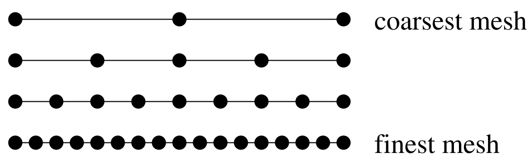
Numerical Methods for Partial Differential Equations WS 2009 / 2010
Tutorial 8 Monday, 14 December 2009, 10.15–11.45, T 212

In this tutorial we consider the MDS (multilevel diagonal scaling) preconditioner. Let $\{\mathcal{T}_\ell\}_{1\le\ell\le L}$ be a family of subdivisions of the interval $\Omega = (0,\,1)$. We start with the uniform (i. e. equidistant) mesh $\mathcal{T}_1$ and a fixed number L of levels and define the meshes $\mathcal{T}_2, \dots, \mathcal{T}_L$ recursively by uniform refinement as shown below.



Let $V = H^1(0,\,1)$ be our working space. For each $\ell = 1, \dots, L$ we define

$$V_\ell := \{v \in V : v|_T \in \mathcal{P}^1 \quad \forall T \in \mathcal{T}_\ell\} \;=\; \operatorname{span}\{\varphi_{\ell,i}\}_{i=0}^{n_\ell}$$

with the nodal basis functions $\{\varphi_{\ell,i}\}_{i=0}^{n_\ell}$ and n_ℓ being the number of elements of $\mathcal{T}_\ell$.

41 Consider two subsequent meshes $\mathcal{T}_\ell$ (coarse mesh) and $\mathcal{T}_{\ell+1}$ (fine mesh) and the finite element function $w_\ell \in V_\ell$ on the coarse mesh $\mathcal{T}_\ell$ with the basis expansion

$$w_\ell(x) \;=\; \sum_{i=0}^{n_\ell} w_{\ell,i}\,\varphi_{\ell,i}$$

and its vector representation $\underline{w}_\ell = (w_{\ell,i})_{i=0}^{n_\ell}$. Clearly $w_\ell \in V_{\ell+1}$, i. e., there exist basis coefficients $(w_{\ell+1,i})_{i=1}^{n_{\ell+1}} =: \underline{w}_{\ell+1}$ such that

$$w_\ell(x) \;=\; \sum_{i=0}^{n_{\ell+1}} w_{\ell+1,i}\,\varphi_{\ell+1,i}$$

Represent $\underline{w}_{\ell+1}$ in terms of $\underline{w}_\ell$ and find a matrix $I_\ell^{\ell+1} \in \mathbb{R}^{n_{\ell+1}\times n_\ell}$ such that

$$\underline{w}_{\ell+1} \;=\; I_\ell^{\ell+1}\,\underline{w}_\ell\,.$$

42 Let $R \in V^*$ be a bounded linear functional. For the fine mesh $\mathcal{T}_{\ell+1}$ define the coefficient vector $\underline{r}_{\ell+1} = (r_{\ell+1,i})_{i=0}^{n_{\ell+1}}$ by $r_{\ell+1,i} := \langle R,\,\varphi_{\ell+1,i}\rangle$. Then,

$$\langle R,\,v_{\ell+1}\rangle \;=\; \sum_{i=0}^{n_{\ell+1}} r_{\ell+1,i}\,v_{\ell+1,i} \;=\; (\underline{r}_{\ell+1},\,\underline{v}_{\ell+1})_{\ell_2} \qquad \forall v_{\ell+1} \in V_{\ell+1}\,.$$

Find a representation of $\underline{r}_\ell = (r_{\ell,i})_{i=0}^{n_\ell}$ in terms of $\underline{r}_{\ell+1}$ such that

$$\langle R,\,v_\ell\rangle \;=\; (\underline{r}_\ell,\,\underline{v}_\ell)_{\ell_2} \qquad \forall v_\ell \in V_\ell\,.$$

Show that

$$\underline{r}_\ell \;=\; I_{\ell+1}^\ell\,\underline{r}_{\ell+1} \qquad \text{with} \qquad I_{\ell+1}^\ell = (I_\ell^{\ell+1})^\top$$

Programming:

43 Write a function `RefineUniform(↓coarse_mesh, ↑fine_mesh)` that computes the refined mesh `fine_mesh`$=\mathcal{T}_{\ell+1}$ from the coarse mesh `coarse_mesh`$=\mathcal{T}_\ell$ as shown above.

44 (a) Write a function `Prolongate(↓coarse_vector, ↑fine_vector)` that computes `fine_vector`$=\underline{w}_{\ell+1} = I_\ell^{\ell+1}\underline{w}_\ell$ from `coarse_vector`$=\underline{w}_\ell$.

(b) Write a function `Restrict(↓fine_residual, ↑coarse_residual)` that computes `coarse_residual`$=\underline{r}_\ell = I_{\ell+1}^{\ell}\underline{r}_{\ell+1}$ from `fine_residual`$=\underline{r}_{\ell+1}$.

45 Implement the MDS preconditioner C_{MDS}^{-1} for a hierarchy of recursively refined meshes $\mathcal{T}_1, \ldots, \mathcal{T}_L$ according to the lecture.

Hint: Use the following structure: (Attention, this is neither an exact C^{++}code nor complete!)

```
class MDSPreconditioner
{
 private:
  int numLevels_;                 // number of levels
  Array<Preconditioner> jprec_;   // Jacobi preconditioner for each level

 public:
  MDSPreconditioner (int levels)
  { numLevels_ = levels; jprec_.resize (numLevels_); }

  // lev: level (0..numLevels_-1)
  void setJacobiPreconditioner (int lev, const Preconditioner& p)
  { jprec_[lev] = p; }

  void solve (const Vector& r, Vector& w) const
  { MDS (numLevels_-1, r, w); }

 private:

  // lev: level (0..numLevels_-1)
  void MDS (int lev, const Vector& r, Vector& w) const
  {
    jprec_[lev].solve (r, w);
    if (lev > 0)
      {
        Vector r_coarse, w_coarse, w_fine;
        Restrict (r, r_coarse);
        MDS (lev-1, r_coarse, w_coarse);    // recursive call
        Prolongate (w_coarse, w_fine);
        w += w_fine;
      }
  }

}; // class MDSPreconditioner
```

46 Solve the problem given in Exercise 28* (see Tutorial 5) with the MDS-preconditioned CG method. Start with a coarse mesh containing only one element and try different numbers of levels $L = 2, 3, \ldots, 7$ etc. such that the finest mesh has at least more than 100 elements. Report the number of iterations to reach the relative accuracy $\varepsilon = 10^{-6}$.

Hint: In main program, perform a loop over all levels. On each level, assemble the stiffness matrix and load vector, and implement the given boundary conditions. Then give the matrix (or its diagonal) to the `MDSPreconditioner` (with the correct level index). Unless you have reached level L, refine your mesh. If you have reached level L, your `MDSPreconditioner` is initialized, and you can run CG in a similar way as in Tutorial 7.