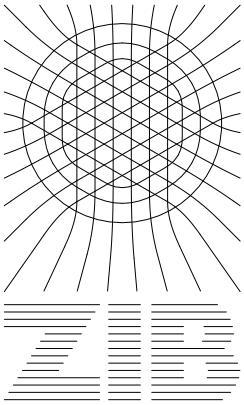
Konrad-Zuse-Zentrum für Informationstechnik Berlin



Adaptive h-p approximation procedures, graded meshes and anisotropic refinement for Numerical Quadrature

Gerhard W. Zumbusch

Konrad–Zuse–Zentrum für Informationstechnologie, Berlin (ZIB)
Heilbronner Str. 10, D–10711 Berlin–Wilmersdorf, Germany
zumbusch@zib–berlin.de
http://www.zib-berlin.de/~bzfzumbu

SC 95-24 (October 1995)

Adaptive h-p approximation procedures, graded meshes and anisotropic refinement for Numerical Quadrature

Gerhard W. Zumbusch[1]

[1]Konrad–Zuse–Zentrum für Informationstechnik Berlin (ZIB), Heilbronner Str. 10, D-10711 Berlin, Germany,
zumbusch@zib-berlin.de, http://www.zib-berlin.de/~bzfzumbu

Abstract—A set of adaptive algorithms for quadrature on multi-dimensional polyhedral domains is presented. Several kinds of refinement are discussed, covering local improvement of quadrature order and splitting the domain into sub-domains, resulting in isotropic, graded or anisotropic grids. The algorithms are pure local heuristics using no a priori knowledge or tuning parameters. This approach was motivated by results from finite element theory for optimal approximation results. Numerical experiments show the optimality of pure local greedy-like algorithms for singularity-type functions typically occurring in finite element computations.

Key Words: Numerical integration; h-p adaptivity; mesh grading; anisotropic meshes; a posteriori error estimation; self-adaptive control.

1 Introduction

The purpose of this paper is twofold: First we want to construct algorithms for numerical quadrature, which employ very low number of function evaluations on the final approximation level. This is of main importance for applications like [1], where each quadrature point is connected to a great amount of computation. However, the whole quadrature algorithm need not be the fastest itself.

Second we treat the problem of numerical quadrature as a model for adaptive finite element algorithms, discarding the linear algebra involved and simplifying the topic of a posteriori error estimation. We want to study the performance of different refinement techniques and the issue of heuristic refinement control. Hence we sometimes use terms from finite elements and we consider those algorithms only, which could be implemented in a boundary value problem context.

The paper is guided by some considerations on h-p finite elements compiled in [2]. Since it is known for piecewise polynomial interpolation, that only the mixture of sub-domain splitting and order elevation is able to break the complexity barrier of algebraic convergence known for a single refinement procedure for functions with singularities, we focus on this class of approximation. We will call them h-p methods.

For example, even for the finite element solution of Poisson's equation $-\Delta u = f$ with $u \in H^{1+\alpha}(\Omega \subset \mathrm{R}^d)$ and piecewise polynomials, a polyhedral domain and errors measured in energy norm or L^2 norm, errors vs. the number of degrees of freedom n, the following estimates hold:

- h–uniform, h–adaptive and asymtotic p-version $|||er||| \leq c\ n^{\beta}$

- h–p–adaptive and pre-asymptotic p-version $|||er||| \leq c\ e^{\delta\ \sqrt[d]{n}}$

The values β and γ depend on the dimension d and demand anisotropic grids for higher dimensions d. We clearly see that we have to use a combination of h- and p-refinement to get optimal approximations, since we do not want to rely on the well suitedness of an initial grid.

The first question that arises is control. We will discuss some strategies for mixing h-refinement (sub-domains) and p-refinement (order elevation). There still seems to be no satisfying h-p control strategy, maybe except for the one dimensional case, where some heuristics perform quite well.

In search of optimality, following the approximation results known, we have to consider more complicated h-refinements than just isotropic bisection of domains or more general a fixed partition scheme. Hence we will have a look at mesh grading, which is able to generate optimal grids in one dimension. Additionally we will employ anisotropic refinement for higher dimensions and we will combine this with locally varying orders (h-p).

2 Numerical Quadrature

Given a polyhedral, open, bounded domain $\Omega \subset \mathrm{R}^d$ and a scalar integrable function f defined on the domain, we want to compute numerically the integral

$$\int_\Omega f(x)\ \mathrm{dx}.$$

We approximate the solution using a quadrature rule $\{(x_i, w_i)|i = 1, \ldots, n\}$ with n quadrature points x_i or function evaluations $f(x_i)$ (computational work).

$$\approx Q(f) := \quad \sum_i f(x_i)\ w_i.$$

We define the (absolute) error of the quadrature formula by

$$er := |Q(f) - \int_\Omega f(x)\ \mathrm{dx}|$$

and we will measure the quality of a formula by the values of error er and work n. We use for instance Gauss-Legendre quadrature rules for one dimensional quadrature and tensor products of one dimensional formulae for quadrature on quadrangles and hexahedra (see e.g. [3, 4]). On Simplices we use either special Gauss formulae or warped formulae of quadrangles and hexahedra.

The first idea to improve a quadrature result for a given Gauss formula is to elevate the order of the formula analogously to the spectral method or finite element p-version. A higher order results in a larger number of quadrature points n and (hopefully) in a smaller error. For a regular function f with bounded higher derivatives we know of course the exponential decay of the error. For polynomial functions f, the formulae will even be exact for orders higher or equal the polynomial degree. To illustrate the converse, we have a look at example 1.

Example 1

$$f(x) = |x|^{-1/2}, \quad \Omega = (0,1) \subset \mathrm{R}$$

The results are depicted in figure 1. Here we do not obtain exponential convergence, but pure algebraic one.

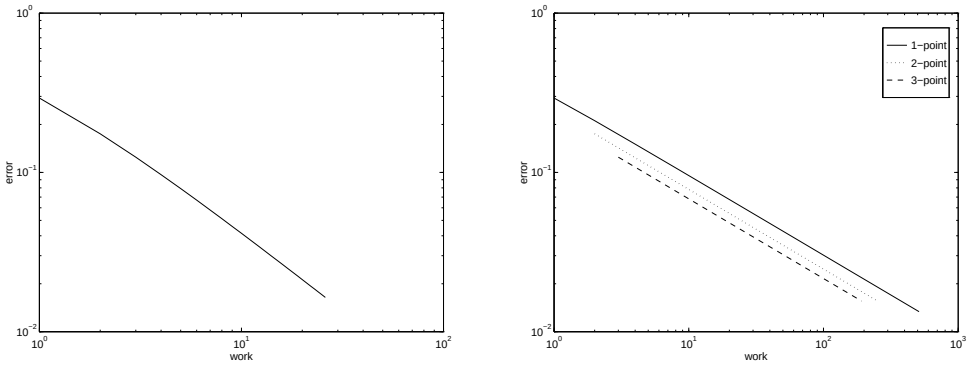


Figure 1: Left: convergence for order elevation (p). Right: convergence for uniform subdivision (h). 1D example.

3 H-Refinement

3.1 Global H-Refinement

Another method to improve the approximation is the decomposition of the domain into several smaller sub-domains. For each sub-domain we apply the usual (affine transformed) quadrature formula.

$$\overline{\Omega} = \bigcup_j \overline{\Omega_j}, \qquad \text{disjoint } \Omega_j$$

This method is also used to decompose a general polyhedral domain into standard shaped elements via a tessellation. The final result is the sum of the results of the sub-domains.

$$\int_\Omega f(x)\ \mathrm{dx} \quad = \sum_j \int_{\Omega_j} f(x)\ \mathrm{dx}$$

For each bisection of all sub-domains the number of quadrature points n can double, the order of approximation is maintained and the error decreases by a factor depending on the regularity of f.

Applying this strategy to example 1, we see in figure 1 an algebraic convergence of the solution, which needs about n-times more quadrature points and is clearly slower.

3.2 H-Adaptive Refinement

Hence we want to improve the h-refinement procedure. We introduce adaptivity. There are several sub-domains, which we can treat differently. We refine (bisect) only those, which have a large error contribution. This will reduce the number of quadrature points n. The main goal is that of error equilibration, which can be proved to have the lowest n for a given error tolerance.

Algorithmically, we have to add some sort of a posteriori error estimation. We estimate local errors on Ω_j comparing two different quality quadrature rules like

$$\epsilon(\Omega_j) \approx |Q_1(f) - Q_2(f)|.$$

Now we have to select those sub-domains to refine. A large number will result in nearly uniform refined grids which is not efficient and a small number results in a large number of "calculate–and–refine" cycles being not efficient either.

We use a threshold criterion:

Algorithm 1 *h-adaptive:*
for all Ω_j*:*
if $\epsilon(\Omega_j) \geq \kappa$ *then subdivide* Ω_j
with threshold κ

One good choice for threshold κ is

$$\kappa := \frac{1}{2} \max_j \epsilon(\Omega_j)$$

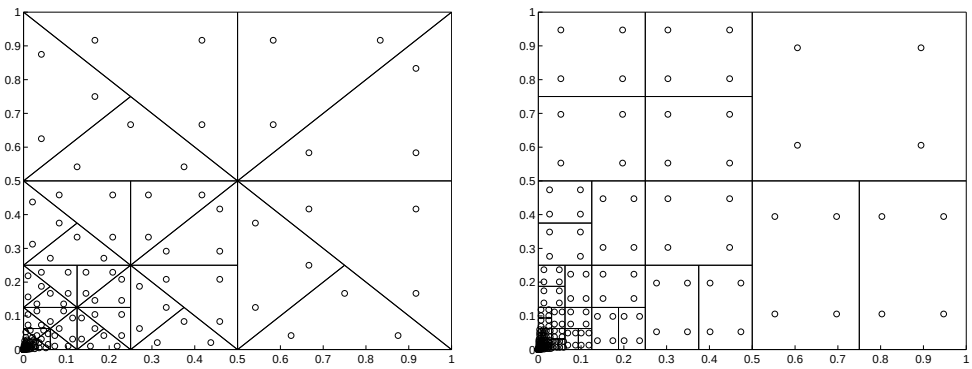


Figure 2: H-adaptive refined grids, 2D example, left: triangles, right: quadrangles. dots mark the quadrature points

Example 2

$$f(x) = \|x\|^{-3/2}, \quad \Omega = (0,1)^2 \subset \mathrm{R}^2$$

.

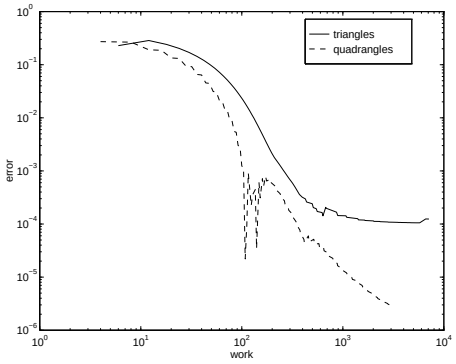


Figure 3: H-Adaptive refined grids, 2D example, convergence.

We use an h-refinement strategy, which bisects each sub-domain marked for refinement along on of the longest edges, to almost maintain aspect ratios of the domains. We consider this refinement as isotropic.

Figures 2 and 3 show the results for example 2. We obtain an exponentially decaying initial convergence phase followed by a slower (asymptotic) algebraic one, when the singularity is resolved (saturation). The grids are highly refined to the singularity until saturation takes place. The quadrangles perform slightly better than the triangles, but are geometrically less flexible.

We want to mention some alternative threshold criteria: we could have forced a certain number of domains being refined, which leads to geometrically increasing n even before saturation. We could enforce a certain part of the total error to be refined, which is essential for a proof of termination of the total algorithm. There is an additional modification proposed by [5]: Define

$$\begin{aligned} \epsilon^+(\Omega_j) &:= \frac{\epsilon(\Omega_j)^2}{\epsilon(\text{father of } \Omega_j)} \\ \kappa^+ &:= \min(\max_j \epsilon^+(\Omega_j), \tfrac{1}{2} \max_j \epsilon(\Omega_j)) \end{aligned}$$

The ϵ^+ are rather estimates for the local errors after refinement than before refinement.

Algorithm 2 *h-adaptive (with extrapolation criterion):*
for all Ω_j*:*
if $\epsilon^+(\Omega_j) \geq \kappa^+$ *then subdivide* Ω_j

For an overview over standard and commercially available adaptive quadrature algorithms we generally refer to [3, 4, 6] and [7]. A more advanced algorithm was proposed by [8, 9].

4 Advanced Refinement

4.1 H-P-Refinement

First we look at the combination of h-refinement (subdivision) and p-refinement (order elevation). This theoretically leads to (sub-) exponential convergence. The question is, how to implement such algorithms in a robust way. We will extend algorithm 1 by h-p features.

4.1.1 Global h-p

Algorithm 3 *global h-p:*

- *perform adaptive h-refinement steps by algorithm 1 until a tolerance tol_s is reached.*
- *then perform adaptive p-refinements by algorithm 1 until the final precision tol is reached.*

Uniform p-refinement could be employed in the second step, too. We call this "global" h-p, because we use a global switching to decide, whether to refine in h- or p-manner. There is a tuning parameter tol_s, which has to chosen. We use $tol_s \approx \sqrt{tol}$, where [10] proposes $tol_s \approx 1/20 \ldots 1/10 \cdot tol$.

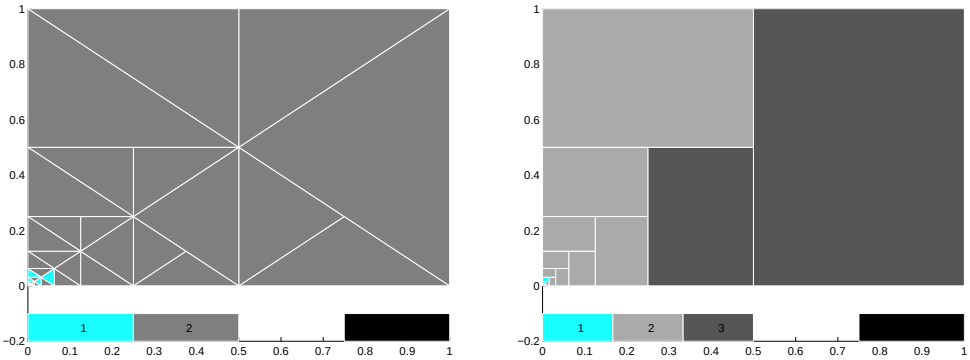


Figure 4: H-P-adaptive refined grids, global strategy, 2D example, left: triangles, right: quadrangles. color indicates different quadrature formulae.

Figure 4 and 5 show grids and convergence for example 2. The switching point is marked extra. We obtain exponential convergence, as desired. There is one point to mention: The elements next to the singularity use an ridiculously high order quadrature formula. This is due to the element size. The element sizes do not decrease fast enough down to the singularity. The error remains relatively high and further refinements have to performed as p-refinements, where h-refinements

were more appropriate. Nevertheless it is a very robust strategy with exponential convergence.

4.1.2 Local h-p

We now turn to an algorithm, called local. We make local decisions whether to refine in h- or p-manner. We want to use that kind of refinement, which leads to the smaller error. Hence we employ two different error estimators, estimating both errors:

$$\begin{aligned} \epsilon_h(\Omega_j) &= \sum \epsilon(\text{ sons of } \Omega_j) \\ \epsilon_p(\Omega_j) &= \epsilon_{\text{order}+1}(\Omega_j) \end{aligned}$$

They are implement by special quadrature formulae. We now construct the following

Algorithm 4 *local h-p:*
if $\epsilon(\Omega_j) \geq \kappa$ *then if* $\epsilon_h(\Omega_j) \leq \epsilon_p(\Omega_j)$
then refine by subdivision (h)
else refine by order elevation (p)

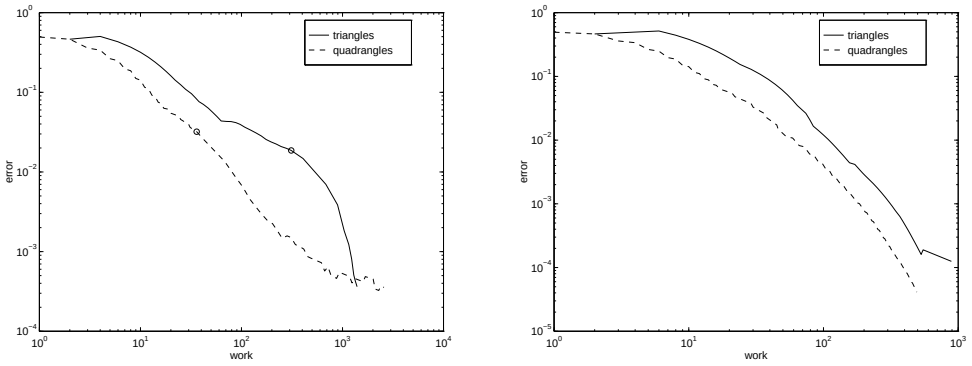


Figure 5: H-P-adaptive refined grids, convergence. left: global strategy with switching form p to h refinement, right: local strategy.

These kinds of local greedy algorithms were proposed for finite elements by [11] and are called 'best basis' for wavelets [12].

We modify algorithm 4 by taking computational work into account. We use a criterion

Algorithm 5 *local h-p (considering work)*
if $\epsilon(\Omega_j) \geq \kappa$ *then if* $\epsilon_h(\Omega_j) g(work_h) \leq \epsilon_p(\Omega_j) g(work_p)$
then refine by subdivision (h)
else refine by order elevation (p)

with a positive increasing function g. We choose $g(x) = e^x$ which stabilizes the convergence history and can be interpreted in the sense of information per work, see e.g. [13].

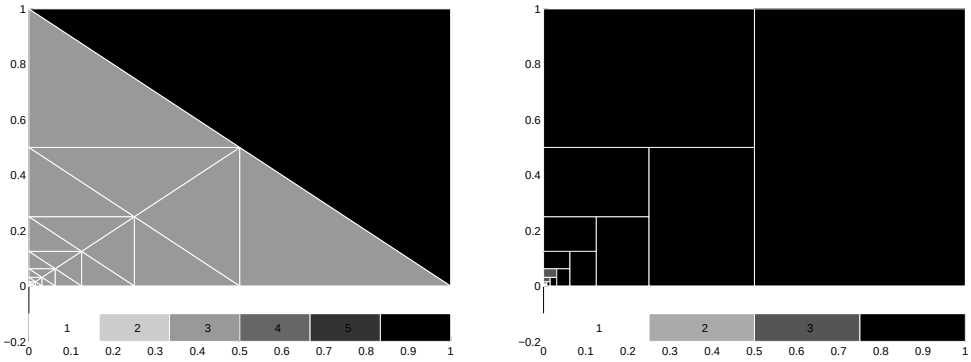


Figure 6: H-P-adaptive refined grids, local strategy, 2D example, left: triangles, right: quadrangles. color indicates different quadrature formulae.

Figures 5 and 6 show the performance of algorithm 5 for example 2. We see the exponential convergence, the grids refined towards the singularity and the distribution of higher order away from the singularity. The local h-p strategy seems to be remarkably more efficient than the global one in this case.

4.2 Graded Refinement

As we have seen in chapter 4.1.1, grid refinement by bisection of domains sometimes does not deliver final grids, which are sharply refined enough. Such grids can be considered to be inefficient due to bisection. For one dimensional h-p finite elements, asymptotically optimal grading factors of $(1 - \sqrt{1/2})/2 \approx .15$ are known. We now introduce graded refinement by

Algorithm 6 *local mesh grading*
refine an edge $(x^{(1)}, x^{(2)})$ *by subdivision (h) defining a new node as*

$$\tilde{x} \;=\; (1-c)\; x^{(1)} \;+\; c\; x^{(2)}$$

with $c \in (0,1)$ *such that* $\epsilon_c(\Omega_j)$ *is minimal*

The error $\epsilon_c(\Omega_j)$ is estimated as usual and the minimization is done by a derivative-free continuous minimization algorithm, up to a certain precision.

There is a more heuristic variant, maximizing the change of the solution:

Algorithm 7 *local mesh grading (solution criterion)*
refine an edge $(x^{(1)}, x^{(2)})$ *by subdivision (h) defining a new node as*

$$\tilde{x} \;=\; (1-c)\; x^{(1)} \;+\; c\; x^{(2)}$$

with $c \in (0,1)$ *such that* $|Q(f, \Omega_j - Q_c(f, \Omega_j)|$ *is maximal*

We apply this algorithm 7 to example 1 in combination with a local h-p-strategy. Figure 7 shows the strongly refined grid and the convergence history. The result seems to be nearly the optimal one known for the one dimensional case.

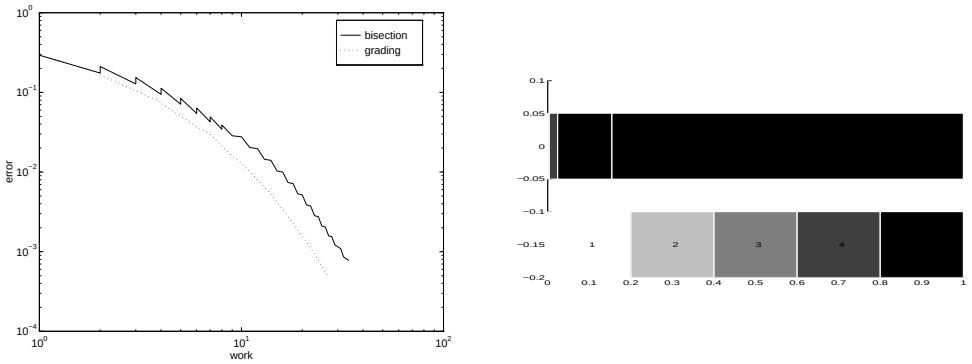


Figure 7: Local h-p refinement on graded grids, 1D example, left: convergence with and without grading, right: graded h-p grid.

4.3 Anisotropic Refinement

We now consider another refinement technique, which is supposed to make substantial savings in computational work for higher dimensions. We implement the anisotropic refinement as a sequence of directed bisection refinements of domains.

Let us suppose, the solution is smooth along an edge and has singular behavior in all directions perpendicular on the edge. We would like to refine the grid mainly in the directions of singularity, which is perpendicular on the edge. In R^3 this leads to a quasi 2 dimensional refinement and a reduction of work from h^{-3} down to h^{-2}.

We neglect any minimum angle conditions and irregularities of the grids and perform refinements just guided by refinement criteria. A maximum angle condition is maintained, supposing all angles of the initial grid are bounded from below. We implement

Algorithm 8 *anisotropic refinement*
refine a domain Ω_j by bisecting an edge k
such that $\epsilon_k(\Omega_j)$ is minimal

The error $\epsilon_k(\Omega_j)$ is estimated by formulae k.

The more heuristic variant like in chapter 4.2 is maximization of the change of the solution.

Algorithm 9 *anisotropic refinement (solution criterion)*
refine a domain Ω_j by bisecting an edge k
such that $|Q(f, \Omega_j - Q_k(f, \Omega_j)|$ is maximal

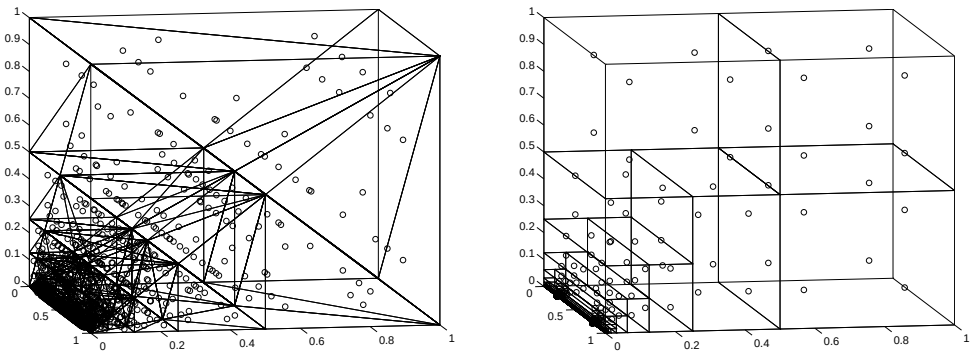


Figure 8: Anisotropic grids, 3D example, left: tetrahedra, right: hexahedra.

We test the algorithm with a model

Example 3

$$f(x) = (x_1^2 + x_2^2)^{-3/4}, \quad \Omega = (0,1)^3 \subset \mathrm{R}^3$$

.

where f is independent from x_3. We expect a quasi 2 dimensional refinement in x_1-x_2-direction. We apply algorithm 9 with h-refinement to example 3. The results are depicted in figures 8 and 9. The quadrangles perform perfectly, because they are oriented axis-parallel. It is not a difficult task to detect the directions of refinement. The tetrahedron case is more interesting here. Although the grid looks not as good, the convergence clearly benefits form anisotropy. The refinement strategy seems to perform good. Hence there is a chance to apply the algorithm to more complicated anisotropic scenarios.

We now want to combine anisotropic refinement with a local h-p strategy. We apply again algorithm 9 to example 3. Figure 9 shows the convergence history. We see that pure h-p refinement with isotropic refinement does not enter the exponential convergence phase. It is slower than anisotropic h-refinement. The combination of h-p and anisotropic refinement performs best for this example, speeding up the anisotropic h-refinement a little bit.

5 Conclusion

We have presented a set of algorithms for adaptive numerical quadrature which are structurally equivalent to some finite element strategies. We have used h-adaptive and different h-p adaptive strategies and extensions for (a posteriori) mesh grading and anisotropic refinement. They were all pure local greedy-like procedures without any a priori knowledge. We have compared these algorithms and combinations of them using some singularity-type model problems in 1, 2 and 3 dimensions with

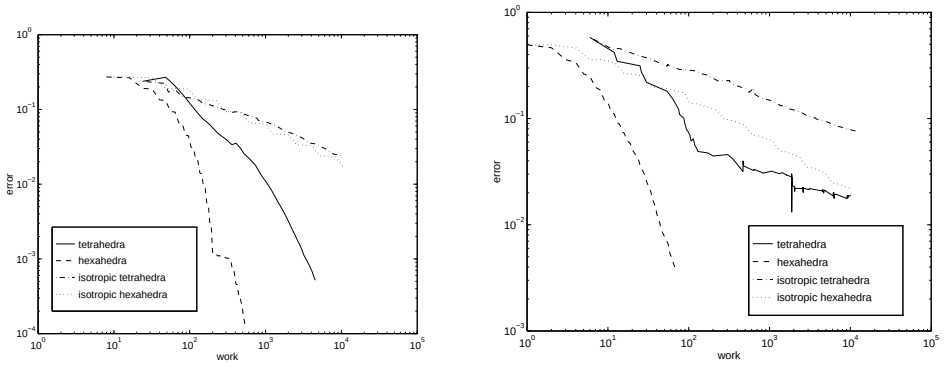


Figure 9: Anisotropic grids, 3D example, convergence, left: subdivision (h), right: local h-p strategy.

hexahedra, tetrahedra quadrangle and triangle grids. The h-p approximations performed nearly optimal for all tests.

We want to give an outlook to generalization to finite elements. The main gap missing is the step from pure local procedures to globally coupled ones. We have to enforce continuity of our approximation as described in [14] for linear elements and in [15] for higher order elements. We also could enforce continuity in a weak sense using e.g. Mortar elements [16]. We have to introduce global linear algebra and we have to extend the a posteriori error estimation like in for anisotropic grids [17, 18]. Anisotropic convergence is treated in [17, 19]. Hence it should be possible to extend the highly efficient h-p procedures presented for more general boundary value problems.

References

[1] Ch. Schütte, M. Dinand, G. Zumbusch, and R. Brinkmann. Dynamics of erbium–doped waveguide lasers: Modelling, reliable simulation, and comparison with experiments. Technical Report SC 95-19, Konrad-Zuse-Zentrum, 1995.

[2] I. Babuška and M. Suri. The p and h-p-versions of the finite element method, basic principles and properties. *SIAM Rev.*, 36(4):578–632, 1994.

[3] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration.* Academic Press, New York, 1975.

[4] P. Deuflhard and A. Hohmann. *Numerical Analyis*, chapter 9. de Gruyter, Berlin, New York, 1995.

[5] W. C. Rheinboldt. On a theory of mesh-refinement processes. *SIAM J. Numer. Anal.*, 17:766–778, 1980.

[6] C. Überhuber. *Computer–Numerik*, volume 2, chapter 12. Springer, Berlin, 1995.

[7] R. Piessens, E. de Doncker-Kapenga, C. W. Überhuber, and D. K. Kahaner. *Quadpack, A Subroutine Package for Automatic Integration*. Springer, Berlin, 1983.

[8] T. Bonk. A new algorithm for multi-dimensional adaptive numerical quadrature. In *9th GAMM-Seminar on Adaptive Methods - Algorithms, Theory and Applications*, volume 46 of *Notes on Numerical Fluid Mechanics*, Braunschweig, 1993. Vieweg.

[9] T. Bonk. *Ein rekursiver Algorithmus zur adaptiven numerischen Quadratur mehrdimensionaler Funktionen*. PhD thesis, Institut für Informatik, TU München, 1995.

[10] A. Patra and J. T. Oden. Problem decomposition for adaptive *hp* finite element methods. Technical report, TICOM, Univ. of Texas at Austin, Tx., 1993.

[11] W. Gui and I. Babuška. The *h*, *p* and *h-p* versions of the finite element method in 1 dimension. i., ii., iii. *Numer. Math.*, 49:577–612, 613–657, 659–683, 1986.

[12] R. R. Coifman and M. V. Wickerhauser. Best-adapted wave packet bases. via ftp, Yale Univ., New Haven, 1990.

[13] G. W. Zumbusch. Symmetric hierarchical polynomials and the adaptive *h-p*-version. Technical Report SC 95-18, Konrad-Zuse-Zentrum, 1995.

[14] C. Mesztenyi, A. Miller, and W. Szymczak. FEARS: Details of mathematical formulation UNIVAC 1100. Technical Report BN-994, Univ. of Maryland, College Park, 1982.

[15] L. Demkowicz, J. T. Oden, W. Rachowicz, and O. Hardy. Toward a universal *h-p* adaptive finite element strategy. i: Constrained approximation and data structure. *Comput. Methods Appl. Mech. Engrg.*, 77:79–112, 1989.

[16] C. Bernardi, Y. Maday, and A. T. Patera. A new nonconforming approach to domain decomposition: The mortar method. Technical report, laboratoire d'Analyse Numérique de Paris VI, 1990.

[17] K. Siebert. An a posteriori error estimator for anisotropic refinement. Technical report, SFB 359, Universität Bonn, 1993.

[18] G. Kunert. Ein Residuenfehlerschätzer für anisotrope Tetraedernetze und Dreiecksnetze in der Finite-Elemente-Methode. Technical Report 95-10, SPC, TU Chemnitz-Zwickau, 1995.

[19] T. Apel and F. Milde. Realization and comparison of various mesh refinement strategies near edges. Technical Report 94-15, SPC, TU Chemnitz-Zwickau, 1994.