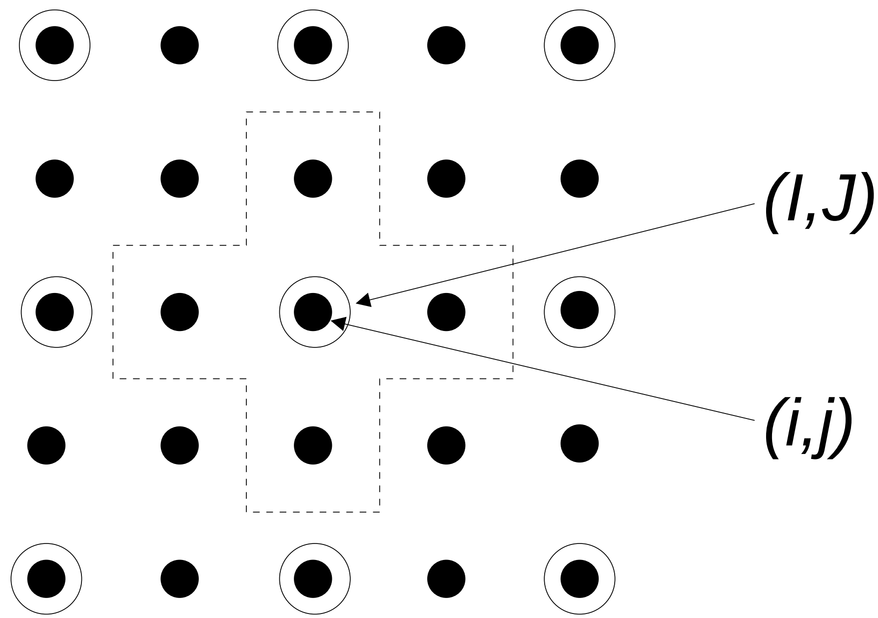
LCS Scheme: Transfer Operators

- Need to discuss details of grid-to-grid transfer operators (i.e. the prolongation and restriction operators)
- Proper construction of $\bar{I}^{\ell}_{\ell+1}$, $I^{\ell}_{\ell+1}$, and $I^{\ell+1}_{\ell}$ extremely important for any multi-grid algorithm, can't treat in depth here (see references)
- Important: multi-grid technique (i.e. the V-cycle) induces non-trivial interactions between transfer operators and relaxation (smoothing) scheme
- Thus, for example, prolongation and restriction operators that work well with red-black Gauss-Seidel relaxation do not necessarily work well with lexicographic Gauss-Seidel
- Now describe specific implementations for transfer operators that work well for problems such as our model equation—as well as similar systems in two and three dimensions—in conjunction with red-black Gauss-Seidel smoothing.

LCS Scheme: Half-weighted Injection



- Illustration of stencil used for the operation of half-weighted restriction.
- Small filled and large open circles represent fine and coarse grid locations respectively
- Task of restriction operator is to define all of the coarse grid values from the fine grid unknowns.
- Dotted line indicates the 5-pt stencil that is used in the half-weighted transfer

LCS Scheme: Half-weighted Injection

- Half-weighted restriction in two dimensions then defined by

$$u^{\ell}_{I,J} = \frac{1}{2}u^{\ell-1}_{i,j} + \frac{1}{8}\left(u^{\ell-1}_{i-1,j} + u^{\ell-1}_{i+1,j} + u^{\ell-1}_{i,j+1} + u^{\ell-1}_{i,j-1}\right) \tag{99}$$

- The term "half-weighting" comes from fact that fine-grid unknown located at the same physical location as target coarse-grid unknown has a weight of $1/2$ in the transfer

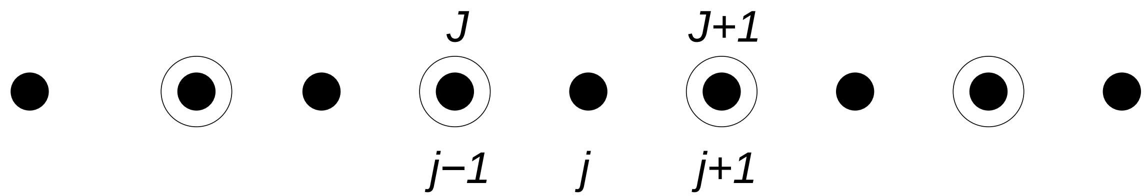
- Half-weighted restriction operator may be analogously defined in 1 and 3 dimensions, where factor of $1/8$ is replaced by $1/4$ and $1/12$, respectively, and sum is over central fine-grid unknown, and its 2 and 6 nearest neighbors respectively.

- Note: "obvious" restriction formula

$$u^{\ell}_{I,J} = u^{\ell-1}_{i,j} \tag{100}$$

which is called *injection*, fails miserably when used with red-black Gauss-Seidel relaxation, although it tends to work well when lexicographically-ordered GS is the smoother

LCS Scheme: Bi-linear Interpolation



- Now consider prolongation operator, $I_{\ell-1}^{\ell}$ (I_{2h}^{h})
- LCS uses this to transfer the coarse grid correction back to the fine grid
- In current case, *bilinear interpolation* is found to work well
- First consider one dimensional problem: interpolation from a mesh Ω^h to Ω^{2h}
- Fine-grid values u_{j-1}^{ℓ} and u_{j+1}^{ℓ} are trivially interpolated as "copies" of corresponding coarse-grid unknowns

$$u_{j-1}^{\ell} = u_{J}^{\ell-1} \quad (101)$$

$$u_{j+1}^{\ell} = u_{J+1}^{\ell-1} \quad (102)$$

LCS Scheme: Bi-linear Interpolation

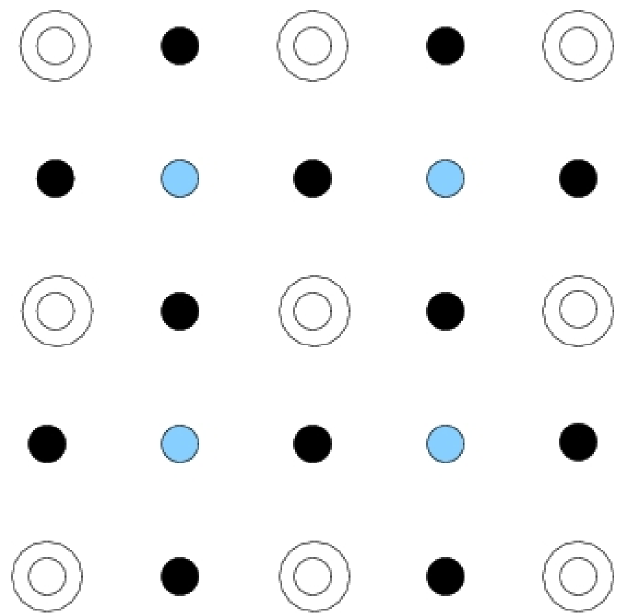
- Intuitively, other half of the fine-grid quantities can be linearly interpolated by "averaging" the neighboring values: assuming u^ℓ_{j-1}, u^ℓ_{j+1} have been defined as above, u^ℓ_j is given by

$$u^\ell_j = \frac{1}{2}\left(u^\ell_{j-1} + u^\ell_{j+1}\right) \tag{103}$$

- Easy to show (Taylor series) that this is accurate to $O(h^2)$

- Moreover, note that (103) is the *unique* formula for linear interpolation of u^ℓ_j from nearest neighbors u^ℓ_{j-1} and u^ℓ_{j+1}.

LCS Scheme: Bi-linear Interpolation

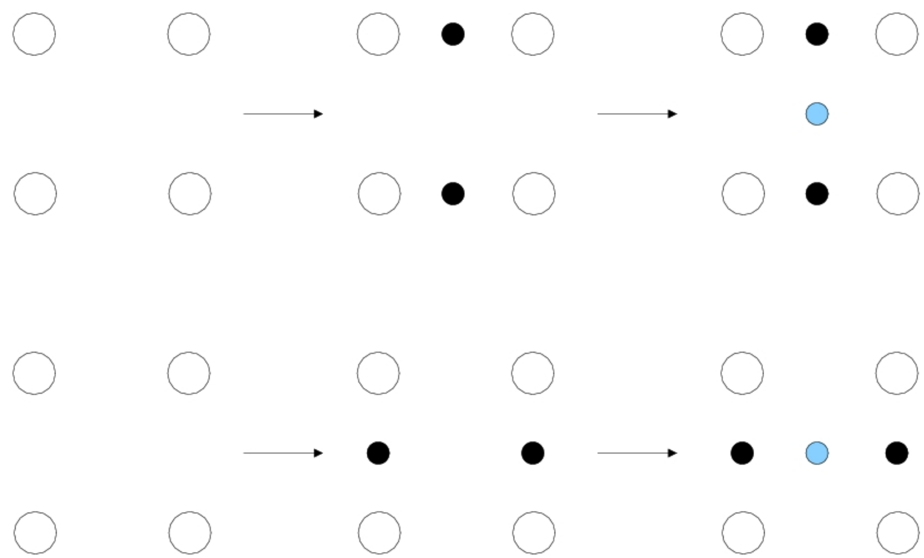


- Figure shows portion of coarse mesh (large open circles) and fine mesh (small circles), in two dimensions

- Three distinct types of fine grid points

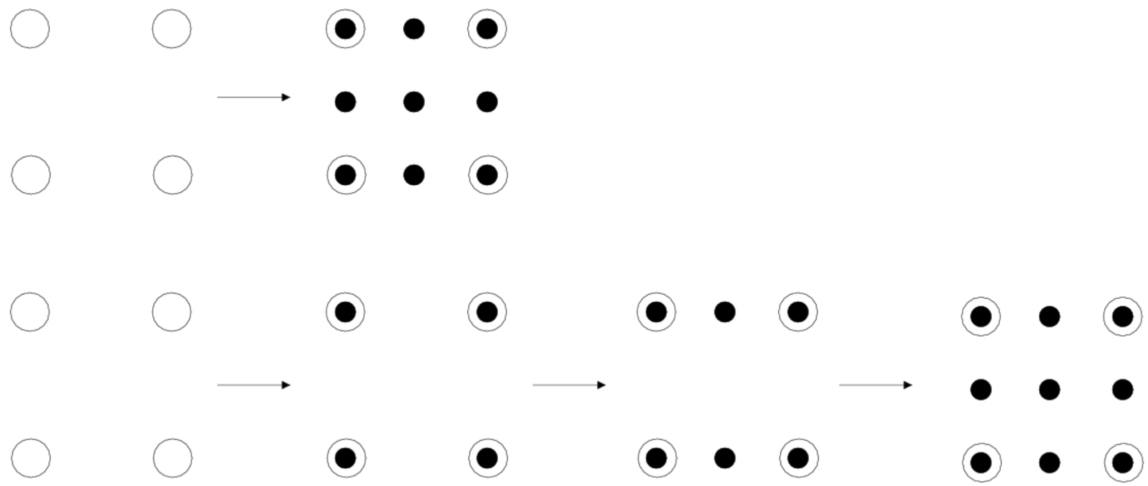
  1. Copies of coarse grid unknowns (small open circles)
  2. Points whose values can be computed using 1-d interpolation in either the x or y direction (small black circles)
  3. Points whose values require genuine 2-d interpolation (light blue circles).

LCS Scheme: Bi-linear Interpolation



- Third class of points have precisely 4 nearest neighbors in the coarse grid
- As can be verified via Taylor series (as well as elementary geometry—how many distinct, non-colinear points are needed to define a plane?), only 3 of these are needed to produce a linear interpolant.
- Thus in 2-d case (as well as in higher dimensions) there is no *unique* formula for linear interpolation from Ω^{2h} to Ω^{h}.
- Figure shows two of the (infinite number of) possible schemes

LCS Scheme: Bi-linear Interpolation



- Would hope that the precise details of the interpolation would not matter
- Experience shows that this is the case
- Are thus free to implement the linear interpolation operator more or less as we please,
- Figure shows an approach with is particularly convenient to implement, and which extends to more dimensions, and to higher order interpolation

LCS Scheme: Transfer Operators

- Last transfer operator to consider is prolongation operator, $\tilde{I}^{\ell}_{\ell-1}$
- Used in multi-level solution process to *initialize* fine-grid unknown from coarse-grid solution coarse grid problem.
- Old rule of thumb due to Brandt suggests using *bi-cubic* interpolation: leading order error term would be $O(h^4)$.
- However, empirically find that *bi-linear* interpolation actually provides better overall performance, again in the context of model problem being smoothed with red-black Gauss-Seidel, and with the other transfer operators defined as above
- Thus, in current case, have

$$\tilde{I}^{\ell}_{\ell-1} \equiv I^{\ell}_{\ell-1} \tag{104}$$

Computational Cost of Multi-Grid

- But (again, in two dimensions)

$$w_{\ell-1} \sim \frac{1}{4} w_\ell \tag{139}$$

so

$$W_\ell \sim (p + \sigma q)\, w_\ell \left(1 + \frac{1}{4}\sigma + \frac{1}{16}\sigma^2 + \cdots + \left(\frac{\sigma}{4}\right)^{\ell-2}\right) + \sigma^{\ell-1} W_1 \tag{140}$$

where W_1 is work needed to *solve* coarsest-grid problem $L^1 u^1 = s^1$.

- So long as the number of CGCs, σ, required to solve any fine grid problem, satisfies $\sigma < 4$, have

$$1 + \frac{1}{4}\sigma + \frac{1}{16}\sigma^2 + \cdots + \left(\frac{\sigma}{4}\right)^{\ell-2} < \left(1 - \frac{\sigma}{4}\right)^{-1} \tag{141}$$

- Thus

$$W_\ell \le w_\ell \left(\frac{p + \sigma q}{1 - \sigma/4}\right) + \sigma^{\ell-1} W_1 \tag{142}$$

Computational Cost of Multi-Grid

- But

$$w_\ell \sim cN_\ell \tag{143}$$

where c is some constant, and from assumption that $\sigma < 4$, we have

$$\sigma^{\ell-1} < 4^{\ell-1} \sim \frac{N_{\ell-1}}{N_1} \tag{144}$$

where N_1 is the number of points on the coarsest grid (another constant).

- Putting these results together, have

$$W_\ell \le N_\ell \left(\frac{c\,(p + \sigma q)}{1 - \sigma/4} + \frac{W_1}{N_1} \right) = O(N_\ell) \tag{145}$$

- Thus, as previously claimed, multi-grid can solve the N algebraic equations resulting from the discretization of elliptic PDEs in $O(N)$ time.