Reduced Order Models With Moving Domains

Enrique Millán Valbuena
463 426 8

Abstract—The research objective is to build a Reduced Order Model (ROM) for a two-dimensional parametrized unsteady PDE with a moving boundary:

- Heat diffusion problem.
- (Bonus) Graetz convection-diffusion problem.

Both the main body of the PDE and the geometrical definition of the moving boundary will be parametrized.

A concise description of the reducing procedure is provided, together with a priori convergence rates for basis size estimation and a posteriori error estimators to certify the use of the Reduced Order Model. Numerical examples to showcase computational costs and implementation details are designed, implemented and validated with the Manufactured Solutions Method.

Index Terms—Reduced Order Model, Moving Domain, FEM, DEIM, POD, Galerkin-Projection

1 Vocabulary Definition

This section defines the vocabulary we will use from now on in the Thesis. It serves as an explanatory document of the graph, which shows how we move from reality to the final reduced model.

First, the analytical model in the infinite dimensional space is introduced. Secondly, we move from an infinite dimensional space to a finite one, the discretization step (explained in the context of the Finite Elements). In this section we define the algebraic Full Order Model (FOM), benchmark of our reduction work. Then, we present the two reduction techniques we shall be using in this work: a reduced basis for the Galerkin projection (ROM) and a collateral basis for the assembly of the operators (HR). Finally, the three possible algebraic reduced problems are presented.

Even if we are only interested in the combination of the reduced and collateral basis to build the complete reduced order model, the error of the final reduced model can be expressed as a combination of the other two separately.

1.1 How To Read The Graph

The graph aims at portraying the following main concepts:

1) Numerical modelling terms (grey boxes).
2) Analysis/Calculus terms (green boxes).
3) Collections of the previous (coloured rectangles).

The line between the first two concepts is sometimes blurry. We seek to put together the ideas from functional analysis, calculus and numerical analysis which are used to build numerical models of parametrized partial differential equations.

1.2 Graph Walkthrough

In the following sections we explain from the top left to the bottom right the different elements contained within the graph.

We start off from reality, which can be dealt with either with experiments, which are out of the scope of this work, or analytical models.

1.2.1 *Analytical Model*

Analytical models are studied within the body of infinite dimensional function spaces, V.

Strong Formulation

The model often starts with a Parametrized Partial Differential Equation (PDE), also known as *strong formulation*.

The strong formulation of an unsteady model can be expressed with three ingredients:

- First order time derivative $\frac{\partial u}{\partial t}$.
- Spatial operator $L_{SF}(u,\tau)$.
- Forcing term $f(\tau)$.

$$\frac{\partial u}{\partial t} + L_{SF}(u,\tau) = f(\tau) \tag{1}$$

The spatial operator and the forcing term depend on a generic parameter τ, which serves to show a general dependency on a modelling parameter μ, time t, or both.

Weak Formulation

If we equip our functional space V with an inner product $(u,v)_\tau$,

$$(u,v)_\tau = \int_{\Omega(\tau)} uv d\Omega, \tag{2}$$

which has a dependency with the parametrization if the geometry $\Omega = \Omega(\tau)$ itself does too, we can derive a variational formulation of the strong formulation with a Galerkin projection,

$$\left(\frac{\partial u}{\partial t}, v\right)_\tau + \left(L_{WF}(u,\tau), v\right)_\tau = \left(f(\tau), v\right)_\tau . \tag{3}$$

This is called the *weak formulation* because the differentiability demanded to the solution space is often smaller than the one present in the strong formulation

1.2.2 *Full Order Model (FOM)*

Eventually, we need to obtain a computable solution of the analytical model. To do so, we require a *discretization*, that is, a representation of our problem in a Finite Dimensional Space, $V_h = \text{span}(\{\phi_0(x), \phi_1(x), \ldots, \phi_{N_h}(x)\})$.

Within the context of the Finite Elements, the basis functions $\phi_i(x)$ allow us to represent our solution as a linear combination of unknown coefficients and basis functions:

$$u_h\left(\tau\right) = \sum_{i=0}^{N_h} u_{h_i}\left(\tau\right)\phi_i(x). \tag{4}$$

The application of the Galerkin projection principle with V_h leads to a system of algebraic equations ,

$$M_h(\tau)\frac{du_h}{dt} + A_h(\tau)u_h = F_h(\tau), \tag{5}$$

where u_h represents the vector of unknown coefficients.

This system of equations is a semi-discrete system of equations, since it is discrete in space but not in time. Further discretization of the time derivative is necessary to obtain the time evolution of the coefficients, but that level of detail is not required to discuss and build the reduced order model.

In numerical terms and within the scope of this work, since in general we do not know the true solution $u(x)$ of the PDE , we refer to the solution of this algebraic system u_h as *the truth*. It is against this value we shall benchmark our reduced order models.

1.3 Reduction Techniques

Solving the previous algebraic problem is often costly in terms of memory and computation time. Two simultaneous issues can be addressed separately from each other, to later be combined:

- State-Space Reduction: replace the generic FE basis functions (compact support) with a problem-based basis functions ψ_i (global support), :
$$V_N = \text{span}(\{\psi_0(x), \psi_1(x), \dots, \psi_N(x)\}),$$
$$u_N(\tau) = \sum_{i=0}^{N} u_{N_i}(\tau)\,\psi_i(x).$$
- System Approximation, a.k.a. hyper-reduction: find a problem-based basis for each algebraic operator:
$$A_h^m(\tau) = \sum_q^{Q_a} \Theta_q^a(\tau) A_q \tag{6}$$

To deal with the algebraic problem and not the continuous problem allows us to have a complete black box approach to the reduction problem.

In broad terms, a problem-based basis can be generated in two ways:

1) Greedy: error-based selection of the best snapshots.
2) Automatic: POD over a collection of solution snapshots.

The first requires an error estimator, the second one a good sampling strategy over the parameter space.

1.3.1 Reduced Order Model (ROM)

If we only make use the reduced basis V_N, we are in the context of a RB-Galerkin procedure. We have less equations to solve, but for each time step we still need to assemble the full order operators and project them unto the reduced basis:
$$M_N(\tau)\frac{du_N}{dt} + A_N(\tau)u_N = F_N(\tau). \tag{7}$$

1.3.2 Hyper-Reduced Full Order Model (HFOM)

If we only make use of the system approximation for the operators, we are going to be skip the assembly step, but our system of equations will remain costly to solve, since we are still in the generic context of the FE basis functions:
$$M_h^m(\tau)\frac{du_h^m}{dt} + A_h^m(\tau)u_h^m = F_h^m(\tau). \tag{8}$$

1.3.3 Hyper Reduced Order Model (HROM)

If we make use the reduced basis V_N and the system approximation for the operators, we are in the context of a HRB-Galerkin procedure. We have less equations to solve, and additionally we can directly assemble the reduced operators without having to build the complete full order operators:
$$M_N^m(\tau)\frac{du_N^m}{dt} + A_N^m(\tau)u_N^m = F_N^m(\tau). \tag{9}$$

2 Error Definitions

There are a suite of errors and their norms that we are concerned with.

- HROM Error:
$$e_N^m(\tau) = u_h(\tau) - \mathbb{V}u_N^m(\tau)$$
$$e_N^m(\tau) = e_h^m(\tau) + e_N(\tau)$$
- HFOM Error:
$$e_h^m(\tau) = u_h(\tau) - u_h^m(\tau) \tag{10}$$
- ROM Error w.r.t. HFOM:
$$e_N(\tau) = u_h^m(\tau) - \mathbb{V}u_N^m(\tau) \tag{11}$$
- ROM Error (skipped in the derivations, why?):
$$e_h(\tau) = u_h(\tau) - \mathbb{V}u_N(\tau) \tag{12}$$

Brief derivation:
$$e_N^m(\tau) = u_h(\tau) - \mathbb{V}u_N^m(\tau)$$
$$e_N^m(\tau) = u_h(\tau) - \mathbb{V}u_N^m(\tau) \pm u_h^m(\tau)$$
$$e_N^m(\tau) = [u_h(\tau) - u_h^m(\tau)] + [u_h^m(\tau) - \mathbb{V}u_N^m(\tau)]$$
$$e_N^m(\tau) = e_h^m(\tau) + e_N(\tau)$$

Additionally, one can study how does the error in the reduction of each operator affect the solution error:
$$e_A^m(\tau) = A_h(\tau) - A_h^m(\tau)$$
$$e_F^m(\tau) = F_h(\tau) - F_h^m(\tau)$$

The question is to determine:
$$e_N^m(\tau) = f\left(e_A^m(\tau), e_F^m(\tau)\right) \tag{13}$$

Finally, one has to investigate if the error made by the discretization scheme in time plays any role in the reduction process.

3 Remaining questions

- How is the collateral basis obtained?
- How are the approximated operators obtained?
- How is an efficient offline-online decomposition achieved?
- How are the errors bounded?