# 3D Printing of Invariant Manifolds in Dynamical Systems

*Patrick R. Bishop, Summer Chenoweth, Emmanuel Fleurantin, Alonso Ogueda-Oliva, Evelyn Sander, and Julia Seay*

## 1. Introduction

Invariant manifolds play a crucial role in understanding the behavior of dynamical systems and ordinary differential equations, providing geometric structures that organize the state space and reveal fundamental aspects of a system's long-term dynamics. These invariant sets, particularly stable and unstable manifolds, organize the dynamical behavior, such as serving as the separation between different basins of attraction. Through their intersections and foliations, the manifolds determine the qualitative structure of the flow. While computational methods for studying these structures have advanced significantly, visualizing and intuitively grasping their complex geometries remains challenging.

This article introduces a novel approach to address this challenge: 3D printing tangible, physical representations of invariant manifolds, as shown in Figures 1–4. In this article, we describe the details of designing printable manifolds with enough detail so that—along with the provided code—a reader will be able to apply these techniques to design and print their own examples. This approach not only enhances our ability to visualize and study these intricate structures but also provides new pedagogical tools for teaching concepts in dynamical systems theory. Related work includes design of 3D printed chaotic attractors [LST20, BPB23, Gag18] and creation of crocheted invariant manifolds [OK04]. These previous efforts address a fundamentally different computational problem: existing 3D printing studies focus on chaotic attractors, created from one-dimensional trajectory curves, which are part of an attracting set and are therefore easily found by starting from any nearby initial condition and removing transient behavior. Computing invariant manifolds is a significantly more challenging problem; invariant manifolds are not attractors, and thus finding them is much more delicate than starting at any nearby point. In addition, printing two-dimensional invariant manifolds is more difficult than printing single trajectories, as surfaces consist of more than one trajectory and thus require geometric reconstruction.

The process of transforming invariant manifolds for flows into physical 3D printed objects involves three distinct phases, each with its own computational and technical challenges. The first phase focuses on generating an accurate numerical representation of the manifold. It consists of two parts: computing the local manifold, and computing the global manifold. For local manifold generation, we employ the Parameterization Method, a technique which provides high-order approximations of the invariant manifold on a fundamental domain near the equilibrium solution. Computing the global manifold is done using a uniform arclength integration scheme, allowing us to extend the manifold to a desired size.

*Patrick R. Bishop is a recent PhD graduate from George Mason University. His email address is* pbishop3@gmu.edu.
*Summer Chenoweth is a PhD student in the Department of Mathematical Studies at George Mason University. Her email address is* schenowe@gmu.edu.
*Emmanuel Fleurantin is a postdoctoral research fellow in the Department of Mathematical Studies at George Mason University. His email address is* efleuran@gmu.edu.
*Alonso Ogueda-Oliva is a PhD student in the Department of Mathematical Studies at George Mason University. His email address is* aogueda@gmu.edu.
*Evelyn Sander is a professor in the Department of Mathematical Studies at George Mason University. Her email address is* esander@gmu.edu.
*Julia Seay is a PhD student in the Department of Mathematical Studies at George Mason University. Her email address is* jseay2@gmu.edu.

*Communicated by* Notices *Associate Editor Vidit Nanda.*



DOI: https://doi.org/10.1090/noti3288

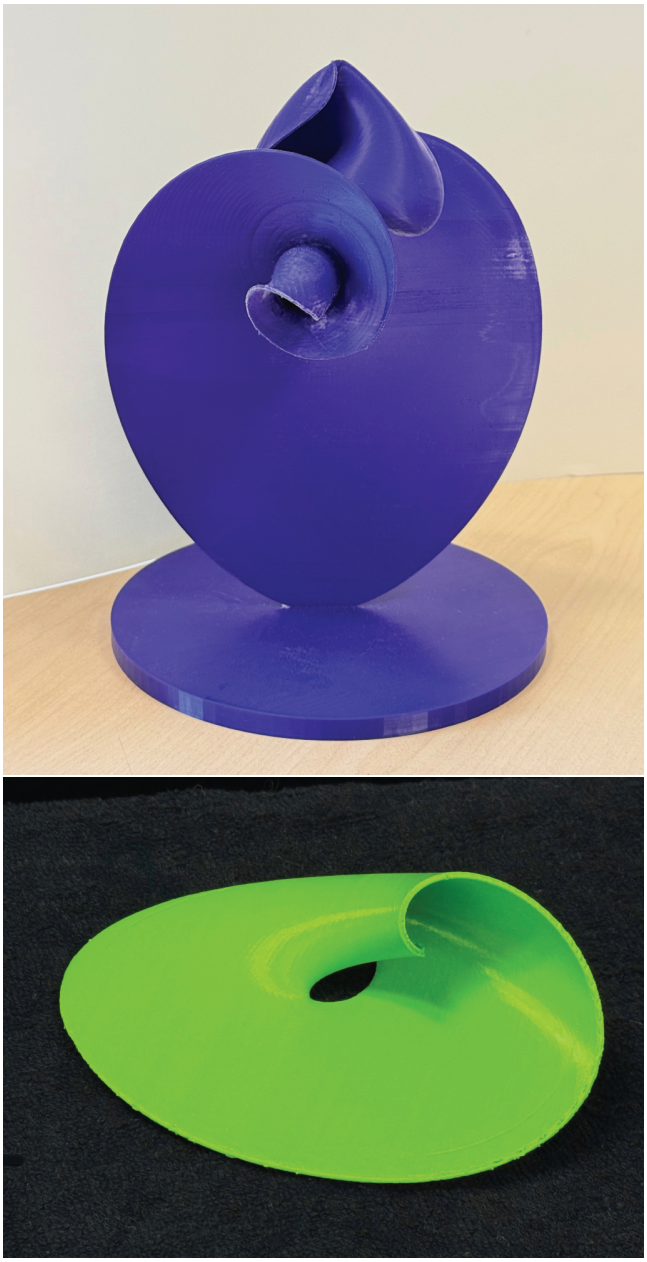

**Figure 1.** Top: A 3D printed stable manifold of the origin for the Lorenz system (10). Bottom: A 3D printed unstable manifold in the Arneodo–Coullet–Tresser system (11) with $\beta = 0.4$ and $\mu = 0.863$.

The second phase addresses the crucial transition from a mathematical object to a printable 3D model. This requires implementing appropriate meshing schemes to create a triangulated surface representation. A key challenge here is the generation of a "thickened" version of the manifold—converting the idealized surface of zero thickness into a physically printable object with appropriate thickness and structural integrity. The meshing algorithm must also account for potential self-intersections and ensure proper orientation of surface normals.

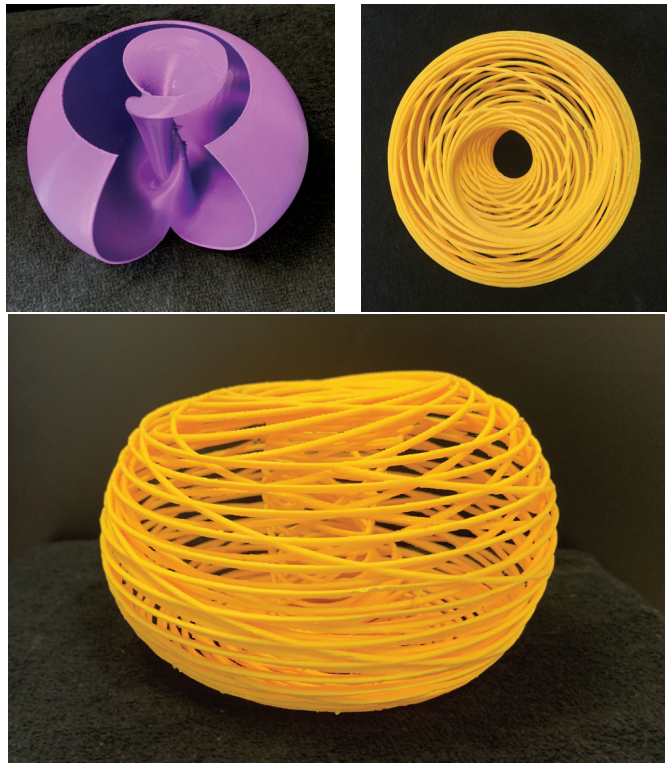

**Figure 2.** 3D printed unstable manifolds of equilibria in the Langford system (12). Top left: Unstable manifold of $p_1 \approx 1.94$ ($\alpha = 0.95$) with complex multilobe structure and cutout revealing internal geometry. Top right/bottom: Two views of unstable manifold of $p_1 \approx 1.84$ ($\alpha = 0.806$) showing spiral structure from above and 3D profile. Grid structure printing provides an alternative to cutouts for displaying intricate details.

The final phase involves preparing the model for actual physical printing using slicing software. This stage requires careful consideration of multiple printing parameters: material selection, support structure generation for overhanging features, optimal build plate orientation to minimize supports while maintaining structural integrity, and layer height settings to balance print quality with production time. The choice of these parameters significantly impacts both the visual quality and the structural stability of the final printed manifold. Additionally, considerations must be made for the scale of the printed object, ensuring that fine geometric details are preserved while maintaining printability and handling requirements.

We end this introduction with a comment on the validity of our prints as reflecting true mathematical objects. There is great current interest in assuring rigor in mathematical visualization, such as in the upcoming conference "Rigorous Illustrations—Their creation and evaluation for mathematical research" [BDHHS26]. The visualization of manifolds used here fits squarely into the intersection of visualization and rigor; several of these manifolds have been rigorously validated using

computer-assisted proofs. Although the software used for rigorous validation differs from what is suited for visualization, the mathematical theory and methods are the same. The field is being disseminated through a number of workshops and summer schools, such as the recent SLMath graduate summer school "Computer-Assisted Proofs in Applied Mathematics" in July 2025 [JS25]. For general results on the validation of invariant manifolds see for example [MJ18, CLMJ18]. The particulars are given for the Lorenz system (top of Figure 1) in [KKMJ18] and for the Langford system (Figure 2) in [CFMJ20]. Similar methods could also be used for the Arneodo–Coullet–Tresser model (bottom of Figure 1).

Our paper proceeds as follows. We do not assume that the reader has a background in either invariant manifolds or 3D printing. Therefore we start by reviewing key concepts in the study of invariant manifolds in Section 2. In Section 3, we present the Parameterization Method for local manifold calculation. Section 4 discusses global manifold computation. Section 5 details the process of converting mathematical structures into printable models. Section 6 demonstrates our approach through three example systems. We conclude, in Section 7, with practical guidelines for reproducing and generalizing our results, providing references to our code repository in Section 8.

## 2. Stable Manifolds

A dynamical system describes evolution under a consistent set of rules. This very general class includes systems governed by principles in many different fields, including applications in physical, biological, chemical, and social sciences. In this paper, we restrict to the case in which the state variable $x \in \mathbb{R}^n$, and our dynamical system can be written as a nonlinear ordinary differential equation of the form

$$\dot{x} = f(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a smooth function. Let $\phi_t(x)$ represent the solution of the differential equation at time $t$ with initial condition $x$, and assume that $\phi_t(x)$ is well defined for all $t \in \mathbb{R}$.[1]

A set $W$ is an *invariant set* if $x \in W$, then $\phi_t(x) \in W$ for all $t$. As a first example, consider a homogeneous linear equation $\dot{x} = Ax$, where $x \in \mathbb{R}^n$, and $A$ is an $n$ by $n$ matrix. If $v$ is an eigenvalue of $A$ with associated eigenvalue $\lambda \in \mathbb{R}$, then the line through the origin parallel to $v$ is an invariant set containing the solution $e^{\lambda t}v$. Another invariant set is the stable subspace $E^s$, defined as the subspace spanned by the $n_s$ linearly independent generalized eigenvectors associated with eigenvalues with negative real part. This space is characterized as the unique set of initial conditions such that $\lim_{t\to\infty} \phi_t(x) \to 0$, where convergence is exponentially fast. Likewise, the unstable subspace $E^u$ of generalized eigenvectors associated with eigenvalues with positive real part, is invariant. It is characterized as the unique set of initial conditions such that $\lim_{t\to-\infty} \phi_t(x) \to 0$, where convergence is exponentially fast. If $A$ is *hyperbolic*, i.e., all eigenvalues have nonzero real part, then $\dim(E^s)+\dim(E^u) = n$. Figure 5a shows the case of

$$A = \begin{pmatrix} -1 & 0 \\ 0 & 2 \end{pmatrix}. \tag{2}$$

In this case,

$$\lambda_1 = -1, \qquad v_1 = (1,0),\ E^s = \{(x,y) : y = 0\}$$
$$\lambda_2 = 2, \qquad v_2 = (0,1),\ E^u = \{(x,y) : x = 0\}.$$

For nonlinear systems (1) (i.e., where $f$ is nonlinear) the *stable manifold theorem* guarantees that near an equilibrium $p$, there are invariant sets which are analogs of the stable and unstable subspaces, called local stable and unstable manifolds. The *linearization* of the differential equation at equilibrium $p$ is given by

$$\dot{y} = Df(p)y. \tag{3}$$

**Figure 3.** Different perspectives of the 3D printed stable manifold of the Lorenz system (10) at the origin. Top: Two views of the larger model showing the characteristic spiral structure and complex geometry. Bottom: Comparison with a smaller version (held in hand for size reference), demonstrating that the same data can be used for various print sizes.

[1] *It is always possible to reparameterize time so that this is the case.*

This linear equation has the closed form solution $e^{tA}y_0$, where $y(0) = y_0$ is any initial condition. We refer to $p$ as a *hyperbolic* whenever $Df(p)$ is hyperbolic.

**Definition 2.1** (Stable and unstable manifolds). Assume that $p$ is a hyperbolic equilibrium for (1). Define the following set of points in $\mathbb{R}^n$.

1. Global stable manifold ($W^s(p)$): The set of all initial conditions $x$ such that $\phi_t(x)$ converges to $p$ for $t \to \infty$.
2. Global unstable manifold ($W^u(p)$): The set of all initial conditions $x$ such that $\phi_t(x)$ converges to $p$ for $t \to -\infty$.

The following theorem locally characterizes the stable and unstable manifolds.[2]

**Theorem 2.2** (Stable manifold theorem [GH83]). *Assume that $p$ is a hyperbolic equilibrium for (1) with corresponding linearization (3). For matrix $A = Df(p)$, let $n_s$ and $n_u$ denote the number of eigenvalues (with multiplicity) with negative and positive real parts respectively. Since $p$ is hyperbolic, $n_s+n_u = n$. Then for sufficiently small $\epsilon > 0$, there exist unique invariant dynamical sets, known as* local *stable and unstable manifolds, defined as follows:*

$$\begin{aligned} W^s_{loc}(p) &= \{x \in W^s(p) : \phi_t(x) \in B_\epsilon(p), t \geq 0\}, \\ W^u_{loc}(p) &= \{x \in W^u(p) : \phi_t(x) \in B_\epsilon(p), t \leq 0\}. \end{aligned} \tag{4}$$

*These local manifolds are surfaces through $p$ that are as smooth as $f$, with dimensions $n_s$, $n_u$ respectively. Let $E^s$ and $E^u$ be the stable and unstable subspaces corresponding to $\dot{y} = Ay$. At $p$, the local manifolds are tangent to $E^s$ and $E^u$ respectively.*

As an example, consider the equation

$$\dot{x}_1 = -x_1, \quad \dot{x}_2 = 2x_2 - x_1^2. \tag{5}$$

The linearization of this equation is $\dot{x} = Ax$ for $A$ in (2), as in Figure 5a. By direct calculation, the general solution is given by $x_1(t) = c_1e^{-t}$ and $x_2(t) = c_1^2/4e^{-2t} + c_2e^{2t}$, where $c_1, c_2 \in \mathbb{R}$ are arbitrary. A direct verification shows that

$$\begin{aligned} W^s(p) &= \left\{(x_1, x_2) : x_2 = x_1^2/4\right\} \text{ and} \\ W^u(p) &= \{(x_1, x_2) : x_1 = 0\}. \end{aligned} \tag{6}$$

See Figure 5b. We have not explained how we came up with these equations; in the next section, we will give a constructive method for calculating invariant manifolds.

## 3. Local Structure

A powerful technique developed by Cabré, Fontich, and de la Llave [CFdlL03] enables the computation of local stable and unstable manifolds. In particular, the method involves finding a parameterization of the local manifold by using its invariance, providing a characterization of the local behavior of all solutions starting near an equilibrium.

[2] *We assume a hyperbolic equilibrium, even though the theorem holds in a more general setting.*

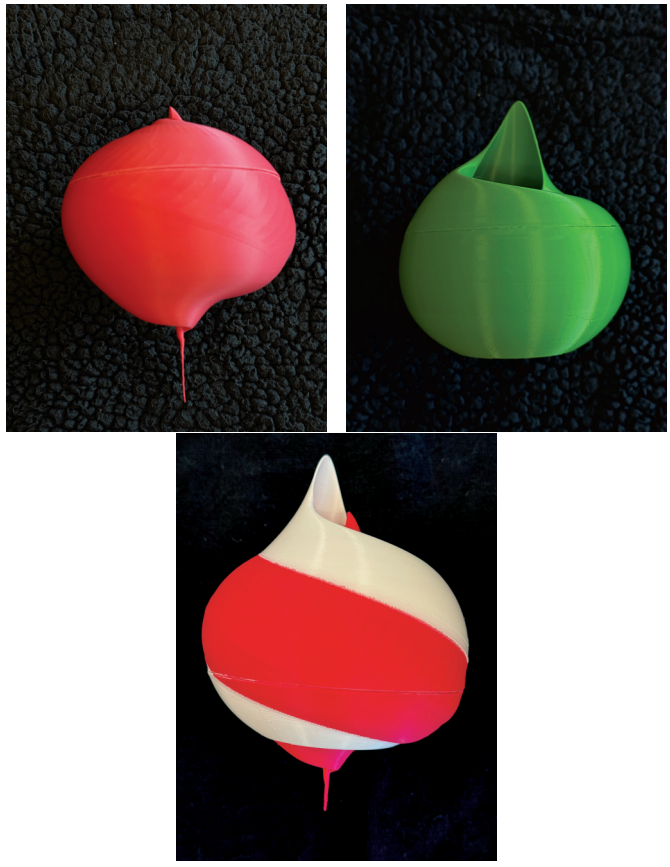

**Figure 4.** 3D prints of intersecting global manifolds in the Langford system. Top: 2D unstable manifold of equilibrium point $p_1$ (left); 2D stable manifold of equilibrium point $p_2$ (right). Bottom: View showing intersection between manifolds with one protruding through the other. Langford system parameter values are given in Section 6.

We first recall the Hartman–Grobman theorem, which locally linearizes dynamics near hyperbolic equilibria.

**Theorem 3.1** (Hartman–Grobman). *Assume that $p$ is a hyperbolic equilibrium for (1), with corresponding linearized equation (3). Then there exists $\delta, s > 0$ and a homeomorphism $h : B_s(0) \to B_\delta(p)$ such that $h$ is a conjugacy between solutions of the linear equation and solutions of the nonlinear equation. That is, for any $y \in B_s(0)$,*

$$\phi_t(h(y)) = h(e^{tA}y).$$

*This implies that $h$ maps trajectories of the linear equation to trajectories of the nonlinear equation, while preserving their topology and orientation.*

The Parameterization Method proceeds as follows. Consider (1) with a hyperbolic equilibrium point $p$, where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a real analytic vector field. Assume $Df(p)$ is diagonalizable[3] with distinct eigenvalues satisfying certain nonresonance conditions. Let the eigenvalues

[3] *The diagonalizability assumption simplifies the recursive computation of coefficients but is not essential as Newton's method may be used.*

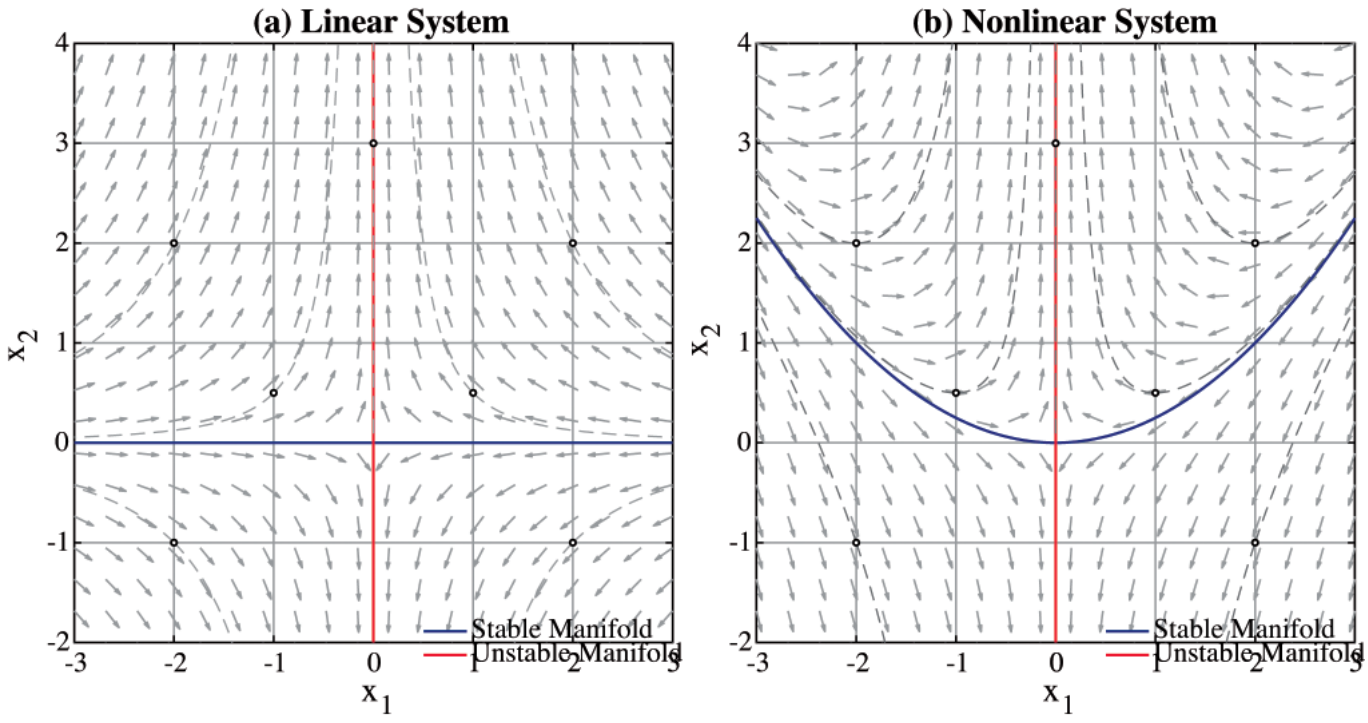


**Figure 5.** Phase portraits for **(a)** linear system $\dot{x}_1 = -x_1$, $\dot{x}_2 = 2x_2$ showing $E^s$ and $E^u$; **(b)** nonlinear system (5) showing $W^s$ and $W^u$ from (6). Gray arrows indicate vector field directions.

be partitioned as $\{\lambda_i^s\}_{i=1}^{n_s}$ with $\mathrm{Re}(\lambda_i^s) < 0$ and $\{\lambda_i^u\}_{i=1}^{n_u}$ with $\mathrm{Re}(\lambda_i^u) > 0$, with corresponding eigenvectors $\{\xi_i^s\}_{i=1}^{n_s}$ and $\{\xi_i^u\}_{i=1}^{n_u}$. As before, let $h$ be the conjugacy function given by the Hartman–Grobman theorem, and let $\theta \in \mathbb{R}^{n_s}$ with $|\theta| \leq 1$. Consider the vector $v_s$ in the stable linear subspace $E^s$ given by $v_s = \theta_1 \xi_s^1 + \cdots + \theta_{n_s} \xi_s^{n_s}$. Then using the scale parameter $s$, define the parameterization $P$ of the stable manifold by

$$P(\theta) = h(sv_s). \tag{7}$$

This gives a parameterization $P : B_1(0) \subset \mathbb{R}^{n_s} \to \mathbb{R}^n$ for $W^s_{\mathrm{loc}}(p)$. Using the same argument as before, differentiating this equation with respect to $t$ and setting $t = 0$, we see that

$$f(P(\theta)) = DP(\theta)\Lambda\theta, \tag{8}$$

where $\Lambda$ is the diagonal matrix with the eigenvalues $\lambda_s^i$, $i = 1, \ldots, n_s$ on the diagonal. In order to approximate $P$, we write $P(\theta)$ as a series solution, and use equation (8) to solve for the coefficients in the series. In the previous example, we were able to compute an exact formula for the stable manifold, but in general we can only approximate the series for $P$ with a finite number of terms. In particular we refer to the truncated series $P(\theta) = \sum_{j,k=1}^{N} P_{jk}\theta_1^j\theta_2^k$ as an *order N approximation*. For a more detailed explanation and implementation of the Parameterization Method, one can refer to [FJ20]. Note that this numerical method assumes sufficient smoothness of the vector field. The accuracy of the approximation improves as the order $N$ increases, subject to the radius of convergence of the series. Additionally, the choice of scale parameter $s$ in (7) affects the size of the computed local manifold patch. Optimal selection of $s$ depends on the specific system properties and desired visualization region.

Now in order to be able to 3D print our results, we are focusing on the case of a three-dimensional phase space. In this case, the only possibilities are that the equilibrium has stable (resp. unstable) manifolds of dimension 0(3), 1(2), 2(1), or 3(0). The first and last case are uninteresting, as one of the two manifolds is locally the whole space. Therefore we only consider the cases where the stable and unstable manifolds are dimensions 1 and 2 or 2 and 1. Computing the one-dimensional manifold is completely analogous but more straightforward to calculate than the two-dimensional one, and therefore we give a detailed description on computing the two-dimensional manifold.[4] A general algorithm to compute the local stable manifold proceeds as follows. We determine our desired order $N$. We compute the stable eigenvalues and use them to initialize the coefficients in the series approximation for $P$. We iteratively compute the values of these coefficients up to order $N$. We then evaluate $P(\theta)$ at the points furthest from the equilibrium $p$, i.e., on the circle $C_0 = \{\theta : |\theta - p| = 1\}$. This topological circle is the set of points that we use in order to compute the global stable manifold (described in the next section). The unstable manifold computation is

[4] *Code is provided for both the one- and two-dimensional manifolds.*

completely analogous; note that the two manifolds must be computed separately.

Before stating the general case, we give a basic understanding of the method by finding a parameterization of the local stable manifold by returning to the nonlinear example from Equation (5) in Section 2. Recall that (5) has a hyperbolic equilibrium $p = 0$ and the linearized equation has eigenvalues $\lambda_1 = -1 < 0$ and $\lambda_2 = 2 > 0$, with corresponding eigenvectors $v_1 = (1,0)$ and $v_2 = (0,1)$. Thus $E^s = \text{span}(v_1)$, the $x$-axis, and $E^u$ is the $y$-axis. By the stable manifold theorem, we know that $W^s_{loc}(p)$ is a one-dimensional curve that intersects $p = 0$, and is tangent to the $x$-axis at 0. By the Hartman–Grobman theorem, we know that there exists $h : B_s(0) \to B_\delta(p)$ such that $h(x\text{-axis}) = W^s_{\text{loc}}(p)$. We use this to define a parameterization $P(\theta)$ for the stable manifold as follows. For every $|\theta| < 1$, define $P(\theta) = h((\theta, 0))$.

We now use the Hartman–Grobman theorem to find an equation for $P$. Since $\lambda_1$ is $-1$ we know that the linear solution starting at $(\theta, 0)$ has value $e^{tA} = (e^{-t}\theta, 0)$ at time $t$. Therefore the conjugacy equation for $h$ implies $\phi_t(P(\theta)) = P(e^{-t}\theta)$. Differentiating the equation with respect to $t$, setting $t = 0$, and rearranging the terms, the conjugacy equation (8) gives $f(P(\theta)) = -\theta DP(\theta)$. We now write $P(\theta)$ in the form of a Taylor series

$$P(\theta) = \begin{pmatrix} a_0 + a_1\theta + a_2\theta^2 + O(\theta^3) \\ b_0 + b_1\theta + b_2\theta^2 + O(\theta^3) \end{pmatrix}$$

where $a_0 = b_0 = 0$ since $p = 0$ is contained in the stable and unstable manifolds. We find the rest of the coefficients by substituting $P$ into the conjugacy equation, giving that $-\theta DP(\theta) = f(P(\theta))$, which implies

$$\begin{pmatrix} -\theta(a_1 + 2a_2\theta + 3a_3\theta^2) \\ -\theta(b_1 + 2b_2\theta) \end{pmatrix} = \begin{pmatrix} -(a_1\theta + a_2\theta^2) \\ 2(b_1\theta + b_2\theta^2) - (a_1\theta + a_2\theta^2)^2 \end{pmatrix} + O(\theta^3).$$

Solving term-by-term, we find that for the first component: $a_1$ is arbitrary (we choose $a_1 = 1$), and $a_k = 0$ for $k > 1$; and for the second component: $b_1 = 0$, $b_2 = 1/4$, and $b_k = 0$ for $k > 2$. Therefore, $P_1(\theta) = \theta$ and $P_2(\theta) = \frac{1}{4}\theta^2$, revealing that the stable manifold is given by: $x_2 = x_1^2/4$, as stated when we first introduced the equation.

In designing the 3D prints, the Parameterization Method is performed in Matlab, yielding a data file with a set of points in $\mathbb{R}^3$ lying on a circle that surrounds the equilibrium within the invariant manifold. For example, in Figure 6, Matlab produces the red disk, and the data file consists of points on the boundary of this disk. The red disk is the hole in the middle of the manifold in Figure 3 (bottom).

## 4. Global Structure

Now that we have a characterization of the local behavior, we describe the global stable and unstable manifolds: They are obtained from the respective local manifolds by letting points in the local manifolds evolve by direct integration. That is,

$$W^s(p) = \bigcup_{t \le 0} \phi_t(W^s_{loc}(p)), W^u(p) = \bigcup_{t \ge 0} \phi_t(W^u_{loc}(p)).$$

Thus once the local manifold has been determined, we need to integrate to obtain the global manifold. Many different numerical methods for computing global manifolds are described in the comprehensive book of Krauskopf, Osinga, and Galán-Vioque [KOGV07], papers of Dellnitz and Junge [DHVMZ16], Krauskopf et al. [KOD+05], Henderson [Hen05], and Haro [HCF+16]. Note that the global manifolds are in general unbounded, so the best we can do is to integrate these equations up to some distance from the equilibrium. In some cases, this simply involves starting on the boundary of the local stable (resp. unstable) manifold, and integrating backward in time for $t$ from 0 to $-T$ (resp. $T$) to get an additional portion of the global stable (unstable) manifold. However, in general this is not possible to integrate simply.

Specifically, let $C_0$ be the circle of points furthest from $p$ on the local manifold. In order to ensure that the points are well spaced, we reparameterize $C_0$ with respect to the arclength parameterization given by the curve $r_0(\eta)$ where $\eta$ varies from 0 to 1 and $r_0(0) = r_0(1)$. We wish to integrate this curve backward (forward) in time. That is, we wish to solve the following ordinary differential equation, where we allow the solution $y$ to depend on the independent variable $t$ but also on the variable $\eta$, as this allows us to parameterize the set of initial conditions:

$$\frac{d}{dt}(y(t,\eta)) = f(y(t,\eta)) \text{ , where } y(0,\eta) = r_0(\eta).$$

While it is possible to write a numerical solver to compute this, since our goal is to create a printable mesh, we can also take advantage of built-in solvers that represent solutions as parameterized surfaces, meaning that we can take advantage of built-in meshing algorithms.

If the eigenvalues $\lambda_1$ and $\lambda_2$ are complex conjugates, then they have the same stretching factor, and the method of direct integration works to give a nice depiction of the global manifold without further modification. However, if $\lambda_1 < \lambda_2$, this method lead to unequal stretching during the integration process, and as $T$ grows the stretching becomes so extreme that our integration essentially only captures a single dimension of the manifold, see Figure 6a.

A solution to this stretching problem [KOD+05] is to integrate in such a way that the time is parameterized by arclength, see Figure 6b. Assume the original differential equation is given by $\dot{x} = f(x)$, where

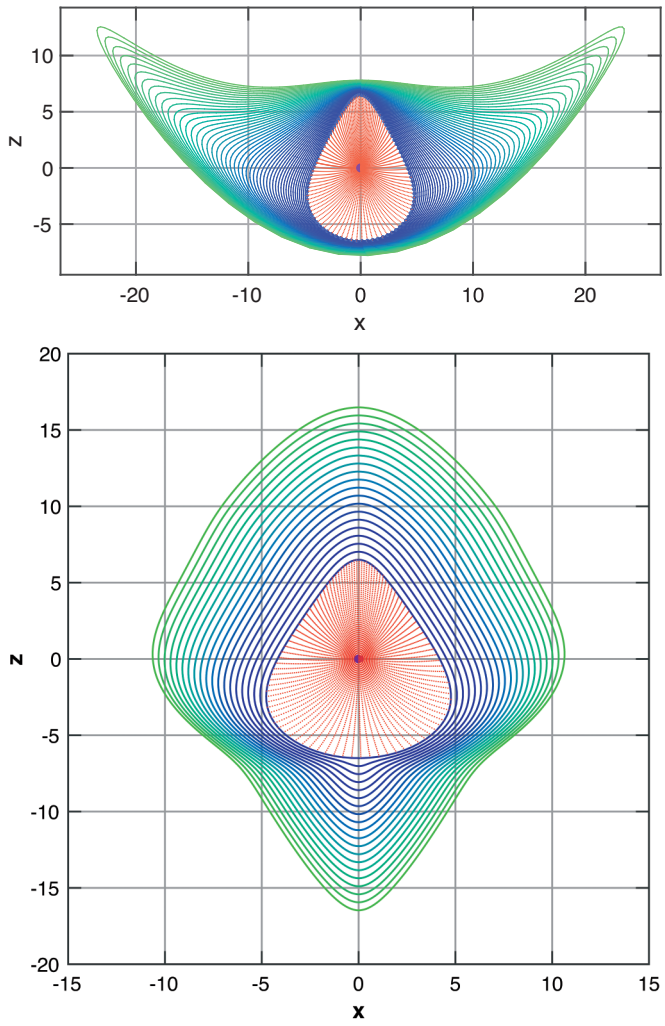


**Figure 6.** Comparison of integration methods for computing the global stable manifold of the Lorenz system (10). Red inner regions represent the local stable manifold $W^s_{loc}(p)$ at the origin. (a) With real, distinct eigenvalues $\lambda_1 < \lambda_2$, direct integration causes unequal stretching. (b) Arclength parameterization resolves this issue.

$x = (x_1, x_2, x_3)$, and $f(x) = (f_1(x), f_2(x), f_3(x))$. Denote $\|f\|_2 = \sqrt{(f_1(x))^2 + (f_2(x))^2 + (f_3(x))^2}$. Our reparameterized differential equation is given by

$$\frac{dx}{d\tau} = \frac{f(x)}{\|f\|_2}, \quad \frac{dt}{d\tau} = \frac{1}{\|f\|_2}. \tag{9}$$

This yields the same trajectories as the original equation, but the time rescaling implies that along the curve in the global manifold with a fixed value of $\tau$, all the points are the same geodesic distance from the starting curve $r_0$. In addition to this modification of our original vector field, if we calculate with the initial conditions for too long, the resulting points will start to bunch up in small regions, thereby not representing the full behavior along a topological circle. Therefore, we break our full length $T$ into $K$ time steps of length $T/K$. Before any time step from $T_j$ to $T_{j+1}$, we find a reparametrization of the outermost points with even spacing via arclength, i.e., of the computed circle of points at time $T_j$.

In the design of the 3D prints, the arclength parameterization of the global manifold is created in Mathematica. Because of the way that Mathematica stores mesh structures for surfaces, this allows us to turn the surface into a printable object using built-in commands, as described in the next section.

## 5. Printing Details

Having established the basic principles and general computational methods for generating invariant manifolds, we now turn to the practical challenges of transforming these mathematical structures into physical objects.

Advances in additive manufacturing technology have revolutionized the creation of mathematical objects [SLR19, LAC17]. From industrial prototyping to educational visualization [Seg12], new techniques and materials continue to enable increasingly sophisticated prints [Zas20]. These developments, coupled with the standardization of 3D printing workflows [JCW+22], provide the foundation for tackling the unique challenges of printing invariant manifolds.

However, despite these technological advances, the specific task of converting computed manifold data into 3D-printable models requires careful consideration of both numerical accuracy and the physical constraints of additive manufacturing. Our methodology addresses several key challenges, including mesh/surface generation, structural stability, and the preservation of mathematical features through the printing process. In what follows, we give an overview of our pipeline from numerical data to finished print.

We start by describing the design, done in our case in the software packages Matlab, Mathematica, and in one case Blender. We follow this by a discussion of setting up for printing, which is done using a *slicer* which gives the precise instructions to the printer as to where to place filament and in which order.

Focusing first on the method of design of a printable mesh (usually stored in an stl file), the main problem that we need to overcome is that a parameterized surface is not actually a solid, in that it contains no volume. In order to be able to print a surface, we need some way to thicken it so that it can print. There are various ways to do this, but we have found that the most reliable way to do it is by using a Matlab program which performs the Parameterization Method for computing local invariant manifolds; followed by bringing the data into Mathematica, see for example `makelorenzmanifold.nb` in the Github repository (Section 8), in order to generate the object using the built-in integration methods for

creating and printing parameterized surfaces. Starting from the points on the circle in the invariant manifold (provided by Matlab) we use a Mathematica `Module` entitled `redistributePoints` to create even spacing on the circle, and use `Interpolation` to parameterize the circle with components `f0,g0`, and `h0`. We create the global invariant manifold using the differential equation solver `NDSolve`, which allows us to specify an initial value to be the curve given by `{f0[x],g0[x],h0[x]}`. We solve the differential equation for time $T$,[5] resulting in a parameterized surface of points. We convert surfaces to three-dimensional printable shells using the `ParametricPlot3D` command with the `Thickness` modifier. Here we scale the function by a uniform factor in order to control the exact thickness. A good rule of thumb is that the object should be at least 1.5-2 mm[6] thick to avoid a lack of structural stability.

While in most of the figures, we have created the full manifold surfaces, we have also developed two other methods of design in order to be able to see the internal structures that would otherwise be hidden by the surface. Figure 2 shows the first of these methods: we have created a grid instead of a full surface. In order to do this, instead of integrating the entire local manifold circle of points, we applied the `NDSolve` command up to time $T$ for a finite number of evenly spaced points along the curve. These points alone would not be connected, and the resulting print would be a bit like a bowl of spaghetti. In order to give rigidity to the grid, we add circles of points at fixed (arclength parameterized) time steps, appearing like a sparse version of the concentric blue-green circles seen in Figure 6b In addition, we demonstrate another way to show the internal structure by using the software Blender to remove a "pie slice" cut into a manifold see Figure 2; to accomplish this, we imported a premade stl into Blender and used Boolean operations to find the set difference of our manifold and a shape of our choosing—in this case, a solid pie-shaped wedge.

There are many other ways that one might consider performing the calculations, and we have in fact tried several other methods, with quite unsatisfying results. For example, we tried using built-in thickening methods in the software Blender to create the solid object, but invariant manifolds are sufficiently complex objects that in general we end up with a mesh file with significant structural errors that make it unprintable. We also tried to do the entire calculation in Matlab, computing the object using triangulation mesh and solidification methods that have been created for Matlab. Again here, the problem is that although in theory it appears to create a perfectly meshed object, we often find that it does not successfully print due to holes and incorrect normals as a result of meshing issues.

We now consider the specifics of how to print once the object has been designed, though this is not a completely separate issue, as usually design is done with ease of printing in mind. The choices described are made within the slicer software. We used the Cura slicer, but we have not found the choice of slicer to be critical. One principle to keep in mind is that it is much easier to print an object if it has a flat bottom rather than a round, nonuniform, or pointy bottom. For example, the Lorenz manifold in Figure 1 is printed with a base attached both for an attractive display and also so that it has a flat base for printing. The bottom manifold is already flat enough in the orientation shown that it does not have to print with a base. The objects in Figure 4 are printed in two halves so that each has a flat base and are glued together after the printing is complete. Another principle of printing is that if there are overhangs, where parts on top have nothing underneath, then the object will need to be printed with supports. In some cases, a clever choice of orientation will avoid too many supports, but with objects this complex, there will never be a way to avoid all supports. For a single extrusion printer, we have found tree supports to be the best choice as they are easier to remove. We have printed all of the depicted objects with a dual extrusion printer, meaning that for the single color objects, we use the second extruder for dissolvable filament. For example, without dissolvable filament, it would be close to impossible to make a good print of the designs used for Figure 2, since they have internal overhangs, and it would not be possible to remove internal supports.

We cannot use dissolvable filament in Figure 4, since in this case we are using the two colors to depict the two intersecting manifolds—which is why we are forced to print in two halves and glue them together. In particular, we have created an stl file for each separate manifold. To print both at the same time, we load both files into the slicer, selecting a different extruder to print each file.

As a final note on the printing process, there are a number of other 3D printing technologies that would yield excellent results in the printing of invariant manifolds. In particular, selective laser sintering and resin printing are higher cost methods that generally give much higher quality prints than the fused-deposition modeling (FDM) printers that we used. Such methods would be quite exciting to try, but we did not have access to these technologies.

## 6. List of Dynamical Systems

To demonstrate the effectiveness of our computational approach and visualization pipeline, we examine three dynamical systems of particular interest. These systems serve

[5] *For large values of $T$, we need to divide up and use the commands* `redistributePoints`, `Interpolation`, *and* `NDSolve` *$N$ times, each for time $T/N$.*

[6] *Units for 3D printing are millimeters.*

as compelling test cases, illustrating both the robustness of our method and its ability to reveal intricate geometric structures through three-dimensional visualization and physical reproduction.

**Example 6.1** (Stable manifold of the Lorenz system). The Lorenz system is a three-dimensional system of ordinary differential equations defined by the vector field

$$\dot{x} = \sigma(y - x), \quad \dot{y} = \varrho x - y - xz, \quad \dot{z} = xy - \beta z, \tag{10}$$

with standard parameters $\sigma = 10$, $\varrho = 28$, and $\beta = 8/3$, where we consider the equilibrium at the origin $p = (0, 0, 0)$. Originally developed by Lorenz (1963) as a simplified model of atmospheric convection, this system is a classical example of sensitive dependence on initial conditions, where nearby trajectories diverge dramatically under the flow. For the standard parameter values, trajectories converge to the famous butterfly attractor, also known as the Lorenz attractor, which occupies a small region in $\mathbb{R}^3$. This attractor is particularly intriguing from a geometric perspective, as it exists between a one-dimensional and two-dimensional object, possessing a fractal dimension that characterizes it as a strange attractor, making it a fundamental example in the study of chaotic dynamical systems.

To visualize the stable manifold of the Lorenz system at $p$ with classical parameters $\sigma = 10$, $\varrho = 28$, and $\beta = 8/3$, we first linearize, noticing that there are two stable eigenvalues $\lambda_1 \approx -22.8 < \lambda_2 \approx -2.7$ (as well as one unstable eigenvalue $\lambda_3 \approx 11.8 > 0$). Figure 6a shows how stretched the manifold would be if we did not use the arclength parameterization. The global manifold structure is computed up to $T = 180$ with 100 total intermediate time steps. The resulting 3D print is shown in Figure 3.

**Example 6.2** (Unstable manifold of the Arneodo–Coullet–Tresser system). The Arneodo–Coullet–Tresser system describes a three-dimensional dynamical system that exhibits Shilnikov chaos. The system is defined by the vector field

$$\dot{x} = y, \quad \dot{y} = z, \quad \dot{z} = -y - \beta z + \mu x(1 - x), \tag{11}$$

with the parameters $\beta = 0.4$ and $\mu = 0.863$ where Shilnikov chaos is known to occur. Originally proposed as a simple model demonstrating spiral chaos, this system is particularly notable for its explicit exhibition of Shilnikov-type homoclinic orbits. The system possesses two equilibria: a saddle-focus at the origin $p_1 = (0, 0, 0)$ and a second equilibrium point $p_2 = (1, 0, 0)$ that undergoes various bifurcations as parameters change, ultimately leading to the formation of a Shilnikov attractor.

To visualize the Shilnikov attractor of the Arneodo system at $\beta = 0.4$ and $\mu = 0.863$, we note that $DF(p_2)$ has unstable eigenvalues $\lambda_{1,2} \approx 0.1542 \pm 1.0930i$. Since these are complex conjugates, we will not need to use the arclength parameterization for our global manifold calculations. The global manifold is computed to $T = 22$. The resulting 3D printed manifold is shown in Figure 1.

**Example 6.3** (Stable and unstable manifolds of the Langford system). The Langford system describes a three-dimensional dynamical system that exhibits both Hopf and cusp bifurcations. The system is defined by the vector field

$$\begin{aligned} \dot{x} &= (z - \beta)x - \delta y \\ \dot{y} &= \delta x + (z - \beta)y \\ \dot{z} &= \tau + \alpha z - \frac{z^3}{3} - (x^2 + y^2)(1 + \varepsilon z) + \zeta z x^3, \end{aligned} \tag{12}$$

with classical parameters $\varepsilon = 0.25$, $\tau = 0.6$, $\delta = 3.5$, $\beta = 0.7$, $\zeta = 0.1$, and bifurcation parameter $\alpha > 0$. Originally derived by truncating a normal form of a simultaneous Hopf/cusp bifurcation to second order, the system includes an additional third-order term that breaks axial symmetry. The system serves as a model for dissipative vortex dynamics and rotating viscous fluids, featuring a $z$-axis invariant subsystem and rich dynamics including periodic orbits and multiple equilibrium solutions.

To visualize the intersection of global invariant manifolds in the Langford system with parameters $\alpha = 1.1022$, $\beta = 0.7$, $\delta = 3.5$, $\gamma = 0.6$, $\zeta = 0.1$, and $\varepsilon = 0.25$, we calculate that the system has three equilibria on the $z$-axis: $p_1 \approx (0, 0, 2.05)$, $p_2 \approx (0, 0, -1.43)$, and $p_3 \approx (0, 0, -0.61)$. For the equilibrium $p_1$, the linearization $DF(p_1)$ has eigenvalues $\lambda_{1,2} \approx 1.35 \pm 3.50i$ and $\lambda_3 \approx -3.08$, corresponding to a saddle-focus with a two-dimensional unstable manifold and a one-dimensional stable manifold. For the equilibrium $p_2$, the linearization $DF(p_2)$ has eigenvalues $\lambda_{1,2} \approx -2.13 \pm 3.50i$ and $\lambda_3 \approx -0.95$, indicating a sink with complex conjugate eigenvalues. The presence of complex conjugate eigenvalues implies that we will not need to parameterize with respect to arclength when computing the global manifold structure.

Figure 4 shows intersections of the resulting structures: the global unstable manifold of $p_2$ with $T = 9.5$ and the global stable manifold of $p_3$ with $T = 2.3$. We see that these two manifolds intersect transversally along curves.

Figure 2 shows 3D printed unstable manifolds for two different $\alpha$ parameter values. The yellow filamentary structure (in the right images) represents the 2D unstable manifold of the equilibrium point $p_1 \approx 1.84$ for $\alpha = 0.806$, while the purple surface shows the 2D unstable manifold of $p_1 \approx 1.94$ for $\alpha = 0.95$ generated using set difference techniques. The yellow invariant manifolds were computed using a numerical integration method that approximates the global invariant structure by iteratively applying the flow to an initial set of points along the local manifold, with interpolation between successive iterations.

## 7. Conclusions

This article has presented a comprehensive computational pipeline for visualizing and physically realizing invariant manifolds of dynamical systems through 3D printing. Our approach combines careful mathematical computation with modern manufacturing techniques, enabling both theoretical study and tactile exploration of these complex geometric structures. The methodology has been successfully demonstrated by computing and visualizing invariant manifolds associated with three dynamical systems: the stable manifold at the origin of the Lorenz system, the unstable manifold of a saddle-focus for the Arneodo–Coullet–Tresser system, and the intersecting stable and unstable manifolds of equilibrium solutions in the Langford system, showcasing the versatility and robustness of our approach.

The reproduction of these results begins with careful system definition and analysis. For any dynamical system of interest, one must first clearly define the vector field and identify relevant equilibrium solutions. The choice of system parameters significantly influences the manifold structure and should be selected based on the specific phenomena one wishes to study. Our computational pipeline then proceeds with the local manifold computation implemented in our provided Matlab/Mathematica scripts. This crucial step requires attention to numerical parameters, particularly the order of parameterization $N$ and the scaling factor, which should be adjusted based on the system's properties and desired accuracy.

Global manifold generation follows where the local manifold boundaries are evolved to capture the complete invariant structure. This phase demands careful monitoring of numerical stability during time integration and may require adaptive step-size control. The resulting geometric data undergoes appropriate transformations to ensure optimal visualization while maintaining mathematical accuracy. The process concludes with the preparation for 3D printing, where computed surfaces are exported to standard formats (stl or obj) and processed through slicing software such as Cura to generate the necessary G-code for physical realization. That is, the slicer creates precise instructions to the printer for how hot to heat the filament, the path of motion, and where to extrude the filament.

As discussed in Section 5, several challenges commonly arise during implementation. Numerical stability issues can emerge during the integration of highly sensitive systems, necessitating careful choice of tolerance parameters and integration schemes. Mesh quality must be maintained throughout the computation, often requiring smoothing algorithms that preserve geometric accuracy. The physical printing process presents its own challenges, particularly in optimizing model orientation to minimize support structures while ensuring structural integrity. These challenges can be addressed through careful parameter selection and monitoring of intermediate results at each stage of the pipeline.

## 8. Code Repository

All code implementing the algorithms and visualization pipeline described in this article is available in our open-source repository: `https://github.com/esander1789/3DPrintingInvariantManifolds`.

ACKNOWLEDGMENT. The authors thank the anonymous reviewer for their helpful comments and suggestions that improved this manuscript. Emmanuel Fleurantin was supported by the National Science Foundation under Grant No. DMS-2137947.

### References

[BDHHS26] David Bachman, Gabriel Dorfsman-Hopkins, Edmund Harriss, and Martin Skrodzki, *Rigorous illustrations—their creation and evaluation for mathematical research*, Institut Henri Poincaré, Paris, 2026, `https://indico.math.cnrs.fr/event/13124/`.

[BPB23] Francesca Bertacchini, Pietro S. Pantano, and Eleonora Bilotta, *Jewels from chaos: a fascinating journey from abstract forms to physical objects*, Chaos **33** (2023), no. 1, Paper No. 013132, 17, DOI 10.1063/5.0130029. MR4538264

[CFdlL03] Xavier Cabré, Ernest Fontich, and Rafael de la Llave, *The parameterization method for invariant manifolds. I. Manifolds associated to non-resonant subspaces*, Indiana Univ. Math. J. **52** (2003), no. 2, 283–328, DOI 10.1512/iumj.2003.52.2245. MR1976079

[CFMJ20] Maciej J. Capiński, Emmanuel Fleurantin, and J. D. Mireles James, *Computer assisted proofs of two-dimensional attracting invariant tori for ODEs*, Discrete Contin. Dyn. Syst. **40** (2020), no. 12, 6681–6707, DOI 10.3934/dcds.2020162. MR4160086

[CLMJ18] Roberto Castelli, Jean-Philippe Lessard, and Jason D. Mireles James, *Parameterization of invariant manifolds for periodic orbits (II): a posteriori analysis and computer assisted error bounds*, J. Dynam. Differential Equations **30** (2018), no. 4, 1525–1581, DOI 10.1007/s10884-017-9609-z. MR3871613

[DHVMZ16] Michael Dellnitz, Mirko Hessel-Von Molo, and Adrian Ziessler, *On the computation of attractors for delay differential equations*, J. Comput. Dyn. **3** (2016), no. 1, 93–112, DOI 10.3934/jcd.2016005. MR3565921

[FJ20] Emmanuel Fleurantin and J. D. Mireles James, *Resonant tori, transport barriers, and chaos in a vector field with a Neimark-Sacker bifurcation*, Commun. Nonlinear Sci. Numer. Simul. **85** (2020), 105226, 29, DOI 10.1016/j.cnsns.2020.105226. MR4067953

[Gag18] Michael Gagliardo, *3d printing chaos*, Proceedings of Bridges 2018: Mathematics, art, music, architecture, education, culture, 2018, pp. 491–494.

[GH83] John Guckenheimer and Philip Holmes, *Nonlinear oscillations, dynamical systems, and bifurcations of vector*

*fields*, Applied Mathematical Sciences, vol. 42, Springer-Verlag, New York, 1983, DOI 10.1007/978-1-4612-1140-2. MR709768

[HCF+16] Àlex Haro, Marta Canadell, Jordi-Lluís Figueras, Alejandro Luque, and Josep-Maria Mondelo, *The parameterization method for invariant manifolds: From rigorous results to effective computations*, Applied Mathematical Sciences, vol. 195, Springer, [Cham], 2016, DOI 10.1007/978-3-319-29662-3. MR3467671

[Hen05] Michael E. Henderson, *Computing invariant manifolds by integrating fat trajectories*, SIAM J. Appl. Dyn. Syst. **4** (2005), no. 4, 832–882, DOI 10.1137/040602894. MR2179490

[MJ18] J. D. Mireles James, *Validated numerics for equilibria of analytic vector fields: invariant manifolds and connecting orbits*, Rigorous numerics in dynamics, Proc. Sympos. Appl. Math., vol. 74, Amer. Math. Soc., Providence, RI, 2018, pp. 27–80. MR3792792

[JCW+22] Anketa Jandyal, Ikshita Chaturvedi, Ishika Wazir, Ankush Raina, and Mir Irfan Ul Haq, *3d printing—a review of processes, materials and applications in industry 4.0*, Sustainable Operations and Computers **3** (2022), 33–42.

[JS25] Jonathan Jaquette and Evelyn Sander, *Graduate summer school: Computer-assisted proofs in applied mathematics*, held July 7–18, 2025, `https://www.slmath.org/summer-schools/1107`.

[KKMJ18] William D. Kalies, Shane Kepley, and J. D. Mireles James, *Analytic continuation of local (un)stable manifolds with rigorous computer assisted error bounds*, SIAM J. Appl. Dyn. Syst. **17** (2018), no. 1, 157–202, DOI 10.1137/17M1135888. MR3749379

[KOD+05] B. Krauskopf, H. M. Osinga, E. J. Doedel, M. E. Henderson, J. Guckenheimer, A. Vladimirsky, M. Dellnitz, and O. Junge, *A survey of methods for computing (un)stable manifolds of vector fields*, Internat. J. Bifur. Chaos Appl. Sci. Engrg. **15** (2005), no. 3, 763–791, DOI 10.1142/S0218127405012533. MR2136745

[KOGV07] Bernd Krauskopf, Hinke M Osinga, and Jorge Galán-Vioque (eds.), *Numerical continuation methods for dynamical systems*, Springer, 2007.

[LAC17] Jian-Yuan Lee, Jia An, and Chee Kai Chua, *Fundamentals and applications of 3d printing for novel materials*, Applied Materials Today **7** (2017), 120–133.

[LST20] Stephen K. Lucas, Evelyn Sander, and Laura Taalman, *Modeling dynamical systems for 3D printing*, Notices Amer. Math. Soc. **67** (2020), no. 11, 1692–1705, DOI 10.1090/noti. MR4201908

[OK04] Hinke M. Osinga and Bernd Krauskopf, *Crocheting the Lorenz manifold*, Math. Intelligencer **26** (2004), no. 4, 25–37, DOI 10.1007/BF02985416. MR2104464

[Seg12] Henry Segerman, *3d printing for mathematical visualisation*, The Mathematical Intelligencer **34** (2012), no. 4, 56–62.

[SLR19] Nurhalida Shahrubudin, Te Chuan Lee, and RJPM Ramlan, *An overview on 3d printing technology: Technological, materials, and applications*, Procedia Manufacturing **35** (2019), 1286–1296.

[Zas20] Mark Zastrow, *The new 3d printing*, Nature **578** (2020), no. 7793, 20–23.

Patrick R. Bishop

Summer Chenoweth

Emmanuel Fleurantin

Alonso Ogueda-Oliva

Evelyn Sander

Julia Seay

## Credits

Figures 1–6 are courtesy of the authors.

Photo of Patrick R. Bishop is courtesy of Patrick R. Bishop.

Photo of Summer Chenoweth is courtesy of Meghan Chenoweth.

Photo of Emmanuel Fleurantin is courtesy of Janice Fleurantin.

Photo of Alonso Ogueda-Oliva is courtesy of Julia Seay.

Photo of Evelyn Sander is courtesy of Alice Wanner.

Photo of Julia Seay is courtesy of Julia Seay.