Multigrid Acceleration

Advanced CFD 04

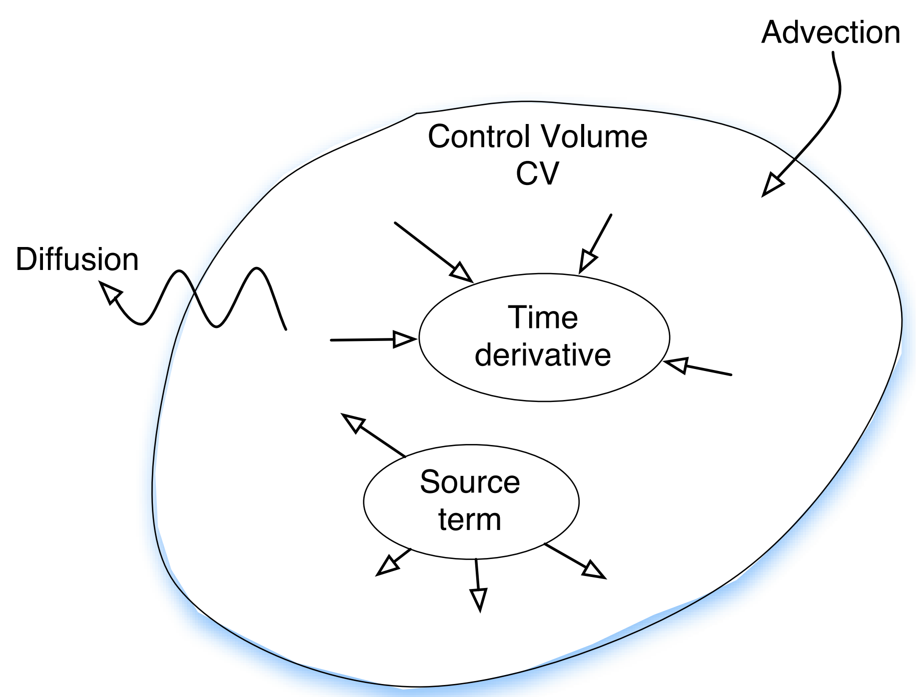
Outline

- The General Scalar Equations
- The Algebraic Equations
- Performance of Iterative Solvers
- Error frequency
- Anisotropy and Performance
- Agglomeration in Multigrid
- Conclusion

The General Scalar Equation

$$\underbrace{\frac{\partial(\rho\phi)}{\partial t}}_{\textit{transient term}} + \underbrace{\nabla.(\rho\mathbf{u}\phi)}_{\textit{convective term}} = \underbrace{\nabla\cdot(\Gamma\nabla\phi)}_{\textit{diffusion term}} + \underbrace{Q_\phi(\phi)}_{\textit{source term}}$$

The scalar equation is a balance equation written in differential form



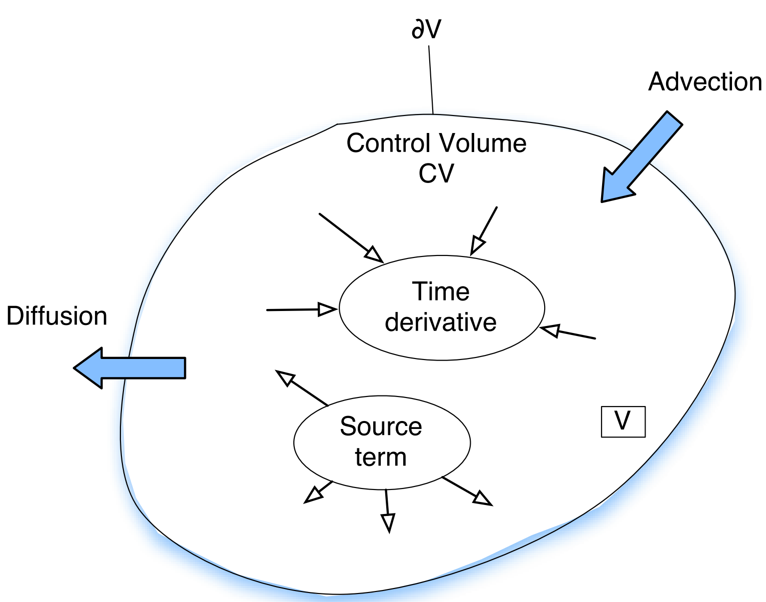
Balance Form

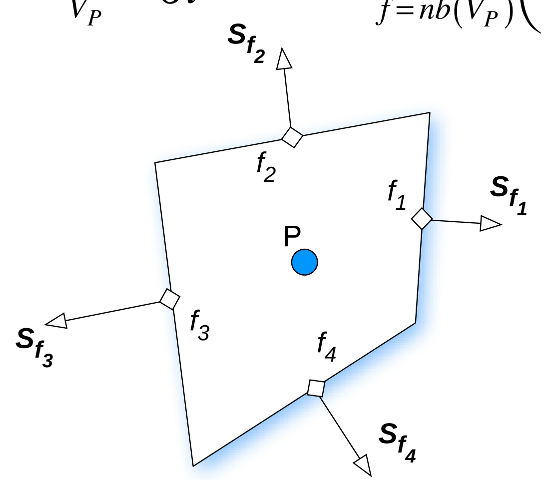
To recover the balance form we integrate over some CV

$$\int_V \frac{\partial(\rho\phi)}{\partial t} dV + \oint_{\partial V} (\rho\mathbf{u}\phi)\cdot d\mathbf{S} = \oint_{\partial V} (\Gamma\nabla\phi)\cdot d\mathbf{S} + \int_V Q(\phi)dV$$

$$\int_V \frac{\partial(\rho\phi)}{\partial t} dV + \sum_{f=nb(V)} \left(\int_f (\rho\mathbf{u}\phi)\cdot d\mathbf{S} \right) - \sum_{f=nb(V)} \left(\int_f (\Gamma\nabla\phi)\cdot d\mathbf{S} \right) = \int_V Q(\phi)dV$$

$$\int_{V_P} \frac{\partial(\rho\phi)}{\partial t} dV + \sum_{f=nb(V_P)} \left(\int_f \mathbf{J}\cdot d\mathbf{S} \right) = \int_{V_P} Q_\phi(\phi)dV_P$$





discrete CV

$$\mathbf{J} = \mathbf{J}^{\mathbf{C}} + \mathbf{J}^{\mathbf{D}}$$
$$= (\rho\mathbf{u}\phi) - (\Gamma\nabla\phi)$$

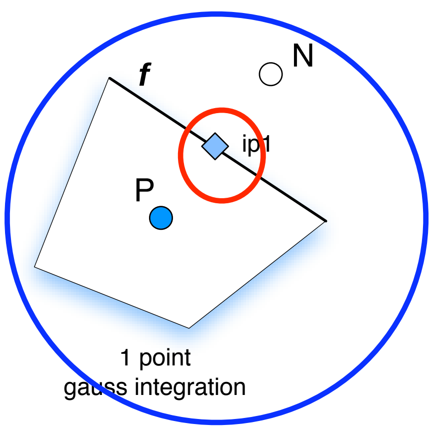
Flux Integration

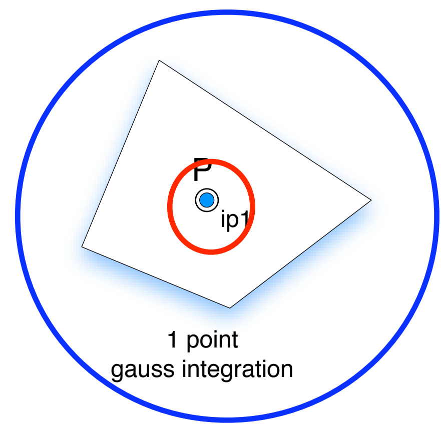
$$\int_f \mathbf{J}\cdot d\mathbf{S} = \sum_{ip(f)} (\mathbf{J}\cdot\mathbf{S})_{ip}$$

$$\sum_{f=nb(V_P)} (-\Gamma\nabla\phi)_f \cdot \mathbf{S}_f$$

$$\int_{V_P} Q(\phi)\, dV_P = \sum_{ip(P)} (Q_{ip}V_{ip})$$

$$Q_P V_P$$





$$\int_{V_P} \frac{\partial(\rho\phi)}{\partial t}\, dV = \frac{(\rho\phi)_P^t - (\rho\phi)_P^\circ}{\Delta t} V_P$$

2D Diffusion

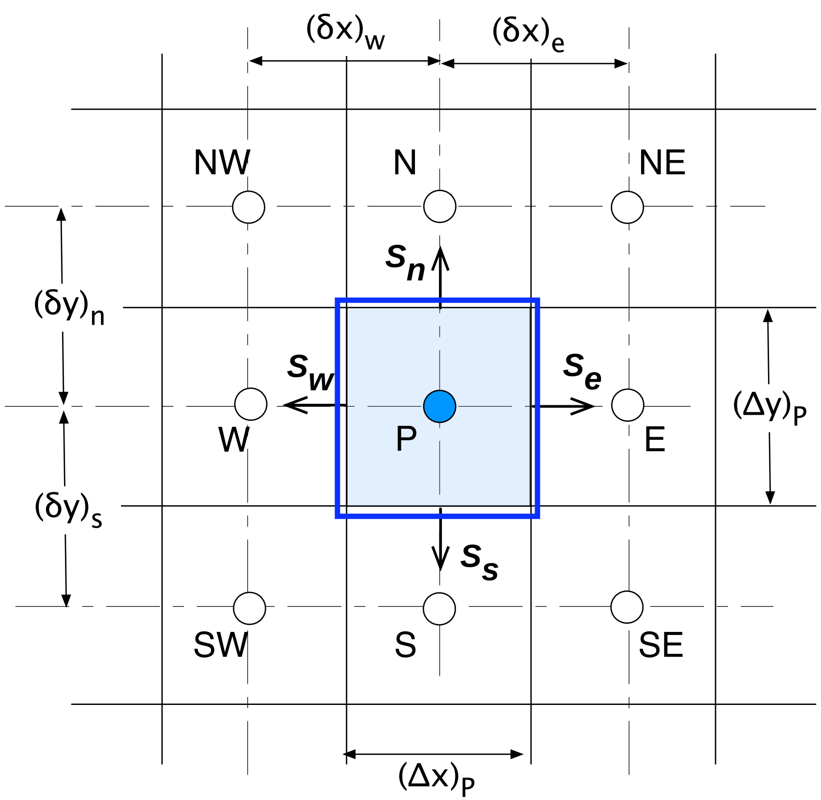
$$\boxed{\left(\frac{\rho V}{\Delta t}+\sum_{NB(P)} a_{NB}^{D}\right)\phi_P^t=\sum_{NB(P)} a_{NB}^{D}\phi_{NB}^{t}+b_p+\frac{\rho V}{\Delta t}\phi_P^{\circ}}$$

$$\frac{(\rho\phi)_P^t-(\rho\phi)_P^{\circ}}{\Delta t}V_P=\left(\frac{\rho V}{\Delta t}\right)_P\left(\phi_P^t-\phi_P^{\circ}\right)$$

$$-(\Gamma\nabla\phi)_f\cdot\mathbf{S}_f=-\Gamma_f\left(\frac{\phi_F-\phi_P}{d_{PF}}\right)\left|S_f\right|$$

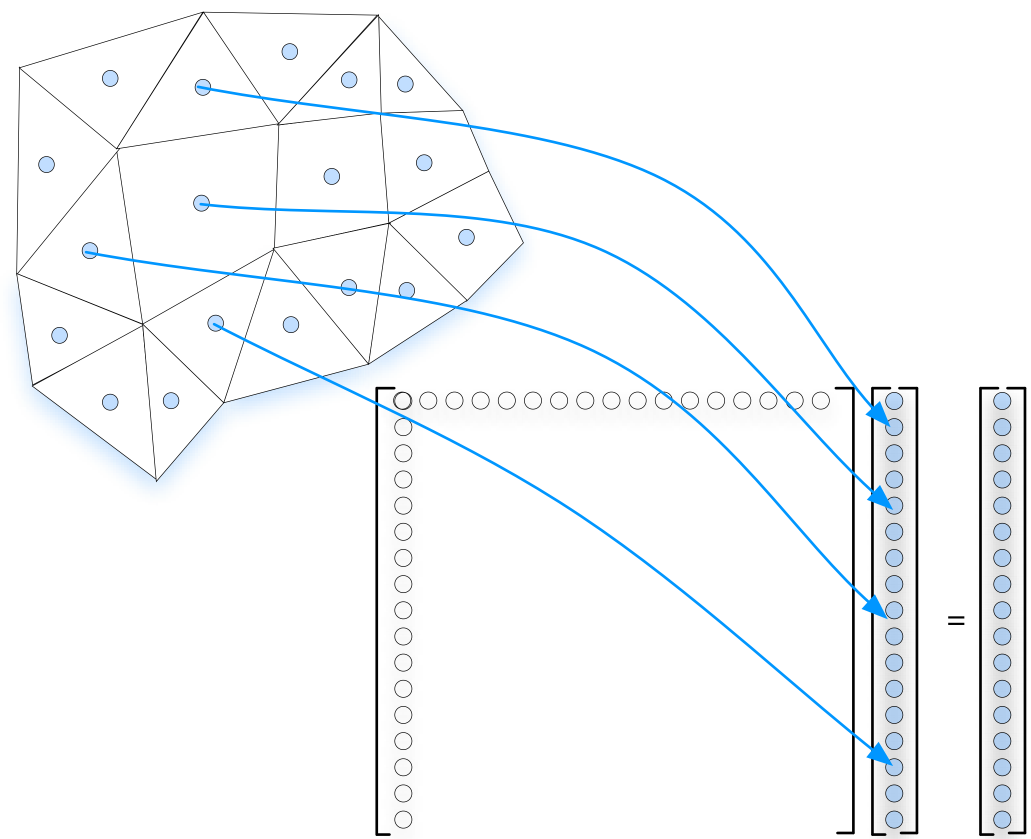
$$=-\frac{\Gamma_f\left|S_f\right|}{d_{PF}}(\phi_F-\phi_P)$$

$$Q_P V_P=b_P \qquad a_F=\Gamma_f\frac{\left|S_f\right|}{\left|d_{PF}\right|}$$



Algebraic Equations

$$\left(\frac{\rho V}{\Delta t}+\sum_{NB(P)} a_{NB}^{D}\right)\phi_P^t = \sum_{NB(P)} a_{NB}^{D}\phi_{NB}^{t} + b_p + \frac{\rho V}{\Delta t}\phi_P^{\circ}$$



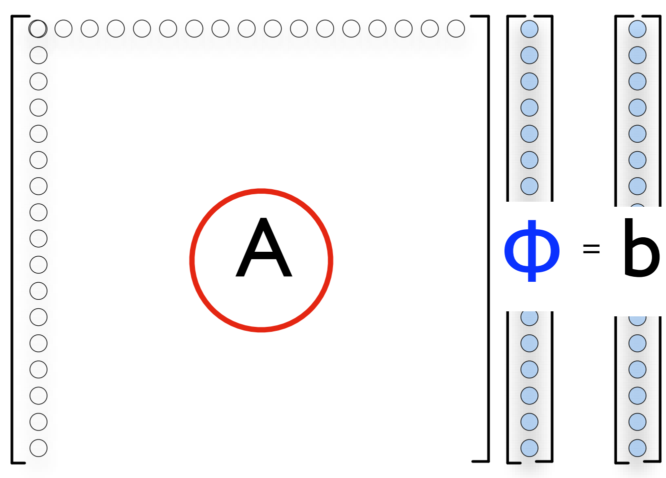
Direct Solvers

Invert whole matrix to obtain *exact* solution

$$\Phi = A^{-1}\, b$$

Very expensive in terms of memory and cpu time

Not needed as coefficients are generally computed iteratively

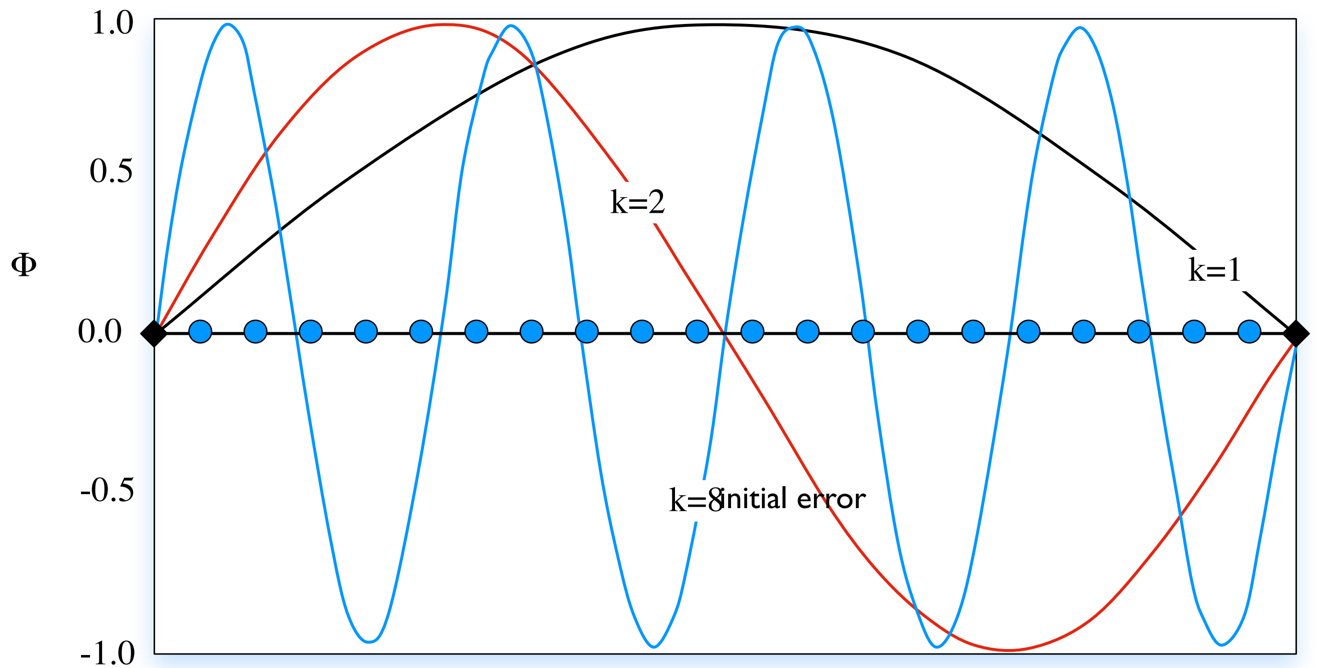


Iterative Solvers

- Field approaches solution iteratively
- Advantage
  - Efficient (memory wise)
  - cheap (compared to direct solvers
- Disadvantage
  - Stalling of convergence rate
  - do not scale with problem size

Effect of Error Distribution

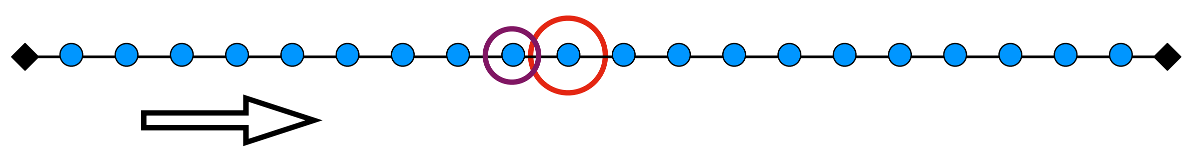
$$\nabla \cdot (\Gamma \nabla \phi) = 0$$



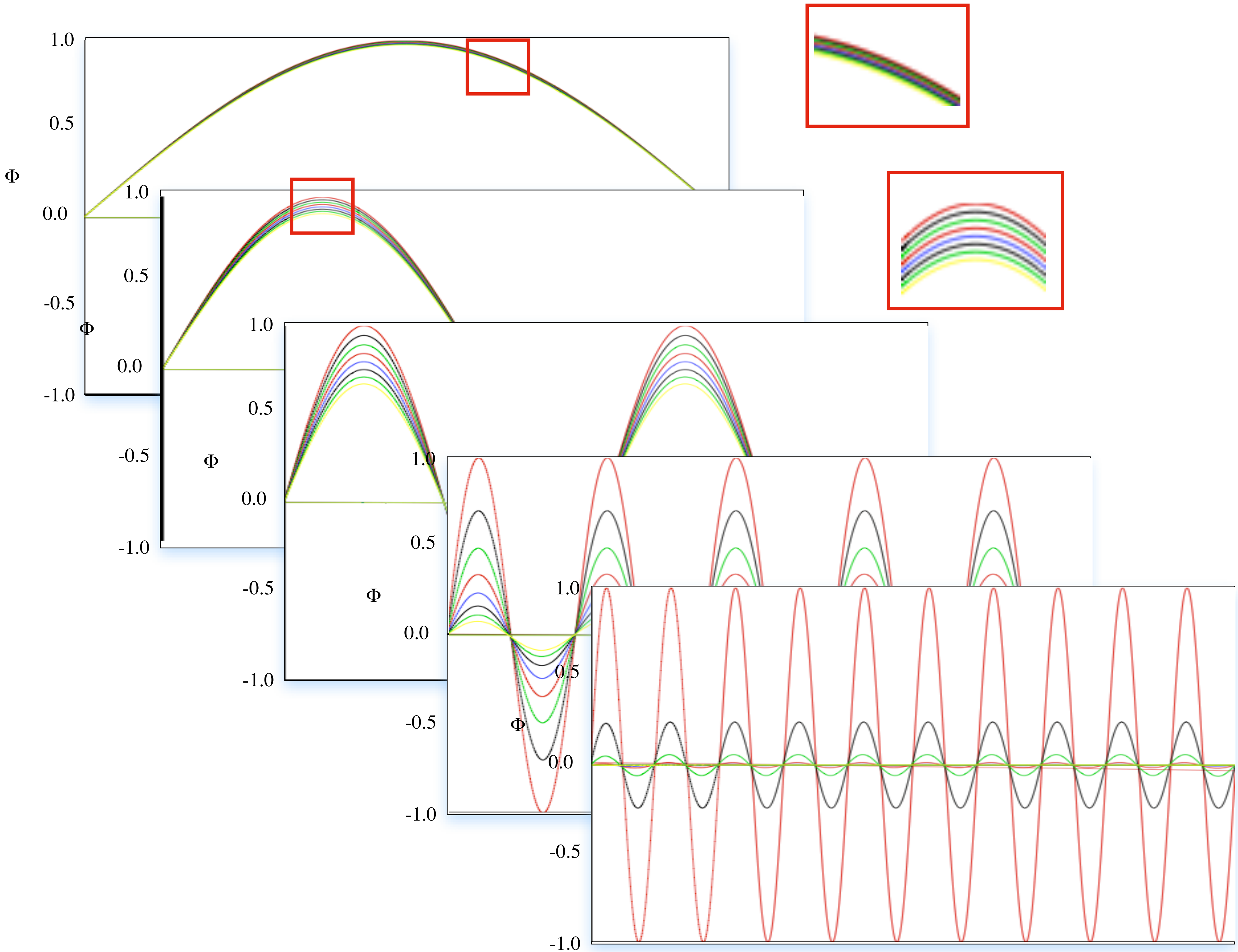
Gauss-Siedel Point Solver

$$a_P \phi_P = \sum_{NB(P)} a_{NB}^D \phi_{NB} + b_p$$

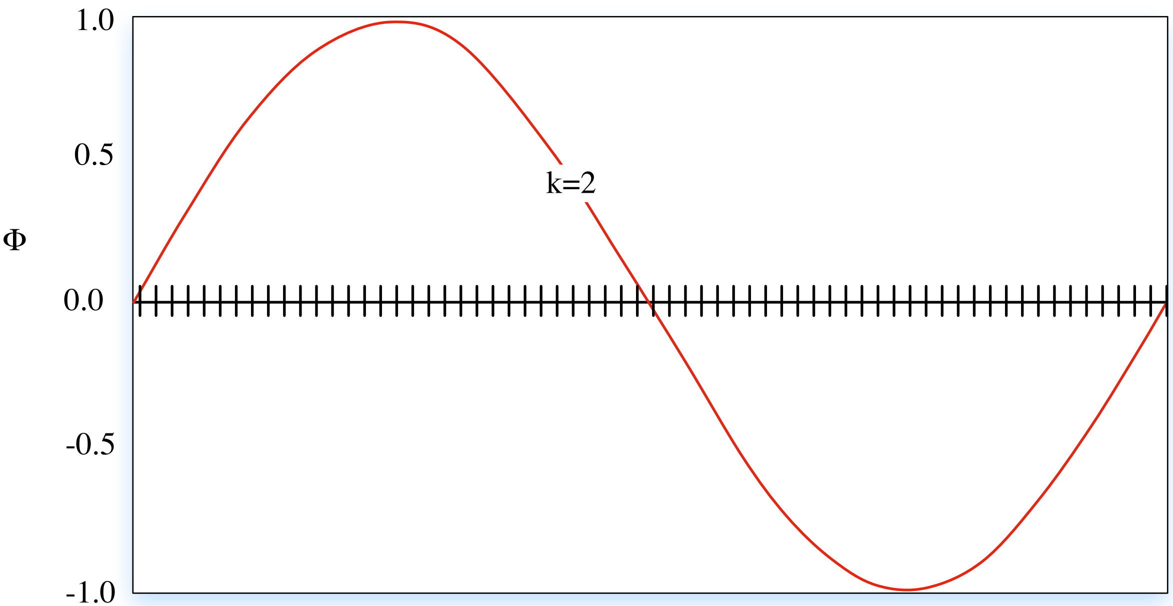
$$\phi_P = \frac{\sum_{NB(P)} a_{NB}^D \phi_{NB} + b_p}{a_P}$$

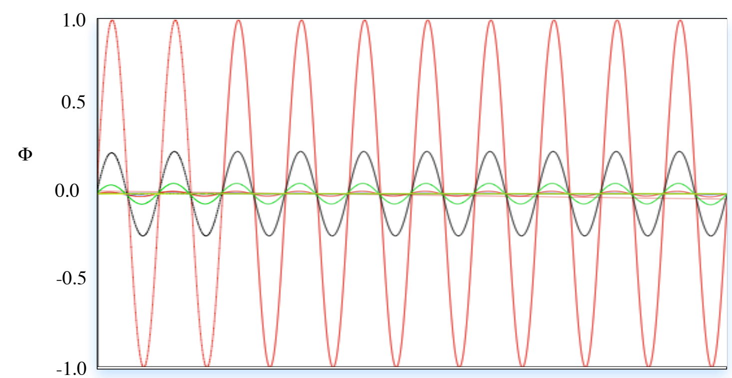


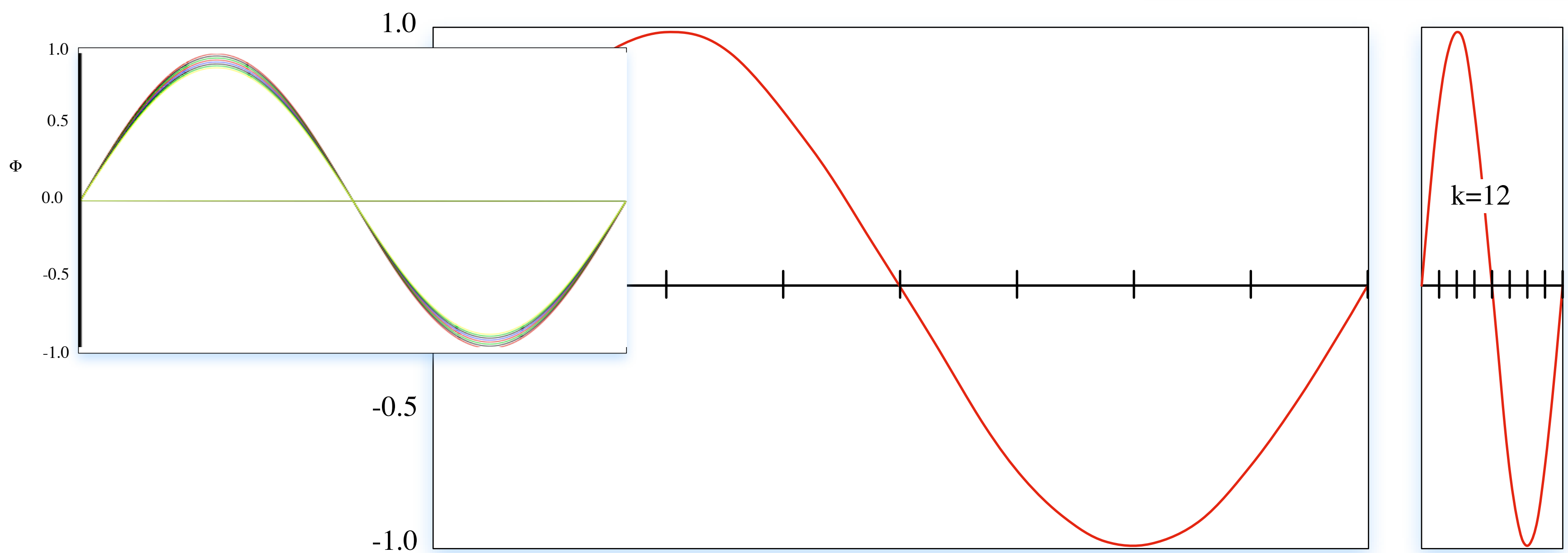
$$\phi_P^l = \frac{1}{a_P}\left(\sum_{NB(P)} a_{NB}^D \phi_{NB}^l + \sum_{NB(P)} a_{NB}^D \phi_{NB}^{l-1} + b_p \right)$$


1.0
0.5
Φ
0.0
-0.5
-1.0
1.0
0.5
Φ
0.0
-0.5
-1.0
1.0
0.5
Φ
0.0
-0.5
-1.0
1.0
0.5
Φ
0.0
-0.5
-1.0
1.0
0.5
Φ
0.0
-0.5
-1.0

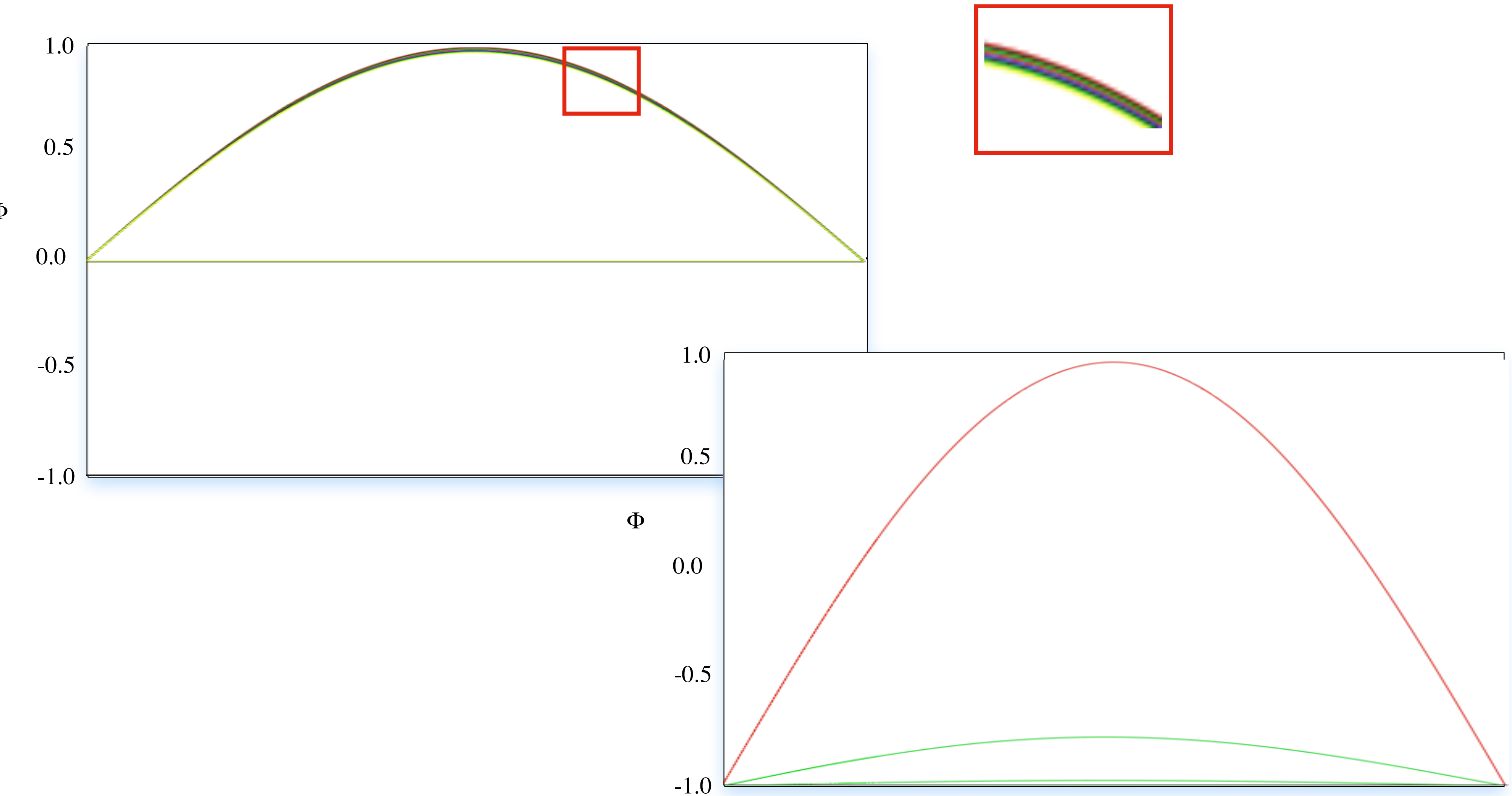
Multigrid Principle







Comparison



Performance of Point Solver

- Properties
  - Good Smoothing of High Frequency Errors
  - Error generally composed of multiple frequencies
  - Stalling when high frequency errors smoothed out

Effect of Disparate Time-Scales

Interpretation of coefficients in terms of time scales

$$\left(\frac{\rho V}{\Delta t}+\sum_{NB(P)} a_{NB}^{CD}\right)\phi_P^t=\sum_{NB(P)} a_{NB}^{D}\phi_{NB}^t+b_p+\frac{\rho V}{\Delta t}\phi_P^{\circ}$$

$$\frac{\rho V}{\Delta t}\left(\phi_P^t-\phi_P^{\circ}\right)=\sum_{NB(P)} a_{NB}^{D}\left(\phi_{NB}-\phi_P\right)+b_p$$

The time scale for propagation of Φ between node P and F

$$a_F^D=\frac{\Gamma_f\left|S_f\right|}{\left|d_{PF}\right|} \qquad \tau_f^D=\frac{\rho\left|d_{PF}\right|^2}{\Gamma_f}=\frac{\rho\Delta V}{a_F^D}$$

Effective Time-Step

$$\left(\frac{\rho\Delta V}{\Delta t}+\sum_{NB(P)} a_{NB}^{CD}\right)\phi_P^{l+1} = \sum_{NB-(P)} a_{NB}^{CD}\phi_{NB}^{l+1} + \sum_{NB+(P)} a_{NB}^{CD}\phi_{NB}^{l} + \left\{b_p + \frac{\rho\Delta V}{\Delta t}\phi_P^{\circ}\right\}$$

$$\left(\frac{\rho\Delta V}{\Delta t}+\sum_{NB(P)} a_{NB}^{CD}\right)\left(\phi_P^{l+!}-\phi_P^{l}\right) = \sum_{NB-(P)} a_{NB}^{CD}\left(\phi_{NB}^{l+1}-\phi_P^{l}\right) + \sum_{NB+(P)} a_{NB}^{CD}\left(\phi_{NB}^{l}-\phi_L^{l}\right) + \left\{b_p + \frac{\rho\Delta V}{\Delta t}\left(\phi_P^{\circ}-\phi_P^{l}\right)\right\}$$

$$\frac{\rho\Delta V}{\Delta t_{eff}}\left(\phi_P^{l+1}-\phi_P^{l}\right) = \sum_{NB(P)} a_{NB}^{CD}\left(\phi_{NB}^{l}-\phi_P^{l}\right) + b_p$$

$$\frac{1}{\Delta t_{eff}} = \frac{1}{\Delta t} + \sum_{NB(P)} \frac{a_{NB}^{CD}}{\rho\Delta V} = \frac{1}{\Delta t} + \sum_{NB(P)} \frac{1}{\tau_{NB}^{CD}}$$

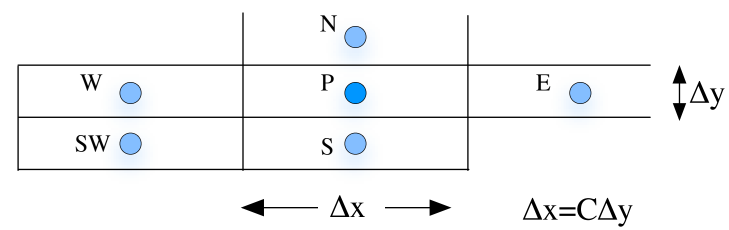
Stalling With Fine Grids

$$\frac{1}{\Delta t_{eff}} = \sum_{NB(P)} \frac{1}{\tau_{NB}^{CD}} = \sum_{NB(P)} \left(\frac{\Gamma_f}{\rho |d_{PF}|^2} \right)$$

$$\Delta t_{eff} = \frac{1}{\left(\frac{2}{\tau_x^{CD}} + \frac{2}{\tau_y^{CD}} \right)}$$

$$\Delta t_{eff} = \frac{\tau_x^{CD}}{2\left(1 + (\Delta x / \Delta y)^2\right)}$$

$$\Delta t_{eff} = \frac{\tau_x^{CD}}{2\left(1 + C^2\right)}$$



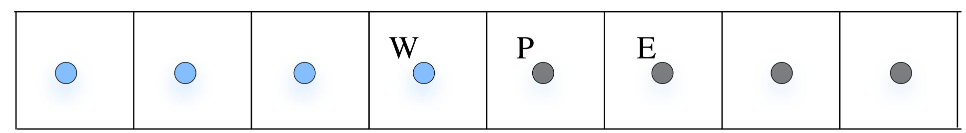
Effect of Anisotropic Coefficients

$$\tau_{21}^{D} = \frac{\rho \Delta x^2}{\Gamma_{12}} = \frac{\rho \Delta x^2}{\Gamma_1 \frac{2C_1}{(1+C)}}$$

$$\Gamma_{12} = \frac{2\Gamma_1 \Gamma_2}{\Gamma_1 + \Gamma_2} = \Gamma_1 \frac{2C}{1+C}$$

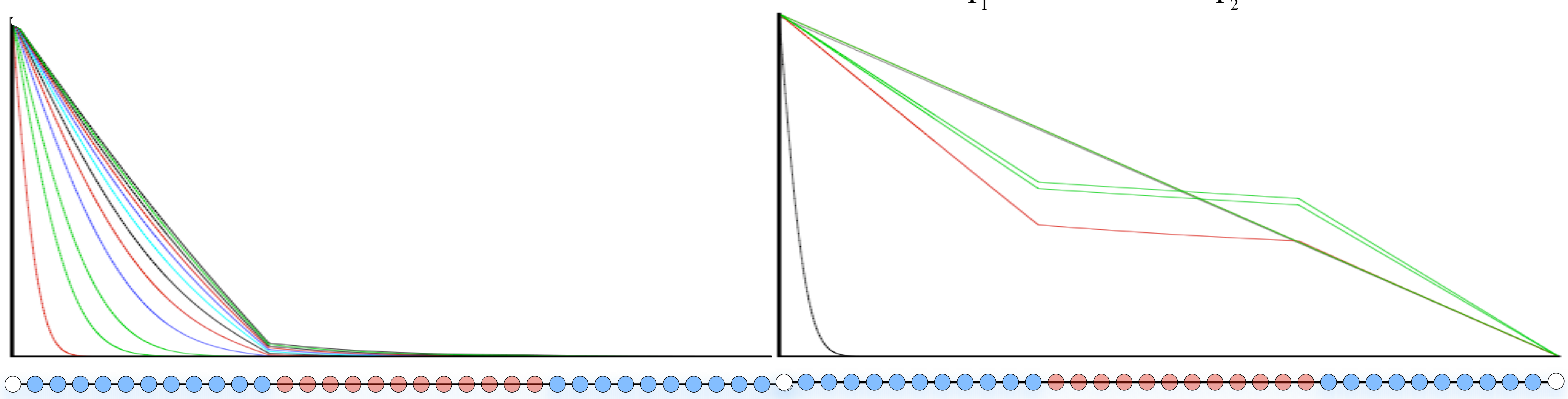
$$\Delta t_{eff} = \frac{1}{\frac{1}{\tau_{12}^{D}} + \frac{1}{\tau_{2}^{D}}} = \frac{\rho \Delta x^2}{\Gamma_1 \frac{2C_1}{(1+C)} + \Gamma_2}$$

$$\Delta t_{eff} = \tau_1^{D} \frac{1+C}{C(3+C)}$$

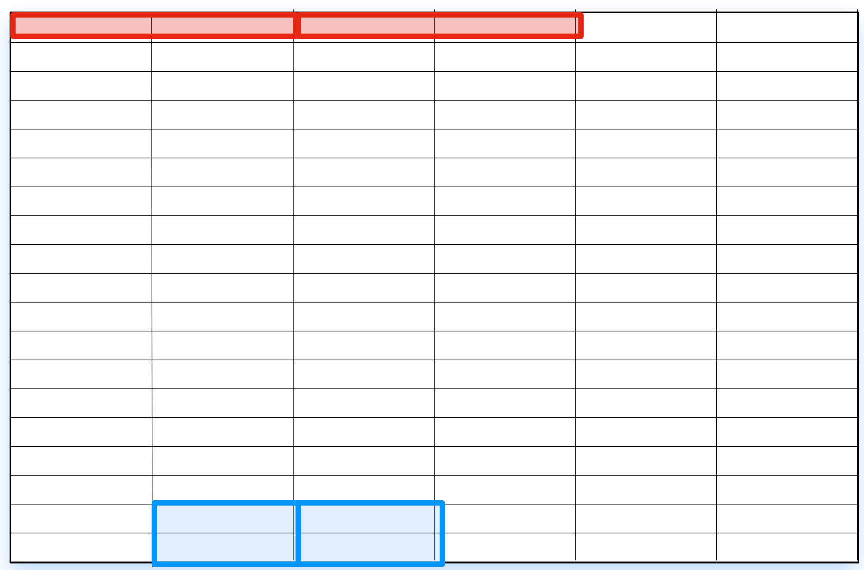


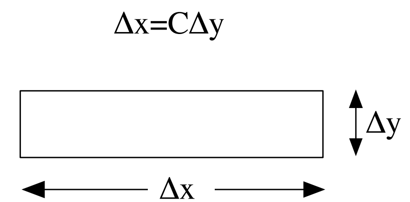
Γ_1 Γ_2=CΓ_1

$$\tau_1^{D} = \frac{\rho \Delta x^2}{\Gamma_1} \qquad \tau_2^{D} = \frac{\rho \Delta x^2}{\Gamma_2}$$



Agglomeration Algorithm



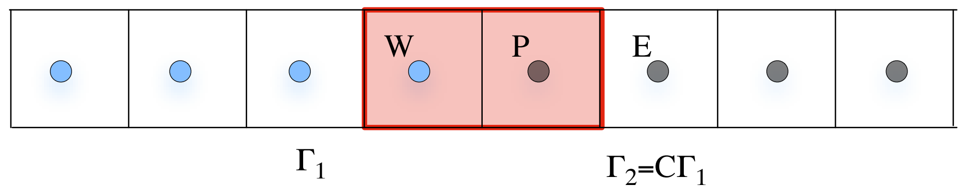


$$\Delta t_{eff} = \frac{\tau_x^{CD}}{2\left(1+10^2\right)}$$

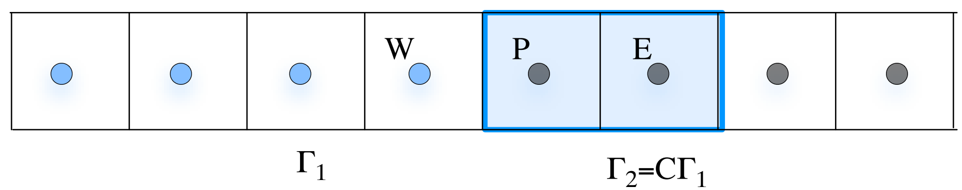
$$\Delta t_{eff\,(aggl)} = \frac{\tau_x^{CD}}{2\left(1+5^2\right)}$$

$$\Delta t_{eff} = 4\frac{\rho\Delta x^2}{\Gamma_1}\frac{1}{(1+10)}$$

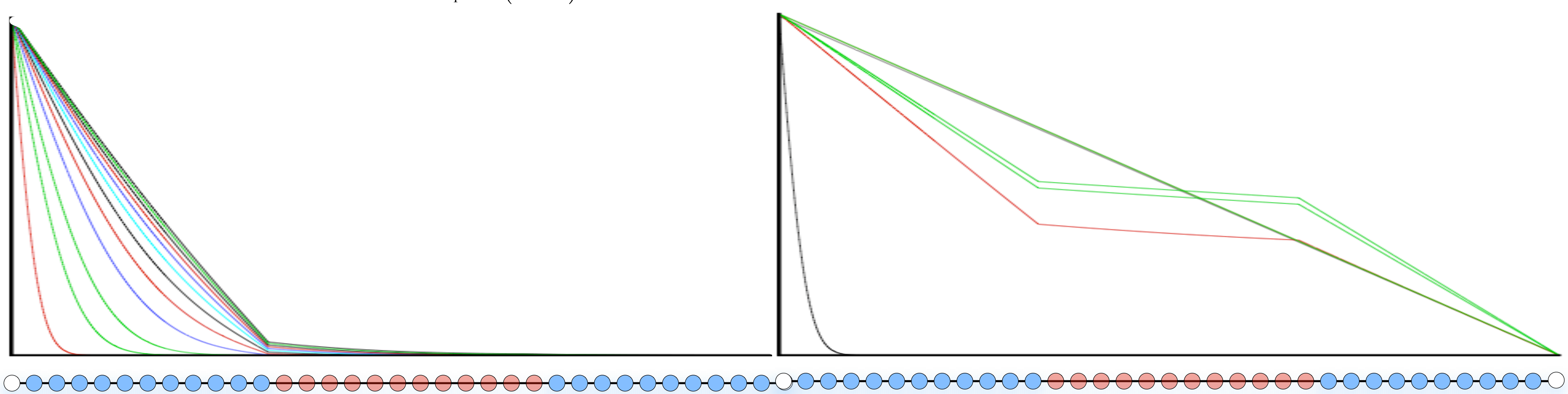
$$\Delta t_{eff} = 4\frac{\rho\Delta x^2}{\Gamma_1(1+C)}$$



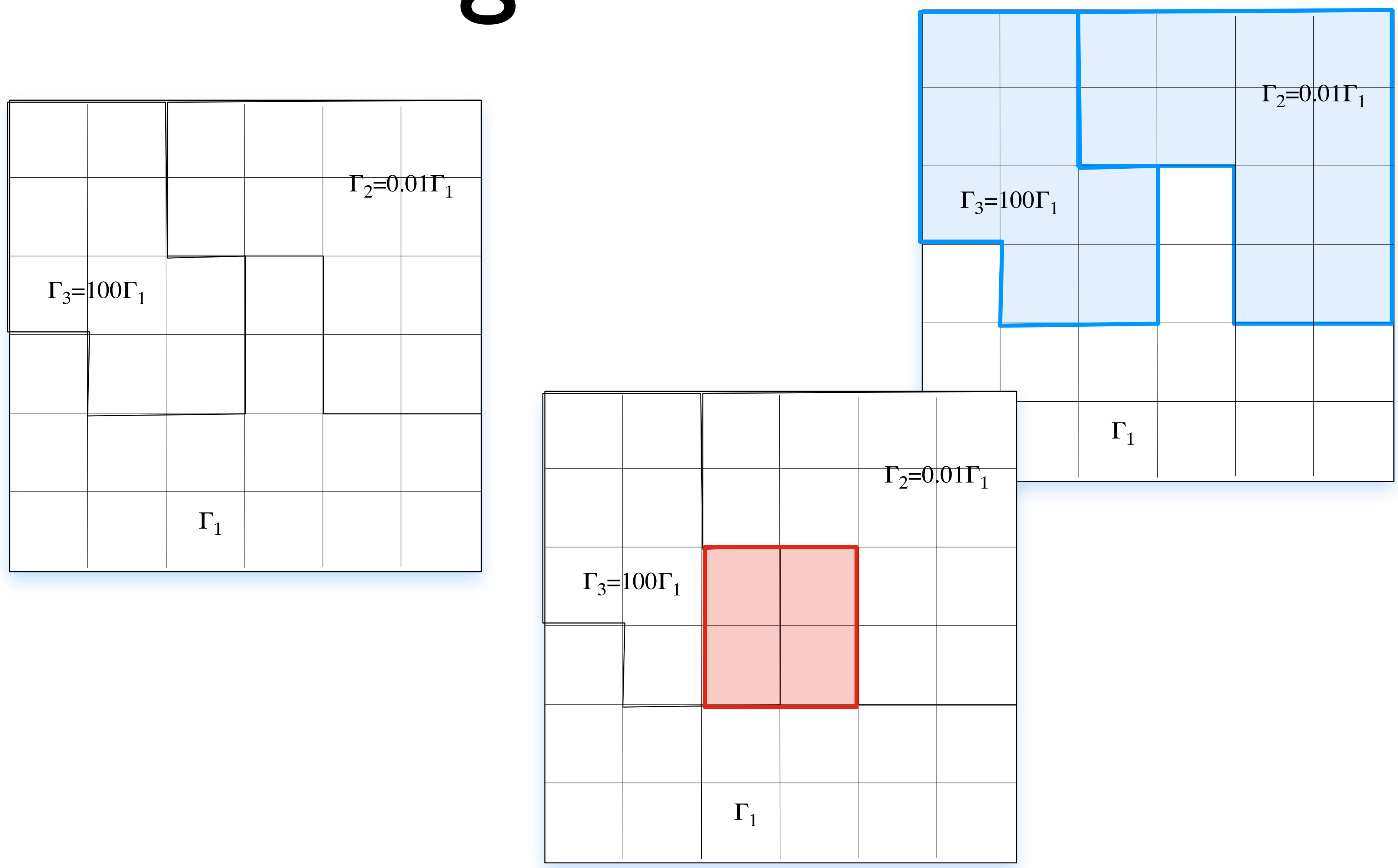
$$\Delta t_{eff} = 4\frac{\rho\Delta x^2}{\Gamma_1}\frac{1+C}{C(3+C)}$$



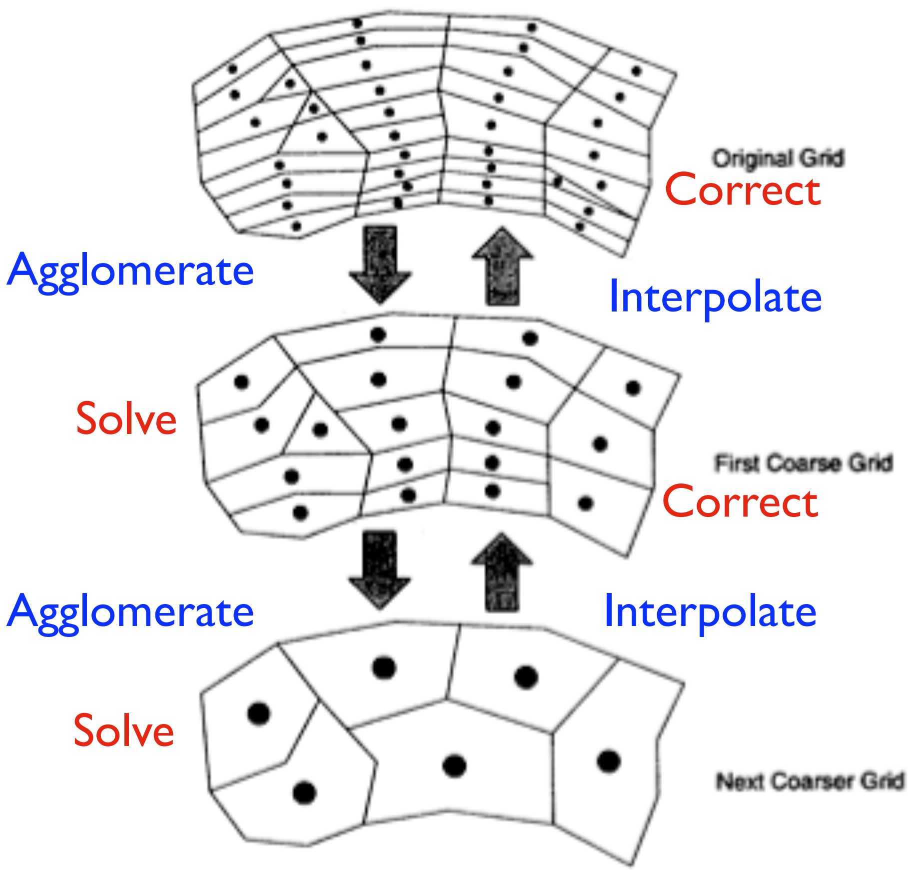
$$\Delta t_{eff} = 4\frac{\rho\Delta x^2}{\Gamma_1}\frac{(1+10)}{10(3+10)}$$

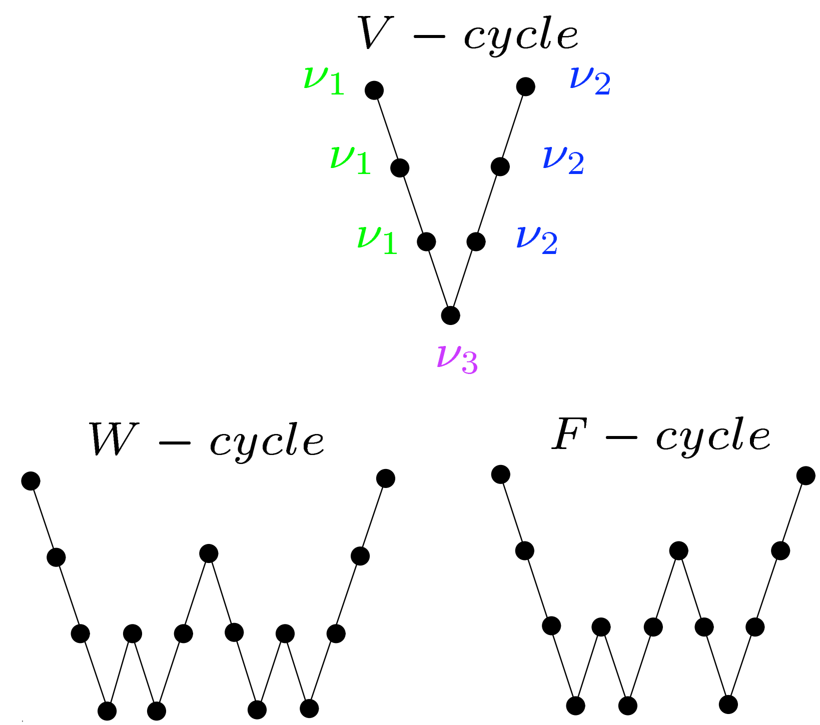


Agglomeration Algorithm



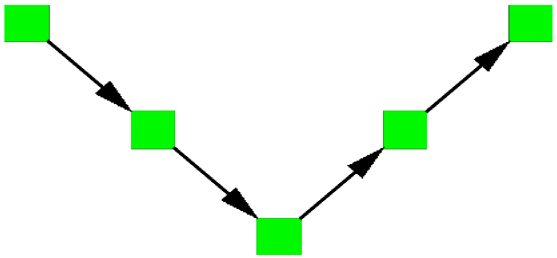
Multigrid Cycles



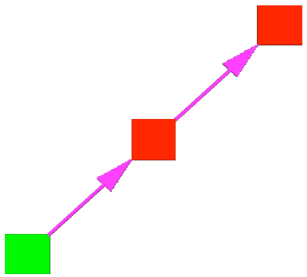


Multigrid Configurations

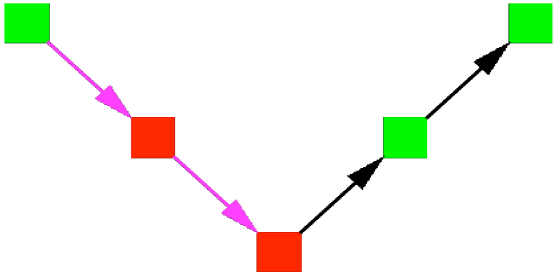
- **Classical: all grid are given**



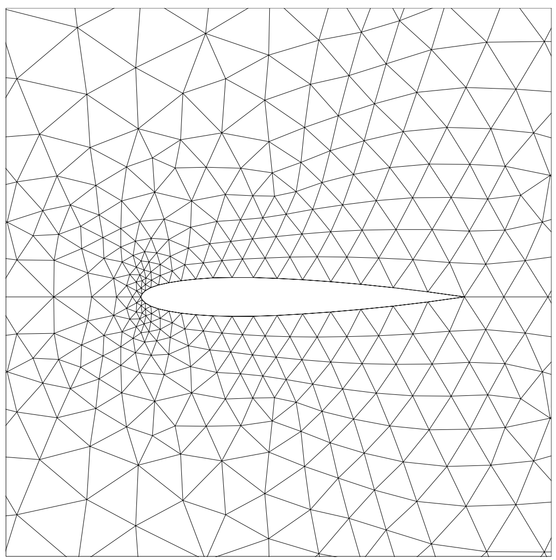
- **Adaptive: coarse grid given - generate fine grids**

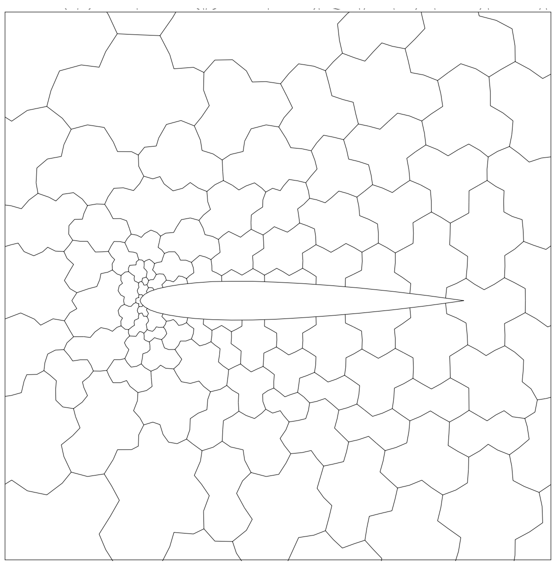


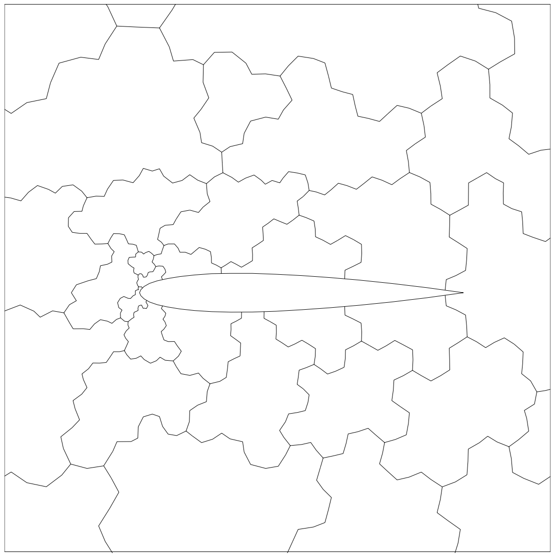
- **Algebraic: fine grid given - construct coarse “grids”**



Demonstration







Conclusion

- Essential Technique for real applications solution
  - Efficient Solver Acceleration
  - Scales with grid size
  - Can be applied to simple Linear Solver and to non-linear Solution Algorithm
  - Suitable to a wide range of applications