# Isogeometric analysis for plasma physics applications

Eric Sonnendrücker

Max Planck Institute for Plasma Physics
and
Technical University of Munich

Rome, March, 20, 2019

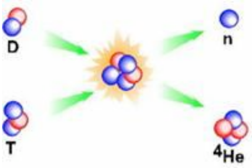
Magnetic fusion: physics and models

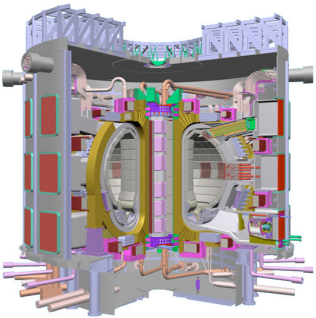
Structure preserving discretisation

Fast solvers for Poisson and implicit MHD

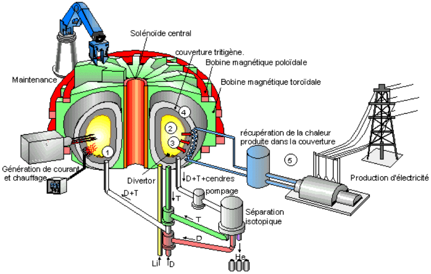
Software framework based on geometric concepts

# Magnetic fusion



- Magnetic confinement (ITER)
- Inertial confinement, laser: LMJ, NIF



# Max-Planck Institute for plasma physics


**Two sites:**

<u>Greifswald (staff ~450)</u>

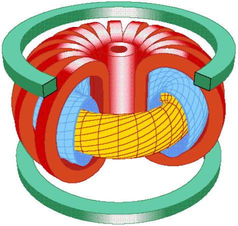
Stellarator Wendelstein 7-X

<u>Garching (staff ~700)</u>

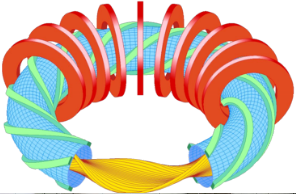
Tokamak ASDEX Upgrade

# Tokamaks and Stellarators

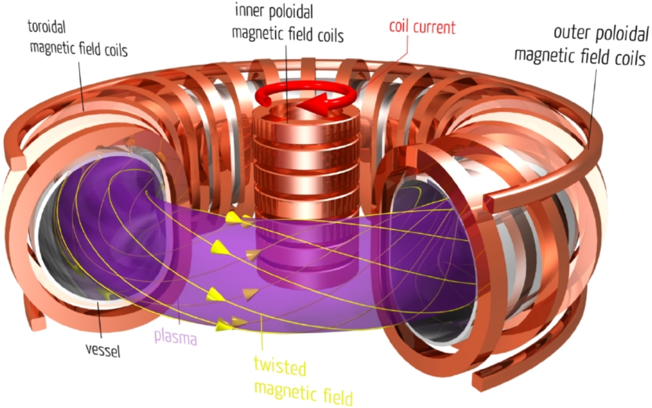
**Tokamak**

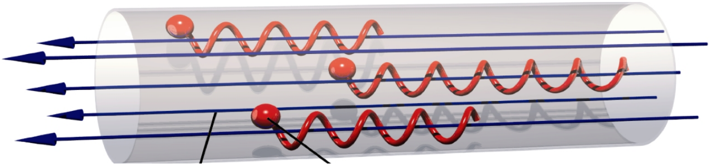



**Stellarator**



# Magnetic field lines in Tokamaks



# A hierarchy of models

- The non-relativistic Vlasov-Maxwell system:

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_s + \frac{q_s}{m_s}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f_s = 0.$$

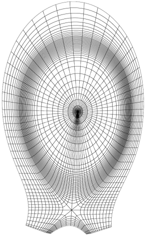
$$\frac{\partial \mathbf{E}}{\partial t} - \operatorname{curl} \mathbf{B} = -\mathbf{J} = \sum_s q_s \int f_s \mathbf{v} \, d\mathbf{v}, \qquad \frac{\partial \mathbf{B}}{\partial t} + \operatorname{curl} \mathbf{E} = 0,$$

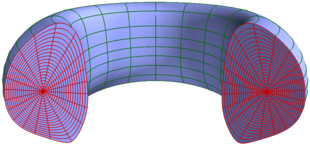
$$\operatorname{div} \mathbf{E} = \rho = \sum_s q_s \int f_s \, d\mathbf{v}, \qquad \operatorname{div} \mathbf{B} = 0.$$

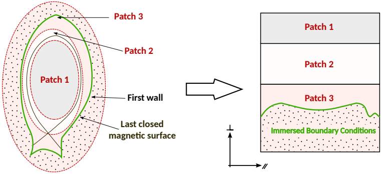
- For low frequency electrostatic problems Maxwell can be replaced by Poisson: Vlasov-Poisson model
- For slowly varying large magnetic field Vlasov can be replaced by gyrokinetic model either electromagnetic or electrostatic.
- Taking the velocity moments, we get the Braginskii model analogous to Euler for non neutral fluids.
- Further assumptions lead to one fluid MHD model.

# The magnetic geometry

- Magnetic field lines stay on concentric topological tori (called flux surfaces)
- Behaviour of plasma very different along and across the magnetic field. Transport and diffusion orders of magnitude larger on flux surfaces.
- Numerical accuracy benefits a lot from aligning mesh on flux surface
- The tokamak wall does not correspond to a flux surface. Embedded boundary needed if complete alignment to flux surface is desired.

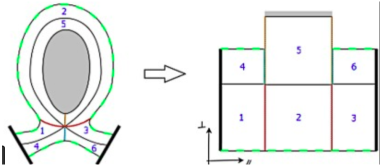


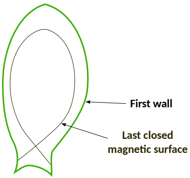


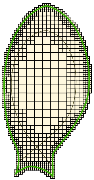


# Different meshing options

- Locally refined cartesian mesh. Neither aligned to flux surfaces nor to Tokamak wall
- Align on flux surfaces only in confined part (closed flux surfaces)
- Mesh based on multiple patches, with B-spline mapping (Tokamesh)
  - Generated numerically from plasma equilibrium.
  - B-spline mapping on each patch.
  - $C^1$ continuity enforced except at O-point and X-point.



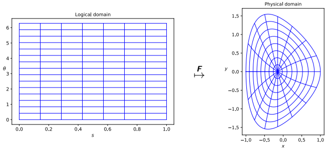



Computational mesh

# $C^1$ smooth polar splines

The B-splines mapping of our central patch $\mathbf{F}(s,\theta) = (x(s,\theta), y(s,\theta))$ collapses to a single point $(x_0, y_0)$ for $s = 0$ ($C^k$ continuity lost at this point)




- Following Toshniwal, Speleers, Hiemstra, Hughes (2017), desired continuity can be restored by linear combinations of the first rows of control points around pole.
- We construct a triangle with vertices $(T_0, T_1, T_2)$ related to control points near pole. Its barycentric coordinates $\lambda_i$ define the three new $C^1$ basis functions

$$\tilde{N}_l(s,\theta) = \lambda_l(x_0, y_0) N_0^s(s) + \left( \sum_{j=0}^{n_\theta - 1} \lambda_l(c^x_{1,j}, c^y_{1,j}) N_j^\theta(\theta) \right) N_1^s(s), \;\; l = 0, 1, 2.$$

- Implemented by Zoni, Güçlü at O-point. X-point being developed.

# Importance of structure preservation in simulations

- For ODEs preservation of symplectic structure essential for long time simulations. Exact preservation of approximate energy enables efficient integrators over very long times.
- In many cases keeping structure of continuous equations at discrete level more important than order of accuracy.
  - Avoid spurious eigenmodes in Maxwell's equations.
  - Avoid spurious perpendicular diffusion in parallel transport.
  - Stability issues when not preserving $\nabla \cdot \mathbf{B} = 0$ or $\nabla \cdot \mathbf{E} = \rho$ in Maxwell or MHD
- Big success of structure preserving methods
  - L-shaped domain for Maxwell's equations
  - Non simply connected domains, *i.e.* annulus, torus. Non trivial space of harmonic functions.

## Hamiltonian systems

- Canonical Hamiltonian structure preserved by symplectic integrators

$$\frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} = \nabla_p H, \quad \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -\nabla_q H \quad \text{with } \mathbf{z} = (\mathbf{q}, \mathbf{p}) : \qquad \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = \mathbb{J}\nabla_z H$$

$$\text{where} \quad \mathbb{J} = \begin{pmatrix} 0_N & I_N \\ -I_N & 0 \end{pmatrix}$$

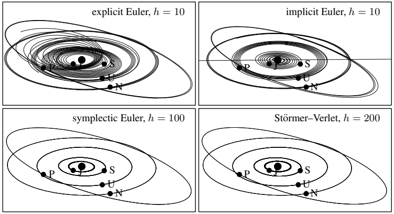
- Non canonical Hamiltonian structure with Poisson matrix $\mathbb{J}(\mathbf{z})$

$$\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = \mathbb{J}(\mathbf{z})\nabla_z H, \qquad \text{Poisson bracket:} \quad \{F, G\} = (\nabla_z F)\mathbf{J}(\mathbf{z})\nabla_z G$$

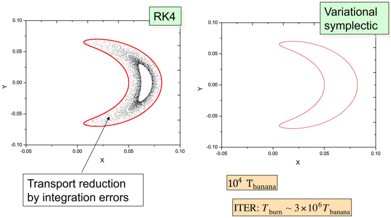
- $\mathbf{J}$ can be degenerate then functionals $C$ such that $\mathbf{J}(\mathbf{z})\nabla_z C = 0$ are Casimirs which are conserved by the dynamics, *e.g.* $\operatorname{div}\mathbf{B} = 0$ for Maxwell or MHD.
- Conservation of Casimirs essential for long time simulations
- Also for infinite dimensional systems

# Structure preservation for dynamical systems

- For ODEs preservation of symplectic structure well known: Symplectic integrators. Exact preservation of approximate energy enables efficient integrators over very long times.
- For long time simulations keeping structure of continuous equations at discrete level more important than order of accuracy.




Hairer, Lubich, Wanner, "Geometric numerical integration"




Hong Qin et al., PoP 16

# Metriplectic structure

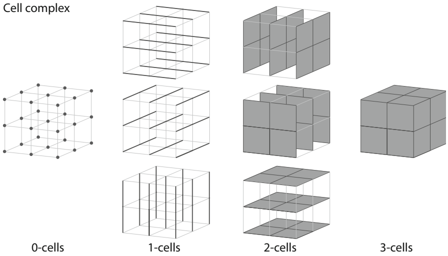
- Introduced by P.J. Morrison (1980s) for fluids and plasmas.
- Dynamical systems arising in physics often combine a symplectic and a dissipative part
- Introducing a hamiltonian $\mathcal{H}$ which is conserved and a free energy (or entropy) $\mathcal{S}$ which is dissipated,

$$\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}t} = \{\mathcal{F}, \mathcal{H}\} + (\mathcal{F}, \mathcal{S}) \qquad \equiv \qquad \frac{\mathrm{d}U}{\mathrm{d}t} = \mathbb{J}(U)\frac{\delta\mathcal{H}}{\delta U} - \mathbb{K}(U)\frac{\delta\mathcal{S}}{\delta U}$$

  with $\mathbb{J}$ a Poisson operator and $\mathbb{K}$ a symmetric semi-positive operator: exact energy preservation and H-theorem (production of entropy)
- Reproduce this structure automatically at discrete level for robust and stable discretisation
- Expression of these elements used for automatic code generation.
- e.g. for kinetic plasma model.

# Geometric description of physics

- Geometric objects provide a more accurate description of physics and also a natural path for discretisation.
  1. Potentials are naturally evaluated at points
  2. The action of a force is measured through its circulation along a path
  3. Current is the flux through a surface of current density
  4. Charge is integral over volume of charge density
- Should be discretized accordingly




- Related to discretization of differential 0-,1-,2- and 3-forms.

# Integral form of Maxwell's equations

| Integral equations | Differential equations |
|---|---|
| $\oint_{\partial \mathbf{S}} \mathbf{H} \cdot d\ell = \int_{\mathbf{S}} \left( \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \right) \cdot d\mathbf{S}$ | $\operatorname{curl} \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$ |
| $\oint_{\partial \mathbf{S}} \mathbf{E} \cdot d\ell = \int_{\mathbf{S}} \left( -\frac{\partial \mathbf{B}}{\partial t} \right) \cdot d\mathbf{S}$ | $\operatorname{curl} \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$ |
| $\oint_{\partial \mathbf{V}} \mathbf{D} \cdot d\mathbf{S} = \int_{\mathbf{V}} \rho d\mathbf{V}$ | $\operatorname{div} \mathbf{D} = \rho$ |
| $\oint_{\partial \mathbf{V}} \mathbf{B} \cdot d\mathbf{S} = 0$ | $\operatorname{div} \mathbf{B} = 0$ |

- **D** and **E** as well as **H** and **B** are related by constitutive equations dependent on material properties.
- Exact discrete version of integral form can be obtained provided degrees of freedom for **H** and **E** are edge integrals and degrees of freedom for **D** and **B** (and **J**) are face integrals.

# Exact relations between degrees of freedom

- Denote by respectively $\mathcal{V}_i$, $\mathcal{F}_i$, $\mathcal{E}_i$, $\mathbf{x}_i$, the volumes (cells), faces, edges and points of the mesh.
- Degrees of freedom are (*e.g.* for $\mathbf{B}$ and $\mathbf{E}$)

$$\mathcal{F}_i(\mathbf{B}) = \int_{\mathcal{F}_i} \mathbf{B} \cdot \mathrm{d}\mathbf{S}, \quad \mathcal{E}_i(\mathbf{E}) = \int_{\mathcal{E}_i} \mathbf{E} \cdot \mathrm{d}\ell, \quad \ldots$$

- Then integral form of Maxwell yields exact relations involving each face and its 4 boundary edges

$$\mathcal{F}_i(\mathbf{J}) + \frac{\partial \mathcal{F}_i(\mathbf{D})}{\partial t} = \mathcal{E}_{i,1}(\mathbf{H}) + \mathcal{E}_{i,2}(\mathbf{H}) - \mathcal{E}_{i,3}(\mathbf{H}) - \mathcal{E}_{i,4}(\mathbf{H}) \tag{1}$$

$$\frac{\partial \mathcal{F}_i(\mathbf{B})}{\partial t} = -\mathcal{E}_{i,1}(\mathbf{E}) - \mathcal{E}_{i,2}(\mathbf{E}) + \mathcal{E}_{i,3}(\mathbf{E}) + \mathcal{E}_{i,4}(\mathbf{E}) \tag{2}$$

- Similar exact relations for divergence constraints.
- This depends only on mesh connectivity and remains true if mesh is smoothly deformed (without tearing).

# Reconstruction of fields from degrees of freedom

- Discrete constitutive equations still needed to couple Ampere and Faraday.
- Need to evaluate fields at arbitrary particle positions.
- The fields associated to different degrees of freedom (point values, edge integrals, face integrals, volume integrals) need to be reconstructed in a compatible manner.
- Related to geometric discretisation of various PDEs:
  - Dual meshes: Mimetic Finite Differences, Compatible Operator Discretisation, Discrete Duality Finite Volumes. Intuitive metric association between primal and dual mesh.
  - Dual operators: Finite Element formulation, mathematically more elaborate: Primal operators (strong form) on primal complex, dual operators (weak form) on dual complex.
- Charge conserving PIC algorithms (Villasenor-Bunemann, Esirkepov,..) can also be understood in this framework.

# Finite Element Exterior Calculus (FEEC)

- Mathematical framework for Finite Element solvers is provided FEEC introduced by Arnold, Falk and Winther. Buffa et al. introduced complex for B-splines.
- Continuous and discrete complexes for splines are the following

$$
\begin{array}{ccccccc}
H^1(\Omega) & \xrightarrow{\mathbf{grad}} & H(\mathbf{curl},\Omega) & \xrightarrow{\mathbf{curl}} & H(\mathrm{div},\Omega) & \xrightarrow{\mathrm{div}} & L^2(\Omega) \\
\downarrow \Pi_0 & & \downarrow \Pi_1 & & \downarrow \Pi_2 & & \downarrow \Pi_3 \\
V_0 & \xrightarrow{\mathbf{grad}} & V_1 & \xrightarrow{\mathbf{curl}} & V_2 & \xrightarrow{\mathrm{div}} & V_3 \\
= & & = & & = & & = \\
\mathcal{S}^{p,p,p} & \longrightarrow & \begin{pmatrix} \mathcal{S}^{p-1,p,p} \\ \mathcal{S}^{p,p-1,p} \\ \mathcal{S}^{p,p,p-1} \end{pmatrix} & \longrightarrow & \begin{pmatrix} \mathcal{S}^{p,p-1,p-1} \\ \mathcal{S}^{p-1,p,p-1} \\ \mathcal{S}^{p-1,p-1,p} \end{pmatrix} & \longrightarrow & \mathcal{S}^{p-1,p-1,p-1}
\end{array}
$$

- Commuting diagram is an essential piece

$$\Pi_1 \mathbf{grad}\psi = \mathbf{grad}\Pi_0\psi, \quad \Pi_2\mathbf{curl}\mathbf{A} = \mathbf{curl}\Pi_1\mathbf{A}, \quad \Pi_3 \mathrm{div}\mathbf{A} = \mathrm{div}\Pi_2\mathbf{A}.$$

# The commuting projection operators

- Commuting diagram by interpolating right degrees of freedom:
  - Elements of $V_0$ are characterized by point values
  - Elements of $V_1$ are characterized by edge integrals
  - Elements of $V_2$ are characterized by surface integrals
  - Elements of $V_3$ are characterized by volume integrals
- $\Pi_0\psi = \psi_h \in V_0$ defined by $\psi_h(\mathbf{x}) = \sum_i c_i^0 \Lambda_i^0(\mathbf{x})$, with $c_i^0$ solution of the interpolation problem $\psi_h(\mathbf{x}_j) = \psi(\mathbf{x}_j)\ \forall j$
- $\Pi_1\mathbf{A} = \mathbf{A}_h \in V_1$ defined by $\mathbf{A}_h(\mathbf{x}) = \sum_i c_i^1 \boldsymbol{\Lambda}_i^1(\mathbf{x})$, with $c_i^1$ solution of

$$\int_{\mathcal{E}_j} \mathbf{A}_h(\mathbf{x}) \cdot \mathrm{d}\ell = \int_{\mathcal{E}_j} \mathbf{A}(\mathbf{x}) \cdot \mathrm{d}\ell \quad \forall j$$

- $\Pi_2\mathbf{B} = \mathbf{B}_h \in V_2$ defined by $\mathbf{B}_h(\mathbf{x}) = \sum_i c_i^2 \boldsymbol{\Lambda}_i^2(\mathbf{x})$, with $c_i^2$ solution of

$$\int_{\mathcal{F}_j} \mathbf{B}_h(\mathbf{x}) \cdot \mathrm{d}\mathbf{S} = \int_{\mathcal{F}_j} \mathbf{B}(\mathbf{x}) \cdot \mathrm{d}\mathbf{S} \quad \forall j$$

- $\Pi_3\varphi = \varphi_h \in V_3$ defined by $\varphi_h(\mathbf{x}) = \sum_i c_i^3 \boldsymbol{\Lambda}_i^3(\mathbf{x})$, with $c_i^3$ solution of $\int_{\mathcal{V}_j} \varphi_h(\mathbf{x}) \mathrm{d}\mathbf{x} = \int_{\mathcal{V}_j} \varphi(\mathbf{x}) \mathrm{d}\mathbf{x}\ \forall j$

# Particle mesh coupling

- Charge and current computed on mesh using smoothed particles. For some given smoothing function $S$, typically a B-spline (at least quadratic to reduce aliasing and variance)

$$f_N(t, \mathbf{x}, \mathbf{v}) = \sum_k w_k S(\mathbf{x} - \mathbf{x}_k(t))\delta(\mathbf{v} - \mathbf{v}_k(t)).$$

- From this expression, we can compute the charge and current densities

$$\rho_N = \sum_k w_k q_k S(\mathbf{x} - \mathbf{x}_k(t)),$$

$$\mathbf{J}_N = \sum_k w_k q_k \mathbf{v}_k S(\mathbf{x} - \mathbf{x}_k(t)).$$

- Discrete values defined by projecting associated charge and current

$$\mathbf{J}_h = \Pi_2(\mathbf{J}_N), \qquad \rho_h = \Pi_3(\rho_N).$$

# Semi-discrete continuity equation

- A direct calculation shows that

$$\frac{\partial \rho_N}{\partial t} = -\sum_k w_k q_k \mathbf{v}_k \cdot \nabla S(\mathbf{x} - \mathbf{x}_k(t)) = -\operatorname{div} \mathbf{J}_N.$$

- Applying $\Pi_3$ we get

$$\Pi_3 \frac{\partial \rho_N}{\partial t} = \frac{\partial \rho_h}{\partial t} = -\Pi_3 \operatorname{div} \mathbf{J}_N = -\operatorname{div} \Pi_2 \mathbf{J}_N = -\operatorname{div} \mathbf{J}_h,$$

  using the commutation property. Hence

$$\frac{\partial \rho_h}{\partial t} + \operatorname{div} \mathbf{J}_h = 0.$$

- Then also

$$\frac{\partial \operatorname{div} \mathbf{E}_h}{\partial t} = -\frac{1}{\varepsilon_0} \operatorname{div} \mathbf{J}_h = \frac{1}{\varepsilon_0} \frac{\partial \rho_h}{\partial t},$$

- Gauss is a consequence of Ampere and initial value as in continuous case.

# Finite Element discretisation

- One equation Faraday or Ampere discretized strongly, with no approximation. The other weakly with integration by parts:
  - Strong Ampere - Weak Faraday. Smooth $J_N$ needed

$$-\frac{\partial \mathbf{D}_h}{\partial t} + c^2 \operatorname{curl} \mathbf{H}_h = \frac{1}{\varepsilon_0} \sum_p q_p \Pi_2[\mathbf{v}_p S(\mathbf{x} - \mathbf{x}_k(t))] = \frac{1}{\varepsilon_0} \mathbf{J}_h,$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega \mathbf{H}_h \cdot \mathbf{C}_h \mathrm{d}\mathbf{x} + \int_\Omega \mathbf{D}_h \cdot \operatorname{curl} \mathbf{C}_h \mathrm{d}\mathbf{x} = 0 \quad \forall \mathbf{C}_h \in V_1.$$

  - Weak Ampere - Strong Faraday

$$-\int_\Omega \frac{\partial \mathbf{E}_h}{\partial t} \cdot \mathbf{F}_h \mathrm{d}\mathbf{x} + c^2 \int_\Omega \mathbf{B}_h \cdot \operatorname{curl} \mathbf{F}_h \mathrm{d}\mathbf{x} = \frac{1}{\varepsilon_0} \int \mathbf{J}_h \cdot \mathbf{F}_h \mathrm{d}\mathbf{x}, \quad \forall \mathbf{F}_h \in V_1,$$

$$\frac{\partial \mathbf{B}_h}{\partial t} + \operatorname{curl} \mathbf{E}_h = 0$$

No smoothing needed because of integral on $\mathbf{J}_h$, but smoothing or filtering can be added.

# GEMPIC framework

- Discretization of fields: Compatible finite elements (discrete de Rham complex), e.g. splines, Fourier, Lagrange:
  - Strong Faraday **E** edge elements (1-form), **B** face elements (2-form)
  - Strong Ampere **B** edge elements (1-form), **E** face elements (2-form)
- Discretization of $f$ with (smoothed) particles $f(t, x, v) = \sum w_p S(x - x_p(t))\delta(v - v_p(t))$, $S$ can be $\delta$ for strong Faraday.
- Plug discretizations for $f$, $E$ and $B$ into Lagrangian to get formulation of equations based on a semi-discrete Hamiltonian and Poisson bracket.
- Total momentum conservation lost except for Fourier basis due to differing FE spaces.
- Time discretizations: Hamiltonian splitting or Discrete Gradient

*Paper on strong Faraday without smoothing:*

*M. Kraus, K. Kormann, P.J. Morrison, ES - JPP 17*

# Strong Faraday discretisation

- For Particle In Cell discretisation, the most adapted is the Action proposed by Low (1958) with a Lagrangian formulation for the particles.
- The field Lagrangian, splitting between particle and field Lagrangian, using standard non canonical coordinates, reads:

$$L_f[X, V, \mathbf{A}, \phi] = \sum_{\mathrm{s}} \int f_s(\mathbf{z}_0, t_0) L_s(\mathbf{X}(\mathbf{z}_0, t_0; t), \dot{\mathbf{X}}(\mathbf{z}_0, t_0; t)) \mathrm{d}\mathbf{z}_0 + \frac{\epsilon_0}{2} \int |\nabla\phi + \frac{\partial \mathbf{A}}{\partial t}|^2 \mathrm{d}\mathbf{x} - \frac{1}{2\mu_0} \int |\nabla \times \mathbf{A}|^2 \mathrm{d}\mathbf{x}.$$

- Distribution function $f$ expressed at initial time. Particle phase space Lagrangian for species $s$, $L_s$, is of the form $\mathbf{p} \cdot \dot{\mathbf{q}} - H$:

$$L_s(\mathbf{x}, \mathbf{v}, \dot{\mathbf{x}}, t) = (m\mathbf{v} + q\mathbf{A}) \cdot \dot{\mathbf{x}} - (\frac{1}{2} m v^2 + q\phi).$$

# Semi-discrete Vlasov-Maxwell equations

- Discretization obtained by plugging expressions for $f_h, \phi_h, \mathbf{A}_h$ into Lagrangian: $\mathbf{E}_h = \partial_t \mathbf{A}_h - \nabla\phi_h$, $\mathbf{B}_h = \operatorname{curl} \mathbf{A}_h$.
- Dynamical variables: particles positions and velocities, spline coefficients of $\mathbf{E}_h$ and $\mathbf{B}_h$: $\mathbf{u} = (\mathbf{X}, \mathbf{V}, \mathbf{e}, \mathbf{b})^\top$.
- Discrete Hamiltonian:

$$\hat{\mathcal{H}} = \frac{1}{2}\,\mathbf{V}^\top \mathbb{M}_p \mathbf{V} + \frac{1}{2}\,\mathbf{e}^\top M_1 \mathbf{e} + \frac{1}{2}\,\mathbf{b}^\top M_2 \mathbf{b}.$$

- Semi-discrete equations of motion expressed with discrete unknowns

$$\dot{\mathbf{X}} = \mathbf{V} \qquad \dot{\mathbf{x}} = \mathbf{v},$$

$$\dot{\mathbf{V}} = \mathbb{M}_p^{-1}\mathbb{M}_q\big(\mathbb{\Lambda}^1(\mathbf{X})\mathbf{e} + \mathbb{B}(\mathbf{X}, \mathbf{b})\mathbf{V}\big) \qquad \dot{\mathbf{v}} = \frac{q_s}{m_s}\left(\mathbf{E} + \mathbf{v} \times \mathbf{B}\right),$$

$$\dot{\mathbf{e}} = M_1^{-1}\big(\mathbb{C}^\top M_2 \mathbf{b}(t) - \mathbb{\Lambda}^1(\mathbf{X})^\top \mathbb{M}_q \mathbf{V}\big) \qquad \frac{\partial \mathbf{E}}{\partial t} = \operatorname{curl} \mathbf{B} - \mathbf{J},$$

$$\dot{\mathbf{b}} = -\mathbb{C}\mathbf{e}(t) \qquad \frac{\partial \mathbf{B}}{\partial t} = -\operatorname{curl} \mathbf{E}.$$

# Semi-discrete Hamiltonian structure preserved for Vlasov-Maxwell

- Semi-discrete equations of motion have following structure:

$$\dot{\mathbf{u}} = \mathcal{J}(\mathbf{u})\,\nabla_{\mathbf{u}}\hat{\mathcal{H}}(\mathbf{u}).$$

- Poisson matrix:

$$\mathcal{J}(\mathbf{u}) = \begin{pmatrix} 0 & \mathbb{M}_p^{-1} & 0 & 0 \\ -\mathbb{M}_p^{-1} & \mathbb{M}_p^{-1}\mathbb{M}_q\mathbb{B}(\mathbf{X},\mathbf{b})\,\mathbb{M}_p^{-1} & \mathbb{M}_p^{-1}\mathbb{M}_q\mathbb{\Lambda}^1(\mathbf{X})M_1^{-1} & 0 \\ 0 & -M_1^{-1}\mathbb{\Lambda}^1(\mathbf{X})^\top\mathbb{M}_q\mathbb{M}_p^{-1} & 0 & M_1^{-1}\mathbb{C}^\top \\ 0 & 0 & -\mathbb{C}M_1^{-1} & 0 \end{pmatrix}.$$

- Defines semi-discrete Poisson bracket:

$$\{F,G\} = \nabla F^\top \mathcal{J}(\mathbf{u})\nabla G \Rightarrow \frac{\mathrm{d}(F(\mathbf{u}))}{\mathrm{d}t} = \nabla F^\top \dot{\mathbf{u}} = \{F(\mathbf{u}),\mathcal{H}(\mathbf{u})\}.$$

- Some properties:
  - Semi-discrete Poisson bracket satisfies Jacobi identity.
  - $\mathbb{C}\mathbb{G} = 0$, $\mathbb{D}\mathbb{C} = 0$.
  - Discrete Gauss' law: $\mathbb{G}^\top M_1\mathbf{e} = -\mathbb{\Lambda}^0(\mathbf{X})^\top\mathbb{M}_q\mathbb{1}_{N_p}$.

# Time-discretization of Poisson structure

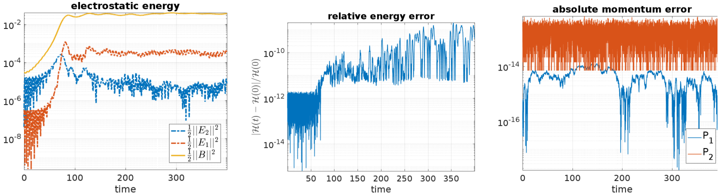
- Poisson form $\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathcal{J}(\mathbf{u})\nabla_{\mathbf{u}}\mathcal{H}$ generalizes symplectic structure
- Thm(Ge, Marsden) Only exact flow preserves both symplectic structure and energy. $\Rightarrow$ need to choose.
- 2 options:
  1. Hamiltonian splitting preserves Poisson structure including Casimirs ($\operatorname{div}\mathbf{B} = 0$, weak Gauss), but only modified energy.
  2. Energy conserving discretisations can be derived based on Discrete Gradient or Average Vector Field methods.

# Weibel instability: Conservation properties.

- PIC (B-Splines): Maximum error in the total energy, Gauss' law, and total momentum until time 500 for simulation with various integrators (Strang splitting $\Delta t = 0.05$).

| Propagator | total energy | Gauss law |
|---|---|---|
| Hamiltonian | 6.9E-7 | 2.1E-13 |
| Boris | 1.3E-9 | 4.8E-4 |
| AVF | 2.1E-16 | 1.1E-6 |
| DiscGrad | 5.9E-11 | 2.2E-15 |

- PIF (Strong Faraday)



# Compressible MHD model

- The compressible ideal MHD model reads

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \mathbf{u}) &= 0,\\
\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) + \nabla p - (\operatorname{curl}\mathbf{B})\times\mathbf{B} &= 0,\\
\frac{\partial \mathbf{B}}{\partial t} - \operatorname{curl}(\mathbf{u}\times\mathbf{B}) &= 0,\\
\frac{\partial p}{\partial t} + \operatorname{div}(p\mathbf{u}) + (\gamma-1)p\operatorname{div}\mathbf{u} &= 0
\end{aligned}
$$

- Used in particular to study instabilities in edge of Tokamak
- Very small viscous and resistive terms need to be added.
- Implicit or semi-implicit discretisations are needed.
- JOREK code based on $C^1$ Bezier splines. New prototype based on FEEC with B-Splines.

# Fast solvers for implicit MHD

*A. Ratnani, M. Mazza in collaboration with C. Manni, H. Speleers (Rome) and S. Serra Capizzano (Como)*

- Models: Reduced MHD, 2D incompressible MHD, full MHD
- Newton - Krylov solver. Time stepping algorithms: splitting, jacobian free.
- Optimal Multigrid for B-Splines for Elliptic problems
- Optimal preconditioning for the (B-Splines) mass matrices (GLT)
- Development of new GLT preconditioner for $H(curl)$- problems needed for MHD.
- GLT is a new theory that allows the study and designing of preconditioners for spline Finite Elements.
  - Spectral properties are described through a notion of the symbol
  - Associated B-Splines symbols have been derived and studied

# Generalized Locally Toeplitz (GLT) theory

## Poisson equation

- The stiffness matrix is ill-conditioned in the low frequencies.
  Classical problem solved by MG preconditioning.
- The stiffness matrix is also ill-conditioned in the high frequencies:
  Problem solvable by GLT theory through a post-smoother.
- the post-smoother is based on a Kronecker product

## Elliptic $H(curl)$-problems

- As for Poisson equation, the matrix presents two kinds of pathologies in low and high frequencies
- Non trivial kernel (infinite in the continuous level)
  $\Longrightarrow$ an Auxiliary Spaces Method is being studied to derive an optimal solver with respect to the physical parameters.

# Fast solvers and Preconditioning

- Tolerance is set to $10^{-12}$
- maximum number of V-Cycles : 1000

| Mesh | Degree 1 | 2 | 3 | 4 | 5 | 6 | Method |
|---|---|---|---|---|---|---|---|
| $8 \times 8 \times 8$ | 11 | 43 | 200 | – | – | – | MG |
| | 2 | 3 | 3 | 5 | 8 | 18 | MG+GLT |
| $16 \times 16 \times 16$ | 16 | 26 | 193 | – | – | – | MG |
| | 4 | 6 | 6 | 7 | 10 | 15 | MG+GLT |
| $32 \times 32 \times 32$ | 15 | 15 | 144 | 474 | – | – | MG |
| | 7 | 8 | 9 | 12 | 18 | 25 | MG+GLT |

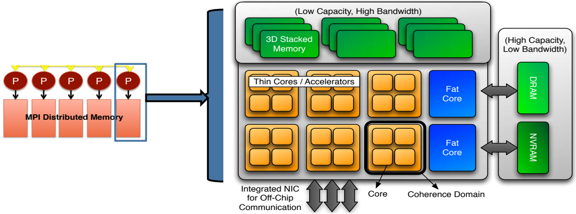
Table: Number of required cycles until convergence for 3D Poisson solver on a cube.

# Software environment for Geometric Numerical Methods and more

- Develop a simple to use, robust and modular framework for going from a physics model first for rapid prototyping and if desired to an efficient HPC code for production.
- Automatically generate stable and consistent discrete model from continuous model
- Enforce main physics properties, e.g.
  - Conservation of energy
  - Increase of entropy
- Simple enough language based on geometric concepts from physics which are automatically transformed into a numerical model by the code for rapidly investigating new physics
- Framework for transforming quickly prototype code into HPC code that runs efficiently on modern supercomputers.

# Issues

- Find a framework to guarantee stability and consistency with physics model.
- Domain specific language for translating geometric physics model into code. Open Source for quality control and reproducibility.
- Complexity and fast changes in modern computing architectures makes hand tuning very cumbersome and time consuming.




- Relevant other projects: FEniCS, Firedrake: Finite Element Method for PDEs

# Domain specific language (DSL)

- Develop specific language based on Python describing geometric structure of physics at the continuous level
- Automatically transformed into a numerical model by the code for rapidly investigating new physics
- DSL should also enable (partly) automatic generation of portable optimized code for modern computer architectures.
- Can be embedded into existing exascale framework, *e.g.* WARP-X based on AMReX for PIC simulations.
- Software environment developed by Ahmed Ratnani and collaborators (Y. Güçlu, S. Hadjout, J. Lakhlili, ...) based on Python and sympy: psydac, sympde, symdec, GeLaTo, pyccel.

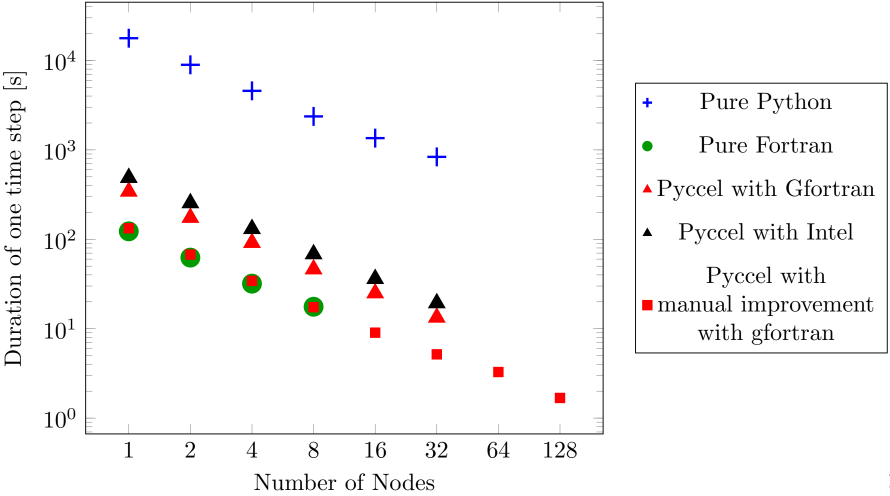
# Pyccel

- Pyccel, a **Fortran** static compiler for **Python** for scientific High-Performance Computing
- Pyccel competes well with the existing solutions
- allows to use mpi4py, blas/lapack, fft, etc
- converts a Python code into a symbolic expression/tree
- arithmetic/memory complexity

**Ongoing and Future work**

- Shared memory parallelism (OpenMP, OpenACC)
- Task Based Parallelism
- Additional decorators
  - Cache optimization
  - Explicit memory management

# Pyccel




1

---

[1] 32 processes per node

# Accelerating Python codes: Benchmarks

**Rosen-Der**

| Tool | Python | Cython | Numba | Pythran | Pyccel-gcc | Pyccel-intel |
|---|---|---|---|---|---|---|
| Timing ($\mu s$) | 229.85 | 2.06 | 4.73 | 2.07 | **0.98** | **0.64** |
| Speedup | – | $\times$ 111.43 | $\times$ 48.57 | $\times$ 110.98 | $\times$ **232.94** | $\times$ **353.94** |

**Black-Scholes**

| Tool | Python | Cython | Numba | Pythran | Pyccel-gcc | Pyccel-intel |
|---|---|---|---|---|---|---|
| Timing ($\mu s$) | 180.44 | 309.67 | 3.0 | 1.1 | **1.04** | $\mathbf{6.56\ 10^{-2}}$ |
| Speedup | – | $\times$ 0.58 | $\times$ 60.06 | $\times$ 163.8 | $\times$ **172.35** | $\times$ **2748.71** |

**Laplace**

| Tool | Python | Cython | Numba | Pythran | Pyccel-gcc | Pyccel-intel |
|---|---|---|---|---|---|---|
| Timing ($\mu s$) | 57.71 | 7.98 | $6.46\ 10^{-2}$ | $6.28\ 10^{-2}$ | $\mathbf{8.02\ 10^{-2}}$ | $\mathbf{2.81\ 10^{-2}}$ |
| Speedup | – | $\times$ 7.22 | $\times$ 892.02 | $\times$ 918.56 | $\times$ **719.32** | $\times$ **2048.65** |

**Growcut**

| Tool | Python | Cython | Numba | Pythran | Pyccel-gcc | Pyccel-intel |
|---|---|---|---|---|---|---|
| Timing ($s$) | 54.39 | $1.02\ 10^{-1}$ | $4.67\ 10^{-1}$ | $8.57\ 10^{-2}$ | $\mathbf{6.27\ 10^{-2}}$ | $\mathbf{6.54\ 10^{-2}}$ |
| Speedup | – | $\times$ 532.37 | $\times$ 116.45 | $\times$ 634.32 | $\times$ **866.49** | $\times$ **831.7** |

# Automatic code generation

- Use Pyccel AST to represent the assembly procedure over one element and the parallel loop over the elements
- Starting from a Bilinear/ Linear form or an Integral expression
  - Generate the associated Python code
  - Convert Python to Fortran using Pyccel
  - Parallel Matrix-Vector multiplication using `MPI_Cart` and subcommunicators + Lapack
  - Matrix-free approach

# Conclusions and future steps

- Magnetic Fusion Simulations need advanced numerical and mathematical models.
- Structure preserving methods starting from Yee and Arakawa and now extended to Splines-FEEC have proven very successful and need to be developed further.
- Progress in aligned IGA grid generation. Still open issues, in particular $C^1$ continuity at X-point and tokamak wall.
- Fast multigrid solvers needed for elliptic and implicit problems. First steps with GLT very promising. Need to be extended to full problem and physics codes.