Model order reduction methods for PDEs and beyond

Sergey Matveev

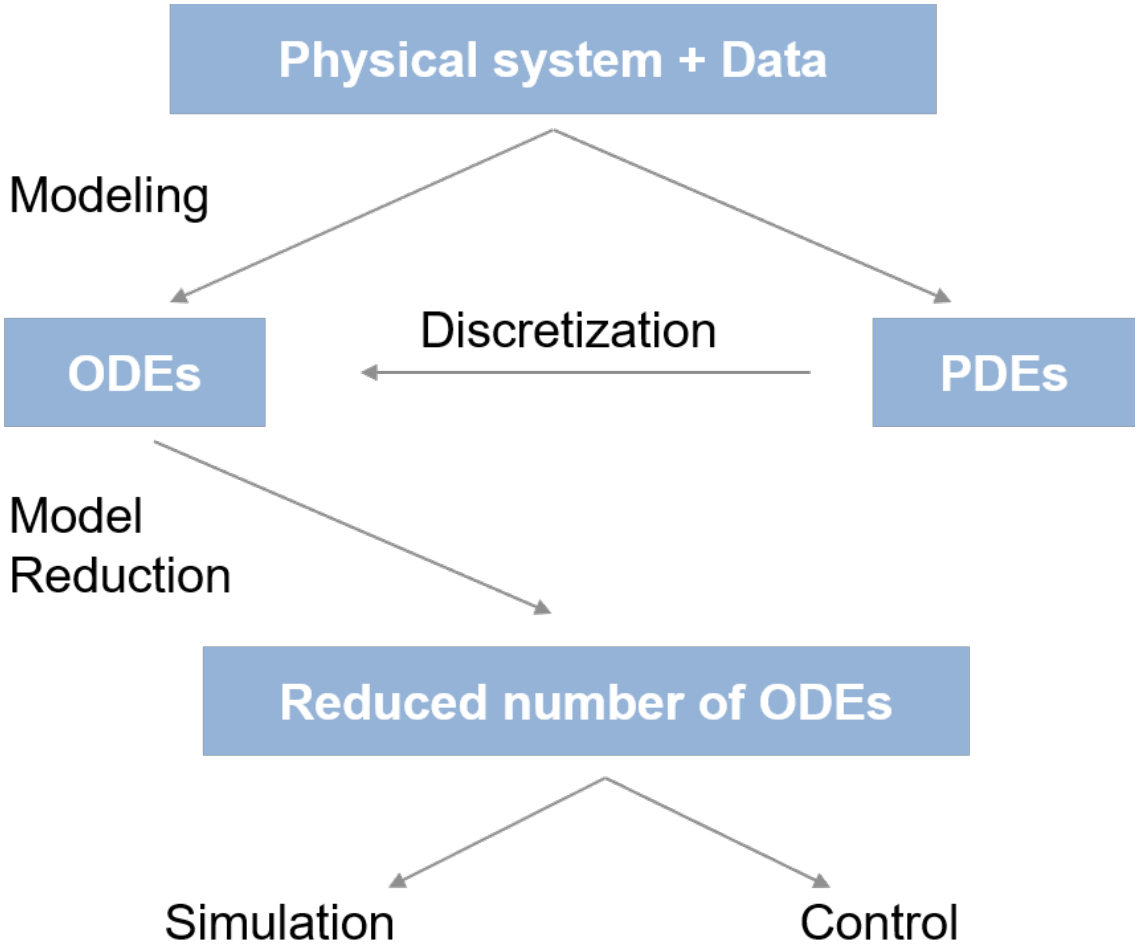
Lomonosov MSU and INM RAS

November 22, 2023

I am grateful to prof. Andrey S. Krylov for invitation and my colleagues:

- Eugene Tyrtyshnikov (POD, cross approximation)
- Ivan Timokhin (POD, cross approximation)
- Alexander Smirnov (POD, DMD)
- Dima Lukashevich (DMD)
- Sergey Petrov (sharing POD-DEIM results)

General diagram about model order reduction methods (see Wiki)



Software implementations:

- pyMOR (very general package) `https://pymor.org/`
- pyDMD `https://pypi.org/project/pydmd/`

Proper orthodongal decomposition

Assume that $u(x,t)$ is a solution of some non-stationary or parametric PDE/ODE, e.g.

$$\frac{\partial u}{\partial t} = A(u) + b(x,t), \text{ in some } \Omega \quad \text{(e.g. } A = \Delta \text{, and also the b.c.)}$$

We seek to get a decomposition with sepatated variables t and x:

$$u(x,t) = \sum_{k=1}^{\infty} a_k(t)\phi_k(x) \approx \sum_{k=1}^{N_{modes}} a_k(t)\phi_k(x).$$

In order to get it one needs some solver or generator of the discretized snapshots which have to be composed into a matrix

$$U = \begin{bmatrix} u(x_1,t_1) & \dots & u(x_n,t_1) \\ \dots & \dots & \dots \\ u(x_1,t_p) & \dots & u(x_n,t_p) \end{bmatrix}$$

Proper orthodongal decomposition

As soon as we get the snapshot martix

$$U = \begin{bmatrix} u(x_1, t_1) & \dots & u(x_n, t_1) \\ \dots & \dots & \dots \\ u(x_1, t_p) & \dots & u(x_n, t_p) \end{bmatrix}$$

One can compute a covariance matrix

$$C = \frac{1}{p-1} U^\top U$$

and apply the eigendecomposition for it. Otherwise one can extract the leading modes of the truncated SVD for the snapshot matrix U.
The extracted eigenvectors (or singular vectors) are orthogonal.
We can interpret t as separate parameter for modelling (see the example further).

Example 1: stationary system

1

Mathematical modeling of porous material

Volume (sometimes also time) averaged flow equations

- Darcy Law

$$\boldsymbol{u} = \frac{1}{\mu}\boldsymbol{K}\nabla p \qquad \mathrm{Re}_p = \frac{\rho_f u d_p}{\mu_f} < 1$$

- Forchheimer's equation (for isotropic porous medium)

$$\nabla p = -\frac{\beta\mu_f(1-\gamma)^2}{d_p^2\gamma^3}\boldsymbol{u} - \frac{\alpha\rho_f(1-\gamma)^2}{d_p^2\gamma^3}|\boldsymbol{u}|\boldsymbol{u}$$

- Hsu and Cheng, 1990, Vafai and Kim, 1990 refinement

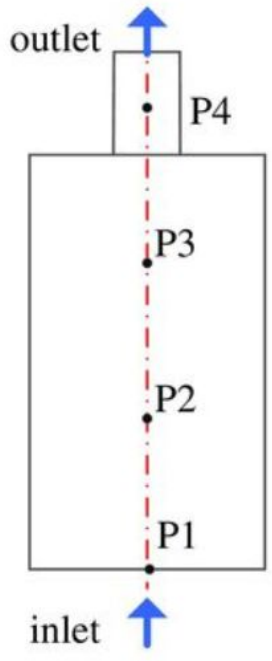
$$\rho_f\left[\frac{1}{\gamma}\frac{\partial \boldsymbol{u}}{\partial \tau} + \frac{1}{\gamma^2}(\boldsymbol{u}\cdot\nabla\boldsymbol{u})\right] = -\nabla p + \mu_e\nabla^2\boldsymbol{u} - \frac{\beta\mu_f(1-\gamma)^2}{d_p^2\gamma^3}\boldsymbol{u} - \frac{\alpha\rho_f(1-\gamma)^2}{d_p^2\gamma^3}|\boldsymbol{u}|\boldsymbol{u}$$

[1] slides from *Application of the Proper Orthogonal Decomposition for Bayesian Estimation of flow parameters in porous medium Zbigniew Buliński, Helcio R.B. Orlande*

Example 1: mathematical modelling of porous material

Problem formulation

$$\rho_f\left[\frac{1}{\gamma}\frac{\partial \boldsymbol{u}}{\partial \tau}+\frac{1}{\gamma^2}(\boldsymbol{u}\cdot\nabla\boldsymbol{u})\right]=-\nabla p+\mu_e\nabla^2\boldsymbol{u}-\frac{\beta\mu_f(1-\gamma)^2}{d_p^2\gamma^3}\boldsymbol{u}-\frac{\alpha\rho_f(1-\gamma)^2}{d_p^2\gamma^3}|\boldsymbol{u}|\boldsymbol{u}$$



Point	Distance from the reactor bottom, m
P1	0.00
P2	0.10
P3	0.20
P4	0.30

Vector of unknown parameters

$$\boldsymbol{K}=\{d_p,\gamma\}$$

Measured data

$$\boldsymbol{X}^*=\{\Delta p_1=p_1-p_4,\Delta p_2=p_2-p_4,\Delta p_3=p_3-p_4\}$$

Example 1: mathematical modelling of porous material

Statistical inverse problem formulation

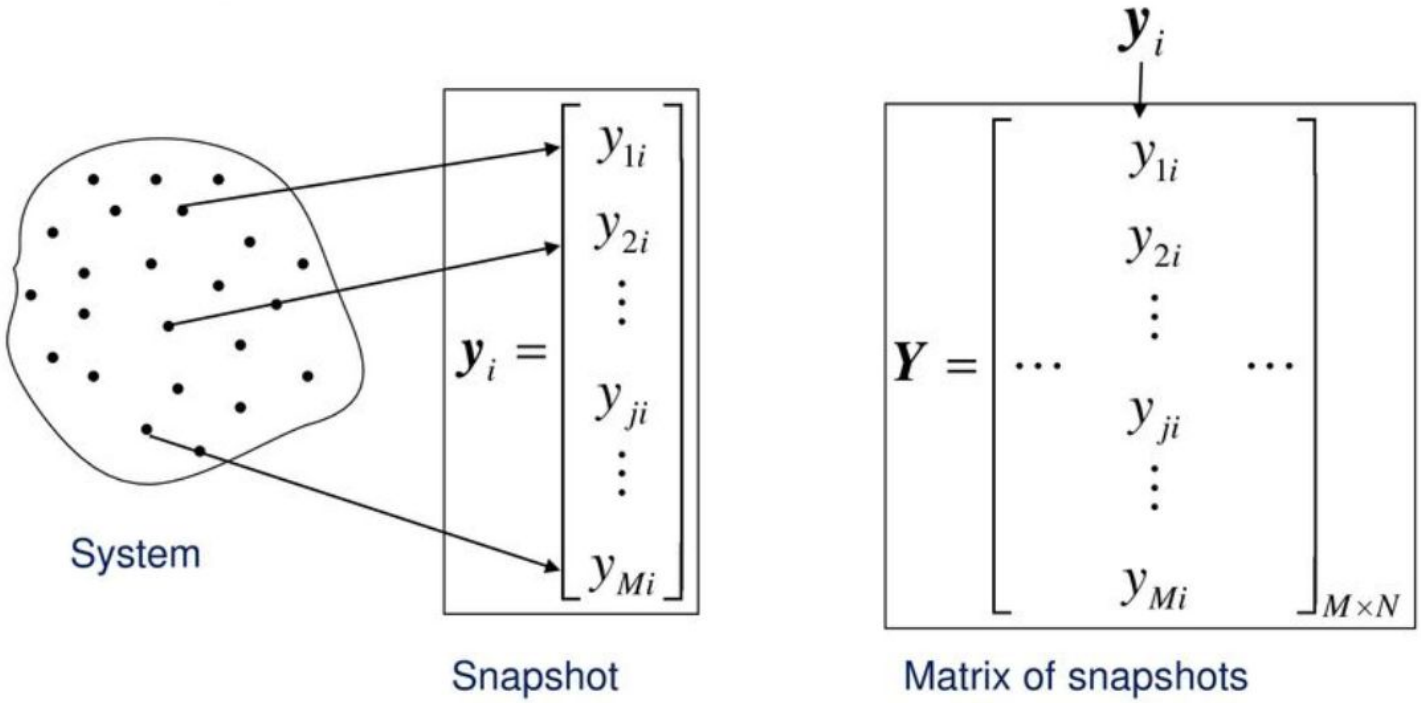
Main features of the Statistical inverse problem formulation

- All unknown quantities (estimated parameters) and input data (measurements) are treated as random variables,
- Knowledge on the realisation of all these quantities is given in the form of the probability distribution,
- Solution of the considered problem is given in a form of probability distribution of unknown quantities,

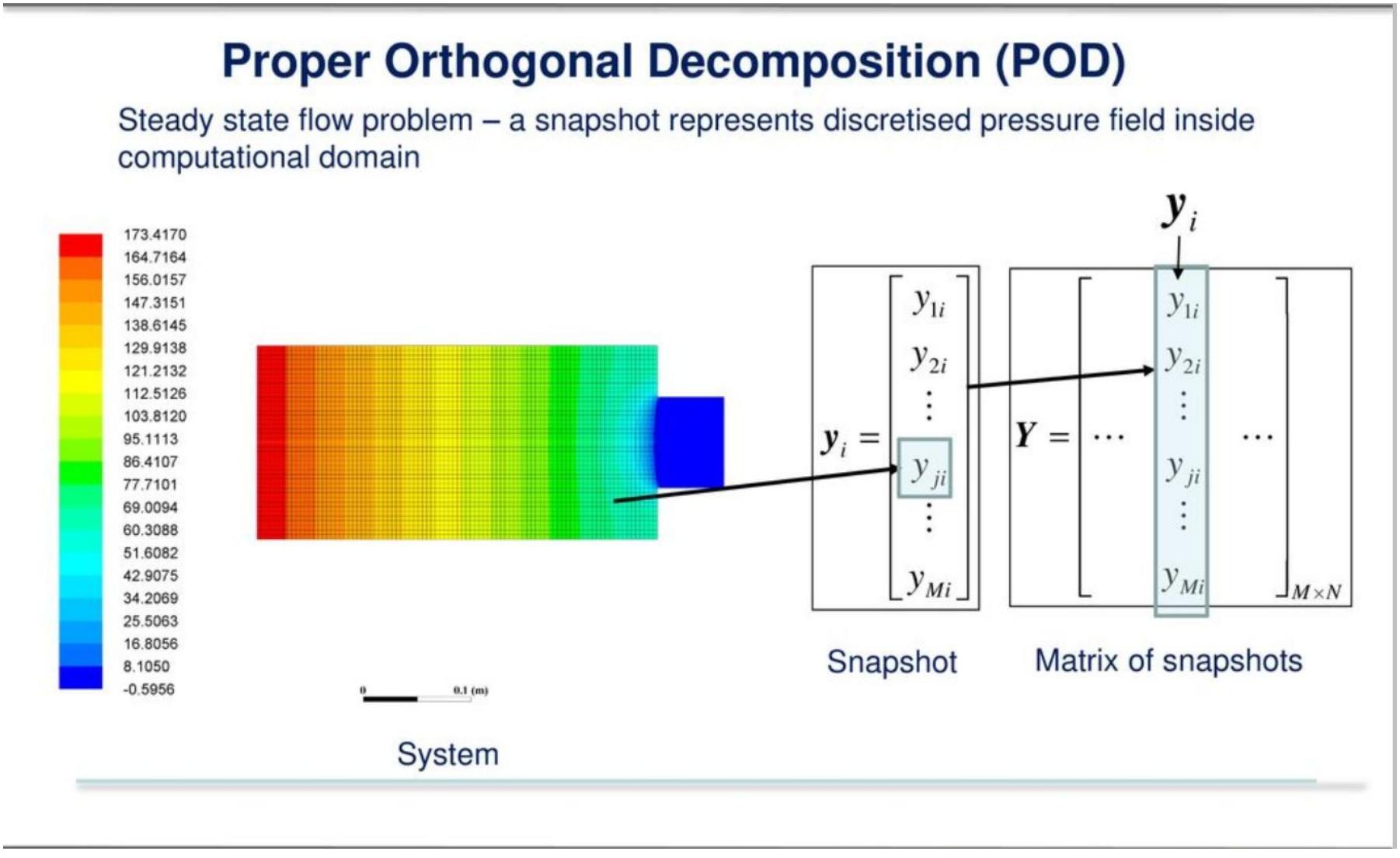
Example 1: mathematical modelling of porous material

Proper Orthogonal Decomposition (POD)

Single snapshot $\boldsymbol{y}_i$ is discretized response (image) of the field under investigation for given spatial coordinates and time



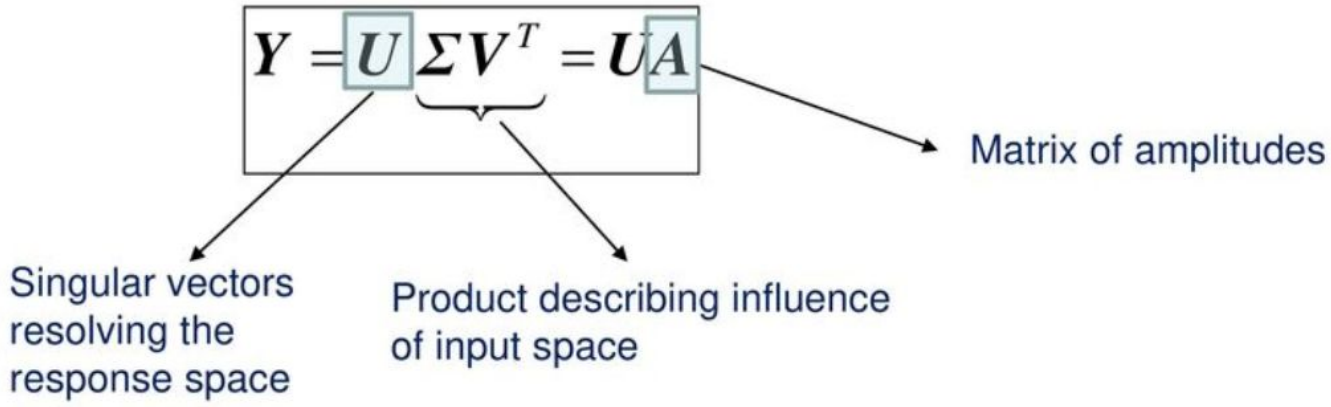
Example 1: mathematical modelling of porous material



Example 1: mathematical modelling of porous material

Proper Orthogonal Decomposition (POD)

Retrieve structure (anatomy) of the snapshot matrix by SVD



May be interpreted as a variables separation

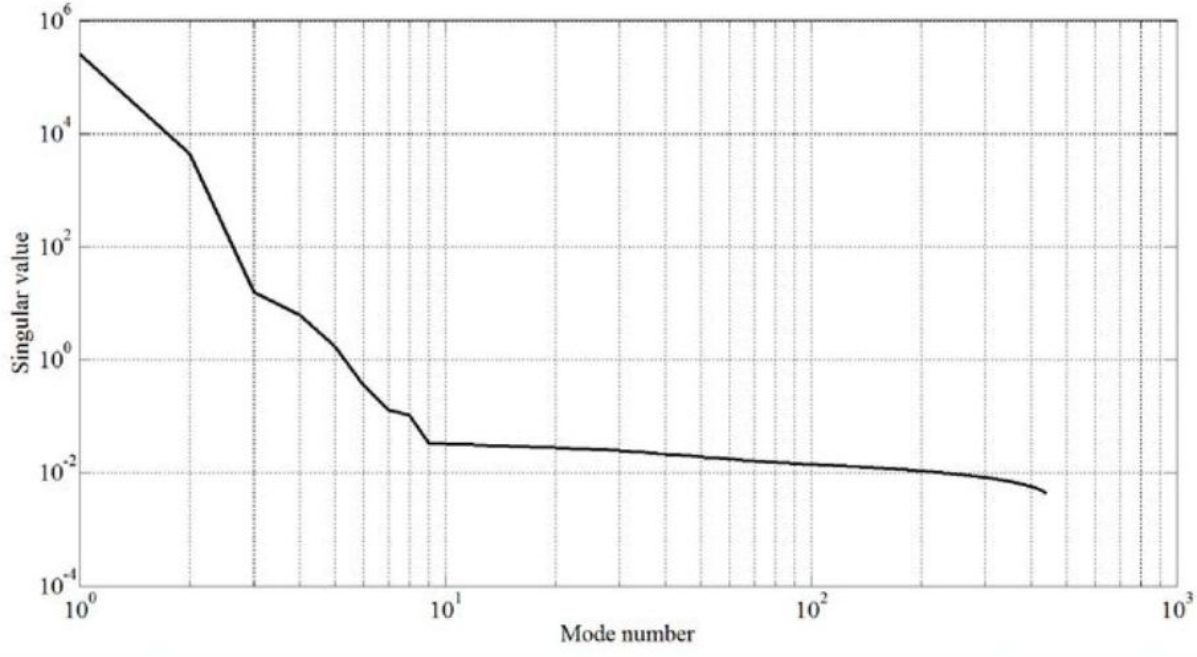
$$Y = U(x,t)A(K)$$

Example 1: mathematical modelling of porous material

Proper Orthogonal Decomposition (POD)

Truncate the input space basis

$$\boldsymbol{Y} \cong \overline{\boldsymbol{Y}} = \overline{\boldsymbol{U}}\,\overline{\overline{\Sigma}}\,\overline{\boldsymbol{V}}^{T} = \overline{\boldsymbol{U}}(\boldsymbol{x},t)\,\overline{\boldsymbol{A}}(\boldsymbol{K})$$



Example 1: mathematical modelling of porous material

Proper Orthogonal Decomposition (POD)

Approximation of the amplitude matrix with a set of basis functions

$$\boldsymbol{Y} \cong \overline{\boldsymbol{Y}} = \overline{\boldsymbol{U}}\,\overline{\boldsymbol{\Sigma}}\,\overline{\boldsymbol{V}}^{T} = \overline{\boldsymbol{U}}(\boldsymbol{x},t)\,\overline{\boldsymbol{A}}(\boldsymbol{K})$$

$$\overline{\boldsymbol{\Sigma}}\,\overline{\boldsymbol{V}}^{T} = \boldsymbol{U}^{T}\boldsymbol{Y} = \overline{\boldsymbol{A}}(\boldsymbol{K}) = \boldsymbol{B}\,\boldsymbol{\Gamma}(\boldsymbol{K})$$

Finally, response of the system for any set of input parameters $\boldsymbol{K}$ can be calculated with trained POD-RBF network

$$\boldsymbol{y}(\boldsymbol{K}) = \boldsymbol{U}\,\boldsymbol{B}\,\boldsymbol{g}\left(\left\|\boldsymbol{K}-\boldsymbol{K}_i\right\|\right)$$

$$\boldsymbol{g}\left(\left\|\boldsymbol{K}-\boldsymbol{K}_i\right\|\right) = \begin{bmatrix} \vdots \\ \varphi\left(\left\|\boldsymbol{K}-\boldsymbol{K}_i\right\|\right) \\ \vdots \end{bmatrix}_{N\times 1}$$

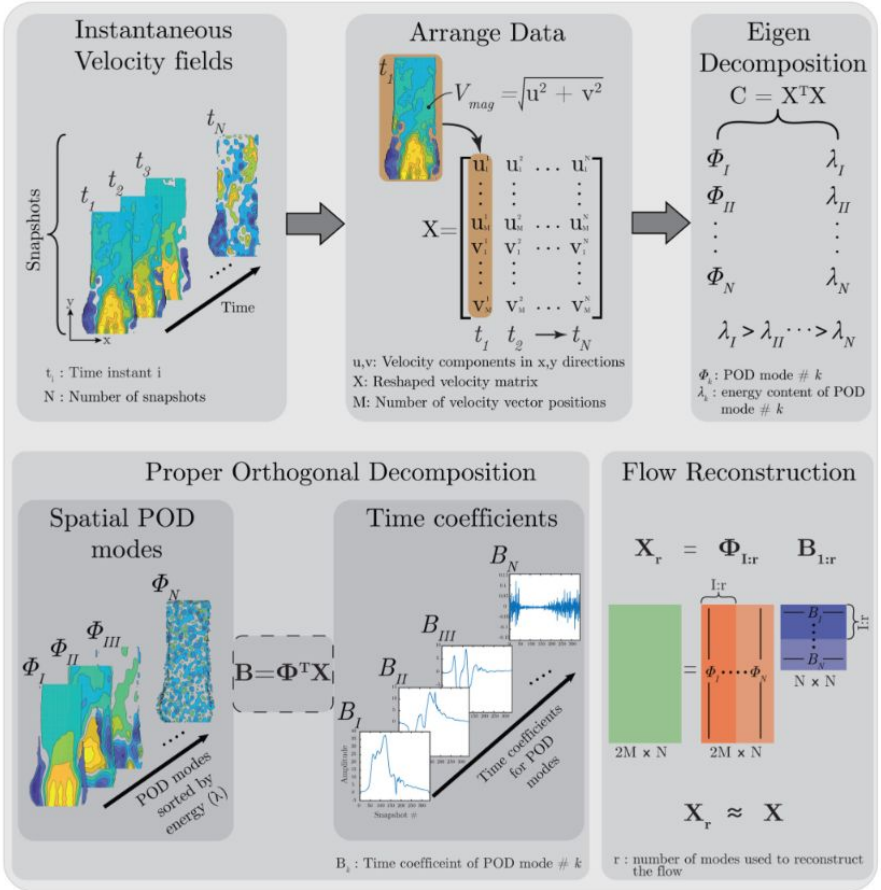
$$\varphi\left(\left\|\boldsymbol{K}-\boldsymbol{K}_i\right\|\right) = \frac{1}{\sqrt{\left\|\boldsymbol{K}-\boldsymbol{K}_i\right\|^2 + c^2}}$$

$$\varphi\left(\left\|\boldsymbol{K}-\boldsymbol{K}_i\right\|\right) = \left\|\boldsymbol{K}-\boldsymbol{K}_i\right\|^2 \ln\left(\left\|\boldsymbol{K}-\boldsymbol{K}_i\right\|\right)$$

$$\varphi\left(\left\|\boldsymbol{K}-\boldsymbol{K}_i\right\|\right) = \exp\left(-\frac{\left\|\boldsymbol{K}-\boldsymbol{K}_i\right\|^2}{c^2}\right)$$

Final scheme for the flow-fields

2



[2]Darwish A. et al. Proper orthogonal decomposition analysis of the flow downstream of a dysfunctional bileaflet mechanical aortic valve //Cardiovascular Engineering and Technology. – 2021. – Vol. 12. – pp. 286-299.

Example 2: POD-DEIM, motivation and context

- Design optimization for electrical machines may take a long time, because it requires solving a PDE a large number of times; [3]
- Although efficient numerical solvers for the considered 2D-magnetostatic equation have already been developed, such a solver is to be called thousands of times to solve the same equation for different combinations of the parameter values, which may include:
  - Modulus of the stator current I
  - Phase angle between rotor and stator currents γ
  - Mechanical rotor angle θ

[3] Petrov S. Model order reduction algorithms in the design of electric machines //Large-Scale Scientific Computing: 12th International Conference, LSSC 2019, Sozopol, Bulgaria, June 10–14, 2019, Revised Selected Papers 12. – Springer International Publishing, 2020. – pp. 140-147.

Example 2: POD-DEIM, considered equation

- 2D magnetostatic equation over magnetic potential ϕ

$$\nabla(\nu(||\nabla\phi(x,y)||;\theta)\nabla\phi(x,y)) = \hat{f}(x,y;I,\gamma),(x,y)\in\Omega;$$

- Nonlinear function ν represents the material properties (thus it depends on mechanical angle);
- Problem region Ω has a complicated structure, involving 'rotor' and 'stator' parts that can move relative to each other;
- The problem has both Dirichlet and periodic boundary conditions.

Example 2: POD-DEIM, magnetostatic equation region

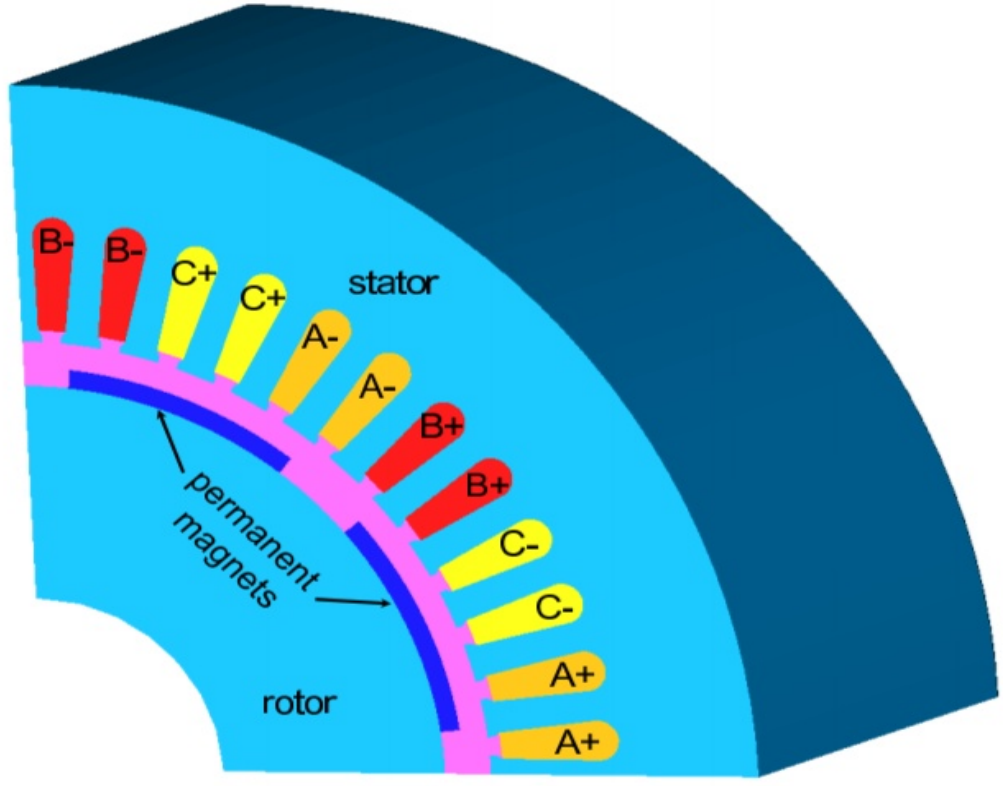


Figure 2. Permanent magnets synchronous machine

Example 2: POD-DEIM, model problem

- Model problem region is simplified from a rectangle in polar coordinates to a rectangle in Cartesian coordinates;
- Nonlinear material properties are changed to analytic expressions (instead of the original representation in the form of splines over measurement data);

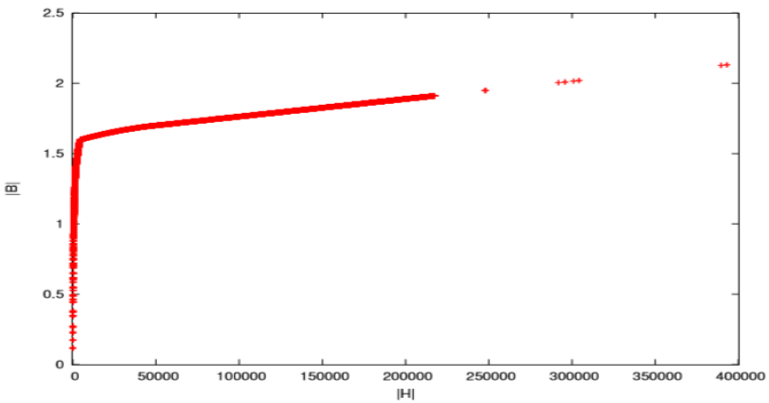


Figure 1: B-H-Curve spline example from literature

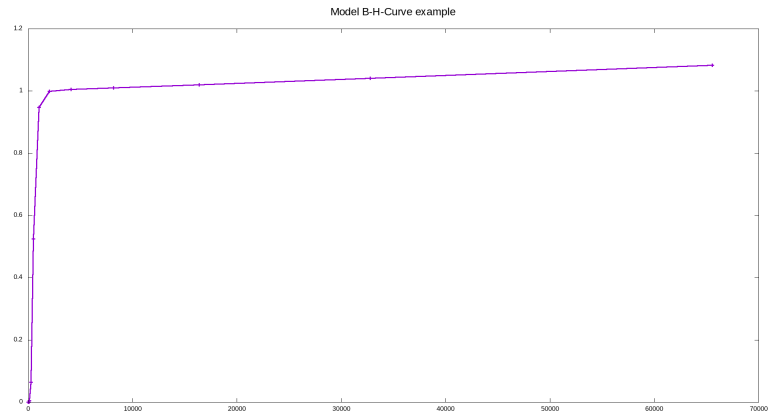


Figure 2: The applied B-H-Curve with analytical expression

- The 'rotor' and 'stator' movement is modelled via a periodic shift of the right-hand side.

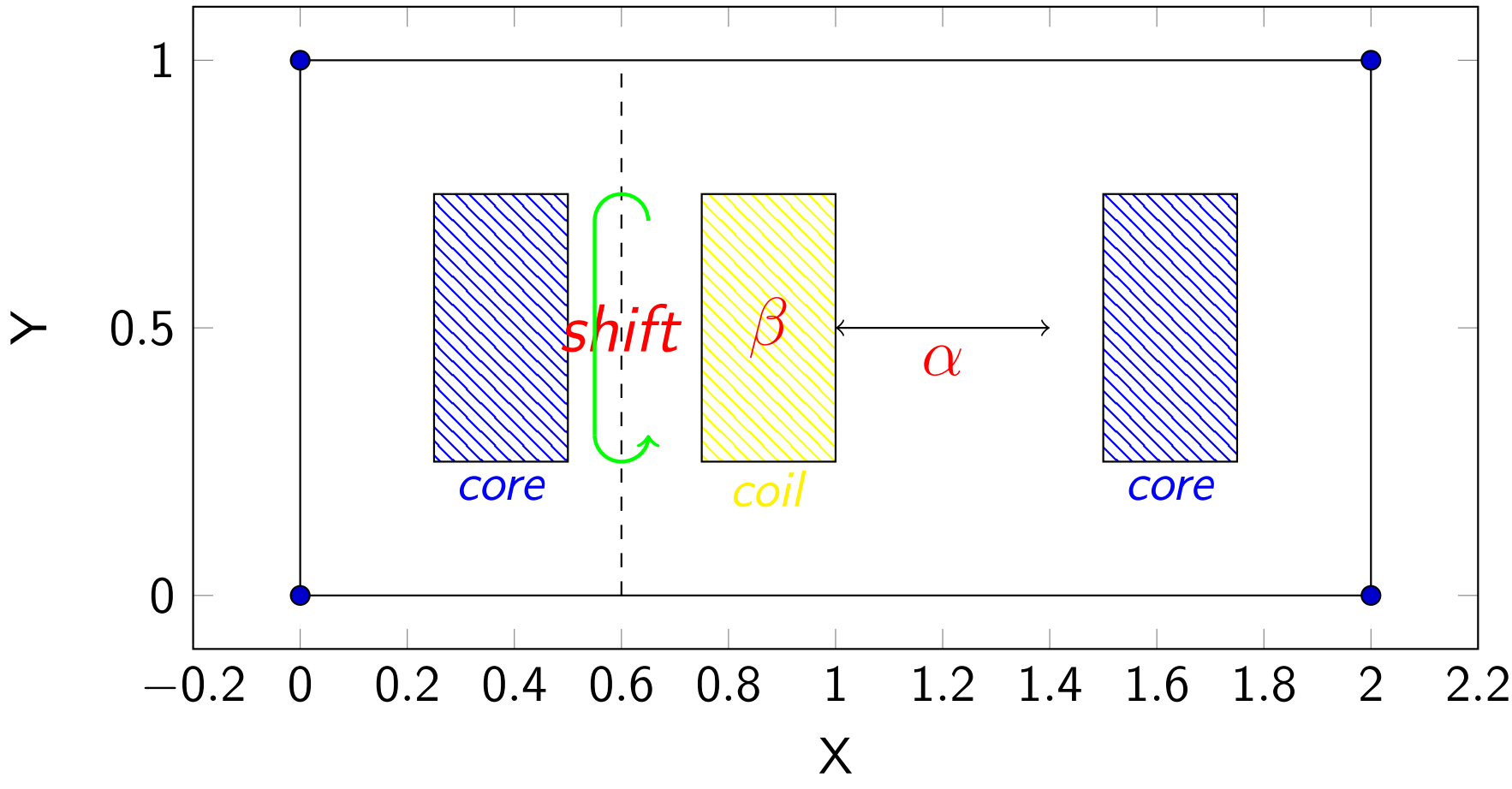
Example 2: POD-DEIM, model problem

In the model problem, we keep the following properties:

- Original differential equation form
- Boundary condition types
- Cyclic shift parameter
- Right-hand side scaling parameter
- Presence of physical objects made of different materials
- Structure of the nonlinear material properties

Example 2: POD-DEIM, model problem parameters

The model problem has three parameters marked in red: right side value at the coil β, horizontal shift of the coil α, and vertical cyclic shift of the right part of the problem region.



Example 2: POD-DEIM, model problem numerical solver

$$\nabla(\nu(||\nabla\phi(x,y)||)\nabla\phi(x,y)) = \hat{f}(x,y), \quad (x,y) \in \Omega$$

The original equation is rewritten in a more convenient form:

$$\vec{z}(x,y) := \nu(||\nabla\phi(x,y)||)\nabla\phi(x,y)$$

$$||\vec{z}|| = \nu(||\nabla\phi||)||\nabla\phi|| = f^{-1}(||\nabla\phi||), f(||\vec{z}||) = ||\nabla\phi||,$$

where f is the B-H-Curve function.
For $g(s) := \frac{f(s)}{s}$

$$\begin{cases} \vec{z}g(||\vec{z}||) = \nabla\phi \\ div(\vec{z}) = \hat{f}, \end{cases}$$

Example 2: POD-DEIM, model problem numerical solver

- Regular quadrilateral grid is used;
- In the discretization the potential values and fluxes are considered as independent variables;
- Control volume mixed finite elements scheme is used;
- The potential is approximated by piecewise-constant functions, the flux values are approximated by piecewise-linear functions.

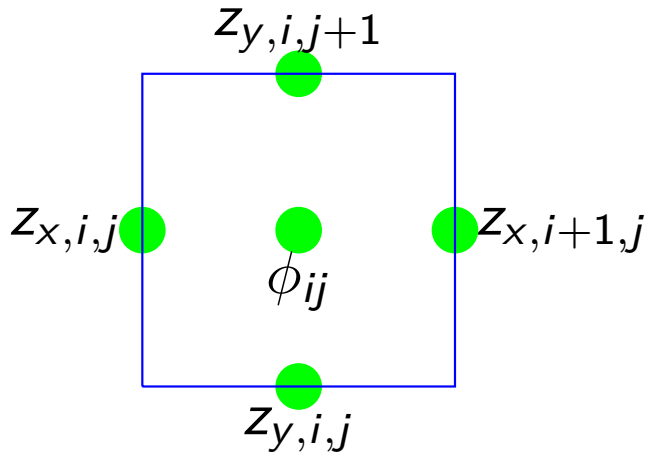


Figure 3: Schematic representation of grid square i, j unknowns

Example 2: POD-DEIM

Assume that the nonlinear numerical scheme used to solve the considered parametric PDE is organised in the following form

$$Ay_\mu + F_\mu(y_\mu) = 0,$$

where A is a linear operator, F_μ – nonlinear part. Both $\mathbb{R}^M \to \mathbb{R}^M$. Denoting by y_μ a solution corresponding to parameter μ, POD method requires the following assumption

$$\text{numrank}_\varepsilon\left([y_{\mu_1}, \dots y_{\mu_k}]\right) \approx \text{const}, \forall k.$$

Apply SVD $[y_{\mu_1}, \dots y_{\mu_k}] \approx V\Sigma Q_V^\top$ The original discrete system then can be projected onto V

$$V^\top AVx_\mu + V^\top PF_\mu(Vx_\mu) = 0$$

Intuitively, the projector P is defined in such a way that it only uses a small fraction of elements of a given vector x to compute Px (so it is sparse)

Example 2: POD-DEIM

Formally $P : \mathbb{R}^M \to \mathbb{R}^M$ is a linear projector with following structure:

$$P = U(S^\top U)^{-1} S^\top,$$

where where U is a matrix of size $M \times r_i$ (r_i is 'interpolation' rank, r_a is 'approximation' rank, $r_i > r_a$), and S is a matrix with only one nonzero element in each column

$$S = [e_{i_1} \quad e_{i_2} \quad \dots \quad e_{i_k}].$$

Such a projector P is uniquely defined by its image U and a set of indices $I_{int} = i_1, i_2, \dots, i_k$. The structure of P implies that vector y elements $y_{i_1} \dots y_{i_k}$, 'selected' by columns of S, are enough to compute Py, and only a small fraction of elements of $F(Vx)$ are used to compute $V^\top PF(Vx)$.

Example 2: POD-DEIM

PDE numerical scheme equations are 'local': for each j, the value of $F(y)_j$ depends only on a small number of elements of y. $C(j)$ defines the set of indices of y defining the value $F(y)_j$.
Let $C(I_{int}) = \mathrm{U}_k\ C(i_k)$. Then $D_{C(I_{int})}$ is a diagonal matrix:

$$\begin{cases} (D_{C(I_{int})})_{jj} = 1, & C \in C(I_{int}) \\ (D_{C(I_{int})})_{jj} = 0, & C \notin C(I_{int}) \end{cases}$$

Then we have $V^\top PF(Vx) = V^\top PF(D_{C(I_{int})} Vx)$. A small fraction of elements of Vx is enough to compute $V^\top PF(Vx)$.
Hence, we get the final structure of the POD-DEIM system of reduced equations

$$V^\top AVx_\mu + V^\top U(S^\top U)^{-1} S^\top F_\mu(D_{C(I_{int})} Vx_\mu) = 0,$$

where $V^\top AV$ and $V^\top U(S^\top U)^{-1}$ are computed in advance from snapshots! This is offline stage.

Example 2: POD DEIM, results

	C-norm	L-2 norm
Values error	0.002642	0.000624
Flux error	0.0002374	0.000328

	Time
Full model solver	41.13 seconds
Reduced solver preparations	0.0533 seconds
Reduced solver iterations	0.0154 seconds

4

[4] Petrov S. Model order reduction algorithms in the design of electric machines //Large-Scale Scientific Computing: 12th International Conference, LSSC 2019, Sozopol, Bulgaria, June 10–14, 2019, Revised Selected Papers 12. – Springer International Publishing, 2020. – pp. 140-147.

Example 2: POD-DEIM, results

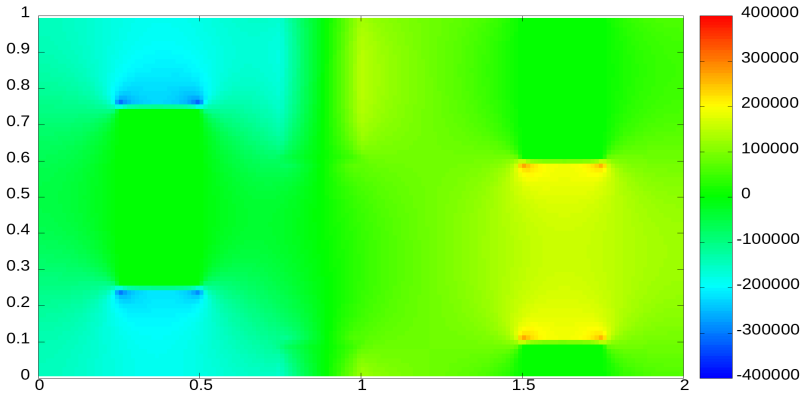


Figure 4: Test 2, true flux x

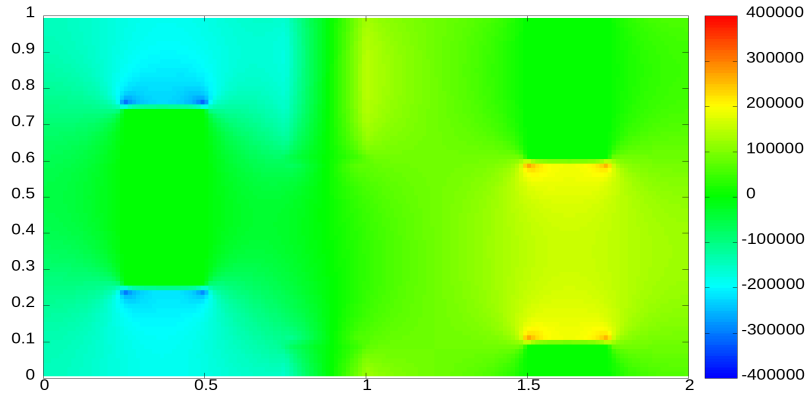


Figure 5: Test 2, reduction flux x

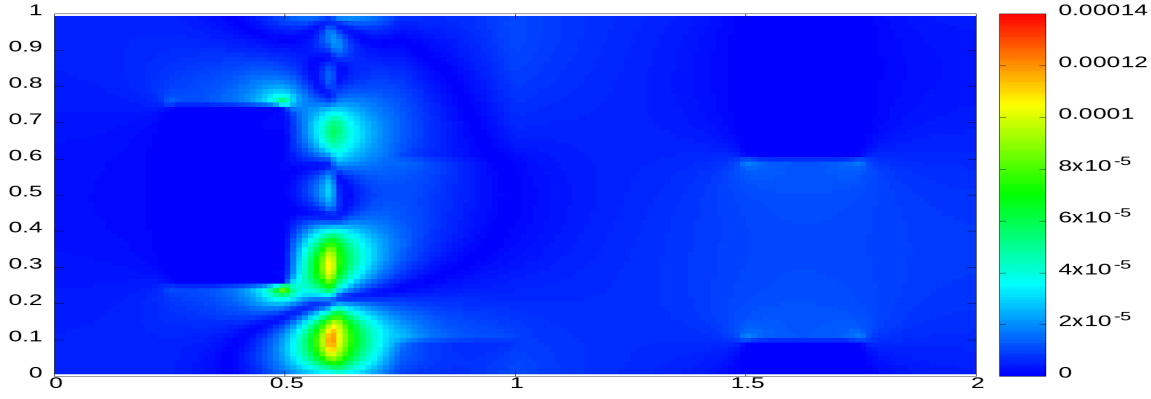


Figure 6: Test 2, flux x relative error

Example 3: POD for non-stationary systems

- In this example our aim is obtain model order reduction for non-stationary problems

$$\frac{df}{dt} = \mathcal{F}(f(t), t), \qquad f(t) \in \mathbb{R}^N$$

- The concrete problem is system of Smoluchowski kinetic equations

$$\frac{dn_k}{dt} = J_k + \frac{1}{2}\sum_{i+j=k} C_{i\,j} n_i n_j - n_k \sum_{j=1}^{N} C_{j\,k} n_j, \qquad k = \overline{1, N},$$

where $C_{i,j} = (i/j)^a + (j/i)^a$ and N is large

- Question 1: Does the low-dimesional basis exist for this class of problems?
- Question 2: If the basis exists then how one can construct and use it efficiently?
- Question 3: Can we vary techniques?

However, we start from answering question 2 and Proper Orthogonal Decomposition

Question 2: If the basis exists then how one can construct and use it efficiently?

Assume, an orthonormal basis exists as columns of $V \in \mathbb{R}^{N \times R}$, such that

$$\left\| n(t) - VV^T n(t) \right\| \ll \|n(t)\|,$$

Then

$$x(t) \equiv V^T n(t), \qquad x(t) \in \mathbb{R}^R$$

and one can recover the approximation of the solution as

$$n(t) \approx \tilde{n}(t) = Vx(t).$$

Question 2: If the basis exists then how one can construct and use it efficiently?

Inroduce a tensor $S \in \mathbb{R}^{N\times N\times N}$:

$$S_{i\,j\,k} = \frac{1}{2}\left(\delta_{i+j,k} - \delta_{i,k} - \delta_{j,k}\right) C_{i\,j},$$

Then original system becomes

$$\frac{dn_k}{dt} = J_k + \sum_{i,j=1}^{N} S_{i\,j\,k} n_i n_j.$$

For the reduced solution we get

$$\frac{d}{dt}\sum_{\alpha=1}^{R} V_{k\,\alpha} x_\alpha(t) \approx J_k + \sum_{i,j=1}^{N} S_{i\,j\,k} \times \left(\sum_{\beta=1}^{R} V_{i\,\beta} x_\beta(t)\right) \times \left(\sum_{\gamma=1}^{R} V_{j\,\gamma} x_\gamma(t)\right).$$

Question 2: If the basis exists then how one can construct and use it efficiently?

Finally, after some technical work and notations

$$\frac{d}{dt}\tilde{x}_\alpha = \tilde{J}_\alpha + \sum_{\beta,\gamma=1}^{R} \tilde{S}_{\alpha\,\beta\,\gamma}\tilde{x}_\beta\tilde{x}_\gamma, \qquad \alpha = 1, 2, \ldots, R$$

$$\tilde{x}(0) = x(0) = V^T n(0).$$

Allowing to switch the complexity from $O(NR_k \log N)$ to $O(R^3)$. It is good if $R \ll N$.
Recovery of the original solution

$$n(t) \approx \tilde{n}(t) = V\tilde{x}(t).$$

Return to Question 1:
Does the low-dimesional basis exist for this class of problems?

Russ. J. Numer. Anal. Math. Modelling 2022; 37(1):1–10

Question 1: Does the low-dimesional basis exist for this class of problems?

We follow the method of snapshots

Step 1 Set $k \leftarrow 1$, $V_0 = 0 \in \mathbb{R}^{N \times 0}$.

Step 2 Calculate $\hat{V}_k$ via the method of snapshots as an approximate reduction basis for the time span $[(k-1)\tau, k\tau]$.

Step 3 If $\left\| (I - V_{k-1} V_{k-1}^T) \hat{V}_k \right\|_2 \leq \varepsilon$, set $V = V_{k-1}$ and exit.

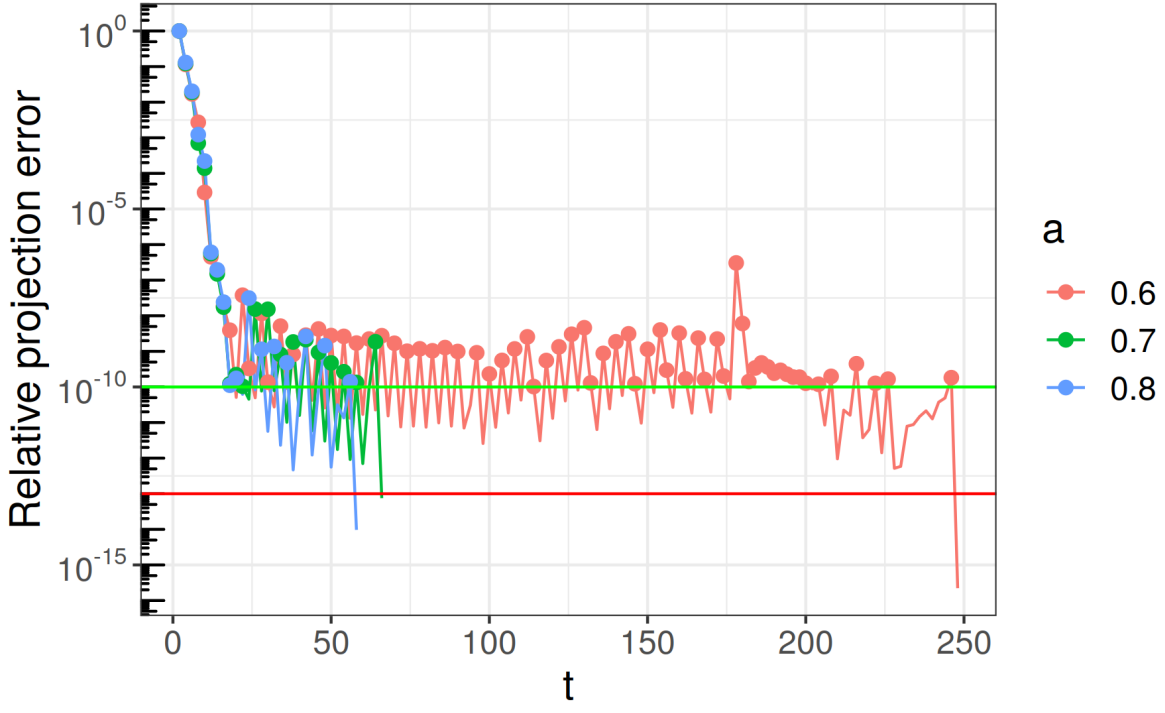
Step 4 Otherwise, set $V_k = V_{k-1} \oplus_\delta \hat{V}_k$.

Step 5 Set $k \leftarrow k + 1$ and repeat from step 2.

This ideology works very well for problems with cyclic solutions and not very useful in case of pure aggregation equations (no sources/no sinks).

Question 1: Does the low-dimesional basis exist for this class of problems?

Yes! And we can find it with method of snapshots!

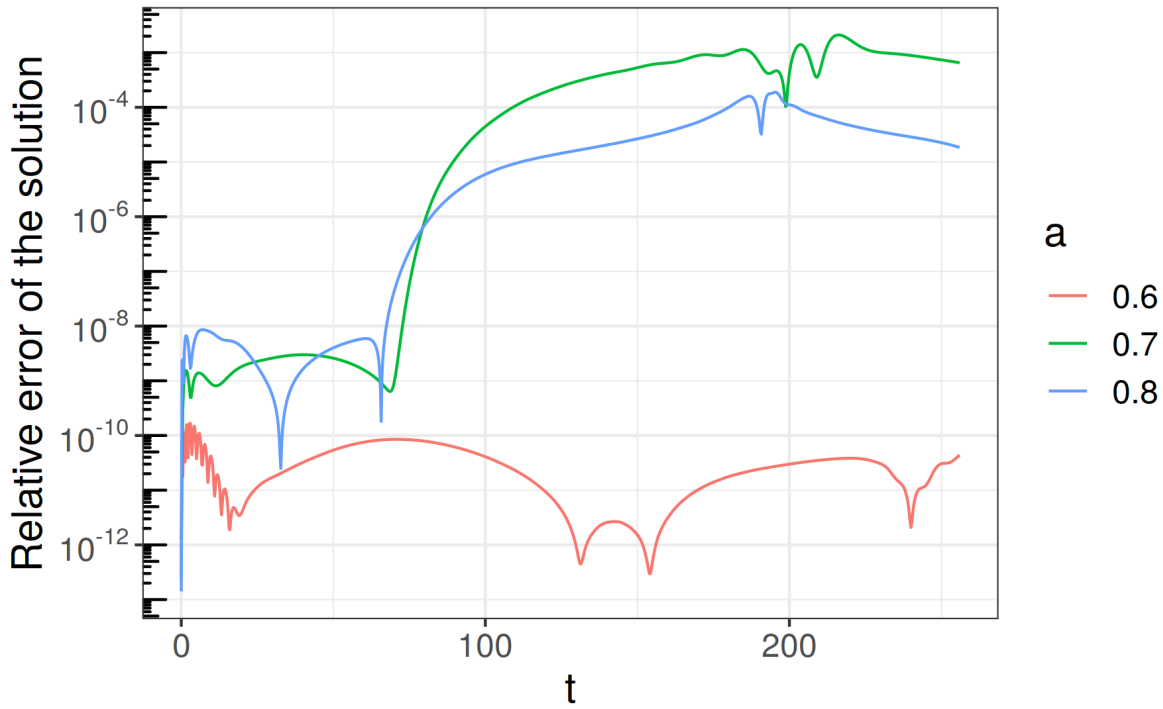


Accumulation of the basis for particular experiments

Russ. J. Numer. Anal. Math. Modelling 2022; 37(1):1–10

Question 1: Does the low-dimesional basis exist for this class of problems?

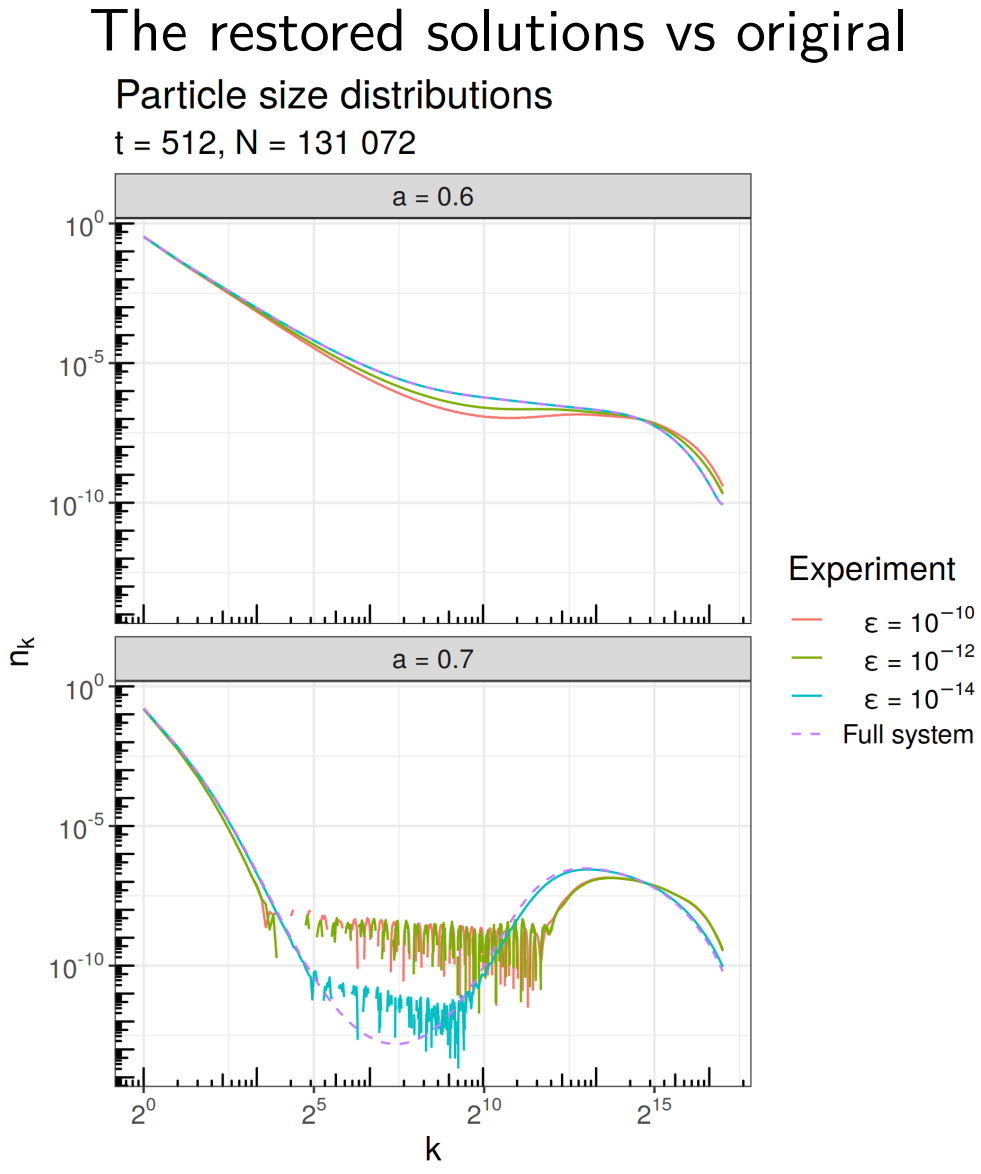
Yes! And we can find it with method of snapshots!



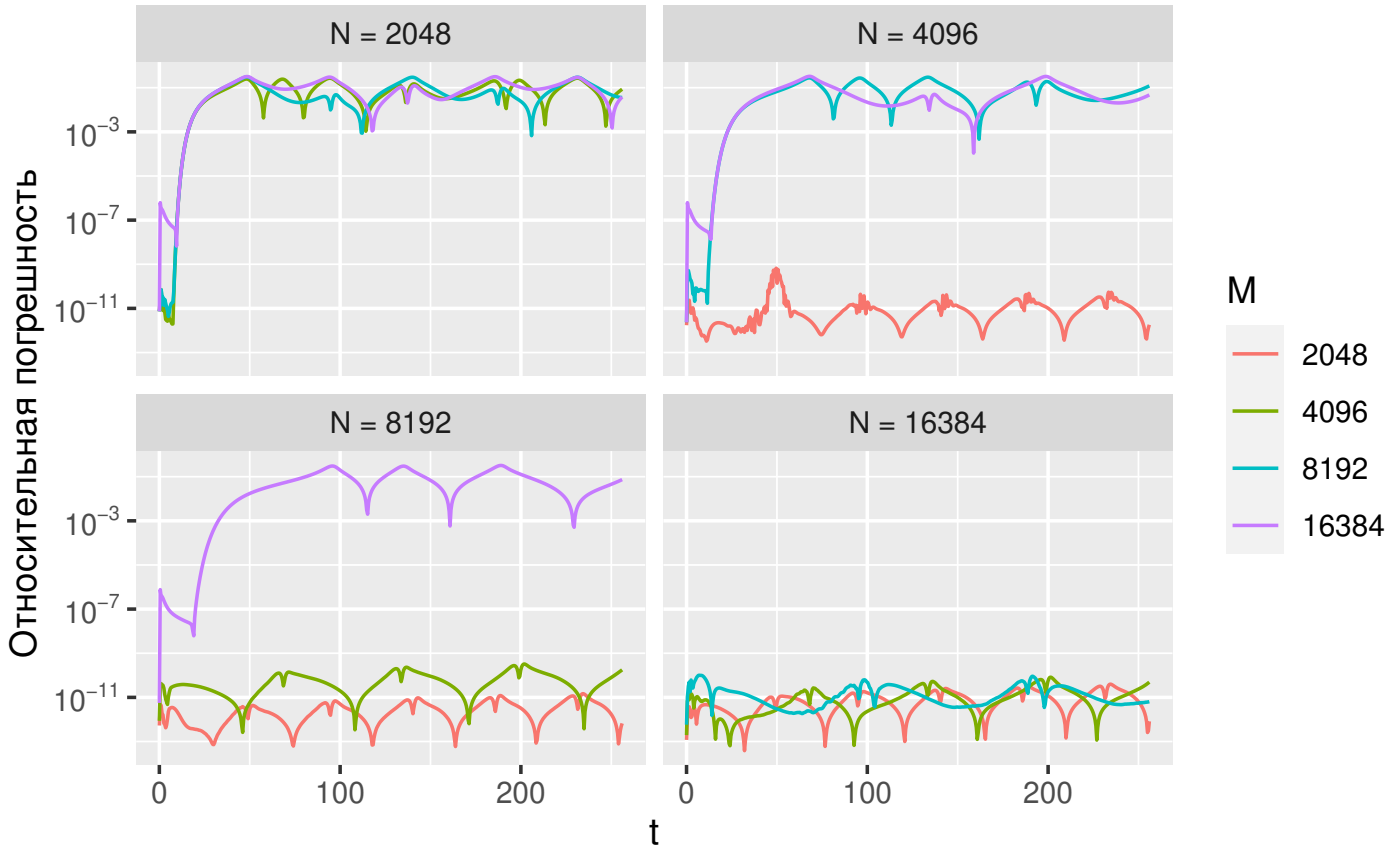
The error changes with switching from the "interpolation" to "extrapolation" mode.

The final speedup of calculations is significant (5-20 times depending on model parameters).

Question 1: Does the low-dimesional basis exist for this class of problems?



Interesting observation: reduced bases can be “truncate” for smaller problems



Truncation of “large” bases allows to approximate smaller systems.
But their solutions are not close to larger systems!
Thus, bases can be re-used.

Comp. Math. and Math. Phys., 2022, Vol. 62, No. 4

Question 3-1: Can we vary techniques? Apply DMD in case of pure coagulation!

- Assume we have a collection of equidistant (in time) snapshots:

$$V_1^n = \{v_1, v_2, ..., v_k\}, \qquad v_i \in \mathbb{R}^N$$

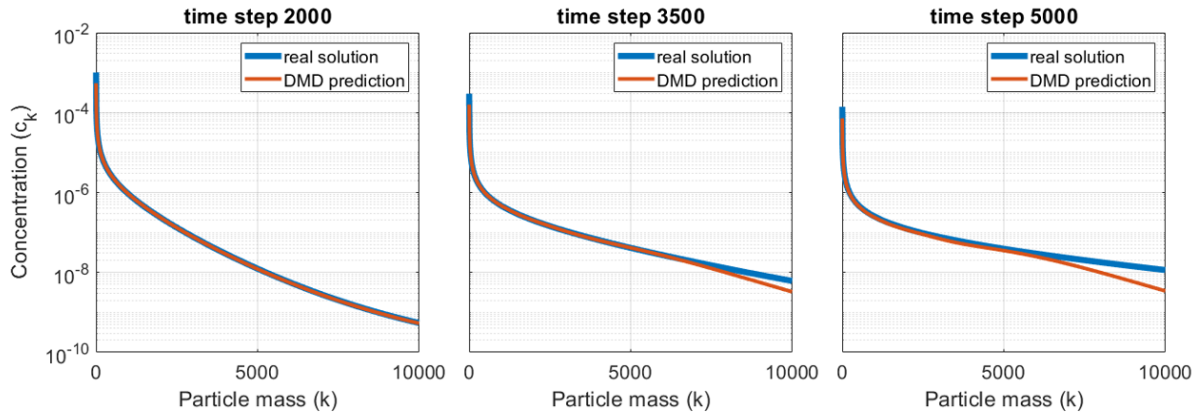
.

- We seek to restore the evolution operator:

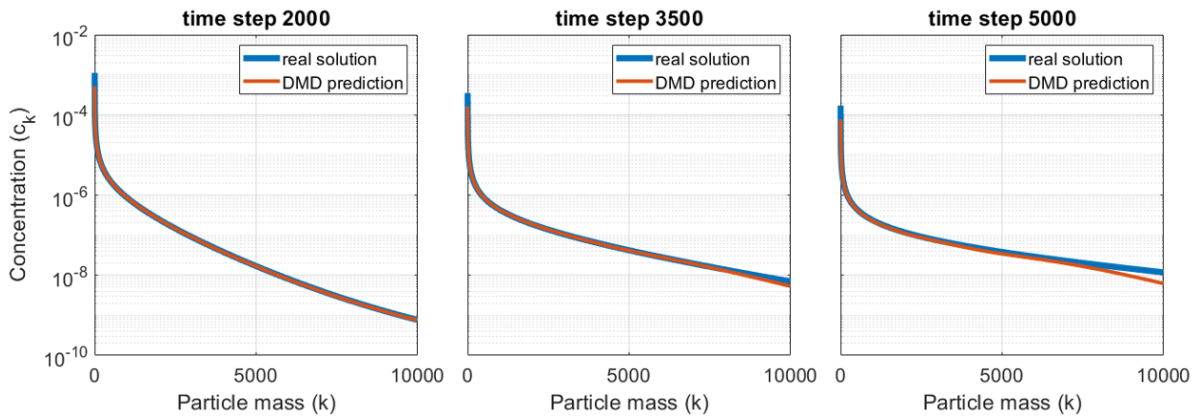
$$v_{i+1} = Av_i \text{ for } i = 1, \ldots, k-1.$$

- Extract small number R of leading dynamic modes for this linear operator (e.g. apply SVD).
- If their number is modest we can extrapolate the time-evlotion faster
- We use the `pyDMD` package.
- Hard if we deal with non-homogeneous aggregation equations (e.g. with advection).

Numerical results



Extrapolation results for binary coagulation.



Extrapolation results for ternary coagulation.

In both cases we use $N = 1000$ snapshots and extract $R = 25$ major dynamic modes. Initial number of ODEs was set 10^4. Works quite well for problem without sink-source terms.

Another interesting applications of DMD

- Shadrin D. et al. System identification-soilless growth of tomatoes //2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). – IEEE, 2019. – pp. 1-6.
- Stasenko N. et al. Dynamic Mode Decomposition and Deep Learning for Postharvest Decay Prediction in Apples //IEEE Transactions on Instrumentation and Measurement. – 2023.

Another interesting applications of DMD-1

Aim: restore the equations from data and apply the control.

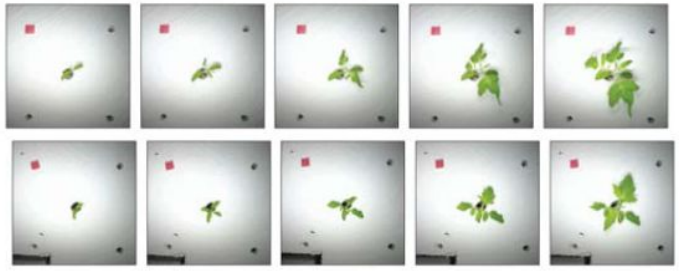


TABLE I
SOME CONTROL PARAMETERS AND THEIR DEFINITIONS

Variable	Description
pH	pH - acidity of an aqueous solution
H	Relative humidity
T	Temperature of ambient air
T_{wet}	Temperature of nutrient solution
EC	Electrical conductivity of feeding solution
F	Recycling flow rate
L	Light duty cycle with applied moving average
F_1	$S(T - T_{wet})/T$
F_2	$e^{-pH}T_{wet}/S^{1.1}$
F_3	$e^{-pH}T_{wet}/S^{1.2}$

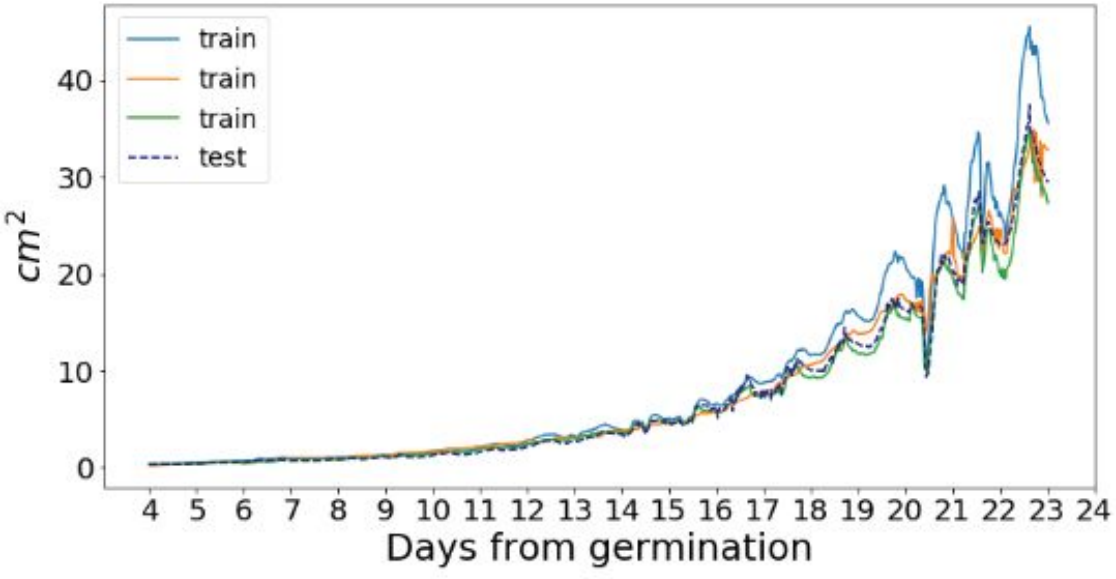


Fig. 3. Leaf area calculations during the experiment.

Another interesting applications of DMD-2

Aim: combine DMD+RNN for predictions in spoiled apples.

`https://github.com/NikitaStasenko/PostharvestDecayApples`

Fig. 1. Sequential images of one apple from the dataset.

Fig. 2. Examples of apples selected for data processing.

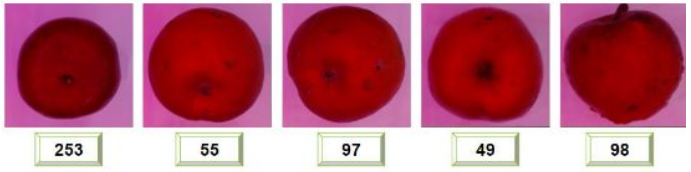


Fig. 3. Apples used for data collection in the greenhouse.

Fig. 4. Example of image annotation and object classes in Supervisely.

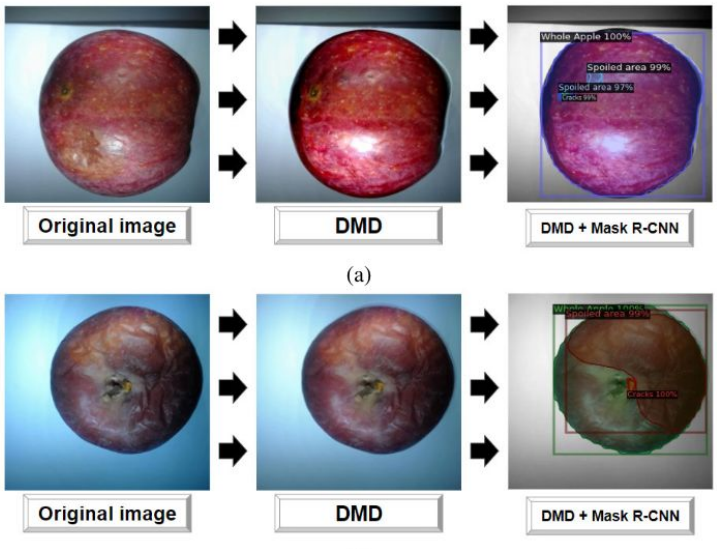
Another interesting applications of DMD-2

Aim: combine DMD+RNN for predictions in spoiled apples.
`https://github.com/NikitaStasenko/PostharvestDecayApples`

TABLE I

COMPARISON OF DMD FITTING TIME ON IMAGES WITH DIFFERENT SHAPES

Number of images in the subset	Number of training images	DMD fitting time on original images, s (640×480×3)	DMD fitting time on original images, s (300×300×3)
202	157	151.564	**26.192**
228	182	204.652	**22.917**
247	191	200.742	**26.195**
280	208	216.816	**24.389**
295	214	238.820	**27.287**
470	357	517.640	**41.295**
545	415	1163.625	**103.770**



If case of additional knowledge on physics of the system? Apply PI-DMD:
`https://github.com/baddoo/piDMD`
The problem of rank-R A apprimation has to be solved on a certain manifold (e.g. tridiagonal).

Question 3-2: Can we vary techniques? Apply cross sampling in case of advection-coagulation!

- Problem setting:

$$\frac{\partial f(t,x,v)}{\partial t} + c(v)\frac{\partial f(t,x,v)}{\partial x} = \mathcal{F}(f(t,x,v)),$$

Initial condition: $f(t=0,x,v) = f_0(x,v)$,

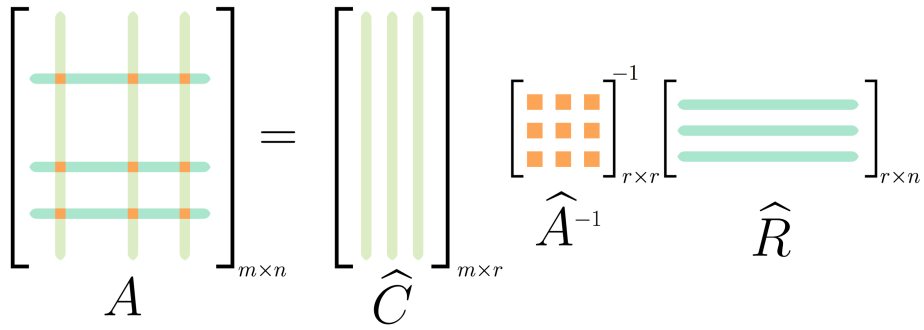
Boundary condition: $f(t,x=0,v) = f_b(t,x)$.

- Useful for aerosol science. We apply it for data from Baikal location.
- If we re-write the exact time-integration scheme – it is a rule for grid function (TVD for the transport operator $\mathcal{A}$)

$$f(t^{k+1},x,v) = f(t^k,x,v) + \Delta t \left[\mathcal{A}(f(t^k,x,v)) + \mathcal{F}(f(t^k,x,v))\right].$$

- Apply the cross approximation and use the structure of aggregation and advection operators.

Matrix cross interpolation basic idea



Theorem (Goreinov, Tyrtyshnikov, Zamarashkin): Let

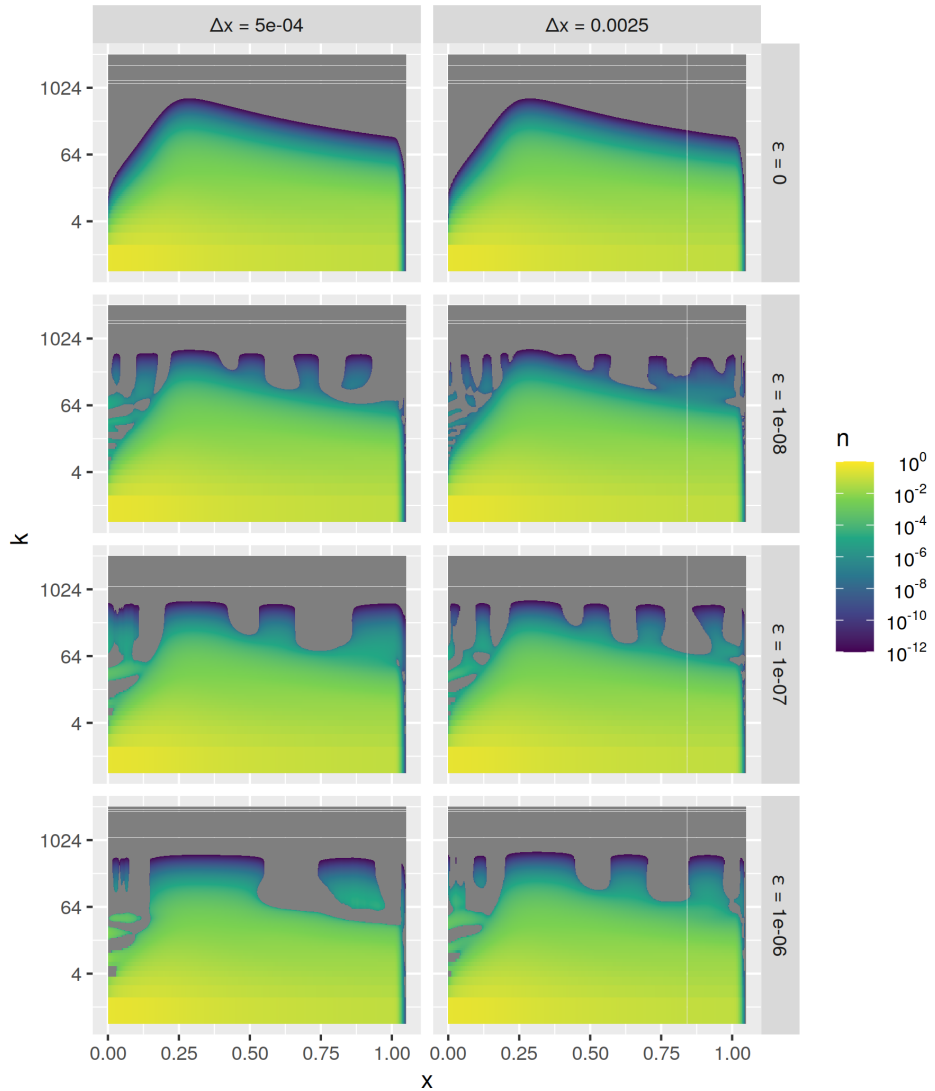
$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix},$$

where X_{11} – block $r \times r$ with maximal absolute value of the determinantamont blocks of size $r \times r$. Then rank-r decomposition

$$X_r = \begin{bmatrix} X_{11} \\ X_{21} \end{bmatrix} X_{11}^{-1} \begin{bmatrix} X_{11} X_{12} \end{bmatrix}$$

leads to an approximation of X in $||*||_C$ with error not worse than $(r+1)^2$ times larger than comparison with optimal approximation.

Numerical results



We adapt the ranks with respect to the target accuracy.

Russ. J. Numer. Anal. Math. Modelling 2021; 36(3):177–181

Numerical results

We obtain the acceleration of computations

Δx	Δt	ε	t, sec.	R_N	Relative error
2.5×10^{-3}	1.25×10^{-3}	—	4.6	—	—
5×10^{-4}	2.5×10^{-4}	—	117	—	—
2.5×10^{-3}	1.25×10^{-3}	10^{-6}	0.68	11	1.4×10^{-3}
2.5×10^{-3}	1.25×10^{-3}	10^{-7}	1.12	14	1.2×10^{-4}
2.5×10^{-3}	1.25×10^{-3}	10^{-8}	3.12	25	4.4×10^{-6}
5×10^{-4}	2.5×10^{-4}	10^{-6}	4.17	9	5.9×10^{-3}
5×10^{-4}	2.5×10^{-4}	10^{-7}	6.29	12	5.6×10^{-4}
5×10^{-4}	2.5×10^{-4}	10^{-8}	9.37	16	4.0×10^{-5}

Russ. J. Numer. Anal. Math. Modelling 2021; 36(3):177–181

Conclusion

- A big picture of model reduction methods is given: POD, POD-DEIM, DMD and low-rank matrix approximations.
- Sometimes they are applicable even for untypical problems (reduction in time).
- Allow to restore the evolution laws from data.
- This approach is quite general for the non-stationary problems.

Thank you for your attention!

Tasks:

1. Download the PyDMD and get through the tutorials.
2. Apply the PyDMD to restoration of ODE operator with matrix (take any non-trivial initial conditions)

$$A = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \\ 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 10 \end{bmatrix}.$$

   How many snapshots do you need?
3. Take any of suggested github examples and run it. Prepare the presentation on your results.
4. Implement the matrix adaptive cross approximation method (pseudo-code will be given).
5. Test the HO-DMD (higher order DMD). * this task will require additional lecture about Tucker decomposition.