POD-DEIM BASED MODEL ORDER REDUCTION FOR SPEED-UP OF FLOW PARAMETRIC STUDIES

MARTIN ISOZ[1]
[1] *Institute of Thermomechanics of the Czech Academy of Sciences,*
Dolejskova 5, 182 00 Prague, Czech Republic,
isozm@it.cas.cz

Keywords: *POD, DEIM, OpenFOAM*

Current progress in numerical methods and available computational power combined with industrial needs promote the development of more and more complex CFD simulations. However, such models are expensive from the point of view of the data storage and the time necessary for their evaluation. Nevertheless, industrial practice often calls for parametric studies and optimizations performed using these models.

The model order reduction (MOR) is a useful tool for accelerating calculations connected to parametric studies or optimizations of complex systems. In this paper, we consider MOR based on the proper orthogonal decomposition (POD) with Galerkin projection, which is well described for example by Pinnau [1] or Volkwein [2]. The problems arising from the nonlinearities present in the original model are addressed within the framework of the discrete empirical interpolation method (DEIM) of Chaturantabut and Sorensen [3].

The presented reduced order model (ROM) creation technique represents an *a posteriori* approach to MOR [4]. Hence, the solution of the full order model (FOM) has to be available for the ROM creation. Moreover, not only the solution of the system is necessary for the POD-DEIM based model order reduction, the method also needs access to several other data structures, which are internal to the CFD solver and which are usually not readily available.

The open-source CFD library OpenFOAM [5] provides an easy access to the internal data structures needed for the ROM construction. Furthermore, OpenFOAM has been successfully applied to the solution of industrial scale problems in various fields of CFD. In this paper, we provide a link between POD-DEIM based MOR and the OpenFOAM library. Then, we use POD-DEIM based MOR to reduce the computational time necessary for evaluation of parametric studies in OpenFOAM.

Link between POD-DEIM based MOR and OpenFOAM

In order to outline the link between POD-DEIM based model order reduction and the OpenFOAM CFD library, we concentrate solely on the steady-state incompressible Navier-Stokes equations for laminar flow,

$$\nabla \cdot (\mathbf{U} \otimes \mathbf{U}) - \nabla \cdot (\nu \nabla \mathbf{U}) = -\nabla p + \boldsymbol{f}, \quad \nabla \cdot \mathbf{U} = 0\,, \tag{1}$$

defined on a simply connected bounded domain $\Omega \subset \mathbb{R}^3$ and supplemented with appropriate boundary data. We study the dependence of the solution $[\mathbf{U},\, p]$ to the system (1) on some parameter $\mu \in D \subset \mathbb{R}$, i.e. $\mathbf{U} : D \times \Omega \to \mathbb{R}^3$, $p : D \times \Omega \to \mathbb{R}$. To be able to apply the proper orthogonal decomposition and discrete empirical interpolation to construct a reduced order model, the full order model has to have the form of

$$\mathbf{0} = A\boldsymbol{y} + \boldsymbol{b}(\mu, \boldsymbol{y}), \quad \forall \mu \in D\,, \tag{2}$$

where A is the system matrix, $\boldsymbol{b}(\mu, \boldsymbol{y})$ is the vector of nonlinearities and $\boldsymbol{y}$ is the vector of unknowns.

After the projection of the full order model solution space V on the ℓ-dimensional subspace V^ℓ spanned by the POD basis $\{\boldsymbol{\psi}\}_{j=1}^{\ell}$ we obtain the following reduced order system,

$$\mathbf{0} = A^\ell \boldsymbol{\eta}^\ell + \boldsymbol{f}^\ell(\mu, \boldsymbol{\eta}^\ell)\,, \tag{3}$$

where we defined the vector-valued mapping

$$\boldsymbol{\eta}^\ell := (\eta_i^\ell)^{\mathrm{T}} : D \to \mathbb{R}^m, \quad i = 1, \ldots, \ell\,, \tag{4}$$

the reduced system matrix

$$A^\ell := (a_{ij}^\ell) \in \mathbb{R}^{\ell \times \ell}, \quad a_{ij}^\ell = \langle A\boldsymbol{\psi}_j, \boldsymbol{\psi}_i \rangle\,, \tag{5}$$

and the ROM nonlinearities

$$\boldsymbol{f}^\ell := (f_i^\ell)^{\mathrm{T}} : D \to \mathbb{R}^\ell, \quad f_i^\ell(\mu, \boldsymbol{\eta}) = \left\langle \boldsymbol{b}\left(\mu, \sum_{j=1}^{\ell} \eta_j \boldsymbol{\psi}_j\right), \boldsymbol{\psi}_i \right\rangle . \tag{6}$$

In the above, we denote by $\langle \cdot, \cdot \rangle$ an inner product. The costs connected to the evaluation of the nonlinearities (6) are mitigated via DEIM, see [3].
The continuity equation $(1)_2$ is pressure free. Thus, the linearization and discretization of (1) on a finite volume mesh Ω^h with m cells leads to a saddle point problem,

$$\begin{pmatrix} M & N^T \\ N & 0 \end{pmatrix} \begin{pmatrix} \mathbf{U}^h \\ p^h \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}^h \\ 0 \end{pmatrix} , \tag{7}$$

where we denoted the discrete representations of the considered functions by the superscript h. The matrix $N \in \mathbb{R}^{m \times 3m}$ coincides with a discrete representation of the ∇ operator. The matrix $M \in \mathbb{R}^{3m \times 3m}$ is slightly more difficult. The Navier-Stokes equations for an incompressible isothermal flow are nonlinear. Hence, the nonlinear convective term $\nabla \cdot (\mathbf{U} \otimes \mathbf{U})$, needs to be linearized during the construction of the matrix M. If we apply the Newton linearization to the nonlinear convective term,

$$\nabla \cdot (\mathbf{U}^n \otimes \mathbf{U}^n) \approx \mathbf{U}^o \cdot \nabla \mathbf{U}^n + \mathbf{U}^n \cdot \nabla \mathbf{U}^o =: \mathcal{P}(\mathbf{U}^o, \mathbf{U}^n), \quad n \ldots \text{current time step/iteration}, \ o \ldots \text{previous iteration} , \tag{8}$$

we can define a linear operator

$$\mathcal{M}(\mathbf{U}^o, \mathbf{U}^n) := \nabla \cdot (\nu \nabla \mathbf{U}^n) + \mathcal{P}(\mathbf{U}^o, \mathbf{U}^n) , \tag{9}$$

and the matrix M is a discrete representation of the operator $\mathcal{M}$.
The matrices M and N are, as results of the FV discretization, large and sparse. The system (7) cannot be directly solved by the available methods of numerical linear algebra [6, 7]. However, if one assumes the matrix M to be regular, it is possible to explicitly express the velocity from the first row of the system (7) and to substitute for it in the second row. Doing so, the following system for one unknown $p^h \in \mathbb{R}^m$ is obtained

$$N M^{-1} N^{\mathrm{T}} p^h = N M^{-1} \boldsymbol{f}^h, \quad \mathbf{U}^h = M^{-1} \left(\boldsymbol{f}^h - N^{\mathrm{T}} p^h \right) . \tag{10}$$

Unfortunately, because M is a large sparse matrix, its inverse is usually not obtainable.
The issue of calculation of the inverse M^{-1} is mitigated by splitting the matrix M as $M = D + L + U$, where D, L and U are a diagonal, and lower and upper triangular matrix, respectively. The inverse M^{-1} is approximated as $M^{-1} \approx D^{-1}$ and the following iterations are constructed,

$$N D^{-1} N^{\mathrm{T}} p^{h,n} = N D^{-1} (\boldsymbol{f}^h - (U + L) \mathbf{U}^{h,o}), \quad \mathbf{U}^{h,n} = D^{-1} \left(\boldsymbol{f}^h - (U + L) \mathbf{U}^{h,o} - N^{\mathrm{T}} p^h \right) \tag{11}$$

The reduced order model generated from the solution to (1) computed by a SIMPLE-based OpenFOAM solver (e.g. simpleFoam) must take into account the above outlined solution procedure. The natural variable for solution techniques for the incompressible Navier-Stokes equations based on the solution of the system (11) is the pressure. Thus, it would seem reasonable to base the reduced order model directly on the relations (11). We propose to rewrite the system $(11)_1$ as

$$\mathbf{0} = A(\mu) p^h + \boldsymbol{b}(\mu, p^h), \quad A(\mu) := N D^{-1} N^{\mathrm{T}}, \quad \boldsymbol{b}(\mu, p^h) := N D^{-1} (\boldsymbol{f}^h - (U + L) \mathbf{U}^h) , \tag{12}$$

where $\mathbf{U}^h$ is the solution for the velocity obtained from the SIMPLE iterations. The system (12) is in the form (3) and can be processed by POD and DEIM.

Remark 1 *The iterations* (11) *do not exactly correspond to the solution procedure applied in the OpenFOAM solvers. For example, the influence of the mesh non-orthogonality on the solution procedure is not mentioned. Consequently, the matrix A and vector b in* (12) *are not actually constructed via the given formulas. Instead, they are obtained from* `fvm::laplacian(rAtU(), p)` *and* `fvc::div(phiHbyA)`*, respectively. Still, the overal structure of A and b is as described above.*

To reduce the order of (12) via the proper orthogonal decomposition and the discrete empirical interpolation, we need to have available the following data (full order system snapshots),

$$\boldsymbol{S} = \left\{ p_i^h := p^h(\mu_i), \mathbf{U}_i^h := \mathbf{U}^h(\mu_i) \right\}_{i=1}^n . \tag{13}$$

Note that the matrix A in (12) depends on the value of the parameter μ. Hence, only matrices $A(\mu)$ for $\mu \in \{\mu_i\}_{i=1}^n$ are available during the reduced order model construction. The matrix $A(\mu)$ for $\mu \notin \{\mu_i\}_{i=1}^n$ has to be approximated via interpolation. Let us denote $[A]_i$ the interpolated matrix. If the linear interpolation is applied to compute $[A]_i(\mu)$, then $\langle [A]_i \boldsymbol{\psi}_j, \boldsymbol{\psi}_i \rangle = [\langle A \boldsymbol{\psi}_j, \boldsymbol{\psi}_i \rangle]_i$ and it is possible to pre-compute the reduced system matrices during the reduced order model construction.

Remark 2 *The linear interpolation is used to evaluate $[A^\ell]_i(\mu)$ for $\mu \notin \{\mu_i\}_{i=1}^n$. Let us order the values of μ at which we saved the snapshots $\boldsymbol{S}$ as $\mu_1 < \mu_2 < \cdots < \mu_n$. Because of the applied linear interpolation, the generated reduced order model can be used only for $\mu \in [\mu_1, \mu_n]$.*

Optimal FOM snapshot selection

We study the dependence of the solution $[\mathbf{U},\, p]$ to the system (1) on the parameter $\mu \in D \subset \mathbb{R}$. Let us assume that we have a prepared reduced order model. Now, let us introduce $\boldsymbol{\mu}_{\mathrm{u}}$, the vector of parameter values for which the full order model was already evaluated. Mind that the current ROM is based on snapshots of the full order model solution computed for $\mu \in \boldsymbol{\mu}_{\mathrm{u}}$. Let $\boldsymbol{\mu}$ denote the vector of parameter values of interest for which the FOM was not yet evaluated. We want to find the best possible $\tilde{\mu} \in \boldsymbol{\mu}$ to include in $\boldsymbol{\mu}_{\mathrm{u}}$. In other words, we seek such $\tilde{\mu} \in \boldsymbol{\mu}$ that expansion of the current matrix of snapshot by the FOM snapshot computed for $\tilde{\mu}$ will improve the most possible the quality of the resulting ROM.
We propose to use the discretized continuity error and define a cost function

$$S_{\mathrm{cont}} = \sum_{P \in \Omega^h} \left| \sum_{f} \Phi_f^{\mathrm{rom}} \right| , \quad \Phi_f^{\mathrm{rom}} = \mathbf{S}_f \cdot \mathbf{U}_{\mathrm{rom}} \,, \tag{14}$$

where $\mathbf{S}_f$ is the face-area vector of the face f. Note that (14) corresponds to a sum of all the local continuity errors in the finite volume mesh Ω^h. Based on numerical experiments, we found that it is suitable to incorporate into the cost function the distance of the current tested parameter to the nearest parameter already included in the snapshots. The modified cost function is

$$S_{\mathrm{cont}}^{\mathrm{nn}} = d_{\mathrm{nn}}^{q} S_{\mathrm{cont}} \,, \tag{15}$$

where d_{nn} marks the distance of the tested parameter $\tilde{\mu}$ to its nearest neighbor in $\boldsymbol{\mu}_{\mathrm{u}}$. The power q may be used to adjust the importance of d_{nn} for the final value of $S_{\mathrm{cont}}^{\mathrm{nn}}$.
The next value of the parameter to evaluate the full order model for, $\tilde{\mu} \in \boldsymbol{\mu}$, can be found as,

$$\tilde{\mu} := \underset{\mu \in \boldsymbol{\mu}}{\operatorname{argmax}}\, S_{\mathrm{cont}}(\mu) \quad \text{or} \quad \tilde{\mu} := \underset{\mu \in \boldsymbol{\mu}}{\operatorname{argmax}}\, S_{\mathrm{cont}}^{\mathrm{nn}}(\mu) \tag{16}$$

The reduced order model may be used to predict the initial guess for p and $\mathbf{U}$ for SIMPLE iterations. The proposed approach to the evaluation of a parametric study with ROM predictor is summarized in Algorithm 1.

Algorithm 1 Parametric study evaluation with POD-DEIM based predictor

```
Require: Vector of parameters of interest $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_d]^{\mathrm{T}}$, $\mu_1 < \mu_2 < \cdots < \mu_d$, target value of the cost function, $\varepsilon_S$
Evaluate FOM at $\mu_1$ and $\mu_d$;
Set $\boldsymbol{\mu}_{\mathrm{u}} := [\mu_1, \mu_d]$;
Set $\boldsymbol{\mu} := \boldsymbol{\mu} \backslash \boldsymbol{\mu}_{\mathrm{u}}$;
while $\boldsymbol{\mu} \neq \emptyset$ do
    Construct ROM from snapshots available for $\boldsymbol{\mu}_{\mathrm{u}}$;
    for each $\mu \in \boldsymbol{\mu}$ do
        Evaluate ROM at $\mu \rightsquigarrow p_{\mathrm{rom}}(\mu)$ and $\mathbf{U}_{\mathrm{rom}}(\mu)$;
        Evaluate $S_{\mathrm{cont}}(\mu)$ from (14);
    end for
    if $\max_{\tilde{\mu} \in \boldsymbol{\mu}} \{S_{\mathrm{cont}}(\tilde{\mu})\} < \varepsilon_S$ then BREAK; end if
    Set $\tilde{\mu} := \operatorname{argmax}_{\tilde{\mu} \in \boldsymbol{\mu}} \{S_{\mathrm{cont}}(\tilde{\mu})\}$;
    Evaluate FOM at $\tilde{\mu}$;
    Set $\boldsymbol{\mu}_{\mathrm{u}} := [\boldsymbol{\mu}_{\mathrm{u}}, \tilde{\mu}]$;
    Set $\boldsymbol{\mu} := \boldsymbol{\mu} \backslash \{\tilde{\mu}\}$;
end while
for each $\mu \in \boldsymbol{\mu}$ do
    Evaluate ROM at $\mu \rightsquigarrow p_{\mathrm{rom}}(\mu)$ and $\mathbf{U}_{\mathrm{rom}}(\mu)$;
    Evaluate FOM at $\mu$ with $p_{\mathrm{rom}}(\mu)$ and $\mathbf{U}_{\mathrm{rom}}(\mu)$ as initial guess;
end for
return Parametric study of FOM
```

Numerical example

The proposed approach was tested on the `cavity tutorial` distributed with the core OpenFOAM installation [5]. The changed parameter was the flow Reynolds number Re, which was alternated via the changes in the viscosity of the considered fluid. The tested Reynolds numbers ranged from 1 to 10000.
In Figure 1, we depict a comparison of the number of SIMPLE iterations needed to obtain a FOM solution with the same level of accuracy. The compared cases are (i) a CFD simulation without any a-priori pressure and velocity fields initialization, (ii) a simulation with the initial guess predicted by a ROM of dimension $\ell = 6$ and the snapshots selected

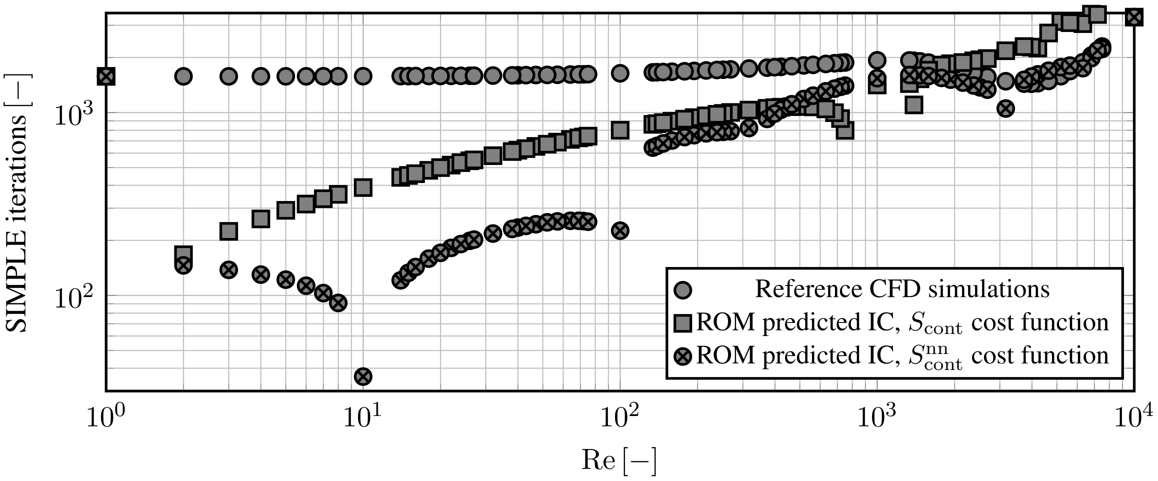

Figure 1: Comparison of the number of SIMPLE iterations needed for the simulation to converge from ROM predicted initial guess of pressure and velocity fields and from a uniform fields initialization. The total number of SIMPLE iterations performed is either 68677 or 99637 in the case of ROM predicted initial guess (it depends on the used cost function) and 126571 otherwise.

via Algorithm 1 and the cost function (14), and (iii) a simulation with the initial guess predicted by a ROM of dimension $\ell = 6$ and the snapshots selected similarly as in (ii), but with the cost function (15).

As it may be seen from Figure 1, when the cost function $S^{\text{nn}}_{\text{cont}}$ was applied, we were able to achieve speed-up of almost two. However, we would like to mention two things. First, the selected range of Reynolds numbers to be simulated is rather large – the studied parameter changes by four orders of magnitude. For most of the practical applications, the range of parameters of interest will be significantly more narrow and the results of the POD-DEIM predictor might be better. Second, using the cost function $S^{\text{nn}}_{\text{cont}}$ defined by (15) instead of S_{cont} introduced in (14) might seem beneficial from the point of view of the final computation speed up. Nevertheless, it seems that the optimal value of the power q in (15) is problem-dependent and we do not have any universally valid estimate for it. In practice, we are not interested in the maximal possible speed up of the calculation but in the actual results of the parametric study. Hence, it does not make much sense to find the optimal value of q for each studied case.

Acknowledgments

This work was supported by the European Regional Development Fund under Grant No. CZ.02.1.01/0.0/0.0/15_003/0000493 (Centre of Excellence for Nonlinear Dynamic Behaviour of Advanced Materials in Engineering). Moreover, the author would like to thank Michael Hinze from the University Hamburg for many helpful discussions on the proper orthogonal decomposition and related topics.

References

[1] R. Pinnau, “Model reduction via Proper Orthogonal Decomposition,” in *Model order reduction: theory, research aspects and applications*, W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, Eds. Berlin, Germany: Springer Berlin Heidelberg, 2008, ch. 2, pp. 95–109.

[2] S. Volkwein, *Proper orthogonal decomposition: theory and reduced-order modelling*, 1st ed. Konstanz, Germany: Universtity of Konstanz, 2013.

[3] S. Chaturantabut and D. C. Sorensen, “Nonlinear model reduction via discrete empirical interpolation,” *SIAM J. Sci. Comp.*, vol. 32, pp. 2737–2764, 2010.

[4] A. Cosimo, A. Cardona, and S. Idelsohn, “General treatment of essential boundary conditions in reduced order morels for non-linear problems,” *Adv. Model. and Simul.*, vol. 3, p. 14, 2016.

[5] OpenCFD, *OpenFOAM: The Open Source CFD Toolbox. User Guide Version 1.4, OpenCFD Limited.* Reading UK, 2007.

[6] M. Benzi, G. H. Golub, and J. Liesen, “Numerical solution of saddle point problems,” *Acta Numerica*, vol. 14, pp. 1–137, 2005.

[7] H. C. Elman and G. H. Golub, “Inexact and preconditioned Uzawa algorithms for saddle point problems,” *SIAM J. Num. Anal.*, vol. 31, pp. 1645–1661, 1994.