Multi-patch

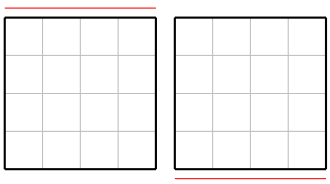
Alexander Hoffmann, Pauline Vidal

March 2024

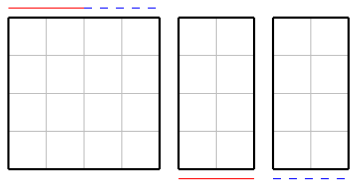
1 Structure for multipatch

- What should be done before September?
- What kind of information do we need to store in each patch?
- What kind of information do the patches need to exchange?

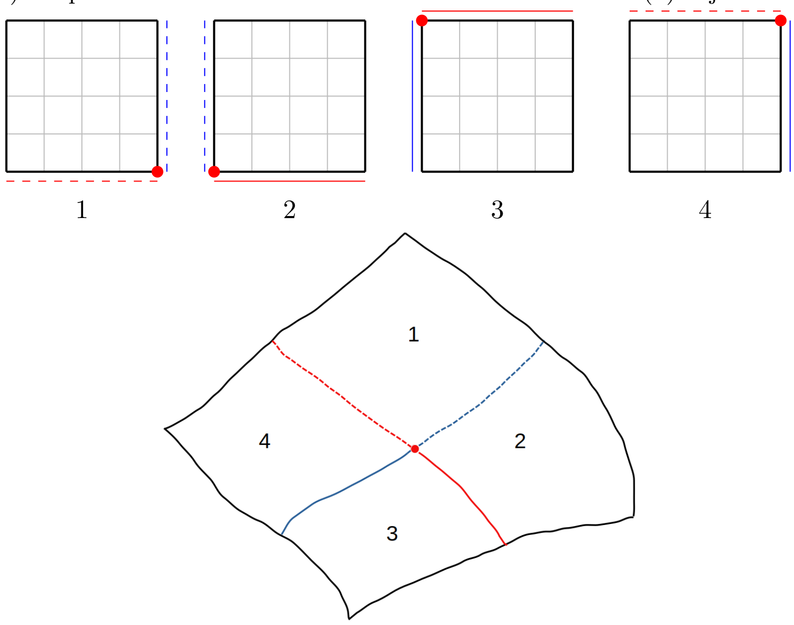
1.1 Type of global space splitting



(a) Simple interface.



(b) T-joint.



(c) X-point in logical and physical domain.

Figure 1: Patches in the logical domain.

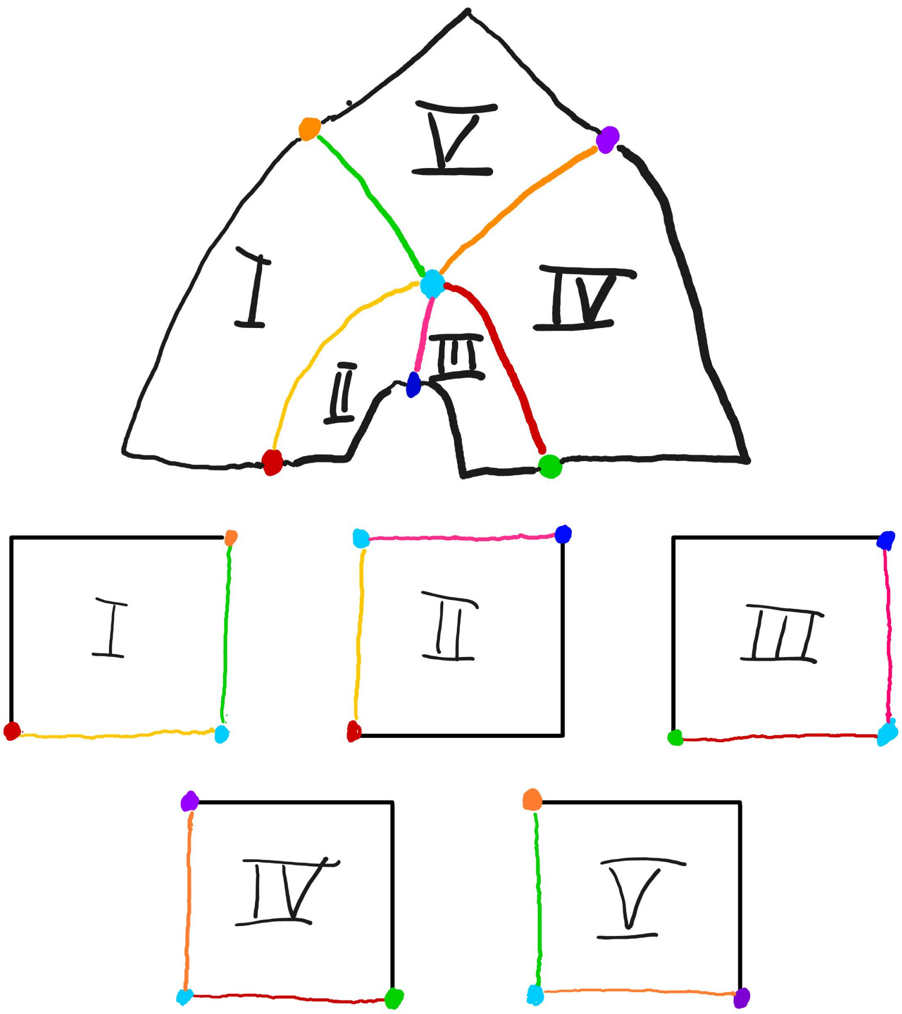


Figure 2: Simple sketch of more complicated geometry with X-point (cyan point). Physical edges and corners are identified on the logical patches. This illustrates the idea how we intend to represent the multipatch geometry using tensor-product patches.

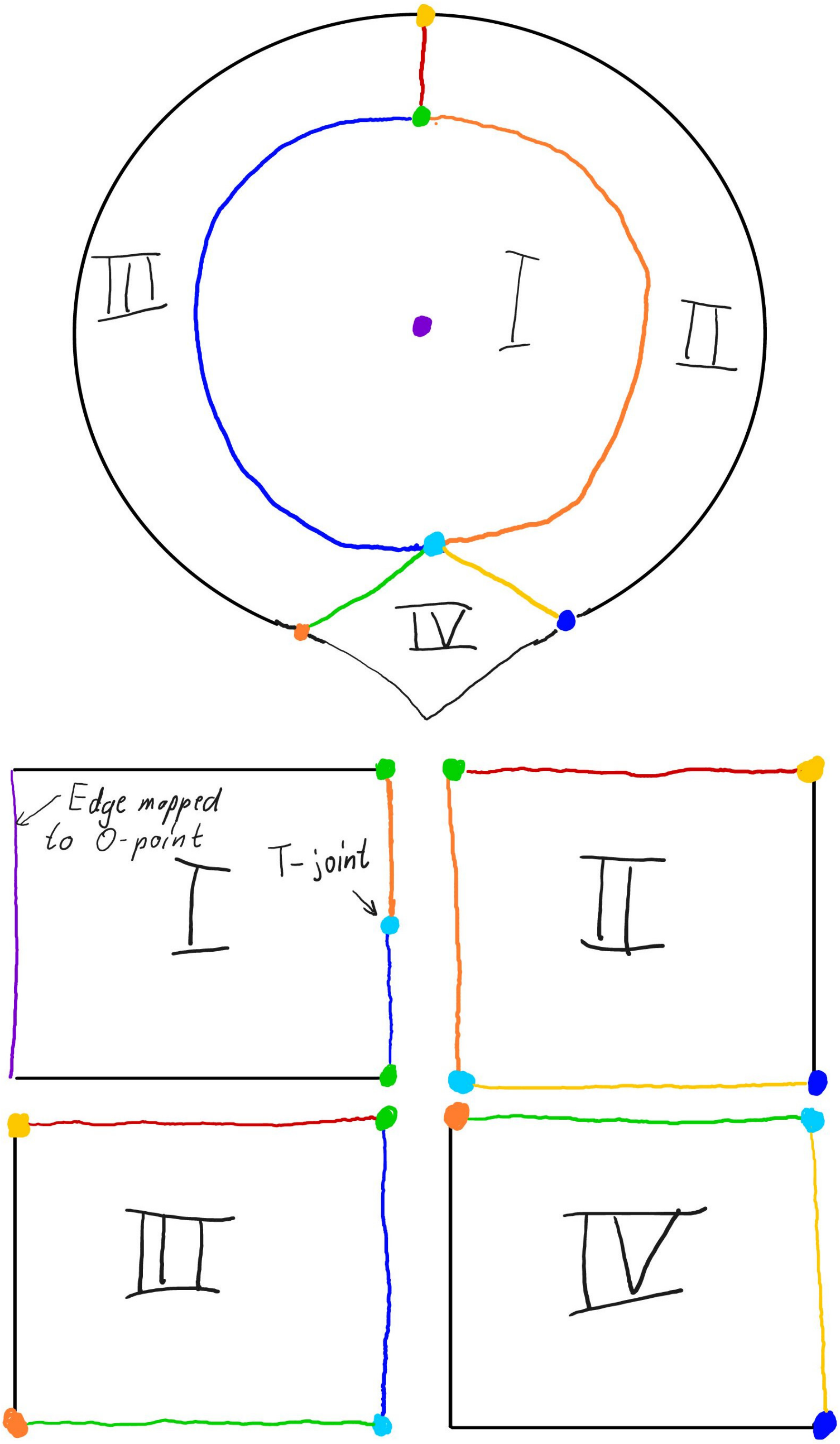


Figure 3: Quick sketch of geometry with O-point (purple point) and T-joint (cyan point). There is a T-joint at the cyan point because edges from patch II and III are connected to the eastern edge of patch I. Note that the y-dimension of patch I is periodic.

2 Advection equation

In the guiding center equations example, the advection equation is given by

$$\partial_t \rho + A \cdot \nabla \rho = 0, \tag{1}$$

with A the advection field defined on the logical domain on the physical domain axis.

Computing the advection field. The advection field is computed from the solution of the Poisson equation. In the guiding center equations example,

$$A = -\nabla \phi \wedge e_z, \tag{2}$$

this computation can be done locally.

Store	• Spline representation (and values?) of ϕ. • Spline representation (and values?) of A. • Jacobian matrix.

Solving the characteristic equation. The equation to solve is

$$\begin{cases} \partial_t X(t; s, x) = A(t, X(t; s, x)), \\ X(s; s, x) = x. \end{cases} \tag{3}$$

we compute

$$\begin{cases} \partial_t X(t^n; t^{n+1}, x) = A(t, X(t^n; t^{n+1}, x)), \\ \hat{X}(t^n; t^{n+1}, x) = \bar{\psi}(t^n; t^{n+1}, x) \end{cases} \tag{4}$$

where $\bar{\psi}$ is a discrete flow computed with a time integration method. In the case of Runge-Kutta method, the advection field needs to be evaluated at intermediate feet.

Example of RK2.

$$\begin{aligned} X^1 &= X^0 - \frac{dt}{2} A(X^0), \\ X^2 &= X^0 - dt A(X^1). \end{aligned} \tag{5}$$

Outside the patch. In case of X^1 is outside the patch we are working on, we need to communicate with the other patches to evaluate the advection field. The same problem appears when we have solved the characteristic equation and want to evaluate the advected function ρ.

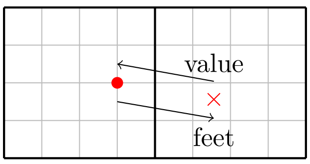


Figure 4: Feet outside the patch in the physical domain.

Exchange	• The outside feet (in the physical domain?). • The evaluated value of the advection field. • The evaluated value of the advected function.
Store	• Spline representation (and values?) of the advected function ρ. • Spline representation (and values?) of the advection field A.

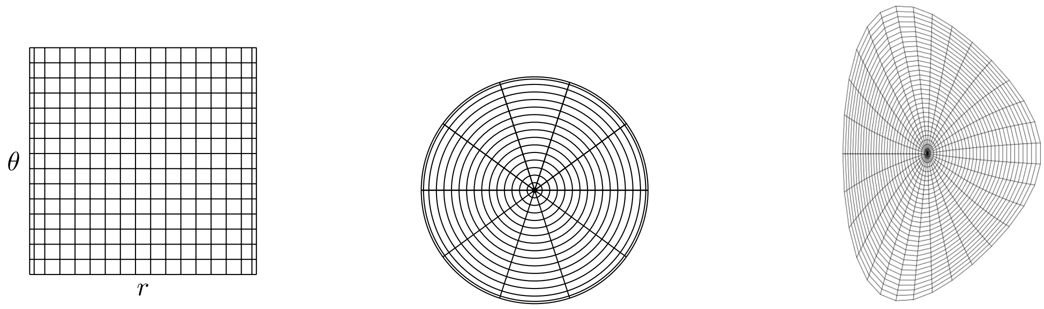
Further questions on outside feet. Problem: the mapping from the logical to the physical domain is not always easily invertible.

- For an advection in the physical domain, if the feet is outside of the domain, how to get the feet in the logical domain of another patch (could be a not invertible mapping for the other patch)?
- For an advection in the pseudo-Cartesian domain, if the feet is outside of the domain, what are the equivalent coordinate in the physical domain?

Two fundamental possibilities:

1. **Advect in physical domain**: If necessary, use control points of the spline mapping to find the patch and then invert the spline mapping to get logical coordinates (preferred by Eric),
2. **Advect in logical/pseudo-cartesian domain**: Extend the coordinates of the patch and find coordinate transformation to logical coordinates of neighboring patches (we do not know yet how to do this, it is just an idea).

If the field lines do not cross the edge (ex. inside of the core), the characteristics do not cross the edge often and in these cases inverting a spline mapping might be feasible. Maybe a clever combination of using logical and physical coordinates in certain situations is the best approach?



(a) Logical domain. (b) Pseudo-Cartesian domain. (c) Physical domain.

Build a spline representation of the advected function. To do it, we

- Evaluate the spline representation of ρ at the characteristic feet [**local but need to communicate**];
- Compute the derivatives at the interfaces of each patch. To do it, there are different methods:
  - advect the derivatives [**local method**];
  - compute the derivatives by:
    - using Lagrange polynomials [**neighbor local (only implies 2 patches)**];
    - using global spline relation [**global method**].
- Build the new spline representation [**local**].

Exchange	• The mesh points around the interfaces for Lagrange interpolation. • The value of ρ around the interfaces for Lagrange interpolation. • The sum of values of the function $\sum_i \alpha_i s(x_i) = \sum_{p\in patches} \sum_{x_i \in p} \alpha_i s(x_i)$.
Store	• Spline representation of the advected function ρ.

2.1 Further questions

About evaluating functions.

- How do we know in which patch we are?
  - For a given point in the physical domain, how do we know in which subset it belongs to?
- Let's assume we know in which patch we are, how can we evaluate a function? How can we get $\rho(x, y)$ with (x, y) in the physical domain (especially when the mapping isn't invertible)?

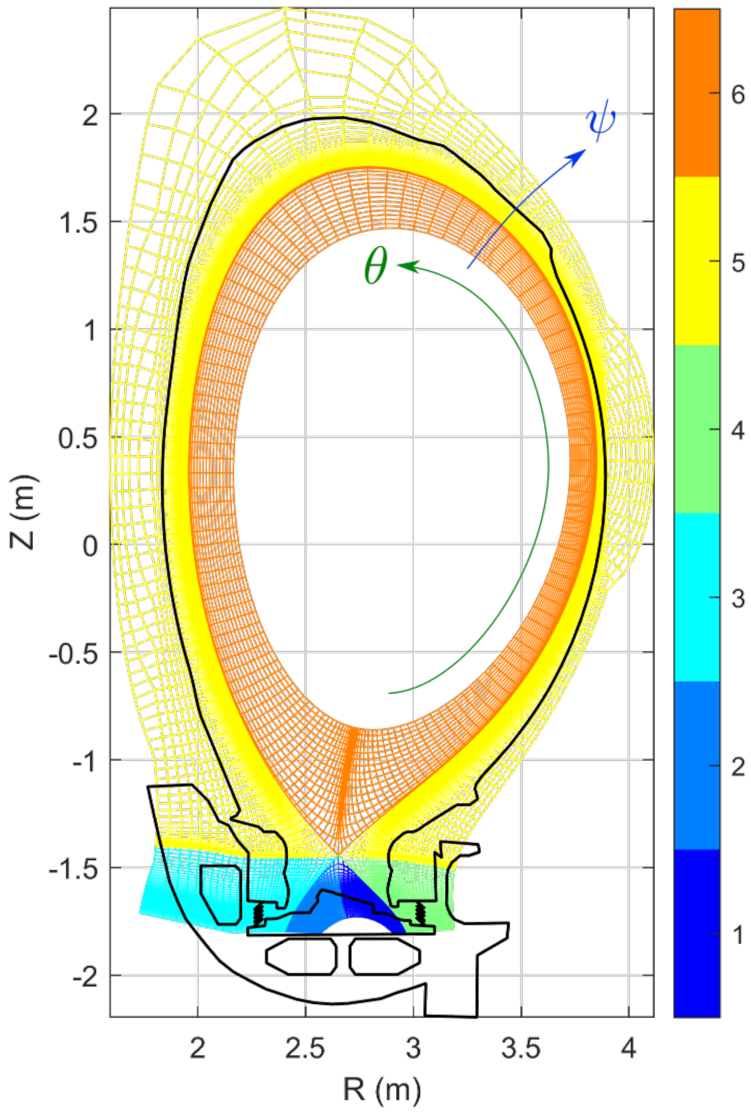


Figure 6: Multi-patch decomposition from SOLEDGE.