Using Numerical Continuation Methods and Galerkin's Method for Finding and Tracing Periodic Solutions in Nonlinear Dynamic Systems

Fabian Späh Mats Bosshammer

SS 2016

Contents

1 Introduction

Finding periodic orbits in dynamic systems has applications in engineering. This project's goal is the implementation and evaluation of methods for this task. This document aims at explaining the methods, techniques and ideas used in a concise and connected manner.

The general problem can be described as follows. Given a system of differential equations $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x})$, find solutions $\mathbf{y}$ such that $\mathbf{y}(t+T) = \mathbf{y}(t)$ for all $t \in \mathbb{R}$. Finding closed-form representations of these solutions is impossible in general. We thus resort to numerical methods for the task.

The tool used for the task of judging whether a given solution is a correct periodic solution is *Galerkin*'s method. This is a form of collocation method, allowing to convert a continuous operator equation to a system of equations. Solving this possibly non-linear system yields a periodic solution.

Given a periodic solution for a system smoothly depending on a parameter, if the parameter varies only slightly, so will the solution. This can be used to track solutions over a broad range of the parameter. Projecting the found periodic orbits to a simpler object (for example a set of scalars) allows to plot them against the parameter in a bifurcation diagram. This allows to evaluate the global behavior of the system. Being able to dynamically create bifurcation diagrams for a given system is a goal of this project.

2 Periodic Orbits as Solutions of System of Equations, the Galerkin Operator

Given a system of $n \in \mathbb{N}$ possibly non-linear, autonomous, ordinary, first-order differential equations $\mathbf{x}$

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}), \tag{1}$$

we are interested in numerically computed, periodic solutions. That is, solutions $\mathbf{y} : \mathbb{R} \to \mathbb{C}^n$ which obey $\mathbf{y}(t) = \mathbf{y}(t + T)$ for all $t \in \mathbb{R}$ and some period $T \in \mathbb{R}$. This is the general case, as differential equations of any degree can be converted to a system of first-order differential equations.

Model Solution candidates need to be modeled in a certain way. The periodicity constraint suggests using a multidimensional trigonometric polynomial of degree $m \in \mathbb{N}$

$$\mathbf{y} \coloneqq \sum_{k=-m}^{m} \mathbf{y}_k \exp\left(i\omega k t\right),$$

where $\mathbf{y}_k \in \mathbb{C}$, $\omega = \frac{2\pi}{T}$. Solution candidates of this form satisfy $\mathbf{y}(t) = \mathbf{y}(t + T)$ by definition.

Optimality Criterion Finding good solutions, that is, functions $\mathbf{y}$ which at least approximate $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$, requires a way of measuring the quality of a solution candidate. In this case, Galerkin's method takes this role. A useful property of the trigonometric polynomial is, that it can be trivially differentiated

$$\frac{d\mathbf{y}}{dt} = \sum_{k=-m}^{m} i\omega k \mathbf{y}_k \exp\left(i\omega k t\right). \tag{2}$$

Employing this property in the definition of the differential equation system yields

$$\begin{aligned} &\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}) \\ \Leftrightarrow\ &\mathbf{f}(t, \mathbf{y}) - \frac{d\mathbf{y}}{dt} = 0 \\ \Leftrightarrow\ &\mathbf{f}(t, \mathbf{y}) - \sum_{k=-m}^{m} i\omega k \mathbf{y}_k \exp\left(i\omega k t\right) = 0. \end{aligned}$$

The difference between these two functions is called the *residual* $\mathbf{r}(t) \coloneqq \mathbf{f}(t, \mathbf{y}) - \frac{d\mathbf{y}}{dt}$. A candidate $\mathbf{y}$ is a solution if and only if $\mathbf{r} = 0$. The solution of the system of differential equations will generally not be representable by a trigonometric polynomial. The residual can thus not genuinely equal zero.

Galerkin's Method Galerkin's method relaxes the equality requirement such that only projections onto a set of so called *trial vectors*, need to vanish. This is equivalent to requiring a projection of $\mathbf{r}$ onto the subspace spanned by the trial vectors to be zero. Choosing the complex oscillations as a basis for this subspace as well is a solid choice. Basically the residual is approximated by a trigonometric polynomial and a solution is required to only minimize this representation. There are other factors supporting this choice: The residual is periodic as well, because of orthogonality many terms can cancel each other out, it allows us to employ the FFT for many operations, and the resulting system of equations is almost balanced.

System of Equations This yields $N = 2m + 1$ equations, one for each trial vector $\mathbf{v}_k = \exp(i\omega k t)$ for $-m \leq k \leq m$

$$\begin{aligned} &\langle \mathbf{r}, \mathbf{v}_k \rangle \\ &= \langle \mathbf{f}(t, \mathbf{y}), \mathbf{v}_k \rangle - \left\langle \frac{d\mathbf{y}}{dt}, \mathbf{v}_k \right\rangle \\ &= \langle \mathbf{f}(t, \mathbf{y}), \mathbf{v}_k \rangle - i\omega k \mathbf{y}_k. \end{aligned} \tag{3}$$

The last step exploits that the complex exponential functions are orthonormal. For these N equations, there are $N + 1$ variables: N unique coefficients and ω. This represents the situation, that at this point there is still one degree of freedom: Each phase shifted version of a solution is still a solution. We thus introduce another generic equation called the *anchor* equation, which basically chooses one of these solutions. In this case $\langle \mathbf{y}(0), (\delta_{1i})_{i \in \mathbb{N}_n} \rangle = 0$ is used: For $t = 0$, the solution needs to intersect the hyperplane defined by being zero in the first component. This can be formulated by requiring the corresponding coefficients to sum up to zero. The anchor equation needs to be adapted to the system considered: If there are no intersections with this plane, another equation needs to be chosen.

Discretization When transferring these theoretical constructs to a practical setting, the main change is that solutions $\mathbf{y}$ are not considered to be functions of continuous time, but vectors representing discrete time. Consequently, solution candidates are represented by linear combinations of discrete vectors as well

$$\mathbf{y} = \left(\sum_{k=0}^{N-1} \mathbf{y}_k \exp\left(i \frac{2\pi}{N} jk \right) \right)_{0 \leq j \leq N-1}.$$

This is no real limitation at that point, as solution candidates are band limited by construction. Define the *DFT matrix* $\mathbf{F} \in \mathbb{R}^{N \times N}$ and the coefficient vector $\mathbf{Y} \in (\mathbb{C}^n)^N$

$$\begin{aligned} \mathbf{F} &\coloneqq N^{-1} \left(\exp\left(-i \frac{2\pi}{N} jk \right) \right)_{0 \leq j,k \leq N-1} \\ \mathbf{Y} &\coloneqq (\mathbf{y}_k)_{0 \leq k \leq N-1}. \end{aligned}$$

$\mathbf{Y}$ is a vector of vectors, while somewhat unusual, this allows for more concise notation later on, than notation as a matrix. Using these constructs allows us to reformulate

$$\mathbf{y} = \mathbf{F}^{-1}\mathbf{Y}.$$

Let $\mathbf{K} = \operatorname{diag}([0 : m] \,||\, [-m : 1])$, where $||$ represents concatenation and the $[a : b]$ notation is shorthand for an integer vector going from $a \in \mathbb{N}$ to $b \in \mathbb{N}$. Considering only the equations from projections onto trial vectors, and considering ω to be known the system of equations becomes

$$\mathbf{F}\mathbf{f}\{\mathbf{F}^{-1}\mathbf{Y}\} - i\omega\mathbf{K}\mathbf{Y} = 0,$$

where $\mathbf{f}$ defines the system of differential equations (see Equation 1).

Compare this to Equation 3. Essentially, $\mathbf{f}\{\mathbf{F}^{-1}\mathbf{Y}\}$ is a sampled version of $\mathbf{f}(t, \mathbf{y})$. Multiplication by $\mathbf{F}$ corresponds to calculating the inner products with the trial vectors, which are the rows of $\mathbf{F}$. The term $i\omega\mathbf{K}\mathbf{Y}$ corresponds to Equation 2, focusing on the coefficients only.

Introducing these new constructs might at first glance appear unnecessarily complicated, because the system of equations was already known, and could be used in the state it was in. However, an implementation in the basic form would have required different treatment for each dynamic system, while in this form, it is only trivially dependent of $\mathbf{f}$ and the degree of the trigonometric polynomial. While knowing the system of equations is important, the core task of finding a good solution requires the Jacobian of the system. The large advantage is, that from this form the system can be derived in a general way.

Deriving the System Let D denote a differential operator. The only variable in the system is $\mathbf{Y}$, thus

$$\begin{aligned} & D\left(\mathbf{F}\mathbf{f}\{\mathbf{F}^{-1}\mathbf{Y}\} - i\omega\mathbf{K}\mathbf{Y}\right) \\ = \mathbf{F}&\frac{d\mathbf{f}\{\mathbf{Y}\}}{d\mathbf{Y}}\left(\mathbf{F}^{-1}\mathbf{Y}\right)\mathbf{F}^{-1}\cdot D\mathbf{Y} - i\omega\mathbf{K}\cdot D\mathbf{Y}, \end{aligned} \tag{4}$$

where $\frac{d\mathbf{f}\{\mathbf{Y}\}}{d\mathbf{Y}}$ is a vector-by-vector derivative, which yields a matrix. However, because both vector's elements are vectors themselves, the entries of the matrix are again vector-by-vector derivatives. This results in an $N \times N$ matrix of matrices with

$$\left(\frac{d\mathbf{f}\{\mathbf{Y}\}}{d\mathbf{Y}}\right)_i^j = \frac{d}{d\mathbf{Y}_j}\mathbf{f}(\mathbf{Y}_i) = \delta_{ij}\frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}}(\mathbf{Y}_i),$$

where $\frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}}$ is the Jacobian of $\mathbf{f}$. At no point different cells of $\frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}}$ interact. Hence, the expression for the Jacobian of the whole system can be considered separately for each partial derivative in the Jacobian of $\mathbf{f}$. For $k, l \in \mathbb{N}_n$ define the $N \times N$ diagonal matrix which extracts a single partial derivative

$$\mathbf{A}_{kl}(\mathbf{Y}) = \operatorname{diag}\left(\left(\frac{\partial f_k}{\partial x_l}(\mathbf{Y}_i)\right)_{i\in\mathbb{N}_N}\right).$$

For each k and l Equation 4 then becomes

$$\left(\mathbf{F}\mathbf{A}_{kl}(\mathbf{F}^{-1}\mathbf{Y})\mathbf{F}^{-1} - i\delta_{kl}\omega\mathbf{K}\right)\cdot D\mathbf{Y} \tag{5}$$

This expression is again free of nested vectors.

Because $\mathbf{A}_{kl}(\mathbf{F}^{-1}\mathbf{Y})$ is diagonal, $\mathbf{A}_{kl}(\mathbf{F}^{-1}\mathbf{Y})\mathbf{F}^{-1}$ is the rows of $\mathbf{F}^{-1}$ each multiplied by a constant. This can also be seen as an element wise multiplication of the columns $\mathbf{c}_k$ of $\mathbf{F}^{-1}$ by a column vector $\mathbf{d}$ defined by the diagonal of $\mathbf{A}_{kl}(\mathbf{F}^{-1}\mathbf{Y})$

$$\begin{aligned} \mathbf{c}_k &\coloneqq \left(\exp\left(i\frac{2\pi}{N}jk\right)\right)_{0\le j\le N-1} \\ \mathbf{d} &\coloneqq \frac{\partial f_k}{\partial x_l}\{\mathbf{Y}\}. \end{aligned}$$

A column of $\mathbf{F}\mathbf{A}_{kl}(\mathbf{F}^{-1}\mathbf{Y})\mathbf{F}^{-1}$ is then the transform of the product of two signals $\mathcal{F}(\mathbf{d}\cdot\mathbf{c}_k)$. Because of the structure of $\mathbf{c}_k$, this corresponds to a simple periodic shift of the Fourier coefficients of the discrete signal $\mathbf{d}$ by k steps. Using this for every column of the complete term yields

$$\mathbf{F}\mathbf{A}_{kl}(\mathbf{F}^{-1}\mathbf{Y})\mathbf{F}^{-1} = \left(\mathcal{F}(\mathbf{d})_{((i-j))_N}\right)_{0\le i,j\le N},$$

that is, a simple circulant matrix.

Rebuilding the complete derivative from these matrices is a matter of subtracting the remaining diagonal term and merging the results. The differential operator D has been left undefined and Equation 5 most of the term is independent of the choice of D. This is important, because in an optimization setting real and imaginary parts of the Fourier coefficients need to be treated separately. However, all these cases can be reduced to $D = \frac{d}{d\mathbf{Y}}$

$$\begin{aligned} \frac{d\Re}{d\Re\mathbf{Y}} &= \Re\frac{d}{d\mathbf{Y}} & \frac{d\Re}{d\Im\mathbf{Y}} &= -\Im\frac{d}{d\mathbf{Y}} \\ \frac{d\Im}{d\Re\mathbf{Y}} &= \Im\frac{d}{d\mathbf{Y}} & \frac{d\Im}{d\Im\mathbf{Y}} &= \Re\frac{d}{d\mathbf{Y}}. \end{aligned}$$

3 Finding Periodic Solutions

A separate problem from evaluating, optimizing and continuing periodic solutions is finding an initial candidate solution. Because continuation is a crucial part of this project, having just one single periodic solution might enable to find many others, through tracing and switching solution branches. Because the systems considered in this work have stable periodic solutions, we focus on this case. When this is not the case, as mentioned in the section about continuation methods (section 4), a homotopy between a trivial system and the target system, combined with continuation methods, might be a promising approach.

Starting from a point in the periodic solution's basin of attraction, one can simply forward integrate such a system. There are several possible problems involved:

- Forward integration can accumulate errors.
- Even if the starting point lies exactly on the periodic trajectory, the sampling interval would probably not be an integer fraction of the period of the periodic trajectory. Periodicity in the sequence of points are not directly related to periodicity in the continuous system.
- Given the nature of the project, it is very likely that period doubling bifurcations are encountered. These provoke situations where two periodic solutions exist which are difficult to distinguish.

Forward integration yields a sequence of points in phase space $(\mathbf{s}_i)_{i\in\mathbb{N}}$. To obtain more manageable data only intersections of the trajectory with a hyperplane $p(\mathbf{x}) = \langle \mathbf{n}, \mathbf{x} - \mathbf{x}_0\rangle = 0$ in a single direction are considered (so called Poincaré sections). To find these, $\mathbf{s}_i, \mathbf{s}_{i+1} = \mathbf{s}_i + h_0 \cdot \mathbf{f}(\mathbf{s}_i)$ with $p(s_i) \leq 0 < p(s_{i+1})$ are searched. Because of continuity, there needs to exists $h \in [0, h_0]$ such that $p(\mathbf{s}_i + h\cdot\mathbf{f}(\mathbf{s}_i)) = 0$, which is found via bisection. Let $(\mathbf{u}_i)_{i\in\mathbb{N}}$ be the sequence of intersections.

To find the number of intersections per period, the intersections are partitioned into $k \in \mathbb{N}$ disjoint clusters $V_{k,i} = \{\mathbf{u}_m \mid m \in k\mathbb{N} + i\}$ for $i \in \mathbb{N}$, $i \leq k$. The relative quality of the i-th cluster can then be assessed using the within-cluster variance $\sum_{v\in V_{k,i}} ||v - \mathrm{E}(V_{k,i})||^2$. The correct number $k_{\min} \in \mathbb{N}$ of intersections in one period is then taken to be the one minimizing the sum of within-cluster variances:

$$k_{\min} \coloneqq \arg\min_{k\in\mathbb{N}} \sum_{i=1}^{k} \sum_{v\in V_{k,i}} ||v - \mathrm{E}(V_{k,i})||^2.$$

For further information about these measures see for example [Halkidi et al., 2001].

In this form the criterion might at best work if the intersection sequence is infinite. When dealing with finite sequences increasing the number of clusters inevitably leads to lower total within-cluster variances. It is thus necessary to constrain the available values for k and discourage the method of overestimating the number of clusters. Trivially an upper limit for k needs to be introduced. Furthermore, the minimality criterion needs to be relaxed: Suppose there are $k_{\min}$ intersections per period, then all multiples of $k_{\min}$ yield equal or lower ratings. One thus wants to choose the minimum k which is in some sense still almost optimal.

4 Tracing Periodic Solutions, Predictor-Corrector Continuation

A central part of this project are numerical continuation methods. Being a large topic, the details are out of the scope of this work, we thus concentrate on conveying a general idea of the concepts. This is especially true for the wealth of methods from other fields, numerical continuation draws upon. Among the things required in the following are basic knowledge of Newton's method in multiple dimensions, forward integrating differential equations (Runge-Kutta methods) and basic vector calculus (Jacobian matrix). For a more thorough introduction to the topic, see [Allgower and Georg, 1990], on which the continuation part of project and this section of this document is based. The following is a reduced introduction of the concepts used for this project.

The base of numerical continuation is formed by the fact that in the vicinity of a solution of an underdetermined continuous system, there are almost always other, similar solutions. Iterative application of this, in the context of systems having one constraint less than unknowns, yields that solutions of the system form a curve (which might be closed). Continuation methods provide means to trace these curves, and thus to derive infinitely many other solutions from a single given solution. A very central use-case of such a method is being able to numerically solve a system without requiring an otherwise needed good starting value. This works by continuously blending two systems with equal numbers of unknowns and equations using a homotopy, which adds one degree of freedom. This way, a known solution from a system can be traced to one of the harder system. However, in the context of this work, where the underdetermination occurs naturally, we are not interested in single solutions, but rather in obtaining the whole continua of solutions.

This project uses a single continuation method: The predictor-corrector method. As the name suggests, it continues the solution curves, by predicting the next point on the curve through linearization at the current point, and afterwards correcting it, to compensate for the non-linear influences.

Davidenko Equation For a matrix $\mathbf{A} \in \mathbb{R}^{N\times(N+1)}$ with full rank, let $\mathbf{t}(\mathbf{A}) \in \mathbb{R}^{N+1}$ denote its *tangent*, that is the unique vector with

$$\begin{aligned} \mathbf{A} \cdot \mathbf{t}(\mathbf{A}) &= 0 \\ ||\,\mathbf{t}(\mathbf{A})|| &= 1 \\ \det \begin{pmatrix} \mathbf{A} \\ \mathbf{t}(\mathbf{A})^T \end{pmatrix} &> 0 \end{aligned}$$

For a given function $\mathbf{h} : \mathbb{R}^{N+1} \to \mathbb{R}^N$ for $N \in \mathbb{N}$, let $\mathbf{J} : \mathbb{R}^{N+1} \to \mathbb{R}^{N\times(N+1)}$ denote its Jacobian. The tangent can be used to define an initial value problem specific to $\mathbf{H}$, the Davidenko equation ([Davidenko, 1953]):

$$\begin{aligned} \frac{d}{dt}\mathbf{c} &= \mathbf{t}(\mathbf{J}(\mathbf{c})) \\ \mathbf{c}(0) &= \mathbf{c}_0, \end{aligned}$$

for a given $\mathbf{c}_0 \in \mathbb{R}^{N+1}$. Solutions $\mathbf{c} : \mathbb{R} \to \mathbb{R}^{N+1}$ to this problem satisfy $\mathbf{h}(\mathbf{c}(t)) = \text{const}$ for all $t \in \mathbb{R}$, as can be seen on

$$\frac{d}{dt}\mathbf{h}(\mathbf{c}) = \mathbf{J}(\mathbf{c}) \cdot \frac{d}{dt}\mathbf{c} = \mathbf{J}(\mathbf{c}) \cdot \mathbf{t}(\mathbf{J}(\mathbf{c})) = 0.$$

From a theoretical perspective, the Davidenko equation is the sole thing needed for continuation. Tracing solutions becomes a matter of finding a solution for the differential equation. In a practical setting, this also includes the same numerical obstacles, and ignores that the underlying functions $\mathbf{h}$ and $\mathbf{J}$ are known. Numerically integrating the

Davidenko equation is the prediction part as indicated earlier. It introduces errors, such that the constancy is no longer guaranteed. However, since $\mathbf{h}$ and $\mathbf{J}$ are known, the point can be *corrected* towards the constant solution curve.

Newton's Method The correction part of the method uses a variant of Newton's method, working in multiple dimensions and with non-square Jacobians. Finding zeros using Newton's method in multiple dimensions works similar to the single dimensional case: A function is linearized, and the zero of the linear problem is considered a good approximation of a zero of the more complex problem. In non-degenerate cases, iterating this procedure quickly converges to a zero, the fixed points of this iteration.

First we consider function $\mathbf{h} : \mathbb{R}^N \to \mathbb{R}^N$ with square Jacobian $\mathbf{J} : \mathbb{R}^N \to \mathbb{R}^{N\times N}$. In point $\mathbf{x}_i \in \mathbb{R}^N$ the next point $\mathbf{x}_{i+1} \in \mathbb{R}^N$ is defined as the root of the linearization in $\mathbf{x}_i$, $\mathbf{x} \mapsto \mathbf{H}(\mathbf{x}_i) + \mathbf{J}(\mathbf{x}_i) \cdot (\mathbf{x} - \mathbf{x}_i)$, which yields

$$\mathbf{x}_{i+1} \coloneqq \mathbf{x}_i - \mathbf{J}^{-1}(\mathbf{x}_i)\mathbf{H}(\mathbf{x}_i).$$

Generalizing the method to $N \times (N + 1)$ Jacobians involves finding a substitute for inverting the Jacobian, which is not defined for non-square matrices. This can be done using *Moore-Penrose* pseudoinverse for a matrix $\mathbf{A} \in \mathbb{R}^{N\times(N+1)}$

$$\mathbf{A}^+ \coloneqq \mathbf{A}^*(\mathbf{A}\mathbf{A}^*)^{-1}.$$

As $\mathbf{A}\mathbf{A}^+ = (\mathbf{A}\mathbf{A}^*)(\mathbf{A}\mathbf{A}^*)^{-1} = \mathbf{I}$ indicates, $\mathbf{A}^+$ is a right-inverse, such that

$$\mathbf{x}_{i+1} \coloneqq \mathbf{x}_i - \mathbf{J}^+(\mathbf{x}_i)\mathbf{H}(\mathbf{x}_i)$$

can be considered Newtons Method for $N \times (N + 1)$ systems. The pseudoinverse allows to solve these underdetermined systems in a least squares sense (for details see [Allgower and Georg, 1990]), that is,

$$\mathbf{x} = \mathbf{A}^+\mathbf{b} \quad \Leftrightarrow \quad ||\mathbf{x}|| = \min\{||\mathbf{x}|| \mid \mathbf{A}\mathbf{x} = \mathbf{b}\}.$$

In a practical setting, one can resort to simply solving

$$\mathbf{J}(\mathbf{x}_i)\mathbf{x} = \mathbf{H}(\mathbf{x}_i)$$

in this manner, instead of costly inverting the Jacobian.

Broyden Updates The formulation in terms of the Moore-Penrose pseudoinverse allows to employ the *Broyden* update methods. These make it possible to avoid having to calculate the entire Jacobian and solving the possibly huge systems, at the cost of using approximations the Jacobian. These updates have not been employed in the implementation of the methods for this project and are out of the scope of this document. However, it should be noted, that the vast majority of processing time in the implementation is used for calculating the tangent of the Jacobian, which is required in every predictor and corrector step. Broyden updates allow to update the tangent as well, making them a priotity goal for a possible extension of the implementation.

5 Case Study: Lorenz System

6 Case Study: Rössler System

The well known Rössler system is given by the equations

$$\begin{aligned} \dot{x} &= -y - z \\ \dot{y} &= x + a \cdot y \\ \dot{z} &= b + z \cdot (x - c) \end{aligned}$$

with parameters a, b, and c. For this example, a and b are fixed to 0.1, while we are varying c.

The system has several periodic solutions for each value of c with different periodicities, though only one is stable at a time. We are interested in the origin and branching of those solutions and, thus, drawing a bifurcation diagram using the map

$$f \mapsto \{\|f(t)\|_2 \mid t \in [0, 2\pi),\ f(t)_1 = 0\} \ .$$

Here, f denotes a single periodic solution. Note, that we use only the approximation given by Galerkin's method.

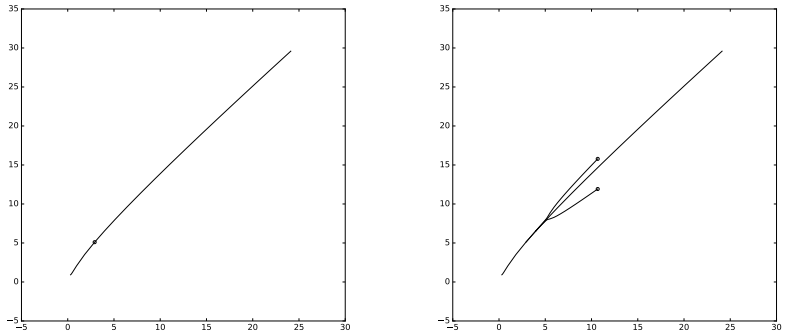
Now, the exploration of the bifurcation with the given tools follows rougly this steps:

1. As a starting point, search for a value of c such that the system has a stable, periodic solution and use the method described in section 3 to find it. 4000 iterations in the transient part with step size of 0.01, 120 generated intersections with the $(x = 0)$-plane and tests for at most 30 periods were sufficient for all initial solutions of the Rössler system. We started at $c = 4$ with 64 samples.

2. Trace out the branch just by following the newly found solution in both directions using the predictor-corrector continuation method (section 4). The parameters $\kappa = 0.4$, $\delta = 3.0$, and $\alpha = 10.0$ are a decent choice for the whole diagram as a good tradeoff between performance and accuracy.

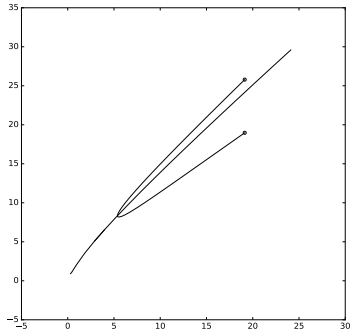
3. It is possible to detect a pitchfork bifurcation by doubling up the periodicity artifically. Using the described predictor-corrector method, the value of c will eventually converge to the bifurcation point, i.e., a point with a singular tangent. Since the method is now stuck, reaching the doubly-periodic solution can be done by solving a slightly perturbed problem for a short time. A short perturbation of strength 0.2 was enough to reach the doubly-periodic solution. Also, it might be necessary to increase the sample size due to increased complexity in the descending branch.

4. Remove the perturbation and continue tracing out the bifurcation, i.e. start over in 1 or 2.

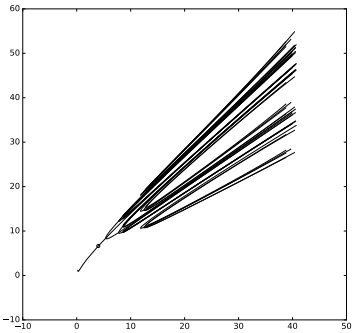


(a)

(b)



(c)



(d)

Figure 1: Tracing out the branch of solutions beginning with the solution for $c = 4$ (a). The bifurcation point in $c \approx 5.376$ is avoided by using a permutation (b), and the two-periodic solutions can be traced out after removing the inaccurate values from before (c). Eventually, we reach (d), a widely traced out bifurcation diagram. Note the solutions with periodicity three and descendants, that do not originate in the peridocity-one branch.

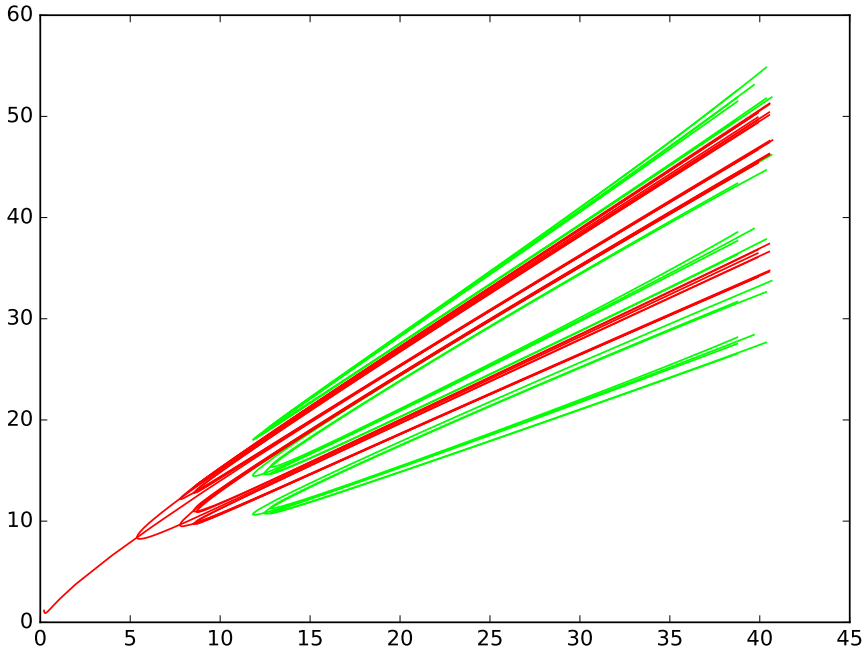


Figure 2: The bifurcation diagram of the Rössler system for varying c. The even-periodic solutions are colored red, the odd-periodic ones green.

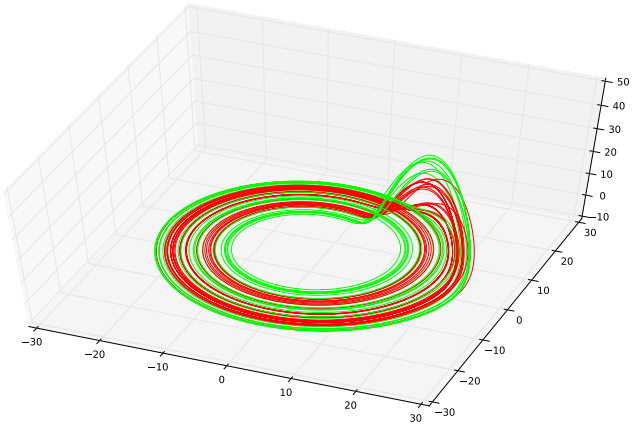


Figure 3: All found solutions for $c = \ldots$ plotted into one diagram. Observe how the odd-periodic and even-periodic solutions dodge since they do not originate from each other.

---

7 Results & Outlook

The project's goals of interactively creating bifurcation diagrams using Galerkin's method and continuation methods could be met. There are some further points of interest. On a practical level, one of the things an implementation of the sort of this project might benefit from are Broyden updates. In its basic form, the method allows to update the Jacobian of the system of equations obtained from the Galerkin operator. Though calculating the Jacobian is a time-consuming task, this is of minor interest, as the effort is negligible compared to the effort for inverting the Jacobian, which is needed for each computation of the tangent.

One of the main sources of complications are orbits which change their direction too fast. As is the case in the Rössler system for high values of γ and in the Lorenz System for low values of ρ. In these regions the optimization procedure converges slowly and the predictor-corrector method's step-size adaption suggests very small steps (as seen from the bifurcation view). There is a variety of possible reasons

Numerical Certain values, for example in the Jacobian, are not representable sufficiently precise.

Modeling A trigonometric polynomial of finite degree can not be used to model infinitely sharp turns. The numerical cause is likely to be a consequence of this choice for a model.

No attempts were made to investigate the definite cause of this behavior, however, choosing a model which is more adapted to these challenges might allow for more accurate results in these areas. Furthermore, solution branches could be traced further, as these effects are a cause of stopping the tracing of solutions in the Rössler, as well as in the Lorenz system. While in the case of the Rössler system this is less of a problem, as the solutions for high γ did not appear to reveal anything of immediate interest, in the Lorenz system the behavior prohibited tracing the solutions in the $\rho \in (0, 2)$ area. It is this area, where there is a huge amount of interaction between the branches of solutions.

Modeling the trajectories as linear combinations of non-equidistant B-splines instead of complex oscillations might yield interesting results.

- The periodicity requirement can be enforced by choosing the basis vectors appropriately.
- Through adaption of the knots, the degree of smoothness can be varied to the point being C^0 only. This can be done by changing *multiplicities* of knots, that is, by having several knots be equal. An interesting point is that this change in continuity can occur dynamically through optimizing the knot vector.

References

[Allgower and Georg, 1990] Allgower, E. L. and Georg, K. (1990). *Numerical Continuation Methods: An Introduction*. Springer Verlag Berlin Heidelberg.

[Davidenko, 1953] Davidenko, D. (1953). On a new method of numerical solution of systems of nonlinear equations. In *Dokl. Akad. Nauk SSSR*, volume 88, pages 601–602.

[Halkidi et al., 2001] Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of intelligent information systems*, 17(2-3):107–145.