Multigrid Method

Monday, 11 April 2022 10:10

Recall the Poisson equation,

$$\nabla^2 \phi = \rho$$

Now let $\mathcal{L}$ be any linear operator on ϕ,

$$\mathcal{L}_h \phi_h = \rho_h \quad (2)$$ on mesh of size h.

Solution $\phi_h = \mathcal{L}_h^{-1} \rho_h$

Let $\tilde{\phi}_h$ denote some approximate solution of (2) on a grid of mesh size h and ϕ_h denote the <u>exact</u> solution of (2).

Then let the <u>correction</u> to the solution be,

$$V_h = \phi_h - \tilde{\phi}_h ,$$

and the residual (defect) be,

$$d_h = \mathcal{L}_h \tilde{\phi}_h - \rho_h$$

$$\boxed{\nabla^2 \phi - \phi^2 = \rho}$$ non-linear PDE

Then $d_h = \mathcal{L}_h \tilde{\phi}_h - \mathcal{L}_h \phi_h$

$d_h = \mathcal{L}_h (\tilde{\phi}_h - \phi_h)$ since $\mathcal{L}_h$ is linear

$$\boxed{\mathcal{L}_h V_h = - d_h \quad (3)}$$

For the Laplace equation $\nabla^2 \phi = 0$ $\rho_h = 0$

so $d_h = \mathcal{L}_h \tilde{\phi}_h$ in this case.

Equation (3) remains unchanged.

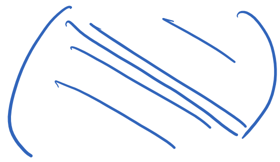
Previously (in relaxation methods) we simplified the operator $\mathcal{L}_h$ in order to approximate the solution. Iterative methods do this by using Jacobi or Gauss-Seidel (G-S) to iterate

$$\hat{\mathcal{L}}_h \hat{V}_h = - d_h$$

number of mesh points

e.g. $\hat{V}_i = - \frac{\omega}{L_{ii}} \left(\sum^{N-1} L_{ij} \hat{V}_j + d_i \right)$ for SOR e.g. $$\hat{v}_i = -\frac{\overset{\omega}{1}}{L_{ii}}\left(\sum_{\substack{j=0\\ i\neq j}}^{N-1} L_{ij}\hat{v}_j + d_i\right)$$

depends on the order of the pointer, typically chess board pattern is used.

$\mathcal{L}_h =$ 

Then the next approximation is

$$\tilde{\Phi}_h^{new} = \tilde{\Phi}_h + \hat{v}_h$$

Now we do something a bit different! Instead we approximate the operator by <u>coarsening the grid.</u>

$$\boxed{\mathcal{L}_H v_H = -d_H}$$

where H is a coarser mesh than h, typically we use $H = 2h$

Since $\mathcal{L}_H$ is of smaller dimension it will be much easier to solve. (Recall SOR scales as N^3, where N is the number of mesh points in one dimension.)

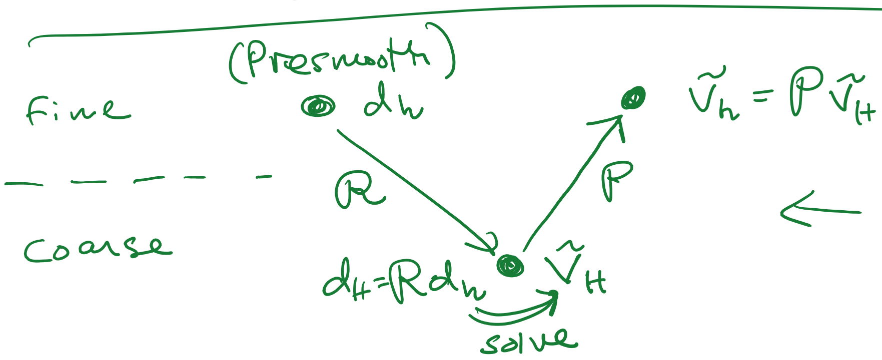
↳ So the above can be solved 8 times faster using SOR.

How do we get this coarse version of the defect, d_H? This is called <u>restriction</u>, R. $\quad d_H = R d_h$ (fine to coarse grid operation)

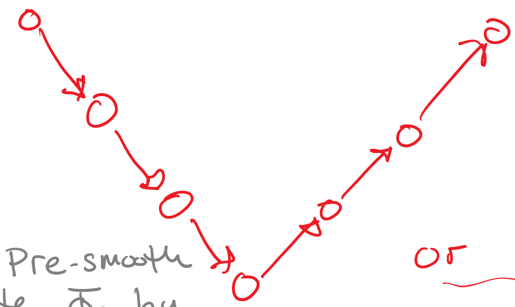
After solving for the correction on the coarse grid, we need to map it back to the fine grid. This is called <u>prolongation</u>, P.

$$\tilde{v}_h = P v_H$$

Finally $\tilde{\Phi}_h^{new} = \tilde{\Phi}_h + \tilde{v}_h$



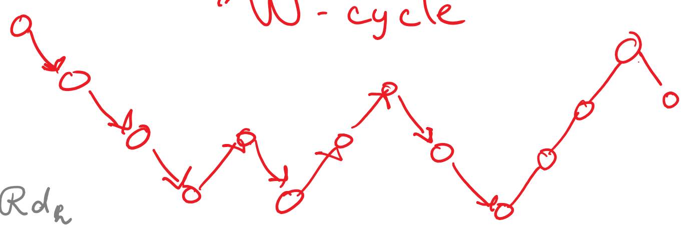
← This is a V-cycle



← This is also a V-cycle

or nest 2-cycles

"W-cycle"



Steps: ① Pre-smooth
Compute $\tilde{\Phi}_h$ by applying a few regular relaxation steps

② Coarse grid correction
a) compute defect: d_h
b) Restrict onto H: $d_H = R d_h$
c) Solve correction v_H
d) prolong: $\tilde{v}_h = P v_H$
e) Compute $\tilde{\Phi}_h^{new} = \tilde{\Phi}_h^{OLD} + \tilde{v}_h$

③ Post smooth $\tilde{\Phi}_h^{new}$.

Imagine v_h expanded as a Fourier series of wavelengths. The lower half of modes in the spectrum are "smooth" modes, the others are higher frequency modes.

Relaxation methods have a very small effect on the smooth modes, but a very large effect on the high frequency modes.

$\Rightarrow$ they are good smoothing operators.

For 2-grid method components of the error (correction) with $\lambda \lesssim 2H$ are not even represented on the coarse grid and cannot be reduced to zero there.

R and P : want $I = PR$

from which $R = P^{+}$, the adjoint operator to the prolongation.

For: $\mathcal{L} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ $\quad P = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$ $\quad \cancel{P = [1]}$ $\quad \cancel{R = [1/4]}$

could be any interpolation order $m_P = 2$

~~straight injection~~

$$R = \frac{1}{4} P^T = \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & \frac{1}{4} & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}$$

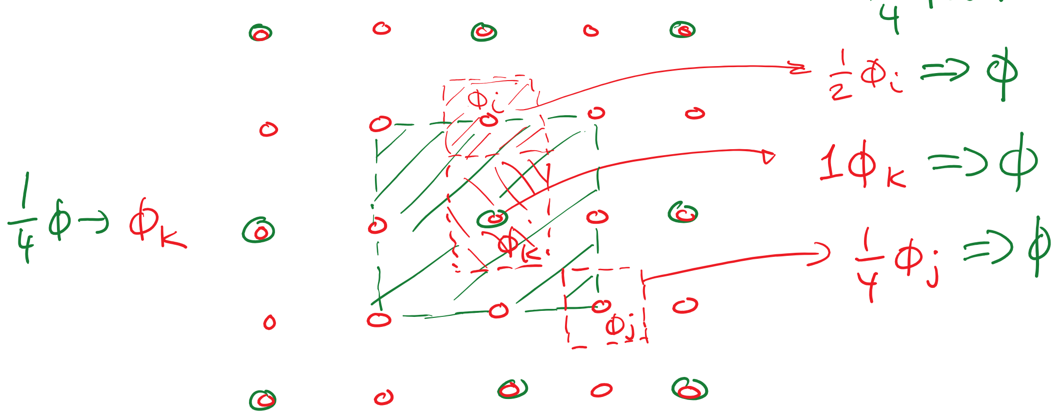
$$\underset{2}{m_P} + \underset{2}{m_R} > \underset{2}{m_L}$$

---

o coarse grid values

o fine grid values

with $\frac{1}{4}$ Normalization



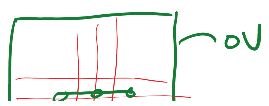
Boundary Conditions?

For starters lets assume that the BC lies on both grid points of h and H.

Here we can proceed by modifying the restriction an prolongation operators with $\emptyset$ for points at the boundary.

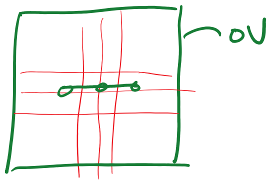
Eg. BC on the left points:

$\mathcal{L} = \begin{bmatrix} 0 & 1 & 0 \\ \end{bmatrix}$ $\quad R = \begin{bmatrix} 0 & \frac{1}{8} & 1/16 \\ \end{bmatrix}$



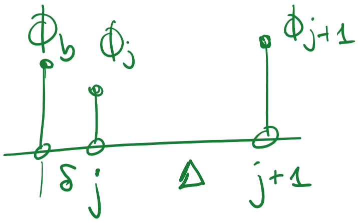
E.g.

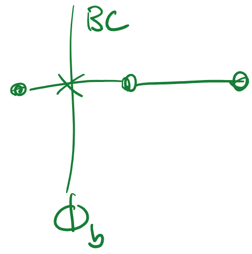
$$\mathcal{L} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad R = \begin{bmatrix} 0 & \frac{1}{8} & \frac{1}{16} \\ 0 & \frac{1}{4} & \frac{1}{8} \\ 0 & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$$



$$P = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{4} \\ 0 & 1 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$$

"General" boundary conditions.





$$\frac{\partial^2 \phi}{\partial x^2} = ? \qquad \phi = ax^2 + bx + c$$

3 unknowns

3 function values

Without loss of generality assume $x_j = 0$

$$c = \phi_j$$

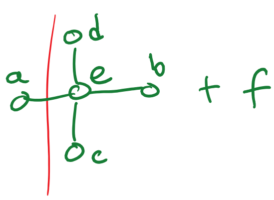
$$\phi_b = a(-\delta)^2 + b(-\delta) + \phi_j \qquad \times \Delta$$

$$+ \quad \phi_{j+1} = a(\Delta)^2 + b\Delta + \phi_j \qquad \times \delta$$

$$\Delta(\phi_b - \phi_j) + \delta(\phi_{j+1} - \phi_j) = a(\Delta\delta^2 + \Delta^2\delta)$$

$$a = \frac{1}{(\delta^2 + \delta\Delta)}(\phi_b - \phi_j) + \frac{1}{(\Delta^2 + \delta\Delta)}(\phi_{j+1} - \phi_j)$$

$$\frac{\partial^2 \phi}{\partial x^2} = 2a = \underbrace{\boxed{\frac{2}{(\delta^2 + \delta\Delta)}(\phi_b}}_{\text{constant}} - \phi_j) + \frac{2}{(\Delta^2 + \delta\Delta)}(\phi_{j+1} - \phi_j)$$

$\mathcal{L} =$  $+ f$

$$d_{ke} = \Big[a_{ke} \cdot \phi_{k-1e} + b_{ke}\phi_{k+1e} + c_{ke}\phi_{ke-1} + d_{ke}\phi_{ke+1}$$

$$+ e_{ke}\phi_{ke} + f_{ke}\Big]$$