Theory of drift Alfven wave instability and micro-tearing mode

Kaixuan Fan

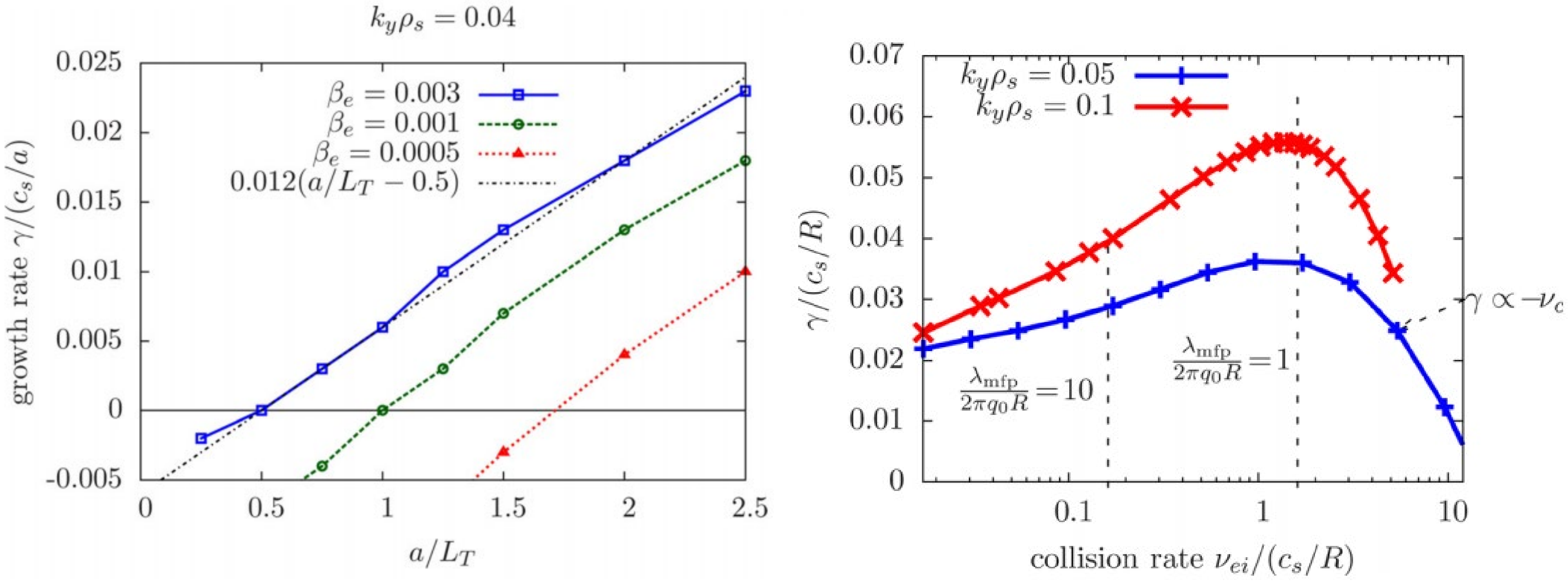
BOUT++ workshop 2023

2023.01.13

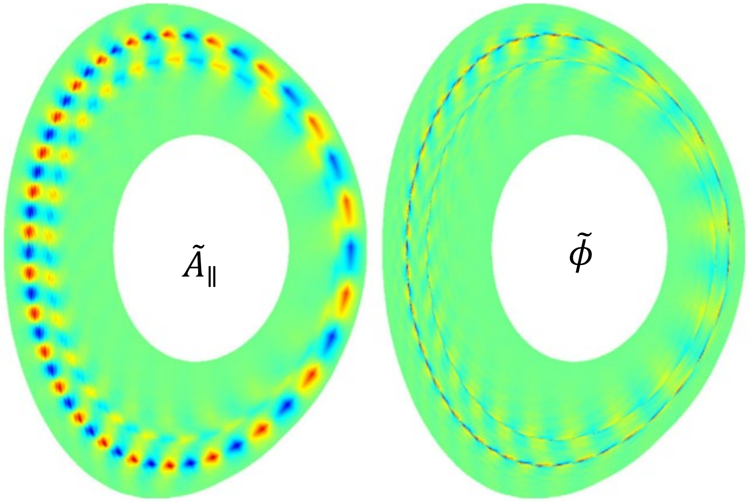
Introduction to micro-tearing mode (MTM)

MTM：

- MTM is a drift-tearing instability and is regarded as a promising driven mechanism of turbulence. It is crucial for the turbulence transport in pedestal region.[1]
- Mode structure has a tearing parity. （$\tilde{\phi}$ is odd, $\tilde{A}_{\parallel}$ is even）
- Enhance the electron thermal transport. MTM can explain the anomalous thermal transport in the experiments. [2-3]
- The growth rate is proportional to the electron temperature gradient and non-linearly related to the collisionality.



Growth rate vs temperature gradient and collisionality. [2]



Poloidal mode structure $m = 24, n = 11$ [2]

[1]T. H. Stix, Phys. Rev. Lett. 30, 833 (1973). [2]H. Doerk, F. Jenko, T. Görler, et al. Phys. Plasmas 19, 055907 (2012); [3]X. Jian, et al, PRL 123,225002 (2019)

The simulation model --- Hassam's model

Ohm's law is replaced by a higher-order electron momentum equation with time-dependent thermal force term included:

$$nm\alpha''\frac{\partial}{\partial t}V_{e\parallel} = en\left(\nabla_\parallel\phi + \frac{\partial}{\partial t}(B_0\psi)\right) - \nabla_\parallel p_e - \alpha n\nabla_\parallel T_e - (en)^2\eta V_{e\parallel} + \frac{\alpha\alpha' n}{\nu_e}\frac{\partial}{\partial t}(\nabla_\parallel T_e)$$

A set of equations for MTM simulation are:

$$nm\alpha''\frac{\partial}{\partial t}V_{e\parallel} = en\left(\nabla_\parallel\phi + \frac{\partial}{\partial t}(B_0\psi)\right) - \nabla_\parallel p_e - \alpha n\nabla_\parallel T_e - (en)^2\eta V_{e\parallel} + \frac{\alpha\alpha' n}{\nu_e}\frac{\partial}{\partial t}(\nabla_\parallel T_e)$$

$$\frac{\partial}{\partial t}n_i = -\left(\frac{1}{B}\hat{\boldsymbol{b}}\times\nabla_\perp\phi + V_{\parallel i}\hat{\boldsymbol{b}}\right)\cdot\nabla n_i - \frac{2n_i}{B}\hat{\boldsymbol{b}}\times\boldsymbol{\kappa}\cdot\nabla_\perp\phi + \frac{2}{ZeB}\hat{\boldsymbol{b}}\times\boldsymbol{\kappa}\cdot\nabla_\perp P_e - n_i B\nabla_\parallel\left(\frac{V_{\parallel i}}{B}\right) + \frac{B}{Ze}\nabla_\parallel\left(\frac{J_\parallel}{B}\right) + S_n,$$

$$\frac{\partial}{\partial t}\varpi = -\left(\frac{1}{B}\hat{\boldsymbol{b}}\times\nabla_\perp\phi + V_{\parallel i}\hat{\boldsymbol{b}}\right)\cdot\nabla\varpi + B^2\nabla_\parallel\left(\frac{J_\parallel}{B}\right) + 2\hat{\boldsymbol{b}}\times\boldsymbol{\kappa}\cdot\nabla P - \frac{2}{3}\hat{\boldsymbol{b}}\times\boldsymbol{\kappa}\cdot\nabla\pi_{ci}$$
$$- \frac{1}{2\Omega_i}\left[n_i Ze\boldsymbol{V}_{\boldsymbol{D_i}}\cdot\nabla\left(\nabla_\perp^2\phi\right) - m_i\Omega_i\hat{\boldsymbol{b}}\times\nabla n_i\cdot\nabla V_E^2\right] + \frac{1}{2\Omega_i}\left[\boldsymbol{V}_{\boldsymbol{E}}\cdot\nabla\left(\nabla_\perp^2 P_i\right) - \nabla_\perp^2\left(\boldsymbol{V}_{\boldsymbol{E}}\cdot\nabla P_i\right)\right],$$

$$\frac{\partial}{\partial t}T_i = -\left(\frac{1}{B}\hat{\boldsymbol{b}}\times\nabla_\perp\phi + V_{\parallel i}\hat{\boldsymbol{b}}\right)\cdot\nabla T_i + \frac{2}{3n_i k_B}\nabla_\parallel q_{\parallel i} + \frac{2m_e}{m_i}\frac{Z}{\tau_e}(T_e - T_i) - \frac{2}{3}T_i\left[\left(\frac{2}{B}\hat{\boldsymbol{b}}\times\boldsymbol{\kappa}\right)\cdot\left(\nabla\phi + \frac{1}{Zen_i}\nabla P_i + \frac{5}{2}\frac{k_B}{Ze}\nabla T_i\right) + B\nabla_\parallel\left(\frac{V_{\parallel i}}{B}\right)\right] - \frac{2\pi_{ci}}{9k_B n_i}\left[\frac{2}{\sqrt{B}}\nabla_\parallel\left(\sqrt{B}V_{\parallel i}\right) - \frac{k_B}{Zen_i B}\hat{\boldsymbol{b}}\cdot\nabla n_i\times\nabla T_i\right] - \frac{4}{3\Omega_i}T_i V_{\parallel i}\hat{\boldsymbol{b}}\times\boldsymbol{\kappa}\cdot\nabla V_{\parallel i} + \frac{2S_i^E}{3n_i} - \frac{T_i S_n}{n_i},$$

$$\frac{\partial}{\partial t}T_e = -\left(\frac{1}{B}\hat{\boldsymbol{b}}\times\nabla_\perp\phi + V_{\parallel e}\hat{\boldsymbol{b}}\right)\cdot\nabla T_e + \frac{2}{3n_e k_B}\nabla_\parallel q_{\parallel e} + 0.71\frac{2T_e}{3en_e}B\nabla_\parallel\left(\frac{J_\parallel}{B}\right) - \frac{2m_e}{m_i}\frac{1}{\tau_e}(T_e - T_i)$$
$$- \frac{2}{3}T_e\left[\left(\frac{2}{B}\hat{\boldsymbol{b}}\times\boldsymbol{\kappa}\right)\cdot\left(\nabla\phi - \frac{1}{en_e}\nabla P_e - \frac{5}{2}\frac{k_B}{e}\nabla T_e\right) + B\nabla_\parallel\left(\frac{V_{\parallel e}}{B}\right)\right] + \frac{2}{3n_e k_B}\eta_\parallel J_\parallel^2 + \frac{2S_e^E}{3n_e} - \frac{T_e S_n}{n_e}.$$

The dispersion relation of MTM

According to Hassam's model, the dispersion relation is

$$\omega^2\left[\omega-\omega_{*n}-(1+\alpha)\omega_{*T}-i\alpha\alpha'\omega\omega_{*T}\nu_{ei}^{-1}\right]^3=i\left(\frac{\eta\Delta'}{4\pi\Delta_c}\right)^5$$

In a slab geometry, we derived local dispersion relation from 5f Eqs,

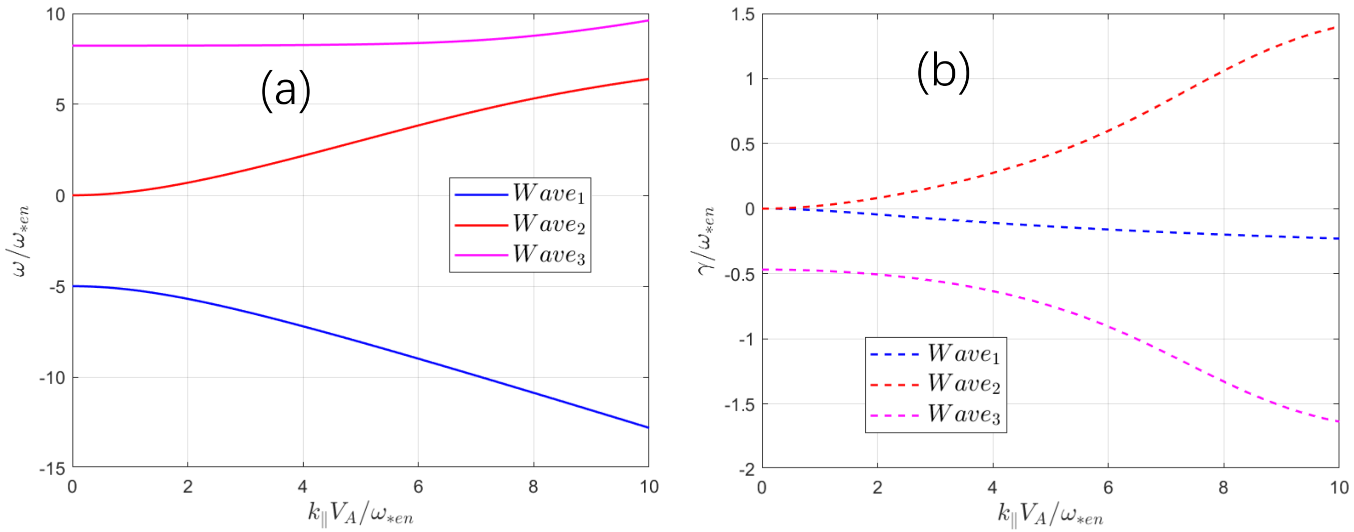
$$-\frac{i\eta}{\mu_0}k_\perp^2\omega(\omega-\omega_{*i})=\frac{\omega-\omega_{*en}-\omega_{*eT}-\alpha\omega_{*eT}(1+\alpha' i\omega/\nu_{ei})}{1-i\alpha'' s\omega/\nu_{ei}}\left(\omega^2-\omega\omega_{*i}-k_\parallel^2V_A^2\right)\quad(1)$$

The MTM is coupled with DAW.

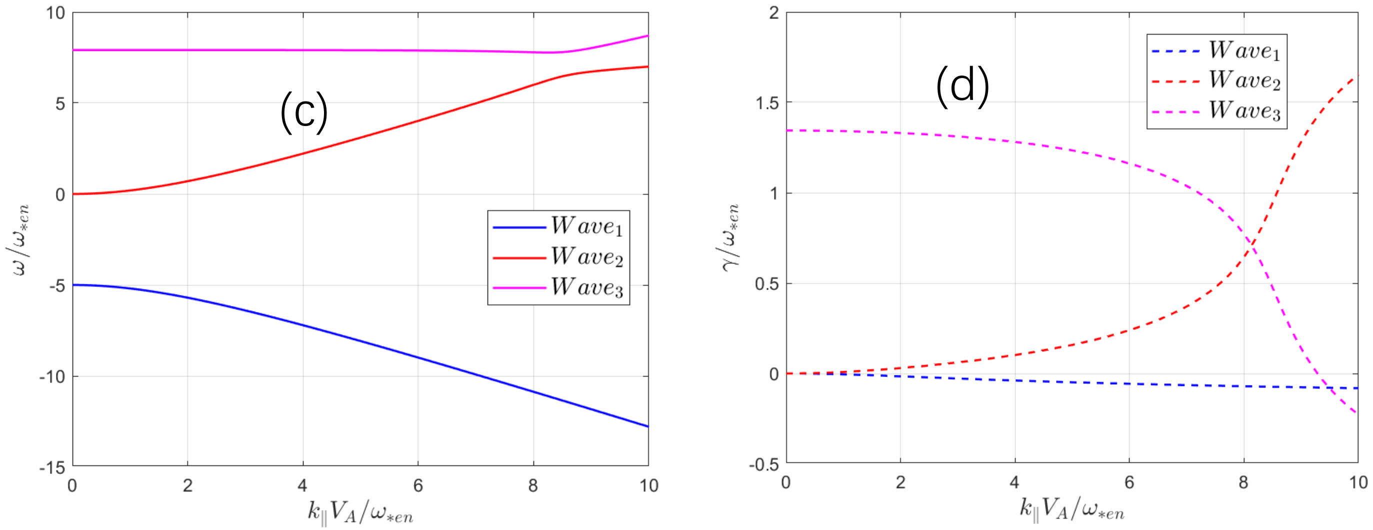
If neglected the time-dependent thermal force term and electron inertial term, the equation reduces to the dispersion relation of drift Alfven instability.

$$-\frac{i\eta}{\mu_0}k_\perp^2\omega(\omega-\omega_{*i})=(\omega-\omega_{*en}-\omega_{*eT}-\alpha\omega_{*eT})\left(\omega^2-\omega\omega_{*i}-k_\parallel^2V_A^2\right)\quad(2)$$

Numerical calculations



Panels (a) and (b) represent the frequency and growth rate vs $k_{\parallel}$ that solved from dispersion relation (2). It is DAW instability



Panels (c) and (d) represent the frequency and growth rate vs $k_{\parallel}$ that solved from dispersion relation (1). It is DAW instability modulated by time-dependent thermal force.

How to implement Hassam's model in the simulation

The evolution equation of Ohm's law is,

$$en_e \frac{\partial}{\partial t}\left(B_0\psi - \frac{\alpha'' m_e}{e} V_{e\parallel} + \frac{\alpha\alpha'}{e\nu_e}\nabla_\parallel T_e\right) = -en_e\nabla_\parallel\phi + \nabla_\parallel p_e + \alpha n_e \nabla_\parallel T_e - en_e \eta j_\parallel$$

$$\frac{\partial}{\partial t}A^* = -\frac{1}{B_0}\nabla_\parallel \phi + \frac{1}{en_e B_0}\nabla_\parallel p_e + \frac{\alpha}{eB_0}\nabla_\parallel T_e - \frac{1}{B_0}\eta j_\parallel \qquad A^* = \psi - \alpha''\frac{m_e}{eB_0}V_{e\parallel} + \frac{\alpha\alpha'}{eB_0\nu_e}\nabla_\parallel T_e$$

The MTM is supposed to be driven up by the time-dependent thermal force

Since we do not evolve ψ directly, ψ is solved from A^* :

$$\nabla_\perp^2\psi - \frac{e^2 n_e \mu_0}{\alpha'' m_e}\psi = -\frac{e^2 n_e \mu_0}{\alpha'' m_e}A^* + \frac{\alpha\alpha'}{B_0\nu_e}\frac{en_e\mu_0}{\alpha'' m_e}\nabla_\parallel T_e \qquad \nabla_\parallel T_e = \nabla_{\parallel 0}T_{e1} - b_0\times\nabla\psi\cdot\nabla T_{e0}$$

$$\nabla_\perp^2\hat{\psi} - \frac{e^2\bar{n}\mu_0}{\alpha'' m_e}\hat{n}_0\hat{\psi} + \alpha\alpha'\frac{k_B\bar{T}_e\bar{n}\mu_0}{\bar{B}}\frac{e\hat{n}_0}{\hat{B}_0\alpha'' m_e\nu_e}b_0\times\nabla\hat{\psi}\cdot\nabla\hat{T}_{e0} = -\frac{e^2\bar{n}\mu_0}{\alpha'' m_e}\hat{n}_0\hat{\psi}^*\alpha\alpha'\frac{k_B\bar{T}_e\bar{n}\mu_0}{\bar{B}\bar{L}}\frac{e\hat{n}_0}{\hat{B}_0\alpha'' m_e\nu_e}\nabla_{\parallel 0}\hat{T}_{e1}$$

The ‘invert_laplace’ and the ‘invert_laplace_MTM’ solver

To deal with the MTM simulation issue, we want to solve the Eq. like this,

$$d\nabla_{\perp}^{2}x + ax + \boldsymbol{b_0} \times \nabla x \cdot \nabla c = b$$

Here, $\boldsymbol{b_0}$ is the direction of the magnetic line

The ‘invert_laplace’ solver is designed to solve x from a equation:

$$d\nabla_{\perp}^{2}x + ax + \frac{1}{\mathrm{c}}\nabla x \cdot \nabla c = b$$

So, we want to develop a new solver based on the ‘invert_laplace’ --- ‘invert_laplace_MTM’

Coding for ‘invert_laplace_MTM’ solver

Imitate the ‘invert_laplace’ solver to create a new class.

First see how ‘invert_laplace’ solver works,

$$d\nabla_\perp^2 f + \frac{1}{c_1}(\nabla_\perp c_2)\cdot\nabla_\perp f + af = b$$

$$d(g^{xx}\partial_x^2 + G^x\partial_x + g^{zz}\partial_z^2 + G^z\partial_z + 2g^{xz}\partial_x\partial_z)f$$
$$+\frac{1}{c_1}(\mathbf{e}^x\partial_x + \mathbf{e}^z\partial_z)c_2\cdot(\mathbf{e}^x\partial_x + \mathbf{e}^z\partial_z)f$$
$$+af = b$$

Fourier transformation

$$d(g^{xx}\partial_x^2 F_z + G^x\partial_x F_z + g^{zz}[ik]^2 F_z + G^z[ik]F_z + 2g^{xz}\partial_x[ik]F_z)$$
$$+\frac{1}{c_1}(\mathbf{e}^x\partial_x c_2)\cdot(\mathbf{e}^x\partial_x F_z + \mathbf{e}^z ikF_z)$$
$$+aF_z = B_z$$

$$d\left(g^{xx}\frac{F_{z,n-1} - 2F_{z,n} + F_{z,n+1}}{dx^2} + G^x\frac{-F_{z,n-1} + F_{z,n+1}}{2dx} - k^2 g^{zz}F_{z,n}\right.$$
$$\left. + ikG^z F_{z,n} + ik2g^{xz}\frac{-F_{z,n-1} + F_{z,n+1}}{2dx}\right)$$
$$+\frac{1}{c_1}\left(\frac{-c_{2,n-1} + c_{2,n+1}}{2dx}\right)\left(g^{xx}\frac{-F_{z,n-1} + F_{z,n+1}}{2dx} + g^{xz}ikF_{z,n}\right)$$
$$+aF_{z,n} = B_{z,n}$$

$$\left(\frac{dg^{xx}}{dx^2} - \frac{dG^x}{2dx} - \frac{g^{xx}}{c_{1,n}}\frac{-c_{2,n-1} + c_{2,n+1}}{4dx^2} - i\frac{dk2g^{xz}}{2dx}\right)F_{z,n-1}$$
$$+\left(-\frac{dg^{xx}}{dx^2} - dk^2 g^{zz} + a + idkG^z + i\frac{g^{xz}}{c_{1,n}}\frac{-c_{2,n-1} + c_{2,n+1}}{2dx}k\right)F_{z,n}$$
$$+\left(\frac{dg^{xx}}{dx^2} + \frac{dG^x}{2dx} + \frac{g^{xx}}{c_{1,n}}\frac{-c_{2,n-1} + c_{2,n+1}}{4dx^2} + i\frac{dk2g^{xz}}{2dx}\right)F_{z,n+1}$$
$$= B_{z,n}$$

Coding for ‘invert_laplace_MTM’ solver

Becomes a matrix problem.

$$C_1 = \frac{dg^{xx}}{dx^2}$$
$$C_2 = dg^{zz}$$
$$C_3 = \frac{2dg^{xz}}{2dx}$$
$$C_4 = \frac{dG^x + g^{xx}\frac{-c2_{n-1}+c2_{n+1}}{2c1_{n}dx}}{2dx}$$
$$C_5 = dG^z + \frac{g^{xz}}{c_{1,n}}\frac{-c_{2,n-1}+c_{2,n+1}}{2dx}$$

$$d\nabla_\perp^2 f + \frac{1}{c_1}(\nabla_\perp c_2)\cdot\nabla_\perp f + af = b$$

$$(C_1 - C_4 - ikC_3)F_{z,n-1} + (-2C_1 - k^2C_2 + ikC_5 + a)F_{z,n} + (C_1 + C_4 + ikC_3)F_{z,n+1} = B_{z,n}$$

So, similarly the coefficients in the matrix of the ‘invert_laplace_MTM’ solver is,

$$C_1 = \frac{dg^{xx}}{\delta x^2}, \quad C_2 = dg, \quad C_3 = \frac{dg^{xz}}{\delta x} \quad C_4 = \frac{dG^x}{2\delta x} \quad C_5 = dG^z + \frac{-c_{n-1}+c_{n+1}}{2\delta x}\frac{1}{e_{1,n}J}$$

$$d\nabla_\perp^2 x + ax + \boldsymbol{b_0}\times\nabla x\cdot\nabla c = b$$

Coding for ‘invert_laplace_MTM’ solver

In the code, just make a little change on this function,

```
/*************************************************************************
 *                                MATRIX ELEMENTS
 *************************************************************************

void Laplacian::tridagCoefs(int jx, int jy, int jz,
                            dcomplex &a, dcomplex &b, dcomplex &c,
                            const Field2D *ccoef, const Field2D *d) {
```

```
if(ccoef != NULL) {
  // A first order derivative term

  if((jx > 0) && (jx < (mesh->ngx-1)))
    coef4 += mesh->g11[jx][jy] * ((*ccoef)[jx+1][jy] - (*ccoef)[jx-1][jy]) / (2.*mesh->dx[jx][jy]*((*ccoef)[jx][jy]));
}
```

⬇

```
if(ccoef != NULL) {
  // A first order derivative term
  if((jx > 0) && (jx < (mesh->ngx-1)))
    // for MTM, reset C5
  coef5 += mesh->g_22[jx][jy]/(mesh->J[jx][jy]*sqrt(mesh->g_22[jx][jy])) * ((*ccoef)[jx+1][jy] - (*ccoef)[jx-1][jy]) / (2.*mesh->dx[jx][jy]);
 // coef5 += mesh->g_22[jx][jy]/(mesh->J[jx][jy]*mesh->J[jx][jy]*mesh->Bxy[jx][jy]) * ((*ccoef)[jx+1][jy] - (*ccoef)[jx-1][jy]) / (2.*mesh->dx[jx][jy]);
}
```

Test the 'invert_laplace_MTM' solver

The solver is designed to solve x from a equation:

$$d\nabla_\perp^2 x + ax + b_0 \times \nabla x \cdot \nabla c = b$$

let $x = x_0$

get:

$$b = d\nabla_\perp^2 x_0 + ax_0 + b_0 \times \nabla x_0 \cdot \nabla c$$

Use the new solver to solve x_1 from the equation:

$$d\nabla_\perp^2 x_1 + ax_1 + b_0 \times \nabla x_1 \cdot \nabla c$$
$$= d\nabla_\perp^2 x_0 + ax_0 + b_0 \times \nabla x_0 \cdot \nabla c$$

The relative error:

$$RE = \left|\frac{x_1 - x_0}{\max(x_0)}\right|$$

$$\nabla_\perp^2 f + f + \mathrm{b}_0 \times \nabla_\perp f \cdot \nabla_\perp c = b$$

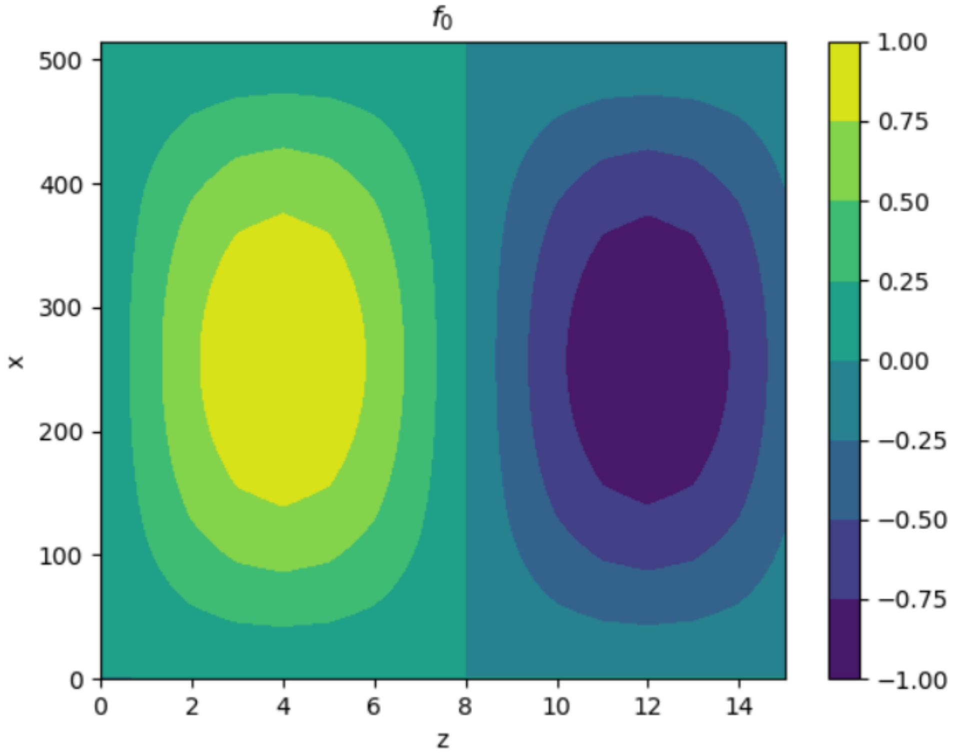
$$f_0 = sin(\pi x)\sin(\mathrm{z})$$

$$c = x + 1$$

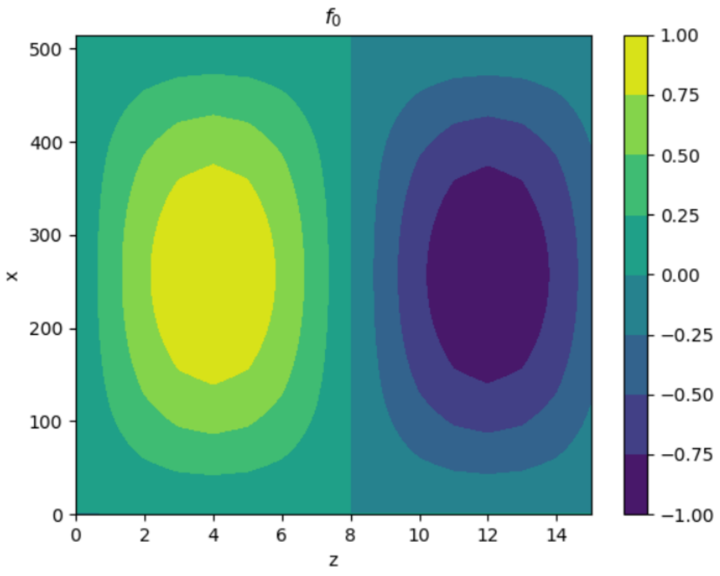
Use the 'invert_laplace' solver to get f_1 from:

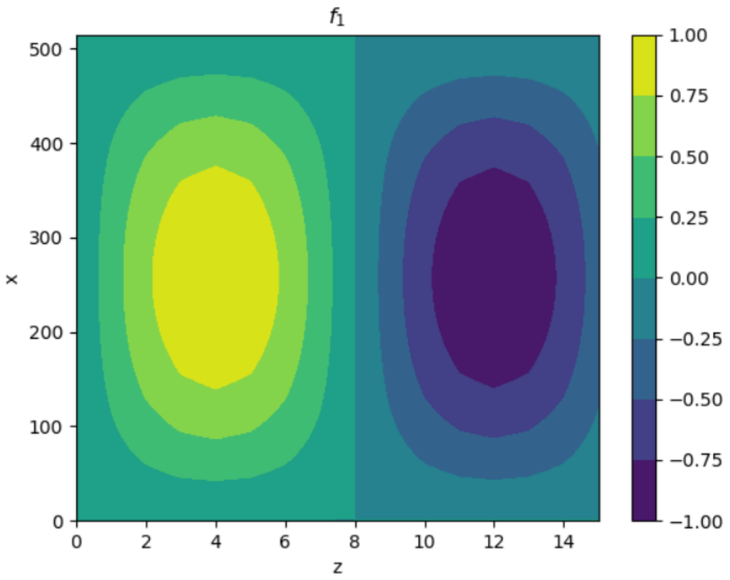
$$\nabla_\perp^2 f_1 + f_1 + \mathrm{b}_0 \times \nabla_\perp f_1 \cdot \nabla_\perp c = \nabla_\perp^2 f_0 + f_0 + \mathrm{b}_0 \times \nabla_\perp f_0 \cdot \nabla_\perp c$$

$$f_1\Big|_{x=0} = 0, \qquad f_1\Big|_{x=1} = 0$$



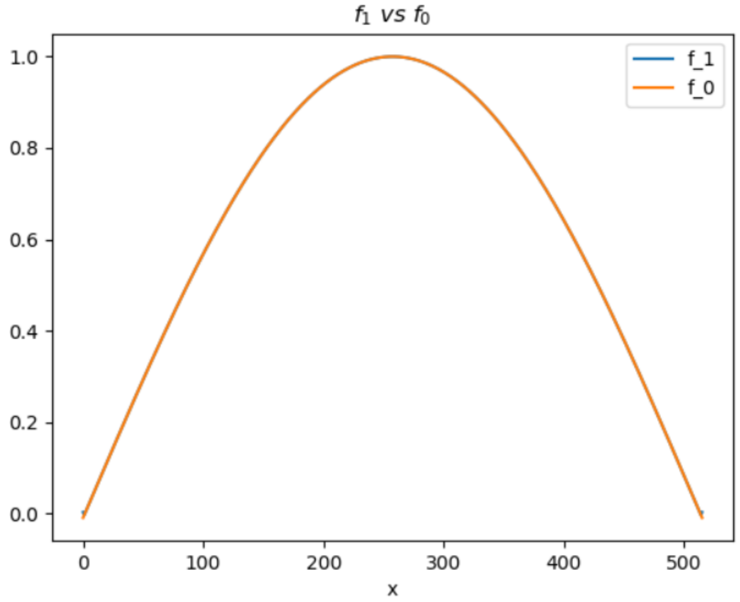
The result from the solver ‘invert-lalapce-MTM’& original input

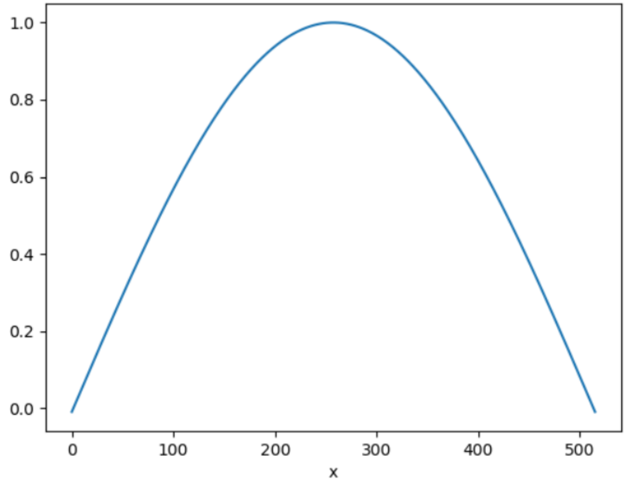


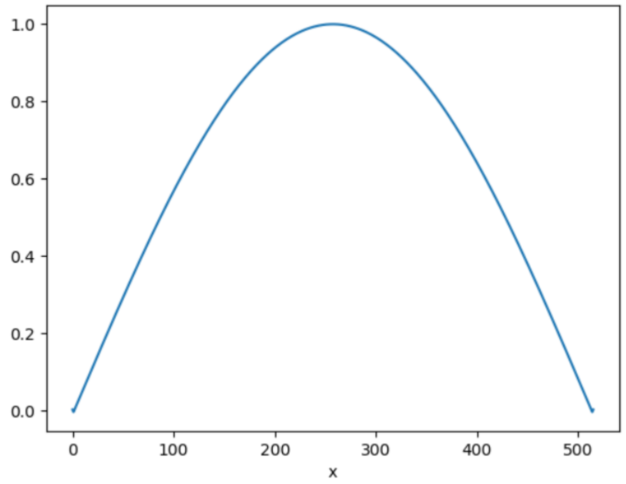


Original input

Result from the solver

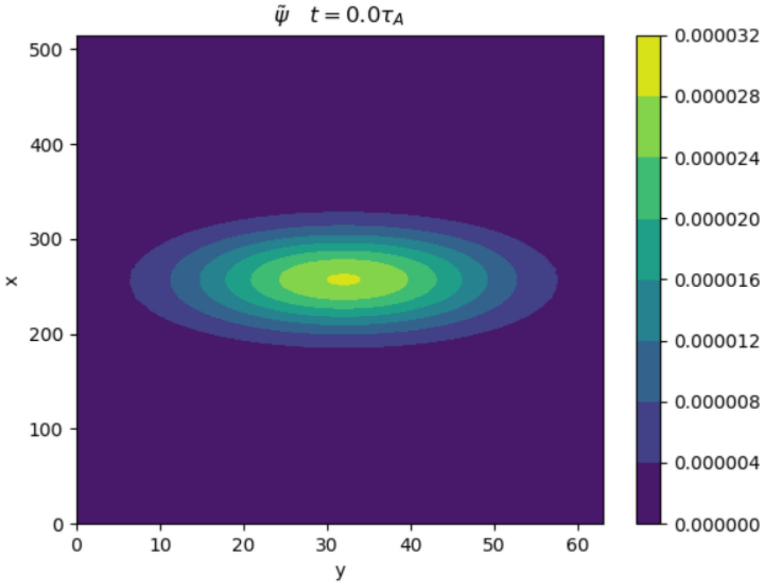






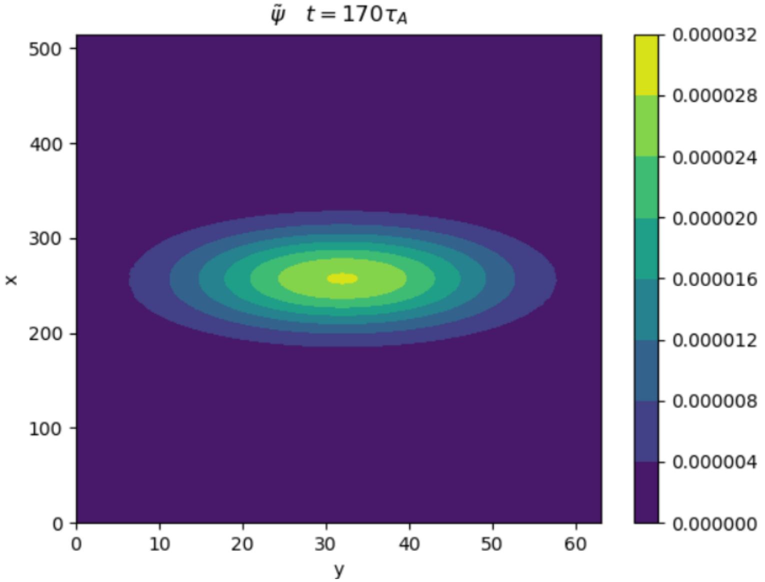
MTM simulation results

Initial setting:

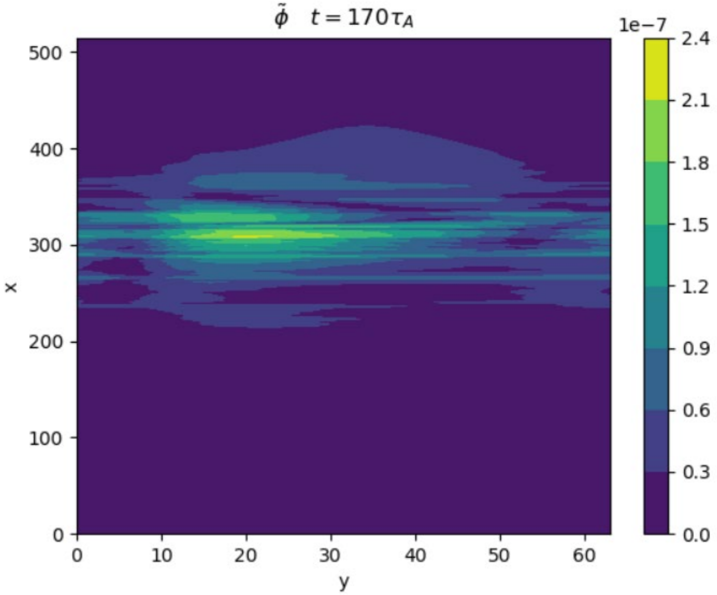


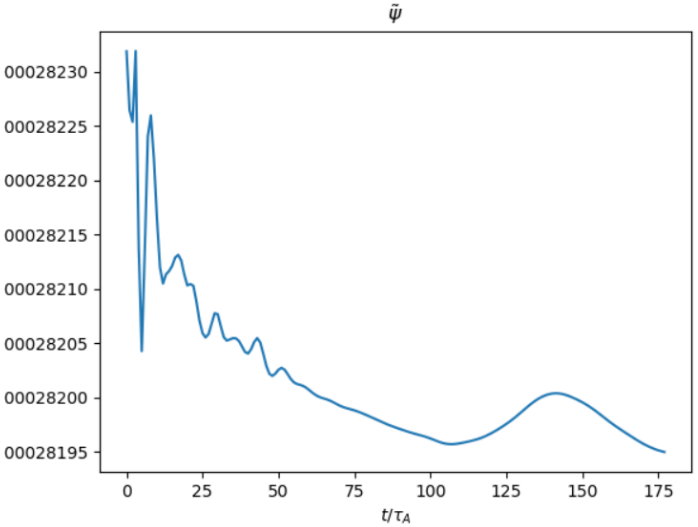
A gaussian function in both x and y direction.

$\tilde{\psi}$



$\tilde{\phi}$





Cannot get a correct mode structure and growth.

Questions

Questions 1:

Does my coding way has some problems?

Questions 2:

Some comments on the simulation setting?

Questions 3:

The possible reasons for not being able to simulate successfully?

Thanks for listening!

Kaixuan Fan

BOUT++ workshop 2023

2023.01.13