SpezialForschungsBereich F 32

Karl–Franzens Universität Graz
Technische Universität Graz
Medizinische Universität Graz

An implementation of hybrid discontinuous Galerkin methods in DUNE

C. Waluga H. Egger

SFB-Report No. 2010–041 Dezember 2010

A–8010 GRAZ, HEINRICHSTRASSE 36, AUSTRIA

Supported by the
Austrian Science Fund (FWF)

SFB sponsors:

- **Austrian Science Fund** (FWF)
- **University of Graz**
- **Graz University of Technology**
- **Medical University of Graz**
- **Government of Styria**
- **City of Graz**

An implementation of hybrid discontinuous Galerkin methods in DUNE

Christian Waluga and Herbert Egger

Abstract We discuss the implementation of hybrid finite element methods in the Distributed and Unified Numerics Environment (DUNE). Such hybrid methods require the approximation of the solution in the interior of the elements, as well as an approximation of the traces of the solution on the element interfaces, i.e., the skeleton. For illustration, we consider a hybrid version of the interior penalty discontinuous Galerkin method for an elliptic model problem. In order to realize the implementation in the DUNE framework, we present a generic extension of the C++ template library DUNE-PDELAB [4] for problems with additional polynomial approximations of spaces defined on the skeleton.

1 Introduction

Let Ω be a bounded polyhedral domain in $\mathbb{R}^d$ with boundary $\partial\Omega = \partial\Omega^D \cup \partial\Omega^N$ consisting of two regular components, a Dirichlet and Neumann part. We consider the Poisson problem with mixed boundary conditions: Given $f \in L^2(\Omega)$, $g \in H^{1/2}(\partial\Omega^D)$, and $h \in L^2(\partial\Omega^N)$, find $u \in H^1(\Omega)$, such that

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega^D \\ \frac{\partial u}{\partial n} = h & \text{on } \partial\Omega^N. \end{cases} \tag{1}$$

Christian Waluga
Aachen Institute for Advanced Study in Computational Engineering Science (AICES), RWTH Aachen University, Schinkelstraße. 2, 52062 Aachen, Germany, e-mail: waluga@aices.rwth-aachen.de

Herbert Egger
Institute for Mathematics in Industry and Technology, TU Chemnitz, Reichenhainerstraße 41, 09126 Chemnitz, Germany, e-mail: herbert.egger@mathematik.tu-chemnitz.de

This problem serves as a model for various physical phenomena, e.g. electrostatics or diffusion processes. In heat conduction, for instance, u denotes the temperature distribution, f models internal heat sources, g is the prescribed temperature at the Dirichlet part, and h the heat flux over the Neumann part of the boundary

2 A hybrid variational principle

In the following, we propose a variational principle, which is satisfied by any regular solution of the Poisson problem (1), and which will be the basis for the hybrid finite element method studied in this paper.

2.1 Preliminaries

Let $\mathscr{T}_h = \{T\}$ be a partition of the domain Ω into a set of simple elements T, such as triangles and quadrilaterals, or tetrahedral, pyramidal, prismatic, and hexahedral elements in two and three spaces dimensions, respectively. We denote by $\mathscr{E}_h^0 := \{E_{ij} = \partial T_i \cap \partial T_j : i > j\}$ the set of intersections between neighboring elements, and $\mathscr{E}_h^D := \{E_i : E_i = \partial T_i \cap \partial \Omega^D\}$ and $\mathscr{E}_h^N := \{E_i : E_i = \partial T_i \cap \partial \Omega^N\}$, are the intersections of the elements with the Dirichlet and Neumann boundary, respectively. The collection of all facets is denoted by $\mathscr{E}_h := \mathscr{E}_h^0 \cup \mathscr{E}_h^D \cup \mathscr{E}_h^N$, and the union $\mathscr{E} := \bigcup_{E \in \mathscr{E}_h} E$ of all interfaces is referred to as the *skeleton*.

2.2 A weak formulation

For $s \geq 0$, let us introduce the broken Sobolev spaces

$$H^s(\mathscr{T}_h) := \{v \in L^2(\Omega) : v_h|_T \in H^s(T) \quad \forall T \in \mathscr{T}_h\} \tag{2}$$

defined on a partition $\mathscr{T}_h$ of the domain Ω, and set $V := H^2(\mathscr{T}_h)$. For $g \in L^2(\partial\Omega^D)$, we denote by

$$\widehat{V}^g := \{\hat{v} \in L^2(\mathscr{E}_h) : \hat{v}_h = g \quad \text{on } \partial\Omega^D\} \tag{3}$$

the set of functions defined over the skeleton with prescribed values on the Dirichlet boundary. As a weak form of the Poisson problem (1), we then consider the following *variational principle*: Find $(u, \hat{u}) \in V \times \widehat{V}^g$ such that

$$a_h(u, \hat{u}, v, \hat{v}) = f_h(v), \qquad \forall (v, \hat{v}) \in V \times \widehat{V}^0, \tag{4}$$

with bilinear and linear forms defined by

$$a_h(u,\hat u,v,\hat v) := \sum_{T\in\mathscr{T}_h}\int_T \nabla u\cdot\nabla v\,dx - \sum_{T\in\mathscr{T}_h}\int_{\partial T}\frac{\partial u}{\partial n}(v-\hat v)\,ds$$
$$- \sum_{T\in\mathscr{T}_h}\int_{\partial T}(u-\hat u)\frac{\partial v}{\partial n}\,ds + \sum_{T\in\mathscr{T}_h}\tau\int_{\partial T}(u-\hat u)(v-\hat v)\,ds,$$

and

$$f_h(v) := \sum_{T\in\mathscr{T}_h}\int_T f\,v\,dx + \sum_{E\in\mathscr{E}_h^N}\int_E hv\,ds.$$

Here, n denotes the outward pointing normal vector on ∂T and τ is a piecewise constant positive stabilization parameter, which will be chosen later.

Theorem 1. *Let $u\in H^1(\Omega)\cap H^2(\mathscr{T}_h)$ denote a solution of the Poisson problem* (1)*, and define $\hat u = u|_{\mathscr{E}}$. Then the tuple $(u,\hat u)$ also solves the variational problem* (4).

Proof. The result follows directly from the definition of the bilinear and linear forms, integration-by-parts, and noting that the flux $\frac{\partial u}{\partial n}$ is continuous across element interfaces.

Remark 1. The extra regularity of the solution u is only required, to ensure that the normal derivative $\frac{\partial u}{\partial n}$ is well defined on single edges. This requirement can be relaxed in several ways, e.g., $u\in H^{3/2+\varepsilon}(\mathscr{T}_h)$ is sufficient to obtain $\frac{\partial u}{\partial n}\in L^2(\mathscr{E}_h)$. Alternatively, one may define $V = H^1(\mathscr{T}_h)$ and $\widehat{V}^g = \{\hat v\in H^{1/2}(\mathscr{E}) : \hat v = g \quad \text{on } \partial\Omega^D\}$, in which case any solution of the Poisson problem satisfies the variational principle. If the scalar product of $H^{1/2}(\mathscr{E})$ is utilized for penalization instead of the weighted L^2-term, then (4) completely characterizes weak solutions of (1), i.e., any solution of the variational formulation is also a solution of the Poisson problem [6].

3 Discretization by finite elements

For ease of presentation, we assume in the following, that $\mathscr{T}_h$ is a shape-regular, quasi-uniform simplicial mesh. Let us denote by $\mathscr{P}^p(T)$ and $\mathscr{P}^p(E)$ the space of polynomials of maximal order p on T and E, respectively. For approximation of the variational principle (4), we utilize the following finite element spaces

$$V_h := \{v_h\in L^2(\Omega) : v_h|_T\in\mathscr{P}^p(T)\quad \forall T\in\mathscr{T}_h\},\qquad \text{and}$$
$$\widehat{V}_h := \{\hat v_h\in L^2(\mathscr{E}) : \hat v_h|_E\in\mathscr{P}^p(E)\quad \forall E\in\mathscr{E}_h\}.$$

In order to incorporate the Dirichlet boundary conditions, we further define the spaces

$$\widehat{V}_h^g := \{v\in\widehat{V}_h : v|_E = \pi_E g,\quad E\in\mathscr{E}_h^D\}, \tag{5}$$

where π_E is the orthogonal projection onto $\mathscr{P}^p(E)$. The space of functions, defined on the skeleton, which vanish on $\partial\Omega^D$ is denoted accordingly by $\widehat{V}_h^0$. Figure 1 displays the association of the degrees of freedom with elements and intersections for a linear approximation on a triangular mesh.

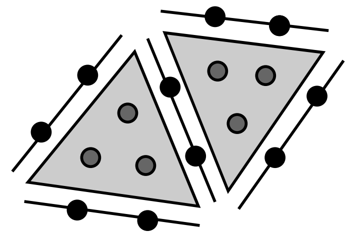

Fig. 1 Degrees of freedom associated with a hybridized discontinuous Galerkin approximation of first order. Black bullets indicate the globally coupled hybrid degrees of freedom and gray bullets denote (inner) element degrees of freedom.

3.1 The discrete method

As approximation for the Poisson problem, we then consider the following finite element discretization of the variational principle (4).

Problem 1 (Hybrid discontinuous Galerkin method). Given $f \in L^2(\Omega)$, $g \in H^{1/2}(\partial\Omega^D)$, and $h \in L^2(\partial\Omega^N)$, find $(u_h, \hat{u}_h) \in V_h \times \widehat{V}_h^g$ such that

$$a_h(u_h, \hat{u}_h, v_h, \hat{v}_h) = f_h(v_h) \qquad \text{for all } v_h \in V_h \text{ and } \hat{v}_h \in \widehat{V}_h^0. \tag{6}$$

To ensure well-definedness of the method, we require that the stabilization parameter τ is chosen as $\tau \geq \alpha \frac{p^2}{h}$ with α depending on the shape regularity of the mesh.

Remark 2. Note that the Dirichlet boundary conditions are explicitly built into the hybrid solutions space $\widehat{V}_h^g$, and no conditions are imposed on the functions in V_h.

Remark 3. Problem 1 is a hybrid version of the symmetric interior penalty method [1]; non-symmetric versions can be defined accordingly. The idea of hybridization goes back at least to [2], where it was used to facilitate the implementation of mixed methods. The approach has been generalized to a large class of methods using discontinuous approximation spaces in [3]. The hybridization of discontinous Galerkin methods has been investigated as a general framework in [5]; see also [6] for a-priori error estimates.

3.2 A-priori error analysis

The analysis of the discrete problem can be carried out with respect to the following mesh-dependent energy norm

$$\|(v,\hat{v})\|_{1,h}^2 := \sum_{T\in\mathscr{T}_h} \left(\|\nabla v\|_T^2 + \|\tau^{1/2}(v-\hat{v})\|_{\partial T}^2 \right).$$

The boundedness of a_h and f_h in the discrete norm follows immediately by the Cauchy-Schwarz inequality, a discrete Friedrichs-inequality, and discrete trace inequalities. The latter also imply the coercivity of the bilinear form a_h; for details, see [6].

Proposition 1. *Let α be sufficiently large. Then for all $(v_h,\hat{v}_h) \in V_h \times \widehat{V}_h^0$, there holds*

$$a_h(v_h,\hat{v}_h;v_h,\hat{v}_h) \geq \tfrac{1}{2}\|(v_h,\hat{v}_h)\|_{1,h}^2.$$

Based on Galerkin orthogonality, the error in the finite element solution can now be bounded by the best-approximation error, which yields the following estimates.

Theorem 2 (A-priori estimates). *Let $u \in H^{s+1}(\mathscr{T}_h)$ denote the solution of the Poisson problem* (1). *Then*

$$\|(u-u_h, u-\hat{u}_h)\|_{1,h} \leq C_s \tfrac{h^\mu}{p^{\mu-1/2}} \|u\|_{s+1,h},$$

with $\mu = \min\{p,s\}$ and constant C_s independent of h, p, and u.

Remark 4. Using the standard duality argument, one can also obtain order optimal rates with respect to the L^2-norm. For a-posteriori estimates derived in the spirit of [13, 12] for mixed and discontinuous Galerkin methods, see e.g. [7, 9].

3.3 Additional aspects

Let us shortly make some remarks concerning generalizations and some implementational aspects of hybrid discontinuous Galerkin methods.

Remark 5 (Generalizations). For ease of presentation, we considered only the case of quasi-uniform simplicial meshes above. The a-priori analysis can easily be generlized to shape-regular meshes, varying polynomial degree, various element types, and hanging nodes; see e.g., [12]. The hybrid DG framework can even be used for mortaring [5, 6, 10].

Remark 6 (Choice of stabilization parameter). While the ellipticity estimate does not deteriorate, if the stabilization parameter α is chosen very large, the conditioning of the system matrix is negatively affected. Since the lower bound for α only depends on discrete trace inequalities [18, 11], sharp explicit lower bounds for α can be computed on simplicial elements or hyper-cubes.

Remark 7 (Static condensation). The discrete variational problem (6) has the form

$$\sum_{T\in\mathscr{T}_h} A_T(u_T,\hat{u}_T) = \sum_{T\in\mathscr{T}_h} f_T,$$

i.e., in contrast to standard discontinuous Galerkin methods, the assembling can be performed elementwise. Since the degrees of freedom associated with the primal space V_h do not couple over element interfaces, the primal variables u_h can be completely eliminated during assembly by *static condensation*. The global system then only involves the hybrid variables $\hat{u}_h$. In the simplest case, one can obtain a global system with the same sparsity pattern as a non-conforming P^1 discretization [8, 6].

4 Implementation in the DUNE framework

In this section, we discuss the implementation of the hybrid DG method, introduced in the previous section, in DUNE[1]. An important ingredient here is the definition of degrees of freedom associated with the interface mesh $\mathscr{E}_h$. We therefore discuss the construction of the approximation space $\widehat{V}_h$ for the hybrid variable in some detail, and present a technique for incorporating the Dirichlet boundary conditions.

In the following, let `R` denote a real value type and `GV` a DUNE grid-view type. The constant `p` represents the polynomial degree p, which is here specified at compile-time. For illustration, we present short snippets of pseudocode; for ease of notation, we assume that the namespaces `Dune` and `Dune::PDELab` are used.

4.1 Construction of the discrete discontinuous Galerkin space V_h

For the definition of the discontinuous Galerkin finite element space V_h, we make use of DUNE-PDELAB's monomial finite element maps. This specialization of a finite element map defines degrees of freedom in the interior of each element $T \in \mathscr{T}_h$. The monomial shape functions of maximal order p are defined in a generic way on a d-dimensional reference element. An instantiation of V_h would look as follows:

```
typedef MonomLocalFiniteElementMap<R, R, d, p> VMap;
typedef GridFunctionSpace<GV, VMap, NoConstraints,
                              Backend> VSpace;

// create the FE map and space for V_h
VMap vmap(GeometryType::simplex);
VSpace vgfs(gv, vmap);
```

Note, that we do not constrain any of the degrees of freedom in V_h.

[1] The implementation presented in this chapter is from the DUNE-module DUNE-HDG [17], which is licensed under version 2 of the GNU General Public License, with the so-called *runtime exception*.

4.2 Construction of the discrete hybrid interface space $\widehat{V}_h$

For the implementation of the hybrid finite element space $\widehat{V}_h$, we have to define degrees of freedom on the interface mesh $\mathscr{E}_h$. In order to do this, we implemented a new monomial finite element map, which provides monomial finite elements of co-dimension 1, whose degrees of freedom are associated with intersections.

The implementation, of this new finite element map is available in the class template `FacetMonomLocalFiniteElementMap` in DUNE-HDG [17]. It is generic and allows all available element-geometry-types and an arbitrary order of approximation. Therefore, the spatial dimension d and the polynomial degree p have to be provided as template parameters for the finite element map. An instantiation of $\widehat{V}_h$ would look as follows:

```
typedef IntersectionIndexSet<GV> IIS;
typedef FacetMonomLocalFiniteElementMap<IIS, R, R,
                                        d, p> HVMap;
typedef GridFunctionSpace<GV, HVMap, FacetConstraints,
  Backend, GridFunctionStaticSize<IIS> > HVSpace;

IIS iis(gv); // the intersection index set

// create the FE map and space for \hat{V}_h
HVMap hvmap(iis, GeometryType::simplex);
HVSpace hvgfs(gv, hvmap, iis);
```

Let us remark that the grid function space is not created in the usual way. We have to provide an intersection index set, which is implemented in DUNE-PDELAB, as a parameter to the `GridFunctionSpace` object. The specialized implementation of the grid function space is then able to define degrees of freedom associated with intersections in our mesh. Also, since we will later constrain the boundary degrees of freedom in $\widehat{V}_h$, we add the newly developed `FacetConstraints` to the template parameters.

4.3 Construction of the composite space $V_h \times \widehat{V}_h$

The generic assembly procedures in DUNE-PDELAB expect a single grid function space as an argument. We thus combine the two previously defined grid function spaces in a standard composite grid function space as follows:

```
typedef CompositeGridFunctionSpace<GFSMapper,
                                   VSpace, HVSpace> GFS;

// create the FE space V_h \times \hat{V}_h
GFS gfs(vgfs, hvgfs);
```

4.4 Boundary conditions

While the Neumann boundary conditions are imposed in a weak sense, the Dirichlet boundary conditions are built into the finite element space $\widehat{V}_h^g$. We do this by constraining the corresponding degrees of freedom associated with the boundary facets by using the `FacetConstraints` as a parameter in the definition of $\widehat{V}_h$, cf. section 4.2.

In order to know, which boundary degrees of freedom need to be constrained, we need a function that evaluates the type of boundary condition at a point $x \in \partial\Omega$. As we have a composite space, we also need a composite grid function here.

```
typedef DummyBoundaryGridFunction<GV>     BCV;
typedef DirichletBoundaryGridFunction<GV> BChV;
typedef CompositeGridFunction<BCV, BChV>  BC;

BCV  bcv(gv);   // dummy constraints for V_h
BChV bchv(gv);  // Dirichlet constraints for \hat{V}_h

// composite for V_h \times \hat{V}_h
BC   bc(bcv, bchv);
```

For the first component, which remains unconstrained, we provide a dummy boundary grid function that does nothing. The second component, i.e. the hybrid part of our function space, has Dirichlet boundary conditions on the whole of $\partial\Omega$ in the example above. The constraints map is the created in the usual way:

```
typedef GFS::ConstraintsContainer<R>::Type C;
C constr;
constraints(bc, gfs, constr);
```

The actual values of the coefficients to be constrained can be determined using the helper function `interpolate_trace`, which can be used to project g to the discrete space $\widehat{V}_h$. This function mimics the behavior of DUNE-PDELAB's function `interpolate`, but works on the traces in $\mathscr{E}_h$ of elements. We apply this function only to the second sub-space of our composite grid function space:

```
typedef GFS::VectorContainer<R>::Type V;
V x(gfs, 0.0); // initialize vector of unknowns
G g(gv);       // function type: u=g on boundary

// select first sub-space from composite
typedef GridFunctionSubSpace<GFS, 1> HVSubSpace;
HVSubSpace hvsub(gfs);

// determine the constrained coefficients
interpolate_trace(g, hvsub, x, false);
```

In order to avoid unnecessary interpolations here, the last parameter `doSkeleton` of `interpolate_trace` can be set to **`false`**, which means that only boundary coefficients are determined.

4.5 The residual operator

The implementation of a local operator in DUNE-PDELAB is based on a weighted residual formulation: Find $(u_h, \hat{u}_h) \in V_h \times \widehat{V}_h^g$, such that

$$\mathscr{R}_h(u_h, \hat{u}_h, v_h, \hat{v}_h) = 0, \qquad \forall\,(v_h, \hat{v}_h) \in V_h \times \widehat{V}_h^0,$$

where $\mathscr{R}_h(u_h, \hat{u}_h, v_h, \hat{v}_h) := a_h(u_h, \hat{u}_h, v_h, \hat{v}_h) - f_h(v_h)$. This can be split up into localized contributions, that can be separately assembled on each volume element in $\mathscr{T}_h$, inner intersection in ${\mathscr{E}_h}^0$ and boundary intersection, in ${\mathscr{E}_h}^D$ and ${\mathscr{E}_h}^N$.

Remark 8. The current implementation of our local operators computes the contribution of the two neighboring elements T_i and T_j to the integrals over an intersection $E_{ij} \in \mathscr{E}_h^0$ at once. This is due to the fact that we need a uniquely defined orientation in our intersection. The current DUNE-Interface however cannot guarantee that an intersection E_{ij} has the same orientation as its counterpart E_{ji} in the neighboring element. This can be seen as an disadvantage of the current implementation, since we sacrifice the locality of our element matrix assembly. The possibility of an efficient implementation of the static condensation procedure (cf. Remark 7) is thus clearly limited.

The implementation of the local operator for the Poisson problem is based on the discrete variational problem (6), and can be found in the template classes `HDGPoisson` and `HDGPoissonJacobian` in the DUNE-HDG module [17]. The functions τ, f and j are template parameters for the local operator and need to be specified for an actual problem. For the computation of τ, there are several default implementations, including an optimal version `OptimalStabilityParameter`, which computes τ with help of the exact expressions of α, which are known for simplex and hyper-cube elements in arbitrary dimensions (cf. Remark 6).

As an example, we show the creation of the grid operator space here:

```
typedef ConstantFunction<GV, R> CF;
typedef OptimalStabilityParameter<GV> Tau;

CF f(gv, 0.0);      // source term: f=0
CF j(gv, 0.0);      // Neumann: j=0 here
Tau tau(p, alpha); // stability parameter

typedef HDGPoissonJacobian<CF, BC, CF, Tau> LocalOp;
typedef GridOperatorSpace<GFS, GFS, LocalOp,
                          C, C, Backend> GOS;
```

```
// create local operator and grid operator space
LocalOp lop(f, bc, j, t);
GOS gos(gfs, constr, gfs, constr, lop);
```

4.6 Additional remarks

Our implementation naturally supports non-matching meshes, since we require that we have a sufficient number of degrees of freedom on each intersection. Using a suitable local error estimator (cf. e.g. [7, 9]), we can then employ the standard mesh adaption routines implemented in DUNE to obtain a generic h-adaptive finite element code.

The system is then assembled in the usual way, which is described in [4]. We solve the problem with a linear solver provided in the iterative solver template library DUNE-ISTL. For visualization, we export a representation of the discrete function u_h to the visualization toolkit (VTK) format.

5 Numerical results

In this section we present some numerical results that were obtained using the DUNE-HDG implementation on a triangular ALUGRID .

We consider a stationary irrotational and incompressible flow velocity field $u(x,y)$ around a sphere with a radius $R = 1/4$ in two dimensions. Since $\nabla \times u = 0$ by assumption, there exists a velocity potential ψ such that $\nabla \psi = (u_2, -u_1)$. In the case of an incompressible flow, the velocity potential ψ then satisfies Laplace's equation: $-\Delta \psi = 0$.

We solve for the velocity potential on $\Omega := (-1,1) \times (-\frac{1}{2}, \frac{1}{2})$ using the hybrid discontinuous Galerkin formulation presented in the previous sections. The boundary conditions are chosen in agreement with the exact solution

$$\psi(x,y) = y\frac{x^2 + y^2 - R^2}{x^2 + y^2}.$$

Figure 2 shows a solution plot for ψ on a triangular mesh. Note that the streamlines of the velocity field $u(x,y)$ coincide with the isolines of its potential function $\psi(x,y)$. In Table 1, we display the numerically observed convergence rates for $p = 2$, which are in good agreement with the theoretical predictions of Theorem 2.

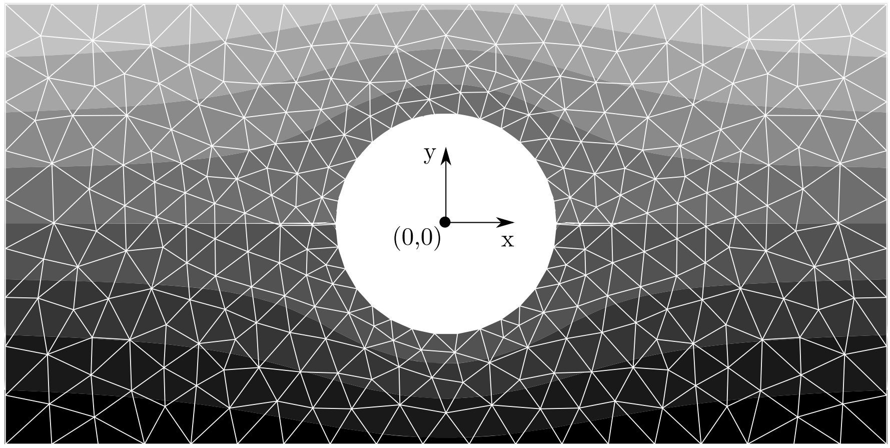

Fig. 2 Triangular mesh (level 0) and plot of ψ for the potential flow around a sphere.

level	L^2	rate	energy	rate
0	$3.22565 \cdot 10^{-5}$	–	$4.64175 \cdot 10^{-3}$	–
1	$4.04213 \cdot 10^{-6}$	3.00	$1.19992 \cdot 10^{-3}$	1.95
2	$5.02869 \cdot 10^{-7}$	3.01	$3.03856 \cdot 10^{-4}$	1.98

Table 1 Errors of the numerical solution of the potential flow flow problem for a second order finite element approximation, and a sequence of uniformly refined meshes.

6 Conclusion

We investigated the implementation of a hybrid discontinuous Galerkin method of a scalar elliptic model problem. Hybrid methods are also available for various other problems, including incompressible flow [15, 10, 9], convection-diffusion problems [8, 16], and the Helmholtz equations [14]. The framework proposed in this paper should allow an easy implemenation also for these problems.

The proposed implementation naturally supports nonconforming adaptive meshes, and allows the extension to a locally varying polynomial degree (*p*-adaption). These cases can also be covered by the presented analysis; for details, see e.g. [9].

One of the basic features of the hybrid discontinuous Galerkin method, is the elementwise assembly process, which provides the possibility for static condensation on the element level. We were not able to implement the assembling as a single loop over elements in the current version of DUNE. We intend however, to further investigate this issue. The static condensation on the element level, and the resulting assembly of the reduced global (Schur complement) system will probably require the implementation of a specialized grid operator space in DUNE-PDELAB.

Acknowledgements The first author would like to thank Steffen Muething (University of Stuttgart) for fruitful discussions, which lead to several improvements of the implementation, also Andreas

Dedner (Warwick Mathematics Institute) and the users of the DUNE mailing list, for helpful comments and hints. Both authors gratefully acknowledge financial support by the DFG through grant GSC 111 and from FWF through grant SFB/F032.

References

1. D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19:742–760, 1982.
2. D. N. Arnold and F. Brezzi. Mixed and nonconforming finite element methods: Implementation, postprocessing and error estimates. *Math. Model. Numer. Anal.*, 19:7–32, 1985.
3. C. Baiocchi, F. Brezzi, and L. D. Marini. *Stabilization of Galerkin Methods and Applications to Domain Decomposition*, pages 343–355. Lecture Notes in Computer Science. Springer, 1992.
4. P. Bastian, F. Heimann, and S. Marnach. Generic implementation of finite element methods in the Distributed and Unified Numerics Environment (DUNE). *Kybernetika*, 46(2):294–315, 2010.
5. B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of discontinuous Galerkin, mixed and continuous Galerkin methods for second order elliptic problems. *SIAM J. Numer. Anal*, 47(2):1319–1365, 2009.
6. H. Egger. A class of hybrid mortar finite element methods for interface problems with non-matching meshes. *preprint AICES-2009-2*, Jan. 2009.
7. H. Egger. A class of a-posteriori error estimators for hybrid discontinuous Galerkin methods. *Technical report, AICES, RWTH Aachen*, 2010.
8. H. Egger and J. Schöberl. A mixed-hybrid-discontinuous Galerkin finite element method for convection-diffusion problems. *IMA J. Numer. Anal.*, Jan. 2009.
9. H. Egger and C. Waluga. Hybrid discontinuous Galerkin methods for incompressible flow problems. part I: the Stokes problem. *in preparation*, Dec. 2010.
10. H. Egger and C. Waluga. A hybrid mortar finite element method for the Stokes problem. *preprint AICES-2010/04-4*, Apr. 2010.
11. E. Georgoulis. Inverse-type estimates on hp-finite element spaces and applications. *Mathematics of Computation*, Jan. 2008.
12. P. Houston, D. Schötzau, T. P. Wihler, and C. Schwab. Energy norm a posteriori error estimation of hp-adaptive discontinuous Galerkin methods for elliptic problems. *Mathematical Models and Methods in Applied Sciences*, 17(1):33–62, 2007.
13. K.Y. Kim. A posteriori error analysis for locally conservative mixed methods. *Mathematics of Computation*, 76(257):43, 2007.
14. P. Monk, J. Schöberl, and A. Sinwel. Hybridizing Raviart-Thomas elements for the Helmholtz equation. *Electromagnetics*, 30(1):149–176, 2010.
15. N.C. Nguyen, J. Peraire, and B. Cockburn. A hybridizable discontinuous Galerkin method for Stokes flow. *Computer Methods in Applied Mechanics and Engineering*, 199(9-12):582–597, Jan 2010.
16. N.C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations. *Journal of Computational Physics*, 228(9):3232–3254, 2009.
17. C. Waluga. DUNE-HDG: A framework for hybrid discontinuous Galerkin methods in DUNE. http://users.dune-project.org/projects/dunehdg, 2010.
18. T. Warburton and J. S. Hesthaven. On the constants in hp-finite element trace inverse inequalities. *Computer Methods in Applied Mechanics and Engineering*, 192(25):2765–2773, 2003.