Multi-grid for Linear Problems (LCS Scheme)

- First discuss a multi-grid algorithm for linear problems: *Linear Correction Scheme* or LCS.

- Can be applied to essentially any elliptic system, but may be useful to think in terms of two dimensional model problem

- First note that multi-grid algorithms generally involve solutions of sets of equations of form
$$L^\ell u^\ell = s^\ell \tag{87}$$
where 'right-hand-side" or "source" function, s^ℓ, may *not* coincide with source function $f^{\ell_{\max}}$ from fine-grid problem
$$L^{\ell_{\max}} u^{\ell_{\max}} = f^{\ell_{\max}} \tag{88}$$

- Also note that will ignore the role of (discretized) boundary conditions in the discussion of multi-grid techniques.

- However, especially for non-Dirichlet conditions, treatment of boundary conditions in multi-grid is non-trivial (see supplied references)

Multi-grid for Linear Problems (LCS Scheme)

- Begin discussion of LCS scheme assuming that grid hierarchy has just two grids: fine grid with mesh spacing h, defines problem of interest

$$L^h u^h = f^h\,, \tag{89}$$

  and coarse grid with mesh spacing $2h$: serves as "accelerator" of fine grid solution process.

- Denote by $\tilde{u}^h$ current approximation to u^h,

- Given some initial estimate of $\tilde{u}^h$, start by applying relaxation: for concreteness (as well as the fact that it is commonly used in practice), assume that are using Gauss-Seidel relaxation with red-black ordering (RBGS).

- Then, after a few sweeps (perhaps as few as 1 or 2!), will find that the residual, $\tilde{r}^h$

$$\tilde{r}^h := L^h \tilde{u}^h - f^h \tag{90}$$

  as well as the error

$$\tilde{e}^h := u^h - \tilde{u}^h \tag{91}$$

  will be *smooth* on the scale of the mesh.

Multi-grid for Linear Problems (LCS Scheme)

- Now think of task of completing fine-grid solution, equivalently, driving the residual, $\tilde{r}^h$, to 0, in terms of computing a *correction*, v^h, such that

$$u^h = \tilde{u}^h + v^h \tag{92}$$

- Due to linearity of L^h, correction satisfies

$$L^h v^h = -\tilde{r}^h \tag{93}$$

- By assumption, residual, $\tilde{r}^h$ is smooth: therefore so must be the correction, v^h (this reasonably assumes that L and L^h are "well-behaved").

- First key observation underlying LCS is that this smoothness in the residual and correction means that can *sensibly* pose a coarse-grid version of (93).

Multi-grid for Linear Problems (LCS Scheme)

- That is, now consider coarse grid problem

$$L^{2h} v^{2h} = -I_h^{2h} \tilde{r}^h \equiv s^{2h}, \qquad (94)$$

- L^{2h} is the coarse-grid difference operator (which employs the same FDA as L^h)
- I_h^{2h} is a fine-grid-to-coarse-grid transfer operator known as a *restriction operator* (details discussed later).
- Now assume that have computed an (approximate) solution, $\tilde{v}^{2h}$, of (94).
- Then update $\tilde{u}^h$ via

$$\tilde{u}^h := \tilde{u}^h + I_{2h}^h \tilde{v}^{2h} \qquad (95)$$

- I_{2h}^h is a coarse-to-fine transfer operator, i.e. a prolongation operator, not necessarily the same as the prolongation operator $\bar{I}_{2h}^h$ discussed previously.

Multi-grid for Linear Problems (LCS Scheme)

- However, I_{2h}^{h} will typically also involve polynomial interpolation
- Interpolation of computed correction, $\tilde{v}^{2h}$, to fine grid will introduce new high-frequency components in residual and solution error.
- But these high-frequency components can be effectively annihilated with a few more relaxation sweeps
- At this point, *all* components of the residual will have been substantially reduced.
- Sequence of posing coarse grid problem, solving it, then updating the fine grid unknown is known as a coarse-grid correction (CGC).
- Next key observation is that can apply the smooth/CGC/smooth process to solve the coarse grid problem (94) itself, then keep recursing, solving problems with mesh spacings, $4h$, $8h$, etc.
- Eventually get to the *coarsest* level: problem there is so small (perhaps 3×3 unknowns, including boundary values!) that it is computationally *trivial* to *solve* the problem (not just smooth it) using relaxation.
- Solution of coarse grid problem then followed by succession of coarse-to-fine transfers of various coarse grid corrections that have been computed

Multi-grid for Linear Problems (LCS Scheme)

- Apply post-CGC relaxation sweeps after each restriction operation to ensure that high-frequency components (re)-introduced by restriction are killed off

- Process of working from finest to coarsest grid then back to finest grid is known as *multi-grid V-cycle*

- Important question: How many relaxation sweeps should be applied before and after each coarse-grid correction?

- Answer generally depends on specifics of problem, but typically introduce parameters, p and q:

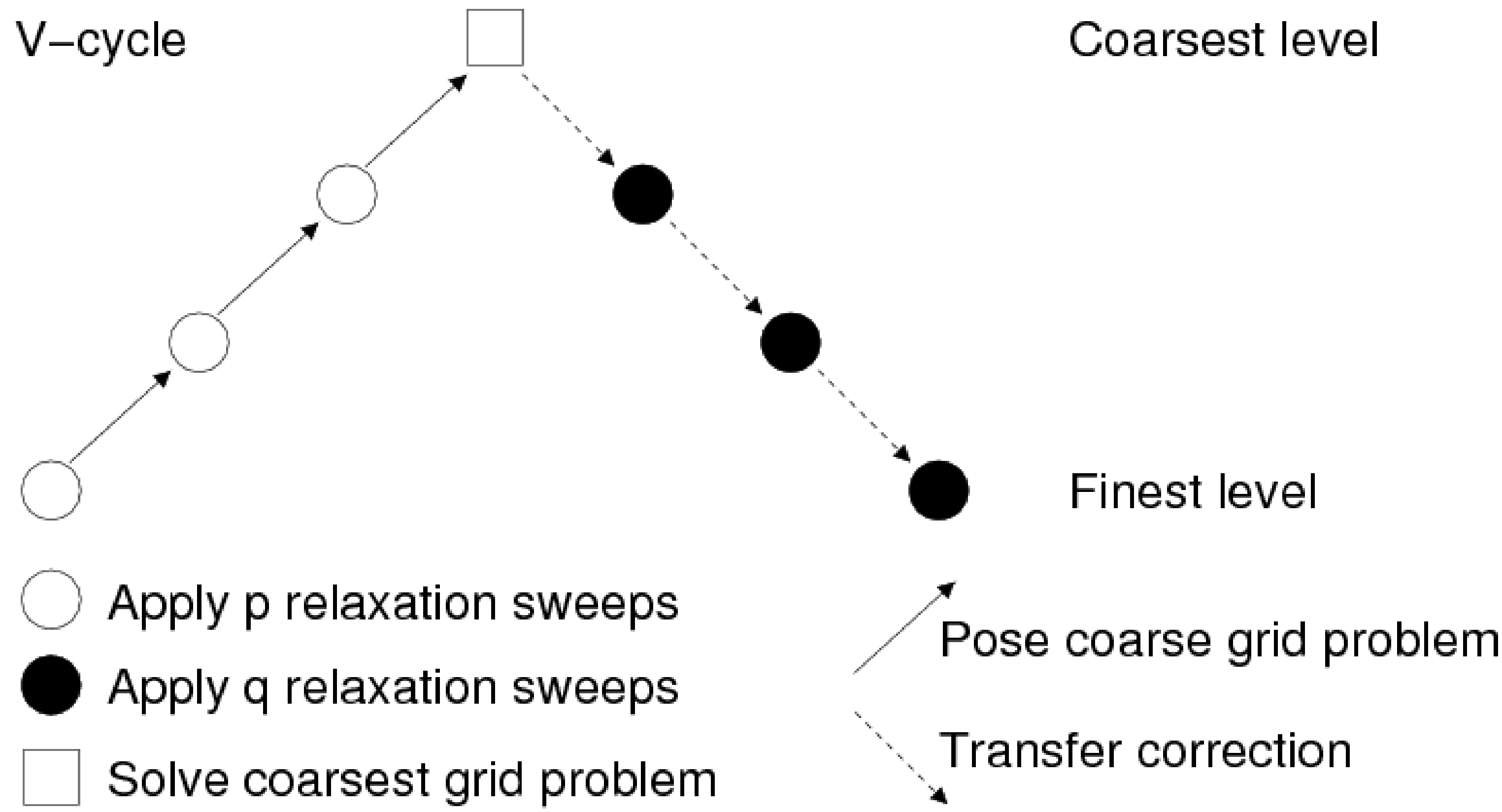
$$p \equiv \text{number of pre-CGC sweeps} \tag{96}$$

$$q \equiv \text{number of post-CGC sweeps} \tag{97}$$

  and then set p and q to specific values.

- Results in what is known as a *fixed V-cycle* algorithm (pseudo-coded later)

LCS Multi-grid V-cycle



- Schematic representation of a LCS multi-grid V-cycle, for the case of 4-level mesh hierarchy.
- Open and filled circles represent pre- and post-CGC relaxation sweeps, respectively; open square denotes *solution* of the coarse-grid problem.
- Intergrid transfers are represented by the two different types of arrows.

Multi-grid for Linear Problems (LCS Scheme)

- Experience shows that, assuming initial estimate of $\tilde{u}^h$ was good (as should be the case if we are using a multi-level strategy), then will find

$$\|\tilde{r}^h\| \sim \|\tau^h\| \tag{98}$$

  after a single V-cycle, and fine grid problem will be effectively solved.

- If need smaller residual, apply additional V-cycles: residual size should go down by constant factor (perhaps 10) per cycle

- Finally, in case of multiple V-cycles, no need to perform pre-CGC smoothing sweeps on the finest level except for *first* V-cycle

Pseudo-code for LCS V-cycle

procedure `lcs_vcycle` (ℓ, `cycle`, p, q)
   Cycle from fine to coarse levels
   do $m = \ell, 2, -1$
      if `cycle` $= 1$ **or** $m \neq \ell$ **then**
         Apply pre-CGC smoothing sweeps
         do p **times**
            $u^m :=$ `relax_rb` (u^m, s^m, h^m)
         end do
         Set up coarse grid problem
         $u^{m-1} := 0$
         $s^{m-1} := -I_m^{m-1} (L^m u^m - s^m)$
      end if
   end do
   Solve coarsest-level problem
   $u^1 :=$ `relax_rb` (u^1, s^1, h^1) **until** $\|\tilde{r}^1\| \leq \epsilon_1$

Pseudo-code for LCS V-cycle (cont.)

Cycle from coarse to fine levels
do $m = 2, \ell, +1$
Apply coarse-grid correction
$u^m := u^m + I^m_{m-1} u^{m-1}$
Apply post-CGC smoothing sweeps
do q **times**
$u^m :=$ `relax_rb` (u^m, s^m, h^m)
end do
end do
end procedure

- Note: have introduced an additional parameter, ϵ_1, which is the convergence criterion to *solve* the coarsest-grid problem by relaxation.

- Cost of relaxation on coarsest grid is typically so small compared to the fine grid work: ϵ_1 can typically be set quite conservatively with essentially no impact on algorithm performance

Pseudo-code for LCS Driver Routine

procedure $\texttt{lcs\_mg}\,(\,\ell_{\max}, \texttt{ncycle}, p, q\,)$
  do $\ell\ =\ 1\,,\,\ell_{\max}\,,\,+1$
    if $\ell = 1$ **then**
      On coarsest level, arbitrarily initialize solution to 0
      $u^1 := 0$
    else
      Initialize solution via prolongation from coarse grid solution
      $u^\ell := \bar{I}^{\ell}_{\ell-1} u^{\ell-1}$
    end if
    Initialize source function
    $s^\ell := f^\ell$
    Perform $\texttt{ncycle}$ *V-cycles*
    do $\texttt{cycle} = 1\ ,\,\texttt{ncycle}$
      $\texttt{lcs\_vcycle}\,(\,\ell, \texttt{cycle}, p, q\,)$
    end do
  end do
end procedure