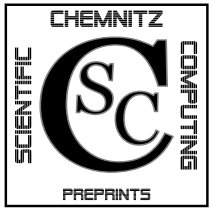
TECHNISCHE UNIVERSITÄT CHEMNITZ

René Schneider

With a new refinement paradigm towards anisotropic adaptive FEM on triangular meshes

CSC/13-02



Chemnitz Scientific Computing Preprints

Chemnitz Scientific Computing Preprints – ISSN 1864-0087

Impressum:

Chemnitz Scientific Computing Preprints — ISSN 1864-0087
(1995–2005: Preprintreihe des Chemnitzer SFB393)

Herausgeber:
Professuren für
Numerische und Angewandte Mathematik
an der Fakultät für Mathematik
der Technischen Universität Chemnitz

Postanschrift:
TU Chemnitz, Fakultät für Mathematik
09107 Chemnitz
Sitz:
Reichenhainer Str. 41, 09126 Chemnitz

http://www.tu-chemnitz.de/mathematik/csc/

TECHNISCHE UNIVERSITÄT CHEMNITZ

Chemnitz Scientific Computing Preprints

René Schneider

With a new refinement paradigm towards anisotropic adaptive FEM on triangular meshes

CSC/13-02

Abstract

Adaptive anisotropic refinement of finite element meshes allows to reduce the computational effort required to achieve a specified accuracy of the solution of a PDE problem. We present a new approach to adaptive refinement and demonstrate that this allows to construct algorithms which generate very flexible and efficient anisotropically refined meshes, even improving the convergence order compared to adaptive isotropic refinement if the problem permits.

CSC/13-02 ISSN 1864-0087 September 2013

Contents

1 Introduction **1**

2 Extension of FEM ansatz spaces **2**

3 Optimality of the extension **4**

4 Application 1: graded refinement **6**

5 Application 2: anisotropic refinement in 2D **8**
5.1 Edge error indicator 9
5.2 Edge oriented anisotropic refinement 12
5.2.1 Red-green-blue-refinement 15
5.2.2 Mesh reconnection by edge swapping 17
5.2.3 Mesh coarsening by edge collapsing 18
5.2.4 Anisotropic adaptive algorithm 20

6 Numerical experiments **21**
6.1 Poisson equation on the unit-square 21
6.2 Poisson equation on an L-shape domain 32
6.3 Reaction-Diffusion 34

7 Conclusions and outlook **35**

[25] X. Li, M.S. Shephard, and M.W. Beall. 3d anisotropic mesh adaptation by mesh modification. *Computer Methods in Applied Mechanics and Engineering*, 194:4915–4950, 2005.

[26] R. Mahmood and P.K. Jimack. Locally optimal unstructured finite element meshes in 3 dimensions. *Computers and Structures*, 82(23–26):2105–2116, 2004.

[27] W.F. Mitchell. Optimal multilevel iterative methods for adaptive grids. *SIAM Journal Scientific Computing*, 13(1):146–167, 1992.

[28] M. Picasso, F. Alauzet, H. Borouchaki, and P.-L. George. A numerical study of some Hessian recovery techniques on isotropic and anisotropic meshes. *SIAM Journal Scientific Computing*, 33(3):1058–1076, 2011.

[29] T. Richter. A posteriori error estimation and anisotropy detection with the dual-weighted residual method. *International Journal for Numerical Methods in Fluids*, 62(1):90–118, 2010.

[30] H.-G. Roos, M. Stynes, and L. Tobiska. *Robust Numerical Methods for Singularly Perturbed Differential Equations*, volume 24 of *Springer Series in Computational Mathematics*. Springer, second edition, 2008.

[31] S. Beuchler and A. Meyer. SPC-PM3AdH v1.0 - Programmer's Manual. Technical Report Preprint SFB393/01-08, TU Chemnitz, Chemnitz, 2001. Available at http://www.tu-chemnitz.de/sfb393/.

[32] R. Schneider. A review of anisotropic refinement methods for triangular meshes in fem. In T. Apel and O. Steinbach, editors, *Advanced Finite Element Methods and Applications*, volume 66 of *Lecture Notes in Applied and Computational Mechanics*, pages 133–152. Springer Berlin Heidelberg, 2013.

[33] D. Wang, R. Li, and N. Yan. An edge-based anisotropic mesh refinement algorithm and its application to interface problems. *Communications in Computational Physics*, 8(3):511–540, 2010.

1 Introduction

Adaptive finite element meshes are well established as a tool to efficiently compute solutions to many PDE problems, e.g. [2, 7]. For many application problems the solution features that are to be approximated by adaptive refinement are of anisotropic character, i.e. these solutions change rapidly in one spatial direction, but slowly in another. In these cases the most efficient approximations can be achieved on anisotropically refined meshes, where the mesh length scales in different directions match the solution behaviour, e.g. [4]. Even though this is well known, the majority of adaptive finite element codes exclude anisotropic meshes.

Most of the recent research into automatic adaptive anisotropic refinement has separated the error estimation, computation of desired mesh length scales and the actual refinement, e.g. [11, 16, 21, 28]. This has been enabled by the description of desired mesh anisotropy via an anisotropic mesh metric and mesh generation software that is able to construct meshes conforming with a given metric field [9, 20]. In each step of such an adaptive procedure the whole mesh is replaced by a new mesh, thus no hierarchy information is retained that could be used in multigrid or multilevel solution techniques for the linear systems.

For quadrangles there are several anisotropic refinement approaches based on splitting an initial quadrangle into two, thereby doubling or halving the aspect ratio [5, 29, 31]. In such a refinement strategy one can keep track of the mesh hierarchy and utilise it in the linear solvers. However, anisotropy can only be generated with stretching directions predefined by the initial mesh, realignment with (arbitrary) directions of solution features is not possible.

For this reason several authors attempt to generalise this procedure to triangular meshes (e.g. [1, 18, 33]), which offer greater geometric flexibility. By bisecting a triangle at one of its edges, one can at least choose the direction in which the aspect ratio is increased among three edges of the triangle. But this is not quite as simple as it might look at first sight. While this kind of refinement does increase anisotropy, it can not achieve the asymptotic behaviour of unidirectional refinement [32, Section 4.3], which is ignored by most of the authors.

In the literature dealing with anisotropic triangular meshing based on mesh metrics, this limitation is overcome by combining element bisection with other local mesh modification operations. In this work we will propose a similar strategy, but without consideration of a mesh metric.

Mesh metrics (as generalised mesh diameters) are a paradigm in the understanding of the finite element method. However, in our view the mesh metric is only a useful tool in characterising (anisotropic) meshes for a priori analysis of methods. It is not a natural requirement of the finite element method itself. This method can be formulated and applied purely based on a mesh of the domain. Element diameters h and mesh metrics are only one tool to analyse the behaviour of the method. This analysis tells us that the approximation error is reduced if the element diameter (in certain directions) is reduced enough, because an upper bound is reduced.

We hope that a more direct approach to modify the mesh in a way that is guaranteed to reduce the error will overcome artificial limitations which may be implied by the mesh metric view of the finite element method. This may enable a different and hopefully better perspective, especially in the context of anisotropic finite elements.

Thus, in this work we propose as the new paradigm for adaptive mesh refinement to consider the effects of individual mesh modifications in terms of their effect on the solution itself, without intermediate steps such as a mesh metric, or at least not restricting to this one tool but considering alternatives.

From this new paradigm we derive new anisotropic adaptive refinement algorithms, which are demonstrated to achieve improved convergence order if the solution has lower dimensional features. They are also demonstrated to perform very well for a class of examples of singularly perturbed reaction diffusion equations. Unfortunately the proposed criterion that is very successful in guiding this anisotropic refinement turns out not to be reliable enough as the sole guide for judging the quality of a given finite element solution, which will provide us with opportunities for future research.

The main part of this paper is organised as follows. We propose a new way of looking at the mesh refinements as extensions to the finite element ansatz space in Section 2. Resulting from this we get a way to chose "best extensions" from the available extensions. In Section 3 we analyse in what way these are "best extensions". In Sections 4 and 5 we consider two possibilities how this can be exploited to effect automatic adaptive anisotropic mesh refinement, where the methods form Section 5 appears to offer better perspectives for practical application. The merits of the resulting proposed refinement algorithms are then studied in Section 6 on a suitable set of example problems. Conclusions and an outlook are given in Section 7.

2 Extension of FEM ansatz spaces

Let us consider the genaral, abstract weak formulation of a PDE problem

$$a(u, v) = b(v) \qquad \forall v \in \mathbb{V} \tag{1}$$

and the corresponding Galerkin discretisation

$$a(u_h, v_h) = b(v_h) \qquad \forall v_h \in \mathbb{V}_h, \tag{2}$$

where $\mathbb{V}$ denotes the function space associated with the PDE problem, and $\mathbb{V}_h \subset \mathbb{V}$ the finite dimensional ansatz space of the discretisation, e.g. the finite element function space, $a(.,.)$ is a bilinear form on $\mathbb{V}$ and $b(.)$ a linear functional on $\mathbb{V}$.

The aim of adaptive mesh refinement can be interpreted as seeking a ansatz space $\mathbb{V}_h$, such that for the solution of the discrete problem (2) the error in some

[13] V. Dolejsi. Anisotropic mesh adaptation for finite volume and finite element methods on triangular meshes. *Computing and Visualisation in Science*, 1:165–178, 1998.

[14] W. Dörfler and V. Heuveline. Convergence of an adaptive hp finite element strategy in one space dimension. *Applied Numerical Mathematics*, 57:1108–1124, 2007.

[15] H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers*. Oxford University Press, 2005.

[16] L. Formaggia, S. Micheletti, and S. Perotto. Anisotropic mesh adaptation in computational fluid dynamics: Application to the advection-diffusion-reaction and the Stokes problems. *Applied Numerical Mathematics*, 51(4):511–533, December 2004.

[17] M. Fortin. Anisotropic mesh adaptation through hierarchical error estimators. In P. Minev and Y. Lin, editors, *Scientific computing and applications*, volume 7, pages 53–65. Nova Science Publishers, 2001.

[18] Sergey Grosman. *Adaptivity in Anisotropic Finite Element Calculations*. PhD thesis, TU-Chemnitz, Chemnitz, Germany, 2006.

[19] W.G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M.-G. Vallet. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independant cfd. part i: general principles. *International Journal for Numerical Methods in Fluids*, 32:725–744, 2000.

[20] F. Hecht. Bidimensional anisotropic mesh generator. Technical report, INRIA, Rocquencourt, 1997. Software: http://www.ann.jussieu.fr/hecht/ftp/bamg/.

[21] W. Huang, L. Kamenski, and J. Lang. A new anisotropic mesh adaptation method based upon hierarchical a posteriori error estimates. *Journal of Computational Physics*, 229(6):2179–2198, 2010.

[22] D.W. Kelly, J.P. de S.R. Gago, O.C. Zienkiewicz, and I. Babuska. A posteriori error analysis and adaptive processes in the finite element method: Part i — error analysis. *International Journal for Numerical Methods in Engineering*, 19:1593–1619, 1983.

[23] R. Kornhuber and R. Roitzsch. On adaptive grid refinement in the presence of internal or boundary layers. *Impact of Computing in Science and Engineering*, 2:40–72, 1990.

[24] G. Kunert and R. Verfürth. Edge residuals dominate a posteriori error estimates for linear finite element methods on anisotropic triangular and tetrahedral meshes. *Numerische Mathematik*, 86:283–303, 2000.

References

[1] J.C. Aguilar and J.B. Goodman. Anisotropic mesh refinement for finite element methods based on error reduction. *Journal of Computational and Applied Mathematics*, 193:497–515, 2006.

[2] M. Ainsworth and J.T. Oden. *A Posteriori Error Estimation in Finite Element Analysis*. Wiley, 2000.

[3] R.E. Bank amd R.K. Smith. A posteriori error estimates based on hierarchical bases. *SIAM Journal on Numerical Analysis*, 30(4):921–935, 1993.

[4] T. Apel. *Anisotropic Finite Elements: Local Estimates and Applications*. Teubner, Leipzig, 1999.

[5] T. Apel, S. Grosman, P.K. Jimack, and A. Meyer. A new methodology for anisotropic mesh refinement based upon error gradients. *Applied Numerical Mathematics*, 50:329–341, 2004.

[6] T. Apel and S. Nicaise. The finite element method with anisotropic mesh grading for elliptic problems in domains with corners and edges. *Mathematical Methods in the Applied Sciences*, 21(6):519–549, 1998.

[7] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Birkhäuser Verlag, 2003.

[8] E. Bänsch. Local mesh refinement in 2 and 3 dimensions. *Impact of Computing in Science and Engineering*, 3:181–191, 1991.

[9] H. Borouchaki, P.L. George, F. Hecht, P. Laug, and E. Saltel. Delaunay mesh generation governed by metric specifications. part i. algorithms. *Finite Elements in Analysis and Design*, 25:61–83, 1997. doi:10.1016/S0168-874X(96)00057-1.

[10] M. Bürg and W. Dörfler. Convergence of an adaptive hp finite element strategy in higher space-dimension. *Applied Numerical Mathematics*, 61:1132–1146, 2011.

[11] W. Cao. On the error of linear interpolation and the orientation, aspect ratio, and internal angles of a triangle. *SIAM Journal on Numerical Analysis*, 43:19–40, 2005.

[12] J.P. de S.R. Gago, D.W. Kelly, O.C. Zienkiewicz, and I. Babuska. A posteriori error analysis and adaptive processes in the finite element method: Part ii — adaptive mesh refinement. *International Journal for Numerical Methods in Engineering*, 19:1621–1656, 1983.

appropriate norm is below a certain tollerance $\mathrm{tol} > 0$,

$$|||u - u_h||| \leq \mathrm{tol}, \tag{3}$$

keeping the dimension of $\mathbb{V}_h$ as low as possible.

As discussed in Section 1, first we will restrict ourselves to construct $\mathbb{V}_h$ only by successive refinement of an initial space $\mathbb{V}_h^{(0)}$, other modifications of $\mathbb{V}_h$ will be discussed later. A refinement is the addition of a set of linearly independent basis functions $\{v_{n+1}, \ldots, v_{n+\ell}\}$, to a given basis of an ansatz space $\mathbb{V}_h^{(k)}$ in order to define the refined ansatz space $\mathbb{V}_h^{(k+1)}$, thus increasing the dimension of $\mathbb{V}_h$ by ℓ. These new basis functions should obviously satisfy

$$v_i \in \mathbb{V} \quad \text{and} \quad v_i \notin \mathbb{V}_h^{(k)} \qquad\qquad \forall i = n+1, \ldots, n+\ell.$$

Which basis functions can be added is usually restricted by the construction method of the ansatz spaces and dependent on the current space $\mathbb{V}_h^{(k)}$, thus we define the set of possible basis functions for the refinement as $\mathbb{S}_h^{(k)+} \subset \mathbb{V}$, which is usually a finite set of linearly independent functions, and replace the first of the above requirements by $v_i \in \mathbb{S}_h^{(k)+}$.

Now, in order to keep the dimension of $\mathbb{V}_h$ during the refinement process as low as possible, one may consider adding only one new basis function in each refinement step ($\ell = 1$). While this makes little sense for most applications, because one is usually also interested in keeping the number of refinement steps low due to the associated cost of each step, we shall explore this scenario of adding only one function per step in order to guide the selection of good functions for the refinement.

In order that one of the possible new ansatz functions $\tilde{v} \in \mathbb{S}_h^{(k)+}$ makes any difference to the solution of the discrete problem (2), it has to satisfy

$$a(u_h, \tilde{v}) \neq b(\tilde{v}),$$

i.e. as a test-function it has to emphasise the error in the equation (2) in comparison to (1).

In this context it makes sense to choose $\tilde{v} \in \mathbb{S}_h^{(k)+}$ such that it maximises the residual,

$$\frac{|a(u_h, \tilde{v}) - b(\tilde{v})|}{|||\tilde{v}|||} \to \max_{\tilde{v} \in \mathbb{S}_h^{(k)+}}. \tag{4}$$

We propose to use (4) as the guiding principle for the adaptive refinement of the ansatz spaces. It's main advantage is that it can be applied to a very large class of problems, as it only requires the weak form (1) of the PDE, the Galerkin discretisation (2) and an appropriate norm $|||.|||$.

The aim of this paper is an initial evaluation of the possibilities and properties of this approach. To simplify this, for the rest of this paper we restrict ourselves to the case where the following assumptions are met.

Assumption 1. *The bilinear form $a(.,.)$ be symmetric, continuous and coercive, with $|||.|||$ denoting the associated energy norm $|||v||| := \sqrt{a(v,v)}$.*

In principle (4) should also be applicable in a more general setting. However, this shall be left to follow up studies.

In the following sections we shall first further justify this choice and explore its merits on an abstract level, before we discuss possible mesh refinement strategies arising from this idea.

3 Optimality of the extension

In [2, equation (6.4)] we find the following characterisation of the error in the energy norm

$$|||u - u_h||| = \sup_{0 \neq \tilde{v} \in \mathbb{V}} \frac{|a(u_h, \tilde{v}) - b(\tilde{v})|}{|||\tilde{v}|||}, \tag{5}$$

which can be shown to hold under Assumption 1. In fact, under these conditions the sup in (5) is achieved for all $\tilde{v} = \lambda(u - u_h)$, $\lambda \in \mathbb{R}$. Thus if we were able to add $e_h := u - u_h$ to the ansatz space, the refinement would reduce the error to zero in one step. Of course, solving this maximisation problem is equivalent to solving the PDE (1) itself, so this is not a practical option.

In a realistic setting, the following lemma justifies (4) as criterion.

Lemma 1. *Let Assumption 1 hold and let a set $\mathbb{S}_* \subset \mathbb{V}$ of possible refinement functions be given. For arbitrary $\tilde{v} \in \mathbb{S}_*$ let $\widetilde{\mathbb{V}}_h := \mathbb{V}_h \cup \mathrm{span}(\tilde{v})$ denote the ansatz space extended by the span of the function $\tilde{v}$, and let $\tilde{u}_h \in \widetilde{\mathbb{V}}_h$ be the corresponding solution of the Galerkin discretisation (2) on $\widetilde{\mathbb{V}}_h$. Then it holds for the errors*

$$|||u - \tilde{u}_h|||^2 \leq |||u - u_h|||^2 - \left[\frac{|a(u_h, \tilde{v}) - b(\tilde{v})|}{|||\tilde{v}|||}\right]^2.$$

Proof. Consider $\tilde{v} \in \mathbb{S}_*$ fixed, $c \in \mathbb{R}$ variable, defining $\tilde{u}_h^* := u_h - c\,\tilde{v}$. Due to the best-approximation property of the FEM (see e.g. [15, Theorem 1.3]) we have

$$\begin{aligned} |||u - \tilde{u}_h|||^2 &\leq |||u - \tilde{u}_h^*|||^2 \\ &= a(u - u_h + c\,\tilde{v}, u - u_h + c\,\tilde{v}) \\ &= |||u - u_h|||^2 + 2c\,a(u - u_h, \tilde{v}) + c^2 a(\tilde{v}, \tilde{v}). \end{aligned}$$

Now choose c such that the righthand side on the last line is minimized with respect to c. The necessary and sufficient condition for this is $2\,a(u - u_h, \tilde{v}) + 2ca(\tilde{v}, \tilde{v}) = 0$, thus

$$c = -\frac{a(u - u_h, \tilde{v})}{a(\tilde{v}, \tilde{v})}.$$

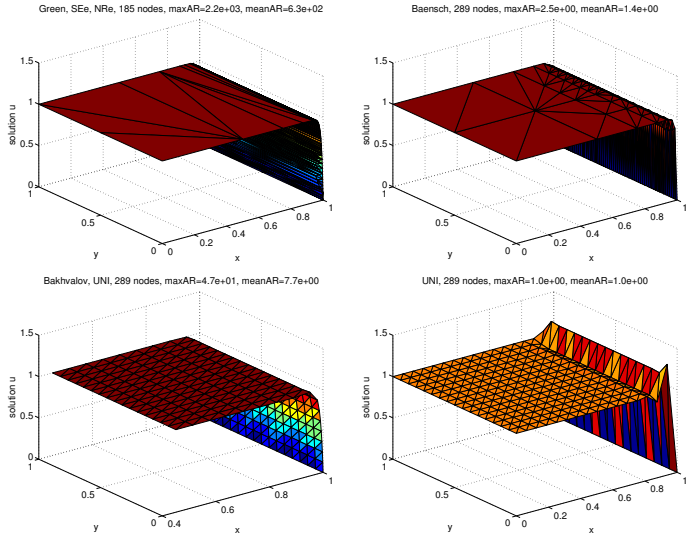


Figure 24: FE solution on meshes resulting from different refinement algorithms for Example 6 with $\kappa = 10^4$

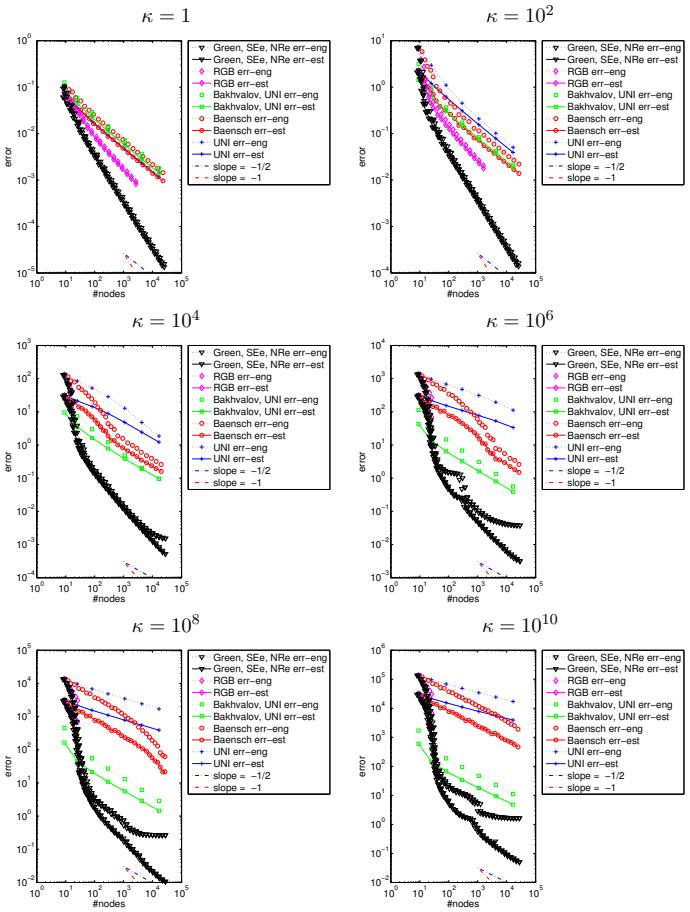


Figure 23: Comparison of different refinement algorithms for Example 6: error and indicator for $\kappa \in \{1;\ 10^2;\ 10^4;\ 10^6;\ 10^8;\ 10^{10}\}$

This specific choice gives

$$\begin{aligned}
|||u - \tilde{u}_h|||^2 &\leq |||u - \tilde{u}_h^*|||^2 \\
&= a(u - u_h + c\,\tilde{v}, u - u_h + c\,\tilde{v}) \\
&= |||u - u_h|||^2 - 2\frac{[a(u - u_h, \tilde{v})]^2}{a(\tilde{v}, \tilde{v})} + \frac{[a(u - u_h, \tilde{v})]^2}{a(\tilde{v}, \tilde{v})} \\
&= |||u - u_h|||^2 - \left[\frac{a(u - u_h, \tilde{v})}{|||\tilde{v}|||}\right]^2 \\
&= |||u - u_h|||^2 - \left[\frac{b(\tilde{v}) - a(u_h, \tilde{v})}{|||\tilde{v}|||}\right]^2 .
\end{aligned}$$

□

Note that the error is reduced by at least the quantity which is maximised in (4). Thus, the selection principle (4) minimises this upper bound on the error in each step.

If in addition we require $a(.,.)$ orthogonality of $\mathbb{S}_*$ to $\mathbb{V}_h$, we get the following optimality result.

Lemma 2 (Optimal error reduction)**.** *Let Assumption 1 hold and $\mathbb{S}_* \subset \mathbb{V}$ be either a finite set of linear independent functions, or a compact set. Let this $\mathbb{S}_*$ be $a(.,.)$-orthogonal to $\mathbb{V}_h$, i.e. $a(\mathbb{S}_*, \mathbb{V}_h) = 0$, and let $\tilde{v}$ be a global maximiser*

$$\tilde{v} = \arg\max_{v \in \mathbb{S}_*} \frac{|a(u_h, v) - b(v)|}{|||v|||}.$$

Then the extended ansatz space $\widetilde{\mathbb{V}}_h := \mathbb{V}_h \cup \operatorname{span}(\tilde{v})$ with the resulting Galerkin solution $\tilde{u}_h \in \widetilde{\mathbb{V}}_h$ fulfills the best approximation property

$$|||\tilde{e}||| := |||u - \tilde{u}_h||| = \min_{v_* \in \mathbb{S}_*} \left(\min_{v \in \mathbb{V}_h \cup \operatorname{span}(v_*)} |||u - v||| \right). \tag{6}$$

Proof. For any $v_* \in \mathbb{S}_*$ consider an arbitrary $u_* \in \mathbb{V}_h \cup \operatorname{span}(v_*)$. This can be represented as

$$u_* = v_h + c\,v_*,$$

with a $v_h \in \mathbb{V}_h$ and $c \in \mathbb{R}$. Since $a(\mathbb{S}_*, \mathbb{V}_h) = 0$ we get

$$\begin{aligned}
|||u - u_*|||^2 &= a(u - v_h + c\,v_*, u - v_h + c\,v_*) \\
&= |||u - v_h|||^2 + 2\,c\,a(u - v_h, v_*) + c^2\,|||v_*|||^2 \\
&= |||u - v_h|||^2 + 2\,c\,a(u, v_*) - 2\,c\,\underbrace{a(v_h, v_*)}_{=0} + c^2\,|||v_*|||^2 \\
&= |||u - v_h|||^2 + 2\,c\,a(u, v_*) + c^2\,|||v_*|||^2.
\end{aligned}$$

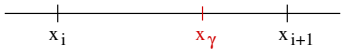


Figure 1: Graded refinement of an edge, with $x_\gamma = (1-\gamma)\, x_i + \gamma\, x_{i+1}$, $\gamma \in (0,1)$.

In order to find the min in the parentheses in (6), for any given $v_* \in \mathbb{S}_*$, we can let

$$|||u - v_h|||^2 \to \min_{v_h \in \mathbb{V}_h} \qquad \text{and} \qquad 2\, c\, a(u, v_*) + c^2\, |||v_*|||^2 \to \min_{c \in \mathbb{R}}$$

independently. These minima are attained for

$$v_h = u_h \qquad \text{and} \qquad c = -\frac{a(u, v_*)}{|||v_*|||^2},$$

giving

$$\begin{aligned} \min_{v \in \mathbb{V}_h \cup \operatorname{span}(v_*)} |||u - v||| &= |||u - u_h|||^2 - 2\frac{(a(u, v_*)\,)^2}{|||v_*|||^2} + \frac{(a(u, v_*)\,)^2}{|||v_*|||^2} \\ &= |||u - u_h|||^2 - \frac{(a(u, v_*)\,)^2}{|||v_*|||^2}. \end{aligned}$$

The term $|||u - u_h|||^2$ is independent of v_*, thus the overall minimum on the right hand side of (6) is taken for $v_* = \widetilde{v}$.

Finally, since $\widetilde{u}_h$ is the unique minimiser of $\min_{v \in \mathbb{V}_h \cup \operatorname{span}(v_*)} |||u - v|||$, equation (6) follows. □

Next we will discuss possible applications of the proposed refinement selection.

4 Application 1: graded refinement

We consider the P_1 FE discretisation of a 1D Dirichlet problem of the Poisson equation:

Example 1.

$$\begin{aligned} -u''(x) &= f(x) && \text{on } \Omega := (0,1), \\ u(0) = u(1) &= 0. \end{aligned}$$

One element shall be refined, but we allow the refinement into two non-equal sized children, where a parameter $\gamma \in (0,1)$ defines the proportions, see Figure 1. Depending on the right-hand-side function f, different values of γ are optimal, as illustrated in Figure 2 for the splitting of the interval $(0,1)$ itself.

the less, this algorithm outperforms isotropic adaptive refinement by several orders of magnitude over large ranges in these experiments, and even outperforms the Bakhvalov meshes.

7 Conclusions and outlook

The proposed approach allows to introduce substantial anisotropy into the mesh and to re-align the mesh with solution features. In particular the results for the reaction diffusion problems show that significant improvements compared to isotropic refinement can be achieved.

However, the approach has to be refined in order to guarantee convergence, since the sum of error indicators (8) is not reliable as error estimator, as demonstrated by the numerical experiments.

It appears promising to combine the approach with some more sophisticated error estimation technique to overcome this limitation. I.e. isotropic (red) refinement could be applied where green refinement is not sufficiently beneficial. Another possibility may be to evaluate (4) in a different way, taking more general or combined mesh modifications into consideration. This, however, shall be the topic of future research.

6.3 Reaction-Diffusion

Example 6. *Consider the reaction-diffusion roblem*

$$\begin{aligned} -\Delta u + \kappa u &= \kappa && \text{on } \Omega := (0,1)^2, \\ u &= 0 && \text{on } \Gamma_1 := \{1\} \times [0,1], \\ u &= 1 - \exp(-\sqrt{\kappa}) && \text{on } \Gamma_2 := \{0\} \times [0,1], \\ \frac{\partial u}{\partial n} &= 0 && \text{on } \Gamma_N := \partial\Omega \setminus \big(\Gamma_1 \cup \Gamma_2\big). \end{aligned}$$

The unique solution to this problem is $u(x,y) := 1 - \exp(\sqrt{\kappa}\,(x-1))$*, i.e. independent of* y*. Thus this is again a problem with a lower dimensional solution feature.*

Figure 23 shows the development of errors during refinement for a range of parameters κ. Note that for large values of κ this may be considered as a singularly perturbed problem, where boundary layers of thickness $\mathcal{O}(\kappa^{-1/2})$ appear. Thus for $\kappa = 10^{10}$ the solution changes from zero to (almost) one in a boundary layer of thickness 10^{-5}. We include Bakhvalov meshes in this comparison to allow better judgement of the performance of the anisotropic refinement algorithms. These meshes are a priori adapted anisotropic meshes, designed for the typical boundary layers of this problem [30, Section 2.4.1].

For $\kappa = 1$ and $\kappa = 10^2$ the results are similar to examples 2 and 3 (a) (Poisson equation), where Algorithm 2 achieves an improved convergence order due to the one-dimensional structure of the solution.

For larger values of κ the advantage of Algorithm 2 becomes even more pronounced compared to (isotropic) Bänsch-refinement. RGB-refinment Algorithm 1 fails early on, because pairs of elements are chosen for blue refinement, where the resulting mesh becomes self-overlapping. Of course this could be avoided by adding exceptions, but the performance of the refinement strategy was quite poor even for the simpler examples, and is not expected to become better by adding these exceptions.

Figure 24 shows the FE solution on the last mesh in the refinement processes before 300 nodes are reached, for $\kappa = 10^4$. Isotropic adaptive refinement (here Bänsch-refinement) is barely able to put two elements across the layer (due to the resulting fine resolution along the layer), while Algorithm 2 achieves a very good resolution of the layer with far fewer nodes already. For larger values of κ it quickly becomes impossible (or prohibitively expensive) to even put a single layer of elements into the boundary layer by isotropic local refinement. The maximum and mean aspect rations are given in the titles of the sub-figures (maxAR and meanAR).

For very large κ we again see some significant degeneration of the efficiency index of the error indicator (8) for Algorithm 2 with resulting almost stagnation of convergence. Our impression is that this results from wrongly stretched elements along the Neumann boundary, which develop early on in the refinement procedure. This results in a similar behaviour as described for Example 3 (b) and (c). None

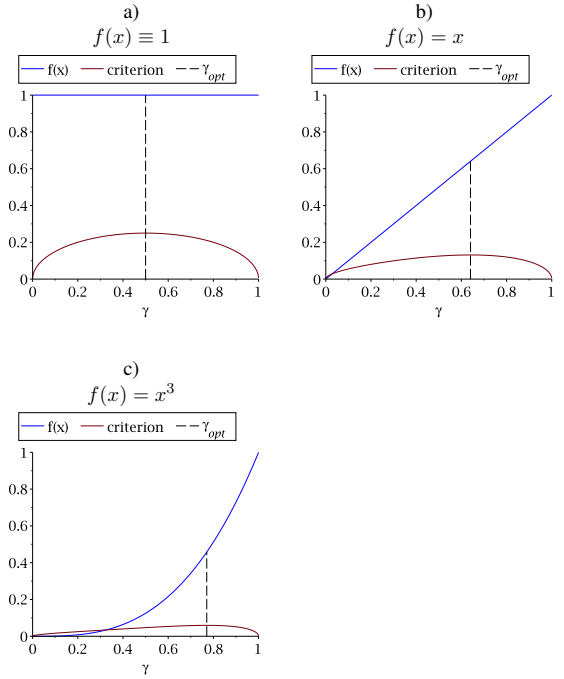


Figure 2: Optimal splitting of an edge, depending on the RHS function f.

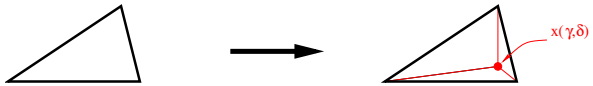


Figure 3: Graded refinement of a triangle, with parametric position of the new node $x(\gamma, \delta)$.

In each of the sub figures, the RHS function f is plotted in blue, and the term $|a(u_h, v_\gamma) - b(v_\gamma)|/|||v_\gamma|||$ from (4) in red, the dotted line marks the optimal value of the parameter γ. Extension of this idea to two or three dimensions could allow automatic generation of graded meshes, as they are for example advantageous at edge or corner singularities (e.g. [6]).

However, as for the linear elements the gradient on the current grid is constant on each element, we see already in the above example, that the new function is $a(.,.)$ orthogonal to the old FE space, independent of the parameter γ. As a consequence, the optimal position of the new node, i.e. the optimal parameter γ depends only on the right-hand-side function f and not on the current solution or on boundary data.

The edge and corner singularities in 2D and 3D, however, are typically not due to singularities of f, but due to the domain geometry, while other singularities arise from boundary conditions. So at least for linear elements an extension of this approach to 2D as in Figure 3 does not appear to generate the desired improvement.

As the second approach to utilise the refinement criterion (4) which we present in the following section shows far greater potential, we will not follow this grading idea further in this paper.

5 Application 2: anisotropic refinement in 2D

Let us consider a triangular element which is to be refined. There are actually several possibilities how such an triangle can be split, see Figure 4 for a selection. Most commonly the so called red refinement is used, which splits the triangle into four triangles of similar shape. This achieves improved spatial resolution in all directions, thus isotropic refinement.

Bisection of the element, also known as green refinement, splits only one of the edges and the element accordingly. Thus spatial resolution is improved in the direction of the edge which is split.

This can be repeated such that two edges are split in one refinement step, or even three. However, splitting all three edges by successive bisection produces the same number of degrees of freedom as the red refinement, so no significant advantage can be seen in this.

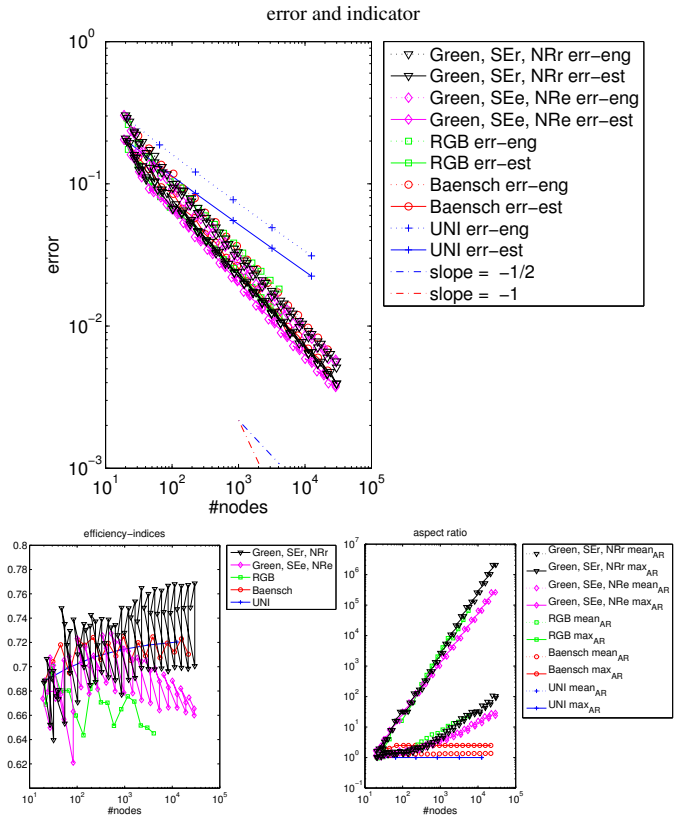


Figure 22: Comparison of different refinement algorithms for Example 5: error and indicator, efficiency, aspect ratios.

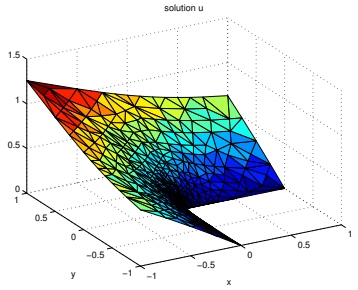


Figure 21: Solution of Example 5.

6.2 Poisson equation on an L-shape domain

Example 5. *Consider the Poisson problem*

$$\begin{aligned} -\Delta u &= 1 && \text{on } \Omega := (-1,1)^2 \setminus [0,1) \times (-1,0], \\ u &= g && \text{on } \Gamma_D := \partial\Omega, \end{aligned}$$

with the well known exact solution (in polar coordinates (r,ϕ), $r \geq 0$, $0 \leq \phi \leq \frac{3}{2}\pi$)

$$u(r,\phi) = r^{\frac{2}{3}} \sin\left(\frac{2}{3}\phi\right), \tag{10}$$

see e.g. [15, Example 1.1.4]. The Dirichlet Data g is set accordingly, see Figure 21.

Figure 21 shows the convergence of the FE solution for the considered algorithms. As well known, the uniform refinement achieves only convergence order $\mathcal{O}(N^{-1/3})$ due to the corner singularity. The error plots show only very mild differences between the adaptive refinement algorithms, all achieving the optimal convergence order of $\mathcal{O}(N^{-1/2})$. The efficiency index stays in tight interval $(0.62, 0.78)$. The edge swapping in Algorithm 2 appears to produce a slight advantage especially during the early stages of the refinement. Visual observation of the FE solution on the developing meshes indicates that edges tend to get aligned with contours of the solution.

Finally we present examples where strongly anisotropic solution features develop naturally, in boundary layers.

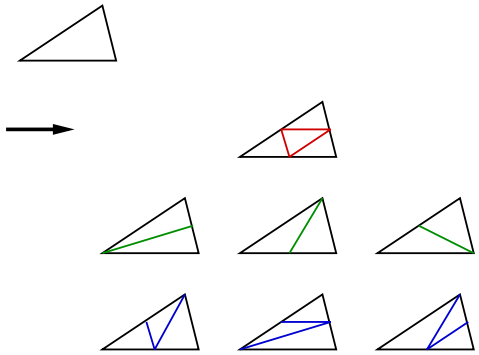


Figure 4: A selection of possible refinements for a triangular element

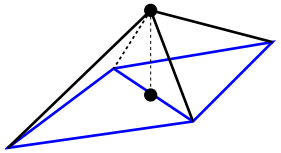


Figure 5: Linear edge-bubble

By selecting which edges to refine, we get the possibility to achieve directional refinement, i.e. anisotropic refinement.

As one expects, this is not quite as simple as it might appear on first sight. To separate different issues, we first discuss the marking of edges according to the principle (4) in the following subsection, before we discuss the possibilities of refinement in more detail.

5.1 Edge error indicator

Let us restrict our considerations to piecewise linear triangular Lagrange elements in 2D. Bisection of an edge and all elements that correspond to it is equivalent to adding a corresponding piecewise linear edge-bubble function to the ansatz space, see Figure 5.

Such an edge-bubble can be assigned to every edge in the mesh, and we denote the set of all edge-bubble functions for the current mesh by $\mathbb{S}_h^{(k)+}$. Thus the guiding principle (4) implies that the edge E whose bubble function $\tilde{v}$ produces the largest

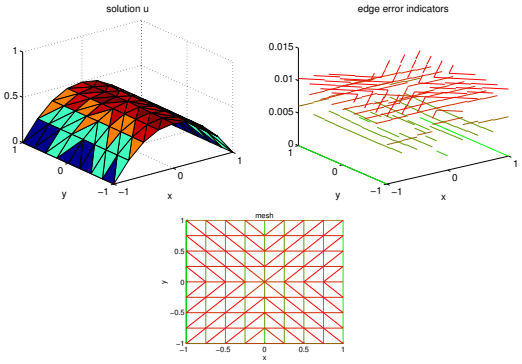


Figure 6: Soltion, edge bubble residuals and mesh for Example 2

value of

$$\eta_{\mathcal{E}} := \frac{|a(u_h, \tilde{v}) - b(\tilde{v})|}{|||\tilde{v}|||} \tag{7}$$

should be refined.

Evaluation of the expression (7) is easily achieved creating a list of the (at most two) elements that belong to each edge, and then temporarily creating the four triangles on which the linear edge bubble is linear. For boundary edges only two such temporary elements are required. A modified assembly routine then evaluates all parts of (7).

To illustrate that this very simple procedure indeed picks up the right directional information, let us consider a very simple model problem.

Example 2. *Consider the Poisson problem*

$$\begin{aligned} -\Delta u &= 1 && \text{on } \Omega := (-1,1)^2, \\ u &= 0 && \text{on } \Gamma_D := \{-1,1\} \times [-1,1], \\ \frac{\partial u}{\partial n} &= 0 && \text{on } \Gamma_N := (-1,1) \times \{-1,1\}. \end{aligned}$$

$u(x,y) = -\frac{1}{2}(x-1)(x+1)$ *is the unique solution to this problem.*

Discretisation of this problem on a uniform axis aligned triangular mesh leads to a solution as depicted in Figure 6. As the exact solution u is constant in y direction, essentially only the length of the elements in x direction has influence on the discretisation error. Thus we expect that only those edges which stretch into x direction produce large values for the expression (7). In the upper right part of Figure

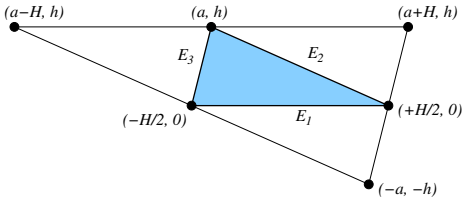


Figure 19: Patch of elements

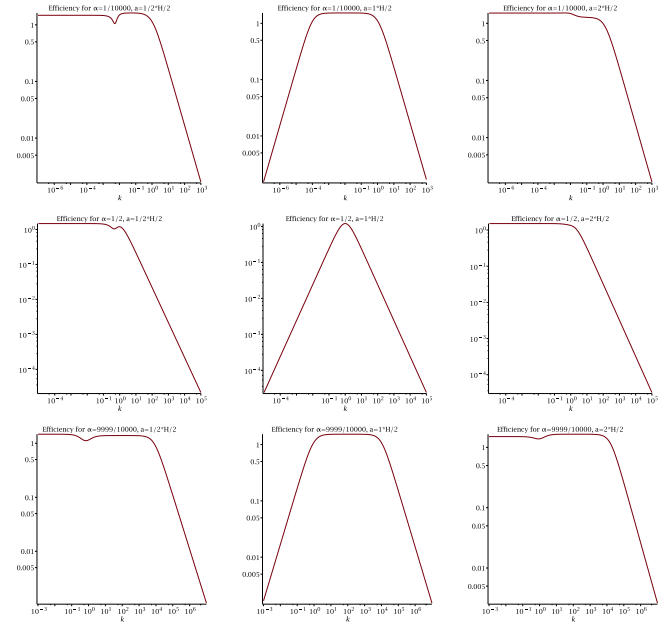


Figure 20: Example 4: Efficiency in dependence of stretching factor k for various settings of α and a.

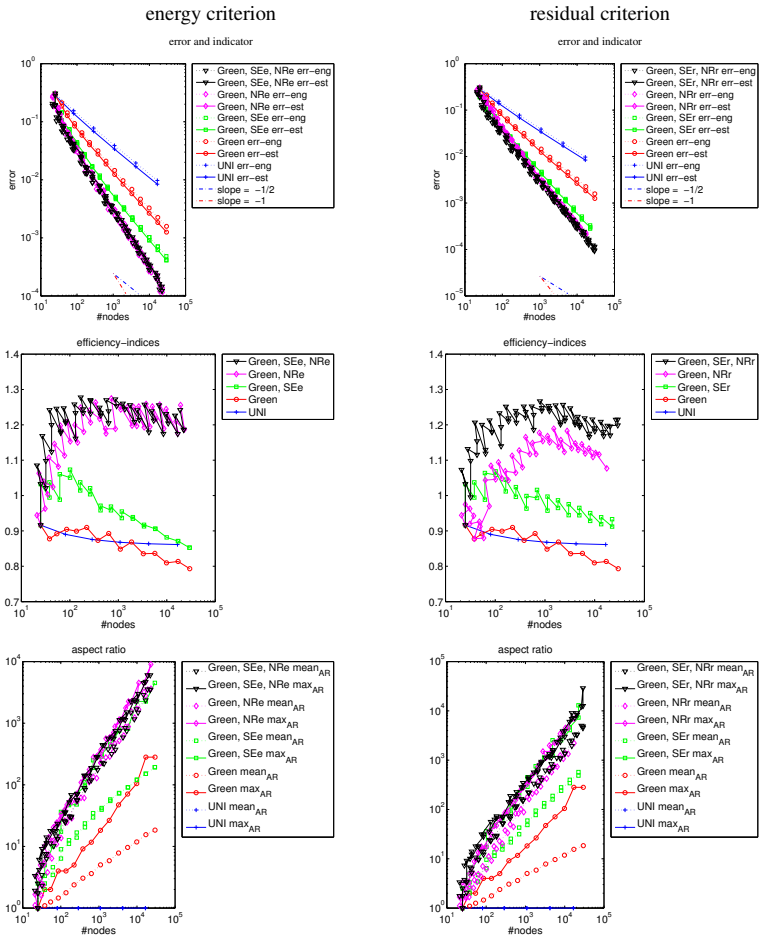


Figure 18: Comparison of different variants of Algorithm 2 for Example 3 (a): error and indicator, efficiency, aspect ratios.

6 all edges of the mesh are plotted with height assigned by edge bubble residual (7), with edges belonging to the Dirichlet boundary Γ_D treated specially. One can clearly observe that all edges parallel to the y axis have significantly smaller values of this residual, approximately 0.0041. However, the edges parallel to the x axis and the diagonal edges produce values fairly close to each other ≈ 0.0085 and ≈ 0.01 respectively. This makes sense, as both of them have the same length when projected to the x axis.

Note that while the values given above are specific to the refinement level, the ratio between them is characteristic to the problem.

Remark 1 (Relation to edge residuals in Residual-type estimators)**.** *The edge bubble residuals defined by (7) should not be confused with the edge residuals $r_{\partial T}$ which are commonly used in residual type error estimators, e.g. [2, Section 2.2]. The latter would for the problem of Example 2 be the jump of the normal derivative of the discrete solution integrated over the edge. Because the gradient is also constant in y direction, these edge residuals would be non-zero only for those edges which are parallel to the y axis, because the gradient changes only orthogonal to these edges. Thus the edge residual $r_{\partial T}$ is not suitable for the decision on refinement of* individual edges*. The edge residual $r_{\partial T}$ is however the dominant component of residual error estimators [24] and in this role well suited to decide the* isotropic *refinement of elements.*

Remark 2 (Relation to hierarchical error estimators)**.** *The edge error indicator (7) is related to a particular variant of the hierarchical error estimator of Bank and Smith [3], even though their discussion focused on p-enrichment (of the whole mesh). At the end of their Section 5.1 they briefly discuss a simplified estimator where the solution of the error approximation problem on the enrichment space is approximated by solving only with the diagonal of the stiffness matrix. Thus the resulting problem decouples, and the solution coefficients are given by*

$$v_{\mathcal{E}} := \frac{|a(u_h, \tilde{v}) - b(\tilde{v})|}{|||\tilde{v}|||^2},$$

where $\tilde{v}$ is the quadratic edge-bubble function for edge $\mathcal{E}$. In this sense $\widetilde{\eta}_{\mathcal{E}} = v_{\mathcal{E}} \, |||\tilde{v}|||$ can be seen as the contribution of edge $\mathcal{E}$ to the global error. Canceling $|||\tilde{v}|||$ we see that $\widetilde{\eta}_{\mathcal{E}} = \eta_{\mathcal{E}}$ from (7). We may consider the piecewise linear edge-bubble functions as approximations to the quadratic edge-bubble functions, thus revealing a relation.

However, [3] mention that the constants of the equivalence of the hierarchical estimator and its simplified version depend on the shape regularity of the triangles.

In light of the relation to the hierarchical error estimator we define the global error indicator

$$\eta := \left(\sum_{\mathcal{E} \in \partial \mathcal{T}} \eta_{\mathcal{E}}^2 \right)^{1/2} \tag{8}$$

by a sum of all edge indicators (7).

Remark 3 (Previous appearances of (7))**.** *After we arrived at the edge error indicators (7) we found several instances of similar expressions in error estimation (besides the less obvious relation to [3] given in Remark 2).*

The closest one is [1], where (7) arises as special case of their error reduction formula (6). This work also gave a link to earlier works by Zienkiewicz et. al. [12, 22] who considered a p-hierarchical error estimate in the context of isotropic refinement h-refinement.

In the context of hp refinement similar expressions appear in the decision process between h and p refinement, e.g. [14, Sec. 3.4] and [10, Sec. 3.2.1].

In [27] the error reduction idea was also considered in the isotropic context, enforced by newest vertex bisection refinement.

We conclude from this representative example that the edge bubble residuals provide directional information on the error. The results which we present in Section 6, will further support this.

5.2 Edge oriented anisotropic refinement

Unfortunately the edge oriented bisection of triangles alone is not sufficient in order to achieve true anisotropic adaptation [32, Section 4.3]. To illustrate this we follow our arguments from [32], and consider them in more detail in the context of (4) and Example 2.

In Figure 7 different directional refinement approaches are given, together with the number of nodes they produce for an initial uniform mesh of four nodes after one, two and three refinements according to each strategy. As for Example 2, refinement is only desired in horizontal direction. Thus variant a) would be ideal. However, this variant, also known as blue refinement due to Kornhuber and Roitzsch [23], can not be achieved by just splitting edges and elements. The two original elements must be treated together and the original diagonal edge must be removed. So this does not fit directly into the framework of the proposed refinement principle (4).

Variant b) does in each step refine the edges indicated most suitable by the refinement criterion, i.e. all edges whose projection onto the x axis has positive length. This is closer to the framework given by (4) and Subsection 5.1, but the number of nodes created due to the refinement is far higher than necessary, as new nodes are also introduced along vertical lines. In fact, asymptotically this will be almost as bad as isotropic refinement (quadrupling the number of nodes instead of doubling as in variant a)).

Variant c) attempts to get closer to a) by ignoring the fact that the diagonal actually produces the larges edge bubble residual. While the number of nodes is the same as for variant a), the error will be far worse, because the edges have the wrong alignment.

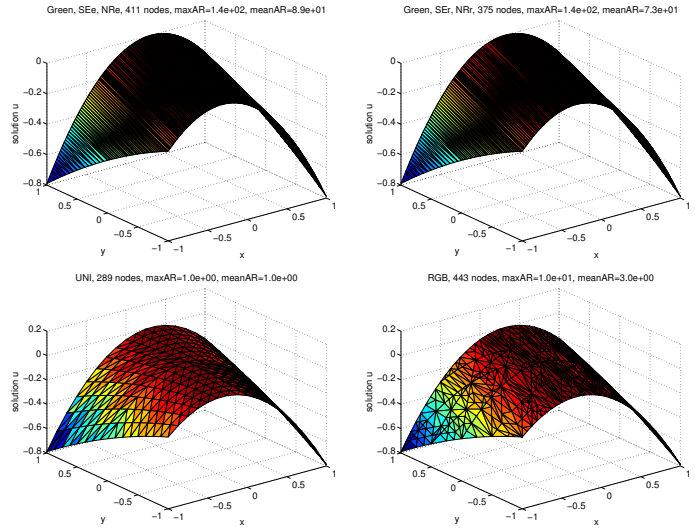


Figure 17: FE solution on meshes resulting from different refinement algorithms for Example 3 (a)

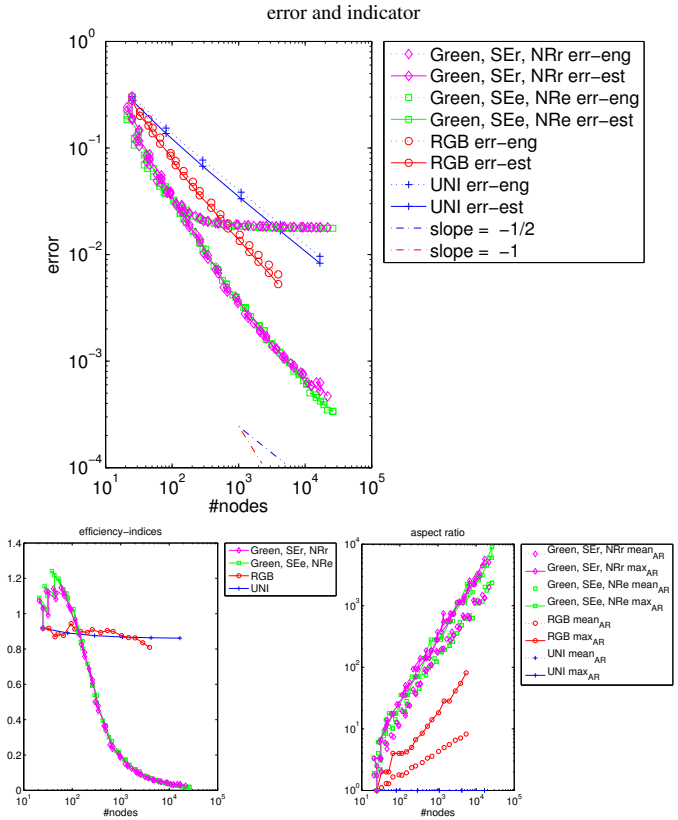


Figure 16: Comparison of different refinement algorithms for Example 3 (c): error and indicator, efficiency, aspect ratios.

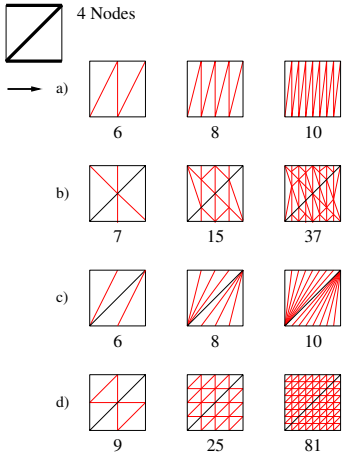


Figure 7: Directional refinement possibilities for triangles, a) blue refinement, b) splitting all edges with positive length in x-direction, c) splitting only edges in horizontal direction, d) red refinement (non-directional).

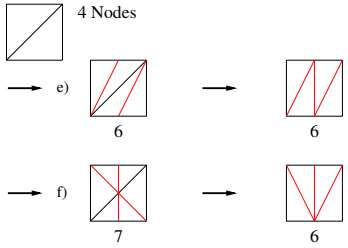


Figure 8: Directional refinement, e) combining variant b) with edge swapping, f) combining b) with node removal by edge collapsing.

For comparison red refinement (uniform) is included as variant d), producing the largest number of nodes.

Several authors combine the splitting of elements with other *mesh modification* operations in order to achieve anisotropic refinement, e.g. [13, 17, 19, 25, 26]: mesh reconnection (edge swapping), mesh coarsening (node removal) and node movement.

For the reasons indicated in Section 4 node movement does not appear practical in the context of our guiding principle (4), thus we will not discuss it here.

On first sight, reconnection and coarsening do not fit in directly with (4) either, but as we will demonstrate in sections 5.2.2 and 5.2.3 they can be used in the context of (4). But first we will demonstrate how these operations allow to attain anisotropic refinement similar to variant a).

Consider a refinement step according to variant c), followed by an edge swap of the badly aligned diagonal edge, as illustrated in Figure 8 e). Repeated application of this procedure would generate the same sequence of meshes as variant a).

In a similar way we may combine a refinement step according to variant c) with the removal of the undesired node in the centre of the mesh patch by collapsing one of the new vertical edges as illustrated in Figure 8 f). The resulting mesh is not identical to the one created by variant a), but it fits the idea of directional refinement, and repeated application of this procedure will generate a sequence of meshes with the same number of nodes as in variant a).

In theory any given mesh can be generated from any given initial triangulation of a domain (subject to constraints on topology and boundary of the meshes) by combining green refinement with reconnection and coarsening in a (possibly infinite) sequence of steps.

In addition we will consider a procedure that attempts to perform a blue refinement as in variant a) by refining a patch of two neighbouring elements simul-

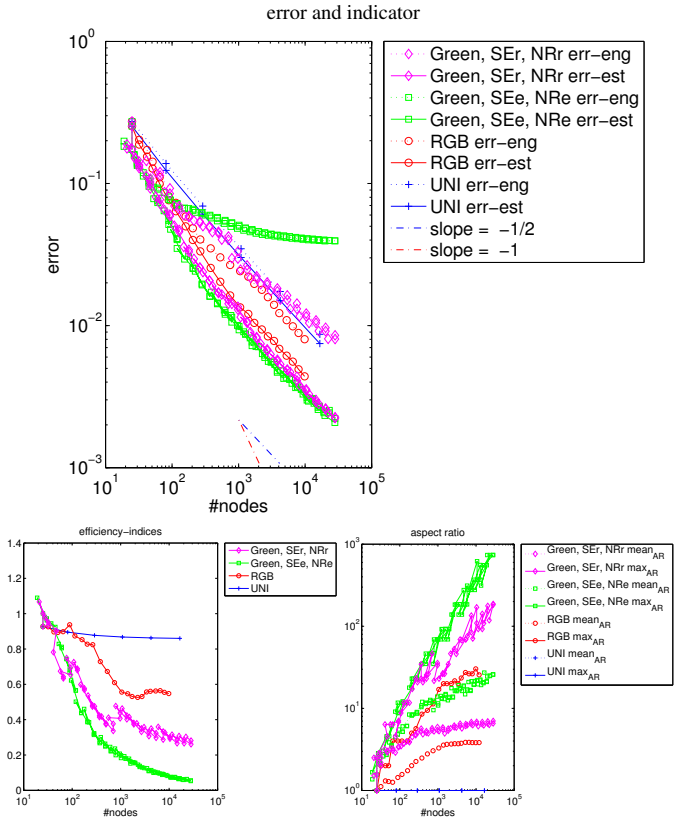


Figure 15: Comparison of different refinement algorithms for Example 3 (b): error and indicator, efficiency, aspect ratios.

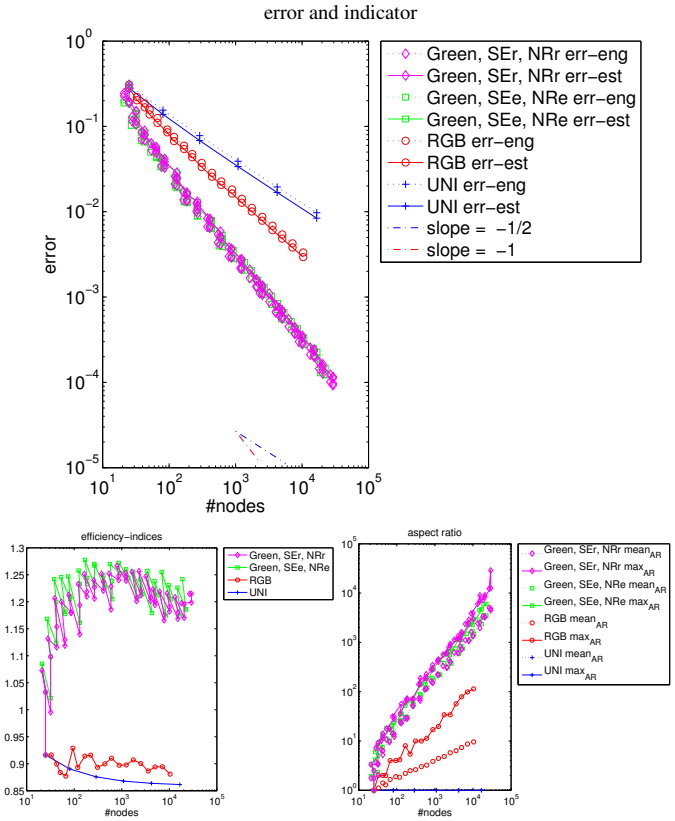


Figure 14: Comparison of different refinement algorithms for Example 3 (a): error and indicator, efficiency, aspect ratios.

taneously. We call this the red-green-blue-refinement (RGB-refinement) strategy, which we describe in the following subsection.

5.2.1 Red-green-blue-refinement

Let us modify the strategy (4) by moving to more practical considerations. Refining only one edge or element per step of the refinement algorithm is clearly computationally too expensive, as this would imply an excessive number of refinement steps, which each require the solution of a relatively large linear system of equations (up to an appropriate accuracy). Instead one should see (7) in the role of an error indicator, and all edges respectively elements with large contributions to the global error should be refined.

The favourable refinement variant a) from Figure 7 (blue refinement) can be applied in a three step approach, summarised in Algorithm 1. The first step considers on each element if it is most appropriate to refine one, two or all three edges. This decision is also taken in the spirit of (4), marking within the element all edges with more then c_1 times the maximum absolute value of the elements largest edge bubble residual, where $c_1 \in (0,1)$ is a constant. We use the value $c_1 = 1/2$ in our experiments in an attempt to facilitate approximate equidistribution of the error, noting that the local error is supposed to be reduced to $\approx 1/2$ if the edge is refined. Once the refinement pattern for the element is decided, an estimation for the error reduction weighted with the number of additional degrees of freedom in the discretisation due to the elements proposed refinement is computed and stored for each element.

In the second step the elements which reduce the error most efficiently by their proposed refinement are marked. The resulting element refinement pattern is then closed by green refinement in order to avoid hanging nodes, i.e. those elements which are not marked for refinement, but which have edges marked for refinement due to neighbouring elements, are marked to refine exactly those edges. Thus a pattern of red, green and blue refinement, representing elements with three, one, respectively two marked edges is created.

The third step then performs this refinement in the usual way for the red and green elements. The blue elements are treated specially. We attempt to group the blue elements into pairs whose marked edges obey the Z-pattern of the original element in Figure 7 (the two horizontal and the diagonal edge). If a pair obeys the pattern and the geometry of the pair permits refinement according to variant a), i.e. the resulting elements do not degenerate, then this blue refinement is executed. This is repeated as long as possible. The remaining blue elements are then refined according to variant b).

In theory this RGB refinement strategy allows the generation of meshes with pure directional refinement, where the number of degrees of freedom is inverse proportional to the discretisation step size in only this one direction. This is a big achievement, and may reduce the number of degrees of freedom required to achieve

Algorithm 1 RGB-refinemnent

Require: a mesh, edge-error-indicators $\eta_{E,i} \geq 0$, $i = 1, \ldots, n_E$, $c_1 \in (0,1)$
Ensure: refined mesh

{1. stage}
for all elements T **do**
  $\eta_{\max,T}$:= largest edge-error-indicator $\eta_{E,j}$ in this element T
  σ_T := set of edges in this element T with $\eta_{E,j} \geq c_1 \eta_{\max,T}$
  $\eta_T := \frac{1}{\#\sigma_T} \sum_{j \in \sigma_T} \eta_{E,j}^2$
end for

{2. stage}
sort the elements in decreasing order of η_T
$T = 1$, $\eta_{sum} = 0$
while $\eta_{sum} <$ sufficient **do**
  mark element T for refinement according to σ_T
  $\eta_{sum} += \eta_T$
  $T++$
end while
mark all elements which would get hanging nodes for green refinement

{3. stage}
red-refine all elements T with $\#\sigma_T == 3$
green-refine all elements T with $\#\sigma_T == 1$
while there are marked elements T with $\#\sigma_T == 2$ **do**
  starting from the first marked element T with $\#\sigma_T == 2$, find as long a string as possible of such marked elements, that obey the Z-Pattern of variant a)
  if string has only one element **then**
    refine that element according to varaint b)
  else
    for each pair of elements in the string **do**
      if refinement of the pair according to varaint a) is possible **then**
        refine the pair according to varaint a), blue refinement
      else
        refine the pair according to varaint b)
      end if
    end for
  end if
  un-mark the elements of the string
end while

At this point we have to remember that (7) provides a lower bound to the error reduction, see Lemma 1. We observe that in these cases the lower bound converges to zero, even though the error does not. The actual error reduction due to refinement of long edges in the direction of "weak curvature" is significantly larger than predicted by Lemma 1.

We investigate this further with the following example.

Example 4. *Consider a patch of P_1 elements as given in Figure 19. Let*

$$u(x,y) := -\frac{1}{2}\left(\alpha\, x^2 + (1-\alpha)\, y^2\right),$$

such that $-\Delta u = 1$ and let u_h be the nodal interpolant of u. We evaluate (7) for the three edges E_1, E_2, E_3 and compare the result to

$$|||u - u_h|||_T := \left(\int_T |\nabla u - \nabla u_h|^2 \, \mathrm{d}\Omega \right)^{1/2},$$

where T is the triangle enclosed by E_1, E_2, E_3. In this sense

$$r := \frac{\left(\eta_{E_1}^2 + \eta_{E_2}^2 + \eta_{E_3}^2\right)^{1/2}}{|||u - u_h|||_T}$$

is the (local) efficiency index of the estimate (8) on triangle T.

We define the geometry parameter a and h as multiples of H, then the resulting expression for the efficiency r is independent of H, it depends only on the factors defining a and h. Specifically let k be the stretching factor, $h := k\,H$.

We used the computer algebra system `Maple` *to evaluate and plot the efficiency r over the stretching factor k for various settings of a and α, see Figure 20. The middle column with $a = H/2$ corresponds to right angled triangles, the right column to obtuse triangles and the left column to acute triangles for $k > k_0$ and obtuse for $k < k_0$. The rows correspond to fixed anisotropy in u, with the centre row corresponding to the isotropic case.*

Interestingly, in the range between the isotropic $k = 1$ and the optimal stretching $k_{opt} = \sqrt{\frac{\alpha}{1-\alpha}}$ we always obtain good efficiency r, close to one. For $k \gg \max(1, k_{opt})$ the estimate always deteriorates, while for $k \ll \min(1, k_{opt})$ it only deteriorates for the right angled triangles.

In our experiments with Algorithm 2 we only observed this deterioration of the efficiency of indicator (8) during refinement (and resulting failure of convergence), when the solution u was significantly anisotropic but not lower dimensional. To illustrate this, we present the following example.

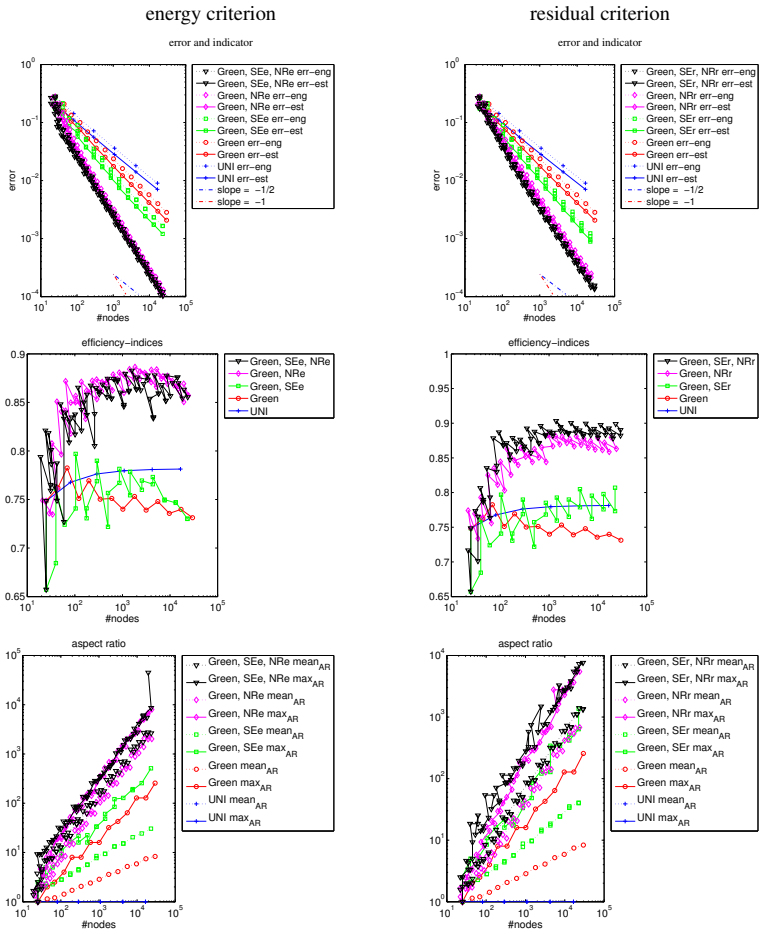


Figure 13: Comparison of different variants of Algorithm 2 for Example 2: error and indicator, efficiency, aspect ratios.

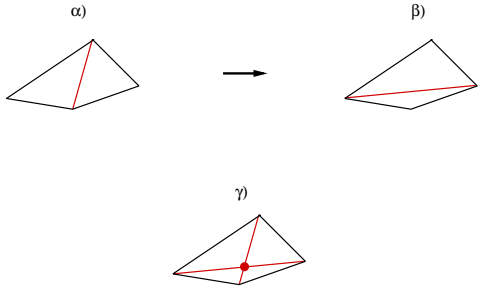


Figure 9: Edge swapping. $\alpha)$ original edge, $\beta)$ candidate swapped edge, $\gamma)$ intersection mesh

a given accuracy enormously if the solution that has to be approximated shows such an anisotropic behaviour. However, the blue refinement crucially depends on the alignment of the initial mesh with the directions of the anisotropy in the solution.

In practise however, the marking for red, green or blue refinement does not always produce the ideal strings of elements. Thus, due to the green closure refinements in each step, after a number of steps it becomes more and more difficult to find strings of elements suitable for variant a), and the other less desirable variants start to dominate. A more complicated variant of Algorithm 1 may be designed to be more robust in this respect.

5.2.2 Mesh reconnection by edge swapping

Our initial motivation for this edge swapping stems from [26], where the edge swapping was used to improve the alignment of the mesh with the solution in a general fashion. Including such an edge swapping step into the adaptive refinement procedure can potentially resolve two issues, the anisotropic refinement according to variant e) and automatic realignment of the mesh with solution features.

In [26] the PDE problem (1) was formulated as energy minimisation problem

$$u := \arg\min_{v \in \mathbb{V}} \frac{1}{2} a(v,v) - b(v).$$

Now, given a mesh and corresponding solution u_h for Lagrangian elements, one can keep the nodal values fixed, but change the connectivity of the mesh, i.e. swap an edge (Figure 9 $\alpha) \rightarrow \beta)$). One considers the function $\hat{u}_h$ which is defined by the old nodal values on the modified mesh. If the energy

$$\frac{1}{2} a(\hat{u}_h, \hat{u}_h) - b(\hat{u}_h) \tag{9}$$

Figure 10: Possible degeneration due to edge swapping

for the modified mesh is less than that for the old mesh $\frac{1}{2}a(u_h, u_h) - b(u_h)$, then the modified mesh obviously allows better approximation of the minimiser $u \in \mathbb{V}$. Likely an even better solution would be attained after recomputing the nodal values corresponding to the new mesh.

Of course one has to safeguard against degeneration of the mesh, as the edge swapping could easily produce self overlapping elements, see Figure 10 for an illustration. We allow the consideration of the swap if the surface area A_0 of the union of both triangles is not changed after swapping (up to accuracy of computing A_0) and none of the new triangles has an area of less than $10^{-2}\, A_0$.

So if the swap would reduce (9) and not degenerate the mesh, then it will be executed, otherwise not. This procedure is used in an iterative process with other mesh modifications as well.

We denote one sweep of this procedure over all interior edges of the mesh as `swap-by-energy`.

A variation of this edge-swapping strategy may be defined in terms of our criterion (4). To this end, consider one interior edge with its two neighbouring elements. We define approximate solutions u_h and $\hat{u}_h$ as above, by swapping the edge but keeping nodal values fixed. For both u_h and $\hat{u}_h$ the quantity (4) is evaluated with $\tilde{v}$ as the linear edge-bubble function defined by the intersection of both candidate edges, see Figure 9 γ). Thus, $\tilde{v}$ is the same for both candidate connectivities. If no mesh degeneration occurs, the connectivity with the lower value of (4) is chosen, since this produces a lower estimated edge error indicator. Note that Galerkin orthogonality is not fulfilled for $\hat{u}_h$ (the modified mesh).

We denote one sweep of this procedure over all interior edges of the mesh as `swap-by-residual`.

5.2.3 Mesh coarsening by edge collapsing

The second mesh modification operation, that in combination with element bisection can allow true anisotropic refinement, is node removal by edge collapsing, variant f). The principle of this operation is illustrated in Figure 11. For each node i in the mesh, one may consider, to move that node along one of the adjacent edges to the neighboring node, thus collapsing the edge and the elements that contain this edge. The node i, the collapsed edge and the collapsed elements are then removed and adjacent elements updated accordingly.

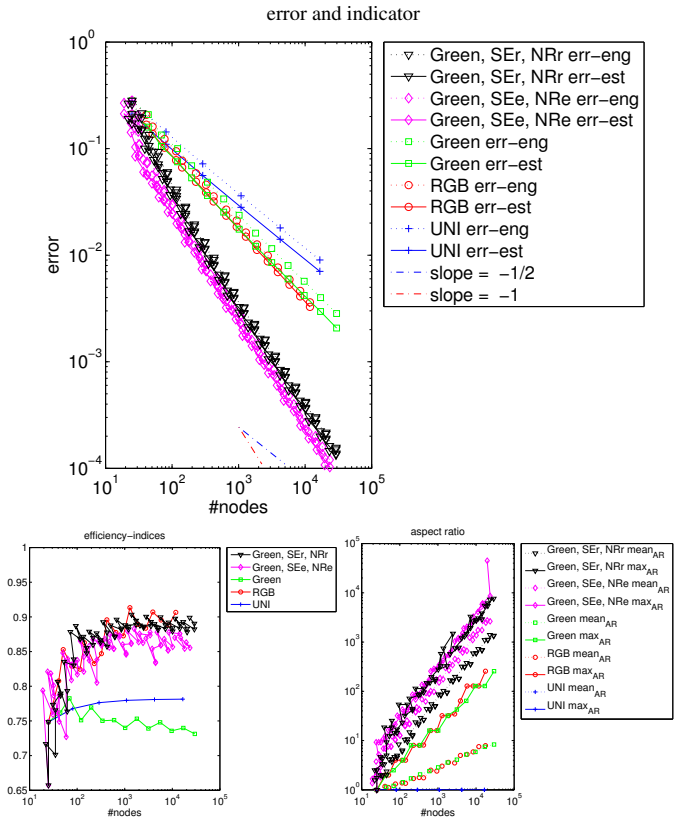


Figure 12: Comparison of different refinement algorithms for Example 2: error and indicator, efficiency, aspect ratios.

the effects of edge swapping (SE) versus edge collapsing (NR), the NR has a far stronger impact while SE gives only a slight improvement. This may be explained by the observation that the refinement pattern e) in Figure 8 requires that the diagonal edge is not refined, for which the edge error indicators in Figure 6 give no justification. However, we observed that edge swapping is a very powerful tool for realignment of the mesh with solution features, which we investigate with the next example.

Example 3. *Consider the Poisson problem*

$$\begin{aligned} -\Delta u &= 1 && \text{on } \Omega := (-1,1)^2, \\ u &= g && \text{on } \Gamma_D := \partial\Omega, \end{aligned}$$

with $g(x,y) := a\,(x+c\,y)^2 + b\,(y-c\,x)^2$. *If* $a+b = -\frac{1}{2\,(1+c^2)}$, *then* $u = g$ *fulfills the PDE, thus in this case it is the unique solution to this problem.*

If $a, b > 0$, *then this solution describes an elliptic paraboloid* $z = u(x,y)$, *with parameter* c *for the stretching directions (* $(c,-1)^T$ *and* $(1,c)^T$ *) and parameters* a, b *for the stretching of the of the principal axes. Thus this example allows to study if realignment of the mesh takes place, and if the appropriate stretching ratios are approached.*

We set $c = \pi$, $b = -a - \frac{1}{2\,(1+c^2)}$ *and consider three settings for* a*:*

(a) $a = 0$ *(parabolic cylinder),*

(b) $a = \frac{1}{10}\,\frac{1}{2\,(1+c^2)}$ *(elliptic paraboloid),*

(c) $a = \frac{1}{100}\,\frac{1}{2\,(1+c^2)}$ *(elliptic paraboloid).*

Figure 14 shows the development of the error in Example 3 (a), which is very similar to that of Example 2 in Figure 12. In Figure 17 the FE solution is plotted for the refinement algorithms of Figure 14, at the last adaptive step before the number of nodes reached 500. The bottom left picture with the mesh from uniform refinement shows that the initial mesh was not well aligned with the stretching directions $(\pi,-1)$ and $(1,\pi)^T$. The bottom right picture shows some mild stretching produced by the RGB-refinement Algorithm 1. The two top pictures illustrate that Algorithm 2 is far better able to re-align the mesh with the stretching directions, and then perform anisotropic refinement.

These results are indeed very promising. Unfortunately the tests with Example 3 (b) and (c) reveal a shortcoming, see figures 15 and 16. Here the solution is not (degenerate) one-dimensional, but shows mild anisotropic behaviour. The optimal aspect ratio r_{opt} is now finite, $r_{\text{opt}} = \sqrt{b/a}$, i.e. $r_{\text{opt}} = \sqrt{10} \approx 3.1623$ in Example 3 (b) and $r_{\text{opt}} = 10$ in Example 3 (b)). Both variants of Algorithm 2 reduce the error indicators at a similar pace as in Example 3 (a), but the actual error stagnates, the efficiency index appears to converge to zero. Investigating this phenomenon closer, we observed that the refinement procedure stretches the mesh way too far, since the error indicators still dominate in the direction of "strong curvature", even when the optimal aspect ratio is exceeded.

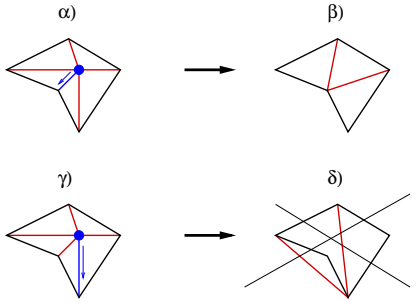


Figure 11: Node removal by edge collapsing. α) original mesh patch with candidate edge-collaps, β) resulting mesh patch; γ) original mesh patch with different candidate edge-collaps, δ) resulting degenerated mesh patch

Again care has to be taken, as this operation may produce degenerate meshes, see Figure 11 $\gamma) \to \delta)$. For this purpose, we consider this operation only if it does not change the area of the element patch (up to computation accuracy), and none of the modified elements has an area of less than 10^{-2} of the area of the patch.

The selection criterion which nodes and edges are to be removed can be designed in the same spirit as the edge-swapping in Section 5.2.2, by consideration of either the energy (9) of the solution, or by the residual (4).

In any case, in order to allow node removal at all, one has to accept a slight deterioration of solution quality. As long as this deterioration is significantly less than the improvement due to the refinement step of the overall adaptive algorithm, the error is reduced overall.

In the energy criterion, for each interior node i of the mesh, the patch of elements containing this node is found and this patches local contribution to the solution energy (9) is computed, denoted by E_{loc}. Then each edge e connected to node i is considered for removal, leaving the nodal values of the solution fixed, the local energy contribution of the modified patch of elements is computed and denoted by $E_{loc,e}$. The value of $E_{inc,i} := E_{loc,e} - E_{loc}$ and the edge number for the edge which produces the least increase in energy are then stored for node i.

When this is done for all interior nodes, these nodes are sorted in increasing order of $E_{inc,i}$. Then starting with the node of least increase in energy, the collapse of the corresponding edges is executed, as long as the sum of energy increase is less than an allowed increase E_{tol}. In this only edges are removed, whose nodes have not been affected by previous node removals in this sweep.

We denote this procedure as `node-remove-by-energy`.

The procedure for the residual-criterion is similar. Instead of choosing the edge

of minimum increase in energy for each node i, we choose the edge whose collapse produces the least value of criterion (4), where the u_h is taken to be the interpolant of the fixed nodal values on the resulting mesh after edge collapse and $\tilde{v}$ as the nodal basis function for node i on the original mesh. To evaluate (4) in this setting, one has to temporarily create an intersection mesh for the patch meshes before and after collapse of the candidate edge, to facilitate the use $\tilde{v}$ and u_h in one assembly routine for $a(u_h, \tilde{v})$ on the patch.

We denote this procedure as `node-remove-by-residual`.

5.2.4 Anisotropic adaptive algorithm

As an alternative to the RGB refinement algorithm, we combine edge oriented green refinement with edge-swapping (variant e)) and edge-collapsing (variant f)) into one general anisotropic adaptive algorithm, which is summarised as Algorithm 2.

Note that it may be possible to replace some of the *assemble and solve for* u_h steps by less expensive operations, or even to leave them out entirely. For example after a refinement it may be sufficient to do just one Gauß-Seidel or Jacobi sweep in order to update the solution u_h to have sufficient information for the edge-swapping or node removal. Indeed one may even use the information from the evaluation of the edge-error-indicators (4) to perform a pseudo-Jacobi-update without even assembling the new stiffness matrix.

However, our main goal in this paper is a first evaluation of the merits of the approach, testing what is possible. Thus, to avoid side effects, the solution u_h is recomputed after each modification sweep of the mesh.

Algorithm 2 Anisotropic adaptive algorithm

Require: initial mesh

1: **while** not satisfied **do**
2: assemble and solve for u_h
3: edge swapping
4: assemble and solve for u_h
5: node removal
6: assemble and solve for u_h
7: evaluate edge-error-indicators
8: edge-oriented refinement
9: **end while**

Further it is not necessary to have all mesh modification operations in the algorithm. In the numerical examples we will also consider variants of this algorithm, where one ore more steps are left out.

6 Numerical experiments

In this section we investigate the performance and robustness of the proposed refinement indicator (7) and the refinement algorithms by testing them on suitably chosen example problems. The results are presented in figures which show the error indicator η ((8), denoted err-est), the energy norm of the error $|||u - u_h|||$ (err-eng), the efficiency index $\eta/|||u - u_h|||$ and the maximum and mean aspect ratios of elements in the mesh ($\max_{\mathrm{AR}}$ and $\mathrm{mean}_{\mathrm{AR}}$) for the different refinement algorithms.

We compare the refinement algorithms

UNI: uniform (red) mesh refinement,

Baensch: isotropic local refinement by bisection (algorithm of Bänsch, [8]),

RGB: RGB-refinement Algorithm 1

and several variants of Algorithm 2 which are distinguished by specifying the components of Algorithm 2 that are used,

Green: green refinement according to edge indicator (7),

SE: edge-swapping (one sweep over all interior edges, as described in Subseciton 5.2.2),

NR: edge-collapsing, resulting in node removal (one sweep as described in Subsection 5.2.3).

For the SE and NR components two variants are considered, either based on the energy considerations (**SEe** and **NRe**), or based on the residuals (**SEr** and **NRr**).

6.1 Poisson equation on the unit-square

First we demonstrate effectiveness and shortcomings of the proposed refinement strategies. To this end, figures 12 and 13 show for Example 2 the development of the error, the indicator, the efficiency index and aspect ratio during refinement with various refinement algorithms. First we observe that the efficiency index stays in the fairly tight interval $(0.65, 1)$ for all of the refinement algorithms. Thus the error indicator can even be interpreted as error estimate in these cases.

In Figure 12 it is easily observed that both variants ("Green SEe NRe" and "Green SEr NRr") of the refinement Algorithm 2 are able to exploit the lower dimensional structure of the solution of this problem, gaining an improved convergence rate of (close to) $\mathcal{O}(N^{-1})$ instead of the $\mathcal{O}(N^{-1/2}) = \mathcal{O}(h)$ in isotropic refinement. The RGB-refinement Algorithm 1 and the Green refinement without edge swapping or edge collapsing also show a slightly improved convergence order, but are significantly outperformed by Algorithm 2. The observed growth in aspect ratio matches the error reduction.

The different variants of Algorithm 2 are explored in Figure 13. The left column shows variants with the energy criterion for edge swapping (SEe) and collapsing (NRe), while the right column shows those with the residual counterpart (SEr, NRr). The difference between these criteria appears to be minor. If we compare