Pursuing Active Manifolds via Nonlinear Optimization

Zachary R. del Rosario*
Stanford University, Stanford CA, 94305, USA

Dimension Reduction is an approach to alleviating the Curse of Dimensionality. Active Subspaces is a dimension reduction technique which seeks linear subspaces which are 'active'; that is, they capture the majority of variation in some quantity of interest (QoI). While useful in practice, Active Subspaces do not always exist. A natural generalization is to seek *Active Manifolds*, curved spaces which capture variations in our QoI. The problem of finding Active Manifolds is posed here as a nonlinear optimization problem, and solved using interior point methods.

Nomenclature

$\mathcal{D}$	Domain of f
$\boldsymbol{W}(\boldsymbol{x})$	Manifold matrix
$\nabla f(\boldsymbol{x})$	Gradient of QoI
ϕ_i	Manifold basis function
$\boldsymbol{\alpha}_j$	Manifold parameter vector
$\boldsymbol{w}_j(\boldsymbol{x})$	Manifold vector
$\boldsymbol{x}$	Input vector
d	Number of manifolds sought
$f(\boldsymbol{x})$	Scalar quantity of interest
k	Number of basis functions
n	Number of sample points in $\mathcal{D}$

I. Introduction

One of the most challenging difficulties facing high-fidelity modeling is the treatment of high-dimensional parameter spaces: the Curse of Dimensionality. Consider a parameter study on some quantity of interest (QoI) f in a space $\mathcal{D} \subseteq \mathbb{R}^m$; a simple heuristic is to use 10 points per dimension, in order to well represent the parameter space. Then the total number of sample points is 10^m. If a computer code implementing our model executes in a fixed time of 1 second, then our parameter study execution time scales exponentially. Figure 1 depicts the aforementioned scenario.

The only reasonable strategy to mitigate this challenge is to perform *dimension reduction*; that is, to reduce m. One scheme for dimension reduction of this sort is to seek *Active Subspaces* – linear subspaces in parameter space along which the majority of variation in our QoI is captured.[1] The Active Subspace approach gives a 'perfect' dimension reduction in the case that our QoI is a Ridge Function; that is, for $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{A} \in \mathbb{R}^{m\times k}$ with $k < m$, we have $f(\boldsymbol{x}) = g(\boldsymbol{A}^T\boldsymbol{x})$. Note that a Ridge Function is constant along directions which are orthogonal to $\boldsymbol{A}$, that is

$$\boldsymbol{W}^T\nabla f = 0 \Leftrightarrow \boldsymbol{W}^T A = 0, \tag{1}$$

where ∇f is the gradient of f, understood to be a column vector. One example of a Ridge Function is $f(\boldsymbol{x}) = \frac{1}{2}(.7x_1 + .3x_2)^2$. In this case, the Active Subspace approach discovers the Active and Inactive directions, depicted in Figure 2. Note that this gives us a 'perfect' dimension reduction, as we can completely neglect changes along the direction $[-.3, .7]^T$. Such Ridge Functions may seem like a contrivance, but they

*PhD Candidate, Aeronautics and Astronautics, 496 Lomita Mall, Stanford CA, AIAA Student Member.

are actually quite common in multivariate Fourier Transforms[2] and physical laws in general.[3] Nevertheless, Ridge Functions are not the only functional form that arises in practice; in this case the Active Subspace is approximate. While approximate Active Subspaces are useful in applications such as aerodynamic shape optimization,[4] scramjet analysis,[5] and hydrologic modeling,[6] it is easy to construct functions which do not admit this sort of low-dimensional structure, even in an approximate sense. In these cases, we would like to do more than simply give up.

A natural generalization of Active Subspaces is to seek *Active Manifolds*; that is, curved subsets of parameter space which capture the variability of a function. Such low-dimensional structures should recover linear subspaces in the case that they exist (i.e. a Ridge Function), and more general spaces when they do not. Note that if a function is differentiable, we can always move along the gradient to capture the full variability of a function – unfortunately this requires perfect knowledge of the function, which brings us back to the Curse of Dimensionality. In practice we must use a limited number of gradient samples (assumed to be available, say through an adjoint solution[7] or automatic differentiation[8]) to numerically approximate such a manifold.

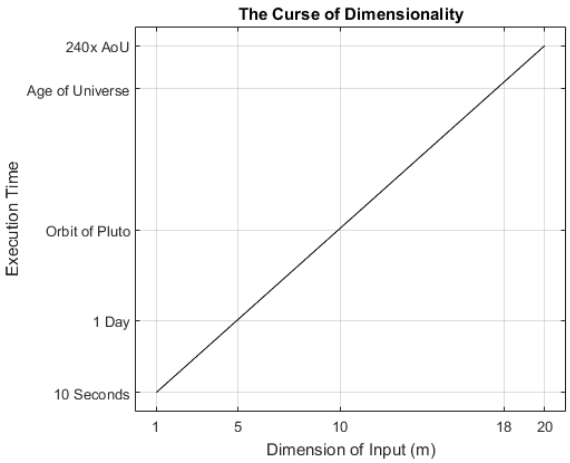

Figure 1. Execution time scales exponentially with the dimension of parameter space.

In this work, we focus on the *identification* of Active Manifolds, and leave their usage to subsequent works. The re-parameterization of a function on an Active Subspace is already laden with important considerations, which is further complicated by generalization to more arbitrary spaces. These are important issues which lie outside the scope of this document.

II. Seeking Active Manifolds

The strategy we will adopt in this work is to generalize the properties of a Ridge Function: By allowing Equation 1 to vary in space, we arrive at

$$\boldsymbol{W}(\boldsymbol{x})^T \nabla f(\boldsymbol{x}) = 0. \tag{2}$$

Equation 2 gives us a set of directions $\boldsymbol{W}(\boldsymbol{x})$ along which $f(\boldsymbol{x})$ does not vary – these directions define *Inactive Manifolds*, while the orthogonal directions define *Active Manifolds*. Note that as long as $f(\boldsymbol{x})$ is differentiable, we know $\boldsymbol{W}(\boldsymbol{x})$ exists, as we can simply perform a QR decomposition on the gradient ∇f augmented with the $m \times m$ identity matrix. While this is mathematically possible, such a scheme is computationally intractable, as it is an infinite dimensional problem. Instead, we must settle on a finite dimensional problem by approximating Equation 2.

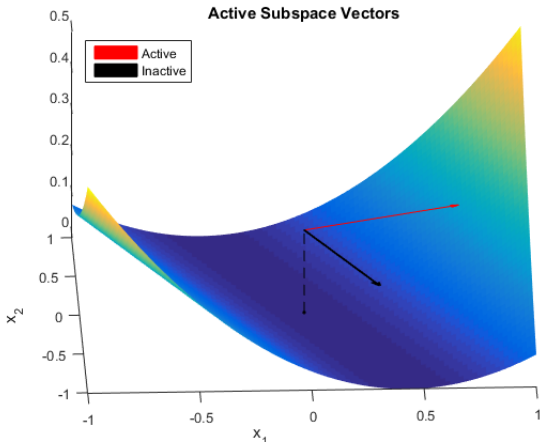

Figure 2. Note that on average, the function changes more along the active directions than the inactive directions; in this case, the change in the inactive direction is exactly zero.

First, we will consider a single direction $\boldsymbol{w}_j(\boldsymbol{x})$, and parameterize on a finite number of basis functions. Choose a set of k differentiable functions $\phi_i : \mathbb{R}^m \to \mathbb{R}$, and set

$$\boldsymbol{w}_j(\boldsymbol{x}) = \sum_{i=1}^{k} \alpha_{ij} \nabla \phi_i(\boldsymbol{x}), \tag{3}$$

where the α_{ij} parameterize $\boldsymbol{w}_j(\boldsymbol{x})$ in a linear fashion. Now we construct $\boldsymbol{W}(\boldsymbol{x})$ from a collection of these $\boldsymbol{w}_j(\boldsymbol{x})$, each parameterized on a different $\boldsymbol{\alpha}_j$ vector, that is

$$\boldsymbol{W}(\boldsymbol{x}) = [\boldsymbol{w}_1(\boldsymbol{x}), \ldots, \boldsymbol{w}_l(\boldsymbol{x})]. \tag{4}$$

Rather than attempt to solve for $\boldsymbol{W}$ all at once, we will attempt to enforce Equation 2 for each $\boldsymbol{w}_j(\boldsymbol{x})$ individually. We must make two concessions though. First, even though the $\boldsymbol{w}_j$ are now parameterized on a finite set, Equation 2 is enforced at all points $\boldsymbol{x} \in \mathcal{D}$: In practice we must sample a number of points $\{\boldsymbol{x}_i\}_{i=1}^n \subseteq \mathcal{D}$ and enforce $\boldsymbol{w}_j^T(\boldsymbol{x}_i)\nabla f(\boldsymbol{x}_i) = 0$ on this set. Second, unless $\boldsymbol{W}$ lies precisely within the span of the $\nabla\phi_i$, Equation 2 will hold only approximately: In practice we will attempt to minimize the residual, defined by

$$R = \|\boldsymbol{M}\boldsymbol{\alpha}\|_2, \tag{5}$$

where

$$\boldsymbol{M} = \begin{bmatrix} \nabla\phi_1^T(\boldsymbol{x}_1)\nabla f(\boldsymbol{x}_1) & \cdots & \nabla\phi_k^T(\boldsymbol{x}_1)\nabla f(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \nabla\phi_1^T(\boldsymbol{x}_n)\nabla f(\boldsymbol{x}_n) & \cdots & \nabla\phi_k^T(\boldsymbol{x}_n)\nabla f(\boldsymbol{x}_n) \end{bmatrix}. \tag{6}$$

Note that if $\boldsymbol{M}\boldsymbol{\alpha} = 0$, then Equation 2 holds exactly for $\boldsymbol{W} = [\boldsymbol{w}]$, parameterized on $\boldsymbol{\alpha}$. Also note that this fixed $\boldsymbol{M}$ defines the residual for each $\boldsymbol{w}_j$, as they are each parameterized on the same basis, albeit with a different $\boldsymbol{\alpha}$. Naïvely, we may seek to minimize the residual R directly; however, this optimization problem has a trivial solution $\boldsymbol{\alpha} = 0$. In practice we constrain the length via $\|\boldsymbol{\alpha}\|_2 \geq 1$. Additionally, note that we wish to solve for a *collection* of $\boldsymbol{\alpha}_j$; we can accomplish this by demanding that each new $\boldsymbol{\alpha}_j$ be orthogonal to the previous ones. Finally, we add a 1-norm term to encourage sparsity; we would like each $\boldsymbol{\alpha}_j$ to be 'simple', in the sense of being sparse. This leads to the following sequence of d optimization problems

$$\begin{aligned} \min\ & \|\boldsymbol{M}\boldsymbol{\alpha}_j\|_2 + \beta\|\boldsymbol{\alpha}_j\|_1, \\ \text{s.t.}\ & \|\boldsymbol{\alpha}_j\|_2 \geq 1, \\ & \boldsymbol{A}_j^T\boldsymbol{\alpha}_j = 0, \end{aligned} \tag{7}$$

for $j = 1, \ldots, d$, where $\beta > 0$ is a tunable weighting parameter, and $\boldsymbol{A}_j = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{j-1}]$, with $\boldsymbol{A}_1 = 0$. There are a few important points to note: First, one should set d equal to the number of Inactive Manifolds sought – one could either define a fixed number of manifolds to seek, or continue running until the residual cannot be reduced under some desired tolerance. The former method is appropriate if the degree of dimension reduction required is already known, while the latter will discover the maximum reduction available, subject to the choice of basis. Second, Problem 7 has a convex objective function but a nonlinear constraint, thus we do *not* enjoy the benefits of a convex problem. This creates issues when attempting to design a solver, which will be addressed in Section III:A.

A. Choice of Basis

The choice of basis functions ϕ_i is crucial for obtaining an Active Manifold. The basis must be chosen to include $\boldsymbol{W}(\boldsymbol{x})$, at least in some approximate sense. In the case of a Ridge Function the Active Manifolds are linear subspaces, so a linear choice of basis $\phi_i(\boldsymbol{x}) = x_i$ is appropriate. In fact, choosing a linear basis and approximating Equation 2 by finding the nullspace via a Singular Value Decomposition (SVD) exactly recovers the Active Subspace procedure! To see this, substitute the linear basis into Equation 6 to find

$$\boldsymbol{M}_l = [\nabla f(\boldsymbol{x}_1), \ldots, \nabla f(\boldsymbol{x}_n)]^T. \tag{8}$$

Equation 2 corresponds to the condition $\boldsymbol{M}_l\boldsymbol{\alpha} = 0$, which is a nullspace computation. This can be accomplished by considering the SVD of $\boldsymbol{M}_l = U\Sigma V^T$; the vectors $\boldsymbol{v}_i$ which correspond to the zero singular values σ_i form a basis for the nullspace of $\boldsymbol{M}_l$. These are found via an eigenvalue decomposition of $\boldsymbol{M}_l^T\boldsymbol{M}_l$, which equals

$$\boldsymbol{M}_l^T\boldsymbol{M}_l = \sum_{i=1}^{n} \nabla f(x_i)\nabla f^T(x_i). \tag{9}$$

Note that the Active Subspace is also found via an eigenvalue decomposition of the $\boldsymbol{C}$ matrix, defined as the weighted average of the outer product of the gradient.[1] Compare Equation 9 to the Monte Carlo approximation of the $\boldsymbol{C}$ matrix

$$\hat{\boldsymbol{C}} = \frac{1}{n}\sum_{i=1}^{n} \nabla f(x_i)\nabla f^T(x_i). \tag{10}$$

Note that up to a factor of n, they are the same! Thus, for the correct choice of basis, the Active Manifold procedure defined above should recover Active Subspaces. This is important, as it shows that if our basis is chosen correctly and Problem 7 approximates Equation 2 well, we can do no worse than Active Subspaces.

In this work, we try a number of different sets of basis functions, defined in Table 1 below. These include both the linear basis and larger sets.

Name	Basis
Linear	x_i
Quadratic	x_i, x_i^2, $\log(\lvert x_i\rvert)$

Table 1. Basis sets used in numerical experiments. Note that each basis function is used in each dimension; thus for $m = 3$, the Linear set has 3 basis functions, while the Quadratic set has 9.

III. Solver

In order to solve Problem 7, a nonlinear solver was designed, implemented, and tested on a number of different quantities of interest f and sets of basis functions ϕ_i.

A. Solver Design

As mentioned in Section II, the optimization problem in question is a nonlinear optimization problem with nonlinear constraints. The solver is an implementation of the interior point method using log-barrier functions and a quasi-Newton method using the BFGS update rule.[9] In this scheme, we first solve the feasibility problem by constructing an objective function on the constraints, then run the interior point method successively relaxing the barriers.

Note that one could construct a log-barrier on the linear constraint $\boldsymbol{A}_j^T\boldsymbol{\alpha}_j = 0$. Instead, we employ *progressive reparameterization*; that is, reparameterizing the problem such that the constraint is automatically satisfied. We accomplish this for step j by constructing a set of vectors $[\tilde{\boldsymbol{q}}_{1j}, \ldots, \tilde{\boldsymbol{q}}_{k-j+1}] = \tilde{\boldsymbol{Q}}_j$ orthogonal to all previous solutions $[\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{j-1}] = \boldsymbol{A}_j$, and defining a new set of $k-j+1$ variables $\tilde{\boldsymbol{\alpha}}_j$ which replace the old $\boldsymbol{\alpha}_j = \tilde{\boldsymbol{Q}}_j\tilde{\boldsymbol{\alpha}}_j$. Note that by this construction $\boldsymbol{\alpha}_j$ is orthogonal to all previous solutions.

Note that the first iteration $j = 1$ of Problem 7 has no linear constraint, so the solver described above is sufficient. For subsequent iterations, we redefine the variables as described above and optimize over $\tilde{\boldsymbol{\alpha}}_j$. Our new optimization problem is

$$\begin{aligned} &\min \ \|\boldsymbol{M}\tilde{\boldsymbol{Q}}_j\tilde{\boldsymbol{\alpha}}_j\|_2 + \beta\|\tilde{\boldsymbol{Q}}_j\tilde{\boldsymbol{\alpha}}_j\|_1, \\ &\text{s.t.} \ \|\tilde{\boldsymbol{Q}}_j\tilde{\boldsymbol{\alpha}}_j\|_2 \geq 1, \end{aligned} \tag{11}$$

where $\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^{k-j+1}$, and $\tilde{\boldsymbol{Q}}_j$ is found by taking the last $k-j+1$ columns of the orthogonal matrix found from an economy QR decomposition of $\boldsymbol{A}_j$, augmeted with the $k \times k$ identity matrix

$$[\boldsymbol{A}_j, \boldsymbol{I}] = \boldsymbol{QR}. \tag{12}$$

Through this scheme, subsequent vectors are guaranteed to be orthogonal to numerical precision. However, note that since the QR decomposition has a complexity of $O(n^3)$, this method is inappropriate for excessively high-dimensional spaces. In these cases, it may be prudent to approximate the linear constraint with a barrier function. However, it is worth noting that since the computational cost we are trying to alleviate is exponential, a cubic cost may still be comparatively cheap.

Finally, we must address the lack of convexity in our optimization problem. Problem 7 is non-convex, and thus may contain local minima. This is problematic with our interior point method, which for a poor choice of initial guess may converge to such a local optimum. To alleviate this issue, we employ a restart heuristic: We begin with a uniform random guess within a unit box in the positive orthant, and if at any stage j the residual (Eq. 5) does not drop below an absolute threshold, we restart that stage with a new initial guess. This is an admittedly simple solution, which is not guaranteed to solve the problem, but it works well enough in practice.

B. Solver Performance

The solver described above was evaluated on a number of different test cases in order to determine its efficacy. These test cases are summarized in Table 2. Note that a Ridge Function in this case is a function of the form

$$f(\boldsymbol{x}) = (\boldsymbol{a}^T \boldsymbol{x})^2, \tag{13}$$

where $\boldsymbol{a}$ is a random vector of unit length. A 'Mixed Function' is a function of the form

$$f(\boldsymbol{x}) = (a_1 x_1 + \cdots + a_{\lfloor m/2 \rfloor} x_{\lfloor m/2 \rfloor} + a_{\lfloor m/2 \rfloor + 1} x^2_{\lfloor m/2 \rfloor + 1} + \cdots + a_m x^2_m), \tag{14}$$

where as before $\boldsymbol{a}$ is a random vector. Thus Equation 14 is 'mixed' in the sense that it has multiple types of terms. Note that the Linear basis can recover the Inactive Subspace of a Ridge Function exactly, while the Quadratic basis can exactly recover the Inactive Manifolds of a Mixed Function. This was intentional; we leave the study of approximate Active Manifolds to a future work.

In addition to a family of objective function, for each test case we fix the dimension of the objective function m, and the number of Inactive Manifolds sought d. In each case d was chosen to maximize the dimension reduction. A residual threshold of $R_t = 10^{-4}$ was set for the reset criteria, and 8 resets per stage were allowed. The $\boldsymbol{x}_i$ are sampled uniform randomly from the unit hypercube on $\mathbb{R}^m$, and in each case a fixed number of 200 samples are taken. After the number of resets is exceeded, the best run of the stage is taken, and the stage advanced.

Test Case	Quantity of Interest	Dimension	d
1	Ridge Function	$m = 3$	2
2	Ridge Function	$m = 10$	9
3	Mixed Function	$m = 3$	2
4	Mixed Function	$m = 10$	9

Table 2. Test cases used to evaluate solver.

As mentioned in Section II:A, the choice of a linear basis recovers Active Subspaces, thus our first test was to determine if the solver could recover this behavior. Figures 9 and 10 show the results of studying Case 1: a three-dimensional ($m = 3$) Ridge Function with a one-dimensional Active Subspace.

Figures 9 and 10 are typical results when studying such a Ridge Function; we see the rapid convergence rate of quasi-Newton's method in both convergence metrics. Since the residual measures the degree to which Equation 2 holds and R is reaching machine zero for this case, this provides strong evidence that the solver is finding the correct Active Subspace. We can further verify this result by checking the subspace distance between the subspace found via our solver, and the Inactive Subspace found via the usual procedure (definition provided in the Appendix). In the case depicted above, the subspace distance is $\text{Dist}(\boldsymbol{W}_{\text{solver}}, \boldsymbol{W}_{\text{AS}}) = 1.0786 \times 10^{-9}$; clearly, the two subspaces are the same to working precision.

Since our solver uses a random initial guess, the solver performance is a random variable. Some runs of the solver give abberant behavior, depicted below. Note that the convergence sequences depicted in Figures 5 and 6 do not qualitatively match; this demonstrates the importance of checking both the residual and objective values. Since the objective is a mixture of both the residual and log-barriers, it is not representative of the true quantity we are attempting to minimize.

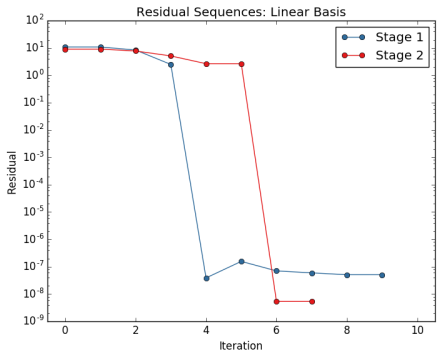

Figure 3. Typical residual sequence for Test Case 1.

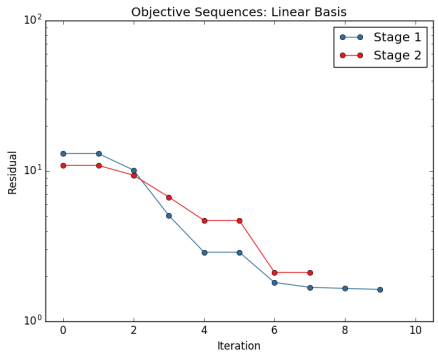

Figure 4. Typical objective sequence for Test Case 1.

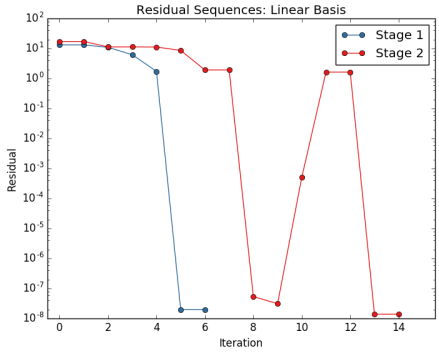

Figure 5. Abberant residual sequence for Test Case 1.

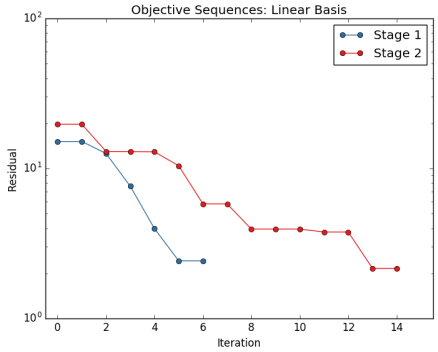

Figure 6. Abberant objective sequence for Test Case 1.

One final remark is that – in general – the number of iterations required by the solver increases with the dimensionality of both the problem and the basis. This dependence is shown in Figures 7 and 8. These figures are produced by running test cases identical to those noted in Table 2, except for varying m. Note that the computational growth rate is slowing with increasing dimension, which is an attractive feature of the process. Note also that for each of these test cases, a *fixed* number of samples $\boldsymbol{x}_i$ are evaluated on f; thus the figures imply that we can get away with a relatively unchanged number of expensive samples, so long as we work harder on the optimization problem. This is precisely what we would like to see from a dimension reduction technique – if we can offload evaluations of f to some other process, then we can significantly accelerate our computational studies. The precise dependence on the number of samples $\boldsymbol{x}_i$ is critical, and a definite candidate for future work.

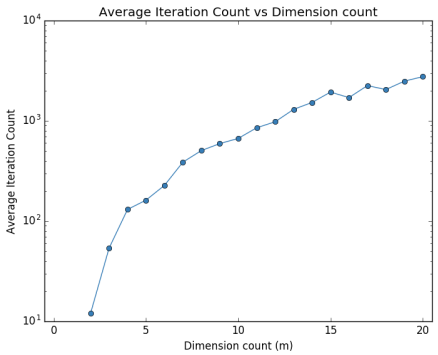

Figure 7. Average iteration count over 10 runs for a Ridge Function objective with a linear basis.

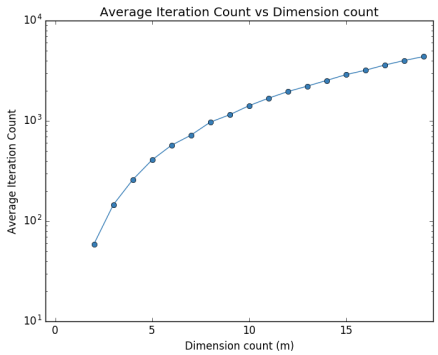

Figure 8. Average iteration count over 10 runs for a Mixed Function objective with a quadratic basis.

The remaining convergence plots are qualitatively similar, and are thus relegated to the Appendix.

IV. Conclusion and Future Work

Concluding remarks, yo.

References

[1]Constantine, P., *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*, SIAM Philadelphia, 2015.

[2]Pinkus, A., *Ridge Functions*, Cambridge University Press, Cambridge, 2015.

[3]Constantine, P. G., del Rosario, Z., and Iaccarino, G., "Many physical laws are ridge functions," *ArXiv e-prints*, May 2016.

[4]T. Lukaczyk, F. Palacios, J. A. and Constantine, P., "Active subspaces for shape optimization," *10th AIAA Multidisciplinary Design Optimization Conference*, 2014.

[5]Constantine, P., Emory, M., Larsson, J., and Iaccarino, G., "Exploiting active subspaces to quantify uncertainty in the numerical simulation of the HyShot II scramjet," *Journal of Computational Physics*, Vol. 302, 2015, pp. 1 – 20.

[6]Jefferson, J. L., Gilbert, J. M., Constantine, P. G., and Maxwell, R. M., "Active subspaces for sensitivity analysis and dimension reduction of an integrated hydrologic model," *Computers & Geosciences*, Vol. 83, 2015, pp. 127 – 138.

[7]Jameson, A., "Aerodynamic design via control theory," *Journal of Scientific Computing*, Vol. 3, 1988.

[8]Rall, L., *Automatic Differentiation: Techniques and Applications*, Vol. 120, Springer, 1981.

[9]Belegundu, A. and Chandrupatla, T., *Optimization concepts and applications in engineering*, Prentice Hall, 1999.

V. Appendix

A. Subspace Distance

The distance between subspaces measures the distance between the ranges of two different linear subspaces, and is defined[1]

$$\text{Dist}(\boldsymbol{W}_1, \boldsymbol{W}_2) = \|\boldsymbol{W}_1\boldsymbol{W}_1^T - \boldsymbol{W}_2\boldsymbol{W}_2^T\|_2. \tag{15}$$

B. Convergence Plots

Here is the full set of convergence plots for every Case study.

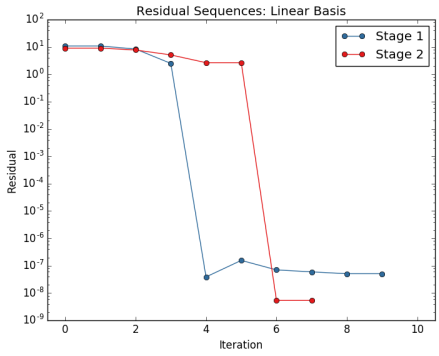

Figure 9. Residual sequence for Case 1.

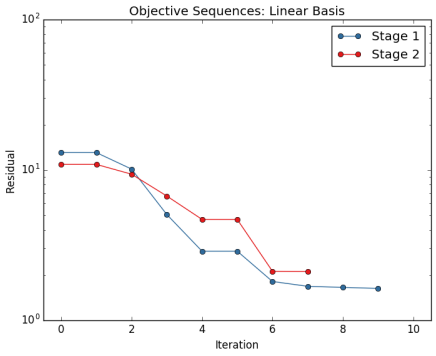

Figure 10. Objective sequence for Case 1.

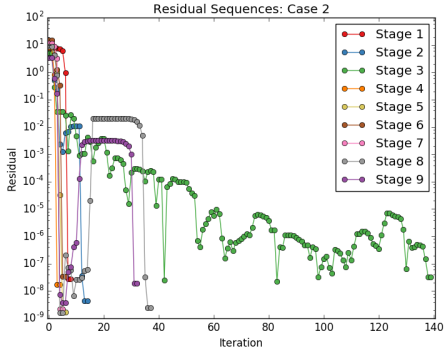

Figure 11. Residual sequence for Case 2.

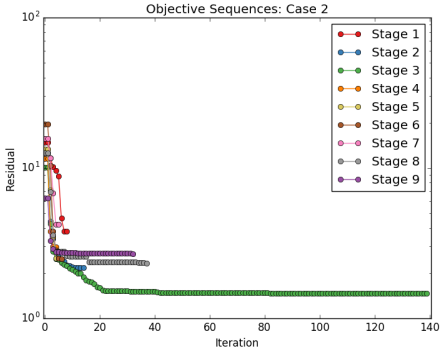

Figure 12. Objective sequence for Case 2.

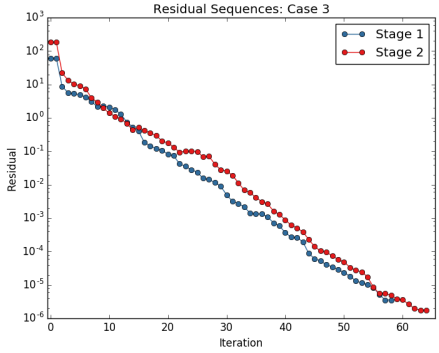

Figure 13. Residual sequence for Case 3.

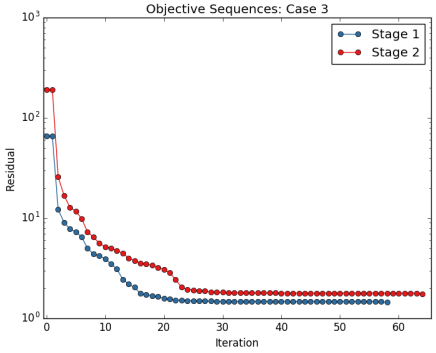

Figure 14. Objective sequence for Case 3.

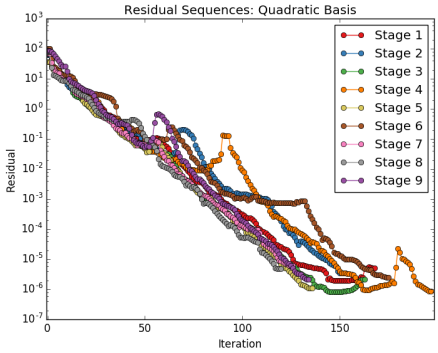

Figure 15. Residual sequence for Case 4.

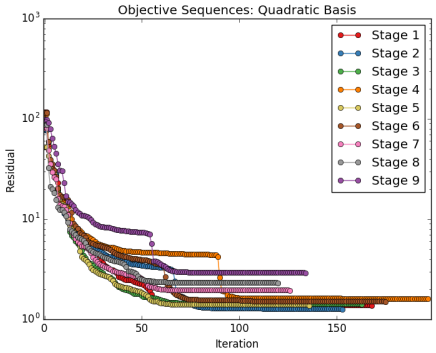

Figure 16. Objective sequence for Case 4.