Active Manifolds: A non-linear analogue to Active Subspaces

Robert A. Bridges [1] **Anthony D. Gruber** [2 1] **Christopher R. Felder** [3 1] **Miki E. Verma** [1] **Chelsey Hoff** [1]

Abstract

We present an approach to analyze $C^1(\mathbb{R}^m)$ functions that addresses limitations present in the Active Subspaces (AS) method of Constantine et al. (2015; 2014). Under appropriate hypotheses, our Active Manifolds (AM) method identifies a 1-D curve in the domain (the active manifold) on which nearly all values of the unknown function are attained, and which can be exploited for approximation or analysis, especially when m is large (high-dimensional input space). We provide theorems justifying our AM technique and an algorithm permitting functional approximation and sensitivity analysis. Using accessible, low-dimensional functions as initial examples, we show AM reduces approximation error by an order of magnitude compared to AS, at the expense of more computation. Following this, we revisit the sensitivity analysis by Glaws et al. (2017), who apply AS to analyze a magnetohydrodynamic power generator model, and compare the performance of AM on the same data. Our analysis provides detailed information not captured by AS, exhibiting the influence of each parameter individually along an active manifold. Overall, AM represents a novel technique for analyzing functional models with benefits including: reducing m-dimensional analysis to a 1-D analogue, permitting more accurate regression than AS (at more computational expense), enabling more informative sensitivity analysis, and granting accessible visualizations (2-D plots) of parameter sensitivity along the AM.

1. Introduction

Scientists and engineers rely on accurate mathematical models to quantify the objects of their studies. Such models can be difficult to analyze because they appear as systems of implicitly defined equations, or they involve a high number of parameters relative to the quantity of data/observations. For example, regression to determine a high number of parameters requires a large amount of data, else the problem is under-determined. Similarly, models

are frequently subject to questions of sensitivity, and to prioritize next step research it is desirable to answer questions such as "According to the mathematical model, which parameters should we change to get the most response in the output?", e.g., (Constantine & Diaz, 2017; Constantine et al., 2014; 2015; 2016; Lukaczyk et al., 2014).

To this end, we consider the problem of analyzing a real-valued function that is C^1 (at least one continuous derivative) defined on a problematically high-dimensional domain, with the aim of performing dimension reduction and sensitivity analysis. One way to address this problem is to tailor methods of dimension reduction to the given model, in order to work in a smaller parameter space while preserving fidelity of the output. Essentially, this entails approximating the original model using fewer inputs.

Given a C^1 function, $f : U \subset \mathbb{R}^m \to \mathbb{R}$ with potentially large m, we fashion our approach on a simple concept from vector calculus. Suppose one is standing at a point $x_0 \in U$, observes the value $c = f(x_0)$, and takes a small step. Then, there are $m - 1$ orthogonal directions (potentially a huge subspace!) that one can step without changing the function (or more precisely, while exhibiting negligible change in f). Yet, there is one special direction, the gradient ∇f_{x_0}, and so long as the step is in this direction, the walker is guaranteed maximal change in f near x_0! More formally, we are simply stating that the m-dimensional tangent space to $\mathbb{R}^m$ at x_0 can be decomposed into an $(m-1)$-dimensional subspace of vectors tangent to the level set $\{x : f(x) = c\}$, and a one-dimensional space consisting of scalar multiples of the gradient ∇f_{x_0}. Armed with this perspective, we explore a powerful hypothesis—by exploiting this decomposition, we can reduce analysis of any $f \in C^1(\mathbb{R}^m)$ to a related $\hat{f} \in C^1(\mathbb{R})$, thereby allowing for analysis of arbitrarily high-dimensional models in a single dimension.

Contributions: We provide the mathematical foundation and pseudo-algorithm for a novel method of analyzing functions with a problematically high-dimensional domain. The method first recovers a curve in the domain, called the active manifold, and then reduces the problem to analysis of the function restricted to only this 1-D manifold by traversing level sets that necessarily intersect the active manifold orthogonally. E.g., see Figure 1.

This work is related to sliced inverse regression (Duan & Li, 1991; Li, 1991), but is directly inspired by and builds upon the the Active Subspaces (AS) method of Constantine (2015; 2014), and we adopt the name Active Manifolds (AM) in recognition. Constantine's AS method seeks an affine subspace inside the domain on which the function changes most on average. Unfortunately, AS does not guarantee there is a lower dimensional subspace to be produced by the algorithm (e.g., all m directions can contribute equally to the response as in $f(x) = |x|^2$). Furthermore, because AS is a summary analysis over the whole domain, the potential for losing important information about the function is great; specifically, the impact of particular parameters on a model can be "averaged out" by AS. Instead, we craft a non-linear analogue by considering iterative local analysis rather than global summary statistics. We provide a rigorous mathematical foundation informing a psuedo-algorithm for function approximation using AM.

In accordance with AS, we assume we can sample both the function value $f(x)$ and the gradient $\nabla f_x := \nabla f(x)$ (or are given a random sample), and consider the problems of regression and sensitivity analysis. We present experiments on known functions to illustrate the method in an easily understandable setting, producing regression results testing AM against AS. Further, we consider a real-world model of a magnetohydrodynamic (MHD) power generator. Following Glaws et al. (2017) who used AS for sensitivity analysis of the MHD model, we now apply AM to see what additional information we can extract. Our results show that, even with their selected model and data, AM offers distinct advantages over AS at the expense of more computation. In particular, we are able to improve estimation error as well as give a more detailed and interpretable characterization of the model's parameter sensitivity.

Benefits of the AM approach are fourfold: (1) Regardless of the input dimension m, AM reduces the problem to analysis of a one-dimensional (1-D) function. In particular, this relieves the burden of AS on the user to choose a suitable subspace for analysis. (2) In initial experiments we exhibit more accurate regression than the AS method with the same input data, though at greater computational expense. (3) Our method allows for finer and more informative sensitivity analysis by providing local rather than global rankings of the input parameters based on their influence. This permits segmenting the active manifold into regions where sensitivity is different. AM also allows the user to see the influence of each parameter *individually* along the manifold, contrasting with AS which gives sensitivity rankings globally in terms of a (perhaps physically meaningless) linear combination of parameters. (4) The 1-D approach of AM allows accessible visualization (2-D plots) to inform understanding of the high-dimensional function.

2. Related Works

Dimension reduction, broadly speaking, is the mapping of data to a lower dimensional space, with the goal of preserving and illuminating some desired characteristics by eliminating unneeded degrees of freedom. See Burges (2010) for an overview. Powerful and well-known techniques include Principal Component Analysis (Pearson, 1901), the Nyström method (Kumar et al., 2012), Isomap (Zhang et al., 2013), Diffusion Maps (De la Porte et al., 2008), and Norm Discriminant Manifold Learning (Liu et al., 2018).

Sliced Inverse Regression (SIG) (Duan & Li, 1991; Li, 1991) and later refinements, e.g., (Coudret et al., 2014; Li & Nachtsheim, 2006), are related to our AM method. The main idea is to model $y : \mathbb{R}^m \to \mathbb{R}$ as $y = f(Bx + \epsilon)$ for unknown f and Gaussian noise ϵ. The goal is to learn B, an $m \times m$ matrix with rank $k \leq m$, which gives a lower dimensional subspace on which y is recoverable. Roughly speaking, Li provides a theorem that states $x \sim N(0, I)$ yields $\mathbb{E}(x|y)$ is in range(B), which informs an algorithm: normalize x; empirically estimate $\mathbb{E}(x|y)$; use the SVD to recover B from the k most significant directions.

Closely related to SIG and a primary driver for this work is Active Subspaces (AS), an idea originally of Russi (2010) but developed extensively by Constantine et al. (2015; 2014). See Sec. 2.1. Many applications have been found for AS, e.g., shape optimization (Lukaczyk et al., 2014), MCMC for Bayesian inverse problems (Constantine et al., 2016), and sensitivity analysis (Constantine & Diaz, 2017).

Emerging research of Zhang & Hinkle (2019) builds on the AM idea of this paper by using ResNets to learn a (generally) non-linear, lower-dimensional transformation of the input variable x with minimal reconstruction error of both the desired function and gradient. This can be considered an analogue to AM with tunable dimension, by using the ResNet statistical machinery to learn the manifold.

A common challenge of the above methods is deciding the dimension of the reduced space. It is often necessary to inspect eigenvalues and make an educated guess about what dimension is needed to capture important information. AM avoids this issue by projecting the relevant features to 1-D in every case. Although the work presented here departs from the use of traditional projective and spectral methods, it is still a manifold modeling method.

2.1. Active Subspaces (AS)

Developed by Constantine et al. (2014), AS is a dimension reduction technique that is both applicable to a wide class of functions (those with C^1 regularity) and accessible to scientists and engineers with limited mathematical background. Even better, AS is fast, because it focuses on affine approximation where linearity can be exploited. In partic-

ular, AS looks for a lower-dimensional affine subspace inside the domain of a function f by computing the directions in which f changes the most on average. As nice as this is, the AS method has three main limitations: first, the inherent restriction present in considering only affine subspaces can create large errors, as we will see in later examples; secondly, visualization can be an issue with this method, since the active subspaces themselves are not guaranteed to be low-dimensional; finally, many functions do not admit an active subspace, e.g. $f(x) = |x|^2$ as mentioned earlier.

Below is a heuristic description of the AS algorithm. See (Constantine, 2015; Constantine et al., 2014) for more details. Assume $f \in C^1([-1, 1]^m)$.

1. Sample $\nabla f_{a_i} := \nabla f(a_i)$ and $f(a_i)$ at N random points $a_i \in [-1, 1]^m$.

2. Compute the eigenvalue decomposition of the matrix
$$\mathbf{C} = \frac{1}{N} \sum_{i=1}^{N} \nabla f_{a_i} \nabla f_{a_i}^T = \mathbf{W \Lambda W}^T.$$

3. Manually inspect the set of eigenvalues $\{\lambda_i\}$ for "large" gaps. If there is a gap between λ_j and λ_{j+1}, we say $S := \text{Span}(\{w_1, ..., w_j\})$ is the "active subspace" of dimension j associated to f, where the w_i are the corresponding eigenvectors.

4. Given an arbitrary point $p \in [-1, 1]^m$, project p orthogonally to $p' := Proj_S(p)$ on the active subspace.

5. Define $\hat{f} : S \to \mathbb{R}$ by $\hat{f}(x) := \text{ave}\{f(a_i) : Proj_S(a_i)$ near $x_0\}$ and obtain the approximation $f(p) \approx \hat{f}(p')$.

The AS algorithm also allows for a bootstrapping procedure, in which active subspaces are computed from random partitions of the input data. Comparing the range of these to the full active subspace gives "confidence" values that are usually plotted along with the values of f on the active subspace. See Constantine (2015) for more details.

3. Active Manifolds (AM)

Here we provide the mathematical foundation for AM and describe a pseudo-algorithm for reducing analysis of the m-dimensional function to its one-dimensional analogue. Examples to illustrate the method are provided, including illustrations of problems or obstructions identified.

3.1. Mathematical Justification:

Recall that the arc length of a C^1 curve $\gamma(t) : [0, 1] \to \mathbb{R}^m$ is given by $S(\gamma) = \int_0^1 |\gamma'(t)| \, dt$. Let $U \subset \mathbb{R}^m$ open and assume $f \in C^1(U)$.

We seek
$$\arg\max \int_0^1 \langle \nabla f(\gamma(t)), \gamma'(t) \rangle \, dt$$

over all C^1 curves $\gamma(t) : [0, 1] \to U$, such that $|\gamma'| = 1$ (constant speed), where $\langle \cdot, \cdot \rangle$ denotes the usual Euclidean inner product. Note that the integrand satisfies

$$\langle \nabla f(\gamma(t)), \gamma'(t) \rangle = |\nabla f(\gamma(t))| \, |\gamma'(t)| \cos \theta$$

where θ is the angle between $\nabla f(\gamma(t))$ and $\gamma'(t)$. Clearly this quantity is maximal when $\theta = 0$, indicating that $\nabla f(\gamma(t))$ and $\gamma'(t)$ point in the same direction; hence, the solution to this optimization problem is

$$\gamma'(t) = \frac{\nabla f(\gamma(t))}{|\nabla f(\gamma(t))|} \,, \tag{1}$$

a constant-speed streamline of ∇f. Specifying a starting point, γ_0, uniquely identifies the flow as furnished by the following standard theorem of differential equations.

Lemma 3.1. *Given $f : U \subset \mathbb{R}^m \xrightarrow{C^1} \mathbb{R}$ and an initial value $\gamma_0 \in U$, there exists a unique local solution $\gamma(t)$ to the system of first-order ordinary differential equations described by (1).*

Proof. Choose any compact and convex subset $K \subset U$ containing γ_0. Since f is C^1, ∇f satisfies the Lipschitz condition, $|\nabla f(x_1) - \nabla f(x_2)| \leq L_K |x_1 - x_2|$, for $x_1, x_2 \in K$ and $L_K < \infty$ is some Lipschitz constant. By Theorem 1 Ch. 6 from Birkhoff & Rota (1969), these conditions are sufficient for the existence and uniqueness of a local solution $\gamma(t)$ to Eqn. (1) about γ_0 in K, which can be reparametrized to have domain $[0, 1]$ as desired. Since K was an arbitrary compact set we have the result. $\square$

Definition 3.2. Let $f : U \subset \mathbb{R}^m \xrightarrow{C^1} \mathbb{R}$. We say that $\mathcal{M} \subset U$ is an **active manifold** defined by f provided there exists a constant-speed parametrization of $\mathcal{M}$, $\gamma(t) : [0, 1] \to U$, such that condition (1) is satisfied for all $t \in [0, 1]$.
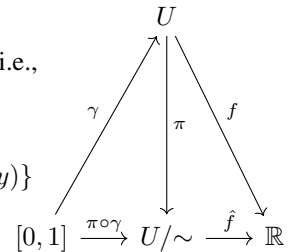
For the following proposition, let f, U be as above and:

- $\mathcal{M} = \text{Im} \, \gamma(t)$ an active manifold of f

- The relation $\sim$ defined by f, i.e., $\forall x, y \in U$, $x \sim y \iff f(x) = f(y)$

- $[x] = \{y \in \mathbb{R}^m : f(x) = f(y)\}$

- $\mathbb{R}^m / \sim = \{[x] : x \in \mathbb{R}^m\}$

- $\pi : \mathbb{R}^m \twoheadrightarrow \mathbb{R}^m / \sim$

$$[0, 1] \xrightarrow{\pi \circ \gamma} U/\sim \xrightarrow{\hat{f}} \mathbb{R}$$

Proposition 3.3. *If $\gamma(t)$ is a solution to Eqn. (1) on an open set U away from points where $\nabla f = 0$, then the following statements hold.*

(i) *$\mathcal{M}$ is an immersed C^1 submanifold of $U \subseteq \mathbb{R}^m$.*

(ii) *$U/\sim$ is a C^1 manifold.*

(iii) $\pi \circ \gamma$ *is a* C^1 *embedding of* $\mathcal{M}$ *in* $\mathbb{R}^m/\sim$.

(iv) $f \circ \gamma : [0,1] \to \mathbb{R}$ *is strictly increasing.*

Proof. (i): Note that $f|_{\mathcal{M}}$ provides a global C^1 chart for $\mathcal{M}$. Further, $\mathcal{M}$ is immersed since $|\gamma'| = 1$, hence γ' does not vanish. (ii): Since f is C^1 and constant on the fibers of π, the map $\hat{f} : U/\sim \to \mathbb{R}$ defined as $\hat{f}([x]) := f(x)$ is C^1. So $U/\sim$ is a C^1 manifold with global chart $\hat{f}$. (iii): $\pi|_{\mathcal{M}}$ is a bijection onto $\pi(\mathcal{M})$ since $\mathcal{M}$ fibers pointwise under π. Since π is linear, it follows that $d\pi|_{\mathcal{M}} = \pi|_{\mathcal{M}}$ is bijective; hence, π is an embedding. (iv): Monotonicity of $f \circ \gamma$ follows directly from the definition: $\forall t \; \gamma(t) \parallel \nabla f(t)$. $\square$

Theorem 3.4. *Suppose the level set* $\{f = \alpha\}$ *is connected and* γ *is any active manifold such that* $\alpha \in Im(f \circ \gamma)$. *Then* $\exists! \; t_0$ *such that* $\gamma \cap \{f = \alpha\} = \{\gamma(t_0)\}$, *and* $\gamma \perp \{f = \alpha\}$.

Proof. The Implicit Function Theorem guarantees that for each $\alpha \in Im f$, the level set $\{x : f(x) = \alpha\}$ is an $(m-1)$-dimensional submanifold of $\mathbb{R}^m$ that is orthogonal to the gradient vector field and therefore to any intersecting active manifold. By hypothesis $\exists \, t_0$ such that $\gamma(t_0) \in \{f = \alpha\}$. Uniqueness follows from monotonicity of $f \circ \gamma$ (Proposition 3.3.*iv*). $\square$

Implication: This theorem implies that if one can recover $f \circ \gamma$ (a 1-D regression problem), then one can recover f on the connected component of any level set touching γ. Concisely, if p is in the component of $A := \{f = f(p)\}$ intersecting γ, one may move freely in the $(m-1)$-dimensional submanifold A transverse to γ without changing f. This motivates our AM pseudo-algorithm.

3.2. Active Manifolds Pseudo-Algorithm:

The AM algorithm has three broad components: (1) Build the active manifold $\mathcal{M} = Im(\gamma(t))$; (2) Approximate the function of interest f on $\mathcal{M}$ with $\hat{f}$; (3) For $p \in U$ traverse the level set to $\mathcal{M}$ to estimate $f(p)$. We require two parameters: δ, a step size for the numerical approximation of paths, and ϵ, a tolerance for when to terminate walking. We now discuss each component in detail.

3.2.1. BUILDING THE ACTIVE MANIFOLD:

For the algorithm, we consider f on the hypercube $U = [-1,1]^m$, assuming one has pre-composed f with a scaling function on a portion of the original domain if necessary. Given starting value $p \in U$, we describe the process of building the active manifold, γ, which will be a one-dimensional curve in $[-1,1]^m$ that moves from a local minimum of f to a local maximum. As are the assumptions of AS, we require observational data of f and ∇f at some samples in the domain. We first build a uniform grid with spacing size ϵ and compute $f, \nabla f$ at each grid point. We then use a gradient ascent/descent scheme with

nearest neighbor search to construct the active manifold. We set the step size $\delta = 2d/3$ where d is the longest diagonal of the hypercubes in our sampled grid. *Note that a grid is not necessary, as one could easily rework the algorithm to accommodate other initial data, e.g., a given set of observations in the hypercube. See Algorithm Note 3.3.1.2.*
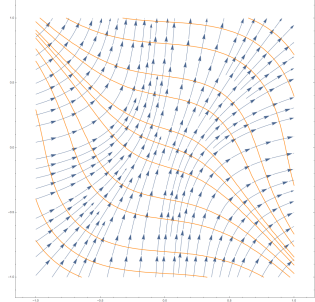

Figure 1. Level sets (orange) of example function $f_3 : \mathbb{R}^2 \to \mathbb{R}$, and its gradient vector field (blue) tangent to the AMs at each point.

1. Build a uniform grid $\mathbf{P} = \{p_i\}$ over $[-1,1]^m$ with spacing size ϵ. Sample f and ∇f at each grid point. See Algorithm Note 3.3.1.1.

2. Given an initial starting point $\gamma_0 \in U$, find the nearest p_i, and set $\gamma_1 := \gamma_0 + \delta \, \nabla f(p_i)/|\nabla f(p_i)|$. Continue this gradient ascent/descent scheme in both directions to find a numerical solution to $\gamma'(t) = \nabla f(\gamma(t))/|\nabla f(\gamma(t))|$ using the samples $\{p_i, f(p_i), \nabla f_{p_i}\}$ from step 1. The algorithm in each direction ends when either the next step would exit $[-1,1]^m$ or would become close to a previous step (we use $\delta/3$ as the closeness parameter). The set $\mathcal{M} = \{\gamma_i\}_i$ is then a discretized active manifold in $[-1,1]^m$. See Algorithm Note 3.3.1.3.

3. Finally we parameterize γ on $[0,1]$ as follows: While the active manifold is built, save the number of steps $0 \le i \le N$ and the function value $z_i = f(\gamma_i)$ at each step. Use this to construct ordered lists $\mathbf{S} := \{i = 0, \dots, N\}$ and $\mathbf{Z} := \{z_i\}$. Scale $\mathbf{S}$ so that $\mathbf{S} := \{i/N\}$ and the domain of γ is $[0,1]$. Note that such a parameterization is necessarily constant-speed.

3.2.2. APPROXIMATING f WITH $\hat{f}$:

To obtain a one-dimensional approximation $\hat{f} \approx f$ defined on the whole of $[0,1]$, we fit the data $\{(i/N, z_i) \in \mathbf{S} \times \mathbf{Z}\}$ to a piecewise-cubic Hermite interpolating polynomial $\hat{f} : [0,1] \to \mathbb{R}$. Note that $\hat{f}$ is globally C^1 and monotone. This furnishes a benefit of our method—$\hat{f}$, which represents f along the active manifold, can be plotted for a useful 2-D visualization of the data, e.g., Figures 4 (left) & 5.

3.2.3. TRAVERSING THE LEVEL SET:

Given a point $p \in \mathbb{R}^m$, we compute $\hat{f}([p])$ by finding $\gamma(t) \in \mathcal{M}$ such that $[p] = [\gamma(t)]$. This requires an iterative process that uses vectors orthogonal to ∇f_p to travel along the level set through $f(p)$ until we intersect the active manifold, γ. For the following algorithm, we assume that
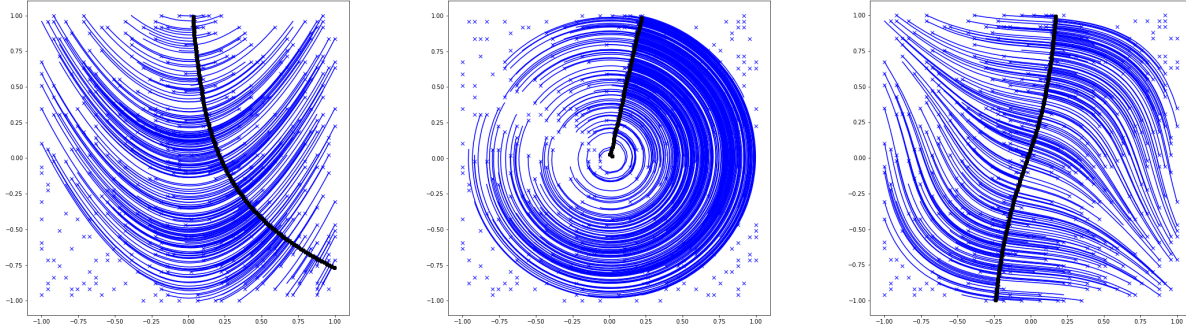
-

Figure 2. From left to right, AM run on test functions f_1, f_2, f_3, respectively. Black bold line is the approximated active manifold, γ, from a random starting point as used in tests of Section 4.1. Test data indicated with blue $\times$, with path traversed to the active manifold in blue. Test data with no path indicates those for which the learned active manifold cannot provide an estimate.
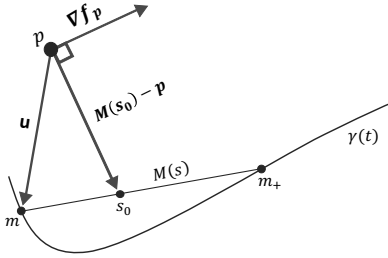

Figure 3. Schematic for the Level Set Algorithm in $\mathbb{R}^2$ when starting point p is one step from active manifold, γ. Notation in schematic matches pseudo-algorithm.

∇f has been normalized to unit length, and tolerance ϵ and step size δ have been specified.

1. From $p \in \mathbb{R}^m$, we must identify the direction $\mathbf{v}$ in which to step. First, we try to step greedily toward the active manifold while remaining on the level set. Let $m :=$ $\arg\min |p - m'|$ for $m' \in \mathcal{M}$ (the closest point on the $\mathcal{M}$ to p) and set $\mathbf{u} := (m - p)/|m - p|$ (the normalized vector from p to m). We wish to step in the direction of

$$\mathbf{v} := \mathbf{u} - \langle \mathbf{u}, \nabla f_p \rangle \nabla f_p, \qquad (2)$$

which is the component of $\mathbf{u}$ tangent to the level set $f^{-1}(p)$. If $\mathbf{u}$ is not nearly colinear with ∇f_p (so that $\mathbf{v}$ is not nearly $\vec{0}$), we keep this $\mathbf{v}$. Usually this is the direction taken for a step. On the other hand, if $\mathbf{v} \approx \vec{0}$, walking toward the active manifold would require stepping in line with ∇f, i.e., up/down hill, which we do not permit. See Algorithm Notes 3.3.1.4 for an if/else loop to define a suitable direction in this case.

2. Step, $p \mapsto p + \delta \mathbf{v}$. We continue the walk according these first two steps until either (a) we are close to the manifold ($|p - m| \leq \epsilon$), (b) we step out of the hypercube ($p \notin [-1, 1]^m$) or (c) the algorithm loops back on itself ($|p - q| \leq \delta/3$ for some previous step q).

3. If $|p - m| \leq \epsilon$, parameterize the line segment, $M(s)$, between m and m_+, where m_+ is the next closest point on the manifold to p. See Figure 3 and Alg. Note 3.3.1.5.

    3.1. Determine s_0 such that $(M(s_0) - p) \perp \nabla f_p$.

    3.2. Evaluate $\hat{f}(\gamma(t_0)) = \hat{f}(M(s_0)) \approx f(p)$, where t_0 is the corresponding point in $[0, 1]$ to $M(s_0)$ via γ.

    Else, the traversal along the level set exited the hypercube (or in rare cases self intersected). For these points the chosen active manifold cannot provide an estimate.

We note that compactness of $[-1, 1]^m$ implies the algorithm necessarily terminates after finite steps based on our stopping conditions. For points p near $\mathcal{M}$, continuity of ∇f implies our algorithm will walk from p approximately along a level set to the necessary intersection with the $\mathcal{M}$.

3.3. Challenges

Some active manifolds are better choices then others. There are regions in the hypercube that will not admit an approximation by the algorithm above, specifically if p is on a level set that is either disconnected or exits the hypercube before reaching an approximated active manifold. In this case, no approximation is guaranteed; e.g., see Figure 2. In testing, we note that the regions of the hypercube for which our algorithm does not produce an approximation give insight into where it is most useful to build a second active manifold for further approximation. Hence, iteratively running this algorithm allows for a "smart" choice for subsequent active manifold starting points. Bootstrapping techniques are commonly used for this purpose, placed at the beginning of algorithms to avoid problematic starting configurations. This may provide a worthwhile addition in our case as well.

3.3.1. ALGORITHM NOTES

1. We have avoided discussing critical points and singularities in the domain of f, because we do not yet have effective methods to deal with these obstructions. Currently, it is possible for the algorithm to "get stuck" near a critical point c_0, forcing the user to restart the algorithm on the "other side" of c_0 in order to get a complete

active manifold through the domain. With that said, this is not difficult to avoid in practice, and we have written our implementation so that there is no practical risk of the algorithm running indefinitely near these points.

2. In practice, one may be given a non-uniform sampling $\{a_i\}_{i=1}^n \subset \mathbb{R}^m$, along with corresponding values $f(a_i)$ and ∇f_{a_i}, instead of values on the parameter grid. In this case, the data is scaled linearly to the cube $[-1, 1]^m$ and the algorithm is applied as usual.

3. One may be enticed to minimize a distance function $D(t) = |\gamma(t) - p|^2$ through differentiation, i.e.

$$0 = \frac{1}{2}\frac{d}{dt}D(t) = \langle \gamma(t) - p, \gamma'(t) \rangle \qquad (3)$$

But this is not computationally efficient, as we must evaluate both $\gamma'(t)$ and $\gamma(t) - p$. Instead, we recommend computing the values of $|\gamma(t) - p|^2$ and searching for the minimum using another, more efficient algorithm.

4. In the case that $\mathbf{v} \approx \vec{0}$ we proceed in order through the following alternative definitions of $\mathbf{v}$:

    4.1. If $p - m$ is colinear with ∇f_p, we try to leverage momentum, stepping in roughly the direction of the previous step. Set $\mathbf{u} := (p_i - p_{i-1})/|p_i - p_{i-1}|$, and $\mathbf{v}$ as in Eq. 2. If this is the first step or $\mathbf{v} \approx \vec{0}$ still, proceed to the next bullet.

    4.2. Next, we attempt to step towards the origin to prevent walking out of the hypercube. Redefine $\mathbf{u} = p - \vec{0}$ and redefine v from Eqn. 2.

    4.3. If $v \approx \vec{0}$ still, choose an arbitrary vector in $\nabla f_p^{\perp}$.

5. We may express $M(s) = (m_+ - m)s + m$, $s \in [0, 1]$ so that the point s_0 on M for which $(M(s_0) - p)) \perp \nabla f_p$ can be determined by solving for s in

$$\langle (m_+ - m)s + (m - p), \nabla f_p \rangle = 0.$$

  We then have

$$s_0 = \frac{\langle p - m, \nabla f_p \rangle}{\langle m_+ - m, \nabla f_p \rangle}.$$

Finally, recall m, m_+ are the $k^{th}, (k+1)^{st}$ steps along the γ, respectively, (Sec. 3.2.1), and m, m_+ are identified with $k/N, (k+1)/N \in [0, 1]$. To apply the spline approximation $\hat{f}(\gamma(t))$, form the bijective linear map $[k/N, (k+1)/N] \mapsto [m, m_+]$ and choose $t_0 \in [k/N, (k+1)/N]$ corresponding to $s_0 \in [m, m_+]$.

4. Examples & Experiments

For proof of concept and comparison to AS method in Constantine 2015, we apply both our AM and AS to data syn-

thesized from the following two-dimensional test functions

$$f_1(x, y) = e^{y - x^2} \qquad (4)$$
$$f_2(x, y) = x^2 + y^2 \qquad (5)$$
$$f_3(x, y) = x^3 + y^3 + 0.2x + 0.6y. \qquad (6)$$

These functions were chosen to illustrate benefits/problems of the method and to be easy to understand and visualize to facilitate verification and validation. We also consider a model magnetohydrodynamic (MHD) power generator considered by Glaws et al. 2017 to which they applied AS, to see what extra information we can extract with AM. Results code on https://github.com/bridgesra/active-manifold-icml2019-code.

4.1. Two-dimensional Test Functions:

We are interested in how well the AS and AM approximations recover the values of a function for arbitrary points inside the domain. For each example function, we approximate with AS and AM and calculate the average error over a set of random test points. The following steps were followed for each experiment:

1. A uniform grid $P = \{p_i\}$ of 10K points was built on $[-1, 1]^2$, with a random 80/20% partition for training/testing.

2. The values $f(p_i)$ and ∇f_{p_i} were computed analytically at each grid point, and gradients were normalized so that $|\nabla f_{p_i}| = 1$.

3. The active subspace and active manifold were built from the training data $\{p_i, \nabla f_{p_i}\}$, using Constantine's software package (Constantine, 2016a) for the AS, and our algorithm for the AM. (Note that AS necessarily requires the original, unnormalized gradients in this step.)

4. The approximate function $\hat{f}$ was fit to the resulting data in each case. For AM, a piecewise-cubic Hermite interpolation to f was used. For AS, Constantine's own optimization algorithm was applied, trained with 100 bootstrap replicates and using a degree-4 polynomial approximation. See Figure 4.

5. The 2000 testing points were projected for AS orthogonally to the active subspace, and for AM to the active manifold using our algorithm 3.2.3. The approximation $\hat{f}$ was then applied to the projected points to generate the values $\hat{f}(p_i)$ for p_i in the training set.

6. The average absolute (ℓ^1) error and average ℓ^2 approximation errors in f were computed for each method by comparing the approximate values coming from $\hat{f}$ against the known analytic values from f.

See Table 1 for the results. Note that AM reduces the average absolute and average ℓ^2 errors by at least an or-

		ℓ^1 **mean**	ℓ^1 **std**	ℓ^2 **mean**	ℓ^2 **std**	n/N **mean**
f_1	AM	6.739E-3	6.826E-4	1.879E-4	1.847E-5	86.7%
	AS	0.585	8.130E-3	0.751	8.600E-3	100%
f_2	AM	0.0158	9.697E-4	4.015E-4	2.562E-5	77%
	AS	0.395	5.484E-3	0.488	6.890E-3	100%
f_3	AM	0.0106	8.442E-4	3.154E-4	2.887E-5	92.9%
	AS	0.982	0.018	1.22	0.0224	100%

Table 1. Regression results for AM and AS run on test functions f_1, f_2, f_3. Mean and standard deviation reported across 9 runs (3 train/test splits $\times$ 3 initial AM starting points) for average ℓ^1, ℓ^2 errors. Also we report the ratio n/N of test points for which the algorithm successfully found an approximation. See Figure 2 for a visualization of one AM run per function.

der of magnitude over AS in each case. Our initial implementation is computationally naive and requires more computation than Constantine's AS package (Constantine, 2016a). In particular, AM was on average an order of magnitude slower than AS for these 2-D functions, and this gap increased when testing on a higher dimensional grid of points. Further, due to the nonlinear nature of AM, we suspect that AM will never be as fast as AS, though it could be sped up significantly with algorithmic engineering. We provide initial performance results of AM and AS in the Supplemental Section.
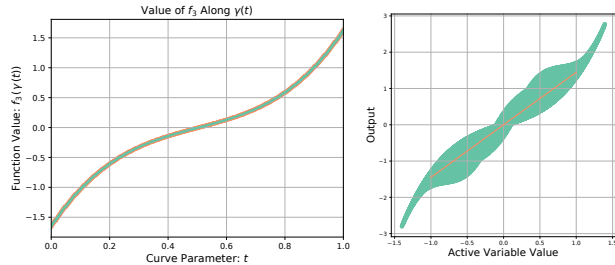

Figure 4. (Left) *AM*: Plots of f_3 along the active manifold (green) and piecewise-cubic Hermite spline approximation from AM (orange). (Right) *AS*: Plots of f_3 along the active subspace and bootstrap replicates (green), and Constantine's optimization algorithms were applied to fit a degree 4 polynomial (orange).

4.2. MHD Power Generator Model

We now revisit the work of Glaws et al. (2017), which applied AS to a model for magnetohydrodynamic (MHD) power generation, and consider the application of AM for sensitivity analysis.

4.2.1. THE HARTMANN PROBLEM

We first consider the so-called Hartmann problem, which models laminar flow between two parallel plates. Following Glaws et al. (2017), we examine separately the average flow velocity u_{avg} and the induced magnetic field B_{ind}, whose solutions are given analytically in terms of the parameters summarized in Table 2. In particular,

$$u_{avg} = -\frac{\partial p_0}{\partial x}\frac{\eta}{B_0^2}\left(1 - \frac{B_0 l}{\sqrt{\eta\mu}}\coth\left(\frac{B_0 l}{\sqrt{\eta\mu}}\right)\right),$$

$$B_{ind} = \frac{\partial p_0}{\partial x}\frac{l\mu_0}{2B_0}\left(1 - 2\frac{\sqrt{\eta\mu}}{B_0 l}\tanh\left(\frac{B_0 l}{2\sqrt{\eta\mu}}\right)\right).$$

To analyze this probem using AM, the following experiment was conducted analogous to Glaws et al.:

1. The cube $[-1,1]^5$ was discretized with a uniform grid of 14^5 evenly-spaced points, and these points were mapped linearly through a dilation map D onto their appropriate ranges as given in Table 2.

2. Function values u_{avg}, B_{ind}, and their gradients were computed analytically on these inputs from the formulae provided by Constantine (2016b).

3. The computed gradients were mapped back to the cube $[-1,1]^5$ through D^{-1} (taking the chain rule into account) and normalized to unit length.

4. The AM algorithm was run on this data from random seed 46 with $\delta = \epsilon = 0.02$.

The fit is nearly exponential in both cases—see the top row of Figure 5—which is expected given that $\coth x$ and $\tanh x$ are rational functions of an exponential variable. We also begin to see why one may prefer our method to AS in some situations, as the top row of Figure 6 shows that there is nonlinear behavior in the derivatives along the active manifold γ_{HB} corresponding to B_{ind} that will be missed with an affine model like AS. Note that Glaws et al. remark in (2017) that for both quantities of interest, a 2-D affine subspace is sufficient to almost completely characterize the output, which is believable given our results.

However, we also see from Figure 6 that the relative influence of the parameters on B_{ind} changes for parameter configurations near the last quarter of the active manifold. Indeed, the applied pressure gradient (blue) begins to overtake both the resistivity (pink) and the previously-dominant applied magnetic field (lime). This behavior is reasonable, as the $\tanh(x)$ function multiplying η "levels off" as $|x|$ increases, so the $\partial p_0/\partial x$ term in the equation for B_{ind} begins to take precedence.

For further analysis, we also compared AS to AM on this data—following the same procedure as for the 2-D test

Variable	**Notation**	**Range (Hartman)**	**Range (MHD)**
Fluid Viscosity	$\log(\mu)$	$\log([.05,.2])$	$\log([.001,.01])$
Fluid Density	$\log(\rho)$	$\log([1,5])$	$\log([.1,10])$
Applied Pressure Gradient	$\log(\frac{\partial p_0}{\partial x})$	$\log([.5,3])$	$\log([.1,.5])$
Resistivity	$\log(\eta)$	$\log([.5,3])$	$\log([.1,10])$
Applied Magnetic Field	$\log(B_0)$	$\log([.1,1])$	$\log([.1,1])$
Magnetic Constant	μ_0	fixed at 1	fixed at 1
Length	l	fixed at 1	fixed at 1

Table 2. Parameters and ranges for the Hartmann and idealized MHD problem reproduced from Glaws et al. (2017). The first five are variable, while the others are fixed at 1.
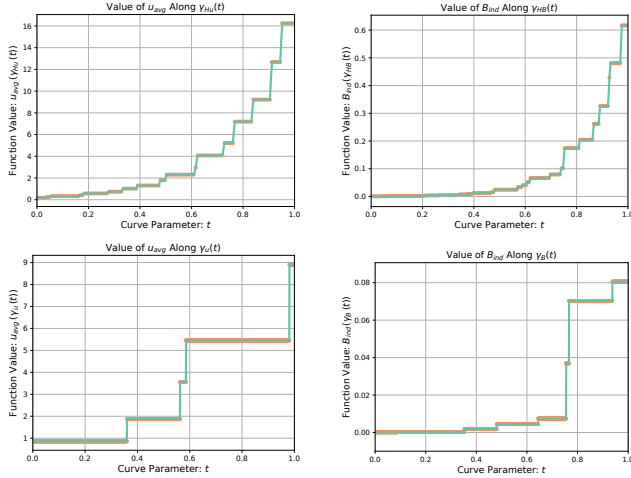
Figure 5. AM-derived function values in orange along parameterization of the active manifold and piecewise-cubic Hermite interpolation fit to these points in green. Top plots depict u_{ave} (left), B_{ind} (right) corresponding to the Hartmann problem (Sec. 4.2.1). Bottom plots depict u_{ave} (left), B_{ind} (right) corresponding to the idealized MHD generator data (Sec. 4.2.2).
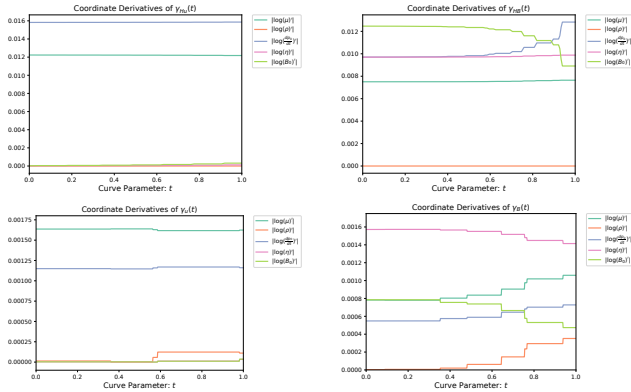

Figure 6. Plots of the magnitude of each parameter (partial) derivative along the active manifolds $\gamma_{(\cdot)}$. Top: γ_{Hu} (left) and γ_{HB} (right) corresponding to the Hartmann problem (Sec. 4.2.1). Bottom: γ_u (left) and γ_B (right) corresponding to the idealized MHD model (Sec. 4.2.2). Note that the AM permits analysis of sensitivity of input variables and especially their changes along the active manifold, which is not possible with AS.

functions. We constructed a uniform grid of 100K points on $[-1, 1]^m$, and ran AM using stepsize 0.15 over three random 98K / 2K test/train splits, each with three initial AM start points. We then ran AS on the same data using Constantine's software package (Constantine, 2016a), along with his optimization algorithms as before. Results are displayed in Table 3, again showing increased accuracy of AM over AS.

4.2.2. IDEALIZED MHD GENERATOR

The situation becomes even more interesting when we apply AM to the set of data provided by Glaws et al. (2017).

	B_{ind}		u_{avg}	
	AM	AS	AM	AS
ℓ^1 mean	0.0367	0.154	1.09	4.87
ℓ^1 std	3.063E-3	5.883E-3	0.255	0.103
ℓ^2 mean	1.286E-3	0.244	0.033	7.02
ℓ^2 std	3.415E-4	0.0116	3.774E-3	0.163

Table 3. Hartmann data average approximation errors reported. As expected, we see a large improvement with AM for B_{ind} where there is nonlinear behavior and less improvement for u_{avg} where parameter influence remains static.

This data is used to study a model for idealized 3D duct flow through a MHD generator. It involves the same parameters considered in the Hartmann problem, but with different ranges, as found in Table 2. The procedure for processing the 483 samples of inputs and outputs was similar to that for the Hartmann problem. The AM algorithm was applied with random seed 46 and step sizes $\delta = \epsilon = 0.002$.

Examining the plot in Figure 6 of the derivatives along the active manifold, γ_B, we see a large amount of change in the influence of each parameter throughout the parameter space. Fluid viscosity (blue-green) and applied magnetic field (lime) have the same amount of effect on the output B_{ind} at the beginning of γ_B, but from the middle to end of γ_B, fluid viscosity increases influence while applied magnetic field decreases. Further, applied pressure gradient (blue) overtakes applied magnetic field. Overall, this shows the influence of each parameter and how it varies along the active manifold. This represents an important case of behavior that is undetectable by AS. Since AS produces global sensitivity rankings as linear combinations of parameters, AS fails to capture the behavior of each parameter individually or changes in influence through the space.

5. Conclusions

We provide mathematical background and introduce a novel algorithm, AM, for analyzing C^1 functions f with a problematic ratio of inputs to observations. Leveraging initial samples of f and ∇f, our algorithm computes an approximate gradient streamline and recovers the function on this 1-D submanifold. Specifically, from a given point of interest p, the algorithm traverses the level set of f through p, intersecting the active manifold at a point that allows for accurate approximation. We provide initial tests on known functions for both accuracy and performance, showing the algorithm outperforms AS in accuracy at greater computational expense. Further, we demonstrate the efficacy of AM in parameter studies using the data and MHD generator model considered by Glaws et al. (2017) to demonstrate the AS method. Our results permit deeper understanding of the effect of each parameter on the function, specifically showing which parameters are most/least influential along an active manifold, and where this sensitivity changes.

Acknowledgements

References

Birkhoff, G. and Rota, G. C. *Ordinary differential equations*. Blaisdell Pub. Co., 1969. URL https://books.google.com/books?id=R5qmAAAAIAAJ.

Burges, C. J. *Dimension reduction: A guided tour*. Now Publishers Inc, 2010.

Constantine, P. *Active subspaces: Emerging ideas for dimension reduction in parameter studies*, volume 2. SIAM, 2015.

Constantine, P. Active subspaces. Github Repository, https://github.com/paulcon/active_subspaces, October 2016a.

Constantine, P. Active subspaces data sets. Github Repository https://github.com/paulcon/as-data-sets, December 2016b.

Constantine, P. and Diaz, P. Global sensitivity metrics from active subspaces. *J. Rel. Eng.& Sys. Safe.*, 162:1–13, 2017.

Constantine, P. et al. Active subspace methods in theory and practice: Applications to kriging surfaces. *SIAM J. Sci. Comput.*, 36(4):A1500–A1524, 2018/08/09 2014. doi: 10.1137/130916138. URL https://doi.org/10.1137/130916138.

Constantine, P. et al. Exploiting active subspaces to quantify uncertainty in the numerical simulation of the hyshot ii scramjet. *J. Comp. Phys.*, 302:1–20, 2015.

Constantine, P. et al. Accelerating markov chain monte carlo with active subspaces. *Journal on Scientific Computing*, 38 (5):A2779–A2805, 2016.

Coudret, R., Girard, S., and Saracco, J. A new sliced inverse regression method for multivariate response. *Computational Statistics & Data Analysis*, 77:285 – 299, 2014. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2014.03.006. URL http://www.sciencedirect.com/science/article/pii/S0167947314000838.

De la Porte, J., Herbst, B., Hereman, W., and van Der Walt, S. An introduction to diffusion maps. In *Proc. PRASA*, 2008.

Duan, N. and Li, K.-C. Slicing regression: a link-free regression method. *The Annals of Statistics*, pp. 505–530, 1991.

Glaws, A. et al. Dimension reduction in magnetohydrodynamics power generation models: Dimensional analysis and active subspaces. *SADM: ASA Data Sci. J.*, 10(5):312–325, 2017.

Kumar, S., Mohri, M., and Talwalkar, A. Sampling methods for the nyström method. *Journal of Machine Learning Research*, 13(Apr):981–1006, 2012.

Li, K.-C. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991. ISSN 01621459. URL http://www.jstor.org/stable/2290563.

Li, L. and Nachtsheim, C. J. Sparse sliced inverse regression. *Technometrics*, 48(4):503–510, 2006. ISSN 00401706. URL http://www.jstor.org/stable/25471242.

Liu, Y., Gao, Q., Gao, X., and Shao, L. $L_{2,1}$-norm discriminant manifold learning. *IEEE Access*, 6:40723–40734, 2018.

Lukaczyk, T. W. et al. Active subspaces for shape optimization. In *AIAA Multidisciplinary Design Optimization Conference*, pp. 1171, 2014.

Pearson, K. On lines and planes of closest fit to systems of points in space. *London, Edinburgh, and Dublin Phil. Mag. J. Sci.*, 2, 1901.

Russi, T. M. *Uncertainty quantification with experimental data and complex system models*. PhD thesis, UC Berkeley, 2010.

Zhang, G. and Hinkle, J. Resnet-based isosurface learning for dimensionality reduction in high-dimensional function approximation with limited data. *arXiv preprint arXiv:1902.10652*, 2019.

Zhang, Z., Chow, T. W. S., and Zhao, M. M-isomap: Orthogonal constrained marginal isomap for nonlinear dimensionality reduction. *IEEE Transactions on Cybernetics*, 43(1):180–191, Feb 2013. ISSN 2168-2267. doi: 10.1109/TSMCB.2012.2202901.