DOI: 10.1111/cgf.13828
Pacific Graphics 2019
C. Theobalt, J. Lee, and G. Wetzstein
(Guest Editors)

Volume 38 (2019), Number 7

ManyLands: A Journey Across 4D Phase Space of Trajectories

Aleksandr Amirkhanov,[1] Ilona Kosiuk,[2] Peter Szmolyan,[2] Artem Amirkhanov,[1]
Gabriel Mistelbauer,[3] M. Eduard Gröller,[1,4] Renata G. Raidou [1]

[1]Institute of Visual Computing & Human-Centered Technology, TU Wien, Austria [2]Institute for Analysis and Scientific Computing, TU Wien, Austria
[3]Otto-von-Guericke University Magdeburg, Germany [4]VRVis Research Center, Austria

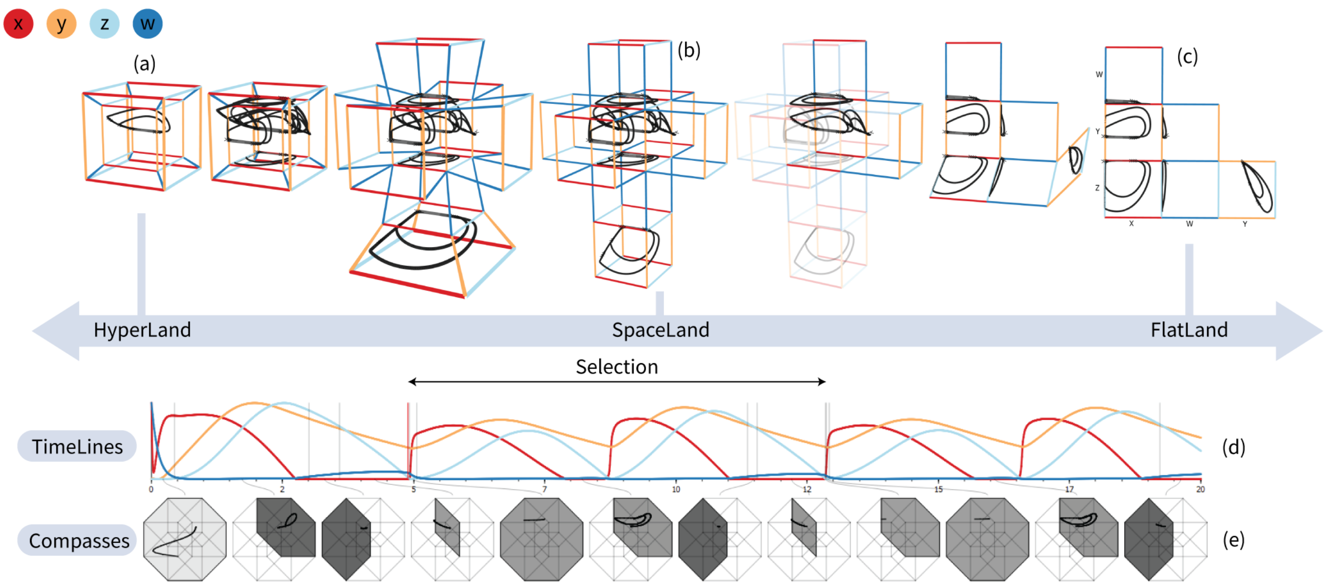

Figure 1: *ManyLands: a visualization application to support the exploration and analysis of 4D dynamical systems trajectories. The capabilities of ManyLands are based on interaction and smooth, animated navigation through phase space. We use the metaphor of traveling across (a) 4D HyperLand, (b) 3D SpaceLand, and (c) 2D FlatLand. Additionally, we employ (d) TimeLines to represent each variable of the dynamical system across time, and (e) compasses to facilitate the analysis of segments of the trajectories.*

Abstract

Mathematical models of ordinary differential equations are used to describe and understand biological phenomena. These models are dynamical systems that often describe the time evolution of more than three variables, i.e., their dynamics take place in a multi-dimensional space, called the phase space. Currently, mathematical domain scientists use plots of typical trajectories in the phase space to analyze the qualitative behavior of dynamical systems. These plots are called phase portraits and they perform well for 2D and 3D dynamical systems. However, for 4D, the visual exploration of trajectories becomes challenging, as simple subspace juxtaposition is not sufficient. We propose ManyLands to support mathematical domain scientists in analyzing 4D models of biological systems. By describing the subspaces as Lands, we accompany domain scientists along a continuous journey through 4D HyperLand, 3D SpaceLand, and 2D FlatLand, using seamless transitions. The Lands are also linked to 1D TimeLines. We offer an additional dissected view of trajectories that relies on small-multiple compass-alike pictograms for easy navigation across subspaces and trajectory segments of interest. We show three use cases of 4D dynamical systems from cell biology and biochemistry. An informal evaluation with mathematical experts confirmed that ManyLands helps them to visualize and analyze complex 4D dynamics, while facilitating mathematical experiments and simulations.

CCS Concepts

• ***Human-centered computing*** → ***Scientific visualization;*** *Visual analytics; Web-based interaction;*

1. Introduction

During the last decades, mathematical modeling and simulations have emerged as important resources to gain a better understanding of the function and dynamics of complex biological and biochemical processes. Guided by experiments, mathematical models of biological processes are formalized as large systems of *ordinary differential equations* (ODEs) describing the time evolution of biochemical species [KLW*16]. Important examples include dynamical systems describing glycolytic oscillations [KS10], i.e., a repetitive biochemical fluctuation in the concentration of metabolites, mathematical models for the cell division cycle [KS16], and models for the explanation of mechanisms behind bipolar disorder [Gol11].

The exploration and analysis of the mathematical behavior of these systems by mathematical domain scientists is anticipated to generate new knowledge on the underlying biological phenomena. This is especially significant for the development of new models that describe biological processes more accurately. Typical and pertinent mathematical questions of biological interest are the existence and stability of *equilibria, periodic oscillations, and switching phenomena* in the behavior of the investigated systems.

Temporal patterns and mathematical behaviors are analyzed using *dynamical systems theory*. Approaches within this area focus on a qualitative analysis of the behavior of dynamical systems, instead of seeking explicit computations of solutions. Geometric techniques like *phase space analysis* study the flow of dynamical systems and the geometry of the phase space, which depicts biologically meaningful solutions, or *trajectories* [Chi06, GH83]. Within the current workflow, *phase portraits*, i.e., representations of typical trajectories in phase space, are employed for 2D and 3D systems. A 2D phase portrait is shown in Figure 2.

Many biological processes are modeled by 4D dynamical systems [KS15]. For 4D systems, phase portrait analysis becomes challenging. The current technology adopted by mathematical domain experts involves generating and analyzing static 2D or 3D phase portraits—subspace projections of 4D dynamical system trajectories. This is done, e.g., in *MATLAB* [Lyn14], or *DYNAMICA* within *Wolfram Mathematica* [KM02]. The juxtaposition of the 2D or 3D phase portraits provides limited insight, as certain complex patterns can be revealed only in the 4D space. Besides, mathematicians sometimes generate their own hand-drawn illustrations to compensate for the limitations of the available technology, to navigate across subspaces, to demonstrate interesting phenomena in their systems, and to externalize their findings and proofs [DR96, KS11]. To link their hand-drawn illustrations to computer-generated 2D or 3D phase portraits and to reconstruct the 4D space, the experts rely on spatial imagination. The current workflow is depicted in Figure 3.

The *contribution* of this paper is the design and implementation of *ManyLands*. It is a web-based application that supports mathematical domain scientists in understanding and analyzing the mathematical behavior of biologically meaningful 4D dynamical system trajectories. It allows them to discover new knowledge within their systems and to illustrate their findings for externalization and education purposes. *ManyLands* offers an interactive and integrated workflow for exploring and analyzing 4D dynamical systems trajectories, as shown in Figure 3 in the rightmost column.

2. Background of 4D Biological Dynamical Systems

A *dynamical system* is characterized by the relationship between state variables and their time derivatives, i.e., it can be represented as a system of ODEs given by:

$$x' = f(x),\ x(0) = x_0,\ x \in \mathbb{R}^n. \tag{1}$$

This allows from a given initial state x_0 of the system to compute the future (and past) states according to the dynamical law f [Chi06,GH83]. Since it is usually impossible to derive an explicit formula for the solution of a nonlinear equation, *dynamical systems theory* has proven to be a powerful resource for understanding *qualitative features* of trajectories through *phase space analysis*. Examples of dynamical systems are shown in Section 6.1, and their mathematical descriptions are provided in our repository [AKS*].

Each of the n independent variables of the system defines a coordinate axis in the corresponding n-dimensional *phase space*. In the context of biological systems, we focus on 4D dynamical systems and their 4D phase space, while the variables x in Eq. 1 represent molecular concentrations (or numbers) within a cell, i.e., concentrations of enzymes or proteins. The *dynamics* of the system represent changes in these concentrations. If $x(t)$ is a solution of the system, $x(t_0)$ defines at each time t_0 a point in the phase space. As the point changes with time, the entire solution $x(t)$ traces out a curve, or *trajectory* in the phase space, as depicted in Figure 2. By this, solutions correspond to geometric objects in phase space. For instance, resting states (when variables do not change) correspond to equilibria points, and oscillations to so-called limit cycle curves. Nowadays, numerical computations of solutions are commonly used to analyze how the system states evolve over time. In particular, numerical simulations of solutions are used as a tool to guide theoretical phase space analysis.

Although all solutions could be mathematically interesting, there are several biological restrictions. For instance, there are no negative concentrations. Therefore, *only few, biologically meaningful trajectories* are explored and analyzed at the end. Observing the geometry of these trajectories in phase space using *phase portraits* allows mathematical domain scientists to understand important dynamics of their systems, and potentially, of the underlying modeled processes [Chi06, GH83]. Examples of dynamics arising in biology, i.e., changes in concentrations, include *switch-like behaviors* and *periodic oscillations*. For instance, the activity of a protein is “off” (inactive), i.e., the corresponding variable has a value that equals zero (or close to zero)—otherwise, it is “on” (active).

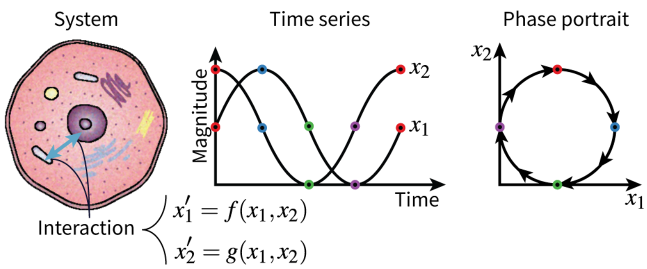

Figure 2: *Illustration of a 2D ODE system of a cellular interaction, the time series of its variables (x_1,x_2) and its phase portrait.*

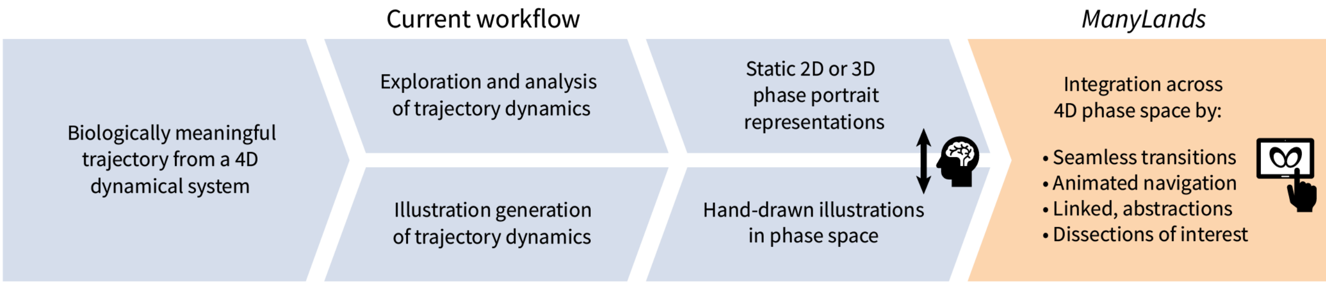

Figure 3: *Comparison of the current analysis workflow of 4D dynamical system trajectories vs. ManyLands. The current workflow (blue) requires mental linking between its components, whereas ManyLands (orange) provides mathematical domain scientists with an integrated solution, based on seamless transitions and animated navigation.*

3. Domain Requirements and Tasks

In Section 1, we discussed the current workflow of mathematical domain scientists and its limitations regarding the analysis of 4D dynamical systems. Mathematical domain scientists need strategies for completing the following tasks:

(T1) Holistic Analysis of 4D Trajectory Dynamics: As several phenomena can be only observed in the entire 4D phase space, domain scientists need to: *(i)* have an overview of the entire space; *(ii)* detect alternations in trajectory behaviors across subspaces; *(iii)* explore and analyze where, how, and why these occur; *(iv)* drill down to subspaces. Instead of using static 2D or 3D phase portraits, they require an integrated approach to *navigate and transit across the entire dimensionality of the phase space* **(T1.a)**. Additional *features*, e.g., slow vs. fast behavior, need to be discovered and highlighted for further exploration **(T1.b)**.

(T2) Dissected Analysis of 4D Trajectory Dynamics: Several phenomena can only be observed in distinct subspaces of the trajectories or in smaller segments thereof. As the behavior of the trajectories is not always known a priori, domain scientists require a mechanism to drill down interactively to a *dissected analysis view* of the phase space that localizes segments of interest **(T2.a)**. This needs to be summarized in a *comprehensive representation* **(T2.b)**, where *navigation* is possible **(T2.c)**.

(T3) Integration of Knowledge Discovery with Generation of Phase Space Illustrations and Animations: In the current workflow, mathematical domain experts often use manual illustrations for investigation, education, or externalization purposes. To reduce the manual and mental effort of domain scientists, functionality for *meaningful illustrations and animations* needs to be integrated with the interactive trajectory computation and representation.

4. Related Work

Several previous works have addressed the exploration of dynamical systems [AS83, GWM*96, WLG97] or objects with a dimensionality higher than three [HIM99]. All these approaches propose to employ multiple subspace projections or additional encodings on the 3D space to incorporate the fourth dimension. For more than three dimensions, parallel coordinates [Ins85] have been employed by Wegenkittl et al. [WLG97] and Grottel [GHWG14]. None of these approaches can fully provide the required functionality of a combined holistic and dissected exploration and analysis of the 4D phase space, as well as the ability to integrate knowledge discovery with the generation of illustrations and animations.

To tackle a dimensionality higher than 3D, previous work has been conducted also concerning discrete data—*rolling the dice* [EDF08] being the most relevant one. It is an interactive method for the exploration of multidimensional data through queries. It is based on a scatterplot matrix that provides an overview, while supporting interactive navigation in the multidimensional space by animated transitions. Animated transitions have been investigated in several other works for the exploration of discrete data [FFT88, RMC91, VWVH*07]. Still, our focus is on strategies for transiting between continuous data—in particular, approaches that employ *slicing* through high-dimensional data, approaches that employ *projections*, and *hybrids* thereof.

HyperSlice [vWvL93] is the first method to depict multiple two-dimensional slices in a trellis plot for the visualization of high-dimensional functions. This is done around a point of interest, common to all 2D slices. The approach is local in nature—similar to other multidimensional strategies, such as the *grand tour* [Asi85]. It requires to repeatedly probe the function, in order to simulate a global view on the data. *Hypermoval* [PBK10] is another slicing approach, which is designed with a strong focus on modeling validation. This is done with 2D or 3D projections of the involved high-dimensional scalar functions around a point of interest. In all cases, only a localized view is provided.

As opposed to slicing approaches, several methods provide a global view on the functions, such as *continuous scatterplots* [BW08], *continuous parallel coordinates* [HW09], or *profile contour plots* [HKC14]. Here, density fields are mapped to either 2D or 3D representations. However, in our case, we are interested in the function itself—not in density fields. Other global views include focus+context approaches like *PolarEyez* [JN02]. Reduction techniques, such as the *HyperCell* [dSB02], take lower-dimensional representations as basis for the visualization of high-dimensional spaces. In *HyperCell*, the user determines 1D to 3D cells, i.e., subspaces, where operations like switching, rotating, and brushing are possible. In the *Worlds Within Worlds* approach [FB90], nested heterogeneous coordinate systems of lower dimensions enable to view and manipulate high-dimensional functions in arbitrary, nested, interactive boxes. Another projection-based method is proposed by Nouanesengsy et al. [NSSV09], where the high-dimensional space is projected onto 2D plots, showing changes of the different parameters. In all these cases, selected projections with reduced dimensionality are offered as static views to the user. An overview on the entire dimensionality, and an easy transition between the represented subspaces is not possible.

To balance between local and global approaches, Torsney-Weir et al. [TWSM17] propose the *Sliceplorer*, where 1D line plots constitute the central view to show changes in a single dimension. By probing through the function and showing all slices in a projection, they obtain a hybrid that offers both global (through projections) and local (through slicing) views on the underlying data. Later, they propose *Hypersliceplorer* [TWMSK18], which works with 2D slices of multi-dimensional shapes. Yet, transitions between subspaces and the generation of illustrations and animations are still not feasible. These, though, are very important in the application of biological dynamical systems for understanding the mathematical behavior and specific phenomena in the trajectories. Other hybrid categories include *exploded view diagrams* [KLMA10], while pure topological approaches [CSA03, GBPW10, CLB11, BP18] are also available, but out of the scope of this work.

Concerning interaction, our work revolves around seamless transitions [TMB02, HR07] and multiple coordinated views [BC87, BMMS91, Wil08]. Our domain experts intend to interact with 4D trajectories in phase space and visually inspect them through different views. However, our domain scientists require to perform their visual analysis in many different subspaces and their intermediate transition stages. Having multiple coordinated views only would not be sufficient, given the limited screen space. Therefore, we provide a different interaction concept based on animated navigation and seamless, continuous transitions—related to the work of Jianu et al. [JDL09], Miao et al. [MDLI*18] and Sorger et al. [SMR*17].

5. The Design of *ManyLands*

We propose *ManyLands* for the exploration and analysis of precomputed 4D trajectories from continuous-time dynamical systems of biological interest. The concepts in *ManyLands* have been inspired by the book *Flatland: a Romance of Many Dimensions* [Abb84]. The mathematical domain scientist—as another *Square*—visits and navigates across all *Lands*, i.e., subspaces, and their respective dimensions. We introduce three different *Lands*: 4D *HyperLand*, 3D *SpaceLand*, and 2D *FlatLand*. We also include an additional linked diagram, called *TimeLines*. The design of *ManyLands* takes into consideration the framework proposed by Gleicher [Gle16] regarding comprehensibility in modeling.

The initial view in *ManyLands* shows the entire 4D dimensionality of the system in *HyperLand* (Figure 1a). Represented as a tesseract, it shows the morphology of the 4D mathematical system in a Schlegel diagram [Som29]. From the 4D space, the domain expert can smoothly travel to all lower-dimensional *Lands*, which are built upon a polytope concept. A space with n dimensions ($n > 0$) is composed of a number $E_{m,n}$ of m-dimensional ($n > m$) subspaces [Cox73], given by:

$$E_{m,n} = 2^{n-m}\binom{n}{m}, \tag{2}$$

where $\binom{n}{m} = \frac{n!}{m!(n-m)!}$ and $n!$ denotes the factorial of n. A tesseract ($n = 4$) based on Eq. 2 contains eight cubes ($m = 3$) and twenty-four squares ($m = 2$). When considering symmetries and rotations in space redundant, the decomposition of a tesseract will result in four unique cubes and six unique squares. This decomposition of subspaces is shown in Figure 4, where redundancies have been omitted only for the squares in (a), and for cubes and squares in (b). From the *HyperLand* (Section 5.1), the domain scientists can smoothly travel to *SpaceLand*, by unfolding the tesseract into its corresponding 3D cubes (Figure 1b, Section 5.2). They can further unfold the cubes into a representation consisting of a matrix of 2D plots, called *FlatLand* (Figure 1c, Section 5.3). *TimeLines* is an additional linked diagram that shows the graphs of all the variables of the dynamical system over the time axis t (Figure 1d, Section 5.4). The navigation across *Lands* is achieved with smooth, animated transitions, motivated by the nature of the data and the tasks. *TimeLines* are linked to the *Lands* through brushing and linking [BC87, BMMS91, Wil08] (Section 5.5). We show how *ManyLands* support the domain expert in their workflow (Sections 5.6–5.8).

5.1. *HyperLand*

Given the mathematical problems investigated by our domain scientists, we need a basis for a single-view analysis of 4D trajectories, which as a whole explores all phase space dimensions. To this end, we created *HyperLand*. In *HyperLand*, the trajectory of a 4D dynamical system is displayed inside a tesseract (Figure 1a). To display a 4D object on a 2D screen, this object must be first projected into a 3D space—from 4D to 3D space, and then to 2D space.

4D to 3D projection: The projection from 4D to 3D space is similar to the one from 3D to 2D space. To project a 4D object to 3D space, each vertex of the object must be multiplied by the word matrix, the view matrix, and the projection matrix. Both the word matrix and the view matrix can be defined by multiplications of rotation and translation matrices. In 4D space, objects can be rotated around six planes—namely, the XY-, YZ-, XZ-, XW-, YW-, ZW-planes. For instance, the rotation in homogeneous coordinates around the XY-plane is given by the matrix:

$$R_{XY} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 & 0 & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3}$$

where α denotes the rotation angle. The rotation matrices around the other planes are defined analogously. The translation in 4D space is given by the matrix:

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 & dx \\ 0 & 1 & 0 & 0 & dy \\ 0 & 0 & 1 & 0 & dz \\ 0 & 0 & 0 & 1 & dw \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{4}$$

where dx, dy, dz, dw denote the translation distances along x, y, z, and w axes, correspondingly. By applying a 4D projection matrix to the object, we obtain the result in the 3D space. The 4D projection

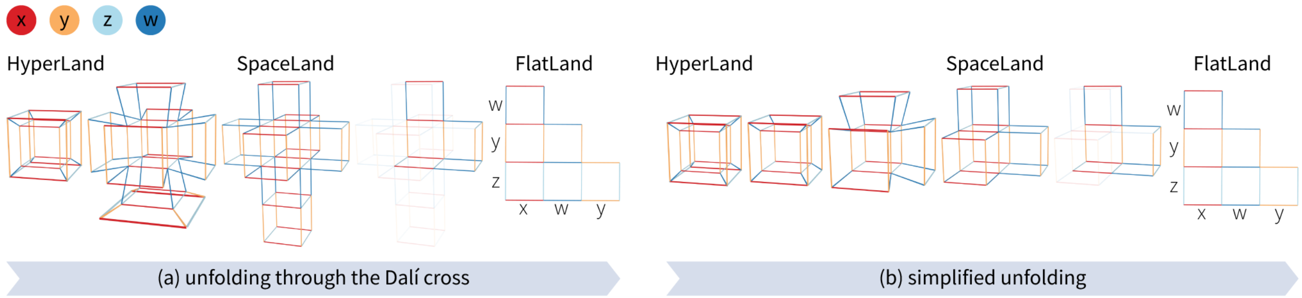

Figure 4: *Two strategies for smooth, animated transitions between HyperLand, SpaceLand and FlatLand. In (a) 3D symmetries are preserved, but 2D symmetries are omitted, while in (b) 3D and 2D symmetries and rotations are disregarded.*

is given by the matrix:

$$P_{4D} = \begin{pmatrix} \frac{n}{r} & 0 & 0 & 0 & 0 \\ 0 & \frac{n}{t} & 0 & 0 & 0 \\ 0 & 0 & \frac{n}{d} & 0 & 0 \\ 0 & 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix}, \quad (5)$$

where n denotes the near clipping plane; f is the far clipping plane; and r, t, d are the dimensions of the truncated pyramid frustum (width, height, and depth).

3D to 2D projection: The object is subsequently projected into 2D, by a conventional 3D-to-2D perspective projection matrix. The 3D projection parameters do not necessarily repeat corresponding parameters in the 4D projection. For example, the 4D and 3D cameras can have different viewing angles, positions, and clipping planes.

Encoding: To encode the perspective distortion of a 4D object in the thickness of the employed primitives, we use 3D objects, such as cylinders and spheres, for rendering the scene (Figure 1). We assign a unique Deuteranopia-safe color to each of the axis-aligned edges of the tesseract based on their orientation in the 4D space to denote the four variables of the system (Figure 1). As requested by the mathematical domain scientists, the axes are labeled according to the domain convention, namely x, y, z, and w for the four variables, but the labels can be changed by the user if required. Axes scaling can be incorporated in the scene, although in most of the cases all variables have already undergone *nondimensionalization* [VGM07] and they are dimensionless. By standardizing the proportions of the tesseract, we provide the user with a familiar depiction of the trajectories. The user can inspect the object in 4D and 3D space with decoupled rotation operations. The 3D rotations are performed in a conventional way through mouse interaction, while 4D rotations are performed using sliders in the interface.

5.2. *SpaceLand*

For 3D trajectory analysis, mathematical domain scientists use widely 2D and 3D phase portraits of their models. *SpaceLand* consists of three dimensions and contains 3D subspace projections from the *HyperLand* (Figure 1b). It is formed by unfolding the bounding tesseract of *HyperLand* and the trajectory into a 3D space by a smooth transition between spaces, as it will be described in Section 5.5. There are two alternative representations of *SpaceLand*, between which the user can switch. In one representation, the mathematical model is shown with eight 3D cubes, containing all eight projections, i.e., subspaces, of the 4D dynamical system (Figure 4a), as discussed in the description of Eq. 2. These plots are connected at their faces. All together, they form a so-called *Dalí Cross*. The name is due to a resemblance to the cross in the artwork *Corpus Hypercubus* [Dal54] of Salvador Dalí. When unfolding *HyperLand* to *SpaceLand*, the resulting *Dalí Cross* contains symmetric cases, which represent redundantly the same 3D spaces—analogously to unfolding a 3D cube into its 2D faces. A second representation excludes symmetric plots from the *Dalí Cross* and consists of only four unique 3D cubes (Figure 4b). Transiting directly from 4D to the *reduced SpaceLand* might be intuitively challenging [HR07], as we discuss in Section 5.5. We, therefore, leave it up to the users to select between a *Dalí Cross* or a *reduced SpaceLand*. We further enable a focus+context approach [CMS99]. To make the transition between *HyperLand* and *SpaceLand* smooth and intuitive, all encodings, e.g., color encodings and labels, are preserved [HR07].

5.3. *FlatLand*

Static 2D phase portraits are conventionally used by mathematical domain scientists, as shown in Figure 2. The tesseract representation of *HyperLand* consists of twenty-four 2D faces. As discussed in the description of Eq. 2, omitting redundancies results into six unique 2D faces. A contiguous arrangement of the unique faces of all 3D cubes produces *FlatLand*, as the result of a smooth unfolding transition from *SpaceLand* (Figure 1c), following the principles of Section 5.5. This layout represents all combinations of 2D subspaces in a *trellis plot* configuration [Cle93]. To preserve the mental model between the spaces [HR07], the *FlatLand* representation inherits all visual encodings from the previous *Lands*.

5.4. *TimeLines*

Apart from phase portraits, mathematical domain scientists also use representations that illustrate the temporal evolution of biological system variables (Figure 2c). The view showing the values of the trajectory variables vs. time is called *TimeLines* (Figure 1d). Obtaining *TimeLines* with a smooth transition from *FlatLand* cannot be achieved with simple unfolding operations that do not interfere with our perception [HR07]. A possibility would be to adopt mechanisms, such as unfolding and transiting the trajectories in *FlatLand* to one-dimensional subspaces, similar to the 1D slices of

Sliceplorer [TWSM17]. This, however, would violate significant principles of animation [TMB02, HR07], as it will be discussed in Section 5.5. Therefore, we decided to have *TimeLines* as a separate static, but linked view and to employ transitions only between the 4D, 3D, and 2D subspaces. *TimeLines* are linked to the other *Lands* through brushing and linking [BC87,BMMS91] and inherit also all visual encodings from them. An additional advantage of the linked *TimeLines* view will become obvious in Section 5.7. There, *TimeLines* are employed to dissect, explore and analyze local system features, supporting the completion of task **(T2)**.

5.5. A Journey Across *Lands*

A straightforward approach to explore all available subspaces would be to use multiple coordinated views [WBWK00]. In our case, this would require a significant amount of screen space. For example, for the simple case of a 4D system, four different views would be required. We would need one view for the 4D subspace, one for the four 3D subspaces, one for the six 2D subspaces—excluding symmetries and rotations—and one for the *TimeLines*. This would also require a significant amount of mental effort, e.g., when tracking the temporal development of a 4D trajectory. Given the temporal evolution of the systems, the varying dimensionality along segments of the trajectories and the interest in localizing where the behavior of the trajectories alternates, we employ smooth transitions between the different subspaces. Smooth, animated transitions are of particular importance for tracking and understanding the current state of the system, and at the same time constitute the interactions between the system variables.

Although animation has been often considered to be a controversial topic [BB99, HR07, RFF*08], animated transitions can be beneficial for providing insights, especially for tracking—as long as the principles of congruence and apprehension are respected [TMB02, HR07]. According to the *congruence principle* [HR07], the structure and content of an animation must correspond to the desired structure and content of the mental representation. To this end, intermediate interpolation stages, e.g., between *Lands*, retain valid data graphics. Also, we use consistent mappings and visual encodings by having standardized transitions across graphic types. We respect correspondence and avoid ambiguity across transitions, e.g., for the axes labeling and coloring, and the trajectory attributes. According to the *apprehension principle* [HR07], the structure and content of the animation must be readily and accurately perceived and comprehended. To this end, we group similar transitions together and minimize occlusion, e.g., of redundant spaces, during transitions. Our transitions remain predictable and simple, but when more complex transitions are needed, such as in the *SpaceLand* to *FlatLand* transition, we use staging, i.e., grouping. The speed of transitions in *ManyLands* is at any case user-controlled. For a smooth transition between the 4D, 3D, and 2D subspaces, the user interacts with a simple slider that can be moved across different subspace configurations and their intermediate stages. The slider is annotated to facilitate traveling from one *Land* to another one, and slider snapping to *Land* positions is implemented. Other standard interactions, i.e., rotating, zooming, and panning, are possible in all *Lands*, but always independently from transitions. Hereby, we discuss all *Land* transitions.

***HyperLand* to *SpaceLand*:** *HyperLand* initially morphs into a *Dalí Cross* representation, which morphs, subsequently, into a *reduced SpaceLand* (Figure 4a). Without the intermediate *Dalí Cross* (Figure 4b), a sudden transition would be less intuitive and less easy to follow, as suggested by the apprehension principle [TMB02, HR07]. Still, the users can hide the symmetric subspaces within the *Dalí Cross*, and can travel directly to the *reduced SpaceLand* (Figure 4b). The transition employs staging, in accordance with the congruence and apprehension principles. First, the trajectory is replicated several times according to the number of the involved subspaces (eight cubes for the *Dalí Cross* and four for the *reduced* case). We need this to move each instance of the trajectory into the corresponding 3D cube. Then, each trajectory instance is projected into the respective cube. This projection is animated, so that the user can track the movement of the trajectory. Afterward, each 3D cube is rotated around one of its faces in the direction that will bring the w-coordinate of each cube to a constant value. During this unfolding, the 4D camera must be moved into a position where the camera points towards the w-axis. So far, all operations are performed only in 4D space. Thus, during the animated unfolding, the dynamical system data are shown unaltered. After the animated unfolding, all vertices of the tesseract are lying on the same 3D space, i.e., their w coordinates are equal. Therefore, the w coordinate is discarded and the object is projected into 3D space.

***SpaceLand* to *FlatLand*:** The transition between *SpaceLand* and *FlatLand* is performed in three stages (Figure 4). First, symmetrical 3D cubes are faded out in the animation, to remove redundant information. Second, the 3D trajectories are replicated into several instances and projected onto the faces of the 3D cubes (Figure 1). Third, the faces are rotated into the image plane. During this rotation, the 3D camera is also orthogonally aligned, providing a distortion-free view of *FlatLand* (Figure 1c).

Animations showing all transition stages in detail are available in our repository [AKS*], and in the supplementary material.

5.6. Holistic Analysis of 4D Trajectory Dynamics (T1)

For the holistic exploration and analysis of the trajectory behavior, domain scientists need an overview of the entire 4D phase space and its subspaces to be able to better observe certain phenomena. This is also important for the detection of alternations in trajectory behaviors across subspaces, and exploring where, how, and why these occur. A thorough exploration of all subspaces and the ability to drill down to each one of them is also required. Our approach for solving these subtasks involves primarily the ability to transition smoothly between *Lands* **(T1.a)**. This is done as described in Sections 5.1–5.5. Apart from the design of the *Lands* and the ability to transition between them, we need to convey further information about the dynamical system **(T1.b)**. For example, we need to convey the presence of *slow-fast dynamics*, i.e., a pronounced phenomenon during which some variables are changing faster than others, the *local dimensionality*, or *switching phenomena*. This provides mathematical domain scientists with important knowledge on how and where to start their analysis. We use an intuitive strategy, where we dissect the trajectory based on one of these three manifestations. The slow-fast behavior, i.e., the velocity vector at each point along the trajectory is given by the right-hand side of the

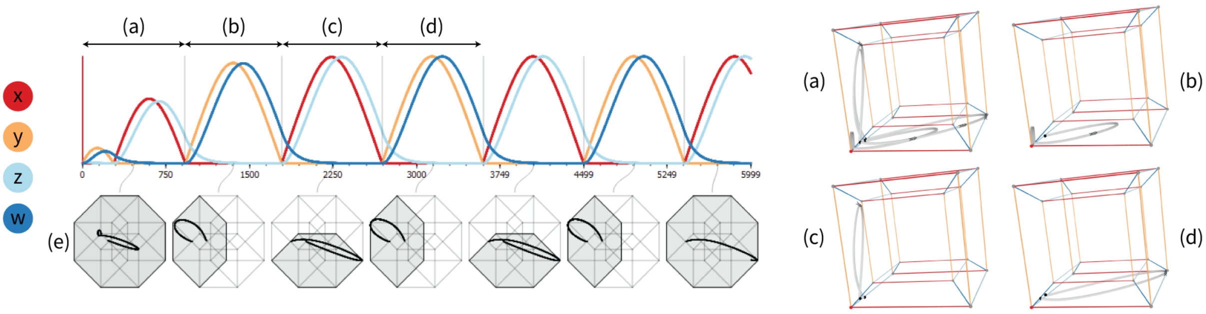

Figure 5: *Depiction of the selection mechanism using brushing and linking between TimeLines and the Lands of ManyLands. Here, we use a HyperLand example. Each selection in TimeLines, denoted with (a), (b), (c), (d) is highlighted in HyperLand to denote the respective segment in 4D. Underneath TimeLines, we depict the phase space compasses (e). These are pictograms, representing respective segments (and their trajectory characteristics) from the dissection of TimeLines.*

ODE system in Eq. 1. The dimensionality is the number of active (non-zero) variables at each trajectory segment. The derivation of switching behaviors is not straightforward, as it is often computed for each model individually. We, therefore, give an initial suggestion to the user based on changes in the slope along the trajectory.

Subsequently, we encode this information directly on the trajectory with colors and glyphs. For the *color encoding of the switches* or *the slow-fast behavior*, we use a logarithmic, perceptually uniform, greyscale colormap [BHH03]. The colormap is truncated on the white side to not interfere with the background color of the scene. The greyscale choice is due to the presence of other colors in the scene, e.g., to encode the axes. We chose this encoding as an "inking" metaphor: the slower a point is moving on the trajectory, the darker is its ink footprint. The logarithmic scaling is used to compensate for the—sometimes—extremely fast or slow motion along the trajectory. For the *glyph encoding of the dimensionality*, we use a corresponding number of arrows, e.g., for a 2D segment, we show two arrows. These two visual encodings are used across all *Lands* to preserve semantics, and demonstrated in the *HyperLands* of Figure 5. A fly-through probe that travels along the trajectory indicates the temporal evolution of a trajectory, similar to the colored dots employed in Figure 2. In the *Lands*, this is a sphere, and in *TimeLines*, a line sliding across time.

The combination of smooth transitions across *HyperLand*, *SpaceLand*, and *FlatLand* with a bi-directionally interactively linked view on the *TimeLines* enables mathematical domain scientists to perform task **(T1)**. They are now able to effectively interact and transit between different visual representations of the dynamical system. The smooth transition between the various *Lands* and link to *TimeLines* provides consistent information across the different subspaces that conveys to the user a holistic view on the entire space of the 4D trajectory.

5.7. Dissected Analysis of 4D Trajectory Dynamics (T2)

The localized exploration and understanding of dynamical systems trajectories and their underlying biological processes is also important. Several phenomena may be observed only in smaller or lower-dimensional regions, i.e., trajectory segments. To provide a strategy for the easy localization, exploration, and analysis of the dynamical system representation across subspaces and trajectory segments, we extend the *TimeLines* to be used as a selection mechanism for subspaces/segments of interest.

As mentioned also for the first task, we can dissect the trajectories represented in *TimeLines* into segments, depending on their dimensionality, slow-fast or switching behavior **(T2.a)**. With trajectory dissections available, we propose an additional representation—the *phase space compasses*, or simply, *compasses*. These are abstracted, small-multiple tesseract pictograms [DBT88] that depict the active variables (and relevant subspaces) of the corresponding trajectory segment [Cle93]. Each individual *compass* is, in essence, an orthogonal projection of the tesseract to the screen space, given by the isometric projection matrix:

$$P = \begin{pmatrix} 1 & 0 & \cos(\alpha) & -\cos(\beta) & 0 \\ 0 & -1 & \sin(\alpha) & \sin(\beta) & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{6}$$

where α and β are the orientations of z and w axes in screen space.

The *compasses* **(T2.b)** are configured in a trellis representation, under the respective segments of the *TimeLines* (Figure 5e). If a segment of a trajectory is inactive in one of its four dimensions, then it "lives" in a 3D domain, and a *compass* denoting the respective 3D subspace is created (Figure 5b–d). A trajectory segment can be mapped together with its characteristics on such a *compass*, using the same visual encodings as within the *Lands*. This is shown in Figures 1e and 5e in greyscale. Hovering over the *compasses* provides a magnified view, to enhance visibility, e.g., when there are too many segments or when the display size is small. Selecting a *compass* provides a focus+context view on the respective segment of the trajectory and a smooth, continuous transition to the respective subspace **(T2.c)**. Afterward, through the interaction described in Section 5.6, the domain scientist can navigate across other subspaces of the segment. Arbitrary selections of segments on the *TimeLines* are possible within and across all *Lands*, using brushing and linking [BC87, BMMS91, Wil08] (Figure 5). The abstracted representation of dissections with *compasses* offers easier navigation across subspaces and segments. It also facilitates interaction and the selection of trajectory segments.

5.8. Integration of Knowledge Discovery with Generation of Phase Space Illustrations and Animations (T3)

The final task for mathematical domain scientists is to generate meaningful, scientific illustrations for further analysis, for education and externalization of their findings, or as a support to their mathematical proofs. These illustrations are usually based on their knowledge of the dynamical systems. Now, they can integrate the newly generated knowledge from the visual analysis, as performed in **(T1–2)**, with the generation of illustrations in the web-based platform of *ManyLands*. We employ a flexible web-based design space, where users can specify the appearance of their representations. Rotations, alterations of the rendering scene, and aesthetic changes to the representations are feasible, with the use of simple, interactive sliders. The results of our illustration generation functionality can be seen throughout all the figures of this paper. Analogously, animations can be generated.

6. Results and Evaluation

After the design of *ManyLands*, we conducted an informal evaluation with four domain scientists, following the guidelines of Isenberg et al. [IIC*13]. The domain scientists are researchers in applied mathematics, working on the exploration of 4D dynamical systems of biological interest. Their field experience varies from *low* (student with few years in the field) to *very high* (professor with >30 years in the field). Two of them have normal and two normal-to-corrected vision. One of the participants had been actively involved in the design of *ManyLands*. At the beginning of the evaluation session, we introduced *ManyLands* to the domain scientists, and we demonstrated the basic functionality and main components of the framework. Afterward, the domain experts proceeded with the investigation of three use cases. A visual environment for the exploration and analysis of the respective 4D dynamical systems was simulated. We used the think-aloud method, allowing the experts to comment and discuss visualizations and potential insights. Discussions among themselves were allowed, like in a real-life collaborative scenario. By providing *ManyLands* as a web-based application, domain scientists could interact with the systems in real-time and generate custom illustrations. All three tasks of Section 3 were executed, allowing domain experts to reason about the employed visual encodings, interaction mechanisms, and potential findings (Section 6.1). Subsequently, our domain experts completed individually a questionnaire to give us feedback on their experience with *ManyLands* (Section 6.2).

6.1. Use Cases

In this section, we present three use cases for the exploration and analysis of 4D trajectories with *ManyLands*. The dynamical systems and initial conditions of these cases were provided by the evaluation participants, based on what phenomena they are currently interested in. These are documented in detail in the supplementary material and in our repository [AKS*]. For the investigation of the trajectories, the domain scientists employed *ManyLands*, as they would do in a real-life analysis scenario. The findings presented in this section have been documented by the domain scientists. Given the focus of a visualization paper, these are not meant for direct inferences on the dynamical systems (even more, on the underlying biological systems)—they are, rather, exemplifications of whether (and how) *ManyLands* enables them to conduct their analysis.

Use Case A—Analysis of Bipolar Disorder: The Goldbeter model of bipolar disorder [Gol11] is explored in the first use case. It is a 4D system of ODEs describing the dynamics of bipolar disorder, i.e., the alternations between manic and depressive state. For certain initial conditions, the model exhibits oscillatory behavior, i.e., mania and depression alternate periodically. Moreover, the four variables of the system change over different time scales, as shown in Figure 6f. The first straightforward observation is that the system exhibits switch-like oscillations, as seen in the *TimeLines* (Figure 6f). By initializing the *HyperLand* and *SpaceLand* view, the corresponding trajectory is presented in the phase space (Figure 6a–b). The color encoding of the trajectory remains mostly grey and does not show big differences in the speed of the trajectory. Starting with a certain initial condition, the orbit converges to the so-called limit cycle, which can be seen easily when moving to *FlatLand* with the "loop-like" geometry (Figure 6c). This phenomenon is already visible in *HyperLand* and *SpaceLand* (Figure 6a–b). Extracting the segments of the trajectories indicated in the *TimeLines* (Figure 6f) and inspecting the respective *compasses* shows also the repetitive pattern of the closed trajectory (Figure 6d–e). This confirms that the observed oscillations are of limit cycle type. The analysis of the bipolar model trajectories in *ManyLands* revealed the switch-like nature of the model dynamics and confirms results obtained from a prior geometric analysis.

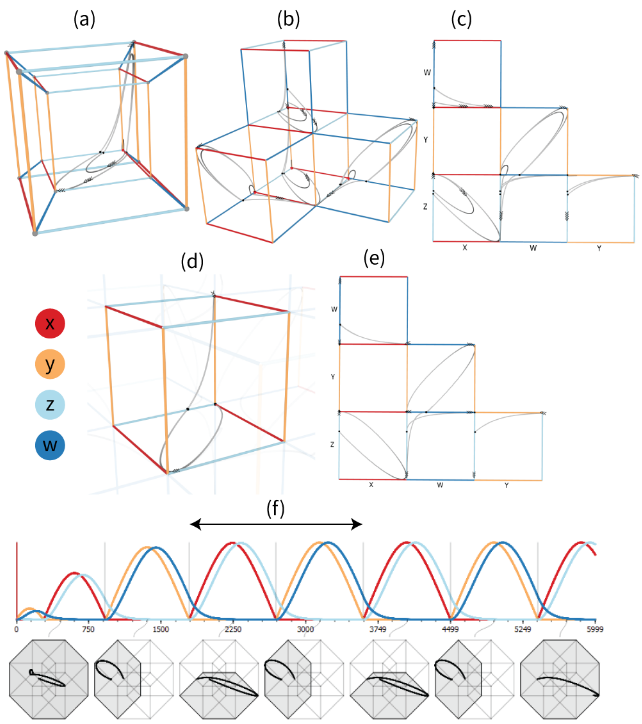

Figure 6: *Analysis conducted by the domain scientists for a trajectory, describing bipolar disorder [Gol11]. With (a)–(f), we denote the visual analysis steps, as described in Section 6.1, Use Case A.*

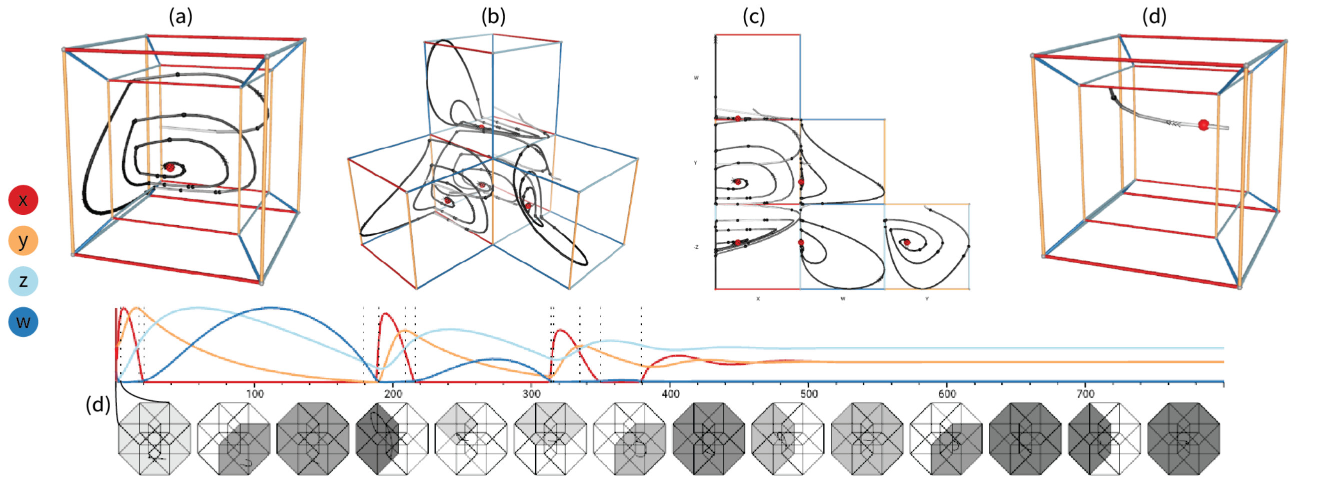

Figure 7: *Analysis conducted by the domain scientists for a trajectory, describing the NF-κB signaling pathway [KJS06]. With (a)–(e), we denote the visual analysis steps of Section 6.1, Use Case B.*

Use Case B—Analysis of NF-κB Pathway: A trajectory from the mathematical model describing the NF-κB signaling pathway [KJS06] is explored in the second use case. The NF-κB system plays a key role in regulating our immune response to infection. When the NF-κB system is activated, the model exhibits damped oscillations for certain initial conditions. This means that the transcription factor NF-κB is located in and out of the nucleus in a periodic fashion, and finally stays in the cytoplasm, i.e., the system settles at a stable equilibrium point. In *ManyLands*, two numerically computed solutions for two different initial conditions have been analyzed simultaneously. The solutions exhibit a transition towards the stable equilibrium point, as shown in *HyperLand*, *SpaceLand*, and *Flatland* (Figure 7a–c, marked with the red dots). This transition happens in a switch-like manner, i.e., the variable x changes its values from non-zero to zero in a repeated fashion and eventually remains at a nonzero constant value, as shown in *TimeLines* (Figure 7e). Also, the *compasses* (Figure 7e) show that some parts of the oscillatory phenomena can be observed in specific subspaces, e.g., the fast (light grey) behavior at the beginning of one trajectory "lives" in 4D (Figure 7d). The application of *ManyLands* to the NF-κB model enabled the geometric analysis of its complicated 4D dynamics—also, for two solutions. In particular, the dissections in *TimeLand* support and complement the mathematical model dissection to segments of interest in phase space.

Use Case C—Analysis of Peroxidase−Oxidase Reaction: The Olsen model for the peroxidase−oxidase reaction [KS15] is explored in the third case. The Olsen model is a chaotic attractor system, for which domain scientists often look only at specific combinations of the ZW, XZW and XYW spaces. The 4D view of *HyperLand* (Figure 8a) can confirm at a glance the chaotic attractor behavior, as denoted by the multitude of loops. By transiting to *FlatLand*—in particular, to the ZW space—the trajectory comes repeatedly close to the origin and spends longer intervals of time there (Figure 8b). With focus+context on *SpaceLand*, the XZW (Figure 8c) and XYW spaces (Figure 8d) are illustrated. In XZW, the system is a chaotic attractor, while in XYW it is a transient. Practically, the system exhibits a slow motion and makes a long excursion in the $XYZW$ space before returning to $(Z,W) = (0,0)$. Additionally, the initial part of *TimeLines* shows changes in speed (Figure 8e). The second part (Figure 8f) manifests periodic behavior, while in-between a switching behavior can be also noticed (Figure 8g). *ManyLands* facilitated the analysis of the complex 4D dynamics of the Olsen model, supporting a geometric analysis of the chaotic attractor and revealing its oscillatory behavior.

In the three explored cases, the transitions from one space to another **(T1)** and the interactive dissection with the compasses **(T2)** have proven to be very helpful for the analysis and understanding of the mathematical behavior of the systems, as well as for the creation of the aesthetic, scientific illustrations of Figures 6–8 **(T3)**.

6.2. User Experience

For the evaluation of the user experience, we designed a questionnaire. The *first part* was related to tasks **(T1-3)**. Each question required an open answer, and a grading in a Likert scale ($++$ to $--$) for the perceived effectiveness, efficiency, and satisfaction. In Figure 9, we summarize the results of the evaluation. One of the participants was involved in the design of *ManyLands*—and these results are presented separately. The visualizations received positive grades for all tasks. Tasks **(T1.b)** and **(T3)** received neutral grades ($=$) by the most experienced domain scientist. For task **(T1.b)**, the reason is that they would like to map additional features onto the trajectory. For task **(T3)**, the scientist commented that this functionality should be tested also with other systems, where there is no prior understanding of the dynamics—as opposed to now, where the dynamics were already known. The most appreciated functionalities were the animated, smooth transitions and the facilitated interaction for trajectory exploration.

In the *second part*, the participants were asked to compare *ManyLands* to what they are currently using and to evaluate the overall usefulness of our tool—including strengths, weaknesses, limitations and future improvements. They commented that *"DYNAMICA allows visualizations only in 3D and 2D. ManyLands gives new possibilities and access into 4D"*. All participants agreed that the visual tool is overall understandable and useful, and that they

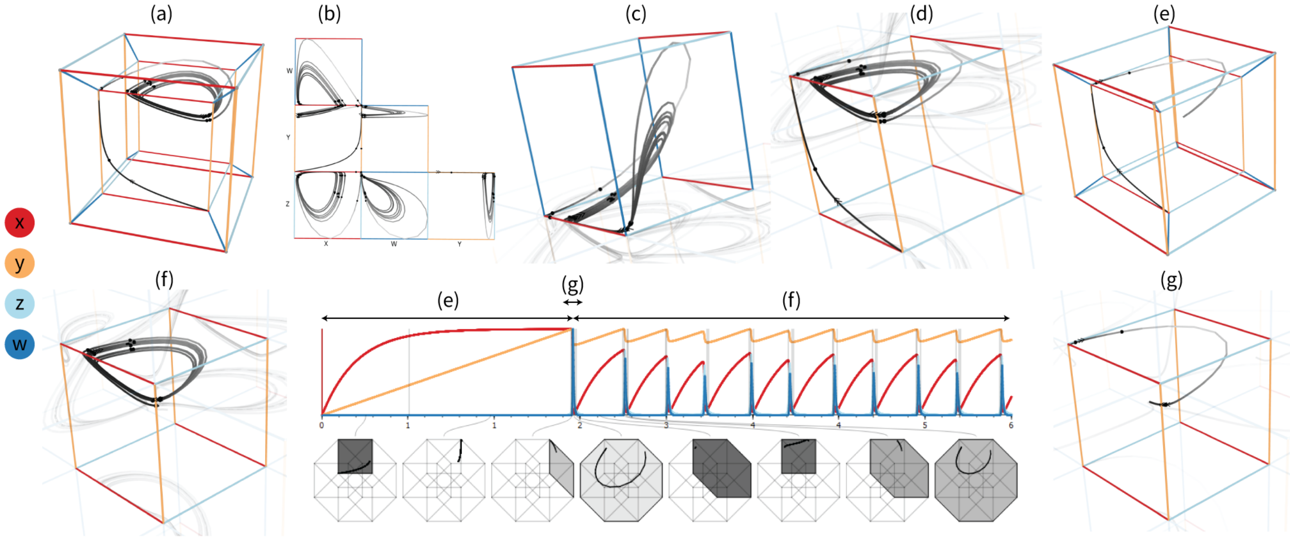

Figure 8: *Analysis conducted by the domain scientists for a trajectory, describing the peroxidase−oxidase reaction [KS15]. With (a)–(g), we denote the visual analysis steps, as described in Section 6.1, Use Case C.*

would be able to use it after training: *"ManyLands already helps. It will help even more after reasonable training, due to the ease of switching between representations"*. One participant stated that *"it helps to understand the underlying geometry of the phase space better, it contributes to the understanding of the model dynamics and supports a geometric mathematical analysis of a given system"*. One experienced participant summarized their overall opinion as: "*ManyLands has a user-friendly interface for rapid re-analysis and visualization. By experimenting with different initial conditions and parameter values, the trajectory plot will update accordingly; this process will assist the development of better understanding of model dynamics. In contrast to other dynamical systems software, it is a straightforward tool for phase space visualization that allows not only to show numerically computed trajectories on a tesseract, but shows also the time evolution of variables. When needed, all possible variants of 2D and 3D phase portraits are shown at once. While dynamical systems knowledge is essential, ManyLands can make a proficient domain scientist more productive, while empowering students with limited visual mathematical/geometrical skills.*". The potential of *ManyLands* is, according to the evaluators, the integrated ability to transition across *Lands*, while interactively inspecting *TimeLines*. Improvement suggestions were related to redefining approximations to reflect better the mathematical descriptions, while one participant suggested that saving viewpoints and snapping back to them would be valuable.

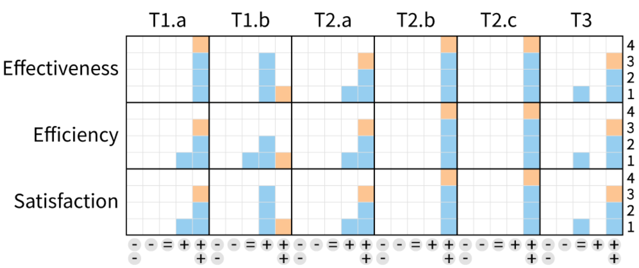

Figure 9: *Evaluation outcomes for user experience. Each task **(T1–3)** has been graded according to its perceived effectiveness, efficiency, and satisfaction on a 5-points Likert scale (−− to ++). We denote the domain expert that was involved in the design of ManyLands with orange squares, and the others with blue squares.*

6.3. Discussion

Our evaluation and the use cases conducted by the mathematical domain experts showed that *ManyLands* introduces new capabilities into the workflow of phase space exploration and analysis of 4D trajectories. This positive outcome is partially because our application has specifically targeted the requirements of Section 3. The introduction of the 4D representation and the use of smooth, animated transitions were proposed by our domain scientists as a suitable solution to their current issues with the use of 2D and 3D phase portraits. For this reason, and as discussed in Section 5.5, we did not investigate further or compare to other obvious alternatives, such as multiple coordinated views [BC87]. Additionally, the visualization literacy of our intended users has also been considered. If *ManyLands* were to be extended to other application domains, an additional evaluation considering all potential alternatives should be conducted. In this case, scalability beyond 4D systems and beyond a small number of trajectories should also be regarded. In the present context, this was considered out of scope by our domain scientists. Scalability to higher-dimensional systems is currently not relevant given their particular focus on 4D systems of biological interest. Considering a vast number of trajectories is also neither biologically justified nor mathematically relevant for them. The trajectories relate to specific initial conditions reflecting biological setups. Therefore, looking into all possible solutions is not necessary, and only in few cases, a small number of them (2-3) might be investigated simultaneously. Parameter sensitivity analysis has also not been investigated, for the same reasons. We propose these topics for future work in Section 7.

7. Conclusions and Future Work

Observing the geometry of dynamical systems is anticipated to provide mathematical domain scientists with a significant understanding of the dynamics, and potentially, of underlying modeled biological processes. Mathematical experts are now restricted to the detached use of computer-generated 2D or 3D phase portraits and of hand-drawn illustrations, hampering an integrated analysis of system trajectories. *ManyLands* is a new visualization tool to support domain scientists to understand, analyze, and illustrate trajectories of 4D dynamical systems that describe biological processes.

There are several promising avenues for future research. Most importantly, the scalability of our proposed approach must be investigated with strategies to accommodate dynamical systems beyond 4D, and the comparison of multiple trajectories. These two topics were currently not relevant for our collaborating domain scientists as they are not encountered in their applications. However, they are ubiquitous in many other application domains, e.g., in the physical and environmental sciences. To support systems higher than 4D, combining *ManyLands* with *rolling the dice* [EDF08] would facilitate querying and navigating the entire n-dimensional ($n > 4$) phase space. Alternatively, the use of non-linear embeddings [MH08] could also be investigated. To compare multiple trajectories emerging from different initial conditions, techniques from ensemble visualization could be incorporated, e.g., contour boxplots [WMK13] and curve boxplots [MWK14], or approaches from weather forecast ensemble visualization [FKRW16, FKRW17]. Other directions include the smart incorporation of additional topological features in the representations and improvements regarding visual perception, e.g., by depth-dependent halos [EBRI09], or by illumination and shadowing [EHS13]. *ManyLands* has opened interesting directions for the exploration of the phase space, enabling the analysis of 4D trajectories of systems that model biological processes.

Acknowledgment: This work has been partially financed by the Visual Analytics for Realistic and Aesthetic Smile Design (SmileAnalytics) project, supported by the Austrian Research Promotion Agency (FFG) project no. 861168.

References

[Abb84] ABBOTT E. A.: *Flatland: a Romance of Many Dimensions*. Seeley & Co, London, 1884. 4

[AKS*] AMIRKHANOV A., KOSIUK I., SZMOLYAN P., AMIRKHANOV A., MISTELBAUER G., GRÖLLER E., RAIDOU R.: Repository of ManyLands. `https://amirkhanov.net/manylands/`. 2, 6, 8

[AS83] ABRAHAM R. H., SHAW C. D.: *Dynamics-The geometry of behavior*. Aerial Press, 1983. 3

[Asi85] ASIMOV D.: The grand tour: a tool for viewing multidimensional data. *SIAM journal on Scientific and Statistical Computing 6*, 1 (1985), 128–143. 3

[BB99] BEDERSON B. B., BOLTMAN A.: Does animation help users build mental maps of spatial information? In *Proceedings of the Symposium on Information Visualization* (1999), IEEE, pp. 28–35. 6

[BC87] BECKER R. A., CLEVELAND W. S.: Brushing scatterplots. *Technometrics 29*, 2 (1987), 127–142. 4, 6, 7, 10

[BHH03] BREWER C. A., HATCHARD G. W., HARROWER M. A.: Colorbrewer in print: a catalog of color schemes for maps. *Cartography and Geographic Information Science 30*, 1 (2003), 5–32. 7

[BMMS91] BUJA A., MCDONALD J. A., MICHALAK J., STUETZLE W.: Interactive data visualization using focusing and linking. In *Visualization* (1991), IEEE, pp. 156–163. 4, 6, 7

[BP18] BALLESTER-RIPOLL R., PAJAROLA R.: Visualization of High-dimensional Scalar Functions Using Principal Parameterizations. *ArXiv e-prints* (Sept. 2018), arXiv:1809.03618. 4

[BW08] BACHTHALER S., WEISKOPF D.: Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1428–1435. 3

[Chi06] CHICONE C.: *Ordinary differential equations with applications*, vol. 34. Springer Science & Business Media, 2006. 2

[CLB11] CORREA C., LINDSTROM P., BREMER P.-T.: Topological spines: A structure-preserving visual representation of scalar fields. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 1842–1851. 4

[Cle93] CLEVELAND W. S.: *Visualizing data*. Hobart Press, 1993. 5, 7

[CMS99] CARD S. K., MACKINLAY J. D., SHNEIDERMAN B.: Using vision to think. In *Readings in information visualization* (1999), Morgan Kaufmann Publishers Inc., pp. 579–581. 5

[Cox73] COXETER H. S. M.: *Regular polytopes*. Courier Corporation, 1973. 4

[CSA03] CARR H., SNOEYINK J., AXEN U.: Computing contour trees in all dimensions. *Computational Geometry 24*, 2 (2003), 75–94. 4

[Dal54] DALÍ S.: Crucifixion (Corpus Hypercubus), 1954. Oil on canvas. Metropolitan Museum of Art, New York. 5

[DBT88] DI BATTISTA G., TAMASSIA R.: Algorithms for plane representations of acyclic digraphs. *Theoretical Computer Science 61*, 2-3 (1988), 175–198. 7

[DR96] DUMORTIER F., ROUSSARIE R.: *Canard cycles and center manifolds*, vol. 577. American Mathematical Society, 1996. 2

[dSB02] DOS SANTOS S. R., BRODLIE K. W.: Visualizing and investigating multidimensional functions. In *Proceedings of the Symposium on Data Visualisation* (2002), Eurographics Association, pp. 173–182. 3

[EBRI09] EVERTS M. H., BEKKER H., ROERDINK J. B., ISENBERG T.: Depth-dependent halos: Illustrative rendering of dense line data. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1299–1306. 11

[EDF08] ELMQVIST N., DRAGICEVIC P., FEKETE J.-D.: Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1539–1148. 3, 11

[EHS13] EICHELBAUM S., HLAWITSCHKA M., SCHEUERMANN G.: LineAO–Improved three-dimensional line rendering. *IEEE Transactions on Visualization and Computer Graphics 19*, 3 (2013), 433–445. 11

[FB90] FEINER S. K., BESHERS C.: Worlds within worlds: Metaphors for exploring n-dimensional virtual worlds. In *Proceedings of the 3rd annual ACM SIGGRAPH Symposium on User Interface Software and Technology* (1990), ACM, pp. 76–83. 3

[FFT88] FISHERKELLER M. A., FRIEDMAN J. H., TUKEY J. W.: Prim-9: An interactive multidimensional data display and analysis system. *Dynamic Graphics for Statistics* (1988), 91–109. 3

[FKRW16] FERSTL F., KANZLER M., RAUTENHAUS M., WESTERMANN R.: Visual analysis of spatial variability and global correlations in ensembles of iso-contours. In *Computer Graphics Forum* (2016), vol. 35, pp. 221–230. 11

[FKRW17] FERSTL F., KANZLER M., RAUTENHAUS M., WESTERMANN R.: Time-hierarchical clustering and visualization of weather forecast ensembles. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 831–840. 11

[GBPW10] GERBER S., BREMER P.-T., PASCUCCI V., WHITAKER R.: Visual exploration of high dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1271–1280. 4

[GH83] GUCKENHEIMER J., HOLMES P.: *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Springer, New York, 1983. 2

[GHWG14] GROTTEL S., HEINRICH J., WEISKOPF D., GUMHOLD S.: Visual analysis of trajectories in multi-dimensional state spaces. In *Computer Graphics Forum* (2014), vol. 33, pp. 310–321. 3

[Gle16] GLEICHER M.: A framework for considering comprehensibility in modeling. *Big Data 4*, 2 (2016), 75–88. 4

[Gol11] GOLDBETER A.: A model for the dynamics of bipolar disorders. *Progress in biophysics and molecular biology 105*, 1-2 (2011), 119–127. 2, 8

[GWM*96] GRÖLLER E., WEGENKITTL R., MILIK A., PRSKAWETZ A., FEICHTINGER G., SANDERSON W. C.: The geometry of wonderland. *Chaos, Solitons & Fractals 7*, 12 (1996), 1989–2006. 3

[HIM99] HANSON A. J., ISHKOV K. I., MA J. H.: Meshview: Visualizing the fourth dimension. *Technical report. Indiana University.* (1999). 3

[HKC14] HALL K. W., KUSALIK P. G., CARPENDALE S.: Profile contour plots: Alternative projections of 3D free energy surfaces. *In IEEE Conference on Scientific Visualization (SciVis) – Poster* (2014). 3

[HR07] HEER J., ROBERTSON G.: Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1240–1247. 4, 5, 6

[HW09] HEINRICH J., WEISKOPF D.: Continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 6 (2009), 1531–1538. 3

[IIC*13] ISENBERG T., ISENBERG P., CHEN J., SEDLMAIR M., MÖLLER T.: A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (Dec. 2013), 2818–2827. 8

[Ins85] INSELBERG A.: The plane with parallel coordinates. *The visual computer 1*, 2 (1985), 69–91. 3

[JDL09] JIANU R., DEMIRALP C., LAIDLAW D.: Exploring 3D DTI fiber tracts with linked 2D representations. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1449–1456. 4

[JN02] JAYARAMAN S., NORTH C.: A radial focus+context visualization for multi-dimensional functions. In *Visualization* (2002), IEEE, pp. 443–450. 3

[KJS06] KRISHNA S., JENSEN M. H., SNEPPEN K.: Minimal model of spiky oscillations in NF-κB signaling. *Proceedings of the National Academy of Sciences 103*, 29 (2006), 10840–10845. 9

[KLMA10] KARPENKO O., LI W., MITRA N., AGRAWALA M.: Exploded view diagrams of mathematical surfaces. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1311–1318. 4

[KLW*16] KLIPP E., LIEBERMEISTER W., WIERLING C., KOWALD A., HERWIG R.: *Systems biology: a textbook*. John Wiley & Sons, 2016. 2

[KM02] KULENOVIC M. R., MERINO O.: *Discrete dynamical systems and difference equations with Mathematica*. Chapman and Hall/CRC, 2002. 2

[KS10] KOSIUK I., SZMOLYAN P.: Geometric Desingularization in Slow-Fast Systems with Application to the Glycolytic Oscillations Model. *AIP Conference Proceedings 1281*, 1 (2010), 235–238. 2

[KS11] KOSIUK I., SZMOLYAN P.: Scaling in singular perturbation problems: blowing up a relaxation oscillator. *SIAM Journal on Applied Dynamical Systems 10*, 4 (2011), 1307–1343. 2

[KS15] KUEHN C., SZMOLYAN P.: Multiscale geometry of the Olsen model and non-classical relaxation oscillations. *Journal of Nonlinear Science 25*, 3 (2015), 583–629. 2, 9, 10

[KS16] KOSIUK I., SZMOLYAN P.: Geometric analysis of the Goldbeter minimal model for the embryonic cell cycle. *Journal of Mathematical Biology 72*, 5 (Apr 2016), 1337–1368. 2

[Lyn14] LYNCH S.: *Dynamical systems with applications using Matlab®*. Springer, 2014. 2

[MDLI*18] MIAO H., DE LLANO E., ISENBERG T., GRÖLLER M. E., BARIŠIĆ I., VIOLA I.: DimSUM: Dimension and scale unifying map for visual abstraction of DNA origami structures. In *Computer Graphics Forum* (2018), vol. 37, pp. 403–413. 4

[MH08] MAATEN L. V. D., HINTON G.: Visualizing data using t-sne. *Journal of machine learning research 9*, Nov (2008), 2579–2605. 11

[MWK14] MIRZARGAR M., WHITAKER R. T., KIRBY R. M.: Curve boxplot: Generalization of boxplot for ensembles of curves. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2654–2663. 11

[NSSV09] NOUANESENGSY B., SEOK S.-C., SHEN H.-W., VIELAND V. J.: Using projection and 2D plots to visually reveal genetic mechanisms of complex human disorders. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)* (2009), IEEE, pp. 171–178. 3

[PBK10] PIRINGER H., BERGER W., KRASSER J.: Hypermoval: Interactive visual validation of regression models for real-time simulation. In *Computer Graphics Forum* (2010), vol. 29, pp. 983–992. 3

[RFF*08] ROBERTSON G., FERNANDEZ R., FISHER D., LEE B., STASKO J.: Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008). 6

[RMC91] ROBERTSON G. G., MACKINLAY J. D., CARD S. K.: Cone trees: animated 3D visualizations of hierarchical information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1991), ACM, pp. 189–194. 3

[SMR*17] SORGER J., MINDEK P., RAUTEK P., GRÖLLER E., JOHNSON G., VIOLA I.: Metamorphers: storytelling templates for illustrative animated transitions in molecular visualization. In *Proceedings of the 33rd Spring Conference on Computer Graphics* (2017), ACM, p. 2. 4

[Som29] SOMMERVILLE D.: *Introduction to the Geometry of N Dimensions*. E.P.Dutton, 1929. 4

[TMB02] TVERSKY B., MORRISON J. B., BETRANCOURT M.: Animation: can it facilitate? *International Journal of Human-Computer Studies 57*, 4 (2002), 247–262. 4, 6

[TWMSK18] TORSNEY-WEIR T., MÖLLER T., SEDLMAIR M., KIRBY R. M.: Hypersliceplorer: Interactive visualization of shapes in multiple dimensions. In *Computer Graphics Forum* (2018), vol. 37, pp. 229–240. 4

[TWSM17] TORSNEY-WEIR T., SEDLMAIR M., MÖLLER T.: Sliceplorer: 1d slices for multi-dimensional continuous functions. In *Computer Graphics Forum* (2017), vol. 36, pp. 167–177. 4, 6

[VGM07] VAN GROESEN E., MOLENAAR J.: *Continuum modeling in the physical sciences*. SIAM Mathematical Modeling and Computation, 2007. 5

[VWVH*07] VIEGAS F. B., WATTENBERG M., VAN HAM F., KRISS J., MCKEON M.: Manyeyes: a site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007). 3

[vWvL93] VAN WIJK J. J., VAN LIERE R.: Hyperslice. In *Visualization* (1993), IEEE, pp. 119–125. 3

[WBWK00] WANG BALDONADO M. Q., WOODRUFF A., KUCHINSKY A.: Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (2000), ACM, pp. 110–119. 6

[Wil08] WILLS G.: Linked data views. In *Handbook of data visualization*. Springer, 2008, pp. 217–241. 4, 7

[WLG97] WEGENKITTL R., LÖFFELMANN H., GRÖLLER E.: Visualizing the behavior of higher dimensional dynamical systems. In *Visualization* (1997), IEEE Computer Society Press, pp. 119–125. 3

[WMK13] WHITAKER R. T., MIRZARGAR M., KIRBY R. M.: Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2713–2722. 11