PEPC: an Open-source Multi-physics Framework for Mesh-free Simulation of N-body Systems

SIAM CSE25, 3-7 March 2025

P. Gibbon

JÜLICH Forschungszentrum

Shaping Change

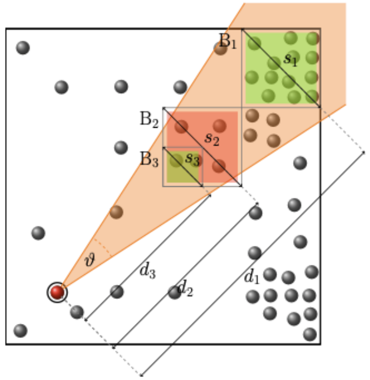
Barnes-Hut tree code: accelerated N-body field-solver

- Goal: Compute geometrically complex N-body interaction, many particles, high dynamic range in density distribution:

$$\phi(r) \propto \frac{1}{r}$$

- Nature of the interaction enables multipole expansion:

$$\phi(r) = \frac{Q}{r} + \frac{x_j D^j}{r^3} + \frac{1}{2}\frac{x_j x_k Q^{jk}}{r^5} + \cdots \quad \text{for } j,k = 1,2,3$$



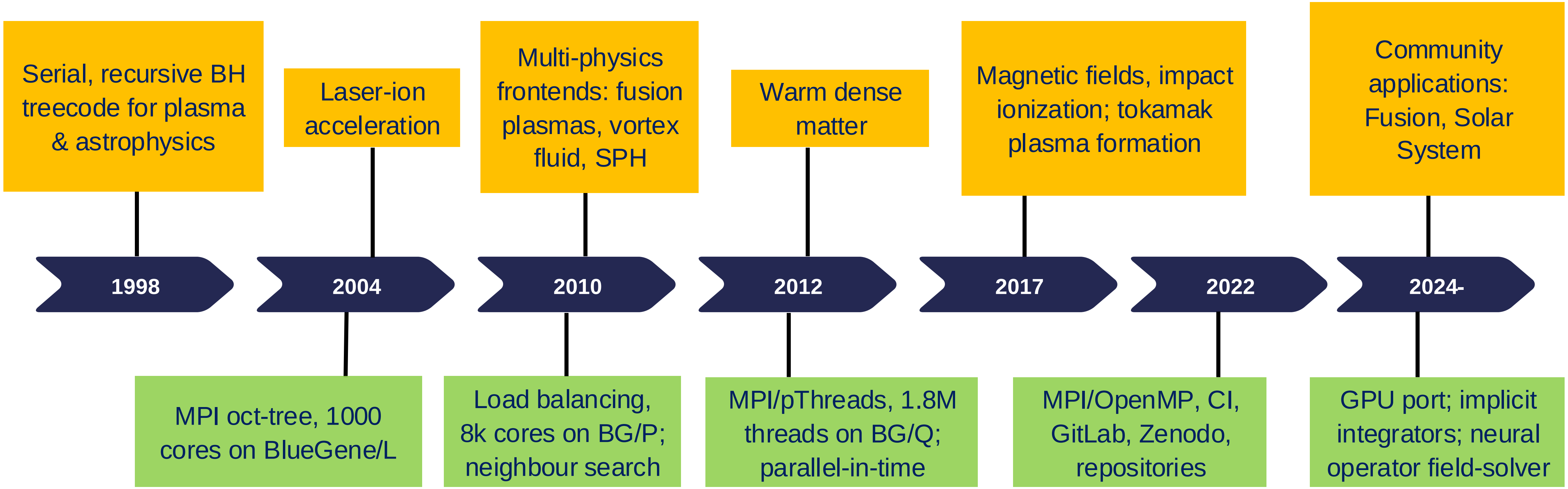
- Multipole acceptance criterion (MAC) to decide if multipoles need resolving
- Helped by a tree-structure to 'combine particles spatially', efficient organisation, load-balancing

Net result: reduce complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\log(N))$

J. Barnes & P. Hut, Nature (1986)

PEPC tree code: a potted history

Developments and applications at the Jülich Supercomputing Centre

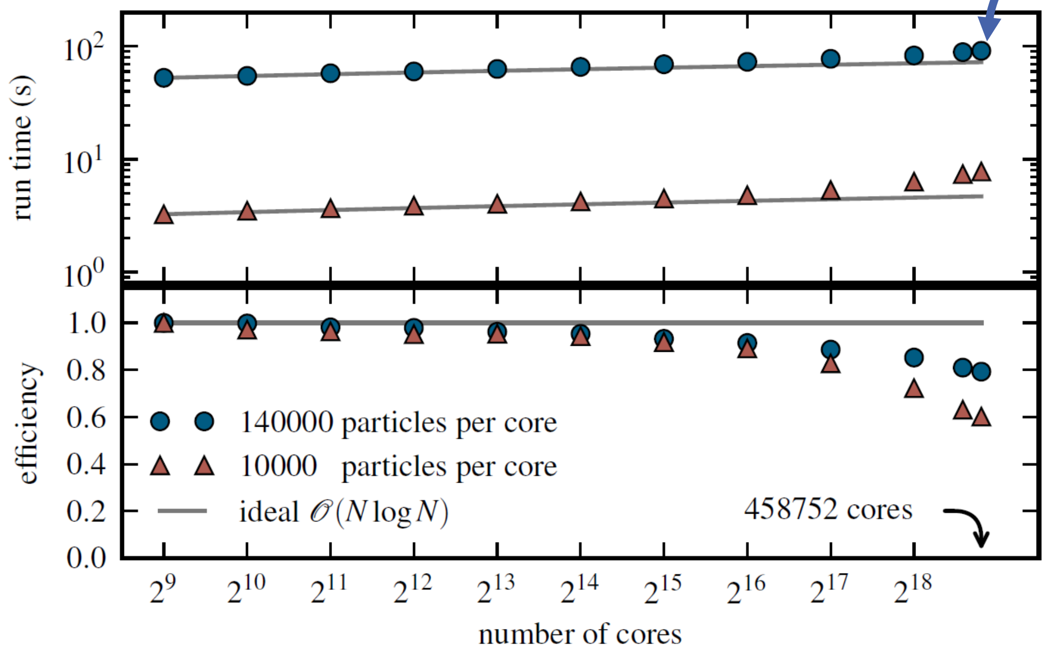


Performance tuning

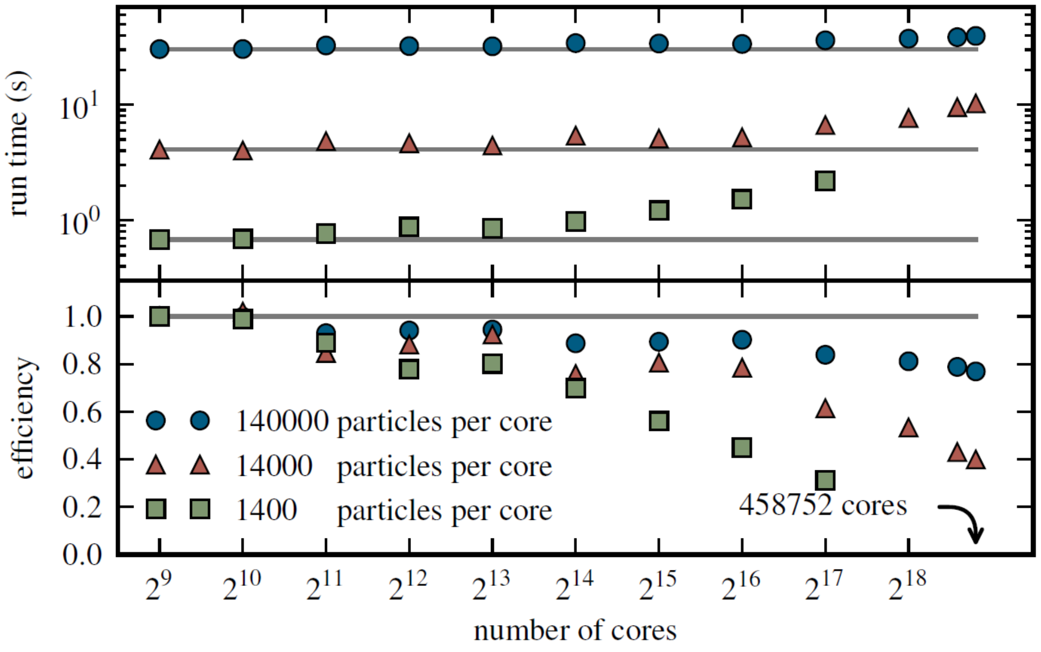
Parallel scalability of single & dual tree algorithms on JUQUEEEN

64 billion particles

Hybrid O(*NlogN*) algorithm



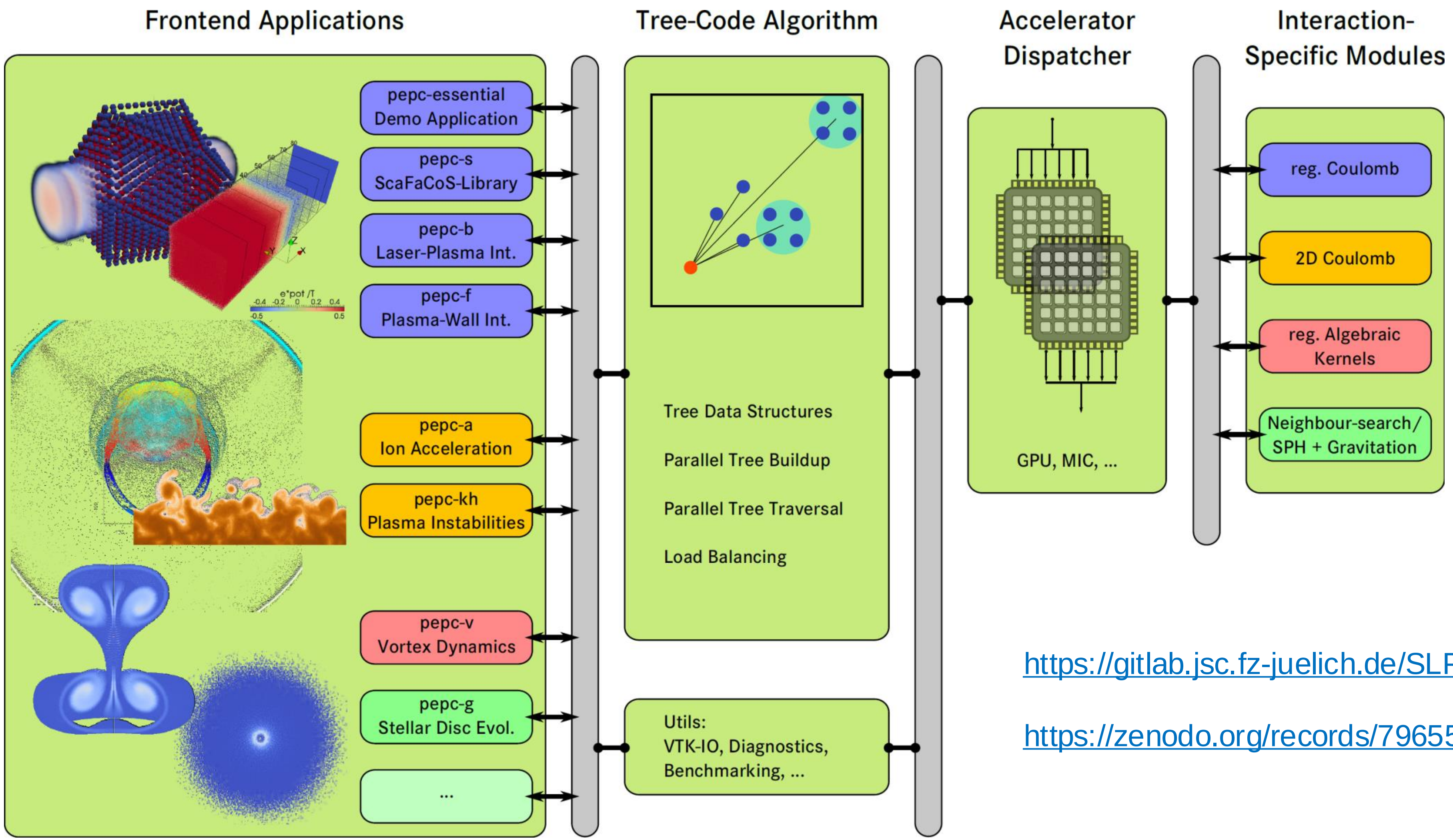
Dual-tree *O(N)* algorithm



B. Steinbusch, M.-L. Henkel, M. Winkel, PG, Advances in Parallel Computing 27, (2016)

JÜLICH Forschungszentrum

PEPC Framework

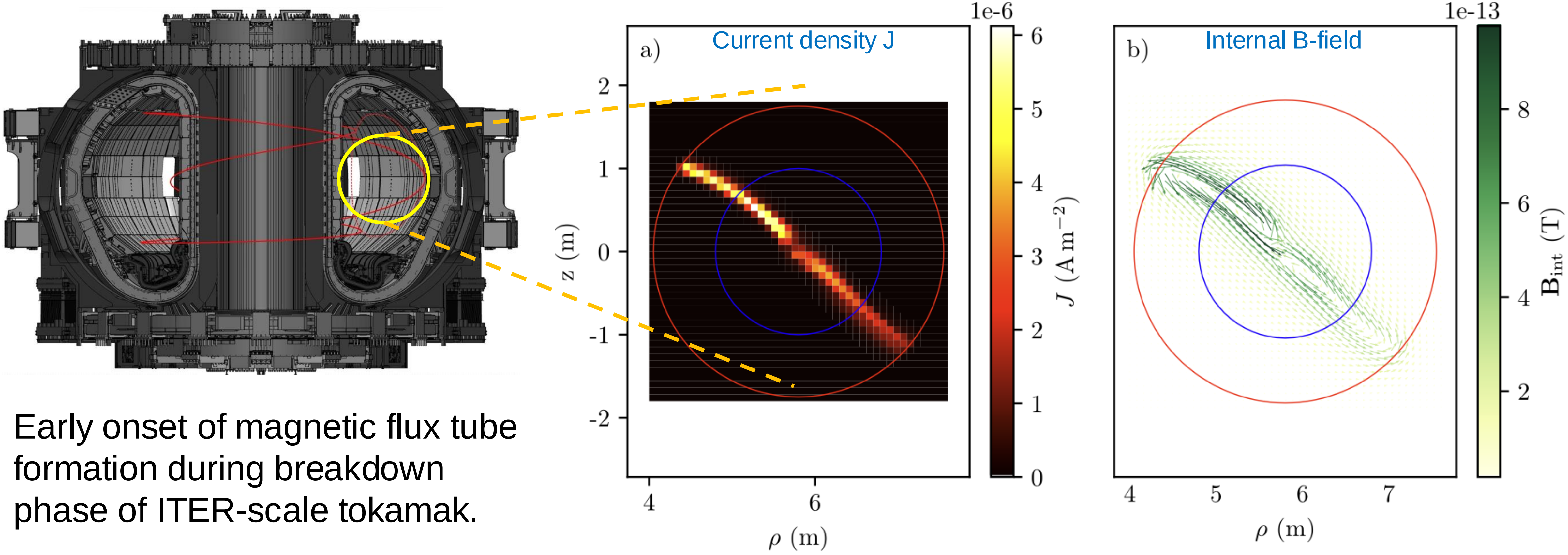


https://gitlab.jsc.fz-juelich.de/SLPP/pepc/pepc

https://zenodo.org/records/7965549

Application highlight 1

Ab initio modeling of tokamak breakdown: dynamic particle populations

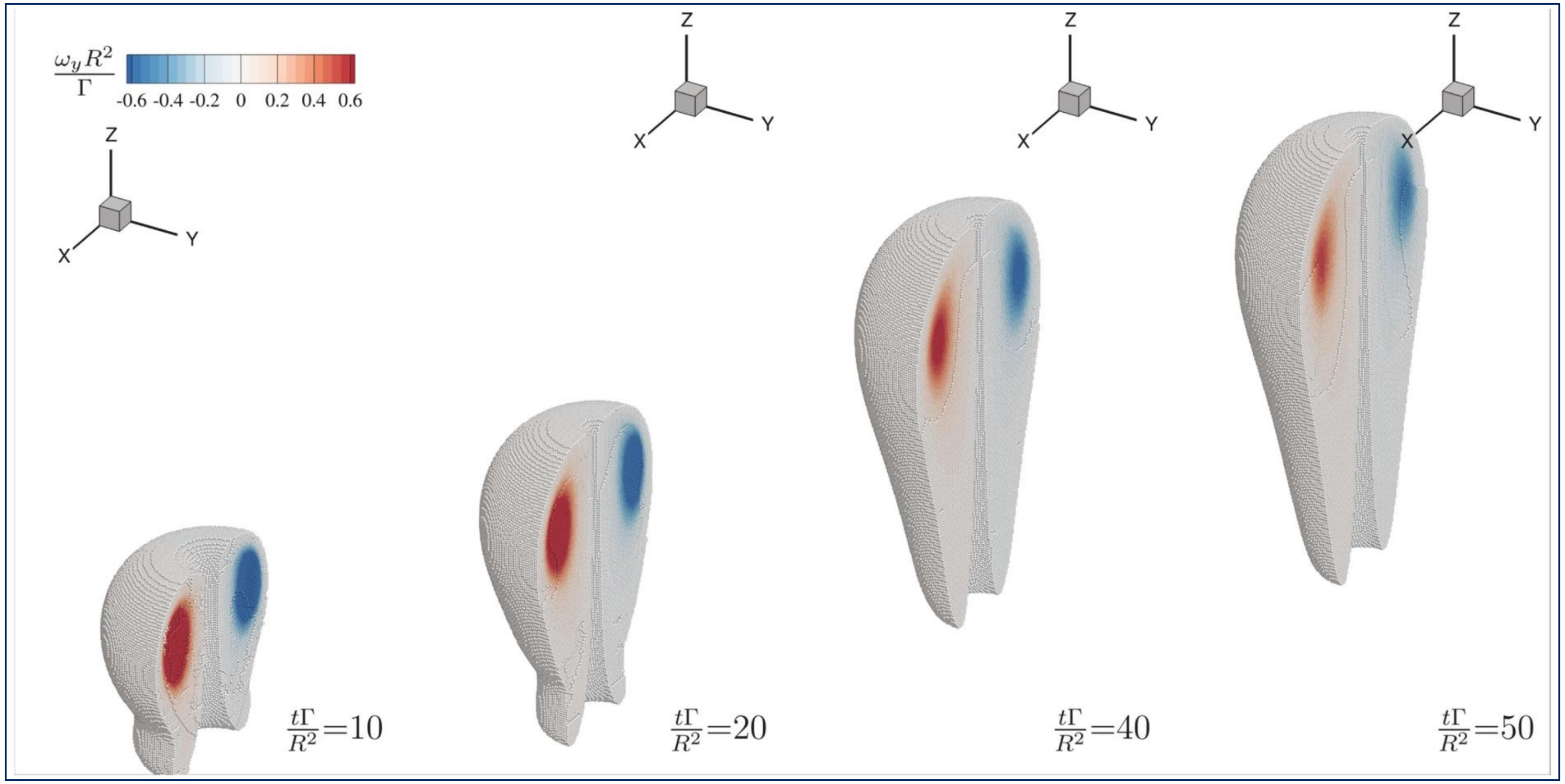


Early onset of magnetic flux tube formation during breakdown phase of ITER-scale tokamak.

[J. Chew, PG, D. Brömmel, T. Wauters, Y. Gribov & P. de Vries, Nucl. Fusion (2024)]

Application highlight 2

Diffused Vortex Hydrodynamics: PEPC-DVH simulation of ring with 1.3 M → 90 M particles



[D. Durante, S. Marrone, D. Brömmel, R. Speck & A. Colagrossi, Math. Comp. Sim. (2024)]

Mesh-free Darwin treecode

$$\nabla \cdot \mathbf{E}^{irr} = 4\pi\rho$$

$$\nabla \times \mathbf{E}^{sol} = -\frac{1}{c}\frac{\partial \mathbf{B}}{\partial t}$$

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

$$\nabla \times \mathbf{B} = \frac{4\pi}{c}\mathbf{J} + \frac{1}{c}\frac{\partial \mathbf{E}^{irr}}{\partial t} + \cancel{\frac{1}{c}\frac{\partial \mathbf{E}^{sol}}{\partial t}}$$

Neglect EM radiation

$$\mathbf{A}(\mathbf{x}_i, t) = \frac{1}{2c}\sum_{j\neq i} q_j \mathbf{v}_j \left[-\tilde{A}_j + 1 - \log\left(r_{ij}^2 + \varepsilon^2\right)\right]$$

$$+\frac{1}{c}\sum_j q_j \tilde{A}_j \frac{\mathbf{x}_i - \mathbf{x}_j(t)}{||\mathbf{x}_i - \mathbf{x}_j(t)||} \times \left[\mathbf{v}_j \times \frac{\mathbf{x}_i - \mathbf{x}_j(t)}{||\mathbf{x}_i - \mathbf{x}_j(t)||}\right]$$

$$\mathbf{E}^{irr}(\mathbf{x}_i) = -\nabla\varphi = 2\sum_{j\neq i} q_j \frac{\mathbf{x}_i - \mathbf{x}_j}{r_{ij}^2 + \varepsilon^2}$$

$$\mathbf{B}(\mathbf{x}_i) = \nabla \times \hat{\mathbf{A}} = -\frac{2}{c}\sum_{j\neq i} q_j \frac{(\mathbf{x}_i - \mathbf{x}_j) \times \mathbf{v}_j}{r_{ij}^2 + \varepsilon^2}$$

- Formulation of 2D *elliptic* field solver + discrete, *smoothed* particles → *O(N* log *N) tree code*
- Point-particle result recovered for $\varepsilon \to 0$
- A & E^{sol} are divergence-free *by design*

Darwin code: 3D potentials

Vector potential:

$$\mathbf{A}(\mathbf{x}_i) = \frac{1}{c\varepsilon}\sum_{j\neq i} q_j \mathbf{v}_j \left[\frac{\varepsilon}{\sqrt{r_{ij}^2+\varepsilon^2}} - \frac{\varepsilon\sqrt{r_{ij}^2+\varepsilon^2}}{r_{ij}^2} + \frac{\varepsilon^3}{2r_{ij}^3}\sinh^{-1}(r_{ij}/\varepsilon)\right] +$$

$$+\frac{1}{c}\sum_{j\neq i} q_j \left[\mathbf{v}_j\cdot(\mathbf{x}_i-\mathbf{x}_j)\right]\frac{\mathbf{x}_i-\mathbf{x}_i}{r_{ij}^4}\left[\frac{3\varepsilon^2+r_{ij}^2}{2\sqrt{r_{ij}^2+\varepsilon^2}} - \frac{3\varepsilon^2}{2r_{ij}}\sinh^{-1}(r_{ij}/\varepsilon)\right]$$

Corresponding magnetic field:

$$\mathbf{B}(\mathbf{x}_i) = -\frac{1}{c}\sum_{j\neq i} q_j \frac{(\mathbf{x}_i-\mathbf{x}_j)\times\mathbf{v}_j}{(r_{ij}^2+\varepsilon^2)^{3/2}}$$

- Recover point-particle result for $\varepsilon\rightarrow 0$
- $\nabla . A = 0$ *by design*
  $\rightarrow$ no divergence cleaning necessary!
- Applications: magnetic reconnection, fusion devices, plasma thrusters

M. Masek, PG, IEEE Trans. Plasma Sci. (2010)

Integration scheme

Darwin fact of life: advancing fields via time differencing is forbidden!

Use *canonical momenta* instead of velocities:

$$\mathbf{P}_i = \frac{\partial \mathcal{L}_i}{\partial \mathbf{v}_i} = m_i \gamma_i \mathbf{v}_i + \frac{q_i}{c}\mathbf{A}(\mathbf{x}_i)$$

$$\mathcal{L}_i = -\frac{m_i c^2}{\gamma_i} - q_i \varphi(\mathbf{x}_i) + \frac{q_i}{c}\mathbf{v}_i \cdot \mathbf{A}(\mathbf{x}_i)$$

Then advance particle positions & momenta with semi-implicit '*Asymmetric Euler Method*':

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t \mathbf{v}_i^n$$

$$\mathbf{P}_i^{n+1} = \mathbf{P}_i^n + q_i \Delta t \left[-\nabla \varphi_i^{n+1} + \frac{1}{c}\nabla\left(\mathbf{A}_i^{n+1} \cdot \mathbf{v}_i^n\right)\right]$$

L. Siddi, G. Lapenta, PG, Phys. Plasmas (2017)

Integration scheme refinements

1. Improved Asymmetric Euler (IAEM, Christlieb, Sands & White, 2024)

$$\boldsymbol{P}^{n+1} = \boldsymbol{P}^n + q\Delta t[\nabla \boldsymbol{A}^{n+1} \cdot \boldsymbol{v}^*]; \qquad \boldsymbol{v}^* \approx \boldsymbol{v}^{n+1} \approx 2\boldsymbol{v}^n - \boldsymbol{v}^{n-1}$$

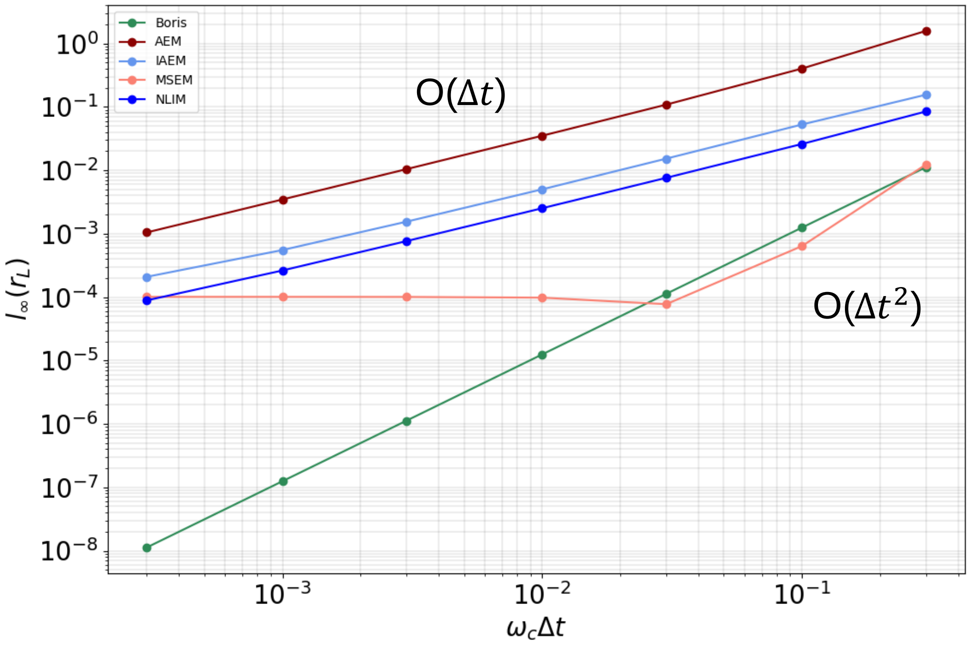
2. Midstep-Euler (MSEM)

$$\boldsymbol{x}^{n+\frac{1}{2}} = \boldsymbol{x}^n + \frac{\Delta t}{2}\boldsymbol{v}^n$$
$$\boldsymbol{P}^{n+\frac{1}{2}} = \boldsymbol{P}^n + q\frac{\Delta t}{2}\left[\nabla \boldsymbol{A}^{n+\frac{1}{2}} \cdot \boldsymbol{v}^{*\frac{1}{2}}\right]; \qquad \boldsymbol{v}^{*\frac{1}{2}} \approx \boldsymbol{v}^{n+\frac{1}{2}} \approx \frac{3}{2}\boldsymbol{v}^n - \frac{1}{2}\boldsymbol{v}^{n-1}$$
$$\ldots$$

3. Nielsen-Lewis implicit (NLIM, 1971)

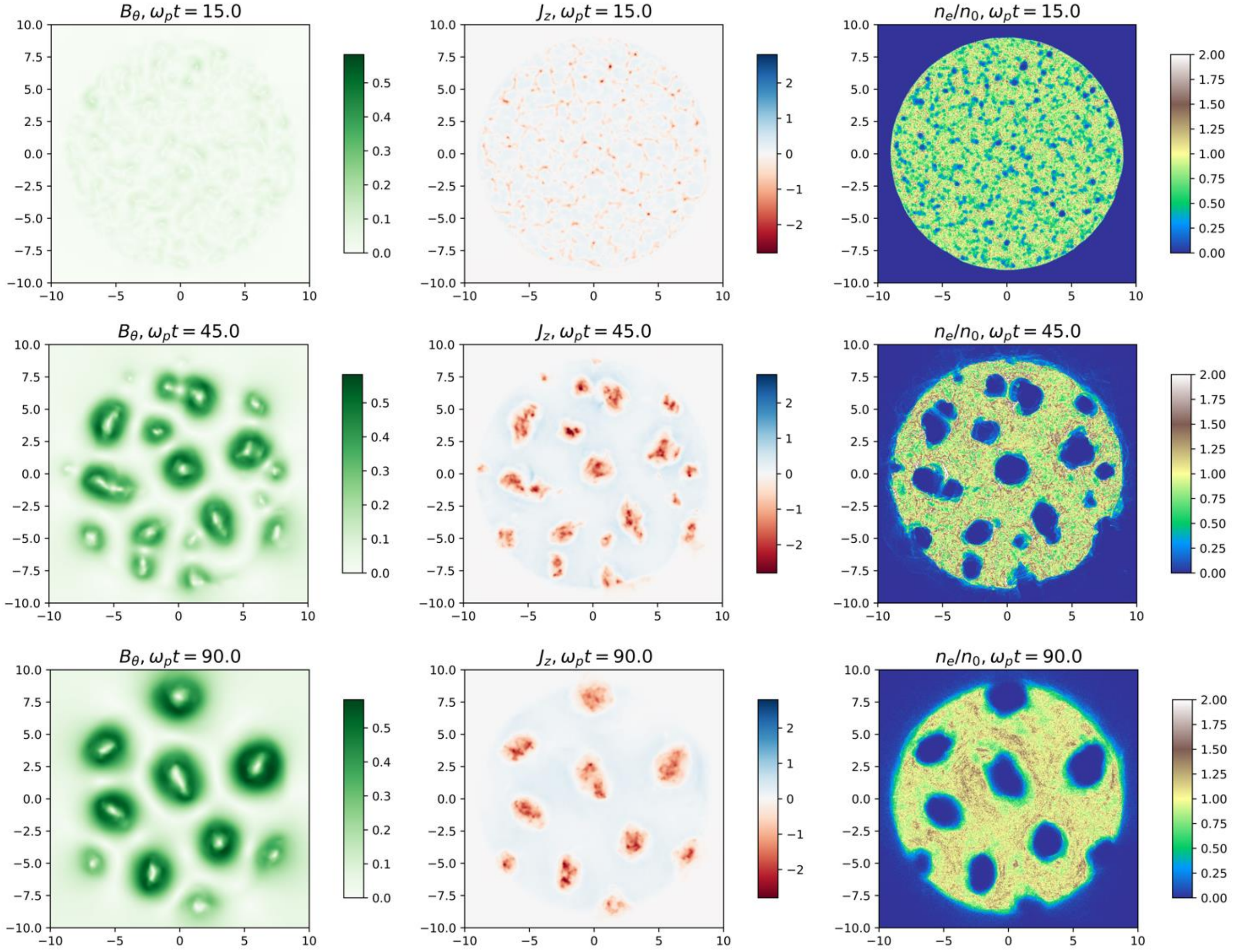
$$\boldsymbol{P}^{n+\frac{1}{2}} = \boldsymbol{P}^{n-\frac{1}{2}} + q\Delta t\, \nabla \boldsymbol{A}^n \cdot \left[\frac{1}{2}\left(\boldsymbol{P}^{n+\frac{1}{2}} + \boldsymbol{P}^{n-\frac{1}{2}}\right) - q\boldsymbol{A}^n\right]$$
$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \Delta t\left[\boldsymbol{P}^{n+\frac{1}{2}} - q\boldsymbol{A}^{n+\frac{1}{2}}\right]$$

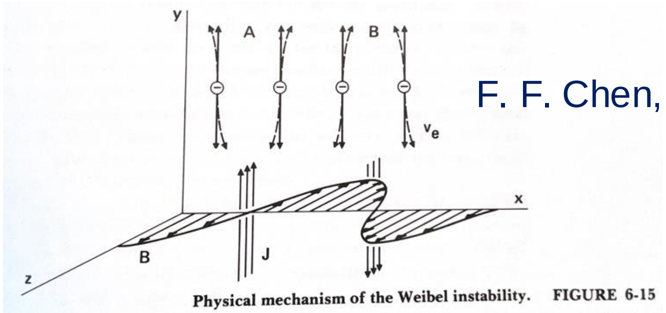
Larmor convergence test ($\nabla\emptyset = 0$)



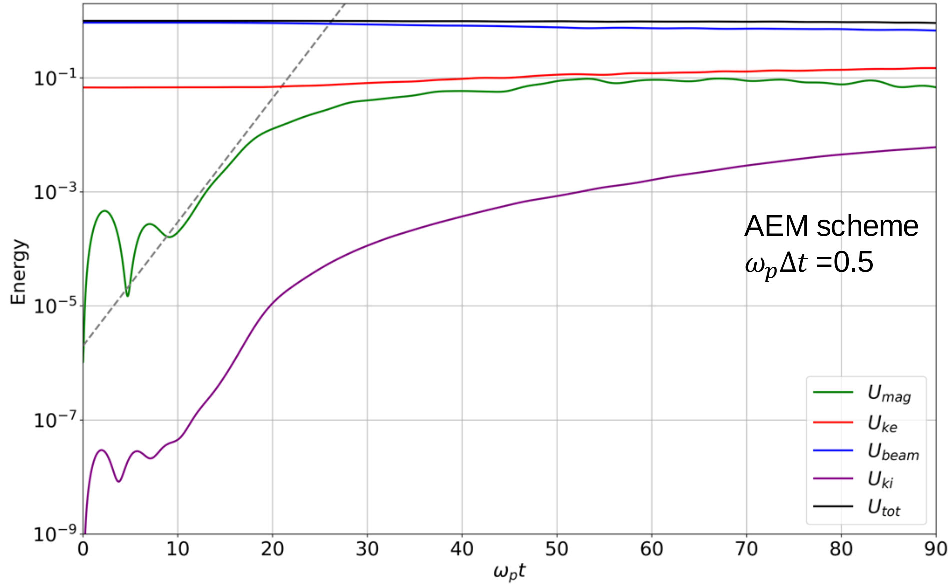
JÜLICH Forschungszentrum

Weibel instability benchmark





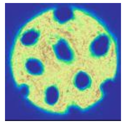
F. F. Chen, 2nd Ed. (1983)

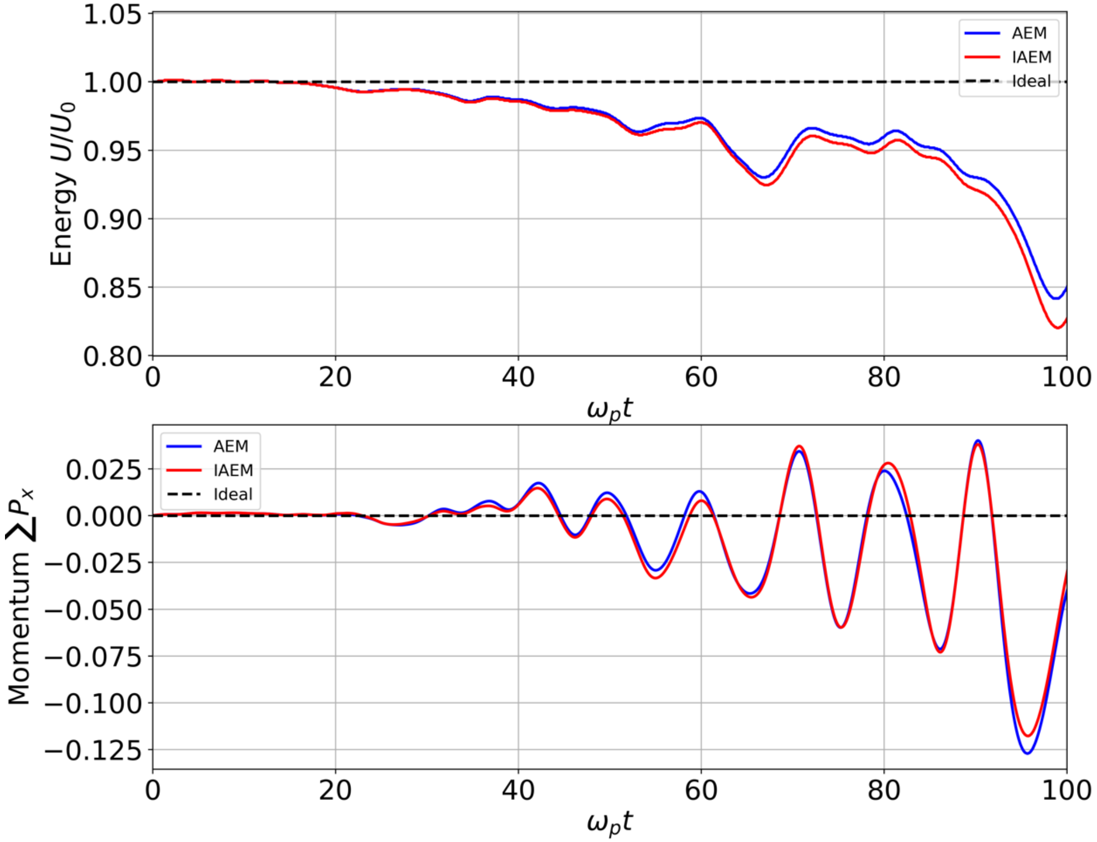


L. Siddi, G. Lapenta, PG, Phys. Plasmas (2017)

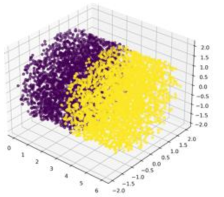
Energy & momentum conservation

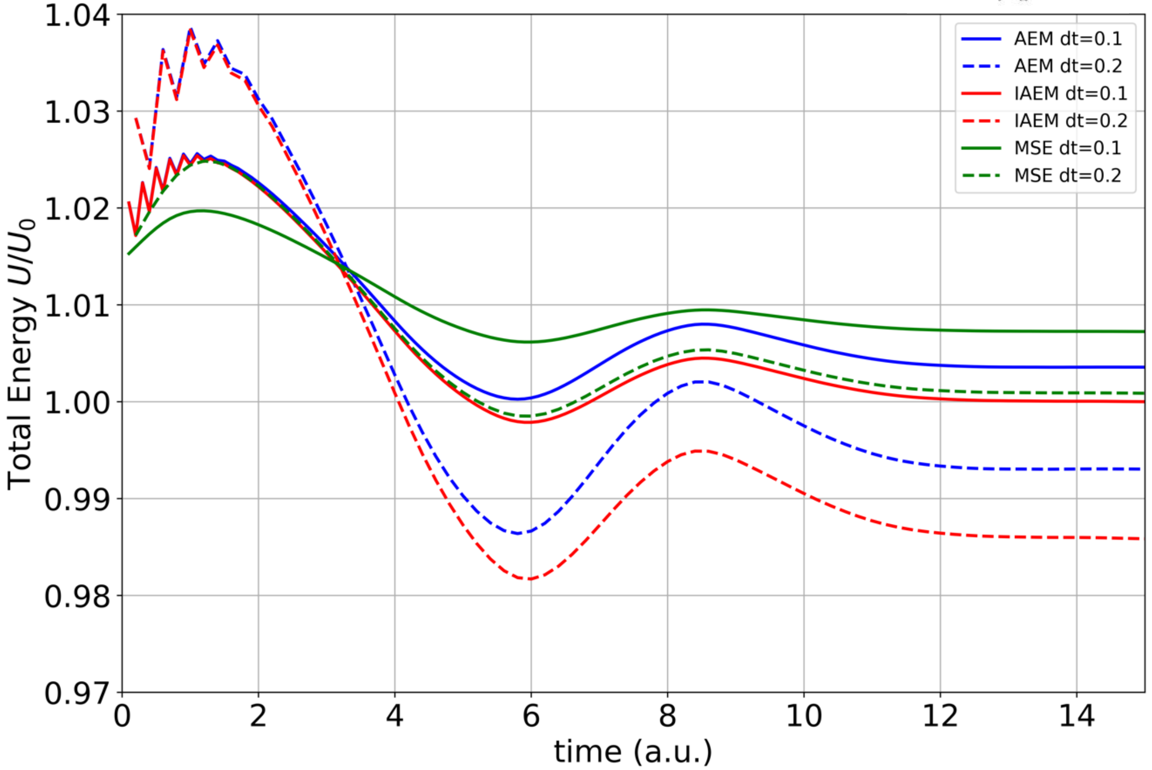
Weibel benchmark





3D colliding beam test

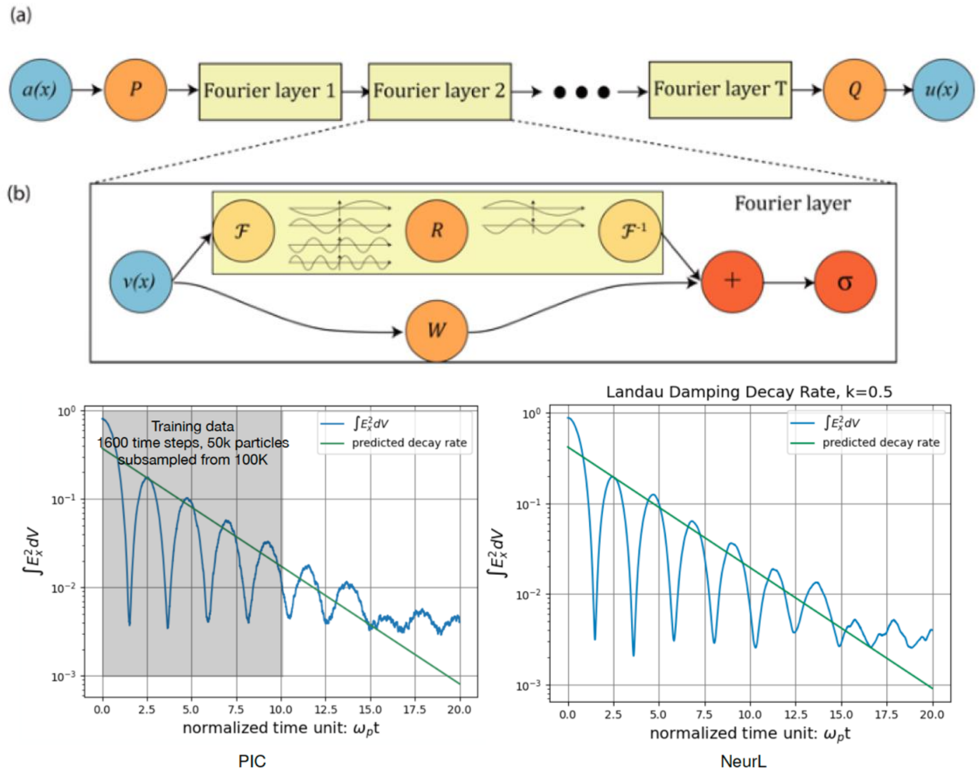




JÜLICH Forschungszentrum

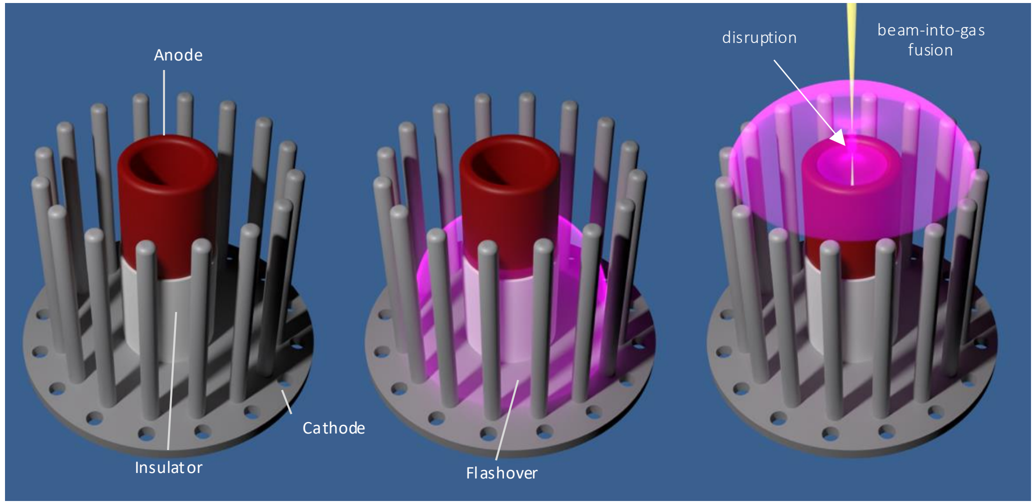
Outlook: future projects

Neural operators as field solver



Sriram Muralikrishnan, (Coupled Problems, 2025)

Implicit integrators: novel fusion devices



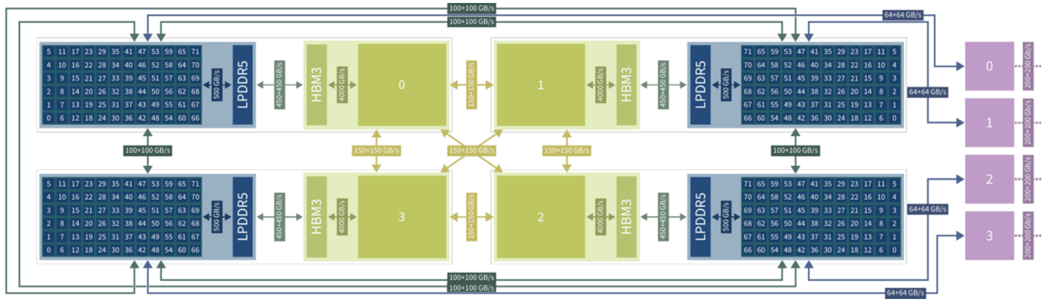
Time evolution (~ 3 ns) of dense plasma focus

Steve Lisgo, ELPIS Fusion

Exascale alignment: GPU Strategy

- **Hardware: 1st European Exaflop system JUPITER > June 2025**

  - 6000 Booster nodes with 4 GH200 superchips
  - 72-core Grace CPU + Hopper GPU
  - NVlink4 GPU-GPU data @ 150 GB/s



- **Paradigm (portability vs performance): OpenACC, OpenMP, Kokkos, CUDA**

- **Two main strategies:**
  - Traversal on CPU; forces on GPU (ChaNGa 2015, FLASH4 2016)
  - GPU traversal + force calculation on GPU (eg: Bonsai 2012, ChaNGa 2019)

Summary

- Open-source mesh-free framework PEPC for long-range N-body problems
- https://gitlab.jsc.fz-juelich.de/SLPP/pepc/pepc
- Good scalability with hybrid MPI/OpenMP programming model
- Darwin plasma model (2D & 3D) with robust (but imperfect!) integrator

Future enhancements:

- Fully implicit, structure-preserving integrator?
- Refactoring for CPU-GPU architectures
- Neural operator as field solver

Thanks to:

Dirk Brömmel, Junxian Chew, Robert Speck, Sriram Muralikrishnan, Lorenzo Siddi, Matthias Winkel, Benedikt Steinbusch, Martin Masek