How to Assemble Conforming Finite Elements on Grids with Hanging Nodes

Peter Bastian

March 17, 2008

1 Detecting Hanging Nodes

Figure 1 (left) shows a simple two-dimensional grid with hanging node refinement. How do we detect hanging nodes (denoted by $\square$ in the figure)?

Let $T_h = \{e_0, \dots, e_{K-1}\}$ denote the (nonconforming) leaf finite element grid. The corresponding set of vertices is denoted by $N_h = \{v_0, \dots, v_{N-1}\}$. Here, N_h includes all vertices of the grid, i. e. also the hanging nodes. Moreover, we denote by $N(e)$ the set of vertices of element e and by $E(v) = \{e \in T_h \mid v \in N(e)\}$ the set of all elements incident at v. Due to the hierarchical grid construction every element $e \in T_h$ is assigned a unique level number $l(e)$. With that notation in place we define the function $S : N_h \to \mathbb{N}$ as

$$S(v) = \min\{l(e) \mid e \in E(v)\}. \tag{1}$$

The numbers $S(v)$ are shown in the vertices in figure 1.

Let $\lambda = (e, e')$ be the directed intersection of elements e and e'. In the grid interface the set of intersections that any element e has with any other element are available. Note that for any $\lambda = (e, e')$ it is defined that $l(e) \geq (e')$, i. e. the higher level leaf element "knows" its lower level neighbero but *not* vice versa. For any intersection $\lambda = (e, e')$ we define $N(\lambda)$ as the set of vertices that are incident at the intersection viewed from element e. This assumes that the intersection always corresponds to a face of e when $l(e) > l(e')$. Now, hanging nodes $H \subseteq N_h$ can be characterized as follows:

$$v \in H \;\Leftrightarrow\; \exists e \in E(v), \exists \lambda = (e, e') : l(e) > l(e') \wedge v \in N(\lambda) \wedge S(v) = l(e) \tag{2}$$

Note that the set H can be determined without extending the grid interface. Note also the in 3 space dimensions and higher hanging nodes can also occur at the boundary of the domain.

Figure 1 (right) shows an example for a grid with conforming red-green refinement. Additional copies of elements ("yellow refinement") are used for purposes not of importance here. The rule above determines $H = \emptyset$ because at *conforming* intersection $\lambda = (e, e')$ with $l(e) > l(e')$ the vertices $v \in N(\lambda)$ cannot be hanging because $S(v) \leq l(e') < l(e)$.

2 Conforming Finite Elements

Although the grid with hanging node refinement is nonconforming the finite element discretization employed on such meshes is nonconforming. We illustrate this here for piece-wise (bi-, tri-) linear finite elements. In most existing codes with hanging nodes the finite element discretization is combined with a gemoetric multigrid solver. This means that the matrix corresponding to the grid T_h is never formed explicitely (only matrices corresponding to the levels are formed). Here, we concentrate on forming a finite element discretization on the "leaf" grid T_h.

The conforming finite element space is defined as

$$V_h = \{v \in C^0(\bar{\Omega}) \mid v|_e \text{ is linear}, e \in T_h\}. \tag{3}$$

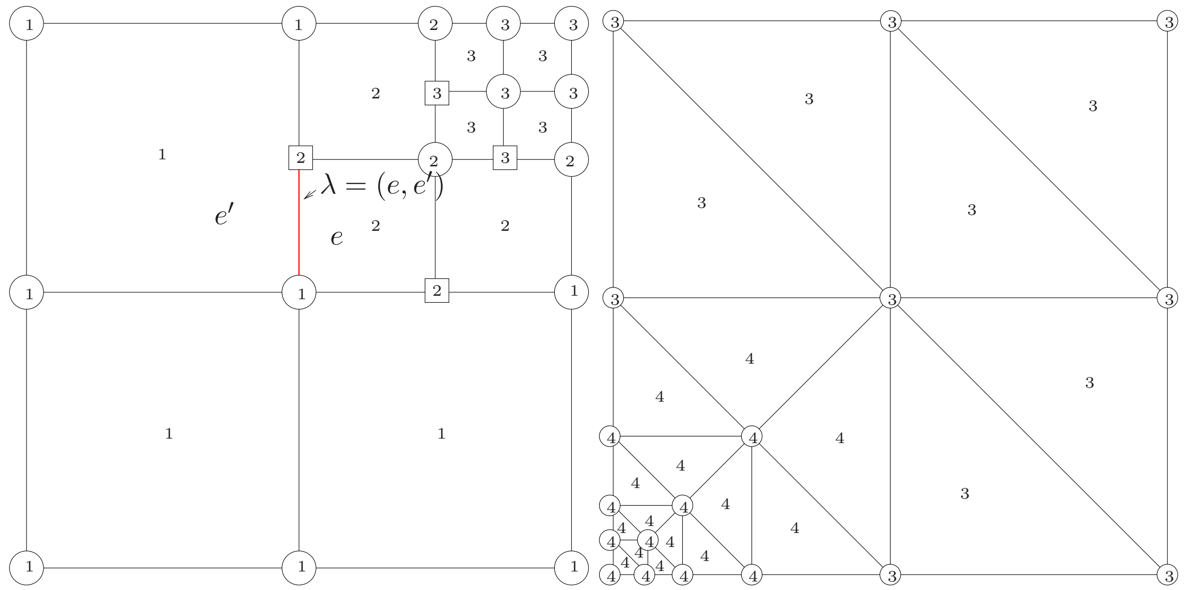

Figure 1: Two-dimensional grid with hanging nodes (left) and without hanging nodes (right). Elements are tagged with level number and vertices v with $S(v)$ defined in the text.

This induces that the value at a hanging node is constrained to the interpolatated value on the coarser side.

2.1 Characterization via Hierarchical Basis

The easiest way to characterize the conforming finite element space is via a hierarchical basis. Observe that for any $v_i \in T_h$, if we *omit* the mesh levels $l > S(v_i)$, the situation is conforming, i. e. vertex v_i is completely surrounded by elements with level $S(v_i)$. With v_i we associate the standard nodal basis function $\phi_i^{S(v)}$ where superscript denotes the mesh level. Now assume that vertices are numbered in such a way that hanging nodes are numbered last:

$$N_h = \{\underbrace{v_0, \ldots, v_{M-1}}_{v \in N_h \setminus H}, \underbrace{v_M, \ldots, v_{N-1}}_{v \in H}\}. \tag{4}$$

Then the conforming hierarchical basis is

$$\hat{\Phi} = \{\hat{\varphi}_i^{S(v_i)} \mid 0 \le i < M\} \tag{5}$$

and one has

$$V_h = \operatorname{span} \hat{\Phi}. \tag{6}$$

For assembling the stiffness matrix this basis can not be recommended because it introduces additional fill-in which is complicated to handle.

2.2 Conforming Composite Basis

We define the finite element space consisting of discontinuous piecewise polynomials:

$$D_h = \{w \in L_2(\Omega) \mid w|_e \in Q_1 \text{ (or } P_1) \}, \tag{7}$$

where $Q_1 = \{u \mid u = \sum_\alpha \mathbf{x}^\alpha, \alpha_i \le 1\}$ and $P_1 = \{u \mid u = \sum_\alpha \mathbf{x}^\alpha, \sum_i \alpha_i \le 1\}$. For every vertex $v_i \in N_h$ (i. e. also hanging nodes) we define $\psi_i \in D_h$ as follows:

$$\forall e \in E(v_i), v_j \in N(e) : \psi_i|_e(\mathbf{x}_j) = \delta_{ij}, \qquad \forall e \in T_h \setminus E(v_i) : \psi_i|_e = 0. \tag{8}$$

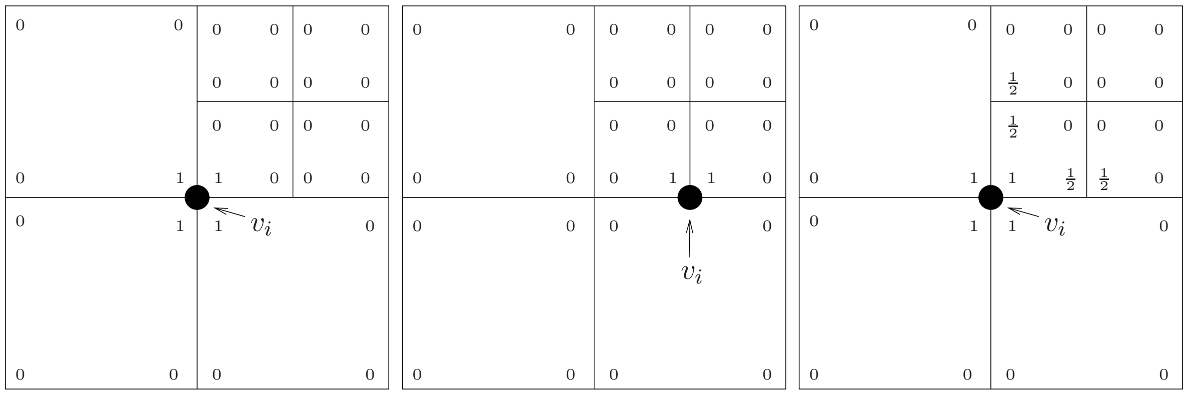

Figure 2: The discontinuous piecewise linear functions ψ_i for two vertices (left and middle figure) and a composite continuous basis function φ_i.

Examples of such functions are shown in the left and middle sketch in figure 2.

For each non-hanging node v_i we combine ψ_i with appropriately scaled functions ψ_k of neighboring hanging nodes to arrive at a conforming basis function φ_i: function :

$$\forall 0 \leq i < M \ : \ \varphi_i = \psi_i + \sum_{k=M}^{N-1} \alpha_{ik} \psi_k. \tag{9}$$

The factors α_{ik} are determined as follows: Let $e \in T_h$ be an element with non-hanging node v_i and and hanging node v_k, then α_{ik} is the evaluation of the element shape function ϕ_i^f at position $\mathbf{x}_k$ *on the father element f of e*. This the evaluation is completely local and uses only the method `geometryInFather()`. The result of this construction is shown for the center node in the right sketch in figure 2.

This defines the conforming composite nodal basis

$$\Phi = \{\varphi_i \mid 0 \leq i < M\}. \tag{10}$$

It remains to show that $\operatorname{span} \Phi = V_h$. Using the composite basis we get for any function $u \in V_h$:

$$u = \sum_{j=0}^{M-1} x_j \varphi_j = \sum_{j=0}^{M-1} x_j \left(\psi_j + \sum_{k=M}^{N-1} \alpha_{jk} \psi_k \right) = \sum_{j=0}^{M-1} x_j \psi_j + \sum_{k=M}^{N-1} \underbrace{\left(\sum_{j=0}^{M-1} \alpha_{jk} x_j \right)}_{=: x_k} \psi_k = \sum_{j=0}^{N-1} x_j \psi_j. \tag{11}$$

That is we have to ways to write a function u: Either using only non-hanging nodes $u = \sum_{j=0}^{M-1} x_j \varphi_j$ or using the simpler basis functions $u = \sum_{j=0}^{N-1} x_j \psi_j$ with the additional constraint $x_k = \sum_{j=0}^{M-1} \alpha_{jk} x_j$ for all $M \leq k < N$.

2.2.1 First way to write the system

The finite element problem now reads (as usual): Find $u \in V_h : a(u, v) = l(v) \, \forall v \in V_h$. Representing u with hanging nodes we get

$$\begin{cases} a\left(\sum\limits_{j=0}^{N-1} x_j \psi_j, \varphi_i \right) = l(\varphi_i) & 0 \leq i < M \\ x_i = \sum\limits_{j=0}^{M-1} \alpha_{ji} x_j & M \leq i < N \end{cases}. \tag{12}$$

Inserting the expression for φ_i and using linearity we arrive at

$$\begin{cases} \sum\limits_{j=0}^{N-1} x_j \left(a(\psi_j, \psi_i) + \sum\limits_{k=M}^{N-1} \alpha_{ik} a(\psi_j, \psi_k) \right) = l(\psi_i) + \sum\limits_{k=M}^{N-1} \alpha_{ik} l(\psi_k) & 0 \le i < M \\ x_i = \sum\limits_{j=0}^{M-1} \alpha_{ji} x_j & M \le i < N \end{cases} . \tag{13}$$

Thus the matrix entries are as follows

$$0 \le i < M \ : \ (A)_{ij} = a(\psi_j, \psi_i) + \sum_{k=M}^{N-1} \alpha_{ik} a(\psi_j, \psi_k) \tag{14}$$

$$M \le i < N, 0 \le j < M \ : \ (A)_{ij} = -\alpha_{ji} \tag{15}$$

$$M \le i, j < N \ : \ (A)_{ij} = \delta_{ij} \tag{16}$$

Note that the sparsity pattern is slightly extended compared to the standard case. However, matrix entries are computed from the entries of the standard local stiffness matrix only the accumulation to the global matrix and the evaluation of the α_{ji} factors is new.

Clearly the linear system to be solved can be written in 2×2 block form, where the first block $x_R = (x_0, \ldots, x_{M-1})^T$ corresponds to the non-hanging nodes and the second block $x_H = (x_M, \ldots, x_{N-1})^T$ corresponds to the hanging nodes:

$$\begin{pmatrix} A_{RR} & A_{RH} \\ A_{HR} & I \end{pmatrix} \begin{pmatrix} x_R \\ x_H \end{pmatrix} = \begin{pmatrix} b_R \\ 0 \end{pmatrix}. \tag{17}$$

2.2.2 Second way to write the system

A second way to write the system is to ignore hanging nodes and write in the usual way:

$$\sum_{j=0}^{M-1} x_j a(\varphi_j, \varphi_i) = l(\varphi_i) \quad 0 \le i < M. \tag{18}$$

In order to compute $a(\varphi_j, \varphi_i)$ we use the expansion in the composite basis:

$$\begin{aligned} a(\varphi_j, \varphi_i) &= a\left(\psi_j + \sum_{l=M}^{N-1} \alpha_{jl} \psi_l, \psi_i + \sum_{k=M}^{N-1} \alpha_{ik} \psi_k \right) \\ &= a(\psi_j, \psi_i) + \sum_{k=M}^{N-1} [\alpha_{jk} a(\psi_k, \psi_i) + \alpha_{ik} a(\psi_j, \psi_k)] + \sum_{l=M}^{N-1} \sum_{k=M}^{N-1} \alpha_{jl} \alpha_{ik} a(\psi_l, \psi_k). \end{aligned} \tag{19}$$

and for the right hand side we obtain as before

$$l(\varphi_i) = l(\psi_i) + \sum_{k=M}^{N-1} \alpha_{ik} l(\psi_k) \tag{20}$$

Formally this second way of writing the system corresponds to an elimination of A_{RH} in (17) by using $x_H = -A_{HR} x_R$:

$$\begin{pmatrix} A_{RR} - A_{RH} A_{HR} & 0 \\ A_{HR} & I \end{pmatrix} \begin{pmatrix} x_R \\ x_H \end{pmatrix} = \begin{pmatrix} b_R \\ 0 \end{pmatrix}. \tag{21}$$

In both variants the sparsity pattern is extended in the following way: All nodes of a leaf element with hanging nodes are connected to all nodes of its father element.

To do: Think about the parallel case.