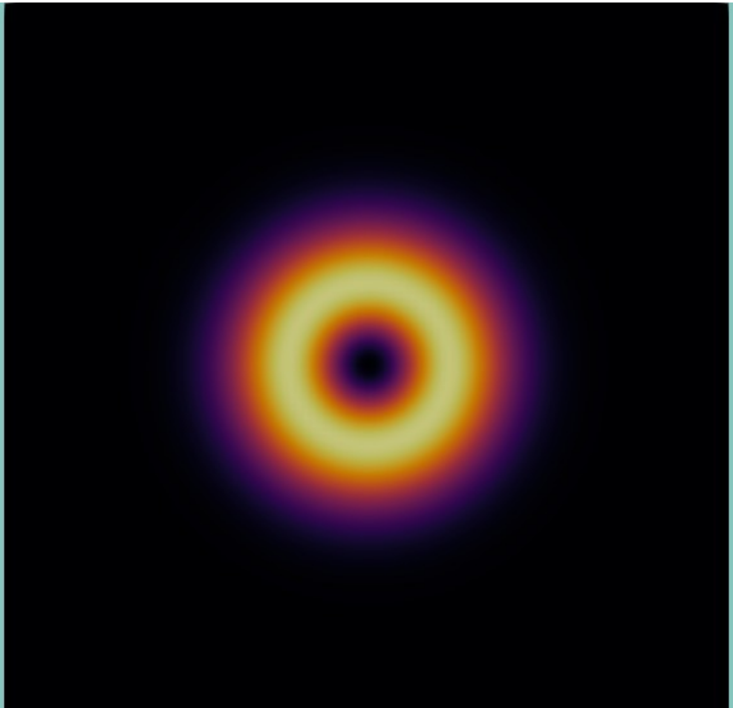
Adaptively Coupled Multiphysics Simulations with Trixi.jl

Simon Candelaresi, Michael Schlottke-Lakemper

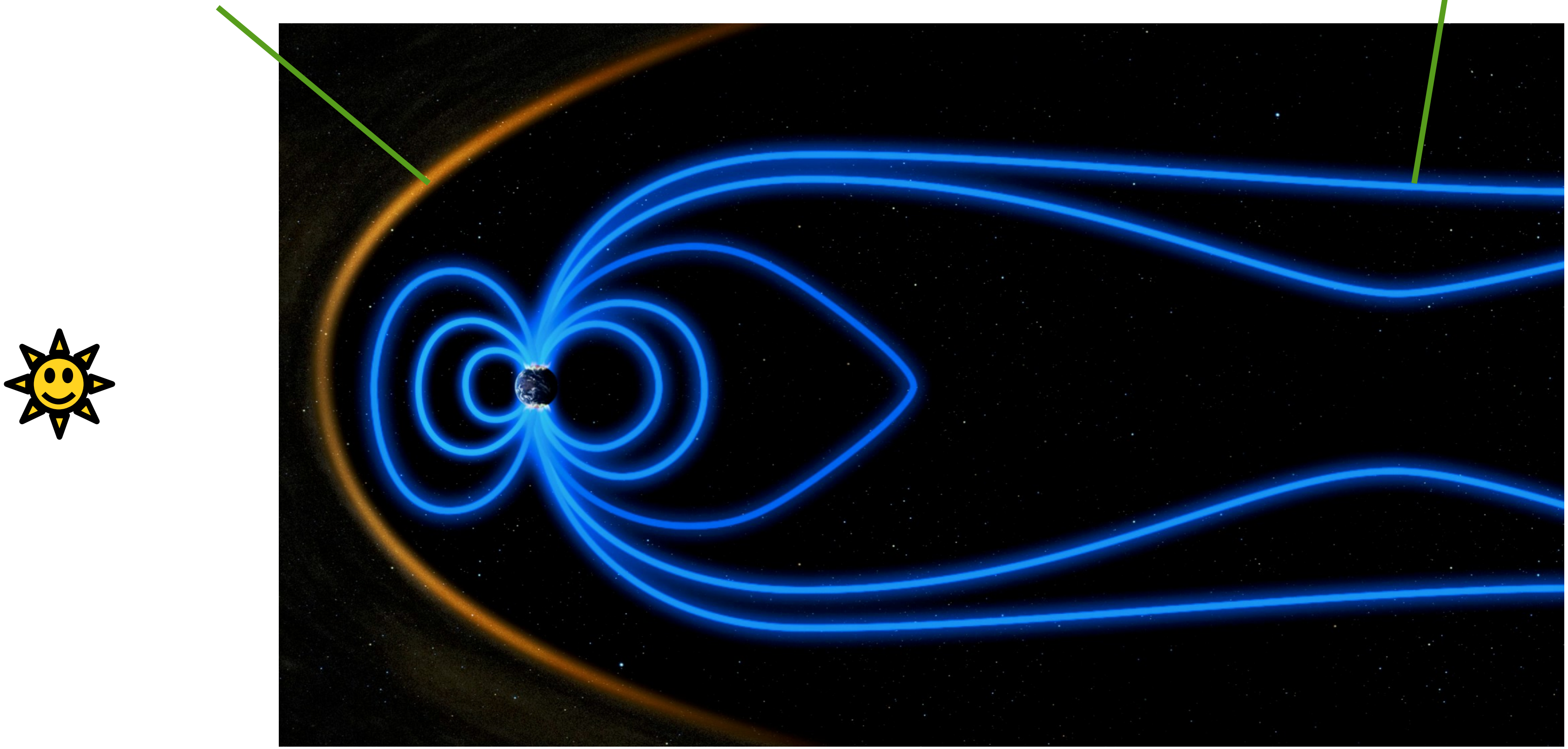


HLRS
High-Performance Computing Center Stuttgart

Universität Augsburg
University

Luca Rüedi
(zuonline.ch)

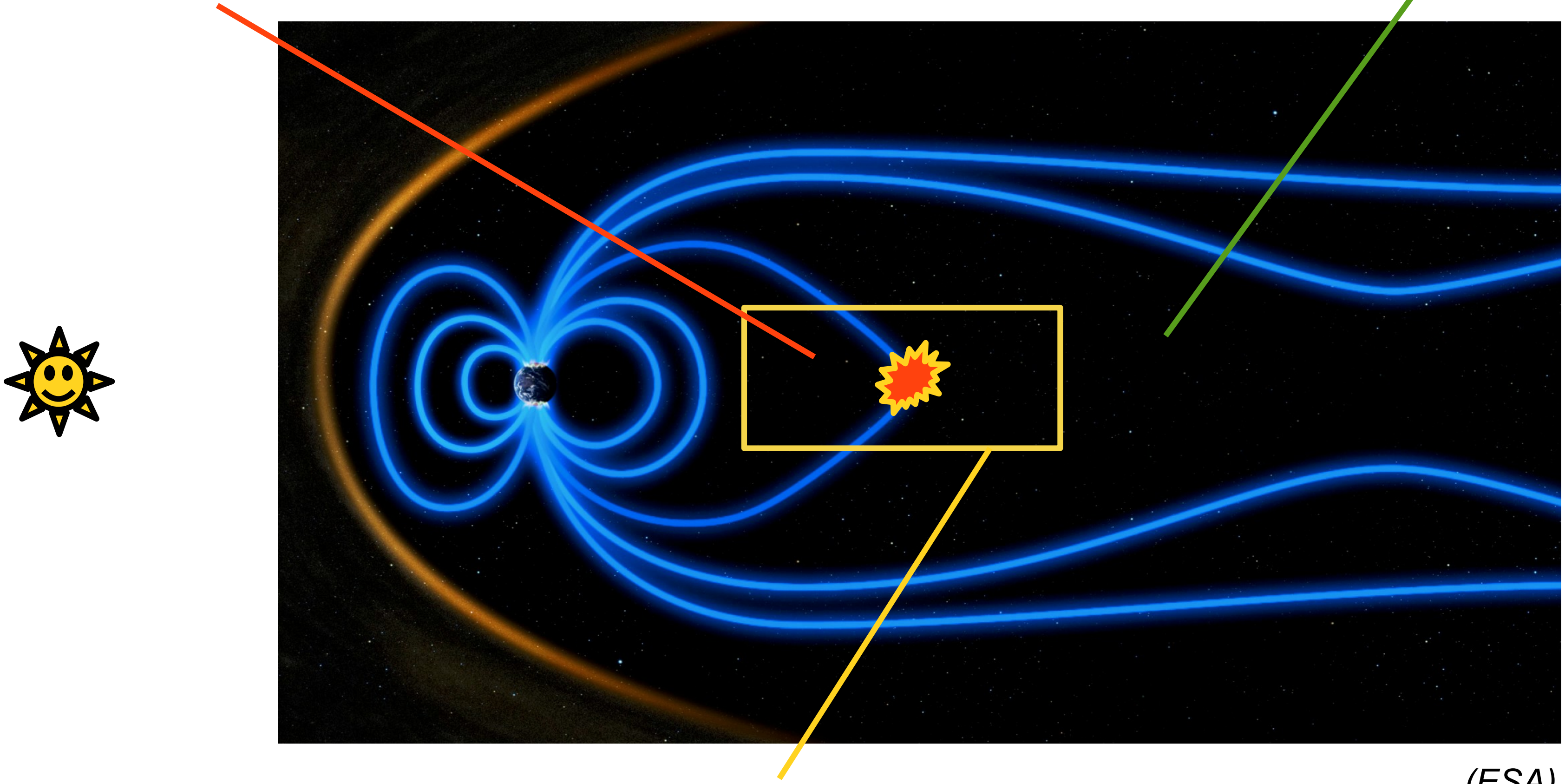
Magnetotail



(ESA)

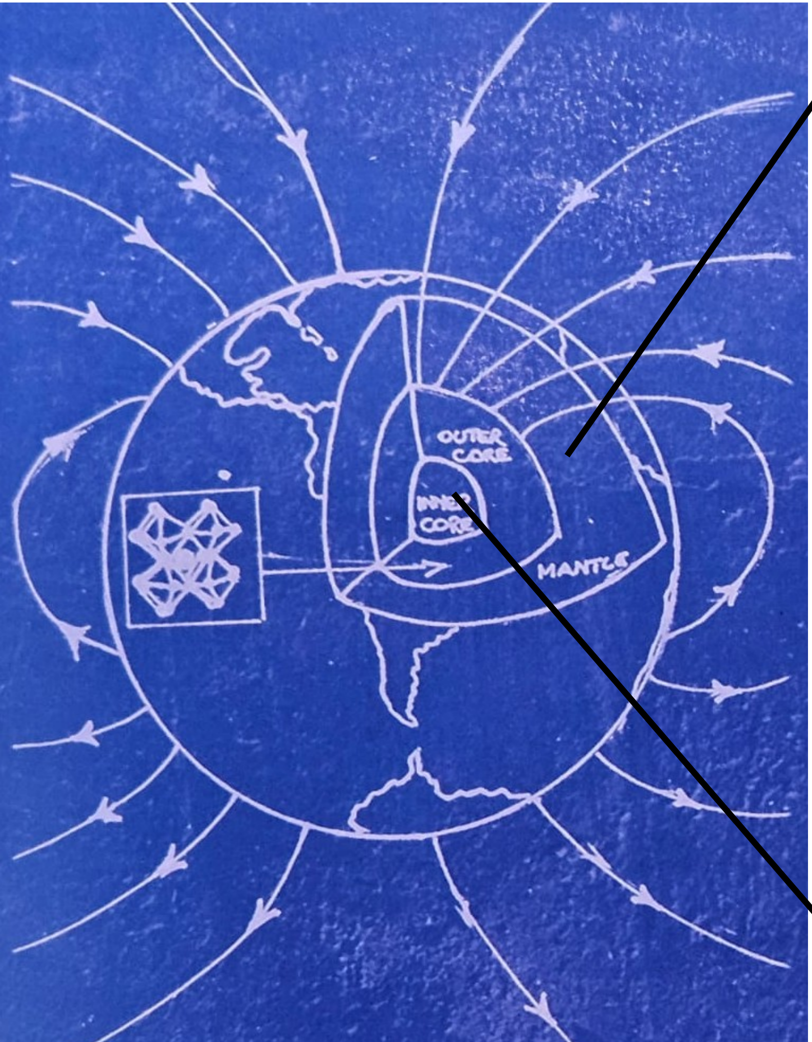
(movie)

Modeling



(ESA)

Coupled Systems



MHD:

$$\frac{\partial \boldsymbol{B}}{\partial t} = \nabla \times (\boldsymbol{U} \times \boldsymbol{B}) + \eta \nabla^2 \boldsymbol{B}$$

$$\frac{\mathrm{D} \ln \rho}{\mathrm{D}t} = -\boldsymbol{\nabla} \cdot \boldsymbol{U}$$

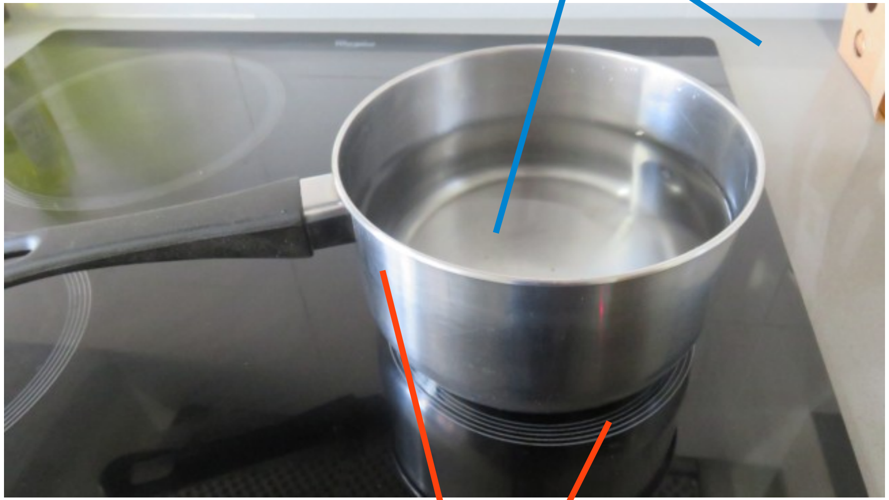
$$\frac{\mathrm{D}\mathbf{U}}{\mathrm{D}t} = -c_{\mathrm{S}}^2 \nabla \left(\frac{\ln T}{\gamma} \ln \rho \right) + \mathbf{J} \times \mathbf{B}/\rho - \mathbf{g} + \mathbf{F}_{\mathrm{visc}}$$

$$\begin{aligned} \frac{\partial \ln T}{\partial t} = &-\mathbf{U} \cdot \nabla \ln T - (\gamma - 1) \nabla \cdot \mathbf{U} \\ &+ \frac{1}{\rho c_V T} \left(\nabla \cdot (K \nabla T) + \eta \mathbf{J}^2 \right. \\ &\left. + 2\rho\nu \mathsf{S} \otimes \mathsf{S} + \zeta \rho (\nabla \cdot \mathbf{U})^2 \right) \end{aligned}$$

heat diffusion: $\frac{\partial T}{\partial t} = \kappa \Delta T$

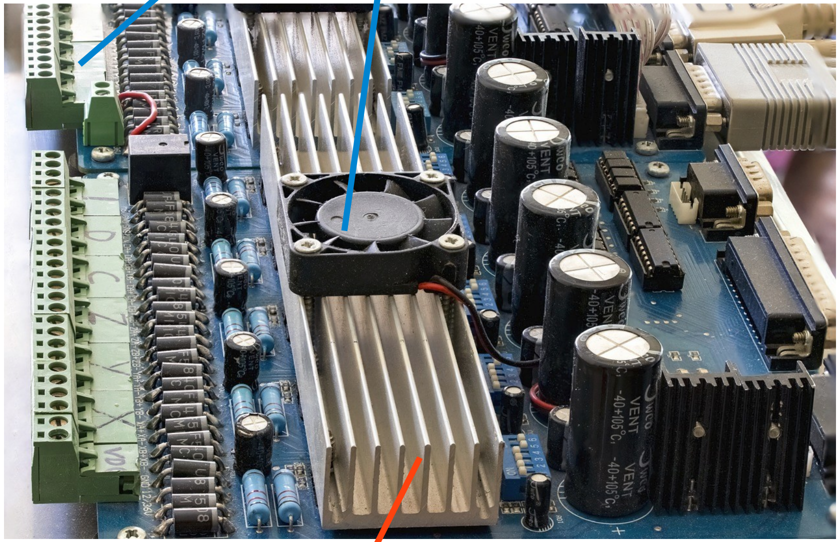
Coupled Systems

Navier-Stokes



heat diffusion

Navier-Stokes



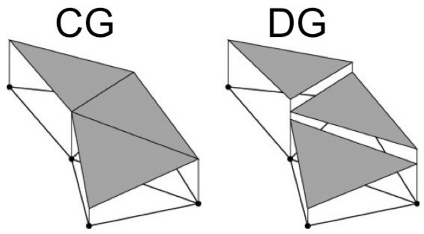
(Ioannis Karathanasis)

heat diffusion

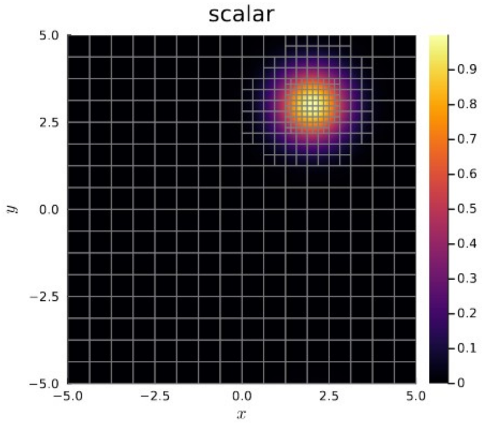
Trixi.jl

open source:
github.com/trixi-framework/Trixi.jl

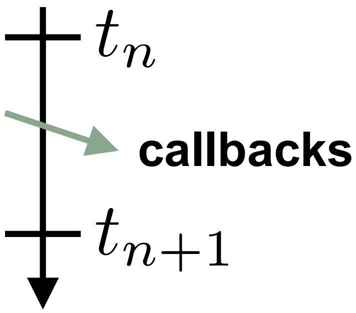
Discontinuous Galerkin



(Stack Overflow)



AMR



OrdinaryDiffEq (SciML)

```
using Trixi

equations = LinearScalarAdvectionEquation2D((0.5, -0.3))
solver = DGSEM(polydeg = 3)
cells_per_dimension = (16, 16)

stepsize_callback = StepsizeCallback(cfl = 1.6)
mesh = StructuredMesh(cells_per_dimension, (-1, -1), (1, 1))

semi = SemidiscretizationHyperbolic(mesh, equations,
                                    initial_condition, solver)
ode = semidiscretize(semi, (0.0, 1.0));

callbacks = CallbackSet(stepsize_callback)
sol = solve(ode, CarpenterKennedy2N54(),
            dt = 1.0, callback = callbacks);
```

Coupling via Converter Functions

Two system with any number of shared variables, including 0:

$$\partial_t u_a + \nabla \cdot f_a(u_a) = 0 \qquad \partial_t u_b + \nabla \cdot f_b(u_b) = 0$$

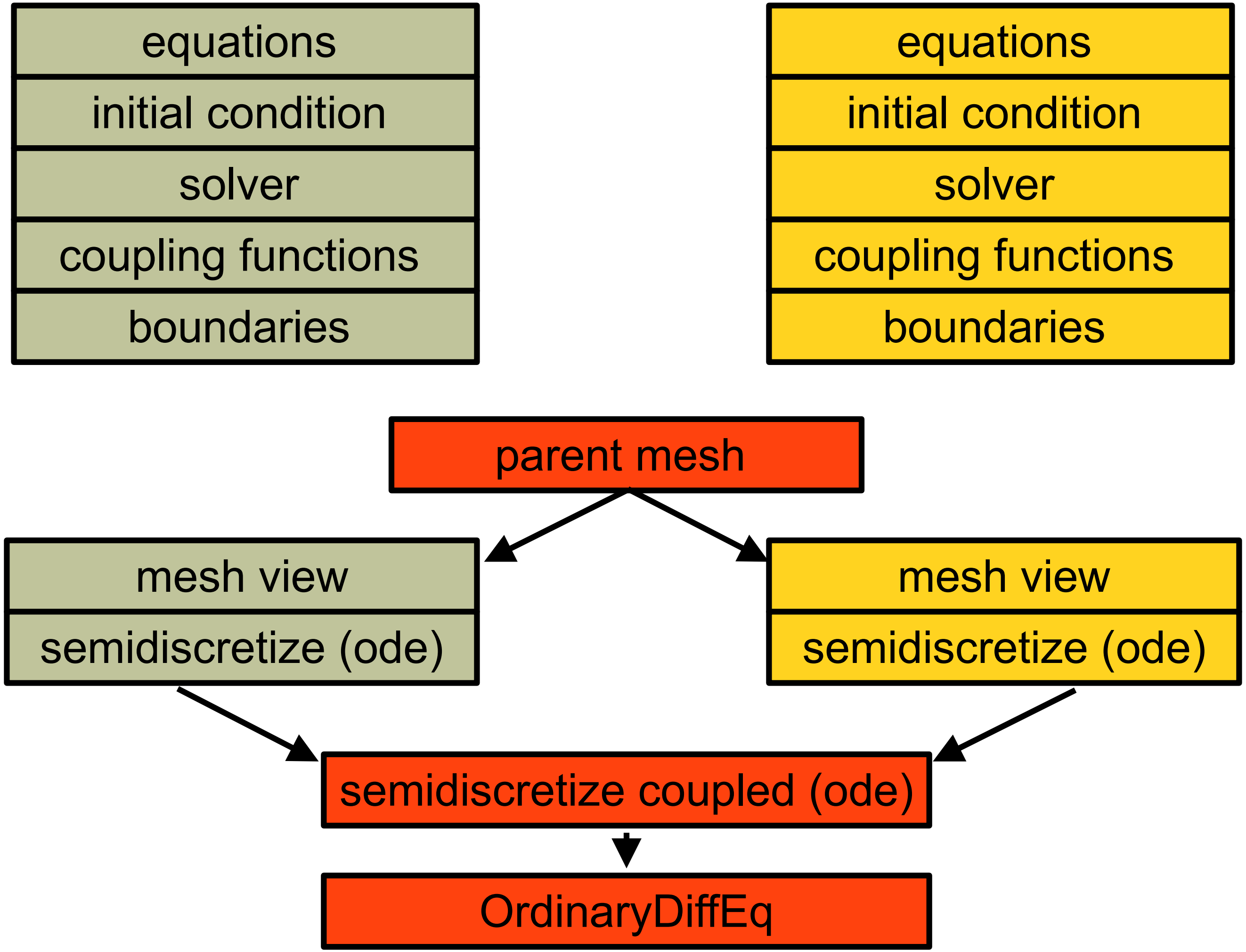
$$c_{ba}(x, u_b) \qquad c_{ab}(x, u_a)$$

2.5d MHD $\rho, v_1, v_2, v_3, p, B_1, B_2, B_3, \Psi$	ρ, v_1, v_2, p 2d Euler

```
coupling_function12 = (x, u, equations_other, equations_own)
                      -> SVector(u[1], u[2], u[3], 0.0, u[4], 0.0, 0.0, 0.0, 0.0)
coupling_function21 = (x, u, equations_other, equations_own) -> SVector(u[1], u[2], u[3], u[5])
```

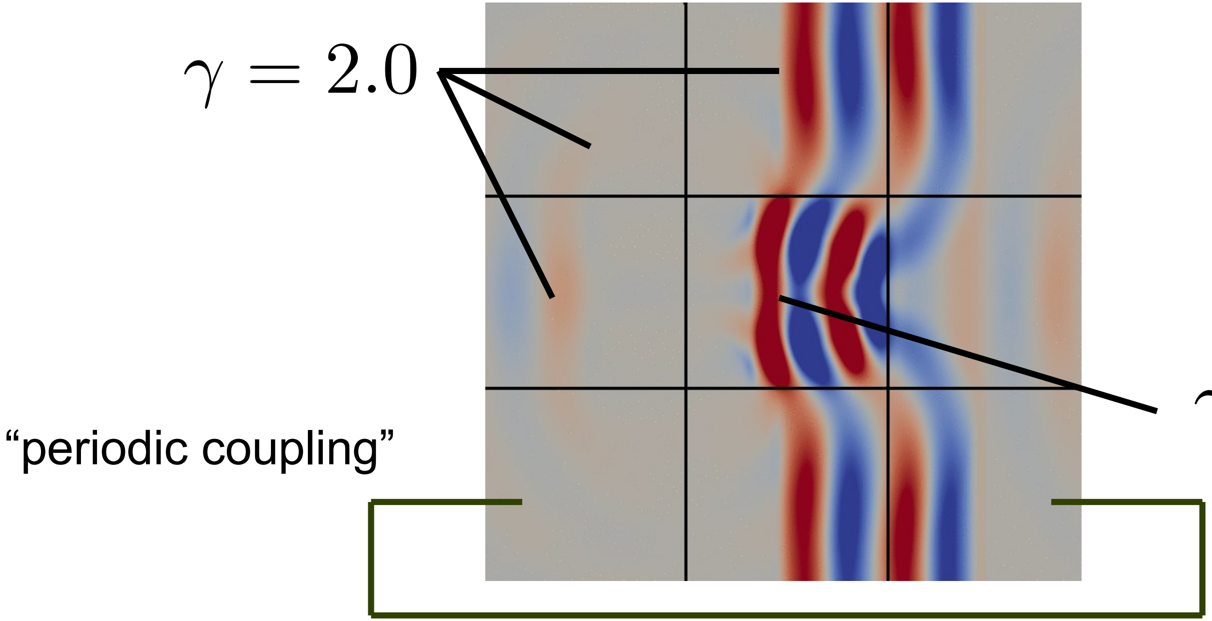
- User can define converter functions.
- Any pair of systems can be coupled.

Work Flow Coupling



Isothermal-Polytropic

$$\partial t \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \end{pmatrix} + \partial x \begin{pmatrix} \rho v_1 \\ \rho v_1^2 + \rho^\gamma \\ \rho v_1 v_2 \end{pmatrix} + \partial y \begin{pmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 + \rho^\gamma \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$



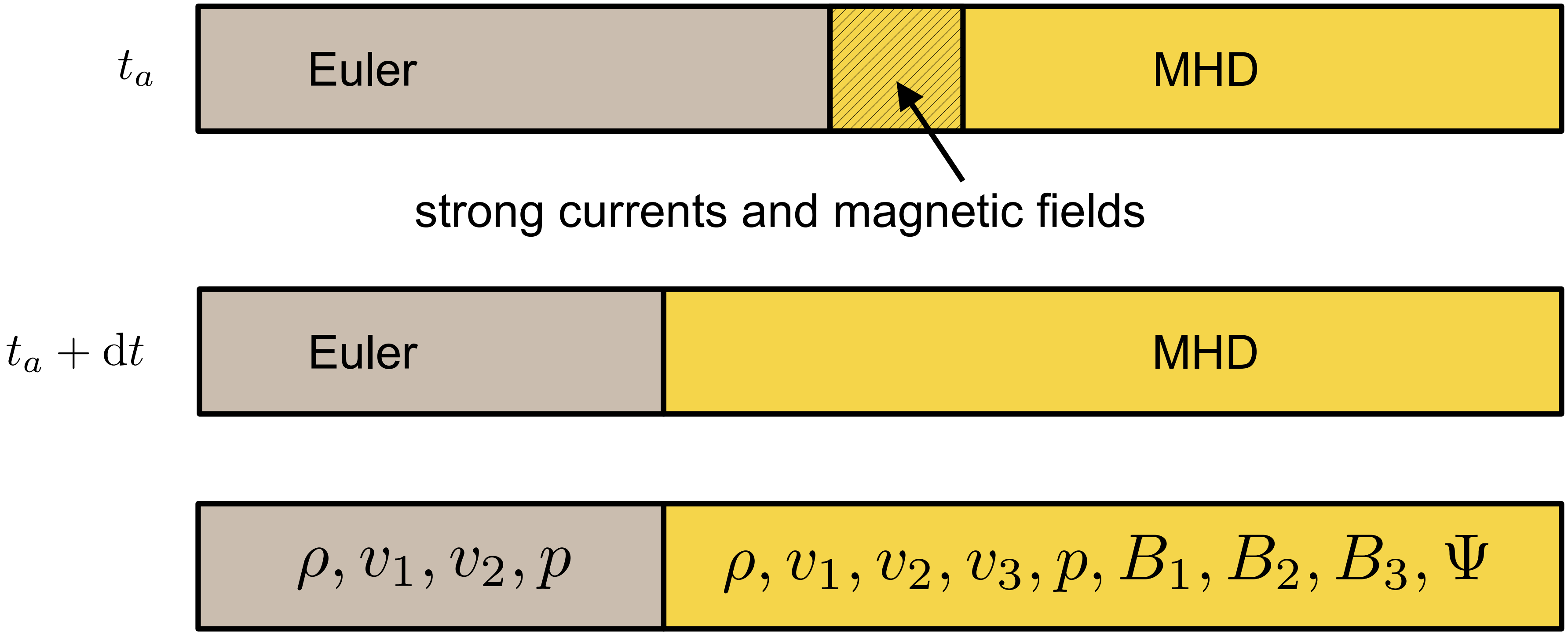
coupling functions:

$$c_{ab} = u^b$$

$$c_{ba} = u^a$$

Isothermal-Polytropic

Adaptive Coupling

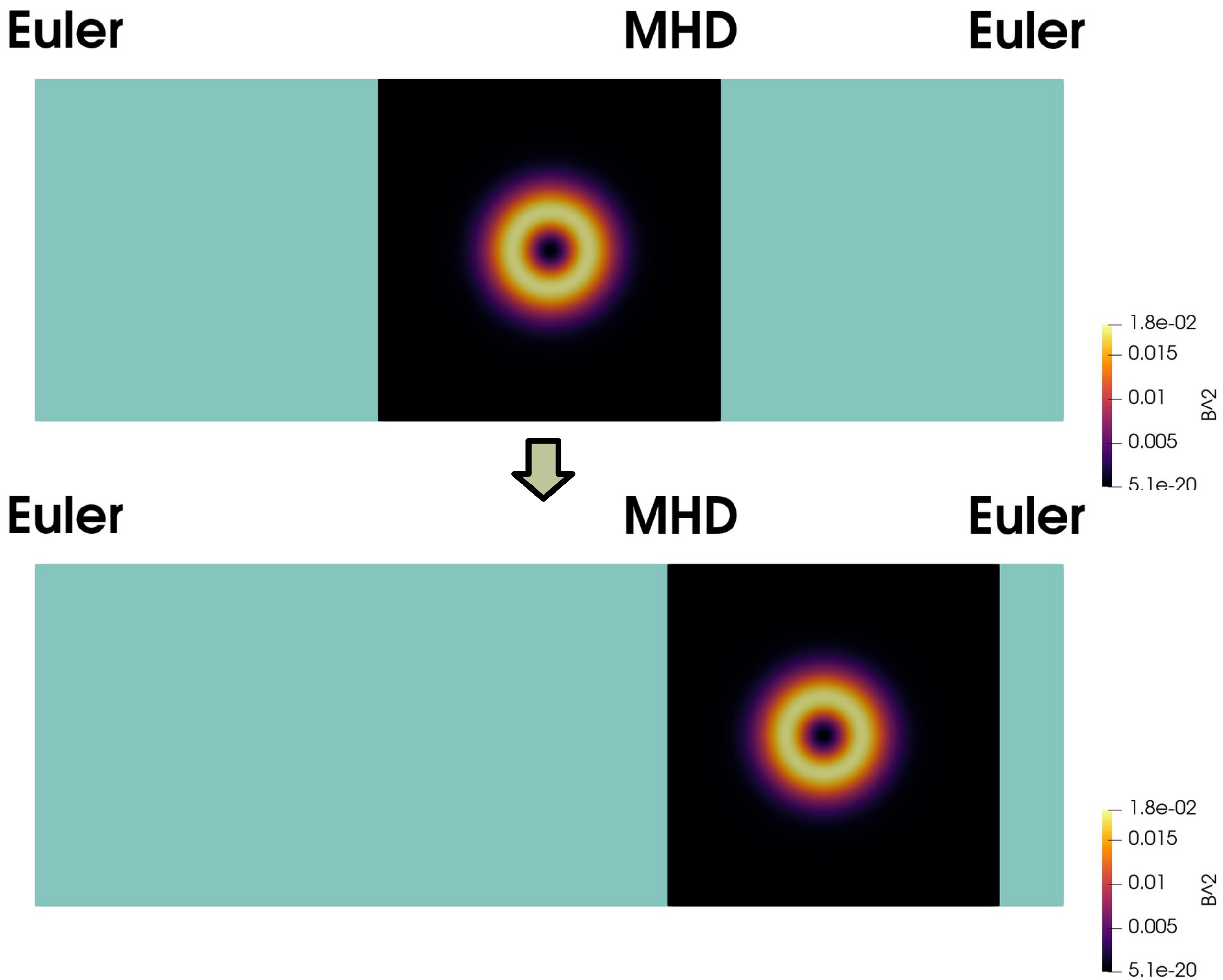


- Use callback functions to remesh.
- Use coupling functions to copy data.

Adaptive Coupling

1. Generate the new grid.
2. Write new u-solution vectors
3. Generate new ODE for OrdinaryDiffEq (integrator).
4. Reinitialize ODE integrator with new problem and new solution vector.

Euler and MHD



Conclusion

- Flexible coupling through converter functions.
- Free domain definitions.
- Adaptive coupling with arbitrary criteria.
- Coupled hierarchy of models.

simon.candelaresi@gmail.com