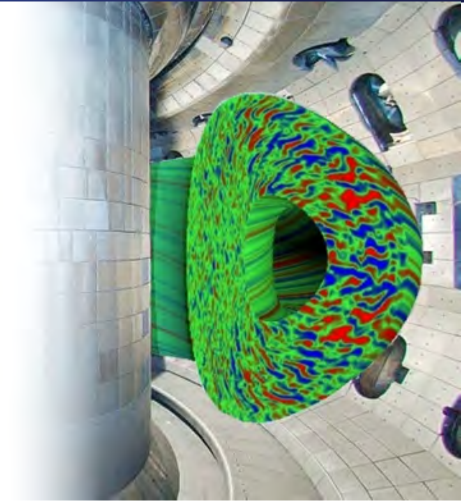
The Advanced Tokamak Modeling Environment (AToM) for Fusion Plasmas

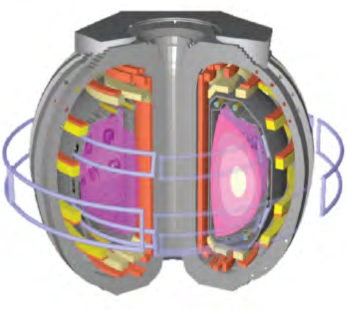
by
J. **Candy**[1] on behalf of the **AToM team**[2]
[1]General Atomics, San Diego, CA
[2]See presentation

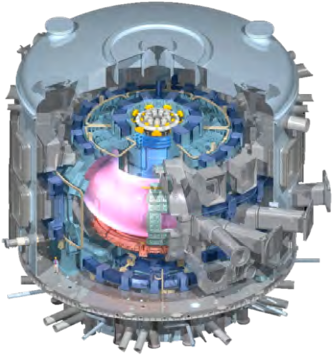


Presented at the
2018 SciDAC-4 PI Meeting
Rockville, MD
23-24 July 2018

AToM Modeling Scope and Vision

Present-day tokamaks	Upcoming burning plasma	Future reactor design
DIII-D	**ITER**	**DEMO**





AToM (2017-2022) Research Thrusts

- AToM^0 was a **3-year SciDAC-3 project** (2014-2017)
- AToM is a new **5-year SciDAC-4 project** (2017-2022)
- The scope of AToM is broad, with **six research thrusts**

`scidac.github.io/atom/`

1. AToM environment, performance and packaging
2. Physics component integration
3. Validation and uncertainty quantification
4. Physics and scenario exploration
5. Data and metadata management
6. Liaisons to SciDAC partnerships

AToM Team

Institutional Principal Investigators (FES)

Jeff Candy	General Atomics
Mikhail Dorf	Lawrence Livermore National Laboratory
David Green	Oak Ridge National Laboratory
Chris Holland	University of California, San Diego
Charles Kessel	Princeton Plasma Physics Laboratory

Institutional Principal Investigators (ASCR)

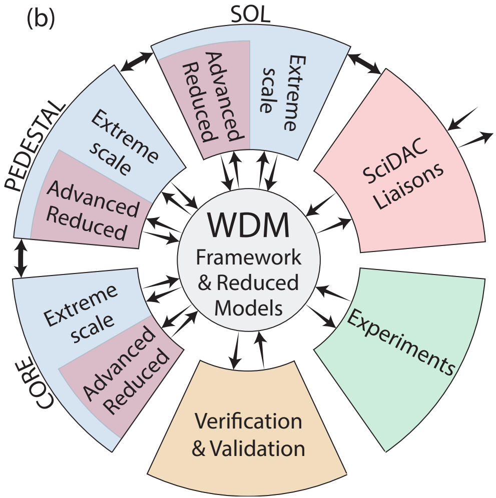
David Bernholdt	Oak Ridge National Laboratory
Milo Dorr	Lawrence Livermore National Laboratory
David Schissel	General Atomics

AToM Team

Funded collaborators (subcontractors in **green**)

O. Meneghini, S. Smith, P. Snyder, D. Eldon, E. Belli, M. Kostuk	GA
W. Elwasif, M. Cianciosa, J.M. Park, G. Fann, K. Law, D. Batchelor	ORNL
N. Howard	MIT
D. Orlov	UCSD
J. Sachdev	PPPL
M. Umansky	LLNL
P. Bonoli	MIT
Y. Chen	UC Boulder
R. Kalling	Kalling Software
A. Pankin	Tech-X

AToM Conceptual Structure

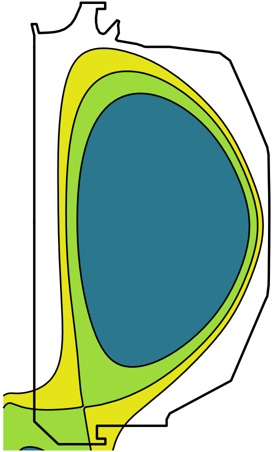


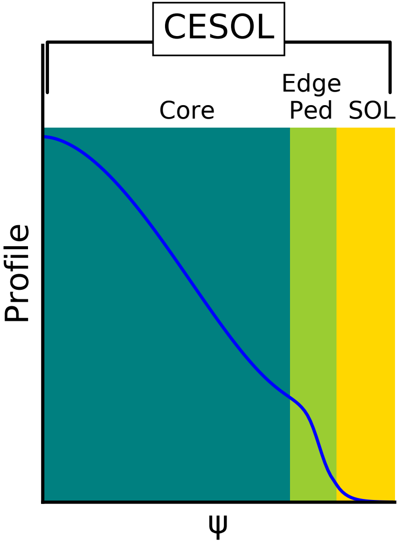
1. Access to experimental data
2. Outreach (liaisons) to other SciDACs
3. Verification and validation, UQ, machine learning
4. Support HPC components
5. Framework provides glue

Adapted from Fig. 24 of *Report of the Workshop on Integrated Simulations for Magnetic Fusion Energy Sciences (June 2-4, 2015)*

Tokamak physics spans multiple space/timescales

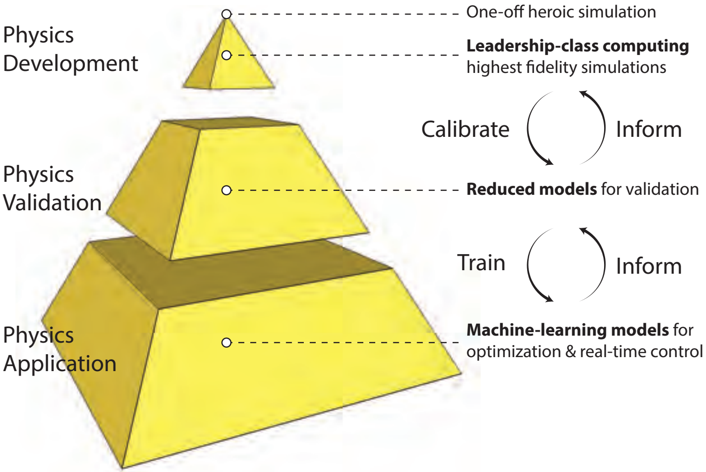
Core-edge-SOL (CESOL) region coupling





Fidelity Hierarchy (Pyramid)

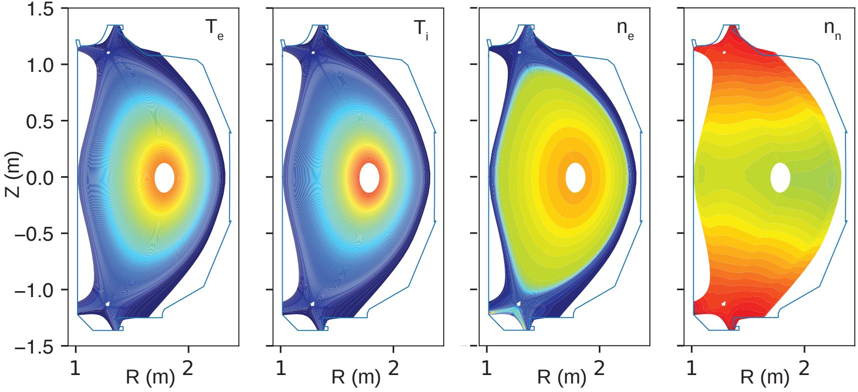
Range of models all the way up to leadership codes



Strive for true WDM capability

Core-edge-SOL (CESOL) region coupling

- Iterative solution procedure to match boundary conditions between regions
- **15 components** coupled



AToM Supports two core-edge integrated workflows

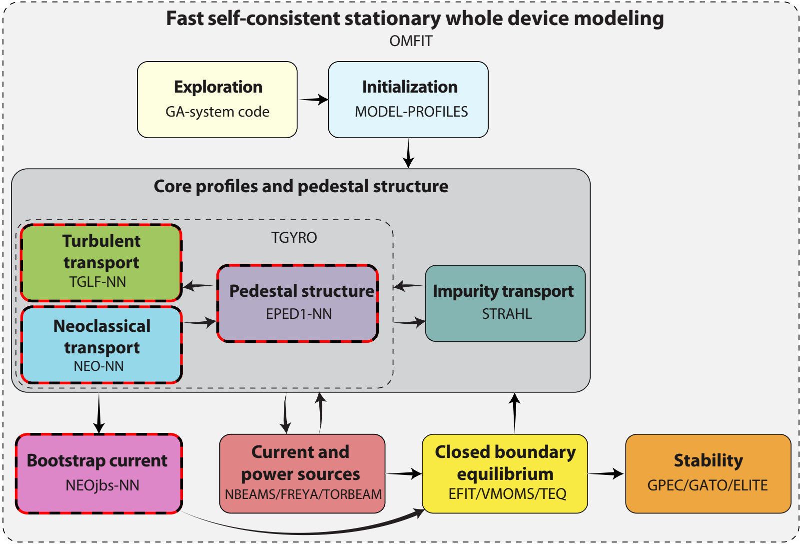
OMFIT-TGYRO and IPS-FASTRAN

- **OMFIT-based** core-pedestal (FAST) workflow:
  - Workflow manager with flexible tree-based data handling/exchange
  - Can use NN-accelerated models for EPED/NEO/TGLF
  - Transport solver based on TGYRO+TGLF

- **IPS-based** core-pedestal-SOL (HPC) workflow:
  - Framework/component architecture using existing codes
  - File-based communication (plasma state)
  - Multi-level (HPC) parallelism
  - Transport solver based on FASTRAN+TGLF

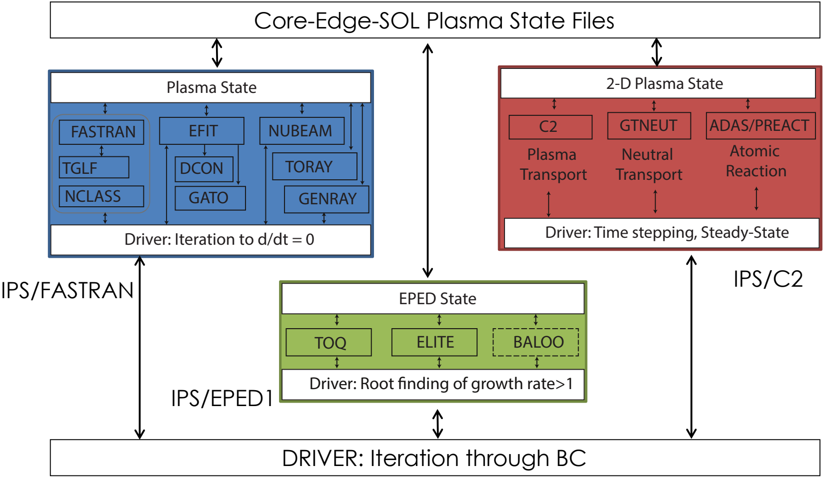
AToM Supports two core-edge integrated workflows

(1) OMFIT-TGYRO



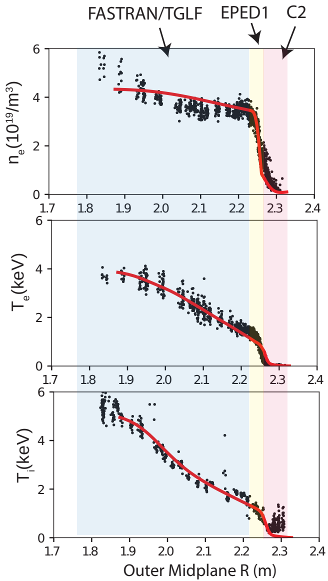
AToM Supports two core-edge integrated workflows

(2) IPS-FASTRAN



AToM Supports two core-pedestal integrated workflows

(2) IPS-FASTRAN: DIII-D ITER baseline discharge

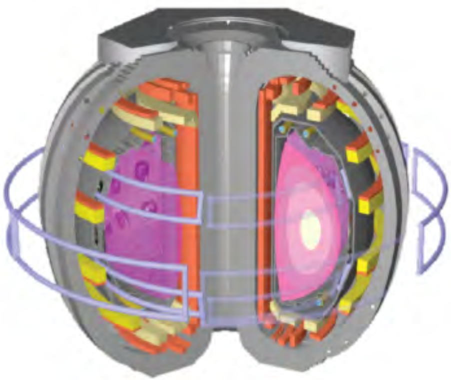


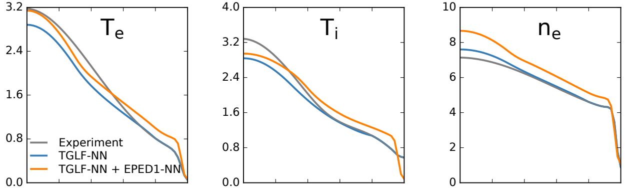
- Manage execution of 15 component codes
  FASTRAN+TGLF+NCLASS+EPED(ELITE+TOQ)+
  NUBEAM+TORAY+EFIT+C2+GTNEUT+CARRE+
  C2MESH+CHEASE+DCON+PEST3
- **Iterative coupling** of core, pedestal, SOL
  - AToM **CESOL** workflow
- Self-consistent heating and current drive
  - NUBEAM, TORAY, GENRAY
- Accuracy highly dependent on **TGLF** and **EPED**

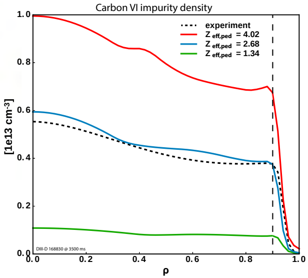
Application: Present day tokamaks

DIII-D (San Diego)

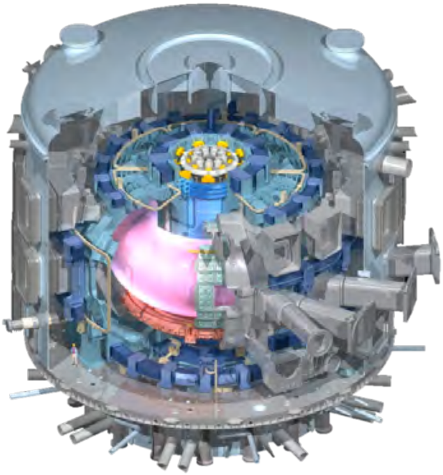
Core-pedestal impurity profile prediction (OMFIT-based)

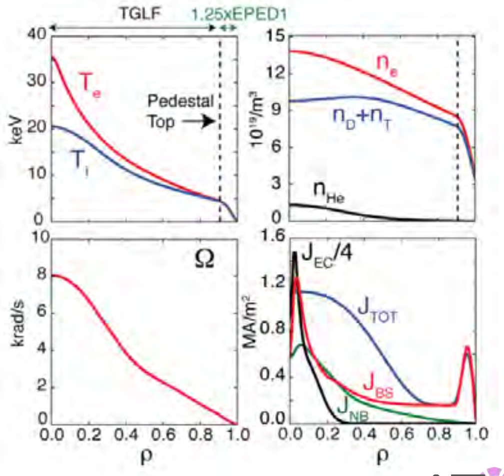






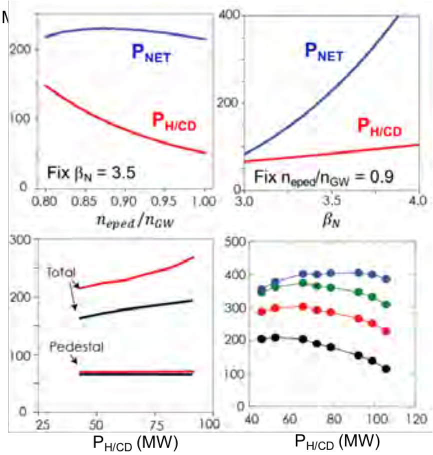
ITER steady-state hybrid scenario modeling (IPS-based)





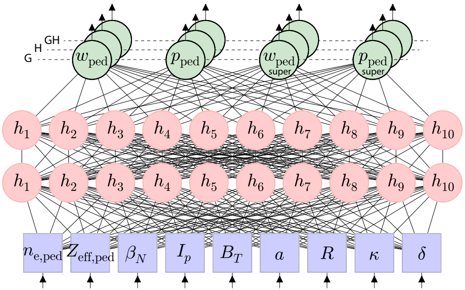
ATOM

C-AT DEMO reactor modeling (IPS-based)

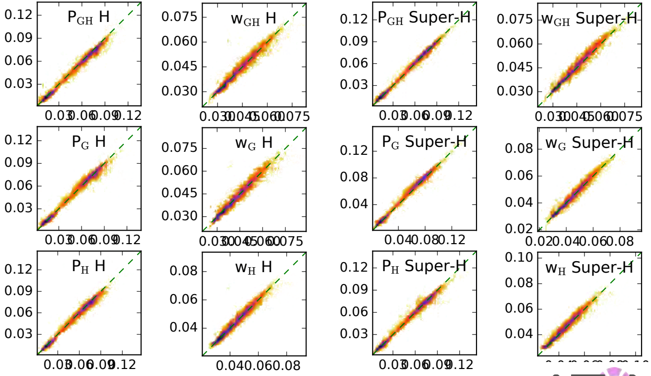


Create EPED1-NN neural net from EPED1 model

- **10 inputs → 12 outputs**
- **normal H mode** solution
- **Super-H mode** solution
- EPED1-NN tightly coupled in TGYRO

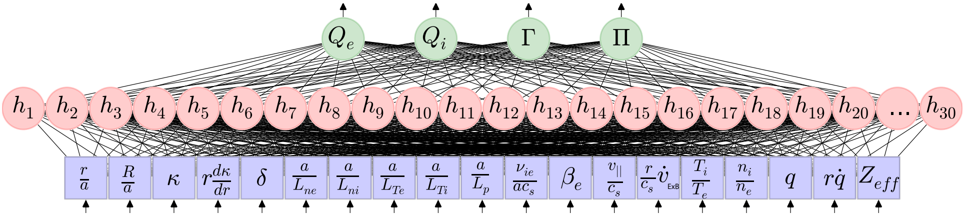


- Database of **20K EPED1 runs** (2M CPU hours)
- DIII-D(3K), KSTAR(700), JET(200), ITER(15K), CFETR (1.2K)



Create TGLF-NN neural net from TGLF reduced model

- **23 inputs** $\rightarrow$ **4 outputs**
- Each dataset has 500K cases from 2300 multi-machine discharges
- Trained with TENSORFLOW
- Must be retrained as TGLF model is updated
- TGLF itself derived from **HPC CGYRO simulation**



TGLF

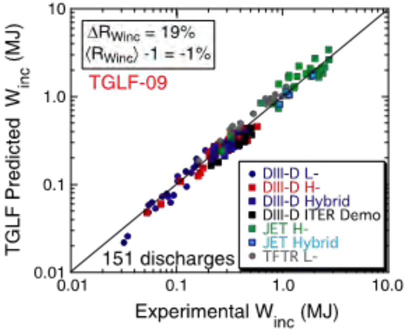
Centerpiece of all AToM predictive modeling workflows

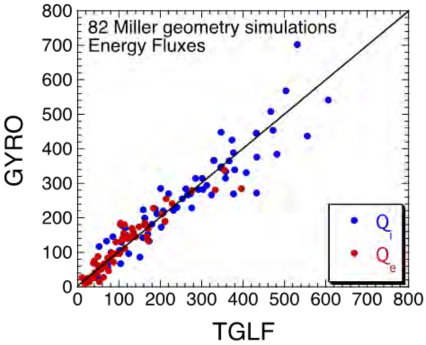
- **Reduced model of nonlinear gyrokinetic flux**
- **Quality of profile prediction** for modeling workflows only as good as the **quality of the reduced model** of turbulent flux
- TGLF is the heart of AToM profile-prediction capability
  - linear gyro-Landau-fluid **eigenvalue solver**
  - coupled with sophisticated **saturation rule**
  - evaluate quasilinear fluxes over range $0.1 < k_\theta \rho_i < 24$
- Saturated potential intensity
  - derived from a database of nonlinear GYRO simulations
  - database resolves only long-wavelength turbulence: $k_\theta \rho_i < 1$
- TGLF **10 million** to **one billion** times faster than GK

TGLF

Ongoing calibration with CGYRO leadership simulations

- TGLF temperature predictions validated with ITPA database
- L-mode edge, EM saturation **discrepancies**
- CGYRO **multiscale simulations needed** for these conditions





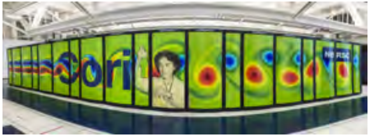
CGYRO

New nonlocal spectral solver for collisional plasma edge

- New (NEO) coordinates, discretization, array distribution
  - **Pseudospectral** velocity space (ξ, v)
  - Fluid limit recovered as $\nu_e \to \infty$ (Hallatschek)
  - 5th-order **conservative upwind** in θ
- Extended physics for **edge plasma**
  - Sugama collision operator (numerically self-adjoint)
  - Sonic rotation including modified Grad-Shafranov
- **Arbitrary wavelength** formulation targets multiscale regime
- **Wavenumber advection** scheme (profile shear/nonlocality)
- Target petascale and exascale architectures (GPU/multicore)
  - **cuFFT/FFTW**
  - **GPUDirect MPI** on compatible systems
  - All kernels hybrid **OpenACC/OpenMP**
- Generate future database for TGLF edge calibration

CGYRO optimization on leadership systems

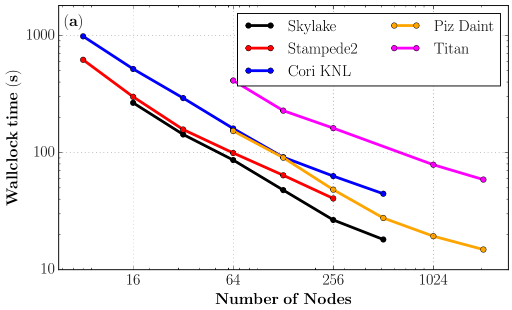
	Cori	Stampede2	Skylake	Titan	Piz Daint
Architecture	CPU	CPU	CPU	CPU/GPU	CPU/GPU
CPU Model	Xeon Phi 7250	Xeon Phi 7250	Xeon Plat 8160	Opteron 6274	Xeon ES-2690 v3
GPU Model				Tesla K20X 6GB	Tesla P100 16GB
Threads/node	272 (128 used)	272 (128 used)	96	16/2688	12/3584
TFLOP/node	3.0	3.0	3.5	1.5 (0.2+1.3)	4.5 (0.5+4.0)
Nodes	9668	4200	1736	18688	5320



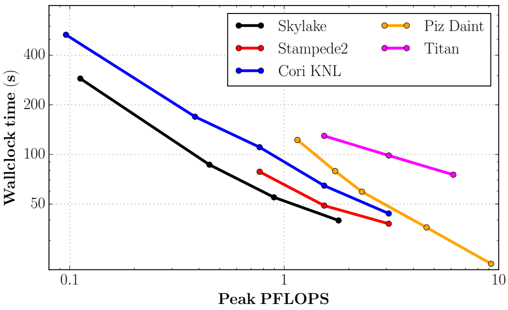
CGYRO Strong scaling

Performance on mid-scale test case

Increasing nodes



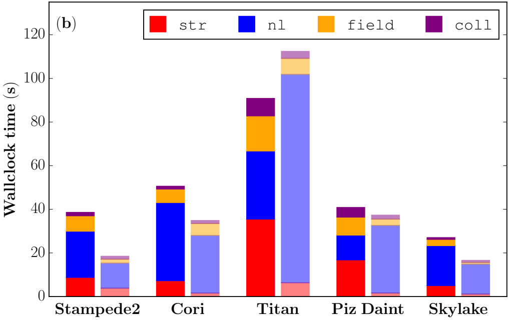
Increasing fraction of peak



CGYRO Kernel performance

Solid/faint bars show computation/communication time

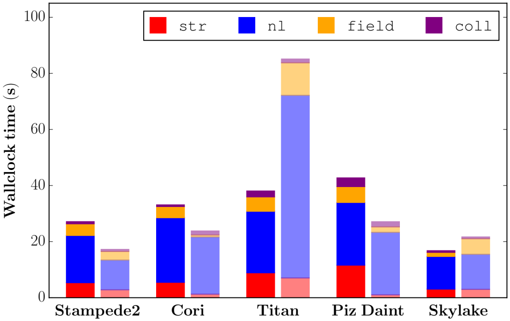
Equal 128 nodes



`str` = streaming/upwind kernel

`nl` = nonlinear kernel

Equal 1.6 PFLOP



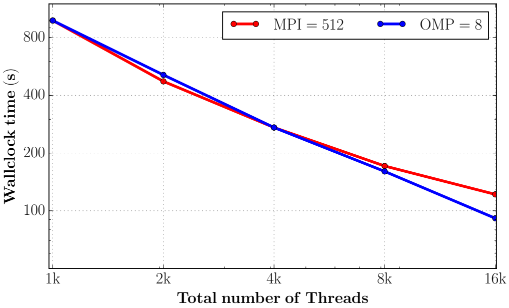
`field` = field solve kernel

`coll` = collision kernel (implict)

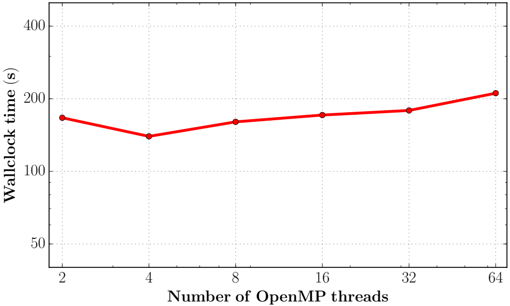
CGYRO OpenMP performance

- Results for **NERSC Cori KNL** (use 128 threads per node)
- Almost **perfect tradeoff** between MPI tasks and OpenMP threads

OMP vs MPI strong scaling



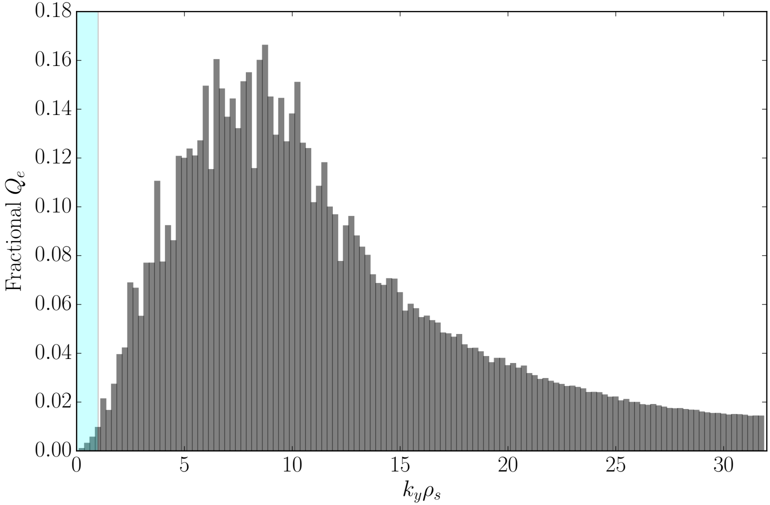
OMP-MPI tradeoff



CGYRO designed from scratch for multiscale

Arbitrary-wavelength spectral formulation

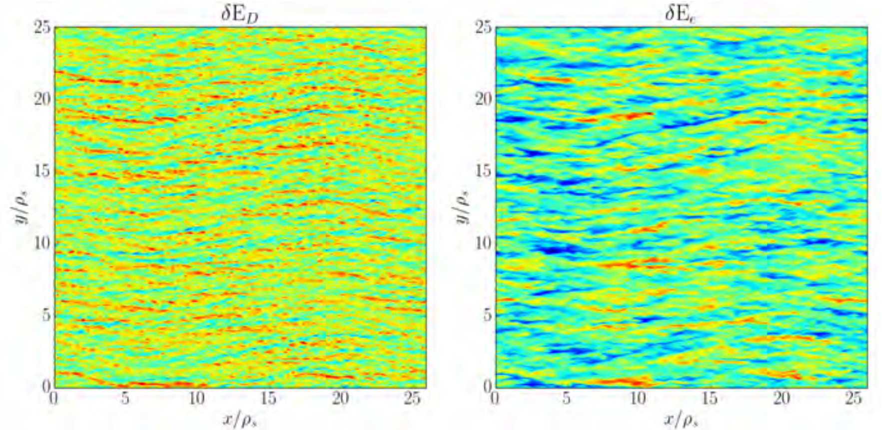
Traditional ion-scale domain shown in **blue**



CGYRO

Low-rotation DIII-D ITER baseline discharge (Haskey, Grierson)

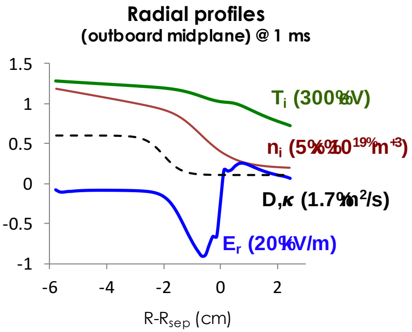
- Nearly all electron flux arises from **multiscale regime**
- Experimental value $Q_e/Q_{\mathrm{GB}} \simeq 8$ accurately recovered

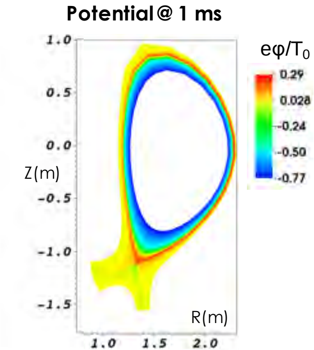


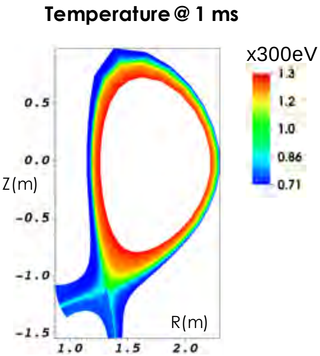
COGENT: Direct Kinetic Eulerian Edge Simulation

Provide future theory-based transport fluxes in SOL

- Kinetic cross-separatrix transport computed by COGENT
- Includes 2D potential and Fokker-Planck ion-ion collisions







AToM Use Cases

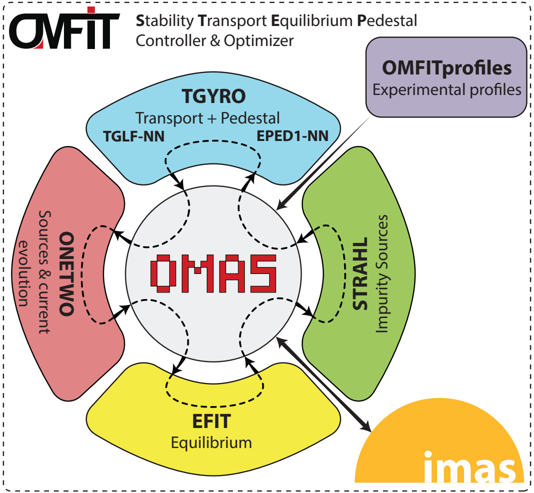
Entry point for collaboration with AToM (UCSD)

- Validation and scenario modeling will be organized about benchmark use cases
  - datasets describing **key plasma discharges** for component and workflow validation
  - effective way to benchmark models, track improvements, **assess performance**
- Each use case will include
  - Magnetic equilibria and profile data in accessible format
  - Repository of calculated quantities (code results)
  - Provenance documentation (shots/publications/models)
- **Ccandidate Use Cases**
  1. DIII-D L-mode shortfall, ITER baseline, steady-state discharges
  2. Alcator C-Mod LOC/SOC plasmas, EDA H-mode toroidal field scan
  3. ITER inductive, hybrid, and steady-state scenarios
  4. ARIES ACT-1/ACT-2 reactor scenarios

Key concept for AToM interaction with other SciDACs

Compliance with the ITER IMAS data model

https://gafusion.github.io/omas



- Transfer data between components using OMAS (python)
- API stores data in format compatible with **IMAS data model**
- Use storage systems other than native IMAS

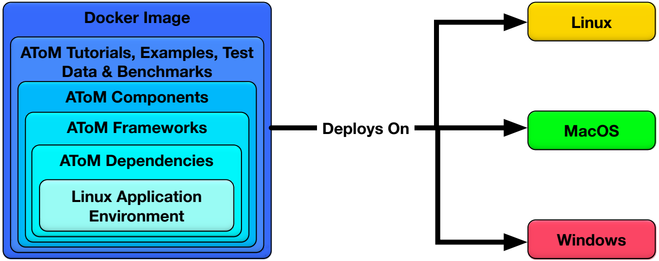
Compliance with the ITER IMAS data model

https://gafusion.github.io/omas

- IMAS is
  "a set of codes, an execution framework, a data schema, data storage infrastructure to support ITER plasma operations and research"
- We confirmed that IMAS has several functional **shortcomings**
  - issues with speed, stability, portability, useability
- **OMAS Solution**:
  - store data according to IMAS schema
  - do not use the IMAS infrastructure itself
  - facilitate data translation to/from IMAS schema
  - lightweight Python library

AToM Environment: Docker

Deploys without building for getting up and running quickly



- Common user environment across multiple platform
- Enables users on nontarget platform to run components locally
- Images composable from a common base
- OMFIT runtime environment currently available as Docker image

AToM Environment: Spack

AToM components installable from AToM Spack repository

- Spack manages installation of dependencies
- HPC-focused, facilitate component option experimentation
- Example usage:

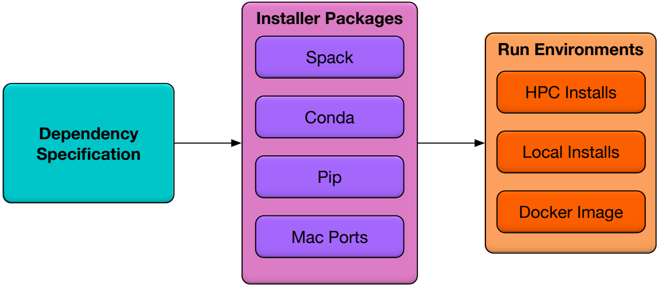
```
# List available AToM packages
$ spack list -t atom

# Install package using spack
$ spack install [package]

# Install AToM tier1 packages
$ spack install atom-tier1
```

AToM Environment: Dependency Specification

For AToM Components with multiple install methods



- Add new dependencies in a single location
- Generates recipes/specs/etc and builds installer packages
- Uploads packages to package manager servers and builds images