Lecture 4

Multigrid

New Directions in Mathematics

Adérito Araújo (alma@mat.uc.pt)
April 12, 2024

Defect Correction

- Let $Ax = b$, $A \in \mathbb{R}^{n\times n}$ nonsingular, and $x^{(0)} \simeq x$ with residual

$$r^{(0)} = b - Ax^{(0)} \neq 0$$

- Goal: try to improve the accuracy by writing

$$x^{(1)} = x^{(0)} + \delta^{(0)}$$

- If the defect correction $\delta^{(0)}$ is computed by

$$A\delta^{(0)} = r^{(0)},$$

  then $x^{(1)} = x$
- Defect correction princilpe: let $A_{approx}^{-1} \simeq A^{-1}$ and consider

$$x^{(1)} = x^{(0)} + \delta^{(0)}, \quad \text{with } \delta^{(0)} = A_{approx}^{-1} r^{(0)}$$

Defect Correction Iteration

- Defect correction iteration:

$$x^{(k+1)} = x^{(k)} + \delta^{(k)} = x^{(k)} + A_{approx}^{-1} r^{(k)}, \quad k = 0, 1, \ldots$$

- The method converges if and only if

$$\rho(I - A_{approx}^{-1} A) < 1$$

- If $A_{approx}^{-1} A \simeq I$ then $\rho(I - A_{approx}^{-1} A) \simeq 0$ (fast convergence)
- Defect correction iteration $\Leftrightarrow$ Richardson iteration
  - Richardson iteration: $R = I - P^{-1} A$ and

    $$x^{(k+1)} = Rx^{(k)} + P^{-1} b \;\Leftrightarrow\; x^{(k+1)} = x^{(k)} + P^{-1} r^{(k)}, \quad k \geqslant 0$$

  - Defect correction iteration: $P^{-1} = A_{approx}^{-1}$, $R = I - A_{approx}^{-1} A$

Two-Grid Method: Coarse and Fine Grids

- Idea: use the defect correction principle with

$$A_{approx}^{-1} = A_{coarse}^{-1} \simeq A_{fine}^{-1} = A^{-1}$$

- What we need: transfer vectors corresponding to the fine grid to vectors corresponding to the coarse grid and vice-versa
- Applications
  - Improve the accuracy of an approximate solution obtained for example by Gaussian elimination
  - Develop fast iterative solutions of linear system arising in the discretization of differential and integral equations

Example: Elliptic Equation (1D)

- Boundary value problem:

$$-u_{xx} = f, \quad 0 < x < 1, \quad u(0) = u(1) = 0$$

- Discretizing the problem (FDM)

$$A_h u_h = f_h,$$

with

$$A_h = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \cdot & \cdot & \cdot & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}, \quad h = \frac{1}{n+1},$$

and $f_{h,j} = f(x_j)$, $u_{h,j} \simeq u(x_j)$, $j = 1, \ldots, n$, on the grid

$$0 = x_0 < x_1 < \ldots < x_n < x_{n+1} = 1, \quad x_j = jh$$

Example: Elliptic Equation (1D)

- Solve the linear system by the JOR method (see: Lecture 1)

$$u_h^{(k+1)} = u_h^{(k)} + \omega D_h^{-1}(\underbrace{f_h - A_h u_h^{(k)}}_{\text{residual}}), \quad k = 0, 1, \ldots.$$

- Exercise 4.1: Prove that the eigenvalues of $I - \omega D_h^{-1} A_h$ are

$$\lambda_{h,j} = 1 - 2\omega \sin^2 \frac{\pi j h}{2}, \quad j = 1, \ldots, n,$$

and the corresponding eigenvectures are

$$v_{h,j} = [\sin(\pi j h), \ldots, \sin(\pi n j h)]^T, \quad j = 1, \ldots, n,$$

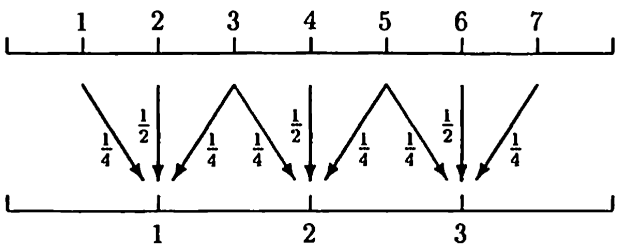
that form an orthogonal basis of $\mathbb{R}^n$.

Choice of the Damping Factor

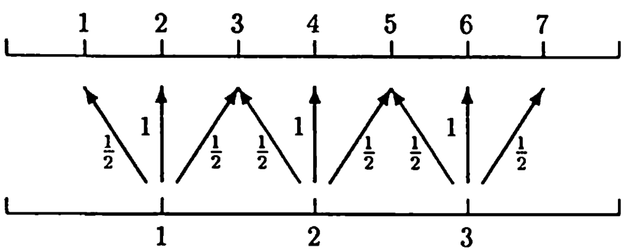
- From Theorem 1.7
$$\omega_{opt} = \frac{2}{2 - \lambda_{h,max} - \lambda_{h,min}} \Rightarrow \omega_{opt} \simeq 1$$
- Use Damped/Underrelaxed Jacobi Iteration (DJI) (i.e, JOR with $0 < \omega < 1$)
- DJI with $w = 0.5$ is a smoothing iteration
  - Since v_j is an orthogonal basis of $\mathbb{R}^n$, the error satisfies
  $$u_h - u_h^{(k)} = \sum_{j=1}^{n} \alpha_j^{(k)} v_{h,j},$$
  and it may proven that $\alpha_j^{(k+1)} = \lambda_{h,j}\alpha_j^{(k)}$
  - Exercise 4.2: Prove that, for $\omega = 0.5$, we have
  $$\left|\alpha_j^{(k+1)}\right| = \cos^2\frac{\pi jh}{2}\left|\alpha_j^{(k)}\right| \leqslant \frac{1}{2}\left|\alpha_j^{(k)}\right|, \quad j = \underbrace{(n+1)/2, \ldots, n}_{\text{high frequencies}}.$$
  - Fast convergence for high frequencies

Coarse Grid Correction

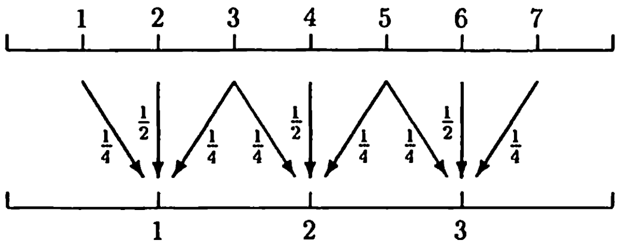
- We consider a fine grid with mesh size h and a coarse grid with mesh size $2h$ (n is odd)
- Restriction



- Prolongation



Restriction: Fine to Coarse Grid



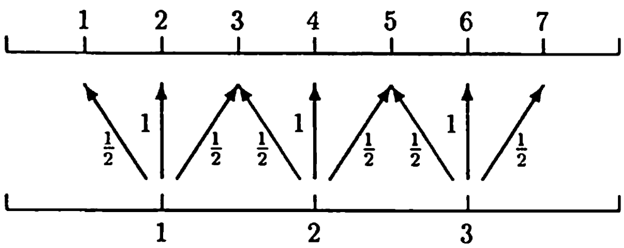
- Restriction: $R_h : \mathbb{R}^n \to \mathbb{R}^{\frac{n-1}{2}}$ such that

$$(R_h u_h)_j = \frac{1}{4}(u_{h,2j-1} + 2u_{h,2j} + u_{h,2j+1}), \quad j = 1, \dots, (n-1)/2$$

- Restriction Matrix: $u_{2h} = R_h u_h$ with

$$R_h = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & & & \\ & & 1 & 2 & 1 & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & 1 & 2 & 1 & \\ & & & & & 1 & 2 & 1 \end{bmatrix}$$

Prolongation: Coarse to Fine Grid



- Prolongation: $P_{2h} : \mathbb{R}^{\frac{n-1}{2}} \to \mathbb{R}^n$ (piecewise linear interpolation)

$$(P_{2h} u_{2h})_{2j} = u_{2h,j}, \qquad j = 1, \dots, (n-1)/2,$$

$$(P_{2h} u_{2h})_{2j-1} = \frac{1}{2}(u_{2h,j} + u_{2h,j-1}), \qquad j = 1, \dots, (n+1)/2$$

- Prolongation Matrix: $u_h = P_{2h} u_{2h}$ with

$$P_{2h} = 2R_h^T = \frac{1}{2}\begin{bmatrix} 1 & & \\ 2 & & \\ 1 & 1 & \\ & 2 & \\ & 1 & 1 \\ & & \vdots \end{bmatrix}$$

Defect Correction Iteration

- Let
$$A_{coarse}^{-1} = P_{2h} A_{2h}^{-1} R_h$$
and compute
$$u_h^{(k+1)} = u_h^{(k)} + A_{coarse}^{-1}(f_h - A_h u_h^{(k)}), \quad k = 0, 1, \ldots$$
- How to compute $\delta_h^{(k)} = A_{coarse}^{-1} r_h^{(k)} = P_{2h} A_{2h}^{-1} R_h r_h^{(k)}$
  1. Restrict $r_h^{(k)} \in \mathbb{R}^n$ to $r_{2h}^{(k)} = R_h r_h^{(k)} \in \mathbb{R}^{\frac{n-1}{2}}$
  2. Solve the $\frac{n-1}{2} \times \frac{n-1}{2}$ system $A_{2h}\delta_{2h}^{(k)} = r_{2h}^{(k)}$
  3. Prolong the solution $\delta_{2h}^{(k)} \in \mathbb{R}^{\frac{m-1}{2}}$ to $\delta_h^{(k)} = P_{2h}\delta_{2h}^{(k)} \in \mathbb{R}^n$

Two-Grid Iteration Scheme

- Two-Grid Iteration Scheme: Combining this coarse grid correction with N steps of the DJI we obtain
$$u_h^{(k+1)} = u_h^{(k,N)} + P_{2h} A_{2h}^{-1} R_h (f_h - A_h u_h^{(k,N)}),$$
for $k = 0, 1, \ldots,$ where $u_h^{(k,N)} = \mathcal{J}^N(A_h, u^{(k)}, f_h)$ denotes the result of N steps of DJI starting with $u^{(k)}$.
- Exercise 4.3: Prove that the iteration matrix corresponding to this two-grid method is given by
$$T_N = \underbrace{(I - P_{2h} A_{2h}^{-1} R_h A_h)}_{\text{coarse-grid correction}} \left(I - \frac{1}{2} D_h^{-1} A_h\right)^N.$$
- $N = 1$ ($T_1 = T$): alternate one step of the DJI (fine grid) with a coarse-grid correction by elimination (coarse grid)

HW Exercises

- Exercise 4.4: Prove that, for $j = 1, \dots, (n-1)/2$,

$$R_h v_{h,j} = c_j^2 v_{2h,j}, \quad R_h v_{h,n+1-j} = -s_j^2 v_{2h,j},$$

and

$$P_{2h} v_{2h,j} = c_j^2 v_{h,j} - s_j^2 v_{h,n+1-j},$$

with

$$c_j = \cos\frac{j\pi h}{2}, \quad s_j = \sin\frac{j\pi h}{2}.$$

- Exercise 4.5: Prove that, for $j = 1, \dots, (n-1)/2$,

$$\begin{bmatrix} T v_{h,j} \\ T v_{h,n+1-j} \end{bmatrix} = s_j^2 c_j^2 \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}}_{Q} \begin{bmatrix} v_{h,j} \\ v_{h,n+1-j} \end{bmatrix},$$

and, since $R_h v_{h,(n+1)/2} = 0$,

$$T v_{h,(n+1)/2} = \frac{1}{2} v_{h,(n+1)/2}.$$

Convergence od Two-Grid Method

Theorem 4.1

For the spectral radius of T we have that $\rho(T) = 0.5$. Consequently, the two-grid iterations converge.

- Proof: From Exercise 4.1 we have, for $j = 1, \dots, (n-1)/2$

$$A_{2h} v_{2h,j} = \frac{1}{h^2}\sin^2(\pi j h) v_{2h,j} = \frac{4}{h^2} c_j^2 s_j^2 v_{2h,j},$$

and then

$$A_{2h}^{-1} v_{2h,j} = \frac{h^2}{4 c_j^2 s_j^2} v_{2h,j}.$$

From Exercise 4.5, since the matrix Q has eigenvalues 0 and 2, it can be proved that the matrix T has the eigenvalues

$$2 c_j^2 s_j^2 = \frac{1}{2}\sin^2 \pi j h, \quad j = 1, \dots, (n+1)/2,$$

and the eigenvalue zero of multiplicity $(n-1)/2$. □

Two-Grid Method: General Formulation

- At the highest level (finest grid) a mesh-size of h is used

$$A_h u_h = f_h$$

- Requirement: a system similar to the one above must be solved at the coarser levels
- A coarse grid with mesh size H is used (coarser mesh Ω_H)
- Galerkin projection:

$$A_H = R_h A_h P_H, \quad f_H = R_h f_h$$

Smoothing Step

- Define a smoother $\mathcal{S}$ (e.g., DJI, SOR, ...) and consider

$$u_h^{(N)} = \mathcal{S}^N(A_h, u_h^{(0)}, f_h)$$

- Smoothing iterations

$$u_h^{(j+1)} = S_h u_h^{(j)} + g_h, \quad j = 0, \ldots, N-1,$$

  where S_h is the smoothing iteration matrix
- Exercise 4.6: Prove that

$$u_h^{(j+1)} = u_h^{(j)} + B_h(f_h - A_h u_h^{(j)}), \quad j = 0, \ldots, N-1,$$

  with $B_h = (I - S_h)A_h^{-1}$, $S_h = I - B_h A_h$, $g_h = B_h f_h$.
- The error $e_h^{(N)}$ and residual $r_h^{(N)}$ after N smoothing steps are

$$e_h^{(N)} = S_h^N e_h^{(0)} = (I - B_h A_h)^N e_h^{(0)}, \quad r_h^{(N)} = (I - A_h B_h)^N r_h^{(0)}$$

Two-Grid Cycle

- Algorithm: Two-Grid Cycle

$u_h = \mathcal{S}^N(A_h, u_h^{(0)}, f_h)$ % Pre-smooth

$r_h = f_h - A_h u_h$ % Get residual

$r_H = R_h r_h$ % Coarsen

Solve $A_H \delta_H = r_H$

$u_h = u_h + P_H \delta_H$ % Correct

$u_h = \mathcal{S}^M(A_h, u_h, f_h)$ % Post-smooth

Two-Grid Cycle

- One iteration of the 2-grid algoritms corresponds to

$$u_h^{(1)} = T_h u_h^{(0)} + g_{M_h}$$

- If $f_h \equiv 0$

$$u_h^{(1)} = S_h^M(S_h^N u_h^{(0)} + P_H A_H^{-1} R_h(-A_h S_h^N u_h^{(0)}))$$

- The 2-grid iteration operator is

$$T_h = S_h^M \underbrace{(I - P_h A_H^{-1} R_h A_h)}_{\text{coarse-grid correction}} S_h^N$$

Multigrid V-cycle

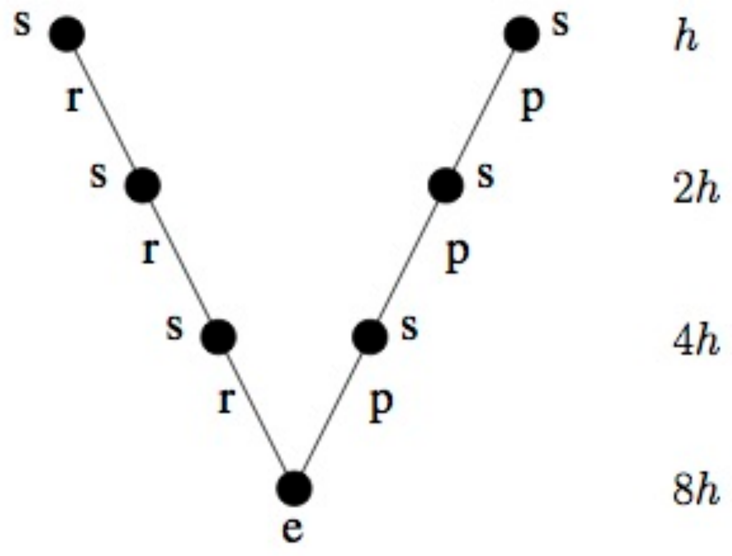


Figure: s - smoothing, r - restriction, p - prolongation, e - exact solver

V-cycle Multigrid

- Algorithm: V-cycle Multigrid

function $u_h = V_{cycle}(A_h, u_h^{(0)}, f_h, N, M)$

$u_h = \mathcal{S}^N(A_h, u_h^{(0)}, f_h)$ % Pre-smooth

$r_h = f_h - A_h u_h$ % Get residual

$r_H = R_h r_h$ % Coarsen

If $(H == h_0)$

Solve $A_H \delta_H = r_H$

else

$\delta_H = V_{cycle}(A_H, 0, r_H, N, M)$ % Recursion

end if

$u_h = u_h + P_H \delta_H$ % Correct

$u_h = \mathcal{S}^M(A_h, u_h, f_h)$ % Post-smooth

end function

- Note: H stands for $2h$ and h_0 for the coarsest mesh-size

W-cycle Multigrid

- More corrections, say γ times on a coarse grid before returning to a finer level
- Algorithm: W-cycle Multigrid

function $u_h = MG(A_h, u_h^{(0)}, f_h, N, M, \gamma)$

$u_h = \mathcal{S}^N(A_h, u_h^{(0)}, f_h)$ % Pre-smooth

$r_h = f_h - A_h u_h$ % Get residual

$r_H = R_h r_h$ % Coarsen

If $(H == h_0)$

Solve $A_H \delta_H = r_H$

else

$\delta_H = MG^{\gamma}(A_h, 0, f_h, N, M, \gamma)$ % Recursion

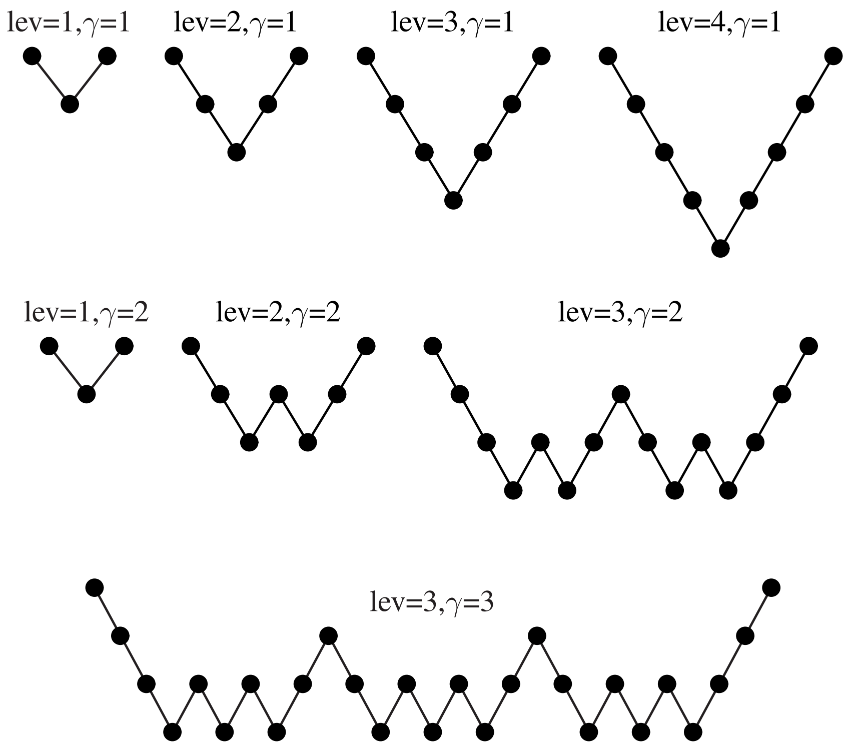
end if

$u_h = u_h + P_H \delta_H$ % Correct

$u_h = \mathcal{S}^M(A_h, u_h, f_h)$ % Post-smooth

end function

V-cycles and W-cycle



Full Multigrid

- Slightly different approach: find an approximation to the solution with only one sweep through the levels, going from bottom to top
- The system is first solved (or smoothed) on a very coarse grid, then one goes to the next finer grid and smoothes the system on this grid and so on, until the finest grid is reached
- Algorithm: Full Multigrid

  Set $h = h_0$ and solve $A_h u_h = f_h$

  for $k = 1$ to p do

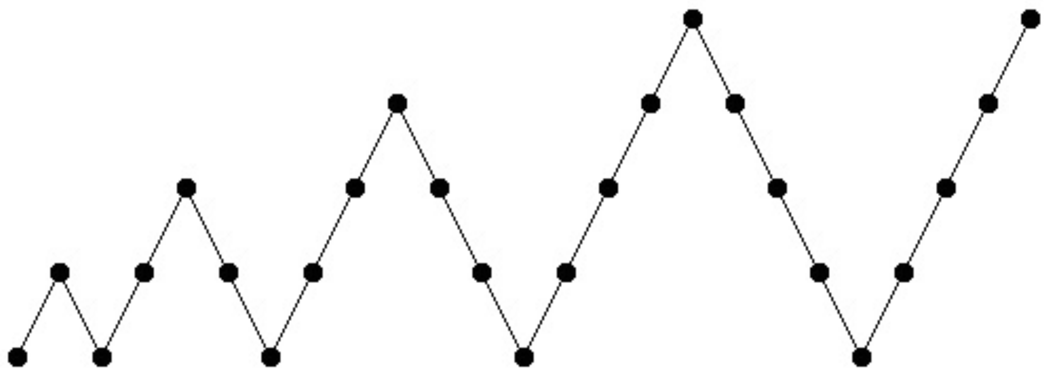
  $u_{h/2} = \hat{P}_h u_h$

  $h = h/2$

  $u_h = MG^{\mu}(A_h, u_h, f_h, N, M, \gamma)$

  end for
- Note: the interpolation operator $\hat{P}_h$ is typically of a lower order than P_h

Full Multigrid V-cycle



Basic Convergence Results

- Cost (storage and computing time)
  - overall costs are dominated by the costs of the finest grid
- Speed of convergence
  - significant acceleration compared with relaxation methods
  - the convergence rate is independent of the number of unknowns
  - constant number of multigrid steps to obtain a given number of digits
  - overall computational work increases only linearly with the number of unknowns