Dynamic Phenomena in Tristable Reaction-Diffusion Models

by

Kevin Rohe
born in Hermeskeil (Germany)

Accepted Dissertation thesis for the partial fulfilment of the requirements for a Doctor of Natural Sciences

Fachbereich 3: Mathematik/Naturwissenschaften
Universität Koblenz-Landau

Reviewer:
Prof. (UMP6) Dr. Christian Fischer
Prof. Dr. Jaime Cisternas

Examiner:
Prof. (UMP6) Dr. Christian Fischer
Prof. Dr. Wolfgang Imhof
Prof. Dr. Stefan Wehner

Date of the oral examination: 21st july 2022

Danksagung

Zuerst möchte ich mich ausdrücklich bei Herrn Wehner für die Aufnahme als Doktorand und die Möglichkeit an diesem interessanten Thema zu arbeiten bedanken.

Ebenfalls danke ich Christian Fischer für die Übernahme meiner Betreuung und die Begleitung bis zur Finalisierung der Arbeit.

I would also like to express my sincere thanks to Jaime Cisternas for the work we did together.

Ich bedanke mich auch bei Prof. Imhof für die abschließende Begleitung der Promotion als fachfremder Prüfer.

Weiterhin bedanke ich mich bei allen meinen Kollegen in der Surface Science Gruppe, besonders bei meinen Leidensgenossen Torben Schlebrowski, Lucas Beucher, Melanie Fritz und Wafa Ghabri, bei Susanne Körsten für Korrekturen und bei Thomas Knieper, Jürgen Schwab und Merten Joost für die technische Unterstützung.

Schließlich bedanke ich mich bei meinen Eltern und Geschwistern, die mich auf diesem Weg unterstützt haben et en particulier à Lise pour ses corrections sévères et pour m'avoir enfin poussé à rendre mon travail.

Summary

The presented study was motivated by the dynamic phenomena observed in basic catalytic surface reactions, especially by bi- and tristability, and the interactions between these stable states. In this regard, three reaction-diffusion models were developed and examined using bifurcation theory and numerical simulations.

A first model was designed to extend the bistable CO oxidation on Ir(111) to include hydrogen and its oxidation. The differential equation system was analyzed within the framework of bifurcation theory, revealing three branches of stable solutions. One state is characterized by high formation rates (upper rate state, UR), while the other two branches display low formation rates (lower rate (LR) & very low rate (VLR) states). The overlapping branches form the shape of a 'swallowtail', the curve of saddle-node bifurcations forming two cusps. Increasing the temperature leads to a subsequent unfolding and hence decreases the system complexity.

A series of numerical simulations representing possible experiments was conducted to illustrate the experimental accessibility (or the lack) of said states. Relaxation experiments show partially long decay times. Quasistatic scanning illustrates the existence of all three states within the tristable regime and their respective conversion once crossing a fold.

A first attempt regarding the state dominance in reaction-diffusion fronts was done. While UR seems to dominate in 1D, a 2D time-evolution shows that LR invades the interphase between UR and VLR.

Subsequently, a generic monospecies mock model was used for the comprehensive study of reaction-diffusion fronts. A quintic polynomial as reaction term was chosen, derived by the sixth-order potential associated with the 'butterfly bifurcation'. This ensures up to three stable solutions(u_0,u_1,u_2), depending on the four-dimensional parameter space.

The model was explored extensively, identifying regions with similar behaviors. A term for the front velocity connecting two stable states was derived, depending only on the relative difference of the states' potential wells. Equipotential curves were found, where the front velocity vanishes of a given front. Numerical simulations on a two-dimensional, finite disk using a triangulated mesh supported these findings.

Additionally, the front-splitting instability was observed for certain parameters. The front solution u_{02} becomes unstable and divides into u_{01} and u_{12}, exhibiting different front velocities. A good estimate for the limit of the front splitting region was given and tested using time evolutions.

Finally, the established mock model was modified from continuous to discrete space, utilizing a simple domain in 1D and three different lattices in 2D (square, hexagonal, triangular). For

low diffusivities or large distances between coupling nodes, fronts can become pinned, if the parameters are within range of the equipotential lines. This phenomenon is known as propagation failure and its extent in parameter space was explored in 1D. In 2D, an estimate was given for remarkable front orientations respective to the lattice using a pseudo-2D approximation.

Near the pinning region, front velocities differ significantly from the continuous expectation as the shape of the curve potential becomes significant. Factors that decide the size and shape of the pinning regions are the coupling strength, the lattice, the front orientation relative to the lattice, and the front itself. The bifurcation diagram shows a snaking curve in the pinning region, each alternating branch representing a stable or unstable frozen front solution.

Numerical simulations supported the observations concerning propagation failure and lattice dependence. Furthermore, the influence of front orientation on the front velocity was explored in greater detail, showing that fronts with certain lattice-dependent orientations are more or less prone to propagation failure. This leads to the possibility of pattern formation, reflecting the lattice geometry. An attempt to quantify the front movement depending on angular front orientation has shown reasonable results and good agreement with the pseudo-2D approach.

Zusammenfassung

Motiviert wurde die vorliegende Arbeit durch die dynamischen Phänomene, die bei grundlegenden katalytischen Oberflächenreaktionen beobachtet werden, insbesondere durch Bi- und Tristabilität und die Wechselwirkungen zwischen diesen stabilen Zuständen. In diesem Zusammenhang wurden drei Reaktions-Diffusions-Modelle entwickelt und auf Bifurkationen analytisch und mittels numerischer Simulationen untersucht.

Das erste Modell wurde entwickelt, um die bistabile CO-Oxidation auf Ir(111) um Wasserstoff und dessen Oxidationsreaktionen zu erweitern. Das Differentialgleichungssystem wurde im Rahmen der Bifurkationstheorie analysiert, wobei drei Zweige stabiler Lösungen gefunden wurden. Einer der Zustände ist durch hohe Bildungsraten gekennzeichnet (upper rate, UR), während die anderen beiden Zweige niedrige Bildungsraten aufweisen (lower rate (LR) & very low rate (VLR)). Die Kurve der Sattel-Knoten-Bifurkationen bildet zwei Spitzen aus, wodurch die sich überschneidenden Zustände die Form eines Schwalbenschwanzes bilden. Eine Temperaturerhöhung führt zur Entfaltung und damit zu einer Komplexitätserniedrigung des Systems.

Um die experimentelle (Un-)Zugänglichkeit dieser Zustände zu veranschaulichen wurde eine Reihe von numerischen Simulationen durchgeführt, die mögliche Experimente widerspiegeln. Relaxationsexperimente zeigen teilweise lange Konvergenzzeiten. Quasi-statisches Scannen des Versuchsparameters zeigt die Existenz aller drei Zustände innerhalb des tristabilen Region und ihre jeweilige Umwandlung beim Verlassen desselben.

Ein erster Versuch bezüglich Reaktions-Diffusions-Fronten zwischen den stabilen Zuständen wurde durchgeführt. In 1D dominiert UR, während in 2D die Interphase zwischen UR und VLR durch den LR Zustand durchdrungen wird.

Anschließend wurde ein generisches ‘Parodie’-Monospezies-Modell für die umfassende Untersuchung von Reaktions-Diffusions-Fronten verwendet. Als Reaktionsterm wurde ein Polynom fünften Grades gewählt. Dies resultiert aus einem polynomischen Potential sechster Ordnung, das mit der “Schmetterlingsbifurkation” verbunden ist. Dies garantiert abhängig von dem vierdimensionalen Parameterraum bis zu drei stabile Lösungen (u_0,u_1,u_2).

Das Modell wurde eingehend untersucht, wobei Regionen mit ähnlichem Verhalten identifiziert wurden. Es wurde ein Term für die Frontgeschwindigkeit zwischen zwei stabilen Zuständen abgeleitet, der eine Abhängigkeit von der relativen Potentialdifferenz der beiden Zustände zeigt. Es wurden Äquipotentialkurven gefunden, bei denen die Geschwindigkeit der zugehörigen Front verschwindet. Numerische Simulationen auf einer zweidimensionalen, endlichen Scheibe unterstützten diese Ergebnisse.

Außerdem wurde die Front-Splitting-Instabilität beobachtet, bei der die Frontlösung u_{02} instabil wird und sich in u_{01} und u_{12} mit je unterschiedlichen Geschwindigkeiten aufteilt. Eine gute Schätzung zu den Grenzen der Front-Splitting-Region wurde gegeben und mit Hilfe von numerischen Zeitentwicklungen überprüft.

Schließlich wurde das etablierte kontinuierliche Modell räumlich diskretisiert, wobei eine einfache Domäne in 1D und drei verschiedene Gitter in 2D (quadratisch, hexagonal, dreieckig) verwendet wurden. Bei niedrigen Diffusivitäten oder großen Abständen zwischen den gekoppelten Knoten können die Fronten 'einfrieren', falls die Parameter in der Nähe einer Äquipotentiallinie liegen. Dieses Phänomen ist als Propagationsversagen (PF) bekannt und sein Ausmaß im Parameterraum (Pinning Region) wurde in 1D untersucht. In 2D wurde zunächst eine Schätzung für die Frontausbreitung in ausgezeichnete Gitterrichtungen mittels einer Pseudo-2D-Näherung vorgenommen.

Nahe der Pinning-Region weichen die Frontgeschwindigkeiten erheblich von der kontinuierlichen Erwartung ab, da die exakte Form des Potentials signifikant wird. Größe und Form der Pinning-Regionen wird von der Kopplungsstärke, dem Gitter, die Frontausrichtung zum Gitter und die Frontlösung selbst entschieden. Das Bifurkationsdiagramm zeigt eine schlängelnde Kurve innerhalb der Pinning-Region, wobei jeder abwechselnde Zweig aus stabilen bzw. instabilen, eingefrorenen Fronten besteht.

Numerische Simulationen bestätigten die Beobachtungen bezüglich des PF und der Gitterabhängigkeit. Darüber hinaus wurde der Einfluss der Frontorientierung auf die Geschwindigkeit genauer untersucht. Es wurde gezeigt, dass Fronten mit ausgezeichneter Orientierung zum Gitter mehr oder weniger anfällig für PF sind. Hieraus resultiert die Möglichkeit zur Stabilisierung von metastabilen Mustern, welche die Gittergeometrie widerspiegelt. Die Quantifizierung der winkelabhängigen Frontausbreitung zeigt plausible Ergebnisse mit einer guten Übereinstimmung zum Pseudo-2D-Ansatz.

Contents

1

Introduction

A recurring theme in nature is the emergence of complex structures through the self-assembly of simple units and afar from equilibrium conditions. Analysis of the driving forces sometimes may be sufficient to understand these phenomena in simple systems with few relevant properties and agents. Otherwise, their description can become more complex as the number of agents and their interactions increases or the state of a system changes (globally or locally). Often, nonlinear dynamics such as positive or negative feedback play a large role.

These systems can often be described as a reaction-transport type system (or reaction-diffusion (RD) in physico-chemical systems), which consists of local change and spatial propagation of the agents under given properties. A prime example of reaction-transport systems with the possibility of self-assembling patterns is heterogeneous catalysis. Here, the interphase between catalyst and reactants defines a limited spatial regime as the site of the chemical reaction.

In the following section 1.1 a short introduction to the kinetics of chemical reactions and related processes is given. Section 1.2 introduces heterogeneous catalysis as an application and motivation for the treatment of reaction-diffusion systems. Finally, section 1.3 will further explore the treatment of RD systems.

1.1 Chemical reactions and their kinetics

Chemical reactions are processes that change chemical substances under the formation of new and/or breaking of old chemical bonds. As described in thermodynamics, they can be classified by the amount of energy they emit (exothermic) or absorb (endothermic); or whether they occur spontaneously (exergonic) or not (endergonic). Simple chemical reactions usually strive towards an equilibrium state, where no further change occurs.

The kinetics of chemical reactions are of utmost importance in various disciplines and applications. While these are influenced by thermodynamics, the behavior of coupled reactions far from equilibrium is less likely to be described only by thermodynamic considerations. The kinetics of different types of chemical reactions with the focus on heterogeneous catalysis are introduced in the following sections.

1.1.1 Reaction kinetics and chemical equilibrium

A chemical process, where the educt(s) E_i react to the product(s) P_j may be generalized as

$$\sum E_i \underset{k_{-r}}{\overset{k_r}{\rightleftharpoons}} \sum P_j \tag{1.1}$$

Two reactions are taking place at the same time, the reaction from E_i to P_j and the reverse reaction from P_j to E_i. Their reaction rate is proportional to the reaction constants k and the activity of the participating reactants, in the following marked by square brackets. The formation rate of P_j can be described by

$$r = \frac{\partial [P_j]}{\partial t} = k_r \cdot \prod [E_i] - k_{-r} \cdot \prod [P_i] \tag{1.2}$$

The reaction constants k_r and k_{-r} are governed by the Arrhenius equation and therefore dependent on the activation energy of the reaction E_a, the temperature T and the ideal gas constant R:

$$k_i = A_i \cdot e^{-\frac{E_a}{R \cdot T}} \tag{1.3}$$

In a closed system, the chemical process will strive to reach a chemical equilibrium, where both reaction steps (forth and back-reaction) occur at the same rate. The equilibrium is quantified by the equilibrium constant K and is defined by the following equation using the chemical activities of the reactants in equilibrium:

$$K = \frac{k_r}{k_{-r}} = \frac{\prod [P_i]}{\prod [E_i]} \tag{1.4}$$

In experimental conditions, the activities are often replaced with the concentrations of the observed reagents.

1.1.2 Transition state and catalysis

A catalyst is an agent, which accelerates a chemical process, that neither changes the reaction equilibrium nor is being depleted in the process. This means, that the acceleration is only a kinetic, not a thermodynamic effect. The presence of the catalyst speeds up both, back and forth reactions in such a way, that the chemical equilibrium is not shifted in any way; but reached in a shorter time.

The activation energies of both processes are decreased by the presence of the catalyst. The activation energy is usually dependent on the energetical stability of the transitional state (TS) of the process. A catalyst interacts stabilizing with the TS and therefore reduces the energy needed

to reach it. Otherwise, a catalyst could enable a different, energetically more favorable TS (see Figure 1.1)$_,$.

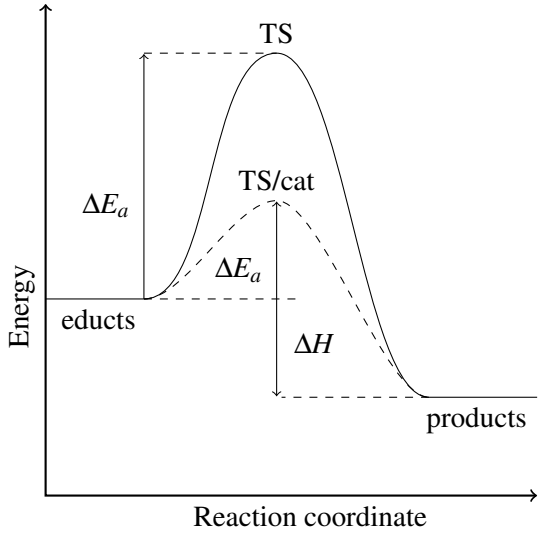


Figure 1.1: Schematic energy diagram for a chemical reaction with and without a catalyst.

Catalysis is divided into three fields: biocatalysis or enzymatic catalysis, homogeneous catalysis, and heterogeneous catalysis. A homogeneous catalyst has the same phase as the reagents. Frequent examples are acid-catalyzed reactions, where H^+ initializes the reaction but is replenished during the process. Biocatalysis can be seen as a special case of homogeneous catalysis, where the enzymes and educts are solved in the same phase. Enzymatic reactions are omnipresent in biological systems and show a rich diversity of (highly selective) reactions and complex kinetics.

In heterogeneous catalysis, the catalyst is of a different state of matter than the reactants. The interphase between catalyst and reactants plays a major role in the reaction kinetics. Therefore, understanding surface processes is fundamental for the comprehension of heterogeneous catalysis.

1.1.3 Adsorption

Adsorption can be described as the process of a fluid phase (reactants) interacting and binding on a static phase (i.e. a catalyst). Langmuir described it as a layer of condensed gas particles on the catalyst [1]. Interaction can occur via weaker van-der-Wals interactions (physisorption) or via strong chemical bonds (chemisorption). The latter is more important concerning catalysis since it severely changes the bonds of the adsorbate. If these are sufficiently weakened, dissociation can occur.

The driving force of adsorption is the minimization of surface energy. The surface particles of crystalline materials are strongly perturbated compared to the bulk particles in the regular lattice and have unsaturated valency compared to their subsurface neighbors. While surface reconstruction can reduce surface energy, adsorption is more effective in filling vacant positions. Adsorption can even induce surface reconstruction to reach higher coverages (see 1.2.2.1)

Therefore surface facets have a strong influence on adsorption behavior. More open planes

with larger free valency tend to have higher surface energy and exhibit larger adsorption energies [2, 3]. In addition, adsorbates prefer specific surface positions, which are more energetically favorable to them (e.g. [4]). The composition of these sites changes with the surface facet. Furthermore, they are not necessarily the same positions where dissociation occurs more likely. Surface defects are often found to act as activated sites for dissociation [5, 6].

The significance of adsorbate interactions on adsorption increases for higher surface coverages. They may be repulsive or attractive and can lead to the formation of ordered adsorbate overlayers. These self-organized patterns represent a coverage-dependent equilibrium between the adsorbate-adsorbate interactions, and adsorbate-adsorbent interactions (e.g. preferential adsorption sites) [7, 8]. For coadsorption processes (multiple different adsorbates), interaction can become quite complex (e.g. oxygen stabilizing water adsorption [9, 10], or hydrogen compressing the adsorbed oxygen overlayer [11])

In terms of kinetics, adsorption rates are proportional to the flux of particles Φ on the surface and the sticking coefficient s, which describes the probability that a given particle adsorbs. The sticking coefficient decreases with increasing surface coverage θ_r. A good approximation is a linear dependency for non-dissociative (one surface site is used, $n = 1$) and a square dependency for dissociative (two surface sites are used $n = 2$) adsorption.

The adsorption rate can then be defined as

$$r_{ads} = \Phi s_0 (1 - \theta_r)^n \tag{1.5}$$

Kinetically, desorption from the surface behaves analog to other chemical reactions (as described in section 1.1) and depends therefore mainly on temperature, the activation energy of desorption, and the activity of adsorbed particles.

However, for large coverages or significant intermolecular interactions on the surface, deviations from ideal behavior can occur. This can be treated e.g. by using a coverage-dependent constants ($A_i(\theta_r)$ and $E_a(\theta_r)$) in the Arrhenius equation (eq.(1.3)) [12].

1.1.4 Diffusion

Diffusion is a process deriving from the Brownian motion of individual particles. This results in a net flux of particles along concentration gradients.

The mathematical foundation of diffusion was first formulated by Adolf Fick, who was inspired by the experiments of Thomas Graham. Fick's law is analogous to the heat equation from Joseph Fourier [13].

In contrast to particles in the gas phase, which move nearly unrestricted, diffusion on surfaces is hindered by the attractive forces of the adsorbent. A periodic crystalline surface creates a periodic lattice of potential wells, which correspond to preferential adsorption sites discussed in the last section. These wells are separated by an energy barrier which diffusion has to overcome [3]. The diffusion of a single particle can then be illustrated by a hopping motion from one well to another.

Macroscopically, the effect of this discontinuous motion is not easily visible but can influence the behavior of adsorbates especially around defects or for large coverages. Nonetheless, a good

agreement can be achieved using Fick's law

$$\frac{\partial \theta}{\partial t} = D\frac{\partial^2 \theta}{\partial x^2} \tag{1.6}$$

utilizing an Arrhenius-type diffusion constant:

$$D = D_0 \cdot e^{-\frac{E_{\text{diff}}}{k \cdot T}} \tag{1.7}$$

1.2 Heterogeneous Catalysis

In heterogeneous catalysis, the catalyst is in a different phase than the reagents. In most cases, a solid, usually crystalline catalyst acts on liquid or gaseous reagents. Many metals and their oxides can act as catalysts, most prominently the so-called platinum group, consisting of ruthenium, rhodium, palladium, osmium, iridium, and platinum [14].

The following section glances over the historic significance of heterogeneous catalysis, in research and application, before exemplifying nonlinear reaction-diffusion dynamics on two studied surface reactions, namely the oxidation of CO and H_2.

1.2.1 Historic Impact

The catalytic effect was known long before industrialization. The first applications were probably in the form of fermentation [15] as a way of preserving and modifying food, producing e.g. alcohols, vinegar, and cheeses. Later, alchemists invented means of producing diethyl ether using sulfuric acid as a catalyst.

The modern concept of a catalyst as a substance that accelerates a chemical reaction without being consumed was first articulated by Elizabeth Fulhame [16], as she noticed the significance of water as *catalyst* in the reduction of soluted metal salts. Though a larger impact had the later systematic experiments of Berzelius, who coined the expression catalysis as well as shaped it into a distinct discipline, discussed by the likes of Faraday, Liebig, and Pasteur.

Around the turn of the 20^{th} century, recent development in thermodynamics and the introduction of statistical concepts provided the framework for the modern theory of catalysis. A large impact had the work of Wilhelm Ostwald, who was awarded the Nobel Prize in 1909 [15, 17].

The theoretical progress in catalysis was accompanied by important changes in technology and culture. Demand of bulk chemicals and crude oil products increased by magnitudes, especially with regard to the two world wars, and catalysis played a major role in satisfying it. The most famous example is the Haber-Bosch-Process [14], in which an iron-based catalyst accelerates the synthesis of ammonia from hydrogen and nitrogen gas. Fritz Haber studied the reaction and first noticed the industrial significance in his experiments. He developed a prototype reactor, for which he earned the Nobel Prize in 1918 [17]. At BASF, Alwin Mittasch tested about 2500 possible catalysts [18], while Carl Bosch made significant improvements to the process by developing high-pressure devices [15]. The latter received the Nobel Prize in 1931 [17].

The synthetic ammonia was mainly used in the synthesis of nitric acid (and thus explosives in World War I) and synthetic fertilizer. The new availability of cheap fertilizer (later) worldwide

improved crop yields and has been stated as a major factor to the population growth in the 20th century [19]. Still, 1% of the global energy consumption in 2005 was devoted to the production of ammonia (110 Mt) [20].

Before and during World War II, a major driving force of research was the transformation of crude oil and coal into needed fuels and chemical precursors. Catalysis improved basic reactions such as cracking, alkylation, and dehydrogenation of hydrocarbons in refineries. The Fischer-Tropsch synthesis was developed in 1938 [15, 21] as a means to 'liquify' coal into hydrocarbons for synthetic fuel, using alkalized iron as catalyst.

While being introduced during World War II, polymers became of large interest in the 1950s due to the popularity of polyethylene and nylon. A breakthrough in the catalytic mass production of polymers was made with the Ziegler-Natta polymerization which was acknowledged with the Nobel Price in 1963 [17, 22]. Originally, solved titanium tetrachloride acted as a catalytic complex and also allowed for rudimentary control over chirality [23].

In the following decades, the focus on the development of catalytic applications shifted from enabling mass production to treating environmental and safety-related issues. Demand for new catalysts increased with stricter regulation of sulfur content in fuels and pollutants in the exhaust gas of cars [24]. Under the label 'green chemistry', catalysis was introduced in new chemical industries (e.g. pharmaceutical industry, to prevent waste, reduce energy consumption, and replace hazardous chemicals [20]. Applications in energy technology such as fuel cells also gained more interest.

Despite these technical advances in these processes and widespread applications, many of the underlying reaction mechanisms could not be conclusively investigated until the 1980s. The first to improve upon the framework provided by Ostwald was Irving Langmuir, who was awarded the Nobel Price 1932[17] for his studies on surfaces and interphases. He proposed a layer of condensed gas on the surface to play a central role in catalytic surface reactions [25].

His idea was further developed by Sir Cyril Norman Hinshelwood [26, 27] (Nobel Price 1956 [17]). The Langmuir-Hinshelwood (LH) mechanism describes the catalytic surface reaction as a multi-stepped process, where only adsorbed particles on neighboring sites undergo reaction. Adsorption and desorption processes then contribute to the replenishment of reactants or surface sites. The process is described by the reaction equations (1.8 - 1.11) and illustrated in Figure 1.2 using CO oxidation as an example.

$$A_g \rightleftharpoons A_{ads} \tag{1.8}$$

$$B_g \rightleftharpoons B_{ads} \tag{1.9}$$

$$A_{ads} + B_{ads} \rightleftharpoons C_{ads} \tag{1.10}$$

$$C_{ads} \rightleftharpoons C_g \tag{1.11}$$

The most discussed alternative reaction pathway is the Eley-Rideal mechanism. Here, only one molecule of a bimolecular process adsorbs on the surface, while the other interacts directly out of the gas phase (shown in equations (1.12 - 1.14))

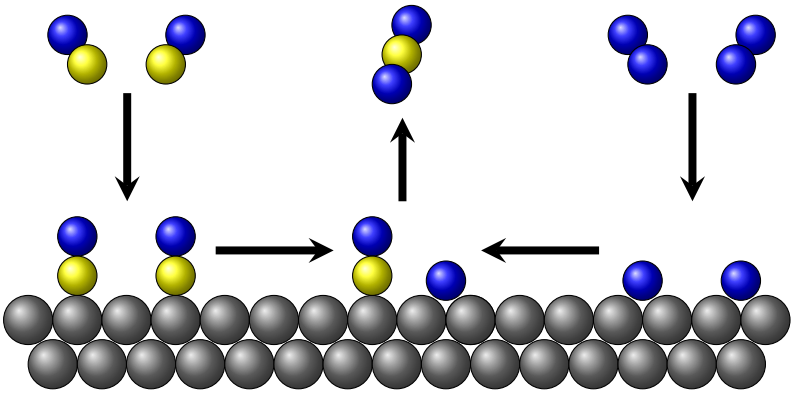


Figure 1.2: Schematic of the Langmuir-Hinshelwood mechanism for $CO + O_2$.

$$A_g \rightleftharpoons A_{ads} \tag{1.12}$$

$$A_{ads} + B_g \rightleftharpoons C_{ads} \tag{1.13}$$

$$C_{ads} \rightleftharpoons C_g \tag{1.14}$$

Insight in this debate was enabled by the advances in pumping and pressure control technology to the point of ultra-high vacuum (UHV). This allowed the setup of an extremely well-defined environment, where the catalytic process is controlled by the movement of single particles, instead of a fluid motion of gases. It also allowed the *in-situ* usage of mass spectrometry and other spectroscopic techniques, such as photoemission electron microscopy (PEEM).

The caveat of using these idealized, controlled conditions is the discrepancy to industrial applications, which tend to utilize high-pressure ($\geq$ 1 bar). The interactions between particles and multiple adsorption layers must be taken into account when extrapolating these findings for 'real' catalytic conditions. This difference is known as *pressure gap* [28]. This is analog to the *material gap* between idealized single-crystal catalysts experiments and real polycrystalline industrial used catalysts.

While the Eley-Rideal mechanism was found for some surface reactions, the group around Gerhard Ertl (Nobel Price 2006 [17]) conducted a conclusive series of experiments on single crystal, platinum group metal catalysts and demonstrated the Langmuir-Hinshelwood mechanism prevalent for many catalytic reactions [29–31].

This development sparked a new wave of interest in the field. Further insight in the surface processes lead to the finding of new surface phenomena (see section 1.2.2) and to improvement of descriptive models (see section 1.3).

1.2.2 Oxidation of carbon monoxide

The CO-oxidation is one of the most investigated catalytic reactions. This is certainly due to its large amount of applications, e.g. in the industrial purification of hydrogen gas or the reduction of CO content in exhaust gases. It became the prototypical monomer-dimer ($A + B_2$) reaction and was the first demonstrated [29] to occur according to the LH mechanism (equations (1.8 - 1.11)).

While the fundamental principle of the mechanism could be elucidated, later experiments showed more complex dynamics extending the predicted behavior.

1.2.2.1 Oscillations on Pt(100) and Pt(110)

Shortly after concluding LH as the reaction mechanism of CO oxidation on Pt with well-defined steady states, further UHV experiments found sustained oscillations for Pt(100) surface [32]. Ertl *et al.* proposed a first addition to the established mechanism based on surface reconstruction: clean Pt(100) exhibits a quasi-hexagonal (5×20)-reconstructed phase [33], while increasing CO-coverage over a critical value $\theta_{CO,cr}$ enforces a phase transition to a (1×1)-unreconstructed surface [34]. After removing the adsorbed CO, the unreconstructed surface is metastable below 350 K. The surface reconstructions and the energetics of the process are illustrated in Figure 1.3.

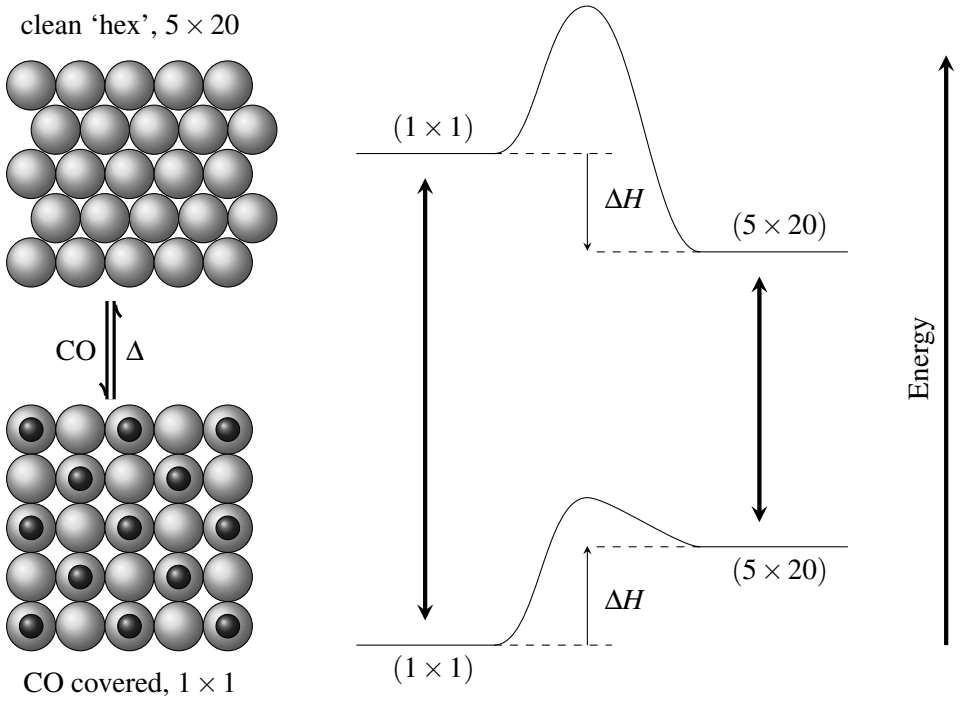


Figure 1.3: Schematic illustration of the surface reconstruction during the CO oxidation on Pt(100), and the energetics of this process, according to [3]. The clean surface exhibits far smaller sticking coefficients regarding O_2 [35]

Oscillations can therefore only start above 350 K as temperature suffices to overcome the activation barrier of reconstruction. Critical for the oscillatory behavior is the difference in oxygen adsorption, as the sticking coefficient increases on the (1×1) surface by orders of magnitude [35]. The increased oxygen availability leads to faster CO_2 formation and a decrease in CO coverage. The subsequent surface reconstruction to the 'hex' structure and reduction of the sticking coefficient restarts the oscillatory cycle.

Further experiments and theoretical analysis demonstrated this mechanism [36, 37] and illustrated its consequence in spatial dimension. A similar scheme was found for rate oscillations on Pt(110) surface investigated thoroughly by Eiswirth *et al.* (i.e. 38–41). The surface exhibits reconstruction between the relaxed (1×2) 'missing row' structure and the regular (1×1). Using

imaging techniques, rate oscillations could be correlated with spiral waves [42–44] which were further investigated concerning wave propagation.

1.2.2.2 Bistability on Ir(111)

While rate oscillations were intensely investigated on (100) and (110) surfaces, this phenomenon was seldom observed on (111) surfaces. On an ideal (111) facing crystal catalyst, the surface atoms are arranged in a regular hexagonal lattice. The surface is flat and reconstruction is strongly inhibited. Small observed oscillations can be attributed to surface defects, though the most common defect in an experiment may be steps, due to small misalignments in cutting the unit crystal (leading to a higher-order surface facet, e.g. (997)).

Without surface reconstruction, the system can exhibit bistability within a range of experimental parameters, meaning multiple stable steady rates could be found, e.g. on Ir(111) [45], Pt(111) [46]. This means, that for given experimental parameters, two different equilibrium states can be observed. Changing the reaction conditions to certain critical values can lead to a discontinuous, significant change.

In the case of the CO oxidation on Ir(111), both steady states form branches that differ qualitatively by their CO_2 formation rate, denoting them as 'upper rate branch' (UR) and 'lower rate branch' (LR). For instance, Wehner *et al.* [45] showed the bistable regime for $420\,K \leq T \leq 530\,K$. With high O_2 partial pressures, the upper rate branch is stable. With increasing CO partial pressure (under equal flow), a critical value is reached, where a transition to the LR branch occurs. To return to the UR branch, a different threshold has to be surpassed; the branches describe a hysteresis.

On a surface level, the steady states correlate with different surface occupations. Relative high oxygen coverage can be correlated with the UR branch, while the LR branch can be seen as reversibly poisoned by CO. This was firstly confirmed by modeling [47] and later shown with PEEM [48, 49]. PEEM experiments also clarified the spatial distribution of the surface species. The minority species forms small islands, the reaction occurs mainly at the boundaries of these islands. If a kinetic phase transition occurs from one branch to another, these islands grow, analog to crystal growth.

It has been shown possible, to enable phase transition utilizing external noise [50–52]. This could be relevant for industrial applications, where sudden catalyst poisoning could be attributed to small statistic occurrence of poison slowly forming sufficient large islands. Larger noise in the feed gas composition lead experimentally to faster island nucleation, thus accelerating the transition [52, 53].

1.2.3 Water formation

The water formation (or hydrogen oxidation) adds a layer of complexity to the previously discussed CO oxidation. While being one of the most basic reactions, it requires at least one intermediate surface species and is rich of possible reaction pathways [54, 55]. The exact reaction conditions and the surface catalyst decides the preferential pathway for the reaction.

The most recognized reaction model for platinum group metal consists of three elementary surface reactions, using only hydroxide as an intermediate species:

$$H_{ads} + O_{ads} \rightleftharpoons OH_{ads} \tag{1.15}$$

$$H_{ads} + OH_{ads} \rightleftharpoons H_2O \tag{1.16}$$

$$OH_{ads} + OH_{ads} \rightleftharpoons H_2O + O_{ads} \tag{1.17}$$

Depending on the reaction temperature and catalyst, the formed water may remain adsorbed, or desorbs quickly. Interestingly, on Pt(111) the reaction was found highly dependable on temperature. For low temperatures ($T < 170$ K) [3], an induction period was observed after which the reaction rapidly starts. Above this temperature, the reaction occurs spontaneously but shows an increase in its apparent activation energy.

An explanation for this was found in an autocatalytic mechanism that lowers the activation barrier below the desorption temperature of water. According to Anton and Cadogan [56], the synproportionation of water and oxygen on Pt(111) has a fairly low activation energy, in the same order of magnitude as water desorption. This enables a chain reaction, covering the surface increasingly with water.

Using STM [57–59], this chain reaction can be observed as chemical waves. The synproportionation takes place at the boundary of oxygen and water islands and formed hydroxide is consequently hydrogenated again to water. Thus, the water domains propagate into the oxygen covered regions. The moving reaction front can be described by a traveling wave [58], showing the significance of transport in the reaction mechanism. There is still debate about which step may be rate limiting [60] and for higher temperatures, the reaction is far less studied. Verheij and Hugenschmidt studied the reaction at conditions $T > 300$ K [61]. Their results were discussed under a model strongly dependent on surface defects as active sites. The recent work of Borodin *et al.* [62] added (non-conclusive) evidence for this conclusion. They realized a series of experiments which they fitted with a simplistic dummy model over a large temperature range. Comparison of their results and the complete reaction model utilizing DFT determined reaction parameters showed large deviations. Due to the large discrepancy between both models, they argued that this 'full-scheme' was still not complete, either because of missing elementary reaction steps, unidentified precursor molecules, or involved active sites.

1.3 Reaction-Diffusion Systems

The nonlinear phenomena shown in sections 1.2.2.1–1.2.3 exemplifies the wide variety of dynamics present in catalytic systems. A more general understanding of reaction-diffusion systems can help predicting the dynamics utilizing simpler equations. A generalized methodology may also be adapted to model a vide variety of subjects, such as neural networks [63], coupled reactors [64, 65] or chemical oscillators [66, 67], population growth [68], virus propagation [69] and formation of vegetational pattern [70].

In a reaction-diffusion system with n participating species, the vector ρ contains n elements which represent a measure of activity of that species. In regards to chemical reactions, ρ often contains the concentrations or partial pressures of the active species, while for surface reactions the

spatial concentrations of molecules or active sites are often used. In a typical reaction-diffusion-model, the terms treating transport and reaction are usually separated. The transport term describes the spatial distribution of the species ρ (as a vector) and is mainly responsible for pattern formation. In the case of diffusion, mostly Fick's law is implemented (see section 1.1.4), but sometimes modified to treat deviations such as non-constant diffusion or time-delayed response.

The reaction term in the equation system includes consumption and production of the species ρ, as well as external feed/discharge not covered by transport. It defines steady states of the system, which play an crucial role in spatiotemporal pattern formation. In 1D, a general reaction-diffusion system is expressed by the rate vector:

$$\frac{\partial \rho}{\partial t} = D\frac{\partial^2 \rho}{\partial x^2} + F(\rho), \tag{1.18}$$

where D is a Fickian diffusion constant and F is a function that reproduces the formation/consumption rates described by the model system. A shorter notation can be used to express the partial derivates:

$$\rho_t = D\rho_{xx} + F(\rho) \tag{1.19}$$

An equation of this type is called partial differential equation (pde), because it depends on derivation of multiple variables. An analytical treatment of pdes is not always possible (e.g. only under certain initial conditions). Here, since in this case both terms influence different properties of the system and are isolated in the equation, it is sensible to first analyze both parts separately to get an intuitive understanding.

1.3.1 Properties of the reaction term

Function F describes the reaction-system as defined by the applied model. It decides the number, type, and stability of steady states of the system. The mathematical expression consists of multiple terms in the form of eq. (1.2) and (1.5), leading (usually) to a nonlinear equation system. For instance, an example LH-model (based on CO+O_2/Ir(111) [52])

$$\begin{aligned} A_g + * &\rightleftharpoons A_{ads} \\ B_g + 2* &\rightarrow 2B_{ads} \\ A_{ads} + B_{ads} &\rightarrow C_g + * \end{aligned}$$

using asterics to denote free surface sites, gives the equation system:

$$\rho = \begin{pmatrix} \rho_A \\ \rho_B \\ \rho_* \end{pmatrix}; \frac{\partial \rho}{\partial t} = F(\rho) = \begin{pmatrix} \Phi_A s_A \rho_* - k_1 \rho_A - k_2 \rho_A \rho_B \\ 2\Phi_B s_B \rho_*^2 - k_2 \rho_A \rho_B \\ -\Phi_A s_A \rho_* - 2\Phi_B s_B \rho_*^2 + k_1 \rho_A + 2k_2 \rho_A \rho_B \end{pmatrix} \tag{1.20}$$

Φ_i depicts the constant gas flow of the precursors, s_i their sticking coefficients, k_1 the desorption constant of A and k_2 the reaction constant of $A+B$, both governed by the Arrhenius law (eq. (1.3)).

The third equation is a linear combination of the other two, thus additional boundary conditions are needed to solve the equation system analytically. Physically sensible constraints may be

applied, such as limiting concentrations to positive values, or applying conservation laws (e.g. a constant number of surface sites):

$$\rho_A + \rho_B + \rho_* = 1. \tag{1.21}$$

Using the conservation law (1.21) to express the surface sites, this model can be reduced to 2D:

$$\rho = \begin{pmatrix} \rho_A \\ \rho_B \end{pmatrix}; \frac{\partial \rho}{\partial t} = F(\rho) = \begin{pmatrix} \Phi_A s_A (1 - \rho_A - \rho_B) - k_1 \rho_A - k_2 \rho_A \rho_B \\ 2\Phi_B s_B (1 - \rho_A - \rho_B)^2 - k_2 \rho_A \rho_B \end{pmatrix} \tag{1.22}$$

This rate vector applied over all possible ρ constructs a vector field. An arbitrary initial condition ρ_0 changes in time according to the field. The curve $\rho(t)$ is called trajectory, the graphic illustration of multiple different trajectories is called phase diagram.

Points $\tilde{\rho}$ which fulfill the condition $\frac{\partial \rho}{\partial \tau} = 0$ are fixed points of the equation system. Steady states in reaction-diffusion systems are fixed points of the equation system. The differential equation system depicted in eq. (1.22) has one trivial ($\rho_A = 0$, $\rho_B = 1$) and three unique, possibly complex solutions. The terms are shown in appendix A and were computed utilizing Sympy [71] in Python. While an analytical solution is not always possible, such systems can be approached numerically (see section 1.3.3).

Different types of fixed points exist and can be classified by the behavior of the trajectories around them. One criterion is the stability of a fixed point. Trajectories strive away from unstable fixed points, and towards stable ones. From this observation, stricter formalized definitions as the Lyapunov stability can be defined:

If there exists an $\varepsilon > 0$, for that all trajectories ρ with $|\rho(t_0) - \tilde{\rho}| < \varepsilon$ then:

$$\tilde{\rho} \text{ is stable if } |\rho(t) - \tilde{\rho}| < \varepsilon \qquad \text{for all } t \geq t_0, \tag{1.23}$$

$$\tilde{\rho} \text{ is asymptotically stable if } \lim_{t \to \infty} \rho(t) = \tilde{\rho} \tag{1.24}$$

To study the stability of a fixed point, we investigate the behavior of trajectories around $\tilde{\rho}$. For the small perturbation $\delta(t) = \rho(t) - \tilde{\rho}$, the reaction system can be linearized:

$$\begin{aligned} \frac{\partial \delta}{\partial t} &= F(\delta) = F(\tilde{\rho}) + F'(\tilde{\rho})(\rho - \tilde{\rho}) + O(|\rho - \tilde{\rho}|^2) \\ &\approx J_{\tilde{\rho}} \delta \end{aligned} \tag{1.25}$$

Here, O depicts the nonlinear deviation, which vanishes as δ approaches zero and the derivative F' is the Jacobian matrix J at the fixed point $\tilde{\rho}$.

$$J_{i,j} = \left. \frac{\partial F_i}{\partial \rho_j} \right|_{\tilde{\rho}} \tag{1.26}$$

Solving eq. 1.25 gives a solution of the form $\delta(t) = Je^{(Jt)}$ which can be expressed by using the n eigenvalues λ_k and eigenvectors $\vec{\lambda}_k$ of J:

$$\delta(t) = \sum_{k=1}^{n} C_k \vec{\lambda}_k e^{(\lambda_k t)} \tag{1.27}$$

The constants C_k depend on the initial condition ρ_0. The behavior of $\delta(t)$ is strongly dependent on the eigenvalues. If all (real parts) of λ_k are negative, then δ approaches zero as time progresses, and thus $\tilde{\rho}$ is asymptotically stable. Otherwise, if at least one eigenvalue is positive, any small deviation δ increases exponentially, meaning $\tilde{\rho}$ is unstable. If none of these conditions are fulfilled (by having at least one zero eigenvalue), then the nonlinear deviations influence the fixed point stability. With complex eigenvalues, the imaginary part introduces rotation into the behavior of the trajectories.

In 2D, the Jacobian matrix

$$J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} \tag{1.28}$$

has the two eigenvalues:

$$\begin{aligned} \lambda_{1,2} &= \frac{1}{2}\left[(J_{11}+J_{22}) \pm \sqrt{(J_{11}+J_{22})^2 - 4(J_{11}J_{22}-J_{12}J_{21})}\right] \\ &= \frac{1}{2}\left[\mathrm{tr}J \pm \sqrt{(\mathrm{tr}J)^2 - 4(\det J)}\right] \end{aligned} \tag{1.29}$$

The trace and determinant of the Jacobian define five sections with different behavior, illustrated in Figure 1.4. Stable fixed points exist in the upper left quadrant of the diagram, while the other quadrants are unstable. The line defined by $(\mathrm{tr}J)^2 - 4(\det J)$ is the boundary for the existence of real eigenvalues. Point e) is a (asymptotically) stable node, while f an unstable node. b) and c) show fixed points whose Jacobian exhibits complex eigenvalues, they show (asymptotically) stable and unstable spiral points. A special case appears at the boundaries (see j), where the trajectories form stable orbits around the fixed point. In the lower quadrants (negative determinant), the fixed point can be described as saddle, as one eigenvalue is positive and the other is negative. The trajectories approach and leave the fixed point in different dimensions.

If multiple solutions $\tilde{\rho}$ exist, changing an experimental parameter of the model can change the stability of these solutions. As illustrated in Figure 1.4, a stable solution becomes unstable by changing sign of either trace or determinant of J. This induces a bifurcation, a phase transition to a different stable fixed point. The unstable solutions form a new branch that can again become stable (see Figure 1.6).

1.3.2 Properties of the diffusion term and synthesis

Diffusion, as described in section 1.1.4, is expected to behave as an equalizing force. Without reaction, any initial distribution ρ_0 should approach a constant value at every point in space.

Solving equation (1.6) (again depending on the initial and boundary conditions) has been accomplished in the same manner as the analog heat equation. Assuming the function $\rho(x,t)$ can

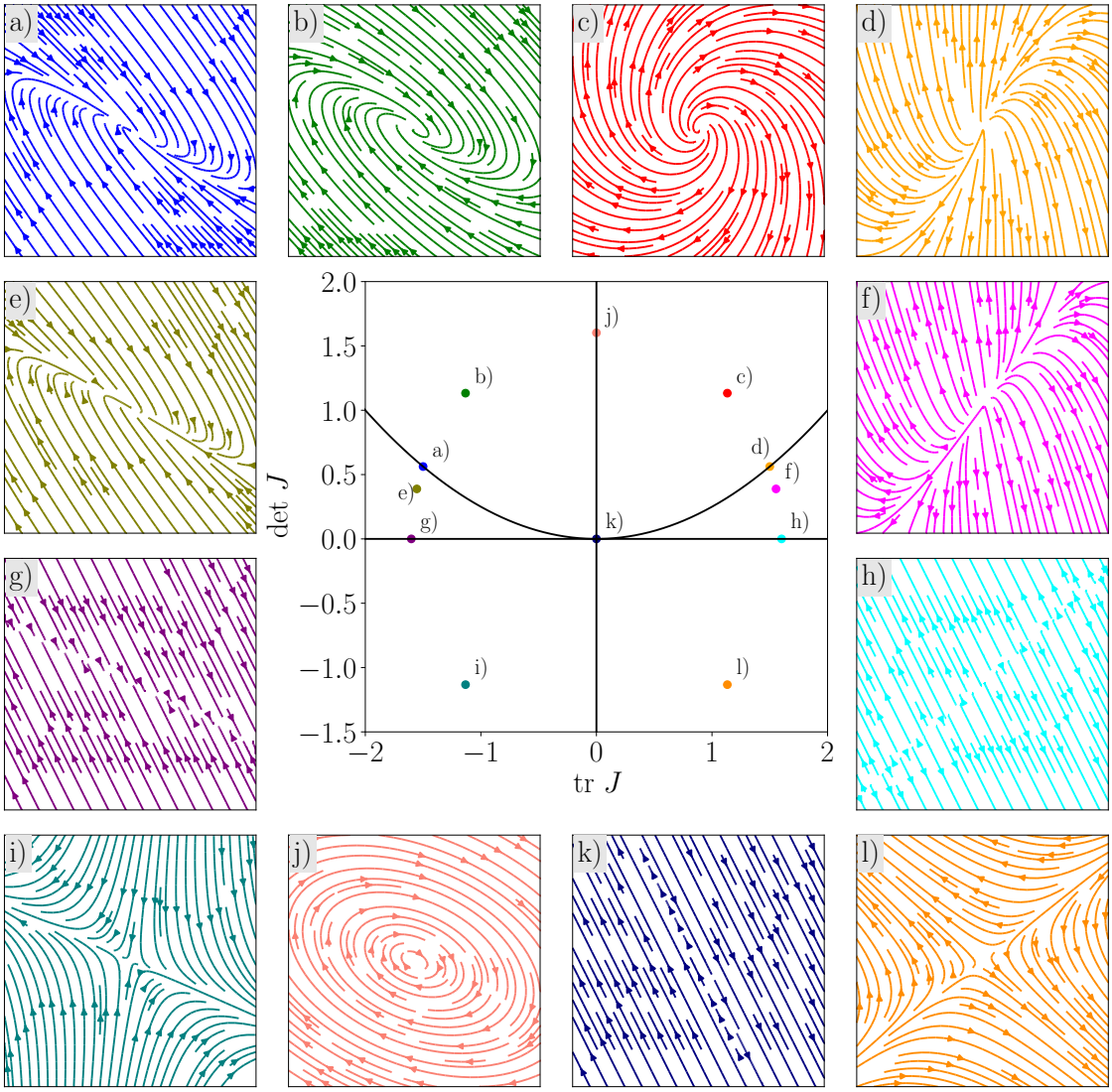


Figure 1.4: Illustration of the linearized stability of fixed points in 2D. The behavior of trajectories around the fixed point depends on the trace and the determinant of the Jacobian matrix. Stable fixed points are located in the upper left quadrant and are illustrated by a), b), and e).

be written as $\rho(x,t) = \nu(t)\mu(x)$, then

$$\frac{\partial(\nu(t)\mu(x))}{\partial t} = D\frac{\partial^2(\nu(t)\mu(x))}{\partial x^2} \tag{1.30}$$

$$\mu(x)\frac{\partial \nu(t)}{\partial t} = D\nu(t)\frac{\partial^2\mu(x)}{\partial x^2}$$

$$\frac{1}{\nu(t)}\frac{\partial \nu(t)}{\partial t} = \frac{D}{\mu(x)}\frac{\partial^2\mu(x)}{\partial x^2} \tag{1.31}$$

Both variables can be completely separated such that the left side of the equation only depends on t while the right side depends on x. This means they must both be equal to a constant value $\mathbf{C}$; resulting in two separate, and solvable ordinary differential equations:

$$\nu_t = \mathbf{C}\nu,\ D\mu_{xx} = \mathbf{C}\mu \tag{1.32}$$

Solving and utilizing $\mathbf{C} = c_0 D$ leads to the following terms:

$$v(t) = c_1 e^{c_0 D t} \tag{1.33}$$

$$\mu(x) = c_2 e^{\sqrt{c_0}x} + c_3 e^{-\sqrt{c_0}x} \tag{1.34}$$

$$\rho(x,t) = c_1 c_2 e^{\sqrt{c_0}x + c_0 D t} + c_1 c_3 e^{-\sqrt{c_0}x + c_0 D t} \tag{1.35}$$

with c_0, c_1, c_2, c_3 being constants, depending on the given initial and boundary conditions. An alternative solution for μ is in the form of a sine (or cosine) wave. The superposition of multiple wave functions is also a valid solution. These solutions fit with the intuition regarding diffusion. In time, an exponential decay towards equilibrium is expected (eq. (1.33)). A plausible prediction in space is a front that spreads between the boundaries.

Concerning boundary conditions regarding the boundary ∂L, common possibilities are Dirichlet conditions and Neumann conditions. For Dirichlet, ρ is fixed at the boundary (e.g. to a remarkable value; $\rho(\partial L, t) = f(\partial L, t)$). Neumann conditions limit the derivative at the boundaries instead (e.g. constant flux; $(\partial \rho(\partial L, t)/\partial n) = g(\partial L, t)$).

The introduction of the reaction term (eq. 1.18) increases the complexity of the equation. The interactions between steady states of the reaction system and diffusion can form traveling waves. The simplest reaction-term which exhibits a traveling wavefront is the logistic Fischer-Equation: $F(\rho) = r\rho(1-\rho)$ [72]. Since the differential equation is only analytically solvable for very basic F, numerical approaches are widespread. The most used are the *finite element method* (FEM) and the *finite difference method* (FDM). Both methods are based on the discretization of the involved dimensions, the latter being more intuitive and easier to implement but is restricted to rectangular geometries [73].

In FEM, the partial derivatives are replaced by the finite difference in discrete space and time. The coupling between the discrete spatial intervals can be expressed as a stencil. Using a discrete linear domain illustrated by N nodes with the distance h, equation (1.18) becomes

$$\frac{\partial \rho_n}{\partial t} = D \frac{\rho_{n-1} + \rho_{n+1} - 2\rho_n}{h^2} + F(\rho_n) \tag{1.36}$$

for any node $n \notin 1, N$.

The resulted equation system can be used directly in numerical integration as illustrated in the previous section. Two example fronts in time are shown in Figure 1.7 in the following section.

1.3.3 Example treatment

In this section, the reaction-diffusion model from equation (1.22) is further utilized as an example to illustrate the behavior of such systems. The reaction constants and parameters used in this section are summarized in Table 1.1, utilizing Y_A as an experimental parameter to change the temperature flow.

Trajectories can be obtained using numerical integration starting from different initial conditions. The integration or evolution of ρ is done iteratively, assuming linearized steps: Starting with an initial condition ρ_0 and an initial slope of $F_0 = F(\rho_0)$, after the small time step Δt, the concentrations shift to $\rho(\Delta t) = \rho_0 + F_0 \cdot \Delta t$. Thus, the function $\rho(t)$ can be computed utilizing

Table 1.1: Parameters utilized for numerical integration of the example model. The values are taken from [52, 74] as this is a variant model.

Parameter	value	unit	Parameter	value	unit
s_A	1	[-]	s_B	0.11	[-]
Φ_A	$0.878Y_A$	ML s^{-1}	Φ_B	$0.878(1-Y_A)$	ML s^{-1}
ν_1	10^{13}	s^{-1}	E_1	140	kJ mol^-1
ν_2	10^5	$(MLs)^{-1}$	E_2	40	kJ mol^-1
T	500	K	R	8.3145	J mol^{-1} K^{-1}

multiple iterations.

The result is of course only an approximation of the real trajectory. Δt governs the accuracy of the integration. Decreasing Δt generally improves the precision while increasing the number of steps to be calculated. This can lead to a significant decline in numerical accuracy, especially for small slopes.

On the other hand, too large time steps can result in missing the fine structure of the curve (e.g. a local minimum), branching out towards a different fixed point, or exhibiting instabilities (e.g. striving towards infinity). An equation is called *stiff*, if it is sensitive toward numerical instabilities. Sophisticated integration schemes (such as BDF [75]) rely on implicit (backward) integration methods [73] to avoid instabilities while trying to reduce the number of necessary iterations.

Example trajectories for multiple random initial conditions utilizing sensible Δt and two different parameters Y_A are illustrated in Figure 1.5. Both examples show convergence towards a steady state. In the first Figure, one stable fixed point seems to exist, while for Y_A, there are two stable fixed points. The behavior of the trajectory in the $\rho_A - \rho_B$-plane reveals two different phases of behavior. Firstly, a linear phase, where both ρ_A and ρ_B 'quickly' decrease in time. The trajectories fall into a so-called slow manifold [76]. Here, the convergence towards the steady states progresses slowly.

For equation systems which are not analytically solvable, numerical solutions can be obtained. In this case, the solutions are too complex for manual treatment, but can be obtained using libraries for symbolic computing such as the Sympy module in Python. Linear stability analysis can be done utilizing the Jacobian matrix:

$$J = -\begin{pmatrix} \Phi_A s_A + k_1 + k_2\tilde{\rho}_B & \Phi_A s_A + k_2\tilde{\rho}_A \\ 2\Phi_B s_B(1-\tilde{\rho}_A-\tilde{\rho}_B) + k_2\tilde{\rho}_B & 2\Phi_B s_B(1-\tilde{\rho}_A-\tilde{\rho}_B) + k_2\tilde{\rho}_A \end{pmatrix}. \tag{1.37}$$

Since all physical constants used in this model are real and positive, the trace of the matrix must be negative. With regard to Figure 1.4, any fixed point of this model is either (asymptotically) stable, or an unstable saddle, depending on the determinant. Omiting negative and complex solutions (imaginary part $> 10^{-8}$), Figure 1.6 can be obtained.

The bifurcation diagram shows exactly two branches of stable solutions connected by a branch of unstable ones. In the ternary diagram, the solutions form the slow manifold observed in Figure 1.5. The model captures the qualitative behavior of the CO oxidation on Ir(111) as discussed in section 1.2.2.2, even tough improvements to fit experimental observations have been done [52].

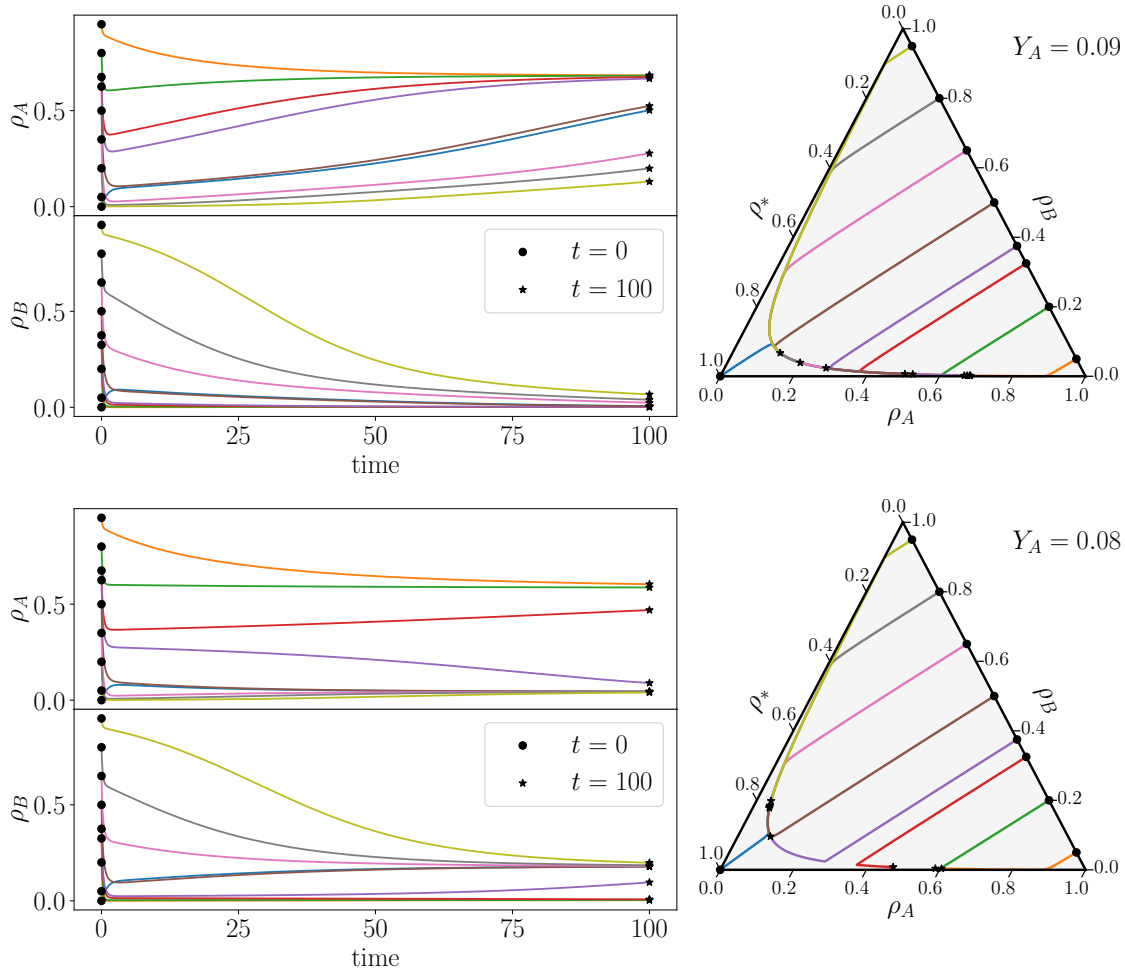


Figure 1.5: Example trajectories for different Y_A. The left side of the both illustrates the numerical integration for ρ_A in time, the right side shows the trajectory in the $\rho_A - \rho_B$-plane. Numerical integration is done using Python.

An example for reaction-diffusion fronts are shown in Figure 1.7, for two pairs of diffusion coefficients D_A, D_B. In both cases, the front connects the two stable states at $Y_A = 0.083$ (LR and UR, as illustrated by the dotted line). The first front propagates to the left, as the UR dominates due to the spreading of B into the LR domain. The opposite happens in the second example, as the LR slowly pushes into the UR domain.

Bibliography

[1] Langmuir I *The Adsorption of Gases on Plane Surfaces of Glass, Mica and Platinum*. (1918) *J. Am. Chem. Soc.* **40**, 1361–1403. doi: 10.1021/ja02242a004.

[2] Mittendorfer F, Seriani N, Dubay O, Kresse G *Morphology of Mesoscopic Rh and Pd Nanoparticles under Oxidizing Conditions*. (2007) *Phys. Rev. B* **76**, 233413. doi: 10.1103/PhysRevB.76.233413.

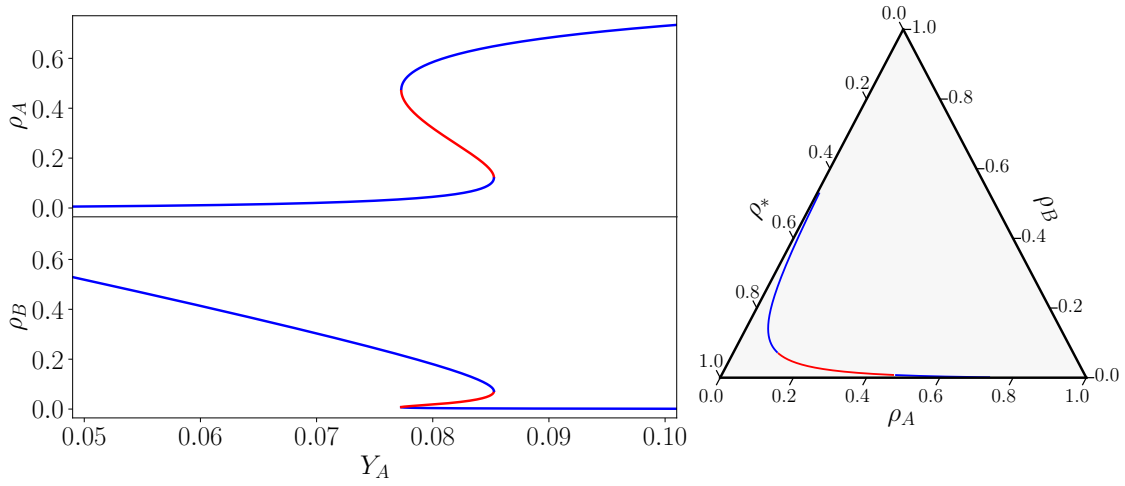


Figure 1.6: Codimension-1-bifurcation diagram of the example system shown for ρ_A as function of Y_A. There are two overlapping stable branches (blue) of steady states. They are connected by an unstable (red) saddle branch.

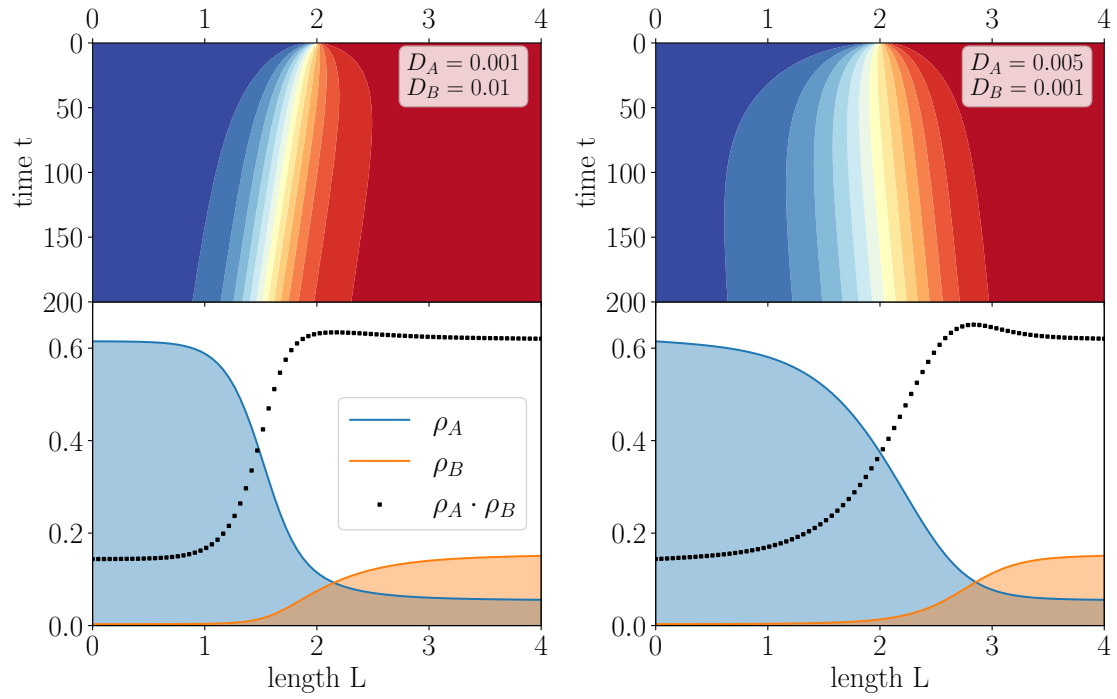


Figure 1.7: Illustration of the development of a reaction-diffusion front for the example model at $Y_A = 0.083$. The upper figures show ρ_A in space and time; the lower figures both fronts at $t = 200$. Higher D leads to larger front broadening. The direction of front propagation can change with the diffusion coefficients.

[3] Ertl G *Reactions at Solid Surfaces*. (2009) New Jersey: Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

[4] Ferrin P, Kandoi S, Nilekar AU, Mavrikakis M *Hydrogen Adsorption, Absorption and Diffusion on and in Transition Metal Surfaces: A DFT Study*. (2012) *Surf. Sci.* **606**, 679–689. doi: 10.1016/j.susc.2011.12.017.

[5] Hammer B *Bond Activation at Monatomic Steps: NO Dissociation at Corrugated Ru(0001)*. (1999) *Phys. Rev. Lett.* **83**, 3681–3684. doi: 10.1103/PhysRevLett.83.3681.

[6] Schaub R, Thostrup P, Lopez N, Lægsgaard E, Stensgaard I, Nørskov JK, Besenbacher F *Oxygen Vacancies as Active Sites for Water Dissociation on Rutile TiO_2(110)*. (2001) *Phys. Rev. Lett.* **87**, 266104. doi: 10.1103/PhysRevLett.87.266104.

[7] Schreiber F, Eberhardt A, Leung TYB, Schwartz P, Wetterer SM, Lavrich DJ, Berman L, Fenter P, Eisenberger P, Scoles G *Adsorption Mechanisms, Structures, and Growth Regimes of an Archetypal Self-Assembling System: Decanethiol on Au(111)*. (1998) *Phys. Rev. B* **57**, 12476–12481. doi: 10.1103/PhysRevB.57.12476.

[8] Nilius N, Rienks EDL, Rust HP, Freund HJ *Self-Organization of Gold Atoms on a Polar FeO(111) Surface*. (2005) *Phys. Rev. Lett.* **95**, 066101. doi: 10.1103/PhysRevLett.95.066101.

[9] Clay C, Cummings L, Hodgson A *Mixed Water/OH Structures on Pd(111)*. (2007) *Surf. Sci.* **601**, 562–568. doi: 10.1016/j.susc.2006.10.024.

[10] Shavorskiy A, Eralp T, Ataman E, Isvoranu C, Schnadt J, Andersen JN, Held G *Dissociation of Water on Oxygen-Covered Rh(111)*. (2009) *J. Chem. Phys.* **131**, 214707–214707. doi: 10.1063/1.3266941.

[11] Demchenko DO, Sacha GM, Salmeron M, Wang LW *Interactions of Oxygen and Hydrogen on Pd(111) Surface*. (2008) *Surf. Sci.* **602**, 2552–2557. doi: 10.1016/j.susc.2008.06.004.

[12] Getman RB, Schneider WF *DFT-Based Coverage-Dependent Model of Pt-Catalyzed NO Oxidation*. (2010) *ChemCatChem* **2**, 1450–1460. doi: 10.1002/cctc.201000146.

[13] Narasimhan TN *Fourier's Heat Conduction Equation: History, Influence, and Connections*. (1999) *Rev. Geophys.* **37**, 151–172. doi: 10.1029/1998RG900006.

[14] Cornils B, Herrmann WA, Wong CH, Zanthoff HW *Catalysis from A to Z: A Concise Encyclopedia*. (2013) Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA fourth, completely revised and enlarged edition edition.

[15] Ertl G, Knözinger H, Weitkamp J *Handbook of Heterogeneous Catalysis*. (1997) Wiley first edition.

[16] Mellor JW *History of the Water Problem*. (1903) *J. Phys. Chem.* **7**, 557–567. doi: 10.1021/j150053a001.

[17] *All Nobel Prizes in Chemistry*. www.nobelprize.org/prizes/lists/all-nobel-prizes-in-chemistry. (accessed 2022-04-23).

[18] Farber E *From Chemistry to Philosophy: The Way of Alwin Mittasch (1869-1953)*. (1966) *Chymia* **11**, 157–178. doi: 10.2307/27757266.

[19] Smil V *Detonator of the Population Explosion*. (1999) *Nature* **400**, 415–415. doi: 10.1038/22672.

[20] Rothenberg G *Catalysis: Concepts and Green Applications*. (2008) Weinheim: Wiley-VCH.

[21] Schulz H *Short History and Present Trends of Fischer–Tropsch Synthesis*. (1999) *Appl. Catal. A: Gen.* **186**, 3–12. doi: 10.1016/S0926-860X(99)00160-X.

[22] Claverie JP, Schaper F *Ziegler-Natta Catalysis: 50 Years after the Nobel Prize*. (2013) *MRS Bull.* **38**, 213–218. doi: 10.1557/mrs.2013.52.

[23] Sinn H, Kaminsky W *Ziegler-Natta Catalysis*. (1980) In *Advances in Organometallic Chemistry* (Eds.: Stone F, West R), vol. 18 pp. 99–149. Academic Press.

[24] Berg W *Legislation for the Reduction of Exhaust Gas Emissions*. (2003) In *Traffic and Environment: Traffic and Environment* (Eds.: Gruden D), The Handbook of Environmental Chemistry pp. 175–253. Berlin, Heidelberg: Springer.

[25] Langmuir I *The Mechanism of the Catalytic Action of Platinum in the Reactions* $2CO+O_2 \rightleftharpoons CO_2$ *and* $2H_2+O_2 \rightleftharpoons 2H_2O$. (1922) *Trans. Faraday Soc.* **17**, 621–654. doi: 10.1039/tf9221700621.

[26] Prichard CR, Hinshelwood CN *CXIV.—The Interaction of Hydrogen and Carbon Dioxide on the Surface of Platinum*. (1925) *J. Chem. Soc., Trans.* **127**, 806–811. doi: 10.1039/CT9252700806.

[27] Hinshelwood CN, Prichard CR *CCVI.—The Interaction of Carbon Dioxide and Hydrogen on the Surface of Tungsten*. (1925) *J. Chem. Soc., Trans.* **127**, 1546–1552. doi: 10.1039/CT9252701546.

[28] Freund HJ, Kuhlenbeck H, Libuda J, Rupprechter G, Baumer M, Hamann H *Bridging the Pressure and Materials Gaps between Catalysis and Surface Science: Clean and Modified Oxide Surfaces*. (2001) *Top. Catal.* **15**, 201–209. doi: 10.1023/A:1016686322301.

[29] Engel T *A Molecular Beam Investigation of He, CO, and* O_2 *Scattering from Pd(111)*. (1978) *J. Chem. Phys.* **69**, 373. doi: 10.1063/1.436363.

[30] Engel T, Ertl G *Elementary Steps in the Catalytic Oxidation of Carbon Monoxide on Platinum Metals*. (1979) *Adv. Catal.* **28**, 1–78. doi: 10.1016/S0360-0564(08)60133-9.

[31] Campbell CT, Ertl G, Kuipers H, Segner J *A Molecular Beam Study of the Catalytic Oxidation of CO on a Pt(111) Surface*. (1980) *J. Chem. Phys.* **73**, 5862–5873. doi: 10.1063/1.440029.

[32] Ertl G, Norton PR, Rüstig J *Kinetic Oscillations in the Platinum-catalyzed Oxidation of CO*. (1982) *Phys. Rev. Lett.* **49**, 177–180. doi: 10.1103/PhysRevLett.49.177.

[33] Van Hove M, Koestner R, Stair P, Bibérian J, Kesmodel L, Bartoš I, Somorjai G *The Surface Reconstructions of the (100) Crystal Faces of Iridium, Platinum and Gold*. (1981) *Surf. Sci.* **103**, 218–238. doi: 10.1016/0039-6028(81)90108-4.

[34] Norton P, Da Vies J, Creber D, Sitter C, Jackman T *The Pt(100) (5×20)⇋(1×1) Phase Transition: A Study by Rutherford Backscattering, Nuclear Microanalysis, LEED and Thermal Desorption Spectroscopy*. (1981) *Surf. Sci.* **108**, 205–224. doi: 10.1016/0039-6028(81)90445-3.

[35] Barteau MA, Ko EI, Madix RJ *The Adsorption of CO, O_2, and H_2 on Pt(100)-(5×20)*. (1981) *Surf. Sci.* **102**, 99–117. doi: 10.1016/0039-6028(81)90310-1.

[36] Imbihl R, Cox MP, Ertl G, Müller H, Brenig W *Kinetic Oscillations in the Catalytic CO Oxidation on Pt(100): Theory*. (1985) *J. Chem. Phys.* **83**, 1578–1587. doi: 10.1063/1.449834.

[37] Imbihl R, Cox MP, Ertl G *Kinetic Oscillations in the Catalytic CO Oxidation on Pt(100): Experiments*. (1986) *J. Chem. Phys.* **84**, 3519–3534. doi: 10.1063/1.450238.

[38] Eiswirth RM, Ertl G *Kinetic Oscillations in the Catalytic CO Oxidation on a Pt(110) Surface*. (1986) *Surf. Sci.* **177**, 90–100. doi: 10.1016/0039-6028(86)90259-1.

[39] Eiswirth RM, Krischer K, Ertl G *Transition to Chaos in an Oscillating Surface Reaction*. (1988) *Surf. Sci.* **202**, 565–591. doi: 10.1016/0039-6028(88)90053-2.

[40] Eiswirth M, Möller P, Wetzl K, Imbihl R, Ertl G *Mechanisms of Spatial Self-Organization in Isothermal Kinetic Oscillations during the Catalytic CO Oxidation on Pt Single Crystal Surfaces*. (1989) *J. Chem. Phys.* **90**, 510–521. doi: 10.1063/1.456501.

[41] Eiswirth RM, Müller P, Ertl G *Periodic Perturbations of the Oscillatory CO Oxidation on Pt(110)*. (1989) *Surf. Sci.* **208**, 13–33. doi: 10.1016/0039-6028(89)90034-4.

[42] Ertl G *Oscillatory Kinetics and Spatio-Temporal Self-Organization in Reactions at Solid Surfaces*. (1991) *Science* **254**, 1750–1755. doi: 10.1126/science.254.5039.1750.

[43] Nettesheim S, von Oertzen A, Rotermund HH, Ertl G *Reaction Diffusion Patterns in the Catalytic CO-oxidation on Pt(110): Front Propagation and Spiral Waves*. (1993) *J. Chem. Phys.* **98**, 9977–9985. doi: 10.1063/1.464323.

[44] Ehsasi M *Application of Photoemission Electron Microscopy to the Study of Surface Reactions*. (1994) *Appl. Surf. Sci.* **76–77**, 89–100. doi: 10.1016/0169-4332(94)90328-X.

[45] Wehner S, Baumann F, Ruckdeschel M, Küppers J *Kinetic Phase Transitions in the Reaction CO+O→CO_2 on Ir(111) Surfaces*. (2003) *J. Chem. Phys.* **119**, 6823–6831. doi: 10.1063/1.1603212.

[46] Berdau M, Yelenin GG, Karpowicz A, Ehsasi M, Christmann K, Block JH *Macroscopic and Mesoscopic Characterization of a Bistable Reaction System: CO Oxidation on Pt(111) Surface*. (1999) *J. Chem. Phys.* **110**, 11551–11573. doi: 10.1063/1.479097.

[47] Wehner S, Baumann F, Küppers J *Kinetic Hysteresis in the CO Oxidation Reaction on Ir(111) Surfaces*. (2003) *Chem. Phys. Lett.* **370**, 126–131. doi: 10.1016/S0009-2614(03)00059-9.

[48] Wehner S, Hoffmann P, Schmeißer D, Brand HR, Küppers J *Spatiotemporal Patterns of External Noise-Induced Transitions in a Bistable Reaction-Diffusion System: Photoelectron Emission Microscopy Experiments and Modeling*. (2005) *Phys. Rev. Lett.* **95**, 1–4. doi: 10.1103/PhysRevLett.95.038301.

[49] Wehner S, Hoffmann P, Schmeisser D, Brand HR, Küppers J *Influence of the Substrate on the Pattern Formation of a Surface Reaction*. (2007) *AIP Conf. Proc.* **913**, 121–126. doi: 10.1063/1.2746735.

[50] Hayase Y, Wehner S, Küppers J, Brand HR *External Noise Imposed on the Reaction-Diffusion System $CO+O_2 \rightarrow CO_2$ on Ir(111) Surfaces: Experiment and Theory*. (2004) *Phys. Rev. E* **69**, 1–15. doi: 10.1103/PhysRevE.69.021609.

[51] Wehner S, Hayase Y, Brand HR, Küppers J *Multiplicative Temperature Noise Applied to a Bistable Surface Reaction: Experiment and Theory*. (2004) *J. Phys. Chem. B* **108**, 14452–14461. doi: 10.1021/jp0497155.

[52] Wehner S *The CO-Oxidation Reaction on Ir(111) Surfaces: Bistability, Noise and Spatio-Temporal Patterns in Experiment and Modeling*. (2009) *Int. J. Bifurc. Chaos Appl. Sci. Eng.* **19**, 2637–2675. doi: 10.1142/S0218127409024384.

[53] Wehner S, Karpitschka S, Burkov Y, Schmeisser D, Küppers J, Brand HR *Stochastic Aspects of Pattern Formation during the Catalytic Oxidation of CO on Pd(111) Surfaces*. (2010) *Physica D* **239**, 746–751. doi: 10.1016/j.physd.2009.06.010.

[54] Qi L, Yu J, Li J *Coverage Dependence and Hydroperoxyl-Mediated Pathway of Catalytic Water Formation on Pt(111) Surface*. (2006) *J. Chem. Phys.* **125**, 054701–054701. doi: 10.1063/1.2227388.

[55] Ford DC, Nilekar AU, Xu Y, Mavrikakis M *Partial and Complete Reduction of O_2 by Hydrogen on Transition Metal Surfaces*. (2010) *Surf. Sci.* **604**, 1565–1575. doi: 10.1016/j.susc.2010.05.026.

[56] Anton AB, Cadogan DC *The Mechanism and Kinetics of Water Formation on Pt(111)*. (1990) *Surf. Sci. Lett.* **239**, 548–560. doi: 10.1016/0039-6028(90)90217-V.

[57] Völkening S, Bedürftig K, Jacobi K, Wintterlin J, Ertl G *Dual-Path Mechanism for Catalytic Oxidation of Hydrogen on Platinum Surfaces*. (1999) *Phys. Rev. Lett.* **83**, 2672–2675. doi: 10.1103/PhysRevLett.83.2672.

[58] Sachs C, Hildebrand M, Völkening S, Wintterlin J, Ertl G *Spatiotemporal Self-Organization in a Surface Reaction: From the Atomic to the Mesoscopic Scale*. (2001) *Science* **293**, 1635–1638. doi: 10.1126/science.1062883.

[59] Sachs C, Hildebrand M, Völkening S, Wintterlin J, Ertl G *Reaction Fronts in the Oxidation of Hydrogen on Pt(111): Scanning Tunneling Microscopy Experiments and Reaction-Diffusion Modeling*. (2002) *J. Chem. Phys.* **116**, 5759–5773. doi: 10.1063/1.1453964.

[60] Näslund LÅ *Reaction Kinetics for the Oxygen Hydrogenation Process on Pt(111) Derived from Temperature-Programmed XPS*. (2013) *Surf. Sci.* **618**, 42–48. doi: 10.1016/j.susc.2013.08.018.

[61] Verheij LK, Hugenschmidt MB *On the Mechanism of the Hydrogen - Oxygen Reaction on Pt(111)*. (1998) *Surf. Sci.* **416**, 37–58. doi: 10.1016/S0039-6028(98)00498-1.

[62] Borodin D, Schwarzer M, Hahn HW, Fingerhut J, Wang Y, Auerbach DJ, Guo H, Schroeder J, Kitsopoulos TN, Wodtke AM *The Puzzle of Rapid Hydrogen Oxidation on Pt(111)*. (2021) *Mol. Phys.* **119**, e1966533. doi: 10.1080/00268976.2021.1966533.

[63] Wang JL, Wu HN, Huang T, Ren SY, Wu J *Pinning Control for Synchronization of Coupled Reaction-Diffusion Neural Networks With Directed Topologies*. (2016) *IEEE Trans. Syst. Man Cybern, Syst.* **46**, 1109–1120. doi: 10.1109/TSMC.2015.2476491.

[64] Laplante JP, Erneux T *Propagation Failure in Arrays of Coupled Bistable Chemical Reactors*. (1992) *J. Phys. Chem.* **96**, 4931–4934. doi: 10.1021/j100191a038.

[65] Booth V, Erneux T, Laplante JP *Experimental and Numerical Study of Weakly Coupled Bistable Chemical Reactors*. (1994) *J. Phys. Chem.* **98**, 6537–6540. doi: 10.1021/j100077a019.

[66] Krug HJ, Pohlmann L, Kuhnert L *Analysis of the Modified Complete Oregonator Accounting for Oxygen Sensitivity and Photosensitivity of Belousov-Zhabotinskii Systems*. (1990) *J. Phys. Chem.* **94**, 4862–4866. doi: 10.1021/j100375a021.

[67] Nolet FE, Rombouts J, Gelens L *Synchronization in Reaction-Diffusion Systems with Multiple Pacemakers*. (2020) *Chaos* **30**, 053139–053139. doi: 10.1063/5.0002251.

[68] Liang J, Yan Q, Xiang C, Tang S *A Reaction-Diffusion Population Growth Equation with Multiple Pulse Perturbations*. (2019) *Commun. Nonlinear Sci. Numer. Simul.* **74**, 122–137. doi: 10.1016/j.cnsns.2019.02.015.

[69] Mammeri Y *A Reaction-Diffusion System to Better Comprehend the Unlockdown: Application of SEIR-type Model with Diffusion to the Spatial Spread of COVID-19 in France*. (2020) *Comput. Math. Biophys.* **8**, 102–113. doi: 10.1515/cmb-2020-0104.

[70] Fernandez-Oto C, Escaff D, Cisternas J *Spiral Vegetation Patterns in High-Altitude Wetlands*. (2019) *Ecol. Complex.* **37**, 38–46. doi: 10.1016/j.ecocom.2018.12.003.

[71] Meurer A, Smith CP, Paprocki M, Čertík O, Kirpichev SB, Rocklin M, Kumar A, Ivanov S, Moore JK, Singh S, Rathnayake T, Vig S, Granger BE, Muller RP, Bonazzi F, Gupta H, Vats S, Johansson F, Pedregosa F, Curry MJ, Terrel AR, Roučka Š, Saboo A, Fernando I, Kulal S, Cimrman R, Scopatz A *SymPy: Symbolic Computing in Python*. (2017) *PeerJ Comput. Sci.* **3**, e103. doi: 10.7717/peerj-cs.103.

[72] Méndez V, Fedotov S, Horsthemke W *Reaction–Transport Systems*. (2010) Springer Series in Synergetics. Berlin, Heidelberg: Springer Berlin Heidelberg.

[73] Goudon T *Mathematics for Modeling and Scientific Computing*. (2016) Hoboken, NJ, USA: John Wiley & Sons, Inc.

[74] Cisternas J, Wehner S, Descalzi O *CO Oxidation on Ir(111) Surfaces under Large Non-Gaussian Noise*. (2012) *J. Chem. Phys.* **137**, 064105–064105. doi: 10.1063/1.4742191.

[75] Byrne GD, Hindmarsh AC *A Polyalgorithm for the Numerical Solution of Ordinary Differential Equations*. (1975) *ACM Trans. Math. Softw.* **1**, 71–96. doi: 10.1145/355626.355636.

[76] Cisternas J, Escaff D, Descalzi O, Wehner S *Stochastic Model Calculation for the Carbon Monoxide Oxidation on Iridium(111) Surfaces*. (2009) *Int. J. Bifurc. Chaos Appl. Sci. Eng.* **19**, 3461–3472. doi: 10.1142/S0218127409024906.

2

Competing ternary surface reaction $CO+O_2+H_2$ on Ir(111)

Kevin Rohe, Jaime Cisternas, Stefan Wehner

2020 *Proc. R. Soc. A.* **476**, 20190712
doi: 10.1098/rspa.2019.0712

The CO oxidation on Platinum-group metals under ultra-high-vacuum conditions is one of the most studied surface reactions. However, the presence of disturbing species and competing reactions are often neglected. One of the most interesting additional gases to be treated is hydrogen, due to its importance in technical applications and its inevitability under vacuum conditions. Adding hydrogen to the reaction of CO and O_2 leads to more adsorbed species and competing reaction steps towards water formation.

In this study, a model for approaching the competing surface reactions $CO+O_2+H_2$ is presented and discussed. Using the framework of bifurcation theory, we show how the steady states of the extended system correspond to a swallowtail catastrophe set with a tristable regime within the swallowtail. We explore numerically the possibility of reaching all stable states and illustrate the experimental challenges such a system could pose. Lastly, an approximative first-principle approach to diffusion illustrates how up to three stable states balance each other while forming heterogeneous patterns.

2.1 Introduction

The development of the understanding of surface reactions during the last century is an interplay between unexpected observations, experiments, and modelling. For example adsorption of thin films was first described by Langmuir [1]; hydrogen exchange was analyzed by Eley and Rideal [2]; the Langmuir-Hinshelwood mechanism was found after extended experiments by Ertl's group [3, 4]; and the possibility of oscillating rates was first observed by Norton and Rüstig [5] and described in the following by Krischer, Eiswirth and Ertl [6–8].

A theoretical approach leads to experimental observation, while vice versa predictions from models initiate new experiments. This interplay has enabled the stepwise progress behind the present picture of surface reactions.

In studies on CO oxidation on Iridium, this combination has also been the case: experiments about, e.g. phase transition and rate hysteresis [9, 10] or noise [11, 12] were followed by detailed modelling; and modelling of the spatiotemporal pattern on such a non-reconstructing catalytic surface [11] was followed by experiments [12–14]. CO oxidation is the prototype reaction for a monomer-dimer surface reaction. It was for example intensively analysed on the non-reconstructing surfaces of the platinum group metals: Platinum [15], Rhodium [16, 17], Palladium [3, 18–21] and Iridium [22–28]. Water formation is the prototype for a dimer-dimer surface reaction. It was studied for example by Imbihl's group experimentally [29, 30] and Zhdanov numerically [31].

The differences between real-world applications and ideal ultra-high-vacuum (UHV) experiments are known as *pressure gap* [32] (atmospheric pressures vs. UHV) and *material gap* [33] (supported catalysts vs. single crystal). A third difference is the quality of the feed gas in UHV experiments, considering the use of highly pure gases and most accurate dosing. This discrepancy could be referred to as '*quality gap*'. One way to address the limited precision of real applications is to add noise [11, 12, 34]. Another way is the acceptance of impurities. Although even the purest commercially available pressurized gas cylinders contain oxygen as 99.99990%, CO as 99.7%, and hydrogen as 99.99990%, in the industrial production of H_2 the presence of impurities is a lot higher. Therefore for addressing impurities, another gas has to be added in a controlled manner to produce a ternary system with competing reactions. For high pressures, this was done by Piccolo *et al.* [35, 36], but never for the better controllable UHV experiments.

To address the presence of hydrogen as a competing surface species, new reaction paths have to be introduced. The focus of this study is, therefore, the modelling of the reaction system and the analysis of the numerical results. As we will show, the new steady states of the system can be studied using bifurcation theory leading to a codimension-2 diagram that features a 'swallowtail' shape.

The structure of the article is as follows: in section II we present the model of the competing reactions and its mathematical form; in section III we use bifurcation analysis for studying the new steady states and propose a scanning protocol for exploring the reaction rates in future experiments and visit all stable states by appropriate manipulation of two control parameters. The presence of spatial patterns is also considered. Finally, we present our conclusions.

2.2 Modelling

Numerous different models for the oxidation of carbon monoxide or hydrogen have been reported in previous studies, e.g. by Bär *et al.* [37] for CO and by Zhdanov [31] for H_2. The herewith presented model was based on our previous studies [26, 27] about CO oxidation on Ir(111). The necessary reaction steps for treating the water formation reaction are discussed in this section.

Hydrogen adsorption will be treated analogously to CO and O_2 adsorption with the sticking coefficient s_H and an exponential factor b. Hydrogen adsorption occurs dissociatively on Ir(111), as does oxygen. Hydrogen desorption, however, occurs at lower temperatures [38] and therefore has to be included (k_3).

Catalytic water formation is a two-step mechanism, unlike the one-step CO oxidation. The first reaction step is the oxidation of a single adsorbed hydrogen atom, resulting in adsorbed hydroxide OH_{ads}. In this step, H_{ads} competes with CO_{ads} for O_{ads}. Formation of water from hydroxide can occur by two different reactions, depending on temperature and occupation. Consequently, we include both the consecutive hydrogenation and the disproportionation of hydroxide in the model.

All reaction steps are potentially reversible under certain conditions. High-temperature processes such as hydroxide desorption [39] were not considered. The reaction rate for each step is defined by either a reaction coefficient k_i or a sticking coefficient (s_H, s_O, s_{CO}).
This results in a total of eleven necessary reactions for the competing oxidations of CO and H_2 on iridium:

$$
\begin{aligned}
O_{2,gas} &\overset{s_O}{\rightleftharpoons} 2O_{ads} \\
CO_{gas} &\underset{k_1}{\overset{s_{CO}}{\rightleftharpoons}} CO_{ads} \\
CO_{ads} + O_{ads} &\xrightarrow{k_2} CO_{2,gas} \\
H_{2,gas} &\underset{k_3}{\overset{s_H}{\rightleftharpoons}} 2H_{ads} \\
H_{ads} + O_{ads} &\underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}} OH_{ads} \\
OH_{ads} + H_{ads} &\underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}} H_2O_{ads} \\
2OH_{ads} &\underset{k_{-6}}{\overset{k_6}{\rightleftharpoons}} H_2O_{ads} + O_{ads} \\
H_2O_{ads} &\xrightarrow[k_{-7}]{k_7} H_2O_{gas}
\end{aligned}
$$

Since the reaction constants have a large uncertainty, a reduced model with fewer equations is preferable. Firstly, slow reactions at 500 K can be neglected, thus omitting the back-reaction of all three hydrogen oxidation steps ($k_{-4} = k_{-5} = k_{-6} = 0$) [39, 40]. Hydroxide disproportioning (k_6) occurs also slowly at such high temperatures, could be important in case of high OH occupation. Secondly, water desorption occurs at far lower temperatures [41] and can be seen as instantaneous at 500 K ($k_{-7} = \infty$). For this reason, re-adsorption of water is also prohibited. Hence, the reduced model consists of only nine reaction steps for the competing reaction between hydrogen, carbon

monoxide and oxygen.

$$\mathrm{O_{2,gas}} \xrightarrow{s_\mathrm{O}} 2\mathrm{O_{ads}} \tag{2.1}$$

$$\mathrm{CO_{gas}} \underset{k_1}{\overset{s_\mathrm{CO}}{\rightleftharpoons}} \mathrm{CO_{ads}} \tag{2.2}$$

$$\mathrm{CO_{ads}} + \mathrm{O_{ads}} \xrightarrow{k_2} \mathrm{CO_{2,gas}} \tag{2.3}$$

$$\mathrm{H_{2,gas}} \underset{k_3}{\overset{s_\mathrm{H}}{\rightleftharpoons}} 2\mathrm{H_{ads}} \tag{2.4}$$

$$\mathrm{H_{ads}} + \mathrm{O_{ads}} \xrightarrow{k_4} \mathrm{OH_{ads}} \tag{2.5}$$

$$\mathrm{OH_{ads}} + \mathrm{H_{ads}} \xrightarrow{k_5} \mathrm{H_2O_{gas}} \tag{2.6}$$

$$2\mathrm{OH_{ads}} \xrightarrow{k_6} \mathrm{H_2O_{gas}} + \mathrm{O_{ads}} \tag{2.7}$$

An additional feature of the model is the constancy of the total flow rate $\Phi = \Phi_\mathrm{CO} + \Phi_{\mathrm{O}_2} + \Phi_{\mathrm{H}_2}$. This condition is similar to the conducted experiments in previous studies [9–14, 22, 34] concerning the CO oxidation. Since the total flow restricts the single gas flows, every gas flow composition can be described as a single point $(\Phi_\mathrm{CO}/\Phi_{\mathrm{O}_2}/\Phi_{\mathrm{H}_2})$ in a ternary graph in which each axis represents the fraction $Y_i = \Phi_i/\Phi$ with $i = \{\mathrm{CO}, \mathrm{H}_2, \mathrm{O}_2\}$. The restriction reduces the degrees of freedom and therefore decreases the number of parameters to control the reaction system.

In this basic model, possible pressure effects such as global coupling [21, 42] are neglected, thus adsorption depends directly on gas flow and is not influenced by changing partial pressures. Therefore we propose the following differential equations for our model:

$$\frac{dn_\mathrm{CO}}{dt} = \Phi_\mathrm{CO} s_\mathrm{CO}(1 - n_\mathrm{CO} - n_\mathrm{H} - n_\mathrm{O} - n_\mathrm{OH})^a - k_1 n_\mathrm{CO} - k_2 n_\mathrm{CO} n_\mathrm{O} \tag{2.8}$$

$$\frac{dn_\mathrm{H}}{dt} = 2\Phi_{\mathrm{H}_2} s_\mathrm{H}(1 - n_\mathrm{CO} - n_\mathrm{H} - n_\mathrm{O} - n_\mathrm{OH})^b - 2k_3 n_\mathrm{H}^2 - k_4 n_\mathrm{H} n_\mathrm{O} - k_5 n_\mathrm{H} n_\mathrm{OH} \tag{2.9}$$

$$\frac{dn_\mathrm{O}}{dt} = 2\Phi_{\mathrm{O}_2} s_\mathrm{O}(1 - n_\mathrm{CO} - n_\mathrm{H} - n_\mathrm{O} - n_\mathrm{OH})^c - k_2 n_\mathrm{CO} n_\mathrm{O} - k_4 n_\mathrm{H} n_\mathrm{O} + k_6 n_\mathrm{OH}^2 \tag{2.10}$$

$$\frac{dn_\mathrm{OH}}{dt} = k_4 n_\mathrm{H} n_\mathrm{O} - k_5 n_\mathrm{H} n_\mathrm{OH} - 2k_6 n_\mathrm{OH}^2 \tag{2.11}$$

The formation rates of the products CO_2 and H_2O can be calculated directly according to equations (2.12) and (2.13).

$$r_{\mathrm{CO}_2} = k_2 n_\mathrm{CO} n_\mathrm{O} \qquad \propto n_\mathrm{CO} n_\mathrm{O} \tag{2.12}$$

$$r_{\mathrm{H}_2\mathrm{O}} = k_5 n_\mathrm{H} n_\mathrm{OH} + k_6 n_\mathrm{OH}^2 \qquad \overset{\propto}{\sim} n_\mathrm{OH} n_\mathrm{H} \tag{2.13}$$

The reaction constants were calculated for each temperature applying the Arrhenius equation using the activation energy and an exponential prefactor:

$$k_i = \nu_i \exp(-E_i/(TR)) \tag{2.14}$$

The activation energies and prefactors were derived from the literature (see table 2.1). All calculations were done for $T = 500$ K if not stated otherwise. This temperature was selected to

ensure bistability for CO oxidation without hydrogen [9].

The main challenge in treating hydrogen oxidation was the large variety of different reaction parameters [29, 31, 43] and simultaneously, the lack of data for the reaction specifically on Ir(111). Considering that the parameters for CO oxidation were chosen to characterise the results of specific experiments [9, 26], water formation parameter must accord with the established constants qualitatively and quantitatively. Due to the high uncertainty in this process, we distance our findings from the physical background by referring to the dummy-molecules **A**, **BB**, **CC** and **BC** instead of referring to CO, H_2, O_2 and OH. Thus our model treats the competing reactions $2\mathbf{A} + \mathbf{CC} \rightarrow 2\mathbf{AC}$ and $2\mathbf{BB} + \mathbf{CC} \rightarrow 2\mathbf{BBC}$.

It should be emphasised that our results are qualitatively reliable in some magnitude of k_i. Also, since the reaction constants for the CO oxidation (or **A** + **CC** reaction respectively) represent the experimental findings, the lower the hydrogen (or **BB**) flow, the better will the model agree with later experiments.

Table 2.1: Model parameters used in the numerical analysis and simulation. *Modified

Parameter	Value	Ref.	Parameter	Value	Ref.
$s_{\mathbf{A}}$	1	[27]	a	1	[27]
$s_{\mathbf{B}}$	0.007	[44]	b	2	[38]
$s_{\mathbf{C}}$	0.11	[27]	c	3	[27]
ML	$1.56 \cdot 10^{15}$ cm^{-2}	[27]	Φ	0.88 ML s^{-1}	[27]
E_1	140 kJ mol^{-1}	[27]	ν_1	$1 \cdot 10^{13}$ s^{-1}	[27]
E_2	40 kJ mol^{-1}	[27]	ν_2	10^5 (ML s)$^{-1}$	[27]
E_3	78 kJ mol^{-1}	[44]*	ν_3	$1.5 \cdot 10^{-1}$ (ML s)$^{-1}$	[44]*
E_4	19 kJ mol^{-1}	[40]	ν_4	6 (ML s)$^{-1}$	[39]*
E_5	10 kJ mol^{-1}	[40]*	ν_5	6 (ML s)$^{-1}$	[39]*
E_6	75 kJ mol^{-1}	[43]	ν_6	$1.3 \cdot 10^{-3}$ (ML s)$^{-1}$	[43]

2.3 Bifurcation Analysis

The differential equations (2.8–2.11) that constitute the proposed model feature several competing nonlinearities that make an analytical treatment impossible, thus one has to rely on approximations or numerical results for specific values of the parameters.

Our results show the existence of up to five equilibrium points (combinations of the coverages that make the right-hand side of the equations equal to zero). The qualitative changes of the equilibrium points as control parameters are varied can be studied with Bifurcation theory, that provides a suitable mathematical language.

In general, changes in the number of solutions induced by variations in one of the control parameters temperature T and fractions Y_{A}, Y_{BB} and Y_{CC}, are described by *saddle-node* bifurcations, i.e. associated with either annihilation or the creation of two solutions: a stable and an unstable solution. Therefore one can visualize transitions between monostable, bistable and tristable scenarios as parameters are varied. These transitions in the context of chemical surface reactions mean abrupt changes in surface occupations and therefore in reaction rates.

Figure 2.1(a) shows reaction rates and surface occupations as function of Y_A. Each colour indicates a constant value of parameter Y_{BB}. As a curve reaches a saddle-node bifurcation, it starts to go backward, and the stability of the solution changes (continuous style indicates stable, dashed style indicates unstable). Depending on the selection of Y_{BB}, up to four saddle-node bifurcations are found, although some of them are hard to see from the figure.

For low values of Y_{BB} such as $Y_{BB} = 0.025$, the behaviour is quite similar to previous findings for CO oxidation: there is a well-defined bistability range where an upper rate (UR) and a very low rate (VLR) state coexist stably. For $Y_{BB} = 0.075$ the unstable section develops two saddle-nodes and a stable section is born: now this section is not visible in the reaction rates, only in the **A** and **B** occupations. The new state has low rates (LR). For $Y_{BB} = 0.125$ one of the saddle-node bifurcations moves to the left to the $Y_A = 0$ edge, so the VLR exists stably almost everywhere. For $Y_{BB} = 0.2$ two saddle-node bifurcations collide and the thin stable branch LR disappears.

With this information, we can define approximative values for the three stable states we established above:

	n_A [ML]	n_B [ML]	n_C [ML]	n_{BC} [ML]
UR:	$\lessapprox 0.05$	≈ 0.05	≈ 0.2	≈ 0.03
LR:	$\lessapprox 0.80$	$\lessapprox 0.7$	≈ 0	≈ 0
VLR:	≈ 0.01	≈ 0.98	≈ 0	≈ 0

These values confirm our previous assumption of a high **B** occupation in the VLR. However, a distinction for LR and VLR is only meaningful for intermediate values of Y_{BB} where there is a clear transition mediated by folds. In terms of reaction rates, no quantitative difference between LR and VLR can be seen. Since reaction step **BC** + **B** is faster compared to **BC** + **BC** and **BC** is the limiting species in both states, the difference in **B** is insignificant. Therefore, both states should be differentiated via occupations.

A more complete analysis of the influence of Y_A and Y_{BB} at the same time can be obtained with the codimension-2 bifurcation diagram depicted in figures 2.1(b) and 2.1(c). Since $Y_A + Y_{BB} + Y_{CC} = 1$, a *ternary* representation is used. The figure includes only the location of the saddle-node bifurcations. Horizontal cuts through the codimension-2 diagram correspond to curves in fig. 2.1(a) for the surface species and calculated rates. Interestingly, the diagram shows how two pairs of saddle-node bifurcations annihilate in two *cusps*, defining a *swallowtail* region where three stable solutions coexist [46, 47]. This is a similar phenomenon to what Cui *et al.* found for CO oxidation [48] in a topological approach.

Within the swallowtail, a tristable regime is expected. Starting from there, crossing a single direct border should result in a bistable regime and the areas further away are expected to be monostable. The new stable state (very low rate, VLR) which is found in the largest area of the diagram is mostly influenced by **BB** flow, therefore high occupation of **B** can be expected.

From our previous studies, we know that higher temperature may induce coalescence of saddle-nodes [9, 10]. Fig. 2.2 illustrates the influence of temperature on the bifurcation behaviour. At 520 K, the tristable regime shrinks along with the bistable regime of the **A** + **CC** reaction. The marginal LR-VLR cusp pointing to the right increases therefore drastically and a new cusp

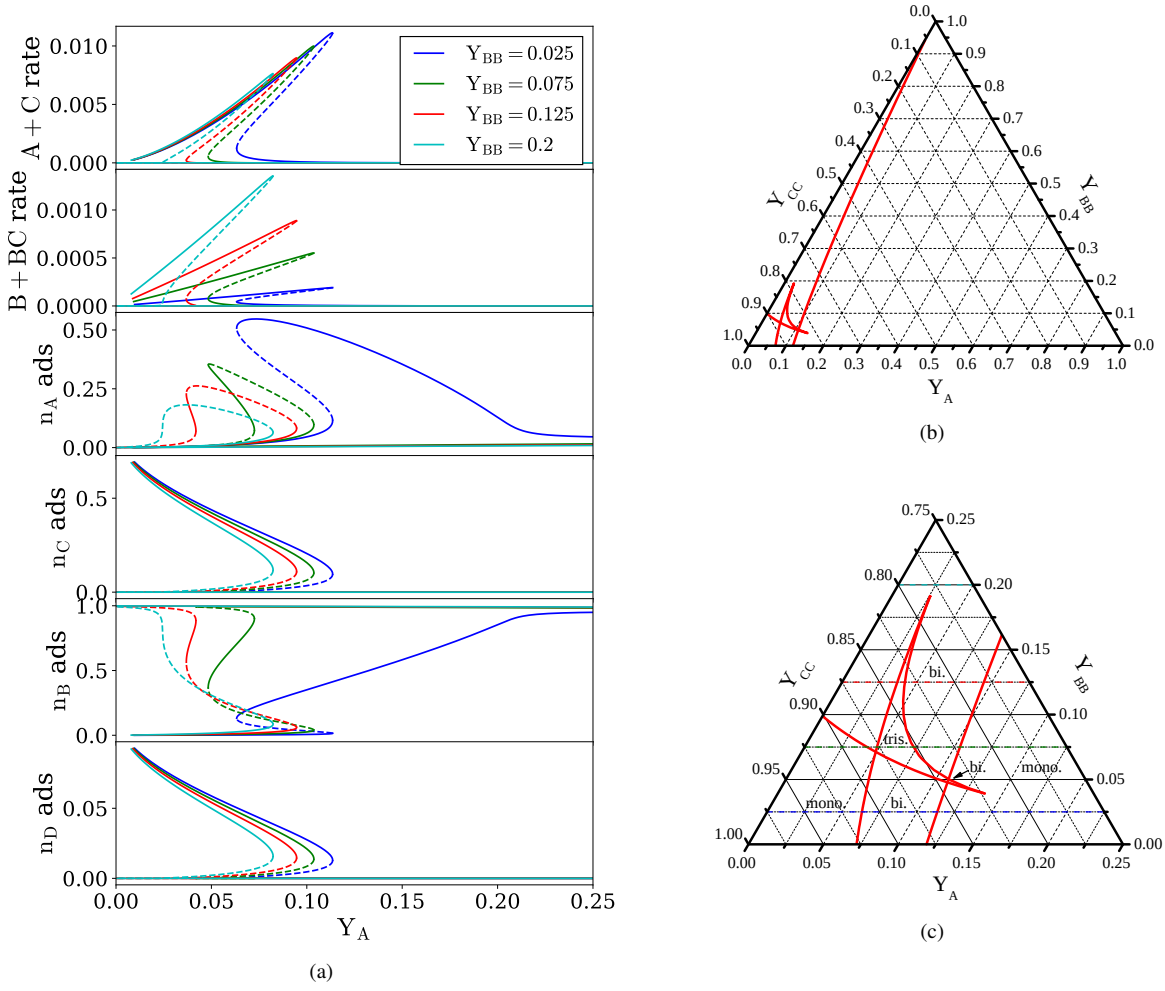
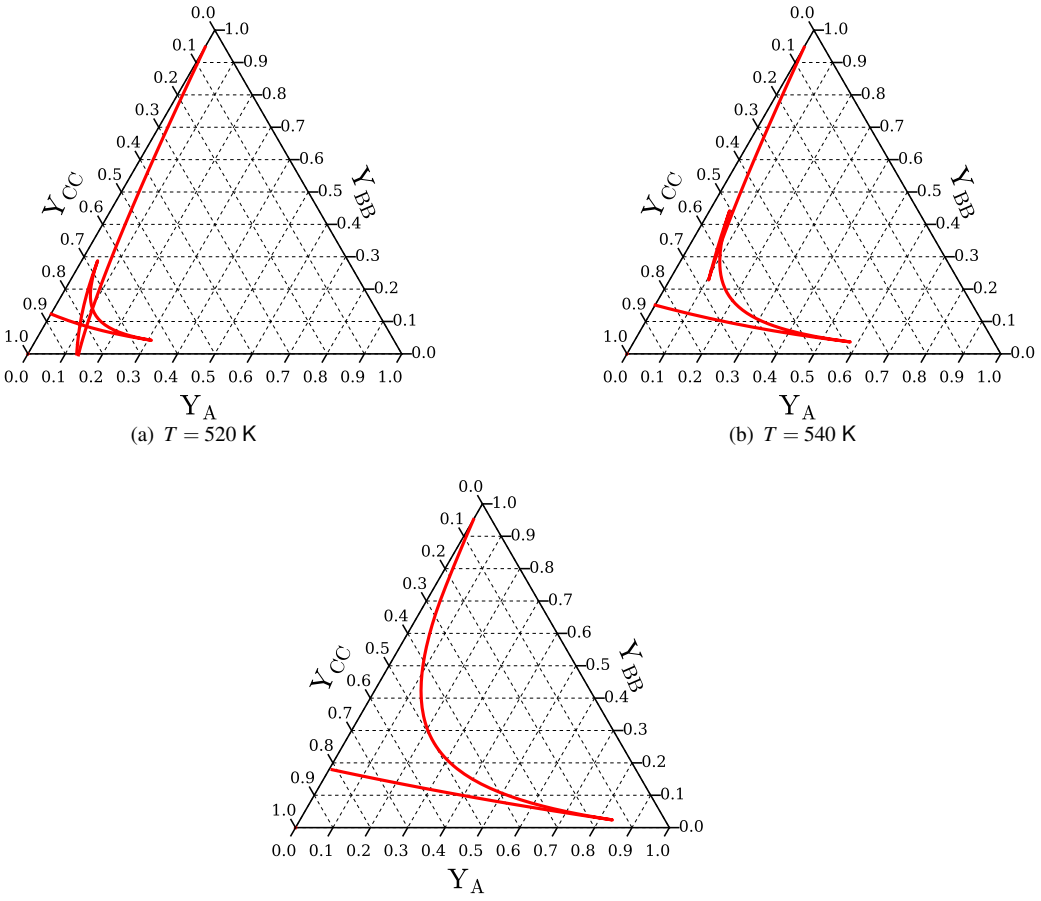


Figure 2.1: (a) Bifurcation diagrams that correspond to varying Y_A with constant Y_{BB} (Y_{BB} in $\{0.025, 0.075, 0.125, 0.20\}$) showing both stable (solid) and unstable (dashed) sections of the branches. The number of saddle-node bifurcations changes from one curve to the next. First two subplots show reaction rates of **A** + **C** and **B** + **BC**. The remaining four subplots show coverages of the four species on the surface. (b) Codimension-2 bifurcation diagram for T = 500 K. The saddle-node bifurcations form 'cusps' through their different interactions. (c) Detailed bifurcation diagram. Curves define regions with one, two and three stable occupation states. Inside 'swallowtail' all three stable states coexist. The coloured horizontal lines (Y_{BB} constant) corresponds to analyses presented in (a). Computations were performed using the bifurcation software AUTO-07p [45].

emerges from the lower $Y_A = 0$ edge. The unfolding continues till 540 K where the tristable region becomes negligible along with LR, as two cusps become closer to each other. A further increase to 560 K leaves only a single cusp and two separate areas: inside the cusp, there is a coexistence between VLR and UR; whereas outside of the cusp a continuous transition between UR, LR and VLR exists, as the selection Y_A, Y_{BB}, Y_{CC} goes around the cusp. Now as the temperature decreases, for instance $T = 480\ K$, the swallowtail moves to the left corner of the diagram and the new phenomena becomes less relevant.

The whole picture depicting the changes in the swallowtail would require two parameters, for



(a) $T = 520$ K

(b) $T = 540$ K

(c) $T = 560$ K

Figure 2.2: Codimension-2 bifurcation diagrams for higher temperatures, showing different interactions between cusps. With increasing temperature, the swallowtail unfolds and finally disappears. Computations were performed using the bifurcation software AUTO-07p [45].

instance, temperature T and total flux Φ, in addition to the Y_A, Y_{BB}, Y_{CC} used to generate each swallowtail. This complete picture should correspond to the *swallowtail catastrophe set* described in the literature [46]: the local minima of a sixth-order polynomial of a single variable. This complete set includes a case where three cusps collide. Here, we only show the influence of T as realistic changes in the total flux of Φ are not able to induce qualitative changes in the swallowtail. For instance, a hundredfold increase in Φ will reduce the size of the swallowtail to one-tenth of its original size, and a reduction of Φ to one-tenth will induce an enlargement of the swallowtail, comparable to what was obtained by raising the temperature to 550 K.

2.4 Simulation of experiment

One aim of this study was to simulate a simple experimental setup, based on our experience from CO oxidation [26]. These simulations can support a following experimental approach to a similar system by uncovering its features and challenges.

In the experiments conducted by Wehner *et al.* [9], certain initial conditions (Y_i, T, Φ) were set,

which lead to an initial occupation state. From there, the gas composition Y_i is shifted gradually by discrete values. For three gases, the most basic analogous setup with constant total flow Φ, one gas flow may be maintained while modifying the other two reciprocally (e.g. $(Y_1/Y_2/Y_3) \rightarrow (Y_1/Y_2 \pm \Delta Y/Y_3 \mp \Delta Y)$). For all simulations, the resolution in Y was limited by the technical limitation of the mass flow controllers (as in [9], $\Delta Y = 0.001$).

2.4.1 Time evolution

Before modelling the experimental conditions, simple simulations with fixed control parameters (gas composition and flow) starting with different initial occupations can be done to show the evolution towards the predicted stable states.

Therefore, we chose a point within the tristable regime and set up a variety of initial occupations. Numerical integration was done and selected trajectories are highlighted in figure 2.3.

All three stable states are reached after a large amount of integration time. Only extremely high values of initial n_B converge to VLR (red). While traces of **A** decelerate the process, **C** destabilises the VLR slowly until a critical composition is reached and LR (brown) is approached. Both, LR and VLR are only reached on long time scales.

Initial conditions with high proportions of **C** or high n_A and low n_B tend to converge to UR (blue). Apparently, for the chosen gas composition, **C** adsorbs faster than **A** and **B**, while $\mathbf{A}+\mathbf{C}$ reaction replenishes occupation sites. In these cases, UR is reached in extremely short times, compared to LR and VLR.

2.4.2 Linear scanning approach

In the following, the term 'scan' will be used for a series of numerical simulations with a stepwise increase/decrease of two gas components, starting and ending at one edge in the ternary diagram (e.g. $(Y_A/1-Y_A/0) \rightarrow (Y_A/0/1-Y_A)$, thus a scan parallel to Y_{CC} axis).

The following pre-step was performed to obtain reliable starting occupations for the scans. We started with the **A + CC** reaction ($Y_{BB} = 0$) as this case represents our previous findings [19, 23, 24]. Three consecutive scans (forward, backward and forward) were run to ensure quasi-static behaviour. The resulting occupations at $Y_A = 0.015$ and $Y_A = 1$ were set as initial conditions for calculating the other edges of the triangle (**BB + CC** reaction and **A + BB** adsorption equilibrium). Two consecutive scans (forward and backwards) were sufficient to obtain information about bistability. Furthermore, convergence tests were performed to select a sufficiently long time step.

All distinguishable occupation states (one or two per set of Y) were collected from data. Each set served as an initial condition for further two scans. The series of scans from the same starting border and in the same direction can be visualized by a vector in the ternary diagram.

A selection of our results is highlighted in figure 2.4. They are representative of the whole set of scans since they show all stable states within the ternary regime. High occupation of one species correlates directly with a specific kinetic state. Therefore, we can apply the kinetic terminology of UR, LR and VLR to identify the specific occupations and vice versa.

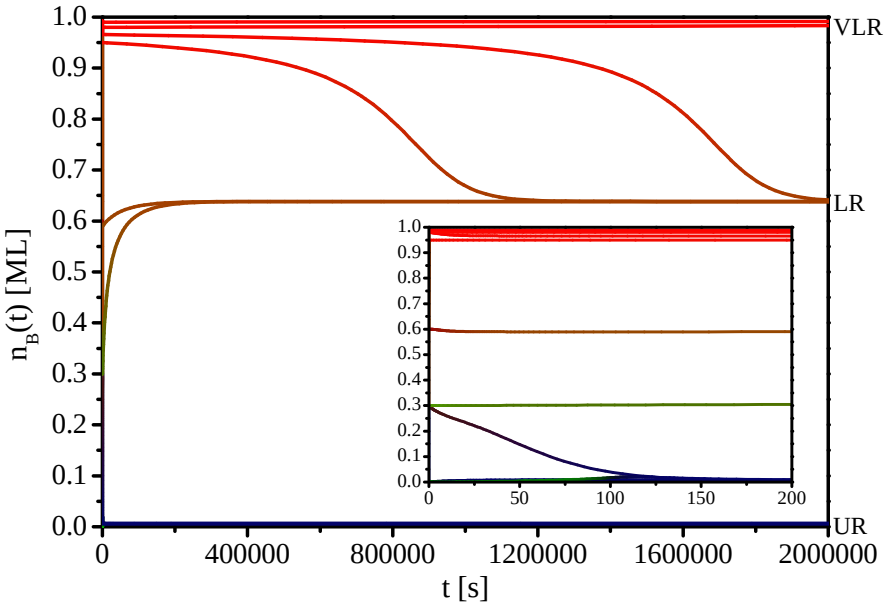


Figure 2.3: Time evolution of n_B for the gas flow composition $Y_A = 0.06$, $Y_{BB} = 0.075$, $Y_{CC} = 0.865$ (within the 'swallowtail') with different initial occupations. The inner graph shows the short-scale evolution while the outer shows the long-scale evolution. The color of each trajectory represents the total surface occupation according to RGB $= (n_B/n_A/n_C + n_{BC})$. Eventually, all trajectories reach one of the stable states (UR: blue, LR: brown, VLR: red), if given a sufficient amount of time.

Concerning the **A** + **C** reaction, the presence of a small amount of **B** lowers r_{AC} at first only marginally, especially if decreasing Y_A for Y_{BB}. It is remarkable however that an amount of $Y_{BB} > 0.05$ is sufficient to ensure n_B close to 1 in most cases. Considering **A** and **B** could represent carbon monoxide and hydrogen, a different result was expected. CO is often found to poison catalytic surfaces, which is especially crucial in PEM fuel cells [49]. Experimentally, quantitative hydrogen desorption occurs at for many platinum group catalysts at temperatures between 300 and 400 K [38, 50], so desorption of **B** might be underrated. However, the mostly hydrogen covered surface is consistent with the theoretical results of Zhdanov [31]. He also found bistability which is compatible with the present results. Compared to our model, he used lower temperatures and higher pressures, but an extremely large desorption coefficient for hydrogen. These differences show the large variety of parameter values which can show qualitatively similar behaviour.

In the first occupation pattern (figure 2.4(a)) UR (high **C**, blue) is visited in all its stable regions. All scans which showed this pattern started from UR within the bistable regime. In terms of reaction rates (figure 2.4(b)), this pattern shows the highest possible rates for **AC** formation, as well as for **BBC** formation. The requirement for these high rates is the high **C** occupation (corresponding to higher **BC** in case of **BBC** reaction).

The second occupation pattern (figure 2.4(c)) illustrates the other extreme case with maximal VLR (high **B**, red). Formation rates (figure 2.4(d)) for both reactions collapse, $r_{AC,max}$ is only half as large as in the UR case, $r_{BBC,max}$ decreases to a fifth. Besides in the monostable UR area, both reactions only occur at a low rate in the UR-LR bistable area. This means that high n_B prevents

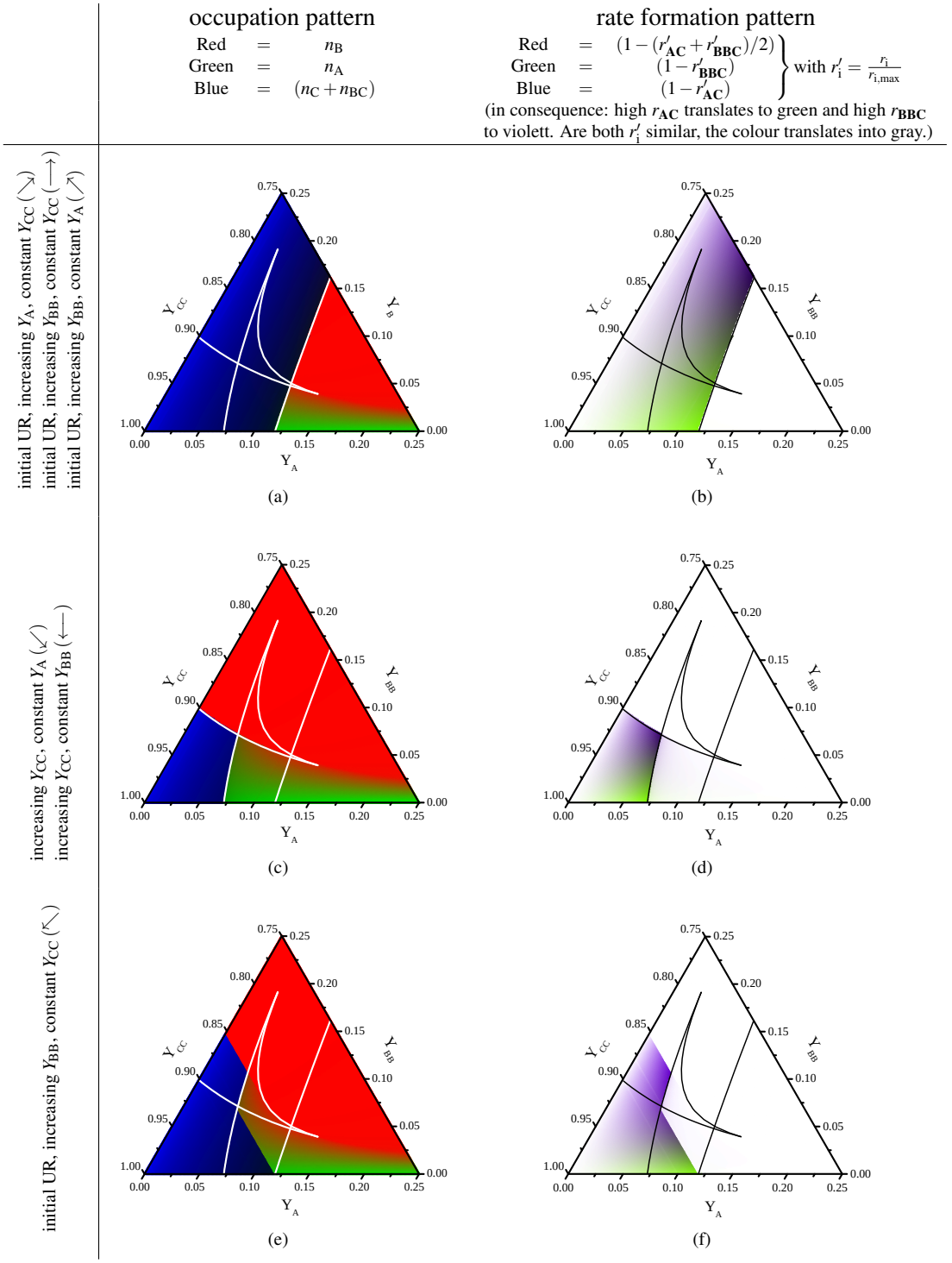


Figure 2.4: Highlighted occupation patterns (left) and resulting product formation rates (right) with different scan vectors for experiment simulations. The Colour schema is derrived by translating n_i (r_i respectively) into RGB values. (a) The blue regions show areas of high n_C corresponding high reaction rates for both reactions (UR). (b) **BBC** formation is faster where **B** is high and **AC** formation is faster where **A** is high. (c) The red region shows areas of high n_B corresponding to VLR. (d) $r_{AC,max}$ is only half as large as in the UR case, $r_{BBC,max}$ decreases to a fifth of before. (e) The green region shows areas of high n_A corresponding to LR. (f) Due to starting with UR, **AC** formation rate reaches the same maximum as in (a) while **BBC** rate maximum increases only marginally compared to (b). The linear edges are due to state transitions when crossing the saddle-node curves.

both reactions more strongly than high **A**.

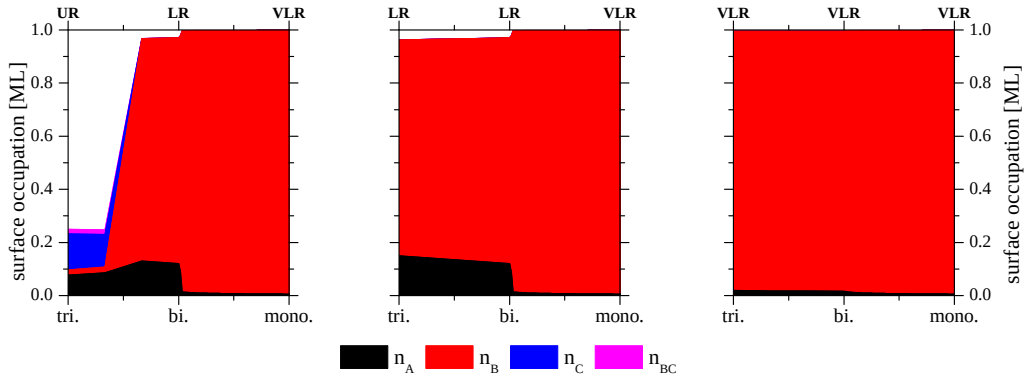
This is also shown in the third occupation pattern (figure 2.4(e)). Since LR (high **A**, green) is dominant in the swallowtail, the reactions occur both at a slow rate. While **BBC** formation (figure 2.4(f)) only accelerates a little in the upper UR area, $r_{AC,max}$ increases to its value in (b) because of the lower UR area.

Although LR reaction rates (**A** + **CC** and **BB** + **CC**) are slightly faster than VLR, we highlight again, that a difference between both might be hard to notice in the experiment looking at only reaction rates. Main rate deciding species is **C** and in both cases, occupation is not significant enough to provide sufficient difference.

An interesting feature in these figures is the presence of sharp lines which are parallel to the direction of the scan and divide the two bistable regimes. They follow the scanning direction and show clear transitions depending on which cusp is crossed. When crossing from tristable to the bistable regime at the left cusp, LR transitions preferably to UR. Otherwise, if the bent line of the swallowtail from LR, VLR is more stable than UR. Because of the simple linear scanning approach, it is also not possible to show LR in the complete swallowtail.

2.4.3 Transition Analysis

To highlight the kinetic phase transitions and obtain information about the global stability of a certain state, simulations along more complex paths were performed. Each path started in the tristable area, crossing two saddle-node curves at different points. Figure 2.5 exemplifies the different behaviour obtained by moving into the bistable VLR/LR regime. Since the difference between both states is subtle and the transition is continuous over the majority of the diagram, this region was particularly interesting.



(a) Starting with UR. (b) Starting with LR. (c) Starting with VLR.

Figure 2.5: Scanning along a path from tristable point (Y_A = 0.107, Y_{BB} = 0.049, Y_{CC} = 0.844) over bistable (LR/VLR) point (Y_A = 0.110, Y_{BB} = 0.049, Y_{CC} = 0.841) to monostable (VLR) point (Y_A = 0.110, Y_{BB} = 0.109, Y_{CC} = 0.781). The figure illustrates the different behaviors when different kinetic states become unstable.

Since VLR can be visited in all three points, no qualitative change or transition occurs (fig. 2.5(c)). Otherwise, when leaving the small bistable regime in LR state, a transition occurs (fig. 2.5(b)).

Only VLR is stable after the bistable point. As seen here, the difference between LR and VLR is quantitatively not large. UR is only stable within the tristable regime (fig. 2.5(a)) which means transition occurs when crossing the edge of the swallowtail. Although both states, LR and VLR are stable in the bistable region, LR is visited. This shows that even though VLR seems dominant in most conditions (see fig. 2.4), in the bistable (LR/VLR)-tip LR is slightly more stable.

This behaviour has to be anticipated in the experiment. When shifting the parameters from a monostable to a bistable regime, no transition will occur. The backwards-scans in all three of the presented cases are equal to figure 2.5(c). In consequence, a single scan neither displays the start of regimes with more stable states nor necessarily reveals the disappearance of a stable state. Only the destabilisation of the visited state can be demonstrated.

2.5 Heterogeneous concentrations

One of the most influencing factors we neglected in our basic model is diffusivity. In general, diffusion of adsorbates is dependent on temperature, strengths of bonding, the species and also coverages [20]. Since particles move on the surface and react only if they find an adequate partner, diffusion can influence the reaction dramatically. This leads often to macroscopic surface patterns: waves, islands, and fronts.

Since the effect of reaction fronts is important but there are still many uncertainties about diffusion mechanisms and timescales, only the most basic approach to diffusion will be used. The simple model is augmented in the following way:

$$\begin{aligned}\frac{\partial n_{\mathrm{CO}}}{\partial t} &= \Phi_{\mathrm{CO}} s_{\mathrm{CO}} (1 - n_{\mathrm{CO}} - n_{\mathrm{H}} - n_{\mathrm{O}} - n_{\mathrm{OH}})^a \\ &\quad - k_1 n_{\mathrm{CO}} - k_2 n_{\mathrm{CO}} n_{\mathrm{O}} + D_{\mathrm{CO}} \Delta n_{\mathrm{CO}} \qquad (2.15)\\ \frac{\partial n_{\mathrm{H}}}{\partial t} &= 2\Phi_{\mathrm{H_2}} s_{\mathrm{H}} (1 - n_{\mathrm{CO}} - n_{\mathrm{H}} - n_{\mathrm{O}} - n_{\mathrm{OH}})^b \\ &\quad - 2k_3 n_{\mathrm{H}}^2 - k_4 n_{\mathrm{H}} n_{\mathrm{O}} - k_5 n_{\mathrm{H}} n_{\mathrm{OH}} + D_{\mathrm{H}} \Delta n_{\mathrm{H}} \qquad (2.16)\\ \frac{\partial n_{\mathrm{O}}}{\partial t} &= 2\Phi_{\mathrm{O_2}} s_{\mathrm{O}} (1 - n_{\mathrm{CO}} - n_{\mathrm{H}} - n_{\mathrm{O}} - n_{\mathrm{OH}})^c \\ &\quad - k_2 n_{\mathrm{CO}} n_{\mathrm{O}} - k_4 n_{\mathrm{H}} n_{\mathrm{O}} + k_6 n_{\mathrm{OH}}^2 + D_{\mathrm{O}} \Delta n_{\mathrm{O}} \qquad (2.17)\\ \frac{\partial n_{\mathrm{OH}}}{\partial t} &= k_4 n_{\mathrm{H}} n_{\mathrm{O}} - k_5 n_{\mathrm{H}} n_{\mathrm{OH}} - 2k_6 n_{\mathrm{OH}}^2 + D_{\mathrm{OH}} \Delta n_{\mathrm{OH}} \qquad (2.18)\end{aligned}$$

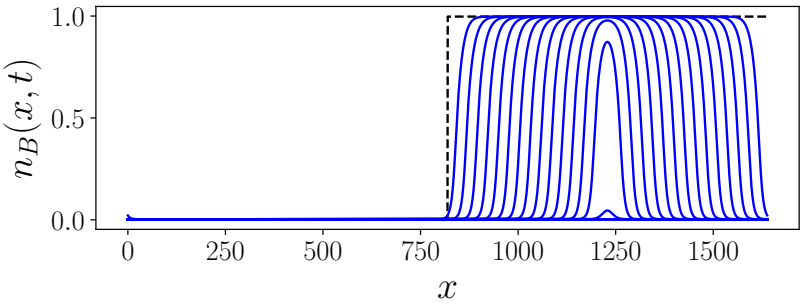
We tested different combinations of Diffusion coefficients but noticed no relevant differences. Selected results for $D_{\mathrm{A}} = 1, D_{\mathrm{B}} = 10, D_{\mathrm{C}} = D_{\mathrm{BC}} = 0.1$ (the spatial length scale is modified accordingly) are presented in Fig. 2.6 and 2.7.

In fig. 2.6, we considered two basic scenarios: one bistable case (UR and VLR coexist); and one tristable case (UR coexists with LR and VLR). These situations are investigated using a simple one-dimensional domain with periodic boundary conditions, prepared with a step-wise constant initial condition that includes each stable state, and integrated with a traditional Cranck-Nicolson semi-implicit scheme [51].

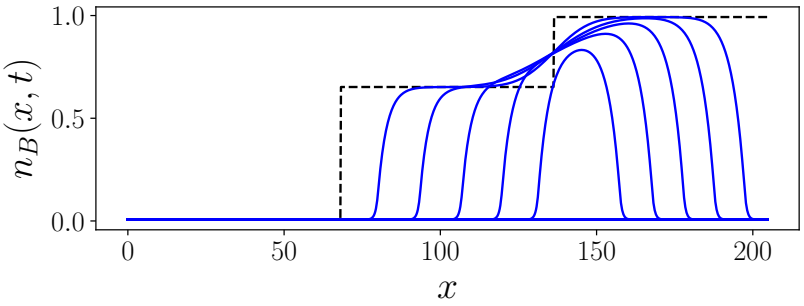
For both cases, we found UR to be dominant, as the other states were close to saddle-node

bifurcations where they become unstable.

Observing the interphase between UR and VLR, Diffusion destabilises the nearly completely occupied VLR which consequently replenishes space for **C** to adsorb. Similar conclusions can be drawn from the observations of the interphase between UR and LR in fig. 2.6(b). **C** adsorbs quickly on vacant sites that became free through diffusion. Due to the increase of n_C, both surface reactions accelerate which has an auto-catalytic effect on the corrosion of the LR.



(a) Y_A = 0.025, Y_{CC} = 0.85, Y_{BB} = 0.125



(b) Y_A = 0.06, Y_{CC} = 0.865, Y_{BB} = 0.075

Figure 2.6: Time evolution of **B** coverage following equations (2.15–2.18) in a periodic one dimensional domain for two different gas compositions in the ternary diagram. The initial condition in each figure is indicated with a dashed line that divides the domain between the stable states. (a) This selection corresponds to a bistable regime, where UR (lower value of n_B) and LR (higher value of n_B) are stable. (b) This selection corresponds to a tristable regime (inside the 'swallow-tail' of Fig. 2.1(c)), where UR, LR, and VLR (highest value of n_B) are stable. In both cases, the coverage becomes smoother over time and the UR state dominates (the part of the domain where $n_B > 0$ shrinks to zero). Numerical integration was performed with Crank-Nicolson semi-implicit method.

As the tristable case is of particular interest, we further examined it in a 2-dimensional approach. A circular domain with zero flux on the boundary is chosen and divided equally by UR, LR and VLR (fig. 2.6). This more complex case illustrates only a slightly different picture compared to the one-dimensional case, where the UR corrodes LR and VLR equally. Firstly, LR invades the interface between both states and extends towards VLR (b). Presumably, the contact with the UR ($n_B \approx 0$) accelerates the transitions from VLR ($n_B \approx 1$) to LR ($n_B \approx 0.6$). Upon further evolution, UR expands towards LR (c) and dominates at the end of the simulation (d).

More detailed experiments should be performed to extract the features of diffusion. Among

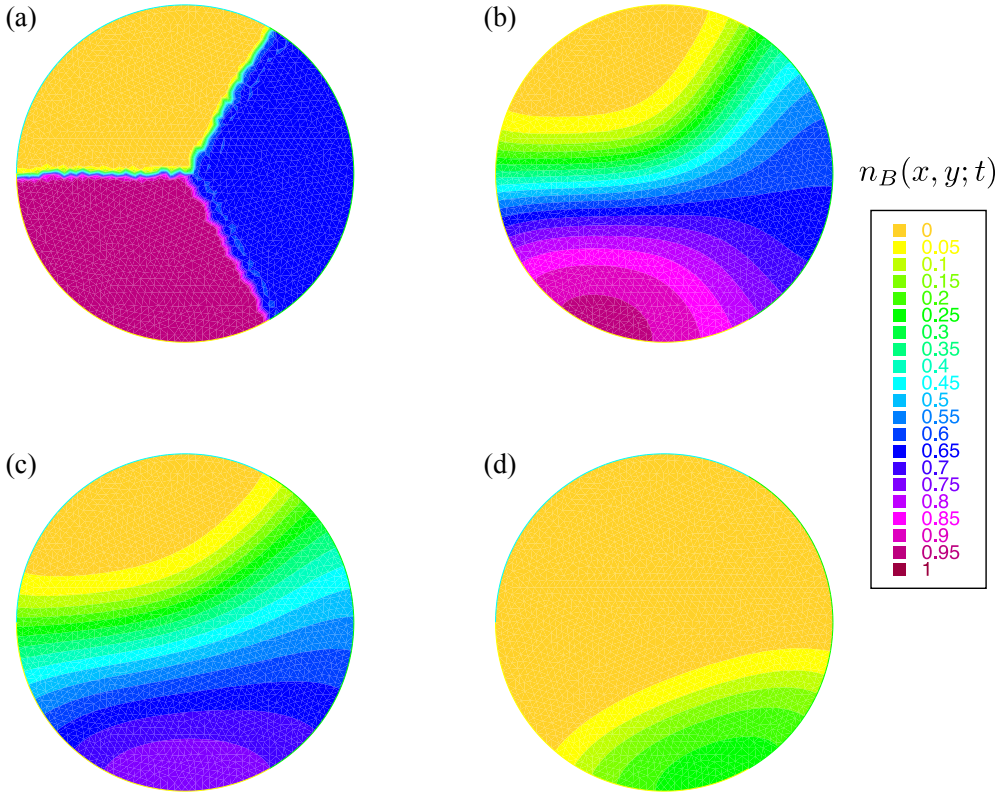


Figure 2.7: Time-evolution of B (hydrogen) coverage following equations (2.15–2.18) in a circular two dimensional domain with zero flux on the boundary and the gas composition of $Y_{\mathrm{A}} = 0.06$, $Y_{\mathrm{BB}} = 0.075$, $Y_{\mathrm{CC}} = 0.865$. The initial condition shown in (a) splits the domain evenly between the three stable states: UR (orange), LR (blue), and VLR (red). As simulation progresses, the LR invades the interface between UR and VLR (b) and starts to extinguish the VLR (c). In the end, the UR state dominates over LR and will cover the whole disc. Computation was performed using `FreeFem++` [52].

the multiple phenomena induced by diffusion, one should mention front instabilities in two spatial dimensions.

2.6 Conclusions

In this article, we have presented theoretical work on a surface reaction system that besides oxygen and carbon monoxide, includes the presence of hydrogen in the feeding gas. This simple change forces the inclusion of several new reaction pathways, new adsorbed species (hydrogen, hydroxide and water), and the water formation that may collaborate or compete with the CO oxidation.

The resulting 4-dimensional system of differential equations can be analysed within the framework of bifurcation theory, revealing a state that features high CO and water formations, thus basically coincides with the upper rate state (UR) that one would expect for CO oxidation; and up to two lower rate states (LR and VLR) with limited formation rates. These three states can coexist stably, and a codimension-2 diagram (using the molar fractions as control parameters) reveals how the saddle-node bifurcations are connected by ‘cusps’ that form a ‘swallowtail’ diagram, suggest-

ing how other additional control parameters such as temperature and total flux can enlarge the tristability region.

The bistability and tristability can be made manifest in experiments by preparing the state of the surface and slowly scanning (quasi-statically) the control parameters in different 'directions'. Our work anticipates formation rates that follow non-trivial changes induced by the underlying structure of the cusps. The tristable region can only be revealed by manipulating the molar fractions along certain 'paths' in the codimension-2 diagram.

Finally, making a simple assumption about the diffusion mechanisms of the adsorbed species, we explored reaction-diffusion fronts that can make one state dominant over the other stable states. Only a detailed analysis of experiments will indicate the most appropriate diffusion mechanisms and the existence of spatio-temporal patterns.

Data accessibility

Python codes for numerical simulations can be accessed in the supplementary material.

Acknowledgments

K.R., J.C. and S.W. thank the general support of Dr. Christian Fischer. K.R. thanks Lise Boulard for proofreading and discussion.

Bibliography

[1] Langmuir I *The Adsorption of Gases on Plane Surfaces of Glass, Mica and Platinum*. (1918) *J. Am. Chem. Soc.* **40**, 1361–1403. doi: 10.1021/ja02242a004.

[2] Eley DD, Rideal EK *Parahydrogen Conversion on Tungsten*. (1940) *Nature* **146**, 401–402. doi: 10.1038/146401d0.

[3] Engel T *A Molecular Beam Investigation of He, CO, and O_2 Scattering from Pd(111)*. (1978) *J. Chem. Phys.* **69**, 373. doi: 10.1063/1.436363.

[4] Engel T, Ertl G *Elementary Steps in the Catalytic Oxidation of Carbon Monoxide on Platinum Metals*. (1979) *Adv. Catal.* **28**, 1–78. doi: 10.1016/S0360-0564(08)60133-9.

[5] Ertl G, Norton PR, Rüstig J *Kinetic Oscillations in the Platinum-catalyzed Oxidation of CO*. (1982) *Phys. Rev. Lett.* **49**, 177–180. doi: 10.1103/PhysRevLett.49.177.

[6] Krischer K, Eiswirth M, Ertl G *Bifurcation Analysis of an Oscillating Surface Reaction Model*. (1991) *Surf. Sci.* **251–252**, 900–904. doi: 10.1016/0039-6028(91)91121-D.

[7] Krischer K, Eiswirth M, Ertl G *Periodic Perturbations of the Oscillatory CO Oxidation on Pt(110): Model Calculations*. (1992) *J. Phys. Chem* **97**, 307–319. doi: 10.1063/1.463629.

[8] Krischer K, Eiswirth RM, Ertl G *Oscillatory CO Oxidation on Pt (110): Modeling of Temporal Self-Organization*. (1992) *J. Phys. Chem.* **96**, 9161–9172. doi: 10.1063/1.462226.

[9] Wehner S, Baumann F, Ruckdeschel M, Küppers J *Kinetic Phase Transitions in the Reaction CO+O→CO_2 on Ir(111) Surfaces*. (2003) *J. Chem. Phys.* **119**, 6823–6831. doi: 10.1063/1.1603212.

[10] Wehner S, Baumann F, Küppers J *Kinetic Hysteresis in the CO Oxidation Reaction on Ir(111) Surfaces*. (2003) *Chem. Phys. Lett.* **370**, 126–131. doi: 10.1016/S0009-2614(03)00059-9.

[11] Hayase Y, Wehner S, Küppers J, Brand HR *External Noise Imposed on the Reaction-Diffusion System CO+O_2→CO_2 on Ir(111) Surfaces: Experiment and Theory*. (2004) *Phys. Rev. E* **69**, 1–15. doi: 10.1103/PhysRevE.69.021609.

[12] Wehner S, Hoffmann P, Schmeißer D, Brand HR, Küppers J *Spatiotemporal Patterns of External Noise-Induced Transitions in a Bistable Reaction-Diffusion System: Photoelectron Emission Microscopy Experiments and Modeling*. (2005) *Phys. Rev. Lett.* **95**, 1–4. doi: 10.1103/PhysRevLett.95.038301.

[13] Hoffmann P, Wehner S, Schmeißer D, Brand HR, Küppers J *Noise-Induced Spatiotemporal Patterns in a Bistable Reaction-Diffusion System: Photoelectron Emission Microscopy Experiments and Modeling of the CO Oxidation Reaction on Ir(111)*. (2006) *Phys. Rev. E* **73**, 1–18. doi: 10.1103/PhysRevE.73.056123.

[14] Wehner S, Hoffmann P, Schmeisser D, Brand HR, Küppers J *The Consequences of Anisotropic Diffusion and Noise: PEEM at the CO Oxidation Reaction on Stepped Ir(111) Surfaces*. (2006) *Chem. Phys. Lett.* **423**, 39–44. doi: 10.1016/j.cplett.2006.03.010.

[15] Berdau M, Yelenin GG, Karpowicz A, Ehsasi M, Christmann K, Block JH *Macroscopic and Mesoscopic Characterization of a Bistable Reaction System: CO Oxidation on Pt(111) Surface*. (1999) *J. Chem. Phys.* **110**, 11551–11573. doi: 10.1063/1.479097.

[16] Matsushima T, Matsui T, Hashimoto M *Kinetic Studies on the CO Oxidation on a Rh(111) Surface by Means of Angle-Resolved Thermal Desorption*. (1984) *J. Chem. Phys.* **81**, 5151–5160. doi: 10.1063/1.447462.

[17] Peden CHF, Goodman DW, Blair DS, Berlowitz PJ, Fisher GB, Oh SH *Kinetics of CO Oxidation by O_2 or NO on Rh(111) and Rh(100) Single Crystals*. (1988) *J. Phys. Chem.* **92**, 1563–1567. doi: 10.1021/j100317a038.

[18] Karpitschka S, Wehner S, Küppers J *Reaction Hysteresis of the CO+O→CO_2 Reaction on Palladium(111)*. (2009) *J. Chem. Phys.* **130**, 1–11. doi: 10.1063/1.3072712.

[19] Wehner S, Karpitschka S, Burkov Y, Schmeisser D, Küppers J, Brand HR *Stochastic Aspects of Pattern Formation during the Catalytic Oxidation of CO on Pd(111) Surfaces*. (2010) *Physica D* **239**, 746–751. doi: 10.1016/j.physd.2009.06.010.

[20] Cisternas J, Karpitschka S, Wehner S *Travelling Fronts of the CO Oxidation on Pd(111) with Coverage-Dependent Diffusion*. (2014) *J. Chem. Phys.* **141**, 164106–164106. doi: 10.1063/1.4898705.

[21] Cisternas J, Wehner S *Detailed Analysis of Transitions in the CO Oxidation on Palladium(111) under Noisy Conditions*. (2018) *J. Chem. Phys.* **149**, 044706–044706. doi: 10.1063/1.5040704.

[22] Hayase Y, Wehner S, Küppers J, Brand HR *The Role of Sampling Time in Measurements of the CO_2 Kinetics of the Bistable Reaction $CO+O_2$ on Ir(111) Surfaces*. (2005) *Physica D* **205**, 15–24. doi: 10.1016/j.physd.2005.02.007.

[23] Cisternas J, Escaff D, Descalzi O, Wehner S *Stochastic Model Calculation for the Carbon Monoxide Oxidation on Iridium(111) Surfaces*. (2009) *Int. J. Bifurc. Chaos Appl. Sci. Eng.* **19**, 3461–3472. doi: 10.1142/S0218127409024906.

[24] Cisternas J, Escaff D, Descalzi O, Wehner S *Carbon Monoxide Oxidation on Iridium(111) Surfaces under Colored Noise*. (2010) *Int. J. Bifurc. Chaos Appl. Sci. Eng.* **20**, 243–254. doi: 10.1142/S0218127410025685.

[25] Cisternas J, Lecaros R, Wehner S *Carbon Monoxide Oxidation on Iridium(111) Surfaces Driven by Strongly Colored Noise*. (2011) *Eur. Phys. J. D* **102**, 91–102. doi: 10.1140/epjd/e2010-00268-3.

[26] Wehner S *The CO-Oxidation Reaction on Ir(111) Surfaces: Bistability, Noise and Spatio-Temporal Patterns in Experiment and Modeling*. (2009) *Int. J. Bifurc. Chaos Appl. Sci. Eng.* **19**, 2637–2675. doi: 10.1142/S0218127409024384.

[27] Cisternas J, Wehner S, Descalzi O *CO Oxidation on Ir(111) Surfaces under Large Non-Gaussian Noise*. (2012) *J. Chem. Phys.* **137**, 064105–064105. doi: 10.1063/1.4742191.

[28] Wehner S, Cisternas J, Descalzi O, Küppers J *Noisy CO Oxidation on Iridium(111) Surfaces*. (2014) *Eur. Phys. J.: Spec. Top.* **223**, 21–41. doi: 10.1140/epjst/e2014-02082-5.

[29] Schaak A, Nieuwenhuys B, Imbihl R *Anisotropic Chemical Waves on a Stepped Surface: The NO+H and O+H Reactions on Rh(533)*. (1999) *Surf. Sci.* **441**, 33–44. doi: 10.1016/S0039-6028(99)00783-9.

[30] Schaak A, Imbihl R *Bistability and Formation of Low Work Function Areas in the O_2+H_2 Reaction on a Rh(111) Surface*. (2000) *J. Chem. Phys.* **113**, 9822–9829. doi: 10.1063/1.1322033.

[31] Zhdanov VP *Kinetic Model of the Hydrogen-Oxygen Reaction on Platinum: Bistability, Chemical Waves and Oscillations*. (1993) *Surf. Sci.* **296**, 261–273. doi: 10.1016/0039-6028(93)91153-G.

[32] Stoltze P, Nørskov JK *Bridging the "Pressure Gap" between Ultrahigh-Vacuum Surface Physics and High-Pressure Catalysis*. (1985) *Phys. Rev. Lett.* **55**, 2502–2505. doi: 10.1103/PhysRevLett.55.2502.

[33] Schloegl R, Schoonmaker RC, Muhler M, Ertl G *Bridging the "Material Gap" between Single Crystal Studies and Real Catalysis*. (1988) *Catal. Lett.* **1**, 237–241. doi: 10.1007/BF00766181.

[34] Wehner S, Hayase Y, Brand HR, Küppers J *Multiplicative Temperature Noise Applied to a Bistable Surface Reaction: Experiment and Theory*. (2004) *J. Phys. Chem. B* **108**, 14452–14461. doi: 10.1021/jp0497155.

[35] Azar M, Caps V, Morfin F, Rousset JL, Piednoir A, Bertolini JC, Piccolo L *Insights into Activation, Deactivation and Hydrogen-Induced Promotion of a Au/TiO_2 Reference Catalyst in CO Oxidation*. (2006) *J. Catal.* **239**, 307–312. doi: 10.1016/j.jcat.2006.02.012.

[36] Piccolo L, Nassreddine S, Morfin F *Surface Study of the Hydrogen-Free or Preferential Oxidation of CO: Iridium vs. Platinum*. (2012) *Catal. Today* **189**, 42–48. doi: 10.1016/j.cattod.2012.02.046.

[37] Bär M, Zülicke C, Eiswirth RM, Ertl G, Eiswirth RM, Ertl G *Theoretical Modeling of Spatiotemporal Self-Organization in a Surface Catalyzed Reaction Exhibiting Bistable Kinetics*. (1992) *J. Chem. Phys.* **96**, 8595–8604. doi: 10.1063/1.462312.

[38] Engstrom JR, Tsai W, Weinberg WH *The Chemisorption of Hydrogen on the (111) and (110)-(1×2) Surfaces of Iridium and Platinum*. (1987) *J. Chem. Phys.* **87**, 3104–3119. doi: 10.1063/1.453048.

[39] Williams WR, Marks CM, Schmidt LD *Steps in the Reaction H_2+O_2⇋H_2O on Platinum: Hydroxy Desorption at High Temperatures*. (1992) *J. Phys. Chem.* **96**, 5922–5931. doi: 10.1021/j100193a051.

[40] Kwasniewski VJ, Schmidt LD *Steps in the Reaction H_2+O_2⇋H_2O on Pt(111): Laser-induced Thermal Desorption at Low Temperature*. (1992) *J. Phys. Chem.* **96**, 5931–5938. doi: 10.1021/j100193a052.

[41] Picolin A, Busse C, Redinger A, Morgenstern M, Michely T *Desorption of H_2O from Flat and Stepped Pt(111)*. (2009) *J. Phys. Chem. C* **113**, 691–697. doi: 10.1021/jp808170f.

[42] Mertens F, Imbihl R, Mikhailov AS *Breakdown of Global Coupling in Oscillatory Chemical Reactions*. (1993) *J. Chem. Phys.* **99**, 8668–8671. doi: 10.1063/1.465590.

[43] Anton AB, Cadogan DC *The Mechanism and Kinetics of Water Formation on Pt(111)*. (1990) *Surf. Sci. Lett.* **239**, 548–560. doi: 10.1016/0039-6028(90)90217-V.

[44] Ibbotson DE, Wittrig TS, Weinberg WH *The Chemisorption of Hydrogen on the (110) Surface of Iridium*. (1980) *J. Chem. Phys.* **72**, 4885–4895. doi: 10.1016/0039-6028(79)90146-8.

[45] Doedel E *AUTO-07P: Continuation and Bifurcation Software for Ordinary Differential Equations*. (2007) Technical report Concordia University Montreal, Canada.

[46] Poston T, Stewart I *The First Seven Catastrophe Geometries*. (1996) Mineola, New York: Dover Publications.

[47] Chirilus-Bruckner M, Doelman A, van Heijster P, Rademacher JD *Butterfly Catastrophe for Fronts in a Three-Component Reaction–Diffusion System*. (2014) *J. Nonlinear Sci.* **25**, 87–129. doi: 10.1007/s00332-014-9222-9.

[48] Cui T, Tang S, Zhang L, Yu D *Swallowtail Model for Predicting the Global Bifurcation Behavior of CO Oxidation Reactions*. (2011) *Sci. China Chem.* **54**, 1072–1077. doi: 10.1007/s11426-011-4294-y.

[49] Farrell CG, Gardner CL, Ternan M *Experimental and Modelling Studies of CO Poisoning in PEM Fuel Cells*. (2007) *J. Power Sources* **171**, 282–293. doi: 10.1016/j.jpowsour.2007.07.006.

[50] Colonell J, Curtiss T, Sibener S *Coverage Dependence of the Kinetics for H_2 Desorption from Rh(111)*. (1996) *Surf. Sci.* **366**, 19–28. doi: 10.1016/0039-6028(96)00807-2.

[51] Vives D, Armero J, Martí A, Ramírez-Piscina L, Casademunt J, Sancho JM, Sagués F *Propagating Fronts in Reaction-Diffusion Systems*. (1998) *J. Math. Chem.* **23**, 239–260. doi: 10.1023/A:1019152720718.

[52] Hecht F *New Development in Freefem++*. (2012) *J. Numer. Math* **20**, 251–265. doi: 10.1515/jnum-2012-0013.

3

Reaction-diffusion fronts and the butterfly set

Jaime Cisternas, Kevin Rohe, Stefan Wehner

2020 *Chaos* **30**, 113138
doi: 10.1063/5.0022298

A single-species reaction-diffusion model is used for studying the coexistence of multiple stable steady states. In these systems one can define a potentiallike functional that contains the stability properties of the states, and the essentials of the motion of wave fronts in one and two dimensional space. Using a quintic polynomial for the reaction term, and taking advantage of the well known butterfly bifurcation, we analyse the different scenarios involving the competition of two and three stable steady states, based on equipotential curves and points. The predicted behaviours, including a front splitting instability, are contrasted to numerical integrations of reaction fronts in two dimensions.

The propagation of fronts in reaction–diffusion systems is relevant in a wide variety of contexts. For bistable situations (two homogeneous stable states in competition), there are abundant results for the propagation speeds. For situations where three stable states coexist, general results are more scarce. We propose here the use of bifurcation theory for a complete characterization of the existence of fronts.

3.1 Introduction

Reaction–diffusion models have proved to be relevant in chemistry, biology, and physics, providing a unitary framework that can account for spatiotemporal patterns, localized structures, traveling waves, spirals, and more complex phenomena. Several extensions of the basic model such as density dependent diffusion, crossdiffusion terms, or time-delays can be considered for making reaction– diffusion models more realistic and for enlarging their range of applicability [1–5].

In this work, we focus on fronts or moving interfaces between two or more stable homogeneous states, in domains that are large enough so that their boundaries can be reasonably ignored. We have in mind, particularly, surface reactions in ultra-high-vacuum conditions such as CO oxidation on platinum group crystals. For the time being, we focus on a single reacting species. Models considering the interaction of multiple species participating in several reactions can exhibit additional bifurcations that go beyond the scope of the present work.

The mathematical problem of predicting the speed of these reaction fronts has been considered extensively in the literature. Under certain conditions, the speed can be obtained without explicit knowledge of the profile of the reaction fronts [6, 7]. Interestingly, zero-velocity fronts have been observed in experiments and in realistic models, indicating equal stability between two homogeneous states [6, 8–12].

When multiple stable states coexist, as in the models considered in Refs. 13–15, the competition between states leads to competition between fronts of different speeds. Other works that have studied the motion of reaction fronts in a tristability regime are Refs. 14 and 16–19. In the context of metastable phases. They found that when three stable phases coexist, not all three possible fronts will exist; and that there are situations when a metastable phase grows transiently before being dominated by the most stable phase. Bifurcations of front solutions connecting two stable states are also possible, but only for reaction–diffusion models with two or more interacting species [14, 20–22]. For instance, Chirilus-Bruckner *et al.* [22] found a rich set of bifurcations, predicting the existence of up to three stable fronts with different speeds connecting the same two stable states, something that is not possible in single-species models.

In this article, our goal was to take advantage of the framework developed in Catastrophe Theory (see, for instance, Ref. 23) and analyze a minimal model of front propagation between three stable states and all its possible scenarios based on the features of a potential function. We do not compute speeds or actual solutions, but only study the algebraic conditions that make the speeds of these fronts equal to zero. The results of our study will shed light on other systems where three states coexist stably, such as Refs. 13 and 15 or 24–26.

In Sec. 3.2, we present the model and summarize simple mathematical results. In Sec.3.3, we pose the problem of equipotential curves and points and show results for three different sections

of the butterfly bifurcation. In Sec. 3.4, we derive conditions for the existence of fronts based on the mechanical analogy and an analysis around the equipotential point. In Sec. 3.5, we locate the curve in parameter space where the front splitting instability (FSI) arises. In Sec. 3.6, we project the existence of a tristability region and equipotential point back to the butterfly bifurcation set. In Sec. 3.7, we will show partial differential equation (PDE) simulations that confirm the results of Secs. 3.4-3.6. We conclude with a discussion on the implications for other reaction–diffusion systems.

3.2 Model

The simplest model of competition between stable homogeneous states is the single-species reaction–diffusion equation (also known as the Allen–Cahn equation in some contexts) in an infinite 1D domain,

$$u_t = u_{xx} + f(u), \tag{3.1}$$

where the nonlinear function $f(u)$ represents the local reaction rate and determines the existence of homogeneous states and their stability. In the following, we specialize our analysis to front solutions of the form $u(x,t) = u(z)$ with $z = x - ct$, and

$$u(-\infty) = u_-, u(+\infty) = u_+,$$

fixed points of the reaction function

$$f(u_-) = f(u_+) = 0,$$

which are both stable under the ODE dynamics $\dot{u} = f(u)$,

$$f'(u_-) < 0, \;\; f'(u_+) < 0.$$

The invasion of an *unstable* state ($f(u) = 0, f'(u) > 0$) may be relevant in certain contexts, but we have decided to narrow the scope of our analysis to connections between strictly stable states. Under these general assumptions, we end up with an ODE for the front solution $u(z)$,

$$0 = u_{zz} + cu_z + f(u). \tag{3.2}$$

Multiplying the front equation by u_z and integrating by parts along the whole real line, one can identify the front speed,

$$c = -\frac{\int_{u_-}^{u_+} f(u)\mathrm{d}u}{\int_{-\infty}^{\infty} u_z^2 \mathrm{d}z} = \frac{Vu_+ - V(u_-)}{\int_{-\infty}^{\infty} u_z^2 \mathrm{d}z}, \tag{3.3}$$

where the potential

$$V(u) \;\text{ satisfies }\; V'(u) = -f(u)$$

.

The stable states $u_\pm$ are now minima of potential $V(u)$. So, even if exact solutions $u(z)$ cannot be found explicitly, one can predict the direction in which the front will move.

The long-time evolution of an initial condition $u(x,0)$ when two stable states coexist has been studied rigorously by Fife and McLeod [16] and Aronson and Weinberger [27]. They proved, using different sets of assumptions, that appropriate initial conditions evolve into the monotonic traveling front $u(x \pm ct)$ joining the two stable states.

For smaller domains, their geometry and boundary conditions also play a role, as was studied, for instance, in Ref. 28. The curvature of fronts can become relevant and allow "frozen" fronts (with zero speed) that are not possible in simpler geometries, as discussed in Ref. 11. In those situations, the speed of the curved front presents a correction, which can be connected with the critical radius a circular area must have before growing.

Equations 3.2 and 3.3 allow a simple mechanical analogy that has been extensively used (see, for instance, Ref. 2): if z is interpreted as time, u as position, c as a (positive) friction, then $-V$ plays the role of a potential energy and the front solution $u(z)$ can be interpreted as the trajectory of a particle of unit mass from the position u_- in the infinite past moving toward u_+ in the infinite future. Both $u\pm$ (minima of $V(u)$) correspond to "hills" in this analogy. Assuming for the sake of the argument that $V(u_-) < V(u_+)$, then c can be interpreted as the critical friction that allows the particle to reach u_+ in the infinite future: $(u(z), u_0(z)) \xrightarrow{z\to\infty} (u_+, 0)$. Now, as was pointed out by Bechhoefer *et al.* [17], this condition for the existence of fronts is not sufficient, as intermediate obstacles may prevent the particle from reaching u_+ and staying there.

To analyze more deeply the effects of multiple coexisting stable states, we propose using a fifth-order reaction function,

$$f_{\alpha\beta\gamma\delta}(u) = -u^5 - \alpha u^3 - \beta u^2 - \gamma u - \delta, \tag{3.4}$$

which may have one, two, or three roots that satisfy $f(u) = 0$ and $f'(u) < 0$, depending on the values of the coefficients. A fourthorder term is not considered since it can be eliminated via a linear transformation.

The previous choice leads to a sixth-order potential,

$$V_{\alpha\beta\gamma\delta}(u) = \frac{1}{6}u^6 + \frac{\alpha}{4}u^4 + \frac{\beta}{3}u^3 + \frac{\gamma}{2}u^2 + \delta u, \tag{3.5}$$

as considered in Catastrophe Theory (see, for instance, Ref. 23). This polynomial, which always has at least one minimum, is the most general unfolding of a degenerate potential of one variable that has five derivatives equal to zero at the same point. By varying the four coefficients α, β, δ, and γ the minima of the potential undergo all the possible transitions between the monostable, bistable, and tristable scenarios.

Figure 3.1 shows several sections of the butterfly bifurcation, giving a simplified description of the role of the four parameters. In these depicted sections, the curves separate regions with different numbers of stable roots. In this work, we are mostly interested in regions, where two and three stable where two and three stable roots coexist and front solutions of 3.2 are possible.

In the language of bifurcation theory, the curves in Fig. 3.1 are saddle-node bifurcations since

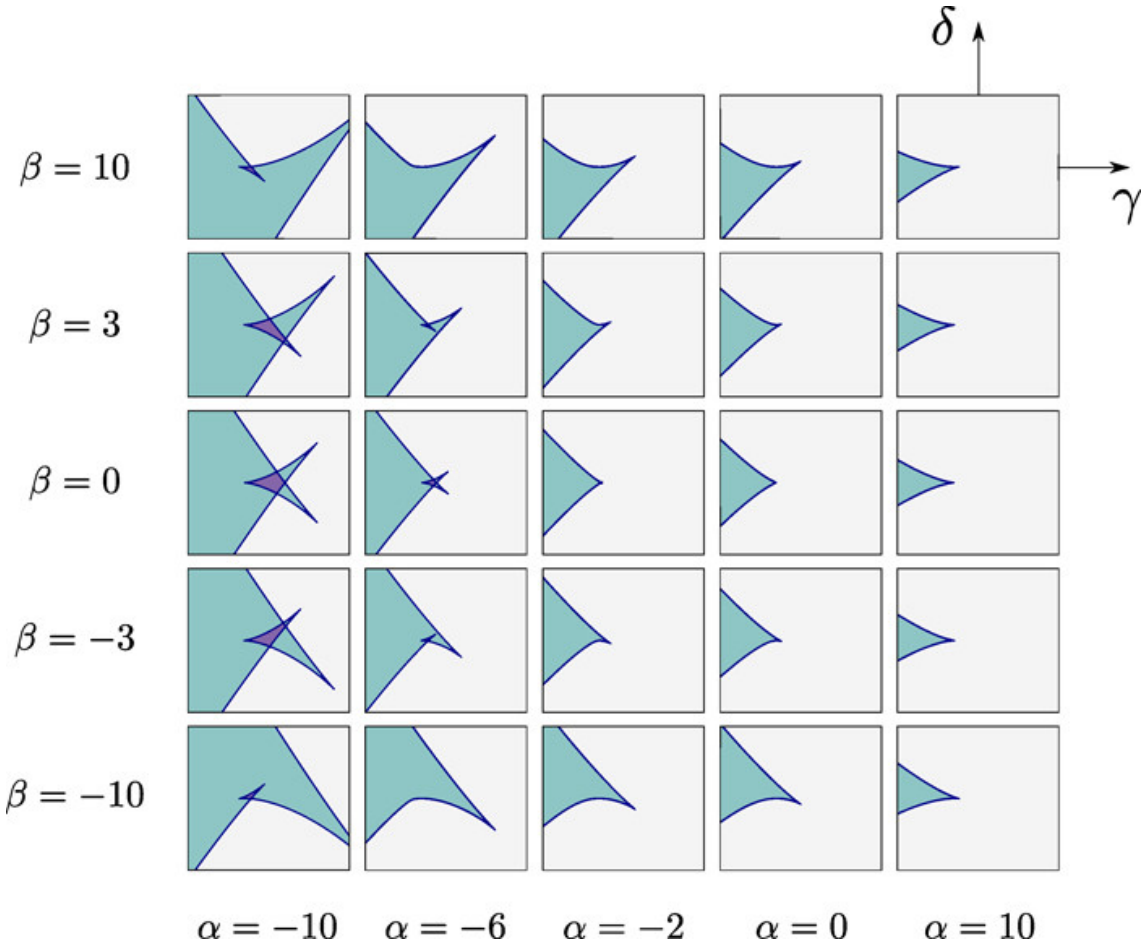


Figure 3.1: Butterfly catastrophe map, depicted as sections of four-dimensional parameter space $(\alpha, \beta, \gamma, \delta)$. Each small square corresponds to a particular choice of α and β and shows curves that indicate where minima of potential $V(u)$ defined in 3.5 appear or disappear. Regions of one stable solution are colored in light gray; two stable solutions in dark cyan; and three stable solutions in magenta. Based on figure by Poston and Stewart [23].

a stable and an unstable solution of $\dot{u} = f(u)$ annihilate at these points. The points where the saddlenode curves develop "cusps" are places, where either two stable solutions and one unstable solution meet (here, cusps that point in direction of growing γ) or one stable and two unstable solutions meet (here, the cusp that points in direction of decreasing γ). For certain particular sections (here, particular selections of α and β), these cusps merge into higher codimension bifurcation points, where whole regions of two or three stable states disappear.

Quintic reaction terms, similar to 3.4, have been considered in the literature of front solutions, but under a different light. For instance, van Saarloos [29] (see also Refs. 6 and [9]) found analytic solutions of the front equation for quintic reaction functions of the form $f(u) = -u^5 + \nu u^3 + \mu u$. Bechhoefer *et al.* [17, 18] considered the existence and interaction of fronts using reaction functions from a restricted set of polynomials $f(u) = -u(u-(1/2-b))(u-1)(u-3/2)(u-2)$. More recently, Stegemerten *et al.* [14] used $f(u) = -u^5 + (5/4)u^3 - (1/4)u + \mu$ and derived intricate sets of solutions using continuation methods, including fronts between both stable and unstable states.

In the context of surface reactions (see, for instance, Ref. 30), quadratic and quintic terms arise from the stoichiometry of the reactions. Here, we use nonlinear terms to control the shape of the reaction term $f(u)$ and consider multiple solutions, which could capture qualitatively what is observed in more complex models, but without assuming a direct connection between each nonlinear term in 3.4 and a particular reaction. Thus, we use the quintic model as a "cartoon" model that allows analytical treatment and gives qualitative insights about the outcomes of complex reaction–diffusion models of surface reactions.

3.3 Equipotential curves and points

In the butterfly diagrams depicted in Fig. 1, there are regions with one, two, or three stable solutions of the fifth-order equation $f(u) = 0, f(u) < 0$ as defined in eq. 3.4. These roots indicate possible stable homogeneous solutions of eq. 3.1.

In regions with a single stable solution u_0, the only possible outcome of the reaction–diffusion model is a homogeneous state $u(x,t) = u_0$ for long times t.

In regions with two stable solutions u_0, u_1 (we assume $u_0 < u_1$), it is possible to have moving fronts $u_{01}(x - c_{01}t)$ connecting $u(-\infty) = u_0$ and $u(\infty) = u_1$, and the direction of motion of the front will depend on the sign of the speed c_{01} i.e., on the values $V(u_0)$ and $V(u_1)$ as predicted by eq. 3.3. If $V(u_0) < V(u_1)$, then $c_{01} > 0$ and state u_0 will invade parts of the space initially covered with u_1; if $V(u_0) > V(u_1)$, then $c_{01} < 0$ and it will be state u_1 invading the whole spatial domain (away from boundaries or defects).

Now, in regions of parameter space with three stable solutions u_0, u_1, u_2 (we assume $u_0 < u_1 < u_2$), there will be up to three front solutions $u_{01}(z)$, $u_{12}(z)$, $u_{02}(z)$ with speeds c_{01}, c_{12}, and c_{02}, which we label 01, 12, and 02 indicating the values at $z = -\infty$ and $z = +\infty$. As before, the relative values of $V(u_0)$, $V(u_1)$, and $V(u_2)$ determine the signs of the speeds and indicate the directions of motion of the fronts: the state with the minimum value of $V(u)$ will invade the other two. However, the details of the process will depend on whether the fronts connecting the three states do exist or not, a question that will be answered in Sec. 3.4.

In this section, we focus on zero-velocity fronts $c = 0$, which exist when there are two stable states u_- and u_+ that satisfy $f(u) = 0, f'(u) < 0$ and have the same potential values,

$$V(u_-) \stackrel{\text{ep}}{=} V(u_+) \tag{3.6}$$

(here, "$\stackrel{\text{ep}}{=}$" means that the equation is valid if both sides are evaluated at particular combinations of the parameters α, β, γ, and δ). This Maxwell or *equipotential condition* indicates that in the absence of other forcings, there are two stable states with the same stability properties. From 3.2, multiplying by $\mathrm{d}u/\mathrm{d}z$ and integrating from $-\infty$ to z (assuming $u_- < u_+$),

$$\frac{u_z^2(z)}{2} \stackrel{\text{ep}}{=} V(u(z)) - V(u_-). \tag{3.7}$$

In the context of the mechanical analogy, these zero-friction solutions are known as *Hamiltonian solutions*.

For bistable surface reactions, the region in the parameter space where two stable solutions coexist is delimited by two saddle-node bifurcations. It has been shown by Bär *et al.* [8] and Wehner *et al.* [11], both in experiments and modeling, that if the width of the bistability region shrinks and disappears in a cusp, then the curve where $V(u_0) = V(u_1)$ will reach the tip of the cusp, separating the bistability region into two subregions where either the u_0 or u_1 state dominates.

In the case of a tristable reaction, i.e., inside a tristability region such as the ones depicted in Fig. 3.1, one can study (in parameter space α, β, γ, δ) the existence of *equipotential curves* where $V(u_0) = V(u_1)$, $V(u_1) = V(u_2)$, or $V(u_0) = V(u_2)$, and of *equipotential points* where $V(u_0) = V(u_1) = V(u_2)$. For these particular points in parameter space, the long term behavior of solutions of the reaction–diffusion 3.1 will look like a subdivision of the spatial domain into patches where $u(x,t)$ is constant and uniform.

In the context of the sixth-order potential 3.5, the equipotential condition leads to two simple bifurcation problems. The equipotential curves are solution sets of the following system of algebraic equations (for given α, β parameters):

$$f(u_0;\gamma,\delta) = 0, \tag{3.8a}$$
$$f(u_1;\gamma,\delta) = 0, \tag{3.8b}$$
$$V(u_0;\gamma,\delta) - V(u_1;\gamma,\delta) = 0, \tag{3.8c}$$

where u_0 and u_1 are distinct solutions of $f(u;\alpha,\beta,\gamma,\delta) = 0$.

The equipotential points are solutions of the following system of algebraic equations (for given α, β parameters):

$$f(u_0;\gamma,\delta) = 0, \tag{3.9a}$$
$$f(u_1;\gamma,\delta) = 0, \tag{3.9b}$$
$$f(u_2;\gamma,\delta) = 0, \tag{3.9c}$$
$$V(u_0;\gamma,\delta) - V(u_1;\gamma,\delta) = 0, \tag{3.9d}$$
$$V(u_1;\gamma,\delta) - V(u_2;\gamma,\delta) = 0, \tag{3.9e}$$

where u_0, u_1, and u_2 are distinct solutions of $f(u;\alpha,\beta,\gamma,\delta) = 0$.

Numerical analysis of algebraic systems (3.8) and (3.9) can be performed with continuation algorithms such as the software AUTO [31]. Assuming that appropriate initial solutions were used, it is possible to reconstruct the whole codimension-2 equipotential curves in the plane (γ, δ) for a given choice of α, β, revealing the inner structure of the bistability and the tristability regions. In Secs. 3.3.1-3.3.3, three qualitatively different sections of the butterfly bifurcation set are analyzed.

3.3.1 First scenario: One bistability region and monostability region

The simplest sections of the butterfly bifurcation set are those without a tristability region in the (γ, δ) plane. For these scenarios, the cusp-shaped bistability region where fronts are possible will

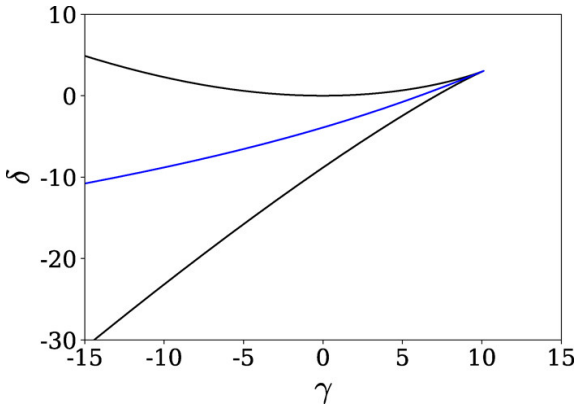


Figure 3.2: The scenario of the butterfly bifurcation is here presented for $\alpha = 2$ and $\beta = 10$. There are two saddle-node bifurcation curves (black) that separate regions with one and two stable states and meet at a cusp: in the area contained by the two saddle-node curves, two stable solutions coexist. The bistability region is divided by an equipotential curve (blue), solution of 3.8, where a zero-velocity front connecting the two states is possible.

be divided by a single equipotential curve $V(u_0) = V(u_1)$, solution of (3.8), which ends at the tip of the cusp, as depicted in Fig. 3.2. At that particular point, the solutions u_0 and u_1 swap roles and the curve retraces back.

For points (γ, δ) above the equipotential curve, $V(u_0) < V(u_1)$, so solution u_0 is the invading state and state u_1 the one being invaded; for points (γ, δ) below the equipotential curve $V(u_0) > V(u_1)$, it is the other way around. This scenario describes what was found in surface reactions experiments like [11].

3.3.2 Second scenario: One tristability region, one bistability region, and one monostability region

For some selection of α and β parameters, a pair of cusps develops along one of the saddle-node curves as depicted in Fig. 3.3. Inside the triangle-shaped region defined by the newly created cusps, three stable states coexist.

Now, only one of the "new" cusps has an equipotential curve that divides the tristability region and ends at the opposing saddle-node curve; at that point, the equipotential curve bounces back and keeps going but it now captures one stable and one *unstable* state having the same potential and then bounces at another saddle-node curve and finally ends at the other new cusp. The "spurious" sections of the curves, indicating an equipotential condition involving one or two unstable solutions, are indicated in red in Fig. 3.3.

The "older" cusp (the one pointing to the right) that penetrates into the monostable region does have an equipotential curve that traces to the left. Depending on α and β, this equipotential curve may or may not pass through the tristability region; so, two subscenarios must be considered. If the curve does touch the saddle-node curves and bounces two times at the saddle-node curves, as shown in Fig. 3.3(a), a tristability point $(\gamma_{ep}, \delta_{ep})$ solution of (3.9) will exist; if there is no such intersection and the curve goes unhindered in the direction of decreasing γ, as shown in Fig. 3.3(b), no equipotential point will exist.

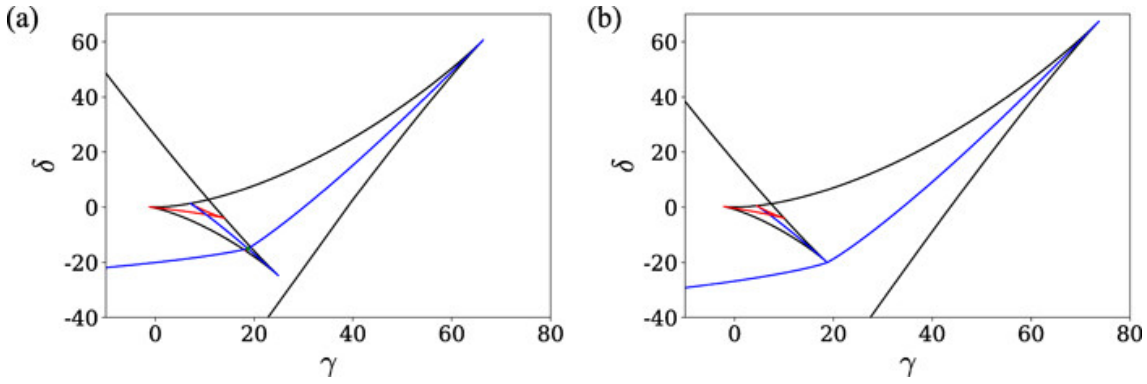


Figure 3.3: The scenario of the butterfly bifurcation set is here depicted for (a) $\alpha = -10, \beta = 6$ and (b) $\alpha = -10, \beta = 8$. There are several saddle-node bifurcation curves, depicted in black, that separate regions with one, two, and three stable states (inside the upper "horn" two stable solutions coexist, inside the central triangular region three stable solutions coexist). The equipotential curves, depicted in blue, are solutions of 3.8. "Spurious" sections of the equipotential curves (involving unstable solutions) are depicted in red. In (a), the equipotential curves intersect defining an equipotential point that is the unique solution of (3.9): at that point, three zero-velocity fronts exist. In (b), there is no intersection: although the three states coexist stably in the triangular region, only two of them can be connected by a zero-velocity front.

A closer look at the triangle-shaped tristability region for the case where an equipotential point exists reveals that there are six distinct subregions where the values of the potential $V(u_0)$, $V(u_1)$, and $V(u_2)$ will have different orderings. These orderings are going to be described in more detail in the next scenario.

3.3.3 Third scenario: One tristability region, three bistability regions, and one monostability region

The third situation corresponds to selections of (α, β) that lead to (γ, δ) planes with two bistable cusps penetrating the monostable region and a diamond-shaped tristability region as shown in Fig. 3.4. Comparing this scenario with the one just considered in Sec. 3.3.2, it looks as if one of the new cusps grew to the right and crossed the other saddle-node curve.

The two equipotential curves born at the right-pointing cusps intersect inside the tristability region at a unique equipotential point $(\gamma_{ep}, \delta_{ep})$ and end at opposing saddle-node curves, where they bounce and become spurious. One of these spurious curves bounces a second time before reaching the tip of the left-pointing cusp. The other spurious curve becomes legitimate after bouncing at a saddle-node, passes through the equipotential point, and then goes unhindered to the left.

A closer look at the subregions of (γ, δ) plane defined by the equipotential curves reveals that there are six distinct cases: different orderings for the values of the potential $V(u_0)$, $V(u_1)$, and $V(u_2)$. Figure 3.6 shows the potentials $V(u)$ for some limited selection of parameters $(\alpha, \beta, \gamma, \delta)$, including the equipotential point shown in Fig. 3.4. Figure 3.5 shows two different sections of Fig. 3.4, indicating the roots, their stability, and the places where equipotential conditions are verified.

For $\beta = 0$, the diagrams in the plane (γ, δ) become symmetric to reflections with respect to γ axis. This case is not relevant in the context of fronts and is not further analyzed. A summary of

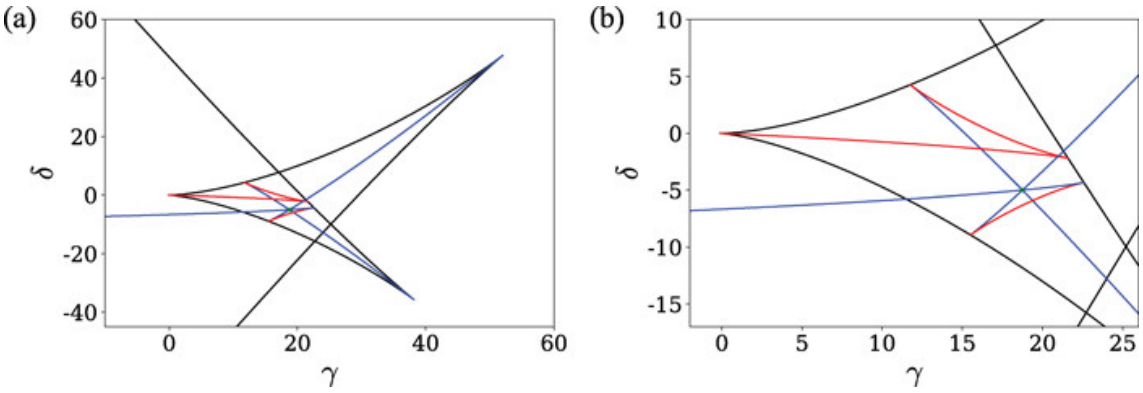


Figure 3.4: (a) The scenario of the butterfly bifurcation is here shown for $\alpha = -10, \beta = 2$. There are several saddle-node bifurcation curves depicted in black that separate regions with one, two, and three stable states: inside the "wings of the butterfly" two stable solutions coexist, and inside the diamond-shaped region, three stable solutions coexist. The equipotential curves depicted in blue are solutions of 3.8, and the equipotential point is the unique solution of 3.9 and the intersection of three equipotential curves. (b) A zoom of the diamond-shaped tristability region. Red segments of equipotential curves are spurious since they represent an equipotential condition between astable and an unstable state.

the possible orderings of the values $V(u_0)$, $V(u_1)$, and $V(u_2)$ is presented in Fig. 3.7. The colored areas indicate whether u_0, u_1 or u_2 is the minimum of $V(u)$, and thus will invade the other two states in the context of a surface reaction modeled by 3.1.

3.4 Existance of fronts inside the tristability region

In Sec. 3.3, and using only the features of the potential $V(u)$ as its polynomial coefficients were varied, we identified up to six subregions where the stable roots u_0, u_1, and u_2 have their potential values $V(u_0)$, $V(u_1)$, and $V(u_2)$ ordered in different ways. The global minimum of V(u) will always be the invading state that in the long term will cover the whole domain. The details of this process will depend on whether the front solutions $u_{01}(z)$, $u_{12}(z)$, and $u_{02}(z)$ exist or not.

We illustrate this issue with the mechanical analogy: a rightmoving $(c > 0)$ front connecting u_- and u_+ is only possible if $V(u_- < V(u_+)$, and the starting hill must be higher. However, there is an additional requirement: no intermediate hill should block the path between u_- and u_+.

In the case of three hills (three local minima of $V(u)$), there are always front connections (moving either to the left or to the right) between u_0 and u_1, and between u_1 and u_2. But the $u_{02}(z)$ front connecting u_0 and u_2 deserves a separate treatment.

As it turns out there are three regions in the (γ, δ) plane. We follow the treatment used by Bechhoefer *et al.* [17].

3.4.1 First subregion: Intermediate hill is highest

Using the mechanical analogy for the case when $V(u_1) < V(u_0)$ and $V(u_1) < V(u_2)$, we can see that both $u_{10}(z)$ and $u_{12}(z)$ fronts exist with speeds c_{10} and c_{12} given by (3.3), but fronts connecting u_0 and u_2 are forbidden in both directions (the mechanical particle does not have enough energy

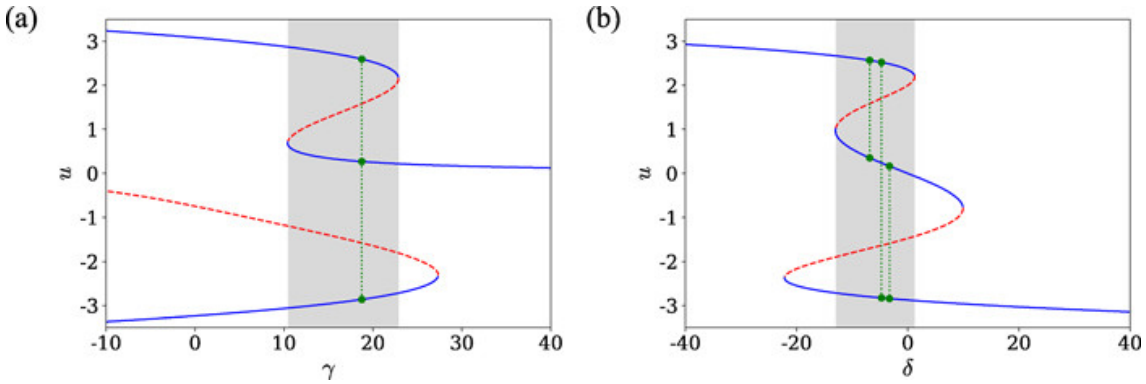


Figure 3.5: Codimension-1 bifurcation diagrams showing the roots of function $f(u)$ as a single parameter is varied. These two diagrams correspond to a horizontal and a vertical cut in Fig. 3.4, respectively. Stable roots are depicted in continuous blue lines; unstable roots in dashed red lines; tristability ranges are indicated with shaded areas. (a) Parameter γ is varied, keeping $\alpha = -10, \beta = 2$, and $\delta = -5$ constant. The three green points indicate that the condition $V(u_0) = V(u_1) = V(u_2)$ is satisfied for $\gamma = 18.75$. (b) Parameter δ is varied, keeping $\alpha = -10, \beta = 2$, and $\gamma = 20$ constant. The three pairs of green points connected by dotted lines indicate that an equipotential condition is satisfied for three different values of δ.

to climb the intermediate hill at u_1).

In the dynamic problem 3.1, state u_1 will dominate everywhere, invading both areas initially covered with states u_0 and u_2. Any interface between u_0 and u_2 will develop a thin u_1 band that will grow and invade both sides of the domain.

3.4.2 Second subregion: Intermediate hill is lowest

If $V(u_1) > V(u_0)$ and $V(u_1) > V(u_2)$, then all three fronts are possible: $u_{01}(z)$, $u_{21}(z)$, and $u_{02}(z)$ (or $u_{20}(z)$ depending on relative values of potential $V(u_0)$ and $V(u_2)$).

In the dynamic problem, either u_0 or u_2 will dominate depending on their potential values, and u_1 will be invaded by both. Areas initially covered by state u_1 will shrink and disappear, leaving behind an interface between u_0 and u_2, which then will move as the state with smaller $V(u)$ dominates on the whole spatial domain.

3.4.3 Third subregion: "Interesting" case

If $V(u_0) < V(u_1) < V(u_2)$ [or the "mirror" case $V(u_0) > V(u_1) > V(u_2)$], then the fronts $u_{01}(z)$ and $u_{12}(z)$ will always exist, but the front $u_{02}(z)$ may or may not exist. So, we will have an additional subdivision in this subregion.

In the dynamic problem, the state of lowest potential (either u_0 or u_2) will always invade the other two in the long term. However, for intermediate times, whether the front $u_{02}(z)$ exists or not will show up as an *instability* in parameter plane.

If the front 02 does exist, the speeds of the three fronts verify $c_{01} > c_{02} > c_{12}$ [in the mirror case $V(u_0) > V(u_1) > V(u_2)$, the order is $c_{21} > c_{20} > c_{10}$], as explained in Ref. 17 using the mechanical analogy: if there is a connection between u_0 and u_2 for a critical friction c_{02}, then a larger critical friction c_{01} will make the particle stop at u_1; if connection u_{02} passes through u_1 before reaching

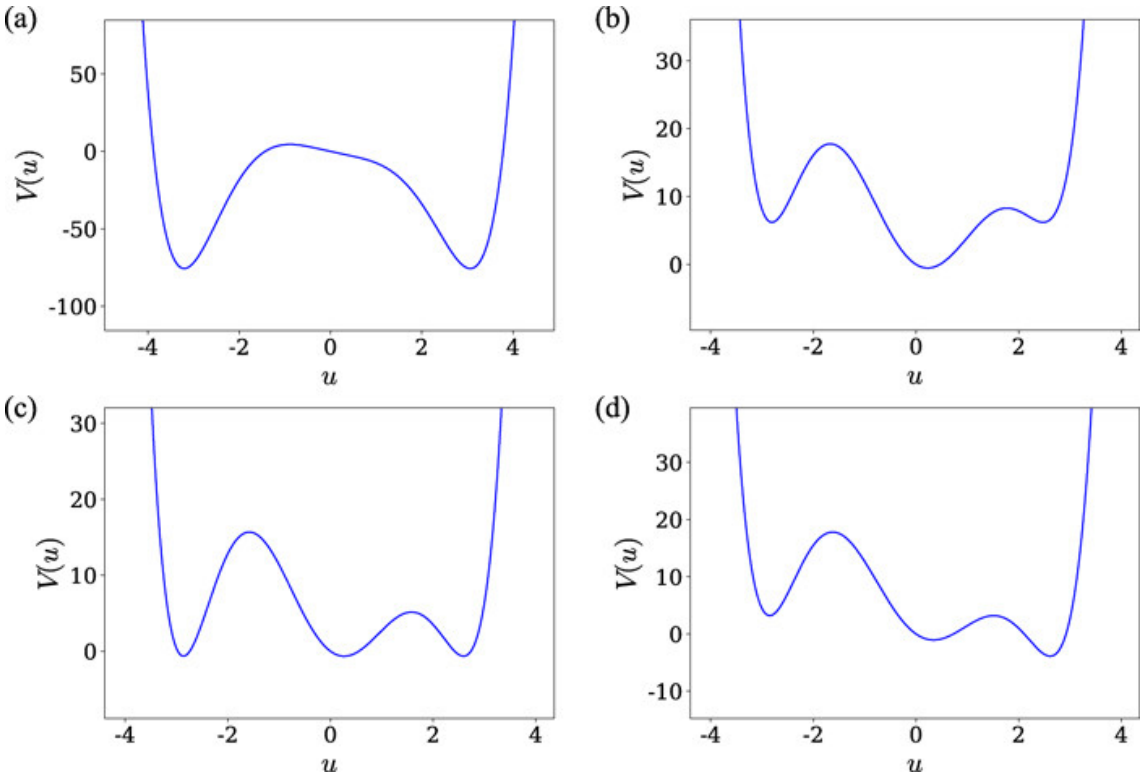


Figure 3.6: Potentials $V(u)$ obtained for different combinations of parameter values, where one or two equipotential conditions are satisfied: (a) $\alpha = -10$, $\beta = 2$, $\gamma = 1.636$, $\delta = -6.558$, and two stable solutions exist and verify $V(u_0) = V(u_2)$; (b) $\alpha = -10$, $\beta = 2$, $\gamma = 20.655$, $\delta = -4.718$, and three stable solutions exist and verify $V(u_0) = V(u_2) > V(u_1)$; (c) $\alpha = -10$, $\beta = 2$, $\gamma = 18.750$, and $\delta = -5$, equipotential point where the three minima $V(u_0) = V(u_1) = V(u_2)$; (d) $\alpha = -10$, $\beta = 2$, $\gamma = 18.790$, $\delta = -6.294$, and a stable and an unstable solution have the same potential (spurious situation).

u_2, then a particle starting at u_1 can also reach u_2 but only for a smaller friction c_{12}.

Now if the front $u_{02}(z)$ does not exist, the order of the speeds reverses: $c_{01} < c_{12}$ [in the mirror case $V(u_0) > V(u_1) > V(u_2)$, the order is $c_{21} < c_{10}$]. In this case, the interesting and paradoxical phenomena of *front splitting* appear [17, 18] in the reaction–diffusion model: fronts 01 and 12 will move in the same direction with speeds $c_{01} < c_{12}$, leaving a growing area of the domain with the metastable state u_1. In the short term, the area that is covered with u_1 is the one that grows although it is continuously invaded. In the long term, it is the state with the lowest potential (either u_0 or u_2) that will eventually cover the whole domain. Such a scenario was analyzed by Fife and McLeod [16] (particularly in Theorem 3.3).

At a point in parameter space where $c_{01} = c_{02} = c_{12}$, the front $u_{02}(z)$ will be the result of merging $u_{01}(z)$ and $u_{12}(z)$, so it will stay around u_1 for an indefinite amount of "time" z.

Rubinstein *et al.* [19] presented a detailed analysis of the fronts close to the equipotential point where $V(u_0) = V(u_1) = V(u_2)$. They found that the fronts $u_{01}(z)$ and $u_{12}(z)$ exist independently and travel with slightly different speeds. For one spatial dimension and assuming $c_{01} > c_{12}$, they considered solutions where the two fronts approached each other until a repulsive interaction balanced the attraction, resulting in a stable "two-stepped" front as shown in Fig. 3.8 As we will show in Sec. 3.5, the fronts $u_{02}(z)$ exist in large regions of parameter space and progressively develop a

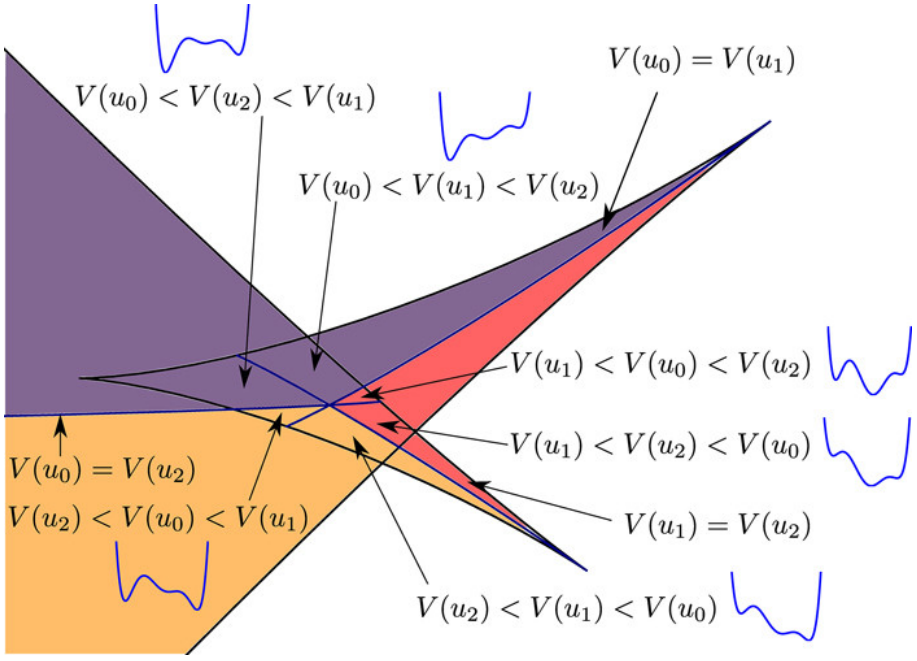


Figure 3.7: Summary of the behaviors of (3.1) as predicted by analysis of the potential $V(u)$ (here, we selected $\alpha = -10$, $\beta = 2$). The proper equipotential curves and the saddle-node curves in (γ, δ) plane define six subregions where the values $V(u_0), V(u_1)$, and $V(u_2)$ have different orderings (a schematic graph of the potential is shown for each one). The colored areas indicate the dominant state that will invade the other two states by moving fronts: u_0 (purple), u_1 (brick), or u_2 (mustard). The intersection of the curves $V(u_0) = V(u_1)$, $V(u_1) = V(u_2)$, and $V(u_0) = V(u_2)$ defines the equipotential point $(\gamma_{ep}, \delta_{ep})$, where the three fronts have the same speed. In the monostable region (not colored), no fronts are possible.

two-stepped profile not only near the equipotential point but in general near the curve in parameter space where the fronts $u_{01}(z)$ and $u_{12}(z)$ have the same speed. Beyond this curve, the front u_{02} does not exist.

3.5 Front splitting instability

Assuming that $V(u_0) < V(u_1) < V(u_2)$ [or the "mirror" case $V(u_0) > V(u_1) > V(u_2)$], answering the question of whether the front solution $u_{02}(z)$ exists for some c_{02} will require solving a PDE, shooting an ODE, or implementing a continuation scheme as in Refs. 14 and 32. We could use variational methods that do not require explicit knowledge of the solution $u(z)$ (see, for instance, Ref. 7 for a general overview of variational methods in this setting). All these methods are approximate and have their own strong and weak points.

In this work, we have favored elementary arguments over more complex analyses. In this spirit, we present here an estimate of the transition boundary where the u_{02} front is born, close to the equipotential point where the three fronts have zero speed.

Let us first compute partial derivatives of the front speed with respect to parameters γ and δ.

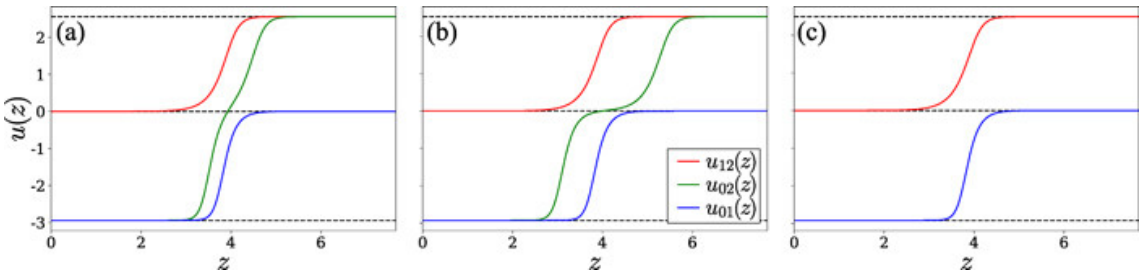


Figure 3.8: Front solutions of (3.2) connecting u_0, u_1, and u_2, stable roots of $f(u)$ (indicated by horizontal dashed lines). Parameters $\alpha = -10, \beta = 2$, and $\delta = 0$ are kept constant as δ takes three values: (a) for $\delta = 17.5$, three right-moving fronts $u_{01}(z)$, $u_{12}(z)$, and $u_{02}(z)$ exist, with speeds satisfying $c_{01} > c_{02} > c_{12}$; (b) for $\delta = 18.0$, the front $u_{02}(z)$ acquires a "two-stepped" profile and $c_{01} \gtrapprox c_{02} \gtrapprox c_{12}$; and (c) for $\gamma = 18.5$, the front $u_{02}(z)$ no longer exists and $c_{01} < c_{12}$. Although the two-stepped front $u_{02}(z)$ may look like a combination of the fronts $u_{01}(z)$ and $u_{12}(z)$, these three solutions correspond to three different speeds.

Recalling (3.3), for any front connecting u_- and u_+,

$$c \int_{-\infty}^{\infty} u_z^2 \mathrm{d}z = V(u_+) - V(u_-) \tag{3.10}$$

where both sides are positive.

Differentiating with respect to γ and evaluating at an equipotential point where $c = 0$ (as was proposed in Ref. 9),

$$\frac{\partial c}{\partial \gamma} \int_{-\infty}^{\infty} u_z^2 \mathrm{d}z \overset{\text{ep}}{=} \frac{\partial V(u_+)}{\partial \gamma} - \frac{\partial V(u_-)}{\partial \gamma} = \frac{u_+^2}{2} - \frac{u_-^2}{2} \tag{3.11}$$

The fact that u_+ and u_- are roots of $f(u)$ and verify the equipotential condition $V(u_+) = V(u_-)$ allowed us to eliminate several terms in the previous equation and will help us to find a simplified expression for the partial derivative. For the integral in the left hand side, we can use (3.7) integrated over the whole real line, which leads to $\int_{-\infty}^{\infty} u_z^2 \mathrm{d}z \overset{\text{ep}}{=} \int_{u_-}^{u_+} \sqrt{2(V(u) - V(u_-))} \mathrm{d}u$. We then obtain for the partial derivative of the speed with respect to γ at the zero-speed point,

$$\frac{\partial c}{\partial \gamma} \overset{\text{ep}}{=} \frac{\frac{u_+^2}{2} - \frac{u_-^2}{2}}{\int_{u_-}^{u_+} \sqrt{2(V(u) - V(u_-))} \mathrm{d}u} \tag{3.12}$$

Repeating the process for the partial derivative of the speed with respect to δ at the zero-speed point,

$$\frac{\partial c}{\partial \delta} \overset{\text{ep}}{=} \frac{u_+ - u_-}{\int_{u_-}^{u_+} \sqrt{2(V(u) - V(u_-))} \mathrm{d}u} \tag{3.13}$$

Now, considering the difference of speeds $c_{01} - c_{12}$ as a function of parameters γ and δ, we get the following expression, valid at the point $(\gamma_{\text{ep}}, \delta_{\text{ep}})$ where both fronts have zero speed:

$$\frac{\partial(c_{01}-c_{12})}{\partial\gamma} \underline{\underline{\text{ep}}} \frac{\frac{u_1^2}{2}-\frac{u_0^2}{2}}{\int_{u_0}^{u_1}\sqrt{2(V(u)-V(u_0))}\mathrm{d}u} \frac{\frac{u_2^2}{2}-\frac{u_1^2}{2}}{\int_{u_1}^{u_2}\sqrt{2(V(u)-V(u_1))}\mathrm{d}u}, \tag{3.14a}$$

$$\frac{\partial(c_{01}-c_{12})}{\partial\delta} \underline{\underline{\text{ep}}} \frac{u_1-u_0}{\int_{u_0}^{u_1}\sqrt{2(V(u)-V(u_0))}\mathrm{d}u} \frac{u_2-u_1}{\int_{u_1}^{u_2}\sqrt{2(V(u)-V(u_1))}\mathrm{d}u}, \tag{3.14b}$$

These partial derivatives make up a gradient vector in (γ, δ) space. Perpendicularly, from this vector, there is a line that passes through ((γ_{ep}, δ_{ep}) where $c_{01}-c_{12}=0$ at linear order. To the left of such a line, the front u_{02} does exist and has a speed c_{02} that sits between c_{01} and c_{12}; to the right of such a line, the front u_{02} does not exist and the front splitting instability arises.

The slope of this line is

$$\frac{\Delta\delta}{\Delta\gamma} = -\frac{(\partial(c_{01}-c_{12})/\partial\gamma)_{\text{ep}}}{(\partial(c_{01}-c_{12})/\partial\delta)_{\text{ep}}} \tag{3.15}$$

We could use this estimate for predicting the disappearance of the 02 front, i.e., the front splitting instability. For instance, using the section $\alpha=-10$, $\beta=2$, where $(\gamma_{\text{ep}},\delta_{\text{ep}})=(18.75,-5)$,

$$\frac{\Delta\delta}{\Delta\gamma} = -5.662$$

and focusing on the line $\delta=0$, inside the interesting region where $V(u_0)<V(u_1)<V(u_2)$, the transition takes place at $\gamma\approx 17.86$. For larger values, there will not be a 02 front.

For the selection $\alpha=-10$, $\beta=6$, where $(\gamma_{\text{ep}},\delta_{\text{ep}})=(18.75,-15)$,

$$\frac{\Delta\delta}{\Delta\gamma} = -1.991$$

For $\alpha=-10$, $\beta=0$, where $(\gamma_{\text{ep}},\delta_{\text{ep}})=(18.75,0)$, one can verify that $(\partial(c_{01}-c_{12})/\partial\delta)_{\text{ep}}=0$, and, thus, the curve of the front splitting instability is vertical.

Now, we must emphasize that the front splitting instability does not require the presence of an equipotential point in the bifurcation diagram.

Using simple continuation techniques implemented in AUTO it is possible to reconstruct the curve in parameter space (γ, δ) where both speeds are equal (for given α, β parameters),

$$D^2(u)_i + cD(u)_i + f(u_i;\gamma,\delta) = 0 \quad (i=1...N) \tag{3.16a}$$

$$D^2(v)_i + cD(v)_i + f(v_i;\gamma,\delta) = 0 \quad (i=1...N) \tag{3.16b}$$

$$\frac{\sum_{i=1}^{N} z_i (D(u)_i)^2}{\sum_{i=1}^{N}(D(u)_i)^2} = 0 \tag{3.16c}$$

$$\frac{\sum_{i=1}^{N} z_i (D(v)_i)^2}{\sum_{i=1}^{N} (D(v)_i)^2} = 0 \tag{3.16d}$$

where $\{u_i\}$ and $\{v_i\}$ represent the discretized fronts $u_{01}(z)$ and $u_{12}(z)$ at the z_i points; and operator D is the discrete derivative. The first two sets of N equations correspond to (3.2) for $u_{01}(z)$ and $u_{12}(z)$ implemented using finite differences. The last two equations pin the two fronts at the center of the discrete domain.

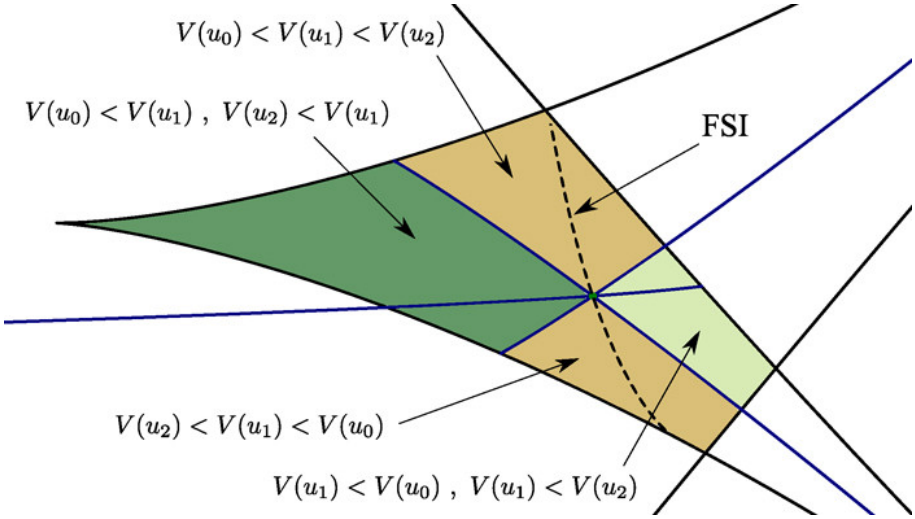


Figure 3.9: As explained in the text, the fronts 01 and 12 exist everywhere inside the tristability region but the front 02 only exists to the left of the dashed boundary where the front splitting instability (FSI) arises. The location of this curve is explained in Sec. 3.5. Colored subregions are analyzed separately in the text.

The resulting curve of solutions of system (3.16) for $\alpha = -10$, $\beta = 2$ was already presented in Fig. 3.9. The curve passes through $(\gamma_{\rm ep}, \delta_{\rm ep})$ with a slope $\Delta\delta/\Delta\gamma \approx -5.67$ that matches the analytical estimate (3.15). For $\alpha = -10$, $\beta = 6$, the curve passes through $(\gamma_{\rm ep}, \delta_{\rm ep})$ with a slope $\Delta\delta/\Delta\gamma \approx -1.99$ that also matches the analytical estimate.

It is interesting at this point to compare our findings with those presented by Stegemerten *et al.* [14] (particularly in the section "The passive cubic-quintic Allen–Cahn equation"). They analyzed the front (3.2) with a restricted quintic reaction term, finding a wealth of fronts between stable and unstable states (here, we only consider fronts between stable states) and nontrivial connections between them. Now, because of the particular choice of parameters they used, they observed a strict separation of front speeds $c_{01} < c_{02} < c_{12}$, and did not observe front splitting instabilities.

These observations and results will be contrasted with numerical integrations of 3.1 in Sec. 3.7.

3.6 The existance of equipotential points and the butterfly set

As it is studied in this article, stable states and their connections experience several transitions as parameters α, β, γ, andδ are varied. In this section, we go back to Fig. 3.1, where interesting combinations of (α, β) were emphasized, those that deserve to be analyzed in more detail with the equipotential condition and the tools of bifurcation theory.

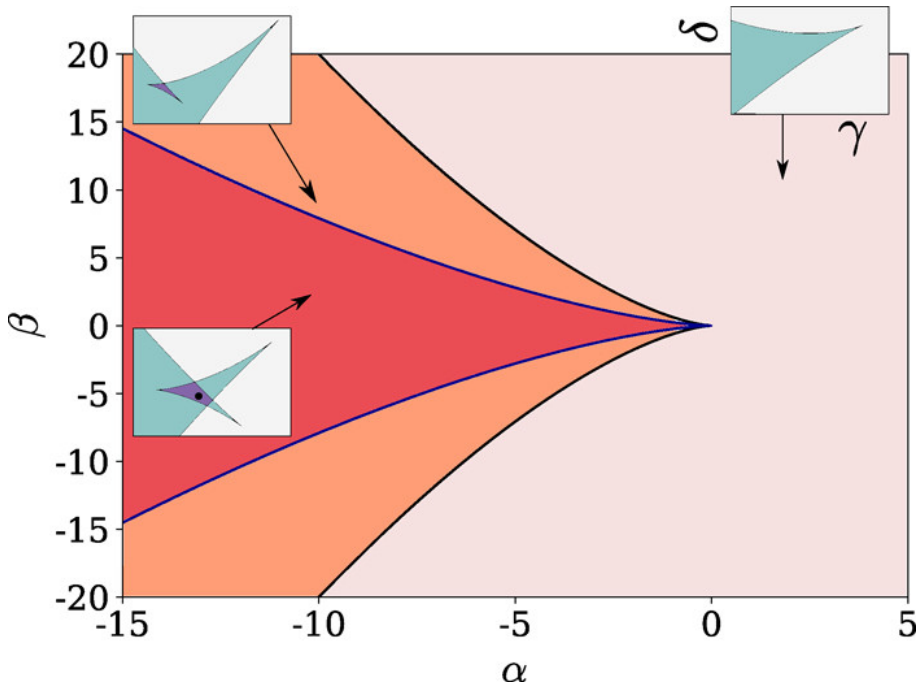


Figure 3.10: Projection of a bifurcation diagram in all four parameters α, β, γ, and γ. Each point in this (α, β) plane summarizes the features present in a (γ, δ) section. Three distinct subregions (identified with different colors) are identified, and a small representative (γ, δ) diagram is included for each one: points (α, β) inside the exterior curves have sections (γ, δ) with tristability regions; points (α, β) inside the interior curves have sections (γ, δ) where an equipotential point exists; outside the exterior curves, no tristability regions exist in the associated (γ, δ) sections.

First, we separate the points (α ,β) where the associated (γ, δ) plane contains a tristability region, no matter how small; loosely speaking, these points lie to the left of the origin in Fig. 3.1. The shrinking of the tristability region appears when two cusps collide and three minima of $V(u)$ become a single one. Points at a saddle-node satisfy $f(u) = f'(u) = 0$; points right at a cusp satisfy $f(u) = f'(u) = f''(u) = 0$. For the collision of cusps, an additional equation should be included.

The points in the (α, β) plane where two cusps coincide can be found by the following algebraic system:

$$f(u;\alpha,\beta,\gamma,\delta) = 0, \tag{3.17a}$$
$$f'(u;\alpha,\beta,\gamma,\delta) = 0, \tag{3.17b}$$
$$f''(u;\alpha,\beta,\gamma,\delta) = 0, \tag{3.17c}$$
$$f'''(u;\alpha,\beta,\gamma,\delta) = 0, \tag{3.17d}$$

The solution set of this system has the shape of a cusp that points in the α direction, is symmetric to reflections in theβ axis, and has its tip at the (0, 0) point. This curve is a boundary: for points (α, β) inside the cusp, there are combinations of (γ, δ) where $f(u)$ has three stable solutions (besides regions with one and two); for points outside the cusp, there are combinations of (γ, δ) that have at most two stable states.

In Sec. 3.3, we showed that for a given (α, β) where there is a tristability region, there is no guarantee that equipotential curves intersect. The boundary where a cusp touches another equipotential curve can also be found with a system of equations,

$$f(u_1;\alpha,\beta,\gamma,\delta) = 0, \tag{3.18a}$$
$$f'(u_1;\alpha,\beta,\gamma,\delta) = 0, \tag{3.18b}$$
$$f''(u_1;\alpha,\beta,\gamma,\delta) = 0, \tag{3.18c}$$
$$f(u_2;\alpha,\beta,\gamma,\delta) = 0, \tag{3.18d}$$
$$V(u_1;\alpha,\beta,\gamma,\delta) - V(u_2;\alpha,\beta,\gamma,\delta) = 0 \tag{3.18e}$$

The solution set of this system has also the shape of a cusp that points in the α direction, is symmetric to reflections in the β axis, and has its tip at the (0, 0) point. For points (α, β) inside the cusp, the associated (γ, δ) will show equipotential curves that intersect at an equipotential point (γ_{ep}, δ_{ep}).

Both cusps are shown in Fig. 3.10. This illustration complements the original butterfly map presented in Fig. 3.1, explaining the different scenarios of the reaction–diffusion model (3.1) when the reaction term is derived from a sixth-order polynomial (3.5).

3.7 PDE Solutions

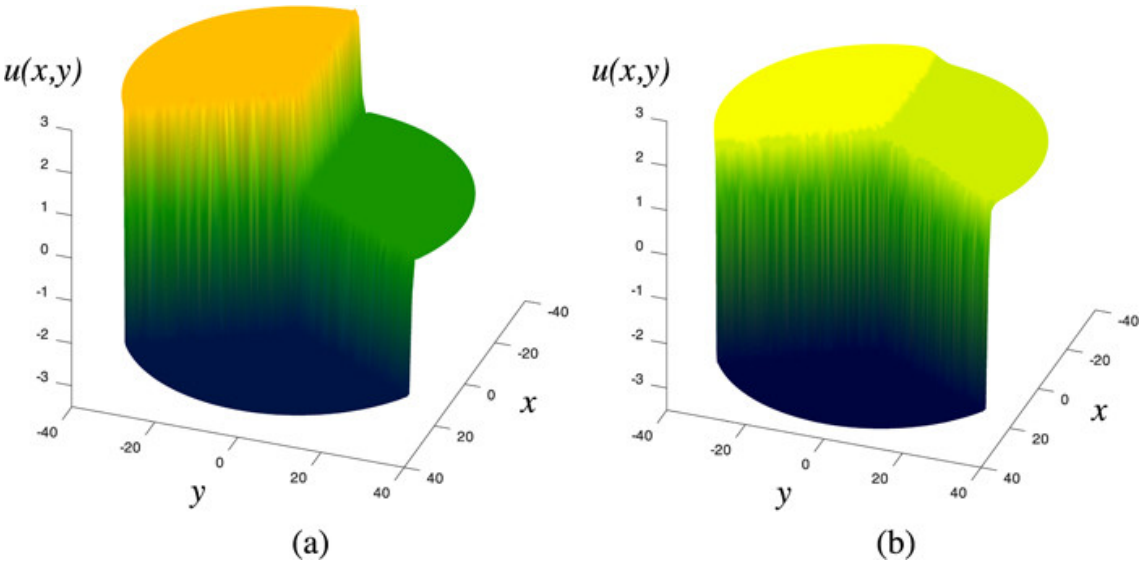


Figure 3.11: Long-time solutions $u(x,y,t)$ of (3.1) with no-flux boundary condition and with an initial condition $u(x,y,0)$ that was equally divided between u_0 (dark blue), u_1 (green),and u_2 (yellow). The two choices of parameters correspond to equipotential points so the speeds of the three fronts connecting these three states are zero; therefore, all three homogeneous solutions coexist peacefully separated by three zero-velocity interfaces. (a) $\alpha = -10$, $\beta = 2$, $\gamma = 18.75$, and $\delta = -5$; (b) $\alpha = -10$, $\beta = 7.8$, $\gamma = 18.75$, and $\delta = -19.5$. As the parameters β and δ are different in the two subfigures, the values of u_0, u_1, and u_2 are also different.

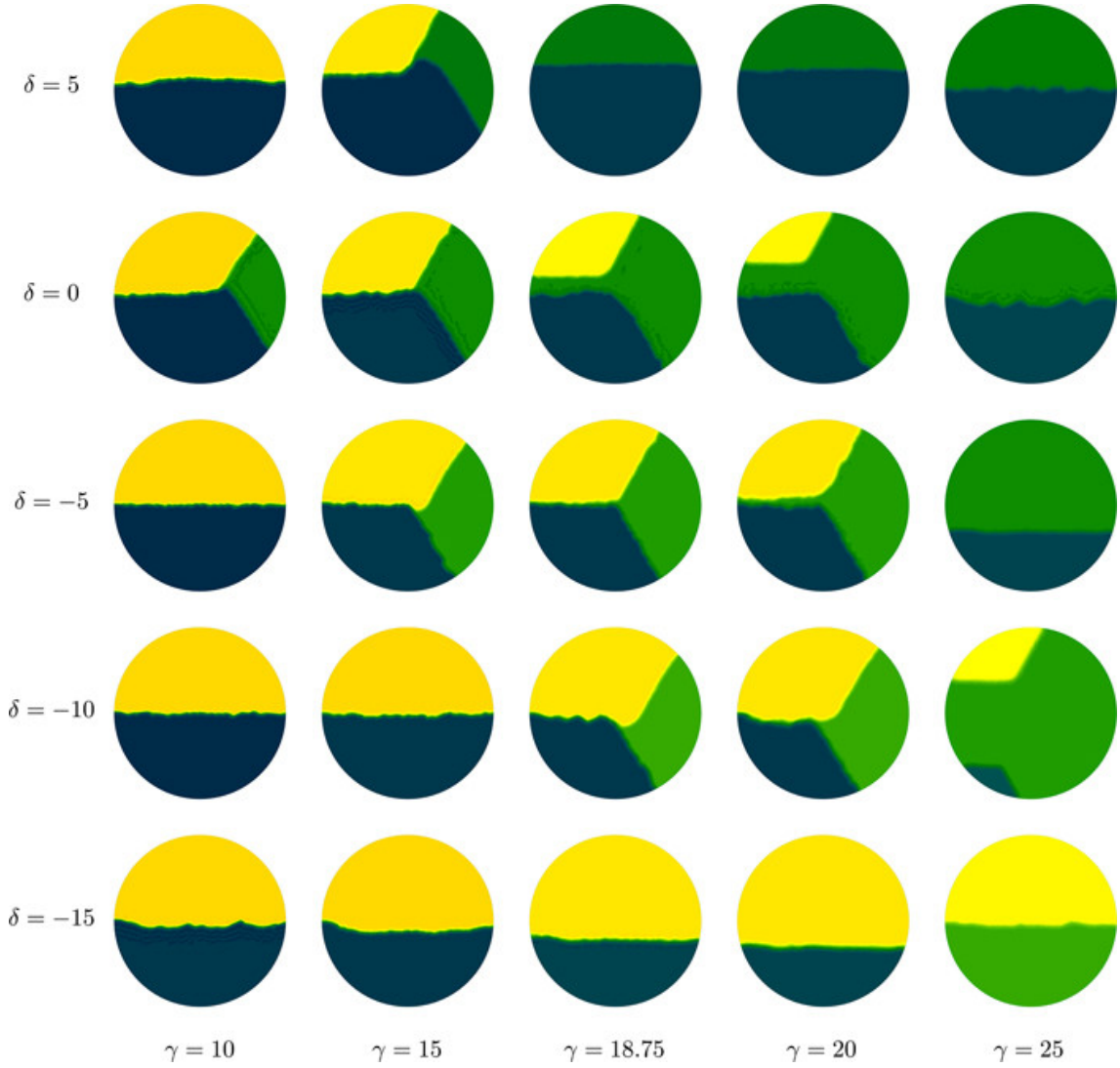


Figure 3.12: Integrations of (3.1) in a two-dimensional domain for different combinations of parameters (γ, δ), evaluated at a time $t = 3$. The plots are organized as to be comparable with the diagram in Fig.3.4 ($\alpha = -10, \beta = 2$). The corner graphs show bistable situations: $V(u_0) > V(u_2)$, no state u_1 (upper-left); $V(u_1) > V(u_0)$, no state u_2 (upper-right); $V(u_2) > V(u_0)$, no state u_1 (bottom-left); and $V(u_1) > V(u_2)$, no state u_0 (bottom-right). At the center of the figure is the equipotential scenario $(\gamma_{\rm ep}, \delta_{\rm ep})$ at which all three front velocities become zero. Around that case, several combinations of parameters achieve different invasion patterns between the three stable states.

In Secs. 3.1-3.6, we explored the front connections between stable states by basically looking at the potential condition.

One could now verify some of the findings by integrating the original reaction–diffusion model (3.1) in an arbitrarily long 1D domain for several combinations of the parameters, as is often done in this context. Here, we wanted to verify our predictions and show that they are relevant also in finite two-dimensional domains. We used the same sort of numerical setup we used in Ref. 15: integrations of evolution (3.1) in a 2D domain with a "free" boundary where $\partial u/\partial n = 0$, and an initial condition $u(x, y; t = 0)$ that is equally divided between the two or three stable solutions,

depending on the parameter values. In this way, we allow the two or three states to interact and develop interfaces that move, determining the invasion sequence and the final dominance. As in Ref. 15, we used the software FreeFem++ [33] that implements the finite element method with an irregular triangular mesh.

A first verification is the existence of equipotential points. Using $\alpha = -10$ and $\beta = 2$, where $(\gamma_{\rm ep}, \delta_{\rm ep}) = (18.75, -5)$, we computed the long-time behavior of the solution $u(x,y,t)$, shown in Fig. 3.11(a). The basic "triple junction" was preserved over time as the interfaces did not experience invasion motion. Interestingly, the front between u_0 and u_2 exhibits some wave motion along the interface, a phenomenon that may look similar to the traveling interface pulses reported in Ref. 34. Even if stationary one-dimensional fronts 01 and 02 have zero speed, in two dimensions, there are transversal effects that decay very slowly. This kind of instability due to sharp interfaces could play a role in a physical context, for instance, in the presence of crystal anisotropies, terraces or defects that are common in surface reactions.

Using $\alpha = -10$ and $\beta = 7.8$, where $(\gamma_{\rm ep}, \delta_{\rm ep}) = (18.75, -19.5)$, the "triple junction" initial condition evolves into Fig. 3.11(b). Comparing with the previous situation, the difference $u_2 - u_1$ becomes smaller and the front 12 becomes smoother. The interface between u_0 (dark blue) and u_2 (light green) suffers a subtle change with respect to Fig. 3.11(a): as one crosses the interface, the front $u(x,y,t)$ develops an intermediate region of moderate slope where $u(x,y,t) \approx u_1$ [see also Fig. 3.8(b)]. This feature corresponds to the "twostepped" fronts that, as proved by Rubinstein *et al.* [19], persist in the neighborhood of equipotential points.

In Fig. 3.12, solutions $u(x,y,t)$ for intermediate values of t are shown, for the selection $\alpha = -10$ and $\beta = 2$ and a discrete set of values of γ and δ. These integrations illustrate different possibilities treated in Secs. 3.2-3.6 and could be contrasted with Figs. 3.7 and 3.9.

Several combinations of (γ, δ) lie outside the tristability region and their initial conditions are divided equally between two states by a single interface that may have moved after some time. In the lower-left part of the diagram, only the states u_0 (dark blue) and u_2 (yellow) are stable, with state u_2 being slightly dominant $c_{20} > 0$; in the upper-right part, only u_0 and u_1 (green) are stable, with u_0 being slightly dominant $c_{01} > 0$, with a single instance of $c_{10} > 0$ for $(\gamma, \delta) = (25, -5)$. At the lower-right corner of the diagram, $(\gamma, \delta) = (25, -15)$, there is a case of $c_{12} \approx 0$.

Inside the tristability region, the outcomes are organized around the equipotential case $(\gamma_{\rm ep}, \delta_{\rm ep}) = (18.75, -5)$ that was analyzed in Fig. 3.11(a). Directly to its left and downward in the diagram, $c_{21} > 0$ and the state u_2 seems to grow. As one moves from the equipotential case to the right or upward, the interface 02 suffers a front splitting instability, and two parallel interfaces 01 and 12 develop and the surface covered with state u_1 begins to grow (at least transiently). This phenomenon is particularly clear at $(\gamma, \delta) = (25, -10)$, where both areas u_0 and u_2 were displaced by the area covered by state u_1.

The results of the simulations can also be interpreted as manifestations of the changes suffered by the potential $V(u)$: with increasing γ, $V(u_1)$ decreases relatively to $V(u_0)$ and $V(u_2)$, thus u_1 being more dominant on the right section of the graph. Similarly, an increasing δ favors the dominance of state u_0.

Figure 3.13 shows a selection of solutions $u(x,t)$ at $t = 5$, using ten selections of parameter values along the line $\delta = 0$. For all values ofδ shown in the figure, $V(u_0) < V(u_1)$ and $V(u_0) <$

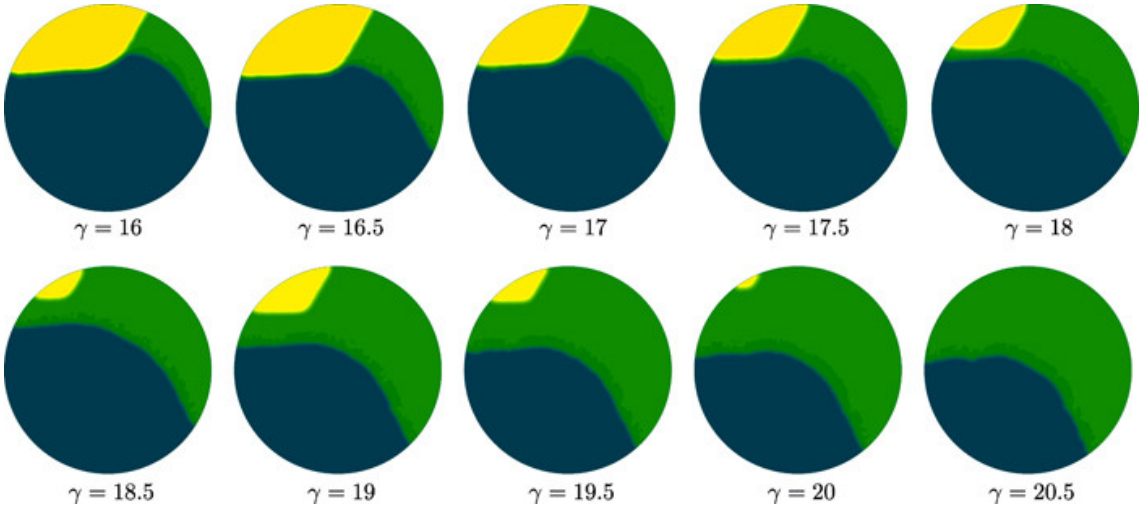


Figure 3.13: Solutions $u(x,y,t)$ for several combinations of parameter values: $\alpha = -10$, $\beta = 2$, $\gamma = 0$, and $16 \leq \delta \leq 20.5$. All integrations were initialized using the same initial condition $u(x,y;0)$ (a disk divided equally between three sectors) and are here shown at $t = 5$. Although the u_0 state invaded the other two in all these cases, it is clear that for $\gamma \geq 18$, the interface between u_0 and u_2 breaks up.

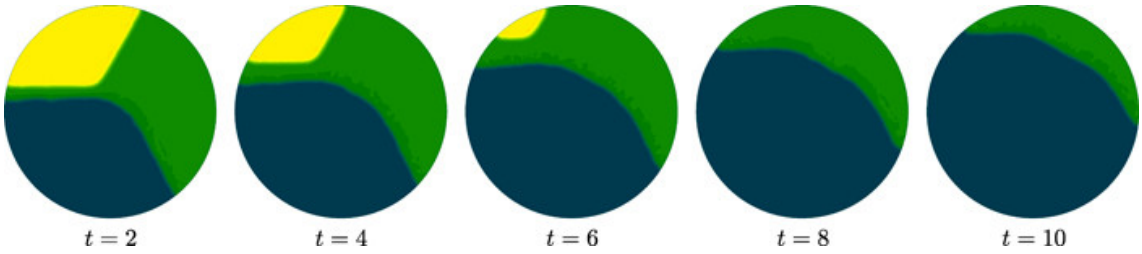


Figure 3.14: Solution $u(x,y,t)$ for parameters $\alpha = -10, \beta = 2, \gamma = 18.5$, and $\delta = 0$, illustrating the front splitting instability: the interface between u_0 (blue) and u_2 (yellow) becomes thicker and a long strip with the state u_1 (green) appears, creating two parallel interfaces, one connecting u_0 and u_1, and other connecting u_1 and u_2. Transiently the area covered by u_1 grows, but in the end, u_0 invades the whole domain.

$V(u_2)$, therefore, the state u_0 dominates in the long term (not shown). For $\gamma =\leq 17.5$, no front splitting can be observed: the front 02 exists and has nearly zero speed. However, for $\gamma \geq 18$, the front 02 gets destabilized, becomes thicker until it finally splits into two separate parallel fronts 01 and 12. For larger values of γ, the state u_1 pushes u_2 out of the boundaries of the disk, and in turn u_0 pushes u_1, becoming in the long run the only dominant state.

A better understanding of the process of front splitting can be gained by looking at the progression for several instants. Figure 3.14 shows the evolution of an initial condition that was equally divided between the three stable states, for parameter values selected such that the front 02 does not exist. As the interface between states u_0 and u_2 becomes wider, it is clear that state u_1 is able to "sneak in" between the other two, creating two separate parallel interfaces 01 and 12. Although in the end state the u_0 invades the whole domain, the area covered by metastable state u_1 grows transiently.

Figure 3.15 illustrates the front splitting behavior furthermore. We define $n_1(t)$ as the fraction

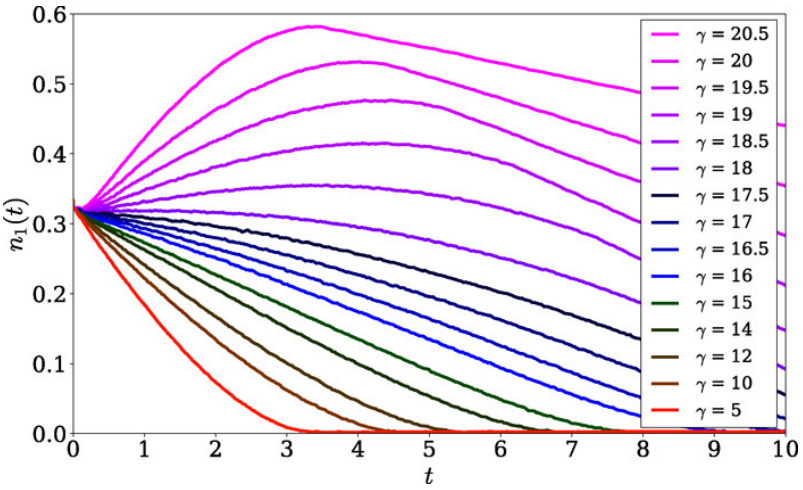


Figure 3.15: The quantity $n_1(t)$ defined by (3.19) measures the fraction of surface covered by metastable state u_1. For different combinations of parameters, the curve $n_1(t)$ shows an overall decay to zero as u_0 is the dominant state, but for some range of values $\gamma \geq 18$, it shows a transient growth, explained by the front splitting instability.

of surface covered by state u_1,

$$n_1(t) = \frac{\int_S \chi_{[u_1-\varepsilon,u_1+\varepsilon]}(u(x,y;t))\mathrm{d}x\mathrm{d}y}{\int_S \mathrm{d}x\mathrm{d}y} \tag{3.19}$$

with

$$\chi_{[a,b]}(u) \begin{cases} 1 & \text{if } u \in [a,b] \\ 0 & \text{otherwise} \end{cases}$$

and low values of ε.

For $\delta = 0$ and low values of γ, the curve decays to zero monotonically. But for $\gamma \geq 18$, $n_1(t)$ shows an initial growth phase that is followed by a decay to zero, illustrating that the transient effect of the front splitting instability can also be detected by a global measure.

3.8 Conclusion

In the present article, we described a cartoonishly simplified one-dimensional model for reaction–diffusion processes with up to three stable states. From Catastrophe Theory, we know that a sixthorder polynomial potential can describe the nonlinear behavior of such systems. The unstable and stable states can then be calculated as the roots of a quintic polynomial.

We extensively studied the bifurcation behavior of the model in parameter space. The existences of front solutions can be deduced using a mechanical analogy. Based on that, we observed equipotential points associated with zero-velocity fronts. These equipotential points and curves can be thoroughly analyzed with continuation methods, identifying regions in parameter space where fronts exhibit similar behaviors. With this simple model, we conclude that the final behavior and front velocities depend on the depths of the potential wells, not on eigenvalues or well sizes. All our observations are supported by simulations in two-dimensional space.

Another interesting phenomenon that can be captured by the sixth-order polynomial potential is the front splitting instability, as described in different contexts [17, 18]. We analyze the conditions of front splitting in our model. With simple assumptions, we can give a basic estimate, which is supported by continuation results and simulations on a two-dimensional, finite disk.

Future work should consider two-species systems using standard forms of polynomials of two variables. A goal would be to find potential conditions that can predict the onset of nonequilibrium Ising-Bloch transition and other bifurcations of fronts [21, 22, 35].

Acknowledgments

K.R., J.C., and S.W. are grateful for the financial support of the Erasmus+ mobility programme of the European Union. The authors are grateful for the general support of Dr. Christian Fischer.

Data Availability

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Bibliography

[1] Murray JD *Mathematical Biology*. (1993) Berlin, Heidelberg: Springer Berlin Heidelberg.

[2] Méndez V, Fedotov S, Horsthemke W *Reaction–Transport Systems*. (2010) Springer Series in Synergetics. Berlin, Heidelberg: Springer Berlin Heidelberg.

[3] Erneux T, Kozyreff G, Tlidi M *Bifurcation to Fronts Due to Delay*. (2010) *Phil. Trans. R. Soc. A.* **368**, 483–493. doi: 10.1098/rsta.2009.0228.

[4] Otto A, Wang J, Radons G *Delay-Induced Wave Instabilities in Single-Species Reaction-Diffusion Systems*. (2017) *Phys. Rev. E* **96**, 052202. doi: 10.1103/physreve.96.052202.

[5] Rombouts J, Gelens L, Erneux T *Travelling Fronts in Time-Delayed Reaction–Diffusion Systems*. (2019) *Phil. Trans. R. Soc. A.* **377**, 20180127. doi: 10.1098/rsta.2018.0127.

[6] Benguria RD, Depassier MC *Exact Fronts for the Nonlinear Diffusion Equation with Quintic Nonlinearities*. (1994) *Phys. Rev. E* **50**, 3701–3704. doi: 10.1103/PhysRevE.50.3701.

[7] Benguria RD, Depassier MC *Speed of Fronts of the Reaction-Diffusion Equation*. (1996) *Phys. Rev. Lett.* **77**, 1171–1173. doi: 10.1103/physrevlett.77.1171.

[8] Bär M, Zülicke C, Eiswirth RM, Ertl G, Eiswirth RM, Ertl G *Theoretical Modeling of Spatiotemporal Self-Organization in a Surface Catalyzed Reaction Exhibiting Bistable Kinetics*. (1992) *J. Chem. Phys.* **96**, 8595–8604. doi: 10.1063/1.462312.

[9] Cisternas J, Depassier MC *Counterexample to a Conjecture of Goriely for the Speed of Fronts of the Reaction-Diffusion Equation*. (1997) *Phys. Rev. E* **55**, 3701–3704. doi: 10.1103/physreve.55.3701.

[10] Wehner S, Hoffmann P, Schmeißer D, Brand HR, Küppers J *Spatiotemporal Patterns of External Noise-Induced Transitions in a Bistable Reaction-Diffusion System: Photoelectron Emission Microscopy Experiments and Modeling*. (2005) *Phys. Rev. Lett.* **95**, 1–4. doi: 10.1103/PhysRevLett.95.038301.

[11] Karpitschka S, Wehner S, Küppers J *Reaction Hysteresis of the CO+O→CO_2 Reaction on Palladium(111)*. (2009) *J. Chem. Phys.* **130**, 1–11. doi: 10.1063/1.3072712.

[12] Cisternas J, Karpitschka S, Wehner S *Travelling Fronts of the CO Oxidation on Pd(111) with Coverage-Dependent Diffusion*. (2014) *J. Chem. Phys.* **141**, 164106–164106. doi: 10.1063/1.4898705.

[13] Cui T, Tang S, Zhang L, Yu D *Swallowtail Model for Predicting the Global Bifurcation Behavior of CO Oxidation Reactions*. (2011) *Sci. China Chem.* **54**, 1072–1077. doi: 10.1007/s11426-011-4294-y.

[14] Stegemerten F, Gurevich S, Thiele U *Bifurcations of Front Motion in Passive and Active Allen-Cahn-type Equations*. (2020) *Chaos* **30**, 053136. doi: 10.1063/5.0003271.

[15] Rohe K, Cisternas J, Wehner S *Competing Ternary Surface Reaction CO+O_2+H_2 on Ir(111)*. (2020) *Proc. R. Soc. A.* **476**, 20190712. doi: 10.1098/rspa.2019.0712.

[16] Fife PC, McLeod JB *The Approach of Solutions of Nonlinear Diffusion Equations to Travelling Front Solutions*. (1977) *Arch. Ration. Mech. Anal.* **65**, 335–361. doi: 10.1007/BF00250432.

[17] Bechhoefer J, Löwen H, Tuckerman LS *Dynamical Mechanism for the Formation of Metastable Phases*. (1991) *Phys. Rev. Lett.* **67**, 1266–1269. doi: 10.1103/PhysRevLett.67.1266.

[18] Tuckerman LS, Bechhoefer J *Dynamical Mechanism for the Formation of Metastable Phases: The Case of Two Nonconserved Order Parameters*. (1992) *Phys. Rev. A* **46**, 3178–3192. doi: 10.1103/PhysRevA.46.3178.

[19] Rubinstein J, Sternberg P, Keller JB *Front Interaction and Nonhomogeneous Equilibria for Tristable Reaction-Diffusion Equations*. (1993) *SIAM J. Appl. Math.* **53**, 1669–1685. doi: 10.1137/0153077.

[20] Rinzel J, Terman D *Propagation Phenomena in a Bistable Reaction-Diffusion System*. (1982) *SIAM J. Appl. Math.* **42**, 1111–1137. doi: 10.1137/0142077.

[21] Hagberg A, Meron E *Pattern Formation in Non-Gradient Reaction-Diffusion Systems: The Effects of Front Bifurcations*. (1994) *Nonlinearity* **7**, 805–835. doi: 10.1088/0951-7715/7/3/006.

[22] Chirilus-Bruckner M, Doelman A, van Heijster P, Rademacher JD *Butterfly Catastrophe for Fronts in a Three-Component Reaction–Diffusion System*. (2014) *J. Nonlinear Sci.* **25**, 87–129. doi: 10.1007/s00332-014-9222-9.

[23] Poston T, Stewart I *The First Seven Catastrophe Geometries*. (1996) Mineola, New York: Dover Publications.

[24] Zemskov EP *Front Bifurcation in a Tristable Reaction-Diffusion System under Periodic Forcing*. (2004) *Phys. Rev. E* **69**, 036208. doi: 10.1103/physreve.69.036208.

[25] Wetzel D *Tristability between Stripes, up-Hexagons, and down-Hexagons and Snaking Bifurcation Branches of Spatial Connections between up- and down-Hexagons*. (2018) *Phys. Rev. E* **97**, 1–7. doi: 10.1103/PhysRevE.97.062221.

[26] Zelnik YR, Gandhi P, Knobloch E, Meron E *Implications of Tristability in Pattern-Forming Ecosystems*. (2018) *Chaos* **28**, 033609–033609. doi: 10.1063/1.5018925.

[27] Aronson D, Weinberger H *Multidimensional Nonlinear Diffusion Arising in Population Genetics*. (1978) *Adv. Math.* **30**, 33–76. doi: 10.1016/0001-8708(78)90130-5.

[28] Shvartsman SY, Shütz E, Imbihl R, Kevrekidis IG *Dynamics on Microcomposite Catalytic Surfaces: The Effect of Active Boundaries*. (1999) *Phys. Rev. Lett.* **83**, 2857–2860. doi: 10.1103/PhysRevLett.83.2857.

[29] van Saarloos W *Front Propagation into Unstable States. II. Linear versus Nonlinear Marginal Stability and Rate of Convergence*. (1989) *Phys. Rev. A* **39**, 6367–6390. doi: 10.1103/physreva.39.6367.

[30] Epstein IR, Pojman JA *An Introduction to Nonlinear Chemical Dynamics: Oscillations, Waves, Patterns, and Chaos*. (1998) Oxford University Press.

[31] Doedel E *AUTO-07P: Continuation and Bifurcation Software for Ordinary Differential Equations*. (2007) Technical report Concordia University Montreal, Canada.

[32] Doedel EJ, Friedman MJ *Numerical Computation of Heteroclinic Orbits*. (1989) *J. Comput. Appl. Math.* **26**, 155–170. doi: 10.1016/0377-0427(89)90153-2.

[33] Hecht F *New Development in Freefem++*. (2012) *J. Numer. Math* **20**, 251–265. doi: 10.1515/jnum-2012-0013.

[34] Makeev A, Imbihl R *Simulation of Traveling Interface Pulses in Bistable Surface Reactions*. (2019) *Phys. Rev. E* **100**, 1–9. doi: 10.1103/PhysRevE.100.042206.

[35] Coullet P, Lega J, Houchmanzadeh B, Lajzerowicz J *Breaking Chirality in Nonequilibrium Systems*. (1990) *Phys. Rev. Lett.* **65**, 1352–1355. doi: 10.1103/physrevlett.65.1352.

4

Propagation failure in discrete reaction-diffusion system based on the butterfly bifurcation

Kevin Rohe, Jaime Cisternas

2022 *Chaos (submitted)*

Reaction-diffusion systems are used in biology, chemistry, and physics to model the interaction of spatially distributed species. Particularly of interest is the spatial replacement of one equilibrium state by another, depicted as traveling waves or fronts. Their profiles and traveling velocity depend on the nonlinearities in the reaction term and on spatial diffusion. If the reaction occurs at regularly spaced points, the velocities also depend on lattice structure and the orientation of the traveling front. Interestingly, there is a wide region of parameters where the speeds become zero and the fronts do not propagate.

In this manuscript, we focus on systems with three stable coexisting equilibrium states that are described by the butterfly bifurcation, and study to what extent the three possible 1D traveling fronts suffer from propagation failure. We demonstrate that discreteness of space affects the three fronts differently. Regions of propagation failure add a new layer of complexity to the butterfly diagram. The analysis is extended to planar fronts traveling through different orientations in regular 2D lattices. Both propagation failure and the existence of preferred orientations play a role in the transient and long-time evolution of 2D patterns.

One of the key features of reaction-diffusion models is the additive combination of reactions between the chemical or biological species and their diffusive spreading. The behavior of traveling waves and stationary patterns depends on the interaction of reaction and diffusion. This is also true for discrete space, where fronts (patterns formed by two uniform states) become 'pinned' to the lattice and do not move unless diffusion is strong enough and a certain nontrivial threshold is overcome. This phenomenon has been widely studied and has found application in concrete experiments. For tristable systems (with three coexisting stable uniform states), the situation is far from simple since there are up to three types of fronts (interfaces between pairs of different states) that behave differently in discrete space. For the sake of concreteness, we will use a single species model with a quintic polynomial as a reaction term. The qualitative behavior of such a system is explained by the butterfly bifurcation diagram that contains all the relevant changes as the coefficients of the polynomial are varied. The overall picture is enriched when several 1D and 2D lattices (and orientations of the fronts) are considered. As we will show, discreteness modifies the basic butterfly diagram in a highly nontrivial way, since the Maxwell (or equipotential) condition is no longer a requirement for non-propagating fronts in discrete space. In 2D, discreteness effects are particularly subtle since they depend also on the alignment of the front with the discrete lattice. As we will show with illustrative examples, the evolution of heterogeneous 2D patterns follows the basic mechanisms predicted by bifurcation analysis.

4.1 Introduction

The oxidation of CO on platinum group metal catalysts is a basic chemical system that has been extensively investigated concerning its spatio-temporal patterns and bifurcation properties. The prevalent reaction mechanism has been shown experimentally by the work of Ertl [1, 2] to follow the Langmuir-Hinshelwood (LH) mechanism, i.e. adsorption, reaction on the surface and consecutive desorption of carbon dioxide [2]. This mechanism is also the foundation to explain more complex behavior such as oscillations [3, 4], traveling waves [5] and spiral waves [6]. The LH model for CO oxidation assumes the form of a reaction-diffusion (RD) system of equations for the adsorbed species.

While this model approaches a specific system using physical and chemical constants measured with high precision, another approach is to utilize a generic reaction-diffusion equation to qualitatively mimic the behavior of the system of interest. This leads to simpler models which still can show complex behavior such as bifurcations and chaos, and allows direct insight into the interaction between multiple stable solutions (see for instance Refs. 7–10). Spatial discreteness is a property of physical systems, which could explain some discrepancies between observations and modeling. For certain cases, this discreteness is a defining feature, e.g. in cellular systems and neuronal networks [11] or cascade reactors [12] where coupled units influence each other.

In heterogeneous catalysis, the surface facet of a crystalline catalyst defines the properties of reaction and diffusion. Different molecules show preferred adsorption sites [13] on the discrete atomic lattice. A reaction front may concur with a surface reconstruction [14] and surface defects such as steps can lead to anisotropic patterns [15, 16]. Quite often discreteness leads to *propagation failure* (PF), where a given front solution does not propagate along the nodes as oth-

erwise expected if diffusivities are too small. Peyrard and Kruskal [17], and Keener [18] proved that this is an inherent property of certain 1D discrete reaction-diffusion models. Propagation failure was investigated in the context of coupled chemical reactors [19] and bistable chemical reactions [20, 21]. It has been proved that propagation failure is connected to the existence of stable non-propagating fronts, which define 'snaking branches' of solutions as some reaction parameter is varied [8, 22–24]. The width of the snaking region in parameter space gets smaller as more dense lattices or larger diffusivities are considered.

In this article, we want to study propagation failure in systems with up to three stable uniform states. In this regard, we build on the insights of Humphries and Wilds [25] but extend the basic model developed in Ref. 9 which exhibits the butterfly bifurcation using a sixth-order polynomial as potential. The name 'butterfly' comes from the general shapes of the regions with one, two or three stable equilibria in a codimension-2 diagram: the tristability region being the body of the butterfly, and the two narrow bistability regions being the wings of the butterfly.

The reason for using the butterfly bifurcation is simplicity. Tristable and multistable scenarios are common in applications involving multiple species and complex nonlinearities (see for instance Refs. 26–28). These systems can be modeled by a variety of sophisticated models with rich bifurcation structures. Now since we want to analyze propagation failure in the simplest tristable RD model, we propose a 'caricature' single-species model with a single polynomial term with up to 5 different roots, 3 of them stable. From Catastrophe Theory (as explained in Ref. 29 and used in our previous article [9]) such polynomial must be quintic and can be conveniently written with only four free coefficients.

We will use the following structure: in section 4.2, the model is introduced and our previous results are shortly recapitulated. Section 4.3 follows with a discussion of extending the model in discrete space, described by regular lattices. The existence of propagation failure as a result of discrete diffusion lattices is discussed in section 4.4. Numerical simulations on a finite 2D domain exploring the influence of different variables, i.e. reaction parameter, discreteness, diffusion grid, angular front orientation conclude our analysis in section 4.5 with numerical simulations on 2D rectangular diffusion grids, varying different parameters of the reaction term and changing the connectivity of the lattice.

4.2 Model

Despite all the extensive work on propagation failure of RD fronts, little has been done for systems with three stable states that can show propagation failure of some of the three fronts. In our previous article, we constructed a single species reaction-diffusion system based on a quintic reaction term that enabled the study of all the bifurcations [9]. Since an extensive description of the model was given there, only a fundamental introduction is provided in the present section.

The most basic model to describe a reaction-diffusion system in an infinite 1D domain is given by

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + f(u), \tag{4.1}$$

where $f(u)$ is a nonlinear function representing the local reaction rate and is the decisive element to the existence and stability of homogeneous states. The diffusion constant D provides the spatial scale but its precise value is not relevant for the qualitative behavior of the solutions.

Focusing on the front solutions of the form $u(x,t) = u(z)$ with $z = x - vt$ and boundary conditions at infinity:

$$u(z) \xrightarrow[z\to-\infty]{} \tilde{u}_- \quad \text{and} \quad u(z) \xrightarrow[z\to+\infty]{} \tilde{u}_+$$

where $\tilde{u}_+$ and $\tilde{u}_-$ are stable roots of the reaction:

$$f(\tilde{u}_+) = f(\tilde{u}_-) = 0, f'(\tilde{u}_+) < 0, f'(\tilde{u}_-) < 0$$

Here the concept of stability of a root $\tilde{u}$ is associated to the behavior of trajectories of the ordinary differential equation $\dot{u} = f(u)$ close to the root $\tilde{u}$. If the velocity $v > 0$ then the front moves to the right and the region covered with $\tilde{u}_-$ grows, effectively 'invading' the state $\tilde{u}_+$.

Since we want to focus on competition between stable homogeneous solutions, we do not consider fronts invading unstable states. In some scenarios, the speeds of these other fronts depend on the reaction term at the state being invaded: it is said that the front is being 'pulled', as opposed to 'pushed' fronts with speeds that depend on the whole reaction term [7]. Using this terminology, all fronts between stable states (as considered in this article) are 'pushed'.

To obtain a model with multiple stable states, we propose using the quintic polynomial

$$f_{\alpha\beta\gamma\delta}(u) = -u^5 - \alpha u^3 - \beta u^2 - \gamma u - \delta, \tag{4.2}$$

which, depending on the parameters α, β, γ, and δ. $f_{\alpha\beta\gamma\delta}(u)$ may have one, two or three stable roots being designated $\tilde{u}_0, \tilde{u}_1$ and $\tilde{u}_2$. These stable roots of $f(u) = 0$ correspond to minima of the sixth-order potential:

$$V_{\alpha\beta\gamma\delta}(u) = \frac{1}{6}u^6 + \frac{\alpha}{4}u^4 + \frac{\beta}{3}u^3 + \frac{\gamma}{2}u^2 + \delta u, \tag{4.3}$$

(satisfying $f_{\alpha\beta\gamma\delta}(u) = -V'_{\alpha\beta\gamma\delta}(u)$). In the following we will drop the dependence on parameters.

$V(u)$ always has at least one minimum and depicts the most general unfolding of a degenerate potential of one variable that has five derivatives equal to zero at the same point. By varying its four coefficients, the minima of the potential undergo all the possible transitions between qualitatively distinct scenarios.

Using a mechanical analogy, the relative 'depths' of the wells of the potential are a measure for the stability of the corresponding solutions, and the 'hill' between two wells can be seen as an energy barrier to be overcome by diffusion. It can be shown that the front velocity is proportional to the difference $V(\tilde{u}_+) - V(\tilde{u}_-)$ and the sign indicates the direction of propagation.

Using this model, we investigated conditions and properties of equipotential points and curves in parameter space. At these locations, the front velocities between multiple coexisting solutions become zero. Fig. 4.1 illustrates a representative (γ, δ)-section (for $\alpha = -10$ and $\beta = 2$). Indicated are regions with 1, 2 or 3 stable roots, the equipotential curves, and their intersection, the equipotential point. The relative values of the potential minima in different sub-regions lead to qualitative

predictions for the front solutions. These findings were verified by numerical simulations on a 1D domain and also on a 2D-disk using a triangulated mesh.

Since the treatment of Eq. (4.1) relies on integration along the continuous space, application of a discrete spatial lattice requires a separate analysis. Necessary modifications to the model are discussed in the following section.

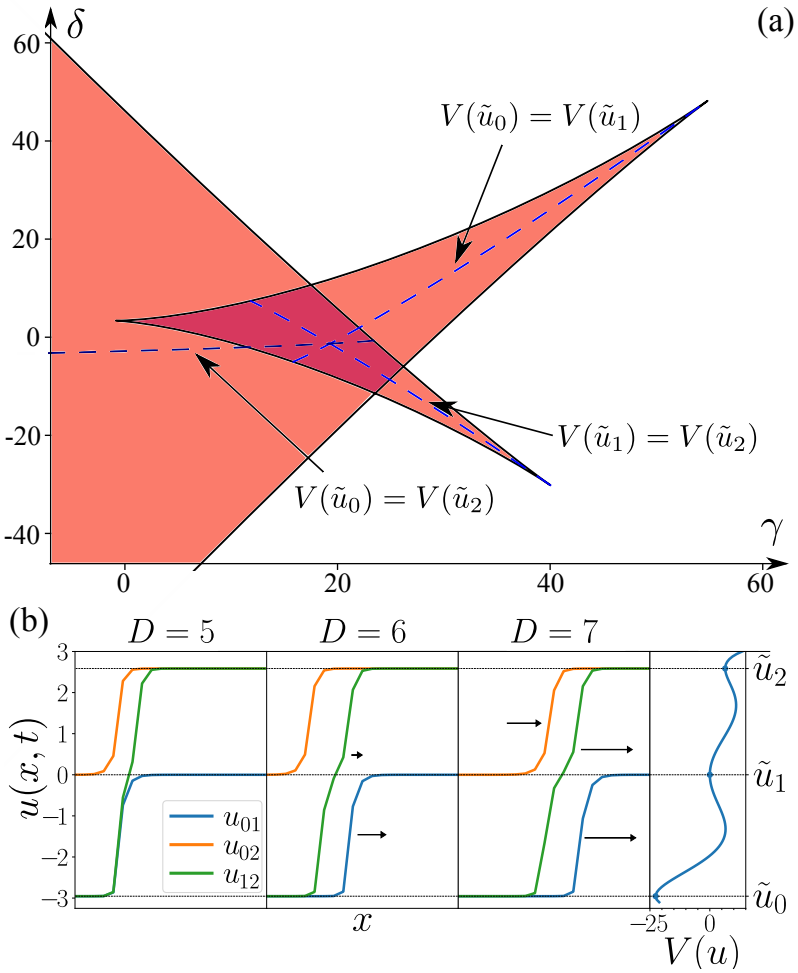


Figure 4.1: (a) Section of the butterfly bifurcation diagram corresponding to the potential function $V(u)$ defined in Eq. (4.3) and the parameters $\alpha = -10$ and $\beta = 2$ (which will further be used as standard). Color indicates the number of minima: 1 (white), 2 (light), and 3 (dark). Dashed lines correspond to equipotential curves where two minima have the same values of the potential function; the intersection at $\gamma = 18.75, \delta = -5$ is the equipotential point of all three minima. (b) Example of PF in different front solutions of Eq. (4.4) for $\gamma = 17, \delta = 0, \Delta x = 1$ and $D \in \{5, 6, 7\}$ over a 1D-strip. For $D = 5$ propagation failure occurs, while for $D = 7$ all three fronts propagate freely. Moving fronts are indicated by arrows with lengths proportional to the observed velocities. The right-most diagram shows the potential $V(u)$ as in Eq. (4.3).

4.3 Discrete lattices

The equivalent of the RD Eq. (4.1) in a (infinite) 1D discrete domain is:

$$\begin{aligned}\frac{\mathrm{d}u_i}{\mathrm{d}t} &= \frac{D}{\Delta x^2}(u_{i-1}+u_{i+1}-2u_i)+f(u_i) \\ &= \frac{D}{\Delta x^2}g(i)+f(u_i)\end{aligned} \tag{4.4}$$

where $u_i = u(x_i)$. The function $g(i)$ will be denoted as interaction term for u_i in the following. The quotient $D/\Delta x^2$ controls the effect of discreteness: for small Δx or large D, the solution of the discrete equation will resemble the solution of the continuous Eq. (4.1).

RD systems may lead to numerical instabilities but we found the quintic term to be rather well-behaved: only (static and traveling) smooth fronts were observed. The checkerboard instability was prevented by using an implicit algorithm. The only relevant instability is the propagation failure that is the main topic of the article.

The discrete RD equation can be written also as a gradient system:

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -\frac{\partial F}{\partial u_i} \tag{4.5}$$

with the Lyapunov functional:

$$F = \sum_k \left[V(u_k) - \frac{D}{4\Delta x^2}\left((u_{k-1}-u_k)^2+(u_{k+1}-u_k)^2\right)\right] \tag{4.6}$$

Its property that $\mathrm{d}F/\mathrm{d}t = -\sum_i(\mathrm{d}u_i/\mathrm{d}t)^2 \leq 0$ means that dynamic evolution is oriented towards states that are minima of F (if such exist). The existence of a Lyapunov functional F prevents oscillating and chaotic fronts [30].

To further study front propagation and the effect of front orientation, we propose the use of three regular 2D lattices illustrated in Fig. 4.2 (square, hexagonal and triangular), which are discussed in the following sections. While there are infinitely many other lattices and orientations that one could consider (see also Refs. 25, 31, 32 for other treatments), these three were chosen due to their simplicity, but they also follow symmetries found on surface catalysts and can be used to model other systems.

4.3.1 Square lattice

The simplest regular 2D lattice is a square array where each node is connected to four other nodes located at distance Δx and an angle of 90°. Due to the 4-fold symmetry, the links between nodes form squares as well. Regarding physical systems, this could be applied to model diffusion on an fcc(100) metal surface.

The discrete equation in a 2D square domain uses a 5-point stencil:

$$\frac{\mathrm{d}u_{i,j}}{\mathrm{d}t} = \frac{D}{\Delta x^2}(u_{i-1,j}+u_{i+1,j}+u_{i,j-1}+u_{i,j+1}-4u_{i,j})+f(u_{i,j}) \tag{4.7}$$

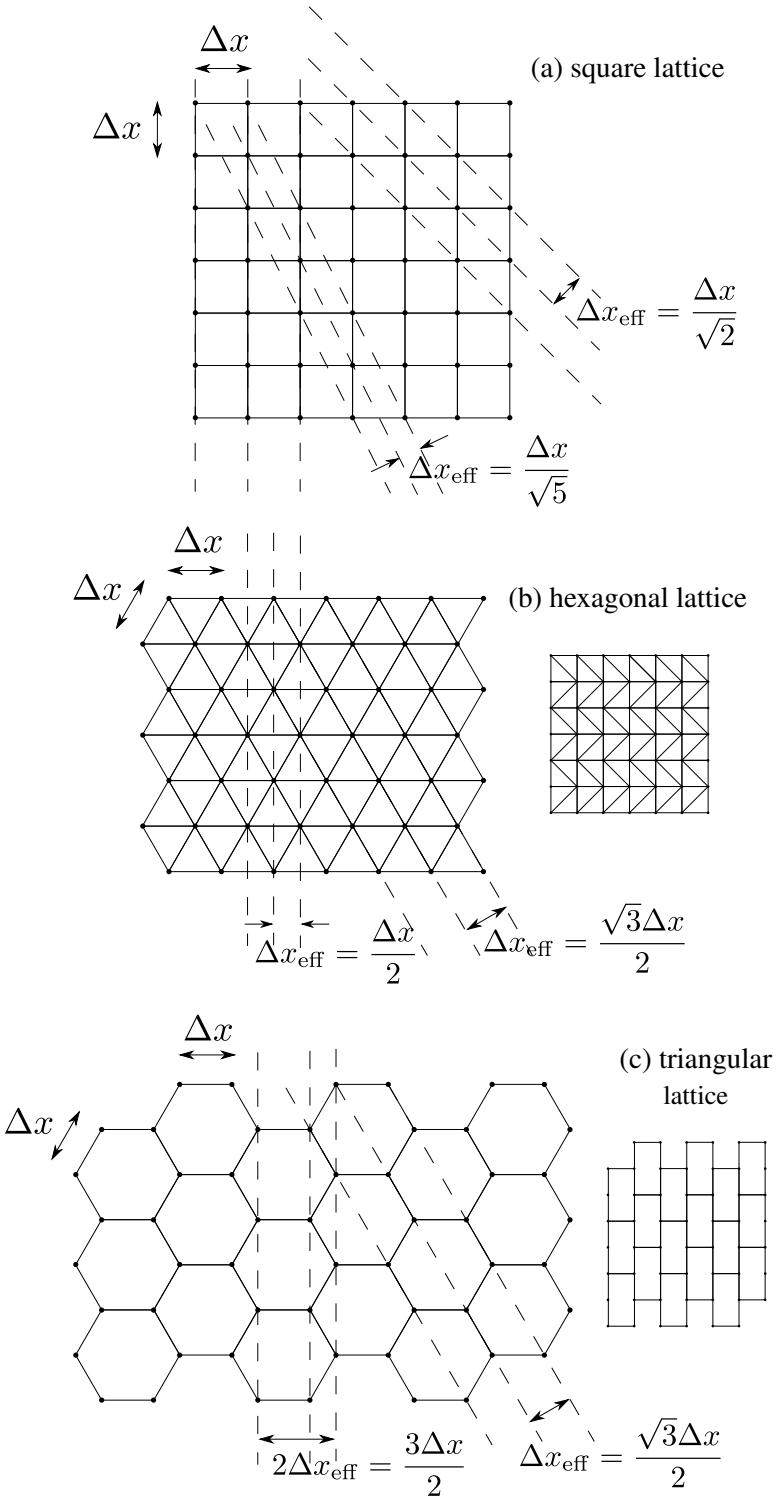


Figure 4.2: Illustration of the chosen diffusion lattices : (a) square, (b) hexagonal, and (c) triangular. The terms are chosen according to the number of neighbors, not according to the shape enclosed by the links. In each case, the separation between the nodes is Δx. For the hexagonal and triangular grids, the mapping to a regular matrix is added, which facilitates writing the finite differences as a 4-point or 7-point stencil. Dashed lines represent isolines of planar fronts oriented in particular directions, that can effectively be captured by 1D discrete lattices.

where we have used $u_{i,j} = u(x_{i,j}, y_{i,j})$. In the limit of the finite difference scheme approaches the Laplacian as $\Delta x \to 0$.

The discrete RD equation can be written also as a gradient system similar to Eq. (4.5) with the Lyapunov functional

$$\begin{aligned} F = \sum_{k,l} \Bigg[V(u_{k,l}) - \frac{D}{4\Delta x^2} \big((u_{k-1,l} - u_{k,l})^2 + (u_{k+1,l} - u_{k,l})^2 \\ + (u_{k,l-1} - u_{k,l})^2 + (u_{k,l+1} - u_{k,l})^2 \big) \Bigg] \end{aligned} \tag{4.8}$$

Assuming that the values of $u_{i,j}$ are constant along parallel lines, with the values at a particular line labeled u_k, the previous equations can be simplified: For fronts advancing parallel to the grid, e.g. the velocity vector is parallel to the x-axis, $\phi = 0°$), the interactions in y disappear. The interaction term can thus be reduced to the same as in 1D (Eq. (4.4)).

Similar considerations can be done for $\phi = 45°$ (fronts parallel to the grid diagonals) and $\phi = 26.6°$ (fronts with a slope of $1:2$), by comparing which lines are connected by a specific link. However, the distance between these lines has to be taken into account. These fronts are exemplified in Fig. 4.2 (a) with their respective Δx_{eff}, their interaction terms are summarized in Table 4.1. While the interaction for $\phi = 45°$ is the same as for $\phi = 0°$, the effective coupling is twice its value due to the smaller line separation. For $\phi = 26.6°$; a different behavior has to be expected concerning propagation failure, as g differs qualitatively.

Not all orientations with the angle ϕ are easily approachable like this, as Δx_{eff} decreases quite quickly. But based on these simple cases, it is clear that the stencil or connectivity pattern of the 1D effective equation depends on the orientation of the front.

For each orientation, the speed of the planar front

$$v = \frac{\Delta x_{\text{eff}}}{\Delta \tau}, \quad \text{where } \Delta\tau \text{ such that } \quad u_{k+1}(t + \Delta\tau) = u_k(t)$$

depends on the separation between lines and the time it takes for the front to advance to the next site.

We repeated the same analysis on the 'Mehrstellen'-variation of the square lattice. It is obtained by including the diagonal connections between nodes. The equation in a 2D square domain uses a 9-point stencil:

$$\begin{aligned} \frac{\mathrm{d}u_{i,j}}{\mathrm{d}t} = \frac{D}{\Delta x^2} \Bigg(\frac{1}{6} (u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i+1,j+1}) \\ + \frac{2}{3} (u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}) - \frac{10}{3} u_{i,j} \Bigg) + f(u_{i,j}) \end{aligned} \tag{4.9}$$

It has been demonstrated (see Ref. 33) that using this stencil, the leading error term of the Laplacian does not have preferred directions. It is not clear whether or not the behavior of $u_{i,j}(t)$ is expected to be spatially isotropic. Its pseudo-2D examples are shown Table 4.1 as well.

lattice	ϕ	Δx_{eff}	interaction term $g(k)$	
square	$0° \mathrel{\widehat{=}} 90°$	Δx	$(u_{k-1} \quad -2u_k \quad +u_{k+1})$	(4.12)
	$26.6°$	$\Delta x / \sqrt{2}$	$(u_{k-2} \quad +u_{k-1} \quad -4u_k \quad +u_{k+1} \quad +u_{k+2})$	(4.13)
	$45°$	$\Delta x / \sqrt{5}$	$(u_{k-1} \quad -2u_k \quad +u_{k+1})$	(4.14)
square & diagonals	$0° \mathrel{\widehat{=}} 90°$	Δx	$(u_{k-1} \quad -2u_k \quad +u_{k+1})$	(4.15)
	$26.6°$	$\Delta x / \sqrt{2}$	$(\frac{1}{6}u_{k-3} +\frac{2}{3}u_{k-2} +\frac{5}{6}u_{k-1} -\frac{10}{3}u_k +\frac{5}{6}u_{k+1} +\frac{2}{3}u_{k+2} +\frac{1}{6}u_{k+3})$	(4.16)
	$45°$	$\Delta x / \sqrt{5}$	$(\frac{1}{6}u_{k-2} +\frac{4}{3}u_{k-1} -3u_k +\frac{4}{3}u_{k+1} +\frac{1}{6}u_{k+2})$	(4.17)
hexagonal	$0° \mathrel{\widehat{=}} 60°$	$\Delta x / 2$	$(u_{k-2} \quad +2u_{k-1} \quad -6u_k \quad +2u_{k+1} \quad +u_{k+2})$	(4.18)
	$30° \mathrel{\widehat{=}} 90°$	$\sqrt{3}\Delta x / 2$	$(u_{k-1} \quad -2u_k \quad +u_{k+1})$	(4.19)
triangular	$0° \mathrel{\widehat{=}} 60°$	$\Delta 3x / 4$	$\begin{cases} (2u_{k-1} \quad -3u_k \quad +u_{k+1}) & k \text{ is odd} \\ (u_{k-1} \quad -3u_k \quad +2u_{k+2}) & k \text{ is even} \end{cases}$	(4.20)
	$30° \mathrel{\widehat{=}} 90°$	$\sqrt{3}\Delta x / 2$	$(u_{k-1} \quad -2u_k \quad +u_{k+1})$	(4.21)

Table 4.1: Effective node distance Δx_{eff} and the interaction term $g(k)$ for the pseudo-2D counterparts with the front orientation ϕ.

4.3.2 Hexagonal and triangular lattices

Both, the hexagonal and triangular lattice are conjugate to each other. In a triangular tiling, the vertices of the triangles can be seen as centerpoints of hexagons and vice versa in an hexagonal tesselation. To reflect the symmetry of the chosen lattice, we denote the lattice where each node is connected to 6 neighbors located at distance Δx as hexagonal lattice. It shows a 6-fold symmetry, each node representing the center of an hexagon, while the links form equilateral triangles. The lattice is more densely packed then the square lattice, covering less area per node using the same Δx. The hexagonal grid could be seen as a model for diffusion on an fcc(111) surface where diffusion between on-top sites is most important.

In the triangular grid (Fig. 4.2 (c)), each node is connected to 3 neighbors located at distance Δx. It is less dense than the square lattice, filling larger space with the same number of nodes. The triangular lattice could be valuable as a model for diffusion on an fcc(111), where diffusion between fcc and hcp sites is most prevalent.

For easier manipulation, both lattices can be mapped into a regular array indexed by i, j with additional connections, as illustrated on the right in Fig. 4.2 (b,c). The equation in a 2D hexagonal domain can be expressed by using a 7-point stencil with a conditional expression:

$$\begin{aligned} \frac{\mathrm{d}u_{i,j}}{\mathrm{d}t} = f(u_{i,j}) + \frac{2D}{3\Delta x^2}(&u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} \\ &- 6u_{i,j} + \begin{cases} u_{i+1,j+1} + u_{i+1,j-1}) & j \text{ is odd} \\ u_{i-1,j+1} + u_{i-1,j-1}) & j \text{ is even} \end{cases} \end{aligned} \tag{4.10}$$

That said, three equivalent expressions can be visualized by turning the matrix representation by 90°. The coupling between nodes $2D/(3\Delta x^2)$ was selected such that the finite difference scheme

captured the Laplacian of a continous quadratic function.

For the triangular lattice, multiple stencils are obtainable by reducing the number of links in the hexagonal lattice. We selected the 4-point stencil represented through:

$$\frac{\mathrm{d}u_{i,j}}{\mathrm{d}t} = f(u_{i,j}) + \frac{4D}{3\Delta x^2}\begin{cases}(u_{i-1,j}+u_{i,j+1}+u_{i,j-1}-3u_{i,j}) & i+j \text{ is odd}\\ (u_{i+1,j}+u_{i,j+1}+u_{i,j-1}-3u_{i,j}) & i+j \text{ is even}\end{cases} \tag{4.11}$$

Similar considerations concerning fronts along isovalue lines were done for both lattices and summarized in Table 4.1 for $\phi = 0°$ and $\phi = 30°$

4.4 Propagation failure in 1D and in 2D

After having defined the discrete lattices one can proceed to evaluate the effect of discreteness and orientation on the behavior of traveling fronts. Although the butterfly potential defined in Eq. (4.3) has four parameters, most of the phenomena of fronts can be captured using a generic two dimensional section (as explained in Ref. 9). In the present treatment we choose $\alpha = -10$ and $\beta = 2$, shown in Fig. 4.1. We designate as $u_{01,k}$ or just u_{01} the front solution that verifies $u_{01,k} \xrightarrow[k\to-\infty]{} \tilde{u}_0$ and $u_{01,k} \xrightarrow[k\to+\infty]{} \tilde{u}_1$. This is done analogously for $u_{12,k}$ and $u_{02,k}$.

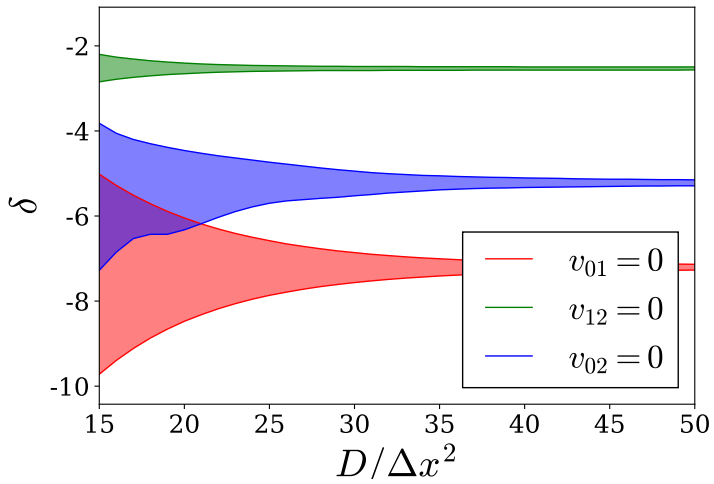


Figure 4.3: Extent of PF for the three fronts as a function of $D/\Delta x^2$ for $\gamma = 17$. While the position of the pinning region is different for each front, it generally shrinks with increasing $D/\Delta x^2$. For sufficiently small $D/\Delta x^2 < 15$, propagation failure dominates for almost any δ.

A selection of the results using Eq. (4.4) are shown in figures 4.3 and 4.4, but they are also representative of 2D fronts using our simplification for certain orientations (e.g. Eq. 4.12 – 4.14), which will be discussed in more detail in section 4.4.1. We designate v_{01}, v_{12}, and v_{02} the effective speed of the front solutions $u_{01,k}(t)$, $u_{12,k}(t)$, and $u_{02,k}(t)$. Figure 4.3 explores a 'vertical section' ($\gamma = 17$) of Fig. 4.1, always inside the tristability region. As expected, for finite $D/\Delta x^2$ there are intervals where the speeds v_{01}, v_{12} and v_{02} of the three fronts are identically zero. The figure shows the upper and the lower limits of the pinning range. At these points the front undergoes

a 'pinning-depinning transition'. The front u_{01} between $\tilde{u}_0$ and $\tilde{u}_1$ seems the more susceptible to propagation failure, and the front u_{12} between $\tilde{u}_1$ and $\tilde{u}_2$, the more immune. The three widths of the propagation failure intervals shrinks to zero as $D/\Delta x^2$ grows.

Figure 4.4 (a) offers the actual values of the speeds v_{01}, v_{12} and v_{02} for $\gamma = 18.75$ and $D/\Delta x^2 = 15$ (where these fronts exist), exhibiting the characteristic discontinuity associated to PF. This section of the butterfly diagram includes the equipotential (also known as Maxwell) point $\gamma = 18.75, \delta = -5$ where all three speeds are zero for any $D/\Delta x^2$, so the three PF intervals include that point.

Figure 4.4 (b) shows an analogue 'horizontal' section of the butterfly diagram for $\delta = -3$ with $D/\Delta x^2 = 15$. Now the PF intervals do not overlap at all, and it is the connection $u_{02,k}$ between $\tilde{u}_0$ and $\tilde{u}_2$ the one that looks more susceptible to PF; while as before, $u_{12,k}$ is only pinned inside a narrow interval. Interestingly, there is a point where the three speeds become equal $v_{01} = v_{12} = v_{02} \neq 0$, and the $u_{02,k}$ front solution ceases to exist. This is the front-splitting instability [9, 34, 35] that now depends on $D/\Delta x^2$. At this point the $u_{02}(x_k)$ solution acquires an infinitely wide step where $u_{02} \approx \tilde{u}_1$ the intermediate stable state that also has a potential value between the other two minima.

Although the signs of the speeds depend on the potential values at the equilibria, the speeds depend on the whole shape of $V(u)$. In this article we used a nonzero coefficient β that leads to a asymmetric butterfly wings (see Fig. 1) and therefore unequal pinning ranges for the fronts u_{01} and u_{12}.

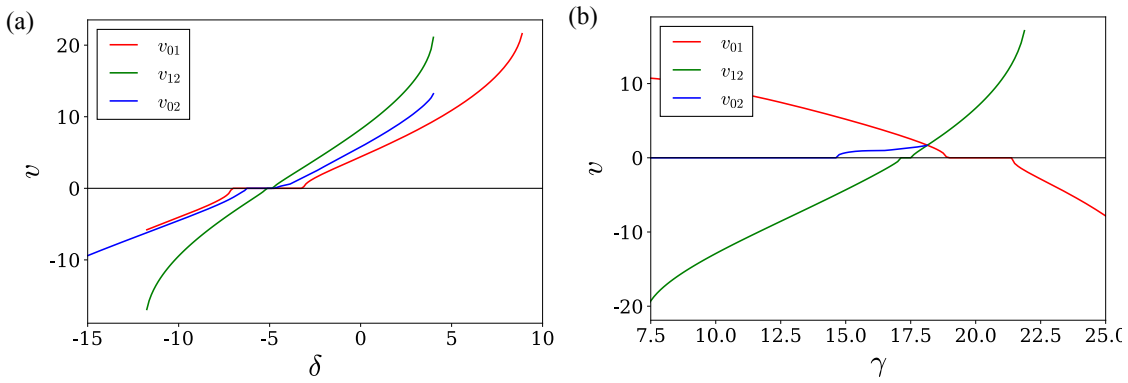


Figure 4.4: Front speeds for solutions of Eq. (4.4) in different sections of Fig. 4.1 using $D/\Delta x^2 = 15$. The sign hints the direction of front propagation. Exemplifies the range of propagation failure depending on front γ or δ respectively. (a) $\gamma = 18.75$ shows PF around the equipotential point. (b) $\delta = -3$ shows both, propagation failure and front splitting instability.

4.4.1 Effect of orientation of 2D fronts

While in the continuous case, the front speed depends continuously on the shape of the potential $V(u)$ (which means the parameters $\alpha, \beta, \gamma, \delta$) and the diffusivity D, in the discrete 1D case, a pinning region where $v = 0$ is expected for small coupling $D/\Delta x^2$.

In 2D, the situation is more complex, since the selected lattice and the front orientation influence the node interaction as worked out in section 4.3. Assuming for instance that the solution is a planar front between states $\tilde{u}_0$ and $\tilde{u}_2$, everything can be reduced to the solution $u_{02,k}$ of a 1D

equation. The change of front speed for the different surface lattices and orientations as a function of δ is illustrated in Fig. 4.5.

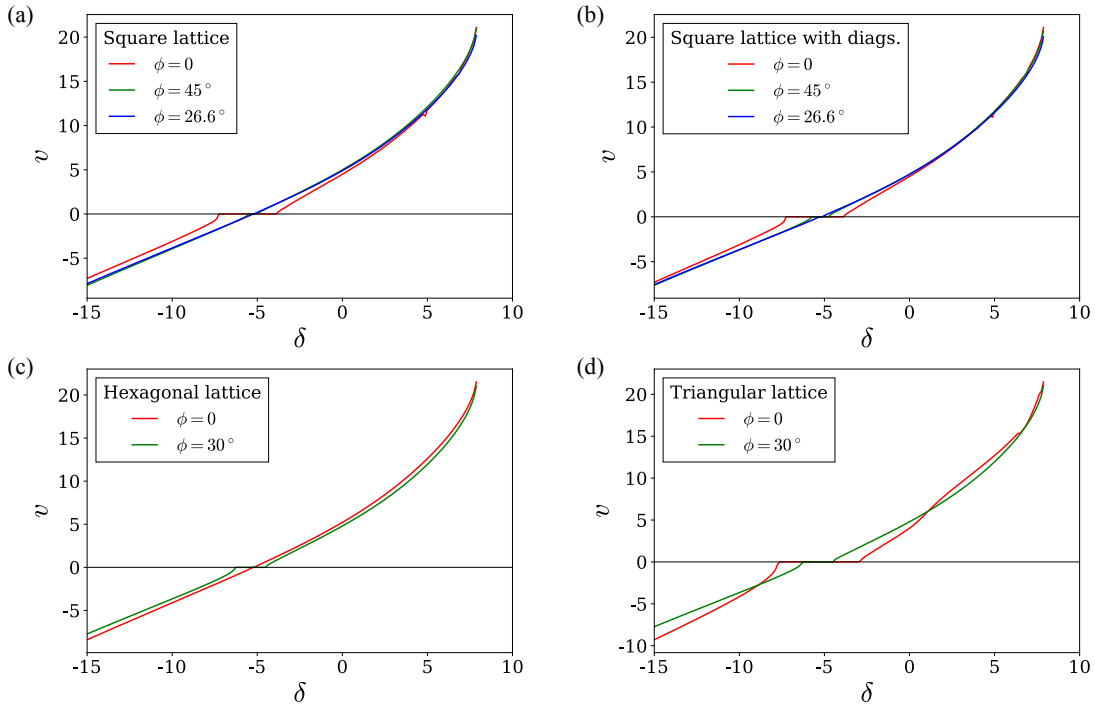


Figure 4.5: Front speeds for the u_{02} front in 2D lattices using the simplified isocurve approach. $\gamma = 17$, $D/\Delta x^2 = 15$. (a) Square lattice Eqs. (4.12,4.13,4.14), (b) Square lattice with diagonal connections Eqs. (4.15,4.16,4.17), (c) Hexagonal lattice Eqs. (4.18,4.19), (d) Triangular lattice Eqs. (4.20,4.21). For large $D/\Delta x^2$ and away from the pinning region, the speeds associated with different orientations approach each other, while near the pinning region, some orientations are *stickier* than others.

In direct comparison, all the figures share some common features. Far from the pinning region or for large $D/\Delta x^2$, the lattice and the front orientation do not play a large role. On the other hand, the size of the pinning region strongly depends on the lattice and the orientation: some can be described as being *stickier*; and the propagation of fronts having isolines parallel to some edges seem more prone to fail.

The results for both square lattices (with and without diagonal connections) are extremely similar. The additional connections do not eliminate the PF along certain orientations and seem to have only a small influence on the qualitative behavior.

Overall, the size of the pinning region tends to decrease with the connectivity of the lattice (triangular lattice $>$ square lattice $>$ hexagonal lattice). This seems plausible, regarding the number of 'active' links in the 'worst case' for propagation: 4 links are used for propagation in the hexagonal lattice while only 2 are active for the square and triangular lattice. Also, the links for the square grid are more effective being exactly perpendicular to the isolines, while the links in the triangular lattice are rotated by 30°.

Certainly, this could lead to anisotropic behavior. Although all orientations should be studied for a conclusive answer, based on our limited evidence we can claim, that in our selection, hexagonal is the most isotropic and the triangular lattice is the most anisotropic case. This is further

discussed based on numerical simulations in section 4.5.3.

4.4.2 Bifurcation structure

The main mechanism behind PF is the existence of static solutions connecting two different states of different potentials. Such nontrivial solutions persist after small changes to the reaction function. One way of understanding this situation is writing the stationary front equation:

$$0 = \frac{D}{\Delta x^2}(u_{k-1} + u_{k+1} - 2u_k) + f(u_k). \tag{4.22}$$

or similar (depending on lattice and orientation) and interpreting the index k as a discrete 'time'. Solutions u_k for $k \in \mathbb{Z}$ that satisfy initial and final conditions:

$$u_k \xrightarrow[k\to-\infty]{} \tilde{u}_- \quad \text{and} \quad u_k \xrightarrow[k\to+\infty]{} \tilde{u}_+$$

should belong to both the unstable manifold $\mathscr{W}^u(\tilde{u}_-)$ and the stable manifold $\mathscr{W}^s(\tilde{u}_+)$. Such heteroclinic connections are robust [22, 23, 36] and when represented as points in parameter space, define a branch that connects all the solutions that are equivalent modulo a shift, shown in Fig. 4.6 (a) in blue. The branch also connects other solutions that are unstable in the original time, shown in Fig. 4.6 (a) in red. This kind of branch traces a snaking trajectory as a parameter is varied, as shown in Fig. 4.6 (b) for each one of the fronts (obtained with the continuation algorithms of software AUTO [37]). The projections of the snaking curves on the parameter axis define the pinning ranges. Each of the snaking branches is centered around the place where the corresponding roots have same potential values (indicated with vertical segments connecting branches of roots in Fig. 4.6 (b)).

There is a simple explanation (borrowed from Ref. 38) for the stability or instability of the front solutions $u_{01,i}$ depicted in Fig. 4.6 (a). Unstable fronts (in red) connect the stable states $\tilde{u}_0 = -2.893$ and $\tilde{u}_1 = 0.438$ and have a node $u_i \approx -1.7$ close close to an unstable root of $f(u)$ that we name $\tilde{u}_{\text{uns}} = -1.558$. At that point $f(\tilde{u}_{\text{uns}}) = 0$ and $f'(\tilde{u}_{\text{uns}}) > 0$. In the limit of vanishing $D/\Delta x^2$ such type of front becomes:

$$u_{01,k} = \begin{cases} \tilde{u}_0 & \text{for } k < 0, \\ \tilde{u}_{\text{uns}} & \text{for } k = 0, \\ \tilde{u}_1 & \text{for } k > 0. \end{cases}$$

and is unstable. Stable fronts (in blue) have nodes far from $\tilde{u}_{\text{uns}}$ and in the limit of vanishing $D/\Delta x^2$ such type of front becomes:

$$u_{01,k} = \begin{cases} \tilde{u}_0 & \text{for } k < 0, \\ \tilde{u}_1 & \text{for } k \geq 0. \end{cases}$$

and is stable.

Now for front solutions between $\tilde{u}_0$ and $\tilde{u}_2$ there are more ways to build unstable connections (in the limit of vanishing $D/\Delta x^2$) since there are two unstable roots $\tilde{u}_{\text{uns}}$ inside of the interval

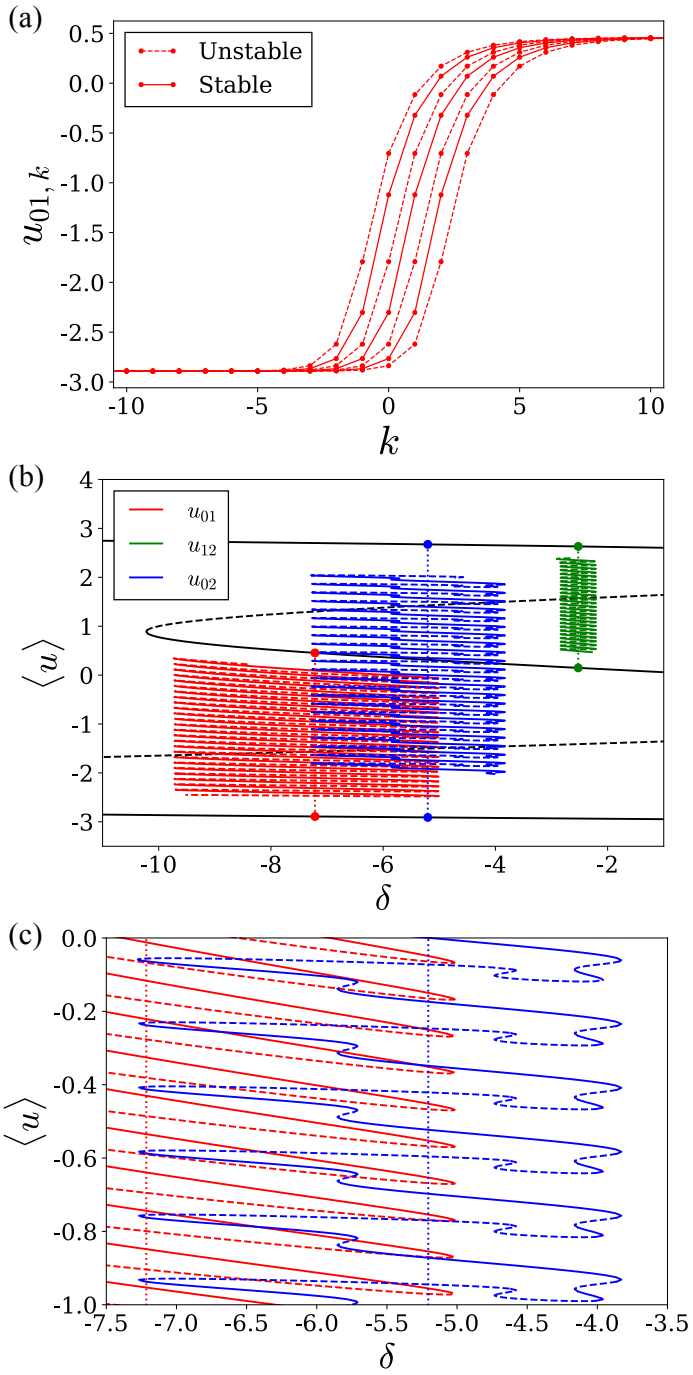


Figure 4.6: (a) Illustration of PF in a 1D domain: Multiple "frozen" front solutions of Eq. (4.4) between $\tilde{u}_0$ and $\tilde{u}_1$ roots of the quintic polynomial with coefficients $\alpha = -10, \beta = 2, \gamma = 17, \delta = -7$, and spatial coupling $D/\Delta x^2 = 15$. (b) Codimension-1 bifurcation diagram for the fronts connecting stable solutions with $\alpha = -10, \beta = 2, \gamma = 17$, $D/\Delta x^2 = 15$, corresponding thus to a vertical section in Fig. 4.4. The frozen front solutions as shown in (a) form alternating stable (solid) and unstable (dashed) branches, following a snaking pattern; points of the same branch located at the same δ correspond to shifted copies of the same two solutions (stable or unstable). Vertical dotted lines connect roots of equal potential are also depicted to emphasize how the snaking is organized around the equipotential condition (compare with Figs. 4.3 and 4.4). (c) Zoom into the fine-structure of the snaking branches. u_{02} exhibits multiple turns and several changes of stability within the same period.

defined by $\tilde{u}_0$ and $\tilde{u}_2$. There are also more ways to build stable connections. A first stable solution corresponds to:

$$u_{02,k} = \begin{cases} \tilde{u}_0 & \text{for } k < 0, \\ \tilde{u}_2 & \text{for } k \geq 0. \end{cases}$$

and a second stable solution corresponds to:

$$u_{02,k} = \begin{cases} \tilde{u}_0 & \text{for } k < 0, \\ \tilde{u}_1 & \text{for } k = 0, \\ \tilde{u}_2 & \text{for } k > 0. \end{cases}$$

Both solutions are stable when coupling vanishes. Summarizing the argument, and for finite coupling, the family of fronts u_{02} has several stable and unstable solutions that are represented in Fig. 4.6 (b) (blue curve).

Some of these multiple (stable and unstable) frozen fronts and their bifurcations appear in Figs. 4.7 and 4.8, for square and triangular lattices respectively, that show the regions where frozen front solutions u_{01}, u_{12} and u_{02} exist (indicated by red, green and blue shading respectively). As the precise location of turning points of u_{02} change with parameters, lattice structure and orientation of the fronts, the upper and lower limits will show peculiar structures (cusps) that are not present in the PF regions of front solutions u_{01} and u_{12}.

As shown by Humphries and Wilds [25], neither simple bistable functions (such as $f(u) = -u(u-a)(u-1)$) nor tristable functions (such as $f(u) = -u(u-b)(u-1/2)(u-a)(u-1)$) exhibit the additional cusps of frozen fronts presented here. As long as the roots are well separated the bifurcation structure of the frozen fronts will remain simple. Only a flexible polynomial, as the one defined in Eq. (4.2), will reveal the full richness of propagation failure in discrete lattices.

In the (γ, δ) plane these three 'horns' have different widths but they are all located along the corresponding equipotential curves (dashed lines) of the butterfly diagram, and they extend outside the central diamond-shaped tristability region (compare with Fig. 4.1). The overlap of the three horns defines a small rectangular region, inside the tristability region, where all three fronts are frozen. This rectangular region shrinks to the equipotential point mentioned before as coupling approaches large values, while it fills the tristability diamond as $D/\Delta x^2$ approaches 0. It is important to notice that the u_{02} front does not exist beyond the overlapping region, as a result of the front splitting instability [9, 34, 35].

4.5 2D Time evolution

A defining aspect for the 2D pattern evolution with regard to discreteness is the implementation of the 2D lattice. The following key aspects have to be considered: a) lattice resolution, b) boundary conditions, c) geometry of the diffusion grid, and d) coupling strength $D/\Delta x^2$ as measure of discreteness. In the present work, we focus mainly on the influence of c) and d) while compromising a) and b).

As discussed (sec. 4.3), we utilize a regular square lattice, an equilateral hexagonal and an

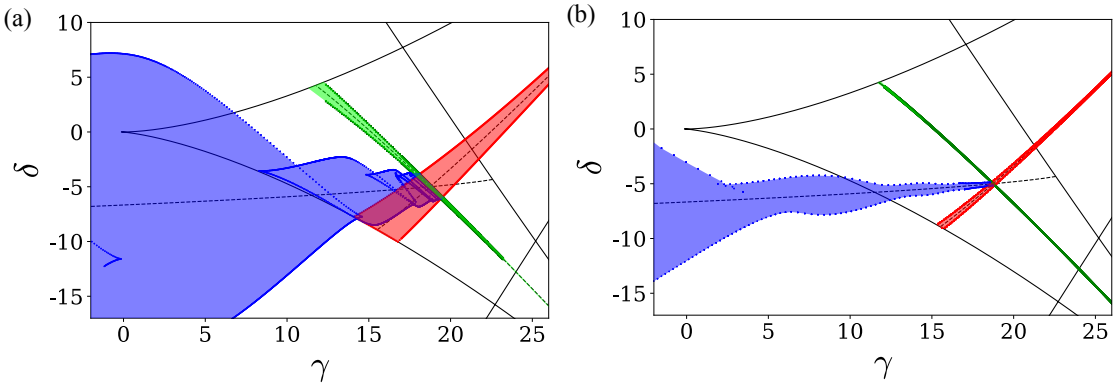


Figure 4.7: The three regions of frozen fronts in a square lattice with $D/\Delta x^2 = 15$. (a) Orientation $\phi = 0$, solutions of Eq. (4.12) (b) Orientation $\phi = 45°$, solutions of Eq. (4.13). Red, green and blue shadings indicate where frozen fronts u_{01}, u_{12} and u_{02} (respectively) exist. The dashed lines indicate the additional folds resulting from the snaking branch (as shown in Fig. 4.6).

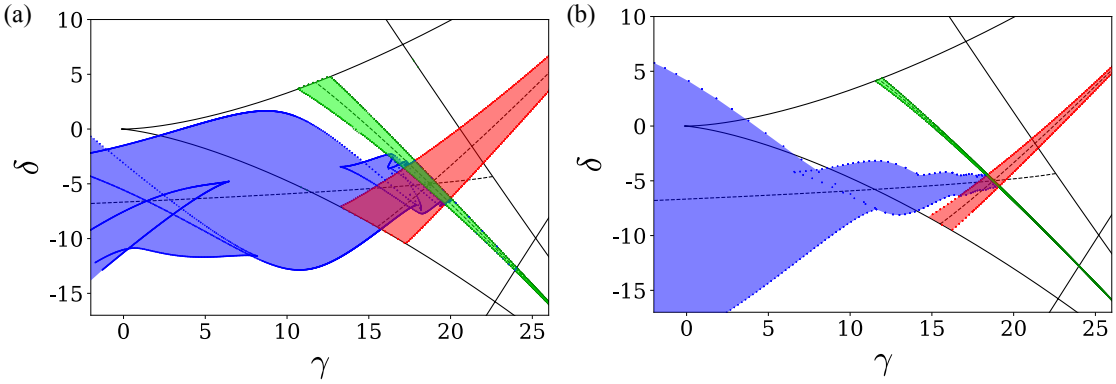


Figure 4.8: The three regions of frozen fronts in a triangular lattice with $D/\Delta x^2 = 15$. (a) Orientation $\phi = 0$, solutions of Eq. (4.20) (b) Orientation $\phi = 30°$, solutions of Eq. (4.21). Red, green and blue shadings indicate where frozen fronts u_{01}, u_{12} and u_{02} (respectively) exist. The dashed lines indicate the aditional folds resulting from the snaking branch (as shown in Fig. 4.6).

equilateral triangular lattice. The discrete diffusion coefficient D was then used as the main variable in numerical simulation. Depending on the lattice, the effective coupling strengths were applied (compare section 4.3.2).

Concerning the resolution, there are four defining factors: Number of nodes, length of the links, area of the surface sample and the lattice itself. A square-shaped surface was utilized, represented by a 100×100 matrix connected by square lattice. The number of nodes in x, y was adjusted to fit the same surface, while maintaining equidistant nodes (Δx). Therefore, the nodal resolution is slightly different for each grid, with the hexagonal resolution being larger and the triangular resolution being smaller.

Simple zero-flux boundary condition was chosen. Possible front distortion at the boundaries could be mostly neglected either by limiting the simulation time, or by utilizing initial pattern were no boundaries are involved. For the latter, a pattern was created by dividing the area into four domains with octagonal rings. Each area is populated with a different solution $\tilde{u_i}$, thus allowing observation of all three fronts (u_{01}, u_{02}, and u_{12}). Also, due to the geometrical shape, a

first evaluation of the effect of angular alignment can be done. The initial patterns are shown in Fig. 4.9.

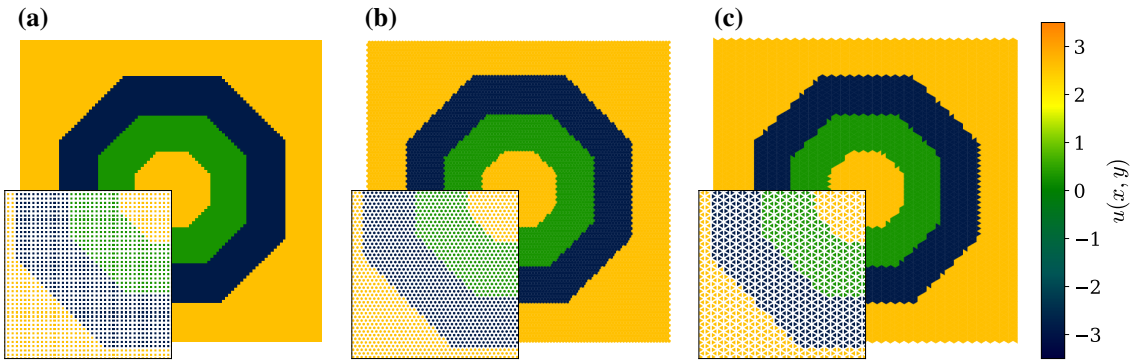


Figure 4.9: Illustration of the initial condition of the surface for $\alpha = -10$, $\beta = 2$, $\gamma = 18.75$, $\delta = -3$. From left to right, the surfaces are constructed with the square, hexagonal and triangular lattice. The colors represent one of the stable solutions (dark blue: $\tilde{u}_0$; green: $\tilde{u}_1$; orange: $\tilde{u}_2$)

The spatio-temporal pattern have been evolved using a sophisticated ordinary differential equation solver (LSODA), to avoid checkerboard instability in Python with parameter values within the tristable area around the equistable point in the plane spanned by $\alpha = -10$ and $\beta = 2$, namely $\gamma \in \{15, 18.75, 20\}$ and $\delta \in \{-3, -5, -7\}$. Coupling was enabled by using different $D \leq 15$ and $\Delta x^2 = 1$.

4.5.1 Diffusion grid

The type of connectivity has been shown to influence the reaction-diffusion system in discrete lattices [39]. As discussed in section 4.3, front propagation is inhibited by a surface 'friction', which can lead to propagation failure. This resistance is especially sensible to front alignment relative to the lattice edges. For instance, a front aligned with the square lattice exhibits more friction than the same front on a 45°-rotated lattice. Thus, a complex front with multiple orientations could propagate or fail to do so at different places. Figures 4.10 and 4.11 illustrate this phenomenon: depending on the parameters, fronts can either propagate freely, remain frozen in the initial state, or propagate until friction inhibits further propagation and a metastable orientation is reached. These metastable structures often reveal the geometry of the underlying grid, e.g. squares with the square lattice or perfect hexagons with the hexagonal lattice.

For $\delta = 15, \gamma = -3$ (Fig. 4.10), the potentials associated to the three stable states satisfy $V(\tilde{u}_0) < V(\tilde{u}_2) < V(\tilde{u}_1)$, so both $\tilde{u}_0$ and $\tilde{u}_2$ invade regions initially covered with $\tilde{u}_1$. The interface between $\tilde{u}_1$ and $\tilde{u}_2$ (u_{12}), freezes after the formation of a lattice-aligned square-shaped area. u_{01} is slower and additionally inhibited in the same dimensions as u_{12}, thus propagation of orientations with less friction lead to the development of a transient, misaligned squared region.

For $\delta = 18.75, \gamma = -3$, the potentials associated to the three stable states satisfy $V(\tilde{u}_0) < V(\tilde{u}_1) < V(\tilde{u}_2)$, and a similar observation can be made. Here, the front solution u_{02} does not exist since front-splitting occurs. However, while $\tilde{u}_1$ can invade the inner regime of $\tilde{u}_2$ completely, diffusion to the outer regime is inhibited and leads again to propagation failure after a square shape

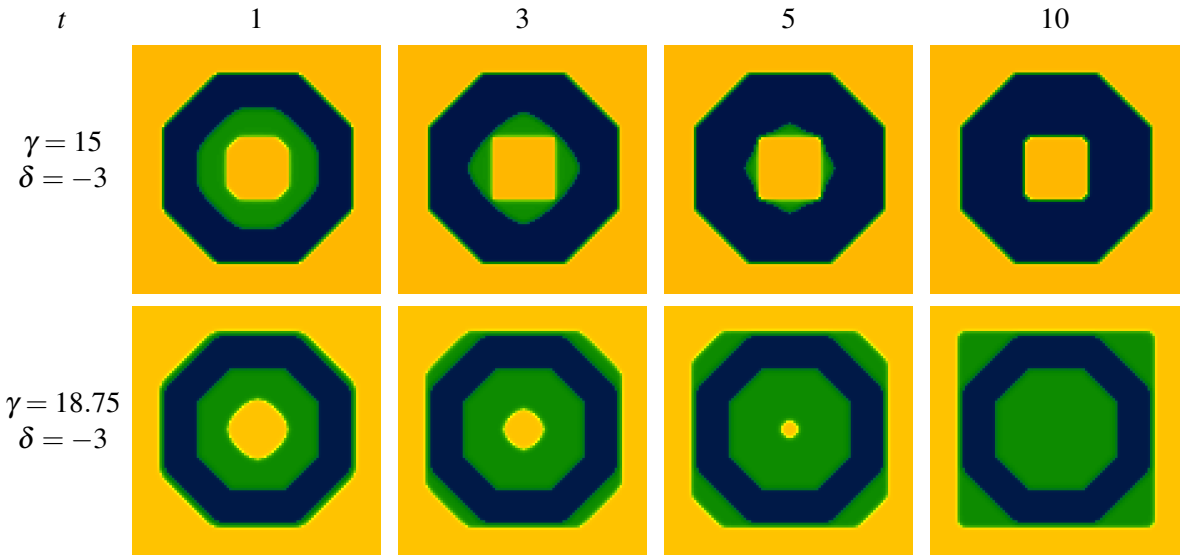


Figure 4.10: Time-evolution of the initial surface pattern using the square lattice and a coupling strength $D/\Delta x^2 = 5$. These examples illustrate the front propagation for the square mesh grid. For the regular square grids, propagation directly parallel to the edges is inhibited, fronts freeze after forming a metastable square-shaped structure.

is reached. This anisotropic behavior can be explained by the facet size of the inner regime.

An analogue explanation can be made for the time-series illustrated in Fig. 4.11, while highlighting the difference due to lattice structure.

The hexagonal lattice is highly symmetric and has a larger connectivity than the square lattice. This more intense coupling leads to fewer orientations being inhibited. This is illustrated by a lower threshold for the metastable hexagonal shape.

The triangular lattice shows opposite properties, it has the lowest symmetry and connectivity of the compared lattices. Due to its relationship with the hexagonal lattice, hexagonal shapes can form, though they are somewhat deformed. Higher thresholds for the coupling strength are required for defreezing the fronts.

The effect of angular front alignment is further analyzed in section 4.5.3.

4.5.2 Coupling strength

As shown in the previous section, a discrete diffusion grid can introduce an additional resistance to diffusion. Depending on the coupling strength between the nodes of a given grid, fronts propagate or stay frozen. A characteristic time-evolution for a series of coupling strengths is shown in Fig. 4.12 (and further shown in SM).

The curves for $\tilde{u}_2$ behave generally as expected: for $D \leq 4$, the front u_{12} is frozen after slight facet optimizations. The fronts start moving for higher D, while for $5 \leq D \leq 7$ a metastable hexagonal region could form, which is destabilized by reaching the boundaries. The region covered with state $\tilde{u}_1$ mostly mirrors the changes in the region covered with state $\tilde{u}_2$, except for $D = 10$, where the slope decreases compared to $D = 9$. This is due to the front u_{01} being much more inhibited than the front u_{12}; propagation failure still occurs at $D = 9$. After the region covered with $\tilde{u}_0$ slowly

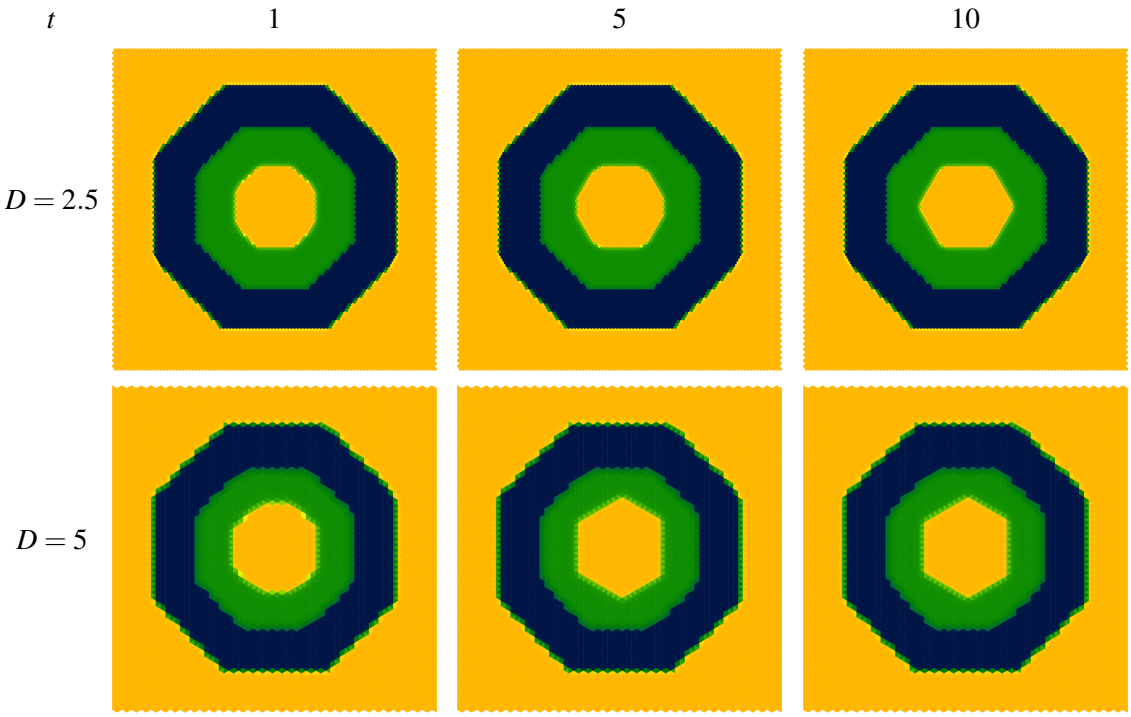


Figure 4.11: Time-evolution of the initial surface pattern for hexagonal and triangular diffusion grid. Parameter values $\gamma = 15$, $\delta = -3$, $\Delta x = 1$ imply that the potentials verify $V(\tilde{u}_2) < V(\tilde{u}_1) < V(\tilde{u}_0)$. Both lattices can produce hexagonal structures; whereas for the triangular lattice, the structure will be an imperfect hexagon.

starts moving, it grows steadily by invading the state $\tilde{u}_1$.

4.5.3 Angular dependent behavior

To further investigate and quantify the consequence of different front orientations to the front velocity, a different initial pattern was used. Instead of using the symmetric shape (Fig. 4.9), the surface was divided by a line with angle ϕ into two equal parts, each half being populated with the value of a different state $\tilde{u}_k$.

Two pattern examples are shown in Fig. 4.13 for different values of D and time t. These results show that boundary effects for long evolution times can't be neglected. For small diffusivity (Fig. 4.13 (a)), a front separation takes place. While the front for $\phi = 30°$ propagates, the left boundary acts as a *nucleation site* where a frozen $\phi = 0°$-front forms. The result is a 'kink' or 'corner', which itself propagates further right. At the right boundary, the front does not behave in the same manner, though a small kink is visible which propagates slightly faster than the $\phi = 30°$-front.

For larger D (Fig. 4.13 (b), a smaller time is necessary to distort the initial front. Also, since the $\phi = 0°$-front can propagate, no sharp kink is formed, but a smooth bending takes place. The effect is again more pronounced at the left side of the shape, where the front propagates faster.

To limit boundary effects on angular dependent front velocities, numerical evolution was done only up to $t = 1$. Spatio-temporal patterns were computed for each lattice (square, hexagonal and triangular) and front solution (u_{01}, u_{02} or u_{12}). Comparison with the initial state allows quantifica-

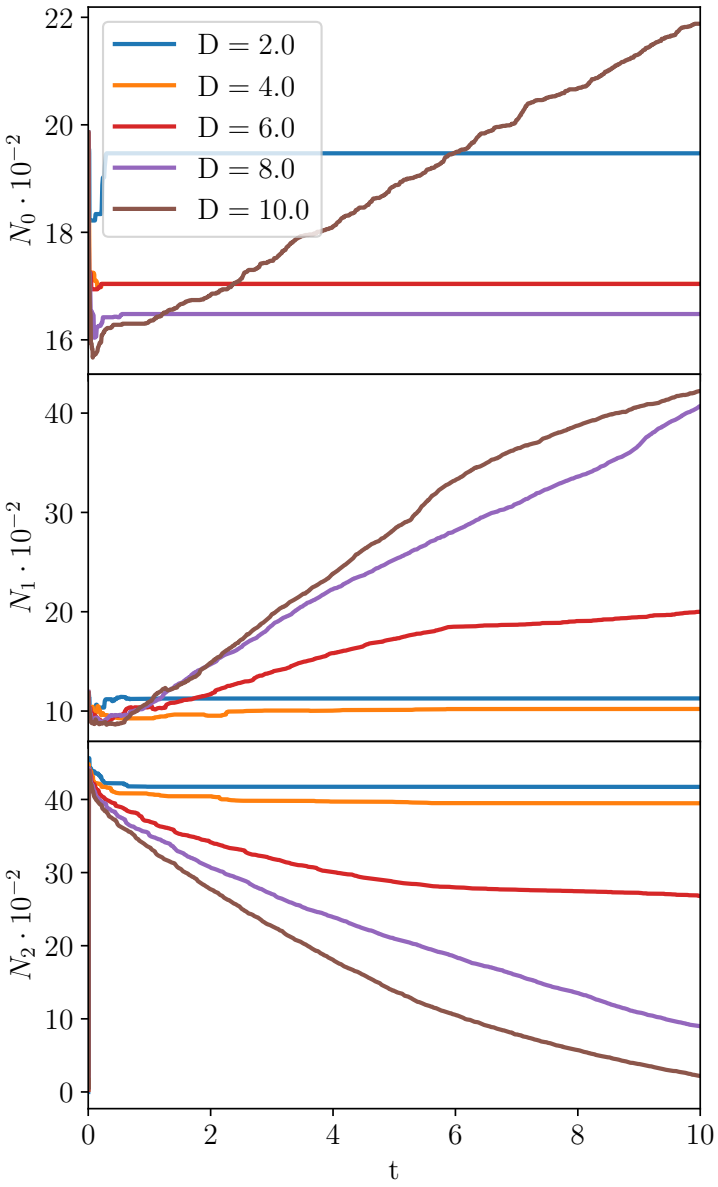


Figure 4.12: Time-evolution of the initial surface pattern with the triangular lattice and varying coupling strengths, illustrated by the relative area change for each stable solution $\tilde{u}_i$ during evolution, defined in Eq. (4.23). Parameter values $\gamma = 18.75$, $\delta = -3$ lead to potential values that verify $V(\tilde{u}_0) < V(\tilde{u}_1) < V(\tilde{u}_2)$.
The graphs exemplify the change of behavior by increasing $D/\Delta x^2$, namely the subsequent propagation failure depending on facet and front.

tion of the change by different methods:

1. Evaluating the number of nodes that deviate less than a small quantity ε from one of the stable states $\tilde{u}$:

$$N_k(t) = \sum_{i,j} \chi_{[\tilde{u}_k - \varepsilon, \tilde{u}_k + \varepsilon]}\left(u_{i,j}(t)\right) \tag{4.23}$$

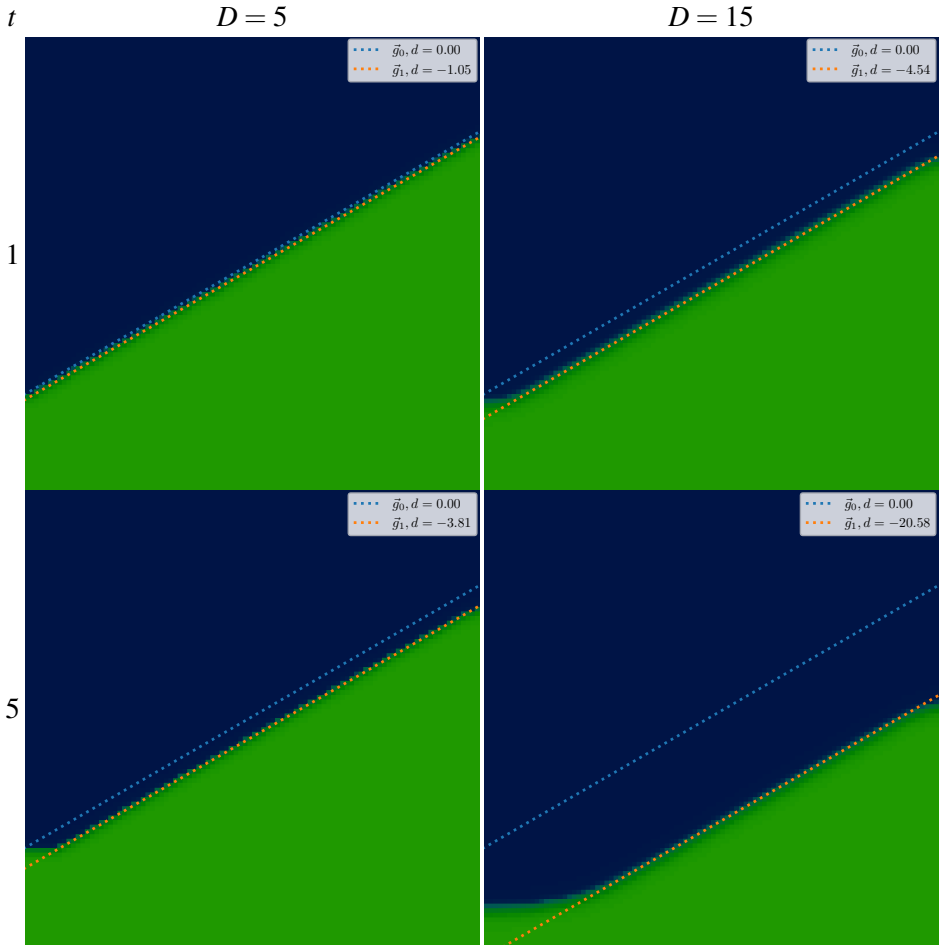


Figure 4.13: Illustration of the boundary effects and the determination of front movement according to Eq. (4.26b) for the front orientation $\phi = 30°$, $\gamma = 15$, $\delta = -5$, $\Delta x = 1$, u_{01}. For $D = 5$ the long evolution time leads to the formation of a pronounced 'corner'. For $D = 15$, diffusion is quite large and all front orientations propagate. As a result, no corner is formed, and a smooth bending of the front is visible. The fitting of $\vec{g}_1$ seems a reliable method for the identification of front motion.

with the indicator function $\chi_{[a,b]}$

$$\chi_{[a,b]}(u) \begin{cases} 1 & \text{if } u \in [a,b] \\ 0 & \text{otherwise} \end{cases} \tag{4.24}$$

and a small values of ε, here we used $\varepsilon = 0.1$. While this may be the most intuitive approach, some nodes may be misattributed or even fall out of the criteria. Also, for every time evolution, up to three different curves have to be analyzed for a complete picture.

2. Calculating the difference of each node to its initial state and integrating over the surface:

$$M(t) = \sum_{i,j} (u_{i,j}(t) - u_{i,j}(0)) \tag{4.25}$$

With this approach, the general strength of change and the propagation direction can be

illustrated. A drawback is, that it could overestimate the front movement due to a broad interphase formation.

3. Neglecting boundary effects and further assuming that the front propagates exactly parallel to its initial state, the distance d between the initial front and the propagated front can be seen as the distance between two straight lines:

$$\vec{g_0} = \vec{a} + \lambda \vec{e}(\phi), \tag{4.26a}$$

$$\vec{g_1} = \vec{g_0} + d(t)\vec{e}(\phi + \frac{\pi}{2}) \tag{4.26b}$$

$\overrightarrow{d}$ being a vector to a point at the initial front and $\overrightarrow{e}(\phi)$ a unit vector with the angle ϕ. The distance d can then be obtained by optimizing the position of $\vec{g_1}$. This method relies on some uncertainty concerning how well the propagated front can be fitted. The minimized error from fitting includes interphase formation, but also the deviation due to the boundary conditions.

In this article, we optimized $\vec{g_1}$ by minimizing the distance to all nodes, not counted by the indicator function Eq.(4.24) for both solutions with a $\varepsilon = 0.1$. This approach is robust as shown in Fig. 4.13. d is computed in the scale of node number, e.g. for ϕ=0, $d = 1$ means a front shift of one row.

For $\gamma = 15, \delta = -5$, the front movement (as defined in Eq. (4.26b)) for different front solutions, diffusion grids, ϕ and D is shown in Fig. 4.14 and will be discussed in detail. (Further data can be obtained in the supporting material).

The spatio-temporal evolution confirms the previous considerations concerning the observed velocities made in section 4.3. In general, all surface lattices show special angles, at which the diffusion behavior is extreme. Around these extremes, front propagation displays a flower petal-like curve.

For the highlighted set of parameters, each front behaves in a very distinct way, independent of the utilized grid. The potential difference $V(\tilde{u}_0) - V(\tilde{u}_2)$ is small, which leads to only small front movement. Additionally, since $\tilde{u}_1$ is a stabilized intermediate of the front solution, propagation is significantly overlapped by front broadening. While increasing D should accelerate propagation, it also broadens the front interphase. These effects increase also the difficulty to accurately estimate the calculated distances. While the trends are consistent, this could result in artifacts as the jumping between minimum and maximum for $\phi = 45°$ using the square lattice (compare $D = 6, 8, 10$). The evolved surface patterns show a frozen front up to $D = 10$.

The front solution for u_{12} can be seen as having polar opposite properties compared to u_{02}. While the potential difference between $\tilde{u}_1$ and $\tilde{u}_2$ is large, the potential maxima connecting both solutions poses only a marginal energy barrier to overcome. Therefore, even small diffusion coefficients are sufficient to overcome propagation failure for most orientations. Therefore, only small D hints at the characteristics of the underlying surface. For $D \geq 6$, the curve looks increasingly circular with only slight distortions at the extreme angles.

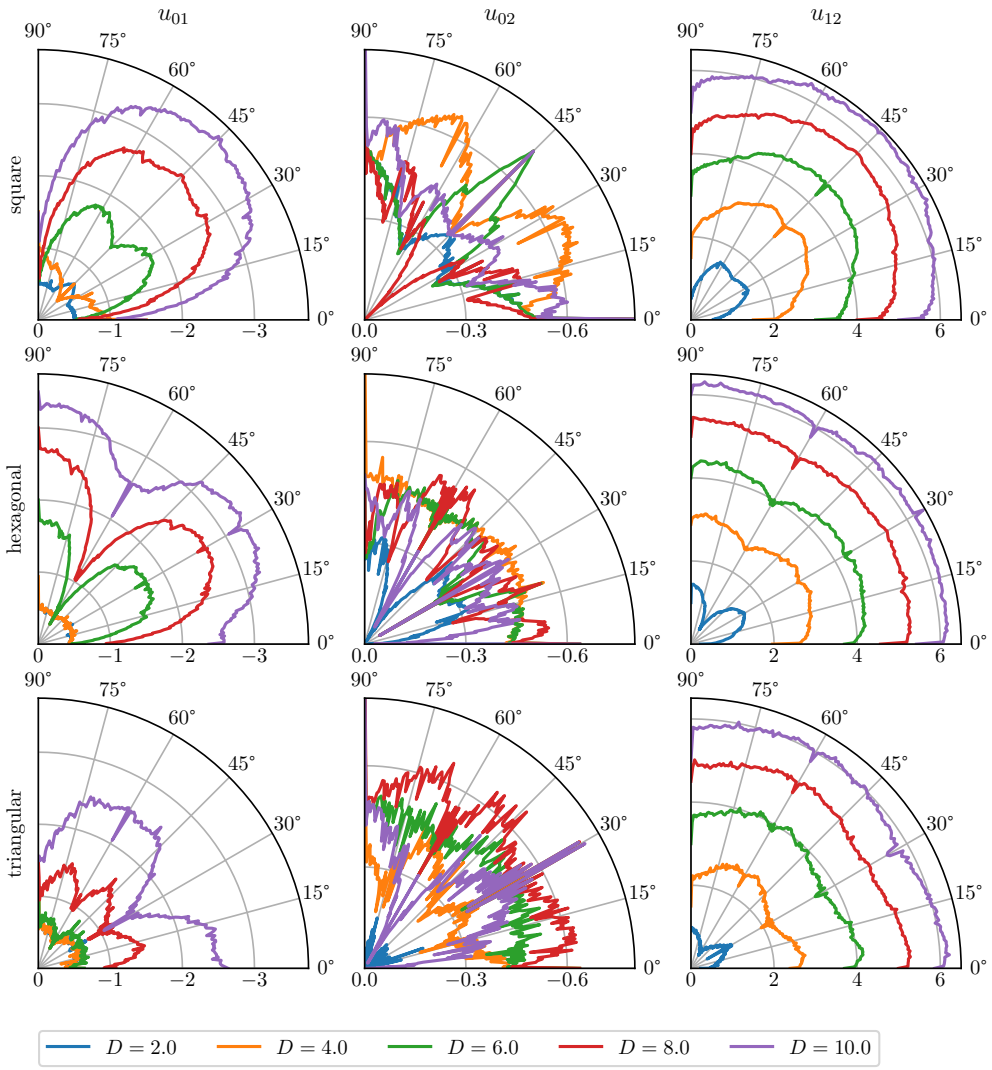


Figure 4.14: Front movement $d(1)$ as function of the front orientation for $\alpha = -10$, $\delta = 2$, $\gamma = 15$, $\delta = -5$ ($V(\tilde{u}_0) < V(\tilde{u}_2) \ll V(\tilde{u}_1)$), $\Delta x^2 = 1$ and different D. Its sign hints at the directions of front movement. For this case, the diffusion grids (from top to bottom: square, hexagonal, and triangular), as well as the evolved front solution (from left to right the: u_{01}, u_{02}, and u_{12}.) show distinct features.

The u_{01} front highlights the properties of the diffusion grid the best. The potential difference is large to enforce a fast-moving front, while the barrier between the solutions exhibits a significant resistance. As seen in Fig. 4.5, the square grid shows a periodicity of 90° with maximal front movement at $\phi = 45°$ and minimal movement at $\phi = 0°$.

The hexagonal and the triangular lattice show a periodicity of 60°, only being shifted by 30° with respect to each other. The difference between their form is that the triangular lattice produces more narrow and significant smaller petals. Further, the petals can be described as having a pronounced twinned structure. Interestingly, the curve shows no reason for the development of the distorted hexagonal pattern seen in Fig. 4.11.

Increasing D has here three effects on the *petals*: 1) they grow out as propagation speed in-

creases; 2) they expand around the extreme angles, as fewer orientations remain frozen; and 3) they change form after propagation failure is inhomogeneously overcome. This effect is seen at the change from $D = 2$ to $D = 4$ where the curves first change from rather frozen, to slow anisotropic propagation. For $D = 6$, the petals exhibit some unexpected changes. For the square lattice, the expected maxima of $\phi = 45°$ is rather a local minimum as the curve is bent inwards. Using the hexagonal lattice, propagation at the sticky angle $\phi = 60°$ jumps up significantly to another frozen state.

A focus may be put on the triangular lattice as its behavior is the most unique of the three lattices. Since the curve for $D = 4$ displays the maxima not at the expected positions, but in the same phase as the hexagonal lattice. Also, front movement is found to be larger in this case. Increasing D leads to a rather circular curve, propagation mostly increases around the expected maxima. As this trend follows for $D = 8$, the twinned peaks around $\phi = 0°$ and $\phi = 60°$ start to grow out using the front movement for $D = 6$ as a baseline.

More evidence that the triangular lattice is the most sticky of the three lattices can be found in the curves for $D = 10$. For u_{12} and u_{02}, there is no meaningful difference in front movement between the three grids, but the u_{01} front propagates on the triangular lattice considerably less far. This indicates that the 'resistance' that the surface offers is still not completely overcome.

4.6 Conclusion

Reaction-diffusion fronts can get pinned to a discrete spatial lattice instead of traveling without deformation as in continuous space. This phenomenon has been studied extensively but using simple bistable models, where there is a single family of fronts.

Here we wanted to understand how the properties of three stable states and three families of fronts, are modified by propagation failure, and how the details of a 2D regular lattice enter into the description.

Using a single species model and a quintic polynomial, we found that the three families of fronts are not pinned in the same way: the widths of the pinning regions differ; the shapes of the snaking branches differ; and the snaking branch of the front between the extreme roots being the most complex. The regions of propagation failure for the three fronts create horns around the equipotential curves, that intersect at a rectangular region where all three fronts are frozen. Again, it was the region associated to the extreme front the one that showed the most structure.

We also found that regular 2D lattices with sparser connectivity lead more often to propagation failure. Although all lattices do have 'weak' orientations that hinder propagation, the triangular lattice was the weakest in the sense that the PF regions were larger.

All these observations are relevant for the propagation of 2D patterns. Complex patterns can be stabilized within the pinning region, while relaxation of certain orientations can lead to patterns reflecting the geometry of the lattice. Lattices of higher symmetry will subsequently lead to local solutions of similar symmetry. Increasing the diffusion constant, the propagation along certain orientations is enabled. The influence of the front orientation was analyzed in detail. Predicting the front orientations at which propagation will be maximized resp. minimized is reasonably possible using a 1D simplification, considering the surface lattice and given orientation. Numerical

evolution of fronts with different orientations shows a flower-like behavior of propagation with a lattice specific periodicity. These observed front velocities can be explained with the intrinsic energy barrier of diffusion in our model and surface resistance that depends on front orientation. For sufficiently large $D/\Delta x^2$, propagation failure can be partly overcome and pattern matching the surface symmetry can be evolved.

The present article considered a single-species RD model. But there are front phenomena that can not be captured with a single species model. For instance the coexistence of front solutions between the same two states. It would be an interesting topic of future research the extension of the present framework to multi-species RD models and the effect of discreteness.

Supplementary Material

Supplementary material is online and contains the python source codes for numerical simulations.

Acknowledgments

J.C. thanks FONDECYT (Chile) for financial support through Grant No. 1210297 and 1200357. K.R. and J.C. are grateful for the financial support of the Erasmus+ mobility programme of the European Union.

Numerical integration has been done using *Python* relying on the packages Numpy [40] and Scipy [41], utilizing Numba [42] to accelerate computation and Matplotlib [43] to generate figures.

Data Availability Statement

Most data that support the findings of this study are available within the article and its supplementary material. Additional information can be requested by the authors.

Bibliography

[1] Engel T, Ertl G *A Molecular Beam Investigation of the Catalytic Oxidation of CO on Pd(111)*. (1978) *J. Chem. Phys.* **69**, 1267–1281. doi: 10.1063/1.436666.

[2] Engel T, Ertl G *Elementary Steps in the Catalytic Oxidation of Carbon Monoxide on Platinum Metals*. (1979) *Adv. Catal.* **28**, 1–78. doi: 10.1016/S0360-0564(08)60133-9.

[3] Ertl G, Norton PR, Rüstig J *Kinetic Oscillations in the Platinum-catalyzed Oxidation of CO*. (1982) *Phys. Rev. Lett.* **49**, 177–180. doi: 10.1103/PhysRevLett.49.177.

[4] Krischer K, Eiswirth M, Ertl G *Bifurcation Analysis of an Oscillating Surface Reaction Model*. (1991) *Surf. Sci.* **251–252**, 900–904. doi: 10.1016/0039-6028(91)91121-D.

[5] Falcke M, Bär M, Engel H, Eiswirth RM *Traveling Waves in the CO Oxidation on Pt(110): Theory*. (1992) *J. Chem. Phys.* **97**, 4555–4563. doi: 10.1063/1.463900.

[6] Nettesheim S, von Oertzen A, Rotermund HH, Ertl G *Reaction Diffusion Patterns in the Catalytic CO-oxidation on Pt(110): Front Propagation and Spiral Waves*. (1993) *J. Chem. Phys.* **98**, 9977–9985. doi: 10.1063/1.464323.

[7] Méndez V, Fedotov S, Horsthemke W *Reaction–Transport Systems*. (2010) Springer Series in Synergetics. Berlin, Heidelberg: Springer Berlin Heidelberg.

[8] Clerc MG, Elías RG, Rojas RG *Continuous Description of Lattice Discreteness Effects in Front Propagation*. (2011) *Phil. Trans. R. Soc. A.* **369**, 412–424. doi: 10.1098/rsta.2010.0255.

[9] Cisternas J, Rohe K, Wehner S *Reaction–Diffusion Fronts and the Butterfly Set*. (2020) *Chaos* **30**, 113138. doi: 10.1063/5.0022298.

[10] Stegemerten F, Gurevich S, Thiele U *Bifurcations of Front Motion in Passive and Active Allen-Cahn-type Equations*. (2020) *Chaos* **30**, 053136. doi: 10.1063/5.0003271.

[11] Ambrosio B, Aziz-Alaoui M *Synchronization and Control of Coupled Reaction–Diffusion Systems of the FitzHugh–Nagumo Type*. (2012) *Comput. Math* **64**, 934–943. doi: 10/f39nh5.

[12] Khatibi S, Cassol GO, Dubljevic S *Linear Model Predictive Control for a Cascade ODE-PDE System*. (2020) In *2020 American Control Conference (ACC)* pp. 4521–4526 Denver, CO, USA. IEEE.

[13] Stamatakis M, Chen Y, Vlachos DG *First-Principles-Based Kinetic Monte Carlo Simulation of the Structure Sensitivity of the Water - Gas Shift Reaction on Platinum Surfaces*. (2011) *J. Phys. Chem. C* **115**, 24750–24762. doi: 10.1021/jp2071869.

[14] Eiswirth M, Möller P, Wetzl K, Imbihl R, Ertl G *Mechanisms of Spatial Self-Organization in Isothermal Kinetic Oscillations during the Catalytic CO Oxidation on Pt Single Crystal Surfaces*. (1989) *J. Chem. Phys.* **90**, 510–521. doi: 10.1063/1.456501.

[15] Wehner S, Hoffmann P, Schmeisser D, Brand HR, Küppers J *The Consequences of Anisotropic Diffusion and Noise: PEEM at the CO Oxidation Reaction on Stepped Ir(111) Surfaces*. (2006) *Chem. Phys. Lett.* **423**, 39–44. doi: 10.1016/j.cplett.2006.03.010.

[16] Wehner S, Hoffmann P, Schmeisser D, Brand HR, Küppers J *Influence of the Substrate on the Pattern Formation of a Surface Reaction*. (2007) *AIP Conf. Proc.* **913**, 121–126. doi: 10.1063/1.2746735.

[17] Peyrard M, Kruskal MD *Kink Dynamics in the Highly Discrete Sine-Gordon System*. (1984) *Physica D: Nonlinear Phenomena* **14**, 88–102. doi: 10.1016/0167-2789(84)90006-X.

[18] Keener JP *Propagation and Its Failure in Coupled Systems of Discrete Excitable Cells*. (1987) *SIAM J. Appl. Math.* **47**, 556–572. doi: 10.1137/0147038.

[19] Laplante JP, Erneux T *Propagation Failure in Arrays of Coupled Bistable Chemical Reactors*. (1992) *J. Phys. Chem.* **96**, 4931–4934. doi: 10.1021/j100191a038.

[20] Erneux T, Nicolis G *Propagating Waves in Discrete Bistable Reaction-Diffusion Systems*. (1993) *Physica D* **67**, 237–244. doi: 10.1016/0167-2789(93)90208-I.

[21] Booth V, Erneux T, Laplante JP *Experimental and Numerical Study of Weakly Coupled Bistable Chemical Reactors*. (1994) *J. Phys. Chem.* **98**, 6537–6540. doi: 10.1021/j100077a019.

[22] Woods P, Champneys A *Heteroclinic Tangles and Homoclinic Snaking in the Unfolding of a Degenerate Reversible Hamiltonian–Hopf Bifurcation*. (1999) *Physica D: Nonlinear Phenomena* **129**, 147–170. doi: 10.1016/S0167-2789(98)00309-1.

[23] Taylor C, Dawes JH *Snaking and Isolas of Localised States in Bistable Discrete Lattices*. (2010) *Physics Letters A* **375**, 14–22. doi: 10.1016/j.physleta.2010.10.010.

[24] Seidel TG, Perego AM, Javaloyes J, Gurevich SV *Discrete Light Bullets in Passively Mode-Locked Semiconductor Lasers*. (2020) *Chaos* **30**, 063102. doi: 10.1063/5.0002989.

[25] Humphries AR, Wilds RP *Traveling Wave Solutions of Lattice Differential Equations through an Embedding Approach*. (2006) Journal Article.

[26] Qi H, Li ZC, Wang SM, Wu L, Xu F, Liu ZL, Li X, Wang JZ *Tristability in Mitochondrial Permeability Transition Pore Opening*. (2021) *Chaos* **31**, 123108. doi: 10.1063/5.0065400.

[27] Huang B, Xia Y, Liu F, Wang W *Realization of Tristability in a Multiplicatively Coupled Dual-Loop Genetic Network*. (2016) *Sci. Rep.* **6**, 28096. doi: 10.1038/srep28096.

[28] Mot LD, Gonze D, Bessonnard S, Chazaud C, Goldbeter A, Dupont G *Cell Fate Specification Based on Tristability in the Inner Cell Mass of Mouse Blastocysts*. (2016) *Biophysical Journal* **110**, 710–722. doi: 10.1016/j.bpj.2015.12.020.

[29] Poston T, Stewart I *The First Seven Catastrophe Geometries*. (1996) Mineola, New York: Dover Publications.

[30] Pazó D, Deza RR, Pérez-Muñuzuri V *Parity-Breaking Front Bifurcation in Bistable Media: Link between Discrete and Continuous Versions*. (2005) *Physics Letters A* **340**, 132–138. doi: 10.1016/j.physleta.2005.03.026.

[31] Van Vleck ES, Mallet-Paret J, Cahn JW *Traveling Wave Solutions for Systems of ODEs on a Two-Dimensional Spatial Lattice*. (1998) *SIAM J. Appl. Math.* **59**, 455–493. doi: 10.1137/S0036139996312703.

[32] Hupkes HJ, Morelli L *Travelling Corners for Spatially Discrete Reaction-Diffusion Systems*. (2020) *Commun. Pure Appl. Anal.* **19**, 1609–1667. doi: 10.3934/cpaa.2020058.

[33] Patra M, Karttunen M *Stencils with Isotropic Discretization Error for Differential Operators*. (2006) *Numer. Methods Partial Differ. Equ.* **22**, 936–953. doi: 10/c3dk65.

[34] Bechhoefer J, Löwen H, Tuckerman LS *Dynamical Mechanism for the Formation of Metastable Phases*. (1991) *Phys. Rev. Lett.* **67**, 1266–1269. doi: 10.1103/PhysRevLett.67.1266.

[35] Rubinstein J, Sternberg P, Keller JB *Front Interaction and Nonhomogeneous Equilibria for Tristable Reaction-Diffusion Equations*. (1993) *SIAM J. Appl. Math.* **53**, 1669–1685. doi: 10.1137/0153077.

[36] Hupkes HJ, Pelinovsky D, Sandstede B *Propagation Failure in the Discrete Nagumo Equation*. (2011) *Proc. Amer. Math. Soc.* **139**, 3537–3537. doi: 10.1090/S0002-9939-2011-10757-3.

[37] Doedel E *AUTO-07P: Continuation and Bifurcation Software for Ordinary Differential Equations*. (2007) Technical report Concordia University Montreal, Canada.

[38] Balmforth NJ, Craster RV, Kevrekidis PG *Being Stable and Discrete*. (2000) *Physica D: Nonlinear Phenomena* **135**, 212–232. doi: 10.1016/S0167-2789(99)00137-2.

[39] Martsenyuk V, Veselska O *On Nonlinear Reaction-Diffusion Model with Time Delay on Hexagonal Lattice*. (2019) *Symmetry* **11**, 758. doi: 10.3390/sym11060758.

[40] Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, del Río JF, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C, Oliphant TE *Array Programming with NumPy*. (2020) *Nature* **585**, 357–362. doi: 10.1038/s41586-020-2649-2.

[41] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat İ, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, SciPy 1.0 Contributors, Vijaykumar A, Bardelli AP, Rothberg A, Hilboll A, Kloeckner A, Scopatz A, Lee A, Rokem A, Woods CN, Fulton C, Masson C, Häggström C, Fitzgerald C, Nicholson DA, Hagen DR, Pasechnik DV, Olivetti E, Martin E, Wieser E, Silva F, Lenders F, Wilhelm F, Young G, Price GA, Ingold GL, Allen GE, Lee GR, Audren H, Probst I, Dietrich JP, Silterra J, Webber JT, Slavič J, Nothman J, Buchner J, Kulick J, Schönberger JL, de Miranda Cardoso JV, Reimer J, Harrington J, Rodríguez JLC, Nunez-Iglesias J, Kuczynski J, Tritz K, Thoma M, Newville M, Kümmerer M, Bolingbroke M, Tartre M, Pak M, Smith NJ, Nowaczyk N, Shebanov N, Pavlyk O, Brodtkorb PA, Lee P, McGibbon RT, Feldbauer R, Lewis S, Tygier S, Sievert S, Vigna S, Peterson S, More S, Pudlik T, Oshima T, Pingel TJ, Robitaille TP, Spura T, Jones TR, Cera T, Leslie T, Zito T, Krauss T, Upadhyay U, Halchenko YO, Vázquez-Baeza Y *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*. (2020) *Nat. Methods* **17**, 261–272. doi: 10.1038/s41592-019-0686-2.

[42] Lam SK, Pitrou A, Seibert S *Numba: A LLVM-based Python JIT Compiler*. (2015) In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC - LLVM '15* pp. 1–6 Austin, Texas. ACM Press.

[43] Hunter JD *Matplotlib: A 2D Graphics Environment*. (2007) *Comput. Sci. Eng.* **9**, 90–95. doi: 10.1109/MCSE.2007.55.

5

Final Conclusion

The shown studies were motivated by the dynamic phenomena observed in elementary catalytic surface reactions, particularly the CO oxidation on Ir(111), which can exhibit two stable kinetic states. A particular focus was the introduction of tristable systems and the study of interactions between these stable states. Therefore, two continuous and one discrete reaction-diffusion models were developed and studied in detail using bifurcation theory and numerical simulations.

First, a physically sound model of the competing catalytic oxidation of carbon monoxide and hydrogen was established. The model extends the well-established framework from previous work on bistable CO oxidation to Ir(111) and draws on the literature on the water formation reaction. The differential equation system was analyzed within the framework of bifurcation theory and the possibility of a third stable kinetic regime was revealed. Besides the previously known UR (upper rate branch, high oxygen coverage, high formation rates) and LR (lower rate branch, high carbon monoxide coverage, low formation rates). The newly found state denoted as VLR (very low rate) is dominantly covered by hydrogen as exhibits only limited conversion.

A codimension-2 diagram (using the molar fractions of the feed gas) reveals a higher-order bifurcation in the shape of a ‘swallowtail’; formed by the saddle-node bifurcations dividing the parameter space into different regimes and exhibiting ‘cusps’. Increasing the temperature leads to a subsequent unfolding and thus decreases the complexity of the system.

A series of possible experiments were designed to test the experimental accessibility and distinguishability (or the lack) of the stable regimes and carried out via numerical simulation. A relaxation experiment showing trajectories from different initial surface coverages enables a first assessment of the decay velocity; hinting that the LR might be obtained by the largest set of initial conditions, but might prove difficult to reach in reasonable time frames. Quasistatic scanning illustrates the existence of all three states within the tristable regime and the transition between

states once becoming unstable. This was further exemplified by choosing a more complex path from tristable to monostable regime.

Lastly, a first test concerning diffusion and state dominance was conducted. Examples show UR dominating over LR and VLR in 1D. A 2D time-evolution reveals a front splitting behavior, as the LR separates the interphase between UR and VLR. This phenomenon illustrates the importance of diffusion in understanding the interactions between these states.

Due to the significance of these first results, greater emphasis was given to the reaction-diffusion fronts. However, it became apparent that the established model had some constraints that could not be easily resolved. Firstly, the model relying on four surface species renders its treatment highly computationally demanding, especially in 2D. Secondly, since no experimental data is available to verify or refine the model and its reaction constants, it is subject to a high degree of uncertainty. The inclusion of additional assumptions regarding its diffusion mechanism renders the model a highly speculative special case.

The solution to both of these challenges was achieved by utilizing a simple monospecies reaction-diffusion equation with a quintic polynomial as reaction term. This mock model has up to three stable solutions (u_0, u_1, u_2) and exhibits the 'butterfly bifurcation' which is related to the previously observed swallowtail bifurcation.

To identify regions with similar behaviors, an extensive exploration of the model in the four-dimensional parameter space was conducted. A term for the front velocity connecting two stable states showed its dependence on the sixth-order polynomial potential. Remarkably, only the potential difference between both solutions influences the velocity, not its shape. Thus, equipotential curves were identified as special features of the model, where the fronts connecting two solutions are inhibited from propagation as their velocity vanishes. These findings were validated by numerical simulations on a two-dimensional, finite disk using a triangulated mesh.

Under certain conditions, it was observed that u_1 invades into the u_{02}-front, similar to the dynamic observed for the physical-motivated model. This is called front-splitting instability, due to the front solution u_{02} becoming unstable itself. Utilizing simple assumptions, a good estimate for the boundaries of the front splitting region was given. This was validated using time evolutions, illustrated qualitatively and quantitatively.

Finally, the numerical simulations in 2D posed a question about the relationship between mesh resolution and the front velocity fronts. Especially for low diffusivity and lower resolution, front propagation was not noticeable over the observed time scales. Furthermore, the question arose whether an arbitrary triangulated mesh was the best surface discretization regarding surface reaction, where the substrate is often a periodic crystal. Adsorbed molecules are known for having preferred surface positions on crystals which restrict the direction of diffusion and lead to the formation of regular overlayers.

Therefore, the established mock model was modified from continuous to discrete space, utilizing a simple domain in 1D and three different lattices in 2D (square, hexagonal, triangular). For high resolutions and thus strong interaction between individual surface nodes (expressed through $D/\Delta x^2$), the discrete model behaves close to the continuous model. For low diffusivities or large

distances between coupling nodes, fronts can become pinned, if the parameters are within range of the equipotential lines. This phenomenon is called propagation failure and its extent in parameter space was explored in 1D and for remarkable 2D front orientations using a pseudo-2D approximation.

Front velocities become less easy to predict utilizing only the depths of the potential wells, with large disagreements near the pinning region. Besides the coupling strength, the size and shape of the pinning region depend on the lattice (hexagonal $<$ triangular), the front orientation relative to the lattice ($45° < 0°$ for the square lattice), and the front ($u_{12} < u_{01}$). Regarding the bifurcation diagram in the pinning region, both stable branches become connected by a snaking curve containing multiple branches of alternating stable and unstable front solutions.

These findings were complemented by numerical pattern evolution, confirming the observations concerning propagation failure and lattice dependence. An interesting observation was made concerning the influence of front orientation on the front velocities. Since some orientations are less prone to propagation failure, there exists a range in which some fronts can move while others cannot. This selective propagation leads to the evolution of patterns reflecting the lattice geometry. An attempt has been made to quantify the angular resistance towards propagation depending on front orientation. While there were some distortions due to boundary effects, the results appear plausible and in reasonable agreement with the pseudo-2D results.

These results expand our understanding of reaction-diffusion systems with three stable states in the context of surface reactions and illustrate the value of generic models as a substitution for more sophisticated ones. Thus, this work can serve as a basis to fertilize future research.

Given the theoretical nature of this study, experimental testing of the observed phenomena (front splitting, propagation failure, and the influence of discreteness) in surface reactions could be pursued using imaging techniques.

In addition, the uncertainty regarding the ternary surface reaction of hydrogen, carbon monoxide, and oxygen can be improved upon. To accomplish this, rigorous and well-controlled experiments are necessary. In particular, the mechanism of the high-temperature catalytic water formation is still controversial due to discrepancies between models and experiments.

A theoretical subject might be to consider two-species systems, which can include dynamic effects such as oscillations and the coexistence of multiple front solutions, which are not present in the single-species model.

A

Exact solutions of the example model

The equation system (1.22) can be solved using computer algebra systems, such as the Sympy package in Python. As first step, an expression for $\rho_A(\rho_B)$ can be derrived, utilizing the first equation:

$$\rho_A(\rho_B) = \frac{\phi_A s_A (1-\rho_B)}{\Phi_A s_A + k_1 + k_2 \rho_B} \tag{A.1}$$

Substituting (A.1) in the second equation and solving for ρ_B gives 4 solutions:

$$\rho_{B,1} = 1, \tag{A.2}$$

$$\rho_{B,2} = -\frac{1}{3}\left(C_0 + \frac{C_1}{C_0} + 3C_2\right), \tag{A.3}$$

$$\rho_{B,3} = \frac{1}{6}\left((1+\sqrt{3}i)C_0 + \frac{(1-\sqrt{3}i)C_1}{C_0} + 6C_2\right), \tag{A.4}$$

$$\rho_{B,4} = \frac{1}{6}\left((1-\sqrt{3}i)C_0 + \frac{(1+\sqrt{3}i)C_1}{C_0} + 6C_2\right), \tag{A.5}$$

utilizing the following substitutions:

$$C_0 = \frac{1}{2}\sqrt[3]{C_3 + \sqrt{C_3^2 - 64C_1^3}} \tag{A.6a}$$

$$C_1 = \frac{C_5^2}{4C_7^2 k_2^2} - \frac{3C_6^2 + 3C_6 k_1}{2C_7 k_2} - \frac{3k_1^2}{k_2^2} + \frac{6k_1}{k_2} \tag{A.6b}$$

$$C_2 = -\frac{C_6}{6C_7} - \frac{2k_1}{3k_2} + \frac{1}{3} \tag{A.6c}$$

$$C_3 = -\frac{9C_4C_5}{C_7^2k_2^3} + \frac{C_5^3}{C_7^3k_2^3} - \frac{108k_1^2}{k_2^2} \tag{A.6d}$$

$$C_4 = C_6^2k_2 + C_6k_1k_2 + 2C_7k_1^2 - 4C_7k_1k_2 \tag{A.6e}$$

$$C_5 = k_2C_6 + (4k_1 - 2k_2)C_7 \tag{A.6f}$$

$$C_6 = \Phi_A s_A \tag{A.6g}$$

$$C_7 = \Phi_B s_B \tag{A.6h}$$

The terms for $\rho_{A,i}$ can subsequently be computed through resubstituting $\rho_{B,i}$

B

Supplementary data for Chapter 2

The following sections contain the source codes for numerical simulation used in chapter 2.

ternary_basics.py

```
from math import *
import numpy as np

def initialize():
    T = 500                    #Temperature [K]
    nIr = 1.56e15              #active sites [cm-2 = 1 ML]
    Fm = 1.37e15/nIr           #maximal flow rate [ML s-1]
    R = 8.3145                   #gas constant [J mol-1 K-1]
    v = np.zeros((11))               #prefactors
    E = np.zeros((11))               #activation enrgies
    k = np.zeros((11))               #reaction constants
    v[0],E[0] = 1e13, 140000 #CO desorption [s-1],[J+1 mol-1]
    v[9],E[9] = 1e5, 40000   #CO+O reaction [ML-1 s-1], [J+1 mol-1]
    s_CO,s_O,s_H = 1,0.11,0.007 #sticking coefficients
    a,b,c = 1,2.0,3            #coverage correction CO
    v[1],E[1] = 1.5e-1, 18.6*4184.0
    v[3],E[3] = 1e16/nIr, 4.6*4184 #H+O reaction [ML-1 s-1], [J mol-1]
    v[5],E[5] = 9e15/nIr,2.5*4184.0
    v[7],E[7] = 2e12/nIr,75000.0  #OH+OH reaction [ML-1 s-1]
    v[2],E[2],k[2] = 0,0,0  #1e-6,56*4184 #O desorption [J mol-1]
    v[4],E[4],k[4] = 0,0,0 #1e2,5*4184 #HO lyse [s-1]
    v[6],E[6],k[6] = 0,0,0 #1e7,37*4184 #H2O lyse [J mol-1]
    v[8],E[8],k[8] = 0,0,0 #1e5,10*4184 #H2O+O reaction [J mol-1]
    v[10],E[10],k[10] = 0,0,0 # 0.69,10.2*4184 #H2O-Desorption [s-1]
```

```
    k[0] = v[0]*exp(-E[0]/(T*R))        #CO desorption [s-1]
    k[9] = v[9]*exp(-E[9]/(T*R))        #CO+O reaction [ML-1 s-1]
    k[1] = v[1]*exp(-E[1]/(T*R))        #H desorption [ML s-1]
    k[3] = v[3]*exp(-E[3]/(T*R))        #H+O reaction [ML s-1]
    k[5] = v[5]*exp(-E[5]/(T*R))        #H+OH reaction [ML s-1]
    k[7] = v[7]*exp(-E[7]/(T*R))        #OH+OH reaction [ML-1 s-1]
    k[8] = v[8]*exp(-E[8]/(T*R))        #H2O+O reaction [ML-1 s-1]
    return T,k,E,v,Fm,s_CO,s_O,s_H,a,b,c

def rhs(t,n,const): #defines function
    k,F_CO,F_O,F_H,s_CO,s_O,s_H,a,b,c = const
    n[abs(n) < tol] = 0.0
    n[n > 1] = 1
    occ = 1. - (n[0] + n[1] + n[2] + n[3])
    m = np.zeros_like(n)
    r = np.zeros((14))          #reaction rates
    r[0] = (F_CO*s_CO*occ**a)   #Adsorption CO
    r[1] = F_H*s_H*occ**b       #Adsorption H
    r[2] = (F_O*s_O*occ**c)     #Adsorption O
    r[3] = (k[0]*n[0])          #Desorption CO
    r[4] = (k[1]*n[1]**2)       #Desorption H
    r[6] = (k[3]*n[1]*n[2])     #Reaction H+O --> OH
    r[8] = (k[5]*n[1]*n[3])     #Reaction OH+H --> H2O
    r[10] = (k[7]*n[3]**2)      #Reaction OH+OH --> H2O + O
    r[12] = (k[9]*n[0]*n[2])    #Reaction CO+O --> CO2
    r[5] = 0#k[2]*n[1]**2       #Desorpton O
    r[7] = 0#k[4]*n[3]          #Reaction HO --> O+H
    r[9] = 0#k[6]*n[4]          #Reaction H2O --> OH+H
    r[11] = 0#k[8]*n[4]*n[2]    #Reaction H2O+O --> OH+OH
    r[13] = 0#k[10]*n[4]        #Desorpton H2O
    m[0] = r[0] - r[3] - r[12] #CO=A
    m[1] = 2*r[1] - 2*r[4] - r[6] + r[7] - r[8] + r[9] #H=B
    m[2] = 2*r[2] - 2*r[5] - r[6] + r[7] + r[10] -r[11] - r[12] #O=C
    m[3] = r[6] - r[7] - r[8] + r[9] - 2*r[10] + 2*r[11] #OH=D
    return m #returns derivatives of specific n

if __name__ == "__main__":
    pass
else:
    pass
tol = np.finfo(float).eps*10
```

border.py

```
from math import *
import numpy as np
from scipy.integrate import solve_ivp
import time
import os
from ternary_basics import *
```

```
T,k,E,v,Fm,s_CO,s_O,s_H,a,b,c = initialize()
tol = np.finfo(float).eps*10
nstep = 1                       #stepwidth for Y
cdic = {1:'LSODA',2:'Radau',3:'BDF'} #integration solvers
Y_H = 0                         #starting value
F_H = Y_H*Fm/1000
rtol,atol=1e-9,1e-6
for calc in [1]: #Test of Solvers
    calca = cdic[calc]
    for tmin in [1e3,1e4,1e5]: #Test of different calculation times
        start=time.strftime("%H:%M:%S")
        counter = 0 #global time
        n = np.zeros((4))         #CO, H, O, OH, H2O [ML]
        n[1]=0
        scan = [] #array for scan-run
        intervall = [0,1000] #intervall for CO step
        Y_O = intervall[0] #starting value
        while Y_O <= intervall[1]:
            max_step,tmax,t = tmin/6,tmin,0
            F_O = Y_O*Fm/1000
            Y_CO = 1000-Y_H-Y_O
            F_CO = Fm-F_O-F_H
            while t!=tmin:
                res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,s_CO
                    ,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=atol,
                    dense_output=True) #Solve diff
                border = res.y<0
                if border.any()==True:
                    A = np.arange(len(res.t))
                    for it in A:
                        if border[:,it].any()==True:
                            if it==0:
                                max_step = max_step/10
                            else:
                                max_step, t = (res.t[it]-res.t[it-1])*0.5,res.t
                                    [it-1]
                                tmax=t+(tmin-t)*0.75
                                n = res.sol(tmax)
                                n[abs(n) < tol] = 0.0
                                break
                else:
                    t = tmax
                    max_step,tmax = tmin/6,tmin
            n = res.sol(tmin)
            n[abs(n) < tol] = 0.0
            Y_O = Y_O+nstep

        Y_O = Y_O-nstep
        scan.append((counter,Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3]))#time, t_r Y_O,
            nCO, nH, nO, nOH
        Y_O = Y_O-nstep
```

```
        while Y_O >= intervall[0]:
            max_step,t = tmin/6,0
            F_O = Y_O*Fm/1000
            Y_CO = 1000-Y_H-Y_O
            F_CO = Fm-F_O-F_H
            while t!=tmin:
                res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,s_CO
                    ,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=atol,
                    dense_output=True) #Solve diff
                border = res.y<0
                if border.any()==True:
                    A = np.arange(len(res.t))
                    for it in A:
                        if border[:,it].any()==True:
                            if it==0:
                                max_step = max_step/10
                            else:
                                max_step, t = (res.t[it]-res.t[it-1])*0.5,res.t
                                    [it-1]
                                tmax=t+(tmin-t)*0.75
                                n = res.sol(tmax)
                                n[abs(n) < tol] = 0.0
                                break
                else:
                    t = tmax
                    max_step,tmax = tmin/6,tmin
            counter = counter+1
            n = res.sol(tmin)
            n[abs(n) < tol] = 0.0
            scan.append((counter,Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3]))#time, t_r
                Y_O, nCO, nH, nO, nOH
            Y_O = Y_O-nstep
        Y_O = intervall[0]+nstep #starting value
        while Y_O <= intervall[1]:
            max_step,t = tmin/6,0
            F_O = Y_O*Fm/1000
            Y_CO = 1000-Y_H-Y_O
            F_CO = Fm-F_O-F_H
            while t!=tmin:
                res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,s_CO
                    ,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=atol,
                    dense_output=True) #Solve diff
                border = res.y<0
                if border.any()==True:
                    A = np.arange(len(res.t))
                    for it in A:
                        if border[:,it].any()==True:
                            if it==0:
                                max_step = max_step/10
                            else:
                                max_step, t = (res.t[it]-res.t[it-1])*0.5,res.t
```

```
                                    [it-1]
                                tmax=t+(tmin-t)*0.75
                                n = res.sol(tmax)
                                n[abs(n) < tol] = 0.0
                                break
                    else:
                        t = tmax
                        max_step,tmax = tmin/6,tmin
                counter = counter+1
                n = res.sol(tmin)
                n[abs(n) < tol] = 0.0
                scan.append((counter,Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3]))#time, t_r
                    Y_O, nCO, nH, nO, nOH
                Y_O = Y_O+nstep
            np.savetxt('border_H'+str(Y_H)+'\\'+str("{:.1f}".format(log10(tmin)))+'
                lgs_'+calca+'.txt', (np.asarray(scan)), fmt="%.0f\t"*4 + "%.12e\t"+
                "%.2e\t"+"%.12e\t"+"%.2e", header='step\tY\-(A)\tY\-(B)\tY\-(C)\tn
                \-(A)\tn\-(B)\tn\-(C)\tn\-(D)', footer="This file was generated on
                "+time.strftime("%d.%m.%Y %H:%M:%S")+" by border_H0.py.\nT = "+str(
                T)+"K\nE = "+str(E)+"\nv = "+str(v)+"\nF_tot = "+str(Fm))
            print('finished calculation_tmin = ', tmin, 'calca = ',calca,'starting
                time = ',start, 'ending time = ', time.strftime("%H:%M:%S"))

Y_O = 0                                 #starting value
F_O = Y_O*Fm/1000
for calc in [2]: #Radau
    calca = cdic[calc]
    for tmin in [1e0,1e1,1e2,1e3,1e4,1e5,1e6,1e7,1e8,1e9]: #Test of different
        calculation times
        start=time.strftime("%H:%M:%S")
        counter = 0 #global time
        n = np.zeros((4))          #CO, H, O, OH, H2O [ML]
        n[0]=9.737279601731e-01
        scan = [] #array for scan-run
        intervall = [0,1000-Y_O] #intervall for CO step
        Y_H = intervall[0] #starting value
        Y_CO = 1000-Y_H-Y_O
        Y_H = intervall[0]+nstep #starting value
        while Y_H <= intervall[1]:
            max_step,tmax,t = tmin/6,tmin,0
            F_H = Y_H*Fm/1000
            Y_CO = 1000-Y_H-Y_O
            F_CO = Fm-F_O-F_H
            while t!=tmin:
                res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,s_CO
                    ,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=atol,
                    dense_output=True) #Solve diff
                border = res.y<0
                if border.any()==True:
                    A = np.arange(len(res.t))
                    for it in A:
```

```
                    if border[:,it].any()==True:
                        if it==0:
                            max_step = max_step/10
                        else:
                            max_step, t = (res.t[it]-res.t[it-1])*0.5,res.t
                                [it-1]
                            tmax=t+(tmin-t)*0.75
                            n = res.sol(tmax)
                            n[abs(n) < tol] = 0.0
                            break
            else:
                t = tmax
                max_step,tmax = tmin/6,tmin
        counter = counter+1
        n = res.sol(tmin)
        n[abs(n) < tol] = 0.0
        scan.append((counter,Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3]))#time, t_r
            Y_O, nCO, nH, nO, nOH
        Y_H = Y_H+nstep

    scan.append((counter,Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3]))#time, t_r Y_O,
        nCO, nH, nO, nOH
    Y_H = Y_H-nstep
    while Y_H >= intervall[0]:
        max_step,tmax,t = tmin/6,tmin,0
        F_H = Y_H*Fm/1000
        Y_CO = 1000-Y_H-Y_O
        F_CO = Fm-F_O-F_H
        while t!=tmin:
            res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,s_CO
                ,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=atol,
                dense_output=True) #Solve diff
            border = res.y<0
            if border.any()==True:
                A = np.arange(len(res.t))
                for it in A:
                    if border[:,it].any()==True:
                        if it==0:
                            max_step = max_step/10
                        else:
                            max_step, t = (res.t[it]-res.t[it-1])*0.5,res.t
                                [it-1]
                            tmax=t+(tmin-t)*0.75
                            n = res.sol(tmax)
                            n[abs(n) < tol] = 0.0
                            break
            else:
                t = tmax
                max_step,tmax = tmin/6,tmin
        counter = counter+1
        n = res.sol(tmin)
```

```
            n[abs(n) < tol] = 0.0
            scan.append((counter,Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3]))#time, t_r
                Y_O, nCO, nH, nO, nOH
            Y_H = Y_H-nstep
        np.savetxt('border_O'+str(Y_O)+'\\'+str("{:.1f}".format(log10(tmin)))+'
            lgs_'+calca+'.txt', (np.asarray(scan)), fmt="%.0f\t"*4 + "%.12e\t"
            *3+"%.12e", header='step\tY\-(A)\tY\-(B)\tY\-(C)\tn\-(A)\tn\-(B)\tn
            \-(C)\tn\-(D)', footer="This file was generated on "+time.strftime(
            "%d.%m.%Y %H:%M:%S")+" by border_O25.py.\nT = "+str(T)+"K\nE = "+
            str(E)+"\nv = "+str(v)+"\nF_tot = "+str(Fm))
        print('finished calculation_tmin = ', tmin, 'calca = ',calca,'starting
            time = ',start, 'ending time = ', time.strftime("%H:%M:%S"))

Y_CO = 0#15                                  #starting value
F_CO = Y_CO*Fm/1000
for calc in [2]: #Radau
    calca = cdic[calc]
    for tmin in [1e10]: #Test of different calculation times
        start=time.strftime("%H:%M:%S")
        counter = 0 #global time
        n = np.zeros((4))          #CO, H, O, OH, H2O [ML]
        n[0]=9.949272479787e-01
        scan = [] #array for scan-run
        intervall = [0,1000-Y_CO] #intervall for CO step
        Y_H = intervall[0] #starting value
        Y_O = 1000-Y_H-Y_CO

        Y_H = intervall[0]+nstep #starting value
        while Y_H <= intervall[1]:
            max_step,tmax,t = tmin/6,tmin,0
            F_H = Y_H*Fm/1000
            Y_O = 1000-Y_H-Y_CO
            F_O = Fm-F_CO-F_H
            while t!=tmin:
                res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,s_CO
                    ,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=atol,
                    dense_output=True) #Solve diff
                border = res.y<0
                if border.any()==True:
                    A = np.arange(len(res.t))
                    for it in A:
                        if border[:,it].any()==True:
                            if it==0:
                                max_step = max_step/10
                                tmax=t+(tmin-t)/5
                            else:
                                t=res.t[it-1]
                                max_step,tmax = ((tmin-t)*0.75)/6,t+(tmin-t)
                                    *0.75
                                n = res.sol(tmax)
                                n[abs(n) < tol] = 0.0
```

```
                        break
            else:
                t = tmax
                max_step,tmax = tmin/6,tmin
        counter = counter+1
        n = res.sol(tmin)
        n[abs(n) < tol] = 0.0
        scan.append((counter,Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3]))#time, t_r
            Y_O, nCO, nH, nO, nOH
        Y_H = Y_H+nstep
    scan.append((counter,Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3]))#time, t_r Y_O,
        nCO, nH, nO, nOH
    Y_H = Y_H-nstep
    while Y_H >= intervall[0]:
        max_step,tmax,t = tmin/6,tmin,0
        F_H = Y_H*Fm/1000
        Y_O = 1000-Y_H-Y_CO
        F_O = Fm-F_CO-F_H
        while t!=tmin:
            res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,s_CO
                ,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=atol,
                dense_output=True) #Solve diff
            border = res.y<0
            if border.any()==True:
                A = np.arange(len(res.t))
                for it in A:
                    if border[:,it].any()==True:
                        if it==0:
                            max_step = max_step/10
                            tmax=t+(tmin-t)/5
                        else:
                            t=res.t[it-1]
                            max_step,tmax = ((tmin-t)*0.75)/6,t+(tmin-t)
                                *0.75
                            n = res.sol(tmax)
                            n[abs(n) < tol] = 0.0
                            break
            else:
                t = tmax
                max_step,tmax = tmin/6,tmin
        counter = counter+1
        n = res.sol(tmin)
        n[abs(n) < tol] = 0.0
        scan.append((counter,Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3]))#time, t_r
            Y_O, nCO, nH, nO, nOH
        Y_H = Y_H-nstep
    np.savetxt('border_CO'+str(Y_CO)+'\\'+str("{:.1f}".format(log10(tmin)))
        +'lgs_'+calca+'.txt', (np.asarray(scan)), fmt="%.0f\t"*4 + "%.12e\t
        "*3+"%.12e", header='step\tY\-(A)\tY\-(B)\tY\-(C)\tn\-(A)\tn\-(B)\
        tn\-(C)\tn\-(D)', footer="This file was generated on "+time.
        strftime("%d.%m.%Y %H:%M:%S")+" by border_O25.py.\nT = "+str(T)+"K\
```

```
            nE = "+str(E)+"\nv = "+str(v)+"\nF_tot = "+str(Fm))
        print('finished calculation_tmin = ', tmin, 'calca = ',calca,'starting
            time = ',start, 'ending time = ', time.strftime("%H:%M:%S"))
```

CO-filling.py

```
from math import *
import numpy as np
from scipy.integrate import solve_ivp
import time
import os
from ternary_basics import *
T,k,E,v,Fm,s_CO,s_O,s_H,a,b,c = initialize()
tol = np.finfo(float).eps*10 #cutoff tolerance
rtol,atol=1e-9,1e-6
cdic = {1:'LSODA',2:'Radau',3:'BDF'} #integration solvers
calc,nstep = 2,1
calca = cdic[calc]
O = np.loadtxt('border_CO15\\9.0lgs_Radau.txt')
header = 'Y\-(A)\tY\-(B)\tY\-(C)\tY\-(A)\ty\tn\-(A)\tn\-(B)\tn\-(C)\tn\-(D)'

i = 1
bi_1,bi_2,mono = [],[],[]
while O[i-1][0]< O[-i][0]:
    summe=0
    for j in [4,5,6,7]:
        summe = abs(O[i-1][j]-O[-i][j])+summe
    if summe>0.0001:
        bi_1.append((O[-i][0],O[-i][1],O[-i][2],O[-i][3],O[i-1][4],O[i-1][5],O[
            i-1][6],O[i-1][7]))
        bi_2.append((O[-i][0],O[-i][1],O[-i][2],O[-i][3],O[-i][4],O[-i][5],O[-i
            ][6],O[-i][7]))
    else:
        mono.append((O[-i][0],O[-i][1],O[-i][2],O[-i][3],O[-i][4],O[-i][5],O[i
            ][6],O[i][7]))
    i = i+1
listdict = {'bi_1':bi_1,'bi_2':bi_2,'mono':mono}
for liste in ['bi_1','mono','bi_2']:
    for tmin in [1e6]:#1e8
        O = np.asarray(listdict[liste])
        start=time.strftime("%H:%M:%S")
        for rows in O:
            n = np.zeros((4))
            Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3],n[4] = *rows
            scan = [(Y_CO,Y_H,Y_O,(2*Y_O+Y_H)/2000,(sqrt(3)*Y_H)/2000,n[0],n
                [1],n[2],n[3])]
            F_O = Y_O*Fm/1000
            while Y_H > 0:
                Y_CO,Y_H = Y_CO+nstep,Y_H-nstep
                F_CO,F_H = Y_CO*Fm/1000,Y_H*Fm/1000
                max_step,tmax,t=tmin/6,tmin,0
```

```
        while t!=tmin:
            res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,
                s_CO,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=
                atol,dense_output=True)
            border = res.y<0
            if border.any()==True:
                A = np.arange(len(res.t))
                for it in A:
                    if border[:,it].any()==True:
                        if it==0:
                            max_step,tmax = max_step/10,t+(tmax-t)/5
                        else:
                            t=res.t[it-1]
                            tmax, max_step = t+(tmin-t)*0.75, ((tmin-t)
                                *0.75)/6
                            n = res.sol(tmax)
                            n[abs(n) < tol] = 0.0
                            break
            else:
                t = tmax
                tmax,max_step=tmin,tmin/6
        n = res.sol(tmin)
        n[abs(n) < tol] = 0.0
        scan.append((Y_CO,Y_H,Y_O,(2*Y_O+Y_H)/2000,(sqrt(3)*Y_H)/2000,n
            [0],n[1],n[2],n[3]))
        name = 'filling_CO15\\O_const\\'+str("{:.1f}".format(log10(tmin
            )))+'lgs\\'+str(int(rows[1]))+'_'+str(int(rows[2]))+'_'+str
            (int(rows[3]))
    np.savetxt(name+'-'+liste+'.txt', (np.asarray(scan)), fmt="%.0f\t"
        *3 +"%.8e\t"*2+ "%.12e\t"*4, header=header,footer="This file
        was generated on "+time.strftime("%d.%m.%Y %H:%M:%S")+" by SPH0
        .py.\nT = "+str(T)+"K\nE = "+str(E)+"\nv = "+str(v)+"\nF_tot =
        "+str(Fm))
    n = np.zeros((4))
    Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3],n[4] = *rows
    scan = [(Y_CO,Y_H,Y_O,(2*Y_O+Y_H)/2000,(sqrt(3)*Y_H)/2000,n[0],n
        [1],n[2],n[3])]
    F_H = Y_H*Fm/1000
    while Y_O > 0:
        Y_CO,Y_O = Y_CO+nstep,Y_O-nstep
        F_CO,F_O = Y_CO*Fm/1000,Y_O*Fm/1000
        max_step,tmax,t=tmin/6,tmin,0
        while t!=tmin:
            res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,
                s_CO,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=
                atol,dense_output=True)
            border = res.y<0
            if border.any()==True:
                A = np.arange(len(res.t))
                for it in A:
                    if border[:,it].any()==True:
```

```
                                if it==0:
                                    max_step,tmax = max_step/10,t+(tmax-t)/5
                                else:
                                    t=res.t[it-1]
                                    tmax, max_step = t+(tmin-t)*0.75, ((tmin-t)
                                        *0.75)/6
                                    n = res.sol(tmax)
                                    n[abs(n) < tol] = 0.0
                                    break
                            else:
                                t = tmax
                                tmax,max_step=tmin,tmin/6
                        n = res.sol(tmin)
                        n[abs(n) < tol] = 0.0
                        scan.append((Y_CO,Y_H,Y_O,(2*Y_O+Y_H)/2000,(sqrt(3)*Y_H)/2000,n
                            [0],n[1],n[2],n[3]))
                        name = 'filling_CO15\\H_const\\'+str("{:.1f}".format(log10(tmin
                            )))+'lgs\\'+str(int(rows[1]))+'_'+str(int(rows[2]))+'_'+str
                            (int(rows[3]))
                    np.savetxt(name+'-'+liste+'.txt', (np.asarray(scan)), fmt="%.0f\t"
                        *3 +"%.8e\t"*2+ "%.12e\t"*4, header=header,footer="This file
                        was generated on "+time.strftime("%d.%m.%Y %H:%M:%S")+" by SPH0
                        .py.\nT = "+str(T)+"K\nE = "+str(E)+"\nv = "+str(v)+"\nF_tot =
                        "+str(Fm))
                print('finished calculation_tmin = ', tmin, 'Liste = ',liste,'starting
                    time = ',start, 'ending time = ', time.strftime("%H:%M:%S"))
```

H-filling.py

```
#author: Kevin Rohe
from math import *
import numpy as np
from scipy.integrate import solve_ivp
import time
import os
from ternary_basics import *
T,k,E,v,Fm,s_CO,s_O,s_H,a,b,c = initialize()
tol = np.finfo(float).eps*10 #cutoff tolerance
rtol,atol=1e-9,1e-6
cdic = {1:'LSODA',2:'Radau',3:'BDF'} #integration solvers
calc,nstep = 2,1
calca = cdic[calc]
CO = np.loadtxt('border_H0\\5.0lgs_Radau.txt')
header = 'Y\-(A)\tY\-(B)\tY\-(C)\tY\-(A)\ty\tn\-(A)\tn\-(B)\tn\-(C)\tn\-(D)'

i = 1
bi_1,bi_2,mono = [],[],[]

while CO[i-1][0]< CO[-i][0]:
    summe=0
    for j in [4,5,6,7]:
```

```
        summe = abs(CO[i-1][j]-CO[-i][j])+summe
    if summe>0.0001:
        bi_1.append((CO[-i][0],CO[-i][1],CO[-i][2],CO[-i][3],CO[i-1][4],CO[i
            -1][5],CO[i-1][6],CO[i-1][7]))
        bi_2.append((CO[-i][0],CO[-i][1],CO[-i][2],CO[-i][3],CO[-i][4],CO[-i
            ][5],CO[-i][6],CO[-i][7]))
    else:
        mono.append((CO[-i][0],CO[-i][1],CO[-i][2],CO[-i][3],CO[-i][4],CO[-i
            ][5],CO[i][6],CO[i][7]))
    i = i+1
listdict = {'bi_1':bi_1,'bi_2':bi_2,'mono':mono}
for liste in ['bi_2','bi_1','bi_2','mono']:
    for tmin in [1e6,1e8]:
        CO = np.asarray(listdict[liste])
        start=time.strftime("%H:%M:%S")
        for rows in CO:
            n = np.zeros((4))
            Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3],n[4] = *rows
            scan = [(Y_CO,Y_H,Y_O,(2*Y_O+Y_H)/2000,(sqrt(3)*Y_H)/2000,n[0],n
                [1],n[2],n[3])]
            F_O = Y_O*Fm/1000
            while Y_CO > 0:
                Y_CO,Y_H = Y_CO-nstep,Y_H+nstep
                F_CO,F_H = Y_CO*Fm/1000,Y_H*Fm/1000
                max_step,tmax,t=tmin/6,tmin,0
                while t!=tmin:
                    res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,
                        s_CO,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=
                        atol,dense_output=True)
                    border = res.y<0
                    if border.any()==True:
                        A = np.arange(len(res.t))
                        for it in A:
                            if border[:,it].any()==True:
                                if it==0:
                                    max_step,tmax = max_step/10,t+(tmin-t)/5
                                else:
                                    t=res.t[it-1]
                                    tmax, max_step = t+(tmin-t)*0.75, ((tmin-t)
                                        *0.75)/6
                                    n = res.sol(tmax)
                                    n[abs(n) < tol] = 0.0
                                    break
                    else:
                        t = tmax
                        tmax,max_step=tmin,tmin/6
                n = res.sol(tmin)
                n[abs(n) < tol] = 0.0
                scan.append((Y_CO,Y_H,Y_O,(2*Y_O+Y_H)/2000,(sqrt(3)*Y_H)/2000,n
                    [0],n[1],n[2],n[3]))
                name = 'filling_H0\\O_const\\'+str("{:.1f}".format(log10(tmin))
```

```
                )+'lgs\\'+str(int(rows[1]))+'_'+str(int(rows[2]))+'_'+str(
                int(rows[3]))
            np.savetxt(name+'-'+liste+'.txt', (np.asarray(scan)), fmt="%.0f\t"
                *3 +"%.8e\t"*2+ "%.12e\t"*4, header=header,footer="This file
                was generated on "+time.strftime("%d.%m.%Y %H:%M:%S")+" by SPH0
                .py.\nT = "+str(T)+"K\nE = "+str(E)+"\nv = "+str(v)+"\nF_tot =
                "+str(Fm))
            n = np.zeros((4))
            Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3],n[4] = *rows
            scan = [(Y_CO,Y_H,Y_O,(2*Y_O+Y_H)/2000,(sqrt(3)*Y_H)/2000,n[0],n
                [1],n[2],n[3])]
            F_CO = Y_CO*Fm/1000
            while Y_O > 0:
                Y_H,Y_O = Y_H+nstep,Y_O-nstep
                F_H,F_O = Y_H*Fm/1000,Y_O*Fm/1000
                max_step,tmax,t=tmin/6,tmin,0
                while t!=tmin:
                    res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,
                        s_CO,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=
                        atol,dense_output=True)
                    border = res.y<0
                    if border.any()==True:
                        A = np.arange(len(res.t))
                        for it in A:
                            if border[:,it].any()==True:
                                if it==0:
                                    max_step,tmax = max_step/10,t+(tmin-t)/5
                                else:
                                    t=res.t[it-1]
                                    tmax, max_step = t+(tmin-t)*0.75, ((tmin-t)
                                        *0.75)/6
                                    n = res.sol(tmax)
                                    n[abs(n) < tol] = 0.0
                                    break
                    else:
                        t = tmax
                        tmax,max_step=tmin,tmin/6
                n = res.sol(tmin)
                n[abs(n) < tol] = 0.0
                scan.append((Y_CO,Y_H,Y_O,(2*Y_O+Y_H)/2000,(sqrt(3)*Y_H)/2000,n
                    [0],n[1],n[2],n[3]))
                name = 'filling_H0\\CO_const\\'+str("{:.1f}".format(log10(tmin)
                    ))+'lgs\\'+str(int(rows[1]))+'_'+str(int(rows[2]))+'_'+str(
                    int(rows[3]))
            np.savetxt(name+'-'+liste+'.txt', (np.asarray(scan)), fmt="%.0f\t"
                *3 +"%.8e\t"*2+ "%.12e\t"*4, header=header,footer="This file
                was generated on "+time.strftime("%d.%m.%Y %H:%M:%S")+" by SPH0
                .py.\nT = "+str(T)+"K\nE = "+str(E)+"\nv = "+str(v)+"\nF_tot =
                "+str(Fm))
        print('finished calculation_tmin = ', tmin, 'Liste = ',liste,'starting
            time = ',start, 'ending time = ', time.strftime("%H:%M:%S"))
```

O-filling.py

```
from math import *
import numpy as np
from scipy.integrate import solve_ivp
import time
import os
from ternary_basics import *
T,k,E,v,Fm,s_CO,s_O,s_H,a,b,c = initialize()
tol = np.finfo(float).eps*10 #cutoff tolerance
rtol,atol=1e-9,1e-6
cdic = {1:'LSODA',2:'Radau',3:'BDF'} #integration solvers
calc,nstep = 2,1
calca = cdic[calc]
O = np.loadtxt('border_O0\\9.0lgs_Radau.txt')
header = 'Y\-(A)\tY\-(B)\tY\-(C)\tY\-(A)\ty\tn\-(A)\tn\-(B)\tn\-(C)\tn\-(D)'

i = 1
bi_1,bi_2,mono = [],[],[]
while O[i-1][0]< O[-i][0]:
    summe=0
    for j in [4,5,6,7]:
        summe = abs(O[i-1][j]-O[-i][j])+summe
    if summe>0.01:
        bi_1.append((O[-i][0],O[-i][1],O[-i][2],O[-i][3],O[i-1][4],O[i-1][5],O[
            i-1][6],O[i-1][7]))
        bi_2.append((O[-i][0],O[-i][1],O[-i][2],O[-i][3],O[-i][4],O[-i][5],O[-i
            ][6],O[-i][7]))
    else:
        mono.append((O[-i][0],O[-i][1],O[-i][2],O[-i][3],O[-i][4],O[-i][5],O[i
            ][6],O[i][7]))
    i = i+1

listdict = {'bi_1':bi_1,'bi_2':bi_2,'mono':mono}
for liste in ['bi_1','bi_2','mono']:
    for tmin in [1e8]:
        O = np.asarray(listdict[liste])
        start=time.strftime("%H:%M:%S")
        for rows in O:
            n = np.zeros((4))
            Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3],n[4] = *rows
            scan = [(Y_CO,Y_H,Y_O,(2*Y_O+Y_H)/2000,(sqrt(3)*Y_H)/2000,n[0],n
                [1],n[2],n[3])]
            F_CO = Y_CO*Fm/1000
            while Y_H > 0:
                Y_H,Y_O = Y_H-nstep,Y_O+nstep
                F_H,F_O = Y_H*Fm/1000,Y_O*Fm/1000
                max_step,tmax,t=tmin/6,tmin,0
                while t!=tmin:
                    res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,
                        s_CO,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=
```

```
                atol,dense_output=True)
            border = res.y<0
            if border.any()==True:
                A = np.arange(len(res.t))
                for it in A:
                    if border[:,it].any()==True:
                        if it==0:
                            max_step,tmax = max_step/10,t+(tmax-t)/5
                        else:
                            t=res.t[it-1]
                            tmax, max_step = t+(tmin-t)*0.75, ((tmin-t)
                                *0.75)/6
                            n = res.sol(tmax)
                            n[abs(n) < tol] = 0.0
                            break
            else:
                t = tmax
                tmax,max_step=tmin,tmin/6
        n = res.sol(tmin)
        n[abs(n) < tol] = 0.0
        scan.append((Y_CO,Y_H,Y_O,(2*Y_O+Y_H)/2000,(sqrt(3)*Y_H)/2000,n
            [0],n[1],n[2],n[3]))
        name = 'filling_O0\\CO_const\\'+str("{:.1f}".format(log10(tmin)
            ))+'lgs\\'+str(int(rows[1]))+'_'+str(int(rows[2]))+'_'+str(
            int(rows[3]))
    np.savetxt(name+'-'+liste+'.txt', (np.asarray(scan)), fmt="%.0f\t"
        *3 +"%.8e\t"*2+ "%.12e\t"*4, header=header,footer="This file
        was generated on "+time.strftime("%d.%m.%Y %H:%M:%S")+" by SPH0
        .py.\nT = "+str(T)+"K\nE = "+str(E)+"\nv = "+str(v)+"\nF_tot =
        "+str(Fm))

    n = np.zeros((4))
    Y_CO,Y_H,Y_O,n[0],n[1],n[2],n[3],n[4] = *rows
    scan = [(Y_CO,Y_H,Y_O,(2*Y_O+Y_H)/2000,(sqrt(3)*Y_H)/2000,n[0],n
        [1],n[2],n[3])]
    F_H = Y_H*Fm/1000
    while Y_CO > 0:
        Y_CO,Y_O = Y_CO-nstep,Y_O+nstep
        F_CO,F_O = Y_CO*Fm/1000,Y_O*Fm/1000
        max_step,tmax,t=tmin/6,tmin,0
        while t!=tmin:
            res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,
                s_CO,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=
                atol,dense_output=True)
            border = res.y<0
            if border.any()==True:
                A = np.arange(len(res.t))
                for it in A:
                    if border[:,it].any()==True:
                        if it==0:
                            max_step,tmax = max_step/10,t+(tmin-t)/5
```

```
                                else:
                                    t=res.t[it-1]
                                    tmax,max_step=t+(tmax-t)*0.75,((tmin-t)
                                        *0.75)/6
                                    n = res.sol(tmax)
                                    n[abs(n) < tol] = 0.0
                                    break
                            else:
                                t = tmax
                                tmax,max_step=tmin,tmin/6
                        n = res.sol(tmin)
                        n[abs(n) < tol] = 0.0
                        scan.append((Y_CO,Y_H,Y_O,(2*Y_O+Y_H)/2000,(sqrt(3)*Y_H)/2000,n
                            [0],n[1],n[2],n[3]))
                        name = 'filling_O0\\H_const\\'+str("{:.1f}".format(log10(tmin))
                            )+'lgs\\'+str(int(rows[1]))+'_'+str(int(rows[2]))+'_'+str(
                            int(rows[3]))
                    np.savetxt(name+'-'+liste+'.txt', (np.asarray(scan)), fmt="%.0f\t"
                        *3 +"%.8e\t"*2+ "%.12e\t"*4, header=header,footer="This file
                        was generated on "+time.strftime("%d.%m.%Y %H:%M:%S")+" by SPH0
                        .py.\nT = "+str(T)+"K\nE = "+str(E)+"\nv = "+str(v)+"\nF_tot =
                        "+str(Fm))
                print('finished calculation_tmin = ', tmin, 'Liste = ',liste,'starting
                    time = ',start, 'ending time = ', time.strftime("%H:%M:%S"))
```

zus.py

```
#author: Kevin Rohe
import os
from math import *
import numpy as np
import time

for filling in ['H0','CO15','O0']:#'H0','CO15','O0'
    for const in ['H','O','CO']:#'H','O','CO'
        for tmin in [1e6,1e8]:
            direc = 'filling_'+filling+'\\'+const+'_const\\'+str("{:.1f}".
                format(log10(tmin)))+'lgs\\'
            print(filling+const+str("{:.1f}".format(log10(tmin)))+' start')
            if os.path.isdir(direc)==True:
                f1 = open('zus_'+filling+const+str("{:.1f}".format(log10(tmin))
                    )+'MONO.txt','ab')
                f2 = open('zus_'+filling+const+str("{:.1f}".format(log10(tmin))
                    )+'MONO+BI_1.txt','ab')
                f3 = open('zus_'+filling+const+str("{:.1f}".format(log10(tmin))
                    )+'MONO+BI_2.txt','ab')
                matches = [ fname for fname in os.listdir(direc) if ('mono' in
                    fname)]
                for j in matches:
                    Data = np.loadtxt(direc+j)
                    Data = Data[(Data[:,5]>=0)*(Data[:,6]>=0)*(Data[:,7]>=0)*(
```

```
                    Data[:,8]>=0)]
                np.savetxt(f1, Data, fmt="%.0f\t"*3 +"%.8e\t"*2+ "%.12e\t"
                    *4)
                np.savetxt(f2, Data, fmt="%.0f\t"*3 +"%.8e\t"*2+ "%.12e\t"
                    *4)
                np.savetxt(f3, Data, fmt="%.0f\t"*3 +"%.8e\t"*2+ "%.12e\t"
                    *4)
            f1.close()
            f4 = open('zus_'+filling+const+str("{:.1f}".format(log10(tmin))
                )+'BI_1.txt','ab')
            matches = [ fname for fname in os.listdir(direc+'\\') if ('bi_1
                ' in fname)]
            for j in matches:
                Data = np.loadtxt(direc+j)
                Data = Data[(Data[:,5]>=0)*(Data[:,6]>=0)*(Data[:,7]>=0)*(
                    Data[:,8]>=0)]
                np.savetxt(f2, Data, fmt="%.0f\t"*3 +"%.8e\t"*2+ "%.12e\t"
                    *4)
                np.savetxt(f4, Data, fmt="%.0f\t"*3 +"%.8e\t"*2+ "%.12e\t"
                    *4)
            f2.close()
            f4.close()
            f5 = open('zus_'+filling+const+str("{:.1f}".format(log10(tmin))
                )+'BI_2.txt','ab')
            matches = [ fname for fname in os.listdir(direc+'\\') if ('bi_2
                ' in fname)]
            for j in matches:
                Data = np.loadtxt(direc+j)
                Data = Data[(Data[:,5]>=0)*(Data[:,6]>=0)*(Data[:,7]>=0)*(
                    Data[:,8]>=0)]
                np.savetxt(f3, Data, fmt="%.0f\t"*3 +"%.8e\t"*2+ "%.12e\t"
                    *4)
                np.savetxt(f5, Data, fmt="%.0f\t"*3 +"%.8e\t"*2+ "%.12e\t"
                    *4)
            f3.close()
            f5.close()
        print(filling+const+str("{:.1f}".format(log10(tmin)))+' finished')
```

zoom.py

```
#author: Kevin Rohe
import os
from math import *
import numpy as np
import time
```

```
Abbrev = ['zus']
header = 'Y\-(A)\tY\-(B)\tY\-(C)\tY\-(A)\ty\tn\-(A)\tn\-(B)\tn\-(C)\tn\-(D)'
for i in Abbrev:
    matches = [ fname for fname in os.listdir("./") if fname.startswith(i) and
        ('MONO+BI' in fname)]
    for file in matches:
        print('imported', file)
        NBF = np.loadtxt(file)
        result = result[(NBF[:,0]<=250)*(result[:,1]<=250)*(result[:,2]>=750)]
        result[:,:1] = result[:,:1]*4
        result[:,2] = 1000-result[:,0]-result[:,1]
        np.savetxt('zoom_'+file, (result), delimiter='\t',fmt="%.0f\t"*3 +"%.8e
            \t"*2+ "%.12e\t"*4, header=header,footer="This file was generated
            on "+time.strftime("%d.%m.%Y %H:%M:%S")+" by zus+zoom")
```

Route.py

```
from math import *
import numpy as np
from scipy.integrate import solve_ivp
import time
import os
from ternary_basics import *
T,k,Fm,s_CO,s_O,s_H,a,b,c = initialize()
tol = np.finfo(float).eps*10 #cutoff tolerance
nstep = 1                    #stepwidth for Y
cdic = {1:'LSODA',2:'Radau',3:'BDF'} #integration solvers
rtol,atol=1e-9,1e-6
nstep = 1 #stepwidth for Y
header = '\tY\-(A)\tY\-(B)\tY\-(C)\tx\ty\tn[CO]\tn[H]\tn[O]\tn[OH]'
tmin = 1e8
cdic = {1:'LSODA',2:'Radau',3:'BDF'}
calc = 2
calca = cdic[calc]
rate = {'VLR':[1.902084229677e-02,9.761826662101e-01,9.615712752842e
    -08,1.290779735582e-08],'UR'[7.690962824681e-02,2.011343927304e
    -02,1.347261516183e-01,1.808516861712e-02],'LR'[1.495968083815e
    -01,8.125823654650e-01,8.457790551004e-06,1.135344299655e-06]}
n=[0,0,0,0]
A = np.array([107,49,1000-107-49])
B = np.array([107,5,1000-107-5])
C = np.array([110,49,1000-110-49])
D = np.array([47,109,1000-47-109])
Eo = np.array([47,65,1000-47-65])
F = np.array([154,5,1000-154-5])
G = np.array([110,109,1000-110-109])

routes = {'ACF':np.array([C,F]),'ACG':np.array([C,G]),'ABF':np.array([B,F]),'
    ABE':np.array([B,Eo]),'ADE':np.array([D,Eo]),'ADG':np.array([D,G])}
for Route in ['ADG']:#,'ADE','ACG','ABF','ABE','ACF']:
```

```
    for nsav in ['UR','VLR','LR']:
        scan = []
        start=time.strftime("%H:%M:%S")
        n[0]=rate[nsav][0]
        n[1]=rate[nsav][1]
        n[2]=rate[nsav][2]
        n[3]=rate[nsav][3]
        W=routes[Route]
        counter2 = 0
        scan.append((1,A[0]*4,A[1]*4,1000-A[0]*4-A[1]*4,(2*A[2]+A[1])/2000,(
            sqrt(3)*A[1])/2000,n[0],n[1],n[2],n[3]))
        al = A
        for element in W:
            counter3 = 1
            bal = element-al
            for val in bal:
                if val>0:
                    bal=bal/val
                    break
            while (al != element).any():
                al=al+bal
                Y_CO,Y_H,Y_O = *al
                F_H,F_CO,F_O = Y_H*Fm/1000,Y_CO*Fm/1000,Y_O*Fm/1000
                tmax,max_step=tmin,tmin/6
                t = 0
                while t!=tmin:
                    res = solve_ivp(rhs,[t,tmax],n,calca,args=((k,F_CO,F_O,F_H,
                        s_CO,s_O,s_H,a,b,c),),max_step=max_step,atol=rtol,rtol=
                        atol,dense_output=True) #Solve diff
                    border = res.y<0
                    if border.any()==True:
                        A = np.arange(len(res.t))
                        for it in A:
                            if border[:,it].any()==True:
                                if it==0:
                                    max_step = max_step/10
                                    tmax=t+(tmax-t)/5
                                else:
                                    t=res.t[it-1]
                                    tmax=t+(tmin-t)*0.75
                                    max_step = (tmax-t)/6#(res.t[it]-res.t[it
                                        -1])
                                    for it in [0,1,2,3]:
                                        if res.sol(tmax)[it]<tol:
                                            n[it]=0
                                        else:
                                            n[it]=res.sol(tmax)[it]
                                    break
                    else:
                        t = tmax
                        tmax=tmin
```

```
                        max_step=tmin/6
                        for it in [0,1,2,3]:
                            if res.sol(tmax)[it]<tol:
                                n[it]=0
                            else:
                                n[it]=res.sol(tmax)[it]
                    scan.append((val,Y_CO*4,Y_H*4,1000-Y_H*4-Y_CO*4,(2*Y_O+Y_H)
                        /2000,(sqrt(3)*Y_H)/2000,n[0],n[1],n[2],n[3]))#Y_O, nCO, nH
                        , nO, nOH
                    counter3 = counter3+1
                counter2 = counter2+1
                al = element
            np.savetxt('Route'+Route+'_'+nsav+'_'+str("{:.1f}".format(log10(tmin)))
                +'lgs.txt', (np.asarray(scan)), delimiter='\t',fmt='%.8e', header=
                header)

Pathbox=[A,B,C,D,Eo,F,G]
Pathlength=[]
header = '\tY\-(A)\tY\-(B)\tY\-(C)\tx\ty\tn[CO]\tn[H]\tn[O]\tn[OH]'
tmin = 1e8
for nsav in ['UR','VLR','LR']:
    val=-1
    R1 = np.loadtxt('RouteACG_'+nsav+'_'+str("{:.1f}".format(log10(tmin)))+'lgs
        .txt')
    R2 = np.loadtxt('RouteADG_'+nsav+'_'+str("{:.1f}".format(log10(tmin)))+'lgs
        .txt')
    R2 = np.flip(R2,0)
    R2 = np.delete(R2,0,0)
    R3 = np.concatenate((R1,R2))

    R1 = np.loadtxt('RouteADE_'+nsav+'_'+str("{:.1f}".format(log10(tmin)))+'lgs
        .txt')
    R1 = np.delete(R1,0,0)
    R3 = np.concatenate((R3,R1))
    R2 = np.loadtxt('RouteABE_'+nsav+'_'+str("{:.1f}".format(log10(tmin)))+'lgs
        .txt')
    R2 = np.flip(R2,0)
    R2 = np.delete(R2,0,0)
    R3 = np.concatenate((R3,R2))

    R1 = np.loadtxt('RouteABF_'+nsav+'_'+str("{:.1f}".format(log10(tmin)))+'lgs
        .txt')
    R1 = np.delete(R1,0,0)
    R3 = np.concatenate((R3,R1))
    R2 = np.loadtxt('RouteACF_'+nsav+'_'+str("{:.1f}".format(log10(tmin)))+'lgs
        .txt')
    R2 = np.flip(R2,0)
    R2 = np.delete(R2,0,0)
    R3 = np.concatenate((R3,R2))
    n=0
    for row in R3:
```

```
        n=n+1
        for point in Pathbox:
            if all([row[1]/4,row[2]/4,1000-row[1]/4-row[2]/4] == point):
                Pathlength.append(n)
                n=0
                break
    k=0
    val=0
    for row in R3:
        val= val+1/Pathlength[k]
        row[0]=val
        for point in Pathbox:
            if all([row[1]/4,row[2]/4,1000-row[1]/4-row[2]/4] == point):
                k=k+1
                break
    np.savetxt('ACGDADEBABFCA'+'_'+nsav+'_'+str("{:.1f}".format(log10(tmin)))+'
        lgs.txt', (R3), delimiter='\t',fmt='%.8e', header=header)
```

C

Supplementary data for Chapter 4

The following sections contain the source codes for numerical simulation used in chapter 4.

discrete_basics.py

```
import numpy as np
import colorsys
import matplotlib.pyplot as plt
from matplotlib import cm
from matplotlib.colors import ListedColormap, LinearSegmentedColormap

def define_ccolormap(): # custom colormap
    it, k = 55, 0
    viridis = cm.get_cmap('viridis', 256)
    newcolors = viridis(np.linspace(0, 1, 4*it +1))
    a, b = 4/6, 1/4

    for i, j in [(3/6, 1/3), (2/6, 1/2), (1/6, 1), (1/12, 1)]:
        for h, v in zip(np.linspace(a, i, it, endpoint = False), np.linspace(b,
             j, it, endpoint = False)):
            col = list(colorsys.hsv_to_rgb(h, 1.0, v))
            col.append(1)
            newcolors[k, :] = col
            k = k +1
            newcmp = ListedColormap(newcolors)
        a, b = i, j

    col = list(colorsys.hsv_to_rgb(a, 1.0, b))
    col.append(1)
```

```
    newcolors[k, :] = col
    newcmp = ListedColormap(newcolors)
    return newcmp
    #         4./6.,1.0,0.25, // dark blue
    #         2./6.,1.0,0.5, // green
    #         1./6.,1.0,1.0, // light yellow

def fp(u,k): #derrivative
    a, b, c, d = k
    return -5*u**4 - 3*a*u**2 - 2*b*u - c
def reacterm(x, k): #reaction term
    a, b, c, d = k
    return -x**5 - a*x**3 - b*x**2 - c*x - d
def potential(x, k): #potential
    a, b, c, d = k
    return -x**6/6 - a*x**4/4 - b*x**3/3 - c*x**2/2 - d*x

def minima(a, b, c, d):
    coeffs = [-1., 0, -a, -b, -c, -d]
    allroots = np.roots(coeffs)
    realroots = [x.real for x in allroots if np.isreal(x)]
    stableroots = [x for x in realroots if fp(x, (a, b, c, d)) < 0]
    return np.sort(stableroots)

def mesh_hex(resol):
    res = int(resol*2/np.sqrt(3)), resol
    xx, yy = np.meshgrid(np.arange(res[1]), np.arange(res[0]), sparse = False,
        indexing = 'xy')
    yy = yy * np.sqrt(3)/2
    xx = xx*1.0
    xx[::2, :] = (xx[::2, :] + 1/2)
    return res, xx, yy

def mesh_tri(resol):
    res = int(resol*2/np.sqrt(3) + 1), resol*2//3 + 1
    xx, yy = np.meshgrid(np.arange(res[1]), np.arange(res[0]), sparse = False,
        indexing = 'xy')
    xx = xx*3.0/2
    xx[::2, ::2] = xx[::2, ::2] + 1/2
    xx[1::2, 1::2] = xx[1::2, 1::2] + 1/2
    yy = yy*np.sqrt(3)/2
    return res, xx, yy

def mesh_reg(resol):
    res = resol, resol
    xx, yy = np.meshgrid(np.arange(res[1]), np.arange(res[0]), sparse = False,
        indexing = 'xy')
    return res, xx, yy

def inioct(res, xx, yy, solutions, resol = 100):#initial octagonal shape
    r1 = resol*3/8
```

```
    r2 = resol*2/8
    r3 = resol/8
    r4 = np.sqrt(2)

    z = (xx + yy)*0.0 + solutions[0]
    for i in np.arange(res[0]):
        for j in np.arange(res[1]):
                radx = (np.abs(xx[i, j] - resol//2))
                rady = (np.abs(yy[i, j] - resol//2))
            if (radx < r3 and rady < r3) and radx + rady < r3*r4:
                z[i, j] = solutions[2]
            elif (radx < r2 and rady < r2) and radx + rady < r2*r4:
                z[i, j] = solutions[1]
            elif (radx < r1 and rady < r1) and radx + rady < r1*r4:
                z[i, j] = solutions[0]
            else:
                z[i, j] = solutions[2]
    return z

def inistripe(xx, yy, resol, phi, soli, solj): #initial angular front
    z = (xx + yy)*0.0
    m = np.tan(phi*np.pi/180)
    y = xx*m + (1 - m)*resol/2
    z[y <= yy] = soli
    z[y > yy] = solj
    return z

if __name__ == "__main__":
    pass
else:
    newcmp = define_ccolormap()
```

discrete_plot.py

```
import numpy as np
import colorsys
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
from matplotlib.collections import PatchCollection
from discrete_basics import newcmp

def patches_reg(xx, yy, z1, rad = 1):
    squares = [mpatches.RegularPolygon((xi, yi), 4, radius=np.sqrt(2)*0.5*rad,
        orientation=np.deg2rad(45), linewidth=0) for xi, yi in zip(xx.flatten()
        , yy.flatten())]
    collection = PatchCollection(squares, cmap = newcmp, norm = plt.Normalize
        (-3.5, 3.5), rasterized = True)
    collection.set_array(z1.flatten())
    return collection

def patches_hex(xx, yy, z1, rad = 1):
```

```
    hexagons = [mpatches.RegularPolygon((xi, yi), 6, radius = np.sqrt(1.1/3)*
        rad, orientation = np.deg2rad(0), linewidth = 0) for xi, yi in zip(xx.
        flatten(), yy.flatten())]
    collection = PatchCollection(hexagons, cmap = newcmp, norm = plt.Normalize
        (-3.5, 3.5), rasterized = True)
    collection.set_array(z1.flatten())
    return collection

def patches_tri(xx, yy, z1, rad = 1):
    temp = np.hstack((xx[::2, ::2].flatten(), xx[1::2, 1::2].flatten())), np.
        hstack((yy[::2, ::2].flatten(), yy[1::2, 1::2].flatten()))
    triangles = [mpatches.RegularPolygon((xi, yi), 3, radius = 1.075 * rad,
        orientation = np.deg2rad(-30), linewidth = 0) for xi, yi in zip(*temp)]
    temp = np.hstack((xx[1::2, ::2].flatten(), xx[::2, 1::2].flatten())), np.
        hstack((yy[1::2, ::2].flatten(), yy[::2, 1::2].flatten()))
    triangles = triangles + [mpatches.RegularPolygon((xi, yi), 3, radius =
        1.075*rad, orientation = np.deg2rad(30), linewidth = 0) for xi, yi in
        zip(*temp)]
    collection = PatchCollection(triangles, cmap = newcmp, norm = plt.Normalize
        (-3.5, 3.5), rasterized = True)
    collection.set_array(np.hstack((z1[::2, ::2].flatten(),z1[1::2, 1::2].
        flatten(), z1[1::2, ::2].flatten(), z1[::2, 1::2].flatten())))
    return collection

def plot_reg(xx, yy, z1, name): #standard-plots
    fig, ax = plt.subplots(1, 1, figsize = (5, 5), dpi = 150)
    ax.add_collection(patches_reg(xx, yy, z1))
    ax.axis('equal');
    ax.axis('off')
    plt.tight_layout()
    fig.savefig(name + ".pdf")
    plt.show()

def plot_hex(xx, yy, z1, name): #standard-plots
    fig, ax = plt.subplots(1, 1, figsize = (5, 5), dpi = 150)
    ax.add_collection(patches_hex(xx, yy, z1))
    ax.axis('equal');
    ax.axis('off')
    plt.tight_layout()
    fig.savefig(name + ".pdf")
    plt.show()

def plot_tri(xx, yy, z1, name): #standard-plots
    fig, ax = plt.subplots(1, 1, figsize = (5, 5), dpi = 150)
    ax.add_collection(patches_tri(xx, yy, z1))
    ax.axis('equal');
    ax.axis('off')
    plt.tight_layout()
    fig.savefig(name + ".pdf")
    plt.show()
```

```
if __name__ == "__main__":
    pass
```

discrete_sim.py

```
import numpy as np
from numba import jit
from scipy.integrate import solve_ivp
from scipy.optimize import minimize

@jit(nopython = True)
def rhs_reg(t,z,k):
    a,b,c,d,diff,dx,res = k
    m = z.reshape(res)
    dz = np.zeros_like(m)
    i = 0
    j = res[0] -1
    dz[0,0] = m[1,0] +m[0,1] -2*m[0,0]
    dz[0,j] = m[i +1,j] +m[i,j -1] -2*m[0,j]
    dz[j,0] = m[j -1,0] +m[j,1] -2*m[j,0]
    dz[j,j] = m[j -1,j] +m[j,j -1] -2*m[j,j]
    lim1 = 1
    for i in range(lim1,res[0] -lim1):
        for j in range(1,res[1] -1):
            dz[i,j] = m[i +1,j] +m[i -1,j] +m[i,j +1] +m[i,j -1] -4*m[i,j]
        j = 0
        dz[i,j] = m[i +1,j] +m[i -1,j] +m[i,j +1] -3*m[i,j]
        dz[j,i] = m[j +1,i] +m[j,i +1] +m[j,i -1] -3*m[j,i]
        j = res[1] -1
        dz[i,j] = m[i +1,j] +m[i -1,j] +m[i,j -1] -3*m[i,j]
        dz[j,i] = m[j -1,i] +m[j,i +1] +m[j,i -1] -3*m[j,i]

    m = dz*diff/dx**2 -m**5 - a*m**3 - b*m**2 - c*m - d
    return m.flatten()

@jit(nopython = True)
def rhs_hex(t,z,k):
    a,b,c,d,diff,dx,res = k
    m = z.reshape(res)
    dz = np.zeros_like(m)
    i = 0
    j1 = res[0] -1
    j2 = res[1] -1
    dz[0,0] = m[0,1] +m[1,0] + (m[1,1]) -3*m[0,0]
    dz[0,j2] = m[0,j2 -1] +m[1,j2] -2*m[0,j2]
    dz[j1,0] = m[j1,1] +m[j1 -1,0] -2*m[j1,0]
    dz[j1,j2] = m[j1,j2 -1] +m[j1 -1,j2] +m[j1 -1,j2 -1] -3*m[j1,j2]

    for i in range(1,res[0] -1):
        for j in range(1,res[1] -1):
            dz[i,j] = m[i,j +1] +m[i,j -1] +m[i +1,j] +m[i -1,j] +(m[i -1,j -1]
```

```
                +m[i +1,j -1])*(i%2) +(m[i +1,j +1] +m[i -1,j +1])*((i +1)%2)
                -6*m[i,j]
        j = 0
        dz[i,j] = m[i,j +1] +m[i +1,j] +m[i -1,j]
                +(m[i +1,j +1] +m[i -1,j +1] -2*m[i,j])*((i +1)%2)
                -3*m[i,j]
        j = res[1] -1
        dz[i,j] = m[i,j -1] +m[i +1,j] +m[i -1,j] +(m[i -1,j -1] +m[i +1,j -1]
            -2*m[i,j])*(i%2) -3*m[i,j]

    for j in range(1,res[1] -1):
        dz[0,j] = m[0,j +1] +m[0,j -1] +m[1,j] +m[1,j +1] -4*m[0,j]
        i = res[0] -1
        dz[i,j] = m[i,j +1] +m[i,j -1] +m[i -1,j] + (m[i -1,j -1])*(i%2) +(m[i
            -1,j +1])*((i +1)%2) -4*m[i,j]

    m = dz*diff*2/(3*dx**2) -m**5 - a*m**3 - b*m**2 - c*m - d
    return m.flatten()

@jit(nopython = True)
def rhs_tri(t,z,k):
    a,b,c,d,diff,dx,res = k
    m = z.reshape(res)
    dz = np.zeros_like(m)
    i = 0
    j1 = res[0] -1
    j2 = res[1] -1

    dz[0,0] = m[1,0] +m[0,1] -2*m[0,0]
    dz[0,j2] = m[1,j2] + (j2%2)*((m[0,j2 -1] -m[0,j2])) -m[0,j2]
    dz[j1,0] = m[j1 -1,0] + ((j1 +1)%2)*(m[j1,1] -m[j1,0]) -m[j1,0]
    dz[j1,j2] = m[j1 -1,j2] + ((j2 +1)%2)*((j1%2)*(m[j1,j2 -1] -m[j1,j2])) +(j2
        %2)*(((j1 +1)%2)*(m[j1,j2 -1] -m[j1,j2])) -m[j1,j2]

    for i in range(1,res[0] -1):
        for j in range(1,res[1] -1):
            dz[i,j] = m[i +1,j] +m[i -1,j] + ((j +1)%2)*((i%2)*(m[i,j -1]) +((i
                +1)%2)*(m[i,j +1])) + (j%2)*(((i +1)%2)*(m[i,j -1]) +(i%2)*(m[
                i,j +1])) -3*m[i,j]
        j = 0
        dz[i,j] = m[i +1,j] +m[i -1,j] +((i +1)%2)*(m[i,j +1] -m[i,j]) -2*m[i,j
            ]
        j = res[1] -1
        dz[i,j] = m[i +1,j] +m[i -1,j] + ((j +1)%2)*(i%2)*(m[i,j -1] -m[i,j])
            +(j%2)*((i +1)%2)*(m[i,j -1] -m[i,j]) -2*m[i,j]
    for j in range(1,res[1] -1):
        i = 0
        dz[i,j] = m[i +1,j] + ((j +1)%2)*((m[i,j +1])) +(j%2)*(( +m[i,j -1]))
            -2*m[i,j]
        i = res[0] -1
        dz[i,j] = m[i -1,j] + ((j +1)%2)*((i%2)*( +m[i,j -1]) +((i +1)%2)*(m[i,
```

```
            j +1])) +(j%2)*(((i +1)%2)*(m[i,j -1]) +(i%2)*( +m[i,j +1])) -2*m[i
            ,j]

    m = dz*diff*4/(3*dx**2) -m**5 - a*m**3 - b*m**2 - c*m - d
    return m.flatten()

def vecfit(dev, xx, yy, phi, resol):
    a = np.array([resol/2 + dev[0]*np.cos(np.deg2rad(phi + 90)), resol/2 + dev
        [0]*np.sin(np.deg2rad(phi + 90))]) #point on fitline, dev=0==ini
    b = np.array([np.cos(np.deg2rad(phi)),np.sin(np.deg2rad(phi))]) #
        directional vector
    p = np.column_stack((xx, yy)) #points in 'front'
    diff = np.cross(p -a,b)/np.linalg.norm(b) #distance between the fitline a +
         b and the front data
    return np.sum(np.abs(diff))

def vecfitter(xx, yy, phi, resol = 100):
    res = minimize(vecfit, 0 ,args = (xx, yy, phi, resol), method = 'SLSQP',
        bounds = [((-resol*0.45, resol*0.45))])
    return (res.fun, res.x[0])


if __name__ == "__main__":
    pass
```

fig9.py

```
import matplotlib.pyplot as plt
import matplotlib as mpl
from mpl_toolkits.axes_grid1 import ImageGrid
from discrete_basics import *
from discrete_plot import *
mpl.rcParams['backend'] = 'pdf'
mpl.rc('font',**{'family':'serif'})
mpl.rcParams['mathtext.fontset'] = 'cm'
def axin_setup(axins):
    axins.set_xlim(10, 50)
    axins.set_ylim(10, 50)
    axins.set_xticklabels([])
    axins.set_yticklabels([])
    axins.set_xticks([])
    axins.set_yticks([])
    return None

resolution = 100
a,b,c,d = -10,2,18.75,-5
solutions = minima(a,b,c,d)

# Set up figure and image grid
fig = plt.figure(figsize=(15, 5), dpi=150)
grid = ImageGrid(fig, 111, nrows_ncols=(1,3), axes_pad=0.15, share_all=True,
```

```
    cbar_location="right", cbar_mode="single", cbar_size="7%", cbar_pad=0.1)

res, xv, yv = mesh_reg(resolution)
z = inioct(res,xv,yv,solutions,resolution)
c = grid[0].add_collection(patches_reg(xv,yv,z))
axins = grid[0].inset_axes([0, 0, 0.5, 0.5])
axins.add_collection(patches_reg(xv,yv,z,0.6))
axin_setup(axins)
res, xv, yv = mesh_hex(resolution)
z = inioct(res,xv,yv,solutions,resolution)
grid[1].add_collection(patches_hex(xv,yv,z))
axins = grid[1].inset_axes([0, 0, 0.5, 0.5])
axins.add_collection(patches_hex(xv,yv,z,0.6))
axin_setup(axins)
res, xv, yv = mesh_tri(resolution)
z = inioct(res,xv,yv,solutions,resolution)
grid[2].add_collection(patches_tri(xv,yv,z))
axins = grid[2].inset_axes([0, 0, 0.5, 0.5])
axins.add_collection(patches_tri(xv,yv,z,0.6))
axin_setup(axins)

for ax in grid:
    ax.set_rasterization_zorder(0)
    ax.axis('equal');
    ax.axis('off')
ax.cax.cla()
cb = mpl.colorbar.Colorbar(ax.cax,c)
cb.ax.tick_params(labelsize=15)
cb.set_label(label=r'$u(x,y)$',weight='bold',fontsize=18)
ax.cax.toggle_label(True)
plt.savefig("9_initials.pdf")
plt.show()
```

fig10_11.py

```
import matplotlib.pyplot as plt
import matplotlib as mpl
from discrete_basics import *
from discrete_plot import *
from discrete_sim import *
from scipy.integrate import solve_ivp
mpl.rcParams['backend'] = 'pdf'
mpl.rc('font',**{'family':'serif'})
mpl.rcParams['mathtext.fontset'] = 'cm'
plt.rc('text', usetex=True)
plt.rc('text.latex', preamble=r'\usepackage{amssymb,amsmath,wasysym}')
plt.rcParams['font.size'] = 10

a, b, c, d = -10, 2, 15, -3
resolution = 100
coup, dx = 5, 1.0
```

```
solutions = minima(a,b,c,d)
res, xv, yv = mesh_reg(resolution)
z = inioct(res,xv,yv,solutions,resolution)
diff = coup*dx**2
z1 = z.flatten()
timeplus=0
for tlimit in [(1,0.001),(3,0.01),(5,0.01),(10,0.01)]: #Looping to goal-values
    to print
    tstep = tlimit[1]
    t = 0
    while (t*tstep+timeplus)<tlimit[0]: #Looping prevents internal issues
        try:
            reso = solve_ivp(rhs_reg,[0,tstep],z1,'LSODA',args=((a,b,c,d,diff,
                dx,res),))
            z1 = reso.y[:,-1]
            t = t+1
        except(MemoryError): #Workaround against memory issue
            fig, ax = plt.subplots(1,1,figsize = (1,1))
            plt.close()
    plot_reg(xv,yv,z1,"10_reg{}_{}_{}_c{:.2f}_d{}_D{}_dx{}_t{:.3f}s".format(
        resolution,a,b,c,d,diff,dx,tlimit[0]))
    timeplus = timeplus+t*tstep

a, b, c, d = -10, 2, 18.75, -3
coup, dx = 5, 1.0
solutions = minima(a,b,c,d)
res, xv, yv = mesh_reg(resolution)
z = inioct(res,xv,yv,solutions,resolution)
diff = coup*dx**2
z1 = z.flatten()
timeplus=0
for tlimit in [(1,0.001),(3,0.01),(5,0.01),(10,0.01)]: #Looping to goal-values
    to print
    tstep = tlimit[1]
    t = 0
    while (t*tstep+timeplus)<tlimit[0]: #Looping prevents internal issues
        try:
            reso = solve_ivp(rhs_reg,[0,tstep],z1,'LSODA',args=((a,b,c,d,diff,
                dx,res),))
            z1 = reso.y[:,-1]
            t = t+1
        except(MemoryError): #Workaround against memory issue
            fig, ax = plt.subplots(1,1,figsize = (1,1))
            plt.close()
    plot_reg(xv,yv,z1,"10_reg{}_{}_{}_c{:.2f}_d{}_D{}_dx{}_t{:.3f}s".format(
        resolution,a,b,c,d,diff,dx,tlimit[0]))
    timeplus = timeplus+t*tstep

a, b, c, d = -10, 2, 15, -3
coup, dx = 2.5, 1.0
solutions = minima(a,b,c,d)
```

```
res, xv, yv = mesh_hex(resolution)
z = inioct(res,xv,yv,solutions,resolution)
diff = coup*dx**2
z1 = z.flatten()
timeplus=0
for tlimit in [(1,0.001),(5,0.01),(10,0.01)]: #Looping to goal-values to print
    tstep = tlimit[1]
    t = 0
    while (t*tstep+timeplus)<tlimit[0]: #Looping prevents internal issues
        try:
            reso = solve_ivp(rhs_hex,[0,tstep],z1,'LSODA',args=((a,b,c,d,diff,
                dx,res),))
            z1 = reso.y[:,-1]
            t = t+1
        except(MemoryError): #Workaround against memory issue
            fig, ax = plt.subplots(1,1,figsize = (1,1))
            plt.close()
    plot_hex(xv,yv,z1,"11_hex{}_{}_{}_c{:.2f}_d{}_D{}_dx{}_t{:.3f}s".format(
        resolution,a,b,c,d,diff,dx,tlimit[0]))
    timeplus = timeplus+t*tstep

a, b, c, d = -10, 2, 15, -3
coup, dx = 5, 1.0
solutions = minima(a,b,c,d)
res, xv, yv = mesh_tri(resolution)
z = inioct(res,xv,yv,solutions,resolution)
diff = coup*dx**2
z1 = z.flatten()
timeplus=0
for tlimit in [(1,0.001),(5,0.01),(10,0.01)]: #Looping to goal-values to print
    tstep = tlimit[1]
    t = 0
    while (t*tstep+timeplus)<tlimit[0]: #Looping prevents internal issues
        try:
            reso = solve_ivp(rhs_tri,[0,tstep],z1,'LSODA',args=((a,b,c,d,diff,
                dx,res),))
            z1 = reso.y[:,-1]
            t = t+1
        except(MemoryError): #Workaround against memory issue
            fig, ax = plt.subplots(1,1,figsize = (1,1))
            plt.close()
    plot_tri(xv,yv,z1.reshape(res),"11_tri{}_{}_{}_c{:.2f}_d{}_D{}_dx{}_t{:.3f}
        s".format(resolution,a,b,c,d,diff,dx,tlimit[0]))
    timeplus = timeplus+t*tstep
```

fig12.py

```
import matplotlib.pyplot as plt
import matplotlib as mpl
import os
from discrete_basics import *
from discrete_plot import *
from discrete_sim import *
from scipy.integrate import solve_ivp
from cycler import cycler
plt.rcParams['axes.prop_cycle']=cycler('color',['#1f77b4', '#ff7f0e', '#d62728'
    , '#9467bd', '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf'])
mpl.rcParams['backend'] = 'pdf'
mpl.rc('font',**{'family':'serif'})
mpl.rcParams['mathtext.fontset'] = 'cm'
plt.rc('text', usetex=True)
plt.rc('text.latex', preamble=r'\usepackage{amssymb,amsmath,wasysym}')
plt.rcParams['font.size'] = 14
location = "coupstrength/"
try:
    os.makedirs(location)
except:
    pass

a, b = -10, 2
resolution = 100
dx = 1.0
tol = 0.1
coupArr = [2,4,6,8,10]
for par in [(18.75,-3)]:
    c,d = par
    solutions = minima(a,b,c,d)
    for method in [(mesh_tri, rhs_tri)]:#
        res, xv, yv = method[0](resolution)
        z = inioct(res,xv,yv,solutions,resolution)
        for coup in coupArr:#
            diff = coup*dx**2
            print("coup = {}".format(coup),end='\r')
            z1 = z.flatten()
            timeplus=0
            filename = method[0].__name__[-3:]+"_a{}_b{}_c{}_d{}_D{}_dx{}_r{}".
                format(a,b,c,d,diff,dx,resolution)
            if not os.path.isfile(location+filename+".txt"):
                for tlimit in [(0.1,0.001),(10,0.01)]:
                    tstep = tlimit[1]
                    t = 0
                    while (t*tstep+timeplus)<tlimit[0]:
                        try:
                            reso = solve_ivp(method[1],[0,tstep],z1,'LSODA',
                                args=((a,b,c,d,diff,dx,res),))
                            z1 = reso.y[:,-1]
```

```
                        t = t+1
                    except(MemoryError):
                        fig, ax = plt.subplots(1,1,figsize = (1,1))
                        plt.close()
                    N_1, N_2, N_3 = len(z1[np.abs(z1-solutions[0])<tol]),
                        len(z1[np.abs(z1-solutions[1])<tol]), len(z1[np.abs
                        (z1-solutions[2])<tol])
                    f = open(location+filename+'.txt','a')
                    f.write("{}\t{}\t{}\t{}\t{}\n".format(t*tstep+timeplus,
                        N_1,N_2,N_3,np.sum(z1-z.flatten())))
                    f.close()
                timeplus = timeplus+t*tstep
    fig, axs = plt.subplots(3,1,figsize=(5,8),sharex=True,dpi=150)
    fig.tight_layout()
    fig.subplots_adjust(hspace=0)
    for coup in coupArr:#
        diff = coup*dx**2#
        filename = method[0].__name__[-3:]+"_a{}_b{}_c{}_d{}_D{}_dx{}_r{}".
            format(a,b,c,d,diff,dx,resolution)
        data1 = np.loadtxt(location+filename+'.txt')
        axs[0].plot(data1[:,0],data1[:,1]/1e2,label=r'$\mathrm{D}='+str(
            diff)+'$',lw=2)
        axs[1].plot(data1[:,0],data1[:,2]/1e2,label=r'$\mathrm{D}='+str(
            diff)+'$',lw=2)
        axs[2].plot(data1[:,0],data1[:,3]/1e2,label=r'$\mathrm{D}='+str(
            diff)+'$',lw=2)
    axs[0].set_ylabel(r'$N_{0} \cdot 10^{-2}$')
    axs[1].set_ylabel(r'$N_{1} \cdot 10^{-2}$')
    axs[2].set_ylabel(r'$N_{2} \cdot 10^{-2}$')
    axs[2].set_xlabel(r't')
    axs[0].set_xlim(0,10)
    axs[1].set_xlim(0,10)
    axs[2].set_xlim(0,10)
    axs[0].legend()
    filename = location+'12_'+method[0].__name__[-3:]+"_a{}_b{}_c{}_d{}_dx
        {}_r{}".format(a,b,c,d,dx,resolution)
    fig.savefig(filename+'_coup.pdf',bbox_inches='tight')
    plt.show()
```

fig13.py

```
import matplotlib.pyplot as plt
import matplotlib as mpl
from discrete_basics import *
from discrete_plot import *
from discrete_sim import *
from scipy.integrate import solve_ivp
from cycler import cycler
plt.rcParams['axes.prop_cycle']=cycler('color',['#1f77b4', '#ff7f0e', '#d62728'
    , '#9467bd', '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf'])
mpl.rcParams['backend'] = 'pdf'
```

```
mpl.rc('font',**{'family':'serif'})
mpl.rcParams['mathtext.fontset'] = 'cm'
plt.rc('text', usetex=True)
plt.rc('text.latex', preamble=r'\usepackage{amssymb,amsmath,wasysym}')
plt.rcParams['font.size'] = 16

a,b,c,d = -10, 2, 15, -5
solutions = minima(a,b,c,d)
resolution = 100
res, xv, yv = mesh_reg(resolution)
dx = 1.0
phi = 30
ini = (0,1)
tol = 0.1
z = inistripe(xv,yv,resolution,phi,solutions[ini[0]],solutions[ini[1]])
for coup in [5,15]:
    diff = coup*dx**2
    z1 = z.flatten()
    timeplus=0
    for tlimit in [(1,0.001),(5,0.01)]:
        tstep = tlimit[1]
        t = 0
        while (t*tstep+timeplus)<tlimit[0]:
            try:
                reso = solve_ivp(rhs_reg,[0,tstep],z1,'LSODA',args=((a,b,c,d,
                    diff,dx,res),))
                z1 = reso.y[:,-1]
                t = t+1
            except(MemoryError):
                fig, ax = plt.subplots(1,1,figsize = (1,1))
                plt.close()
        timeplus = timeplus+t*tstep
        fig, ax = plt.subplots(1,1,figsize = (5,5), dpi=150)
        ax.add_collection(patches_reg(xv,yv,z1))

        m = resolution*np.array([np.cos(np.deg2rad(phi)),np.sin(np.deg2rad(phi)
            )]) #directional vector
        a0 = np.array([50, 50]) #point on fitline, dev=0==ini
        ax.plot(np.vstack((a0-m,a0+m))[:,0], np.vstack((a0-m,a0+m))[:,1],ls='
            dotted',lw=2,label=r"$\vec{{g}}_0, d = {:.2f}$".format(0))
        mask = np.array(np.abs(z1-solutions[ini[0]])>tol) * np.array(np.abs(z1-
            solutions[ini[1]])>tol) #all points in front
        error, distance = vecfitter(xv.flatten()[mask],yv.flatten()[mask],phi)

        a1 = np.array([50+distance*np.cos(np.deg2rad(phi+90)), 50+distance*np.
            sin(np.deg2rad(phi+90))]) #point on fitline, dev=0==ini
        ax.plot(np.vstack((a1-m,a1+m))[:,0], np.vstack((a1-m,a1+m))[:,1],ls='
            dotted',lw=2,label=r"$\vec{{g}}_1, d = {:.2f}$".format(distance))
        ax.set_xlim(-.5,99.5)
        ax.set_ylim(-.5,99.5)
        ax.axis('off')
```

```
        plt.legend()
        plt.tight_layout()
        fig.savefig("13_reg{}_30u02^^_{}_{}_c{:.2f}_d{}_D{}_dx{}_t{:.3f}s.pdf".
            format(resolution,a,b,c,d,diff,dx,tlimit[0]))
        plt.show()
```

fig14.py

```
import matplotlib.pyplot as plt
import matplotlib as mpl
import os
from discrete_basics import *
from discrete_plot import *
from discrete_sim import *
from scipy.integrate import solve_ivp
from cycler import cycler
plt.rcParams['axes.prop_cycle']=cycler('color',['#1f77b4', '#ff7f0e', '#d62728'
    , '#9467bd', '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf'])
location = "orientation/"
try:
    os.makedirs(location)
except(FileExistsError):
    pass
tol = 0.1
a,b,c,d = -10, 2, 15, -5
solutions = minima(a,b,c,d)
resolution = 100
dx = 1.0
dataloc = location+"npz_a{}_b{}_c{}_d{}/".format(a,b,c,d)
try:
    os.makedirs(dataloc)
except(FileExistsError):
    pass
results = []

for method in [(mesh_tri, rhs_tri,3), (mesh_reg, rhs_reg,4), (mesh_hex, rhs_hex
    ,6)]:
    res, xv, yv = method[0](resolution)
    for coup in [2,4,6,8,10]:
        diff = coup*dx**2
        for i,j in [(0,1),(0,2),(1,2)]:
            for phi in np.arange(0,90.5,0.5):
                z = inistripe(xv,yv,resolution,phi,i,j)
                data = []
                print("{}_front u_{}{}, D = {}, phi = {}".format(method[0].
                    __name__[-3:],i,j,diff,phi), end="\r")
                try:
                    with np.load(dataloc+method[0].__name__[-3:]+"_D{}_dx{}_r{}
                        _phi{}_{}_t1.npz".format(diff,dx,resolution,phi,i+j))
                        as temp:
                        z1 = temp['arr_0'].flatten()
```

```
                except(FileNotFoundError):
                    z1 = z.flatten()
                    timeplus=0
                    for tlimit in [(0.1,0.001),(1,0.01)]:
                        tstep = tlimit[1]
                        t = 0
                        while (t*tstep+timeplus)<tlimit[0]:
                            try:
                                reso = solve_ivp(method[1],[0,tstep],z1,'LSODA'
                                    ,args=((a,b,c,d,diff,dx,res),))
                                z1 = reso.y[:,-1]
                                t = t+1
                            except(MemoryError):
                                fig, ax = plt.subplots(1,1,figsize = (1,1))
                                plt.close()
                        timeplus = timeplus+t*tstep
                    np.savez_compressed(dataloc+method[0].__name__[-3:]+"_D{}
                        _dx{}_r{}_phi{}_{}_t1".format(diff,dx,resolution,phi,i+
                        j),z1)

                N_1 = len(z1[np.abs(z1-solutions[i])<tol])
                N_2 = len(z1[np.abs(z1-solutions[j])<tol])
                mask = np.array(np.abs(z1-solutions[i])>tol) * np.array(np.abs(
                    z1-solutions[j])>tol)
                error, distance = vecfitter(xv.flatten()[mask],yv.flatten()[
                    mask],phi)

                results.append([method[2],coup,i+j,phi,N_1,N_2,np.sum(z1-z.
                    flatten()),distance, error])
results = np.asarray(results)
filename = "{}_{}_{}_c{:.2f}_d{}.txt".format(resolution,a,b,c,d)
np.savetxt(location+filename, results)

mpl.rcParams['backend'] = 'pdf'
mpl.rc('font',**{'family':'serif'})
mpl.rcParams['mathtext.fontset'] = 'cm'
plt.rc('text', usetex=True)
plt.rc('text.latex', preamble=r'\usepackage{amssymb,amsmath,wasysym}')

filename = "{}_{}_{}_c{:.2f}_d{}.txt".format(resolution,a,b,c,d)
data = np.loadtxt(location+filename)
data[:,3] = np.deg2rad(data[:,3])
fig, axs = plt.subplots(3, 3,figsize = (8,8), subplot_kw={'projection': 'polar'
    }, dpi=150)
solutions = minima(a,b,c,d)
for method in [('reg', 4, 0), ('hex', 6, 1), ('tri', 3, 2)]:
    mask1 = data[:,0] == method[1]
    for diff in [2.0,4.0,6.0,8.0,10.0]:#,15.0]:#
        mask2 = data[:,1] == diff
        for i in range(3):
            axs[i,0].set_rticks([-4,-3,-2,-1,0])
```

```
            axs[i,0].set_rlim(0,-3.75)
            axs[i,1].set_rticks([0.0,-0.3,-0.6,-0.9])
            axs[i,1].set_rlim(0,-0.8)
            axs[i,2].set_rticks([0,2,4,6])
            axs[i,2].set_rlim(0,6.5)
            for j in range(3):
                axs[j,i].set_thetamin(0)
                axs[j,i].set_thetamax(90)
                axs[j,i].set_xticks(np.deg2rad([0,15,30,45,60,75,90]))
                axs[j,i].xaxis.set_tick_params(pad=-2)
                axs[j,i].yaxis.set_tick_params(pad=-2)
        for ini in [1,2,3]:
            mask3 = data[:,2] == ini
            data1 = data[mask1*mask2*mask3]
            axs[method[2],ini-1].plot(data1[:,3],data1[:,7],label=r'$D={}$'.
                format(diff))
axs[2,1].legend(loc="lower center",bbox_to_anchor=(0., -0.4, 0.7, -0), ncol=5)
axs[0,0].set_ylabel('square')
axs[1,0].set_ylabel('hexagonal')
axs[2,0].set_ylabel('triangular')
axs[0,0].set_title(r'$u_{01}$')
axs[0,1].set_title(r'$u_{02}$')
axs[0,2].set_title(r'$u_{12}$')
fig.savefig("14_dx{}_r{}_angular_d.pdf".format(dx,resolution))
plt.show()
```