Three Anisotropic Benchmark Problems for Adaptive Finite Element Methods

Pavel Solin[a,b], Ondrej Certik[a,c], Lukas Korous[d]

[a] *Department of Mathematics and Statistics, University of Nevada, Reno, USA*
[b] *Institute of Thermomechanics, Academy of Sciences of the Czech Republic, Prague*
[c] *Institute of Physics, Academy of Sciences of the Czech Republic, Prague*
[d] *Charles University, Prague, Czech Republic*

Abstract

In this paper we provide three benchmark problems with known exact solutions that can be used to assess the ability of adaptive finite element algorithms to handle anisotropically-behaved solutions. The first one is a Poisson equation with a smooth solution that only changes in one spatial direction. The second one is a singularly-perturbed linear elliptic equation whose solution exhibits a boundary layer, and the last one is a two-equation system that contains a boundary layer in one solution component only. In an appendix we show sample results obtained with the open source library HERMES.[1]

Keywords: Benchmark problem, Anisotropic solution, Boundary layer, Finite element method

1. Introduction

The number of adaptive finite element libraries is growing – let us mention (in alphabetical order) for example Alberta [1], DealII [2], FEniCS [3], FETK [4], Hermes [5], libMesh [6], Phaml [7], PHG [8], 2dhp90 [9] and there are many others. A natural question that arises is how do they compare to each other? Unfortunately, comparison efforts are usually inhibited at the very beginning by diverse installation requirements, supporting libraries, input and output data formats, and different usage of various codes. And even if

Email addresses: solin@unr.edu (Pavel Solin), ondrej@certik.cz (Ondrej Certik), lukas.korous@gmail.com (Lukas Korous)

[1] http://hpfem.org/hermes

these problems could be overcome, not many benchmarks with known exact solutions are available to test various aspects of automatic adaptivity.

At this point we would like to acknowledge the pioneering work of Dr. William Mitchell (NIST) who not only collected a suite of 12 benchmark problems for adaptive FEM [3], but who also implemented and compared several *hp*-adaptive finite element algorithms by various authors [4].

This paper presents three benchmark problems with anisotropically-behaved solutions (not contained in [3]) that are designed to test the ability of adaptive algorithms to handle anisotropic refinements. The benchmark problems and their solutions are formulated in Sections 2 - 4. In a separate second part of the paper (Appendix A) we show for illustration sample results obtained with the open source HERMES library (http://hpfem.org/hermes).

2. Benchmark No. 1 "Beginner"

The first benchmark problem is a Poisson equation

$$-\Delta u = \sin(x) \tag{1}$$

in the domain $\Omega = (0, \pi)^2$, equipped with a zero Dirichlet boundary condition on the left edge, zero Neumann boundary conditions on the top and bottom edges, and a Neumann condition $\partial u/\partial \nu = -1$ on the right edge. Here ν is the unit outer normal vector to the boundary. The exact solution $u(x, y) = \sin(x)$ to this problem is shown in Fig. 1.

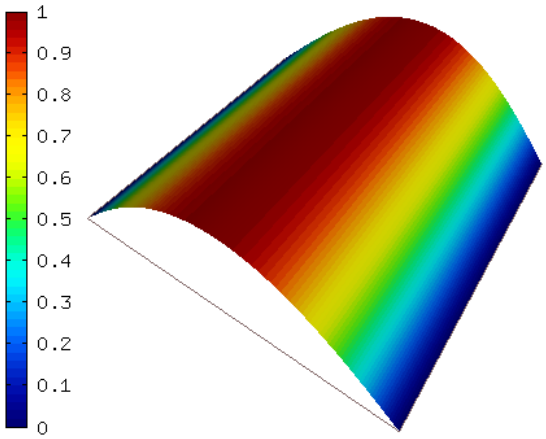

Figure 1: Exact solution to the first benchmark problem.

The goal of the benchmark is to attain a relative error below 10^{-4} % in the H^1-norm with as few degrees of freedom (DOF) as possible. Using 1D analysis [8], one can show that the minimum number of DOF needed is 16.

3. Benchmark No. 2 "Intermediate"

Next we consider a singularly perturbed elliptic equation

$$-\Delta u + k^2 u - k^2 = g \tag{2}$$

where k is a real constant. Here we use the value $k = 100$ and the problem can be made harder when k is increased. Equation (2) is equipped with homogeneous Dirichlet boundary conditions and solved in the domain $\Omega = (-1, 1)^2$. For convergence studies we use a manufactured exact solution

$$v(x, y) = \hat{u}(x)\hat{u}(y), \tag{3}$$

where

$$\hat{u}(x) = 1 - \frac{e^{kx} + e^{-kx}}{e^k + e^{-k}} \tag{4}$$

is the exact solution to a similar one-dimensional problem $-\Delta u + k^2 u - k^2 = 0$ in the interval $(-1, 1)$, equipped with homogeneous Dirichlet boundary conditions. The right-hand side g is calculated by inserting (3) into (2).

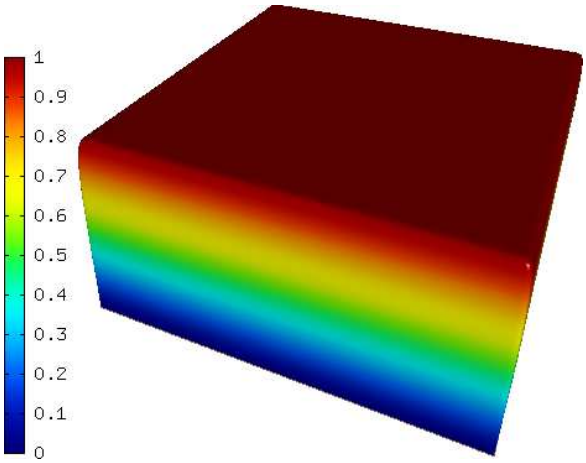

Figure 2: Exact solution to the second benchmark problem for $k = 100$.

As in the previous section, also here the goal is to attain a relative error less or equal to 10^{-4} % in the H^1-norm with as few degrees of freedom (DOF) as possible.

4. Benchnark No. 3 "Advanced"

The Fitzhugh-Nagumo equation [2, 5],

$$\begin{aligned} -d_u^2 \Delta u - f(u) + \sigma v &= g_1, \\ -d_v^2 \Delta v - u + v &= g_2, \end{aligned} \tag{5}$$

is a prominent example of an activator-inhibitor system in two-component reaction-diffusion equations. The unknowns u, v represent the voltage and v-gate, respectively, and the nonlinear function

$$f(u) = \lambda u - u^3 - \kappa \tag{6}$$

describes how an action potential travels through a nerve. Since the focus of this benchmark is on adaptivity rather than on nonlinear solver capabilities, we can replace (6) with a simpler function

$$f(u) = u \tag{7}$$

which makes the problem linear. In (5), d_u, d_v, σ are positive constants, and $g_1(x, y)$ and $g_2(x, y)$ are spatially-dependent source terms. Both u and v are zero on the boundary of our domain of interest which is a square $\Omega = (-1, 1)^2$.

For convergence studies we define an exact solution pair $u(x, y)$ and $v(x, y)$, where

$$u(x, y) = \cos\left(\frac{\pi}{2}x\right) \cos\left(\frac{\pi}{2}y\right)$$

and $v(x, y)$ is taken from (3),

$$v(x, y) = \hat{u}(x)\hat{u}(y).$$

Both u and v are inserted into (5) in order to calculate the corresponding source functions g_1 and g_2. The exact solution pair u, v is shown in Figs. 3 and 4.

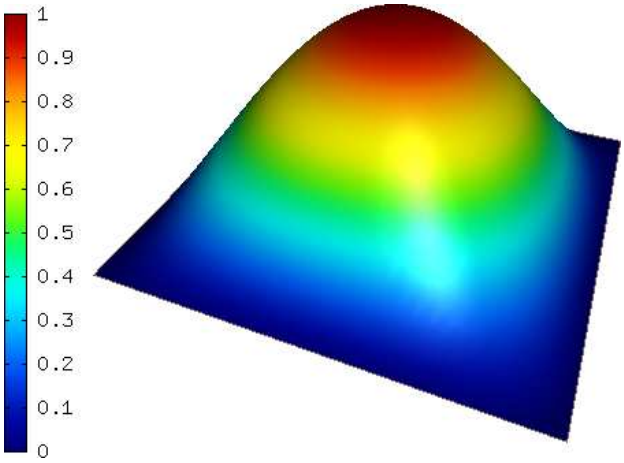

Figure 3: Exact solution to the third benchmark problem (first component u).

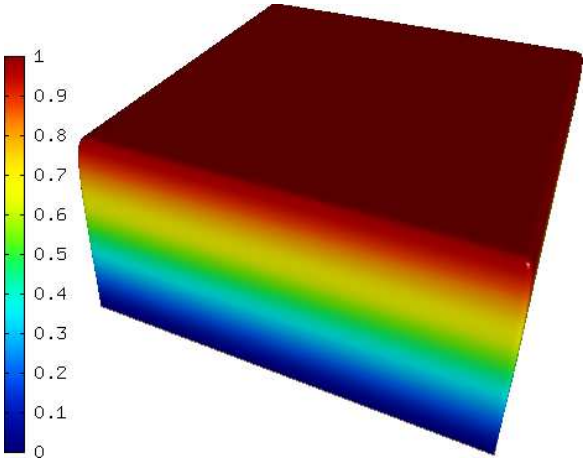

Figure 4: Exact solution to the third benchmark problem (second component v).

The goal of this benchmark is to attain a relative error less or equal to 10^{-2} % in the product H^1-norm (square root of the sum of component norms squared) with as few degrees of freedom (DOF) as possible.

5. Conclusion and Outlook

We have presented three new benchmark problems with known exact solutions that can be used to assess the ability of adaptive FEM codes to handle

anisotropic refinements. Since for adaptive FEM codes the resulting mesh should be observed as part of the final solution, we are currently expanding our research of optimal meshes.

6. Acknowledgment

This work was supported by Subcontract No. 00089911 of Battelle Energy Alliance (DOE intermediary) as well as by the Grant No. IAA100760702 of the Grant Agency of the Academy of Sciences of the Czech Republic. The second autor was partly supported by the LC06040 research center project in the Czech Republic.

Appendix A. Sample Results Obtained with HERMES

In the second part of this paper we provide sample solutions to the above benchmark problems using the HERMES library. The results are provided for illustration purposes only and they should not be used as reference values.

Brief Description of the HERMES *Library*

HERMES is an open source C++ library for rapid development of adaptive *hp*-FEM and *hp*-DG solvers. The library is developed by the *hp*-FEM group at the University of Nevada, Reno (http://hpfem.org) and it is available under the GPL license.

HERMES offers eight different modes of automatic adaptivity:

1. H_ISO (*h*-adaptivity with isotropic refinements),
2. H_ANISO (*h*-adaptivity with possibly anisotropic refinements),
3. P_ISO (*p*-adaptivity with isotropic polynomial degrees in elements),
4. P_ANISO (*p*-adaptivity with possibly anisotropic polynomial degrees in elements),
5. HP_ISO (*hp*-adaptivity with isotropic polynomial degrees in elements and isotropic spatial refinements),
6. HP_ANISO_H (*hp*-adaptivity with isotropic polynomial degrees in elements and possibly anisotropic spatial refinements),
7. HP_ANISO_P (*hp*-adaptivity with possibly anisotropic polynomial degrees in elements and isotropic spatial refinements),
8. HP_ANISO (*hp*-adaptivity with possibly anisotropic polynomial degrees in elements and possibly anisotropic spatial refinements).

Main Idea of the Adaptivity Algorithm

In each adaptivity step the algorithm performs a global *hp* mesh refinement, calculates an approximate (reference) solution u_{ref} on the refined mesh, obtains its closest representant u_{coarse} on the original coarse mesh via an orthogonal projection, and evaluates the approximate error function

$$e_{h,p} = u_{ref} - u_{coarse}.$$

In fact, u_{coarse} is the low-order part of u_{ref} and so $e_{h,p}$ is an excellent indicator where u_{coarse} needs to be improved. Next, the algorithm identifies elements with the largest error norm, and selects them for refinement. Last, in a loop over all selected elements K, the algorithm projects u_{ref} locally on all refinement candidates on K. Here is where the various adaptivity modes differ – the simplest one is H_ISO with just one refinement option (a 4-split) and mode 8 is the most general with around 100 options. This looks like lots of CPU time, but in reality the projections are extremely fast since each refinement candidate comes with a local orthonormal basis. Among all considered refinements of K, the algorithm select the one which yields the smallest projection error (weighted with the number of new degrees of freedom that the candidate adds to the discrete problem).

The algorithm is completely PDE-independent, it does not contain any tuning parameters, and in contrast to various popular gradient-based techniques it works equally well for low-order and high-order elements. Thanks to its robustness and simplicity, the algorithm works very well for multiphysics PDE systems where it is used in combination with multimesh *hp*-FEM [7].

Benchmark No. 1

We begin with adaptive *hp*-FEM with anisotropic refinements (adaptivity mode HP_ANISO in HERMES), starting from a mesh that contains only one bilinear element. The initial mesh is shown in Fig. A.5 (left). In a few adaptivity steps, the polynomial degree of this element is increased anisotropically and the element is never refined in space. The resulting single-element mesh with 16 DOF is shown in Fig. A.5 (right). The diagonal pattern indicates different polynomial degrees (8 in the x-direction and 1 in the y-direction).

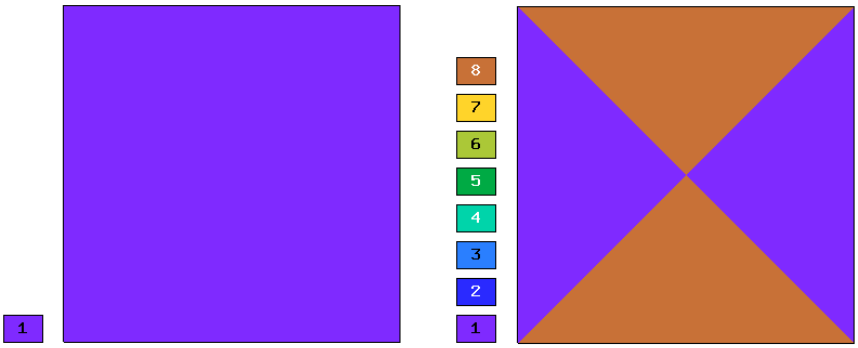

Figure A.5: Initial mesh (left) and final mesh (right) for *hp*-FEM with anisotropic refinements.

The final relative error estimate in H^1-norm was 3.6796e-05 % and it was identical to the exact error in all printed digits.

We also solved this benchmark with adaptive *h*-FEM with quadratic elements, with anisotropic refinements enabled and disabled (adaptivity modes H_ANISO and H_ISO in HERMES, respectively). Final meshes for the *h*-FEM computations are not shown here – they were so fine that they appeared as black squares. The *h*-FEM with isotropic refinements was stopped after reaching 100,000 DOF. Figs. A.6 and A.7 compare all three adaptivity modes from the point of view of DOF and CPU convergence.

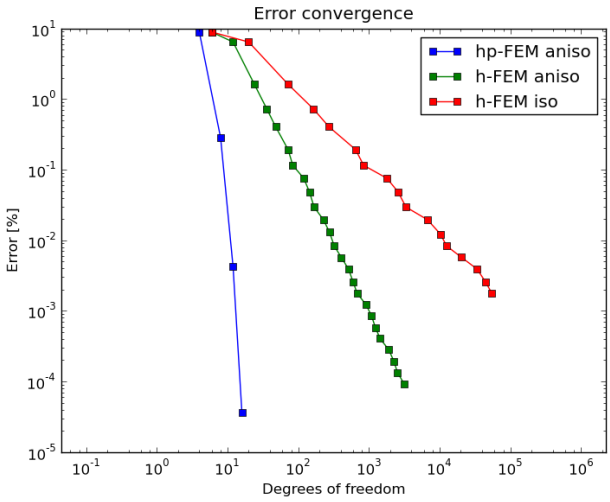

Figure A.6: DOF convergence graphs.

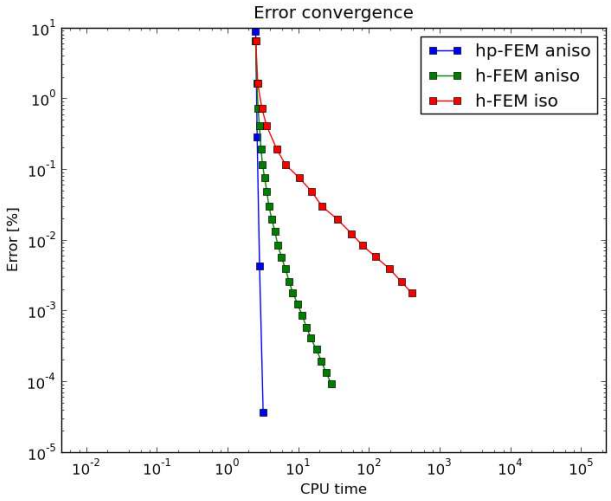

Figure A.7: CPU time convergence graphs.

Benchmark No. 2

First we employ *hp*-FEM with anisotropic refinements (HP_ANISO mode in HERMES) and start from a 100-element mesh that was obtained through several consecutive refinements towards the boundary. Note that herewith we use some a-priori knowledge of the solution. The initial mesh and the final mesh with 3,913 DOF are shown in the left and right part of Fig. A.8, respectively.

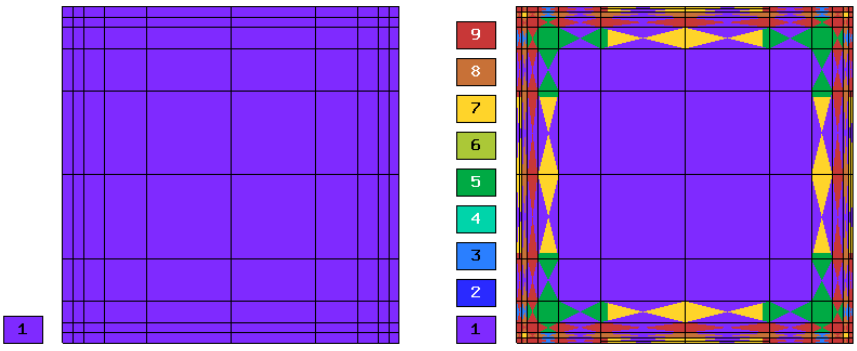

Figure A.8: Initial mesh (left) and final mesh (right) for *hp*-FEM with anisotropic refinements.

We also solved this benchmark using adaptive h-FEM with quadratic elements, with anisotropic refinements enabled and disabled (adaptivity modes H_ANISO and H_ISO in HERMES, respectively). The corresponding final meshes are shown in Fig. A.9.

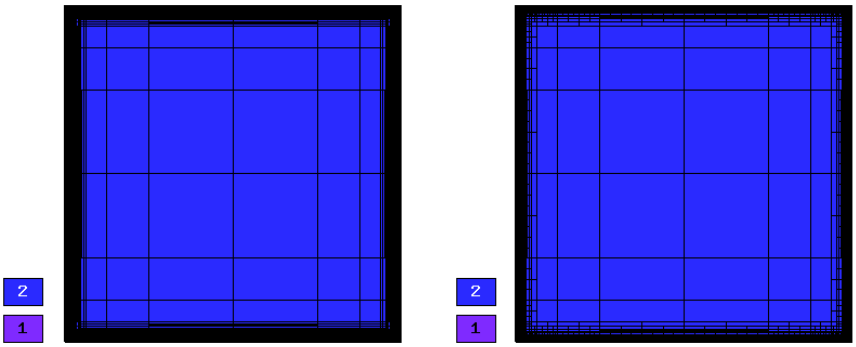

Figure A.9: Resulting meshes for h-FEM with quadratic elements. Anisotropic refinements enabled (left) and disabled (right).

Again, computations were stopped when the number of DOF exceeded 100,000. Figs. A.10 and A.11 compare all three adaptivity modes from the point of view of DOF and CPU time convergence.

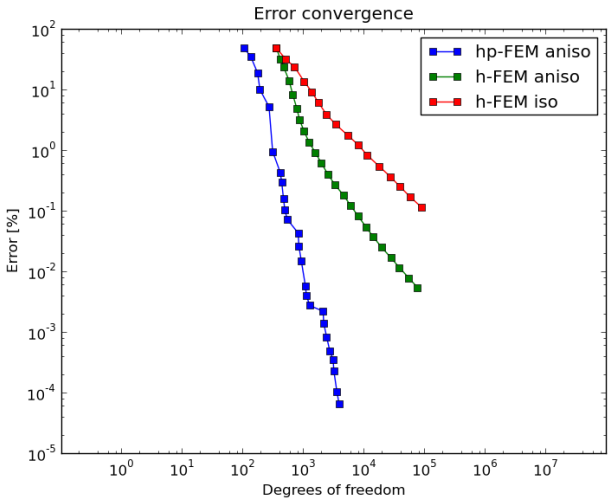

Figure A.10: DOF convergence graphs.

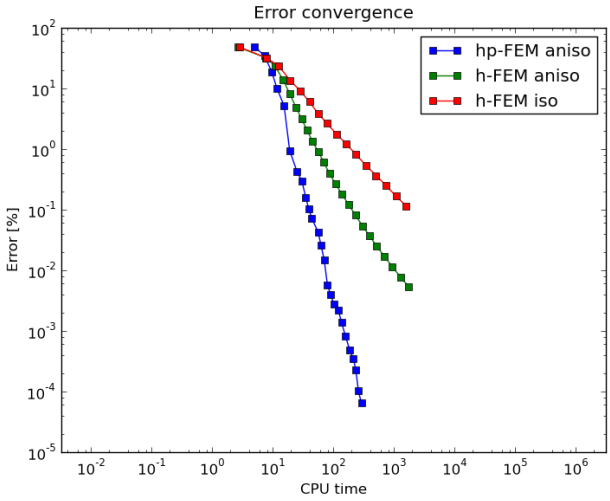

Figure A.11: CPU time convergence graphs.

Benchmark No. 3

As in the previous two cases, we begin with *hp*-FEM with anisotropic refinements (HP_ANISO mode in HERMES). The initial mesh for the first solution component is just one quadratic element and the initial mesh for the second component has 100 quadratic elements, as shown in Fig. A.12

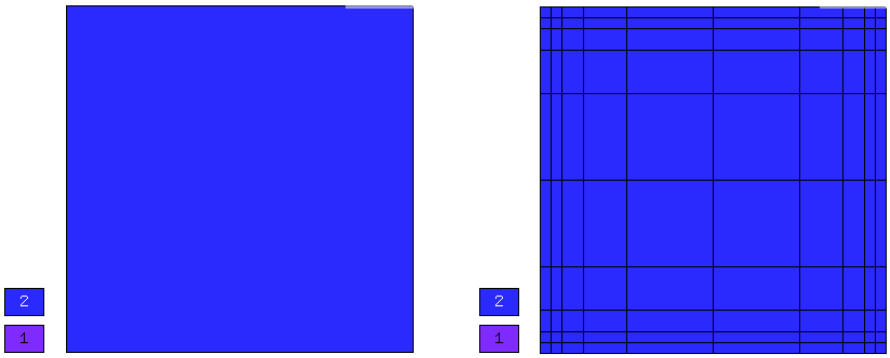

Figure A.12: Initial meshes for the two solution components.

The final meshes for the first and second solution components are shown in Fig. A.13. They contain 49 and 1,809 DOF, respectively (total of 1,858 DOF).

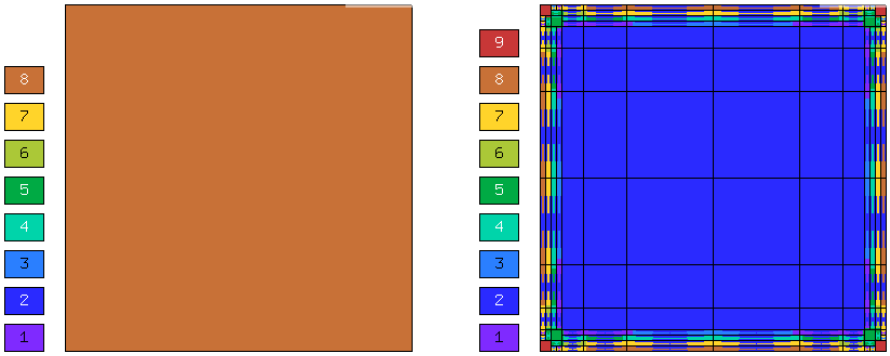

Figure A.13: Final meshes for both solution components, *hp*-FEM with anisotropic refinements.

Again we also solved this benchmark using adaptive *h*-FEM with quadratic elements, with anisotropic refinements enabled and disabled (adaptivity modes H_ANISO and H_ISO in HERMES, respectively). The corresponding final meshes for the anisotropic case are shown in Fig. A.14. They contain 4,961 DOF and 49,873 DOF (total of 54,834 DOF).

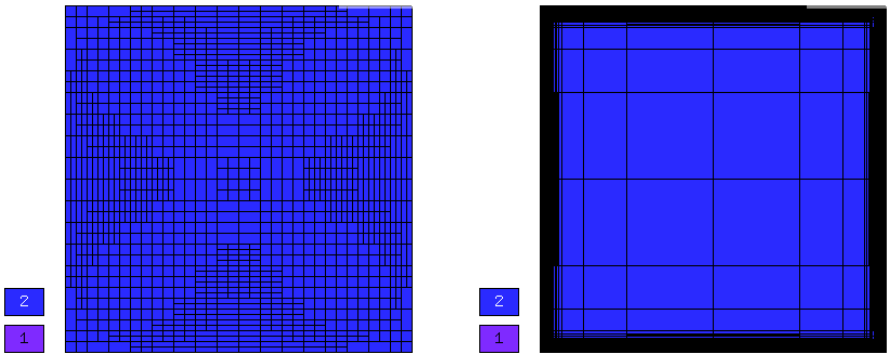

Figure A.14: Final meshes for both solution components, *h*-FEM with quadratic elements and anisotropic refinements enabled.

The final meshes for the isotropic case are shown in Fig. A.15. They contain 961 DOF and 81,185, DOF (total of 82,146, DOF).

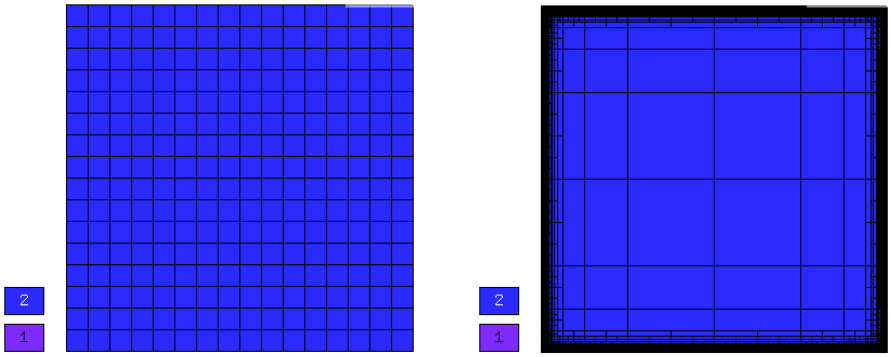

Figure A.15: Final meshes for both solution components, h-FEM with quadratic elements and isotropic refinements.

Also in this case computations were stopped when the number of DOF exceeded 100,000. Figs. A.16 and A.17 compare all three adaptivity modes from the point of view of DOF and CPU time convergence.

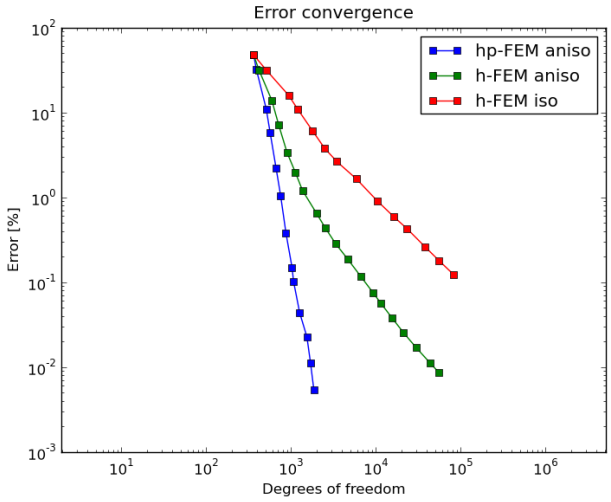

Figure A.16: DOF convergence graphs.

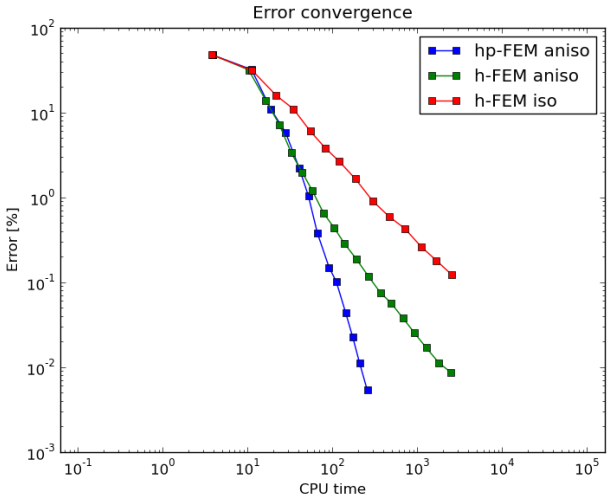

Figure A.17: CPU time convergence graphs.

References

[1] D. Estep, G. Carey, V. Ginting, S. Tavener, T. Wildey: A Posteriori Error Analysis of Multiscale Operator Decomposition Methods for Multiphysics Models, SciDAC 2008, Journal of Physics: Conference Series 125 (2008) 12 - 75.

[2] R. Fitzhugh: Mathematical Models of Excitation and Propagation in Nerve. Chapter 1, pp. 1-85 in H.P. Schwan, ed. Biological Engineering, McGraw-Hill Book Co., N.Y. (1969).

[3] W. Mitchell: A Collection of 2D Elliptic Problems for Testing Adaptive Algorithms, NISTIR 7668, 2010 (available online).

[4] W. Mitchell: A Survey of hp-Adaptive Strategies for Elliptic Partial Differential Equations, Annals of the European Academy of Sciences, to appear (available online).

[5] J. Nagumo, S. Arimoto, S. Yoshizawa: An Active Pulse Transmission Line Simulating Nerve Axon. Proc. IRE 50, 20612070 (1962).

[6] P. Solin, D. Andrs, J. Cerveny, M. Simko: PDE-Independent Adaptive *hp*-FEM Based on Hierarchic Extension of Finite Element Spaces. J. Comput. Appl. Math. 233 (2010) 3086-3094.

[7] P. Solin, J. Cerveny, L. Dubcova, D. Andrs: Monolithic Discretization of Linear Thermoelasticity Problems via Adaptive Multimesh *hp*-FEM, J. Comput. Appl. Math 234 (2010) 2350 - 2357.

[8] P. Solin. K. Segeth, I. Dolezel: Higher-Order Finite Element Methods, Chapman & Hall / CRC Press, Boca Raton, 2003.

We also include online references to the FEM codes mentioned in this paper. All these URLs were active on September 30, 2010:

References

[1] Alberta: http://www.alberta-fem.de/

[2] DealII (Differential Equations Analysis Library) http://www.dealii.org/

[3] FEniCS: http://www.fenics.org/wiki/FEniCS_Project

[4] FETK (Finite Element Toolkit): http://www.fetk.org/

[5] Hermes (Higher-Order Finite Element System) http://hpfem.org/hermes.

[6] libMesh: http://libmesh.sourceforge.net/

[7] Phaml (Parallel Hierarchical Adaptive MultiLevel Project): http://math.nist.gov/phaml/

[8] PHG (Parallel Hierarchical Grid) http://lsec.cc.ac.cn/phg/

[9] 2dhp90: http://users.ices.utexas.edu/~leszek/2dhp90.html