Finite Element Reduced Basis (Proper Ortogonal Decomposition) Approach for Geometrically Parametrised Stokes Flow

Nirav Vasant Shah

May 17, 2017

Contents

1 Introduction

Stokes flow, also known as Viscous flow, has its application in the fields of mechanical engineering such as oil flow in journal bearing, aerodynamics such as flow around wings, environmental engineering such as groundwater flow and biomedical applications such as cardiovascular flows. All these flows are governed by incompressible stokes flow equation. The governing equations are solved numerically by method such as Finite Element Method. Rapid and economic evaluation of output has given rise to the parametrization of partial differential equation and subsequently, Reduced Basis Methods. The following chapters demonstrate the geometric parametrization of partial differential equation.

2 Method and model

2.1 Conceptual model

Viscous flows are characterised by negligible inertial force and absence of velocity fluctuation in directions parallel as well as perpendicular to flow. The Reynolds number for such flow could be as low as, Re$\ll$1. When velocity or geometric dimensions are appropriately dimensioned such low Reynolds number is achieved. The shear stress in these flows are balanced against pressure drop and externally applied force. The important fluid property in these flow is viscocity that creates strong resistance to random movement so that the flow remains laminar and does not turn turbulent. Fluids such as, heavy oils or blood, have viscocity that creeps the flow. Due to this reason, viscous flow is also known as creeping flow.

2.2 Mathematical model

[7]

The governing equations for incompressible stokes flow are conservation equations: Mass conservation and momentum conservation. The coservations equations are derived based on concept of control volume and control surface. Control volume is the volume, fixed or moving with constant velocity in space, through which fluid moves. Control surface is the surface enclosing control volume. These equations are solved with the help of Finite Element Method(FEM). All equations can be derived from Reynold's transport equation:

$$\frac{dB}{dt}|_{system} = \frac{d}{dt}\int_{cv} b\rho dV + \int_{cs} b\rho v.dA \tag{1}$$

cv = Control volume
cs = Control surface
B = Extensive property under consideration
b = Intensive property corressponding to B
ρ = Density of fluid
v = Velocity of fluid at the control surface

By substituting B as momentum and correspondingly b as velocity and neglecting inertial terms we obtain stokes equation:

$$-\nu\Delta U + \nabla P = f \tag{2}$$

ν = Dynamic viscocity
U = Velocity of fluid
P = Pressure of fluid
f = External force

This is known as strong form of stokes flow equation.

Also by using Reynold's transport theoreom for mass conservation and substituting B as mass and corresspondingly b as 1 and neglecting change in density we obtain mass consevation for incompressible flow:

$$\nabla.U = 0 \tag{3}$$

This condition is known as divergence free condition.

It should be noted that by ignoring inertial terms the momentum equation is simplified significantly as the nonlinear terms are no longer present. Similarly by incompresible flow condition the density is no longer present in mass conservation which removes one unknown, density. If density is present, it must be coupled by equation of state with pressure and temperature given by momentum equation and energy equation respectively (Coupled system of equation).

2.3 Finite element formulation

[3]

Equation (2) is known as the strong form of the stokes equation. The other representation of strong form, the weak form i.e. in bilinear and linear continuous function has advantages for statement of uniqueness and existence of solution. Also the need for differentiability of solution function is comparatively relaxed in weak form as compared to that in strong form. The weak form can be mathematically understood as below.

The domain $\Omega \in \mathbb{R}^D$ has domain boundary $\partial\Omega$ with Dirichlet boundary condition on segments of boundary $\partial\Omega$, Γ_i^D where $1 \leq i \leq d_v$. The objective is to seek solution (u,p) in suitable function space $(\mathbb{V}, \mathbb{Q})$ in finite dimensional subspace of original domain i.e. (u,p) $\in (\mathbb{V}\left(\Omega^d\right), \mathbb{Q}\left(\Omega^d\right))$

The weak form is derived by multiplying the strong form with test function (v,q) $\in (\mathbb{V}, \mathbb{Q})$, integrating over the domain, applying intergration by parts and applying Gauss Divergence Theoreom we obtain,

$$\int_\Omega \nu \nabla v \nabla U - \int_\Omega p \nabla v = \int_\Omega f v + \int_{\partial\Omega} v(p - \nu \nabla U) \tag{4}$$

Similarly, for mass conservation

$$\int q \nabla . U = 0 \tag{5}$$

Here, u, v $\in H^1(\Omega)$ and p, q $\in L^2(\Omega)$.

In (4), the left hand side is bilinear and right hand side is linear. Hence, this form is also referred as bilinear form and represented hereby as,

$$a(u, p, v) = f(v) \tag{6}$$

Where, $a : \mathbb{V} \times \mathbb{Q} \to \mathbb{R}$ and $f : \mathbb{V} \to \mathbb{R}$

Two important characteristics of bilinear form are, continuity condition and the inf-sup condition, which are stated as below:

1. Bilinear form a is continuous if

$$|a(u, v)| \leq \gamma ||u||_{\mathbb{V}} ||v||_{\mathbb{V}} \tag{7}$$

Further Bilinear form is coercive if there exists c≥0 such that for all v $\in \mathbb{V}$

$$a(v, v) \geq c ||v||^2{}_{\mathbb{V}} \tag{8}$$

2. Weakly coercive bilinear forms satisfy the inf-sup condition,

$$\exists \beta > 0 \tag{9}$$

such that

$$\inf_{v \in \mathbb{V}} \sup_{w \in \mathbb{W}} \frac{a(v, w)}{||v||_{\mathbb{V}} ||w||_{\mathbb{W}}} \geq \beta \tag{10}$$

In the case of stokes problem the ansatz function ϕ selected is taylor-hood function space(P_2, P_1; Lagrange element), that is piecewise quadratic polynomial for velocity and piecewise continuous linear polynomial for pressure. We also introduce inner product of ansatz function M as,

$$M = \langle \phi, \phi \rangle_{\mathbb{V}} \tag{11}$$

2.4 Parametrization of finite element weak form

[6], [3]

In the context of Reduced Order Modelling we consider parametrized weak formulation. The parameter is whole set of input information that eventually influences the solution and derived outputs. Parameter can enter as fluid parameter(fluid mechanics) or material parameter(solid mechanics),

geometrical parameter or flow(fluid mechanics) or loading condition(solid mechanics).

The parameters μ are selected from defined parameter space $\mathbb{P}$. In parametrized setting the weak form can be represented as a: $\mathbb{V} \times \mathbb{V} \times \mathbb{P} \rightarrow \mathbb{R}$ and f: $\mathbb{V} \times \mathbb{P} \rightarrow \mathbb{R}$.

Hence, with parametrization the weak form can be represented as, we seek u(μ) $\in \mathbb{V}$

$$a(u, v; \mu) = f(v; \mu) \forall v \in \mathbb{V} \tag{12}$$

and evaluate output,

$$s(\mu) = l(u(\mu); \mu) \tag{13}$$

2.4.1 Geometric parametrization

[6]

In the present problem we consider geometric parametrisation. The geometric parametrisation requires μ-independent domain called reference domain. The reference domain is one realisation at a known parameter set $\mu_0 \in \mathbb{P}$. The parametric mapping, T, between Ω and $\Omega(\mu)$ is defined as ($\Omega_0(\mu)$)=T($\Omega; \mu$). We also divide reference domain into mutually non-overlapping subdomains $\Omega_0^l(\mu)$.

$$\Omega_0(\mu) = \cup_{l=1}^{L_\Omega} \Omega_0(\mu) \tag{14}$$

$$\Omega_0^l(\mu) = T^l(\Omega^l; \mu) \tag{15}$$

The domain consist of a square plate with triangular obstacle in the middle of bottom edge of the plate as shown in 1.

The parameters here (μ_1, μ_2) are co-ordinates of the tip of obstacle. On reference domain these parameters are $\mu_1 = 0.5$ and $\mu_2 = 0.3$. The domain is further subdivided into five subdomains as shown in 2.

The domain are numbered from bottom right to bottom left in counter-clockwise direction.

The transformation from reference domain to parametrised domain is given by,

$$T^l(x; \mu) = G^l(\mu)x + c^l(\mu) \tag{16}$$

for x $\in \Omega^l$ and $\mu \in \mathbb{P}$
Here,
$G^l : \mathbb{P} \rightarrow \mathbb{R}^{dXd}$
$c^l : \mathbb{P} \rightarrow \mathbb{R}^d$
Also, the corresponding Jacobian is given by $J^l(\mu) = |det(G^l(\mu))|, 1 \leq l \leq$

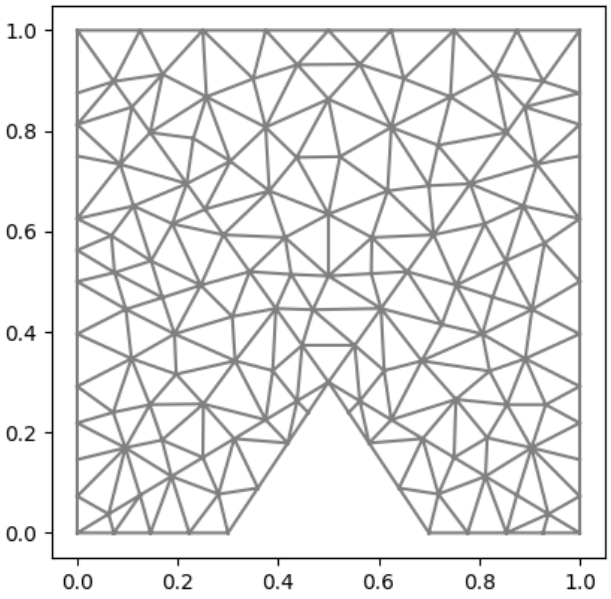

Figure 1: Domain: Square plate with triangular obstacle

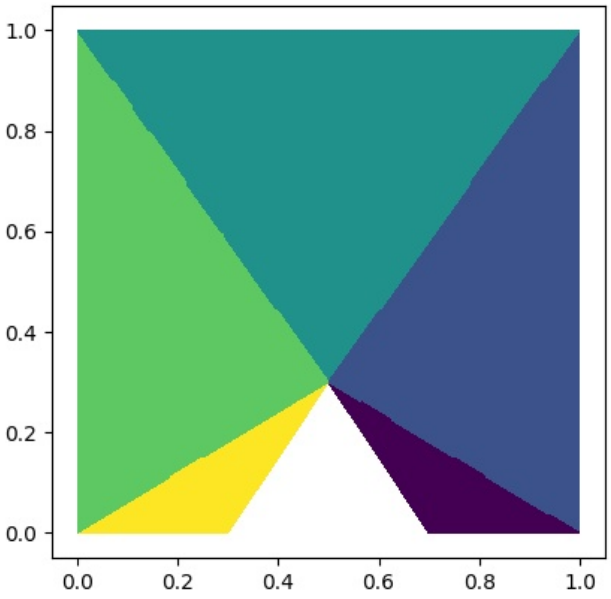

Figure 2: Division of domain into different subdomains

L_Ω

The corressponding bilinear and linear terms in weak form are given by,

$$a(w, v; \mu) = \Sigma_{l=1}^{L_\Omega} \int_{\Omega^l} D(w)^T K^l(\mu) D(v) \tag{17}$$

$$K^l(\mu) = J^l(\mu)(\hat{G})^l(\mu)(K_0^l(\mu))((\hat{G})^l(\mu))^T \tag{18}$$

$$f(v; \mu) = \Sigma_{l=1}^{L_\Omega} \int_{\Omega^l} F^l(\mu) v \tag{19}$$

$$\hat{G} = \begin{bmatrix} (G(\mu)^l)^-1 & 0 \\ 0 & 1 \end{bmatrix} \tag{20}$$

The domain was divided into 5 subdomains and transformation for each subdomain is given by,

if

$$x = \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} \tag{21}$$

Transformation matrix for each domain can be given by,

$$T_1 = \begin{bmatrix} (\mu_1 - 0.5)/0.3 & 1 \\ \mu_1/0.3 & 0 \end{bmatrix} \tag{22}$$

$$T_2 = \begin{bmatrix} 0 & (2 - 2 * \mu_1) \\ 1 & (0.6 - 2 * \mu_1) \end{bmatrix} \tag{23}$$

$$T_3 = \begin{bmatrix} (0.5 - \mu_1)/0.7 & 1 \\ (1 - \mu_2)/0.7 & 0 \end{bmatrix} \tag{24}$$

$$T_4 = \begin{bmatrix} 0 & \mu_1/0.5 \\ 1 & (\mu_1 - 0.3)/0.5 \end{bmatrix} \tag{25}$$

$$T_5 = \begin{bmatrix} (\mu_1 - 0.5)/0.3 & 1 \\ \mu_1/0.3 & 0 \end{bmatrix} \tag{26}$$

2.5 Reduced order modeling

2.5.1 Reduced basis space

The idea behind Reduced Basis methods is to generate an approximate solution to high fidelity problem belonging to a low-dimensional subspace $V_N \subset V_h$ of dimension $N << N_h$. This is done by construction of a basis of V_N from basis function of truth space.

The reduced basis(ψ) is constructed by linear combination of FE basis function.

$$\psi = \phi B \tag{27}$$

Here the transformation matrix from finite element space to reduced basis space is defined by $B \in \mathbb{R}^{N_h X N}$. These reduced basis satisfy orthonormality criteria.

The reduced basis space is than specified by $\mathbb{B} = \text{span}\{\psi_1, \psi_2, ..., \psi_N\}$.

The approximation of function f in $\mathbb{B}$ is given by,

$$f_{rb} = \Sigma_{j=1}^{N} \psi_j C_j \tag{28}$$

Here, C_j is coefficient of each basis function.
For Galerkin orthogonality,

$$C_j = B^T * M * f \tag{29}$$

2.5.2 Offline online decomposition

[3]

During the offline stage the High Fidelity solution is calculated for suitable number, n_s snapshots. From these snapshots reduced basis space, reduced stiffness matrix and reduced right hand side are derived. The reduced basis is stored to be used during online stage.

During the online stage the affine decomposition is used to assemble reduced stiffness matrix and to assemble reduced right hand side. Subsequently, the problem is solved efficiently without using high-fidelity solution. This results in reduced memory and CPU time consumption and rapid evaluation of output at given parameter $\mu \in \mathbb{P}$.

2.5.3 Affine decomposition

The bilinear and linear form of (4) are transformed from reference domain to parametric domain. The terms in weak form are decomposed according to

the affine decompositions. In the fortunate case of affine parameter dependence the differential operator can be writen as L, the bilinear term a and linear functional (or right hand side) f for each subdomain can be expressed as,

$$a = \Sigma_{l=1}^{L_\Omega} a_i^l(u, v; \mu) \tag{30}$$

$$f = \Sigma_{l=1}^{L_\Omega} f_i^l(u, v; \mu) \tag{31}$$

The terms in the affine expansion of $a(u, v; \mu)$, θ_a, f(v)=(0,0) and θ_f for subdoain 1 and its boundary are expressed as,

$$a_0(u, v; \mu) = \int \nu * \partial u/\partial x_1 * \partial v/\partial x_1 * dx(1) \tag{32}$$

$$\theta_a^0 = (1 + ((\mu_1 - 0.5)^2 2)/(\mu_2^2)) * m2/0.3 \tag{33}$$

$$a_1(u, v; \mu) = \int \nu * \partial u/\partial x_2 * \partial v/\partial x_2 * dx(2) \tag{34}$$

$$\theta_a^1 = ((0.3/\mu_2)^2) * \mu_2/0.3 \tag{35}$$

$$a_2(u, v; \mu) = \int \nu * (\partial u/\partial x_1 * \partial v/\partial x_2 + \partial u/\partial x_2 * \partial v/\partial x_1) * dx(1) \tag{36}$$

$$\theta_a^2 = (0.3/m2 * (0.5 - \mu_1)/\mu_2) * \mu_2/0.3 \tag{37}$$

$$a_3(u, v, p, q; \mu) = \int (-p * \partial v_1/\partial x_1 + q * \partial u_1/\partial x_1) * dx(1) \tag{38}$$

$$\theta_a^3 = \mu_2/0.3 \tag{39}$$

$$a_4(u, v, p, q; \mu) = \int (-p * \partial v_1/\partial x_2 + q * \partial u_1/\partial x_2) * dx(1) \tag{40}$$

$$\theta_a^4 = 0 \tag{41}$$

$$a_5(u, v, p, q; \mu) = \int (-p * \partial v_2/\partial x_1 + q * \partial u_2/\partial x_1) * dx(1) \tag{42}$$

$$\theta_a^5 = -(\mu_1 - 0.5)/\mu_2 * \mu_2/0.3 \tag{43}$$

$$a_6(u, v, p, q; \mu) = \int (-p * \partial v_2/\partial x_2 + q * \partial u_2/\partial x_2) * dx(1) \tag{44}$$

$$\theta_a^6 = 1 \tag{45}$$

$$f(v) = \int (0 * v_1 + 0 * v_2) * ds(1) \tag{46}$$

$$\theta_f^0 = \mu_2/0.3 \tag{47}$$

2.5.4 Proper orthogonal decomposition

[3]

In the case of Proper Orthogonal Decomposition, n_s snapshots are used to derive reduced basis. We will use snapshot matrix $S \in \mathbb{R}^{N_h \times n_s}$. We also use here matrix M from (11),

$$V = S^T * M * S \tag{48}$$

The eigendecomposition of matrix V provides the eigenvectors ζ and eigen values λ. The eigenvalues are then sorted to take eigenvectors corresponding to highest eigenvalues in reduced basis space. The number of eigenvectors and eigenvalues are selected based on maximum dimension(N_{max}) of reduced basis space. It can also be selected based on the maximum affordable error.

An indication of reducibility of the problem is given by assessing the eigenvalues of the problem. A faster decay of eigenvalues indicates that only few eigen vectors retain most of the "energy" and the model can be reduced significantly. In contrast, slow decay of eigenvalues indicate less potential for reduction.

2.5.5 Error analysis

A good reduced order model is the one which reduces the error, measured as,

$$e(\mu)_{\mathbb{W}} = ||u - u_{rb}||_{\mathbb{W}} \tag{49}$$

where, $\mathbb{W}$ is suitable function space.

The error measurement gives the indication how well reduced space resembles to truth space. In the case of reduced order model, the error norm that are useful are the norm in truth space and also norm in Hilbert space such as L2 norm. The error in truth space is useful to assess the output evaluated from reduced solution and error in norm such as L2 is useful in case of assessing the approximation of solution.

The error in Truth Space in measured as,

$$e(\mu)_{\mathbb{V}} = ||u(\mu) - u_{rb}(\mu)||_{\mathbb{V}} \tag{50}$$

This is also referred as Energy norm.

Here, $\mathbb{V}$ corressponds to H^1 space for velocity and L^2 for pressure.

The error norm in L2 space is measured as ,

$$e(\mu)_{L2} = ||u(\mu) - u_{rb}(\mu)||_{L^2} \tag{51}$$

3 Implementation

The software framework used for the purpose is RBniCS[2] based on FEniCS[5]. The source code can be found in file SolveObstaclePodStokes.py at link:

https://github.com/niravshah241/stokes-flow-reduced-order-modelling

3.1 The Obstacle class

The Obstacle class has been derived from EllipticCoercivePODBase from standard RBniCS Library.

class Obstacle(ShapeParametrization(EllipticCoercivePODBase)):

The arguments to the class initialisation (in sequence) are the function space containing all the degrees of freedom, mesh of the domain, subdomains over which the mesh is divided, the boundary markers over the domain boundaries and list of boundary conditions.

def __init__(self, W, mesh, subd, bound, bc_list):

The subsequent steps store the copy of reference mesh in order to have copy during the mesh movement and domain deformation. The shape transformation expressions are the mapping from reference domain to deformed domain. The data specified on the reference mesh such as boundary_markers are also transformed onto deformed mesh. The subdomain and boundary measurements performed by Measure create data structure that is compatible during assembly of global matrix.

self.dx = Measure("dx")(subdomain_data=subd)
self.ds = Measure("ds")(subdomain_data=bound)

The compute_theta_a and compute_theta_f are multipliers for affine expansion, assemble_truth_a contains bilinear terms of finite element weak formulation and assemble_truth_f contains linear terms of finite element weak formaulation. The keep_diagonal argument ensures that the sparsed assembled matrix has necessary preallocated elements.

The error is computed by below function in both the norms explained in error analysis.

def compute_error(self, N=None, skip_truth_solve=False):

The error_analysis function is useful for summarising the error over the number of basis and each parameter and in turn helps to assess the quality of reduced results over truth results. This function is overwritten over equivalent function in RBniCS library to provide better graphical representation.

def error_analysis(self, N=None):

The parameter set is generated by the function generate_train_or_test_set. Argument n specifies number of parameter set and argument sampling specifies the probability distribution function for generating parameter set. This function is also overwritten over equivalent function in RBniCS standard library to add Gaussian distribution based sampling.

def generate_train_or_test_set(self, n, sampling):

3.2 The main file

In the main code the mesh, subdomains and boundary_markers are created by mesh generator mshr[4].

A mixed function space and boundary conditons are defined and given to Obstacle class as argument.

PETSc[1] solver is used due to its compatibility with RBniCS.

parameters.linear_algebra_backend = 'PETSc'

Subsequently, parameter range is specified for generating snapshots. These parameter can be generated either as per random distribution(by argument "random" below) or uniform distribution(by argument "linspace" in place of "random" below). Than the maximum dimension of reduced basis space is specified.

mu_range = [(0.15, 0.2), (0.2, 0.4)]
obstacle.setmu_range(mu_range)
obstacle.setxi_train(10, False, "random")
obstacle.setNmax(5)

After setting the parameter space for snapsots the offline phase is initiated with one gien parameter.

first_mu = (0.5, 0.3)
obstacle.setmu(first_mu)
obstacle.offline()

In similar manner, online phase is initiated.

online_mu = (0.5,0.3)
obstacle.setmu(online_mu)

obstacle.online_solve()

For error analysis the number of parameters and it's distribution are specified.

obstacle.setxi_test(10, False, "linspace")
obstacle.error_analysis()

In order to save and plot the solution the solution needs to be projected back onto individual space from which mixed function space $\mathbb{W}$ is created.

(u,p)=split(obstacle.snapshot)
u=project(u,VectorFunctionSpace(mesh, "Lagrange", 2, dim=2))
p=project(p,FunctionSpace(mesh, "Lagrange", 1))

The mesh is moved and reset to reference after plotting based on the parameter value corresspoding to the mesh.

obstacle.move_mesh() obstacle.reset_reference()

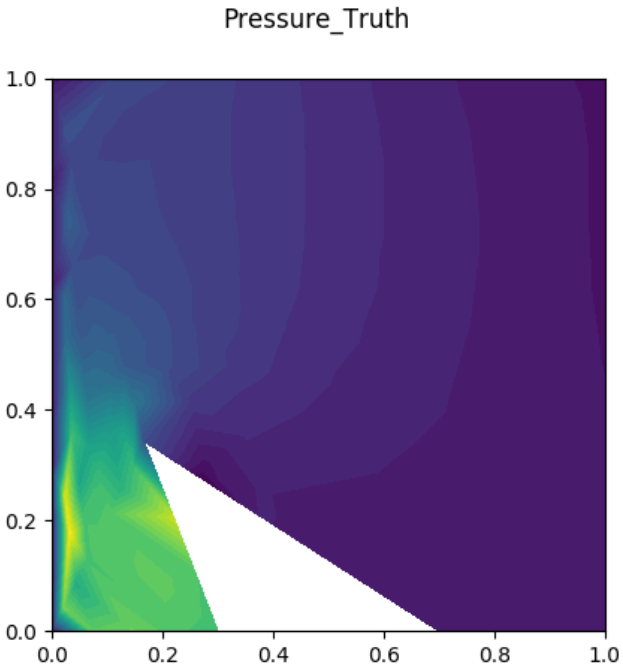

Figure 3: Pressure reconstruction from truth model

4 Results and discussion

The computed velocity and pressure can be found below in 3, 4, 5 and 6.

4.1 Parameter selection and parameter space enrichment

For the generation of snapshots, it is important to consider that the reduced basis solution has good accuracy over the whole parameter space. For this purpose the parameters for generation of reduced basis space and for error analysis are selected such that the selected parameters are deteremined in order to have uniform placing of parameters over the whole parametric space. Apart from the distribution the parameter space enrichment also plays crucial role in validating the model. In the present analysis minimum 30 parameters are used for contructing reduced basis space and minimum 50 parameters are used for error analysis.

Figure 7 and 8 show the L2 norm of error and energy norm of the error

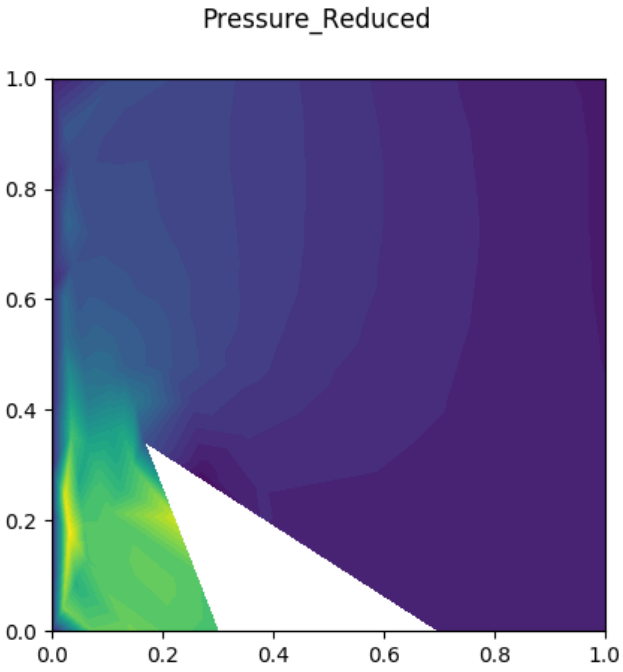

Figure 4: Pressure reconstruction reduced model

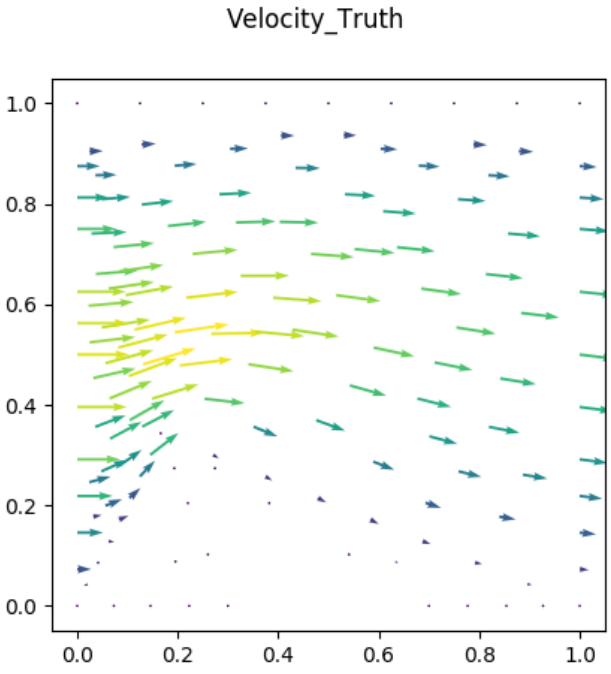

Figure 5: Velocity reconstruction from truth model

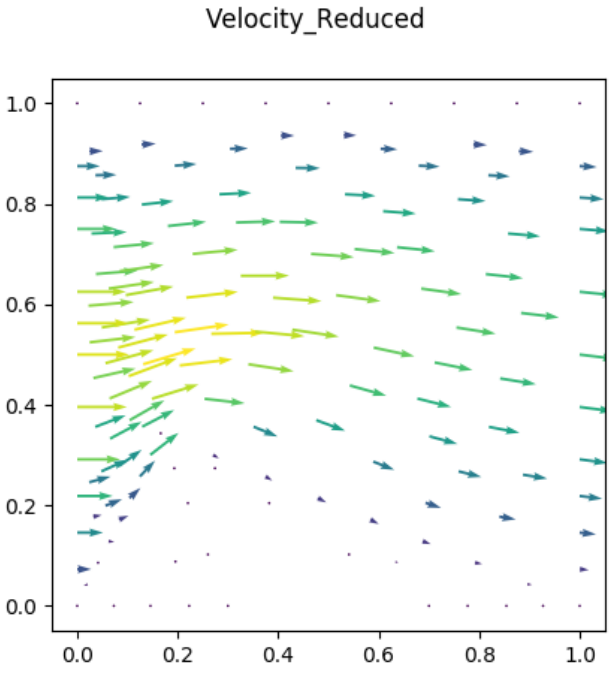

Figure 6: Velocity reconstruction from reduced model

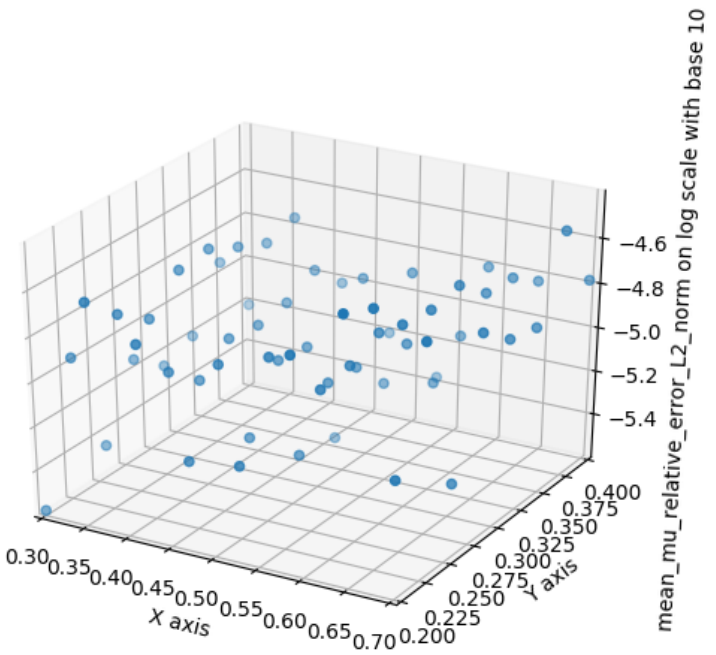

Figure 7: L2 norm of Error over parameter space

between truth solution and reduced order solution. The maximum error is of the order of $\mathbb{O}\left(10^{-6}\right)$.

4.2 Eigenvalue decay

The eigen values on logarithmic scale with respect to size of reduced basis space is shown in 9 shows rapid decay. The order of eigenvalue ranges from $\mathbb{O}(2)$ to less than $\mathbb{O}(-12)$ with number of snapshots less than 30. A consistent decrease also shows that any expansion of reduced basis space improves the accuracy of solution.

4.3 Simulation time

Figure 10 shows the time taken for online solution vs number of basis constituting redued basis space. It can be seen that with increase in size of reduced basis space the time taken for solving the problem by reduced basis method increases the time taken for online solution very marginally. The

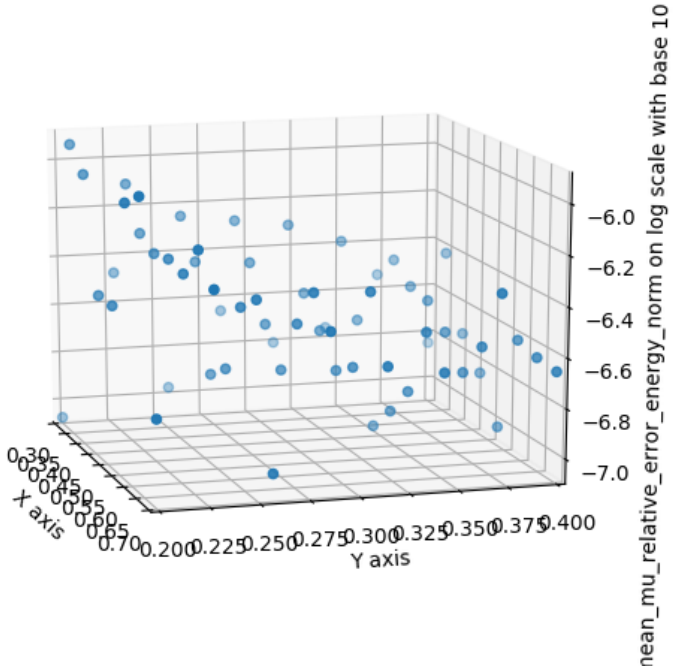

Figure 8: Energy norm of Error over parameter space

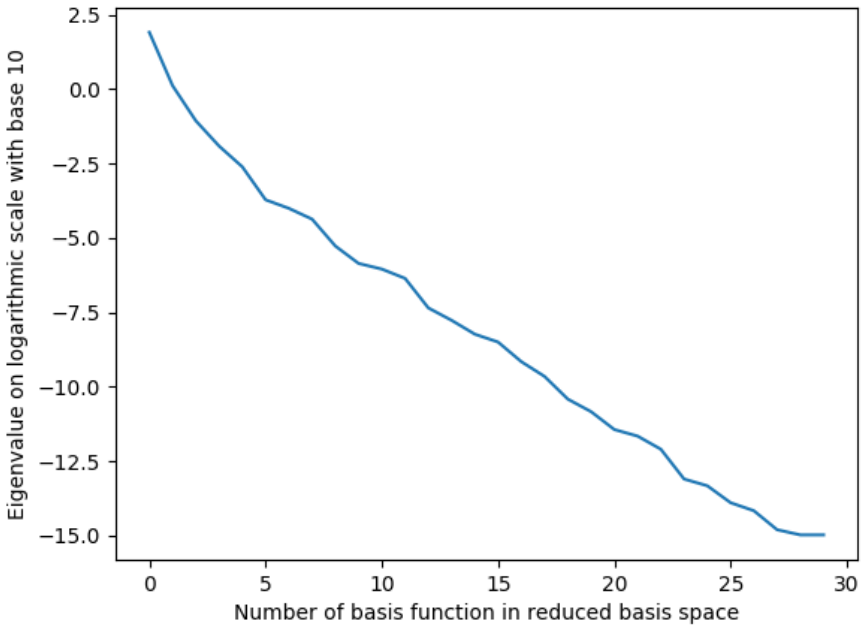

Figure 9: Eigenvalue decay

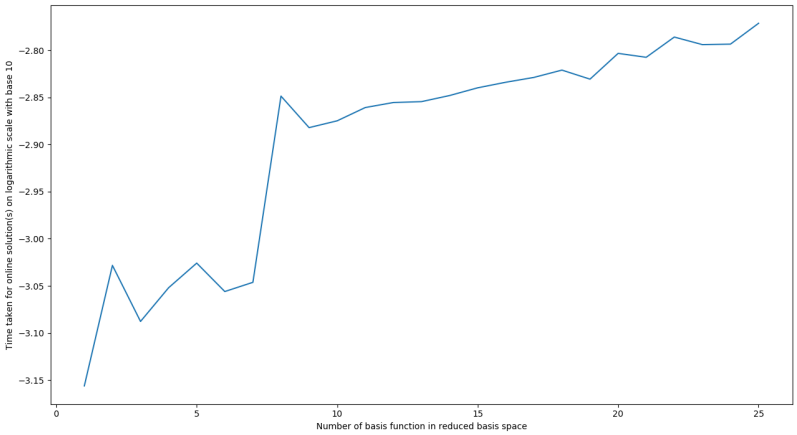

Figure 10: Time taken for Reduced basis approximation

time taken for same problem for full order solution is approximately 0.263 seconds. Compared to reduced basis solution with 25 basis functions the time taken by full order solution is approximately 155 times time taken by online solution.

4.4 Error analysis

Figure 11 represents the energy norm of error between reduced order solution and full order solution. The error in L2 norm is represented by Figure 12. The error is computed with reduced space derived from 30 snapshots and is compared against 60 set of parameters. A continuous decrease in L2 error is consistent with the continuous decay in eigenvalues of snapshot. A continuous decay in energy norm of error helps to predict required accuracy of solution to have pre-determined accuracy of output.

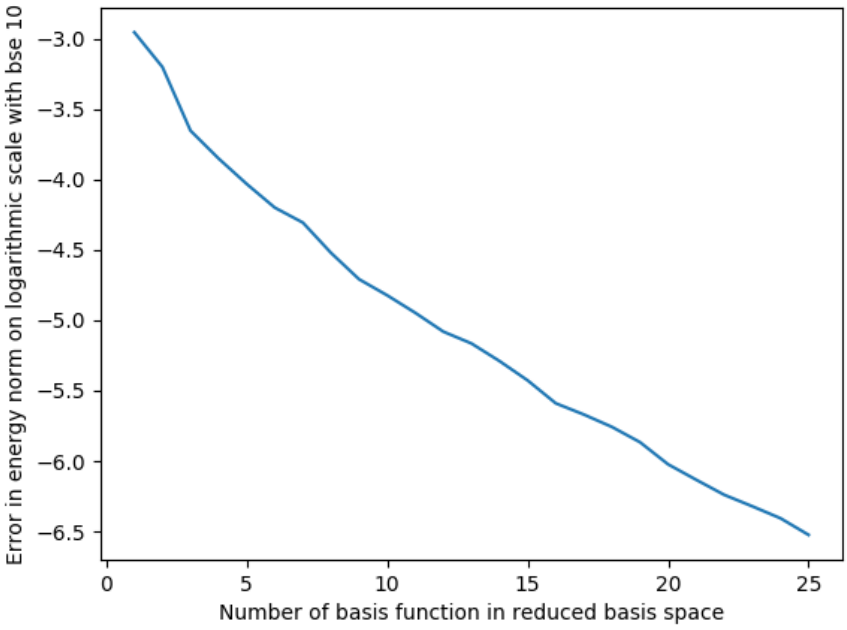

Figure 11: Error energy norm

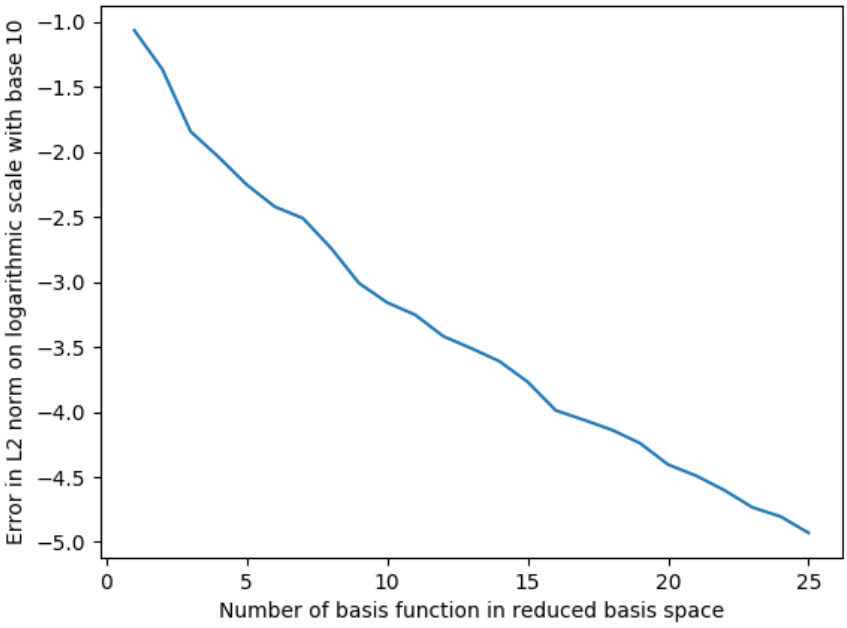

Figure 12: Error L2 norm

References

[1] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.7, Argonne National Laboratory, 2016.

[2] F Ballarin, A Sartori, and G Rozza. Rbnics–reduced order modelling in fenics. 2015.

[3] Fritzen F. Introduction to model order reduction of mechanial systems. Lecture Script, University of Stuttgart, 2016.

[4] Benjamin Kehlet. Mshr - mesh generation in fenics. Talk at FEniCS'14 workshop in Paris, June 2014.

[5] Anders Logg, Kent-Andre Mardal, and Garth N Wells. Automated solution of differential equations by the finite element method: The fenics book, 2012.

[6] Alfio Quarteroni, Gianluigi Rozza, and Andrea Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1(1):3, 2011.

[7] F.M. White. *Fluid Mechanics*. McGraw-Hill international editions. McGraw-Hill, 2003.