dune-dpg: A Library for Discontinous Petrov Galerkin Using Optimal Test Spaces

Felix Gruber and Angela Klewinghaus

joint work with Olga Mula

IGPM, RWTH Aachen

Dune User Meeting 2015 in Heidelberg

igpm Institute for Geometry and Applied Mathematics | RWTH AACHEN UNIVERSITY

Motivation:

Variational formulation:

Find $u \in \mathcal{U}$ such that

$$b(u, v) = \ell(v) \qquad \text{for all } v \in \mathcal{V}$$

for Hilbert spaces $\mathcal{U}, \mathcal{V}$, continuous bilinear form $b : \mathcal{U} \times \mathcal{V} \to \mathbb{R}$ and $\ell \in \mathcal{V}'$.

Goal: Stability

igpm Institute for Geometry and Applied Mathematics | RWTH AACHEN UNIVERSITY

Motivation:

Variational formulation:

Find $u_h \in \mathcal{U}_h$ such that

$$b(u_h, v_h) = \ell(v_h) \qquad \text{for all } v_h \in \mathcal{V}_h$$

for Hilbert spaces $\mathcal{U}, \mathcal{V}$, continuous bilinear form $b : \mathcal{U} \times \mathcal{V} \to \mathbb{R}$ and $\ell \in \mathcal{V}'$.

Goal: Stability

- Discrete approximation yields a best approximation (up to a constant)

$$\|u - u_h\|_{\mathcal{U}} \lesssim \inf_{w_h \in \mathcal{U}_h} \|u - w_h\|_{\mathcal{U}} \,.$$

igpm Institute for Geometry and Applied Mathematics | RWTH AACHEN UNIVERSITY

Motivation:

Variational formulation:

Find $u_h \in \mathcal{U}_h$ such that

$$b(u_h, v_h) = \ell(v_h) \qquad \text{for all } v_h \in \mathcal{V}_h$$

for Hilbert spaces $\mathcal{U}, \mathcal{V}$, continuous bilinear form $b : \mathcal{U} \times \mathcal{V} \to \mathbb{R}$ and $\ell \in \mathcal{V}'$.

Goal: Stability

- Discrete approximation yields a best approximation (up to a constant)

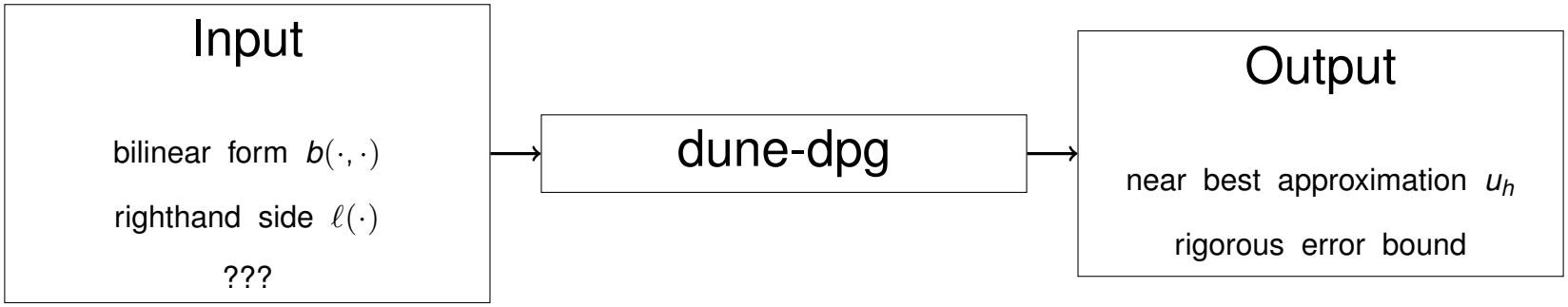
$$\|u - u_h\|_{\mathcal{U}} \lesssim \inf_{w_h \in \mathcal{U}_h} \|u - w_h\|_{\mathcal{U}} \ .$$

- Residual-based error bound

$$\|u - u_h\|_{\mathcal{U}} \simeq \|b(u_h, \cdot) - \ell\|_{\mathcal{V}'} \ .$$

igpm Institute for Geometry and Applied Mathematics | RWTH AACHEN UNIVERSITY

Ultimate Goal:



Choose $\mathcal{U}_h$ to Ensure Good Approximation Properties, Choose $\mathcal{V}_h$ to Ensure Stability

How to choose $\mathcal{V}_h$ for given $\mathcal{U}_h$?

Literature: Barret/Morton, Demkowicz/Gopalakrishnan, Dahmen/Cohen/Welper, Dahmen/Huang/Schwab/Welper, Broersen/Stevenson, Broersen/Dahmen/Stevenson

igpm Institute for Geometry and Applied Mathematics | RWTH AACHEN UNIVERSITY

Choose $\mathcal{U}_h$ to Ensure Good Approximation Properties, Choose $\mathcal{V}_h$ to Ensure Stability

How to choose $\mathcal{V}_h$ for given $\mathcal{U}_h$?

- Optimal test norm

$$\|\cdot\|_{\mathcal{V},opt} := \sup_{u\in\mathcal{U}} \frac{b(u,v)}{\|u\|_{\mathcal{U}}}$$

  - yields residual based error bound

$$\|u_h - u\|_{\mathcal{U}} = ||b(u_h,\cdot) - \ell||_{\mathcal{V}',opt}$$

Literature: Barret/Morton, Demkowicz/Gopalakrishnan, Dahmen/Cohen/Welper, Dahmen/Huang/Schwab/Welper, Broersen/Stevenson, Broersen/Dahmen/Stevenson

igpm Institute for Geometry and Applied Mathematics | RWTH AACHEN UNIVERSITY

Choose $\mathcal{U}_h$ to Ensure Good Approximation Properties, Choose $\mathcal{V}_h$ to Ensure Stability

How to choose $\mathcal{V}_h$ for given $\mathcal{U}_h$?

- Optimal test norm

$$\|\cdot\|_{\mathcal{V},opt} := \sup_{u\in\mathcal{U}} \frac{b(u,v)}{\|u\|_{\mathcal{U}}}$$

  - yields residual based error bound

$$\|u_h - u\|_{\mathcal{U}} = ||b(u_h,\cdot) - \ell||_{\mathcal{V}',opt}$$

- Optimal test space $\mathcal{V}_h = B^{-*}R_{\mathcal{U}}\mathcal{U}_h$, where $R_{\mathcal{U}} : \mathcal{U} \to \mathcal{U}'$ is the Rieszisomorphism, that is $\mathcal{V}_h := T(\mathcal{U}_h)$ where

$$\langle T(u_h), v\rangle_{\mathcal{V},opt} = b(u_h, v)\,, \qquad \forall\, v \in \mathcal{V}$$

  - yields best approximation property

$$\|u - u_h\|_{\mathcal{U}} = \inf_{w_h\in\mathcal{U}_h} \|u - w_h\|_{\mathcal{U}}$$

Literature: Barret/Morton, Demkowicz/Gopalakrishnan, Dahmen/Cohen/Welper, Dahmen/Huang/Schwab/Welper, Broersen/Stevenson, Broersen/Dahmen/Stevenson

Computation of the Optimal Test Space

We have to compute $\mathcal{V}_h$ by solving

$$\langle T(u_h), v \rangle_{\mathcal{V},opt} = b(u_h, v)\,, \qquad \forall\, v \in \mathcal{V}$$

Literature: Barret/Morton, Demkowicz/Gopalakrishnan, Dahmen/Cohen/Welper, Dahmen/Huang/Schwab/Welper, Broersen/Stevenson, Broersen/Dahmen/Stevenson

igpm Institute for Geometry and Applied Mathematics | RWTH AACHEN UNIVERSITY

Computation of the Optimal Test Space

We have to compute $\mathcal{V}_h$ by solving

$$\langle T(u_h), v \rangle_{\mathcal{V},opt} = b(u_h, v)\,, \qquad \forall\, v \in \mathcal{V}$$

- This is infinite dimensional

Literature: Barret/Morton, Demkowicz/Gopalakrishnan, Dahmen/Cohen/Welper, Dahmen/Huang/Schwab/Welper, Broersen/Stevenson, Broersen/Dahmen/Stevenson

igpm Institute for Geometry and Applied Mathematics | RWTH AACHEN UNIVERSITY

Computation of the Optimal Test Space

We have to compute $\mathcal{V}_h$ by solving

$$\langle T(u_h), v \rangle_{\mathcal{V},opt} = b(u_h, v)\,, \qquad \forall\, v \in \mathcal{V}$$

- This is infinite dimensional
  - Projection approach: Replace $\mathcal{V}$ by some good enough finite dimensional test search space $\hat{\mathcal{V}} \subset \mathcal{V}$

Literature: Barret/Morton, Demkowicz/Gopalakrishnan, Dahmen/Cohen/Welper, Dahmen/Huang/Schwab/Welper, Broersen/Stevenson, Broersen/Dahmen/Stevenson

Computation of the Optimal Test Space

We have to compute $\mathcal{V}_h$ by solving

$$\langle T(u_h), v \rangle_{\mathcal{V},opt} = b(u_h, v)\,, \qquad \forall\, v \in \mathcal{V}$$

- This is infinite dimensional
  - Projection approach: Replace $\mathcal{V}$ by some good enough finite dimensional test search space $\hat{\mathcal{V}} \subset \mathcal{V}$
- We cannot (afford to) compute the optimal test norm and solve global problems

Literature: Barret/Morton, Demkowicz/Gopalakrishnan, Dahmen/Cohen/Welper, Dahmen/Huang/Schwab/Welper, Broersen/Stevenson, Broersen/Dahmen/Stevenson

Computation of the Optimal Test Space

We have to compute $\mathcal{V}_h$ by solving

$$\langle T(u_h), v \rangle_{\mathcal{V},opt} = b(u_h, v)\,, \qquad \forall\, v \in \mathcal{V}$$

- This is infinite dimensional
  - Projection approach: Replace $\mathcal{V}$ by some good enough finite dimensional test search space $\hat{\mathcal{V}} \subset \mathcal{V}$
- We cannot (afford to) compute the optimal test norm and solve global problems
  - replace the optimal test norm by an equivalent but localizable and computable norm
  - use an ultraweak variational formulation
  - (alternatively: use a saddlepoint formulation)

Literature: Barret/Morton, Demkowicz/Gopalakrishnan, Dahmen/Cohen/Welper, Dahmen/Huang/Schwab/Welper, Broersen/Stevenson, Broersen/Dahmen/Stevenson

Computation of the Optimal Test Space

We have to compute $\mathcal{V}_h$ by solving

$$\langle T(u_h), v\rangle_{\mathcal{V},opt} = b(u_h, v)\,, \qquad \forall\, v \in \mathcal{V}$$

- This is infinite dimensional
  - Projection approach: Replace $\mathcal{V}$ by some good enough finite dimensional test search space $\hat{\mathcal{V}} \subset \mathcal{V}$
- We cannot (afford to) compute the optimal test norm and solve global problems
  - replace the optimal test norm by an equivalent but localizable and computable norm
  - use an ultraweak variational formulation
  - (alternatively: use a saddlepoint formulation)

We can compute an approximation for $\mathcal{V}_h$ by solving

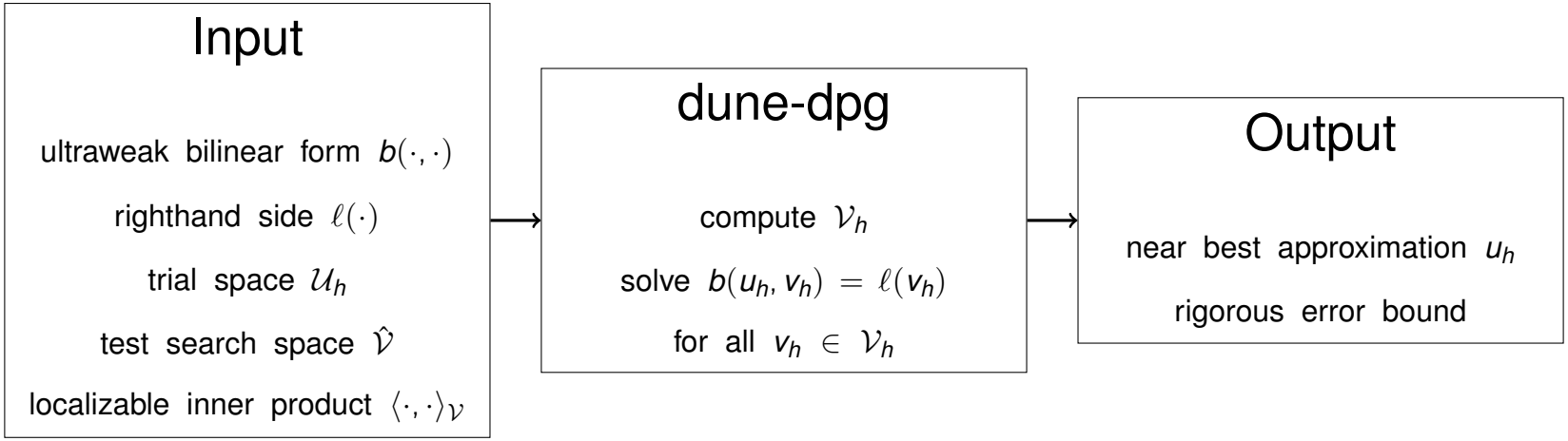
$$\langle \hat{T}(u_h), v\rangle_{\mathcal{V},K} = b_K(u_h, v)\,, \qquad \forall\, v \in \hat{\mathcal{V}}\big|_K\,,$$

where K is a single cell and $\langle\cdot,\cdot\rangle_{\mathcal{V},K}$ and $b_K(\cdot,\cdot)$ denote restrictions to K.

Literature: Barret/Morton, Demkowicz/Gopalakrishnan, Dahmen/Cohen/Welper, Dahmen/Huang/Schwab/Welper, Broersen/Stevenson, Broersen/Dahmen/Stevenson

igpm Institute for Geometry and Applied Mathematics | RWTH AACHEN UNIVERSITY

dune-dpg:



igpm Institute for Geometry and Applied Mathematics | RWTH AACHEN UNIVERSITY

Implementation: Layout of dune-dpg

rough overview of a DPG solver:

```
define grid
define trial space U_h and test search space V̂
define bilinear form and inner product

assemble system matrix and rhs vector
interpolate boundary values

solve system
compute a posteriori error estimates
[refine adaptively]
```

Implementation: BilinearForm / InnerProduct

- In our case, a bilinear form can be seen as a sum of integral terms, e. g.

$$b_K((u,q),v) := \int_K cvu - \int_K \beta \cdot \nabla v u + \int_{\partial K} uq\beta \cdot n.$$

Implementation: BilinearForm / InnerProduct

- In our case, a bilinear form can be seen as a sum of integral terms, e. g.

$$b_K((u,q),v) := \int_K cvu - \int_K \beta \cdot \nabla v u + \int_{\partial K} uq\beta \cdot n.$$

- The bilinear form is used to create a local system matrix

$$M_{ij}^K := b_K((u_j,q_j),v_i)$$

  which is a sum of matrices given by the integrals above.

Implementation: BilinearForm / InnerProduct

- In our case, a bilinear form can be seen as a sum of integral terms, e. g.

$$b_K((u,q),v) := \int_K cvu - \int_K \beta \cdot \nabla v u + \int_{\partial K} uq\beta \cdot n.$$

- The bilinear form is used to create a local system matrix

$$M_{ij}^K := b_K((u_j,q_j),v_i)$$

  which is a sum of matrices given by the integrals above.

- Thus we introduce an `IntegralTerm` class to generate the local matrix for a single integral. The `BilinearForm` is then parametrized over several `IntegralTerms` and sums up their local matrices.

Implementation: BilinearForm / InnerProduct

- In our case, a bilinear form can be seen as a sum of integral terms, e. g.

$$b_K((u,q),v) := \int_K cvu - \int_K \beta \cdot \nabla v u + \int_{\partial K} uq\beta \cdot n.$$

- The bilinear form is used to create a local system matrix

$$M_{ij}^K := b_K((u_j,q_j),v_i)$$

  which is a sum of matrices given by the integrals above.

- Thus we introduce an `IntegralTerm` class to generate the local matrix for a single integral. The `BilinearForm` is then parametrized over several `IntegralTerms` and sums up their local matrices.

- Abstractly, an inner product is simply a symmetric bilinear form over a single space. Thus, we can reuse the "sum over `IntegralTerms`" logic to define an `InnerProduct` class.

Implementation: Optimal Testspace

- Localfunctions: Linear combinations of localfunctions of the test search space
  - Parameter: Localfunctions Test Search Space, Coefficient Matrix
- Functions:
  - Parameter: Trial Spaces, Test Search Spaces, Bilinear Form, Inner Product
  - uses IndexSet of Trial Spaces
  - in LocalView.bind(): Uses Trial Spaces, Test Search Spaces, Bilinear Form, Inner Product to set up and solve

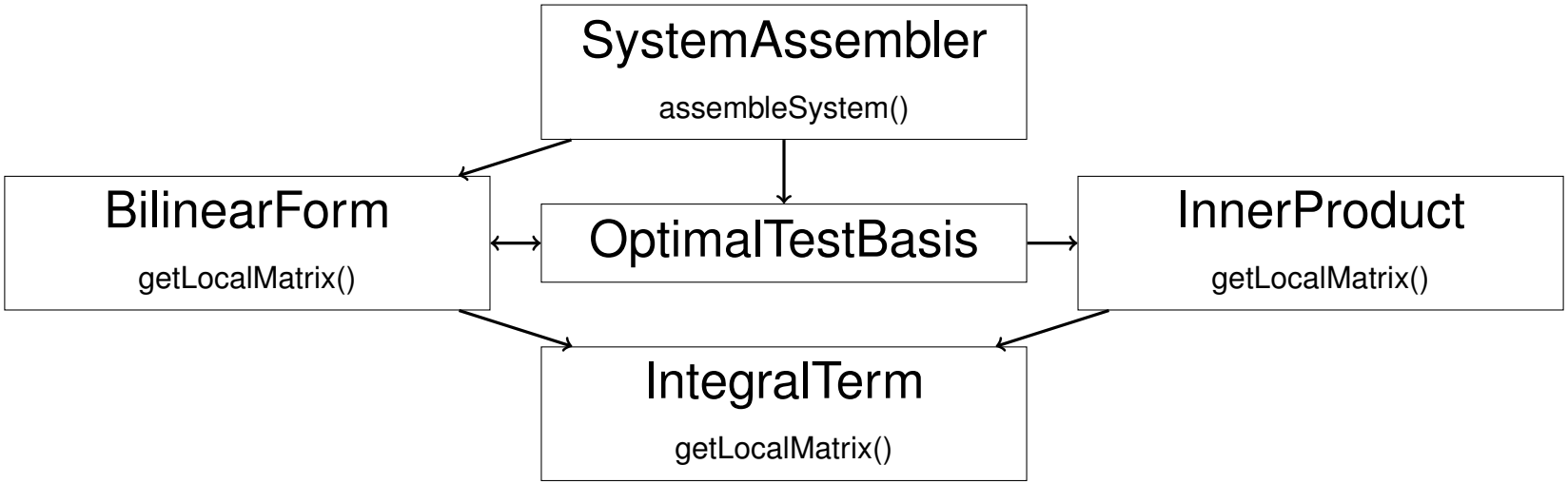
$$\langle \hat{T}(u_h), v \rangle_{\mathcal{V},K} = b_K(u_h, v)\,, \qquad \forall\, v \in \hat{\mathcal{V}}\big|_K\,,$$

  for every (local) solution basis function u_h on the current (physical) cell.
  $\rightarrow$ yields coefficient matrix for Localfunctions

Implementation: SystemAssembler

- Building a global system matrix and right hand side for our method is handled by the `SystemAssembler` class.
- The `SystemAssembler` loops over all cells and uses the `BilinearForm` and `OptimalTestBasis` to build local matrices.
- The local matrices built by the `BilinearForm` are then copied into a global system matrix. (Thanks to dune-functions we have a mapping between local and global indices.)
- Similarily the right hand side of the system is assembled cell by cell.
- `SystemAssembler` also later applies the boundary values to the system.

Implementation: Overview of dune-dpg



Example: Transport

Transport equation:

$$\begin{aligned} \beta \cdot \nabla u + cu &= f \quad \text{on } \Omega, \\ u &= g \quad \text{on } \Gamma_-. \end{aligned}$$

Challenge:

- not "second order elliptic" but hyperbolic

Theoreticel foundations:

- D. Broersen, W. Dahmen, R. Stevenson

Applications:

- discrete ordinates discretization of Boltzman equation (F. Gruber, joint work with W. Dahmen, O. Mula)
- limit case of the convection-diffusion equation (A. Klewinghaus, joint work with W. Dahmen)
- porous media (W. Dahmen, V. König, S. Müller)

We'd like to thank Oliver Sander for his introduction into dune(-functions).

Results Transport

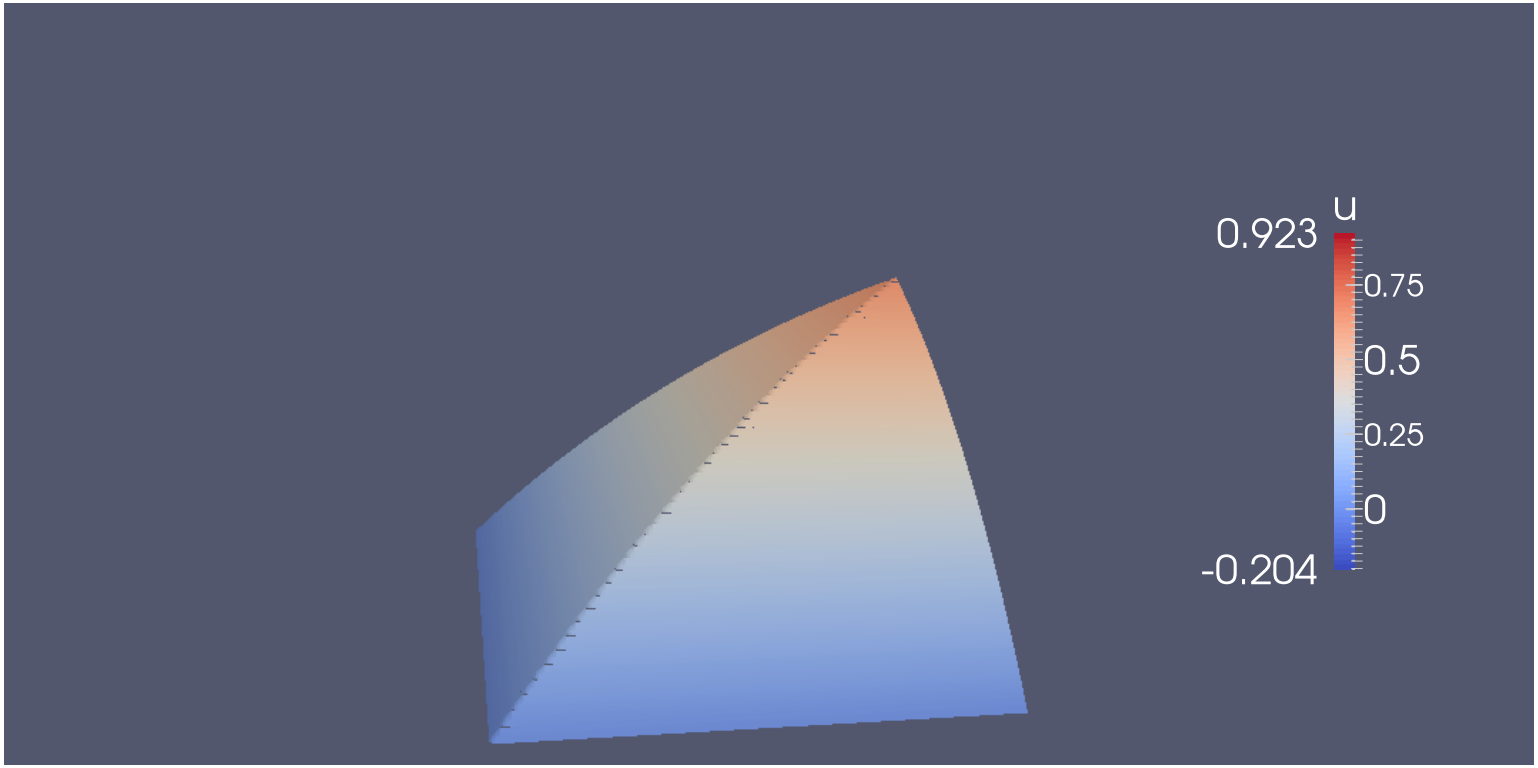


Figure: $\beta = (1, 1)$, $c = 1$, $f = 1$

Results Transport

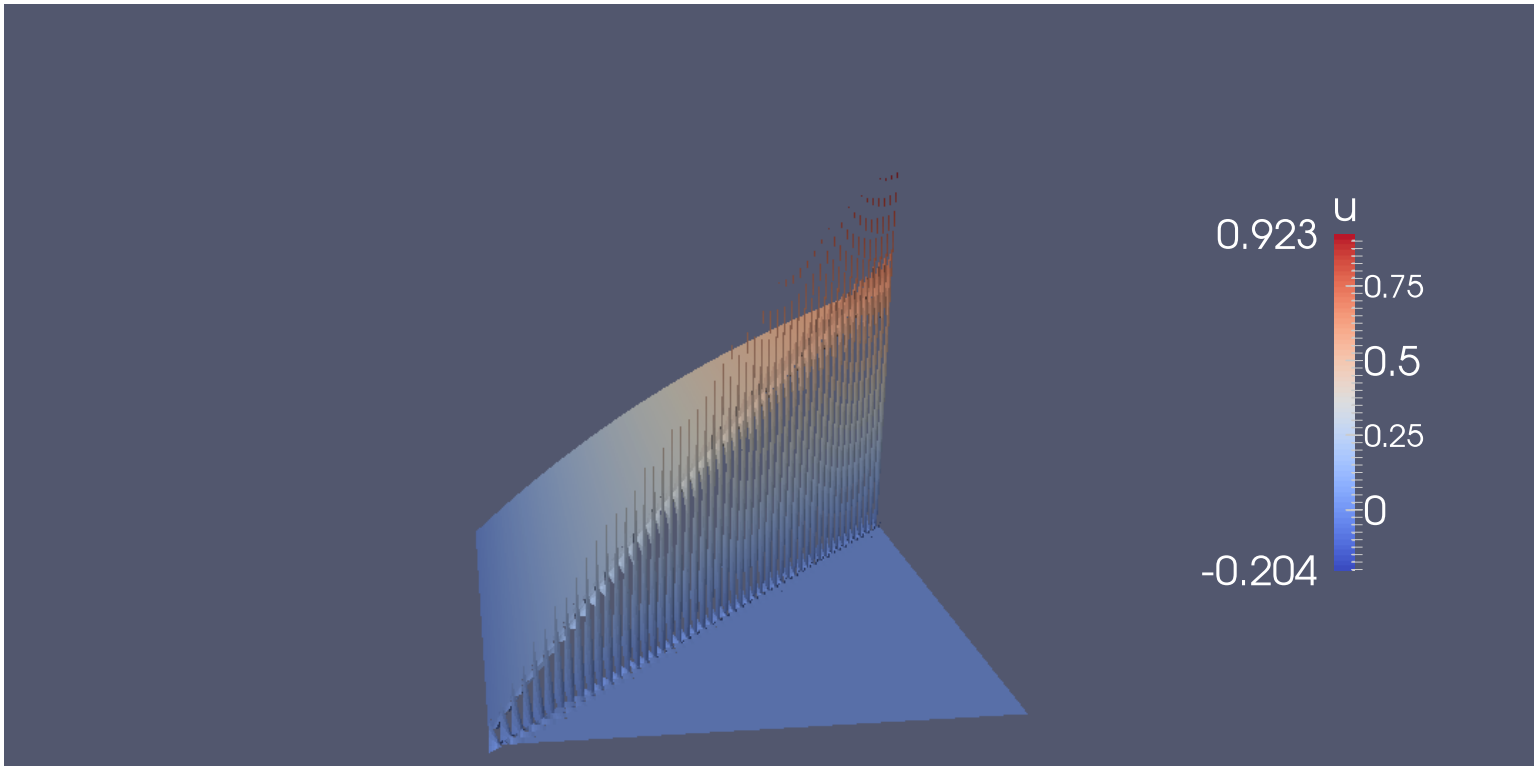
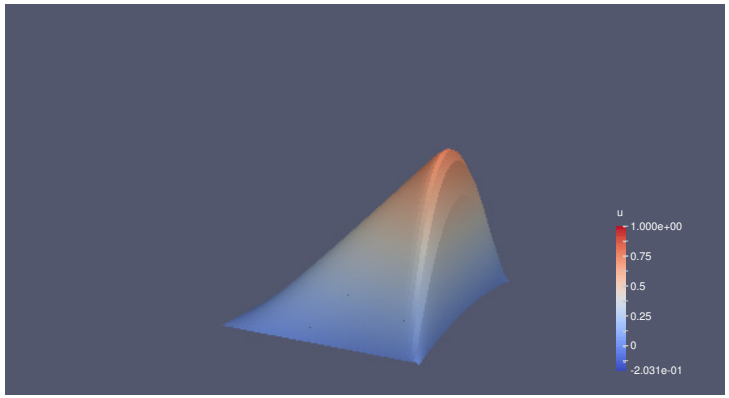
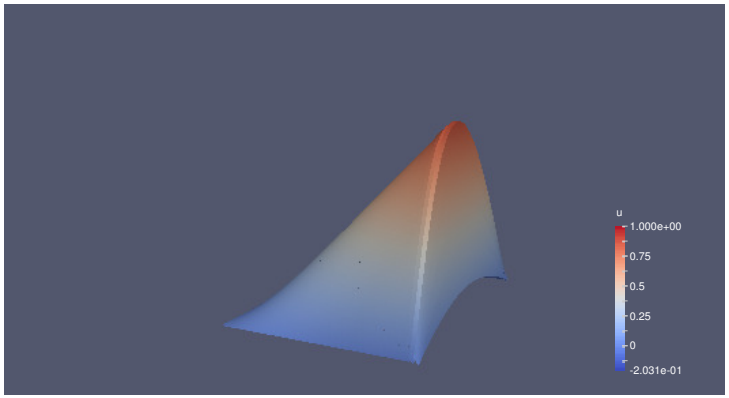


Figure: $\beta = (1, 1)$, $c = 1$, $f = 1$, if $x_1 < x_2$, or $f = 0$ else

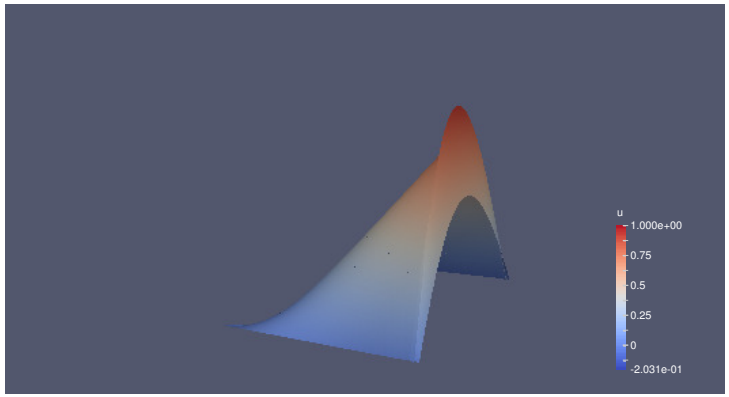
Results Convection Diffusion



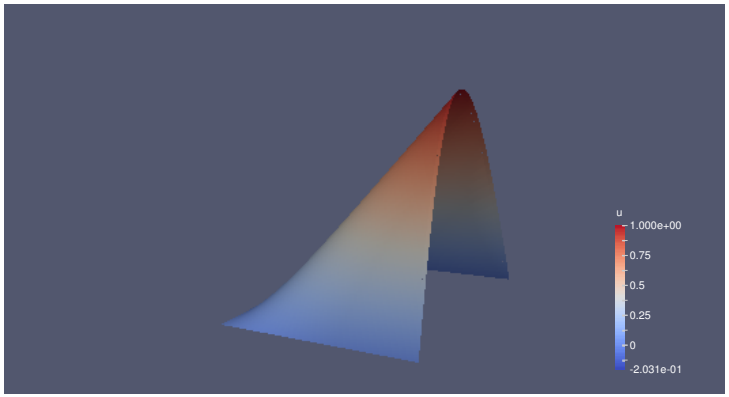
(a) $\epsilon = 0.02$



(b) $\epsilon = 0.01$



(c) $\epsilon = 0.002$



(d) $\epsilon = 0.00001$

Figure: $\Omega = (0,1)^2$, regular square grid with 50×50 elements ($h = 0.02$)