Consider solution of Navier-Stokes equations (incompressible)

$$\rho = \text{const} \Rightarrow \begin{cases} \rho_t + \nabla\cdot(\rho\vec{u}) = 0 \Rightarrow \nabla\cdot\vec{u} = 0 \\ \rho\vec{u}_t + \rho(\vec{u}\cdot\nabla)\vec{u} = -\nabla p + \mu\nabla^2\vec{u} \end{cases}$$

↳ expresses viscosity (simpler form that uses ρ=const)

Choose units $\rho = 1$

$$(1)\quad \begin{cases} \nabla\cdot\vec{u} = 0 \\ \vec{u}_t + (\vec{u}\cdot\nabla)\vec{u} = -\nabla p + \nu\nabla^2\vec{u} \end{cases}$$

(Nondimensionalize w.r.t. ref. length, time $\Rightarrow \nu = \frac{1}{Re}$)

$$Re = \frac{U_{ref}\,L_{ref}}{\nu}$$

System (1) analyze eigenmodes: mixed $\begin{cases} \text{elliptic} \\ \text{parabolic} \\ \text{hyperbolic} \end{cases}$

To avoid solving a B.V.P. for $\begin{pmatrix}\vec{u} \\ p\end{pmatrix}$ simultaneously use operator splitting a.k.a. "Projection Method"

1) Solve the hyperbolic part (convective terms)

$$\begin{cases} \vec{u}_t + (\vec{u}\cdot\nabla)\vec{u} = -\nabla\chi \\ \vec{u}(t^n,\vec{x}) = \vec{u}^n(\vec{x}) \end{cases} \rightarrow \vec{u}^*$$

χ some approx. of p

2) Solve the parabolic part

$$\begin{cases} \vec{u}_t = \nu\nabla^2\vec{u} \\ \vec{u}(t^n,\vec{x}) = \vec{u}^*(\vec{x}) \end{cases} \rightarrow \vec{u}^+$$

3) $\vec{u}^+$ $\nabla\cdot\vec{u}^+ \neq 0$ generally

$$\vec{u}^{n+1} = \vec{u}^+ + \vec{v}$$

$$\nabla\cdot\vec{u}^{n+1} = 0 \Rightarrow \quad \nabla\cdot\vec{v} = -\nabla\cdot\vec{u}^+$$

$$\vec{v} = \nabla\phi \Rightarrow \quad \nabla^2\phi = -\nabla\cdot\vec{u}^+$$

In Projection method two costly linear solves arises for steps (2) & (3).

(2) Forward in Time Centered in Space (FTCS) leads to $\Delta t \simeq O(\Delta x^2)$ (Forward Euler)

Semi-implicit (A-M of 2^{nd} order; trapezoid)

$$\vec{u}_t = -\nu\nabla^2\vec{u} \iff$$

$$\vec{u}^{n+1} - \vec{u}^n = -\frac{\nu\Delta t}{2}\left(\nabla_h^2\vec{u}^n + \nabla_h^2\vec{u}^{n+1}\right) \iff$$

(3) $$\left(I + \frac{\nu\Delta t}{2}\right)\nabla_h^2\underline{\vec{u}^{n+1}} = \left(I - \frac{\nu\Delta t}{2}\right)\nabla_h^2\vec{u}^n$$

$\to$ Crank-Nicolson algorithm

In step (3) $\nabla_h^2\phi = -\nabla_h\cdot\vec{u}^+$ (4)

General form of linear systems arising from (3), (4)

$Ax = b$ $A \sim \begin{pmatrix} \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots \end{pmatrix}$ $N = m$. of grid

$$Ax = b \qquad A \sim \begin{pmatrix} \ddots & \ddots & & 0 \\ \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots \\ 0 & & \ddots & \ddots \end{pmatrix}$$

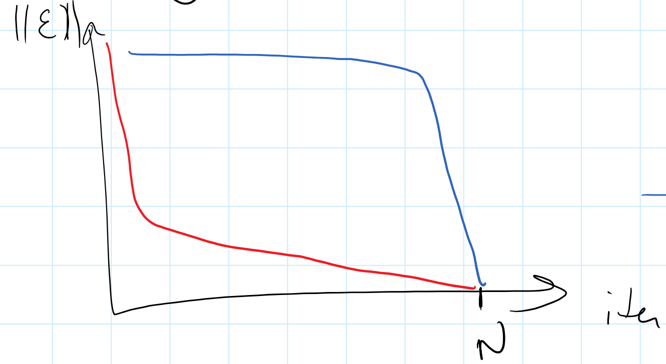
$N =$ no. of grid points

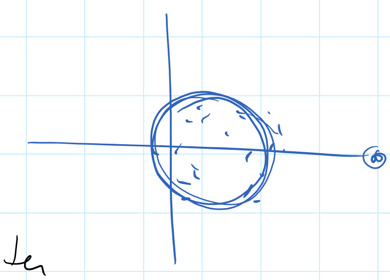
$A \in \mathbb{R}^{N \times N}$ memory requirements to store A

at most $5N$ (non-Cartesian grids) or just 2 scalars for unif. Cartesian grid

unif. Cart. grid $\quad A = \begin{pmatrix} -4 & 1 & & 1 & & \\ 1 & -4 & 1 & & \ddots & \\ & \ddots & \ddots & \ddots & & 1 \\ 1 & & \ddots & \ddots & \ddots & \\ & \ddots & & 1 & -4 & 1 \\ & & 1 & & 1 & -4 \end{pmatrix}$

Sparsity is not preserved by LU factorization





$$\begin{pmatrix} -2 & 1 & & \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{pmatrix}$$

$$\frac{\partial}{\partial x} e^{i\alpha x} = i\alpha \, e^{i\alpha x}$$

Gauss-Seidel $\qquad Ax = b$

$$A = L + D + U \quad \Rightarrow$$

$$Dx = b - Lx - Ux$$

$$x^{n+1} = D^{-1}\left(b - Lx^n - Ux^n\right) \qquad \text{Jacobi}$$

$$(6) \quad x^{n+1} = D^{-1}(b - Lx^{n+1} - Ux^{n}) \qquad \text{Gauss-Seidel}$$

$$(7) \quad x^{n+1} = D^{-1}(b - Lx^{n} - Ux^{n+1})$$

Red-Black Gauss-Seidel alternate between (6) & (7).

G-S typically converges slowly, linearly

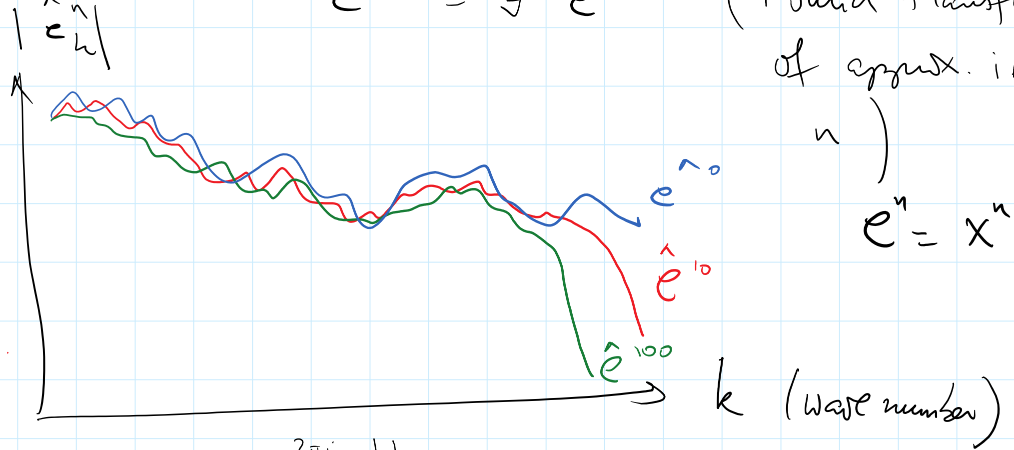
$$\|x^{n+1} - x\| = \alpha \|x^{n} - x\|$$

$$\alpha \simeq 0.98$$

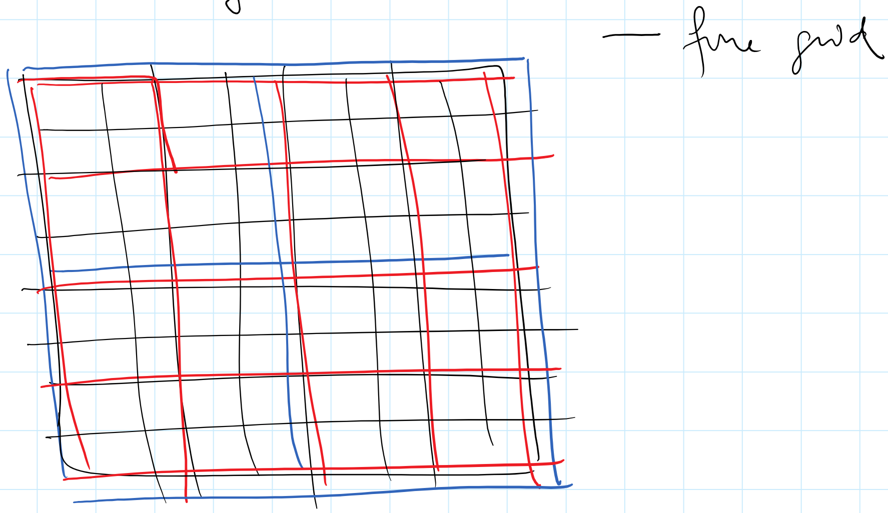
Multigrid:

$$\hat{e}^{n} = \mathcal{F} e^{n}$$ (Fourier transform of approx. in iteration n)

$$e^{n} = x^{n} - x$$



$$\hat{e}^{n}_{k} = \sum_{j=0}^{N-1} e^{\frac{2\pi i}{N} jkh} e^{n}_{j}$$



V-grid algorithm

V-grid algorithm

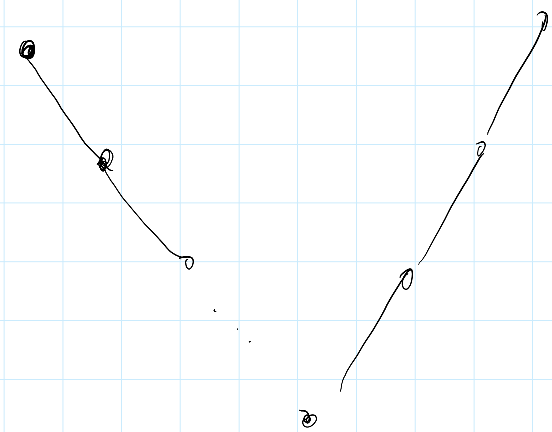
finest level L

$L-1$

$L-2$

$\vdots$

1



M-G algorithm $\qquad Ax = b$

(8) $\quad r^n = b - Ax^n$ = residual

(9) $\quad \delta^n = x - x^n$

$A \cdot (9) \qquad A\delta^n = Ax - Ax^n = b - Ax^n = r^n$

$$A\delta^n = r^n$$

r^{n}_{l}: $n \to$ iteration number, $l \to$ grid level

$$r^n_L = b - Ax^n_L$$

G-S $\Rightarrow \quad \delta^n_L = GS(A, r^n_L, \delta^n_L = 0)$

$$x^{n+1}_L = x^n_L + \delta^n_L$$

Introduce restriction operator $\delta^n_{L-1} = R^L_{L-1}\delta^n_L$

Fine-to-Coarse $\downarrow$

Ex: $R^L_{L-1} = \frac{1}{8}\begin{pmatrix} & 1 & \\ 1 & 4 & 1 \\ & 1 & \end{pmatrix}$

$$r^n_{L-1} = b - AR^L_{L-1}x^n_L$$

$$\delta^n_{L-1} = GS(A, r^n_{L-1}, R^L_{L-1}\delta^n_L)$$

$$\delta^n_{L-1} = GS(A, r^n_{L-1}, R^L_{L-1}\delta^n_L)$$

Coarse-to-fine

P = prolongation operator

$$x^n_L = P^{L-1}_L x^n_{L-1};$$

$$\delta^n_L = P^{L-1}_L \delta^n_{L-1}$$

M.G. ideal efficiency → computational complexity $O(N \log N)$

Krylov methods $O(N^2)$

Gauss elim, LU $O(N^3/3)$