

An Adaptive Multigrid Strategy for Convection-Diffusion Problems

Daniela Vasileva¹, Anton Kuut², and Pieter W. Hemker²

¹ Institute of Mathematics and Informatics, Bulgarian Academy of Sciences,
Acad. G. Bonchev str., bl. 8, BG-1113 Sofia, Bulgaria
vasileva@math.bas.bg

² Centrum voor Wiskunde en Informatica (CWI),
P.O. Box 94079, NL-1090 GB Amsterdam, The Netherlands
{P.W.Hemker, A.Kuut}@cwi.nl

Abstract. For the solution of convection-diffusion problems we present a multilevel self-adaptive mesh-refinement algorithm to resolve locally strong varying behavior, like boundary and interior layers. The method is based on discontinuous Galerkin (Baumann-Oden DG) discretization. The recursive mesh-adaptation is interwoven with the multigrid solver. The solver is based on multigrid V-cycles with damped block-Jacobi relaxation as a smoother. Grid transfer operators are chosen in agreement with the Galerkin structure of the discretization, and local grid-refinement is taken care of by the transfer of local truncation errors between overlapping parts of the grid.

We propose an error indicator based on the comparison of the discrete solution on the finest grid and its restriction to the next coarser grid. It refines in regions, where this difference is too large. Several results of numerical experiments are presented which illustrate the performance of the method.

Keywords: convection-dominated problems, adaptive refinement, multigrid, discontinuous Galerkin method.

1 Introduction

Recently new interest arose in application of discontinuous Galerkin (DG) methods for the solution of partial differential equations of convection-diffusion type. An important reason is their ability to conveniently handle difficulties related to grid- and order-adaptation. This motivates our present work on self-adaptive DG discretisation which is combined with a multigrid (MG) method so that optimal efficiency can be expected.

A detailed description of the multigrid approach and the corresponding smoothing analysis in the case of discontinuous Galerkin methods with constant coefficients can be found in [5, 6, 7].

The paper is organized as follows. Section 2 concerns the governing equation and its discretisation. The third section describes the multigrid h -adaptive refinement algorithm. The adaptive criterion is presented in the fourth section and the last section contains results from numerical experiments.

2 Discontinuous Galerkin Discretisation

We consider the linear boundary value problem:

$$\begin{aligned} -\varepsilon \Delta u + \nabla \cdot (\mathbf{b}u) + cu &= f \quad \text{in } \Omega \subset \mathbb{R}^d, \quad d = 1, 2, 3, \\ u(\mathbf{x}) &= u_0(\mathbf{x}) \quad \text{on } \Gamma = \partial\Omega, \end{aligned} \quad (1)$$

where $\mathbf{x} = (x_1, \dots, x_d)$, $\varepsilon > 0$ is a small parameter, the coefficients $\mathbf{b}(\mathbf{x}) = (b_1(\mathbf{x}), \dots, b_d(\mathbf{x})) \in (C^1(\Omega))^d$, $c(\mathbf{x}) \geq 0$, $c(\mathbf{x}) \in L_\infty(\Omega)$ and the right-hand side $f(\mathbf{x}) \in L_2(\Omega)$. We assume that Ω allows a regular partitioning $\Omega_h = \{\Omega_e \mid \cup_e \Omega_e = \Omega, \Omega_i \cap \Omega_j = \emptyset, i \neq j\}$, into equally sized square cells Ω_e of size h .

As weak form for (1) we use Baumann-Oden's [2, 1] discontinuous Galerkin formulation: find $u \in H^1(\Omega_h)$, such that

$$L(u, v) = F(v) \quad \text{for all } v \in H^1(\Omega_h), \quad (2)$$

where $H^1(\Omega_h)$ is the broken Sobolev space

$$H^1(\Omega_h) = \{u \in L_2(\Omega) \mid u|_{\Omega_e} \in H^1(\Omega_e), \forall \Omega_e \in \Omega_h\},$$

$$\begin{aligned} L(u, v) &= \sum_{\Omega_e \in \Omega_h} \left(\int_{\Omega_e} (\varepsilon \nabla u \cdot \nabla v - \nabla v \cdot \mathbf{b}u + cuv) \, d\mathbf{x} + \int_{\partial\Omega_e \setminus \Gamma_-} vu^- \mathbf{b} \cdot \mathbf{n} \, ds \right) \\ &\quad + \int_{\Gamma_{\text{int}} \cup \Gamma} \left(\varepsilon \langle \nabla v \rangle \cdot [u] - \varepsilon \langle \nabla u \rangle \cdot [v] \right) \, ds, \\ F(v) &= \sum_{\Omega_e \in \Omega_h} \int_{\Omega_e} vf \, d\mathbf{x} + \int_{\Gamma} \varepsilon \nabla v \cdot \mathbf{n} u_0 \, ds - \int_{\Gamma_-} vu_0 \mathbf{b} \cdot \mathbf{n} \, ds, \end{aligned}$$

$\Gamma_- = \{\mathbf{x} \in \partial\Omega \mid (\mathbf{b} \cdot \mathbf{n})(\mathbf{x}) < 0\}$ denotes the instream boundary of the domain Ω and \mathbf{n} is the unit outward pointing normal on the boundary. With u^- , the 'upwind' value of u is denoted, defined by $u^- = \lim_{\epsilon \rightarrow 0} u(\mathbf{x} - \epsilon \mathbf{b}(\mathbf{x}))$. The interior cell boundaries are denoted by $\Gamma_{\text{int}} = \cup_e \partial\Omega_e \setminus \partial\Omega$.

The jump operator $[\cdot]$ for a scalar valued function $w(\mathbf{x})$ and the average operator $\langle \cdot \rangle$ for a vector valued function $\boldsymbol{\tau}(\mathbf{x})$ are defined at the common interface $\Gamma_{i,j} = \overline{\Omega}_i \cap \overline{\Omega}_j$ between two adjacent cells Ω_i and Ω_j by

$$[w(\mathbf{x})] = w(\mathbf{x})|_{\partial\Omega_i} \mathbf{n}_i + w(\mathbf{x})|_{\partial\Omega_j} \mathbf{n}_j, \quad \langle \boldsymbol{\tau}(\mathbf{x}) \rangle = \frac{1}{2} (\boldsymbol{\tau}(\mathbf{x})|_{\partial\Omega_i} + \boldsymbol{\tau}(\mathbf{x})|_{\partial\Omega_j}).$$

For the DG discretisation of (2) we take: find $u_h \in S_h$ such that

$$L(u_h, v_h) = F(v_h) \quad \text{for all } v_h \in S_h, \quad (3)$$

where $S_h = \{\sum_e \phi_e, \phi_e \in \mathcal{P}_3(\Omega_e), \Omega_e \in \Omega_h\}$ denotes the space of piecewise cubic polynomials on the partitioning Ω_h . In order to introduce a basis of S_h we first take the polynomials basis on the one-dimensional unit interval,

$$\phi_{2n+m}(t) = t^{n+m} (1-t)^{n+1-m}, \quad n = 0, 1, \quad m = 0, 1. \quad (4)$$

Then, on the unit cube in \mathbb{R}^d we use a basis of tensor-product polynomials based on (4) and a basis for $\mathcal{P}_3(\Omega_e)$ is obtained by the usual affine mapping.

The coefficients \mathbf{b} , c and the right-hand side f are approximated using the same set of basic functions (or a set of lower order). The discretisation (3) yields a linear system $L_h u_h = f_h$, where the matrix L_h has a diagonal block structure with blocks of size $4^d \times 4^d$. We order the basic functions *cell-wise* or *point-wise* (for details see [5, 6, 7]), depending on the equation coefficients and h .

3 MG Realization on the Adaptive Grid

For the discontinuous Galerkin method we describe the application of an h -self-adaptive multi-level algorithm [3, 4].

The algorithm to determine the mesh is closely connected with the discretization and consist of several stages. In the first stage the equation is discretized and solved on the global coarsest grid Ω_0 with step size h_0 . In later stages, $k = 1, 2, \dots$, some cells of Ω_{k-1} are selected for refinement. These cells are all divided into 2^d smaller cells of equal size, which together form the grid Ω_k on a subset of $\overline{\Omega_{k-1}}$. The solution on Ω_k is interpolated from Ω_{k-1} and several relaxation sweeps are made on the interior of Ω_k , followed by a coarse grid correction on the whole of Ω_{k-1} . Thus, by recursive application, multigrid V-cycles are made. More details are given below.

By the recursive construction of the meshes Ω_k we see that the meshes cover nested areas $\overline{\Omega_k} \subset \overline{\Omega_{k-1}}$, and that all cells in these meshes form a tree-structure. In this tree, for $k > 0$ each cell has one *father* and possibly 2^d *children*.

Restrictions and prolongations. Given the nested partitioning $\{\Omega_k\}$ for the domain $\Omega = \overline{\Omega_0}$, on each mesh Ω_k we have a space of piecewise cubic polynomials S_k and the restriction of S_k to $\overline{\Omega_{k+1}}$ is a subset of S_{k+1} . This induces a natural prolongation $P_{k+1,k} : S_k \rightarrow S_{k+1}$ on $\overline{\Omega_{k+1}}$. The restriction operator *for the residues*, $\bar{R}_{k,k+1} : S_{k+1} \rightarrow S_k$ is defined as the adjoint of $P_{k+1,k}$. Because of the consistency of these operators with the DG discretisation, the Galerkin relation exists between the discretisation on the coarse and the fine grid $L_k = \bar{R}_{k,k+1} L_{k+1} P_{k+1,k}$. We use another restriction $R_{k,k+1}$ *for the solution*, which preserves the function values and the derivatives at the coarse cell vertices. It is a left-inverse of the prolongation, $R_{k,k+1} P_{k+1,k} = I_k$, where I_k is the identity operator on S_k .

The internal boundaries. In the case of local refinement the finer grid Ω_k usually covers only a part of the domain, covered by the coarser grid Ω_{k-1} . So some cells $\Omega_e \in \Omega_k$ have no neighbours on the same grid at some of their faces, but these faces will be not on the boundary $\partial\Omega$. We call this the *internal boundary*. We take care that no internal boundaries coincide for different levels k , i.e., if $\Omega_e \in \Omega_k$, then all the neighbours of its father $F(\Omega_e)$ exist on Ω_{k-1} . This can always be ensured performing some additional refinements of neighbouring cells.

At the internal boundaries for the discretization on Ω_k we take Dirichlet boundary conditions, derived by interpolation from Ω_{k-1} .

The relative truncation error. If u_k is the solution of the fine grid system, i.e., $L_k u_k = f_k$, then its restriction to the coarser grid Ω_{k-1} satisfies

$$L_{k-1} R_{k-1,k} u_k = \bar{R}_{k-1,k} f_k + \tau_{k-1,k}(u_k),$$

where the relative local truncation error $\tau_{k-1,k}(y_k)$ is defined by

$$\tau_{k-1,k}(y_k) = L_{k-1}R_{k-1,k}y_k - \bar{R}_{k-1,k}L_k y_k.$$

During the computation, however, we do not know the fine grid solution u_k , but only an approximation \tilde{u}_k . So to obtain an accurate solution on the coarse grid, that corresponds to the solution on the fine grid (where this exists), we solve the coarse grid system $L_{k-1}\tilde{u}_{k-1} = \tilde{f}_{k-1}$, where

$$\tilde{f}_{k-1} = \begin{cases} f_{k-1} & \text{on } \overline{\Omega_{k-1}} \setminus \overline{\Omega_k}, \text{ where no finer grid exists,} \\ \bar{R}_{k-1,k}\tilde{f}_k + \tau_{k-1,k}(\tilde{u}_k) & \text{on } \overline{\Omega_k}, \text{ where } \tilde{u}_k \text{ is the current approximation on the finer grid.} \end{cases}$$

MG iteration. Each multigrid V-cycle on level k , denoted by $MS(k, \nu_1, \nu_2)$, consists of the following steps:

1. Perform ν_1 (pre-) relaxation steps (damped block-Jacobi relaxation) on the discrete system $L_k\tilde{u}_k = \tilde{f}_k$, taking as initial approximation

$$\tilde{u}_k = \begin{cases} R_{k,k+1}\tilde{u}_{k+1} & \text{on } \overline{\Omega_{k+1}}, \\ \tilde{u}_k & \text{on } \overline{\Omega_k} \setminus \overline{\Omega_{k+1}}; \end{cases}$$

2. If $k > 0$
 - perform $MS(k-1, \nu_1, \nu_2)$;
 - Compute the correction $\tilde{u}_k = \tilde{u}_k + P_{k,k-1}(\tilde{u}_{k-1} - R_{k-1,k}\tilde{u}_k)$;
3. With the current approximation for \tilde{u}_k perform ν_2 (post-) relaxation steps on the discrete system $L_k\tilde{u}_k = \tilde{f}_k$.

On the coarsest level $MS(0, \nu_1, \nu_2)$ consists of $\nu_1 + \nu_2$ relaxation sweeps on Ω_0 . Because this is a very coarse grid, an alternative way to solve $L_0\tilde{u}_0 = \tilde{f}_0$ is by Gaussian elimination.

4 Adaptation Criterion

On level Ω_0 all cells are refined at least once. Let K be the current number of grid levels. In order to decide which unrefined cells in Ω_k , $k = 1, 2, \dots, K$, in the structure will be further refined, we compare the current solution $u_k(\mathbf{x})$ and its restriction to the previous level, $R_{k-1,k}u_k(\mathbf{x})$. The reason for this choice is the following. Let $u(\mathbf{x})$ be the exact solution of our problem (1) and let $R_l u(\mathbf{x})$ be its restriction to Ω_l , where (similar to $R_{l,l+1}$) the restriction $R_l : H^1(\Omega) \rightarrow S_l$ preserves the function values and the derivatives at the vertices of each $\Omega_e \in \Omega_l$. Then, with $u \in C^M(\Omega_e)$,

$$\|R_{l+1}u - R_l u\|_{C(\Omega_e)} < C\|u\|_{C^M(\Omega_e)}2^{-lM}, \quad \text{for } M = 1, 2, 3, 4.$$

From this we derive that for piecewise C^M -functions, $M \leq 4$, and *asymptotically for large k and l*

$$\|R_{l+1}u - R_l u\|_{C(\Omega_e^k)} \lesssim q^{l+1-k}\|R_k u - R_{k-1}u\|_{C(\Omega_e^{k-1})} \quad \text{for } l \geq k, \quad (5)$$

with $q = 2^{-M}$. Here $\Omega_e^k \in \Omega_k$ and $\Omega_e^{k-1} := F(\Omega_e^k)$ is its father. For finite $l \geq k$ estimate (5) may be not true: then always a smooth function $u(\mathbf{x})$ can be constructed such that $\|R_k u - R_{k-1} u\|_{C(\Omega_e^{k-1})} = 0$, but $\|R_{l+1} u - R_l u\|_{C(\Omega_e^k)} > 0$ for $l \geq k$.

Let $\tilde{R}u$ be the restriction of u to the unrefined cells of $\cup_{k=1}^K \Omega_k$, where for each unrefined cell $\Omega_e^k \in \Omega_k$ we define $\tilde{R}u := R_k u$, then under the assumption of (5) we estimate $\|u - \tilde{R}u\|_{C(\Omega_e^k)}$ only by using $\|R_k u - R_{k-1} u\|_{C(\Omega_e^{k-1})}$ and q . As during the computation we do not know $R_k u$ and $R_{k-1} u$, we use the best available approximant u_k . With $r_k := \|u_k - R_{k-1,k} u_k\|_{C(\Omega_e^{k-1})}$,

$$\|u - R_k u\|_{C(\Omega_e^k)} \leq \sum_{l=k}^{\infty} \|R_{l+1} u - R_l u\|_{C(\Omega_e^k)} \lesssim \sum_{l=k}^{\infty} q^{l+1-k} r_k = \frac{q}{1-q} r_k.$$

With \mathbb{T} a desired tolerance and $r_k \frac{q}{(1-q)} \leq \mathbb{T}$ for all unrefined $\Omega_e^k \in \cup_{k=1}^K \Omega_k$, we have

$$\|u - \tilde{R}u\|_{C(\Omega)} = \max_{\Omega_e^k} \|u - R_k u\|_{C(\Omega_e^k)} \lesssim \max_{\Omega_e^k} r_k \frac{q}{(1-q)} \leq \mathbb{T}.$$

Notice that we can estimate the local smoothness of the solution by estimating q from computable equivalents of (5). Let

$$r_l := \|R_l u_k - R_{l-1} u_k\|_{C(\Omega_e^{l-1})}, \quad l \leq k,$$

where $\Omega_e^{l-1} := F(\Omega_e^l)$, $l \leq k$. Then asymptotically we expect $q = \lim_{l \rightarrow \infty} r_l / r_{l-1}$. Based on this relation and additional heuristics we arrive at an estimated local value q_{est} . Then we introduce the local error estimate $\eta(\Omega_e^k) := r_k \frac{q_{\text{est}}}{(1-q_{\text{est}})}$ and we refine cell Ω_e^k if

$$\eta(\Omega_e^k) > \mathbb{T} \quad \text{or} \quad q_{\text{est}} \geq 1.$$

As the amount of work, needed to compute r_l in the 2D and 3D case, can not be neglected, in the examples below we only estimate r_l even in the 1D case.

5 Examples

In all examples in this section, the gridrefinements started on an initial grid with mesh size 1. Ten multigrid sweeps with 1 pre- and 1 post-smoothing iteration ($\nu_1 = \nu_2 = 1$) are performed in each stage.

Example 1. We consider the one dimensional equation

$$\varepsilon u'' + (x-1)u' = -\varepsilon\pi^2 \cos(\pi(x-1)) - \pi(x-1) \sin(\pi(x-1)), \quad x \in (0, 2),$$

with Dirichlet boundary conditions, corresponding to the following exact solution

$$u(x) = \operatorname{erf}\left((x-1)/\sqrt{2\varepsilon}\right) + \cos(\pi(x-1)).$$

In Fig. 1 the exact solution and the difference between the exact and the approximate solution are plotted for $\varepsilon = 0.0001$ and $\mathbb{T} = 0.01$. The corresponding

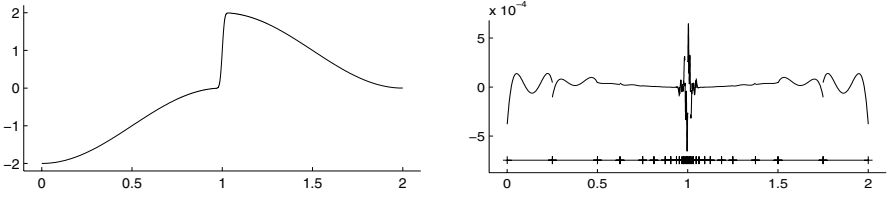


Fig. 1. The exact solution for Example 1 and the corresponding error for $T = 0.01$

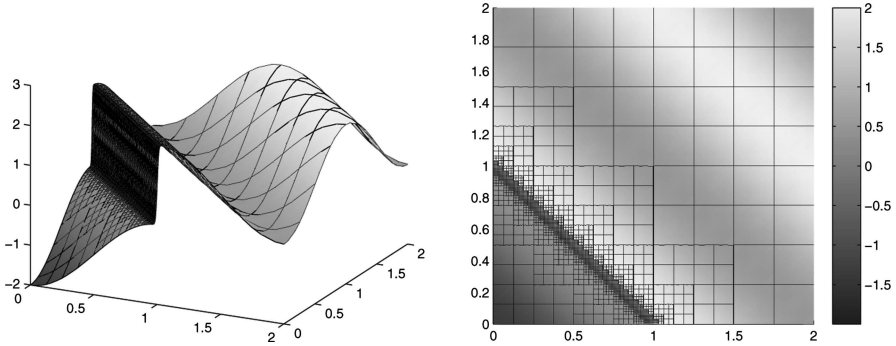


Fig. 2. The approximate solution for Example 2

grid is shown at the bottom of the right picture: 8 levels are used in order to achieve the prescribed tolerance. Similar results are obtained for $T = 0.001$, then 9 levels are used. In both cases the grid is properly refined in the interior layer area and the C -norm of the error is less than T . The total number of cells in the final grid is respectively $N = 28$ and $N = 32$.

Example 2. In two dimensions a similar problem is considered, with an interior layer skew to the mesh:

$$\varepsilon \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + (x + y - 1) \left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \right) = f(x, y), \quad x \in (0, 2), \quad y \in (0, 2),$$

$$f(x, y) = -2\varepsilon\pi^2 \cos(\pi(x + y - 1)) - 2\pi(x + y - 1) \sin(\pi(x + y - 1)),$$

with Dirichlet boundary conditions, corresponding to the following exact solution

$$u(x, y) = \operatorname{erf} \left((x + y - 1) / \sqrt{2\varepsilon} \right) + \cos(\pi(x + y - 1)).$$

In Fig. 2 the approximate solution is plotted for $\varepsilon = 0.0001$ and $T = 0.01$. The grid is refined around the interior layer and 8 levels are used to achieve the prescribed tolerance. The C -norm of the difference between the exact and the approximate solution is less than the prescribed tolerance.

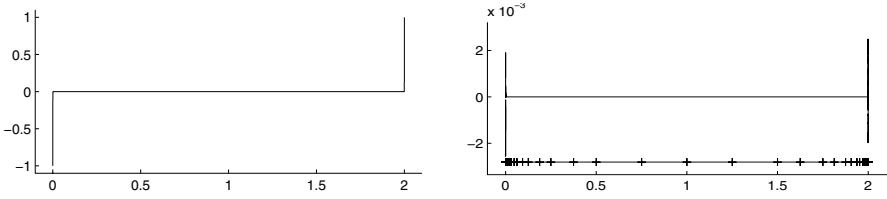


Fig. 3. The exact solution for Example 3 and the corresponding error for $\mathbb{T} = 0.01$

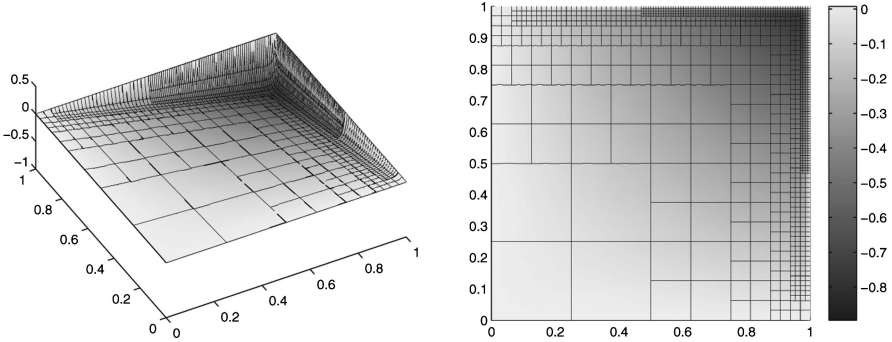


Fig. 4. The approximate solution for Example 4

Example 3. The third example is a one-dimensional problem with a turning point and two boundary layers:

$$\varepsilon u'' - (x - 1)u' - u = 0, \quad x \in (0, 2),$$

with Dirichlet boundary conditions, corresponding to the following exact solution

$$u(x) = \frac{\operatorname{erf}((x - 1)/\sqrt{2\varepsilon}) \exp((x - 1)^2/2\varepsilon)}{\operatorname{erf}(1/\sqrt{2\varepsilon}) \exp(1/2\varepsilon)}.$$

In Fig. 3 the exact solution and the difference between the exact and the approximate solution are plotted, for $\varepsilon = 0.0001$ and $\mathbb{T} = 0.01$. To achieve the result the algorithm uses 14 levels and the resulting C -norm of the error is less than 0.01. If we require $\mathbb{T} = 0.001$, 15 levels are used and the maximal error is less than 0.001.

Example 4. The following two dimensional problem is considered

$$\varepsilon \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - x \frac{\partial u}{\partial x} - y \frac{\partial u}{\partial y} - 2u = f(x, y), \quad x \in (0, 1), \quad y \in (0, 1),$$

the right-hand side $f(x, y)$ and Dirichlet boundary conditions correspond to the exact solution

$$u(x, y) = -xy(1 - \exp((x - 1)/\varepsilon))(1 - \exp((y - 1)/\varepsilon)).$$

The approximate solution for $\varepsilon = 0.01$ and $T = 0.01$ is plotted in Fig. 4. The grid is properly refined around the boundary layers and the C -norm of the error is less than 0.01. The algorithm uses 8 levels. Note, in this case the solution is sufficiently smooth and if we take $q = 2^{-4}$, almost the same grid refinement is achieved.

6 Conclusion

One- and two-dimensional numerical experiments demonstrate that the proposed self-adaptive mesh-generation, embedded in a multigrid strategy and applied with the Baumann-Oden discontinuous Galerkin method, can be successfully used for the automatic resolution of boundary and interior layers in the solution of convection-dominated problems. The strategy is based on the comparison of the numerical approximation on the finest and the one-but-finest grid. Thus, it makes use of the local regularity of the solution. The method used to estimate the local regularity, q_{est} , is still based on heuristic arguments. Although the numerical experiments are quite satisfying, the mathematical motivation needs a more solid theoretical base. This is a subject for further research.

Acknowledgment. D. Vasileva acknowledges the fellowship from the European Consortium for Informatics and Mathematics (ERCIM) for her nine months work at CWI.

References

1. Baumann, C.E.: An hp -adaptive discontinuous finite element method for computational fluid dynamics. PhD thesis, the University of Texas at Austin (1997)
2. Baumann, C.E., Oden, J.T.: A discontinuous hp finite element method for convection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.*, **175** (1999) 311–341
3. Brandt, A.: Multi-level adaptive solutions to boundary value problems. *Math. Comp.*, **31** (1977) 333–390
4. Hemker, P.W.: On the structure of an adaptive multi-level algorithm. *BIT*, **20** (1980) 289–301
5. Hemker, P.W., Hoffmann, W., van Raalte, M.H.: Two-level Fourier analysis of a multigrid approach for discontinuous Galerkin discretisation. *SIAM J. Sci. Comp.*, **25** (2004) 1018–1041
6. Hemker, P.W., van Raalte, M.H.: Fourier two-level analysis for higher dimensional discontinuous Galerkin discretisation. *Computing and Visualization in Science*, **7** (2004) 159–172.
7. van Raalte, M.H.: Multigrid Analysis and Embedded Boundary conditions for discontinuous Galerkin Discretization. PhD thesis, the University of Amsterdam (2004)