

Turbulent Dissipation in Simple Vlasov Systems

James Juno

April 16, 2019



UNIVERSITY OF
MARYLAND

The Gkeyll (and Hyde) Framework

"It is one thing to mortify curiosity, another to conquer it."

- The Gkeyll framework is flexible suite of solvers for plasma physics being developed at the Princeton Plasma Physics Lab, UMD, Virginia Tech, and UNH
- Gkeyll team: **Ammar Hakim**, Greg Hammett, Jason TenBarge, Petr Cagas, Noah Mandell, Manaure Francisquez, Tess Bernard, Valentin Skoutnev, and Liang Wang
- Relevant publications:
 - J. Juno, A. Hakim, J. TenBarge, E. Shi, W. Dorland. Discontinuous Galerkin algorithms for fully kinetic plasmas. *J. Comp. Phys.*, 353 (2018).
 - A. Hakim and J. Juno. Generating a Quadrature and Matrix-free Discontinuous Galerkin algorithm for (plasma) kinetic equations. *In preparation.*



<https://bitbucket.org/ammarhakim/gkyl/src/default/>

conda install -c gkyl gkyl

The Vlasov-Maxwell System

$$\frac{\partial f}{\partial t} + \nabla_{\mathbf{z}} \cdot (\alpha f) = 0$$

where $\nabla_{\mathbf{z}} = (\nabla_{\mathbf{x}}, \nabla_{\mathbf{v}})$ and $\alpha = (\mathbf{v}, \frac{q_s}{m_s}(\mathbf{E} + \mathbf{v} \times \mathbf{B}))$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = 0$$

$$\epsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t} - \nabla \times \mathbf{B} = -\mu_0 \mathbf{J}$$

$$\nabla \cdot \mathbf{B} = 0$$

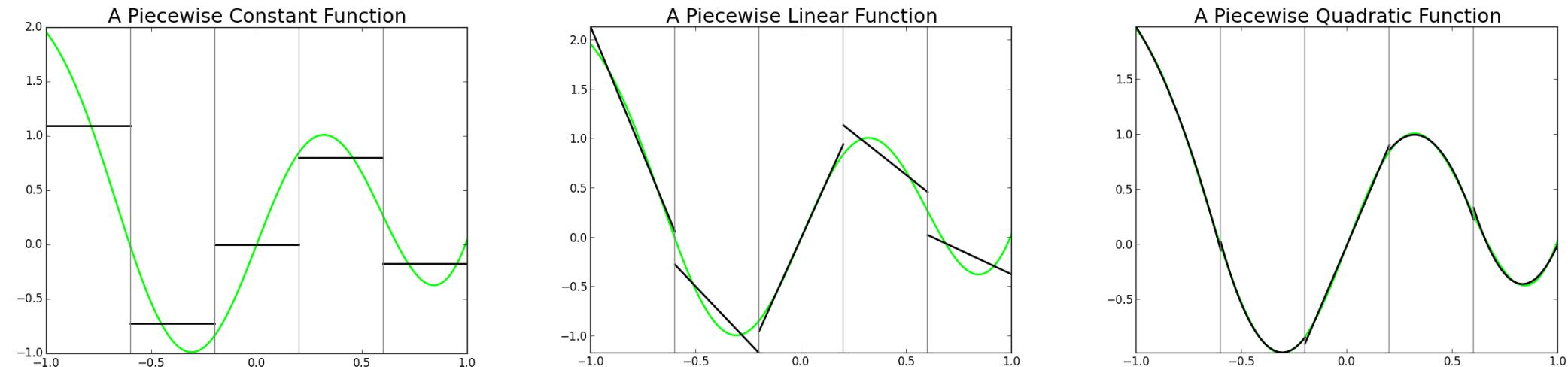
$$\nabla \cdot \mathbf{E} = \rho_c / \epsilon_0$$

$$\rho_c = \sum_s q_s \int_{-\infty}^{\infty} f_s(t, \mathbf{x}, \mathbf{v}) d\mathbf{v} \quad \mathbf{J} = \sum_s q_s \int_{-\infty}^{\infty} \mathbf{v} f_s(t, \mathbf{x}, \mathbf{v}) d\mathbf{v}$$

This equation system is high-dimensional (**up to 6 dimensions plus time**)

The Discontinuous Galerkin Finite Element Method

- We choose to use the discontinuous Galerkin framework as our spatial discretization because it combines aspects of
 - Finite elements: high order accuracy and ability to handle complicated geometries
 - Finite volume: locality of data and stability enforcing limiters



The Discrete Vlasov Equation

- What does the discontinuous Galerkin discretization of the Vlasov equation look like?
 - Consider a phase space mesh \mathcal{T} with cells $K_j \in \mathcal{T}, j = 1, \dots, N$.
 - Then the problem formulation is, find $f_h \in \mathcal{V}_h^p$, such that for all $K_j \in \mathcal{T}$,

$$\int_{K_j} w \frac{\partial f_h}{\partial t} d\mathbf{z} + \oint_{\partial K_j} w^- \mathbf{n} \cdot \hat{\mathbf{F}} dS - \int_{K_j} \nabla_z w \cdot \alpha_h f_h d\mathbf{z} = 0$$
$$f_h(\mathbf{z}, t) = \sum_n^{N_p} F_n(t) w_n(\mathbf{z}) \quad \mathcal{V}_h^p = \{v : v|_{K_j} \in \mathbf{P}^p, \forall K_j \in \mathcal{T}\},$$

Upwind fluxes are used for the streaming term and a relaxed global Lax-Friedrichs flux is used for the acceleration

$$\mathbf{n} \cdot \hat{\mathbf{F}} = \frac{1}{2} \mathbf{n} \cdot \left(\boldsymbol{\alpha}_h^+ (f_h^+ + f_h^-) - \boldsymbol{\tau} (f^+ - f^-) \right)$$

$$\text{where } \boldsymbol{\tau} = \max_{\mathcal{T}} \left(\frac{q}{m} \mathbf{E}_h + \frac{q}{m} \mathbf{v} \times \mathbf{B}_h \right)$$

note that the phase space flux is continuous at corresponding surface interfaces

The Discrete Maxwell Equations

$$\int_{\Omega_j} \phi \frac{\partial \mathbf{B}_h}{\partial t} d\mathbf{x} + \oint_{\partial \Omega_j} d\mathbf{s} \times (\phi^- \hat{\mathbf{E}}_h) - \int_{\Omega_j} \nabla \phi \times \mathbf{E}_h d\mathbf{x} = 0$$

$$\epsilon_0 \mu_0 \int_{\Omega_j} \phi \frac{\partial \mathbf{E}_h}{\partial t} d\mathbf{x} - \oint_{\partial \Omega_j} d\mathbf{s} \times (\phi^- \hat{\mathbf{B}}_h) + \int_{\Omega_j} \nabla \phi \times \mathbf{B}_h d\mathbf{x} = -\mu_0 \int_{\Omega_j} \phi \mathbf{J}_h d\mathbf{x}$$

with central fluxes

$$\hat{\mathbf{E}}_h = [\![\mathbf{E}]\!]$$

$$\hat{\mathbf{B}}_h = [\![\mathbf{B}]\!]$$

$$[\![g]\!] \equiv (g^+ + g^-)/2$$

or upwind fluxes

$$\hat{E}_2 = [\![E_2]\!] - c \{B_3\}$$

$$\hat{E}_3 = [\![E_3]\!] + c \{B_2\}$$

$$\hat{B}_2 = [\![B_2]\!] + \{E_3\}/c$$

$$\hat{B}_3 = [\![B_3]\!] + \{E_2\}/c$$

$$\{g\} \equiv (g^+ - g^-)/2$$

Note that upwind fluxes are defined with respect to a local coordinate system where subscripts 2 and 3 define the two directions tangential to the surface

Proving conservation relations

- The continuous Vlasov-Maxwell system has a number of conserved quantities
 - Density
 - Momentum
 - Energy
 - etc.
- What conserved quantities does our spatial discretization retain?

$$\int_{K_j} w \frac{\partial f_h}{\partial t} d\mathbf{z} + \oint_{\partial K_j} w^- \mathbf{n} \cdot \hat{\mathbf{F}} dS - \int_{K_j} \nabla_z w \cdot \alpha_h f_h d\mathbf{z} = 0$$

- Can choose the test function carefully, as long as test function is in the solution space

$$\sum_j \int_{K_j} \frac{\partial f_h}{\partial t} d\mathbf{z} + \sum_j \oint_{\partial K_j} \mathbf{n} \cdot \hat{\mathbf{F}} dS = 0$$

$$\sum_s \sum_j \int_{K_j} \frac{1}{2} m_s |\mathbf{v}|^2 \frac{\partial f_{h_s}}{\partial t} d\mathbf{z} + \sum_s \sum_j \oint_{\partial K_j} \frac{1}{2} m_s |\mathbf{v}|^2 \mathbf{n} \cdot \hat{\mathbf{F}} dS - \underbrace{\sum_s \sum_j \int_{K_j} \nabla_{\mathbf{z}} \left(\frac{1}{2} m_s |\mathbf{v}|^2 \right) \cdot \alpha_{h_s} f_{h_s} d\mathbf{z}}_{\sum_j \int_{\Omega_j} \mathbf{J}_h \cdot \mathbf{E}_h d^3 \mathbf{x}} = 0$$

Conservation Relations

- The discrete system conserves total density
- The discrete phase space flow is incompressible $\nabla_{\mathbf{z}} \cdot \boldsymbol{\alpha}_h = 0$
- Electromagnetic energy is conserved when using central fluxes, and bounded when using upwind fluxes

$$\sum_j \frac{d}{dt} \int_{\Omega_j} \left(\frac{\epsilon_0}{2} |\mathbf{E}_h|^2 + \frac{1}{2\mu_0} |\mathbf{B}_h|^2 \right) d^3\mathbf{x} \leq - \sum_j \int_{\Omega_j} \mathbf{J}_h \cdot \mathbf{E}_h d^3\mathbf{x}$$

- The total energy is conserved when central fluxes are used for Maxwell's equations

$$\frac{d}{dt} \sum_j \sum_s \int_{K_j} \frac{1}{2} m |\mathbf{v}|^2 f_h d\mathbf{z} + \frac{d}{dt} \sum_j \int_{\Omega_j} \left(\frac{\epsilon_0}{2} |\mathbf{E}_h|^2 + \frac{1}{2\mu_0} |\mathbf{B}_h|^2 \right) d^3\mathbf{x} = 0$$

Choosing a basis set

- Key algorithm question: what sort of basis set should be chosen?
 - Common choice: tensor product basis
 - Other choices: reduced basis sets like the Serendipity Element space, or maximal order basis

$$N_{\text{tensor}} = (p+1)^d \quad N_{\text{serendipity}} = \sum_{i=0}^{\min(d,p/2)} 2^{d-i} \binom{d}{i} \binom{p-i}{i} \quad N_{\text{max-order}} = \frac{(p+d)!}{p!d!}$$

Lagrange Tensor	Polynomial Order	1	2	3	4	5	6	7
Dimension	$(p+1)^d$							
2		4	9	16	25	36	49	64
3		8	27	64	125	216	343	512
4		16	81	256	625	1296	2401	4096
5		32	243	1024	3125	7776	16807	32768
6		64	729	4096	15625	46656	117649	262144

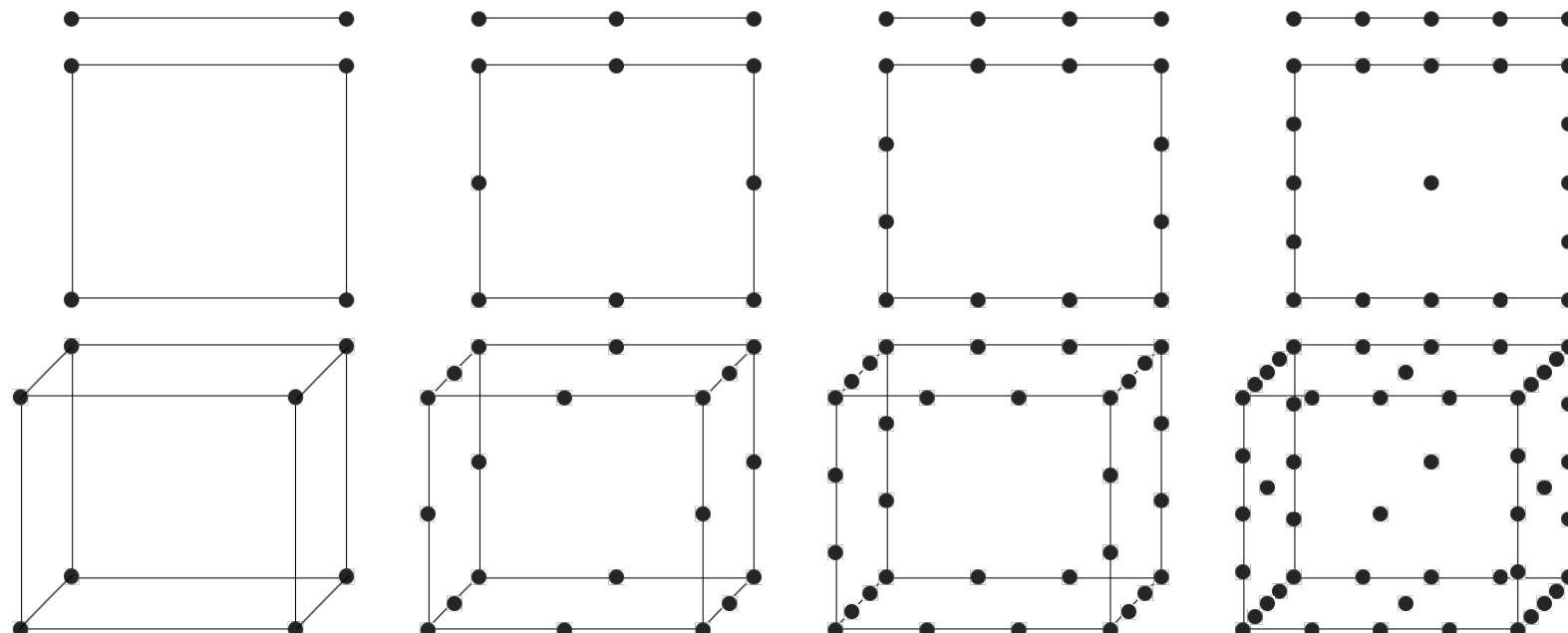
Table 1: Number of degrees of freedom internal to a cell in the Lagrange Tensor polynomial space

Serendipity Element	Polynomial Order	1	2	3	4	5	6	7
Dimension	$\sum_{i=0}^{\min(d,p/2)} \binom{d}{i} \binom{p-i}{i}$							
2		4	8	12	17	23	30	38
3		8	20	32	50	74	105	144
4		16	48	80	136	216	328	480
5		32	112	192	352	592	952	1472
6		64	256	448	880	1552	2624	4256

Table 2: Number of degrees of freedom internal to a cell in the Serendipity polynomial space

Defining the basis set

- We've chosen a basis set, but monomials may not be the best basis function expansion in the cell
 - Monomials lead to a poorly conditioned mass matrix
 - Clever choice of polynomials may have favorable computational properties
- When constructing the basis function expansion within the cell, it is common in the DG community to employ a *nodal basis*, where nodes are specified on a reference cell and basis functions take the value of 1 at one node and zero at all other nodes



A critical point: the importance of accurate integration

- A subtlety to the discretization of the Vlasov-Maxwell system is our conservation relations are *implicit* and require that certain integrals are computed exactly

$$\frac{d}{dt} \sum_j \sum_s \int_{K_j} \frac{1}{2} m |\mathbf{v}|^2 f_h d\mathbf{z} + \frac{d}{dt} \sum_j \int_{\Omega_j} \left(\frac{\epsilon_0}{2} |\mathbf{E}_h|^2 + \frac{1}{2\mu_0} |\mathbf{B}_h|^2 \right) d^3\mathbf{x} = 0$$

- This is different than standard fluid algorithms, where errors in the integration (aliasing errors) are often tolerated
- Exactly integrating the terms in the Vlasov equation with numerical quadrature is a nontrivial cost

$$\int_{K_j} w \frac{\partial f_h}{\partial t} d\mathbf{z} + \oint_{\partial K_j} w^- \mathbf{n} \cdot \hat{\mathbf{F}} dS - \int_{K_j} \nabla_{\mathbf{z}} w \cdot \boldsymbol{\alpha}_h f_h d\mathbf{z} = 0 \quad \rightarrow \quad \mathcal{O}(N_q N_p)$$

Orthonormal bases to the rescue

- The fundamental operations in our algorithm can be thought of as tensor-tensor products, for example the volume term for the Lorentz acceleration:

$$\mathcal{C}_{ijk} = \int_{K_l} \nabla_{\mathbf{v}} w_i(\mathbf{z}) \cdot \frac{q}{m} (\mathbf{E}_j(t) \phi_j(\mathbf{x}) + \mathbf{v} \times \mathbf{B}_j(t) \phi_j(\mathbf{x})) f_k(t) w_k(\mathbf{z}) d\mathbf{z}$$

- Naively, this tensor has $N_c N_p^2$ components, and that's not any better than direct quadrature..
- But we could choose our basis expansion to be a *modal, orthonormal* expansion
- Then these tensors would be sparse and we could do sparse tensor products!
 - Note that for a tensor product basis, this would correspond to a basis of Legendre polynomials

Making the update efficient using Maxima

```
double VlasovVol1x1vTensorP2(const double *w, const double *dxv, const double *EM, const double *f, double *out)
{
// w[NDIM]: Cell-center coordinates. dxv[NDIM]: Cell spacing. EM/f: Input EM-field/distribution function. out: Incremented output
    double dv0dx0 = dxv[1]/dxv[0];
    double w0dx0 = w[1]/dxv[0];
    const double dv10 = 2/dxv[1];
    const double *E0 = &EM[0];
    const double dv1 = dxv[1], wv1 = w[1];

    double abar0[3];

    abar0[0] = E0[0];
    abar0[1] = E0[1];
    abar0[2] = E0[2];

    double incr1[9];

    for(unsigned int i=0; i<9; ++i){

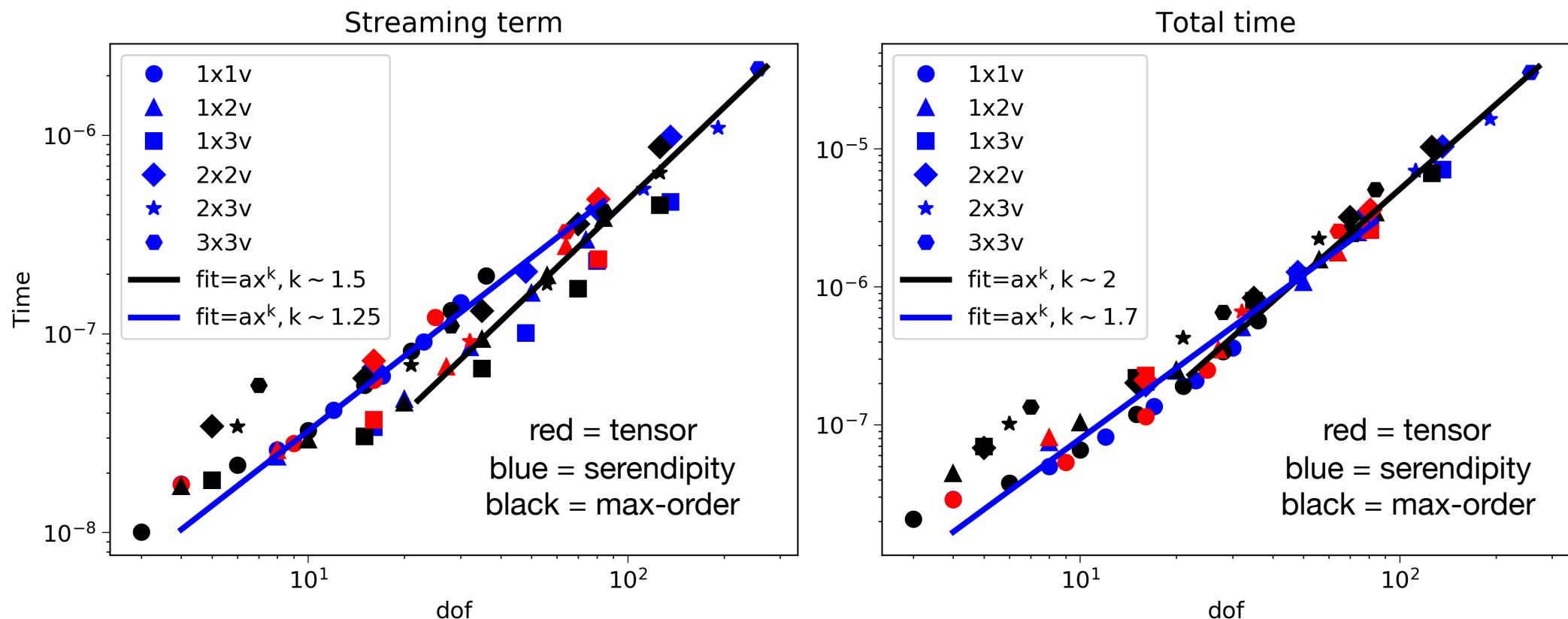
        incr1[i]=0.0;
    };

    const double amid1 = 0.7071067811865475*abar0[0]-0.7905694150420947*abar0[2];
    incr1[2] = 1.224744871391589*abar0[2]*f[4]+1.224744871391589*abar0[1]*f[1]+1.224744871391589*abar0[0]*f[0];
    incr1[3] = abar0[1]*(1.095445115010332*f[4]+1.224744871391589*f[0])+1.095445115010332*f[1]*abar0[2]+1.224744871391589*abar0[0]*f[1];
    incr1[5] = 2.738612787525831*abar0[2]*f[6]+2.738612787525831*abar0[1]*f[3]+2.738612787525831*abar0[0]*f[2];
    incr1[6] = 1.224744871391589*abar0[0]*f[4]+abar0[2]*(0.7824607964359517*f[4]+1.224744871391589*f[0])+1.095445115010332*abar0[1]*f[1];
    incr1[7] = abar0[1]*(2.449489742783178*f[6]+2.738612787525831*f[2])+2.449489742783178*abar0[2]*f[3]+2.738612787525831*abar0[0]*f[3];
    incr1[8] = abar0[2]*(1.749635530559413*f[6]+2.738612787525831*f[2])+2.738612787525831*abar0[0]*f[6]+2.449489742783178*abar0[1]*f[3];

    out[1] += 3.464101615137754*f[0]*w0dx0+f[2]*dv0dx0;
    out[2] += incr1[2]*dv10;
    out[3] += 3.464101615137754*f[2]*w0dx0+incr1[3]*dv10+(0.8944271909999159*f[5]+f[0])*dv0dx0;
    out[4] += 7.745966692414834*f[1]*w0dx0+2.23606797749979*f[3]*dv0dx0;
    out[5] += incr1[5]*dv10;
    out[6] += 7.745966692414834*f[3]*w0dx0+incr1[6]*dv10+(2.0*f[7]+2.23606797749979*f[1])*dv0dx0;
    out[7] += 3.464101615137755*f[5]*w0dx0+incr1[7]*dv10+0.8944271909999161*f[2]*dv0dx0;
    out[8] += 7.745966692414834*f[7]*w0dx0+incr1[8]*dv10+2.0*f[3]*dv0dx0;
    return std::abs(w0dx0)+dv0dx0/2+std::abs(amid1)/dv1;
}
```

- For reference, this update is ~300 multiplications with a nodal basis (now ~70)

Scaling of the update



- Key takeaways
 - This is the scaling of the **full** update, not the update per dimension
 - The cost saving is thus $\sim \frac{dN_q}{N_p} \sim 20$ in 2X3V
 - The reduction in the cost of the streaming term is extremely large
 - A pure Boltzmann solver using an orthonormal model basis may thus be viable

Adkins & Schekochihin 2016

- In the vein of Kraichnan and Batchelor, Adkins and Schekochihin have presented a solvable model of Vlasov-Poisson turbulence in the weakly collisional limit under a number of “brutal” assumptions
 - A stochastic electric field is imposed, i.e. the electromagnetic fields are not self-consistent
 - The driven electric field is single-scaled
 - The driven electric field is a white noise field, i.e. no time correlation

$$\frac{\partial f_e}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_e - \frac{e}{m_e} (\mathbf{E} + \mathbf{E}_{ant}) \cdot \nabla_{\mathbf{v}} f_e = \nu \nabla_{\mathbf{v}} \cdot \left((\mathbf{v} - \mathbf{u}_e) f_e + \frac{T_e}{m_e} \nabla_{\mathbf{v}} f_e \right)$$
$$\frac{\partial \mathbf{E}}{\partial t} = -e \int \mathbf{v} f_e d\mathbf{v} \quad \mathbf{E}_{ant} = \sum_j a_j \cos(\mathbf{k}_j \cdot \mathbf{x} + \phi_j)$$

- The amplitudes and phases of the driver are given by solving a Langevin equation as detailed in TenBarge et al. 2014

$$\frac{da}{dt} = -i\Omega_0 a + \Gamma_0 a + F_a$$

$$F_a = A_0 \sqrt{12 \frac{|\Gamma_0|}{\Delta t}} u_n$$

$$\frac{da}{dt} = \sqrt{8} A_0 u_n$$

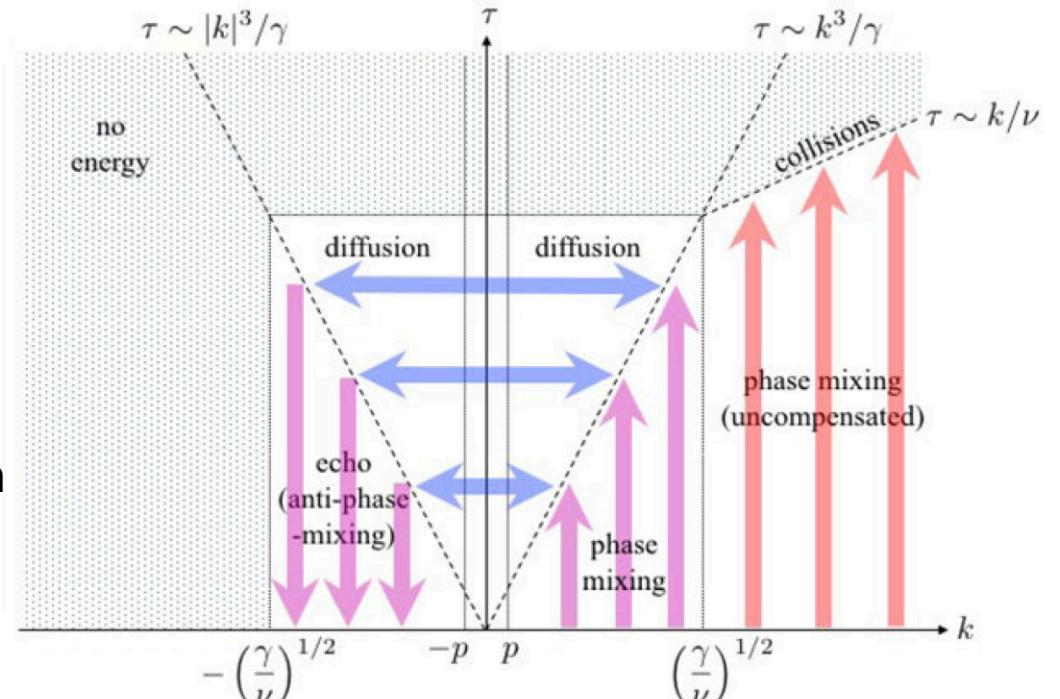
A summary of their results

- There are two important parameters, the collision frequency ν and a parameter related to the driving amplitude $\gamma \sim a_{jRMS}$
- The ratio of these two parameters defines a scale analogous to a Kolmogorov cut-off scale, but for kinetic plasmas
 - In other words, for $k > k_\nu = \left(\frac{\gamma}{\nu}\right)^{\frac{1}{2}}$ analogous to viscosity, there is an efficient route to thermalization
 - This route is simply phase mixing of the distribution function. Prior to this cut-off, the phase mixing of the distribution function is balanced by the nonlinear plasma echo, keeping energy in “large” velocity space scales
- As a result, Adkins and Schekochihin find a steep Hermite spectrum at low wave number

$$\sim k^{-2} m^{-\frac{3}{2}}$$
- And at higher wave numbers past the cut off

$$\sim k^{-2} m^{-\frac{1}{2}}$$
- **Note:** The Hermite transform is analogous to the Fourier transform, but in velocity space — moments of order m correspond to structures with scale

$$\delta v \sim \frac{\pi}{\sqrt{m}}$$

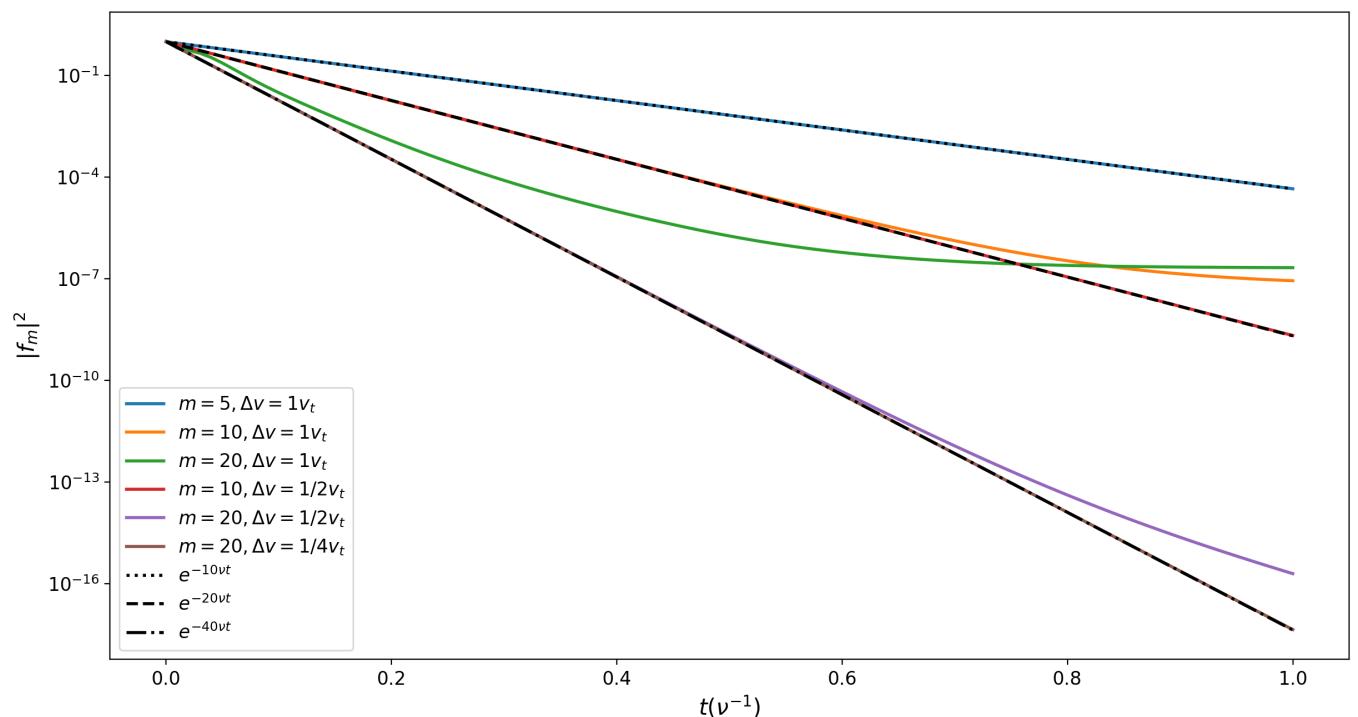


Hermite transform definition

$$f(x, v) = \sum_k \exp(ikx) \sum_m \frac{H_m \left(\frac{v-u}{\sqrt{\frac{2T}{M}}} \right) \exp \left(-M \frac{(v-u)^2}{2T} \right)}{\sqrt{2^m m!}} g_{k,m}$$

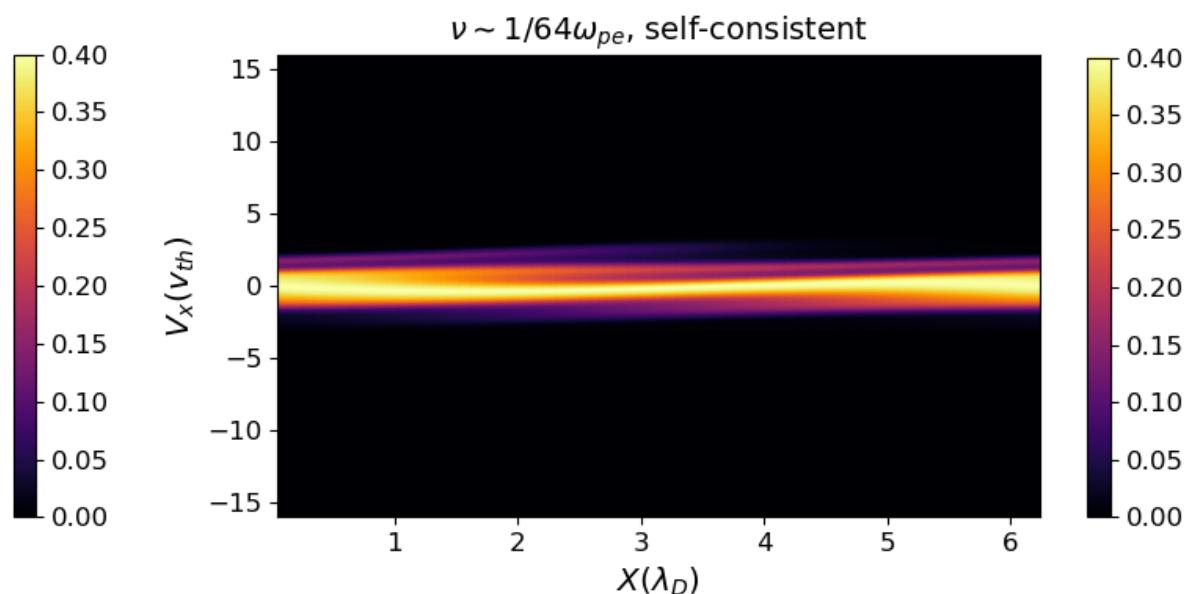
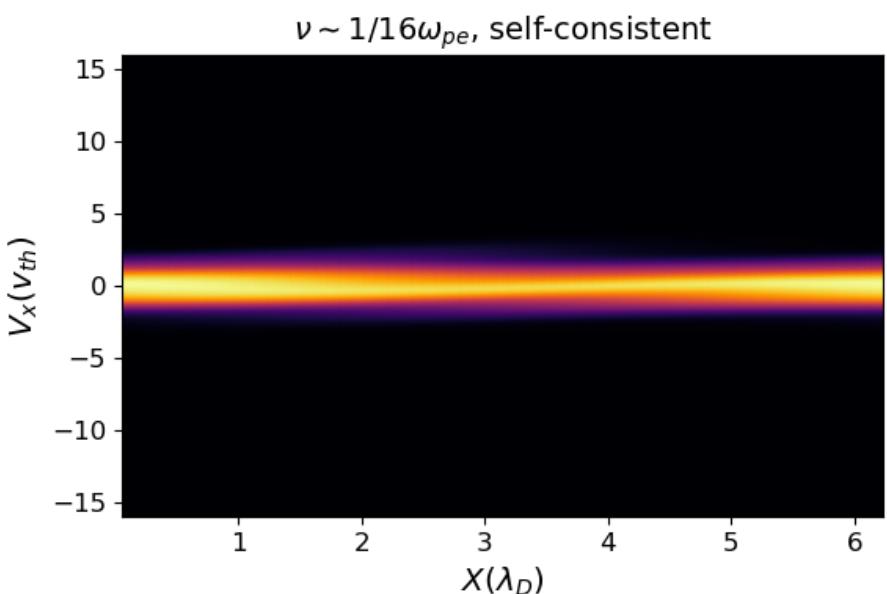
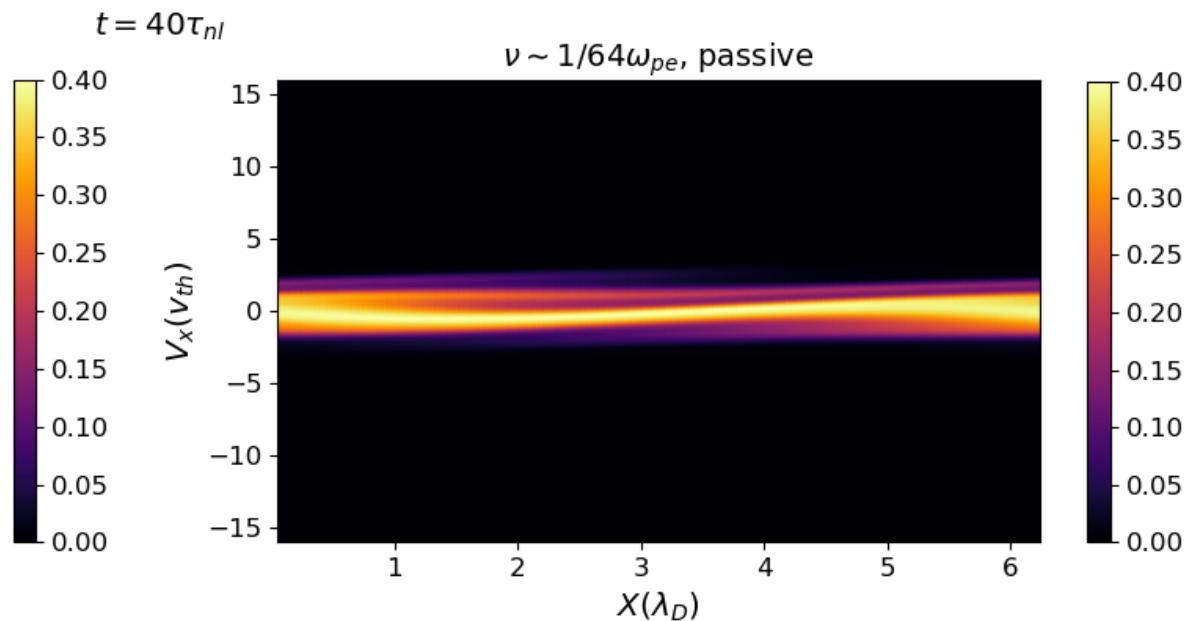
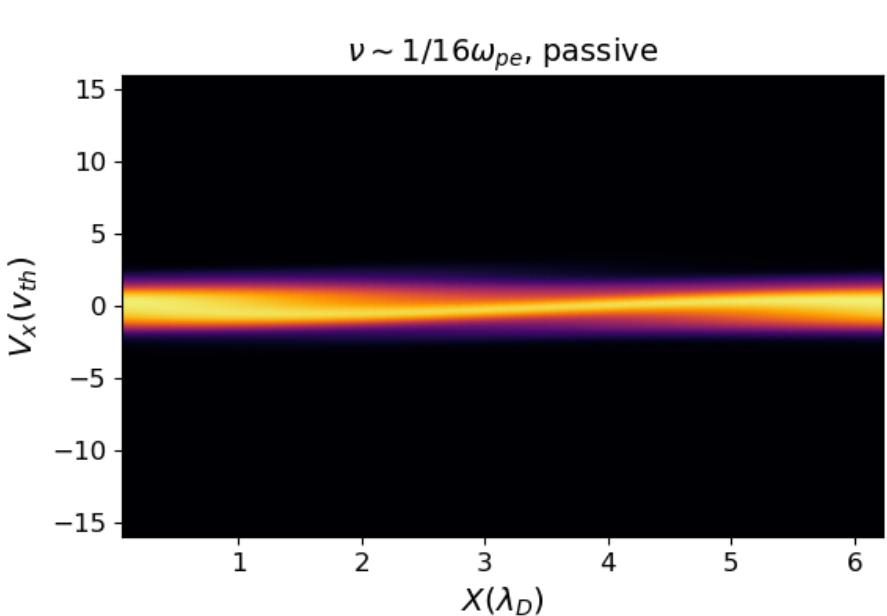
$$g_{k,m} = \int \frac{dx}{2\pi L} \exp(-ikx) \int dv \frac{H_m \left(\frac{v-u}{\sqrt{\frac{2T}{M}}} \right)}{\sqrt{2^m m!}} f(x, v)$$

$$\begin{aligned} \mathcal{F}(\mathcal{H}(C[f(x, v)])) &= \\ &- \nu m g_{k,m} \end{aligned}$$



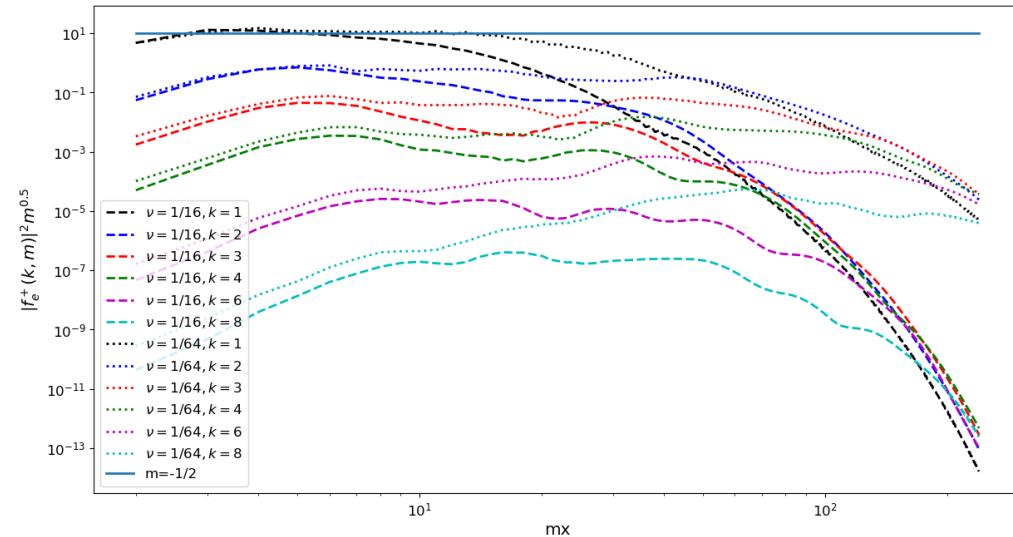
Stochastically driven, passive vs. self-consistent

- 4 simulations, 2 passive and 2 self-consistent.

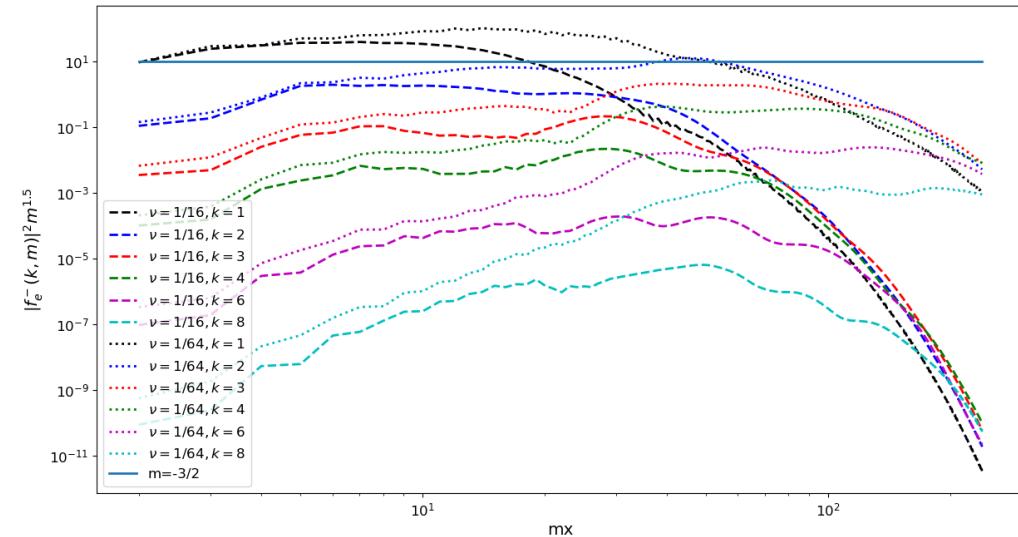


Comparing different collisionalities

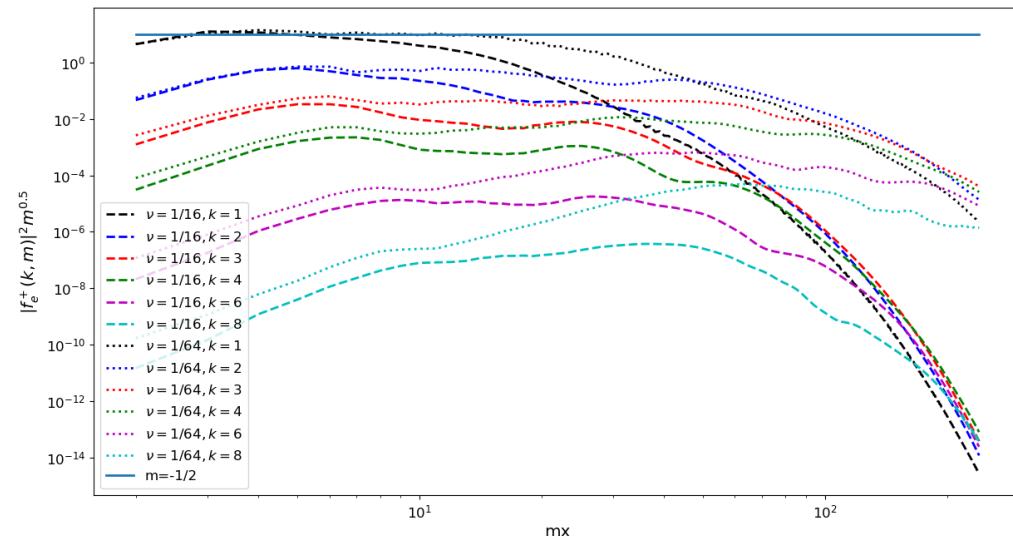
passive phase-mixing part



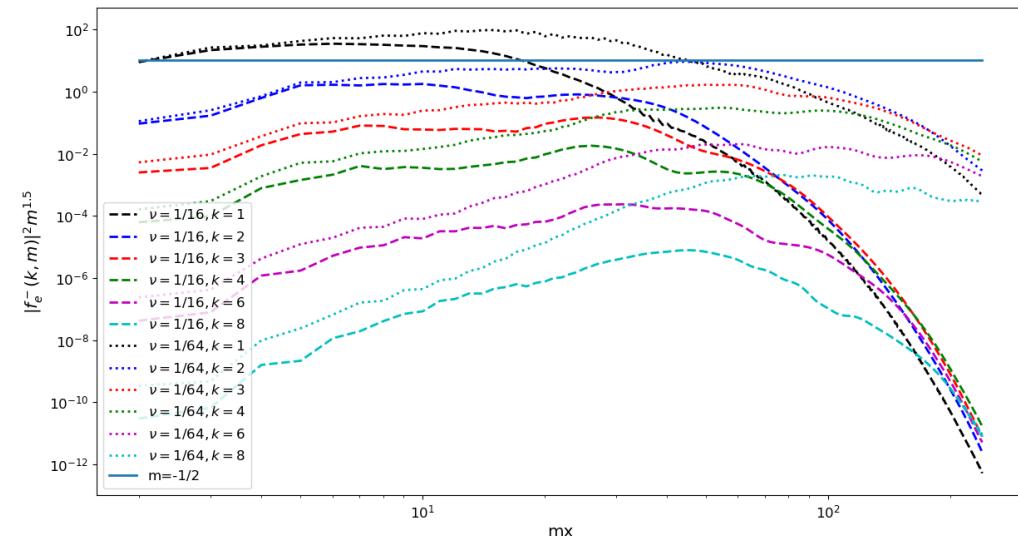
passive echo part



self-consistent phase-mixing part

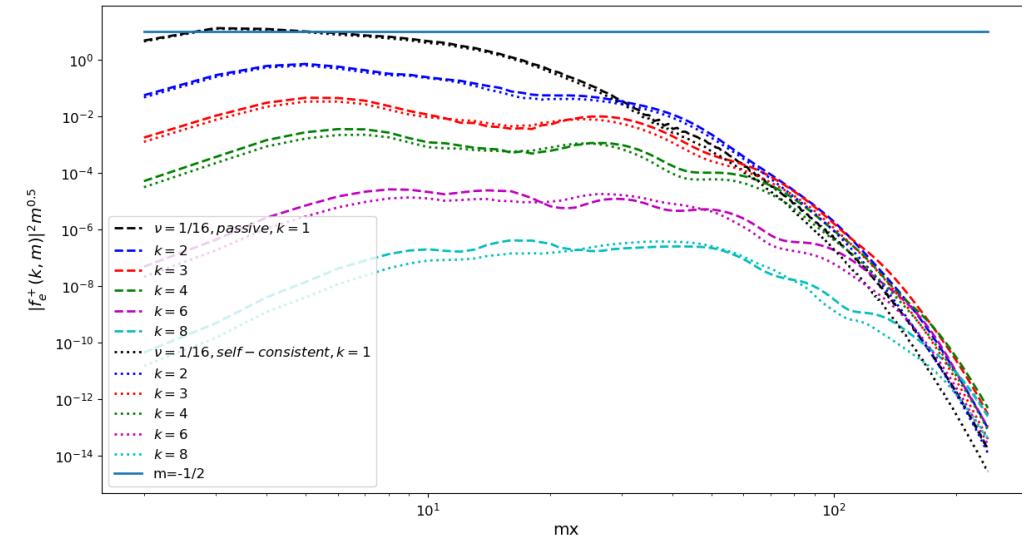


self-consistent echo part

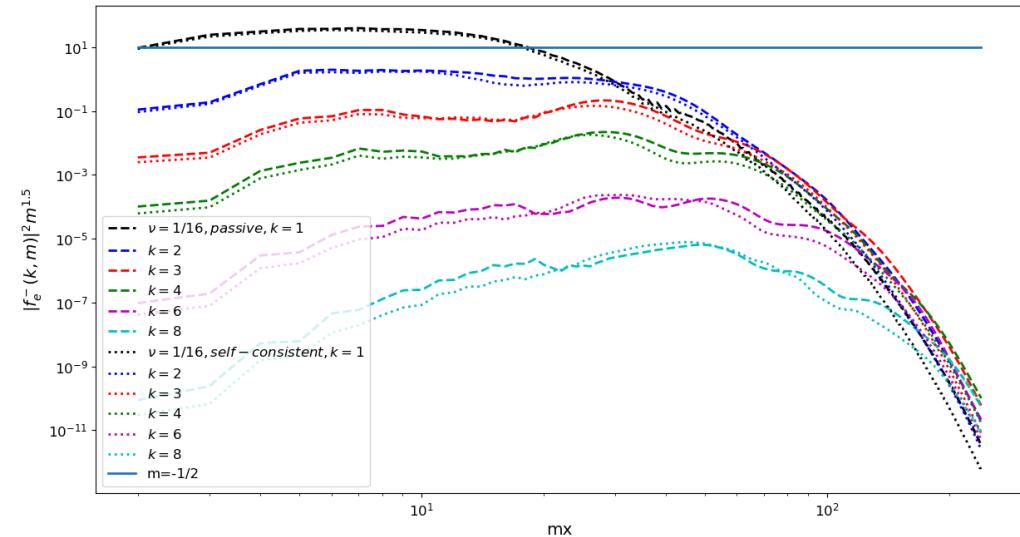


Comparing passive vs. self-consistent

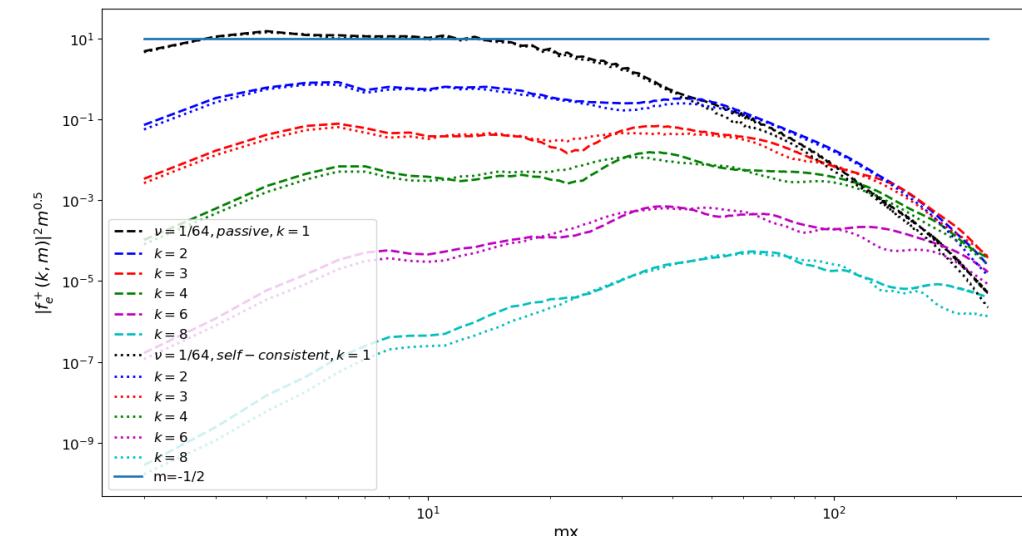
passive phase-mixing part



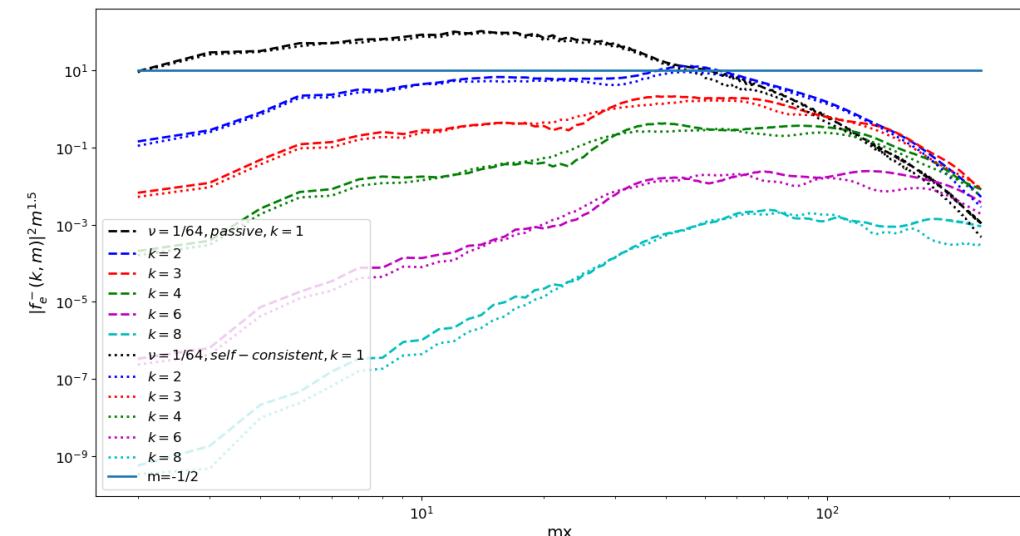
passive echo part



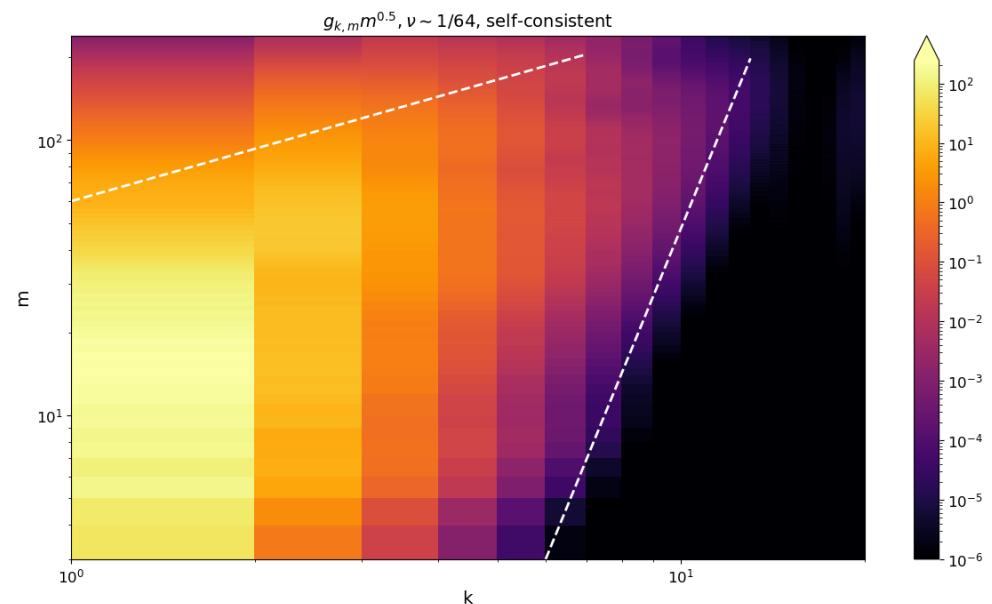
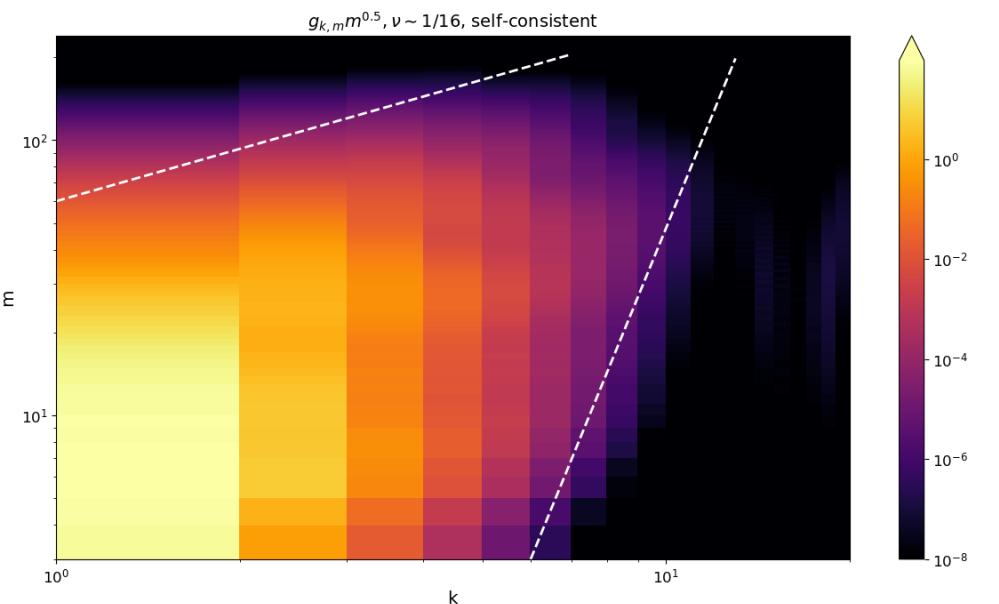
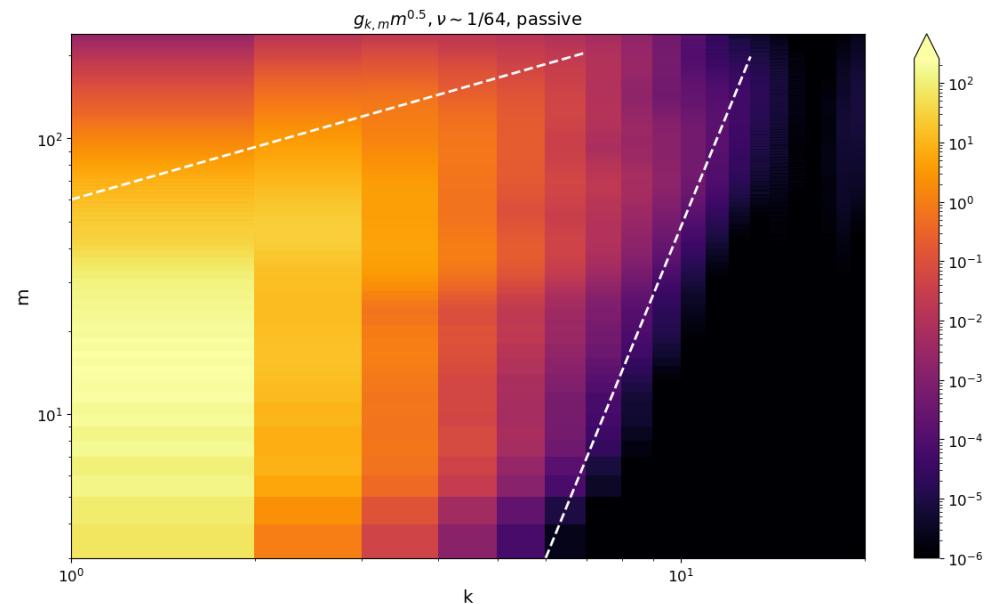
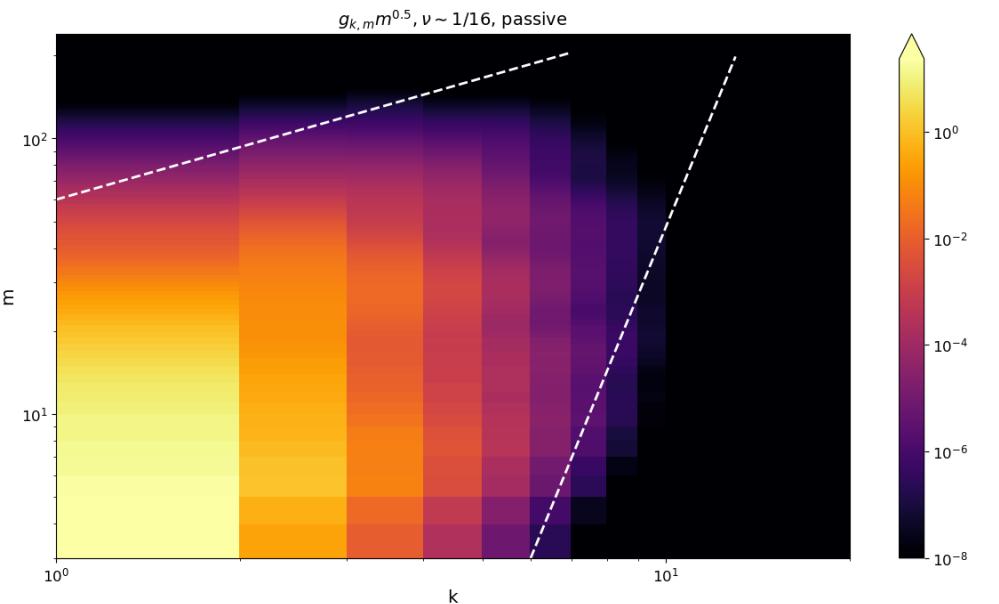
self-consistent phase-mixing part



self-consistent echo part

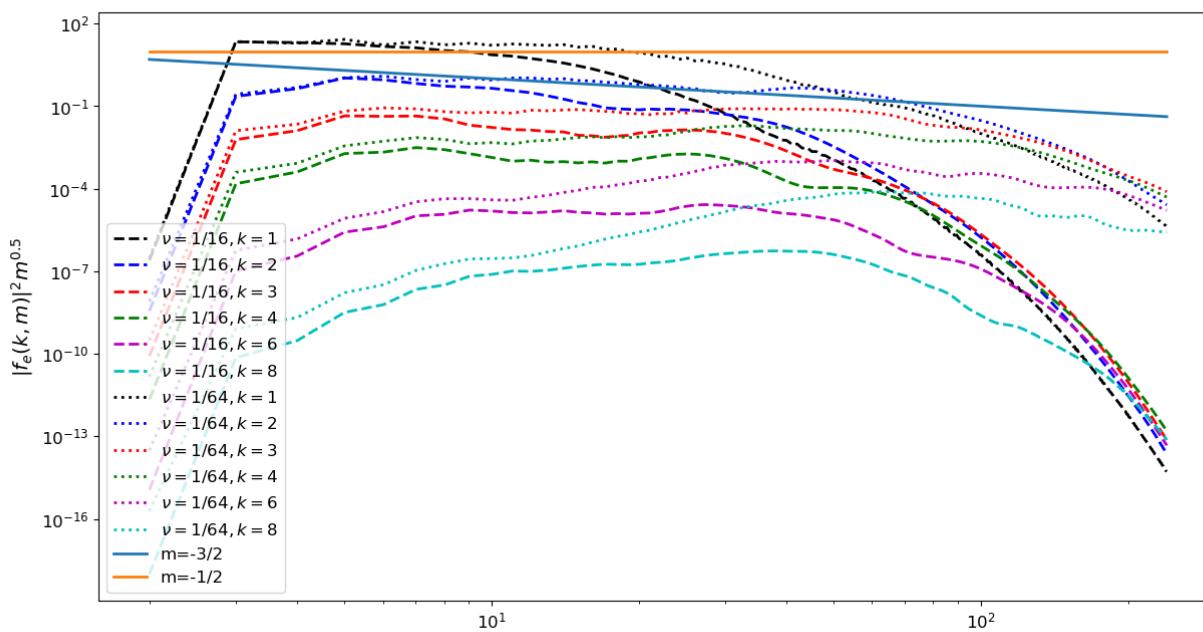


Total Fourier-Hermite transform

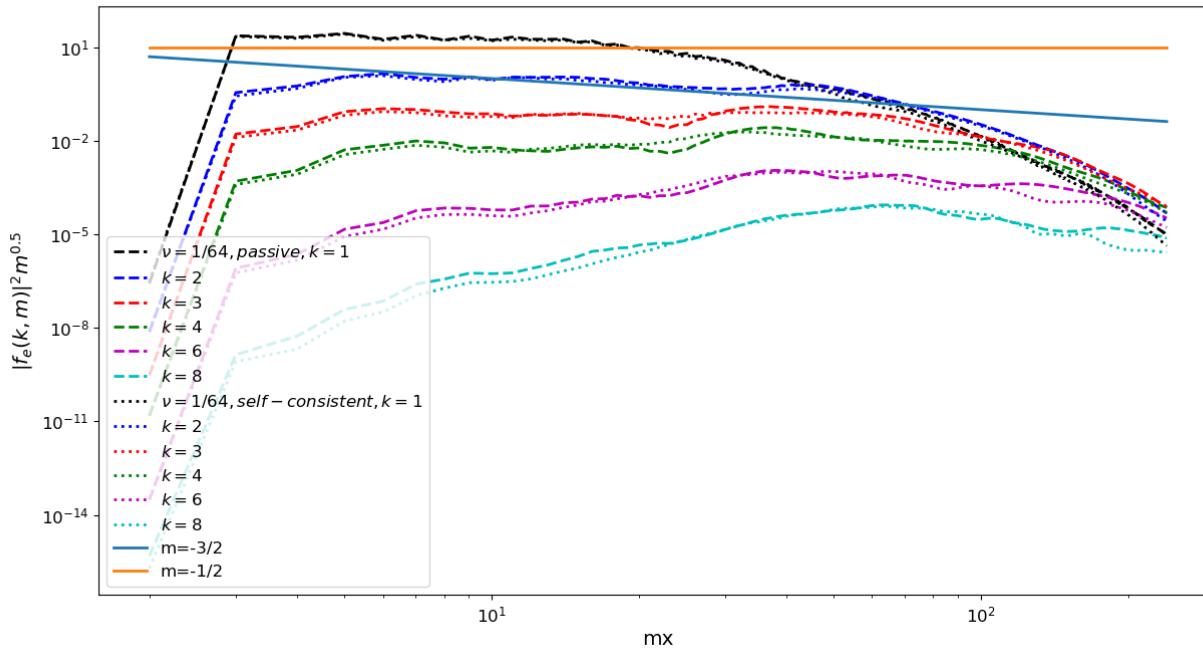


1D Total Fourier-Hermite Spectrum

Collision comparison

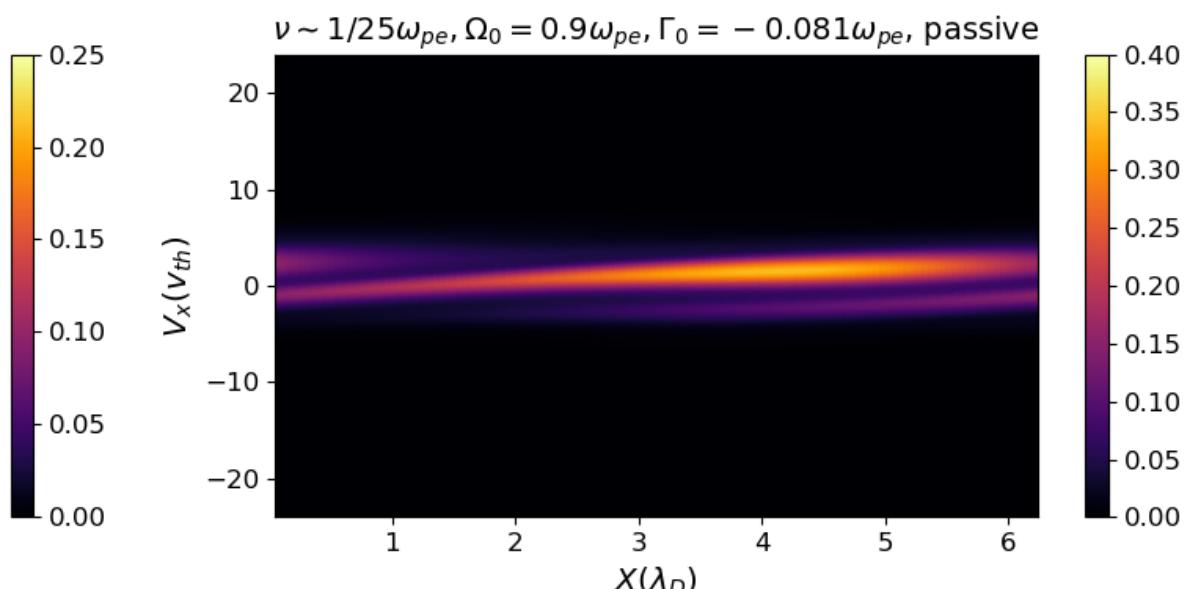
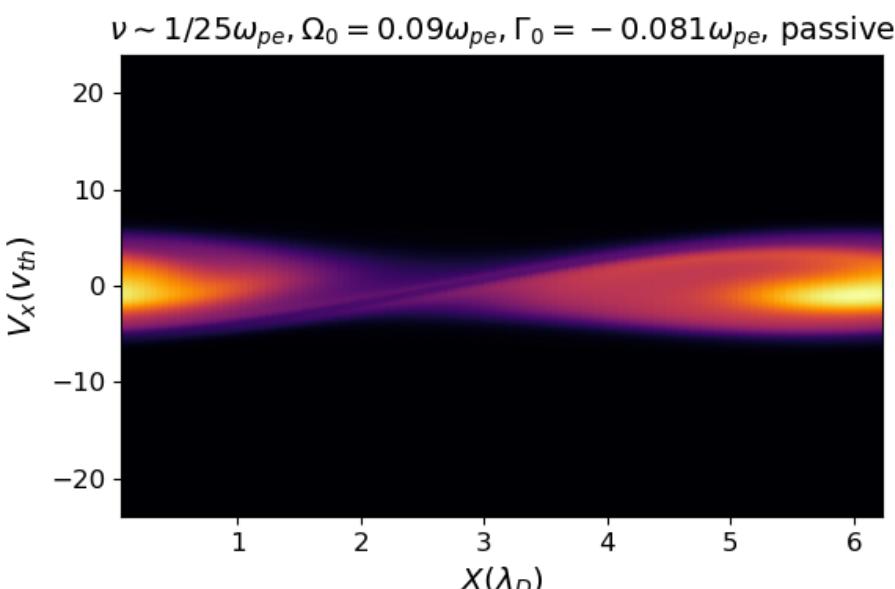
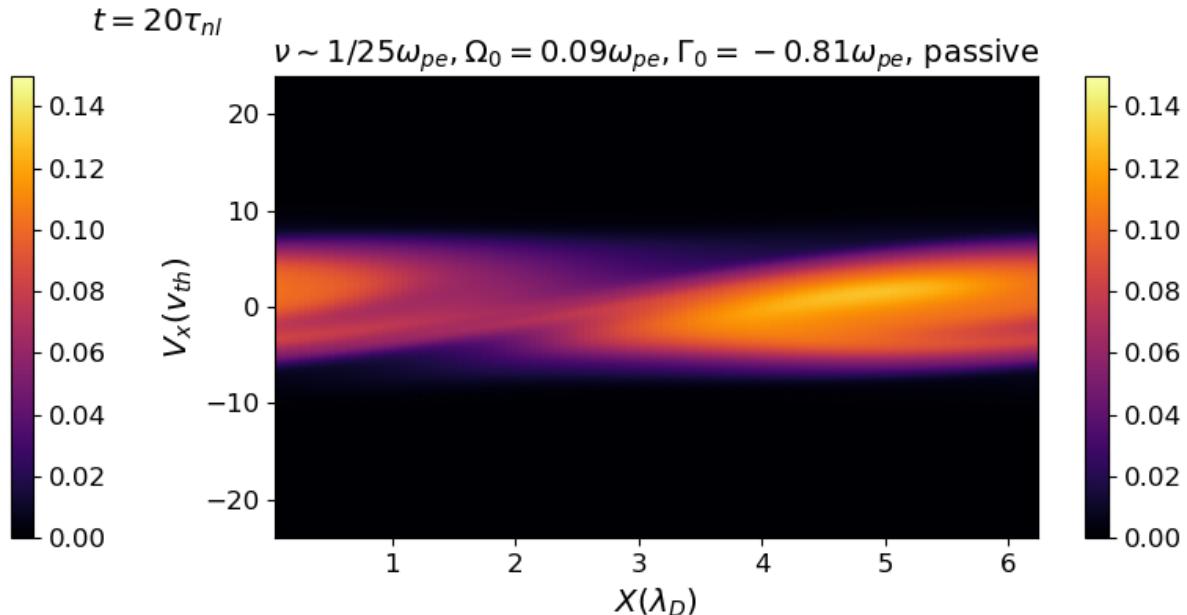
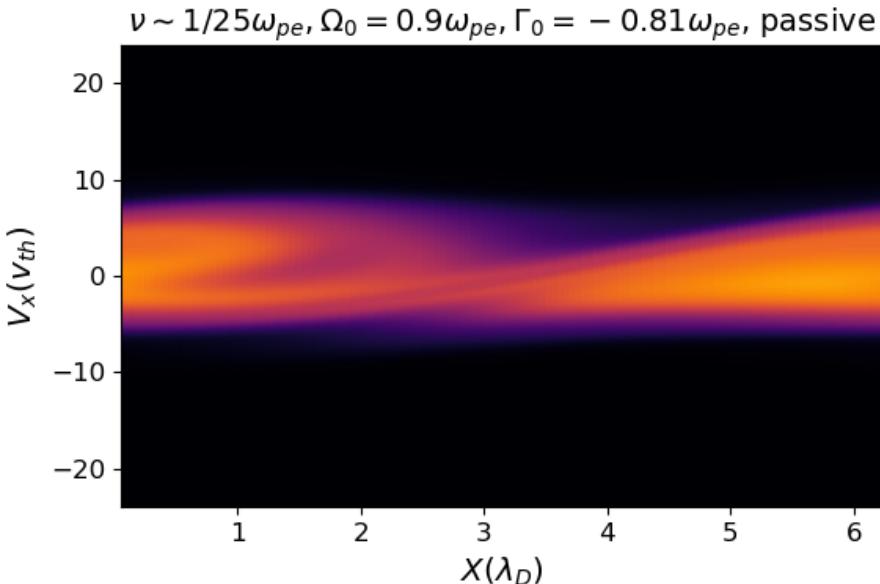


Model comparison

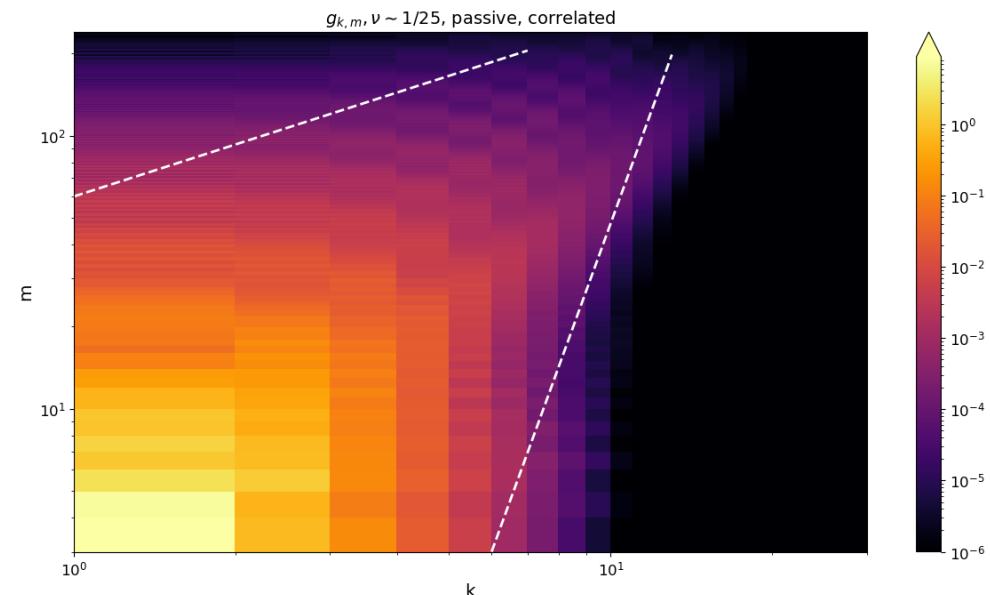
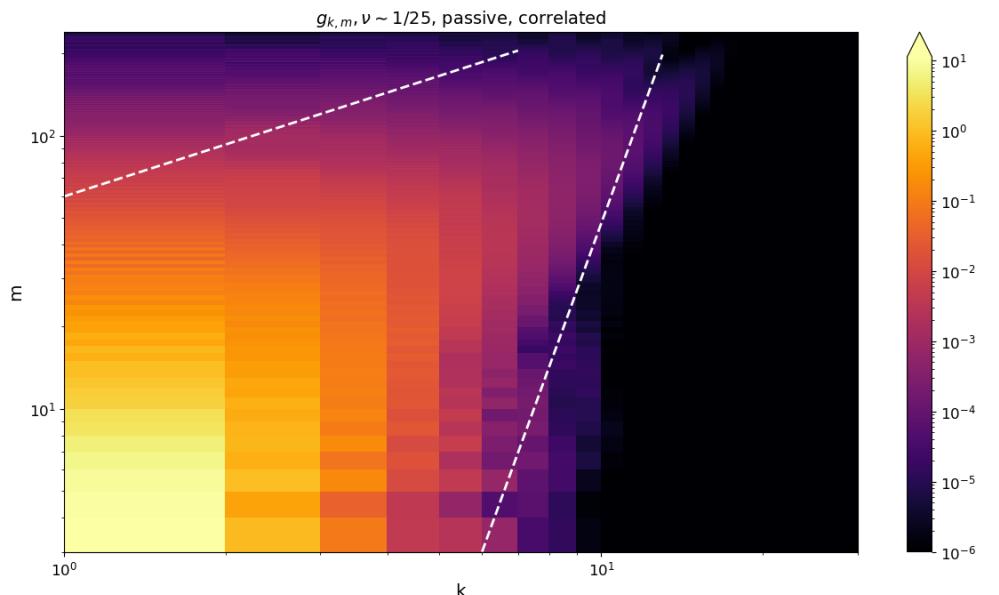
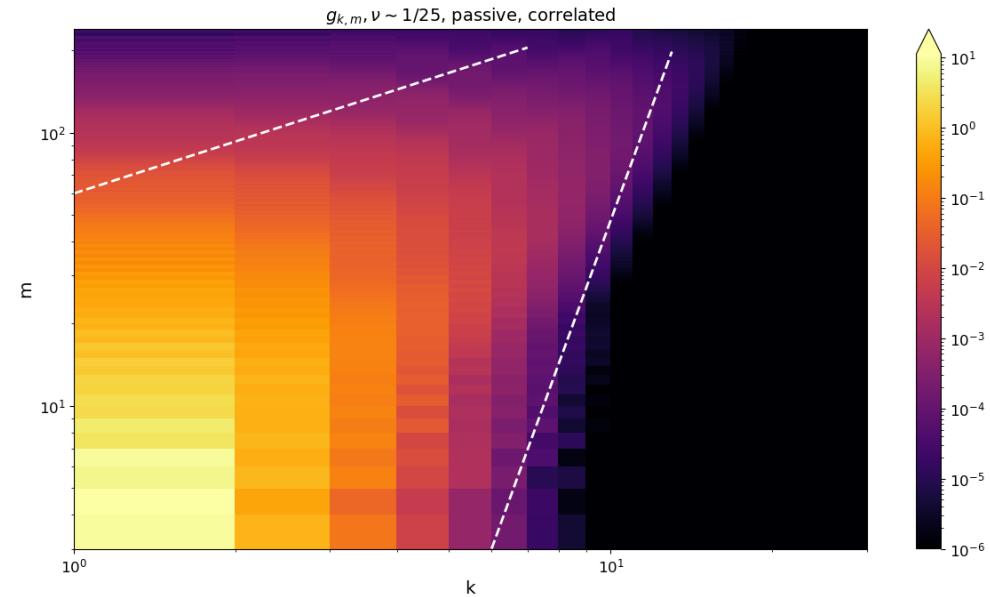
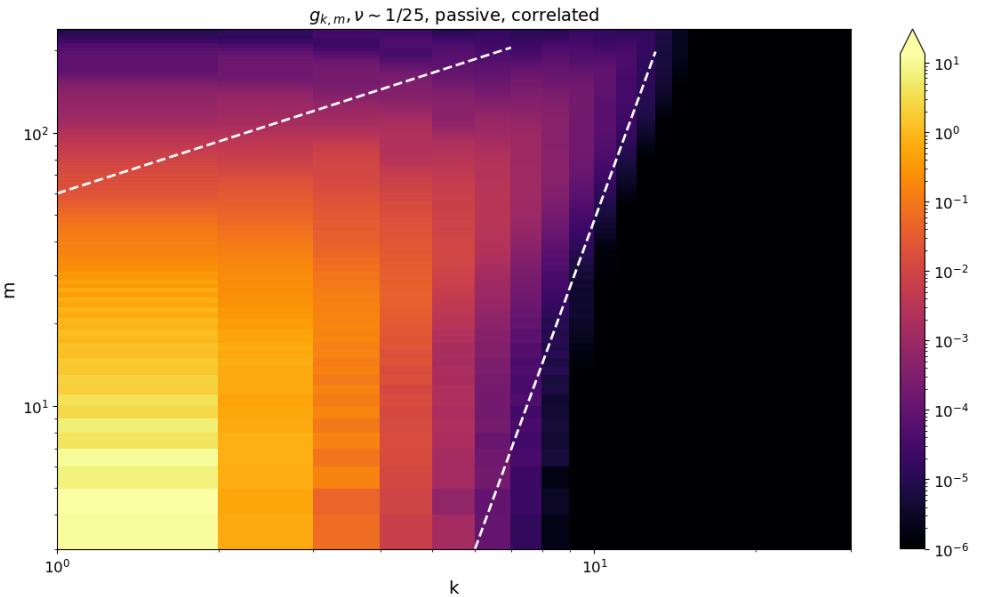


Correlated Driving

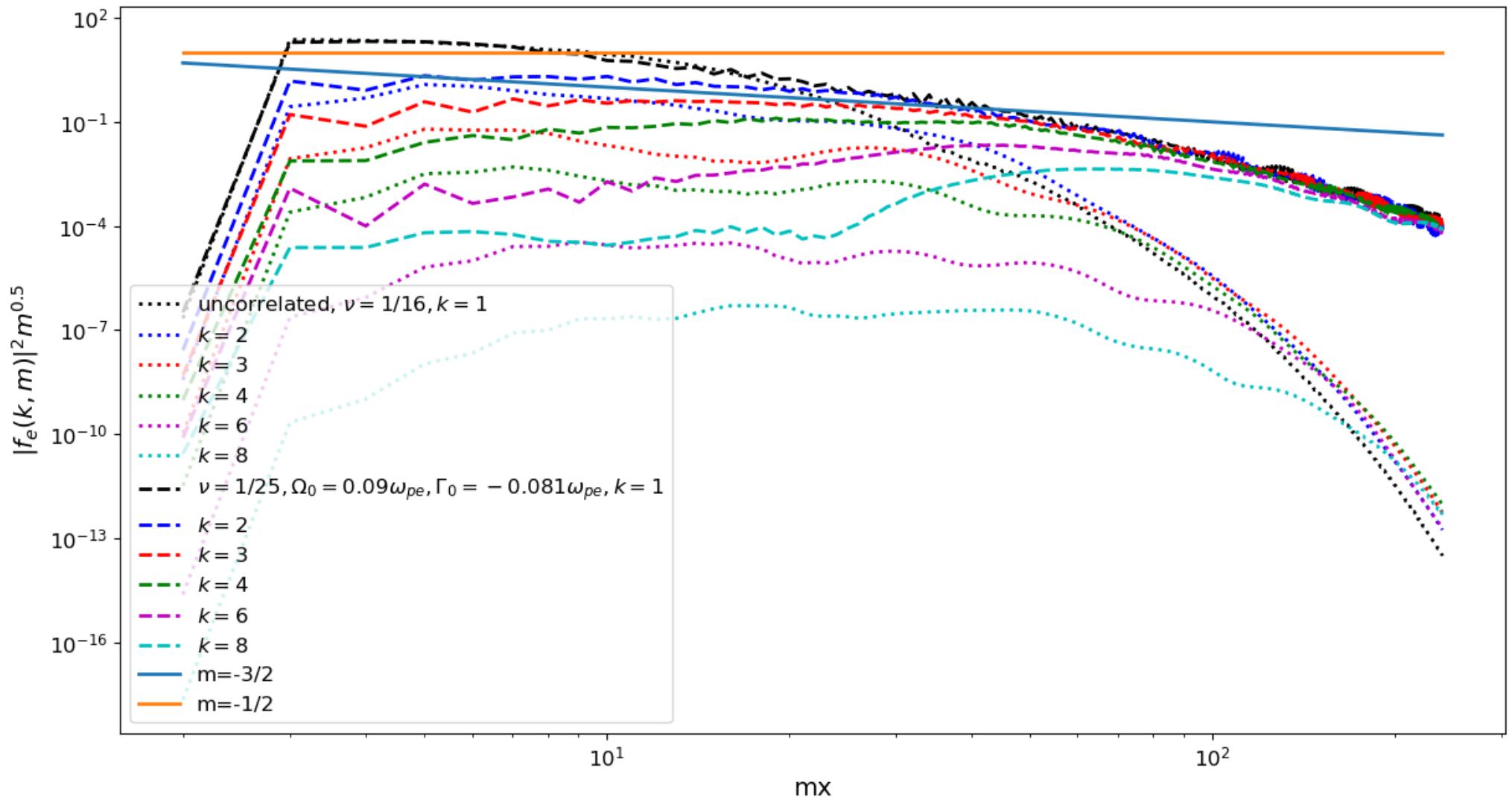
- 4 simulations, varying drive frequency and de-correlation rate



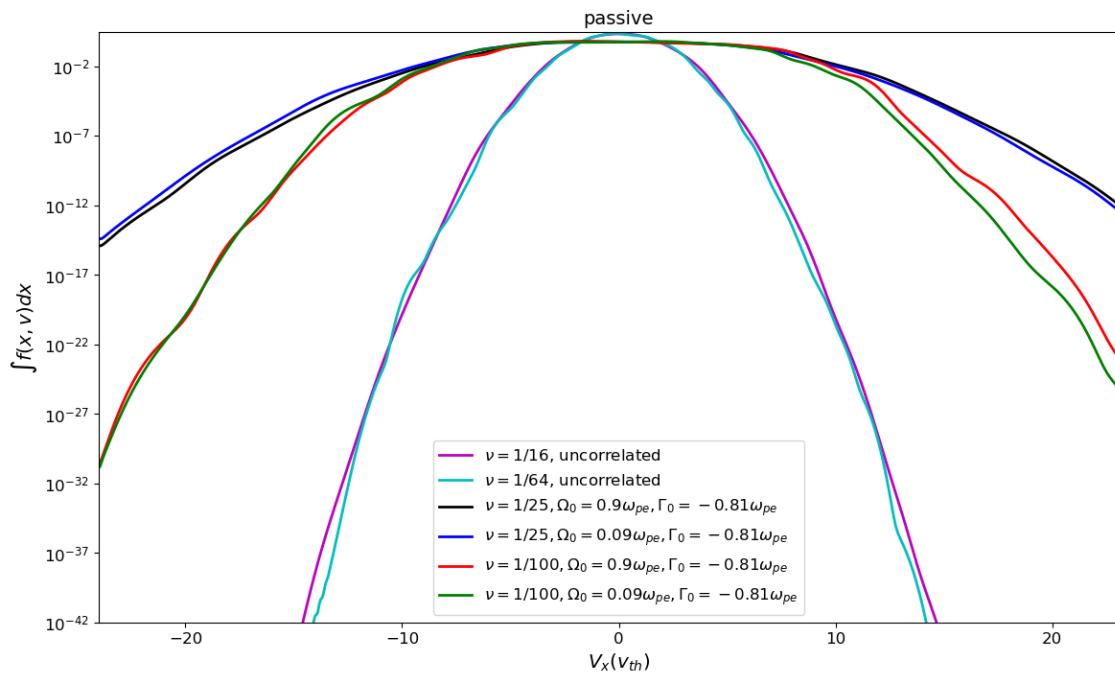
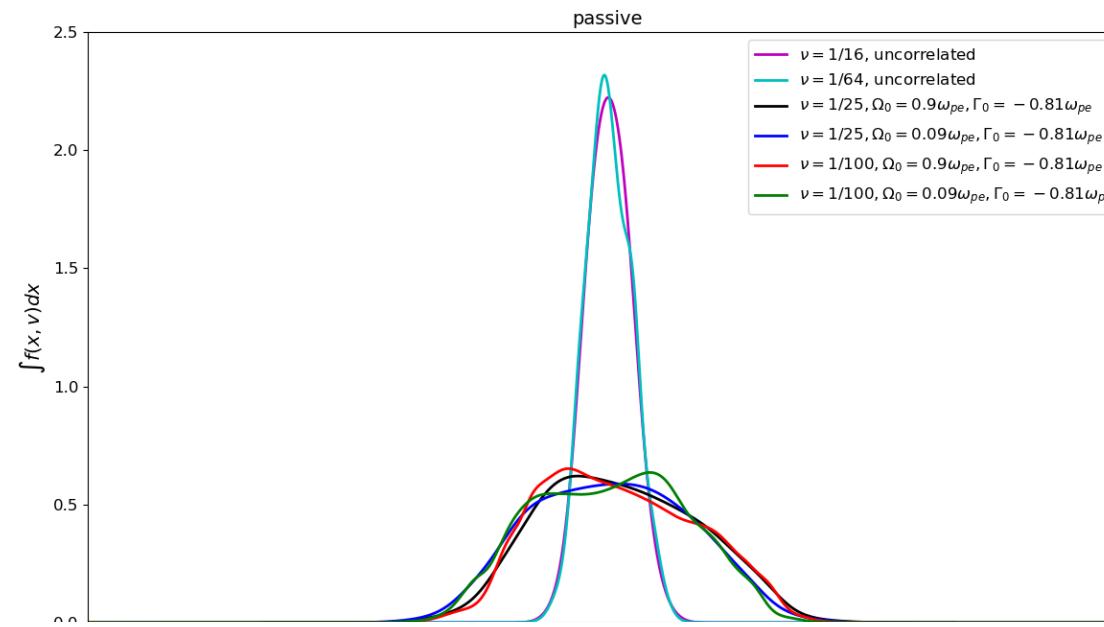
Total Fourier-Hermite transform



Total Fourier-Hermite transform 1D comparison



Correlated Driving, lack of echo?



Summary and Future Outlook

- Adkins & Schekochihin theory is true in the specified limits
 - Theory extends to the regime where the particles can feed back on the fields if the driving is still stochastic
 - Theory seems to break down in the presence of correlated driving if the correlated driver leads to trapped particles -> Trapped particles suppress the echo
- Need to complete the picture with correlated, self-consistent simulations
 - Fine tune parameters to obtain larger dynamic range?