5.3. **Multigrid.** We now consider a very effective method for the solution of the linear system of equations that arises from the finite element discretization of elliptic boundary value problems.

The idea of multigrid is to combine two different methods: one that is effective in reducing the high frequency components of the error and one that is effective in reducing the low frequency components of the error. We have already seen an example of an iterative method that reduces the high frequency components of the error. A key idea of multigrid is that the smooth part of the error can be approximated well by a coarser grid than the one on which we started. To describe this idea, we consider first a two-grid algorithm for solving the linear system arising from the discretization:

$$\text{Find } u_h \in V_h \text{ such that } a(u_h, v_h) = (f, v_h), \quad v_h \in V_h,$$

where we let V_h denote the space of continuous piecewise linear functions on a grid of maximize size h. We assume that the mesh was obtained from a mesh of size $2h$ by joining the midpoints of each triangle to form 4 subtriangles. We let R denote a smoothing operator. For example, we showed previously that the iteration

$$x^{n+1} = x^n - \alpha(Ax^n - b).$$

damps out the high frequency eigenmodes for $0 < \rho(A) < 2$. For the problem we are solving, the vector x^n will be the vector of coefficients in the expansion of the approximate solution u_h^n in terms of the basis functions $\{\phi_j\}$, i.e., $u_h^n = \sum_j x_j^n \phi_j$. The vectors Ax^n and b are defined by:

$$(Ax^n)_i = \sum_j a(\phi_j, \phi_i) x_j^n = a(u_h^n, \phi_i), \qquad b_i = (f, \phi_i).$$

We choose the usual basis functions such that $\phi_i = 1$ at vertex p_i and zero at all other vertices. Then any piecewise linear function v can be written as $v = \sum_i v(p_i)\phi_i$.

It will be convenient at times to write equations in terms of the finite element functions, rather than the vector of coefficients in the expansion in terms of the basis functions. One way to do this is to define a mesh-dependent inner product on the space V_h by

$$(u, v)_h = h^2 \sum_k u(p_k) v(p_k)$$

where the sum ranges over all the vertices p_k of the mesh. Using this inner product, we then define $f_h \in V_h$ and an operator A_h mapping V_h to V_h by

$$(f_h, v)_h = (f, v), \quad v \in V_h, \qquad (A_h u, v)_h = a(u, v), \quad u, v \in V_h.$$

We then note that

$$b_i = (f, \phi_i) = (f_h, \phi_i)_h = h^2 \sum_k f_h(p_k)\phi_i(p_k) = h^2 f_h(p_i)$$

$$(Ax^n)_i = a(u_h^n, \phi_i) = (A_h u_h^n, \phi_i)_h = h^2 (A_h u_h^n)(p_i).$$

With this notation, our smoothing iteration becomes:

$$\sum_i x_i^{n+1}\phi_i = \sum_i x_i^n \phi_i - \alpha \sum_i [(Ax^n)_i - b_i]\phi_i = \sum_i x_i^n \phi_i - h^2\alpha \sum_i [(A_h u_h^n)(p_i) - f_h(p_i)]\phi_i.$$

Rewriting this, we get,

$$u_h^{n+1} = u_h^n - h^2\alpha[A_h u_h^n - f_h] \equiv R u_h^n.$$

Then the two level algorithm is described as follows:

Let u_h^0 be an initial approximation to u in the space V_h.

1. *Smoothing Step*: Perform m smoothing steps, i.e., define $u_h^m = R^m u_h^0$.

2. *Coarse-Grid Correction*: Compute the solution $w_{2h} \in V_{2h}$ to the problem: Find $w_{2h} \in V_{2h}$ such that

$$a(u_h^m + w_{2h}, v_{2h}) = (f, v_{2h}), \quad v_{2h} \in V_{2h}.$$

Then set $u_h^{m+1} = u_h^m + w_{2h}$.

To write this down in a simple way, we define intergrid transfer operators. The coarse-to-fine operator $I_{2h}^h : V_{2h} \to V_h$ is defined by $I_{2h}^h v = v$ for all $v \in V_{2h}$ (since V_{2h} is subspace of V_h). The fine-to-coarse intergrid transfer operator $I_h^{2h} : V_h \to V_{2h}$ is defined by:

$$(I_h^{2h} w, v)_{2h} = (w, I_{2h}^h v)_h, \quad v \in V_{2h}, \quad w \in V_h.$$

Note, if we choose $v = \phi_i$ where ϕ_i is the basis function on the coarse grid associated with the vertex p_i, then $(I_h^{2h} w, v)_{2h} = 4h^2 I_h^{2h} w(p_i)$. Now $I_{2h}^h \phi_i$ will be equal to 1 at p_i, equal to 1/2 at vertices of the fine mesh adjacent to p_i and equal to zero at all other vertices. Hence,

$$(w, I_{2h}^h \phi_i)_h = h^2[w(p_i) + (1/2)\sum_j w(p_j)],$$

where the sum is taken over adjacent vertices. Hence,

$$I_h^{2h} w(p_i) = [w(p_i) + (1/2)\sum_j w(p_j)]/4.$$

Using this notation, we may rewrite the variational equation as:

$$\begin{aligned}(A_{2h} w_{2h}, v_{2h})_{2h} = a(w_{2h}, v_{2h}) &= (f, v_{2h}) - a(u_h^{k,1}, v_{2h}) = (f_h, v_{2h})_h - (A_h u_h^{k,1}, v_{2h})_h \\ &= (f_h, I_{2h}^h v_{2h})_h - (A_h u_h^{k,1}, I_{2h}^h v_{2h})_h = (I_h^{2h}(f_h - A_h u_h^{k,1}), v_{2h})_{2h}.\end{aligned}$$

Hence,

$$w_{2h} = A_{2h}^{-1} I_h^{2h}(f_h - A_h u_h^m).$$

Then

$$u_h^{m+1} = u_h^m + I_{2h}^h w_{2h} = u_h^m + I_{2h}^h A_{2h}^{-1} I_h^{2h}(f_h - A_h u_h^m).$$

In the two level scheme, we computed the exact solution of the linear system on the coarse level. Instead, we could extend this idea by incorporating additional levels. To describe the complete algorithm for several levels, we first choose a coarse triangulation $\mathcal{T}_0$ with mesh size h_0. We then subdivide each triangle into four congruent triangles by joining the midpoints of the edges and denote by $\mathcal{T}_1$ the resulting triangulation. Continue in this manner producing meshes $\mathcal{T}_1, \ldots, \mathcal{T}_N$. With each triangulation $\mathcal{T}_k$, we associate a space of continuous, piecewise linear polynomials that we denote by V_N. The problem we want to solve is:

$$\text{Find } u_N \in V_N \text{ such that } a(u_N, v) = (f, v), \quad v \in V_N.$$

We now construct a recursive algorithm for obtaining an approximate solution.

The kth level iteration: We let $MG(k, z_0, g)$ denote the approximate solution of the equation $A_k z = g$ obtained from the kth level iteration with initial guess z_0.

For $k = 0$, $MG(0, z_0, g) = A_0^{-1} g$, i.e., the solution obtained from a direct method.

For $k \geq 1$, $MG(k, z_0, g)$ is obtained recursively in 3 steps. Let $m_1 > 0$ and $m_2 \geq 0$ be integers and $p = 1$ or 2.

Presmoothing step: For $1 \leq l \leq m_1$, let $\quad z_l = z_{l-1} + \alpha h^2 (g - A_k z_{l-1}) = R z_{l-1}$.

Error correction step: Let $\bar{g} = I_k^{k-1}(g - A_k z_{m_1})$ and $q_0 = 0$. For $1 \leq i \leq p$, let

$$q_i = MG(k-1, q_{i-1}, \bar{g}).$$

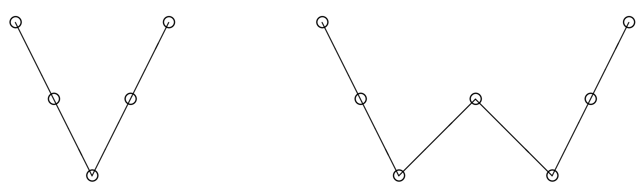
Then $\quad z_{m_1+1} = z_{m_1} + I_{k-1}^k q_p$.

Post-smoothing step: Set $z_{m_1+m_2+1} = R^{m_2} z_{m_1+1}$, where

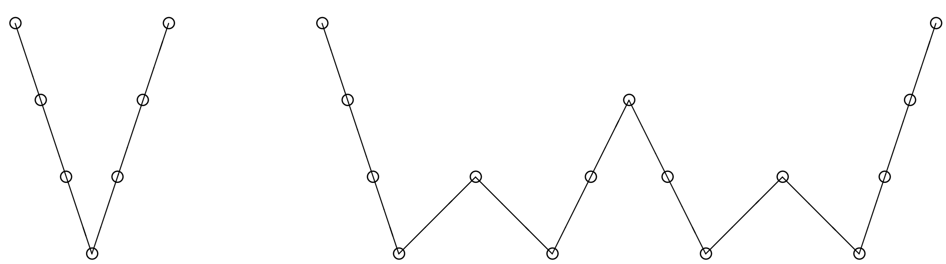
$$z_l = R z_{l-1} \equiv z_{l-1} + \alpha h^2 (g - A_k z_{l-1}).$$

Then the output of the kth level iteration is: $MG(k, z_0, g) = z_{m_1+m_2+1}$.

When $p = 1$ this is called the V-cycle method and when $p = 2$, it is called the W-cycle method. This is illustrated in the following diagrams.



V-Cycle and W-cycle on three levels



V-Cycle and W-cycle on four levels

The Full Multigrid Algorithm:

For $k = 0$, let $\hat{u}_0 = A_0^{-1} f_0$.

For $k = 1, \cdots, N$, the approximate solutions $\hat{u}_k$ are obtained recursively by the following algorithm:

$$\begin{aligned} u_0^k &= I_{k-1}^k \hat{u}_{k-1} \\ u_l^k &= MG(k, u_{l-1}^k, f_k), \quad 1 \le l \le r, \\ \hat{u}_k &= u_r^k. \end{aligned}$$

So we start at level 0 and at each finer level, we take the initial guess to be $I_{k-1}^k \hat{u}_{k-1}$.

To establish convergence of the multigrid algorithm, one first shows that the kth level iteration is a contraction with respect to the energy norm $\|v\|_E^2 = a(v, v)$ with a contraction number γ independent of k. This means that if z denotes the exact solution of the linear system at level k, i.e., corresponding to the subspace V_k and $MG(k, z_0, g)$ denotes the approximation obtained by multigrid with initial guess z_0, then for some $\gamma < 1$,

$$\|z - MG(k, z_0, g)\|_E \le \gamma \|z - z_0\|_E.$$

Then we have the following convergence result.

Theorem 9. *Let u_N denote the exact solution of the finite element method on the mesh $\mathcal{T}_N$ and $\hat{u}_N$ the approximation obtained by the full multigrid algorithm. If the kth level iteration is a contraction with a contraction number γ independent of k and if r is large enough, then there exists a constant $C > 0$ independent of h such that*

$$\|u_N - \hat{u}_N\|_1 \le C h_N \|u\|_{2,\Omega}.$$

Note that since $\|u - u_N\| \le C h_N \|u\|_{2,\Omega}$, using multigrid to approximately solve the linear system only introduces an error of the same magnitude we are already making by using the approximation scheme. The reason multigrid is so successful is that the work involved (number of arithmetic operations) is proportional to the dimension of the finite element space (an asymptotically optimal result).