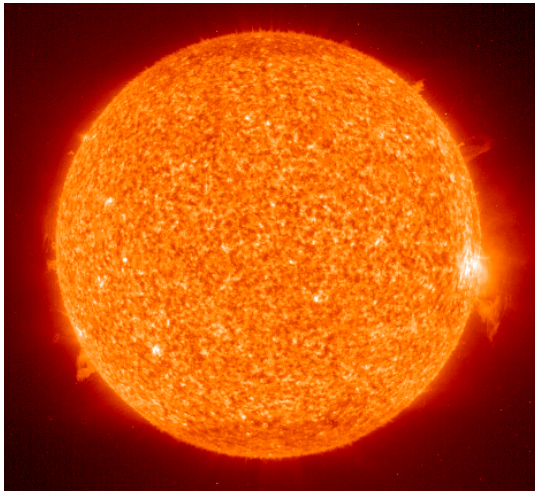
Frank Jenko

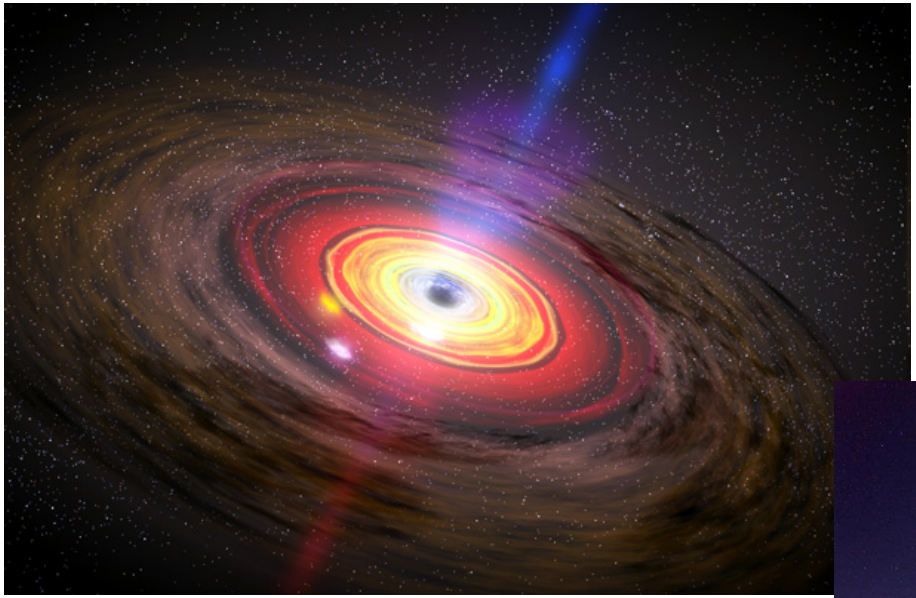
Turbulence in fusion and astrophysical plasmas: A driver for exascale computing

Professor, University of California, Los Angeles
Director, Max Planck Institute for Plasma Physics

International HPC Summer School
Boulder, 25-30 June 2017

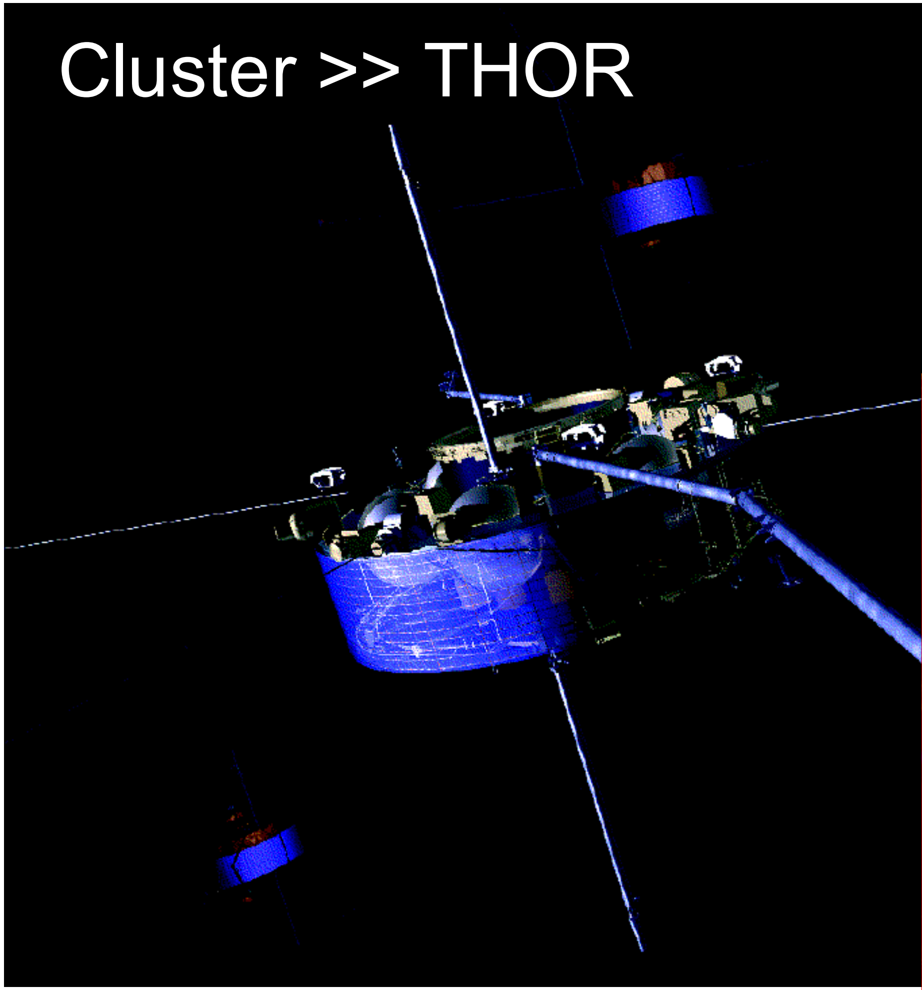
More than 99% of the visible universe are turbulent plasmas

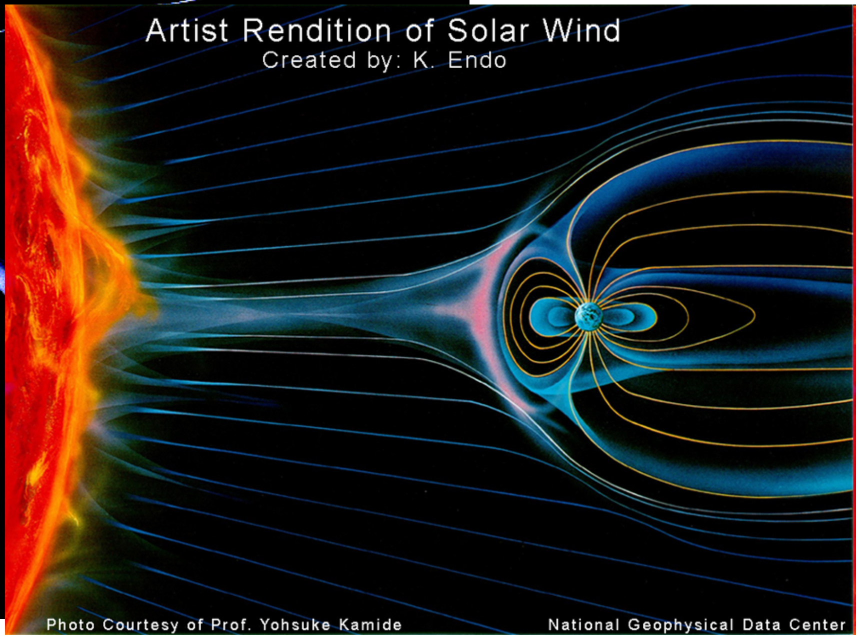




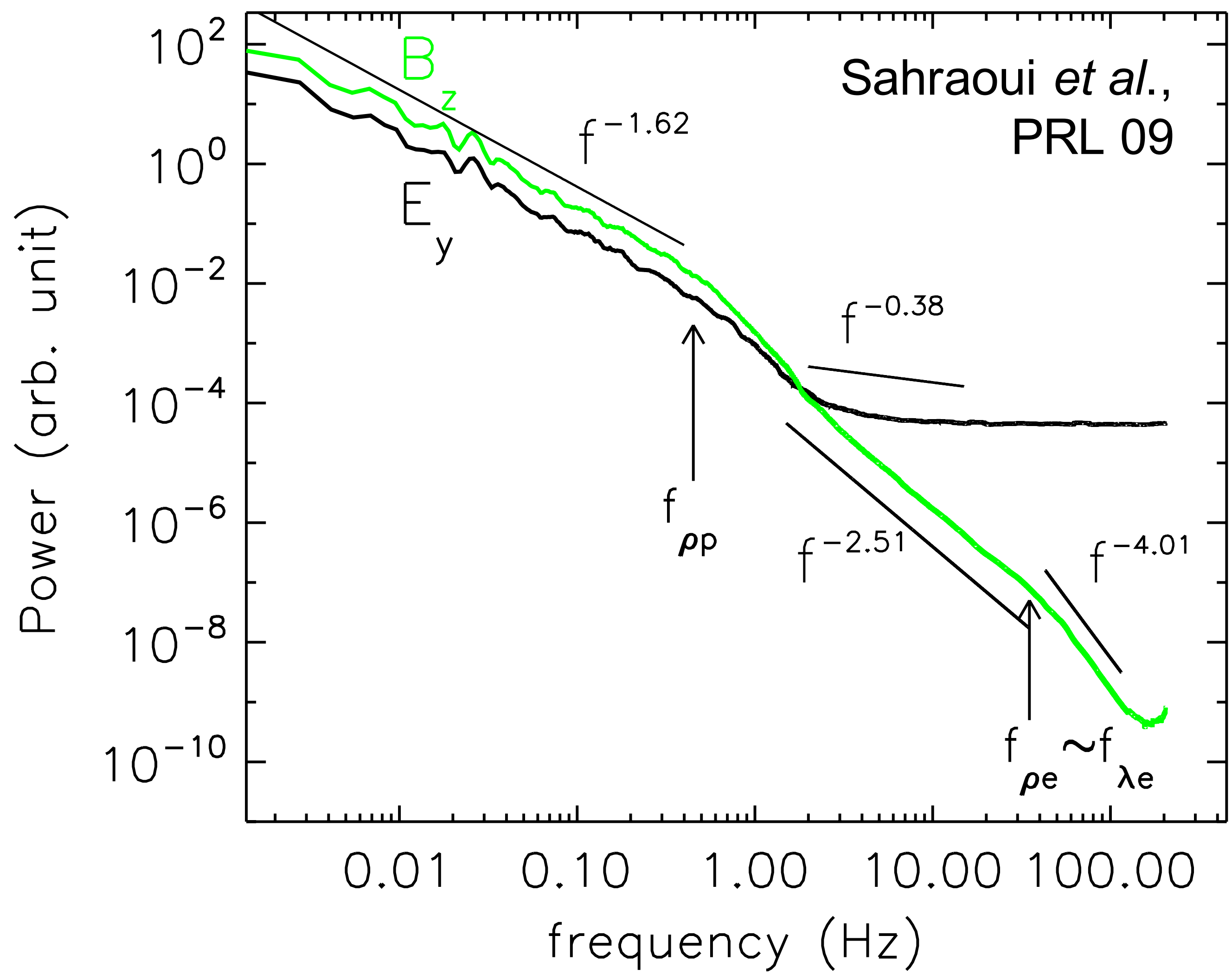
EXCITING NEW OBSERVATIONS

In situ measurements of plasma turbulence at various scales!

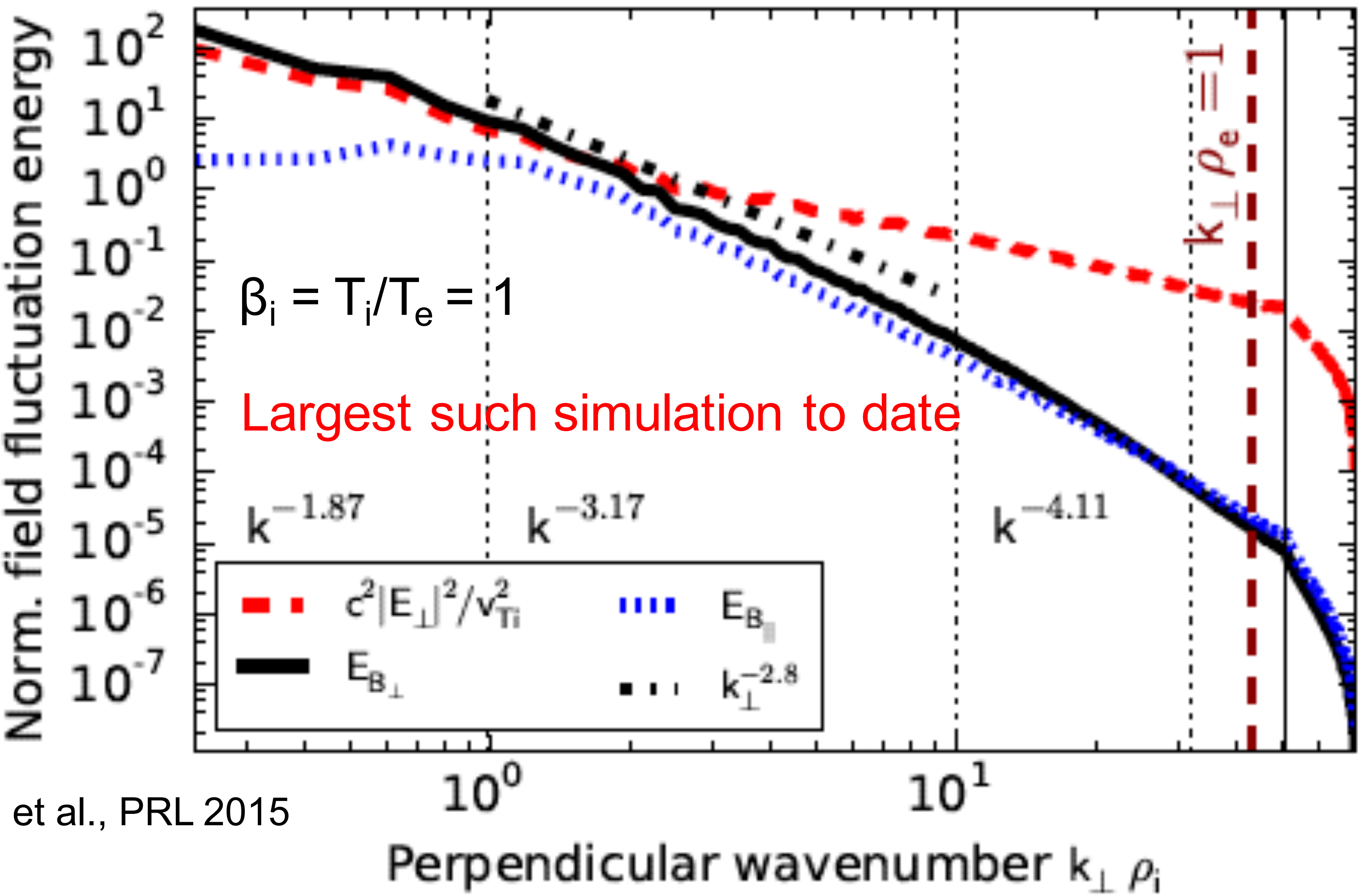




THE SOLAR WIND AS PLASMA TURBULENCE LABORATORY



KINETIC SIMULATIONS (GENE CODE)

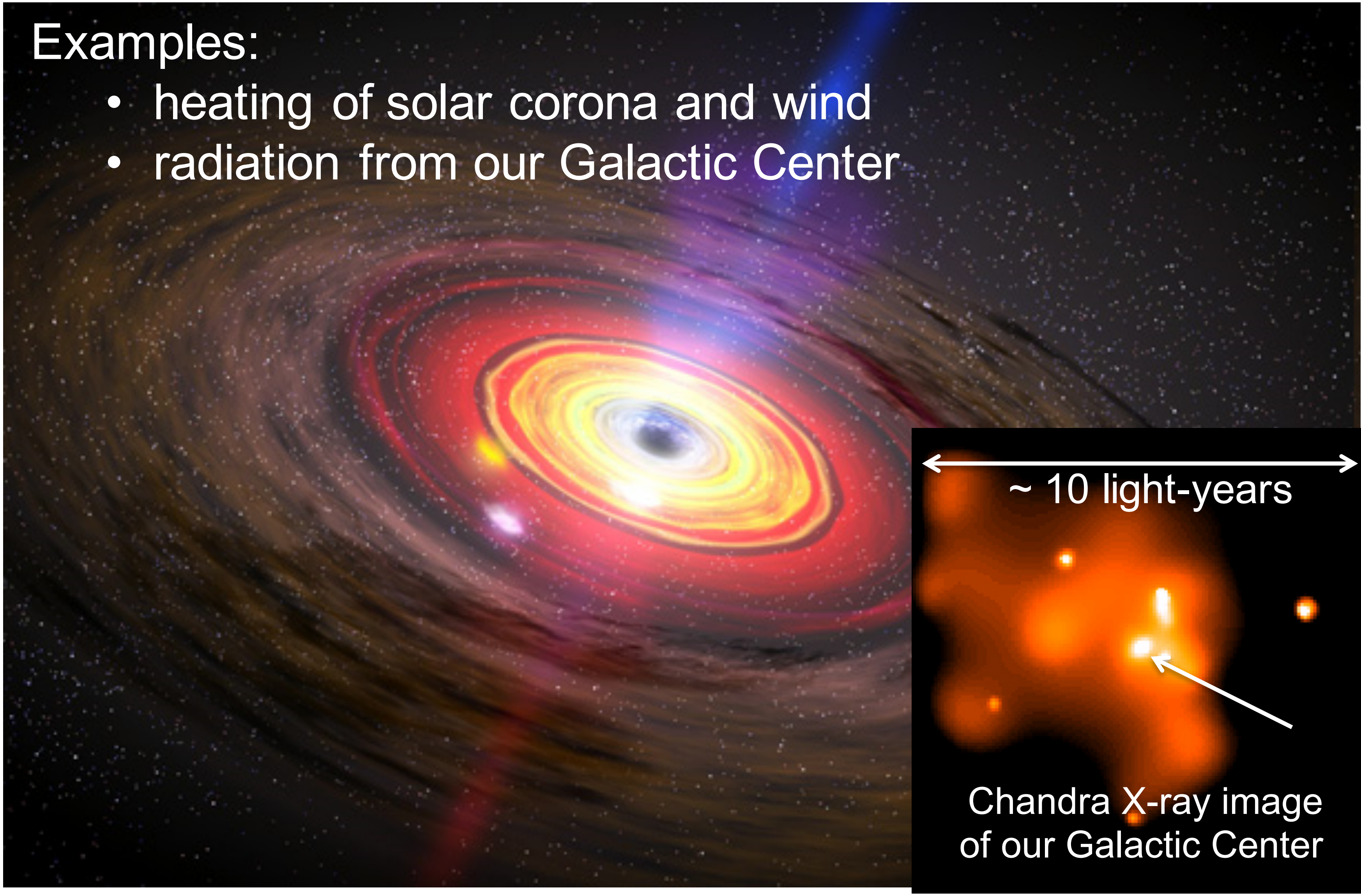


Told et al., PRL 2015

WHY DO WE CARE?

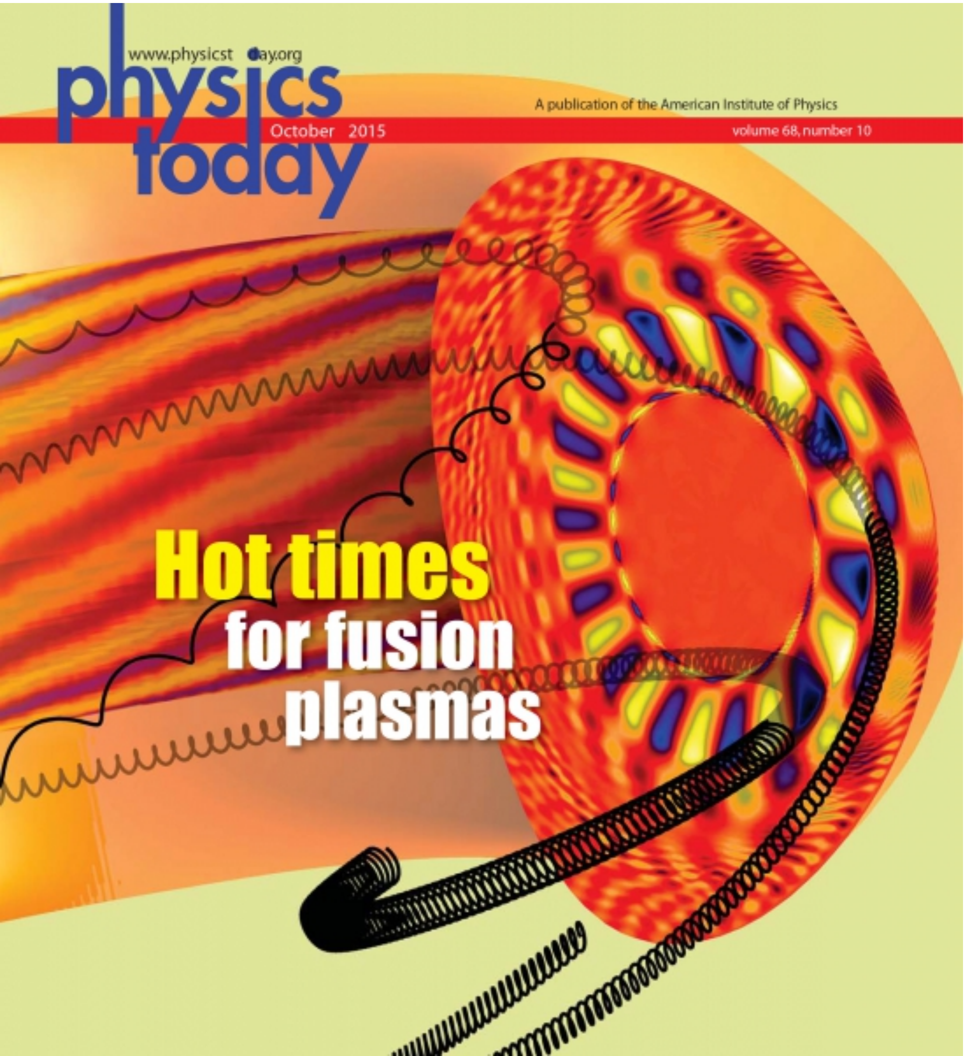
Examples:

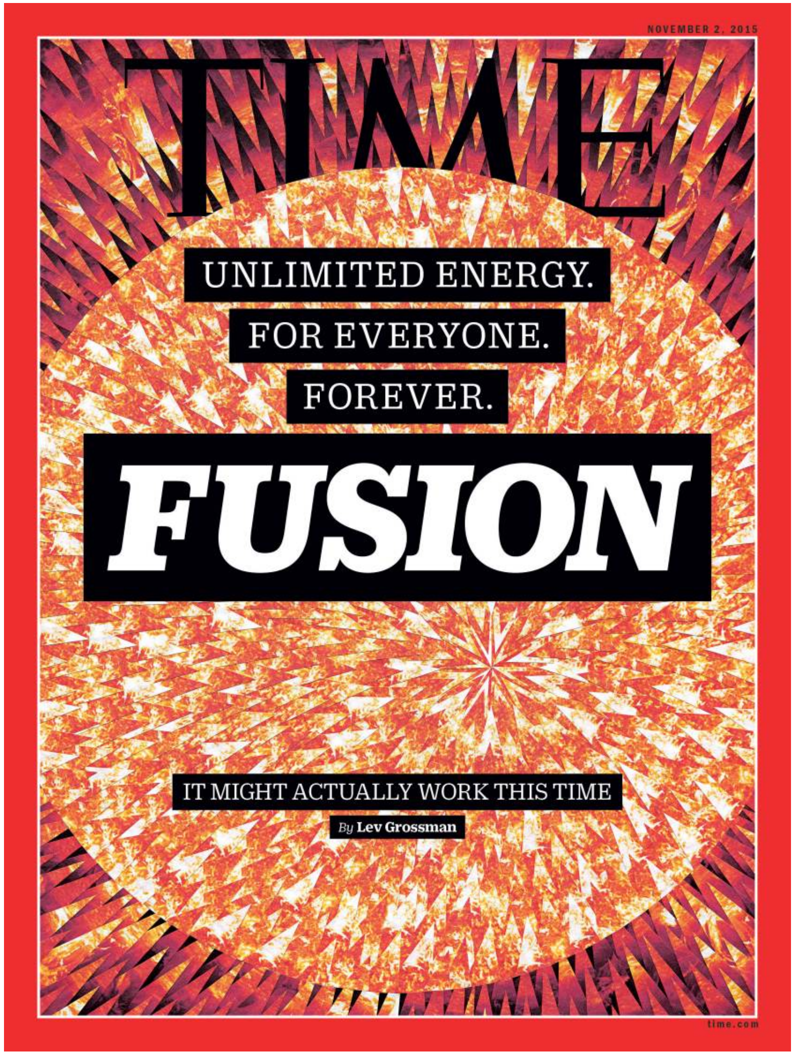
- heating of solar corona and wind
- radiation from our Galactic Center



Chandra X-ray image of our Galactic Center

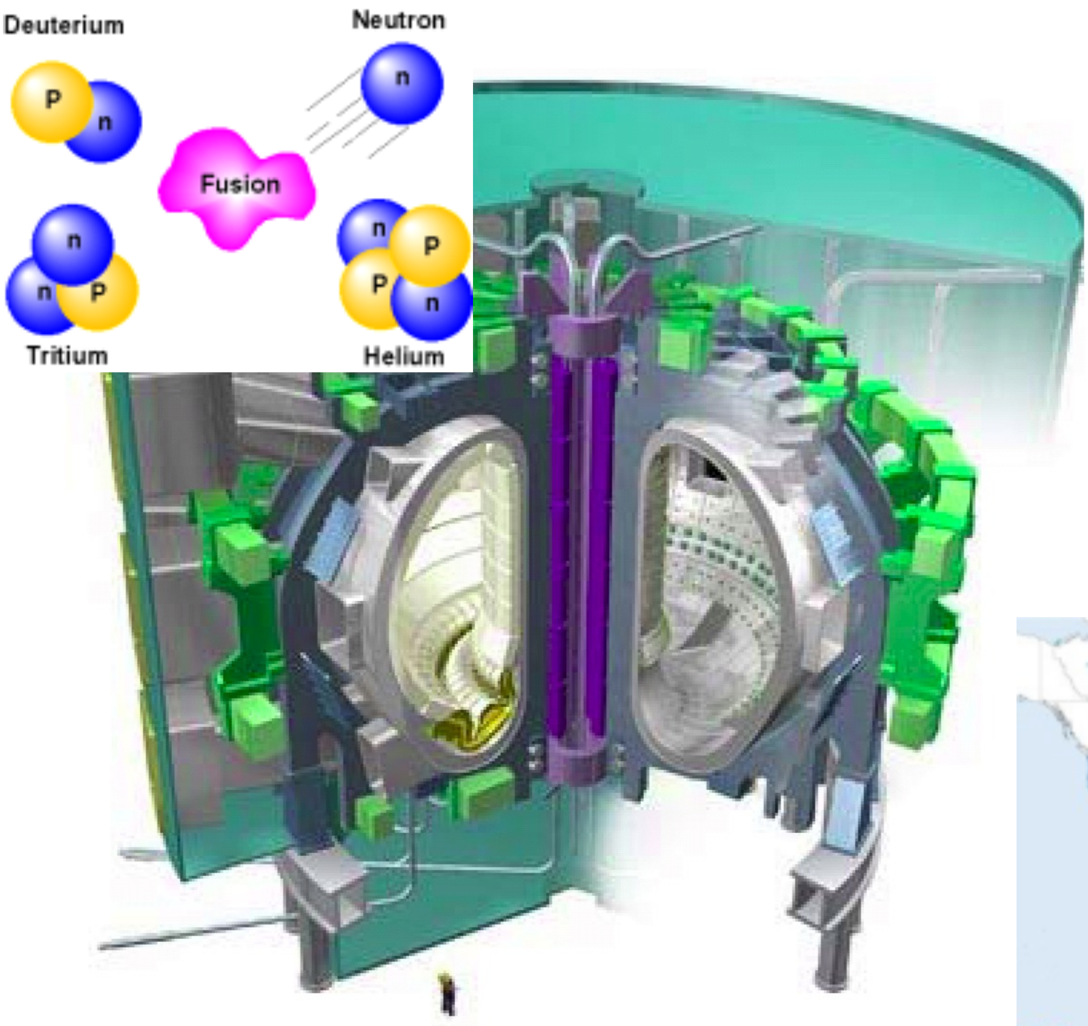
Fusion energy in the news
(just two examples)





Turbulent fusion plasmas: 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

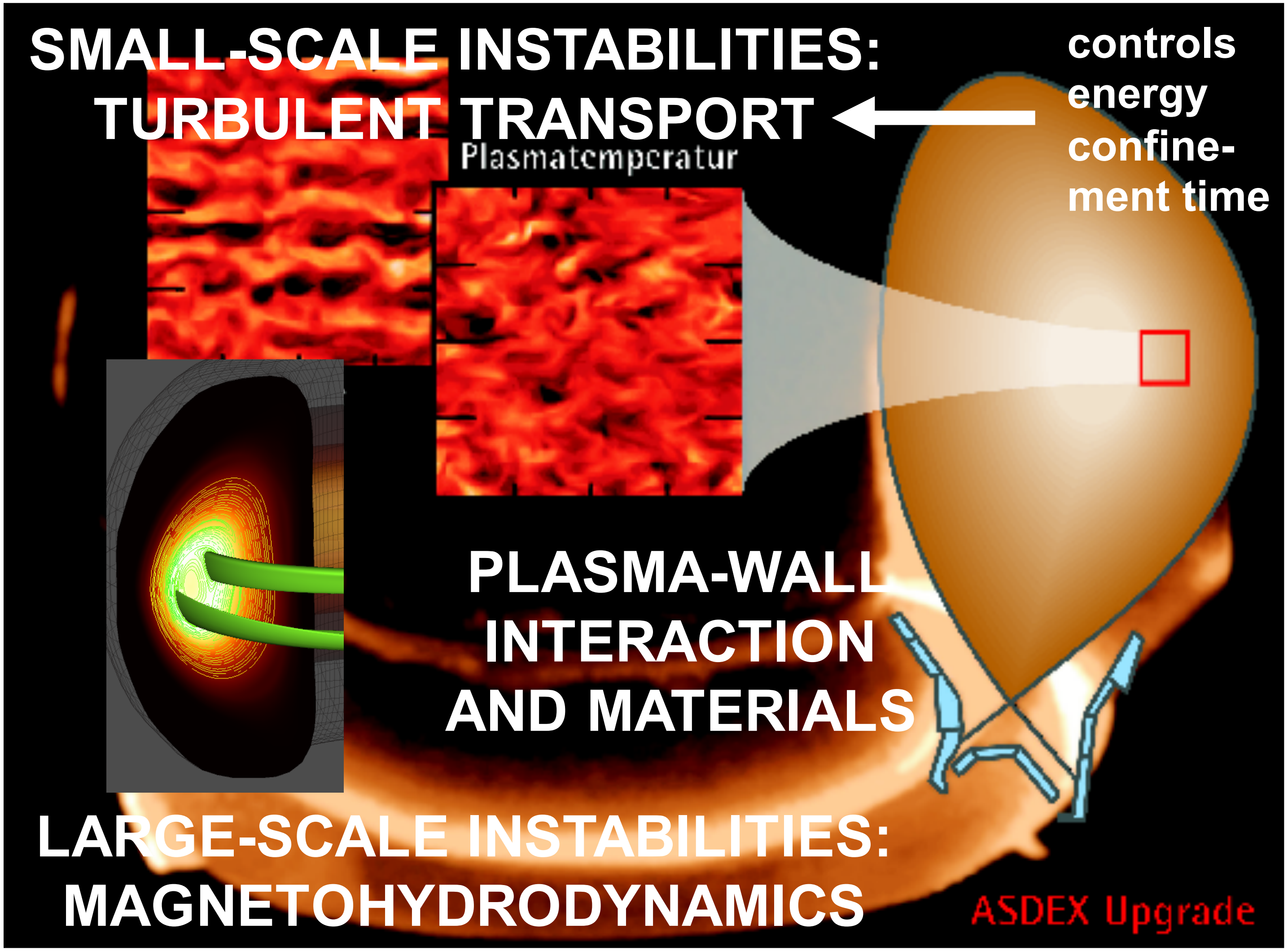
ITER CONSTRUCTION SITE IN SOUTHERN FRANCE

The resources for fusion energy are practically unlimited

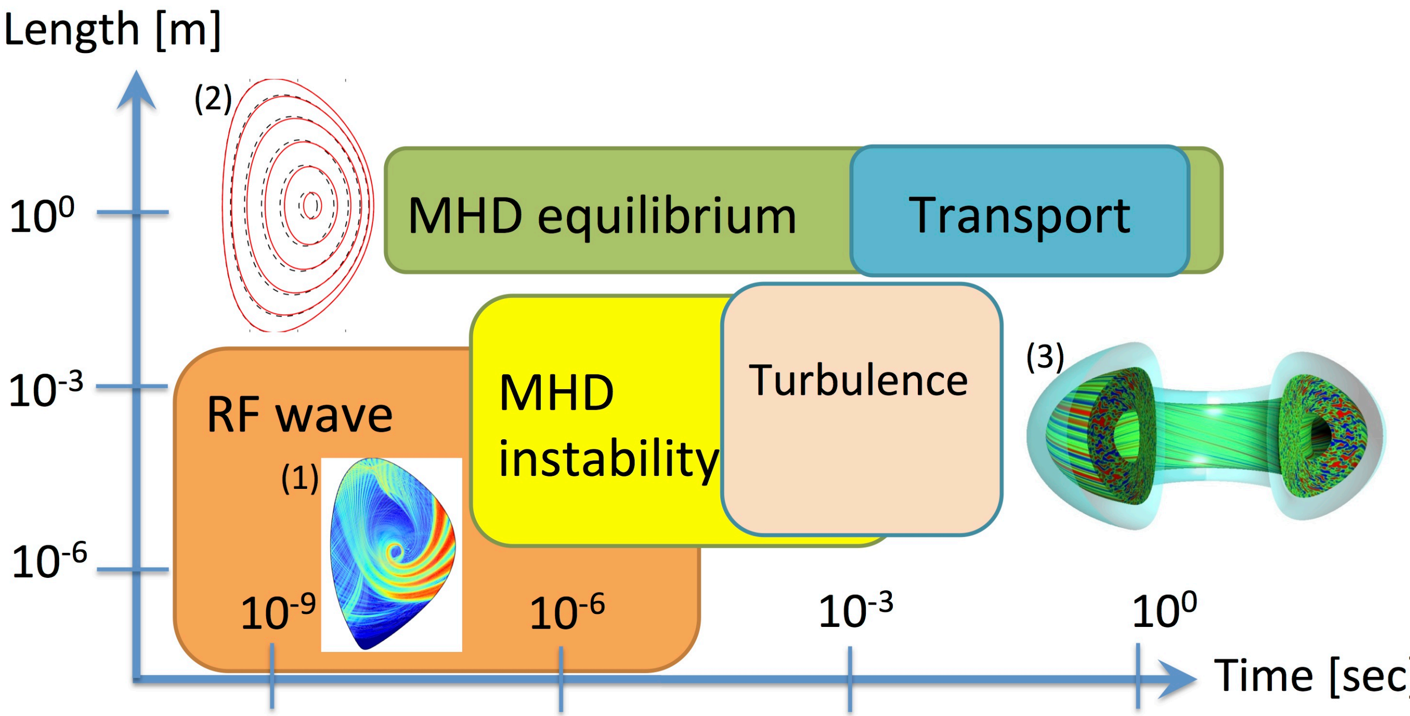
Deuterium in a bath tub full of water and Lithium in a used laptop battery suffice for a family over 50 years

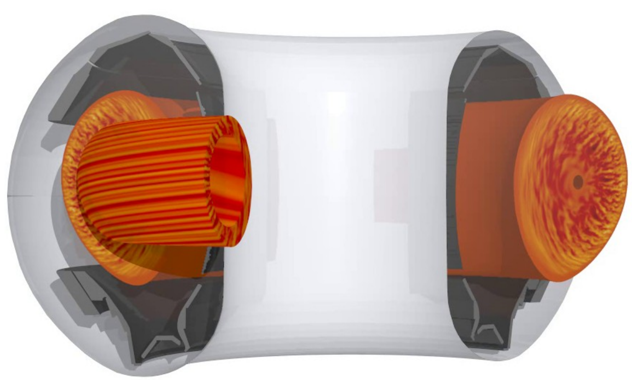
Extreme computing with the GENE code

3 key challenges for fusion physics



HUGE RANGE OF SPACE-TIME SCALES





Applied mathematics

Computer science

Plasma astrophysics

Complex systems

incl. **High Performance Computing** *and* **Data Analytics**

Kinetic description required

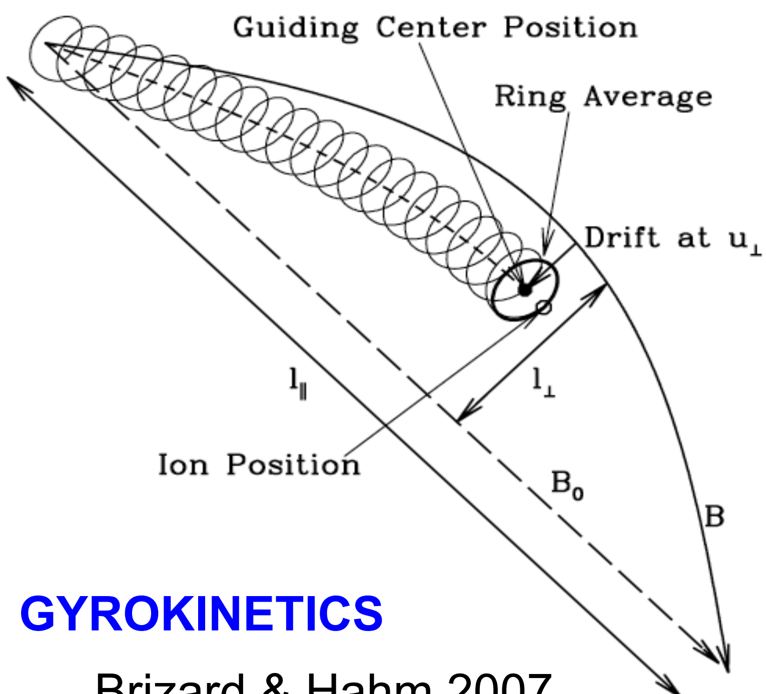
Weakly collisional system: 6D Vlasov-Maxwell equations

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \nabla f_\alpha + \frac{q_\alpha}{m_\alpha}\left[\mathbf{E} + \frac{\mathbf{v} \times \mathbf{B}}{c}\right] \cdot \nabla_v f_\alpha = 0$$

α = particle species

$$f_\alpha = f_\alpha(\mathbf{x}, \mathbf{v}, t)$$

...from the Liouville equation via the BBGKY hierarchy



GYROKINETICS

Brizard & Hahm 2007

Strong background magnetic field:

Eliminate fast gyromotion; consider slow dynamics of **guiding centers**

$$f = f(\mathbf{X}, v_\parallel, \mu; t)$$

$$\boxed{\frac{\partial f}{\partial t} + \dot{\mathbf{X}} \cdot \frac{\partial f}{\partial \mathbf{X}} + \dot{v}_\parallel \frac{\partial f}{\partial v_\parallel} = 0}$$

The gyrokinetic GENE code (Jenko+ PoP 2000)

http://genecode.org

- Modes of operation:
  - delta-f & full-f (gradient-driven, flux-driven)
  - flux-tube & full-flux-surface & full-torus

- Unique combination of various FDM, FEM, and spectral methods (CFD)

- Comprehensive physics: kinetic electrons, electromagnetic effects, collisions, realistic MHD equilibria, electron-scale turbulence…

- Publicly available, world-wide user base from 30+ scientific institutions

- Output to date: >150 papers (>20 PRLs)

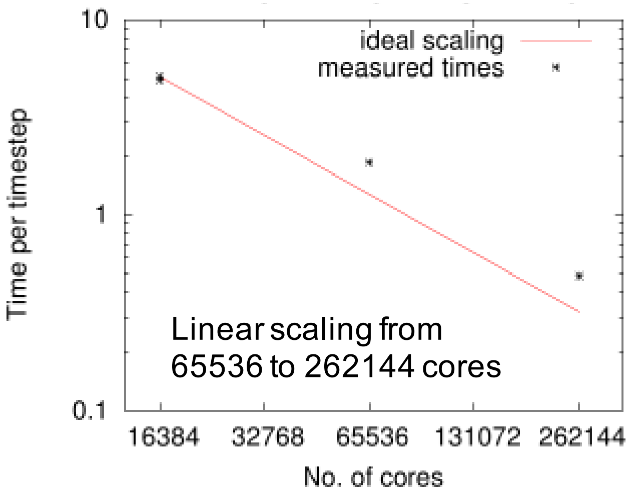
- Scales well on many leading HPC systems

GENE on top-level HPC resources

Ranked #1 out of 68 proposals in PRACE Early Access Call (2010)

First grid-based (gyro-)kinetic code to receive an INCITE Award (2016)

Strong scaling of GENE on BlueGene/P



Gyrokinetic Electromagnetic Numerical Experiment

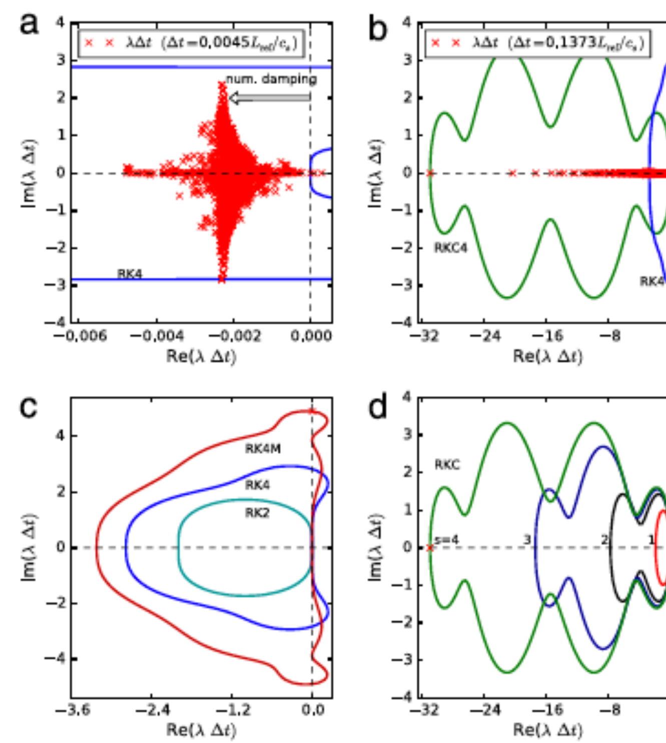
Some computational challenges

- 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
- GENE has been ported to GPGPU systems like Titan

GENE parallelization

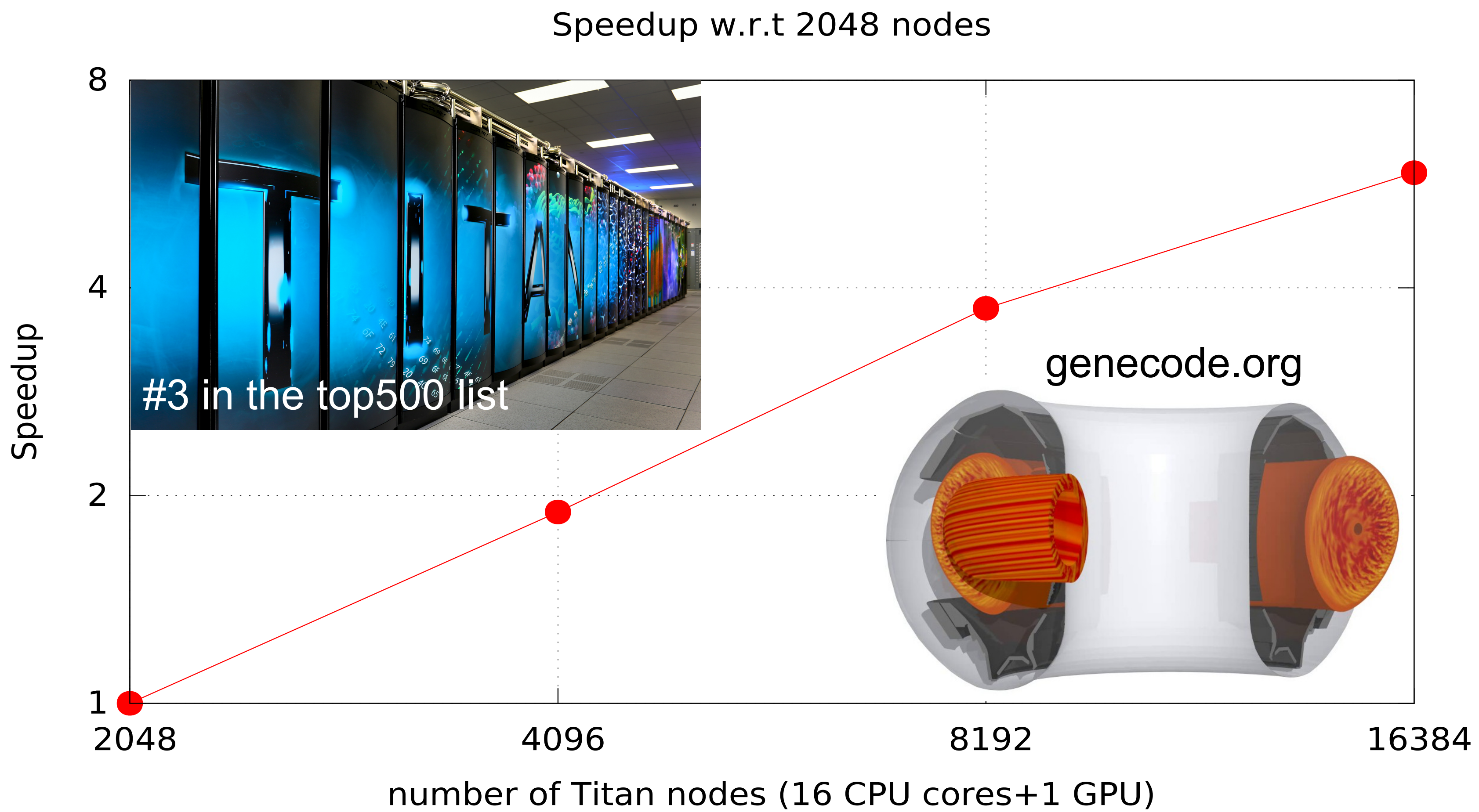
Parallelization/optimization strategy:

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



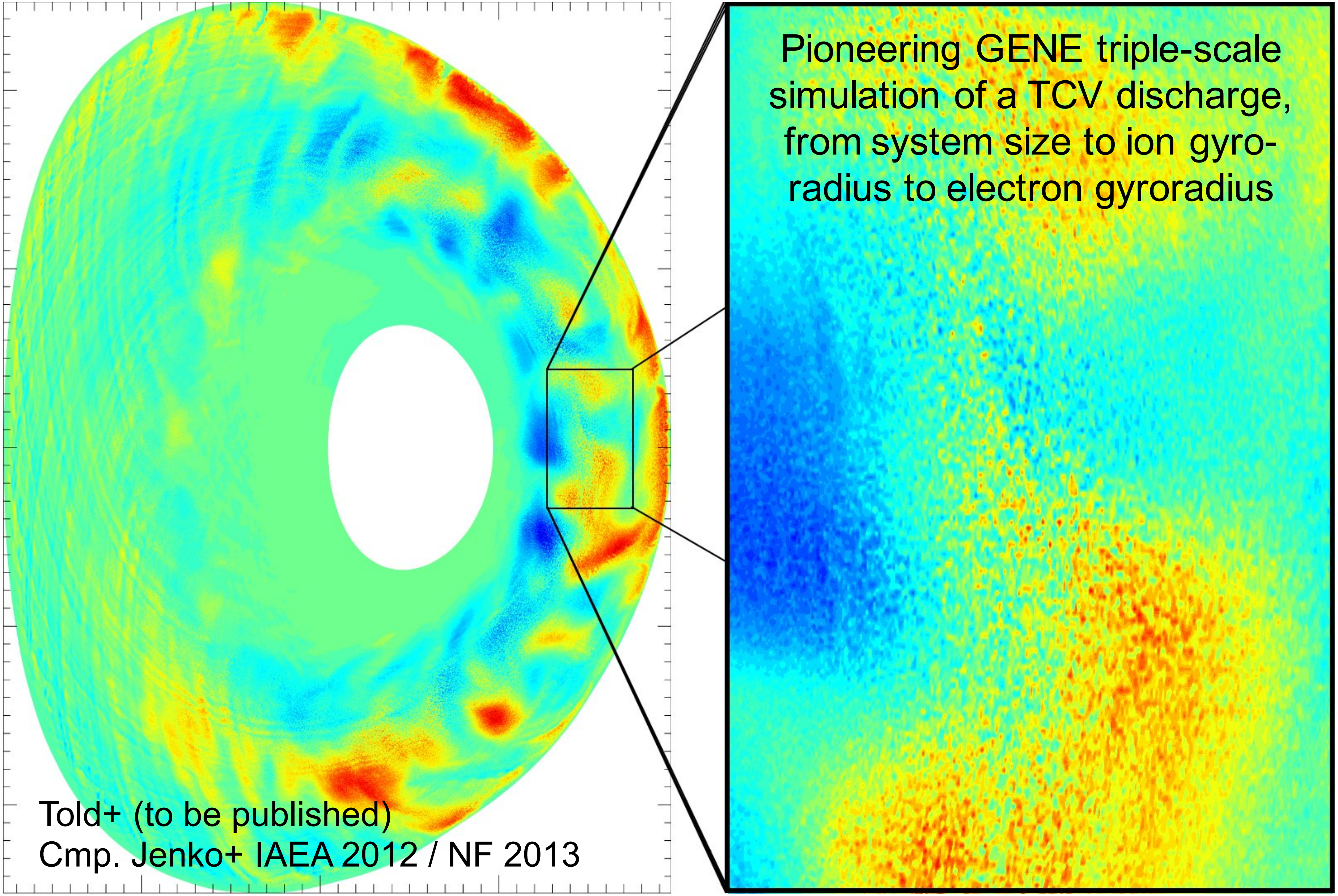
H. Doerk and F. Jenko,
Computer Physics Communications ***185****, 1938 (2014)*

Strong scaling of GENE code on Titan



Parallel performance of the GENE code on Titan: Strong scaling from 2k nodes to 16k nodes (corresponding to ~90% of the machine)

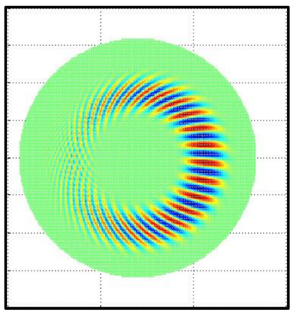
Multiscale simulations

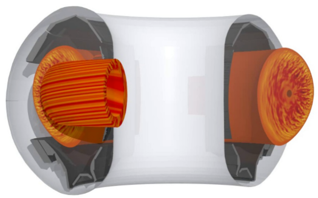


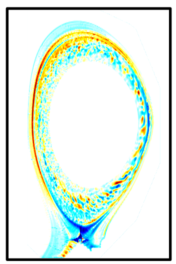
TOWARDS A "VIRTUAL" TOKAMAK

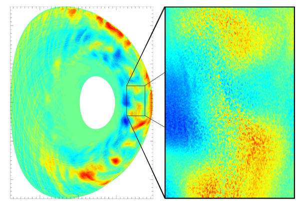
Goals: prepare and interpret **ITER** discharges, guide the development of **power plants**

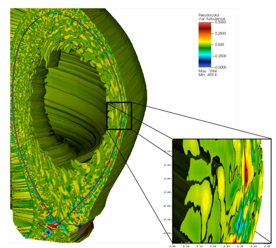
Increasing fidelity & modeling capability with increasing computing power

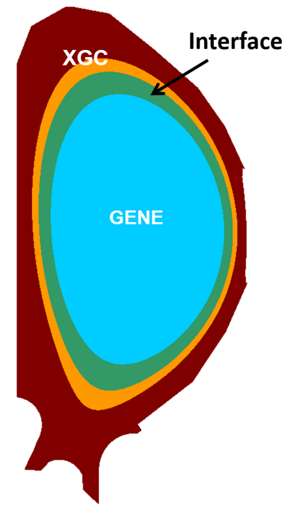


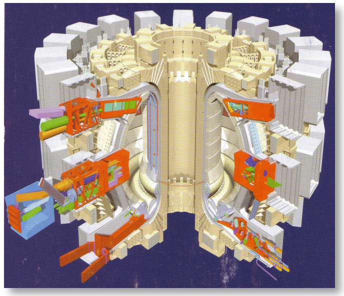












Gigaflops
Core: ion-scale electrostatic physics in simplified geometry

Teraflops
Core: adding kinetic electron electromagnetic physics in a torus

Edge: ion+neutral electrostatic physics in a torus

Petaflops
Core: adding electron-scale physics

Edge: adding kinetic electron electrostatic physics

Exaflops
Core-edge coupled studies of whole-device ITER, incl. turbulence, MHD instability, fast particles, heating, and plasma-wall interactions

Beyond
Whole device modeling of all relevant fusion science

Acknowledgements: ECP

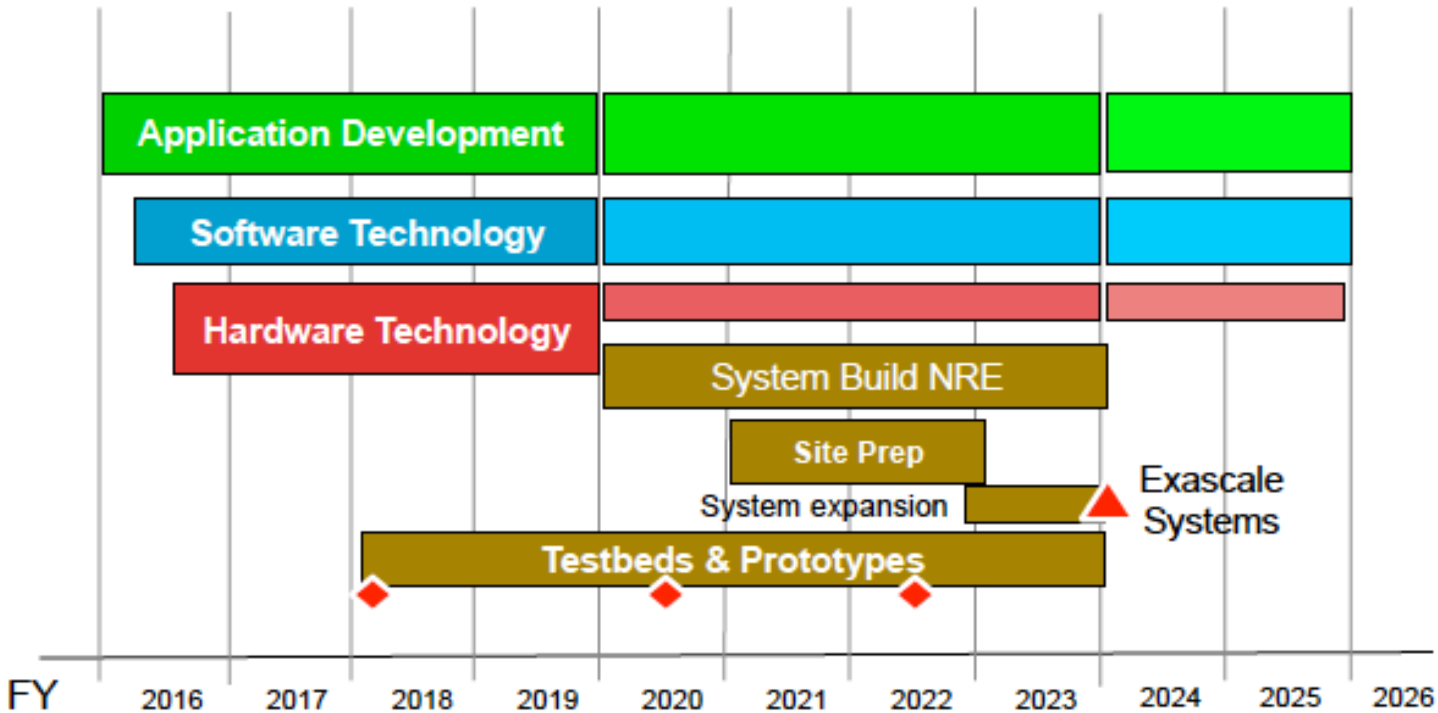
ECP Timeline

USA: Exascale Computing Project

https://exascaleproject.org

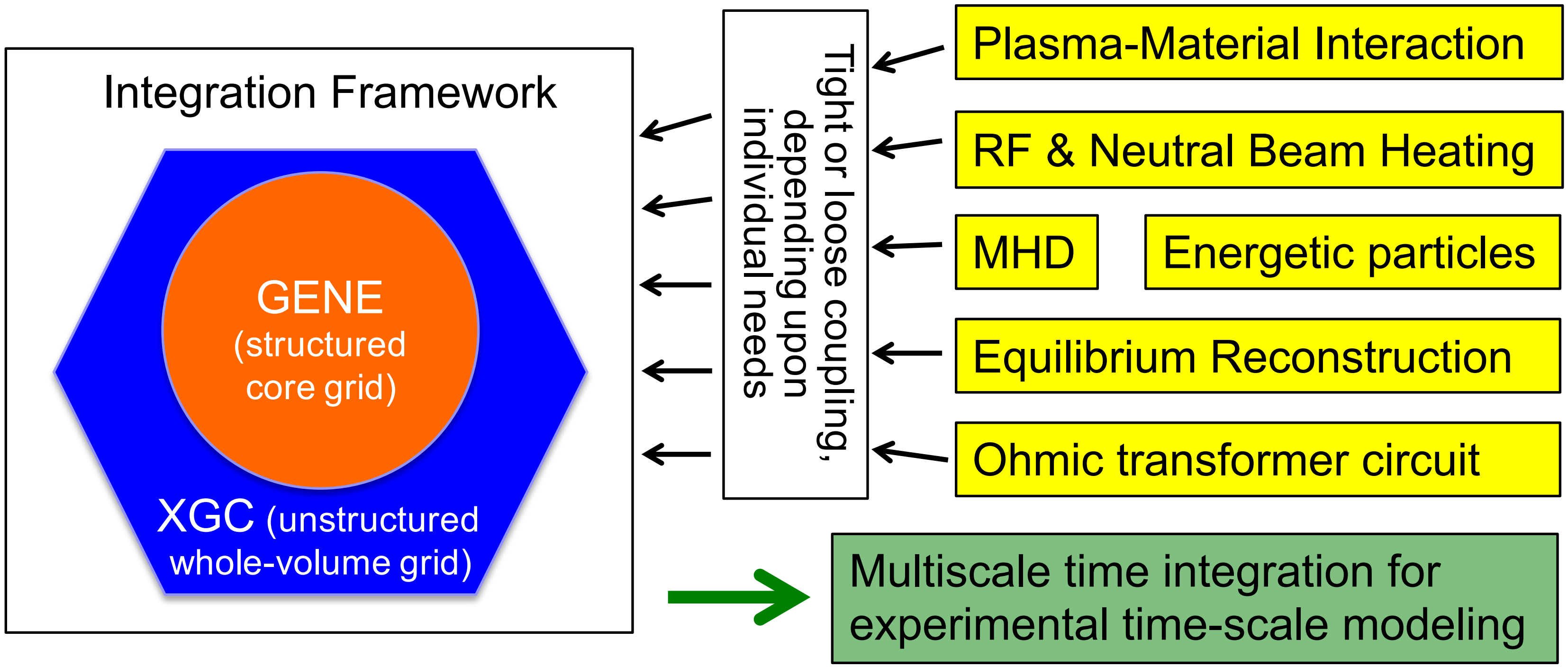
The Project has three phases:

- Phase 1 – R&D before DOE facilities exascale systems RFP in 2019
- Phase 2 – Exascale architectures and NRE are known. Targeted development
- Phase 3 – Exascale systems delivered. Meet Mission Challenges



EXASCALE COMPUTING PROJECT

Step-wise approach to kinetic-based whole device modeling



1st step

- Build an integration framework that can service tight-coupling of the living GENE and XGC
- Demonstrate the scalability and portability of the integration framework on current leadership class HPCs

2nd step

- Expand the integration framework to a whole-device modeling
  - Serve both tight and loose couplings
- Adapt to the moving hardware & software targets toward exascale

Beyond “brute force”

Sparse grid combination technique

Cartesian grid

•Regular data structure

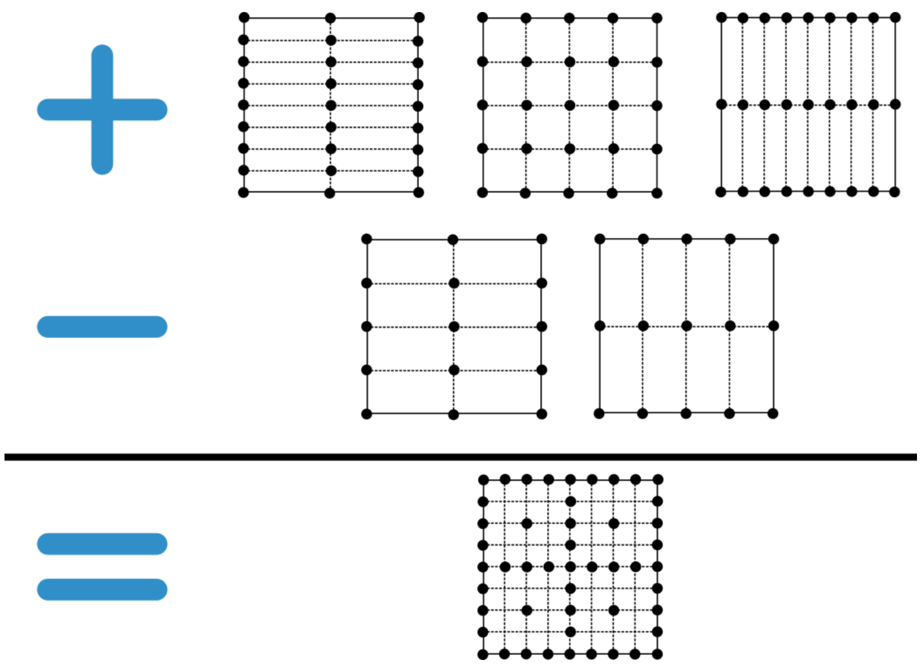
•Huge number of grid points for high-dimensional problems

"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**



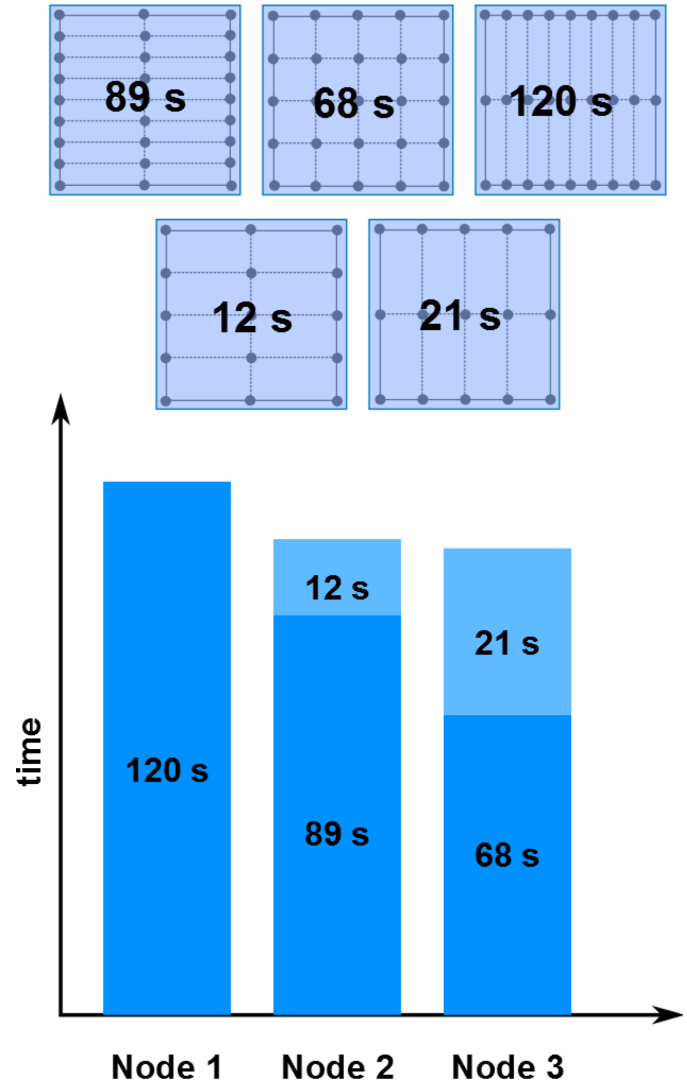
A new level of parallelism

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

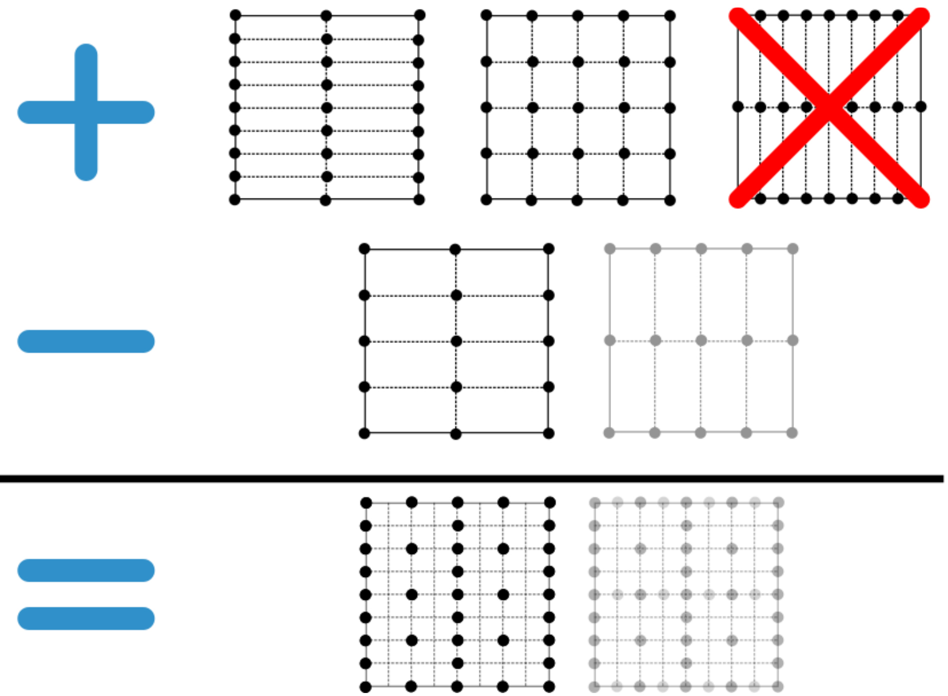
- A simple **load-model** estimates the runtime required for each grid
- A **scheduler** creates an optimal load balancing to minimize idle cores



Spin-off: Algorithmic fault tolerance

Hardware failures (10^{5-7} 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

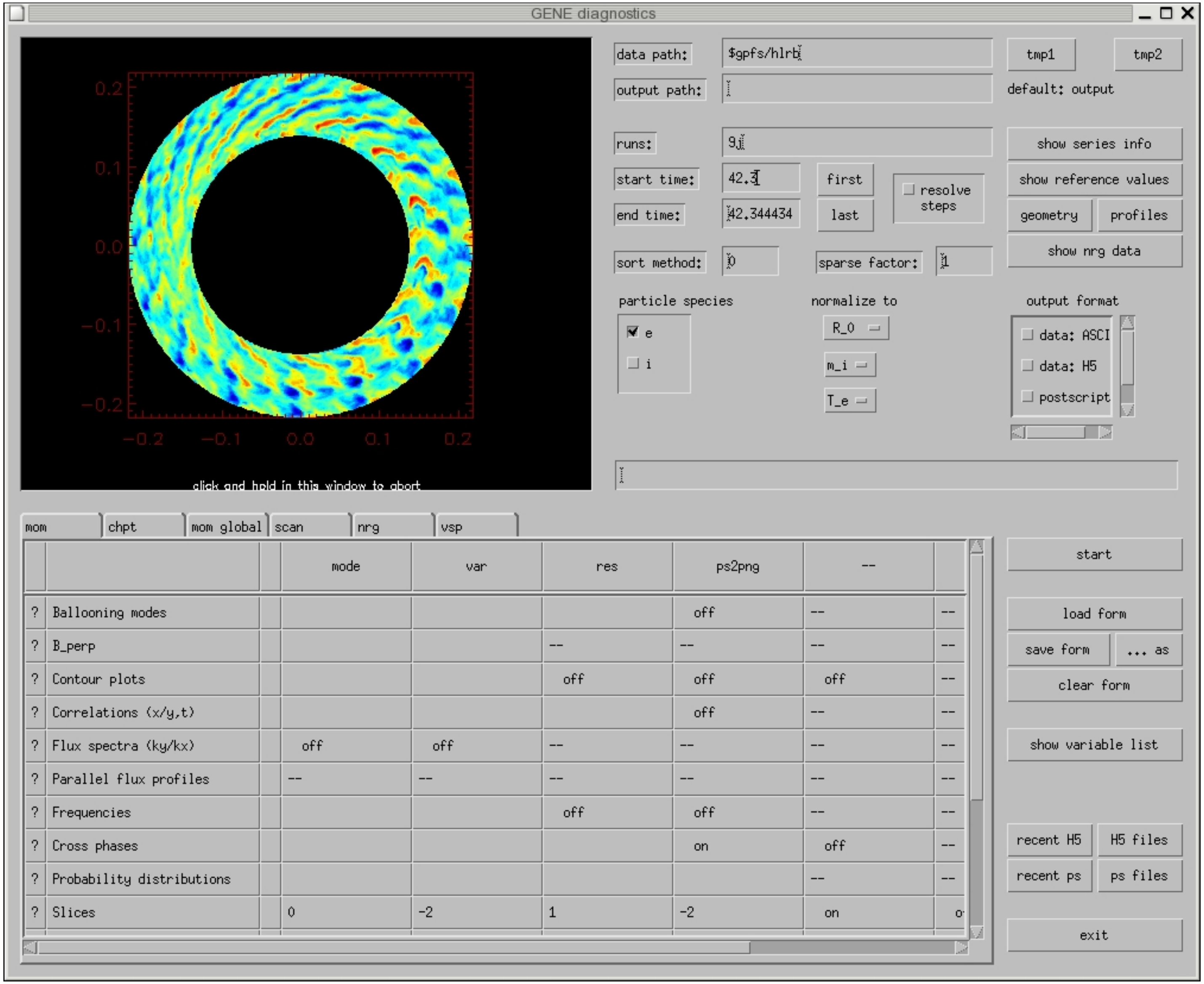
Such techniques may be very useful on future exascale architectures

Some practical challenges

Building an international user base

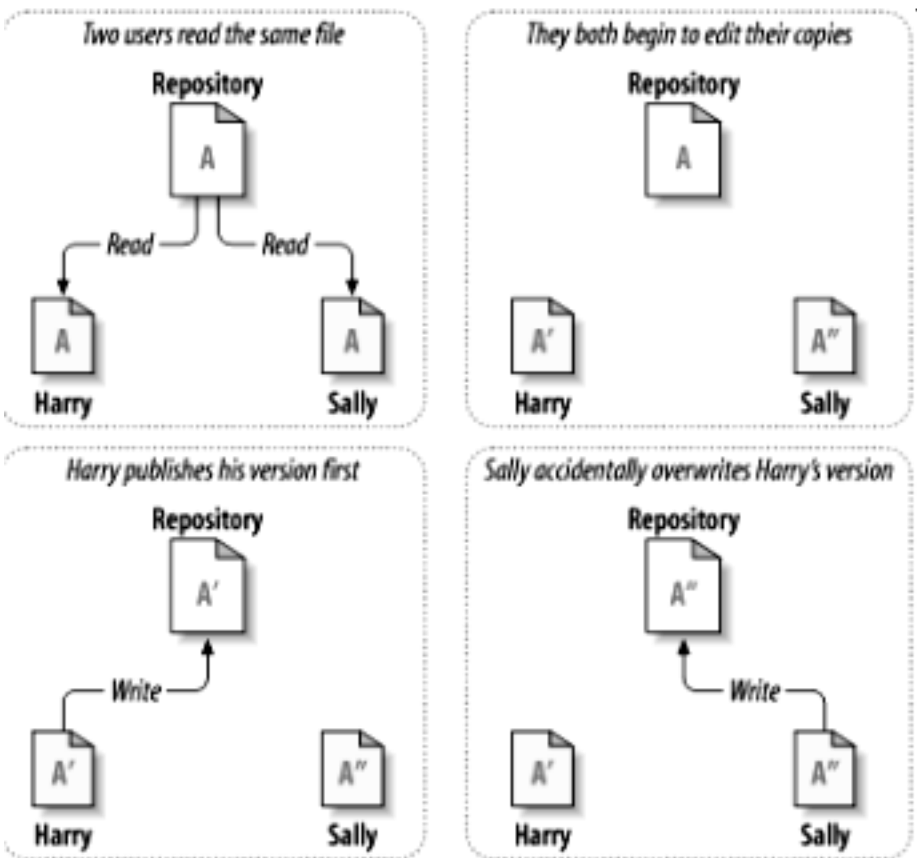
- Developing a cutting-edge code like GENE requires many man-years and a lot of endurance
- Much work went into making the code user-friendly
- Release versions of the code are distributed via a website
- The code comes with documentation (50-page manual and 50-page tutorial)
- A lot of time and energy also goes into direct user support
- This is a service to the community, with costs and benefits

Output: GENE Diagnostics Tool



The need for version control

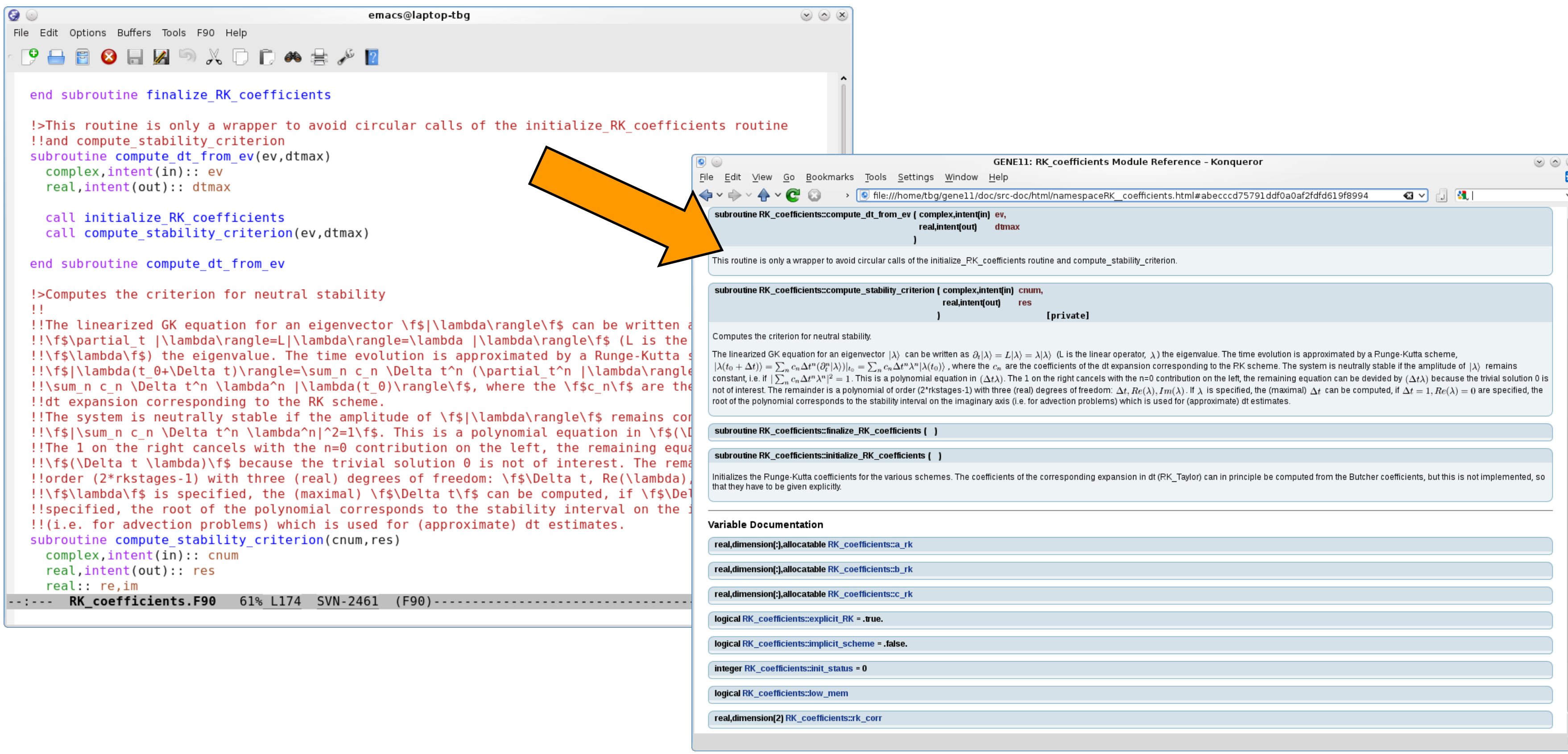
- Keep track of changes(!) – even in single developer cases, e.g., for more efficient bug hunting
- Avoid "destructive" interference between developers/developer groups:



taken from: [*Version Control with Subversion, http://svnbook.red-bean.com*]

In-Source-Documentation

- Even for single developers:
  Try to add meaningful comments in the source code! There's always the possibility that your code shall be used/extended by other people
- OpenSource software tools like www.doxygen.org help to create a HTML/PDF documentation:



Test Suite

- ***Even the most careful developer introduces bugs***
  … and often at places which you would never guess

- Hence, define a ***certain set of tests*** which
  - is representative for your typical simulations
  - checks the parallelization in different ways
  - can be run in a reasonable amount of time

  and ***which should be checked*** before every major commit and after each port to a new machine/compiler (compiler bugs)!!

- The GENE-Test-Suite is a perl-based script:
  - calling the code subsequently with either a default set of input files or some special purpose set (like high-resolved simulations)
  - comparing the output up to a certain accuracy (the machine accuracy depends on the architecture!)
  - monitoring the wall clock time to check the efficiency of the installation

  (the parallelization is checked in GENE itself)

The future

Big Data Meets HPC

Two recent waves of innovations affecting science (= main drivers of the expansion of the role of the mathematical sciences):

High Performance Computing & Big Data

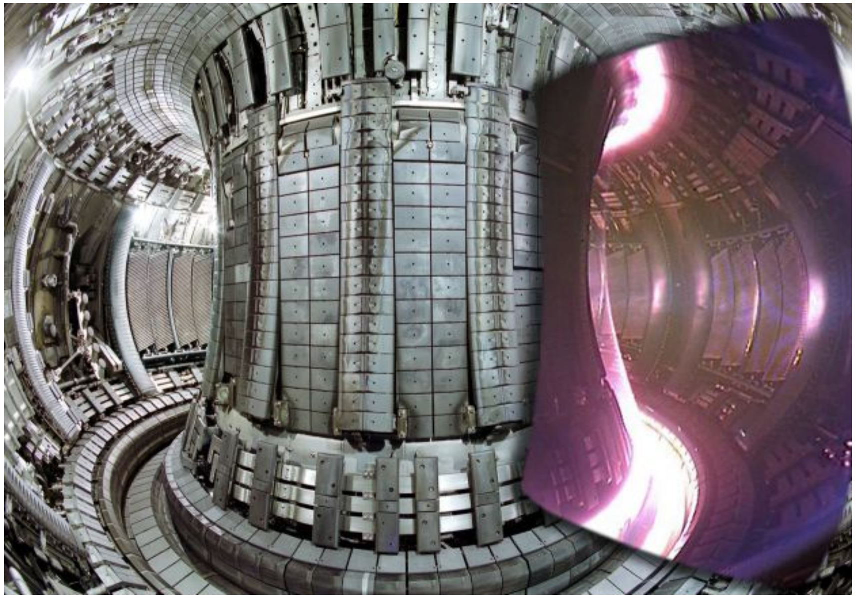
Currently, these themes are usually addressed rather independently – but they are intrinsically linked:

- **HPC needs Big Data** for dealing with increasingly large data sets
  - ✓ Communication bottleneck on the path to exascale computing
  - ✓ Develop novel ways of representing, reducing, reconstructing, and transferring huge amounts of data *(need new algorithms!)*
- **Big Data needs HPC** for analyzing increasingly large data sets
  - ✓ Data analytics becomes ever more compute-intensive

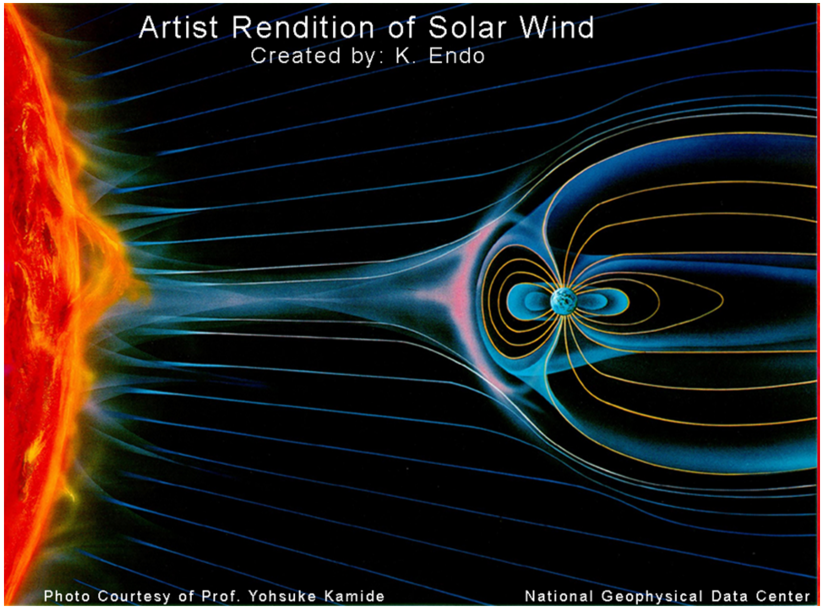
IPAM Long Program, UCLA (fall 2018)

Two key challenges

Virtual fusion devices



Space weather prediction



Plasma turbulence:

Where fascinating physics, extreme computing, and global challenges meet