales. Para la red ERK reducida, tomamos el vector:

$$(k_1, k_3, k_{cat}, k_{off}, m, n, \ell_1, \ell_3, \ell_{cat}, \ell_{off}) = (3, 4, 1, 5, 6, 8, 7, 11, 12, 5)$$

y se obtienen 3 estados estacionarios sobre \mathbb{C}^* , todos reales. Y para ERK irreducible tenemos:

$$(k_1, k_3, k_{cat}, k_{off}, \ell_1, \ell_3, \ell_{cat}, \ell_{off}, m_2, m_3, n_1, n_2) = (3, 1, 5, 6, 5, 11, 13, 41, 7, 8, 12, 21),$$

donde se obtienen 3 estados sobre \mathbb{C}^* , todos reales. \square

3.3.1. Método por Resultantes

Acabamos la sección utilizando argumentos estándar de resultantes para acotar la cantidad de estados estacionarios de la subred mínima biestable, siguiendo el razonamiento de [5].

En primer lugar, a partir de la parametrización de la Proposición 1.2.1, podemos hacer un despeje de la siguiente manera:

- Despejamos x_4, \dots, x_{12} en función de x_1, x_2, x_3 y las constantes a_i .
- Remplazamos lo obtenido en las leyes de conservación, excepto en la primera en la que hacemos la primera ecuación menos la segunda y la tercera.

Así obtenemos las siguientes expresiones:

$$c_1 - c_2 - c_3 = x_1 - x_2 - x_3 + \frac{a_5 a_9 a_{10} x_1 x_2}{a_8 x_2 + a_{13} x_3 + a_4 a_9 a_{13} x_3} + \quad (3.1)$$

$$+ \frac{a_5 a_7 a_{10} x_1 (a_8 x_2 + a_{13} x_3)}{a_1 a_{11} (a_8 x_2 + a_{13} x_3 + a_4 a_9 a_{13} x_3)} +$$

$$+ \frac{a_5 a_{10} x_1 x_2 (a_8 x_2 + a_2 a_7 a_8 x_2 + a_{13} x_3 + a_2 a_7 a_{13} x_3)}{a_1 a_3 a_{12} x_3 (a_8 x_2 + a_{13} x_3 + a_4 a_9 a_{13} x_3)},$$

$$c_2 = x_2 + \frac{a_5 a_{10} x_1 x_2 (a_8 x_2 + a_{13} x_3)}{a_8 x_2 + a_{13} x_3 + a_4 a_9 a_{13} x_3} + \frac{a_5 a_7 a_{10} x_1 (a_8 x_2 + a_{13} x_3)}{a_1 a_{11} (a_8 x_2 + a_{13} x_3 + a_4 a_9 a_{13} x_3)} + \quad (3.2)$$

$$+ a_{10} x_1 x_2,$$

$$c_3 = x_3 + \frac{a_5 a_{10} x_1 x_2 (a_8 x_2 + a_2 a_7 a_8 x_2 + a_{13} x_3 + a_2 a_7 a_{13} x_3)}{a_1 a_3 a_{12} x_3 (a_8 x_2 + a_{13} x_3 + a_4 a_9 a_{13} x_3)} + \quad (3.3)$$

$$+ \frac{a_5 a_{10} x_1 x_2 (a_8 x_2 + a_{13} x_3)}{a_1 (a_8 x_2 + a_{13} x_3 + a_4 a_9 a_{13} x_3)} + \frac{a_5 a_9 a_{10} a_{13} x_1 x_2 x_3}{a_8 x_2 + a_{13} x_3 + a_4 a_9 a_{13} x_3},$$

y de aquí tenemos que la máxima cantidad de estados estacionarios de la red ERK mínima biestable coincide con la máxima cantidad de soluciones positivas de este sistema racional. Multiplicando por los denominadores, obtenemos un sistema polinomial.

Simplificaremos nuestro problema para reducirlo a un problema de una variable. Primero, observamos que x_1 es lineal en las tres ecuaciones, por lo que podemos sacar factor común y obtener $x_1 = \gamma_1(x_2, x_3)$, $x_1 = \gamma_2(x_2, x_3)$, $x_1 = \gamma_3(x_2, x_3)$. Luego tomando $g_1 = \gamma_3 - \gamma_2$ y $g_2 = \gamma_1 - \gamma_2$, por esta construcción tenemos que:

Proposición 3.3.3. *Definiendo g_1, g_2, γ_1 como arriba, la máxima cantidad de soluciones del sistema $g_1 = g_2 = 0$ con $\gamma_1(x_2, x_3) > 0$ coincide con la máxima cantidad de estados estacionarios de la mínima subred biestable; donde $g_1, g_2, \gamma_1 \in \mathbb{Q}[a_1, \dots, a_{13}, c_1, c_2, c_3][x_2, x_3]$.*

Sea R la resultante de g_1, g_2 con respecto a la variable x_2 . Tenemos luego el siguiente resultado:

Proposición 3.3.4. *Sea R como ya definido. Fijemos $(a, c) \in \mathbb{R}_{>0}^{16}$. Supongamos que $R|_{(a,c)}$ tiene a lo sumo tres raíces en el intervalo $(0, \min\{c_1, c_3\})$ y para cada $x_3 > 0$ fijo la ecuación $g_1(x_2, x_3)|_{(a,c)} = 0$ tiene a lo sumo una única solución positiva. Entonces, el sistema dado por (3.1), (3.2), (3.3) admite a lo sumo tres soluciones positivas (x_1, x_2, x_3) .*

Demostración. Por la Proposición 2.4.2, sabemos que $R \in \langle g_1, g_2 \rangle \cap \mathbb{Q}[a_1, \dots, a_{13}, c_1, c_2, c_3, x_3]$.

Pero a su vez, por el Teorema 2.4.2 vale que si $\pi_{x_2} : \mathbb{C}^{18} \rightarrow \mathbb{C}^{17}$ dada por:

$$\pi_{x_2}(a_1, \dots, a_{13}, c_1, c_2, c_3, x_2, x_3) = (a_1, \dots, a_{13}, c_1, c_2, c_3, x_3)$$

Entonces:

$$\overline{\pi_{x_2}(V)} = V(\langle g_1, g_2 \rangle_{x_2}) = V(\langle g_1, g_2 \rangle \cap \mathbb{Q}[a_1, \dots, a_{13}, c_1, c_2, c_3, x_3]).$$

(Evidentemente, el orden de las variables es irrelevante en el Teorema 2.4.2).

Luego $\overline{\pi(V(g_1, g_2))} \subseteq V(R)$. Supongamos ahora que tenemos (a, c) fijo como en el enunciado. Como $R|_{(a,c)}$ tiene a lo sumo tres soluciones en el intervalo $(0, \min\{c_1, c_3\})$, claramente las soluciones de $g_1 = g_2 = 0$ tienen a lo sumo tres posibles valores para x_3 en ese intervalo. Para cada uno de estos valores de x_3 , $g_1(x_2, x_3)$ tiene una única solución como mucho, por lo que hay a lo sumo tres soluciones (x_2, x_3) con $x_3 < \min\{c_1, c_3\}$.

Pero recordando que todas nuestras variables son positivas y por la leyes de conservación, cualquier solución necesariamente debe verificar que $x_3 < c_3$ y por construcción de g_1, g_2 se puede verificar que $x_3 < c_1$. \square

Luego, enunciamos dos criterios para los cuales se puede verificar que implican la condición de esta proposición.

Corolario 3.3.1. *Cualquiera de las siguientes condiciones sobre los parámetros son suficientes para garantizar que el sistema (3.1),(3.2),(3.3) tiene a lo sumo tres soluciones positivas:*

- $a_i = 1$ para todo $i \neq 9$ (c_1, c_2, c_3, a_9 libres).
- a_9 y c_2 son suficientemente grandes, el resto de los valores de a_i son suficientemente grandes y $b > c_1/c_3 > 1$ y $c_2 > c_3 + 1$.

Ambos criterios se basan en el cálculo simbólico de R y la regla de signos de Descartes para acotar la cantidad de raíces positivas.

Capítulo 4

Programas

4.1. Método de Polítopos de Newton.

Enunciamos ahora el algoritmo de polítopo de Newton, un algoritmo teóricamente muy simple que permite hallar la raíz de un polinomio donde otro polinomio es positivo.

Proposición 4.1.1 (Método de Polítopos de Newton). *Sean $f, g \in \mathbb{R}[x_1, \dots, x_s]$. Supongamos que α es un vértice positivo de $\text{Newt}(f)$ y que β_+ y β_- son vértices positivos y negativos de $\text{Newt}(g)$ respectivamente.*

Sean $N_f(\alpha)$, $N_g(\beta_+)$, $N_g(\beta_-)$ los conos normales a los vértices como en la Definición 2.3.8. Si tenemos que $(N_f(\alpha))^\circ \cap (N_g(\beta_+))^\circ \neq \emptyset \neq (N_f(\alpha))^\circ \cap (N_g(\beta_-))^\circ$, entonces existe un $x^ \in \mathbb{R}_{>0}^s$ de modo que $f(x^*) > 0$ y $g(x^*) = 0$.*

Demostración. Nuestra demostración será constructiva, daremos un algoritmo para hallar x^* . Dado un vector $w = (w_1, \dots, w_s)$, denotamos $f_w(t) = f(t^{w_1}, \dots, t^{w_s})$.

PROC: Método de polítopo de Newton

INPUT: $f, g, \alpha, \beta_+, \beta_-$ como en el enunciado

OUTPUT: $x^* \in \mathbb{R}_{>0}^s$ de modo que $f(x^*) > 0$ y $g(x^*) = 0$.

1. Obtener $z \in (N_f(\alpha))^\circ \cap (N_g(\beta_+))^\circ$ y $w \in (N_f(\alpha))^\circ \cap (N_g(\beta_-))^\circ$
2. Construir f_z, g_z y g_w, f_w .
3. Obtener $\tau_z = \inf\{t^* > 0 : f_z(t) > 0 \text{ y } g_z(t) > 0 \text{ si } t > t^*\}$ y $\tau_w = \inf\{t^* > 0 : f_w(t) > 0 \text{ y } g_w(t) < 0 \text{ si } t > t^*\}$. Definir $T = \max\{\tau_z, \tau_w\} + 1$.
4. Definir $h(r) = f_{rz+(1-r)w}(T)$.
5. **mientras** $\min\{h(r) : r \in [0, 1]\} \leq 0$ **hacer**:
6. Definir $T = 2T$.
7. Definir $h(r) = f_{rz+(1-r)w}(T)$

8. Definir $r^* = \operatorname{argmin}\{|g_{rz+(1-r)w}(T)| : r \in [0, 1]\}$.

9. **Retornar** $T^{(r^*z+(1-r^*)w)}$

En primer lugar, veamos que el algoritmo termina. Escribimos a f de la siguiente manera:

$$f = a_+x^\alpha + (a_1x^{\sigma_1} + \dots + a_dx^{\sigma_d})$$

donde evidentemente $a_+ > 0$. Por el lema anterior, τ_z, τ_w existen. El mínimo de h en $[0, 1]$ alcanza al ser continua en un compacto. En primer lugar, para todo $r \in [0, 1]$, por convexidad del cono exterior, $rz + (1-r)w \in N_f(\alpha)^\circ$. Por lo tanto, tenemos que vale que:

$$\langle rz + (1-r)w, \alpha - \sigma_i \rangle > 0$$

Luego, por continuidad en un compacto, existe $\delta > 0$ de modo que:

$$\delta = \min_{r \in [0,1]} \min_{1 \leq i \leq d} \{\langle rz + (1-r)w, \alpha - \sigma_i \rangle\}$$

Por otro lado, sea $\beta = \inf_{r \in [0,1]} \langle rz + (1-r)w, \alpha \rangle$. Entonces tenemos que para todo $t > 0$ y $r \in [0, 1]$:

$$f_{rz+(1-r)w}(t) = a_+t^{\langle rz+(1-r)w, \alpha \rangle} + \sum_{i=1}^d a_it^{\langle rz+(1-r)w, \sigma_i \rangle} > a_+t^\beta - \left(\sum_{i=1}^d |a_i|\right)t^{\beta-\delta}$$

y como $\delta > 0$, tenemos que a partir de un t_0 , la expresión será positiva.

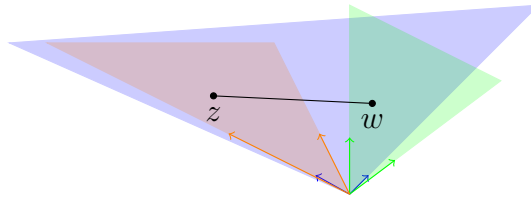


Figura 4.1: El cono azul representa el cono asociado a α , el verde y el naranja a β_+ y β_- respectivamente. z, w son las intersecciones correspondientes.

Veamos ahora que el algoritmo es correcto. Para esto debemos verificar que al acabar el ciclo el valor retornado verifique que $f(x^*) > 0$ y $g(x^*) = 0$. Lo primero se deduce de que estamos considerando $T \geq \tau$ por lo que $f_{rz+(1-r)w}(T) > 0$ para todo $r \in [0, 1]$ debido a que $z, w \in N_\sigma(\alpha)$ que es convexo.

Similarmente, debido a que $T \geq \tau$, $g_z(T) < 0$ y $g_w(T) > 0$ así que por Bolzano existe $r \in (0, 1)$ de modo que $|g_{rz+(1-r)w}(T)| = 0$. Luego el r^* que realiza el mínimo del módulo es el que nos sirve. \square

Es importante notar que la elección de α, β_+, β_- deben ser vértices del polítopo, y no solamente exponentes de monomios no nulos de los polinomios. Por ejemplo,

$g(x, y) = x^2 + y^2 - 2xy = (x - y)^2$ nunca es negativo, a pesar de que el coeficiente del término $(1, 1)$ es negativo. Esto se refleja en que $(1, 1)$ es combinación afín de $(2, 0)$ y $(0, 2)$, por lo que no es vértice del polítopo de Newton asociado.

A priori este argumento no es suficiente para el criterio de Routh-Hurwitz del Teorema 2.2.4 que necesita que $H_n = 0$ y que $H_i > 0$ para $1 \leq i \leq n - 1$. Para esto, como hacen en [20], hacemos:

- Reducimos nuestro polinomio a uno de 5 variables fijando $k_{off} = \ell_{off} = 1$ y las variables $x_3 = x_4 = x_6 = x_7 = x_9 = x_{10} = 1$, así obteniendo polinomios $h_i \in \mathbb{Q}[k_{cat}, y_1, y_2, y_5, y_8]$.
- $h_1, h_2, h_3 > 0$ que verifican numéricamente los autores de [20].

Luego, debemos encontrar un punto $P = (k_{cat}, y_1, y_2, y_5, y_8) \in \mathbb{R}_{>0}^5$ de modo que $h_4(P) > 0, h_5(P) > 0$ y $h_6(P) = 0$. Para esto generalizamos el algoritmo anterior de la siguiente manera: buscamos vertices positivos α_1 en $\text{Newt}(h_4)$, α_2 en $\text{Newt}(h_5)$ y β_+ y β_- como antes para $g = h_6$ de modo que existan

$$z \in N_{h_4}(\alpha_1)^\circ \cap N_{h_5}(\alpha_2)^\circ \cap N_{h_6}(\beta_+)^\circ$$

$$w \in N_{h_4}(\alpha_1)^\circ \cap N_{h_5}(\alpha_2)^\circ \cap N_{h_6}(\beta_-)^\circ$$

Por precisamente los mismos argumentos, existe τ a partir del cual si $T \geq \tau$ entonces $(h_4)_{rz+(1-r)w}(T) > 0$ y $(h_5)_{rz+(1-r)w}(T) > 0$ para $r \in [0, 1]$, y que además $(h_6)_z(T) > 0$ y $(h_6)_w(T) < 0$. Por Bolzano, existe $r \in [0, 1]$ para el cual $(h_6)_{rz+(1-r)w}(T) = 0$. Luego $T^{rz+(1-r)w}$ cumple todo lo pedido.

4.1.1. Implementación

Implementamos el algoritmo en Python (ver cuadernos adjuntos a la tesis). Adicionalmente, el código se encuentra en el Apéndice (ver A.1). Usamos el paquete NumPy [13] a lo largo del código. Para esto construiremos una clase 'Polinomios'. Los polinomios se inician con una lista vacía y término independiente 0. Los métodos `agregar_monomios` y `elegir_término_independiente` que añaden monomios no constantes y fijan término independiente respectivamente.

Los monomios del polinomio forman un diccionario que nombramos `f.variables_y_coeficientes`, donde la llave es una tupla de duplas, donde la primera coordenada se refiere a la variable y la segunda la potencia a la que está elevada. Colocamos solamente potencias mayores a 0 (caso contrario, no colocamos la dupla con la variable). El valor al que accede la llave es el coeficiente que multiplica al monomio. A modo de ejemplo: si nuestro polinomio es $f(x, y, z) = xy + 3z^3 - xz$, tendríamos que `f.variables_y_coeficientes = {((x, 1), (y, 1)) : 1, ((z, 3),) : 3, ((x, 1), (z, 1)) : -1}`.

Finalmente tenemos el método `evaluar_polinomio`, que simplemente evalúa el polinomio en un punto dado.

A partir de los polinomios obtenidos en [20], calculamos el polinomio característico. Usamos la librería `Sympy` [12] para operar, construir la matriz de Hurwitz y obtener las funciones h_4, h_5, h_6 . Para calcular determinantes, usamos `Intertools` para generar permutaciones. Notar que en un principio estas expresiones poseen denominadores $y_1 y_2^3$, pero como estamos trabajando con variables positivas, esto no afecta el signo de h_4, h_5, h_6 por lo que lo quitamos. El código de estos cálculos está en §A.2.

Obtenido h_4, h_5, h_6 lo guardamos en formato `.txt` y lo importamos a Python. Usando una función que transforma strings en polinomios en nuestra clase personalizada, podemos aplicar el algoritmo.

Con respecto al algoritmo propiamente dicho, usamos la librería `SciPy` [24] para la función `Convexhull` (que extrae un subconjunto de vertices de un polítopo) y para programación lineal.

Para encontrar la raíz de $g_{rw+(1-r)z}$, habiendo fijado el T , usamos el método regula-falsi. No obstante, debido al tamaño de h_6 (que en formato `.txt` ocupa 2 megabytes), no tenemos la suficiente precisión para hallar la raíz. A modo expositivo, usamos polinomios más simples como ejemplo, aunque cabe mencionar que escogimos polinomios con grado máximo similar a los de los polinomios reales.

Damos dos versiones del algoritmo. La primera es una versión como la de la Proposición 4.1.1, mientras que la segunda es más general y permite pedir que tanto h_4 como h_5 sean positivos. Estas se encuentran en las secciones A.1.1 y A.1.2 del Apéndice, respectivamente.

4.1.2. Otra Aplicación

A modo ilustrativo, daremos otra aplicación de las ideas del algoritmo de polítopo de Newton a un trabajo más reciente. Analizamos la existencia de bifurcaciones de Hopf para el sistema de la sección 6.1 de [2] en busca redes chicas que presenten bifurcaciones de Hopf. De aquí obtenemos el siguiente sistema:

$$\begin{cases} \dot{x} = k_1 - k_2x - k_4xz \\ \dot{y} = k_2x - k_3yz \\ \dot{z} = k_3yz - k_4xz \end{cases} \quad (4.1)$$

En este caso, en lugar de usar el algoritmo completo, usaremos algunos de sus principios para encontrar el punto de bifurcación. En efecto, se puede ver que al estudiar la matriz de Hurwitz, el primer término es siempre positivo, luego basta estudiar h_2 y h_3 . Notablemente, al calcular h_2 obtenemos:

$$h_2 = k_2 - k_3y + k_3z + k_4x + k_4z$$

El único término negativo es $-k_3y$, por lo que evaluando en $k_3 = y = 1$, si el resto de las otras variables son suficientemente grandes, tenemos que es positivo. Por lo que en este caso

no es necesario buscar α , simplemente podemos escoger un valor razonablemente grande de T y luego verificar que $(h_2)_{rz+(1-r)w}(T) > 0$ que es suficiente.

Para buscar una raíz de h_3 repetimos la idea del algoritmo, buscando un vértice positivo y uno negativo, w, z respectivamente, y evaluando en puntos de la forma $T^{zr+(1-r)w}$, ya que para T suficientemente grande, en $r = 0$ da positivo y $r = 1$ da negativo.

Así obtenemos aproximadamente el punto

$$(x, y, z) = (0,41782816885715746, 1, 0,15692696166791112)$$

y también constantes de reacción $k_1 = 0,15974007245560917, k_2 = 16384,0, k_3 = 1, k_4 = 14858,68000093848$ para los cuales tenemos una bifurcación de Hopf con respecto a k_1 . Efectivamente, se verifica que $h_3(k, x) = 0, h_2(k, x) > 0$ y que $\partial_{k_1} h_3(k, x) \neq 0$. El código se encuentra en § A.1.3.

En ambos problemas vemos que “evaluar correctamente” puede ser muy importante, sobre todo por la complejidad computacional inherente al algoritmo (ver la demostración de Teorema 3.2.1). Aun así, combinando ambos enfoques (es decir, usando las técnicas del algoritmo y evaluando inteligentemente para reducir variables), obtenemos herramientas no sólo para demostrar la existencia de bifurcaciones sino que para hallarlas. Esperamos que el trabajo hecho en Python pueda servir para estos propósitos.

4.2. Otros programas

Realizamos los siguientes programas para verificar numéricamente algunos de los resultados del capítulo 3. El código de estos programas se encuentra presente en el Apéndice en la Sección A.2 y también adjuntos a esta tesis.

- Calculamos la función crítica para la demostración de 3.1.1 para calcular su numerador y denominador y estudiar su signo.
- Verificamos la estabilidad de los puntos de equilibrio de la demostración 3.1.3 mirando el signo de los autovalores de la matriz Jacobiana.
- Simulamos numéricamente de acuerdo con el resultado 3.2.1 y observamos la presencia de oscilaciones.
- Calculamos los polinomios necesarios para poder aplicar el criterio de Routh-Hurwitz a la red de § 4.1.2.
- El cálculo de h_4, h_5, h_6 para la red ERK reducida de acuerdo con la Sección 4.1.1.

Apéndice A

Códigos

En este apéndice se encuentra el código desarrollado a lo largo de la tesis.

A.1. Código del Algoritmo de Polítopo de Newton

Introducimos ahora el código referido al algoritmo del polítopo de Newton (4.1.1).

Polinomios y polítopos asociados.

```
import scipy
from scipy import optimize
from scipy.spatial import ConvexHull
import numpy as np

class Variables:
    def __init__(self, name):
        self.name = name
        self.simbolos = []

    def agregar_variable(self, variable):
        self.simbolos.append(variable)

    #ubica posición de la variable en la lista de variables
    def numero_de_variable(self, variable):
        return self.simbolos.index(variable)

class Polinomio:
    def __init__(self, name):
        self.name = name
        self.variables_y_coeficientes = {}
        self.constante = 0

    def agregar_monomios(self, lista_de_monomios):
        self.variables_y_coeficientes.update(lista_de_monomios)

    def evaluar(self, punto):
        temp = list(self.variables_y_coeficientes.keys())
        coefs = list(self.variables_y_coeficientes.values())
```

```

    sum = 0
    for i in range(len(temp)):
        prod = coefs[i]
        for dupla in temp[i]:
            j = V.numero_de_variable(dupla[0])
            prod = prod * np.power(punto[j], dupla[1])
        sum += prod
    return sum + self.constante

def elegir_termino_independiente(self, valor):
    self.constante = valor

#definimos las variables

V = Variables('V')

V.agregar_variable('k')
V.agregar_variable('y1')
V.agregar_variable('y2')
V.agregar_variable('y5')
V.agregar_variable('y8')

dimension = len(V.simbolos)

def evaluar_monomio(monomio, punto):
    variables = list(monomio.keys())
    coeficiente = monomio.values()
    sum = coeficiente
    for dupla in variables:
        j = V.numero_de_variable(dupla[0])
        prod = prod * np.power(punto[j], dupla[1])
    sum += prod
    return sum

#definimos la mantisa sobre la cual buscaremos el r

mantisa = np.linspace(0,1,10000)

tol = 1e-10

#construimos el polítopo de Newton asociado a un polinomio

def vertice_asociado(monomio):
    punto = np.zeros(dimension)
    for dupla in monomio:
        variable = dupla[0]
        k = V.numero_de_variable(variable)
        punto[k] = dupla[1]
    return punto

def politopo_asociado(polinomio):
    temp = polinomio.variables_y_coeficientes.keys()
    exponentes = []
    vertices_politopo = []

```



```

for mon in temp:
    punto = vertice_asociado(mon)
    exponentes.append(punto)

hull = ConvexHull(exponentes)
indices = hull.vertices
for i in indices:
    vertices_politopo.append(exponentes[i])
return vertices_politopo

#construimos el cono normal de un polítopo asociado a un vértice

def matriz_cono_normal(politopo, vertice):
    M = []
    v = np.zeros(dimension)
    for punto in politopo:
        fila = np.subtract(punto, vertice)
        if not( np.array_equal(fila, v) ):
            #agregamos esta condición para no tener filas nulas en la matriz
            M.append(fila)
    return M

#Usamos programación lineal para determinar si el interior de dos conos se intersecan ,
#usamos un optimizador cualquiera

def interseccion_conos(politopo_f, politopo_g, vertice_f, vertice_g):
    cono_normal_f = matriz_cono_normal(politopo_f, vertice_f)
    cono_normal_g = matriz_cono_normal(politopo_g, vertice_g)
    M = np.concatenate((cono_normal_f, cono_normal_g))
    n = len(M)

    optimizador = -np.ones(dimension)

    vector_tolerancia = np.zeros(n)

    for i in range(n):
        vector_tolerancia[i] = -0.2

    res = scipy.optimize.linprog(optimizador, A_ub=M,
                                b_ub=vector_tolerancia, bounds = [-2,2])
    return res

```

A.1.1. Primera Versión del Algoritmo

```

def algoritmo_de_politopo_de_newton(f,g):
    politopo_f = politopo_asociado(f)
    politopo_g = politopo_asociado(g)

    #debemos clasificar los vertices de f y g entre positivos y negativos ,
    #para esto debemos recuperar el monomio que corresponde al vértice
    #y mirar el signo del coeficiente.

```

```

vertices_positivos_f = []

for vertice in politopo_f:
    monomio = []
    for i in range(dimension):
        variable = V.simbolos[i]
        if vertice[i] > 0:
            dupla = ( variable , int(vertice[i]))
            monomio.append(dupla)

    monomio = tuple(monomio)
    if f.variables_y_coeficientes[monomio] > 0:
        vertices_positivos_f.append(vertice)

vertices_positivos_g = []
vertices_negativos_g = []

for vertice in politopo_g:
    monomio = []
    for i in range(dimension):
        variable = V.simbolos[i]
        if vertice[i] > 0:
            dupla = ( variable , int(vertice[i]))
            monomio.append(dupla)
    monomio = tuple(monomio)
    if g.variables_y_coeficientes[monomio] > 0:
        vertices_positivos_g.append(vertice)
    else:
        vertices_negativos_g.append(vertice)

#buscamos vertices de modo que los conos correspondientes se intersequen
for alfa in vertices_positivos_f:
    for beta_mas in vertices_positivos_g:
        for beta_menos in vertices_negativos_g:
            if (interseccion_conos(f,g, alfa , beta_mas).success and
                interseccion_conos(f,g, alfa , beta_menos).success):

                punto1 = interseccion_conos(f,g, alfa , beta_mas).x
                punto2 = interseccion_conos(f,g, alfa , beta_menos).x

                return buscar_punto( punto1 , punto2 , f , g)

return "No_hay_interseccion_entre_el_interior_de_los_conos"

#funciones auxiliares:

def es_positivo(f, mon):
    coeficiente = f.variables_y_coeficientes[mon]
    return(coeficiente > 0)

#"eleva" T a r * punto1 + (1-r)*punto2 y evalua el polinomio en estos puntos

def exponenciar_el_segmento(punto1,punto2 , polinomio,T):

```

```

i=0
temp = 0
for r in mantisa:
    tau=exponenciar_valor( T, np.multiply(punto1 ,r) + np.multiply(punto2 ,(1-r)) )
    temp = polinomio.evaluar(tau)
    if temp < 0.1:
        return False
return True

#construimos un punto donde cada coordenada proviene de hacer  $T^{\text{vector\_exponente}[i]}$ 
def exponenciar_valor(T,vector_exponente):
    punto = np.zeros(dimension)
    for i in range(dimension):
        punto[i] = np.power(T,vector_exponente[i])
    return punto

def buscar_punto(punto1 , punto2 , f,g):

    T = 2

    while g.evaluar(exponenciar_valor(T,punto1)) < 0 or
           g.evaluar(exponenciar_valor(T,punto2))> 0 or
           np.amin(exponenciar_el_segmento(punto1 ,punto2 ,f, T) , None) < 0.1:
        T = 2*T

    #hacemos biseccion para obtener r necesario
    a = 0
    b = 1
    r_final = 1

    while abs.(g.evaluar(exponenciar_valor(T,punto1*r_final+punto2*(1-r_final)))) > tol:

        r_final = ( a + b ) / 2
        if g.evaluar(exponenciar_valor(T,punto1*r_final+punto2*(1-r_final)))<0:
            a = r_final
        else:
            b = r_final

    #realizamos este paso para obtener el r con el que tenemos la raíz de g
    punto_final = np.zeros(dimension)
    for i in range(dimension):
        punto_final[i] = np.power(T, r_final * punto1[i] + (1-r_final) * punto2[i])

    return punto_final

```

A.1.2. Segunda Versión del Algoritmo

```

#version generalizada del algoritmo de newton

def interseccion_conos_generalizada(politopo_f1 ,politopo_f2 ,politopo_g ,

```

```

        vertice_f1 , vertice_f2 , vertice_g ):

    cono_normal_f1 = matriz_cono_normal(politopo_f1 , vertice_f1 )
    cono_normal_f2 = matriz_cono_normal(politopo_f2 , vertice_f2 )
    cono_normal_g = matriz_cono_normal(politopo_g , vertice_g )

    M = np.concatenate((cono_normal_f1 , cono_normal_f2 , cono_normal_g))
    n = len(M)

    optimizador = np.ones(dimension)

    vector_tolerancia = np.zeros(n)

    for i in range(n):
        vector_tolerancia[i] = -1

    res = scipy.optimize.linprog(optimizador ,
                                A_ub=M, b_ub=vector_tolerancia , bounds = [-2,2])
    return res

def algoritmo_de_politopo_de_netwon_genrealizado(f1,f2,g):
    politopo_f1 = politopo_asociado(f1)
    politopo_f2 = politopo_asociado(f2)
    politopo_g = politopo_asociado(g)

    #debemos clasificar los vertices de f1, f2 y g entre positivos y negativos ,
    #para esto debemos recuperar el monomio que corresponde al vértice y
    #mirar el signo del coeficiente.

    vertices_positivos_f1 = []

    for vertice in politopo_f1:
        monomio = []
        for i in range(dimension):
            variable = V.simbolos[i]
            if vertice[i] > 0:
                dupla = ( variable , int(vertice[i]))
                monomio.append(dupla)
        monomio = tuple(monomio)
        if f1.variables_y_coeficientes[monomio] > 0:
            vertices_positivos_f1.append(vertice)

    vertices_positivos_f2 = []

    for vertice in politopo_f2:
        monomio = []
        for i in range(dimension):
            variable = V.simbolos[i]
            if vertice[i] > 0:
                dupla = ( variable , int(vertice[i]))
                monomio.append(dupla)
        monomio = tuple(monomio)
        if f2.variables_y_coeficientes[monomio] > 0:

```

```

    vertices_positivos_f2.append(vertice)

vertices_positivos_g = []
vertices_negativos_g = []

for vertice in politopo_g:
    monomio = []
    for i in range(dimension):
        variable = V.simbolos[i]
        if vertice[i] > 0:
            dupla = ( variable , int(vertice[i]))
            monomio.append(dupla)
    monomio = tuple(monomio)
    if g.variables_y_coeficientes[monomio] > 0:
        vertices_positivos_g.append(vertice)
    else:
        vertices_negativos_g.append(vertice)

#buscamos vertices de modo que los conos correspondientes se intersequen
for alfa1 in vertices_positivos_f1:
    for alfa2 in vertices_positivos_f2:
        if interseccion_conos(politopo_f1 ,politopo_f2 , alfa1 , alfa2).success:
            for beta_mas in vertices_positivos_g:
                if interseccion_conos_generalizada(politopo_f1 ,politopo_f2 ,
                    politopo_g , alfa1 , alfa2 , beta_mas).success:
                    for beta_menos in vertices_negativos_g:
                        if interseccion_conos_generalizada(politopo_f1 ,politopo_f2 ,
                            politopo_g , alfa1 , alfa2 , beta_menos).success:
                            print(' alfa1=', alfa1)
                            print(' alfa2=', alfa2)
                            print(' beta_mas=', beta_mas)
                            print(' beta_menos=', beta_menos)
                            punto1 = interseccion_conos_generalizada(politopo_f1 ,politopo_f2 ,
                                politopo_g , alfa1 , alfa2 , beta_mas).x
                            punto2 = interseccion_conos_generalizada(politopo_f1 ,politopo_f2 ,
                                politopo_g , alfa1 , alfa2 , beta_menos).x
                            return buscar_punto_generalizada( punto1 , punto2 , f1 , f2 , g)

return "No_hay_interseccion_entre_el_interior_de_los_conos"

def buscar_punto_generalizada(punto1 , punto2 , f1 , f2 , g):
    T = 2

    while g.evaluar(exponenciar_valor(T,punto1)) < 0 or
        g.evaluar(exponenciar_valor(T,punto2))> 0 or
        np.amin(exponenciar_el_segmento(punto1,punto2 ,f1 , T) , None) < 0.1 or
        np.amin(exponenciar_el_segmento(punto1,punto2 ,f2 , T) , None) < 0.1:

        T = 2*T

    a = 0
    b = 1
    r_final = 1
    while abs(g.evaluar(exponenciar_valor(T,punto1*r_final+punto2*(1-r_final)))) > tol:

```

```

fa = g.evaluar(exponenciar_valor(T,punto1 * a + punto2 * (1-a)))
fb = g.evaluar(exponenciar_valor(T,punto1 * b + punto2 * (1-b)))
r_final = (a*fb-b*fa)/(fb-fa)
if g.evaluar(exponenciar_valor(T,punto1 * r_final + punto2 * (1-r_final))) *
g.evaluar(exponenciar_valor(T,punto1 * a + punto2 * (1-a))) < 0:
    b = r_final
else:
    a = r_final

punto_final = np.zeros(dimension)
for i in range(dimension):
    punto_final[i] = np.power(T, r_final * punto1[i] + (1-r_final) * punto2[i])

print('El_punto_obtenido_es', punto_final )
print('Al_evaluar_h6_en_el_punto_obtenemos', g.evaluar(punto_final))
print('Al_evaluar_h4_obtenemos', f1.evaluar(punto_final))
print('Al_evaluar_h5_obtenemos', f2.evaluar(punto_final))

return punto_final

#corremos el algoritmo para polinomios más simples
 #(Pero notar que los grados son similares)

h4_str = "96*k**19*y1**7*y2**4*y5**4*y8_+_128*k**19*y1**7*y2**4*y5**3*y8**2_+_88*k**19*y1**6*y2**5*y5**4*y8_+_64*k**19*y1**6*y2**4*y5**5*y8_+_32*k**19*y1**6*y2**3*y5**5*y8_+_16*k**19*y1**6*y2**2*y5**6*y8_+_8*k**19*y1**6*y2**1*y5**7*y8_+_4*k**19*y1**6*y2**0*y5**8_+_4*k**18*y1**7*y2**4*y5**4*y8_+_12*k**18*y1**7*y2**3*y5**5*y8_+_6*k**18*y1**7*y2**2*y5**6*y8_+_3*k**18*y1**7*y2**1*y5**7*y8_+_1*k**18*y1**7*y2**0*y5**8_+_3*k**17*y1**8*y2**4*y5**4*y8_+_9*k**17*y1**8*y2**3*y5**5*y8_+_4*k**17*y1**8*y2**2*y5**6*y8_+_2*k**17*y1**8*y2**1*y5**7*y8_+_1*k**17*y1**8*y2**0*y5**8_+_2*k**16*y1**9*y2**4*y5**4*y8_+_6*k**16*y1**9*y2**3*y5**5*y8_+_3*k**16*y1**9*y2**2*y5**6*y8_+_1*k**16*y1**9*y2**1*y5**7*y8_+_1*k**16*y1**9*y2**0*y5**8_+_1*k**15*y1**10*y2**4*y5**4*y8_+_4*k**15*y1**10*y2**3*y5**5*y8_+_2*k**15*y1**10*y2**2*y5**6*y8_+_1*k**15*y1**10*y2**1*y5**7*y8_+_1*k**15*y1**10*y2**0*y5**8_+_1*k**14*y1**11*y2**4*y5**4*y8_+_3*k**14*y1**11*y2**3*y5**5*y8_+_1*k**14*y1**11*y2**2*y5**6*y8_+_1*k**14*y1**11*y2**1*y5**7*y8_+_1*k**14*y1**11*y2**0*y5**8_+_1*k**13*y1**12*y2**4*y5**4*y8_+_2*k**13*y1**12*y2**3*y5**5*y8_+_1*k**13*y1**12*y2**2*y5**6*y8_+_1*k**13*y1**12*y2**1*y5**7*y8_+_1*k**13*y1**12*y2**0*y5**8_+_1*k**12*y1**13*y2**4*y5**4*y8_+_1*k**12*y1**13*y2**3*y5**5*y8_+_1*k**12*y1**13*y2**2*y5**6*y8_+_1*k**12*y1**13*y2**1*y5**7*y8_+_1*k**12*y1**13*y2**0*y5**8_+_1*k**11*y1**14*y2**4*y5**4*y8_+_1*k**11*y1**14*y2**3*y5**5*y8_+_1*k**11*y1**14*y2**2*y5**6*y8_+_1*k**11*y1**14*y2**1*y5**7*y8_+_1*k**11*y1**14*y2**0*y5**8_+_1*k**10*y1**15*y2**4*y5**4*y8_+_1*k**10*y1**15*y2**3*y5**5*y8_+_1*k**10*y1**15*y2**2*y5**6*y8_+_1*k**10*y1**15*y2**1*y5**7*y8_+_1*k**10*y1**15*y2**0*y5**8_+_1*k**9*y1**16*y2**4*y5**4*y8_+_1*k**9*y1**16*y2**3*y5**5*y8_+_1*k**9*y1**16*y2**2*y5**6*y8_+_1*k**9*y1**16*y2**1*y5**7*y8_+_1*k**9*y1**16*y2**0*y5**8_+_1*k**8*y1**17*y2**4*y5**4*y8_+_1*k**8*y1**17*y2**3*y5**5*y8_+_1*k**8*y1**17*y2**2*y5**6*y8_+_1*k**8*y1**17*y2**1*y5**7*y8_+_1*k**8*y1**17*y2**0*y5**8_+_1*k**7*y1**18*y2**4*y5**4*y8_+_1*k**7*y1**18*y2**3*y5**5*y8_+_1*k**7*y1**18*y2**2*y5**6*y8_+_1*k**7*y1**18*y2**1*y5**7*y8_+_1*k**7*y1**18*y2**0*y5**8_+_1*k**6*y1**19*y2**4*y5**4*y8_+_1*k**6*y1**19*y2**3*y5**5*y8_+_1*k**6*y1**19*y2**2*y5**6*y8_+_1*k**6*y1**19*y2**1*y5**7*y8_+_1*k**6*y1**19*y2**0*y5**8_+_1*k**5*y1**20*y2**4*y5**4*y8_+_1*k**5*y1**20*y2**3*y5**5*y8_+_1*k**5*y1**20*y2**2*y5**6*y8_+_1*k**5*y1**20*y2**1*y5**7*y8_+_1*k**5*y1**20*y2**0*y5**8_+_1*k**4*y1**21*y2**4*y5**4*y8_+_1*k**4*y1**21*y2**3*y5**5*y8_+_1*k**4*y1**21*y2**2*y5**6*y8_+_1*k**4*y1**21*y2**1*y5**7*y8_+_1*k**4*y1**21*y2**0*y5**8_+_1*k**3*y1**22*y2**4*y5**4*y8_+_1*k**3*y1**22*y2**3*y5**5*y8_+_1*k**3*y1**22*y2**2*y5**6*y8_+_1*k**3*y1**22*y2**1*y5**7*y8_+_1*k**3*y1**22*y2**0*y5**8_+_1*k**2*y1**23*y2**4*y5**4*y8_+_1*k**2*y1**23*y2**3*y5**5*y8_+_1*k**2*y1**23*y2**2*y5**6*y8_+_1*k**2*y1**23*y2**1*y5**7*y8_+_1*k**2*y1**23*y2**0*y5**8_+_1*k**1*y1**24*y2**4*y5**4*y8_+_1*k**1*y1**24*y2**3*y5**5*y8_+_1*k**1*y1**24*y2**2*y5**6*y8_+_1*k**1*y1**24*y2**1*y5**7*y8_+_1*k**1*y1**24*y2**0*y5**8_+_1*k**0*y1**25*y2**4*y5**4*y8_+_1*k**0*y1**25*y2**3*y5**5*y8_+_1*k**0*y1**25*y2**2*y5**6*y8_+_1*k**0*y1**25*y2**1*y5**7*y8_+_1*k**0*y1**25*y2**0*y5**8_+_1"
h5_str ="320*k**22*y1**7*y2**4*y5**5*y8_+_768*k**22*y1**7*y2**4*y5**4*y8**2_+_640*k**22*y1**6*y2**5*y5**4*y8_+_128*k**22*y1**6*y2**4*y5**5*y8_+_64*k**22*y1**6*y2**3*y5**5*y8_+_32*k**22*y1**6*y2**2*y5**6*y8_+_16*k**22*y1**6*y2**1*y5**7*y8_+_8*k**22*y1**6*y2**0*y5**8_+_4*k**21*y1**7*y2**4*y5**4*y8_+_16*k**21*y1**7*y2**3*y5**5*y8_+_8*k**21*y1**7*y2**2*y5**6*y8_+_4*k**21*y1**7*y2**1*y5**7*y8_+_2*k**21*y1**7*y2**0*y5**8_+_4*k**20*y1**8*y2**4*y5**4*y8_+_12*k**20*y1**8*y2**3*y5**5*y8_+_6*k**20*y1**8*y2**2*y5**6*y8_+_3*k**20*y1**8*y2**1*y5**7*y8_+_1*k**20*y1**8*y2**0*y5**8_+_3*k**19*y1**9*y2**4*y5**4*y8_+_8*k**19*y1**9*y2**3*y5**5*y8_+_4*k**19*y1**9*y2**2*y5**6*y8_+_2*k**19*y1**9*y2**1*y5**7*y8_+_1*k**19*y1**9*y2**0*y5**8_+_2*k**18*y1**10*y2**4*y5**4*y8_+_6*k**18*y1**10*y2**3*y5**5*y8_+_3*k**18*y1**10*y2**2*y5**6*y8_+_1*k**18*y1**10*y2**1*y5**7*y8_+_1*k**18*y1**10*y2**0*y5**8_+_1*k**17*y1**11*y2**4*y5**4*y8_+_4*k**17*y1**11*y2**3*y5**5*y8_+_2*k**17*y1**11*y2**2*y5**6*y8_+_1*k**17*y1**11*y2**1*y5**7*y8_+_1*k**17*y1**11*y2**0*y5**8_+_1*k**16*y1**12*y2**4*y5**4*y8_+_3*k**16*y1**12*y2**3*y5**5*y8_+_1*k**16*y1**12*y2**2*y5**6*y8_+_1*k**16*y1**12*y2**1*y5**7*y8_+_1*k**16*y1**12*y2**0*y5**8_+_1*k**15*y1**13*y2**4*y5**4*y8_+_2*k**15*y1**13*y2**3*y5**5*y8_+_1*k**15*y1**13*y2**2*y5**6*y8_+_1*k**15*y1**13*y2**1*y5**7*y8_+_1*k**15*y1**13*y2**0*y5**8_+_1*k**14*y1**14*y2**4*y5**4*y8_+_1*k**14*y1**14*y2**3*y5**5*y8_+_1*k**14*y1**14*y2**2*y5**6*y8_+_1*k**14*y1**14*y2**1*y5**7*y8_+_1*k**14*y1**14*y2**0*y5**8_+_1*k**13*y1**15*y2**4*y5**4*y8_+_1*k**13*y1**15*y2**3*y5**5*y8_+_1*k**13*y1**15*y2**2*y5**6*y8_+_1*k**13*y1**15*y2**1*y5**7*y8_+_1*k**13*y1**15*y2**0*y5**8_+_1*k**12*y1**16*y2**4*y5**4*y8_+_1*k**12*y1**16*y2**3*y5**5*y8_+_1*k**12*y1**16*y2**2*y5**6*y8_+_1*k**12*y1**16*y2**1*y5**7*y8_+_1*k**12*y1**16*y2**0*y5**8_+_1*k**11*y1**17*y2**4*y5**4*y8_+_1*k**11*y1**17*y2**3*y5**5*y8_+_1*k**11*y1**17*y2**2*y5**6*y8_+_1*k**11*y1**17*y2**1*y5**7*y8_+_1*k**11*y1**17*y2**0*y5**8_+_1*k**10*y1**18*y2**4*y5**4*y8_+_1*k**10*y1**18*y2**3*y5**5*y8_+_1*k**10*y1**18*y2**2*y5**6*y8_+_1*k**10*y1**18*y2**1*y5**7*y8_+_1*k**10*y1**18*y2**0*y5**8_+_1*k**9*y1**19*y2**4*y5**4*y8_+_1*k**9*y1**19*y2**3*y5**5*y8_+_1*k**9*y1**19*y2**2*y5**6*y8_+_1*k**9*y1**19*y2**1*y5**7*y8_+_1*k**9*y1**19*y2**0*y5**8_+_1*k**8*y1**20*y2**4*y5**4*y8_+_1*k**8*y1**20*y2**3*y5**5*y8_+_1*k**8*y1**20*y2**2*y5**6*y8_+_1*k**8*y1**20*y2**1*y5**7*y8_+_1*k**8*y1**20*y2**0*y5**8_+_1*k**7*y1**21*y2**4*y5**4*y8_+_1*k**7*y1**21*y2**3*y5**5*y8_+_1*k**7*y1**21*y2**2*y5**6*y8_+_1*k**7*y1**21*y2**1*y5**7*y8_+_1*k**7*y1**21*y2**0*y5**8_+_1*k**6*y1**22*y2**4*y5**4*y8_+_1*k**6*y1**22*y2**3*y5**5*y8_+_1*k**6*y1**22*y2**2*y5**6*y8_+_1*k**6*y1**22*y2**1*y5**7*y8_+_1*k**6*y1**22*y2**0*y5**8_+_1*k**5*y1**23*y2**4*y5**4*y8_+_1*k**5*y1**23*y2**3*y5**5*y8_+_1*k**5*y1**23*y2**2*y5**6*y8_+_1*k**5*y1**23*y2**1*y5**7*y8_+_1*k**5*y1**23*y2**0*y5**8_+_1*k**4*y1**24*y2**4*y5**4*y8_+_1*k**4*y1**24*y2**3*y5**5*y8_+_1*k**4*y1**24*y2**2*y5**6*y8_+_1*k**4*y1**24*y2**1*y5**7*y8_+_1*k**4*y1**24*y2**0*y5**8_+_1*k**3*y1**25*y2**4*y5**4*y8_+_1*k**3*y1**25*y2**3*y5**5*y8_+_1*k**3*y1**25*y2**2*y5**6*y8_+_1*k**3*y1**25*y2**1*y5**7*y8_+_1*k**3*y1**25*y2**0*y5**8_+_1*k**2*y1**26*y2**4*y5**4*y8_+_1*k**2*y1**26*y2**3*y5**5*y8_+_1*k**2*y1**26*y2**2*y5**6*y8_+_1*k**2*y1**26*y2**1*y5**7*y8_+_1*k**2*y1**26*y2**0*y5**8_+_1*k**1*y1**27*y2**4*y5**4*y8_+_1*k**1*y1**27*y2**3*y5**5*y8_+_1*k**1*y1**27*y2**2*y5**6*y8_+_1*k**1*y1**27*y2**1*y5**7*y8_+_1*k**1*y1**27*y2**0*y5**8_+_1*k**0*y1**28*y2**4*y5**4*y8_+_1*k**0*y1**28*y2**3*y5**5*y8_+_1*k**0*y1**28*y2**2*y5**6*y8_+_1*k**0*y1**28*y2**1*y5**7*y8_+_1*k**0*y1**28*y2**0*y5**8_+_1"
h6_str = "2333462222*k**13*y1**5*y2**7*y5**6*y8_+_6493784*k**6*y1**5*y2**4*y5**3*y8**8_+_128*k**6*y1**5*y2**3*y5**4*y8_+_64*k**6*y1**5*y2**2*y5**5*y8_+_32*k**6*y1**5*y2**1*y5**6*y8_+_16*k**6*y1**5*y2**0*y5**7*y8_+_8*k**5*y1**6*y2**4*y5**3*y8_+_40*k**5*y1**6*y2**3*y5**4*y8_+_20*k**5*y1**6*y2**2*y5**5*y8_+_10*k**5*y1**6*y2**1*y5**6*y8_+_5*k**5*y1**6*y2**0*y5**7*y8_+_4*k**4*y1**7*y2**4*y5**3*y8_+_12*k**4*y1**7*y2**3*y5**4*y8_+_6*k**4*y1**7*y2**2*y5**5*y8_+_3*k**4*y1**7*y2**1*y5**6*y8_+_1*k**4*y1**7*y2**0*y5**7*y8_+_2*k**3*y1**8*y2**4*y5**3*y8_+_6*k**3*y1**8*y2**3*y5**4*y8_+_3*k**3*y1**8*y2**2*y5**5*y8_+_1*k**3*y1**8*y2**1*y5**6*y8_+_1*k**3*y1**8*y2**0*y5**7*y8_+_1*k**2*y1**9*y2**4*y5**3*y8_+_4*k**2*y1**9*y2**3*y5**4*y8_+_2*k**2*y1**9*y2**2*y5**5*y8_+_1*k**2*y1**9*y2**1*y5**6*y8_+_1*k**2*y1**9*y2**0*y5**7*y8_+_1*k**1*y1**10*y2**4*y5**3*y8_+_2*k**1*y1**10*y2**3*y5**4*y8_+_1*k**1*y1**10*y2**2*y5**5*y8_+_1*k**1*y1**10*y2**1*y5**6*y8_+_1*k**1*y1**10*y2**0*y5**7*y8_+_1*k**0*y1**11*y2**4*y5**3*y8_+_1*k**0*y1**11*y2**3*y5**4*y8_+_1*k**0*y1**11*y2**2*y5**5*y8_+_1*k**0*y1**11*y2**1*y5**6*y8_+_1*k**0*y1**11*y2**0*y5**7*y8_+_1"

@title Transformar string en polinomio

def transformar_monomio(signo , string):
    num_begin = 0
    num_end = len(string)
    hay_potencia = False
    potencia = 1

    if string[0] == 'y' or string[0] == 'k':
        coeficiente = 1
    elif 'y' in string and 'k' in string:
        num_end = min(string.find('y'),string.find('k')) - 1
        coeficiente = float(string[:num_end])

    elif 'y' in string:
        num_end = string.find('y') - 1
        coeficiente = float(string[:num_end])

    elif 'k' in string:
        num_end = string.find('k') - 1

```

```

    coeficiente = float(string[:num_end])

    #en este caso obtenemos el termino independiente
    else:
        if '10^' in string:
            num_end = string.find('10^') - 1
            coeficiente = float(string[:num_end])
            potencia = 10 ** (int(string[num_end + 4 :]))
            coeficiente = signo * coeficiente * potencia
            return coeficiente
        return float(string[:num_end])

    if '10^' in string:
        num_end = string.find('10^') - 1
        hay_potencia = True

    if hay_potencia:
        final_exponente = string[num_end+1:].find('*') + num_end + 1
        potencia = 10 ** (int(string[num_end + 4 : final_exponente]))

    coeficiente = signo * coeficiente * potencia

    exponentes = []
    for variable in V.simbolos:
        if variable in string:
            j = string.find(variable)
            l = len(variable)
            if j+l < len(string) - 1 and string[j+l] == '^':
                exp_end = j + l + 1
                while string[exp_end].isdigit() and exp_end < len(string)-1:
                    exp_end += 1
                if exp_end == j + l + 1:
                    exponentes.append((variable , int(string[j+l+1])))
                else:
                    exponentes.append((variable , int(string[j+l+1:exp_end])))

            else:
                exponentes.append((variable , 1))

    llave = tuple(exponentes)
    return { llave : coeficiente }

def transformar_en_polinomio(string):
    Res = Polinomio('Res')
    m = len(string)
    temp = string
    end = 0
    begin = 0
    signo = 1

```

```

glo = 1

if temp[0] == '-':
    temp = temp[1:]
    signo = -1
    begin = 1

if '+' in temp and '-' in temp:
    end = min(temp.find('+'), temp.find('-'))
elif '+' in temp:
    end = temp.find('+')
elif '-' in temp:
    end = temp.find('-')
    signo = -1
else:
    end = m+1

finalizado = False
monomio = string[ begin : end]

if isinstance(transformar_monomio(signo , monomio), float):
    Res.elegir_termino_independiente(transformar_monomio(signo , monomio))
else:
    Res.agregar_monomios(transformar_monomio(signo , monomio))

while not finalizado:
    signo = 1

    if '+' in temp and '-' in temp:
        begin = min(temp.find('+'), temp.find('-'))+1
        if temp[begin - 1] == '-':
            signo = -1
    elif '+' in temp:
        begin = temp.find('+') + 1
    elif '-' in temp:
        begin = temp.find('-') + 1
        signo = -1
    else:
        begin = 0

    temp = temp[begin:]

    if '+' in temp and '-' in temp:
        end = min(temp.find('+'), temp.find('-'))
    elif '+' in temp:
        end = temp.find('+')
    elif '-' in temp:
        end = temp.find('-')
    else:
        end = m+1
        finalizado = True

```



```

monomio = temp[ : end]
if isinstance(transformar_monomio(signo , monomio), float):
    Res.elegir_termino_independiente(transformar_monomio(signo , monomio))
else:
    Res.agregar_monomios(transformar_monomio(signo , monomio))

return Res

h4_str = h4_str.replace("_", "")
h4_str = h4_str.replace("**", "^")
h4 = transformar_en_polinomio(h4_str)
h5_str.replace("_", "")
h5_str = h5_str.replace("**", "^")
h5 = transformar_en_polinomio(h5_str)
h6_str.replace("_", "")
h6_str = h6_str.replace("**", "^")
h6 = transformar_en_polinomio(h6_str)

t0 = time.perf_counter()
res = algoritmo_de_politopo_de_netwon_genrealizado(h4,h5,h6)
tf =time.perf_counter()

print('El tiempo de ejecución fue', tf - t0)

h6_pol = Poly(h6_str)

h6_derivado = h6_pol.diff('k')

#lo escribimos como diccionario para poder evaluarlo , de acuerdo a '.eval()' de Sympy

punto_dict = {}

for variable in V.simbolos:
    j = V.numero_de_variable(variable)
    punto_dict.update({ variable : res[j]})

print('La derivada de h6 en el punto es', h6_derivado.eval(punto_dict))

```

A.1.3. Aplicación a Otro Sistema

```

#Aplicamos las ideas del algoritmo a otro sistema

V = Variables('V')

x, y, z, w = symbols("x_y_z_w")
k1, k2, k3, k4 = symbols("k1_k2_k3_k4")

D1 = k2 - k3*y + k3*z + k4*x + k4*z
D2 = D1*(-k2*k3*y +k2*k3*z+k2*k4*x+k3*k4*x*z-k3*k4*y*z + k3*k4*z**2)-2*k2*k3*k4*x*z
D2 = expand(D2)

```

```

# Por Liu , queremos  $D1 > 0$ ,  $D2 = 0$ 
#(y la derivada de D2 respecto del parámetro correspondiente no nula)

#Por inspección, notamos que  $y=1$ ,  $k3 = 1$  es una buena sustitución,
#ya que los términos de D1 quedan positivos y por ende para variables
#suficientemente grandes de  $x, y, z$  tenemos que  $D1 > 0$ 

C1 = D1.subs([(y,1),(k3,1)])
C2 = D2.subs([(y,1),(k3,1)])

C2 = expand(C2)
C2 = C2.subs([(x,y1),(z,y2)])
C2_str = str(C2)
C2_str = C2_str.replace("_", "")
C2_str = C2_str.replace("**", "^")

lista_variables = ['y1', 'y2', 'k2', 'k3', 'k4']
for variable in lista_variables:
    V.agregar_variable(variable)

dimension = len(V.simbolos)
C2_pol = transformar_en_polinomio(C2_str)

#calculamos el polítopo asociado, debemos modificar la función debido a que tiene
#'pocos terminos' y que conexhull necesita que el polítopo
#tenga dimensión = a la cantidad de variables

def politopo_asociado_modificado(polinomio):
    temp = polinomio.variables_y_coeficientes.keys()
    exponentes = []
    vertices_politopo = []
    for mon in temp:
        punto = vertice_asociado(mon)
        exponentes.append(punto)
    exponentes_guardados = exponentes

    exponentes = np.delete(exponentes, 3, axis=1)

    hull = ConvexHull(exponentes)
    indices = hull.vertices
    for i in indices:
        vertices_politopo.append(exponentes_guardados[i])
    return vertices_politopo

politopo = politopo_asociado_modificado(C2_pol)

#separamos los vertices de C2 en positivos y negativos.
#Como para T suficientemente grandes C1 es positivo,
#directamente realizamos la optimización

vertices_positivos_C2 = []
vertices_negativos_C2 = []

```

```

for vertice in politopo:
    monomio = []
    for i in range(dimension):
        variable = V.simbolos[i]
        if vertice[i] > 0:
            dupla = ( variable , int(vertice[i]))
            monomio.append(dupla)
    monomio = tuple(monomio)
    if C2_pol.variables_y_coeficientes[monomio] > 0:
        vertices_positivos_C2.append(vertice)
    else:
        vertices_negativos_C2.append(vertice)

#escojemos un vertice de cada uno

beta_mas = vertices_positivos_C2[0]
beta_menos = vertices_negativos_C2[1]
politopo_trivial = []
vertice_trivial = np.zeros(dimension)

#buscamos un punto interior a cada uno, podemos repetir el argumento de
#programación dinámica con el polítopo nulo

punto1 = interseccion_conos(politopo , politopo , beta_mas , beta_mas ).x
punto2 = interseccion_conos(politopo , politopo , beta_menos , beta_menos ).x

#con un valor razonablemente grande de T, podemos anular el '-1' que
#es la unica parte negativa de C1. Probamos T=2
T = 2

#usamos bisección
while C2_pol.evaluar(exponenciar_valor(T,beta_mas)) < 0 or
C2_pol.evaluar(exponenciar_valor(T,punto2))> 0:
    T = 2*T
a = 0
b = 1
r_final = 1
tol = 1e-08
while abs(C2_pol.evaluar(exponenciar_valor(T,punto1*r_final+punto2*(1-r_final)))) > tol:
    r_final = (a+b)/2
    if C2_pol.evaluar(exponenciar_valor(T,punto1*r_final+punto2*(1-r_final))) > 0:
        b = r_final
    else:
        a = r_final

punto_final = np.zeros(dimension)
for i in range(dimension):
    punto_final[i] = np.power(T, r_final * punto1[i] + (1-r_final) * punto2[i])

#evaluamos en C1 y verificamos que da positivo
#derivamos D2 y verificamos que no da 0

```

```

D2_derivado = diff(D2,k1)

C1_en_punto = C1.subs([(x,punto_final[0]),(y,1),(z,punto_final[1]),(k3,1),
(k1,punto_final[2]),(k2,punto_final[3]),(k4,punto_final[4])])
D2_derivado_en_punto = D2.subs([(x,punto_final[0]),(y,1),(z,punto_final[1]),(k3,1),
(k1,punto_final[2]),(k2,punto_final[3]),(k4,punto_final[4])])
print('El punto obtenido es x=', punto_final[0], ', y=',1, 'z=', punto_final[1], ',
con constantes de reacción k1=', punto_final[2], ', k2=', punto_final[3], ',
k3=',1, 'k4=', punto_final[4])
print('Al evaluar D2 en el punto obtenemos', C2_pol.evaluar(punto_final))
print('Al evaluar D1 en el punto obtenemos', C1_en_punto)
print('La derivada de D2 con respecto a k1 en el punto da', D2_derivado_en_punto )

```

A.2. Código de Otros Programas

Introducimos el código escrito para la sección 4.2.

Función crítica

```

import numpy as np
import matplotlib.pyplot as plt

from sympy import *
# from sympy import Matrix
# from sympy.abc import x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12
x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12 = symbols("x1_x2_x3_x4_x5_x6_x7_x8_x9_
a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13 = symbols("a1_a2_a3_a4_a5_a6_a7_a
c1, c2, c3 = symbols("c1_c2_c3")
k1, k3, koff, kcat, m2, m3, l1, l3, loff, lcat, n1, n3 = symbols("k1_k3_koff_kcat_m2_m3_

# Asumimos k_on=l_on=0

h1 = x1 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12 - c1
h2 = x2 + x7 + x8 + x11 - c2
h3 = x3 + x4 + x5 + x6 - c3
h4 = a12*x3*x12 - x4
h5 = a3*x4 - x5 - a2*x8
h6 = a13*x3*x9 - x6
h7 = a5*x11 - a4*x6 - x7
h8 = a11*x2*x10 - x8
h9 = a9*x7 - x6
h10 = a7*x5 - x8
h11 = a10*x1*x2 - x11
h12 = x7 - a1*x5

# Calculamos la matriz jacobiana respecto de x1,...,x12

H = Matrix([h1, h2, h3, h4, h5, h6, h7, h8, h9, h10, h11, h12])
X = Matrix([x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12])
J = H.jacobian(X)

D = J.det()

```

```

# Sustituimos con la parametrización (a2,a4,x1,...,x12) ->
#(a1,a3,a5,a7,a9,a10,a11,a12,a13,x1,...,x12)
Ds = D.subs([(a1, x7/x5), (a3, (a2*x8+x5)/x4), (a5, (a4*x6+x7)/x11), (a7,x8/x5),
(a9,x6/x7), (a10,x11/(x1*x2)), (a11,x8/(x2*x10)), (a12,x4/(x3*x12)), (a13,x6/(x3*x9))])

# Sustituimos a2=m3/l_cat, a4=n3/k_cat
Dss = Ds.subs([(a2,m3/lcat),(a4,n3/kcat)])

Dsst = together(Dss)

print(Dsst)
#Observamos que el numerador es negativo y el denominador es positivo
#ya que todas las variables son no negativas

```

Verificación de estabilidad

```

import numpy as np
import matplotlib.pyplot as plt

from sympy import *
# from sympy import Matrix
# from sympy.abc import x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12
x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12 = symbols("x1_x2_x3_x4_x5_x6_x7_x8_x9_x10_x11_x12")
a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13 = symbols("a1_a2_a3_a4_a5_a6_a7_a8_a9_a10_a11_a12_a13")
c1, c2, c3 = symbols("c1_c2_c3")
k1, k3, koff, kcat, m2, m3, l1, l3, loff, lcat, n1, n3, kon = symbols("k1_k3_koff_kcat_m2_m3_l1_l3_loff_lcat_n1_n3_kon")

h1 = x1 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12 - c1
h2 = x2 + x7 + x8 + x11 - c2
h3 = x3 + x4 + x5 + x6 - c3
h4 = a12*x3*x12 - x4
h5 = a3*x4 - x5 - a2*x8
h6 = a13*x3*x9 - x6
h7 = a5*x11 - a4*x6 - x7
h8 = a11*x2*x10 - x8
h9 = a9*x7 - x6 - a8*x2*x9
h10 = a7*x5 - x8
h11 = a10*x1*x2 - x11
h12 = x7 - a1*x5

# Calculamos la matriz jacobiana respecto de x1,...,x12

H = Matrix([h1, h2, h3, h4, h5, h6, h7, h8, h9, h10, h11, h12])
X = Matrix([x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12])
J = H.jacobian(X)

#fijamos las constantes de reaccion como en la demostracion
Js = J.subs([(a1, x7/x5), (a3, (a2*x8+x5)/x4), (a5, (a4*x6+x7)/x11), (a7,x8/x5),
(a9,(a8*x2*x9+x6)/x7), (a10,x11/(x1*x2)), (a11,x8/(x2*x10)), (a12,x4/(x3*x12)), (a13,x6/(x3*x9))])
Jss = Js.subs([(a2,m3/lcat),(a4,n3/kcat),(a8,kon/n3)])
Jsss = Jss.subs([(c1,46),(c2,13),(c3,13),(k1,2),(k3,1.1),(kcat,1),(kon,5),(koff,15),
(11,2),(13,1.1),(lcat,1),(loff,10),(m2,20),(m3,10),(n1,20),(n3,10)])

#evaluamos ahora la matriz jacobiana en los distintos puntos de equilibrio

```

```
Jsss1 = Jsss.subs([(x1, 20.72107755),(x2, 0.2956877203),(x3,3.248789181),
(x4,7.821850626),(x5,0.7821850626), (x6,1.147175131), (x7,0.7821850626),
(x8,0.7821850626),(x9,0.1765542587), (x10,1.322653950), (x11,11.13994215),
(x12,1.324191138)])
```

```
#print(Jsss1)
```

```
e = np.linalg.eig(np.array(Jsss1).astype(np.float64))
print(e[0])
```

Hurwitz Aplicado al Otro Sistema

```
import numpy as np
import matplotlib.pyplot as plt

from sympy import *
# from sympy import Matrix
# from sympy.abc import x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12
x, y, z, w = symbols("x_y_z_w")
k1, k2, k3, k4 = symbols("k1_k2_k3_k4")

# Red

f1 = k1 - k2*x -k4*x*z
f2 = k2*x-k3*y*z
f3 = k3*y*z - k4*x*z

# Calculamos la matriz jacobiana respecto de x,y,z

H = Matrix([f1, f2, f3])
X = Matrix([x, y, z])
J = H.jacobian(X)

p = J.charpoly(w)

print(p)

D1 = k2 - k3*y + k3*z + k4*x + k4*z
D2 = D1*(-k2*k3*y+k2*k3*z+k2*k4*x+k3*k4*x*z-k3*k4*y*z + k3*k4*z**2)-2*k2*k3*k4*x*z
```

Simulaciones de Oscilaciones de ERK reducido

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp

def f_ERK(t,x,k1,k3,kcat,koff,m,l1,l3,lcat,loff,n):
    x1,x2,x3,x4,x5,x6,x7,x8,x9,x10 = x
    dx1 = -k1*x1*x2+n*x6*x8+lcat*x10
    dx2 = -k1*x1*x2+kcat*x4+koff*x4
    dx3 = k1*x1*x2-k3*x3
    dx4 = k3*x3-kcat*x4-koff*x4
```

```

dx5 = m*x2*x7-11*x5*x8+kcat*x4
dx6 = -n*x6*x8+koff*x4
dx7 = -m*x2*x7+loff*x10
dx8 = -11*x5*x8+loff*x10+lcat*x10
dx9 = 11*x5*x8-13*x9
dx10 = -loff*x10+13*x9-lcat*x10
return [dx1, dx2, dx3, dx4, dx5, dx6, dx7, dx8, dx9, dx10]

kcat = 3
a = 28.7459

argumentos = (kcat+1)*a**2,(kcat+1)*a**2,kcat,1,a,kcat*a**2+1,
(kcat*a**2+1)/a**2,kcat*a**2,1,1

print(argumentos)

t0 =0
tF = 5

x0 = [1,1,1,a**2,1+0.05,a**2-0.05,1/a,1,a**2,1]

sol = solve_ivp(f_ERK, [t0, tF], x0, args=(argumentos),
                dense_output=True, method='BDF')

t = np.linspace(t0, tF, 300)

z = sol.sol(t)

plt.plot(t, z[0].T, label='x1')
plt.plot(t, z[1].T, label='x2')
plt.plot(t, z[2].T, label='x3')
plt.plot(t, z[4].T, label='x5')
plt.plot(t, z[6].T, label='x7')
plt.plot(t, z[7].T, label='x8')
plt.plot(t, z[9].T, label='x10')
plt.legend()
plt.show()

plt.plot(x0[0],x0[2], '*g', label='valor_inicial', markersize=15)
plt.plot(z[0].T,z[2].T)
plt.plot(z[0].T[-1],z[2].T[-1], '*m', label='valor_final', markersize=15)
plt.xlabel('x1')
plt.ylabel('x3')
plt.legend()
plt.show()

```

Cálculo de h_4, h_5, h_6 .

```

import sympy as sp
from sympy import *
from sympy.abc import k,a,b,c,d
from itertools import permutations
import time

```

```
k, y1, y2, y5, y8 = symbols('k_y1_y2_y5_y8')
```

```
def matriz_de_hurwitz(lista_polinomio):
    n = len(lista_polinomio)
    M = np.zeros([n,n])
    M = sp.Matrix(M)
    for i in range(n):
        for j in range(n):
            if 2*i-j<0 or 2*i-j > n-1:
                M[i,j] = 0
            else:
                M[i,j] = lista_polinomio[2*i+-j]
    return M
```

```
def subdeterminante(M,k):
    S = Matrix(np.zeros([k,k]))
    for i in range(k):
        for j in range(k):
            S[i,j] = M[i,j]
    S = S.applyfunc(lambda p: p.as_expr())
    sum = 0
    for perm in permutations(range(0,k)):
        prod = perm_parity(perm)
        for i in range(k):
            prod = prod * S[i,perm[i]]
        sum += prod
    return sum
```

```
def perm_parity(lst):
    for i in range(0,len(lst)-1):
        if lst[i] != i:
            signo *= -1
            mn = min(range(i,len(lst)), key=lst.__getitem__)
            lst = list(lst)
            lst[i],lst[mn] = lst[mn],lst[i]
            lst = tuple(lst)
    return signo
```

```
p_7 = poly(1, [k])
p_6 = poly(((y2 + (k + 1)*y1*y2)/y2 + (y8 + (k + 1)*y5*y8)/y8 + (k + 1)*y8 + (k + 1)*y2
p_5 = poly((((k + 1)*y1 + ((y8 + (k + 1)*y5*y8)*(y2 + (k + 1)*y1*y2)/y8 + (y2/y1 + (k + 1)
p_4 = poly((((y8 + (k + 1)*y5*y8)/y8 + (k + 1)*y8 + 4*k + 4)*(k + 1)*y1 + ((k + 1)*y5*(y
p_3 = poly((((k + 1)*y5 + ((y8/y5 + 4*k + 4)*(k + 1)*y5*y8 + y8*(4*k + 4))/y8 + (4*k + 4
p_2 = poly((((k + 1)*y5*(4*k + 4) + (((4*k + 4)*y8/y5 + (3*k + 3)*(k + 1) + (2*k + 1)*(k
p_1 = poly((k + 1)*y1*((k + 1)*y5*((3*k + 2)*(k + 1) + (2*k + 1)*(k + 1) + (k + 1)**2) +
p_0 = poly((1/y1*y2)*((4*(k + 1)**3*y5*y8*k + ((k*(2*k + 2)*(1/y5 + 1)*y8 + k*(k + 1))*
```

```
caracteristico = [p_0,p_1,p_2,p_3,p_4,p_5,p_6,p_7]
caracteristico_simplificado = []
```



```

for pol in caracteristico:
    pol = expand(pol.as_expr())
    pol = simplify(pol.as_expr())
    caracteristico_simplificado.append(pol)

H = matriz_de_hurwitz(caracteristico_simplificado)

def determinante(M,n):
    if n == 1:
        return M.col(0)[0]
    sum = 0
    print(M)
    for i in range(n):
        N = M.col_del(0)
        N = M.row_del(i)
        sum += paridad(i) * M.col(0)[i] * determinante(N,n-1)

    return sum

def paridad(n):
    if n % 2 == 0:
        return 1
    return -1

h4 = subdeterminante(H,4)
h5 = subdeterminante(H,5)
h6 = subdeterminante(H,6)

h4 = y1 * y1**3 * h4
h5 = y1**3 * h5
h6 = y1**3 * h6

```

Bibliografía

- [1] B. Alberts, A. Johnson, J. Lewis, D. Morgan, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. 500 Tips. Garland Science, 2014.
- [2] Murad Banaji and Balázs Boros. The smallest bimolecular mass action reaction networks admitting andronov–hopf bifurcation. *Nonlinearity*, 36(2):1398, 2023.
- [3] Balázs Boros. Dynamic system properties of biochemical reaction systems. Master’s thesis, Department of Mathematics, Faculty of Science, VU University Amsterdam, Amsterdam, 2008.
- [4] Carsten Conradi, Elisenda Feliu, Maya Mincheva, and Carsten Wiuf. Identifying parameter regions for multistationarity. *PLoS Comput. Biol.*, 13(10):e1005751, 2017.
- [5] Carsten Conradi, Nida Obatake, Anne Shiu, and Xiaoxian Tang. Dynamics of ERK regulation in the processive limit. *J. Math. Biol.*, 82:article 32, 2021.
- [6] Carsten Conradi and Anne Shiu. A global convergence result for processive multisite phosphorylation systems. *Bulletin of Mathematical Biology*, 77:126–155, 2015.
- [7] W.A. Coppel. *Stability and Asymptotic Behavior of Differential Equations*. Heath mathematical monographs. Heath, 1965.
- [8] David A. Cox, John Little, and Donal O’Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer Publishing Company, Incorporated, 4th edition, 2015.
- [9] David A. Cox, John B. Little, and Donal O’Shea. *Using Algebraic Geometry*, volume 185 of *Graduate Texts in Mathematics*. Springer, first edition, 1998.
- [10] Gheorghe Craciun, J. William Helton, and Ruth J. Williams. Homotopy methods for counting reaction network equilibria. *Mathematical Biosciences*, 216(2):140–149, 2008.
- [11] Alicia Dickenstein, Mercedes Perez Millan, Anne Shiu, and Xiaoxian Tang. Multistationarity in structured reaction networks. *B. Math. Biol.*, 81(5):1527–1581, 2019.
- [12] Eli S Groban, Arjun Narayanan, and Matthew P Jacobson. Conformational changes in protein loops and helices induced by post-translational phosphorylation. *PLOS Computational Biology*, 2(4):1–13, 04 2006.

- [13] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [14] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research*. McGraw-Hill, New York, NY, USA, seventh edition, 2001.
- [15] Badal Joshi and Anne Shiu. Atoms of multistationarity in chemical reaction networks. *Journal of Mathematical Chemistry*, 51:153–178, 2012.
- [16] Hassan K Khalil. *Nonlinear systems; 3rd ed.* Prentice-Hall, Upper Saddle River, NJ, 2002. The book can be consulted by contacting: PH-AID: Wallet, Lionel.
- [17] Yuri A. Kuznetsov. *Elements of Applied Bifurcation Theory (2nd Ed.)*. Springer-Verlag, Berlin, Heidelberg, 1998.
- [18] Ana Legaspi. Multiestacionariedad en redes de reacciones bioquímicas. Master’s thesis, Depto. de Matemática, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, 2020.
- [19] Wei-Min Liu. Criterion of hopf bifurcations without using eigenvalues. *Journal of Mathematical Analysis and Applications*, 182:250–256, 1994.
- [20] Nida Obatake, Anne Shiu, Xiaoxian Tang, and Angélica Torres. Oscillations and bistability in a model of ERK regulation. *J. Math. Biol.*, 79(4):1515–1549, 2019.
- [21] Aleena L. Patel and Stanislav Y. Shvartsman. Outstanding questions in developmental ERK signaling. *Development*, 145(14), 07 2018. dev143818.
- [22] Boris Rubinstein, Henry H. Mattingly, Alexander M. Berezhkovskii, and Stanislav Y. Shvartsman. Long-term dynamics of multisite phosphorylation. *Molecular Biology of the Cell*, 27:2331 – 2340, 2016.
- [23] Robert Vandervorst. Topological methods for nonlinear differential equations. www.few.vu.nl/~vdvorst/Lectures2014a.pdf, 2014.
- [24] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

- [25] Carsten Wiuf and Elisenda Feliu. Power-law kinetics and determinant criteria for the preclusion of multistationarity in networks of interacting species. *SIAM J. Appl. Dyn. Syst.*, 12:1685–1721, 2013.
- [26] Günter M Ziegler. *Lectures on polytopes*, volume 152. Springer Science & Business Media, 2012.

