Fast Fourier Transformations for gyrokinetic plasma simulations

Joachim Hein

In collaboration with Colin Roach (PI), Wayne Arter, Michael Barnes, Greg Colyer, Nuno Loureiro

EPCC, University of Edinburgh

Outline

- Magnetic confinement fusion research
- Simulating plasma turbulence
- Gyrokinetic code GS2
- Profiles from GS2
- Performance FFTW version 2 and version 3
- HECToR floating point units (SSE instructions)

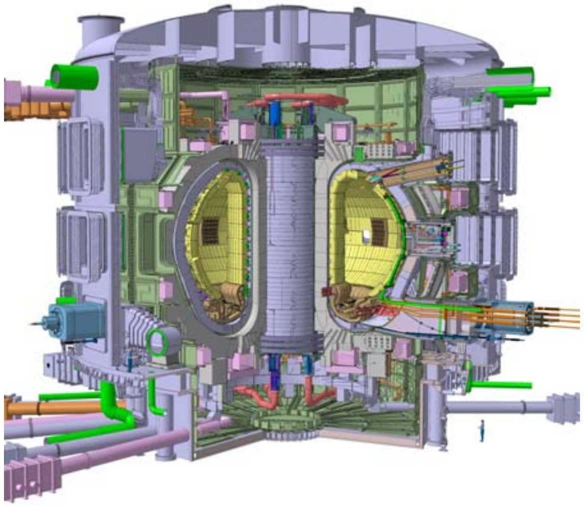
Magnetic confinement fusion reasearch

- Thermonuclear fusion powers stars
- Great potential to generate energy
- Most advanced technical path: Tokamak
  - Particles are confined by a magnetic field (≈ 5 Tesla)
  - Culham: JET and MAST
  - ITER: next generation Tokamak in Cadarache France
- Focus on nuclear reaction:

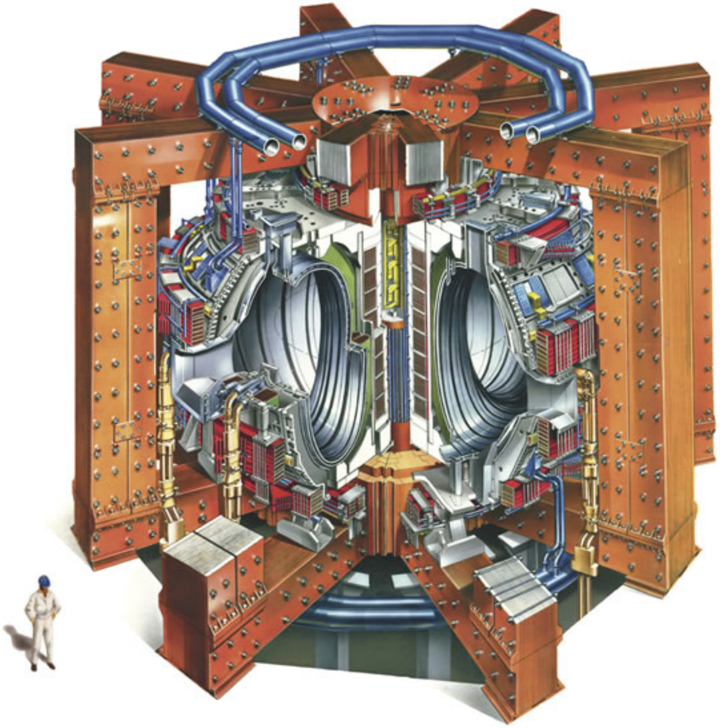
$$D^{+} \quad + \quad T^{+} \rightarrow {}^{4}He^{++} \text{ (3.5 MeV)} \quad + \quad n \text{ (14.1 MeV)}$$

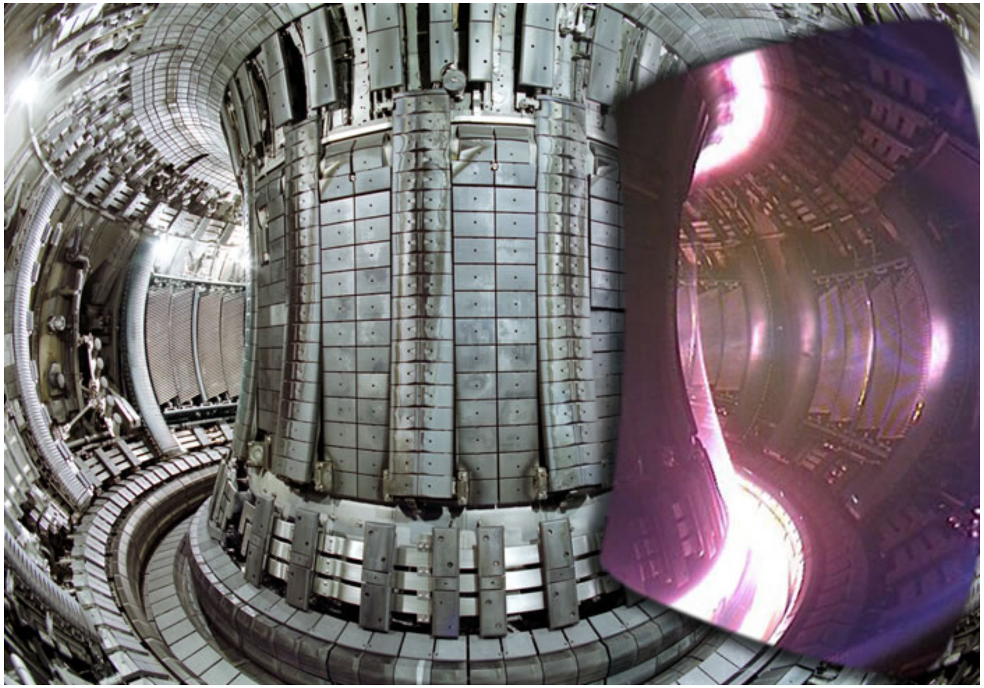
- Many technical challenges, e.g.:
  - Temperatures of 10^8 K
  - **Losses of heat and particles (fuel)**
  - Neutron fluxes cause material damage

ITER



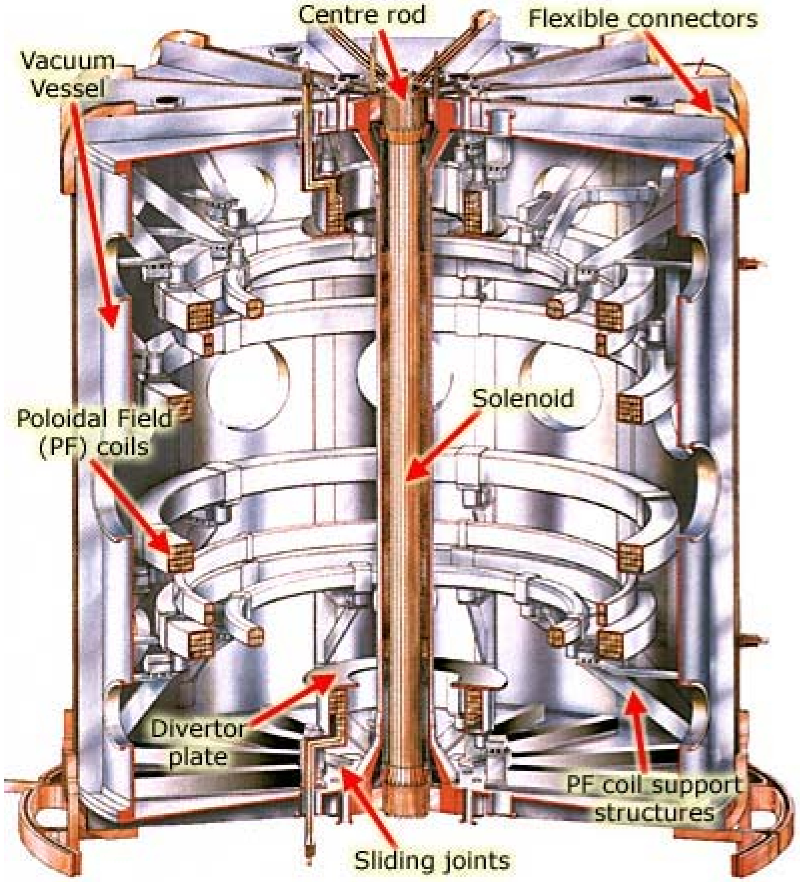
EFDA **J**oint **E**uropean **T**orus (JET)

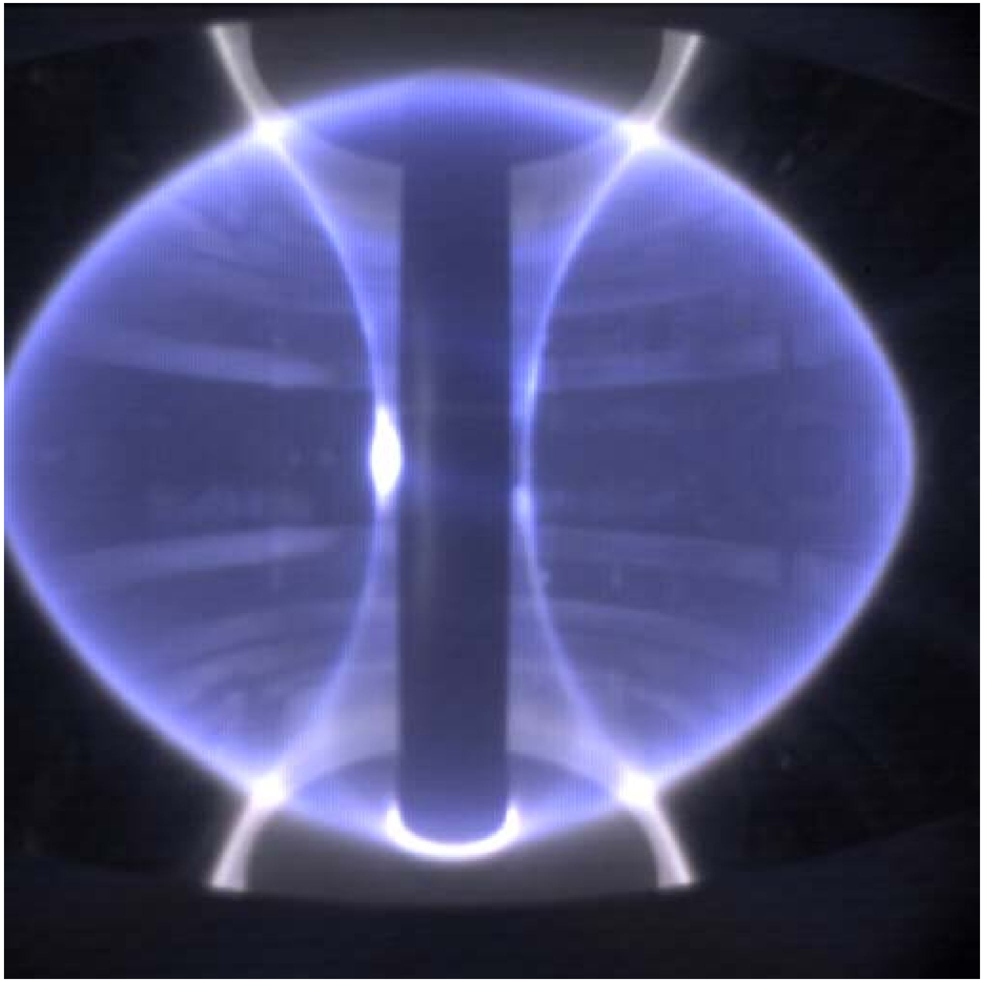




Images: EFDA

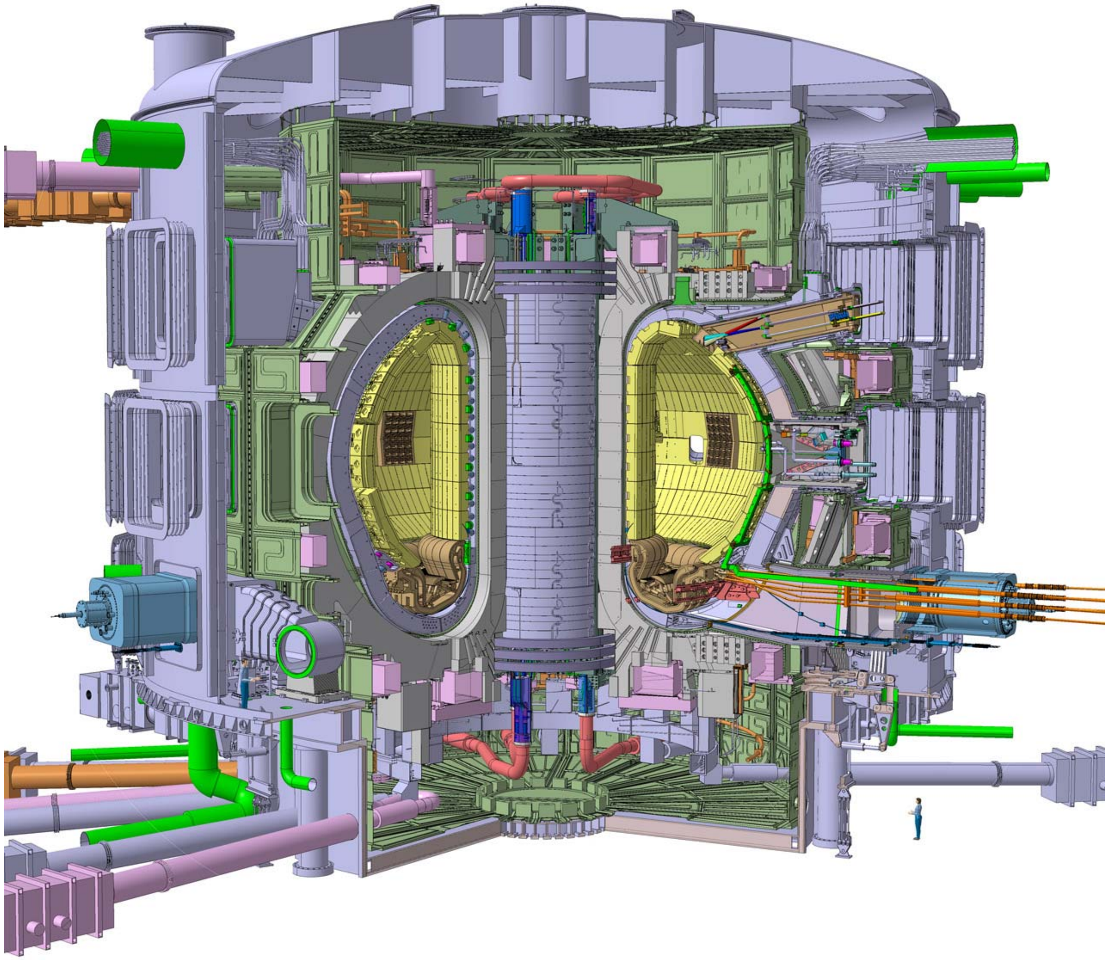
UKAEA: Mega Amp Spherical Tokamak





ITER

- 30 m tall (Plasma R=5.6m)
- 23,000 tons
- First plasma in 2018
- Goal:
  - Q > 10
  - ~ 500 MW power



Plasma Turbulence

- Understanding particle and heat loss will
  - help scientists to optimise efficiency of confinement
  - and therefore fusion performance

- Turbulence is believed to be responsible for these losses

- Simulating a full Tokamak is quite a challenge
  - Eddy turnover time: ≈1μs
  - Confinement time: ≈1s
  - Electron Larmor radius: 0.1mm
  - Tokamak plasma size: R = O(1m) (ITER has R≈6m)

- Range of codes to investigate different plasma phenomena
  - E.g.: Magneto hydrodynamic, gyrokinetic, particle …

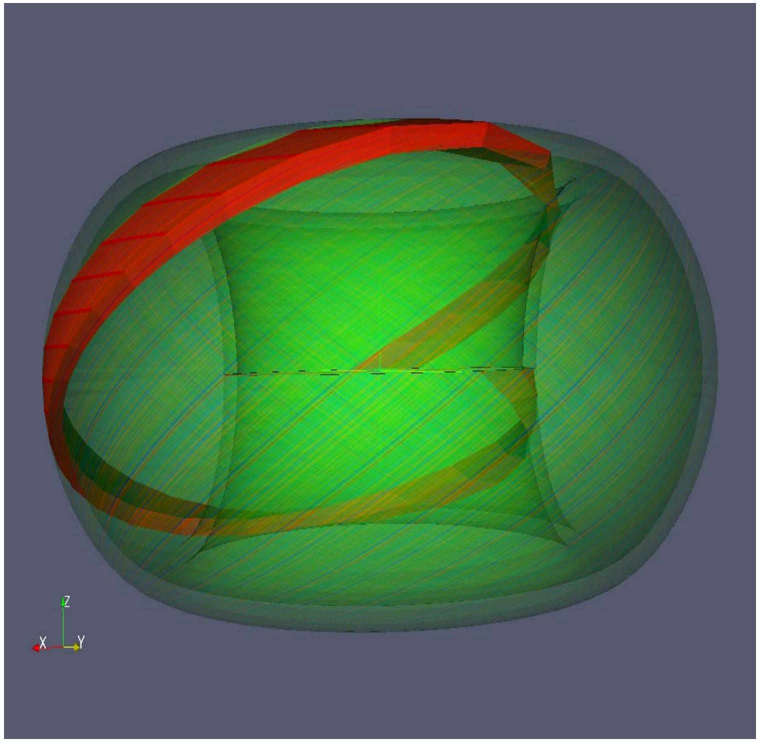
GS2: gyrokinetic plasma simulations

- GS2 is a mature open source code
- Developed by Dorland, Kotschenreuther and Liu
- International contributors including Oxford, York and UKAEA
- http://gs2.sourceforge.net/
- **Rem:** Astro version of GS2: AstroGK

- Averaging fast Larmor orbit motion gives 5D phase space (3 spatial and 2 velocity coordinates)
- Leads to Gyrokinetic equations: Distribution function in 5D
- Solver: linear step implicit, nonlinear step explicit
- Suitable: short perpendicular wavelength plasma turbulence

Fluxtube domain

- Particle movement most rapid along the field lines ⇒ flux tube domain attractive



- Restrict simulation domain to this flux tube
- Domain decomposition
  - 5D arrays offer a lot of scope
- F90 and MPI

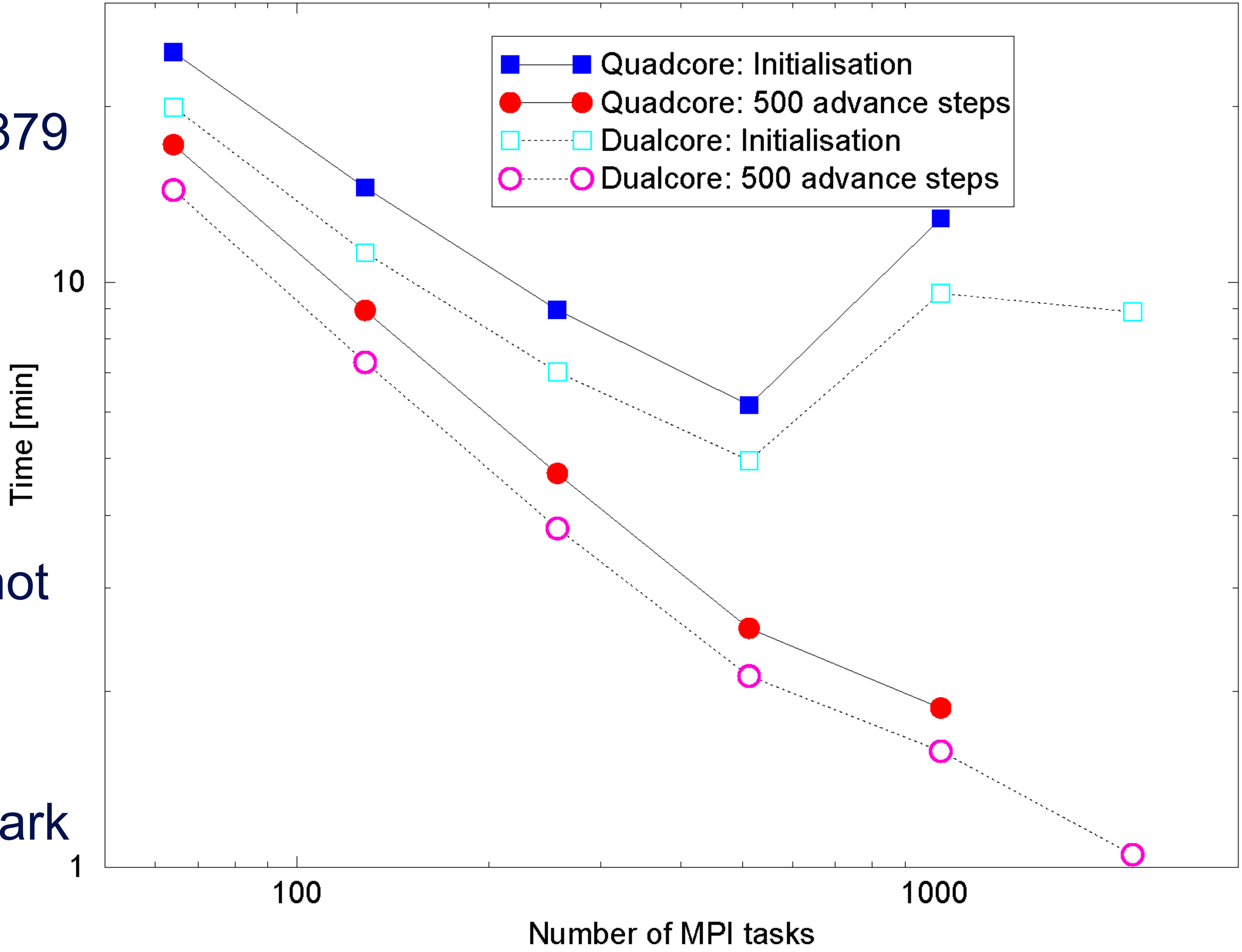
Input file (smallish problem)

- 2 species of charged particles
- 96 × 96 Fourier modes in x, y direction (anti-aliased)
- 31 points along the field lines
- Velocity space:
  - 8 energy levels
  - 32 pitch angles
  - 2 directions perpendicular to **B**
  - Total 512 velocity space points

Scaling of GS2 on HECToR

SVN:

Revision: 879



Remark:

This is not the best scaling benchmark

Profiling on 256 cores (July 2009)

Rem: Differences of 1500 steps and 200 steps, Craypat tool. Sampled 100 times/sec, though we get only about 60 ticks/sec -> follow up with CRAY

	Samples	Percentage
Total	**47966**	**100 %**
User	27421	57.17%
ETC	18911	39.43%
MPI	1634	3.4%

User time on 256 cores

User	**27421**	**57.17%**
dist_fn_get_source_term_set_source	4536	9.46%
gs2_transforms_transform2_5d_accel_	6276	13.08%
dist_fn_get_source_term_	2133	4.45%
dist_fn_invert_rhs_1_	1969	4.10%
nonlinear_terms_add_nl_	4949	10.32%
dist_fn_getan_	727	1.52%
le_grids_integrate_species_	576	1.20%
dist_fn_getfieldeq1_	312	0.65%

Rem: 44.8% accounted for – leaves about 12% in other stuff

ETC time on 256 cores

ETC	**18911**	**39.43%**
fftwi_no_twiddle_12	3473	7.24%
PtlEQPeek	836	1.74%
rfftw_c2hc	2516	5.25%
fftw_hc2real_12	2208	4.60%
fast_nal_poll	478	1.00%
PtlEQGet	392	0.82%
fftwi_twiddle_8	1438	3.00%
fftw_hc2hc_backward_8	1413	2.95%
PtlEQGet_internal	290	0.60%
MPIDI_CRAY_smpdev_progress	277	0.58%
ftn_strcmp	400	0.83%
fftw_no_twiddle_12	870	1.81%

30.4% accounted

24.9% FFTW

Ptl routines: MPI related

MPI time on 256 cores

MPI	**1634**	**3.41%**
mpi_allreduce_	1241	2.59%
mpi_barrier_	247	0.51%

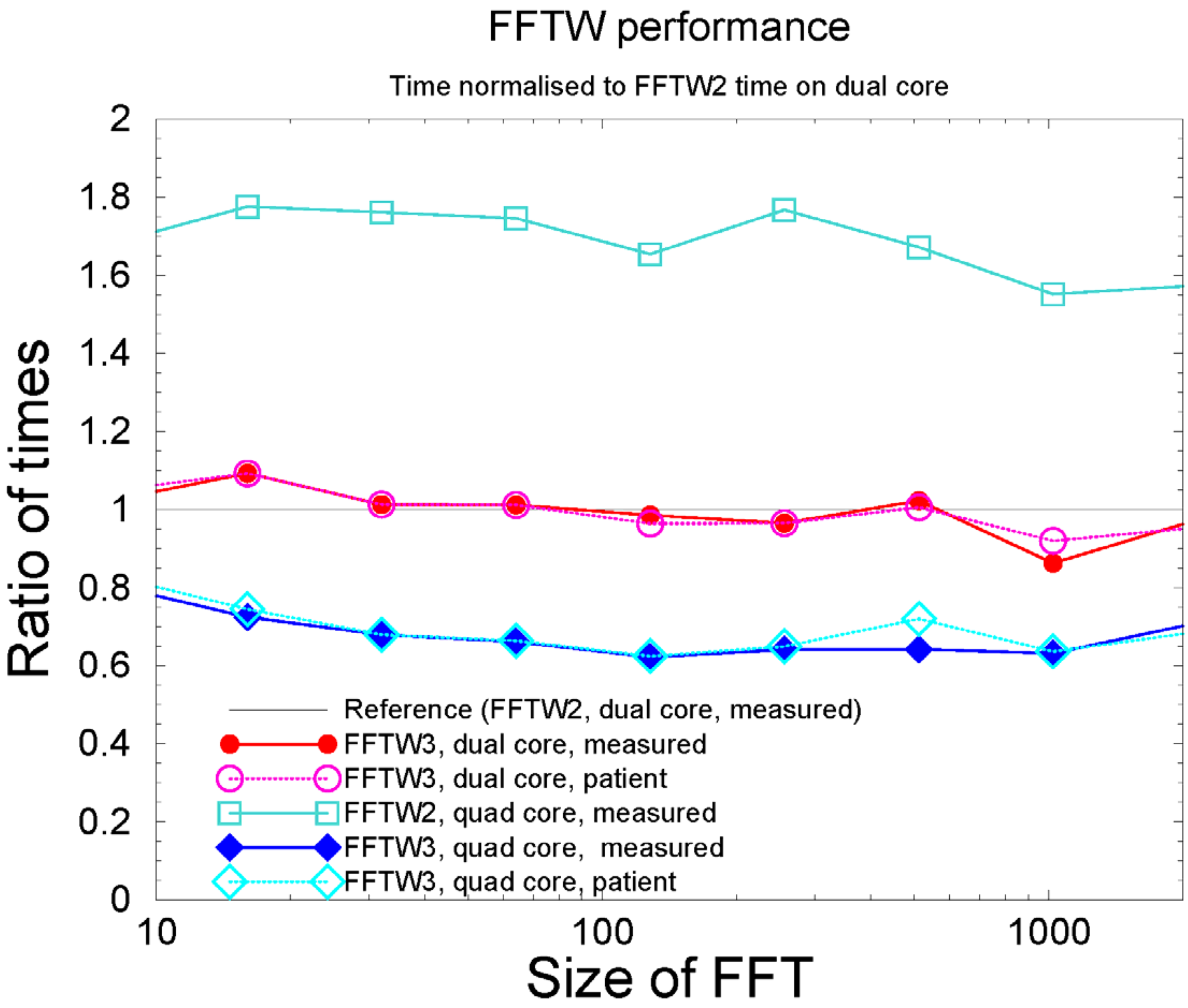
Quick reminder on FFT libraries

- FFT library calculates things like:

$$\tilde{f}(k) = \sum_{x=1}^{N} f(x) \exp\left(-2\pi \mathrm{i} \frac{kx}{N}\right)$$

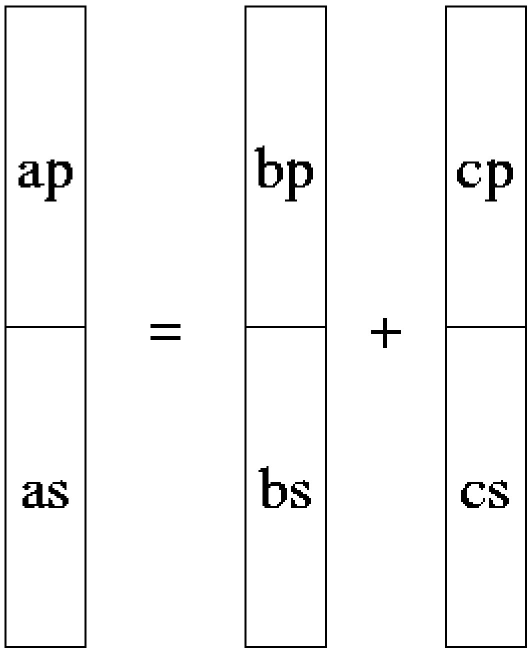
- Typically a two stage process on
  - Plan creation (initialisation - involves e.g. trigonometric functions)
    - Typically: Do once for each *N* and store
    - Use many times
  - Calculate the FFT using the stored plan
    - Done many times during a scientific program

FFTW performance (artificial benchmark)



SSE instructions on OPTERON

- Many processors offer 128 bit floating point registers & units
- Capable of retiring two (dp) or four (sp) operations per cycle



- These typically have strong alignment constraints (16 byte)
  - Hard to resolve for the compiler
  - Using a library might be an alternative (e.g. FFTW3)

Moving GS2 from FFTW2 to FFTW3

- Benefits:
  - Benchmarking suggests improved performance
  - FFTW2 is legacy code – move GS2 to the modern times

- Problems/Difficulties/Requirements
  - Keep FFTW2 as an option, at least now (pre-processor option)
  - Interface incompatible between FFTW2 and FFTW3
    - Pointers are now part of the plan (requirement for SSE instructions)
    - Input and output array needs to be know at plan creation
    - Requires re-engineering of the code
  - Code needs to run on a number of different architectures
  - 16-byte aligned allocates unavailable in Fortran (fftw_malloc C only)
  - Easy option: copy into work arrays or estimated plans
  - Better implementation: avoid the copies, use measured plans

Thanks

- Questions
- Comments