Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Home Affairs FDHA
Federal Office of Meteorology and Climatology MeteoSwiss

Dusk & Dawn - Introduction and Overview

ESIWACE2 Summer School, 23rd August 2021

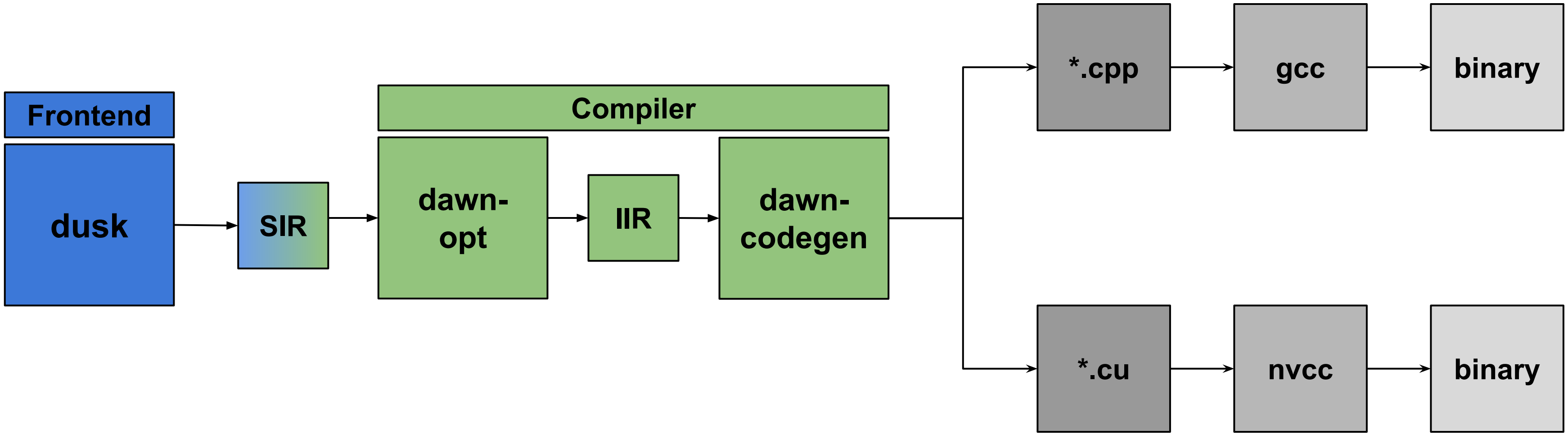
Speaker: Christoph Müller, MeteoSwiss

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Overview

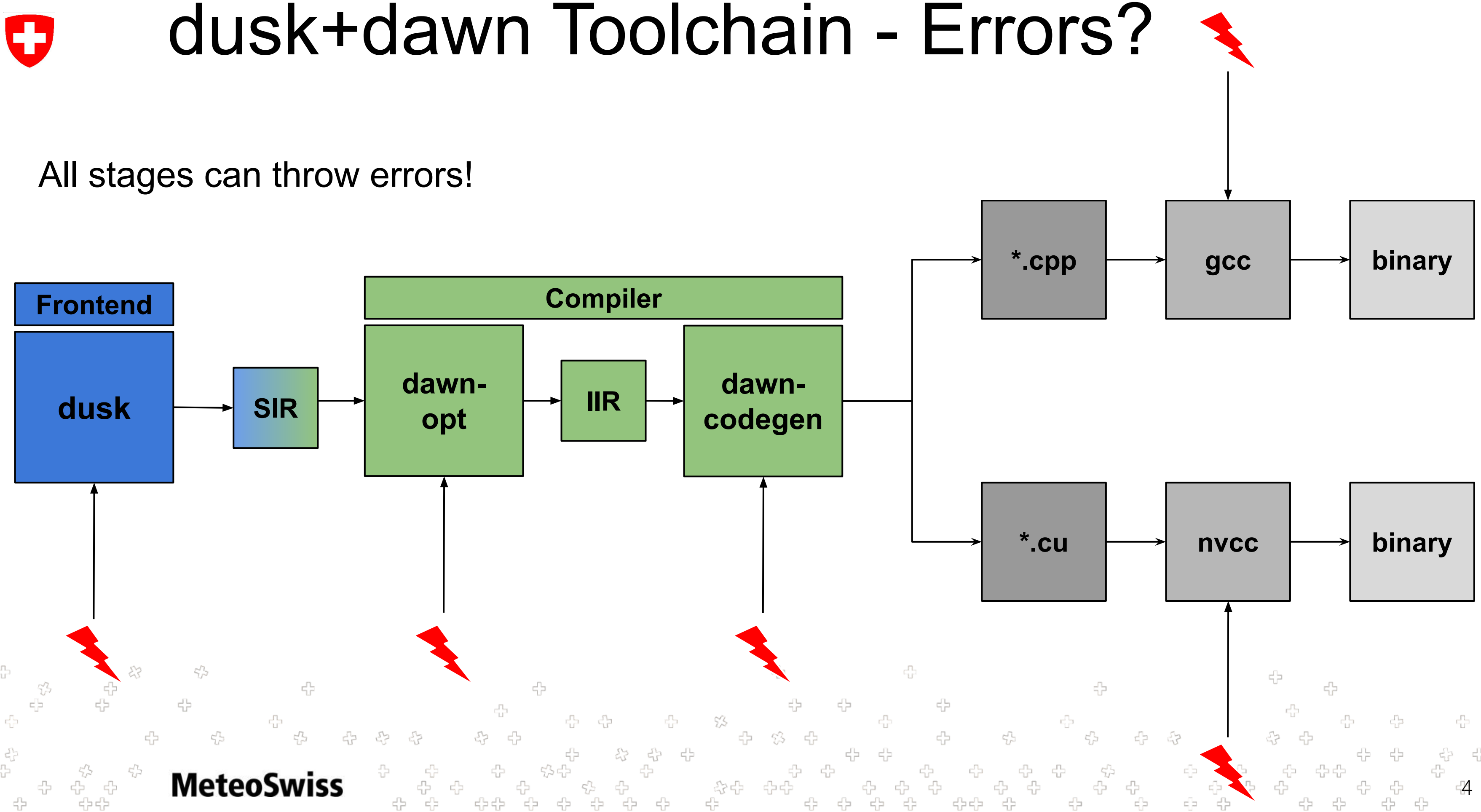
- Introduction to dusk+dawn toolchain
- Point-wise stencils
- Simple reductions
  - Gradient, Divergence, Curl
  - (Laplacian)

dusk+dawn Toolchain



dusk+dawn Toolchain - Errors?

- All stages can throw errors!



MeteoSwiss

Demonstration

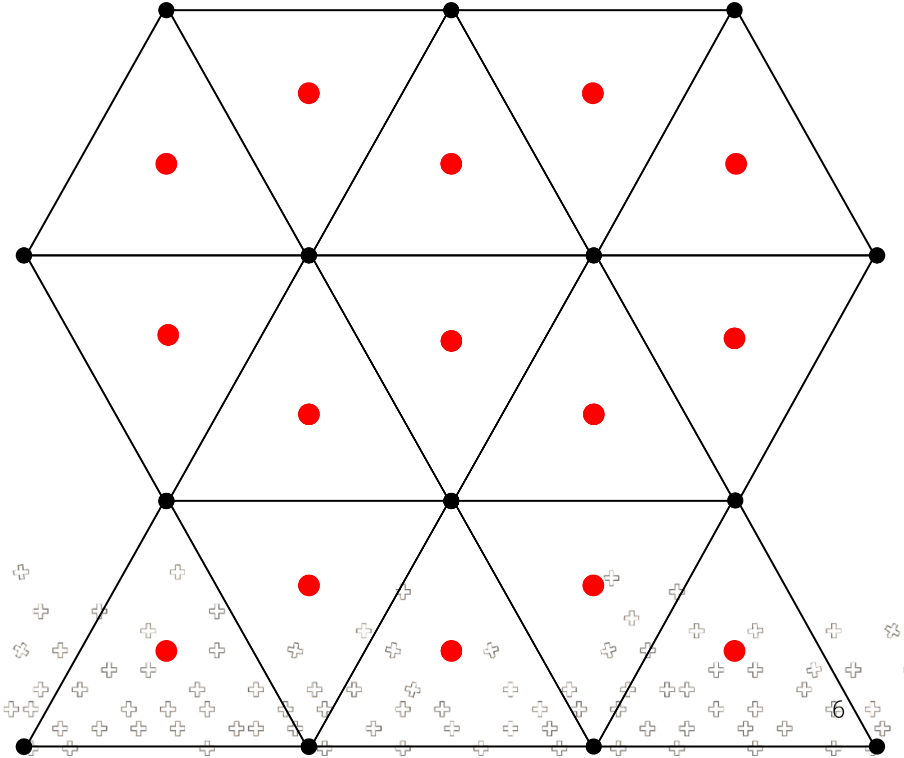
- ‘intro’ folder

Point-wise stencils

- Simple point-wise stencil

```
@stencil
def point(in: Field[Cell],
          out: Field[Cell]):
    with levels_downward:
        out = in + 1
```

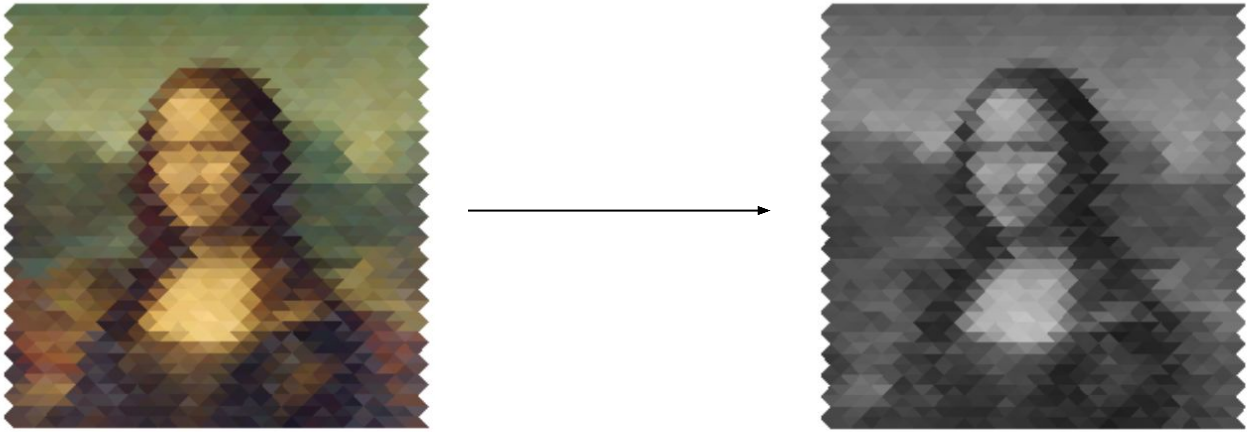
- No neighborhood accesses, no reductions



MeteoSwiss

Image processing

- Can do image processing using stencil compiler



Demonstration

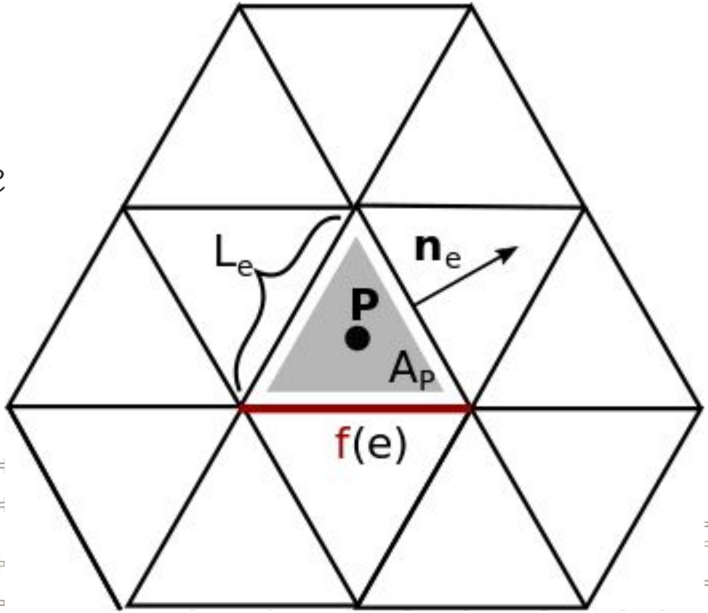
- ‘hello_world’ folder

dusk & dawn - Differential Operators Exercise

The Gradient

- The Gradient is an operator in a scalar field f that returns a vector ∇f
- It points into the direction of the steepest slope of the scalar field, and its magnitude is a measure on how steep that slope is
- On a (triangular) FVM mesh, the Gradient can be approximated using:

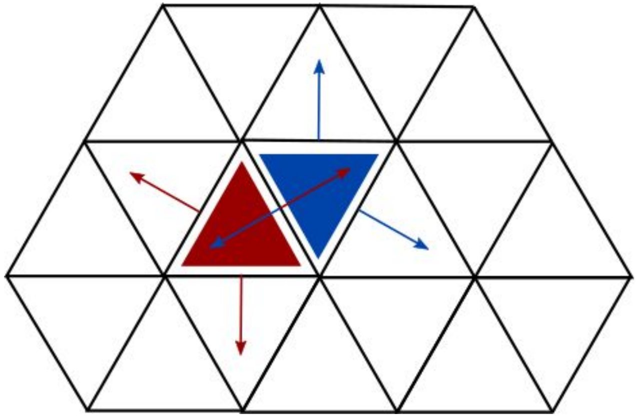
$$\langle \nabla f(\mathbf{P}) \rangle_{FVM} = \frac{1}{A_P} \sum_{e=1}^{3} f(e) L_e \mathbf{n}_e$$



- Note that this equation only holds if the normals $\mathbf{n}_f$ point outside the triangle with centroid P

A Note on Geometrical Factors

- The equation for the gradient on the last slide holds only if all normals point outside
- However, one clearly can't arrange a mesh such that all normals point outside

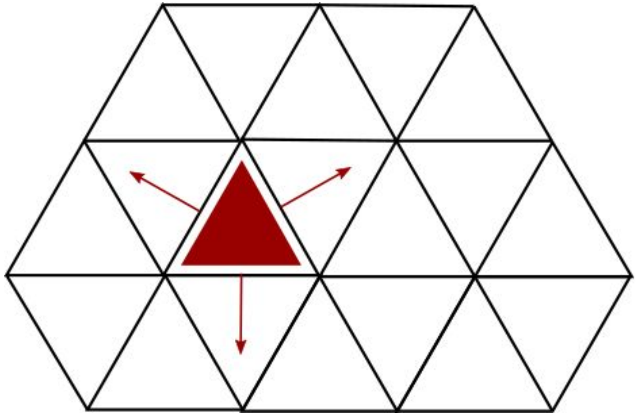


- To check, for each normal, in which side it points during the computation of the gradient is not efficient (involves the evaluation of a dot product)
- For this exercise, geometric factors (`edge orientation`) have been prepared for you to ensure just that. Be aware that you **need to multiply by the edge orientation every time you access a normal**! (e.g. nx*`edge_orientation`)

A Note on Geometrical Factors

- The equation for the gradient on the last slide holds only if all normals point outside
- However, one clearly can't arrange a mesh such that all normals point outside

edge_orientation(**cell_a**,:) = [1, 1, 1]

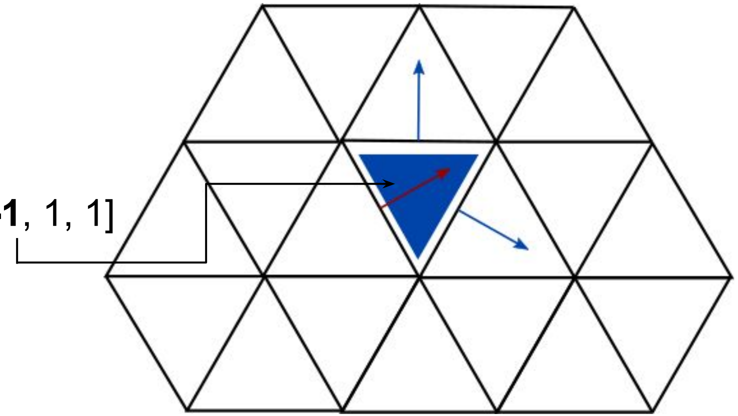


- To check, for each normal, in which side it points during the computation of the gradient is not efficient (involves the evaluation of a dot product)
- Usually, this is solved by pre-computing a geometrical factor. In this case three signs for each triangle, that indicate whether the normal is flipped during the summation
- This is a prime example of a sparse dimension, in this case `Field[Cell > Edge]`

A Note on Geometrical Factors

- The equation for the gradient on the last slide holds only if all normals point outside
- However, one clearly can't arrange a mesh such that all normals point outside

edge_orientation(**cell_b**,:) = [**-1**, 1, 1]

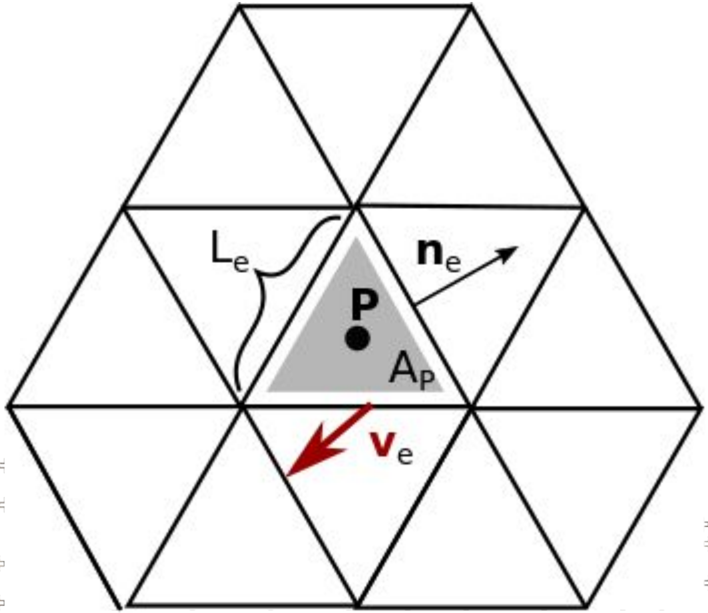


- To check, for each normal, in which side it points during the computation of the gradient is not efficient (involves the evaluation of a dot product)
- Usually, this is solved by pre-computing a geometrical factor. In this case three signs for each triangle, that indicate whether the normal is flipped during the summation
- This is a prime example of a sparse dimension, in this case `Field[Cell > Edge]`

The Divergence

- The Divergence is an operator on a vector field **v** = [u, v] that returns a scalar **∇ ▪v**
- The scalar indicates the intensity of the source (extension / contraction) of a vector field (e.g. the divergence of heating, and hence expanding, air would be positive)

$$\langle \nabla \cdot \mathbf{v}(\mathbf{P}) \rangle_{FVM} = \frac{1}{A_P} \sum_{e=1}^{3} (\mathbf{v}_e \cdot \mathbf{n}_e)\, L_e$$

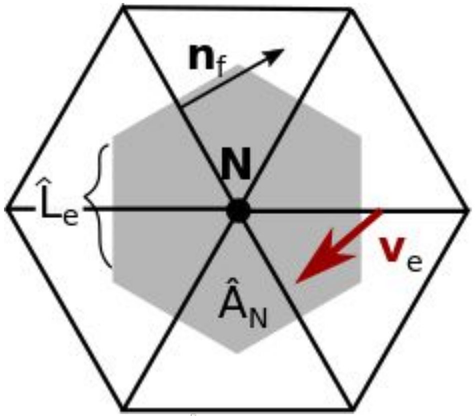


- Note that this equation only holds if the normals $\mathbf{n}_f$ point outside the triangle with centroid P

The Curl

- The Curl is an operator on a vector field **v** = [u, v, w] that returns a vector **∇×v**
- Strictly speaking, the curl is only defined in a 3d field, but emulating it in 2D using **v** = [u, v, 0] will reveal that only ever the third component of **∇×v** takes a value (other than zero).
- So **for our intents** and purposes, the **curl is scalar (!)** and is a measure for the rotation of the vector field **v**

$$\langle \nabla \times \mathbf{v}(\mathbf{N}) \rangle_{FVM} = \frac{1}{\hat{A}_N} \sum_{e=1}^{6} (\mathbf{n}_e \cdot \mathbf{v}_e)\, \hat{L}_e$$



- Note that this equation only holds if the normals $\mathbf{n}_e$ form a left handed coordinate system with vectors $\mathbf{e}_i$ (i.e. unit vectors along the edges) and a vector **z** pointing into the plane

Exercise

- The exercise consists of implementing these three differential operators
- The function for the **gradient** is this wave function**:**

Function:

$$f(x,y) = sin(x)cos(x)$$

Gradient:

$$\nabla f(x,y) = \begin{bmatrix} \cos(x)\sin(y) \\ sin(x)cos(x) \end{bmatrix}$$

Exercise

- The exercise consists of implementing these three differential operators
- The vector field for the **divergence** and **curl** are the spherical harmonics:

Field:

$$\begin{cases} u(x,y) := & \frac{1}{4}\sqrt{\frac{105}{2\pi}} \cos 2x \, \cos^2 y \, \sin y \quad , \\ v(x,y) := & \frac{1}{2}\sqrt{\frac{15}{2\pi}} \cos x \, \cos y \, \sin y \quad . \end{cases}$$

Divergence:

$$\nabla \cdot \mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = \frac{-1}{2\sqrt{2\pi}} \left(\sqrt{105} \, \sin 2x \, \cos^2 y \, \sin y \, , + \sqrt{15} \, \cos x \, \cos 2y \right)$$

Curl:

$$\nabla \times \mathbf{v} = \left(\frac{1}{4}\sqrt{\frac{105}{2\pi}} \right) \cos(2x)\cos(y)(\cos(y)\cos(y) - 2\sin(y)\sin(y)) - \left(\frac{1}{2}\sqrt{\frac{15}{2\pi}} \right) \cos(y)\sin(x)\sin(y)$$

Variable Reference

An overview over all variables is given below. The ones in bold are the ones you're supposed to be writing to. The others can be treated as read only.

f: Field[Edge]	wave function on edges
f_x: Field[Cell], **f_y**: Field[Cell]	gradient of wave function on cells
nx: Field[Edge], ny: Field[Edge]	cell normals on edges (x and y component)
L: Field[Edge]	edge lengths
A: Field[Cell]	cell areas
edge_orientation: Field[Cell > Edge]	sparse dimension that indicates which normals need to be flipped for gradient / div computation (+1/-1)
u: Field[Edge], v: Field[Edge]	vector field / spherical harmonics on edges
uv_div: Field[Cell]	divergence of vector field /spherical harmonics on cells
dualL: Field[Edge]	dual edge length $\hat{L}_e$ (c.f. slide 8)
dualA: Field[Vertex]	dual cell area $\hat{A}_N$ (c.f. slide 8)
uv_curl: Field[Vertex]	curl of vector field / spherical harmonics on vertices
edge_orientation: Field[Vertex > Edge]	sparse dimension that indicates which normals need to be flipped in curl computation (+1/-1)

Hints

- divergence and gradient operate on the same neighbor chain
- the computations are all quite short
  - do not overthink things
  - all three operators can be expressed in one or two lines

Exercise

- ‘differential_ops’ folder

dusk & dawn - Vector Laplacian

The Laplacian

$$\nabla^2(f(\mathbf{p})) = \nabla(\nabla f(\mathbf{p})) = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y} + \frac{\partial^2 f}{\partial^2 z}$$

- Takes a scalar function (scalar field) and returns a scalar
- Informally, the Laplacian of a function f at a point **p** measures by how much the average value of f over small spheres or balls centered at **p** deviates from f(**p**).
- Very wide spread in lots of physical equations
  - e.g. diffusion of chemical components, spread of heat in homogenous materials

The Vector Laplacian

$$\nabla^2 \mathbf{v} = \nabla (\nabla \mathbf{v}) = \nabla (\nabla \cdot \mathbf{v}) - \nabla \times (\nabla \times \mathbf{v})$$

- Takes a vector function (vector field) and returns a vector
- Also very relevant in physics, e.g. in the Navier Stokes Equation

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) = \rho \mathbf{f} - \nabla p + \mu \left(\nabla^2 \mathbf{v} \right)$$

f: body forces
p: pressure
mu: viscosity

- $\mu(\nabla^2 \mathbf{v})$ are the viscous stresses in the the fluid

The Vector Laplacian

$$\nabla^2 \mathbf{v} = \nabla(\nabla \mathbf{v}) = \nabla(\nabla \cdot \mathbf{v}) - \nabla \times (\nabla \times \mathbf{v})$$

- This form of the vector Laplacian is quite general
- There are more straightforward for Cartesian coordinates
- However, the **normal component** of the form above lends itself very well to implementation on a FVM mesh:

$$\nabla^2 (\mathbf{v} \cdot \mathbf{n}) = (\nabla \cdot \mathbf{n}) \underbrace{[\nabla \cdot \mathbf{v}]}_{\text{divergence}} - (\nabla \cdot \tau) \underbrace{[\nabla \times \mathbf{v}]}_{\text{curl}}$$

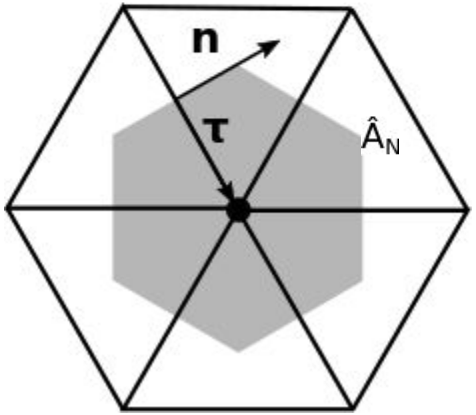
MeteoSwiss

c.f. Hui Wan - Developing and testing a hydrostatic atmospheric dynamical core on triangular grids

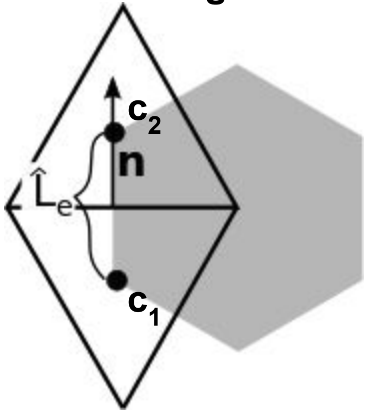
The Vector Laplacian

$$\nabla^2(\mathbf{v}\cdot\mathbf{n}) = \text{grad}_n \underbrace{[\nabla\cdot\mathbf{v}]}_{\text{divergence}} - \text{grad}_\tau \underbrace{[\nabla\times\mathbf{v}]}_{\text{curl}}$$

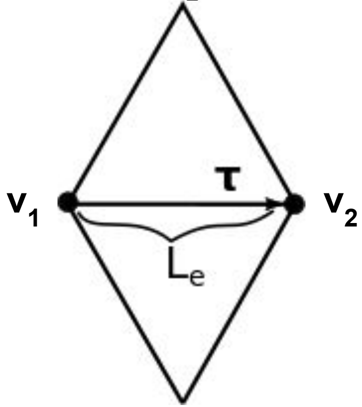
sketch triangular mesh



directional gradient n



directional gradient τ



$$\langle\nabla_n f\rangle_e = \frac{f(\mathbf{c_2}) - f(\mathbf{c_1})}{\hat{L}_e} \qquad \langle\nabla_\tau f\rangle_e = \frac{f(\mathbf{v_2}) - f(\mathbf{v_1})}{L_e}$$

MeteoSwiss

The Vector Laplacian

$$\nabla^2 (\mathbf{v} \cdot \mathbf{n}) = \text{grad}_n \underbrace{[\nabla \cdot \mathbf{v}]}_{\text{divergence}} - \text{grad}_\tau \underbrace{[\nabla \times \mathbf{v}]}_{\text{curl}}$$

- Divergence is located on cells
- Curl is located on vertices

→ Perfect fit to compute vector Laplacian of normal component of velocity on edges

Geometrical Factors

- Just like for the computation of the curl and gradient, we again need geometrical factors to compute the directional gradients
- We need to make sure that the meshing library always returns the cell neighbor $\mathbf{c}_1$ first ("in the direction of the normal"), and $\mathbf{c}_2$ second ("in the opposite direction of the normal")
- The same argument goes for the tangential gradient

→ for the gradient along the **normal** the mesh has the correct property, **no factor needed**
→ for the gradient along the **tangent** a factor called `tangent_orientation` is given

Exercise

- For the exercise, you are about to re-use the divergence and curl operators you already implemented in the previous exercise
- The vector field is again made up of the spherical harmonics:

Field:

$$\begin{cases} u(x,y) := & \frac{1}{4}\sqrt{\frac{105}{2\pi}} \cos 2x \cos^2 y \sin y \quad , \\ v(x,y) := & \frac{1}{2}\sqrt{\frac{15}{2\pi}} \cos x \cos y \sin y \quad . \end{cases}$$

Vector Laplacian:

$$\nabla^2 \mathbf{v} = \begin{bmatrix} -\left(\sqrt{\frac{105}{2\pi}}\right) \cos(2 \cdot x) \cos(y) \cos(y) \sin(y) \\ \left(-2 \cdot \sqrt{\frac{15}{2\pi}}\right) \cos(x) \sin(y) \cos(y) \end{bmatrix}$$

Variable Reference

An overview over all variables is given below. The ones in bold are the ones you're supposed to be writing to. The others can be treated as read only.

u: Field[Edge], v: Field[Edge]	vector field / spherical harmonics on edges
nx: Field[Edge], ny: Field[Edge]	cell normals on edges (x and y component)
L: Field[Edge]	edge lengths
A: Field[Cell]	cell areas
uv_div: Field[Cell]	divergence of vector field /spherical harmonics on cells
uv_curl: Field[Vertex]	curl of vector field / spherical harmonics on vertices
grad_of_curl: Field[Edge]	tangential gradient of curl
grad_of_div: Field[Edge]	normal gradient of divergence
uv_nabla_2: Field[Edge]	normal component of vector gradient of uv·n
L: Field[Edge]	edge length
dualL: Field[Edge]	dual edge length $\hat{L}_e$ (c.f. slide 6)
A: Field[Cell]	cell area
dualA: Field[Vertex]	dual cell area $\hat{A}_N$ (c.f. slide 6)

...

wiss

Variable Reference

An overview over all variables is given below. The ones in bold are the ones you're supposed to be writing to. The others can be treated as read only.

`tangent_orientation: Field[Edge]`	field indicating which tangential gradients need to be flipped
`edge_orientation_vertex: Field[Vertex>Edge]`	sparse dimension that indicates which normals need to be flipped for curl computation
`edge_orientation_cell: Field[Cell>Edge]`	sparse dimension that indicates which normals need to be flipped for gradient / div computation (+1/-1)

MeteoSwiss

Hints

- A few text hints are given on the next slide
- A skeleton of the solution is given on the last slide
- Please use it only if you're seriously stuck

Hints

- The gradients are weighted reductions
- You only need to add three additional lines to the ops you have written before

MeteoSwiss

Hints

```
with levels_upward as k:
    # compute curl (on vertices)
    uv_curl =
    # compute divergence (on cells)
    uv_div =
    # first term of of nabla2 (gradient of curl)
    grad_of_curl =
    # second term of of nabla2 (gradient of divergence)
    grad_of_div =
    # finalize nabla2 (difference between the two gradients)
    uv_nabla2 =
```

Exercise

- ‘laplacian’ folder