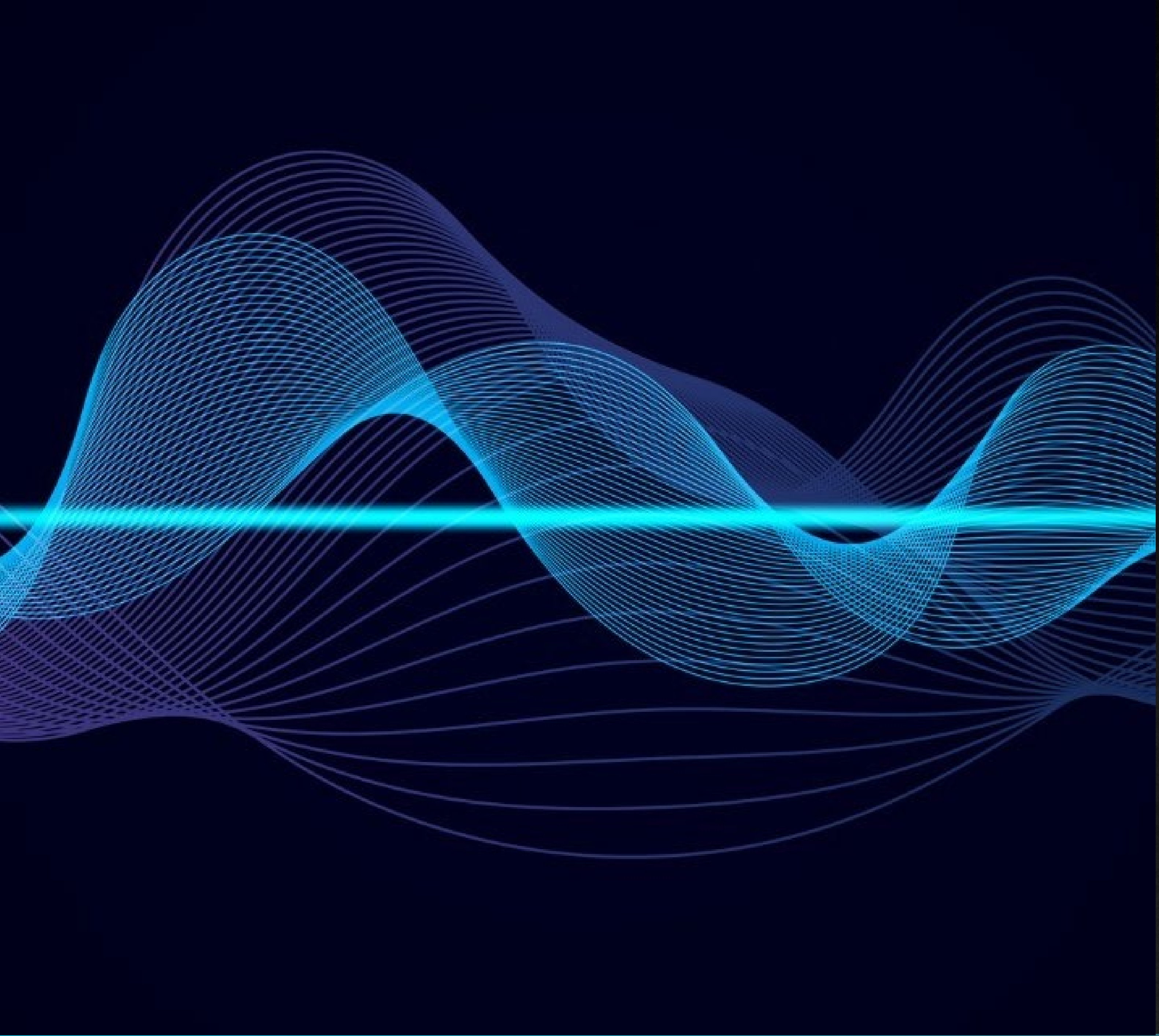


Computational Physics

PHYS 6260

Elliptic Solvers / Multigrid

Announcements:

- HW7: Due Friday 3/29
- Progress report: Due Mon 4/1

Lecture Outline

We will cover these topics

- Elliptic PDEs
- Relaxation method review
- Multigrid method

Different types of PDEs

- Nomenclature comes from the discriminant (B^2 – AC) of the 2nd order PDE form

$$Au_{xx} + 2Bu_{xy} + Cu_{yy} + Du_x + Eu_y + F = 0$$

- Elliptic ($B^2 - AC < 0$): Time independent. Solutions are always smooth even if the boundary conditions are not.
- Parabolic ($B^2 - AC = 0$): Time-dependent. Represent diffusion-like processes. Information travels at infinite speeds. Smooth solution.
- Hyperbolic ($B^2 - AC > 0$): Time-dependent. If the PDE is non-linear, shocks can appear. Information travels at a finite speed. Smoothness depends on initial and boundary conditions.

Elliptic problems in Physics

- Gravitational and electric potentials (Poisson / Laplace Equations)

$$\nabla^2\Phi = 4\pi G\rho \qquad\qquad \nabla^2\Phi = -\frac{\rho}{\epsilon}$$

- Helmholtz equation (time-independent solution of the wave equation)

$$(\alpha + \nabla\cdot\beta\nabla)\Phi = \mathrm{f}(\vec{x})$$

- Often arises by discretizing or separating out time in a PDE

Mixed PDEs in Physics

- Sometimes we have systems with mixed PDE types
- Incompressible hydrodynamics, where the Poisson equation enforces some constraint

$$U_t + U \cdot \nabla U + \nabla p = 0; \quad \nabla \cdot U = 0$$

Mixed PDEs in Physics

- Fluid dynamics with self gravity

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0$$

$$\frac{\partial (\rho U)}{\partial t} + \nabla \cdot (\rho U U) + \nabla p = \rho \nabla \Phi$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho U E + U p) = \rho U \cdot \nabla \Phi$$

$$\nabla^2 \Phi = 4\pi G \rho$$

Elliptic PDEs and Boundary Conditions

- No time-dependent – field responds instantly to the boundary conditions and source
- There is no propagation speed
- To obtain a solution, iterative methods, like the relaxation method, are used
- Treatment of boundary conditions are essential

Review of Relaxation Method

- After making an initial guess, we iterate on the solution
- Jacobi method
  - With the Laplace equation (no source terms), it's basically averaging the adjacent points
  - Iterate until the error reaches a pre-determined tolerance
- Gauss-Seidel method
  - Same as Jacobi method but we use the data as they become available
  - Common form is known as the red-black method

Errors and Norms

- There are many different norms that can be used to determine the error
- General p-norm

$$||e||_p = \left(\Delta x \sum_{i=1}^{N} |e_i|^p \right)^{1/p}$$

- L2 (Euclidean) norm

$$||e||_2 = \left(\Delta x \sum_{i=1}^{N} |e_i|^2 \right)^{1/2}$$

- L1 norm

$$||e||_1 = \Delta x \sum_{i=1}^{N} |e_i|$$

- Inf norm

$$||e||_\infty = \max_i \{|e_i|\}$$

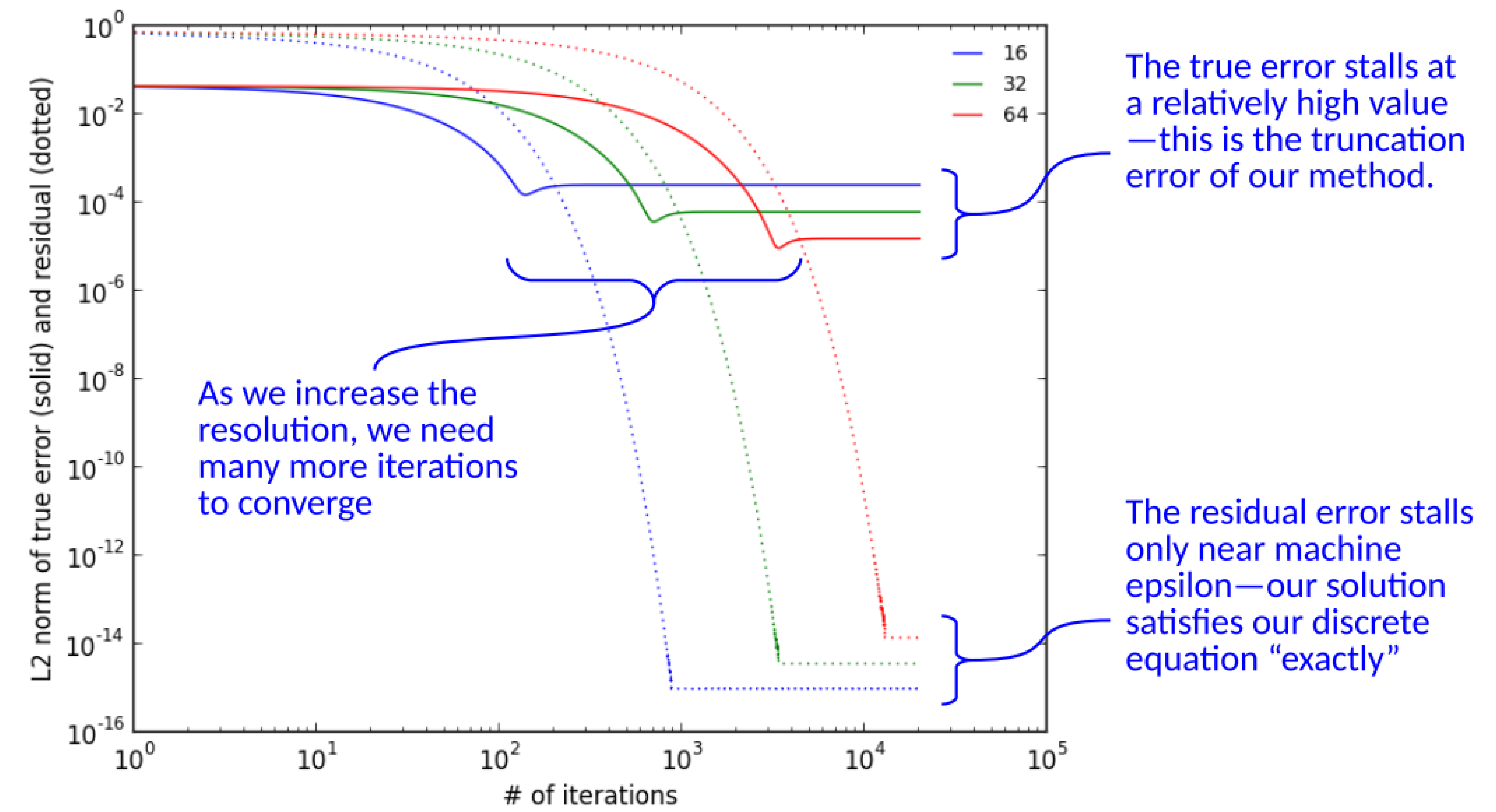
Errors and Norms

- The norm gives us a single number to measure whether we have converged

- The choice of norm should not matter – if we converge, we should converge in all norms
- L2 falls between L1 and the inf-norm in magnitude
- L1 and L2 are more “global” – all values contribute

- Errors e_i can either be (1) deviation from analytical solution or (2) more generally, the change from the previous iteration
- Stop when $||e|| < \epsilon||f||$, where ϵ is a constant and predetermined

Errors and Norms



Relaxation Observations

- Observe that the higher-frequency (shorter wavelength) errors smooth away fastest

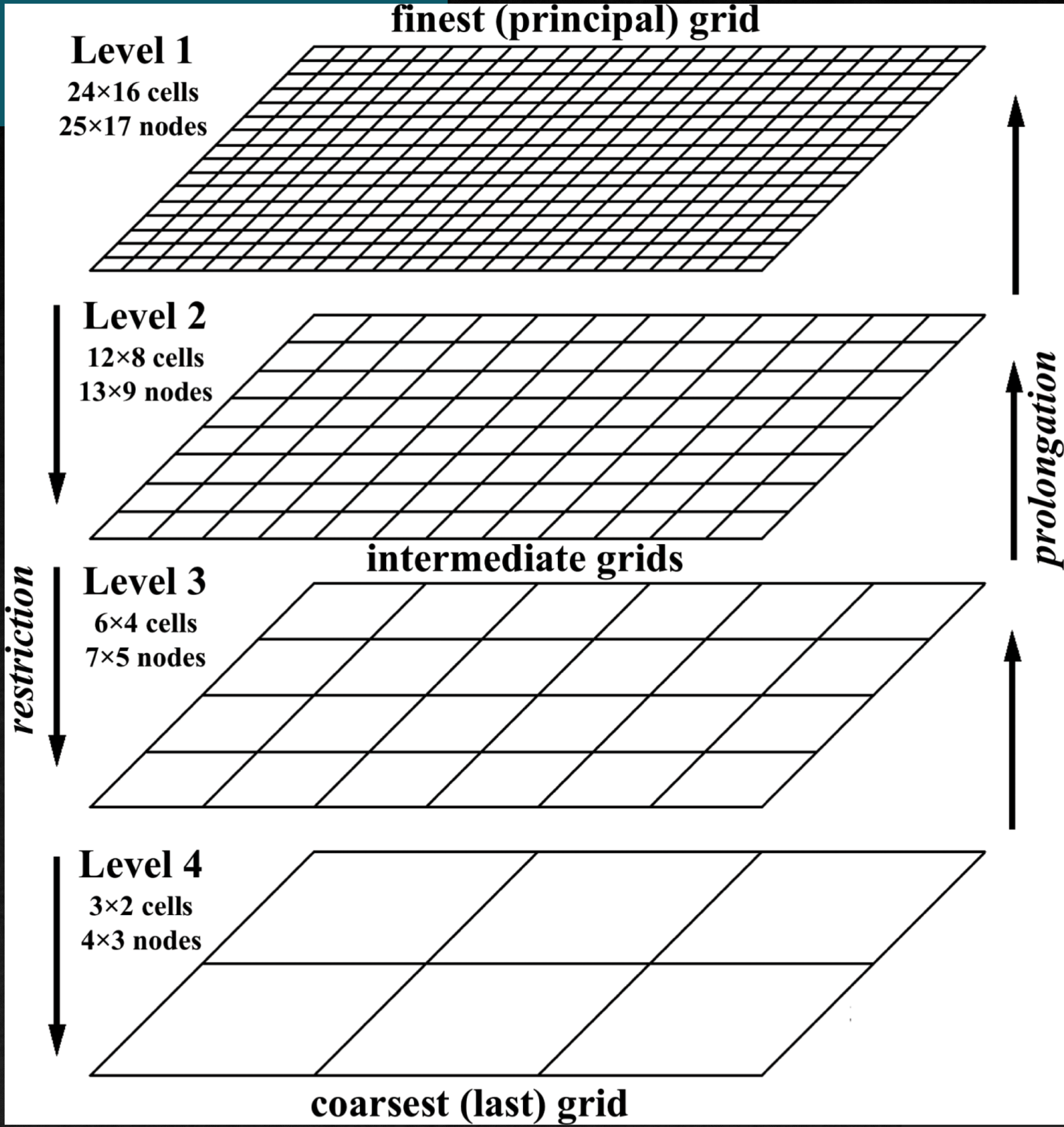
- In the relaxation method, every zone is linked to every other zone
  - If an error has a wavelength of N zones, then N iterations are required to communicate it across

- We can use this feature to accelerate the solution

→ Multigrid solver :: variable resolution during the solve

Multigrid method

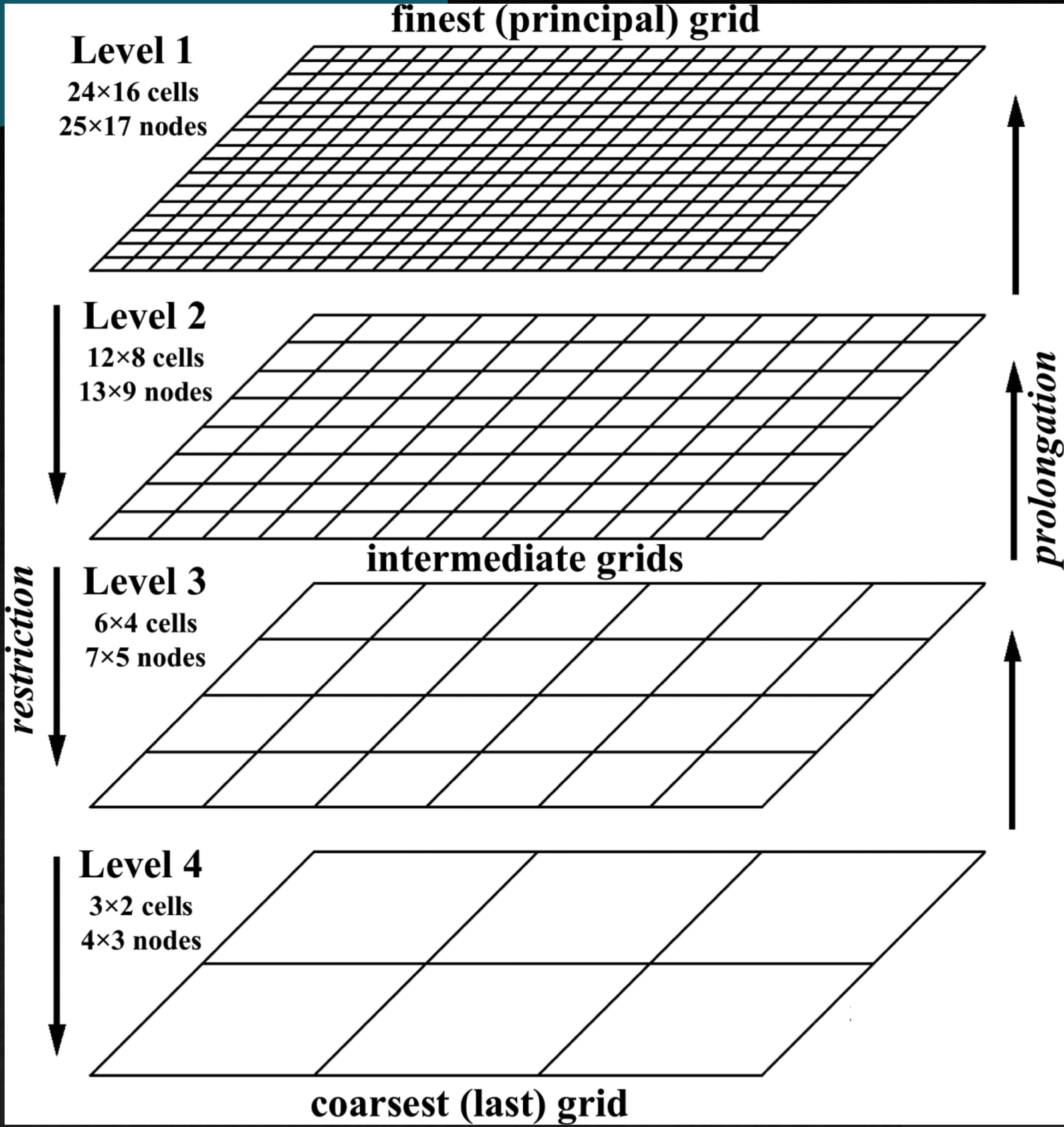
- Multigrid is a widely used method to accelerate the convergence of relaxation
- Eliminates the short wavelength errors on the original grid
- Coarsens the problem and eliminates the formerly long wavelength errors on the new coarser grid



Multigrid method

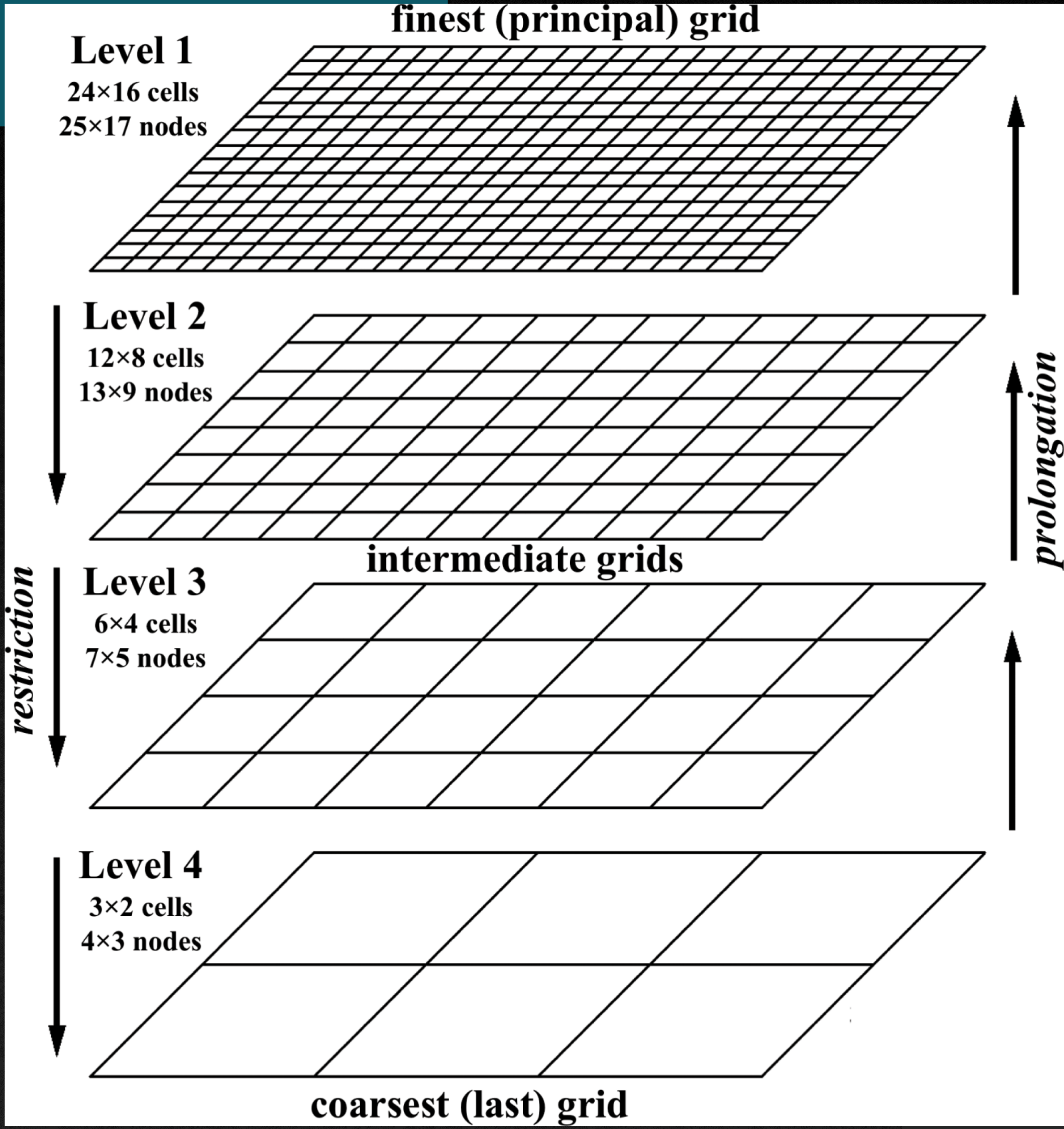
- Relies on a method to move the solution up and down a hierarchy of grids

- By coarsening the data, we accelerate the convergence as compared to straight relaxation



Multigrid method

- We need a way to transfer the data back and forth between the coarse and fine grids
- Restriction: take fine data and transfer it to the coarse grid
- Prolongation: use coarse data to initialize the finer cells
- The restriction and prolongation operations will depend on the type of the grid
- We will only worry about doubling the resolution (most common)

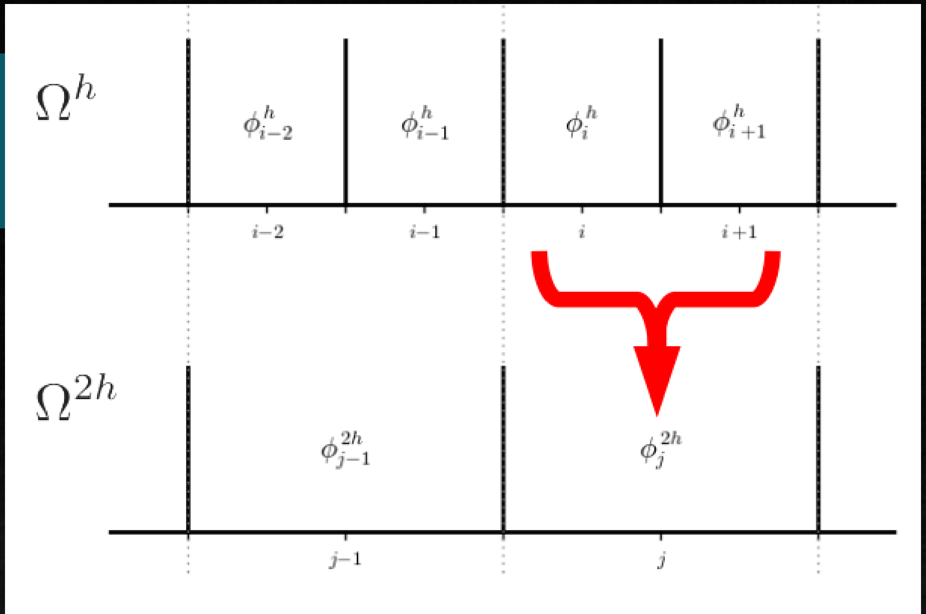


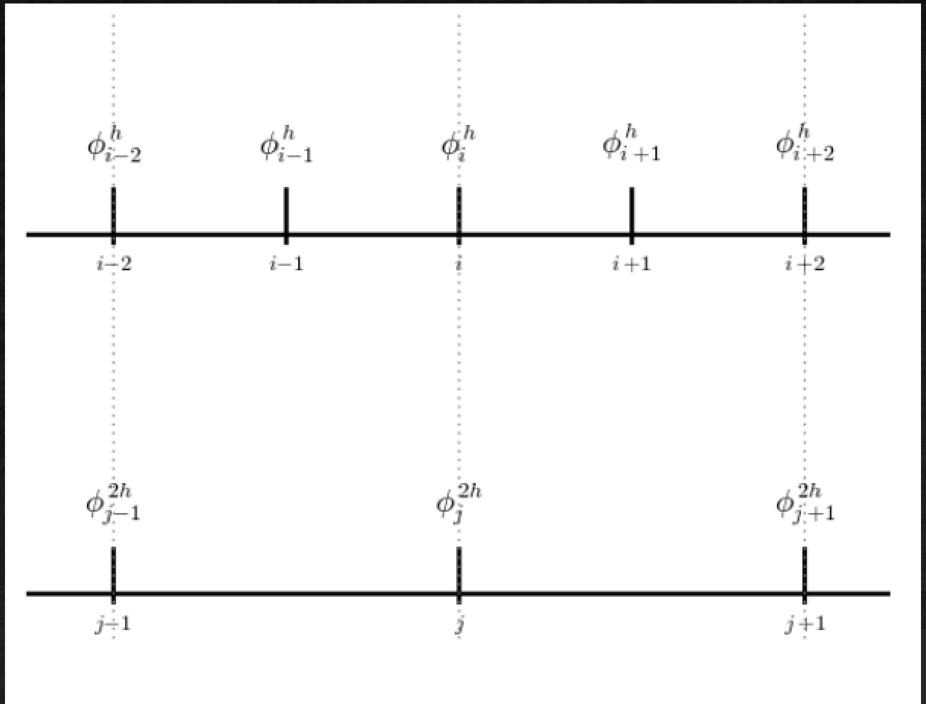
Restriction

- Moving data from fine to coarse mesh
- Conservative quantities: average

$$\phi_j^{2h} = \frac{1}{2}(\phi_i^h + \phi_{i+1}^h)$$

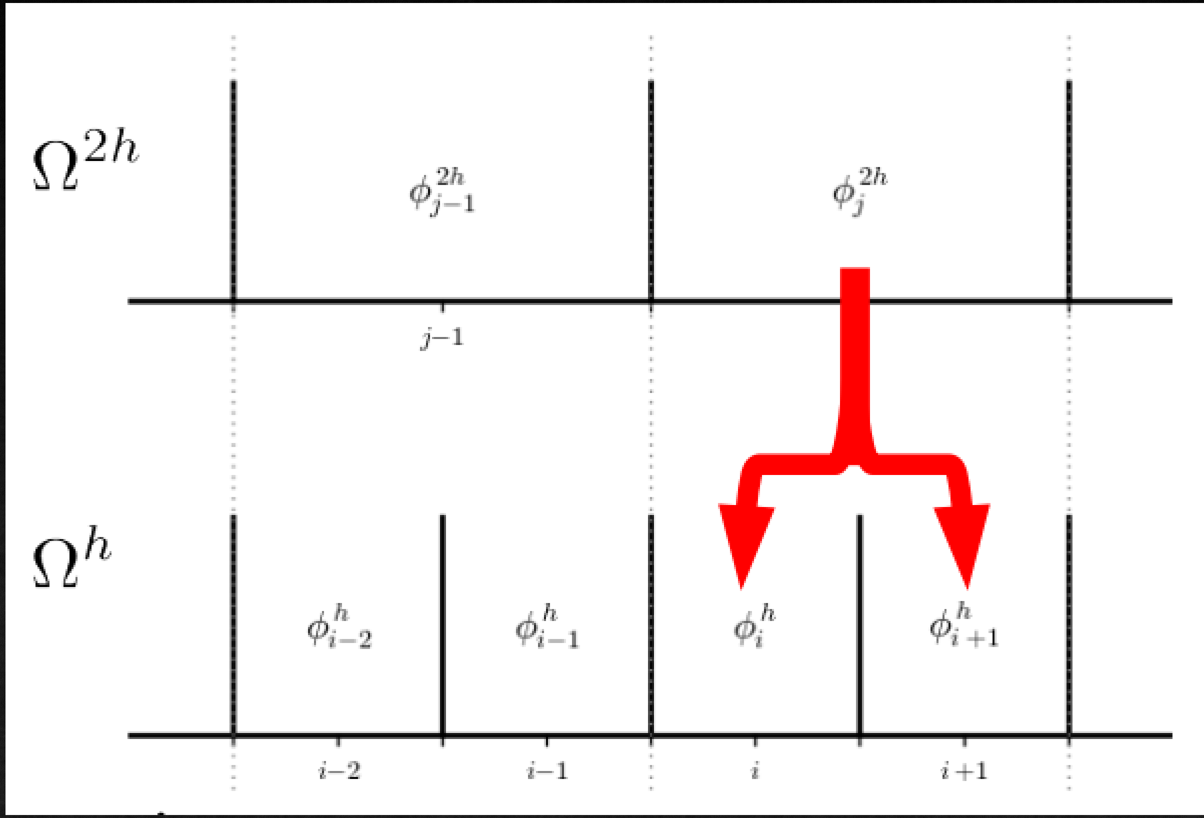
- For finite-differencing grids, only one of the points corresponds exactly to a coarse point
- Simply copy the value





Prolongation

- Moving data from coarse to fine mesh
- Finite-volume mesh: always 2 fine cells within a single coarse cell
- Simple method: direct injection
$$\phi_i^h = \phi_j^{2h}; \quad \phi_{i+1}^h = \phi_j^{2h}$$
- Better: linear reconstruction
$$\phi(x) = m(x - x_j) + \phi_j^{2h}$$
$$m = (\phi_{j+1}^{2h} - \phi_{j-1}^{2h})/(2\Delta x^{2h})$$



Implementation

- Let's first denote the discretized Laplacian operator as L

$$\nabla^2\Phi = f \;\rightarrow L^h\Phi^h = f^h$$

where the superscript h denotes the solution with resolution = h

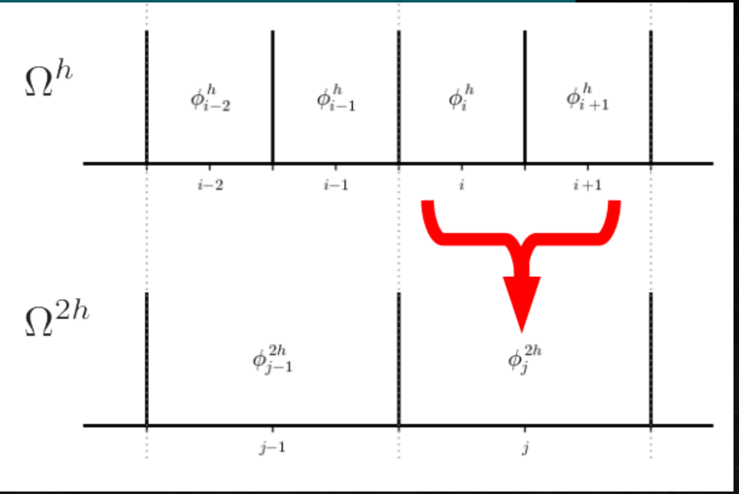
- Denote the approximate solution as v and the error is $e = \Phi - v$
- Our operator is linear, giving the residual r

$$Le = L\Phi - Lv = f - Lv = r$$

- Our error satisfies the same type of equation with the residual as the source
- We can relax on the error and use it to correct our current guess v

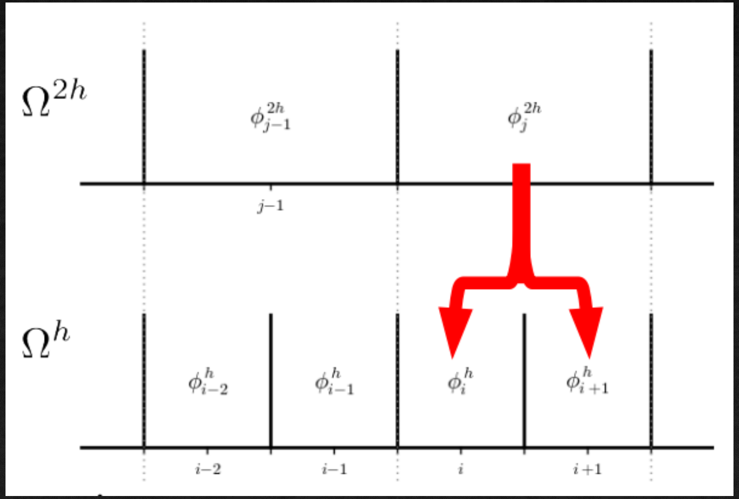
Implementation: two-grid correction scheme

1. Relax $L^h\Phi^h = f^h$ on h to obtain the approximation v^h
   a) Compute the residual: $r^h = f^h - Lv^h$
   b) Restrict r^h to 2h producing r^{2h}
   c) Solve $L^{2h}e^{2h} = r^{2h}$ on 2h
   d) Prolong e^{2h} to h to produce e^h



2. Correct the approximation on h with the error from 2h through

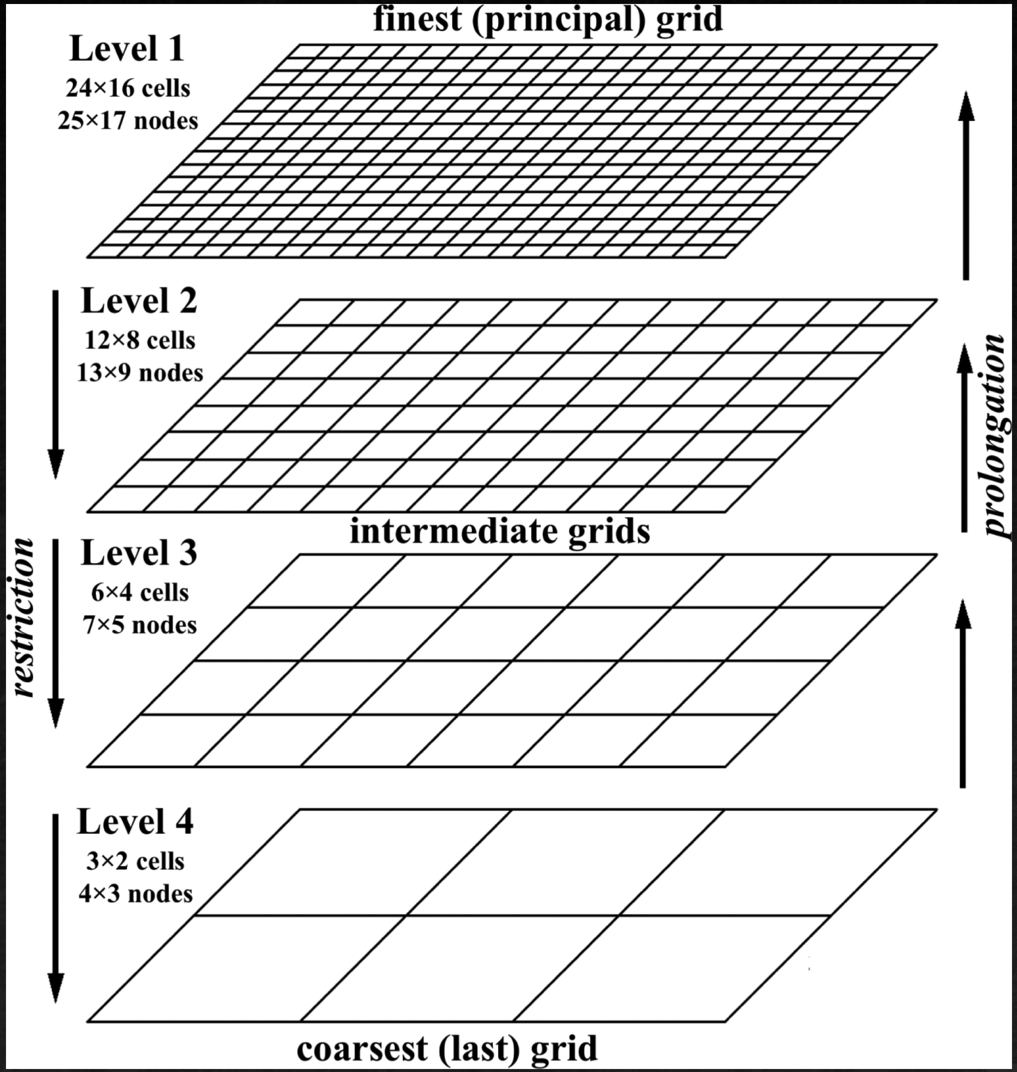
$$v^h \leftarrow v^h + e^{2h}$$



3. Relax $L^h\Phi^h = f^h$ with initial guess v^h

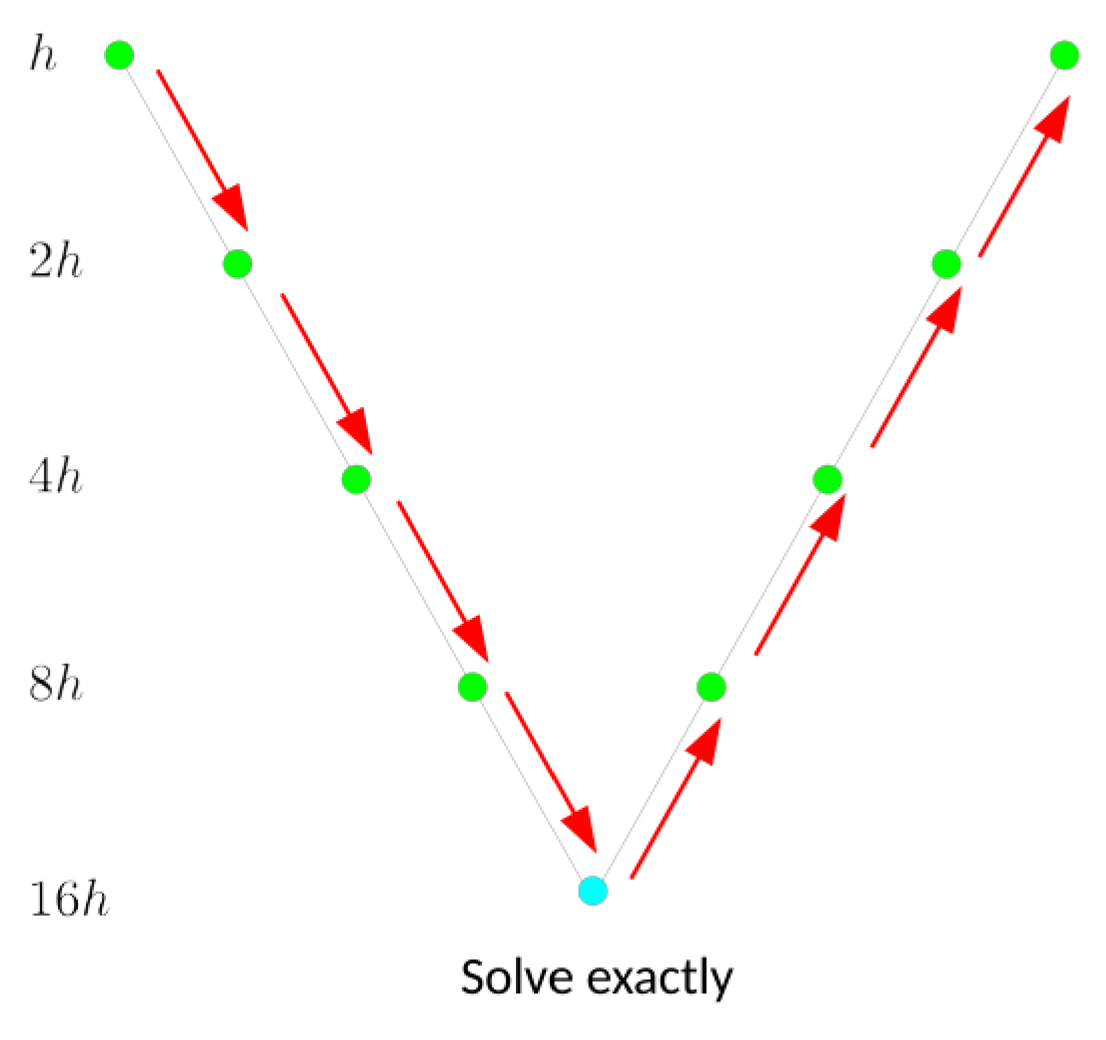
Recursive two-grid correction scheme

- We still have a step that says we need to "solve" on the coarse grid
- Try to avoid large iteration counts by recursively restricting the grids until reaching the smallest grid possible
- Exactly solve the problem (e.g. direct relaxation) on the coarsest grid



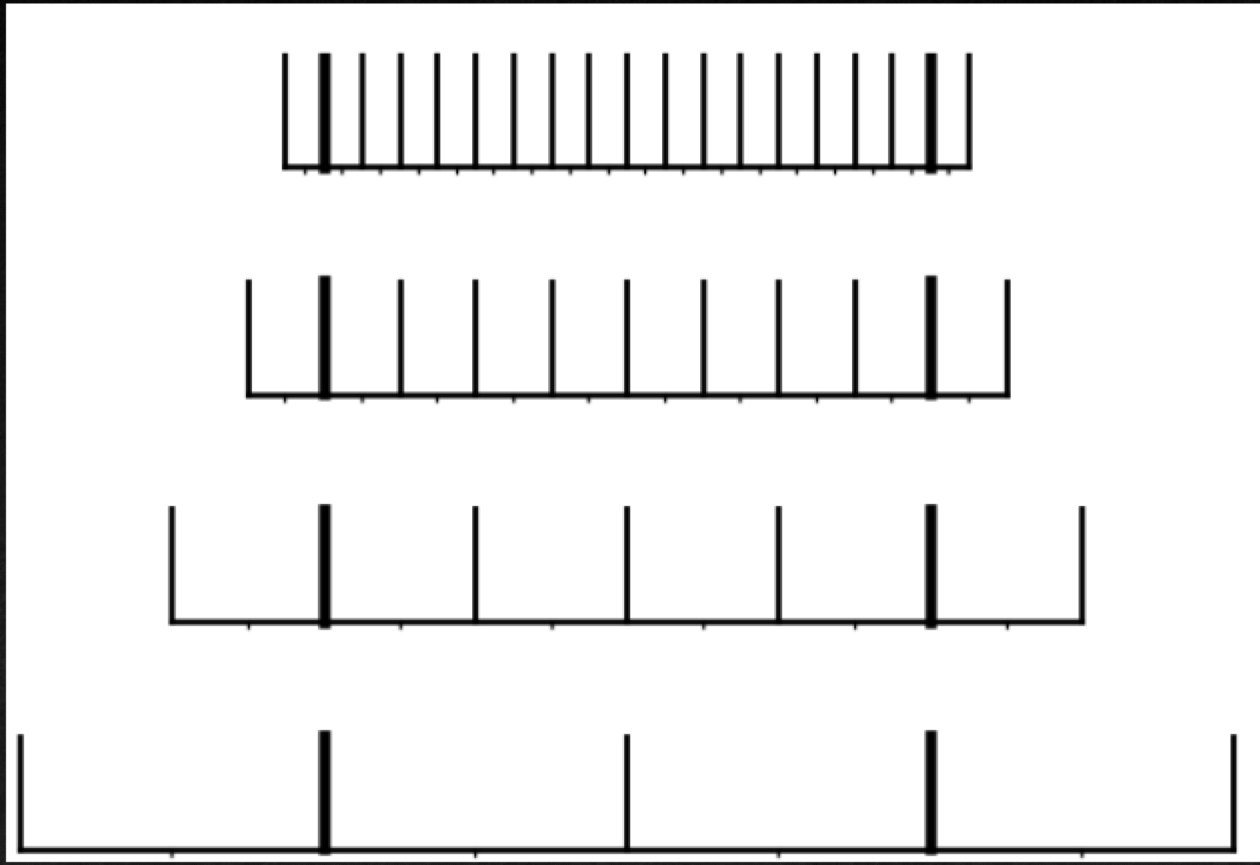
V-cycles

- Simplest hierarchy
- At each level, do a few smoothing operations to eliminate the short wavelength errors
- Coarsen and solve the error equation on the coarse grid
- Once you reach the coarsest grid, solve exactly
- On the upward part, transfer the error, correct, and smooth a few times before passing it to the next higher level



Bottom solver

- Consider the hierarchy on the right
  - Each has a single ghost cell
  - Coarsest grid has 2 zones in the interior – minimum to enforce BCs



- In general, not restricted to powers of 2 refinement but it will affect the coarsest grid size

- Somes more elaborate methods (like conjugate gradient) are used for the bottom solve

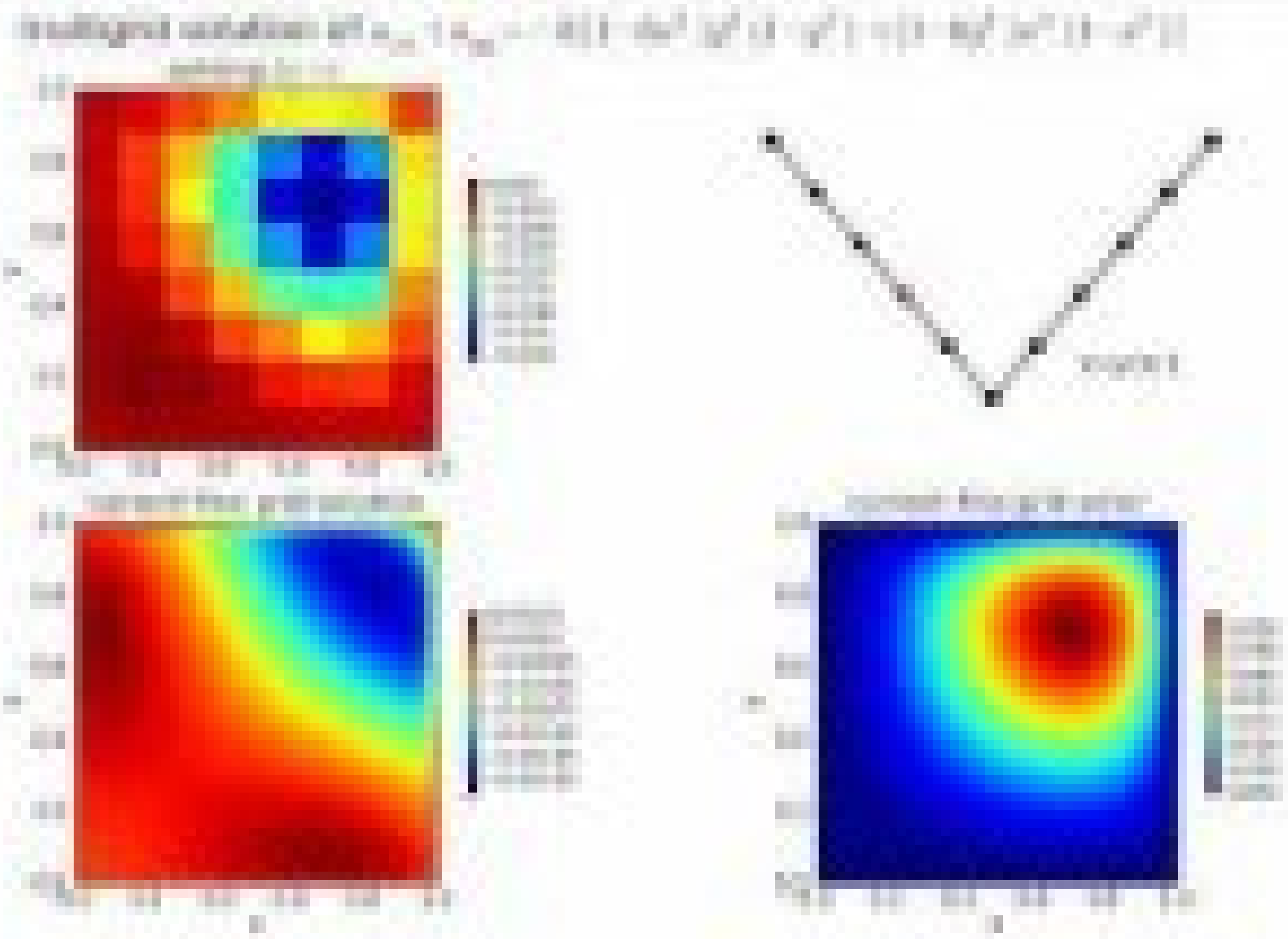
Stopping criteria

- Continue to iterate (perform V-cycle after V-cycle) until convergence is reached

- Typical convergence criteria

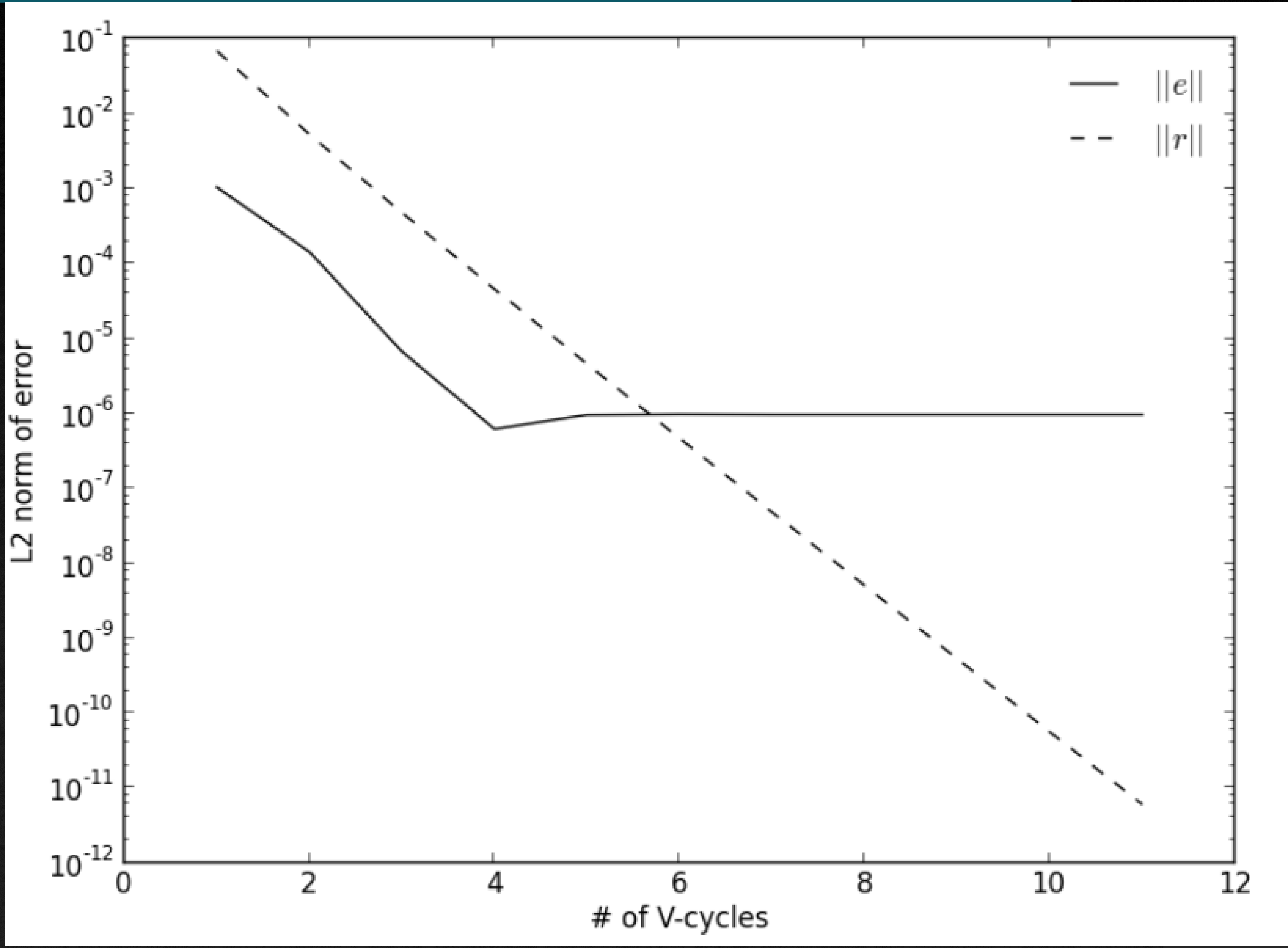
$$||r|| < \epsilon||f||$$

- The source norm provides a scale to measure against
- If the source is zero, we stop when $||r|| < \epsilon$

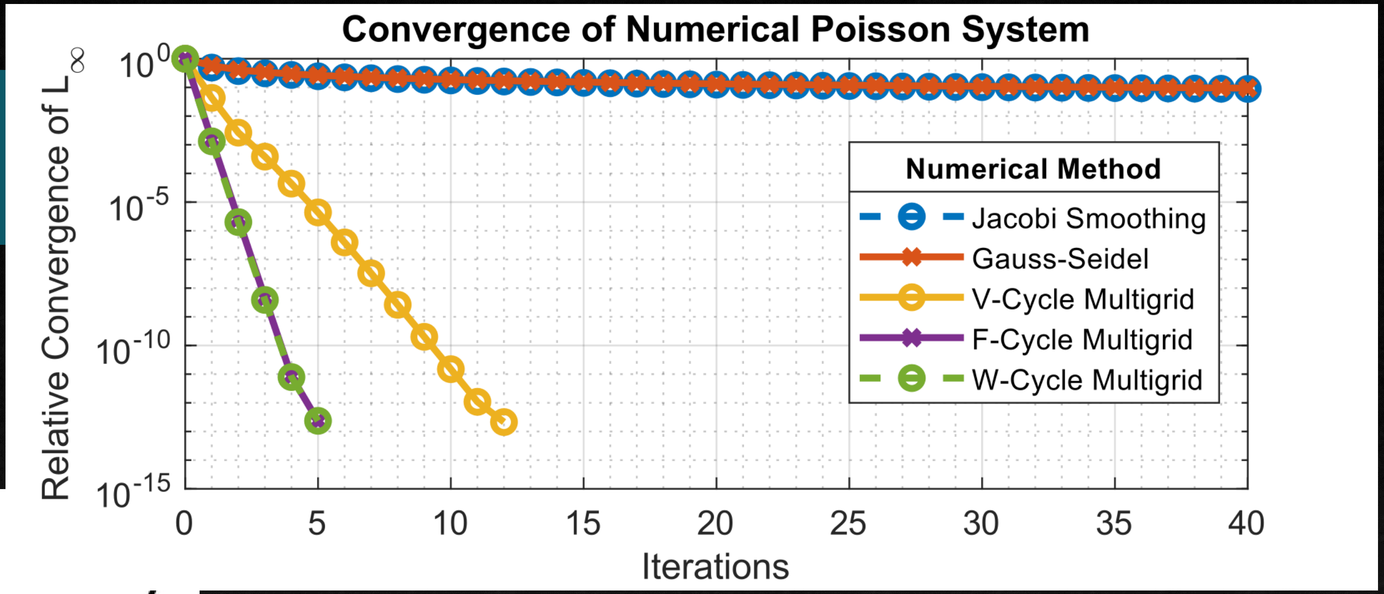


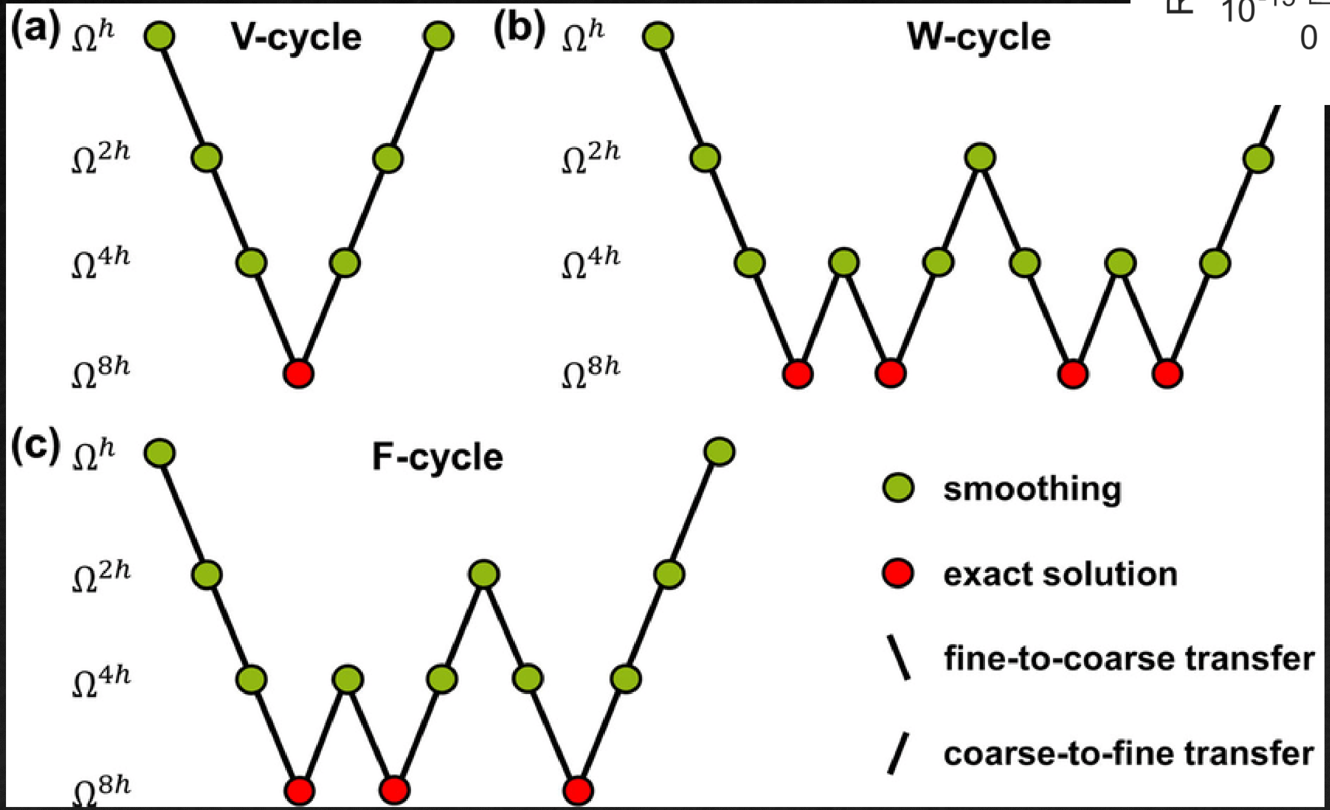
Typical performance

- Notice that each V-cycle reduces the residual by about an order of magnitude, which is a good rule-of-thumb



Other variations





Multigrid V-Cycle: Solving **PHI** in PDE ***f*** **(PHI) = F**