Project NEPTUNE

Work Package 2: Fluid Referent Models

Ben Dudson, Peter Hill, David Dickinson, Steven Wright, Edward Higgins

University of York

David Moxey

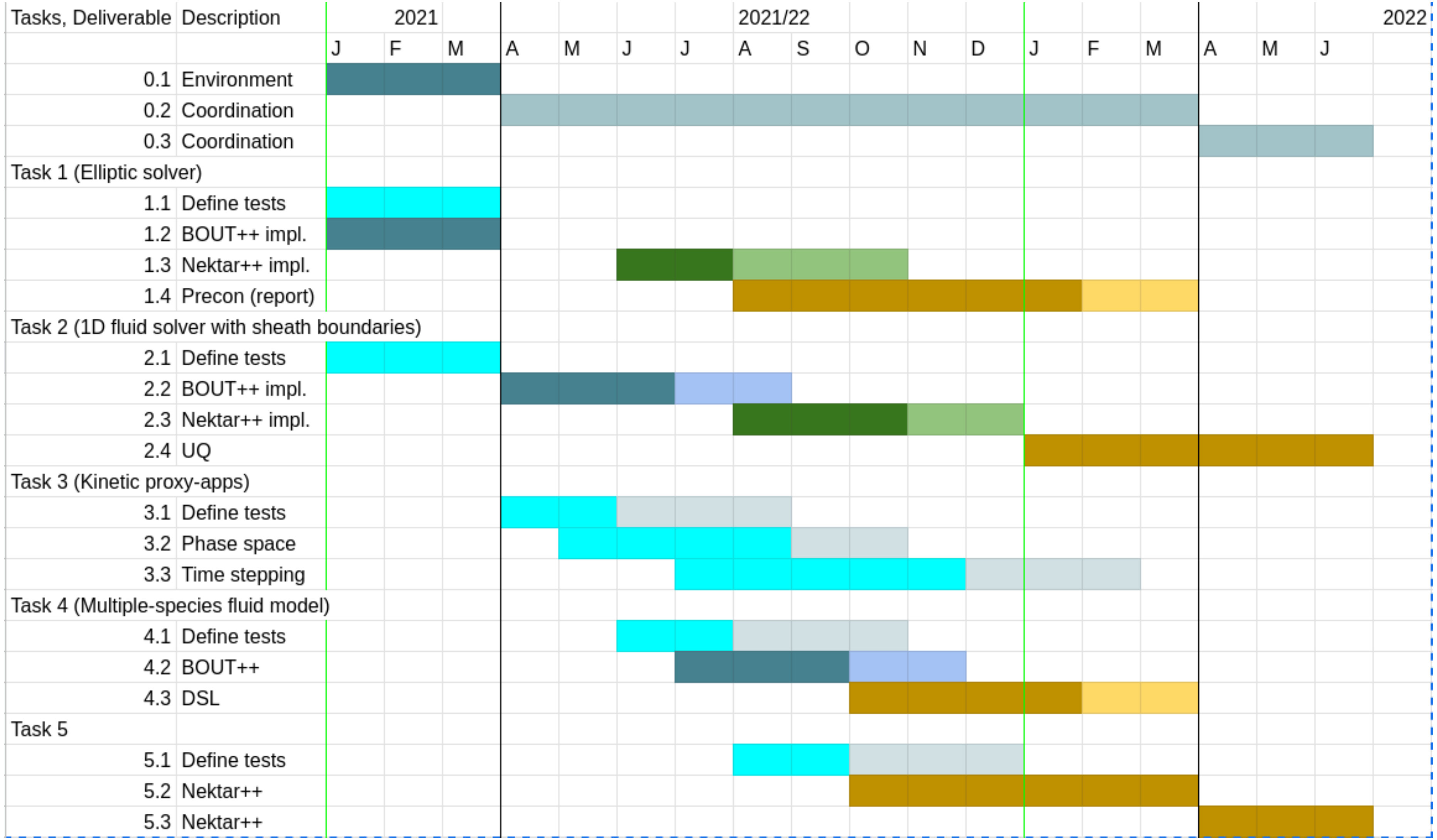
King's College London

Met Office

UK Research and Innovation

UK Atomic Energy Authority

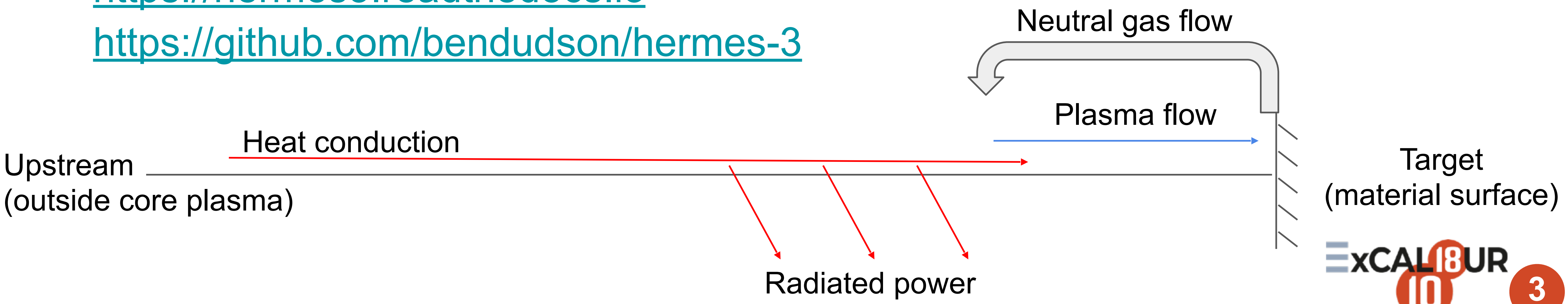
Overall Project Timeline



1D Fluid Model
SD1D

Using SD1D and Hermes-3, built on BOUT++, to develop tests and examples

- **SD1D**, a 1D model of hydrogen plasmas https://doi.org/10.1088/1361-6587/ab1321 https://github.com/boutproject/SD1D/
- **Hermes-3**, a 1D/2D/3D multi-species model under development https://hermes3.readthedocs.io https://github.com/bendudson/hermes-3

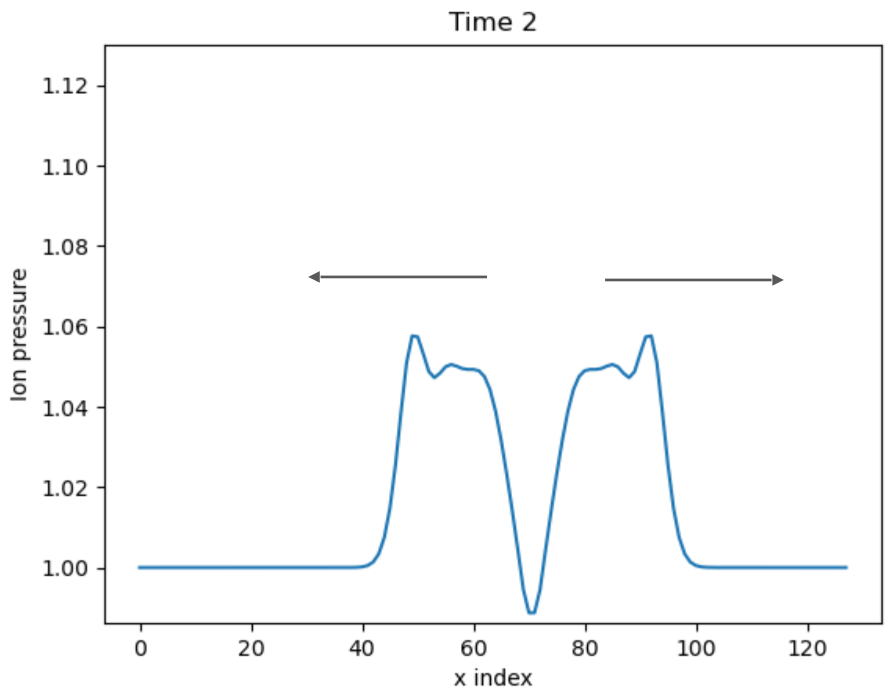


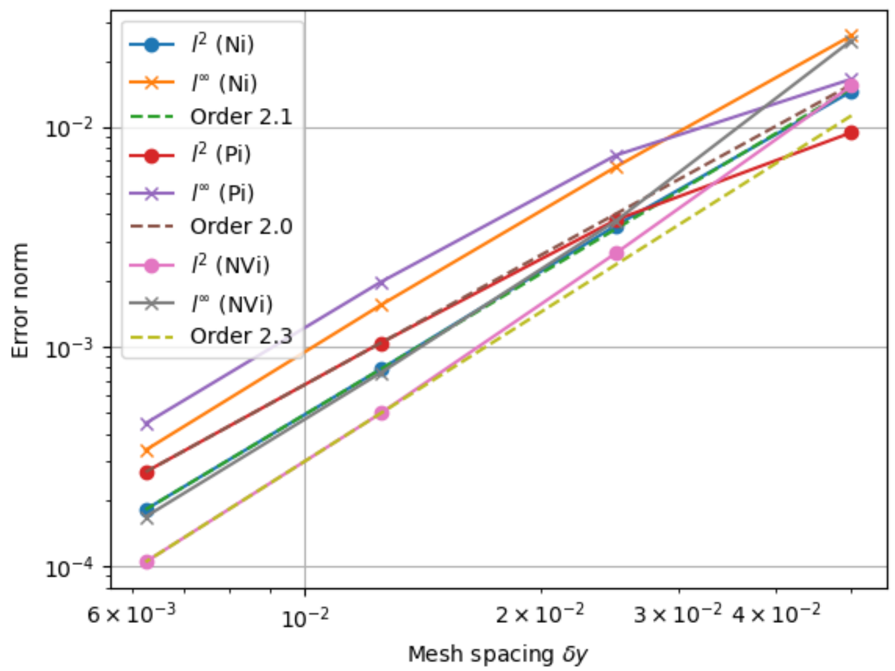
1D Fluid Model

Component Tests

Check individual parts of the model

- 1D fluid flow, periodic domain. MMS convergence test
  https://github.com/bendudson/hermes-3/tree/master/tests/integrated/1D-fluid





1D Fluid Model

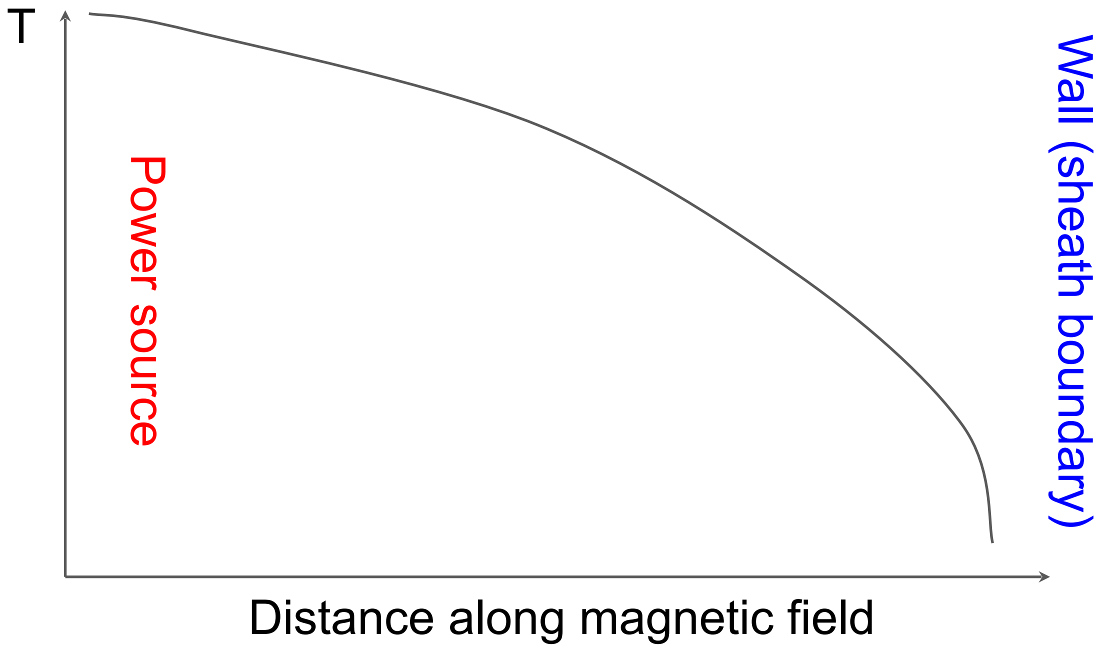
Component Tests

Check individual parts of the model

- 1D fluid flow, periodic domain. MMS convergence test
  https://github.com/bendudson/hermes-3/tree/master/tests/integrated/1D-fluid

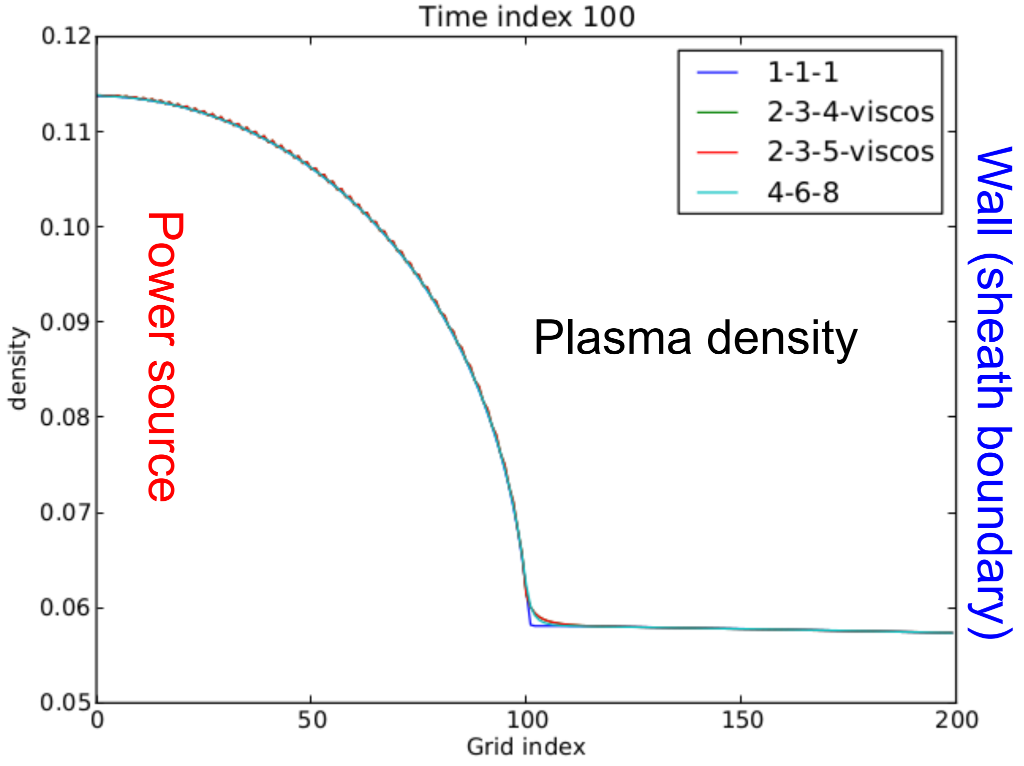
- Nonlinear heat conduction
  Analytic solution (2-point model)

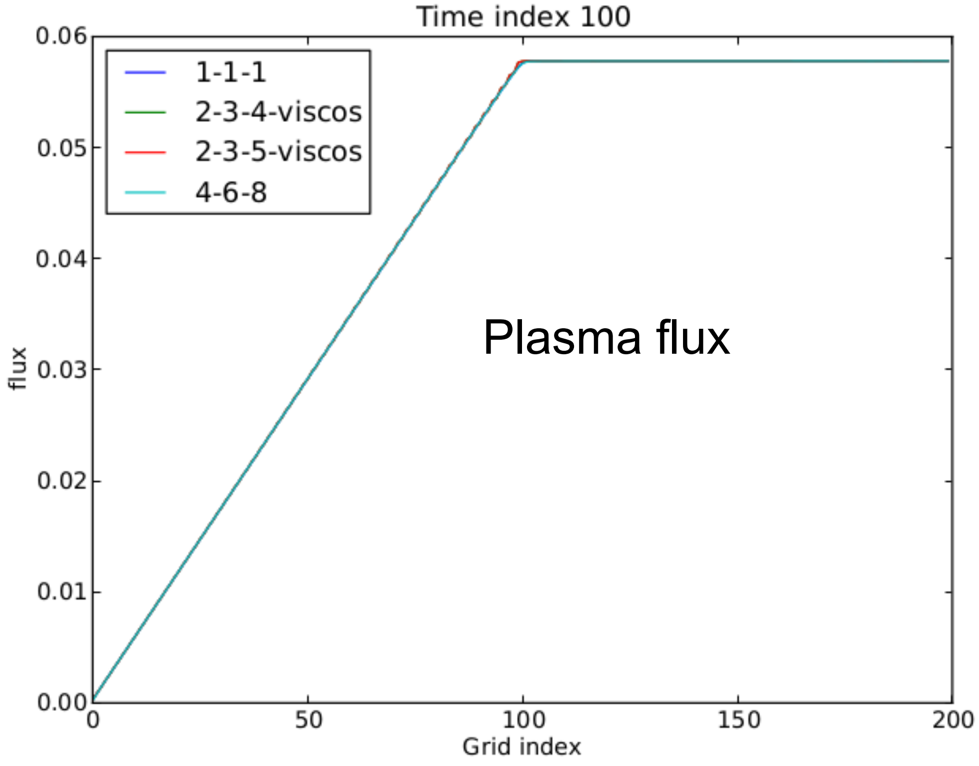
- Conservation checks
  E.g. particle balance in a closed system



1D Fluid Model
Sources

- Outflow boundary conditions on one side, no-flow on the other
- Source over part of the domain -> Sharp changes in gradients



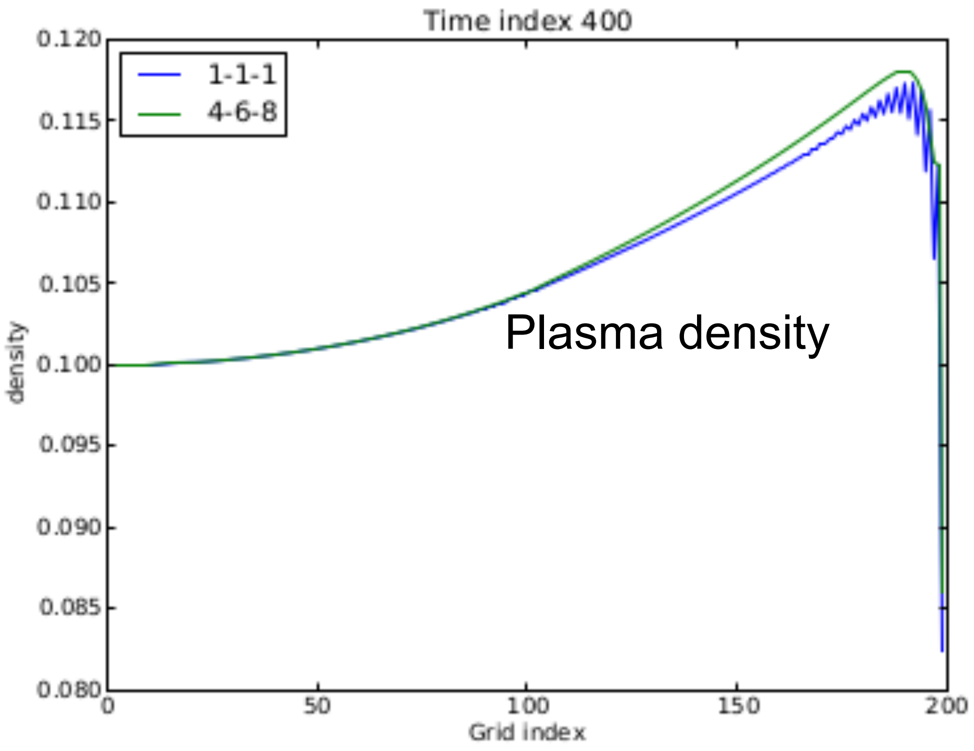


1D Fluid Model

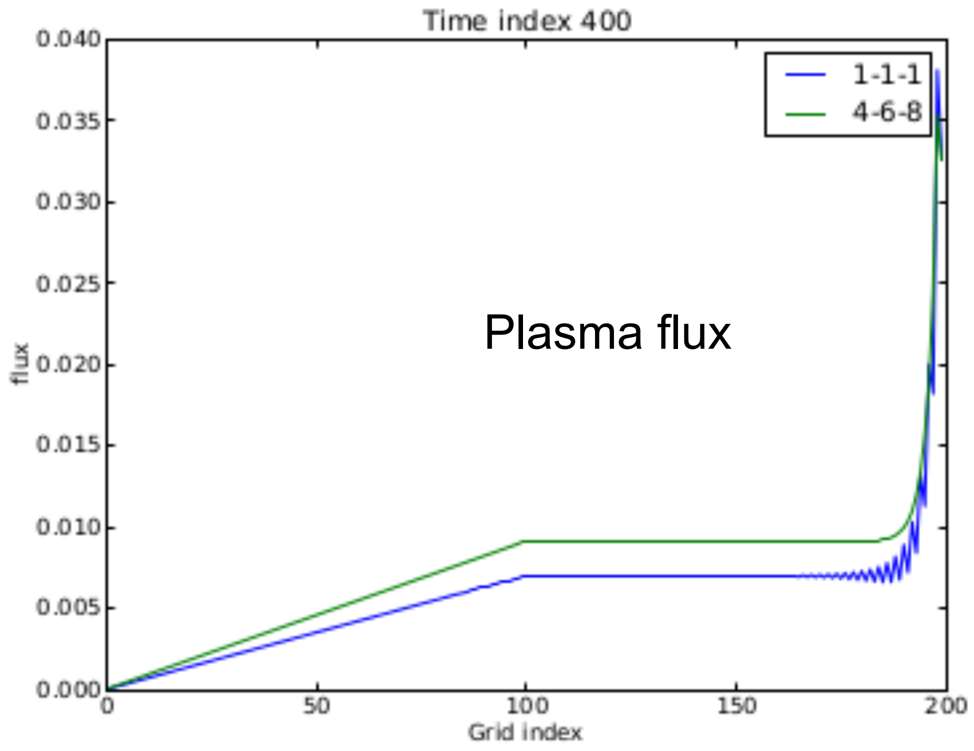
Boundary Conditions

Sheath boundary conditions (sound speed)

recycling of plasma -> neutral gas -> plasma (atomic reactions)



(a) Density n



(b) Flux $nV_{||}$

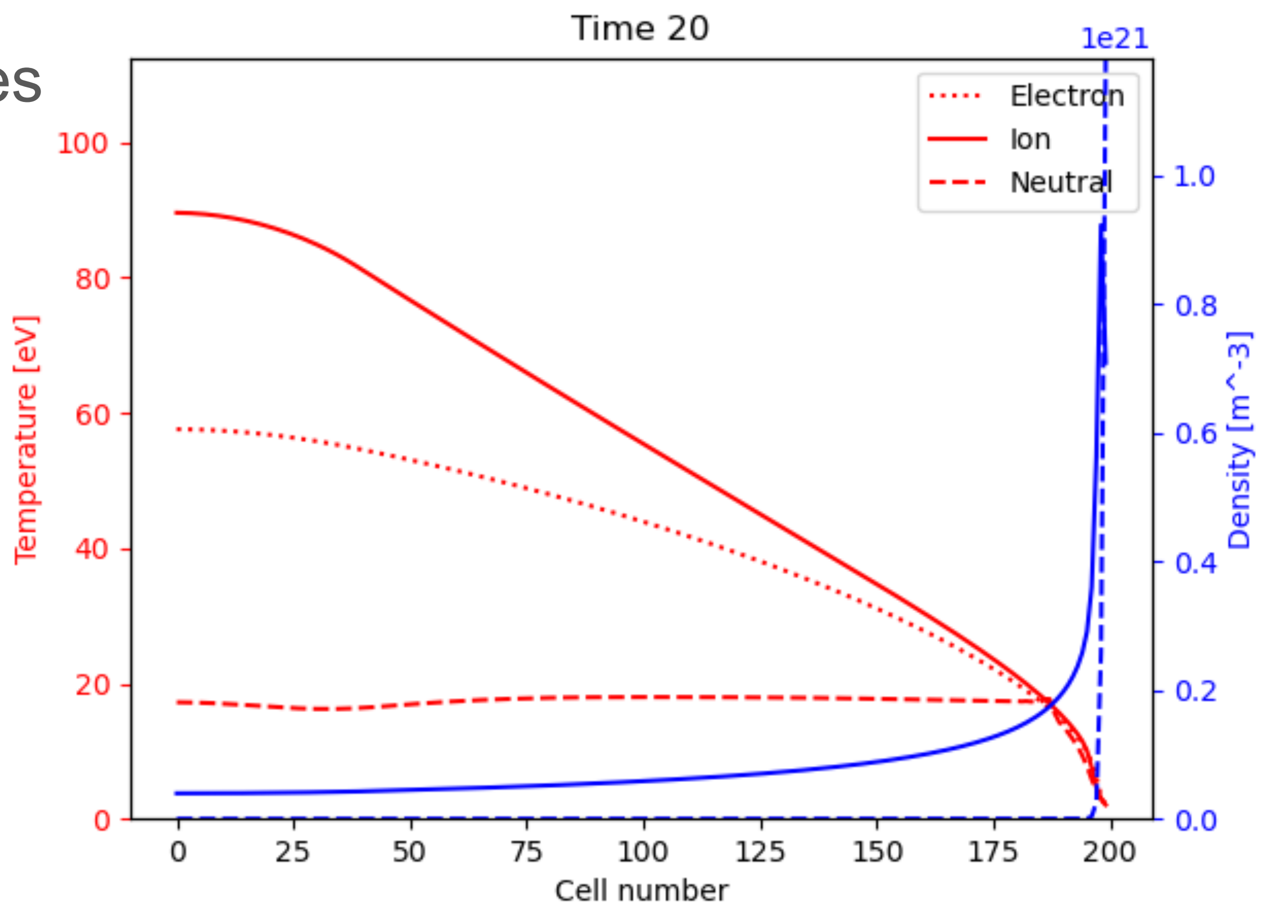
1D Fluid Model

Multi Species

- Separate ion and electron temperatures
- Coupling (collisions) between species

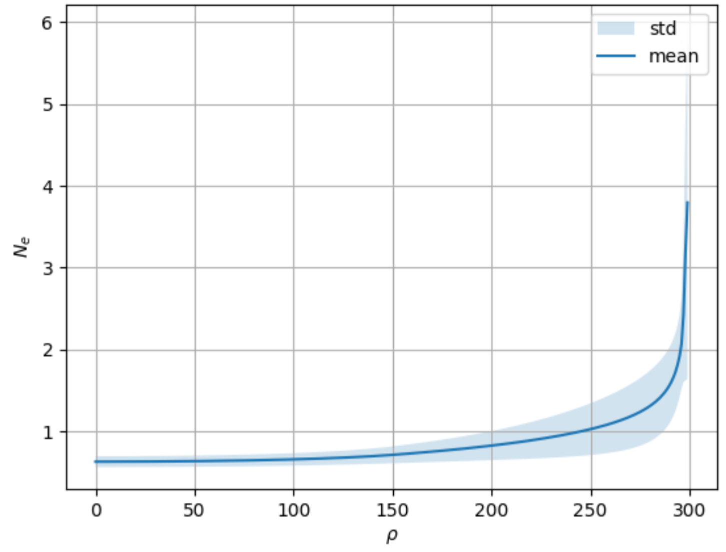
- Extension to multiple ion species
  D, T, He, Be, W, Ne, Ar, …

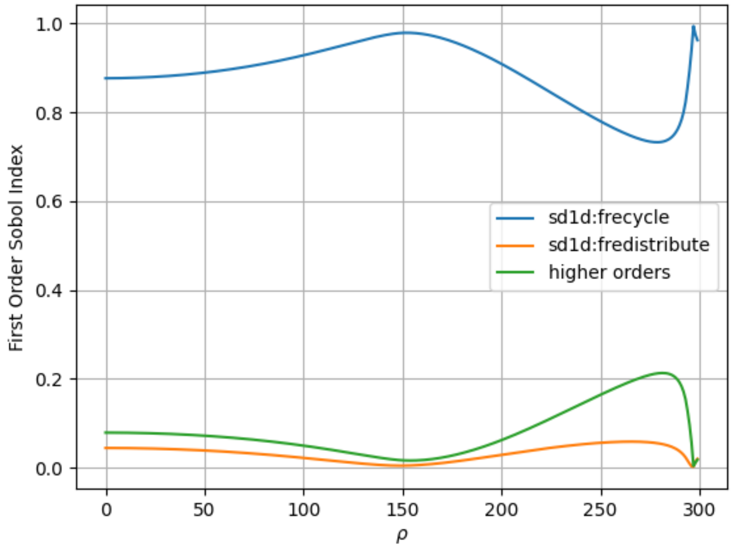
- Reactions and collisions between species



1D Fluid Model

UQ





- Developed example implementations of non-intrusive UQ workflows using EasyVVUQ with SD1D.
- Explore role of uncertainty in upstream sources and downstream boundary conditions.
- Helps identify key parameters to refine in order to reduce key uncertainties.

ExCALIBUR 10

Project NEPTUNE

Work Package 2: Fluid Referent Models

Ben Dudson, Peter Hill, David Dickinson, Steven Wright, Edward Higgins

University of York

David Moxey

King's College London

Met Office

UK Research and Innovation

UK Atomic Energy Authority

Project NEPTUNE

Work Package 4: Support and Coordination

Steven Wright, Ben Dudson, Peter Hill, David Dickinson, Edward Higgins, Chris Ridgers

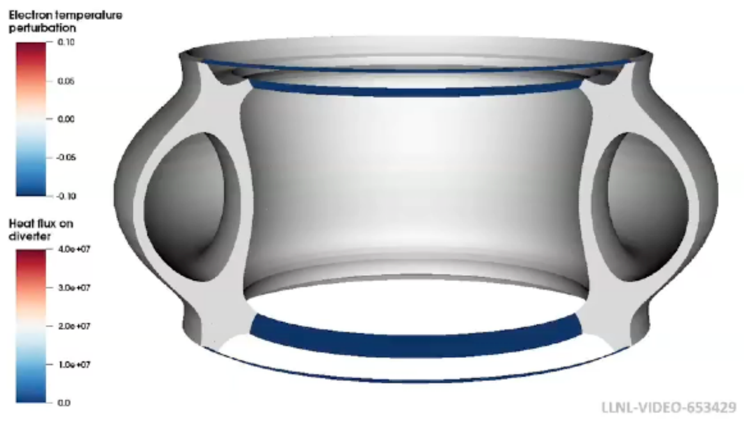
University of York

Gihan Mudalige, Tom Goffrey, Ben McMillan

University of Warwick

Project NEPTUNE

NEutrals & Plasma TUrbulence Numerics for the Exascale



NEKTAR++
SPECTRAL/HP ELEMENT FRAMEWORK

EPOCH

- **WP1 Numerical Representation**
  - **Algorithms, coupling between WP2 and WP3**
- **WP2 Plasma Multi Physics Model**
  - **Fluid models for plasma edge**
- **WP3 Neutral Gas & Impurity Model**
  - **Particle models for region outside and just inside the plasma**
- **WP4 Code Structure & Coordination**
  - **Best practices in engineering a *Performance Portable* Exascale-ready code**

ExCALIBUR

FM-WP4 Code Structure and Coordination

Funded Projects

Our work has been funded by two projects (so far):

- **Investigate DSL and Code Generation Techniques (Jan 2021 — Sept 2021)**
- **Support and Coordination (Jan 2022 — Dec 2022)**
- **Software Support Procurement (Sept 2022 — Feb 2024)**

FM-WP4 Code Structure and Coordination

Investigate DSL and code generation techniques

Our first project was based around four deliverables:

- **Survey of available (and upcoming) technologies**
- **Identification of representative applications**
- **Evaluation of approaches to performance portability for Fusion applications**
- **Identification of best practices**

FM-WP4 Code Structure and Coordination

Support and Coordination

Our second project was also based around four deliverables:

- **Periodic revision of Hardware and Software Reviews**
- **Development of an FEM-PIC mini-application**
- **Performance evaluation of FEM-PIC application**
- **Outline proposal for a particle-methods DSL**

FM-WP4 Code Structure and Coordination

Support and Coordination

Our second project was also based around four deliverables:

- Periodic revision of Hardware and Software Reviews
- Development of an FEM-PIC mini-application
- Performance evaluation of FEM-PIC application
- Outline proposal for a particle-methods DSL

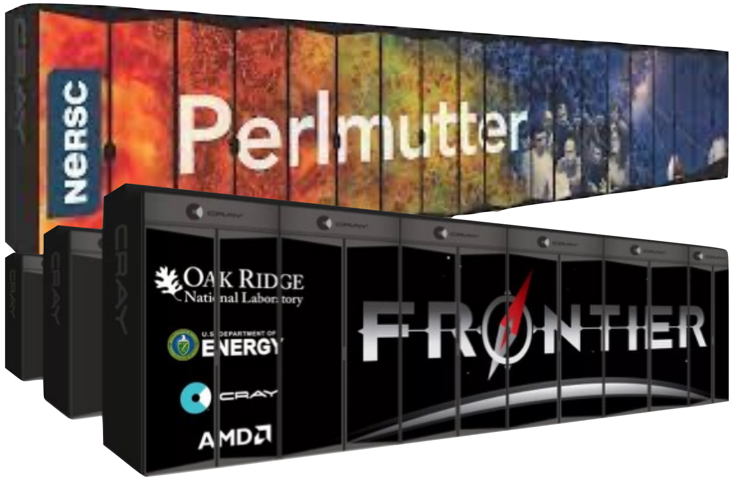
Context

Pre- and Post-Exascale Hardware

- All pre- and post-Exascale systems will be heterogenous (… except Fugaku)
- Most of the FLOP/s will be provided by GPU accelerators
    - NVIDIA Hopper
    - AMD Instinct
    - Intel Xe
- Most systems will use x86_64 hosts from Intel and AMD
  (+ perhaps some NVIDIA Grace)

Context

Pre- and Post-Exascale Hardware



DoE have 4 Cray Shasta systems installed or in development

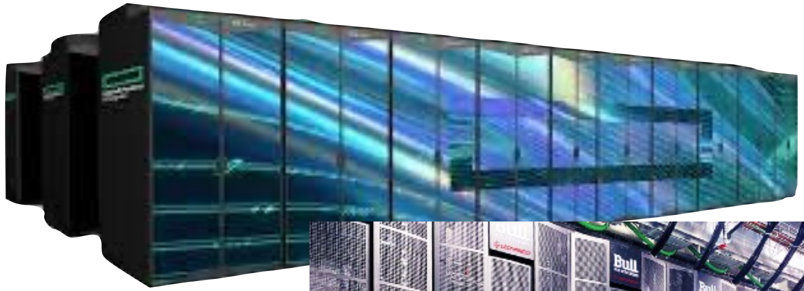
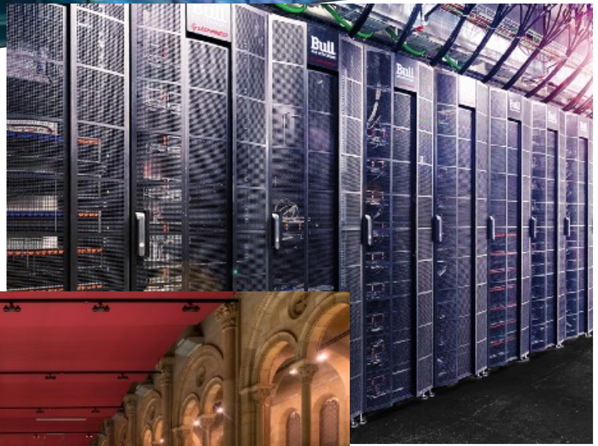
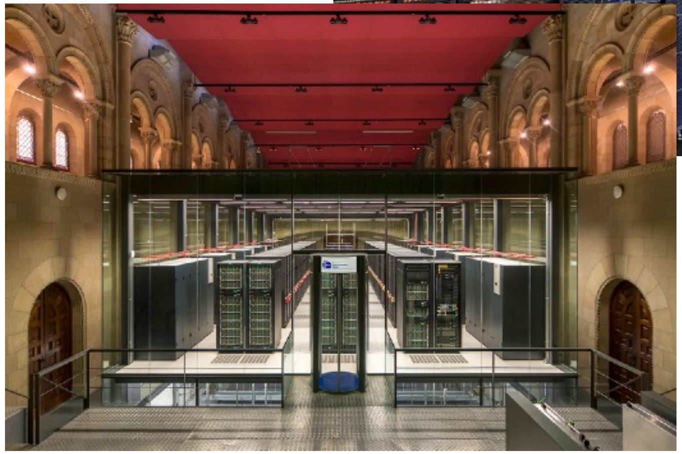
- Perlmutter
  - 70.87 PFLOP/s (achieved… so far!)
  - AMD EPYC CPUs (Milan) with NVIDIA A100 GPUs
- Frontier
  - 1.1 EFLOP/s 🎉 🎉 🎉 (achieved)
  - AMD EPYC CPUs (Trento) with AMD Instinct (MI250X) GPUs
- Aurora (To Appear)
  - >1 EFLOP
  - Intel Xeon CPUs (Sapphire Rapids) with Intel Xe (PV) GPUs
- El Capitan (To Appear)
  - 2+ EFLOP/s
  - AMD EPYC CPUs (Genoa) with AMD Instinct GPUs

Context

Pre- and Post-Exascale Hardware

Europe has 3 pre-Exascale systems in development

- LUMI
  - 151 PFLOP/s (Achieved)
  - AMD EPYC (Trento) CPUs with AMD Instinct (MI250X) GPUs
- LEONARDO
  - 300+ PFLOP/s
  - Intel Xeon CPUs (Sapphire Rapids) with NVIDIA A100 GPUs
- MareNostrum 5
  - 300+ PFLOP/s
  - 4 distinct PFLOP/s systems
    - Intel Xeon CPUs (Sapphire Rapids) with NVIDIA H100 GPUs
    - Intel Xeon CPUs (Emerald Rapids) with Rialto Bridge GPUs
    - Intel Xeon CPUs (Sapphire Rapids)
    - NVIDIA Grace Superchips

Context

Pre- and Post-Exascale Hardware

The UK

- ARCHER-2
  - ~28 PFLOP/s
  - AMD EPYC CPUs
- Isambard
  - ARM CPUs (ThunderX2 and A64FX)
  - AMD EPYC CPUs
  - Intel Xeon CPUs
  - IBM Power9 CPUs
  - NVIDIA P100 and NVIDIA V100 GPUs
- Viking, Avon, Bede, etc

ExCALIBUR

Context

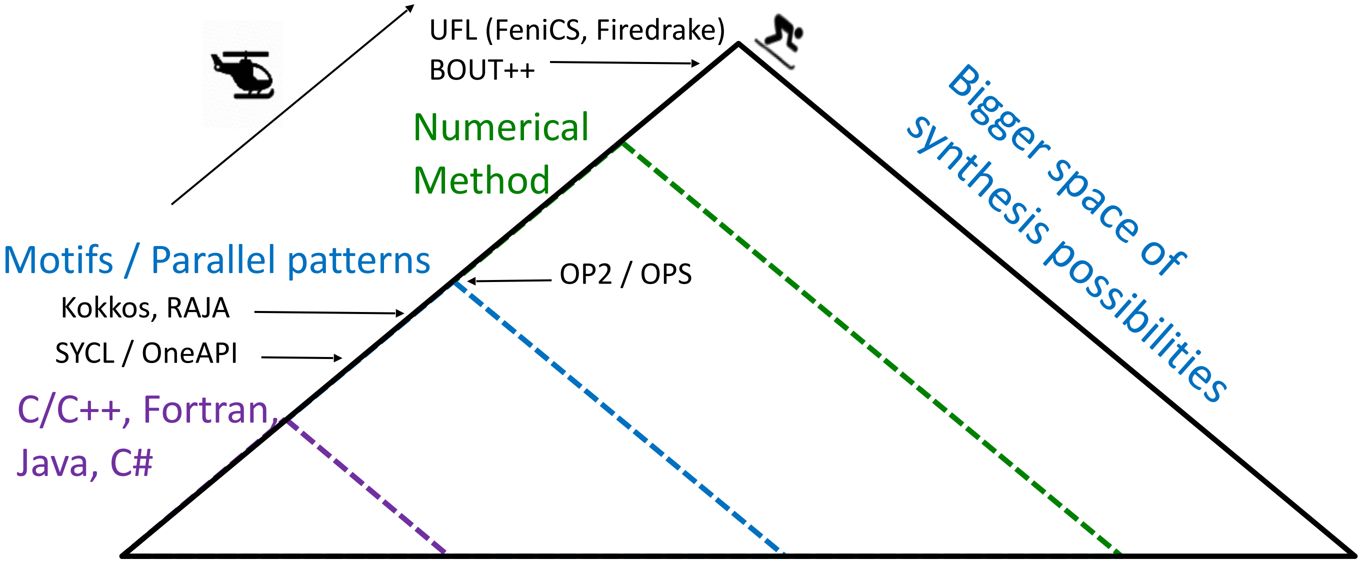
Software Approaches to Exascale

- Proliferation of Parallel programming models and software
- Open standards
  - Directives – OpenMP, OpenACC
  - OpenCL and SYCL
- Proprietary models
  - Almost as if each vendor introduces their own proprietary model
  - NVIDIA - OpenACC, CUDA
  - AMD - HIP/ROCm
  - Intel - OneAPI (SYCL + Intel specific software)
- Template libraries
  - Kokkos and RAJA (championed by the DoE labs)

Context

Software Approaches to Exascale

- **Domain Specific Languages**
  - **Code-gen from mathematical equations - Bout++, FeniCS, Firedrake, Devito**
  - **Parallel patters/motifs based Code-gen - OP2/OPS**



Adapted from: *Synthesis versus Analysis: What Do We Actually Gain from Domain-Specificity?* Keynote talk at the LCPC 2015. Paul H. J. Kelly (Imperial College London)

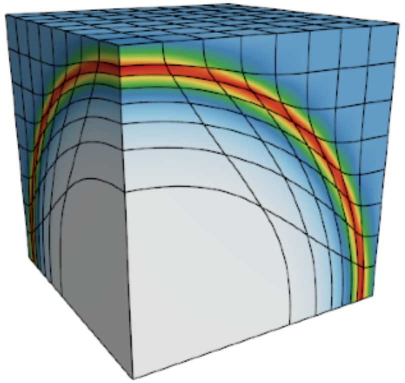
Applications for Evaluation

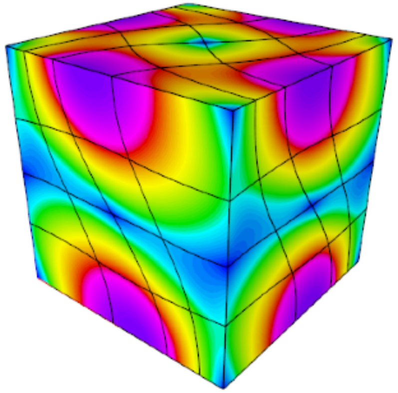
Identification of representative proxy applications

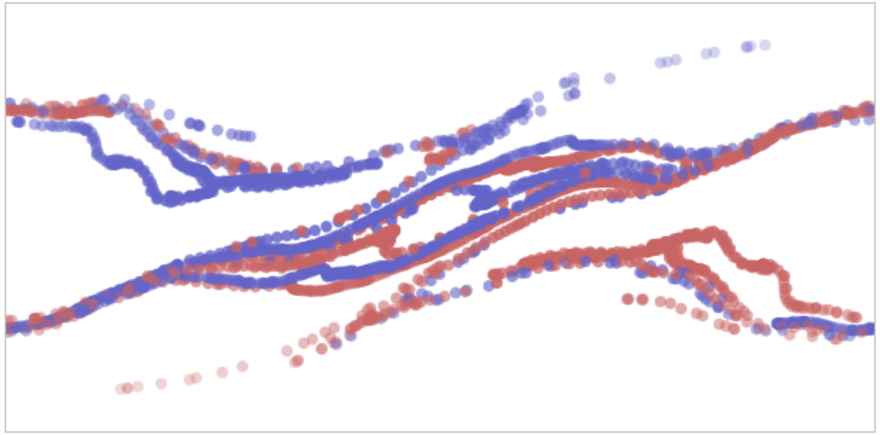
- NEPTUNE requires both Fluid and Particle models
- BOUT++/Nektar/EPOCH are likely too large to evaluate for portability
- Our evaluation is based on mini-applications that implement similar schemes
    - Finite Differencing
    - Finite Element Method
    - High-order FEM
    - Particle-in-Cell

Applications for Evaluation

Identification of representative proxy applications







- Fluid Codes
  - HEAT — Finite difference mini-app
  - TEALEAF – Finite difference mini-app
  - MINIFE – Finite element mini-app
  - LAGHOS – High-order FEM mini-app
- Particle Codes
  - CABANAPIC – structured PIC demonstrator
  - VPIC – Vectorised, structured PIC code
  - EMPIRE-PIC – Unstructured FEM-PIC code
- … plus more apps to be added!

ExCALIBUR

Evaluation Methodology

A Metric for Performance Portability

- **Evaluation based on Pennycook et al.'s metric [1]**

$$\text{Ᵽ}(a,p,H) = \begin{cases} \dfrac{|H|}{\displaystyle\sum_{i \in H} \frac{1}{e_i(a,p)}} & \text{if } i \text{ is supported } \forall i \in H \\ 0 & \text{otherwise} \end{cases}$$

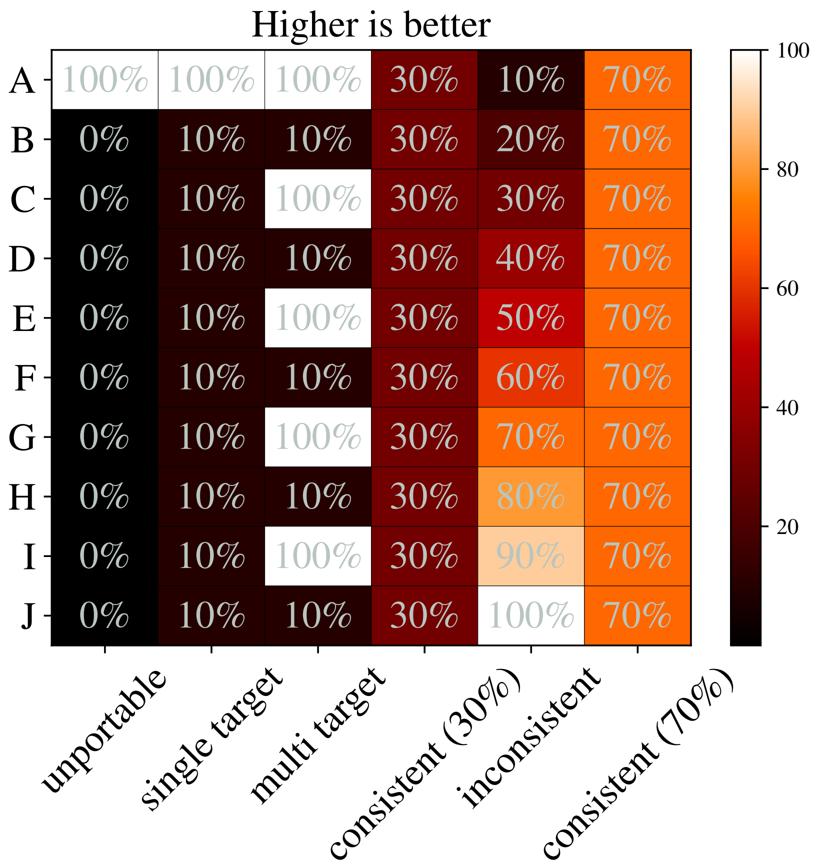
- **Find harmonic mean of applications performance efficiency**
  - **Application efficiency – achieved performance / best recorded performance**
  - **Architectural efficiency – achieved performance / theoretical max**

[1] S.J. Pennycook, J.D. Sewall, and V.W. Lee. Implications of a metric for performance portability. Future Generation Computer Systems, 92:947 –958, 2019.

Evaluation Methodology

Interpreting and Visualising Performance Portability Metrics

- **A single portability metric may hide useful insights**
- **Analysis augmented with visualisations from Sewall et al. [2]**
- **Example (from [2])**
  - **Synthetic data set with 6 variants**
  - **Boxplots show median portability, spread and outliers**
  - **Cascade plots show how performance portability changes as platforms are added**

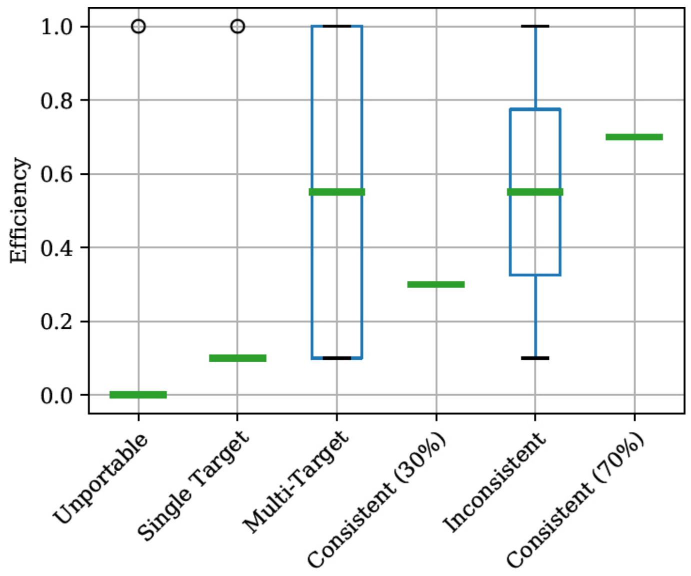


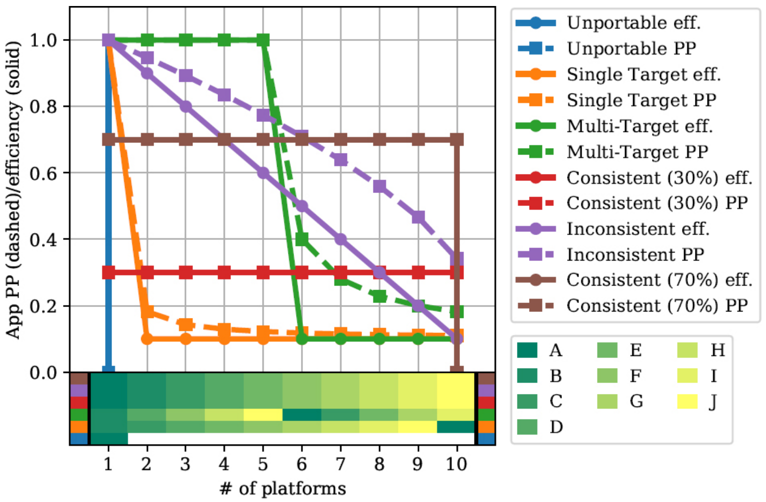
[2] J.D. Sewall, S.J. Pennycook, D. Jacobsen, T. Deakin, and S. McIntosh-Smith. Interpreting and visualizing performance portability metrics. In 2020 P3HPC Workshop, pages 14–24, 2020.

Evaluation Methodology

Interpreting and Visualising Performance Portability Metrics

- **A single portability metric may hide useful insights**
- **Analysis augmented with visualisations from Sewall et al. [2]**
- **Example (from [2])**
  - **Synthetic data set with 6 variants**
  - **Boxplots show median portability, spread and outliers**
  - **Cascade plots show how performance portability changes as platforms are added**



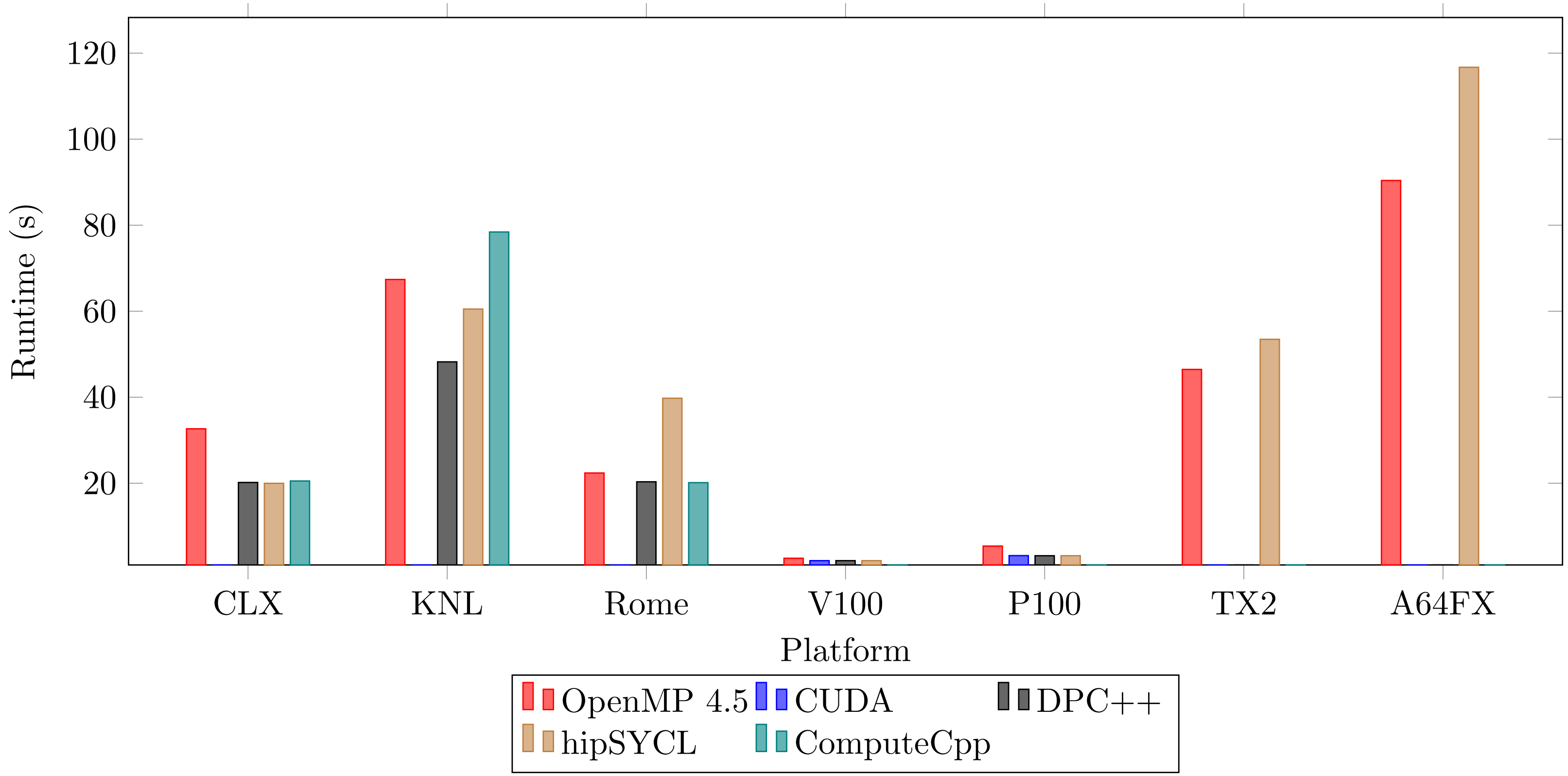


[2] J.D. Sewall, S.J. Pennycook, D. Jacobsen, T. Deakin, and S. McIntosh-Smith. Interpreting and visualizing performance portability metrics. In 2020 P3HPC Workshop, pages 14–24, 2020.

Evaluation of Approaches

Heat

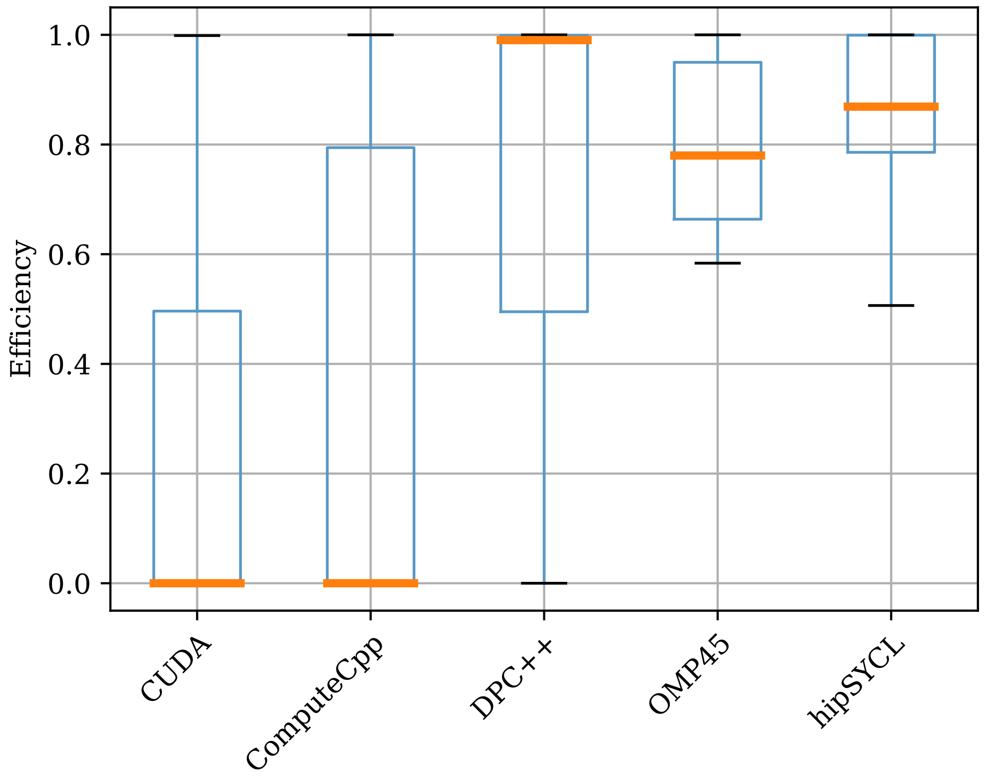
- Simple finite difference heat equation on structured grid
- Results collected from Isambard

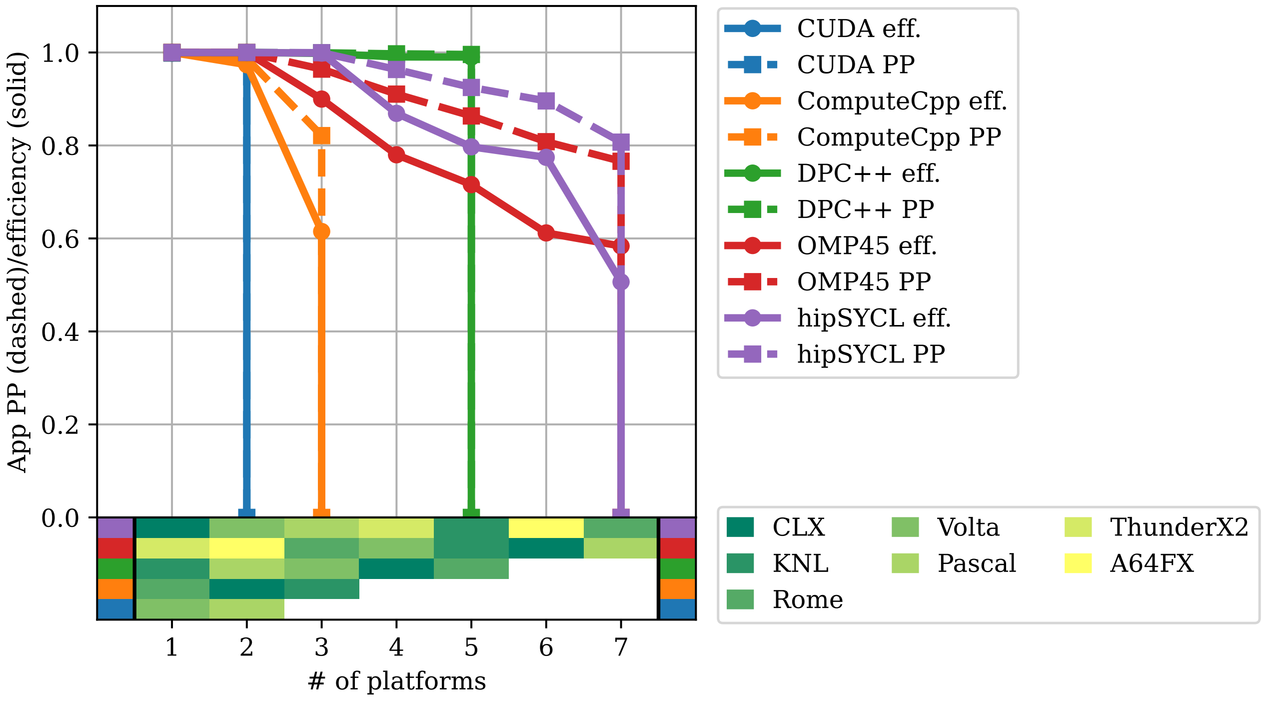


Evaluation of Approaches

Heat

- **Simple finite difference heat equation on structured grid**
- **Results collected from Isambard**



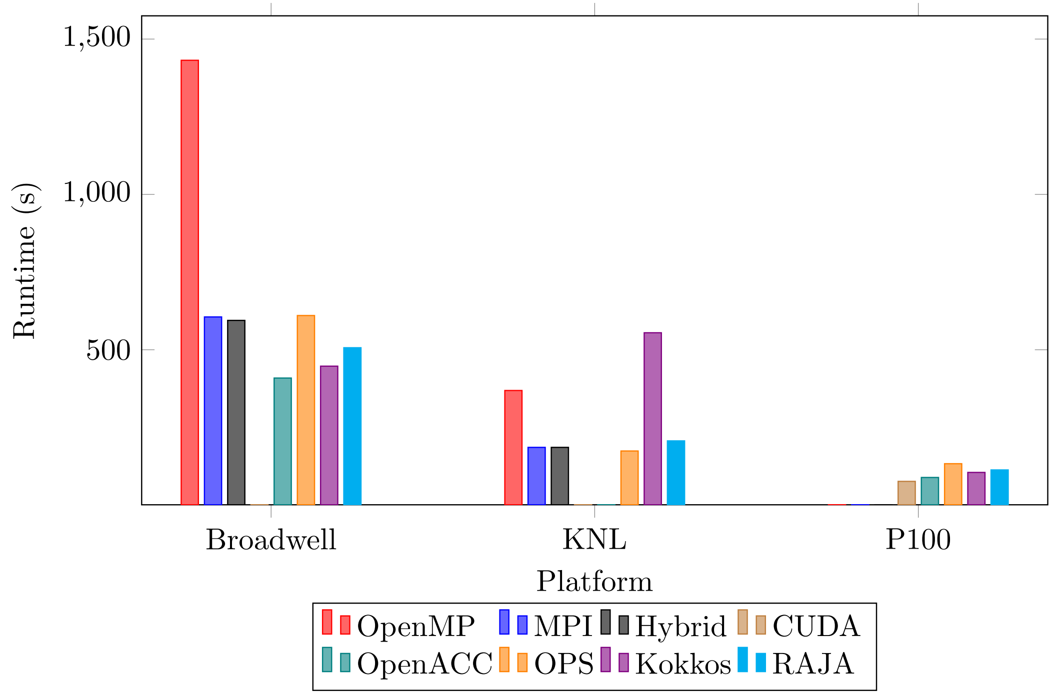


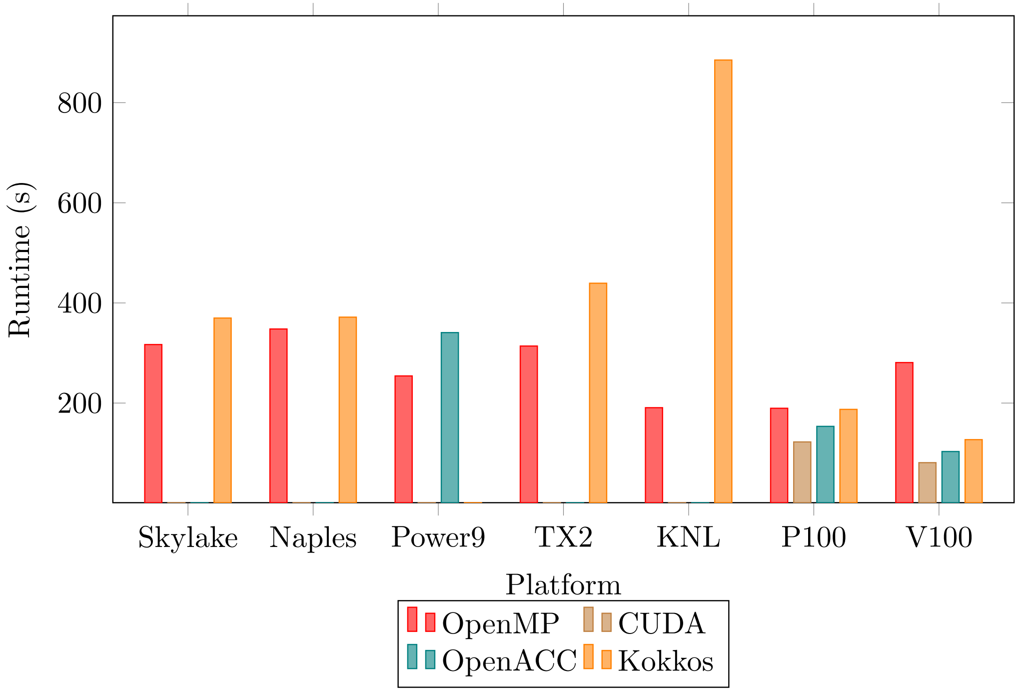
ExCALIBUR

Evaluation of Approaches

TeaLeaf

- **Finite differencing on structured grid**
- **Performance data taken from two studies [3,4]**





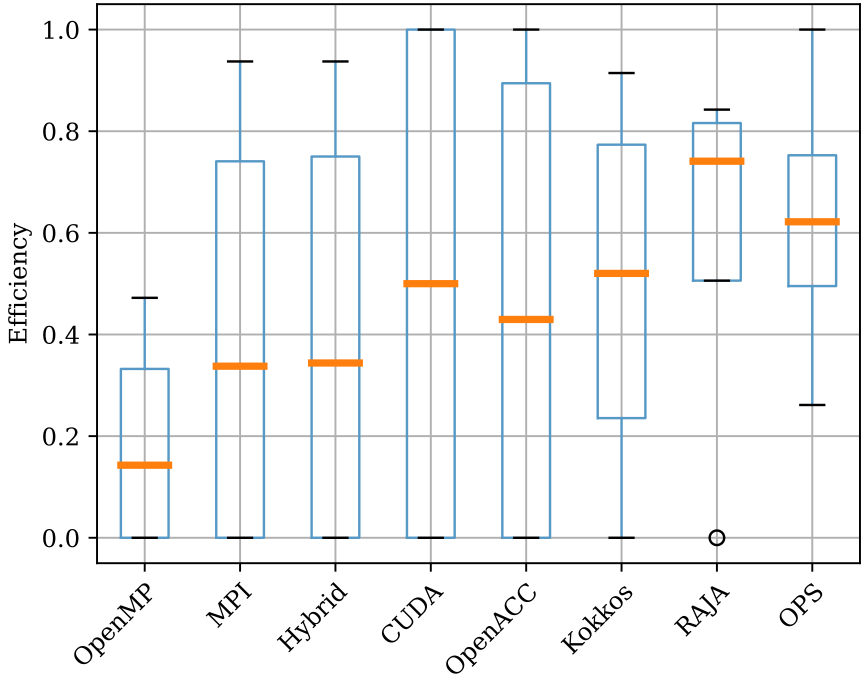
[3] R. O. Kirk, G. R. Mudalige, I. Z. Reguly, S. A. Wright, M. J. Martineau, and S. A. Jarvis. Achieving Performance Portability for a Heat Conduction Solver Mini-Application on Modern Multi-core Systems. In IEEE CLUSTER, pages 834–841, Sep. 2017

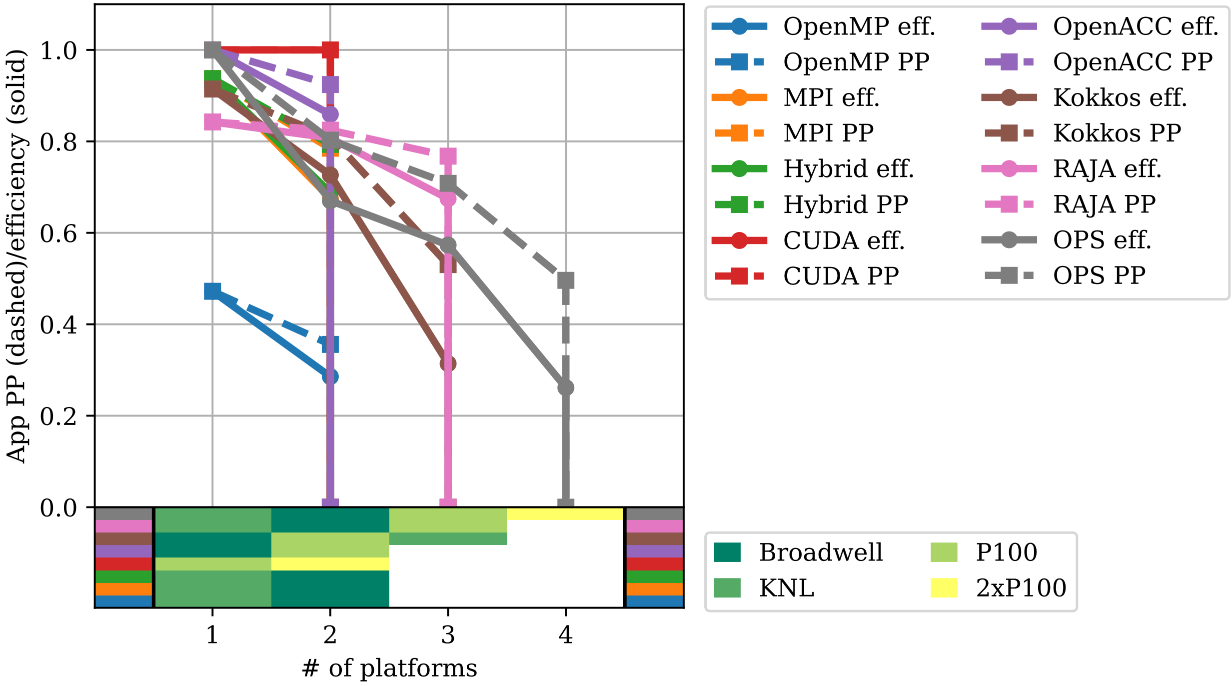
[4] T. Deakin, S. McIntosh-Smith, J. Price, A. Poenaru, P. Atkinson, C. Popa, and J. Salmon. Performance portability across diverse computer architectures. In P3HPC Workshop, pages 1–13, 2019.

Evaluation of Approaches

TeaLeaf

- **Finite differencing on structured grid**
- **Performance data taken from two studies [3,4]**



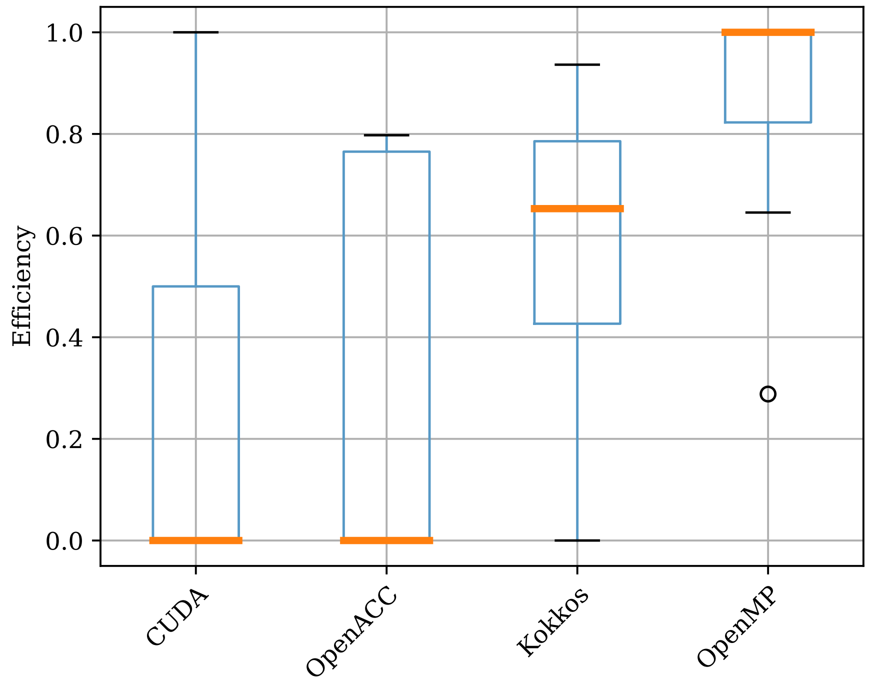


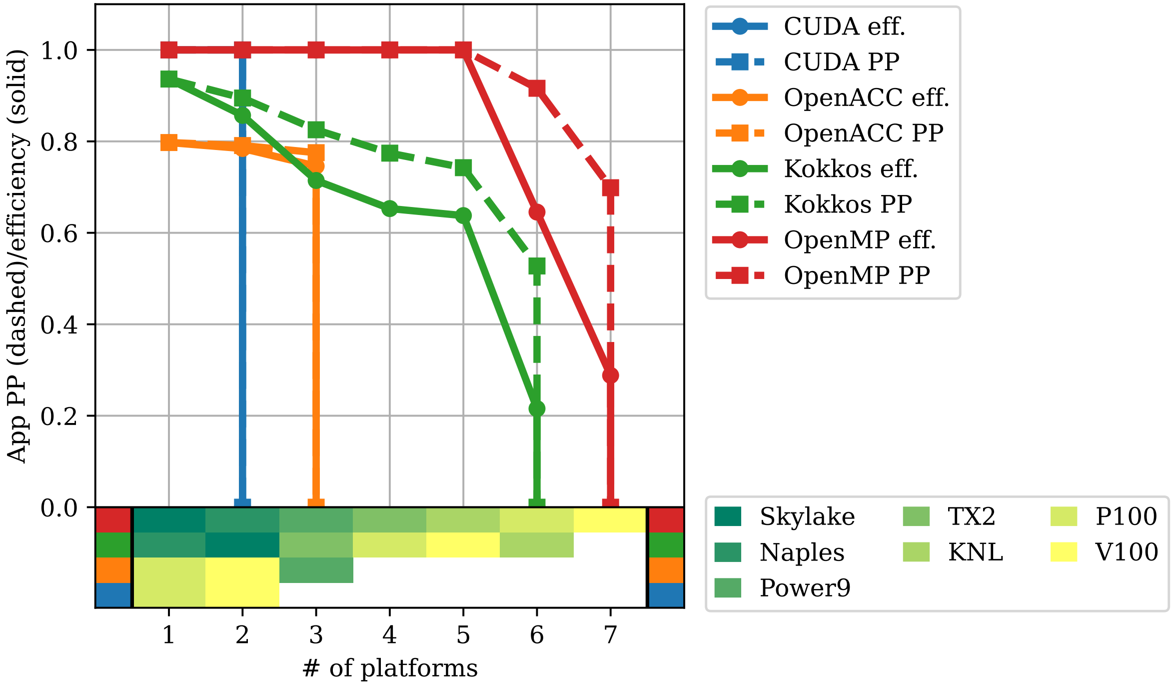
[3] R. O. Kirk, G. R. Mudalige, I. Z. Reguly, S. A. Wright, M. J. Martineau, and S. A. Jarvis. Achieving Performance Portability for a Heat Conduction Solver Mini-Application on Modern Multi-core Systems. In IEEE CLUSTER, pages 834–841, Sep. 2017
[4] T. Deakin, S. McIntosh-Smith, J. Price, A. Poenaru, P. Atkinson, C. Popa, and J. Salmon. Performance portability across diverse computer architectures. In P3HPC Workshop, pages 1–13, 2019.

Evaluation of Approaches

TeaLeaf

- **Finite differencing on structured grid**
- **Performance data taken from two studies [3,4]**





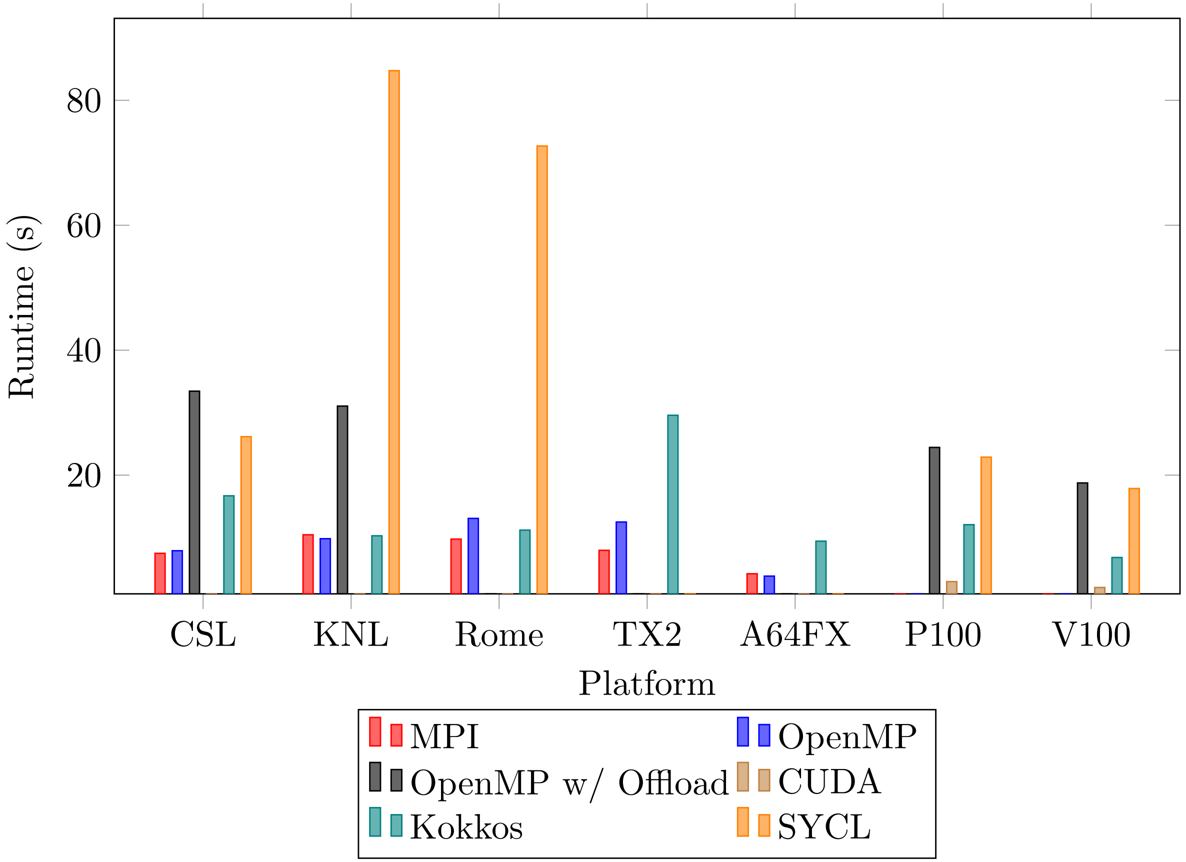
[3] R. O. Kirk, G. R. Mudalige, I. Z. Reguly, S. A. Wright, M. J. Martineau, and S. A. Jarvis. Achieving Performance Portability for a Heat Conduction Solver Mini-Application on Modern Multi-core Systems. In IEEE CLUSTER, pages 834–841, Sep. 2017

[4] T. Deakin, S. McIntosh-Smith, J. Price, A. Poenaru, P. Atkinson, C. Popa, and J. Salmon. Performance portability across diverse computer architectures. In P3HPC Workshop, pages 1–13, 2019.

Evaluation of Approaches

miniFE

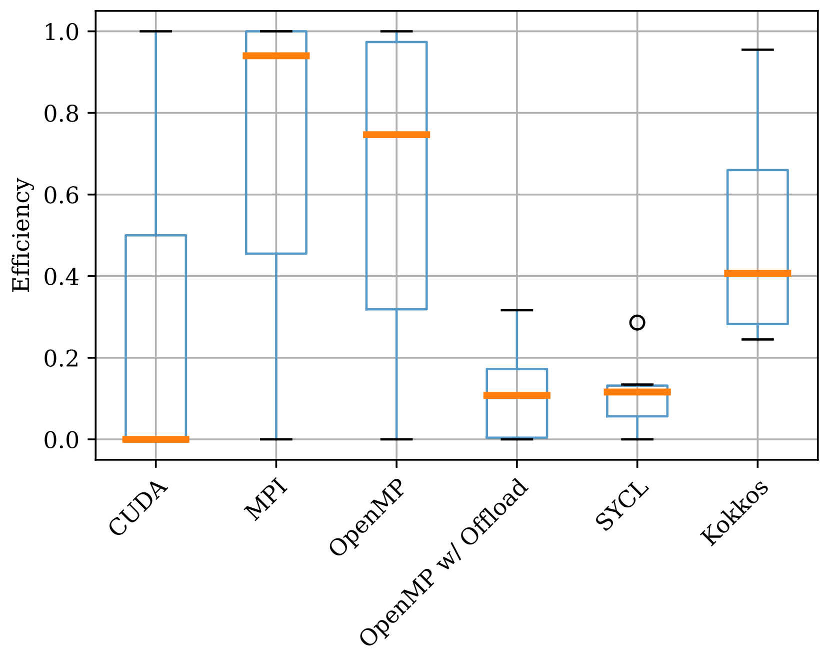
- **Finite element method code, part of Mantevo (now ECP Proxy Apps)**
- **Results collected from Isambard**

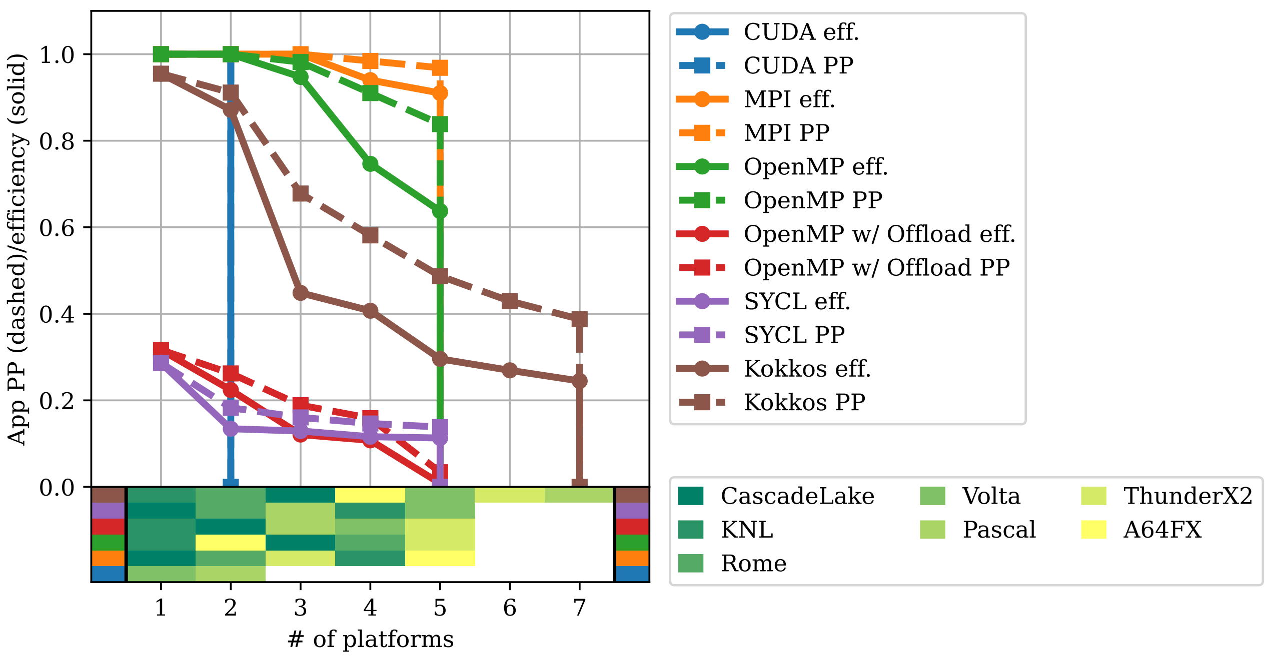


Evaluation of Approaches

miniFE

- **Finite element method code, part of Mantevo (now ECP Proxy Apps)**
- **Results collected from Isambard**

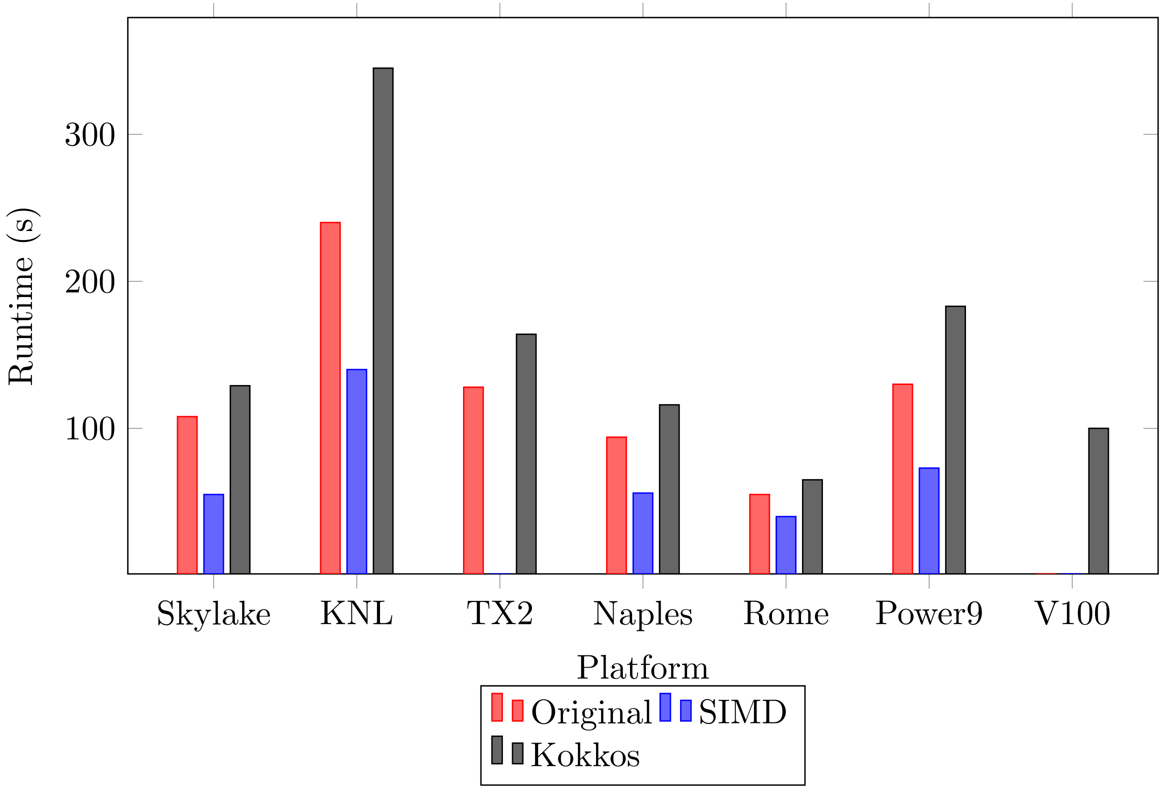




Evaluation of Approaches

VPIC/VPIC 2.0

- **Hand implemented vectorised kernels (native)**
- **VPIC 2.0 in Kokkos [5]**

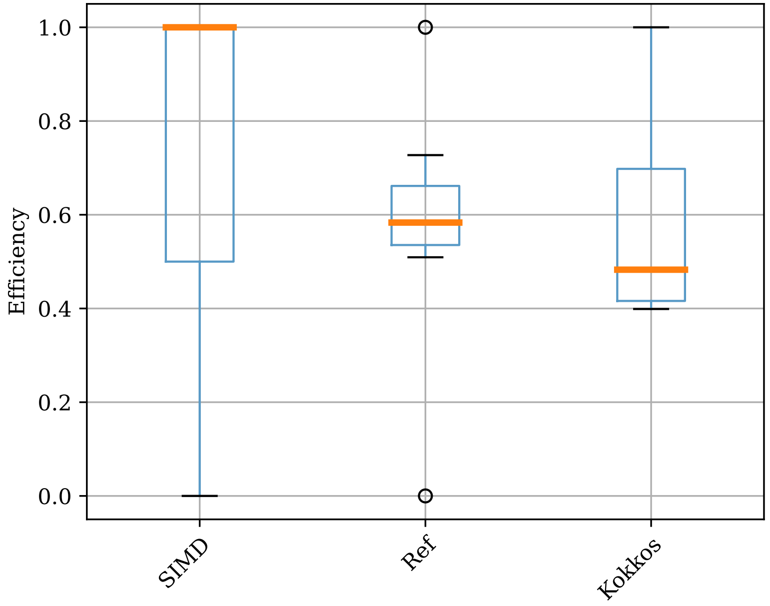


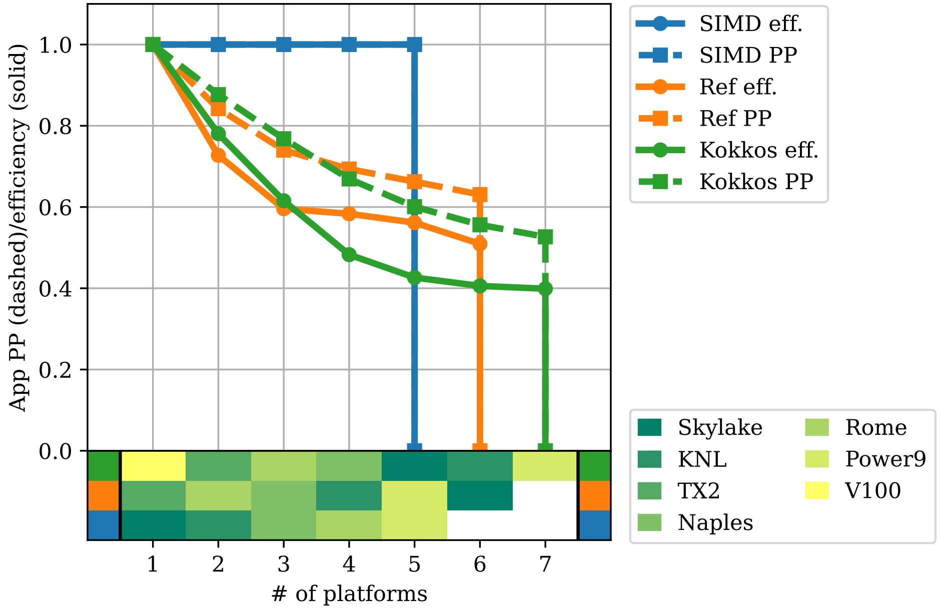
[5] R. Bird, N. Tan, S.V. Luedtke, S. Harrell, M. Taufer, and B. Albright. VPIC 2.0: Next Generation Particle-in-Cell Simulations. IEEE Transactions on Parallel and Distributed Systems, pages 1–1,2021.

Evaluation of Approaches

VPIC/VPIC 2.0

- **Hand implemented vectorised kernels (native)**
- **VPIC 2.0 in Kokkos [5]**



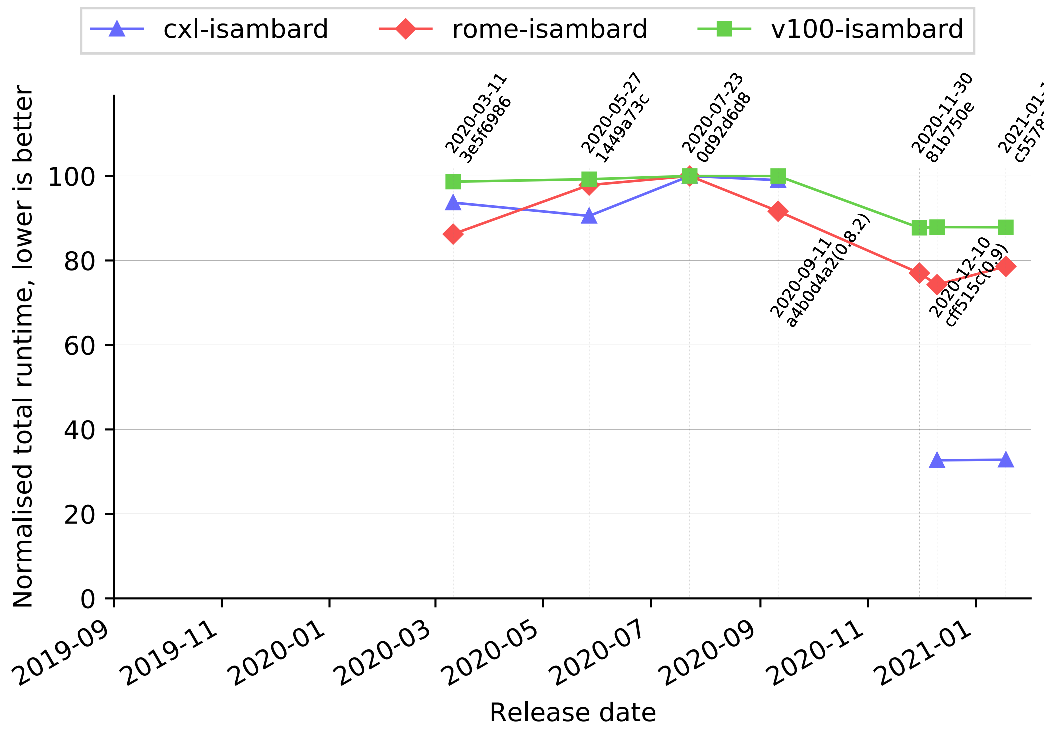


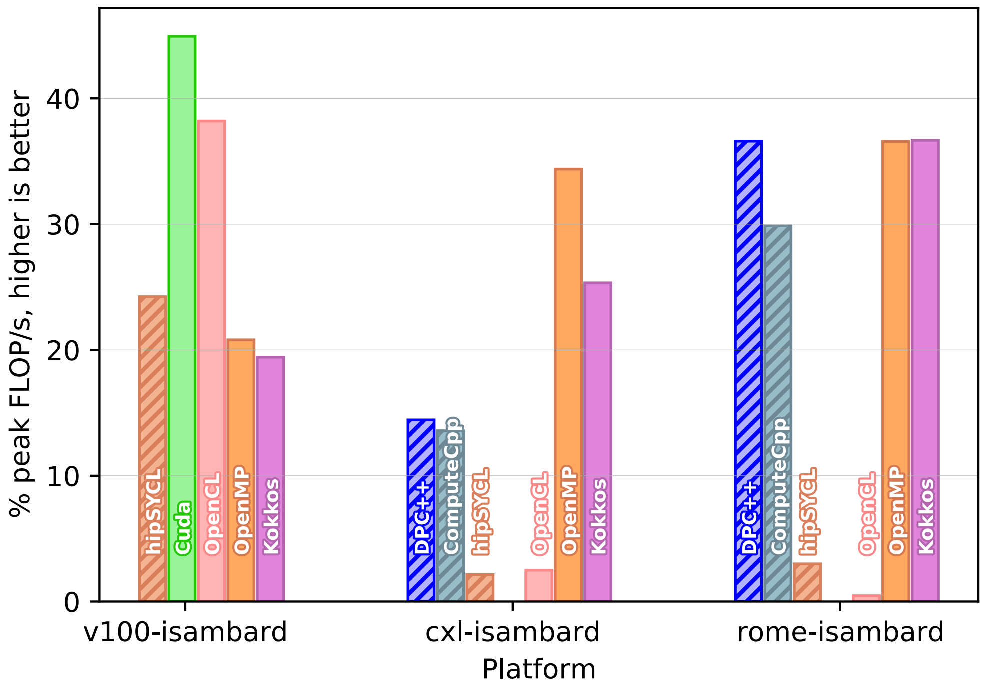
[5] R. Bird, N. Tan, S.V. Luedtke, S. Harrell, M. Taufer, and B. Albright. VPIC 2.0: Next Generation Particle-in-Cell Simulations. IEEE Transactions on Parallel and Distributed Systems, pages 1–1,2021.

Evaluation of Approaches

Additional Results

- **Lin et al. have been tracking SYCL implementation maturity**
- **Evaluate BabelStream, BUDE and Cloverleaf**





[8] W.-C. Lin, T. Deakin, and S. McIntosh-Smith. On measuring the maturity of SYCL implementations by tracking historical performance improvements. In International Workshop on OpenCL, IWOCL'21, New York, NY, USA, 2021. ACM

ExCALIBUR

Evaluation of Approaches

Observations

- Pragma-based approaches (OpenMP, OpenACC)
  - Easiest to implement (or add) 👍
  - OpenMP can offer good portability across heterogeneous platforms 👍
  - … but really this requires different directives based on platform 👎
- Programming model approaches (Kokkos, RAJA, SYCL)
  - Kokkos and RAJA offer similar levels of performance portability 👍
  - SYCL evaluation currently limited, but we expect performance to improve with a change of compiler and compiler maturity 👍
- High-level DSL approaches (OPS, (py)OP2, UFL)
  - OPS/OP2 show good portability, but are parallel pattern based 👍

FM-WP4 Code Structure and Coordination

Support and Coordination

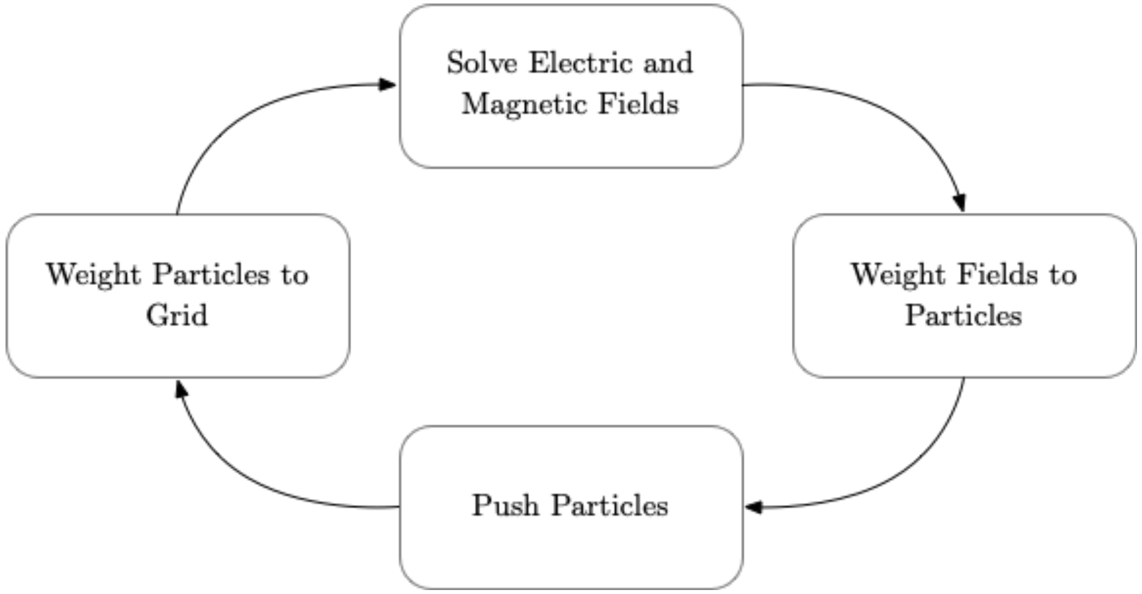
Our second project was also based around four deliverables:

- Periodic revision of Hardware and Software Reviews
- Development of an FEM-PIC mini-application
- Performance evaluation of FEM-PIC application
- Outline proposal for a particle-methods DSL

Developing an Unstructured PIC Mini-app

FEM-PIC

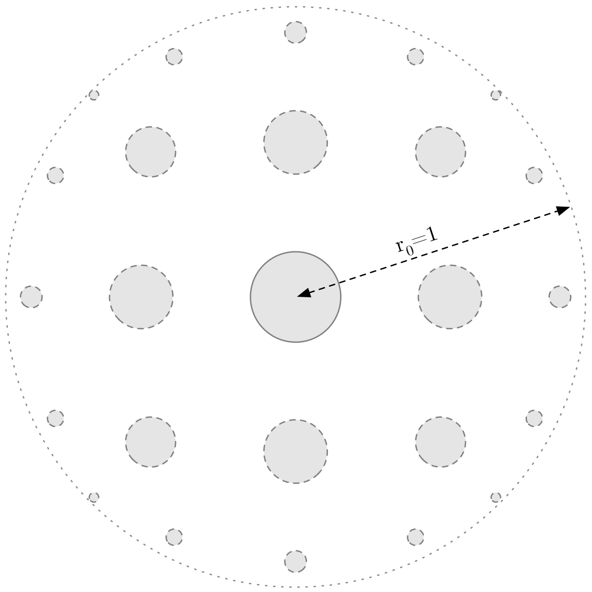
- Our evaluation has been limited to DoE PIC codes
  - All developed with Kokkos
  - All structured grid (except EMPIRE-PIC, but this is export controlled ☹️)
- Unstructured grid adds complexity that cannot easily be captured
  - Particle push necessitates a gather and scatter



Developing an Unstructured PIC Mini-app

FEM-PIC

```c
struct particle {
    double part_p[3];
    double part_pos[3];
    double weight;
    double charge;
    double mass;
    struct particle *next;
    struct particle *prev;
    int id;
};
```

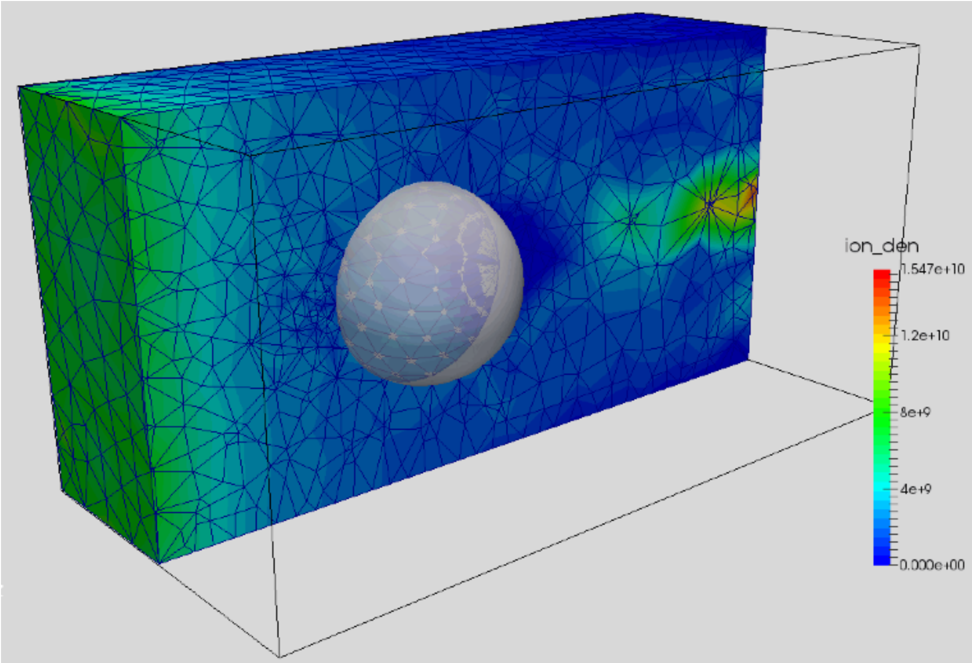


- **Goal is to add particles to an existing mini-application**
    - **Mimic storage used by EPOCH**
    - **Extract performance data from three key steps in particle movement**
- **~1000-3000 LOC**
    - **Enables rapid porting and evaluation**
    - **Enables analysis of novel storage techniques, particle shapes etc**

Developing an Unstructured PIC Mini-app

Progress

- Identified two candidate "base" applications
  - FEM implementation based on "Computational Physics" by Jos Thijssen
  - FEM-PIC from particleincell.com
- FEM-PIC likely the best candidate, but work needed to make it "look" more like EPOCH and EMPIRE



FM-WP4 Code Structure and Coordination

Support and Coordination

Our second project was also based around four deliverables:

- **Periodic revision of Hardware and Software Reviews**
- **Development of an FEM-PIC mini-application**
- **Performance evaluation of FEM-PIC application**
- **Outline proposal for a particle-methods DSL**

Particle-methods DSL

Outline Proposal

- Many DSLs available for mesh-based codes
- Very few for particle-based codes
    - Notable examples include:
        - CoPA/Cabana — a library for data storage and manipulation of particle data
        - HartreeParticleDSL — Particle methods DSL, using Regent programming language
        - PPMD — Particle DSL mainly for particle-particle interactions (e.g. molecular dynamics)

Particle-methods DSL

Outline Proposal

- **Need for a Particle DSL that can interact with a Mesh DSL**
- **Following development of FEM-PIC mini-application:**
  - **Field solver will be redeveloped using OP2**
  - **Sketch out requirements for a DSL for particles**
  - **Proposal of an API**

PROJECT NEPTUNE

WORK PACKAGE 4: SUPPORT AND COORDINATION

Steven Wright, Ben Dudson, Peter Hill, David Dickinson, Edward Higgins, Chris Ridgers

University of York

Gihan Mudalige, Tom Goffrey, Ben McMillan

University of Warwick

Met Office

UK Research and Innovation

UK Atomic Energy Authority