2023 Competition on Computational Astrophysics Problem I

Chan Park (IBS)

2023.01.30 @ UNIST

2023 Winter School on Numerical Relativity and Gravitational Waves

Overview

- Development Routine
  - IDE: VS Code
  - Test: Jupyter Notebook
  - Submission: Git
  - Code Review
- Numerical Analysis
  - Finite Difference Method
  - Iterative Method
  - Multigrid Method
- Problems

Development Routine

VS Code

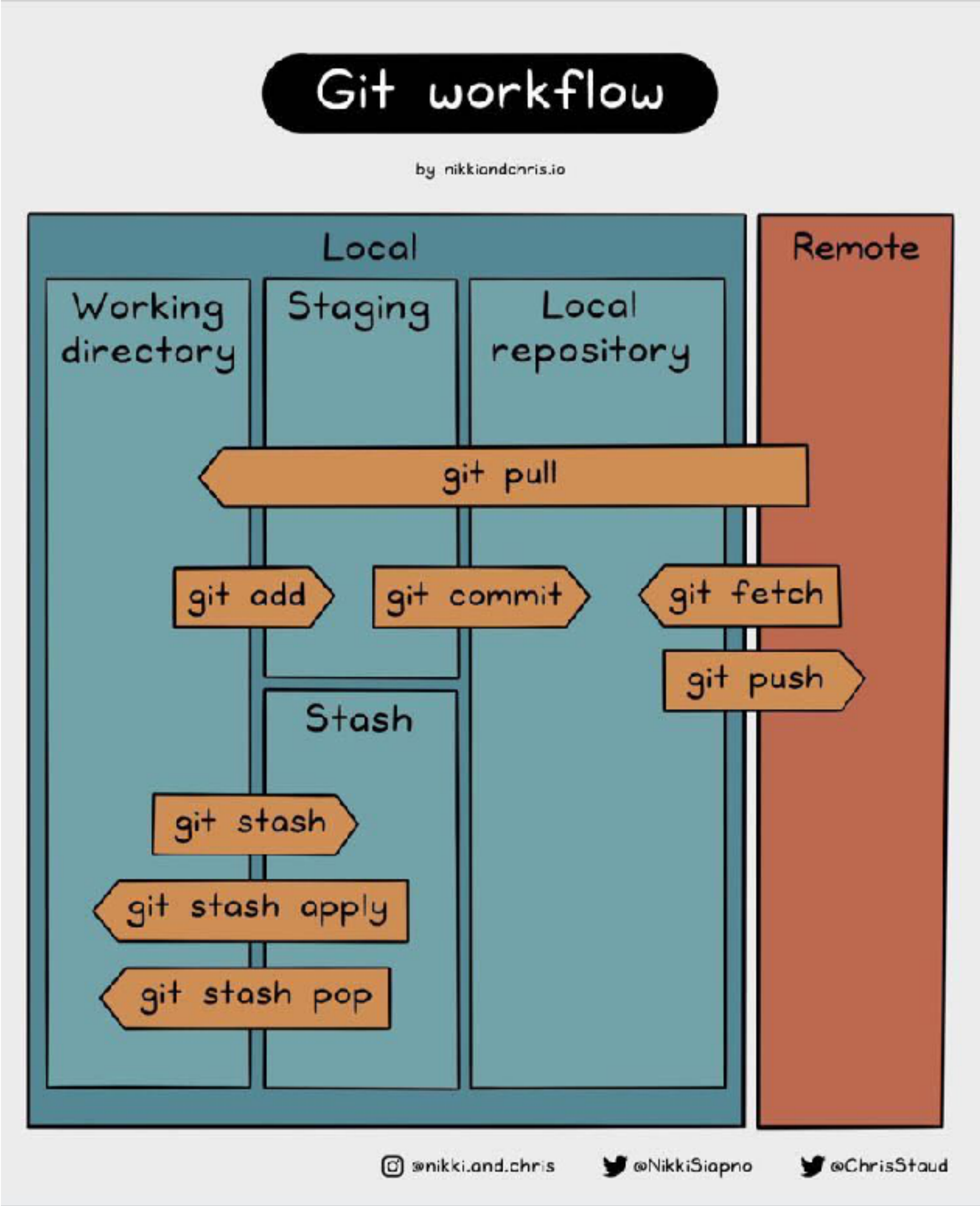
- IDE: Integrated Development Environment
- It can open the filesystem of remote server through SSH.
- Supporting OS: Windows / Mac / Linux
- https://code.visualstudio.com

Jupyter Notebook

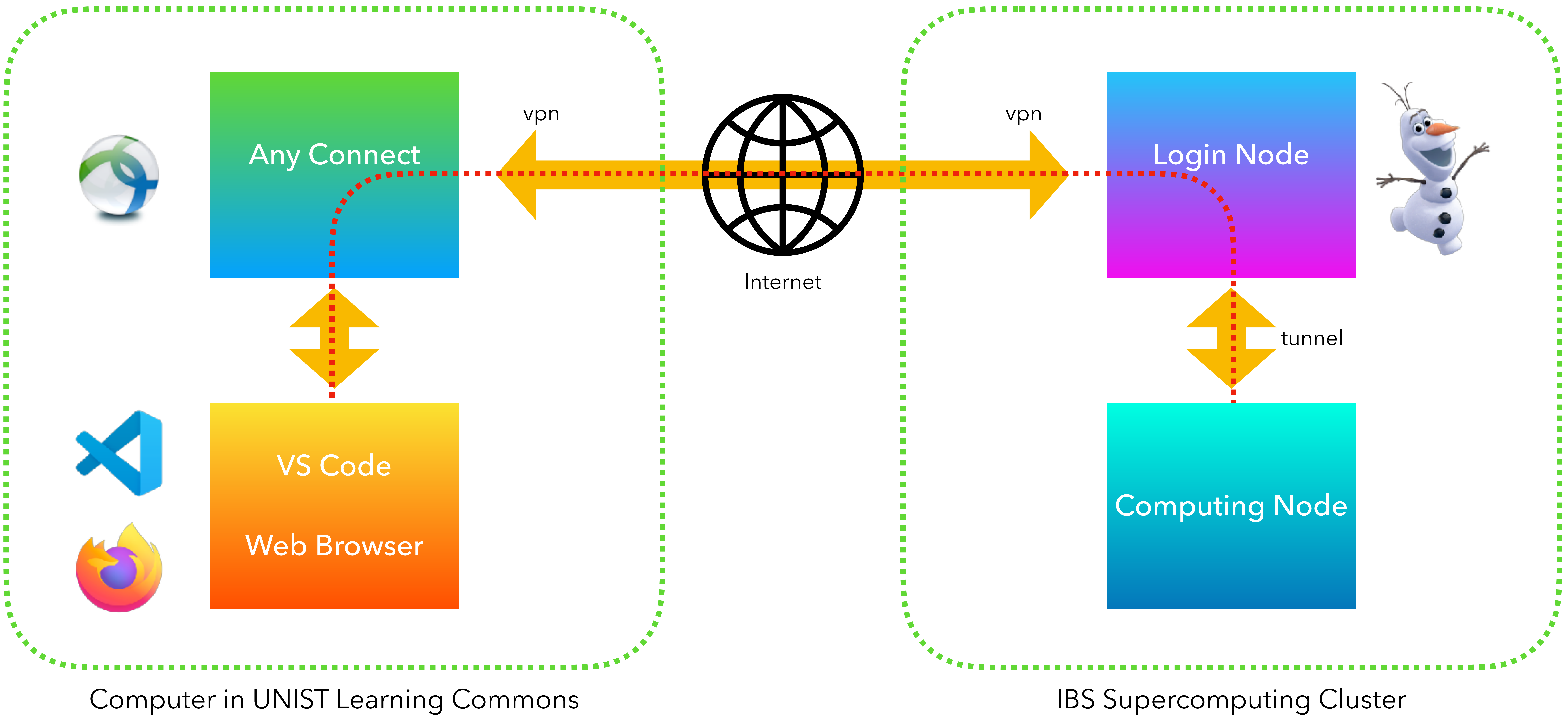
- An interactive interface for Python through web
- In general, you can use any web browser in any OS.
- We will use jupyter notebook in VS Code to test codes.

Git

- Version Control System
- Distribution of example code and submission of answer code are conducted by git.



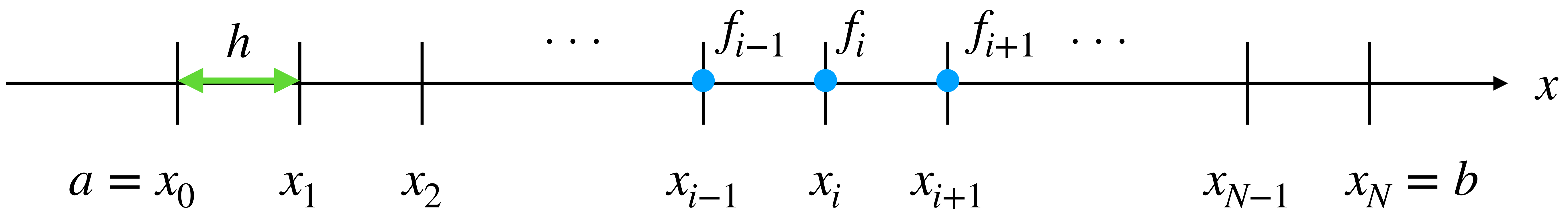
Network Diagram



Numerical Analysis: Finite Difference Method

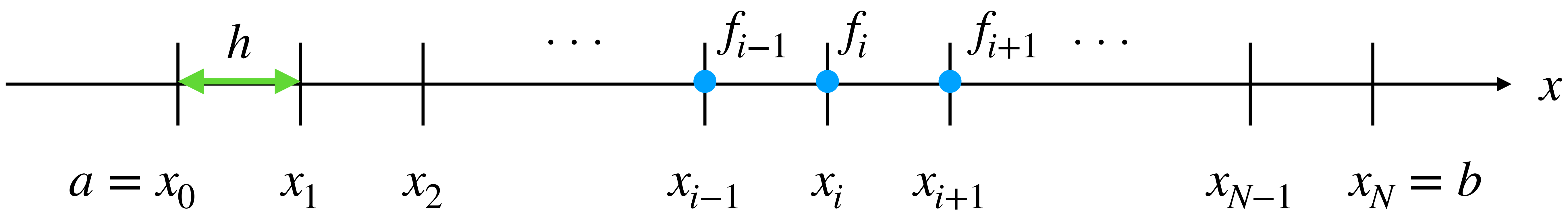
Discretization

- For function $f(x) : [a, b] \to \mathbb{R}$
- Uniform discretization
  - $x_i = ih$ for integer $0 \leq i \leq N$ and $h = (b - a)/N$ where N is the number of cells
  - $f_i = f(x_i)$



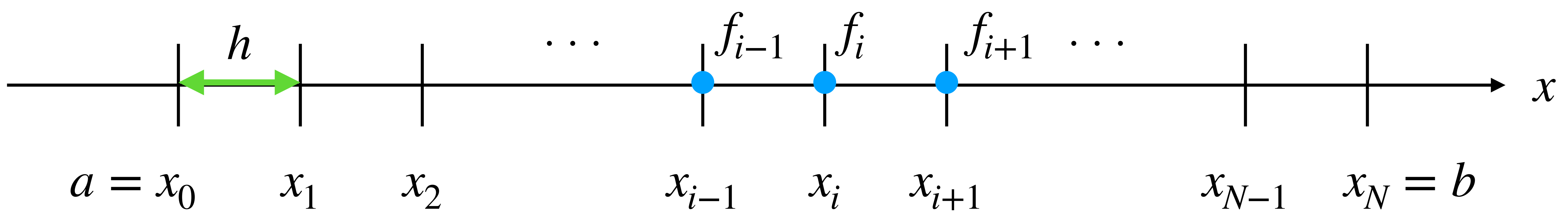
Derivatives by Finite Difference Method (FDM)

- Taylor Expansions
  - $f_{i+1} = f\left(x_{i+1}\right) = f\left(x_i + h\right) = f(x_i) + hf'\left(x_i\right) + \frac{1}{2}h^2 f''\left(x_i\right) + \frac{1}{3!}h^3 f^{(3)}\left(x_i\right) + \frac{1}{6!}h^4 f^{(4)}\left(x_i\right) + O\left(h^5\right)$
  - $f_{i-1} = f\left(x_{i-1}\right) = f\left(x_i - h\right) = f(x_i) - hf'\left(x_i\right) + \frac{1}{2}h^2 f''\left(x_i\right) - \frac{1}{3!}h^3 f^{(3)}\left(x_i\right) + \frac{1}{6!}h^4 f^{(4)}\left(x_i\right) + O\left(h^5\right)$
- First-Order FDM
  - $f'\left(x_i\right) = \frac{1}{h}\left(f_{i+1} - f_i\right) + O\left(h\right)$
- Second-Order FDM
  - $f'\left(x_i\right) = \frac{1}{2h}\left(f_{i+1} - f_{i-1}\right) + O\left(h^2\right)$
  - $f''(x) = \frac{1}{h^2}\left(f_{i+1} + f_{i-1} - 2f_i\right) + O\left(h^2\right)$



Elliptic PDE as a Sparse Matrix

- Simple Problem
  - Given $\frac{d^2 f}{dx^2} = f''(x) = \rho(x)$ with the homogeneous boundary condition: $f(a) = f(b) = 0$, solve $f(x)$
- Second-Order Finite Difference Method
  - $\frac{1}{h^2}\left(f_{i+1} + f_{i-1} - 2f_i\right) \simeq \rho_i$



- In sparse matrix
  - $$\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & \ddots & & \\ & & \ddots & \ddots & & \\ & & & & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix}\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{N-1} \end{bmatrix} \quad \rightarrow \boldsymbol{A}\boldsymbol{f} = \boldsymbol{\rho}$$
- In principle, we can get f_i by the inverse of the above matrix.

Numerical Analysis: Iterative Method

Difficulties on Solving Linear System

- Linear System
  - $$\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & \ddots & & \\ & & \ddots & \ddots & & \\ & & & & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix}\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{N-1} \end{bmatrix}$$
  - $\boldsymbol{A}f = \rho \rightarrow f = \boldsymbol{A}^{-1}\rho$
- Matrix inversion of large matrix requires a tremendous amount of calculation and storage space. Eventually solving in this way is almost impossible.
- We seek an approximate solution by iterative method instead of exact solution of the linear system.
- The error of approximate solution can be controlled as desired.

Gauss-Seidel Method

- From the FDM
  - $\frac{1}{h^2}\left(f_{i+1} + f_{i-1} - 2f_i\right) \simeq \rho_i$
- We get relaxation as
  - $f_i \simeq \frac{1}{2}\left(f_{i+1} + f_{i-1} - \rho_i h^2\right)$
- We set iterative algorithm as
- For a number of steps
  - For $i \in [1,N)$
    - $f_i \leftarrow \frac{1}{2}\left(f_{i+1} + f_{i-1} - \rho_i h^2\right)$

i	0	...	i-1	i	i+1	...	N
f	0	...	f[i-1] →	f[i]	← f[i+1]	...	0
rho		...		rho[i] ↑		...	

Jacobian Method

- For a number of steps
  - For $i \in [1, N)$
    - $f_i^{\text{new}} \leftarrow \frac{1}{2}\left(f_{i+1} + f_{i-1} - \rho_i h^2\right)$
  - switch$\left(f_i^{\text{new}}, f_i\right)$

i	0	...	i-1	i	i+1	...	N
f	0	...	f[i-1]	f[i]	f[i+1]	...	0
f_new		...		f_new[i]		...	
rho		...		rho[i]		...	

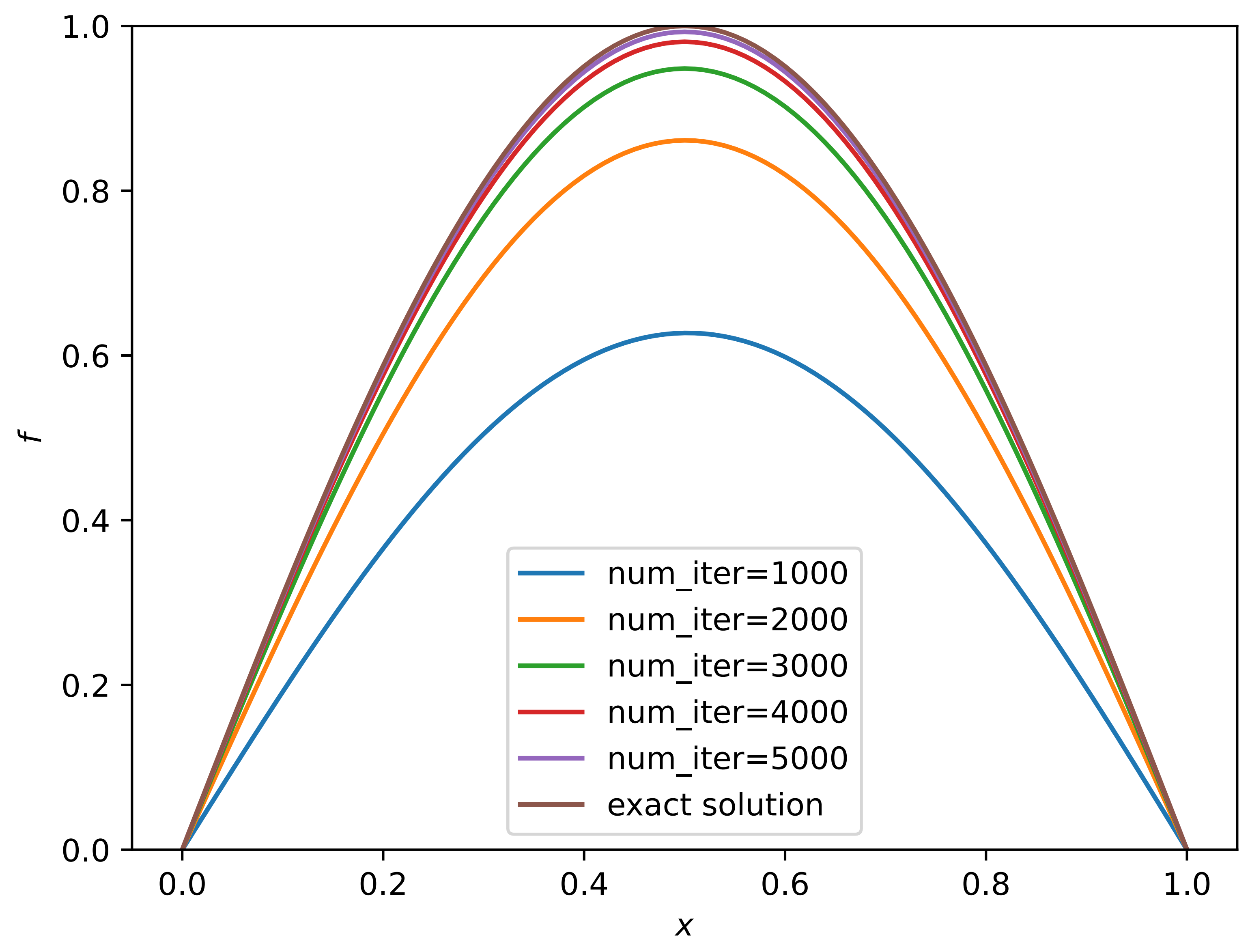
Red-Black Ordering Method

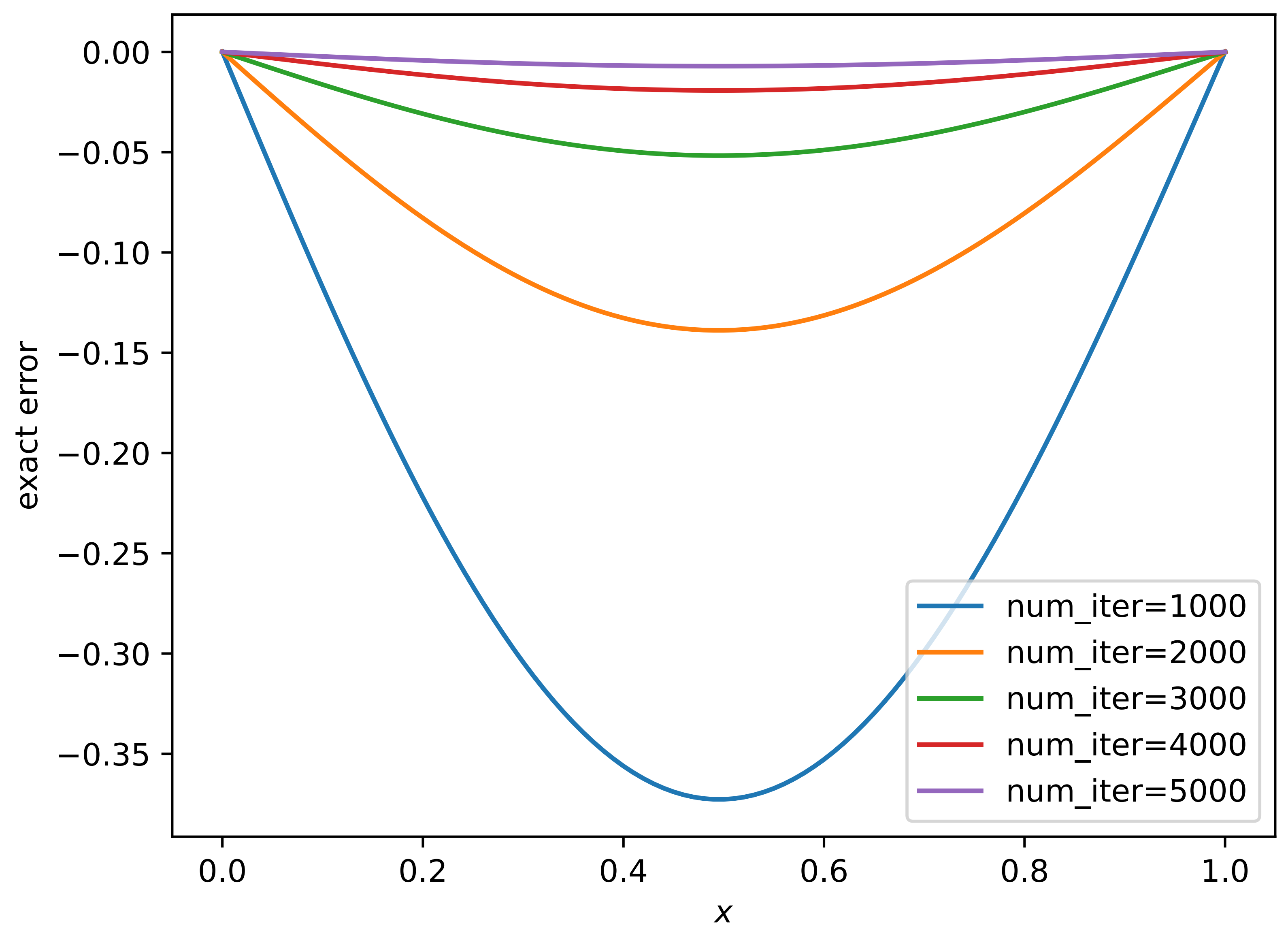
- For a number of steps
  - For $i \in [1, N)$ by step 2
    - $f_i \leftarrow \frac{1}{2}\left(f_{i+1} + f_{i-1} - \rho_i h^2\right)$
  - For $i \in [2, N)$ by step 2
    - $f_i \leftarrow \frac{1}{2}\left(f_{i+1} + f_{i-1} - \rho_i h^2\right)$

i	0	1	2	...	N−2	N−1	N
f	0	...	...	...	...	...	0
rho		...	...	...	...	...	

Result

- Example
    - $\rho(x) = -\pi^2 \sin(\pi x)$
- Exact Solution
    - $f(x) = \sin(\pi x)$
- Number of Cells
    - $N = 100$
- As the number of iteration increases, the numerical solution converges to $\sin(\pi x)$.





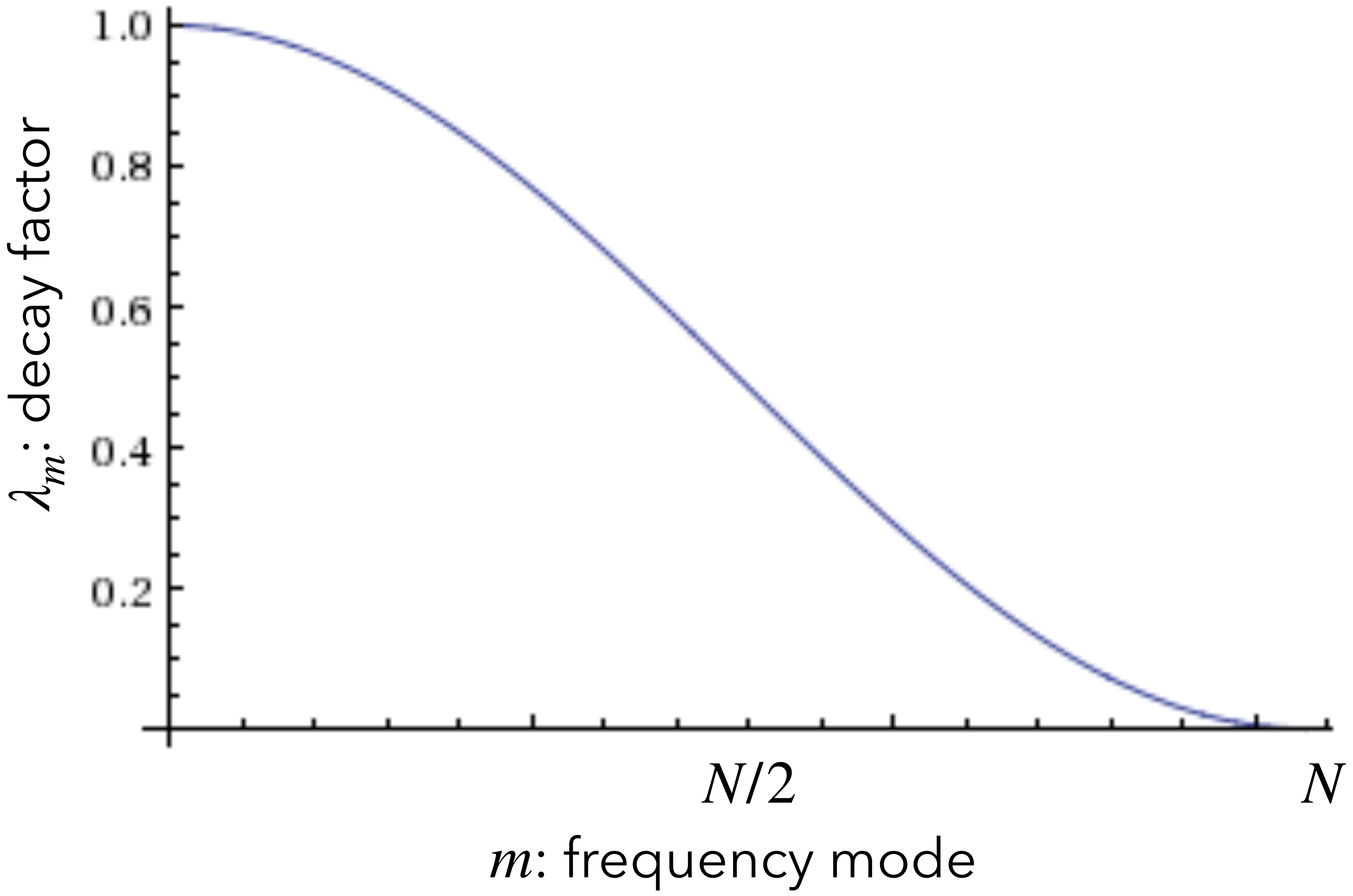
Residual

- How to measure the convergence when we don't know an exact solution?
- Residual
  - $r = \rho - \boldsymbol{A}f$ goes to 0 when f converges to the solution
- Exact error
  - $e = \bar{f} - f$ where $\bar{f}$ is the exact solution
- Relation between the exact error and the residual
  - $\boldsymbol{A}e = r$ because $\boldsymbol{A}\bar{f} = \rho$
- Solving $\boldsymbol{A}e = r$ is equivalent to solving the original equation $\boldsymbol{A}f = \rho$

Numerical Analysis: Multigrid

Difficulties on High Frequency Modes

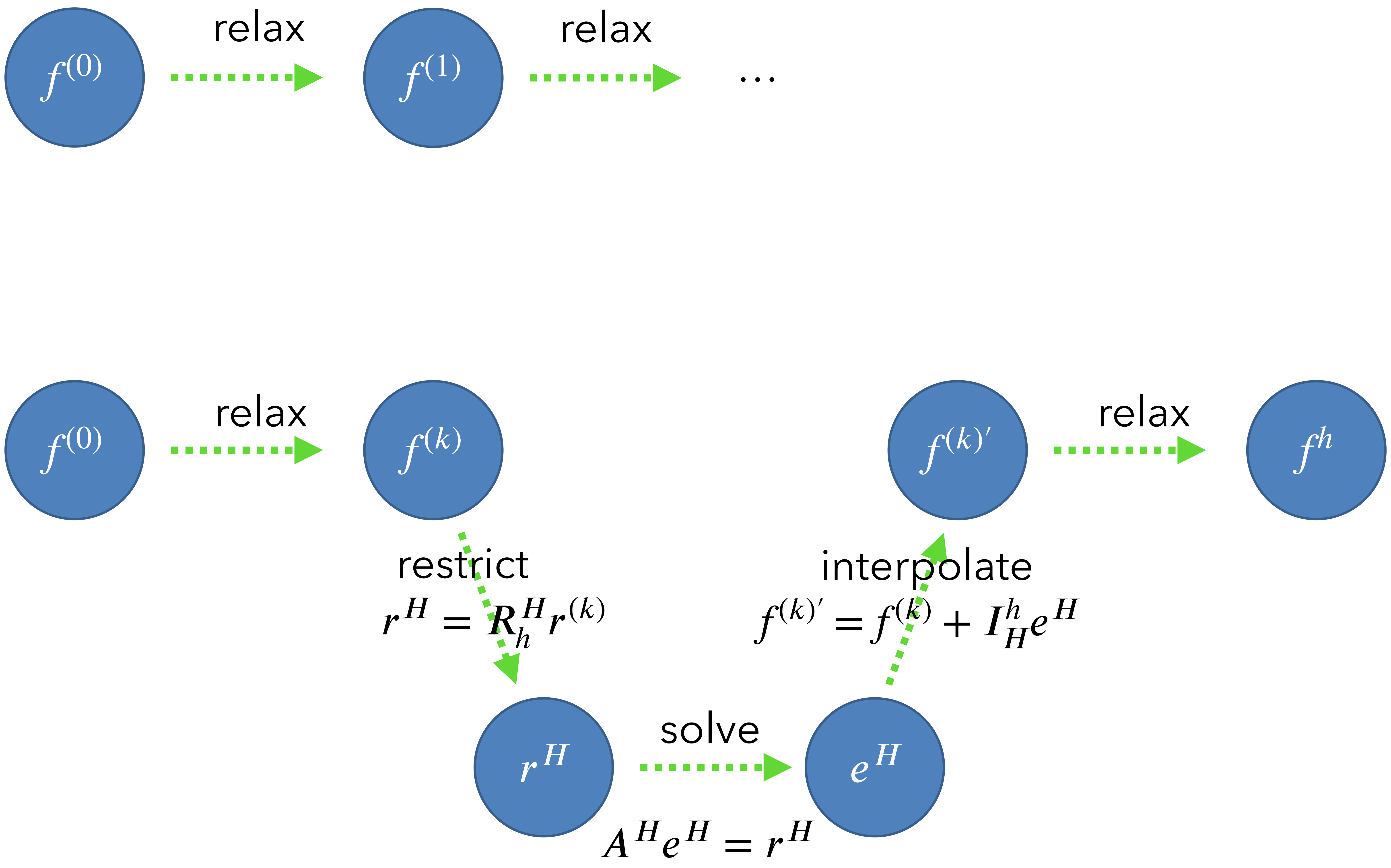
- Residual-Error equation
  - $\boldsymbol{A}e = r$
- An error of m mode is decay by the factor λ_m
  - $\left(\lambda_m\right)^n$ where n is the number of iterations
- Can we devise a method that efficiently decay low mode errors?
- How about to use a coarse grid?



Decay factors for frequency modes in the iterative method

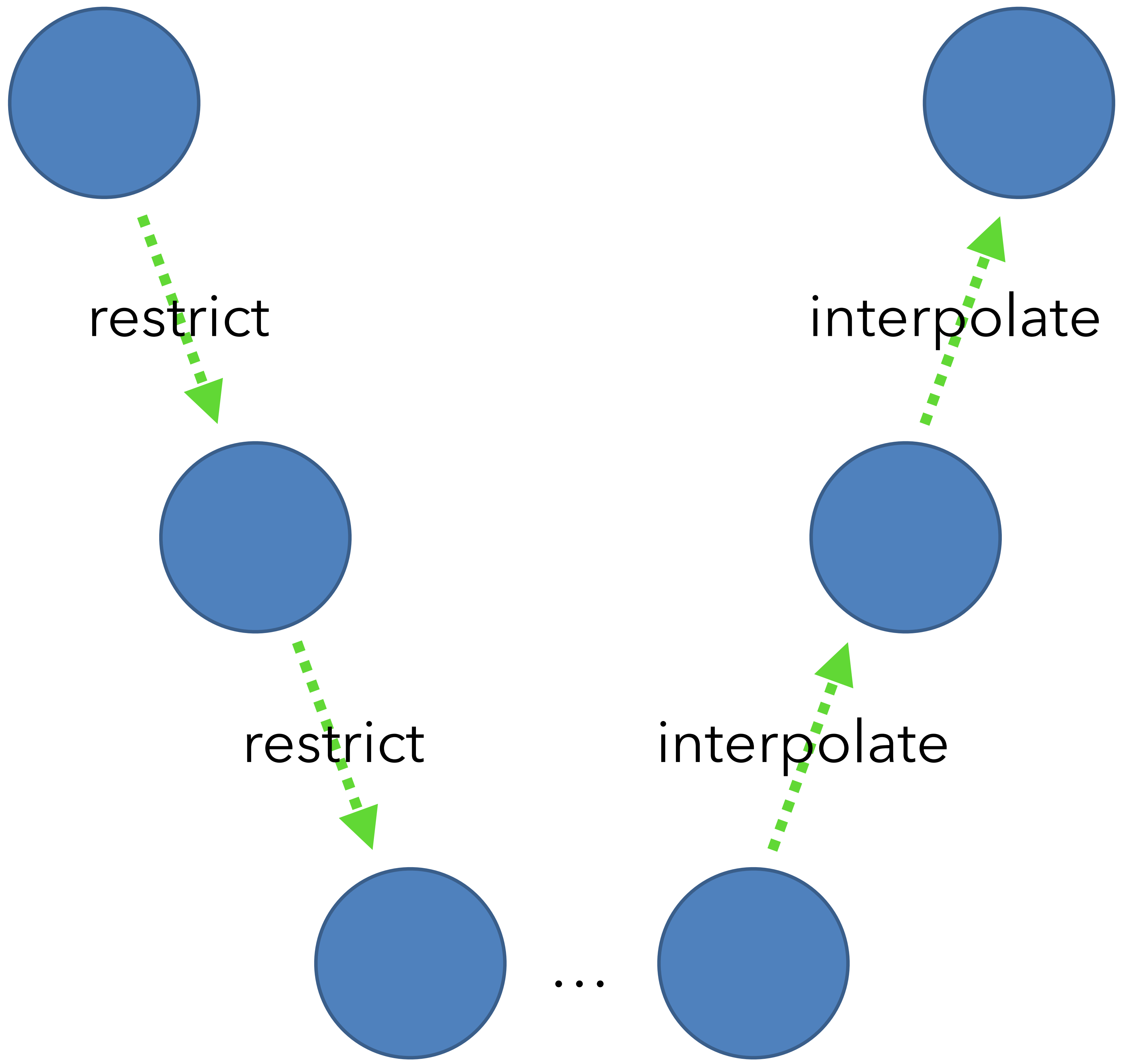
Two-Grid Method

- TwoGrid(f, ρ)
  - relax (f, ρ)
  - $r \leftarrow$ residual of f
  - $r^H \leftarrow$ restriction of r
  - $e^H \leftarrow 0$
  - solve $\boldsymbol{A}e^H = r^H$
  - $e \leftarrow$ interpolation of e^H
  - $f \leftarrow f + e$
  - relax (f, ρ)



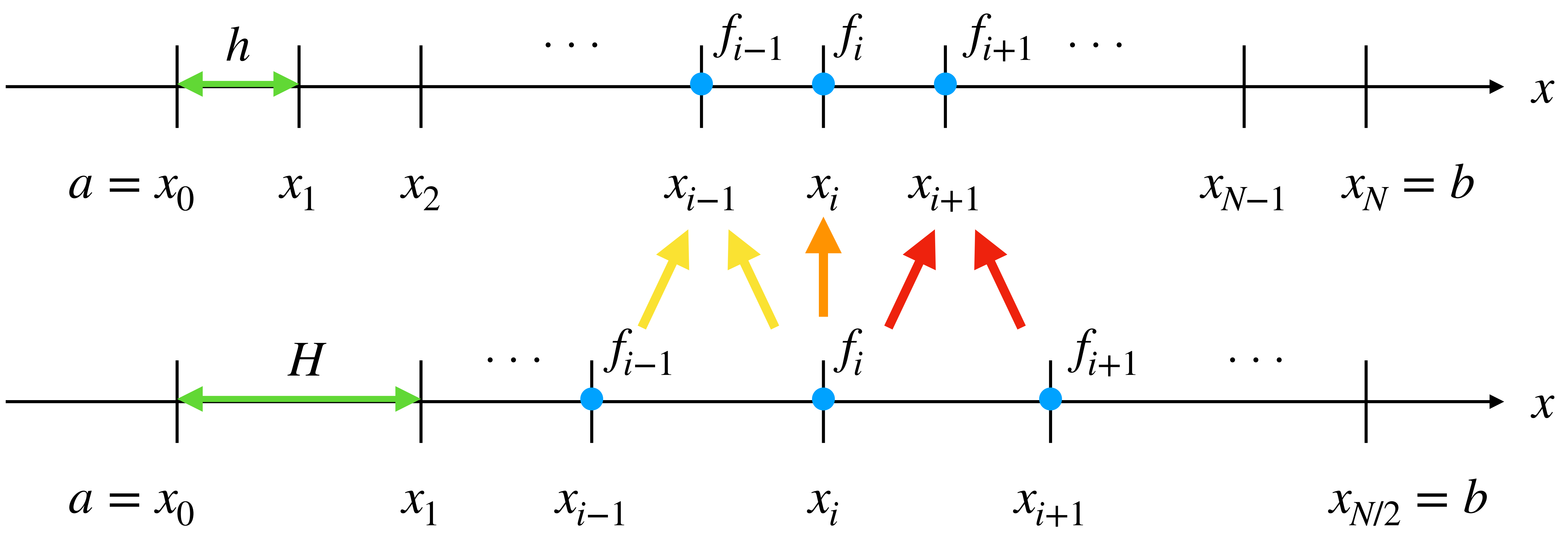
Multigrid Method

- Recursive two-grid method
- In the most coarse grid, only once iteration solves exactly.
- MultiGrid$\left(f, \rho, n\right)$
  - if $n = 1$ then relax $\left(f, \rho\right)$ and return
  - relax $\left(f, \rho\right)$
  - $r \leftarrow$ residual of f
  - $r^H \leftarrow$ restriction of r
  - $e^H \leftarrow 0$
  - MultiGrid$\left(e^H, r^H, n-1\right)$
  - $e \leftarrow$ interpolation of e^H
  - $f \leftarrow f + e$
  - relax $\left(f, \rho\right)$



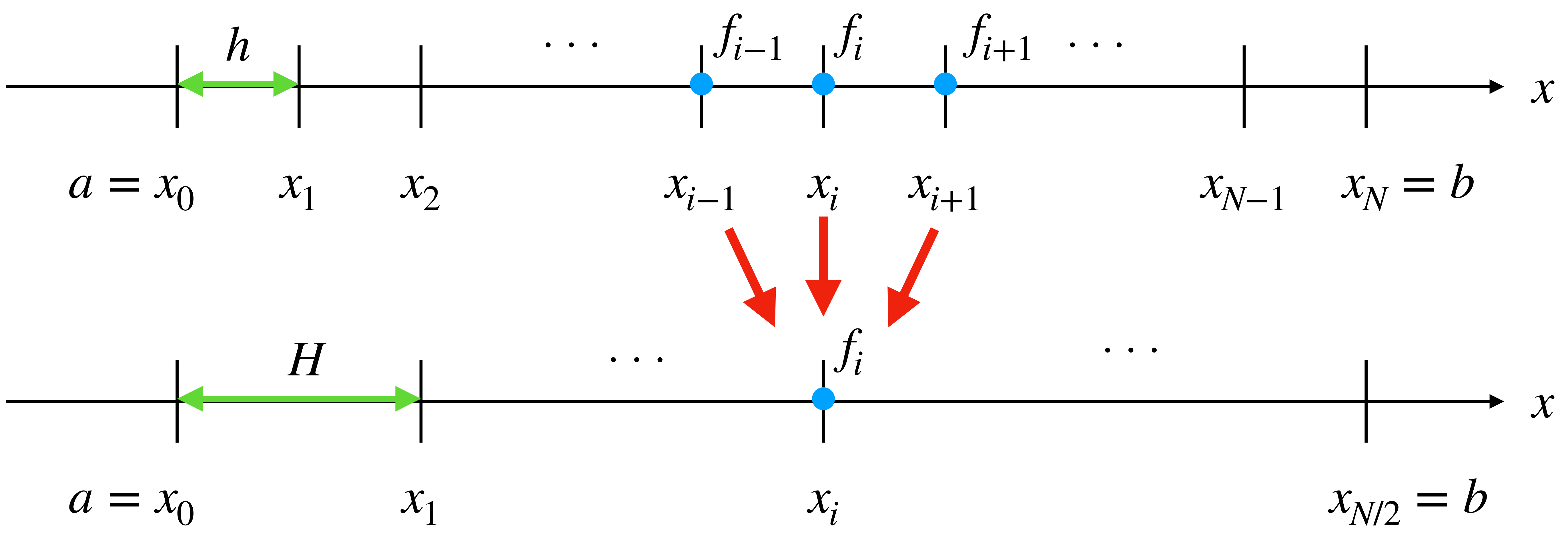
Linear Interpolation

- Interpolation$\left(f^h, f^H\right)$
  - For $i \in \left[1, N\right)$
    - If i is even then
      - $f_i^h \leftarrow f_{i/2}^H$
    - else
      - $f_i^h \leftarrow \frac{1}{2}\left(f_{\lfloor i/2 \rfloor}^H + f_{\lfloor i/2 \rfloor + 1}^H\right)$



Full Weighting Restriction

- Restriction $\left(f^H, f^h\right)$
  - For $i \in \left[1, N\right)$
    - $f_i^h \leftarrow \frac{1}{4} f_{2i-1}^H + \frac{1}{2} f_{2i}^H + \frac{1}{4} f_{2i+1}^H$
- It is adjoint to the interpolation in the sense
  - $\left\langle a^H | R_h^H b^h \right\rangle = \left\langle b^h | I_H^h a^H \right\rangle$



Problem I

Problem I

- 1-1. Iterative Relaxation: 20 pt
- 1-2. Interpolation: 10 pt
- 1-3. Restriction: 10 pt
- 1-4. Multigrid Method: 30 pt
- 1-5. Black Hole: 30 pt

Let's start the competition!!