A Moment-Based Kinetic Model for Efficient Numerical Implementation

Rogério Jorge[1]

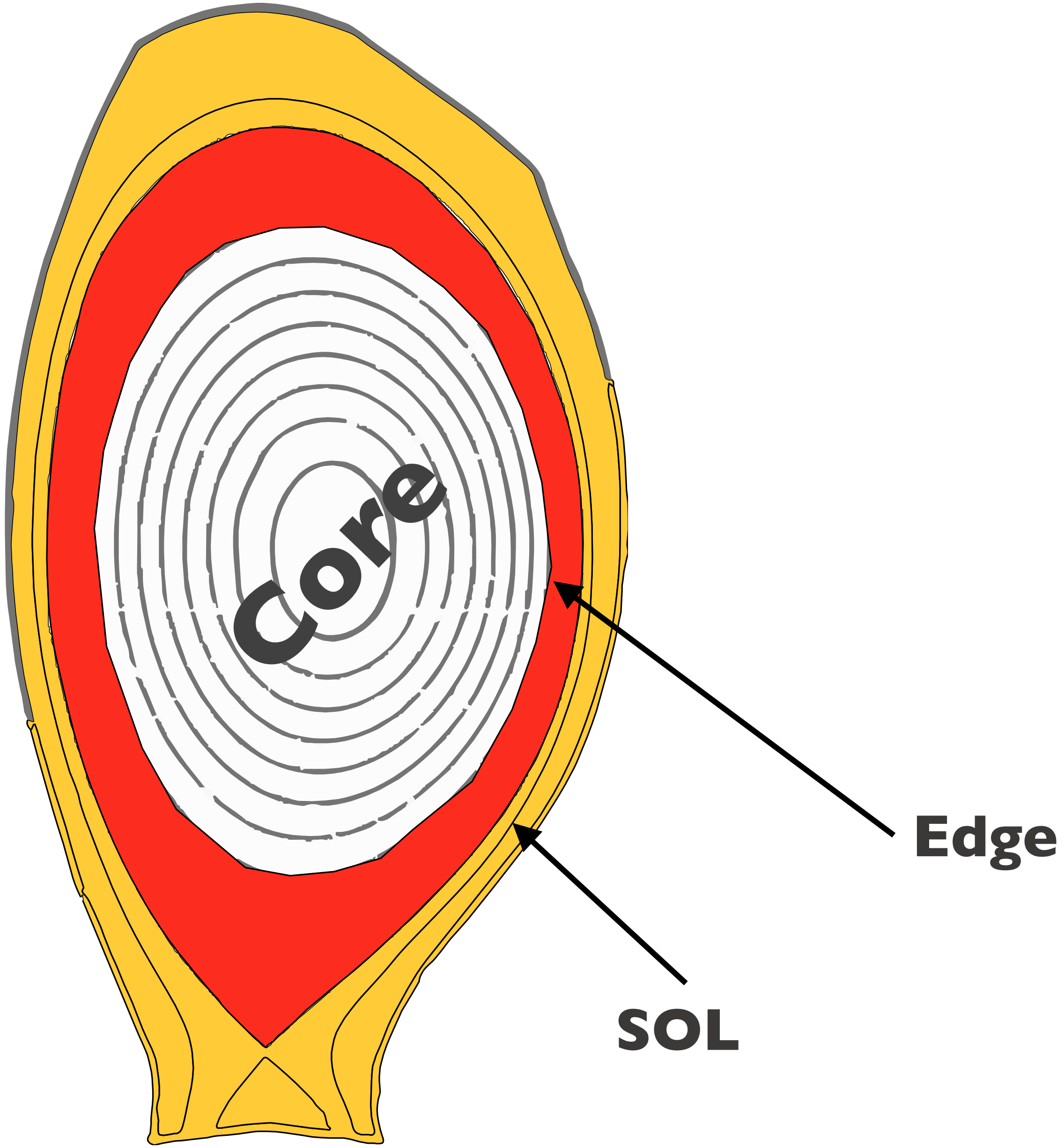
B. Frei[2], P. Ricci[2], N. F. Loureiro[3]

[1]University of Maryland, [2]EPFL, [3]MIT

UNIVERSITY OF MARYLAND

June 13, 2019 – PASC conference

Motivation: Simulate The Plasma Periphery (Edge + SOL)



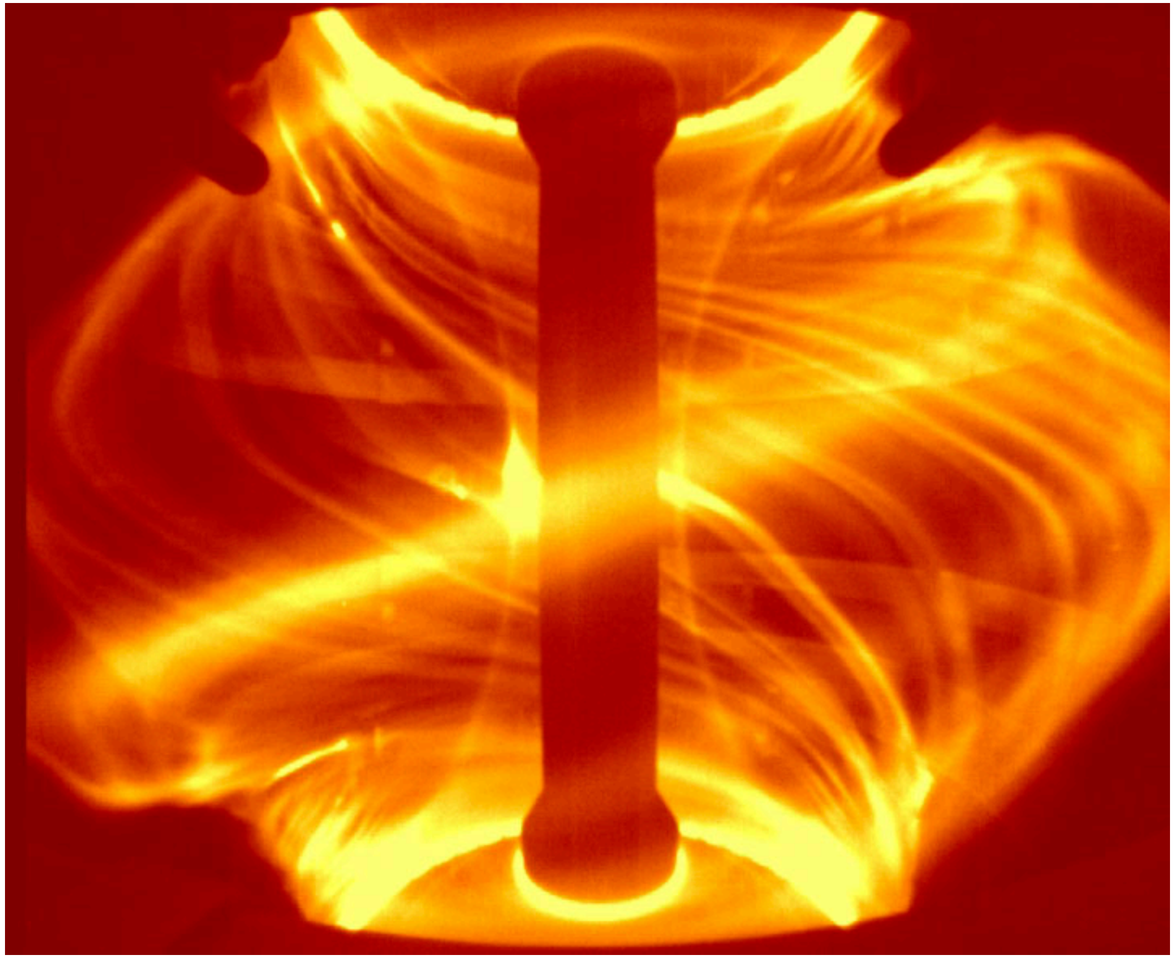
- ❑ Core boundary conditions
- ❑ Heat exhaust
- ❑ Plasma fueling and ashes removal
- ❑ Impurity control

Properties of Periphery Turbulence

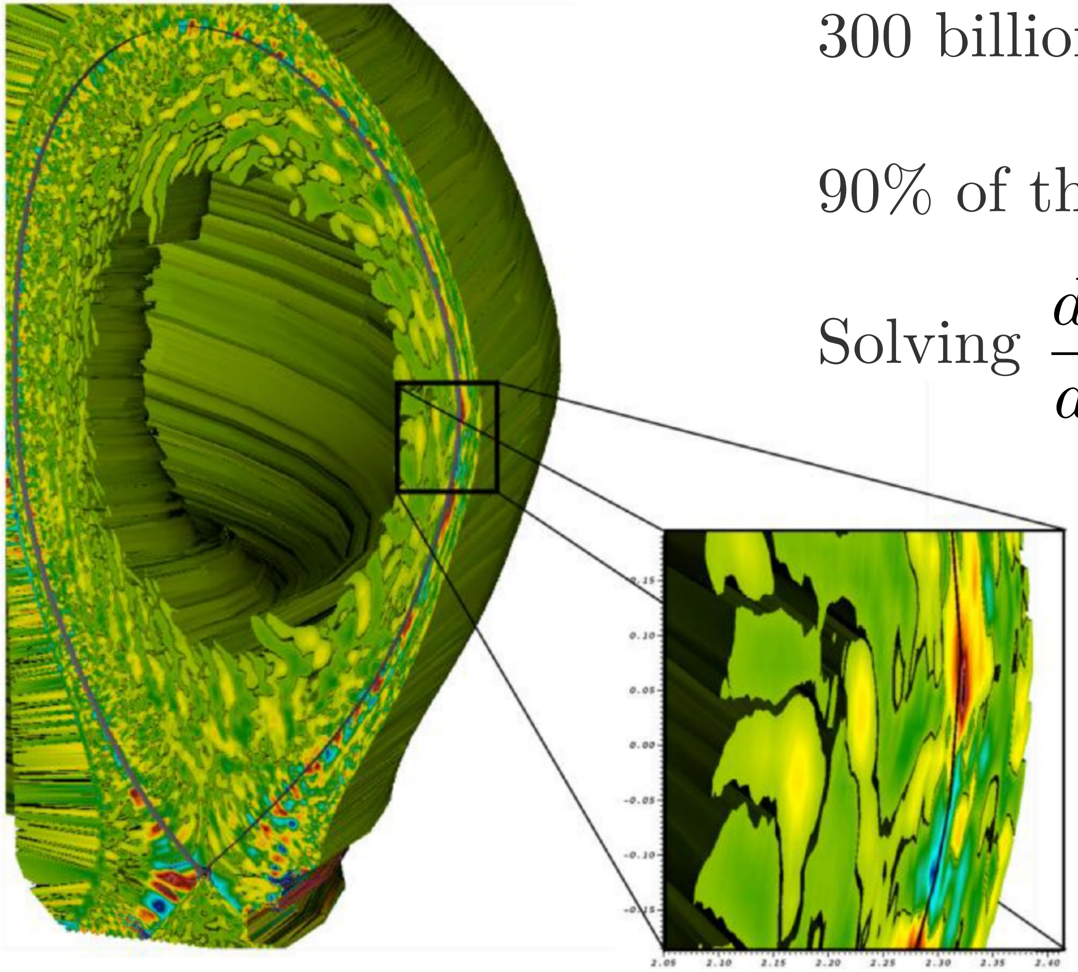
- Low-frequency
- Large and Small Scale Fluctuations
- Wide range of temperatures and densities

↓

Arbitrary Collisionality Gyrokinetic Model



Kinetic Simulations



300 billion particles

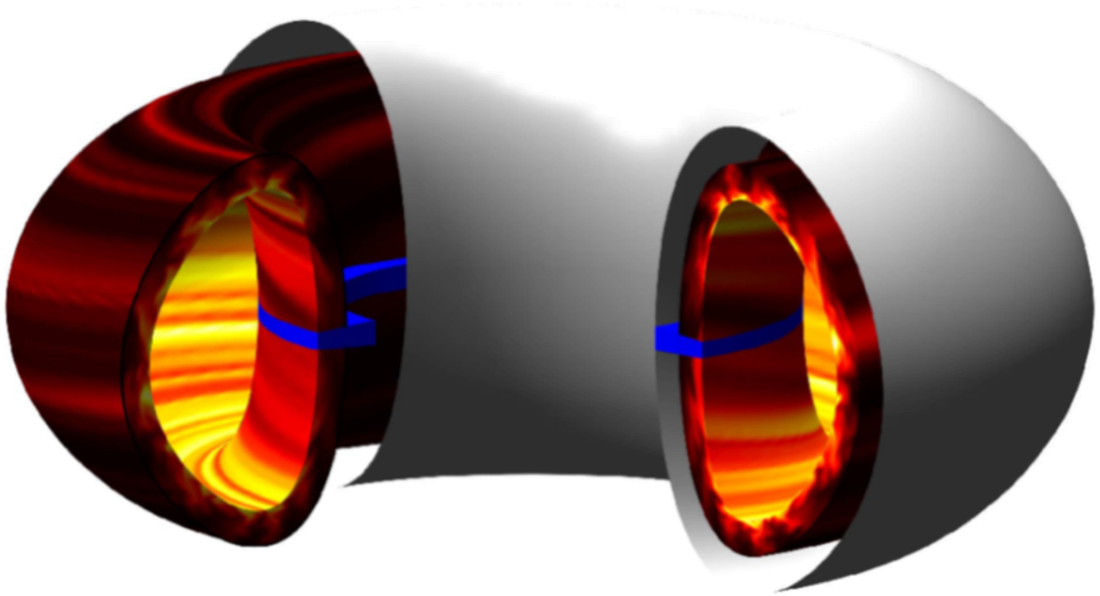
90% of the 27 petaflop Titan supercomputer

Solving $\frac{df}{dt} = \ldots$

Extremely Expensive

XGC1 code
Chang et. al., Nuclear Fusion **57** (2017)

Fluid Simulations



GBS code, Swiss Plasma Center, EPFL
Ricci et. al., Plasma Phys. Controlled Fusion **54** (2012)

Considerably less expensive

SOL Confinement time scales

Solving $\frac{dn}{dt} = \ldots$

Assume High Collisionality

Our Model

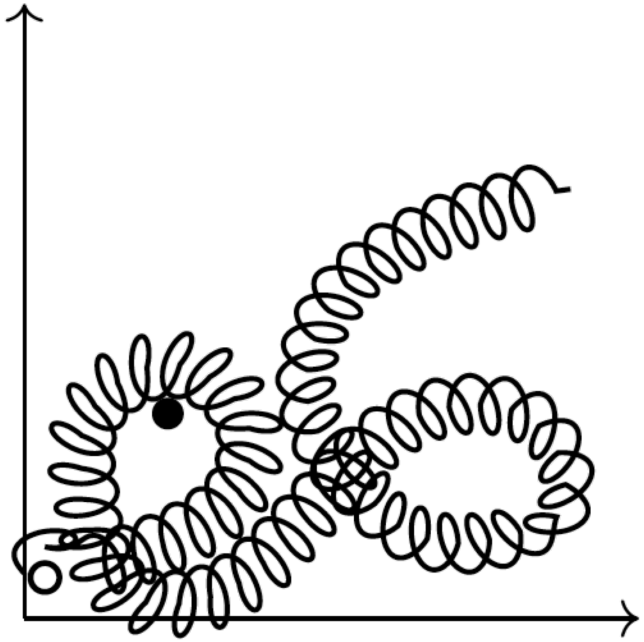
Retain

- Both large scale and small scale fluctuations (full-F)
- Second order
- Full Coulomb collisions
- Numerical efficiency

How we proceed

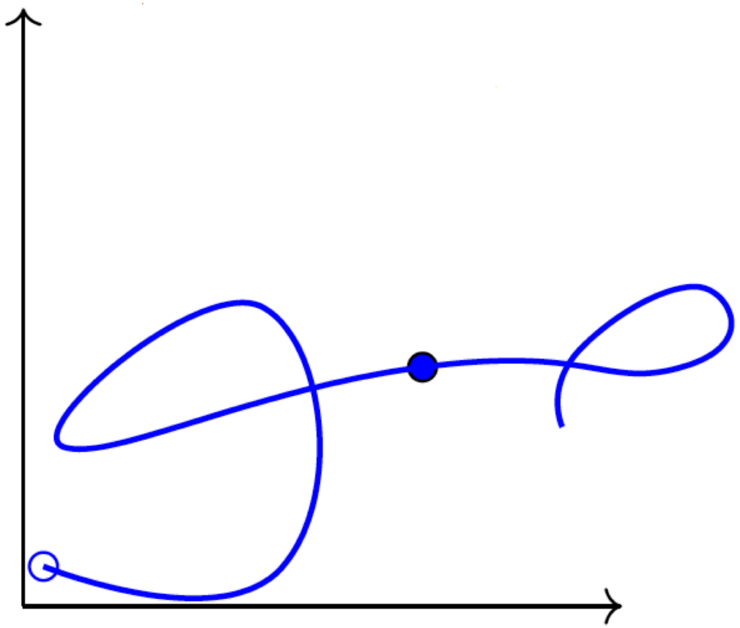
Single Particle Dynamics → Gyrokinetic Theory → Moment Hierarchy

Single Particle Dynamics – Hamiltonian Perturbation Theory



2-Step Phase-Space

Reduction



Particle Lagrangian

$L(\mathbf{x}, \mathbf{v})$

Lagrangian

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

(cyclic coordinate θ, conserved μ)

Compute equations of motion

$(\dot{\boldsymbol{R}}, \dot{v}_{\|})$

From Single Particle to Particle Distribution

Gyrokinetic Equation $$\frac{\partial F}{\partial t} + \dot{\mathbf{R}} \cdot \nabla F + \dot{v}_{\|} \frac{\partial F}{\partial v_{\|}} = \langle C(F) \rangle$$

Challenges

- 5-D + time
- Full Coulomb Collisions
- Coupling to Maxwell's equations (integro-differential system)

$$\nabla \cdot \mathbf{E} = \int \delta(\mathbf{R} + \rho - \mathbf{x}) F d\mathbf{R} dv_{\|} d\mu d\theta$$

These challenges can be successfully approached by using a moment hierarchy

Our Goal – Turn Gyrokinetic Eq. Into a Hierarchy of Fluid-Like Eqs.

Expand GK Equation into a Set of 3D

Moment Hierarchy Equations

$$\frac{dn}{dt} = \ldots$$

$$\frac{dv}{dt} = \ldots$$

$$\frac{dT}{dt} = \ldots$$

$$\ldots$$

Retain necessary kinetic effects and no more

Advantages of a Moment Hierarchy Model

Set of fluid-like equations with reasonable computational cost

Maxwell's equations are converted to a simple sum of moments

Tune the number of moments according to the desired level of accuracy (function of collisionality)

Reduce to the fluid model at high collisionality

3 Steps To Build a Moment Hierarchy Model

1. Choose an orthogonal polynomial basis for F

$$F = F_M \sum_{p,j} N^{pj}(\mathbf{R}) H_p(v_\parallel) L_j(\mu)$$

2. Project kinetic equation onto basis

$$\int (\text{GK Eq.})\, H_p L_j dv_\parallel d\mu$$

3. Obtain evolution equation for the basis coefficients

$$\frac{\partial N^{pj}}{\partial t} = \ldots$$

From Gyrokinetic Equation to Moment Hierarchy

$$\int (\text{GK Eq.})\, H_p L_j \, dv_{\|} \, d\mu$$

Spatial evolution of Moments + Fields

Fluid Operator (density, velocity, temperature)

$$\frac{\partial N^{pj}}{\partial t} + \nabla \cdot \dot{\mathbf{R}}^{pj} - \frac{\sqrt{2p}}{v_{th}} \dot{v}_{\|}^{p-1j} + \mathcal{F}^{pj} = C^{pj}$$

Time Evolution

Forces included at p>0

Collisions

Example – 1D Linear Gyrokinetic Moment Hierarchy

Phase Mixing
(coupling with other moments)

Collisions

$$\boxed{\frac{\partial N^{pj}}{\partial t}} + \boxed{\frac{1}{2}\frac{\partial N^{p+1j}}{\partial z} + p\frac{\partial N^{p-1j}}{\partial z}} = \boxed{2\delta_{p,1}K_j(k_\perp \rho)E_\parallel} + \boxed{C^{pj}}$$

Time Evolution

Electric Field Drive

Example – 1D Linear Gyrokinetic Moment Hierarchy

Phase Mixing
(coupling with other moments)

Collisions

$$\frac{\partial N^{pj}}{\partial t} + \frac{1}{2}\frac{\partial N^{p+1j}}{\partial z} + p\frac{\partial N^{p-1j}}{\partial z} = 2\delta_{p,1}K_j(k_\perp \rho)E_\parallel + C^{pj}$$

Time Evolution

Electric Field Drive

Semi-Collisional Closure $\left(N^{p+1j} \ll N^{pj}\right)$

$$p\frac{\partial N^{p-1j}}{\partial z} \sim C^{pj} \sim f(p,j)N^{pj} \longrightarrow N^{pj} \sim \frac{p}{f(p,j)} k_\parallel \lambda_{mfp} N^{p-1j}$$

$$\left(f(p,j) \sim p^{3/2}\right)$$

Projection of the Full Coulomb Collision Operator

$$C^{pj} = \int \langle C(F) \rangle H_p L_j dv_{||} d\mu$$

with $C(F) = \frac{\partial}{\partial \mathbf{v}} \cdot [\mathbf{H}(F)F] + \frac{\partial^2}{\partial \mathbf{v} \partial \mathbf{v}} : [\mathbf{G}(F)F]$

- Bilinear
- Tensorial Nature
- Gyroaveraging Operation
- No parallel/perpendicular velocity symmetries

$$H(F) \sim \int \frac{F(\mathbf{v}')}{|\mathbf{v} - \mathbf{v}'|} d\mathbf{v}'$$

Not immediate…

Collision Operator - Spherical Harmonic Decomposition

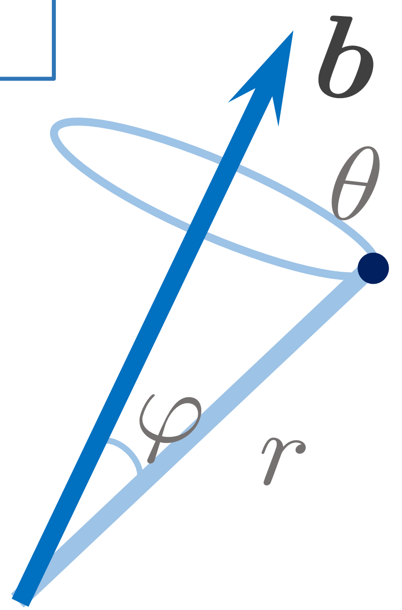
Expand the Rosenbluth potentials

$$H(F) \sim \int \frac{F(\mathbf{v}')}{|\mathbf{v}-\mathbf{v}'|} d\mathbf{v}'$$

Electrostatic potential of a charge distribution F in velocity space

in a Taylor series

$$\frac{1}{|\mathbf{v}-\mathbf{v}'|} = \begin{cases} \sum_l \frac{(-\mathbf{v}')^l}{l!} \cdot \partial^l_{\mathbf{v}} \left(\frac{1}{v}\right), & v' \leq v \\ \sum_l \frac{(-\mathbf{v})^l}{l!} \cdot \partial^l_{\mathbf{v}'} \left(\frac{1}{v'}\right) & v < v' \end{cases}$$



with spherical harmonics as coefficients

$$(-1)^l v^{l+1} \left(\frac{\partial}{\partial \mathbf{v}}\right)^l \frac{1}{v} \sim \sum_{m=-l}^{l} Y_{lm}(\varphi, \theta) \hat{\mathbf{e}}^{lm}$$

Spherical Harmonics

Gyroangle

Basis Tensors $\hat{\mathbf{e}}^{lm}$

Collision Operator – Expansion in Spherical Harmonics

$$C[F] \sim C[Y_{lm}] \sim Y_{lm}$$

Gyroaveraging Procedure

$$\langle Y_{lm}(\varphi, \theta) \rangle \longrightarrow T_{lm}^{pj} \; H_p(v_\parallel) L_j(\mu)$$

$C^{pj} = T_{lm}^{pj}$ x Sum of moments N^{pj}

Towards a Numerical Implementation

Numerical and theoretical investigation of linear modes

GK non-linear model reduced to the drift-kinetic linear limit

$$\frac{\partial N^{pj}}{\partial t} = \sum_{s,t} \boxed{D^{pj}_{st}} N^{st}$$

↓

MoliDK Code (MoliGK under development)

Numerical Model - Workflow

- ❑ Compute collisional coefficients C_{pj}
- ❑ Build coefficient matrix D^{pj}_{st} from moment-hierarchy equation
- ❑ Look for eigenvalues of D^{pj}_{st}
- ❑ Solve first-order autonomous ODE system $\dfrac{\partial N^{pj}}{\partial t} = \sum_{s,t} D^{pj}_{st} N^{st}$
- ❑ Compare with fluid/collisionless result

Numerical Model - Workflow

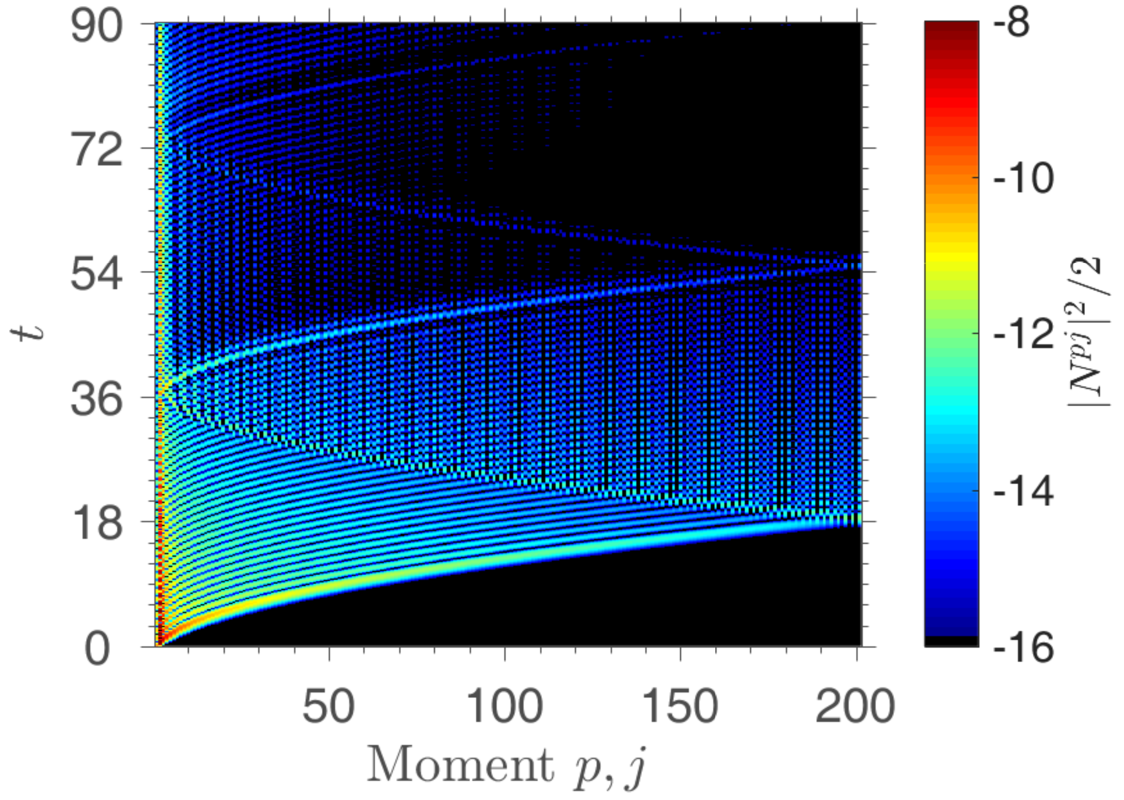
main

```
#!/bin/bash
# version 0.1 - R. Jorge SPC/EPFL November 2017
# version 1.0 - September 2018
#======START================
 echo "===================MOLIDK====================="
 echo "Moment Linear Drift-Kinetic Solver with Coulomb Collisions"
#======INPUT PARAMETERS=====
## Number of moments
collI=-2;       #collI=-2 -> Dougherty, -1 -> Coulomb, 0 -> Lenard-Bernstein, other ->
hyperviscosity exponent
pmaxe=20;       #parallel electron moments
jmaxe=1;        #perpendicular electron moments
pmaxi=20;       #parallel ion moments
jmaxi=1;        #perpendicular ion moments
## Collisional Coulomb sum bounds
lmaxx=15;       #upper bound collision operator first sum first species
kmaxx=15;       #upper bound collision operator second sum first species
nmaxx=lmaxx;    #upper bound collision operator first sum second species
qmaxx=kmaxx;    #upper bound collision operator second sum second species
## Physical Parameters
tau=0.01;        #Ti/Te
nu=12.3;#61.5;#14.5; #0.18;             #electron/ion collision frequency
mu=0.023;               #sqrt(m_e/m_i)
kpar=1.0;               #normalized parallel wave number to the major radius
kperp=0.1;              #normalized perpendicular wave number to the Larmor radius
alphaD=0.0;             #(k*Debye length)^2
Rn=10;          #Major Radius / Background density gradient length
RTe=0;                  #Major Radius * normalized kperp / Background electron temperature
gradient length
RTi=0;                  #Major Radius * normalized kperp / Background ion temperature
```

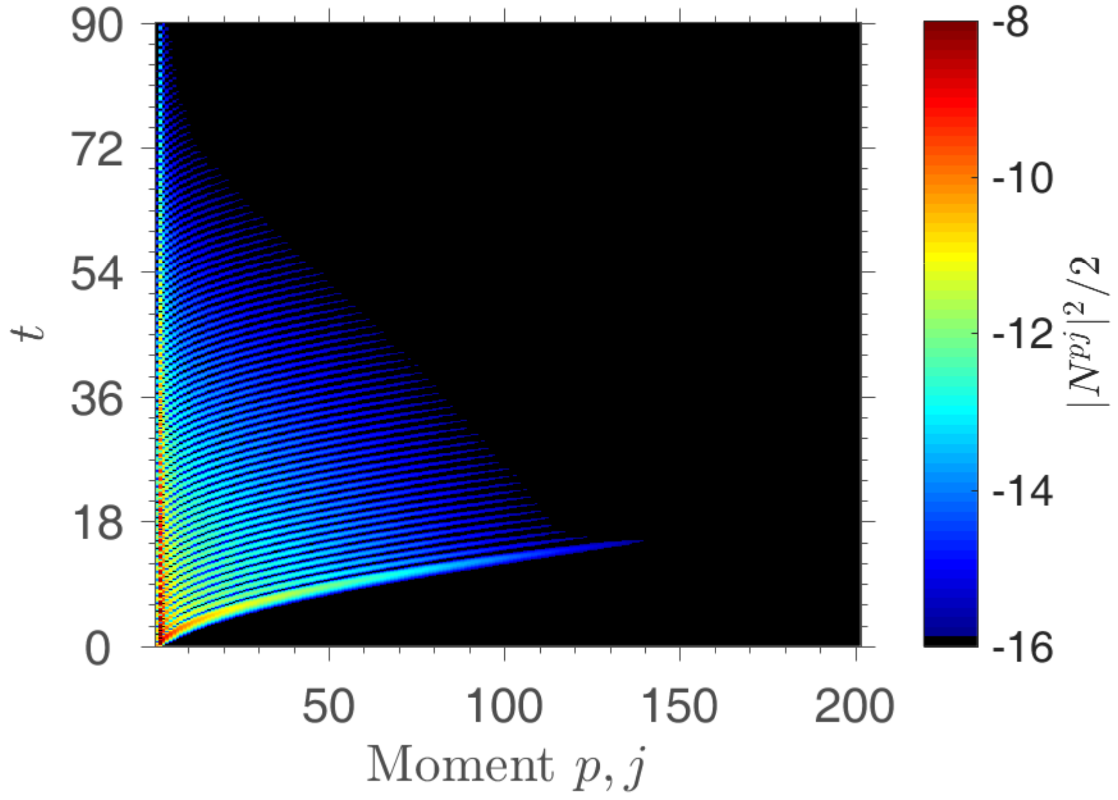
Numerical Model – Time Evolution Example

50 Hermite, 4 Laguerre Polynomials

2 minutes wall-clock time on a Macbook Pro 2016



Low Collisionality



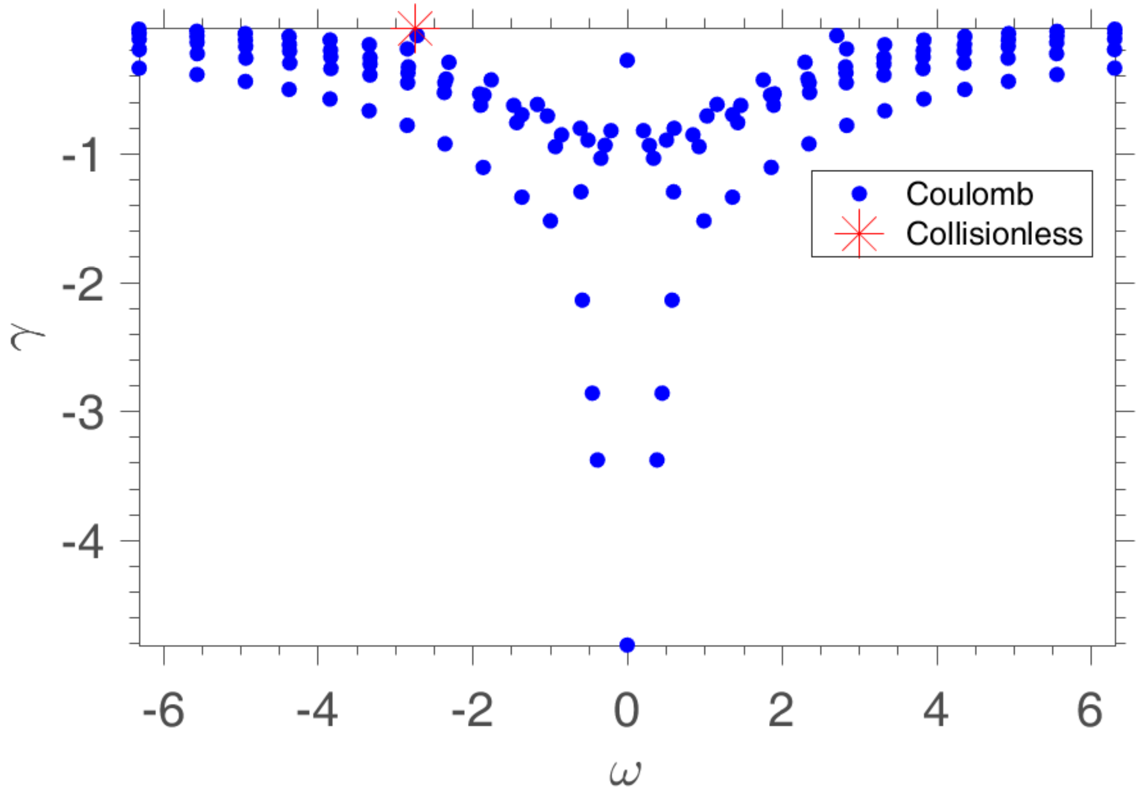
High Collisionality

Numerical Model – Eigenvalue Analysis Example

50 Hermite, 4 Laguerre Polynomials

2 minutes wall-clock time on a Macbook Pro 2016

Growth Rate



Frequency

Numerical Model - Difficulties

Develop algorithms for the following functions

T_{lk}^{pj} $$v^l P_l\left(\frac{v_\parallel}{v}\right) L_k^{l+1/2}(v^2) = \sum_{p=0}^{l+2k} \sum_{j=0}^{k+l/2} T_{lk}^{pj} H_p\left(v_\parallel\right) L_j\left(v_\perp^2\right)$$

↓

Integrate both sides

↓

$$T_{lk}^{pj} = \sum_{q=0}^{l/2} \sum_{v=0}^{p/2} \sum_{i=0}^{k} \sum_{r=0}^{q} \sum_{s=0}^{\min(j,i)} \sum_{m=0}^{k-i} \frac{(-1)^{q+i+j+v+m}}{2^{\frac{3l+p}{2}+m+v-r}} \times \binom{l}{q}\binom{2(l-q)}{l}\binom{q}{r}\binom{r}{j-s}\binom{r}{i-s}\binom{s+r}{s} r! \times \frac{(k-i+l-1/2)!\,(l+p+2(m-r-v)-1)!!}{(p-2v)!\,(k-i-m)!\,(l+m-1/2)!\,v!\,m!}$$

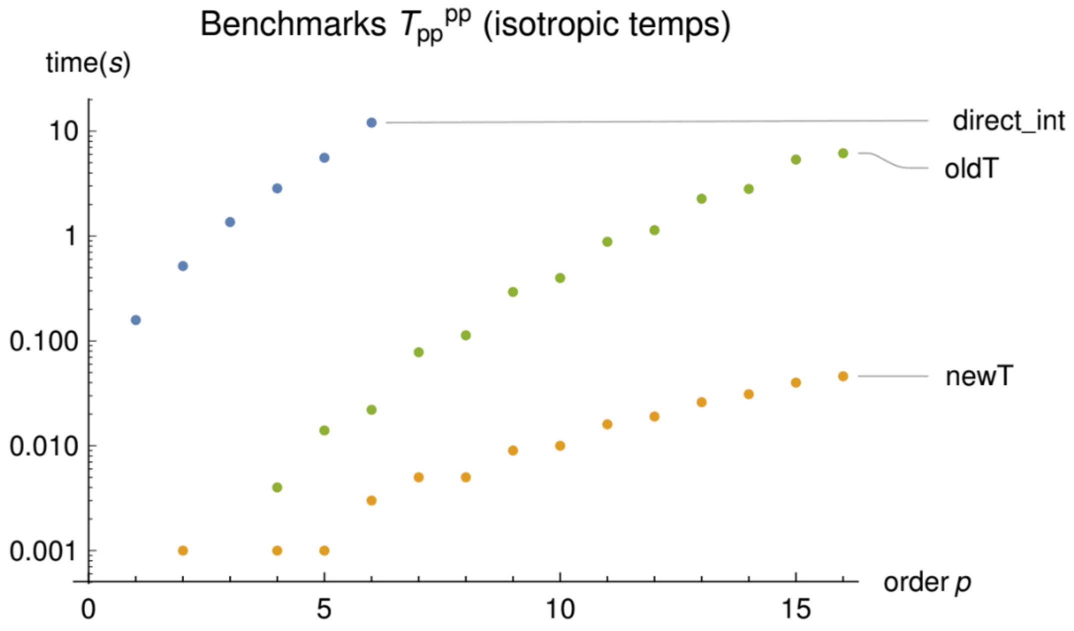
Numerical Model - Difficulties

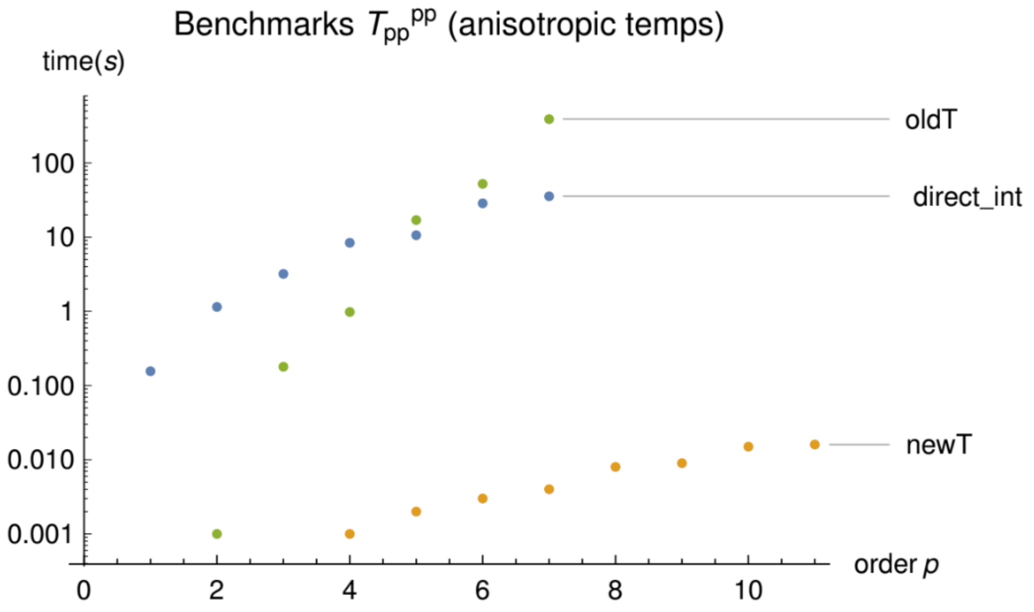
Develop algorithms for the following functions

$$T_{lk}^{pj} \qquad v^l P_l\left(\frac{v_\parallel}{v}\right) L_k^{l+1/2}(v^2) = \sum_{p=0}^{l+2k} \sum_{j=0}^{k+l/2} T_{lk}^{pj} H_p\left(v_\parallel\right) L_j\left(v_\perp^2\right)$$

Obtain a recursion relation using polynomial orthogonality relations

Optimization using double Hermite integration techniques from Integrals and Series: Special functions, A. Platonovich, 2002





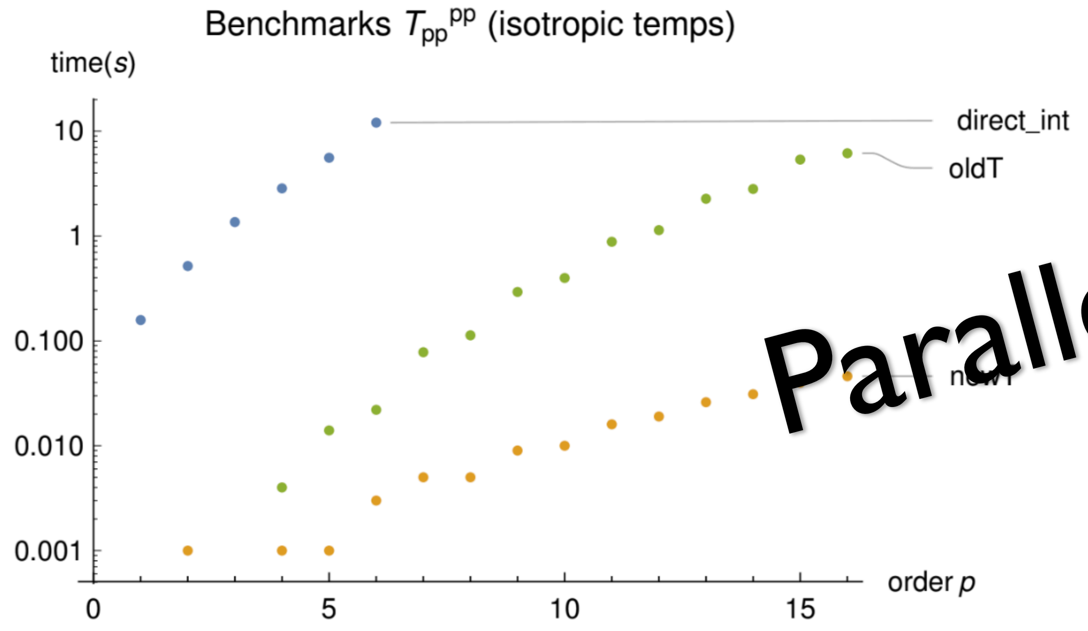
Numerical Model - Difficulties

Develop algorithms for the following functions

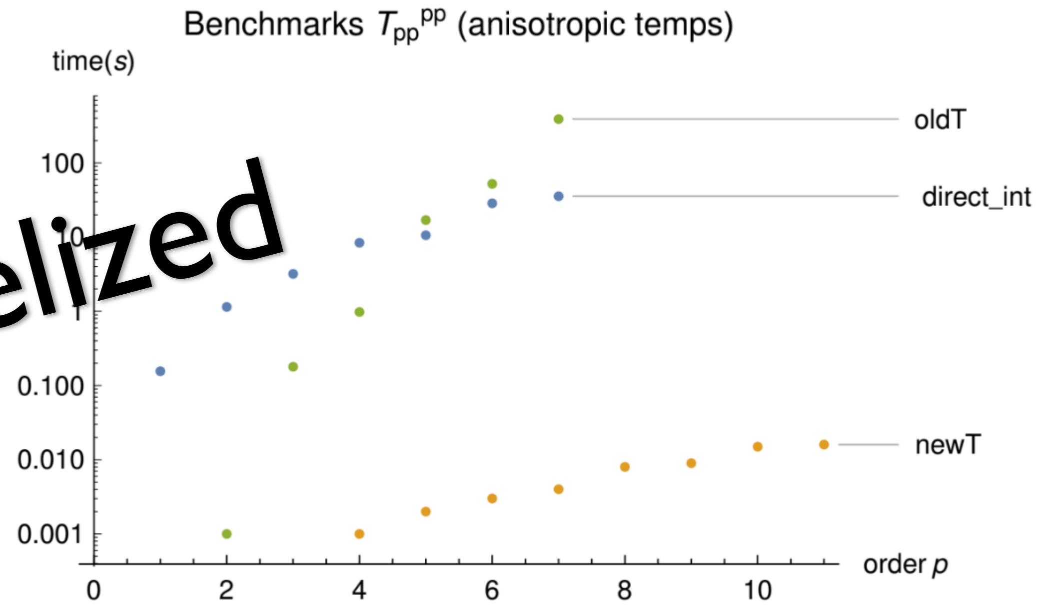
$$T_{lk}^{pj} \qquad v^l P_l\left(\frac{v_\parallel}{v}\right) L_k^{l+1/2}(v^2) = \sum_{p=0}^{l+2k} \sum_{j=0}^{k+l/2} T_{lk}^{pj} H_p\left(v_\parallel\right) L_j\left(v_\perp^2\right)$$

Obtain a recursion relation using polynomial orthogonality relations

Optimization using double Hermite integration techniques from Integrals and Series: Special functions, A. Platonovich, 2002



Parallelized



Numerical Model - Difficulties

Develop algorithms for the following functions

T_{lk}^{pj}

$$v^l P_l\left(\frac{v_\parallel}{v}\right) L_k^{l+1/2}(v^2) = \sum_{p=0}^{l+2k} \sum_{j=0}^{k+l/2} T_{lk}^{pj} H_p\left(v_\parallel\right) L_j\left(v_\perp^2\right)$$

Obtain a recursion relation using polynomial orthogonality relations

$d_j^{l,n}$

$$d_j^{l,n} = \sum_{j_k \mid \sum_{k=1}^{h} j_k = j} (-1)^h \prod_{k=1}^{h} t_{j_k}^{l-\sum_{g=1}^{k-1} j_g, n-\sum_{g=1}^{k-1} j_g}$$

Integer partitions

(Mathematica built-in function)

Numerical Model - Difficulties

Develop algorithms for the following functions

T_{lk}^{pj}

$$v^l P_l\left(\frac{v_\parallel}{v}\right) L_k^{l+1/2}(v^2) = \sum_{p=0}^{l+2k} \sum_{j=0}^{k+l/2} T_{lk}^{pj} H_p\left(v_\parallel\right) L_j\left(v_\perp^2\right)$$

Obtain a recursion relation using polynomial orthogonality relations

$d_j^{l,n}$

$$d_j^{l,n} = \sum_{j_k \mid \sum_{k=1}^{h} j_k = j} (-1)^h \prod_{k=1}^{h} t_{j_k}^{l-\sum_{g=1}^{k-1} j_g, n-\sum_{g=1}^{k-1} j_g}$$

Integer partitions

(Mathematica built-in function)

Non-Parallelized

Numerical Model - Difficulties

Develop algorithms for the following functions

T_{lk}^{pj}

$$v^l P_l\left(\frac{v_\parallel}{v}\right) L_k^{l+1/2}(v^2) = \sum_{p=0}^{l+2k} \sum_{j=0}^{k+l/2} T_{lk}^{pj} H_p\left(v_\parallel\right) L_j\left(v_\perp^2\right)$$

Obtain a recursion relation using polynomial orthogonality relations

$d_j^{l,n}$

$$d_j^{l,n} = \sum_{j_k | \sum_{k=1}^h j_k = j} (-1)^h \prod_{k=1}^h t_{j_k}^{l-\sum_{g=1}^{k-1} j_g, n-\sum_{g=1}^{k-1} j_g}$$

Compute "integer partitions" (Mathematica built-in function)

$\boldsymbol{P}_{a_1,a_2,\ldots,a_l}^l$

$$\boldsymbol{P}^0 = 1 \quad \boldsymbol{P}^1 = \boldsymbol{b} \quad \boldsymbol{P}^2 = \boldsymbol{bb} - \frac{\boldsymbol{I}}{3}$$

Need to compress high-dimensional tensors for memory purposes

Numerical Model - Difficulties

Develop algorithms for the following functions

T_{lk}^{pj}

$$v^l P_l\left(\frac{v_\parallel}{v}\right) L_k^{l+1/2}(v^2) = \sum_{p=0}^{l+2k} \sum_{j=0}^{k+l/2} T_{lk}^{pj} H_p\left(v_\parallel\right) L_j\left(v_\perp^2\right)$$

Obtain a recursion relation using polynomial orthogonality relations

$d_j^{l,n}$

$$d_j^{l,n} = \sum_{j_k | \sum_{k=1}^{h} j_k = j} (-1)^h \prod_{k=1}^{h} t_{j_k}^{l-\sum_{g=1}^{k-1} j_g, n-\sum_{g=1}^{k-1} j_g}$$

Compute "integer partitions" (Mathematica built-in function)

$\boldsymbol{P}_{a_1,a_2,\ldots,a_l}^l$

$$\boldsymbol{P}^0 = 1 \quad \boldsymbol{P}^1 = \boldsymbol{b} \quad \boldsymbol{P}^2 = \boldsymbol{bb} - \frac{\boldsymbol{I}}{3}$$

Need to compress high-dimensional tensors for memory purposes

Parallelized

Numerical Model - Difficulties

Develop algorithms for the following functions

$$T_{lk}^{pj} \qquad v^l P_l\left(\frac{v_\parallel}{v}\right) L_k^{l+1/2}(v^2) = \sum_{p=0}^{l+2k} \sum_{j=0}^{k+l/2} T_{lk}^{pj} H_p\left(v_\parallel\right) L_j\left(v_\perp^2\right)$$

Obtain a recursion relation using polynomial orthogonality relations

$$d_j^{l,n} \qquad d_j^{l,n} = \sum_{j_k | \sum_{k=1}^{h} j_k = j} (-1)^h \prod_{k=1}^{h} t_{j_k}^{l-\sum_{g=1}^{k-1} j_g, n-\sum_{g=1}^{k-1} j_g}$$

Compute "integer partitions" (Mathematica built-in function)

$$\boldsymbol{P}_{a_1,a_2,\ldots,a_l}^{l} \qquad \boldsymbol{P}^0 = 1 \quad \boldsymbol{P}^1 = \boldsymbol{b} \quad \boldsymbol{P}^2 = \boldsymbol{bb} - \frac{\boldsymbol{I}}{3}$$

Need to compress high-dimensional tensors for memory purposes

$\boldsymbol{P}^{l+n-2i} \cdot \overline{\boldsymbol{P}^l \cdot^i \boldsymbol{P}^n}$ Traceless symmetrization of the i-th inner tensor product

Bottleneck for non-linear calculations

Numerical Model - Difficulties

Develop algorithms for the following functions

T_{lk}^{pj}

$$v^l P_l\left(\frac{v_\parallel}{v}\right) L_k^{l+1/2}(v^2) = \sum_{p=0}^{l+2k} \sum_{j=0}^{k+l/2} T_{lk}^{pj} H_p\left(v_\parallel\right) L_j\left(v_\perp^2\right)$$

Obtain a recursion relation using polynomial orthogonality relations

$d_j^{l,n}$

$$d_j^{l,n} = \sum_{j_k | \sum_{k=1}^{h} j_k = j} (-1)^h \prod_{k=1}^{h} t_{j_k}^{l-\sum_{g=1}^{k-1} j_g, n-\sum_{g=1}^{k-1} j_g}$$

Compute "integer partitions" (Mathematica built-in function)

$\boldsymbol{P}^l_{a_1,a_2,\dots,a_l}$

$$\boldsymbol{P}^0 = 1 \quad \boldsymbol{P}^1 = \boldsymbol{b} \quad \boldsymbol{P}^2 = \boldsymbol{b}\boldsymbol{b} - \frac{\boldsymbol{I}}{3}$$

Need to compress high-dimensional tensors for memory purposes

$\boldsymbol{P}^{l+n-2i} \cdot \overline{\boldsymbol{P}^l \cdot^i \boldsymbol{P}^n}$ Traceless symmetrization of the i-th inner tensor product

Bottleneck for non-linear calculations

Non-Parallelized

Numerical Model - Outlook

Possibility of precomputing matrix coefficients for later use

Symmetrization of high dimensional tensors only needed in the non-linear case

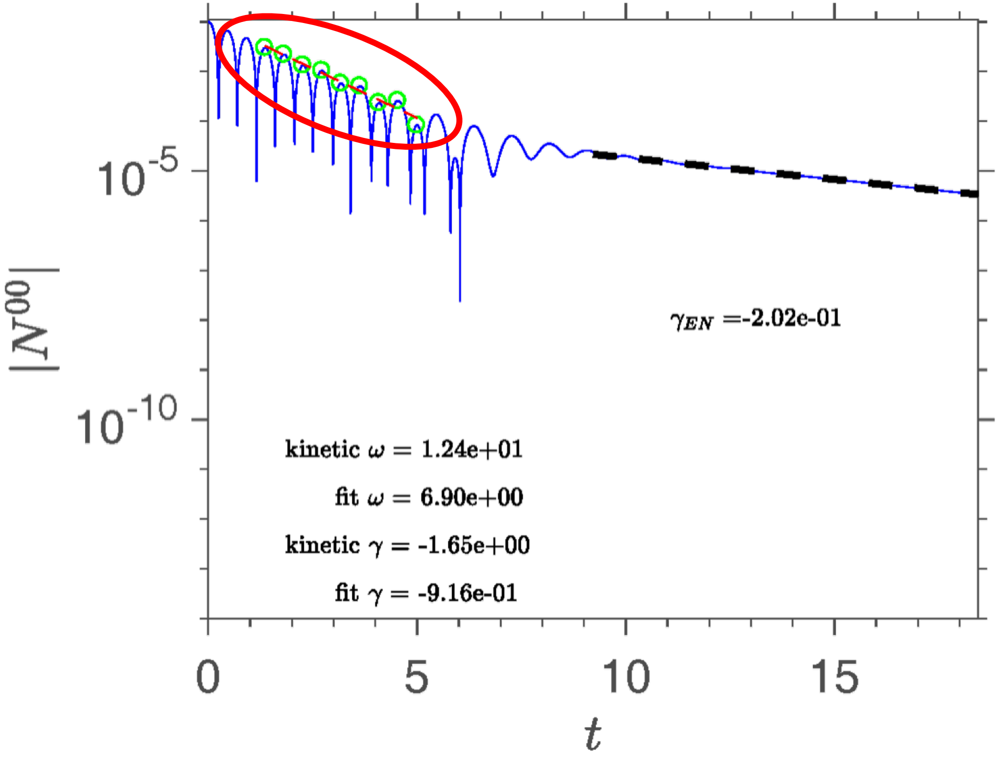
Show feasibility for the linear case, develop algorithms for the non-linear case (HPC)

Electron Plasma Waves –Damping at Arbitrary Collisionality

- One spatial and two velocity dimensions
- Electron perturbations only
- Time-evolution of $N^{00} \sim \phi$

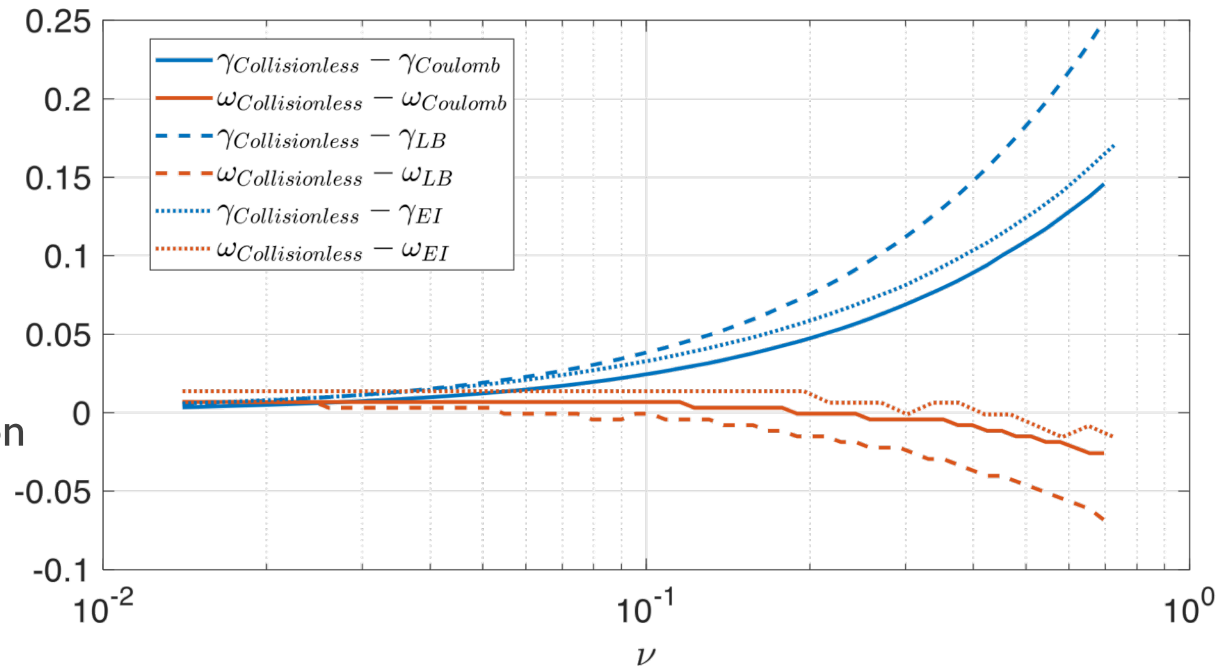
- Collisional and Landau Damping computed for the first time with the full Coulomb collision operator

R. Jorge et. al., JPP **85** (2), 2019



Parameter scan

Three collision operators



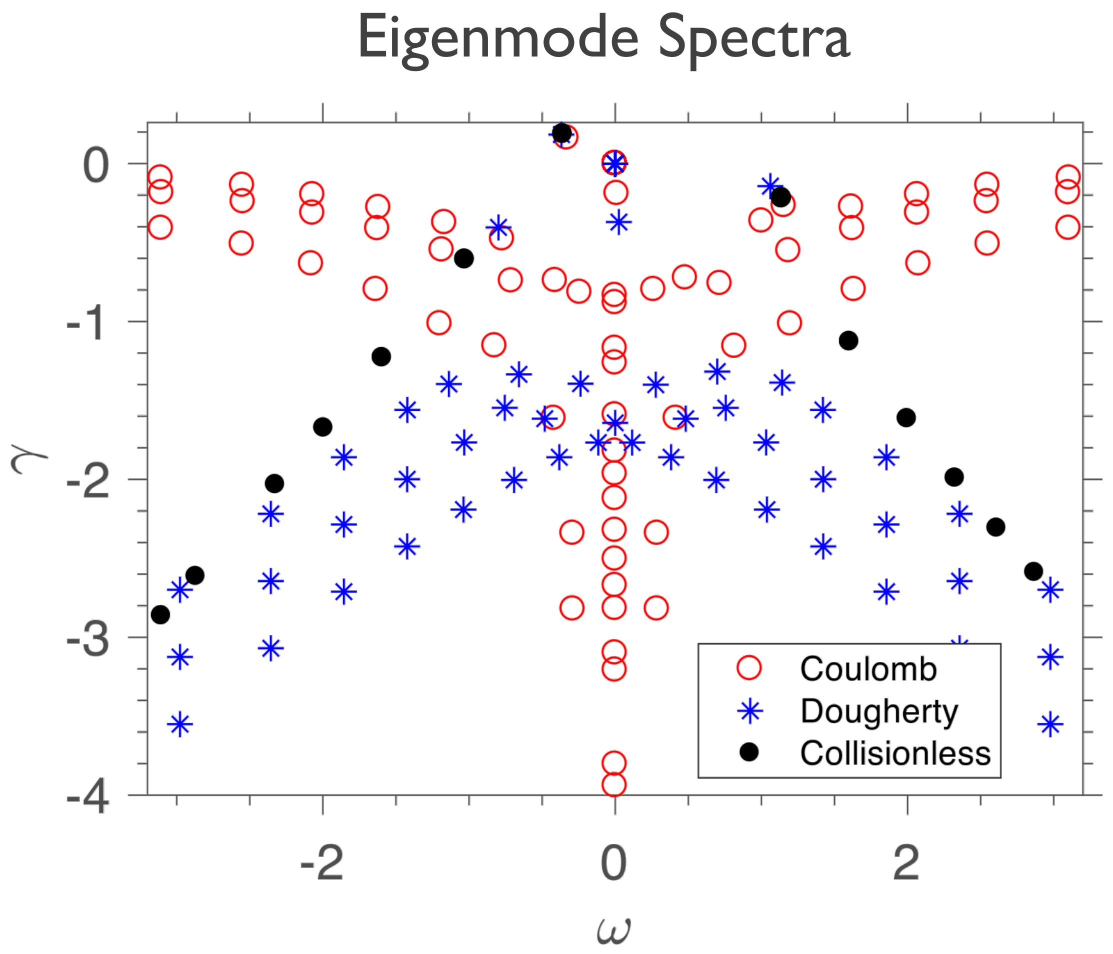
Collisionality

First Study on Coulomb Collision Effects on the Drift-Wave Instability

- Two spatial and two velocity dimensions in slab geometry
- Electron and ion perturbations
- Finite background density gradient

- Important deviations between full Coulomb collision operator and presently used collision operators

R. Jorge *et al.*, PRL **121** (16), 2018



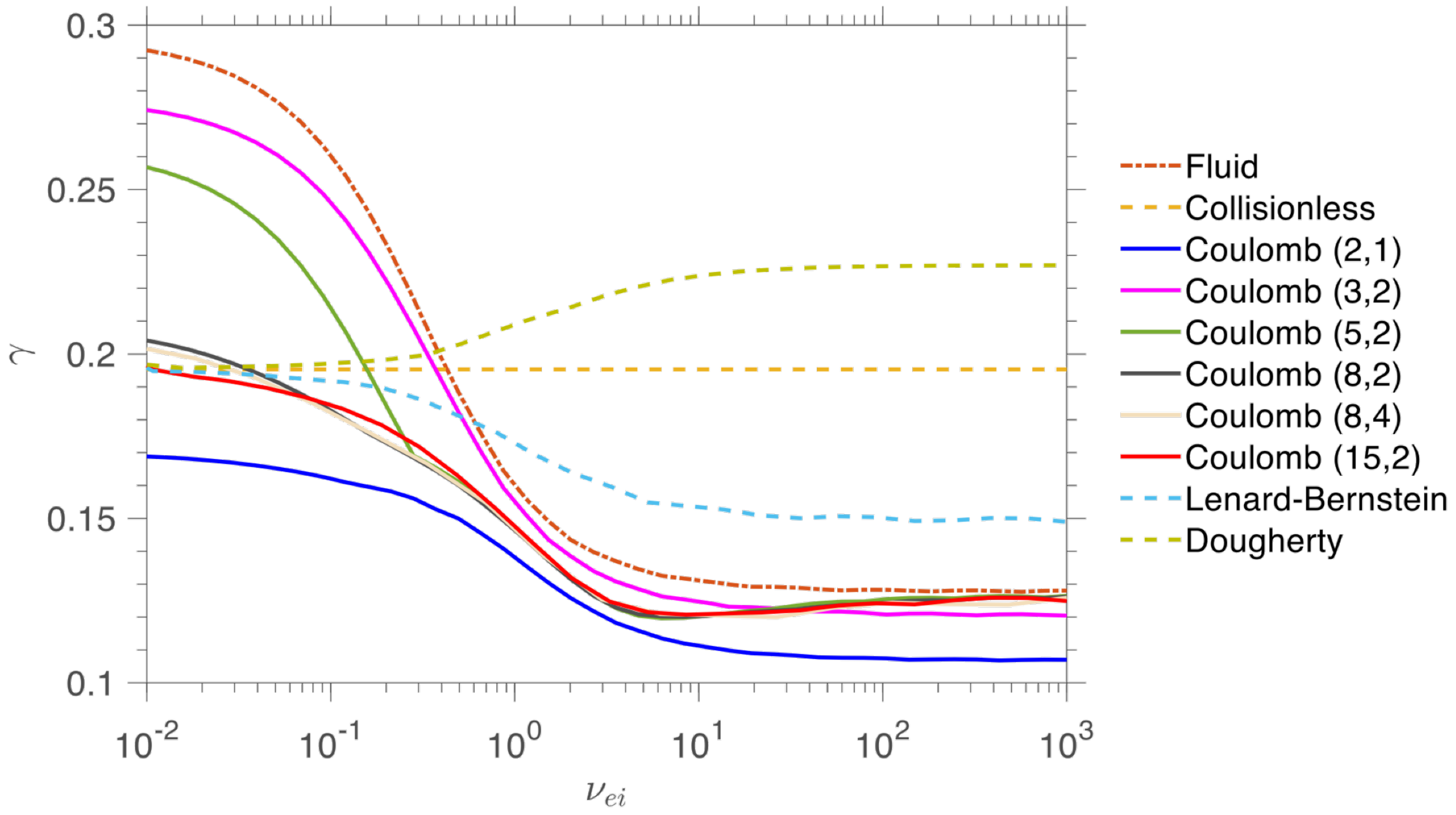
First Study on Coulomb Collision Effects on the Drift-Wave Instability

- Two spatial and two velocity dimensions in slab geometry
- Electron and ion perturbations
- Finite background density gradient

- Important deviations between full Coulomb collision operator and presently used collision operators

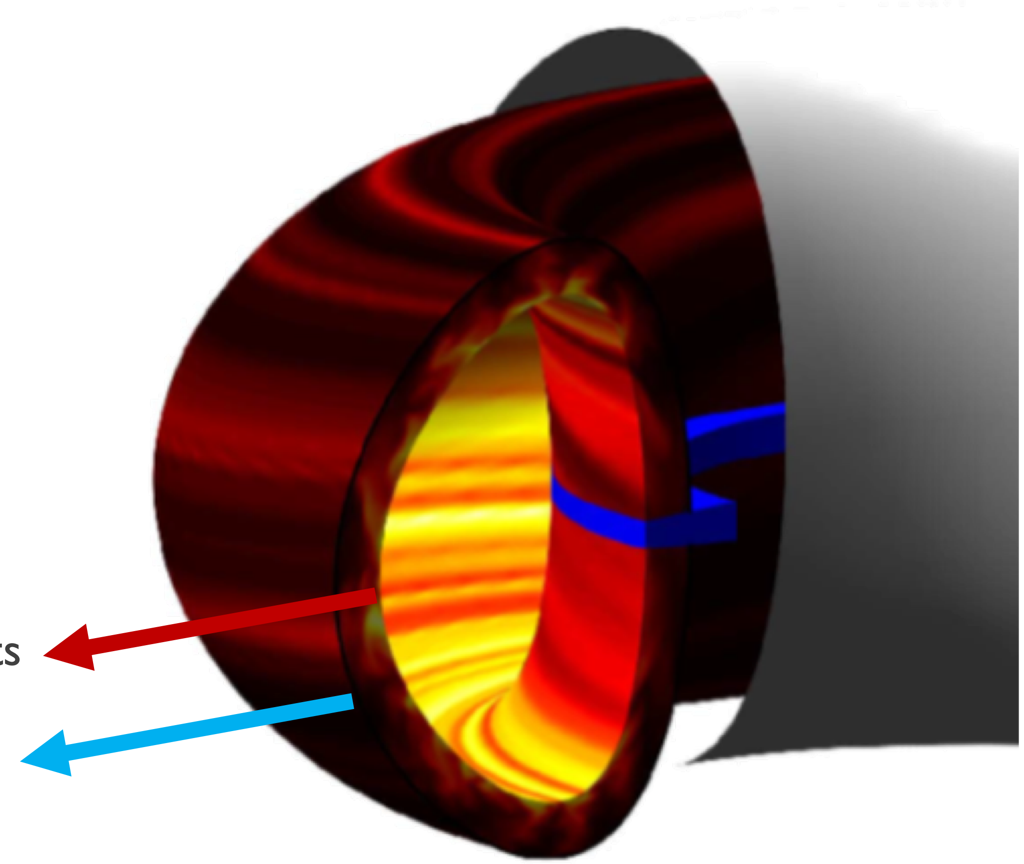
Instability Growth Rate

R. Jorge *et al.*, PRL **121** (16), 2018



Outlook

- For the first time, nonlinear Coulomb collisions are included in an edge model
- Numerical efficiency shown for the linear case
- Future
  - Obtain insight of the physics at Edge and SOL with non-linear simulations



Less collisional, more moments

Collisional, a few moments

Summary

	This Work	Qin et al. 2007[1]	Hahm et al. 2009[2]	Dimits et al. 2012[3]	Madsen et al. 2013[4]	Mandell et al. 2018[5]
Large Scales	EM $\mathrm{O}(\epsilon^2)$	EM $\mathrm{O}(\epsilon)$	ES $\mathrm{O}(\epsilon^2)$	ES $\mathrm{O}(\epsilon^2)$	ES $\mathrm{O}(\epsilon)$	ES $\mathrm{O}(\epsilon)$
Small Scales	EM $\mathrm{O}(\epsilon^2)$	EM $\mathrm{O}(\epsilon^2)$	EM $\mathrm{O}(\epsilon^2)$	EM $\mathrm{O}(\epsilon^2)$	EM $\mathrm{O}(\epsilon)$	ES $\mathrm{O}(\epsilon)$
Collisions	Yes	No	No	No	No	Simplified
Poisson's Eq.	Moments	$\int (\dots)\, d^3v$	Long-Wavelength Limit	$\int (\dots)\, d^3v$	Long-Wavelength Limit, Padé	Moments
$\mathrm{B}_{\parallel}^{*}$	Exact	Exact	Exact	Exact	$\mathrm{O}(\epsilon)$	B

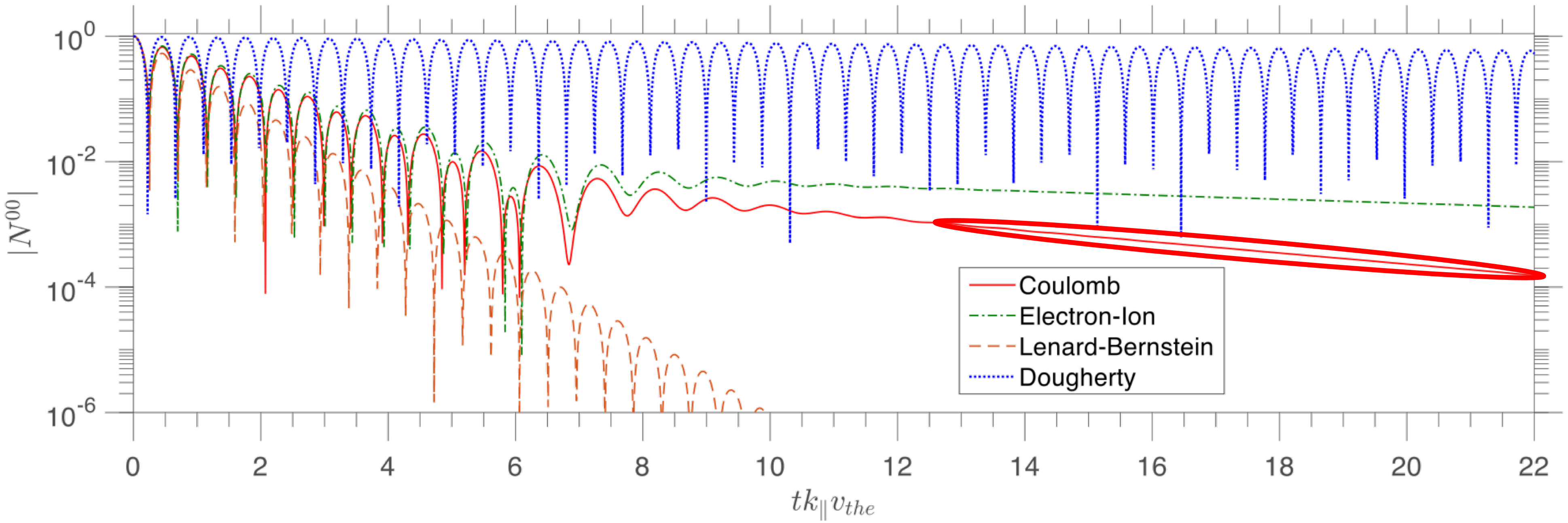
[1] Qin *et al.*, Physics of Plasmas **14**, 056110 (2007)
[2] Hahm *et al.*, Physics of Plasmas **16**, 022305 (2009)
[3] Dimits, Physics of Plasmas **19**, 022504 (2012)
[4] Madsen, Physics of Plasmas **20**, 072301 (2013)
[5] Mandell *et al.*, J. Plasma Phys **84**, 905840108 (2018)

Electron Plasma Waves – Entropy Mode

- One spatial and two velocity dimensions
- Electron perturbations only
- Time-evolution of $N^{00} \sim \phi$

- Electron-ion collisions known to yield long-time zero-frequency behaviour
- Full Coulomb collisions increase the entropy mode damping rate

R. Jorge et. al., JPP **85** (2), 2019



Zero-frequency entropy mode

Single Particle Dynamics – First Step

Large Scales $\epsilon \sim k_\perp \rho_i \ll 1 \,, \; \frac{e\phi}{T_e} \sim 1$

Start from

$$L = q\mathbf{A}\cdot\dot{\mathbf{x}} - q\phi - \frac{mv^2}{2}$$

Split between parallel and perpendicular velocity

Describe guiding center motion $O(\epsilon^2)$

$$L = q\mathbf{A}^*\cdot\dot{\mathbf{R}} - q\phi^* - \frac{mv_\parallel^2}{2} + \mu\frac{m\dot{\theta}}{q}$$

Single Particle Dynamics – Second Step

Small Scales $\epsilon \sim \frac{e\phi}{T_e} \ll 1\,,\; k_\perp \rho_i \sim 1$

Start from

$$L = q\mathbf{A}^* \cdot \dot{\mathbf{R}} - q\phi^* - \frac{mv_\parallel^2}{2} + \mu \frac{m\dot{\theta}}{q}$$

Introduce small scale fluctuations

Describe gyro center motion $O(\epsilon^2)$

$$L = q\mathbf{A}^* \cdot \dot{\mathbf{R}} - q\phi^* - \frac{mv_\parallel^2}{2} + \mu \frac{m\dot{\theta}}{q} - q\left\langle \phi - v_\parallel A_\parallel \right\rangle - q\frac{\partial \left\langle \left(\widetilde{\phi} - v_\parallel \widetilde{A}_\parallel\right)^2 \right\rangle}{\partial \mu}$$

Single Particle Dynamics – Equations of Motion

$$\dot{\mathbf{R}} = v_{||}\mathbf{b} + \mathbf{v}_{\mathbf{E}\times\mathbf{B}} + \text{Other Drifts}$$

Including large scale $\frac{\nabla\phi_0(\mathbf{R}) \times \mathbf{B}}{B^2}$

and small scale $\frac{\nabla \langle\phi_1(\mathbf{x})\rangle \times \mathbf{B}}{B^2}$ fluctuations

Curvature Drift

Grad-B Drift

Polarization Drift

Single Particle Dynamics – Equations of Motion

$$\dot{\mathbf{R}} = v_{||}\mathbf{b} + \mathbf{v}_{\mathbf{E}\times\mathbf{B}} + \text{Other Drifts}$$

$$\dot{v}_{||} = qE_{||} + \mu\nabla_{||}B + \text{Non-Linear Forces}$$

Including large scale $\nabla_{||}\phi_0(\mathbf{R})$

and small scale $\nabla_{||}\left\langle\phi_1(\mathbf{x})\right\rangle$ fluctuations

Single Particle Dynamics – Equations of Motion

$$\dot{\mathbf{R}} = v_{||}\mathbf{b} + \mathbf{v}_{\mathbf{E}\times\mathbf{B}} + \text{Other Drifts}$$

$$\dot{v}_{||} = qE_{||} + \mu\nabla_{||}B + \text{Non-Linear Forces}$$

$$\dot{\mu} = 0$$

↓

Conserved Adiabatic Invariant

Properties of Periphery Turbulence

- Low-frequency $\omega \ll \Omega_i$
- Large Scale Fluctuations

  $k_\perp \rho_i \ll 1 \qquad \frac{e\phi}{T_e} \sim 1$

- Small Scale Fluctuations

  $k_\perp \rho_i \sim 1 \qquad \frac{e\phi}{T_e} \ll 1$

→ **Gyrokinetic Theory**

$$\epsilon = k_\perp \rho_i \frac{e\phi}{T_e} \ll 1$$

- Wide range of temperatures and densities

  $T_e \sim 1 - 10^3 \text{ eV} \qquad n \sim 10^{18} - 10^{20} \text{ m}^{-3}$

→ Arbitrary Collisionality

Example – 1D Linear Gyrokinetic Moment Hierarchy

Phase Mixing
(coupling with other moments)

Collisions

$$\frac{\partial N^{pj}}{\partial t} + \frac{1}{2}\frac{\partial N^{p+1j}}{\partial z} + p\frac{\partial N^{p-1j}}{\partial z} = 2\delta_{p,1} K_j(k_\perp \rho) E_\parallel + C^{pj}$$

Time Evolution

$$\text{Kernel} \quad K_j(x) = \frac{1}{j!}\left(\frac{x}{2}\right)^{2j} e^{-x^2/4}$$

Analytical Closed formula for the Gyroaverage operation

Full finite Larmor radius effects

Physics That We Are Able To Capture – High Collisionality

$$F = F_M(1 + \delta F) \quad \text{with} \quad \delta F = \sum_{p,j} N^{pj} H_p L_j$$

Semi-collisional closure

$$\delta F \sim \frac{\lambda_{\mathrm{mfp}}}{L_{\|}}$$

→

Drift-Reduced Braginskii equations retrieved

$$N^{30} \sim q_{\|} = -\chi_{\|} \nabla_{\|} T$$

Moments of the Collisional Operator

$$C^{pj} = \int \langle C(F) \rangle H_p L_j dv_{||} d\mu$$

After integration

$$C^{pj} = \text{Kernel} \times \mathrm{f(n)} \times \text{moments } N^{pj} \text{ of } \mathrm{F}$$