Randomized model order reduction

Voloskov Dmitry
Rozzonelli Gabriel
Ishimbaev Marsel
Stupin Maxim

Problem statement

- Input problems might be complex: non-linear and multidimensional, for example, in fluid dynamics (Rowley, 2005)
- Reduce the dimensionality - approximate the system with fewer linear equations
- Methods of choice: Proper Orthogonal Decomposition (POD, POD-DEIM) and Dynamic Mode Decomposition (DMD) (Schmid, 2010)
- Both can be done using SVD, the goal is to improve SVD using randomized matrix decompositions (rSVD) (Alla, Kutz, 2019)

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} - \theta \Delta u(x,t) - \mu \left(u(x,t) - u^3(x,t) \right) = 0 \ , & (x,t) \in \Omega \times [0,T] \\ u(x,0) = u_0(x) \ , & x \in \Omega \\ u(\cdot,t) = 0 \ , & t \in [0,T] \end{cases}$$

Proper Orthogonal Decomposition (POD)

Assuming:

- We know the solution of the system at some evenly spaced times (called *snapshots*)

Then:

- Truncated SVD of the matrix formed from snapshots gives a low-rank basis ψ to project our dynamics onto.

```
# Let Y be the matrices of snapshots
function POD(Y, l):

    # Compute SVD of Y
    U, Σ, V = SVD(Y)
    # Keep the first l columns of U
    ψ = U[:,:l]

    return ψ
```

Discrete Empirical Interpolation Method (DEIM)

Assuming:

- We know the solution of the system at some evenly spaced times (called *snapshots*) and the corresponding snapshots of nonlinear part.

Then:

- Truncated SVD of the matrix formed from snapshots gives a low-rank basis ψ to project our dynamics onto.
- We find the points which impact on the nonlinear part projection is the biggest.
- Calculate nonlinear term in these points.
- Interpolate it to the whole domain.

```
# Let Y be the matrices of snapshots
# Let X be the matrix of nonlinear snapshots
function POD-DEIM(Y, X):

    # Compute SVD of Y
    U, Σ, V = SVD(Y)
    # Keep the first l columns of U
    ψ = U[:,:l]
    # Compute SVD of ψᵀX
    Ux, Σx, Vx = SVD(ψψᵀX)
    # Find projection matrix P
    P = DEIM(Ux)

    return ψ, P
```

Dynamic Mode Decomposition (DMD)

Equation-free method

Assuming:

- We know the solution of the system at some evenly spaced times (called *snapshots*)

Then:

- DMD provides the best eigen-decomposition of the best fit linear system relating the snapshots.

```
# Let Y be the matrices of snapshots
function DMD(Y):

    # Split snapshots
    Y0, Y1 = Y[:,:-1], Y[:,1:]
    # Compute SVD of Y0
    U, Σ, V = SVD(Y0)
    # Compute the matrix A
    A = U^T(Y1)V^TΣ^-1
    # Compute eigen-decomposition of A
    W, Λ = eig(A)
    # Set ψ
    ψ = (Y1)VΣ^-1W

    return ψ
```

Randomized Singular Value Decomposition (rSVD)

Replace SVD with its randomized version in POD and DMD:

```
# Randomized SVD of a matrix Y of size n × m
function rSVD(Y, p):

    # Draw a Gaussian random matrix of size m × p with mean 0 and variance 1
    Ω = random((m, p), 0, 1)
    # Form a sample matrix
    X = YΩ
    # Compute the QR decomposition of X
    Q, R = QR(X)
    # Build a matrix B
    B = QᵀY
    # Compute the SVD decomposition of B
    Û, Σ, V = SVD(B)
    # Compute final U
    U = QÛ

    return U, Σ, V
```

Numerical experiments

Equation:

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} - \theta \Delta u(x,t) - \mu \left(u(x,t) - u^3(x,t) \right) = 0 \ , & (x,t) \in \Omega \times [0,T] \\ u(x,0) = u_0(x) \ , & x \in \Omega \\ u(\cdot,t) = 0 \ , & t \in [0,T] \end{cases}$$

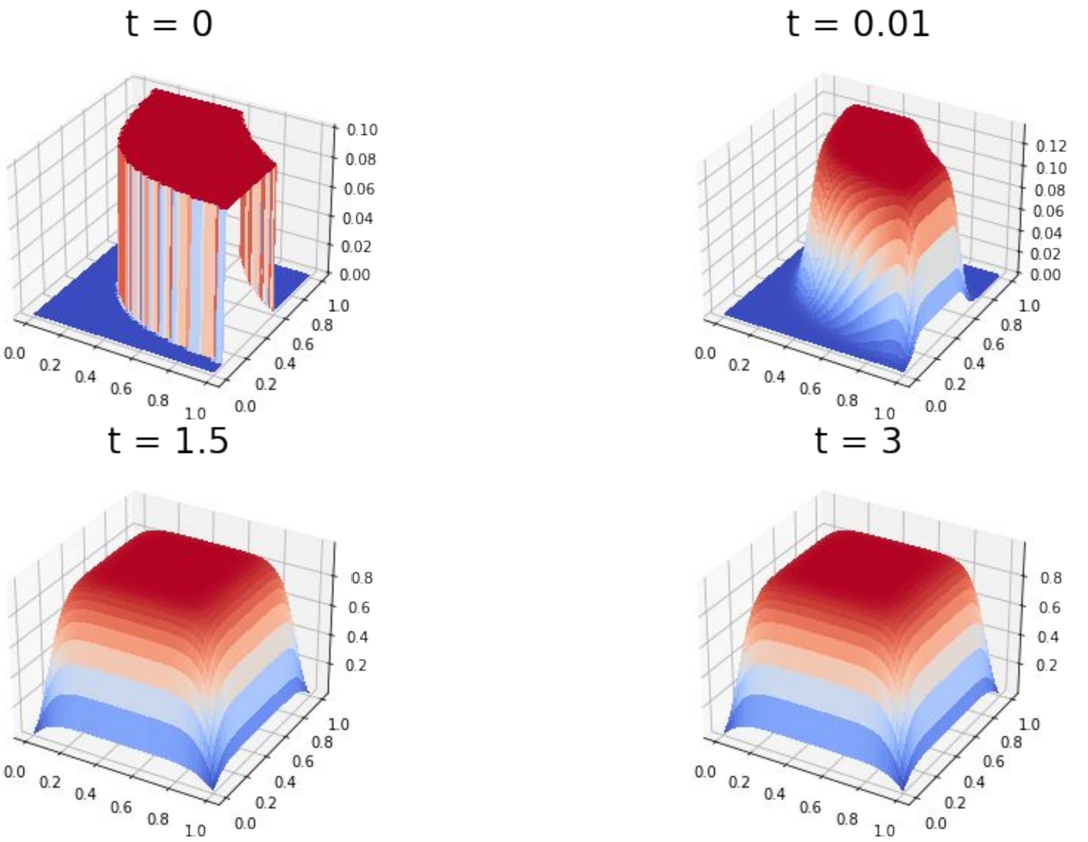
$$\Omega = [0,1] \times [0,1] \qquad x = (x_1, x_2)$$

$$\theta = 0.1 \qquad y_0(x) = \begin{cases} 0.1 \ , & 0.1 < x_1 x_2 < 0.6 \\ 0 \ , & \text{elsewhere} \end{cases}$$

$$\mu = 30$$

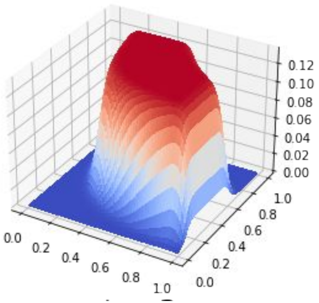
$$T = 5$$

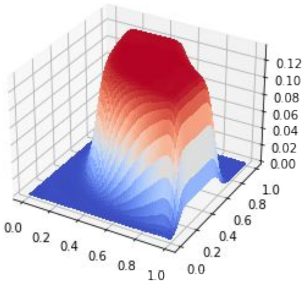
Numerical Experiments. Original Snapshots

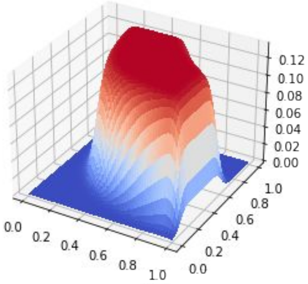


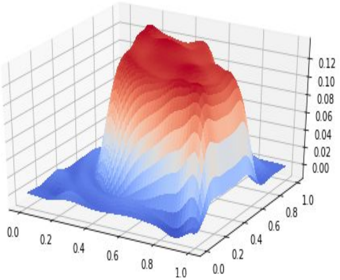
Solution Comparison

t = 0.01









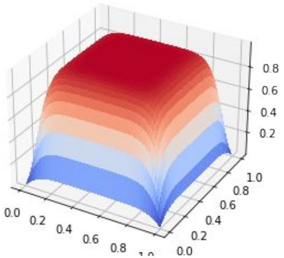
Original

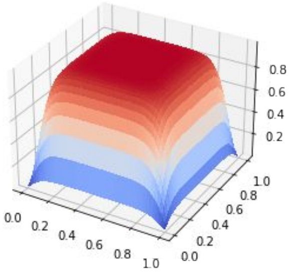
POD

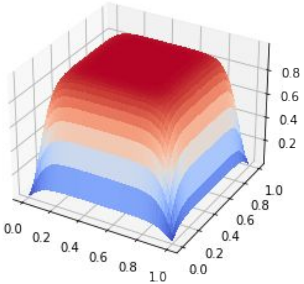
DMD

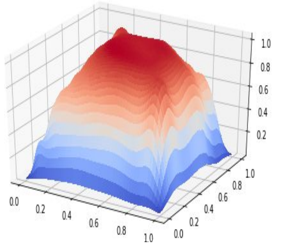
cPOD-cDEIM

t = 3

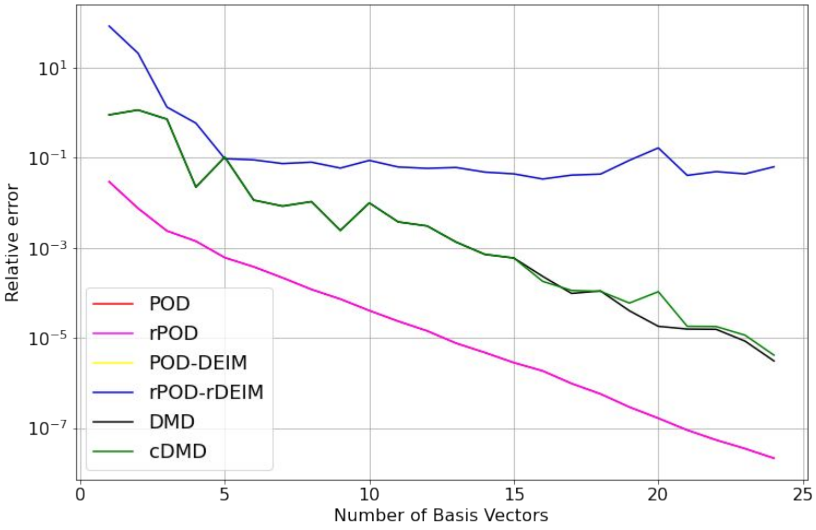


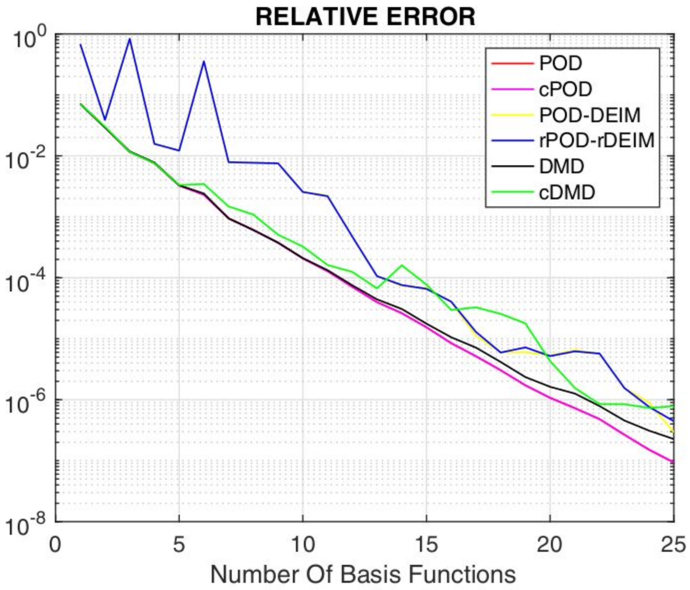




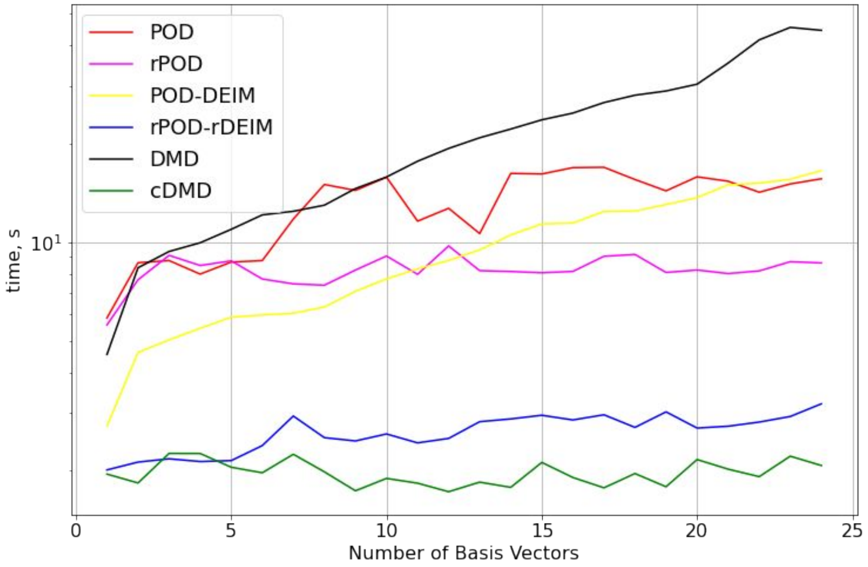


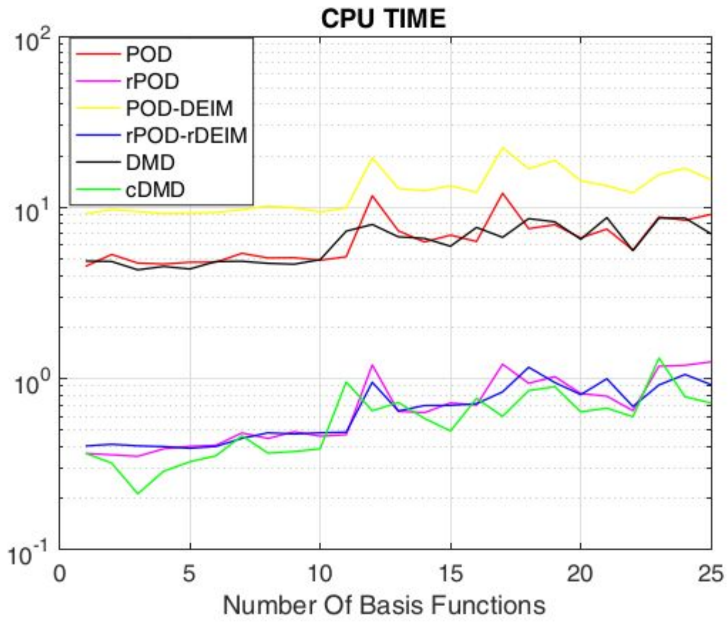
Comparison plots: Error vs. #components



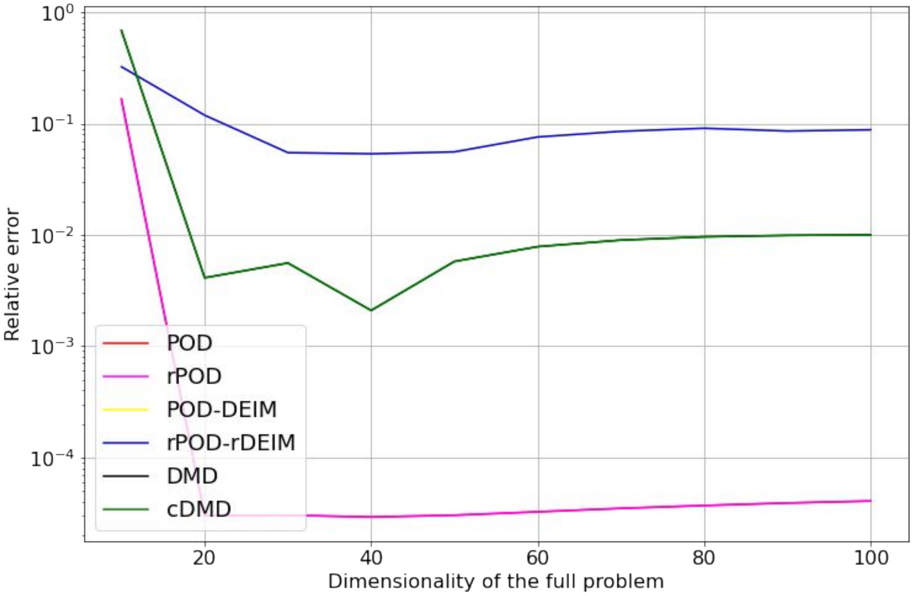


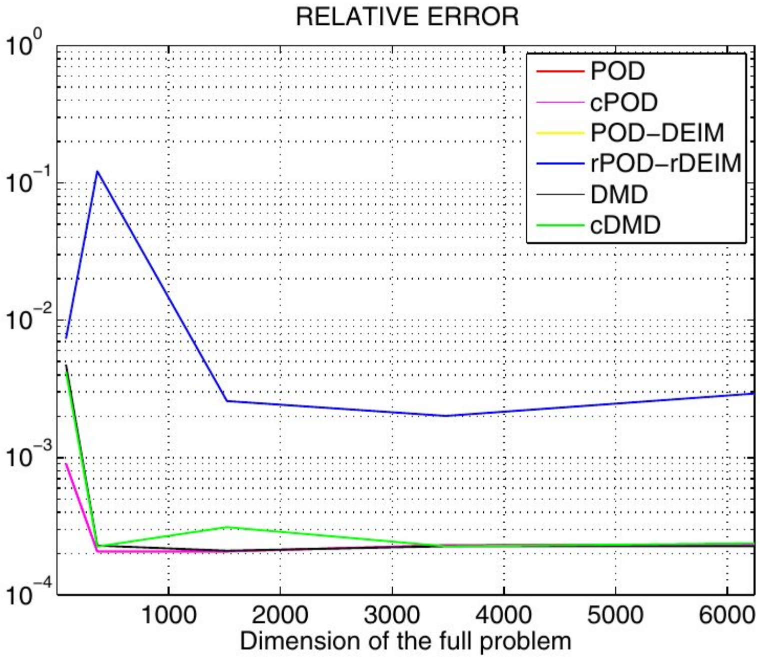
CPU Time vs. #Components



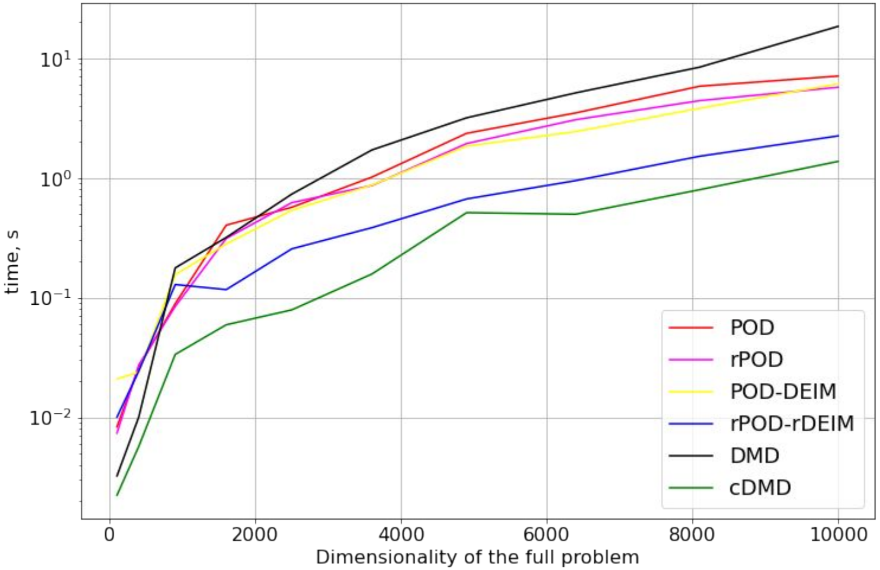


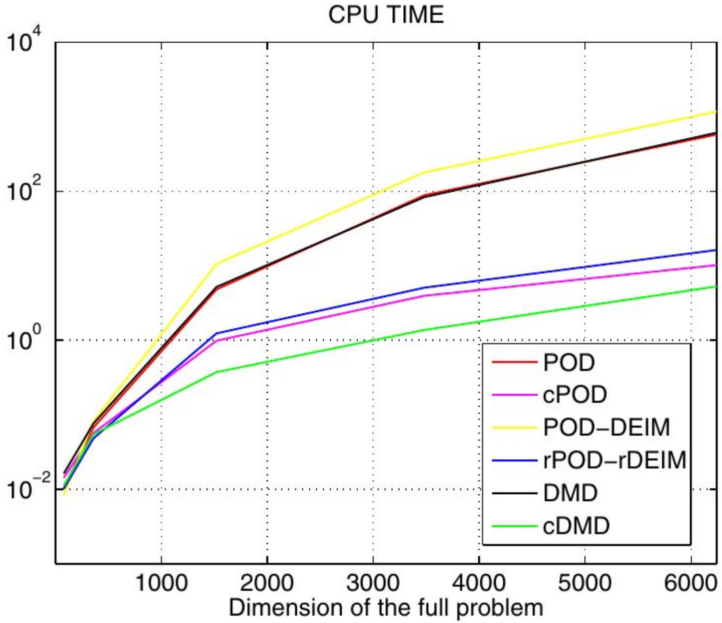
Error vs. Dimensionality of the full problem





CPU Time vs. Dimensionality of the full problem





Conclusions

- The randomized SVD calculation allows to significantly reduce the computational cost of the offline stage of reduced order modelling approaches.
- The most significant speed up is achieved for DMD and POD-DEIM algorithm.
- The speedup increases as the dimensionality of the problem increase.

References

1. Alla, A., & Kutz, J. N. (2019). Randomized model order reduction. *Advances in Computational Mathematics*, *45*(3), 1251-1271. (main source)

2. Rowley, C. W. (2005). Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, *15*(03), 997-1013.

3. Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, *656*, 5-28.

4. Chaturantabut, Saifon, and Danny C. Sorensen. “Nonlinear Model Reduction via Discrete Empirical Interpolation.” *SIAM Journal on Scientific Computing* 32, no. 5 (January 2010): 2737–64. https://doi.org/10.1137/090766498.