Numerical Methods for Partial Differential Equations WS 2012 / 13
Tutorial 9 Monday, December 3, 2012, 10.15–11.45, S2 054

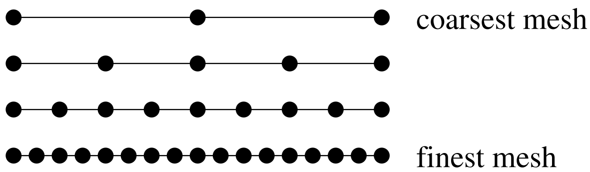
---

46 Show that for the iterates of the Richardson method and the method of steepest descent (Gradientenverfahren),

$$x_k \in x_0 + \mathcal{K}_k(A, r_0)$$

holds. Argue the same for the iterates of the Conjugate Gradient method, where you may use results from Lemma 1.63.

In the next three examples, we consider the MDS (multilevel diagonal scaling) preconditioner. Let $\{\mathcal{T}_\ell\}_{\ell=1}^{L}$ be nested meshes of the interval (0, 1). We start with a fixed mesh $\mathcal{T}_1$ (with a few elements) and construct the other meshes by uniform refinement as shown below.



For each $\ell = 1, \dots, L$ we define

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

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

47 Consider two consecutive meshes $\mathcal{T}_\ell$ and $\mathcal{T}_{\ell+1}$ and the corresponding finite element spaces $V_\ell \subset V_{\ell+1}$. Recall that for every $w_\ell \in V_\ell$ there exist a vector $\underline{w}_\ell = [w_{\ell,i}]_{i=0}^{n_\ell}$ such that

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

Let $w_\ell \in V_\ell$ be fixed. Since $w_\ell \in V_{\ell+1}$ as well, there exists a vector $\underline{w}_{\ell+1} = [w_{\ell+1,i}]_{i=0}^{n_{\ell+1}}$ such that

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

Write the coefficients $w_{\ell+1,i}$ in terms of $w_{\ell,i}$. Find a matrix $I_\ell^{\ell+1} \in \mathbb{R}^{n_{\ell+1}+1\times n_\ell+1}$ such that

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

48 Let $R \in V^*$ denote a bounded linear functional. Let $\ell < L$ be fixed and define the residual 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,

$$\langle R,\,v_{\ell+1}\rangle \;=\; \sum_{i=0}^{n_{\ell+1}} r_{\ell+1,i}\,v_{\ell+1,i} \;=\; (\underline{r}_{\ell+1},\,\underline{v}_{\ell+1})_{\ell^2}$$

for all $v_{\ell+1} \in V_{\ell+1}$ with basis representation $\underline{v}_{\ell+1}$.
As above, define $\underline{r}_\ell = [r_{\ell,i}]_{i=0}^{n_\ell}$ with $r_{\ell,i} := \langle R,\,\varphi_{\ell,i}\rangle$.

Write $\underline{r}_\ell$ in terms of $\underline{r}_{\ell+1}$ such that $\langle R,\,v_\ell\rangle \;=\; (\underline{r}_\ell,\,\underline{v}_\ell)_{\ell^2}$ for all $v_\ell \in V_\ell$ with basis representation $\underline{v}_\ell$. Show that

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

49 Consider the $H^1(0,1)$-coercive bilinear form $a(u,v) = \int_0^1 u'v' + uv\,dx$, and let $V_\ell, \ell = 1, ..., L$ be defined as above. (This corresponds to the classical problem $-u'' + u = f$ in $(0,1)$ with $-u'(0) = g_0$ and $u'(1) = g_1$.)

Let the subspace decomposition $V_L = \sum_{\ell=1}^{L} \sum_{i=0}^{n_\ell} V_{\ell,i}$ with $V_{\ell,i} = \operatorname{span}\{\varphi_{\ell,i}\}$ be given.

The MDS preconditioner C_L^{-1} is defined via

$$C_L^{-1} \underline{r}_L = \underline{w}_L \ ,$$

where the vector $\underline{r}_L$ corresponds to the functional R (as above) and $\underline{w}_L$ to the function w_L, where

$$w_L = \sum_{\ell=1}^{L} w_\ell = \sum_{\ell=1}^{L} \sum_{i=0}^{n_\ell} w_{\ell,i}\ \varphi_{\ell,i} \ ,$$

with $w_{\ell,i}$ such that

$$a(w_{\ell,i}\ \varphi_{\ell,i}, \varphi_{\ell,i}) = \langle R, \varphi_{\ell,i} \rangle \ .$$

Write C_L^{-1} explicitly in terms of the matrices $I_\ell^{\ell+1}$ and $D_\ell = diag(K_\ell)$, where K_ℓ is the stiffness matrix on level ℓ, i.e. $[K_\ell]_{i,j} = a(\varphi_{\ell,j}, \varphi_{\ell,i})$.

Programming.

---

50 Define a C++class `JacobiPreconditioner` which implements the Jacobi Preconditioner $C_h = D_h = diag(K_h)$. Implement a member function which solves the linear system $C_h \underline{w}_h = \underline{r}_h$ for $C_h = D_h$ (diagonal) and for a given vector $\underline{r}_h$. The class declaration should look similar to this:

```
class JacobiPreconditioner{
public:
  JacobiPreconditioner (const SMatrix& mat);
  ~JacobiPreconditioner ();
  Solve(const Vector& r, Vector& s); //solves C.s=r
  (...)
private:
  Vector diag_;
  (...)
};
```

51 Now solve the boundary value problem of example **45** with the preconditioned Conjugate Gradient Method with a Jacobi preconditioner (adapt `pcg.hh` from the webpage). Report the number of iterations for 2^k elements, $k = 4, \dots, 10$. Compare to the number of iterations needed by CG (Exercise **45**).