An introduction to Reduced-Order Models (ROM)

Martí Burcet*

E-mail: martiburcet@gmail.com

Abstract

Reduced-order models (ROM) are a numerical technique to reduce the computational cost by finding a basis of functions that allows the representation of new solutions as a combination of them. In this paper two methods are presented and discussed: the Proper Orthogonal Decomposition (POD),which is an *a posteriori* technique because it is needed to solve at least once the problem with the complete basis before reducing it, and the Proper Generalized Decomposition (PGD), which is an *a priori* ROM because the basis is constructed as an enrichment process. Therefore PGD can also be used as a solver for multi-variable problems when it is not affordable to get directly a solution of the full problem.

Keywords: Reduced-order model (ROM), Proper Orthogonal Decomposition (POD), Proper Generalized Decomposition (PGD)

Introduction

When solving a model problem in the context of numerical simulation we sough for a discrete solution in a space of approximate functions associated to the mesh. This space have as many functions as nodes have the mesh (M). This may imply for instance for transient

problems, to compute M values at each time step and for non-linear problems, a system of equations of size M. For large problems this may lead to computational expensive solutions.

However the dimension of the space of functions to sough for the solution in many cases is too large and actually the solution may live in a subspace of it. So it makes sense to look for a reduced-order model solution which will be substantially cheaper to compute. The construction of the ROM will imply a computationally expensive process which will be done just one time but, further solutions constructed with the ROM solution will be substantially cheaper to compute. The ROM based solutions due to its cheap computationally cost are specially likely to be used in real time simulations.

In this article we outline two reduced-order models: the Proper Orthogonal Decomposition (POD) and the Proper Generalised Decomposition (PGD). We expose the main differences of them and we also give some advice of which cases it is better one or another.

Proper Orthogonal Decomposition (POD)

As an illustrative example of a two variables function, assume a variable of interest depending both on space and time $u(x,t)$ and assume that the discrete solution is known at nodal level in each time step. This is assume the solution is stored in a matrix of $M \times P$ dimension being M the number of nodes of the spatial discretization and P the number of time steps. Therefore the solution for all nodes at time n (u^n) corresponds to the n-th column of the matrix.

The idea of POD is to find the most relevant or characteristic structure $\phi(X)$ among u^m such that the solution at any time step could be built combining these characteristic

structures. To do so we have to maximise the scalar quantity α (1)

$$\alpha = \frac{\sum_{m=1}^{p} \left[\sum_{i=1}^{M} \phi(x_i) u^m(x_i)\right]^2}{\sum_{i=1}^{M} \left(\phi(x_i)\right)^2} \tag{1}$$

This maximisation problem requires solving the eigenvalue problem (2)

$$\mathbf{c}\phi = \alpha\phi \qquad where \qquad \mathbf{c} = \sum_{m=1}^{P} u^m \cdot (u^m)^T \tag{2}$$

with $\mathbf{c}$ being a symmetric and definite positive operator.

Building of the ROM from POD

The construction of the ROM solution is done selecting the eigenvectors ϕ_i associated to the N largest eigenvalues solution of (2). The weight of the eigenvalues generally decrease exponentially, so N normally is taken as the number of eigenevalues within a margin with respect to the largest one (e.g. 10^{-8}). In general for a time discretization with enough intervals, N is much lower than M so the reduction is justified.

Then the construction of a new solution based on the ROM is done combining the "basis" functions ϕ_i. For example in an explicit time discretization, the new solution is computed from the ROM basis as (3):

$$u^{m+1} \approx \sum_{i=1}^{N} \phi_i \zeta_i^{m+1} = \mathbf{B}\zeta^{m+1} \tag{3}$$

where ζ^{m+1} are the coefficients of the interpolation to be determined and $\mathbf{B}$ is the matrix of N basis functions ϕ_i by columns.

So the coefficients ζ^{m+1} are found solving an algebraic system of equations of size N

instead of M, what implies an important reduction in finding the *a posteriori* solution. This approach is likely to be used in finding the solution at new time steps in the base of a small ROM computed only for few time intervals. Another case where POD is useful is in finding new solutions of a similar problem with slight changes in parameters, but the method can exhibit lack of robustness.

Proper Generalized Decomposition (PGD)

Consider a general problem depending on D variables of any kind: space, time, material parameters, geometry, boundary conditions... The space where to seek for a solution is the cross product of each space separetely, that is $(x_1, \ldots, x_D) \in \Omega_1 \times \cdots \times \Omega_D$.

If we aimed to find a "general" solution in the form of POD that would means storing in a D-th order tensor the solution for all possible combinations between all variables in what the problem depends. It is clear that both the combinations and the storing capacity required to solve a problem such that grows exponentially and we will run out of computational resources very quickly.

The alternative that PGD offers to that problem is the construction of a separable approximate solution of the form

$$u(x_1, \ldots, x_D) \approx u^N(x_1, \ldots, x_D) = \sum_{i=1}^{N} F_i^1(x_1) \times \cdots \times F_i^D(x_D) \tag{4}$$

where N is the number of terms of the representation and $F_i^j(x_j)$ is a function only depending on x_j, both of them known in advance of the construction of the PGD solution. The reduction of the space of possible solutions is significant because the original problem with a mesh of M nodes was of dimension M^D while the PGD solution is of size $N \times M \times D$.

Building of the ROM from PGD

The construction of the PDG solution is done with successive enrichment of the already computed solution. This means that for a solution u^n the enrichment $n+1$ will be constructed as

$$u^{n+1}(x_1, \ldots, x_D) = u^n(x_1, \ldots, x_D) + F^1_{n+1}(x_1) \times \cdots \times F^D_{n+1}(x_D) \tag{5}$$

what means that at each enrichment only D terms have to be computed. An interesting property of this construction is that we can improve the representation if this is not good enough adding more terms without having to start again.

Numerical examples done using PGD show that the number of terms needed to represent the solution doesn't depend on the problem size D but on the separable character of the exact solution to represent. This is a good new because we can consider as much variables as required in the problem without worrying that this would affect the computation of the PGD solution. In addition numerical examples show that as the number of terms (N) increases the solution converges to the exact solution of the complete tensor product approximation. In particular in a case where the solution wasn't separable, the solution would be enriched until considering all the terms of the functional space (M^D).

Comparison between POD and PGD

The main difference between POD and PGD is that to get the basis for the ROM in POD it is needed to compute at least once the solution of the problem with the complete approximation basis. Therefore we talk of POD as an *a posteriori* ROM. On the other hand to get a ROM solution for the PGD method we don't need to solve the problem with the complete basis but enrich the basis of the PGD from the previous solution until we get the desired precision. So in this case we talk of an *a priori* approach.

Regarding the sources of errors, when we solve a similar problem in POD with the basis functions of a complete basis approximation the number of degrees of freedom scales with the number of functions in the reduced order basis. Therefore there is a difficulty of capturing all the details of a different model with the reduced order basis and an inherent error will occur. However in PGD the sources of error arise from the spatial discretization and also in the number of terms taken to represent the new solution. So with a fine enough mesh and with enough terms of the basis in a separable problem we will converge to the solution.

About the reduction in computational resources, POD requires to solve once the model with the complete basis functions, so for problems involving many variables (in practice more than two) it is untreatable to solve the full problem to then get the ROM. It is still useful despite this to solve similar problems in a fast and quite accurate manner. On its site, PGD can be used to solve problems with lots of variables that in many cases cannot be solved at once. So apart of a ROM to get a basis of functions to construct new solutions in a fast way, it can also be seen as an efficient solver to get solutions of multi-variable problems that couldn't be solved in a direct way because of the high number of degrees of freedom.

Conclusions

Reduced-order models are a useful and powerful tool to reduce the computational cost of constructing new solutions of a model problem. The computation of the basis functions that allow to build the ROM solution is a computational costly phase but allows the fast construction of new similar solutions in a much faster way.

While POD is an *a posteriori* ROM because it is needed to compute once the solution with the complete basis, PGD is an *a priori* ROM because the new approximation is obtained as an enrichment of the previous ones. There PGD can also viewed as a solver to obtain

soluions specially in case of multi-variable problems which are unaffordable to solve at once a part of a reduction technique.

References

(1) Francisco Chinesta, Roland Keunings, Adrien Leygue. *The Proper Generalized Decomposition for Advanced Numerical Simulations: A primer*. SPRINGER BRIEFS IN APPLIED SCIENCES AND TECHNOLOGY, Springer Cham Heidelberg New York Dordrecht London, 2014.

(2) Sergio Zlotnik, Pedro Díez, David Modesto and Antonio Huerta. *Proper Generalized Decomposition of a geometrically parametrized heat problem with geophysical applications*. INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, 2014.