Parametric Matrix Models for Equation-Informed Reduced Order Modeling

Linearity, Polynomial Nonlinearity, Arbitrary Nonlinearity, Macrostate Data, and Higher-Order Linear Operators

Patrick Cook

Lee Research Group

1 Linearity

1.1 Proper Orthogonal Decomposition Review

A review of the Proper Orthogonal Decomposition (POD) technique is best done through a simple example. Consider the following parametric coupled linear differential equation initial value problem

$$\begin{aligned}\frac{d\mathbf{u}}{dt} &= \mathbf{A}(\vec{c})\mathbf{u} + \mathbf{b}(\vec{c})\\ \mathbf{u}(0) &= \mathbf{u}_0,\end{aligned}$$

where $\mathbf{u} \in \mathbb{C}^n$, $\mathbf{A}(\vec{c}) \in \mathbb{R}^p \to \mathbb{C}^{n\times n}$, and $\mathbf{b}(\vec{c}) \in \mathbb{R}^p \to \mathbb{C}^n$. That is, the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ depend on p real-valued "model parameters", $\vec{c}$. For the rest of this note we will omit the bold face notation for vectors and matrices and the type of a variable will be clear from the context. Given N snapshots for the solutions to this equation, we can form the snapshot matrix

$$X = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_N \\ | & | & & | \end{bmatrix},$$

which is $\in \mathbb{C}^{n\times N}$. The snapshots can be solution vectors at different times, different model parameters, or both. The POD technique projects the original problem onto the subspace spanned by the first r principle components of the snapshot data, which are the r left singular vectors of the snapshot matrix X. The POD projector, P, is given by

$$\begin{aligned}X &\xrightarrow{\text{SVD}} U\Sigma V^H\\ P &= U[:, 1:r]\end{aligned}$$

where we have adopted the convention that the singular values are in descending order and where $U[:, 1:r]$ denotes the first r columns of U. The reduced order model (ROM) is then given by

$$\begin{aligned}\frac{dy}{dt} &= \tilde{A}(c)y + \tilde{b}(c)\\ y(0) &= y_0,\\ &\text{where}\\ \tilde{A}(c) &= P^H A(c) P,\\ \tilde{b}(c) &= P^H b(c),\\ y_0 &= P^H u_0.\end{aligned}$$

Note that all objects in the ROM are in the much smaller space of size r, $\tilde{A} \in \mathbb{C}^{r\times r}$, $\tilde{b} \in \mathbb{C}^r$, and $y \in \mathbb{C}^r$. In general, the complexity of solving the model in the full space is $\mathcal{O}\left(n^3\right)$, while the complexity of solving the model in the reduced space is $\mathcal{O}\left(r^3\right)$, which is much smaller for $r \ll n$. Once the ROM is solved, the solution in the original space can be recovered by performing the inverse projection, $u \approx Py$. Note that as $r \to n$, $PP^H \to \mathbb{I}_n$, and the ROM converges to the full order model.

1.2 Parametric Matrix Models

The POD technique requires that the exact elements of the operators in the full-order (high-fidelity) model are known. In stark contrast, the Parametric Matrix Model (PMM) method requires only knowledge of the form of the equations. That is, how each operator depends on model parameters and how each operator acts on each other. In this case, we still write the ROM following the POD technique, but the elements of the operators are now learned via gradient descent. Explicitly, the PMM for the previous example is

$$\begin{aligned}\frac{dy}{dt} &= \tilde{A}(c)y + \tilde{b}(c)\\ y(0) &= y_0.\end{aligned}$$

Where now the elements of $\tilde{A}(c)$, $\tilde{b}(c)$, and y_0 are learned via gradient descent[1]. This relies on knowledge of the exact or approximate parameter dependence of A and b. A common case is the affine case:

$$\begin{aligned}A(c) &= A_0 + \sum_{i=1}^{p} c_i A_i,\\ b(c) &= b_0 + \sum_{i=1}^{p} c_i b_i,\end{aligned}$$

for which the PMM would follow the same form

$$\begin{aligned}\tilde{A}(c) &= \tilde{A}_0 + \sum_{i=1}^{p} c_i \tilde{A}_i,\\ \tilde{b}(c) &= \tilde{b}_0 + \sum_{i=1}^{p} c_i \tilde{b}_i,\end{aligned}$$

and learn the elements of $\tilde{A}_0$, $\tilde{A}_i$, $\tilde{b}_0$, and $\tilde{b}_i$ via gradient descent.

2 Polynomially Nonlinear Terms

2.1 Proper Orthogonal Decomposition

Consider the simplest possible nonlinear term,

$$u^2.$$

This is not an inner product, but instead the elementwise square. Sometimes this is explicitly written as

$$u \odot u \quad \text{or} \quad u^{\odot 2}.$$

Whenever nonlinear operations appear in the full-order model, the POD technique requires that the reduced order model be projected to the full space to compute the nonlinear terms before being re-projected to the reduced space. This is computationally expensive, but careful associativity can be used to recover efficient scaling. In this case,

$$u \odot u \xrightarrow{\text{POD}} P^H \left[(Py) \odot (Py)\right].$$

Switching to index notation, the i^{th} index of y in the reduced order computation of $u \odot u$ is

$$\begin{aligned}y_i &= (P^H)_{i\mu}(Py)_\mu (Py)_\mu\\ &= \left[(P^H)_{i\mu} P_{\mu j} P_{\mu k}\right] y_j y_k\\ &= T^{(2)}_{ijk} y_j y_k,\end{aligned}$$

where $T^{(2)}_{ijk} = (P^H)_{i\mu} P_{\mu j} P_{\mu k}$ is a rank-3 tensor $\in \mathbb{C}^{r\times r\times r}$ which we refer to as the "nonlinearity tensor" for the elementwise square operation. The indeces i, j, and k are in the reduced dimension, while the index μ is in the full dimension. This tensor can be precomputed at a cost of $\mathcal{O}\left(nr^3\right)$ which is efficient for $r \ll n$.

[1] In practice, the elements of u_0 are usually known and as such y_0 is not learned but instead is set to $P^H u_0$.

Consider the general case

$$u^q \quad \equiv \quad u \odot \overset{q}{\cdots} \odot u \quad \equiv \quad u^{\odot q}.$$

for $q \in \mathbb{N}$. It is easy to see that the nonlinearity tensor for this operation is

$$T^{(q)}_{ij_1j_2\cdots j_q} = (P^H)_{i\mu} P_{\mu j_1} P_{\mu j_2} \cdots P_{\mu j_q},$$

which can be precomputed at a cost of $\mathcal{O}\left(nr^{q+1}\right)$. This is not efficient beyond $q = 2$. The reason for this is that, in general, the cost to solve the reduced order model is $\mathcal{O}\left(r^3\right)$, and thus for $q > 2$ the cost of precomputing the nonlinearity tensor is greater than the cost of solving the reduced order model, which changes the overall scaling of the method. In fact, even just applying this tensor in the reduced space costs $\mathcal{O}\left(r^{q+1}\right)$.

2.2 Parametric Matrix Models

The PMM method can be used to learn the nonlinearity tensor for any q, and thus bypass the prohibitive cost of computing the nonlinearity tensor for $q > 2$. This is done similarly to linear PMMs. For the case of the elementwise exponentiation to the power of q, the i^{th} element in the reduced space is

$$\tilde{T}^{(q)}_{ij_1j_2\cdots j_q} y_{j_1} y_{j_2} \cdots y_{j_q},$$

where the elements of the nonlinearity tensor are now learned via gradient descent. However, this still requires learning $\mathcal{O}\left(r^{q+1}\right)$ elements, which is both computationally and data intensive. Compounding this, the cost to apply this tensor in the reduced space is still $\mathcal{O}\left(r^{q+1}\right)$. This is where the PMM employs further approximations to the nonlinearity tensor. The simplest approximation that works well is to decompose the nonlinearity tensor in the same way as is required to construct it exactly, but with a reduced dimension. That is, we write

$$\tilde{T}^{(q)}_{ij_1j_2\cdots j_q} = \left(\tilde{Q}^H\right)_{i\mu} \tilde{Q}_{\mu j_1} \tilde{Q}_{\mu j_2} \cdots \tilde{Q}_{\mu j_q},$$

where $\tilde{Q} \in \mathbb{C}^{s \times r}$ with $r \leq s \ll n$ and the elements of $\tilde{Q}$ are learned via gradient descent instead of the elements of $\tilde{T}^{(q)}$. This reduces the number of elements to learn to $\mathcal{O}\left(sr\right)$. By applying each $\tilde{Q}$ to each y in the reduced space when computing the nonlinear term,

$$\begin{aligned}
\tilde{T}^{(q)}_{ij_1j_2\cdots j_q} y_{j_1} y_{j_2} \cdots y_{j_q} &= \left(\tilde{Q}^H\right)_{i\mu} \tilde{Q}_{\mu j_1} \tilde{Q}_{\mu j_2} \cdots \tilde{Q}_{\mu j_q} y_{j_1} y_{j_2} \cdots y_{j_q} \\
&= \left(\tilde{Q}^H\right)_{i\mu} \left(\tilde{Q}y\right)_{\mu} \left(\tilde{Q}y\right)_{\mu} \cdots \left(\tilde{Q}y\right)_{\mu} \\
&= \left(\tilde{Q}^H\right)_{i\mu} \left[\left(\tilde{Q}y\right)^{\odot q}\right]_{\mu},
\end{aligned}$$

the cost is reduced to just $\mathcal{O}\left(2sr\right)$.

3 Arbitrary Nonlinearity

Arbitrary nonlinear terms can be reduced to polynomially nonlinear terms via "lifting transformations" [1]. This is done by introducing new variables and operators to the full-order model before projecting down to the reduced space. Consider a simple example

$$\frac{du}{dt} = A\sin(u),$$

where $\sin(u)$ denotes the elementwise sin of the vector u. One could consider replacing $\sin(u)$ with a Taylor series expansion, but a more sophisticated approach is to introduce a new variable $v = \sin(u)$, so

$$\begin{aligned}
\frac{du}{dt} &= Av, \\
\frac{dv}{dt} &= \cos(u) \odot \frac{du}{dt} = \cos(u) \odot (Av).
\end{aligned}$$

This still requires a non-polynomial nonlinear operation, so we introduce another new variable $w = \cos(u)$, so

$$\begin{aligned}\frac{du}{dt} &= Av,\\ \frac{dv}{dt} &= w \odot (Av),\\ \frac{dw}{dt} &= -\sin(u) \odot \frac{du}{dt} = -v \odot (Av).\end{aligned}$$

And now all nonlinear operations are simple elementwise products.

This process can be done in general for any arbitrary nonlinear term at the expense of a larger full-order model that is potentially less well-conditioned and more stiff.

4 Macrostate Data

We now consider a case where the POD method is not possible, regardless of the complexity of the nonlinear terms. Suppose instead of the state vector, we only had snapshots for some "macrostate" of the system. That is, we know the form of the governing equations of the system as well as the equation relating the state vector to the data we have, but no snapshots of the state vector itself. A common and realistic example is that of probability densities in quantum mechanics. Consider the time-dependent Schrödinger equation

$$\begin{aligned}\rho(t) &= \psi(t)^* \odot \psi(t),\\ i\frac{\partial \psi}{\partial t} &= H\psi,\\ \psi(0) &= \psi_0,\end{aligned}$$

but we only have snapshots of the probability density, $\rho = \psi^* \odot \psi$. In general, there does not exist any equation that describes the dynamics of this system purely in terms of ρ. You can convince yourself of this by considering the case where ψ is a gaussian wavepacket with non-zero momentum, $\psi_0 \sim e^{-\frac{1}{2}(x-x_0)^2} e^{ik_0 x}$ and thus $\rho_0 \sim e^{-(x-x_0)^2}$. In this case all the information about the momentum is lost in ρ. Without snapshots of the state vector ψ, we cannot construct a POD basis and thus the POD method is not possible.

All is not lost, however, as the PMM method can still be used. We start by pretending that we had the projector to the POD basis, P_ψ. We are still able to compute the projector to the POD basis of the macrostate data, P_ρ. We use these two projectors to construct the reduced order model,

$$\begin{aligned}p &= P_\rho^H \left[(P_\psi \phi)^* \odot (P_\psi \phi) \right],\\ i\frac{d\phi}{dt} &= P_\psi^H H P_\psi \phi,\\ \phi(0) &= P_\psi^H \psi_0,\end{aligned}$$

where p is the density in the reduced space and ϕ is the state vector in the reduced space. We can treat the nonlinear operation of $\psi^* \odot \psi$ as in the previous sections to group the various projectors into a single tensor $\tilde{T}^{(2)}_{ijk}$ whose elements are to be learned via gradient descent. All other parts of the reduced order model that depend on the inaccessible P_ψ can similarly be grouped together into operators to be learned via gradient descent.

$$\begin{aligned}p_i &= \tilde{T}^{(2)}_{ijk} \phi_j^* \phi_k,\\ i\frac{d\phi}{dt} &= \tilde{H}\phi,\\ \phi(0) &= \tilde{\phi}_0,\end{aligned}$$

where the prediction of ρ in the original space is then $P_\rho p$. This method carefully avoids the need for any snapshots of the state vector directly.

5 Higher-Order Linear Operators

Finally, we consider the case where the governing equations of the system contain higher-order linear operators. These are simple tensor contractions beyond rank-2. Consider the following example

$$\frac{dx}{dt} = \lambda^x_{xx} x^2 + \lambda^x_{xy} xy + \lambda^x_{yy} y^2,$$
$$\frac{dy}{dt} = \lambda^y_{yx} yx + \lambda^y_{yy} y^2 + \lambda^y_{xx} x^2,$$

where $\lambda^i_{jk}, x, y \in \mathbb{C}$. This system may equivalently be written as

$$\frac{du_i}{dt} = A^i_{jk} u_j u_k,$$
$$u = \begin{bmatrix} x \\ y \end{bmatrix},$$

where $A^i_{jk} = \lambda^i_{jk}$ and $u \in \mathbb{C}^2$. The projection to the POD basis follows the same form as before,

$$\frac{dw_i}{dt} = \left(P^H\right)_{i\mu_i} A^{\mu_i}_{\mu_j \mu_k} P_{\mu_j j} P_{\mu_k k} w_j w_k,$$
$$w = P^H u.$$

Again by associativity, the transformed operator can simply be written as a single rank-3 tensor,

$$T_{ijk} = \left(P^H\right)_{i\mu} A^{\mu}_{\nu\sigma} P_{\nu j} P_{\sigma k},$$

and the PMM for this system would just learn the elements of T_{ijk} via gradient descent, or the elements of some decomposition of T_{ijk}.

Notice that this method yields equivalent reduced order models to polynomially nonlinear terms. This is because an equivalent way to handle polynomially nonlinear terms is to consider them as higher-order linear operators.

References

[1] Boris Kramer and Karen E. Willcox. Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition. *AIAA Journal*, 57(6):2297–2307, 2019.