FEniCS on a Moebius strip

Solving PDEs over manifolds with FEniCS

Marie E. Rognes

Center for Biomedical Computing
Simula Research Laboratory

FEniCS '13, March 18, 2013

Thanks to Colin J. Cotter, David A. Ham and Andrew McRae!

An old friend abroad

Poisson's equation with homogenous Dirichlet bcs

Find u such that

$$\begin{aligned} -\Delta u &= 1 \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

Finite element variational form

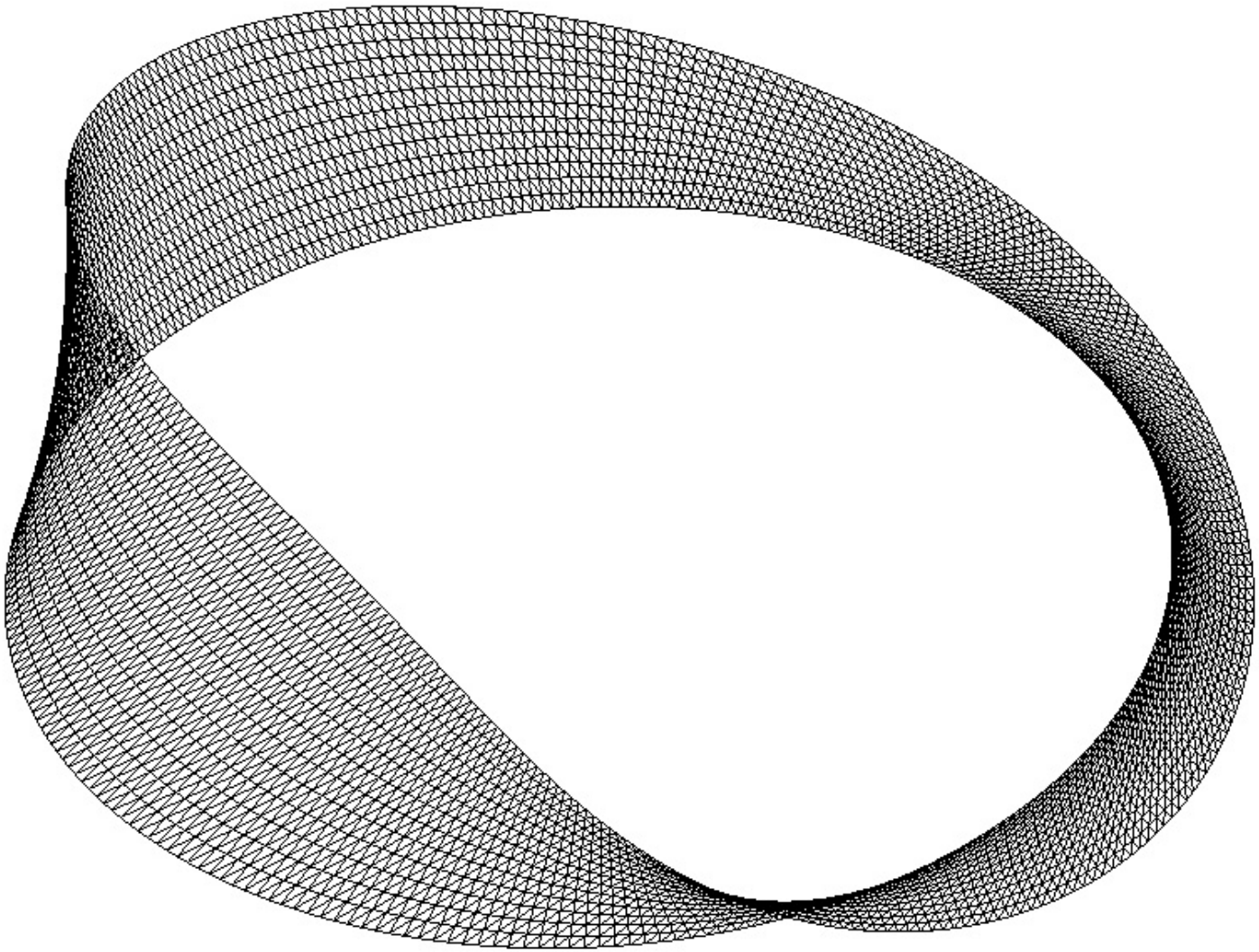
Find $u_h \in V_h(\mathcal{T}_h)$ such that

$$\langle \nabla u_h, \nabla v \rangle_\Omega = \langle 1, v \rangle_\Omega$$

for all $v \in V_h(\mathcal{T}_h)$.

We are interested in the case where Ω is embedded in $\mathbb{R}^n$ but has topological dimension m with $1 \leq m \leq n \leq 3$.

Given a mesh of your favorite manifold

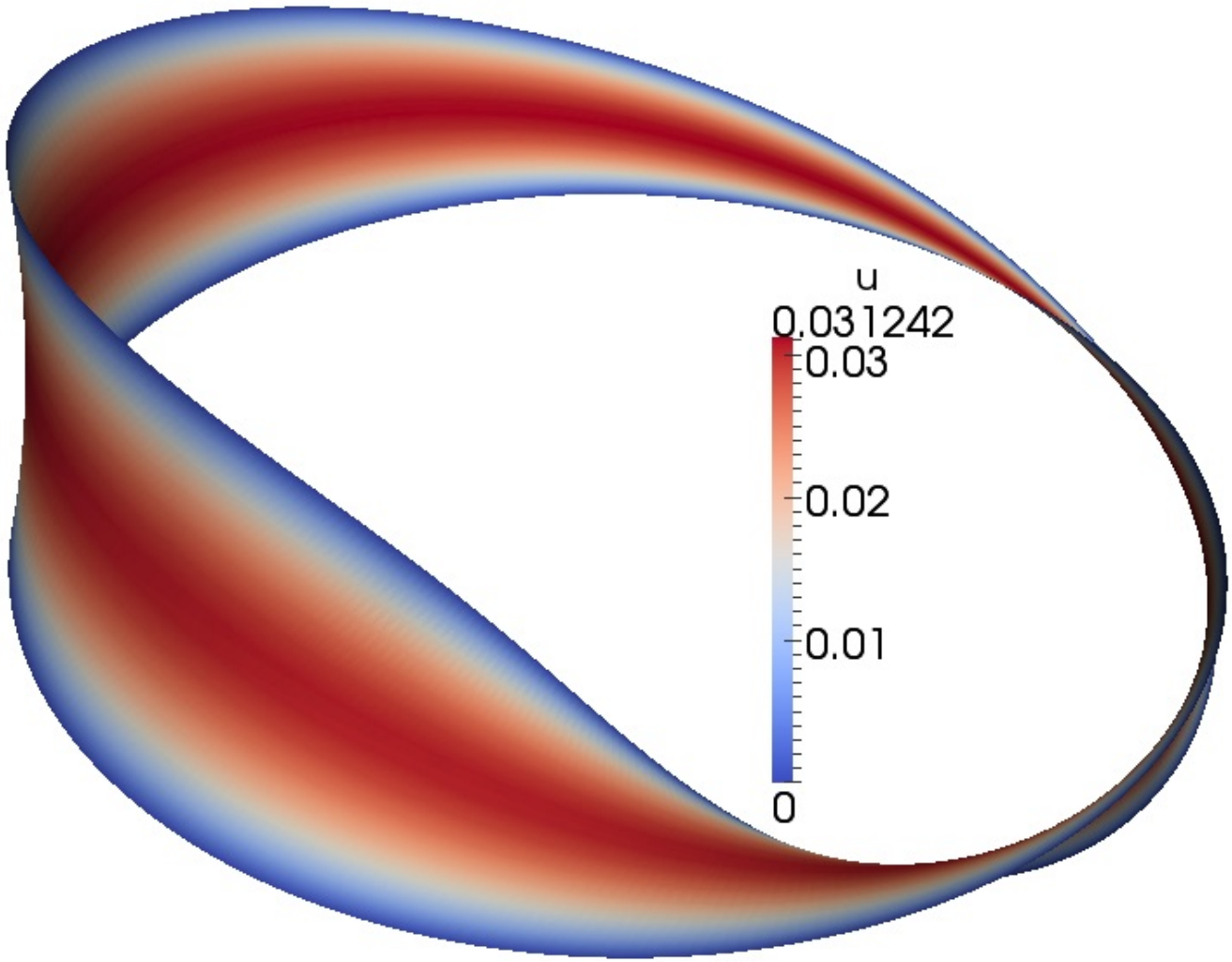


[Moebius strip mesh generated by script provided by Harish Narayanan, 2009]

The solver code is identical to the case where the geometrical equals the topological dimension

```
from dolfin import *

# Input mesh
mesh = Mesh("Moebius2.xml.gz")

# Define and solve problem as usual on this mesh
V = FunctionSpace(mesh, "Lagrange", 1)
u = TrialFunction(V)
v = TestFunction(V)

a = inner(grad(u), grad(v))*dx
f = Constant(1.0)
L = f*v*dx

u = Function(V)
bc = DirichletBC(V, 0.0, "on_boundary")
solve(a == L, u, bc)
```

The resulting solution seems plausible



[Available in DOLFIN trunk (bzr branch lp:dolfin) and in FEniCS 1.2]

Interpreting UFL over manifolds

Finite elements can be defined on simplicial cells with differing geometric and topological dimension

```
# Define triangle cell embedded in R^3
cell = Cell("triangle", 3)
cell.geometric_dimension() == 3 # True
cell.topological_dimension() == 2 # True

# Define elements as usual
Q = FiniteElement("Lagrange", cell, 1)
RT = FiniteElement("RT", cell, 1)

# Define Lagrange vector element
# Value dimension default to geometric dimension
V = VectorElement("Lagrange", cell, 1)

# Arguments defined over V will have 3 components:
u = Coefficient(V)
u[0], u[1], u[2]
```

The UFL gradient is defined via the natural definition of the directional derivative

The differential operators `dx`, `Dx`, `div`, `rot`, `curl` follows.

```
cell = Cell("triangle", 3)
V = FiniteElement("Lagrange", cell, 1)
u = Coefficient(V)
u.dx(2) # What does this mean?
```

Interpret

$$\texttt{grad(u)} := \nabla u$$

Define

```
u.dx(i) = grad(u)[i]
Dx(u, i) = grad(u)[i]
```

Define $\nabla u(x) \in \mathbb{R}^n$ via

$$\nabla u(x) \cdot v = \lim_{\epsilon \to 0} \frac{u(x + \epsilon v) - u(x)}{\epsilon}$$

for v in the tangent space.

Measures are defined with reference to the topological dimension of the mesh

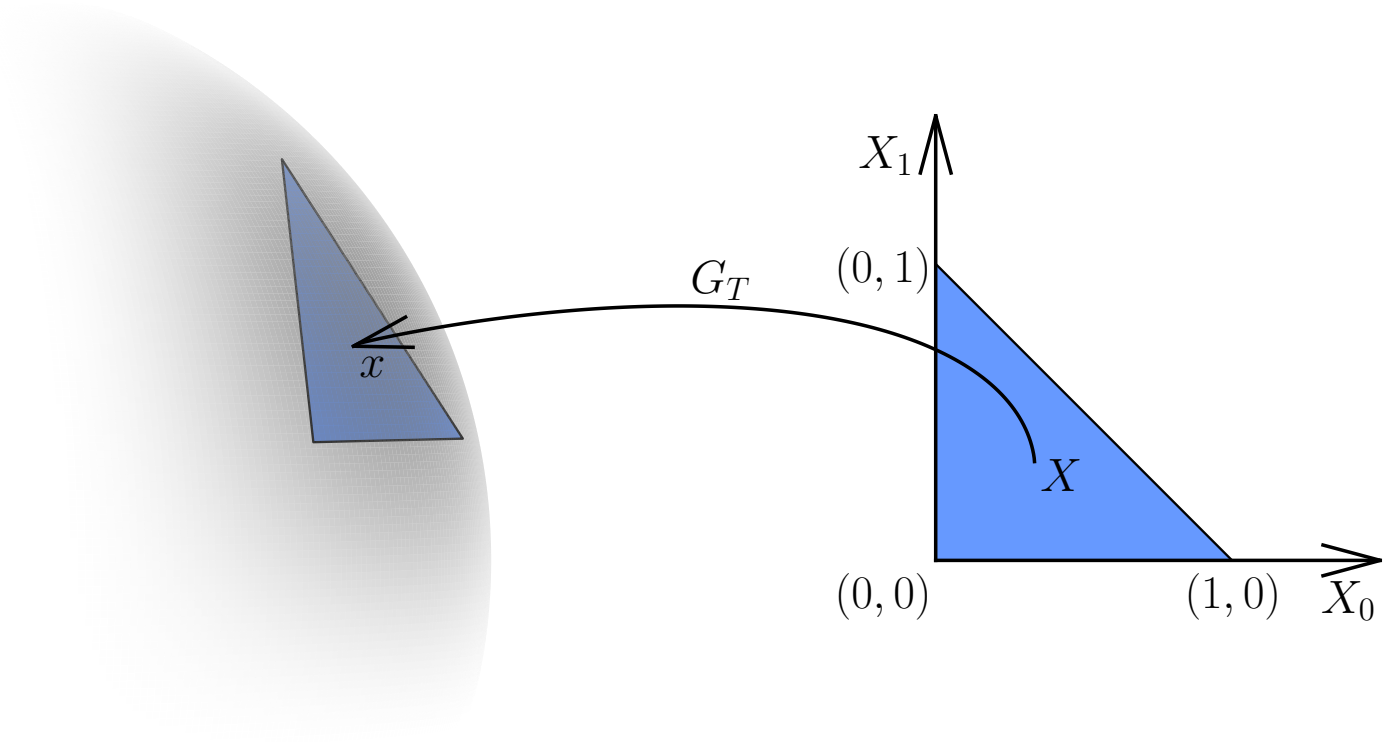
$$\texttt{I*dx} := \sum_{T \in \mathcal{T}} \int_T I \, \mathrm{d}x, \quad \texttt{I*ds} := \sum_{e \in \mathcal{E}^e} \int_e I \, \mathrm{d}s, \quad \texttt{I*dS} := \sum_{e \in \mathcal{E}^i} \int_e I \, \mathrm{d}s.$$

For a mesh of geometric dimension n and topological dimension m, $\mathrm{d}x$ and $\mathrm{d}s$ refer to the standard integration measures on R^m and R^{m-1} respectively.

```
# Integrate 1 over the exterior facets of the mesh
I = Constant(1.0)
a = I*ds
A = assemble(a, mesh=mesh)

# Integrate 1 over the cells of the surface mesh
surface = BoundaryMesh(mesh, "exterior")
b = I*dx
B = assemble(b, mesh=surface)
```

Compiling forms over manifolds

Form code generation in FFC is based on pulling the form back to a reference element



Define element transformation $G_T : T_0 \to T$ and its rectangular Jacobian J_T

$$x = G_T(X), \quad J_T(X) = \frac{\partial G_T(X)}{\partial X} = \frac{\partial x}{\partial X}$$

The Gram determinant is the appropriate generalized determinant of rectangular Jacobians

Map for scalar fields (and affine vector fields):

$$\phi(x) = \Phi(X)$$

The transform of the mass matrix follows:

$$\int_T \phi(x)\,\psi(x)\,\mathrm{d}x = \int_{T_0} \Phi(X)\,\Psi(X)\;|J|\;\mathrm{d}X,$$

using the Gram determinant

$$|J| = \det(J^T J)^{1/2}.$$

The Moore-Penrose pseudoinverse is the appropriate pseudoinverse of rectangular Jacobians

The natural transform for gradients:

$$\nabla_x \phi(x) = (J^\dagger)^T \nabla_X \Phi(X)$$

uses the Moore-Penrose pseudo-inverse:

$$J^\dagger = \left(J^T J\right)^{-1} J^T.$$

The transform of the stiffness matrix follows

$$\int_T \nabla \phi \cdot \nabla \psi \, \mathrm{d}x = \int_{T_0} \left((J^\dagger)^T \nabla \Phi\right) \cdot \left((J^\dagger)^T \nabla \Psi\right) |J| \, \mathrm{d}X.$$

$H(\mathrm{curl})$ and $H(\mathrm{div})$ elements map as usual with the generalized geometry definitions

But cell orientation can not be determined locally

Map $H(\mathrm{curl})$ functions via the covariant Piola:

$$\phi(x) = (J^\dagger)^T \, \Phi(X).$$

Map $H(\mathrm{div})$ functions via the contravariant Piola:

$$\phi(x) = \pm |J|^{-1} J \, \Phi(X).$$

In flat space, the sign of the Jacobian determinant ensures that contravariant Piola elements are mapped correctly. In contrast, the Gram determinant carries no sign. Sign is therefore (and must be) determined by combining local and global information.

Ocean modelling on the sphere

[Numerical experiments thanks to Andrew McRae/Colin J. Cotter]

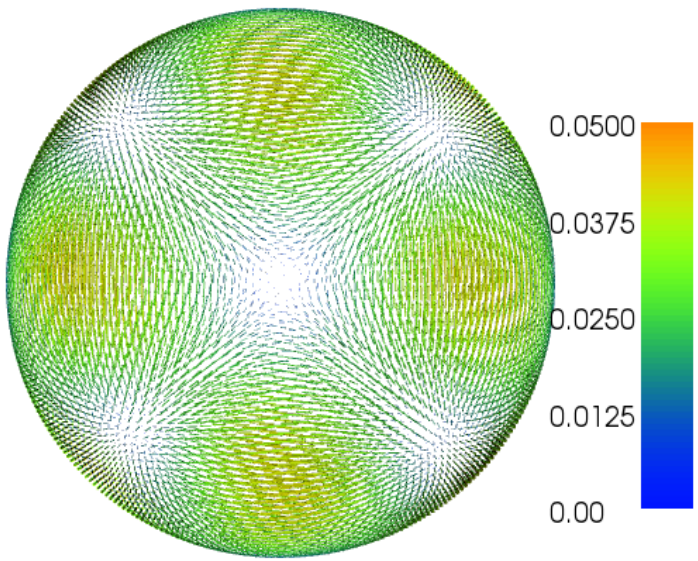
Mixed Poisson on the sphere: $H(\mathrm{div}) \times L^2$

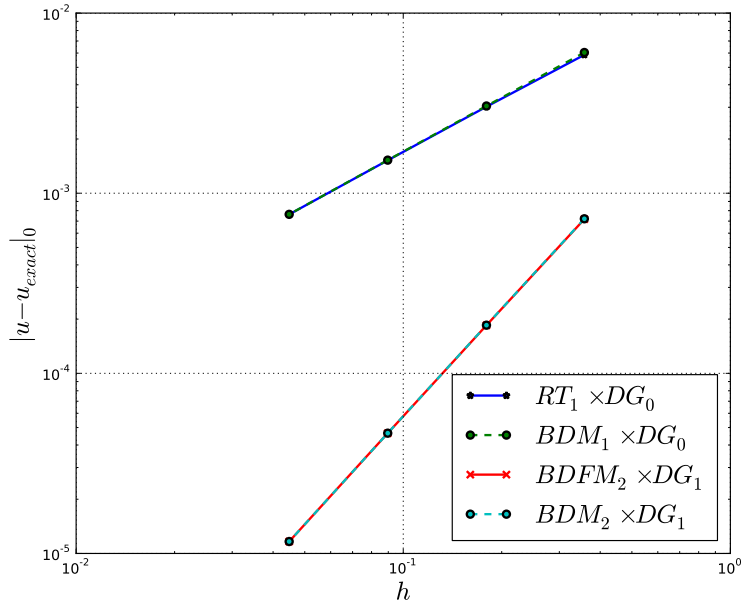
How to provide global orientation information

Find $(\sigma, u, r) \in \Sigma \times V \times R = W \subset H(\mathrm{div}) \times L^2 \times \mathbb{R}$ such that

$$\langle \sigma, \tau \rangle + \langle \mathrm{div}\, \sigma, v \rangle + \langle \mathrm{div}\, \tau, u \rangle + \langle r, v \rangle + \langle t, u \rangle = \langle g, v \rangle$$

for all $(\tau, v, t) \in W$.






```
mesh = Mesh("sphere.xml.gz")
global_normal = Expression(("x[0]", "x[1]", "x[2]"))
mesh.init_cell_orientations(global_normal)
```

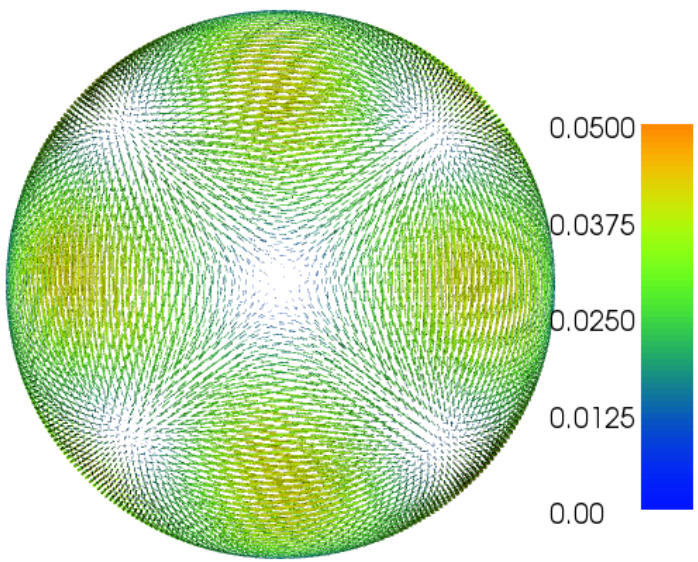
Mixed Poisson on the sphere: $L^2 \times H^1$

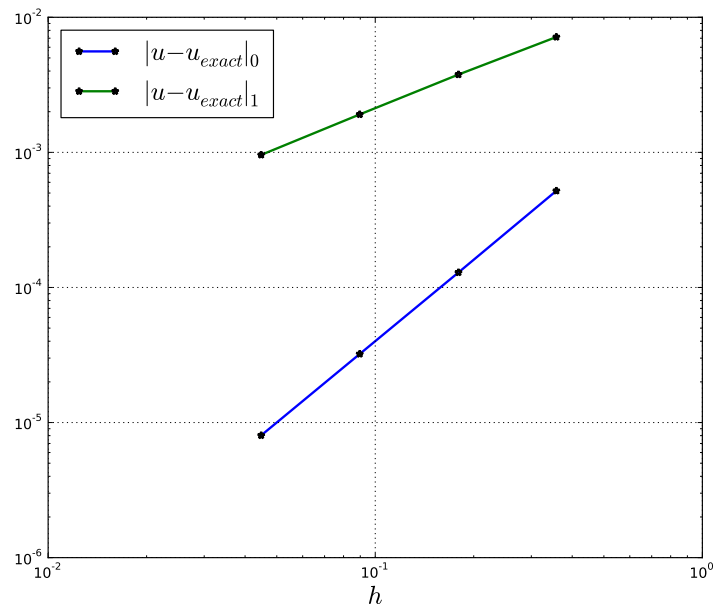
How to force a vector-field into the tangent space via a Lagrange multiplier

Find $(\sigma, u, l, r) \in W \subset L^2 \times H^1 \times L^2 \times \mathbb{R}$ such that

$$\langle \sigma, \tau \rangle + \langle \tau, \nabla u \rangle - \langle \sigma, \nabla v \rangle - \langle l, \tau \cdot n \rangle + \langle k, \sigma \cdot n \rangle + \langle r, v \rangle + \langle t, u \rangle = -\langle g, v \rangle$$

for all $(\tau, v, k, t) \in W$. Take $W = DG_1^3 \times CG_2 \times DG_1 \times \mathbb{R}$.

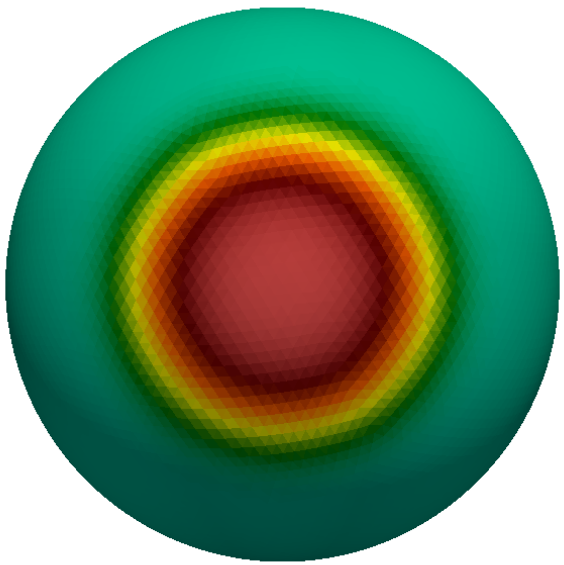


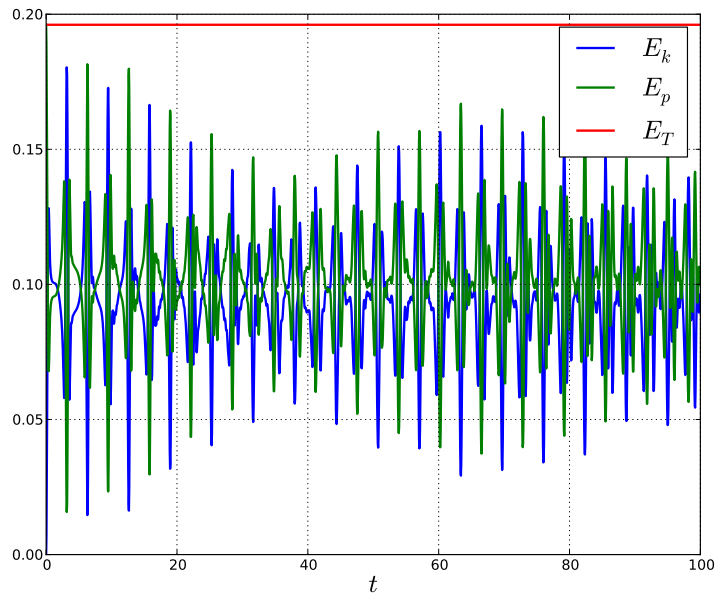


Shallow water equations over the surface of the earth

Find the velocity u and the depth perturbation D, given gravity g and base layer depth H

$$u_t + f u^{\perp} + g \nabla D = 0$$
$$D_t + H \operatorname{div}(u) = 0$$





Current status

Anything that works for meshes with geometric and topological dimension n for $n = 1, 2, 3$, should also work for the case of topological dimension m and geometric dimension n for $1 \leq m \leq n \leq 3$.

Limitations

- No mixed spaces on cells of differing dimensions (still): cannot combine spaces on facets with spaces on cells for instance.
- No curved elements (still): linear tessellations only