Low-Rank Dynamic Mode Decomposition using Riemannian Manifold Optimization

Palash Sashittal and Daniel J. Bodony

Abstract— **We present a method for non-intrusive data-driven reduced order modeling of high-dimensional dynamical systems using a new low-rank extension of Dynamic Mode Decomposition (DMD). A matrix optimization problem with a rank-constraint on the solution is formulated and results in a non-convex optimization problem. We propose two methods to solve the optimization problem. The first is an iterative subspace projection method that is computationally efficient but can only give the optimal solution under certain conditions. In the second method we perform Riemannian optimization on Grassmanian manifolds. Using a model equation for fluid flows, we evaluate the performance of the proposed methods on complex linearized Ginzburg-Landau equations in the supercritical globally unstable regime.**

I. INTRODUCTION

Fluid flows are governed by partial differential equations that, when discretized, result in very high-dimensional systems for numerical simulation that challenge the largest computers. There is therefore a need to obtain simple reduced order models that approximate the dynamics of the system. This model reduction is often done using data available from high-fidelity numerical simulations or data measured from experiments. One common approach is to approximate the trajectories of the dynamical system on a lower dimensional subspace, such as Petrov-Galerkin projection [1]. However, these methods require the knowledge of the equations governing the high-dimensional system that may not be completely known for complex systems.

To overcome the need for known equations, non-intrusive data-driven methods are used to infer reduced order models from data with limited or no knowledge about the underlying dynamical system. This includes for example, methods such as Linear Inverse Modeling (LIM) [2], Autoregressive Modeling [3] and Dynamic Mode Decomposition (DMD) [4]. Lately, DMD and its variants have had a measurable impact in modal decomposition and reduced order control of fluid flows.

In this study we focus on a new rank-constrained variant of the DMD. Suppose that the system is governed by the following linear dynamics,

$$x_{k+1} = Ax_k, \quad x_0 = \theta,$$

where $k \in \mathbb{Z}$ denotes the iteration number, $x \in \mathbb{R}^m$ denotes the state variable, $A \in \mathbb{R}^{m\times m}$ is the state transition matrix and $\theta \in \mathbb{R}^m$ is the initial condition. The low-rank DMD problem is formulated as

$$\min_{\hat{A}:rank(\hat{A})\leq r} \left\| Y - \hat{A}X \right\|^2 \tag{1}$$

where $\|\cdot\|$ is the Frobenius norm, $\hat{A} \in \mathbb{R}^{m\times m}$ is the inferred state transition matrix and $X, Y \in \mathbb{R}^{m\times n}$ are data matrices formed by n pairs of data snapshots with Y showing the action of A on X,

$$X := (x_1|\cdots|x_n), \quad Y := (x'_1|\cdots|x'_n),$$

where $x'_i \approx Ax_i$ for $i = 1, \cdots, n$. This rank-constrained optimization problem has been studied in [5] and [6]. The approximate solution proposed by the authors in [6] is the r^{th} order approximation of the unconstrained optimal solution. The approximation is performed either by truncated singular value decomposition or by a sparsity promoting algorithm. The authors in [7] showed that there exists a closed form solution for problem(1). However, due to the huge size of the solution when m is large, it is computationally expensive to extract modes from the solution required for model reduction. As an alternative, we suggest an iterative gradient-based algorithm. This work is very closely related to optimal mode decomposition [8] in which the low-rank solution is restricted to have identical left and right images and to use a first-order optimization method to find the solution. In this work we solve the more general problem and use a second-order optimization method.

II. MATHEMATICAL FORMULATION

We consider the following optimization problem equivalent to (1),

$$\min_{L,D,R} \left\| Y - LDR^TX \right\|^2 \tag{2}$$

where both $L, R \in \mathbb{R}^{m\times r}$ and $L^TL = R^TR = I_r$ ($r \times r$ identity matrix) and D is the r-ranked matrix approximating the dynamics of the underlying system. Observe that (2) seeks the optimal left and right images of the state transition matrix in contrast to standard DMD. A key step to solving (2) is to find the optimal solution for D as a function of L and R. For a fixed L and R there exists a closed form optimal solution for D and we can find it as follows.

Let $J(L, D, R)$ denote the objective function,

$$\begin{aligned} J(L, D, R) &= \left\| Y - LDR^TX \right\|^2 \\ &= tr(Y^TY - 2Y^TLDR^TX + X^TRD^TDR^TX) \end{aligned}$$

This work was sponsored by the Office of Naval Research (ONR) as part of the Multidisciplinary University Research Initiatives (MURI) Program, under grant number N00014-16-1-2617. The views and conclusions contained herein are those of the authors only and should not be interpreted as representing those of ONR, the U.S. Navy or the U.S. Government.

P. Sashittal and D. Bodony are with Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA sashitt2@illinois.edu, bodony@illinois.edu

where $tr(Y)$ = trace of Y. Taking derivative of J with respect to D we get

$$\frac{\partial J}{\partial D} = -2L^TYX^TR + 2DR^TXX^TR.$$

Setting this derivative to zero gives the optimal D as function of L and R as

$$D(L,R) = (L^TYX^TR)(R^TXX^TR)^{-1}. \qquad (3)$$

This eliminates D from our problem. We solve for the optimal solution of L and R and find the corresponding D using (3). The optimization problem can thus be re-written as

$$\min_{L,R} \left\|Y - LL^TY(X^TR(R^TXX^TR)^{-1}R^TX)\right\|^2. \qquad (4)$$

The second term in the parenthesis is an orthogonal projection in the subspace spanned by X^TR and we define

$$Q_R = X^TR(R^TXX^TR)^{-1}R^TX.$$

Notice that $Q_R = Q_R^T$ and $Q_RQ_R = Q_R$. Using this, the objective function of the optimization problem (4) is

$$\begin{aligned} J(L,R) &= \left\|Y - LL^TYQ_R\right\| \\ &= tr(Y^TY - Q_R^TY^TLL^TYQ_R) \\ &= \|Y\|^2 - \left\|L^TYQ_R\right\|^2. \end{aligned}$$

Since $\|Y\|^2$ is a constant, the optimization problem can be reduced to

$$\min_{L,R} (-\left\|L^TYQ_R\right\|^2). \qquad (5)$$

We will denote this objective function with $G(L,R)$. Note that the objective function is invariant under orthogonal transformation of L and R. For any given $P \in \mathbb{R}^{r\times r}$ such that $P^TP = I_r$, $G(L,R) = G(LP,R)$ and also $G(L,RP) = G(L,R)$. This implies that only the space spanned by the columns of L and R are of interest which are represented by equivalent classes $[L]$ and $[R]$. Hence the optimization problem can be equivalently performed over the set of r-dimensional subspaces in $\mathbb{R}^m$ which form a Grassmanian manifold $\mathcal{G}_{r,m}$. In matrix representation an element of $\mathcal{G}_{r,m}$ is specified by an orthogonal basis of the r-dimensional subspace. The manifold we perform optimization over is a product manifold of two Grassmanian manifolds

$$\mathcal{M} := \mathcal{G}_{r,m} \times \mathcal{G}_{r,m}$$

An element of $\mathcal{M}$, in matrix representation, will be characterized by $M = (L,R)$.
In the following sections, we describe two iterative methods to solve the optimization problem (5) on this manifold

1) Subspace projection
2) Gradient descent on Grassmannian manifolds

III. Subspace Projection

Since Q_R is an orthogonal projection in the column space of X^TR, following *Lemma* 1 given in [8], if C_R is an orthogonal matrix such that $\text{Im}(C_R) = \text{Im}(X^TR)$, then $C_RC_R^T = Q_R$. Using this we can transform the optimization problem to

$$\max_{L,R} \left\|L^TYC_R\right\|^2$$

with an additional constraint given by $\text{Im}(C_R) = \text{Im}(X^TR)$. Here, $L^TL = C_R^TC_R = I_r$. We make the following observations,

1) If C_R is fixed, the optimal L is given by an orthogonal basis of the subspace spanned by the r left singular vectors of AC_R with the highest singular values
2) If L is fixed, the optimal C_R is given by an orthogonal basis subspace spanned by the r left singular vectors of A^TL with the highest singular values

To show the first observation, consider the case when C_R is fixed. Let $AC_R = S$. By using the definition of Frobenius norm we get,

$$\left\|L^TS\right\|^2 = tr(L^TSS^TL).$$

The maximum trace is obtained only if L is an orthogonal basis of the space spanned by the eigenvectors of SS^T corresponding to the largest eigenvalues. This is the same as the left singular vectors corresponding to the largest singular values of S. Similar argument goes for the case when L is fixed.

Observe that since C_R is given by the left singular vectors of Y^TL by the optimality condition, we have $\text{Im}(C_R) = \text{Im}(Y^TL)$. But the constraint is that $\text{Im}(C_R) = \text{Im}(X^TR)$. If R is found such that $X^TR \approx Y^TL$, both the optimality condition and the constraint are satisfied. Therefore, we find R by solving the following optimization problem

$$\min_{R} \left\|Y^TL - X^TR\right\|^2.$$

This is known as the 'Orthogonal Procrustes problem'. Solution to this problem is given by $R = UV^T$ where U, V are the left and the right singular vectors of XY^TL. Therefore,

$$\begin{aligned} [U,V] &= \text{SVD}(XY^TL) \\ R &= UV^T. \end{aligned}$$

Remark 1: Even though R is solution of an optimization problem, C_R may not satisfy its optimality condition exactly. However, the constraint for the optimization is always exactly satisfied.
Remark 2: If the optimal L does not differ excessively from the left singular vectors of A and B, then the optimal solution is given by $R = L$ and we get back the optimal mode decomposition solution.

We can now construct an iterative algorithm which when converged will give the optimal solution to the optimization problem. In algorithm 1, $\Pi(A)$ is a function that returns an orthogonal basis of $\text{Im}(A)$

Algorithm 1 Subspace projection method

```
Require: $Y \in \mathbb{R}^{m\times n}$, $X \in \mathbb{R}^{m\times n}$
 1: Guess initial $L_0$,$R_0$ and compute $C_{R_0} \leftarrow \Pi(X^T R_0)$
 2: $k = 0$
 3: repeat
 4:     $L_{k+1} \leftarrow \Pi(Y C_{R_k})$
 5:     $R_{k+1} \leftarrow \arg\min \left\|Y^T L_k - X^T R\right\|^2$
 6:     $C_{R_{k+1}} \leftarrow \Pi(X^T R_{k+1})$
 7:     $\epsilon \leftarrow (G(L_{k+1}, R_{k+1}) - G(L_k, R_k))/G(L_k, R_k)$
 8:     $k \leftarrow k + 1$
 9: until $\epsilon \leq$ threshold
10: $D = (L_k^T Y X^T R_k)(R_k^T X X^T R_k)^{-1}(R_k^T L_k)$
11: return $L_k, D, R_k$
```

The optimality of the solution obtained from this algorithm depends on the solution of R in step 5 of the algorithm. If there exists an R such that $Y^T L = X^T R$, then if the algorithm converges, it will give the optimal solution. In practice we see that the algorithm converges very quickly and gives a solution closer to the optimal as r increases.

IV. Gradient descent method

In this section, we give an overview of Riemmanian manifold optimization and how it differs from optimization in Euclidean space. The reader is referred to [9] and [10] for a detailed review on Riemmanian optimization methods.

Consider optimization in Euclidean space, $\mathcal{E}$, where at iteration k we are at point x_k and need to update our position in some direction in the tangent space, $dx_k \in \mathcal{T}_{x_k}\mathcal{E}$. The search direction dx_k is taken along the derivative of the objective function at the point x_k. However in Riemannian space this derivative in general might not lie in the tangent space of manifold at the point x_k. Therefore, we need to consider the projection of the derivative on the tangent plane to find the search direction dx_k.

The update rule for the optimization in the two spaces is also different. In Euclidean optimization, the tangent space at some point x_k is same as the ambient space with the origin translated to that point. Therefore, the update is simply $x_{k+1} = x_k + dx_k$ with the new point at iteration $k+1$ as $x_{k+1} \in \mathcal{E}$. Now consider the same case with $x_k \in \mathcal{M}$ where $\mathcal{M}$ is some Riemannian manifold and search direction $dx_k \in \mathcal{T}_{x_k}\mathcal{M}$. The same update rule $x_k + dx_k$ might not be defined since x_k and dx_k belong to different spaces and even if it is defined, x_{k+1} need not belong to $\mathcal{M}$. Therefore, we need a way to move in direction dx_k while staying on the manifold $\mathcal{M}$. We do this using the Exponential map. It defines a curve that forms the shortest path between two points on the manifold. Therefore the exponential map takes an element from the tangent space $\mathcal{T}_{x_k}\mathcal{M}$ and maps it to an element in $\mathcal{M}$ so that

$$x_{k+1} = Exp_{x_k}(dx_k).$$

For high dimensional manifolds, this method of update can be very computationally expensive. There exists a more efficient generalization of the exponential map called a retraction. In this study we use a first-order retraction on Grassmanian manifolds. For a detailed study on retractions the reader is referred to [10]. We can now discuss the geometry of the problem and the trust-region optimization algorithm.

A. Geometry of $\mathcal{M}$ with its canonical metric

We define the Riemannian metric on the manifold $\mathcal{M}$ as

$$g_c((dL_1, dR_1), (dL_2, dR_2)) = tr(dL_1^T dL_2) + tr(dR_1^T dR_2)$$

for $(dL_1, dR_1), (dL_2, dR_2) \in \mathcal{T}_{L,R}\mathcal{M}$. This metric is similar to the Euclidean metrics because for Grassmanian manifolds the Euclidean metric and the canonical metric are the same. To describe the geometry of $\mathcal{M}$ it is sufficient to describe the geometry of Grassmanian manifolds. Consider the objective function $F(L)$ defined on the Grassmanian manifold where $L \in \mathcal{G}_{r,m}$. Let $\nabla F(L)$ and $\nabla^2 F(L)$ be the Euclidean gradient and Hessian,

$$(\nabla F(L))_{ij} = \frac{dF}{dL_{ij}}, \quad \text{and}$$
$$(\nabla^2 F(L))_{ij,kl} = \frac{d^2 F}{dL_{ij} dL_{kl}}.$$

The Riemannian gradient of the function at $[L]$ is defined as the tangent vector, $\operatorname{grad} F(L) \in \mathcal{T}_L\mathcal{G}_{r,m}$ such that,

$$tr((\nabla F)^T dL) = g_c(\operatorname{grad} F(L), dL) \ \forall \ dL \in \mathcal{T}_L\mathcal{G}_{r,m}.$$

This gives the following relation between Euclidean and Riemannian gradients,

$$\operatorname{grad} F(L) = \nabla F(L) - LL^T \nabla F(L). \tag{6}$$

Using the definition of Hessian in [9], the action of the Riemannian Hessian on any tangent vector dL can be described as follows,

$$\begin{aligned} \operatorname{Hess} F(L)[dL] = (I_m - LL^T)\nabla^2 F(L)[dL] \\ - dL(L^T \nabla F(L)). \end{aligned} \tag{7}$$

The exponential map on the Grassmanian manifold in the direction of dL is given by

$$L(t) = LV\cos(\Sigma t)V^T + U\sin(\Sigma t)V^T \tag{8}$$

where $dL = U\Sigma V^T$ is the singular value decomposition of dL and $t \in [0, 1]$. A more computationally efficient way of performing gradient update is to perform the following retraction

$$\text{Retr}_L(dL) = UV^T \tag{9}$$

where $(L + dL) = U\Sigma V^T$ is the singular value decomposition of $(L + dL)$. Another common method of retraction uses QR-factorization instead of SVD [10]. We are now in a position to define the gradients and Hessians of $G(L, R)$.

B. Euclidean gradient and Hessian

We first find the Euclidean derivative and Hessian of $G(L,R)$ with respect to the matrices L and R. The objective function G can also be written as

$$G = -tr(Q_R Y^T L L^T Y).$$

The gradient of G with respect to L is given by

$$\frac{\partial G}{\partial L} = -2YQ_RY^TL.$$

The gradient of G with respect to R is

$$\begin{aligned}\frac{\partial G}{\partial R} &= -\frac{\partial tr(Q_RY^TLL^TY)}{\partial R}\\ &= -\frac{\partial tr(L^TYQ_RY^TL)}{\partial R}\\ &= -\frac{\partial tr((L^TYX^TR)(R^TXX^TR)^{-1}(R^TXY^TL))}{\partial R}.\end{aligned}$$

Computing and simplifying the derivative results in the following expression,

$$\frac{\partial G}{\partial R} = -2X(P - Q_RP),$$

where $P = Y^TLL^TYX^TR(R^TXX^TR)^{-1}$. Therefore, the Euclidean derivative of the objective function G is

$$\nabla G(L,R) = (-2YQ_RY^TL, -2X(P - Q_RP)).$$

For the Hessian we use the identity

$$d(X^{-1}) = -X^{-1}d(X)X^{-1}.$$

The Euclidean Hessian comes out to be

$$\begin{aligned}\nabla^2 G(L,R)[dL,dR] = (&-2Y dQ_RA^TL - 2YQ_RY^TdL,\\ &-2X((I_n - Q_R)dP - dQ_RP)).\end{aligned}$$

The expressions of dQ_R and dP are given in Appendix I.

C. Riemannian gradient and Hessian

Using the Riemannian gradient definition in (6) we get

$$\begin{aligned}\operatorname{grad} G(L,R) = (&-2(I_m - LL^T)YQ_RY^TL,\\ &-2(I_m - RR^T)X(P - Q_RP)).\end{aligned}$$

The action of Riemannian Hessian on some tangent vector $(dL,dR) \in \mathcal{T}_{L,R}\mathcal{M}$ can be obtained using (7) to get,

$$\begin{aligned}&\operatorname{Hess} G(L,R)[dL,dR] =\\ &(-2(I_m - LL^T)(YdQ_RA^TL + YQ_RY^TdL)\\ &+ 2dL(L^TYQ_RY^TL),\\ &-2(I_m - RR^T)X((I_n - Q_R)dP - dQ_RP)\\ &+ 2dR(X(P - Q_RP))).\end{aligned}$$

Using these we can now perform second-order gradient descent on manifold $(\mathcal{M}, g_c)$.

D. Trust region algorithm

In the previous section we defined the gradient and Hessian of the objective function $G(L,R)$ on the manifold $\mathcal{M}$. In this section we describe the Riemannian trust-region algorithm [11] used to solve the optimization problem. At every k^{th} iteration we solve the following trust-region subproblem,

$$\begin{aligned}\min_{(dL,dR)\in\mathcal{T}_{L_k,R_k}\mathcal{M}} m_{L_k,R_k}(dL,dR) &= G(L_k,R_k)\\ &+ g_c(grad\, G(L_k,R_k),(dL,dR))\\ &+ \frac{1}{2}g_c(Hess\, G(L_k,R_k)[dL,dR],(dL,dR))\end{aligned} \tag{10}$$

such that $g_c((dL,dR),(dL,dR)) \leq \Delta_k^2$. The trust region is defined as a ball centered at 0 in $\mathcal{T}_{L_k,R_k}\mathcal{M}$ with a radius $\Delta_k > 0$. Thus, the subproblem finds the tangent vector that minimizes m_{L_k,R_k} within a ball of radius Δ_k. The solution of the trust-region subproblem (dL_k, dR_k) gives the direction of update to the next iteration. This problem is solved iteratively using the truncated conjugated gradient method. A candidate for the next iterate on manifold $\mathcal{M}$ is found using the retraction function,

$$\begin{aligned}(L_{k+1},R_{k+1}) &= \mathrm{Retr}_{L_k,R_k}(dL_k,dR_k)\\ &= (U_LV_L^T, U_RV_R^T),\end{aligned}$$

where U_L and U_R are the left singular vectors and V_L and V_R are the right singular vectors of $L_k + dL_k$ and $R_k + dR_k$ respectively. The decision of whether to accept this candidate or reject it is based on the following ratio,

$$\rho_k = \frac{G(L_k,R_k) - G(L_{k+1},R_{k+1})}{m_{L_k,R_k}(0) - m_{L_k,R_k}(dL_k,dR_k)}, \tag{11}$$

where ρ_k indicates our confidence in the accuracy of the quadratic model $m_{L,R}(dL,dR)$. If ρ_k is greater than a certain threshold ρ', we accept the candidate; otherwise it is rejected. In this study we set $\rho' = 0.1$. Further, ρ_k also informs the radius of the trust-region. A small ρ_k reduces our confidence in the trust-region and Δ_k is decreased and, likewise, if ρ_k is large and close to 1 we increase Δ_k.

If $\rho_k < 0.1$, then the quadratic model $m_{L,R}(dL,dR)$ is inaccurate, so the candidate is rejected and the trust-region radius is reduced. If $\rho_k > 0.1$, the candidate is always accepted; however, if $\rho_k < 1/4$, the radius is reduced. If $\rho_k > 3/4$ and $gc((dL,dR),(dL,dR)) = \Delta_k^2$, then the candidate is accepted and the radius is also increased. The maximum ceiling for the trust-region radius is set to $\sqrt{r}$. Algorithm 2 shows the logical sequence of steps.

The stopping criterion for convergence is placed on the norm of the gradient of $G(L,R)$ (defined by the canonical metric on $\mathcal{M}$) with a threshold of 10^{-6}.

This concludes the algorithm to solve the new low-rank DMD problem. The solution to this problem with this algorithm will hereafter be referred to as lr-DMD.

Algorithm 2 Trust-region algorithm

```
Require: $Y \in \mathbb{R}^{m\times n}$, $X \in \mathbb{R}^{m\times n}$
Guess initial $L_0$,$R_0$
$k = 0$
repeat
    Solve trust-region subproblem (10) to get $(dL_k, dR_k)$
    Compute $\rho_k$ from (11)
    if $\rho_k < 1/4$ then
        $\Delta_{k+1} = \Delta_k/4$
    else if $\rho_k > 3/4$ and $\|(dL_k, dR_k)\|_{g_c} = \Delta_k$ then
        $\Delta_k = \min(2\Delta_k, \sqrt{r})$
    else
        $\Delta_{k+1} = \Delta_k$
    end if
    if $\rho_k > 0.1$ then
        $(L_{k+1}, R_{k+1}) = \mathrm{Retr}_{L_k,R_k}(dL_k, dR_k)$
    else
        $(L_{k+1}, R_{k+1}) = (L_k, R_k)$
    end if
    $k \leftarrow k + 1$
until $\|\mathrm{grad}\, G(L_k, R_k)\|_{g_c} <$ threshold
$D = (L_k^T Y X^T R_k)(R_k^T X X^T R_k)^{-1}(R_k^T L_k)$
return $L_k, D, R_k$
```

V. Building reduced order models

In this section we describe two ways of building reduced order models from the lr-DMD solution. Consider the dynamical system

$$x_{k+1} = Ax_k, \quad x_0 = \theta, \tag{12}$$

where $x_k, \theta \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m\times m}$. The first model reduction technique is based on the SVD of the state transition matrix. From the solution of the optimization problem (L, D, R), we can get the following reduced order model,

$$v_{k+1} = R^T L D v_k, \quad v_0 = R^T \theta, \tag{13}$$

with $x_k = L(R^T L)^{-1} v_k$ and $v_k \in \mathbb{R}^r$. Since the matrix $R^T L \in \mathbb{R}^{r\times r}$ is small its inverse is not very expensive which makes this model reduction feasible. The second technique is based on eigenvalue decomposition of the state transition matrix. Given the solution (L, D, R), we consider the eigenvalue decomposition of $D(R^T L) = V\Lambda V^{-1}$ to get the following reduced order model

$$v_{k+1} = \Lambda v_k, \quad v_0 = (R^T L V)^{-1} R^T \theta, \tag{14}$$

with $x_k = LVv_k$ to get back the state variable in $\mathbb{R}^m$.

VI. Dynamical system with control

So far we have looked at data from a single trajectory of a linear system with no control input. The data matrices Y and X generated from such snapshots have very similar left images. However, when considering dynamical systems with control inputs, the state trajectory can differ significantly from the image of the state transition matrix. For such cases, optimization over both left and right images of the lr-DMD solution allows us to capture the proper dynamics of the system. Consider a general controlled dynamical system

$$x_{k+1} = Ax_k + Bu_k, \quad x_0 = \theta,$$

with some control input sequence u_k and known control matrix B. The data matrices can be constructed as follows

$$X := (x_0|\cdots|x_{n-1}), \quad Y := (x_1 - Bu_0|\cdots|x_n - Bu_{n-1}).$$

The lr-DMD method can be applied on these data matrices to infer a low-rank approximation of A.

VII. Numerical Experiments

In this section we evaluate the performance of lr-DMD and compare it to DMD [4] and OMD [8] for two problems. We use ManOpt [12] to implement algorithm 2 in MATLAB.

A. *Fabricated toy problem*

We fabricate a toy problem to evaluate the performance of lr-DMD on a controlled linear system. We consider a dynamical system of dimension $m = 50$ and a state transition matrix of rank 30. The system is governed by the following dynamics

$$x_{k+1} = Ax_k + u_k, \quad x_0 = \theta,$$

where $x_k \in \mathbb{R}^m$ is the state variable and $u_k \in \mathbb{R}^m$ is the control input. The state transition matrix is generated using random vectors that are independently sampled from a normal distribution such that the rank of the resulting matrix is 30. The input sequence u_k and initial condition θ are also generated by independent samples of the same distribution. The data snapshots are generated for $n = 30$ as follows,

$$X := (x_0|\cdots|x_{n-1}), \quad Y := (x_1 - u_0|\cdots|x_n - u_{n-1}).$$

The performance of the reduced order models is evaluated by the following error metric

$$\epsilon = \left\| Y - \hat{A}X \right\|$$

where $\hat{A}$ is the reduced order state transition matrix and $\|.\|$ indicates the Frobenius norm. This metric indicates the quality of data reconstruction of the reduced order model. Figure 1 shows the error ϵ as we increase the rank of the $\hat{A}$ for the three reduced order models. Clearly lr-DMD outperforms OMD and DMD at all rank approximations. This can be attributed to the fact that the left singular vectors of the data matrices Y and X differ significantly.

B. *Ginzburg-Landau equation*

Even though we have only considered real matrices in this study so far, all the methods and derivations can trivially be extended to complex matrices. As a model problem for fluid flows, we consider the linearized complex Ginzburg-Landau equation,

$$\frac{\partial q}{\partial t} + \nu\frac{\partial q}{\partial x} = \mu(x)q + \gamma\frac{\partial^2 q}{\partial x^2},$$
$$\text{with} \quad \mu(x) = \mu_0 - c_u^2 + \mu_2 x^2/2,$$

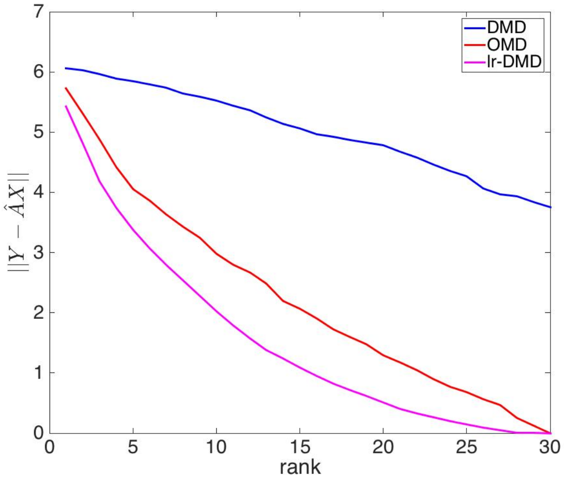

Fig. 1. The error for projected DMD(blue), OMD(red) and Low-rank DMD(magenta) for increasing rank r of the reduced order model applied to a random linear system with control.

where the real part of $q(x,t)$ represents velocity or streamfunction perturbation amplitude. The value of constants like ν, c_u, μ_0 and μ_2 have been taken corresponding to the supercritical, globally unstable regime in [13]. The dimension of the system is $m = 220$ and we discretize in time with $dt = 1$. The initial condition is taken as a guassian near the origin and we generate snapshots for $n = 15$ iterations.

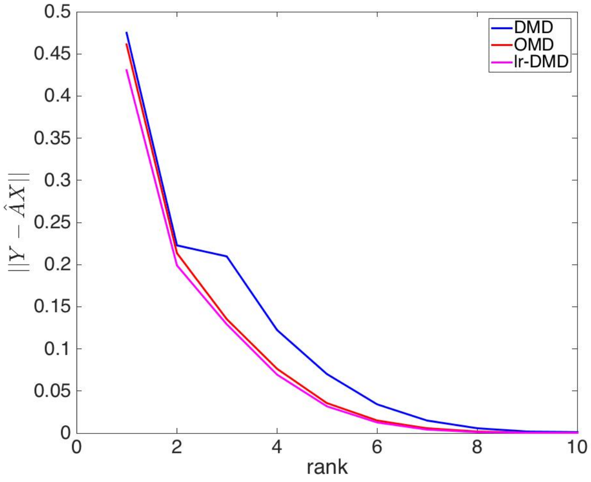

Fig. 2. The error for projected DMD(blue), OMD(red) and Low-rank DMD(magenta) for increasing rank r of the reduced order model applied to linearized complex Ginzburg-Landau equations.

Figure 2 shows the error metric ϵ for the three methods for different rank approximations. In this case we see OMD performs nearly as well as lr-DMD whereas DMD shows greater error for ranks 3 to 8. The improved performance of both DMD and OMD compared to the previous experiment can be attributed to the fact that the data matrices have nearly identical left images. It is expected that OMD will perform nearly as well as lr-DMD when the data matrices satisfy this condition.

VIII. Conclusions

In this study we find the low-rank dynamic mode decomposition (lr-DMD) solution using two methods. The first method is a subspace projection method which is computationally efficient and gives near-optimal solution under certain conditions. The second method is Riemannian trust-region based gradient descent on Gassmannian manifolds. We compute the gradient and Hessian of the objective function to use in the trust-region method for the optimization. Both these methods make no assumption on the subspace spanned by the data matrices Y and X which is suitable for control problems where the state trajectory can lie in a subspace significantly different to the span of the state transition matrix. We demonstrate the performance of lr-DMD on a fabricated controlled dynamical system and linearized complex Ginzburg-Landau equations. For both cases lr-DMD outperformed both DMD and OMD in data reconstruction.

Appendix I
Details for Hessian

The expressions of dQ_R and dP used in the definition of the Euclidean Hessian are described as follows,

$$
\begin{aligned}
dQ_R = {} & X^T dR(R^T X X^T R)^{-1} R^T B \\
& - X^T R(R^T X X^T R)^{-1}(dR^T X X^T R)(R^T X X^T R)^{-1} R^T X \\
& - X^T R(R^T X X^T R)^{-1}(R^T X X^T dR)(R^T X X^T R)^{-1} R^T X \\
& + X^T R(R^T X X^T R)^{-1} dR^T X
\end{aligned}
$$

$$
\begin{aligned}
dP = {} & Y^T dL L^T Y X^T R(R^T X X^T R)^{-1} \\
& + Y^T L dL^T Y X^T R(R^T X X^T R)^{-1} \\
& + Y^T L L^T Y X^T dR(R^T X X^T R)^{-1} \\
& - Y^T L L^T Y X^T R(R^T X X^T R)^{-1}(dR^T X X^T R)(R^T X X^T R)^{-1} \\
& - Y^T L L^T Y X^T R(R^T X X^T R)^{-1}(R^T X X^T dR)(R^T X X^T R)^{-1}
\end{aligned}
$$

References

[1] K. Carlberg, C. Bou-Mosleh, and C. Farhat, "Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations," *International Journal for Numerical Methods in Engineering*, vol. 86, no. 2, pp. 155–181, 2011.

[2] C. Penland, "Random forcing and forecasting using principal oscillation pattern analysis," *Monthly Weather Review*, vol. 117, no. 10, pp. 2165–2185, 1989.

[3] H. Akaike, "Fitting autoregressive models for prediction," *Annals of the institute of Statistical Mathematics*, vol. 21, no. 1, pp. 243–247, 1969.

[4] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.

[5] M. R. Jovanović, P. J. Schmid, and J. W. Nichols, "Sparsity-promoting dynamic mode decomposition," *Physics of Fluids*, vol. 26, no. 2, p. 024103, 2014.

[6] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: theory and applications," *arXiv preprint arXiv:1312.0041*, 2013.

[7] P. Héas and C. Herzet, "Low-rank approximation and dynamic mode decomposition," *arXiv preprint arXiv:1610.02962*, 2016.

[8] P. J. Goulart, A. Wynn, and D. Pearson, "Optimal mode decomposition for high dimensional systems," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 4965–4970.

[9] A. Edelman, T. A. Arias, and S. T. Smith, “The geometry of algorithms with orthogonality constraints,” *SIAM journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[10] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[11] P.-A. Absil, C. G. Baker, and K. A. Gallivan, “Trust-region methods on riemannian manifolds,” *Foundations of Computational Mathematics*, vol. 7, no. 3, pp. 303–330, 2007.

[12] N. Boumal, B. Mishra, P.-A. Absil, R. Sepulchre *et al.*, “Manopt, a matlab toolbox for optimization on manifolds.” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1455–1459, 2014.

[13] S. Bagheri, D. Henningson, J. Hoepffner, and P. Schmid, “Input-output analysis and control design applied to a linear model of spatially developing flows,” *Applied Mechanics Reviews*, vol. 62, no. 2, p. 020803, 2009.