

Code Agros2D and Possibilities of its Use for Modeling and Design of Magnetic Bearings

Pavel Karban, František Mach, Pavel Kůs, Lukáš Korous, Ivo Doležel

University of West Bohemia, Univerzitní 26, 30614 Pilsen, Czech Republic, {karban, fmach, pkus, korous, idolezel}@kte.zcu.cz

Abstract—A multi-platform software Agros2D is presented, whose purpose is to model complex 2D multiphysics problems in engineering practice. The code is intended for monolithic solving of systems of generally nonlinear and nonstationary partial differential equations describing particular physical fields. The numerical schemes implemented in the code are based on a fully adaptive higher-order finite element method. The paper discusses the unique features of the code and its internal structure. Its power is demonstrated on an example aimed at modeling of a passive axial magnetic bearing.

I. INTRODUCTION

Numerous engineering problems are characterized by cooperation of several physical fields. In many cases one field (for example electromagnetic) produces another field (for example thermal) and these fields influence mutually one another.

Even when the physical essences and continuous mathematical models of many phenomena of this kind are well known, their numerical (or computer) modeling is still a challenging problem. Their continuous mathematical models consist of two or more generally nonlinear and nonstationary partial differential equations describing the distributions of involved physical fields. This system is often supplemented with a set of ordinary differential equations describing corresponding circuits (electric, magnetic, thermal, hydraulic, etc.). The coefficients of all these equations can be complicated functions of physical parameters of present materials and media that, however, depend on the state variables such as temperature or pressure.

For the solution of the indicated problems we can use nowadays two groups of codes. The first group are commercial professional codes developed by software companies and used both in industry and academy, the second group is represented by open source codes developed and used mostly in the academic sphere. The existing commercial codes are mostly based on lower-order finite element methods, work with elements of the first and second orders and miss advanced adaptive techniques. On the other hand, they are user-friendly due to very well elaborated preprocessors and postprocessors). In this group, just COMSOL Multiphysics works with elements up to the fifth order and a standard is h -adaptivity implemented in it.

The development of codes with advanced adaptive methods is intensively conducted mostly in the open-source academia groups. But even when they often contain strong algorithms and procedures, their user friendliness is rather low and some

of them are yet not suitable for solving complicated technical problems. Nevertheless, even in this domain the progress is fast. Our group has been developing for years software Agros2D [1], [2] based on the library Hermes2D [3].

The aim of this paper is to show the structure of the code, its unique features, and an illustrative example aimed at modeling of a passive axial magnetic bearing.

II. BASIC CHARACTERISTICS AND UNIQUE FEATURES OF AGROS2D

Agros2D is a multi-platform engineering software for solution of nonlinear coupled problems from a number of engineering disciplines (example of its main window being depicted in Fig. 1) The code representing a powerful preprocessor

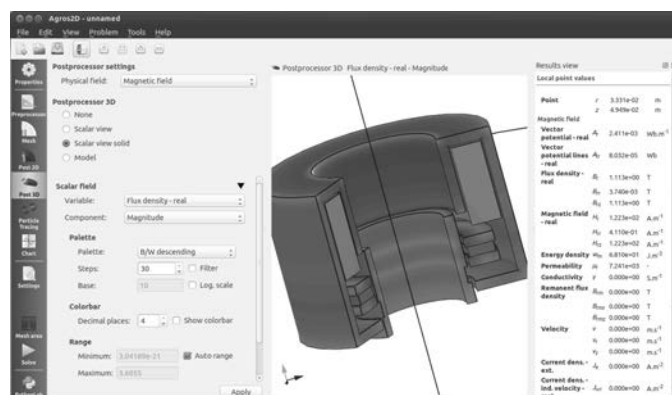


Figure 1. Main window of Agros2D

and postprocessor cooperates with the library Hermes that includes the most advanced numerical algorithms based on a fully adaptive higher-order finite element method and also discontinuous Galerkin method. Both packages written in C++ are freely distributable under the GNU General Public License.

The codes exhibit quite unique features, for example:

- Solution of multiphysics problems characterized by weak or hard coupling of the PDE system) with multi-mesh technology (every physical field can be calculated on a different mesh generally varying in time).
- Steady state, transient and harmonic analysis for physical fields (electric currents, electrostatics, magnetic, heat transfer, structural mechanics, radio frequency, acoustics, incompressible flow).

- Advanced implementations of the Newton and Picard solvers for the solutions of problems described by nonlinear PDEs.
- Full space (h , p and hp -adaptivity) [4] and also time adaptivity.
- Triangular, quadrilateral and curved elements with hanging nodes of any level.
- Advanced optimization techniques using the genetic algorithms (simulated annealing, NSGA II and several others).
- Higher-order methods of particle tracing, higher-order Eggshell Method for computation of forces acting in nonlinear magnetic fields.

Some of these features important for modeling and design of the magnetic bearings are described in more detail:

III. SOLUTION OF NONLINEAR MAGNETIC PROBLEMS BY NEWTON METHOD

The use of the Newton method is a well-established option for solving nonlinear PDEs, see, e.g., [5], [6]. The theory of its convergence is, however, rather complicated and the results are restricted to simple academic examples. The matter is even more complicated due to the necessity of damping, attempts to save computation time by using the same Jacobian for several successive steps, or other variants of the algorithm. In the case of solution of complicated problems arising in the engineering practice, one usually has to fine-tune the method for each problem individually in order to secure sufficiently fast convergence and avoid extensively long calculation.

The theory of the Newton method is widely discussed in the literature. Most of the results are, however, from the field of large nonlinear systems not coming from from the area of partial differential equations. There are generally several parameters of the Newton method, that may be changed to secure the maximum speed of the convergence process. We will first present the way Jacobian and residual are derived from the weak formulation. In the following, detailed implementation of the Newton solver will be described, together with all different variants and parameters.

A. Damping

By doing one step of the Newton method with damping coefficient $c_D \leq 1$ we mean using equation

$$Y^{k+1} = Y^k + c_D D^{k+1}. \quad (1)$$

The calculated direction D^{k+1} is used, but the step in that direction is made shorter. It makes the Newton method more secure. Damping is the key for solving difficult problems with strong nonlinearities. Without damping, some problems would not be solvable. On the other hand, using small fixed damping coefficient for all problems would seriously damage convergence of the method for simpler problems. The solution is the use of more variants of the method. Some of the parameters controlling damping in our algorithm follow:

- Selection of fixed or automatic damping.

- Damping coefficient c_D for fixed damping (if $c_D = 1$, then no damping is used) and maximum acceptable damping coefficient for the automatic damping.

The following points only apply for the automatic damping.

- Maximum allowed ratio r_D between the actual and previous residual norm. If this ratio is exceeded, the step is refused, the damping factor is decreased and the step is repeated.
- Coefficient for decrease of the damping factor in the situation described in the previous item. For all calculations in this paper, the value 1/2 has been used.
- Number of steps, after which the damping factor may be increased again.

B. Jacobian reuse

The modified Newton method (using the same Jacobian for all iterations) has been proven useful by many authors. The advantage of not evaluating the Jacobian in each step is huge and justifies a somewhat slower convergence. For some problems, however, it is not applicable, since the method would not converge at all. In our algorithm, we introduced several parameters that control reuse of the Jacobian and allow benefit from both approaches.

- Jacobian reuse may be switched off.
- Maximum allowed ratio r_J between the actual and previous residual norms for Jacobian reuse.
- Maximum number m_J of the steps with the same Jacobian.

If the Jacobian reuse is allowed, after each full step with calculation of Jacobian, at most m_J steps with this Jacobian are performed. If, however, in some of those steps the residual is not reduced by the factor r_J , this step (with the reused Jacobian) is refused and the Jacobian is recalculated again. Again, it is not clear what are the optimal values of m_J for general nonlinear problems. Not to reuse Jacobian is the safest way, but, for some problems, the Jacobian reuse may speed up the algorithm significantly. On the other hand, using large values of r_J close to 1 (which means that even very small decrease of the residual norm is accepted) may be dangerous. In some cases such setting increases the possibility of getting in a wrong direction and ending up in a local minimum. Values $r_J \geq 1$ are definitely not recommended.

IV. FORCE CALCULATION METHODS

The evaluation of forces and torques acting on ferromagnetic elements in non-linear magnetic fields is still a challenge. The algorithms based on the Virtual Work and Maxwell Stress Tensor are slow, the convergence is rather poor and accuracy of the results is far from being high. Lower-order Eggshell Methods are generally better [7], but they often suffer from the same reasons Agros2D offers an efficient higher-order eggshell method whose algorithms are described below.

A. Computation of Forces Using Maxwell Stress Tensor

The Maxwell stress tensor S_M can be written in the form

$$S_M = -\frac{1}{2\mu}(\mathbf{B} \cdot \mathbf{B})\mathbf{I} + \frac{1}{\mu}\mathbf{B} \otimes \mathbf{B}, \quad (2)$$

where \mathbf{B} is the magnetic flux density vector, μ denotes the permeability, \mathbf{I} is the unit diagonal matrix and the sign \otimes represents the dyadic product. The force \mathbf{F}_M acting on the body of volume V and boundary S is now given by the integral

$$\mathbf{F}_M = \int_S S_M dS, \quad (3)$$

where S denotes the outward unit normal vector to the boundary of the body. In the same way the torque acting on the same body is given by integral

$$\mathbf{T}_M = \int_S \mathbf{r} \times (S_M dS), \quad (4)$$

where \mathbf{r} is the position vector.

B. Eggshell method

The Eggshell method is based on introducing a thin shell over the surface of the body and a function satisfying the conditions $\gamma = 0$ along the external boundary of the shell and $\gamma = 1$ along its internal boundary. This function can be

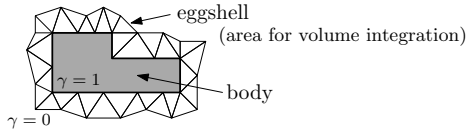


Figure 2. Principal scheme of Eggshell method

chosen in many ways. In our case, we used the solution of the Laplace equation in the eggshell with the prescribed boundary conditions. From (3) and definition of γ , the resultant force on the body can be expressed in the form

$$\mathbf{F}_M = \int_V S_M \cdot \text{grad } \gamma dV, \quad (5)$$

where V is the volume of the eggshell. The torque can be determined similarly. In our implementation, we use the solution of the Laplace equation in the eggshell domain obtained using higher-order finite elements. These bring additional smoothness to the solution and we can choose their orders arbitrarily. Since the eggshell domain is covered with very few elements, the additional computational cost caused by increasing the element order is quite negligible.

V. ILLUSTRATIVE EXAMPLE

The power of the code and also of the method itself is demonstrated on the computation of the axial force acting on the passive magnetic bearing for a novel design of a valve (see Fig. 3). The principal arrangement of the bearing together with the principal dimensions is depicted in Fig. 4.

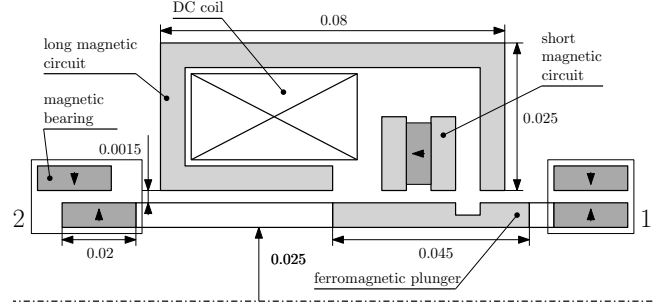


Figure 3. Magnetic circuit of valve with axial bearing with main dimensions

A. Mathematical model

The continuous mathematical model of the problem is represented by strongly nonlinear partial differential equations providing the distribution of static magnetic field in the system. The distribution of magnetic field is described in terms of magnetic vector potential \mathbf{A} by the equation

$$\text{curl} \left(\frac{1}{\mu} (\text{curl } \mathbf{A} - \mathbf{B}_r) \right) = 0, \quad (6)$$

where μ stands for the magnetic permeability and \mathbf{B}_r is the remanent magnetization. A sufficiently distant artificial boundary is characterized by the Dirichlet condition $\mathbf{A} = 0$.

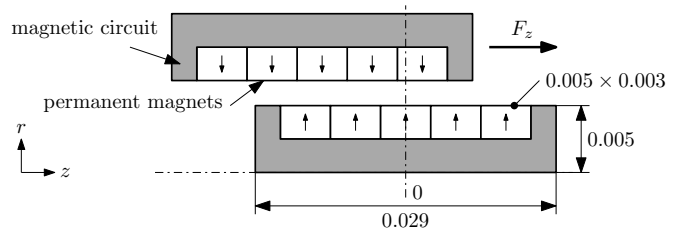


Figure 4. Detail of principal scheme of bearing

Magnetic parts are made of M-50 Bearing Steel whose saturation characteristic is shown in Fig. 5. The permanent

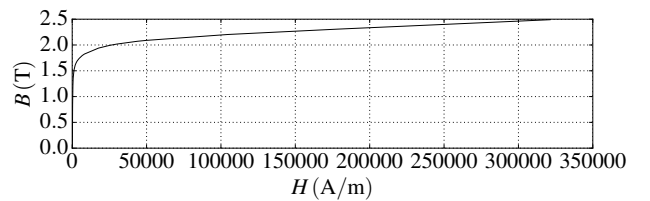


Figure 5. Saturation curve of bearing steel M-50

magnets VMM10 are of NdFeB type and their manufacturer provides the following parameters: remanence $B_r = 1.4$ T, relative permeability in the second quadrant $\mu_r = 1.2$ and maximum allowable temperature is 80 °C.

B. Computation

Table I contains the parameters of computation needed for determining the forces at one position of the axial bearing

($z = 0.01$ m). The number of DOFs was comparable, the prescribed residual norm was 10^{-4} .

Table I. COMPARISON OF PARAMETERS OF COMPUTATION USING AGROS2D, COMSOL AND FEMM (UNKNOWN RESIDUAL ACCURACY)

	Agros2D 3.2	COMSOL 4.4 Newton's	COMSOL 4.4 Dogleg	FEMM 4.2
DOFs	69037	69426	69426	79908
time (s)	41	74	54	(25)
iterations	31	54	43	(21)
force (N)	8.91	8.34	8.34	8.28

Fig. 6 depicts the behavior of the absolute norm of residual on the number of iterations. Fig. 7 shows the influence of

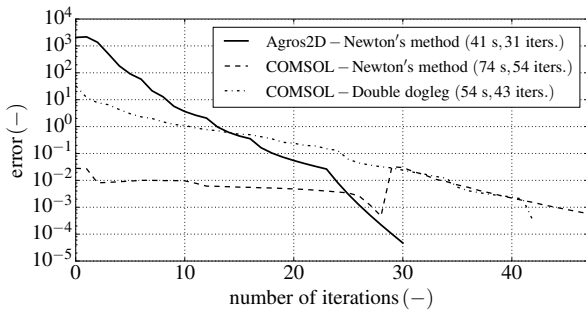


Figure 6. Comparison of absolute norm of residual on number of iterations selection of the damping factor on the number of iterations. Fig. 8 depicts the mesh for the solution of the magnetic field

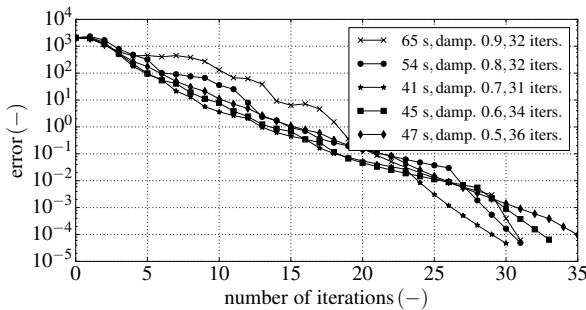


Figure 7. Influence of damping factor on curve error versus iterations

after the process of the automatic adaptivity. The number of DOFs is 69037 and solution of the system exhibits an error of 0.4 % compared to the reference solution.

C. Obtained results

Fig. 9 shows the distribution of magnetic field in the system. Fig. 10 shows the static characteristic computed by Agros2D and FEMM using the Maxwell tensor, Eggshell methods and virtual work principle. It is clear that even the results obtained by the Eggshell method are much better (the curve is smoother).

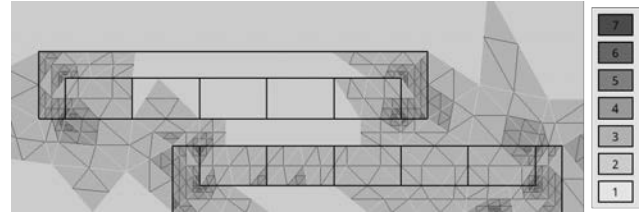


Figure 8. Discretization mesh and polynomial order used for computation

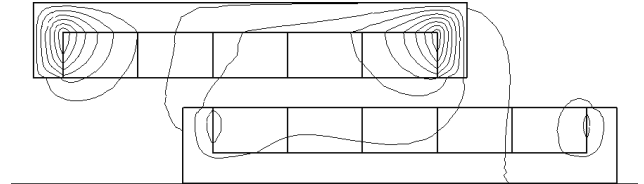


Figure 9. Magnetic force lines

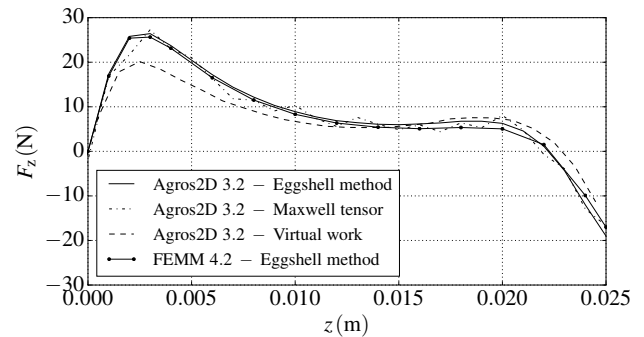


Figure 10. Comparison of Agros2D and FEMM for calculating force acting on the bearing

VI. TIME ADAPTIVITY

Time adaptivity is an important tool in improving efficiency of solution of time-dependent partial differential equations. Many problems exhibit rapid changes in solution in some parts of the time interval, while they do not change significantly otherwise. Using fixed time step for such problems would be inefficient. Rather than that, Agros2D allows the user to select adaptive time stepping. In such a case, optimal time step is determined automatically using an error estimate in the form of difference of two methods with different order.

VII. OPTIMIZATION

A continuous improvement of computers and their increasing computational power brings the possibility to solve completely new types of engineering problems. Not only larger and more complicated problems may be solved thank to the possibility of solving larger discrete problems, but also large sequence of calculations of the same model with different parameters may be performed. It can be done in parallel (thank to inherited parallelism of the considered problem), but also using ordinary PC, since the calculation time of each run may be short enough. It opens completely new areas,

such as parametric analysis, material and shape optimization, inverse problems, etc., for an ordinary PC users. Agros2D brings several possibilities to perform such type of multiple calculations in a systematic way.

A. Scripting in Agros2D using Python language

The first possibility is the use of scripting. It is an integral part of the Agros2D software, which largely enhances its capabilities. Instead of developing a new scripting language, as it is usual in many commercial codes, we decided to use Python. This choice has several advantages. First of all, most users know Python, at least on an elementary level, which enables them to use scripting in Agros2D without the necessity to learn a new language. Another advantage is the possibility to use whole range of already developed scientific libraries for post-processing and visualization of results. This combination of advanced finite-element calculations with large amount of ready-to-use Python libraries is a very powerful tool.

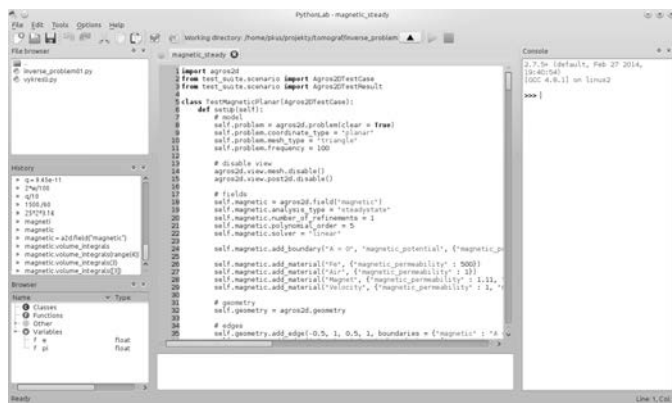


Figure 11. Main window of PythonLab

In Agros2D, almost every action or setting (including geometry definition, material specification, solver setting, evaluation of post-processor quantities, etc.), that can be done through graphical user interface, can be done from the script as well. This is a very important feature for more advanced analyses. It is also possible to use the GUI to create a basic model and convert it to an equivalent script, which can be used as a starting point.

B. Optimization framework OptiLab

The integration of a Python interpret into the application is a truly powerful tool that allows the user to perform whole range of different and specific analyses. Each user has different needs and no GUI, however complicated, can fulfill all such requirements. One of the most often used application of previously described functionality is a parametric analysis and optimization. Although it is perfectly possible to perform all the analysis (development of optimization algorithm, management of different designs, post-processing, graphs and figures generation) using our scripting tool PythonLab, it is convenient to approach those standard problems of engineering design in a more systematic way. A newly developed optimization framework OptiLab is intended to provide such service.

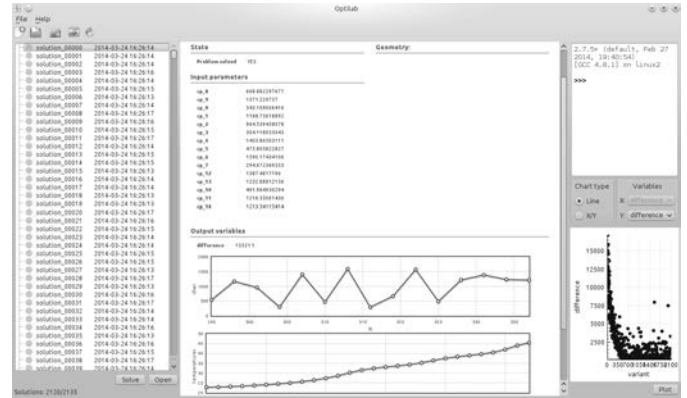


Figure 12. Main window of OptiLab

C. Parametric analysis

Parametric analysis can be viewed as a first part of the optimization process. It is very simple and productive using Agros2D and OptiLab. First an ordinary model is created with ease using Agros2D graphical user interface. After preliminary calculations are carried out and the model is verified, parametric analysis may start. Almost any number quantity defining shape, material properties or boundary conditions can be replaced by a variable defined in the startup script. This may be useful alone, but becomes extremely important after conversion of the model into Python script. All such defined variables may be controlled using the script itself (say, using for-cycle) or using diagnostic tools of OptiLab.

The OptiLab application is intended primarily as a management tool for results of multiple calculations analysis, no matter how it has been obtained, e.g. simple parameter sweep, optimization using gradient methods, genetic algorithms, multi-criteria optimization, etc. Its role is in managing sets of results described by design parameters such as geometry dimensions, material properties, etc. and calculated quantities, such as point values, integral values, etc. Entries corresponding to individual calculations may also contain images, graphs, convergence information and other more complex results, so that the analysis may be performed on those calculated data only without the necessity to perform time-consuming finite element calculations again.

D. Optimization algorithms

Optimization is a more advanced step in the engineering design, which automates the search of design parameters, which has been described in the previous section. The work-flow, however, does not differ much from the previously described case. One also has to create a parametric model and can use OptiLab for management of obtained parametric results and to create various graphs, statistics, etc. The only difference is the need of employment of some optimization algorithm. The user can create his own, use some of standard optimization algorithms implemented in some of Python's scientific libraries or use one of the tools provided in our framework. The only thing that has to be done is to define optimization functionals

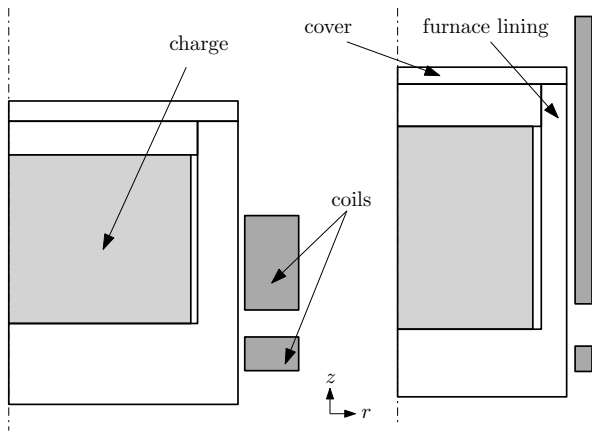


Figure 13. Two variants of geometry design of inductor

based on results of the calculation of the model. We have implemented some basic optimization algorithms, conjugated gradients and genetic algorithms.

E. Multi-objective optimization

In some situations, it is not possible to define one functional only, which should be maximized or minimized. In such cases, more functionals may be defined, see, e.g., [8]. Then it is not possible to find an optimal design, since the requirements usually go against each other (such as maximization of output power and minimization of energy consumption). Instead of that, a Pareto front (such as example shown in Fig. 14) is found. Designs on this Pareto front are all optimal in a sense that it is not possible to improve one functional without worsening the other one. Tools for a simple work with multi-objective optimization are also included in the OptiLab toolbox.

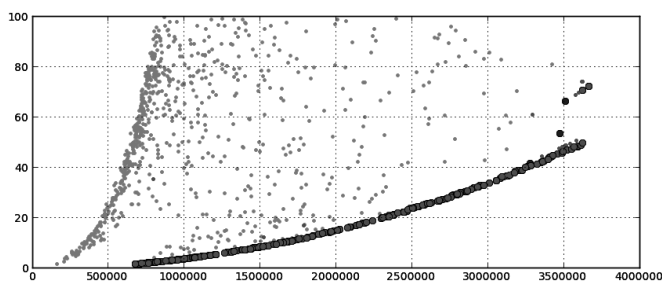


Figure 14. Pareto front as a result of optimization algorithm

The use of OptiLab is not restricted to multiple calculations with Agros2D software. It can be used as a stand alone application, provided that some other application would generate data in a required format. It is allowed by the fact that even though Agros2D, PythonLab and OptiLab are intended as collaborating pieces of software (and by this collaboration a maximal efficiency may be achieved), they can all work as individual applications. This loose interconnection brings large

flexibility and allows to solve much more complicated and non-standard problems that one huge, overcomplicated application would permit.

F. Illustrative example

For the concrete examples of optimization of various engineering problems, that have been done using our software, we refer the reader to some of our previous works [8], [9], [10]. In order to show what type of results can be obtained using optimization tools we included sample results of one such example. The goal was to optimize shape of a furnace and an inductor used to heat up the metal. Two possible variants of such design can be seen in Fig. 13. The goal was to maximize heat generated inside the metal (x axis) and, at the same time, minimize differences in temperature of various areas of the metal (y axis in Fig. 14, respectively). Those two goals cannot be achieved at the same time, so a Pareto front is formed by optimal designs. Each point in the figure corresponds to one calculation with different parameters. Large points are those on the Pareto front. It is than a task for an expert to choose appropriate trade-off.

VIII. ACKNOWLEDGEMENTS

This work was supported by the European Regional Development Fund and Ministry of Education, Youth and Sports of the Czech Republic (project No. CZ.1.05/2.1.00/03.0094: Regional Innovation Center for Electrical Engineering - RICE) and by the project GACR P102/11/0498 (Grant Agency of the Czech Republic).

REFERENCES

- [1] Karban, P., Mach, F., Kus, P., Panek, D., Dolezel, I. *Numerical Solution of Coupled Problems Using Code Agros2D*. Computing 95 (2013), No. 1, Supplement, pp 381–408, DOI 10.1007/s00607-013-0294-4.
- [2] Agros2D, <http://www.agros2d.org>
- [3] Hermes, <http://www.hpfem.org/hermes>
- [4] Solin, P., Cerveny, J., Dolezel, I. *Arbitrary-Level Hanging Nodes and Automatic Adaptivity in the hp-FEM*. Math. Comput. Simul., 2008, Volume 77, pp. 117–132.
- [5] Wriggers, P., *Nonlinear Finite Element Methods*, Springer-Verlag, Berlin Heidelberg, 2010.
- [6] Deuffhard, P., *Newton Methods for Nonlinear Problems*, Springer Series in Computational Mathematics, Volume 35, 2004.
- [7] Henrotte, F., Felden, M., vander Giet, M., Hameyer, K. *The Eggshell Approach for the Computation of Electromagnetic Forces in 2D and 3D*. COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering 23 (2004), No. 4, pp. 996–1005.
- [8] Di Barba, P., Dolezel, I., Karban, P., Kus, P., Mach, F., Mognaschi, M.E., Savini, A. *Multiphysics field analysis and multiobjective design optimization: a benchmark problem*. Inverse Problems in Science and Engineering 22 (2014), No. 7, pp. 1214–1225.
- [9] Mach, F., Kus, P., Karban, P., Dolezel, I. *Optimization of the system for induction heating of nonmagnetic cylindrical billets in rotating magnetic field produced by permanent magnets*. Computing 95 (2013), No. 1, Supplement, pp 537–552.
- [10] Mach, F., Kus, P., Karban, P., Dolezel, I. *Optimized arrangement of device for electrostatic separation of plastic particles* ELEKTRO (2012), pp 431–434.