A evaluation proposal for the Wuerth centrifugal pumper
by W.D. Bauer, © 2.7.02

1. Introduction

In a previous article the author described Wuerth's rotor experiment [1]. The result of the experiment was that a overunity effect was measured by Wuerth and confirmed by a independent experimenter . From the standpoint of the conventional theory these results are not possible.Other own experiments of the author showed not any significant deviations to classical mechanics so far.
Due to the permanent interest in the theme the author writes this proposal here which should settle the question if the evaluation will be done

2. Description

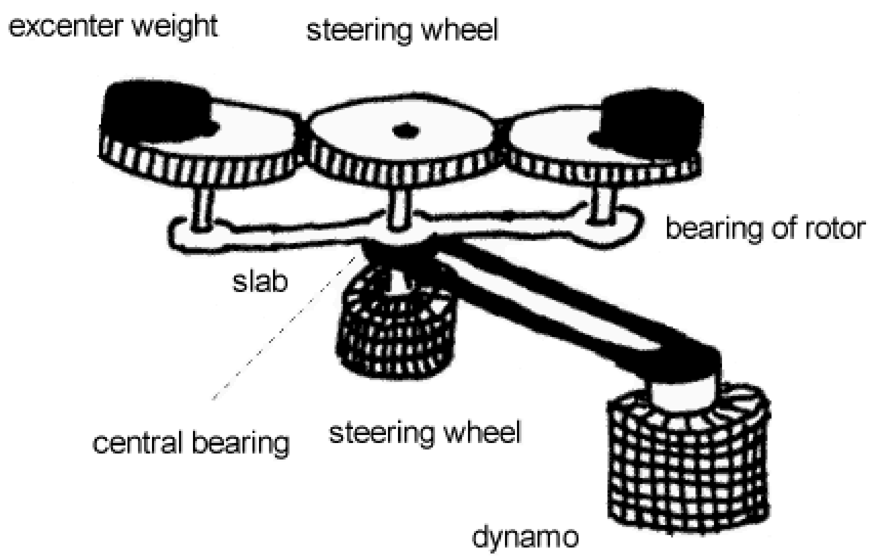


Fig.1: the Wuerth centrifugal force pumper

In principle Wuerth's pumper [2] is nothing else as an special planetary gear unit (in German Planetengetriebe) where excenter weights are placed on the outer gear wheels, comp. figure above. The unit is set into rotation initially by an motor - dynamo (or by hand) at the central axis who drives the slab.
If the setup has a rotation of 1-2 Hz typically the rotation is maintained by steering the central gear wheel back and forth dependent from the actual position of the weights on the disks. Wuerth's claims regarding this setup are:
If phase and amplitude of the steering action is chosen appropriately it is possible to win energy from axis driving the slab.
The steering signal itself has the form of a steplike stairlike (not exactly) periodical function.
In effect the whole system is feedback loop with a control where the person (or a motor) at the steering wheel tries to give work input or to take away braking output in such a way that the work output at the axis of the slab is optimized . Wuerth claims that the work at the steering wheel is in sum an work output work if integrated over time.

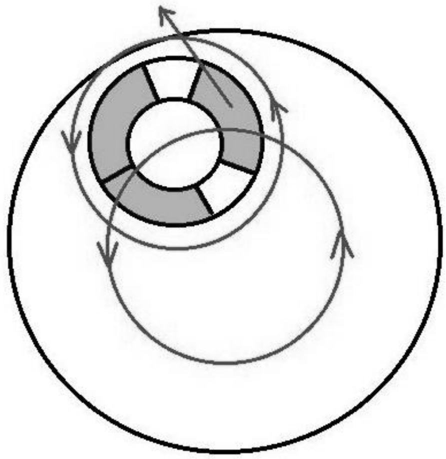


Fig.2: Acc. to Wuerth additional forces deviating from Newtomian mechanics appear in cycloid motion.

3. Possible alternatives for the evaluation

Principally there exist two alternatives :

The experimental option

Wuerth himself believes that his measured effects are due to deviation to Newton mechanics for certain cyclic or cycloid motions. He believes that on the cycloid path additional forces appear which are illustrated roughly in the picture below. Wuerth´s idea can be tested principally by the following setup:

Instead of a weight a bowl is set on a outer pumper disk. The free movement of the bowl is blocked by two force sensors situated in at an angle of 90 degree. They take up perpendicular force components from the bowl force in motion.

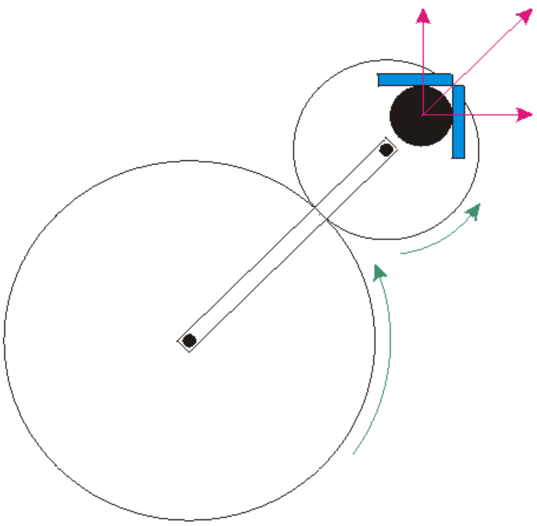


Fig.3: the experimental test setup for evaluation of Newtons law,
blue blocks indicate force sensors. Theoretical forces should be calculated acc. to eq. 3

The forces of the bowl can be calculated acc. to classical mechanics by

$$\vec{\mathbf{F}} = m.\ddot{\vec{\mathbf{X}}}$$

with

$$\vec{\mathbf{X}} = \begin{pmatrix} r_1\cos(\alpha) + r_2\cos(\beta) \\ r_1\sin(\alpha) + r_2\sin(\beta) \end{pmatrix} \quad \text{where } \beta = \eta\alpha$$

This can be compared to the measured values from the force sensors.

The theoretical option
The answer from the standpoint of theory is that no overunity cycles are possible !
Acc. to the knowledge of the author there are two general theorems regarding energy conservation.

1.) Hamilton's energy conservation
Hamiltonian energy conservation means: If the description of a mechanic system contains skeleronomic constraints, non moving coordinates and conservative forces it holds

$$H = T + V = \text{constant}$$

where H is the Hamilton energy, T is the kinetic energy and V is the potential energy
The proof can be found in [3].

2) Energy conservation in a cycle acc. to Bruhn [4]
In the context of the discussion regarding the previous rotator-article of the author G. Bruhn generalized the problem and formulated the following theorem of energy conservation:

"Let $\{ q_k(t) {:=} k{=}1....n \}$ denote a (C^1 and piecewise C^2) solution of the equations of motion, a so called path. Then the work done by the external generalized forces Z_k along the path $\{q_k(t)\}$during the time interval $[t_0 , t_1]$ is given by the work integral $\int_{t_0}^{t_1} \sum_k Z_k \dot{q}_k dt$ of the external forces Z_k . The amount of work equals the change of the total system. This amount equals the change of the Hamilton energy of the system $E(q_k, \dot{q}_k) = T(q_k, \dot{q}_k) + V(q_k)$ }during the time interval $[t_0 , t_1]$." □

The proof of this theorem can be found in the appendix 1 . The theorem has the following important consequence:

"If a mechanical system returns to its initial state $(q_k(t_0), \dot{q}_k(t_0))$ after a time cycle [t_0 , t_1], then the total work done by the external forces during the cycle is balanced, meaning we have energy conservation" □

If both sentences are compared with the construction and working conditions of Wuerth's setup it seems at first sight that it is not so easy to reconcile it with the presumptions of 1) and 2)
The input surely is no signal which can be described uniquely by a potential. Therefore Hamiltonian energy conservation cannot be asssumed.
Bruhn's cyclic energy conservation is not applicable because the cycle is not closed in the phase space due to the control feedback loop in the setup.
However, a consequence of Bruhn's theorem is that if energy leaves the system via constraint forces, then H decreases until the existing lower border of the Hamilton energy H is reached because T =1/2 mv^2 has a lower border which is zero.

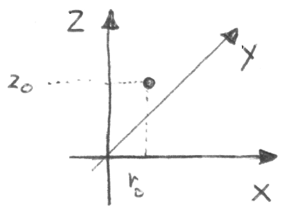
Nevertheless we checked the problem theoretically in more detail and can give a full description of the movement of the pumper which can serve for comparison if a evaluation is done. This is done in the next section.

3. The theoretical model of the centrifugal pumper

Determination of the path of a single mass point
In our calculation we take account only for the excenter weights and neglect any other masses of the setup.

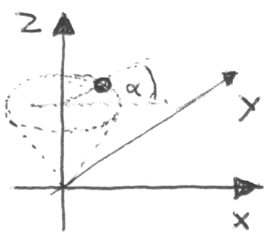
In order to derive path and movement of a single mass point of the pumper weights we start from a single mass point placed in the excenter weight described by the vector

$$\overline{\mathbf{R}}_0 = \begin{pmatrix} r \\ 0 \\ z \end{pmatrix}$$



The origin here is at the centre of a bearing in the slab.

This point is rotated during the motion of the pumper around the vertical axis of the supporting disk by the angle



$$\alpha = \beta + \eta(\varphi - \delta)$$

The total angle $\alpha(t)$ comprises the contribution of a fixed point at position β on the wheel, the influence of the gear translation η, the contribution of the angular motion $\varphi(t)$ of the slab and the steering action $\delta(t)$ from outside.

The new position of the point is described now by

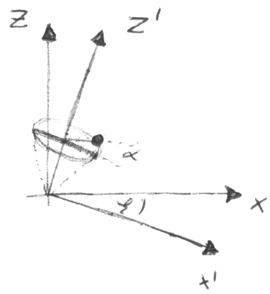
$$\overline{\mathbf{X}}'' = \overline{\overline{\boldsymbol{\alpha}}}(t).\overline{\mathbf{R}}_0$$

with $\overline{\overline{\boldsymbol{\alpha}}}(t)$ being the rotation matrix defined as

$$\overline{\overline{\boldsymbol{\alpha}}}(t) = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Similarly like in the real setups we include additionally now the possibility of tilting the vertical axis by the constant angle ξ. The new position of the point is described now by

$$\overline{\mathbf{X}}' = \overline{\overline{\boldsymbol{\xi}}}.\overline{\mathbf{X}}'' = \overline{\overline{\boldsymbol{\xi}}}.\overline{\overline{\boldsymbol{\alpha}}}.\overline{\mathbf{R}}_{\mathbf{0}}$$

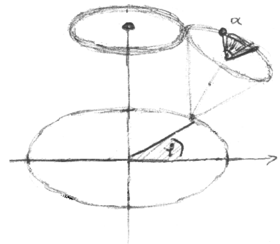


where $\overline{\overline{\boldsymbol{\xi}}}$ is a constant rotation matrix defined as

$$\overline{\overline{\boldsymbol{\xi}}} = \begin{pmatrix} \cos(\xi) & 0 & \sin(\xi) \\ 0 & 1 & 0 \\ -\sin(\xi) & 0 & \cos(\xi) \end{pmatrix}$$

In setup shown in fig.1 we have the trivial case with $\xi = 0$.

Now, the point is shifted away by the distance ΔR from the centre and rotated by the angle φ (t)around the present vertical z-axis. The new position $\bar{\mathbf{X}}$(t) of the point is described now by



$$\bar{\mathbf{X}}(t) = \bar{\bar{\varphi}}.\bar{\mathbf{X}}' + \bar{\mathbf{S}} = \bar{\bar{\varphi}}.\bar{\bar{\xi}}.\bar{\mathbf{X}}'' + \bar{\mathbf{S}} = \bar{\bar{\varphi}}.\bar{\bar{\xi}}.\bar{\bar{\alpha}}.\bar{\mathbf{R}}_0 + \bar{\mathbf{S}}$$

where the rotation matrix $\bar{\bar{\varphi}}$ (t) and "shift" vector $\bar{\mathbf{S}}$(t) are defined as

$$\bar{\bar{\varphi}}(t) = \begin{pmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \bar{\mathbf{S}}(t) = \Delta R \begin{pmatrix} \cos(\varphi) \\ \sin(\varphi) \\ 0 \end{pmatrix}$$

The resulting vector $\bar{\mathbf{X}}$(t) describes the motion of the mass point of the excenter weight dependent from the time t . The velocity $\bar{\mathbf{V}}$(t) of the point can be determined now by differentiation of $\bar{\mathbf{X}}$ with respect to t

$$\bar{\mathbf{V}}(t) = \frac{\partial \bar{\mathbf{X}}}{\partial t} = \left(\dot{\bar{\bar{\varphi}}}.\bar{\bar{\xi}}.\bar{\bar{\alpha}} + \bar{\bar{\varphi}}.\bar{\bar{\xi}}.\dot{\bar{\bar{\alpha}}} \right).\bar{\mathbf{R}}_0 + \dot{\bar{\mathbf{S}}}$$

Determination of the equation of motion

In order to calculate the time-dependent equations of motion we use the Lagrange equations of first kind. The differential Lagrange energy $d\mathcal{L}$ of a single point is

$$d\mathcal{L} = \rho(r,\beta,z)\left(\frac{1}{2}\bar{\mathbf{V}}^2 - \bar{\mathbf{G}}.\bar{\mathbf{X}} \right) dr\, d\beta\, dz$$

where $\bar{\mathbf{G}}$=(0,0,g) is the vector of the field of gravitation and $\rho(r,\beta,z)$ is the mass density of the weight. The total Lagrange energy $\mathcal{L}$ of the setup is calculated by integration of the above formula. The integration allows to add up the contribution of all masses of and on the outer gear wheels in the form of disk sectors of a cylinder . This methods avoids the complicated application of the tensors of inertia in the equation of motion.

In order to simplify our calculation we will take account for only one sector in the following. The equations of motion can be derived using the Euler-Lagrange equation

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\delta}} - \frac{\partial \mathcal{L}}{\partial \delta} = Z_\delta$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\varphi}} - \frac{\partial \mathcal{L}}{\partial \varphi} = Z_\varphi$$

where $Z_{\varphi/\delta}$ are the integrated constraint forces exerted by the motors.
If the expression of $\mathcal{L}$ is evaluated, integrated and inserted in the equations of motion above we get the solution (see in appendix 2 or 3 or click on each Z symbol below) by help of a math package like MATHEMATICA.

See formula of Z_δ or formula of Z_φ in appendix 2 or 3

The program can be found under the name pumper.nb in the attached documentation package pumper.zip .
In order to simulate input we choose as the constraints of motion the stair-like signal

$$\delta = kt + \sum_{i=1}^{n} a_i \sin(\eta(\dot{\varphi}_0 - k)t)$$

where the a_i are the coefficients of an appropriately chosen Fourier serie and k indicates the mean slope of the input signal. The output is simulated by the constraint

$$\varphi = \dot{\varphi}_0 t$$

where $\dot{\varphi}_0$ is the constant angular velocity of the slab whose axis is the work output.
The whole theory is implemented in the MATHEMATICA file pumper.nb to be found in this directory in the documentation file pumper.zip .

4. Discussion

The value of the method of derivation here consists in the fact that the derivation is done without any reference of the complicated mathematics of inertial tensors.

5. Bibliography

1) W.D. Bauer see at http://www.overunity-theory.de/rotator/rotatornew.pdf
2) see at http://www.wuerth-ag.de
3) Kuypers F. Klassische Mechanik Wiley VCH Weinheim 1997
4) G.W. Bruhn siehe http://www.mathematik.tu-darmstadt.de/~bruhn

Appendix 1: Proof of Bruhn's energy theorem

The equations of motion $\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = Z_k$,($k=1,....,n$), yield

$$E_{[t_0,t_1]} = \int_{t_0}^{t_1} \sum_k Z_k \dot{q}_k dt = \int_{t_0}^{t_1} \sum_k \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} \right) \dot{q}_k \, dt = \int_{t_0}^{t_1} \sum_k \left[\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k}\dot{q}_k - \left(\frac{\partial L}{\partial \dot{q}_k}\ddot{q}_k + \frac{\partial L}{\partial q_k}\dot{q} \right) \right] dt$$

But since $T(q_k,\dot{q}_k)$ is homogeneous of order 2 with respect to the variables $\dot{q}_k$ (comp.[1]), we obtain

$$\sum_k \frac{\partial L}{\partial \dot{q}_k}\dot{q}_k = \sum_k \frac{\partial T}{\partial \dot{q}_k}\dot{q}_k = 2T$$

Thus

$$E_{[t_0,t_1]} = \int_{t_0}^{t_1} \frac{d}{dt}(2T - L)dt = \int_{t_0}^{t_1} \frac{d}{dt}(T + V)dt = [E(q_k(t), \dot{q}_k(t))]_{t_0}^{t_1}$$

Appendix 2: output of Z_{del} from MATHEMATICA in C-notation

```
Zdel=(deltaz*Eta*Rho*(-8*(8*(Power(rin,3) - Power(rout,3))*Sin(deltaBeta/2.)*
          (-2*deltaR*Cos(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi)*
             Sin(2*Phi) -
            Cos(Xi)*(2*deltaR*Cos(2*Phi) + (2*z0 + deltaz)*Sin(Xi))*
             Sin(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi)) +
         3*(Power(rin,4) - Power(rout,4))*Sin(deltaBeta)*
          Power(Sin(Xi),2)*
          Sin(2*Beta0 + deltaBeta - 2*Eta*Del + 2*Eta*Phi))*
       Power(DPhidt,2) +
      16*(8*g*(Power(rin,3) - Power(rout,3))*Sin(deltaBeta/2.)*
          Sin(Xi)*Sin(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi) -
         3*(Power(rin,4) - Power(rout,4))*deltaBeta*Eta*
          D2Deldt2 +
         (3*(Power(rin,4) - Power(rout,4))*deltaBeta*Eta +
            4*(Power(rin,3) - Power(rout,3))*
             Cos(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi)*Sin(deltaBeta/2.)*
             (-2*deltaR*Cos(2*Phi) + (2*z0 + deltaz)*Sin(Xi)) +
            Cos(Xi)*(3*(Power(rin,4) - Power(rout,4))*deltaBeta +
               8*(Power(rin,3) - Power(rout,3))*deltaR*
                Sin(deltaBeta/2.)*Sin(2*Phi)*
                Sin(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi)))*
          D2Phidt2)))/192.  ;
```

Appendix 3: output of Z_{phi} from MATHEMATICA in C-notation

```
Zphi=(deltaz*Rho*(4*Power(rout,2)*deltaBeta*
       (6*deltaR*(2*z0 + deltaz)*Sin(Xi)*Sin(2*Phi)*
          Power(DPhidt,2) +
         (3*Power(z0,2) + 6*Power(deltaR,2) + 3*z0*deltaz +
            Power(deltaz,2) -
            (3*Power(z0,2) + 3*z0*deltaz + Power(deltaz,2))*Cos(2*Xi) -
            6*deltaR*(2*z0 + deltaz)*Cos(2*Phi)*Sin(Xi))*
          D2Phidt2) +
      4*Power(rin,2)*deltaBeta*(-6*deltaR*(2*z0 + deltaz)*Sin(Xi)*Sin(2*Phi)*
          Power(DPhidt,2) +
         (-3*Power(z0,2) - 6*Power(deltaR,2) - 3*z0*deltaz -
            Power(deltaz,2) +
            (3*Power(z0,2) + 3*z0*deltaz + Power(deltaz,2))*Cos(2*Xi) +
            6*deltaR*(2*z0 + deltaz)*Cos(2*Phi)*Sin(Xi))*
          D2Phidt2) +
      8*Power(rout,3)*Sin(deltaBeta/2.)*
       (4*g*Eta*Sin(Xi)*Sin(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi) +
         Power(Eta,2)*(4*deltaR*Cos(Xi)*
             Cos(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi)*Sin(2*Phi) -
            2*(-2*deltaR*Cos(2*Phi) + (2*z0 + deltaz)*Sin(Xi))*
             Sin(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi))*
          Power(DDeldt,2) -
         Eta*(-((2*z0 + deltaz)*
               Cos(Beta0 + deltaBeta/2. - 2*Xi - Eta*Del + Eta*Phi)) -
            2*(2*z0 + deltaz)*Eta*
             Cos(Beta0 + deltaBeta/2. - Xi - Eta*Del + Eta*Phi) +
            4*z0*Eta*Cos(Beta0 + deltaBeta/2. + Xi - Eta*Del + Eta*Phi) +
            2*deltaz*Eta*Cos(Beta0 + deltaBeta/2. + Xi - Eta*Del + Eta*Phi) +
            2*z0*Cos(Beta0 + deltaBeta/2. + 2*Xi - Eta*Del + Eta*Phi) +
            deltaz*Cos(Beta0 + deltaBeta/2. + 2*Xi - Eta*Del + Eta*Phi) -
            4*deltaR*Sin(Beta0 + deltaBeta/2. - Eta*Del + (-2 + Eta)*Phi) +
            4*deltaR*Eta*Sin(Beta0 + deltaBeta/2. - Eta*Del + (-2 + Eta)*Phi) +
            2*deltaR*Sin(Beta0 + deltaBeta/2. - Xi - Eta*Del +
               (-2 + Eta)*Phi) -
            2*deltaR*Eta*Sin(Beta0 + deltaBeta/2. - Xi - Eta*Del +
               (-2 + Eta)*Phi) +
            2*deltaR*Sin(Beta0 + deltaBeta/2. + Xi - Eta*Del +
               (-2 + Eta)*Phi) -
            2*deltaR*Eta*Sin(Beta0 + deltaBeta/2. + Xi - Eta*Del +
               (-2 + Eta)*Phi) +
            4*deltaR*Sin(Beta0 + deltaBeta/2. - Eta*Del + (2 + Eta)*Phi) +
            4*deltaR*Eta*Sin(Beta0 + deltaBeta/2. - Eta*Del + (2 + Eta)*Phi) +
            2*deltaR*Sin(Beta0 + deltaBeta/2. - Xi - Eta*Del + (2 + Eta)*Phi) +
            2*deltaR*Eta*Sin(Beta0 + deltaBeta/2. - Xi - Eta*Del +
               (2 + Eta)*Phi) +
            2*deltaR*(1 + Eta)*
             Sin(Beta0 + deltaBeta/2. + Xi - Eta*Del + (2 + Eta)*Phi))*
          DDeldt*DPhidt +
```

```
    (4*deltaR*(-2 + Eta)*(-1 + Eta)*Power(Sin(Xi/2.),2)*
        Sin(Beta0 + deltaBeta/2. - Eta*Del + (-2 + Eta)*Phi) -
       2*(2*z0 + deltaz)*Eta*(Eta + Cos(Xi))*Sin(Xi)*
        Sin(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi) +
       4*deltaR*(1 + Eta)*(2 + Eta)*Power(Cos(Xi/2.),2)*
        Sin(Beta0 + deltaBeta/2. - Eta*Del + (2 + Eta)*Phi))*
     Power(DPhidt,2) +
    2*Eta*(Cos(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi)*
        (2*deltaR*Cos(2*Phi) - (2*z0 + deltaz)*Sin(Xi)) -
       2*deltaR*Cos(Xi)*Sin(2*Phi)*
        Sin(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi))*
     D2Deldt2 +
    2*(2*deltaR*(-1 + Eta)*(-1 + Cos(Xi))*
        Cos(Beta0 + deltaBeta/2. - Eta*Del + (-2 + Eta)*Phi) -
       2*deltaR*(1 + Eta)*(1 + Cos(Xi))*
        Cos(Beta0 + deltaBeta/2. - Eta*Del + (2 + Eta)*Phi) +
       2*(2*z0 + deltaz)*(Eta + Cos(Xi))*
        Cos(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi)*Sin(Xi))*
     D2Phidt2) +
 3*Power(rin,4)*(2*Eta*Sin(deltaBeta)*Power(Sin(Xi),2)*
     Sin(2*Beta0 + deltaBeta - 2*Eta*Del + 2*Eta*Phi)*
     (2*DDeldt - DPhidt)*
     DPhidt +
    4*deltaBeta*Eta*(Eta + Cos(Xi))*D2Deldt2 -
    (deltaBeta*(3 + 4*Power(Eta,2) + 8*Eta*Cos(Xi) + Cos(2*Xi)) -
       2*Cos(2*Beta0 + deltaBeta - 2*Eta*Del + 2*Eta*Phi)*Sin(deltaBeta)*
        Power(Sin(Xi),2))*D2Phidt2) +
 3*Power(rout,4)*(-2*Eta*Sin(deltaBeta)*Power(Sin(Xi),2)*
     Sin(2*Beta0 + deltaBeta - 2*Eta*Del + 2*Eta*Phi)*
     (2*DDeldt - DPhidt)*
     DPhidt -
    4*deltaBeta*Eta*(Eta + Cos(Xi))*D2Deldt2 +
    (deltaBeta*(3 + 4*Power(Eta,2) + 8*Eta*Cos(Xi) + Cos(2*Xi)) -
       2*Cos(2*Beta0 + deltaBeta - 2*Eta*Del + 2*Eta*Phi)*Sin(deltaBeta)*
        Power(Sin(Xi),2))*D2Phidt2) +
 8*Power(rin,3)*Sin(deltaBeta/2.)*
  (-4*g*Eta*Sin(Xi)*Sin(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi) -
    2*Power(Eta,2)*(2*deltaR*Cos(Xi)*
        Cos(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi)*Sin(2*Phi) +
       (2*deltaR*Cos(2*Phi) - (2*z0 + deltaz)*Sin(Xi))*
        Sin(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi))*
     Power(DDeldt,2) +
    Eta*(-((2*z0 + deltaz)*
          Cos(Beta0 + deltaBeta/2. - 2*Xi - Eta*Del + Eta*Phi)) -
       2*(2*z0 + deltaz)*Eta*
        Cos(Beta0 + deltaBeta/2. - Xi - Eta*Del + Eta*Phi) +
       4*z0*Eta*Cos(Beta0 + deltaBeta/2. + Xi - Eta*Del + Eta*Phi) +
       2*deltaz*Eta*Cos(Beta0 + deltaBeta/2. + Xi - Eta*Del + Eta*Phi) +
       2*z0*Cos(Beta0 + deltaBeta/2. + 2*Xi - Eta*Del + Eta*Phi) +
       deltaz*Cos(Beta0 + deltaBeta/2. + 2*Xi - Eta*Del + Eta*Phi) -
```

```
    4*deltaR*Sin(Beta0 + deltaBeta/2. - Eta*Del + (-2 + Eta)*Phi) +
    4*deltaR*Eta*Sin(Beta0 + deltaBeta/2. - Eta*Del + (-2 + Eta)*Phi) +
    2*deltaR*Sin(Beta0 + deltaBeta/2. - Xi - Eta*Del +
      (-2 + Eta)*Phi) -
    2*deltaR*Eta*Sin(Beta0 + deltaBeta/2. - Xi - Eta*Del +
      (-2 + Eta)*Phi) +
    2*deltaR*Sin(Beta0 + deltaBeta/2. + Xi - Eta*Del +
      (-2 + Eta)*Phi) -
    2*deltaR*Eta*Sin(Beta0 + deltaBeta/2. + Xi - Eta*Del +
      (-2 + Eta)*Phi) +
    4*deltaR*Sin(Beta0 + deltaBeta/2. - Eta*Del + (2 + Eta)*Phi) +
    4*deltaR*Eta*Sin(Beta0 + deltaBeta/2. - Eta*Del + (2 + Eta)*Phi) +
    2*deltaR*Sin(Beta0 + deltaBeta/2. - Xi - Eta*Del + (2 + Eta)*Phi) +
    2*deltaR*Eta*Sin(Beta0 + deltaBeta/2. - Xi - Eta*Del +
      (2 + Eta)*Phi) +
    2*deltaR*(1 + Eta)*
     Sin(Beta0 + deltaBeta/2. + Xi - Eta*Del + (2 + Eta)*Phi))*
  DDeldt*DPhidt +
 2*((-2*deltaR*(-2 + Eta)*(-1 + Eta)*Power(Sin(Xi/2.),2)*
       Sin(Beta0 + deltaBeta/2. - Eta*Del + (-2 + Eta)*Phi) +
      (2*z0 + deltaz)*Eta*(Eta + Cos(Xi))*Sin(Xi)*
       Sin(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi) -
      2*deltaR*(1 + Eta)*(2 + Eta)*Power(Cos(Xi/2.),2)*
       Sin(Beta0 + deltaBeta/2. - Eta*Del + (2 + Eta)*Phi))*
     Power(DPhidt,2) +
    Eta*(Cos(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi)*
       (-2*deltaR*Cos(2*Phi) + (2*z0 + deltaz)*Sin(Xi)) +
      2*deltaR*Cos(Xi)*Sin(2*Phi)*
       Sin(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi))*
     D2Deldt2 +
    (-2*deltaR*(-1 + Eta)*(-1 + Cos(Xi))*
       Cos(Beta0 + deltaBeta/2. - Eta*Del + (-2 + Eta)*Phi) +
      2*deltaR*(1 + Eta)*(1 + Cos(Xi))*
       Cos(Beta0 + deltaBeta/2. - Eta*Del + (2 + Eta)*Phi) -
      2*(2*z0 + deltaz)*(Eta + Cos(Xi))*
       Cos(Beta0 + deltaBeta/2. - Eta*Del + Eta*Phi)*Sin(Xi))*
     D2Phidt2))))/48.  ;
```