(Multiple) Goal-Oriented Adaptivity with Multimesh *hp*-FEM

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

The purpose of this document is to suggest that we develop a new method for goal-oriented adaptivity based on the multimesh *hp*-FEM. The underlying idea is the old one proposed by Rannacher et al., but we should be able to compute more efficiently than people who use the same mesh for the primal and dual problems. I will start with the description of the dual problem and how it is typically used for goal-oriented adaptivity. I will avoid some theoretical aspects which are not necessary for implementation – more mathematical details can be found in the 2005 CMAME paper with Demkowicz available at http://hpfem.math.unr.edu/people/pavel/new_papers/2004_sode.pdf. You can also find it by going to my home page → Publications → Papers → CTRL+F "Goal". The dual problem is derived in Section 2.2, and the usual adaptivity algorithm (for single quantity of interest) is described in Section 3.1.

Quantities of interest

In practice, one is often interested in the accurate resolution of some particular *quantity of interest* rather than in the accurate convergence in global norm. Note – so far we only have been doing the latter in Hermes. The quantity of interest can be any linear form $q(u)$ of the solution u. Recall that a linear form is a linear functional that returns real numbers. We may be interested in the value of u at some point $\boldsymbol{x}_0$ in the computational domain Ω. Then,

$$q(u) = u(\boldsymbol{x}_0).$$

If we like to achieve precise resolution along a boundary part Γ, we will use a linear form

$$q(u) = \frac{1}{|\Gamma|} \int_\Gamma u(\boldsymbol{x}) \, \mathrm{d}S.$$

Or, we can be interested in resolving the solution accurately in some subdomain Ω_0. Then the corresponding quantity of interest would be

$$q(u) = \frac{1}{|\Omega_0|} \int_{\Omega_0} u(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}.$$

These are the most usual ones. But we can define many others that, for example, will correspond to the average of oscillatory solutions, flow in a given direction, one solution component in some subdomain, etc. The only limitation is that the quantity of interest must be linear in u.

Dual problem

Assume the original (primal) problem in the form: Find $u \in V$ such that

$$a(u, v) = l(v) \quad \text{for all } v \in V. \tag{1}$$

Here $a()$ is a bilinear form, so in the case of nonlinear problems, we have to use a linearization first. I will skip the derivation (see Paragraph 2.2.2 in the above mentioned paper). The dual problem reads: Find $v \in V$ such that

$$a(u, v) = q(u) \quad \text{for all } u \in V. \tag{2}$$

We see that (2) is fairly similar to (1). The only difference is that we have another right-hand side and that the arguments in the bilinear form $a()$ have different order (in the sense that now the first argument is the test function and the second one the solution). In other words, (2) can also be written as: Find $u \in V$ such that

$$a(v, u) = q(v) \quad \text{for all } v \in V.$$

To distinguish between the solution u to the primal problem and the solution v to the dual problem, in the following we will stay with the notation from (2).

On the discrete level, if

$$SY = F$$

is the matrix system for the primal problem (1), then the dual problem has the form

$$S^T \hat{Y} = Q.$$

In other words, one does not need to assemble a new stiffness matrix for the dual problem, just another right-hand side using $q()$ instead of $l()$.

Goal-oriented adaptivity

The basic error estimate which is used in the goal-oriented adaptivity is

$$|q(u) - q(u_{h,p})| \le C \sum_{i=1}^{M} \|u - u_{h,p})\|_{K_i} \|v - v_{h,p}\|_{K_i} \tag{3}$$

(again see the above mentioned paper for derivation). Here, C is some unknown constant, M is the number of elements, K_i is ith element in the mesh, and $\|w\|_{K_i}$ is the error of w in the element K_i measured in the global norm. Further, u is the unknown exact solution to the primal problem and $u_{h,p}$ its approximation, and v is the unknown exact solution to the dual problem and $v_{h,p}$ its approximation. In practical computations, as usual, we replace u with $u_{reference}$ and v with $v_{reference}$.

The estimate (3) will not affect the way we refine an element. This will be done exactly as before using projections of $u_{reference}$ on the candidates. However, what will change is the selection of elements for refinement. So far we computed $\|u_{reference} - u_{h,p}\|_{K_i}$ on each element, and sorted all elements according to their values. Instead, we will need to solve the dual problem on both the coarse and reference meshes, and calculate the products

$$\|u_{reference} - u_{h,p}\|_{K_i} \|v_{reference} - v_{h,p}\|_{K_i}. \tag{4}$$

Estimate (3) can be generalized easily to the case of multiple quantities of interest – one just needs to calculate all corresponding dual problems, and extend the product (4) to involve contributions from all of them. Let's assume that w is the solution to another dual problem corresponding to some additional quantity of interest $r(u)$. Then

$$|q(u) - q(u_{h,p})| + |r(u) - r(u_{h,p})| \le C \sum_{i=1}^{M} \|u - u_{h,p})\|_{K_i} \left(\|v - v_{h,p}\|_{K_i} + \|w - w_{h,p}\|_{K_i}\right) \tag{5}$$

and so on.

Standard algorithm

Typically, people start from a coarse mesh, compute the primal and dual solutions, and estimate the errors (in low-order FEM they do not need reference solution, but we need it so let's work with it). Then they sort elements according to the values (4) and mark some of them for refinement. Then they refine the mesh, and repeat the process. In particular, they solve all problems – the primal one and one or more dual ones – in every adaptivity step. However, typically the primal solution has large errors where the dual solutions are very accurate and vice versa. So, lots of degrees of freedom are wasted both ways.

New results that we can obtain

What we should do is to use multimesh *hp*-FEM to solve the primal and all dual problems one by one adaptively to a certain accuracy, starting from a common master mesh. This accuracy does not have to be very high since meshes obtained using goal-oriented adaptivity are much sparser compared to the adaptivity in global norm. This will give us all the reference solutions we need, and they would be defined on individual meshes. (And this could be done in parallel.)

Then we would start the goal-oriented adaptation process itself: In every step we would only need to solve on the coarse mesh, which could be (a) the same and (b) individual for the primal and all dual problems. This would not matter so much on the coarse mesh level since as stated above, meshes obtained using goal-oriented adaptivity are very small.

I think that we have in Hermes almost everything ready to do this. The only thing that is missing is enabling the products (4) for the selection of elements to be refined. This would be a fantastic paper.