A classical electromagnetic simulation of plasma filaments in a Dense Plasma Focus

Warwick Dumas, John Guillory, Eric Lerner

July 13, 2016

Contents

Part I
Model and rearrangements

1 Model for 3 species

1.1 Introduction

A 3-fluid model is proposed with deuterons, singly positive deuteron ions, and electrons. We use a transport model which recognises the velocity dependence of electron-ion collisions, and that the heat flux vector is not zero. This model is basically derived from [Golant1977] but with comparisons to other sources in the fully ionised case. We assume that the fluids remain close to Maxwellian throughout. Evolution of $\{n_s, v_s, T_s\}$ for each species s is briefly as follows.

Continuity equation

In the fluid frame:

$$\frac{\partial n_s}{\partial t} = \text{compression}\,[= -n_s \nabla \cdot v_s] + \text{net production rate} \tag{1}$$

Acceleration equation

In the fluid frame:

$$\frac{\partial v_s}{\partial t} = -\frac{1}{nm}\nabla\left(nT\right) - \frac{1}{nm}\nabla\cdot\Pi + \frac{Z_s e}{m_s}\left(E + \frac{v_s \times B}{c}\right) + \frac{\delta v_s}{\delta t} \tag{2}$$

where these terms are

Thermal pressure $-\frac{1}{nm}\nabla\left(nT\right)$ Assuming near-isotropy of temperature and an ideal gas, the equation of state is $P_s = n_s T_s I_{3\times 3}$. Thermal pressure acceleration is then $-\frac{1}{nm}\nabla\left(nT\right)$.

Acceleration due to viscous forces $-\frac{1}{nm}\nabla\cdot\Pi$ For ions and electrons we use the viscous stress tensor from [NRLFormulary], whereas for neutrals we use only a momentum diffusion component. The connection between heat conduction and viscous acceleration is noted in [Braginskii1965].

Acceleration due to Lorentz force $\frac{Z_s e}{m_s}\left(E + \frac{v_s \times B}{c}\right)$ Here $Z_e = -1$, $Z_i = 1$. e is a positive value.

Rate of change due to collisions $\frac{\delta v_s}{\delta t}$ Where the sum is over other species β,

$$\frac{\delta v_s}{\delta t} = \sum_\beta \frac{\delta v_s}{\delta t}_{s\beta} = -\frac{1}{m_s}\sum_\beta R_{s\beta} \tag{3}$$

Notice our definition of R matches [Golant1977] p.185 not [NRLFormulary] p.36 which is a factor n different.

JG: "Inter-species friction also needs to take account of ionisation and charge-exchange."

So far it is to include the effects of ionisation, applied separately to other collisional friction.

Excitations and charge exchange are so far not included in the model.

Temperature evolution equation

In the fluid frame,

$$\frac{\partial T_s}{\partial t} = -\frac{2}{3}T_s\nabla\cdot v_s - \frac{2}{3n}\nabla\cdot q_s - \frac{2}{3n}\Pi:\nabla v_s - \frac{2}{3}\sum_\beta\left(\frac{m_s m_\beta}{m_s+m_\beta}\frac{\delta v_s}{\delta t}_{s\beta}\cdot(v_s-v_\beta)\right) + \frac{\delta T_s}{\delta t} \tag{4}$$

where these terms are

Compressive heating $-\frac{2}{3}T_s\nabla\cdot v_s$ This represents the adiabatic compressive heating of a monatomic (ideal?) gas.

Note also, the change in directed kinetic energy from the acceleration due to thermal pressure:

$$\frac{d}{dt}\left(\mathrm{DKE}\right) = m_s v_s n_s\left(-\frac{1}{n_s m_s}\nabla\left(n_s T_s\right)\right) = -v_s\nabla\left(n_s T_s\right)$$

so that between thermal pressure and compressive heating,

$$\frac{d}{dt}\left(\mathrm{DKE}+\frac{3}{2}nT\right) = -v_s\nabla\left(n_s T_s\right) - n_s T_s\nabla\cdot v_s = -\nabla\cdot\left(n_s T_s v_s\right) \tag{5}$$

the rate of net heat flow into a cell. May be missing a trick here - what frame?

Divergence of heat flux vector $-\frac{2}{3n}\nabla\cdot q_s$ This can give rise to both heat conduction $\frac{2}{3n_s}\nabla\cdot(\kappa_s\nabla T_s)$ and a thermoelectric drift of heat.

Viscous heating $-\frac{2}{3n}\Pi:\nabla v_s$ This will represent additional thermal energy that cancels the change in directed kinetic energy under viscous momentum flow.

Frictional heating, inc. resistive heating $-\frac{2}{3}\sum_\beta\left(\frac{m_s m_\beta}{m_s+m_\beta}\frac{\delta v_s}{\delta t}_{s\beta}\cdot(v_s-v_\beta)\right)$ This arises due to the change in directed kinetic energy on inter-species momentum transfer.

Inter-species heat transfer effect $\frac{\delta T}{\delta t}$ Note that [Golant1977] p.176 states that if velocity-dependent R_{ei} is used then we have velocity-dependent heat transfer likewise. However, they by no means explain quite what it is, except in the 'isotropic random velocity distribution', 5-moment case.

We include separately the effects of ionisation on heat transfer.

Per [Golant1977] eqns (2.20) and (6.69) (the second part of their (6.69) is the frictional heating already mentioned), for the velocity independent case:

$$\frac{\delta T_s}{\delta t} = -\sum_\beta\frac{2m_s m_\beta}{\left(m_s+m_\beta\right)^2}\nu_{s\beta}^{\mathrm{MT}}\left(T_s-T_\beta\right) \tag{6}$$

and in the 5-moment velocity dependent case they just replace ν^{MT} with an averaged collision frequency $\bar{\nu}$.

1.1.1 Field equations

Maxwell's equations read

$$\nabla\cdot E = 4\pi\rho \qquad \text{[Gauss]} \tag{7}$$

$$\nabla\times E = -\frac{1}{c}\frac{\partial B}{\partial t} \qquad \text{[Faraday]} \tag{8}$$

$$\nabla\cdot B = 0 \tag{9}$$

$$\nabla \times B = \frac{4\pi}{c} J + \frac{1}{c}\frac{\partial E}{\partial t} \qquad [\text{Ampere} - \text{Maxwell}] \tag{10}$$

As explained in [Hewett1985, BL91] the Darwin approximation is obtained if we neglect the div-free part of the displacement current. This "converts Maxwell's equations from hyperbolic to elliptic in character", neglecting retardation effects and taking the propagation of purely electromagnetic modes to be instantaneous. Where J_T indicates the 'transverse', ie div-free part of J and E_L indicates the curl-free part of E:

$$\nabla \times B = \frac{4\pi}{c} J_T = \frac{4\pi}{c} J + \frac{1}{c}\frac{\partial E_L}{\partial t} \tag{11}$$

The boundary conditions for the curl-free part of E include that the circuit current $I_z(t)$ follows a prescribed trajectory. We can regard this as adding a potential gradient E field due to the circuit, which scales according to the voltage and results in the circuit current, or, we can regard it as setting the scalar potential ϕ on the electrodes. The only additional field other than that is $B_z += 2.7\text{Gauss}$ due to the earth's magnetic field and an electromagnetic coil. Any surface charge is treated as part of the simulation domain. Here, J_T denotes the transverse part of J; that is the div-free part.

1.1.2 Fields via potentials

Suppose we shall be setting out to find B from Ampere's Law $\frac{4\pi}{c} J = \nabla \times B$ and Gauss's Law for magnetism: $\nabla \cdot B = 0$.

There are two great advantages in approaching B via the magnetic potential A: firstly that it will be a given that $\nabla \cdot B = 0$, and secondly that we can pass to dealing with Laplacians instead of curls. So write, for any vector field A where these expressions are well-defined:

$$\nabla \times \nabla \times A - \nabla(\nabla \cdot A) = -\nabla^2 A$$

Therefore if we find A that solves

$$\nabla^2 A = -\frac{4\pi}{c} J \tag{12}$$

this shall be equivalent with

$$\frac{4\pi}{c} J = \nabla \times \nabla \times A - \nabla(\nabla \cdot A) \tag{13}$$

However, this clearly is a Helmholtz decomposition of J and therefore setting $B = \nabla \times A$, we may conclude that we shall then have

$$\frac{4\pi}{c} J_T = \nabla \times \nabla \times A = \nabla \times B \tag{14}$$

Thus, solving a Poisson equation for A implies that we find a solution to the Ampere-Darwin equation for $B = \nabla \times A$. And,

$$\frac{4\pi}{c} J_L = -\nabla(\nabla \cdot A)$$

This indicates that if J has a part that is not divergence-free, then A is not Coulomb gauge. But that is not important. On the other hand, if $\nabla \cdot J$ is small then so is $\nabla \cdot A$, but again - this assumption is by no means necessary.

For intuition, having Poisson is advantageous since we can think about the solution to Poisson as the equilbrium of a birth-diffusion system, with birth rate J (note that the integral of J over all domain space is vector zero, so an equilibrium does exist). Amongst other things, this reveals that it is reasonable to think of A as smooth even though the fluid profiles are not necessarily smooth.

Now if we deal with the E field, we may express (8) as

$$\nabla \times E = -\frac{1}{c}\frac{\partial(\nabla \times A)}{\partial t} \tag{15}$$

Now this is not equivalent with simply saying $E_T = -\frac{1}{c}\frac{\partial A}{\partial t}$ because we have no idea that $\nabla \cdot A = 0$, ie that "A is Coulomb gauge". But we do know that the remainder of E is a scalar potential gradient, so if we write

$$E = -\nabla\phi - \frac{1}{c}\frac{\partial A}{\partial t} \tag{16}$$

Then we can choose ϕ to achieve (7), although this does not mean $-\nabla^2\phi = 4\pi\rho$ except when $\nabla \cdot A = 0$. (But we said in fact, $\nabla \cdot A$ is such that its gradient is J_L: it is highest at a place charge is accumulating.) We should recognise that $-\nabla\phi$ is not necessarily the electrostatic/longitudinal/curl-free part of E, because we have not said anything about what we are getting from $\nabla \cdot A$.

In summary then, if we can solve

$$\nabla^2 A = -\frac{4\pi}{c}J \tag{17}$$

$$-\nabla^2\phi - \nabla \cdot \left(\frac{1}{c}\frac{\partial A}{\partial t}\right) = 4\pi\rho \tag{18}$$

Then for Darwin, it is sufficient to take

$$B = \nabla \times A \tag{19}$$

$$E = -\nabla\phi - \frac{1}{c}\frac{\partial A}{\partial t} \tag{20}$$

Birdsall and Langdon suggest that e.g. we instead take the curl of J to get $\nabla^2 B = \frac{4\pi}{c}\nabla \times J$. That is not particularly suitable for us, because the unsmooth nature of J means that any attempt to describe its curl might well turn out to be fundamentally the wrong approach.

1.1.3 Boundary conditions

Inner boundary condition for A_z:

Consider the central disk of radius $r_{reverse}$ =2.8cm where the reverse current flows. Within this disk A_z is a solution of Laplace and it has a boundary which might as well have a Dirichlet, and we know that the values on the boundary do not vary by much if at all. (The reasoning here .. no, there is not a Dirichlet given. But we can propose a thought-experiment where there is an iteration between setting the outside values and the inside values.)

Therefore A_z (and ϕ) can be treated as flat, a short distance in from the edge – a Neumann condition.

Outer boundary condition for A_z:

We also put a flat Neumann condition for A_z at the outermost radius. Argument to be given.

Note that given

$$\frac{\partial A_z}{\partial y} = \frac{\partial A_z}{\partial x} = 0$$

then equivalently, $B_x = B_y = 0$ at the outer edge since we are assuming that all $\partial/\partial z$ derivatives are zero:

$$B = \left(\frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z}, \frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x}, \frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y}\right) = \left(\frac{\partial A_z}{\partial y}, -\frac{\partial A_z}{\partial x}, \frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y}\right)$$

Inner boundary condition for A_x, A_y:

A_{xy} within r_{ins} is a solution of Laplace with the property that $A_{xy}(r_{ins}, \theta) = -A_{xy}(r_{ins}, \theta + \pi)$. It follows that $A_{xy}(0,0) = (0,0)$ and that in a central disk, $\frac{\partial A_{xy}}{\partial r} \approx A_{xy}/r$.

(Therefore it follows that an inner mesh does not need to extend far into the anode for us to supply reasonable boundary conditions. Radius such as 2.7 cm is sufficient.)

Outer boundary conditions for A_x, A_y:

Note that $\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y} = B_z$.

$$(\nabla \times A)_{xy} = 0$$

The most convenient choice is that if we slope inwards at the inner edge then we slope outwards at the outer edge with a cancelling slope to make the residuals sum to zero. Since the outer circle is much larger, we want a smaller slope A/r_{outer}.

(What implication does this have for B_z?)

The following argument applies: Write

$$\nabla^2 f = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial f}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 f}{\partial \theta^2} + \frac{\partial^2 f}{\partial z^2}$$

Assume that A_r, A_θ are becoming radial functions as we leave the domain going outwards. Then it follows that

$$\begin{pmatrix} A_x \\ A_y \end{pmatrix}(r_{max}, \theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} A_x \\ A_y \end{pmatrix}(r_{max}, 0)$$

But then

$$\frac{\partial A_x}{\partial \theta} = A_x(r_{max}, 0)\sin\theta + A_y(r_{max}, 0)\cos\theta$$

$$\frac{\partial^2 A_x}{\partial \theta^2} = -A_x(r_{max}, 0)\cos\theta + A_y(r_{max}, 0)\sin\theta = -A_x$$

Therefore

$$0 = \nabla^2 A_x = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial A_x}{\partial r}\right) - \frac{1}{r^2}A_x$$

Write

$$0 = \frac{\partial}{\partial r}\left(r\frac{\partial A_x}{\partial r}\right) - \frac{1}{r}A_x = r\frac{\partial^2 A_x}{\partial r^2} + \frac{\partial A_x}{\partial r} - \frac{1}{r}A_x$$

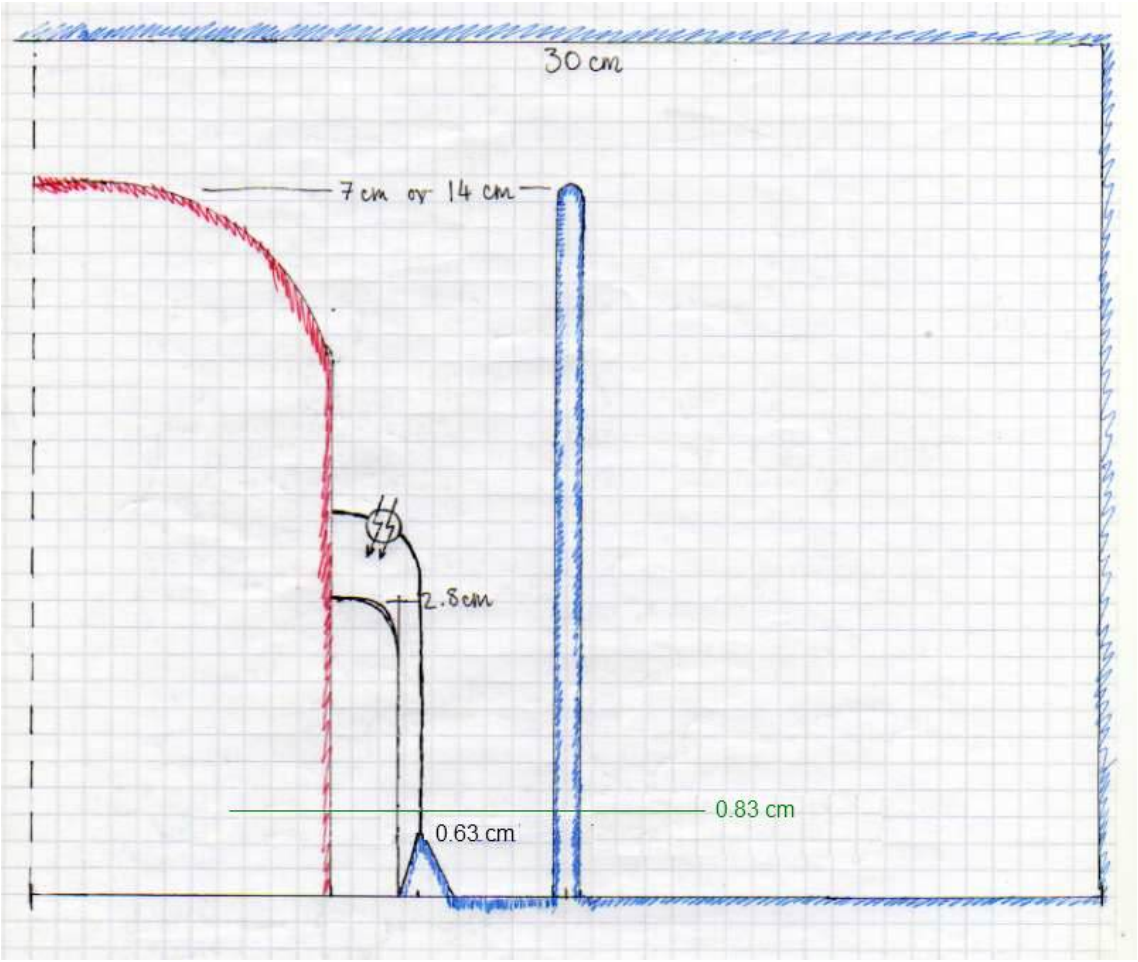
If we try a polynomial then the solution is $A_x = C_1 r + C_2/r$. Obviously $C_1 = 0$.

Boundary conditions for ϕ [not finished - notes]

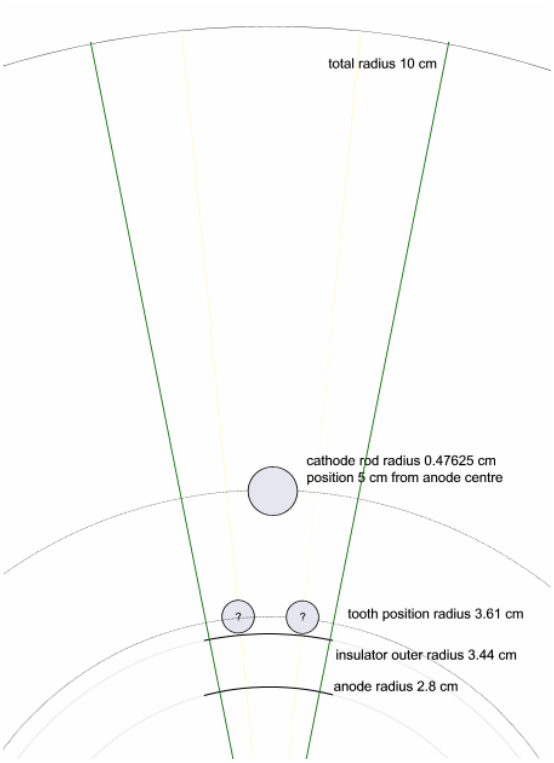
- The potential ϕ needs to be anchored, and we might as well say that in the anode, $\phi = 0$. We assume it is equal across the insulator, underneath any surface charge.
- At the outer boundary assume flat Neumann $\frac{\partial \phi}{\partial r} = 0$. (Was that it? check)

1.1.4 The device physical configuration

The side plan of the experiment is as shown:



and the plan view is as shown:



1.1.5 Converting from magnetic to Cartesian coordinates

Let us be clear on what Cartesian tensor applies when elements are given in magnetic coordinates. Let

$$E_{\|} = \frac{E\cdot B}{\|B\|}b \tag{21}$$

$$E_{\perp} = E - \frac{E\cdot B}{\|B\|}b \tag{22}$$

$$E_{\times} = \frac{E_{\perp}\times B}{\|B\|} = \frac{\left(E-\frac{E\cdot B}{B}b\right)\times B}{\|B\|} = \frac{E\times B}{\|B\|} \tag{23}$$

and suppose we are told e.g. [NOTE MINUS on Hall] something similar to:

$$J = \sigma_{\|}E_{\|} + \sigma_{\perp}E_{\perp} - \sigma_{\wedge}E_{\times} \tag{24}$$

sometimes annotated

$$J = \sigma_{\|}E_{\|} + \sigma_{P}E_{\perp} - \sigma_{H}E_{\times}$$

for 'Pedersen' and 'Hall' conductivities respectively.

First rewrite

$$J = \sigma_{\perp}E + \left(\sigma_{\|}-\sigma_{\perp}\right)\frac{E\cdot B}{\|B\|}b - \sigma_{\wedge}\frac{E\times B}{\|B\|} \tag{25}$$

So remains to write $\frac{E\cdot B}{\|B\|}b$ as E premultiplied by a matrix:

$$\frac{E\cdot B}{\|B\|}b = \frac{1}{B\cdot B}\begin{pmatrix}\left(E_xB_x+E_yB_y+E_zB_z\right)B_x\\ \left(E_xB_x+E_yB_y+E_zB_z\right)B_y\\ \left(E_xB_x+E_yB_y+E_zB_z\right)B_z\end{pmatrix} = \frac{\begin{pmatrix}B_xB_x & B_xB_y & B_zB_x\\ B_yB_x & B_yB_y & B_yB_z\\ B_xB_z & B_yB_z & B_zB_z\end{pmatrix}}{B\cdot B}E \tag{26}$$

and $\frac{E\times B}{\|B\|}$ likewise:

$$-\frac{E\times B}{\|B\|} = \frac{B\times E}{\|B\|} = \frac{1}{\|B\|}\begin{pmatrix}0 & -B_z & B_y\\ B_z & 0 & -B_x\\ -B_y & B_x & 0\end{pmatrix}E \tag{27}$$

Therefore in summary, when someone gives a tensor in magnetic coordinates, what they actually mean is

$$\begin{aligned} J &= \sigma_{\perp}E + \left(\sigma_{\|}-\sigma_{\perp}\right)\frac{E\cdot B}{\|B\|}b - \sigma_{\wedge}\frac{E\times B}{\|B\|}\\ &= \begin{pmatrix}\sigma_{\perp}+\left(\sigma_{\|}-\sigma_{\perp}\right)\frac{B_xB_x}{B\cdot B} & -\sigma_{\wedge}\frac{B_z}{\|B\|}+\left(\sigma_{\|}-\sigma_{\perp}\right)\frac{B_xB_y}{B\cdot B} & \sigma_{\wedge}\frac{B_y}{\|B\|}+\left(\sigma_{\|}-\sigma_{\perp}\right)\frac{B_xB_z}{B\cdot B}\\ \sigma_{\wedge}\frac{B_z}{\|B\|}+\left(\sigma_{\|}-\sigma_{\perp}\right)\frac{B_xB_y}{B\cdot B} & \sigma_{\perp}+\left(\sigma_{\|}-\sigma_{\perp}\right)\frac{B_yB_y}{B\cdot B} & -\sigma_{\wedge}\frac{B_x}{\|B\|}+\left(\sigma_{\|}-\sigma_{\perp}\right)\frac{B_yB_z}{B\cdot B}\\ -\sigma_{\wedge}\frac{B_y}{\|B\|}+\left(\sigma_{\|}-\sigma_{\perp}\right)\frac{B_xB_z}{B\cdot B} & \sigma_{\wedge}\frac{B_x}{\|B\|}+\left(\sigma_{\|}-\sigma_{\perp}\right)\frac{B_yB_z}{B\cdot B} & \sigma_{\perp}+\left(\sigma_{\|}-\sigma_{\perp}\right)\frac{B_zB_z}{B\cdot B}\end{pmatrix}E\\ &= \begin{pmatrix}\sigma_{\perp}+\left(\sigma_{\|}-\sigma_{\perp}\right)b_x^2 & -\sigma_{\wedge}b_z+\left(\sigma_{\|}-\sigma_{\perp}\right)b_xb_y & \sigma_{\wedge}b_y+\left(\sigma_{\|}-\sigma_{\perp}\right)b_xb_z\\ \sigma_{\wedge}b_z+\left(\sigma_{\|}-\sigma_{\perp}\right)b_xb_y & \sigma_{\perp}+\left(\sigma_{\|}-\sigma_{\perp}\right)b_y^2 & -\sigma_{\wedge}b_x+\left(\sigma_{\|}-\sigma_{\perp}\right)b_yb_z\\ -\sigma_{\wedge}b_y+\left(\sigma_{\|}-\sigma_{\perp}\right)b_xb_z & \sigma_{\wedge}b_x+\left(\sigma_{\|}-\sigma_{\perp}\right)b_yb_z & \sigma_{\perp}+\left(\sigma_{\|}-\sigma_{\perp}\right)b_zb_z\end{pmatrix}E \end{aligned} \tag{28}$$

1.2 Collision frequencies

1.2.1 Electron-ion collisions

Averaged collision frequency, from [Golant1977] (5.116), (7.145),

$$\bar{\nu_{ei}} = \frac{4}{3}\sqrt{2\pi}\frac{ne^4}{m_e^{1/2}T_e^{3/2}}\left(\ln\Lambda_e\right) \tag{29}$$

This is the same formula given in [NRLFormulary] p.37 as $1/$"τ_e", the "basic collisional time" in the fully ionised case. This is the averaged frequency that applies when e-i collisions are velocity-dependent.

They give a numerical value

$$\bar{\nu_{ei}} = \frac{1}{3.44 \times 10^5} \frac{n_i \left(\ln \Lambda_e\right)}{T_{e[\mathrm{eV}]}^{3/2}}$$

Note: Not sure what units are being used to give this constant factor. But note that for the formulary, T here is in eV so requires Boltzmann's constant to be introduced.

Our calculation of $\ln \Lambda$ is given by the following. First set

$$\left(\ln \Lambda\right)_{lowT} = 23 - \frac{1}{2} \ln \left(n T_{\mathrm{e[eV]}}^{-3} \right)$$

$$\left(\ln \Lambda\right)_{highT} = 24 - \frac{1}{2} \ln \left(n T_{\mathrm{e[eV]}}^{-2} \right)$$

$$\left(\ln \Lambda\right)_{putative} = \left(\ln \Lambda\right)_{lowT} \left(\frac{1}{2} - 2 \frac{\left|T_{e[eV]} - 10\right| \left(T_{e[eV]} - 10\right)}{1 + 4 \left(T_{e[eV]} - 10\right)^2} \right) + \left(\ln \Lambda\right)_{highT} \left(\frac{1}{2} + 2 \frac{\left|T_{e[eV]} - 10\right| \left(T_{e[eV]} - 10\right)}{1 + 4 \left(T_{e[eV]} - 10\right)^2} \right) \tag{30}$$

$$\left(\ln \Lambda\right)_{used} = \left(\ln \Lambda\right)_{put} + \frac{2}{1 + \frac{1}{2} \left(\ln \Lambda\right)_{put} + \frac{1}{4} \left(\ln \Lambda\right)_{put}^2 + \frac{1}{8} \left(\ln \Lambda\right)_{put}^3 + \frac{1}{16} \left(\ln \Lambda\right)_{put}^4}$$

This way avoided extra calls to exp or log.

Note: It has been found in the past that when implementing that change to a different formula for $T_e > 10$eV, this transition also needs to be done smoothly so as to avoid causing wrinkles - this has thrown us more than once.

There is also a quantum ceiling which is described in Golant et al. Back-of-envelope spreadsheet calculation suggests it is not relevant. It applies in cold dense conditions since T^2/n is what governs it and it acts as a ceiling on $\ln \Lambda$. But with $n = 10^{20}$ and $T = 0.5$eV, we are only down to about 29 as the ceiling.

Ion-ion collisions

Let (as in Golant p.179)

$$\nu_{ii} = \bar{\nu_{ii}} = \frac{4}{3} \frac{\sqrt{\pi}}{\sqrt{m_i}} \frac{e^4 n_i \left(\ln \Lambda_i\right)}{T_i^{3/2}} \tag{31}$$

This matches the NRL Formulary's ν_i. That makes sense: from Braginskii, the "characteristic collision rates" ν_e, ν_i given in the Formulary represent electron-ion and ion-ion rates.

Note that the Formulary's "μ" represents $m_i/m_{proton} = 2$ and so

$$\nu_{ii} = 2^{-1/2} \frac{1}{2.09 \times 10^7} n \left(\ln \Lambda_{ion}\right) T_{\mathrm{ion[eV]}}^{-3/2}$$

Coulomb logarithm Formulary gives, for singly-charged ions, p.34, with T presumably in eV,

$$\lambda_{ii} = 23 - \ln \left(\frac{1}{T_i} \left(2 \frac{n_i}{T_i} \right)^{1/2} \right) = 23 - \frac{1}{2} \ln \left(2 \frac{n_i}{T_i^3} \right) = 23 - \frac{1}{2} \left(\ln 2 + \ln n_i - 3 \ln T_i \right)$$

On the other hand, [Golant1977] p.42 points out that we take the minimum of (T_i, T_e) here (p.39 also). It gives

$$\lambda_{ii} = 23 + \frac{3}{2} \ln T_i - \frac{1}{2} \ln n_i \tag{32}$$

without a $-\frac{1}{2}\ln 2$ in sight. I am going with this since the formulary's version is for "mixed ion-ion collisions".

NRL Formulary states

$$\lambda_{\alpha\beta} = \ln \Lambda_{\alpha\beta} = \ln\left(\frac{r_{max}}{r_{min}}\right)$$

where

$$u = v_\alpha - v_\beta$$

$$r_{min} = max\left\{\frac{e_\alpha e_\beta}{m_{\alpha\beta}\langle u\rangle^2}, \frac{\hbar}{2m_{\alpha\beta}\langle u\rangle}\right\}$$

$$r_{max} = \left(\frac{4\pi\sum n_\gamma e_\gamma^2}{k_B T_\gamma}\right)^{-1/2}$$

where the summation extends over all species γ for which $\langle u\rangle^2 < v_{T_\gamma}^2 = k_B T_\gamma / m_\gamma$. "If either $\langle u\rangle/\omega_{c\alpha} < r_{max}$ or $\langle u\rangle/\omega_{c\beta} < r_{max}$ then the theory breaks down." However see Golant p.40 "becomes invalid at very high magnetic fields when the Larmor radius of the electrons is less than the Debye length. Such fields substantially change the trajectory of colliding electrons' motion at large impact parameters. An allowance for this effect changes the Coulomb logarithm".

Golant then tells

$$\lambda_{\alpha\beta} = \frac{1}{2}\ln\frac{\left(m_\alpha m_\beta/\left(m_\alpha + m_\beta\right)\right)T^2}{e^2\hbar^2 n} = \frac{1}{2}\left(\ln\left(\left(m_\alpha m_\beta/\left(m_\alpha + m_\beta\right)\right)T^2\right) - \ln\left(e^2\hbar^2 n\right)\right)$$

which is to be accepted if it gives a LESSER $\lambda_{\alpha\beta}$. Planck's constant can be 1.0546×10^{-27}erg-seconds or 1.0546×10^{-34} Joule-seconds. This quantum ceiling is high; at $n = 10^{20}$ and $T = 0.5$eV, I make it about 29; it requires cold dense conditions for it to be relevant.

1.2.2 Collisions with neutrals

See [Golant1977] eqn 6.55 for the relative temperature. However, [NRLFormulary] p.39 just says that in the case that neutrals are colder and denser, $\frac{T_s}{m_s}$ is the temperature used.

Neutral-neutral collisions

From [Vallance1], eqn (9.3),

$$\nu_{nn} = \sigma_{nn}\langle\text{speed}\rangle n$$

and $\langle\text{speed}\rangle \propto \sqrt{\frac{T}{m}}$.

It then begs the question what we should take for the cross section σ_{nn}.

Various sources give a description of velocity-dependent cross sections.

(JG: the factor $\sqrt{\frac{8}{\pi}} \approx 1.6$ may be taken into account when cross sections are given.)

In a kinetic model only hard-sphere collisions are considered ([Vallance1] p.17); this would seem to be a lower bound on the cross section. The book [TKBose], which I do not have, appears to contain some information about finding cross sections based on Lennard-Jones.

Also Claire Vallance elsewhere [VallanceReaction] eqn (2.2):

"Landau, Lifshitz and Schiff proposed an expression to relate the collision cross section to the Lennard-Jones parameters ε and r_0 and to the relative velocity v of the colliding particles."

$$\sigma(v) = 8.083\left(\frac{2\varepsilon r_0^6}{\hbar v}\right)^{2/5}$$

But we are seeking a formula for a velocity-independent cross section for now.

There is a database [OakRidgeTableDD] which contains cross sections for, apparently, temperatures. This source is also preferable because as noted below, what we need is the viscosity cross

section and this is given there. Note that elsewhere on Oak Ridge site [OakRidgeHH+] it is stated that 1 atomic unit 'of square distance' is 2.8e-17 cm^2.

That seems to treat elastic collisions. I do not know much about inelastic neutral-neutral collisions.

A question that arises: do we assume that all neutral atoms are D_1 since the high temperatures already lead to molecules breaking down?

Note that [McDaniel1949] p.23 states the following:

> It is now appropriate to introduce two additional cross sections: the diffusion cross section q_D [also called momentum-transfer cross section] and the viscosity cross section q_η [...] q_η appears in the treatment of viscosity and heat conductivity of gases. [...] **The coefficients of viscosity and heat conductivity are inversely related to the viscosity cross-section.** These cross sections are actually of greater significance in the kinetic theory of gases than is $q_{scattering}$ which never appears directly in rigorous discussions of transport phenomena.
>
> [...]
>
> q_D [= momentum-transfer cross section] differs appreciably from the total elastic scattering cross-section only when the scattering is distinctly anisotropic.
>
> [...]

So it is important to use viscosity cross sections in determining κ_n.

Ion-neutral collisions

Subsection 6.8.3 of [KrallTrivelpiece1973], p.321, says Spitzer gives the result for species α hitting (possibly, cold and still) neutrals:

$$\nu_{m\alpha} = \frac{1}{\tau_\alpha} = n_n \sigma_s \left(\frac{T_\alpha}{m_\alpha} \right)^{1/2} \tag{33}$$

(Possibly m stands for momentum transfer?). [NRLFormulary] p.39 has something similar "for scattering of charged particles".

The cross section data here will come from [OakRidgeTableDD+]. Note that [OakRidge64] mentions elastic and inelastic "scattering of D2+ ions by Deuterium".

I think for ion-neutral, we should use the relative temperature [GZS1977] (6.55):

$$T_{\alpha\beta} = \frac{T_\beta m_\alpha + T_\alpha m_\beta}{m_\alpha + m_\beta}$$

and the reduced mass $m_\alpha m_\beta / (m_\alpha + m_\beta)$ so that we get

$$\nu_{in} = n_n \sigma_{in} \left(\frac{T_\beta m_\alpha + T_\alpha m_\beta}{m_\alpha m_\beta} \right)^{1/2} = n_n \sigma_{in} \left(\frac{T_i}{m_i} + \frac{T_n}{m_n} \right)^{1/2}$$

which for the variance of a difference of uncorrelated variables, makes perfect sense.

Electron-neutral collisions

In [Golant1977] p.50, it says that above 3eV, "the collision frequency $\nu_{ea} = n_a s_{ea} v$ is velocity independent; for H, $\nu_{ea} = 1.6 \times 10^{-7} n_a$; for He, $\nu_{ea} = 7 \times 10^{-8} n_a$". It also says that for $T_e < 3$eV, the cross-section can be assumed constant. However, from elsewhere I get $\nu_{en} \approx \sqrt{\frac{m_i}{m_e}} \nu_{in}$. Not sure which source though and I think it's assuming the temperatures are the same.

My best guess conclusion: I will just use the ion-neutral cross sections for electron-neutral, for now, (but evaluated based on electron temperature).

Since collision frequencies do not depend on mass but only on density, $\nu_{ne} = \frac{n}{n_n} \nu_{en}$.

1.3 Ionisation

The rate of ionisation is

$$\begin{aligned} n_e n_n S\left(T_{e[\mathrm{eV}]}\right) &= n_e n_n 10^{-5} \frac{\left(T_{e[\mathrm{eV}]}/E^0_\infty\right)^{1/2}}{\left(E^0_\infty\right)^{3/2}\left(6+\frac{T_{e[\mathrm{eV}]}}{E^0_\infty}\right)} \exp\left(-\frac{E^0_\infty}{T_{e[\mathrm{eV}]}}\right) \\ &= n_e n_n 10^{-5} \frac{T^{1/2}_{e[\mathrm{eV}]}}{E^0_\infty\left(6E^0_\infty + T_{e[\mathrm{eV}]}\right)} \exp\left(-\frac{E^0_\infty}{T_{e[\mathrm{eV}]}}\right) \end{aligned} \tag{34}$$

and we assume that $E^0_\infty \approx 13.6\mathrm{eV}$. Some claim 14.9 ([jspf]) but we think this is wrong for us.

JG: "See Ecker and Kroll; in a dense plasma it's < 13.6".

Radiative recombination is neglected since it is assumed that the energy released is likely to break down a neutral atom nearby. We cannot take it into account without also taking account of that phenomenon.

Therefore the rate of recombination [for sufficiently high degree of ionisation] is the 3-body recombination rate:

$$n_i n_e^2 \alpha_3 = n_i n_e^2 8.75 \times 10^{-27} T^{-4.5}_{e[\mathrm{eV}]} \tag{35}$$

and NOT the full rate from the Formulary:

$$2.7 \times 10^{-13} T^{-1/2}_{e[\mathrm{eV}]} + n_e 8.75 \times 10^{-27} T^{-4.5}_{e[\mathrm{eV}]}$$

Therefore the net ionisation rate is

$$\frac{\partial n}{\partial t} += n_e n_n S\left(T_{e[\mathrm{eV}]}\right) - n_e n_i \left(\alpha_r + n_e \alpha_3\right)$$

1.3.1 Effect of ionisation on temperatures

Apart from potential energy becoming heat, we keep momentum and heat conserved when ionisation takes place. A simple way to achieve this is just to leave the velocity of the new ions and electrons unchanged from the neutral velocity. Meanwhile we assign their temperatures to be $T_n/2$. Then thermal energy is conserved, if defined as

$$\mathcal{E} = \int \frac{3}{2}\left(n_e T_e + n_i T_i + n_n T_n\right) dx$$

This is before the conversion of potential energy of the atoms to the thermal energy density is taken into account.

We subtract $\mathrm{E}^0_\infty \approx 13.6\mathrm{eV}$ for every new ion, and we apply this to T_e. We add $\mathrm{E}^0_\infty \approx 13.6\mathrm{eV}$ for every recombination, and we apply this to T_n.

Assumption 1.3.1.1:

In the recombination case we always share ion and electron heat into neutrals and in the case of ionisation we always share neutral heat into electrons and ions. That is, it is not just the net ionisation rate that has any effect.

Assumption 1.3.1.2:

We change temperature to account for the change in directed kinetic energy when momentum is transferred between species for ionization. Kinetic energy is conserved. Momentum is conserved.

Note: We have not yet tried (and I do not try here) to include the inter-species energy transfer due to charge exchange.

Possibly here is a poor estimate of the heat transfer because we are going to assume that the neutrals that get ionised are sampled in such a way that there is no correlation with the random velocity of the neutral. However, at least it is conserving energy. A more complex treatment is possible; the goal here is to produce something that is workable.

We said

$$\frac{\partial n_{\rm ion}}{\partial t} = \frac{\partial n_{\rm e}}{\partial t} = n_{\rm e} n_{\rm n} S\left(T_{e[eV]}\right) - n_i n_{\rm e}^2 \alpha_3\left(T_{e[eV]}\right) \tag{36}$$

$$S = 10^{-5} \frac{T_{e[eV]}^{1/2}}{E_0\left(6E_0 + T_{e[eV]}\right)} \exp\left(-\frac{E_0}{T_{e[eV]}}\right)$$

We are shortly going to work out the consequences for T and v of assuming conservation of momentum and conservation of energy.

The ionization cooling is

$$\frac{\partial T_e}{\partial t} = -\frac{2}{3n_e}\frac{\partial n_e}{\partial t}\,[13.6\text{eV}] \tag{37}$$

1.3.2 Conserving momentum

Define

$$n_{ionise} = \int_{t_k}^{t_{k+1}} \left.\frac{dn_e}{dt}\right|_{ionising} dt \tag{38}$$

$$n_{recomb} = \int_{t_k}^{t_{k+1}} \left.\frac{dn_n}{dt}\right|_{recombining} dt \tag{39}$$

We want the total of momentum to be the same as beforehand. Assume that ionisation and recombination both transfer momentum between species:

$$v_{ion,k+1} = \frac{\left(n_k^{ion} - n_{recomb}\right) v_{ion,k} + n_{ionise} v_{n,k}}{n_{ion,k+1}} \tag{40}$$

$$v_{e,k+1} = \frac{\left(n_k^{e} - n_{recomb}\right) v_{e,k} + n_{ionise} v_{n,k}}{n_{e,k+1}} \tag{41}$$

$$v_{n,k+1} = \frac{\left(n_{n,k} - n_{ionise}\right) v_{n,k} + n_{recomb}\left(\frac{m_i}{m_n} v_{i,k} + \frac{m_e}{m_n} v_{e,k}\right)}{n_{n,k+1}} \tag{42}$$

It is easy to verify that this represents conservation.

1.3.3 Conserving kinetic energy

For electrons and ions Define the t_{k+1} directed kinetic energy of the ions and electrons:

$$\text{DKE}_{k+1}^{e+i} = \frac{1}{2}\left(m_e n_{k+1}^e v_{e,k+1}^2 + m_{ion} n_{k+1}^{ion} v_{ion,k+1}^2\right) \tag{43}$$

Consider the kinetic energy of the ionising neutrals: assuming (wrong of course) that they have the same velocity distribution as neutrals in general :

$$K_{ionise} = n_{ionise}\left(\frac{1}{2} m_n v_n^2 + \frac{3}{2} T_n\right) \tag{44}$$

Along with the kinetic energy coming from the existing ions and electrons that have not been converted, this forms the total kinetic energy of electrons and ions at time t_{k+1}:

$$K_{k+1}^{e+i} = \left(n_k^e - n_{recomb}\right)\left(\frac{m_e v_{e,k}^2}{2} + \frac{3}{2} T_k^e\right) + \left(n_k^i - n_{recomb}\right)\left(\frac{m_{ion} v_{ion,k}^2}{2} + \frac{3}{2} T_k^{ion}\right) + n_{ionise}\left(\frac{1}{2} m_n v_n^2 + \frac{3}{2} T_n\right)$$

Between the ion and electron thermal energy we now have to make up a difference:

$$\frac{3}{2}\left(n^{ion}_{k+1}T^{ion}_{k+1}+n^{e}_{k+1}T^{e}_{k+1}\right)=K^{e+i}_{k+1}-\mathrm{DKE}^{e+i}_{k+1} \tag{45}$$

There will be a modelling assumption here in how we apportion the additional energy between electrons and ions.

To study the difference $K^{e+i}_{k+1}-\mathrm{DKE}^{e+i}_{k+1}$, we can break it down into terms in v and terms not in v. These latter, which do not appear in DKE, represent just sharing the heat from the converted species:

$$n^{ion}_{k+1}T^{ion}_{k+1}+n^{e}_{k+1}T^{e}_{k+1}=\left(n^{e}_{k}-n_{recomb}\right)T^{e}_{k}+\left(n^{i}_{k}-n_{recomb}\right)T^{ion}_{k}+n_{ionise}\left(T_n\right)+[\text{terms in } v]$$

Thus apart from the terms in v, if we allow that T_n makes an equal contribution to both electron and ion thermal energy, we get:

$$n^{ion}_{k+1}T^{ion}_{k+1}=\left(n^{i}_{k}-n_{recomb}\right)T^{ion}_{k}+\frac{1}{2}n_{ionise}T_n \tag{46}$$

$$n^{e}_{k+1}T^{e}_{k+1}=\left(n^{e}_{k}-n_{recomb}\right)T^{e}_{k}+\frac{1}{2}n_{ionise}T_n \tag{47}$$

Now let us account for the v terms. Using the above expressions for v_{k+1}:

$$n^{e}_{k+1}v^{2}_{e,k+1}=\frac{\left(\left(n^{e}_{k}-n_{recomb}\right)v_{e,k}+n_{ionise}v_{n,k}\right)^{2}}{n_{e,k+1}} \tag{48}$$

$$n^{ion}_{k+1}v^{2}_{ion,k+1}=\frac{\left(\left(n^{ion}_{k}-n_{recomb}\right)v_{ion,k}+n_{ionise}v_{n,k}\right)^{2}}{n_{ion,k+1}} \tag{49}$$

$$\begin{aligned}2\left(K^{e+i}_{k+1}-\mathrm{DKE}^{e+i}_{k+1}\right) &= m_e\left(n^{e}_{k}-n_{recomb}\right)v^{2}_{e,k}+m_{ion}\left(n^{i}_{k}-n_{recomb}\right)v^{2}_{ion,k}+n_{ionise}m_nv^{2}_{n}\\ &\quad -m_e\frac{\left(\left(n^{e}_{k}-n_{recomb}\right)v_{e,k}+n_{ionise}v_{n,k}\right)^{2}}{n_{e,k+1}}-m_{ion}\frac{\left(\left(n^{ion}_{k}-n_{recomb}\right)v_{ion,k}+n_{ionise}v_{n,k}\right)^{2}}{n_{ion,k+1}}\\ &\quad +[\text{terms only in } T]\end{aligned} \tag{50}$$

$$\begin{aligned}&= m_e\frac{n_{ionise}}{n_{e,k+1}}\left(n^{e}_{k}-n_{recomb}\right)v^{2}_{e,k}+m_{ion}\frac{n_{ionise}}{n_{ion,k+1}}\left(n^{ion}_{k}-n_{recomb}\right)v^{2}_{ion,k}\\ &\quad +\left(m_n-\left(\frac{m_e}{n_{e,k+1}}+\frac{m_{ion}}{n_{ion,k+1}}\right)n_{ionise}\right)n_{ionise}v^{2}_{n,k}\\ &\quad -2m_e\frac{\left(n^{e}_{k}-n_{recomb}\right)n_{ionise}}{n_{e,k+1}}v_{e,k}v_{n,k}-2m_{ion}\frac{\left(n^{ion}_{k}-n_{recomb}\right)n_{ionise}}{n_{ion,k+1}}v_{ion,k}v_{n,k}\end{aligned}$$

So the conclusion, assign to $3n^{ion}_{k+1}T^{ion}_{k+1}$:

$$3n^{ion}_{k+1}T^{ion}_{k+1}\ +=\ m_{ion}\frac{n_{ionise}}{n_{ion,k+1}}\left(n^{ion}_{k}-n_{recomb}\right)\left(v_{ion,k}-v_{n,k}\right)^{2} \tag{51}$$

$$3n^{e}_{k+1}T^{e}_{k+1}\ +=\ m_e\frac{n_{ionise}}{n_{e,k+1}}\left(n^{e}_{k}-n_{recomb}\right)\left(v_{e,k}-v_{n,k}\right)^{2} \tag{52}$$

For a time-derivative we can drop the terms that go to zero as fast as h^2:

$$\frac{\partial T_{ion}}{\partial t}\ +=\ \frac{m_{ion}}{3n_{ion}}\left.\frac{dn_e}{dt}\right|_{ionising}\left(v_{ion}-v_n\right)^{2} \tag{53}$$

$$\frac{\partial T_{e}}{\partial t}+=\frac{m_e}{3n_e}\left.\frac{dn_e}{dt}\right|_{ionising}\left(v_e-v_n\right)^{2} \tag{54}$$

For neutrals the new directed kinetic energy is

$$\mathrm{DKE}_{k+1} = m_n n_{k+1}^n v_{n,k+1}^2 = m_n \frac{\left(\left(n_{n,k} - n_{ionise}\right) v_{n,k} + n_{recomb}\left(\frac{m_i}{m_n} v_{i,k} + \frac{m_e}{m_n} v_{e,k}\right)\right)^2}{n_{n,k+1}} \tag{55}$$

whereas the new total kinetic energy is

$$K_{k+1}^{e+i} = \left(n_k^n - n_{ionise}\right)\left(\frac{m_n v_{n,k}^2}{2} + \frac{3}{2} T_k^n\right) + n_{recomb}\left(\frac{1}{2} m_{ion} v_{ion}^2 + \frac{1}{2} m_e v_e^2 + \frac{3}{2} T_{ion} + \frac{3}{2} T_e\right) \tag{56}$$

We want

$$\frac{3}{2} n_{k+1}^n T_{k+1}^n = K_{k+1}^{e+i} - \mathrm{DKE}_{k+1}^{e+i} \tag{57}$$

BUT also recall that where there is a frictional heating involving electrons, we give all that frictional heating to electrons.

First consider the heat sharing:

$$T_{n,k+1} = \frac{\left(n_k^n - n_{ionise}\right) T_k^n + n_{recomb}\left(T_k^e + T_k^{ion}\right)}{n_{n,k+1}} \tag{58}$$

Then we are left with the "frictional" heating:

$$3 n_{k+1}^n T_{k+1}^n + = \left(n_k^n - n_{ionise}\right) m_n v_{n,k}^2 + n_{recomb}\left(m_{ion} v_{ion}^2 + m_e v_e^2\right) - m_n \frac{\left(\left(n_{n,k} - n_{ionise}\right) v_{n,k} + n_{recomb}\left(\frac{m_i}{m_n} v_{i,k} + \frac{n}{n}\right.\right.}{n_{n,k+1}}$$

$$= \frac{n_{recomb}}{n_{n,k+1}}\left(n_k^n - n_{ionise}\right) m_n v_{n,k}^2 - 2 m_n \frac{n_{recomb}\left(n_{n,k} - n_{ionise}\right)}{n_{n,k+1}} v_{n,k}\left(\frac{m_{ion}}{m_n} v_{i,k} + \frac{m_e}{m_n} v_{e,k}\right) + n_{recomb}\left(\left(m_{ion} v_{ion}^2 + m\right.\right.$$

$$= \frac{n_{recomb}}{n_{n,k+1}}\left(n_k^n - n_{ionise}\right) m_n \left(v_{n,k} - \left(\frac{m_{ion}}{m_n} v_{i,k} + \frac{m_e}{m_n} v_{e,k}\right)\right)^2 - \frac{n_{recomb}}{m_n}\left(m_{ion} v_{i,k} + m_e v_{e,k}\right)^2 + n_{recomb}\left(m_{ion} v_{ion}^2 + \right.$$

$$= \frac{n_{recomb}}{n_{n,k+1}}\left(n_k^n - n_{ionise}\right) m_n \left(v_{n,k} - \frac{m_{ion}}{m_n} v_{i,k} - \frac{m_e}{m_n} v_{e,k}\right)^2 + n_{recomb} \frac{m_{ion} m_e}{m_n}\left(v_k^{ion} - v_k^e\right)^2 \tag{59}$$

===

$$= \frac{n_{recomb}}{n_{n,k+1}}\left(n_k^n - n_{ionise}\right) m_n \left(\left(v_{n,k} - \frac{m_{ion}}{m_n} v_{i,k} - \frac{m_e}{m_n} v_{e,k}\right)^2\right) + n_{recomb} \frac{m_{ion} m_e}{m_n}\left(v_k^{ion} - v_k^e\right)^2$$

If we neglect some terms that are going to zero as h^2:

$$3 T_{k+1}^n + \approx \frac{n_{recomb}}{n_{n,k+1}}\left(m_n \left(v_{n,k} - \frac{m_{ion}}{m_n} v_{i,k} - \frac{m_e}{m_n} v_{e,k}\right)^2 + \frac{m_{ion} m_e}{m_n}\left(v_k^{ion} - v_k^e\right)^2\right) \tag{60}$$

Thing is that we somehow wanted to separate the v_e related part from the v_{ion} related part so that we could then decide to add it for T_e instead. Rearrange:

$$3 T_{k+1}^n + \approx \frac{n_{recomb}}{n_{n,k+1}}\left(m_{ion}\left(v_{n,k} - v_{i,k}\right)^2 + m_e\left(v_{n,k} - v_{e,k}\right)^2\right) \tag{61}$$

So instead send the electron part elsewhere:

$$3 T_{k+1}^n + \approx \frac{n_{recomb}}{n_{n,k+1}} m_{ion}\left(v_{n,k} - v_{i,k}\right)^2 \tag{62}$$

$$3T^e_{k+1}+ \approx \frac{n_{recomb}}{n_{e,k+1}} m_e \left(v_{n,k} - v_{e,k}\right)^2 \tag{63}$$

I am not sure about the logic of applying this heating to T_e but it probably doesn't matter.

$$\frac{\partial T_{ion}}{\partial t}+ = \frac{m_{ion}}{3n_{ion}} \left.\frac{dn_e}{dt}\right|_{ionising} \left(v_{ion} - v_n\right)^2 \tag{64}$$

$$\frac{\partial T_n}{\partial t}+ = \frac{m_{ion}}{3n_n} \left.\frac{dn_n}{dt}\right|_{recombining} \left(v_{ion} - v_n\right)^2 \tag{65}$$

$$\frac{\partial T_e}{\partial t}+ = \frac{m_e}{3n_e} \left(\left.\frac{dn_e}{dt}\right|_{ionising} + \left.\frac{dn_n}{dt}\right|_{recombining} \right) \left(v_e - v_n\right)^2 \tag{66}$$

The choice to put frictional heating due to recombination into neutrals and frictional heating due to ionisation into ions, is arbitrary.

It is logical that recombination supplies the same frictional heating as ionisation.

1.3.4 Summary

In summary then, if we write time derivatives, including heat sharing, the frictional heating and ionization heating,

$$\frac{\partial T_n}{\partial t} = \frac{1}{n_n} \left.\frac{dn_n}{dt}\right|_{recombining} \left(T_e + T_{ion} - T_n + \frac{m_{ion}}{3} \left(v_{ion} - v_n\right)^2 + \frac{2}{3} \left[13.6\text{eV}\right] \right) \tag{67}$$

$$\frac{\partial T_{ion}}{\partial t} = \frac{1}{n_{ion}} \left.\frac{dn_{ion}}{dt}\right|_{ionising} \left(\frac{T_n}{2} - T_{ion} + \frac{m_{ion}}{3} \left(v_{ion} - v_n\right)^2 \right) \tag{68}$$

$$\begin{aligned} \frac{\partial T_e}{\partial t} &= \frac{1}{n_e} \left.\frac{dn_e}{dt}\right|_{ionising} \left(\frac{T_n}{2} - T_e - \frac{2}{3} \left[13.6\text{eV}\right] \right) \\ &+ \frac{1}{n_e} \left(\left.\frac{dn_e}{dt}\right|_{ionising} + \left.\frac{dn_n}{dt}\right|_{recombining} \right) \frac{m_e}{3} \left(v_e - v_n\right)^2 \end{aligned} \tag{69}$$

We might wish to redistribute the heating between ions and neutrals, always sharing per inverse particle mass or just equally.

- It can be seen that if it is taken at face value, heat sharing is very important for T_e, comparable magnitude to ionization heating (although, heat soaking for temperature equilibration may be rapid anyway...).

Alternatively, if the ionising and recombining amounts are given, then we can take

$$n_{n,k+1} T_{n,k+1} \approx \left(n^n_k - n_{ionise}\right) T^n_k + n_{recomb} \left(T^e_k + T^{ion}_k + \frac{m_{ion}}{3} \left\| v_i - v_n \right\|^2_k + \frac{2}{3} \left[13.6\text{eV}\right] \right) \tag{70}$$

$$n^{ion}_{k+1} T^{ion}_{k+1} \approx \left(n^{ion}_k - n_{recomb}\right) T^{ion}_k + \frac{1}{2} n_{ionise} T_n + \frac{m_{ion} n_{ionise}}{3} \left\| v_i - v_n \right\|^2_k \tag{71}$$

$$\begin{aligned} n^e_{k+1} T^e_{k+1} &\approx \left(n^e_k - n_{recomb}\right) T^e_k + \frac{1}{2} n_{ionise} T_n + m_e \left(n_{ionise} + n_{recomb}\right) \left\| v_e - v_n \right\|^2_k \\ &- n_{ionise} \frac{2}{3} \left[13.6\text{eV}\right] \end{aligned} \tag{72}$$

One suspects there is a rearrangement including the $O\left(h^2\right)$ terms that I have missed here, and the symmetry of the equations is broken due to throwing away some of those terms. Probably some cancel out. But it doesn't matter much.

1.4 Heat flux vectors

1.4.1 Electron heat flux

From [Golant1977]

$$\frac{5}{2}n\frac{T}{m}\nabla T + q\times\omega_{ce} + 1.87\bar{\nu_{ei}}q - \frac{3}{2}nT\bar{\nu_{ei}}\left(v_e - v_i\right) = 0 \tag{73}$$

J.Guillory: "Other terms neglected; their eqn 9.158"

$$\nu_{ee} + \bar{\nu_{ei}} = 1.87\bar{\nu_{ei}}$$

is the same as the ν_{ei} mentioned previously.

However, we wish to treat the 3-fluid case. In general the stationary state of the heat flux, their (6.80),(9.31):

$$\frac{5}{2}\frac{nT_e}{m}\nabla T_e + q_e\times\omega_{ce} = \frac{\delta q}{\delta t} \tag{74}$$

Their (6.86) reads, assuming velocity-independence for collisions with neutrals but velocity dependence for electron-ion collisions,

$$\frac{\delta q_e}{\delta t} = -\left(\nu_{en} + 1.87\bar{\nu_{ei}}\right)q_e + \frac{3}{2}nT\bar{\nu_{ei}}\left(v_e - v_i\right) \tag{75}$$

Thus, our best general equation for the stationary state of electron heat flux is:

$$\frac{5}{2}n_e\frac{T_e}{m_e}\nabla T_e - q_e\times\omega_{ce} + \left(\nu_{en} + 1.87\bar{\nu_{ei}}\right)q_e - \frac{3}{2}n_eT_e\bar{\nu_{ei}}\left(v_e - v_i\right) = 0 \tag{76}$$

Golant *et al* explicitly state (p.179) that they use "$\bar{\nu_{ei}}$, the averaged collision from (5.116)" so apparently this is not a mistake.

I believe in view of the discussion above, that for ν_{en} we should use the viscosity cross-section, and so have ν_{en}^{η}, because the main place that the heat flux is going to appear is in the heat conduction and we know viscosity cross section should appear there.

Do you agree?

Solve (76) for q: let

$$\nu = \nu_{e\heartsuit} = \nu_{en}^{\eta} + 1.87\bar{\nu_{ei}} \tag{77}$$

Then

$$\left(\nu_{e\heartsuit} - \omega_{ce}\times\right)q = -\frac{5}{2}n\frac{T}{m}\nabla T + \frac{3}{2}T\bar{\nu_{ei}}n_e(v_e - v_i) \tag{78}$$

Recalling (**??**), let

$$\Upsilon_{e\heartsuit} = \left(\Upsilon^{\nu_{en}^{\eta}+1.87\bar{\nu_{ei}}}\right) = \left(\Upsilon^{\nu_{e\heartsuit}}\right) = \nu_{e\heartsuit}\left(\nu_{e\heartsuit} - \omega_{ce}\times\right)^{-1} \tag{79}$$

Then

$$q_e = -\frac{5}{2}\frac{n_eT_e}{m\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e + \frac{3}{2}n_eT_e\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\left(v_e - v_i\right) \tag{80}$$

When we take the divergence $\nabla\cdot q_e$, these two summands give rise to heat conduction and a thermoelectric heat drift effect.

1.4.2 Ion and neutral heat flux

Their eqn (6.87), (6.88), assuming velocity-independent ion-neutral collisions, neglecting electron-ion and taking ion-ion velocity dependent, where

$$\nu_{ii} = \bar{\nu_{ii}} = \frac{4}{3}\frac{\sqrt{\pi}}{\sqrt{m_i}}\frac{e^4n_i\left(\ln\Lambda_i\right)}{T_i^{3/2}} \tag{81}$$

as given above,

$$\frac{\delta q_i}{\delta t} = -\left(\frac{3}{4}\nu_{in} + \frac{4}{5}\nu_{ii}\right) q_i + \frac{1}{4}\frac{n_i}{n_n}\nu_{in} q_n + \frac{5}{8}\nu_{in} n_i \left(T_i - T_n\right)\left(v_i - v_n\right) \tag{82}$$

$$\frac{\delta q_n}{\delta t} = -\left(\frac{3}{4}\nu_{ni} + \frac{1}{4}\nu_{nn}\right) q_n + \frac{1}{4}\frac{n_n}{n_i}\bar{\nu_{ni}} q_i + \frac{5}{8}\nu_{ni} n_n \left(T_n - T_i\right)\left(v_n - v_i\right) \tag{83}$$

No idea what the bar is doing.

Once again it's the question – we should use viscosity cross section for heat conduction and viscosity, therefore do we use it in the heat flux vector?

Stationary states:

$$\frac{5}{2}\frac{n_i T_i}{m_i}\nabla T_i + \omega_{ci} \times q_i - \frac{\delta q_i}{\delta t} = 0 \tag{84}$$

$$\frac{5}{2}\frac{n_n T_n}{m_n}\nabla T_n - \frac{\delta q_n}{\delta t} = 0 \tag{85}$$

Inserting $\frac{\delta q_i}{\delta t}$ and moving some terms across:

$$\omega_{ci} \times q_i + \left(\frac{3}{4}\nu_{in} + \frac{4}{5}\nu_{ii}\right) q_i - \frac{1}{4}\frac{n_i}{n_n}\nu_{in} q_n = -\frac{5}{2}\frac{n_i T_i}{m_i}\nabla T + \frac{5}{8}\nu_{in} n_i \left(T_i - T_n\right)\left(v_i - v_n\right) \tag{86}$$

Inserting $\frac{\delta q_n}{\delta t}$ and moving some terms across:

$$\left(\frac{3}{4}\nu_{ni} + \frac{1}{4}\nu_{nn}\right) q_n - \frac{1}{4}\frac{n_n}{n_i}\bar{\nu_{ni}} q_i = -\frac{5}{2}\frac{n_n T_n}{m_n}\nabla T_n + \frac{5}{8}\nu_{ni} n_n \left(T_n - T_i\right)\left(v_n - v_i\right) \tag{87}$$

We use the second relation to substitute for q_n in the first and get q_n of q_i:

$$q_n = \left(\frac{3}{4}\nu_{ni} + \frac{1}{4}\nu_{nn}\right)^{-1}\left(-\frac{5}{2}\frac{n_n T_n}{m_n}\nabla T_n + \frac{5}{8}\nu_{ni} n_n \left(T_n - T_i\right)\left(v_n - v_i\right) + \frac{1}{4}\frac{n_n}{n_i}\bar{\nu_{ni}} q_i\right) \tag{88}$$

Then

$$\omega_{ci} \times q_i + \left(\frac{3}{4}\nu_{in} + \frac{4}{5}\nu_{ii}\right) q_i - \frac{n_i}{n_n}\nu_{in}\frac{\left(-\frac{5}{2}\frac{n_n T_n}{m_n}\nabla T_n + \frac{5}{8}\nu_{ni} n_n \left(T_n - T_i\right)\left(v_n - v_i\right) + \frac{1}{4}\frac{n_n}{n_i}\bar{\nu_{ni}} q_i\right)}{\left(3\nu_{ni} + \nu_{nn}\right)} = RHS_{ion}$$

Then

$$\omega_{ci} \times q_i + \left(\frac{3}{4}\nu_{in} + \frac{4}{5}\nu_{ii} - \frac{1}{4}\frac{\nu_{in}\bar{\nu_{ni}}}{\left(3\nu_{ni} + \nu_{nn}\right)}\right) q_i =$$

$$n_i \nu_{in}\frac{\left(-\frac{5}{2}\frac{T_n}{m_n}\nabla T_n + \frac{5}{8}\nu_{ni}\left(T_n - T_i\right)\left(v_n - v_i\right)\right)}{\left(3\nu_{ni} + \nu_{nn}\right)} - \frac{5}{2}\frac{n_i T_i}{m_i}\nabla T_i + \frac{5}{8}\nu_{in} n_i \left(T_i - T_n\right)\left(v_i - v_n\right) =$$

$$-\frac{5}{2}\left(\frac{n_i T_i}{m_i}\nabla T_i + \frac{\nu_{in}}{3\nu_{ni} + \nu_{nn}}\frac{n_i T_n}{m_n}\nabla T_n\right) + \frac{5}{8}\nu_{in} n_i \left(1 + \frac{\nu_{ni}}{3\nu_{ni} + \nu_{nn}}\right)\left(T_i - T_n\right)\left(v_i - v_n\right) \tag{89}$$

So where

$$\nu_{i\heartsuit} = \frac{3}{4}\nu_{in} + \frac{4}{5}\nu_{ii} - \frac{1}{4}\frac{\nu_{in}\bar{\nu_{ni}}}{\left(3\nu_{ni} + \nu_{nn}\right)} \tag{90}$$

and

$$\Upsilon^{+}_{i\heartsuit} = \left(\Upsilon\left(\nu_{i\heartsuit}, \omega_{ci}\right)\right)^{Transpose} = \nu_{i\heartsuit}\left(\nu_{i\heartsuit} + \omega_{ci}\times\right)^{-1} \tag{91}$$

$$\begin{aligned} q_i \;=\; & -\frac{5}{2}\frac{n_i T_i}{m_i \nu_{i\heartsuit}}\Upsilon^{+}_{i\heartsuit}\nabla T_i - \frac{5}{2}\left(\frac{\nu_{in}}{3\nu_{ni} + \nu_{nn}}\right)\frac{n_i T_n}{m_n \nu_{i\heartsuit}}\Upsilon^{+}_{i\heartsuit}\nabla T_n \\ & + \frac{5}{8}\frac{\nu_{in}}{\nu_{i\heartsuit}} n_i \left(1 + \frac{\nu_{ni}}{3\nu_{ni} + \nu_{nn}}\right)\left(T_i - T_n\right)\Upsilon^{+}_{i\heartsuit}\left(v_i - v_n\right) \end{aligned} \tag{92}$$

It's not obvious that anything is gained, but we could insert this into the expression for q_n , (88). For simplicity we should either drop it from that expression entirely, or just use the value of q_i to get q_n once the program has calculated q_i. Rearrange:

$$\nu_{n\heartsuit} = \frac{3}{4}\nu_{ni} + \frac{1}{4}\nu_{nn}$$

$$q_n = -\frac{5}{2}\frac{n_n T_n}{m_n \nu_{n\heartsuit}}\nabla T_n + \frac{5}{8}\frac{\nu_{ni}}{\nu_{n\heartsuit}} n_n \left(T_n - T_i\right)\left(v_n - v_i\right) + \frac{1}{4}\frac{n_n}{n_i}\frac{\bar{\nu_{ni}}}{\nu_{n\heartsuit}} q_i \tag{93}$$

and now rewrite

$$\begin{aligned} q_i \;\; = \;\; & -\frac{5}{2}\frac{n_i T_i}{m_i \nu_{i\heartsuit}}\Upsilon^{+}_{i\heartsuit}\nabla T_i - \frac{5}{8}\left(\frac{\nu_{in}}{\nu_{i\heartsuit}}\right)\frac{n_i T_n}{m_n \nu_{n\heartsuit}}\Upsilon^{+}_{i\heartsuit}\nabla T_n \\ & + \frac{5}{8}\frac{\nu_{in}}{\nu_{i\heartsuit}} n_i \left(1 + \frac{\nu_{ni}}{4\nu_{n\heartsuit}}\right)\left(T_i - T_n\right)\Upsilon^{+}_{i\heartsuit}\left(v_i - v_n\right) \end{aligned} \tag{94}$$

1.5 Momentum transfer rates

They write that once the velocity dependence of the momentum-transfer through collisions is recognised, the momentum transfer has two components, one that depends on the velocity difference and one that depends on the temperature gradient. (cf [GZS1977] (6.58))

$$\left.\frac{\partial}{\partial t}\right|_{collisions} v_s = \frac{1}{m_s}\sum_{\beta} R_{s\beta} = \frac{1}{m_s}\sum_{\beta}\left(R^{u}_{s\beta} + R^{T}_{s\beta}\right) \tag{95}$$

They treat separately, R_{ei} and R_{sn}.

1.5.1 Electron-ion R_{ei}

p. 227 "Let us find the numerical values of the effective collision frequencies and the coefficients g^T for the case where the frequencies of collisions with neutral atoms are velocity independent and these collisions make no contribution to the thermal force (....) For this case the collision term representing the effect of electron-ion collisions on the directed velocity of the electrons when their distribution is near-Maxwellian can be found using ... (in the 8-moment approximation): (7.144 - 7.146)

$$\left.\frac{\delta v_e}{\delta t}\right|_{ei} = -\bar{\nu_{ei}}\left(v_e - v_i\right) + \frac{3}{5}\bar{\nu_{ei}}\frac{q_e}{nT_e} \tag{96}$$

where the averaged collision frequency is what is given above. That page concerns the unmagnetised case but later on they state the same thing about the magnetised case, cf [GZS1977] (9.116).

This is the only way that anisotropy of R^{u}_{ei} is introduced by them: anisotropy of the friction part only comes from terms in $(v_e - v_i)$ inside q_e.

Inserting the above expression for q_e then:

$$q_e = -\frac{5}{2}\frac{n_e T_e}{m\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e + \frac{3}{2} n_e T_e \frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\left(v_e - v_i\right)$$

$$\left.\frac{\delta v_e}{\delta t}\right|_{ei} = -\frac{3}{2}\frac{\bar{\nu_{ei}}}{m\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e + \left(\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit} - 1_{3\times 3}\right)\bar{\nu_{ei}}\left(v_e - v_i\right) \tag{97}$$

(Note that this does not yet include the tweak that takes 0.8 to 0.71 in the fully ionised case.)

Note that

$$\left.\frac{\delta v_i}{\delta t}\right|_{ie} = -\frac{m_e n_e}{m_i n_i}\left.\frac{\delta v_e}{\delta t}\right|_{ei}$$

1.5.2 Charged-neutral R_{sn} : velocity-independent case

From [Golant1977] eqn 6.61, where one of s, β is neutrals:

$$\left.\frac{\delta v_s}{\delta t}\right|_{s\beta} = -\sum_{\beta} \frac{m_\beta}{m_s + m_\beta} \nu_{s\beta} \left(v_s - v_\beta\right) \tag{98}$$

Remember that we shall take account of the ionisation contribution to the momentum transfer separately.

1.6 Viscous acceleration

$$\left.\frac{\partial v_i}{\partial t}\right|_{viscosity} = -\frac{1}{n_i m_i} \nabla \cdot \Pi_i \tag{99}$$

For Π_i we use the formula given in [NRLFormulary] but before we continue, we discuss the unmagnetised case and apply some rearrangements. We are given

$$\Pi_{jk} = -\alpha_s \frac{nT}{\nu_{i\Sigma,\eta}} \left(dv_j/dx_k + dv_k/dx_j - \frac{2}{3} \left(1_{[j==k]}\right) \nabla \cdot v \right) \tag{100}$$

where

$$\alpha_s = 0.96\,[\text{ions}]\,;\, 0.73\,[\text{electrons}] \tag{101}$$

We **make an assumption for 2D that** $\frac{d}{dz}\left(\textbf{all}\right) = 0$ but it is clearly not hard to perform these same rearrangements otherwise. Let

$$\Pi_{xx} = -\alpha_s \frac{nT}{\nu_{i\Sigma,\eta}} \left(\frac{4}{3}\frac{\partial v_x}{\partial x} - \frac{2}{3}\frac{\partial v_y}{\partial y} \right) \tag{102}$$

$$\Pi_{xy} = -\alpha_s \frac{nT}{\nu_{i\Sigma,\eta}} \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) \tag{103}$$

We drop the α_s in what follows. The contribution to $\frac{d}{dt}v_x$ involves

$$\begin{aligned} \frac{\partial}{\partial x}\Pi_{xx} + \frac{\partial}{\partial y}\Pi_{xy} &= -\frac{nT}{\nu_{i\Sigma,\eta}} \left(\nabla^2 v_x + \frac{1}{3}\left(\frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_y}{\partial x \partial y} \right) \right) \\ &\quad -\frac{\partial}{\partial x}\left(\frac{nT}{\nu_{i\Sigma,\eta}}\right)\left(\frac{4}{3}\frac{\partial v_x}{\partial x} - \frac{2}{3}\frac{\partial v_y}{\partial y} \right) - \frac{\partial}{\partial y}\left(\frac{nT}{\nu_{i\Sigma,\eta}}\right)\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) \end{aligned}$$

It makes sense to split $\nabla \cdot \Pi$ into terms from v_x and from v_y since necessarily, they are collected separately.

$$\begin{aligned} \left(-1/Mn\right)\left(\nabla \cdot \Pi\right)_x &= \frac{1}{n}\left(\frac{nT}{M\nu}\left(\frac{4}{3}\frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_x}{\partial y^2} \right) + \frac{4}{3}\frac{\partial}{\partial x}\left(\frac{nT}{M\nu}\right)\frac{\partial v_x}{\partial x} + \frac{\partial}{\partial y}\left(\frac{nT}{M\nu}\right)\frac{\partial v_x}{\partial y} \right. \\ &\quad \left. + \frac{1}{3}\frac{nT}{M\nu}\frac{\partial^2 v_y}{\partial x \partial y} - \frac{2}{3}\frac{\partial}{\partial x}\left(\frac{nT}{M\nu}\right)\frac{\partial v_y}{\partial y} + \frac{\partial}{\partial y}\left(\frac{nT}{M\nu}\right)\frac{\partial v_y}{\partial x} \right) \end{aligned} \tag{104}$$

We may recognise that the v_x part represents diffusion (with ellipsoid contours having a major axis in the x-direction).

1.7 Heat flux divergence

1.7.1 Electron heat flux divergence

$$\frac{\partial T_e}{\partial t} += \frac{1}{n_e}\nabla\cdot\left(\frac{5}{3}\frac{n_e T_e}{m\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e\right) - \frac{1}{n_e}\nabla\cdot\left(n_e T_e\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\left(v_e - v_i\right)\right) \tag{105}$$

The first term is heat conduction so put

$$\kappa_\parallel^e = \frac{5}{2}\frac{n_e T_e}{m\nu_{e\heartsuit}} \tag{106}$$

The second term clearly represents a thermoelectric drift of heat, at a velocity

$$\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\left(v_e - v_i\right)$$

1.7.2 Ion heat flux divergence

We said

$$\begin{aligned} q_i &= -\frac{5}{2}\frac{n_i T_i}{m_i\nu_{i\heartsuit}}\Upsilon^+_{i\heartsuit}\nabla T_i - \frac{5}{8}\left(\frac{\nu_{in}}{\nu_{i\heartsuit}}\right)\frac{n_i T_n}{m_n\nu_{n\heartsuit}}\Upsilon^+_{i\heartsuit}\nabla T_n \\ &\quad + \frac{5}{8}\frac{\nu_{in}}{\nu_{i\heartsuit}}n_i\left(1+\frac{\nu_{ni}}{4\nu_{n\heartsuit}}\right)\left(T_i - T_n\right)\Upsilon^+_{i\heartsuit}\left(v_i - v_n\right) \end{aligned} \tag{107}$$

$$\begin{aligned} \frac{\partial T_i}{\partial t} \; += \; & \frac{1}{n_i}\nabla\cdot\left(\frac{5}{3}\frac{n_i T_i}{m_i\nu_{i\heartsuit}}\Upsilon^+_{i\heartsuit}\nabla T_i\right) + \frac{1}{n_i}\nabla\cdot\left(\frac{5}{12}\left(\frac{\nu_{in}}{\nu_{i\heartsuit}}\right)\frac{n_i T_n}{m_n\nu_{n\heartsuit}}\Upsilon^+_{i\heartsuit}\nabla T_n\right) \\ & -\frac{1}{n_i}\nabla\cdot\left(\frac{5}{12}\frac{\nu_{in}}{\nu_{i\heartsuit}}\left(\frac{4\nu_{n\heartsuit}+\nu_{ni}}{4\nu_{n\heartsuit}}\right)n_i\left(T_i - T_n\right)\Upsilon^+_{i\heartsuit}\left(v_i - v_n\right)\right) \end{aligned} \tag{108}$$

The first two terms are ion heat conduction and, an effect of the curvature of neutral temperatures.

That second term looks to be net energy neutral for the same reason that the neutral heat conduction is net energy neutral. Let

$$\kappa_\parallel^{\text{ion}} = \frac{5}{2}\frac{n_i T_i}{m_i\nu_{i\heartsuit}} \tag{109}$$

$$\kappa_\parallel^{\text{i}\leftarrow\text{n}} = \frac{1}{4}\left(\frac{\nu_{in}n_i}{\nu_{i\heartsuit}n_n}\right)\frac{n_n T_n}{m_n\nu_{n\heartsuit}} \tag{110}$$

The third term appears problematic. The T_i part is fine, it represents a drift of ion heat similar to the thermoelectric drift of electron heat. But for the T_n part not to generate energy, it looks like we need some kind of answering term in neutrals - maybe for now it is better to argue for when this second term is small.

==

More sensible way perhaps, if we wanted it simple: go again, drop the effect of q_i on q_n from the start, let $q_n \Rightarrow q_i$, instead of solving simultaneously.

==

For now let's just drop the neutral -> ion effect.

==

To turn the T_i part into a drift:

$$-\frac{1}{n_i}\nabla\cdot\left(\frac{5}{12}\frac{\nu_{in}}{\nu_{i\heartsuit}}\left(\frac{4\nu_{n\heartsuit}+\nu_{ni}}{4\nu_{n\heartsuit}}\right)n_i\left(T_i - T_n\right)\Upsilon^+_{i\heartsuit}\left(v_i - v_n\right)\right)$$

but that leaves behind the T_n part that probably nearly balanced it.

1.7.3 Neutral heat flux divergence

We want to write something like, in the fluid frame,

$$\frac{\partial}{\partial t}T_{neutral} \quad += \quad \frac{2}{3n_n}\nabla\cdot\left(\kappa_{neutral}\nabla T_{neutral}\right)$$

We need to define κ_n. First let's look according to gas references.
According to [Vallance1],

$$\kappa = \frac{1}{2}\lambda_{mfp}\left\langle \text{speed}\right\rangle n_n$$

where λ_{mfp} is the mean free path. So in the absence of ions,

$$\kappa_n = \frac{1}{2}\frac{T_n n_n}{m_n \nu_{nn,\eta}}$$

Recall that [McDaniel1949] states that it is important to use viscosity cross sections in determining κ_n. Now suppose we also want to take account of the effects of collisions with charged particles on the mean free path. One simple way:

$$\lambda_{mfp} = \frac{\left\langle \text{speed}\right\rangle}{\nu_{nn,\eta}+\nu_{ni,\eta}+\nu_{ne,\eta}}$$

$$\kappa_n = \frac{1}{2}\frac{T_n n_n}{m_n\left(\nu_{nn,\eta}+\nu_{ni,\eta}+\nu_{ne,\eta}\right)} \tag{111}$$

Note that this is self-consistent in the following sense: if we split up a gas into two notional components that are actually the same, we get back the same result for κ as by looking at it as one gas.

==

Meanwhile, if instead we look from the perspective of $\nabla\cdot q_n$,

$$\left(\frac{3}{4}\nu_{ni}+\frac{1}{4}\nu_{nn}\right)q_n - \frac{1}{4}\frac{n_n}{n_i}\bar{\nu_{ni}}q_i = -\frac{5}{2}\frac{n_n T_n}{m_n}\nabla T_n + \frac{5}{8}\nu_{ni}n_n\left(T_n-T_i\right)\left(v_n-v_i\right)$$

If we drop the contribution from q_i, rightly or wrongly, then we'd have

$$q_n = -\frac{5}{2}\frac{n_n T_n}{m_n\left(\frac{3}{4}\nu_{ni}+\frac{1}{4}\nu_{nn}\right)}\nabla T_n + \frac{5}{8}\frac{\nu_{ni}}{\frac{3}{4}\nu_{ni}+\frac{1}{4}\nu_{nn}}n_n\left(T_n-T_i\right)\left(v_n-v_i\right) \tag{112}$$

Why does the ∇T_n term disagree sharply with what we get for a neutral gas?
If $\nu_{ni}=0$ then we would have

$$\frac{\partial T_n}{\partial t} - \frac{2}{3n_n}10\nabla\cdot\frac{n_n T_n}{m_n\nu_{nn}^{\eta}}\nabla T_n \tag{113}$$

which is many times greater than advertised.

Something is dramatically out of whack somewhere between these two sources.

Golant et al (7.25) confirms the factor 10.

$$\kappa^{\text{neutral}}_{[\text{GSKB}]} = 10\frac{n_n T_n}{m_n\nu_{nn}^{\eta}} \tag{114}$$

1.8 Frictional heating

Note that

$$\frac{d}{dt}\left(\text{DKE}\right) = m_\beta v_\beta n_\beta \frac{\delta v_\beta}{\delta t}_{\beta s} + m_s v_s n_s \frac{\delta v_s}{\delta t}_{s\beta} = m_s n_s \frac{\delta v_s}{\delta t}_{s\beta}\cdot\left(v_s-v_\beta\right) \tag{115}$$

Ion-neutral case Let

$$\frac{dv_s}{dt} = \frac{m_{s'}}{m_s + m_{s'}} \nu_{ss'} \left(v_{s'} - v_s\right)$$

$$\text{DKE} = \frac{1}{2} m_s n_s v_s^2 + \frac{1}{2} m_{s'} n_{s'} v_{s'}^2$$

$$\begin{aligned}
\frac{d}{dt}\left(\text{DKE}\right) &= \frac{m_s m_{s'}}{m_s + m_{s'}} \left(v_s n_s \nu_{ss'} \left(v_{s'} - v_s\right) + v_{s'} n_{s'} \nu_{s's} \left(v_s - v_{s'}\right)\right) \\
&= \frac{m_s m_{s'}}{m_s + m_{s'}} \frac{\nu_{ss'}}{n_s'} \left(v_s n_s n_{s'} \left(v_{s'} - v_s\right) + v_{s'} n_{s'} n_s \left(v_s - v_{s'}\right)\right) \qquad (116) \\
&= \frac{m_s n_s m_{s'} n_{s'}}{m_s + m_{s'}} \left(\frac{\nu_{ss'}}{n_s'}\right) \left(-\left(v_{s'} - v_s\right)^2\right) < 0 \qquad (117)
\end{aligned}$$

We are then, I think, told (in a place in Golant I can no longer find) to apportion heat energy according to mass of other particle:

$$\frac{\partial T_s}{\partial t} = \frac{2}{3 n_s} \frac{m_{s'}}{\left(m_s + m_{s'}\right)} \frac{m_s n_s m_{s'} n_{s'}}{\left(m_s + m_{s'}\right)} \left(\frac{\nu_{ss'}}{n_s'}\right) \left(v_{s'} - v_s\right) \cdot \left(v_{s'} - v_s\right)$$

$$\frac{\partial T_s}{\partial t} = \frac{2}{3} \frac{m_s m_{s'}^2}{\left(m_s + m_{s'}\right)^2} \nu_{ss'} \left(v_{s'} - v_s\right)^2 \qquad (118)$$

Electron-neutral case For simplicity we just apportion all the additional heat to electrons:

$$\frac{\partial T_e}{\partial t} += \frac{2}{3} \frac{m_e m_n}{m_e + m_n} \nu_{en} \left(v_n - v_e\right)^2 \qquad (119)$$

Electron-ion case Alternatively suppose we calculated a general collisional momentum-transfer rate $\frac{\delta v_e}{\delta t}_{ei}$. Then since

$$\frac{\delta v_i}{\delta t}_{ie} = -\frac{\delta v_e}{\delta t}_{ei} \frac{m_e n_e}{m_i n_i}$$

$$\frac{d}{dt}\left(\text{DKE}\right) = m_i v_i n_i \frac{\delta v_i}{\delta t}_{ie} + m_e v_e n_e \frac{\delta v_e}{\delta t}_{ie} = m_e n_e \frac{\delta v_e}{\delta t}_{ei} \cdot \left(v_e - v_i\right) \qquad (120)$$

Put all that heat into electrons – a reasonable approximation; $1/n_e$ cancels, so

$$\frac{\partial T_e}{\partial t} = -\frac{2}{3} m_e \frac{\delta v_e}{\delta t}_{ei} \cdot \left(v_e - v_i\right) \qquad (121)$$

So suppose we worked out $\frac{\delta v_e}{\delta t}_{ei}$ and we want to get frictional heating; it's then easy.

1.8.1 Full expression for electron "heating": rearranging thermoelectric

We might sometimes be concerned that terms are being rolled up in $\frac{\delta v_e}{\delta t}_{ei} \cdot \left(v_e - v_i\right)$ that are better treated as some kind of drift of heat. Using (97) the "frictional heating" term due to electron-ion collisions is then

$$\frac{\partial T_e}{\partial t} += -\frac{2}{3} m_e \left(-\frac{3}{2} \frac{\bar{\nu_{ei}}}{m \nu_{e\heartsuit}} \Upsilon_{e\heartsuit} \nabla T_e + \left(\frac{9}{10} \frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}} \Upsilon_{e\heartsuit} - 1_{3\times 3}\right) \bar{\nu_{ei}} \left(v_e - v_i\right)\right) \cdot \left(v_e - v_i\right)$$

$$\frac{\partial T_e}{\partial t} += \left(\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}} \Upsilon_{e\heartsuit} \nabla T_e\right) \cdot \left(v_e - v_i\right) + \frac{2 m_e}{3} \bar{\nu_{ei}} \left(\left(1_{3\times 3} - \frac{9}{10} \frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}} \Upsilon_{e\heartsuit}\right) \left(v_e - v_i\right)\right) \cdot \left(v_e - v_i\right) \qquad (122)$$

The second term is resistive heating; the first term can, if we wish, be rearranged as a bonus thermoelectric-type drift:

$$\left(\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e\right)\cdot(v_e-v_i)=(v_e-v_i)^{Transpose}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e=\left(\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon^{+}_{e\heartsuit}(v_e-v_i)\right)\cdot\nabla T_e \qquad (123)$$

Now recognise what happens when we try to combine this with the thermoelectric heat drift from the heat flux divergence: first split that up

$$-\frac{1}{n}\nabla\cdot\left(n_eT_e\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}(v_e-v_i)\right)=-\left(\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}(v_e-v_i)\right)\cdot\nabla T_e-\frac{1}{n}T_e\nabla\cdot\left(n\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}(v_e-v_i)\right)$$

The opposing signs on the Hall components mean that if we can cancel our bonus drift against $-\left(\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon^{-}_{e\heartsuit}(v_e-v_i)\right)\cdot\nabla T_e$ then the parallel and Pedersen components are cancelled and we are left with only, a doubled Hall drift of T_e:

$$-\left(\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon^{-}_{e\heartsuit}(v_e-v_i)\right)\cdot\nabla T_e+\left(\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon^{+}_{e\heartsuit}(v_e-v_i)\right)\cdot\nabla T_e=\left(2\frac{\bar{\nu_{ei}}}{\left(\nu^2_{e\heartsuit}+\omega^2\right)}(v_e-v_i)\times\omega_e\right)\cdot\nabla T_e$$

However I can then find nothing useful to do with the term that is then left over. Viz, what we have shown, alternative expressions for the total thermoelectric term in $\frac{\partial T_e}{\partial t}$:

$$\begin{aligned}\frac{\partial T_e}{\partial t}\ +=\ & -\frac{1}{n}\nabla\cdot\left(n_eT_e\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}(v_e-v_i)\right)+\left(\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon^{+}_{e\heartsuit}(v_e-v_i)\right)\cdot\nabla T_e\\ =\ & \left(2\frac{\bar{\nu_{ei}}}{\left(\nu^2_{e\heartsuit}+\omega^2\right)}(v_e-v_i)\times\omega\right)\cdot\nabla T_e-\frac{1}{n}T_e\nabla\cdot\left(n\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}(v_e-v_i)\right)\end{aligned} \qquad (124)$$

It probably needs to be considered how, between shifting heat and applying a compressive heating factor, these sorts of expressions can be achieved. If we are content to roll up frictional "heating" (121) then the rest is easy, a straightforward drift of heat; but this might not always be best.

Of course, it is clear why (in the presence of density gradients) $\left(\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon^{+}_{e\heartsuit}(v_e-v_i)\right)\cdot\nabla T_e$ can be a net energy term, whereas the heat flux thermoelectric drift is energy neutral.

1.9 Inter-species heat transfers

Note an important fact, [Braginskii1965] p.277: for inter-species transfer of heat it is always the momentum-transfer cross-section that is relevant.

1.9.1 Try to work out heat transfer in the 'anisotropic random velocity distribution', 8-moment case:

"In the velocity-dependent case we have other terms." Leaving ionisation separate, from (p.174):

$$n\frac{\delta T}{\delta t}=\frac{2}{3}\frac{\delta\left(n\left\langle K\right\rangle\right)}{\delta t}-mnu\frac{\delta u}{\delta t}$$

That latter term seems to be there to cancel off the change in directed kinetic energy.

They do not present the solution in the case that the random velocity distribution is not assumed to be isotropic.

==

Note: they work in total kinetic energy and this seems the only viable way - we have to recognise afterwards, which terms are contributing to random energy.

==

In view of their (6.67) and (6.71), which assume that the random velocity distribution IS isotropic although the collision frequency velocity-dependent:

$$\left(\frac{\delta\left(n_s m_s\left\langle\frac{w_s^2}{2}\right\rangle\right)}{\delta t}\right)_{s\beta} \quad =? \quad -\frac{m_s m_\beta}{m_s+m_\beta} n_s\left\langle w_0 w \nu_{s\beta}^{MT}(v)\right\rangle$$

$$= \quad -\frac{m_s m_\beta}{m_s+m_\beta} n_s \bar{\nu_{s\beta}}\left(\frac{T_s-T_\beta}{m_s+m_\beta}\right)$$

since it is noted after (6.68) that the second term there is just changing the directed energy.

$$\frac{\delta T_s}{\delta t} = -\frac{2}{3}\frac{m_s m_\beta}{(m_s+m_\beta)^2}\bar{\nu_{s\beta}}\left(T_s-T_\beta\right)$$

Why factor of $\frac{1}{3}$ relative to their velocity-independent expression?

Heat flux is defined as (6.14)

$$q = \frac{nm}{2}\langle w\,(w\cdot w)\rangle$$

"characterizes the energy transfer in a system where the directed velocity is zero".

Random velocity near Maxwellian: (6.35) [(8-moment: 5 indt viscosity stress tensor terms would be 13-moment - see p.164]

$$f = \left(\frac{m}{2\pi T}\right)^{3/2}\exp\left(-\frac{mw^2}{2T}\right)\left(1-\sum_k q_k\frac{mw_k}{nT^2}\left(1-\frac{mw^2}{5T}\right)\right)$$

The total kinetic energy change for the species seems to include

$$\left\langle v_0 v \nu_{s\beta}^{MT}(v)\right\rangle = \left\langle\left(u_0+w_0\right)\left(u+w\right)\nu_{s\beta}^{MT}(v)\right\rangle$$

where w_s, w_β have been turned into a centre of inertia velocity w_0 and relative velocity $w \perp w_0$.

I wasn't able to complete this part at this time. It requires another big look.

2 Summary of model differential equations

Of course they are not used in this form.

Base coefficients

Let

$$\omega_{ce} = \frac{eB}{m_e c} \qquad ; \qquad \omega_{ci} = \frac{eB}{m_i c}$$

$$\nu_{ei} = \bar{\nu_{ei}} = \frac{1}{3.44 \times 10^5} \frac{n_e \left(\ln \Lambda_e\right)}{T_{e[\mathrm{eV}]}^{3/2}}$$

$$\nu_{ii} = 2^{-1/2} \frac{1}{2.09 \times 10^7} \frac{n \left(\ln \Lambda_{ion}\right)}{T_{\mathrm{ion}[\mathrm{eV}]}^{3/2}}$$

$$\nu_{en}^{\eta} = n_n \sigma_{en}^{\eta} \left(\frac{T_e}{m_e}\right)^{1/2}$$

$$\nu_{en}^{\mathrm{MT}} = n_n \sigma_{en}^{\mathrm{MT}} \left(\frac{T_e}{m_e}\right)^{1/2}$$

$$\nu_{in} = n_n \sigma_{in} \left(\frac{T_i}{m_i} + \frac{T_n}{m_n}\right)^{1/2}$$

$$\nu_{e\heartsuit} = \nu_{en}^{\eta} + 1.87 \bar{\nu_{ei}}$$

$$\nu_{i\heartsuit} = \frac{3}{4}\nu_{in}^{\eta} + \frac{4}{5}\nu_{ii} - \frac{1}{4}\frac{\nu_{in}^{\eta}\nu_{ni}^{\eta}}{\left(3\nu_{ni}^{\eta} + \nu_{nn}^{\eta}\right)}$$

$$\nu_{n\heartsuit} = \frac{3}{4}\nu_{ni}^{\eta} + \frac{1}{4}\nu_{nn}^{\eta}$$

$$\Upsilon_{e\heartsuit} = \Upsilon_{e\heartsuit}^{-} = \Upsilon\left(\nu_{e\heartsuit}, \omega_{ce}\right) = \nu_{e\heartsuit}\left(\nu_{e\heartsuit} - \omega_{ce}\times\right)^{-1}$$

$$\Upsilon_{i\heartsuit}^{+} = \Upsilon\left(\nu_{i\heartsuit}, \omega_{ci}\right)^{Transpose} = \nu_{i\heartsuit}\left(\nu_{i\heartsuit} + \omega_{ci}\times\right)^{-1}$$

Continuity equation

In fluid frame,

$$\frac{\partial n_s}{\partial t} = -n_s \nabla \cdot v_s + \text{net production rate}$$

Ion acceleration

In fluid frame:

$$\begin{aligned}
\frac{\partial v_i}{\partial t} \quad = \quad & -\frac{1}{n_i m_i}\nabla\left(n_i T_i\right) - \frac{1}{n_i m_i}\nabla\cdot\Pi_i + \frac{e}{m_i}E - \omega_{ci}\times v_i \\
& + [\text{effect of new ions at } v_n] - \frac{m_n}{m_i + m_n}\nu_{in}^{\mathrm{MT}}\left(v_i - v_n\right) \\
& + \frac{3}{2}\frac{\nu_{ie}}{m_i \nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e - \frac{m_e}{m_i}\left(1_{3\times 3} - \frac{9}{10}\frac{\nu_{ei}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\nu_{ie}\left(v_i - v_e\right)
\end{aligned} \tag{125}$$

since

$$\left.\frac{\delta v_i}{\delta t}\right|_{ie} = -\frac{m_e n_e}{m_i n_i}\left.\frac{\delta v_e}{\delta t}\right|_{ei}$$

Electron acceleration

In fluid frame:

$$\begin{aligned}\frac{\partial v_e}{\partial t} &= -\frac{1}{n_e m_e}\nabla\left(n_e T_e\right) - \frac{1}{n_e m_e}\nabla\cdot\Pi_e - \frac{e}{m_e}E + \omega_{ce}\times v_e \\ &+ \left[\text{effect of new e's at } v_n\right] - \frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}\left(v_e - v_n\right) \\ &-\frac{3}{2}\frac{\nu_{ei}}{m\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e - \left(1_{3\times 3} - \frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\nu_{ei}\left(v_e - v_i\right)\end{aligned} \tag{126}$$

Neutral acceleration

In fluid frame:

$$\begin{aligned}\frac{\partial v_n}{\partial t} &= -\frac{1}{n_n m_n}\nabla\left(n_n T_n\right) - \frac{1}{n_n m_n}\nabla\cdot\Pi_n + \left[\text{effect of new neutrals at } \tfrac{m_i v_i + m_e v_e}{m_n}\right] \\ &-\frac{m_e}{m_e+m_n}\nu_{ne}^{\mathrm{MT}}\left(v_n - v_e\right) - \frac{m_i}{m_i+m_n}\nu_{ni}^{\mathrm{MT}}\left(v_n - v_i\right)\end{aligned} \tag{127}$$

where $\nabla\cdot\Pi_n$ gives rise to isotropic viscosity with no Coulombic part, ie, p_x does not affect p_y.

Ion temperature evolution

In fluid frame,

$$\begin{aligned}\frac{\partial T_i}{\partial t} &= -\frac{2}{3}T_i\nabla\cdot v_i + \frac{2}{3n_i}\nabla\cdot\left(\kappa^{\mathrm{ion}}\nabla T_i\right) - \frac{2}{3n_i}\Pi:\nabla v_i + \frac{2}{3}\frac{m_i m_n^2}{\left(m_i+m_n\right)^2}\nu_{in}\left(v_n - v_i\right)^2 \\ &+ \left[\text{effect of new ions at } \tfrac{T_n}{2}\right] - \sum_{\beta\neq i}\frac{2m_i m_\beta}{\left(m_i+m_\beta\right)^2}\nu_{i\beta}^{\mathrm{MT}}\left(T_i - T_\beta\right)\end{aligned} \tag{128}$$

where

$$\kappa^{\mathrm{ion}} = \kappa_{\parallel}^{\mathrm{ion}}\Upsilon_{i\heartsuit}^{+} = \frac{5}{2}\frac{n_i T_i}{m_i\nu_{i\heartsuit}}\Upsilon_{i\heartsuit}^{+} \tag{129}$$

and the effect of $\left(T_i - T_n\right)\left(v_i - v_n\right)$ and ∇T_n on $\frac{\partial}{\partial t}T_i$ via $\nabla\cdot q_i$ has here been neglected (although q_n affected $\nu_{i\heartsuit}$).

The term $\frac{2}{3n_i}\Pi:\nabla v_i$ is best captured via the rate of change of directed kinetic energy on a viscous flow:

And so far I did not get beyond the velocity-independent inter-species heat transfer formula.

Electron temperature evolution

In fluid frame,

$$\begin{aligned}
\frac{\partial T_e}{\partial t} &= \\
\text{compressive htg} \quad & -\frac{2}{3}T_e\nabla\cdot v_e \qquad (130)\\
\text{heat conduction} \quad & +\frac{2}{3n_e}\nabla\cdot\left(\kappa^e\nabla T_e\right)\\
\text{viscous htg} \quad & -\frac{2}{3n_e}\Pi:\nabla v_e\\
\text{thermoelectric heat drift} \quad & -\frac{1}{n_e}\nabla\cdot\left(n_eT_e\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\left(v_e-v_i\right)\right)\\
\text{frictional+resistive} \quad & +\frac{2}{3}\frac{m_em_n}{m_e+m_n}\nu_{en}\left(v_n-v_e\right)^2-\frac{2}{3}m_e\frac{\delta v_e}{\delta t}_{ei}\cdot\left(v_e-v_i\right)\\
\text{ionisation-related} \quad & +\left[\text{effect of new e's at } \tfrac{T_n}{2}\right]+\frac{2\mathrm{E}^0_\infty}{3n_e}\left[\text{net ion production rate}\right]\\
\text{inter-species transfer} \quad & -\sum_{\beta\neq e}\frac{2m_em_\beta}{\left(m_e+m_\beta\right)^2}\nu^{\mathrm{MT}}_{e\beta}\left(T_e-T_\beta\right) \qquad (131)
\end{aligned}$$

where

$$\kappa^e=\kappa^e_\parallel\Upsilon_{e\heartsuit}=\frac{5}{2}\frac{n_eT_e}{m\nu_{e\heartsuit}}\Upsilon_{e\heartsuit} \qquad (132)$$

A rearrangement of the thermoelectric terms from heat flux divergence and from resistive "heating" was given at (124).

The contribution of ionisation cooling and recombination heating was spelled out :

$$\frac{\partial T_n}{\partial t}=\frac{1}{n_n}\frac{dn_n}{dt}\bigg|_{recombining}\left(T_e+T_{ion}-T_n+\frac{m_{ion}}{3}\left(v_{ion}-v_n\right)^2+\frac{2}{3}\left[13.6\mathrm{eV}\right]\right) \qquad (133)$$

$$\frac{\partial T_{ion}}{\partial t}=\frac{1}{n_{ion}}\frac{dn_{ion}}{dt}\bigg|_{ionising}\left(\frac{T_n}{2}-T_{ion}+\frac{m_{ion}}{3}\left(v_{ion}-v_n\right)^2\right) \qquad (134)$$

$$\begin{aligned}
\frac{\partial T_e}{\partial t} &= \frac{1}{n_e}\frac{dn_e}{dt}\bigg|_{ionising}\left(\frac{T_n}{2}-T_e-\frac{2}{3}\left[13.6\mathrm{eV}\right]\right)\\
&\quad+\frac{1}{n_e}\left(\frac{dn_e}{dt}\bigg|_{ionising}+\frac{dn_n}{dt}\bigg|_{recombining}\right)\frac{m_e}{3}\left(v_e-v_n\right)^2
\end{aligned}$$

or given the ionising amount in a time interval,

$$n^{ion}_{k+1}T^{ion}_{k+1}=\left(n^{ion}_k-n_{recomb}\right)T^{ion}_k+\frac{1}{2}n_{ionise}T_n+\frac{m_{ion}n_{ionise}}{3}\left(v_{ion,k}-v_{n,k}\right)^2 \qquad (135)$$

$$n_{n,k+1}T_{n,k+1}=\left(n^n_k-n_{ionise}\right)T^n_k+n_{recomb}\left(T^e_k+T^{ion}_k\right)+n_{recomb}\left(\frac{m_{ion}}{3}\left(v_{n,k}-v_{i,k}\right)^2+\frac{2}{3}\left[13.6\mathrm{eV}\right]\right) \qquad (136)$$

$$\begin{aligned}
n^e_{k+1}T^e_{k+1} &= \left(n^e_k-n_{recomb}\right)T^e_k+\frac{1}{2}n_{ionise}T_n+m_e\left(n_{ionise}+n_{recomb}\right)\left(v_{n,k}-v_{e,k}\right)^2\\
&\quad-n_{ionise}\frac{2}{3}\left[13.6\mathrm{eV}\right] \qquad (137)
\end{aligned}$$

Neutral temperature evolution

In fluid frame,

$$\begin{aligned}\frac{\partial T_n}{\partial t} &= -\frac{2}{3}T_n\nabla\cdot v_n + \frac{2}{3n_n}\nabla\cdot(\kappa^n\nabla T_n) - \frac{2}{3n_n}\Pi:\nabla v_n + \frac{2}{3}\frac{m_i^2 m_n}{(m_i+m_n)^2}\nu_{ni}(v_n-v_i)^2 \\ &\quad + [\text{effect of new neutrals at } T_e+T_i] - \sum_{\beta\neq n}\frac{2m_n m_\beta}{(m_n+m_\beta)^2}\nu_{n\beta}^{\mathrm{MT}}(T_n-T_\beta) \qquad (138)\end{aligned}$$

where we use a scalar thermal conductivity

$$\kappa_{[\mathrm{GSKB}]}^{\mathrm{neutral}} = 10\frac{n_n T_n}{m_n \nu_{n\heartsuit}}$$

or per [Vallance1] I infer

$$\kappa^n = \frac{1}{2}\frac{n_n T_n}{m_n(\nu_{nn}^\eta + \nu_{in}^\eta + \nu_{en}^\eta)}$$

So that requires some sensitivity test.

See also 1.1.2-1.1.3 concerning field potentials.

2.1 Tweaks to the model in the fully ionised case based on 13M upwards

Material in Notes3.lyx

3 Tensor and ODE Ohm's Laws from fluid models

3.1 Introduction

First let's say what seems the most logical to do in order to create an Ohm's Law from a fluid dynamic model. Note:

- The advantage of the kinetic-derived conductivities is that they have been tested in various ways and validated (particularly 21-moment). Or so one assumes ?
- An advantage of the fluid-derived tensor is that if we then do not use it for when electrons and ions are out of mutual velocity equilibrium, then using a backward method for the fluid dynamic model will converge to the same result as ν increases.
- Including neutrals is fundamental, indispensible. But it's not clear that we have hold of a kinetic-derived conductivity tensor for the 3-fluid case.

So we have some motivation to see what is the best we can do in the accessible approach that starts from momentum evolution equations. In the lab frame

$$\frac{\partial v_e}{\partial t} = -\left(\nabla v_e\right) v_e - \frac{\nabla\left(n_e T_e\right)}{n_e m} - \frac{\nabla\cdot\Pi_e}{n_e m} - \frac{q}{m}\left(E + \frac{v_e\times B}{c}\right) + \left.\frac{\partial}{\partial t}\right|_{collisions} v_e \tag{139}$$

$$\frac{\partial v_i}{\partial t} = -\left(\nabla v_i\right) v_i - \frac{\nabla\left(n T_i\right)}{n m_i} - \frac{\nabla\cdot\Pi_i}{n m_i} + \frac{q}{m_i}\left(E + \frac{v_i\times B}{c}\right) + \left.\frac{\partial}{\partial t}\right|_{collisions} v_i \tag{140}$$

we can take an equilibrium of say, $v_e - v_i$, but say it was an equilibrium of v_e. Then pulling to the left hand side the terms that are like a matrix times v_e:

$$\left(-\left(\frac{q}{m}\frac{B}{c}\times\right) - \left.\frac{\partial}{\partial t}\right|_{collisions}\right) v_e = -\left(\nabla v_e\right) v_e - \frac{\nabla\left(n_e T_e\right)}{n_e m} - \frac{\nabla\cdot\Pi_e}{n_e m} - \frac{q}{m}E \tag{141}$$

then we simply invert a matrix

$$v_e = \left(\left(\frac{q}{m}\frac{B}{c}\times\right) + \left.\frac{\partial}{\partial t}\right|_{collisions}\right)^{-1}\left(\left(\nabla v_e\right) v_e + \frac{q}{m}E + \frac{\nabla\left(n_e T_e\right)}{n_e m} + \frac{\nabla\cdot\Pi_e}{n_e m}\right) \tag{142}$$

If the matrix to invert looks complicated then it can be done with a symbolic package, but the only benefit of that is for human examination in any case. In practice it will be more efficient just to compute the resistivity tensor and invert it at runtime.

3.1.1 Simplest example: the Krook model and the Alfven conductivity tensor

First let us state what happens if we start from the simplest model of inter-species momentum transfer, the Krook model, that is isotropic and velocity-independent. Per [GZS1977] :

$$\frac{\partial v_e}{\partial t} = -\left(\nabla v_e\right) v_e - \frac{\nabla\left(n_e T_e\right)}{n_e m} - \frac{\nabla\cdot\Pi_e}{n_e m} - \frac{q}{m}\left(E + \frac{v_e\times B}{c}\right) - \sum_\beta \frac{m_\beta}{m + m_\beta}\nu_{e\beta,\mathrm{MT}}\left(v_e - v_\beta\right) \tag{143}$$

$$\frac{\partial v_{\mathrm{ion}}}{\partial t} = -\left(\nabla v_i\right) v_i - \frac{\nabla\left(n T_i\right)}{n m_i} - \frac{\nabla\cdot\Pi_i}{n m_i} + \frac{q}{m_i}\left(E + \frac{v_i\times B}{c}\right) - \sum_\beta \frac{m_\beta}{m_i + m_\beta}\nu_{i\beta,\mathrm{MT}}\left(v_i - v_\beta\right) \tag{144}$$

Now drop the viscous terms and advection terms. Not justified in my view, though [GZS1977] try to justify by saying (in various places) that the advection term is like v^2 and that the off-diagonal stress tensor elements are smaller than the on-diagonal elements.

Rearranging some terms to replace v_e with $v_e - v_i$:

$$\begin{aligned}\frac{\partial (v_e - v_i)}{\partial t} &= -\frac{\nabla (n_e T_e)}{n_e m_e} + \frac{\nabla (n_i T_i)}{n_i m_i} - \left(\frac{q}{m_e} + \frac{q}{m_i}\right) E + \omega_{ce} \times (v_e - v_i) + (\omega_{ce} + \omega_{ci}) \times v_i \\ &+ \left(\frac{m_n \nu_{in}^{\mathrm{MT}}}{m_i + m_n} - \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n}\right)(v_i - v_n) - \left(\frac{m_e \nu_{ie}^{\mathrm{MT}} + m_i \nu_{ei}^{\mathrm{MT}}}{m_i + m_e} + \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n}\right)(v_e - v_i)\end{aligned}$$

Therefore in equilibrium of the relative electron velocity, changing the signs on both sides,

$$\begin{aligned}&\left(\frac{m_e \nu_{ie}^{\mathrm{MT}} + m_i \nu_{ei}^{\mathrm{MT}}}{m_i + m_e} + \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n} - \omega_{ce}\times\right)(v_i - v_e) \\ = \;& \frac{q (m_e + m_i)}{m_e m_i} E + \frac{\nabla (n_e T_e)}{n_e m_e} - \frac{\nabla (n_i T_i)}{n_i m_i} \\ & - (\omega_{ce} + \omega_{ci}) \times v_i + \left(\frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n} - \frac{m_n \nu_{in}^{\mathrm{MT}}}{m_i + m_n}\right)(v_i - v_n) \qquad (145)\end{aligned}$$

Note that $\nu_{ie} = \nu_{ei} \frac{n_e}{n_i}$.
Therefore let

$$\nu^{e-i} = -\frac{m_e \nu_{ie}^{\mathrm{MT}} + m_i \nu_{ei}^{\mathrm{MT}}}{m_i + m_e} - \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n} = -\frac{(m_e n_e / n_i + m_i)}{m_i + m_e} \nu_{ei}^{\mathrm{MT}} - \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n}$$

(Obviously most of the time, $\nu^{e-i} \approx -\nu_{ei}^{\mathrm{MT}} - \nu_{en}^{\mathrm{MT}}$.) Then:

$$\begin{aligned}q n (v_i - v_e) &= \left(\frac{m_e + m_i}{m_e m_i}\right) q^2 n \left(\nu^{e-i} - \omega_{ce}\times\right)^{-1} \qquad (146) \\ & \left(E + \frac{\frac{\nabla (n_e T_e)}{n_e m_e} - \frac{\nabla (n_i T_i)}{n_i m_i} - (\omega_{ce} + \omega_{ci}) \times v_i + \left(\frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n} - \frac{m_n \nu_{in}^{\mathrm{MT}}}{m_i + m_n}\right)(v_i - v_n)}{\left(\frac{q (m_e + m_i)}{m_e m_i}\right)}\right)\end{aligned}$$

Now we could use the fact that

$$\begin{aligned}(\nu - \omega\times)^{-1} &= \qquad (147) \\ \begin{pmatrix} \nu & \omega_z & -\omega_y \\ -\omega_z & \nu & \omega_x \\ \omega_y & -\omega_x & \nu \end{pmatrix}^{-1} &= \frac{1}{\nu (\nu^2 + \omega \cdot \omega)} \begin{pmatrix} \nu^2 + \omega_x^2 & \omega_x \omega_y - \nu \omega_z & \omega_x \omega_z + \nu \omega_y \\ \nu \omega_z + \omega_x \omega_y & \nu^2 + \omega_y^2 & \omega_y \omega_z - \nu \omega_x \\ \omega_x \omega_z - \nu \omega_y & \omega_y \omega_z + \nu \omega_x & \nu^2 + \omega_z^2 \end{pmatrix}\end{aligned}$$

Compare this to the tensor referred to in [Alfven1950], referencing older works such as Engel and Steenbeck 1934. From (p.47 and p.49), using some standard physics notation such as ω for $\|\omega\|$, and without worrying too much what ν_e exactly Alfven had in mind,

$$\sigma_{\parallel} = \frac{q^2 n_e}{m_e \nu_e} \qquad (148)$$

$$\sigma_{\perp} = \frac{q^2 n_e}{m_e \nu_e (1 + \omega_e^2 / \nu_e^2)} = \frac{q^2 n_e}{m_e \nu_e} \frac{\nu_e^2}{\nu_e^2 + \omega^2} \qquad (149)$$

$$\sigma_{\wedge} = \frac{\omega_e}{\nu_e} \sigma_{\perp} = \frac{q^2 n_e}{m_e \nu_e} \frac{\omega \nu_e}{\nu_e^2 + \omega^2} \qquad (150)$$

(Alfven does not state which sign his Hall direction is, but I think we can infer that nothing is untoward.)

So

$$\left(\sigma_{\parallel} - \sigma_{\perp}\right) = \frac{\omega^2}{\nu_e^2 + \omega^2} \sigma_{\parallel}$$

Therefore per (28) and (147), the corresponding Cartesian tensor is (for $\nu = \nu_e$, $\omega = \omega_e$)

$$\begin{aligned}\sigma_{Alfven} &= \frac{\frac{q^2 n_e}{m_e \nu}}{\nu^2+\omega^2}\begin{pmatrix} \nu^2+\omega^2 b_x^2 & -\omega\nu b_z + \omega^2 b_x b_y & \omega\nu b_y + \omega^2 b_x b_z \\ \omega\nu b_z + \omega^2 b_x b_y & \nu^2+\omega^2 b_y^2 & -\omega\nu b_x + \omega^2 b_y b_z \\ -\omega\nu b_y + \omega^2 b_x b_z & \omega\nu b_x + \omega^2 b_y b_z & \nu^2+\omega^2 b_z b_z \end{pmatrix} \\ &= \frac{q^2 n}{m}\left(\nu - \omega\times\right)^{-1} := \frac{q^2 n_e}{m_e \nu}\Upsilon^{\nu}_{Alfven} = \sigma_{\parallel}\Upsilon^{\nu}_{Alfven} \end{aligned} \tag{151}$$

It makes a kind of sense that this sort of tensor is used in a space plasma context because if it is true that on large scales, temperature gradients and velocity gradients are not important [Somov2013], that gets rid of the thermal force as well as the viscous force. This tensor still features in pedagogical material about astrophysics [solarsystem.de, Somov2013].

So in summary, we have shown that by taking the Krook model and inverting a matrix, you get the 3-fluid version of what we shall call the **Alfven conductivity** tensor.

3.2 Using the model of collisions given above

3.2.1 Introductory version including magnetic coordinates

We have supposed that only the e-i collisions are treated as velocity-dependent. Then

$$\frac{\partial v_e}{\partial t} = -\left(\nabla v_e\right)v_e - \frac{\nabla\left(n_e T_e\right)}{n_e m} - \frac{\nabla\cdot\Pi_e}{n_e m} - \frac{q}{m}\left(E + \frac{v_e\times B}{c}\right) - \frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}\left(v_e - v_n\right) + \frac{R_{ei}}{m_e} \tag{152}$$

$$\frac{\partial v_{\mathrm{ion}}}{\partial t} = -\left(\nabla v_i\right)v_i - \frac{\nabla\left(nT_i\right)}{n m_i} - \frac{\nabla\cdot\Pi_i}{n m_i} + \frac{q}{m_i}\left(E + \frac{v_i\times B}{c}\right) - \frac{m_n}{m_i+m_n}\nu_{in}^{\mathrm{MT}}\left(v_i - v_n\right) - \frac{n_e}{n_i}\frac{R_{ei}}{m_i} \tag{153}$$

where $R_{ie} = -\frac{n_e}{n_i}R_{ei}$ follows from conservation of momentum $m_i n_i v_i + m_e n_e v_e$. We said

$$\frac{R_{ei}}{m_e} = -\frac{3}{2}\frac{\bar{\nu_{ei}}}{m\left(\nu_{en}+1.87\bar{\nu_{ei}}\right)}\Upsilon^{\nu_{en}+1.87\bar{\nu_{ei}}}_{Alfven}\nabla T + \left(\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{en}+1.87\bar{\nu_{ei}}}\Upsilon^{\nu_{en}+1.87\bar{\nu_{ei}}}_{Alfven} - 1_{3\times 3}\right)\bar{\nu_{ei}}\left(v_e - v_i\right)$$

Dropping advection and viscosity, as before, for convenience:

$$\begin{aligned} 0 &= \frac{\partial\left(v_e - v_i\right)}{\partial t} = \\ &-\frac{\nabla\left(n_e T_e\right)}{n_e m_e} + \frac{\nabla\left(n_i T_i\right)}{n_i m_i} - \left(\frac{q}{m_e}+\frac{q}{m_i}\right)E + \omega_{ce}\times\left(v_e - v_i\right) + \left(\omega_{ce}+\omega_{ci}\right)\times v_i \qquad (154)\\ &+\left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n} - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(v_i - v_n\right) - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\left(v_e - v_i\right) + \left(1+\frac{n_e m_e}{n_i m_i}\right)\frac{R_{ei}}{m_e} \\ &= -\frac{\nabla\left(n_e T_e\right)}{n_e m_e} + \frac{\nabla\left(n_i T_i\right)}{n_i m_i} - \left(\frac{q}{m_e}+\frac{q}{m_i}\right)E + \left(\omega_{ce}+\omega_{ci}\right)\times v_i \qquad (155)\\ &+\left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n} - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(v_i - v_n\right) + \left(1+\frac{n_e m_e}{n_i m_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\left(\nu_{en}+1.87\bar{\nu_{ei}}\right)}\Upsilon^{\nu_{en}+1.87\bar{\nu_{ei}}}_{Alfven}\nabla T\right) \\ &-\left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n} + \left(1+\frac{n_e m_e}{n_i m_i}\right)\left(1_{3\times 3} - \frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{en}+1.87\bar{\nu_{ei}}}\Upsilon^{\nu_{en}+1.87\bar{\nu_{ei}}}_{Alfven}\right)\bar{\nu_{ei}} - \omega_{ce}\times\right)\left(v_e - v_i\right) \end{aligned}$$

I shall take only the $(v_e - v_i)$ term from R_{ei} over to the left hand side. Then we shall have

$$\begin{aligned} v_e - v_i &= \left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n} + \left(1+\frac{n_e m_e}{n_i m_i}\right)\left(1_{3\times 3} - \frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{en}+1.87\bar{\nu_{ei}}}\Upsilon^{\nu_{en}+1.87\bar{\nu_{ei}}}_{Alfven}\right)\bar{\nu_{ei}} - \omega_{ce}\times\right)^{-1} \qquad (156)\\ &\left(-\frac{\nabla\left(n_e T_e\right)}{n_e m_e} + \frac{\nabla\left(n_i T_i\right)}{n_i m_i} - \frac{q\left(m_e+m_i\right)}{m_e m_i}E + \left(\omega_{ce}+\omega_{ci}\right)\times v_i\right. \qquad (157)\\ &\left.+\left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n} - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(v_i - v_n\right) - \frac{3}{2}\frac{\bar{\nu_{ei}}\left(1+\frac{n_e m_e}{n_i m_i}\right)}{m_e\left(\nu_{en}+1.87\bar{\nu_{ei}}\right)}\Upsilon^{\nu_{en}+1.87\bar{\nu_{ei}}}_{Alfven}\nabla T\right) \end{aligned}$$

$$qn\left(v_i - v_e\right) =$$

$$q^2 n \frac{m_e + m_i}{m_e m_i}\left(\frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n} + \left(1 + \frac{n_e m_e}{n_i m_i}\right)\left(1_{3\times 3} - \frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{en} + 1.87\bar{\nu_{ei}}}\Upsilon_{Alfven}^{\nu_{en}+1.87\bar{\nu_{ei}}}\right)\bar{\nu_{ei}} - \omega_{ce}\times\right)^{-1} \tag{158}$$

$$\left(E + \frac{\frac{\nabla(n_e T_e)}{n_e m_e} - \frac{\nabla(n_i T_i)}{n_i m_i} - \left(\omega_{ce} + \omega_{ci}\right)\times v_i - \left(\frac{m_n \nu_{in}^{\mathrm{MT}}}{m_i + m_n} - \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n}\right)\left(v_i - v_n\right)}{\frac{q(m_e+m_i)}{m_e m_i}}\right. \tag{159}$$

$$\left. + \frac{m_e m_i}{q\left(m_e + m_i\right)}\frac{3}{2}\frac{\bar{\nu_{ei}}\left(1 + \frac{n_e m_e}{n_i m_i}\right)}{m_e\left(\nu_{en} + 1.87\bar{\nu_{ei}}\right)}\Upsilon_{Alfven}^{\nu_{en}+1.87\bar{\nu_{ei}}}\nabla T\right) \tag{160}$$

So then, for a 3-component plasma with collisional momentum transfer that is velocity-dependent for e-i but velocity-independent for e-n, i-n, we have $\sigma^{\circledast} =$

$$q^2 n \frac{m_e + m_i}{m_e m_i}\left(\frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n} + \left(1 + \frac{n_e m_e}{n_i m_i}\right)\left(1_{3\times 3} - \frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{en}^{\mathrm{MT}} + 1.87\bar{\nu_{ei}}}\Upsilon_{Alfven}^{\nu_{en}+1.87\bar{\nu_{ei}}}\right)\bar{\nu_{ei}} - \omega_{ce}\times\right)^{-1} \tag{161}$$

$$\approx \frac{q^2 n}{m_e}\left(\nu_{en}^{\mathrm{MT}} + \left(1_{3\times 3} - \frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{en}^{\mathrm{MT}} + 1.87\bar{\nu_{ei}}}\Upsilon_{Alfven}^{\nu_{en}+1.87\bar{\nu_{ei}}}\right)\bar{\nu_{ei}} - \omega_{ce}\times\right)^{-1} = \sigma_{simplified}^{\circledast} \tag{162}$$

and its simplified operand is (since generally $\nabla_j\left(n_i T_i\right) < \nabla_j\left(n_e T_e\right)$ and $n_i \sim n_e$ so that $\frac{\nabla(n_e T_e)}{n_e m_e} > \frac{m_i}{m_e}\frac{\nabla(n_i T_i)}{n_i m_i}$)

$$\begin{aligned}\sigma^{-1} J &= E - \frac{B\times v_i}{c} + \frac{m_e}{q}\left(\frac{\nabla\left(n_e T_e\right)}{n_e m_e} - \left(\frac{1}{2}\nu_{in}^{\mathrm{MT}} - \nu_{en}^{\mathrm{MT}}\right)\left(v_i - v_n\right)\right)\\ &\quad + \frac{3}{2qm_e}\frac{\bar{\nu_{ei}}}{\left(\nu_{en} + 1.87\bar{\nu_{ei}}\right)}\Upsilon_{Alfven}^{\nu_{en}+1.87\bar{\nu_{ei}}}\nabla T\end{aligned} \tag{163}$$

It might not at this point be obvious that there is a nice form for the elements of σ symbolically. Far less that we can put it into magnetic coordinates. But carry on:

Writing T for a generalised time, let

$$T^{-1} = \nu_{en} + \left(1_{3\times 3} - \frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{en}^{\mathrm{MT}} + 1.87\bar{\nu_{ei}}}\Upsilon_{Alfven}^{\nu_{en}+1.87\bar{\nu_{ei}}}\right)\bar{\nu_{ei}} - \omega_{ce}\times$$

Again this is just for fun, no sane person would program it by computing the conductivity instead of resistivity. Let

$$\nu_1 = \nu_{en} + \bar{\nu_{ei}} \tag{164}$$

$$\nu_{e\heartsuit} = \nu_{en} + 1.87\bar{\nu_{ei}} \tag{165}$$

$$a_2 = -\frac{9}{10}\frac{\left(\bar{\nu_{ei}}\right)^2}{\nu_{e\heartsuit}\left(\nu_{e\heartsuit}^2 + \omega\cdot\omega\right)} \tag{166}$$

so that

$$\begin{aligned}T^{-1} &= \nu_1 1_{3\times 3} - \left(\omega_e\times\right) + a_2\begin{pmatrix}\nu_{e\heartsuit}^2 + \omega_x^2 & \omega_x\omega_y - \nu\omega_z & \omega_x\omega_z + \nu\omega_y\\ \nu\omega_z + \omega_x\omega_y & \nu^2 + \omega_y^2 & \omega_y\omega_z - \nu\omega_x\\ \omega_x\omega_z - \nu\omega_y & \omega_y\omega_z + \nu\omega_x & \nu^2 + \omega_z^2\end{pmatrix}\\ &= \left(\nu_1 + a_2\nu_{e\heartsuit}^2\right)1_{3\times 3} - \left(\left(1 - a_2\nu_{e\heartsuit}\right)\omega_e\times\right) + a_2\begin{pmatrix}\omega_x^2 & \omega_x\omega_y & \omega_x\omega_z\\ \omega_x\omega_y & \omega_y^2 & \omega_y\omega_z\\ \omega_x\omega_z & \omega_y\omega_z & \omega_z^2\end{pmatrix}\end{aligned} \tag{167}$$

We could work out the adjugate matrix but just try Wolfram Alpha instead: putting $a = a_2$ and

$$b = (1 - a_2\nu_{e\heartsuit}) = 1 + \frac{9}{10}\frac{\left(\bar{\nu_{ei}}\right)^2}{\left(\nu_{e\heartsuit}^2 + \omega\cdot\omega\right)} \in (1, 1.2574)$$

$$c = \left(\nu_1 + a_2\nu_{e\heartsuit}^2\right) = \nu_{en} + \bar{\nu_{ei}} - \frac{9}{10}\frac{\left(\bar{\nu_{ei}}\right)^2\nu_{e\heartsuit}}{\left(\nu_{e\heartsuit}^2 + \omega\cdot\omega\right)} = \nu_{en} + \bar{\nu_{ei}}\left(\frac{\nu_{e\heartsuit}\left(\nu_{e\heartsuit} - \frac{9}{10}\bar{\nu_{ei}}\right) + \omega\cdot\omega}{\nu_{e\heartsuit}^2 + \omega\cdot\omega}\right)$$

$$T^{-1} = \begin{pmatrix} a\omega_x^2 + c & a\omega_x\omega_y + b\omega_z & a\omega_x\omega_z - b\omega_y \\ a\omega_x\omega_y - b\omega_z & a\omega_y^2 + c & a\omega_y\omega_z + b\omega_x \\ a\omega_x\omega_z + b\omega_y & a\omega_y\omega_z - b\omega_x & a\omega_z^2 + c \end{pmatrix}$$

Solution:

$$T = \frac{1}{\left(a\omega^2 + c\right)\left(b^2\omega^2 + c^2\right)}\times \tag{168}$$

$$\begin{pmatrix} c^2 + ac\omega^2 + \left(b^2 - ac\right)\omega_x^2 & -\left(ab\omega^2 + bc\right)\omega_z + \left(b^2 - ac\right)\omega_x\omega_y & \left(ab\omega^2 + bc\right)\omega_y + \left(b^2 - ac\right)\omega_x\omega_z \\ \left(ab\omega^2 + bc\right)\omega_z + \left(b^2 - ac\right)\omega_x\omega_y & c^2 + ac\omega^2 + \left(b^2 - ac\right)\omega_y^2 & -\left(ab\omega^2 + bc\right)\omega_x + \left(b^2 - ac\right)\omega_y\omega_z \\ -\left(ab\omega^2 + bc\right)\omega_y + \left(b^2 - ac\right)\omega_x\omega_z & \left(ab\omega^2 + bc\right)\omega_x + \left(b^2 - ac\right)\omega_y\omega_z & c^2 + ac\omega^2 + \left(b^2 - ac\right)\omega_z^2 \end{pmatrix}$$

Very surprisingly, it's not even hard to recognise what it is for magnetic coordinates.
Firstly note that we have the unmagnetised collision frequency

$$\begin{aligned} a\omega^2 + c &= -\frac{9}{10}\frac{\left(\bar{\nu_{ei}}\right)^2}{\nu_{e\heartsuit}\left(\nu_{e\heartsuit}^2 + \omega\cdot\omega\right)}\omega^2 + \nu_1 + a_2\nu_{e\heartsuit}^2 = \nu_{en} + \bar{\nu_{ei}}\left(1 - \frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\right) \\ &= \nu_{en} + \bar{\nu_{ei}}\left(\frac{0.97\bar{\nu_{ei}} + \nu_{en}}{\nu_{e\heartsuit}}\right) = \nu^{[B=0]} \end{aligned} \tag{169}$$

and therefore set

$$\nu^{aug} = c = \nu^{[B=0]} - a\omega^2 = \nu^{[B=0]} + \frac{9}{10}\frac{\bar{\nu_{ei}}^2\omega^2}{\nu_{e\heartsuit}\left(\nu_{e\heartsuit}^2 + \omega^2\right)} = \nu^{[B=0]} + \nu^B\frac{\omega^2}{\nu_{e\heartsuit}^2 + \omega^2} \tag{170}$$

So factor out $\nu^0 = \nu^{[B=0]}$:

$$T = \frac{1}{\nu^0\left(b^2\omega^2 + c^2\right)}\begin{pmatrix} c\nu^0 + \left(b^2 - ac\right)\omega_x^2 & -b\nu^0\omega_z + \left(b^2 - ac\right)\omega_x\omega_y & b\nu^0\omega_y + \left(b^2 - ac\right)\omega_x\omega_z \\ b\nu^0\omega_z + \left(b^2 - ac\right)\omega_x\omega_y & c\nu^0 + \left(b^2 - ac\right)\omega_y^2 & -b\nu^0\omega_x + \left(b^2 - ac\right)\omega_y\omega_z \\ -b\nu^0\omega_y + \left(b^2 - ac\right)\omega_x\omega_z & b\nu^0\omega_x + \left(b^2 - ac\right)\omega_y\omega_z & c\nu^0 + \left(b^2 - ac\right)\omega_z^2 \end{pmatrix} \tag{171}$$

Bear in mind we want to come out with something that looks like $\frac{1}{\nu}$ so at the moment, the ν^0 in the denominator is filling that role. Temporarily let

$$\sigma_0^{\circledast} = \frac{q^2 n}{m_e\nu^0}$$

Then

$$\sigma^{\circledast} = \sigma_0^{\circledast}\Upsilon^{\circledast} = \sigma_0^{\circledast}\left(\nu^0 T\right)$$

Now let's remember that $\sigma_\wedge^{\circledast}, \sigma_\parallel^{\circledast}$ are coefficients on $\hat{B}$ products not ω products. Unfortunate notation of b here. Let's call it β.

$$\sigma_\parallel^{\circledast} = \sigma_0^{\circledast}\left(\frac{\left(\beta^2 - ac\right)\omega^2 + c\nu^0}{\beta^2\omega^2 + c^2}\right) = \sigma_0^{\circledast} = \frac{q^2 n}{m_e\nu^0} \tag{172}$$

Conclusion: σ in magnetic coordinates

We allowed velocity-dependent e-i momentum transfer but velocity-independent e-n and i-n momentum transfer.
Then where

$$v_{e\heartsuit} = 1.87\bar{\nu_{ei}} + \nu_{en} \tag{173}$$

$$\nu^0 = \nu^{[B=0]} = \nu_{en} + \bar{\nu_{ei}}\left(\frac{v_{e\heartsuit} - \frac{9}{10}\bar{\nu_{ei}}}{v_{e\heartsuit}}\right) = \nu_{en} + \bar{\nu_{ei}}\left(\frac{0.97\bar{\nu_{ei}} + \nu_{en}}{1.87\bar{\nu_{ei}} + \nu_{en}}\right) \tag{174}$$

$$\nu^B = \frac{9}{10}\frac{\bar{\nu_{ei}}^2}{v_{e\heartsuit}} = \frac{9}{10}\frac{\bar{\nu_{ei}}^2}{1.87\bar{\nu_{ei}} + \nu_{en}} \tag{175}$$

$$\nu_{aug} = \nu^0 + \nu^B\frac{\omega^2}{\nu_{e\heartsuit}^2 + \omega^2} \tag{176}$$

$$\beta = 1 + \frac{9}{10}\frac{\bar{\nu_{ei}}^2}{\nu_{e\heartsuit}^2 + \omega^2} \tag{177}$$

Note that β is near 1.25 when neutral collisions are not important but $\omega < \nu$; near 1 in other cases.

$$\sigma_{\parallel}^{\circledast} = \frac{q^2 n}{m_e \nu^0} \tag{178}$$

$$\sigma_{\perp}^{\circledast} = \sigma_{\parallel}^{\circledast}\frac{\nu_{aug}\nu^0}{\nu_{aug}^2 + \beta^2\omega^2} = \sigma_{\parallel}^{\circledast}\frac{\left(\nu^0 + \nu^B\frac{\omega^2}{\nu_{e\heartsuit}^2+\omega^2}\right)\nu^0}{\left(\beta^2\omega^2 + \left(\nu^0 + \nu^B\frac{\omega^2}{\nu_{e\heartsuit}^2+\omega^2}\right)^2\right)} \tag{179}$$

$$\sigma_{\wedge}^{\circledast} = \sigma_{\parallel}^{\circledast}\frac{\beta\nu^0\omega}{\nu_{aug}^2 + \beta^2\omega^2} = \sigma_{\parallel}^{\circledast}\frac{\beta\nu^0}{\left(\beta^2\omega^2 + \left(\nu^0 + \nu^B\frac{\omega^2}{\nu_{e\heartsuit}^2+\omega^2}\right)^2\right)} \tag{180}$$

3.3 ODE Ohm's Law

We create the Ohm's Law by setting $\frac{\partial}{\partial t}(v_e - v_i) = 0$ and neglecting the advection terms and the ion viscous term. This is small compared to the electron viscous term. Thus

$$\begin{aligned} 0 \;=\; & -\frac{\nabla(n_e T_e)}{n_e m_e} + \frac{\nabla(n_i T_i)}{n_i m_i} - \frac{\nabla\cdot\Pi_e}{n_e m}\left(v_e, v_e^{neighbours}\right) - \left(\frac{q}{m_e} + \frac{q}{m_i}\right)E + (\omega_{ce} + \omega_{ci}) \times v_i \\ & + \left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i + m_n} - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e + m_n}\right)(v_i - v_n) + \left(1 + \frac{n_e m_e}{n_i m_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T\right) \\ & - \left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e + m_n} + \left(1 + \frac{n_e m_e}{n_i m_i}\right)\left(1_{3\times3} - \frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}} - \omega_{ce}\times\right)(v_e - v_i) \end{aligned} \tag{181}$$

Now we expand v_i, v_n so that we can collect all terms in v_e, E. Note that $\omega_{ce} \times v_i$ cancels between the bottom line and the top line;

$$\begin{aligned}
0 &= -\frac{\nabla(n_e T_e)}{n_e m_e} + \frac{\nabla(n_i T_i)}{n_i m_i} - \frac{\nabla\cdot\Pi_e}{n_e m}\left(v_e, v_e^{neighbours}\right) - \left(\frac{q}{m_e} + \frac{q}{m_i}\right) E \\
&+ (\omega_{ce} + \omega_{ci}) \times \left(v_i^0 + \sigma_i E + \beta_i^e v_e\right) \\
&+ \left(\frac{m_n \nu_{in}^{\mathrm{MT}}}{m_i + m_n} - \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n}\right)\left(v_i^0 + \sigma_i^E E + \beta_i^e v_e - v_n^0 - \sigma_n^E - \beta_n^E v_e\right) \\
&+ \left(1 + \frac{n_e m_e}{n_i m_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e \nu_{e\heartsuit}} \Upsilon_{e\heartsuit} \nabla T\right) \\
&- \left(\frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n} + \left(1 + \frac{n_e m_e}{n_i m_i}\right)\left(1_{3\times 3} - \frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}} - \omega_{ce}\times\right)\left(v_e - v_i^0 - \sigma_i^E E - \beta_i^e v_e\right) \quad (182)
\end{aligned}$$

Let

$$\nu_{3\times 3}^{\mathrm{FANCY}} = \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n} + \left(1 + \frac{n_e m_e}{n_i m_i}\right)\left(1_{3\times 3} - \frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}$$

$$\begin{aligned}
0 &= -\frac{\nabla\cdot\Pi_e}{n_e m}\left(v_e, v_e^{neighbours}\right) \\
&- \frac{\nabla(n_e T_e)}{n_e m_e} + \frac{\nabla(n_i T_i)}{n_i m_i} + (\omega_{ce} + \omega_{ci}) \times \left(v_i^0\right) + \left(\frac{m_n \nu_{in}^{\mathrm{MT}}}{m_i + m_n} - \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n}\right)\left(v_i^0 - v_n^0\right) \\
&+ \left(1 + \frac{n_e m_e}{n_i m_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e \nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T\right) - \left(\nu_{3\times 3}^{\mathrm{FANCY}} - \omega_{ce}\times\right)\left(-v_i^0\right) \\
&+ \left(-\left(\frac{q}{m_e} + \frac{q}{m_i}\right) 1_{3\times 3} + (\omega_{ce} + \omega_{ci}) \times \sigma_i^E + \left(\frac{m_n \nu_{in}^{\mathrm{MT}}}{m_i + m_n} - \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n}\right)\left(\sigma_i^E - \sigma_n^E\right) + \left(\nu_{3\times 3}^{\mathrm{FANCY}} - \omega_{ce}\times\right)\right. \\
&+ \left.\left((\omega_{ce} + \omega_{ci}) \times \beta_i^e + \left(\frac{m_n \nu_{in}^{\mathrm{MT}}}{m_i + m_n} - \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n}\right)\left(\beta_i^e - \beta_n^e\right) - \left(\nu_{3\times 3}^{\mathrm{FANCY}} - \omega_{ce}\times\right)\left(1 - \beta_i^e\right)\right) v_e
\end{aligned}$$

It's fine in this form, but applying a few cancellations anyway:

$$\begin{aligned}
0 &= -\frac{\nabla\cdot\Pi_e}{n_e m}\left(v_e, v_e^{neighbours}\right) - \frac{\nabla(n_e T_e)}{n_e m_e} + \frac{\nabla(n_i T_i)}{n_i m_i} + \omega_{ci} \times v_i^0 + \left(\frac{m_n \nu_{in}^{\mathrm{MT}}}{m_i + m_n} - \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n}\right)\left(v_i^0 - v_n^0\right) \\
&+ \left(1 + \frac{n_e m_e}{n_i m_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e \nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T\right) + \nu_{3\times 3}^{\mathrm{FANCY}} v_i^0 \\
&+ \left(-\left(\frac{q}{m_e} + \frac{q}{m_i}\right) 1_{3\times 3} + \omega_{ci} \times \sigma_i^E + \left(\frac{m_n \nu_{in}^{\mathrm{MT}}}{m_i + m_n} - \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n}\right)\left(\sigma_i^E - \sigma_n^E\right) + \nu_{3\times 3}^{\mathrm{FANCY}} \sigma_i^E\right) E \\
&+ \left((\omega_{ce} + \omega_{ci}) \times \beta_i^e + \left(\frac{m_n \nu_{in}^{\mathrm{MT}}}{m_i + m_n} - \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n}\right)\left(\beta_i^e - \beta_n^e\right) - \left(\nu_{3\times 3}^{\mathrm{FANCY}} - \omega_{ce}\times\right)\left(1 - \beta_i^e\right)\right) v_e \quad (184)
\end{aligned}$$

Now again let

$$\frac{\nabla\cdot\Pi_e}{n_e m} = \left[\frac{\nabla\cdot\Pi_e}{n_e m}; \beta_{self}\right] v_e + \left[\frac{\nabla\cdot\Pi_e}{n_e m}; \beta^{neighs}\right] v_e^{neighbours} \quad (185)$$

Therefore the negative of the terms in v_e amounts to

$$\begin{aligned}
\chi^{-1} &= \nu_{3\times 3}^{\mathrm{FANCY}}\left(1 - \beta_i^e\right) - \left(\omega_{ce}\times\right) - \omega_{ci} \times \beta_i^e \\
&- \left(\frac{m_n \nu_{in}^{\mathrm{MT}}}{m_i + m_n} - \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n}\right)\left(\beta_i^e - \beta_n^e\right) + \left[\frac{\nabla\cdot\Pi_e}{n_e m}; \beta_{self}\right] \quad (186)
\end{aligned}$$

Let

$$\begin{aligned}
\sigma_e &= \chi\left(-\left(\frac{q}{m_e} + \frac{q}{m_i}\right) + \left(\omega_{ci}\times\right)\sigma_i^E + \left(\frac{m_n \nu_{in}^{\mathrm{MT}}}{m_i + m_n} - \frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n}\right)\left(\sigma_i^E - \sigma_n^E\right)\right. \\
&+ \left.\left(\frac{m_n \nu_{en}^{\mathrm{MT}}}{m_e + m_n} + \left(1 + \frac{n_e m_e}{n_i m_i}\right)\left(1_{3\times 3} - \frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}\right)\sigma_i^E\right)
\end{aligned}$$

Therefore applying χ to the Ohm's Law equation, we have

$$\begin{aligned} v_e &= \chi\left(-\frac{\nabla\left(n_eT_e\right)}{n_em_e}+\frac{\nabla\left(n_iT_i\right)}{n_im_i}+\omega_{ci}\times v_i^0+\left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n}-\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(v_i^0-v_n^0\right)\right. \\ &\quad +\left(1+\frac{n_em_e}{n_im_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T\right)+\left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}+\left(1+\frac{n_em_e}{n_im_i}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}\right)v_i^0\Bigg) \\ &\quad -\chi\left[\frac{\nabla\cdot\Pi_e}{n_em};\beta^{neighs}\right]v_e^{neighbours}+\sigma_eE \end{aligned}$$

Therefore let

$$\varepsilon^{Ohm}=\frac{4\pi}{c}qN\left(-v_e+v_e^0-\chi\left[\frac{\nabla\cdot\Pi_e}{n_em};\beta^{neighs}\right]v_e^{neighbours}+\sigma_eE\right)n^{\alpha-1} \tag{187}$$

where

$$\begin{aligned} v_e^0 &= \chi\left(-\frac{\nabla\left(n_eT_e\right)}{n_em_e}+\frac{\nabla\left(n_iT_i\right)}{n_im_i}+\omega_{ci}\times v_i^0+\left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n}-\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(v_i^0-v_n^0\right)\right. \\ &\quad \left.+\left(1+\frac{n_em_e}{n_im_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T\right)+\left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}+\left(1+\frac{n_em_e}{n_im_i}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}\right)v_i^0\right) \end{aligned} \tag{188}$$

We can calculate v_e^0,χ,σ_e before we get into the ODE solver. We also need to know the coefficients $\left[\frac{\nabla\cdot\Pi_e}{n_em};\beta\right]$ that come from the viscous acceleration. It seems likely that we need the tensors β_i^e,σ_i^E and vector v_i^0 to enter the solver.

3.4 Tensor Ohm's Law

The inviscid Ohm's Law can be obtained simply by assuming that the viscous terms are zero in the conclusions above. viz, let

$$\nu_{3\times3}^{\mathrm{FANCY}}=\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}+\left(1+\frac{n_em_e}{n_im_i}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}} \tag{189}$$

$$\begin{aligned} \chi^{-1} &= \nu_{3\times3}^{\mathrm{FANCY}}\left(1-\beta_i^e\right)-\left(\omega_{ce}\times\right)-\omega_{ci}\times\beta_i^e \\ &\quad -\left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n}-\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(\beta_i^e-\beta_n^e\right) \end{aligned} \tag{190}$$

$$\begin{aligned} \sigma_e &= \chi\left(-\left(\frac{q}{m_e}+\frac{q}{m_i}\right)+\left(\omega_{ci}\times\right)\sigma_i^E+\left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n}-\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(\sigma_i^E-\sigma_n^E\right)\right. \\ &\quad \left.+\left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}+\left(1+\frac{n_em_e}{n_im_i}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}\right)\sigma_i^E\right) \end{aligned} \tag{191}$$

Then let

$$v_e=v_e^0+\sigma_eE$$

with no dependence on other electron velocity values.

3.5 Initial setup Ohm's Law

Initially we need to make additional assumptions in order to solve for the initial A, J, E fields. Firstly assume

$$\frac{\partial A_z}{\partial t}/A_z = \frac{\partial I_z}{\partial t}/I_z \tag{192}$$

and therefore the E field with the magnetoinductive contribution is

$$E_z = E_z^{circuit} - \frac{1}{c}\left(\frac{\partial I_z/\partial t}{I_z}\right) A_z \tag{193}$$

Now initially we will write an Ohm's Law assuming

- $B = 0$. This will be inconsistent; we can, if we wish, iterate a few times and hopefully so converge.
- In initial conditions, $n_e = n_i$. We shall also assume $T_e = T_i = T_n$.
- Advective drift, thermal pressure, the thermal force and the viscous momentum flux in z direction are all zero since we assume $\frac{d}{dz}\text{all} = 0$.

Now first let's set $\frac{\partial}{\partial t} v_{neut} = 0$.

This implies

$$\begin{aligned} 0 &= \frac{\partial v_n}{\partial t} = -\frac{m_e}{m_e+m_n}\nu_{ne}^{\mathrm{MT}}\left(v_n - v_e\right) - \frac{m_i}{m_i+m_n}\nu_{ni}^{\mathrm{MT}}\left(v_n - v_i\right) \\ &= -\frac{m_e}{m_e+m_n} n\sigma_{en}^{\mathrm{MT}}\left(\frac{T_e}{m_e}\right)^{1/2}\left(v_n - v_e\right) - \frac{m_i}{m_i+m_n} n\sigma_{in}^{MT}\left(\frac{T_i}{m_i}+\frac{T_n}{m_n}\right)^{1/2}\left(v_n - v_i\right) \end{aligned}$$

We said that we use the same cross-sections given the same ion and electron temperatures, so

$$0 = -\frac{m_e}{m_e+m_n}\left(\frac{T_e}{m_e}\right)^{1/2}\left(v_n - v_e\right) - \frac{m_i}{m_i+m_n}\left(\frac{T_i}{m_i}+\frac{T_n}{m_n}\right)^{1/2}\left(v_n - v_i\right)$$

It is seen that $\nu_{ne} \approx \sqrt{m_e/m_i}\nu_{ni}$ so basically the neutral velocity should be much closer to the ion velocity than to the electron velocity:

$$v_n = \frac{\frac{m_e}{m_e+m_n}\left(\frac{T_e}{m_e}\right)^{1/2} v_e + \frac{m_i}{m_i+m_n}\left(\frac{T_i}{m_i}+\frac{T_n}{m_n}\right)^{1/2} v_i}{\frac{m_e}{m_e+m_n}\left(\frac{T_e}{m_e}\right)^{1/2} + \frac{m_i}{m_i+m_n}\left(\frac{T_i}{m_i}+\frac{T_n}{m_n}\right)^{1/2}}$$

or just from the top line,

$$v_n = \frac{\frac{m_e}{m_e+m_n}\nu_{ne}^{\mathrm{MT}} v_e + \frac{m_i}{m_i+m_n}\nu_{ni}^{\mathrm{MT}} v_i}{\frac{m_e}{m_e+m_n}\nu_{ne}^{\mathrm{MT}} + \frac{m_i}{m_i+m_n}\nu_{ni}^{\mathrm{MT}}} = \frac{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} v_e + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}} v_i}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}$$

Now the acceleration equations for electrons and ions read (bearing in mind that for $B = 0$, $\Upsilon_{e\heartsuit} = 1$, and that $\nu_{ie} = \nu_{ei}$ initially)

$$0 = \frac{\partial v_i}{\partial t}_z = \frac{e}{m_i}E - \frac{m_n}{m_i+m_n}\nu_{in}^{\mathrm{MT}}\left(v_i - \frac{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} v_e + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}} v_i}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}\right) - \frac{m_e}{m_i}\left(1 - \frac{9}{10}\frac{\nu}{1.87\nu_e}\right.$$

$$0 = \frac{\partial v_e}{\partial t}_z = -\frac{e}{m_e}E - \frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}\left(v_e - \frac{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} v_e + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}} v_i}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}\right) - \left(1 - \frac{9}{10}\frac{\nu_e}{1.87\nu_{ei}}\right.$$

Or:

$$0 = \frac{\partial v_i}{\partial t}_z = \frac{e}{m_i}E - \left(\frac{m_e m_n \nu_{ne}^{\mathrm{MT}} \nu_{in}^{\mathrm{MT}}}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}} + \frac{m_e}{m_i}\left(1 - \frac{9}{10}\frac{\nu_{ei}}{1.87\nu_{ei}+\nu_{en}}\right)\nu_{ei}\right)\left(v_i - v_e\right)$$

$$0 = \frac{\partial v_e}{\partial t}_z = -\frac{e}{m_e}E - \left(\frac{m_i m_n \nu_{ni}^{\mathrm{MT}} \nu_{en}^{\mathrm{MT}}}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}} + \left(1 - \frac{9}{10}\frac{\nu_{ei}}{1.87\nu_{ei}+\nu_{en}}\right)\nu_{ei}\right)\left(v_e - v_i\right)$$

However, then the factors on $(v_i - v_e)$ are opposite and in a ratio of m_e/m_i :

$$0 = \frac{\partial v_i}{\partial t}_z = \frac{e}{m_i}E - \frac{m_e}{m_i}\nu_{ei\oplus}\left(v_i - v_e\right)$$

$$0 = \frac{\partial v_e}{\partial t}_z = -\frac{e}{m_e}E - \nu_{ei\oplus}\left(v_e - v_i\right)$$

$$\nu_{ei\oplus} = \frac{m_i m_n \nu_{ni}^{\mathrm{MT}} \nu_{en}^{\mathrm{MT}}}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}} + \left(1 - \frac{9}{10}\frac{\nu_{ei}}{1.87\nu_{ei}+\nu_{en}^{\mathrm{MT}}}\right)\nu_{ei}$$
$$\left[\approx \nu_{en}^{\mathrm{MT}} + \nu_{ei} - \frac{9}{10}\frac{\nu_{ei}}{1.87\nu_{ei}+\nu_{en}^{\mathrm{MT}}}\right]$$

The two equations are both equivalent to

$$v_i - v_e = \frac{e}{m_e \nu_{ei\oplus}}E \tag{194}$$

Therefore to determine the solution we make an additional assumption: there is no net z momentum of the plasma.

$$n\left(m_i v_i + m_e v_e\right) + n_n m_n v_n = 0$$

$$n\left(m_i v_i + m_e v_e\right) + n_n m_n \frac{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} v_e + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}} v_i}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}} = 0 =$$

$$\left(nm_i + \frac{n_n m_n m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}\right)v_i + \left(nm_e + \frac{n_n m_n m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}}}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}\right)v_e$$

This is not necessarily a thing to expect, but, it will do for now. Therefore once we have computed E in view of (194),

$$\left(nm_i + \frac{n_n m_n m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}\right)v_i + \left(nm_e + \frac{n_n m_n m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}}}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}\right)\left(v_i - \frac{e}{m_e\nu_{ei\oplus}}E\right) = 0$$

ie

$$\frac{\left(nm_i + \frac{n_n m_n m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}} + nm_e + \frac{n_n m_n m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}}}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}\right)}{\left(nm_e + \frac{n_n m_n m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}}}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}\right)}v_i = \frac{e}{m_e\nu_{ei\oplus}}E$$

And we can see how we then compute v_n.

3.6 Initial setup Ohm's Law with viscosity

Again unlike in the usual case, we try to set

$$v_n = \frac{\frac{m_e}{m_e+m_n}\left(\frac{T_e}{m_e}\right)^{1/2} v_e + \frac{m_i}{m_i+m_n}\left(\frac{T_i}{m_i}+\frac{T_n}{m_n}\right)^{1/2} v_i}{\frac{m_e}{m_e+m_n}\left(\frac{T_e}{m_e}\right)^{1/2} + \frac{m_i}{m_i+m_n}\left(\frac{T_i}{m_i}+\frac{T_n}{m_n}\right)^{1/2}}$$

and

$$0 = \frac{\partial v_i}{\partial t}_z = \frac{e}{m_i}E - \frac{m_e}{m_i}\nu_{ei\oplus}\left(v_i - v_e\right)$$

$$\nu_{ei\oplus} = \frac{m_i m_n \nu_{ni}^{\mathrm{MT}} \nu_{en}^{\mathrm{MT}}}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}} + \left(1 - \frac{9}{10}\frac{\nu_{ei}}{1.87\nu_{ei}+\nu_{en}^{\mathrm{MT}}}\right)\nu_{ei}$$

$$\left[\approx \nu_{en}^{\mathrm{MT}} + \nu_{ei} - \frac{9}{10}\frac{\nu_{ei}}{1.87\nu_{ei}+\nu_{en}^{\mathrm{MT}}}\right]$$

$$\frac{m_e}{m_i}\nu_{ei\oplus}v_i = \frac{e}{m_i}E + \frac{m_e}{m_i}\nu_{ei\oplus}v_e$$

$$v_i = \frac{e}{m_e}\nu_{ei\oplus}^{-1}E + v_e$$

This gives us σ_i, β_i^e and gives us σ_n, β_n^e. Trouble is that here we might end up with v_e small compared to v_i which is not what we want.

Pick a different condition instead:

$$n\left(m_i v_i + m_e v_e\right) + n_n m_n v_n = 0$$

$$n\left(m_i v_i + m_e v_e\right) + n_n m_n \frac{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} v_e + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}} v_i}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}} = 0 =$$

$$\left(nm_i + \frac{n_n m_n m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}\right)v_i + \left(nm_e + \frac{n_n m_n m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}}}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}\right)v_e$$

Therefore our assumption is that $v_{i,n}$ are in a certain ratio to v_e. (This should be NEGATIVE!)

$$\beta_i^e = -\frac{nm_e m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}} + \left(n_n m_n + nm_e\right)m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}}}{\left(nm_i + n_n m_n\right)m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}} + nm_i m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}}} \tag{195}$$

$$v_n = \frac{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}\beta_i^e}{m_e\left(m_i+m_n\right)\nu_{ne}^{\mathrm{MT}} + m_i\left(m_e+m_n\right)\nu_{ni}^{\mathrm{MT}}}v_e = \beta_n^e v_e \tag{196}$$

Now dropping ω and Υ, put

$$\begin{aligned} 0 = \frac{\partial\left(v_e - v_i\right)}{\partial t} = & -\frac{\nabla\left(n_e T_e\right)}{n_e m_e} + \frac{\nabla\left(n_i T_i\right)}{n_i m_i} - \left[\frac{\nabla\cdot\Pi_e}{n_e m}\right]\left(v_e\right) - \left[\frac{\nabla\cdot\Pi_e}{n_e m}\right]\left(v_e^{neighbours}\right) - \left(\frac{q}{m_e}+\frac{q}{m_i}\right)E + \\ & + \left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n} - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(\beta_i^e - \beta_n^e\right)v_e + \left(1+\frac{n_e m_e}{n_i m_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\left(\nu_{en}+1.87\bar{\nu_{ei}}\right)}\nabla T\right) \\ & - \left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n} + \left(1+\frac{n_e m_e}{n_i m_i}\right)\left(1_{3\times 3} - \frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{en}+1.87\bar{\nu_{ei}}}\right)\bar{\nu_{ei}}\right)\left(1-\beta_i^e\right)v_e \end{aligned} \tag{197}$$

Thus

$$\begin{aligned} & \left[\frac{\nabla\cdot\Pi_e}{n_e m}\right]\left(v_e\right) + \left(-\left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n} - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(\beta_i^e-\beta_n^e\right) + \left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n} + \left(1+\frac{n_e m_e}{n_i m_i}\right)\left(1-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{en}+1.87}\right.\right.\right. \\ = & -\frac{\nabla\left(n_e T_e\right)}{n_e m_e} + \frac{\nabla\left(n_i T_i\right)}{n_i m_i} - \left[\frac{\nabla\cdot\Pi_e}{n_e m}\right]\left(v_e^{neighbours}\right) - \left(\frac{q}{m_e}+\frac{q}{m_i}\right)E + \left(1+\frac{n_e m_e}{n_i m_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\left(\nu_{en}+1.87\bar{\nu_{ei}}\right.\right. \end{aligned}$$

Therefore in this way we form χ^{-1} and let

$$v_e^0 = \chi\left(-\frac{\nabla\left(n_e T_e\right)}{n_e m_e} + \frac{\nabla\left(n_i T_i\right)}{n_i m_i} + \left(1+\frac{n_e m_e}{n_i m_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\left(\nu_{en}+1.87\bar{\nu_{ei}}\right)}\nabla T\right)\right)$$

$$\sigma_e = -\chi\left(\frac{q}{m_e}+\frac{q}{m_i}\right)$$

$$v_e = v_e^0 + \sigma_e E - \chi\left[\frac{\nabla\cdot\Pi_e}{n_e m}\right]\left(v_e^{neighbours}\right) \tag{198}$$

3.7 Ohm's Law for case that $J_{xy}=0$

Start again from just $\frac{\partial}{\partial t}\left(v_e - v_i\right)_z = 0$ for determining v_{ez} only.

$$\begin{aligned}\frac{\partial v_i}{\partial t} &= -\frac{1}{n_i m_i}\nabla\left(n_i T_i\right) - \frac{1}{n_i m_i}\nabla\cdot\Pi_i + \frac{e}{m_i}E - \omega_{ci}\times v_i \\ &\quad -\frac{m_n}{m_i+m_n}\nu_{in}^{\mathrm{MT}}\left(v_i - v_n\right) \\ &\quad +\frac{3}{2}\frac{\nu_{ie}}{m_i\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e - \frac{m_e}{m_i}\left(1_{3\times3}-\frac{9}{10}\frac{\nu_{ei}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\nu_{ie}\left(v_i - v_e\right)\end{aligned} \tag{199}$$

$$\begin{aligned}\frac{\partial v_e}{\partial t} &= -\frac{1}{n_e m_e}\nabla\left(n_e T_e\right) - \frac{1}{n_e m_e}\nabla\cdot\Pi_e - \frac{e}{m_e}E + \omega_{ce}\times v_e \\ &\quad -\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}\left(v_e - v_n\right) \\ &\quad -\frac{3}{2}\frac{\nu_{ei}}{m\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e - \left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\nu_{ei}\left(v_e - v_i\right)\end{aligned} \tag{200}$$

$$\begin{aligned}\frac{\partial\left(v_{ez}-v_{iz}\right)}{\partial t} &= -\frac{e}{m_e}E + \omega_{ce}\times v_e - \frac{e}{m_i}E + \omega_{ci}\times v_i \\ &\quad -\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}\left(v_e - v_n\right) + \frac{m_n}{m_i+m_n}\nu_{in}^{\mathrm{MT}}\left(v_i - v_n\right) \\ &\quad -\left(1+\frac{m}{M}\right)\frac{3}{2}\frac{\nu_{ei}}{m\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e - \left(1+\frac{m}{M}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\nu_{ei}\left(v_e - v_i\right)\end{aligned}$$

$$\begin{aligned}\frac{\partial\left(v_{ez}-v_{iz}\right)}{\partial t} &= -\left(\frac{q}{m_e}+\frac{q}{m_i}\right)E_z + \left(\omega_{ce}+\omega_{ci}\right)_x v_{iy} - \left(\omega_{ce}+\omega_{ci}\right)_y v_{ix} \\ &\quad +\left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n} - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(v_i - v_n\right)_z + \left(1+\frac{m_e}{m_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T\right)_z \\ &\quad -\left(\left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}+\left(1+\frac{m_e}{m_i}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}-\omega_{ce}\times\right)\left(v_e - v_i\right)\right)_z\end{aligned} \tag{201}$$

Now since $\left(v_e - v_i\right)_{x,y} = 0$,

$$\left(\left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}+\left(1+\frac{m_e}{m_i}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}-\omega_{ce}\times\right)\begin{pmatrix}0\\0\\v_{ez}-v_{iz}\end{pmatrix}\right)_z =$$

$$\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\left(v_{ez}-v_{iz}\right) + \left(1+\frac{m_e}{m_i}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)_{zz}\bar{\nu_{ei}}\left(v_{ez}-v_{iz}\right) - 0$$

Then

$$\begin{aligned}
0 &= -\left(\frac{q}{m_e}+\frac{q}{m_i}\right)E_z + \left(\omega_{ce}+\omega_{ci}\right)_x v_{iy} - \left(\omega_{ce}+\omega_{ci}\right)_y v_{ix} \\
&+ \left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n} - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(v_{iz}-v_{nz}\right) + \left(1+\frac{m_e}{m_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T\right)_z \\
&- \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\left(v_{ez}-v_{iz}\right) - \left(1+\frac{m_e}{m_i}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)_{zz}\bar{\nu_{ei}}\left(v_{ez}-v_{iz}\right) \qquad (202)
\end{aligned}$$

Now expand $v_i = v_i^0 + \beta^{ion}v_{ez} + \sigma^{ion}E_z$ where β^{ion} and σ^{ion} are 3-vectors.

$$\begin{aligned}
0 &= -\left(\frac{q}{m_e}+\frac{q}{m_i}\right)E_z + \left(1+\frac{m_e}{m_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T\right)_z \\
&+ \left(\omega_{ce}+\omega_{ci}\right)_x\left(v_{iy}^0+\beta_y^{ion}v_{ez}+\sigma_y^{ion}E_z\right) - \left(\omega_{ce}+\omega_{ci}\right)_y\left(v_{ix}^0+\beta_x^{ion}v_{ez}+\sigma_x^{ion}E_z\right) \qquad (203) \\
&+ \left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n} - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(v_i^0-v_n^0+\left(\beta_z^{ion}-\beta_z^{neut}\right)v_{ez}+\left(\sigma_z^{ion}-\sigma_z^{neut}\right)E_z\right)_z \\
&- \left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n} + \left(1+\frac{m_e}{m_i}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)_{zz}\bar{\nu_{ei}}\right)\left(v_{ez}-v_{iz}^0-\beta_z^{ion}v_{ez}-\sigma_z^{ion}E_z\right) \qquad (204)
\end{aligned}$$

ie

$$\begin{aligned}
0 &= -\left(\frac{q}{m_e}+\frac{q}{m_i}\right)E_z + \left(1+\frac{m_e}{m_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T\right)_z \\
&+ \left(\omega_{ce}+\omega_{ci}\right)_x v_{iy}^0 - \left(\omega_{ce}+\omega_{ci}\right)_y v_{ix}^0 \\
&+ \left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n} - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(v_{iz}^0-v_{nz}^0\right) \\
&- \left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n} + \left(1+\frac{m_e}{m_i}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)_{zz}\bar{\nu_{ei}}\right)\left(-v_{iz}^0\right) \\
&+ \left(\omega_{ce}+\omega_{ci}\right)_x\sigma_y^{ion}E_z - \left(\omega_{ce}+\omega_{ci}\right)_y\sigma_x^{ion}E_z \\
&+ \left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n} - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(\sigma_z^{ion}-\sigma_z^{neut}\right)E_z \\
&- \left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n} + \left(1+\frac{m_e}{m_i}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)_{zz}\bar{\nu_{ei}}\right)\left(-\sigma_z^{ion}E_z\right) \\
&+ \left(\omega_{ce}+\omega_{ci}\right)_x\left(\beta_y^{ion}v_{ez}\right) - \left(\omega_{ce}+\omega_{ci}\right)_y\left(\beta_x^{ion}v_{ez}\right) \\
&+ \left(\frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n} - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}\right)\left(\beta_z^{ion}-\beta_z^{neut}\right)v_{ez} \\
&- \left(\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n} + \left(1+\frac{m_e}{m_i}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)_{zz}\bar{\nu_{ei}}\right)\left(1-\beta_z^{ion}\right)v_{ez}
\end{aligned}$$

Therefore let

$$\nu^{Overall} = \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n} + \left(1+\frac{m_e}{m_i}\right)\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)_{zz}\bar{\nu_{ei}}$$

$$\nu_{in,en}^{stuff} = \frac{m_n\nu_{in}^{\mathrm{MT}}}{m_i+m_n} - \frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}$$

$$\mathrm{denom} = \left(\omega_{ce}+\omega_{ci}\right)_x\beta_y^{ion} - \left(\omega_{ce}+\omega_{ci}\right)_y\beta_x^{ion} + \nu_{in,en}^{stuff}\left(\beta_z^{ion}-\beta_z^{neut}\right) - \left(\nu^{Overall}\right)\left(1-\beta_z^{ion}\right) \qquad (205)$$

$$\sigma_e = \frac{-1}{\mathrm{denom}}\left(-\left(\frac{q}{m_e}+\frac{q}{m_i}\right) + \left(\omega_{ce}+\omega_{ci}\right)_x\sigma_y^{ion} - \left(\omega_{ce}+\omega_{ci}\right)_y\sigma_x^{ion} + \nu_{in,en}^{stuff}\left(\sigma_z^{ion}-\sigma_z^{neut}\right) + \nu^{Overall}\sigma_{ion}^z\right)$$

$$
\begin{aligned}
v_{0z}^{e} &= \frac{-1}{\text{denom}}\left(\left(1+\frac{m_e}{m_i}\right)\left(-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T\right)_z+\left(\omega_{ce}+\omega_{ci}\right)_x v_{iy}^0\right.\\
&\left.-\left(\omega_{ce}+\omega_{ci}\right)_y v_{ix}^0+\nu_{in,en}^{stuff}\left(v_{iz}^0-v_{nz}^0\right)+\nu^{Overall}v_{iz}^0\right) \qquad (206)
\end{aligned}
$$

3.8 Go again but with $dv_e/dt = 0$

I think the problem with $\frac{d}{dt}\left(v_e - v_i\right) = 0$ is that it doesn't really make sense in the weakly ionised case. Consider just inertialess electrons instead and it is much simpler: instantaneously set v_e from

$$
\begin{aligned}
\frac{dv_e}{dt}=0 &= -\frac{\nabla\left(n_eT_e\right)}{n_em_e}-\frac{q}{m_e}E+\omega_{ce}\times v_e+[\text{effect of new e's at } v_n]\\
&-\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}\left(v_e-v_n\right)-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e\\
&-\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}\left(v_e-v_i\right) \qquad (207)
\end{aligned}
$$

Let's say for now we continue to drop the ionisation contribution. Inconsistently but nvm we have to limit things.

Now put

$$
\nu_{3\times3}^{\mathrm{COEFF}}=\frac{m_n\nu_{en}^{\mathrm{MT}}}{m_e+m_n}+\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}
$$

$$
\begin{aligned}
\left(\nu_{3\times3}^{\mathrm{COEFF}}-\omega_{ce}\times\right)v_e &= -\frac{q}{m_e}E-\frac{\nabla\left(n_eT_e\right)}{n_em_e}-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e\\
&+\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}v_n+\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}v_i \qquad (208)
\end{aligned}
$$

Now be careful: sometimes we want $v_e\left(v_i, v_n\right)$ but sometimes we already have $v_i\left(v_e\right), v_n\left(v_e\right)$. Can write out that case also.

For the case where v_i, v_n are given:

$$
\begin{aligned}
v_e &= -\frac{q}{m}\left(\nu_{3\times3}^{\mathrm{COEFF}}-\omega_{ce}\times\right)^{-1}E+ \qquad (209)\\
&\left(\nu_{3\times3}^{\mathrm{COEFF}}-\omega_{ce}\times\right)^{-1}\left(-\frac{\nabla\left(n_eT_e\right)}{n_em_e}-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e+\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}v_n+\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}v_i\right)
\end{aligned}
$$

So that's fairly straightforward. We can use this to do frictional heating by determining v_e during heavy species acceleration.

For the case that v_i, v_n were found as linear functions of E, v_e:

$$
\begin{aligned}
\left(\nu_{3\times3}^{\mathrm{COEFF}}-\omega_{ce}\times\right)v_e &= -\frac{q}{m_e}E-\frac{\nabla\left(n_eT_e\right)}{n_em_e}-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e \qquad (210)\\
&+\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}\left(v_n^0+\sigma_nE+\beta_nv_e\right)+\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}\left(v_i^0+\sigma_iE+\beta_iv_e\right)\\
&= \left(-\frac{q}{m_e}+\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}\sigma_n+\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}\sigma_i\right)E\\
&-\frac{\nabla\left(n_eT_e\right)}{n_em_e}-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e\\
&+\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}\left(v_n^0+\beta_nv_e\right)+\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}\left(v_i^0+\beta_iv_e\right)
\end{aligned}
$$

$$\left(\nu_{3\times3}^{\mathrm{COEFF}}-\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}\beta_n-\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}\beta_i-\omega_{ce}\times\right)v_e \;=\; \left(-\frac{q}{m_e}+\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}\sigma_n+\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}\sigma_i\right)E \quad (211)$$
$$-\frac{\nabla\left(n_eT_e\right)}{n_em_e}-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\nabla T_e+\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}v_n^0+\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}v_i^0$$

3.8.1 With no x-y currents

$$\begin{aligned}\frac{dv_{ez}}{dt}=0 \;&=\; -\frac{q}{m_e}E_z+\left(\omega_{cex}v_{ey}-\omega_{cey}v_{ex}\right)+[\text{effect of new e's at } v_n]\\ &\quad -\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}\left(v_{ez}-v_{nz}\right)-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\left(\Upsilon_{e\heartsuit}\nabla T_e\right)_z\\ &\quad -\left(\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}\begin{pmatrix}0\\0\\v_{ez}-v_{iz}\end{pmatrix}\right)_z\end{aligned} \quad (212)$$

$$\left(\omega_{ce}\times v_e\right)_z=\omega_{cex}v_{ey}-\omega_{cey}v_{ex}=\omega_{cex}v_{iy}-\omega_{cey}v_{ix}$$

(Of course we must make sure that we take account of $\omega_{ce}\times$ when accelerating the ions.)

$$\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}v_{ez}+\left(\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}\begin{pmatrix}0\\0\\v_{ez}\end{pmatrix}\right)_z \;=\; -\frac{q}{m_e}E_z+\omega_{cex}v_{iy}-\omega_{cey}v_{ix}-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\left(\Upsilon_{e\heartsuit}\nabla T\right.$$
$$+\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}v_{nz}+\left(\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\nu\right.$$

Now what is

$$\left(\left(1_{3\times3}-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}\right)\bar{\nu_{ei}}\begin{pmatrix}0\\0\\v_{ez}\end{pmatrix}\right)_z=\left(1-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{zz}^{e\heartsuit}\right)\bar{\nu_{ei}}v_{ez}$$

$$\left(\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}+\left(1-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{zz}^{e\heartsuit}\right)\bar{\nu_{ei}}\right)v_{ez} \;=\; -\frac{q}{m_e}E_z+\omega_{cex}v_{iy}-\omega_{cey}v_{ix}-\frac{3}{2}\frac{\bar{\nu_{ei}}}{m_e\nu_{e\heartsuit}}\left(\Upsilon_{e\heartsuit}\nabla T_e\right)_z$$
$$+\frac{m_n}{m_e+m_n}\nu_{en}^{\mathrm{MT}}v_{nz}+\left(1-\frac{9}{10}\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{zz}^{e\heartsuit}\right)\bar{\nu_{ei}}v_{iz} \quad (214)$$

4 Viscous momentum flows

4.1 Source material

Per the NRL Formulary, let the rate-of-strain tensor be given by

$$W_{jk} = \frac{\partial v_j}{\partial x_k} + \frac{\partial v_k}{\partial x_j} - \frac{2}{3}\delta_{jk}\nabla \cdot v$$

and this is intended to apply in the magnetic coordinates used there.

Then define the viscous stress tensor Π: for a set of viscosity coefficients $\eta^{species}$, where arbitrarily we shall relabel $z = b,\ x = \perp,\ y = \wedge$ and from here on, the xyz notation of the magnetic coordinates is not used:

$$\Pi_{bb} = -\eta_0 W_{bb}$$

$$\Pi_{\perp b} = -\eta_2 W_{\perp b} - \eta_4 W_{\wedge b}$$

$$\Pi_{\perp\perp} = -\frac{\eta_0}{2}\left(W_{\perp\perp} + W_{\wedge\wedge}\right) - \frac{\eta_1}{2}\left(W_{\perp\perp} - W_{\wedge\wedge}\right) - \eta_3 W_{\perp\wedge}$$

$$\Pi_{\wedge b} = -\eta_2 W_{\wedge b} + \eta_4 W_{\perp b}$$

$$\Pi_{\wedge\wedge} = -\frac{\eta_0}{2}\left(W_{\perp\perp} + W_{\wedge\wedge}\right) + \frac{\eta_1}{2}\left(W_{\perp\perp} - W_{\wedge\wedge}\right) + \eta_3 W_{\perp\wedge}$$

$$\Pi_{\wedge\perp} = -\eta_1 W_{\perp\wedge} + \frac{\eta_3}{2}\left(W_{\perp\perp} - W_{\wedge\wedge}\right)$$

Per the NRL Formulary, then we shall have (where p shall mean momentum)

$$\frac{dp_\alpha}{dt} = m_\alpha n_\alpha \frac{dv_\alpha}{dt} + = -\nabla \cdot \Pi_\alpha$$

$$mn\frac{dv_x}{dt} + = -\frac{\partial}{\partial x}\Pi_{xx} - \frac{\partial}{\partial y}\Pi_{xy} - \frac{\partial}{\partial z}\Pi_{xz} \tag{215}$$

in magnetic coordinates. (Or perhaps transposed but it doesn't matter as Π is symmetric.)

4.1.1 Definition of viscosity coefficients η

Looking here at ion viscosity; electron is a bit different but not very.

First, Braginskii $\omega/\nu \to \infty$, fully ionised :

$$\eta_0 = 0.96\frac{nT}{\nu} = 0.96\eta_* \tag{216}$$

$$\eta_2 = \frac{6}{5}\eta_*\frac{\nu^2}{\omega^2} \tag{217}$$

$$\eta_4 = \eta_*\frac{\nu}{\omega} \tag{218}$$

He has also $\eta_3 = \frac{1}{2}\eta_4$ and $\eta_1 = \frac{1}{4}\eta_2$, exact for ions but only rough for electrons.

Landau-Lifschitz Volume 10, Pitaevskii & Lifshitz, 1981 has

$$\eta_1 = \frac{2\pi^{1/2}\left(Ze\right)^4 \lambda_i n_i^2}{5\left(MT\right)^{1/2}\omega_{ci}^2} = \frac{1}{4}\eta_2 \tag{219}$$

$$\eta_3 = \frac{nT}{2\omega_{ci}} = \frac{1}{2}\eta_4 \tag{220}$$

which gives a little idea of how the numerical factors might be arrived at(?).

It is still a fair approximation for the electron tensor that $\eta_1 = \frac{1}{4}\eta_2$ and $\eta_3 = \frac{1}{2}\eta_4$.

4.2 My speculative viscous stress tensor formula

I have yet to get hold of a source that treats even the fully ionised case for general ω/ν; generally texts eschew the subject saying that the expressions are awkward. Instead we have the unmagnetised case and the highly magnetised case.

If we take the m_α from the left hand side then we see that

$$\frac{\eta_0}{m} \propto \frac{nT}{m\nu}$$

showing that η_0/m is akin to the thermal conductivity κ. In the case of a neutral monatomic gas, [Reif2010] Formula 12-4-12 gives the relationship

My speculative version for the general case, based on the Alfven-type conductivity tensor that appears in the thermal conductivities in the NRL Formulary, involves modifying only the viscosity coefficients:

$$\eta_\parallel = \eta_0 \tag{221}$$

$$\eta_2 = \eta_\perp \approx \frac{1}{0.96}\frac{6}{5}\eta_\parallel \frac{\nu^2}{\omega^2+\nu^2} \tag{222}$$

$$\eta_4 = \eta_\times = \frac{1}{0.96}\eta_\parallel \frac{\nu\omega}{\omega^2+\nu^2} \tag{223}$$

$$\eta_1 = \frac{1}{4}\eta_2 \; ; \; \eta_3 = \frac{1}{2}\eta_4 \tag{224}$$

Viscous effects and thermal conductivity are kindred processes (emphasised by Braginskii) so this guess seems reasonable - though we should get hold of a source to verify.

NOTE: The above is roughly an approximation to the NRL Formulary version. See also Braginskii p.253 which appears to disagree.

Note: from [NRLFormulary], the relevant ν is "ν_i" for ions, and the relevant ω is ω_{ion}, however the factor of m_{ion}/m_e cancels with the fact that the momentum here is denominated in g cm/s – so we deal with the flux of $m_{ion}n_{ion}v_{ion}$ and $m_e n_e v_e$.

To go even further: we need to deal with multicomponent plasma. One obvious idea is to use the expressions from thermal conductivity to substitute for part of η_0, so as to create dependence on neutral collisions (with viscosity cross section of course).

Zhdanov is the book that treats multicomponent, contains the definitive information but again looks at the high B-field case, using the Grad(1949) method of moments as with Balescu. Therefore a good way in the future might be to buy the Zhdanov book, use it for the multicomponent high-B-field case, and as here intuit what might happen for intermediate B fields.

Now in my speculative (and for electrons, simplified) version let's rewrite :

$$\Pi_{bb} = -\eta_\parallel W_{bb} \tag{225}$$

$$\Pi_{\perp b} = -\eta_\perp W_{\perp b} - \eta_\times W_{\wedge b} \tag{226}$$

$$\Pi_{\perp\perp} = -\frac{\eta_\parallel}{2}\left(W_{\perp\perp}+W_{\wedge\wedge}\right) - \frac{\eta_\perp}{8}\left(W_{\perp\perp}-W_{\wedge\wedge}\right) - \frac{\eta_\times}{2}W_{\perp\wedge} \tag{227}$$

$$\Pi_{\wedge b} = -\eta_\perp W_{\wedge b} + \eta_\times W_{\perp b} \tag{228}$$

$$\Pi_{\wedge\wedge} = -\frac{\eta_\parallel}{2}\left(W_{\perp\perp}+W_{\wedge\wedge}\right) + \frac{\eta_\perp}{8}\left(W_{\perp\perp}-W_{\wedge\wedge}\right) + \frac{\eta_\times}{2}W_{\perp\wedge} \tag{229}$$

$$\Pi_{\wedge\perp} = -\frac{\eta_\perp}{4}W_{\perp\wedge} + \frac{\eta_\times}{4}\left(W_{\perp\perp}-W_{\wedge\wedge}\right) \tag{230}$$

Or,

$$\Pi_{bb} = -\eta_{\|}\left(2\frac{\partial v_b}{\partial b} - \frac{2}{3}\nabla\cdot v\right) \tag{231}$$

$$\Pi_{\perp b} = -\eta_{\perp}\left(\frac{\partial v_{\perp}}{\partial b} + \frac{\partial v_b}{\partial \perp}\right) - \eta_{\times}\left(\frac{\partial v_{\wedge}}{\partial b} + \frac{\partial v_b}{\partial \wedge}\right) \tag{232}$$

$$\Pi_{\perp\perp} = -\eta_{\|}\left(\frac{\partial v_{\perp}}{\partial \perp} + \frac{\partial v_{\wedge}}{\partial \wedge} - \frac{2}{3}\nabla\cdot v\right) - \frac{\eta_{\perp}}{4}\left(\frac{\partial v_{\perp}}{\partial \perp} - \frac{\partial v_{\wedge}}{\partial \wedge}\right) - \frac{\eta_{\times}}{2}\left(\frac{\partial v_{\wedge}}{\partial \perp} + \frac{\partial v_{\perp}}{\partial \wedge}\right) \tag{233}$$

$$\Pi_{\wedge b} = -\eta_{\perp}\left(\frac{\partial v_{\wedge}}{\partial b} + \frac{\partial v_b}{\partial \wedge}\right) + \eta_{\times}\left(\frac{\partial v_{\perp}}{\partial b} + \frac{\partial v_b}{\partial \perp}\right) \tag{234}$$

$$\Pi_{\wedge\wedge} = -\eta_{\|}\left(\frac{\partial v_{\perp}}{\partial \perp} + \frac{\partial v_{\wedge}}{\partial \wedge} - \frac{2}{3}\nabla\cdot v\right) + \frac{\eta_{\perp}}{4}\left(\frac{\partial v_{\perp}}{\partial \perp} - \frac{\partial v_{\wedge}}{\partial \wedge}\right) + \frac{\eta_{\times}}{2}\left(\frac{\partial v_{\wedge}}{\partial \perp} + \frac{\partial v_{\perp}}{\partial \wedge}\right) \tag{235}$$

$$\Pi_{\wedge\perp} = -\frac{\eta_{\perp}}{4}\left(\frac{\partial v_{\wedge}}{\partial \perp} + \frac{\partial v_{\perp}}{\partial \wedge}\right) + \frac{\eta_{\times}}{2}\left(\frac{\partial v_{\perp}}{\partial \perp} - \frac{\partial v_{\wedge}}{\partial \wedge}\right) \tag{236}$$

4.3 Implementation in a Cartesian simulation

Preliminary Let

$$\perp = \frac{v - (v\cdot b)\, b}{\|v - (v\cdot b)\, b\|} \tag{237}$$

$$\wedge = b \times \perp \tag{238}$$

Clearly

$$\begin{pmatrix} v_b \\ v_{\perp} \\ v_{\wedge} \end{pmatrix} = \begin{pmatrix} b_x & b_y & b_z \\ \perp_x & \perp_y & \perp_z \\ \wedge_x & \wedge_y & \wedge_z \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$$

$$\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} b_x & b_y & b_z \\ \perp_x & \perp_y & \perp_z \\ \wedge_x & \wedge_y & \wedge_z \end{pmatrix}^{-1} \begin{pmatrix} v_b \\ v_{\perp} \\ v_{\wedge} \end{pmatrix} = \begin{pmatrix} b_x & \perp_x & \wedge_x \\ b_y & \perp_y & \wedge_y \\ b_z & \perp_z & \wedge_z \end{pmatrix} \begin{pmatrix} v_b \\ v_{\perp} \\ v_{\wedge} \end{pmatrix} \tag{239}$$

since

$$\begin{pmatrix} b_x & b_y & b_z \\ \perp_x & \perp_y & \perp_z \\ \wedge_x & \wedge_y & \wedge_z \end{pmatrix} \begin{pmatrix} b_x & \perp_x & \wedge_x \\ b_y & \perp_y & \wedge_y \\ b_z & \perp_z & \wedge_z \end{pmatrix} = \begin{pmatrix} b\cdot b & b\cdot \perp & b\cdot \wedge \\ b\cdot \perp & \perp\cdot\perp & \perp\cdot\wedge \\ b\cdot \wedge & \perp\cdot\wedge & \wedge\cdot\wedge \end{pmatrix}$$

Therefore [here b is not changing since we are just looking at the changes due to viscosity],

$$\frac{\partial}{\partial t}\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} b_x & \perp_x & \wedge_x \\ b_y & \perp_y & \wedge_y \\ b_z & \perp_z & \wedge_z \end{pmatrix} \frac{\partial}{\partial t}\begin{pmatrix} v_b \\ v_{\perp} \\ v_{\wedge} \end{pmatrix} \tag{240}$$

Now consider that from (215), we have such as

$$mn\frac{\partial v_b}{\partial t} = -\nabla\cdot\begin{pmatrix} \Pi_{b\perp} \\ \Pi_{b\wedge} \\ \Pi_{bb} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial \perp} \\ \frac{\partial}{\partial \wedge} \\ \frac{\partial}{\partial b} \end{pmatrix}\cdot\begin{pmatrix} \Pi_{b\perp} \\ \Pi_{b\wedge} \\ \Pi_{bb} \end{pmatrix} \tag{241}$$

Therefore

$$mn\frac{\partial v_x}{\partial t} = -b_x\begin{pmatrix} \frac{\partial}{\partial \perp} \\ \frac{\partial}{\partial \wedge} \\ \frac{\partial}{\partial b} \end{pmatrix}\cdot\begin{pmatrix} \Pi_{b\perp} \\ \Pi_{b\wedge} \\ \Pi_{bb} \end{pmatrix} - \perp_x\begin{pmatrix} \frac{\partial}{\partial \perp} \\ \frac{\partial}{\partial \wedge} \\ \frac{\partial}{\partial b} \end{pmatrix}\cdot\begin{pmatrix} \Pi_{\perp\perp} \\ \Pi_{\perp\wedge} \\ \Pi_{\perp b} \end{pmatrix} - \wedge_x\begin{pmatrix} \frac{\partial}{\partial \perp} \\ \frac{\partial}{\partial \wedge} \\ \frac{\partial}{\partial b} \end{pmatrix}\cdot\begin{pmatrix} \Pi_{\wedge\perp} \\ \Pi_{\wedge\wedge} \\ \Pi_{\wedge b} \end{pmatrix} \tag{242}$$

Motivation We are going to want to regard the viscous effect as a flow of momentum from one cell to another, by applying the divergence theorem. What we are then aiming to compute in this approach, is the amount of p_x to send across a cell boundary given the nearby values of v which determine an estimate of ∇v, and given b at the edge. This is not just so we can actually carry it out; we need something we can actually insert into $\frac{\partial(v_e - v_i)}{\partial t} = 0$. However, getting a linear combination at runtime is sufficient; our algebra can anticipate it.

Divergence theorem application In a given coordinate system,

$$\int_{cell} \nabla \cdot f \; dx = \int_{\partial cell} f \cdot \hat{n} = \sum_{edges} f_{avg} \cdot \perp^{edge}_{(outward)} \tag{243}$$

where $\perp^{edge}$ is an outward normal vector that has the length of the edge. (For the sign consider a square with $(1,0,0)$ as the normal on the right and $(-1,0,0)$ on the left ; $\int \frac{\partial f_x}{\partial x}$ gets positive from right.)

Note that $\perp^{edge}_z = 0$ for 2D.

So each negated divergence term in $mn\frac{\partial v_x}{\partial t} = -\nabla \cdot f$, for any f, will represent an INWARD derivative on each edge, summed, ***to determine what is a net flow of*** p_x because it will feature in the divergence of the neighbouring cell, equal and opposite. However, Π is then mostly negative terms; so basically a positive outward gradient of v is contributing to an increase of p here – as it should be.

Because divergence is invariant to transformations between orthonormal coordinate bases,

$$\begin{pmatrix} \frac{\partial}{\partial b} \\ \frac{\partial}{\partial \perp} \\ \frac{\partial}{\partial \wedge} \end{pmatrix} \cdot \begin{pmatrix} \Pi_{bb} \\ \Pi_{b\perp} \\ \Pi_{b\wedge} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} \cdot \left(\begin{pmatrix} b_x & \perp_x & \wedge_x \\ b_y & \perp_y & \wedge_y \\ b_z & \perp_z & \wedge_z \end{pmatrix} \begin{pmatrix} \Pi_{bb} \\ \Pi_{b\perp} \\ \Pi_{b\wedge} \end{pmatrix} \right) \tag{244}$$

so proceeding that way round and applying (243),

$$\begin{aligned} mn\frac{\partial v_x}{\partial t} + &= -\left(b_x \begin{pmatrix} b_x & \perp_x & \wedge_x \\ b_y & \perp_y & \wedge_y \\ b_z & \perp_z & \wedge_z \end{pmatrix} \begin{pmatrix} \Pi_{bb} \\ \Pi_{b\perp} \\ \Pi_{b\wedge} \end{pmatrix} + \perp_x \begin{pmatrix} b_x & \perp_x & \wedge_x \\ b_y & \perp_y & \wedge_y \\ b_z & \perp_z & \wedge_z \end{pmatrix} \begin{pmatrix} \Pi_{\perp b} \\ \Pi_{\perp\perp} \\ \Pi_{\perp\wedge} \end{pmatrix} + \wedge_x \begin{pmatrix} b_x & \perp_x & \wedge_x \\ b_y & \perp_y & \wedge_y \\ b_z & \perp_z & \wedge_z \end{pmatrix} \right. \\ &= -\left(\begin{pmatrix} b_x & \perp_x & \wedge_x \\ b_y & \perp_y & \wedge_y \\ b_z & \perp_z & \wedge_z \end{pmatrix} \begin{pmatrix} b_x\Pi_{bb} + \perp_x \Pi_{\perp b} + \wedge_x \Pi_{\wedge b} \\ b_x\Pi_{b\perp} + \perp_x \Pi_{\perp\perp} + \wedge_x \Pi_{\wedge\perp} \\ b_x\Pi_{b\wedge} + \perp_x \Pi_{\perp\wedge} + \wedge_x \Pi_{\wedge\wedge} \end{pmatrix} \right) \cdot \perp^{edge} \\ &= -\left(\begin{pmatrix} b_x & \perp_x & \wedge_x \\ b_y & \perp_y & \wedge_y \\ b_z & \perp_z & \wedge_z \end{pmatrix} \begin{pmatrix} b_x\Pi_{bb} + \perp_x \Pi_{\perp b} + \wedge_x \Pi_{\wedge b} \\ b_x\Pi_{b\perp} + \perp_x \Pi_{\perp\perp} + \wedge_x \Pi_{\wedge\perp} \\ b_x\Pi_{b\wedge} + \perp_x \Pi_{\perp\wedge} + \wedge_x \Pi_{\wedge\wedge} \end{pmatrix} \right)^{Transpose} \perp^{edge} \\ &= -\begin{pmatrix} b_x\Pi_{bb} + \perp_x \Pi_{\perp b} + \wedge_x \Pi_{\wedge b} \\ b_x\Pi_{b\perp} + \perp_x \Pi_{\perp\perp} + \wedge_x \Pi_{\wedge\perp} \\ b_x\Pi_{b\wedge} + \perp_x \Pi_{\perp\wedge} + \wedge_x \Pi_{\wedge\wedge} \end{pmatrix} \cdot \left(\begin{pmatrix} b_x & b_y & b_z \\ \perp_x & \perp_y & \perp_z \\ \wedge_x & \wedge_y & \wedge_z \end{pmatrix} \perp^{edge} \right) \end{aligned}$$

This then shows what we could have got by going the other way: putting $\perp^{edge}$ into magnetic coordinates and applying the divergence theorem to div in magnetic coordinates.

Transformed viscous tensor version Moreover we can also put it another way: from (245),

$$mn\frac{\partial v_x}{\partial t} + = -\left(\begin{pmatrix} b_x & \perp_x & \wedge_x \\ b_y & \perp_y & \wedge_y \\ b_z & \perp_z & \wedge_z \end{pmatrix} \begin{pmatrix} \Pi_{bb} & \Pi_{\perp b} & \Pi_{\wedge b} \\ \Pi_{b\perp} & \Pi_{\perp\perp} & \Pi_{\wedge\perp} \\ \Pi_{b\wedge} & \Pi_{\perp\wedge} & \Pi_{\wedge\wedge} \end{pmatrix} \begin{pmatrix} b_x \\ \perp_x \\ \wedge_x \end{pmatrix} \right) \cdot \perp^{edge} \tag{248}$$

from which it is seen that the Cartesian momentum vector fluxes are found as the columns of

$$\begin{pmatrix} \Pi_{xx} & \Pi_{xy} & \Pi_{xz} \\ \Pi_{yx} & \Pi_{yy} & \Pi_{yz} \\ \Pi_{zx} & \Pi_{zy} & \Pi_{zz} \end{pmatrix} = \begin{pmatrix} b_x & \perp_x & \wedge_x \\ b_y & \perp_y & \wedge_y \\ b_z & \perp_z & \wedge_z \end{pmatrix} \begin{pmatrix} \Pi_{bb} & \Pi_{\perp b} & \Pi_{\wedge b} \\ \Pi_{b\perp} & \Pi_{\perp\perp} & \Pi_{\wedge\perp} \\ \Pi_{b\wedge} & \Pi_{\perp\wedge} & \Pi_{\wedge\wedge} \end{pmatrix} \begin{pmatrix} b_x & b_y & b_z \\ \perp_x & \perp_y & \perp_z \\ \wedge_x & \wedge_y & \wedge_z \end{pmatrix} \tag{249}$$

which can then be dotted with the outward edge normal. Notice that the CMVF matrix $\Pi_{(xyz)}$ is symmetric since the transpose is the same:

$$\Pi_{(xyz)} = \begin{pmatrix} b_x & b_y & b_z \\ \perp_x & \perp_y & \perp_z \\ \wedge_x & \wedge_y & \wedge_z \end{pmatrix}^T \begin{pmatrix} \Pi_{bb} & \Pi_{\perp b} & \Pi_{\wedge b} \\ \Pi_{b\perp} & \Pi_{\perp\perp} & \Pi_{\wedge\perp} \\ \Pi_{b\wedge} & \Pi_{\perp\wedge} & \Pi_{\wedge\wedge} \end{pmatrix} \begin{pmatrix} b_x & \perp_x & \wedge_x \\ b_y & \perp_y & \wedge_y \\ b_z & \perp_z & \wedge_z \end{pmatrix}^T = \Pi_{(xyz)}$$

Finally, we can then say that

$$\begin{pmatrix} mn\frac{\partial v_x}{\partial t} \\ mn\frac{\partial v_y}{\partial t} \\ mn\frac{\partial v_{zx}}{\partial t} \end{pmatrix} += \begin{pmatrix} \Pi_{xx} & \Pi_{xy} & \Pi_{xz} \\ \Pi_{yx} & \Pi_{yy} & \Pi_{yz} \\ \Pi_{zx} & \Pi_{zy} & \Pi_{zz} \end{pmatrix} \perp^{edge} \tag{250}$$

This gives two routes:

4.3.1 Route 1 - work more in magnetic coordinates

Note that for a direction vector a,

$$\begin{aligned} \frac{\partial v_b}{\partial a} &= a_x\frac{\partial v_b}{\partial x} + a_y\frac{\partial v_b}{\partial y} + a_z\frac{\partial v_b}{\partial z} = \left(a_x b_x\frac{\partial v_x}{\partial x} + a_x b_y\frac{\partial v_y}{\partial x} + a_x b_z\frac{\partial v_z}{\partial x}\right) + a_y\frac{\partial v_b}{\partial y} + a_z\frac{\partial v_b}{\partial z} \\ &= b^T\left[\frac{\partial v_i}{\partial x_j}\right]a \end{aligned} \tag{251}$$

Therefore

1. Given $\frac{\partial v_i}{\partial x_j}$ as linear combinations of nearby v, we can infer magnetic-coordinate derivatives $\frac{\partial v_\wedge}{\partial \wedge}, \frac{\partial v_\perp}{\partial b}$ etc as linear combinations of nearby v.
2. Therefore infer the elements of the magnetic-coordinated tensor $\Pi_{(b\perp\wedge)}$ as linear combinations of nearby v.
3. Infer as linear combinations, the Cartesian momentum flux vectors that form $\Pi_{(xyz)}$ found as (249).
4. Infer as linear combinations the outward momentum flux in x, y, z by dotting each column of $\Pi_{(xyz)}$ with the edge normal vector.

4.3.2 Route 2 - work more in native coordinates

We can instead rely more on analytical rearrangements that might or might not reveal something clarifying about the nature of viscosity.

1. Work out formulae for coefficients on $\frac{\partial v_i}{\partial x_j}$ in each element of the partially transformed matrix

$$\begin{pmatrix} \Pi_{bb} & \Pi_{\perp b} & \Pi_{\wedge b} \\ \Pi_{b\perp} & \Pi_{\perp\perp} & \Pi_{\wedge\perp} \\ \Pi_{b\wedge} & \Pi_{\perp\wedge} & \Pi_{\wedge\wedge} \end{pmatrix} \begin{pmatrix} b_x & b_y & b_z \\ \perp_x & \perp_y & \perp_z \\ \wedge_x & \wedge_y & \wedge_z \end{pmatrix}$$

   and therefore be able to infer each of these elements as a linear combination of $\frac{\partial v_i}{\partial x_j}$.

2. Use (247): dotting with the edge vector in magnetic coordinates, infer as linear combinations, the outward momentum flux in x, y, z.

4.3.3 Route 3

Or, we could go even further: work out formulae for coefficients on $\frac{\partial v_i}{\partial x_j}$ in each element of $\Pi_{(xyz)}$ and so skip Step 1 and Step 2 of Route 1.

To do each of these we are only applying (251) repeatedly, so even numerically we should probably come to the same result...

4.4 Calculating coefficients in the Cartesian viscous tensor $\Pi_{(xyz)}$

Using the rule (251) we can compute the coefficient c_{xy} on $\frac{\partial v_x}{\partial y}$ (etc) in each of the elements of the transformed tensor (249). Note that up to now we have not used (237) and it seems to not make any difference to anything. We can pick arbitrary axes orthogonal to b and have no effect, since $v_\perp$ never appears in anything, only such as $\frac{\partial v_\wedge}{\partial \perp}$ and we can say little about what that will be.

Firstly

$$\begin{aligned}
\Pi_{xx} &= b_x\left(b_x\Pi_{bb} + \perp_x \Pi_{\perp b} + \wedge_x \Pi_{\wedge b}\right) \\
&\quad + \perp_x \left(b_x\Pi_{b\perp} + \perp_x \Pi_{\perp\perp} + \wedge_x \Pi_{\wedge\perp}\right) \\
&\quad + \wedge_x \left(b_x\Pi_{b\wedge} + \perp_x \Pi_{\perp\wedge} + \wedge_x \Pi_{\wedge\wedge}\right) \\
&= b_x^2\Pi_{bb} + \perp_x^2 \Pi_{\perp\perp} + \wedge_x^2 \Pi_{\wedge\wedge} \\
&\quad + 2b_x \perp_x \Pi_{\perp b} + 2\wedge_x \perp_x \Pi_{\wedge\perp} + 2b_x \wedge_x \Pi_{b\wedge}
\end{aligned} \tag{252}$$

Suppose we have a go at Π_{xx}. First substitute the expressions for Π_{bb}:

$$\begin{aligned}
\Pi_{xx} &= b_x^2\left(-\eta_\parallel\left(2\frac{\partial v_b}{\partial b} - \frac{2}{3}\nabla\cdot v\right)\right) \\
&\quad + \perp_x^2\left(-\eta_\parallel\left(\frac{\partial v_\perp}{\partial \perp} + \frac{\partial v_\wedge}{\partial \wedge} - \frac{2}{3}\nabla\cdot v\right) - \frac{\eta_\perp}{4}\left(\frac{\partial v_\perp}{\partial \perp} - \frac{\partial v_\wedge}{\partial \wedge}\right) - \frac{\eta_\times}{2}\left(\frac{\partial v_\wedge}{\partial \perp} + \frac{\partial v_\perp}{\partial \wedge}\right)\right) \\
&\quad + \wedge_x^2\left(-\eta_\parallel\left(\frac{\partial v_\perp}{\partial \perp} + \frac{\partial v_\wedge}{\partial \wedge} - \frac{2}{3}\nabla\cdot v\right) + \frac{\eta_\perp}{4}\left(\frac{\partial v_\perp}{\partial \perp} - \frac{\partial v_\wedge}{\partial \wedge}\right) + \frac{\eta_\times}{2}\left(\frac{\partial v_\wedge}{\partial \perp} + \frac{\partial v_\perp}{\partial \wedge}\right)\right) \\
&\quad + 2b_x \perp_x \left(-\eta_\perp\left(\frac{\partial v_\perp}{\partial b} + \frac{\partial v_b}{\partial \perp}\right) - \eta_\times\left(\frac{\partial v_\wedge}{\partial b} + \frac{\partial v_b}{\partial \wedge}\right)\right) \\
&\quad + 2\wedge_x \perp_x \left(-\frac{\eta_\perp}{4}\left(\frac{\partial v_\wedge}{\partial \perp} + \frac{\partial v_\perp}{\partial \wedge}\right) + \frac{\eta_\times}{2}\left(\frac{\partial v_\perp}{\partial \perp} - \frac{\partial v_\wedge}{\partial \wedge}\right)\right) \\
&\quad + 2b_x \wedge_x \left(-\eta_\perp\left(\frac{\partial v_\wedge}{\partial b} + \frac{\partial v_b}{\partial \wedge}\right) + \eta_\times\left(\frac{\partial v_\perp}{\partial b} + \frac{\partial v_b}{\partial \perp}\right)\right)
\end{aligned} \tag{253}$$

We will never aim to combine the terms in different η so we might as well handle each of them separately, and do some simplification before continuing. Parallel part:

$$\begin{aligned}
\Pi_{xx} \;+\!= \; & -\eta_\parallel\left(b_x^2\left(2\frac{\partial v_b}{\partial b} - \frac{2}{3}\nabla\cdot v\right) + \perp_x^2\left(\frac{\partial v_\perp}{\partial \perp} + \frac{\partial v_\wedge}{\partial \wedge} - \frac{2}{3}\nabla\cdot v\right) + \wedge_x^2\left(\frac{\partial v_\perp}{\partial \perp} + \frac{\partial v_\wedge}{\partial \wedge} - \frac{2}{3}\nabla\cdot v\right)\right) \\
= \; & \eta_\parallel\frac{2}{3}\nabla\cdot v - \eta_\parallel\left(2b_x^2\frac{\partial v_b}{\partial b} + \left(\perp_x^2 + \wedge_x^2\right)\left(\frac{\partial v_\perp}{\partial \perp} + \frac{\partial v_\wedge}{\partial \wedge}\right)\right)
\end{aligned}$$

Perpendicular part:

$$\begin{aligned}
\Pi_{xx} \;+\!= \; & -\frac{\eta_\perp}{4}\left(\perp_x^2 - \wedge_x^2\right)\left(\frac{\partial v_\perp}{\partial \perp} - \frac{\partial v_\wedge}{\partial \wedge}\right) \\
& -2\eta_\perp\left(b_x \perp_x \left(\frac{\partial v_\perp}{\partial b} + \frac{\partial v_b}{\partial \perp}\right) + \frac{\wedge_x \perp_x}{4}\left(\frac{\partial v_\wedge}{\partial \perp} + \frac{\partial v_\perp}{\partial \wedge}\right) + b_x \wedge_x \left(\frac{\partial v_\wedge}{\partial b} + \frac{\partial v_b}{\partial \wedge}\right)\right)
\end{aligned}$$

Cross part:

$$\begin{aligned}\Pi_{xx} \;+= \;& -\frac{\eta_\times}{2}\left(\perp_x^2 - \wedge_x^2\right)\left(\frac{\partial v_\wedge}{\partial \perp} + \frac{\partial v_\perp}{\partial \wedge}\right) \\ & +2\eta_\times\left(-b_x \perp_x \left(\frac{\partial v_\wedge}{\partial b} + \frac{\partial v_b}{\partial \wedge}\right) + \frac{\wedge_x \perp_x}{2}\left(\frac{\partial v_\perp}{\partial \perp} - \frac{\partial v_\wedge}{\partial \wedge}\right) + b_x \wedge_x \left(\frac{\partial v_\perp}{\partial b} + \frac{\partial v_b}{\partial \perp}\right)\right)\end{aligned}$$

Therefore the coefficient on $\frac{\partial v_x}{\partial x}$ is

$$\begin{aligned}c_{xx} \;=\; & \eta_\parallel \frac{2}{3} - \eta_\parallel\left(2b_x^2 b_x b_x + \left(\perp_x^2 + \wedge_x^2\right)^2\right) \\ & -\frac{\eta_\perp}{4}\left(\perp_x^2 - \wedge_x^2\right)^2 - 2\eta_\perp\left(2\left(b_x \perp_x\right)^2 + \frac{\left(\wedge_x \perp_x\right)^2}{2} + 2\left(b_x \wedge_x\right)^2\right) \\ & -\eta_\times\left(\perp_x^2 - \wedge_x^2\right)\left(\wedge_x \perp_x\right) + 2\eta_\times\left(-2b_x \perp_x \wedge_x b_x + \frac{\wedge_x \perp_x}{2}\left(\perp_x^2 - \wedge_x^2\right) + 2b_x \wedge_x b_x \perp_x\right)\end{aligned}$$

Work on the $\eta_\perp$ coefficient:

$$-\frac{1}{4}\left(\perp_x^2 - \wedge_x^2\right)^2 - \left(\wedge_x \perp_x\right)^2 - 4\left(\left(b_x \perp_x\right)^2 + \left(b_x \wedge_x\right)^2\right)$$

$$= -\frac{\left(\perp_x^2 + \wedge_x^2\right)^2}{4} - 4\left(\left(b_x \perp_x\right)^2 + \left(b_x \wedge_x\right)^2\right)$$

Cross coefficient:

$$-4b_x \perp_x \wedge_x b_x + 4b_x \wedge_x b_x \perp_x = 0$$

Therefore

$$\begin{aligned}c_{xx} \;=\; & \eta_\parallel \frac{2}{3} - \eta_\parallel\left(2b_x^4 + \left(\perp_x^2 + \wedge_x^2\right)^2\right) \\ & -\eta_\perp\left(\frac{\left(\perp_x^2 + \wedge_x^2\right)^2}{4} + 4\left(\left(b_x \perp_x\right)^2 + \left(b_x \wedge_x\right)^2\right)\right)\end{aligned}$$

Even it's own on-diagonal coefficient looks forbidding. Better to go with Route 1?

4.4.1 Work out something for the general element

Element ij is given by

$$\begin{aligned}\Pi_{ij} \;=\; & b_i\left(b_j \Pi_{bb} + \perp_j \Pi_{\perp b} + \wedge_j \Pi_{\wedge b}\right) \\ & + \perp_i \left(b_j \Pi_{b\perp} + \perp_j \Pi_{\perp\perp} + \wedge_j \Pi_{\wedge\perp}\right) \\ & + \wedge_i \left(b_j \Pi_{b\wedge} + \perp_j \Pi_{\perp\wedge} + \wedge_j \Pi_{\wedge\wedge}\right) \\ =\; & b_i b_j \Pi_{bb} + \perp_i \perp_j \Pi_{\perp\perp} + \wedge_i \wedge_j \Pi_{\wedge\wedge} \\ & + \left(b_i \perp_j + \perp_i b_j\right)\Pi_{b\perp} + \left(\wedge_i \perp_j + \wedge_j \perp_i\right)\Pi_{\wedge\perp} + \left(b_i \wedge_j + b_j \wedge_i\right)\Pi_{b\wedge}\end{aligned} \tag{254}$$

Substitute the formulas (231-236) for Π_{bb}:

$$\begin{aligned}
\Pi_{ij} &= b_i b_j \left(-\eta_{\|}\left(2\frac{\partial v_b}{\partial b} - \frac{2}{3}\nabla\cdot v\right)\right) \\
&+ \perp_i \perp_j \left(-\eta_{\|}\left(\frac{\partial v_\perp}{\partial \perp} + \frac{\partial v_\wedge}{\partial \wedge} - \frac{2}{3}\nabla\cdot v\right) - \frac{\eta_\perp}{4}\left(\frac{\partial v_\perp}{\partial \perp} - \frac{\partial v_\wedge}{\partial \wedge}\right) - \frac{\eta_\times}{2}\left(\frac{\partial v_\wedge}{\partial \perp} + \frac{\partial v_\perp}{\partial \wedge}\right)\right) \\
&+ \wedge_i \wedge_j \left(-\eta_{\|}\left(\frac{\partial v_\perp}{\partial \perp} + \frac{\partial v_\wedge}{\partial \wedge} - \frac{2}{3}\nabla\cdot v\right) + \frac{\eta_\perp}{4}\left(\frac{\partial v_\perp}{\partial \perp} - \frac{\partial v_\wedge}{\partial \wedge}\right) + \frac{\eta_\times}{2}\left(\frac{\partial v_\wedge}{\partial \perp} + \frac{\partial v_\perp}{\partial \wedge}\right)\right) \\
&+ (b_i \perp_j + \perp_i b_j)\left(-\eta_\perp\left(\frac{\partial v_\perp}{\partial b} + \frac{\partial v_b}{\partial \perp}\right) - \eta_\times\left(\frac{\partial v_\wedge}{\partial b} + \frac{\partial v_b}{\partial \wedge}\right)\right) \\
&+ (\wedge_i \perp_j + \wedge_j \perp_i)\left(-\frac{\eta_\perp}{4}\left(\frac{\partial v_\wedge}{\partial \perp} + \frac{\partial v_\perp}{\partial \wedge}\right) + \frac{\eta_\times}{2}\left(\frac{\partial v_\perp}{\partial \perp} - \frac{\partial v_\wedge}{\partial \wedge}\right)\right) \\
&+ (b_i \wedge_j + b_j \wedge_i)\left(-\eta_\perp\left(\frac{\partial v_\wedge}{\partial b} + \frac{\partial v_b}{\partial \wedge}\right) + \eta_\times\left(\frac{\partial v_\perp}{\partial b} + \frac{\partial v_b}{\partial \perp}\right)\right)
\end{aligned}$$

Put into magnetic components and simplify:

$$\begin{aligned}
\Pi_{ij} &= \frac{2}{3}\delta_{ij}\eta_{\|}\nabla\cdot v - \eta_{\|}\left(2b_i b_j\frac{\partial v_b}{\partial b} + (\perp_i\perp_j + \wedge_i\wedge_j)\left(\frac{\partial v_\perp}{\partial \perp} + \frac{\partial v_\wedge}{\partial \wedge}\right)\right) \\
&-\frac{\eta_\perp}{4}(\perp_i\perp_j - \wedge_i\wedge_j)\left(\frac{\partial v_\perp}{\partial \perp} - \frac{\partial v_\wedge}{\partial \wedge}\right) - \frac{\eta_\perp}{4}(\wedge_i \perp_j + \wedge_j \perp_i)\left(\frac{\partial v_\wedge}{\partial \perp} + \frac{\partial v_\perp}{\partial \wedge}\right) \\
&-\eta_\perp(b_i \perp_j + \perp_i b_j)\left(\frac{\partial v_\perp}{\partial b} + \frac{\partial v_b}{\partial \perp}\right) - \eta_\perp(b_i \wedge_j + b_j\wedge_i)\left(\frac{\partial v_\wedge}{\partial b} + \frac{\partial v_b}{\partial \wedge}\right) \\
&-\frac{\eta_\times}{2}(\perp_i\perp_j - \wedge_i\wedge_j)\left(\frac{\partial v_\wedge}{\partial \perp} + \frac{\partial v_\perp}{\partial \wedge}\right) + \frac{\eta_\times}{2}(\wedge_i \perp_j + \wedge_j \perp_i)\left(\frac{\partial v_\perp}{\partial \perp} - \frac{\partial v_\wedge}{\partial \wedge}\right) \\
&-\eta_\times(b_i \perp_j + \perp_i b_j)\left(\frac{\partial v_\wedge}{\partial b} + \frac{\partial v_b}{\partial \wedge}\right) + \eta_\times(b_i \wedge_j + b_j\wedge_i)\left(\frac{\partial v_\perp}{\partial b} + \frac{\partial v_b}{\partial \perp}\right)
\end{aligned} \tag{255}$$

This expression is actually useful even in Route 1 since it takes us from $\frac{\partial v_\perp}{\partial \perp}$ to $\Pi_{(xyz)}$, allowing us to skip Step 2 of Route 1.

What is the coefficient on ik in element ij?

There is definitely more working out worth doing. And we should stick with the transformed perspective I think.

$$\begin{aligned}
c_{ii} &= \frac{2}{3}\delta_{ij}\eta_{\|} - \eta_{\|}\left(2b_i b_j b_i^2 + (\perp_i\perp_j + \wedge_i\wedge_j)\left(\perp_i^2 + \wedge_i^2\right)\right) \\
&-\frac{\eta_\perp}{4}\left((\perp_i\perp_j - \wedge_i\wedge_j)\left(\perp_i^2 - \wedge_i^2\right) + 2(\wedge_i \perp_j + \wedge_j \perp_i)\perp_i \wedge_i\right) \\
&-2\eta_\perp\left((b_i \perp_j + \perp_i b_j) b_i \perp_i + (b_i \wedge_j + b_j\wedge_i) b_i\wedge_i\right) \\
&-\eta_\times(\perp_i\perp_j - \wedge_i\wedge_j)\wedge_i \perp_i + \frac{\eta_\times}{2}(\wedge_i \perp_j + \wedge_j \perp_i)\left(\perp_i^2 - \wedge_i^2\right) \\
&-2\eta_\times(b_i \perp_j + \perp_i b_j) b_i \wedge_i + 2\eta_\times(b_i \wedge_j + b_j\wedge_i) b_i \perp_i
\end{aligned} \tag{256}$$

It is actually worth collecting terms in $\frac{\partial v_\perp}{\partial b}$ since η components will be already known.

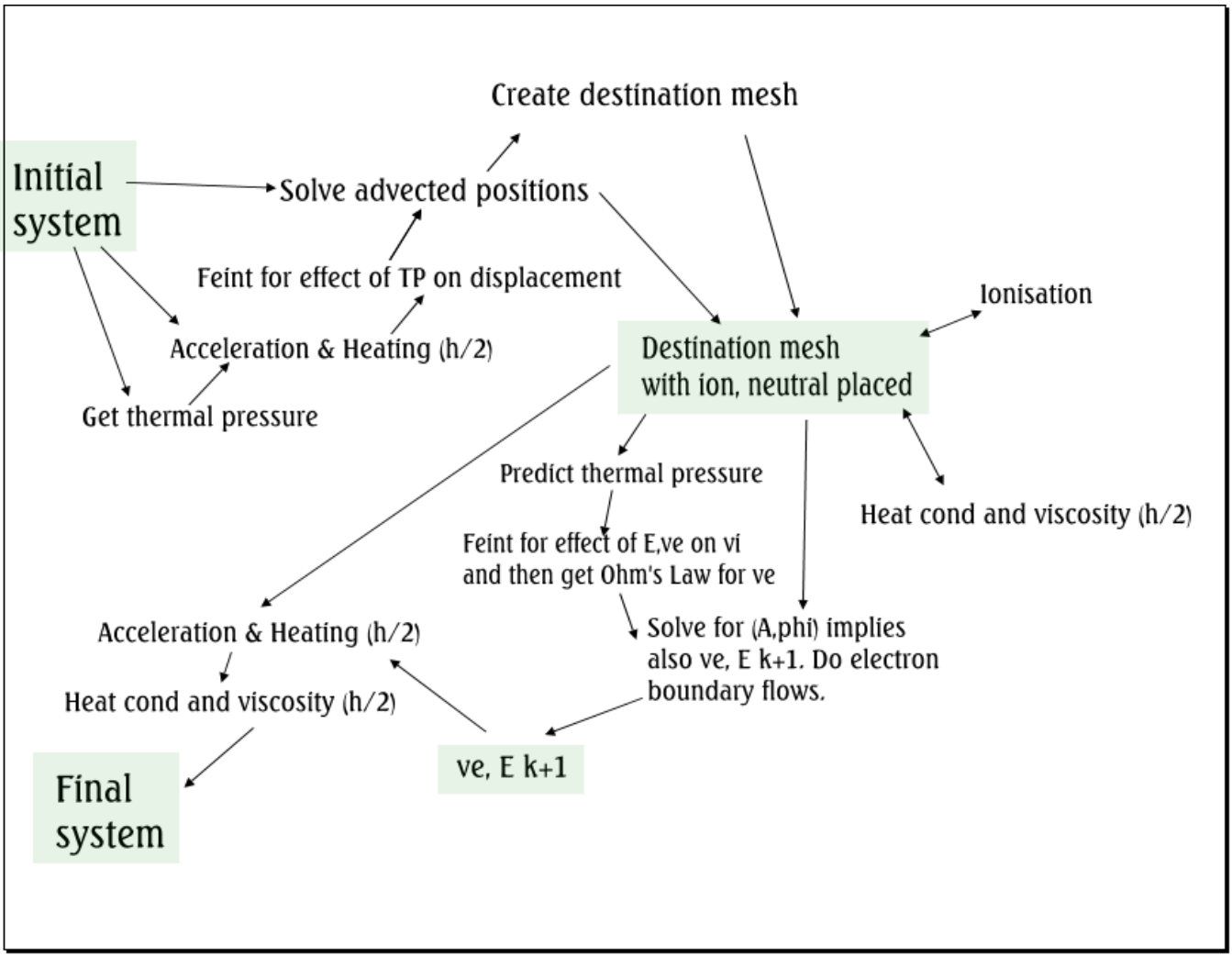
$$
\begin{aligned}
\Pi_{ij} \;=\; & \frac{2}{3}\delta_{ij}\eta_\parallel \nabla\cdot v - \eta_\parallel\left(2b_ib_j\frac{\partial v_b}{\partial b}\right) \\
& + \frac{\partial v_\perp}{\partial \perp}\left(-\eta_\parallel\left(\delta_{ij}-b_ib_j\right) - \frac{\eta_\perp}{4}\left(\perp_i\perp_j - \wedge_i\wedge_j\right) + \frac{\eta_\times}{2}\left(\wedge_i\perp_j + \wedge_j\perp_i\right)\right) \\
& + \frac{\partial v_\wedge}{\partial \wedge}\left(-\eta_\parallel\left(\delta_{ij}-b_ib_j\right) + \frac{\eta_\perp}{4}\left(\perp_i\perp_j - \wedge_i\wedge_j\right) - \frac{\eta_\times}{2}\left(\wedge_i\perp_j + \wedge_j\perp_i\right)\right) \\
& + \left(\frac{\partial v_\perp}{\partial b} + \frac{\partial v_b}{\partial \perp}\right)\left(-\eta_\perp\left(b_i\perp_j + \perp_i b_j\right) + \eta_\times\left(b_i\wedge_j + b_j\wedge_i\right)\right) \\
& + \left(\frac{\partial v_\wedge}{\partial b} + \frac{\partial v_b}{\partial \wedge}\right)\left(-\eta_\perp\left(b_i\wedge_j + b_j\wedge_i\right) - \eta_\times\left(b_i\perp_j + \perp_i b_j\right)\right) \\
& + \left(\frac{\partial v_\wedge}{\partial \perp} + \frac{\partial v_\perp}{\partial \wedge}\right)\left(-\frac{\eta_\perp}{4}\left(\wedge_i\perp_j + \wedge_j\perp_i\right) - \frac{\eta_\times}{2}\left(\perp_i\perp_j - \wedge_i\wedge_j\right)\right)
\end{aligned}
$$

Part II
Numerical methods

5 Structure of a system advance step

Here I shall lay out the sequence for a system advance, and it shall become clear from the later chapters why this general sequence is used.

5.0.2 Sequence of a simulation step



1. Compute momentum addition rates due to thermal pressure at time t_k.

2. Accelerate and perform frictional heating for half a step, in each cell.

3. Perform feint for half a step to establish, in each cell, a linear relation between a^{TP}_{k+1} and position at t_{k+1}. Average on to vertices. ie,

$$\int_{t_k}^{t_{k+1}} v_{ion} = d^0_{ion} + F_{h^2} a^{TP}_{ion,k+1}$$

   (Assume for subcycle that $\frac{m_e}{m_i} a^{TP}_e \approx a^{TP}_i$ in view of $n^{k+1}_e \approx n^{k+1}_i$.)

4. Solve for advected positions for ion and for neutral, taking account of the second order pressure effect.

5. Average vertex positions to create destination mesh M_{k+1} (and interpolate A_k to the new mesh).
6. Place ion and neutral into new mesh, including compressive heating. (Place electron as ion.)
7. Infer B from A_k on mesh M_{k+1}.
8. Run $h/4$ of heat conduction, thermoelectric effect and viscous momentum flows.
9. Perform ionisation and recombination.
10. Run $h/4$ of heat conduction, thermoelectric effect and viscous momentum flows.
11. Compute momentum addition rates due to thermal pressure at time t_{k+1} based on our description so far.
12. Perform feint for half a step to establish a linear relation between ion v_{k+1} and E_{k+1}, v^e_{k+1}. This serves as input in computing a tensor Ohm's Law giving v^e_{k+1} as a linear function of E_{k+1}.

$$v_{e,k+1} = v^{[E=0]}_{e,k+1} + \sigma_e E_{k+1}$$

13. Solve for $A_{k+1}, \phi_{k+1}, J_{k+1}$ and the circuit p.d., per Backward (pre-)Maxwell, ie allowing $J_{k+1} = \sigma\left(-\nabla\phi_{k+1} - \frac{1}{ch}\left(A_{k+1} - A_k\right) + E_{ext}\right)$ and that charge density is formed from a Backward Euler step using this J_{k+1}. E_{ext} is set to attain the prescribed current $I^{prescribed}_{z,k+1}$. (This will also output an E field, which will be used only for the ion half-step next time.)
14. Calculate $B_{k+1} = \nabla \times A_{k+1}$.
15. Compute momentum addition rates due to thermal pressure at time t_{k+1} based on our description so far.
16. Accelerate and apply frictional heating for half a step. (Note that the ion fluid is left in position although the updated ion velocity is based on E_{k+1}.)
17. Run $h/2$ of heat conduction, thermoelectric effect and viscous momentum flows.

Note on computing thermal pressure In particular this involves assuming that we can integrate the gradient of nT over an offset cell and assume that nT is known at the centre of a cell. If nT is a poor estimate then the additional momentum due to pressure is poorly apportioned but there is not a qualitative error.

A bigger concern can be when there is a shock front that is very steep: in this case, whereas the gradient formula considers the case of infinitesimal timestep and information crossing an infinitesimal distance, in a finite time h the physically correct pressure is spread over a finite radius, with SD similar to $\sqrt{h^2T/m}$: the same as in a collisionless case because collisions do not impede the propagation of momentum due to random velocities. Half the additional momentum for the fluid comes from electrons so this can be quite a large distance compared with a cell width. The crucial thing is that h^2 appears, not h. So e.g. $h = 10^{-10}, T = 10^{-11}$ then $\sqrt{h^2T/m} = 10^{-2}$cm. This could be more than the whole width of a shock front

In this case we would be better off using an integral technique to infer where pressure is felt. But this has not been successful so far, so we are stuck with the integral of ∇nT over a small volume instead.

6 Advection (and compressive heating)

Properties we desire:

- Lagrangian
  - because we are told eventually the plasma will inhabit a small area and high resolution will be needed there. (Multiresolution in fixed cells is harder / less elegant; fine resolution everywhere is inefficient.)
  - because it is the easiest way to follow high density fronts without artificial phenomena emerging. Including but not limited to, the dissipation that happens under Eulerian moves.
- Conservative of mass and momentum. Heat conserved, given compressive heating.
  - necessary for any hope of qualitatively correct treatment where there are shock fronts or unsmooth phenomena
- No spatial derivatives estimated.
- Most elegant if we can avoid CFL-incurring assumptions (e.g. FV with boundary flows assumes that only a small proportion of fluid exits the cell). We might eventually want to have cell widths less than 1 micron and speeds of 1e6 cm/s so that would imply a timestep of at most, 1e-12 s, perhaps less. *
- Compressive heating done robustly and simply.
- Multi-species flow relative to one bulk mesh
  - one overall mesh can be used for the inter-species processes, to avoid over-complexity.
  - avoid assuming everywhere that the ion and neutral velocities are always equal.
- Need to avoid numerical instability when there is a region being compressed and giving artificial oscillations under a naive method.
- Willing to assume continuous deformation of each species.
  - This one assumption is very useful. Continuous velocity is not unreasonable since there is viscosity. Conversely, allowing overlaps between advected objects, would make robust compressive heating non-trivial for a start.
- Conceptually simple scheme.
- Allow for an arbitrary change of the mesh, preferably at the same time as the step.

* Note that I did not yet come up with a way of avoiding using boundary flows for electron current. This 'CFL' is there far less important on the whole, because we may hope that the profile of n_e must end up very close to that of n_{ion}. (But it's certainly still far from perfect.)

6.1 The advection procedure

Let the existing mesh, on which we have mass, momentum and heat defined for all species on all cells, be called M_k. We can describe a general integral-based approach to advection-compression as follows:

1. **Advect species:** For each heavy species, decide how to move the vertices, continuously deforming the fluid, to create advected species meshes $A_{ion}(M_k)$ and $A_{neutral}(M_k)$ which are formed with the same set of triangles but advected vertices. These meshes are unchanged from M_k except that the vertices are shifted.

2. **Apply adiabatic heating** to calculate the new heat in $A_{ion}(M_k)$ and $A_{neutral}(M_k)$ based on cell area change, since the cell can be treated as an adiabat and recover the compressive heating term of dT/dt in the limit $h \to 0$.

3. **Create new mesh:** Average the vertex positions from $A_{ion}(M_k)$ and $A_{neutral}(M_k)$ to create the new bulk mesh M_{k+1}, weighting by species density at each vertex. Perform any desired modifications to M_{k+1} including re-Delaunerization and destroying any overlaps (usually there are none), preventing cells getting beyond a maximum size, etc. Wrap the vertices across periodic boundaries. (Of course, to get an Eulerian simulation we just take $M_{k+1} = M_k$.)

4. **Placement:** Place $A_{ion}(M_k)$ on to M_{k+1}. To do this apply an intra-cell model of $(n, v, T)_{k+1}$ on $A_{ion}(M_k)$ – the model must fit the mass, momentum and heat in each cell – and integrate this model over all intersections of $A_{ion}(M_k)$ cells with M_{k+1} cells, to create (mass, momentum, heat) on the cells of M_{k+1}.

Second order pressure effect prevents overcompression / instability: Once a move takes the system beyond the equilibrium where thermal pressure is reversed, and if the timestep is too long for one location, then again it can get into a negative loop that magnifies oscillations.

Therefore an important detail is that in order to perform the advection step 1, we do not just use the existing velocity v_k to perform the advection. Instead we have to take account of the nonlinearly changing thermal pressure when advection is performed. We seek to anticipate the change in heat in a species cell due to the heavy species move, and this informs a position solver that takes account of acceleration at second-order. (In fact, we use Heun: the first half of the step uses the t_k pressure and the second half the t_{k+1} pressure.) The adiabatic heating of a cell can be computed from its change in area. This is a nonlinear solve with a global adaptive timestep. We can do all this because the vertex move represents a continuous deformation of the species fluid.

We use only the thermal pressure for the advecting species, for this second-order effect of acceleration on displacement. This is for simplicity and not for any particularly sound reason. But the main point of the position solver is to produce robustness, not accuracy.

Alternative: Hydrodynamic equivalent It is possible that we didn't need to allow for relative flows of species, but I did not want to assume this to begin with. It is easy to envisage a hydrodynamic version where we take into account overall thermal pressure in advecting vertices per the bulk mesh; all species are in that case advected with the bulk velocity. We still do then need an intra-cell model in case that there are Delaunay flips going on, but a lot of computer work is skipped since most cells just retain their contents.

6.2 The species vertex advection

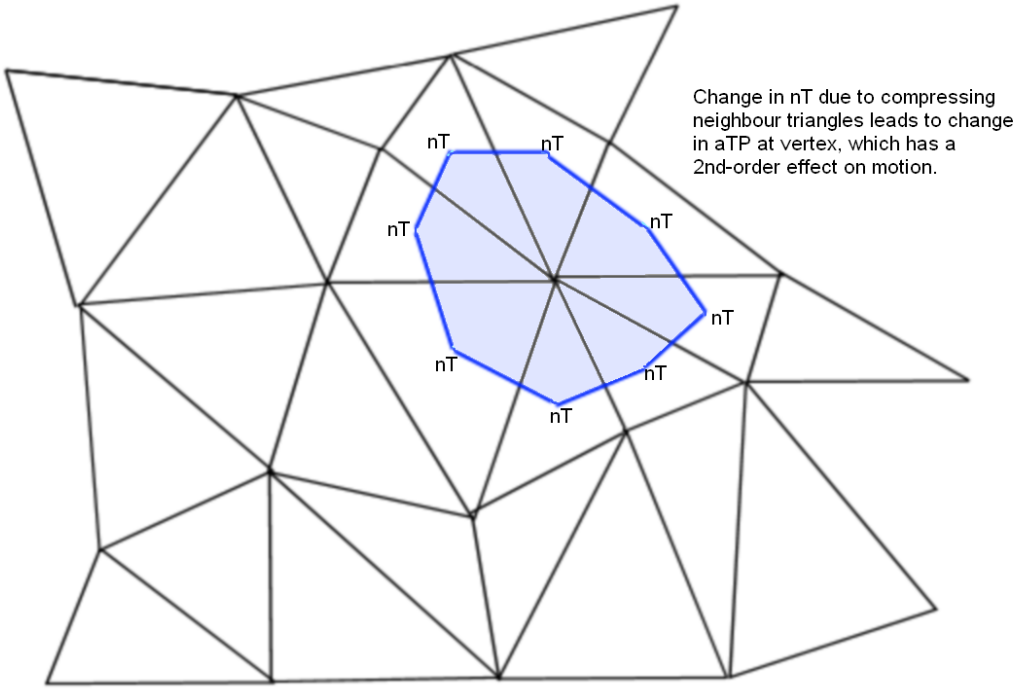
Thinking of ions and neutrals,

1. It is nice to be certain that nothing gets crushed, which comes naturally out of solving for the new species vertex positions.

2. Avoiding overlaps means we can do adiabatic compressive heating legitimately on the advected mesh, which is well-tried - there are many other ways to get compressive heating which do not work well.

3. Defining pressure on vertices means using the fundamental nT data if we defined mass, momentum on triangles. It is the same way as we have been calculating it to get additional momentum. It's good to do this same way for the t_{k+1} pressure contribution to displacement.

Therefore the most logical thing is to solve for the new vertex positions. We make some assumptions about electrons and find a solution where the displacement is consistent with the $(nT)_{k+1}$ obtained and its accelerating effect. We will need to work with the actual force within the Voronoi cell because the cell edges may be deformed as part of the move.



- We predetermine a formula that relates the thermal pressure force (or momentum addition rate) at time t_{k+1} to the displacement $\int_{t_k}^{t_{k+1}} v\ dt$ at the vertex. This is a mass-weighted average of the linear formulas that we get for the cells.

$$d = d_0 + F_{2\times 2} a_{k+1}^{TP}$$

- This leaves us with nonlinear equations to solve since $(nT)_{cell}$ is nonlinearly related to the area of the cell.

- We do not include the pressure from the other species but see notes in algorithm below. We could sensibly take e_{k+1} pressure to be distributed based on ion $k+1$ pressure - remember that to take $n_e \approx n_i$ is a very accurate approximation wherever it matters. We could assume that as we move ion vertices, T_e will ratio in the same way as T_i in the same location: density changing at the same rate in the same place.

- We might well want to allow that although we use E_k initially to get a 'draft' ion motion, we can do a further small ion move due to finding E_{k+1}. However we have to be aware that for electrons, E_{k+1} and a_{k+1}^{eTP} are countervailing large forces.

It is notable that if we connect centroids and medians, we obtain polygons that always have 1/3 of the area of their cell clusters. For now, we ascribe them 1/3 of the mass. I will refer to these as

quasi-Voronoi (QV) polygons. In doing integrals around the edge, we still make the centroids the only nodes. (We could and probably should improve that.)

The algorithm implemented is based on regarding the solution as the equilibrium of the system

$$\frac{dx_{k+1}}{dt} = -x_{k+1} + x_k + d_h^0 + F_{h^2} a_{k+1}^{TP} \tag{257}$$

where d_h^0 is the pre-calculated displacement, absent a_{k+1}^{TP}, and F_{h^2} is a pre-calculated matrix of coefficients.

6.2.1 Discussion - Implicit vertex advection vs. unilateral cell trajectories

It was found (a long time previously) that just moving vertices explicitly, without guarding against over-compression, does not work. Something will get crushed against the boundary.

My previous way of guarding against over-compression implied double-counting the pressure effect on motion (2nd order but not small). I wrote out a method to map each cell trajectory unilaterally, allowing that they then overlap. That is a bit unsatisfactory in that it is treating the external forces on a cell as fixed when deciding on its trajectory. That is a way that tries to get robustness while introducing a second-order error, whereas we would ideally prefer to get robustness by taking proper account of the second-order effect of thermal pressure on displacement.

So as mentioned, I now implemented a Lagrangian implicit method for the vertex motion, seeking to allow (more-or-less trapezoidally) for the thermal pressure momentum addition rate supplied by the new (t_{k+1}) profile of nT. It is taken into account in determining the displacement $\int_{t_k}^{t_{k+1}} v\ dt$ for each ion and neutral vertex. However, we do not use prespecified coefficients for this scheme: because we want to respect the shorter timescale for evolving v, we determine at runtime the effect that a_{k+1} should have on $\int_{t_k}^{t_{k+1}} v\ dt$. Given experience with E and ρ, it may well turn out that this method for heavy species needs to be modified to use Backward Euler also.

In another respect, this solve is rather different from the electron situation. Since we are going to do a Lagrangian ion move, it makes sense to work with that and take acceleration on vertices. The alternative is messy: predict nT on Voronoi cells due to anticipated Eulerian flows through Voronoi boundaries and solve for $\int_{t_k}^{t_{k+1}} v$ in cells. Matching the actual move seems better.

The idea of allowing overlaps of advected cells did have a lot of intuitive appeal. However it is not a priori necessary in order to simulate the model. We do not have a situation like the inviscid Berger's equation because for us, pressure rapidly cancels the velocity that would let one parcel catch up with another. We can still allow that mass can flow between cells due to hypo-diffusion, by which I mean the effects of the random velocity distribution. Assuming instant restoration of a Maxwellian distribution implies choosing not to retain multiple values for average velocity, and when velocity is a continuous function, it makes sense to allow that each vertex position has a defined velocity and that, apart from random motion, fluid within a triangle of vertices stays within it.

The downfall of this non-overlap method may be that solving for the implicit positions can become too difficult. So far, the method used features an adaptive global timestep; it is easy for the method to fail by this timestep becoming indefinitely short. So far it is coping fine.

6.2.2 The position solver

The position solver has the following steps:

1. Add up species mass in each QV polygon.
2. Loop:
   (a) Get putative areas and so, nT for each cell. This involves a call to *pow(·,2/3)*, for which we might well wish to write a speed-up function.

(b) Calculate momentum addition rate in each QV polygon,

$$\frac{dp}{dt} = \begin{pmatrix} -\sum_{cell\ i} \left(cc_2^{(y)} - cc_1^{(y)}\right) \frac{1}{2}\left((nT)_i + (nT)_{i+1}\right) \\ -\sum_{cell\ i} \left(cc_1^{(x)} - cc_2^{(x)}\right) \frac{1}{2}\left((nT)_i + (nT)_{i+1}\right) \end{pmatrix} \tag{258}$$

where $i+1$ indicates the more anticlockwise cell than i, surrounding the vertex. For edge cells, the first and last edges terminate at the median of the cell. Divide by mass to give acceleration rate a_{k+1}^{TP} which is taken to apply at the vertex. Consequently, calculate $\frac{dx}{dt}$ as per the above.

$$\frac{dx_{k+1}}{dt} = -x_{k+1} + x_k + d_h^0 + F_{h^2} a_{k+1}^{TP}$$

(For inner/outer edge vertices, get rid of any radial component of $\frac{dx}{dt}$.)

(c) Now to make a second-order accurate step, we find d^2x/dt^2.

i. Calculate the rate of change of area for each cell, and therefore take for each cell,

$$\frac{d}{dt}(nT) = -\frac{5}{3}\frac{dArea}{dt}\frac{nT}{Area}$$

ii. For each QV cell, take the rate of change of the momentum addition rate to be:

$$\begin{pmatrix} -\sum_{cell\ i} \left(\frac{d}{dt}cc_{i+1}^{(y)} - \frac{d}{dt}cc_i^{(y)}\right) \frac{1}{2}\left((nT)_i + (nT)_{i+1}\right) + \left(cc_{i+1}^{(y)} - cc_i^{(y)}\right) \frac{1}{2}\left(\frac{d}{dt}(nT)_i + \frac{d}{dt}(nT)_{i+1}\right) \\ -\sum_{cell\ i} \left(\frac{d}{dt}cc_i^{(x)} - \frac{d}{dt}cc_{i+1}^{(x)}\right) \frac{1}{2}\left((nT)_i + (nT)_{i+1}\right) + \left(cc_i^{(x)} - cc_{i+1}^{(x)}\right) \frac{1}{2}\left(\frac{d}{dt}(nT)_i + \frac{d}{dt}(nT)_{i+1}\right) \end{pmatrix} \tag{259}$$

where $i+1$ indicates the more anticlockwise cell than i, surrounding the vertex. For first and last edges of edge cells, we have to do some extra work which involves a lot of predictable details.

iii. Calculate

$$\frac{d}{dt}a_{k+1}^{TP} = \frac{d^2p}{dt^2}/mass_{QV} \tag{260}$$

$$\frac{d^2x}{dt^2} = -\frac{dx}{dt} + F_{h^2}\frac{d}{dt}a_{k+1}^{TP} \tag{261}$$

iv. Now set putative coordinates:

$$x_{k+1}^{putative} = x_{k+1} + h_{traj}\frac{dx}{dt} + \frac{1}{2}h_{traj}^2\frac{d^2x}{dt^2} \tag{262}$$

If we are an inner/outer edge vertex, project to the corresponding radius.

v. We now proceed to a loop to adjust the global trajectory timestep h_{traj} until we are satisfied things are not moving too quickly.

A. Take the shoelace formula for the area of each putative triangle. If the sign of the formula has changed compared to the beginning of this trajectory step, then the triangle has flipped; try again, multiplying h_{traj} by 0.2. If the area has decreased to < 0.4 of its previous value, reduce h_{traj} to try to linearly attain just that reduction.

B. If a triangle was deficient, linearly interpolate for the reduced h_{traj} to get a new $x_{k+1}^{putative}$ and go again.

C. Otherwise, if nothing is deficient, try increasing h_{traj} by factor 1.6, until it is near its maximum.

D. Moreover, if nothing is deficient we can test for convergence. We require two conditions: (i) for each vertex, in view that $h_{traj} = 1$ is the trajectory time length to apparently reach our implied $x_{k+1} = x_{k+1}^{putative} + \frac{dx}{dt}$,

$$\text{compare} = \max\left(\left\|\frac{dx}{dt}\right\|, \left\|\frac{dx}{dt} + \frac{d^2x}{dt^2}\right\|\right) \tag{263}$$

and the criterion is

$$\text{compare} < \text{threshold} + 0.001 \left\|x_{k+1}^{putative} - x_k\right\| \tag{264}$$

(ii) For each triangle, we want to test

$$\left\|\frac{dArea}{dt}/Area\right\| = \left\|\frac{d\left(nT\right)}{dt}/\left(\frac{5}{3}nT\right)\right\| < 0.01 \tag{265}$$

3. Regarding the periodic boundary, the only decent way is to leave the advected points unwrapped, so that cell periodicity information does not have to be refreshed. This means we have to deal with wrapping them afterwards, perhaps after we take the average in the case of creating the new bulk mesh.

Note:

Multimesh would allow us to really move the mesh even for Jacobi, when it otherwise could get stuck not moving due to the pressure of the other points. It is meant to all be translated at once. It is a fact that moving a point leftward propagates such an error that another point would have to move leftward, so we can try some principle that vector error is annihilated on large chunks.

Is multimesh rather hard in that we have to still abide by the constraint of staying inside the neighbour polygon? It's OK if we distort disjoint chunks so that what is in a chunk cannot end up being moved into another chunk. Moving coarse vertices pulls the fine vertices in the neighbour coarse cells. This cannot cause bad shearing or crashing.

Note 2 (not used) :

Given an hypothetical guess of displacement d, we can infer the implied displacement $d_*\left(d\right)$ and try moving some distance in that direction, calculating $\frac{d}{d\lambda}$ and perhaps $\frac{d^2}{d\lambda^2}$. We don't want to cross the implied position in the dimension that we are moving in, note bene. We should generally try to chase the implied position.

6.3 Placement

In order to perform placement there are basically two tasks.

1. Create an intra-cell model of n, T, v. (It would be most desirable to choose T, v since they are directly subject to smoothing, but so far we use nT, nv.)

2. Identify intersections between each cell being placed, and the cells of the destination mesh, and proceed to integrate the modelled n, nT, nv to accumulate mass, momentum and heat in the destination cells.

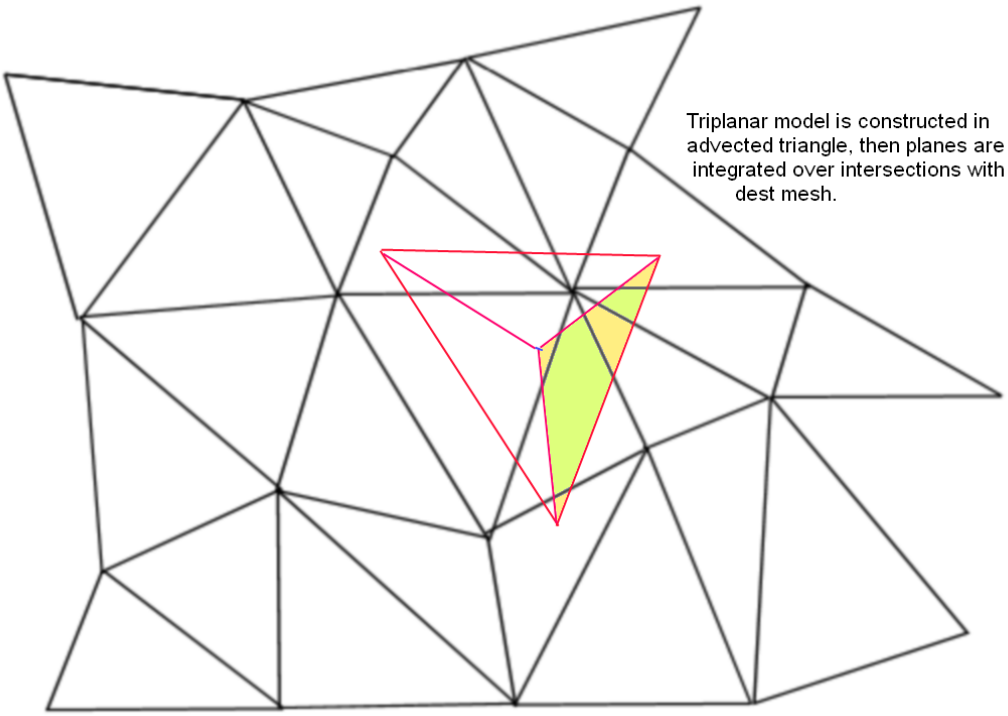
Here I am basically only going to document task 1, which is the less straightforward topic.

Task 2 is based on a search algorithm which

- starts from a triangle seed inferred from searching for the dest cell containing the centroid of the advected cell.

- work contiguously, keeping a flag on dest cells to indicate whether they have been searched, and a list of unsearched cells generated by adding neighbours of all cells in which an intersection is found, iff their flag is off.

- when we exhaust the list of unsearched candidate cells, we have searched a contiguous region edged by dest cells with no intersection.
- the intersections can be any polygonal shape, so a routine is then needed to integrate a plane (which will be from a subset of the advected cell) over any polygon.



6.3.1 Discussion - Placement and integration vs. boundary flows

Now before we describe these steps in more detail, let us consider the advantages and disadvantages relative to computing boundary flows, within a conservative finite volume context. Here we have no time-derivative used, but instead conceptually, an integral-based method where we find the new mass in a cell T by integrating over the region $A^{-1}(T)$ that maps into T, similar to just advecting particles. But if we could instead assume that $A_{ion}(M_k)$ and M_{k+1} will be very close, so that we are able to just estimate the boundary flux of (mass, momentum, heat) considering $v_{ion} - v_{bulk}$.

- This way is more robust: we cannot end up with negative mass due to outflows.
- There is not a CFL speed limit for advection. We can drift one mesh by some large distance relative to the minimum inter-vertex spacing, and place it on to the other mesh no problem. Even a spike of velocity is survivable, at least as long as it stays within the domain.
- We need to avoid assuming that $A_{ion}(M_k)$ and M_{k+1} will be very close, if we want to apply Delaunay flips etc.
- Our way, we do not need to assume that only direct neighbours (with shared walls) receive mass.
- It is much slower to do our way. Computing intersections and integrals takes time. In practice, for us this is not important since compared to solving fields and currents, it is a short amount of time. But if we cared, we could detect the right conditions for using boundary flows.
- Eulerian flows are not so elegant as regards adiabatic heating.

6.3.2 Triplanar modelling

To use a triplanar model to describe n, nT and nv, we split a triangle cell into three sub-triangles joined at the cell centre ($\frac{x_0+x_1+x_2}{3}$) and in each of these we assume that the variables shall be linear. For simplicity let us treat n. The central value is set to get the mass. The average of a planar variable over a triangle is the average of its values at the corners; so, the average n for a triangle with planar n and corner values n_0, n_1, n_2 is $n_{avg} = \frac{1}{3}(n_0 + n_1 + n_2)$. Therefore if we choose a central value n_C then the mass being represented for our cell is

$$\frac{Area}{3}\left(\frac{n_1+n_2+n_C}{3} + \frac{n_1+n_0+n_C}{3} + \frac{n_2+n_0+n_C}{3}\right) = Area\left(\frac{2n_0+2n_1+2n_2+3n_C}{9}\right)$$

Therefore to achieve the correct mass,

$$n_C = 3\frac{mass}{Area} - \frac{2}{3}(n_0 + n_1 + n_2) \tag{266}$$

Note that polygon areas are found expediently using the Gauss shoelace formula.

Now we apply a criterion to rule out applying this triplanar model if the situation is not suitable.

We generally reject a fitting if the central value is much outside the highest and lowest of the corners, ie, require $\max(n_0, n_1, n_2) > n_C > \min(n_0, n_1, n_2)$, although we may allow a small amount of leeway if n_C is within e.g. $(0.95n_{avg}, 1.05n_{avg})$. If a fitting is not accepted then the default is to use a flat model of n (cf the minmod slope limiter). In practice this is handled by creating some temporary vertices in memory and populating all with n_{avg}.

Whether n's model failed or succeeded, we then take a triplanar model of nv. We use the same formula

$$(nv)_C = 3\frac{\int_{cell} nv}{Area} - \frac{2}{3}(n_0v_0 + n_1v_1 + n_2v_2)$$

and similar criteria for failure. Likewise nT, to partition heat into the sub-triangles.

We then have 3 sub-triangles with assigned mass, momentum, and heat, and vertex values for n, v, T.

Triplanar improvement Decided to stick with triplanar models for placement, but improve it a bit: we want it to feel more like minmod, and the principles for setting the corner values should be:

- Attain total mass.
- For adjacent corners, $n_1^{desired} > n_2^{desired} \Rightarrow n_1 \geq n_2$.
- Unless $n_{cell} > n_{corners}$, the maximum/minimum value for n_C is the same as an attained corner. $n_C \in \left(n_{below}^{aimed}, n_{above}^{aimed}\right)$ by the previous principle and if n_C attains n_{above}^{aimed} then so should n_{above}.
- The max distance from cell n, for the corner n, is the distance indicated by n^{aimed}. It cannot move further away from n_{cell}.
- Until a corner can be attained, we give the same slope in two directions.

So we no longer default to flat, except in the case that n_{corner} is more than (or less than) n_{cell} for all corners.

We don't try to introduce medians - it's too complicated for right now. This much is an improvement.

The typical situation should (still) be that the corners are all attained and $n_C \in \left(n_{below}^{aimed}, n_{above}^{aimed}\right)$.

The algorithm:

1. If $n_{cell} < min\left(n_{1,2,3}\right)$ or $n_{cell} > max\left(n_{1,2,3}\right)$ default to uniform $n_C = n_{1,2,3}^{attain} = n_{cell}$, a la minmod.

2. Assess n_C implied by n_{cell} and $n_{1,2,3}$.

$$n_C^{implied} = 3\left(n_{cell} - \frac{2}{9}\left(n_1 + n_2 + n_3\right)\right) \tag{267}$$

   We require that n_C inhabit the same interval between corners as n_{cell}. If $n_C^{implied}$ does, then we accept it and all $n_{1,2,3}$ are attained.

3. Otherwise take n_C to the min or max of that interval. We then see which corners can be attained.
   WLOG, $n_1 < n_2 < n_3$ so let us consider

   (a) If $n_2 < n_{cell} < n_3$ and $n_C^{implied}$ is too low (because n_3 is very large). Then we would take $n_C = n_2$ and even attaining n_1, we shall not be able to attain n_3.

   (b) If $n_1 < n_{cell} < n_2$ and $n_C^{implied}$ is too low. Then we take $n_C = n_1$ and test what we attain with equal values of $n_2^{attain}, n_3^{attain}$. This will show whether n_2 can be attained or not.

   (c) Likewise for $n_C^{implied}$ too high.

- For a triangle with 1 or 2 edge vertices it is a bit different. The problem seems best approached by correcting the estimates of variables at edge vertices to reflect that, given the inner vertex estimates of n, we try to achieve a total mass: if we have three triangles adjoining an insulator vertex, corners n_a, n_b, then go with

$$n_{ins\ vert} = \frac{\frac{2}{3}m_1 + m_2 + \frac{2}{3}m_3 - \frac{1}{3}\left(Area_1 + Area_2\right)n_a - \frac{1}{3}\left(Area_2 + Area_3\right)n_b}{\frac{1}{3}Area_1 + \frac{1}{3}Area_2 + \frac{1}{3}Area_3} \tag{268}$$

- Same for nT, given that n succeeds. For nv it is dimensionwise.

Note: In the case of using vertex-centred cells, the logical thing is to chunk the cell into triangles that extend from the centre to the edge. In this case it would be more pleasant in that the planes are then closer to Delaunay triangles.

7 Implicit fields and current

Introduction In this chapter, I describe the implicit methods for current and field evolution which I shall call Backward Gauss and Backward Ampere-Faraday. Here we assume that either a tensor Ohm's Law applies - which may or may not be one that allows for viscous terms - or that at any rate we have estimated a linear relationship between E_{k+1} and the electron displacement $\int_{t_k}^{t_{k+1}} v_e dt$.

The idea of implicit field calculations has some history in plasma simulation; one such code is the laser fusion code at Imperial. But "implicit fields" often refers to implicit Maxwell evolution - so for example, charge density does not enter into it. That sets up different challenges.

7.0.3 Justify applying differential operators without expecting problems

We could make some discretisation of the differential operators and hence form a set of linear equations. However, ordinarily in this project we expect nothing but grief by working from differential operators, mostly because they are applied to variables for which spatial derivatives cannot be estimated. The data will contain, if not truly non-smooth features, at least apparent numerical cliffs.

As mentioned earlier, recall the connection between Poisson and the equilibrium of a birth-diffusion system. This may justify a differential-based approach, since birth-diffusion must result in a profile of A that is heavily smoothed compared to J, so in particular it is not unreasonable to apply a discretised Laplacian for $\nabla^2 A$ – we can also regard it in terms of diffusive averaging. The main difference between applying birth-diffusion steps and applying relaxation (e.g. Jacobi's method) is that for the latter we alter the timestep at each vertex to the step that has a zero coefficient on the existing A value.

7.1 Backward Gauss

7.1.1 Electrostatic / "plasma oscillations" instability

Given some relative acceleration of electrons during a step, there will be some equilibrium where the amount of charge density created is consistent with the Coulombic (or "electrostatic") field pulling them back. If that equilibrium is overshot numerically, then that deviation from the correct charge density will create a greater restoring force. If we keep applying Gauss's Law to infer the potential gradient E_{ES}, increasing oscillations may well result. The problem is not one of a physically unstable system, but one with an inherent negative feedback that happens to get larger with n. Hence for a dense plasma, the problem of simulating in the presence of Gauss's Law becomes particularly acute.

The most elegant solution to this situation, since there is a large negative feedback in the system, is to use an implicit approach which I shall call Backward Gauss:

> The accelerating E field is only the one that is consistent with the resulting charge density.

1. Assume for advection that $\int_{t_k}^{t_{k+1}} v_e dt = h v^e_{k+1}$.
2. Assume that v^e_{k+1} is implied by E_{k+1} and not E_k.

This means that Gauss's Law is to hold at the end of the timestep once the move is completed, either applying acceleration already (or taking Ohm's Law for current) from the potential gradient so obtained. In the case of no Ohm's Law, this is the same thing as Backward Euler for the dynamic system (ρ, J, ϕ).

$$\nabla \cdot E_{k+1} = 4\pi \rho_{k+1}$$

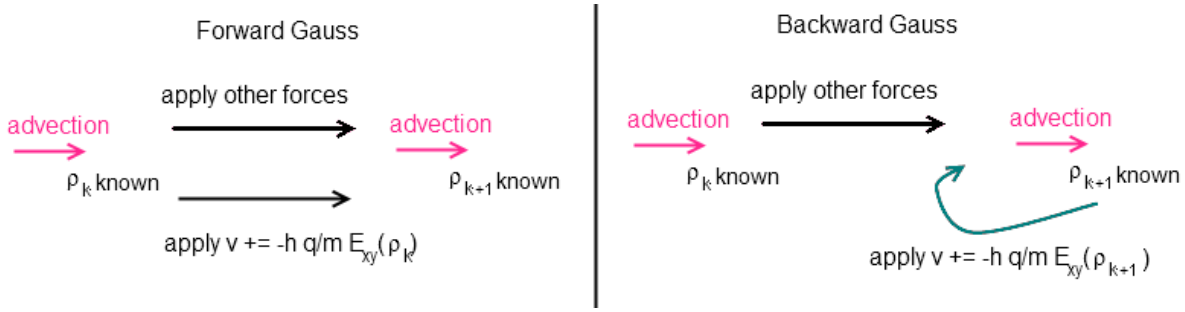
Even with the most viable geometric configuration for discretising the ODEs, this Backward Gauss creates a need for solving large-scale linear equations, which are numerically challenging because of the vast disparity in conductivities in different places. The method for doing that is discussed in a later chapter.

7.1.2 Backward Gauss vs Forward Gauss

[Conclude that Fwd Gauss can't work but note that Maxwell evolution could end up competitive.]

If E_{xy} is the electric field allowed to accelerate the system before the step from t_k to t_{k+1}, it will have a second-order effect on the displacement during that step. Since we assume our electrostatic E field is determined instantaneously by Coulombic forces,

The former is easier to calculate - in essence it requires a solution to Poisson - since ρ_k is independent of this E. But, Backward Gauss is a lot more robust. The difference can be seen in terms that if we forget about the rest of the model, we perform either a Backward Euler or Forward Euler step for the densities and velocities. Suppose that $A(v)$ is an advection operator given velocity field v; then our Backward Euler step consists in taking (e.g.) $v_{e,k+1} = v_{e,k} - h\frac{q}{m}E_{k+1}$ and $n_{e,k+1} = A(v_{e,k+1})\, n_{e,k}$. In practice since we do not combine other processes with advection, the difference is as shown in this picture:



In our initial setup, the equilibrium charge density that will exist is order of magnitude 10^8 charges/cm^3. On a step of length $h = 10^{-14}$, a putative charge density exceeding 10^9 can be observed, before any E_{ES} is applied. So it is not unreasonable to think that Forward Gauss should require a timestep at most 10^{-14} and possibly much smaller. Since we perform shock-proof advection with polygon clipping and it takes time to update the E field, having more than 10^6 steps doesn't seem a great prospect, on any hardware - so we would be limiting our endeavours to the hundreds of nanoseconds.

Post-advection snapback will never work; Gauss computations have to be on the destination mesh.

Once this involved a post-advection snapback, but that way never worked stably in practice. The stability is only found when we ensure that the relative advection applied to electrons actually is what the solver is predicting. Therefore the place to do the E_{k+1} solve is the destination bulk mesh. We want to end up with it there. For a long time I tried to perform a pre-advection acceleration that would bring about Gauss's Law at the end of the advection step. This proved to be infeasible. It is just a fact that the advection procedure, which includes a second-order allowance for anti-compression effects, is not predictable by a simple Eulerian set of boundary flows.

The aims are to get Gauss's Law with a ρ field found on the final mesh and to also apply an acceleration due to $E_{k+1,cell}$ that doesn't send us off at a tangent. We advect first, electrons with ions, calculate E_{ES} and retrospectively apply the second-order displacement that follows from it. This means we can apply the actual boundary flows that we predict during the solver, so it is certain that we achieve the ρ_{k+1} that the solver expects. We have to reapply compressive heating for electrons when we re-flow them.

Boundary flow move for electrons Unfortunately, for it to be perfectly predicted what ρ will be generated, the electron move has to be done with just boundary flows, as mentioned.

- The fundamental problem for Lagrangian electron vertex moves is that any E solver is going to require many iterations of placing meshes on to each other (polygon clipping and integrating planar n over polygons) to work out ρ. It would be possible to put the placement on to GPU and make some gains that way but for now the placement procedure has to be assumed to be

costly. Even to work out $\frac{\partial \rho}{\partial \phi}$ is probably easiest done by trying a small change and measuring, so requires some polygon clipping.

- Meanwhile the big disadvantage of Eulerian current flow is that we cannot avoid having a CFL condition, and in 3D this might start to hurt, as the fast current is in the vertical direction. Though as noted, since the current is nearly divergence free, CFL would not matter so much if we sought only to deal with the density itself: information is effectively propagated since density has to all flow out the other side when it enters a cell. However, we also have to transport electron heat, and it seems CFL is a disadvantage we cannot avoid. The only solution is to switch over to a Lagrangian view where we move electron and ion meshes over each other - as noted, something which seems daunting.

- Including electron thermal pressure at t_{k+1} does not seem like it should be numerically essential: since charge density ρ is small, it seems reasonable to hope that electron fluid will not become over-compressed, as we guarded against that already for ions.

7.2 Backward Ampere-Faraday

Inductive instability: This is the one that used to kill us for a long time with the 1 1/2 D simulation. As with the others this is a negative feedback loop which magnifies oscillations, here caused by the dependence of J on E and the dependence of E via Faraday on the change in B. The solution that I worked out at that time, was restricted to 1D. As with Gauss, the dense conditions make the instability far more acute.

In recent years, I invented a way to cope in the 2D/3D case : the backward Ampere-Faraday equations, requiring us to solve for A, J and inductive E so that Ampere, Faraday and an Ohmic relation are all satisfied at time t_{k+1}. During 2015 we saw this come to fruition.

====

>>>Finish this section.

7.3 Backward Darwin for tensor Ohm case

Suppose we have a simple set of reduced Ohm's Laws:

$$v_{k+1}^{ion} = v_0^{ion} + \sigma_{ion} E_{k+1}$$

$$v_{k+1}^{e} = v_0^{e} + \sigma_{e} E_{k+1}$$

Write

$$J_0 = e\left(v_0^{ion} - v_0^{e}\right)$$

$$\sigma_J = e\left(n_{k+1}^{ion}\sigma_{ion} - n^{e}\sigma_{e}\right) \tag{269}$$

to get the J that we shall use in Ampere-Darwin:

$$J_{k+1}^{Use} = J_0 + \sigma_J E_{k+1} \tag{270}$$

We cannot necessarily use n_{k+1}^{e} - but it should not change much since $n_e \approx n_{ion}$ in any case. n_{ion} is known when we perform the solve because we do not re-flow ions; E only affects their acceleration for the remainder of the step.

Inside the insulator, J_0 is set to represent the return current, or zero where there is no overlap with the return current ring. The reverse current is set to total $-I_z^{presc}$ throughout the solve.

Then we are going to assume that since n_{k+1}^{ion} is given – see remark about having to use electron boundary flows –

$$\rho_{k+1} = e\left(n_{k+1}^{ion} - n_k^{e} + h\nabla\cdot\left(n_e v_{k+1}^{e}\right)\right) \tag{271}$$

and this will become an enforced fact.We are going to assume that

$$E_{k+1} = -\nabla\phi_{k+1} - \frac{1}{c}\left(\frac{A_{k+1} - A_k}{h}\right) + E_z^{ext} \tag{272}$$

This E is obviously deficient since we are using a lagged $\frac{\partial A}{\partial t}$ estimate (cf [BL91]). However, for now it is what was implemented.

Therefore we shall have Backward Gauss (cf (18)):

Note we can assume temporarily that $-\nabla \cdot E^{ext} = 0$;

$$\nabla^2\phi_{k+1} + \nabla\cdot\left(\frac{1}{c}\left(\frac{A_{k+1} - A_k}{h}\right)\right) - \nabla\cdot E^{ext} + 4\pi e\left(n_{k+1}^{ion} - n_k^e + h\nabla\cdot\left(n_e v_{k+1}^e\right)\right) = 0 \tag{273}$$

$$v_{k+1}^e = v_e^0 + \sigma_e E = v_e^0 + \sigma_e\left(-\nabla\phi_{k+1} - \frac{1}{c}\left(\frac{A_{k+1} - A_k}{h}\right) + E_z^{ext}\right)$$

Expand:

$$\begin{aligned}\nabla^2\phi_{k+1} + \nabla\cdot\left(\frac{1}{c}\left(\frac{A_{k+1} - A_k}{h}\right)\right) - \nabla\cdot E^{ext} + 4\pi e\left(n_{k+1}^{ion} - n_k^e + h\nabla\cdot\left(n_e v_0^e\right)\right)& \\ -4\pi h e\nabla\cdot\left(n_e\sigma_e\left(\nabla\phi_{k+1} + \frac{1}{c}\left(\frac{A_{k+1} - A_k}{h}\right) - E_z^{ext}\right)\right) &= 0\end{aligned} \tag{274}$$

and Backward Ampere-Darwin (cf (17))

$$\nabla^2 A_{k+1} + \frac{4\pi}{c}J_{k+1}^{Use} = 0 \tag{275}$$

Expand:

$$\nabla^2 A_{k+1} + \frac{4\pi}{c}\left(J_0 + \sigma_J\left(-\nabla\phi_{k+1} - \frac{1}{c}\left(\frac{A_{k+1} - A_k}{h}\right) + E_z^{ext}\right)\right) = 0 \tag{276}$$

Meanwhile we are also seeking

$$I_z^{presc} = \int_{[r>r_{ins}]} J_z dx = \int_{[r>r_{ins}]}\left(J_0 + \sigma_J\left(-\nabla\phi_{k+1} - \frac{1}{c}\left(\frac{A_{k+1} - A_k}{h}\right) + E_z^{ext}\right)\right)_z dx \tag{277}$$

After the solve, we update $B_{k+1} = \nabla \times A_{k+1}$.

7.4 Backward Darwin with electron viscosity - NOTES

The residuals and variables

We set residuals $\varepsilon^{Gauss}, \varepsilon^{Ampere}, \varepsilon^{Ohm}$ at each data location, that will be zero when discretised ODE equations are satisfied. Including the electron viscous contribution in the Ohm's Law makes it into an ODE in v_e but we could choose to work in $v_e - v_i$. We choose v_e because using $v_e - v_i$ with triangle-centred data creates a nightmarish amount of dependencies given (279). Therefore we have the 7-dimensional residual vector $\left(\varepsilon^{Gauss}, \varepsilon^{Ampere}, \varepsilon^{Ohm}\right)$ which is to be set to zero by adjusting the 7-dimensional (ϕ, A, v_e). We also set the single value E_z^{ext} for the external electric field, and want to achieve one further equation,

$$\int_{\{r>R_{ins}\}} q\left(n_i v_i - n_e v_e\right)_z = I_z^{prescribed} \tag{278}$$

Working in $v_e - v_i$ would have been more convenient for several reasons, such as that it is not as easy to argue for the priority of an error in v_e. However, the main issue is being able to leave ions where they are, and this we can do anyway.

Anticipating the t_{k+1} values of v_{ion}, v_{neut}

In practice when we enter the ODE solver we shall assume that we have pre-computed a set of coefficients to form $v_{ion}^{k+1}, v_{neut}^{k+1}$ as linear combinations

$$v_{ion,k+1} = v_{ion}^0 + \beta_i^e v_{e,k+1} + \sigma_i^E E \quad (279)$$

$$v_{neut,k+1} = v_n^0 + \beta_n^e v_{e,k+1} + \sigma_n^E E \quad (280)$$

It may be worthwhile to explain, why we do not just use the available half-time value "$v_{i,k+1/2}$" as v_i and have done with it. This would mean that once we do evolve v_i we would be getting a more different result – Ampere's Law would actually be further from holding in the system we do end up with. It is more honest and more accurate to use an estimate of the v_i that is that we will end up with at t_{k+1}.

About the E field

We shall assume that the relevant E field is

$$E = -\frac{A_{k+1} - A_k}{ch} - \nabla\phi_{k+1} + E_z^{ext} \quad (281)$$

[add explanation about why this was chosen - discussion of how A is not Coulomb gauge; where is it stored?]

Note that this is a bit of a mix-and-match, because we have no leapfrog : $-\frac{A_{k+1}-A_k}{ch}$ corresponds to time $t_{k+1/2}$ whereas, we also have $-\nabla\phi_{k+1}$. This just means that for v_{k+1} being affected by E, we are using a backward term as regards ϕ but a midpoint term as regards A.

We are mostly concerned with taking account of the feedbacks from A to itself and from ϕ to itself that would exist in an explicit system. The inductive part basically serves to accelerate z velocity, which affects the A field, while the electrostatic part serves to alter electron motion in the plane, which affects the ϕ field.

7.4.1 Relative scaling of residuals

When we seek to minimize an error criterion such as the sum of squares of $\varepsilon^{Gauss}, \varepsilon^{Ampere}, \varepsilon^{Ohm}$ and ε^{I_z}, we implicitly give a priority to each of the types of residuals. If we give the same priority to getting the correct v_e as to getting the correct $10^6 A$ then we give 10^6 times less priority to getting the correct A. Therefore the relative scaling of equations is a choice that has to be based heuristically on physics.

Considering Maxwell's equations, it is not hard to infer that we should be giving equal priority to setting E and B fields. So suppose that we allow

$$\varepsilon^{Gauss} = \nabla \cdot E - 4\pi\rho$$

$$\varepsilon^{Ampere} = \nabla^2 A - \frac{4\pi}{c} J = -\nabla \times B + (\nabla(\nabla \cdot A)) - \frac{4\pi}{c} J[\text{check!}]$$

Then where ε^{Ohm} contains terms in $v_e - v_i$, we would want the factor $\frac{4\pi}{c} qn$ to appear against $(v_e - v_i)$. Since we work in v_e, we have to rely on the idea that adding v_i just represents a translation and does not change the necessary scaling.

$$\varepsilon^{Ohm} = \frac{4\pi}{c} qn\,(v_e - [\text{inferred formula for } v_e])$$

Meanwhile it seems reasonable that we include the factor $\frac{4\pi}{c}$ for ε^{I_z} because this will make contributions to it at each vertex similar to the contributions to ε^{Ampere}, so that

$$\varepsilon^{I_z} = \int_{\{r>R_{ins}\}} \frac{4\pi}{c} q\,(n_i v_i - n_e v_e)_z - \frac{4\pi}{c} I_z^{prescribed}$$

[check against code]

As well as relative scaling of $\varepsilon^{Ampere}, \varepsilon^{Gauss}, \varepsilon^{Ohm}$ relative to each other, we can consider scaling them at each location relative to the scaling at other locations. Particularly, we may want to use ε formulae that are stripped of proportionality to n. It seems from experience that for $n = 10^{18}$, we should want to allow that ρ is (perhaps 10^6 times) further from $\nabla \cdot E/4\pi$ than it is for $n = 10^{12}$. We can consider applying a factor $n^{\alpha-1}$ to all the residuals, ie such as

$$\varepsilon^{Gauss} = (\nabla \cdot E - 4\pi\rho)\, n^{\alpha-1}$$

It has been observed that if $\alpha = 1$ (no factor) then we do eventually find the Backward Gauss solver going below floating point – e.g. trying to solve for an error in charge density of ~10^4 on a density of 10^{19}. It remains to be seen if $\alpha = 0$ is a good choice: does it make sense to put so much more effort into the low density regions that do not matter? Perhaps the peak of density just means that a higher error threshold could be applied everywhere – or perhaps we should take $\alpha = 1/2$.

7.5 What schemes in time can we do?

What did not work

I wanted to apply a_k, the acceleration due to E and thermal pressure from time t_k, for the interval $\left(t_k, t_{k+1/2}\right)$ and then switch to a_{k+1} for $\left(t_{k+1/2}, t_k\right)$. The crude switchover is because it is in practice non-trivial to entertain applying a_{k+1} on the t_k mesh and vice versa, when frictional heating etc is included. So the sequence I tried was:

1. Perform first half of momentum evolution on t_k mesh
2. For heavy species, run feint subcycle to find the relationship between pressure at t_{k+1} and displacement over $\left(t_{k+1/2}, t_{k+1}\right)$. Solve for new vertex positions. Create new mesh.
3. Advect all to new mesh, advecting cell data per the species vertex move. Move electrons with ions.
4. Run feint subcycle (on t_{k+1} mesh) to find the relationship between E_{k+1} applied over $\left(t_{k+1/2}, t_{k+1}\right)$ and $\int_{t_k}^{t_{k+1}} v_e dt$. Or for more stability and less accuracy, find $v^e_{k+1}\left(E_{k+1}\right)$ and assume for advection that $\int_{t_{k+1/2}}^{t_{k+1}} v_e dt = \frac{h}{2} v^e_{k+1}$.
5. Solve for (A, ϕ) and flow electrons (and ions) accordingly.
6. Perform second half of momentum evolution on t_{k+1} mesh, given E_{k+1}.

The result of this is that even with the modification for a Backward Euler electron half-step, the effect of E_{ES} on ρ is too feeble. E has to work too hard to cancel out the aberration caused by the forward half of the timestep, so in the next forward step v_e is heading the other way; increasing oscillations are the corollary.

What does work

Changing back to full Backward Euler current flow prevents nasty things from happening. The sequence with Backward Euler:

1. For heavy species, run feint subcycle to find the relationship between pressure at t_{k+1} and displacement over (t_k, t_{k+1}). Solve for new vertex positions. Create new mesh.
2. Advect all to new mesh, advecting cell data per the species vertex move. Move electrons with ions.

3. Run feint subcycle (on t_{k+1} mesh) to find the relationship between E_{k+1} applied over (t_k, t_{k+1}) and v^e_{k+1}. Assume for advection that $\int_{t_k}^{t_{k+1}} v_e dt = h v^e_{k+1}$.

4. Solve for (A, ϕ) and flow electrons (and ions) accordingly.

5. Perform momentum evolution over (t_k, t_{k+1}) on t_{k+1} mesh, given E_{k+1}.

Some other salient points however:

- We get ρ_{k+1} on mesh t_{k+1} and it is the actual ρ_{k+1} we then have - we are not somehow predicting advection and then doing it a different way.

- Since we run the subcycle on mesh t_{k+1} we are able to have a realistic effect of E on J - instead of ignoring the second-order effects of friction as previously.

- Since electrons are not heavy, electron thermal pressure would on its own accelerate them a large amount. Some additional momentum is transferred to neutrals or ions through friction but, the E_{k+1} is mostly there to cancel what is left over, transferring most of the remaining additional momentum to ions. So it was important that we are able to assess the thermal pressure we are going to be applying, before we go to the E solver.

8 Approaching discretised ODE solutions for Backward Darwin

We now need to determine what it means for a simulation to use the Backward Darwin equations (274, 276, 277) together with the boundary conditions discussed earlier.

From 8.3 onwards, I will be discussing the procedure for that. There are 3 steps we shall make in order to approach this:

1. Discretisation decisions : decide on geometric configuration.
2. Write the discretised equations by integrating over small regions.
3. Develop a geometric multimesh solver for this set of sparse linear equations.

This chapter will handle steps 1 and 2. First we describe why we aim for a direct solution.

8.1 Direct solution versus other approaches

To solve for Backward Gauss there are various cunning methods that might be employed. Since the electric field caused by a putative charge density is generally in the right direction, it can be tempting to think that we should make putative forward steps, solve Poisson, and try adding a little of that E to see if we can get closer to a mutually consistent solution.

However, it was found that determining the correct magnitude to use for this addition is pretty hard. Inevitably you end up looking ahead to ask, if we add λ times the E then how fast is it reducing ρ_{k+1}?. Then it becomes clear: this way, with a global coefficient on a function that is not quite the addition we need, appears far less efficient than performing relaxation for Backward Gauss itself.

Were we to only be solving Ampere (ignoring the inductive effect of changing B on current), we could write the solution for B according to a Biot-Savart integration. But even if we could find an exact solution of the same sort for our problem, this is highly inefficient: doing a quadrature for B at every location, separately, fails to share information between the evaluations. This will turn out to take a very long time because of that, compared to better methods that do find the whole B field at once. Previously I focused on how the birth-diffusion perspective could allow us to approach the problem without discretised differential operators. However, the optimal radius of effect for diffusion is actually very small, because then information can be shared in the next diffusion step, so birth-diffusion naturally leads just to relaxation (Jacobi). Then relaxation is better because it does involve chalking out a set of linear equations to which multimesh can be applied. Poisson can be solved very efficiently with multigrid-type methods. However, the birth-diffusion viewpoint can be seen as a foundation for believing that we _can_ use the differential approach.

Direct solution faces its own challenges compared to just a generic sparse linear equation solve, because however we set it up, there are vastly differing flow rates between cells or vertices, due to the differences in n_e. There may be some value in borrowing some ideas from the indirect approach with repeated feints, e.g. we could consider adding a scaled move to ϕ that is given by solving Poisson for the putative charge distribution. (I have not yet explored that idea.)

8.2 Finite volume -based differential operator discretisations

8.2.1 Gradient

Suppose we wish to estimate the gradient of a scalar field ϕ within a polygon D, and we know ϕ on the corners of D.

Heuristically, note that in integrating $\frac{d\phi}{dx}$, we take $(\phi\,(\text{right}) - \phi\,(\text{left}))$ integrated with respect to y. Therefore we can say that if the value of ϕ is given at corners of a polygon and we apply a trapezoidal rule,

$$\int_{cell} \frac{d\phi}{dx} = \sum \phi_{edge}\,(y_{i+1} - y_i) = \frac{1}{2}\sum (\phi_{i+1} + \phi_i)\,(y_{i+1} - y_i) \tag{282}$$

where i increases anticlockwise. Meanwhile for $\frac{d\phi}{dy}$ the opposite holds:

$$\int_{cell} \frac{d\phi}{dy} = \frac{1}{2} \sum \left(\phi_{i+1} + \phi_i\right) \left(x_i - x_{i+1}\right)$$

This same method can be applied to integrating $1 = \frac{dx}{dx}$ on the shape, leading to the shoelace formula for area:

$$\int_{cell} 1 = \frac{1}{2} \sum \left(x_{i+1} + x_i\right) \left(y_{i+1} - y_i\right)$$

In summary, if a set of polygon corner positions (x_i, y_i) are known to be sorted, either clockwise or anticlockwise:

$$\nabla\phi\left(D\right) := \frac{\int_D \nabla\phi dx}{\int_D 1 dx} = \frac{1}{\sum x_i \left(y_{i+1} - y_{i-1}\right)} \begin{pmatrix} \sum \phi_i \left(y_{i+1} - y_{i-1}\right) \\ \sum \phi_i \left(x_{i-1} - x_{i+1}\right) \end{pmatrix} \tag{283}$$

Note : that seems to be a rearrangement.

8.2.2 Divergence

Integrating divergence of a vector field v over a region D, we get a line integral around the boundary of the normal v.

$$\int_D \nabla \cdot v \; dx = \int_{\partial D} v \cdot \hat{n} \; ds$$

Therefore on a polygon D,

$$\int_D \nabla \cdot v \; dx = \sum v_{edge \; i} \cdot \perp_i \tag{284}$$

where $\perp_i$ is the normal to edge i having magnitude as the length of edge i.

8.2.3 Curl

To get B from A we try to use the Kelvin-Stokes Theorem. This states that the flux through a surface of $\nabla \times A$ is equal to the line integral of A around the edge. Therefore (careful to get the sign right)

$$\int_R \left(\nabla \times A\right)_z dxdy = \int_{\partial R} A \cdot \partial R$$

and to get $\int_{\partial R} A \cdot \partial R$, clearly we can apply a model of A that interpolates between polygon corner values of A. Therefore dividing by area we get average B_z on each cell.

$$B_z = \frac{1}{\text{Area}} \sum_{sides} \int_0^1 \left(tA\left(x_{j+1}\right) + \left(1-t\right) A\left(x_j\right)\right) \cdot \frac{x_{j+1} - x_j}{\left\|x_{j+1} - x_j\right\|} dt \tag{285}$$

$$= \frac{1}{\text{Area}} \sum_{sides} \left(\left(\int_0^1 tdt = \frac{1}{2} \right) \frac{\left(\left(A^x_{j+1} - A^x_j\right)\left(x^x_{j+1} - x^x_j\right) + \left(A^y_{j+1} - A^y_j\right)\left(x^y_{j+1} - x^y_j\right)\right)}{\left\|x_{j+1} - x_j\right\|} + A\left(x_j\right) \cdot \frac{x_{j+1} - x_j}{\left\|x_{j+1} - x_j\right\|} \right.$$

$$= \frac{1}{\text{Area}} \sum_{sides} \left(\frac{1}{2} \frac{\left(A_{j+1} + A_j\right) \cdot \left(x_{j+1} - x_j\right)}{\left\|x_{j+1} - x_j\right\|} \right) \tag{2}$$

It remains to ask how we get $B_{x,y}$. We could do the same thing on each side of a prism but instead for now we just note that $\partial/\partial z \left(A_{x,y}\right) = 0$ and therefore $B_x = \frac{\partial}{\partial y} A_z, B_y = -\frac{\partial}{\partial x} A_z$. Therefore

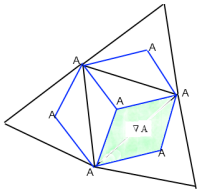
$$\int_R B_x dxdy = \int_{x-projection(R)} \left(A_z\left(\text{top}\right) - A_z\left(\text{bottom}\right)\right) dx$$

8.2.4 Laplacian

Given the above, if we want to integrate Laplacian of a scalar ϕ over a polygon D, we may regard it as

$$\int_D \nabla^2\phi \; dx = \int_D \nabla \cdot \nabla\phi \; dx = \sum (\nabla\phi)_{edge\ i} \cdot \perp_i \tag{287}$$

(For vector Laplacian the same simply applies to each dimension of the vector.)



Now from here we can proceed in one of two ways.

1. Firstly we may be able to exploit some Delaunayhood properties:

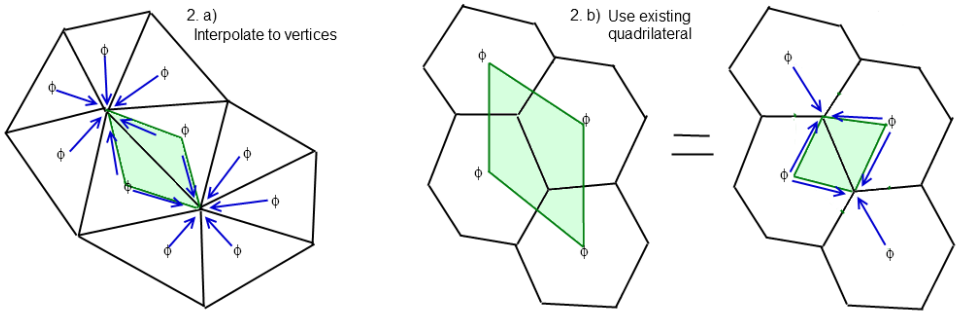
   (a) If ϕ is defined on circumcenters of Delaunay triangles D, then $(\nabla\phi)_{edge\ i} \cdot \hat{\perp}_i$ is obtained as the gradient between ϕ at the circumcenter of this triangle and ϕ at the circumcenter of its neighbour.

   (b) If ϕ is defined on vertices and we desire to integrate $\nabla^2\phi$ over true Voronoi cells, rather than say, cells formed by linking triangle centroids, then the same thing applies: the vector to the neighbour ϕ is normal to the edge of D and the two values provide an instant estimate of $(\nabla\phi)_{edge\ i} \cdot \hat{\perp}_i$.

2. Alternatively, if ϕ is defined on a set of points surrounding the edge, including through averaging, then we may infer $(\nabla\phi)_{edge\ i}$ per (283). This will come about in one of two ways:

   (a) If the D's are triangles and ϕ is defined on triangle centroids, we have to infer the values at the vertices by some suitable interpolation. (The interpolation weights that mostly work are "triangle angle/distance" - but this is not justified here.) This means the Laplacian depends on the values of ϕ in all indirect neighbour triangles of D, ie those that share a vertex with it.

   (b) If D's are polygons with the honeycomb property that 3 walls meet at each corner, $(\nabla\phi)_{edge\ i}$ is always to be inferred from 4 ϕ values, all of which belong to either this D or its direct neighbours.



There are several reasons why Way 1 is not actually that desirable.

- Circumcenters can turn out to actually be very close together, if triangles are near right-angled. This creates a need for special treatment in a solve.

- It cannot extend to another case, that we are going to need: $\nabla \cdot (\sigma\nabla\phi)$ where σ is a tensor. Taking a quadrilateral of ϕ about each edge gives us a way to assess a vector E instead of just normal E.

8.2.5 Concerning the Desbrun cotangent Laplacian

I note in passing that Way 1 above is equivalent to the so-called Desbrun cotan Laplacian, a discretised Laplacian popular in some fields. Information about the Desbrun Laplacian may be found in: [MeyerDesbrun2002, PinkallPolthier93, DesbrunMeyer00, LXZ08].

Let δ_i be the distance from central point x_0 to a neighbour x_i. Let α_{0i} and β_{0i} be the angles opposite edge $0i$ in the triangles sharing the edge $0i$ (Desbrun's label for them). Apparently, the length of an edge of the Voronoi cell formed about x_0 with respect to its neighbours, is found as

$$d = \frac{1}{2}\left(\cot\alpha_{0i} + \cot\beta_{0i}\right)\delta_i \tag{288}$$

and the area of a Voronoi cell is

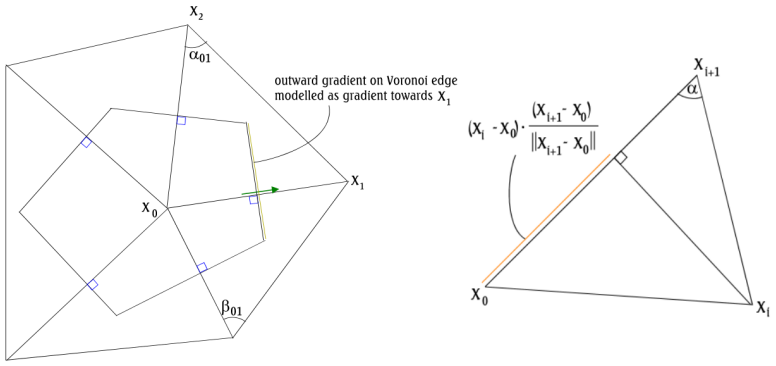
$$Area_{Voronoi} = \frac{1}{8}\sum\left(\cot\alpha_{0i} + \cot\beta_{0i}\right)\delta_i^2 \tag{289}$$

This approximating the Laplacian as described above (cf [DesbrunMeyer00])

$$\frac{\int_{Voronoi}\left[\nabla^2 u\right]dx}{Area_{Voronoi}} = \frac{\int_{\partial Voronoi}\nabla u\cdot\hat{n}d\ell}{Area_{Voronoi}}$$

Hence

$$\nabla^2 u \approx \nabla^2_{Desbrun}u = \frac{\frac{1}{2}\sum\frac{u_i-u_0}{\delta_i}\left(\delta_i\left(\cot\alpha_{0i}+\cot\beta_{0i}\right)\right)}{Area_{Voronoi}} = \frac{\frac{1}{2}\sum\left(\cot\alpha_{0i}+\cot\beta_{0i}\right)\left(u_i-u_0\right)}{\frac{1}{8}\sum\left(\cot\alpha_{0i}+\cot\beta_{0i}\right)\delta_i^2}$$



Working with cotangents The cotangent is adjacent over opposite in a right-angled triangle. Therefore for the adjacent side on α_{0i} we can take, by projecting $x_i - x_0$ on to the direction from x_0 to x_{i+1},

$$\|x_{i+1}-x_0\| - (x_i-x_0)\cdot\frac{(x_{i+1}-x_0)}{\|x_{i+1}-x_0\|}$$

and for the opposite side, $(x_i - x_0)\cdot\frac{(x_{i+1}-x_0)^\perp}{\|x_{i+1}-x_0\|}$. Therefore

$$\cot\alpha_{0i} = \frac{(x_{i+1}-x_0)\cdot(x_{i+1}-x_0) - (x_i-x_0)\cdot(x_{i+1}-x_0)}{(x_i-x_0)\cdot(x_{i+1}-x_0)^\perp} = \frac{(x_{i+1}-x_i)\cdot(x_{i+1}-x_0)}{(x_i-x_0)\cdot(x_{i+1}-x_0)^\perp}$$

Therefore

$$\begin{aligned}\nabla^2_{Desbrun}u &= \frac{\frac{1}{2}\sum\left(\frac{(x_{i+1}-x_i)\cdot(x_{i+1}-x_0)}{(x_i-x_0)\cdot(x_{i+1}-x_0)^\perp} + \frac{(x_{i-1}-x_i)\cdot(x_{i-1}-x_0)}{(x_i-x_0)\cdot(x_{i-1}-x_0)^\perp}\right)(u_i-u_0)}{\frac{1}{8}\sum\left(\frac{(x_{i+1}-x_i)\cdot(x_{i+1}-x_0)}{(x_i-x_0)\cdot(x_{i+1}-x_0)^\perp} + \frac{(x_{i-1}-x_i)\cdot(x_{i-1}-x_0)}{(x_i-x_0)\cdot(x_{i-1}-x_0)^\perp}\right)(x_i-x_0)\cdot(x_i-x_0)} \\ &:= \sum w_i\left(u_i-u_0\right)\end{aligned} \tag{290}$$

QED. Note that in a Delaunay mesh we are guaranteed that the cotangents are always positive because we can rule out obtuse triangles: these would have a circumcenter outside the triangle.

8.3 Discretisation of the ODEs: the ways that can work

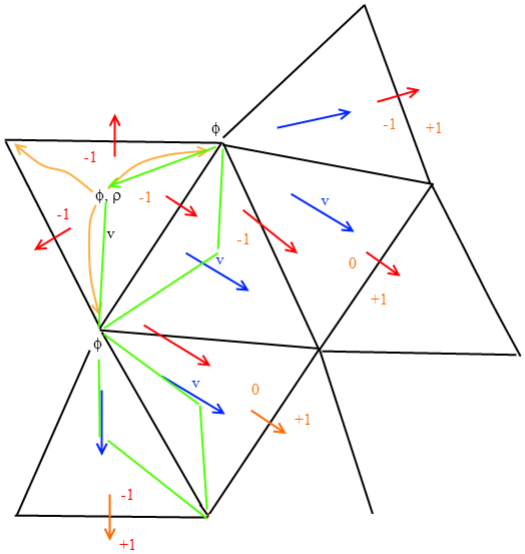
We need to decide the geometric configuration: particularly, whether various variables live on triangles or on vertices.

Some properties that are desirable in a geometric configuration:

- There is a sense in which the value associated with an equation (e.g. ϕ at a point is associated with Gauss's Law integrated over a small region surrounding that point) is the main influence on the residual of that equation. Get as close as possible to the diagonal dominance that would make Jacobi run smoothly.
- Especially, the relevant variable for an equation should be located in the same place as the variable targeted in the equation. ϕ should not be located offset from ρ, nor A from J.
- There is not a need to separately re-compute E or another field for the purposes of the rest of the simulation. (This can result in instability.)
- Complexity - especially number of dependencies - is manageable.
- It should be the same configuration as is used in the simulation proper. Otherwise whatever re-mapping is necessary, runs the risk of causing problems, upsetting Darwin.
- As far as possible, symmetric coefficients : if Y_1 affects ε_2 then Y_2 affects ε_1 similarly.
- When everything lives on vertices, there is the greatest simplicity for multimesh.
- Generally, avoid reliance on Delaunay / circumcenters, for the reasons discussed above.

8.3.1 Tri-centered configuration

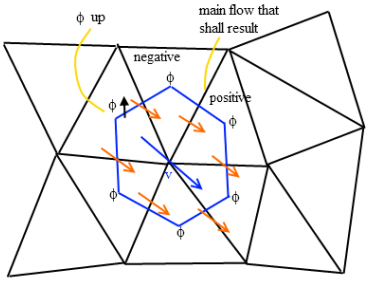
In the case that v_e is just a linear function of E, a tri-centered structure can work fine. It cannot work if v is the subject of an Ohmic ODE. For example, just putting all variables on triangles, ϕ triangle affects ϕ vertex which affects v in an indirect neighbour, and so transmits charge to a further-out cell.



Necessary sleight of hand with E_{edge}: We don't want that so we use some sleight of hand. ϕ on cells, together with ϕ interpolated on vertices, provides E_{edge}. This is used to determine v_{edge} in tandem with v on each side of the edge, using conductivity σ that is averaged from the two sides. This avoids what would otherwise happen, that we would be posting mass into cells the other side of a row.

We could not do this if we were solving at the same time for v in an ODE, because in that case, the cell v being obtained, has to be used in anger. We cannot take a v for performing current flows that is constructed separately, because it is too complicated to try to cancel the effect of local ϕ already included in the v solution.

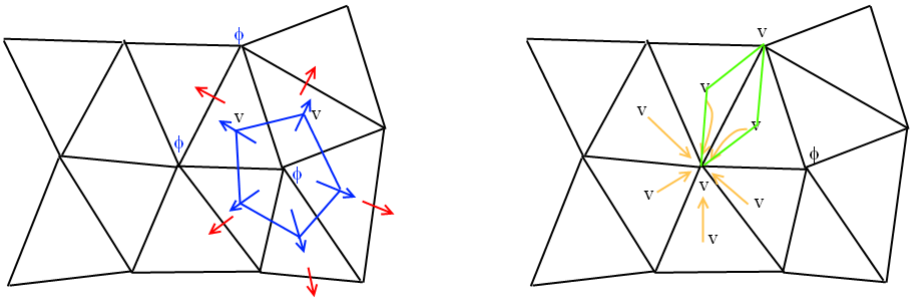
However, absent viscosity we can proceed by creating and using E_{edge} and there is no reason for it not to work. The discretised equations are given in what follows.



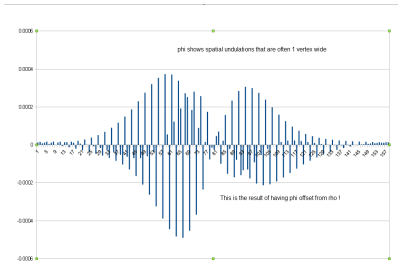
Things that do not work : v on vertices, ϕ on cells

It is not good to put v on vertex-centred cells, and ϕ on cells, because then also, we get a big deviation from diagonal dominance. Some of the edges meeting at the vertex are more normal to the line from the vertex to the centroid where ϕ is located – so the main effect of changing ϕ is to push charge between some other cells.

Things that do not work: ϕ on vertices, everything else on tris



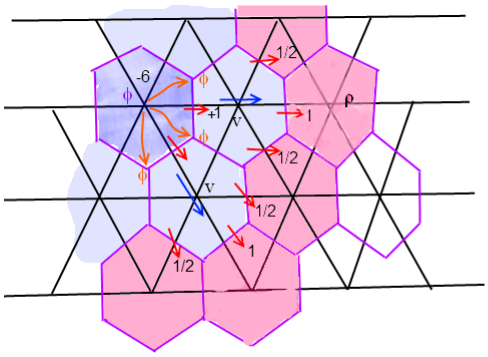
- The number of ϕ values is half the number of Gauss equations. This means that the equations are not perfectly satisfied.
- Configurations with an offset from the equation being solved, generally fail. Usual experience using clusters to give rise to equations is that we see a forest of error spikes grow after some time; that is due to the fact that ϕ is offset from ρ. Imagining in 1D, if we have one high positive charge cell, that leads to the corner ϕ pushing equally on a negative-charge cell on the other side of the ϕ location. That may be pushing positive into another positive-charge cell. Some actual results on a row of cells:

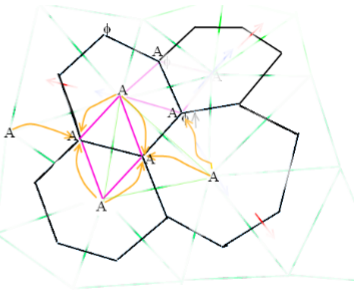


8.3.2 Vertex-centered configuration

If the rest of simulation is designed to match, then it is possible to make a vertex-centered configuration work, placing all variables on vertices. Not only that, but it should be suitable for an ODE Ohm's Law, or an ODE relationship coming from a dynamic acceleration routine.

- For $\nabla^2 A$, we are only affected by the A values in neighbouring QV cells.
- If we now want to be able to include the viscous case, we must decide to let v_{edge} come from v_{vertex}, and this will be affected by (the interpolated) ϕ on vertices. Setting ϕ in cells, taking ϕ average (in fact, interpolant using a plane from 3 values) on to the dual mesh, we then apply E_{cell} to v_{cell}.





This left figure illustrates how changing ϕ in a location has the greatest effect in its own cell. If ϕ is simple-averaged on to offset positions, to infer v in an Ohmic ODE, then there is a knock on effect on cells further out - but the charge is being dispersed. In the triangle case, an indirect neighbour E field is affected by ϕ and then affects a further afield cell; the total effect on the solution is greater than just spreading charge away from the ϕ position.

Here, actual neighbours will have ρ move the same way as in the cell where ϕ changes, but to a lesser extent. It seems quite conceivable that changes of this kind could sum to a decent solution of Gauss equations.

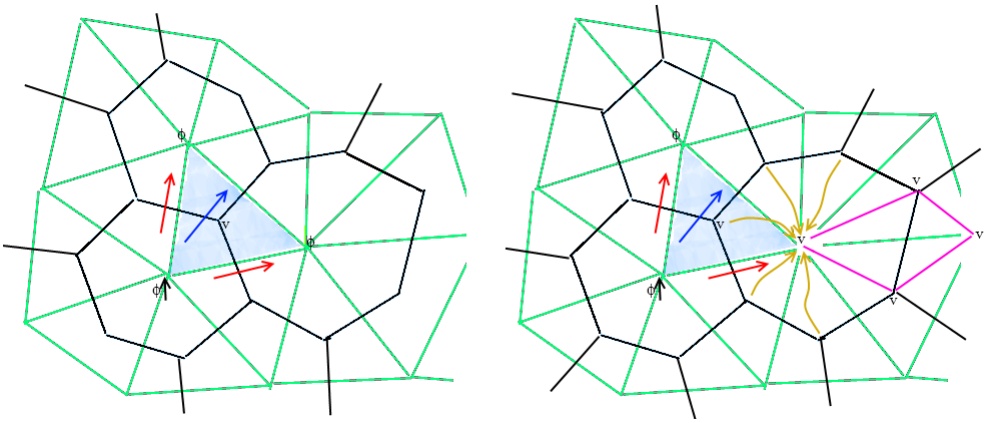
Note that when ϕ changes, $\nabla \cdot E$ does not change in this cell at all. For the neighbour cells, changing E of its own does not affect $\nabla \cdot E$; the neighbour change in E does affect it. Suppose that ϕ increases at the central cell. Then E in the neighbours points inwards. Therefore $\nabla \cdot E$ in the neighbours has decreased slightly since $\nabla \cdot E$ comes from the outward E normal to the edges. Meanwhile, the density of electrons decreases in the central cell and its neighbours; ρ increases. Thus $\varepsilon^{Gauss} = \nabla \cdot E - 4\pi\rho$ is affected in the same direction by both changes. Still the effect on ε^{Gauss} in the central cell should be much the greatest; even when we are close to equality, ε^{Gauss} is still mostly affected by changing ρ.

Note that for the ρ effect, on an equilateral mesh with constant density and conductivity, there is an exact balance: the desired change in ρ in the central cell comes at the cost of the same, same-sign change distributed between its neighbours. But this balance is tipped by having the additional effect via $\nabla \cdot E$.

With v on offset (triangle) cells, ϕ on vertices: not as good

We will be forced to split $\int nv$ into vertex-centred cells afterwards so this is not a good way.

In this case we would have a Gauss equation for each vertex-centred cell affected by v on each triangle cell. It is more viable than putting v on vertices and ϕ on triangles because the main result of changing ϕ is now not untoward. But still not as sensible as putting everything on vertices.



8.3.3 More about the vertex-centred data structure

We can form Voronoi cells from the circumcenters of triangles or in 3D, as the locus of nearest points to a vertex. However this is not the best choice since with magnetism we cannot often directly exploit the fact that edges are normal to cell edges. Instead we should form vertex-centred cells from centroids; I sometimes call these Quasi-Voronoi. In graphics it is called a barycentric mesh, though this seems to be something of a misnomer as far as we are concerned.

- The biggest difficulty is that when two vertices (mentally, any data objects that have a stored position) move towards each other, you want there to be compression of an object between them in order to make thermal pressure keep them spaced apart. The most simple way is therefore that nT lives on the dual mesh from the vertices. However, if we arranged for a spline construction to provide nT on the dual points, it might help somewhat. (There are still configurations where nT is constant across cells which have some vertices closer together than others.)

- For a 3D mesh with equilateral tetrahedra, the vertex-centred cells likely become icosahedra. The icosahedron version is easy to envisage because it has 6 vertex neighbours nearly in a plane and 3 at the top and bottom. They are close to being Voronoi cells, in the sense that in contrast to tetrahedra which have extensions far from the centroid, the icosahedron is close to the locus of nearest points. This is a good property: we may more reasonably think that a variable is more constant over such a cell, than over a tetrahedron.

- There is a natural breakup into planes between corners are the centre of a polygon. When that polygon is a triangle however, connecting to the corners looks less reasonable than connecting to medians: the triangles become obtuse. When the polygon is a hexagon then the 6 triangles are a natural way of dividing the cell. The dual values should be estimated in concert, or, we use a minmod type approach as now.

- In 3D, to get a subdivision into smaller volumes where a scalar is modelled as linear is also obvious: take tetrahedra coming out from the centre to the faces.

- If we instead chose to slice off the corners of triangle/tetrahedral cells to form vertex-centred cells, then we have a mixture of polyhedra in 3D: icosahedra alongside truncated tetrahedra. That is less elegant, although it does address the problem that for thermal pressure we would like to have $\int nT$ live on cells between the moving vertices.

- Here when we take an average to the dual (offset) points, we only average 3 values. It means the way of defining the average becomes obvious, linear interpolation – rather than fairly arbitrary.

- Importantly, the values of A_k and ϕ_k are directly available (at least, the values in the fluid frame) because vertices have a life between timeslices.

- Importantly for the solver, as observed it means that to get grad across an edge we use only neighbours of this vertex-centred cell – therefore to get $\nabla^2 A$ and $\nabla \cdot \Pi_e$ we use only neighbours of this cell.
- Apart from being good for solver complexity and robustness, that property also saves on coefficient memory requirements - which on GPU, for 3D, is important.

8.4 Discretising Backward Darwin ODEs : no electron viscosity

Recall that we need a set of linear equations to represent the ODEs, in effect applying a discretisation to the differential operators. We assume electron viscosity is not in effect. For a triangle-centered solver, applying E_{edge} formed from a quadrilateral of ϕ values around each edge. We obtain a set of equations by integrating the Backward Darwin equations over each triangle, using the differential operator integrals from Subsection 8.2.

8.4.1 Backward Ampere discretisation

Let

$$\varepsilon^{Ampere} = \int_{triangle} \left(\nabla^2 A_{k+1}\right) dx + \frac{4\pi}{c} \int_{triangle} \left(J_0 + \sigma_J \left(-\nabla \phi_{k+1} - \frac{1}{c} \left(\frac{A_{k+1} - A_k}{h} \right) + E_z^{ext} \right) \right) dx$$

where

$$\sigma_J = e \left(n_{k+1}^{ion} \sigma_{ion} - n^e \sigma_e \right)$$

Naturally, $\int_{triangle} \left(\nabla^2 A_{k+1}\right) dx$ is approached via the Laplacian method sketched out above (287): it depends on indirect neighbour A. In order to set up a solver, we assign coefficients to these indirect neighbour values according to their contribution both directly and via the vertices.

Approximate:

$$\int_{triangle} e \left(n_{k+1}^{ion} \sigma_{ion} - n^e \sigma_e \right) \left(\nabla \phi_{k+1} \right) dx = e \left(N_{k+1}^{ion} \sigma_{ion} - N^e \sigma_e \right) \left(\frac{\int_{triangle} \left(\nabla \phi_{k+1} \right) dx}{Area_{cell}} \right)$$

We are then getting

$$\begin{aligned} \varepsilon^{Ampere} \quad = \quad & \int_{triangle} \left(\nabla^2 A_{k+1}\right) dx + \frac{4\pi}{c} q \left(N_i v_i^0 - N_e v_e^0 \right) \\ & + \frac{4\pi}{c} e \left(N_{k+1}^{ion} \sigma_{ion} - N^e \sigma_e \right) \left(-\overset{\text{estimate}}{\nabla} \phi_{k+1} - \frac{A_{k+1} - A_k}{ch} + E_z^{ext} \right) \end{aligned} \tag{291}$$

On the other hand for triangles overlapping the arc $r = R_{reverse} = 2.8$ cm, we set

$$\varepsilon^{Ampere} = \int_{cell} \left[\nabla^2 A\right] - \frac{4\pi}{c} \left(-I_z^{prescribed} \right) \theta \tag{292}$$

where θ is the fraction of the 1/16 arc at 2.8 cm that lies within this triangle. For non-domain triangles not overlapping this arc,

$$\varepsilon^{Ampere} = \int_{cell} \left[\nabla^2 A\right] \tag{293}$$

8.4.2 Backward Gauss discretisation

Let

$$\int_{cell}\left(\nabla^2\phi_{k+1}+\nabla\cdot\left(\frac{1}{c}\left(\frac{A_{k+1}-A_k}{h}\right)\right)-\nabla\cdot E^{ext}+4\pi e\left(n^{ion}_{k+1}-n^e_k+h\nabla\cdot(n_e v^e_0)\right)\right.$$
$$\left.-4\pi he\nabla\cdot\left(n_e\sigma_e\left(\nabla\phi_{k+1}+\frac{1}{c}\left(\frac{A_{k+1}-A_k}{h}\right)-E^{ext}_z\right)\right)\right)dx \;=\; \varepsilon^{Gauss}$$

Then $\int_{cell}(\nabla^2\phi_{k+1})$ creates coefficients which are determined according to (287). If we take a uniform E^{ext} then $\nabla\cdot E^{ext}=0$. v^e_0 can be assessed on each edge so that we apply (284) to $h\int_{cell}\nabla\cdot\left(n^{edge}_e v^e_0\right)dx$.

In the triangle-centered case, we then need to create on each edge, E^{edge}:

$$E^{edge}=-\nabla\phi_{k+1}-\frac{1}{c}\left(\frac{A^{edge}_{k+1}-A^{edge}_k}{h}\right)+E^{ext}_z$$

and this provides coefficients on ϕ_{k+1} at vertices and on triangles; and A_{k+1} on this triangle and its direct neighbours. Then we proceed to:

$$\begin{aligned}\varepsilon^{Gauss} \;=\; &\int_{cell}\left(\nabla^2\phi_{k+1}\right)dx+\sum\left(\frac{A^{edge}_{k+1}-A^{edge}_k}{ch}\right)\cdot\perp_i+4\pi e\left(N^{ion}_{k+1}-N^e_k\right)\\ &+4\pi he\sum n^{edge}_e v^e_{0,edge}\cdot\perp_i+4\pi he\sum n^{edge}_e\left(\sigma^{edge}_e E^{edge}\right)\cdot\perp_i\end{aligned}\tag{294}$$

In the vertex-centered case, we can either do the same thing, or we can choose to construct v implied on each cell via

$$v^e_{cell}=v^e_{0,cell}+\sigma_e E_{cell}=\sigma_e\left(-\nabla\phi_{k+1}-\frac{1}{c}\left(\frac{A_{k+1}-A_k}{h}\right)+E^{ext}_z\right)\tag{295}$$

That then also is the approach that can work when v is determined by an ODE itself.

We will want to calculate ρ that we are expecting to obtain, in the interests of development and debugging.

Of course, v_{neigh} and A_{neigh} must be rotated to apply contiguously.

8.4.3 Discretised equation for $\frac{4\pi}{c}I_z$

As before we take $\frac{4\pi}{c}I_z=\frac{4\pi}{c}\int_{\mathbb{R}^2}J_z$ as the target of an equation, associated to E^{ext}_z. We are given

$$I^{presc}_z=\int_{[r>r_{ins}]}J_z dx=\int_{[r>r_{ins}]}\left(J_0+\sigma_J\left(-\nabla\phi_{k+1}-\frac{1}{c}\left(\frac{A_{k+1}-A_k}{h}\right)+E^{ext}_z\right)\right)_z dx$$

Therefore

$$\frac{4\pi}{c}I_z=\frac{4\pi q}{c}\sum_{\text{domain cells}}\left(N_i v^{0z}_i+N_e v^{0z}_e+\left((N_i\sigma_i-N_e\sigma_e)\left(-\frac{A_{k+1}-A_k}{ch}-\nabla\phi+E^{ext}_z\right)\right)_z\right)\tag{296}$$

$$\varepsilon^{Iz}=\frac{4\pi}{c}\left(I^{prescribed}_z-I_z\right)$$

We actually shall create coefficients for the direct effect of chE_z. This scaling doesn't matter much but it creates a symmetry because the coefficient on $A_{z,i}$ for $\frac{4\pi}{c}I_z$ will be the same as the coefficient on chE_z for $\frac{4\pi}{c}J_z$.

8.5 Discretised ODEs : vertex-centered case, ODE Ohm's Law

8.5.1 ODE Ohm discretisation

8.6 The inner mesh

We are going to want a mesh that goes within the insulator and anode. To make up an inner boundary condition for A at the insulator would not be the correct approach since A on the insulator cannot humanly be predicted in advance. It is important to explicitly take account of the current flowing the reverse vertical direction at the edge of the anode; for one thing, it gives us zero total z current.

We do not have any real reason to change the vertex positions in this inner mesh as the simulation progresses. This would be more complicated than leaving them fixed and equilaterally positioned. For some time, I tried using a slimmed data object for an inner vertex, to save memory, but this was a needless complexity. Non-domain vertices are a minority of the vertices.

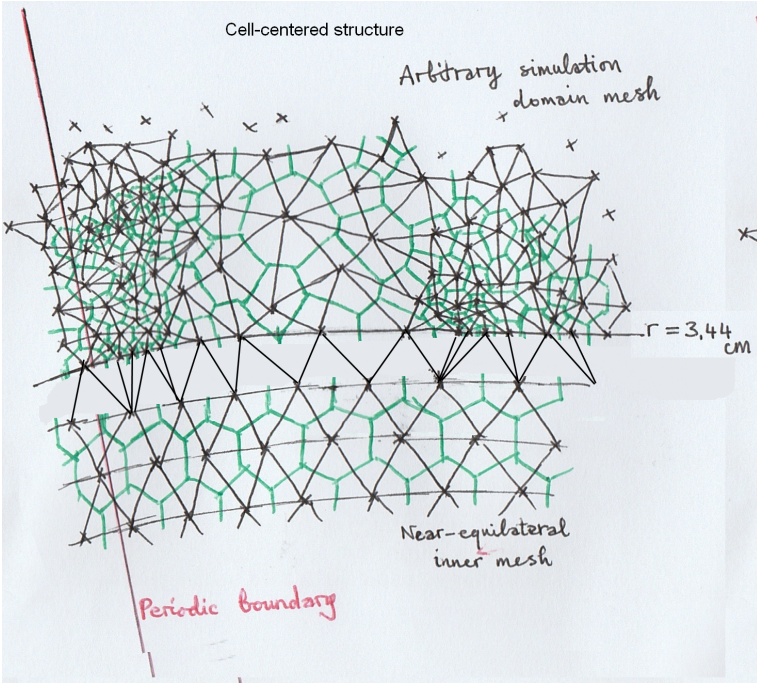
Within the insulator, we are solving Laplace's eqn. Intuitively, 0.1cm inside the insulator, there is sufficient smoothing that 0.1 cm is becoming reasonable resolution - though we do not have to go that far. The anode current is very simple and its long-range effects do not require high resolution around it either. So we can accept a lower resolution for the inner mesh than for the domain initial uniform mesh.

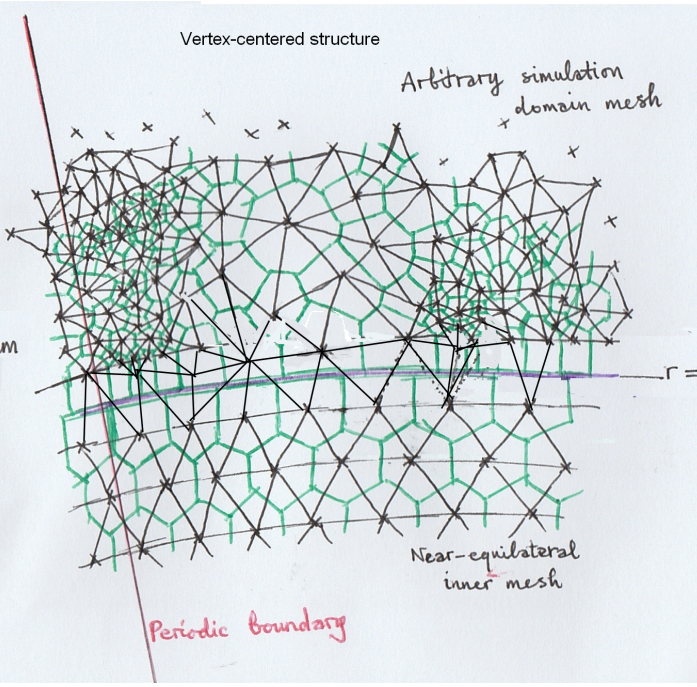
As long as we continue the domain to a sufficient radius, well outside the plasma, we do not need a further mesh outside that. But we do need to mesh within the cathode rod, should it be introduced.

The interface between the fixed inner mesh and the domain mesh Cells absolutely must abut the insulator. No other way will succeed. Therefore if we have a cell-centred data structure, some vertices have to live on the insulator. These . There is not a need for special wedge cells against the insulator - we put actual vertices on there, not projections. We carry out Delaunay flips only within the domain, even though it is not convex. We never create cells that cross the insulator. We have to have some special code for deciding if we want to move a vertex on to the insulator or off of it - which I shall call vertex skips.

On the other hand, in the vertex-centered data structure, we put no vertices on the insulator. We create the vertex-centered cells and wherever one would cross the insulator, we move the corners to reside on the insulator instead. That is, a position on the insulator, within a given triangle, is used in place of the triangle centroid.

Note: Instead of having an inner mesh, could we merely extend the coarser, auxiliary meshes into the anode? The problem with that is we need a fully specified problem on the finest level. The purpose of the coarser levels is not exactly to solve the same problem but on a coarser mesh, so we cannot use them to provide boundary values near the insulator.


Cell-centered structure
Arbitrary simulation domain mesh
r = 3,44 cm
Near-equilateral inner mesh
Periodic boundary


Vertex-centered structure
Arbitrary simulation domain mesh
Near-equilateral inner mesh
Periodic boundary

9 Geometric Galerkin multimesh

9.0.1 Introduction

Given a large set of linear equations generated by an ODE, we want to use an iterative method to approach a solution. The linear equations are numerically unfavourable. For instance, electric conductivity may vary by many orders within the domain. Some Krylov methods were attempted but did not succeed. It is possible that preconditioning could ameliorate matters, but this is more of an art than a science.

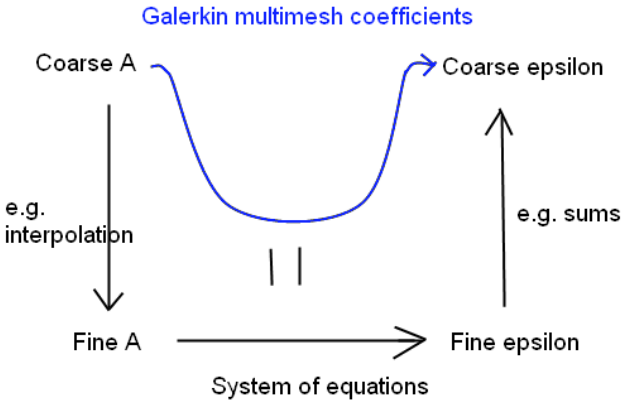
Multigrid on the other hand can has a relatively simple intuition. Pedagogical material and recipes usually treat the case of a Cartesian grid but the idea of applying multigrid to an arbitrary mesh is also well-known.

9.1 Geometric Galerkin multimesh solver

In introductions to modern multigrid, the distinction is drawn between geometric multigrid - usually given in terms of a Cartesian set of grids in which each coarse grid is a subset of the finer level grid and where moving a coarse-level value has an effect on the local finer values - as opposed to algebraic multigrid which has no restrictions on how the coarse levels are formed. Here we create a method for an arbitrary mesh where the coarser levels DO have a geometric interpretation. However we do not follow the pattern that the coarser mesh is a subset of the finer.

We do want to use the idea that a coarse residual ε is linked to a local region because we want to achieve the squashing of the 'low-frequency' error. From the perspective of a finer level, if we can use the coarser level steps to ensure that on small regions, the sum of the residuals is zero, then finer mesh smoothing steps will quickly be able to deal with annihilating error within those regions. As with the electric solver, we can have in mind that the total of residuals over all space is (vector) zero - so we can regard the job at hand as being to annihilate error within progressively smaller regions.

The multimesh is based on Galerkin, which means, as explained previously, that we can regard every coarser-level move of an A value as adding a scaled function to the finer-level A values and therefore, given a target coarse ε that is a function of the fine ε, infer logically the effect of changing coarse A on coarse ε. The same idea then applies between any consecutive pair of meshes.



9.1.1 Constructing Galerkin coefficients

I began with an "earthquake" type of multimesh where each value . This is advantageous from the perspective of constructing a control loop because it can be guaranteed that

The Galerkin property implies that

To construct Galerkin coefficients at the coarser level, we can work from the finer level. Everywhere that there is a dependency of some ε on some x, we .

It is necessary to make allowances for effects across the periodic boundary. Therefore especially,

9.1.2 Prolongation function

For the "prolongation" step, ie, the effect that an additional coarse A has on the A values at the fine vertices, we will want to be affecting things locally (to get a property similar to diagonal dominance for the coarser mesh: we want the biggest influence of coarse A_i^{coarse} to be on the fine ε that affect ε_i^{coarse}). Previously for ϕ I tried an earthquake method, with constants added to the finer ϕ in a region; this produces a marked cliff at the edge which is then subject to erosion. Interpolation of some sort, which is more conventional (at least for the classic Cartesian multigrid), creates a situation where to achieve a push out of a region's boundary, we also have to push outwards within that region. That is advantageous for one thing because there is no cliff to erode and so the gains are retained at the finer level.

It is not clear what interpolation is most efficient for these problems. On each vertex, I store an index into the coarser mesh triangle list, an index into the vertex list to indicate whose Voronoi cell it is in, and 3 weights that allow us to quickly interpolate from the corners of that coarse triangle. To get linear interpolation, the relative weights are given by the areas of the opposing triangles within that triangle if it were subdivided by this vertex. However, I might consider instead squaring these weights before normalising.

9.1.3 Restriction function

The initial coarser-level ε 3-vector is a function of the finer-level ε 3-vectors in a Voronoi-like cell. This step is called by some people 'restriction'. We want the choice of ε_{coarse} to enable the object of annihilating error in progressively smaller regions to be achieved, so one obvious choice is just to take the sum of ε_{fine} in coarse Voronoi cells.

This is not what we now do, because it spoils the symmetry of the coefficient matrix at the coarser level, and that gets progressively worse with each level. Symmetry is important for several reasons: firstly, it provides the conditions for Jacobi to converge (and the degree of asymmetry otherwise seen does seem credibly to prevent Jacobi converging); secondly, it implies that $\sum \varepsilon$ should be constant (apart from rounding errors) and indeed the same on every level; thirdly, it makes debugging easier. The only way we can get symmetry is when the restriction uses the same weights as are used for prolongation. ie where

$$\left(A_{fine}^{additional}\right)_i = w_{1i}A\left(x_1\right) + w_{2i}A\left(x_2\right) + w_{3i}A\left(x_3\right) \tag{297}$$

$$\varepsilon_{coarse}\left(x_1\right) = \sum_i w_{1i}\varepsilon_i \tag{298}$$

For then where we have a symmetric contribution on the fine mesh:

$$\varepsilon_i^{fine} += \beta_{ij}A_j^{fine}$$

$$\varepsilon_j^{fine} += \beta_{ij}A_i^{fine}$$

and x_1^C is a corner of the coarse triangle containing fine point x_i, and x_2^C is a corner of the coarse triangle containing fine point x_j, we get symmetric coarse coefficients:

$$\varepsilon_{coarse}\left(x_1^C\right) += w_{1i}\beta_{ij}w_{2j}A_{coarse}\left(x_2^C\right) \tag{299}$$

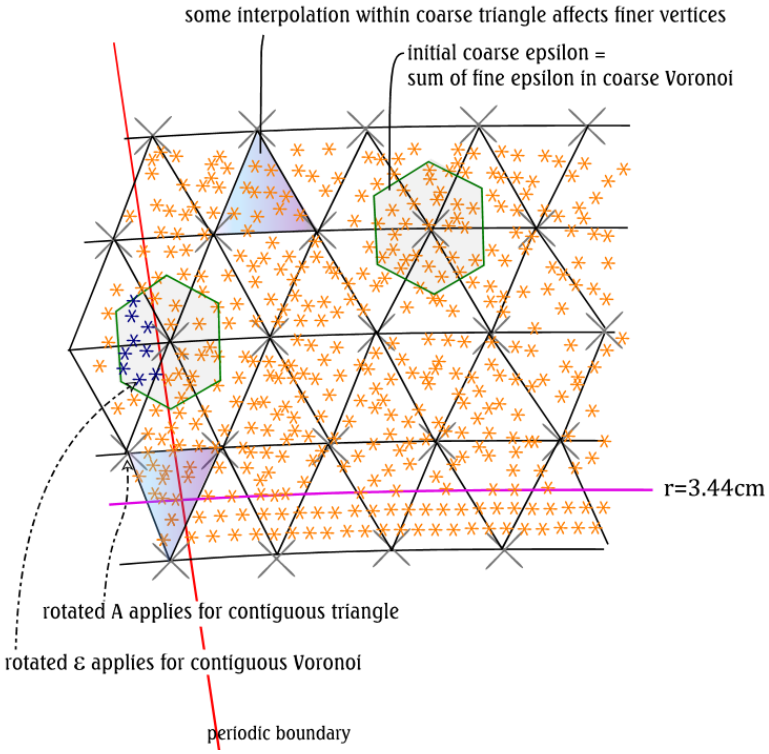
$$\varepsilon_{coarse}\left(x_2^C\right) += w_{1i}\beta_{ij}w_{2j}A_{coarse}\left(x_1^C\right) \tag{300}$$

(Of course we write without periodic wrapping for simplicity.)

For ϕ on the finest level we are interpolating on to centroids; for A we are interpolating on to vertices. From the perspective of the above discussion it makes no difference.

9.1.4 Periodic boundary handling

We must rotate ε contributions so that the Voronoi cell is contiguous, not wrapped across the PB. The fine A is affected by some kind of interpolant from coarse A with periodic wrapping that is similar: we make the coarse triangle a contiguous image around the fine vertex. These choices enable us to ignore the PBC when accumulating coefficients and just know that on each level, an A contribution to an equation for an ε is rotated if and only if the contributing vertex is across the PB.



However we need to be very careful: within coarser levels we no longer have only neighbour contributions to ε, since we are allowing that fine A can affect fine ε across an edge (or more than 1 edge, near a vertex) of coarse triangles – and this certainly will happen – so giving an interaction between the A and ε of two coarse points that are not direct neighbours. This means it is not sufficient to approach periodicity by just storing a flag for those that have neighbours across the PB and rotating contributions according to the signs of the x positions. Instead we need to create a "look_out_for_periodic" flag as we populate the auxiliary coefficients. For an auxiliary vertex with $x < 0$, we store the gradient that is for x_{rot}/y_{rot} for (x, y) rotated $2\pi/32$ clockwise; to the left of this, neighbours will be unwrapped; to the right they are rotated anticlockwise iff the "look_out_for_periodic" flag is set. This is the best we can do without actually storing a 2x2 matrix for every coefficient.

9.2 Jacobi-Richardson Least Squares

Jacobi's method Basic implementations of multigrid often use Jacobi's method at each level. This is logical since Jacobi's method reduces the high-frequency errors and therefore it suits perfectly the multigrid approach where the purpose of the multigrid is to affect different residual frequencies while the main purpose of the iterator at each level is to affect the frequency band between the highest on that level and the highest on the next level. However, for our problem it is quite difficult to find discretisation configurations on which Jacobi itself is stable. The well-known sufficient conditions for Jacobi convergence include diagonal dominance, which can be hard to attain (regardless of how we associate variables with equations).

Generalised Jacobi's method It is convenient to mention that although we may choose to associate equations bijectively with variables for which we solve, we still can need to generalise Jacobi's method where there are multiple interrelated ODEs being solved.

Krylov methods It is sometimes said that using Krylov methods within multigrid is an active area of research. My experience with this was that it does not work at all well, because Krylov methods do not generally have the property of working on the high-frequency residuals. Therefore where Krylov methods work at all, they are subject to rather little benefit from visiting coarser levels. On the other hand, we do need a property that happens to be common to various Krylov methods, which is guaranteed improvement of an error norm, which must be based on treating various vectors (e.g. Lanczos vectors) as regressors for the remaining residuals.

Jacobi-Richardson Least Squares Therefore I had to invent a method which we shall call Jacobi-Richardson Least Squares. On finding that underrelaxation cannot be indefinitely guaranteed not to break down, it becomes clear that we need to determine dynamically what coefficient (or "weight") to use for the Jacobi move. For Jacobi Least Squares, we could create a Jacobi vector

$$\delta_{Jacobi} = \overline{\beta}$$

where here we have assumed that in some sense variable i should be associated with residual i (and).

It is not the case that we can always get convergence with Jacobi Least Squares. It is possible to get into a situation where the remaining residuals are "A-orthogonal" to the Jacobi vector and no scaling of it will provide progress. However, with a modification it is seen to work, which is when we also use the vector of residuals itself as a regressor (iterating by subtracting the residuals from the values is Richardson's method). <– check

Therefore a Jacobi-Richardson step reads as follows:

- Determine (generalised) Jacobi vector
- Collect sums of products such as $\sum \beta x$ and $\sum \beta^2$
- Solve the resulting matrix equation to . If we have only 1 dimension then this is a 2x2 matrix inverse; if there are 7 dimensions and 14 separate regressors then it is a 14x14 Lower-Upper decomposition, and double precision is sufficient for it.
- Update $x \mapsto x + \gamma_1 x_1 + \gamma_2 x_2$

===

Quickly recall: optimize the scaling of several directions.

$$\varepsilon_{new} = A\left(x + \gamma_1 x_1 + \gamma_2 x_2 + \gamma_3 x_3\right) - b = \varepsilon_{old} + \gamma_1 A x_1 + \gamma_2 A x_2 + \gamma_3 A x_3$$

Aim to minimise RSS

$$\begin{aligned} \frac{\partial \Sigma \varepsilon^2}{\partial \gamma_1} &= 0 = \Sigma\left(\varepsilon_{old} + \gamma_1 A x_1 + \gamma_2 A x_2 + \gamma_3 A x_3\right)_i \left(A x_1\right)_i \\ &= \Sigma \varepsilon_{old} \left(A x_1\right)_i + \gamma_1 \Sigma \left(A x_1\right)_i^2 + \gamma_2 \Sigma \left(A x_1\right)_i \left(A x_2\right)_i + \gamma_3 \Sigma \left(A x_1\right)_i \left(A x_3\right)_i \end{aligned}$$

So we have to solve

$$\begin{pmatrix} \sum \left(A x_1\right)_i \left(A x_1\right)_i & \sum \left(A x_1\right)_i \left(A x_2\right)_i & \sum \left(A x_1\right)_i \left(A x_3\right)_i \\ \sum \left(A x_1\right)_i \left(A x_2\right)_i & \sum \left(A x_2\right)_i \left(A x_2\right)_i & \sum \left(A x_2\right)_i \left(A x_3\right)_i \\ \sum \left(A x_1\right)_i \left(A x_3\right)_i & \sum \left(A x_2\right)_i \left(A x_3\right)_i & \sum \left(A x_3\right)_i \left(A x_3\right)_i \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{pmatrix} = \begin{pmatrix} -\sum \varepsilon_{old} \left(A x_1\right)_i \\ -\sum \varepsilon_{old} \left(A x_2\right)_i \\ -\sum \varepsilon_{old} \left(A x_3\right)_i \end{pmatrix}$$

Nothing special we can say except that it's a symmetric matrix.
Method of cofactors: what is overall determinant?

$$B_{11}\left(B_{22}B_{33} - B_{23}B_{32}\right) - B_{12}\left(B_{21}B_{33} - B_{23}B_{31}\right) + B_{13}\left(B_{21}B_{32} - B_{22}B_{31}\right)$$

Does symmetry really do anything?

$$B_{11}B_{22}B_{33} - B_{11}B_{23}^2 - B_{22}B_{13}^2 - B_{33}B_{12}^2 + 2B_{12}B_{23}B_{13}$$

Matrix of signed minors:

$$\begin{pmatrix} B_{22}B_{33} - B_{23}^2 & B_{23}B_{31} - B_{21}B_{33} & B_{21}B_{32} - B_{22}B_{31} \\ B_{13}B_{32} - B_{12}B_{33} & B_{11}B_{33} - B_{13}^2 & B_{12}B_{31} - B_{11}B_{32} \\ B_{12}B_{23} - B_{13}B_{22} & B_{13}B_{21} - B_{11}B_{23} & B_{11}B_{22} - B_{12}^2 \end{pmatrix}$$

Then transpose - no effect because symmetric - and divide by determinant.
Check inverse:

$$\begin{pmatrix} B_{22}B_{33} - B_{23}^2 & B_{23}B_{31} - B_{21}B_{33} & B_{21}B_{32} - B_{22}B_{31} \\ B_{13}B_{32} - B_{12}B_{33} & B_{11}B_{33} - B_{13}^2 & B_{12}B_{31} - B_{11}B_{32} \\ B_{12}B_{23} - B_{13}B_{22} & B_{13}B_{21} - B_{11}B_{23} & B_{11}B_{22} - B_{12}^2 \end{pmatrix} B = \left(B_{11}B_{22}B_{33} - B_{11}B_{23}^2 + B_{12}B_{23}B_{31} - B_{12}^2 B\right.$$

So then we get $\gamma_{1,2,3}$.
==========
For 2x2 the matrix of signed minors over the determinant:

$$\frac{\begin{pmatrix} B_{22} & -B_{21} \\ -B_{12} & B_{11} \end{pmatrix}}{B_{11}B_{22} - B_{12}^2}$$

FROM ANOTHER FILE Least Squares back of envelope
==========
Suppose there are 6 regressors $x_{1,\ldots,6}$ – in reality such as Jacobi for moving (A_x, A_y, A_z), and each of these has some effect on a vector of 7 residuals at each point.
Alternatively take 10 or fully 14 regressors; doesn't matter right here. 10 may be most practical.
We want to solve

$$\frac{\partial \sum \varepsilon_i^2}{\partial \gamma_1} = 0 = 2 \sum \varepsilon_i \left(\gamma_1, \gamma_2, \text{etc}\right) \frac{\partial \varepsilon_i}{\partial \gamma_1}$$

So let [if x_{1j} is the CHANGE in that regressor :]

$$\varepsilon_i = \varepsilon_{i,existing} + \gamma_1 \left(\sum \beta_{1ij} x_{1j}\right) + \gamma_2 \left(\sum \beta_{2ij} x_{2j}\right)$$

Yes - this is how it was done.
But note that here we have multiple ε_i at a location and these all fall into the list.
==
Then

$$0 = \sum \varepsilon_i^{new} \left(\sum \beta_{1ij} x_{1j}\right) = \sum \left(\varepsilon_{i,existing} + \gamma_1 \left(\sum \beta_{1ij} x_{1j}\right) + \gamma_2 \left(\sum \beta_{2ij} x_{2j}\right)\right) \left(\sum \beta_{1ij} x_{1j}\right)$$

$$0 = \sum \left(\varepsilon_{i,existing} + \gamma_1 \left(\sum \beta_{1ij} x_{1j}\right) + \gamma_2 \left(\sum \beta_{2ij} x_{2j}\right)\right) \left(\sum \beta_{2ij} x_{2j}\right)$$

$$\begin{pmatrix} \left(\sum \beta_{1ij} x_{1j}\right)\left(\sum \beta_{1ij} x_{1j}\right) & \left(\sum \beta_{1ij} x_{1j}\right)\left(\sum \beta_{2ij} x_{2j}\right) & . & . \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ . \\ . \end{pmatrix} = \begin{pmatrix} -\sum \varepsilon_{i,existing} \left(\sum \beta_{1ij} x_{1j}\right) \\ -\sum \varepsilon_{i,existing} \left(\sum \beta_{2ij} x_{2j}\right) \\ . \\ . \end{pmatrix}$$

Anyway, here we can apply LU to solve this small set of equations.
Last piece of puzzle.

==========

HMM. I am not sure this is the preferable way. We are adding increments ...
It could be OK as long as we regularly recalculate ε from scratch.
Alternatively go from zero. We might need to.

$$0 = \sum \left(\varepsilon_{i,0} + \gamma_1 \left(\sum \beta_{1ij} x_{1j} \right) + \gamma_2 \left(\sum \beta_{2ij} x_{2j} \right) \right) \left(\sum \beta_{1ij} x_{1j} \right)$$

Well, issue here: we are adjusting A, ϕ, υ by some amount. So need to then put

$$0 = \sum \left(\varepsilon_{i,0} + \left(\sum \beta_{1ij} \left(\phi_{1j}^{existing} + \gamma_1 x_{1j} \right) \right) + \gamma_2 \left(\sum \beta_{2ij} \left(... \right) \right) \right) \left(\sum \beta_{1ij} \left(\phi_{1j}^{existing} + \gamma_1 x_{1j} \right) \right)$$

This is just an awkward way round of doing same thing, starts with refreshing epsilon.
Only question then, how frequently to refresh.
We do refresh every step I think.

Why not Gauss-Seidel? Equivalently with the JRLS, we can apply the idea of least squares to the Gauss-Seidel method which involves systematically updating each value in turn to solve a particular equation. Here we take account of the sum of squares of affected residuals when we update a value. This method is thus based on trying to optimise the L_2 norm of the residuals by moving one value at a time.

This LSGS method was not seen to succeed on its own, though on occasion it can have some synergy with JRLS. It is not worth implementing just for performance, given the added complexity. But as a last resort can be worth trying if there is a barrier that JRLS cannot break through.

Jacobi Least Squares can be much more effective than LSGS because where the equations imply a strong "positive correlation" between two values, moving one at a time is far harder than moving them both in tandem. (In fact if ever circumcenters are used for the discretisation, they can be very close together and this can necessitate treating those pairs differently: first the values are assumed equal and then finally, their corresponding equations can be solved simultaneously with a 2x2 matrix for "Jacobi". However, this highlights why it is never a good idea to base the discretisation on circumcenters.)

========= notes:=========

- Rather than setting Dirichlet, we can choose to amalgamate the equations on the coarsest level, fitting them all into a matrix via least squares. (Topic, have to write about.)
- Note: We always need to use high precision (double-double or above) for the coarsest level LU.
- It proves rather difficult to get coarsest level convergence consistently without Dirichlet for ϕ. If we set a value to 0 in LU, we dropped an equation and get an error. If we don't, we get ϕ with 10^8 added and this means a poor fit due to the sensitivity of the equations. We can get good results by setting $\phi_{last} = 0$ during the "solve" step with the U matrix, but not perfect.
- For 3D we obviously do have a Dirichlet of sorts. We are going to have to set the cathode potential, in the same way we are now setting E_z, to achieve an overall current into the cathode.
- We proceed in 2D tri-based by assuming
  - $\phi_{ins} = 0$ when we form a quadrilateral next to an insulator vertex.
  - $\phi_{ins} = 0$ when we look into the insulator for $\nabla \cdot E$
  - However there is no current into the insulator whatever the value of E_r, because of resistivity.
  - I expect this will spoil $\Sigma\varepsilon$ but in SD maybe it still will converge.

10 Evolving velocity, temperature and ionisation

10.0.1 Introduction

The aim here is to take into account the fast intra-cell processes such as energy transfer between species, as (an inner) part of running inter-cell diffusion of heat and momentum. It became apparent that if we perform heat conduction separately to inter-species heat transfer, then we run the risk that a bottleneck develops. Let's suppose that in a cell, $n_{ion} > 100n_{neutral}$ but in its neighbour, $n_{neutral} > 100n_{ion}$. At high temperatures there can be a very fast flow of heat into the neighbour within the ion species, but also a very fast transfer of heat into neutrals in the neighbour. This means if the conduction and transfer are performed in sequence, we do not transfer enough heat. A valid result would be still found when the timestep is restricted so that there are not big changes in any temperature. However, that could be prohibitively short. If we perform the conduction and transfer together, we can see that heat is both entering and leaving the small ion mass at a high rate, and be able to use an appropriate shared substep for the joint procedure. Meanwhile it is also thoroughly clear that we also need to apply frictional heating at the same time as inter-species heat transfer: again because this latter can be a fast process, especially if the species experiencing heating is only a small proportion in the location.

Since the timestep for an explicit scheme including the inter-species transfer processes could be indefinitely short, these are handled implicitly. This is computationally far easier than making the rest implicit, as it only requires solving a 3x3 matrix for the species temperatures within each cell. Using this approach is certainly more feasible than trying to reckon with the problems of applying an all-explicit method to stiff equations. At the same time, we are trying to avoid the code complexity (and possibly long runtime) of making an overall implicit method that requires solving large-scale (non-)linear equations.

To handle the inter-cell processes, we treat the intra-cell advance as a black box with an acceleration or temperature advance routine which can be called by the inter-cell routine. The intra-cell routine receives a linear trajectory for the rate of change of v (or heat) due to inter-cell flux, and the inter-cell routine that supplies this trajectory can observe the endpoint t_{k+1} flux that follows from the results of the intra-cell acceleration.

The same principles are applied for v, with inter-cell momentum flux due to viscous forces, as for T, with inter-cell heat flux due to heat conduction. We still treat the two separately, incurring an error by neglecting their interaction. This error can be mitigated by looping through substeps of v advance and T advance during an overall simulation step.

The approach for the intra-cell routines The intra-cell routine for acceleration, with given linear trajectory of contributions to dv/dt due to inter-cell momentum flux, is quite straightforward, since there are linear acceleration equations and we can simply invert 3x3 matrices. Therefore I used Backward Improved Euler. For the intra-cell heat advance however, we include ionisation and recombination, to make sure a big error is not made by treating this separately. Because of the nonlinearity of ionisation (with its effect on temperature), this would require a costly root-finding method to do even a full Backward Euler step. It is easy to just use n_{k+1} as the multiplier for the time-integrated ionisation rate, to avoid running out of the converting species. But this is one reason we do not use semi-implicit: that advantage is lost and the step could fail easily. Instead we first try just an intra-cell adaptive timestep. If this fails, we need to consider using a root-finding method, as will be explained further.

The approach for the inter-cell routines The method to then also apply the inter-cell flows is meant to be a pretty robust method but is not implicit. It relies on the following reasoning: we want to find an interval where the actual trajectory, of the inter-cell flux, can be treated as linear, so basically that means we can linearly extrapolate (quasi-Taylor) from our initial position, and should then make the step short enough for that tangent to remain close to the actual curve. This is an

adaptive global timestep. The proximity of the inputted endpoint flux to the trajectory of the flux can be signalled by the proximity to the endpoint flux outputted by running the intra-cell routine.

From this basic idea there are several developments: since we can estimate the second derivative, we can use that to provide a better estimate of the endpoint flux; the second derivative also gives us an idea of what step to try where the distance of the trajectory from a linear approximation is small; a new estimate of it can be used in shortening timesteps. We can assume that outputted endpoint flux is modelled as a linear function of inputted, at least when the two are reasonably close and not moving far; this gives rise to a way of moving towards a fixed point.

10.0.2 Sequence of a simulation step

- [take pressure]
- Viscosity and intra-cell acceleration for time $h/2$ –> heat to add in each cell
- Heat conduction, ionisation and intra-cell heating for time $h/2$
- Feint ion, neutral move -> displacement(pressure)
- Position solve [+ compressive htg]
- Create new mesh and place fluids
- Feint acceleration for time $h/2$ –> $v_{k+1}^{ion}\left(v_{k+1}^{e}, E_{k+1}\right)$ –> Ohm's Law for $v_{k+1}^{e}\left(E_{k+1}\right)$
- Potential solver –> get $E_{k+1}, v_{k+1}^{e}, B_{k+1}$
- [take pressure]
- Viscosity and intra-cell acceleration for time $h/2$ –> heat to add in each cell
- Heat conduction, ionisation and intra-cell heating for time $h/2$

10.0.3 Alternative sequence

If we want E to change linearly instead of being piecewise constant in time, then the thing is that to accelerate from the start of the step, we have to already look to the end. The end E_{k+1} is determined on the new mesh. Therefore the new mesh has to be determined at the beginning of the step. Moreover, the ϕ is determined because of the relative electron move, which requires to know the ion move.

Therefore the alternative sequence looks something like:

- [take pressure at t_k]
- Feint ion, neutral move for time h -> get linear function: displacement(pressure)
- Position solve [+ compressive htg]
- Create new mesh and place fluids : already placed ion move for (t_k, t_{k+1}).
- Feint acceleration for time h –> $v_{k+1}^{ion}\left(v_{k+1}^{e}, E_{k+1}\right)$ –> Ohm's Law for $v_{k+1}^{e}\left(E_{k+1}\right)$
- Potential solver –> get $E_{k+1}, v_{k+1}^{e}, B_{k+1}$
- [take estimate of pressure at t_{k+1}, or just re-estimate intermittently]
- Viscosity and intra-cell acceleration for time h –> heat to add in each cell
- Heat conduction, ionisation and intra-cell heating for time h

10.1 The inter-cell routine, for advancing momentum and heat

- The outer routine that manages the inter-cell flux and treats the intra-cell routine as a black box, is roughly the same for both the case of momentum and of heat.

This outer loop is aiming to find and accept steps where a linear-in-time approximation to the inter-cell flux is supplied to the intra-cell black box, and two conditions are achieved:

1. The second-order error from $h^2 \frac{\partial^2}{\partial t^2}$ (flux) is small enough that a linear approximation to the flux trajectory is good.

2. The approximation to the final-time ('endpoint') inter-cell flux (of momentum or heat) is okay.

When we input a hypothetical endpoint flux to the intra-cell black box, and run acceleration (or advance T), we can then infer an outputted endpoint flux from the velocity or temperature obtained. The proximity of this output to the input is something we can use to infer that the input is correct: if things are well-behaved then the true value lies between the input and the output. This proximity also allows us to assess the departure from a linear trajectory of flux at the same time, if the input is coming from the tangent to the trajectory at time t_k.

Note to self.. It is a somewhat useful fact that if the change in v or T during the step is say, less than 10%, then we expect a predictor-corrector iteration to get us closer to the true endpoint flux. (For instance, whereas $\exp(-0.1) = 0.904837$, if we first go to 0.9 from 1.0 then we iterate: $1 - 0.05 * 1 - 0.05 * 0.9 = 0.905$.) This suggests a predictor-corrector (Heun-like) step can give us the ability to provide a linear input to the "black box" of intra-cell acceleration, for a suitably controlled timestep.

10.1.1 The procedure

1. Estimate of how to step forward – both how far and what endpoint flux – comes from initial knowledge of $\frac{df}{dt}, \frac{d^2 f}{dt^2}$ where f is the inter-cell flux. First we seek to obtain $\frac{df}{dt}$ empirically at t_k.

   (a) Set $h_{short} = \min\{10^{-3} h_{step}, 10^{-6} h_{sub}\}$ where h_{step} is the previous step, h_{sub} is the total step to be performed.

   (b) On the initial step, run $Internal\,(h_{short}, f_k, f_k)$. Generally, set $f_{project} = f_k + \frac{h_{short}}{h_{step}}(f_k - f_{k-1})$ and run $Internal\,(h_{short}, f_k, f_{project}) :\ v_k \mapsto v_k^{+short}$ to then infer f_k^{+short}.

   (c) Restore the previous system with v_k and infer the estimate of $\frac{df}{dt}_k$:

   $$\widehat{\frac{df}{dt}}_k = \frac{f_k^{+short} - f_k}{h_{short}}$$

   .

2. Now we estimate $\frac{d^2 f}{dt^2}$. For the initial step, take $\widehat{\frac{d^2 f}{dt^2}} = 0$. Otherwise, use

   $$\widehat{\frac{d^2 f}{dt^2}} = \frac{1}{0.5\,(h_{step} + h_{short})} \left(\widehat{\frac{df}{dt}}_k - \frac{f_k - f_{k-1}}{h_{step}} \right)$$

3. Compute subtimestep to first try for step, as follows.

   (a) Start from $h_{putative}$ either 1.2 times the previous h_{step} or initially, some fraction of h_{sub}.

(b) Limit step $h_{putative}$ to where, for all cells,

$$\left(\frac{df^{ion}}{dt}_k \cdot \frac{df^{ion}}{dt}_k\right) h_{put}^2 < P_{maxchg}^2 \left(f_k^{ion} \cdot f_k^{ion} + P_{avg}\overline{[f_k^{ion} \cdot f_k^{ion}]}\right)$$

$$\left(\frac{df^{neut}}{dt}_k \cdot \frac{df^{neut}}{dt}_k\right) h_{put}^2 < P_{maxchg}^2 \left(f_k^{neut} \cdot f_k^{neut} + P_{avg}\overline{[f_k^{neut} \cdot f_k^{neut}]}\right)$$

and for heat only, (since we do not apply acceleration and viscosity for electrons),

$$\left(\frac{df^{e}}{dt}_k \cdot \frac{df^{e}}{dt}_k\right) h_{put}^2 < P_{maxchg}^2 \left(f_k^{e} \cdot f_k^{e} + P_{avg}\overline{[f_k^{e} \cdot f_k^{e}]}\right)$$

where

$$\overline{[f_k^s \cdot f_k^s]} = \frac{1}{N_{cells}} \sum_{cells} f_k^s \cdot f_k^s$$

and if any of these constraints was binding, reduce h_{put}^2 by a further safety factor 0.8. P_{maxchg} is 0.05 and P_{avg} is 0.015.

(c) Shrink further to get an integer number of remaining steps. Set $N_{steps\ remain} = \text{ceiling}\left[\left(h_{sub} - t_k\right)/h$ set $h_{put} = \left(h_{sub} - t_k\right)/N_{steps\ remain}$.

4. If this is not the initial step, shorten the step further to ensure that we do not think there is a large departure from a linear trajectory. Reduce h_{step} from h_{put} to where for each species, all cells,

$$\left(\frac{d^2 f^{species}}{dt^2}_k \cdot \frac{d^2 f^{species}}{dt^2}_k\right) \frac{h_{step}^4}{16} < P_{maxerr}^2 \left(f_k^{species} \cdot f_k^{species} + P_{avg}\overline{\left[f_k^{species} \cdot f_k^{species}\right]}\right)$$

and $P_{maxerr} = 5 \times 10^{-5}$. For the initial step just let $h_{step} = h_{put}$.

5. Now project to a guess for the endpoint flux at $t_k + h_{step}$:

$$f_{k+1}^{project} = f_k + h_{step}\frac{df}{dt}_k + \frac{h_{step}^2}{2}\frac{d^2 f}{dt^2}_k$$

recalling that initially, we estimate $\frac{d^2 f}{dt^2}$ as 0.

6. Attempt to advance a step, running $Internal\left(h_{step}, f_k, f_{k+1}^{project}\right) :\ v_k \mapsto v_{k+1}^{feint} \Rightarrow f_{k+1}^{output}$.

(a) For each species and for all cells, we require

$$\left(f_{k+1}^{project} - f_{k+1}^{output}\right)^2 < P_{maxend}^2 \left(f_{k+1}^{output} \cdot f_{k+1}^{output} + P_{avg}\overline{[f_k \cdot f_k]}\right)$$

and if this is violated somewhere, multiply h_{step}^2 by the ratio

$$\left(P_{maxend}^2 \left(f_{k+1}^{output} \cdot f_{k+1}^{output} + P_{avg}\overline{[f_k \cdot f_k]}\right)\right) / \left(\left(f_{k+1}^{project} - f_{k+1}^{output}\right)^2\right)$$

where $P_{maxend} = P_{maxerr}/2$.

(b) If we have to reduce h_{step}, multiply h_{step}^2 by safety factor 0.8 and shrink to get an integer number of steps remaining. We then have to run again.

(c) Each time we run again, let

$$f_{k+1}^{project} = f_k + h_{step}\frac{df}{dt}_k + \frac{h_{step}^2}{2}\frac{d^2 f}{dt^2}_k$$

7. [Optional]. To get a closer fit to the true flux trajectory than just by now treating $f_{k+1} = f_{k+1}^{output}$, we can now try to improve as follows:

   (a) Run step again using $Internal\left(h_{step}, f_k, f_{k+1}^{output}\right)$ and from this infer $f_{k+1}^{out^2} := g_2$.

   (b) Create λ^s for each species, for blending, as explained below: calling $f_2 = f_{k+1}^{output}$, $f_1 = f_{k+1}^{project}$,

$$\lambda^{species} = -\frac{\sum \left(f_{2i} - g_{2i}\right)\left(f_{1i} - 2f_{2i} + g_{2i}\right)}{\sum \left(f_{1i} - 2f_{2i} + g_{2i}\right)^2}$$

   (c) For each species let $f_{k+1}^{blend} = \lambda f_{k+1}^{project} + (1-\lambda) f_{k+1}^{output}$. Make step $Internal\left(h_{step}, f_k, f_{k+1}^{blend}\right)$ and finally infer f_{k+1}.

   Note that so far when doing this, λ is observed to be positive and very small. In other words, f_{k+1}^{output} is an excellent approximation to the best endpoint flux, at least most of the time.

8. Now that a step has been accepted, accumulate the frictional heat into a storage array. Accumulate viscous heating from

$$\frac{h_{step}}{2}\left(\text{heat rate}\left(f_k\right) + \text{heat rate}\left(f_{k+1}\right)\right)$$

   which is possible since both f_k and f_{k+1} were computed by the momentum flux calculation routine and not just extrapolated.

9. Return to step 1 until h_{sub} has been exhausted.

10.1.2 Try again 23/03/16

The results of the above are that the step will remain fairly short, like 10^{-15}, for a long time initially, even allowing a 20% change in flux, as v and therefore flux are very small.

I think the best thing is to drop the requirement for a small change in flux, and instead look only for linearity. This means f'' must be estimated including on the 1st step.

I then increased the max error threshold for departure from linear, up to 1%, and the endpoint flux error threshold to 0.1%. Now it drops back dramatically to 6e-18 due to the latter, but then recovers to 5e-12.

Probably the initial behaviour could be much improved by introducing a quadratic approx to where the endpt trajectory departed.

^^ This was now done.

10.1.3 Criterion for excessive curvature : explanation

Suppose a function is modelled over linear and its values at the ends of an interval of length h are known perfectly. Then the model for $f\left(h/2\right)$ is

$$f\left(h/2\right)^{linear} = \frac{1}{2}f_0 + \frac{1}{2}\left(f_0 + hf_0' + \frac{h^2}{2}f'' = f_1\right)$$

whereas in fact

$$f\left(h/2\right) = f_0 + \frac{h}{2}f_0' + \frac{1}{2}\frac{h^2}{4}f''$$

So for this error to be less than 5% + a small absolute threshold, we desire

$$\frac{h^2}{8}f'' < 0.05\left(f_0 + \frac{h}{2}f_0' + \frac{h^2}{8}f''\right) + \text{absthresh}$$

For which it is sufficient:

$$\frac{h^2}{8}f'' < 0.05\left(f_0 + \frac{h}{2}f_0'\right) + \text{absthresh}$$

By making h small, this will rapidly become the case.

The exception is when the linear approximation is close to zero but there is a small hump in reality. To help with this we can use

$$\frac{h^2}{8}f'' < 0.05\left(\frac{f_0 + f_1}{2} + 0.01\left[\text{typical } f\right]\right) + \text{absthresh} \tag{301}$$

adding a small extra (e.g. 1% of typical) to the right hand side.

How do we handle this for 3-vector f?

$$\frac{h^4}{64} < \left(\frac{1}{f''\cdot f''}\right)\left(0.05^2\left(\left(\frac{f_0+f_1}{2}\right)^2 + 0.01\left[\text{typical } f^2\right]\right) + \text{absthresh}^2\right)$$

or

$$\frac{h^4}{64}f''\cdot f'' < \left(0.05^2\left(\left(f_0 + \frac{h}{2}f_0'\right)^2 + 0.01\left[\text{typical } f^2\right]\right) + \text{absthresh}^2\right)$$

$$\frac{h^4}{64}f''\cdot f'' < \left(0.05^2\left(f_0^2 + hf_0f_0' + \frac{h}{4}^2 f_0'^2 + 0.01\left[\text{typical } f^2\right]\right) + \text{absthresh}^2\right)$$

since $f_0'^2 > 0$ this is given if we have

$$\frac{h^4}{64}f''\cdot f'' < \left(0.05^2\left(f_0^2 + min\left(hf_0f_0', 0\right) + 0.01\left[\text{typical } f^2\right]\right) + \text{absthresh}^2\right)$$

The max h to satisfy this is given by ? Use $h_{existing}$ for the hf_0f_0' and then repeat with the calculated $h_{max} < h_{existing}$ or go the other side of it.

10.1.4 λ-blend of two inter-cell flux arrays: explanation

The flux array is a high-dimensional object. However there are good reasons to think that we can use a special property: if we have a deviation ε from the true endpoint flux in our input endpoint flux, then the output will tend to lie in an opposing direction. Therefore it is reasonable to search for a better solution on the line between the input and output.

In general, assume we make 2 sensible feints from f_1, f_2. Then assume - which must become approximately true if f_1, f_2 are sufficiently close:

$$g\left(\lambda f_1 + (1-\lambda)f_2\right) = \lambda g_1 + (1-\lambda)g_2$$

We seek to minimize

$$\sum\left(\lambda\left(f_{1i} - g_{1i}\right) + (1-\lambda)\left(f_{2i} - g_{2i}\right)\right)^2$$

$$\frac{d\Sigma}{d\lambda} = 0 = 2\sum\left(\lambda\left(f_{1i} - g_{1i}\right) + (1-\lambda)\left(f_{2i} - g_{2i}\right)\right)\cdot\left(f_{1i} - g_{1i} - f_{2i} + g_{2i}\right)$$

$$\lambda\sum\left(f_{1i} - g_{1i} - f_{2i} + g_{2i}\right)^2 + \sum\left(f_{2i} - g_{2i}\right)\cdot\left(f_{1i} - g_{1i} - f_{2i} + g_{2i}\right) = 0$$

$$\lambda = -\frac{\sum\left(f_{2i} - g_{2i}\right)\cdot\left(f_{1i} - g_{1i} - f_{2i} + g_{2i}\right)}{\sum\left(f_{1i} - g_{1i} - f_{2i} + g_{2i}\right)^2}$$

In the case that $f_2 = g_1$:

$$\lambda = -\frac{\sum\left(f_{2i} - g_{2i}\right)\left(f_{1i} - 2f_{2i} + g_{2i}\right)}{\sum\left(f_{1i} - 2f_{2i} + g_{2i}\right)^2} \tag{302}$$

10.2 Intra-cell acceleration routine and viscous flux

As explained, we have to consider that we need to do actual acceleration at the same time as viscous momentum transfer between cells. Therefore we have still two things to consider: how to determine the momentum fluxes given a set of velocity values, and, intra-cell how to evolve velocity given the net rate of inflow or outflow.

- Note that so far, we take the electron velocity to be either given or an unknown. It would certainly be possible to run with electron velocity included in order to produce a linear relationship similar to an Ohm's Law, dynamically. However this requires then to avoid the assumption that E is constant, or the results will be unfavourable: the sign of the effect of E will be indeterminate. Dropping the assumption that E is constant during a half-step is fairly challenging (but could be worthwhile).

10.2.1 Intra-cell given input from outside

We begin by writing a linear equation for the six-vector $\begin{pmatrix} v_{ion} \\ v_{neutral} \end{pmatrix}$. Let a suitably defined transfer rate be denoted r_m and for electrons and ions combined, $r_m^{combined}$. Then the acceleration equation is a stiff linear ODE:

$$\begin{aligned} \frac{dv}{dt} &= \begin{pmatrix} -r_{in}^{combined} & 0 & -\omega_{iy} & r_{in}^{combined} & 0 & 0 \\ 0 & -r_{in}^{combined} & \omega_{ix} & 0 & r_{in}^{combined} & 0 \\ \omega_{iy} & -\omega_{ix} & -r_{in}^{z} & 0 & 0 & r_{in}^{z} \\ r_{ni} & 0 & 0 & -r_{ni} & 0 & 0 \\ 0 & r_{ni} & 0 & 0 & -r_{ni} & 0 \\ 0 & 0 & r_{ni} & 0 & 0 & -r_{ni} \end{pmatrix} v + \begin{pmatrix} a_{ion,x}^{TP} + \omega_{iy} v_{ez} + \nu_{ie} v_{ex} \\ a_{ion,y}^{TP} - \omega_{iy} v_{ez} + \nu_{ie} v_{ey} \\ \frac{q}{M} E_z + \nu_{ie} v_{ez} \\ a_{neut,x}^{TP} + \nu_{ne} v_{ex} \\ a_{neut,y}^{TP} + \nu_{ne} v_{ey} \\ \nu_{ne} v_{ez} \end{pmatrix} \\ &:= Fv + a^0 \end{aligned} \tag{303}$$

- However, we need to do more than this: we also need to take account of an inward flux of momentum at the same time. If this is taken as a given, we can put the rate of velocity increase into the a^0 vector.

Alternatively, we could of course rewrite the equivalent for momentum.

10.2.2 Backward Improved Euler

We can write down a second-order fully backward scheme by supposing that we make a backward half-step from t_{k+1} to $t_{k+1/2}$, using that mid-time system state to infer the derivative for the step from t_k to t_{k+1}. This method I call Backward Improved Euler. This can be reduced for our linear case as follows:

$$X_{k+1} - X_k = hf\left(X_{k+1/2}, t_{k+1/2}\right) = h\left(FX_{k+1/2} + a_{k+1/2}^0\right) \tag{304}$$

$$X_{k+1} - X_{k+1/2} = \frac{h}{2}\left(FX_{k+1} + a_{k+1}^0\right) \tag{305}$$

$$X_{k+1} - X_k = h\left(F\left(X_{k+1} - \frac{h}{2}a_{k+1}^0 - \frac{h}{2}FX_{k+1}\right) + a_{k+1/2}^0\right) \tag{306}$$

$$\left(1 - hF + \frac{h^2}{2}F^2\right) X_{k+1} = X_k + ha_{k+1/2}^0 - \frac{h^2}{2}Fa_{k+1}^0 \tag{307}$$

$$X_{k+1} = \left(1 - hF + \frac{h^2}{2}F^2\right)^{-1}\left(X_k + ha_{k+1/2}^0 - \frac{h^2}{2}Fa_{k+1}^0\right) \tag{308}$$

This allows a timestep that does not have to be constrained by the rate of inter-species velocity equilibration.

10.2.3 Compute viscous momentum flux rate between cells for each species

To infer the rate of momentum flow, of each dimension, through a cell wall, we have to estimate ∇v, for each dimension, at the wall. A gradient of v_z can stimulate a transfer of p_x or p_y.

It turns out it appears more efficient to actually do the whole computation each time than to pre-calculate a set of coefficients that can take us from ∇v to the flux of p, even at each vertex. It takes about 10 times as much effort to create the pre-calculated coefficients as to work with an actual ∇v.

For ions:

1. At each edge, compute $\eta_{\parallel}, \eta_{\perp}, \eta_{\times}$ that were defined in subsection 4.2. The values to apply at the edge can be averaged from the surroundings.
2. Over each cell edge, calculate ∇v, recalling subsubsection 8.2.1.
3. Thereby calculate ∇v in magnetic coordinates, ie $\frac{\partial v_b}{\partial \perp}$ etc. using eqn (251).
4. Calculate the rate-of-strain tensor W.
5. Calculate the magnetic-coordinates momentum flux tensor $\Pi_{b\perp\wedge}$, per subsection 4.2.
6. Transform to the Cartesian momentum flux tensor Π_{xyz} per eqn (249).
7. We therefore infer the ion momentum flux across this edge: $\mathfrak{flux}\left(p_x\right) = \Pi_{xx} \perp_x^{edge} + \Pi_{xy} \perp_y^{edge}$ where $\perp^{edge}$ is a vector normal to the edge and as long as the edge.
8. This contributes to a stored array representing the rate of change of v in each cell due to the inter-cell momentum flux, referred to above as f. We can do this because cell mass is not changing:
$$\frac{dv_x}{dt} = \frac{1}{N_{ion} m_{ion}} \frac{dp_x}{dt}$$
.

For neutrals: We assume that
$$\eta_{neutral} = \frac{2}{3} \kappa_{neutral}$$
and as mentioned there are rival definitions for $\kappa_{neutral}$. For now I took the greater one. Then
$$\mathfrak{flux}\left(p_z^{neut}\right) = -\eta_{neutral} \left(\nabla v_n\right) \cdot \perp^{edge} \tag{309}$$
and as for ions, this allows us to accumulate the net rate of change of v^{neut} in each cell.

10.2.4 Viscous heating

When in the viscous inter-cell momentum flux calculation routine, we also have to work out viscous heating. If done any other way we run much more risk of encountering viscous cooling which is then some trouble to get rid of. By looking at the viscous heating effects of each cell wall flux, we stand the best chance of seeing positive heating every time, since notwithstanding magnetism, I expect it is still typical that momentum will only flow in the direction that decreases directed kinetic energy in the two cells.

In fact,
$$\begin{aligned} \frac{\partial}{\partial t}\left[\text{DKE}\right] &= m N_1 v_1 \left.\frac{\partial v_1}{\partial t}\right|_{flux} + m N_2 v_2 \left.\frac{\partial v_2}{\partial t}\right|_{flux} \\ &= v_1 \left.\frac{\partial p_1}{\partial t}\right|_{flux} + v_2 \left.\frac{\partial p_2}{\partial t}\right|_{flux} = \mathfrak{flux}_{1\to 2}\left(p_x\right)\left(v_2 - v_1\right) \end{aligned} \tag{310}$$

so if the amount of heat denominated NT is desired, this is

$$\frac{\partial}{\partial t}\left(\frac{3}{2}N_1T_1+\frac{3}{2}N_2T_2\right)=-\frac{\partial}{\partial t}\left[\mathrm{DKE}\right]=\mathfrak{flux}_{1\to 2}\left(p_x\right)\left(v_1-v_2\right) \tag{311}$$

which as noted, I hope to be usually positive. (If not, discard.)

The viscous heat rate has to be stored in a separate array and in the event of a substep being accepted, the outer routine has to accumulate heat into the permanent storage array ready to be applied in the heating steps.

10.3 Intra-cell heat and ionisation routine, and inter-cell heat flux calculation

As noted, we create a routine for intra-cell temperature advance, which takes into account a linear trajectory for influx of heat from the neighbouring cells. Inter-species heat transfer may need to be treated implicitly, since it may be fast. It is unfortunately non-trivial to write an overall implicit step because of the nonlinear formulae for ionisation / recombination rates.

Let N_s be the number of species s particles in the cell. We seek here to take multiple processes into account simultaneously, for heat:

- inter-cell heat flux contribution which takes the form of 3 rates $H_s^{\mathrm{Intercell}}=\frac{\partial}{\partial t}\left(N_sT_s\right)$ following a linear path between given values $\frac{\partial}{\partial t}\left(N_sT_s\right)\left(t_k\right),\ \frac{\partial}{\partial t}\left(N_sT_s\right)\left(t_{k+1}\right)$. Thermoelectric drift will be included in this, as well as diffusive heat flux. (The species advection effect and compressive heating are treated separately.)
- frictional and viscous heating, which takes the form of a given amount to be added to N_sT_s during a full run of the 'outer' calling routine, giving rise to a rate H_s^{FV} by dividing by the full interval time length.
- inter-species heat transfer.
- heat changes due to ionisation and recombination.

It is logical to work in heat and not temperature because, since we could not anticipate number of ions at t_{k+1} from outside this routine, many of our additional quantities are denominated in heat. Even though ionisation rate is a function of temperature, it is nonlinear anyway.

10.3.1 Intra-cell heat and ionisation routine: preliminaries

The inter-cell heating and frictional/viscous heating are added:

$$\begin{aligned} H_s\left(t\right) &= H_s^{\mathrm{FV}}+H_s^{\mathrm{Intercell}}\left(t_k\right)+\frac{t-t_k}{t_{k+1}-t_k}\left(H_s^{\mathrm{Intercell}}\left(t_{k+1}\right)-H_s^{\mathrm{Intercell}}\left(t_k\right)\right) &(312)\\ &:= H_s^0+\left(t-t_k\right)\frac{\partial H_s}{\partial t} &(313)\end{aligned}$$

For the inter-species transfer, collect a heat transfer rate matrix U: firstly note that

$$\frac{\partial T_s}{\partial t}=\sum_{\beta\neq s}2M_{s\beta}\nu_{s\beta}^{\mathrm{MT}}\left(T_s-T_\beta\right) \tag{314}$$

where

$$M_{in}=\frac{m_nm_i}{\left(m_n+m_i\right)^2} \tag{315}$$

That is,

$$\frac{\partial}{\partial t}\begin{pmatrix} T_n \\ T_i \\ T_e \end{pmatrix} = \begin{pmatrix} -2M_{in}\nu_{ni}^{\mathrm{MT}} - 2M_{en}\nu_{ne}^{\mathrm{MT}} & 2M_{in}\nu_{ni}^{\mathrm{MT}} & 2M_{en}\nu_{ne}^{\mathrm{MT}} \\ 2M_{in}\nu_{in}^{\mathrm{MT}} & -2M_{in}\nu_{in}^{\mathrm{MT}} - 2M_{ie}\overline{\nu_{ie}} & 2M_{ie}\overline{\nu_{ie}} \\ 2M_{en}\nu_{en}^{\mathrm{MT}} & 2M_{ie}\overline{\nu_{ei}} & -2M_{ie}\overline{\nu_{ei}} - 2M_{en}\nu_{en}^{\mathrm{MT}} \end{pmatrix}\begin{pmatrix} T_n \\ T_i \\ T_e \end{pmatrix} \tag{316}$$

Therefore

$$\frac{\partial}{\partial t}\begin{pmatrix} N_nT_n \\ N_iT_i \\ N_eT_e \end{pmatrix} = \begin{pmatrix} -2M_{in}\nu_{ni}^{\mathrm{MT}} - 2M_{en}\nu_{ne}^{\mathrm{MT}} & 2M_{in}\frac{N_n}{N_i}\nu_{ni}^{\mathrm{MT}} & 2M_{en}\frac{N_n}{N_e}\nu_{ne}^{\mathrm{MT}} \\ 2M_{in}\frac{N_i}{N_n}\nu_{in}^{\mathrm{MT}} & -2M_{in}\nu_{in}^{\mathrm{MT}} - 2M_{ie}\overline{\nu_{ie}} & 2M_{ie}\frac{N_i}{N_e}\overline{\nu_{ie}} \\ 2M_{en}\frac{N_e}{N_n}\nu_{en}^{\mathrm{MT}} & 2M_{ie}\frac{N_e}{N_i}\overline{\nu_{ei}} & -2M_{ie}\overline{\nu_{ei}} - 2M_{en}\nu_{en}^{\mathrm{MT}} \end{pmatrix}\begin{pmatrix} N_nT_n \\ N_iT_i \\ N_eT_e \end{pmatrix}$$

However, note that

$$\frac{N_e}{N_i}\overline{\nu_{ei}} = \frac{n_e}{n_i}\overline{\nu_{ei}} = \overline{\nu_{ie}} \tag{317}$$

and so

$$\begin{aligned}\frac{\partial}{\partial t}\begin{pmatrix} N_nT_n \\ N_iT_i \\ N_eT_e \end{pmatrix} &= \begin{pmatrix} -2M_{in}\nu_{ni}^{\mathrm{MT}} - 2M_{en}\nu_{ne}^{\mathrm{MT}} & 2M_{in}\nu_{in}^{\mathrm{MT}} & 2M_{en}\nu_{en}^{\mathrm{MT}} \\ 2M_{in}\nu_{ni}^{\mathrm{MT}} & -2M_{in}\nu_{in}^{\mathrm{MT}} - 2M_{ie}\overline{\nu_{ie}} & 2M_{ie}\overline{\nu_{ei}} \\ 2M_{en}\nu_{ne}^{\mathrm{MT}} & 2M_{ie}\overline{\nu_{ie}} & -2M_{ie}\overline{\nu_{ei}} - 2M_{en}\nu_{en}^{\mathrm{MT}} \end{pmatrix}\begin{pmatrix} N_nT_n \\ N_iT_i \\ N_eT_e \end{pmatrix} \\ &:= U\begin{pmatrix} N_nT_n \\ N_iT_i \\ N_eT_e \end{pmatrix}\end{aligned} \tag{318}$$

Label $\left.\frac{\partial(NT)}{\partial t}\right|_{\mathrm{I+R}}$ the rate of NT change due to ionisation and recombination. Then in total here,

$$\frac{\partial}{\partial t}\begin{pmatrix} N_nT_n \\ N_iT_i \\ N_eT_e \end{pmatrix} = \begin{pmatrix} H_n^0 + (t-t_k)\frac{\partial H_n}{\partial t} \\ H_i^0 + (t-t_k)\frac{\partial H_i}{\partial t} \\ H_e^0 + (t-t_k)\frac{\partial H_e}{\partial t} \end{pmatrix} + \left.\frac{\partial}{\partial t}\right|_{\mathrm{I+R}}\begin{pmatrix} N_nT_n \\ N_iT_i \\ N_eT_e \end{pmatrix} + U\begin{pmatrix} N_nT_n \\ N_iT_i \\ N_eT_e \end{pmatrix} \tag{319}$$

10.3.2 Evolution method: some RK2, some Backward Euler

Method 1 : Attempt Improved Euler

The approach for an attempted second-order step is as follows:

- If possible just use a fully explicit RK2 step.
- If there are problems, try implicit linear rearrangements.
- If the step is deemed to have failed, try making the timestep shorter.
- Ultimately, we may need a way of crashing out of the routine with a failure code, if behaviour is too unfavourable given the inputted $H_s^{\mathrm{Intercell}}(t_{k+1})$.

Write the rate of ionising of neutrals:
and let the rate of recombining be, of whichever of {electrons, ions} is in the minority:

Getting a suitable half-step In what follows, t_k refers to the t we have already reached and t_{k+1} refers to a further substep — ie different notation from having (t_k, t_{k+1}) for the whole step – should come back and change this.

1. Before looping, calculate $n_k^e S_k(T_k^e)$, $n_k R_k = max\{n_k^e, n_k^i\} R_k(T_k^e)$ where

$$n^e S = n^e \frac{10^{-5} T_{e[eV]}^{1/2} \exp\left(-\frac{E_0}{T_{e[eV]}}\right)}{E_0\left(6E_0 + T_{e[eV]}\right)} \tag{320}$$

$$nR = max\{n_e, n_i\}\left(n_e 8.75 \times 10^{-27} T_{e[eV]}^{-4.5}\right) \tag{321}$$

and compute inter-species heat transfer matrix U.

2. Loop to make half-step:

   (a) Calculate the amounts (or densities) that will be ionised and recombined in time $h_{step}/2$, based on the system at time t_k:

$$n_{ionise} = n_k^n \frac{h_{step}}{2} n_k^e S_k$$

$$n_{recombine} = min\left\{n_k^e, n_k^i\right\} \frac{h_{step}}{2} n_k R_k$$

$$n_{1/2}^n = n_k^n - n_{ionise} + n_{recombine}$$

   If that will make us run out of one species, $n_{1/2} < 0$, instead take

$$n_{ionise} = n_k^n \left(\frac{0.5 h_{step} n_k^e S_k}{1 + 0.5 h_{step} \left(n_k^e S_k + n_k R_k\right)} \right) \tag{322}$$

$$n_{recombine} = min\left\{n_e, n_i\right\} \left(\frac{0.5 h_{step} n_k R_k}{1 + 0.5 h_{step} \left(n_k^e S_k + n_k R_k\right)} \right) \tag{323}$$

   which signifies using the $n_{1/2}$ values to multiply the rates.

   (b) The corresponding heat change for the amounts of ionisation $N_{ionise} = Area \cdot n_{ionise}$, $N_{recombine} = Area \cdot n_{recombine}$ is

$$\left(\begin{array}{c} N_n T_n \\ N_i T_i \\ N_e T_e \end{array} \right) += \left(\begin{array}{c} -N_{ionise} T_k^n + N_{recombine} \left(T_k^e + T_k^i + \frac{m_i}{3} \left\|v_i - v_n\right\|_k^2 + \frac{2}{3} k_B E_\infty^0\right) \\ -N_{recomb} T_k^i + N_{ionise} \left(\frac{T_k^n}{2} + \frac{m_i}{3} \left\|v_i - v_n\right\|_k^2\right) \\ N_{recomb} \left(-T_k^e + \frac{m_e}{3} \left\|v_e - v_n\right\|_k^2\right) + N_{ionise} \left(\frac{T_k^n}{2} + \frac{m_e}{3} \left\|v_e - v_n\right\|_k^2 - \frac{2}{3} k_B E_\infty^0\right) \end{array} \right) = \tag{324}$$

   (c) Form

$$\left(\begin{array}{c} N_n T_n \\ N_i T_i \\ N_e T_e \end{array} \right)_{1/2} = \left(\begin{array}{c} N_n T_n \\ N_i T_i \\ N_e T_e \end{array} \right)_k + \frac{h_{step}}{2} \left(\begin{array}{c} H_n\left(t_k\right) \\ H_i\left(t_k\right) \\ H_e^0\left(t_k\right) \end{array} \right) + \left(\begin{array}{c} D_n \\ D_i \\ D_e \end{array} \right) + \frac{h_{step}}{2} U \left(\begin{array}{c} N_n T_n \\ N_i T_i \\ N_e T_e \end{array} \right)_k \tag{325}$$

   (d) If any of these $N_s T_s < 0$, try again with implicit heat soak:

$$\left(1 - \frac{h_{step}}{2} U\right) \left(\begin{array}{c} N_n T_n \\ N_i T_i \\ N_e T_e \end{array} \right)_{1/2} = \left(\begin{array}{c} N_n T_n \\ N_i T_i \\ N_e T_e \end{array} \right)_k + \frac{h_{step}}{2} \left(\begin{array}{c} H_n\left(t_k\right) \\ H_i\left(t_k\right) \\ H_e^0\left(t_k\right) \end{array} \right) + \left(\begin{array}{c} D_n \\ D_i \\ D_e \end{array} \right) \tag{326}$$

   (e) Test each of the species: we want (for some chosen constants $\alpha_{1,2}$)

$$\begin{aligned} \left(T_{1/2} - T_k\right)^2 &< \alpha_1^2 \left(T_k^2 + \alpha_2^2 \left(T_k^{avg}\right)^2\right) \\ \left(T_{1/2} - T_k\right) &> -T_k/2 \end{aligned}$$

   so repeat with smaller timestep if this is not being achieved.

3. Calculate

$$v_{1/2} =$$

4. Now we make a step with $t_{1/2}$ as midpoint. Compute $n^e_{1/2}S_{1/2}$,$n_{1/2}R_{1/2}$. Try, for the amount of ionisation during a step of length h_{step},

$$N_{ionise} = N^n_{1/2} n^e_{1/2} S_{1/2} h_{step}$$

$$N_{recombine} = min\left\{n^e_{1/2}, n^i_{1/2}\right\} n_{1/2} R_{1/2} h_{step}$$

But if we ran out of a species, instead put

$$N_{ionise} = N^n_{1/2} \frac{n^e_{1/2} S_{1/2} h_{step}}{1 + n^e_{1/2} S_{1/2} h_{step} + n_{1/2} R_{1/2} h_{step}}$$

$$N_{recombine} = min\left\{n^e_{1/2}, n^i_{1/2}\right\} \frac{n_{1/2} R_{1/2} h_{step}}{1 + n^e_{1/2} S_{1/2} h_{step} + n_{1/2} R_{1/2} h_{step}}$$

5. Then we can compute the change of NT due to these:

$$\begin{pmatrix} N_n T_n \\ N_i T_i \\ N_e T_e \end{pmatrix} += \begin{pmatrix} -N_{ionise} T^n_{1/2} + N_{recombine}\left(T^e_{1/2} + T^i_{1/2} + \frac{m_i}{3}\left\|v_i - v_n\right\|^2_{1/2} + \frac{2}{3}k_B E^0_\infty\right) \\ -N_{recomb} T^i_{1/2} + N_{ionise}\left(\frac{T^n_{1/2}}{2} + \frac{m_i}{3}\left\|v_i - v_n\right\|^2_{1/2}\right) \\ N_{recomb}\left(-T^e_{1/2} + \frac{m_e}{3}\left\|v_e - v_n\right\|^2_{1/2}\right) + N_{ionise}\left(\frac{T^n_{1/2}}{2} + \frac{m_e}{3}\left\|v_e - v_n\right\|^2_{1/2} - \frac{2}{3}k_B E^0_\infty\right) \end{pmatrix} = \tag{327}$$

$$\begin{pmatrix} N_n T_n \\ N_i T_i \\ N_e T_e \end{pmatrix}_{k+1} = \begin{pmatrix} N_n T_n \\ N_i T_i \\ N_e T_e \end{pmatrix}_k + h_{step} \begin{pmatrix} H_n\left(t_{k+1/2}\right) \\ H_i\left(t_{k+1/2}\right) \\ H_e\left(t_{k+1/2}\right) \end{pmatrix} + \begin{pmatrix} D_n \\ D_i \\ D_e \end{pmatrix}_{1/2} + h_{step} U \begin{pmatrix} N_n T_n \\ N_i T_i \\ N_e T_e \end{pmatrix}_{1/2} \tag{328}$$

If this fails with $T < 0$, we try putting $h_{step}U$ on the left:

$$(1 - h_{step}U) \begin{pmatrix} N_n T_n \\ N_i T_i \\ N_e T_e \end{pmatrix}_{k+1} = \begin{pmatrix} N_n T_n \\ N_i T_i \\ N_e T_e \end{pmatrix}_k + h_{step} \begin{pmatrix} H_n\left(t_{k+1/2}\right) \\ H_i\left(t_{k+1/2}\right) \\ H_e\left(t_{k+1/2}\right) \end{pmatrix} + \begin{pmatrix} D_n \\ D_i \\ D_e \end{pmatrix}_{1/2} \tag{329}$$

6. We can do tests:

   (a) for $T < 0$. If this happens, it might be a sign that there is a problematic input.

   (b) for net ionisation rate: if $\frac{dn}{dt}_{k+1} / \frac{dn}{dt}_k > -0.95$ then we could re-run with a shorter timestep.

We repeat such steps until h_{sub} is exhausted.

10.3.3 Possibility for 3rd order Heun:

We could then, in the case that the step is successful, ie we did not have to make any compromises that damage the order of accuracy, make a 3rd-order Heun step. This means

$$X_{k+3/2} = X_k + \frac{3}{2}h\left(\frac{1}{4}\frac{dX}{dt}_k + \frac{3}{4}\frac{dX}{dt}_{k+1}\right) \tag{330}$$

10.3.4 Failsafe method

If the above does not succeed, it does not look like semi-implicit will probably succeed either. Therefore I will, in the case that the explicit method is taking too long, default to solving for a Backward Euler or another unconditionally stable step (Backward Improved Euler).

1. Solve the ∂T equations to give T as a linear function of n_{ionise}/n^e_{k+1} and $n_{recombine}/n^n_{k+1}$, which are nonlinear functions of n_{ionise} and $n_{recombine}$.

2. For each value of T_e there is an implied value of n_{ionise} and $n_{recombine}$. Therefore using the solved-out ∂T equations it (nonlinearly) implies a value for itself. We now want to solve for the fixed point of this mapping.

3. I have thought of a simple 1-dimensional derivative-free root-finding method which seems like the number of evaluations should be fairly low compared to bisection:

   (a) Given an interval containing T_e, infer from a line between the two endpoint $f(T_e)$, where $f(T_e)$ crosses zero. This becomes the next marker.

   (b) Whichever side now contains T_e, try to pick a point which is a short distance the other side of zero from our solved point. If this succeeds, it forms the new interval and we go again. If not, the new interval is obvious.

10.3.5 Getting second order implicit – for reference

We can treat 1st order Backward Euler RK[-1] as a black box from the perspective of a cell flux Runge-Kutta, to use RK4 or any other method, but this will not get us to a higher order overall. It would have the advantage that we could work into it a reevaluation of κ as part of the "derivative evaluation" in RK.

We could do likewise with 2nd order Backward Improved Euler RK[-2] internal to the cells. However to get 2nd order overall, we would then have to also find a way with RK[-2] to handle the cross-effects between the inter-species and inter-cell flows.

There is another approach. Given the first order step, we can try to achieve second order using Richardson extrapolation, cognate with the Lawson-Swayne method or the Lawson-Morris-Gourlay method.

- A time-honoured system for getting second-order stable steps: apply Richardson extrapolation. If we run 2 first order steps vs 1 first order step, then

  $$Y = 2Y_2 - Y_1$$

  is second order. In the case that both Y_1 and Y_2 are Backward Euler steps, this method is called **Lawson-Morris-Gourlay**.

- Take an implicit step of $\frac{h}{2+\sqrt{2}}$ then again from that step $\frac{h}{2+\sqrt{2}}$. Then $Y_{k+1} = Y_2 + \sqrt{2}(Y_2 - Y_1)$. In this case that the foundational steps are Backward Euler, this method is called **Lawson-Swayne** (Check correct before using.)

10.3.6 Inter-cell heat flux

Now note that for electrons there is also a thermoelectric effect which will go into our calculated inter-cell flux. Basically we are addressing the following equations:

$$\frac{\partial T_n}{\partial t} += \frac{2}{3n_n}\nabla\cdot(\kappa^n\nabla T_n) \tag{331}$$

$$\frac{\partial T_i}{\partial t} += \frac{2}{3n_i}\nabla\cdot\left(\kappa^{\text{ion}}\nabla T_i\right) \tag{332}$$

$$\frac{\partial T_e}{\partial t} = \frac{2}{3n_e}\nabla\cdot(\kappa^e\nabla T_e) - \frac{1}{n_e}\nabla\cdot\left(n_e T_e\frac{\bar{\nu_{ei}}}{\nu_{e\heartsuit}}\Upsilon_{e\heartsuit}(v_e - v_i)\right) \tag{333}$$

- Therefore the heat conduction routine also includes the thermoelectric effect. But if currents in the plane are not being included then this may be rather limited.

- We are forced to model heat flow rather than the effect on $\frac{\partial T}{\partial t}$ because since ionisation will be applied at the same time, species mass is not constant.

The conductivities were:

$$\kappa^{\text{ion}} = \kappa_{\parallel}^{\text{ion}} \Upsilon_{i\heartsuit}^{+} = \frac{5}{2} \frac{n_i T_i}{m_i \nu_{i\heartsuit}} \Upsilon_{i\heartsuit}^{+} \tag{334}$$

$$\kappa^{e} = \kappa_{\parallel}^{e} \Upsilon_{e\heartsuit} = \frac{5}{2} \frac{n_e T_e}{m \nu_{e\heartsuit}} \Upsilon_{e\heartsuit} \tag{335}$$

In practice this means

$$\kappa^{ion} = \frac{\kappa_{\parallel}^{ion}}{\nu_{i\heartsuit}^2 + \omega_{ci}^2} \begin{pmatrix} \nu_{i\heartsuit}^2 + \omega_{ci.x}^2 & \omega_{ci.x}\omega_{ci.y} + \nu_{i\heartsuit}\omega_{ci.z} \\ \omega_{ci.x}\omega_{ci.y} - \nu_{i\heartsuit}\omega_{ci.z} & \nu_{i\heartsuit}^2 + \omega_{ci.y}^2 \end{pmatrix} \tag{336}$$

$$\kappa^{e} = \frac{\kappa_{\parallel}^{e}}{\nu_{e\heartsuit}^2 + \omega_{ce}^2} \begin{pmatrix} \nu_{e\heartsuit}^2 + \omega_{ce.x}^2 & \omega_{ce.x}\omega_{ce.y} - \nu_{e\heartsuit}\omega_{ce.z} \\ \omega_{ce.x}\omega_{ce.y} + \nu_{e\heartsuit}\omega_{ce.z} & \nu_{e\heartsuit}^2 + \omega_{ce.y}^2 \end{pmatrix} \tag{337}$$

For neutrals, take either

$$\kappa_{[\text{GSKB}]}^{\text{neutral}} = 10 \frac{n_n T_n}{m_n \nu_{n\heartsuit}}$$

or per [Vallance1]

$$\kappa^{n} = \frac{1}{2} \frac{n_n T_n}{m_n \left(\nu_{nn}^{\eta} + \nu_{in}^{\eta} + \nu_{en}^{\eta} \right)}$$

As with the viscous momentum flux, we examine each cell edge in turn and decide on the flux of heat across that edge, recalling (284).

10.3.7 Previous ionisation and recombination method - for reference

When I wrote the following robust method for a nonlinear ODE, I was quite proud of it. However, again it is just trying to get robustness and relying on the overall timestep for accuracy. Actually in the specified initial conditions, which are far from ionisation equilibrium, a timestep such as $h = 10^{-14}$ is necessary or we do get visible ridges due to solving for the ionisation roughly. It will now make more sense to run a subcycle in each cell (perhaps with this method since it's already implemented) that has some respect for the ionisation rate timescale. Indeed, it makes sense to combine ionisation (and other processes that induce inelastic friction) with the Ohmic subcycle routine.

^^

That remark is also out of date. The following description is good for getting the basic idea.

We can make ionisation a subroutine of the heat routine to ensure that we do not incorrectly favour soaking heat to neutrals over ionisation.

Put ionisation and heat soaking together – great – by using Taylor-implicit ionisation and recombination rate, or say 2nd-order Taylor-implicit Te. We can show on a spreadsheet that assuming the step is limited to a 3% change in Te then the 2nd order expansion about the t_k time-derivative will be good for the range of parameters. A step up from the semi-implicit approach.

This a really good plan.

==========

The model equations

$$\frac{\partial n_{\text{ion}}}{\partial t} = \frac{\partial n_{\text{e}}}{\partial t} = n_{\text{e}} n_{\text{n}} S\left(T_{e[eV]}\right) - n_i n_{\text{e}} \left(\alpha_r \left(T_{e[eV]}\right) + n_e \alpha_3 \left(T_{e[eV]}\right) \right) \tag{338}$$

For the purposes of having new values on a cell, all we need to do is calculate the new ionised fraction

$$\lambda = \frac{n_i}{n_i + n_n} = \frac{n_e}{n_i + n_n}$$

(in practice we may allow $n_e \approx n_i$ but this is an unnecessary detail here).

Then

$$\frac{\partial \lambda}{\partial t} = \frac{\partial n_e}{\partial t}/n_{tot} = \lambda n_{tot}\left(1-\lambda\right) S\left(T_{e[eV]}\right) - \lambda^2 n_{tot}\left(\alpha_r\left(T_{e[eV]}\right) + \lambda n_{tot}\alpha_3\left(T_{e[eV]}\right)\right) \tag{339}$$

where

$$S = 10^{-5}\sqrt{T_{e[eV]}}\frac{\exp\left(-\frac{E_0}{T_{e[eV]}}\right)}{E_0\left(6E_0 + T_{e[eV]}\right)}$$

$$\alpha_r = 2.7 \times 10^{-13}/\sqrt{T_{e[eV]}}$$

$$\alpha_3 = 8.75 \times 10^{-27}/T_{e[eV]}^{9/2}$$

A forward Euler step can easily overshoot the equilibrium and indeed exit $(0,1)$. Any explicit RK method may sometimes fail if the disequilibrium situation is sufficiently harsh, so if we chose to upgrade to e.g. RK4 classic in the usual case, we still need to allow a default in the case that it fails. There is a complicated nonlinear equation if we try to solve for the equilibrium.

We have to take into account the coupling with T_e. We said in Revamp_model7.lyx that in the net ionisation case (where $t > t_k$)

$$\begin{aligned} T_e\left(t\right) &= \frac{2}{3}\frac{\frac{3}{2}n_{e,k}T_{e,k} + \left(\frac{3}{2}\frac{T_{n,k}}{2} - 13.6\right)\left(n_e\left(t\right) - n_{e,k}\right)}{n_e\left(t\right)} \\ &= T_{e,k}\frac{\lambda_k}{\lambda\left(t\right)} + \left(\frac{T_{n,k}}{2} - \frac{2}{3}13.6\right)\frac{\lambda\left(t\right) - \lambda_k}{\lambda\left(t\right)} \end{aligned} \tag{340}$$

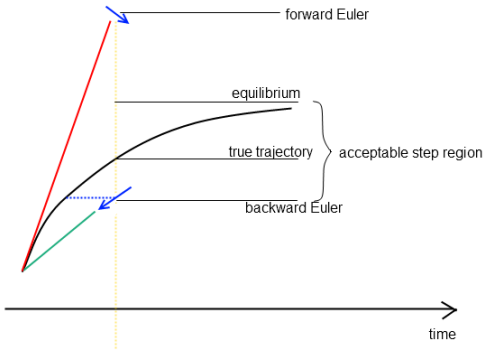
and in the net recombination case

$$T_e\left(t\right) = T_{e,k} - \frac{2}{3}13.6\frac{\lambda\left(t\right) - \lambda_k}{\lambda\left(t\right)} \tag{341}$$

Substituting the expressions for $T_e\left(t\right)$ in $S, \alpha_{r,3}$ gives rise to a nonlinear 1-dimensional ODE for $\lambda\left(t\right)$. e.g. without simplifying,

$$\begin{aligned} \frac{\lambda_{k+1}^{bwdT} - \lambda_k}{h} &= \lambda n_{tot}\left(1-\lambda\right)10^{-5}\left(T_{e,k}\frac{\lambda_k}{\lambda} + \left(\frac{T_{n,k}}{2} - \frac{2}{3}13.6\right)\frac{\lambda - \lambda_k}{\lambda}\right)^{1/2}\frac{\exp\left(-\frac{E_0}{T_{e,k}\frac{\lambda_k}{\lambda} + \left(\frac{T_{n,k}}{2} - \frac{2}{3}13.6\right)\frac{\lambda-\lambda_k}{\lambda}}\right)}{E_0\left(6E_0 + T_{e,k}\frac{\lambda_k}{\lambda} + \left(\frac{T_{n,k}}{2} - \frac{2}{3}13.6\right)\right.} \\ &-\lambda^2 n_{tot}2.7 \times 10^{-13}\left(T_{e,k}\frac{\lambda_k}{\lambda} + \left(\frac{T_{n,k}}{2} - \frac{2}{3}13.6\right)\frac{\lambda - \lambda_k}{\lambda}\right)^{-1/2} \\ &-\lambda^3 n_{tot}^2 8.75 \times 10^{-27}\left(T_{e,k}\frac{\lambda_k}{\lambda} + \left(\frac{T_{n,k}}{2} - \frac{2}{3}13.6\right)\frac{\lambda - \lambda_k}{\lambda}\right)^{-9/2} \end{aligned}$$

To solve for a backward Euler step, this nonlinear equation would require us to use a root-finding method. However, consider that we expect there to be a monotonic progress towards the equilibrium from above and below. Therefore there is an alternative way to get a robust step with time-local error $O\left(h^2\right)$ that allows us to accept values in an interval: see figure.



The Backward Euler value will lie on the same side of equilibrium as the initial value of λ (and in fact, since $d\lambda/dt$ reduces as we approach equilibrium, the true solution value will lie between the Backward Euler value and the equilibrium). We know that the solution trajectory should be stable (and qualitatively right) if we do not overshoot the equilibrium and we know that anything between a backward and forward Euler step has local error $O(h^2)$ in the limit $h \to 0$.

Therefore we develop the following algorithm for locating a value that is supposed to lie in the interval between Backward Euler and equilibrium.

1. Try a forward Euler step (or other explicit RK method).

$$\lambda_{putative} = \lambda_k + h\frac{d\lambda}{dt}(t_k)$$

2. If $\lambda_{putative} < 0$ or $\lambda_{putative} > 1$ then explicit failed - go to step 3. Otherwise calculate $T_e(\lambda_{putative})$. If $T_e(\lambda_{putative}) < 0$ then explicit failed - go to step 4. Now calculate $\frac{d\lambda}{dt}(\lambda_{putative}, T_e(\lambda_{putative}))$. ~~If this has the opposite sign to $\frac{d\lambda}{dt}(t_k)$ then explicit failed - go to step 4.~~ Changed 16/05/15: We do not demand it is the same sign, just that if it changed signs, then the magnitude is somewhat reduced.

3. Otherwise, we accept $\lambda_{k+1} = \lambda_{putative}$. The end.

4. We split out 2 cases but for brevity consider the net ionisation case. Create the initial search interval: $\lambda_{high} = \min(1, \lambda_{putative}), \lambda_{low} = \lambda_k$. Now loop: bisect to get $\lambda_{test} = \frac{1}{2}(\lambda_{high} + \lambda_{low})$.

5. Calculate $T_e(\lambda_{test})$. If $T_e(\lambda_{test}) < 0$ then we are overshooting equilibrium - take $\lambda_{high} = \lambda_{test}$ and bisect again; back to step 4.

6. Calculate $\frac{d\lambda}{dt}(\lambda_{test}, T_e(\lambda_{test}))$. If it is the opposite sign to $\frac{d\lambda}{dt}(t_k)$ then we are overshooting equilibrium - take $\lambda_{high} = \lambda_{test}$ and bisect again; back to step 4.

7. Now test whether we are beyond Backward Euler. If

$$\lambda_{test} - h\frac{d\lambda}{dt}(\lambda_{test}, T_e(\lambda_{test})) > \lambda_k + \varepsilon_{threshold}$$

   then we accept $\lambda_{k+1} = \lambda_{test}$. Otherwise, set $\lambda_{low} = \lambda_{test}$ and bisect again; back to step 4.

Once we have λ_{k+1}, to perform the updates as detailed in Revamp_model5.lyx is straightforward.

Note that if we ever did need to apply Brent's method, it looks like $1/\lambda$ might be the more suitable variable to use.

10.4 CUDA notes

- Compute inter-species coefficient in cells. Based on what? It may be constant throughout.
- Compute κ 2x2. (Return here after small step.) Then do what with it? Store back to global? We want to avoid recalculating if we change step length—??
- Coefficients for creating flux from T can be loaded to registers, not shared.
- Shared memory contains only T – ?? and should probably also contain for vertices. So this is 1.5 variables only. So we can still get away with 512 threads per block.
- 1.5 x 8 = 12. 48000/12 = 4000 cells worth of data that we could load. Do load 512*(3+1+3) and do rolling rows so that we can reduce the amount of fetches to shared. Rolling rows ftw! It's slightly harder when the threads more than exhaust the number of cells.

- We need positions in order to calculate gradient, and hence given κ, the flux rate given T. Whether that is at same time or not, we have to therefore have 1.5 x 2 for that. 2000 cells could be stored of that.
- What about kappa? We could calculate multiple times from each place but that seems a little silly. Can we, having got κ (4 vars??) each cell, re-store for each edge and thereafter load into registers? As long as we store twice for each edge.
- Coefficients for matrix within cell: these should involve registers.

10.5 Performing feints

Sometimes we want to not advance v but instead generate a linear relationship, either $v\,(E, v_e)$ or displacement as a function of thermal pressure acceleration rate a^{TP}. For feints we do not try to include viscous effects, but in each cell advance the coefficients for the desired linear equation.

11 Mesh structure and maintenance

11.1 Data structure [mostly cut and paste from older document -likely o o d]

We envisage a scheme in which the essential stored data takes the form of $\{\text{mass}, \text{momentum}, \text{heat}\}$ stored on each cell for each species. We use different units for each species: the value we store for "mass" is $\int_{cell} n$, the total number of particles (per vertical cm) in the cell. The value we store for "momentum" is the integral $\int_{cell} nv$, so that an estimate of velocity comes from $\int_{cell} nv / \int_{cell} n$, and "heat" is $\int_{cell} nT$. We take mass and energy to be conserved quantities in most of the procedures we are going to apply. The cells are all triangles and the mesh is maintained as a Delaunay mesh given the vertex positions; vertices are never created or destroyed. Vertex positions must lie within the domain - they are wrapped across the periodic boundary and bounced away from the insulator if they ever hit it. In general we move the vertices with the fluid, and occasionally we jump or swim them to better track the bulk mass. The reason for this approach is

1. Be capable of focusing information on the area of interest. Towards the end, the area of interest is thought to be 50 micron wide. So we might for instance go from 50 micron typical cell width down to 0.5 micron.
2. Resolve it well when there are narrow walls of plasma and avoid a lot of the smearing we would have by flowing across a mesh (this was seen to be bad).
3. We use a triangle mesh anyway - I tried to use a square mesh for about a year and with the reflective curved boundary it caused big problems every time. This is unfortunate, because square mesh and Gaussian go well together.

Obviously separate species meshes would have some of these advantages more fully - using a single mesh is a practical halfway house. Not using separate meshes avoids having to find overlaps when doing frictional procedures. Separate species meshes require quite a lot of bookkeeping. My effort with particles was written with separate species meshes and that did prove to be a lot of work.

11.2 Mesh maintenance

11.2.1 Mesh reconstruction

We identify overlaps by finding where two triangles share an edge and the unshared vertices are both on the same side of it. When an overlap is found, we pick a vertex to disconnect from the mesh by

trying to find which of the 4 points is in the interior of some triangle (to do this efficiently, we do a radar search moving outwards through triangle neighbours). We keep disconnecting vertices in this way until all overlaps are gone.

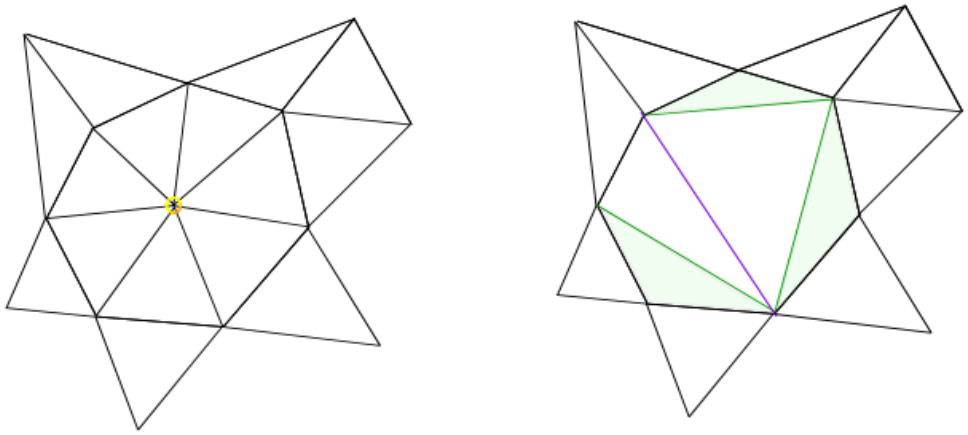
To disconnect a vertex VertDisco and connect its neighbours, we proceed as follows (several unsuccessful approaches were tried first):

- First establish that the neighbouring vertices form 1 circle if linked by the adjacent triangles. So far this has always been the case. As we do so, infer the periodic winding number of each vertex (-1,0,1) relative to VertDisco. Note that for the purpose of this circle, we can have a proxy vertex INS_VERT or HIGH_VERT if some of the adjacent 'triangles' are quadrilateral wedges.

- Get rid of (apply disconnect algorithm to) any neighbour vertices that have only 3 neighbours. Since two adjacent such vertices would give a closed mesh, we know that this should not result in an infinite loop. (For the same reason, viewing the vertex disconnections as happening in sequence, we must be left with at least 3 vertex neighbours for our vertex VertDisco.) This is so that we should be left with a convex polygon when we remove VertDisco.

- For each vertex in the circle of neighbours, remove each of the affected triangles from its list.

- Now we go around attempting to make triangles from linked triples around the edge of our circle.



Suppose such a candidate triangle is (a, b, c) s.t. (a, x, b) and (b, x, c) were triangles containing x =VertDisco.

- If (a, b, c) are the only neighbours, then we accept the triangle. (This in principle could cause errors; and they may be avoidable with extra care in this neighbour connection algorithm.)

  - Test whether (a, y, c) is a triangle that exists for some y, bearing in mind a or c could be proxy vertices.
  - If (a, y, c) exists, we reject (a, b, c) and move on to considering (b, c, d). We add a to the list of 'unconnected points'.
  - If (a, y, c) did not exist, we accept (a, b, c) and next consider (c, d, e). In other words, we start from the next point not covered by a triangle. We add a, c to the list of 'unconnected points'.
  - We continue this way until we get back to the beginning, ensuring that if (a_0, a_1, a_2) was not accepted that we do try (a_{-1}, a_0, a_1).
  - As long as there are at least 3 'unconnected points', we start again and repeat this loop. Wherever a point has been covered over by a triangle, a new triple is now available to consider.

- As we set the triangles we were able to use our periodic winding numbers to set the periodicity of the triangle. We now proceed to recalculate transverse vectors for edge testing, and reset the triangle neighbour lists.

- If there were N_{tri} triangles adjacent to VertDisco, we shall have connected its neighbours with $N_{tri} - 2$ triangles. Mark the two remaining triangles in memory as scrapped; we shall re-use them for reconnecting a vertex. Mark our vertex VertDisco as now disconnected.

Once all overlaps are gone, we reconnect the disconnected vertices, in their same locations of course, by dividing the triangles that they inhabit into 3 triangles each time.

11.2.2 Mesh re-delaunerisation

The Re-Delaunerisation algorithm examines each pair of joined triangles and considers whether a Delaunay flip is necessary. We cycle through every triangle T and compare it to each of its neighbours T_1. We test whether the unshared vertex q of T_1 is within the circumcircle of T. If so, we need to flip. Note that since triangles can be periodic, we make sure to create an unwrapped image of both triangles on one side.

11.2.3 Vertex swimming and jumps

As well as maintaining a mesh that is overlap-free and Delaunay, we want to ensure that the mesh follows the plasma. We particularly wish to avoid having cells that have only a small amount of mass.

One way to do this is to "swim" the vertices: we reposition them, recreate the mesh, and then map the cell data across in a way that is similar to a standard advection move (described below) but with zero drift.

How a vertex swimming move takes place: We take the objective to be to minimize the sum of squared cell masses, which we call $\Omega = \sum m_i^2$.

We wish to ensure that every single vertex repositioning will yield a reduction in this objective function, so we partition the set of vertices into subsets which we shall call volleys. Each volley contains only points that are not neighbours. Then for each volley, we perform a candidate move, and populate the cell masses. We then only accept those moves that yielded an improvement. One could easily envisage a scheme where several candidate moves are considered at once; however this requires quite a bit more storage.

To perform a candidate vertex move, we proceed as follows:

1. Take the gradient of the objective function. This is estimated from

$$\frac{d\Omega}{dx} = \frac{d}{dx}\sum m_i^2 = 2\sum m_i \frac{dm_i}{dx}$$

$$\frac{dm_i}{dx} = n^{tot}\frac{d}{dx}Area_i$$

   where m_i is the mass for the ith cell that adjoins this vertex and n^{tot} is the total density at the vertex. So as the vertex sweeps out an additional area to add or subtract from a cell, we assume that the vertex density is the density for that area.

2. We then normalise $\nabla\Omega$ and set a putative magnitude for the candidate move, in direction $-\nabla\Omega$:

$$x_{new} = x + t\frac{\nabla\Omega}{\|\nabla\Omega\|}$$

First let $\frac{d}{dt}$ represent the rate of change given unit movement of the vertex in the $\nabla\Omega$ direction, and solve a linearised equation to try to set $\frac{d}{dt}\Omega = 0$:

$$m_i \approx m_{i,k} + \frac{dm_i}{dt}_k t$$

so set

$$\sum \left(m_{i,k} + \frac{dm_i}{dt}_k t \right) \frac{dm_i}{dt}_k = 0$$

which is achieved from

$$t = -\frac{\sum m_{i,k} \frac{dm_i}{dt}_k}{\sum \left(\frac{dm_i}{dt}_k\right)^2}$$

This will generally be negative.

$$\frac{dm_i}{dt}_k = \frac{dm_i}{dx} \left(\frac{\nabla\Omega}{\|\nabla\Omega\|} \right)_x + \frac{dm_i}{dy} \left(\frac{\nabla\Omega}{\|\nabla\Omega\|} \right)_y$$

3. Then we also make sure that the magnitude is not greater than $\frac{1}{3}$ of the minimum side length for any of the adjoining cells.

4. We discard the move if it is in such a direction that the cell with least mass is having its area decreased.

5. Finally we have to verify that this move is not taking our vertex outside the adjoining cells, which could yet happen. To do this ...

It is seen that acceptance rates are reasonable. The moves do a fair job of reducing the overall SD of masses, which is their aim, but the global minimum cell mass does not always improve - only sometimes. Therefore we likely need a separate procedure aimed specifically at increasing the minimum cell mass.

I have also implemented a more radical procedure aimed at dealing with meshes that are far from being sensible - vertex jumps. We make a list of the "least useful" vertices - say 10% of the total number - where usefulness is defined as the average of the neighbouring masses (?). We make a list of the "most needful" triangles. We then move through the lists making comparisons between the least useful and most needful. If a vertex is jumped then its neighbours will not be jumped.

To jump a vertex, we disconnect it from its neighbours (then linking the neighbours as in the section above) and reconnect it in the centre of the most needful triangle. We then run the re-delaunerisation procedure when vertex jumping is completed.

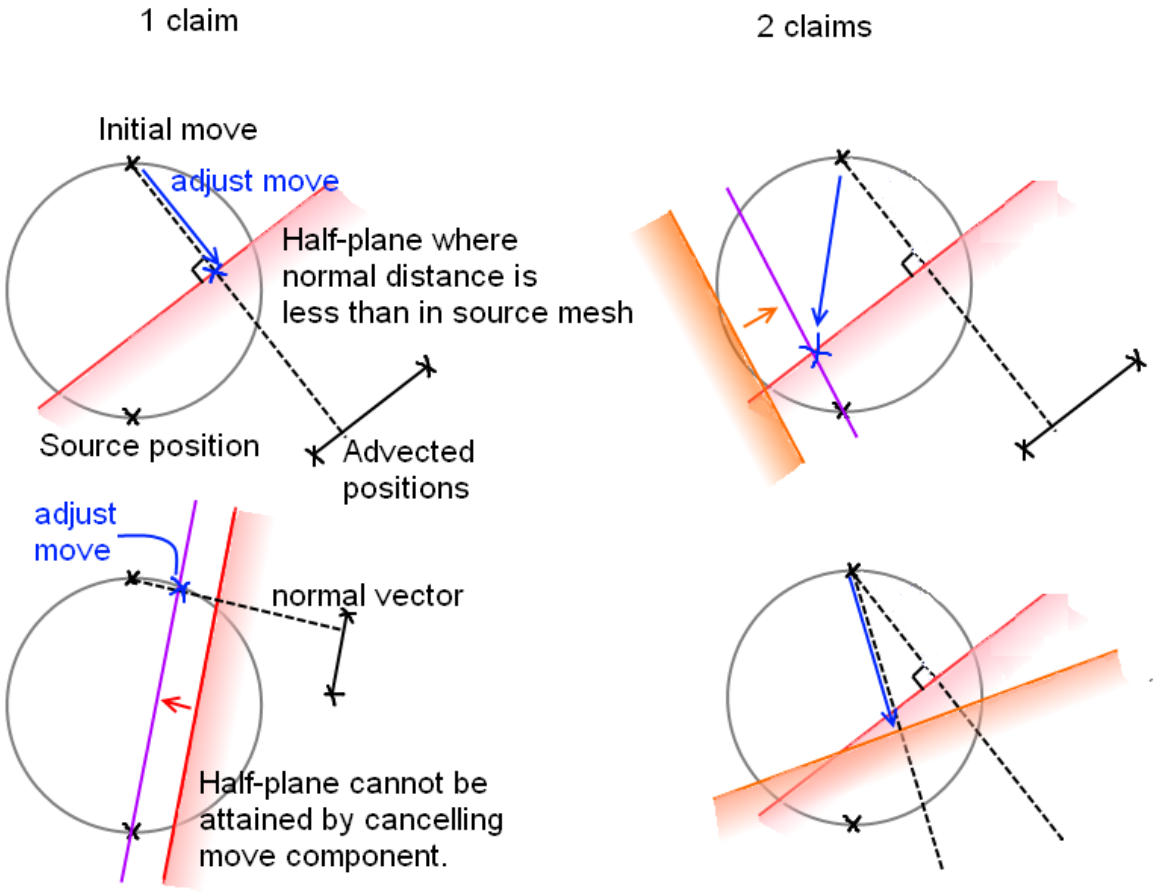
Whether for jumping or swimming, we use the same cell repopulation algorithm as during advection: create triplanar models of data and map planar triangles on to the new mesh, inferring the integrals of planar variables over the intersections. It would be reasonable to just insert the mesh modification steps into a system advance as a t_{k+1} mesh is being created.

11.3 Limiting vertex advection

Cells can get too large for sensible field solving if we let them. Therefore we have to control their size, which we do by either tracking back the mesh vertices or preventing them from moving in certain ways. The latter is preferred, to avoid shuffling of the mesh once the constraint is in effect.

We do not limit the species advection, of course, but only the advection of the mesh itself. Therefore this creates an Eulerian move, relative to the mesh which is eventually held fixed.

The procedure is as follows.



1. For each domain vertex (this here does not include insulator vertex) store original putative move. To begin with set the proposed move equal to the original move.

2. For each domain vertex, examine triangles for this vertex, in the source mesh.

   (a) By using the opposite edge normal unit vector, find the normal distance. For any that have normal distance beyond threshold:

   (b) Calculate normal distance in destination mesh, both for the proposed move and the original putative move.

   (c) If the normal distance for the original move is greater than the source distance * $(1 + 10^{-7})$ and

   $$(\text{initial move}) \cdot \perp^{dest} < -10^{-9}$$

   so that moving against the initial move, will reduce the normal distance, then this triangle is added to the list of viable claims.

   (d) If the normal distance for the existing proposed move is greater than the source distance * $(1 + 10^{-7})$ and

   $$(\text{existing move}) \cdot \perp^{dest} < -10^{-9}$$

   then mark this vertex as needing attention.

3. For each marked vertex, see how many viable claims exist:

   (a) If there is 1 claim, then the new proposed position is found by moving normally from the original putative position, until either the source mesh normal distance is restored, or, perpendicular to the normal vector we are parallel with the unadvected position.

   (b) If there are 2 claims, we create two lines representing half-plane edges. They are advanced, normally for their triangles, as far as the unadvected position. Then we should

      i. Test moving normally from the original putative advected position to each of these half-planes in turn: does moving to 1 of them put us inside the other? If so, accept this as the new proposed position.

      ii. Find the intersection of the two lines. This is the new proposed position. (It may be the original unadvected position.)

   (c) If there are 3 or more claims, just try to directly cancel the original move vector until we are within all of the half-planes, or the move is reduced to zero. This gives the new proposed position.

4. If any vertices had to be repositioned, return to step 2.

Note that intentionally, this algorithm means that the accessible locus is a disk with both the original unadvected position and the original putative advected position on opposite edges. This is what we can reach by assuming that when we amend our move, we only delete a component of the original move vector. That is, on picking a unit direction d, the maximum extent of move is $-(\text{initial move})\cdot d$, the projection of our initial move on to that direction.

Part III
Analysis of results

References

[McDaniel1949] E. McDaniel. , http://www.virginia.edu/ep/Interactions/McDaniel%20-%20Fundamental%20concepts%20of%20collisions.pdf

[NRLFormulary] D. Book, J. D. Huba. NRL Plasma Formulary. 2009.

[Reif2010] F. Reif. Fundamentals of Statistical and Thermal Physics. 2010

[Hewett1985] D. W. Hewett. Elimination of electromagnetic radiation in plasma simulation: the Darwin or magnetoinductive approximation. Space Science Reviews 42,1. pp 29-40. 1985.

[Golant1977] Golant, Zhilinsky, Sakharov, Brown. Fundamentals of Plasma Physics. Wiley, 1980. (Izdatel'stvo Atomizdat 1977)

[Braginskii1965] S.I.Braginskii, Transport Processes In a Plasma. 1965.

[KrallTrivelpiece1973] Krall, Trivelpiece. 1973.

[BL91] C.K. Birdsall, A.B. Langdon. Plasma Physics via Computer Simulation. (IoP series in plasma physics) Taylor and Francis, 2005 (hardback 1991).

[MT] M. V. Tretyakov, G. N. Milstein. Stochastic Numerics for Mathematical Physics.

[Vallance1] http://vallance.chem.ox.ac.uk/pdfs/PropertiesOfGasesLectureNotes.pdf

[VallanceReaction] http://vallance.chem.ox.ac.uk/pdfs/ReactionDynamicsLectureNotes2012.pdf

[NatPhysLab] http://www.kayelaby.npl.co.uk/general_physics/2_2/2_2_4.html

[SolarSystemSchool] http://www.solar-system-school.de/lectures/space_plasma_physics_2007/Lecture_4

[OakRidge64] http://web.ornl.gov/info/reports/1964/3445605367741.pdf p.191-192, 195

[OakRidgeTableDD+] http://www-cfadc.phy.ornl.gov/elastic/ddp/tel-DP.html

[OakRidgeTableDD] http://www-cfadc.phy.ornl.gov/elastic/dd0/tel.html

[CollisionTable] http://course1.winona.edu/wng/Tables/Collision-properties.htm

[OakRidgeElastic] http://www-cfadc.phy.ornl.gov/elastic/elastic0.html

[OakRidgeHH+] http://www-cfadc.phy.ornl.gov/elastic/ApJ2008-dir/elastic.dat

[jspf] http://www.jspf.or.jp/PFR/PDF/pfr2010_05-S2032.pdf

[TKBose] High Temperature Gas Dynamics By Tarit K. Bose

[MeyerDesbrun2002] Meyer, Desbrun, Schroder, Barr. Discrete differential geometry operators for triangulated 2-manifolds. International Workshop on Visualization and Mathematics, Berlin, Germany, 2002. http://www.multires.caltech.edu/pubs/diffGeoOps.pdf

[PinkallPolthier93] Pinkall, Polthier 1993: http://page.mi.fu-berlin.de/polthier/articles/diri/diri_jem.pdf [Results treating the Dirichlet energy.]

[DesbrunMeyer00] Desbrun, Meyer, Schroder, Barr: Discrete Differential-Geometry Operators in nD, 2000: http://www.gvu.gatech.edu/people/official/jarek/graphics/papers/09DiscreteOpsDes [note in particular Appendix A.]

[LXZ08] Liu, Xu, Zhang: A discrete scheme of Laplace-Beltrami operator and its convergence over quadrilateral meshes. Computers & Mathematics with Applications. Vol 55, Issue 6, March 2008. http://www.sciencedirect.com/science/article/pii/S0898122107005731 [mentions a few different 'discretised Laplacians'.]

[Murray] Glenn Murray. inside.mines.edu/fs_home/gmurray/ArbitraryAxisRotation/

[TK94] Taylor, Kriegman. Yale Technical Report 9405, year 1994.

[Alfven1950] H. Alfven, Cosmical Electrodynamics, Oxford University Press 1950.

[GZS1977] V.E. Golant, A.P. Zhilinskii, I.E. Sakharov. Fundamentals of Plasma Physics. 1977 (English translation 1980, Wiley).

[solarsystem.de] http://www.solar-system-school.de/lectures/space_plasma_physics_2007/Lecture_4

[Somov2013] B.Somov, Cosmic Plasma Physics, 2013.

[SGR2001] P.Song, T.I. Gombosi and A.J. Ridley, Three-fluid Ohm's law. preprint.

[EpperleinHaines1986] E.M. Epperlein, M.G. Haines. Plasma transport coefficients in a magnetic field by direct numerical solution of the Fokker-Planck equation. Physics of Fluids 29, issue 4 (1986)