In [1]:
```
%matplotlib inline
import numpy as np

from astropy import units as u
# from astropy.constants import c
from astropy.constants import k_B
```

In [2]:
```
%%javascript
MathJax.Hub.Config({
    TeX: { equationNumbers: { autoNumber: "AMS" } }
});
```

As discussed in our previous telecon, we can use equation 10.19 from Stahler & Palla's book

$$\delta u = \delta u_i \left(1 - \frac{i\omega}{n_i \langle \sigma_{in} u_i' \rangle}\right)^{-1} ,$$

where $n_i \langle \sigma_{in} u_i' \rangle$ is "the frequency with which a given natural atom or molecule is struck by ions", δu is the perturbation on the neutral's velocity, and δu_i is the perturbation on the ion's velocity.

The other relation to use is equation 10.21 from Stahler & Palla's book

$$\frac{\omega^2}{k^2} = \frac{B_o^2}{4\pi\rho_o}\left(1 - \frac{i\omega}{n_i \langle \sigma_{in} u_i' \rangle}\right) ,$$

which relates the ω and k.

Finally, equation 10.17 from Stahler & Palla's book relates the perturbation in the magnetic field as

$$\delta u_i = -\frac{\omega}{k}\frac{\delta B}{B_o} .$$

We estimate the velocity perturvations as the velocity dispersions (**Q: should this be the non-thermal velocity dispersion or the thermal+non-thermal value?**) derived from the spectral lines and therefore rewrite equation ??? as

$$\left|\frac{\delta u_i}{\delta u}\right|^2 = 1 + \left(\frac{\omega}{n_i \langle \sigma_{in} u_i' \rangle}\right)^2 \approx \left(\frac{\sigma_v(\mathrm{N_2H^+})}{\sigma_v(\mathrm{NH_3})}\right)^2 .$$

Now, equation ??? can also be rewritten as

$$\left|\frac{\omega^2}{k^2}\right| = \left(\frac{B_o^2}{4\pi\rho_o}\right)\left[1 + \left(\frac{\omega}{n_i \langle \sigma_{in} u_i' \rangle}\right)^2\right]^{1/2} \approx \frac{B_o^2}{4\pi\rho_o}\frac{\sigma_v(\mathrm{N_2H^+})}{\sigma_v(\mathrm{NH_3})} .$$

Combining equations ???, ??? and ??? we obtain

$$\frac{\delta B}{B_o} = \sqrt{\sigma_v(\mathrm{N_2H^+})\sigma_v(\mathrm{NH_3})}\frac{\sqrt{4\pi\rho_o}}{B_o}$$

or

$$\delta B = \sqrt{4\pi\rho_o}\sqrt{\sigma_v(\mathrm{N_2H^+})\sigma_v(\mathrm{NH_3})}$$

In [3]:
```
def sigma_thermal(mu_mol, tk=10*u.K):
    """
    Returns the sound speed for temperature Tk and molecular weight mu.
    This is also used to determine the thermal velocity dispersion of
    a molecular transition.

    """
    return np.sqrt(k_B * tk/(mu_mol * u.u)).to(u.km/u.s)

gauss_B = (u.g/u.cm)**(0.5)/u.s
equiv_B = [(u.G, gauss_B, lambda x: x, lambda x: x)]

B_o = np.array([100, 150]) * u.uG
c_sound = sigma_thermal(2.38, tk=10*u.K).cgs
sigma_ion = 0.62 * c_sound
sigma_neutral = 0.47 * c_sound
rho_o = (7e4 / u.cm**3 * u.u * 2.8).cgs
```

In [4]:
```
(np.sqrt(sigma_ion*sigma_neutral * 4*np.pi * rho_o) / B_o.to((u.g/u.cm)**(1/2)/u.s, equivalencies=equiv_B))
```

Out[4]: [0.20404732, 0.13603155]

In [5]:
```
((np.sqrt(sigma_ion*sigma_neutral * 4*np.pi * rho_o) / B_o) * B_o).to(u.uG, equivalencies=equiv_B)
```

Out[5]: [20.404732, 20.404732] μG

This corresponds to a variation of 20 μG, which corresponds between 13 to 20\% of the field.

If we want to estimate the wavelength then we will need to estimate $n_i \langle \sigma_{in} u_i' \rangle$ to derive ω and then obtain k.

In [ ]: