

INTRODUCTION TO MULTIGRID METHODS

LONG CHEN

1. ALGEBRAIC EQUATION OF TWO POINT BOUNDARY VALUE PROBLEM

We consider the discretization of Poisson equation in one dimension:

$$(1) \quad -u'' = f, \quad x \in (0, 1) \quad u(0) = u(1) = 0.$$

For any integer N , we consider a uniform grid, denoted by \mathcal{T}_h , of the interval $[0, 1]$ as follows:

$$0 = x_0 < x_1 < \dots < x_N < x_{N+1} = 1, x_j = jh, j = 0 : N + 1,$$

where $h = 1/(N + 1)$ is the length of each subinterval. Let ϕ_i be the hat basis function at x_i . For a linear finite element function $v = \sum_{i=1}^N v_i \phi_i$, we denote by $\mathbf{v} = (v_1, \dots, v_N)^t$ the vector formed by the coefficients.

The algebraic system obtained from the linear finite element discretization is

$$(2) \quad \mathbf{A}\mathbf{u} = \mathbf{b},$$

where

$$\mathbf{A} = \text{diag}(-1, 2, -1), b_i = h^2 f(x_i).$$

Due to the special structure of the matrix \mathbf{A} , we can write out eigenvalues and eigenvectors of \mathbf{A} explicitly.

Proposition 1.1. *If $\mathbf{A} = \text{diag}(b, a, b)$ be a $N \times N$ tri-diagonal matrix, then the eigenvalue of \mathbf{A} is*

$$\lambda_k = a + 2b \cos \theta_k, \quad k = 1, \dots, N$$

and its corresponding eigenvector is:

$$\boldsymbol{\xi}_k = I_h \sin(k\pi x) = (\sin \theta_k, \sin 2\theta_k, \dots, \sin N\theta_k),$$

where I_h is the nodal interpolation and

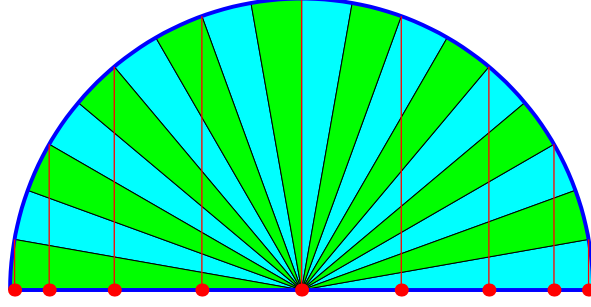
$$\theta_k = k\theta = k\pi h = \frac{k\pi}{N + 1}$$

Proof. It can be easily verified by the direct calculation. □

It is interesting to note that eigenvectors are independent of a and b . Applying to the special case that $a = 2$ and $b = -1$, we have

$$(3) \quad \lambda_k = 2(1 - \cos \theta_k) = 4 \sin^2 \frac{\theta_k}{2}.$$

Notice that $\lambda_1 = \mathcal{O}(h^2)$ and $\lambda_N = \mathcal{O}(1)$ and therefore $\kappa(\mathbf{A}) = \mathcal{O}(h^{-2})$, i.e., the matrix \mathbf{A} is ill conditioned. For finite difference method, the corresponding matrix is \mathbf{A}/h^2 and for finite element method \mathbf{A}/h . The scaling will introduce a scaling of all eigenvalues but not the condition number.

FIGURE 1. $\cos \theta_k$

Exercise 1.2. Apply Proposition 1.1 to the mass matrix $M = (m_{ij})$ with $m_{ij} = \int_0^1 \phi_i \phi_j dx$ and conclude it is well conditioned.

Exercise 1.3. Figure out the eigenvalues and eigenvectors of the stiffness matrix for the Neumann problem.

2. SMOOTHING PROPERTY OF RICHARDSON ITERATION

The integer k in the function $\sin(k\pi x)$ is known as frequency. For a uniform grid of $[0, 1]$ with length $h = 1/(N + 1)$, the range of k is $1, \dots, N$. Note that for $k \geq N + 1$, $I_h(\sin k\pi x) = I_h(\sin \text{ mod } (k, N + 1)\pi x)$. That is the grid \mathcal{T}_h cannot see the frequency higher than $1/h$. So the coarser grid \mathcal{T}_{2h} can only see the frequency less than $1/(2h)$. The frequency, and the corresponding angles, which can be only captured by the fine grid \mathcal{T}_h but not \mathcal{T}_{2h} is in the range of

$$\frac{1}{2}(N + 1) \leq k \leq N, \quad \frac{\pi}{2} \leq \theta_k \leq \frac{N}{N + 1}\pi < \pi,$$

and will be called high frequency (relative to \mathcal{T}_h). We shall show the classic iteration methods will smooth out the high frequency part of the error very quickly while left with low frequency part.

We shall apply the simplest iterative method, Richardson iteration, to solve (2). Recall that one iteration of Richardson method is

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \omega(\mathbf{b} - \mathbf{A}\mathbf{u}^k).$$

The error equation

$$\mathbf{u} - \mathbf{u}^{k+1} = (I - \omega\mathbf{A})(\mathbf{u} - \mathbf{u}^k)$$

and therefore

$$\|\mathbf{u} - \mathbf{u}^{k+1}\|_{\mathbf{A}} \leq \rho(I - \omega\mathbf{A})\|\mathbf{u} - \mathbf{u}^k\|_{\mathbf{A}}.$$

When $\omega \in (0, 2/\lambda_{\max}(\mathbf{A}))$, the method is convergent and when $\omega = 2/(\lambda_{\min}(\mathbf{A}) + \lambda_{\max}(\mathbf{A}))$ it achieves the optimal rate

$$\rho = \frac{1 - \kappa(\mathbf{A})}{1 + \kappa(\mathbf{A})} \leq 1 - Ch^2.$$

It means that the norm of the error, as a summation of squares of all components, will decay with a slow rate $1 - Ch^2$.

We shall do a fine analysis for the decay rate of different components of the error. First we need to change the coordinate. The set of all eigenvectors of \mathbf{A} will form an orthogonal basis of \mathbb{R}^N . We expand $e^0 = \mathbf{u} - \mathbf{u}^0$ in this coordinate.

$$e^0 = \sum_{k=1}^N \alpha_k \boldsymbol{\xi}^k.$$

We then get

$$e^m = \mathbf{u} - \mathbf{u}^m = (I - \omega \mathbf{A})^m e^0 = \sum_{k=1}^N \alpha_{m,k} \boldsymbol{\xi}^k,$$

where

$$\alpha_{m,k} = (1 - \omega \lambda_k)^m \alpha_k.$$

Since eigenvectors are also A -orthogonal, we have

$$\|e^m\|_A = \left(\sum_{k=1}^N \alpha_{m,k}^2 \|\boldsymbol{\xi}_k\|^2 \right)^{1/2}, \quad \|e^0\|_A = \left(\sum_{k=1}^N \alpha_k^2 \|\boldsymbol{\xi}_k\|^2 \right)^{1/2}.$$

The k th component coefficient decays with different rates

$$|\alpha_{m,k}| \leq (1 - \omega \lambda_k)^m |\alpha_k|.$$

We choose $\omega = 1/4$ to simplify the rate as

$$\rho_k = |1 - \omega \lambda_k| = \frac{1}{2}(1 + \cos \theta_k).$$

From the graph of ρ_k , it is easy to see that

$$\rho_1 \approx 1 - Ch^2,$$

but

$$\rho_N \leq Ch^2, \text{ and } \rho_{(N+1)/2} = 1/2.$$

This means that high frequency components get damped very quickly while the low frequency component very slowly. Gauss-Seidel method, as a better iterative method, has the same affect. Indeed it is a better smoother than the Richardson method, although it is little bit more complicated to analyze theoretically. Interesting enough, Jacobi method, corresponding to $\omega = 1/2$, does not have a smoothing property. We use the same example to compute

$$\rho_1 = \rho_N \approx 1 - Ch^2,$$

which, if $\alpha_N \neq 0$, can not practically get very small for large N .

Exercise 2.1. The damped Jacobi ωD^{-1} with $0 < \omega < 1$ will have smoothing effect. Try to find the optimal ω in the sense that

$$\min_{\omega} \max_{(N+1)/2 \leq k \leq N} \rho_k(I - \omega D^{-1}A),$$

where $\rho_k(I - \omega D^{-1}A)$ means the contraction factor of the k -th eigenmode.

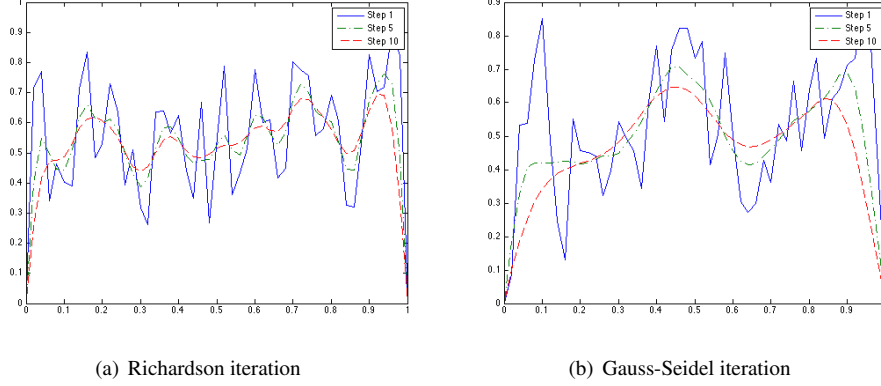


FIGURE 2. Smoothing effect of classic iteration methods.

3. MULTIGRID METHODS

The multigrid methods is based on the two observation

- High frequency will be damped by smoother.
- Low frequency can be approximated well by coarse grid.

Note that the low-frequency errors on a fine mesh becomes high-frequency errors on a coarser mesh. For the coarse grid problem, we can apply the smoothing and the separation of scales again. Recursively apply the smoothing to each level results in the classical formulation of multigrid. **More on the classical recursive definition of Multi-Grid methods.**

4. STABLE DECOMPOSITION

The local Fourier mode analysis can only be applied to uniform grids and constant coefficients case. For finite element methods of elliptic equations with variable coefficients based on general triangulations, the eigenvalue and eigenvectors of the stiffness matrix is not easy to find out. Indeed it is even harder than solving the linear algebraic equation. We shall look at the smoother and multigrid method from the space decomposition and subspace correction.

Given a triangular mesh \mathcal{T}_h , let us consider the trivial decomposition of the linear finite element space \mathbb{V}_h based on \mathcal{T}_h , i.e., $\mathbb{V}_h = \sum_{i=1}^N \mathbb{V}_{h,i}$ with $\mathbb{V}_{h,i} = \text{sp}(\phi_i)$. Recall that the classic iterative methods can be interpret as PSC or SSC methods based on this decomposition. The convergent rate depends on the stability and the quasi-orthogonality of the space decomposition. The quasi-orthogonality can be easily derived from the locality of the finite element basis. Thus below we study the stability in details.

For any $v \in \mathbb{V}_h$, write $v = \sum_{i=1}^N v_i$. It is stable in L^2 norm (see Exercise 4.1) but not in H^1 -norm. That is the inequality

$$(4) \quad \sum_{i=1}^N |v_i|_1^2 \leq C |v|_1^2$$

does not hold for a universal constant C . Indeed using inverse inequality and Poincare inequality, one can prove (4) with constant CN . The constant CN is sharp due to the existence of low frequency. As an extreme example, one can choose $v = \sum_{i=1}^N \phi_i$. Then

v is flat except in a band near the boundary. The derivative of v is thus zero in most region while $|\nabla v_i|$ is always of order $1/h$.

Exercise 4.1. For a finite element function v , prove the basis decomposition is stable in L^2 norm

$$\|v\|^2 \approx \sum_{i=1}^N \|v_i\|^2.$$

Hint: Prove it in one triangle first. The L^2 norm of a linear function can be evaluated exactly using mid points rule.

Fortunately (4) holds for high frequency. A function $v \in \mathbb{V}_h$ is of high frequency if $\|h^{-1}v\| \lesssim |v|_1$, i.e., the function oscillates with scale h . Note that the inverse inequality $|v|_1 \lesssim \|h^{-1}v\|$ holds for both high and low frequency. So by inverse inequality, the stability of decomposition in L^2 norm, and the definition of high frequency, we have

$$\sum_{i=1}^N |v_i|_1^2 \lesssim h^{-2} \sum_{i=1}^N \|v_i\|^2 \lesssim h^{-2} \|v\|^2 \lesssim |v|_1^2.$$

Therefore Gauss-Seidel, Richardson or weighted Jacobi is still a good smoother for damping the high frequency. The low frequency will be taken care of by coarse grid correction.

Let \mathcal{T}_H be a coarse grid with mesh size $H = 2h$ and \mathbb{V}_h the corresponding finite element space. Let us present the two-level methods based on the decomposition

$$(5) \quad \mathbb{V}_h = \sum_{i=1}^N \mathbb{V}_{h,i} + \mathbb{V}_H,$$

and apply SSC with exact solver for each subspace problem. This is equivalent to applying a Gauss-Seidel iteration on the fine grid and then solving the coarse grid problem exactly.

Theorem 4.2. *The two level SSC based on the decomposition (5) is uniformly convergent.*

Proof. By the abstract convergent theory, it suffices to find a stable decomposition. For any $v \in \mathbb{V}_h$, we let $v_H = P_H v$ and $v_h = v - v_H$, and decompose $v_h = \sum_{i=1}^N v_{h,i}$. Recall that P_H is the projection in the $(\cdot, \cdot)_A$ inner product. So

$$|v|_1^2 = |v_H|_1^2 + |v_h|_1^2.$$

Now we prove v_h is of high frequency. To this end, we use the L^2 error estimate of the H^1 -projection (which requires the H^2 regularity assumption; see the FEM chapter)

$$\|v_h\| = \|v - P_H v\| \lesssim h|v - P_H v|_1 = h|v_h|_1.$$

Therefore

$$|v_H|_1^2 + \sum_{i=1}^N |v_{h,i}|_1^2 \lesssim |v_H|_1^2 + |v_h|_1^2 = |v|_1^2.$$

□

The result and proof above can be easily generalized to multilevel decomposition. We refer to the Chapter: Convergence Theory of Multigrid Method for details. In the following sections, we will focus on the implementation.

5. TWO LEVEL METHODS AND TRANSFER OPERATORS

We shall use two level methods to illustrate how to realize operators by matrices. The space decomposition we choose is

$$\mathbb{V} = \mathbb{V}_1 + \mathbb{V}_2 \quad \text{with} \quad \mathbb{V}_1 \subset \mathbb{V}_2 = \mathbb{V}.$$

We call \mathbb{V} fine space and \mathbb{V}_1 coarse space since it is usually based on a coarse mesh.

Recall that the PSC for this two level decomposition in operator form is

- (1) $r = f - Au^k$;
- (2) $e = I_1 R_1 I_1^t r + I_2 R_2 I_2^t r$;
- (3) $u^{k+1} = u^k + e$.

The matrix form of step 1 and 3 is trivial. We only discuss the realization of step 2. Namely given a residual r , return a correction $e = Br$.

Since $\mathbb{V}_2 = \mathbb{V}$, $I_2 = I_2^t = I$. The solver R_2 can be a simple Richardson ωI with $\omega = 1/\lambda_{\max}(A) = h^2$. However in the matrix form, $\mathbf{R}_2 = 1/\lambda_{\max}(\mathbf{A}) = h^{d-2}$. There is a scaling between operator form and matrix form due the identification of \mathbb{R}^N and $(\mathbb{R}^N)'$ by the mass matrix. More accurate smoothers can be choose as weighted Jacobi method $\mathbf{R}_2 = \omega \mathbf{D}^{-1}$ or Gauss-Seidel method $\mathbf{R}_2 = (\mathbf{D} + \mathbf{L})^{-1}$.

The transformation to the coarse \mathbb{V}_1 is not easy. There are three operators to realize: I_1 , R_1 , and I_1^t .

Prolongation operator. Let us first discuss the operator $I_1 : \mathbb{V}_1 \rightarrow \mathbb{V}_2$. By the definition, it is the natural inclusion $\mathbb{V}_1 \hookrightarrow \mathbb{V}$ i.e. treat a function $u_1 \in \mathbb{V}_1$ as a function in $u_1 \in \mathbb{V}$ since $\mathbb{V}_1 \in \mathbb{V}_2$. So the operator is the identity and we change the notation to $I_1^2 : \mathbb{V}_1 \rightarrow \mathbb{V}_2$. But the matrix representation is different since we have different basis in \mathbb{V}_1 and \mathbb{V}_2 ! We use a 1-D two level grids in Figure xx to illustrate the different basis functions in fine and coarse grids. In this example $\mathbf{I}_1^2 : \mathbb{R}^3 \rightarrow \mathbb{R}^5$ will map a vector with small size to one with bigger size and thus called *prolongation* operator. We determine this map by the following two observations:

- (1) \mathbf{u}_1 and $\mathbf{u}_2 = \mathbf{I}_1^2 \mathbf{u}_1$ represent the same function in \mathbb{V}_2 ;
- (2) a function in \mathbb{V}_2 is uniquely determined by the values at the nodal points.

For nodes in both fine grids and coarse grids,

$$\mathbf{u}_2(1) = \mathbf{u}_1(1), \quad \mathbf{u}_2(3) = \mathbf{u}_1(2), \quad \mathbf{u}_2(5) = \mathbf{u}_1(3).$$

For the nodes only existing in the fine grids, by (1), values at these nodes can be evaluated in the coarse grids. Since we are using linear elements, we get

$$\mathbf{u}_2(2) = (\mathbf{u}_1(1) + \mathbf{u}_1(2))/2, \quad \mathbf{u}_2(4) = (\mathbf{u}_1(3) + \mathbf{u}_1(5))/2.$$

In matrix form, $\mathbf{I}_1^2 \in \mathbb{R}^{5 \times 3}$ can be written as

$$\begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 1 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \end{bmatrix}$$

To define the prolongation, we need and only need to know the correspondences of the index of nodes between two grids. Different index mapping will give different prolongation matrix. A better hierarchical index is given in Figure xxx (b). In this index, the

prolongation matrix can be written as

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 \end{bmatrix}.$$

The presence of the identity matrix can save the computation $\mathbf{I}_1^2 \mathbf{x}$.

The construction of the interpolation operator can be easily generalized to high dimensions for linear element. The only information is the index map between coarse grid points and fine grid points. We classify the grid points in the fine grid into two groups:

- C: the points in both fine and coarse grids
- F: the points in fine grid only.

For group F, we can use HB (hierarchical basis) matrix with $\text{HB}(:, 2:3)$ being two parent nodes of the node $\text{HB}(:, 1)$. Note that $\text{HB}(:, 1)$ is the index in the fine level while $\text{HB}(:, 2:3)$ are in the coarse level. Then the interpolation at the grids points in F can be realized

$$\text{uf}(\text{HB}(1:\text{end}, 1)) = (\text{uc}(\text{HB}(1:\text{end}, 2)) + \text{uc}(\text{HB}(1:\text{end}, 3))) / 2;$$

For group C, although those grid points are in both coarse and fine grids, their indices could be different. For example, in Fig xxx (a), the 3-rd point in the fine grid is the 2-nd one in the coarse grid. Therefore we need an index mapping, say `coarseNodeFineIdx`, for points in group C. The interpolation is straightforward

$$\text{uf}(\text{coarseNodeFineIdx}) = \text{uc};$$

With appropriate data structure, the prolongation matrix do not need to be formed.

Restriction operator. How to compute $\mathbf{I}_1^t = \mathbf{Q}_1$? Recall that to compute a L^2 projection, we need to invert the mass matrix which is not cheap. Fortunately, we are not really computing the L^2 projection of a function. Instead we are dealing with a functional! Let's recall the definition

$$(6) \quad (\mathbf{Q}_1 r, u_1) = (r, u_1) = (r, \mathbf{I}_1 u_1).$$

$\mathbf{Q}_1 r$ is simply to restrict the action of the dual $r \in (\mathbb{V}_2)'$ to the elements in \mathbb{V}_1 only. It is better to write as $\mathbf{I}_2^1 : (\mathbb{V}_2)' \rightarrow (\mathbb{V}_1)'$ and call it restriction. Note that $\mathbb{V}_1 \subset \mathbb{V}_2$ implies that $(\mathbb{V}_2)' \subset (\mathbb{V}_1)'$. So the operator \mathbf{I}_2^1 is also a natural inclusion of functional. Again r and $\mathbf{I}_2^1 r$ will have different vector representations. The matrix form of (6) is

$$(7) \quad (\mathbf{I}_2^1 r)^t \mathbf{u}_1 = r^t \mathbf{I}_1^2 \mathbf{u}_1,$$

which implies

$$\mathbf{I}_2^1 = (\mathbf{I}_1^2)^t.$$

The restriction matrix for the 1-D example with the index mapping in Fig xxx is

$$\begin{bmatrix} 1 & 0 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1 & 0 & 1/2 \end{bmatrix}.$$

Exercise 5.1. Use HB and `coarseNodeFineIdx` to code the restriction without forming the matrix.

Smoother in the coarse space. The last component is the smoother R_1 . If we know a priori the information on the PDE, we can simply choose an appropriate scaling. For example, for finite element discretization of Poisson equation on uniform grids, it coincides with the standard 5-point stencil up to a scaling h^2 . One step of Gauss-Seidel iteration can be implemented using `for` loops:

```

1  for i = 2:N-1
2      for j = 2:N-1
3          u(i,j) = (b(i,j) + (u(i-1,j) + u(i+1,j) + u(i,j-1) + u(i,j+1))) / 4;
4      end
5  end

```

For general operator A , if we want to choose more accurate local subspace solver, say Gauss-Seidel method, we need to know the matrix A_1 . Of course we can assemble one if we have the coarse grid. But there are several reasons to abandon this approach. First, the assembling is time consuming. Indeed this is one of the criticism of finite element methods comparing with finite difference scheme. Second it requires the information of the mesh and PDE. Then it will be problem dependent. Third, we have a better way to do it.

Recall that the operator $A_1 : \mathbb{V}_1 \rightarrow \mathbb{V}_1$ is just the restriction of A to the space \mathbb{V}_1 . Namely

$$(A_1 u_1, v_1) := (A u_1, v_1) = (A I_1 u_1, I_1 v_1) = (I_1^t A I_1 u_1, v_1),$$

which implies $A_1 = I_1^t A I_1$ and in the matrix form

$$A_1 = I_2^1 A_2 I_1^2.$$

So we can apply a triple product to form the stiffness matrix in the coarse grid.

6. SSC AND MULTIGRID METHOD

In this section, we shall discuss implementation of successive subspace correction method when the the subspaces are nested. Let $\mathbb{V} = \sum_{i=J}^1 \mathbb{V}_i$ be a space decomposition into nested subspaces, i.e. the subspaces are nested, i.e.

$$\mathbb{V}_1 \subset \mathbb{V}_2 \subset \cdots \subset \mathbb{V}_J = \mathbb{V}.$$

Recall that the operator formation of SSC method is

```

1  function e = SSC(r)
2      % Solve the residual equation Ae = r by SSC method
3      rnew = r;
4      for i = J:-1:1
5          ri = Ii'*rnew;    % restrict the residual to subspace
6          ei = Ri*ri;      % solve the residual equation in subspace
7          e = e + Ii*ei;   % prolongate the correction to the big space
8          rnew = r - A*e;  % update residual
9      end

```

Here we change the `for` loop from `J:-1:1` to reflect to the ordering from fine to coarse. The operators $I_i^t = Q_i : \mathbb{V} \rightarrow \mathbb{V}_i$ and $I_i : \mathbb{V}_i \rightarrow \mathbb{V}$ are related to the finest space. When the subspaces are nested, we do not need to return to the finest space every time. Suppose $r_i = I_i^t(r - A e_{\text{old}})$ in the subspace \mathbb{V}_i is known, and the correction e_i is added to

$e_{\text{new}} = e_{\text{old}} + e_i$. We can compute r_{i-1} by the relation:

$$\begin{aligned} r_{i-1} &= Q_{i-1}(r - Ae_{\text{new}}) \\ &= Q_{i-1}Q_i(r - Ae_{\text{old}} - Ae_i) \\ &= Q_{i-1}(r_i - Q_iAQ_i^t e_i) \\ &= Q_{i-1}(r_i - A_i e_i). \end{aligned}$$

Here in the second step, we make use of the nest property $\mathbb{V}_{i-1} \subset \mathbb{V}_i$ to write $Q_{i-1} = Q_{i-1}Q_i$. Similarly the correction step can be also done accumulatively. Let us rewrite the correction as

$$e = e_J + I_{J-1}e_{J-1} + \dots + I_1e_1.$$

The correction can be computed by the loop

$$e_i = e_i + I_{i-1}^i e_{i-1}, \quad i = 2 : J$$

Therefore only the prolongation and restriction operators between consecutive levels are needed.

By these discussion, SSC on a nested space decomposition will results a V-cycle multi-grid method. We summarize the algorithm below. We use notation e_i, r_i to emphasis that in each level we are solving the residual equation $A_i e_i = r_i$ and assume the transfer operators and discretization matrices have been computed already using the method discussed in the previous section.

```

1 function e = Vcycle(r, J)
2 ri = cell(J,1); ei = cell(J,1);
3 ri{J} = r;
4 for i = J:-1:2
5     ei{i} = R{i}\ri{i}; % pre-smoothing
6     ri{i-1} = Res{i-1}*(ri{i}-Ai{i}*ei{i}); % update and restrict residual
7 end
8 ei{1} = Ai{1}\ri{1}; % exact solver in the coarsest level
9 for i = 2:J
10    ei{i} = ei{i} + Pro{i}*ei{i-1}; % prolongate and correct
11    ei{i} = ei{i} + R{i}'\ (ri{i}-Ai{i}*ei{i}); % post-smoothing
12 end
13 e = ei{J};

```

In the second loop (/) part, we add a post-smoothing step and choose R_i^t as the smoother which is the transpose of the pre-smoothing operator. For example, if $R_i = (D_i + L_i)^{-1}$ is the forward Gauss-Seidel method, then the post-smoothing is backward Gauss-Seidel $(D_i + U_i)^{-1}$. This choice will make the operator B symmetric and thus can be used as preconditioner.

7. ALGEBRAIC MULTIGRID METHOD

The multigrid methods discussed in the previous sections depends heavily on the geometry of the underlying meshes and therefore called geometric multigrid methods. In most applications, the grid could be totally unstructured without hierarchical structure. In some cases, only the matrix is given without any grid information. It would be desirable to still solve the algebraic equation using multi-grid idea.

Looking the procedure carefully, the hierarchical structure of grids is used to construct the transfer operators. After that, the matrix equation in the coarse grid can be assembled and the smoother can be algebraically taking as G-S $\text{tril}(A)$ or weighted Jacobi

$\omega \text{diag}(A)$. Two essential ingredients are needed to construct the prolongation operator from a coarse grid to a fine grid

- (1) Index map from coarse nodes to fine nodes.
- (2) Weight used in the interpolation of evaluating the fine variables.

Let us revisit these two ingredients in an algebraic way.

Coarsening. Recall that the node in the fine level can be classified into \mathcal{C} and \mathcal{F} . Now if only matrix is given, a node will be understood as an abstract point. No coordinate is associated to it. Suppose A is an $N \times N$ matrix. The fine nodes is the index set $\mathcal{N} = \{1, 2, \dots, N\}$. A subset \mathcal{C} of \mathcal{N} will be identified as the node of a ‘coarse grid’ and the rest is \mathcal{F} , i.e. $\mathcal{N} = \mathcal{C} \cup \mathcal{F}$. In addition, for any $i \in \mathcal{F}$, the neighboring ‘coarse nodes’ $\mathcal{J}(i) \subset \mathcal{C}$ should be found. In hierarchical meshes case, $\mathcal{J}(i)$ is simply HB array which only contains two coarse nodes. In summary we need to pick up \mathcal{C} and construct $\mathcal{J}(i)$ for all $i \in \mathcal{F}$.

From the given matrix A , we could construct a weighted graph $\mathcal{G} = \mathcal{G}(A) = (\mathcal{V}, \mathcal{E})$. The vertices are the node set \mathcal{N} and the edge $[i, j]$ exists if $a_{ij} \neq 0$.