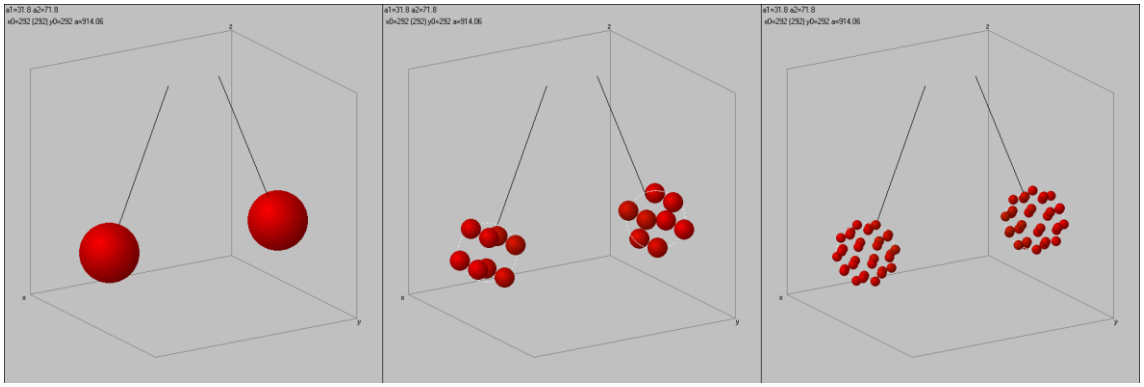
Canton

By Antonio Carlos M. de Queiroz, Federal University of Rio de Janeiro, Brazil

Version 2.3b – 26/10/2018

The program Canton has as main function the simulation of a set of "pith balls" suspended from strings, with fixed potentials on them, computing how the balls stabilize. The name refers to John Canton, British scientist that studied electricity in the 1700's. Pair of "pith balls" suspended from strings are often cited in old literature as "Canton balls".

The balls can be are approximated as point charges, or as composite groups of point charges placed at the corners of tetrahedrons, octahedrons, cubes, icosahedrons, dodecahedrons, or triacontahedrons. This allows considering the distribution of electrical charges in the surfaces of the balls, with the corresponding effects in capacitances and forces taken into account.



Balls represented by one, eight, or thirty-two charges.

Also available are lines, squares, toroids, a ball with a suspension line made with a string of balls, and programmable flat plates.

Each ball is specified by the coordinates of the suspension point, x_0, y_0, z_0, the length of the suspension string, L, horizontal and vertical rotation angles θ_x, θ_z, radius R and mass m. The description of the configuration is given in a text, one command per line:

```
Bname Conductor x0 y0 z0 L Angx Angz Radius Mass
Oname Conductor x0 y0 z0 L Angx Angz Roctahedron Mass [Rball]
Ename Conductor x0 y0 z0 L Angx Angz Rtetrahedron Mass [Rball]
Cname Conductor x0 y0 z0 L Angx Angz Rcube Mass [Rball]
Iname Conductor x0 y0 z0 L Angx Angz Ricosahedron Mass [Rball]
Dname Conductor x0 y0 z0 L Angx Angz Rdodecahedron Mass [Rball]
Rname Conductor x0 y0 z0 L Angx Angz Rtriacontahedron Mass [Rball]
Lname Conductor x0 y0 z0 L Angx Angz Rline Mass [Rball]
Tname Conductor x0 y0 z0 L Angx Angz Rtoroid Mass [Rball]
Sname Conductor x0 y0 z0 L Angx Angz Rsquare5x5 Mass [Rball]
Qname Conductor x0 y0 z0 L Angx Angz Rsquare10x10 Mass [Rball]
Fname Conductor x0 y0 z0 L Angx Angz Rplaque5x5 Mass [Rball]
Gname Conductor x0 y0 z0 L Angx Angz Rplaque10x10 Mass [Rball]
Xname Conductor x0 y0 z0 L Angx Angz Rshape5x5 Mass [Rball]
Yname Conductor x0 y0 z0 L Angx Angz Rshape10x10 Mass [Rball]
Zname Conductor x0 y0 z0 L Angx Angz Rshape15x15 Mass [Rball]
Hname Conductor x0 y0 z0 L Angx Angz Rball Mass Rline_ball Mass_line
* Comment
.V Conductor Voltage
.X  5 comments below determine the shape of element x *x...x
.Y 10 comments below determine the shape of element y *x........x
.Z 15 comments below determine the shape of element z *x............x
```

Units in cm, degrees, grams and volts. For objects composed of several balls, the radius of the balls can be specified too. If not given an optimized default value is used, which produces the right capacitance for an insulated ball, $4\pi\varepsilon_0 r$, where r is the radius of the balls and $\varepsilon_0 = 8.854187817\times10^{-12}$ is the permittivity of vacuum. These values are (see the appendix):

```
Tetrahedron: 0.46234592535r
Octahedron: 0.3743113315r
Cube: 0.3262848751r
Icosahedron: 0.2627574554r
Dodecahedron: 0.2074629423r
Triacontahedron: 0.1604139269r
```

For the triacontahedron, small identical balls don't result with identical charges for an insulated main ball, but with two different values depending if the balls belong to the set forming an icosahedron or a dodecahedron. The last parameter, if present, makes all the balls identical. If a negative value is given, it is replaced by the value above. The default is different balls, that will have identical charges in an insulated ball. For these the values of the radius are 0.1598652556*r* (dodecahedron balls) or 0.1613508447*r* (icosahedron balls).

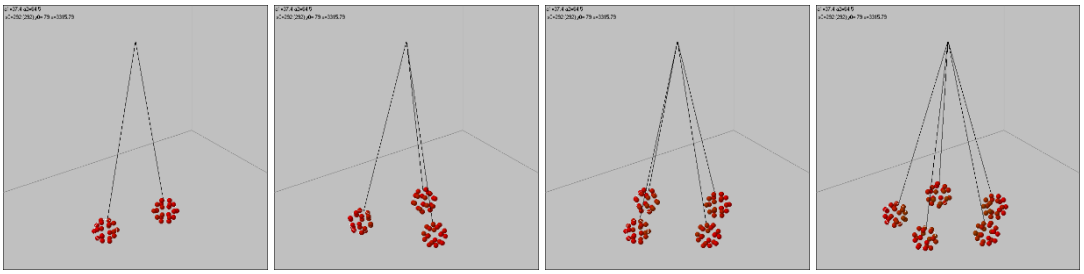
Starting from the initial configuration, the program performs a simplified time-domain simulation, stopping after a number of steps or when the forces moving the balls fall below a given limit. Each ball belongs to a conductor, which can have a specified potential. If a voltage is not assigned the conductor is at 0 V. Fixed balls can be specified by setting the line length to zero. The strings are rigid and rotate horizontally and vertically in their supports, with balls following the movement of the strings.

The program allows a three-dimensions visualization of the configuration, which can be adjusted using the mouse. The balls are colored according with their charge density. The right mouse button opens a menu with several options for visualization control, capacitance calculations and stability analysis.

Example:

Two to five balls with a common suspension point, with mass 0.4 g, radius 0.75 cm, string length 13.5 cm, at 20 kV, modeled as dodecahedrons. The input file for four balls is:

```
* Four spheres
dall1 all 0 0 0 13.5    0  10 0.75 0.4
dall2 all 0 0 0 13.5  180  10 0.75 0.4
dall2 all 0 0 0 13.5   90  10 0.75 0.4
dall2 all 0 0 0 13.5  -90  10 0.75 0.4
.V all 20000
```



Two to five balls at 20 kV.

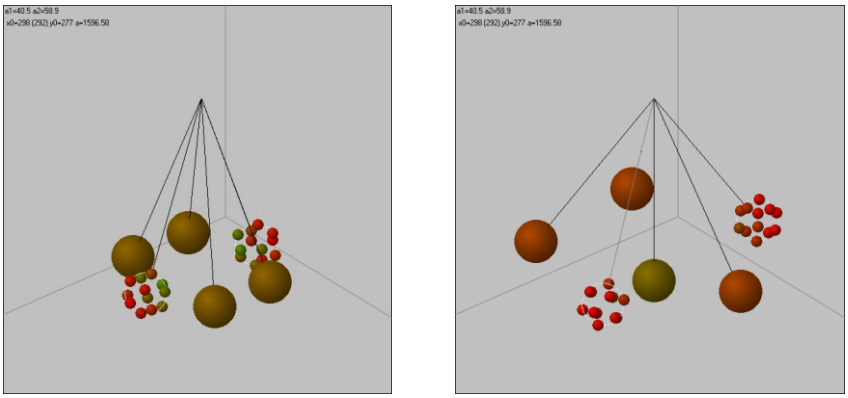
Balls	Angle	Separation cm	Capacitance pF
Two	10.7151	3.5200	1.4527
Three	13.1605	3.8238	1.9549
Four	14.5022	3.2809	2.3465
Five	15.2523	2.6750	2.6447

The angles with the vertical increase with more balls, but the separation of the balls is maximum for three balls. More balls don't produce stable solutions as regular polygons.

Example:

The file below describes six balls suspended from a common point, two of them modeled by icosahedrons.

```
* Six spheres
iall1 all 0 0 10 20    0 20 2 0.4
ball2 all 0 0 10 20   60 20 2 0.4
ball3 all 0 0 10 20  120 20 2 0.4
iall4 all 0 0 10 20  180 20 2 0.4
ball5 all 0 0 10 20  -60 20 2 0.4
ball6 all 0 0 10 20 -120 20 2 0.4
.V all 30000
```



6 spheres, with two models. Initial configuration and solution.

The solution found by the program is a pentagon with one ball at the center. Initially the solution appears to be hexagonal, with the two icosahedrons at slightly higher angle, but it soon becomes unstable, changes to another unstable configuration, and finally change to the final configuration. If identical balls were used, the solution would stay hexagonal, but any perturbation would result in the other solution.

Squares:

Square arrays of balls are also available, with 5×5, 10×10, and 15×15 balls, fixed or programmable, where the balls can be set as present or not. For the squares, the default radius of the balls results in the capacitance of a square sheet, assumed as 0.4081084689 pF for the 1×1 cm sheet [4]:

```
5×5: 0.1376686573r
10×10: 0.06642043070r
15×15: 0.04347756960r
```

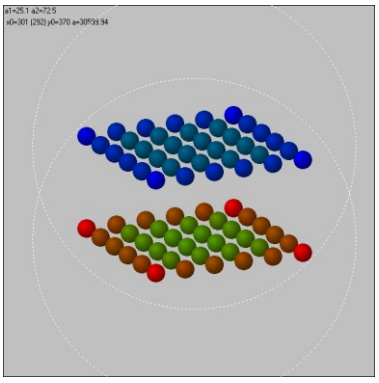
If the option of use squares is set, the balls have the capacitance of a square sheet with uniform charge density [1], resulting is somewhat smaller capacitance (~0.402 pF for the 1×1 cm sheet), and if two squares are aligned vertically the elastance between uniformly charged square plates is used [1]. The variations of the squares, elements s, q, f, g, x, y, and z, use the same values. This allows the program to calculate with reasonable precision the capacitances between flat plates aligned vertically. The squares can also be used suspended, as the regular "pith balls".

The capacitances between the conductors can be calculated at the end of the analysis, allowing the use of the program as a general capacitance calculator. For this are especially useful the flat squares, flat squares with corner links (f, g), and the programmable flat squares (x, y, z), which can be used to build any structure.

Example:

In [1] an example is given for the capacitance between two square plates with 1 cm sides modelled by 6×6 elements. The example can be reproduced in the program with the input file below. A programmable flat plate "y" is used. As it has 10×10 balls, just 6 are used. To produce 1 cm plates, the radius is set to 0.5×10/6 = 0.833 cm.

```
* Parallel plates 1x1 cm, 6x6 (Reitan 1959)
.Y
*..........
*..........
*..xxxxxx..
*..xxxxxx..
*..xxxxxx..
*..xxxxxx..
*..xxxxxx..
*..xxxxxx..
*..........
*..........
yx a 0 0 0 0 0 0 0.8333333333 0.4
yy b 0 0 0.005 0 0 0 0.8333333333 0.4
.V a 30000
```



The separation of the plates in the description above is of 0.005 cm, and 0.5 cm in the figure. The table below compares the result of the program (CC) with the ones in the paper (CR). There is some difference, possibly due to low precision in the original calculations.

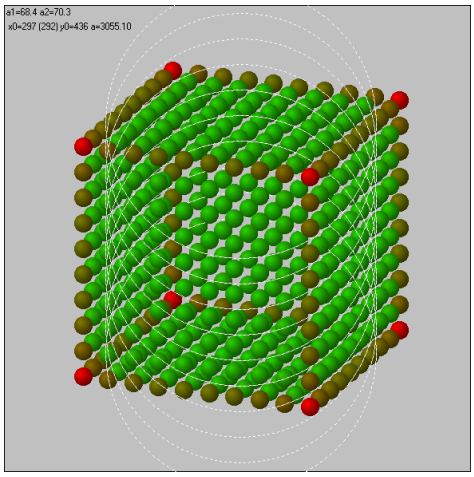
Separation cm	CR pF	CC pF
0.005	17.74	17.96
0.025	3.7892	3.8008
0.05	2.0295	2.0323
0.10	1.1324	1.1338
0.20	0.6629	0.6638
0.50	0.3750	0.3755
1.00	0.2801	0.2805

Example:

A cube with 10 cm of side should have a capacitance of 7.351036 pF [2], [3].

A cube can be made with the programmable squares:

```
* Cube 15x15x15, elements at the limits: 7.351036 pF
.Y
*xxxxxxxxxx
*xxxxxxxxxx
*xxxxxxxxxx
*xxxxxxxxxx
*xxxxxxxxxx
*xxxxxxxxxx
*xxxxxxxxxx
*xxxxxxxxxx
*xxxxxxxxxx
*xxxxxxxxxx
yx all  0 0 0  0  0 0 5.5555555555 0.4
yx all  0 0 10 0  0 0 5.5555555555 0.4
.Y
*xxxxxxxxxx
*x........x
*x........x
*x........x
*x........x
*x........x
*x........x
*x........x
*x........x
*xxxxxxxxxx
yx all  0 0 1.1111111111   0  0 0 5.5555555555 0.4
yx all  0 0 2.2222222222   0  0 0 5.5555555555 0.4
yx all  0 0 3.3333333333   0  0 0 5.5555555555 0.4
yx all  0 0 4.4444444444   0  0 0 5.5555555555 0.4
yx all  0 0 5.5555555555   0  0 0 5.5555555555 0.4
yx all  0 0 6.6666666666   0  0 0 5.5555555555 0.4
yx all  0 0 7.7777777777   0  0 0 5.5555555555 0.4
yx all  0 0 8.8888888888   0  0 0 5.5555555555 0.4
.V all 30000
```



With the option to use squares not used, the capacitance results as 7.5750 pF, with 3% of error. There is no sense to use squares in this problem, what would correspond to stack flat sheets with the programmed patterns (results in 7.5183 pF).

By default, the program does not calculate immediately the charge distribution in large assemblies, unless the corresponding option is set. This allows fast edition and visualization of the system, since the calculation may be slow.

The program always calculates the complete capacitance matrix of the system. For just two conductors it calculates also the capacitances when both are at opposite voltages, what places the two capacitances to the infinity of the objects in series, in parallel with the capacitance between the objects.

For the parallel capacitor example, for 2 mm separation the program lists:

```
Capacitance matrix:
C[a,a]=0.779825299012327 pF
C[a,b]=-0.54784674933736 pF
C[b,b]=0.779825299012327 pF
Capacitances:
C(a,b)=0.54784674933736 pF
C(a,infinity)=0.231978549674967 pF
C(b,infinity)=0.231978549674967 pF
```

```
Capacitance for balanced voltages=0.663836024174843 pF
```

Simulation method:

The positions of the objects are given by horizontal and vertical angles, θ_x and θ_z. Each object has a mass m, a radius r, and a string length L, fixed at the point x_0, y_0, z_0.

The cycle below is repeated to find a fixed-point solution:

1) The positions of the objects are calculated:

$$x = x_0 + L \sin\theta_x \cos\theta_x$$
$$y = y_0 + L \sin\theta_z \sin\theta_x$$
$$z = z_0 - L \cos\theta_z$$

2) The elastance matrix is calculated from the geometry. **V**=[**S**]**Q**, where **V** is the voltage vector and **Q** the charge vector. The elements at the main diagonal of [**S**] are the inverses of the capacitances of the balls, $1/(4\pi\varepsilon_0 r)$, and the others the negatives of the induction coefficients between different spheres, $-1/(4\pi\varepsilon_0 d_{ij})$, where d_{ij} is the distance between the centers of the two spheres of the pair.

3) The capacitance matrix is calculated as the inverse of the elastance matrix. [**C**]=[**S**]$^{-1}$.

4) The charges are calculated by **Q**=[**C**]**V**.

5) The forces on the objects are calculated, electrical forces by adding $Q_iQ_j/(4\pi\varepsilon_0 d_{ij}^2)$ for all pairs of spheres, and the gravitational force by mg, where g = 9.81 is the gravitational constant. They are decomposed in three orthogonal directions as f_x, f_y, and f_z.

6) The rotational forces are calculated (case for simple balls):

$$f_{rx} = -f_x \sin\theta_x + f_y \cos\theta_x$$
$$f_{rz} = (f_x \cos\theta_x + f_y \sin\theta_x)\cos\theta_z + f_z \sin\theta_z$$

7) The angles are updated, assuming acceleration starting from rest and small movement.

$$\theta_x(t_0 + \Delta t) = \theta_x(t_0) + \frac{\Delta t^2}{2mL} f_{rx} \operatorname{cosec}\theta_z$$

$$\theta_z(t_0 + \Delta t) = \theta_z(t_0) + \frac{\Delta t^2}{2mL} f_{rz}$$

Singularities are avoided by limiting m and cosec θ_z. L=0 means no movement.

The simulation may produce apparently strange movements when the lines are vertical and becomes unstable with large Δt, consequences of the approximation in the angle update equations.

In the case of objects composed of several balls, to calculate the rotation forces one method is to calculate the forces f_x, f_y, and f_z as above for all the balls and then calculate f_{rx} and f_{ry} considering their sums. This corresponds to apply all the forces at the center of the object and, although quite precise for spheres, is not exact and may produce artifacts as rotation for asymmetrical objects (this is used if the option "correct rotation forces" is not set). An exact method requires the calculation of equivalent rotational forces at the centers of the objects, that are then added. For each ball it is implemented as the following calculations:

a) Distance in the horizontal plane between the ball at x_1, y_1, z_1 and the suspension point at x_0, y_0, z_0: $d_{01xy} = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$.
b) Angle between the x axis and the horizontal projection of suspension line of the ball: $\theta_{xe} = \arctan(y_1 - y_0, x_1 - x_0)$.
c) Horizontal rotation force in the ball: $f_{rxe} = -f_x \sin\theta_{xe} + f_y \cos\theta_{xe}$.
d) Projection of the distance d_{01xy} on the direction from the suspension center to the center of the object: $d_{01z} = d_{01xy} \cos(\theta_{xe} - \theta_x)$.
e) Angle between the vertical and the projection: $\theta_{ze} = \arctan(d_{01z}, z_0 - z_1)$.
f) Vertical rotation force in the ball: $f_{rze} = (f_x \cos\theta_x + f_y \sin\theta_x)\cos\theta_{ze} + f_z \sin\theta_{ze}$.
g) Equivalent horizontal rotation force at the center: $f_{rx} = f_{rxe} d_{01xy}/L \operatorname{cosec}\theta_z$.
h) Equivalent vertical rotation force at the center: $f_{rz} = f_{rze} d_{01z}/L \operatorname{cosec}\theta_{ze}$.

The functions cosec is limited in ±1000 to minimize problems with vertical lines or balls directly under the suspension point.

Stability:

The update equations are in the form $\boldsymbol{\theta}\ (t_0+\Delta t) = \boldsymbol{\theta}\ (t_0)+\Delta t\ \mathbf{F}(\boldsymbol{\theta})$, forward Euler approximation for the solution of $d\boldsymbol{\theta}/dt = \mathbf{F}(\boldsymbol{\theta})$, system of nonlinear state equations.

The eigenvalues of the Jacobian matrix of $\mathbf{F}(\boldsymbol{\theta})$ can be used to verify the stability of a solution. The program can generate the Jacobian matrix after a precise convergence to a solution. It can be copied to a mathematical manipulations program external to the simulator that can compute its eigenvalues.

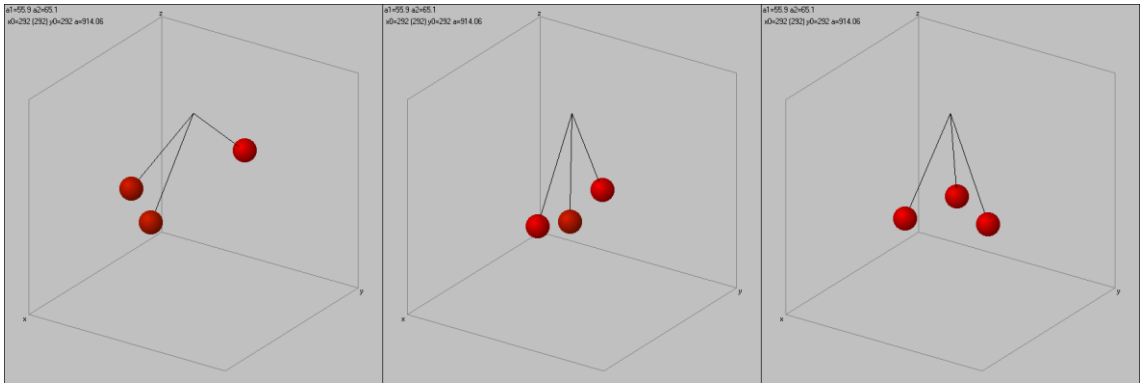
All eigenvalues with negative real part mean that the solution is stable. One or more with positive real part mean that the solution is unstable.

There are two eigenvalues for each sphere or object. The functions derived, numerically, are arranged in the Jacobian matrix in the sequence, for each object *i*:

$$f_{2i-1} = \frac{\Delta t}{2mL} f_{rx} \operatorname{cosec}\ \theta_z$$
$$f_{2i} = \frac{\Delta t}{2mL} f_{rz}$$

Note that Δt scales the Jacobian matrix and should scale the eigenvalues too. Changing its value is useful to evaluate the precision of the calculation.

Example:



Three balls, starting in the same plane. Initial configuration, unstable solution, and stable solution.

A configuration with three or more balls suspended from a common point, initialized with all the balls in the same plane will remain in the same plane during the calculation, but the fixed-point solution found is unstable. For three balls, the description of the system, three balls with 2-cm radius suspended by 20-cm strings, with two weighting 0.5 grams, and one 0.6 grams, initially at different angles in the same plane, at 30 kV, is:

```
ball1 all 0 0 15 20 0 -50 2 0.6
ball2 all 0 0 15 20 0  40 2 0.5
ball3 all 0 0 15 20 0  70 2 0.5
.V all 30000
```

The solution found puts the balls at -27.1°, 1.57°, and 31.1°. The Jacobian matrix for this solution, with Δt = 0.1, is found as:

```
-0.423777  -0.000000   0.110778  -0.000000   0.312999  -0.000000
-0.000005  -5.497393  -0.000004   3.207857  -0.000002   0.105499
10.080389   0.000000   0.336360   0.000000 -10.416749   0.000000
-0.000043   3.851241   0.000017  -9.841161   0.000060   3.538555
 0.289090   0.000000  -0.105730   0.000000  -0.183360   0.000000
 0.000002   0.126599  -0.000005   3.540159  -0.000002  -5.760000
```

The eigenvalues of this matrix are:

```
-13.11, -5.746, -2.238, -1.323, -0.2212, 1.274
```

The positive eigenvalue indicates instability. A perturbation in the initial angles leads to another solution, with the three spheres approximately in a triangle (inexact due to the different weights). The program lists the solution as:

```
ball1 all 0 0 15 20 -34.0999384973409 -20.6657246906453 2 0.6
ball2 all 0 0 15 20 -93.3796876856218 24.4882166609624 2 0.5
ball3 all 0 0 15 20 25.1798106966359 24.4882166626852 2 0.5
```

The Jacobian matrix is:

```
-2.340475   0.000000   1.170174   1.093984   1.170301  -1.093984
-0.000003  -3.649371   0.065899   1.082152  -0.065898   1.082152
 1.018023   0.460048  -2.031735  -0.278453   1.013706   0.673061
 0.163506   1.299006  -0.047858  -3.986856  -0.115620  -1.037468
 1.017913  -0.460048   1.013822  -0.673061  -2.031729   0.278453
-0.163489   1.299006   0.115627  -1.037468   0.047890  -3.986856
```

And the eigenvalues are now all negative, indicating stability:

```
-3.923, 0, -6.125, -3.179, -2.409, -2.391
```

If the three spheres are made identical, the first solution puts a sphere at the vertical and the other two symmetrically placed at both sides of it. There is no positive eigenvalue, but there are two at 0. Any small asymmetry splits them in one positive and other negative. The case of multiple eigenvalues at 0 may also indicates instability, but not always. As other examples, four identical balls suspended from the same point form a stable square up to a certain voltage, when the square solution becomes unstable. Five balls can form a stable pentagon, also up to a certain voltage, or a square with a sphere at the center, also stable up to a certain voltage. The solutions as regular polygons with 6 or more identical balls suspended from a common point are always unstable.

The calculation of eigenvalues may be numerically problematic. The solution must be precise, or the analysis may produce wrong conclusions. The force error must be set to a low value for a precise solution. The program allows change of the angle variation used in the numerical computation of the derivatives in the Jacobian matrix. Too high or too low values may generate erroneous matrices.

For composite objects, as the equivalent rotational forces are used to calculate the movements, also just two variables describe each object.

Simulation with speed considered:

The program can also simulate the system considering the speed of the balls, in a true time-domain simulation. The final fixed-point solution is the same, but there are more problems with instability, due to the forward Euler method used, requiring a smaller Δt, usually 10 times smaller, and consequent larger number of time steps for the solution. The state of each sphere of composite object is described by two angles and two angular speeds. A Jacobian matrix, not implemented, would require four lines and columns for each object. In this case the update equations, including a damping coefficient α, are:

$$\dot{\theta}_x(t_0+\Delta t) = \dot{\theta}_x(t_0) + \frac{\Delta t}{m}\left(\frac{f_{rx}}{L}\operatorname{cosec}\theta_z(t_0) - \alpha\dot{\theta}_x(t_0)\right)$$
$$\dot{\theta}_z(t_0+\Delta t) = \dot{\theta}_z(t_0) + \frac{\Delta t}{m}\left(\frac{f_{rz}}{L} - \alpha\dot{\theta}_z(t_0)\right)$$
$$\theta_x(t_0+\Delta t) = \theta_x(t_0) + \Delta t\dot{\theta}_x(t_0)$$
$$\theta_z(t_0+\Delta t) = \theta_z(t_0) + \Delta t\dot{\theta}_z(t_0)$$

Appendix:

The values of the radius of the small balls composing a ball can be found experimenting with the program or by analysis, by computing the capacitance matrix for the configuration of point charges and comparing the total capacitance with the capacitance of an insulated sphere of radius *r*. Some exact values are listed below for the simplest figures:

$$\text{Tetrahedron:}\quad \frac{6\sqrt{6}+32}{101}r = 0.46234592535r$$
$$\text{Octahedron:}\quad \frac{8\sqrt{2}+22}{89}r = 0.3743113315r$$

$$\text{Cube:}\quad \frac{106\sqrt{6}+188\sqrt{3}+120\sqrt{2}+820}{4827}r = 0.3262848751r$$

The radius of a small sphere in an $n\times n$ square with side $2r$ representing a square sheet, assuming uniform charge distribution in the square, used when the options of use squares is set [2] is:

$$\frac{2r}{4n\,\mathrm{Ln}(\sqrt{2}+1)} = \frac{1}{3.525494348}\frac{2r}{n} = 0.2836481642\frac{2r}{n}$$

The induction coefficient, or elastance, between two aligned squares with side a separated by a distance d is ([1] just gives a plot):

$$k = \frac{4/a^2}{4\pi\varepsilon_0}\int_0^{\frac{a}{2}}\int_0^{\frac{a}{2}}\frac{dxdy}{\sqrt{x^2+y^2+d^2}} =$$

$$= \frac{1/a^2}{4\pi\varepsilon_0}\left\{2a\mathrm{Ln}\left(\frac{(\sqrt{2}\sqrt{2d^2+a^2}+a)^2}{4d^2+a^2}\right) - 4d\,\tan^{-1}\left(\frac{\sqrt{2}a^2}{4d\sqrt{2d^2+a^2}}\right)\right\}$$

References:

[1] D. K. Reitan, "Accurate determination of the capacitance of rectangular parallel-plate capacitors," J. Appl. Phys., 30, 172, 1959.
[2] D. K. Reitan and T. J. Higgins, "Calculation of the electrical capacitance of a cube," J. Appl. Phys., 22, 223, 1951.
[3] Chi-Ok Hwang and Michael Mascagni, "Electrical capacitance of the unit cube," J. Appl. Phys. 95, 3798 2004.
[4] F.H. Read, "Capacitances and singularities of the unit triangle, square, tetrahedron and cube," COMPEL, 23, 2, pp. 572-578, 2004.