Discontinuous galerkin reduced basis approximation for direct numerical simulation of Navier Stokes equation: Master Thesis Report

Nirav Vasant Shah,
M.Sc. Candidate, Water Resources Engineering and Management (WAREM),
University of Stuttgart,
Stuttgart, Deutschland

Supervisor: Prof. Dr. Bernard Haasdonk,
Institute of Applied Analysis and Numerical Simulation,
Universitity of Stuttgart,
Stuttgart, Deutschalnd

Co-advisor: PI Prof. Gianluigi Rozza,
Scuola Internazionale Superiore di Studi Avanzati,
Trieste, Italy

Co-advisor: Dr. Martin Hess,
Scuola Internazionale Superiore di Studi Avanzati,
Trieste, Italy

October 23, 2017

Contents

1 Navier Stokes equation

1.1 Engineering perspective, physical domain and mathematical formulation

The subject of mathematical applications in fluid mechanics starts with one of the variants of Navier Stokes equation. Almost all processes of fluid mechanics require considerations related to Navier Stokes equation. Hence the importance of Navier Stokes equation is impossible to be ignored as far as mathematical approaches in fluid mechanics are concerned. The numerical method for incompressible equation is far more simple as compared to compressible Navier Stokes equation. This is due to density dependent constraint of fulfilling equation of state. Before proceeding further on deriving and understanding Navier Stokes equation we introduce notations related to the physical domain.

The domain is denoted by $\Omega \subseteq \mathbb{R}^d$. The domain is divided into subdomains known as 'Elements'. The subdomains boundaries are denoted by $\partial\Omega$. The domain boundaries are divided into Dirichlet and Neumann boundary. Apart from Dirichlet boundaries the subdomain boundaries are also considered in $\partial\Omega$ i.e. $\partial\Omega_D \cup \partial\Omega_N \subset \partial\Omega$, where, Dirichlet boundary is denoted by $\partial\Omega_D$ and Neumann boundary is denoted by $\partial\Omega_N$. The subdomain boudnaries which are not part of either Dirichlet boundary or Neumann boundary are internal or interelement boundaries. Also during consideration of jump across interface it is important to differentiate between elements on two sides of boundary for interelement flux formulations. The element contributing to flux is denoted by $\partial\Omega^+$ and the neighbouring element to which the flux is given is denoted by $\partial\Omega^-$. The subdomains are triangles i.e. triangular mesh is used. These triangles are denoted by τ.

The governing equations for incompressible stokes flow are conservation equations: Mass conservation and momentum conservation. The cnoservations 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. 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

If in the above equation B is substituted as momentum, correspondingly b as velocity we receive sum of external forces acting on the system. This sum of forces arise from stresses (shear stresses and normal stresses) and body forces such as weight. Equating external force with change in momentum and with the application of Gauss divergence theoreom we derive Navier Stokes equation,

$$-2\nabla.(\nu\nabla^s u) + (1/\rho)\nabla p + (u.\nabla)u = f \tag{2}$$

in Ω

The incompressible mass conservation equation can be written as

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

in Ω

The boundary conditions can be expressed as,

Dirichlet boundary:

$$u = u_D \tag{4}$$

on $\partial\Omega_d$

Neumann boundary:

$$-pn + 2\nu(n.\nabla^s)u = g \tag{5}$$

on $\partial\Omega_N$

Where,

u = flow velocity and $u : \Omega \to \mathbb{R}^d$

p = pressure and $p : \Omega \to \mathbb{R}$

ν and ρ = kinematic viscocity and density (fluid properties) and $\nu : \Omega \to \mathbb{R}$ and $\rho : \Omega \to \mathbb{R}$

f = external force $f : \mathbb{R}^d \to \mathbb{R}^d$

u_D = specified flow velocity at Dirichlet boundary $u_D : \partial\Omega_D \to \mathbb{R}^d$

n = normal unit vector $n : \partial\Omega \to \mathbb{R}^d$

g = specified Neumann flux $g : \partial\Omega_N \to \mathbb{R}^d$

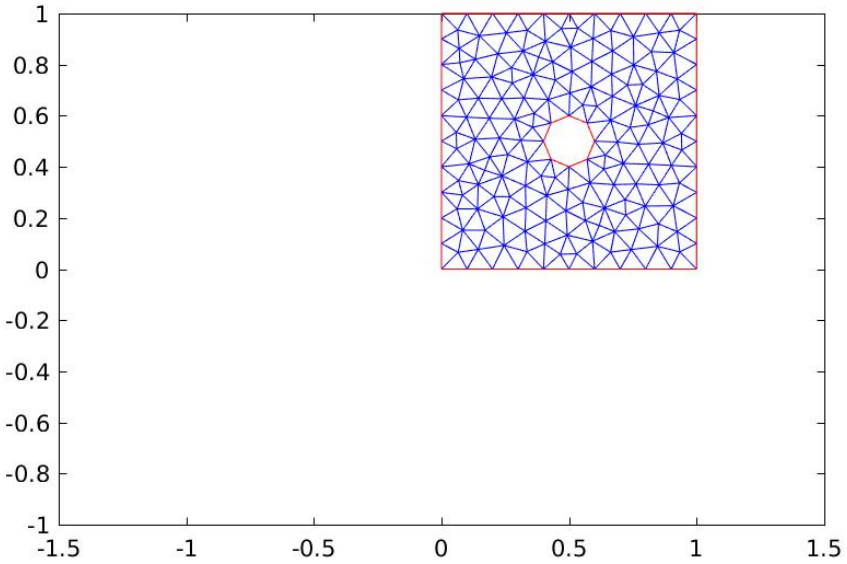


Figure 1: Domain

1.2 Direct numerical simulation

To be added

2 Description of physical Domain

to be shifted to numerical experiments

The domain considerd in present problem is a unit square with circular obstacle with center of circle coinciding with center of square.

3 Discontinuous Galerkin Method

In the context of discontinuous galerkin method we introduce function space $V(\tau_h)$ and $Q(\tau_h)$ for analytical solution of velocity and analytical solution of pressure respectively. The space containing high fidelity solution is called

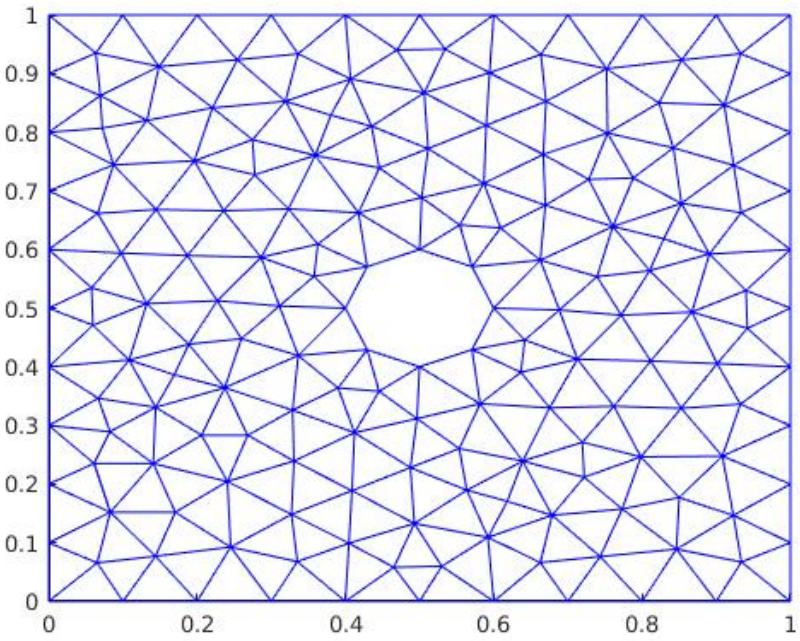


Figure 2: Mesh

truth space denoted by V_h. The dimension of V_h is denoted as N_h. (The subscript h hereafter refers to truth space.)

$$V = \{\phi \in L^2(\Omega) | v|_k \in P^D(K) \forall K \in \tau_h\} \tag{6}$$

$$Q = \{\psi \in L^2(\Omega) | q|_k \in P^{D-1}(K) \forall K \in \tau_h\} \tag{7}$$

Here, P^D denotes space of polynomials of degree at most D over Ω.

We apply similar procedure as in Finite element method i.e. multiplying the partial differential equation by test function and intergration by parts. However, we note that our test function is not continuous on the interface. Hence, we require flux approximations and jumps at the interface. These requirements have given rise to different discontinuous Galerkin methods. For explanation of each method we refer to literatures such as [2] for local discontinuous galerkin and [1] for Compact discontinuous Galerkin and Interior penalty method.

Discontinuous Galerkin methods for Navier Stokes equation were compared by [1]. The local discontinuous Galerkin(LDG) method extends the computational stencil beyond immediate neighbours whereas compact discontinuous Galerkin(CDG) and interior penalty method(IPM) only connect to neighbouring elements. The CDG method provides more flexibility with respect to stabilisation constant at the cost of additional simulation effort related to computation of lifting operator, while the IPM method requires restrictions on penalty parameter in order to maintain coercivity of bilinear form. Both methods, CDG and IPM, have almost similar convergenece rates.

LATER TO BE EDITED

3.1 Weak form of Navier Stokes compact discontinuous Galerkin formulation

The weak form of Navier Stokes equation for compact discontinuous galerkin method is given by

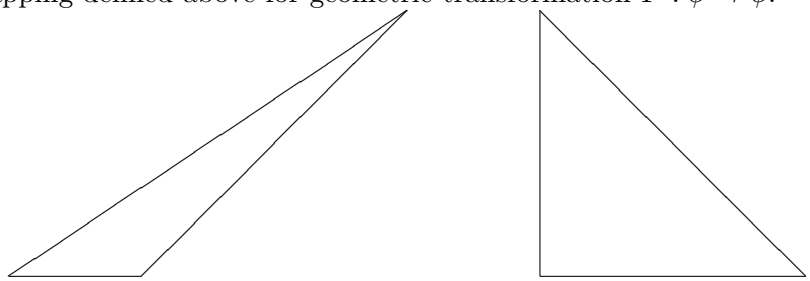
LATER TO BE EDITED

$$a = a \tag{8}$$

4 Global and local co-ordinate system

The integral terms of weak form are solved on a reference triangle instead of element itself. This helps to optimise the The co-ordinate system in which reference triangle lies is called reference or local co-ordinate system and the co-ordinate system in which element itself lies is called global co-ordinate system. In present analysis the triangle in local co-ordinate system has vertices (x, y) ,in order, as (0,0), (1,0) and (0,1). The geometric transformation is performed by defining a mapping $F : T \rightarrow \hat{T}$, where column vector $T = [x_1; y_1; x_2; y_2; x_3; y_3]$ holds co-ordinates of global element and column vector $\hat{T} = [\hat{x}_1; \hat{y}_1; \hat{x}_2; \hat{y}_2; \hat{x}_3; \hat{y}_3;]$ holds co-ordinates of local element. As mentioned previously $\hat{T} = [0; 0; 1; 0; 0; 1]$. (In the following chapters the image of geometry or function in global co-ordinate system onto local co-ordinate system is denoted by $\hat{}$)

We consider image of global function space on reference triangle. This function space is known as local basis function space. The local velocity function space is denoted by $\hat{V}$ and local velocity function space is denoted by $\hat{Q}$. The global basis function is transformed onto local basis function mapping defined above for geometric transformation $F : \phi \rightarrow \hat{\phi}$.



Global geometry (left) to Local geometry (right)

$$t : T(x) = C + GX \tag{9}$$

Here, G is 2×2 matrix and C is 2×1 matrix. X is co-ordinate of point in global co-ordinate system such that $X = (x; y)$. The corressponding point in local co-rodinate point is denoted by $\hat{X} = (\hat{x}; \hat{y})$.

The volume integral of function $f(x)$ is evaluated from integral on local element $f(\hat{x})$ as

$$\int_{\Omega} f(x)dx = \int_{\hat{\Omega}} f(\hat{x}) \det(G)d\hat{x} \tag{10}$$

For boundary integral of function $f(x)$, we consider boundary of unit length in local co-ordinate for boundatry of length l in global co-oridnate and transform the integral as,

$$\int_{\partial\Omega} f(x)ds = \int_{\partial\hat{\Omega}} f(\hat{x})ld\hat{s} \tag{11}$$

5 Jump and average operators

The jump operator of quantity u at an internal boundary is defined as,

$$[u] = u^+ \cdot n^+ + u^- \cdot n^- \tag{12}$$

where n is normal to the cell boundary pointing outwards from the cell.
Similarly the average operator is defined as,

$$\{u\} = \frac{u^+ + u^-}{2} \tag{13}$$

6 Implementation aspects

We introduce aditional notations from [1] for basis function for pressure as ψ and basis function for velocity as ϕ

Here, (p, q) refers to,

If p and q are scalars,

$$(p, q) = \int_\Omega pqd\Omega \tag{14}$$

If p and q are vectors,

$$(p, q) = \int_\Omega p \cdot qd\Omega \tag{15}$$

If p and q are tensors,

$$(p, q) = \int_\Omega p : qd\Omega \tag{16}$$

i.e. (.,.) denotes L_2 inner product.

The matrices from weak form of Navier Stokes equation have been assembled in 3 steps,

1. Evaluating function at vertex of local element and transform to global geometry

2. Performing integral of function over local element and transform to global geometry(if not or partially done in step 1)
3. Performing a loop over all elements and allocate integral at position in global matrix(for bilinear terms)/global vector(for linear terms) according to index of element degree of freedom in global degree of freedom vector.

6.1 Examples

In following examples we also use following variables:
J = Jacobian matrix
ndof = indices of degrees of freedom of an element in global degrees of freedom vector
JIT = Jacobian Inverse Traspose = $[J^{-1}]^T$ f = external force or right hand side in navier stokes equations's strong form n = normal to the boundary pointing outward of the element u_d = Dirichlet velocity Γ_d = Dirichlet boundary t = neumann value

1. $(\nabla\phi, \nabla\phi)$:

Files:
The global basis function for Velocity ϕ is mapped in local co-ordinate system by transformation : $\phi \to \hat{\phi}$.
Step 1: $res_1[i,j] = \hat{\phi_i}^T * JIT^T * JIT * \hat{\phi_j}$
Step 2: $res_2 = \int_{\hat{\Omega}} res_1 * \det(J) d\hat{\Omega}$
Step 3: Loop over each element and performing following operation in each loop res_3[ndof of element,ndof of element] = res_2

2. $(\psi, \nabla \cdot \phi)$:
source code : pressure_velocity_continuity_local , pressure_velocity_continuity_integral, pressure_velocity_continuity_assembly.m
Presure basis function ψ and velocity basis function ϕ are mapped onto local coordinate system.

Step 1: On local element following function is evaluated $res_1[i] = (\psi, \partial \cdot \phi)$ where $\phi = JIT * p\hat{h}i$ Step 2: $res_2 = \int_{\hat{\Omega}} res_1 * \det(J) d\hat{\Omega}$ Step 3: Loop over all elements and perform following operation in each loop res_3[ndof per element pressure,ndof per element velocity] = res_2
The transpose of above matrix is used in momentum conservation equation for (ϕ, ψ).

3. (f, ϕ):

Source code: source, source_integral, source_assembly
Step 1: $res_1[i] = \hat{\phi}_i^T * f(x)$
Step 2: $res_2 = \int_{\hat{\Omega}} res_1 * \det(J)$
Step 3: Loop over each element and performing following operation in each loop res_3[ndof of element] $= res_2$

NOTE: The external force in step 1 corressponding to global co-ordinates of point.

4. $(\psi, n \cdot u_d)_{\Gamma_d}$:

Source code: q_n_u_d, q_n_u_d integral, q_n_u_d assembly

Step 1: On local element following function is evaluated $res_1[i] = \hat{\psi}_i * n \cdot u_d(x)$ where, u_D from global co-ordinate of point x
Step 2: For each element and multiply by length of boundary to transform to global integral $res_2 = \int_{\Gamma_{delement}} res_1 |\Gamma_{delement}|$
Step 3: Loop over element having Dirichlet boundary and perform following operation in each loop res_3[ndof of element] $= res_2$

5. $(t, \phi)_{\Gamma_N}$:

source code: t_v, t_v_integral, t_v_assembly
Step 1: On local element following function is evaluated $res_1[i] = \hat{\phi}_i * t$
Step 2: An integral is performed over an element $res_2[i] = \int_{\Gamma_{Nelement}} res_1 |\Gamma_{Nelement}|$
Step 3: Loop over element having Neumann boundary and perform following operation in each loop res_3[ndof of element] $= res_2$

6. $(u_D, \phi)_{\Gamma_D}$:

source code: u_d_v, u_d_v_integral, u_d_v_assembly
Step 1: On local element following function is evaluated $res_1[i] = \hat{\phi}_i * u_D$
Step 2: An integral is performed over an element $res_2[i] = \int_{\Gamma_{Delement}} res_1 |\Gamma_{Delement}|$
Step 3: Loop over element having Dirichlet boundary and perform following operation in each loop res_3[ndof of element] $= res_2$
7. $(u_D, \phi)_{\Gamma_D}$:

source code: u_d_v, u_d_v_integral, u_d_v_assembly

Step 1: On local element following function is evaluated $res_1[i] = \hat{\phi}_i * u_D$
Step 2: An integral is performed over an element $res_2[i] = \int_{\Gamma_{Delement}} res_1 |\Gamma_{Delement}|$
Step 3: Loop over element having Dirichlet boundary and perform following operation in each loop res_3[ndof of element] $= res_2$

8. $(\phi, [[n \cdot \psi]])$

source code: pressure_average_velocity_basis_jump_local_plus, pressure_average_velocity_basis_jump_
pressure_average_velocity_basis_jump_integral_plus, pressure_average_velocity_basis_jump_integral_minu
pressure_average_velocity_basis_jump_assembly

The integral is implemented as $(\phi, n \cdot \psi^+) - (\phi, n \cdot \psi^-)$ where first term is integral on element itself and the second is integral on neghbouring element.

Step 1: On local element following function is evaluated on all elements $res_1^+[i] = \phi(n \cdot \psi^+)$. Step 2: The integration performed on the neighbouring element as $res_2^+[i] = \int_{\hat{\Omega}} res_1^+[i] det(J) d\hat{\Omega}$ Step 3: On local element following function is evaluated on all elements $res_1^-[i] = \phi(n \cdot \psi^-)$. Step 4: The integration performed on the neighbouring element as $res_2^-[i] = \int_{\hat{\Omega}} res_1^-[i] det(J) d\hat{\Omega}$ Step 5: The assembly is performed as res^+[ndof per element velocity, ndof per element pressure] $= res_2^+$ and res^-[ndof per element velocity, ndof per element pressure of neighbouring element] $= res_2^-$ and assembly matrix $res = res^+ + res^-$.

The same procedure is adopted for jump $(\psi, [[n \cdot \phi]])$ in continuity equation.

References

[1] A. Montlaur, S. Fernandez-Mendez, J. Peraire, and A. Huerta. Discontinuous galerkin methods for the navierstokes equations using solenoidal approximations. *International Journal for Numerical Methods in Fluids*, 64(5):549–564, 2010.

[2] P.-O. Persson, J. Bonet, and J. Peraire. Discontinuous galerkin solution of the navierstokes equations on deformable domains. *Computer Methods in Applied Mechanics and Engineering*, 198(17):1585 – 1595, 2009.