TEXAS A&M UNIVERSITY
Engineering

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

Petrobras Workshop: Scientific Machine Learning (SciML) and Data-Driven Model Reduction for Reservoir Simulation

Eduardo Gildin
Petroleum Engineering Department at
Texas A&M University

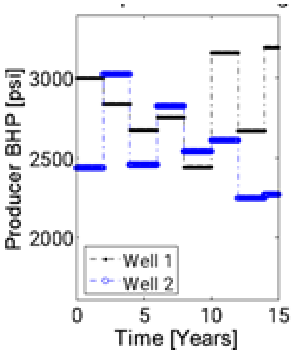
August 11-15, 2025

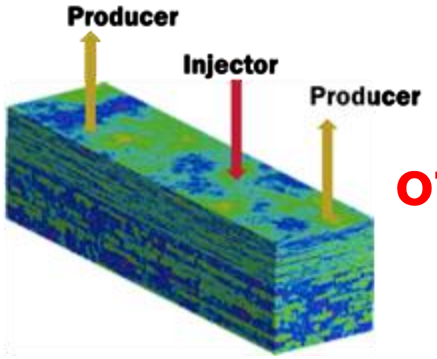
TEXAS A&M UNIVERSITY
Engineering

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

Non-Linear Model Reduction

POD-Hybrid-based ROM



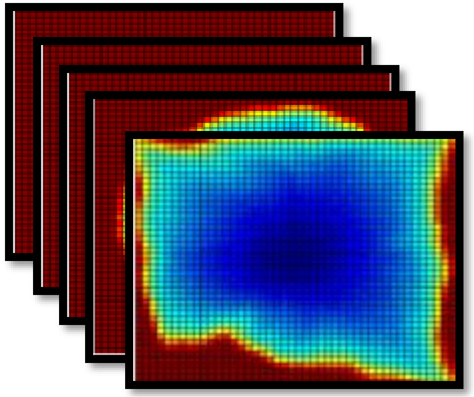


offline

$$\mathbf{R}\left(\mathbf{x}^{n+1},\mathbf{x}^{n},\mathbf{u}^{n+1}\right)=\mathbf{F}\left(\mathbf{x}^{n+1}\right)+\mathbf{A}\left(\mathbf{x}^{n+1},\mathbf{x}^{n}\right)+\mathbf{Q}\left(\mathbf{x}^{n+1},\mathbf{u}^{n+1}\right)$$

$$\mathcal{J}=\frac{\partial \mathcal{R}}{\partial \mathbf{x}}$$

X→ state (pressures, saturations)



Form Φ

$$[U,S,V]=SVD(X)$$

$$\dot{x}=f(x)+g(x)u$$

$$r=\dot{x}_r-f(\Phi x_r)-g(\Phi x_r)u$$

$$\dot{x}_r=\Phi^T f(\Phi x_r)+\Phi^T g(\Phi x_r)u$$

Approximate

$\mathbf{x}\approx\mathbf{\Phi}\mathbf{x}_r$

$\mathbf{\Phi}\in\mathbb{R}^{N\times r}$

$$\mathbf{\Phi}=\mathbf{U}[:,1:\mathbf{r}]$$

Project

$\mathbf{\Psi}^{\top}\mathbf{r}=0$

(Galerkin: $\mathbf{\Psi}=\mathbf{\Phi}$)

Question 1→ how to find proper projections?

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

Proper Orthogonal Decomposition → POD

Proper Orthogonal Decomposition : POD

(Non-linear systems).

$\dot{x} = f(x, t, u)$

$\dot{x}(t) = Ax(t) + Bu(t)$

$y(t) = Cx(t) + Du(t)$

For the time being → $\dot{x}(t) = Ax(t) + f(x(t)) + Bu(t)$

Approx using Projection → $x(t) = V x_r(t)$

$V \in \mathbb{R}^{n\times r}$ $\quad W^T V = I$

$W \in \mathbb{R}^{n\times r}$

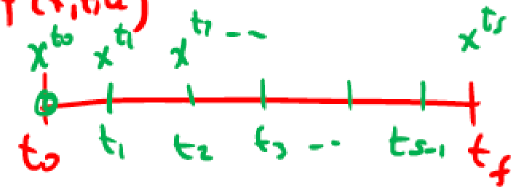
$V^T V = I_r$ $\quad$ (I & $V = W$ → orthogonal Proj)

$$\dot{x}_r(t) = V^T A V x_r(t) + V^T f(V x_r(t)) + V^T B u(t)$$

$$\boxed{\dot{x}_r(t) = A_r x_r(t) + V^T f(\overbrace{V x_r(t)}^{n\times 1}) + B_r u(t)}$$

Computation still in "n"

- Let's assume I can simulate $\dot{x} = f(x,t,u)$
- Collect snapshots



$$\underbrace{X = \left[x(t_0) \quad x(t_1) \quad \cdots \quad x(t_s) \right]} \in \mathbb{R}^{n \times s}$$

Consequence → any state $x(t)$ is in the image of X.

$$x(t) = \alpha_1 v_1 + \alpha_2 v_2 + \cdots \alpha_n v_n$$

Basis v_i's for $\mathrm{Im}(X)$

$\Downarrow$

IDEA of POD → Extract the dominant subspace of X to construct V

$$V = [v_1 \quad v_2 \quad \cdots \quad v_r]$$ r dominant subspaces

If the matrix V is found, then

$$x(t_k) \approx \sum_{i=1}^{r} \alpha_i^k v_i \rightarrow \alpha_i^k = \langle v_i, x(t_k) \rangle$$

inner product

$$v_i^T x(t_k) \rightarrow$$ Do for all v_i's, α_i's

$$\left[x(t_1) \;\; x(t_2) \;\; \cdots \;\; x(t_s) \right] \approx \left[v_1 \;\; v_2 \cdots \;\; v_r \right] \begin{bmatrix} \alpha_1^1 & \alpha_1^2 & \cdots & \alpha_1^s \\ \vdots & \vdots & & \vdots \\ \alpha_r^1 & \alpha_r^2 & & \alpha_r^s \end{bmatrix}$$

Note: $\begin{bmatrix} \alpha_1^k \\ \alpha_2^k \\ \vdots \\ \alpha_r^k \end{bmatrix} = \begin{bmatrix} \langle v_1, x(t_k) \rangle \\ \langle v_2, x(t_k) \rangle \\ \vdots \end{bmatrix} = V^T x(t_k)$

Projection matrix

End up with: $\boxed{X_{\text{approx.}} \approx \boxed{V V^T} X}$

Snapshot

Hence: given snapshots $X \in \mathbb{R}^{n\times s}$

Find $V \in \mathbb{R}^{n\times r}$, $V^T V = I_r$

$$\min: \underbrace{\| X - \underline{V V^T X} \|}_{\text{take SVD of } X.} = \| X - X_{approximated} \|$$

POD-DEIM

1. Solve the original nonlinear system to get the snapshots

$$X = (x_{t_1}, \ldots x_{t_m})$$

2. Get the POD vectors of rank q from SVD of X

$$X = \tilde{U}\Sigma\tilde{V}^T, V = (\tilde{u}_1, \ldots, \tilde{u}_q)$$

3. Use V to get the ROM

$$V^T E V \frac{dz(t)}{dt} = V^T f(Vz(t)) + V^T Bu(t)$$

How to deal with $f(Vz(t))$?
An effective way is to approximate the nonlinear function by projecting it onto a subspace with dimension $l << n$, that approximates the subspace spanned by the snapshots of the nonlinear function.

$$f(t) \approx Uc(t), U = (u_1, \ldots, u_l)$$

DEIM Algorithm - Greedy

To determine $c(t)$, we select m different rows from the overdetermined system

$$f(t) = Uc(t).$$

In particular , consider a matrix

$$P = [e_{\wp_1}, \dots, e_{\wp_l}] \in R^{n \times m},$$

Suppose $P^T U$ is nonsingular, then

$$P^T f(t) = P^T U c(t) \Rightarrow c(t) = (P^T U)^{-1} P^T f(t)$$

so that,

$$f(t) \approx Uc(t) = U(P^T U)^{-1} P^T f(t).$$

How to compute U and how to specify the indices $\wp_i, i = 1, \dots, l$?

Compute U:

1. Collect the snapshots of $f(x(t))$ into a matrix $F=(f(x_{t_1}),\ldots,f(x_{t_m}))$.
2. Apply SVD to $F: F=U^F\Sigma(V^F)^T$
3. $U=(u_1^F,\ldots,u_l^F)$.

Come back to $V^T f(Vz(t))$:

$$f(Vz(t)) \approx U(P^TU)^{-1}P^T f(Vz(t)).$$

If $f(x(t))=(f_1(x_1(t)),\ldots,f_n(x_n(t)))$, then

$$P^T f(Vz(t))=(f_{\wp_1}(\tilde{x}_{\wp_1}),\ldots,f_{\wp_l}(\tilde{x}_{\wp_l}))^T$$

where $\tilde{x}=Vz(t)$.

Finally,

$$V^T f(Vz(t)) \approx \underline{V^TU(P^TU)^{-1}}P^T f(Vz(t))$$

can be precomputed before solving the ROM

Computation of $V^T f(Vz(t))$ during solving ROM is independent of n.

Algorithm Discrete Empirical Interpolation Method (DEIM)

Input : POD basis $\{u_i^F\}_{i=1}^l$ for F

Output : $\vec{\wp}=[\wp_1,\ldots\wp_l]^T \in R^l$

1. $[|\rho|,\wp_1]=\max\{|u_1^F|\}$
2. $U=[u_1^F], P=[e_{\wp_1}], \vec{\wp}=[\wp_1]$
3. for $i=2$ to l do
4. Solve $(P^TU)\alpha=P^Tu_i^F$ for α, where $\alpha=(\alpha_1,\ldots,\alpha_{i-1})^T$
5. $r=u_i^F-U\alpha$
6. $[|\rho|,\wp_l]=\max\{|r|\}$
7. $U\leftarrow[U\, u_i^F], P\leftarrow[P\, e_{\wp_i}], \vec{\wp}\leftarrow\begin{bmatrix}\vec{\wp}\\ \wp_i\end{bmatrix}$
8. end for