Block-Diagonal Preconditioning for the *hp*-FEM

Pavel Solin
hp-FEM group, University of Nevada, Reno
Academy of Sciences, Prague, Czech Republic
http://hpfem.org
July 2009

This is an old idea by Babuska et al. I like to write it up since this preconditioning is (almost) PDE-independent and it also works for the multimesh *hp*-FEM. Of course, it will perform best for elliptic problems, but it would be very interesting to try it for other problem types as well. It should definitely become part of Hermes soon. I will start with describing brielfy why it is useful to do preconditioning at all.

Basic idea of preconditioning

Let $SY = F$ be the matrix system obtained from the finite element discretization. We insert there a nonsingular matrix P and its inverse to obtain $SP^{-1}PY = F$. Denote $SP^{-1} = R$ and $PY = X$. The condition number of R should be less than the condition number of S, so that the problem $RX = F$ is easier to solve than the original problem $SY = F$. With X in hand, one retrieves Y from the problem $PY = X$. The preconditioner should be designed in such a way that this is easy as well. So, we have two conditions that make a good preconditioner:

1. SP^{-1} should be a matrix with low condition number.
2. P should be a matrix which is very easy to solve – for example diagonal, block-diagonal with small blocks, etc.

In the solution of the system $RX = F$, the matrix R usually is not constructed explicitly. An iterative solver needs to be able to calculate efficiently a product of R with an arbitrary vector V. This is done in two steps:

1. Denote $P^{-1}V = W$ and solve for W from the system $PW = V$
2. Multiply S with W.

Also look at Wikipedia, they have a nice explanation under the keyword ”Preconditioner”.

Construction of the block-diagonal preconditioner

Let’s explain the idea in 2D, in three dimensions the situation is analogous. A prerequisite for this preconditioner is that higher-order edge functions are enumerated edge-wise (edge functions belonging to the same edge are neighbors in the list of DOF). The stiffness matrix S has the well-known 3×3 block structure:

VV	*VE*	*VB*
EV	*EE*	*EB*
BV	*BE*	*BB*

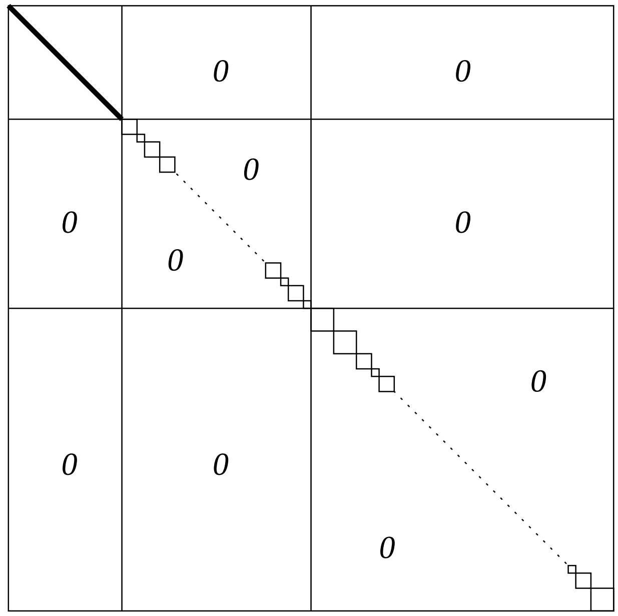
The only three blocks of interest regarding the preconditioner are the diagonal blocks VV (containing products of vertex functions with other vertex functions), EE (products of higher-order edge functions with other edge functions), and BB (bubble functions with other bubbles).

Since the supports of bubble functions are elementwise-local, obviously the BB block is block-diagonal. Each of its diagonal blocks corresponds to one element and its size is given by the number of bubble functions on that element.

With the above described enumeration of edge functions, the block EE also contains small diagonal blocks. Each one corresponds to one mesh edge and contains the products of all edge functions on that edge. There are other nonzero elements in the EE block but we will forget them.

The VV block doen not have any nice structure, unless we do some special enumeration of vertices. Or, unless we use domain decomposition and split the vertices into several groups according to the subdomains and the interface part. But let's not do this now – instead, we will just forget all off-diagonal elements of the VV block.

Summarized, the preconditioner matrix P will be a submatrix of S with zeros in place of the blocks EV, BV, BE, VE, VB, EB, with the block BB untouched, and with the blocks EE and VV reduced as described above. The structure of the matrix P is as follows:


0
0
0
0
0
0
0
0
0
0