Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **1**, No. 1, pp. 56-68 (2009)

February 2009

On Scientific Data and Image Compression Based on Adaptive Higher-Order FEM

P. Solin[1,*] and D. Andrs[2]

[1] *Department of Mathematics and Statistics, University of Nevada, Reno, USA.*
[2] *Institute of Thermomechanics, Prague, Czech Republic.*

Received 27 October 2008; Accepted (in revised version) 19 December 2008
Available online 16 February 2009

Abstract. We present a novel compression algorithm for 2D scientific data and images based on exponentially-convergent adaptive higher-order finite element methods (FEM). So far, FEM has been used mainly for the solution of partial differential equations (PDE), but we show that it can be applied to data and image compression easily. The adaptive compression algorithm is trivial compared to adaptive FEM algorithms for PDE since the error estimation step is not present. The method attains extremely high compression rates and is able to compress a data set or an image with any prescribed error tolerance. Compressed data and images are stored in the standard FEM format, which makes it possible to analyze them using standard PDE visualization software. Numerical examples are shown. The method is presented in such a way that it can be understood by readers who may not be experts of the finite element method.

AMS subject classifications: 68U10, 94A08, 94A11

Key words: Data compression, image compression, adaptive hp-FEM, orthogonal projection, best approximation problem, error control.

1 Introduction

The finite element method (FEM) is the most widely used numerical method for the solution of partial differential equations (PDEs) — see, e.g., [3, 6, 10]. The PDEs describe various natural processes on macroscopic scale, such as fluid flow, elasticity, heat transfer, electromagnetics, etc. The FEM splits the computational domain into a set of geometrically simple subdomains (elements) such as quadrilaterals in 2D or hexahedra in 3D. Inside the elements, the physical fields are approximated by polynomials. The coefficients of the polynomials are called *degrees of freedom (DOF)*. Performance of an adaptive FEM algorithm is the rate at which the approximation error

*Corresponding author.
URL: `http://hpfem.math.unr.edu/people/pavel`
Email: `solin@unr.edu` (P. Solin), `andrs@unr.edu` (D. Andrs)

decreases. This rate can be measured either in terms of the CPU (computer) time or the number of DOF in the discrete problem.

The hp-FEM is a modern version of FEM capable of extremely fast (exponential) convergence [1,2]. In practical computations, adaptive hp-FEM routinely outperforms standard adaptive FEM by orders of magnitude in terms of both the number of DOF and CPU time [8, 9]. The extremely high efficiency of the hp-FEM has its roots in the approximation theory: Very smooth functions with small local changes are approximated optimally using large elements equipped with high-degree polynomials, while small low-degree elements are much more efficient in areas where the solution exhibits important small-scale features.

The outline of this paper is as follows: In Section 2 we describe the main idea of how adaptive hp-FEM can be applied to data and image compression. In Sections 3 we illustrate the methodology on three different examples. Conclusions and outlook are drawn in Section 4.

2 Applying FEM to data and image compression

The typical application of FEM is to approximate unknown solutions of PDE. However, FEM can also be applied to data and image compression naturally as follows: In 2D, the computational domain Ω is a rectangle containing the image. Usually, the domain is split into a finite number of pixels, and a greyscale image is represented by a discontinuous, pixel-wise constant function f. A color image consists of three such functions f_r, f_g, f_b for the red, green, and blue components, respectively. Unlike standard image compression algorithms such as JPEG, however, our method is not restricted to pixel-wise constant functions f – the data can be represented by an arbitrary real function f defined in Ω.

2.1 Finite element approximation

Let us explain briefly the way FEM works. The method uses a *finite element mesh*, which is a collection of nonoverlapping convex polygons covering Ω. Given the shape of Ω in our application, it is natural to use rectangular elements. A finite element mesh is said to be *regular* if no vertex of an element lies inside of an edge of another one, and *irregular* otherwise. This is illustrated in Fig. 1.

Most existing finite element codes use regular meshes, since they are easier to implement and analyze mathematically. However, when used with automatic adaptive algorithms, such meshes produce *regularity-enforced refinements* which slow down their convergence [7]. Irregular meshes are much better for adaptive algorithms since element refinement is a local operation, i.e., it does not cause any changes in neighboring elements. Technical details of hp-FEM approximation on arbitrarily-irregular meshes lie beyond the scope of this presentation, and we refer to [7].

The mesh over Ω consists of M elements $K_1, K_2, \ldots, K_M$ which are equipped with polynomial degrees $1 \leq p_i = p(K_i)$. Is standard FEM, the polynomial degree typically

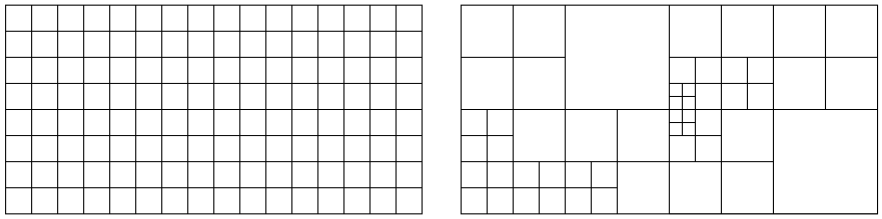
Figure 1: Regular mesh (left) and irregular mesh (right).

is uniformly $p_i = 1$ or $p_i = 2$ for all $i = 1, 2, \dots, M$. In the hp-FEM, polynomial degrees in elements vary. The geometry and polynomial degrees of elements determine uniquely a linear space of continuous, piecewise-polynomial functions

$$V_{h,p} = \{v;\ v \text{ is continuous in } \Omega;\ v \text{ restricted to } K_i \text{ is polynomial of degree } p_i\}. \tag{2.1}$$

In practice, we allow different *directional polynomial degrees* on quadrilateral elements, but this is a detail that we can skip for the moment (see, e.g., [9]).

The linear space $V_{h,p}$ has some dimension N and a basis consisting of piecewise-polynomial functions $v_1, v_2, \dots, v_N$. Any function $u_{h,p}$ in the space $V_{h,p}$ can be expressed uniquely as a linear combination of the basis functions v_j with some real coefficients y_j,

$$u_{h,p} = \sum_{j=1}^{N} y_j v_j. \tag{2.2}$$

Typical *variational (weak) formulation* of a (linear, stationary) PDE reads: Find $u_{h,p}$ in the space $V_{h,p}$ such that the identity

$$a(u_{h,p}, v_i) = l(v_i), \tag{2.3}$$

holds for all basis functions $v_1, v_2, \dots, v_N$. Both the bilinear form a and the linear form l are dictated by the PDE.

Of course, the coefficients $y_1, y_2, \dots, y_N$ are unknown at the moment. However, we can substitute (2.2) into (2.3) to obtain

$$\sum_{j=1}^{N} a(v_j, v_i) y_j = l(v_i) \quad \text{for all } v_1, v_2, \dots, v_N, \tag{2.4}$$

which is a system of linear algebraic equations of the form

$$SY = F. \tag{2.5}$$

The stiffness matrix $S = (s_{ij})_{i,j=1}^{N}$ has elements $s_{ij} = a(v_j, v_i)$, the unknown vector Y has the form $Y = (y_1, y_2, \dots, y_N)^T$, and the right-hand side vector F is defined as $F = (l(v_1), l(v_2), \dots, l(v_N))^T$. After system (2.5) is solved and the coefficients $y_1, y_2, \dots, y_N$ are known, the approximate solution $u_{h,p}$ is constructed via (2.2).

2.2 Best approximation of f on a given mesh

Consider a finite element mesh consisting of elements $K_1, K_2, \ldots, K_M$ equipped with polynomial degrees $p_1, p_2, \ldots, p_M$. This mesh defines a linear space $V_{h,p}$ of the form (2.1). In general, the function f representing the data or image of course does not lie in $V_{h,p}$. However, whenever f lies in some inner product space[†] V such that $V_{h,p} \subset V$ (which always is the case in practice), then the finite element method allows us to find its best approximation in $V_{h,p}$ easily.

Let us use the symbol $(u,v)_V$ for the inner product in V and $\|u\|_V$ for the corresponding norm related to $(u,v)_V$ via

$$\|u\|_V = \sqrt{(u,u)_V}.$$

By a classical result (see, e.g., [4]), the best approximation of f in $V_{h,p}$ is its orthogonal projection onto $V_{h,p}$. The orthogonal projection $u_{h,p}$ of f is defined via a simple condition making the difference $f - u_{h,p}$ orthogonal to all basis functions of $V_{h,p}$:

$$(f - u_{h,p}, v_i)_V = 0 \quad \text{for all } v_1, v_2, \ldots, v_N. \tag{2.6}$$

Using (2.2), condition (2.6) can be rewritten into

$$\sum_{j=1}^{N} \underbrace{(v_j, v_i)_V}_{a(v_j,v_i)} y_j = \underbrace{(f, v_i)_V}_{l(v_i)} \quad \text{for all } v_1, v_2, \ldots, v_N. \tag{2.7}$$

Hence we obtained a system of linear algebraic equations of the form (2.5). With a standard choice of the hp-FEM basis [9], the stiffness matrix S is sparse and symmetric positive definite, and thus it can be solved very efficiently using various preconditioned iterative matrix solvers [5].

In spaces of continuous, piecewise polynomial functions there are two natural inner products one can use: the L^2 product that only takes into account function values,

$$(u,v)_{L^2} = \int_\Omega u(x,y)v(x,y)\,\mathrm{d}x\mathrm{d}y,$$

and the H^1 product which also uses gradients,

$$(u,v)_{H^1} = \int_\Omega u(x,y)v(x,y) + \nabla u(x,y) \cdot \nabla v(x,y)\,\mathrm{d}x\mathrm{d}y.$$

Let us remark that although these products define global norms, the adaptivity algorithm can be designed in such a way that it focuses on specific *quantities of interest* [8].

If we are compressing a standard image, then the function f is discontinuous and thus it does not lie in the space $V = H^1$. However, there is an easy way to convert it in a lossless way into a continuous, pixel-wise bilinear function: In step 1, the function values of f from pixel centers are shifted into pixel vertices as illustrated in Fig. 2.

[†]See Appendix A in [6] for an introductory course on linear, normed, and inner product spaces.

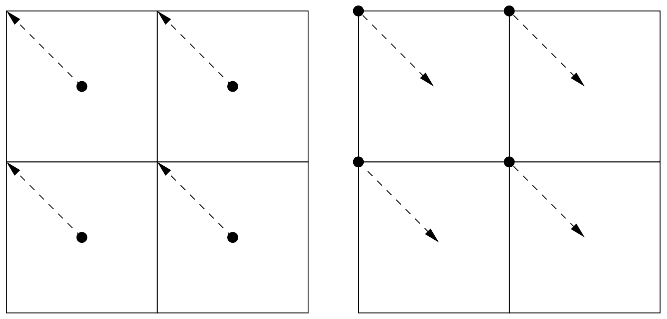
Figure 2: Left: lossless conversion of a discontinuous, pixel-wise constant function into a continuous function which is pixel-wise bilinear. Right: inverse transformation.

In step 2, values along the bottom and right edges are defined by duplicating values at adjacent vertices. Fig. 2 also depicts the inverse transformation. Note that the composition of the forward and backward transformations yields the original pixel-wise constant function.

2.3 Automatic adaptivity

In order to reduce the compression error $\|f - u_{h,p}\|_V$ optimally, we employ the adaptive hp-FEM algorithm based on arbitrary-level hanging nodes [7]. We do not find it necessary to reprint the algorithm here since it is used as-is, and moreover, it is part of the C/C++/Python library Hermes2D[‡] which is freely available to the reader under the GPL license.

Let us only remark that in our case the adaptive hp-FEM is much faster compared to solving a PDE, since are not looking for an unknown exact solution – our "exact solution" is the function f. This removes the need for a-posteriori error estimation, which is the most difficult and CPU time-consuming step of adaptive algorithms including [7]. In our case, we simply use the error function $f - u_{h,p}$ to guide the refinement decisions.

3 Examples

Next let us illustrate the performance of the adaptive hp-FEM algorithm on several examples: a simple image containing three diagonal squares, a satellite photograph of the Earth, and the standard image compression benchmark Lena.

3.1 Diagonal squares

First, let us compress a 512×512 image containing three diagonal squares (Fig. 3) to a prescribed relative error 0.01% in the L^2-norm.

[‡]http://spilka.math.unr.edu/projects/hermes2d

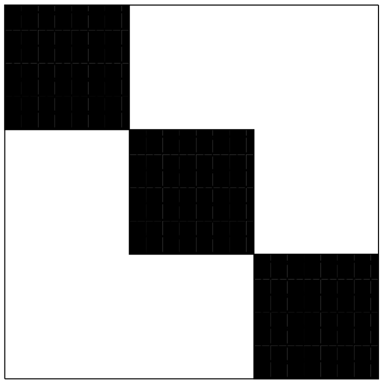
Figure 3: Sample 512 × 512 image containing three diagonal squares.

The algorithm starts with a single lowest-degree element covering the entire image, and it converges quickly during a few iterations. Figs.4-6 show the *hp*-FEM approximations along with the corresponding meshes. The symbols $NDOF$ and ERR stand for the number of degrees of freedom and the relative approximation error measured in L^2-norm.

Various colors in the meshes represent polynomial degrees of finite elements. The warmer the color, the higher the polynomial degree. Dark blue stands for $p = 1$, light blue for $p = 2$, etc. The presence of two different colors in an element indicates different polynomial orders in the horizontal and vertical directions.

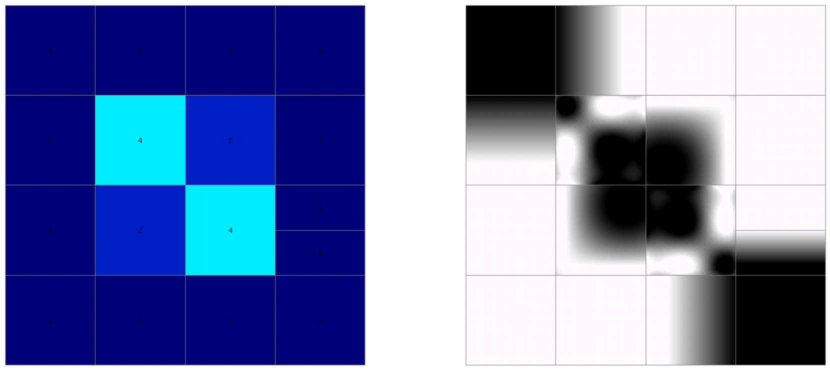
Figure 4: Mesh and approximation, $NDOF = 41$, $ERR = 18.1160\%$.

The reader can see that the image is represented well already with a relative error of 1%. We have observed the same with other images as well, which indicates that this accuracy level could be enough for the compression of images. However, other applications including scientific data compression may require a more accurate approximation.

The convergence history of the adaptive process is shown in Fig. 7.

Comparing fairly the performance of our method to JPEG or any other compres-

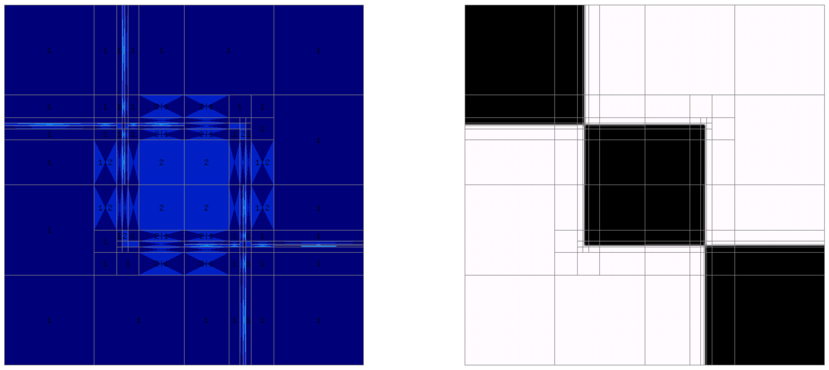
Figure 5: Mesh and approximation, $NDOF = 105$, $ERR = 6.5249\%$.

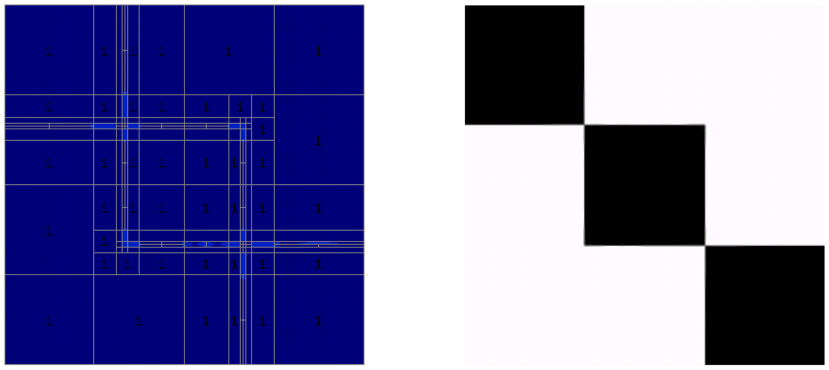
Figure 6: Mesh and approximation, $NDOF = 313$, $ERR = 0.9619\%$.

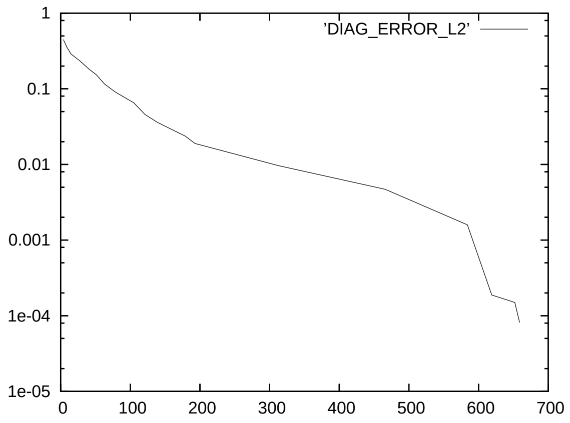

Figure 7: Convergence of the adaptive hp-FEM algorithm. Horizontal and vertical axes represent the number of DOF and the relative error in semi-logarithmic scale, respectively. The algorithm stopped after achieving a relative error of 0.01%.

sion technique which does not provide the compression error is problematic. In principle, this would be possible but one would have to implement an algorithm that measures the accuracy of the JPEG compression.

3.2 Satellite photograph of earth

Next we compress a satellite photograph of Earth shown in Fig. 8 (courtesy of NASA). The dimensions of this image are 1024×512 pixels.

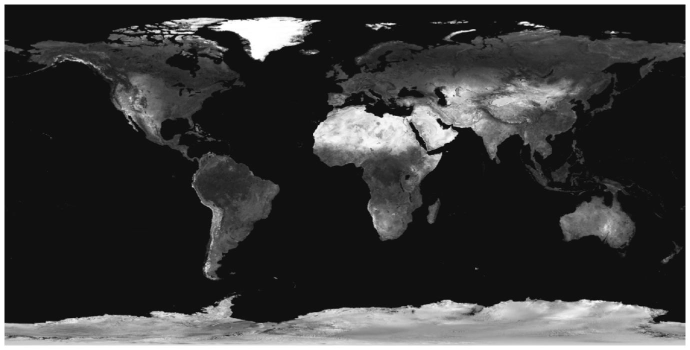
Figure 8: Satellite image of Earth.

In this case we set the relative error tolerance to 0.1% in the L^2 norm. The following Figs. 9 – 13 show the progress of the adaptive algorithm. Similarly to the previous example, the reader can see that a meaningful information about the structure of the image is captured already with an extremely small number of degrees of freedom (see Fig. 11).

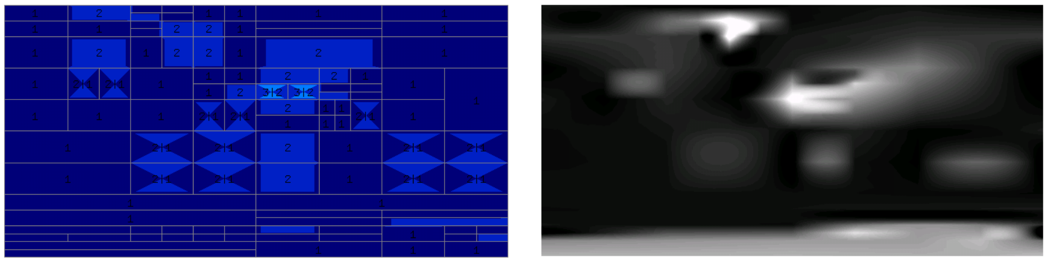
Figure 9: Mesh and approximation, $NDOF = 148$, $ERR = 26.1780\%$.

Fig. 14 shows the approximation error as a function of the number of degrees of freedom. Notice that after a fast initial error drop, the convergence curve becomes a straight line, i.e., it is exponential:

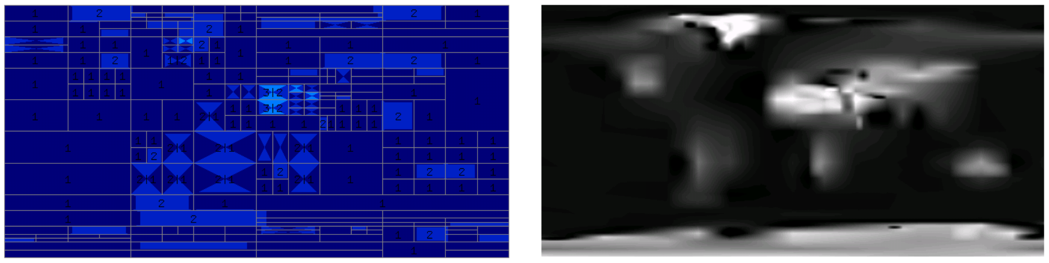
Figure 10: Mesh and approximation, $NDOF = 419$, $ERR = 18.2095\%$.

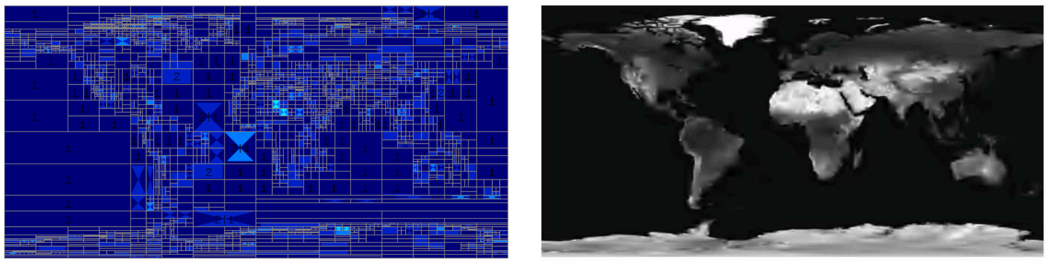
Figure 11: Mesh and approximation, $NDOF = 3778$, $ERR = 6.3895\%$.

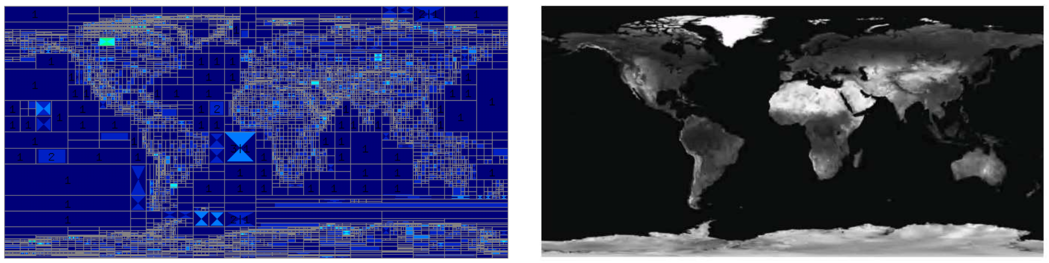
Figure 12: Mesh and approximation, $NDOF = 12125$, $ERR = 3.0938\%$.

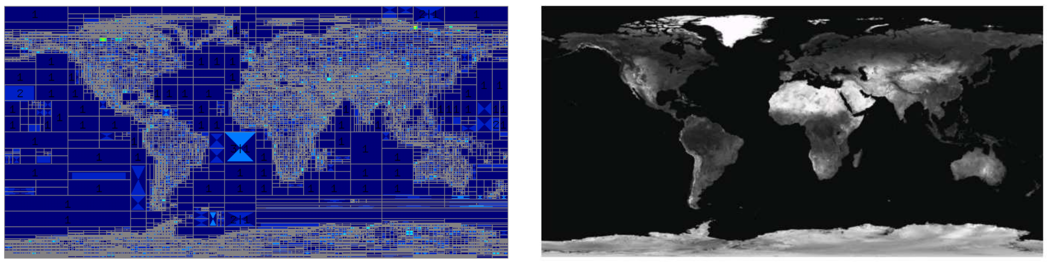
Figure 13: Mesh and approximation, $NDOF = 40901$, $ERR = 1.1916\%$.

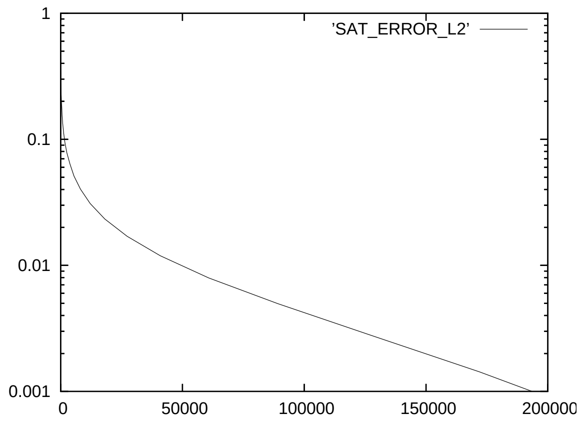

Figure 14: Convergence of the adaptive hp-FEM algorithm. Horizontal and vertical axes represent the number of DOF and the relative error in semi-logarithmic scale, respectively. The algorithm stopped after achieving the prescribed relative error of 0.1%.

3.3 Lena

The next example is a standard image compression benchmark test case called *Lena*. This is a 512 $\times$ 512 image with many different features such as large smooth surfaces, small details, blurred parts, sharp edges, etc. The image of Lena is shown in Fig. 15.

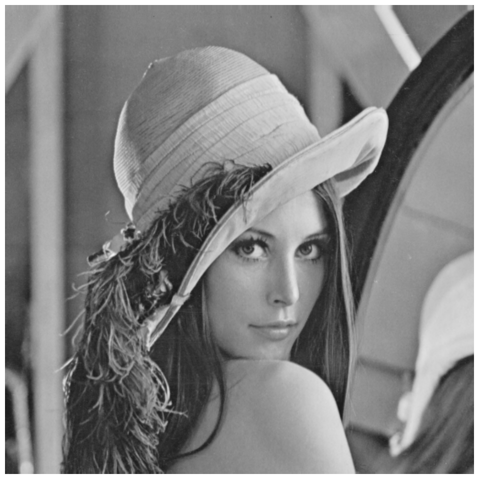

Figure 15: Lena: standard image compression benchmark problem.

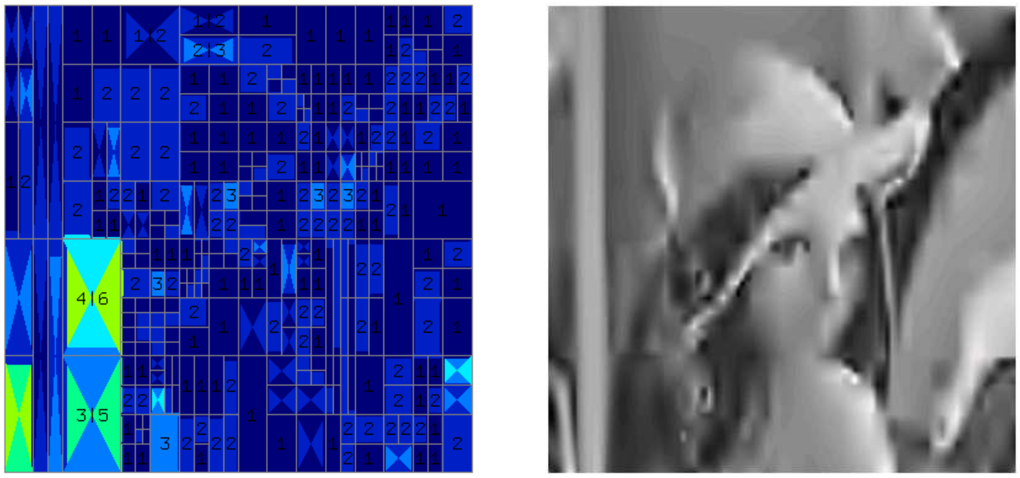
Figure 16: Mesh and approximation, $NDOF = 420$, $ERR = 10.0344\%$.

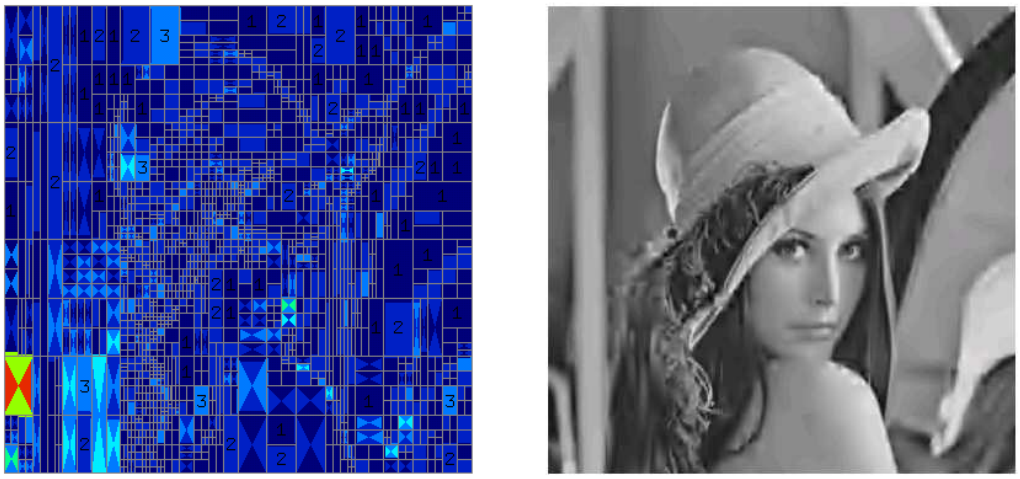
Figure 17: Mesh and approximation, $NDOF = 2129$, $ERR = 4.7012\%$.

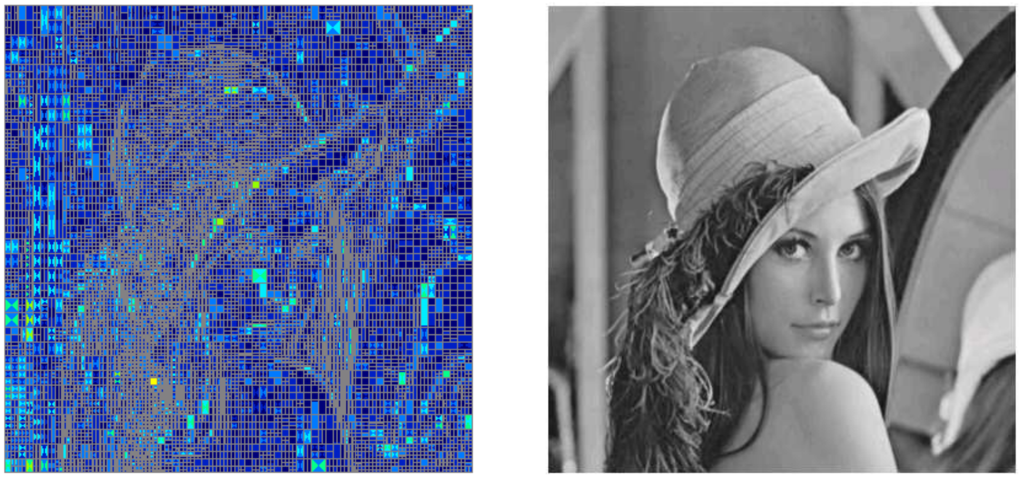
Figure 18: Mesh and approximation, $NDOF = 22372$, $ERR = 1.0449\%$.

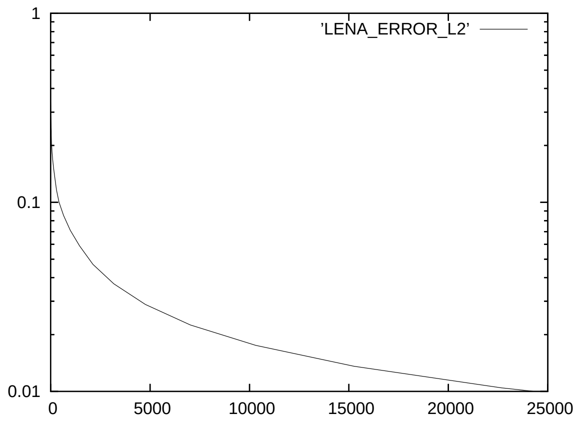

Figure 19: Convergence of the adaptive hp-FEM algorithm. Horizontal and vertical axes represent the number of DOF and the relative error in semi-logarithmic scale, respectively. The algorithm stopped after achieving a relative error of 1%.

Fig. 19 shows the approximation error as a function of the number of degrees of freedom.

4 Conclusion and outlook

In the examples above, the approximation error was measured in the so-called L^2-norm (also called least-squares norm). This is not the only option. Examples of other norms include the H^1-seminorm, H^1-norm, L^∞-norm, and others. While the L^2-norm measures volumetrical differences in function values, H^1-seminorm measures volumetrical differences in gradients, H^1-norm both the function values and gradients, and the L^∞-norm measures absolute differences in function values. The choice of the norm may influence the performance of the adaptive compression algorithm as well as the quality of the compressed images.

On the computer-scientific level, we need to find out the best possible data format to store the degrees of freedom and the finite element mesh. For the degrees of freedom, we will begin with applying the quantization tables used by JPEG, and we will see whether they can be used, need to be adjusted, or whether we need to invent a dffferent format. The mesh will not be stored via the coordinates of vertices. Our preliminary calculations indicate that it will be more efficient to store the first (coarsest element) plus the history of refinements.

Both the mathematical and computer-scientific aspects mentioned above involve nontrivial open problems and need to be investigated systematically.

Acknowledgments

The authors acknowledge the financial support of the Czech Science Foundation (Project No. 102/07/0496) and of the Grant Agency of the Academy of Sciences of the Czech Republic (Project No. IAA100760702)

References

[1] I. BABUŠKA AND W. GUI, *The h, p and hp-versions of the finite element method in one-dimension - Part III. The adaptive hp-version*, Numer. Math., 49 (1986), 659-683.
[2] I. BABUŠKA, B. SZABO, AND I.N.KATZ, *The p-version of the finite element method*, SIAM J. Numer. Anal., 18 (1981), pp. 515-545.
[3] K.J. BATHE, *Finite Element Procedures*, Prentice-Hall, Englewood Cliffs, (1995).
[4] W. RUDIN, *Functional Analysis*, McGraw-Hill Series in Higher Mathematics, (1991).
[5] Y. SAAD, *Iterative Methods for Sparse Linear Systems, Second Edition*, SIAM, Philadelphia, (2003).
[6] P. SOLIN, *Partial Differential Equations and the Finite Element Method*, J. Wiley & Sons, (2005).
[7] P. SOLIN, J. CERVENY AND I. DOLEZEL, *Arbitrary-Level hanging nodes and automatic adaptivity in the hp-FEM*, Math. Comput. Simul. 77 (2008), pp. 117-132.
[8] P. SOLIN AND L. DEMKOWICZ, *Goal-oriented hp-adaptivity for elliptic problems*, Comput. Methods Appl. Mech. Engrg. 193 (2004), pp. 449-468, .
[9] P. SOLIN, K. SEGETH AND I. DOLEZEL, *Higher-Order Finite Element Methods*, Chapman & Hall/CRC Press, (2003).
[10] O.C. ZIENKIEWICZ AND R.L. TAYLOR, *Finite Element Method- Basic Formulation and Linear Problems*, Vol. 1 McGraw-Hill Publ., New York, (1989), pp. 648.