Frank Jenko

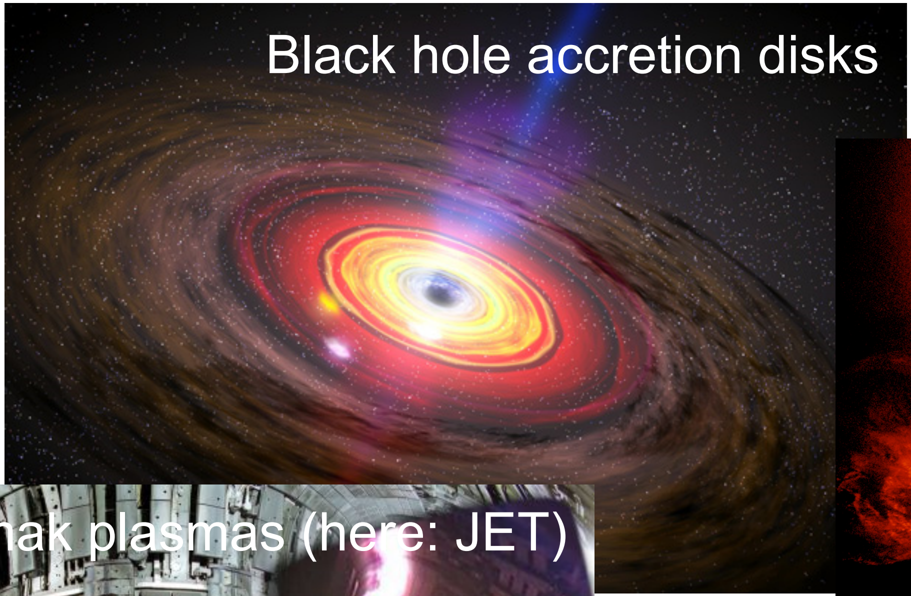
www.ipp.mpg.de/~fsj

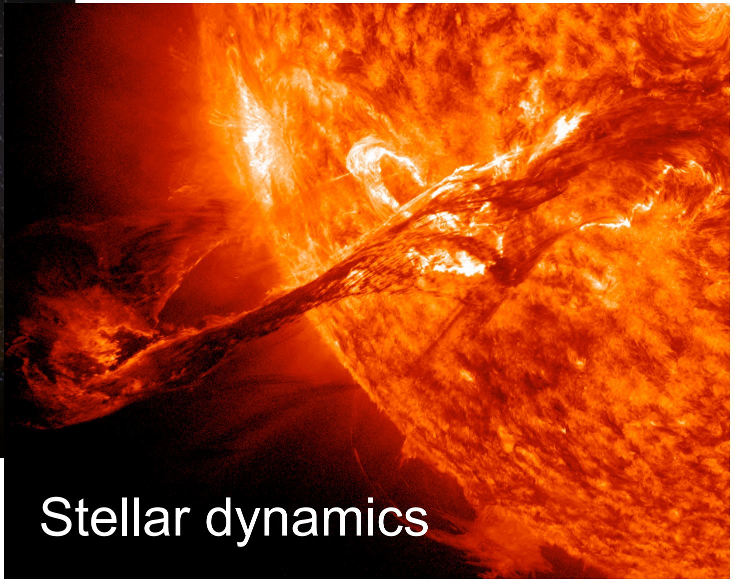
State-of-the-art turbulence simulations for fusion and astrophysical plasmas with GENE

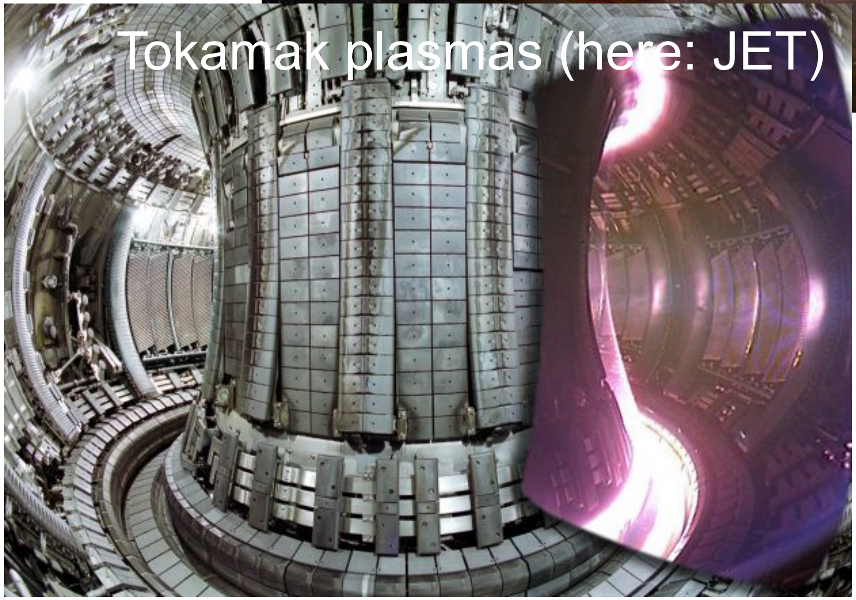
Max Planck Institute for Plasma Physics, Garching
University of Ulm

SuperMUC Review Workshop
Garching, 8-9 July 2014

Our plasma universe: More than 99% of the visible universe is in a plasma state



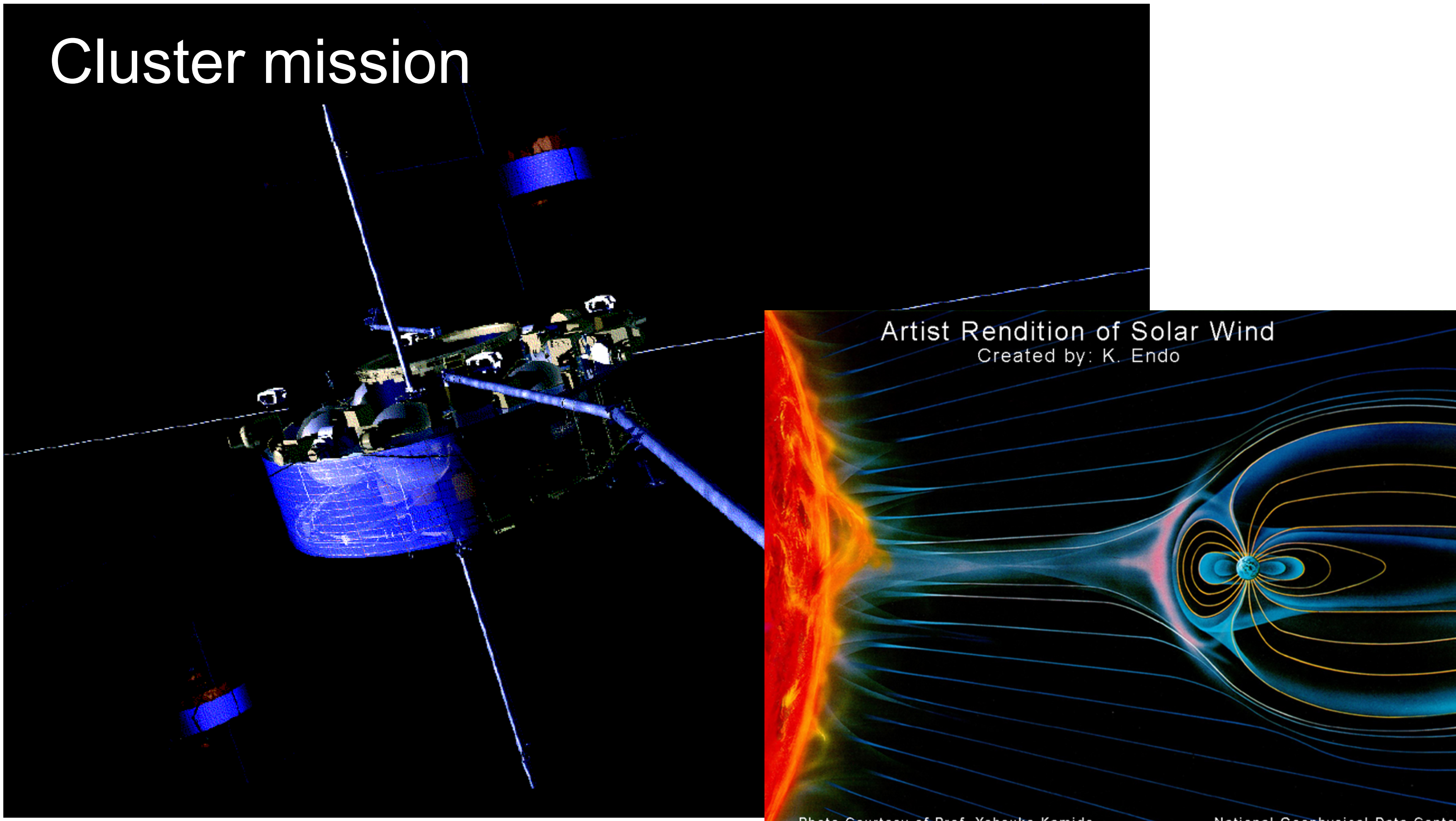




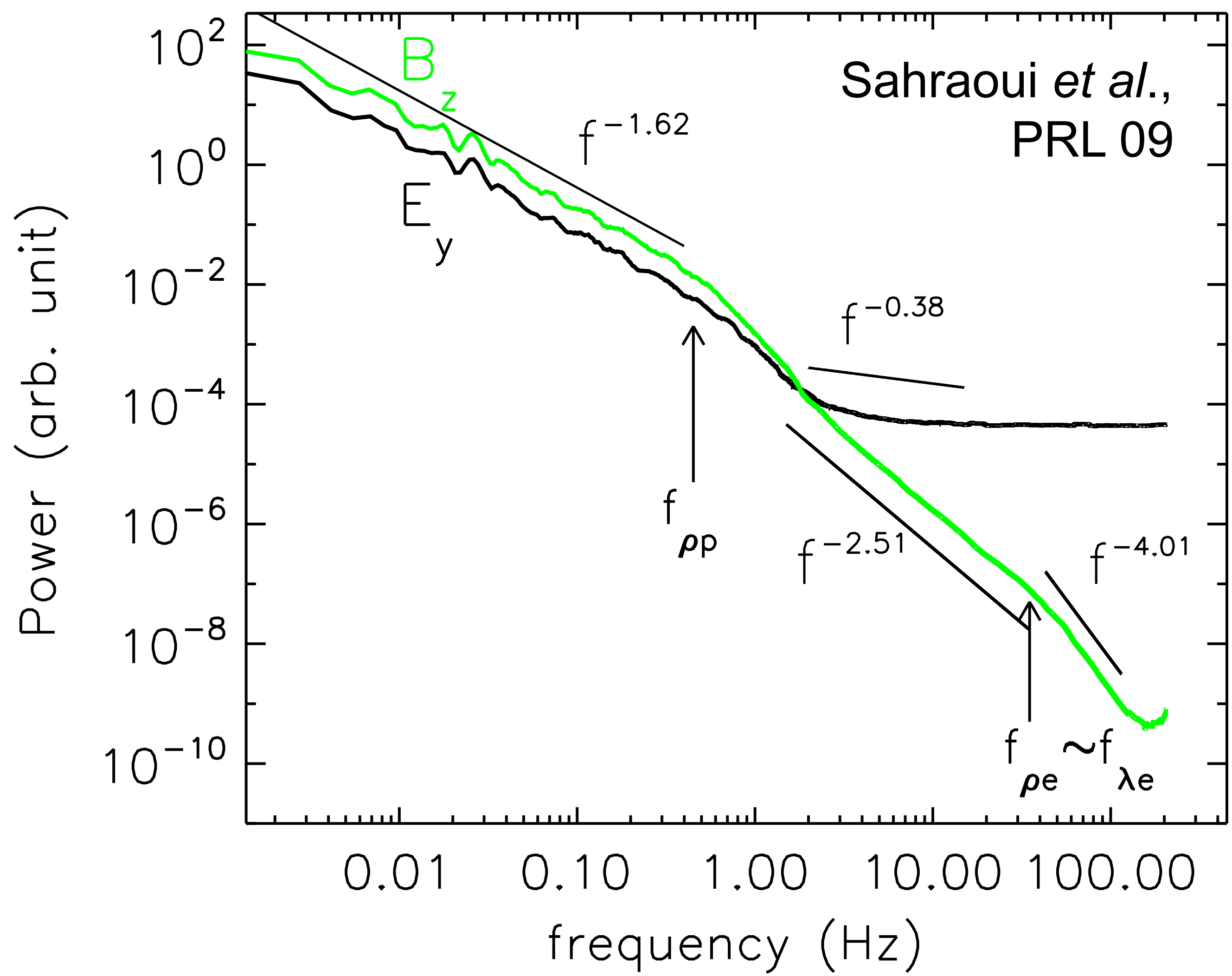
Turbulence is widely recognized as an **important open problem** in modern physics & astrophysics

EXCITING NEW OBSERVATIONS

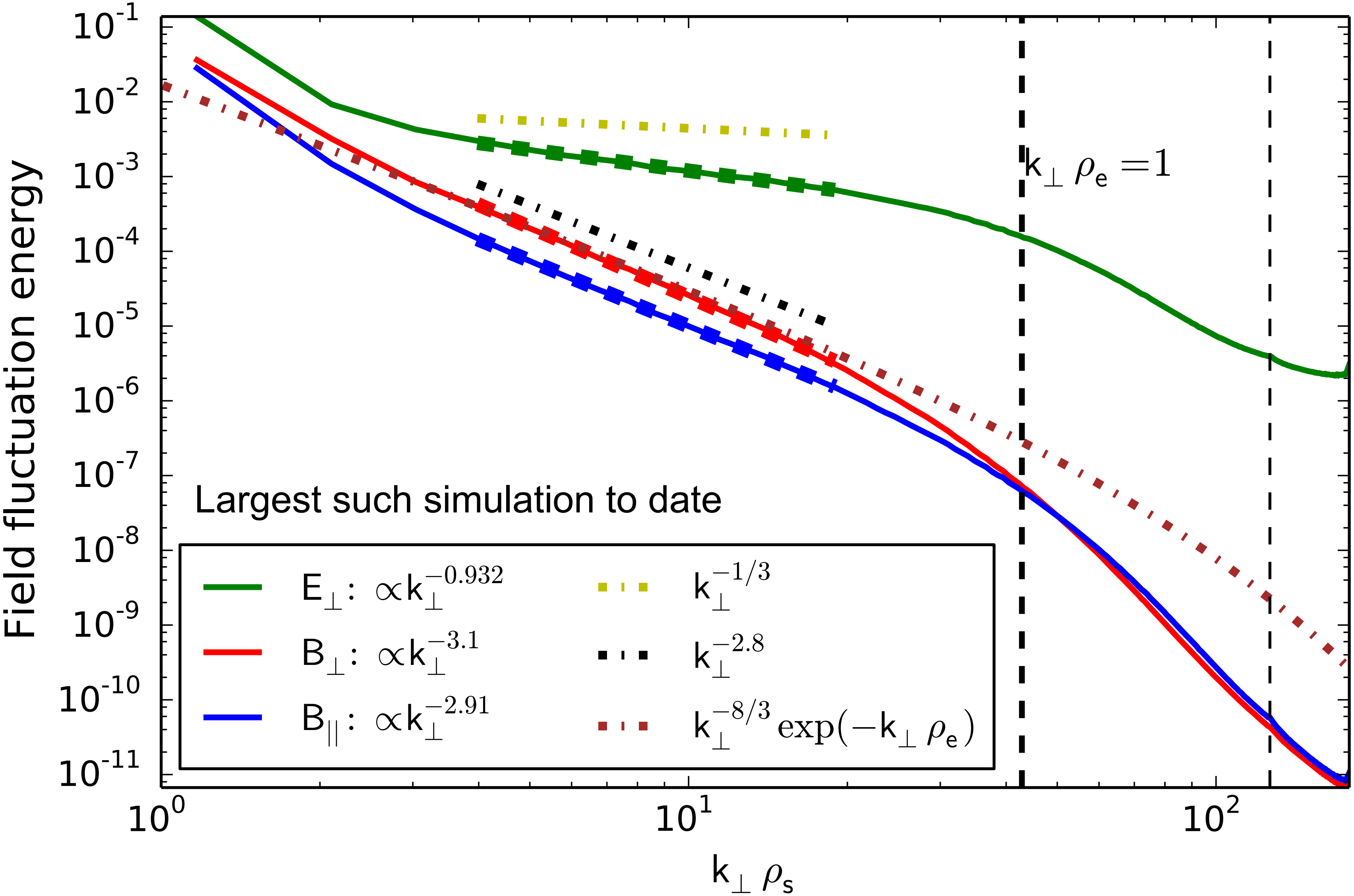
In situ measurements of plasma turbulence at various scales!



THE SOLAR WIND AS PLASMA TURBULENCE LABORATORY



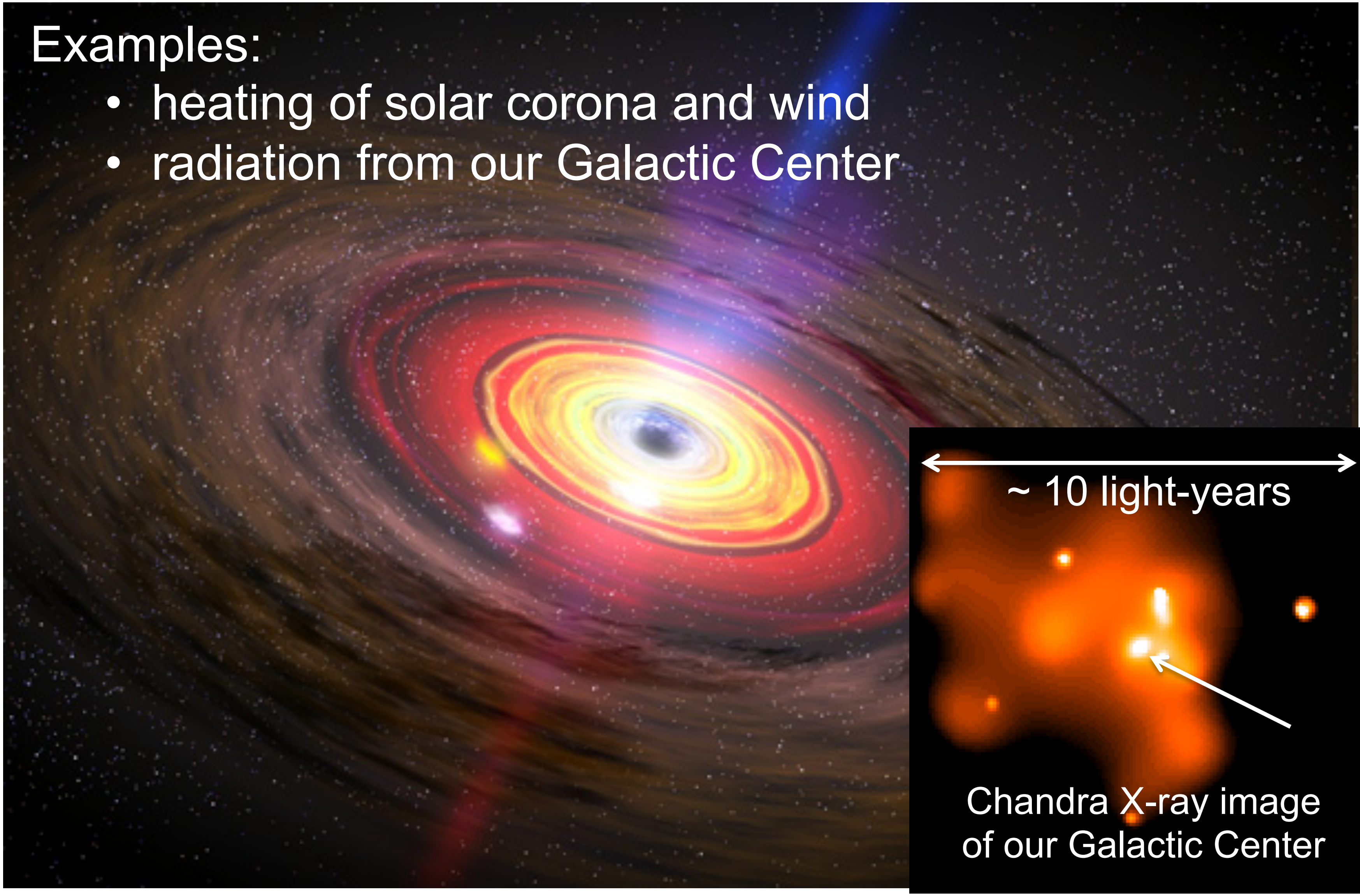
KINETIC SIMULATIONS (GENE CODE)



WHY DO WE CARE?

Examples:

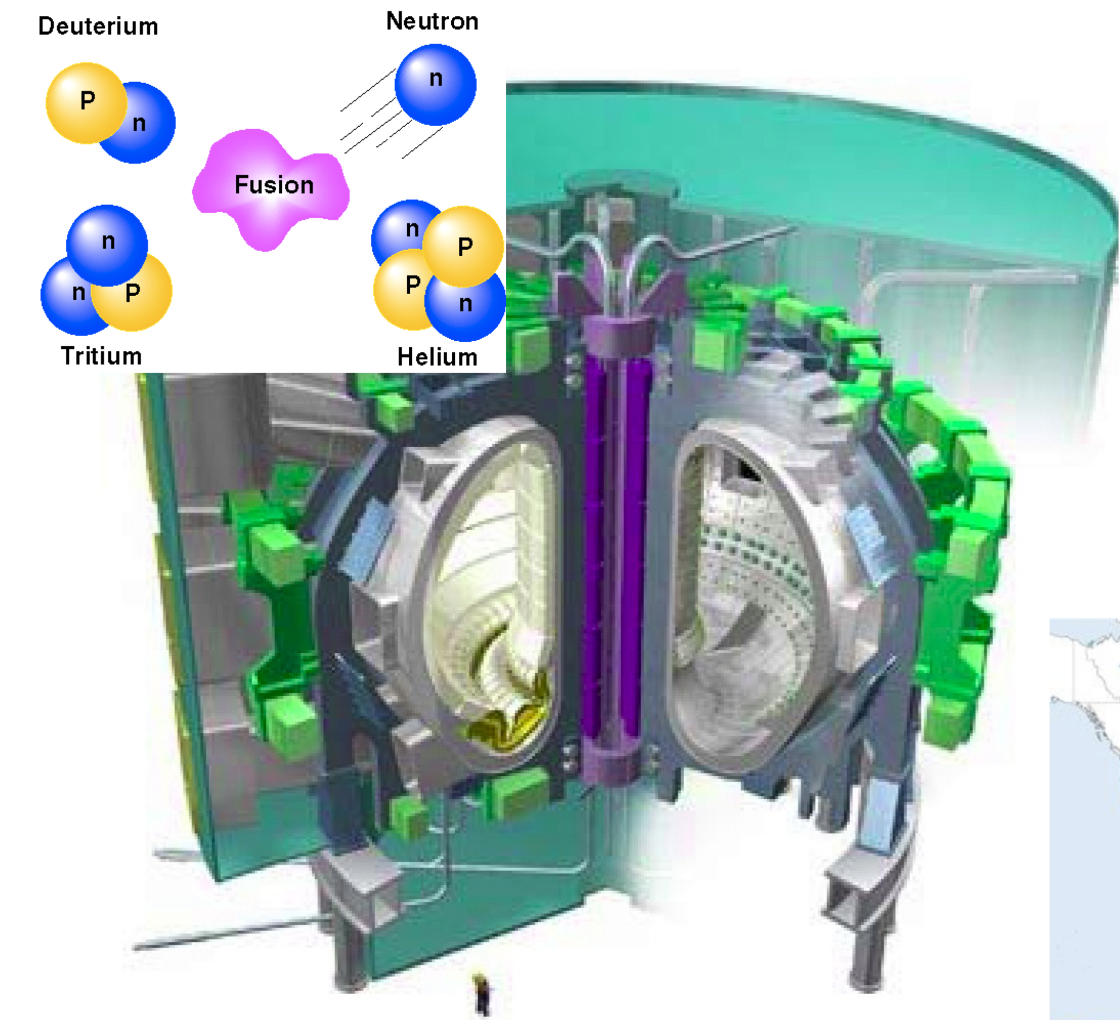
- heating of solar corona and wind
- radiation from our Galactic Center



Chandra X-ray image of our Galactic Center

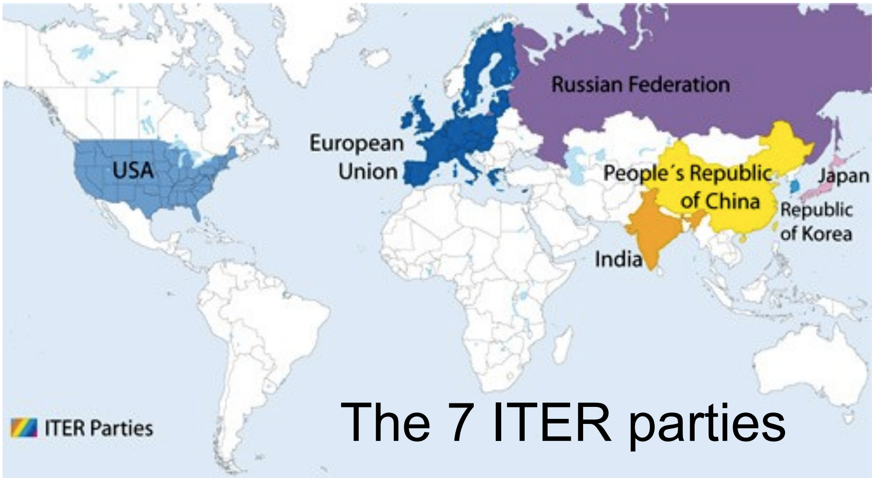
Plasmas in fusion research: ITER

Idea: New source of CO_2 free energy for centuries to come



Magnetic confinement in a large tokamak

Goal: 500 MW of fusion power



The 7 ITER parties

The resources for fusion energy are practically unlimited

Deuterium in a bath tub full of water and Lithium in a used laptop battery suffice for a family over 50 years

Global Gyrokinetic Simulation of Turbulence in

ASDEX Upgrade

GENE

gene.rzg.mpg.de

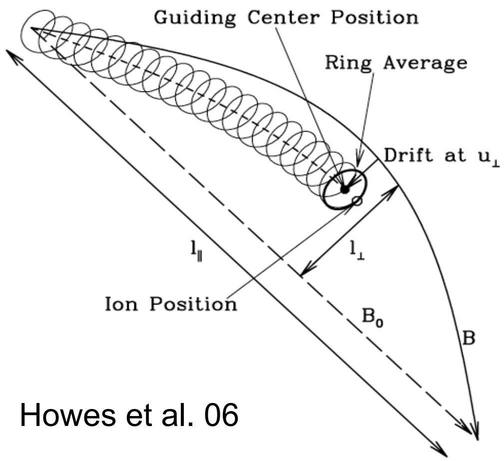
gene@ipp.mpg.de

Extreme computing with the GENE code

Ab initio microturbulence: NL gyrokinetics

Microturbulence in weakly collisional plasmas requires a kinetic description!

Vlasov-Maxwell equations $\left[\frac{\partial}{\partial t} + \mathbf{v}\cdot\frac{\partial}{\partial \mathbf{x}} + \frac{q}{m}\left(\mathbf{E} + \frac{\mathbf{v}}{c}\times\mathbf{B}\right)\cdot\frac{\partial}{\partial \mathbf{v}}\right] f(\mathbf{x},\mathbf{v},t) = 0$



Howes et al. 06

From kinetics (6D) to "gyrokinetics" (5D):
For strongly magnetized plasmas, remove fast gyromotion, consider **guiding center dynamics**

$$\boxed{\frac{\partial f}{\partial t} + \dot{\mathbf{X}}\cdot\frac{\partial f}{\partial \mathbf{X}} + \dot{v}_\parallel\frac{\partial f}{\partial v_\parallel} = 0} \qquad f = f(\mathbf{X}, v_\parallel, \mu; t)$$

$$\dot{\mathbf{X}} = v_\parallel \mathbf{b} + \frac{B}{B^*_\parallel}\left(\boxed{\frac{v_\parallel}{B}\bar{\mathbf{B}}_{1\perp}} + \boxed{\frac{c}{B^2}\bar{\mathbf{E}}_1\times\mathbf{B}} + \frac{\mu}{m\Omega}\mathbf{b}\times\nabla(B+\bar{B}_{1\parallel}) + \frac{v_\parallel^2}{\Omega}(\nabla\times\mathbf{b})_\perp\right)$$

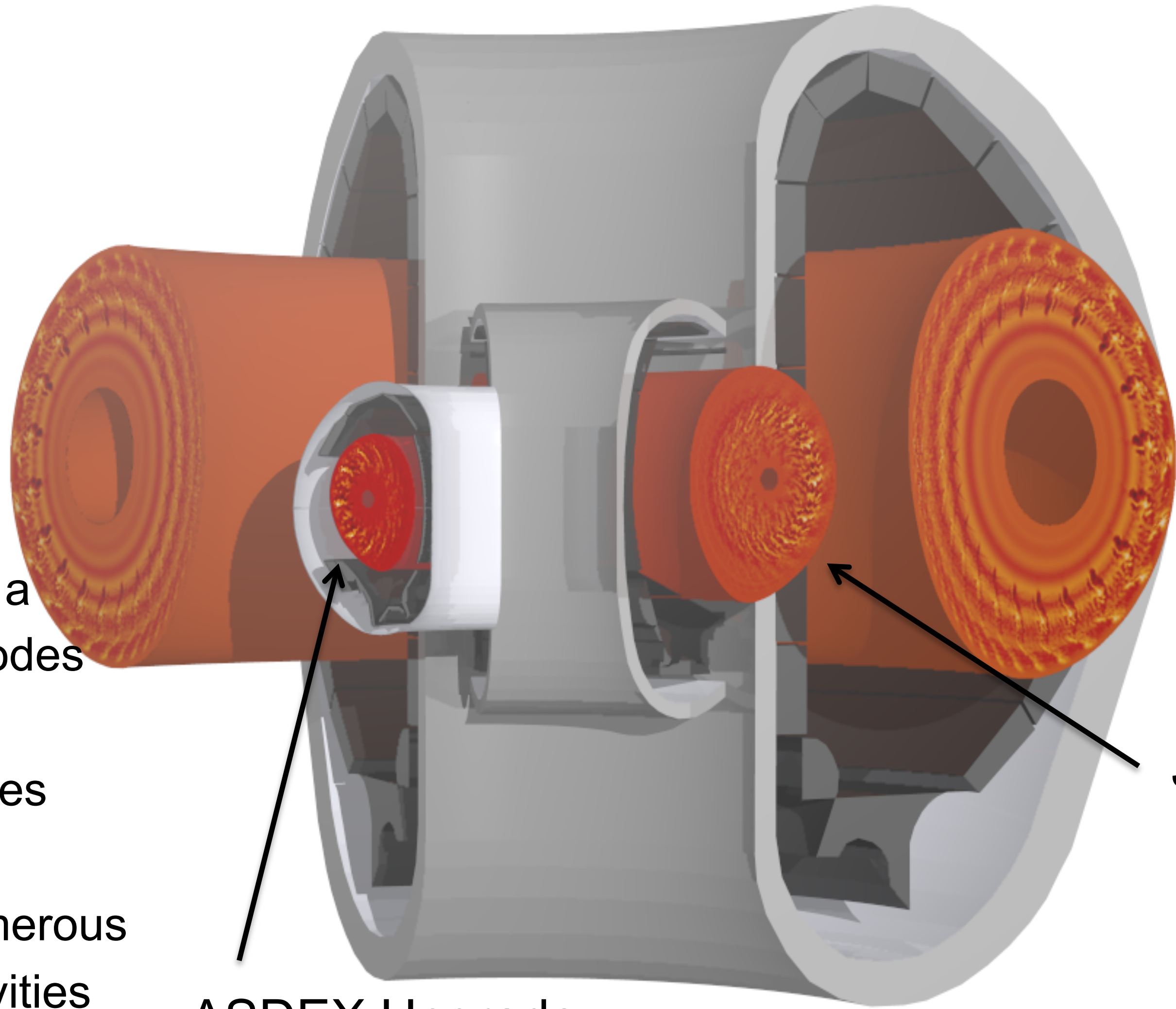
motion along fluctuating field lines — ExB drift

$$\dot{v}_\parallel = \frac{\dot{\mathbf{X}}}{m v_\parallel}\cdot\left(e\bar{\mathbf{E}}_1 - \mu\nabla(B+\bar{B}_{1\parallel})\right) \qquad \dot{\mu} = 0$$

Rigorously derived in the 1980s; **enormous reduction of spatio-temporal scales**

Global GK runs for actual tokamaks

Example: GENE code

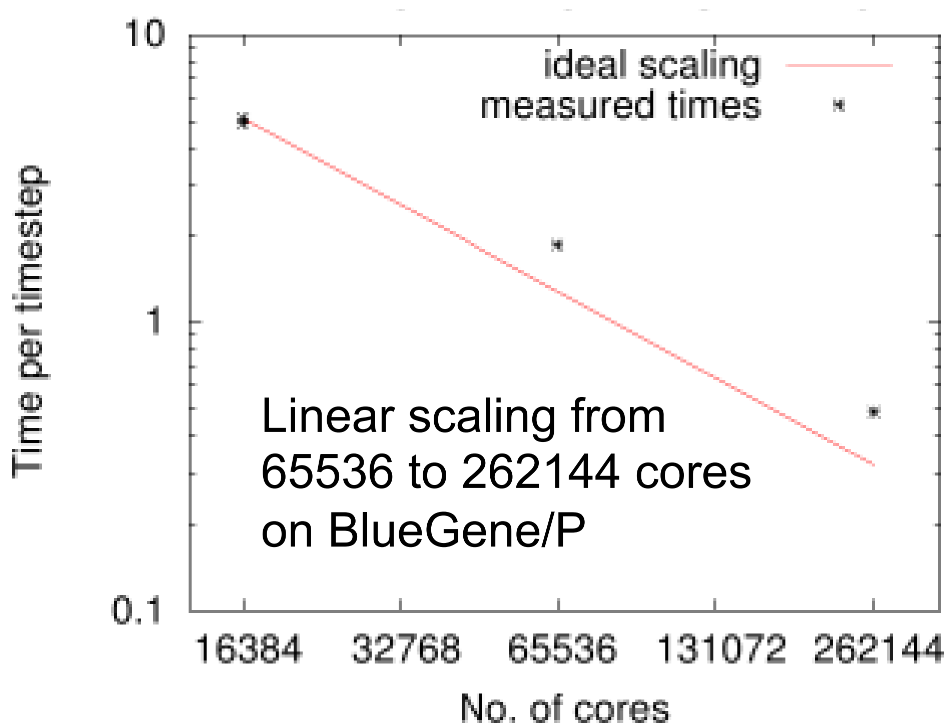


Today, there exists a variety of NL GK codes based on different numerical techniques

Verification via numerous benchmarking activities

Gyrokinetic code GENE (F. Jenko *et al.*, 1999 –)

- Code is *publicly available and widely used* (http://gene.rzg.mpg.de)
- Part of the *Unified European Application Benchmark Suite* (PRACE)
- First PRACE Call: Ranked #1 out of 65 projects from all areas of science



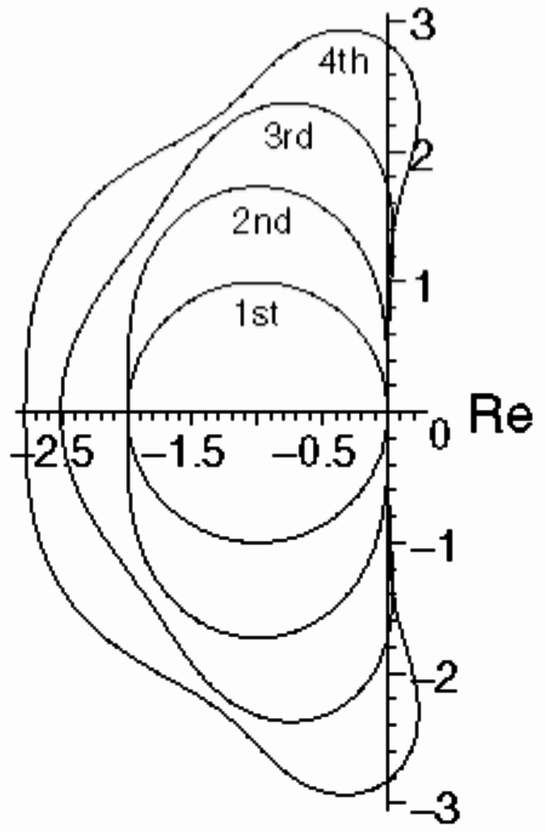
On SuperMUC:
Up to ~16 kcores

Gyrokinetic Electromagnetic Numerical Experiment

GENE parallelization

Parallelization/optimization strategy:

- high-dimensional domain decomposition
- either pure MPI or mixed MPI/OpenMP paradigm
- optimal subroutines and processor layout determined during initialization phase (à la FFTW)
- time step is chosen in an optimal way



Thanks to F. Merz (now at IBM)

Some computational challenges

- GENE runs are compute intensive; large individual runs may require up to tens of millions of core-hours
- Large runs use many billion grid points and require many TB of short-term storage
- Many different HPC platforms are used in parallel
- Recently, GENE has been ported to GPGPU & MIC systems

GENE on GPGPU systems

T. Dannert, M. Rampp

- we have ***initially*** achieved a speedup of 2 of the complete code with the previous hardware generation (Nehalem 4-core CPUs and Fermi M2090 GPUs)

- the speedup marginalizes on the current hardware generation (SandyBridge 8-core and Kepler K20x GPUs)

- we have achieved an in-depth understanding of the reasons:
  - the CPU version of GENE has improved during the GPU development
  - a roofline analysis shows: GENE is memory bound, and the data transfer limits the GPU speedup
  - possible improvements:
    - PCIe 3.0
    - move more computations to the GPU: transfer-to-compute ratio decreases

Applied math: Iterative eigenvalue solvers

Using SLEPc, an extension of PETSc
J.E. Roman, Valencia

Available methods:

- Power Iteration (also Inverse Power Iteration, Rayleigh Quotient Iteration)
- Subspace Iteration with Rayleigh-Ritz projection
- Arnoldi method
- Lanczos method
- Krylov-Schur (with and without Harmonic Extraction)
- Davidson methods (as of recently)

Recent result:

Using the preconditioned Jacobi-Davidson method is about twelve times faster as Krylov-Schur method with harmonic extraction and twice as fast as the preconditioned Krylov-Schur method.

Christoph Kowitz, TUM, Master’s Thesis

Sparse grid combination technique

With D. Pflüger (Stuttgart) & M. Griebel (Bonn) & H.-J. Bungartz (TUM) et al.

Cartesian grid

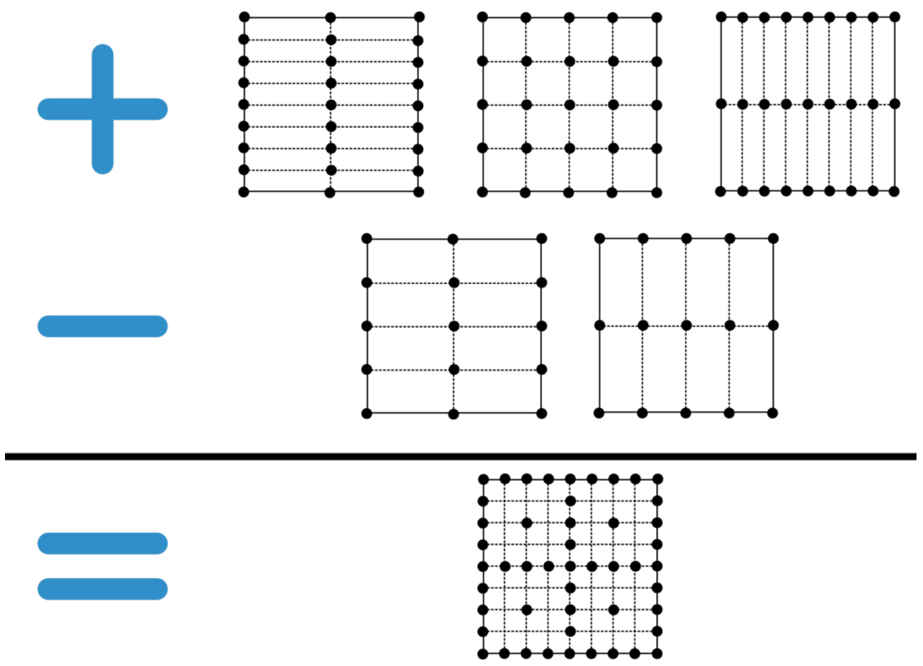
- Regular data structure
- Huge number of grid points for high-dimensional problems
  "curse of dimensionality"

Resolution: 33 grid points per dimension	**2D**	**5D**
Cartesian grid	1,089	39,135,393
Combination tech.	641	206,358

Combination technique

- Good approximation of the Cartesian grid solution
- smaller number of grid points
- existing code (GENE) can be used more or less as is

- **applicable to other high-dimensional grid-based problems**



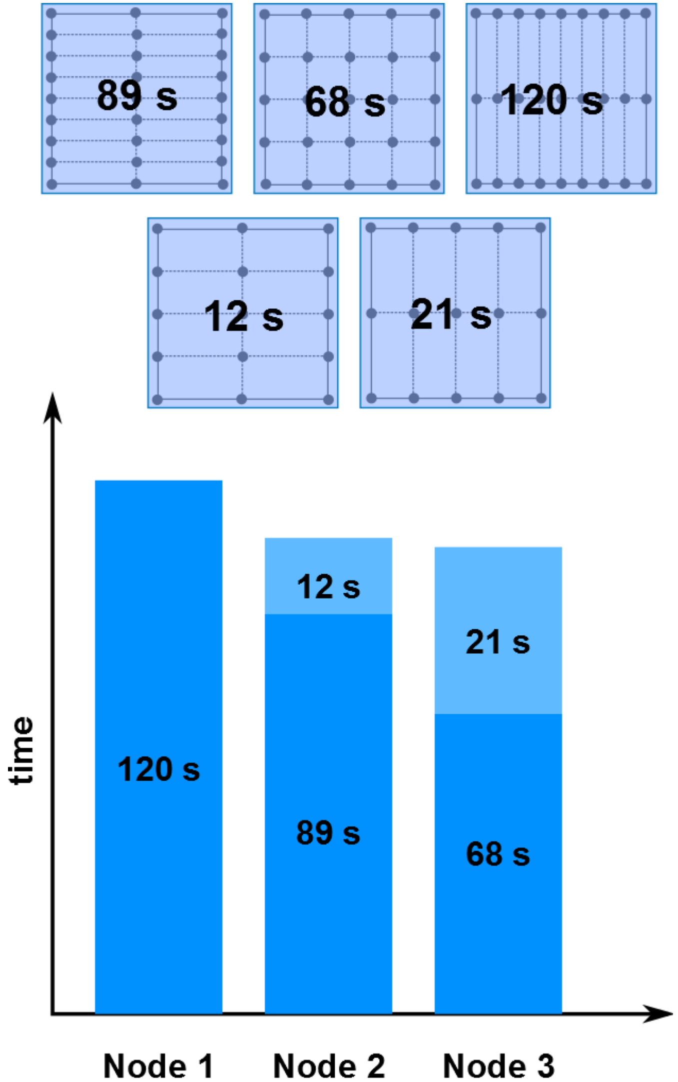
A new level of parallelism

Dual parallelism

- Independent grid setups from the combination technique + massively parallel GENE runs
- Run times of the instances tend to vary strongly

Optimize the load balance

- A simple **load-model** estimates the runtime required for each grid
- A **scheduler** creates an optimal load balancing to minimize idle cores



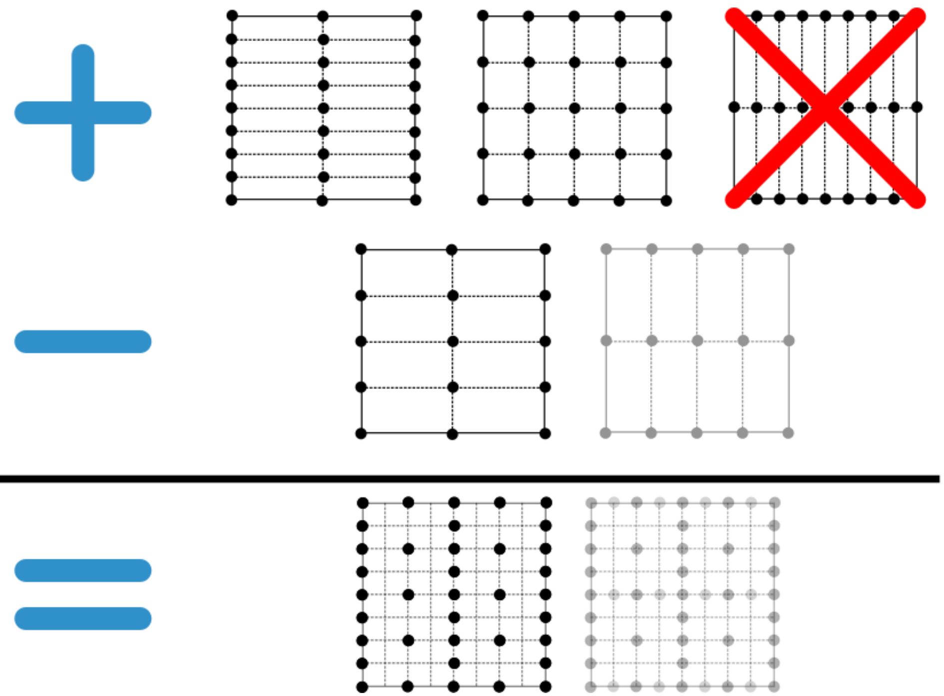
Spin-off: Algorithmic fault tolerance

Hardware failures ($10^{5\text{-}7}$ cores)

- The whole simulation has to be restarted from the last checkpoint file
- In the combination technique, only a single GENE instance would crash

Two ways to handle the failure

- The combination technique recovers an approximation
- Only a single GENE instance is rerun – which is much smaller than the full problem



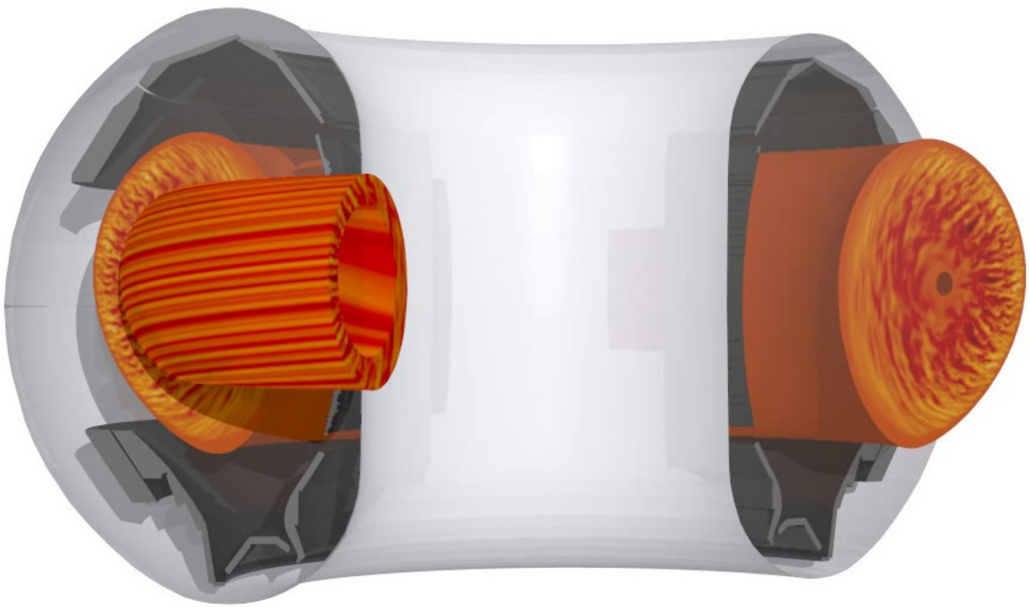
Such techniques may be very useful on future exascale architectures

SPPEXA

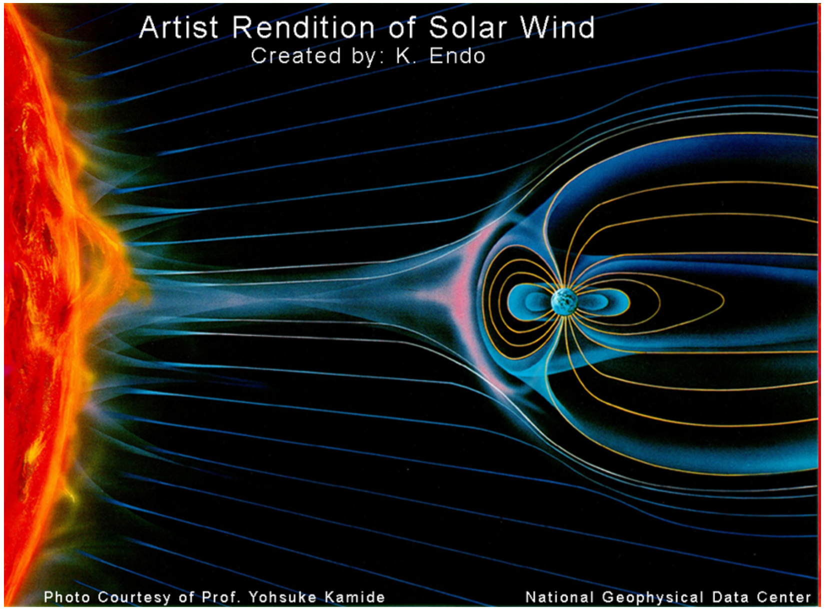
The future

Two key challenges

Virtual fusion devices



Space weather prediction



- Plasma turbulence: Where fascinating physics, extreme computing, and global challenges meet
- More information: gene.rzg.mpg.de