ADVANCED ADAPTIVE ALGORITHMS IN FINITE ELEMENT METHOD OF HIGHER ORDER OF ACCURACY

Ivo Doležel, Pavel Karban, František Mach, Bohuš Ulrych

CZECH UNIVERSITY OF TECHNOLOGY IN PRAGUE, CZECH REPUBLIC

UNIVERSITY OF WEST BOHEMIA IN PILSEN, CZECH REPUBLIC

ISTET'11

Sixteenth International Symposium on Theoretical Electrical Engineering -- ISTET'11
July 25-27, 2011
University of Klagenfurt, Klagenfurt, Austria

Outline

1. Introduction
2. Elements of adaptivity
3. Codes Hermes and Agros
4. Illustrative examples
5. Conclusion

1. Introduction

All advanced techniques for numerical solution of physical fields contain special algorithms for automatic adaptivity of discretization meshes. Their purpose is to reduce the error of solution to the lowest possible level and they are applied at the moment when local errors of solution are higher than the acceptable tolerance. These errors defined as the differences between the current numerical solution and exact solution are usually caused by

- **locally rougher or inappropriately structured mesh,**
- **presence of singular points,**
- **curvilinear boundaries or interfaces approximated by polygonal lines, etc.**

In all these cases, such errors must be identified in the course of computation and appropriate measures have to be taken for their fixing.

2. Elements of adaptivity

Consider an equation

$$Lf = 0$$

where L is a differential operator and f a function whose distribution over some domain Ω is to be found. If f' is its approximation obtained by numerical solution of the above equation, the absolute and percentage relative errors δ and η are defined by the relations

$$\delta = f - f', \qquad \eta = 100\left|\delta / f\right|.$$

Other quantities that can be checked in this way are the norms. Usually, one works with the following norms

$$\|e\| = \left|\int_{\Omega} \delta\left(L\delta\right)\mathrm{d}\Omega\right|^{1/2},$$

$$\|e\|_{L^2} = \left|\int_{\Omega} \delta^2 \mathrm{d}\Omega\right|^{1/2},$$

$$\|e\|_{H^1} = \left|\int_{\Omega}\left(\delta^2 + \left(\operatorname{grad}\delta\right)\cdot\left(\operatorname{grad}\delta\right)\right)\mathrm{d}\Omega\right|^{1/2}.$$

Unfortunately, the exact solution f is known only in very simple analytically solvable cases. Moreover, there exists no general and universal method that would provide a good estimation of the error for an arbitrary partial differential equation.

Local errors of the solution can be estimated from higher derivatives of the investigated quantity in the elements, but this approach is complicated and not very practical. That is why it is better to work with a reference solution f_{ref} instead, which is a more accurate solution than the current solution f. Usually it is a solution obtained on a finer mesh. An obvious disadvantage of this approach to automatic adaptivity is its higher computational cost.

In the next paragraphs I will show the elements of adaptivity that are implemented in our own codes Hermes and Agros. I will focus attention on the 2D versions of both codes (where these elements are on a very high level). Our team intensively works on adaptivity in 3D, but the development and implementation of the corresponding techniques will take several more years of intensive work.

3. Codes Hermes and Agros

More than 8 years our group has been developing our own code for numerical solution of sets of second-order partial differential equations of general types. This code written mostly in C++ consists of two parts: Hermes and Agros.

Hermes is a library of most advanced numerical algorithms for monolithic and full adaptive solution of systems of generally nonlinear and evolutionary partial differential equations based on the finite element method of higher order of accuracy.

http://hpfem.org/hermes/

Agros is a powerful user interface serving for preprocessing and postprocessing of the problems solved.

http://agros2d.org/

Both these codes are freely distributable under the GNU General Public License. And both of them use the most advanced adaptivity techniques.

The algorithms of automatic adaptivity implemented in 2D Hermes and Agros are divided into the following groups:

- ***h*-adaptivity, *p*-adaptivity, *hp*-adaptivity, handling with hanging nodes,**
- **work with curved elements,**
- **combination of triangular and quadrilateral elements (quadrilateral elements are advantageous, for example, in subdomains with anisotropy).**

The reference solution is understood as a solution calculated on a more than twice finer mesh. Hermes and Agros work not only with the maximum local error over the element, but also with its distribution over it.

hp-adaptivity

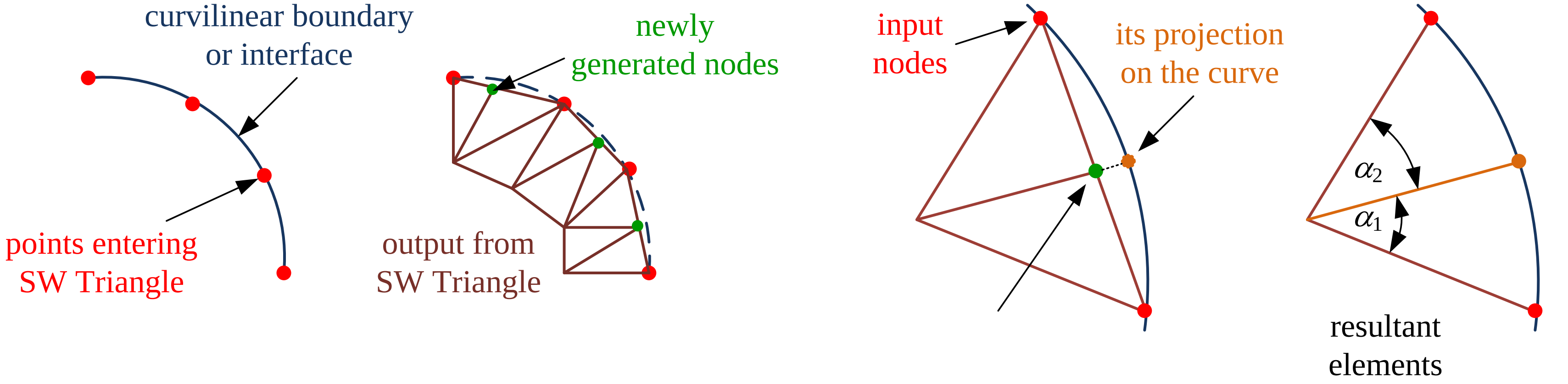
Hermes2D works with elements up to the tenth order. Each physical field can be solved on quite a different mesh that best corresponds to its particulars. Special powerful higher-order techniques of mapping are then used to avoid any numerical errors in the process of assembly of the stiffness matrix.

In evolutionary processes every mesh can change in time, in accordance with the real evolution of the corresponding physical quantities.

There are no problems with the hanging nodes appearing along the boundaries of subdomains whose elements have to be refined. Usually the hanging nodes bring about a considerable increase of the number of the degrees of freedom (DOFs). The code contains higher-order techniques for respecting these nodes without any need of an additional refinement of the external parts neighboring with the refined subdomain.

Curved elements

Agros2D discretizes 2D domains using SW Triangle. The corresponding input data for modeling curvilinear boundaries or interfaces in Triangle are given by a series of points lying on this line (together with the markers carrying information that these points belong to such a line) while the output is represented by a set of triangular elements. In the second step Agros2D repeats analyzing the curved lines and when any of the newly generated nodes approximating the curve does not lie on it, it is automatically projected on the original arc. At the same time a special procedure determines the corresponding angle α.

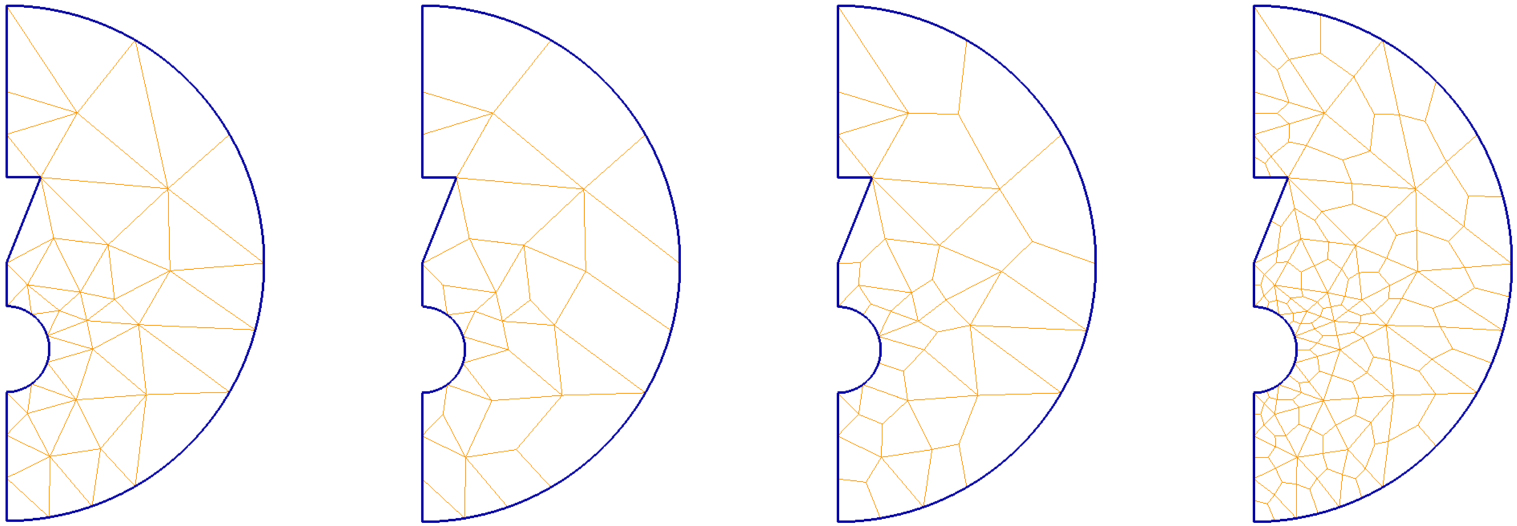


Combination of triangles and quadrilaterals

This combination is used in the following two cases:

- existence of anisotropic domains (material anisotropy or field anisotropy) in the definition area of the problem or
- domains with smooth distribution of the investigated field quantity.

In both cases, application of this technique may significantly reduce the number of DOFs.



4. Illustrative examples

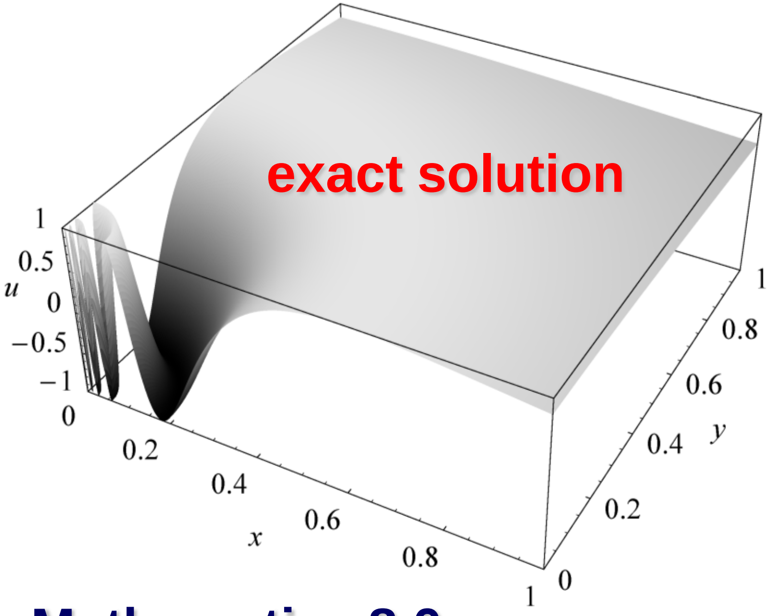
The first example is inspired by the solution of the Schrödinger equation describing the interaction between two atoms. It can be found in a benchmark example collection F. M. Mitchell, “A Collection of 2D Elliptic Problems for Testing Adaptive Algorithms, ”*NISTIR* 7668, 2010, that serves for the comparison of capabilities of various existing codes. The equation in the form

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right)+\frac{u}{\left(r+\alpha\right)^{4}}=\frac{r-\alpha}{r\left(r+\alpha\right)^{3}}\cos\left(\frac{1}{r+\alpha}\right), \qquad r=\sqrt{x^{2}+y^{2}}$$

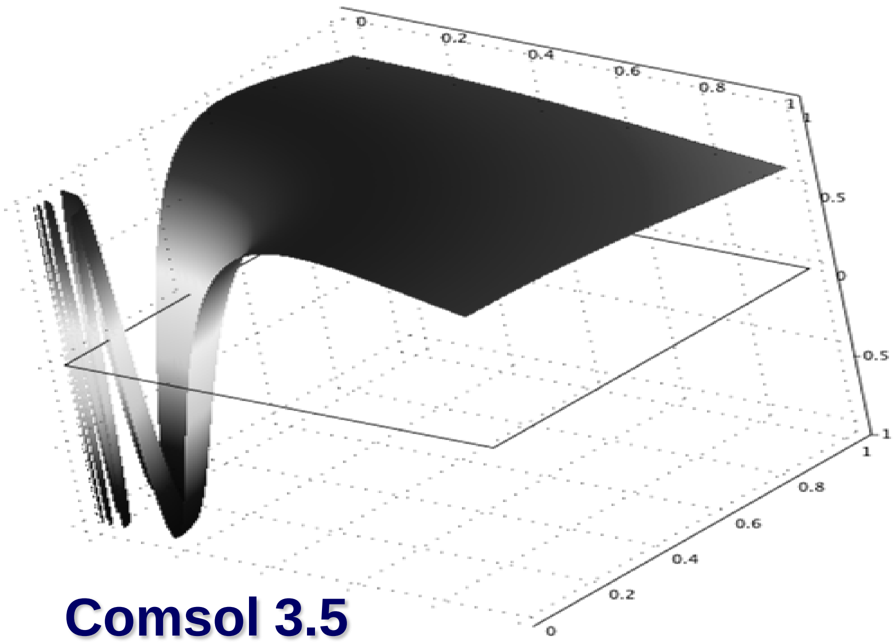
is solved on a unit square, whose low left corner is at the origin of the Cartesian coordinate system. Its particular analytical solution is

$$u=\sin\left(\frac{1}{\alpha+r}\right)$$

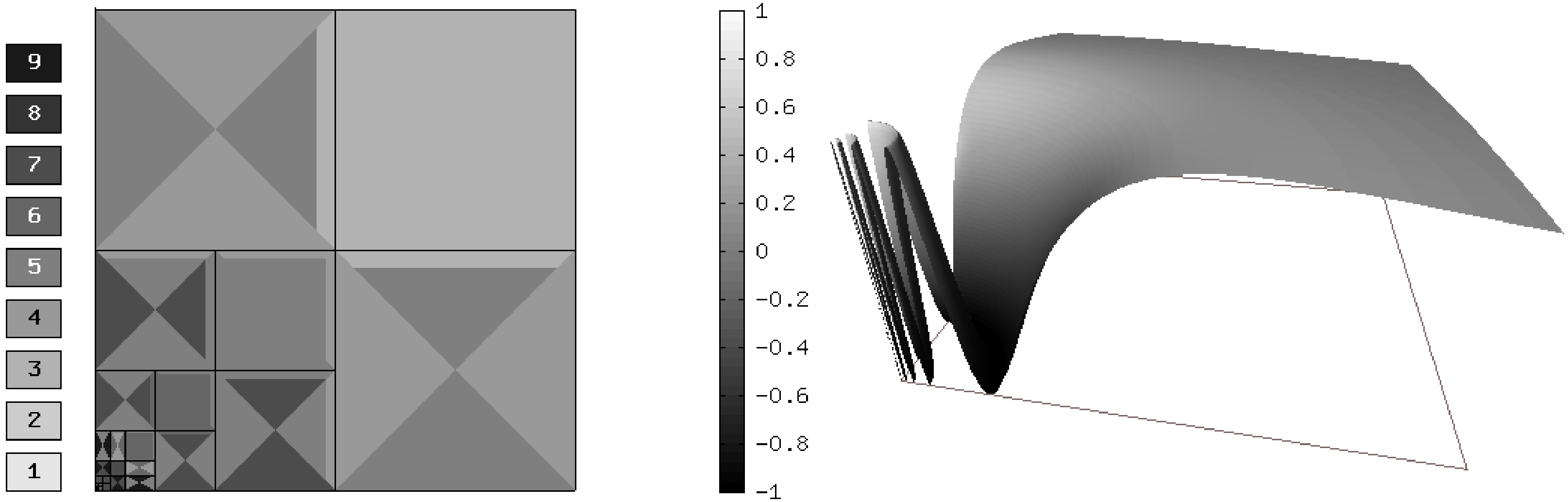
and we shall consider the corresponding boundary conditions along the circumference of the square. The solution oscillates near the origin and the coefficient α is inversely proportional to the number of oscillations. For $\alpha=1/(10\pi)$ we obtain ten oscillations.



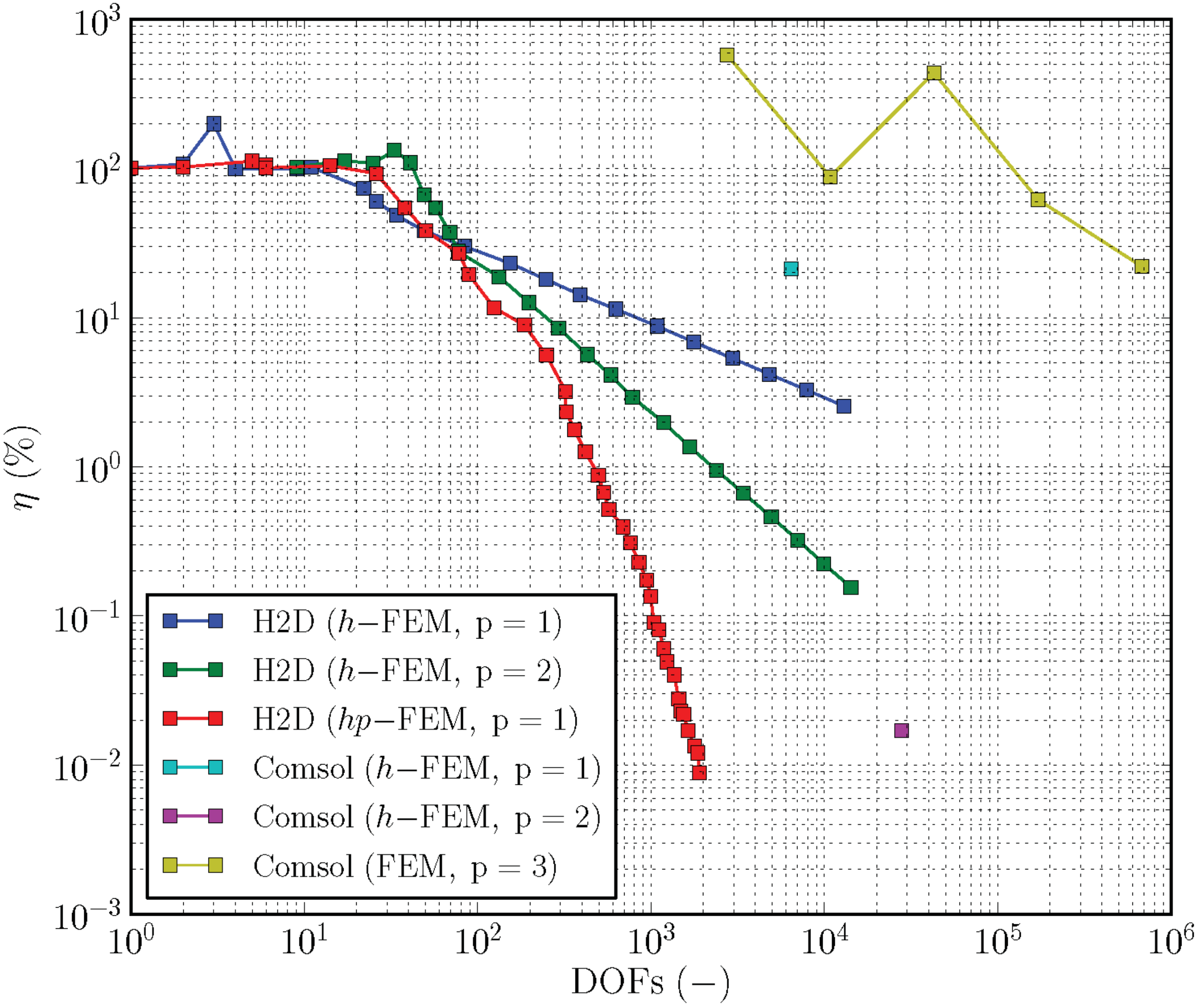
Mathematica 8.0



Comsol 3.5

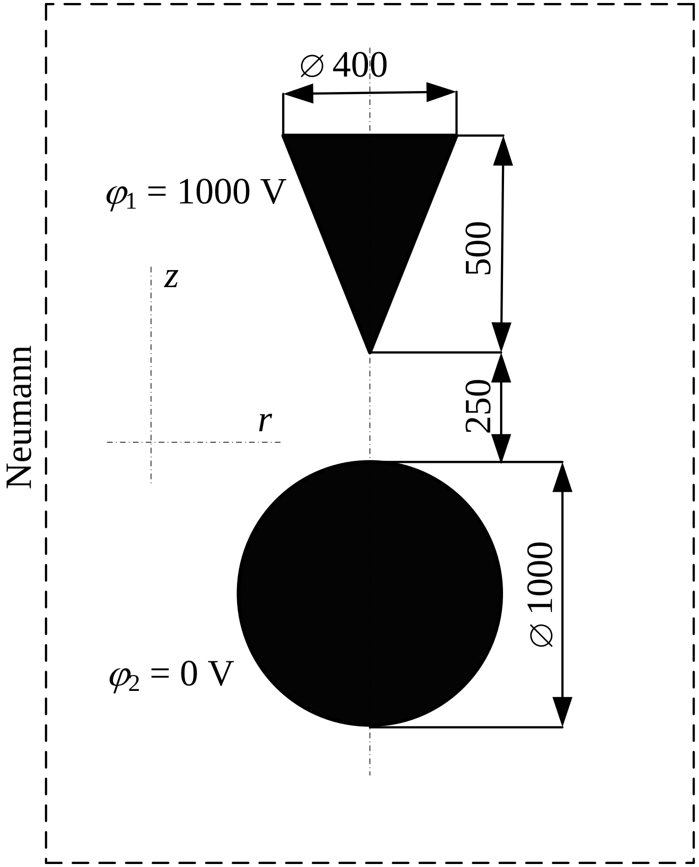


Hermes2D + Agros2D

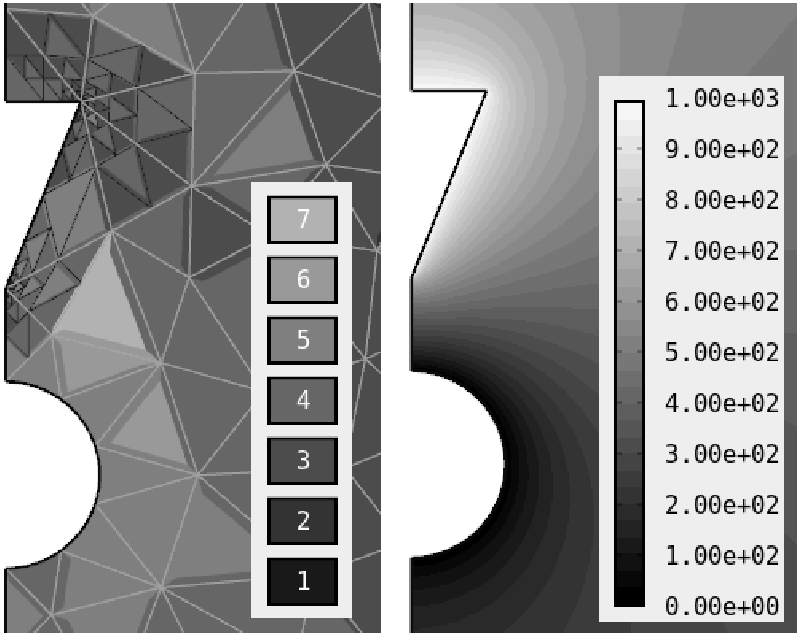


Convergence rate

The second example is more universal and shows application of the curvilinear elements, handling with a singular point, and combination of triangular and quadrilateral elements. The singular point means a point at which it is not possible to define the normal and the gradient of the solution grows there over all limits.

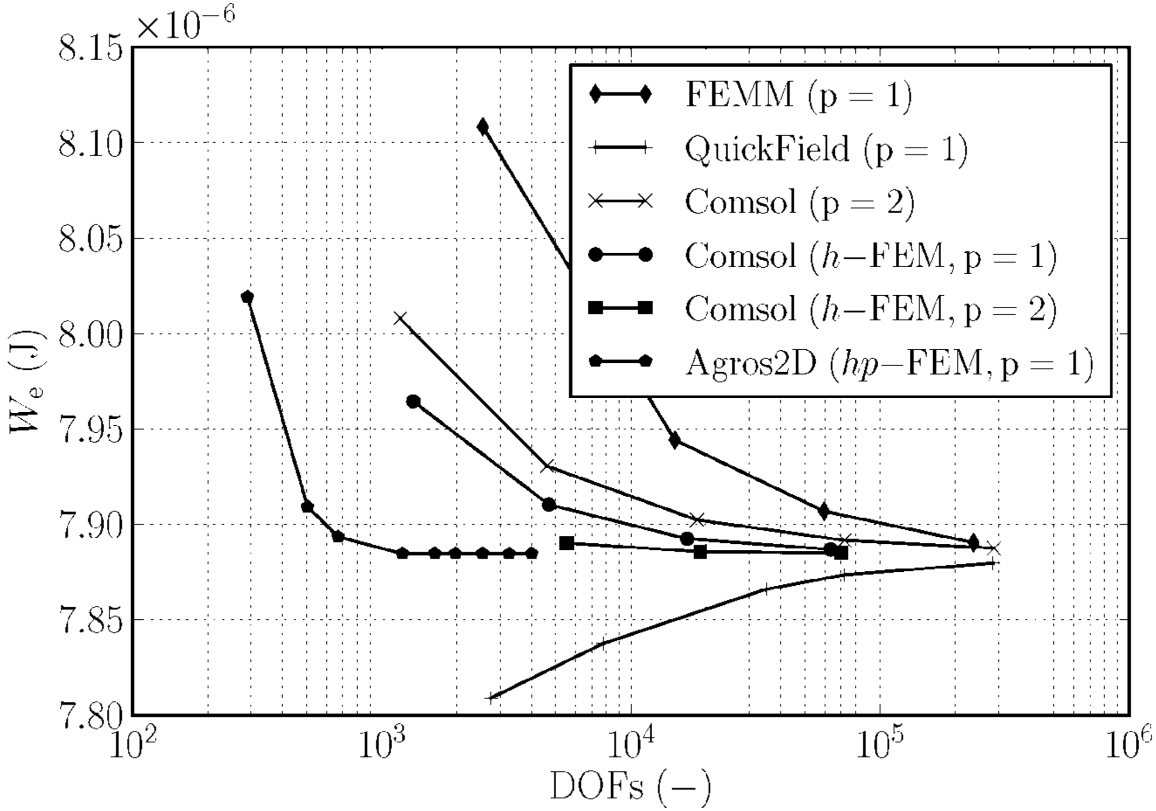


Axisymmetric spark gap (dimensions in mm)



Original rougher mesh (white lines) and final mesh after adaptivity (dark lines). Calculation using Agros2D, number of DOFs 1977, relative error of solution $\eta = 0.307\ \%$.

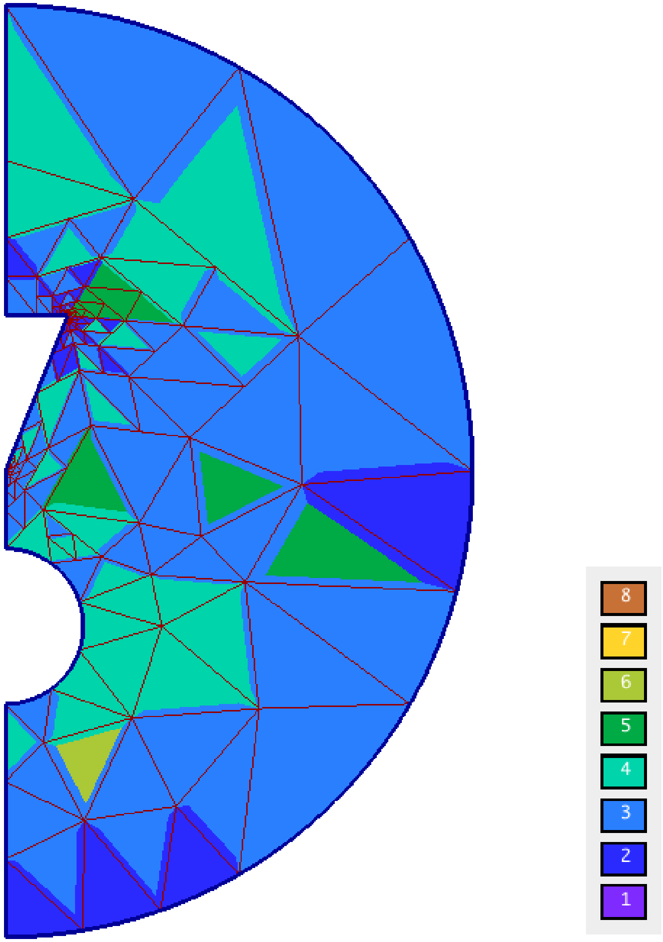
The fine elements and hanging nodes were generated mostly in the region of the peak of the conical electrode. The spherical electrode is simulated by curvilinear elements, i.e., quite precisely (and the distribution of electric field in its vicinity, therefore, is modeled with a very high accuracy).



Value of the total electrostatic energy in the system as a function of the number of DOFs.

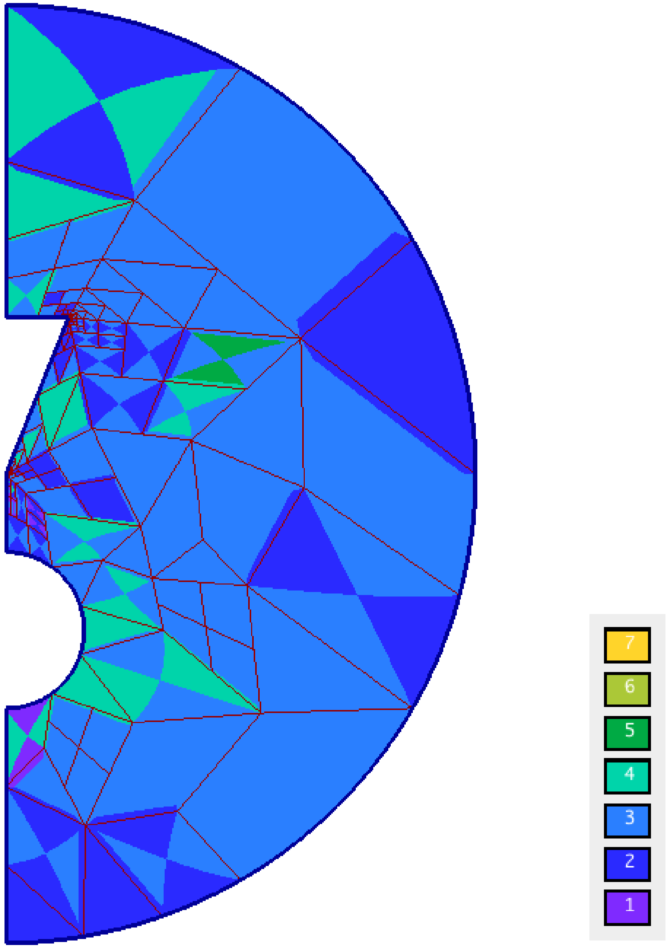
Codes FEMM and QuickField only work with linear elements without adaptivity and the results obviously converge very slowly. Faster is the convergence in Comsol Multiphysics, mainly with switched-on adaptivity. But this code does not support the hanging nodes, so that much more elements are needed.

Comparison of two types of meshes in Agros



triangles - 1402 DOFs

$\eta < 1$ %



triangles + quadrilaterals - 1186 DOFs

5. Conclusion

The methods of adaptivity described in the presentation and implemented into our codes Agros and Hermes represent a powerful tool whose application leads to significant savings in DOFs in comparison with various available SW (by one order and more) at the same or higher accuracy of the results.

Next work in the field will be aimed at the following items:

- **further improvement of criteria for the choice of the *h*-adaptivity, *p*-adaptivity and *hp*-adaptivity (in present versions of codes Agros and Hermes they are based on the evaluation of the distribution of the local error over the element),**
- **finding more effective criteria for introduction of quadrilaterals into the mesh pattern in case that the current distribution is smooth (a more reliable definition of “smoothness“, and**
- **implementation of elements of the goal-oriented adaptivity.**

..... and finally:

Thank you for your kind attention !