Performance Analysis of Massively-Parallel Computational Fluid Dynamics

11th ICHD, Singapore,

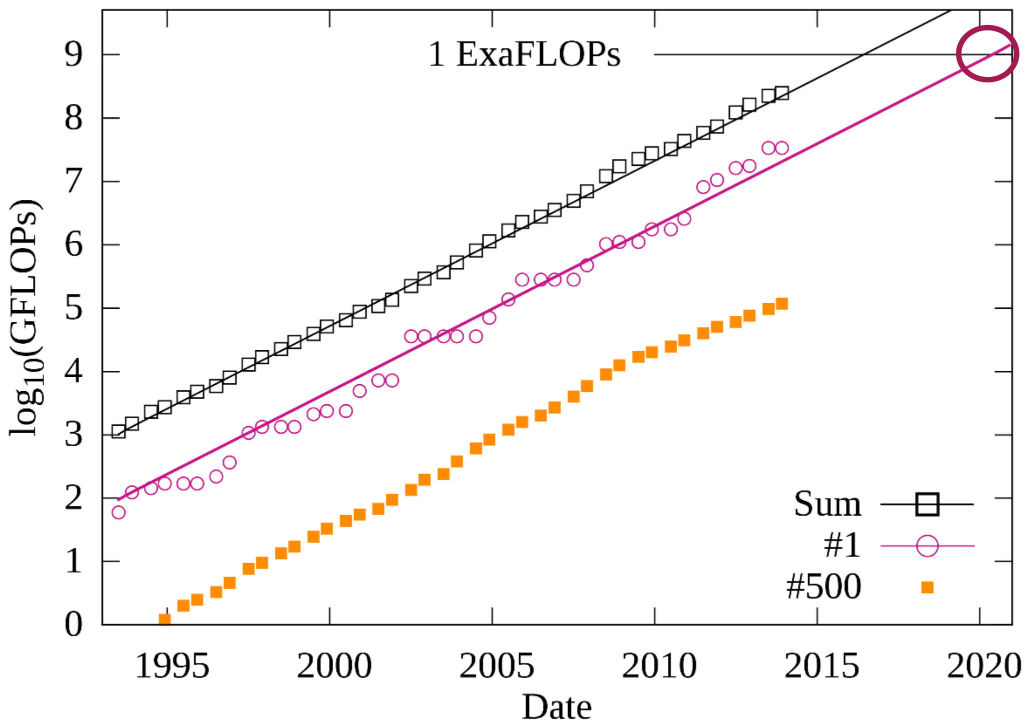
J. Hawkes

Supervisors - Prof. S. R. Turnock | Prof. S. J. Cox | Dr. A. B. Philips | Dr. G. Vaz

19th-24th Oct., 2014

Background

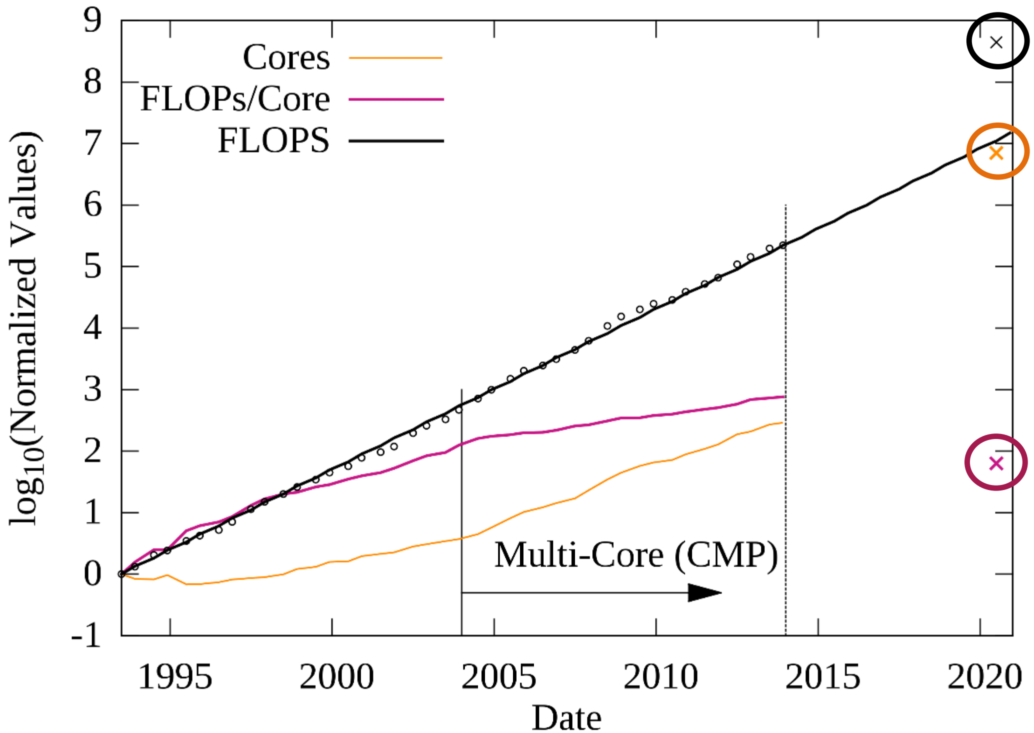
- Exascale computing…
  - First computer capable of 10^{18} FLOPs in 2020.



GFLOPs of the Top500 (#1, #500 and Sum)

Background

- ... But its architecture will be completely different to current supercomputers.
  - Limited by power consumption, rather than hardware speed.
  - Shift towards MASSIVE CONCURRENCY.
  - Mostly intra-node concurrency (shared memory).



Average of the Top500 supercomputers, with the first exascale machine superimposed.

The Strong Scalability Problem

- We've been used to weak scaling – making our problems larger to match increases in concurrency.
- Massive increase in concurrency makes this impractical - we need to split our problems into smaller chunks (strong scaling).
- Memory capacity growing half as fast as overall compute power; finite cap on problem size grows slower than core-count.
- Most challenging CFD lies in unsteady simulations. We can only parallelize spatial dimensions, and we have to do this more to increase our unsteady simulation capabilities.

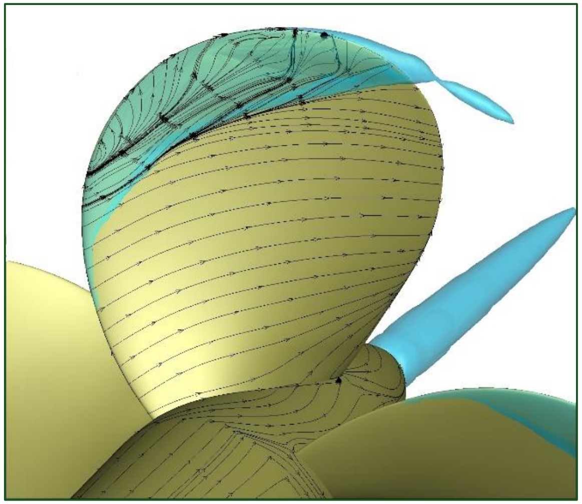
Objectives

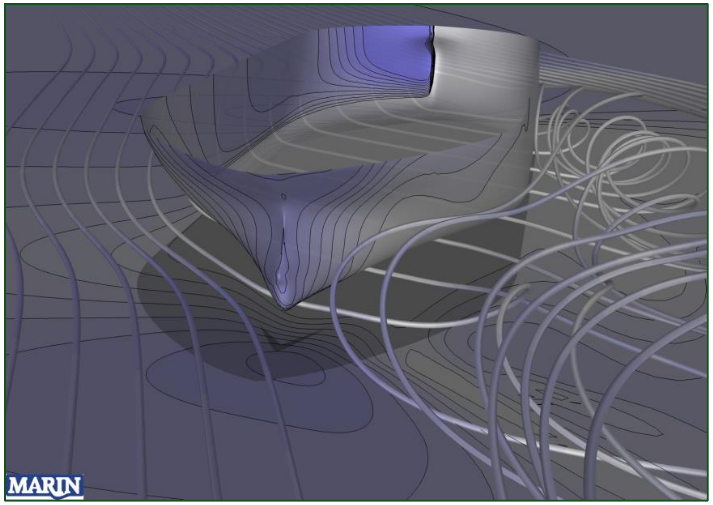
- Investigating limitations to strong scalability
- Considering a range of common user settings to cover a broad range of applications
  - Discretization Schemes
  - Turbulence Models
  - Grid Structure
  - Linear Equation-System Solvers

Experimental Setup

ReFRESCO

- Maritime-optimized RANS code
- Various Eddy-Viscosity models
- State-of-the-art sliding, deforming and adaptive meshes
- Developed at MARIN (Netherlands), IST (Portugal), USP-TPN (Brazil), TUDelft (Netherlands), University of Southampton (United Kingdom)

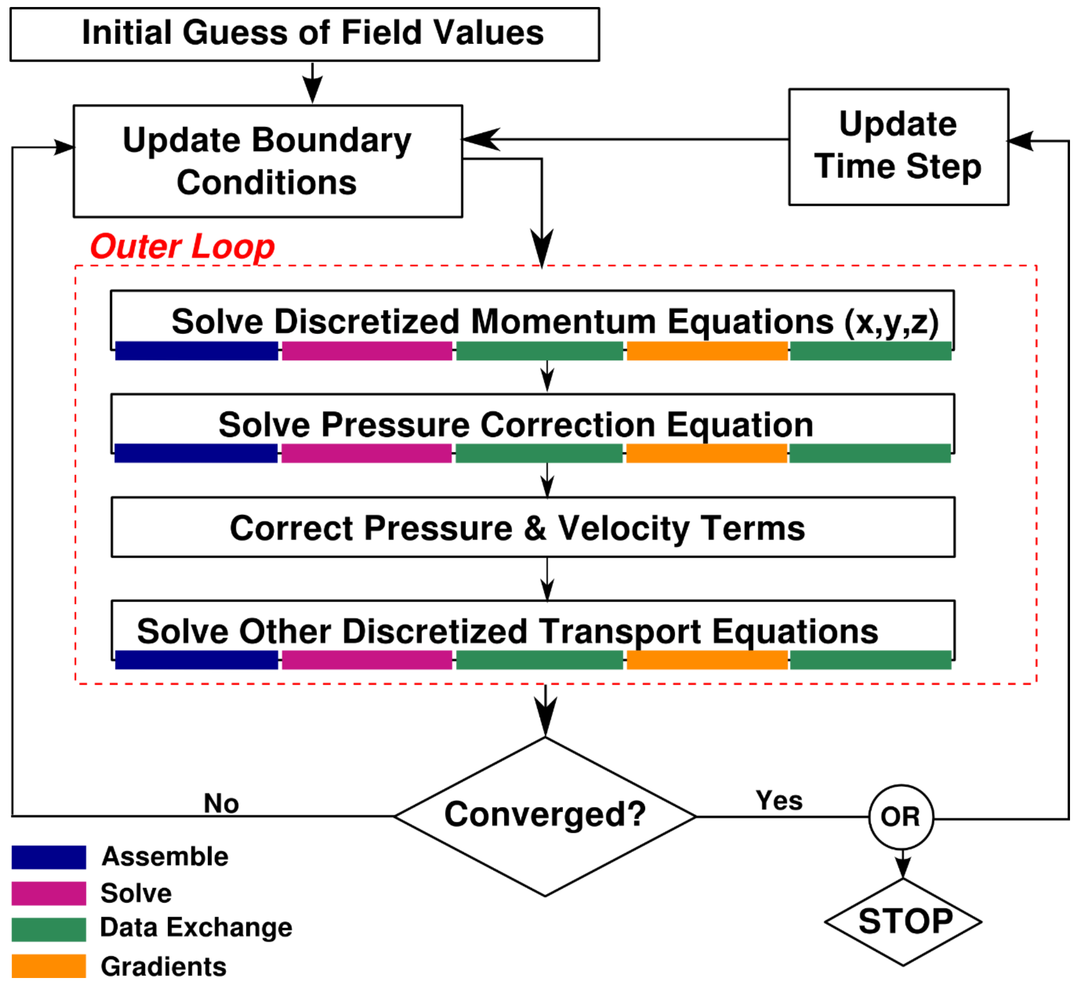




ReFRESCO Cavitation and Manoeuvring simulations [marin.nl]

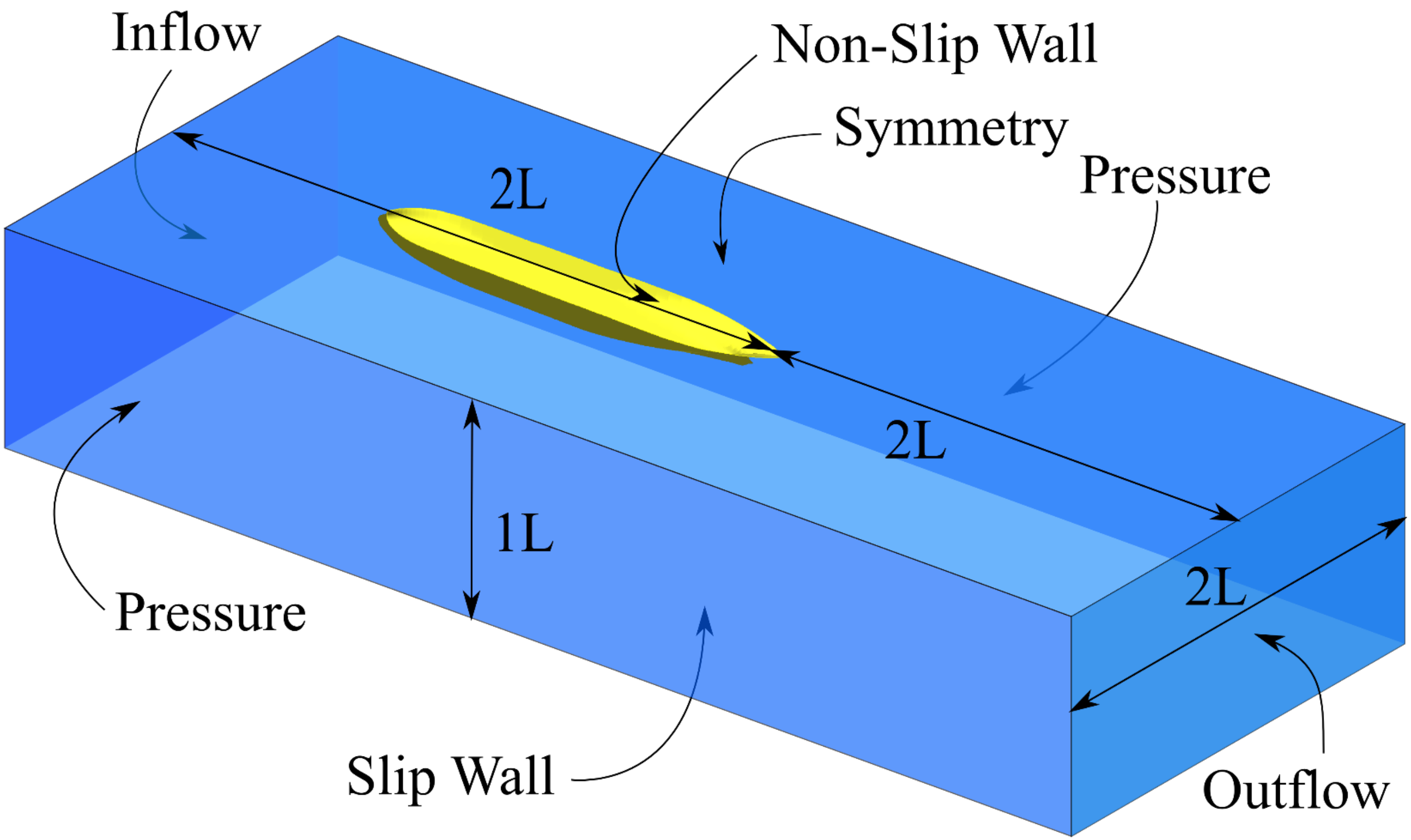
Profiling

- Run-time of SIMPLE split into its key parts.
  - Assembly of Linear Systems
  - Solution to those systems
  - MPI Data Exchange
  - Calculations of Gradients
  - Other



KVLCC2

- Double-body wind-tunnel simulation



Numerical Setup

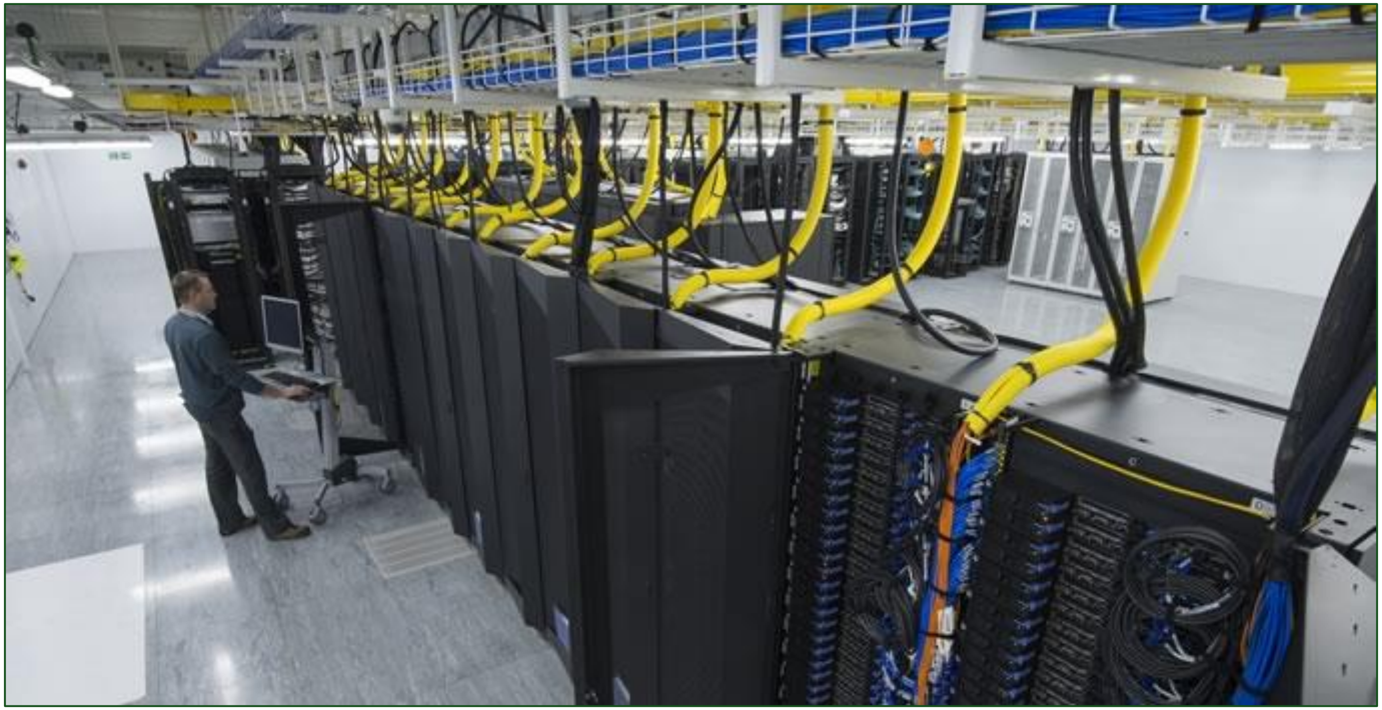
- Segregated Solver
- 2.67m cells
- k-ω shear stress transport (SST-2003)

- Momentum/Turbulence Equation
  - Block Jacobi + GMRES
  - ILCT 1%
  - Explicit 0.15, Implicit 0.8-0.85 (ramped)
  - QUICK or Upwind (1^{st} Order)

- Pressure Equation
  - ML + GMRES
  - ILCT 1%
  - Explicit 0.1

IRIDIS4

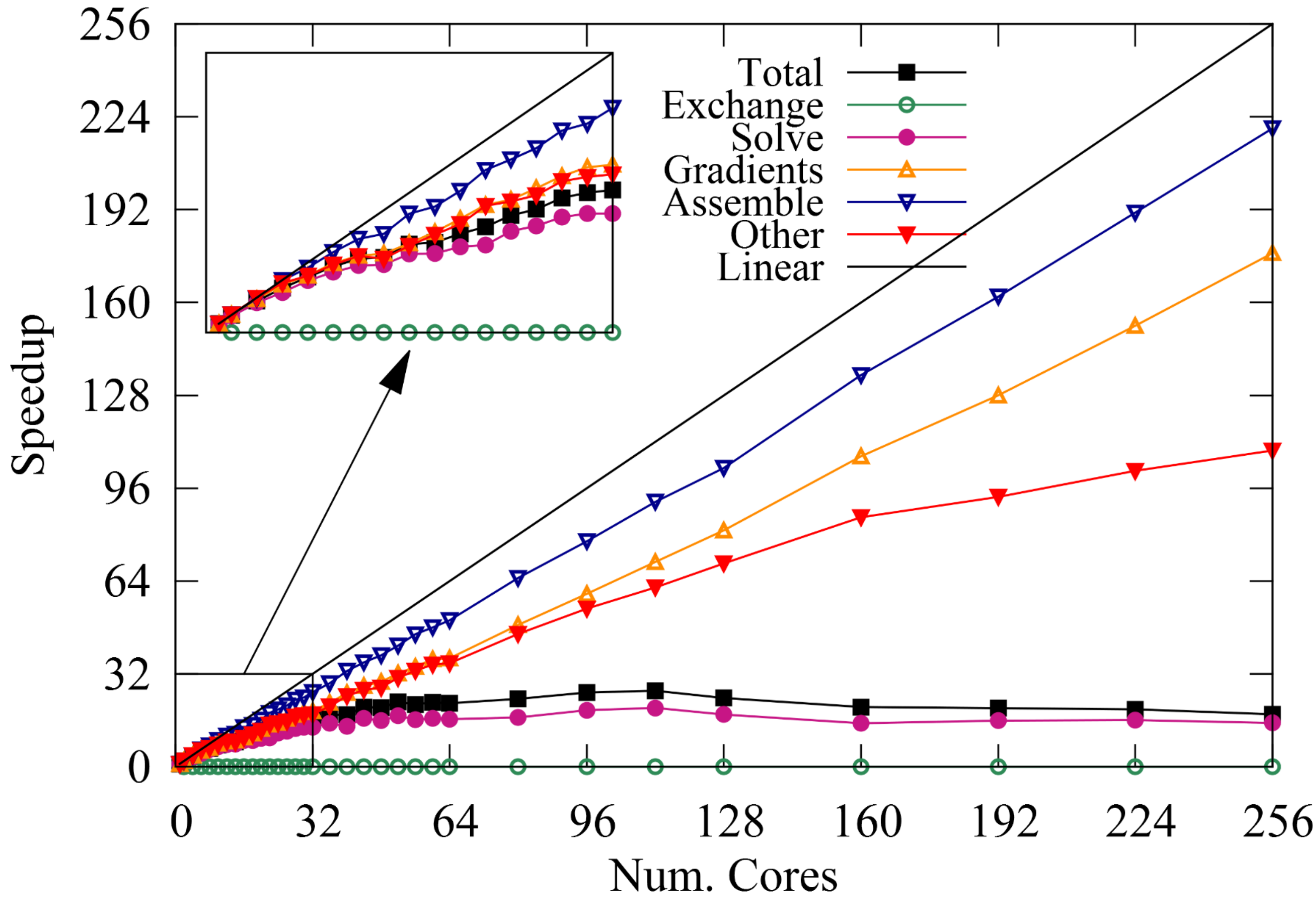
- University of Southampton Supercomputer
- #179 on Top500 in November 2013
- 12,200 cores at 2.6Ghz (16 cores per node)



[UoS, Computational Modelling Group – cmg.soton.ac.uk]

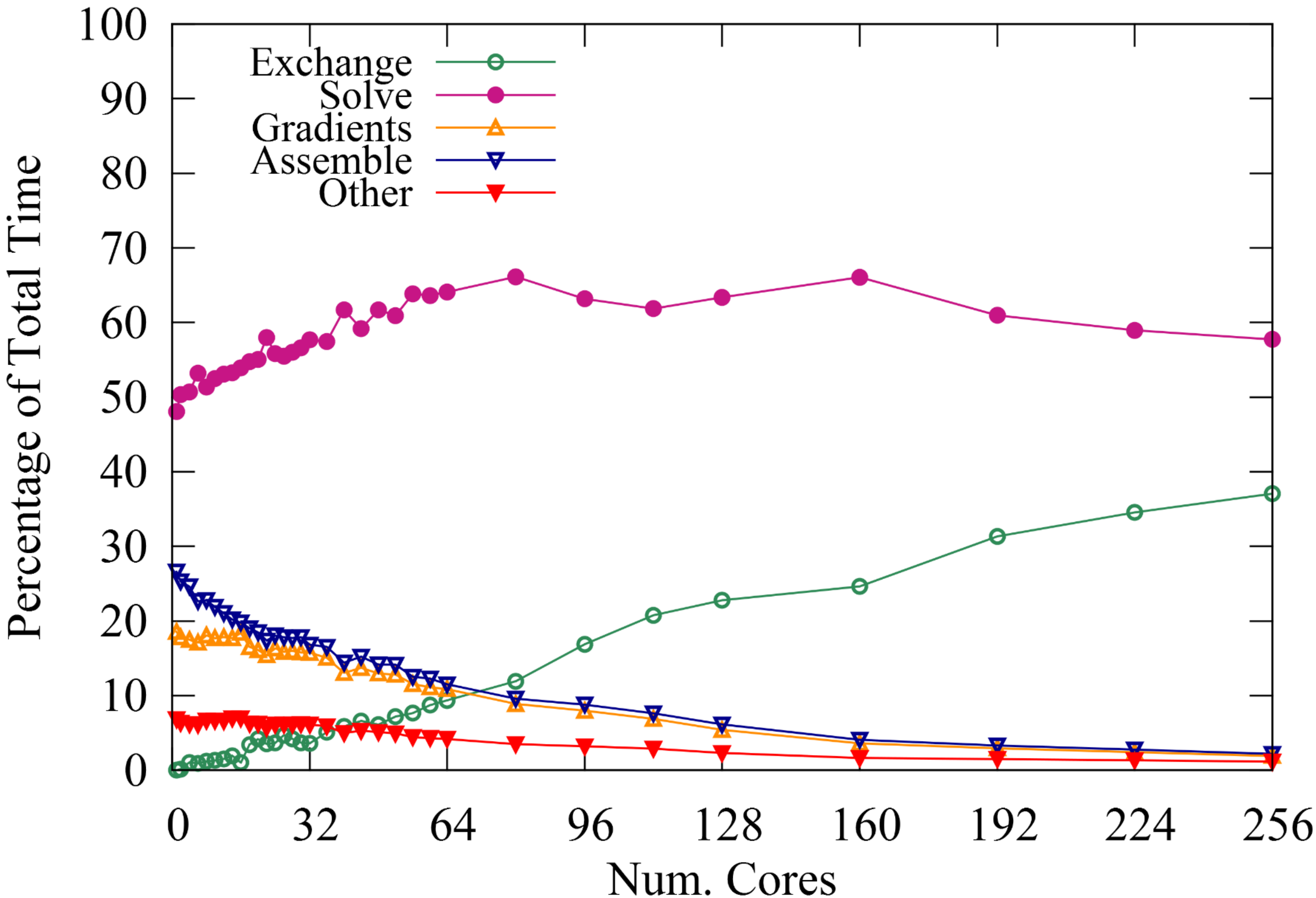
Results

↳ Basic Scalability



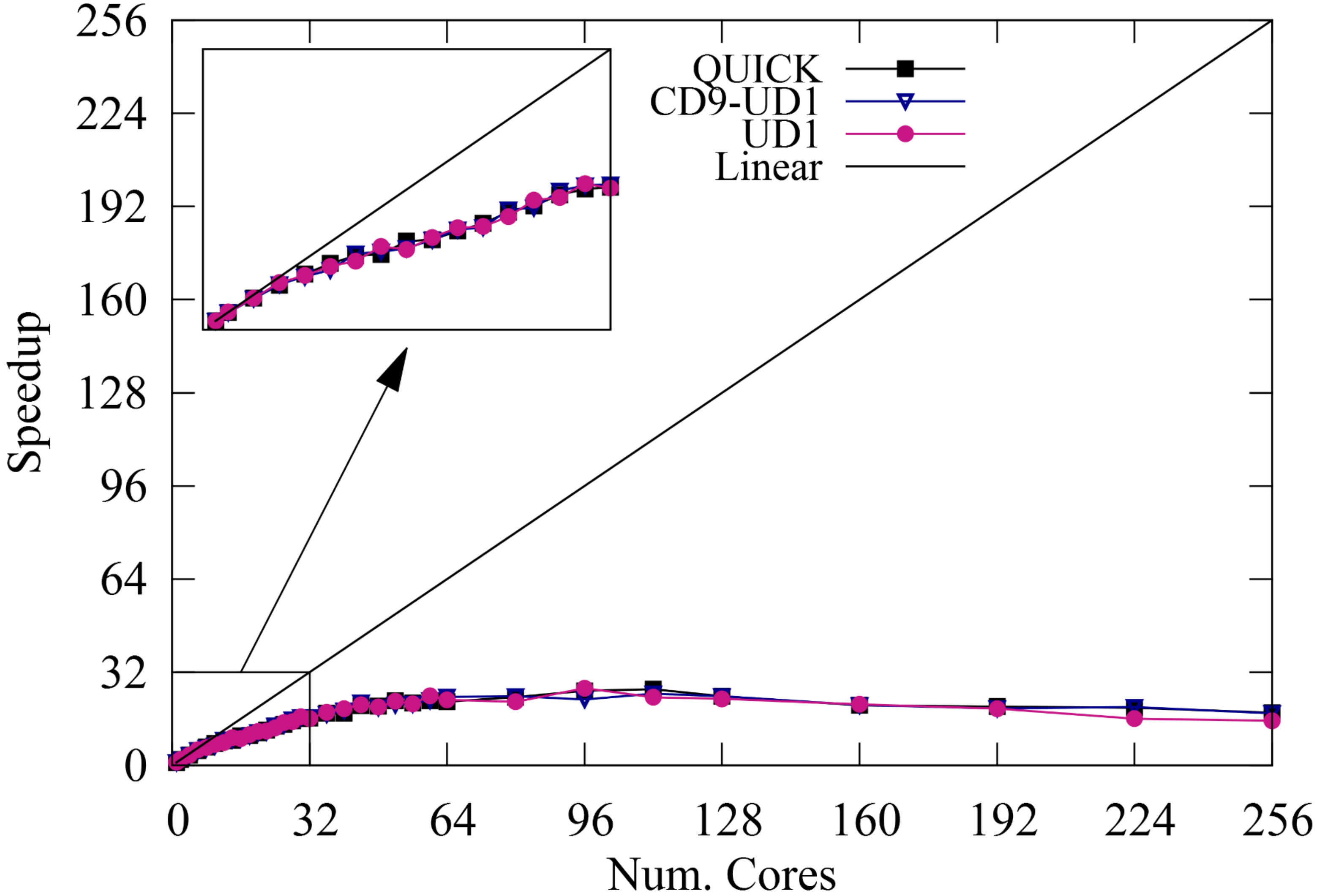
Scalability of profiled routines.

Basic Scalability



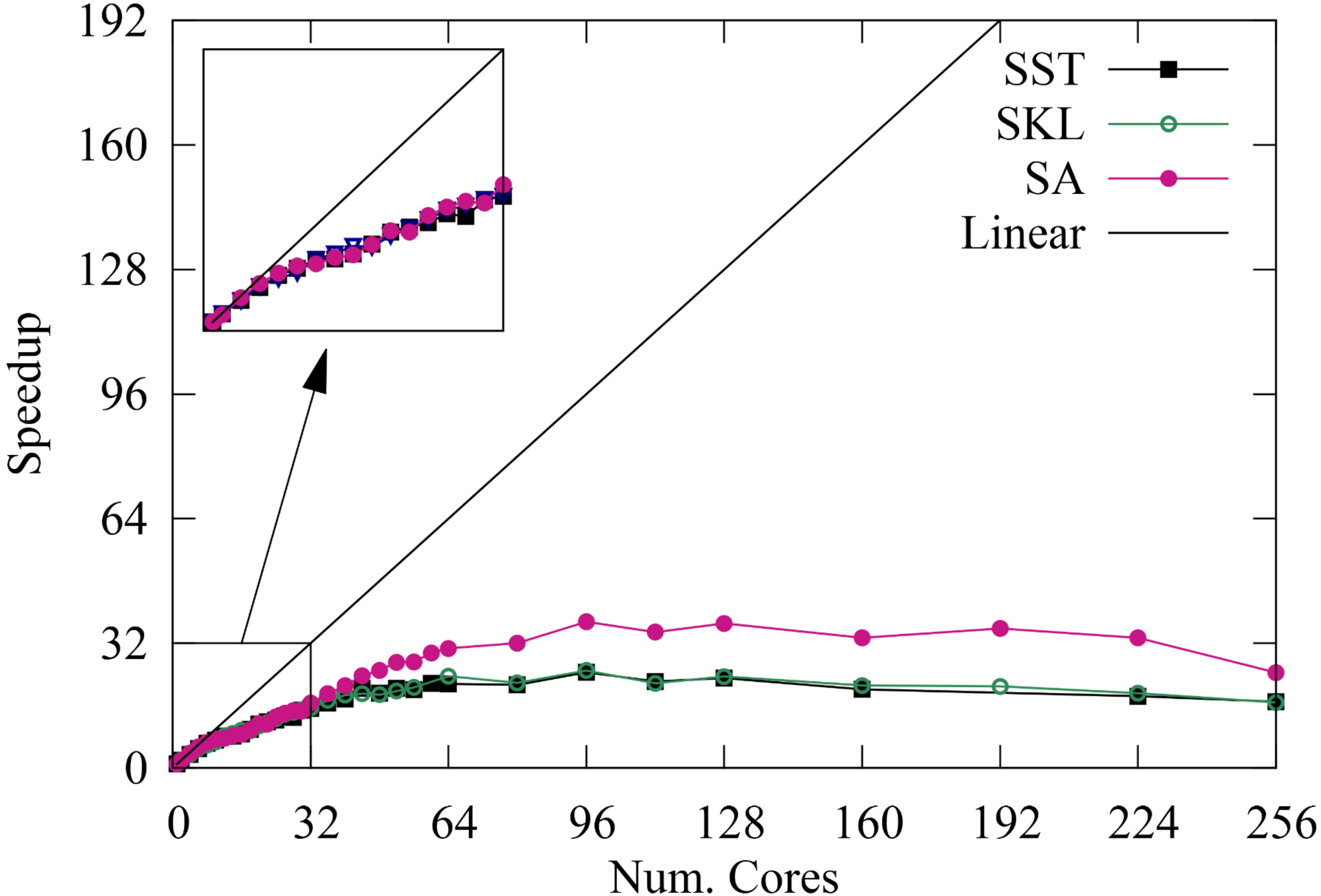
Proportions of time spent in profiled routines.

Convective Discretization Scheme



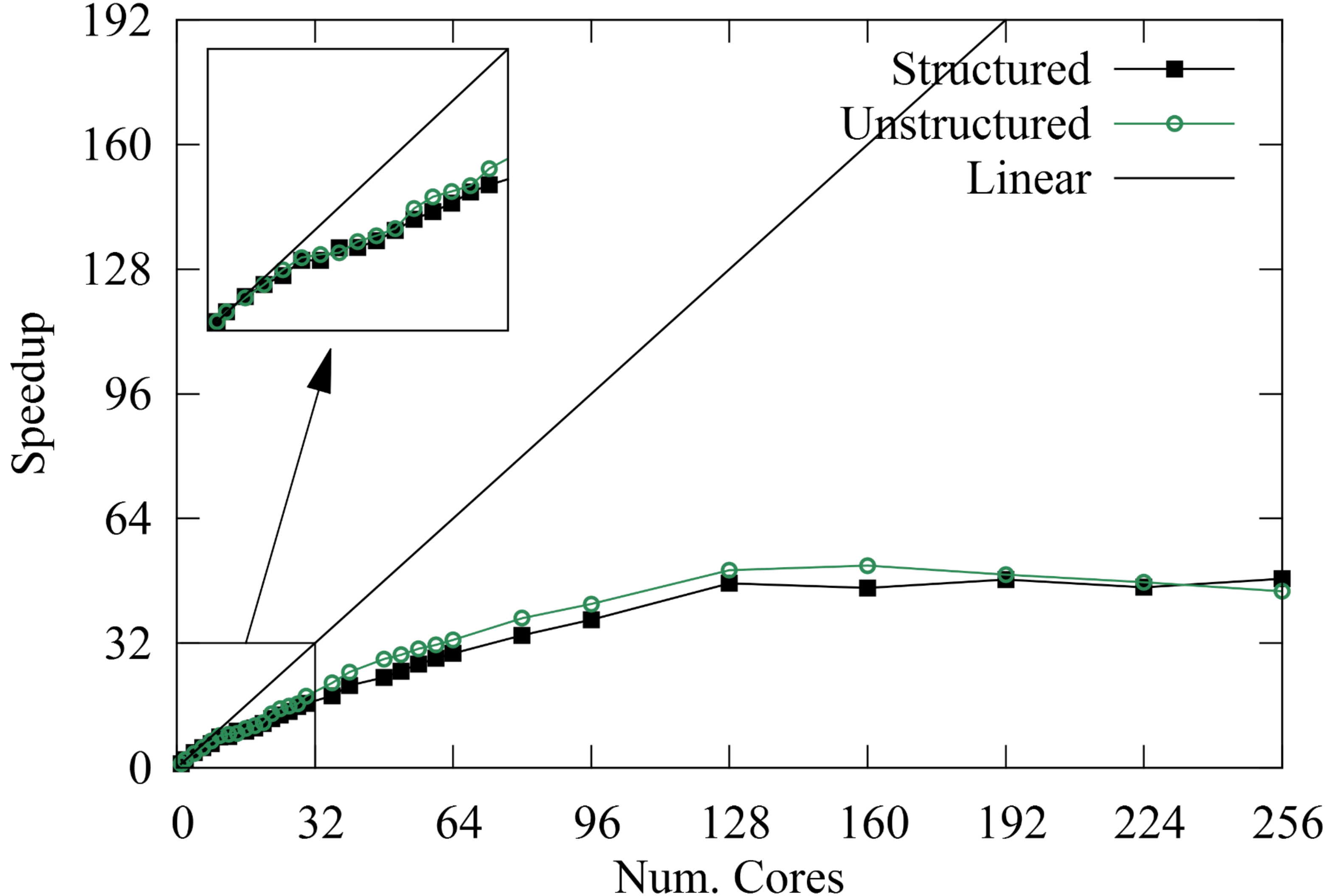
Scalability of total runtime with various convective discretization schemes.

↳ Turbulence Models



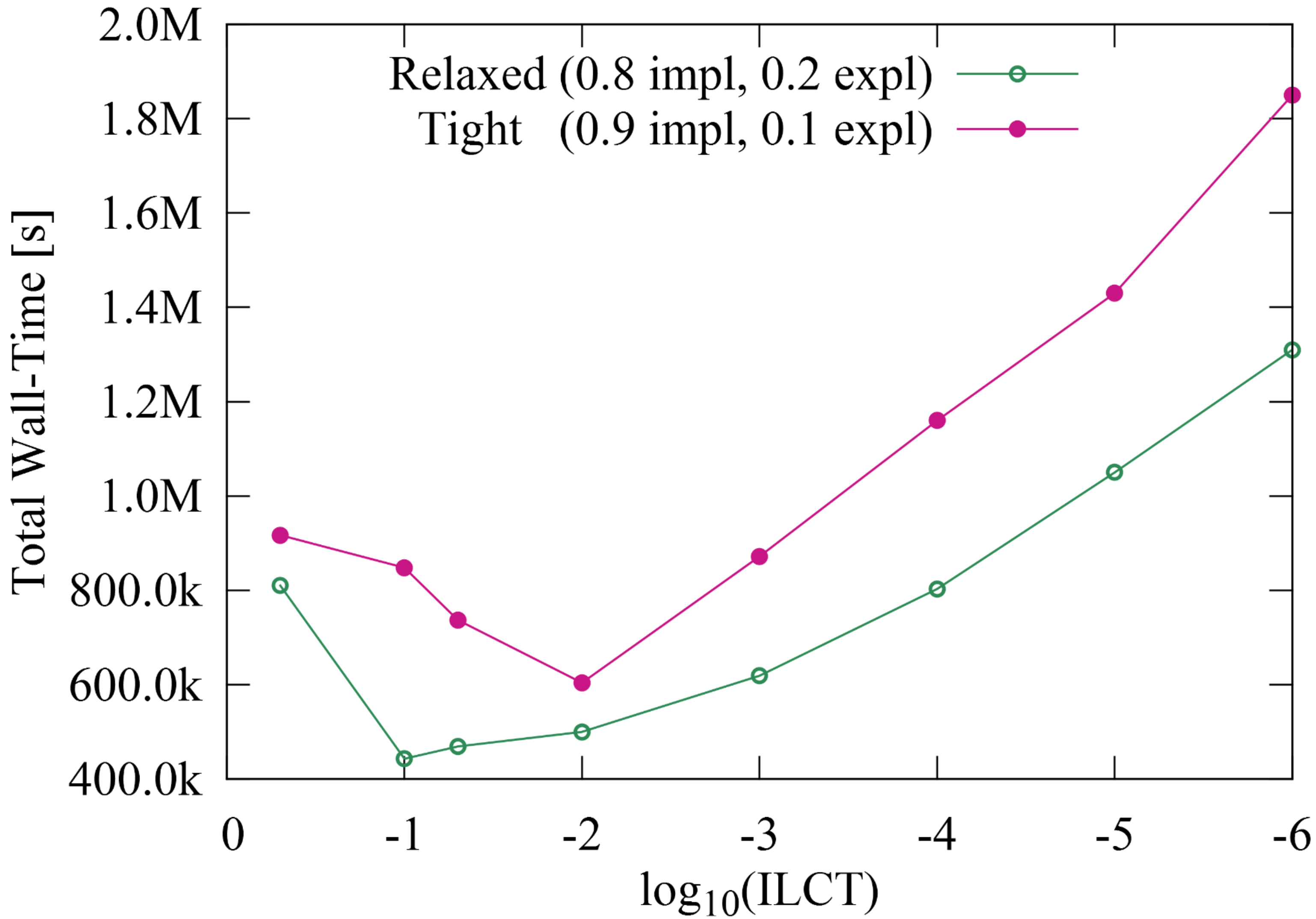
Scalability of total runtime with various turbulence models.

Structured vs. Unstructured



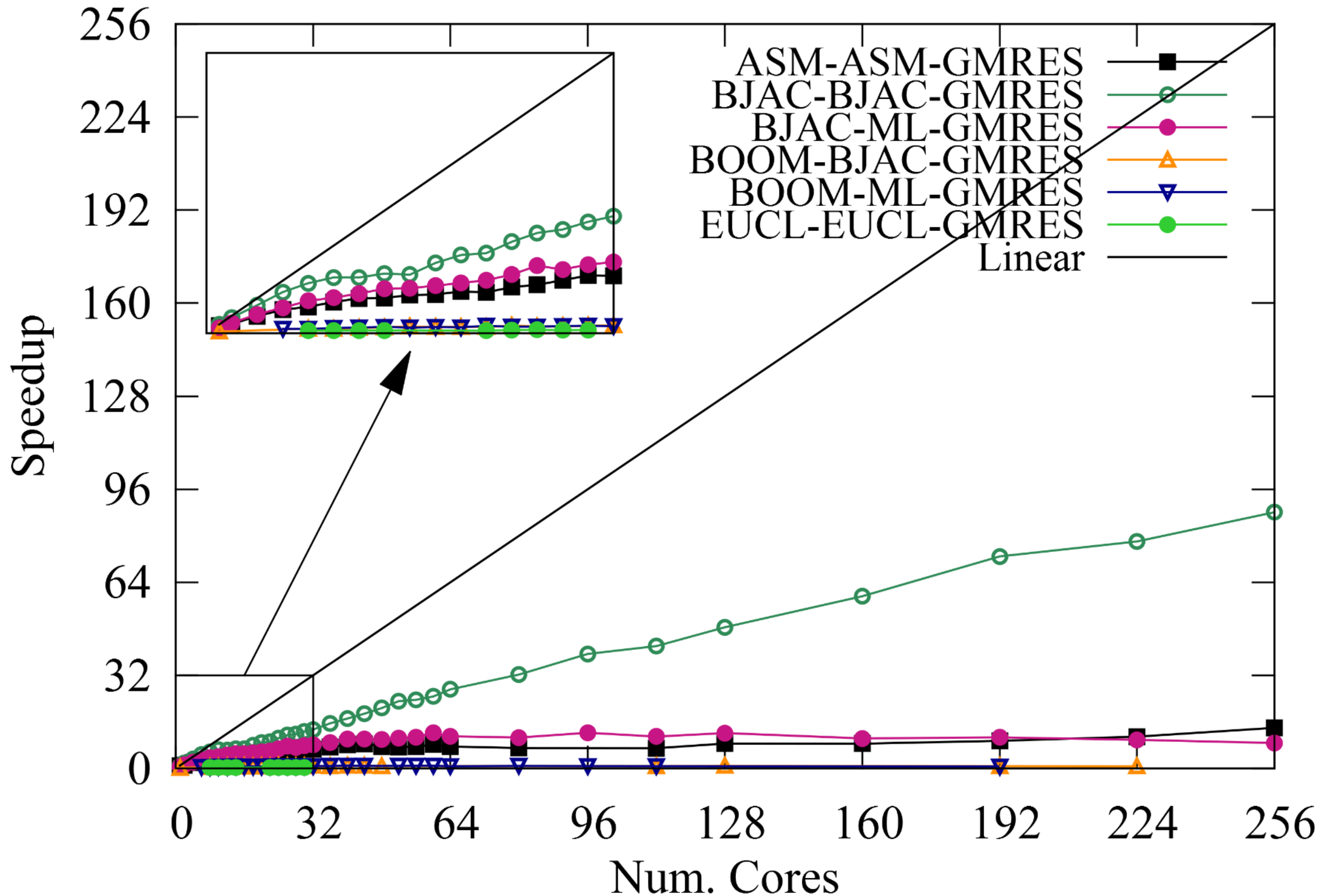
Scalability of total runtime comparing a structured grid to an unstructured grid. Some interpolation error.

Linear Equation-System Solvers



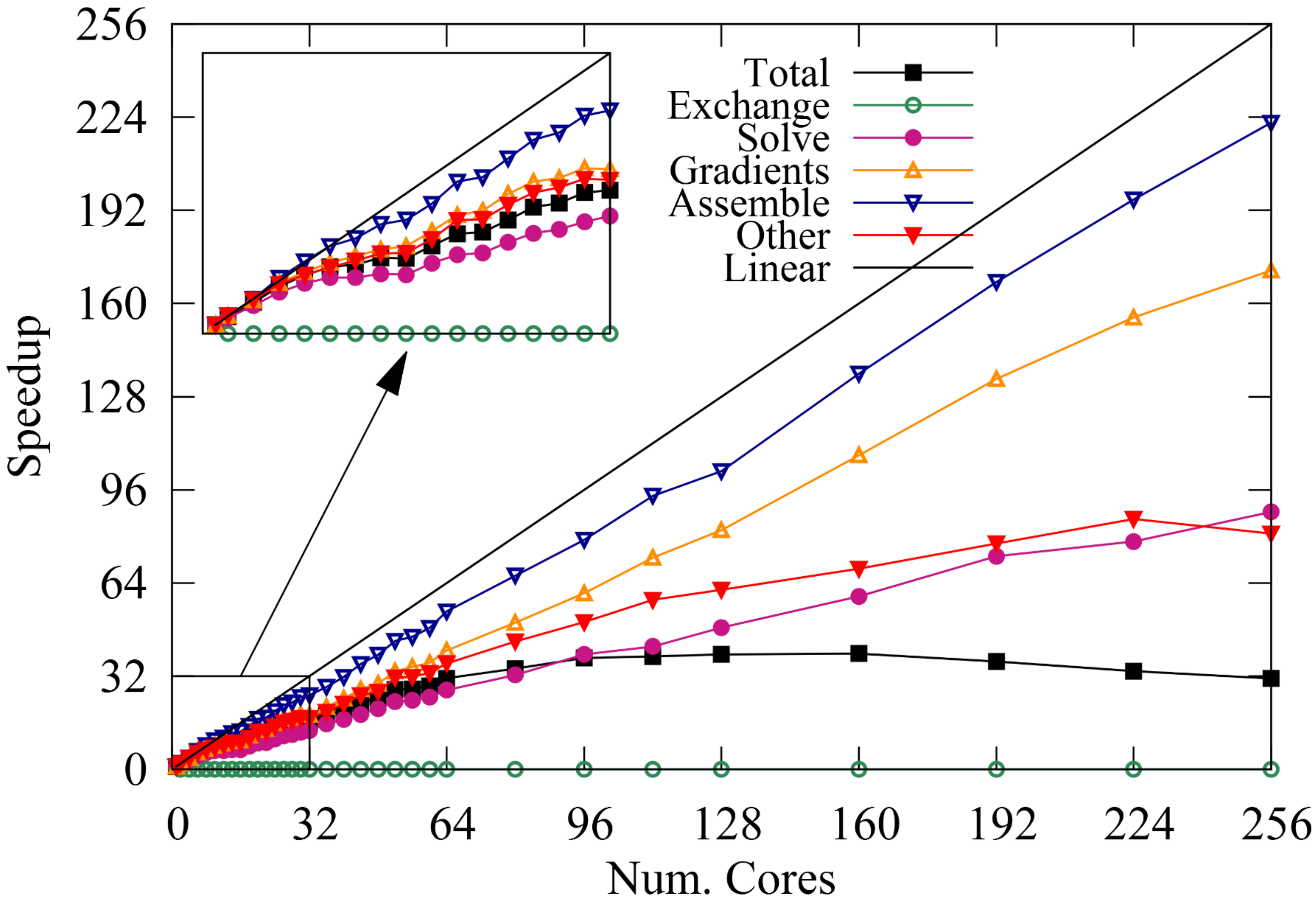
Effect on total run-time of Inner Loop Convergence Tolerance with various outer loop relaxation factors.

Linear Equation-System Solvers



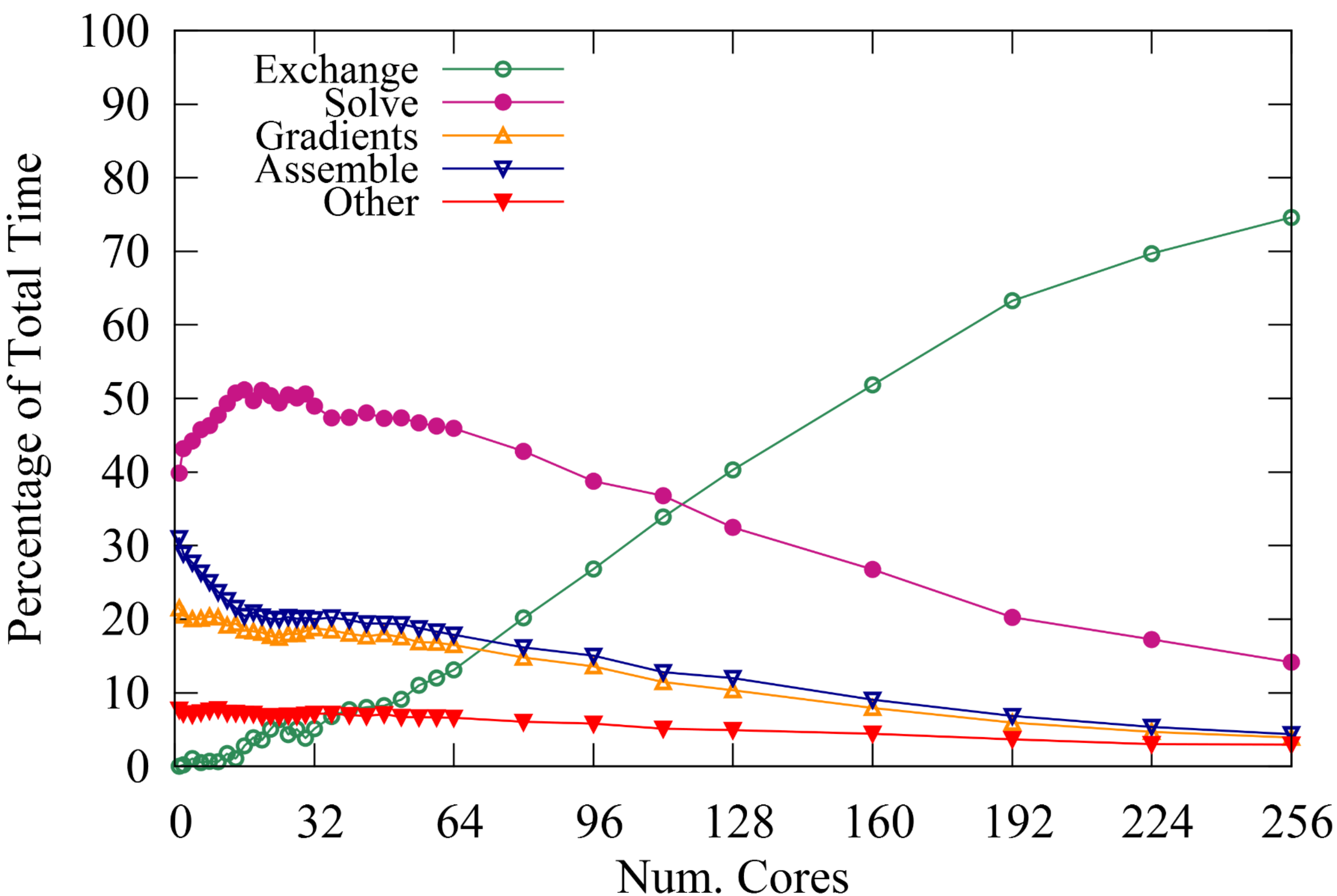
Scalability of solve routines with varying pre-conditioner. BCGS and GMRES show similar trends.

↳ Optimized Scalability



Scalability of profiled routines.

Optimized Scalability



Proportions of time spent in profiled routines.

Conclusions

Conclusions

- Discretization scheme had little effect on overall scalability, *assembly* routines are not a bottleneck.
- Two-equation Eddy-Viscosity models were considerably more expensive and non-scalable than one-equation models – mostly due to non-scalable *solve* routines.
- Mesh structure had little effect on overall scalability.
- Choice of pre-conditioner had a large effect on the *solve* routines, but still a bottleneck.
- *MPI Data Exchange* is a bottleneck at higher node-count.

- *Solve* routines are a bottleneck to shared-memory parallelization.
- *Exchange* routines are a bottle neck to distributed-memory parallelization.

Thank You

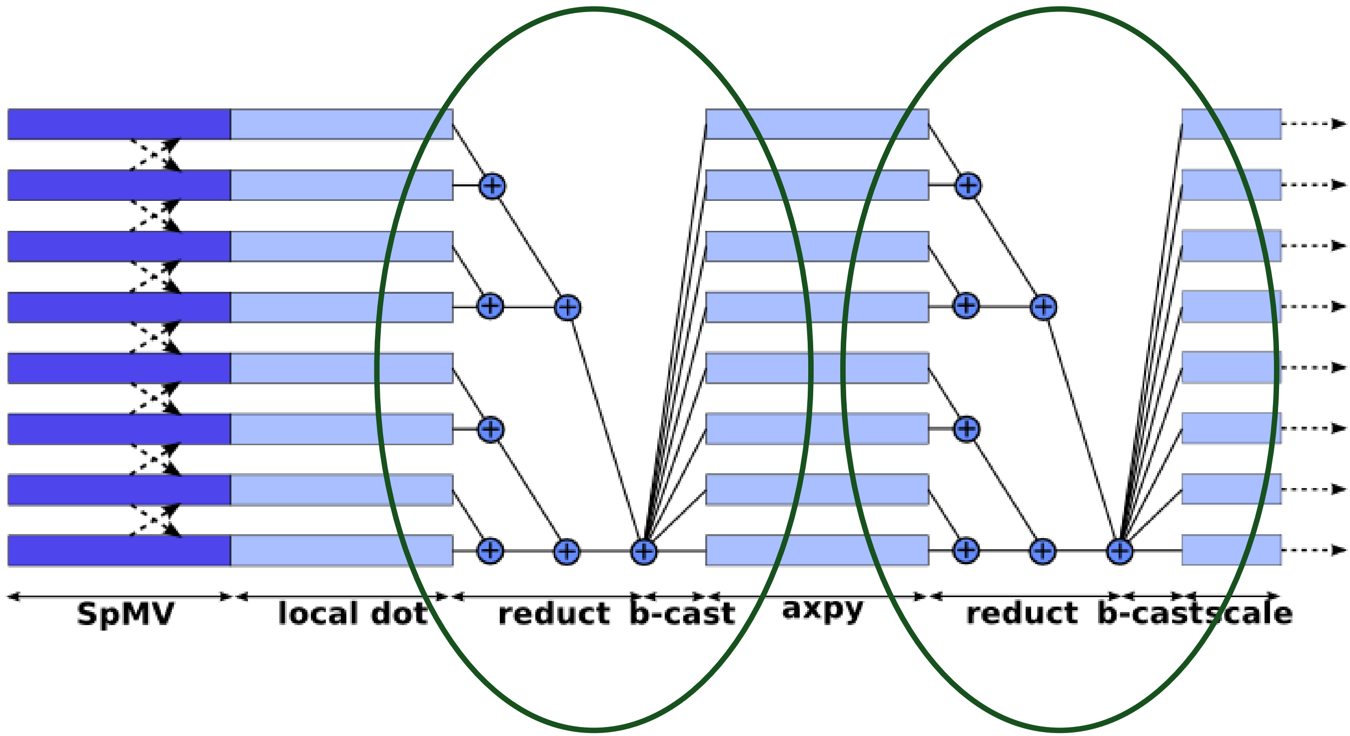
J. Hawkes

J.Hawkes@soton.ac.uk

Further Work

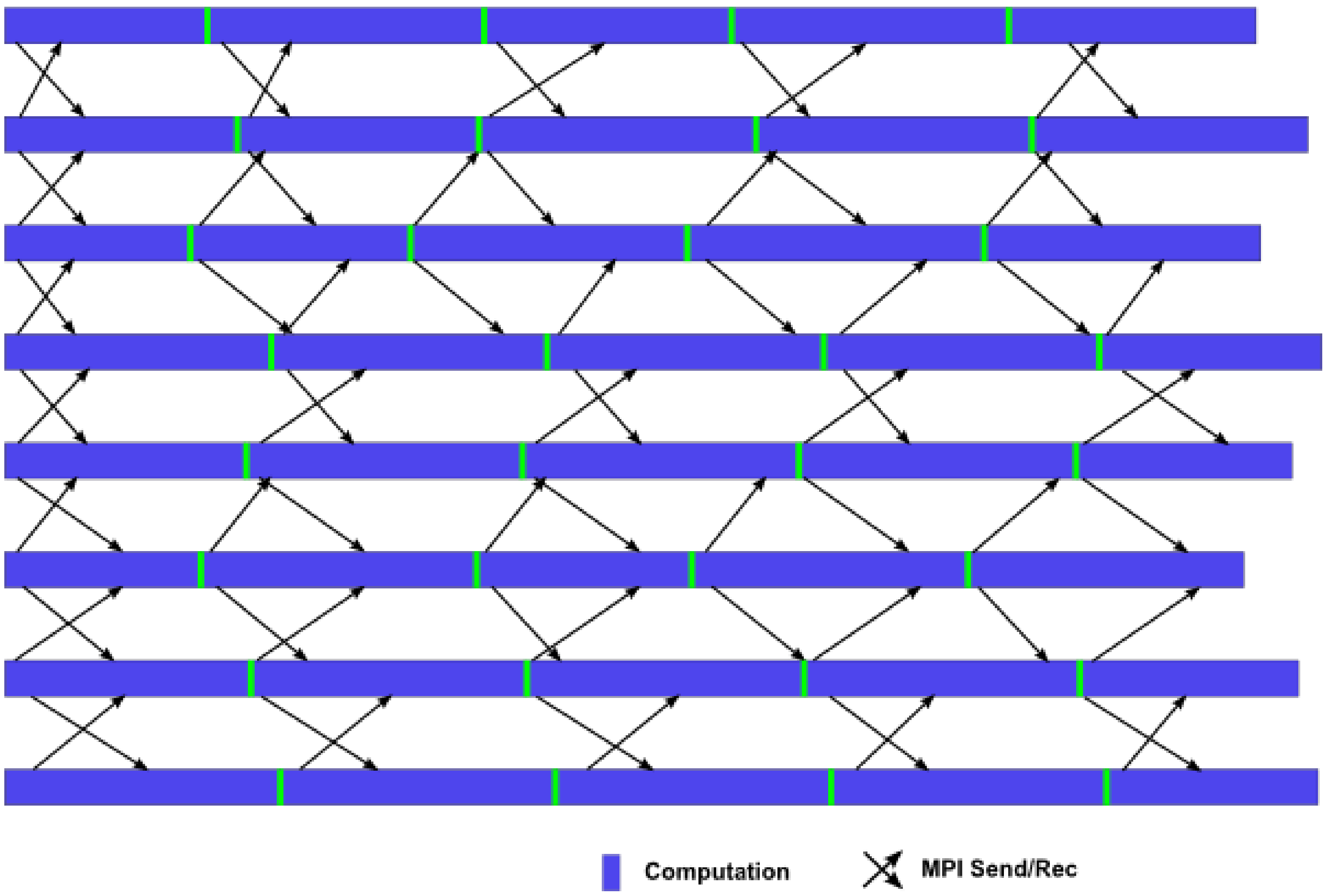
↳ So Why Isn't the Solver Scalable?

- Most codes use a Krylov Subspace (KSP) method such as GMRES.



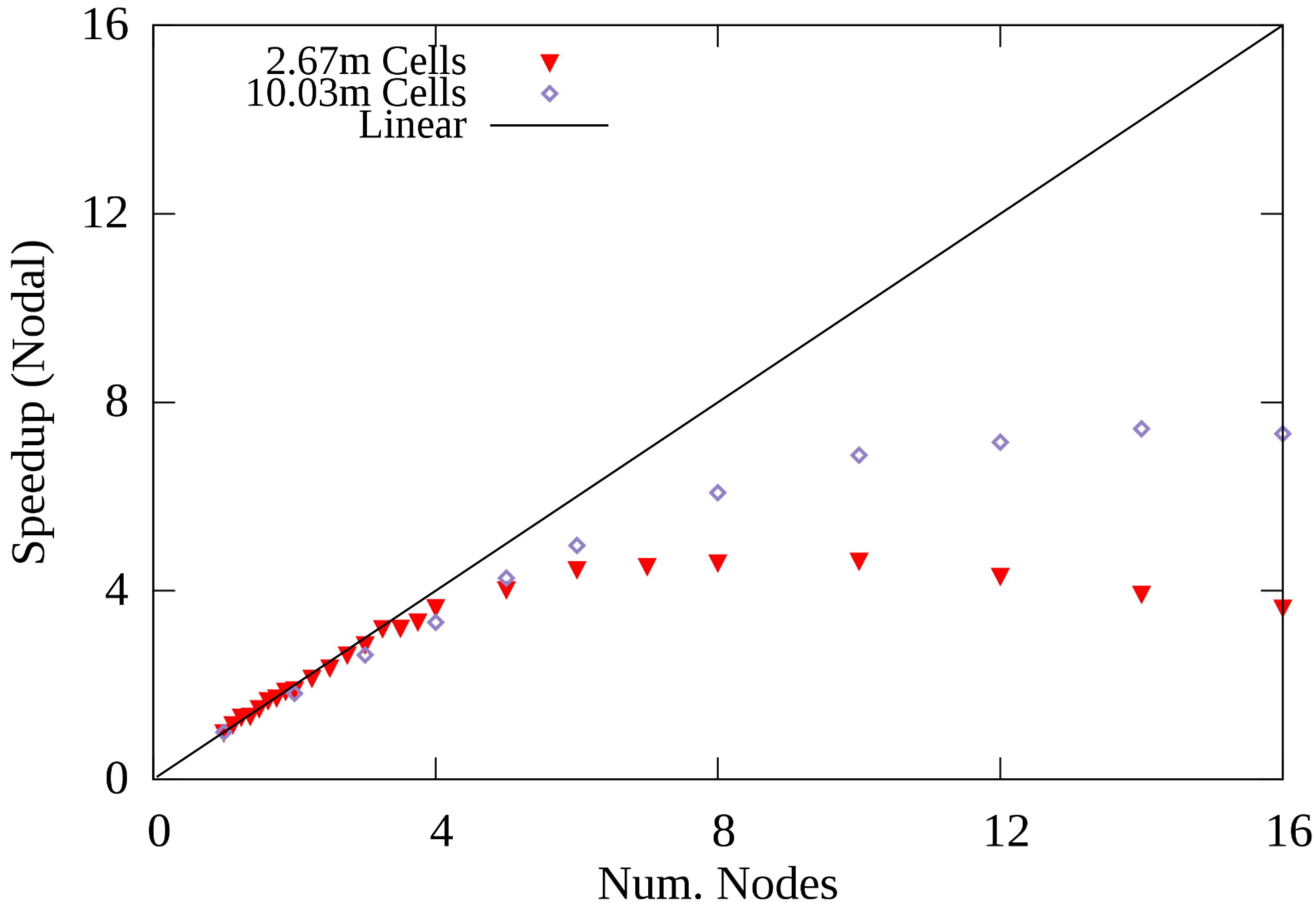
Chaotic Iterative Methods

- CIM: Maximum use of computation and communication.



Q & A

Nodal Speedup



Nodal speed-up up to ~80k cells per core. Can achieve higher with lower processes-per-node.