



Deposited via The University of York.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/235564/>

Version: Published Version

Conference or Workshop Item:

Goffrey, T., Bennett, K., Arber, T.D. et al. (2025) Development of the EPOCH and EPOCcpp Codes. In: High Power Laser Christmas Meeting, 17 Dec 2025, Abingdon.

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Development of the EPOCH and EPOCHpp Codes

T. Goffrey^{*1}, K. Bennett¹, T. D. Arber¹, S. Morris¹, B. Gosling¹, S. A. Wright², A. Naden², S. Bulut² and C. Ridgers.²

1) Centre for Fusion Space and Astrophysics, University of Warwick, UK. 2) University of York, UK.

t.goffrey.1@warwick.ac.uk



Abstract

EPOCH

EPOCH Workshop: University of York, 26/27 March 2027

EPOCH++

- EPOCH¹ 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 1800 publications

- Learn the physics and numerical approaches of the EPOCH code.
- Learn about alternative approaches: hybrid methods and cylindrical PIC.
- Get help from the developers setting up a problem relevant to your research.



SIGN UP

- C++ templating allows 1D/2D/3D within same codebase.
- Defined physics package interface - easy for users to extend.
- CPU and GPU support and performance.
- Input file compatibility with EPOCH.
- Python interface to provide direct control and visualisation of simulations.
- Feature complete release: Early 2026!

EPOCH Use Case: LPI

EPOCH has been used extensively to model LPI, recently reproducing experimental data from PALS² and more recently NIF (Figure 1). For more details, see Ben Gosling's talk on Wednesday.

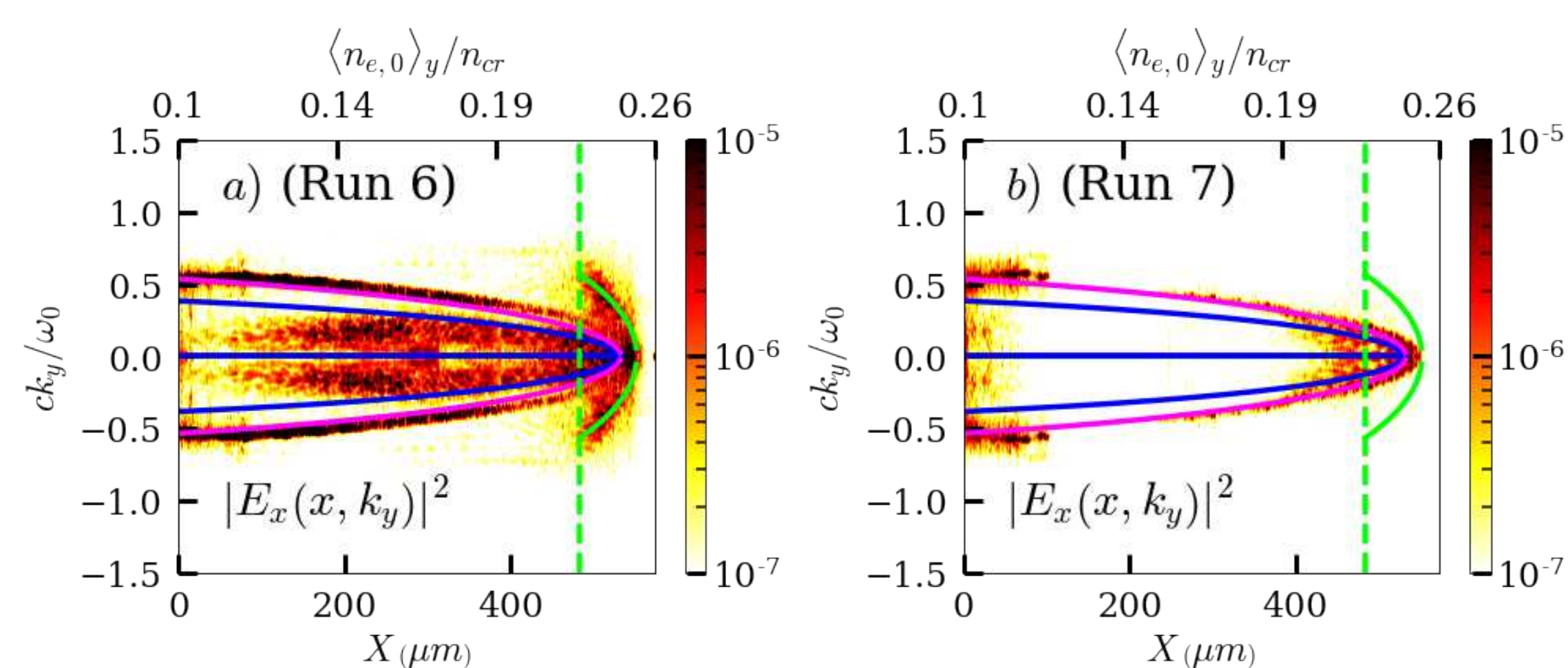


Figure 1: Transverse Fourier transform of the E_x field during early linear growth for circular (a) and out-of-plane (b) polarisation. Pink lines show the predicted perpendicular wave numbers for this mechanism, whilst blue and green curves indicate convective backscatter SRS and TPD

Upcoming Developments

New release of the core EPOCH code featuring new physics packages (recombination, linear Breit Wheeler, Compton Scattering and electron-positron annihilation) and minor bug fixes expected late 2025/early 2026.

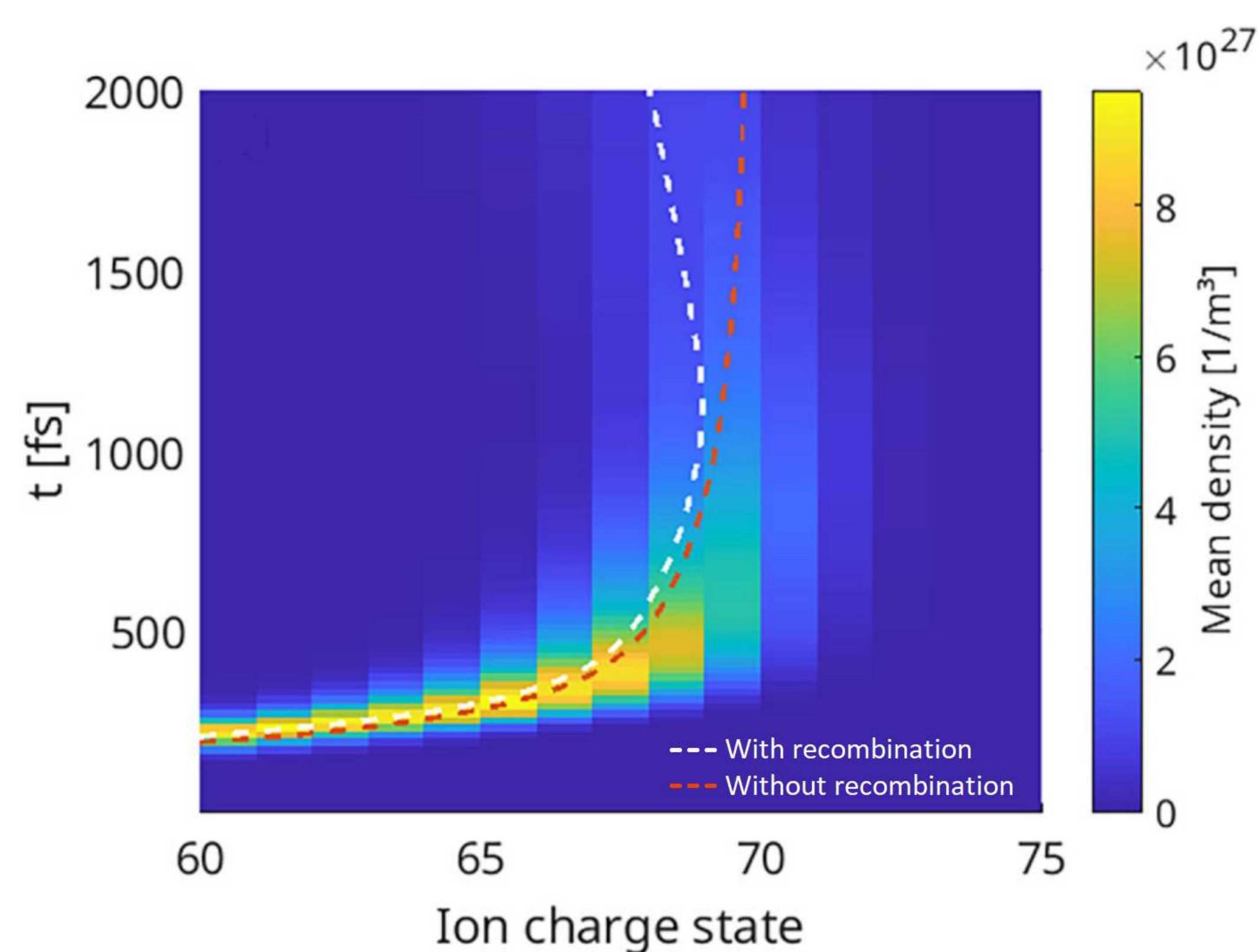


Figure 2: Distribution of charge-states in the initial target region over time, when a 300 nm target is shot by a $4 \times 10^{20} \text{ Wcm}^2$ laser pulse. Taken from Morris et al³.

References

- [1] Tony D Arber, Keith Bennett, Christopher S Brady, Alistair Lawrence-Douglas, MG Ramsay, Nathan J Sircombe, Paddy Gillies, Roger G Evans, Holger Schmitz, Anthony R Bell, et al. Contemporary particle-in-cell approach to laser-plasma modelling. *Plasma Physics and Controlled Fusion*, 57(11):113001, 2015.
- [2] B Gosling, TD Arber, G Cristoforetti, P Nicolai, and LA Gizzi. Laser-plasma instabilities driven by 1ω pulses at shock ignition conditions. *Physics of Plasmas*, 32(9), 2025.
- [3] S Morris, DO Gericke, S Fritzsche, J Machado, JP Santos, and M Afshari. Recombination effects in laser-driven acceleration of heavy ions. *Physical Review E*, 111(6):065209, 2025.

Development and Testing of the EPOCHpp Code

The performance and scaling of EPOCH and EPOCHpp are compared. A key design feature of EPOCHpp is to at least match the MPI performance of EPOCH.

Simulation parameters

Resolution	$nx = ny = 500, 50\text{ppc}$
Boundary Conditions	Open(x), Periodic(y).
Simulation Time	100fs
Laser	10^{22} Wcm^{-2} Gaussian beam.

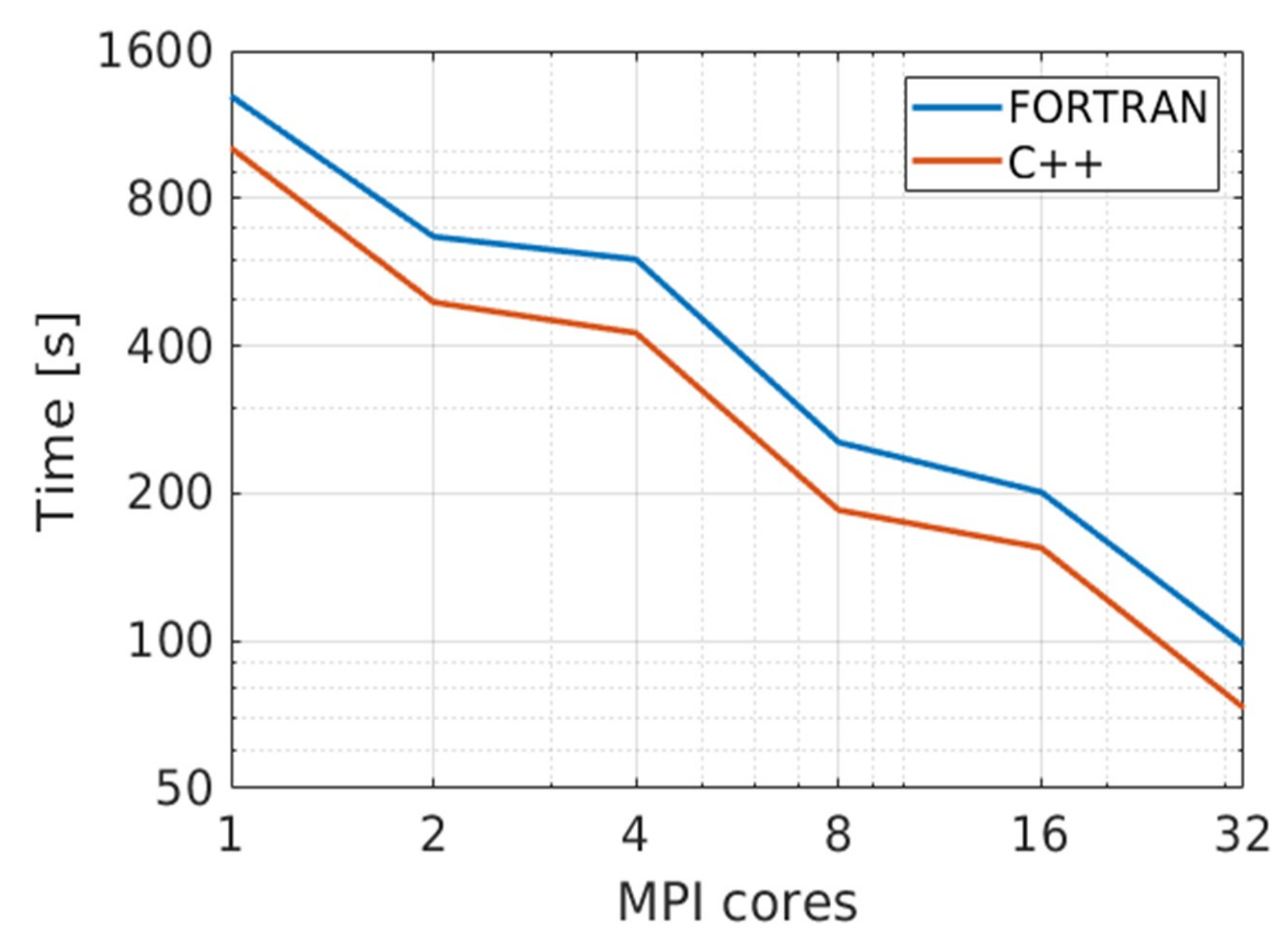


Figure 3: Both codes show similar scaling, with C++ approximately 30% faster.

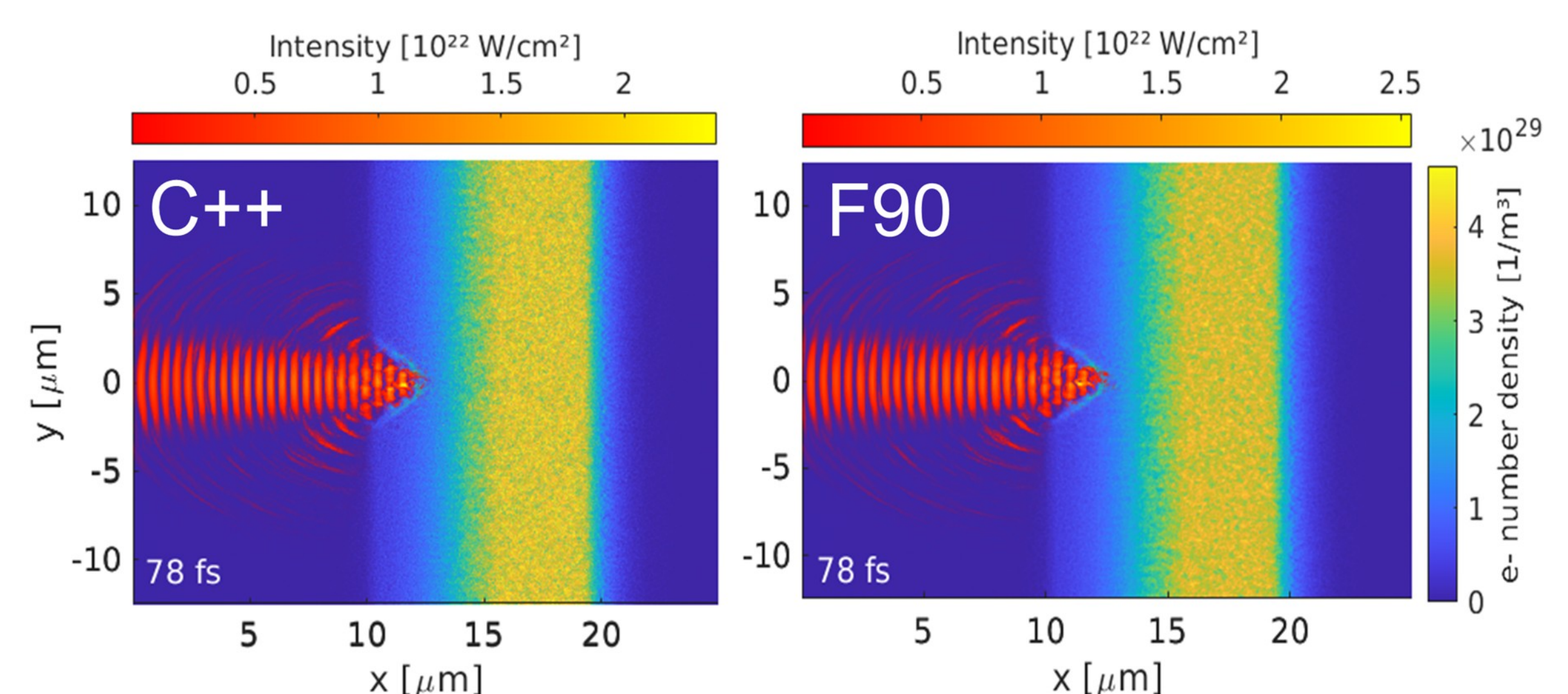


Figure 4: Similar results are observed with both versions of the code. EPOCHpp has also been used to develop new I/O methods, aiming to improve both performance and portability. Results are quoted for two runs, small ($nx = ny = 128, 64 \text{ ppc}$, 4 cores, total data 740MB) and large ($nx = ny = 1024, 128 \text{ ppc}$, 64 cores, total data 95GB).

Method/Code	Write Speed (Small)	Write Speed (Large)
EPOCH & SDF	18.53 MB/s	23.67 MB/s
EPOCHpp & POSIX	352.29 MB/s	3107.63 MB/s
EPOCHpp & ADIOS	176.15 MB/s	4291.46 MB/s

Acknowledgments

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