Performance of *hp*-Adaptive Strategies for Elliptic Partial Differential Equations [1]

William F. Mitchell and Marjorie A. McClain

Applied and Computational Mathematics Division, NIST, Gathersburg, MD 20899

Keywords: adaptive mesh refinement, *hp*-adaptive strategy, *hp*-FEM
PACS: ???

We consider the use of *hp*-adaptive finite element methods for the solution of elliptic partial differential equations. As with *h*-adaptive refinement, local error indicators can be used to determine which elements should be refined, but they are not sufficient to also determine whether an element should be refined by h or by p. A method for making that determination is called an *hp*-adaptive strategy. A number of strategies have been proposed, but it is not clear which ones perform best under different situations, or even if any of the strategies are good enough to be used as a general purpose solver. In this paper we present a summary of an experimental comparison of several *hp*-adaptive strategies. Full details can be found in [**?**].

We consider the elliptic partial differential equation

$$-\mathrm{div}(A\nabla u) + r(x,y)u = f(x,y) \quad \text{in } \Omega \subset \Re^2$$

with Dirichlet, Neumann or mixed boundary conditions.

We solve the equation using the Galerkin finite element method with triangular elements. The basic form of the *hp*-adaptive algorithm is

begin with a very coarse grid
form and solve the linear system
repeat
  determine which elements to refine and whether to refine by h or p
  refine elements
  form and solve the linear system
until the global error estimate is below a given tolerance τ

For triangle *h*-refinement, the newest node bisection method is used. *p*-refinement means increasing the degree of the element by one, followed by enforcing the minimum rule for the edges. We use the usual *a posteriori* error indicator given by solving a local Neumann residual problem. The global error estimate is given by the square root of the sum of the squares of the error indicators.

Several *hp*-adaptive strategies have been proposed over the years. The strategies considered in this study are

- use of *a priori* knowledge of solution regularity (APRIORI), Ainsworth and Senior [**?**],
- type parameter (TYPEPARAM) Gui and Babuška [**?**],
- estimate regularity using larger p estimates (NEXT3P), Ainsworth and Senior [**?**],
- estimate regularity using smaller p estimates, (PRIOR2P), Süli, Houston and Schwab [**?**],
- Texas 3 step (T3S), Oden and Patra [**?**],
- alternate h and p (ALTERNATE), a variant on Texas 3 step,
- nonlinear programming (NLP), Patra and Gupta [**?**],
- predict error estimate on assumption of smoothness (SMOOTH_PRED), Melenk and Wohlmuth [**?**],

---

[1] **Preprint submitted to Proceedings of the 2014 International Conference on Numerical Analysis and Applied Mathematics**

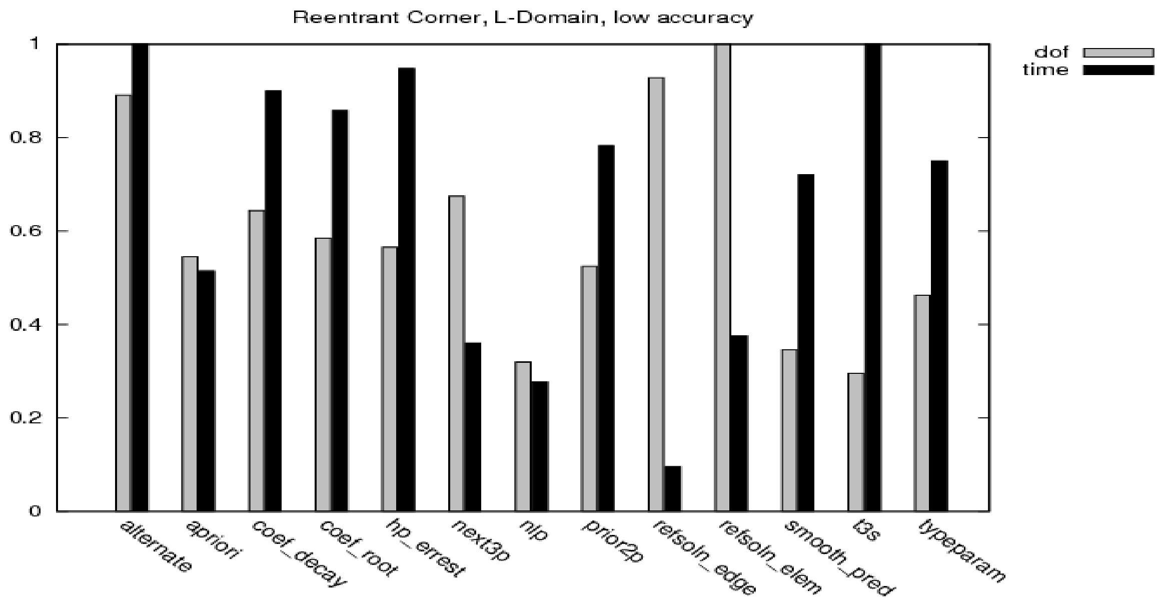

FIGURE 1. Relative performance of the strategies in degrees of freedom and wall clock time for low accuracy ($\tau = 10^{-2}$) solution of the L-shaped domain problem.

- larger of h-based and p-based error indicators (H&P_ERREST), Schmidt and Siebert [**?**],
- decay rate of Legendre coefficients (COEF_DECAY), Mavriplis [**?**],
- root test on Legendre coefficients (COEF_ROOT), Houston, Senior and Süli [**?**],
- edge-based reference solution (REFSOLN_EDGE), Demkowicz [**?**], and
- element-based reference solution (REFSOLN_ELEM), Šolín, Červený and Doležel [**?**].

Details of all these strategies can be found in [**?**].

A numerical experiment to compare the hp-adaptive strategies' performance was performed using a suite of 20 2D elliptic test problems with various difficulties that adaptive refinement should locate, including point singularities on the boundary, point singularities in the interior, boundary layers, steep gradients in the interior, and highly oscillatory solutions. The problems are classified as easy problems, hard problems, and problems with a singularity. The full details of the test problems can be found in [**?**].

Each problem is solved with each hp-adaptive strategy using the hp-adaptive algorithm given above. The problems are solved at low accuracy, typically $\tau = 10^{-2}$, and high accuracy, typically $\tau = 10^{-6}$. At the end of each run the number of degrees of freedom and total "wall clock" time to solution are recorded.

The full results of the experiment are given in [**?**] in bar charts as illustrated in Figure 1. The gray bars indicate the number of degrees of freedom required to reach the tolerance, and the black bars indicate the computation time required to reach the tolerance. All results are scaled by the value of the strategy that performed best, so, for example, a value of 1.0 indicates the best strategy, and a value of 0.2 indicates the strategy needed five times as many degrees of freedom or took five times longer than the best strategy.

Here we give a summary of the results. We consider a strategy to have been good for a particular problem if its degrees of freedom (or computation time) is within a factor of two of the strategy that performed best on that problem. For each category of problems in (low accuracy, hi accuracy)$\times$(easy, hard, singular), we count the number of problems for which each strategy did good. These numbers are presented in Tables 1 and 2.

We found that the REFSOLN_EDGE and REFSOLN_ELEM strategies performed best in degrees of freedom, and are comparable to each other. However, they are considerably more expensive than the other strategies, except NLP, in computation time. For problems with known point singularities and no other significant features, APRIORI appears to be the less expensive method of choice. COEF_DECAY appears to be the best choice as a general strategy across all categories of problems, whereas many of the other strategies perform well in particular categories and are reasonable

TABLE 1. Number of problems for which each strategy required less than twice as many degrees of freedom as the best performing strategy.

	low accuracy			high accuracy		
strategy	easy	hard	singular	easy	hard	singular
ALTERNATE	0	0	5	0	2	1
APRIORI	3	1	7	2	1	8
COEF_DECAY	1	0	7	0	0	2
COEF_ROOT	2	0	6	0	0	2
H&P_ERREST	1	0	4	0	0	0
NEXT3P	2	1	4	0	1	0
NLP	2	0	4	0	0	0
PRIOR2P	2	0	4	0	0	1
REFSOLN_EDGE	5	5	10	5	5	9
REFSOLN_ELEM	5	5	10	5	4	8
SMOOTH_PRED	0	0	0	0	0	1
T3S	1	0	1	0	2	0
TYPEPARAM	2	1	3	3	2	2

TABLE 2. Number of problems for which each strategy required less than twice as much computation time as the best performing strategy.

	low accuracy			high accuracy		
strategy	easy	hard	singular	easy	hard	singular
ALTERNATE	1	0	5	1	0	1
APRIORI	3	2	4	3	2	8
COEF_DECAY	5	2	8	3	4	6
COEF_ROOT	3	2	7	2	3	9
H&P_ERREST	3	2	8	2	4	5
NEXT3P	0	0	0	1	0	0
NLP	0	0	0	0	0	0
PRIOR2P	3	2	8	0	1	7
REFSOLN_EDGE	0	0	0	0	0	1
REFSOLN_ELEM	0	0	0	1	1	5
SMOOTH_PRED	2	4	6	2	4	3
T3S	1	3	7	3	4	3
TYPEPARAM	3	2	7	4	0	1

in general.