There are three important topics left to consider:

- Programs for multigrid (SCSCwebpage/cs_multigrid)
- Convergence of multigrid. (Saad Section 13.5)
- Algebraic multigrid. (Saad Section 13.6)

Programs for multigrid (SCSCwebpage/cs_multigrid)

Convergence of multigrid

We’ll follow Saad in talking about the analysis of a 2-level V-cycle.

Solving the problem exactly on the $2h$ grid is usually too expensive, so that is not a practical algorithm.

But the general V-cycle is a perturbed version of this, since the coarse mesh problem is then solved inexactly rather than exactly.

Notation

- We let $R_h$ be the operator takes values on grid $h/2$ and produces values on grid $h$. (This moves from fine to coarse grid.) We let $P_h$ be the operator takes values on grid $h$ and produces values on grid $h/2$. (This moves from coarse to fine grid.)
- We choose $R_h = P_h^T$.
- $A_{2h} = R_{2h} A_h P_{2h}$

Since we only work with 2 meshes, we’ll drop the subscripts on $R$ and $P$. 

1
V-Cycle: 2 levels

\[ v_h = V\text{-Cycle}(v_h, A_h, f_h, \eta_1, \eta_2) \]

Perform \( \eta_1 \) G-S iterations on \( A_h u_h = f_h \) using \( v_h \) as the initial guess, obtaining an approximate solution that we still call \( v_h \).

Compute \( v_{2h} \) to solve \( A_{2h} v_{2h} = R(f_h - A_h v_h) \).

Set \( v_h = v_h + P v_{2h} \).

Perform \( \eta_2 \) G-S iterations on \( A_h u_h = f_h \) using \( v_h \) as the initial guess, obtaining an approximate solution that we still call \( v_h \).

Analysis

- We want to express this as a stationary iterative method:
  
  \[ v_h^{(new)} = M v_h + c. \]

- The algorithm:
  - Apply \( \eta_1 \) G-S iterations. This iteration matrix is \( M_{GS}^{\eta_1} \). This operator is called the smoother.
  - Solve the coarse grid problem and add the correction.
    * We get the residual by taking \( -A_h \) times the current guess.
    * Then we restrict to the coarse grid with \( R \) and apply \( A_{2h}^{-1} \).
    * Then we prolong using \( P \).
    * Then we add this onto the iterate.
  - Apply \( \eta_2 \) G-S iterations. This iteration matrix is \( M_{GS}^{\eta_2} \).
  - So the matrix \( M \) is
    
    \[
    M = M_{GS}^{\eta_2} (I - PA_{2h}^{-1} R A_h) M_{GS}^{\eta_1} \equiv M_{GS}^{\eta_2} T M_{GS}^{\eta_1}.
    \]

- The following analysis (without loss of generality) takes \( \eta_1 = 0 \).

Assumptions

(Saad, Section 13.5) We make two (standard) assumptions, one concerning the smoother and one concerning the grids.

- **Smoothing property:**
  
  \[
  \| M_{GS}^{\eta_2} z \|_A^2 \leq \| z \|_A^2 - \alpha \| A_h z \|_{D^{-1}}^2,
  \]
  
  for all \( z \) and for some constant \( \alpha \) independent of \( h \).
  
  \( (A = A_h, D = \text{diag}(A_h).) \)
• **Approximability:**

\[
\min_w \| z - Pw \|^2_D \leq \beta \|z\|^2_A,
\]

where the minimum is taken over all vectors \( w \) on the coarse grid, and where the constant \( \beta \) is independent of \( h \).

---

**Properties**

- **P1:** \( I - T = PA^{-1}_h RA_h \) is a projection operator, meaning that \((I - T)^2 = I - T \) and \( T^2 = T \).

  **Proof:**

  \[
  (I - T)^2 = (PA^{-1}_h RA_h)(PA^{-1}_h RA_h) = PA^{-1}_h A^{-1}_h RA_h = PA^{-1}_h RA_h = I - T.
  \]

  The proof that \( T^2 = T \) follows easily.

- **P2:**

  \[
  T^T A_h = A_h - A_h R^{-1}_h A^{-1}_h P^T A_h = A_h(I - PA^{-1}_h RA_h) = A_h T.
  \]

- **P3:**

  \[
  TP = (I - PA^{-1}_h RA_h)P = P - PA^{-1}_h RA_h P = P - PA^{-1}_h A_{2h} = P - P = 0.
  \]

- **P4:** \( \| Tz \|^2_A \leq \|z\|^2_A \).

  **Proof:**

  \[
  \|z\|^2_A = \|(T + (I - T))z\|^2_A = z^T T^T A_h Tz - 2z^T T^T A_h (I - T)z + z^T (I - T)^T A_h (I - T)z = \|Tz\|^2_A + \|(I - T)z\|^2_A,
  \]

  and the result follows. (We have used **P1** and **P2** to get rid of the plum colored term in the middle line.)

- **CS:** The Cauchy-Schwartz inequality says

  \[ |x^T y| \leq \|x\| \|y\| \].
• **P5**: For all $y$, if $z = Tw$ (i.e., $z$ is in the range of $T$), then by $P3$,

$$z^T A_hPy = w^T P h A_hPy = w^T T A h TPy = 0.$$  

• **P6**: Therefore, if $z$ is in the range of $T$, then $\|z\|_A \leq \sqrt{\beta} \| A_h z \|_{D^{-1}}$.

**Proof**: We let $y$ be the minimizer in the definition of approximability. Then

$$\|z\|^2_A = z^T A_h z$$

$$= z^T A_h D^{-1/2} D^{1/2} (z - Py)$$  

(by $P5$)

$$\leq \| D^{-1/2} A_h z \| \| D^{1/2} (z - Py) \|$$  

(by $CS$)

$$= \| A_h z \|_{D^{-1}} \| (z - Py) \|_D$$

$$\leq \sqrt{\beta} \| A_h z \|_{D^{-1}} \|z\|_A.$$  

(by Approximability)

• Finally,

$$\| M_{GS}^h Tz \|^2_A \leq \| Tz \|^2_A - \alpha \| A_h Tz \|^2_{D^{-1}}$$  

(by Smoothing)

$$\leq \| Tz \|^2_A - \frac{\alpha}{\beta} \| Tz \|^2_A$$  

(by $P6$)

$$= \left( 1 - \frac{\alpha}{\beta} \right) \| Tz \|^2_A$$

$$\leq \left( 1 - \frac{\alpha}{\beta} \right) \|z\|^2_A.$$  

(by $P4$)

• If we let $z$ denote the error in the solution before applying a multigrid iteration, the inequality above guarantees reduction in the error (when measured in the $A_h$-norm) at a rate independent of $h$, so the number of iterations necessary to reduce the error by a given amount is constant, independent of $h$, so it can be done using an amount of work proportional to a work unit.

---

**Algebraic multigrid**

We introduced multigrid by presenting grids.

There have been many attempts to extend the ideas to gridless problems.

Let’s think of $h$ as a parameter related to the size of the matrix problem, so that $A_h$ is (at most) twice as big as $A_{2h}$.

The definition of a multigrid method depends on only two things:

• We choose $R_h = P_h^T$.

• $A_{2h} = R_{2h} A_h P_{2h}$ and this matrix is (at most) half the size of $A_h$. 

---


This is easy to accomplish in general.

The convergence analysis depends on two additional assumptions:

- **Smoothing property:**
  \[
  \|M_{GS}^h z\|_A^2 \leq \|z\|_A^2 - \alpha \|A_h z\|_{D^{-1}}^2,
  \]
  for all \(z\) and for some constant \(\alpha\) independent of \(h\).

- **Approximability:**
  \[
  \min_w \|z - Pw\|_D^2 \leq \beta \|z\|_A^2,
  \]

This is not easy, and we will abandon hope of it.

---

**One approach to Algebraic Multigrid: Graph coloring**

- Use the graph of the matrix \(A\) to define the geometry of the problem.
- Color the nodes of the graph to define the coarse grid.
- Define the restriction operator \(R\) to take a weighted average of values on the fine grid to create a value on the coarse grid.

See Saad’s “guiding principles” on p. 442.

---

**A second approach to Algebraic Multigrid: ILU**

Suppose, by coloring the graph or by other means, we can obtain a permutation matrix \(Q\) so that

\[
QAQ^T = \begin{bmatrix} B & F \\ E & C \end{bmatrix}
\]

where \(B\) is block diagonal.

Then our matrix problem can be written

\[
\begin{bmatrix} B & F \\ E & C \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}.
\]

Note that

\[
\begin{bmatrix} B & F \\ E & C \end{bmatrix} = \begin{bmatrix} I & 0 \\ EB^{-1} & I \end{bmatrix} \begin{bmatrix} B & F \\ 0 & S \end{bmatrix}
\]

where the **Schur complement** is

\[
S = C - EB^{-1}F.
\]

So we have reduced our problem to:
• Let \( w_1 = f_1 \) and solve \( w_2 = f_2 - EB^{-1}w_1 \).
• Solve \( Su_2 = w_2 \).
• Solve \( Bu_1 = w_1 - Fu_2 \).

and we only need to be able to solve linear systems involving \( B \) and \( S \) easily.

The problem involving \( S \) can be thought of as the coarse grid problem.

Repeating this recursively (next, looking for a similar partition of \( S \)) gives a multilevel algorithm.

When the Schur complement gets too dense, we can substitute an approximation to it and then apply a V-cycle.

---

**Final words**

• We used GS as a smoother. Other SIMs (e.g., Jacobi) can also be used.
• (Geometric) multigrid is a terrific method for pde’s and certain other problems.
• Algebraic multigrid needs a lot more work.
• Saad’s Section 13.7 is titled, “Multigrid vs. Krylov Methods.” A better approach is to think of multigrid as a preconditioner.
  – If multigrid works well, then only 1 iteration of the Krylov method is used, and the algorithm is just multigrid.
  – If multigrid is slow, the Krylov method will accelerate it.