

Structure-preserving finite element methods

Artur Palha¹

¹Delft Institute of Applied Mathematics, Delft University of Technology

@CWI NUM-SCARS 2026

ERCOFTAC SIG 55

Overview

- Introduction and motivation
- Structure-preserving FEM
- Applications
- Mantis.jl

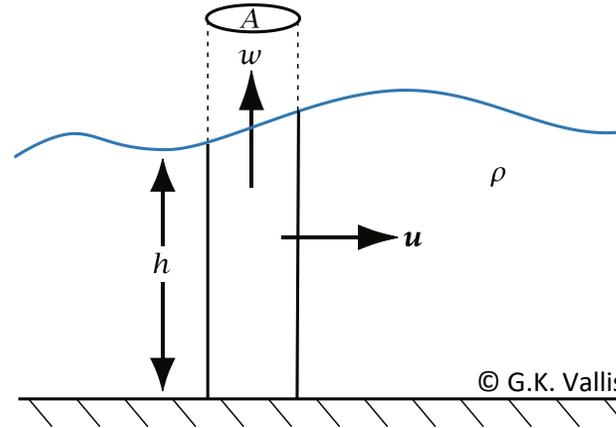
Introduction and motivation

Arakawa and Lamb grids

The equations and possible discretizations

$$\frac{\partial h}{\partial t} + \nabla \cdot (h\mathbf{u}) = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla (hg) = 0$$

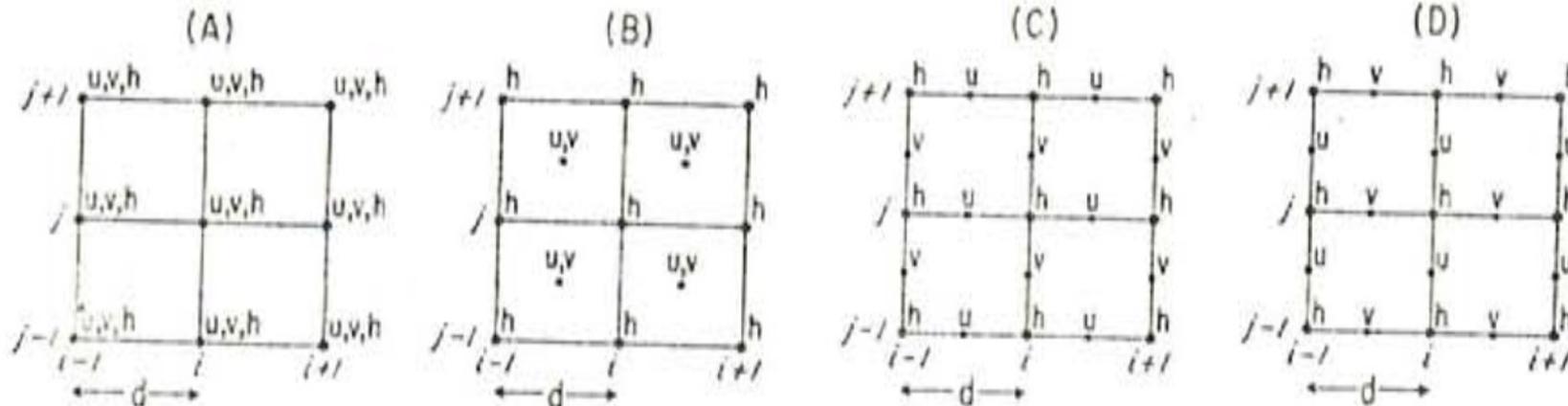


Methods in Computational Physics
Vol. 17, 1977

Computational Design of the Basic
Dynamical Processes of the UCLA
General Circulation Model

AKIO ARAKAWA AND VIVIAN R. LAMB
DEPARTMENT OF ATMOSPHERIC SCIENCES
UNIVERSITY OF CALIFORNIA
LOS ANGELES, CALIFORNIA

Studied the properties of the famous Arakawa grids



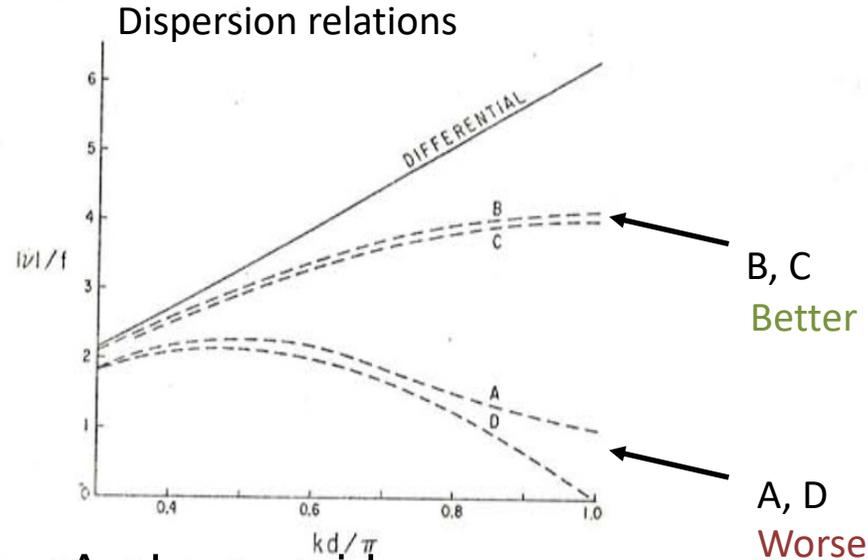
All finite differences but unknowns placed at different locations

Arakawa and Lamb grids

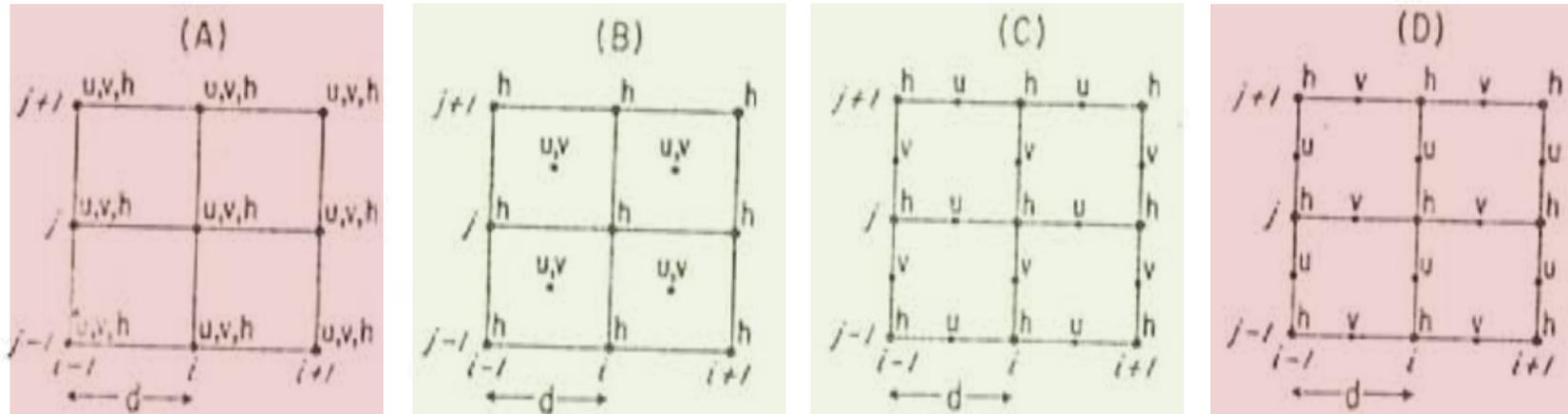
The equations and possible discretizations

$$\frac{\partial h}{\partial t} + \nabla \cdot (h\mathbf{u}) = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla (hg) = 0$$



Studied the properties of the famous Arakawa grids



All finite differences but unknowns placed at different locations

Methods in Computational Physics
Vol. 17, 1977

Computational Design of the Basic
Dynamical Processes of the UCLA
General Circulation Model

AKIO ARAKAWA AND VIVIAN R. LAMB
DEPARTMENT OF ATMOSPHERIC SCIENCES
UNIVERSITY OF CALIFORNIA
LOS ANGELES, CALIFORNIA

Quality of numerical
results depends on
placement of DOFs

Some perform
better than others

Arakawa and Lamb grids

The equations and possible discretizations

Section II describes the principles of mathematical modeling that were followed in the computational design of the basic dynamical processes of the model. The basic principle employed in selecting a space finite-difference scheme from the many that share the same order of accuracy was a requirement that the scheme maintain discrete analogs of a number of physically important integral constraints of the continuous atmosphere. Energy propagation properties in physical space, as well as in spectral space, were also considered in the selection of a scheme.

Section III describes space finite-difference schemes for homogeneous incompressible flow, with and without a free surface. Section III, A shows that the dispersion properties of inertia-gravity waves are highly scheme-dependent and that from the point of view of geostrophic adjustment there is only one satisfactory distribution (staggering) of the dependent variables into grid points.

Methods in Computational Physics
Vol. 17, 1977

Computational Design of the Basic
Dynamical Processes of the UCLA
General Circulation Model

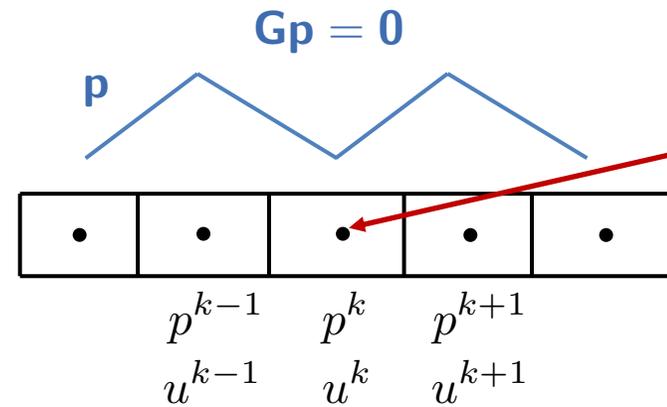
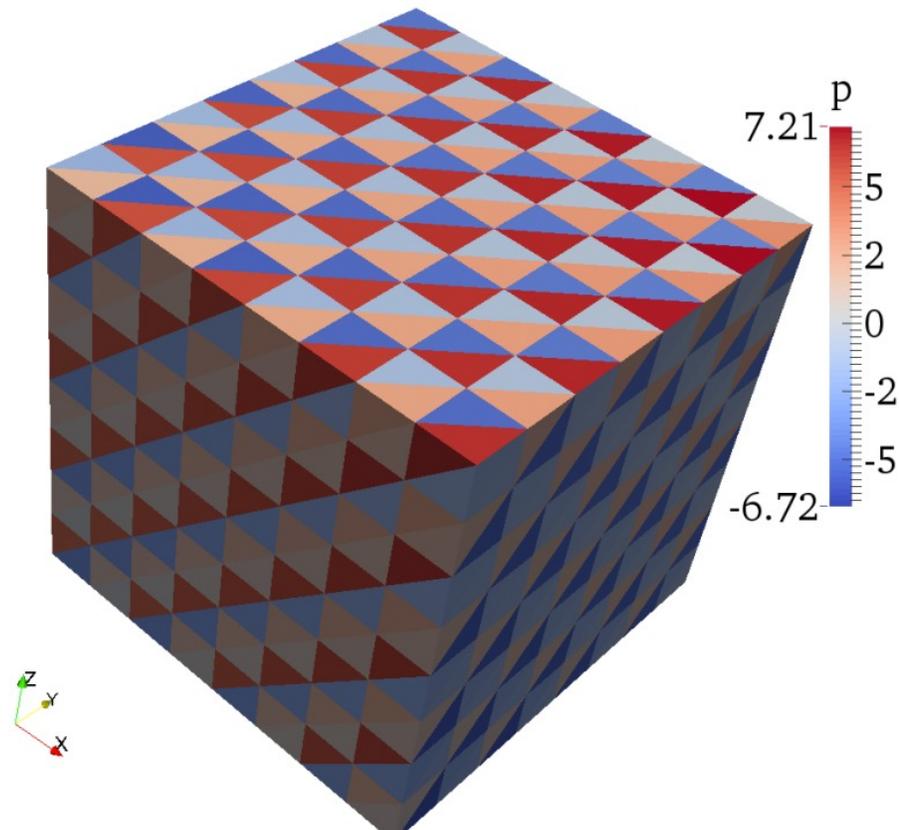
AKIO ARAKAWA AND VIVIAN R. LAMB

DEPARTMENT OF ATMOSPHERIC SCIENCES
UNIVERSITY OF CALIFORNIA
LOS ANGELES, CALIFORNIA

Stokes and Navier-Stokes equations

It is not trivial to obtain adequate discretizations

Navier-Stokes (checkerboard pressure)



$$\nabla p \approx \mathbf{G}p$$

$$(\mathbf{G}p)^k := \frac{p^{k+1} - p^{k-1}}{2\Delta x}$$

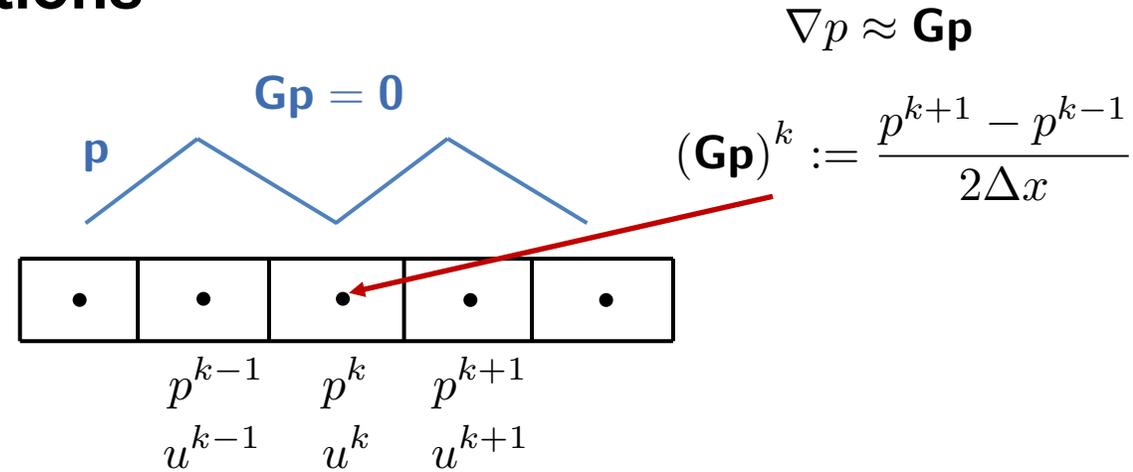
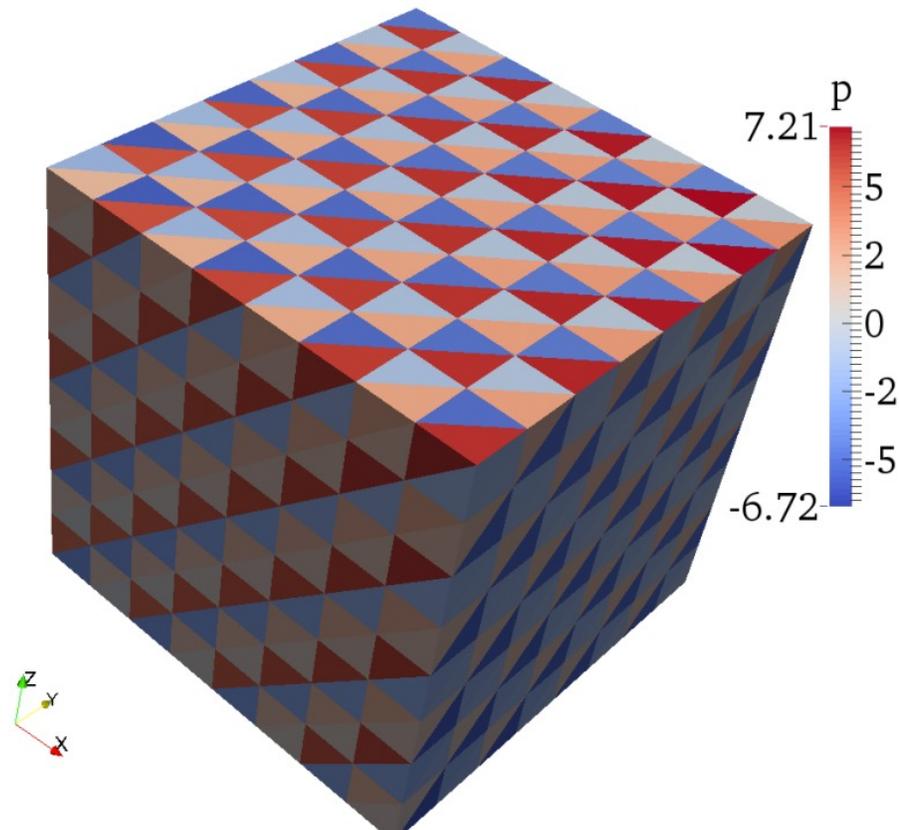
Stokes flow

$$\begin{cases} \Delta \vec{u} - \nabla p = \vec{f} \\ \nabla \cdot \vec{u} = 0 \end{cases}$$

Stokes and Navier-Stokes equations

It is not trivial to obtain adequate discretizations

Navier-Stokes (checkerboard pressure)



Stokes flow

$$\begin{cases} \Delta \vec{u} - \nabla p = \vec{f} \\ \nabla \cdot \vec{u} = 0 \end{cases}$$



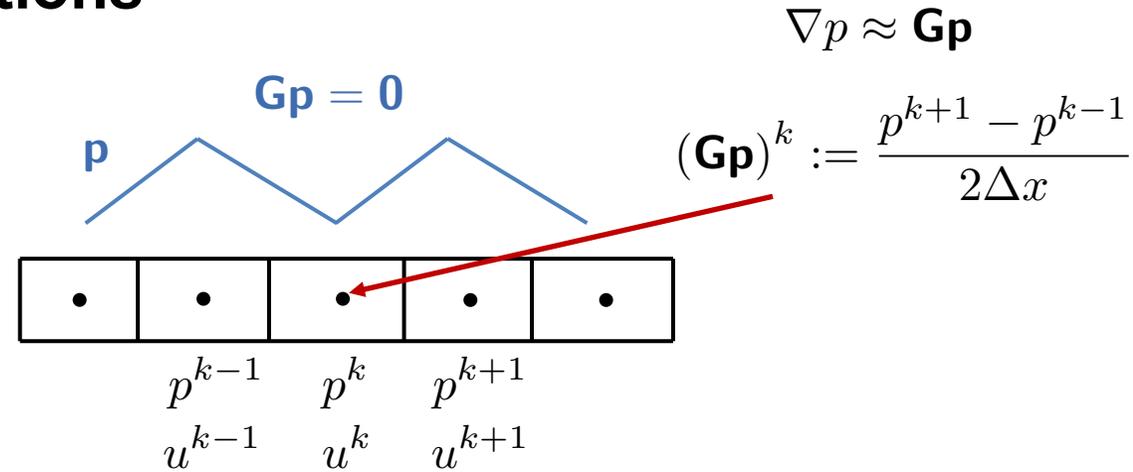
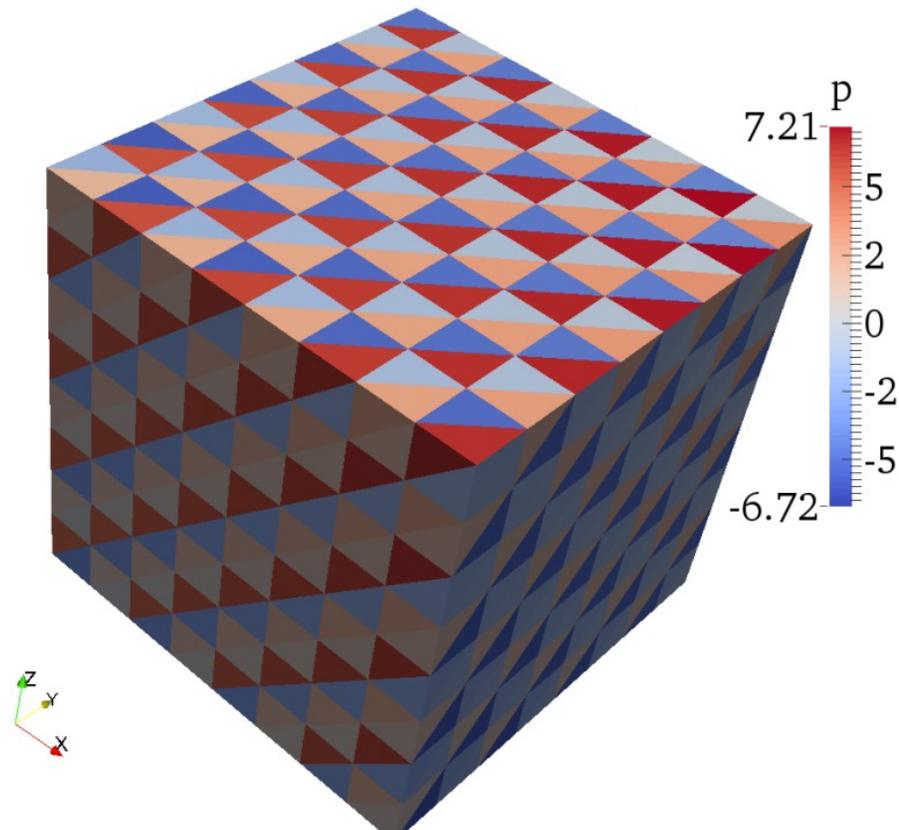
Algebraic system of equations

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_h \\ \mathbf{p}_h \end{bmatrix} = \begin{bmatrix} \mathbf{f}_h \\ \mathbf{0} \end{bmatrix}$$

Stokes and Navier-Stokes equations

It is not trivial to obtain adequate discretizations

Navier-Stokes (checkerboard pressure)



Stokes flow

$$\begin{cases} \Delta \vec{u} - \nabla p = \vec{f} \\ \nabla \cdot \vec{u} = 0 \end{cases}$$



Algebraic system of equations

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_h \\ \mathbf{p}_h \end{bmatrix} = \begin{bmatrix} \mathbf{f}_h \\ \mathbf{0} \end{bmatrix}$$



Schur complement

$$\mathbf{G}^T \mathbf{A}^{-1} \mathbf{G} \mathbf{p}_h = \mathbf{G}^T \mathbf{A} \mathbf{f}_h$$

If $\mathcal{N}(\mathbf{G}) \neq \{0\}$

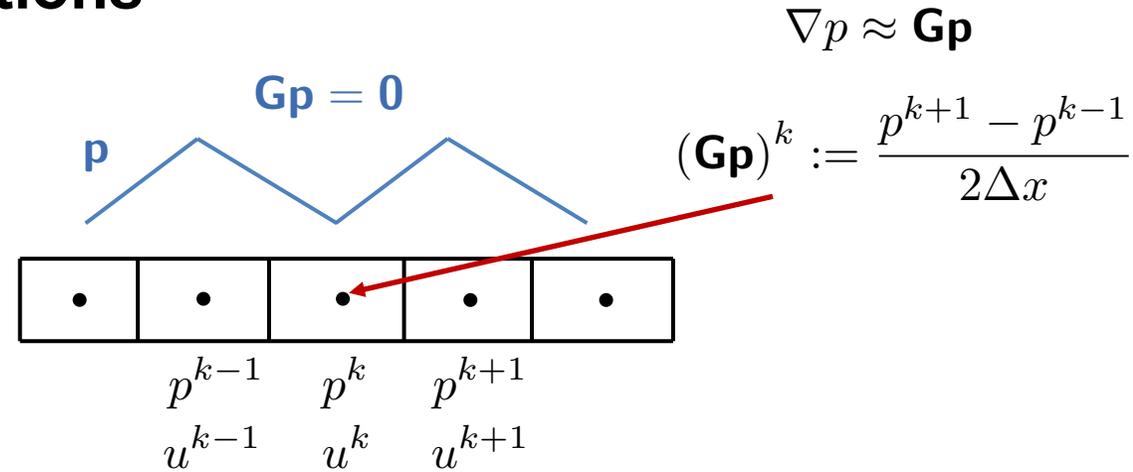
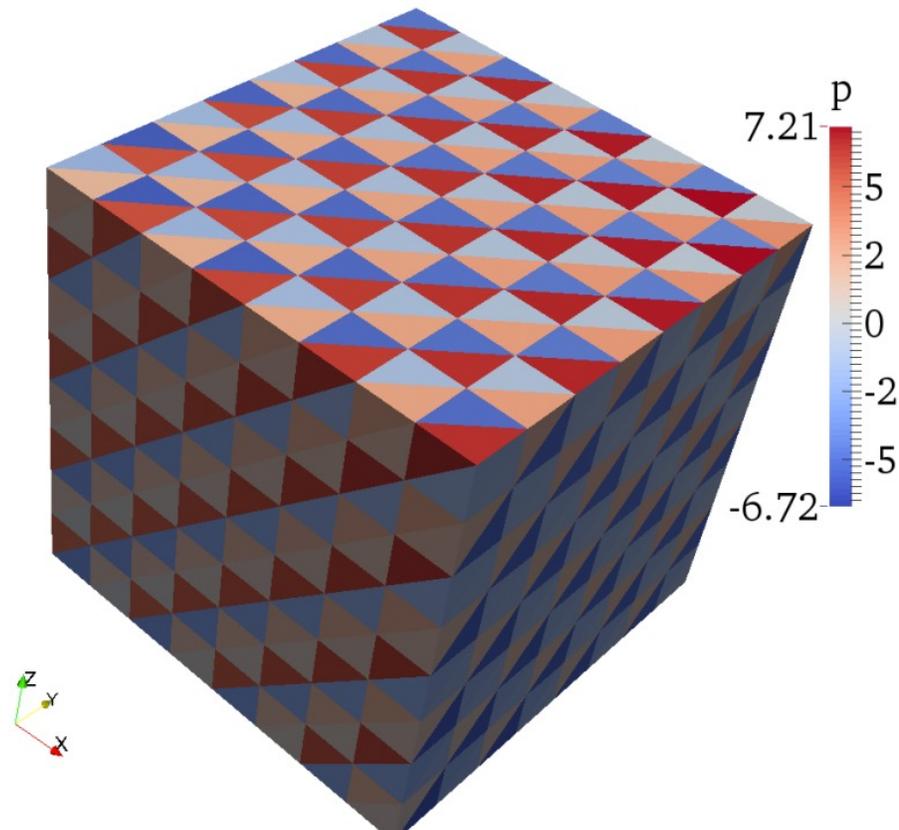


System is singular

Stokes and Navier-Stokes equations

It is not trivial to obtain adequate discretizations

Navier-Stokes (checkerboard pressure)



Stokes flow

$$\begin{cases} \Delta \vec{u} - \nabla p = \vec{f} \\ \nabla \cdot \vec{u} = 0 \end{cases}$$



Algebraic system of equations

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_h \\ \mathbf{p}_h \end{bmatrix} = \begin{bmatrix} \mathbf{f}_h \\ \mathbf{0} \end{bmatrix}$$



Schur complement

$$\mathbf{G}^T \mathbf{A}^{-1} \mathbf{G} \mathbf{p}_h = \mathbf{G}^T \mathbf{A} \mathbf{f}_h$$

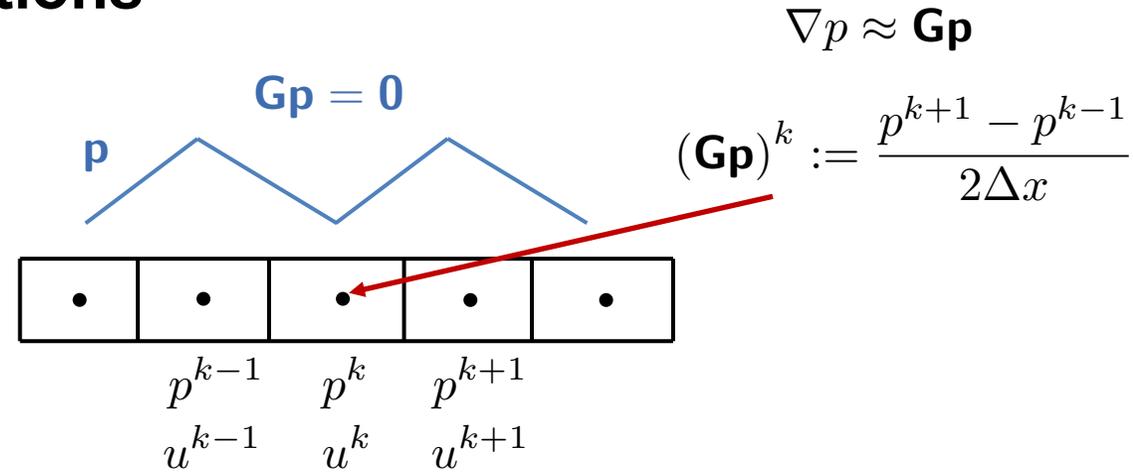
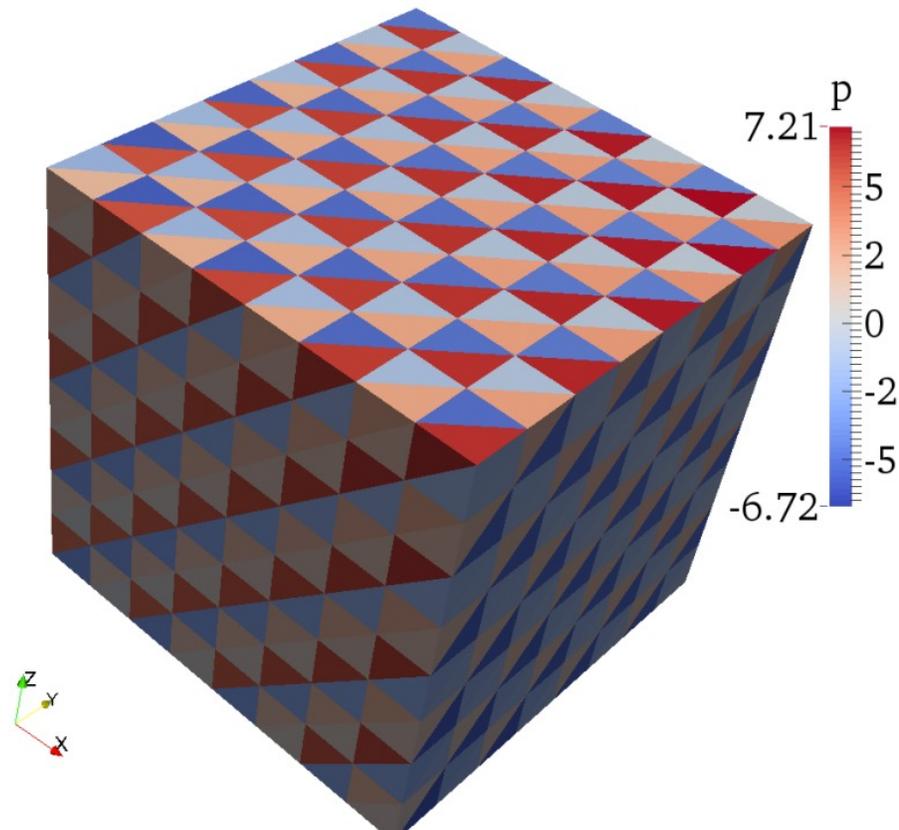
FEM

$$\langle \nabla \cdot \vec{\sigma}, p \rangle = 0 \quad \not\Rightarrow \quad p = \text{constant}$$

Stokes and Navier-Stokes equations

It is not trivial to obtain adequate discretizations

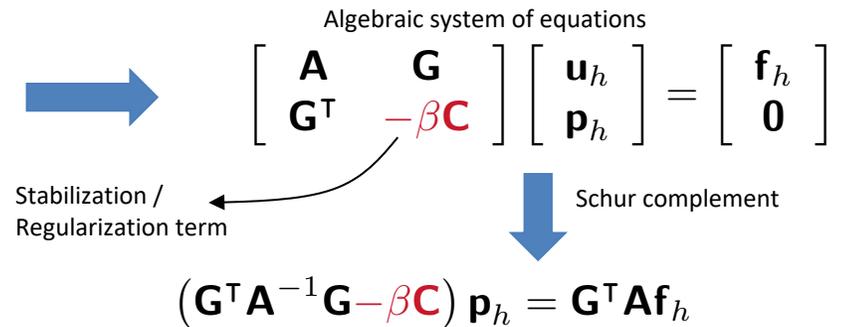
Navier-Stokes (checkerboard pressure)



Stokes flow

$$\begin{cases} \Delta \vec{u} - \nabla p = \vec{f} \\ \nabla \cdot \vec{u} = 0 \end{cases}$$

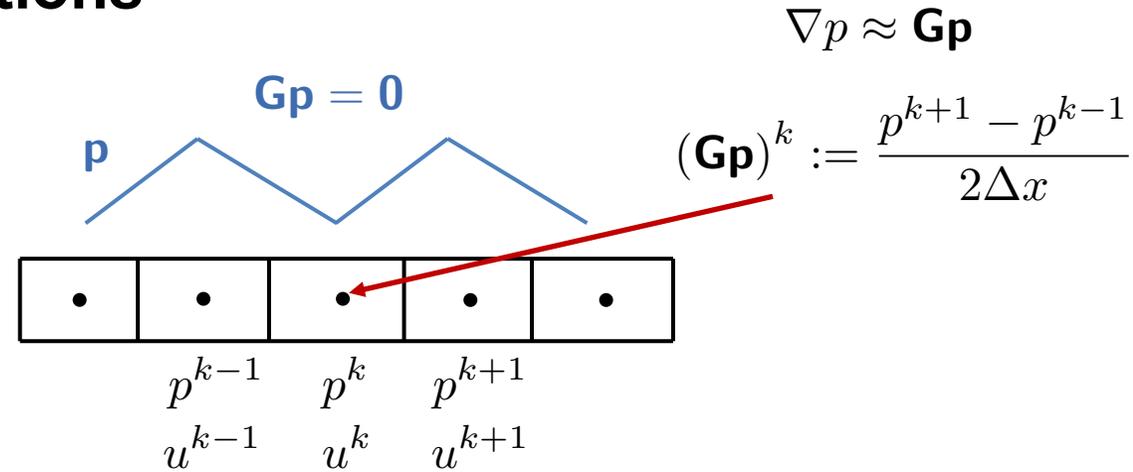
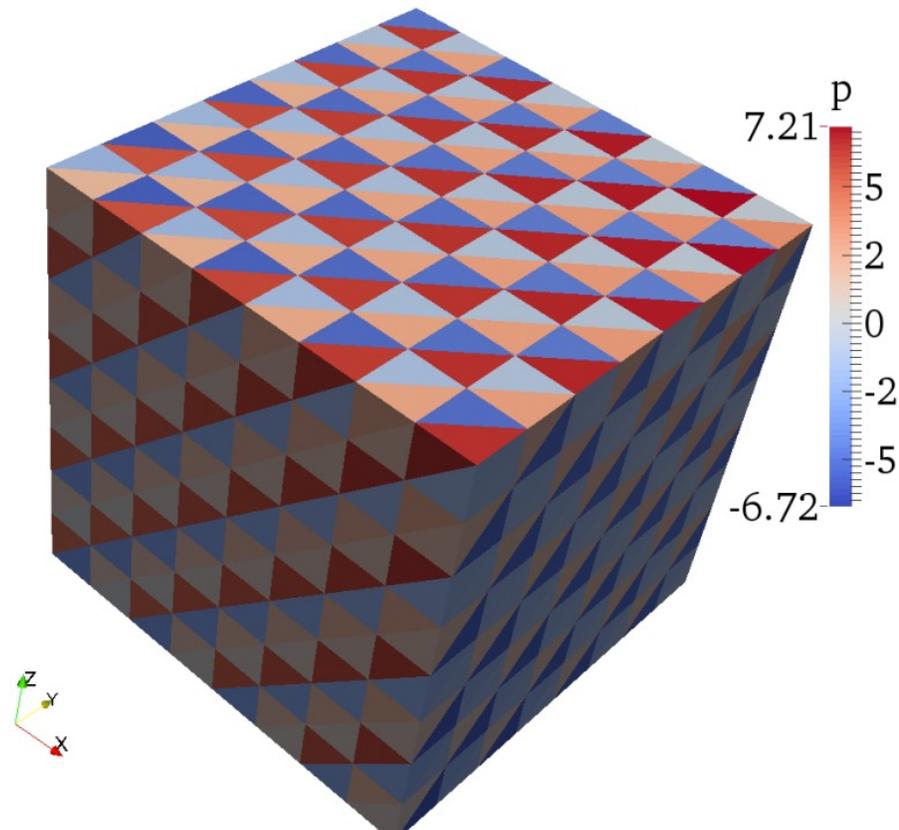
$\beta > 0$ and $\beta \ll 1$



Stokes and Navier-Stokes equations

It is not trivial to obtain adequate discretizations

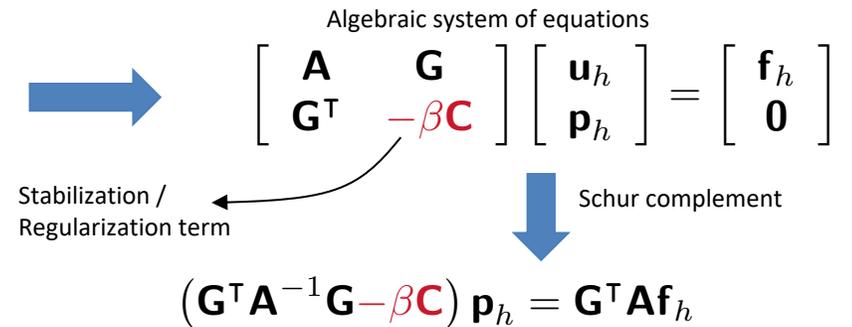
Navier-Stokes (checkerboard pressure)



Stokes flow

$$\begin{cases} \Delta \vec{u} - \nabla p = \vec{f} \\ \nabla \cdot \vec{u} = 0 \end{cases}$$

$\beta > 0$ and $\beta \ll 1$



But...

$$\mathbf{B}^T \mathbf{u}_h = 0$$

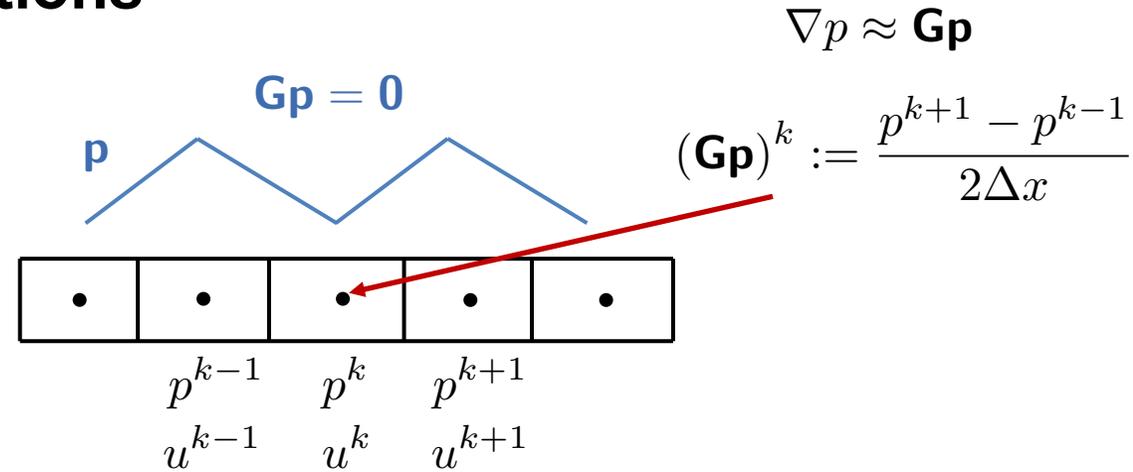
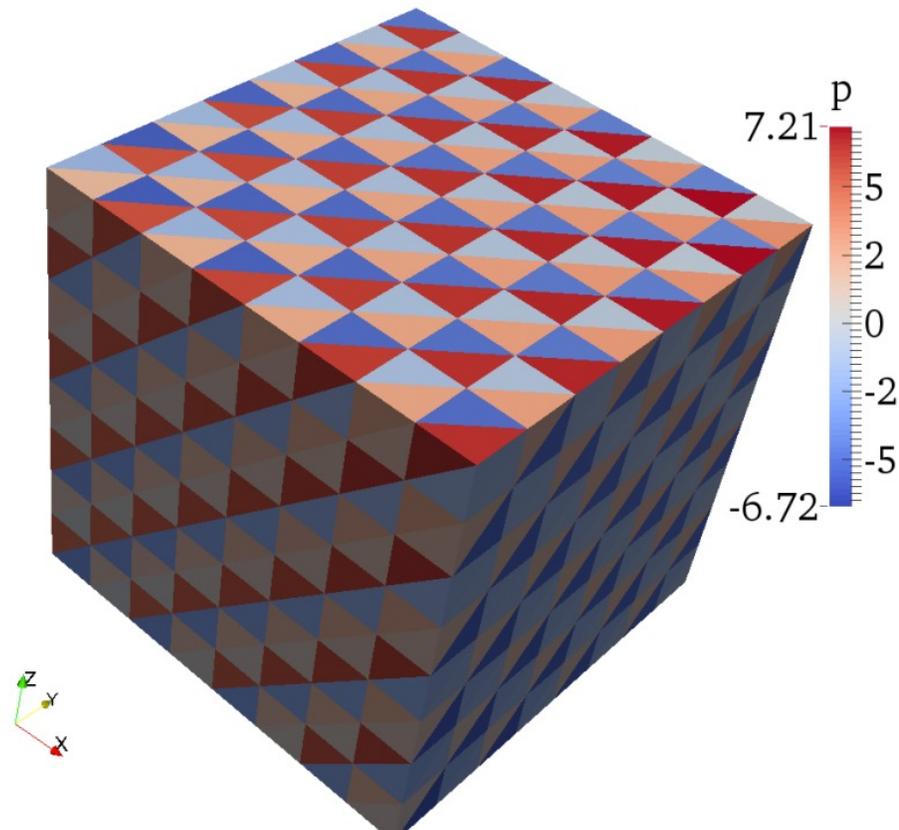


$$\mathbf{B}^T \mathbf{u}_h = \beta \mathbf{C} \mathbf{p}_h \approx 0$$

Stokes and Navier-Stokes equations

It is not trivial to obtain adequate discretizations

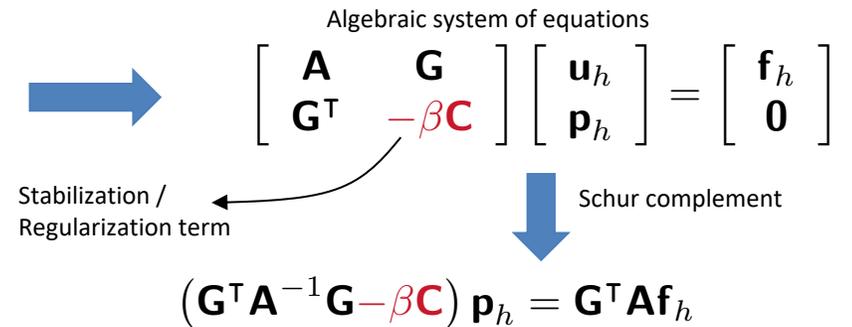
Navier-Stokes (checkerboard pressure)



Stokes flow

$$\begin{cases} \Delta \vec{u} - \nabla p = \vec{f} \\ \nabla \cdot \vec{u} = 0 \end{cases}$$

$\beta > 0$ and $\beta \ll 1$



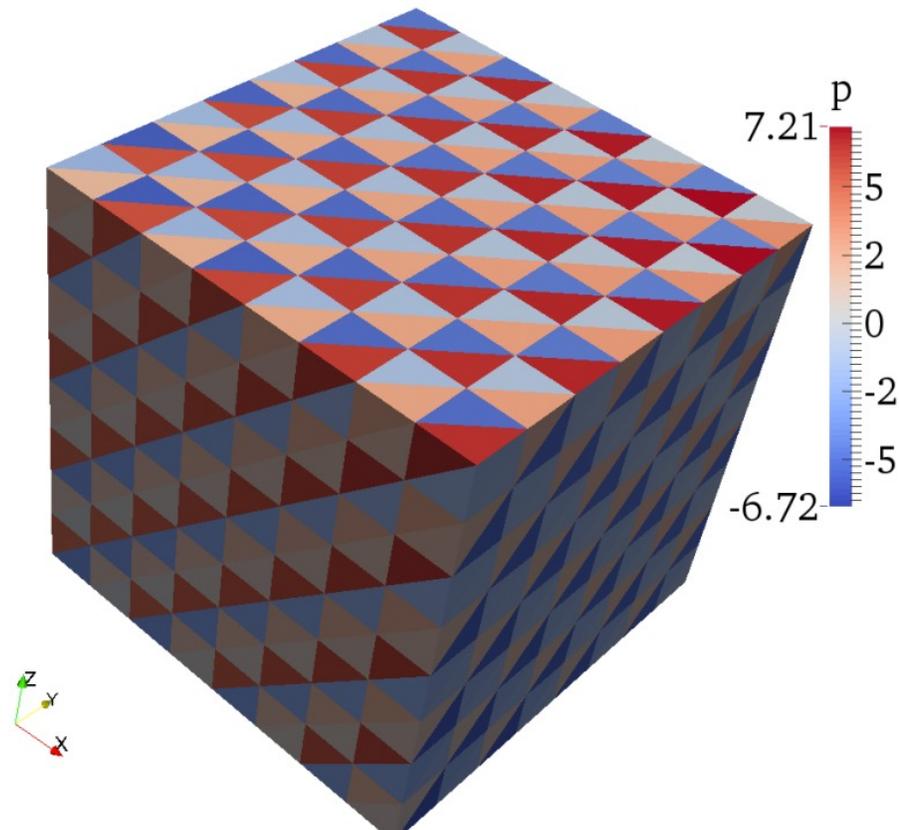
But...

$$\nabla \cdot \vec{u} = 0 \quad \longrightarrow \quad \nabla \cdot \vec{u}_h = \beta p_h \approx 0$$

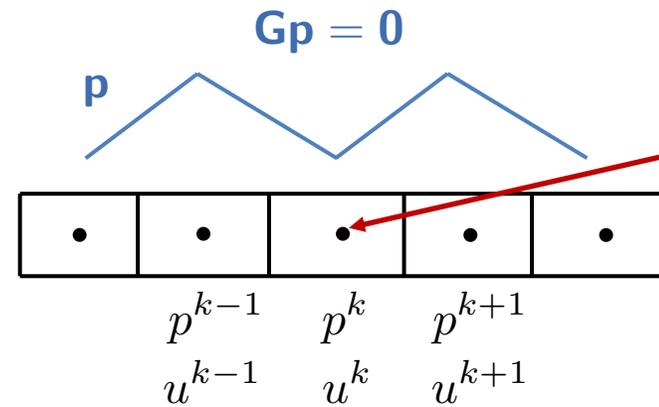
Stokes and Navier-Stokes equations

It is not trivial to obtain adequate discretizations

Navier-Stokes (checkerboard pressure)



$$\nabla p \approx \mathbf{G}p$$

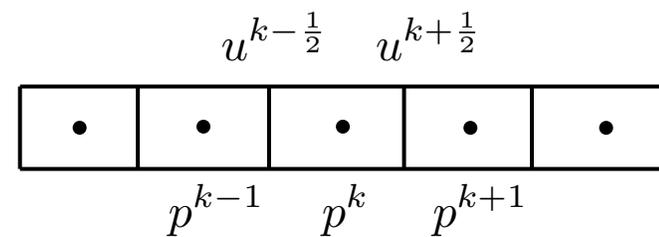


$$(\mathbf{G}p)^k := \frac{p^{k+1} - p^{k-1}}{2\Delta x}$$

Staggering solves this issue without stabilization



Structure preserving discretizations generalise and formalise these ideas



Intro and motivation

Key take-away

Different placement of DOFs leads to numerical methods with different properties. Even when using the same method (FD, FV, FEM). Some are **better**, others are **worse**.

How to obtain the **better numerical methods in a systematic way?**

Spatial discretization

Semi-discretization of shallow waters

Spatial discretization shallow waters

Underlying mathematical structure: de Rham complex

Topological equations

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

$$\nabla \times \vec{H} = \vec{J} + \frac{\partial \vec{D}}{\partial t}$$

$$\nabla \cdot \vec{D} = \rho$$

$$\nabla \cdot \vec{B} = 0$$

Constitutive equations

$$\vec{D} = \epsilon \vec{E}$$

$$\vec{B} = \mu \vec{H}$$

Spatial discretization shallow waters

Underlying mathematical structure: de Rham complex

Topological equations

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

$$\nabla \times \vec{H} = \vec{J} + \frac{\partial \vec{D}}{\partial t}$$

$$\nabla \cdot \vec{D} = \rho$$

$$\nabla \cdot \vec{B} = 0$$

Constitutive equations

$$\vec{D} = \epsilon \vec{E}$$

$$\vec{B} = \mu \vec{H}$$

Implicit relations

$$\nabla \cdot \nabla \times \vec{E} = 0 \quad \Rightarrow \quad \frac{\partial}{\partial t} \nabla \cdot \vec{B} = 0$$

$$\nabla \cdot \nabla \times \vec{H} = 0 \quad \Rightarrow \quad \frac{\partial \rho}{\partial t} + \nabla \cdot \vec{J} = 0$$

$$\frac{\partial \vec{B}}{\partial t} = 0 \quad \Rightarrow \quad \vec{E} = \nabla \phi$$

Spatial discretization shallow waters

Underlying mathematical structure: de Rham complex

Topological equations

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

$$\nabla \times \vec{H} = \vec{J} + \frac{\partial \vec{D}}{\partial t}$$

$$\nabla \cdot \vec{D} = \rho$$

$$\nabla \cdot \vec{B} = 0$$

Constitutive equations

$$\vec{D} = \epsilon \vec{E}$$

$$\vec{B} = \mu \vec{H}$$

Implicit relations

$$\nabla \cdot \nabla \times \vec{E} = 0 \quad \Rightarrow \quad \frac{\partial}{\partial t} \nabla \cdot \vec{B} = 0$$

$$\nabla \cdot \nabla \times \vec{H} = 0 \quad \Rightarrow \quad \frac{\partial \rho}{\partial t} + \nabla \cdot \vec{J} = 0$$

$$\frac{\partial \vec{B}}{\partial t} = 0 \quad \Rightarrow \quad \vec{E} = \nabla \phi$$

Fundamental identities

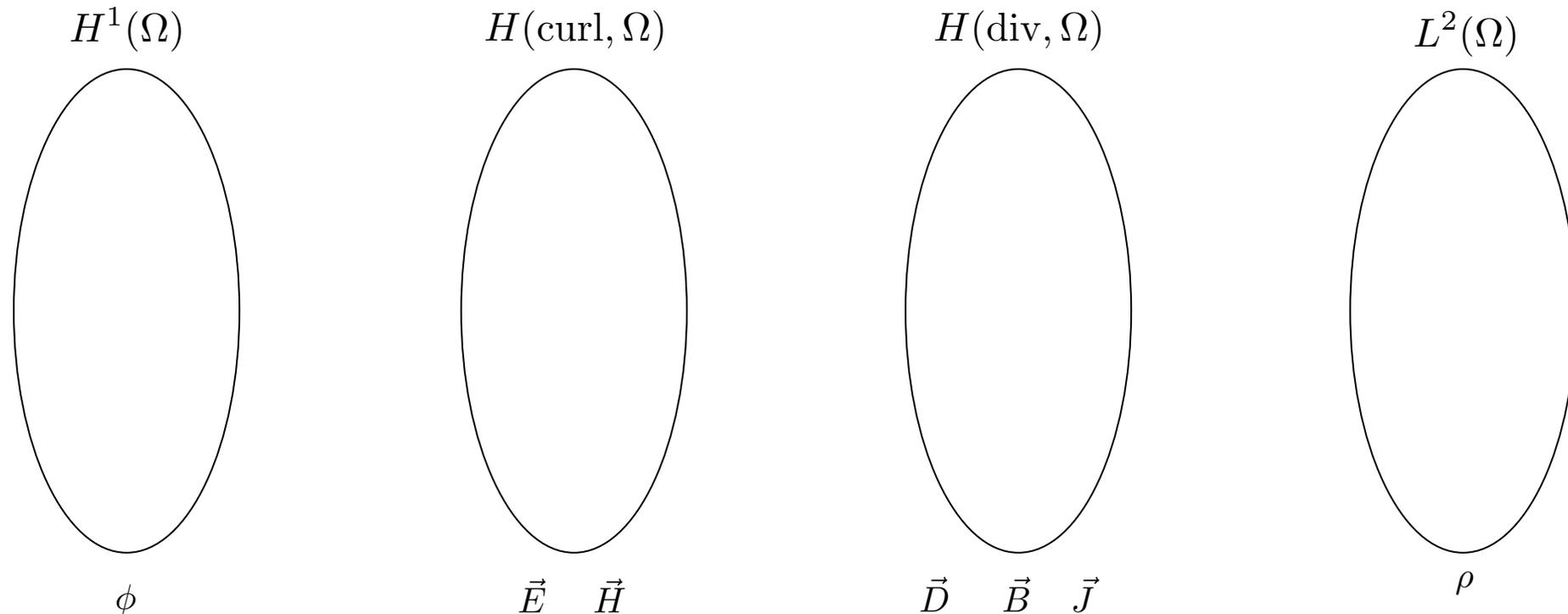
$$\nabla \cdot \nabla \times \vec{A} = 0$$

$$\nabla \times \nabla \psi = 0$$

Spatial discretization shallow waters

Underlying mathematical structure: de Rham complex

Example: Electromagnetics (Maxwell's equations)

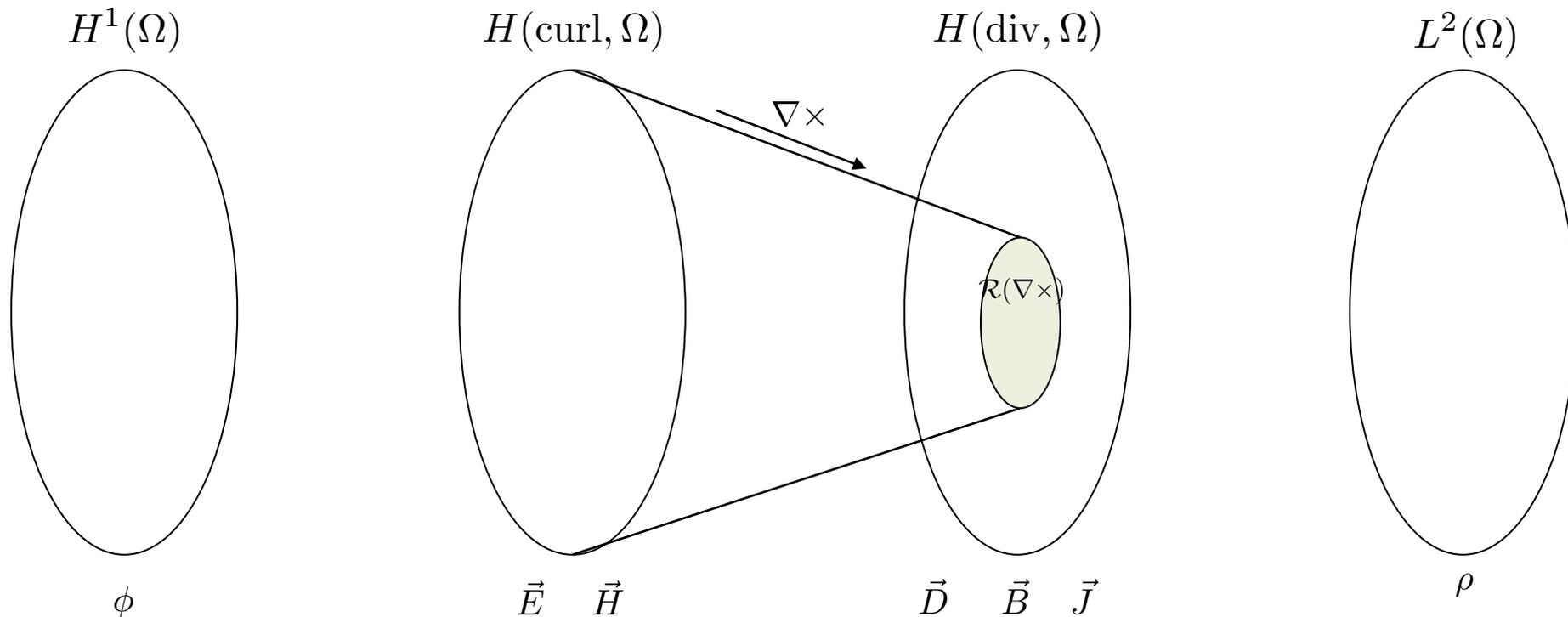


Such that: $\nabla\phi$ $\nabla \times \vec{E}, \nabla \times \vec{H}$ $\nabla \cdot \vec{B}, \nabla \cdot \vec{D}, \nabla \cdot \vec{J}$ make sense.

Spatial discretization shallow waters

Underlying mathematical structure: de Rham complex

Example: Electromagnetics (Maxwell's equations)

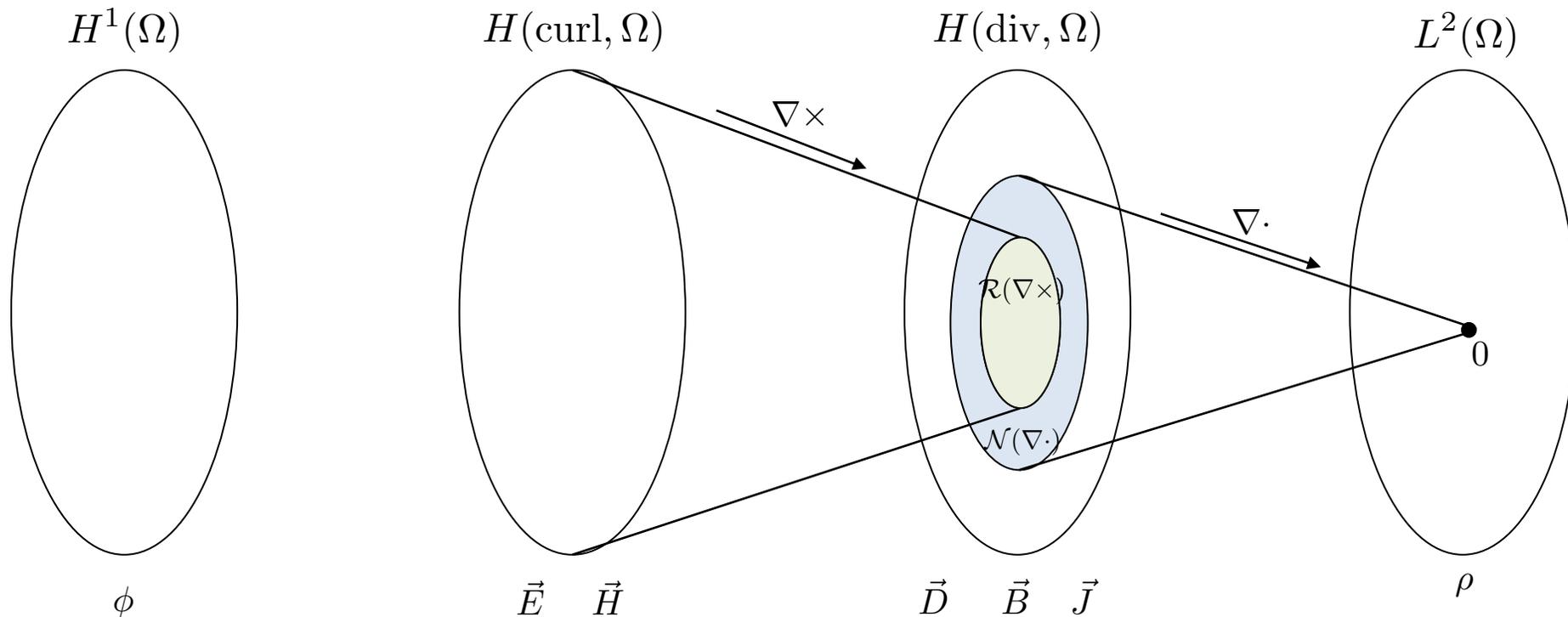


Such that: $\nabla \cdot \nabla \times \vec{E}, \nabla \cdot \nabla \times \vec{H}$ make sense.

Spatial discretization shallow waters

Underlying mathematical structure: de Rham complex

Example: Electromagnetics (Maxwell's equations)

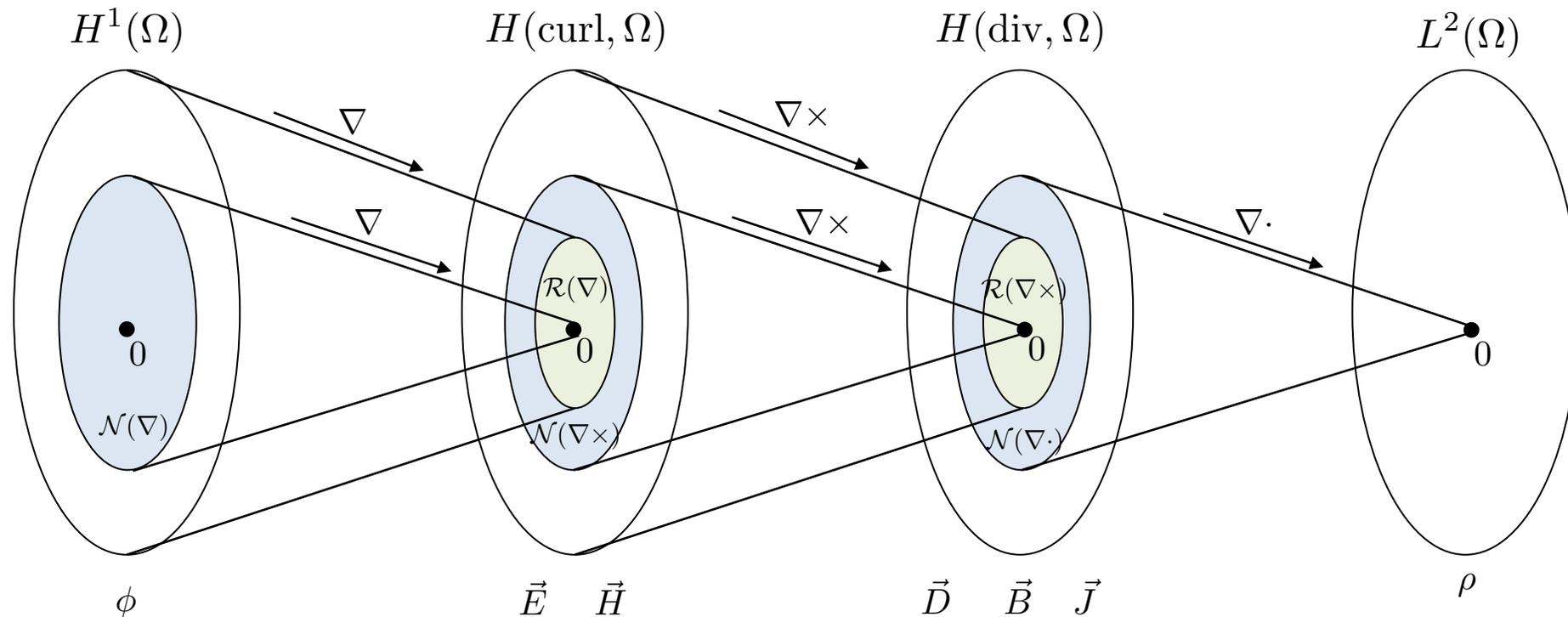


Such that: $\nabla \cdot \nabla \times \vec{E} = \nabla \cdot \nabla \times \vec{H} = 0$ make sense.

Spatial discretization shallow waters

Underlying mathematical structure: de Rham complex

Example: Electromagnetics (Maxwell's equations)

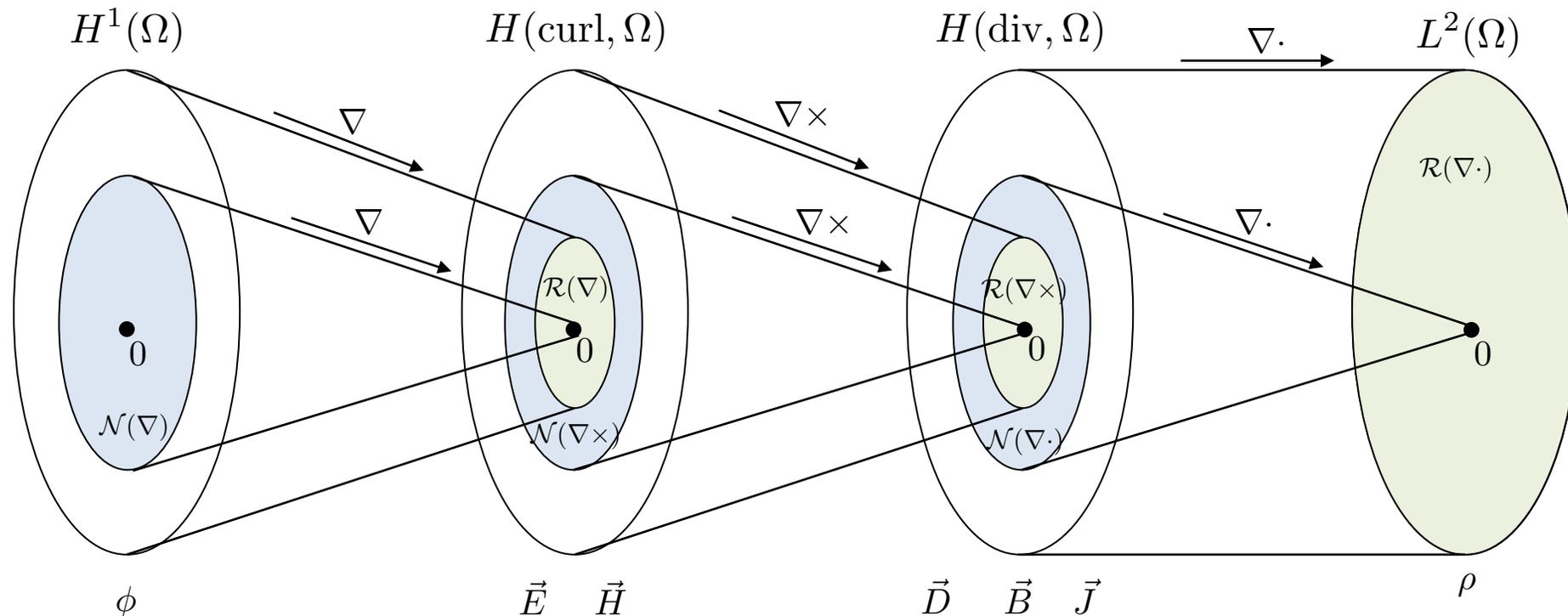


Such that: $\nabla \phi$ $\nabla \times \vec{E}, \nabla \times \vec{H}$ $\nabla \cdot \vec{B}, \nabla \cdot \vec{D}, \nabla \cdot \vec{J}$ make sense.

Spatial discretization shallow waters

Underlying mathematical structure: de Rham complex

Example: Electromagnetics (Maxwell's equations)



Such that: $\nabla \phi$ $\nabla \times \vec{E}, \nabla \times \vec{H}$ $\nabla \cdot \vec{B}, \nabla \cdot \vec{D}, \nabla \cdot \vec{J}$ make sense.

Spatial discretization shallow waters

Underlying mathematical structure: de Rham complex

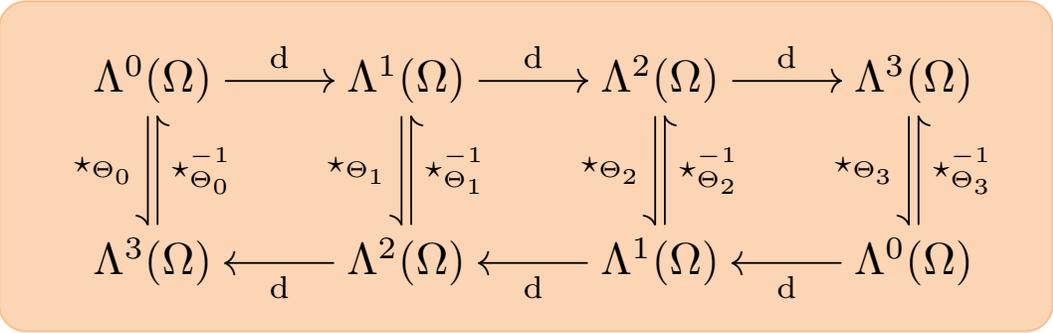
Arnold, Falk, Winther
Bossavit

Spaces constitute an *exact sequence*: the *de Rham complex*

$$H^1(\Theta_0, \Omega) \xrightarrow{\nabla} H(\nabla \times, \Theta_1, \Omega) \xrightarrow{\nabla \times} H(\nabla \cdot, \Theta_2, \Omega) \xrightarrow{\nabla \cdot} L^2(\Theta_3, \Omega)$$

Spatial discretization sha

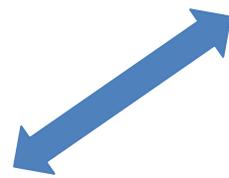
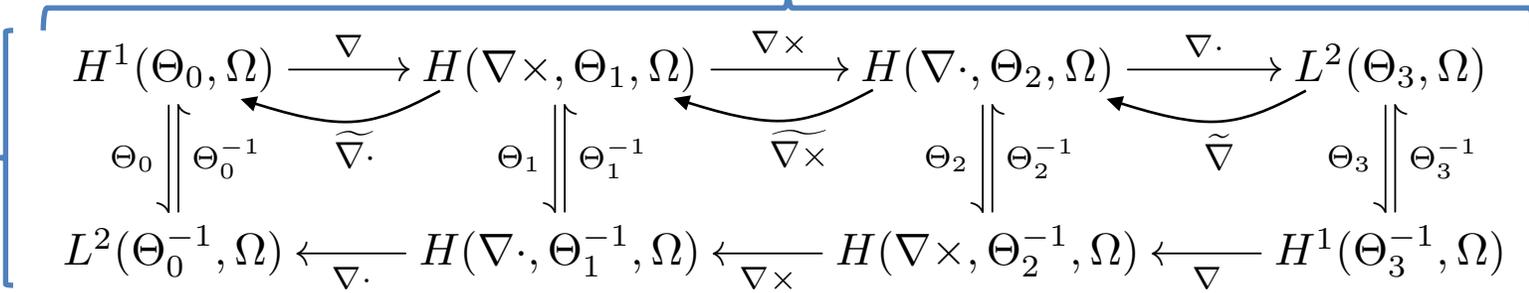
Underlying mathematical structure: de Rham



$$\delta := \star^{-1} d \star$$

Topological relations

Constitutive relations



$$\begin{aligned}
 \tilde{\nabla} &:= \Theta_2^{-1} \nabla \Theta_3 \\
 \tilde{\nabla} \times &:= \Theta_1^{-1} \nabla \times \Theta_2 \\
 \tilde{\nabla} \cdot &:= \Theta_0^{-1} \nabla \cdot \Theta_1
 \end{aligned}$$

- Cohomology and harmonic forms: $\mathcal{N}(d^k) / \mathcal{R}(d^{k-1}) \rightarrow \mathfrak{H}$
- Dual complex: $d_k^* := \Theta_{k-1}^{-1} d^{n-k} \Theta_k$
- Duality: $\mathcal{N}(d^k)^\perp = \mathcal{R}(d_{k+1}^*)$
- Hodge decomposition: $\Lambda^k = \mathcal{R}(d^{k-1}) \oplus \mathfrak{H} \oplus \mathcal{R}(d_{k+1}^*)$
- ...

$$d^0 \rightarrow \nabla, \quad d^1 \rightarrow \nabla \times, \quad d^2 \rightarrow \nabla \cdot$$

$$\alpha^k = d\phi^{k-1} + \gamma^k + d^* \psi^{k+1}$$

Spatial discretization shallow waters

Underlying mathematical structure: de Rham complex

$$\begin{array}{ccccccc}
 G_h(\Theta_0^h, \Omega_h) & \xrightarrow{\mathbf{G}^p} & C_h(\Theta_1^h, \Omega_h) & \xrightarrow{\mathbf{C}^p} & D(\Theta_2^h, \Omega_h) & \xrightarrow{\mathbf{D}^p} & S(\Theta_3^h, \Omega_h) \\
 \Theta_0^h \int (\Theta_0^h)^{-1} & & \Theta_1^h \int (\Theta_1^h)^{-1} & & \Theta_2^h \int (\Theta_2^h)^{-1} & & \Theta_3^h \int (\Theta_3^h)^{-1} \\
 \tilde{S}_h((\Theta_0^h)^{-1}, \Omega_h) & \xleftarrow{\mathbf{D}^d} & \tilde{D}_h((\Theta_1^h)^{-1}, \Omega_h) & \xleftarrow{\mathbf{C}^d} & \tilde{C}_h(\Theta_2^{-1}, \Omega_h) & \xleftarrow{\mathbf{G}^d} & \tilde{G}_h((\Theta_3^h)^{-1}, \Omega_h)
 \end{array}$$

We just need to find a discrete version!

Discrete world

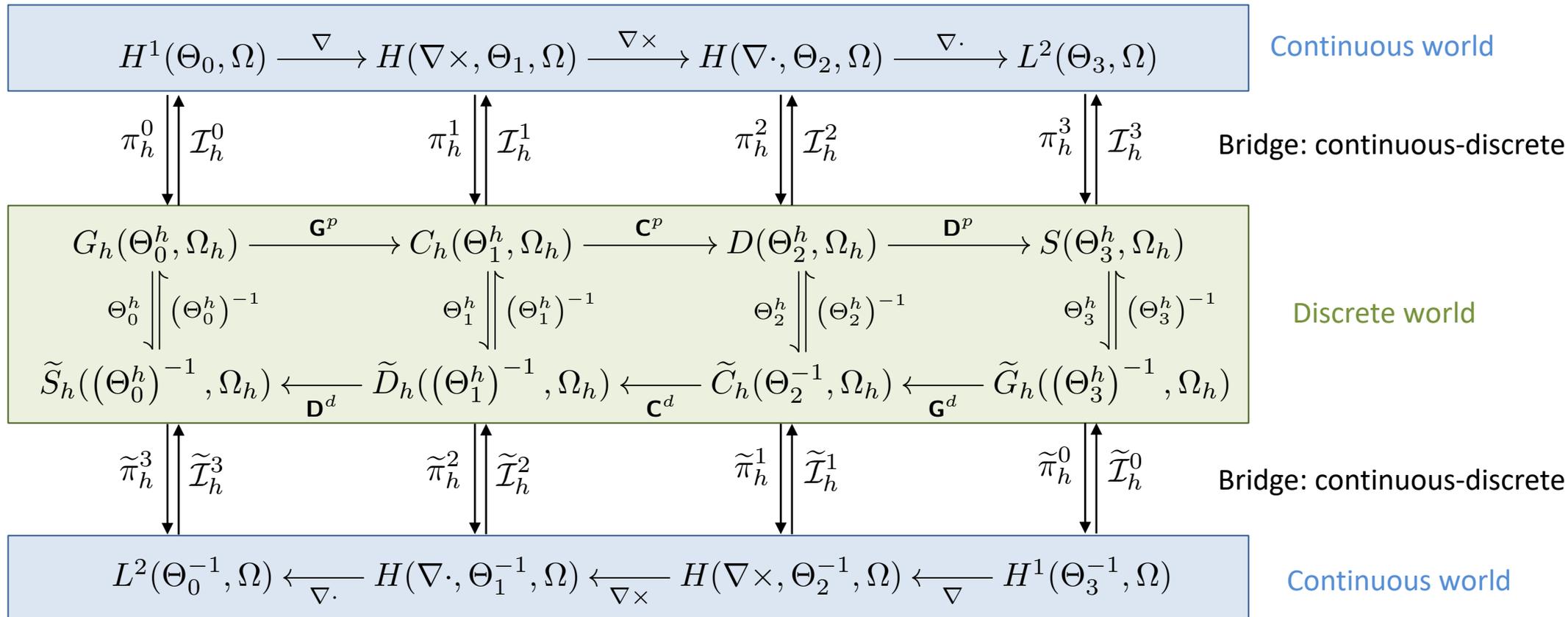
Not quite! It needs to relate to the continuous

Spatial discretization shallow waters

If equal, it would be the holy grail!

Underlying mathematical structure: discrete de Rham complex

$$\mathcal{I}_h^k \neq (\pi_h^k)^{-1}$$



π_h^k Projection onto the primal discrete space k

\mathcal{I}_h^k Interpolator into the continuous primal space k

$\tilde{\pi}_h^k$ Projection onto the dual discrete space k

$\tilde{\mathcal{I}}_h^k$ Interpolator into the continuous dual space k

Spatial discretization shallow waters

How does this compare to Arakawa and Lamb?

Spatial discretization shallow waters

How does this compare to Arakawa and Lamb?

Arakawa and Lamb

$$\frac{\partial h}{\partial t} + \nabla \cdot \mathbf{F} = 0$$

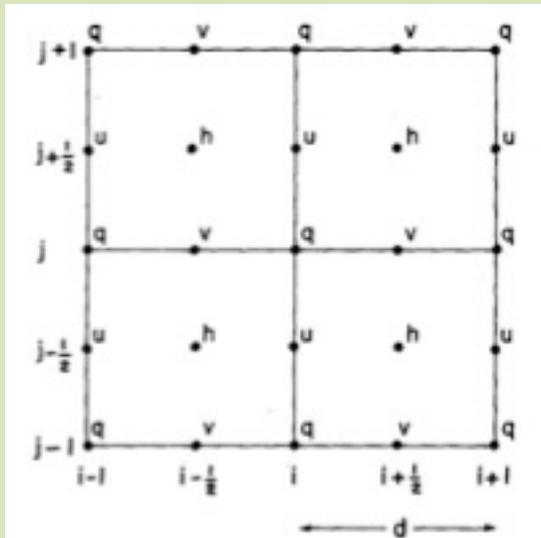
$$\frac{\partial \mathbf{u}}{\partial t} + q \mathbf{e}_z \times \mathbf{F} + \nabla (K + hg) = 0$$

$$K := \frac{1}{2} \mathbf{u} \cdot \mathbf{u}$$

$$q := \frac{\omega}{h}$$

$$\omega := \mathbf{e}_z \cdot \nabla \times \mathbf{u}$$

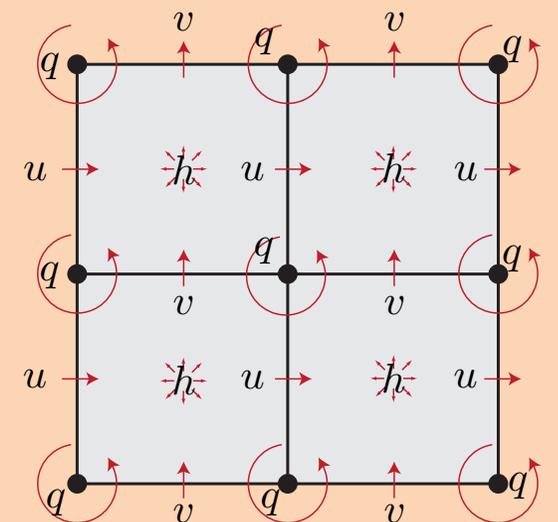
$$\mathbf{F} := h\mathbf{u}$$



e.g., FEM

$$\left\{ \begin{array}{l} \langle \sigma_h, \frac{\partial \mathbf{u}_h}{\partial t} \rangle = -\langle \sigma_h, \mathbf{q}_h \times \mathbf{F}_h \rangle + \langle \nabla \cdot \sigma_h, K_h + gh_h \rangle, \quad \forall \sigma_h \in D_p(\Omega_h) \\ \langle \theta_h, \frac{\partial h_h}{\partial t} \rangle = -\langle \theta_h, \nabla \cdot \mathbf{F}_h \rangle, \quad \forall \theta_h \in S_p(\Omega_h) \\ \langle \sigma_h, \mathbf{F}_h \rangle = \langle \sigma_h, h_h \mathbf{u}_h \rangle, \quad \forall \sigma_h \in D_p(\Omega_h) \\ \langle \theta_h, K_h \rangle = \langle \theta_h, \frac{1}{2} \mathbf{u}_h \cdot \mathbf{u}_h \rangle, \quad \forall \theta_h \in S_p(\Omega_h) \\ \langle \xi_h, h_h \mathbf{q}_h \rangle = \langle \nabla \times \xi_h, \mathbf{u}_h \rangle, \quad \forall \xi_h \in G_p(\Omega_h) \end{array} \right.$$

Lowest order
p = 1



Spatial discretization shallow waters

How does this compare to Arakawa and Lamb?

Arakawa and Lamb

$$\frac{\partial h}{\partial t} + \nabla \cdot \mathbf{F} = 0$$

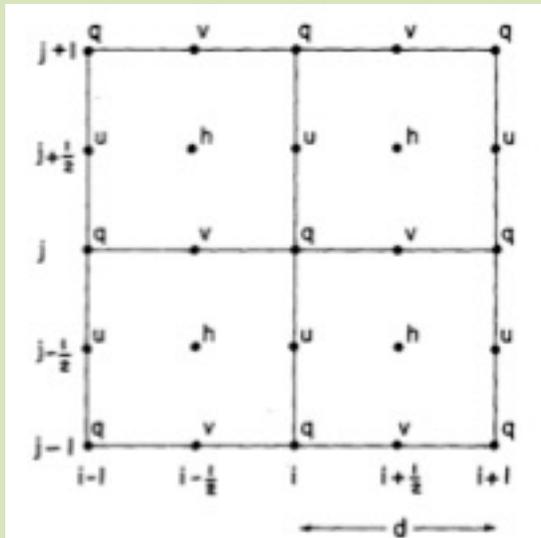
$$\frac{\partial \mathbf{u}}{\partial t} + q \mathbf{e}_z \times \mathbf{F} + \nabla (K + hg) = 0 \quad \text{All pointwise values}$$

$$K := \frac{1}{2} \mathbf{u} \cdot \mathbf{u}$$

$$q := \frac{\omega}{h}$$

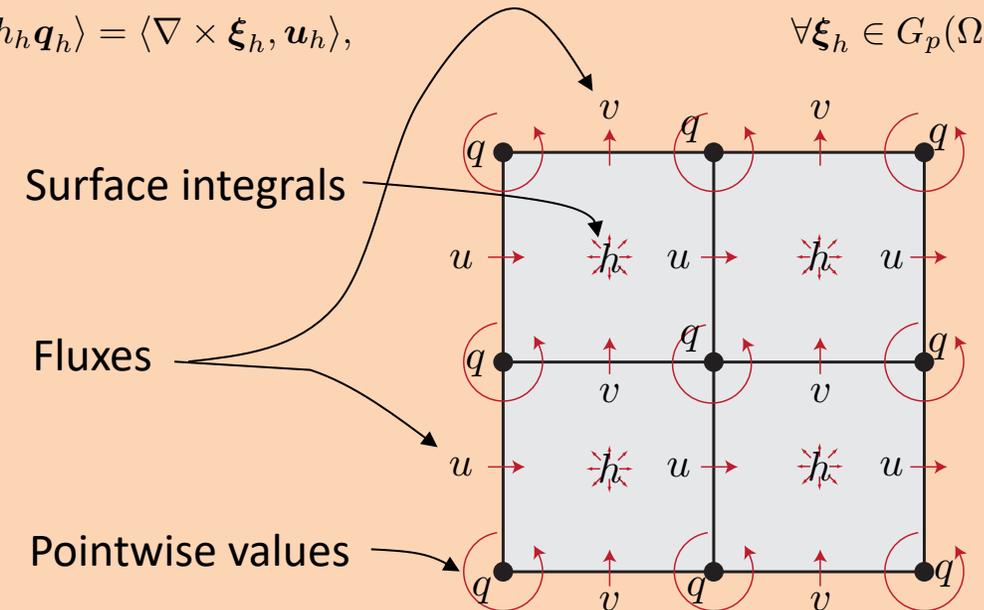
$$\omega := \mathbf{e}_z \cdot \nabla \times \mathbf{u}$$

$$\mathbf{F} := h\mathbf{u}$$



e.g., FEM

$$\left\{ \begin{array}{l} \langle \boldsymbol{\sigma}_h, \frac{\partial \mathbf{u}_h}{\partial t} \rangle = -\langle \boldsymbol{\sigma}_h, \mathbf{q}_h \times \mathbf{F}_h \rangle + \langle \nabla \cdot \boldsymbol{\sigma}_h, K_h + gh_h \rangle, \quad \forall \boldsymbol{\sigma}_h \in D_p(\Omega_h) \\ \langle \theta_h, \frac{\partial h_h}{\partial t} \rangle = -\langle \theta_h, \nabla \cdot \mathbf{F}_h \rangle, \quad \forall \theta_h \in S_p(\Omega_h) \\ \langle \boldsymbol{\sigma}_h, \mathbf{F}_h \rangle = \langle \boldsymbol{\sigma}_h, h_h \mathbf{u}_h \rangle, \quad \forall \boldsymbol{\sigma}_h \in D_p(\Omega_h) \\ \langle \theta_h, K_h \rangle = \langle \theta_h, \frac{1}{2} \mathbf{u}_h \cdot \mathbf{u}_h \rangle, \quad \forall \theta_h \in S_p(\Omega_h) \\ \langle \boldsymbol{\xi}_h, h_h \mathbf{q}_h \rangle = \langle \nabla \times \boldsymbol{\xi}_h, \mathbf{u}_h \rangle, \quad \forall \boldsymbol{\xi}_h \in G_p(\Omega_h) \end{array} \right.$$



Spatial discretization shallow waters

How does this compare to Arakawa and Lamb?

Arakawa and Lamb

$$\frac{\partial h}{\partial t} + \nabla \cdot \mathbf{F} = 0$$

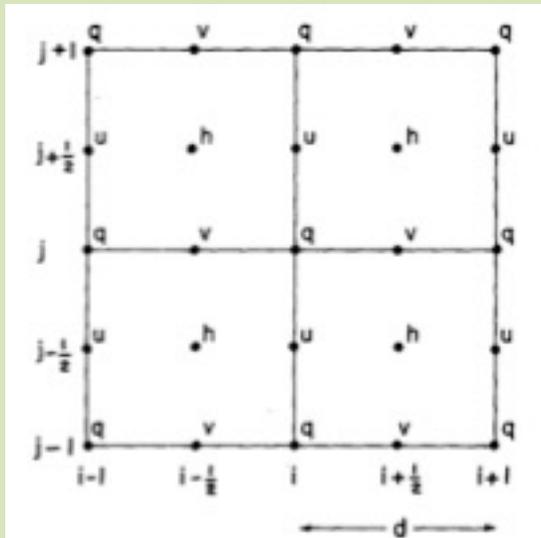
$$\frac{\partial \mathbf{u}}{\partial t} + q \mathbf{e}_z \times \mathbf{F} + \nabla (K + hg) = 0 \quad \text{All pointwise values}$$

$$K := \frac{1}{2} \mathbf{u} \cdot \mathbf{u}$$

$$q := \frac{\omega}{h}$$

$$\omega := \mathbf{e}_z \cdot \nabla \times \mathbf{u}$$

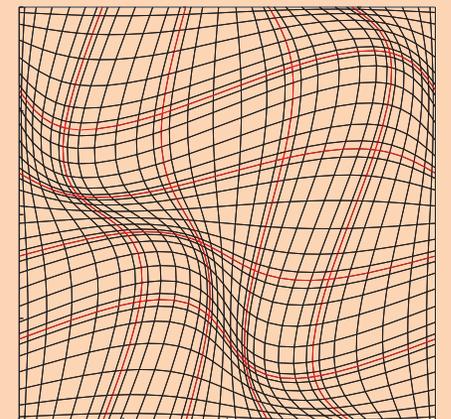
$$\mathbf{F} := h\mathbf{u}$$



e.g., FEM

$$\left\{ \begin{array}{l} \langle \sigma_h, \frac{\partial \mathbf{u}_h}{\partial t} \rangle = -\langle \sigma_h, \mathbf{q}_h \times \mathbf{F}_h \rangle + \langle \nabla \cdot \sigma_h, K_h + gh_h \rangle, \quad \forall \sigma_h \in D_p(\Omega_h) \\ \langle \theta_h, \frac{\partial h_h}{\partial t} \rangle = -\langle \theta_h, \nabla \cdot \mathbf{F}_h \rangle, \quad \forall \theta_h \in S_p(\Omega_h) \\ \langle \sigma_h, \mathbf{F}_h \rangle = \langle \sigma_h, h_h \mathbf{u}_h \rangle, \quad \forall \sigma_h \in D_p(\Omega_h) \\ \langle \theta_h, K_h \rangle = \langle \theta_h, \frac{1}{2} \mathbf{u}_h \cdot \mathbf{u}_h \rangle, \quad \forall \theta_h \in S_p(\Omega_h) \\ \langle \xi_h, h_h \mathbf{q}_h \rangle = \langle \nabla \times \xi_h, \mathbf{u}_h \rangle, \quad \forall \xi_h \in G_p(\Omega_h) \end{array} \right.$$

Directly obtain a high order (arbitrary p) approximation



4x4 elements of degree 7

Spatial discretization shallow waters

How does this compare to Arakawa and Lamb?

Arakawa and Lamb

$$\frac{\partial h}{\partial t} + \nabla \cdot \mathbf{F} = 0$$

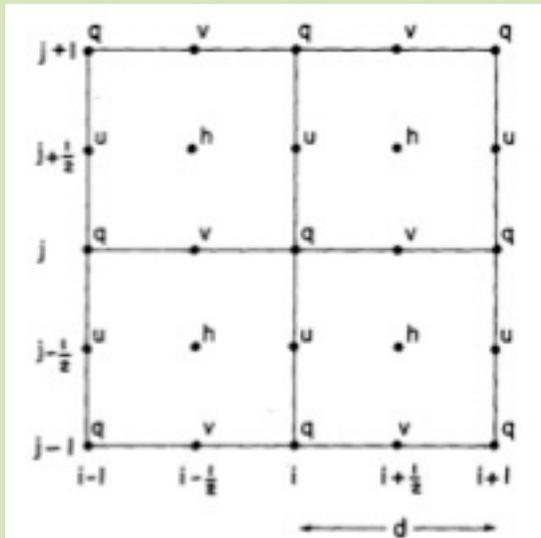
$$\frac{\partial \mathbf{u}}{\partial t} + q \mathbf{e}_z \times \mathbf{F} + \nabla (K + hg) = 0 \quad \text{All pointwise values}$$

$$K := \frac{1}{2} \mathbf{u} \cdot \mathbf{u}$$

$$q := \frac{\omega}{h}$$

$$\omega := \mathbf{e}_z \cdot \nabla \times \mathbf{u}$$

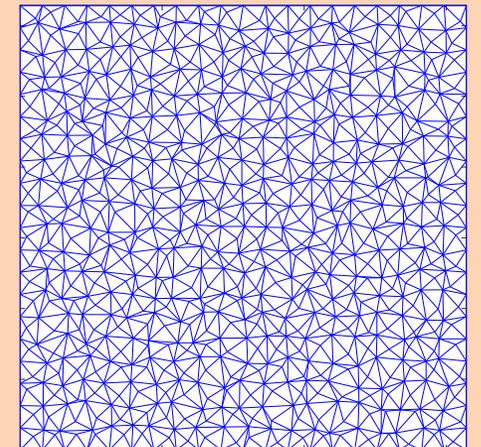
$$\mathbf{F} := h\mathbf{u}$$



e.g., FEM

$$\left\{ \begin{array}{l} \langle \sigma_h, \frac{\partial \mathbf{u}_h}{\partial t} \rangle = -\langle \sigma_h, \mathbf{q}_h \times \mathbf{F}_h \rangle + \langle \nabla \cdot \sigma_h, K_h + gh_h \rangle, \quad \forall \sigma_h \in D_p(\Omega_h) \\ \langle \theta_h, \frac{\partial h_h}{\partial t} \rangle = -\langle \theta_h, \nabla \cdot \mathbf{F}_h \rangle, \quad \forall \theta_h \in S_p(\Omega_h) \\ \langle \sigma_h, \mathbf{F}_h \rangle = \langle \sigma_h, h_h \mathbf{u}_h \rangle, \quad \forall \sigma_h \in D_p(\Omega_h) \\ \langle \theta_h, K_h \rangle = \langle \theta_h, \frac{1}{2} \mathbf{u}_h \cdot \mathbf{u}_h \rangle, \quad \forall \theta_h \in S_p(\Omega_h) \\ \langle \xi_h, h_h \mathbf{q}_h \rangle = \langle \nabla \times \xi_h, \mathbf{u}_h \rangle, \quad \forall \xi_h \in G_p(\Omega_h) \end{array} \right.$$

Use either structured or unstructured grids



Structure preserving discretizations

Key take-aways

1. Conservation of the underlying structure of a PDE is key for accurate numerical discretizations (de Rham complex: preserve exactness and cohomology)
2. Numerical approximation goes (far) beyond truncation error.
3. PDEs have different complexes associated to it.
4. Numerical discretizations reproduce a discrete version of these complexes

**How to preserve these properties in time?
That is another story**

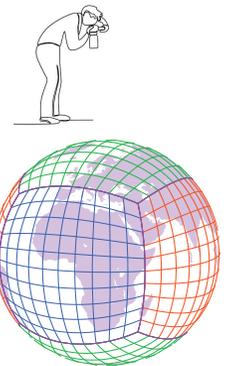
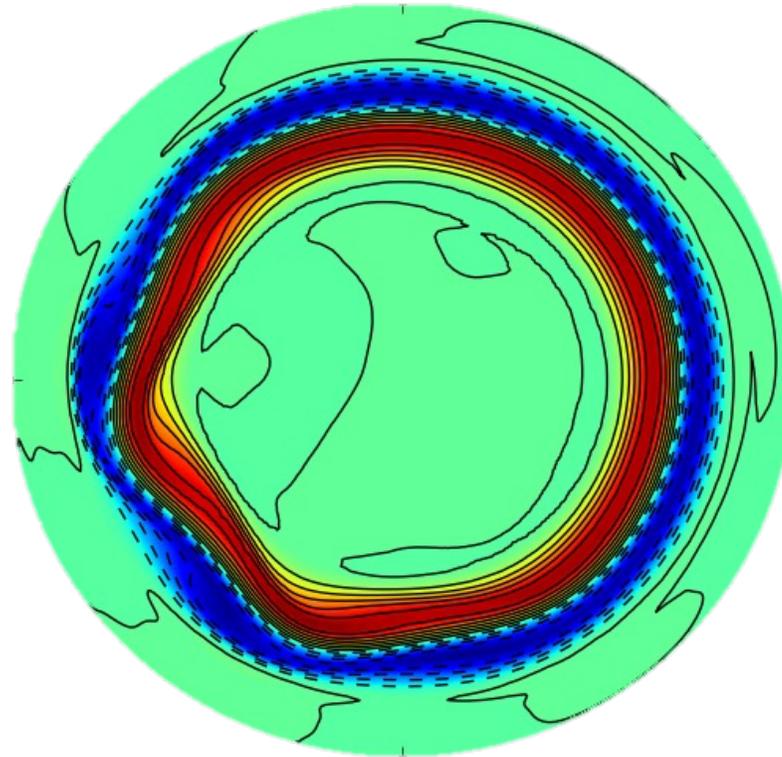
Some applications

Time dependent problems

Shallow waters on a sphere

Nonlinear instability on a sphere: non-affine mesh (day 4)

View of the North pole

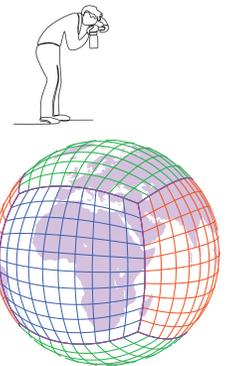
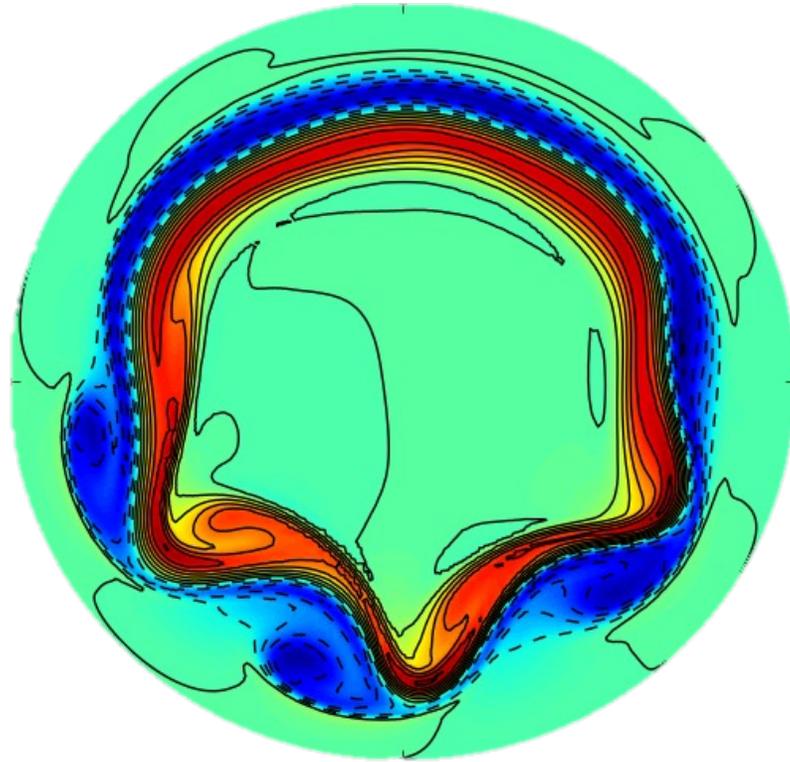


Time dependent problems

Shallow waters on a sphere

Nonlinear instability on a sphere: non-affine mesh (day 5)

View of the North pole

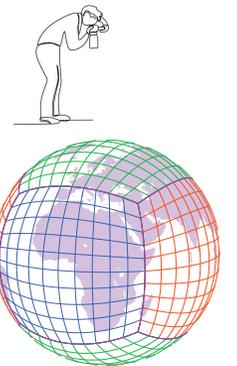
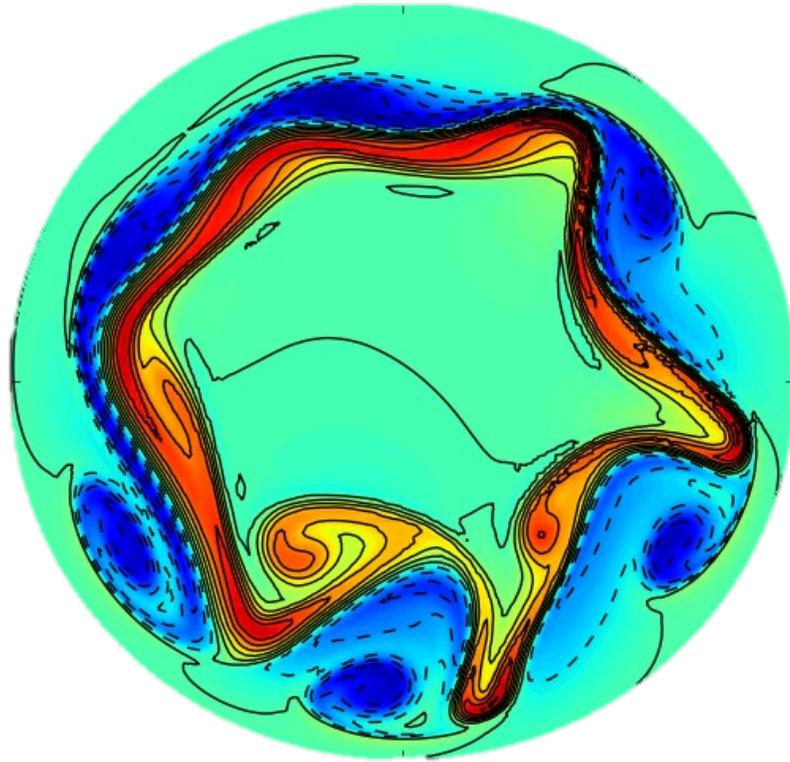


Time dependent problems

Shallow waters on a sphere

Nonlinear instability on a sphere: non-affine mesh (day 6)

View of the North pole

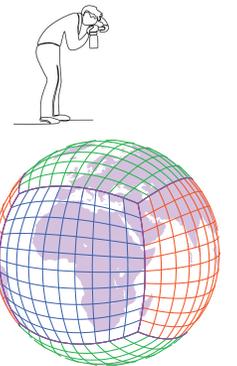
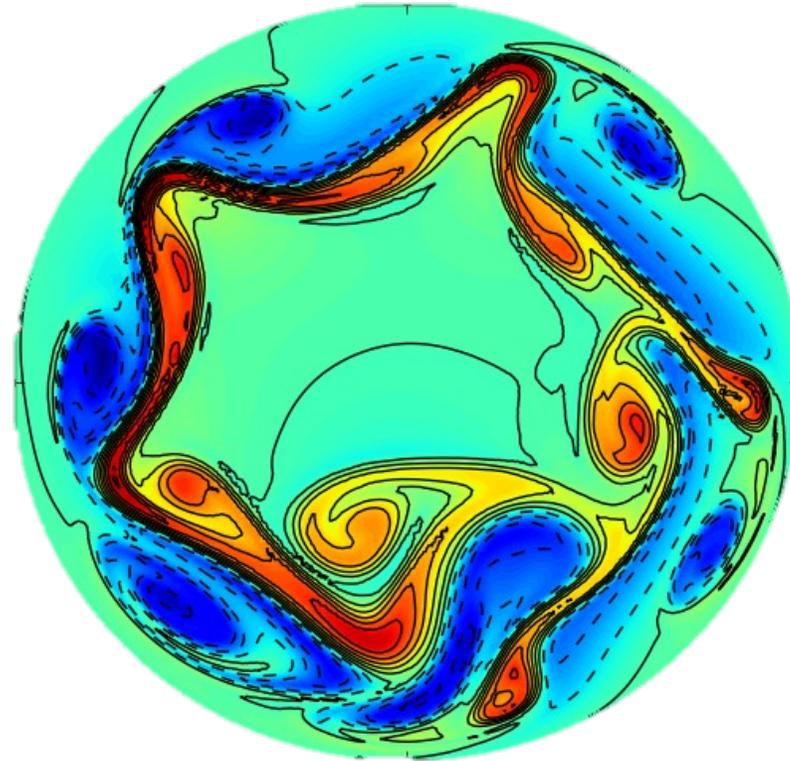


Time dependent problems

Shallow waters on a sphere

Nonlinear instability on a sphere: non-affine mesh (day 7)

View of the North pole



Time dependent problems

Shallow waters on a sphere

$N = 1024 \times 512 = 524\,288$

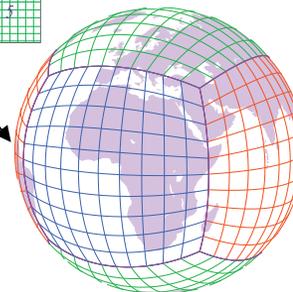
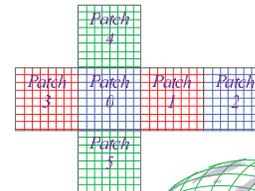
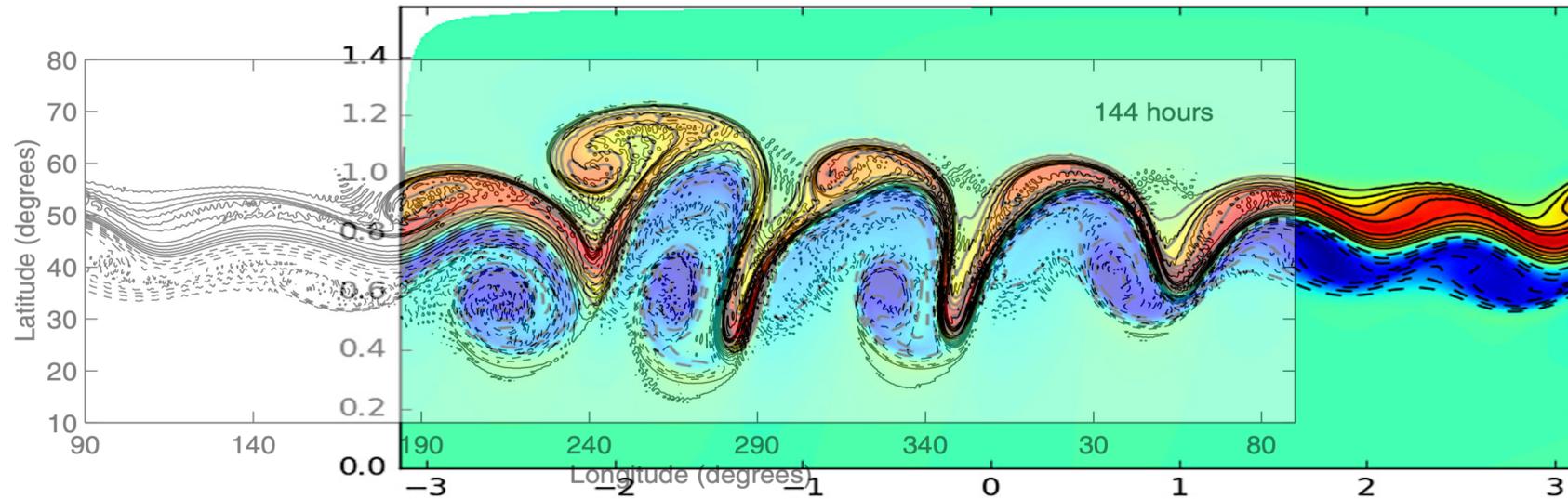
Spectral solver

$dt = 30s$

$dL = 0.35^\circ$



Simulation day 6



$N = 32 \times 32 \times 6 = 6144$

$p = 3$

$dt = 40s$

$dL = 0.9375^\circ$

Time dependent problems

MagnetoHydrodynamics: dual field formulation

$$\text{Diag} \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \\ 0 \\ \mathbf{0} \\ \varsigma \mathbf{I} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \partial_t \mathbf{u} \\ \partial_t \boldsymbol{\omega} \\ \partial_t P \\ \partial_t \mathbf{E}_l \\ \partial_t \mathbf{B} \\ \partial_t \mathbf{j} \end{bmatrix} = \begin{bmatrix} \begin{array}{ccc|ccc} -\boldsymbol{\omega} \times & -R_f^{-1} \nabla \times & -\nabla & 0 & 0 & -\varsigma \mathbf{B} \times \\ R_f^{-1} \nabla \times & -R_f^{-1} \mathbf{I} & 0 & 0 & 0 & 0 \\ -\nabla \cdot & 0 & 0 & 0 & 0 & 0 \end{array} \\ \hline \begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & \nabla \times & -\mathbf{I} \\ 0 & 0 & 0 & -\nabla \times & 0 & 0 \\ -\varsigma \mathbf{B} \times & 0 & 0 & \mathbf{I} & 0 & -\varsigma R_m^{-1} \mathbf{I} \end{array} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\omega} \\ P \\ \mathbf{E}_l \\ \mathbf{B} \\ \mathbf{j} \end{bmatrix}$$

Fluid: Navier-Stokes Electromagnetic-fluid coupling

Conserved quantities

$$\mathcal{E} := \mathcal{K} + \mathcal{M} := \frac{1}{2} \langle \mathbf{u}, \mathbf{u} \rangle_{\Omega} + \frac{\varsigma}{2} \langle \mathbf{B}, \mathbf{B} \rangle_{\Omega}$$

$$\mathcal{H}_c := \langle \mathbf{B}, \mathbf{u} \rangle_{\Omega}$$

$$\mathcal{H}_m := \langle \mathbf{B}, \mathbf{A} \rangle_{\Omega}$$

Electromagnetics: Maxwell Fluid-electromagnetic coupling

\mathbf{u} : Velocity
 $\boldsymbol{\omega}$: Vorticity
 P : Total pressure ($p + \frac{1}{2} \mathbf{u} \cdot \mathbf{u}$)
 \mathbf{E}_l : Electric field ($\varsigma \mathbf{E}$)
 \mathbf{B} : Magnetic flux density
 \mathbf{j} : Current density

R_f : Fluid Reynolds number
 R_m : Magnetic Reynolds number
 ς : Coupling number

Dissipative Hamiltonian form

$$\mathcal{G} \partial_t \mathbf{x} = (\mathcal{J} + \mathcal{R}) \mathbf{x}$$

Breaks down in the ideal case

$$R_f \rightarrow \infty$$

Time dependent problems

MagnetoHydrodynamics: dual field formulation

Primal formulation

Given $\tilde{\omega}, \tilde{\mathbf{B}} \in H(\text{curl}, \Omega)$, find $\mathbf{u}, \mathbf{E}, \mathbf{j} \in H(\text{curl}, \Omega)$, $\omega, \mathbf{B} \in H(\text{div}, \Omega)$, and $P \in H^1(\Omega)$ such that:

$$\left\{ \begin{array}{ll} \left\langle \frac{\partial \mathbf{u}}{\partial t}, \boldsymbol{\nu} \right\rangle_{\Omega} = -\langle \tilde{\omega} \times \mathbf{u}, \boldsymbol{\nu} \rangle_{\Omega} - \langle \nabla P, \boldsymbol{\nu} \rangle_{\Omega} - \mathbf{R}_f^{-1} \langle \omega, \nabla \times \boldsymbol{\nu} \rangle_{\Omega} - \varsigma \langle \tilde{\mathbf{B}} \times \mathbf{j}, \boldsymbol{\nu} \rangle_{\Omega}, & \forall \boldsymbol{\nu} \in H(\text{curl}, \Omega), \\ \langle \omega, \boldsymbol{\beta} \rangle_{\Omega} - \langle \nabla \times \mathbf{u}, \boldsymbol{\beta} \rangle_{\Omega} = 0, & \forall \boldsymbol{\beta} \in H(\text{div}, \Omega), \\ \langle \mathbf{u}, \nabla \phi \rangle_{\Omega} = 0, & \forall \phi \in H^1(\Omega), \end{array} \right.$$

$$\left\{ \begin{array}{ll} \left\langle \frac{\partial \mathbf{B}}{\partial t}, \boldsymbol{\beta} \right\rangle_{\Omega} + \langle \nabla \times \mathbf{E}, \boldsymbol{\beta} \rangle_{\Omega} = 0, & \forall \boldsymbol{\beta} \in H(\text{div}, \Omega), \\ \langle \mathbf{j}, \boldsymbol{\nu} \rangle_{\Omega} - \langle \mathbf{B}, \nabla \times \boldsymbol{\nu} \rangle_{\Omega} = 0, & \forall \boldsymbol{\nu} \in H(\text{curl}, \Omega), \\ \mathbf{R}_m^{-1} \langle \mathbf{j}, \boldsymbol{\nu} \rangle_{\Omega} - \langle \mathbf{E}, \boldsymbol{\nu} \rangle_{\Omega} + \langle \tilde{\mathbf{B}} \times \mathbf{u}, \boldsymbol{\nu} \rangle_{\Omega} = 0, & \forall \boldsymbol{\nu} \in H(\text{curl}, \Omega). \end{array} \right.$$

 ω, \mathbf{B}

Dual formulation

Given $\omega, \mathbf{B} \in H(\text{div}, \Omega)$, find $\tilde{\mathbf{u}}, \tilde{\mathbf{E}}, \tilde{\mathbf{j}} \in H(\text{div}, \Omega)$, $\tilde{\omega}, \tilde{\mathbf{B}} \in H(\text{curl}, \Omega)$, and $\tilde{P} \in L^2(\Omega)$ such that:

$$\left\{ \begin{array}{ll} \left\langle \frac{\partial \tilde{\mathbf{u}}}{\partial t}, \boldsymbol{\beta} \right\rangle_{\Omega} = -\langle \omega \times \tilde{\mathbf{u}}, \boldsymbol{\beta} \rangle_{\Omega} + \langle \tilde{P}, \nabla \cdot \boldsymbol{\beta} \rangle_{\Omega} - \mathbf{R}_f^{-1} \langle \nabla \times \tilde{\omega}, \boldsymbol{\beta} \rangle_{\Omega} - \varsigma \langle \mathbf{B} \times \tilde{\mathbf{j}}, \boldsymbol{\beta} \rangle_{\Omega}, & \forall \boldsymbol{\beta} \in H(\text{div}, \Omega), \\ \langle \tilde{\omega}, \boldsymbol{\nu} \rangle_{\Omega} - \langle \tilde{\mathbf{u}}, \nabla \times \boldsymbol{\nu} \rangle_{\Omega} = 0, & \forall \boldsymbol{\nu} \in H(\text{curl}, \Omega), \\ \langle \nabla \cdot \tilde{\mathbf{u}}, \sigma \rangle_{\Omega} = 0, & \forall \sigma \in L^2(\Omega), \end{array} \right.$$

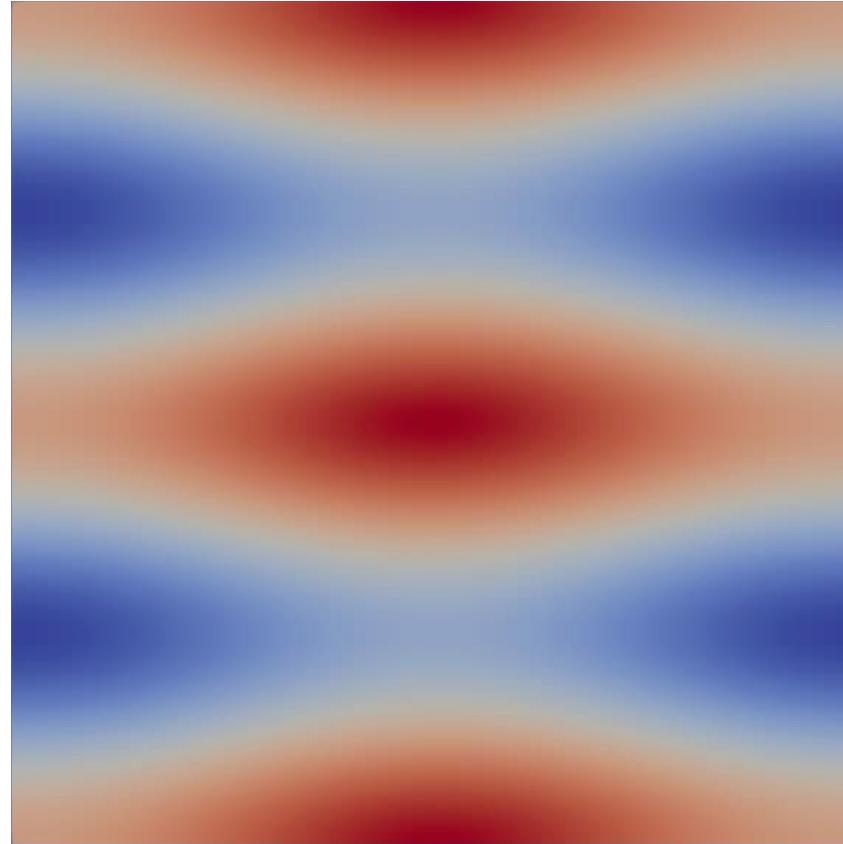
$$\left\{ \begin{array}{ll} \left\langle \frac{\partial \tilde{\mathbf{B}}}{\partial t}, \boldsymbol{\nu} \right\rangle_{\Omega} + \langle \tilde{\mathbf{E}}, \nabla \times \boldsymbol{\nu} \rangle_{\Omega} = 0, & \forall \boldsymbol{\nu} \in H(\text{curl}, \Omega), \\ \langle \tilde{\mathbf{j}}, \boldsymbol{\beta} \rangle_{\Omega} - \langle \nabla \times \tilde{\mathbf{B}}, \boldsymbol{\beta} \rangle_{\Omega} = 0, & \forall \boldsymbol{\beta} \in H(\text{div}, \Omega), \\ \mathbf{R}_m^{-1} \langle \tilde{\mathbf{j}}, \boldsymbol{\beta} \rangle_{\Omega} - \langle \tilde{\mathbf{E}}, \boldsymbol{\beta} \rangle_{\Omega} + \langle \mathbf{B} \times \tilde{\mathbf{u}}, \boldsymbol{\beta} \rangle_{\Omega} = 0, & \forall \boldsymbol{\beta} \in H(\text{div}, \Omega). \end{array} \right.$$

 $\tilde{\omega}, \tilde{\mathbf{B}}$

Time dependent problems

(Brugnoli, AP, Toshniwal, Zhang)

MagnetoHydrodynamics: Orszag-Tang vortex test



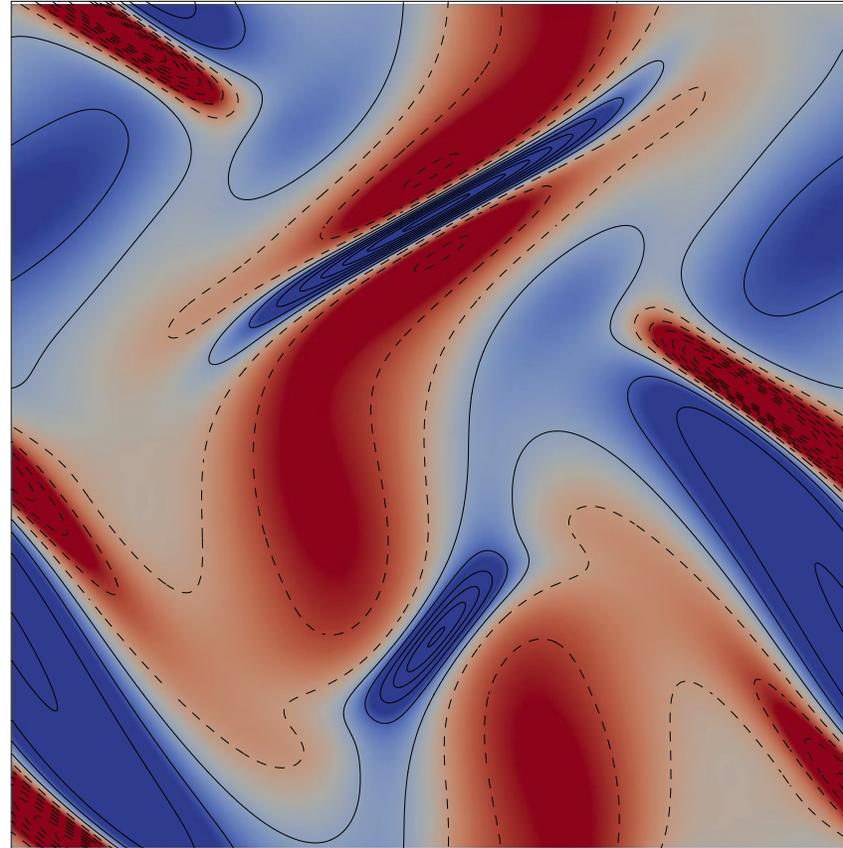
$$t \in (0, 0.7]$$

256 x 256 elements
 $p = 1$ de Rham complex
 $dt = 0.005$

Time dependent problems

(Brugnoli, AP, Toshniwal, Zhang)

MagnetoHydrodynamics: Orszag-Tang vortex test



$t = 0.51$

Reference contours:
Cordoba, Marliani (2000)

Projection method
Second order upwinding
Adaptive refinement

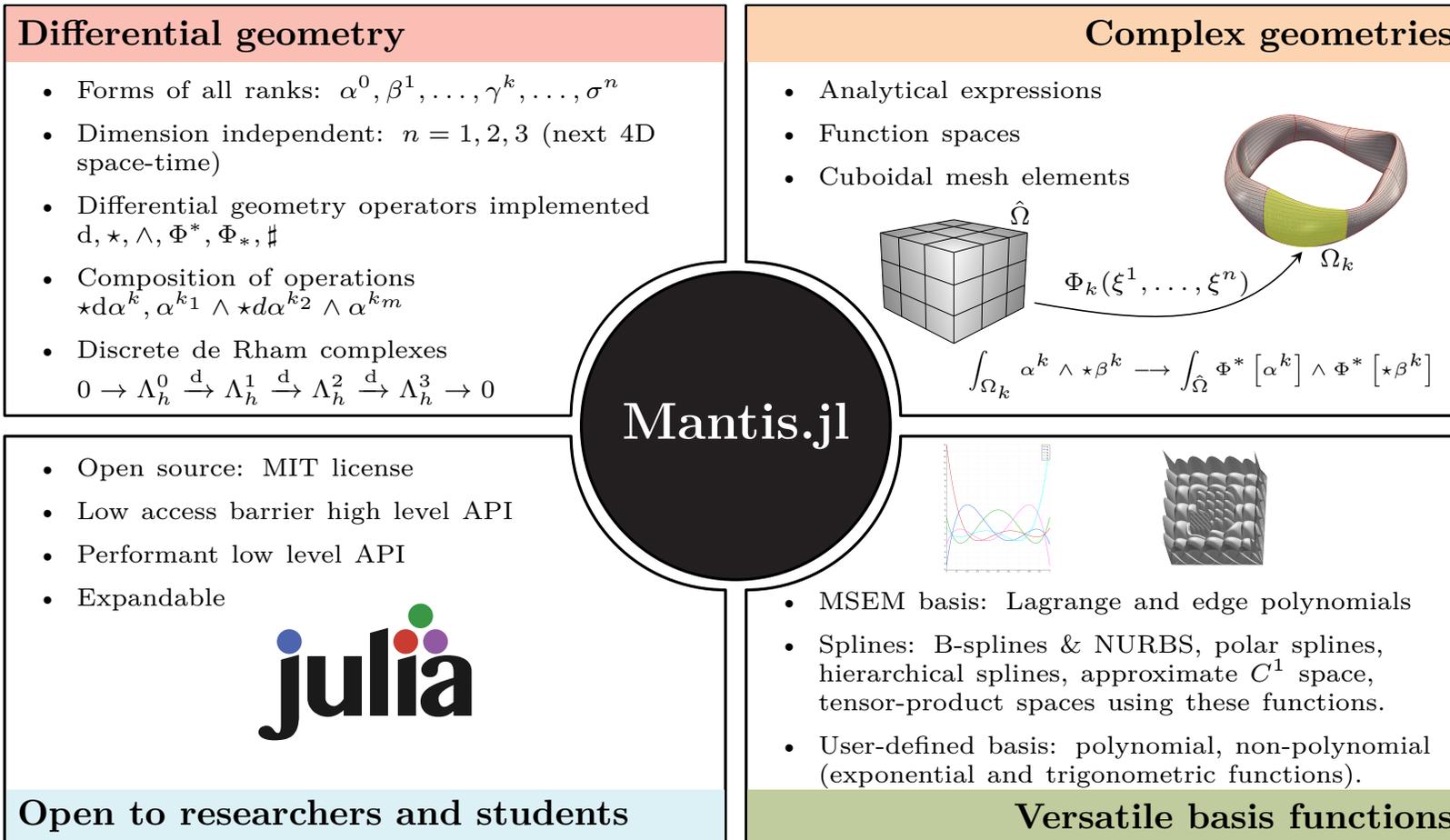
Start: 1024x1024

End (t = 0.75): 32768 x 32768 (finest)

Thank you

Mantis.jl

A differential geometry structure-preserving FEM library in Julia



A differential geometry structure-preserving FEM library in Julia

Mantis.jl represents manifolds through mappings

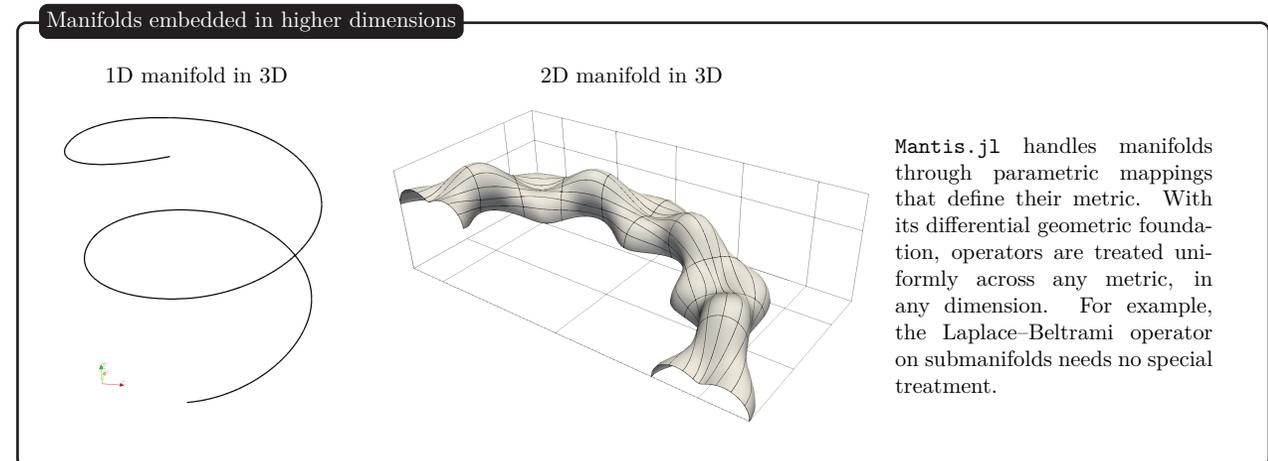
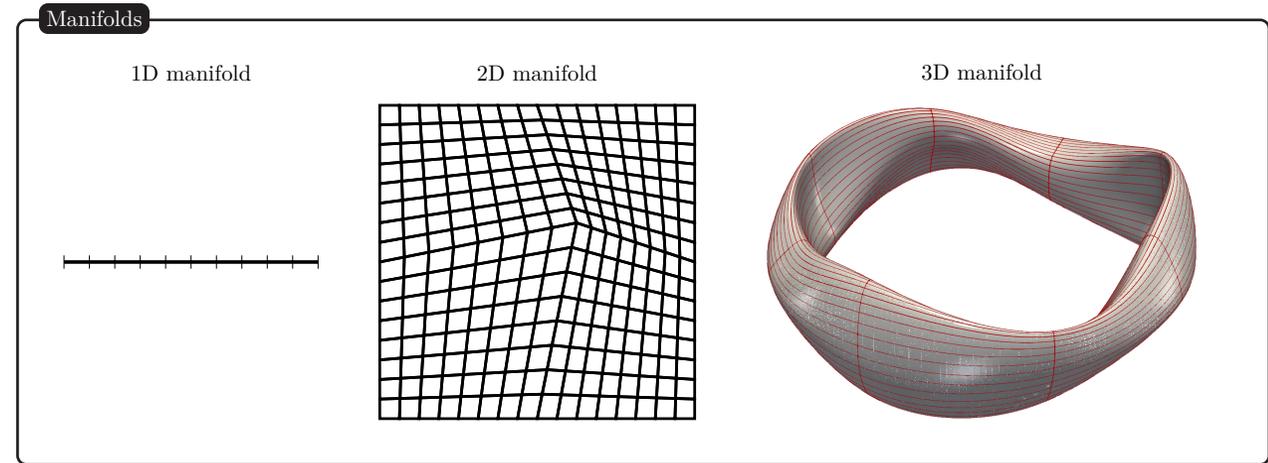
$$\Phi : \hat{\Omega} \subset \mathbb{R}^n \mapsto \mathbb{R}^m, \quad n \leq m,$$

where Φ is injective, continuous, and element-wise differentiable.

Departing from traditional FEM libraries that rely on the Jacobian, J , Mantis.jl adopts a differential geometric approach: metric tensor $g = J^\top J$, pullbacks Φ^* , and pushforward Φ_* .

This enables PDEs to be pulled-back and solved intrinsically on the parametric manifold with the metric induced by the mapping and then pushed-forward.

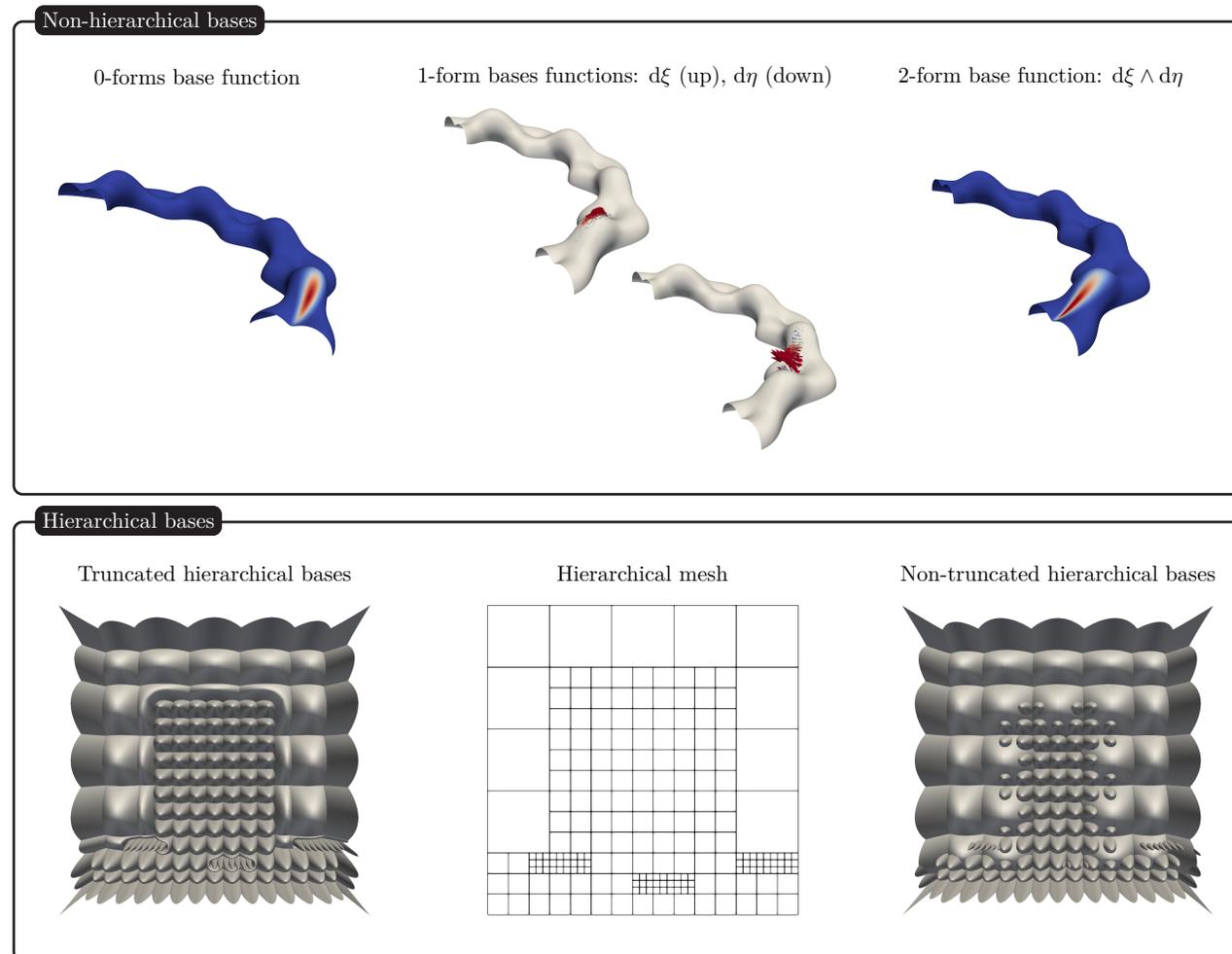
- **Dimension-agnostic design:** the same machinery works for curves, surfaces, and volumes in arbitrary embedding dimensions.
- **Exact geometric compatibility:** using pullbacks/pushforwards ensures transformations of fields respect their geometric nature (scalar, vector, k-form).
- **Intrinsic formulation:** geometry is expressed through the metric tensor, so PDEs are posed intrinsically without dependence on embedding choices.
- **High-order geometry:** by allowing mappings as FEM fields of analytical expressions, Mantis.jl naturally supports high-order curved geometries.
- **Structure-preserving:** the metric-based formulation aligns with FEEC and guarantees consistency of operators across geometries.



A differential geometry structure-preserving FEM library in Julia

Computations in `Mantis.jl` can be performed using a wide variety of discrete differential form spaces. These include standard finite elements differential forms with the lowest regularity, but also several generalizations. The underlying meshes are assumed to be composed of cuboidal elements, with support for simplicial elements planned for the future.

- **Univariate discrete differential forms:**
 - mimetic finite element spaces
 - polynomial B-splines
 - multi-degree B-splines
 - generalized B-splines (e.g., trigonometric, exponential)
- **Multivariate extensions through tensor products:** All form spaces can be combined with each other through tensor products to create new form spaces in higher dimensions.
- **Unstructured multivariate extensions:** `Mantis.jl` allows users to implement their own form spaces through an extraction-based interface. This enables simulations on complex geometries. Polar and toroidal splines for axisymmetric domains, and approximate $C1$ splines are included in the library.
- **Hierarchical bases:** Form spaces in `Mantis.jl` can be locally refined using a hierarchical approach, thus enabling adaptive simulations. Both truncated and non-truncated hierarchical bases are supported.



Define the continuous weak problem

$$\begin{cases} d^*d\phi^0 = -f^0 & \text{on } \Omega \\ \text{tr}(\phi^0) = 0 & \text{on } \partial\Omega \end{cases}$$



Given $f^0 \in L^2\Lambda^0(\Omega)$, find $\phi^0 \in \Lambda_h^0(\Omega)$ such that

$$\int_{\Omega} d\psi^0 \wedge \star d\phi^0 = \int_{\Omega} \psi^0 \wedge \star f^0 \quad \forall \psi^0 \in \Lambda_{h,0}^0(\Omega)$$

Define the continuous weak problem

$$\begin{cases} d^*d\phi^0 = -f^0 & \text{on } \Omega \\ \text{tr}(\phi^0) = 0 & \text{on } \partial\Omega \end{cases}$$



Given $f^0 \in L^2\Lambda^0(\Omega)$, find $\phi^0 \in \Lambda_h^0(\Omega)$ such that

$$\int_{\Omega} d\psi^0 \wedge \star d\phi^0 = \int_{\Omega} \psi^0 \wedge \star f^0 \quad \forall \psi^0 \in \Lambda_{h,0}^0(\Omega)$$


Define the geometry

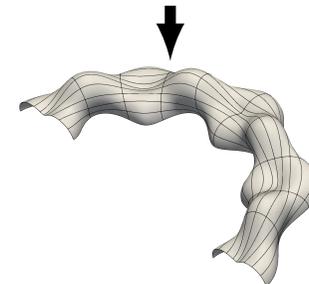
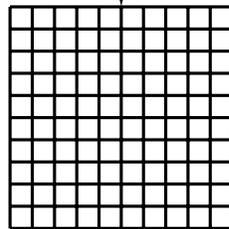
```

1 import Mantis
2
3 starting_point = (0.0, 0.0)
4 box_size = (0.5, 0.5)
5 num_elements = (10, 10)
6 cartesian_geo = Mantis.Geometry.create_cartesian_box(
7     starting_point, box_size, num_elements
8 )
9 geometry = cartesian_geo
    
```

Map
Geometry

```

1 function mapping(x::AbstractVector)
2     # Define the mapping
3     return [Φ[k](x[1], x[2]) for k in 1:3]
4 end
5 function dmapping(xi::AbstractVector)
6     # Define the Jacobian of the mapping: J
7     return [J[k,j][x[1], x[2]] for k in 1:3, j in 1:2]
8 end
9 # Define the mapped geometry
10 dimension = (2, 3) # 2-manifold in R^3
11 curved_mapping = Mantis.Geometry.Mapping(dimension,
12     mapping, dmapping)
13 geometry = Geometry.MappedGeometry(cartesian_geo,
14     curved_mapping)
    
```



Mantis.jl generates geometries with identical interfaces. For example, for the solver, a 2D Cartesian geometry is indistinguishable from a geometry representing a curved 2D manifold embedded in \mathbb{R}^3 . The same applies for a 3D manifold (geometry).

Define form spaces

```
1 # Create a FEM function space for the 0-forms
2 p = (3, 3) # polynomial degree 3 in both directions
3 k = (2, 2) # maximal smoothness in both directions
4 B = Mantis.FunctionSpaces.create_bspline_space(
5     starting_point,
6     box_size,
7     num_elements,
8     p,
9     k,
10 )
11 # Create the 0-form function space combining
12 #     - FEM space
13 #     - Geometry
14  $\Lambda_0$  = Mantis.Forms.FormSpace(0, geometry, (B,), " $\Lambda_0$ ")
```

Hierarchical
spaces



de Rham
complex

```
1 # Hierarchical parameters.
2 truncate = true # true = THB, false = HB
3 num_sub = (2, 2) # Subdivisions per dim per step.
4
5 H = Mantis.Forms.create_hierarchical_de_rham_complex(
6     geometry, p, k, num_sub, truncate)
7  $\Lambda_0$  = H[0] # Use 0-form space
```

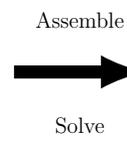
Mantis.jl can generate individual form spaces from FEM spaces, but it is also possible to generate all spaces associated to a discrete de Rham complex. As for geometries, form spaces share the same interface for the assembly step, allowing for easy interchange of spaces (e.g., non-hierarchical and hierarchical).

Weak form, assembly, and solution

```

1   $\Lambda$  = Mantis.Forms.wedge
2  * = Mantis.Forms.hodge
3  function zero_form_hodge_laplacian(inputs, element_id)
4      qnodes = Mantis.Quadrature.get_nodes(inputs.quad_rule)
5      qweights = Mantis.Quadrature.get_weights(
6          inputs.quad_rule)
7
8       $\psi^0$  = inputs.space_test
9       $d\psi^0$  = Mantis.Forms.exterior_derivative( $\psi^0$ )
10      $d\phi^0$  = Mantis.Forms.exterior_derivative(
11         inputs.space_trial)
12
13     wedgeA =  $d\psi^0 \wedge *d\phi^0$ 
14     wedge_A, Ainds = Mantis.Forms.evaluate(wedgeA,
15         element_id, qnodes)
16     # - Computation of row and column indices redacted. -
17     Av = vec(sum(qweights .* wedge_A[1]; dims=1))
18
19     wedgeb =  $\psi^0 \wedge$  inputs.forcing
20     wedge_b, br = Mantis.Forms.evaluate(wedgeb,
21         element_id, qnodes)
22     bv = vec(sum(qweights .* wedge_b[1]; dims=1))
23
24     return (Ar, Ac, Av), (br[1], bv)
25 end

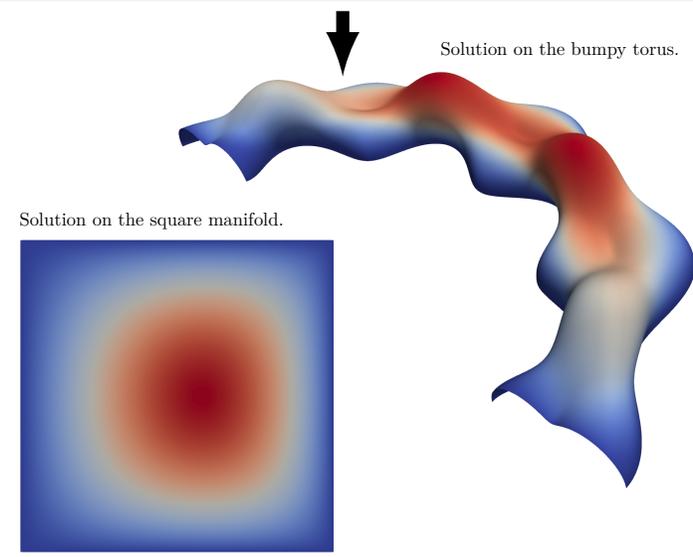
```



```

1  # Set essential boundary conditions
2  bc_dirichlet = Dict{Mantis.Assemblers.get_trace_dofs(B)
3      => 0.0}
4
5  # Assemble system
6  lhs, rhs = Mantis.Assemblers.assemble(
7      zero_form_hodge_laplacian, wfi, bc_dirichlet)
8
9  # Solve
10  $\phi^0$  = Mantis.Forms.FormField( $\Lambda^0$ , " $\phi^0$ ")
11  $\phi^0$ .coefficients .= lhs \ rhs

```



Thank you