Stepwise precession of a double elastic-spherical pendulum

Luis Sainz de los Terreros,
Universidad Politécnica de Madrid,
Spain.
luis.sainzdelosterreros@upm.es

Introduction

The physical effect known as stepwise precession is a surprising fact that takes place in the movement of an elastic pendulum (see Ref. 1-2), i.e., a mass at the endpoint of an elastic (not-bending) spring, the other point attached to a fixed pivot. When this pendulum is initially set near the downward equilibrium vertical configuration, with the length of the spring stretched by the gravity force, there is a chance to move in such a way that the springing vibrational mode of movement and the rotational "swing" mode interchange periodically their respective energies. The nonlinear dynamics of this system shows a two-fold astonishing fact: Following a properly chosen initial impulse, the initial vibrational energy begins to decrease after a lapse of time and tranferred to a progressive larger amplitude swing motion, this last, when projected on the x-y plane, resembles instantaneous eliptical rotational motion but with an increasing major-axis amplitude and larger excentricity. The second surprise is that this interchange between vibrational-swing kinetic energies is reversed and then a revival of a new cycle of similar movement takes place, but now on a *new* swinging plane. This periodic precession of the swing plane goes on with a finite and predictable change in the angle of swinging plane, and therefore, so to speak, in a stepwise precession form.

The package *ElasticSphericalPendulum* attempts to find and simulate the same dynamical behaviour for the natural extension of a **double elastic-spherical pendulum**. , i.e., a rigid spherical pendulum hanging from an elastic pendulum. It is to be noticed that the other possible configuration, that is, the spherical-elastic double pendulum (with the elastic spring pendulum hanging from the fixed-length spherical pendulum) doesn't seem to manifest so neatly such an extraordinary phenomenum.

The key point is that the stepwise precession effect arises in a coherent way on the movements of both pendulum masses, only if both pendulums are **initially aligned near the downward equilibrium configuration**. Beside this, the frequency of vibrational motion (which depends on the elastic Hooke constant K and the masses M1 and M2) must be set next to a precise value with a precise ratio to the frequency of the swing pendular frequency, this condition known as *the resonant condition*.

ElasticSphericalPendulum

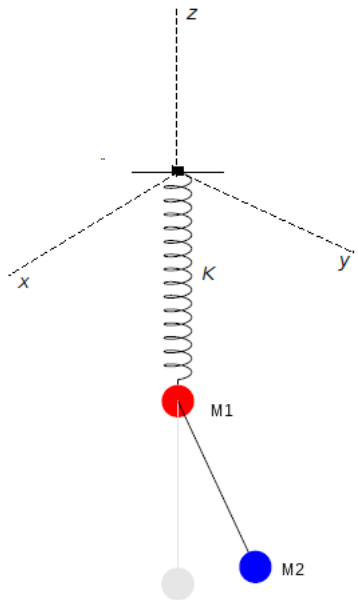


Fig. 1: A double elastic-spherical pendulum

>
>

Loading the package

Starting NOTE: This package uses external data files. These are generated after running any example. Each example numerical simulation includes two principal steps:
1. *Generation of the input parameters and example data.*
2. *Numerical integration of the coupled, nonlinear second-order differential equations.*

To load the package just follow the next commands. You can also run the entire worbook document, but for a better understanding of the input and output data and its significance it's better to proceed step by step. Two independent examples are included, and should be run after a new restart and package load.

```
> restart;
> with(plots) :
> read("ElasticSphericalPendulum_pck_code.mpl");
```

$$ElasticSphericalPendulum := \textbf{module}(\ )\ \ldots\ \textbf{end module} \quad (2.1)$$

```
> with(LinearAlgebra) :
> Digits := 15;
```

$$Digits := 15 \quad (2.2)$$

The package is loaded: the commands in the package are listed.

```
> with(ElasticSphericalPendulum)
```

$$[ElasticSphericalDsolver, generateInitial]$$

>

Initial data and parameters

> $\varphi := \dfrac{\arctan\left(\dfrac{0.006}{1.012}\right)\cdot 180}{evalf(\mathrm{Pi})}$; **# selected to provide the designed cartesian coordinates of the masses.**

$$\varphi := 0.339694317312642 \tag{3.1}$$

> $S := Vector[row]\left(3, \left[\sin\left(\dfrac{\varphi\cdot evalf(\mathrm{Pi})}{180.0}\right), 0.0, -\cos\left(\dfrac{\varphi\cdot evalf(\mathrm{Pi})}{180.0}\right)\right], datatype = float[8]\right)$

$$S := \begin{bmatrix} 0.0059287496 & 0. & -0.9999824248 \end{bmatrix} \tag{3.2}$$

> $S \cdot S$

$$1.0000000000 \tag{3.3}$$

> $q := 1.01201778640496\cdot S$;

$$q := \begin{bmatrix} 0.0060000000 & 0. & -1.0120000000 \end{bmatrix} \tag{3.4}$$

> $q + S$

$$\begin{bmatrix} 0.0119287496 & 0. & -2.0119824248 \end{bmatrix} \tag{3.5}$$

> omega := $Vector[row]([0.00483201581027668, 0.0, 0.0])$;

> $CrossProduct(\mathrm{omega}, q)$; **# gives the out-of-plane y- coordinate initial velocity for mass 1**

$$\omega := \begin{bmatrix} 0.0048320158 & 0. & 0. \end{bmatrix}$$

$$\begin{bmatrix} -0. & 0.0048900000 & 0. \end{bmatrix} \tag{3.6}$$

> $CrossProduct(\mathrm{omega}, S)$;

$$\begin{bmatrix} -0. & 0.0048319309 & 0. \end{bmatrix} \tag{3.7}$$

> $g := evalf(\mathrm{Pi}^2)$: $K := 41.0$: $L0 := 1.20 - \dfrac{2\cdot g}{K}$:'$L0$'$= L0$; **# parameters. Masses are unitary (M1=M2=1Kg). L=1.20 is the equilibrium spring length**

$$L0 = 0.7185558829 \tag{3.8}$$

>

> $generateInitial(K, L0, 0.006, 0.0, -1.01200000000000, 0.00, 0.00489, 0.0, S, \mathrm{omega})$

$$\text{"Equilibrium point spring length"}, 1.2000000000 \tag{3.9}$$

> $readdata(\text{"./dataSYMPKE.dta"}, [float\ \$\ 12])$; **# generated initial data**

$$[[0.0059287496, 0., -0.9999824248, 0.0060000000, 0., -1.0120000000, 0.0245090341, -0., 0.0001453105, 0.0000592140, -28.0800525278, -28.0799933138]] \tag{3.10}$$

>

The initial generated data is as follows:

1. The six cartesian components of the vectors S and q;
2. The three components of the total initial angular momentum
3. Kinetic, potential and total (conserved) energy

Numerical integration. Main results

The input data to the dynamic equations solver, are:

1. *Nsteps* : Number of integration steps (the time length of each step is $DeltaT = 0.0025\ s.$)
2. *K* : The spring stiffness constant.
3. *L0* : The natural length of the spring without stretching.

```
> Nsteps := 85000 : st := time( ) : ElasticSphericalDsolver(Nsteps, K, L0); time( ) - st
```

$$297.6880000000 \tag{4.1}$$

```
> DeltaT·85000; #   total length of time integration  (in secs).
```

$$212.5000000000 \tag{4.2}$$

The output consists on 12 items:

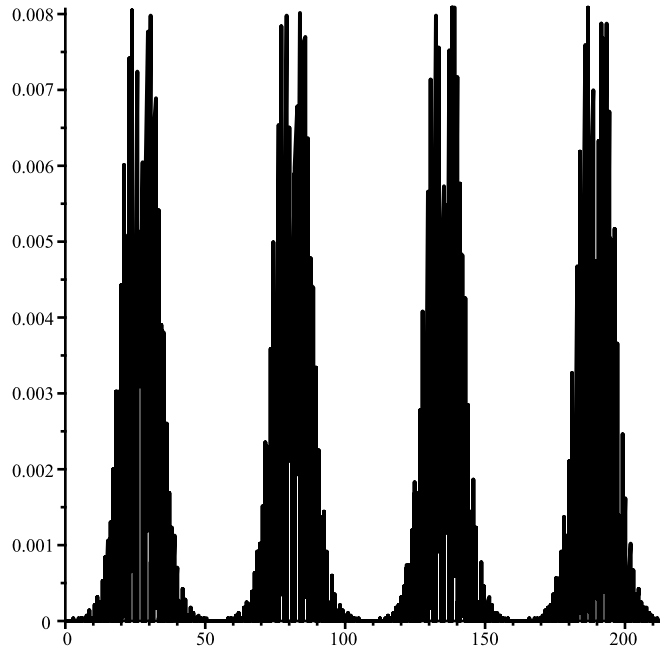
- the six cartesian components of the vectors S and q;
- kinetic, potential and total energy;
- the oscilatting variable length of spring vector q in the direction of the spherical pendulum, $\gamma(t) = q \cdot S$.

- The swing kinetic energy term $\dot{S}.\dot{S} = (\dot{S})^2$
- The alignment control variable, signed by the square modulus of the vector product $(S \times q)^2$

```
> sols := readdata("./PKESYM.dta", [float $ 12]) :
> nops(sols);
```

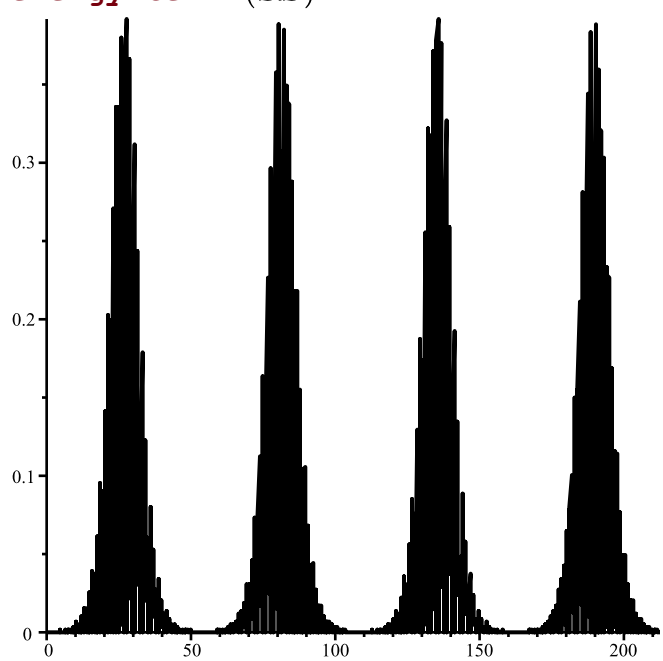
$$85001 \tag{4.3}$$

```
> plots[pointplot]([seq([i * DeltaT, sols[i, 12]], i = 1 .. Nsteps)], connect = true, color
      = "Black") ;
  # alignment control variable
```



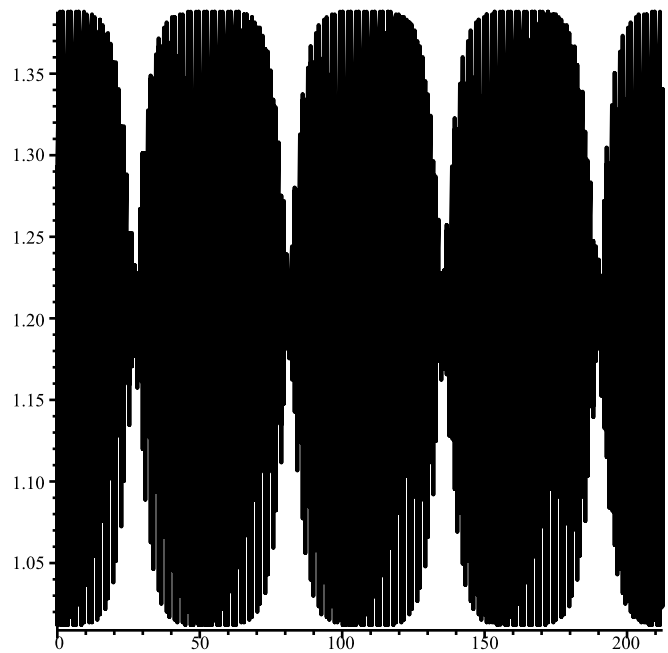
> $plots[pointplot]([seq([i*DeltaT, sols[i, 11]], i = 1 .. Nsteps)], connect = true, color = \text{"Black"})$;

 # swing kinetic energy term $(\dot{S}.\dot{S})$



> $plots[pointplot]([seq([i*DeltaT, sols[i, 10]], i = 1 .. Nsteps)], connect = true, color = \text{"Black"})$;

 # stretching oscillations of the spring



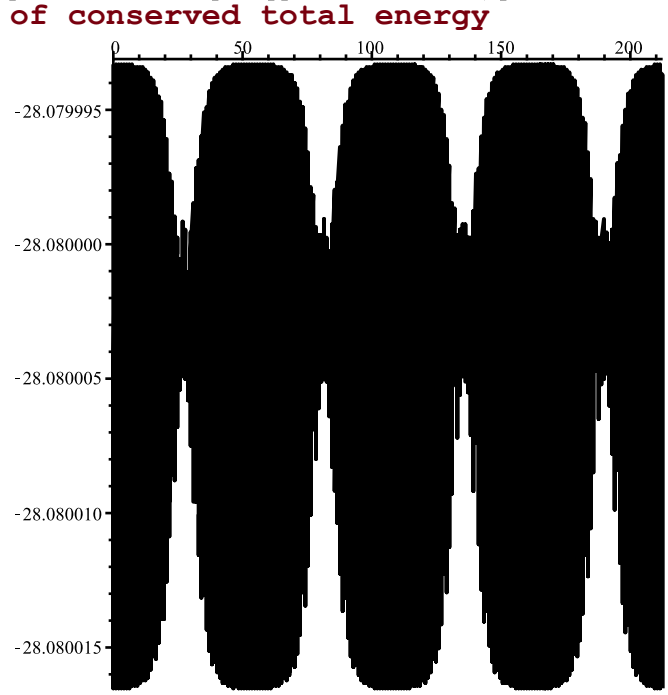
> $\frac{evalf(2\cdot \mathrm{Pi})\cdot 151}{Nsteps\cdot DeltaT}$; $\omega Z = \sqrt{\frac{K}{2}}$;

The amplitude and frequency of oscillations of the spring is signed by $\gamma(t) = q \cdot S$ and is well approximated by ωZ

$$4.4647575595$$

$$\omega Z = 4.5276925691 \tag{4.4}$$

> $plots[pointplot]([seq([i * DeltaT, sols[i, 9]], i = 1 .. Nsteps)], connect = true, color = \text{"Black"})$; **# test of quality of conserved total energy**



> $gred := plots[pointplot]([seq([sols[i,4], sols[i,5]], i = 1 .. Nsteps)], connect = true, color = \text{"LightCoral"}, scaling = constrained)$

[Length of output exceeds limit of 1000000] **(4.5)**

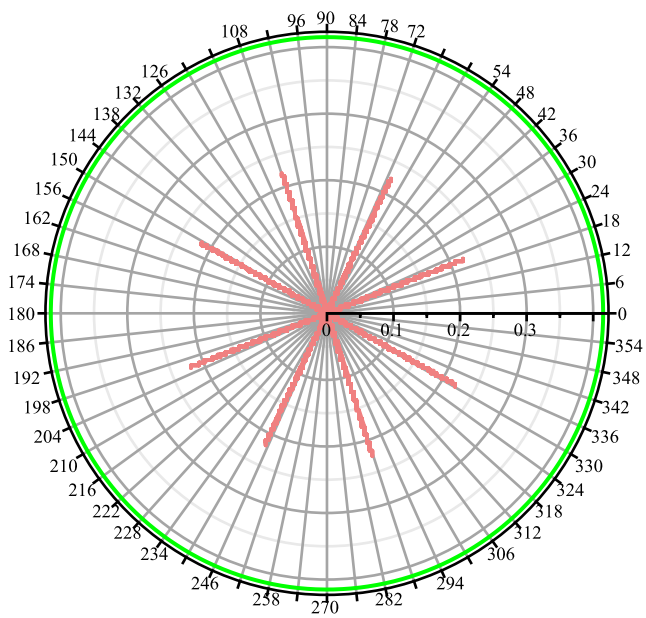
> $gblue := plots[pointplot]([seq([sols[i,4] + sols[i,1], sols[i,5] + sols[i,2]], i = 1 .. Nsteps)], connect = true, color = \text{"SteelBlue"}, scaling = constrained)$

[Length of output exceeds limit of 1000000] **(4.6)**

> $circ := plots[polarplot](0.415, angularunit = degrees, tickmarks = [\text{"decimalticks"}, 60], color = green) :$

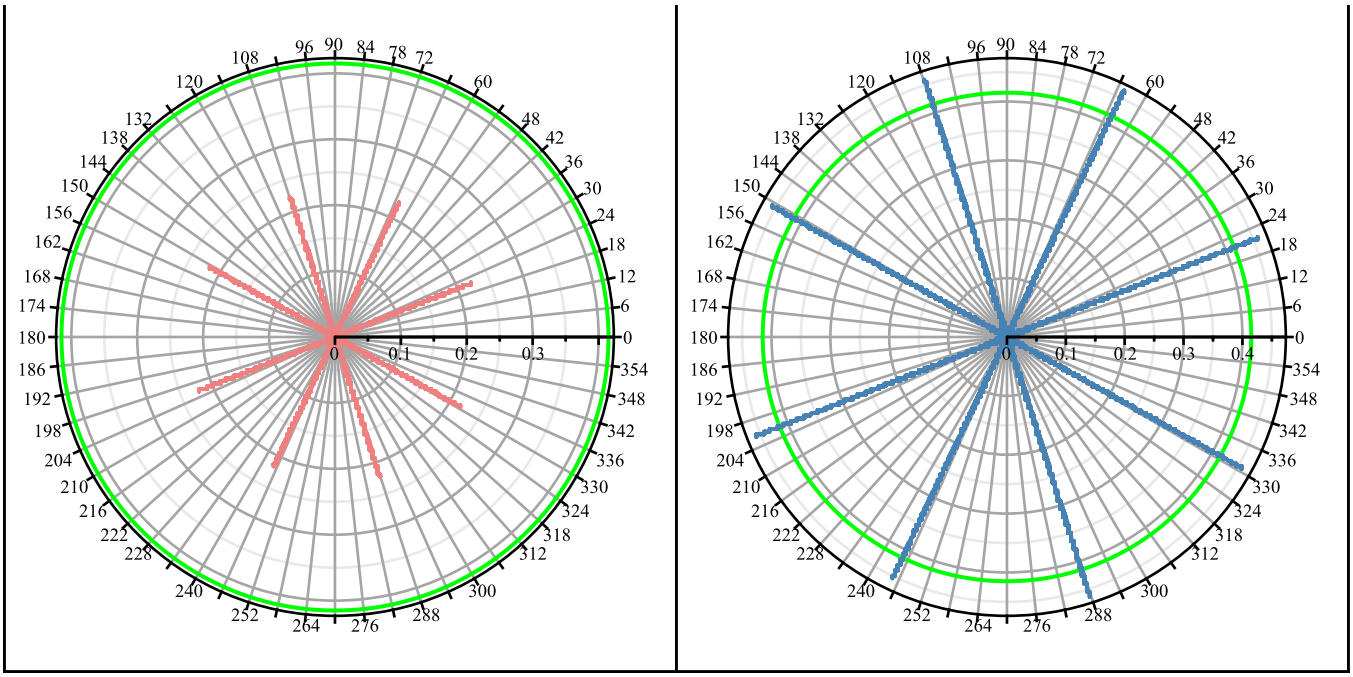
> $circ2 := plots[polarplot](0.415, angularunit = degrees, tickmarks = [\text{"decimalticks"}, 60], color = green) :$

> $plots[display](circ, gred);$



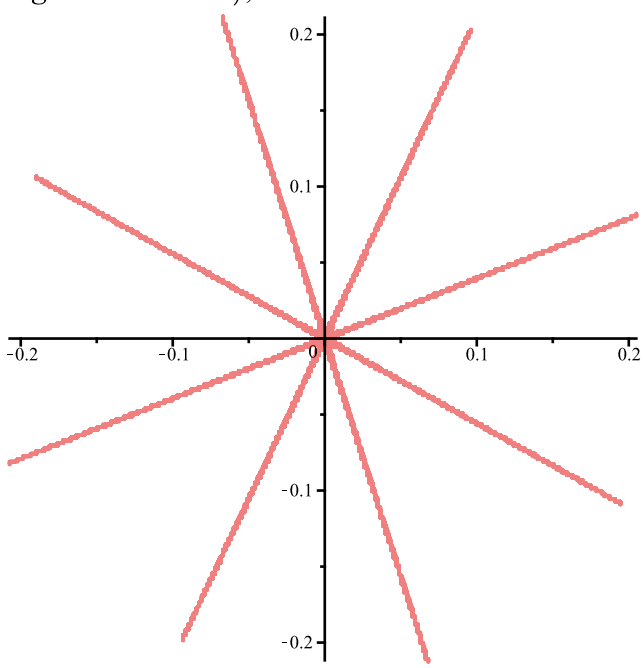
swing and stepwise precession with polar diagram

> $plots[display](circ, gred);$	> $plots[display](circ2, gblue);$



> $gr1 := plots[pointplot]([seq([sols[i,4], sols[i,5]], i = 1 .. Nsteps)], connect = true, color = \text{"LightCoral"}, scaling = constrained):$

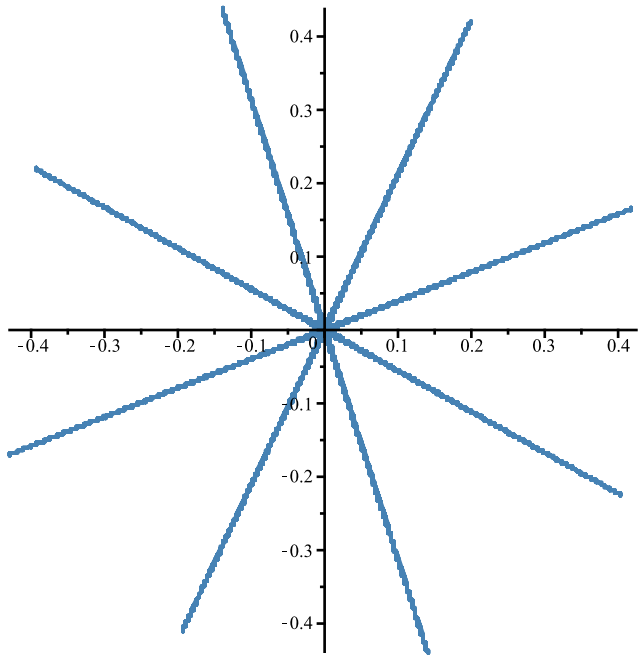
> $plots[pointplot]([seq([sols[i,4], sols[i,5]], i = 1 .. Nsteps)], connect = true, color = \text{"LightCoral"}, scaling = constrained);$



> $\#\, plots[pointplot]([seq([sols[i,4], sols[i,5]], i = 1 .. Nsteps)], connect = true, color = \text{"LightCoral"}, scaling = constrained)$

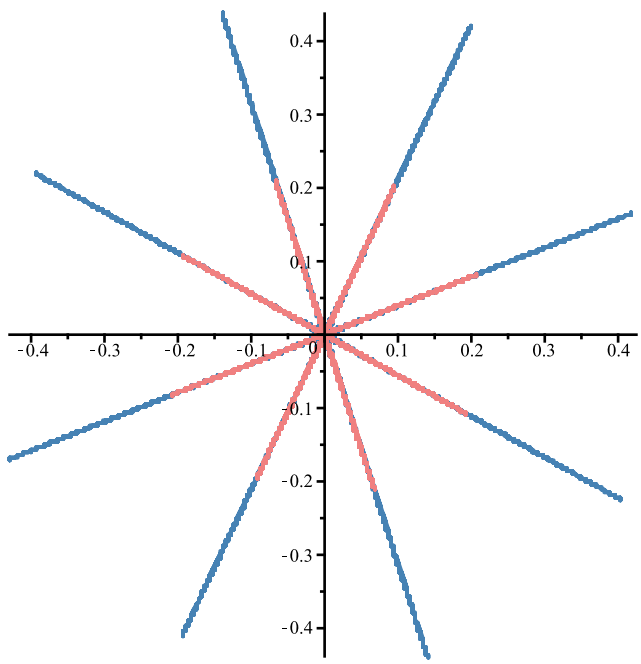
> $plots[pointplot]([seq([sols[i,4] + sols[i,1], sols[i,5] + sols[i,2]], i = 1 .. Nsteps)], connect = true, color = \text{"SteelBlue"}, scaling = constrained);$

You can combine these xy- projections, and convince yourself that the two pendulums evolves in phase and aligned.

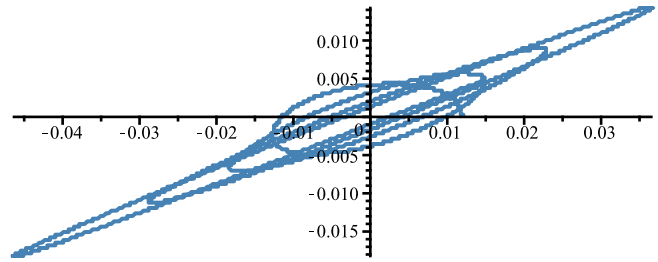


> $gr2 := plots[pointplot]([seq([sols[i,4] + sols[i,1], sols[i,5] + sols[i,2]], i = 1 .. Nsteps)], connect = true, color = \text{"SteelBlue"}, scaling = constrained)$:

> $plots[display](gr2, gr1)$;

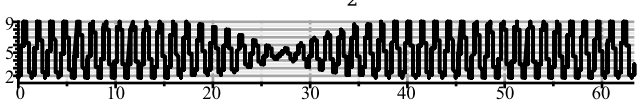


The first stages (for the first seconds of time) of the swing movement are shown at the figure at right, for the mass 2. The definite direction of the swing makes its appearance.	> $plots[pointplot]([seq([sols[i,4] + sols[i,1], sols[i,5] + sols[i,2]], i = 1 .. 4330)], connect = true, color = \text{"SteelBlue"}, scaling = constrained)$;

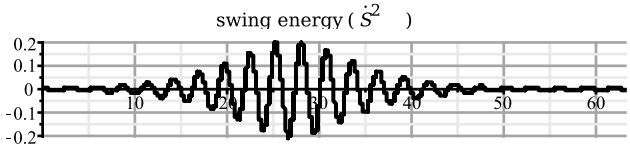


It's to be noticed how the variation between swing energy and vibrational energy evolves with mutual interchange:

vibrational energy ($\frac{K}{2}\cdot(|q|-L0)^2$)



swing energy ($\dot{S}^2$)



> *plots*[*pointplot*]([*seq*([*i* * *DeltaT*, *sols*[*i*, 11]], *i* = 1 .. 4330)], *connect* = *true*, *color* = "Black", *scaling* = *unconstrained*) ;

The swing energy grows slowly as the amplitude along a definite axis direction takes higher values.

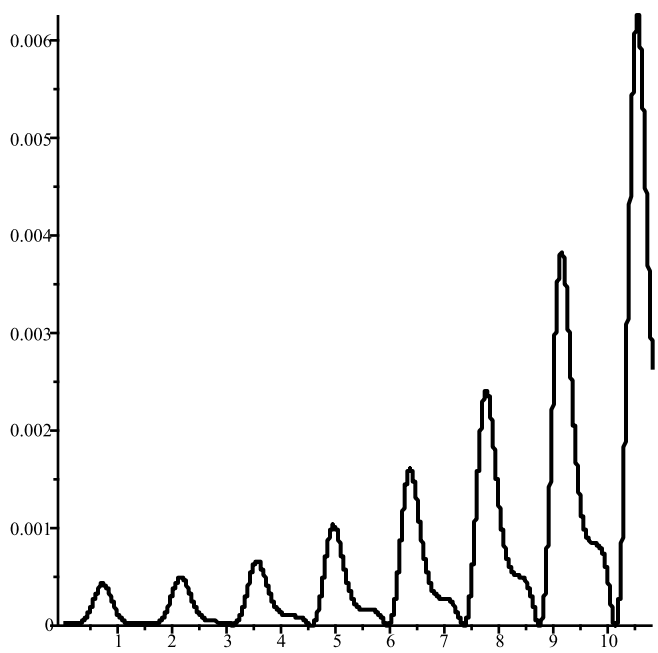
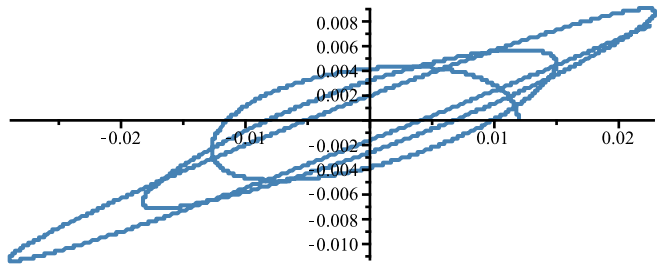


Table 1: Initial stage of swing movement

> *plots*[*pointplot*]([*seq*([*sols*[*i*, 4] + *sols*[*i*, 1], *sols*[*i*, 5] + *sols*[*i*, 2]], *i* = 1 .. 3300)], *connect* = *true*, *color* = "SteelBlue", *scaling* = *constrained*)



A note about the mathematical model and numerical simulation

The system studied in this worbook has five degrees of fredom, with six cartesian coordinates for the two position vectors q and S and the constraint given by the fixed length of the spherical pendulum. The full set of coupled second order differential equations are highly nonlinear and needs a numerical integration algorithm. We have selected a symplectic algorithm under the *Shake-Verlet* form, which assures the invariant energy conservation in time, as well as constant z-component of the total angular momentum.

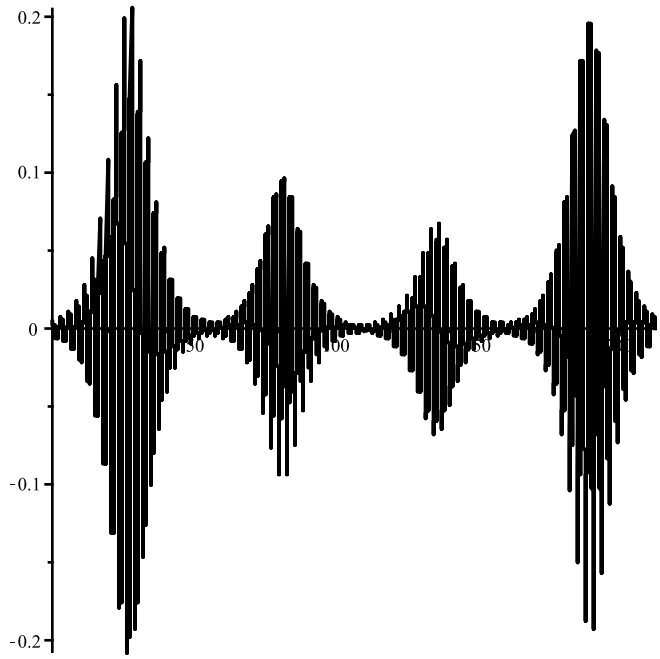
$$\textbf{(1)}\ \ddot{q} = -g\cdot\vec{k} - K\cdot\left(1 - \frac{L0}{|q|}\right)\cdot q - 2\cdot\lambda\cdot S; \qquad \textbf{(2)}$$
$$\ddot{S} = K\cdot\left(1 - \frac{L0}{|q|}\right)\cdot q + 4\cdot\lambda\cdot S;$$

where the Lagrange multipler may be written with the aid of the constraint condition $S\bullet\dot{S} = 0$.

The z-component of the movement of both masses, may be well approximated in the linear approximation, as harmonic oscillating funtions with frecuency $\omega_Z = \sqrt{\frac{K}{2}}$. This approximation assumes the horizontal components q_x and q_y are small enough and negligible. In contrast to this, the horizontal components of the rotational "swing" movement *cannot* be approximated by a linear differential equation, the coupling kinetic nonlinear term $\dot{S}^2$ comes now into play and cannot be ignored.

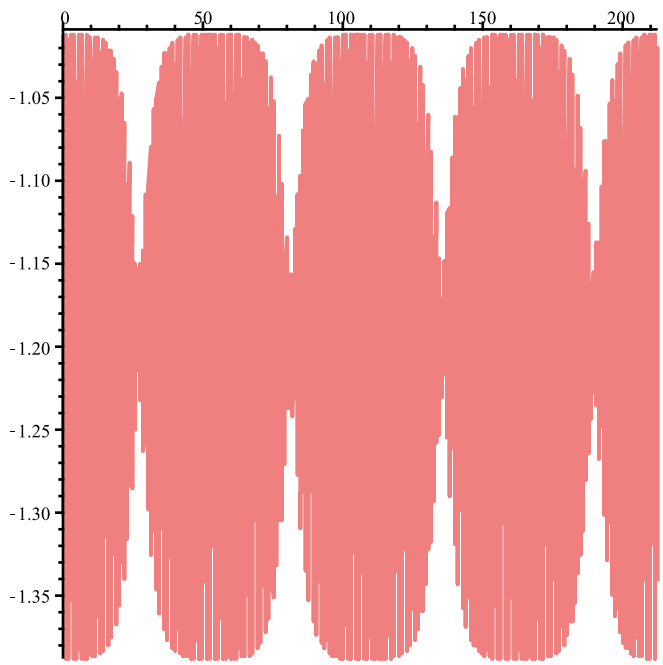
All this may be understood, once the numerical oscillating solutions are represented in time:

```
> plots[pointplot]([seq([i*DeltaT, sols[i, 4]], i = 1 .. Nsteps)], connect = true, color = "Black"); # qx versus time (mass 1)
```



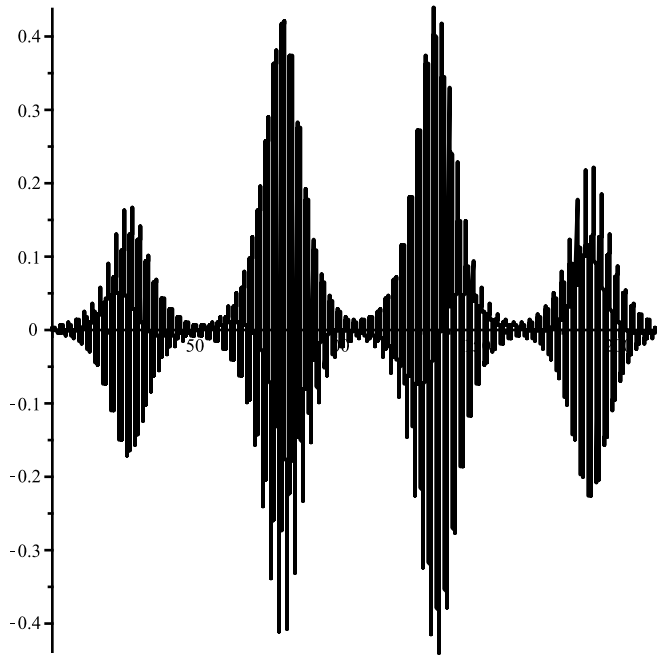
> $plots[pointplot]([seq([i*DeltaT, sols[i, 6]], i = 1 .. Nsteps)], connect = true, color = \text{"LightCoral"});$

```
# qz vs time (mass 1)
```



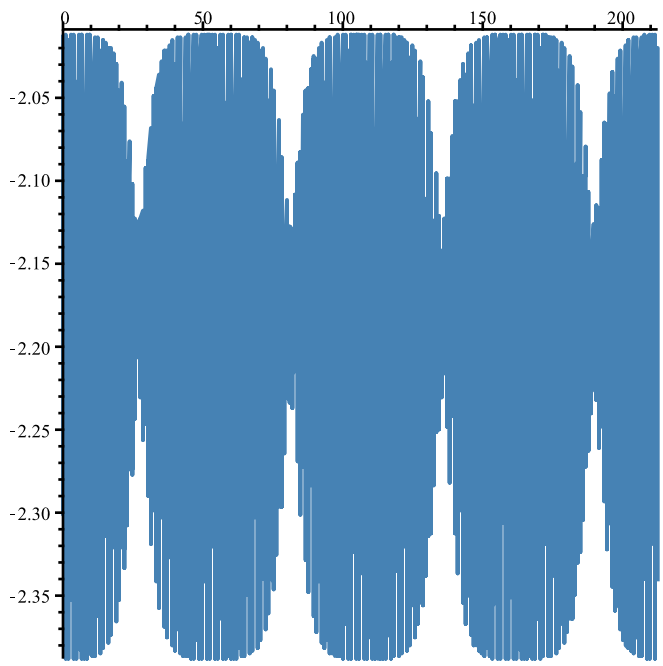
> $plots[pointplot]([seq([i*DeltaT, sols[i, 5] + sols[i, 2]], i = 1 .. Nsteps)], connect = true, color = \text{"Black"});$

```
# (q+S) y-component vs time (mass 2)
```



> $plots[pointplot]([seq([i*DeltaT, sols[i,6]+sols[i,3]], i=1..Nsteps)], connect=true, color="SteelBlue");$

```
# (q+S) z-component vs time (mass 2)
```



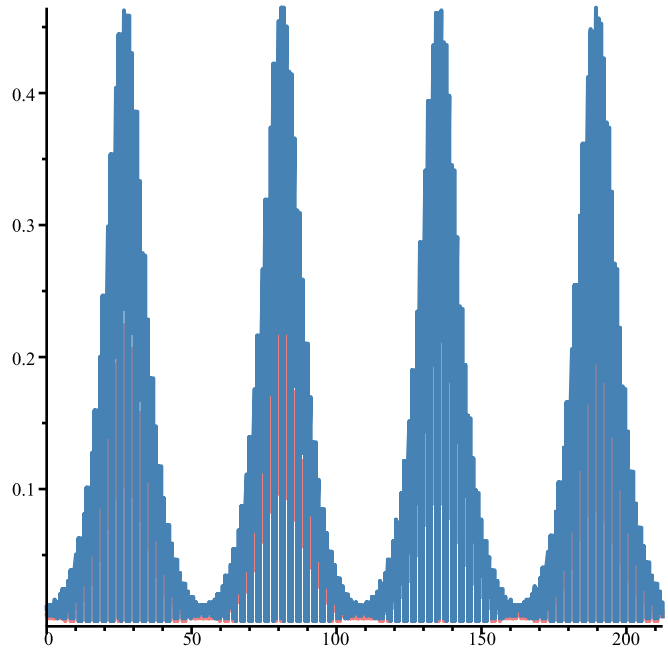
> $amred := plots[pointplot]([seq([i*DeltaT, \mathrm{sqrt}(sols[i,4]^2+sols[i,5]^2)], i=1..Nsteps)], connect=true, color="LightCoral");$

[Length of output exceeds limit of 1000000] **(5.1)**

> $amblue := plots[pointplot]([seq([i*DeltaT, \mathrm{sqrt}((sols[i,1]+sols[i,4])^2+(sols[i,2]+sols[i,5])^2)], i=1..Nsteps)], connect=true, color="SteelBlue");$

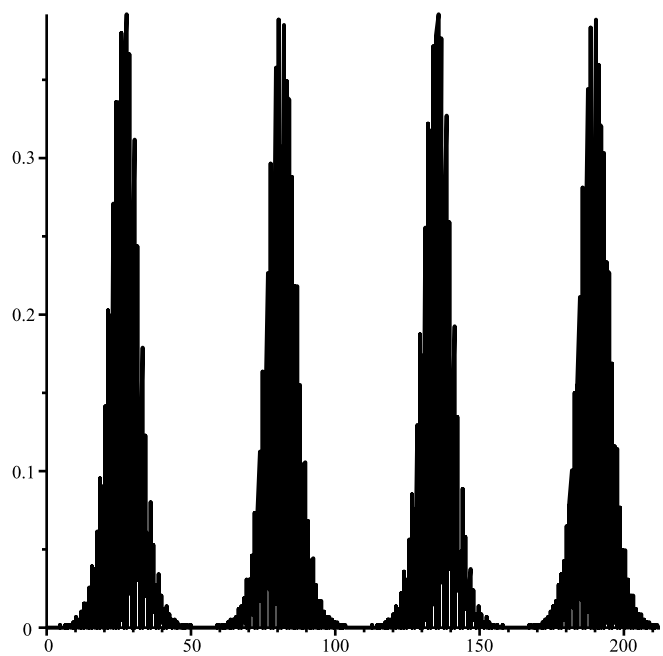
[Length of output exceeds limit of 1000000] **(5.2)**

> $plots[display](amred, amblue);$ `# swing amplitudes during the stepwise precession`



> $plots[pointplot]([seq([i*DeltaT, sols[i, 11]], i = 1 .. Nsteps)], connect = true, color = \text{"Black"})$;

 # swing kinetic energy ($\dot{S}.\dot{S}$)



> # $gxt := plots[pointplot]([seq([i*DeltaT, sols[i, 4]], i = 1 .. 4000)], connect = true, color = \text{"Black"})$;

> # $gyt := plots[pointplot]([seq([i*DeltaT, sols[i, 5]], i = 1 .. 4000)], connect = true, color = \text{"Black"})$;

>

>

>

Animation of the stepwise precession

```
> with(plottools) :

> L1 := s → [[sols[s, 1], sols[s, 2]]];
> L2 := s → [[sols[s, 1] + sols[s, 4], sols[s, 2] + sols[s, 5]]]


> my_plot := proc(n, $)
                 return display(
                      pointplot(L1(n), 'color' = "Black", connect = true),
                      pointplot(L2(n), 'color' = "Black",          connect = true),
                      disk([sols[n, 4], sols[n, 5]], 0.01, color = red),
                      disk([sols[n, 4] + sols[n, 1], sols[n, 5] + sols[n, 2]], 0.0075, color = blue))
     ;                                                                          end proc;
```

$$L1 := s \mapsto [[sols_{s,1}, sols_{s,2}]]$$

$$L2 := s \mapsto [[sols_{s,1} + sols_{s,4}, sols_{s,2} + sols_{s,5}]]$$

```
my_plot := proc(n, $)                                                                    (6.1)
    return plots:-display(plots:-pointplot(L1(n), 'color' = "Black", connect = true), plots:-
    pointplot(L2(n), 'color' = "Black", connect = true), plottools:-disk([sols[n, 4], sols[n, 5]],
    0.0100000000, color = red), plottools:-disk([sols[n, 1] + sols[n, 4], sols[n, 2] + sols[n, 5]],
    0.0075000000, color = blue))
end proc
> T := 150;

> size := 0.5;


> anim_options := 'view' = [-size ..size, -size ..size],
                    'scaling' = 'constrained','axes'='boxed',
                   'labels' = [typeset('x', '[m]'), typeset('y', '[m]')],
                   'frames' = T* 10 + 1,        # +1 for t=0 as frame 1
                    'digits' = 3;
```

> #size:=0.85+1.2+.5;

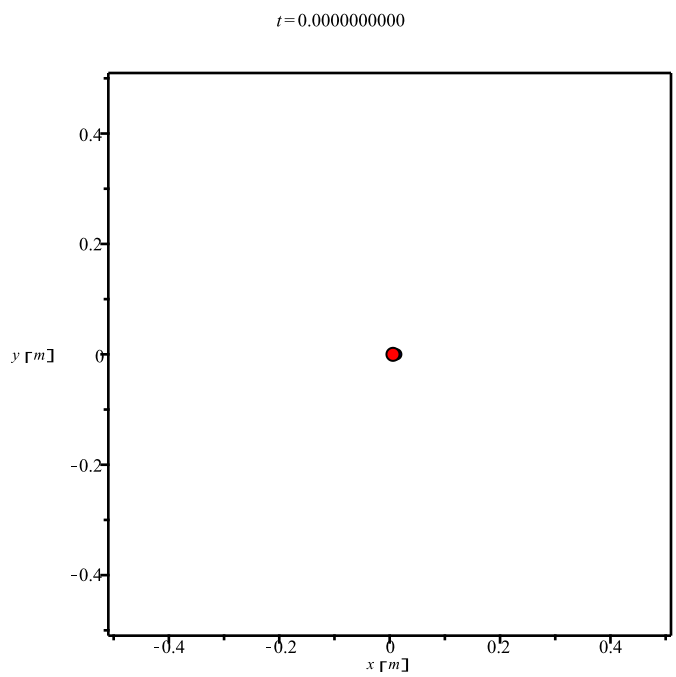
$$T := 150$$

$$size := 0.5000000000$$

$$anim_options := view = [-0.5000000000..0.5000000000, -0.5000000000..0.5000000000], scaling = constrained, axes = boxed, labels = [typeset(x, [m]), typeset(y, [m])], frames = 1501, digits = 3 \quad (6.2)$$

>

> animate(
my_plot,
[40 * round(t * 10) + 1],
t = 0 .. T,
anim_options)



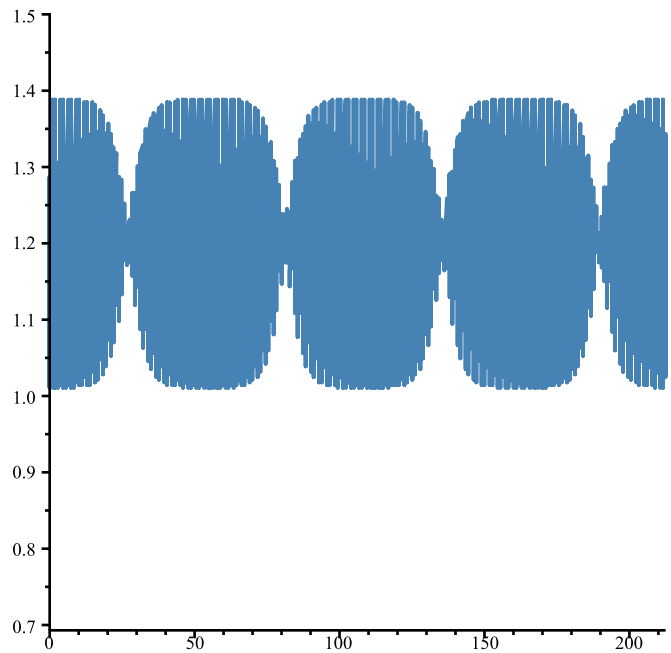
>

> Nsteps

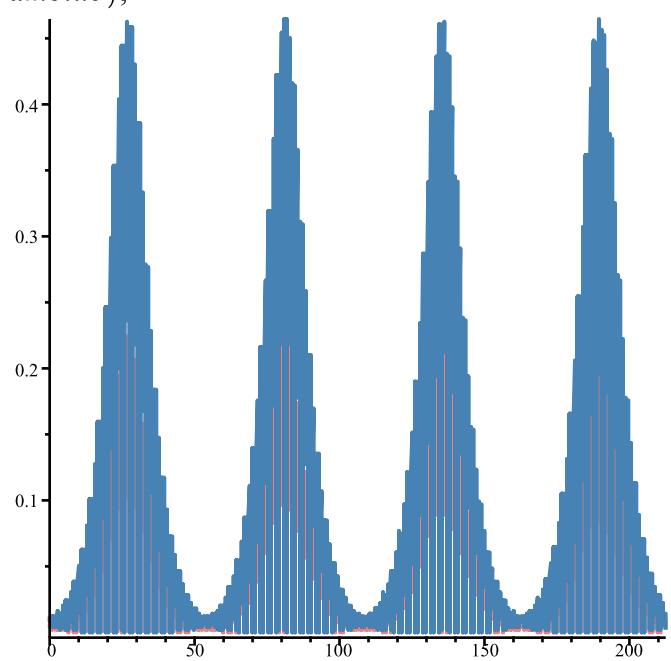
$$85000 \quad (6.3)$$

> plots[pointplot]([seq([i * DeltaT, sols[i, 10]], i = 1 .. Nsteps)], connect = true, color = "SteelBlue", view = [0 ..DeltaT·Nsteps, 0.7 ..1.5]);

```
# vibrational amplitude
```



> $plots[display](amred, amblue);$



Second example

restart and load the package

> $restart;$

> $with(plots):$

> $with(LinearAlgebra):$

> **read**("ElasticSphericalPendulum_pck_code.mpl");

$$ElasticSphericalPendulum := \textbf{module}(\ )\ \ldots\ \textbf{end module} \tag{7.1.1}$$

> $Digits := 15;$

$$Digits := 15 \tag{7.1.2}$$

The package is loaded: the commands in the package are listed.

> $with(ElasticSphericalPendulum)$

$$[ElasticSphericalDsolver, generateInitial]$$

>

>

We use a new set of parameters:

> $K := 41.0 : L0 := 0.75 :$

With these values the stretched equilibring length of the spring follows: $L = L0 + \frac{2 \cdot g}{K} = 1.231444117$.

The initial conditions are:

> $q := Vector[row](3, [0.04, 0.0, -1.08], datatype = float[8])$

> $S := Vector[row](3, [0.0370116605098804, 0., -0.999314833766771])$

> omega $:= Vector[row]([0.02565, 0.0, 0.0]);$

> $CrossProduct(\text{omega}, q);$

$$q := \begin{bmatrix} 0.0400000000000000 & 0. & -1.08000000000000 \end{bmatrix}$$

$$S := \begin{bmatrix} 0.0370116605098804 & 0. & -0.999314833766771 \end{bmatrix}$$

$$\omega := \begin{bmatrix} 0.02565 & 0. & 0. \end{bmatrix}$$

$$\begin{bmatrix} -0. & 0.0277020000000000 & 0. \end{bmatrix} \quad \textbf{(7.1)}$$

> $generateInitial(K, L0, 0.04, 0.0, -1.08, 0.00, 0.0277020000000000, 0.0, S, \text{omega})$;

$$\text{"Equilibrium point spring length"}, 1.23144411712631 \quad \textbf{(7.2)}$$

> $readdata(\text{"./dataSYMPKE.dta"}, [float\ \$\ 12])$;

$$[[0.0370116605098804, 0., -0.999314833766771, 0.04, 0., -1.08, 0.140817222063713, -0., 0.00521545266902641, 0.00180598087296713, -28.9387075595929, -28.9369015787199]] \quad \textbf{(7.3)}$$

>

> $K, L0$

$$41.0, 0.75 \quad \textbf{(7.4)}$$

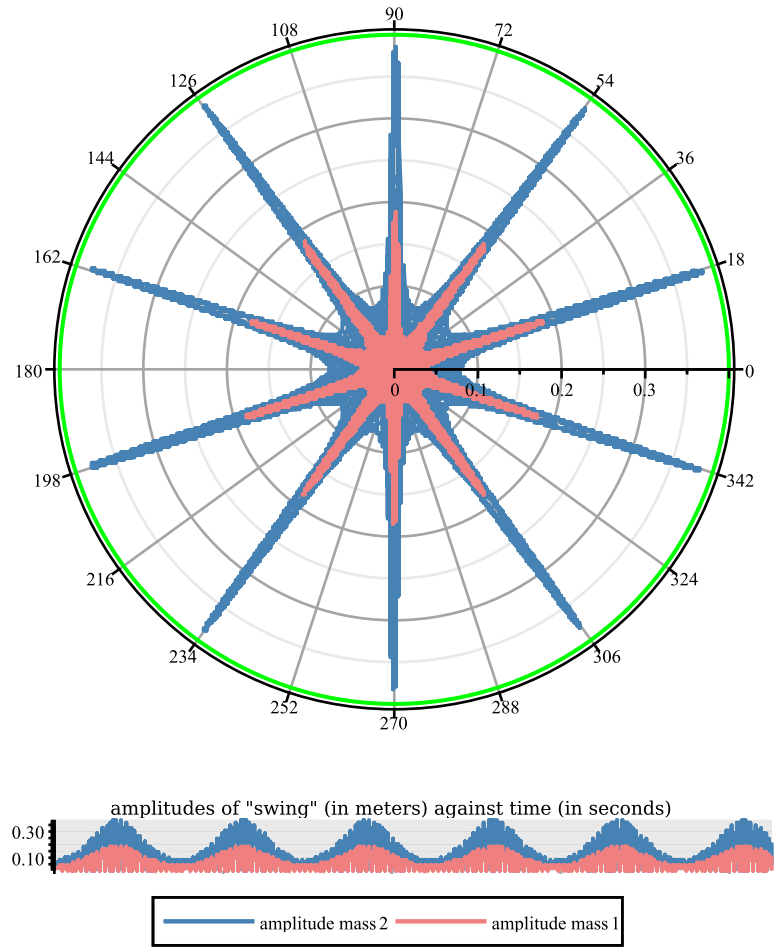
> $Nsteps := 85000 : st := time(\ ) : ElasticSphericalDsolver(Nsteps, K, L0); time(\ ) - st;$

$$288.563 \quad \textbf{(7.5)}$$

>

>

Table 2: precession angles



In this second simulation we can see a notable figure from the results of the numerical simulation: the angle of the stepwise precession is now almost equal to 36º , an integer submultiple of the full circular round, with five precession steps before restarting a new cycle of precession along the same paths.

The prediction of the angular precession step and the amplitudes of swing (~ 37 cm. for the spherical pendulum mass) is a difficult mathematical task that deserves further study.

> $sols := readdata(\text{"./PKESYM.dta"}, [float\$12])$:

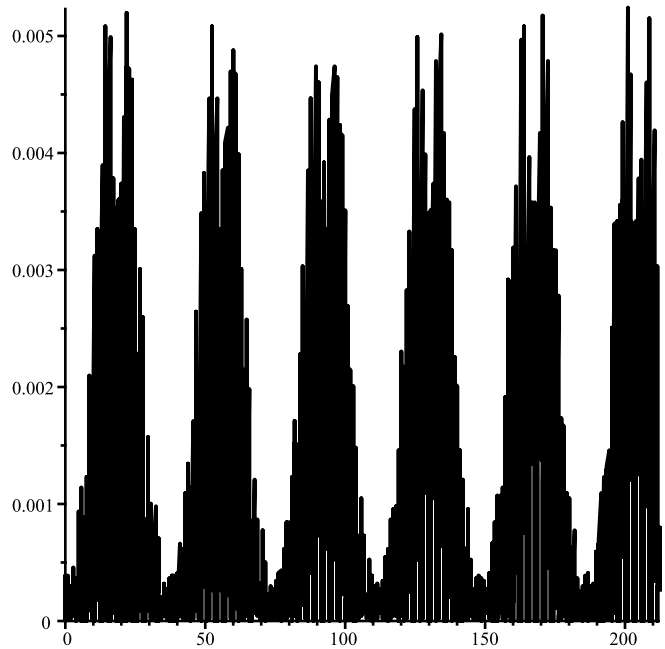
> $nops(sols)$;

$$85001 \tag{7.6}$$

> $DeltaT \cdot Nsteps$;

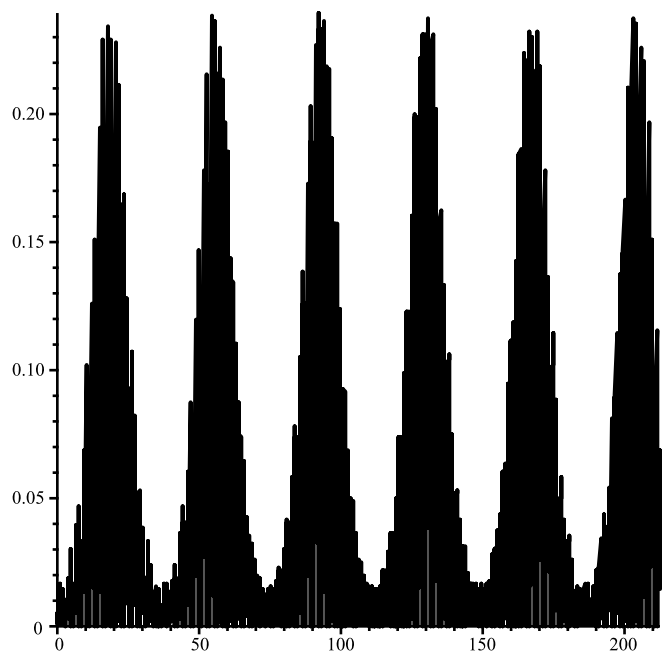
$$212.5000 \tag{7.7}$$

> $plots[pointplot]([seq([i * DeltaT, sols[i, 12]], i = 1 .. Nsteps)], connect = true, color = \text{"Black"})$;



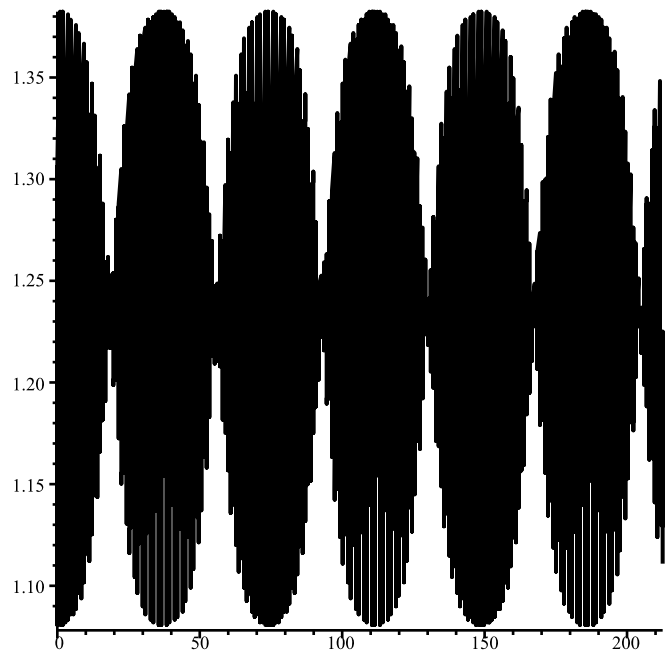
> $plots[pointplot]([seq([i*DeltaT, sols[i, 11]], i = 1 .. Nsteps)], connect = true, color = \text{"Black"});$

 # kinetic "swing" energy term, $\dot{S}\cdot\dot{S}$



> $plots[pointplot]([seq([i*DeltaT, sols[i, 10]], i = 1 .. Nsteps)], connect = true, color = \text{"Black"});$

 # stretching oscillation of the spring

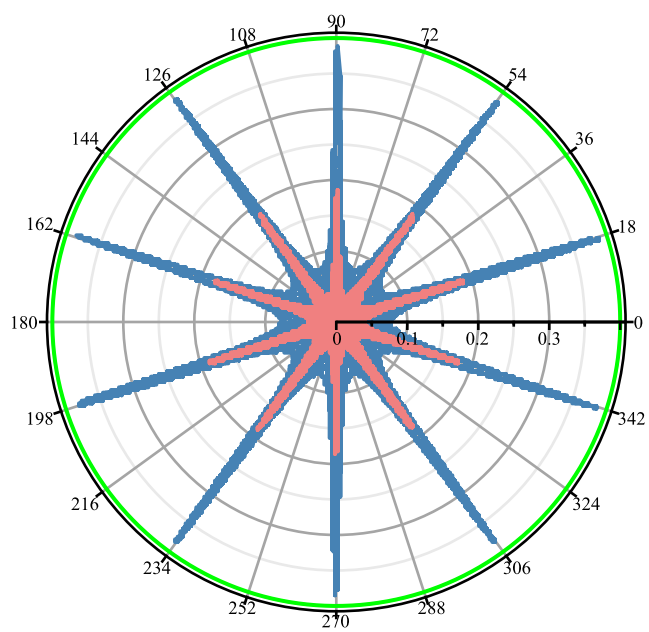


> $gr2 := plots[pointplot]([seq([sols[i,1] + sols[i,4], sols[i,2] + sols[i,5]], i = 1 .. Nsteps)], connect = true, color = \text{"SteelBlue"})$:

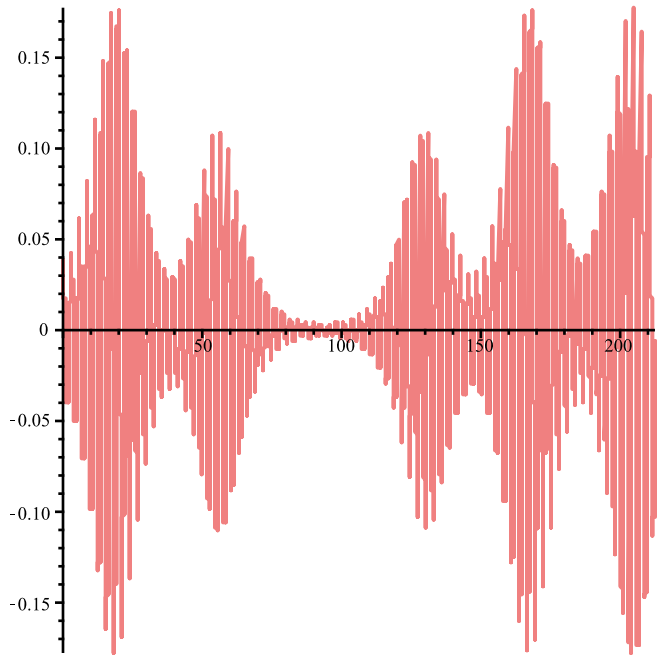
> $circ := plots[polarplot](0.4, angularunit = degrees, tickmarks = [\text{"decimalticks"}, 20], color = green)$:

> $gr1 := plots[pointplot]([seq([sols[i,4], sols[i,5]], i = 1 .. Nsteps)], connect = true, color = \text{"LightCoral"})$:

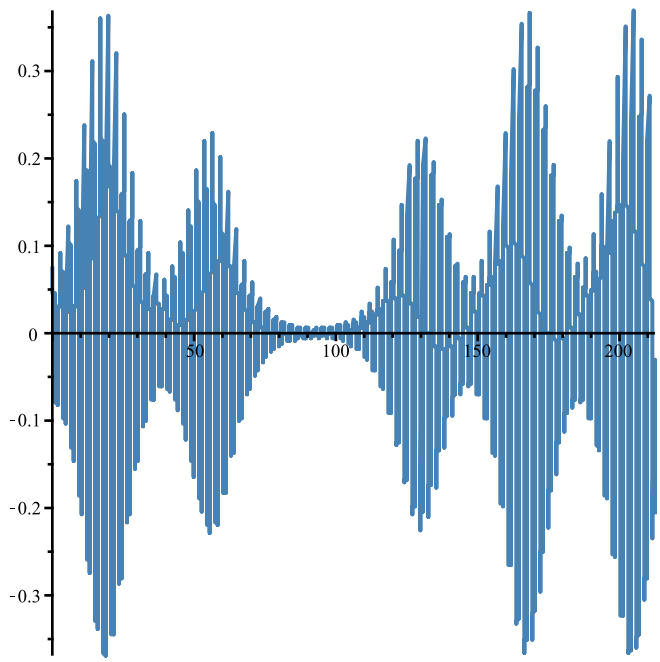
> $plots[display](gr2, gr1, circ)$;



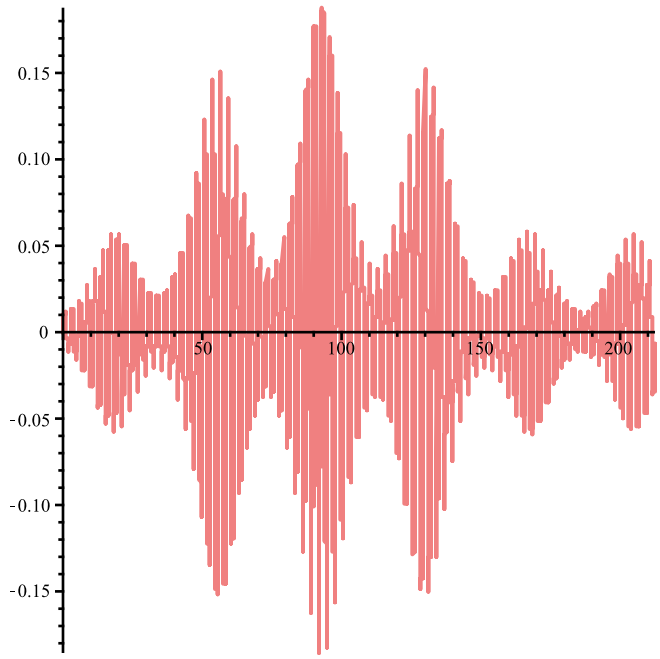
> $plots[pointplot]([seq([i * DeltaT, sols[i,4]], i = 1 .. Nsteps)], connect = true, color = \text{"LightCoral"})$



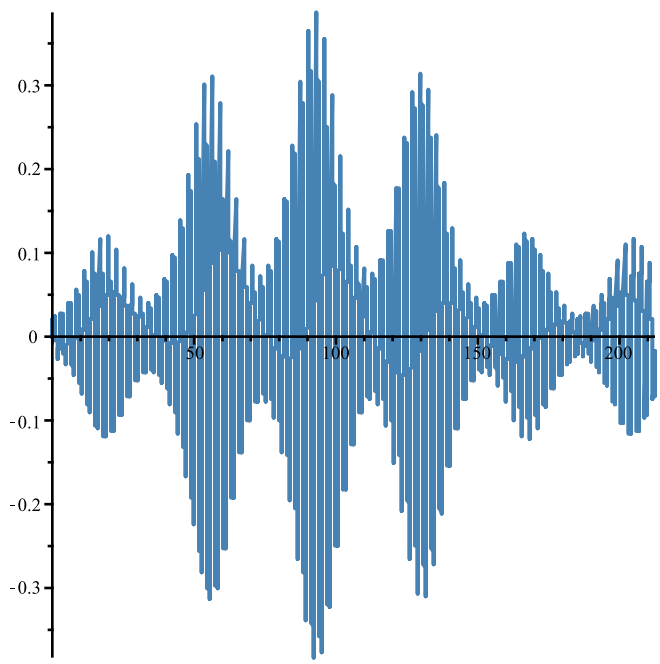
> $plots[pointplot]([seq([i*DeltaT, sols[i,1]+sols[i,4]], i=1..Nsteps)], connect=true, color="SteelBlue")$



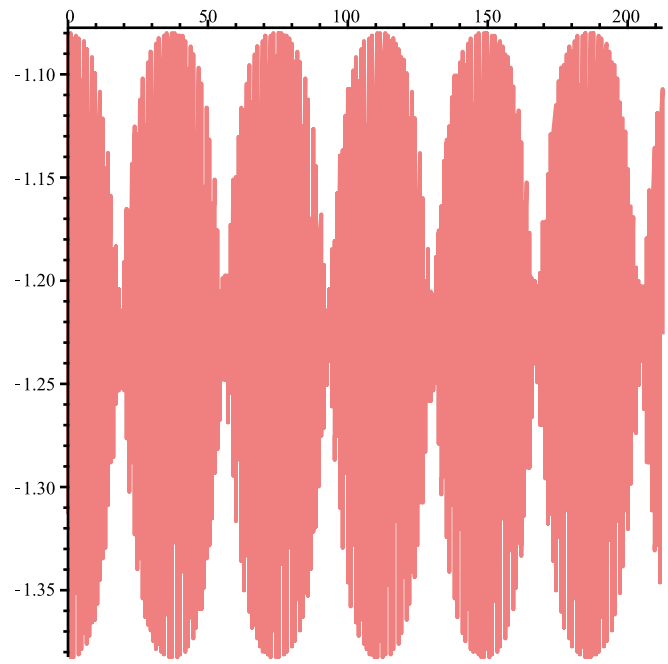
> $plots[pointplot]([seq([i*DeltaT, sols[i,5]], i=1..Nsteps)], connect=true, color="LightCoral")$



> $plots[pointplot]([seq([i*DeltaT, sols[i,2]+sols[i,5]], i=1..Nsteps)], connect=true, color="SteelBlue")$



> $plots[pointplot]([seq([i*DeltaT, sols[i,6]], i=1..Nsteps)], connect=true, color="LightCoral");$



>

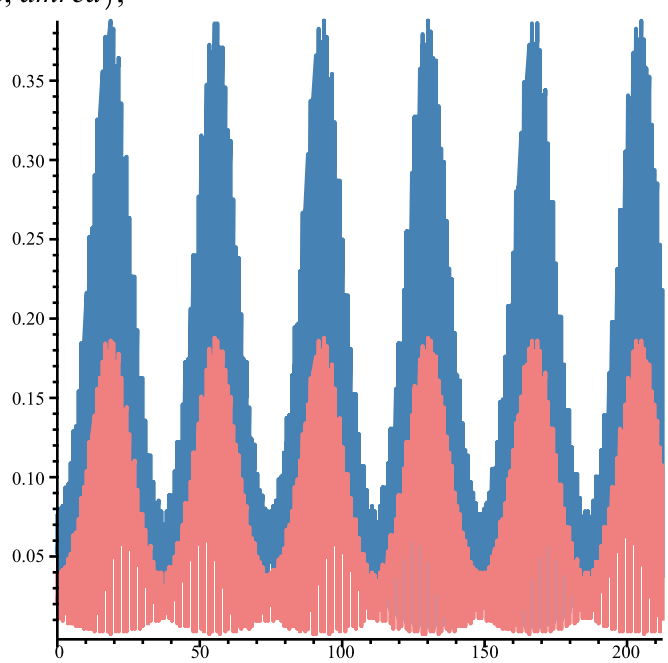
> $amred := plots[pointplot]\left(\left[seq\left(\left[i * DeltaT, \mathrm{sqrt}\left(sols[i,4]^2 + sols[i,5]^2\right)\right], i = 1 .. Nsteps\right)\right], connect = true, color = \text{"LightCoral"}\right);$

[Length of output exceeds limit of 1000000] **(7.8)**

> $amblue := plots[pointplot]\left(\left[seq\left(\left[i * DeltaT, \mathrm{sqrt}\left((sols[i,1] + sols[i,4])^2 + (sols[i,2] + sols[i,5])^2\right)\right], i = 1 .. Nsteps\right)\right], connect = true, color = \text{"SteelBlue"}\right);$

[Length of output exceeds limit of 1000000] **(7.9)**

> $plots[display](amblue, amred);$



>

>

References

1. *Resonant motions of the three-dimensional elastic pendulum*, Peter Lynch, International Journal of Non-Linear Mechanics, 37 (2002) pp. 345-367
2. *Stepwise Precession of the Resonant Swinging Spring*, Darryl D. Holm and Peter Lynch, SIAM Applied Dynamical Systems, Vol. 1, No. 1, pp. 44-64

Conclusions

The stepwise precession of a double elastic-spherical pendulum is numerically simulated and exhibits the same physical effect as the already studied on the single elastic pendulum. A key condition is the initial alignment of the two elements, the spring pendulum and the spherical rigid one, along the same direction, slightly deviated from the vertical equilibrium symmetry axis. Although this precession might remind the Foucault pendulum, the origin of this (completely different) effect is hidden in the nonlinear character of the dynamics and the internal symmetries of this Hamiltonian system.

Legal Notice: © 2021. Maplesoft and Maple are trademarks of Waterloo Maple Inc. Neither Maplesoft nor the authors are responsible for any errors contained within and are not liable for any damages resulting from the use of this material. This application is intended for non-commercial, non-profit use only. Contact the authors for permission if you wish to use this application in for-profit activities.