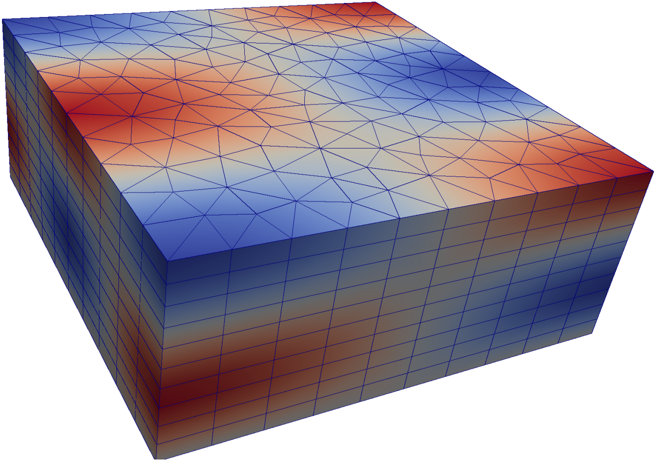
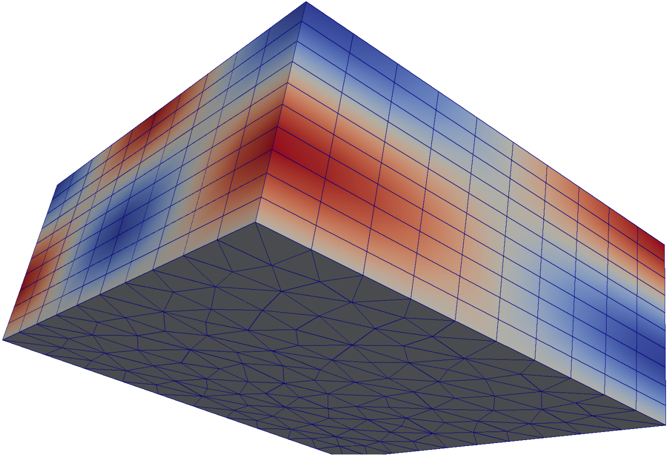


Solution to a Helmholtz equation in $[0,1] \times [0,1] \times [0,0.4]$:

$$u = \cos(\pi x)\cos(2\pi y)\sin\left(1.5\pi\frac{z}{0.4}\right)$$



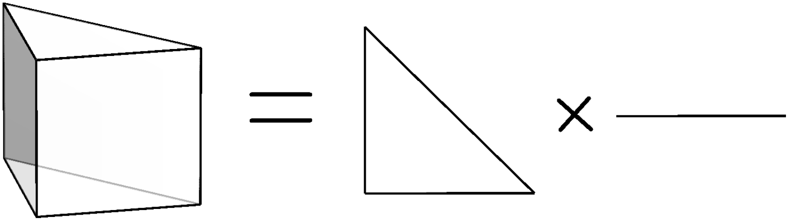
Solution to a Helmholtz equation in $[0,1] \times [0,1] \times [0,0.4]$:

$$u = \cos(\pi x)\cos(2\pi y)\sin\left(1.5\pi \frac{z}{0.4}\right)$$

Part II:
Function spaces on extruded meshes

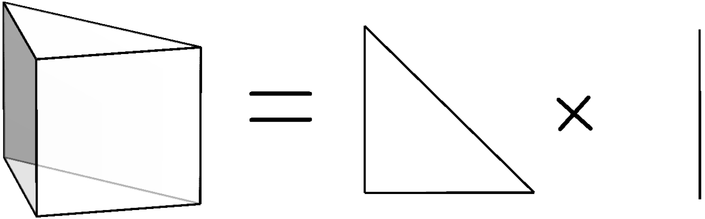
Function spaces

Geometrically:



Function spaces

Geometrically:



$$(x, y, z) \in \text{prism} \iff (x, y) \in \text{triangle},\ \ z \in \text{interval}$$

Outer-product elements

Recall: within a cell, we have local basis functions $\Phi_i(\vec{X})$.

Suppose we have $\{\Phi_i^{2\mathrm{D}}(X, Y)\}$ and $\{\Phi_j^{1\mathrm{D}}(Z)\}$. Then there is a natural way to generate basis functions on the extruded cell:

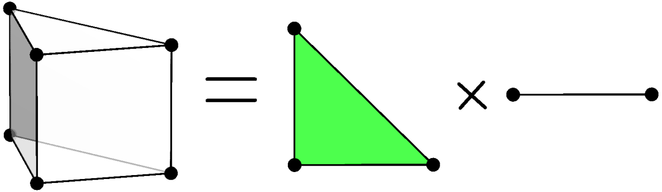
$$\Phi_{i,j}^{\mathrm{extr}}(X, Y, Z) := \Phi_i^{2\mathrm{D}}(X, Y) \times \Phi_j^{1\mathrm{D}}(Z)$$

Bonus: if $\{\Phi_i^{2\mathrm{D}}\}$ and $\{\Phi_j^{1\mathrm{D}}\}$ are *nodal*, then so is $\{\Phi_{i,j}^{\mathrm{extr}}\}$.

Worked example:

CG1 on triangles: (nodal) basis functions are $\{X,\ Y,\ 1 - X - Y\}$.
CG1 on intervals: (nodal) basis functions are $\{Z,\ 1 - Z\}$.

$\implies$ CG1 $\times$ CG1 on prisms: (nodal) basis functions are $\{XZ, X(1-Z), YZ, Y(1-Z), (1-X-Y)Z, (1-X-Y)(1-Z)\}$.



Firedrake/UFL syntax:

```
mesh = ExtrudedMesh(...)

V_horiz = FiniteElement("CG", triangle, 1)
V_vert = FiniteElement("CG", interval, 1)
V_elt = OuterProductElement(V_horiz, V_vert)

V = FunctionSpace(mesh, V_elt)

# in this case, we could have used the more compact
# FunctionSpace(mesh, "CG", 1, vfamily="CG", vdegree=1)
```

Vector-valued spaces

Note: RT/BDM/Nédélec-type elements, NOT VectorFunctionSpace

Problems:

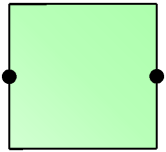
- Dimension mismatch. If Φ_i^{2D} takes values in $\mathbb{R}^2$ (or $\mathbb{R}$) and Φ_j^{1D} is scalar-valued, their product is in $\mathbb{R}^2$ (or $\mathbb{R}$), never $\mathbb{R}^3$.
- The result effectively needs to be “padded” with zeros to produce a function taking values in $\mathbb{R}^3$.
- In some situations, there is no single “correct” padding...

Vector-valued spaces

Trouble in extruded 1D:

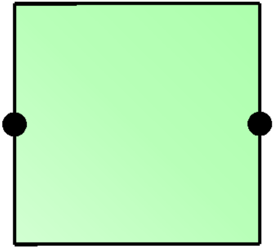
```
V_horiz = FiniteElement("CG", interval, 1)
V_vert = FiniteElement("DG", interval, 0)
V_elt = OuterProductElement(V_horiz, V_vert)
```

- Element is scalar-valued
- Basis functions X and $1 - X$ (indpt. of Y)
- Horizontally continuous
- Vertically discontinuous



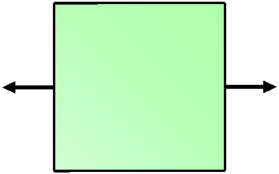
Vector-valued spaces

How to make this vector-valued?

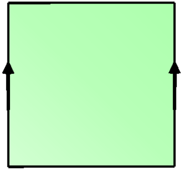


Vector-valued spaces

How to make this vector-valued? Two possibilities:



- Basis functions $(X, 0)$ and $(1-X, 0)$
- Continuous *normal* component
- Discontinuous *tangential* component

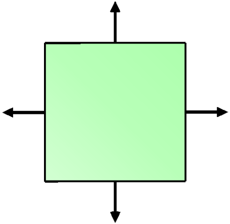


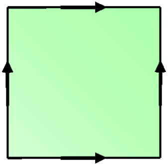
- Basis functions $(0, X)$ and $(0, 1-X)$
- Continuous *tangential* component
- Discontinuous *normal* component

Vector-valued spaces

Solution:

- Additional operators `HDiv()` and `HCurl()` to "expand" an `OuterProductElement` into a vector-valued element of the correct dimension, with the desired continuity.

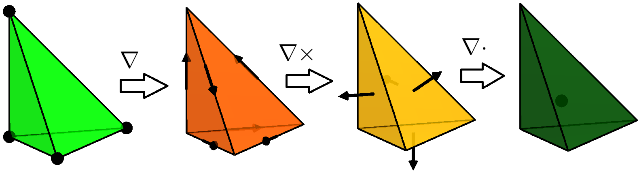




Motivation: de Rham complexes

- Sequence of function spaces, linked by differential operators and more.
- Useful in *mixed* problems (stability properties).

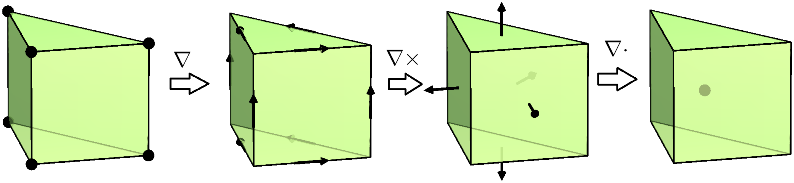
E.g. lowest-order $\mathcal{P}^-$ complex on tetrahedra:



$$\mathtt{CG}_1 \xrightarrow{\nabla} \mathtt{N1curl}_1 \xrightarrow{\nabla\times} \mathtt{RT}_1 \xrightarrow{\nabla\cdot} \mathtt{DG}_0$$

Motivation: de Rham complexes

On prisms, the equivalent is



For this, we need *product complexes*.

Product complexes

For any 2 complexes $(U_0, \ldots, U_m)$ and $(V_0, \ldots, V_n)$, we can build a *product complex*[*] $(W_0, \ldots, W_{m+n})$, where

$$W_i = \bigoplus_{j+k=i} U_j \otimes V_k$$

- The U_i and V_i are just `FiniteElement(...)`
- $\otimes$ becomes `OuterProductElement`
- $\oplus$ becomes `EnrichedElement`, or just '$+$'.

[*]Douglas N. Arnold, Daniele Boffi and Francesca Bonizzoni: *Finite element differential forms on curvilinear cubic meshes and their approximation properties.*

Product complexes example

Here, we will use

- the 2D complex (CG_1, RT_1, DG_0), and
- the 1D complex (CG_1, DG_0),

The product complex is then

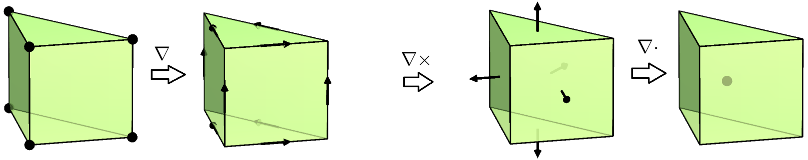
$$
\begin{aligned}
W_0 &= CG_1 \otimes CG_1 \\
W_1 &= (RT_1 \otimes CG_1) \oplus (CG_1 \otimes DG_0) \\
W_2 &= (DG_0 \otimes CG_1) \oplus (RT_1 \otimes DG_0) \\
W_3 &= DG_0 \otimes DG_0
\end{aligned}
$$

Function spaces

```
U0 = FiniteElement("CG", triangle, 1)
U1 = FiniteElement("RT", triangle, 1)
U2 = FiniteElement("DG", triangle, 0)

V0 = FiniteElement("CG", interval, 1)
V1 = FiniteElement("DG", interval, 0)

W0 = OuterProductElement(U0, V0)
W1 = HCurl(OPE(U1, V0)) + HCurl(OPE(U0, V1))
W2 = HDiv(OPE(U2, V0)) + HDiv(OPE(U1, V1))
W3 = OuterProductElement(U2, V1)

W0fs = FunctionSpace(mesh, W0)
...
```

Function spaces

```
W0 = OuterProductElement(U0, V0)
W1 = HCurl(OPE(U1, V0)) + HCurl(OPE(U0, V1))
W2 = HDiv(OPE(U2, V0)) + HDiv(OPE(U1, V1))
W3 = OuterProductElement(U2, V1)
```



$CG_1 \otimes CG_1$ $\quad (RT_1 \otimes CG_1) \oplus (CG_1 \otimes DG_0)$ $\quad (DG_0 \otimes CG_1) \oplus (RT_1 \otimes DG_0)$ $\quad DG_0 \otimes DG_0$

Function spaces

Bonus:

Vincent Dumoulin (Imperial MSc student) + David Ham + Lawrence Mitchell

$\Longrightarrow$ UFL-like language, based on FEEC instead of vector calculus (in-built support for complexes and product complexes)

Limitations

- Should be taking advantage of product structure when doing quadrature; currently doing more operations than necessary.
- No FIAT-level support for run-time function evaluation within an element (not supported in Firedrake anyway).
- Shortcuts taken in Jacobian calculation $\implies$ Jacobian not exact on *radially* extruded meshes (to be fixed when we implement non-affine support).

Conclusions

- Firedrake has support for extruded meshes, which are appropriate in geophysical and other contexts.
- For a modest number of layers, performance approaches that of a structured mesh, since we can visit a whole column for each unstructured cell access.
- Appropriate function spaces can be generated by manipulating existing UFL `FiniteElement` objects, using `OuterProductElement` and `HDiv`/`HCurl`.

http://www.firedrakeproject.org

New arrivals at *the zoo*