Extended Essay in Mathematics

Research Question:

How should you locate points on a sphere so that the minimum distance between any two of them is as large as possible?

Name: Sung Jun **KIM**

Candidate Number: 003258 -067

School: South Island School, Hong Kong

Word Count: 3685

Submission Date: May 2011

Sung Jun Kim
003258 -067
South Island School

The Abstract

This extended essay deals with the exploration of maximum separations of points in curved paths. This investigation can help to find one possible way of such points on a sphere. The investigation solves the problem on a ring and then extends the concepts to a sphere and discusses potential challenges and solutions to the numerical approach, whilst progressively increasing the number of points, to ultimately answer the research question: 'How should you locate points on a sphere so that the minimum distance between any two of them is as large as possible?'

The scope of this extended essay is concerned primarily with a strategy to find different distributions of points and restricted to the method in which was undertaken. Through various attempts in providing a proper method towards this research, MATLAB, a numerical computing environment, was the best option. Although it was a new experience to use MATLAB and start programming skills, a numeric approach would be the only pragmatic way to solve this problem, as an algebraic method would be far beyond the scope of IB Mathematics. The investigation contains many underlying physics principles such as charge forces and Newton's law in order to help with the investigation. Through MATLAB a system can be created that contain interacting forces on charged particles, where their movements can be observed.

The investigation undertaken leads to the conclusion that it is not easy to come up with a strategy where you can have a truly equally distanced distribution. The strategy performed allows us to determine the points enabling uniform distribution. Although the positions with uniform distribution were obtained on a ring, with a sphere it was challenging. Furthermore, in order to get identical maximum distances for every pair of points, an alternate force should have been utilized.

Sung Jun Kim
003258 -067
South Island School

Contents

Sung Jun Kim
003258 -067
South Island School

Introduction

I had come across this research question through searching up some unsolved general problems and saw the one particular problem "How *should you locate 13 cities on a* spherical planet so that the minimum distance between any two of is as large as possible?"[1]. I had chose this as an inspiration to my current research question from the other range of topics because I was very interested in geometry and shapes, so the title intrigued me, and I had wanted to research about it more. Furthermore the answer to this question may be quite useful. For example, if Google wanted to map the entire earth using satellites, and they had 13 of them, they would need the skill to place the satellites equally far away. This can also improve the Valence Shell Electron Pair Repulsion theory of Molecular Geometry, so that chemists can accurately predict the shapes of molecules. Hence brought forth my research question: "*How should you locate points on a sphere so that the minimum distance between any two of them is as large as possible*?"

Initial Exploration

First of all, my meaning of the distance in a sphere is to be on the surface of the sphere. This would mean I would have to form great circles in order to find the distances between two points on the spherical planet.

I will approach this problem by first starting small with 1, 2, 3 and 4 points on a sphere. By doing this I may find a pattern into going to a larger number of points.

I could try to work out the maximum angle of separation between lines radiating from a point in 3 dimensions with angles between the lines being equal.

The angle of separation between vectors can be found using the dot product, so we can try to make a standard method, for example say working out the maximum angle of separation between 10 lines radiating from a point, with each angle being equal.

---

[1] http://2000clicks.com/MathHelp/PuzzleUnsolved.aspx

Sung Jun Kim
003258 -067
South Island School
Some number of points may have an obvious answer. For 4, 6, 8, 12, or 20 points, one might consider the platonic solids. There is a sphere, and you are able to place n points on it. For these platonic solids their points are equivalent to each other, when you relabel or rotate them. So you would try to do the same in this experiment as well, and make the distance between every pair of points as large as possible. When there is no point, which is special or unique among others, this distribution would be the one that is uniform and spread out.

When there are 4 points, they would form the vertices of a tetrahedron. When there are 6 the points would be vertices of an octahedron. With 8, a cube, 12, an icosahedron, and 20, a dodecahedron. However, are platonic solids really the answer to my question? Platonic solids do indeed have points with their nearest neighbors all equally distanced away. Although they may have congruent edges, and seem to be the answer, I should not automatically assume it without further investigation.

But with 5 points, it gets more complicated. There are some possible layouts I have found such as the vertices of a pentagon or trigonal bipyramidal. However I don't think any of these is the answer. When comparing with a pentagon, it is obvious that a trigonal bipyramidal is the closer answer; however if you take notice of the mid-section, there are 3 points in one great circle, which have all the other points equally distant from each other with the exception of the top and bottom points.

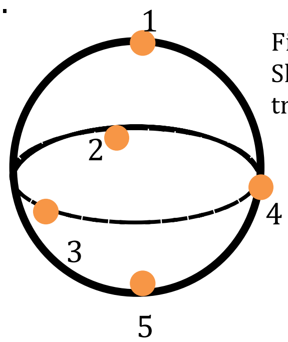


Figure 1
Shape for five points – trigonal bipyramidal

With closer observation 1 is closer to 2, 3 and 4, than with 5, and that 1 is further away from 5 than any point is from 2,3 or 4; therefore no rotation or relabeling can make 1 equivalent to 4. However, this may not be true once we take arc length into account, which is what really matters in the case for my investigation, or move 1 and 5 closer together but then it will no longer be a sphere.

I had also made a theory myself, where maybe a higher average distance can be made when you make several cross sections on the sphere. This will only be necessary when there are more than 3 points. The number of points on one half of the sphere would have to be equal to the other half as well, if the points are evenly spread out. Therefore the number of points on the cross sections/ parts would have to be balanced.

For 4 points, there will be two cross sections on the sphere so that it would be

(2,2)

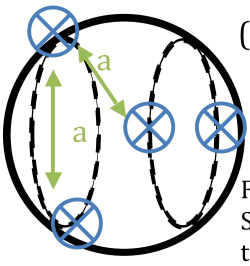


Figure 2
Shape for four points – tetrahedron (proved in my own way)

and if you rotate the sphere the points form a tetrahedron. The distance a is shown here as equal between the two points in a section and also with another in the other cross section. This is so that they will be equally spaced away. This also means that their arc length from one point to the other will be equal.

This could go on for more points, however this only shows a qualitative result. Nothing quantitative or mathematical. The result would be based on my perception making is biased and unreliable.

Therefore I had attempted an algebraic approach, but later discovered that this would also get too complicated later on, and I would need a numerical method. That is how I decided to develop some programming skills.

Sung Jun Kim
003258 -067
South Island School

Numerical Approach

Getting the equations and formulae

The idea I came up with is to use some force on the points which emit from other points, so they can readjust to a position which they are stable at. In other words, I will use interacting particles that repel one another until their forces on each other balances. The force I decided to use is inspired from nature and based on the charge force $F_e = k\frac{q_1 q_2}{r^2}$, where $k = q_1 = q_2 = 1$, as this is the simplest force where objects closer together interact more, but the sign of their relative displacement does not matter.

I will also use spherical polar coordinates to help with my investigation.

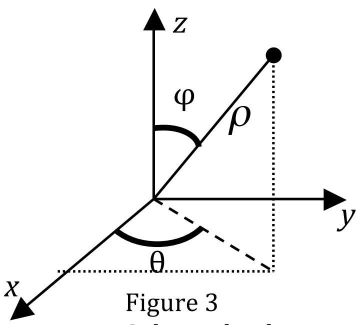


Figure 3
Spherical polar coordinates

Therefore, $x = \rho \sin\varphi \cos\theta$, $y = \rho \sin\varphi \sin\theta$, $z = \rho \cos\varphi$

The choice of a coordinate system is helpful as to make my experiment simpler as I can work in 2 dimensions with the two angles φ and θ. Otherwise I might have had to work with (x, y, z) which could make things more complicated.

Also to make the model of my investigation simple I will say that this spherical planet has a radius of 1. Therefore, distance of any given point from the center is fixed as they are in a sphere. Therefore if the radius of the circle were 1, the length for each position vector would also be one, making $\rho = 1$.

A very interesting possibility arises from confining particles to move on the surface of a sphere. Each particle will fell two unequal forces from every other particle. This is what should allow us to obtain our final configuration of positions up to a rotation of the sphere.

---

[2] http://mathworld.wolfram.com/SphericalCoordinates.html

Another point to enforce is that the force on one particle due to the next would have to always act only on the great circle connecting the two.

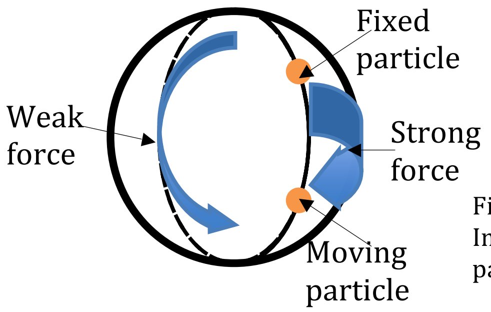


Figure 4
Interacting forces on particles on a great circle

My main equation will look something like this: $m\vec{a}_j = \sum_i \vec{F}_{ij}$

Where the mass (m) will equal 1 to simplify, a is acceleration, i and j are two different particles, F is the Force.

The force will look like: $\frac{1}{r_{ij}^{\,2}} - \frac{1}{r_{ji}^{\,2}}$. This is due to the force from one side being stronger than the other side.

$F = ma$

$\Delta v = at$

$\Delta x = vt$

$latitude : \delta \quad \rightarrow \quad$ colatitude φ of spherical coordinate $\delta = 90 - \varphi$

$longitude : \lambda \quad \rightarrow \quad \lambda = 90 - \theta$

$$\vec{r} = \rho \begin{bmatrix} \cos\lambda\cos\delta \\ \sin\lambda\cos\delta \\ \sin\delta \end{bmatrix}$$

[3]

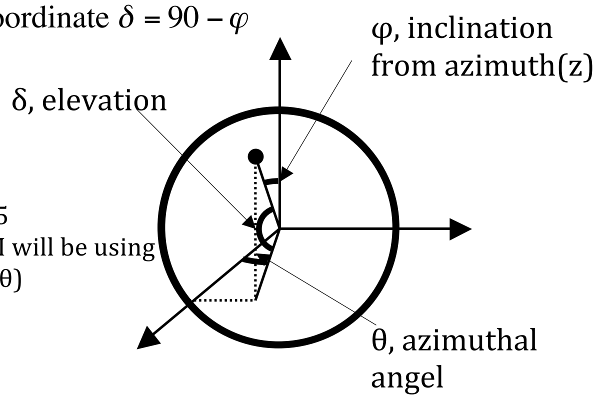


Figure 5
Angles I will be using
(φ and θ)

Figure 6
Conversion table for φ, θ, δ and λ with limits/boundaries

Conversion			
φ	θ	δ	λ
0 to π	0 to 2π	[-π/2, π/2]	[-π, π]

Given two points P(1) and P(2). To complete the sphere distance, we can use arclength $l = \alpha\rho$ (derived from formula $l = \theta r$).

[3] http://mathworld.wolfram.com/SphericalCoordinates.html

Sung Jun Kim
003258 -067
South Island School

$$\cos\alpha = \cos\delta_1\cos\delta_2(\sin\lambda_1\sin\lambda_2 + \cos\lambda_1\cos\lambda_2) + \sin\delta_1\sin\delta_2$$
$$= \cos\delta_1\cos\delta_2(\cos(\lambda_1 - \lambda_2)) + \sin\delta_1\sin\delta_2$$

Then distance between two points on a sphere

$$r = \rho(\cos^{-1}[\cos\delta_1\cos\delta_2\cos(\lambda_1 - \lambda_2) + \sin\delta_1\sin\delta_2]$$

$$F = \frac{1}{r^2}$$

unit vectors in a unit sphere ($\rho = 1$)

$$\hat{r} = \sin\theta\cos\varphi\hat{x} + \sin\theta\sin\varphi\hat{y} + \cos\theta\hat{z}$$
$$\hat{\theta} = \cos\theta\cos\varphi\hat{x} + \cos\theta\sin\varphi\hat{y} - \sin\theta\hat{z}$$
$$\hat{\varphi} = -\sin\varphi\hat{x} + \cos\varphi\hat{y}$$

Suppose (φ_1, θ_1) …… (φ_n, θ_n) is solution. Then any rotation of the sphere will get us another solution. To simplify this, we fix one point at the top of the sphere (0,0).

To solve this numerically, I decided to use MATLAB.

<u>Using MATLAB</u>

My first goal would be to investigate two points. I will fix one point and see whether the other point ends up diametrically opposite.

P(1): (φ_1, θ_1)

P(2): (φ_2, θ_2)

Point 1 is fixed to (0,0). Point 2 is allowed to move, and its initial conditions are picked to be (π/2,0) with no initial velocity.

However with two points, you will be able to create a great circle no matter where they are. It follows that this problem can be reduced from a 2D problem to a 1D problem. Therefore we'll only work with phi for now, making the particles:

P(1): (φ_1), which is the fixed point, and P(2): (φ_2), which is allowed to move.

$F_{net} = ma$

$v = \dot{x} = \dfrac{dx}{dt}$

$a = \dot{v} = \dfrac{d^2x}{dt^2} = \ddot{x}$

since arc length is the product of radius and its angle,

it follows $r = \rho\varphi$

$\therefore F_{net} = m\ddot{x} = m\rho\ddot{\varphi}$

And as I try to work it further :

$F_{net} = ma$

$k\dfrac{q_1q_2}{r^2} = m\ddot{x}$

$k\dfrac{q_1q_2}{(\varphi_2 - \varphi_1)^2} = m\rho\ddot{\varphi}_2$

$k = q_1 = q_2 = m = \rho = 1$

$\dfrac{1}{(\varphi_2 - \varphi_1)^2} = \ddot{\varphi}_2$

but there is another force on the opposite of the geodesic

$\therefore \ddot{\varphi}_2 = \dfrac{1}{(\varphi_2 - \varphi_1)^2} - \dfrac{1}{[2\pi - (\varphi_2 - \varphi_1)]^2}$

This is a second order differential equation, and may be solved numerically by treating it as a system of two first order differential equations:

$$\dot{z}_1 = z_2$$
$$\dot{z}_2 = \frac{1}{(z_1 - \varphi)^2} - \frac{1}{[2\pi - (z_1 - \varphi)]^2}$$

Taking φ = 0 for fixed point, I use MATLAB's ODE solver based on Runge-Kutta algorith to obtain $\varphi_2(t)$.

Now, I can use these equations and input it into my M-File. I will use a comment to remind me of the layout, thus:

```
%: phi_2"=1/(phi_2-phi_1)^2-1/(2*pi-(phi_2-phi_1)^2)
```

In MATLAB a function file will be in the following form[4]:

```
function f = odefilename(t,y)
f = < Insert a function of t and/or y here >;
```

[4] http://coweb.cc.gatech.edu/process/197

In my case I won't use f but z as my phi and by saving the M-File as test.m it contains the following function:

```
function zprime=test(t,z)
zprime=[z(2);1/(z(1)-0)^2-1/(2*pi-(z(1)-0))^2]
```

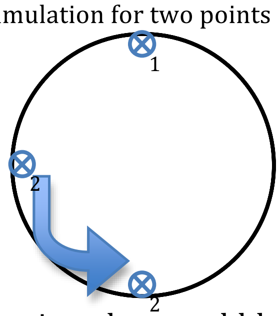
I have a zero for my phi_1 as that point is fixed and won't change.

In the command window I will need to set the initial position and velocity for my point and time span. I will then use ode45 to help formulate data for my graph and then plot z against t.

```
z0=[pi/2;0];
```

This helps me set the initial position of the point as pi/2 (which would have to be greater than 0 and less than 2pi) and its initial velocity as 0.

Figure 7
Simulation for two points



tspan=[0 100];

– This shows my time span ranges from 0 to 100

[t,z]=ode45('test',tspan,z0); – This will help provide data.

plot(t,z) – and this plots a graph for z against t.

My guess for the graph would be that the line converges at pi as that would be the furthest point away.

However the graph I get is:

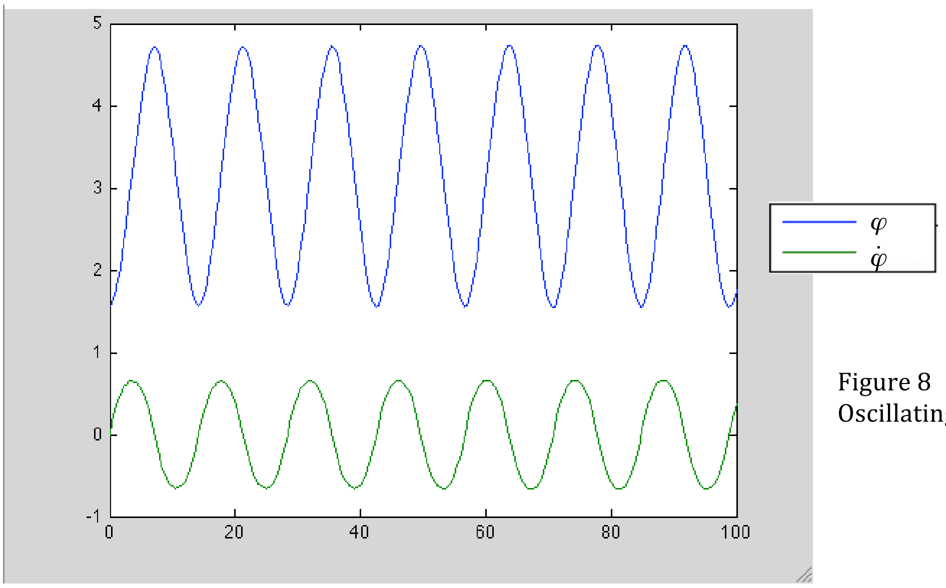


Figure 8
Oscillating graph

The blue line shows the position of the particle and the green line shows its velocity. The particle appears to oscillate about an equilibrium point of π. This shows that I need a damping force for my equation to settle the particle.

<u>Damping Force</u>

From physics F_{damp} is proportional to velocity so I'll add: $F_{damp} = -b\dot{\varphi}$, where b is the damping coefficient. Therefore my equation will then be $m\vec{a}_j = \sum_i \vec{F}_{ij} - b\vec{v}_j$.

Therefore I will add '`-5*z(2)`' to my equation. Now I get this graph.

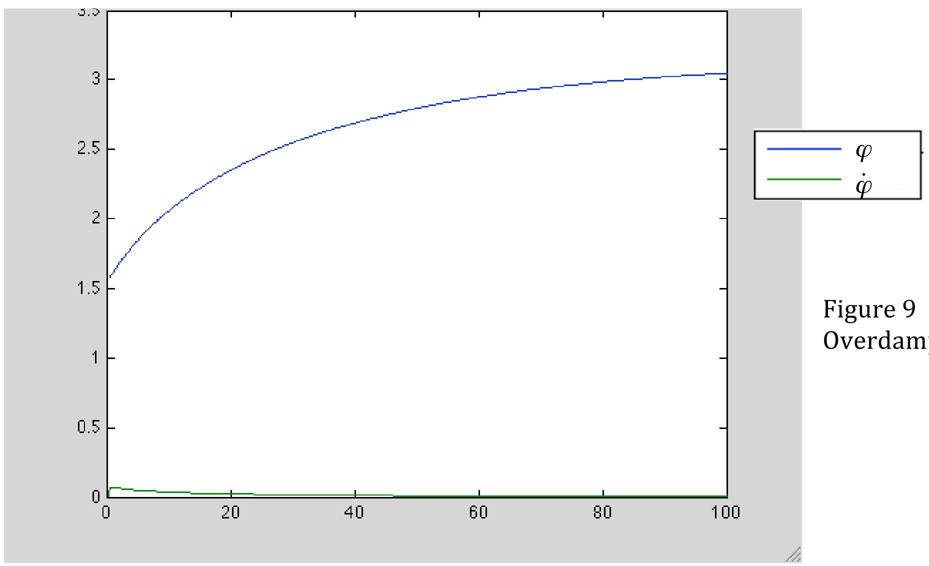


Figure 9
Overdamped

This graph shows clearly how the point becomes into π, however, I believe this is overdamped, leading to conclude that I would need to decrease the damping force into `-0.5*z(2)`

Now the graph I get is critically damped:

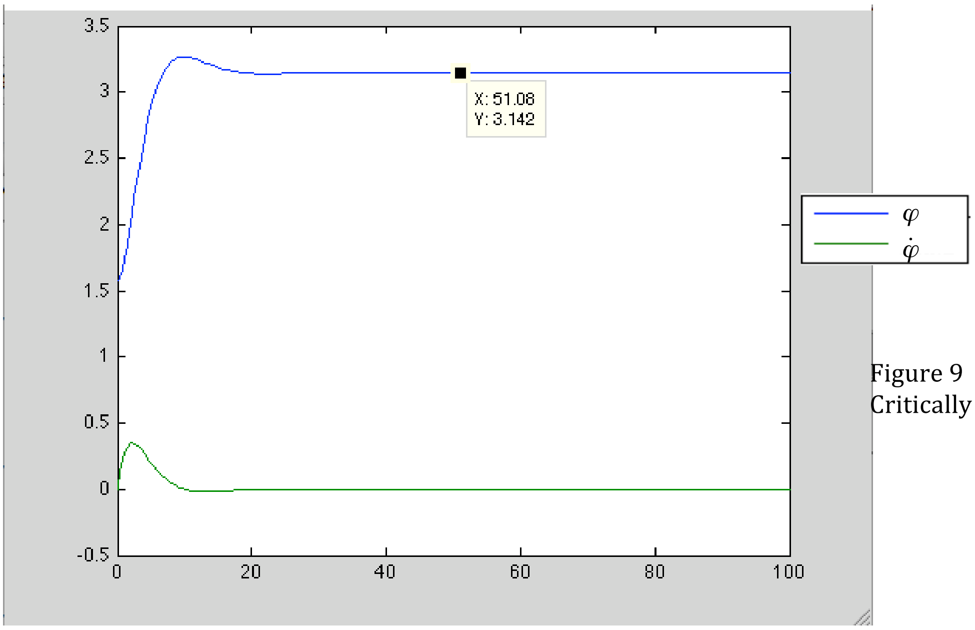


Figure 9
Critically damped

This shows that the programming works and my hypothesis are right. Now I will add another point to this model. With two points we can create a unique great circle no matter where their positions are located. However with three points, there may not be a great circle connecting them. This might mean the one dimension we are working on may not work, but it is clear that once the forces interact and push the points away, they will eventually line themselves up into one great circle; therefore working in the current situation is still applicable. Also, I am still working in the geodesic of a great circle as with three points the shape would be an equilateral triangle.

However, I will test to see if it works on three points but without the forces of the 2nd and 3rd point interacting. This is to make sure that adding another point won't create a problem to the overall situation. This means that these two points should converge at the opposite side of the first point into π.

Therefore my new M-File names test2.m will become:

```
function zprime=test2(t,z)
zprime=[z(2);1/(z(1)-0)^2-1/(2*pi-(z(1)-0))^2-
0.5*z(2);z(4);1/(z(3)-0)^2-1/(2*pi-(z(3)-0))^2-0.5*z(4)]
```

Now my initial points would then be, say, π/2 and 3π/2.

z0=[pi/2;0;3*pi/2;0];

and I will make the rest the same and input

[t,z]=ode45('test2',tspan,z0);

to get a graph from my new M-File, which shows:

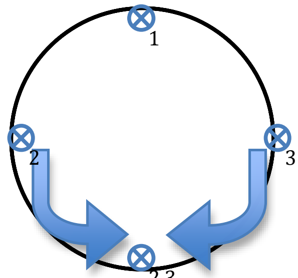


Figure 10
Simulation with three points where the two moving points don't interact

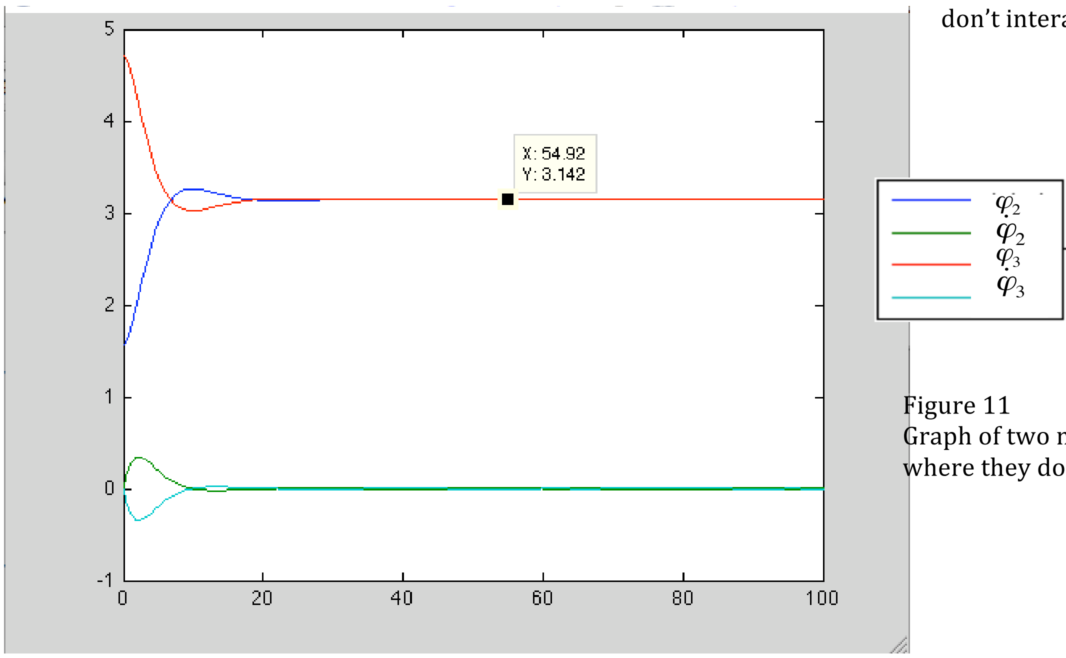


Figure 11
Graph of two moving points where they don't interact

I will then change my initial positions to see if they still converge at π. My new initial positions are:

z0=[1;0;3;0];

The new graph shows:

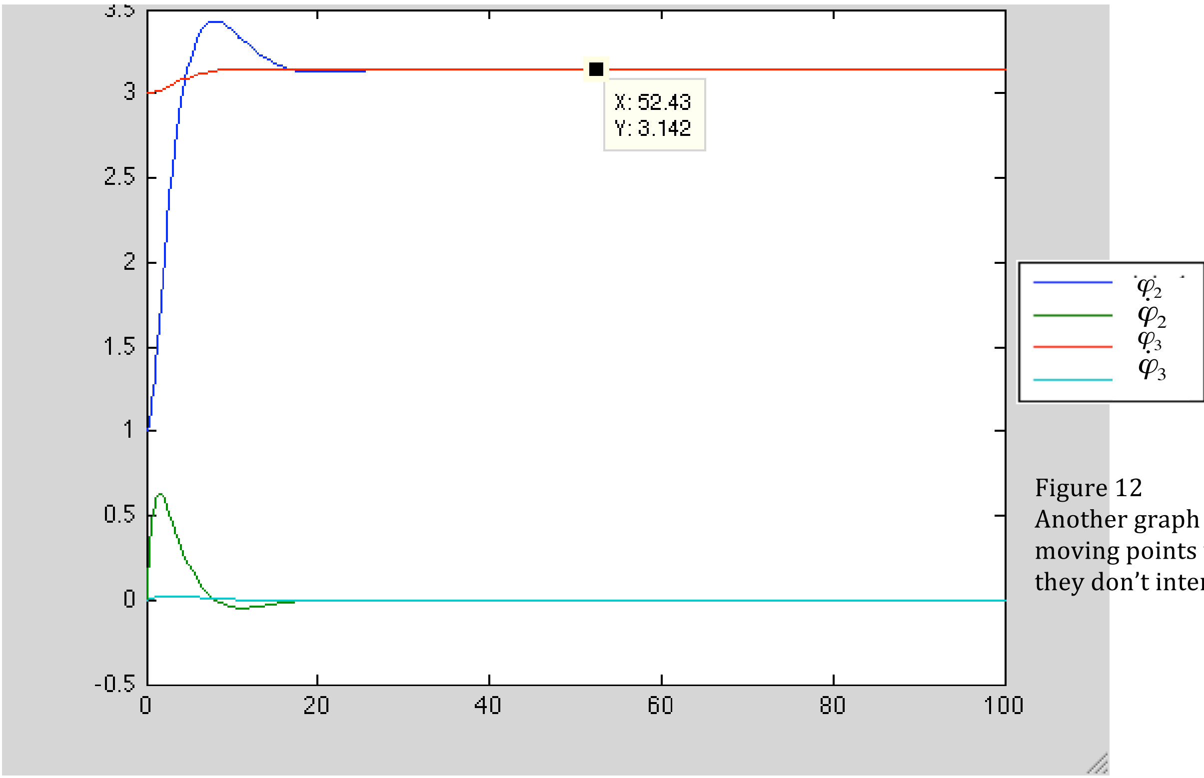


Figure 12
Another graph of two moving points where they don't interact

This proves that the formula still works. Now I will need to input the forces on these two points interacting with each other. To do that all I have to do is add the same formula I used.

Also the interacting term, in second point due to the third is $F_{23,\mathrm{int}} = \dfrac{1}{(\varphi_2 - \varphi_3)^2} - \dfrac{1}{[2\pi - (\varphi_2 - \varphi_3)]^2}$, and in third point due to second is $F_{32,\mathrm{int}} = -F_{23,\mathrm{int}}$, by Newton's third law of action, reaction pairs.

For the second point I will add $+\dfrac{1}{[z(1)-z(3)]^2} - \dfrac{1}{\{2\pi - [z(1)-z(3)]\}^2}$. And for the third point it will be the same but opposite forces, $-\dfrac{1}{[z(1)-z(3)]^2} + \dfrac{1}{\{2\pi - [z(1)-z(3)]\}^2}$.

Sung Jun Kim
003258 -067
South Island School
Which leads me to change my M-File with a new name test3.m into:

```
function zprime=test3(t,z)
zprime=[z(2);1/(z(1)-0)^2-1/(2*pi-(z(1)-0))^2-
0.5*z(2)+1/(z(1)-z(3))^2-1/(2*pi-(z(1)-
z(3)))^2;z(4);1/(z(3)-0)^2-1/(2*pi-(z(3)-0))^2-0.5*z(4)-
1/(z(1)-z(3))^2+1/(2*pi-(z(1)-z(3)))^2]
```

If I do the same method (initial points as z0=[1;0;3;0];) with this new m-file I get:

An error? Showing:

```
Warning: Failure at t=1.785160e+00.  Unable to meet integration
tolerances without reducing the step size below the smallest value
allowed (3.552714e-15) at time t.
> In ode45 at 371
```

Fixing an error

The initial conditions were φ_2 = 1 and φ_3 = 3. However, if I change the points so that the position of the second point is larger than the third point, in order words, without loss of generality let $\varphi_2 > \varphi_3$. So my new initial conditions are φ_2 = 3 and φ_3 = 1 then the graph obtained is:

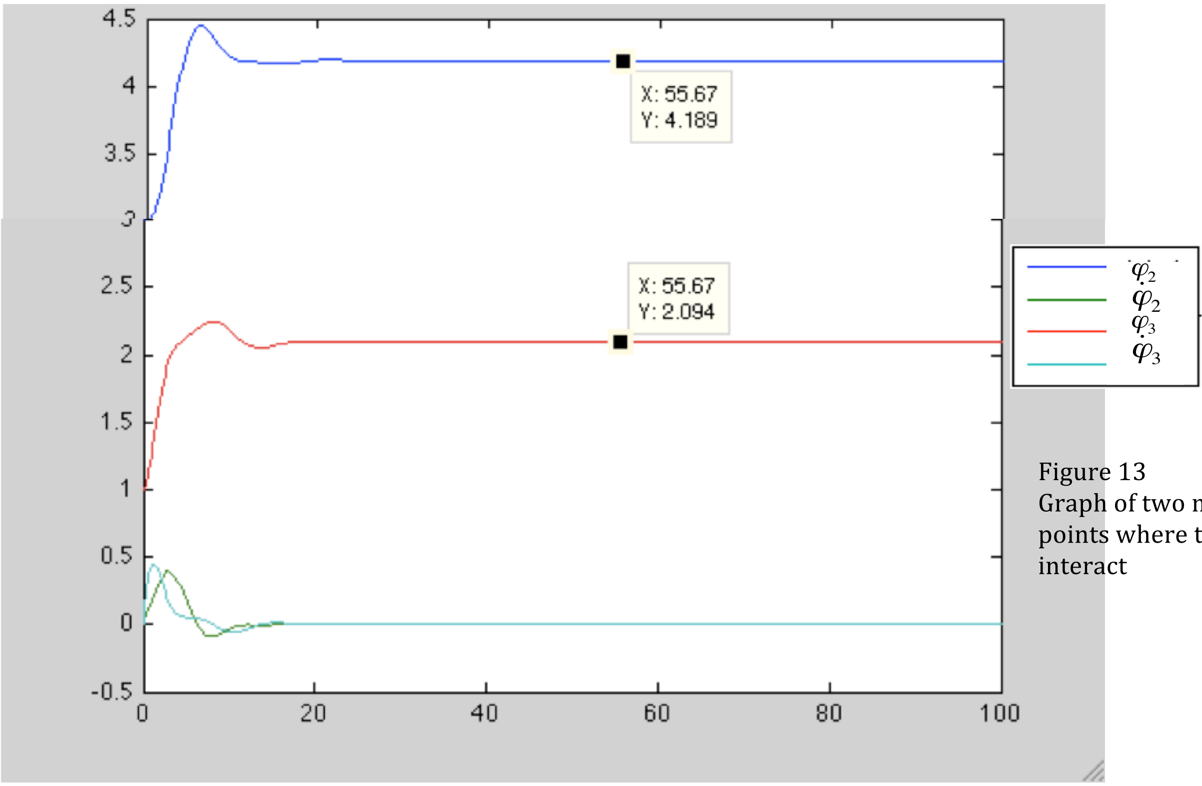


Figure 13
Graph of two moving points where they do interact

This shows that one is on 2π/3 and the other on 4π/3, meaning that the three points make a triangle which is the correct shape. Now, the only problem I need to fix for this problem is that I would need to find the reason why it doesn't work for all values for the position of the points but when one is larger than the other.

First I will give some conditions to the arrangements.

Lets say:

1. Second point's position is smaller than third point.
2. Positive force is anticlockwise and negative force is clockwise
3. Talking about forces relative to the second point

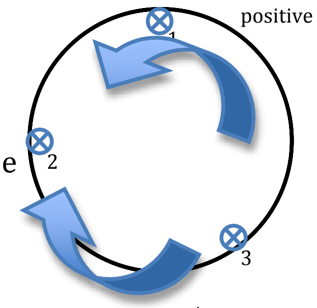


Figure 14
Net force of particles

The positive force acting on the second point is $\frac{1}{\{2\pi - [z(3) - z(1)]\}^2}$ and the negative force is $\frac{1}{[z(1) - z(3)]^2}$.

If I change the conditions so that the second point is positioned ahead of the third point:

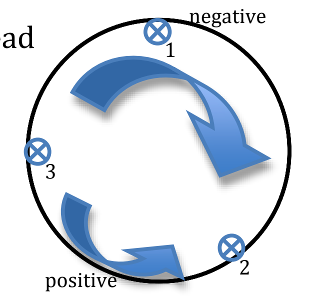


Figure 15
Net force of particles

The positive force acting on the second point is $\frac{1}{[z(1) - z(3)]^2}$ and the negative force is $\frac{1}{\{2\pi - [z(1) - z(3)]\}^2}$.

From these results, by using absolute and sign function (+1 when the number is greater than zero, -1 when number is less than zero) I am able to make my model work for all initial positions, except overlapping ones.

The sign in front of any force would be shown by the sign that results from the relative point subtracted by the point form, which the force comes from. Also the absolute value for the force is needed before subtracting from 2π.

Therefore my new M-File will look like this:

```
function zprime=test3(t,z)
zprime=[z(2);1/(z(1)-0)^2-1/(2*pi-(z(1)-0))^2-
0.5*z(2)+sign(z(1)-z(3))*1/(z(1)-z(3))^2-sign(z(1)-
z(3))*1/(2*pi-abs(z(1)-z(3)))^2;z(4);1/(z(3)-0)^2-
1/(2*pi-(z(3)-0))^2-0.5*z(4)+sign(z(3)-z(1))*1/(z(1)-
z(3))^2-sign(z(3)-z(1))*1/(2*pi-abs(z(1)-z(3)))^2]
```

The above simply shows that the equation for an interacting force becomes, $F_{23,\text{int}} = sign(\varphi_2 - \varphi_3)\frac{1}{(\varphi_2 - \varphi_3)^2} - sign(\varphi_2 - \varphi_3)\frac{1}{[2\pi - abs(\varphi_2 - \varphi_3)]^2}$, whe

And if I use this for when z0=[3;0;1;0];

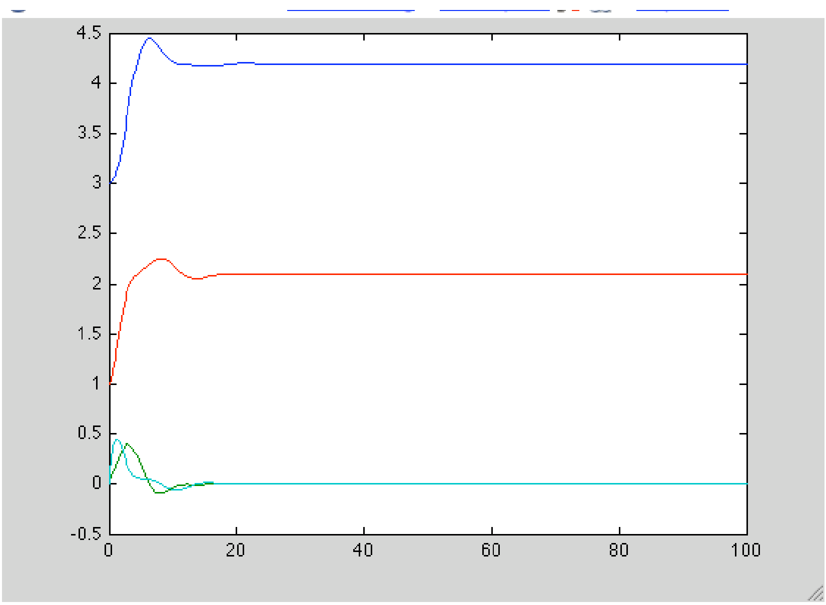


Figure 16
Graph of two moving points where they do interact

and when z0=[1;0;3;0];

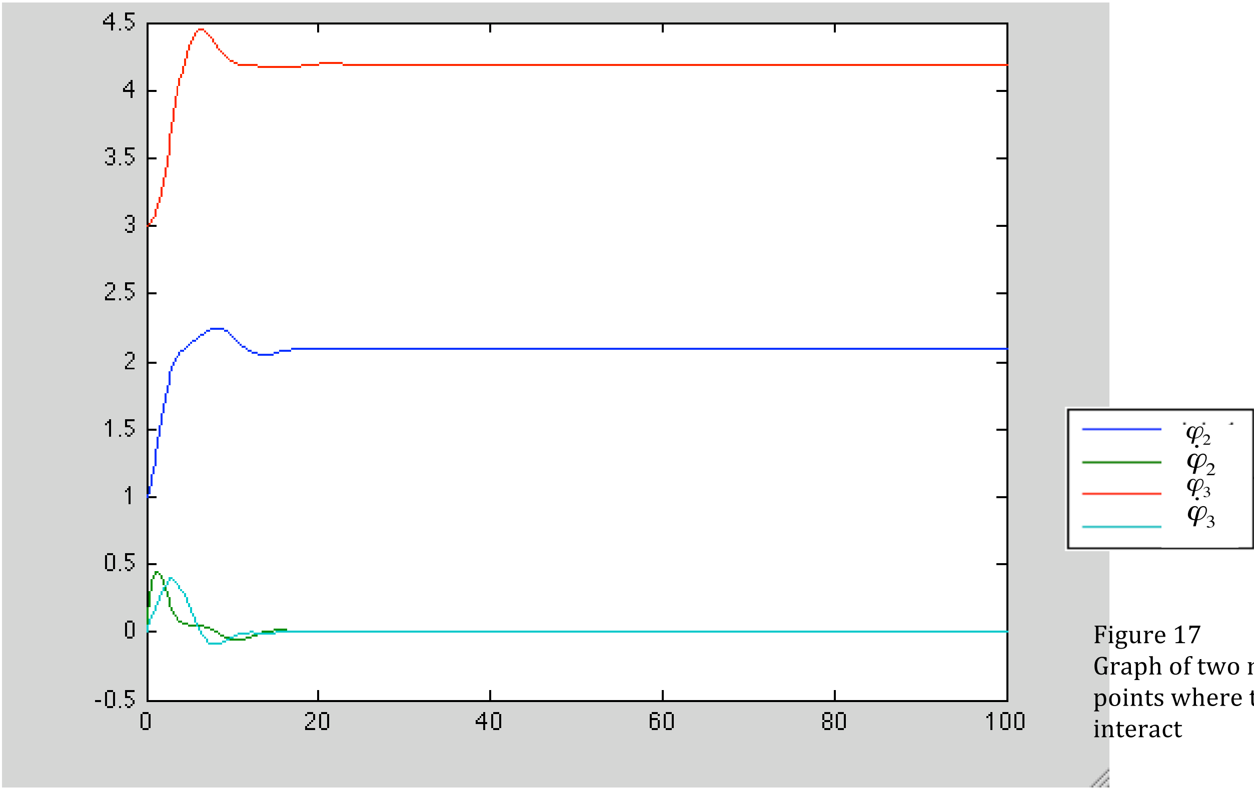


Figure 17
Graph of two moving points where they d interact

Both positions, when either is bigger than the other, works.

Now I will add one more point so that there are 4 points. I will still work in one great circle to see my result.

Here I would need to add more positions and more forces on each point. There will now be 3 forces on each point. However, it isn't that hard now as I can use the same method to get each force. Formulaically, you can see by:

$$F_{i,\text{int}} = \sum_{j \neq i} F_{ij}$$

where $F_{ij} = \dfrac{1}{(\varphi_i - \varphi_j)^2} - \dfrac{1}{[2\pi - (\varphi_i - \varphi_j)]^2}$

and $F_{ji} = -F_{ij}$

Sung Jun Kim
003258 -067
South Island School
My new M-File will thus look like:

```
function zprime=test5(t,z)
zprime=[z(2);1/(z(1)-0)^2-1/(2*pi-(z(1)-0))^2-
0.5*z(2)+sign(z(1)-z(3))*1/(z(1)-z(3))^2-sign(z(1)-
z(3))*1/(2*pi-abs(z(1)-z(3)))^2+sign(z(1)-z(5))*1/(z(1)-
z(5))^2-sign(z(1)-z(5))*1/(2*pi-abs(z(1)-
z(5)))^2;z(4);1/(z(3)-0)^2-1/(2*pi-(z(3)-0))^2-
0.5*z(4)+sign(z(3)-z(1))*1/(z(1)-z(3))^2-sign(z(3)-
z(1))*1/(2*pi-abs(z(1)-z(3)))^2+sign(z(3)-z(5))*1/(z(5)-
z(3))^2-sign(z(3)-z(5))*1/(2*pi-abs(z(5)-
z(3)))^2;z(6);1/(z(5)-0)^2-1/(2*pi-(z(5)-0))^2-
0.5*z(6)+sign(z(5)-z(1))*1/(z(1)-z(5))^2-sign(z(5)-
z(1))*1/(2*pi-abs(z(1)-z(5)))^2+sign(z(5)-z(3))*1/(z(3)-
z(5))^2-sign(z(5)-z(3))*1/(2*pi-abs(z(3)-z(5)))^2]
```

φ_2 And then if I just input any random initial positions, z0=[4;0;5;0;1;0]; and run it through the new M-File, the graph I get is:

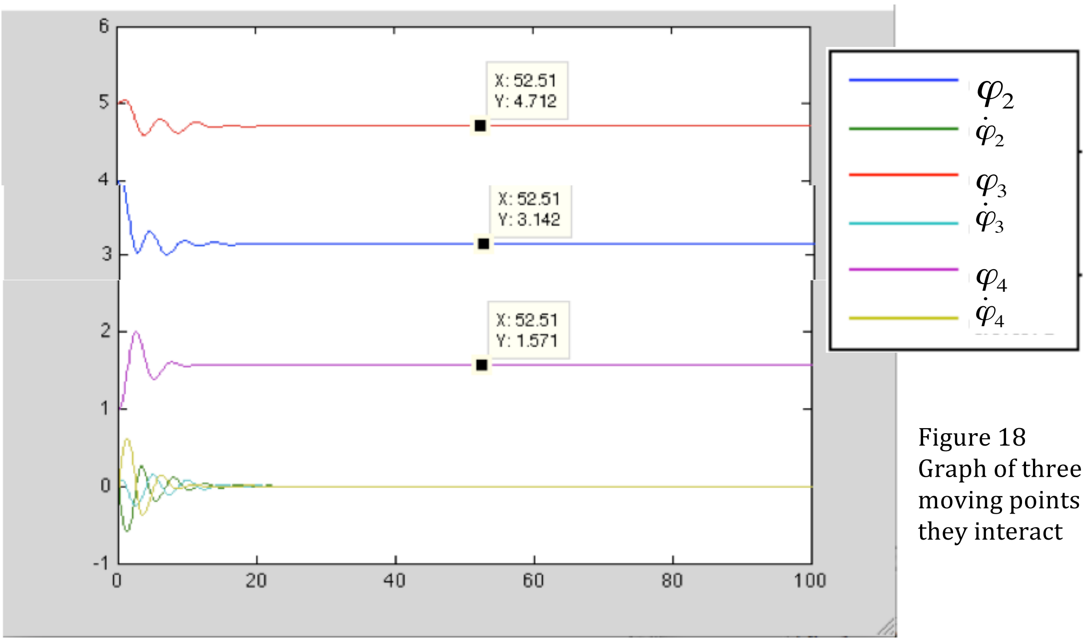


Figure 18
Graph of three moving points where they interact

From this graph the points make the shape of a square. This shows that the model indeed works. Now I will have to not work with a great circle, but with a sphere.

Sung Jun Kim
003258 -067
South Island School

Moving to a sphere

The foundation is still the same: $m\vec{a}_j = \sum_i \vec{F}_{ij} - b\vec{v}_j$, where $\sum_i \vec{F}_{ij}$ is the interacting force and $b\vec{v}_j$ is the damping force.

However, we need to now implement the kinematics in spherical coordinates:

$$\vec{r} = r\hat{r}$$
$$\vec{v} = \dot{r}\hat{r} + r\dot{\varphi}\hat{\varphi} + r\dot{\theta}\sin\varphi\hat{\theta}$$
$$\begin{aligned}\vec{a} &= (\ddot{r} - r\dot{\varphi}^2 - r\dot{\theta}^2\sin^2\varphi)\hat{r} \\ &+ (r\ddot{\varphi} + 2\dot{r}\varphi - r\dot{\theta}^2\sin\varphi\cos\varphi)\hat{\varphi} \\ &+ (r\ddot{\theta}\sin\varphi + 2\dot{r}\dot{\theta}\sin\varphi + 2r\dot{\varphi}\dot{\theta}\cos\varphi)\hat{\theta}\end{aligned}$$

We can simplify this as we are constraining the particle on a sphere; therefore r = 1 meaning $\dot{r}$ = 0 and that we can ignore the $\hat{r}$ part as we can't move in or out of the sphere, but just on the surface. Also, we can say m = 1 to simplify even more. By substituting these values into our original equation we get:

$$\begin{aligned}&(\ddot{\varphi} - \dot{\theta}^2\sin\varphi\cos\varphi)\hat{\varphi} + (\ddot{\theta}\sin\varphi + 2\dot{\varphi}\dot{\theta}\cos\varphi)\hat{\theta} \\ &= \sum_i \vec{F}_{ij} - b(\dot{\varphi}\hat{\varphi} + \dot{\theta}\sin\varphi\hat{\theta})\end{aligned}$$

This is a vector equation, so looking at the components, I can get two equations.

$$\hat{\varphi} : \ddot{\varphi} - \dot{\theta}^2\sin\varphi\cos\varphi = \sum_i \frac{1}{{r_{ij}}^2}\frac{\Delta\varphi}{r_{ij}} + \sum_i \frac{1}{(2\pi - r_{ij})^2}\frac{\Delta\varphi}{2\pi - r_{ij}} - b\dot{\varphi}$$
$$\hat{\theta} : \ddot{\theta}\sin\varphi + 2\dot{\varphi}\dot{\theta}\cos\varphi = \sum_i \frac{1}{{r_{ij}}^2}\frac{\Delta\theta}{r_{ij}} + \sum_i \frac{1}{(2\pi - r_{ij})^2}\frac{\Delta\theta}{2\pi - r_{ij}} - b\dot{\theta}\sin\varphi$$

Through these two second order differential equations, I can solve it numerically by treating it as a system of four first order differential equations:

$$\varphi = z_1, \dot{\varphi} = z_2, \theta = z_3, \dot{\theta} = z_4$$
$$\dot{z}_1 = z_2$$
$$\dot{z}_2 = {z_4}^2\sin z_1\cos z_1 + \sum_i \frac{1}{{r_{ij}}^2}\frac{\Delta\varphi}{r_{ij}} + \sum_i \frac{1}{(2\pi - r_{ij})^2}\frac{\Delta\varphi}{2\pi - r_{ij}} - bz_2$$
$$\dot{z}_3 = z_4$$
$$\dot{z}_4 = \frac{1}{\sin z_1}[-2\cos z_1 z_2 z_4 + \sum_i \frac{1}{{r_{ij}}^2}\frac{\Delta\theta}{r_{ij}} + \sum_i \frac{1}{(2\pi - r_{ij})^2}\frac{\Delta\theta}{2\pi - r_{ij}} - bz_4\sin z_1$$

Modulus

I also implemented more functions than I did before. One is modulus. This is implemented because of arc length purposes as to limit the magnitude of the forces. If the particle goes haywire and jumps around, the angle can accumulate to a huge number, whilst realistically the angle is always limited. For φ, it ranges from 0 to π. However, we cannot just do φ mod π. This is because theoretically when a particle is pushed around, and the φ angle changes, lets say, from π-1 to π+1. The actual φ angle of this new position would be π-1, not π+1 mod π, which would be 1. This is shown evidently in the diagram:

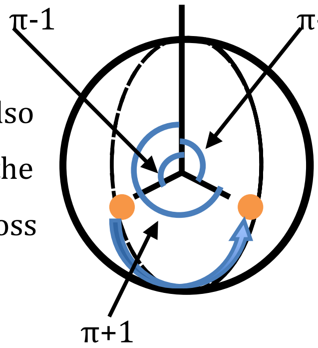


Figure 19
Explaining how modulus works in φ

Therefore it should be π – φ mod π. However it also changes the θ as well, by adding π to the angle, as the particle instantly gains that much angle if it goes across the top or bottom of the sphere.

For θ it is just θ mod 2π.

Normal direction

Another function, which I had implemented, is the normal direction function. This will help provide the direction of the vector of the particle. This is done by crossing several vectors.

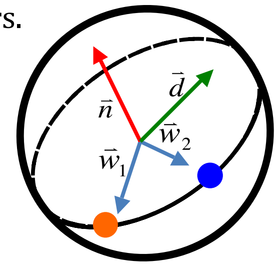


Figure 20
Explaining how normal direction works

Here the orange particle is fixed, and the blue particle is free to move. Through crossing the vectors we can get the direction the blue particle will go to.

$$\vec{n} = \vec{w}_1 \times \vec{w}_2$$
$$\vec{d} = \vec{w}_2 \times \vec{n}$$

as $\vec{d}$ would be the direction of the moving particle, we can convert it and get the next positions for the φ and θ angle $(\varphi_{new}, \theta_{new})$. Then, with this, I am able to get $(\varphi_{new}, \theta_{new}) - (\varphi, \theta) = (\Delta\varphi, \Delta\theta)$.

Sung Jun Kim
003258 -067
South Island School
I had created a new m-file called normdir.m for this function. The m-file contains:

```
function [deltaphi,deltatheta]=normdir(p0,t0,phi,theta)
delta=.5;
p.phi=phi;
p.theta=theta;
[x0,y0,z0]=sph2cart(t0,pi/2-p0,1);
[p.x,p.y,p.z]=sph2cart(p.theta,pi/2-p.phi,1);
dir=cross(cross([x0,y0,z0],[p.x,p.y,p.z]),[p.x,p.y,p.z]);
dir=dir/norm(dir);
pnew.x=p.x+delta*dir(1);
pnew.y=p.y+delta*dir(2);
pnew.z=p.z+delta*dir(3);
[thetanew,phinew,rnew]=cart2sph(pnew.x,pnew.y,pnew.z);
phinew=pi/2-phinew;
deltaphi=(phinew-p.phi);
deltatheta=(thetanew-p.theta);
```

Generating Sphere and Vectors

Further functions that I added include generating translucent spheres, and showing vector line, in order for me to see clearly what is happening to the particles throughout the investigation. The commands for these are:

```
xcart=sin(z(:,1)).*cos(z(:,3))
ycart=sin(z(:,1)).*sin(z(:,3))
zcart=cos(z(:,1))
xdot=cos(z(:,1)).*z(:,2).*cos(z(:,3))+sin(z(:,1)).*z(:,4)
*(-1).*sin(z(:,3));
ydot=cos(z(:,1)).*z(:,2).*sin(z(:,3))+sin(z(:,1)).*z(:,4)
.*cos(z(:,3));
zdot=sin(z(:,1)).*z(:,2);
[x,y,z]=sphere;
figure;
hold on;
surf(x,y,z);
colormap gray;
alpha(.4)
plot3(xcart,ycart,zcart,'r');
quiver3(xcart,ycart,zcart,xdot,ydot,zdot,'g-');
hold off;
daspect([1 1 1])
```

Results on Sphere

With these command and settings, I will attempt to do two particles in a sphere, and see what happens.

For a sphere, I would not put my fixed particle at the top. This is because of a part in the equation, which is $\frac{1}{\sin z_1}$. Here, z_1 = φ and if φ becomes 0 or 2π, and the sine of these will be zero, so the equation will end up exploding as the divisor is zero. Therefore, I will make my fixed point at (π/2, 0) and my initial point (2π/3, π/2) with no initial velocity. Presumably, it will end at (π/2, π)

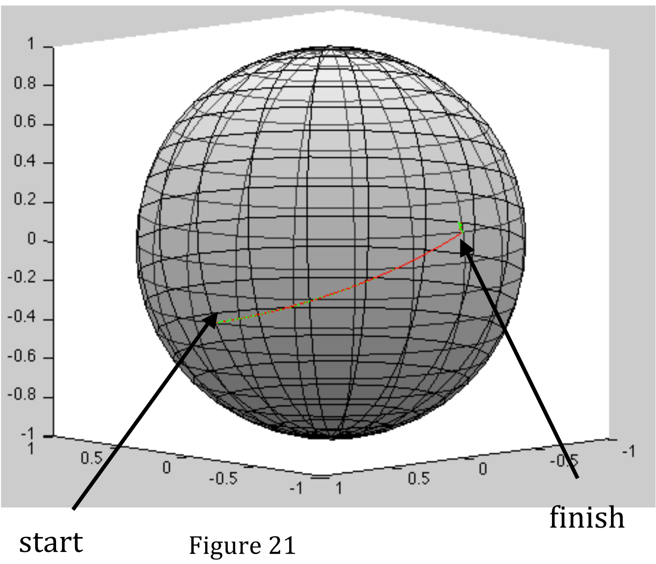


Figure 21
Simulation of a moving particle interacting with the fixed particle on a sphere.

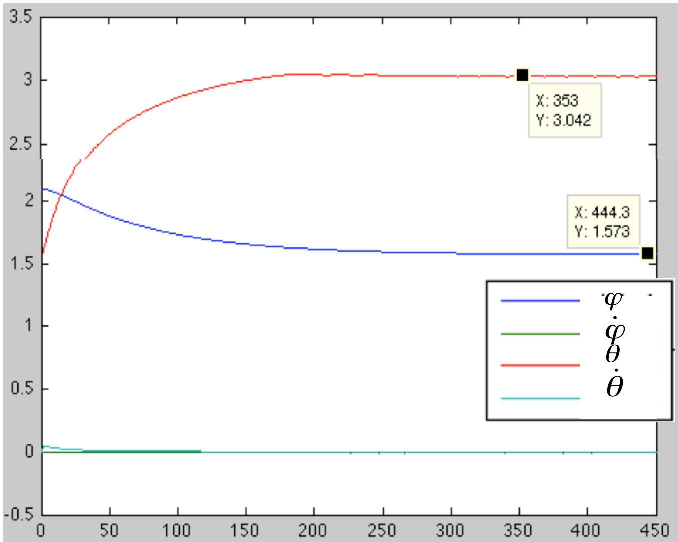


Figure 22
Graph of one moving point on a sphere

Although there seems to be a small uncertainty in which the particle doesn't fully reach the desired position, but it is inarguable that it is near it.

However, once I add another point, there seems to be an error somewhere as no matter what I do, I don't seem to get the right answer.

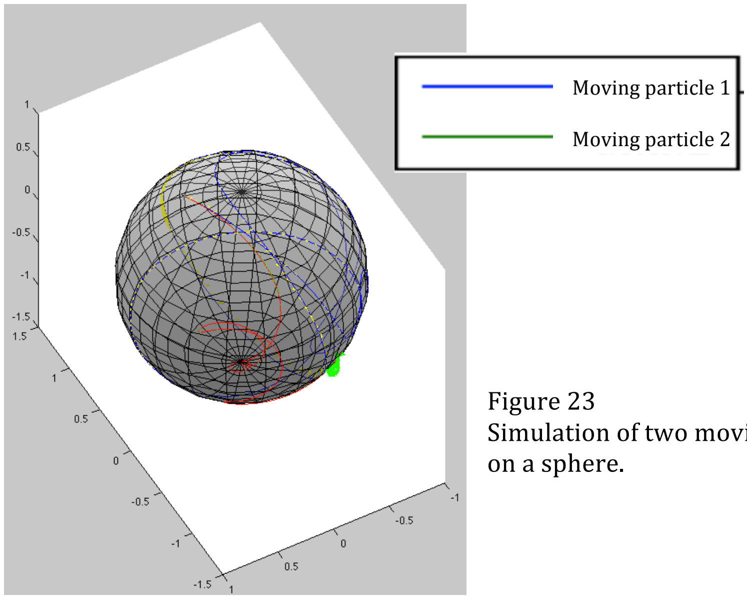


Figure 23
Simulation of two moving particles interacting on a sphere.

Although my theory and formula may be correct, there seems to be some problem to apply it to the MATLAB functions in order to get proper results.

Sung Jun Kim
003258 -067
South Island School

Conclusion

I see this as a fairly interesting problem in a purely mathematical point of view as there are not many solutions to this question. With the two possible interpretations in equal distribution, where every pair is equally distanced apart, and another where the neighboring particles are equally distanced apart. Uniform distribution of particles refer to all the neighboring points being equally distanced away, though an alternate better definition would be every pair of points' distance is as large as possible.

I was able to solve the problem on the ring, whilst solving the problem on the sphere was technically challenging and I faced various challenges such as trying to implement both φ and θ angles. The suggested possible numerical approaches I had tried require more work, and in order to solve the problem for n particles I would need 4n first order differential equations, which would be a somewhat time-consuming task. If I had went to 13 points, I would need up to 52 first order differential equations. Also one possible reason why the model as of yet doesn't work is the numerical switching of theta when phi exceeds π. In future work, this should be implemented. Also, one limitation of this suggested strategy is that it is expected to solve for a set of points, which are more or less uniformed distributed on the sphere. However, it doesn't give a solution to a set of points for which each distance between any given pair is the same and maximized. For this, another force field that doesn't depend on distance would need to be discovered. I might have needed to come up with another force than $\frac{1}{r^2}$ as this force is not as good as it makes particles which are far away have too small of an effect, making their interaction negligible.

Sung Jun Kim
003258 -067
South Island School
Bibliography:

ComSwiki. “Solving ODEs in Matlab -- tutorial (Difference).” Web. http://coweb.cc.gatech.edu/process/194.diff?id=8

ComSwiki. “Defining an ode function in a M-file.” Web. http://coweb.cc.gatech.edu/process/197

Wikipedia. “Wikipedia: Spherical coordinate system.” Web. http://en.wikipedia.org/wiki/Spherical_coordinate_system#Integration_and_differentiation_in_spherical_coordinates

Weisstein, Eric W. "Spherical Coordinates." From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/SphericalCoordinates.html

Eugen Dedu. “UnsolvedPbs”. Web. http://lifc.univ-fcomte.fr/~dedu/math/unsolvedPbs.txt

Patrick Boucher. “Points on a Sphere.” 4 October 2006. Web. http://www.xsi-blog.com/archives/115