Conference or Workshop Item:
Bennett, Keith, Morris, Stuart, Goffrey, T. et al. (4 more authors) (2024) Re-engineering the EPOCH PIC code in C++. In: High Power Laser Christmas Meeting, 16-18 Dec 2024, Abingdon.

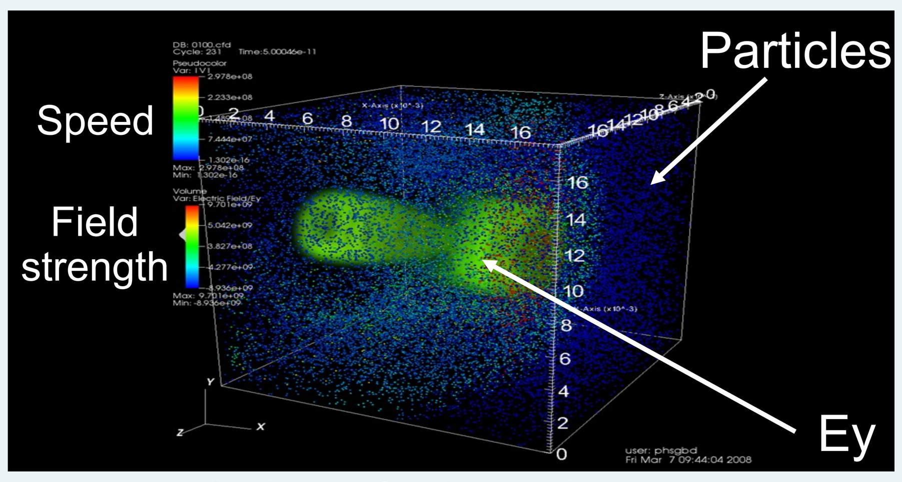
Re-engineering the EPOCH PIC code in C++

K. Bennett, S. Morris, T. Goffrey, T. Arber - *University of Warwick*
S. A. Wright, A. Naden, S. Bulut - *University of York*

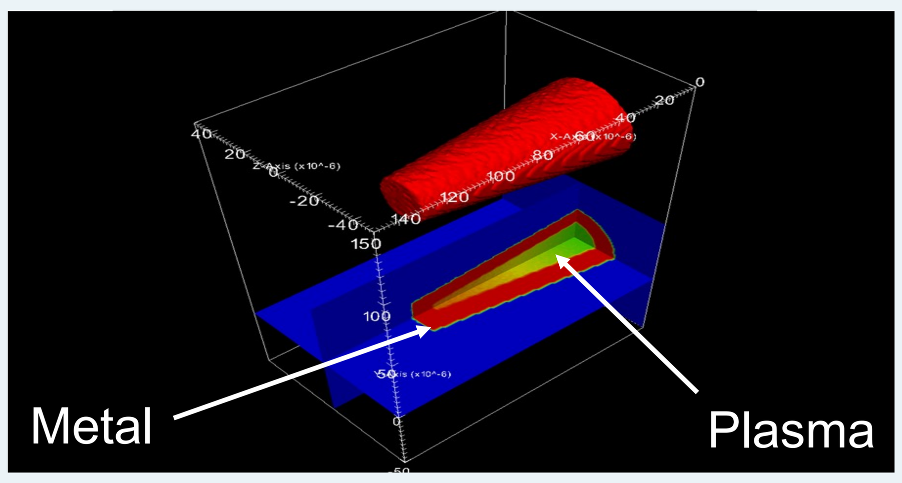
It's new!

The Extendable PIC Open Collaboration (EPOCH)

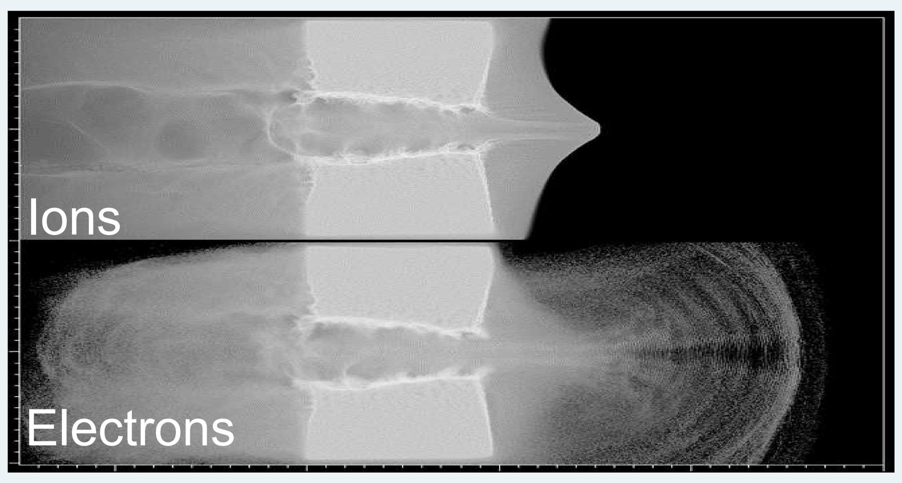
- EPOCH [1] is a relativistic EM-PIC code, which takes a simple text-file as input.
- EPOCH is written in F95 with MPI and scales to 32,000 cores on ARCHER2
- It has evolved since 2015 to include QED, radiation, ionisation, collisions, and cylindrical geometry.
- The code has been cited in over 1300 publications



3D simulation of a laser in underdense plasma (wakefield)



Setup of a 3D plasma-filled metal cone target for fast ignition simulation



2D simulation of a foil burn-through experiment (species density plotted)

C++ upgrade

- Advantages:
  - C++ templating: allows 1D, 2D and 3D in the same code
  - Derived classes: easily add features, like new physics packages
  - Modern HPC tools:
    - kokkos • CPU/GPU portability library
    - ADIOS • High performance parallel data input/output
    - openPMD • Portable particle/mesh data conventions
- Will be easier to implement run-time diagnostics
- Reads the same input decks as the FORTRAN code
- New documentation and examples provided

C++ structure

- Code is structured into classes, stored in a main Simulation class
- Various structures used to hold multiple derived lists
- Derived lists use polymorphism to allow easy extendibility

Deck class
- Stores lines from input deck
- Maths parser evaluates terms

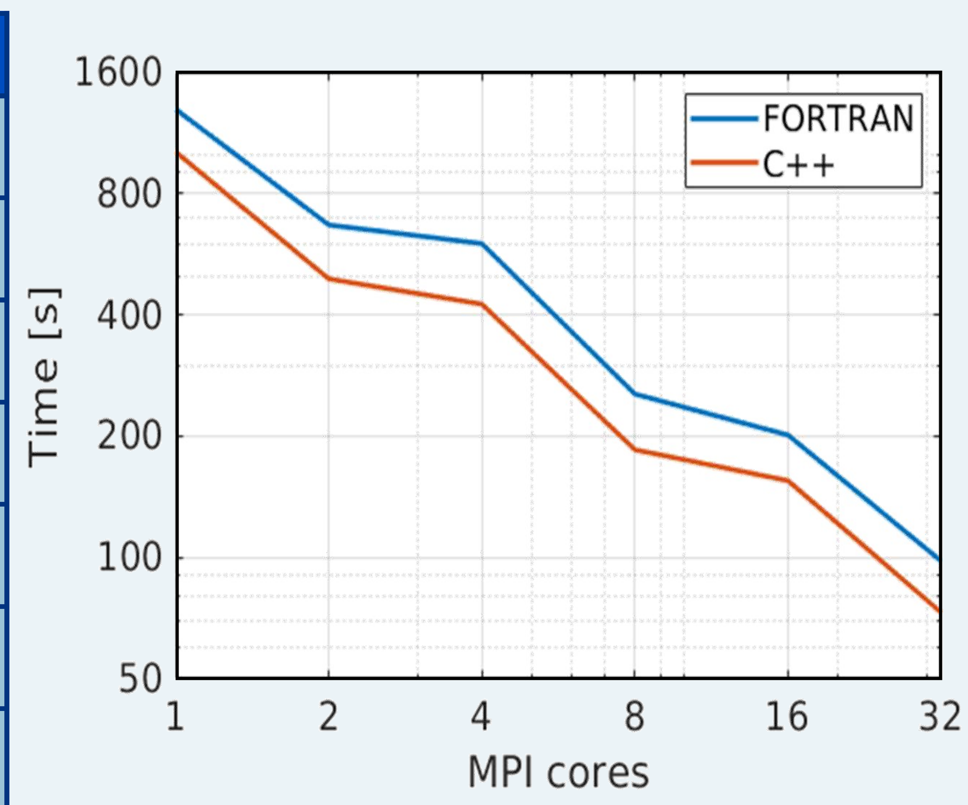
physics_package_manager class
- Stores physics package list
- Each package in same format

Simulation class
- Holds fields and species list
- Runs simulation
- Templated for 1D, 2D, 3D

Scaling tests

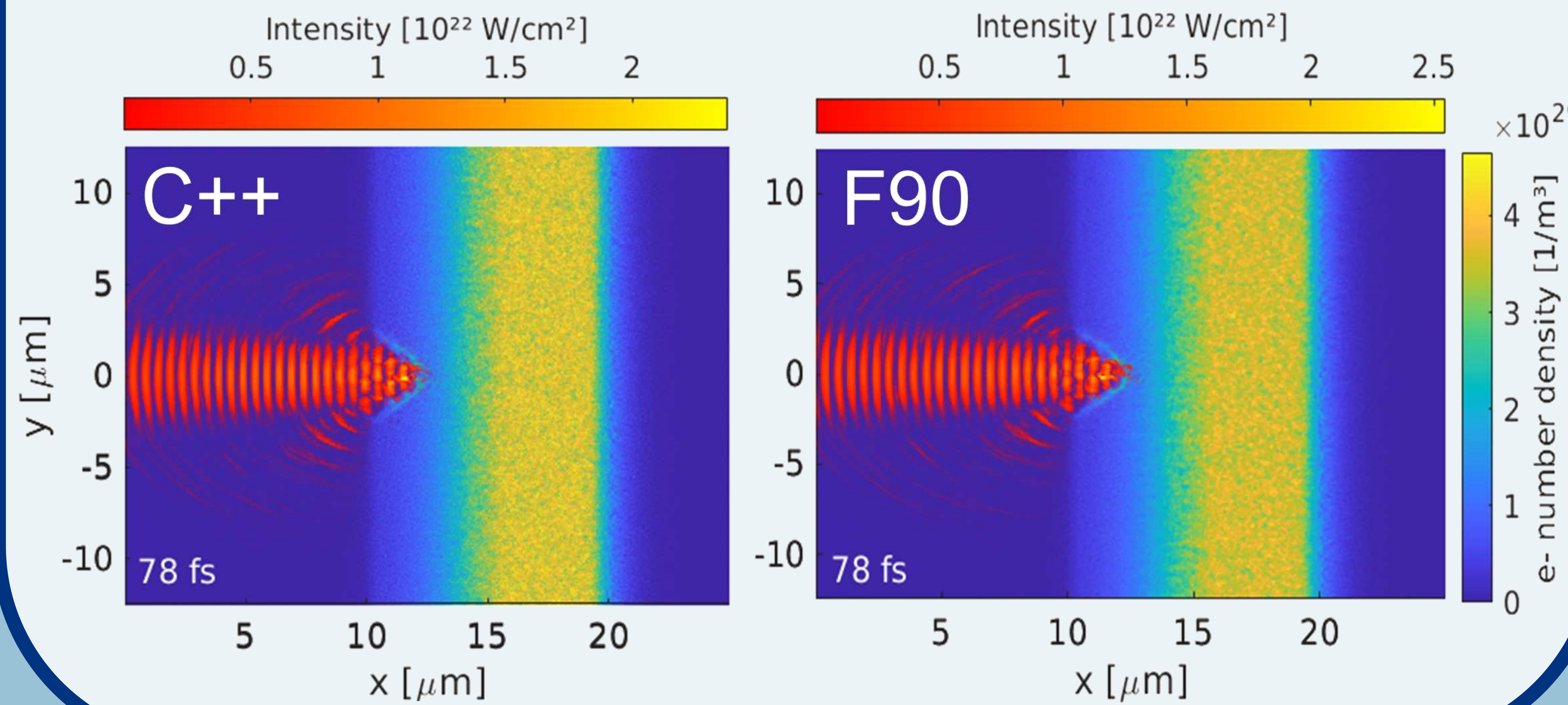
- Performance tested up to 32 cores against FORTRAN code
- Scaling comparable between two codes, with C++ 30-40% faster
- Comparison performed on 2D laser hole-boring example

Simulation parameters
(500 x 500) cells
(25 x 25) μm^2 window
x BC: open
y BC: periodic
50 ppc (40 e$^-$, 10 C^{6+})
10^{22} Wcm^{-2} Gaussian beam
5 μm ionised C target
Pre-plasma, 2 μm scale
100 fs simulated time



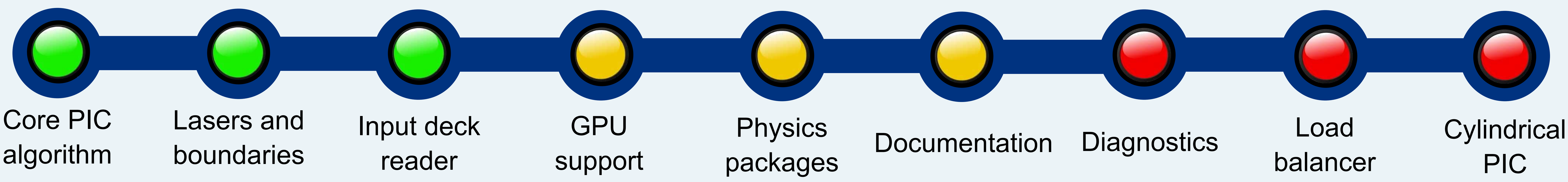
Simulation runtime as a function of MPI core count for FORTRAN and C++ code versions

- Both codes yield the same hole-boring results:



Project progress

- Visualisation of code progress. Green tasks are complete, yellow are works in progress, red are yet to start.



Acknowledgements:
This project was supported by the EPSRC grants EP/W03008X/1 and EP/W029111/1

UNIVERSITY of York · WARWICK THE UNIVERSITY OF WARWICK · UKRI Engineering and Physical Sciences Research Council

References:
[1] T. D. Arber, *Plasma Phys. Control Fusion*, 57(11). (2015)