A STUDY OF MULTIGRID METHOD AS A SCALABLE PRECONDITIONER FOR SOLVING LINEAR SYSTEMS ARISING FROM FE DISCRETIZATION OF NS EQUATIONS

ADVANCED NUMERICAL ANALYSIS – PROJECT

INTRODUCTION

- In this project, a study of scalable preconditioners focussing on the Multi Grid Methods has been performed, where we discuss in detail its convergence independent of the mesh size, see the results of MG implementation to a simple Poisson problem & its usage as a preconditioner for solving saddle point linear systems arising from FE discretization of Navier Stokes equations.
- The following are the topics discussed:
  - A brief introduction to FE Discretization of the NS equations
  - Saddle Point formulation and its properties in brief
  - Multigrid Method
    - Introduction
    - Operators
    - Algorithms & Cycles
    - Properties, Theorems & Convergence
    - MG method implementation for a 2D FD discretized Poisson Problem
  - Use of Multigrid as Preconditioners for Stokes problem

FE DISCRETIZATION OF THE NS PROBLEM

- Mathematical Formulation:
  - The general Navier Stokes equation for incompressible flows is given as follows

$$\rho\frac{\partial u}{\partial t} - \mu\Delta u + \rho(u.\nabla)u + \nabla p = f; t > 0$$

$$Unsteady - Diffusion + Convection$$

$$u = u(x,t); p = p(x,t);\ \mathrm{x} \in\ \Omega;\ \Omega \in R^3$$

subjected to initial conditions $u(x,0) = u_0, \forall x \in \Omega;$

and boundary conditions $\forall t > 0, \quad u(x,t) = \varphi(x,t), \quad \forall x \in \Gamma_D, \quad t > 0 \quad \Rightarrow Dirichlet\ BC$

$$\left(v\frac{\partial u}{\partial n} - pn\right)(x,t) = \psi(x,t), \qquad \forall x\ \in\ \Gamma_N \Rightarrow Neumann\ BC$$

  - The incompressible continuity equation is given as

$$\nabla. u = 0; \qquad u = u(x,t); \qquad \mathrm{x} \in\ \Omega; t > 0$$

FE DISCRETIZATION OF THE NS PROBLEM

- Weak Formulation:

Definition of Spaces used in the formulation:

Space of square integrable functions : $L^2(\Omega) = \left\{ f : \Omega \longrightarrow R \middle| \int_\Omega |f(x)|^2 d\Omega < \infty \right\}$

Sobolov space: $H^k(\Omega) = \{f \in L^2 | D^\alpha f \epsilon L^2(\Omega) \quad \forall \alpha : \ |\alpha| \leq k\}$

The weak formulation is obtained by multiplying the stationary NS equation with a test function $\boldsymbol{v} \in \boldsymbol{V} \in \boldsymbol{H^1}(\boldsymbol{\Omega})$ and the continuing equation with test function $\boldsymbol{q} \in \boldsymbol{Q} \in \boldsymbol{L^2}(\boldsymbol{\Omega})$ **.** After applying integration by parts and Green's theorem, the weak formulation can be compactly expressed as

$$a(u,v) + c(u,u,v) + b(v,p) = f(u,v), \qquad \forall v \in V$$

$$b(u,q) = 0, \qquad \forall q \in Q$$

$$a(u,v) = \int_\Omega v \nabla u . \nabla v \, d\Omega; \quad c(u,u,v) = \int_\Omega \big((u.\nabla)u\big). v \, d\Omega; \quad b(u,q) = -\int_\Omega q(\nabla.u)\, d\Omega;$$

$$f(u,v) = \int_\Omega f.v \, d\Omega, \qquad \forall \mathrm{v} \in V, q \in Q$$

v, q are chosen such that they satisfy the Dirichilet BC.

Here, *a* & *b* are bilinear operators while *c* is a trilinear operator. For the Stokes problem, the *c(u,u,v)* disappears.

FE DISCRETIZATION OF THE NS PROBLEM

- Galerkin Approximation:

Building a numerical approximation from the weak formulation involves choosing finite spaces $V_h \subset V; \; Q_h \subset Q.$

Therefore choosing basis functions: $\varphi_i, \; i = \{1..N_v\}$ for $V_h \; \& \quad \psi_i, \; i = \{1..N_Q\}$ for Q_h .

The Stokes equation can be given as:

$$\sum_{j=1}^{N_V} u_j a(\varphi_j, \phi_i) + \sum_{j=1}^{N_Q} p_j b(\varphi_j, \psi_i) = (f, \varphi_i); \qquad \sum_{j=1}^{N_V} u_j b(\varphi_j, \psi_i) = 0,$$

$$A_{ij} = a(\varphi_j, \phi_i); \;\; B_{ij} = b(\varphi_j, \psi_i)$$

We choose the basis functions such that the emerging matrices A & B are sparse. For this we divide the domain into triangular elements and build basis functions for every node. The basis functions are chosen such that they have compact support i.e. φ_j has the value zero for all other elements other than j. We use piecewise polynomial functions for the basis functions with P_k ,k representing the degree of polynomial used.

FE DISCRETIZATION OF THE NS PROBLEM

- Stability & Convergence of Galerkin Approximation:

Theorem: The Galerkin approximation discussed before admits one and only one unique solution if the following conditions hold

1. $a(.,.) \textit{is continuous on } V_h \times V_h \textit{ with constant } \gamma$
2. $a(.,.) \textit{is coercive on } V_h^0 = \{v_h \in V_h : b(v_h, q_h) = 0, \forall q_h \in Q_h\}, \textit{with constant } \alpha$
3. $b(.,.) \textit{ is continous on } V_h \times Q_h, \textit{with constant } \delta$
4. $\textit{there exists a constant } \beta \textit{ s.t. } \inf_{q_h \epsilon Q_h} \sup_{v_h \epsilon V_h} \frac{b(v_h,q_h)}{\|v_h\|_{H^2}\|q_h\|_{L^2}} \geq \beta, \ \forall q_h \neq 0, v_h \neq 0$

With the convergence results

$$\|\mathbf{u} - \mathbf{u_h}\|_V \leq \left(1 + \frac{\delta}{\beta}\right)\left(1 + \frac{\gamma}{\alpha}\right) \inf_{\mathbf{v_h} \in V_h} \|\mathbf{u} - \mathbf{v_h}\|_V + \frac{\delta}{\alpha} \inf_{q_h \in Q_h} \|p - q_h\|_Q,$$

$$\|p - p_h\|_Q \leq \frac{\gamma}{\beta}\left(1 + \frac{\gamma}{\alpha}\right)\left(1 + \frac{\delta}{\beta}\right) \inf_{\mathbf{v_h} \in V_h} \|\mathbf{u} - \mathbf{v_h}\|_V + $$

$$+ \left(1 + \frac{\delta}{\beta} + \frac{\delta\gamma}{\alpha\beta}\right) \inf_{q_h \in Q_h} \|p - q_h\|_Q$$

The convergence results heavily depends on the inf – sup constant β, hence a small value for the same will loosen the bound on the solution. To ensure this we need to have the space V to have larger dimension the space Q. The simplest couple spaces which satisfy the inf-sup condition is the one which uses P_1 for Q_h and P_2 for V_h, known as Taylor hood approximation.

SADDLE POINT SYSTEMS & ITS PROPERTIES

- The FE discretization of the Stokes/NS equations gives rise to the linear system as follows:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix};$$

- With A & B having the block form $A = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, B = [B_1 \quad B_2]$ for the 2D problem.
- We define the Pressure & Velocity Mass matrices as

$$Q_{ij} = \int_{\Omega} \varphi_i \psi_j d\Omega; \qquad (Q_v)_{ij} = \int_{\Omega} \varphi_i \varphi_j d\Omega$$

  Both the matrices as s.p.d.

- The various matrices involved can be seen as operators:
  - A ⟶ Laplacian operator for Stokes, Convection Diffusion operator for NS
  - B ⟶ discrete divergence operator
  - B^T ⟶ discrete gradient operator
  - Q ⟶ discrete identity operator

SADDLE POINT SYSTEMS & ITS PROPERTIES

The properties of the saddle point systems can be given as follows:

- $\forall\, A \in R^{n\times n}, B \in R^{m\times n}, A$ is s.p.d for Stokes, while for NS, A is not symmetric but $\frac{1}{2}(A + A^T)$ is s.p.d
- The condition number of A: $\kappa(A) \propto h^{-2}$
- Analysing the generalized singular values of B, we obtain a bound for the inf – sup constant as:

  $\beta \leq \inf_{q\neq 0} \frac{(BA^{-1}B^T q,q)^{1/2}}{(Qq,q)^{1/2}} \equiv \sigma_{min}$, where $BA^{-1}B^T$ is known as the (Pressure)Schur complement.
- It can be proved that: for a problem with all Dirichilet BC, discretized using a stable approximation on a shape regular & quasi uniform subdivision of R^2, the mass matrix Q is spectrally equivalent to the schur complement and the inf-sup constant gets bounded as:

$$\beta^2 \leq \frac{(BA^{-1}B^T q, q)}{(Qq, q)} \leq 1, \forall q \neq 0, q \neq 1$$

- As a result of the spectral equivalence it can be shown that irrespective of the mesh size h , the matrix $Q^{-1}BA^{-1}B^T$ will always have eigenvalues on a very narrow interval, i.e. the condition number of $Q^{-1}BA^{-1}B^T$ can be bounded independently of h.
- The above bounds cannot be driven for matrix A as its condition number gets very high as the mesh is refined more and more.

MULTIGRID METHOD - INTRODUCTION

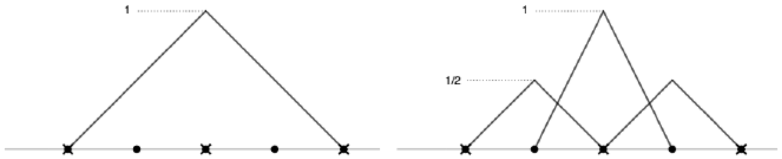
- Multigrid is a class of methods entirely different from projection methods like Krylov subspaces, with interesting properties like a convergence rate independent of the problem size.
- They are also well known as preconditioners which are scalable. This means, if we try to solve a linear system arising from the discretization of a PDE, with a mesh size h , the GMRES / MINRES, will converge in a few iterations which is independent of h.
- The main idea behind this method, is to solve a problem discretized using mesh size of h on a mesh size of $2h$. i.e. the problem is originally discretized using a fine mesh size h, but to solve we translate this discretized system onto a coarse mesh of size $2h$, with minimal loss of information and bring back the solution vector to the fine mesh size h.
- We have two sets of grids/mesh over here: a fine grid and a coarse grid. The method is to traverse between the grids using some operators that minimize the loss of information while traversing.

MULTIGRID - OPERATORS

- Let us define the spaces of coarse & fine grids as S_{2h} & S_h with basis functions ϕ_j^{2h} & ϕ_j^h and dimensions n_{2h} & n_h, respectively.
- Prolongation Operator:
  - Prolongation operator is used to traverse from the coarse mesh to the fine mesh.

    $\boldsymbol{I_{2h}^h} : \boldsymbol{S_{2h}} \longrightarrow \boldsymbol{S_h},$ **s.t.** $\phi_j^{2h} = \sum_i p_{ij}\, \phi_i^h$, where the coefficients p_{ij} have to determined.
  - Let $\boldsymbol{v_{2h}} \in S_{2h}$ & $\boldsymbol{v_h} \in S_h$, we want the operator to map the vector $\boldsymbol{v_{2h}}$ on to $\boldsymbol{v_h}$ s.t. $I_{2h}^h v_{2h} = v_h \implies \boldsymbol{S_{2h}} \subset \boldsymbol{S_h},$ i.e. we interpolate in such a manner that value of the function at the nodes doesn't change in the fine and coarse grids. Hence each function in S_{2h} is exactly represented in S_h

Figure 3.1: Basis function in S_{2h} and S_h.



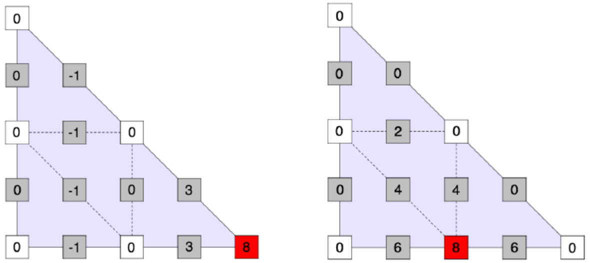
  - Hence the vector $\boldsymbol{v_{2h}}$ can be defined in the fine grid space as follows:

$$\boldsymbol{v_{2h}} = \sum_{\boldsymbol{j}} v_j^{2h} \phi_j^{2h} = \sum_{\boldsymbol{j}} v_j^{2h} \sum_i p_{ij}\, \phi_i^h = \sum_{\boldsymbol{i}}\, \phi_i^h \sum_j p_{ij}\, v_j^{2h} = \sum_{\boldsymbol{i}} v_i^h \phi_i^h; \quad \implies \boldsymbol{v_{2h}} = P\boldsymbol{v_h},$$

P is known as the prolongation matrix, with coefficient values $p_{ij},$ which are nothing but the values of the coarse grid basis functions on the fine grid nodes

MULTIGRID - OPERATORS

- The interpolation weights used to calculate the prolongation matrix heavily depends on the position of the node considered. Hence the prolongation matrix needs information from the underlying mesh. There are also Algebraic Multi Grid methods which can be used for non-hierarchical / unstructured grids which is beyond scope of this study.



(a) Vertex node (b) Mid-edge node

- Restriction Operator:
  - It is opposite of the prolongation operator; it restricts the fine grid to the coarse grid, represented as $\boldsymbol{I_h^{2h}} : \boldsymbol{S_h} \longrightarrow \boldsymbol{S_{2h}}$ & the restriction matrix $R = P^T$.
- Galerkin Coarse Grid operator:
  - With the prolongation and restriction operator the system $Ax = f$ in the fine grid can be represented in the coarse grid as

$$f_{2h} = R\, f_h = P^T \;\; f_h; \qquad Ax_h = f_h, \quad P^T Ax_h = P^T f_h, \quad P^T A\, P\, P^T x_h = f_{2h}, \quad P^T APx_{2h} = f_{2h}$$

$$\bar{A}\, x_{2h} = f_{2h}, \qquad \boldsymbol{\bar{A} = P^T AP}$$

  - $\boldsymbol{\bar{A}}$ is called the Galerkin Coarse grid operator, and it represents the matrix A in the coarse grid.

MULTIGRID - OPERATORS

- Smoothing Operator:
  - As the space S_{2h} is included in the space S_h , we can write $S_h = S_{2h} + B_h$, where B_h is the component lost while restricting from the fine space to the coarse space.
  - To minimize the loss, we apply smoothing operators, which are generally stationery iterative methods like Jacobi or Gauss Siedel. By applying the smoother, the error at the k-th iteration is e^k is given as $e^k = (I - M^{-1} A)^k e^0$.
  - With a couple of iterations the component of error in B_h tends to 0, hence the error e^k can be restricted to the coarse grid with very little loss of information. As the error e^k cannot be directly computed we use the residual:

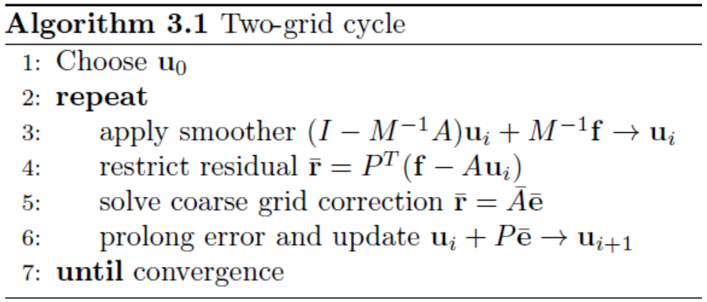
    $$r^k = f - Ax^k = Ae^k,$$

    And the restricted residual is calculated as $\overline{r} = P^T r = P^T Ae = P^T AP\,\overline{e} = \bar{A}\,\overline{e}$
  - Choosing the proper smoother depends on the type of the problem and forms a very important

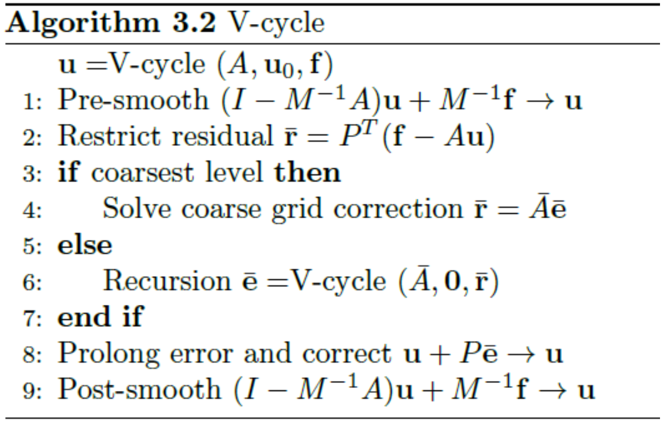
MULTIGRID – ALGORITHM & CYCLES

- Basic Algorithm:

  Summarizing the use of all operators in the algorithm is given as follows:

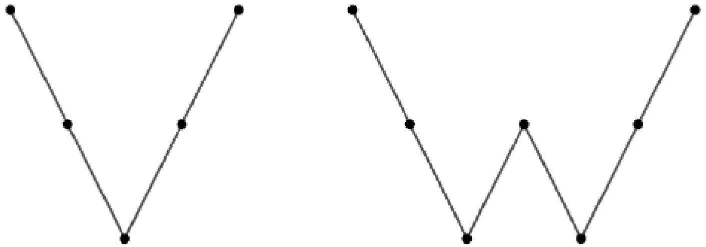


```
Algorithm 3.1 Two-grid cycle
Choose $\mathbf{u}_0$
repeat
    apply smoother $(I - M^{-1}A)\mathbf{u}_i + M^{-1}\mathbf{f} \to \mathbf{u}_i$
    restrict residual $\bar{\mathbf{r}} = P^T(\mathbf{f} - A\mathbf{u}_i)$
    solve coarse grid correction $\bar{\mathbf{r}} = \bar{A}\bar{\mathbf{e}}$
    prolong error and update $\mathbf{u}_i + P\bar{\mathbf{e}} \to \mathbf{u}_{i+1}$
until convergence
```



```
Algorithm 3.2 V-cycle
$\mathbf{u}$ =V-cycle $(A, \mathbf{u}_0, \mathbf{f})$
Pre-smooth $(I - M^{-1}A)\mathbf{u} + M^{-1}\mathbf{f} \to \mathbf{u}$
Restrict residual $\bar{\mathbf{r}} = P^T(\mathbf{f} - A\mathbf{u})$
if coarsest level then
    Solve coarse grid correction $\bar{\mathbf{r}} = \bar{A}\bar{\mathbf{e}}$
else
    Recursion $\bar{\mathbf{e}}$ =V-cycle $(\bar{A}, \mathbf{0}, \bar{\mathbf{r}})$
end if
Prolong error and correct $\mathbf{u} + P\bar{\mathbf{e}} \to \mathbf{u}$
Post-smooth $(I - M^{-1}A)\mathbf{u} + M^{-1}\mathbf{f} \to \mathbf{u}$
```

- V cycles & W cycles:
  - The modification to the basic algorithm comes two main things: a) restrict until the coarsest grid is reached; b) apply the smoothing operator symmetrically, i.e. Pre-smooth (before restricting the residual) and Post- smooth prolongation & correction of the error.
  - The V & W cycles vary slightly with the W cycle consuming more computational power but being more accurate. A schematic diagram of the V & W cycles can be seen as follows:



MULTIGRID – PROPERTIES, THEOREMS & CONVERGENCE

- The upcoming properties are derived for the standard Poisson($-\Delta u = f$; only diffusion $\Longrightarrow \cong$ Stokes problem) problem, and they all rely on the following assumptions:
  1. The grids used are quasi-uniform;
  2. They are made of shape regular elements;
  3. The problem considered is H^2 regular;
- Some useful Lemmas:
  1. If linear finite elements are used to solve a Poisson problem and u is regular enough, then there exists a constant C s.t.

     $$\|\nabla(u_h - u)\|_{L^2} \leq C\, h\|u\|_{H^2}.$$

     Combining this Lemma with H^2 regularity, the L^2 norm of $\|\nabla(u_h - u)\|$ can be controlled by the L^2 norm of $\|f\|$
  2. The norm of $\boldsymbol{f}$ (coefficient vector representing the function f in FE space) can be controlled b the norm of f with a constant C as:

     $$h\|f\|_{L^2} \leq C\|\boldsymbol{f}\|$$
- Using these Lemma's we now show as follows that the Galerkin approximation and Smoothing can be bounded independently of mesh size h.
- ***Approximation property:*** *Under the previously stated assumptions, the following relation holds:*

  $$\|(A^{-1} - P\bar{A}^{-1}P^T)\boldsymbol{y}\|_A \leq C\|\boldsymbol{y}\|, \quad \forall y \in \mathbb{R}^n$$

  With the constant C independent of h.

$$\bar{A} = P^T A P$$
$$\bar{A}^{-1} = P^{-1}A^{-1}P^{T^{-1}}$$
$$P\,\bar{A}^{-1}P^T = A^{-1}$$

MULTIGRID – PROPERTIES, THEOREMS & CONVERGENCE

- ***Smoothing property:*** *Consider a smoother with* $M = \theta I, \forall \theta \in \mathbb{R}$ *and with the eigenvalues of* $(I - M^{-1} A)$ *in the interval* $[-\sigma, 1]$ *with* $0 \leq \sigma < 1$, *independently of* h. *The following approximation property holds*

$$\left\| A(I - M^{-1} A)^k \boldsymbol{y} \right\|_A \leq \eta(k) \|\boldsymbol{y}\|_A, \quad \forall y \in \mathbb{R}^n$$

where $\eta(k) \longrightarrow 0, as\ k \longrightarrow \infty$

- Proof:

Let $(I - M^{-1} A) = I - A/\theta$; $\boldsymbol{z_i}, \lambda_i$ are the orthogonal eigenvectors and its corresponding eigenvalues of A**.** Hence, $\left(I - \frac{A}{\theta}\right) \boldsymbol{z_i} = \boldsymbol{z_i} \lambda_i$ and any $\boldsymbol{y}$ can be written as $\boldsymbol{y} = \sum_i c_i \boldsymbol{z}_i$**.**

$$\left\| A(I - M^{-1} A)^k \boldsymbol{y} \right\|^{\mathbf{2}} = \left\| \sum_i c_i A(I - A/\theta)^k \boldsymbol{z_i} \right\|^{\mathbf{2}} = \left\| \sum_i c_i A \lambda_i^k \boldsymbol{z_i} \right\|^{\mathbf{2}} = \left\| \sum_i c_i \lambda_i^k \theta (1 - \lambda_i) \boldsymbol{z_i} \right\|^{\mathbf{2}}$$

$$= \sum_i c_i^2 \lambda_i^{2k}\, \theta^2\, (1 - \lambda_i)^2 (\boldsymbol{z_i}, \boldsymbol{z}_i) \leq \max_{\boldsymbol{\lambda} \in [-\sigma, 1]} (\lambda_i^{2k} (1 - \lambda_i))\, \theta \sum_i c_i^2\, \theta\, (1 - \lambda_i)\, (\boldsymbol{z_i}, \boldsymbol{z}_i)$$

The quantity $\lambda^{2k}(1 - \lambda)$ attains maximum when $\lambda = -\sigma\ \ or\ \lambda = 2k/(2k + 1)$

$$\left\| A(I - M^{-1} A)^k \boldsymbol{y} \right\|^{\mathbf{2}} \leq \max_{\boldsymbol{\lambda} \in [-\sigma, 1]} \left[\frac{1}{2ke}, \sigma^{2k}(1 + \sigma) \right] \theta \sum_i c_i^2\, \theta\, (1 - \lambda_i)\, (\boldsymbol{z_i}, \boldsymbol{z}_i)$$

$$\leq \max_{\boldsymbol{\lambda} \in [-\sigma, 1]} \left[\frac{\theta}{2ke}, \theta \sigma^{2k}(1 + \sigma) \right] (A\boldsymbol{y}, \boldsymbol{y}) = \max_{\boldsymbol{\lambda} \in [-\sigma, 1]} \left[\frac{\theta}{2ke}, \theta \sigma^{2k}(1 + \sigma) \right] \|\boldsymbol{y}\|_A^{\mathbf{2}}$$

$$\text{where} \quad \eta(k) = \max_{\boldsymbol{\lambda} \in [-\sigma, 1]} \left[\frac{\theta}{2ke}, \theta \sigma^{2k}(1 + \sigma) \right]$$

MULTIGRID – PROPERTIES, THEOREMS & CONVERGENCE

- ***Convergence of Multigrid****: If both the approximation & smoothing property hold, then the two-grid cycle converges & contraction rate is independent of h.*
- Proof:

For the convergence of the method, we study the behaviour of $\boldsymbol{e}_{i+1} = \boldsymbol{u} - \boldsymbol{u}_{i+1}$,

Let $\boldsymbol{u}_{\boldsymbol{s}}$ & $\boldsymbol{u}_{\boldsymbol{s}-}$ be the result of k & $k-1$ smoothing steps on the current iterate $\boldsymbol{u}_i$

$$\boldsymbol{u}_{i+1} = \boldsymbol{u}_{\boldsymbol{s}} + P\bar{\boldsymbol{e}} = \boldsymbol{u}_{\boldsymbol{s}} + P\,\bar{A}^{-1}\bar{r} = \boldsymbol{u}_{\boldsymbol{s}} + P\,\bar{A}^{-1}P^T(\boldsymbol{f} - A\,\boldsymbol{u}_s) = \boldsymbol{u}_{\boldsymbol{s}} + P\,\bar{A}^{-1}P^T A(\boldsymbol{u} - \boldsymbol{u}_s); \qquad \boldsymbol{u}_{\boldsymbol{s}} = (I - M^{-1}A)\;\boldsymbol{u}_{\boldsymbol{s}-} + M^{-1}\boldsymbol{f}$$

$$\boldsymbol{u} - \boldsymbol{u}_{\boldsymbol{s}} = \boldsymbol{u} - [(I - M^{-1}A)\;\boldsymbol{u}_{\boldsymbol{s}-} + M^{-1}A\boldsymbol{u}] = (I - M^{-1}A)(\boldsymbol{u} - \boldsymbol{u}_{\boldsymbol{s}-}) = (I - M^{-1}A)^k(\boldsymbol{u} - \boldsymbol{u}_{\boldsymbol{i}}) = (I - M^{-1}A)^k\boldsymbol{e}_i$$

$$\boldsymbol{u}_{i+1} = \boldsymbol{u} - (I - M^{-1}A)^k\boldsymbol{e}_i + P\,\bar{A}^{-1}P^T A(I - M^{-1}A)^k\boldsymbol{e}_i$$

$$\boldsymbol{e}_{i+1} = \boldsymbol{u} - \boldsymbol{u}_{i+1} = (A^{-1} - P\,\bar{A}^{-1}P^T)A(I - M^{-1}A)^k\,\boldsymbol{e}_i$$

From the approximation and smoothing property we obtain the following:

$$\|\boldsymbol{e}_{i+1}\|_A = \left\|(A^{-1} - P\,\bar{A}^{-1}P^T)A(I - M^{-1}A)^k\,\boldsymbol{e}_i\right\|_A \le C\left\|A(I - M^{-1}A)^k\,\boldsymbol{e}_i\right\|_A \le C\eta(k)\|\boldsymbol{e}_i\|_A$$

Since $\eta(k) \to 0$ as $k \to \infty$, the error at the $i+1$ iteration after applying some $\bar{k}$ iterations can be written as $\|\boldsymbol{e}_{i+1}\|_A \le \gamma\|\boldsymbol{e}_i\|_A$, where γ is lesser than 1 & independent of h .

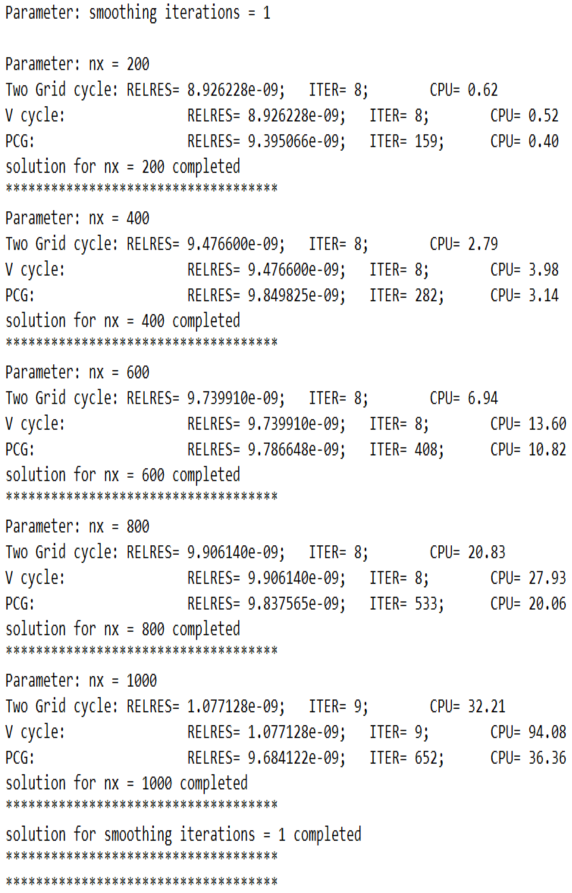
MULTIGRID – PROPERTIES, THEOREMS & CONVERGENCE

- The previously derived convergence relies on the fact that the eigenvalues of the smoothing matrix are bounded away from -1, independently of h.
- Wkt, the stationary iterative methods converge only when the spectral radius of the iteration matrix is ≤ 1, which implies that even one eigenvalue can tend to -1. Hence, the smoother should be chosen very carefully.
- Hence, with a proper choice of smoother, multigrid can produce a convergence with same number of iterations independent of the problem size. This implies that the growth of computational time will be linear wrt the no. of nodes of the mesh.
- This property is referred to as *Scalability*.

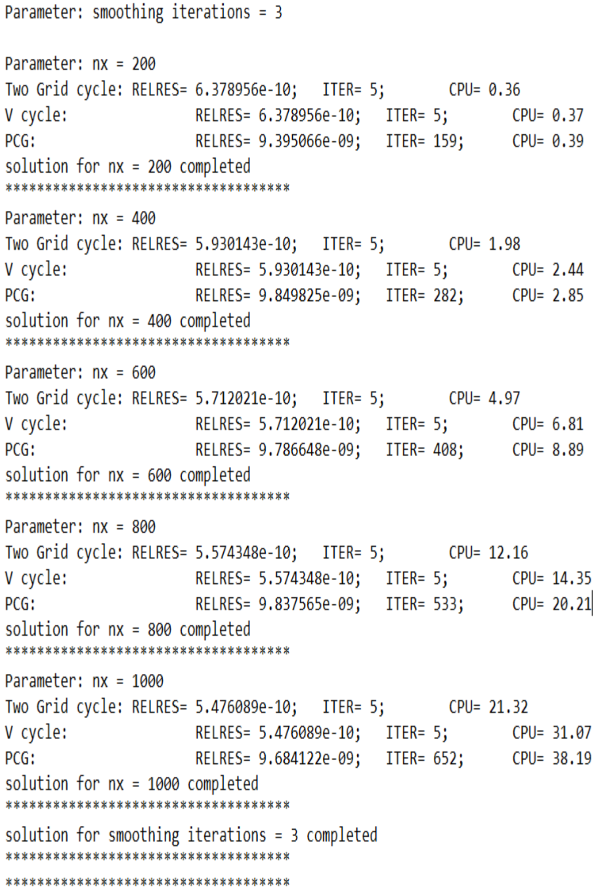
MULTIGRID AS A SOLVER FOR POISSON PROBLEM WITH FD DISCRETIZATION

- To understand the working of multigrid methods, a simple implementation of the multigrid method as solver for the standard 2D Poisson problem discretized using Finite Difference method has been done.
- The two-grid cycle and V cycle are compared with PCG solver (IC preconditioning) for various mesh sizes(internal nodes from 200 to 1000)
- The selection of smoother is very important over here. When a Jacobi smoother was selected, the multigrid method failed to converge, while with the GS preconditioner the multigrid methods converge independent of the mesh size.
- The number of iterations and time taken depends on the number of smoothing iterations applied during each iteration of the MG method.
- The results are provided in the next slides.

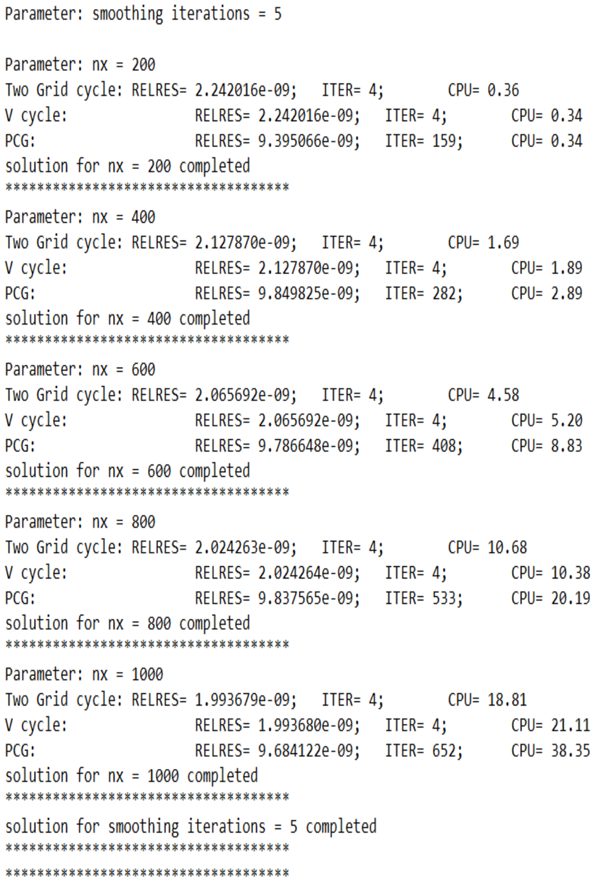
MULTIGRID AS A SOLVER FOR POISSON PROBLEM WITH FD DISCRETIZATION



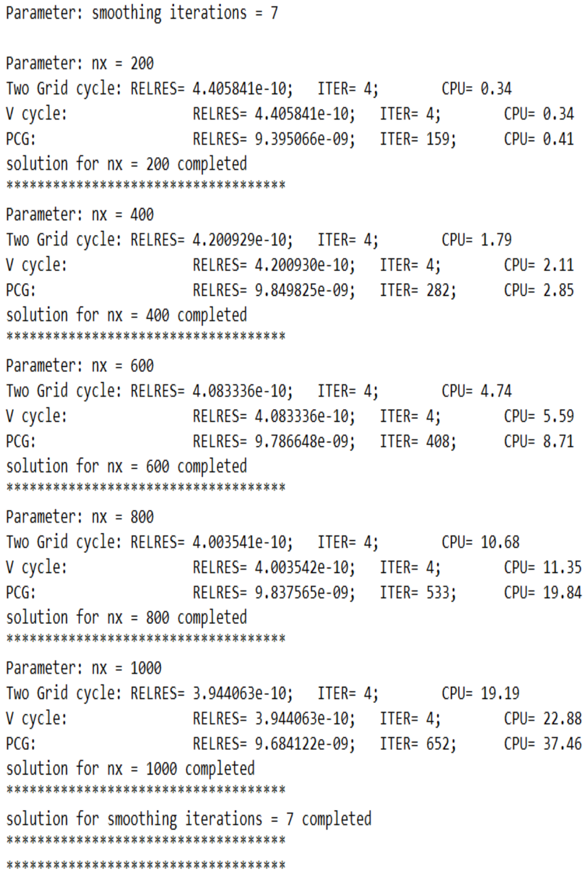
```
Parameter: smoothing iterations = 1

Parameter: nx = 200
Two Grid cycle: RELRES= 8.926228e-09;   ITER= 8;        CPU= 0.62
V cycle:                RELRES= 8.926228e-09;   ITER= 8;        CPU= 0.52
PCG:                    RELRES= 9.395066e-09;   ITER= 159;      CPU= 0.40
solution for nx = 200 completed
************************************

Parameter: nx = 400
Two Grid cycle: RELRES= 9.476600e-09;   ITER= 8;        CPU= 2.79
V cycle:                RELRES= 9.476600e-09;   ITER= 8;        CPU= 3.98
PCG:                    RELRES= 9.849825e-09;   ITER= 282;      CPU= 3.14
solution for nx = 400 completed
************************************

Parameter: nx = 600
Two Grid cycle: RELRES= 9.739910e-09;   ITER= 8;        CPU= 6.94
V cycle:                RELRES= 9.739910e-09;   ITER= 8;        CPU= 13.60
PCG:                    RELRES= 9.786648e-09;   ITER= 408;      CPU= 10.82
solution for nx = 600 completed
************************************

Parameter: nx = 800
Two Grid cycle: RELRES= 9.906140e-09;   ITER= 8;        CPU= 20.83
V cycle:                RELRES= 9.906140e-09;   ITER= 8;        CPU= 27.93
PCG:                    RELRES= 9.837565e-09;   ITER= 533;      CPU= 20.06
solution for nx = 800 completed
************************************

Parameter: nx = 1000
Two Grid cycle: RELRES= 1.077128e-09;   ITER= 9;        CPU= 32.21
V cycle:                RELRES= 1.077128e-09;   ITER= 9;        CPU= 94.08
PCG:                    RELRES= 9.684122e-09;   ITER= 652;      CPU= 36.36
solution for nx = 1000 completed
************************************

solution for smoothing iterations = 1 completed
************************************

************************************
```



```
Parameter: smoothing iterations = 3

Parameter: nx = 200
Two Grid cycle: RELRES= 6.378956e-10;   ITER= 5;        CPU= 0.36
V cycle:                RELRES= 6.378956e-10;   ITER= 5;        CPU= 0.37
PCG:                    RELRES= 9.395066e-09;   ITER= 159;      CPU= 0.39
solution for nx = 200 completed
************************************

Parameter: nx = 400
Two Grid cycle: RELRES= 5.930143e-10;   ITER= 5;        CPU= 1.98
V cycle:                RELRES= 5.930143e-10;   ITER= 5;        CPU= 2.44
PCG:                    RELRES= 9.849825e-09;   ITER= 282;      CPU= 2.85
solution for nx = 400 completed
************************************

Parameter: nx = 600
Two Grid cycle: RELRES= 5.712021e-10;   ITER= 5;        CPU= 4.97
V cycle:                RELRES= 5.712021e-10;   ITER= 5;        CPU= 6.81
PCG:                    RELRES= 9.786648e-09;   ITER= 408;      CPU= 8.89
solution for nx = 600 completed
************************************

Parameter: nx = 800
Two Grid cycle: RELRES= 5.574348e-10;   ITER= 5;        CPU= 12.16
V cycle:                RELRES= 5.574348e-10;   ITER= 5;        CPU= 14.35
PCG:                    RELRES= 9.837565e-09;   ITER= 533;      CPU= 20.21
solution for nx = 800 completed
************************************

Parameter: nx = 1000
Two Grid cycle: RELRES= 5.476089e-10;   ITER= 5;        CPU= 21.32
V cycle:                RELRES= 5.476089e-10;   ITER= 5;        CPU= 31.07
PCG:                    RELRES= 9.684122e-09;   ITER= 652;      CPU= 38.19
solution for nx = 1000 completed
************************************

solution for smoothing iterations = 3 completed
************************************

************************************
```

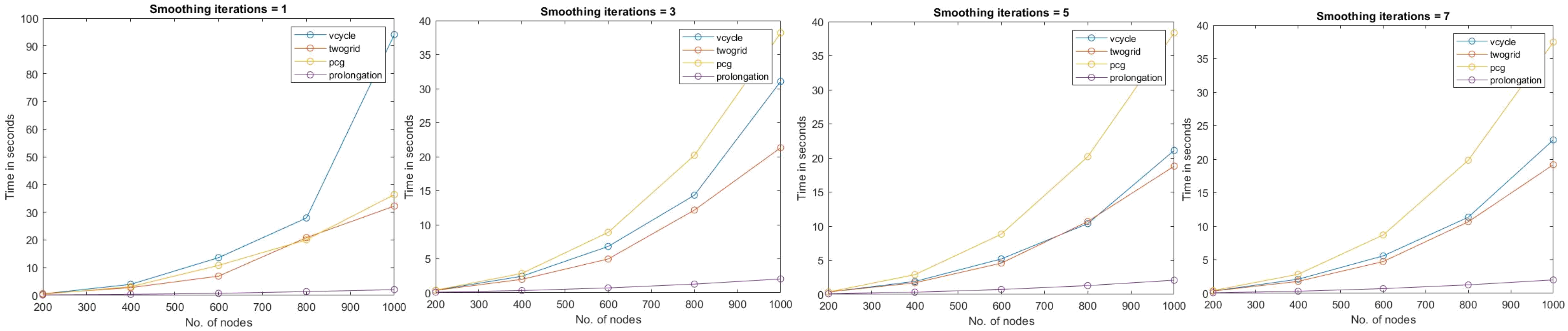


```
Parameter: smoothing iterations = 5

Parameter: nx = 200
Two Grid cycle: RELRES= 2.242016e-09;   ITER= 4;        CPU= 0.36
V cycle:                RELRES= 2.242016e-09;   ITER= 4;        CPU= 0.34
PCG:                    RELRES= 9.395066e-09;   ITER= 159;      CPU= 0.34
solution for nx = 200 completed
************************************

Parameter: nx = 400
Two Grid cycle: RELRES= 2.127870e-09;   ITER= 4;        CPU= 1.69
V cycle:                RELRES= 2.127870e-09;   ITER= 4;        CPU= 1.89
PCG:                    RELRES= 9.849825e-09;   ITER= 282;      CPU= 2.89
solution for nx = 400 completed
************************************

Parameter: nx = 600
Two Grid cycle: RELRES= 2.065692e-09;   ITER= 4;        CPU= 4.58
V cycle:                RELRES= 2.065692e-09;   ITER= 4;        CPU= 5.20
PCG:                    RELRES= 9.786648e-09;   ITER= 408;      CPU= 8.83
solution for nx = 600 completed
************************************

Parameter: nx = 800
Two Grid cycle: RELRES= 2.024263e-09;   ITER= 4;        CPU= 10.68
V cycle:                RELRES= 2.024264e-09;   ITER= 4;        CPU= 10.38
PCG:                    RELRES= 9.837565e-09;   ITER= 533;      CPU= 20.19
solution for nx = 800 completed
************************************

Parameter: nx = 1000
Two Grid cycle: RELRES= 1.993679e-09;   ITER= 4;        CPU= 18.81
V cycle:                RELRES= 1.993680e-09;   ITER= 4;        CPU= 21.11
PCG:                    RELRES= 9.684122e-09;   ITER= 652;      CPU= 38.35
solution for nx = 1000 completed
************************************

solution for smoothing iterations = 5 completed
************************************

************************************
```



```
Parameter: smoothing iterations = 7

Parameter: nx = 200
Two Grid cycle: RELRES= 4.405841e-10;   ITER= 4;        CPU= 0.34
V cycle:                RELRES= 4.405841e-10;   ITER= 4;        CPU= 0.34
PCG:                    RELRES= 9.395066e-09;   ITER= 159;      CPU= 0.41
solution for nx = 200 completed
************************************

Parameter: nx = 400
Two Grid cycle: RELRES= 4.200929e-10;   ITER= 4;        CPU= 1.79
V cycle:                RELRES= 4.200930e-10;   ITER= 4;        CPU= 2.11
PCG:                    RELRES= 9.849825e-09;   ITER= 282;      CPU= 2.85
solution for nx = 400 completed
************************************

Parameter: nx = 600
Two Grid cycle: RELRES= 4.083336e-10;   ITER= 4;        CPU= 4.74
V cycle:                RELRES= 4.083336e-10;   ITER= 4;        CPU= 5.59
PCG:                    RELRES= 9.786648e-09;   ITER= 408;      CPU= 8.71
solution for nx = 600 completed
************************************

Parameter: nx = 800
Two Grid cycle: RELRES= 4.003541e-10;   ITER= 4;        CPU= 10.68
V cycle:                RELRES= 4.003542e-10;   ITER= 4;        CPU= 11.35
PCG:                    RELRES= 9.837565e-09;   ITER= 533;      CPU= 19.84
solution for nx = 800 completed
************************************

Parameter: nx = 1000
Two Grid cycle: RELRES= 3.944063e-10;   ITER= 4;        CPU= 19.19
V cycle:                RELRES= 3.944063e-10;   ITER= 4;        CPU= 22.88
PCG:                    RELRES= 9.684122e-09;   ITER= 652;      CPU= 37.46
solution for nx = 1000 completed
************************************

solution for smoothing iterations = 7 completed
************************************

************************************
```

MULTIGRID AS A SOLVER FOR POISSON PROBLEM WITH FD DISCRETIZATION



- From the above results we can see that, V cycle takes more time than PCG when the number of smoothing iterations are very low. Each recursive call of the V cycle needs a prolongation/restriction matrix to be calculated, to pass from the fine to coarsest grid. This calculation of prolongation/restriction matrix adds an overhead for the V cycle.
- As the smoothing iterations are increased both the V cycle & two grid converge quicker and in a smaller number of iterations. Beyond 7 smoothing iterations (for internal nodes up to 1000), there is no improvement in the time taken for the MG method to converge.
- As the number of internal nodes (nx) increases, the condition number of the matrix increases drastically leading to a greater number of iterations and greater time for the solution, while the MG method converges with same number of iterations and very quicker than that of the PCG solver.

MULTI GRID AS PRECONDITIONERS FOR STOKES PROBLEM

- Finally, we would like to use the Multigrid method as preconditioners for the saddle point formulation as seen previously.
- We just see one type of preconditioner for the Stokes problem, and see how we can use Multigrid property to obtain a convergence which is independent of the mesh size.
- A block diagonal preconditioner can be used for the Stokes problem, which takes the form:

$P = \begin{bmatrix} P_A & 0 \\ 0 & T \end{bmatrix}$, P_A is the preconditioner for the $(1,1)$ block & T is an approximation of the Schur complement.

- When $P_A = A$ & $T = S$, the preconditioned matrix has only three eigenvalues $\lambda = 1, \frac{1}{2} \pm \frac{\sqrt{5}}{2}$
- Preconditioning would require the inverse of both A & S. Since inverting both A & S, would be computationally expensive, we take suitable approximations.
- When approximations are used, the eigenvalues lay in three narrow intervals with bounds that are independent of the mesh size.

PRECONDITIONERS FOR BLOCKS

- For $(1{,}1)$ block: We would like to use Multigrid scheme, but we wouldn't want to write down the matrix associated with Multigrid explicitly. Hence, a routine is needed to apply A^{-1}which is done with the V or W cycles.
- The main issue lies in the choosing the right Multigrid scheme. For the Stokes problem, Multigrid with a damped Jacobi preconditioner, Galerkin coarse grid operator & application of a V-cycle, is sufficient.
- For Schur complement: We would like to exploit the fact that the mass matrix & the schur complement are spectrally equivalent and the ability to exploit the structure of the mass matrix, which is mostly tridiagonal / pentadiagonal /similar
- Hence with $P_S = Q$, can be solved easily with the normal MATLAB backslash operator. A simpler way is to use the diagonal approximation of Q. Choosing the diagonal of Q is also spectrally equivalent of to the mass matrix.
- The function handle of the preconditioners implement the following operations: $P_A x_1 = y_1$; $P_S x_2 = y_2$, where the first equation is solved using the Multigrid scheme.

REFERENCES

- The thesis of Filippo Zanetti :“Block preconditioners for saddle point linear systems arising in the FE discretization of the Navier-Stokes equations - Application to the driven cavity problem”
- PROGRAMMING OF MULTIGRID METHODS - LONG CHEN
- Solution Algorithms for the Incompressible Navier-Stokes Equations – Howard C. Elman
- Preconditioners for saddle point problems arising in computational fluid dynamics - Howard C. Elman
- A Multigrid Tutorial, 2nd Edition - William L. Briggs, Van Emden Henson, Steve F McCormick