Dielectric Tensor - Cold plasma 08-06-16

N. T. Gladd

Initialization: Be sure the files *NTGStylesheet2.nb* and *NTGUtilityFunctions.m* is are in the same directory as that from which this notebook was loaded. Then execute the cell immediately below by mousing left on the cell bar to the right of that cell and then typing “shift” + “enter”. Respond “Yes” in response to the query to evaluate initialization cells.

In[28]:=
```
SetDirectory[NotebookDirectory[]];
(* set directory where source files are located *)
SetOptions[EvaluationNotebook[], (* load the StyleSheet *)
 StyleDefinitions → Get["NTGStylesheet2.nb"]];
Get["NTGUtilityFunctions.m"]; (* Load utilities package *)
```

Original notes *Dielectric Tensor - Cold plasma 05-24-15 (also many other times, starting circa 1968)*

Purpose

I use Mathematica to derive the dielectric tensor for a cold magnetized plasma, a structure used when analyzing the properties of waves that exist in this plasma. This derivation is a right of passage for students of plasma physics.

My primary interest in this series of notebooks is illustrate how Mathematica can be used to facilitate the mathematical operations involved in plasma derivations and calculations. Development of the framework for analyzing waves in a cold magnetized plasma case serves well as a basic template for more complex examples. I won’t discuss the underlying physics, most texts on plasmas cover this topic. My favorite is still the text from which I learned *Principles of Plasma Physics*, N. A. Krall and A. W. Trivelpiece for plasmas, coauthored by my friend, mentor and former boss Nick Krall.

See https://arxiv.org/pdf/physics/0701257v1.pdf for the treatment of a similar problem using a different Mathematica programming style.

Derivation of the cold plasma dielectric tensor is rather straightforward. In a subsequent notebook, I will repeat the derivation of the dielectric tensor using kinetic theory, in which case the calculation is considerable more involved.

Dispersion equation and dielectric tensor

Consider Ampere's law and Faraday's law in Gaussian notation

$$\nabla \times \vec{B}\left(\vec{x}, t\right) = \frac{4\pi}{c}\vec{j}\left(\vec{x}, t\right) + \frac{1}{c}\frac{\partial \vec{E}\left(\vec{x}, t\right)}{\partial t} \tag{1}$$

$$\nabla \times \vec{E}\left(\vec{x}, t\right) = -\frac{1}{c}\frac{\partial \vec{B}\left(\vec{x}, t\right)}{\partial t} \tag{2}$$

The current $\vec{j}$ is assumed to be related to the electric field through a conductivity tensor $\overleftrightarrow{\sigma}$

$$\vec{j} = \overleftrightarrow{\sigma} \cdot \vec{E} \tag{3}$$

Under the assumption of plane-wave waves, these three equations can be used the derive the expression

$$\overleftrightarrow{\mathcal{D}} \cdot \overrightarrow{\delta \mathrm{E}} = 0 \tag{4}$$

The dispersive properties of waves are described by the expression

$$\det(\overleftrightarrow{\mathcal{D}}) = 0 \tag{5}$$

The first derivation below will consist of using Mathematica methods to derive the matrix $\mathcal{D}$. Along the way, the dielectric tensor

$$\overleftrightarrow{\epsilon} = \left(\overleftrightarrow{1} + \frac{4\pi i}{\omega}\overleftrightarrow{\sigma}\right) \tag{6}$$

is identified.

The explicit form for $\mathcal{D}$ derived below in Section 1 is (w1[6])

$$\overleftrightarrow{\mathcal{D}} = \begin{pmatrix} 1 - \frac{c^2 k_y^2}{\omega^2} - \frac{c^2 k_z^2}{\omega^2} + \epsilon_{x,x} & \frac{c^2 k_x k_y}{\omega^2} + \epsilon_{x,y} & \frac{c^2 k_x k_z}{\omega^2} + \epsilon_{x,z} \\ \frac{c^2 k_x k_y}{\omega^2} + \epsilon_{y,x} & 1 - \frac{c^2 k_z^2}{\omega^2} - \frac{c^2 k_x^2}{\omega^2} + \epsilon_{y,y} & \frac{c^2 k_y k_z}{\omega^2} + \epsilon_{y,z} \\ \frac{c^2 k_x k_z}{\omega^2} + \epsilon_{z,x} & \frac{c^2 k_y k_z}{\omega^2} + \epsilon_{z,y} & 1 - \frac{c^2 k_y^2}{\omega^2} - \frac{c^2 k_x^2}{\omega^2} + \epsilon_{z,z} \end{pmatrix}$$

The second calculation will assume a uniform plasma embedded in a constant unidirectional magnetic field. In the limit of a cold fluid plasma, the specific expressions for the elements of the conductivity and dielectric tensors are derived.

For the cold fluid plasma, the starting equations are continuity

$$\frac{\partial n\left(\vec{x}, t\right)}{\partial t} + \nabla \cdot \left(n\left(\vec{x}, t\right)\vec{v}\left(\vec{x}, t\right)\right) = 0 \tag{7}$$

and motion

$$m\, n\left(\vec{x}, t\right) \frac{d\, \vec{v}\left(\vec{x}, t\right)}{dt} = q\, n\left(\vec{x}, t\right)\left(\vec{E}\left(\vec{x}, t\right) + \frac{\vec{v}\left(\vec{x}, t\right) \times \vec{B}\left(\vec{x}, t\right)}{c}\right) \tag{8}$$

The explicit form for the dielectric tensor derived below in Section 2 is w2["dielectric tensor"][1]

$$\overleftrightarrow{\epsilon} = \begin{pmatrix} 1 - \frac{\omega\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} & -\frac{i\, \omega_c\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} & 0 \\ \frac{i\, \omega_c\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} & 1 - \frac{\omega\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} & 0 \\ 0 & 0 & 1 - \frac{\omega_p^2}{\omega^2} \end{pmatrix}$$

Mathematica preliminaries

I predefine a number of Mathematica functions and rules
MakeScalar, MakeVector, GenerateComponentEquations, perturbationRules, equilibriumRules, LinearizeEquation,Eikonal,MakeEikonalRule, ApplyEikonal, def[ω_c], def[ω_p]
These are defined below in Section **Functions.**

I Deriving the dispersion equation

Starting with Ampere's law (1) and Faraday's law (1), I derive the dispersion equation for plane wave perturbations of a uniform plasma in a constant magnetic field

In[30]:=
```
w1["Ampere"][1] = Inactivate[Curl[ℬ, {x, y, z}] == 4 π/c j + 1/c D[ℰ, t], Curl | D]
```

Out[30]=
$$\nabla_{\{x,y,z\}} \times \mathcal{B} == \frac{4\, j\, \pi}{c} + \frac{\partial_t \mathcal{E}}{c}$$

I perform operations that transform this high level representation of Ampere's law into a form from which calculations can be performed. The function *Inactivate* temporarily prevents the functions Curl and D from evaluating. The first step is to introduce explicit time and space dependence.

In[31]:=
```
w1["Ampere"][2] = w1["Ampere"][1] /. (MakeVector /@ {ℬ, ∂t ℰ, j})
```

Out[31]=
$$\nabla_{\{x,y,z\}} \times \{\mathcal{B}_x[x, y, z, t], \mathcal{B}_y[x, y, z, t], \mathcal{B}_z[x, y, z, t]\} == \left\{\frac{\partial_t \mathcal{E}_x[x, y, z, t]}{c} + \frac{4\, \pi\, j_x[x, y, z, t]}{c}, \frac{\partial_t \mathcal{E}_y[x, y, z, t]}{c} + \frac{4\, \pi\, j_y[x, y, z, t]}{c}, \frac{\partial_t \mathcal{E}_z[x, y, z, t]}{c} + \frac{4\, \pi\, j_z[x, y, z, t]}{c}\right\}$$

With explicit space and time dependence established, the operators can be "activated."

In[32]:=
```
w1["Ampere"][3] = w1["Ampere"][2] // Activate[#, Curl | D] &
```

Out[32]=
$$\left\{-\mathcal{B}_y^{(0,0,1,0)}[x,y,z,t]+\mathcal{B}_z^{(0,1,0,0)}[x,y,z,t],\ \mathcal{B}_x^{(0,0,1,0)}[x,y,z,t]-\mathcal{B}_z^{(1,0,0,0)}[x,y,z,t],\ -\mathcal{B}_x^{(0,1,0,0)}[x,y,z,t]+\mathcal{B}_y^{(1,0,0,0)}[x,y,z,t]\right\} ==$$
$$\left\{\frac{4\pi j_x[x,y,z,t]}{c}+\frac{\mathcal{E}_x^{(0,0,0,1)}[x,y,z,t]}{c},\ \frac{4\pi j_y[x,y,z,t]}{c}+\frac{\mathcal{E}_y^{(0,0,0,1)}[x,y,z,t]}{c},\ \frac{4\pi j_z[x,y,z,t]}{c}+\frac{\mathcal{E}_z^{(0,0,0,1)}[x,y,z,t]}{c}\right\}$$

The operators expanded into the form of a {lhs$_1$, lhs$_2$,...} == {rhs$_1$, rhs$_2$,...}. The function *GenerateComponentEquations* transforms the resulting expression into a list of equations {lhs$_1$ == rhs$_1$, lhs$_2$ == rhs$_2$,... }.

In[33]:=
```
w1["Ampere"][4] = w1["Ampere"][3]  // GenerateComponentEquations
```

Out[33]=
$$\left\{-\mathcal{B}_y^{(0,0,1,0)}[x,y,z,t]+\mathcal{B}_z^{(0,1,0,0)}[x,y,z,t]==\frac{4\pi j_x[x,y,z,t]}{c}+\frac{\mathcal{E}_x^{(0,0,0,1)}[x,y,z,t]}{c},\right.$$
$$\mathcal{B}_x^{(0,0,1,0)}[x,y,z,t]-\mathcal{B}_z^{(1,0,0,0)}[x,y,z,t]==\frac{4\pi j_y[x,y,z,t]}{c}+\frac{\mathcal{E}_y^{(0,0,0,1)}[x,y,z,t]}{c},$$
$$\left.-\mathcal{B}_x^{(0,1,0,0)}[x,y,z,t]+\mathcal{B}_y^{(1,0,0,0)}[x,y,z,t]==\frac{4\pi j_z[x,y,z,t]}{c}+\frac{\mathcal{E}_z^{(0,0,0,1)}[x,y,z,t]}{c}\right\}$$

I perturb these equations using rules of the type (see perturbationRules in Functions section below)

$$n \to n_0 + \epsilon\,\delta n(x,\ y,\ z,\ t)$$
$$v_x \to v_{x0} + \epsilon\,\delta v_x(x,\ y,\ z,\ t),\quad v_y \to v_{y0} + \epsilon\,\delta v_y(x,\ y,\ z,\ t),\quad v_z \to v_{z0} + \epsilon\,\delta v_z(x,\ y,\ z,\ t)$$
$$\ldots$$

and impose the conditions of no equilibrium flows, currents, electric field, and uniform magnetic field on the equilibrium state (see equilibriumRules in Functions section below).

The imposition of these rules leads to

In[34]:=
```
w1["Ampere"][5] = w1["Ampere"][4] /. perturbationRules /. equilibriumRules
```

Out[34]=
$$\left\{-\epsilon\,\delta\mathcal{B}_y^{(0,0,1,0)}[x,y,z,t]+\epsilon\,\delta\mathcal{B}_z^{(0,1,0,0)}[x,y,z,t]==\frac{4\pi\epsilon\,\delta j_x[x,y,z,t]}{c}+\frac{\epsilon\,\delta\mathcal{E}_x^{(0,0,0,1)}[x,y,z,t]}{c},\right.$$
$$\epsilon\,\delta\mathcal{B}_x^{(0,0,1,0)}[x,y,z,t]-\epsilon\,\delta\mathcal{B}_z^{(1,0,0,0)}[x,y,z,t]==\frac{4\pi\epsilon\,\delta j_y[x,y,z,t]}{c}+\frac{\epsilon\,\delta\mathcal{E}_y^{(0,0,0,1)}[x,y,z,t]}{c},$$
$$\left.-\epsilon\,\delta\mathcal{B}_x^{(0,1,0,0)}[x,y,z,t]+\epsilon\,\delta\mathcal{B}_y^{(1,0,0,0)}[x,y,z,t]==\frac{4\pi\epsilon\,\delta j_z[x,y,z,t]}{c}+\frac{\epsilon\,\delta\mathcal{E}_z^{(0,0,0,1)}[x,y,z,t]}{c}\right\}$$

Linearize the equations

In[35]:=
```
w1["Ampere"][6] = w1["Ampere"][5] // LinearizeEquation
```

Out[35]=
$$\left\{-\delta\mathcal{B}_y^{(0,0,1,0)}[x,y,z,t]+\delta\mathcal{B}_z^{(0,1,0,0)}[x,y,z,t]==\frac{4\pi\,\delta j_x[x,y,z,t]}{c}+\frac{\delta\mathcal{E}_x^{(0,0,0,1)}[x,y,z,t]}{c},\right.$$
$$\delta\mathcal{B}_x^{(0,0,1,0)}[x,y,z,t]-\delta\mathcal{B}_z^{(1,0,0,0)}[x,y,z,t]==\frac{4\pi\,\delta j_y[x,y,z,t]}{c}+\frac{\delta\mathcal{E}_y^{(0,0,0,1)}[x,y,z,t]}{c},$$
$$\left.-\delta\mathcal{B}_x^{(0,1,0,0)}[x,y,z,t]+\delta\mathcal{B}_y^{(1,0,0,0)}[x,y,z,t]==\frac{4\pi\,\delta j_z[x,y,z,t]}{c}+\frac{\delta\mathcal{E}_z^{(0,0,0,1)}[x,y,z,t]}{c}\right\}$$

and apply a plane wave eikonal. For example,

$$\delta j_x(\vec{x},\,t)=\delta j_x(\vec{k},\omega)\,e^{i\left(\vec{k}\cdot\vec{x}-\omega t\right)}=\delta\hat{j}_x\,e^{i\left(\vec{k}\cdot\vec{x}-\omega t\right)}$$

In[36]:=
```
w1["Ampere"][7] = w1["Ampere"][6] // ApplyEikonal
```

Out[36]=
$$\left\{\delta\hat{\mathcal{B}}_z k_y-\delta\hat{\mathcal{B}}_y k_z==-\frac{4\,i\,\pi\,\delta\hat{j}_x}{c}-\frac{\omega\,\delta\hat{\mathcal{E}}_x}{c},\right.$$
$$\left.-\delta\hat{\mathcal{B}}_z k_x+\delta\hat{\mathcal{B}}_x k_z==-\frac{4\,i\,\pi\,\delta\hat{j}_y}{c}-\frac{\omega\,\delta\hat{\mathcal{E}}_y}{c},\ \delta\hat{\mathcal{B}}_y k_x-\delta\hat{\mathcal{B}}_x k_y==-\frac{4\,i\,\pi\,\delta\hat{j}_z}{c}-\frac{\omega\,\delta\hat{\mathcal{E}}_z}{c}\right\}$$

The same procedure is used for Faraday's law.

In[37]:=
```
w1["Faraday"][1] = Inactivate[Curl[ℰ, {x, y, z}] == -1/c D[ℬ, t], Curl | D]
```

Out[37]=
$$\nabla_{\{x,y,z\}}\times\mathcal{E}==-\frac{\partial_t\mathcal{B}}{c}$$

Having shown the details for Ampere, I can streamline the analogous calculation for Faraday

In[38]:=
```
w1["Faraday"][2] =
 w1["Faraday"][1] /. (MakeVector /@ {ℰ, ∂t ℬ}) // Activate[#, Curl | D] & //
  GenerateComponentEquations
```

Out[38]=
$$\left\{-\mathcal{E}_y^{(0,0,1,0)}[x,y,z,t]+\mathcal{E}_z^{(0,1,0,0)}[x,y,z,t]==-\frac{\mathcal{B}_x^{(0,0,0,1)}[x,y,z,t]}{c},\right.$$
$$\mathcal{E}_x^{(0,0,1,0)}[x,y,z,t]-\mathcal{E}_z^{(1,0,0,0)}[x,y,z,t]==-\frac{\mathcal{B}_y^{(0,0,0,1)}[x,y,z,t]}{c},$$
$$\left.-\mathcal{E}_x^{(0,1,0,0)}[x,y,z,t]+\mathcal{E}_y^{(1,0,0,0)}[x,y,z,t]==-\frac{\mathcal{B}_z^{(0,0,0,1)}[x,y,z,t]}{c}\right\}$$

In[39]:=
```
w1["Faraday"][3] =
 w1["Faraday"][2] /. perturbationRules /. equilibriumRules // LinearizeEquation //
  ApplyEikonal
```

Out[39]=
$$\left\{\delta\hat{\mathcal{E}}_z\, k_y - \delta\hat{\mathcal{E}}_y\, k_z == \frac{\omega\, \delta\hat{\mathcal{B}}_x}{c},\ -\delta\hat{\mathcal{E}}_z\, k_x + \delta\hat{\mathcal{E}}_x\, k_z == \frac{\omega\, \delta\hat{\mathcal{B}}_y}{c},\ \delta\hat{\mathcal{E}}_y\, k_x - \delta\hat{\mathcal{E}}_x\, k_y == \frac{\omega\, \delta\hat{\mathcal{B}}_z}{c}\right\}$$

Proceeding

In[40]:=
```
w1[1] = Solve[w1["Faraday"][3], {δℬ̂x, δℬ̂y, δℬ̂z}][[1]]
```

Out[40]=
$$\left\{\delta\hat{\mathcal{B}}_x \to \frac{c\,\left(\delta\hat{\mathcal{E}}_z\, k_y - \delta\hat{\mathcal{E}}_y\, k_z\right)}{\omega},\ \delta\hat{\mathcal{B}}_y \to -\frac{c\,\left(\delta\hat{\mathcal{E}}_z\, k_x - \delta\hat{\mathcal{E}}_x\, k_z\right)}{\omega},\ \delta\hat{\mathcal{B}}_z \to \frac{c\,\left(\delta\hat{\mathcal{E}}_y\, k_x - \delta\hat{\mathcal{E}}_x\, k_y\right)}{\omega}\right\}$$

Use these rules to eliminate $\delta\hat{\mathcal{B}}$ from Ampere's law

In[41]:=
```
w1[2] = w1["Ampere"][7] /. w1[1] /. a_ == b_ → a - b == 0
```

Out[41]=
$$\left\{\frac{4\,i\,\pi\,\delta\hat{j}_x}{c} + \frac{\omega\,\delta\hat{\mathcal{E}}_x}{c} + \frac{c\,k_y\,\left(\delta\hat{\mathcal{E}}_y\, k_x - \delta\hat{\mathcal{E}}_x\, k_y\right)}{\omega} + \frac{c\,k_z\,\left(\delta\hat{\mathcal{E}}_z\, k_x - \delta\hat{\mathcal{E}}_x\, k_z\right)}{\omega} == 0,\right.$$
$$\frac{4\,i\,\pi\,\delta\hat{j}_y}{c} + \frac{\omega\,\delta\hat{\mathcal{E}}_y}{c} - \frac{c\,k_x\,\left(\delta\hat{\mathcal{E}}_y\, k_x - \delta\hat{\mathcal{E}}_x\, k_y\right)}{\omega} + \frac{c\,k_z\,\left(\delta\hat{\mathcal{E}}_z\, k_y - \delta\hat{\mathcal{E}}_y\, k_z\right)}{\omega} == 0,$$
$$\left.\frac{4\,i\,\pi\,\delta\hat{j}_z}{c} + \frac{\omega\,\delta\hat{\mathcal{E}}_z}{c} - \frac{c\,k_x\,\left(\delta\hat{\mathcal{E}}_z\, k_x - \delta\hat{\mathcal{E}}_x\, k_z\right)}{\omega} - \frac{c\,k_y\,\left(\delta\hat{\mathcal{E}}_z\, k_y - \delta\hat{\mathcal{E}}_y\, k_z\right)}{\omega} == 0\right\}$$

I simplify this expression

In[42]:=
```
w1[3] = MapEqn[(Expand[# c / ω]) &, w1[2]]
```

Out[42]=
$$\left\{\frac{4\,i\,\pi\,\delta\hat{j}_x}{\omega} + \delta\hat{\mathcal{E}}_x + \frac{c^2\,\delta\hat{\mathcal{E}}_y\, k_x\, k_y}{\omega^2} - \frac{c^2\,\delta\hat{\mathcal{E}}_x\, k_y^2}{\omega^2} + \frac{c^2\,\delta\hat{\mathcal{E}}_z\, k_x\, k_z}{\omega^2} - \frac{c^2\,\delta\hat{\mathcal{E}}_x\, k_z^2}{\omega^2} == 0,\right.$$
$$\frac{4\,i\,\pi\,\delta\hat{j}_y}{\omega} + \delta\hat{\mathcal{E}}_y - \frac{c^2\,\delta\hat{\mathcal{E}}_y\, k_x^2}{\omega^2} + \frac{c^2\,\delta\hat{\mathcal{E}}_x\, k_x\, k_y}{\omega^2} + \frac{c^2\,\delta\hat{\mathcal{E}}_z\, k_y\, k_z}{\omega^2} - \frac{c^2\,\delta\hat{\mathcal{E}}_y\, k_z^2}{\omega^2} == 0,$$
$$\left.\frac{4\,i\,\pi\,\delta\hat{j}_z}{\omega} + \delta\hat{\mathcal{E}}_z - \frac{c^2\,\delta\hat{\mathcal{E}}_z\, k_x^2}{\omega^2} - \frac{c^2\,\delta\hat{\mathcal{E}}_z\, k_y^2}{\omega^2} + \frac{c^2\,\delta\hat{\mathcal{E}}_x\, k_x\, k_z}{\omega^2} + \frac{c^2\,\delta\hat{\mathcal{E}}_y\, k_y\, k_z}{\omega^2} == 0\right\}$$

and introduce the conductivity tensor via $\vec{\delta j} = \overleftrightarrow{\sigma} \cdot \vec{\delta E}$

In[43]:=
```
conductivityRule = δĵa_ → σa,x δℰ̂x + σa,y δℰ̂y + σa,z δℰ̂z
```

Out[43]=
$$\delta\hat{j}_{a\_} \to \delta\hat{\mathcal{E}}_x\, \sigma_{a,x} + \delta\hat{\mathcal{E}}_y\, \sigma_{a,y} + \delta\hat{\mathcal{E}}_z\, \sigma_{a,z}$$

In[44]:=
```
w1[4] = w1[3] /. conductivityRule // ExpandAll
```

Out[44]=
$$\left\{\delta\hat{\mathcal{E}}_x + \frac{c^2\,\delta\hat{\mathcal{E}}_y\,k_x\,k_y}{\omega^2} - \frac{c^2\,\delta\hat{\mathcal{E}}_x\,k_y^2}{\omega^2} + \frac{c^2\,\delta\hat{\mathcal{E}}_z\,k_x\,k_z}{\omega^2} - \frac{c^2\,\delta\hat{\mathcal{E}}_x\,k_z^2}{\omega^2} + \frac{4\,i\,\pi\,\delta\hat{\mathcal{E}}_x\,\sigma_{x,x}}{\omega} + \frac{4\,i\,\pi\,\delta\hat{\mathcal{E}}_y\,\sigma_{x,y}}{\omega} + \frac{4\,i\,\pi\,\delta\hat{\mathcal{E}}_z\,\sigma_{x,z}}{\omega} == 0,\right.$$
$$\delta\hat{\mathcal{E}}_y - \frac{c^2\,\delta\hat{\mathcal{E}}_y\,k_x^2}{\omega^2} + \frac{c^2\,\delta\hat{\mathcal{E}}_x\,k_x\,k_y}{\omega^2} + \frac{c^2\,\delta\hat{\mathcal{E}}_z\,k_y\,k_z}{\omega^2} - \frac{c^2\,\delta\hat{\mathcal{E}}_y\,k_z^2}{\omega^2} + \frac{4\,i\,\pi\,\delta\hat{\mathcal{E}}_x\,\sigma_{y,x}}{\omega} + \frac{4\,i\,\pi\,\delta\hat{\mathcal{E}}_y\,\sigma_{y,y}}{\omega} + \frac{4\,i\,\pi\,\delta\hat{\mathcal{E}}_z\,\sigma_{y,z}}{\omega} == 0,$$
$$\left.\delta\hat{\mathcal{E}}_z - \frac{c^2\,\delta\hat{\mathcal{E}}_z\,k_x^2}{\omega^2} - \frac{c^2\,\delta\hat{\mathcal{E}}_z\,k_y^2}{\omega^2} + \frac{c^2\,\delta\hat{\mathcal{E}}_x\,k_x\,k_z}{\omega^2} + \frac{c^2\,\delta\hat{\mathcal{E}}_y\,k_y\,k_z}{\omega^2} + \frac{4\,i\,\pi\,\delta\hat{\mathcal{E}}_x\,\sigma_{z,x}}{\omega} + \frac{4\,i\,\pi\,\delta\hat{\mathcal{E}}_y\,\sigma_{z,y}}{\omega} + \frac{4\,i\,\pi\,\delta\hat{\mathcal{E}}_z\,\sigma_{z,z}}{\omega} == 0\right\}$$

Construct the elements of the matrix dispersion equation $\overleftrightarrow{\mathcal{D}} \cdot \overrightarrow{\delta E} = 0$

In[45]:=
```
w1[5] = Table[
  Table[Coefficient[w1[4][[i, 1]], component], {component, {δℰx, δℰy, δℰz}}], {i, 1, 3}]
```

Out[45]=
$$\left\{\left\{1 - \frac{c^2\,k_y^2}{\omega^2} - \frac{c^2\,k_z^2}{\omega^2} + \frac{4\,i\,\pi\,\sigma_{x,x}}{\omega},\ \frac{c^2\,k_x\,k_y}{\omega^2} + \frac{4\,i\,\pi\,\sigma_{x,y}}{\omega},\ \frac{c^2\,k_x\,k_z}{\omega^2} + \frac{4\,i\,\pi\,\sigma_{x,z}}{\omega}\right\},\right.$$
$$\left\{\frac{c^2\,k_x\,k_y}{\omega^2} + \frac{4\,i\,\pi\,\sigma_{y,x}}{\omega},\ 1 - \frac{c^2\,k_x^2}{\omega^2} - \frac{c^2\,k_z^2}{\omega^2} + \frac{4\,i\,\pi\,\sigma_{y,y}}{\omega},\ \frac{c^2\,k_y\,k_z}{\omega^2} + \frac{4\,i\,\pi\,\sigma_{y,z}}{\omega}\right\},$$
$$\left.\left\{\frac{c^2\,k_x\,k_z}{\omega^2} + \frac{4\,i\,\pi\,\sigma_{z,x}}{\omega},\ \frac{c^2\,k_y\,k_z}{\omega^2} + \frac{4\,i\,\pi\,\sigma_{z,y}}{\omega},\ 1 - \frac{c^2\,k_x^2}{\omega^2} - \frac{c^2\,k_y^2}{\omega^2} + \frac{4\,i\,\pi\,\sigma_{z,z}}{\omega}\right\}\right\}$$

It is conventional to introduce the dielectric tensor $\overleftrightarrow{\epsilon} = 1 + \frac{4\,\pi i}{\omega}\overleftrightarrow{\sigma}$

In[46]:=
```
w1[6] = w1[5] /. 4 i π σa_,b_ / ω → ϵa,b - If[ToString[a] == ToString[b], 1, 0]
```

Out[46]=
$$\left\{\left\{1 - \frac{c^2\,k_y^2}{\omega^2} - \frac{c^2\,k_z^2}{\omega^2} + \epsilon_{x,x},\ \frac{c^2\,k_x\,k_y}{\omega^2} + \epsilon_{x,y},\ \frac{c^2\,k_x\,k_z}{\omega^2} + \epsilon_{x,z}\right\},\right.$$
$$\left\{\frac{c^2\,k_x\,k_y}{\omega^2} + \epsilon_{y,x},\ 1 - \frac{c^2\,k_x^2}{\omega^2} - \frac{c^2\,k_z^2}{\omega^2} + \epsilon_{y,y},\ \frac{c^2\,k_y\,k_z}{\omega^2} + \epsilon_{y,z}\right\},$$
$$\left.\left\{\frac{c^2\,k_x\,k_z}{\omega^2} + \epsilon_{z,x},\ \frac{c^2\,k_y\,k_z}{\omega^2} + \epsilon_{z,y},\ 1 - \frac{c^2\,k_x^2}{\omega^2} - \frac{c^2\,k_y^2}{\omega^2} + \epsilon_{z,z}\right\}\right\}$$

In[47]:=
```
w1[6] // MatrixForm
```

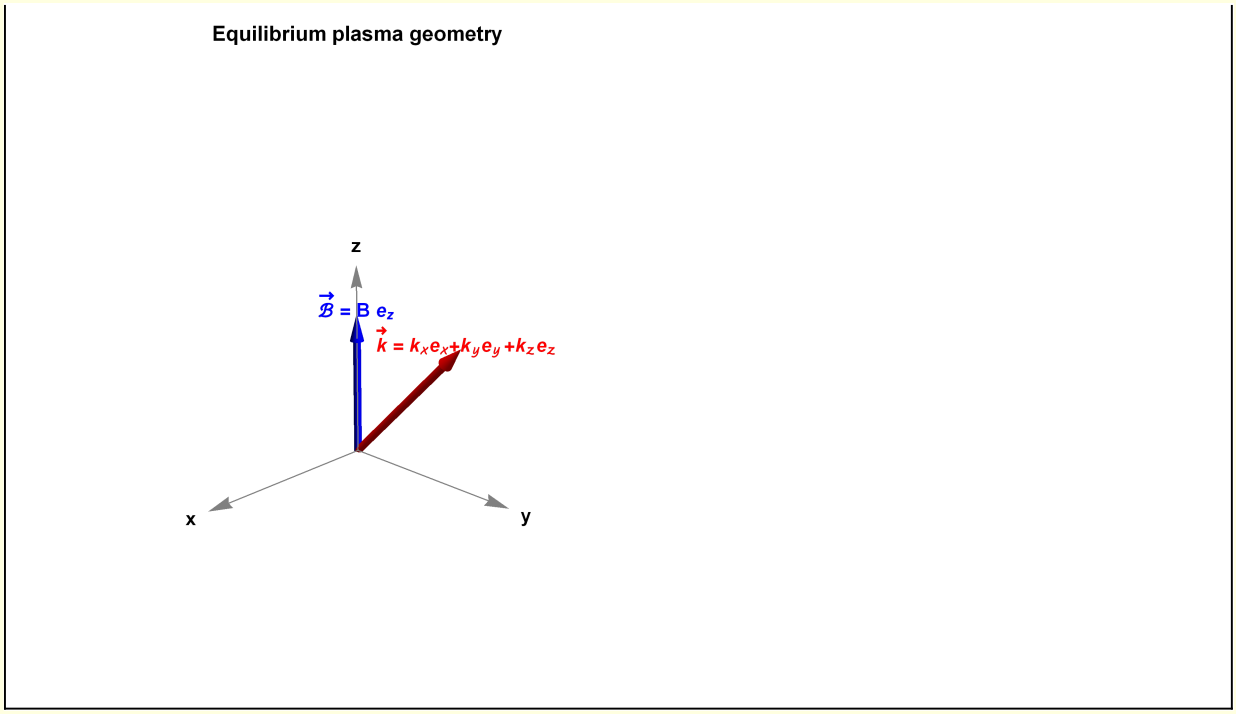
Out[47]//MatrixForm=
$$\begin{pmatrix} 1 - \frac{c^2\,k_y^2}{\omega^2} - \frac{c^2\,k_z^2}{\omega^2} + \epsilon_{x,x} & \frac{c^2\,k_x\,k_y}{\omega^2} + \epsilon_{x,y} & \frac{c^2\,k_x\,k_z}{\omega^2} + \epsilon_{x,z} \\ \frac{c^2\,k_x\,k_y}{\omega^2} + \epsilon_{y,x} & 1 - \frac{c^2\,k_x^2}{\omega^2} - \frac{c^2\,k_z^2}{\omega^2} + \epsilon_{y,y} & \frac{c^2\,k_y\,k_z}{\omega^2} + \epsilon_{y,z} \\ \frac{c^2\,k_x\,k_z}{\omega^2} + \epsilon_{z,x} & \frac{c^2\,k_y\,k_z}{\omega^2} + \epsilon_{z,y} & 1 - \frac{c^2\,k_x^2}{\omega^2} - \frac{c^2\,k_y^2}{\omega^2} + \epsilon_{z,z} \end{pmatrix}$$

which is the desired result. The dielectric terms depend on the details of the plasma model. Also, any element of the dielectric tensor (or the equivalent conductivity tensor) implies a sum over each of the plasma species. Thus, for a plasma consisting of ions and electrons, the following interpretation is implied

$$\overleftrightarrow{\epsilon} \to \overleftrightarrow{\epsilon}_i + \overleftrightarrow{\epsilon}_e$$

2 Cold plasma modeled by fluid equations

A homogeneous cold plasma with a uniform magnetic field is assumed.



The continuity equation is processed first

In[48]:=
```
w2["continuity"][1] = Inactivate[D[n, t] + Div[n v, {x, y, z}] == 0, D | Div]
```

Out[48]=
$$\partial_t n + \nabla_{\{x,y,z\}} \cdot (n\, v) == 0$$

In[49]:=
```
w2["continuity"][2] =
 w2["continuity"][1] /. {MakeScalar[n], MakeVector[v]} // Activate[#, D | Div] &
```

Out[49]=
$$n^{(0,0,0,1)}[x, y, z, t] + v_z[x, y, z, t]\, n^{(0,0,1,0)}[x, y, z, t] + n[x, y, z, t]\, v_z^{(0,0,1,0)}[x, y, z, t] + v_y[x, y, z, t]\, n^{(0,1,0,0)}[x, y, z, t] + n[x, y, z, t]\, v_y^{(0,1,0,0)}[x, y, z, t] + v_x[x, y, z, t]\, n^{(1,0,0,0)}[x, y, z, t] + n[x, y, z, t]\, v_x^{(1,0,0,0)}[x, y, z, t] == 0$$

In[50]:=
```
w2["continuity"][3] =
 w2["continuity"][2] /. perturbationRules /. equilibriumRules //
   LinearizeEquation // ApplyEikonal
```

Out[50]=
$$-\omega\,\delta\hat{n} + \delta\hat{v}_x\,k_x\,n_0 + \delta\hat{v}_y\,k_y\,n_0 + \delta\hat{v}_z\,k_z\,n_0 == 0$$

In[51]:=
```
w2["continuity"][4] = Solve[w2["continuity"][3], δn̂][[1, 1]] // ExpandAll
```

Out[51]=
$$\delta\hat{n} \to \frac{\delta\hat{v}_x\,k_x\,n_0}{\omega} + \frac{\delta\hat{v}_y\,k_y\,n_0}{\omega} + \frac{\delta\hat{v}_z\,k_z\,n_0}{\omega}$$

and then the equation of fluid motion

In[52]:=
```
w2["motion"][1] =
 Inactivate[m n DDt[v, v] == n q (ℰ + 1/c Cross[v, ℬ]), DDt | Cross | Plus]
```

Out[52]=
$$m\,n\,\text{DDt}[v, v] == n\,q\left(\mathcal{E} + \frac{v \times \mathcal{B}}{c}\right)$$

where the functions *DDt* is defined below to implement the convective derivative $\frac{d}{dt} = \frac{\partial}{\partial t} + v \cdot \nabla$. Again, this notebook focuses on the calculation. A plasma physics text should be consulted to appreciate the context in which these fluid equations are applicable.

Now that the vector terms have been expanded, it is preferable to rewrite this expression as a list of equations.

In[54]:=
```
w2["motion"][2] =
 w2["motion"][1] /. {MakeScalar[n], MakeVector[v], MakeVector[ℰ], MakeVector[ℬ]}
```

Out[54]=
$$m\,n[x, y, z, t]\,\text{DDt}[\{v_x[x, y, z, t], v_y[x, y, z, t], v_z[x, y, z, t]\},\ \{v_x[x, y, z, t], v_y[x, y, z, t], v_z[x, y, z, t]\}] ==$$
$$q\,n[x, y, z, t]\left(\{\mathcal{E}_x[x, y, z, t], \mathcal{E}_y[x, y, z, t], \mathcal{E}_z[x, y, z, t]\} + \frac{1}{c}\{v_x[x, y, z, t], v_y[x, y, z, t], v_z[x, y, z, t]\} \times \{\mathcal{B}_x[x, y, z, t], \mathcal{B}_y[x, y, z, t], \mathcal{B}_z[x, y, z, t]\}\right)$$

In[55]:=
```
w2["motion"][3] = Activate[w2["motion"][2], DDt | Cross]
```

Out[55]=
$$\begin{aligned}&\{m\,n[x,y,z,t]\,\left(v_x^{(0,0,0,1)}[x,y,z,t]+v_x[x,y,z,t]\,v_x^{(1,0,0,0)}[x,y,z,t]\right),\\&\ m\,n[x,y,z,t]\,\left(v_y^{(0,0,0,1)}[x,y,z,t]+v_y[x,y,z,t]\,v_y^{(0,1,0,0)}[x,y,z,t]\right),\\&\ m\,n[x,y,z,t]\,\left(v_z^{(0,0,0,1)}[x,y,z,t]+v_z[x,y,z,t]\,v_z^{(0,0,1,0)}[x,y,z,t]\right)\}==\\&q\,n[x,y,z,t]\,\Bigg(\{\mathcal{E}_x[x,y,z,t],\mathcal{E}_y[x,y,z,t],\mathcal{E}_z[x,y,z,t]\}+\\&\quad\Big\{\frac{1}{c}\left(-v_z[x,y,z,t]\,\mathcal{B}_y[x,y,z,t]+v_y[x,y,z,t]\,\mathcal{B}_z[x,y,z,t]\right),\frac{1}{c}\\&\quad\left(v_z[x,y,z,t]\,\mathcal{B}_x[x,y,z,t]-v_x[x,y,z,t]\,\mathcal{B}_z[x,y,z,t]\right),\frac{1}{c}\\&\quad\left(-v_y[x,y,z,t]\,\mathcal{B}_x[x,y,z,t]+v_x[x,y,z,t]\,\mathcal{B}_y[x,y,z,t]\right)\Big\}\Bigg)\end{aligned}$$

In[56]:=
```
w2["motion"][4] = Activate[w2["motion"][3], Plus]
```

Out[56]=
$$\begin{aligned}&\{m\,n[x,y,z,t]\,\left(v_x^{(0,0,0,1)}[x,y,z,t]+v_x[x,y,z,t]\,v_x^{(1,0,0,0)}[x,y,z,t]\right),\\&\ m\,n[x,y,z,t]\,\left(v_y^{(0,0,0,1)}[x,y,z,t]+v_y[x,y,z,t]\,v_y^{(0,1,0,0)}[x,y,z,t]\right),\\&\ m\,n[x,y,z,t]\,\left(v_z^{(0,0,0,1)}[x,y,z,t]+v_z[x,y,z,t]\,v_z^{(0,0,1,0)}[x,y,z,t]\right)\}==\\&\{q\,n[x,y,z,t]\,\Bigg(\frac{1}{c}\left(-v_z[x,y,z,t]\,\mathcal{B}_y[x,y,z,t]+v_y[x,y,z,t]\,\mathcal{B}_z[x,y,z,t]\right)+\\&\quad\mathcal{E}_x[x,y,z,t]\Bigg),q\,n[x,y,z,t]\\&\quad\left(\frac{1}{c}\left(v_z[x,y,z,t]\,\mathcal{B}_x[x,y,z,t]-v_x[x,y,z,t]\,\mathcal{B}_z[x,y,z,t]\right)+\mathcal{E}_y[x,y,z,t]\right),q\\&\ n[x,y,z,t]\\&\quad\left(\frac{1}{c}\left(-v_y[x,y,z,t]\,\mathcal{B}_x[x,y,z,t]+v_x[x,y,z,t]\,\mathcal{B}_y[x,y,z,t]\right)+\mathcal{E}_z[x,y,z,t]\right)\}\end{aligned}$$

The procedure of activating DDt and Cross before activating Plus is kludgy. The reason is that you need to have the vector expressions expand into lists before they are added. To see the problem try the “motion” calculation without initially inactivating Plus.

In[57]:=
```
w2["motion"][5] = w2["motion"][4] // GenerateComponentEquations
```

Out[57]=
$$\begin{aligned}&\{m\,n[x,y,z,t]\,v_x^{(0,0,0,1)}[x,y,z,t]+m\,n[x,y,z,t]\,v_x[x,y,z,t]\,v_x^{(1,0,0,0)}[x,y,z,t]==\\&\ -\frac{1}{c}q\,n[x,y,z,t]\,v_z[x,y,z,t]\,\mathcal{B}_y[x,y,z,t]+\frac{1}{c}\\&\ q\,n[x,y,z,t]\,v_y[x,y,z,t]\,\mathcal{B}_z[x,y,z,t]+q\,n[x,y,z,t]\,\mathcal{E}_x[x,y,z,t],\\&m\,n[x,y,z,t]\,v_y^{(0,0,0,1)}[x,y,z,t]+m\,n[x,y,z,t]\,v_y[x,y,z,t]\,v_y^{(0,1,0,0)}[x,y,z,t]==\\&\ \frac{1}{c}q\,n[x,y,z,t]\,v_z[x,y,z,t]\,\mathcal{B}_x[x,y,z,t]-\frac{1}{c}\\&\ q\,n[x,y,z,t]\,v_x[x,y,z,t]\,\mathcal{B}_z[x,y,z,t]+q\,n[x,y,z,t]\,\mathcal{E}_y[x,y,z,t],\\&m\,n[x,y,z,t]\,v_z^{(0,0,0,1)}[x,y,z,t]+m\,n[x,y,z,t]\,v_z[x,y,z,t]\,v_z^{(0,0,1,0)}[x,y,z,t]==\\&\ -\frac{1}{c}q\,n[x,y,z,t]\,v_y[x,y,z,t]\,\mathcal{B}_x[x,y,z,t]+\frac{1}{c}\\&\ q\,n[x,y,z,t]\,v_x[x,y,z,t]\,\mathcal{B}_y[x,y,z,t]+q\,n[x,y,z,t]\,\mathcal{E}_z[x,y,z,t]\}\end{aligned}$$

In[58]:=
```
w2["motion"][6] =
 w2["motion"][5] /. perturbationRules /. equilibriumRules // LinearizeEquation
```

Out[58]=
$$\left\{m\, n_0\, \delta v_x^{(0,0,0,1)}[x, y, z, t] == \frac{B\, q\, n_0\, \delta v_y[x, y, z, t]}{c} + q\, n_0\, \delta \mathcal{E}_x[x, y, z, t],\right.$$
$$m\, n_0\, \delta v_y^{(0,0,0,1)}[x, y, z, t] == -\frac{B\, q\, n_0\, \delta v_x[x, y, z, t]}{c} + q\, n_0\, \delta \mathcal{E}_y[x, y, z, t],$$
$$\left. m\, n_0\, \delta v_z^{(0,0,0,1)}[x, y, z, t] == q\, n_0\, \delta \mathcal{E}_z[x, y, z, t]\right\}$$

In[59]:=
```
w2["motion"][7] = w2["motion"][6] /. Sol[def[ω_c], B]
```

Out[59]=
$$\left\{m\, n_0\, \delta v_x^{(0,0,0,1)}[x, y, z, t] == m\, n_0\, \omega_c\, \delta v_y[x, y, z, t] + q\, n_0\, \delta \mathcal{E}_x[x, y, z, t],\right.$$
$$m\, n_0\, \delta v_y^{(0,0,0,1)}[x, y, z, t] == -m\, n_0\, \omega_c\, \delta v_x[x, y, z, t] + q\, n_0\, \delta \mathcal{E}_y[x, y, z, t],$$
$$\left. m\, n_0\, \delta v_z^{(0,0,0,1)}[x, y, z, t] == q\, n_0\, \delta \mathcal{E}_z[x, y, z, t]\right\}$$

In[60]:=
```
w2["motion"][8] = w2["motion"][7] // ApplyEikonal
```

Out[60]=
$$\left\{-m\, \omega\, \delta \hat{v}_x\, n_0 == -i\, q\, \delta \hat{\mathcal{E}}_x\, n_0 - i\, m\, \delta \hat{v}_y\, n_0\, \omega_c,\right.$$
$$\left. -m\, \omega\, \delta \hat{v}_y\, n_0 == -i\, q\, \delta \hat{\mathcal{E}}_y\, n_0 + i\, m\, \delta \hat{v}_x\, n_0\, \omega_c,\ -m\, \omega\, \delta \hat{v}_z\, n_0 == -i\, q\, \delta \hat{\mathcal{E}}_z\, n_0\right\}$$

In[61]:=
```
w2["motion"][9] = Solve[w2["motion"][8], {δv̂_x, δv̂_y, δv̂_z}][[1]] // ExpandAll
```

Out[61]=
$$\left\{\delta \hat{v}_x \to \frac{i\, q\, \omega\, \delta \hat{\mathcal{E}}_x}{m\, \omega^2 - m\, \omega_c^2} - \frac{q\, \delta \hat{\mathcal{E}}_y\, \omega_c}{m\, \omega^2 - m\, \omega_c^2},\ \delta \hat{v}_y \to \frac{i\, q\, \omega\, \delta \hat{\mathcal{E}}_y}{m\, \omega^2 - m\, \omega_c^2} + \frac{q\, \delta \hat{\mathcal{E}}_x\, \omega_c}{m\, \omega^2 - m\, \omega_c^2},\ \delta \hat{v}_z \to \frac{i\, q\, \delta \hat{\mathcal{E}}_z}{m\, \omega}\right\}$$

The perturbed velocities are

In[62]:=
```
w2["motion"][9] // ColumnForm
```

Out[62]=
$$\delta \hat{v}_x \to \frac{i\, q\, \omega\, \delta \hat{\mathcal{E}}_x}{m\, \omega^2 - m\, \omega_c^2} - \frac{q\, \delta \hat{\mathcal{E}}_y\, \omega_c}{m\, \omega^2 - m\, \omega_c^2}$$
$$\delta \hat{v}_y \to \frac{i\, q\, \omega\, \delta \hat{\mathcal{E}}_y}{m\, \omega^2 - m\, \omega_c^2} + \frac{q\, \delta \hat{\mathcal{E}}_x\, \omega_c}{m\, \omega^2 - m\, \omega_c^2}$$
$$\delta \hat{v}_z \to \frac{i\, q\, \delta \hat{\mathcal{E}}_z}{m\, \omega}$$

The current density is

In[63]:=
```
w2["current density"][1] = q n_0 {δv̂_x, δv̂_y, δv̂_z} /. w2["motion"][9]
```

Out[63]=
$$\left\{q\, n_0 \left(\frac{i\, q\, \omega\, \delta \hat{\mathcal{E}}_x}{m\, \omega^2 - m\, \omega_c^2} - \frac{q\, \delta \hat{\mathcal{E}}_y\, \omega_c}{m\, \omega^2 - m\, \omega_c^2}\right),\ q\, n_0 \left(\frac{i\, q\, \omega\, \delta \hat{\mathcal{E}}_y}{m\, \omega^2 - m\, \omega_c^2} + \frac{q\, \delta \hat{\mathcal{E}}_x\, \omega_c}{m\, \omega^2 - m\, \omega_c^2}\right),\ \frac{i\, q^2\, \delta \hat{\mathcal{E}}_z\, n_0}{m\, \omega}\right\}$$

The conductivity tensor is

In[64]:=
```
w2["4 π i/ω σ⃡"][1] = 4 π I/ω w2["current density"][1]
```

Out[64]=
$$\left\{\frac{1}{\omega} 4\, \mathbb{i}\, \pi\, q\, n_0 \left(\frac{\mathbb{i}\, q\, \omega\, \delta\hat{\mathcal{E}}_x}{m\,\omega^2 - m\,\omega_c^2} - \frac{q\, \delta\hat{\mathcal{E}}_y\, \omega_c}{m\,\omega^2 - m\,\omega_c^2}\right), \frac{1}{\omega} 4\, \mathbb{i}\, \pi\, q\, n_0 \left(\frac{\mathbb{i}\, q\, \omega\, \delta\hat{\mathcal{E}}_y}{m\,\omega^2 - m\,\omega_c^2} + \frac{q\, \delta\hat{\mathcal{E}}_x\, \omega_c}{m\,\omega^2 - m\,\omega_c^2}\right), -\frac{4\,\pi\, q^2\, \delta\hat{\mathcal{E}}_z\, n_0}{m\,\omega^2}\right\}$$

In[66]:=
```
w2["4 π i/ω σ⃡"][2] = w2["4 π i/ω σ⃡"][1] /. Sol[def[ω_p], n_0] // Simplify // ExpandAll
```

Out[66]=
$$\left\{-\frac{\omega\, \delta\hat{\mathcal{E}}_x\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} - \frac{\mathbb{i}\, \delta\hat{\mathcal{E}}_y\, \omega_c\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2}, -\frac{\omega\, \delta\hat{\mathcal{E}}_y\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} + \frac{\mathbb{i}\, \delta\hat{\mathcal{E}}_x\, \omega_c\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2}, -\frac{\delta\hat{\mathcal{E}}_z\, \omega_p^2}{\omega^2}\right\}$$

In[67]:=
```
w2["4 π i/ω σ⃡"][3] = Table[Table[Coefficient[w2["4 π i/ω σ⃡"][2][[i]], component],
    {component, {δℰ̂x, δℰ̂y, δℰ̂z}}], {i, 1, 3}]
```

Out[67]=
$$\left\{\left\{-\frac{\omega\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2}, -\frac{\mathbb{i}\, \omega_c\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2}, 0\right\}, \left\{\frac{\mathbb{i}\, \omega_c\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2}, -\frac{\omega\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2}, 0\right\}, \left\{0, 0, -\frac{\omega_p^2}{\omega^2}\right\}\right\}$$

In[68]:=
```
w2["4 π i/ω σ⃡"][3] // MatrixForm
```

Out[68]//MatrixForm=
$$\begin{pmatrix} -\frac{\omega\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} & -\frac{\mathbb{i}\, \omega_c\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} & 0 \\ \frac{\mathbb{i}\, \omega_c\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} & -\frac{\omega\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} & 0 \\ 0 & 0 & -\frac{\omega_p^2}{\omega^2} \end{pmatrix}$$

The dielectric tensor is

In[69]:=
```
w2["dielectric tensor"][1] = IdentityMatrix[3] + w2["4 π i/ω σ⃡"][3]
```

Out[69]=
$$\left\{\left\{1 - \frac{\omega\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2}, -\frac{\mathbb{i}\, \omega_c\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2}, 0\right\}, \left\{\frac{\mathbb{i}\, \omega_c\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2}, 1 - \frac{\omega\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2}, 0\right\}, \left\{0, 0, 1 - \frac{\omega_p^2}{\omega^2}\right\}\right\}$$

In[70]:=
```
w2["dielectric tensor"][1] // MatrixForm
```

Out[70]//MatrixForm=
$$\begin{pmatrix} 1 - \frac{\omega\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} & -\frac{\mathbb{i}\, \omega_c\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} & 0 \\ \frac{\mathbb{i}\, \omega_c\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} & 1 - \frac{\omega\, \omega_p^2}{\omega^3 - \omega\, \omega_c^2} & 0 \\ 0 & 0 & 1 - \frac{\omega_p^2}{\omega^2} \end{pmatrix}$$

and the specific form for the dielectric tensor has been obtained.

Functions

The following are functions that facilitate the handling of vector expressions.

In[71]:=
```
Clear[MakeScalar, MakeVector];
(* These functions implement rules that transform a symbol in to a
  scalar or vector form that included explicit space-time dependence *)
MakeScalar[arg_] := arg → arg[x, y, z, t];
(* using x,y,z for vector components and x, y,
z for variables simplifies some pattern matching operations *)
MakeVector[arg_] := arg → {arg_x[x, y, z, t], arg_y[x, y, z, t], arg_z[x, y, z, t]};
(* Handles the case of "vectoring" the time derivative of a vector quantity *)
MakeVector[Inactivate[D[arg_, t], D]] :=
 Inactivate[D[arg, t], D] → {Inactivate[D[ arg_x[x, y, z, t], t], D],
   Inactivate[D[ arg_y[x, y, z, t], t], D], Inactivate[D[ arg_z[x, y, z, t], t], D]}
```

In[7]:=
```
Clear[GenerateComponentEquations];
(* This transforms {a, b, c} == {d, e, f} to {a==c,b==d,c==e} *)
GenerateComponentEquations[lhsList_ == rhsList_] :=
 Thread[Equal[ExpandAll[lhsList], ExpandAll[rhsList]]]
```

Convective derivative: The functions *DDt* and *vDel* implement the convective derivative

$$\frac{d}{dt} = \frac{\partial}{\partial t} + v \cdot \nabla$$

In[9]:=
```
Clear[DDt];
DDt[{vx_, vy_, vz_}, {fx_, fy_, fz_}] :=
  D[{fx, fy, fz}, t] + vDel[{vx, vy, vz}, {fx, fy, fz}];
DDt[{vx_, vy_, vz_}, f_] := D[f, t] + vDel[{vx, vy, vz}, f];
```

In[12]:=
```
Clear[vDel];
vDel[{vx_, vy_, vz_}, f_] :=
  vx D[f, x] + vy D[f, y] + vz D[f, z];
vDel[{vx_, vy_, vz_}, {fx_, fy_, fz_}] :=
 {vx D[fx, x], vy D[fy, y], vz D[fz, z]}
```

perturbation rules: I define some rules that will be used to perturb the fluid equations

In[15]:=
```
perturbationRules =
  { n → ((n0 + ϵ δn[#1, #2, #3, #4]) &),
    vx → ((vx0 + ϵ δvx[#1, #2, #3, #4]) &),
    vy → ((vy0 + ϵ δvy[#1, #2, #3, #4]) &),
    vz → ((vz0 + ϵ δvz[#1, #2, #3, #4]) &),
    jx → ((jx0 + ϵ δjx[#1, #2, #3, #4]) &),
    jy → ((jy0 + ϵ δjy[#1, #2, #3, #4]) &),
    jz → ((jz0 + ϵ δjz[#1, #2, #3, #4]) &),
    ℰx → ((ℰx0 + ϵ δℰx[#1, #2, #3, #4]) &),
    ℰy → ((ℰy0 + ϵ δℰy[#1, #2, #3, #4]) &),
    ℰz → ((ℰz0 + ϵ δℰz[#1, #2, #3, #4]) &),
    ℬx → ((ℬx0 + ϵ δℬx[#1, #2, #3, #4]) &),
    ℬy → ((ℬy0 + ϵ δℬy[#1, #2, #3, #4]) &),
    ℬz → ((ℬz0 + ϵ δℬz[#1, #2, #3, #4]) &)};
```

equilibriumRules: I introduce specific rules that describe the equilibrium — homogeneous density n_0 = constant, no flows $\overrightarrow{v_0}$ = 0, no equilibrium electric field $\overrightarrow{\mathcal{E}_0}$ = 0, uniform equilibrium magnetic field $\overrightarrow{\mathcal{B}_0} = \mathcal{B}_0 \overrightarrow{1_z}$.

In[16]:=
```
equilibriumRules =
 {vx0 → 0, vy0 → 0, vz0 → 0, jx0 → 0, jy0 → 0, jz0 → 0, ℰx0 → 0, ℰy0 → 0, ℰz0 → 0,
  ℬx0 → 0, ℬy0 → 0, ℬz0 → B}
```

Out[16]=
```
{vx0 → 0, vy0 → 0, vz0 → 0, jx0 → 0, jy0 → 0,
 jz0 → 0, ℰx0 → 0, ℰy0 → 0, ℰz0 → 0, ℬx0 → 0, ℬy0 → 0, ℬz0 → B}
```

I define the following function for linearizing an equation

In[17]:=
```
Clear[LinearizeEquation];
LinearizeEquation[lhs_ == rhs_] :=
  Coefficient[ExpandAll[lhs], ϵ] == Coefficient[ExpandAll[rhs], ϵ];
LinearizeEquation[eqnList_List] := LinearizeEquation /@ eqnList;
```

I introduce a plane wave eikonal form for the solutions and some tools for simplifying expressions that arise when this solution type is assumed.

In[20]:=
```
Clear[Eikonal, MakeEikonalRule, ApplyEikonal];
Eikonal[x_, y_, z_, t_] := Exp[I kx x + I ky y + I kz z - I ω t];
MakeEikonalRule[var_[x_, y_, z_, t_]] :=
 var → Function[{x, y, z, t}, vâr Eikonal[x, y, z, t]];
eikonalRules = MakeEikonalRule /@ {δℰx[x, y, z, t],
    δℰy[x, y, z, t], δℰz[x, y, z, t], δℬx[x, y, z, t], δℬy[x, y, z, t],
    δℬz[x, y, z, t], δvx[x, y, z, t], δvy[x, y, z, t], δvz[x, y, z, t],
    δjx[x, y, z, t], δjy[x, y, z, t], δjz[x, y, z, t], δn[x, y, z, t]};
eikonalTerm = Eikonal[x, y, z, t];
ApplyEikonal[eqn_] :=
  Module[{step},
   step[1] = eqn /. eikonalRules;
   step[2] = MapEqn[(# / (I eikonalTerm)) &, step[1]] // ExpandAll];
ApplyEikonal[eqnList_List] := ApplyEikonal /@ eqnList;
```

Introduce the gyrofrequency and plasma frequency

In[26]:=
```
def[ωc] = ωc == q B / (m c);
def[ωp] = ωp^2 == 4 π q^2 n0 / m;
```

Visualization

The magnetized plasma geometry and wave vector

```
Module[{O, ex, ey, ez, Bvec, kVec, T, Vec, G = Graphics3D},
 T[text_, position_] := Text[Style[text, Bold, FontSize → 10], position];
 Vec[vec_] := {Arrowheads[0.05], Arrow[Tube[vec, 0.02]]};
 {O, ex, ey, ez} = {{0, 0, 0}, {1, 0, 0}, {0, 1, 0}, {0, 0, 1}};
 Bvec = {Blue, Vec[{O , 0.75 ez} ], T["B = B e_z", 0.8 ez]};
 kVec =
  {Red, Vec[{O , 0.75 {0.1, 1, 1}} ], T["k = k_x e_x+k_y e_y +k_z e_z ", 0.8 {0.1, 1, 1}]};

 Show[
  G[{{Gray, Arrow[{O, ex}]}, {Gray, Arrow[{O, ey}]},
    {Gray, Arrow[{O, ez}]}, {Bvec, kVec}}, Boxed → False],
  G[{T["x", 1.1 ex], T["y", 1.1 ey], T["z", 1.1 ez]}],

  PlotRange → {{-0.4, 1.4}, {-0.4, 1.4}, {0, 1.4}},
  ViewPoint → {2, 2, 1}, ImageSize → {350, 350}, SphericalRegion → True,
  PlotLabel → Stl["Equilibrium plasma geometry"]]]
```