Gyroviscosity in the NIMROD code

A.Y. Pankin

D. D. Schnack

C. Sovinec

S. E. Kruger

Outline

- Introduction
- Two possible implementations
- Testing GV in the NIMROD code
- Convergence
- Summary

Gyroviscous force

$$\varepsilon\xi\frac{\partial \mathbf{V}_i}{\partial t} + \xi^2\delta\mathbf{V}_i\cdot\nabla\mathbf{V}_i = -\frac{1}{n}\delta\left(\nabla p_i + \frac{\Pi_{i0}}{p_0}\nabla\cdot\Pi_i\right) + \xi(\mathbf{E}+\mathbf{V}_i\times\mathbf{B})$$

$\varepsilon = \omega/\Omega_i$ – ratio of the characteristic frequency to the ion gyro-frequency

$\xi = V_0/V_{thi}$ – ratio of the characteristic flow velocity to the ion thermal speed

$\delta = \rho_i/L$ – ratio of the ion Larmor (gyro-) radius to the macroscopic scale length

Properties of Fluid Models

Model	V_i	ω	β	$\mathbf{J}\times\mathbf{B}$	Whistlers†	KAW††
Hall MHD	V_{thi}/δ	Ω_{ci}	$O(\delta^2)$	$mn\frac{d\mathbf{V}_i}{dt} + O(\delta)$	Yes	No
Ideal MHD	V_{thi}	$\delta\Omega_{ci}$	$O(\delta)$	$O(\delta)$	No	No
Drift	δV_{thi}	$\delta^2\Omega_{ci}$	$O(1)$	$\nabla p + O(\delta^2)$	No	Yes

GV force is not caused by particle collisions

$\Downarrow$

GV force is nor dissipative

Stress tensor

$$\Pi_i = \Pi_{\parallel} + \Pi_{\perp} + \Pi_{gv}$$

$$\Pi_{\parallel} = -\frac{3}{2}\eta_0(\mathbf{b}\cdot\mathbf{W}\cdot\mathbf{b})\left(\mathbf{I}-\frac{1}{3}\mathbf{bb}\right)$$

$$\Pi_{\perp} = -\eta_1\Big\{(\mathbf{I}-\mathbf{bb})\cdot\mathbf{W}\cdot(\mathbf{I}-\mathbf{bb}) - \frac{1}{2}(\mathbf{I}-\mathbf{bb})(\mathbf{I}-\mathbf{bb}):\mathbf{W}$$

$$+4\big[(\mathbf{I}-\mathbf{bb})\cdot\mathbf{W}\cdot\mathbf{bb} + transpose\big]\Big\}$$

$$\Pi_{gv} = \frac{\eta_3}{2}\big[\mathbf{b}\times\mathbf{W}\cdot(\mathbf{I}+3\mathbf{bb}) - (\mathbf{I}+3\mathbf{bb})\cdot\mathbf{W}\times\mathbf{b}\big],$$

where **W** is the rate of strain tensor: $\mathbf{W} = \nabla\mathbf{V} + (\nabla\mathbf{V})^T - \frac{2}{3}\mathbf{I}(\nabla\cdot\mathbf{V})$

$$\eta_3 = \frac{nT_i}{2\Omega},\ \eta_0 = 0.96\frac{nT_i}{\nu},\ \eta_1 = \frac{3}{10}\frac{nT_i\nu}{\Omega^2},$$

$\Omega = \dfrac{eB}{m_i}$ is the ion gyro-frequency

First Implantation: Hooke's law

Linear relationship between stress and rate of strain :

$$\Pi_{ij} = E_{ijkl} W_{kl}$$

Elastic constant tensor :

$$E_{ijkl} \equiv \frac{\partial \Pi_{ij}}{\partial W_{kl}}$$

Symmetry :

$$\underbrace{E_{ijkl} = E_{jikl} \qquad E_{ijkl} = E_{ijlk}}_{\Pi \text{ and } \mathbf{W} \text{ are symmetric}} \qquad \underbrace{E_{ijkl}(\mathbf{B}) = E_{klij}(-\mathbf{B})}_{\mathbf{B} \text{ is a pseudo-vector (Onsager)}}$$

Elastic constant tensor for a magnetized plasma :

$$E_{ijkl} = \frac{1}{2}\eta_3\left(\varepsilon_{imk} b_m u_{lj} + \varepsilon_{jmk} b_m u_{li}\right) \quad , \quad u_{\alpha\beta} = \delta_{\alpha\beta} + 3 b_\alpha b_\beta$$

$$E_{xxxy} = E_{xxyx} = -E_{yyxy} = -E_{yyyx} = -\eta_3$$

$$E_{xyxx} = E_{yxxx} = -E_{xyyy} = -E_{yxyy} = \frac{1}{2}\eta_3$$

$$E_{xzyz} = E_{zxxy} = E_{zxyz} = E_{zxzy}$$

$$= -E_{yzxz} = -E_{zyxz} = -E_{zyzx} = -E_{yzzx} = -\eta_4$$

(Cartesian representation, $\mathbf{b} = \mathbf{e}_z$)

81 components, 16 non - zero, 1 independent (single parameter dependence)

First Implantation: continue

Cartesian rate of strain tensor:

$$\mathbf{W} = \nabla\mathbf{V}_i + \nabla\mathbf{V}_i^T - \frac{2}{3}\mathrm{I}\nabla\cdot\mathbf{V}_i \qquad (\nabla\mathbf{V})_{kl} = \frac{\partial V_l}{\partial x_k}$$

$$W_{ij} = A_{ijkl}(\nabla\mathbf{V})_{kl} \qquad A_{ijkl} = \delta_{il}\delta_{kj} + \delta_{ik}\delta_{lk} - \frac{2}{3}\delta_{ij}\delta_{km}\delta_{ml}$$

Write stress directly in terms of rate of strain:

$$\Pi_{ij} = E_{ijkl}A_{klmn}(\nabla\mathbf{V})_{mn} = D_{ijmn}(\nabla\mathbf{V})_{mn}$$

$$D_{ijmn} = \frac{1}{2}\eta_3\Big[\varepsilon_{izn}\left(\delta_{mj} + 3\delta_{mz}\delta_{jz}\right) + \varepsilon_{izm}\left(\delta_{nj} + 3\delta_{nz}\delta_{jz}\right)$$
$$+\varepsilon_{jzn}\left(\delta_{mi} + 3\delta_{mz}\delta_{iz}\right) + \varepsilon_{jzm}\left(\delta_{nj} + 3\delta_{nz}\delta_{iz}\right)\Big]$$

Assumes $\mathbf{B}$ locally aligned with z – axis

Symmetries:

$$D_{ijmn} = D_{jimn} \quad , \quad D_{ijmn} = D_{ijnm} \quad , \quad D_{ijmn} = -D_{mnij}$$

These symmetries assure GV stress non - dissipative:

$$\int \mathbf{V}\cdot\nabla\Pi_{GV}\,d^3x = 0$$

First Implantation: Computation of GV stress

1. Compute D_{ijmn} once at beginning of run

2. Transform D_{ijmn} to cylindrical coordinates at each spatial location, depending on local direction of magnetic field

$$D'_{\alpha\beta\gamma\delta} = S_{i\alpha} S_{j\beta} D_{ijkl} S_{k\gamma} S_{l\delta} \qquad i,\ j,\ k,\ l = x, y, x \qquad \alpha,\ \beta,\ \gamma,\ \delta = r, \theta, Z$$

$$\mathbf{S} = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\cos\beta\sin\alpha & \cos\beta\cos\alpha & \sin\beta \\ \sin\beta\sin\alpha & -\sin\beta\cos\alpha & \cos\beta \end{pmatrix} \quad ,$$

$$\alpha = \tan^{-1}\left(-\frac{B_r}{B_\theta}\right) \quad , \qquad \beta = \tan^{-1}\left(-\frac{\sqrt{B_r^2 + B_\theta^2}}{B_z}\right) \quad .$$

3. Stress tensor components in cylindrical coordinates is

$$\Pi'_{ij} = D'_{ijmn} \left(\nabla \mathbf{V}\right)'_{mn}$$

$$\left(\nabla \mathbf{V}\right)' = \begin{pmatrix} \dfrac{\partial V_r}{\partial r} & \dfrac{\partial V_\theta}{\partial r} & \dfrac{\partial V_z}{\partial r} \\ \dfrac{1}{r}\dfrac{\partial V_r}{\partial \theta} - \dfrac{V_\theta}{r} & \dfrac{1}{r}\dfrac{\partial V_r}{\partial \theta} + \dfrac{V_r}{r} & \dfrac{1}{r}\dfrac{\partial V_z}{\partial \theta} \\ \dfrac{\partial V_r}{\partial z} & \dfrac{\partial V_\theta}{\partial z} & \dfrac{\partial V_z}{\partial z} \end{pmatrix}$$

4. FE divergence in cylindrical coordinates

$$\left(-\nabla \cdot \Pi_{GV}\right)_{\mathbf{p}} = \int dV \left(\nabla \alpha_{\mathbf{p}} \cdot \mathbf{D}' \cdot \nabla \alpha_{\mathbf{q}}\right) \cdot \overline{\mathbf{V}}_{\mathbf{q}}$$

Second Implementation

Based on weak formulation of the stress tensor force

$$-\int d\mathbf{x}\mathbf{A}^{*}_{\nu,p,n}\cdot\nabla\cdot\Pi = \int d\mathbf{x}\left(\nabla\mathbf{A}^{*}_{\nu,p,n}\right)^{T} : \Pi - \oint d\mathbf{S}\cdot\Pi\cdot\mathbf{A}^{*}_{\nu,p,n}$$

IS ZERO
for non-slip
boundary conditions

$$\nabla\mathbf{A}^{*}_{r,p,n} = \begin{pmatrix} \frac{\partial\alpha_{p,n}}{\partial r} & 0 & 0 \\ \frac{\partial\alpha_{p,n}}{\partial z} & 0 & 0 \\ -\frac{in}{r}\alpha & 0 & \frac{\alpha}{r} \end{pmatrix} \quad \nabla\mathbf{A}^{*}_{z,p,n} = \begin{pmatrix} 0 & \frac{\partial\alpha_{p,n}}{\partial r} & 0 \\ 0 & \frac{\partial\alpha_{p,n}}{\partial z} & 0 \\ 0 & -\frac{in}{r}\alpha & 0 \end{pmatrix} \quad \nabla\mathbf{A}^{*}_{\phi,p,n} = \begin{pmatrix} 0 & 0 & \frac{\partial\alpha_{p,n}}{\partial r} \\ 0 & 0 & \frac{\partial\alpha_{p,n}}{\partial z} \\ -\frac{\alpha}{r} & 0 & -\frac{in}{r}\alpha \end{pmatrix}$$

Find product $\left(\nabla\mathbf{A}^{*}_{\nu,p,n}\right)^{T} : \Pi$

and heating $-\nabla\mathbf{V} : \Pi$ -> $-\int d\mathbf{x}\mathbf{A}^{*}_{\nu,p,n}\cdot\nabla\mathbf{V} : \Pi$

$$\mathbf{W} = \nabla \mathbf{V}_i + \nabla \mathbf{V}_i^T - \frac{2}{3} \mathbf{I} \nabla \cdot \mathbf{V}_i$$

$$\left(\mathbf{I} + 3\mathbf{bb}\right)$$

$$\mathbf{W} \cdot \left(\mathbf{I} + 3\mathbf{bb}\right)$$

$$\mathbf{b} \times \mathbf{W} \cdot \left(\mathbf{I} + 3\mathbf{bb}\right)$$

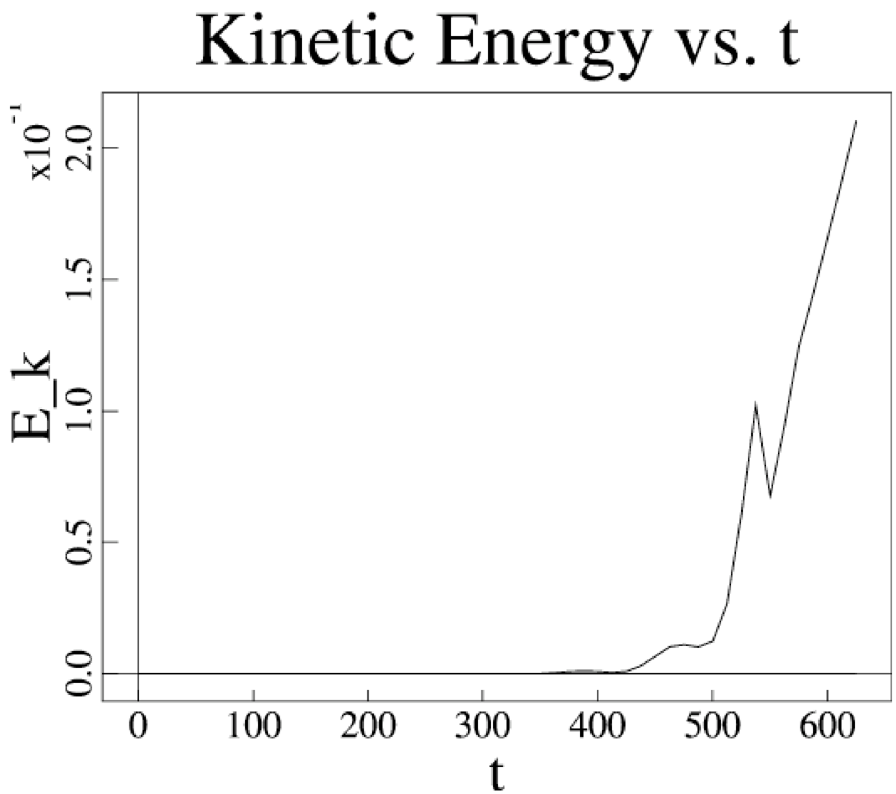
$$\frac{\eta_3}{2}\left[\mathbf{b} \times \mathbf{W} \cdot \left(\mathbf{I} + 3\mathbf{bb}\right) - \left(\mathbf{I} + 3\mathbf{bb}\right) \cdot \mathbf{W} \times \mathbf{b}\right]$$

$$\Pi_i = \Pi + \Pi_{gv}$$

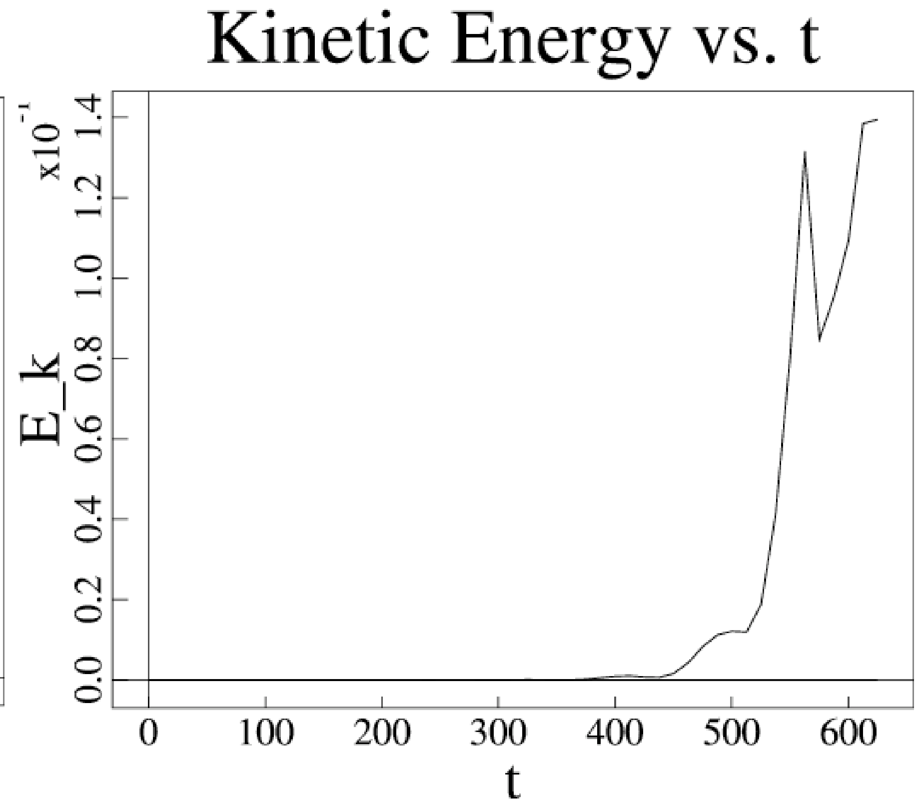
```
c-----------------------------------------------------------------------
c       stress tensor for gyro-viscosity
c-----------------------------------------------------------------------
        IF (gyr_visc>0) THEN
          ALLOCATE(wtmp(3,3,SIZE(int,2),SIZE(int,3)))
          ALLOCATE(b_cross_w(3,3,SIZE(int,2),SIZE(int,3)))
          DO imode=1,nmodes
            wtmp=0.
            DO iy=1,ncy
              DO ix=1,ncx
                vtmp(1:3,1)=vten(1:3,ix,iy,imode)
                vtmp(1:3,2)=vten(4:6,ix,iy,imode)
                vtmp(1:3,3)=vten(7:9,ix,iy,imode)
                ptmp=vtmp+TRANSPOSE(vtmp)
                btmp(1:3,1)=3.*(be_eq(1:3,ix,iy)+be(1:3,ix,iy,imode))
     &                       *(be_eq(1,ix,iy)+be(1,ix,iy,imode))
     &                       /btot2(ix,iy)
                btmp(1:3,2)=3.*(be_eq(1:3,ix,iy)+be(1:3,ix,iy,imode))
     &                       *(be_eq(2,ix,iy)+be(2,ix,iy,imode))
     &                       /btot2(ix,iy)
                btmp(1:3,3)=3.*(be_eq(1:3,ix,iy)+be(1:3,ix,iy,imode))
     &                       *(be_eq(3,ix,iy)+be(3,ix,iy,imode))
     &                       /btot2(ix,iy)
                btmp(1,1) = 1+btmp(1,1)
                btmp(2,2) = 1+btmp(2,2)
                btmp(3,3) = 1+btmp(3,3)
                DO i1=1,3
                  DO i2=1,3
                    DO i3=1,3
                      wtmp(i1,i2,ix,iy)=wtmp(i1,i2,ix,iy)+ptmp(i1,i3)
     &                                  *btmp(i3,i2)
                    ENDDO
                  ENDDO
                ENDDO
              ENDDO
            ENDDO
            call math_cart_cross(b_cross_w(1,:,:,:),
     &                    be_eq(:,:,:)+be(:,:,:,imode),
     &                    wtmp(:,1,:,:),1._r8)
            call math_cart_cross(b_cross_w(2,:,:,:),
     &                    be_eq(:,:,:)+be(:,:,:,imode),
     &                    wtmp(:,2,:,:),1._r8)
            call math_cart_cross(b_cross_w(3,:,:,:),
     &                    be_eq(:,:,:)+be(:,:,:,imode),
     &                    wtmp(:,3,:,:),1._r8)
            DO iy=1,ncy
              DO ix=1,ncx
                ptmp = gyr_visc*(ti_eq(1,ix,iy)*zeff)/
     &               (1.92e4*btot2(ix,iy))*
     &               (b_cross_w(:,:,ix,iy)
     &               +TRANSPOSE(b_cross_w(:,:,ix,iy)))
                piten(1:3,ix,iy,imode)=
     $            piten(1:3,ix,iy,imode)-ptmp(1:3,1)
                piten(4:6,ix,iy,imode)=
     $            piten(4:6,ix,iy,imode)-ptmp(1:3,2)
                piten(7:9,ix,iy,imode)=
     $            piten(7:9,ix,iy,imode)-ptmp(1:3,3)
```

```
              ENDDO
            ENDDO
          ENDDO
          DEALLOCATE(wtmp,b_cross_w)
          IF (eq_flow/='none') THEN
            ALLOCATE(wtmpr(3,3,SIZE(int,2),SIZE(int,3)))
            ALLOCATE(b_cross_wr(3,3,SIZE(int,2),SIZE(int,3)))
            DO iy=1,ncy
              DO ix=1,ncx
                vtmpr(1:3,1)=veqten(1:3,ix,iy)
                vtmpr(1:3,2)=veqten(4:6,ix,iy)
                vtmpr(1:3,3)=veqten(7:9,ix,iy)
                ptmpr=vtmpr+TRANSPOSE(vtmpr)
                btmpr(1:3,1)=3.*be_eq(1:3,ix,iy)*be_eq(1,ix,iy)
     &                       /btot2(ix,iy)
                btmpr(1:3,2)=3.*be_eq(1:3,ix,iy)*be_eq(2,ix,iy)
     &                       /btot2(ix,iy)
                btmpr(1:3,3)=3.*be_eq(1:3,ix,iy)*be_eq(3,ix,iy)
     &                       /btot2(ix,iy)
                btmpr(1,1) = 1+btmpr(1,1)
                btmpr(2,2) = 1+btmpr(2,2)
                btmpr(3,3) = 1+btmpr(3,3)
                vtmpr=0.
                DO i1=1,3
                  DO i2=1,3
                    DO i3=1,3
                      wtmpr(i1,i2,ix,iy)=wtmpr(i1,i2,ix,iy)
     &                                   +ptmpr(i1,i3)*btmpr(i3,i2)
                    ENDDO
                  ENDDO
                ENDDO
              ENDDO
            ENDDO
            call math_cart_cross(b_cross_wr(1,:,:,:),be_eq,
     &                    wtmpr(:,1,:,:),1._r8)
            call math_cart_cross(b_cross_wr(2,:,:,:),be_eq,
     &                    wtmpr(:,2,:,:),1._r8)
            call math_cart_cross(b_cross_wr(3,:,:,:),be_eq,
     &                    wtmpr(:,3,:,:),1._r8)
            DO iy=1,ncy
              DO ix=1,ncx
                ptmpr = gyr_visc*zeff*ti_eq(1,ix,iy)/
     &               (1.92e4*btot2(ix,iy))
     &               *(b_cross_wr(:,:,ix,iy)
     &                 +TRANSPOSE(b_cross_wr(:,:,ix,iy)))
                pi_veq(1:3,ix,iy)=
     $            pi_veq(1:3,ix,iy)-ptmpr(1:3,1)
                pi_veq(4:6,ix,iy)=
     $            pi_veq(4:6,ix,iy)-ptmpr(1:3,2)
                pi_veq(7:9,ix,iy)=
     $            pi_veq(7:9,ix,iy)-ptmpr(1:3,3)
              ENDDO
            ENDDO
            DEALLOCATE(wtmpr,b_cross_wr)
          ENDIF
        ENDIF
```

Testing GV implementation



Without gyroviscosity



With gyroviscosity

Testing GV implementation Dispersion relation

Parallel Propagation

Dispersion relation: $(\omega \pm \omega_4)(\omega \pm \omega_W) = \omega_A^2$

$$\omega_A = \frac{B_0 |k_z|}{\sqrt{\mu_0 mn}} = V_A |k_z|$$

$$\omega_4 = \frac{\eta_4 k_z^2}{mn} = \frac{V_{th_i}^2}{2\Omega} k_z^2 = \frac{1}{2}(\rho_i k_z)^2 \Omega$$

$$\omega_W = \frac{B_0 k_z^2}{\mu_0 ne} = \left(\frac{\omega_A}{\Omega}\right)^2 \Omega = \frac{1}{\beta}(\rho_i k_z)^2 \Omega$$

If $\rho_i k_z << 1$, the solutions for the left and right polarized waves are

$$\omega_{L\pm} = V_A k_z \left[\pm 1 + \frac{1+\beta}{2\sqrt{\beta}}(\rho_i k_z)\right]$$

$$\omega_{R\pm} = V_A k_z \left[\pm 1 - \frac{1+\beta}{2\sqrt{\beta}}(\rho_i k_z)\right]$$

Perpendicular Propagation

Dispersion relation: $\dfrac{\omega^2}{\omega_s^2 + \omega_A^2} = 1 + \dfrac{\omega_3^2}{\omega_s^2 + \omega_A^2}$

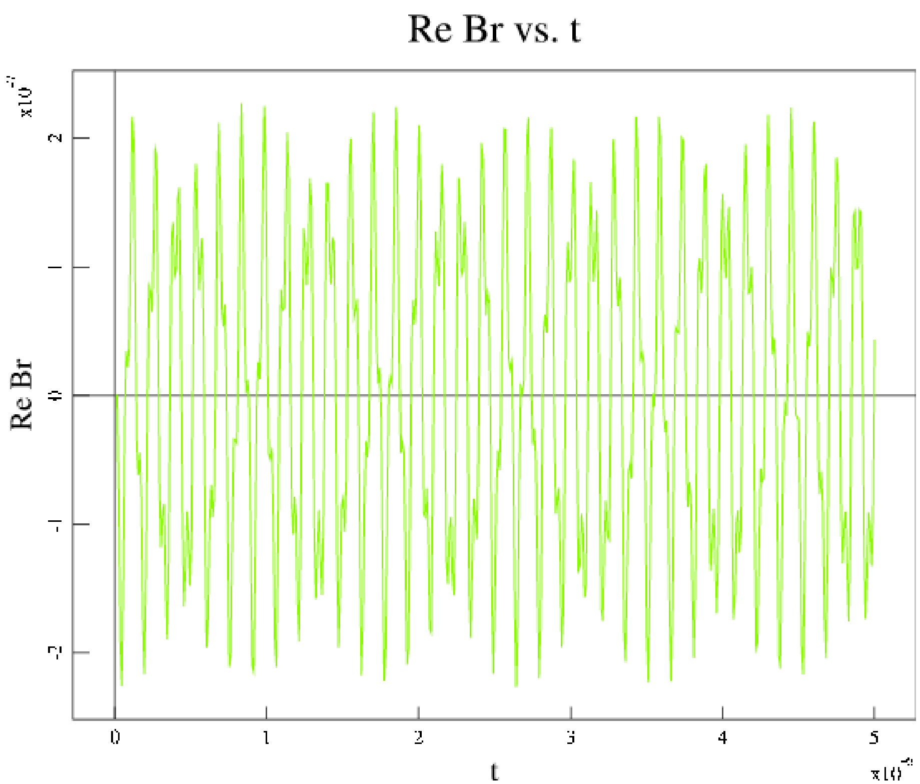
$\omega_s^2 = C_s^2 k_x^2$, $\omega_3 = \omega_4 / 2$.

$$\omega^2 = V_A^2 k_x^2 \left[1 + \frac{\gamma\beta}{2} + \frac{\beta}{16}(\rho_i k_x)^2\right]$$

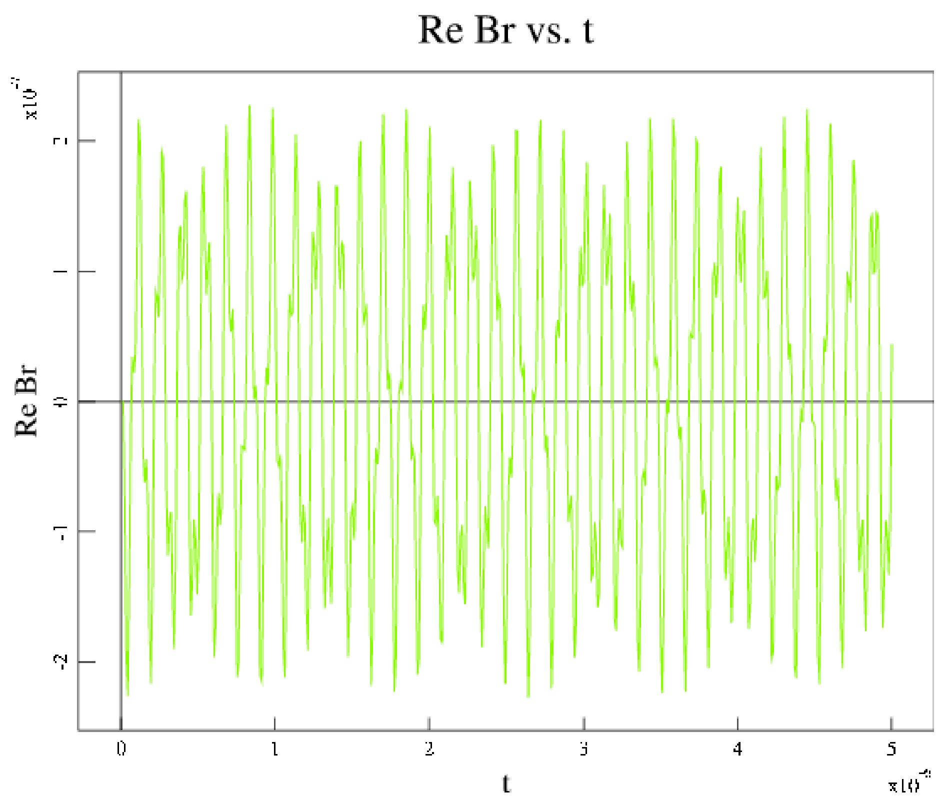
The mode is elliptically polarized in the plane perpendicular to **B**

$$\frac{iV_y}{V_x} = \pm \frac{1}{4} \frac{\sqrt{\beta}}{1 + \gamma\beta/2} \rho_i k_x$$

Testing GV implementation



Without gyroviscosity



With gyroviscosity