Towards near-real-time discontinuous Galerkin gyrokinetic (DG) modeling of fusion plasmas: collisions via super time stepping (STS)

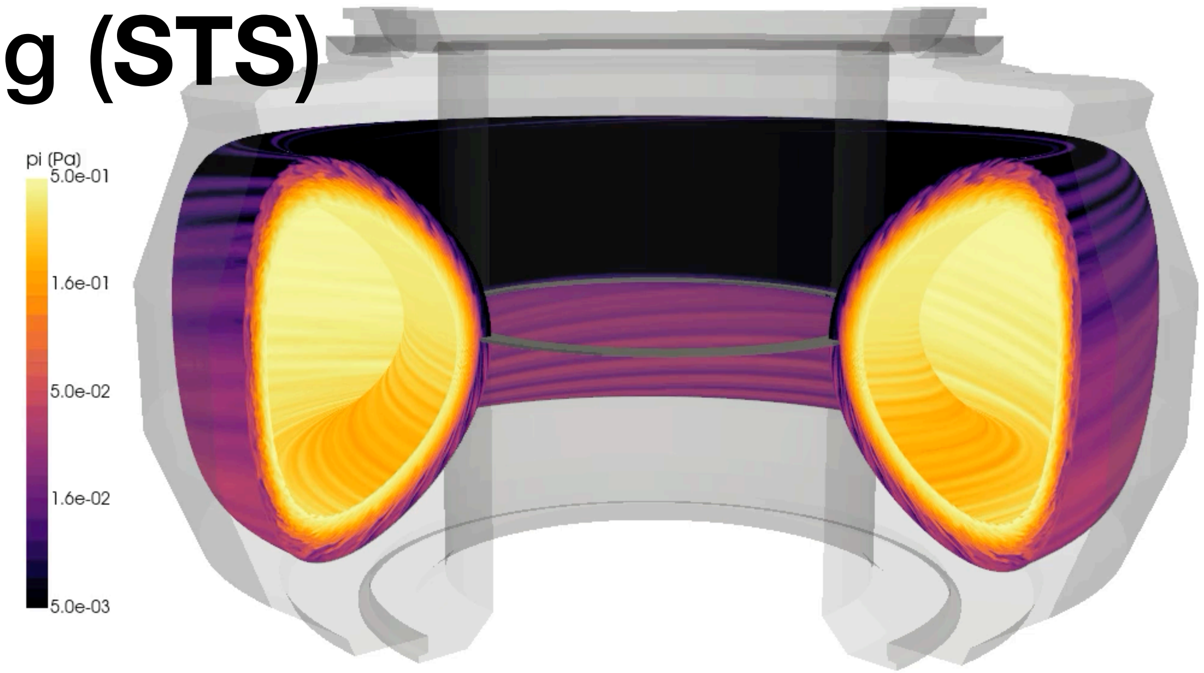
Manaure Francisquez[2]

[2] Princeton Plasma Physics Laboratory, mfrancis@pppl.gov

Mustafa Aggul[3,4]**, Sylvia Amihere**[3,4]**, Dan Reynolds**[3,4]

[3] University of Maryland Baltimore County.

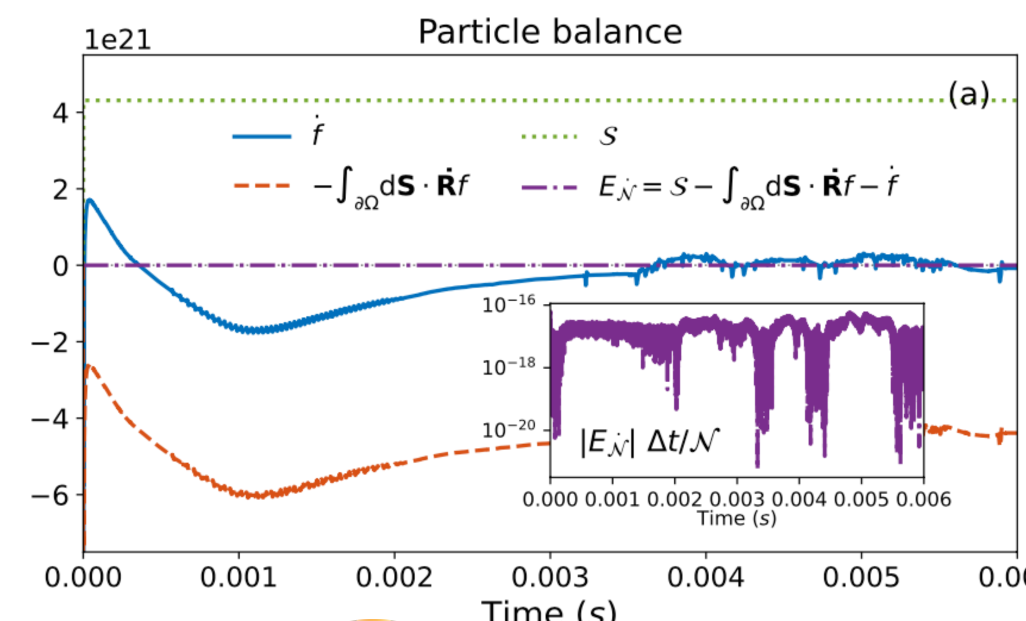
[4] Southern Methodist University.



Gkeyll

t=0.504 ms

5D simulation of DIII-D negative triangularity edge plasma (PPPL & General Atomics)



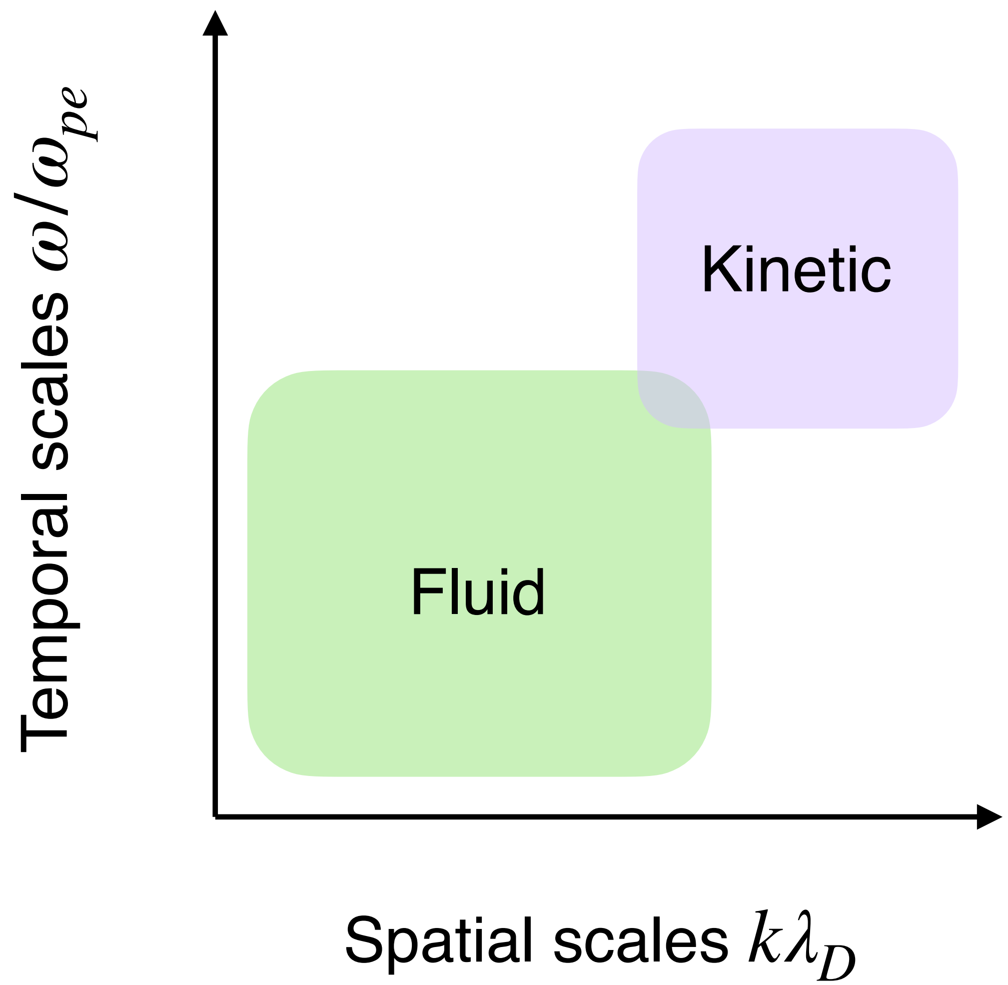
Other contributors: Ammar Hakim, Greg Hammett, Tess Bernard, Jimmy Juno, Jonathan Gorard, Maxwell Rosen, Grant Johnson, Akash Shukla, Dingyun Liu, Antoine Hoffmann, Jonathan Roeltgen.

[1] https://gkeyll.readthedocs.io/.

PRINCETON PLASMA PHYSICS LABORATORY

Plasmas have large spatio-temporal separation

⇒Requires different models



ω : fluctuation frequency
ω_{pe} : plasma frequency
k : fluctuation wave number
λ_D : Debye length

In Gkeyll[1] we strive to cover the entire range with a suite of models.

[1] https://gkeyll.readthedocs.io/.

PPPL PRINCETON PLASMA PHYSICS LABORATORY

Gkeyll models lab & space plasmas with fluids

Classical fluids[2]

$$\frac{\partial(m_s n_s)}{\partial t}+\frac{\partial(m_s n_s u_{s,i})}{\partial x_i}=0$$

$$\frac{\partial(m_s n_s u_{s,i})}{\partial t}+\frac{\partial \mathcal{P}_{s,ij}}{\partial x_j}=n_s q_s\left(E_i+\epsilon_{ijk}u_{s,j}B_k\right)$$

5-moment

$$\frac{\partial \mathcal{E}_s}{\partial t}+\frac{\partial}{\partial x_i}(p_s+\mathcal{E}_s)u_{s,i}=n_s q_s u_{s,i}E_i$$

10-moment

$$\frac{\partial \mathcal{P}_{s,ij}}{\partial t}+\frac{\partial \mathcal{Q}_{s,ijk}}{\partial x_k}=n_s q_s u_{[s,i}E_{j]}+\frac{q_s}{m_s}\epsilon_{[ikl}\mathcal{P}_{s,kj]}B_l$$

Relativistic fluids[3]

$$\frac{1}{\sqrt{-g}}\left(\partial_t\left(\sqrt{\gamma}\tau\right)+\partial_i\left\{\sqrt{-g}\left[\tau\left(v^i-\frac{\beta^i}{\alpha}\right)+pv^i\right]\right\}\right)=\alpha\left(T^{\mu t}\partial_\mu\alpha-T^{\mu\nu}\Gamma^t_{\nu\mu}\right),$$

$$\frac{1}{\sqrt{-g}}\left(\partial_t\left(\sqrt{\gamma}S_j\right)+\partial_i\left\{\sqrt{-g}\left[S_j\left(v^i-\frac{\beta^i}{\alpha}\right)+p\delta^i_j\right]\right\}\right)=T^{\mu\nu}\left(\partial_\mu g_{\nu j}-\Gamma^\sigma_{\nu\mu}g_{\sigma j}\right),$$

$$\frac{1}{\sqrt{-g}}\left(\partial_t\left(\sqrt{\gamma}D\right)+\partial_i\left\{\sqrt{-g}\left[D\left(v^i-\frac{\beta^i}{\alpha}\right)\right]\right\}\right)=0$$

$$\tau=\rho hW^2-p-\rho W,\qquad S_i=\rho hW^2 v_i,\qquad D=\rho W$$

$$v^i=\frac{u^i}{\alpha u^t}+\frac{\beta^i}{\alpha},\qquad W=\alpha u^t=\frac{1}{\sqrt{1-\gamma_{ij}v^i v^j}}$$

[2] J. Ng's PhD Thesis. Princeton University (2019).
[3] J. Gorard et al. arxiv/2410.02549 (2024), arxiv/2505.05299 (2025).

Gkeyll models lab & space plasmas with fluids & kinetics

Classical Vlasov-Maxwell[4]:

$$\frac{\partial f_s}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{v} f_s) + \nabla_{\mathbf{v}} \cdot \left(\frac{q_s}{m_s} [\mathbf{E} + \mathbf{v} \times \mathbf{B}] f_s \right) = C[f_s] + S_s$$

Relativistic Vlasov-Maxwell[5]:

$$\frac{\partial f_s}{\partial t} + \nabla \cdot \left(\frac{\mathbf{p}}{\gamma} f_s \right) + \nabla_{\mathbf{p}} \cdot \left(\frac{q_s}{m_s} \left[\mathbf{E} + \frac{\mathbf{p}}{\gamma} \times \mathbf{B} \right] f_s \right) = S_s$$

Gyrokinetic[6]:

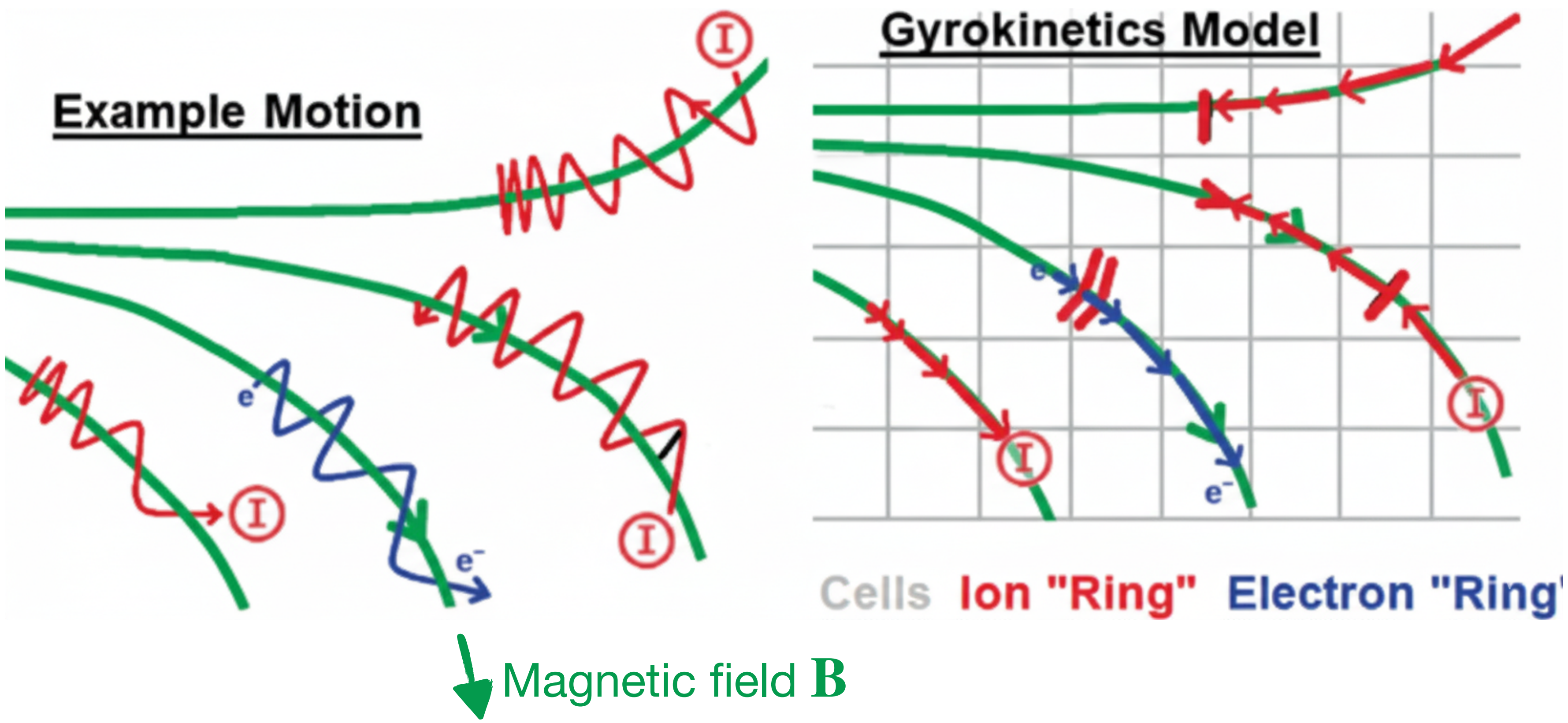
$$\frac{\partial (\mathcal{J} f_s)}{\partial t} + \nabla \cdot \mathcal{J} \dot{\mathbf{R}} f_s + \frac{\partial}{\partial v_\parallel} \left(\dot{v}_\parallel^H - \frac{q_s}{m_s} \frac{\partial A_\parallel}{\partial t} \right) \mathcal{J} f_s = \mathcal{J} C[f_s] + \mathcal{J} S_s$$

There's also a fluid-kinetic hybrid model in Gkeyll[7].

[4] J. Juno, et al. JCP 353, 110 (2018).
[5] S. Zheng, et al. arxiv/2509.13419 (2025).
[6] M. Francisquez, et al. arxiv/2505.10753.
[7] J. Juno, et al. JPP 91, E129 (2025).

Gyrokinetics: reduces dimensionality & time-scales.

Gyrokinetics is a reduced-kinetic model for magnetized plasmas which averages over gyromotion & orders out frequencies larger than the gyrofrequency (ω_c).



⇒ A 3D2V model, rather than the 3D3V of Vlasov.

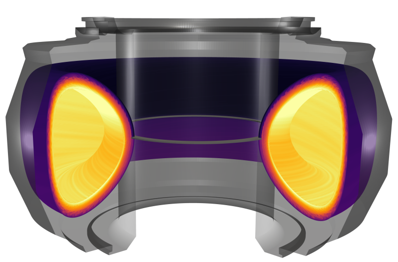
⇒ No need to resolve ω_c (take larger time steps, computationally cheaper)

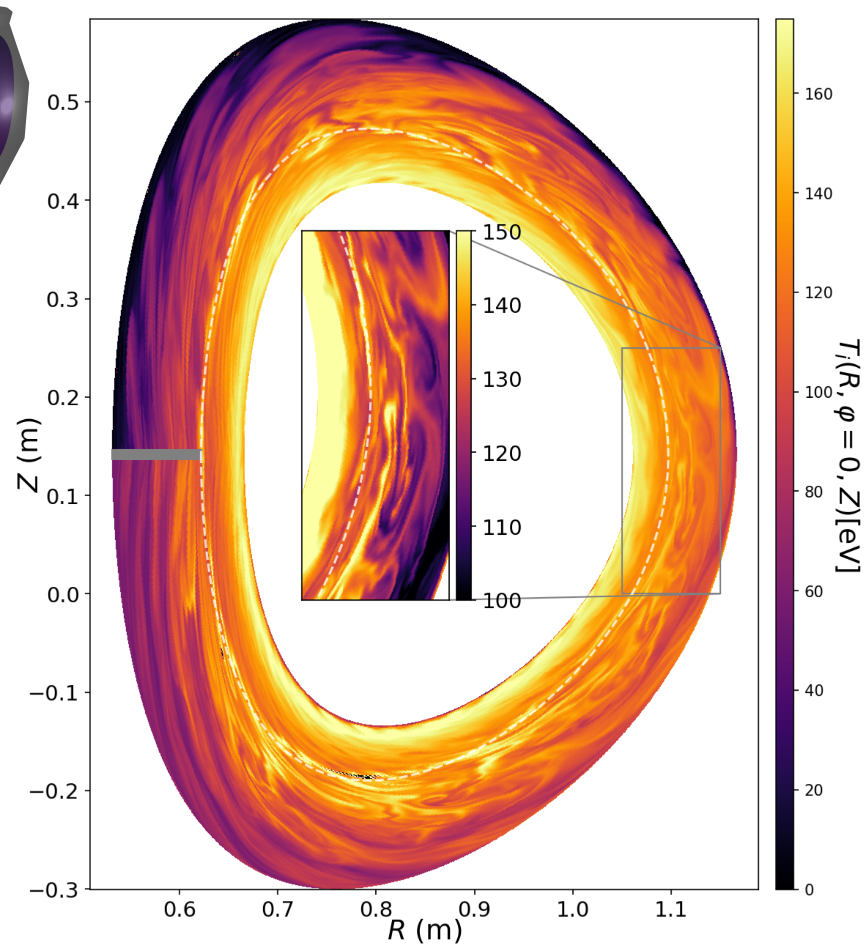
Gyrokinetics: towards predictive modeling of fusion plasmas

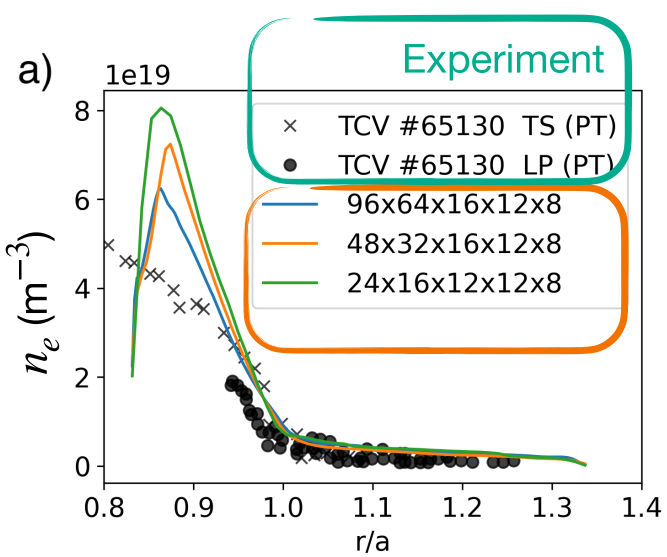
Evolve the distribution of a charged particle species, f_s, on a $(\mathbf{R}, v_\parallel, \mu)$ grid:

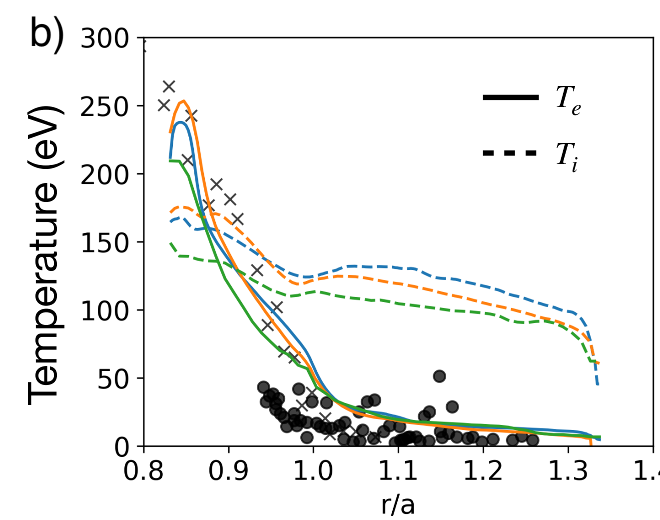
$$\frac{\partial(\mathcal{J}f_s)}{\partial t} + \underbrace{\nabla\cdot\mathcal{J}\dot{\mathbf{R}}f_s + \frac{\partial}{\partial v_\parallel}\left(\dot{v}_\parallel^H - \frac{q_s}{m_s}\frac{\partial A_\parallel}{\partial t}\right)\mathcal{J}f_s}_{\text{advection}} = \underbrace{\mathcal{J}\mathcal{C}_s}_{\text{collisions}} + \underbrace{\mathcal{J}\mathcal{D}_s}_{\text{diffusion}} + \underbrace{\mathcal{J}\mathcal{S}_{Qs}}_{\text{heating}} + \underbrace{\mathcal{J}\mathcal{R}_s}_{\text{reactions}}$$

- Showed reasonable agreement for modeling the TCV tokamak in Switzerland only employing **3 inputs**: magnetic field, heating power and particle count[8]:









[8] A. Hoffmann, et al. (arXiv:2510.11874)

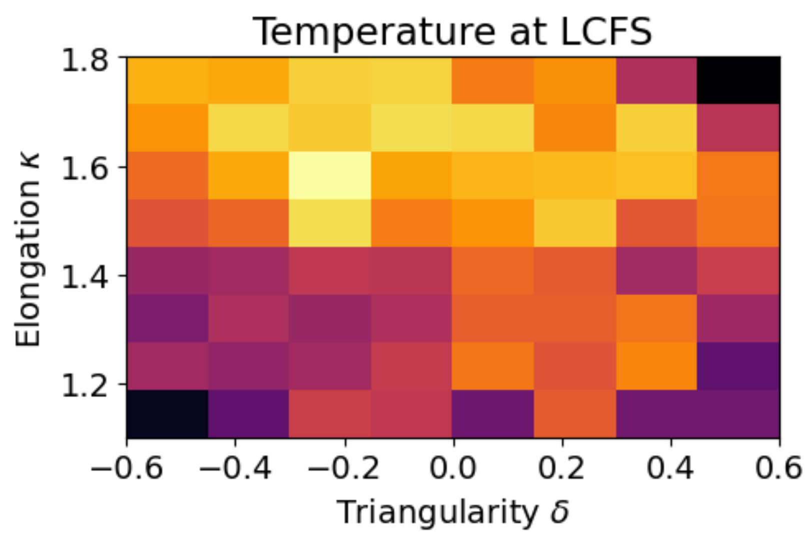
PPPL PRINCETON PLASMA PHYSICS LABORATORY

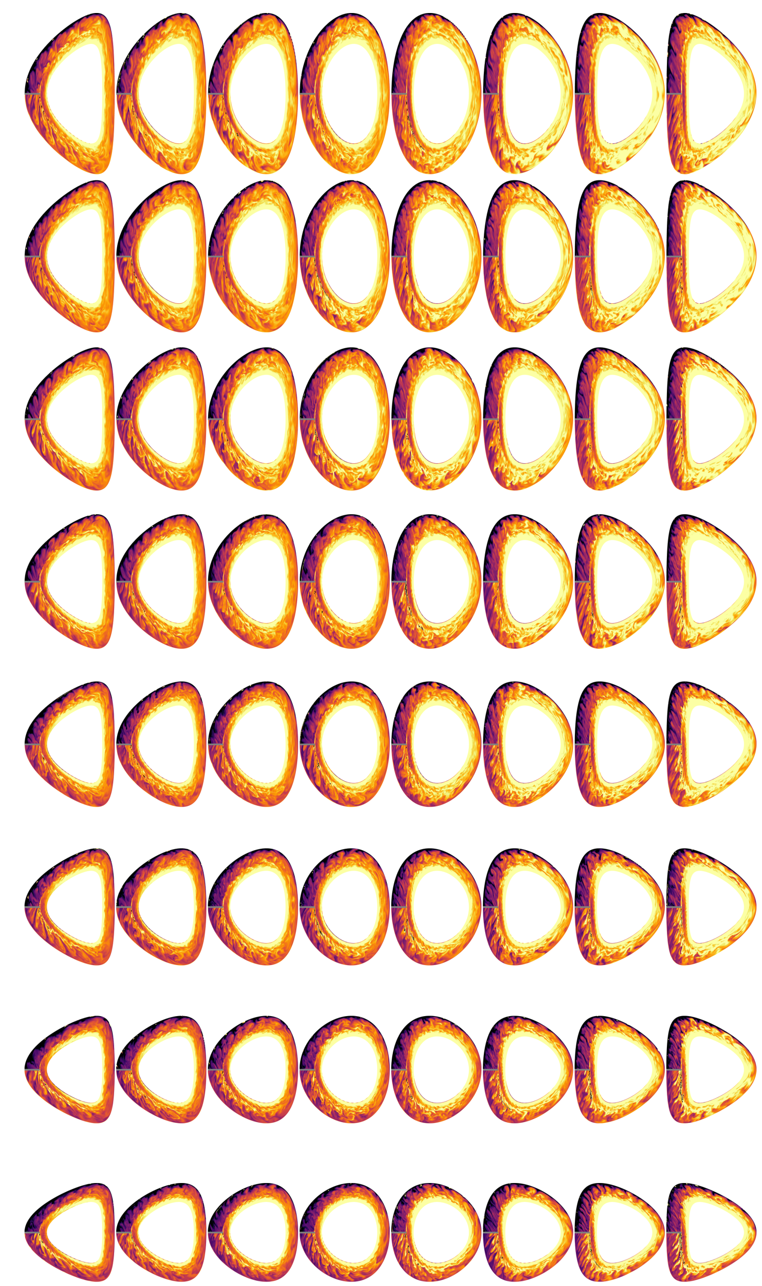
Gyrokinetics: towards predictive & near-real time modeling of fusion plasmas

Near-real time (in between shots or overnight) gyrokinetics may be possible, thanks to:

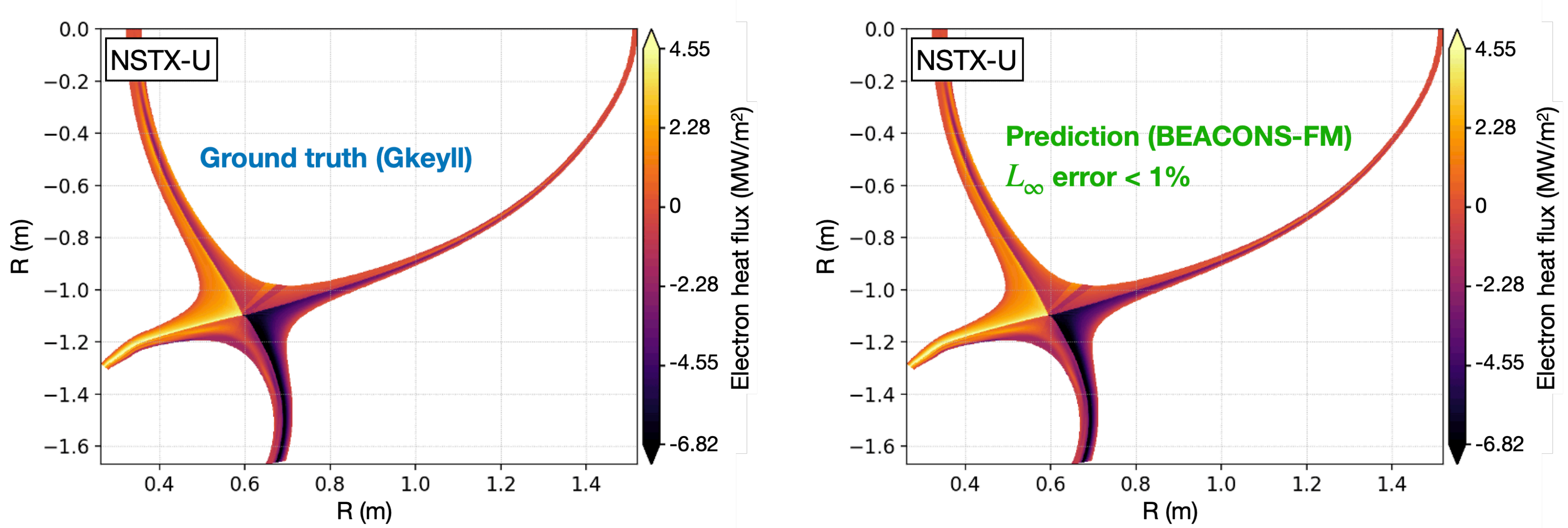
1. Novel DG algorithms robust at any resolution + GPUs.

Example:
scan plasma shape (elongation & triangularity), running 64 shapes to steady state in O(48 h) on a few GPUs (each)[9].





2. Surrogates & foundation models (FM)[10].

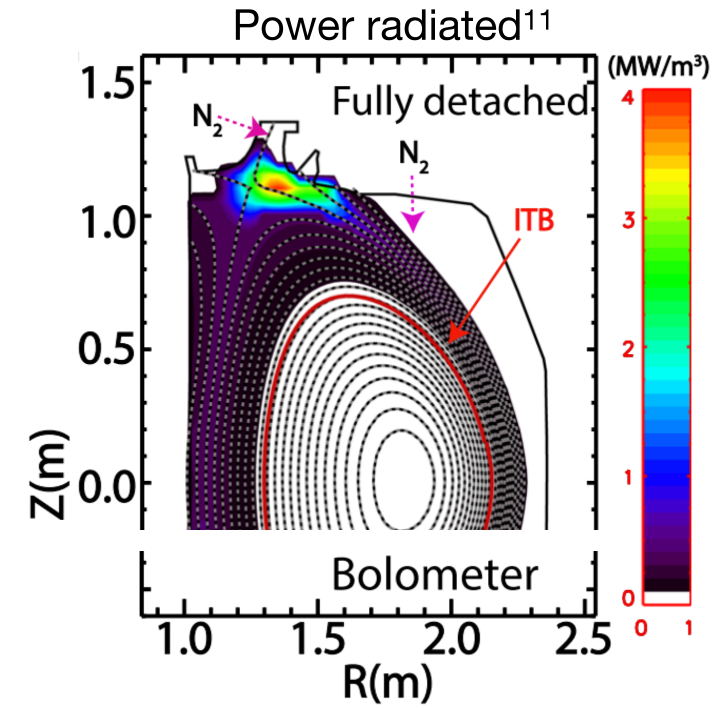


[9] A. Hoffmann, et al. (in preparation).

[10] J. Gorard, et al. (in preparation).

PPPL PRINCETON PLASMA PHYSICS LABORATORY

Collisional regimes are still challenging



- Some regimes inject gas to dissipate heat near the wall These regions have very high collision frequency (ν_{sr}).

collisions

$$\frac{\partial(\mathcal{J} f_s)}{\partial t} + \nabla\cdot\mathcal{J}\dot{\mathbf{R}} f_s + \frac{\partial}{\partial v_\parallel}\left(\dot{v}_\parallel^H - \frac{q_s}{m_s}\frac{\partial A_\parallel}{\partial t}\right)\mathcal{J} f_s = \mathcal{J}\mathcal{C}_s + \dots$$

- We use a Dougherty collision operator[12]:

$$\mathcal{C}_s^{\text{elastic}} = \sum_r \nu_{sr}\frac{\partial}{\partial v_\parallel}\left[\left(v_\parallel - u_{\parallel sr}\right) + v_{t,sr}^2\frac{\partial}{\partial v_\parallel}\right] f_s + \dots$$

$$u_\parallel = u_\parallel(\mathbf{R}; f_s)$$
$$v_{t,sr}^2 = v_{t,sr}^2(\mathbf{R}; f_s, f_r)$$

- Gkeyll time integrates this term **explicitly** with a 3-stage 3rd order SSP-RK.
- Time step scales like $\Delta t \sim \min\left[\frac{1}{\nu_{sr}}\left(\frac{\Delta v_\parallel}{v_\parallel - u_\parallel} + \frac{\Delta v_\parallel^2}{v_{t,sr}^2}\right)\right]$
- Makes Δt impractically small.

Can we accelerate these simulations with novel time integration algorithms (**STS**) in SUNDIALS?

[11] L. Wang, et al. Nat. Comm. 12, 1365 (2021).

[12] M. Francisquez, et al. NF 60 (2020).

STS in a nutshell

- **Super-time stepping (STS)**:

$$
\begin{aligned}
& f^n = f(t^n), \qquad \mathcal{L}_{diff}(f^n) = \partial_t f^n \Big|_{diff}, \\
& - - - - - - - - - - - - - - - \\
& \quad \text{stage } 0: \\
& \qquad\qquad f^1 = f^n + c_0 \; \Delta t \; \mathcal{L}(f^n), \\
& - - - - - - - - - - - - - - - \\
& \quad \text{stage k}: \\
& \qquad f^{k+1} = d_k \; f^k + e_k \; \Delta t \; \mathcal{L}_{diff}(f^k) + f_k \; \Delta t \; \mathcal{L}_{diff}(f^{k-1}),
\end{aligned}
$$

  - Number of stages (s) is determined by the maximum eigenvalue of the $\mathcal{L}_{diff}(f^n)$ operator.

Gkeyll - SUNDIALS coupling (step 1)

- Gkeyll stores each $f_s(\mathbf{R}, v_{\|}, \mu)$ in a separate `gkyl_array` and has functions (methods) to do arithmetics and linear algebra with & reductions of `gkyl_array`'s.
- Need to wrap a `gkyl_array`'s data in SUNDIALS' `N_Vector` data structure.

Followed a two step process:

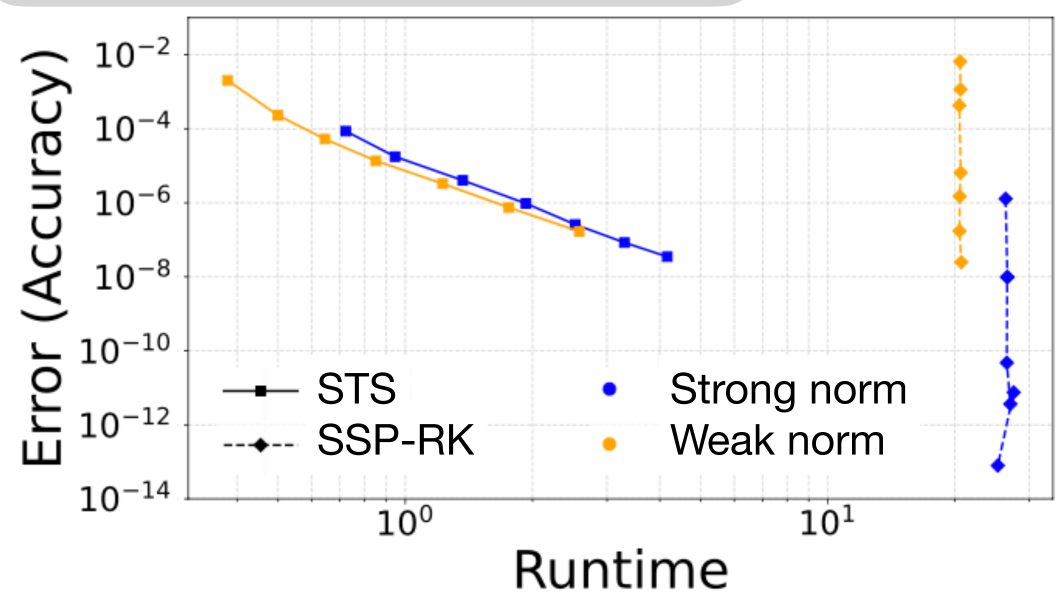
1. Created a test program solving $\partial_t f = \partial_x\left[\nu(x)\partial_x f\right]$ for **one** species/`gkyl_array` using Gkeyll's infrastructure & DG methods (**M. Aggul**)[13].
   - Created an `N_Vector` wrapper of our `gkyl_array` & methods.

$f(\mathbf{R}, v_{\|}, \mu)$

```
gkyl_array
  void *data;
  ...
```

```
N_Vector
  void *content;
  <gkyl_array methods>
  ...
```

   - Used this program to test Super-Time-Stepping (**STS**) methods with DG[15].

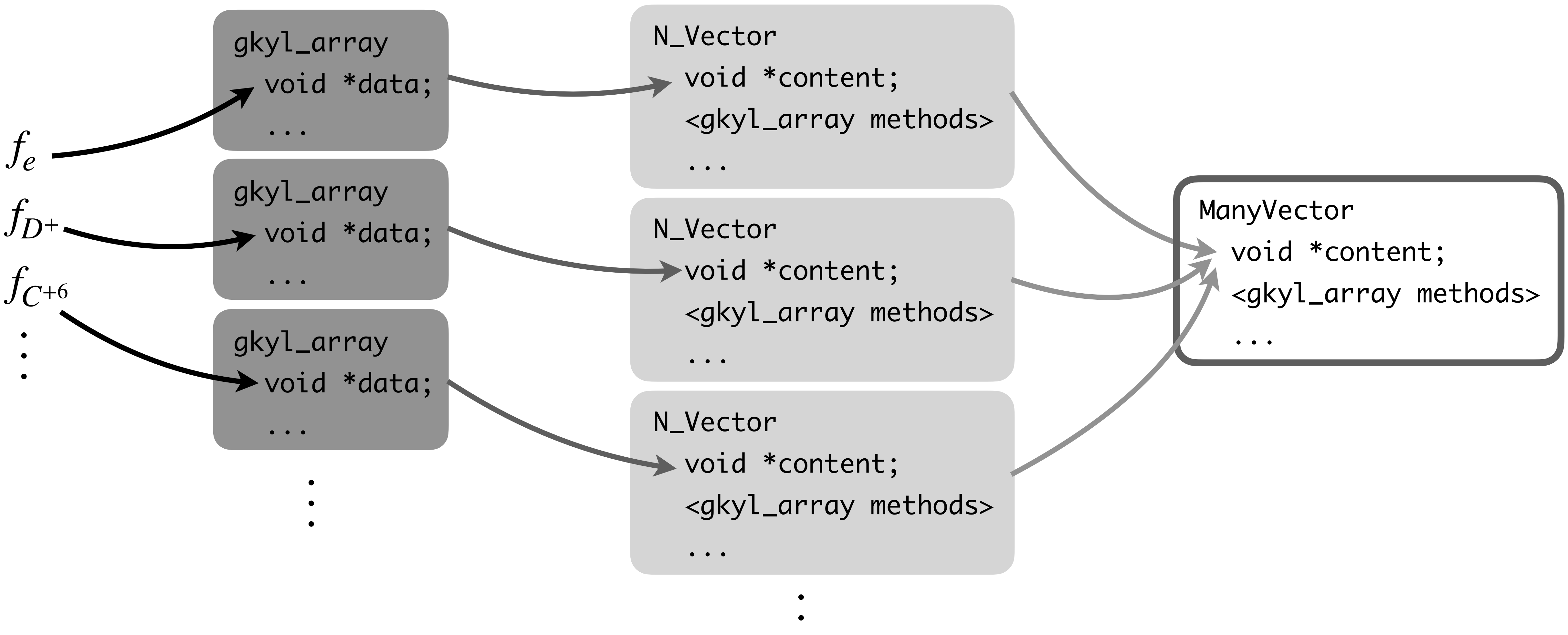


[13] M. Aggul, et al. https://arxiv.org/abs/2601.14508 (submitted to CAMWA).

Gkeyll - SUNDIALS coupling (step 2)

- Gkeyll's more complicated. Some of the complexities/requirements of the coupling:
  - Needs to work for any number of f_s.
  - Needs to work for all models (i.e. fluids, Vlasov, gyrokinetics).

2. Wrap each f_s 's `gkyl_array` in an `N_Vector` & wrap all `N_Vectors` in a `ManyVector`:

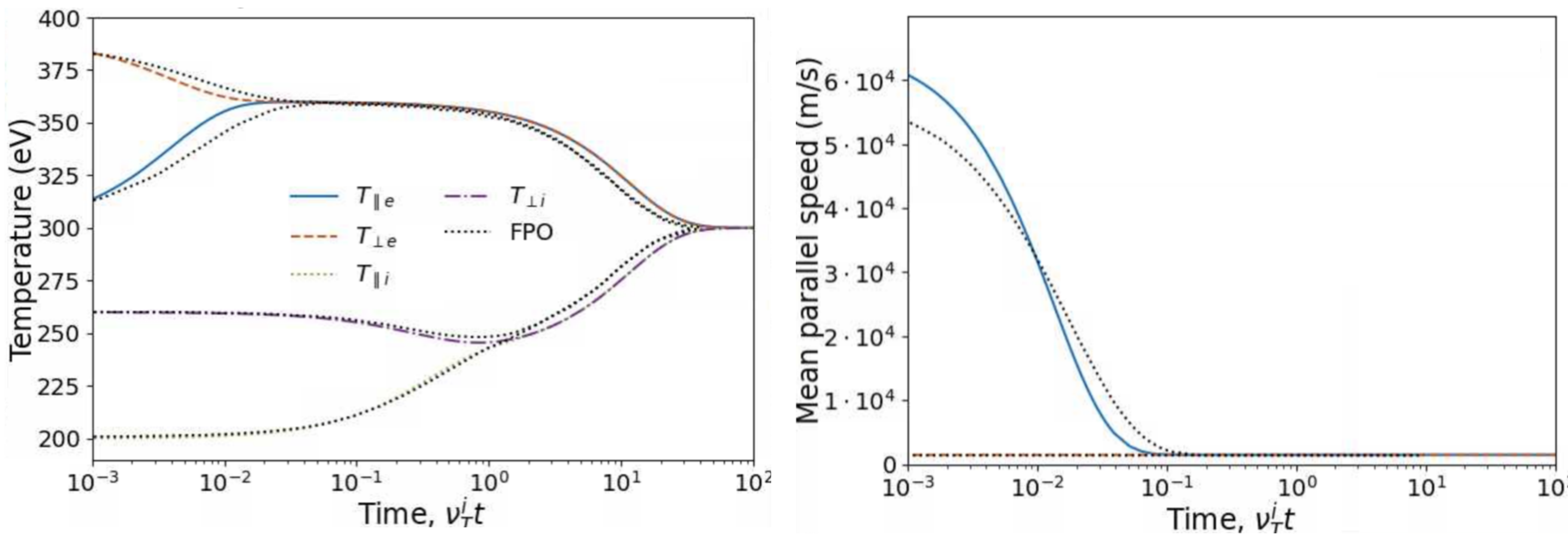


Gkeyll - SUNDIALS use case: collisional relaxation

- Can now step Dougherty operators w/ SUNDIAL's STS.
- Input file only has trivially small addition. →

```
.sundials_stepper = {
  .relative_tolerance = 1e-7,
  .absolute_tolerance = 1e-12,
  .method = GKYL_SUNDIALS_LSRK_METHOD_RKL_2,
},
```

- A typical test of this operator is collisional relaxations between electrons & ions[14]:



- On a single core of a 2023 Apple M3 MacBookPro this took:
  - 29 min with Gkeyll using 3-stage 3rd order SSP-RK.
  - 32 sec with Gkeyll-SUNDIALS using STS (RKL2).

54X speedup!

[14] M. Francisquez, et al. JPP 88, 905880303 (2022).
Also R. Hager, et al JCP (~2018), P. Ulbl, et al. (2022).

Gkeyll - SUNDIALS next steps

- Use it in production calculations.
- Characterize performance with parameter regime.
- Optimize (e.g. memory storage is high)
- Try new methods for:
  - Evaluating the maximum eigenvalue of the Dougherty operator.
  - Mixing SSP-RK for collisionless terms and STS for collisions.