```
> restart:
  with(plots):
```

Compute waveforms for eccentric binaries, using equations from Favata, PRD 84, 124013 (2012). Make sounds and plots.

Construct GW signal in terms of polarizations and antenna patterns:

```
> h:=hp*Fp+hx*Fx;
```

$$h := Fp\ hp + Fx\ hx \tag{1}$$

```
> Fp:=1/2*(1+cos(theta)^2)*cos(2*phi)*cos(2*Psi)-cos(theta)*sin(2*
  phi)*sin(2*Psi);

  Fx:=1/2*(1+cos(theta)^2)*cos(2*phi)*sin(2*Psi)+cos(theta)*sin(2*
  phi)*cos(2*Psi);
```

$$Fp := \frac{1}{2}\left(1+\cos(\theta)^2\right)\cos(2\,\phi)\cos(2\,\Psi) - \cos(\theta)\sin(2\,\phi)\sin(2\,\Psi)$$

$$Fx := \frac{1}{2}\left(1+\cos(\theta)^2\right)\cos(2\,\phi)\sin(2\,\Psi) + \cos(\theta)\sin(2\,\phi)\cos(2\,\Psi) \tag{2}$$

The Newtonian order polarizations (eqs. 2.12a,b and 2.23a-c) are [don't confuse capital and lower-case angles---they are different]

```
> hp:=B*((1+cos(Theta)^2)*(G1*cos(2*(phi_orb-Phi))-2*G2*sin(2*
  (phi_orb-Phi)))-sin(Theta)^2*G3);
  hx:=2*B*cos(Theta)*(2*G2*cos(2*(phi_orb-Phi))+G1*sin(2*(phi_orb-
  Phi)));
```

$$hp := B\left(\left(1+\cos(\Theta)^2\right)\left(G1\cos(-2\,phi\_orb + 2\,\Phi) + 2\,G2\sin(-2\,phi\_orb + 2\,\Phi)\right) - \sin(\Theta)^2\,G3\right)$$

$$hx := 2\,B\cos(\Theta)\left(2\,G2\cos(-2\,phi\_orb+2\,\Phi) - G1\sin(-2\,phi\_orb+2\,\Phi)\right) \tag{3}$$

Here, B=Btmp, but we leave it unassigned for now so we can rescale the overall amplitude easily. Below, A is an overally adjustment parameter (A=1 in reality).

```
> Btmp:=A*eta*Mtot/R;
```

$$Btmp := \frac{A\,\eta\,Mtot}{R} \tag{4}$$

```
> G1:= -1/p_M*(2+3*et*cos(v)+et^2*cos(2*v));
  G2:=1/p_M*(1+et*cos(v))*et*sin(v);
  G3:=1/p_M*et*(et+cos(v));
```

$$G1 := -\frac{2+3\,et\cos(v)+et^2\cos(2\,v)}{p\_M}$$

$$G2 := \frac{(1+et\cos(v))\,et\sin(v)}{p\_M}$$

$$G3 := \frac{et\,(et+\cos(v))}{p\_M} \tag{5}$$

where p_M = p/M.

NOTE THIS EQN BELOW WAS FIXED:

```
> phi_orb:=v+pomega;
```

$$phi\_orb := v + pomega \tag{6}$$

Explicit eqn for r/M is needed to plot the orbit:

```
> r_M:=p_M/(1+et*cos(v));
```

$$r\_M := \frac{p\_M}{1 + et\cos(v)} \tag{7}$$

Enter ODE system:

The PERIFLAG = 0 or 1 (latter if periastron advance is turned on); RRFLAG = 0 or 1 (latter if radiation reaction is turned on)

```
> dvdt:=1/M*1/p_M^(3/2)*(1+et*cos(v))^2;
  detdt:=-eta/15/M*1/p_M^4*et*(1-et^2)^(3/2)*(304+121*et^2)*RRFLAG;
  dpomegadt:=PERIFLAG*3/M/p_M^(5/2)*(1-et^2)^(3/2);
```

$$dvdt := \frac{(1 + et\cos(v))^2}{M\, p\_M^{3/2}}$$

$$detdt := -\frac{1}{15}\,\frac{\eta\, et\left(-et^2+1\right)^{3/2}\left(121\, et^2+304\right) RRFLAG}{M\, p\_M^4}$$

$$dpomegadt := \frac{3\, PERIFLAG\left(-et^2+1\right)^{3/2}}{M\, p\_M^{5/2}} \tag{8}$$

```
> p_M:=p0_M/C0*et^(12/19)*(304+121*et^2)^(870/2299);
  C0:=e0^(12/19)*(304+121*e0^2)^(870/2299);
```

$$p\_M := \frac{p0\_M\, et^{12/19}\left(121\, et^2+304\right)^{\frac{870}{2299}}}{C0}$$

$$C0 := e0^{12/19}\left(121\, e0^2+304\right)^{\frac{870}{2299}} \tag{9}$$

```
> eq_v:=diff(v(t),t)=subs(et=et(t),v=v(t),dvdt);
```

$$eq\_v := \frac{d}{dt}\, v(t) = \frac{(1 + et(t)\cos(v(t)))^2}{M\left(\frac{p0\_M\, et(t)^{12/19}\left(121\, et(t)^2+304\right)^{\frac{870}{2299}}}{e0^{12/19}\left(121\, e0^2+304\right)^{\frac{870}{2299}}}\right)^{3/2}} \tag{10}$$

```
> eq_et:=diff(et(t),t)=subs(et=et(t),v=v(t),detdt);
```

$$eq\_et := \frac{d}{dt}\, et(t) = -\frac{1}{15}\,\frac{\eta\, e0^{48/19}\left(121\, e0^2+304\right)^{\frac{3480}{2299}}\left(-et(t)^2+1\right)^{3/2} RRFLAG}{M\, p0\_M^4\, et(t)^{29/19}\left(121\, et(t)^2+304\right)^{\frac{1181}{2299}}} \tag{11}$$

```
> eq_pomega:=diff(pomega(t),t)=subs(et=et(t),v=v(t),dpomegadt);
```

(12)

$$eq\_pomega := \frac{d}{dt}\, pomega(t) = \frac{3\, PERIFLAG \left(-et(t)^2+1\right)^{3/2}}{M \left(\frac{p0\_M\, et(t)^{12/19} \left(121\, et(t)^2+304\right)^{\frac{870}{2299}}}{e0^{12/19} \left(121\, e0^2+304\right)^{\frac{870}{2299}}}\right)^{5/2}} \quad \textbf{(12)}$$

Enter in initial condition eqs:

```
> ic_v:=v(0)=0;
  ic_et:=et(0)=e0;
  ic_pomega:=pomega(0)=pomega0;
```

$$ic\_v := v(0) = 0$$

$$ic\_et := et(0) = e0$$

$$ic\_pomega := pomega(0) = pomega0 \quad \textbf{(13)}$$

Here we define the initial p_0/M in terms of the starting GW frequency and initial eccentricity, e0. To do this we start with following relationship between the p0 and the initial orbital frequency:

NOTE MODIFICATION HERE:

```
> p0_M:=(1-e0)^2/(2*Pi*M*forb0)^(2/3);
```

$$p0\_M := \frac{1}{2}\, \frac{(1-e0)^2\, 2^{1/3}}{(\pi\, M\, forb0)^{2/3}} \quad \textbf{(14)}$$

NOTE MODIFICATIONS IN SEVERAL LINES BELOW (IGNORE LINES THAT ARE COMMENTED OUT WITH #):

Next, we relate forb0 to the GW frequency of peak emission as follows:

```
> forb0:=fgw0/nMAX;
  #forb0:=fgw0/nPEAK;
  #forb0:=fgw0/2;
```

$$forb0 := \frac{fgw0}{nMAX} \quad \textbf{(15)}$$

```
> nMAX:=subs(et=e0,1+(-1.21197202522301*et^3+.644633617827521*
  et^2+2.87300320459655*et+1)/(1-et)^1.50727697024794);

  nPEAK:=subs(et=e0,1+(-5.12572928292396*et^3+8.06106912808853*et^2
  -2.68924859926098*et+1)/(1-et)^1.56490221994572);
```

$$nMAX := 1 + \frac{-1.21197202522301\, e0^3 + 0.644633617827521\, e0^2 + 2.87300320459655\, e0 + 1}{(1-e0)^{1.50727697024794}}$$

$$nPEAK := 1 + \frac{-5.12572928292396\, e0^3 + 8.06106912808853\, e0^2 - 2.68924859926098\, e0 + 1}{(1-e0)^{1.56490221994572}} \quad \textbf{(16)}$$

[The above function is derived in a separate worksheet, where we fit the above function to the peaks of the gn(et) function from Peters & Matthews.

The ending condition will be when p(t)/M=6+2et(t) is satisfied. That is equivalent to the following equation being zero:

```
> stopeqn:=subs(et=et(t),p_M-6-2*et)=0;
```

$$stopeqn := \frac{1}{2}\left((1-e0)^2\, 2^{1/3}\, et(t)^{12/19}\left(121\, et(t)^2+304\right)^{\frac{870}{2299}}\right) \Bigg/ \left(\left(\left(\pi\, M\, fgw0\right) \Big/ \left(1 + \frac{-1.21197202522301\, e0^3 + 0.644633617827521\, e0^2 + 2.87300320459655\, e0 + 1}{(1-e0)^{1.50727697024794}}\right)\right)^{2/3} e0^{12/19}\left(121\, e0^2+304\right)^{\frac{870}{2299}}\right) - 6 - 2\, et(t) = 0 \quad \textbf{(17)}$$

Define the total mass in units of seconds (so time is in sec, freq in hertz); also define eta:

```
> M:=(m1sun+m2sun)*MSUN_SEC;
  MSUN_SEC:=4.925491025e-6;
  eta:=m1sun*m2sun/(m1sun+m2sun)^2;
```

$$M := (m1sun + m2sun)\ MSUN\_SEC$$

$$MSUN\_SEC := 0.000004925491025$$

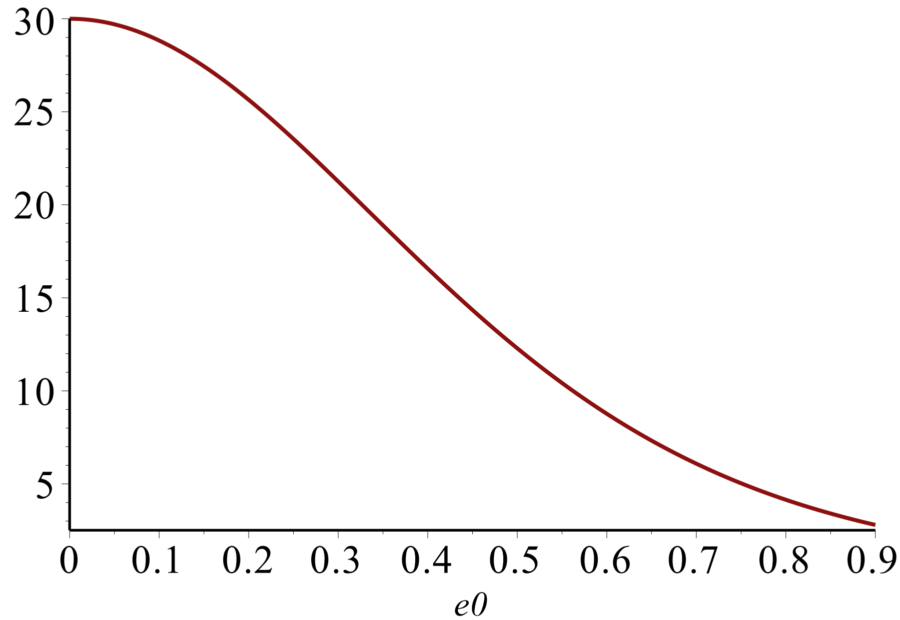
$$\eta := \frac{m1sun\ m2sun}{(m1sun + m2sun)^2} \quad \textbf{(18)}$$

NOTE NEW LINE HERE: THIS ALLOWS STARTING FREQUENCY TO CHANGE DEPENDING ON e0

```
> fgw0:=fgw0circ/(1+e0^2)^(4);
```

$$fgw0 := \frac{fgw0circ}{\left(e0^2+1\right)^4} \quad \textbf{(19)}$$

```
> plot([30/(1+e0^2)^(4)],e0=0..0.9);
```



Enter in constants needed to solve the ODEs **[CHOOSE new parameter fgw0circ=30 Hz as a fixed value; user does not choose]**

```
> odeconsts:=[e0=0.9,fgw0circ=30,m1sun=3,m2sun=3,pomega0=0.3,
  PERIFLAG=1,RRFLAG=1];
```

$$odeconsts := [e0 = 0.9, fgw0circ = 30, m1sun = 3, m2sun = 3, pomega0 = 0.3, PERIFLAG = 1, RRFLAG = 1] \tag{20}$$

NEW CHECK: Check that ICs don't violate stop condition (should be "true"). NOTE THAT THIS IS VIOLATED FOR LARGE MASSES AND LARGE e0. IF FALSE, SAY "THIS SYSTEM IS WITHIN IT'S LAST STABLE ORBIT; CHOOSE LOWER MASSES OR A LOWER ECCENTRICITY."

```
> evalb(evalf(subs(odeconsts,p0_M>6+2*e0)));
```

$$true \tag{21}$$

```
> solntmp:=dsolve(subs(odeconsts,[eq_v,eq_et,eq_pomega,ic_v,ic_et,
  ic_pomega]),[v(t),et(t),pomega(t)],type=numeric,stop_cond=[subs
  (odeconsts,stopeqn)],method=rkf45,output=listprocedure,maxfun=-1);
```

$$solntmp := [t = \textbf{proc}(t) \ \ldots \ \textbf{end proc}, v(t) = \textbf{proc}(t) \ \ldots \ \textbf{end proc}, et(t) = \textbf{proc}(t) \ \ldots \ \textbf{end proc}, pomega(t) = \textbf{proc}(t) \ \ldots \ \textbf{end proc}] \tag{22}$$

```
> solntmp(1000);
Warning, cannot evaluate the solution further right of 8.4419321,
event #1 triggered a halt
Warning, cannot evaluate the solution further right of 8.4419321,
event #1 triggered a halt
Warning, cannot evaluate the solution further right of 8.4419321,
event #1 triggered a halt
Warning, cannot evaluate the solution further right of 8.4419321,
event #1 triggered a halt
```

$$[t(1000) = 8.44193212247198, v(t)(1000) = 649.504561215250, et(t)(1000) = 0.0694273331867109, pomega(t)(1000) = 94.1169344898837] \tag{23}$$

```
> tstop:=8.4419321;
```

```
#tstop:=5;
```

$tstop := 8.4419321$ **(24)**

```
> soln:=dsolve(subs(odeconsts,[eq_v,eq_et,eq_pomega,ic_v,ic_et,
ic_pomega]),[v(t),et(t),pomega(t)],type=numeric,stop_cond=[subs
(odeconsts,stopeqn)],method=rkf45,output=listprocedure,maxfun=-1,
range=0..tstop);
```

$soln := [t = \mathbf{proc}(t)\ \ldots\ \mathbf{end\ proc}, v(t) = \mathbf{proc}(t)\ \ldots\ \mathbf{end\ proc}, et(t) = \mathbf{proc}(t)\ \ldots\ \mathbf{end\ proc},$ **(25)**

$pomega(t) = \mathbf{proc}(t)\ \ldots\ \mathbf{end\ proc}]$

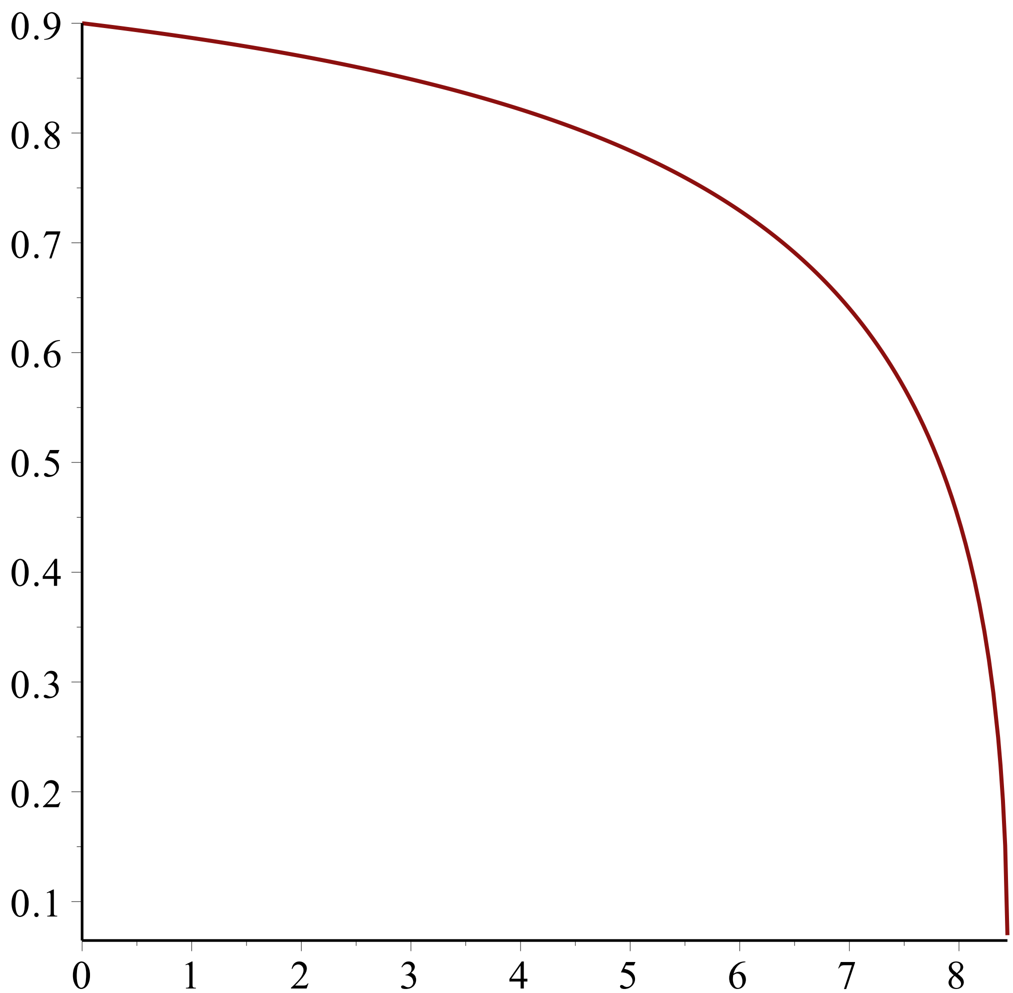
Define the functions:

```
> v_t:=subs(soln,v(t)):
et_t:=subs(soln,et(t)):
pomega_t:=subs(soln,pomega(t)):
```

```
> et_t(100);
Warning, extending a solution obtained using the range argument
with 'maxfun' large or disabled is highly inefficient, and may
consume a great deal of memory. If this functionality is desired,
it is suggested to call dsolve without the range argument
Warning, cannot evaluate the solution further right of 8.4419321,
event #1 triggered a halt
```

0.0694273331867103 **(26)**

```
> plot(et_t,0..tstop);
```




```
> #plot(et_t,0.999*tstop..tstop);
```

Now we plug those solutions into other equations to construct the orbit and the waveform:

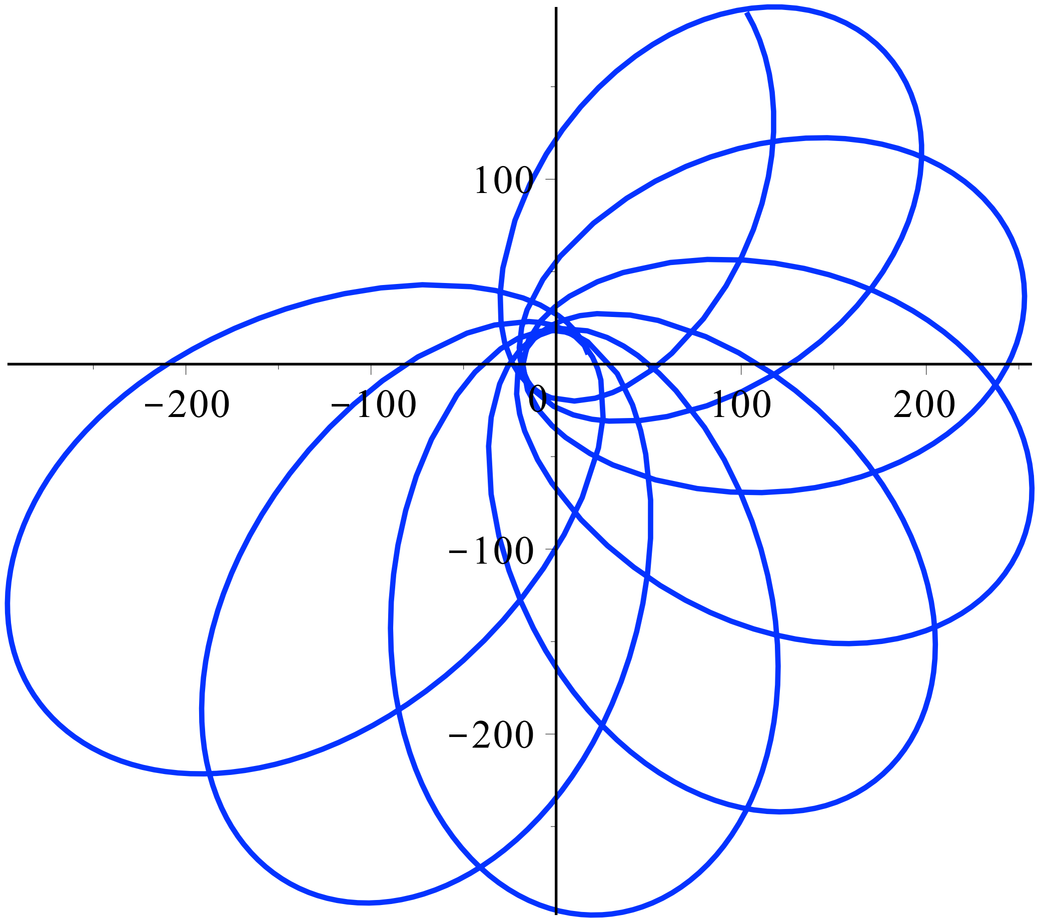
```
> r_t:=T->evalf(subs(odeconsts,et=et_t(T),v=v_t(T),r_M)):
  phi_orb_t:=T->evalf(subs(odeconsts,et=et_t(T),v=v_t(T),pomega=
  pomega_t(T),phi_orb)):
```

Construct x and y coordinates of the orbit:

```
> x_t:=T->r_t(T)*cos(phi_orb_t(T)):
  y_t:=T->r_t(T)*sin(phi_orb_t(T)):
> x_t(10);
Error, (in et_t) cannot evaluate the solution further right of 
8.4452166, probably a singularity
> r_t(0.99*tstop);
```

$$16.0537744384898 \quad (27)$$

```
> plot([x_t,y_t,0.0*tstop..0.3*tstop],color=blue,thickness=2,
  numpoints=500,scaling=constrained);
```




```
> hconsts:=[Theta=Pi/4,theta=Pi/4,phi=7*Pi/3,Phi=0,Psi=Pi/3];
```

$$hconsts := \left[\Theta = \frac{1}{4}\,\pi, \theta = \frac{1}{4}\,\pi, \phi = \frac{7}{3}\,\pi, \Phi = 0, \Psi = \frac{1}{3}\,\pi\right] \quad (28)$$

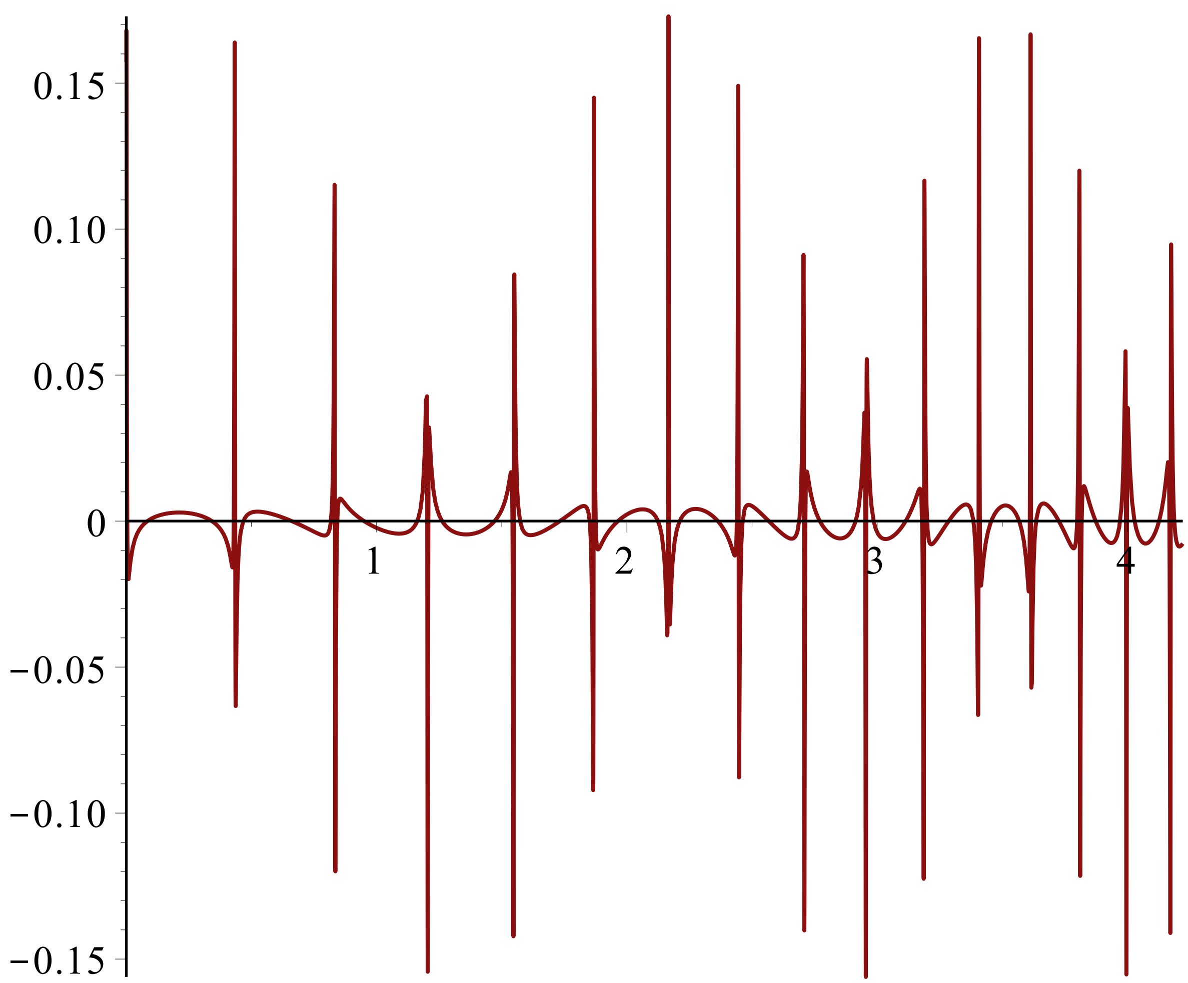
```
> h_t:=T->evalf(subs(B=1,hconsts,odeconsts,et=et_t(T),pomega=pomega_t
  (T),v=v_t(T),h));
```

$$h\_t := T \rightarrow evalf\left(subs\left(B = 1, hconsts, odeconsts, et = et\_t(T), pomega = pomega\_t(T), v = v\_t(T), h\right)\right) \quad (29)$$

```
> h_t(5);
```

$$-0.00439029331551145 \quad (30)$$

```
> plot(h_t,0.0*tstop..0.5*tstop,numpoints=500);
```




```
> 0.0*tstop..0.5*tstop;
```

$$0. ..4.22096605 \tag{31}$$

Generate time series points:

Need to choose an appropriately small timestep.

NOTE THAT I MADE SEVERAL CHANGES HERE; NOT ALL LINES BELOW ARE USED.

..

```
> Tgw_LSO:=1/nLSO/fgw_LSO;
  fgw_LSO:=1/Pi/M/6^(3/2);
```

$$Tgw\_LSO := \frac{1}{nLSO\, fgw\_LSO}$$

$$fgw\_LSO := \frac{1}{36}\ \frac{\sqrt{6}}{\pi\ (0.000004925491025\ m1sun + 0.000004925491025\ m2sun)} \tag{32}$$

```
> Tgw_INIT:=1/nINIT/fgw_INIT;
  fgw_INIT:=fgw0;
```

$$Tgw\_INIT := \frac{1}{nINIT\, fgw\_INIT}$$

$$fgw\_INIT := \frac{fgw0circ}{\left(e0^2 + 1\right)^4} \tag{33}$$

```
> Tgw_geomean:=sqrt(Tgw_LSO*Tgw_INIT);
```

(34)

$$Tgw\_geomean := \sqrt{6}\,\sqrt{\frac{\pi\,(0.000004925491025\,m1sun + 0.000004925491025\,m2sun)\,\sqrt{6}\,\left(e0^2+1\right)^4}{nLSO\;nINIT\;fgw0circ}} \tag{34}$$

```
> evalf(subs(odeconsts,nLSO=2,Tgw_LSO));
  evalf(subs(odeconsts,nINIT=4,Tgw_INIT));
  evalf(subs(odeconsts,nLSO=6,nINIT=4,Tgw_geomean));
```

0.0006822562696

0.08944026008

0.004510032453 **(35)**

HERE I USE THE MINIMUM OF TGW_LSO OR 1/1000 (WHICH IS THE MINIMUM SAMPLE PERIOD REQUIRED BY MAPLE'S AUDIO GENERATOR). You should experiment to see what values of nLSO>=2 are sufficient for resolution and speed (eg, try 2, 4, or 6).

```
> dt:=min(evalf(subs(odeconsts,nLSO=2,Tgw_LSO)),1/1000.0);
  1/dt;
```

$dt := 0.0006822562696$

1465.724896 **(36)**

If choosing a non-evolving orbit, instead let's try:

```
> #dt:=min(evalf(subs(odeconsts,nINIT=4,Tgw_INIT)),1/1000.0);
```

Generate t-values from 0 to tstop in steps of dt:

```
> tstop;
```

8.4419321 **(37)**

```
> t_pts:=[seq(0+i*dt,i=0..floor(tstop/dt))]:
> t_pts[nops(t_pts)];
  nops(t_pts);
```

8.441556824

12374 **(38)**

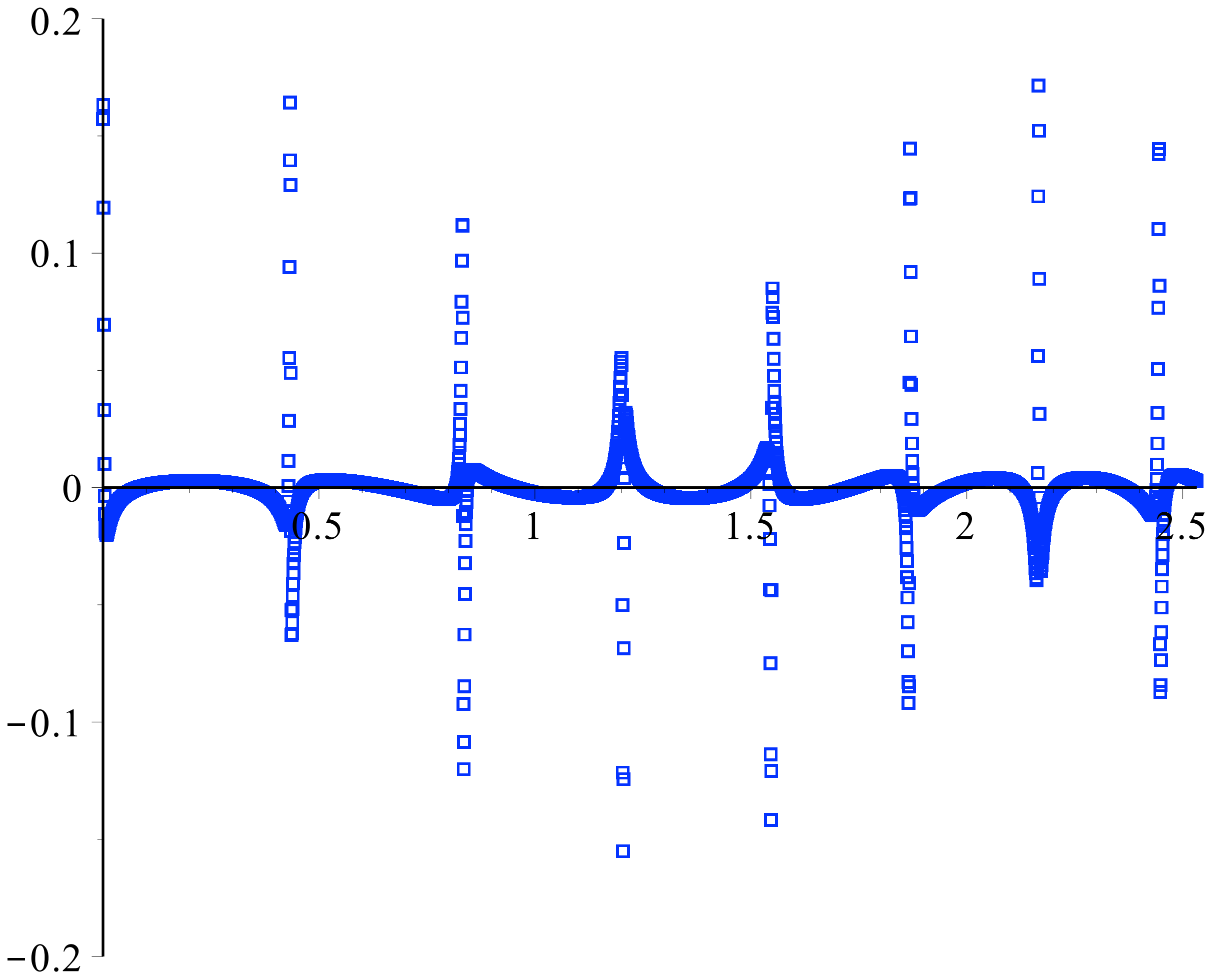
```
> st:=time():
  ht_pts:=[seq(h_t(t_pts[i]),i=1..nops(t_pts))]:
  evalf((time()-st)/60);
  st:=time():
```

0.193200000 **(39)**

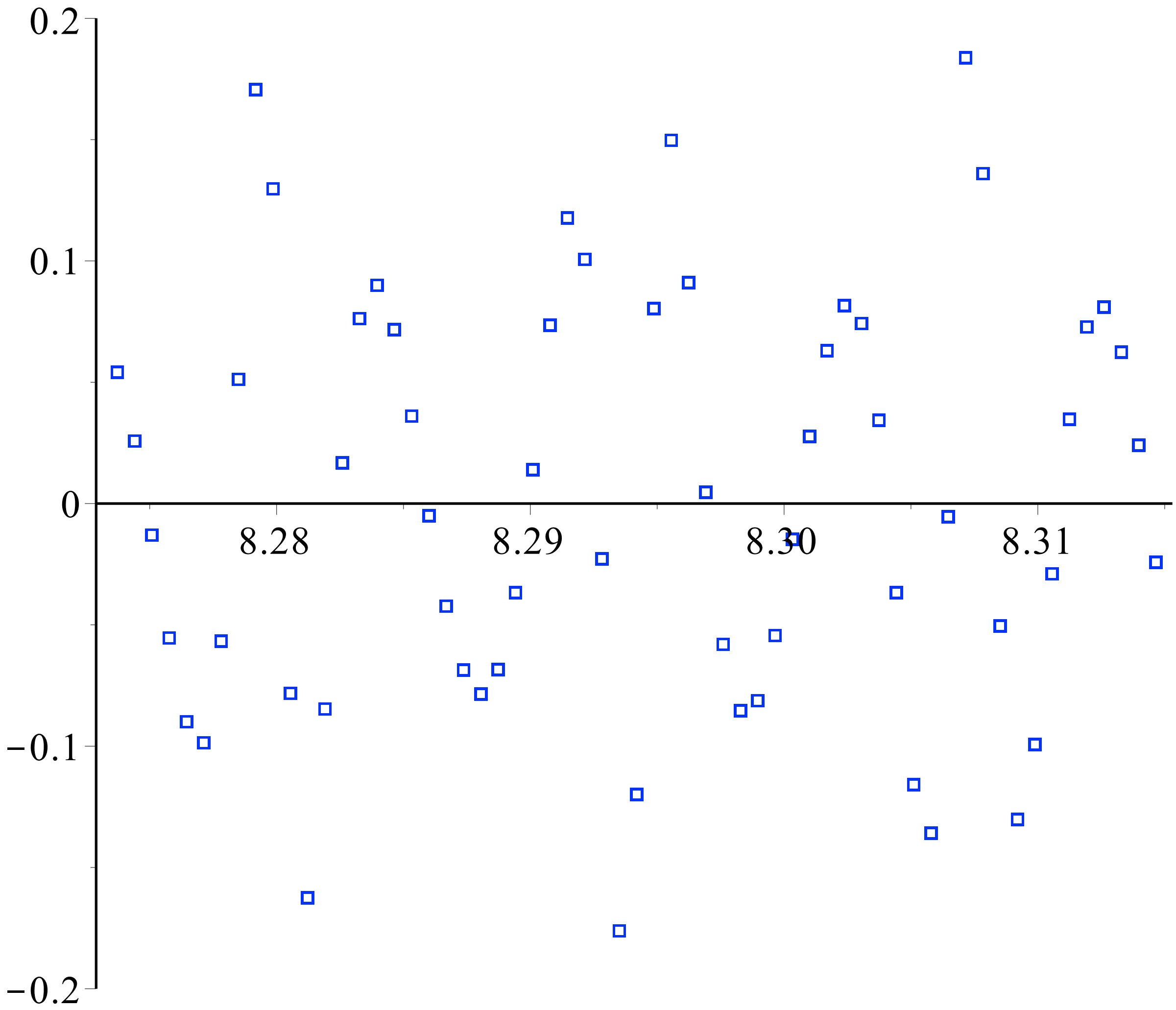
```
> tstop;
```

8.4419321 **(40)**

```
> t_h_pts:=[seq([t_pts[i],ht_pts[i]],i=1..nops(t_pts))]:
> pointplot(t_h_pts,symbol=box,connect=false,color=blue,view=[0.0*
  tstop..0.3*tstop,-0.2..0.2]);
  pointplot(t_h_pts,symbol=box,connect=false,color=blue,view=[0.98*
  tstop..0.985*tstop,-0.2..0.2]);
```



0.2
0.1
0
−0.1
−0.2
0.5
1
1.5
2
2.5




```
> odeconsts;
```

$[e0 = 0.9, fgw0circ = 30, m1sun = 3, m2sun = 3, pomega0 = 0.3, PERIFLAG = 1, RRFLAG = 1]$ **(41)**

Write out t-h data to file:

```
> f1:=fopen("t_h_e0=09_m1=m2=3_RR+PERI.txt",WRITE):
> writedata(f1,t_h_pts):
> fclose(f1):
```

```
> 1/dt;
```

$$1465.724896 \qquad \textbf{(42)}$$

```
> h_t(0);
```

$$0.157236860818794 \qquad \textbf{(43)}$$

Normalize array so maximum value is 1:

```
> htmax:=max([seq(abs(ht_pts[i]),i=1..nops(t_pts))]);
```

$$htmax := 0.338937903780839 \qquad \textbf{(44)}$$

```
> with(AudioTools):
```

```
> aud_array:=ht_pts/htmax:
```

```
> aud_create:=Create(aud_array,channels=1,rate=floor(1/dt)):
> Write("aud_e0=09_m1=m2=3_RR+PERI.wav",aud_create):
```