Technische Universität München
Lehrstuhl für Informatik V
Dr. T. Neckel, D. Pflüger

SS 2010
Sheet 8
24.6.2010

Algorithms of Scientific Computing (Algorithmen des Wissenschaftlichen Rechnens)

Generating Systems

A useful tool when dealing with hierarchical bases is the representation through *generating systems*. We will see how this works for the onedimensional case.

Consider the space V_n of piecewise linear, continuous functions $u : [0,1] \to \mathbb{R}, u(0) = u(1) = 0$ living on a regular grid with mesh width $h_n = 2^{-n}$.

$$\tilde{\Psi}_n := \bigcup_{l=1}^{n} \{\phi_{l,i} : 1 \le i < 2^l\}$$

is not a basis for $n > 1$, yet it is a generating system: for every $u \in V_n$ one can find a (not necessarily unique) representation as a linear combination

$$u = \sum_{l=1}^{n} \tilde{w}_l = \sum_{l=1}^{n} \sum_{1 \le i < 2^l} v_{l,i}\phi_{l,i}\,, \qquad \text{for } \tilde{w}_l \in V_l.$$

Write the coefficients $v_{l,i}$ in ascending order with respect to l first and i then. The resulting vector shall further be referenced by $\vec{v}^E$.

Outline algorithms that for given vector $\vec{v}^E$ compute representations of the same function in the generating system $\tilde{\Psi}_n$

- using the nodal basis: $\vec{v}^{E,N}$ with $v_{l,i} = 0$ for $l < n$ *Algorithm NB:*

  $l = 1, \ldots, n-1$:
  $\quad i = 1, \ldots, 2^l - 1$
  $\quad\quad v_{l+1,2i-1} \mathrel{+}= v_{l,i}/2$
  $\quad\quad v_{l+1,2i} \mathrel{+}= v_{l,i}$
  $\quad\quad v_{l+1,2i+1} \mathrel{+}= v_{l,i}/2$
  $\quad\quad v_{l,i} = 0$

- using the hierarchical basis: $\vec{v}^{E,H}$ with $v_{l,i} = 0$ for even i. *Algorithm HB:*

  $l = n-1, \ldots, 1$:
  $\quad i = 1, \ldots, 2^l - 1$
  $\quad\quad v_{l+1,2i-1} \mathrel{-}= v_{l+1,2i}/2$
  $\quad\quad v_{l+1,2i+1} \mathrel{-}= v_{l+1,2i}/2$
  $\quad\quad v_{l,i} \mathrel{+}= v_{l+1,2i}$
  $\quad\quad v_{l+1,2i} = 0$

Now back to Finite Elements! Consider the mass matrix (coefficient matrix B for L_2 scalar product) — but this also works for other bilinear forms.

For now we assemble B^E naively, i.e. we ignore the fact that we actually do not have a basis anymore and just compute the scalar products of the ansatz functions.

(i) Show that the result of the product $B^E \vec{v}^E$ is the same for all possible representations of function u in the generating system.

The component of the vector $B^E \vec{v}^E$ that is associated with $x_{l,i}$ reads

$$\sum_{l',j} \left(\int_0^1 \phi_{l,i} \cdot \phi_{l',j} \right) v_{l',j} \, dx = \int_0^1 \phi_{l,i} \cdot \left(\sum_{l',j} \phi_{l',j} v_{l',j} \right) dx$$

which depends on the function under consideration

$$u = \sum_{l',j} \phi_{l',j} v_{l',j}.$$

(ii) Knowing the coefficients of $B^E \vec{v}^E$ for level l, how can you compute the coefficients for level $l-1$ without recomputing the scalar product or B^E itself?

$$\begin{aligned} \int_0^1 \phi_{l,i} \cdot u \, dx &= \int_0^1 \left(\frac{1}{2}\phi_{l+1,2i-1} + \phi_{l+1,2i} + \frac{1}{2}\phi_{l+1,2i+1} \right) \cdot u \, dx \\ &= \frac{1}{2} \int_0^1 \phi_{l+1,2i-1} \cdot u \, dx + \int_0^1 \phi_{l+1,2i} \cdot u \, dx + \frac{1}{2} \int_0^1 \phi_{l+1,2i+1} \cdot u \, dx \end{aligned}$$

leads to the following algorithm N–E:

$l = n-1, \ldots, 1$:
  $i = 1, \ldots, 2^l - 1$
    $(B^E \vec{v}^E)_{l,i} = \frac{1}{2}(B^E \vec{v}^E)_{l+1,2i-1} + (B^E \vec{v}^E)_{l+1,2i} + \frac{1}{2}(B^E \vec{v}^E)_{l+1,2i+1}$

(iii) How can you get from B^E to a representation of the mass matrix $B^N \in \mathbb{R}^{(2^n-1)\times(2^n-1)}$ in the nodal basis?

With scissors: In the given ordering of the unknowns we find diagonal block B^N in the lower right of B^E.

(iv) Find matrices S_1, S_2 such that

$$B^E = S_1 B^N S_2.$$

So far we have been concerned with the effect the matrix B^E has on vector $\vec{v}^E$. That shall be our starting point here as well. We have to make sure that

$$B^E \vec{v}^E = S_1 B^N S_2 \vec{v}^E$$

holds.

We know that algorithm NB transforms arbitrary $\vec{v}^E$ to an "equal" representation with regard to the nodal basis. Let $S_2 = S_{NB}$ the matrix that applies this transformation, cuts the leading zeros of the image vector $\vec{v}^{E,N}$ and thus computes $\vec{v}^N$. We then have

$$B^N S_2 \vec{v}^E = B^N \vec{v}^N = \vec{f}^N$$

Now we can consider $\vec{f}^N$ as the lower part of the "full" right hand side $\vec{f}^E$. From (iii) we know that we can compute the first components using algorithm N–E. Defining $S_1 = S_{N-E}$ then yields $B^E = S_{N-E} B^N S_{NB}$ since we have

$$S_{N-E} B^N S_{NB} \vec{v}^E = \vec{f}^E = B^E \vec{v}^E.$$

Furthermore $S_{KB} = S_{N-E}^T$ holds (from the basic steps and the order of execution).

(v) Is it possible to get a finite element approximation through a linear system of equations with B^E being the coefficient matrix?

B^E is singular, however one can solve the linear system $B^E \vec{v}^E = S_{N-E} \vec{f}^N$ (the right hand side $\vec{f}^N$ being with respect to the nodal basis) using a method like for instance the Conjugate Gradients. *The result one gets is* one *possible representation of the solution that can still be transformed to the nodal basis or the hierarchical basis.*

In fact one finds the CG method to converge very fast when used together with this generating system.