Computer Science • 14 (1) 2013 http://dx.doi.org/10.7494/csci.2013.14.1.97

Piotr Gurgul
Marcin Sieniek
Maciej Paszyński
Łukasz Madej
Nathan Collier

TWO-DIMENSIONAL HP-ADAPTIVE ALGORITHM FOR CONTINUOUS APPROXIMATIONS OF MATERIAL DATA USING SPACE PROJECTION

Abstract *In this paper we utilize the concept of the L^2 and H^1 projections used to adaptively generate a continuous approximation of an input material data in the finite element (FE) base. This approximation, along with a corresponding FE mesh, can be used as material data for FE solvers. We begin with a brief theoretical background, followed by description of the hp-adaptive algorithm adopted here to improve gradually quality of the projections. We investigate also a few distinct sample problems, apply the aforementioned algorithms and conclude with numerical results evaluation.*

Keywords adaptive finite element method, projection operator, digital material representation

1. Introduction

Space projections constitute an important tool, which finds its use in diverse applications including finite element (FE) analysis [1]. It might be used, for example, to create an approximation of a generic bitmap in the FE space. Such a bitmap can represent e.g. morphology of the digital material representation (DMR) during FE analysis of material behavior under deformation and exploitation conditions [4]. Due to the crystallographic nature of polycrystalline material, particular features are characterized by different properties. To properly capture FE solution gradients, a specific refined meshes have to be created. Accurate identification of material data is important in several meterial science simulations, including Step-and-Flash Imprint Lithography ([8]) or strain-stress analysis ([4], [5]).

The operator can be applied iteratively on a series of increasingly refined meshes, which results in an improving fidelity of the approximation. The importance of even preliminary version of such an algorithm has already been proven ([3], [9]).

A number of adaptive algorithms for FE mesh refinements are known. HP adaptation is one of the most complex and accurate, as it provides an exponential convergence with the number of degrees of freedom ([2]). Our goal in this work is to apply the full hp adaptive algorithm to the projection operator in order to observe the predicted exponential convergence.

The method described in this paper differs from the Projection-Based Interpolation (PBI) operator, researched by some of the authors of this paper in [9] and [3]. PBI also delivers H^1 interpolations of an input function, but does not guarantee their optimality for the given amount of degrees of freedom. Instead, it is well suited for local computations and the aforementioned papers describe its local implementations. This is not the case for the method described in this paper, which guarantees optimal H^1 solutions, but is not particularly suitable for distributed processing.

We test the entire idea on selected microstructures, modeling of which poses an important problem to the material science community. Recently observed needs of the automotive and aerospace industries for new metallic materials that can meet strict requirements regarding weight/property ratio constitute a driving force for development of modern steel grades. Complicated thermomechanical operations are applied to obtain highly sophisticated microstructures with combination of e.g. large grains, small grains, inclusions, precipitates, multi-phase structures etc. These microstructure features and interactions between them at the micro-scale level during manufacturing or exploitation stages eventually result in elevated material properties at the macro-scale level. To support experimental research on these materials a numerical material model that can take mentioned microstructure features explicitly into account during FE analysis of processing and exploitation conditions has to be used. One of the solutions to deal with the explicit representation of microstructure features during numerical analysis is an approach based on the Digital Material Representation (DMR) [4]. There are two major issues that have to be addressed in this method. The first one is the development of algorithms for creation of structures that can represent

real morphology of both single and two phase microstructures [5]. The second one, which is addressed in the this work, is a problem of meshing of the created DMR as due to the nature of obtained microstructure a significant solution gradients (strain, stress etc.) are expected during numerical modeling. The third mesh is used to illustrate a case where a mesh generated for continuous representation of material data is useful for further FEM computations.

The paper is structured in the following way: section 2 contains the mathematical background for projection methods. In section 3 we describe the adaptive algorithm used to refine the base of the interpolant. Section 4 presents results we obtained when applying the described method to some microstructure images. We conclude the paper in section 5.

2. Space projections as a minimization problem

2.1. Elementary concepts

A L^2 projection in the space V may be stated as the following minimization problem: given an arbitrary function $f(x)$, find $u^V(x) \in V$ such that

$$\min \|f(x) - u^V(x)\|_{L^2(\Omega)} \tag{1}$$

Since $u^V(x) = \sum_{i=1}^{n} N_i(x)U_i$ where $\text{span}(\{N_i\}_{i=1\ldots n}) = V$, we need to determine $\{U_i\}_{i=1\ldots n}$, the coefficients of the linear combination of basis functions.

Given $\|f(x)\|_{L^2} = \int_\Omega f(x)\,\mathrm{d}\Omega$ we differentiate the equation with respect to its coefficients and set them equal to zero in order to find the minimum.

$$\frac{\partial}{\partial U_j}\left[\int_\Omega \left(f(x) - \sum_{i=1}^{n} N_i(x)U_i\right)^2 \mathrm{d}\Omega\right] = 0 \tag{2}$$

This leads to a linear system:

$$M \cdot U = F \tag{3}$$

where

$$M_{j,i} = \int_\Omega N_j(x)N_i(x)\,\mathrm{d}\Omega \tag{4}$$

$$F_j = \int_\Omega f(x)N_j(x)\,\mathrm{d}\Omega \tag{5}$$

the solution to which is the L^2 projection $u^V(x)$ of $f(x)$ onto the space V.

2.2. Including derivatives

The method above only minimizes the square of the error in the function itself. Derivative information can also be included by minimizing the sum of the error in the

function as well as for its gradients. This approach is called H^1 projection and can be expressed as a similar minimization problem. Given an arbitrary function $f(x)$, find $u^V(x) \in V$ such that:

$$\min \|f(x) - u^V(x)\|_{H^1(\Omega)} \tag{6}$$

which is equivalent to:

$$\min \left[\int_\Omega \left(f(x) - u^V(x)\right)^2 \mathrm{d}\Omega + \alpha \int_\Omega \left(\nabla f(x) - \nabla u^V(x)\right)^2 \mathrm{d}\Omega \right] \tag{7}$$

where α is a scalar parameter which scales the derivative portions. This leads to a similar linear system, defined by the matrix and vector:

$$M_{j,i} = \int_\Omega N_j(x) N_i(x) + \alpha \nabla N_j(x) \cdot \nabla N_i(x) \, \mathrm{d}\Omega \tag{8}$$

$$F_j = \int_\Omega f(x) N_j(x) d\Omega + \alpha \int_\Omega \nabla f(x) \cdot \nabla N_j(x) \mathrm{d}\Omega \tag{9}$$

We perform the above optimization in the space of hierarchical basis functions known from the hp-adaptive FE method and defined in [1].

When using this base, the solution of the equation 3 with the above $M_{j,i}$ and F_j leads to an interpolation of $f(x)$ in the $H^1(\Omega)$ space.

2.3. Projection properties

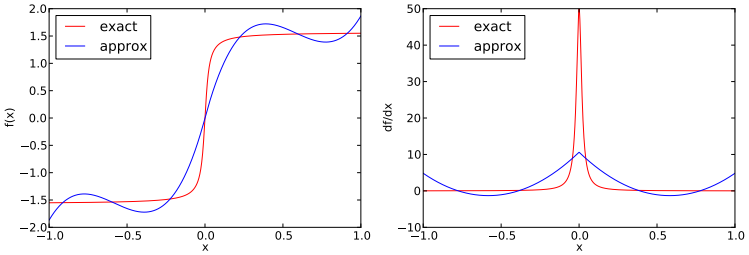
In order to demonstrate features of these methods of interpolation, we consider first a one-dimensional example by taking $f(x) = \operatorname{arc\,tg}(50x)$. If we take the pure L^2 projection ($\alpha = 0.0$), we can project this function onto a function space of two elements of cubic polynomials. The results of this are presented in Figure 1.

This method of interpolation by projecting onto function spaces is well suited for a FE code where the matrices one needs to form are common to that numerical method [1]. However, it is important to remember that the linear system in this case represents a projection and not a partial differential equation. As such, boundary conditions are not necessarily needed. Depending on the application, it may be desirable or not to enforce end conditions, which can be done as if they were Dirichlet boundary conditions in an FE code.

We have performed the same test projection and plot the results side by side with the original image. Note that the end conditions of the curve now interpolate the exact answer. However, this comes at the cost of reduced accuracy. In both cases, we list the value of the L^2 norm and note that the value increases for the case with end conditions set.

Another feature of this interpolation method is that oscillations may occur. While function $-\frac{\pi}{2} \leqslant f(x) \leqslant \frac{\pi}{2}$, the approximation dips above and below these values. There is nothing in the formulation which prevents this from occuring. Projection-based interpolation will minimize the error in the approximation, but does not guarantee any maximum or minimum.

Approximation of the function $f(x)$ (left) and its derivative (right). The L^2 norm of the error is 0.3806.



Approximation of the function $f(x)$ (left) and its derivative (right) with end conditions set. The L^2 norm of the error is 0.3977.

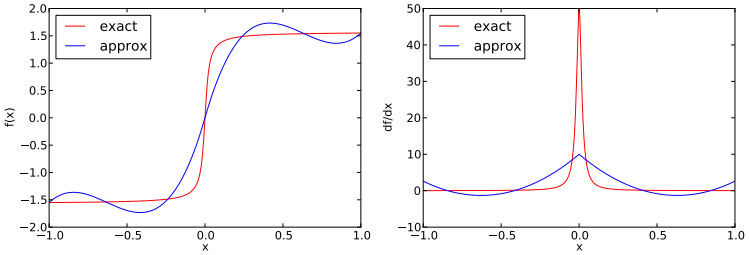


Figure 1. Projection of $f(x)$ onto a space of two cubic polynomials.

It may be of interest to accurately represent derivatives of the function as well as the function itself. In this case the parameter α must be chosen carefully. If $\alpha = 0$, then no derivative information is part of the minimization. If $\alpha = 10 \times 10^{10}$ is chosen, then the result heavily favors the derivatives. If we examine the matrix M for the H^1 projection, we obtain

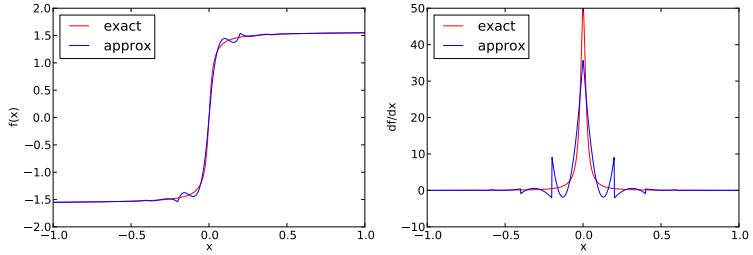
$$M_{j,i} = \int_\Omega N_j(x)N_i(x) + \alpha \nabla N_j(x) \cdot \nabla N_i(x) \, \mathrm{d}\Omega \tag{10}$$

Since the basis functions are $\mathcal{O}(1)$ and gradients are of order $\mathcal{O}(\frac{1}{h})$, if we select $\alpha = h^2$, it will evenly weight the derivative and function portions. All described cases are shown in Figure 2.

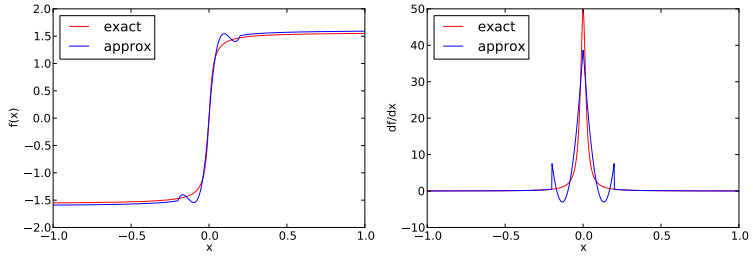
3. Adaptive algorithm for space refinement

To obtain a continuous interpolation of discontinuous data we apply the above method to a function U. As a result, we get an approximation u such that $||U-u||_V$ is minimal in a space of test functions V. The quality of the approximation depends on the choice of the space V. Since it is difficult to determine *a priori* the precision corresponding to a given choice of V, the process is repeated for a series of self-containing spaces $\{V_t\}_{t=1...m}$ where V_1 corresponds to the initial mesh and V_m reflects the the first mesh, for which the precission requirement is met.

$\alpha = 0$, good representation of the function, poor representation of derivatives



$\alpha = 10 \times 10^{10}$, good representation of the derivative but not the function



$\alpha = h^2 = 0.04$, balanced representation of both the derivative and the function

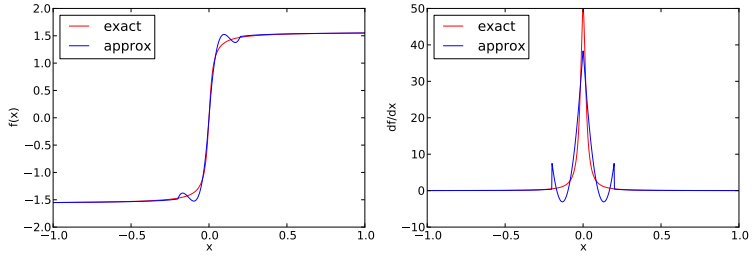


Figure 2. Approximation of the function $f(x)$ (left) and its derivative (right) by H^1 projection onto a space of two cubic polynomials.

In step t, three meshes are generated:

- a coarse mesh corresponding to V_t – equal to the optimal mesh from the previous step,
- a fine mesh corresponding to V_t^{fine} – created by uniform refinement of the coarse mesh,
- an optimal mesh corresponding to V_t^{opt} – which becomes a coarse mesh in the next step.

V_t^{opt} is determined based on V_t, such that on each finite element K:

$$\frac{\|u^{V_t^{fine}} - u^{V_t}\|_{H^1(K)} - \|u^{V_t^{fine}} - u^{V_{t+1}}\|_{H^1(K)}}{|V_{t+1}|_K - |V_t|_K} = \min_{V_t \subset V_t^w \subseteq V_t^{fine} : |V_t^w|_K - |V_t|_K \neq 0} \frac{\|u^{V_t^{fine}} - u^{V_t}\|_{H^1(K)} - \|u^{V_t^{fine}} - u^{V_t^w}\|_{H^1(K)}}{|V_t^w|_K - |V_t|_K} \quad (11)$$

where:

- $|V|_K$ is the number of basis functions in the base of V with supports inside the boundaries of the element K,
- V_t^{fine} is the space created from V_t by the h-refinement of every single element of the mesh corresponding to the space V_t, followed by the p-refinement of all new elements.

Although the above definition refers to a solution of the problem in every single V_t^w, there is no need to solve it in all these spaces. Instead, we solve the problem on the mesh corresponding to V_t (*coarse mesh*) and on the mesh corresponding to V_t^{fine} (*fine mesh*). Then, we can project space V_t^{fine} onto any of its subspaces V_t^w over element K according to a scheme presented in [1].

Formalization of these steps results in algorithm 1 (see also [6]) for choosing an optimal space V_{t+1} for a given V_t.

Algorithm 1 Choice of an optimal mesh for the following iteration of the adaptive algorithm

$u^{V_t} \leftarrow$ solve the minimization problem in V_t (on the coarse mesh)
$V_t^{fine} \leftarrow$ refine V_t
$u^{V_t^{fine}} \leftarrow$ solve the minimization problem in V_t^{fine} (on the fine mesh)
$max\_err \leftarrow 0$
for all coarse mesh elements K **do**
  $V_t^{opt} \leftarrow V_t^{fine}$
  for all $V_t \subset V_t^w \subseteq V_t^{fine}$ over element K such that $|V_t^w|_K - |V_t|_K \neq 0$ **do**
    $u^{V_t^w} \leftarrow$ compute a projection of $u^{V_t^{fine}}$ onto V_t^w
    $err(V_t^w, K) \leftarrow \frac{\|u^{V_t^{fine}} - u^{V_t}\|_{H^1(K)} - \|u^{V_t^{fine}} - u^{V_t^w}\|_{H^1(K)}}{|V_t^w|_K - |V_t|_K}$
    if $err(V_t^w, K) < err(V_t^{opt}, K)$ **then**
      $V_t^{opt} \leftarrow V_t^w$
    end if
  end for
  add all basis functions from V_t^{opt} with supports on K to V_{t+1}
  if $err(V_t^{opt}, K) > max\_err$ **then**
    $max\_err \leftarrow err(V_t^{opt}, K)$
  end if
end for
return V_{t+1}, max_err

Algorithm 1 is performed iteratively until the stop condition is met, i.e. $\max_{K\in\Omega} err(V_t^w, K) < desired\_precission$. The complete version becomes algorithm 2.

Algorithm 2 Adaptive projection algorithm

```
err ← ∞
t ← 1
V_t ← space corresponding to a trivial initial mesh
while err > desired_precission do
    V_{t+1}, err ← perform algorithm 1 on V_t
    V_t ← V_{t+1}
    t ← t + 1
end while
return V_{t+1}
```

3.1. Polynomial sequences

The above algorithm works for any choice of a sequence of spaces V_t which have the property that $V_t \in V_{t+1}$. In our work we use the following sequence of spaces spanned over so called hierarchical base functions:

In 1D these functions are defined recursively on the base element $[0, 1]$:

$$\chi_1(\xi) = 1 - \xi \tag{12}$$
$$\chi_2(\xi) = \xi \tag{13}$$
$$\chi_3(\xi) = (1 - \xi)\xi \tag{14}$$
$$\chi_l(\xi) = (1 - \xi)\xi(2\xi - 1)^{l-3} \forall_{l=4..p+1} \tag{15}$$

In higher dimensons base functions are defined as tensor products of 1D base functions. In particular in 2D a tensor product of the above functions donates:

$$\phi_1(\xi_1, \xi_2) = \chi_1(\xi_1)\chi_1(\xi_2) \tag{16}$$
$$\phi_2(\xi_1, \xi_2) = \chi_2(\xi_1)\chi_1(\xi_2) \tag{17}$$
$$\phi_3(\xi_1, \xi_2) = \chi_2(\xi_1)\chi_2(\xi_2) \tag{18}$$
$$\phi_4(\xi_1, \xi_2) = \chi_1(\xi_1)\chi_2(\xi_2) \tag{19}$$

This group of base functions contributes to so called vertex shape functions, which have their maxima on mesh vertices.

Higher order tensor products donate edge shape functions:

$$\phi_{5,j}(\xi_1, \xi_2) = \chi_{2+j}(\xi_1)\chi_1(\xi_2) \forall_{j=1..p_1-1} \tag{20}$$
$$\phi_{6,j}(\xi_1, \xi_2) = \chi_2(\xi_1)\chi_{2+j}(\xi_2) \forall_{j=1..p_2-1} \tag{21}$$
$$\phi_{7,j}(\xi_1, \xi_2) = \chi_{2+j}(\xi_1)\chi_2(\xi_2) \forall_{j=1..p_3-1} \tag{22}$$
$$\phi_{8,j}(\xi_1, \xi_2) = \chi_1(\xi_1)\chi_{2+j}(\xi_2) \forall_{j=1..p_4-1} \tag{23}$$

and face shape functions defined as:

$$\phi_{9,i,j}(\xi_1, \xi_2) = \chi_{2+i}(\xi_1)\chi_{2+j}(\xi_2)\forall_{j=1..p_h-1}\forall_{j=1..p_v-1} \tag{24}$$

where:

- p_1, p_2, p_3, p_4 are edge adaptation orders in each direction,
- p_h, p_v are face adaptation orders horizontally and vertically (respectively).

Base functions defined this way are especially convenient in adaptive algorithms, since to refine the base one needs to simply add functions of higher orders to it and the rest of the base remains intact. This is called p-adaptation in the FEM community as opposed to h-adaptation – the process of dividing a finite element into new elements of smaller size. Both methods are widely discussed in [1] and do not constitute any part of the findings described in this research and they are not discussed here in detail. It is important, however, that combining them (into so called hp-adaptation) is required to get exponential convergence of the solution with respect to the number of degrees of freedom (i.e. functions in the base).

It is worth to mention that in practical applications intensive p-adaptation leads to uneven matrices, which tend to misbehave when supplied to iterative solvers. For this reason, direct solvers such as [7] are usually used for adaptive algorithms instead.

4. Numerical experiments

As a proof of concept we tested our solution on a few bitmaps representing selected microstructures described below. Three kinds of digital material representations have been created for the purpose of the current experiment. The first one depicts a commonly used simplified representation of a two phase microstructure. Using this approach, an influence of different volume fractions of hard martensitic phase in soft ferritic matrix can be investigated. The second one represents a single phase polycrystalline microstructure, where subsequent grains are clearly distinguished. The third one represents a multi-layer composite material.

To properly capture grain morphology with homogenous FE meshes, very fine finite element meshes have to be used. This, in turn, leads to excessive computational time. The finer the FE mesh is, the better description of the phase boundary shape is obtained. With coarser meshes some microstructure features can be even neglected. However, to reduce computational time and maintain high accuracy of the solution along mentioned phase/grain boundaries, proposed hp adaptation technique can be used to obtain specific non-uniform FE meshes that are refined along the phase/grain boundaries.

4.1. Preprocessing of the images

Since the described algorithm operates over $[0, 1] \times [0, 1]$ square, each bitmap required some extra preprocessing. We converted *.bmp* files into input *.dat* files by performing the following steps:

- Since the value of material data is encoded in the bitmap's luma Y, we obtain it according to the formula below:

$$Y = 0.299 \times Red + 0.587 \times Green + 0.114 \times Blue \tag{25}$$

  This kind of encoding is frequently used, as it makes *visually neighbouring* pixels to be *neighbours* also numerically.
- Normalize the output value set to the interval $[0, 1]$.
- Save bitmap as a 2D matrix of floating-point numbers.
- Compute value of $f(x, y) : (x, y) \in [0, 1] \times [0, 1]$ by scaling and mapping (x, y) to the nearest pixel on the bitmap.
- Compute value of $\frac{\partial f(x,y)}{\partial x}$ and $\frac{\partial f(x,y)}{\partial y}$ using differential quotient.

4.2. Discussion of the results

The hp-adaptive algorithm is executed for 20 iterations until the error decrese rate falls below the desired 1%. Behaviour of the mesh after 10 and 20 iterations for all three cases has been depicted in Figures 4, 5 and 6. In the first case, adaptation focuses on 5 squares surrounding the distinct pieces of a different material. As the adaptation proceeds, the interface between materials becomes visible more clearly.

The refinement of meshes is also reflected in the solutions. The results after 20 iterations for the second and third microstructure have been presented in Figures 7 and 4.2. These figures may be thought of as normalized continuous representations of the initial microstructure images (Figure 3).

Figure 3. Microstructure images used for computations.

For the first bitmap we obtained an acceptable solution only after 10 iterations. This is presented on Figure 7. All microstructure images passed to the algorithm had a few fetures which made them especially tricky to interpolate continuously – namely irregular shapes, sharp edges and a wide range of values. While shapes turned out to be reflected in quite a faithful manner with h-adaptation, sharp edges required elevated degrees of p-adaptation. This, by nature, leads to the Runge effect, where

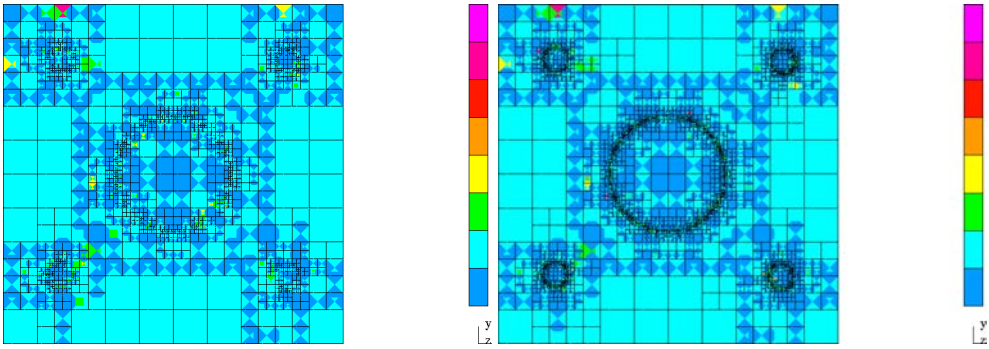


Figure 4. Mesh density and p-refinement levels after 10 and 20 iterations for the first microstructure.

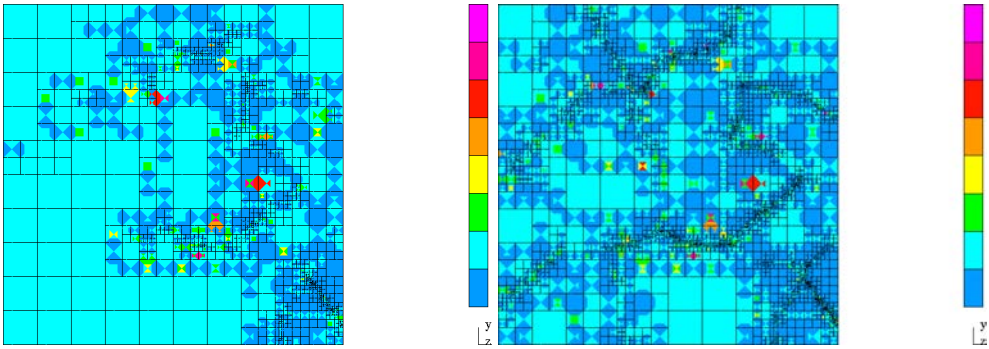


Figure 5. Mesh density and p-refinement levels after 10 and 20 iterations for the second microstructure.

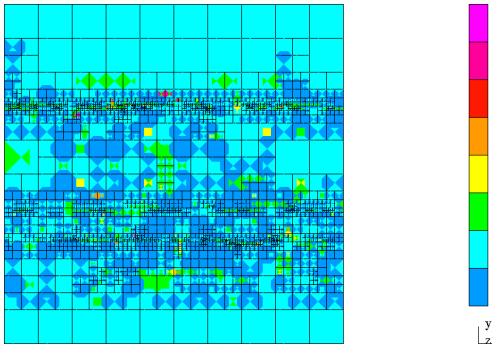


Figure 6. Mesh density and p-refinement levels for the third microstructure after 30 iterations.

interpolant oscillates at the edges of an interval. To avoid this effect, one should supply the upper bound for p-adaptation level in some applications.

The obtained meshes can be used as an input for the FE modeling of material behavior during processing or exploitation stages. We tested the feasibility of this application using examples 2 and 3, for which we consider the heat transfer and linear elasticity problems respectively. In general, the applicability of the grids for the finite element method simulations depends on the sensitivity of the physical phenomena to the changes of material properties.

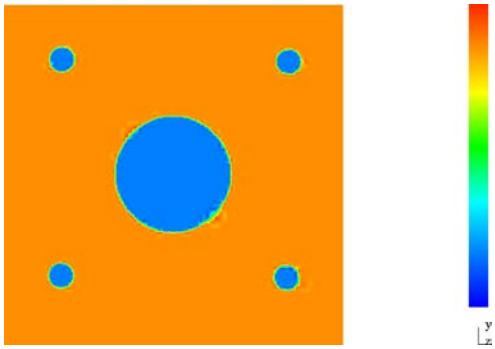


Figure 7. The approximation of material data after 10 iterations for the first microstructure.

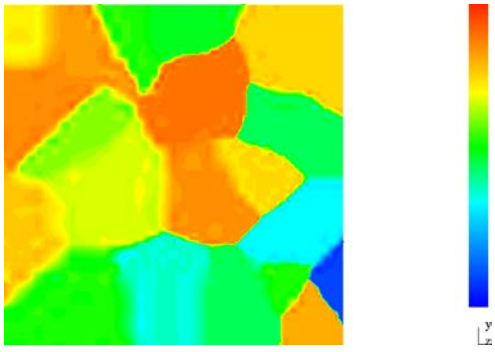


Figure 8. The approximation of material data after 20 iterations for the second microstructure.

First test concerned the heat transfer problem. We solved the heat transfer equation over the second mesh with Dirichlet boundary condition on the bottom and Neuman boundary condition on top, left and right sides. We assumed that the heat transfer coefficient K changes for different material. We also assumed that the temperature over the Dirichlet boundary varies with different materials. From the numerical results presented in Figures 7 and 4.2 it follows that the heat transfer is not sensitive to the changes in material data. The adaptivity are only necessary for the Dirichlet boundary.

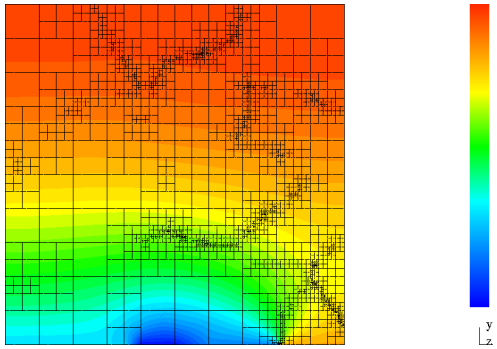


Figure 9. The solution to the heat transfer problem over that mesh (temperature scalar field).

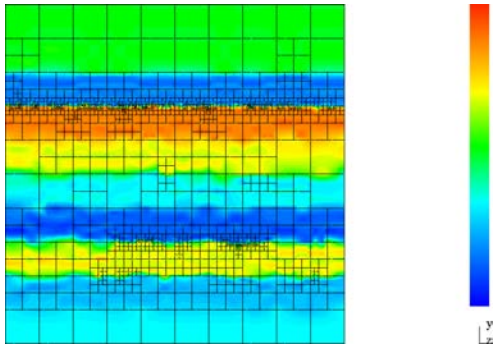


Figure 10. The approximation of material data after 20 iterations for the third microstructure.

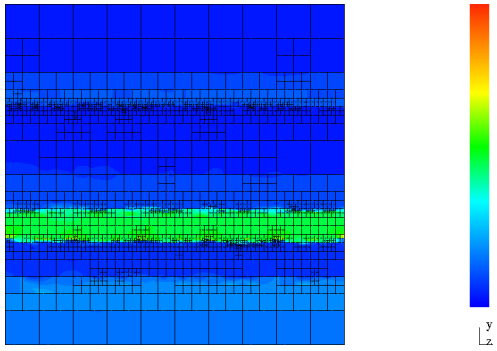


Figure 11. The solution to the linear elasticity problem (norm of the displacement vector) over a further refined mesh.

The second numerical problem concerned the linear elasticity. We subjected the third multi-layered mesh to the crushing forces. Namely, we assumed fixed zero Dirichlet boundary conditions on the bottom, the crushing force on the top, with zero Neumann boundary conditions on both sides. We assumed different Young modulus for different layers denoted by different colors. Our adaptive procedure has improved the accuracy of the linear elasticity problem significantly. The linear elasticity problem solved on the uniform unrefined grid suffers from 50 percent numerical error (measured as relative error in H1 norm). The same problem solved over our hp adapted mesh (Figures 4.2 and 11) delivers 10 percent numerical error. In other words our adaptive procedure has reduced the numerical error almost one order of magnitude.

4.3. Convergence evaluation

In Figure 12 we present convergence curves for the first and the second case.

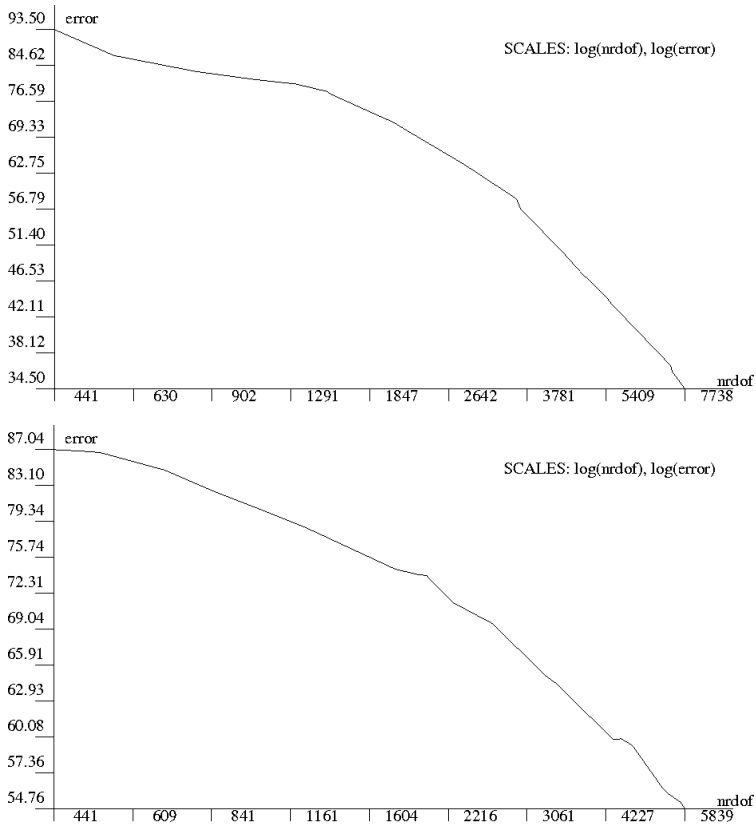


Figure 12. Convergence curves, nrdof – number of degrees of freedom, error – absolute error decrease rate in terms of H^1 norm.

Since they are drawn in logarithmic scale, error decrease rate is virtually exponential with respect to the number of degrees of freedom (marked as "nrdof"). In both cases it took the similar amount of required steps to obtain a satisfactory solution.

5. Conclusions and future work

This paper presents a way of incorporating a well known H^1 projection concept into an adaptive algorithm used to prepare material data. The described method allows for generation of a smooth, continuous interpolation of given arbitrary data, along with an initial pre-adapted mesh suitable for further processing by an FE solver. Thanks to combining h- and p- adaptivity it was possible to observe exponential convergence with the number of degrees of freedom, as predicted by the theory.

We identify several areas for development. The most obvious one is generalization of this project to 3D. Apart from this, it is desirable to experiment with more bitmaps and various input parameters (e.g. boundary conditions or image conversion algorithms). Besides, more sophisticated digital material representations should be investigated (e.g. complex two phase or triple phase microstructures that are typical in automotive industry). Feasibility of applying the hp adaptation technique to the evolving morphology of microstructure could be evaluated as well.

Acknowledgements

The work of the first and second authors have been supported by Polish National Science Centre funds based on the decision number DEC-2011/03/N/ST6/01397.

The work of the third author has been supported by Polish grant no. 820/N-Czechy 2010/0.

The work of the first author was partly supported by The European Union by means of European Social Fund, PO KL Priority IV: Higher Education and Research, Activity 4.1: Improvement and Development of Didactic Potential of the University and Increasing Number of Students of the Faculties Crucial for the National Economy Based on Knowledge, Subactivity 4.1.1: Improvement of the Didactic Potential of the AGH University of Science and Technology "Human Assets", No. UDA POKL.04.01.01-00-367/08-00.

References

[1] Demkowicz L.: *Computing With Hp-adaptive Finite Elements. Vol. 1: One and Two Dimensional Elliptic and Maxwell Problems.* Chapman & Hall CRC, Texas, 2006.

[2] Demkowicz L. F.: *Computing with hp-ADAPTIVE FINITE ELEMENTS: Volume 1 One and Two Dimensional Elliptic and Maxwell problems.* CRC Press, United States of America, 2007.

[3] Gurgul P., Sieniek M., Magiera K., Skotniczny M.: Application of multi-agent paradigm to hp-adaptive projection-based interpolation operator. *Journal of Computational Science*, 2011.

[4] Madej L., Cybulka P., Perzyński K., Rauch L.: Numerical analysis of strain inhomogeneities during deformation on the basis of the three dimensional digital material representation. *Computer Methods in Materials Science*, 10:375–380, 2011.

[5] Madej L., Rauch L., Perzyński K., Cybulka P.: Digital material representation as an efficient tool for strain inhomogeneities analysis at the micro-scale level. *Archives of Civil and Mechanical Engineering*, 11:661–679, 2011.

[6] Paszyński M.: *Graph Grammar-Driven Parallel Adaptive PDE Solvers*. Uczelniane Wydawnictwa Naukowo-Dydaktyczne AGH, Krakow, 2009.

[7] Paszyński M., Pardo D., Torres-Verdin C., Demkowicz L., Calo V.: A parallel direct solver for self-adaptive hp finite element method. *Journal of Parallel and Distributed Computing*, 70(3):270–281, 2010.

[8] Paszyński M., Romkes A., Collister E., Meiring J., Demkowicz L.: On the modeling of step-and-flash imprint lithography using molecular statics models. *CG Willson ICES Report*, pp. 05–38, 2005.

[9] Sieniek M., Gurgul P., Skotniczny M., Magiera K., Paszyński M.: Agent-oriented image processing with the hp-adaptive projection-based interpolation method. *Procedia Computer Science*, 4(1):1844–1853, 2011.

Affiliations

Piotr Gurgul
AGH University of Science and Technology, Krakow, Poland

Marcin Sieniek
AGH University of Science and Technology, Krakow, Poland

Maciej Paszyński
AGH University of Science and Technology, Krakow, Poland, `maciej.paszynski@agh.edu.pl`

Łukasz Madej
AGH University of Science and Technology, Krakow, Poland

Nathan Collier
King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

Received: 11.05.2012
Revised: 6.08.2012
Accepted: 3.12.2012