

## Solving Ordinary Differential Equations

Lecture 2

Physics 200  
Laboratory

Monday, February 7th, 2011

Physics (nature?) provides a plethora of ordinary differential equations (relations involving a function of a single variable, its derivatives, and some external function of time, say) – Newton’s second law, the time-independent Schrödinger equation, and the equations governing the generation of spherically symmetric electromagnetic and gravitational fields, are a few examples. While there are a variety of techniques for solving specific problems, and an interesting class of techniques for characterizing the solutions to problems in the absence of an explicit form, it always surprises (slash embarrasses) me that a large number of physically relevant, interesting, and well-posed problems have little in the way of complete solution.

Enter numerical methods. We will start by describing a few different classes of problem that can be set up naturally using physics that you already know, and then suggest that there is a straightforward way to solve these. While giddy with delight, we have to exercise caution – a computer can provide (in the best case) a relevant solution to a specific problem, but it fails to generate insight into the general cases, and can fail miserably even for simple, solvable problems. A panacea, it ain’t, and we will discuss ways of analyzing and quantifying the errors that a computer can (and will) make. What we have is a complementary tool, one to augment our analytic skills and provide information that can extend our analysis. In this series of labs, these numerical methods are meant to take their place, along with partial exact solutions, perturbative observations, special cases, and all the other bits and pieces you have studied, in your collective problem-solving toolbox. One advantage of numerical solutions is that they are often fiendishly simple to describe and understand, even as their deficiencies are profound.

## 2.1 Physical Problems

### 2.1.1 Newton's Second Law

You have encountered a variety of ordinary differential equations already – the first was probably Newton's second law:

$$m \ddot{x}(t) = F(t) \quad (2.1)$$

governing the motion of a body of mass  $m$  under the influence of a force  $F(t)$  (potentially depending on position, velocity, and time).

Many of the forces that you use in Newton's second law yield linear second order ODE's, like the damped driven harmonic oscillator:

$$m \ddot{x} = -k(x(t) - a) - b \dot{x}(t) + F(t). \quad (2.2)$$

There are “exact solutions” to these<sup>1</sup>, and we will use those to test our numerical method.

Remember that the differential equation itself is only part of the physical story – we must also provide initial conditions or boundary conditions. For a second order differential equation like Newton's second law, you must give two constants, and we typically associate these with the position and velocity at time  $t = 0$ , so we would agree to provide  $x(0)$  and  $\dot{x}(0)$ . Or, our observational input might be “the mass was at location  $a$  at time  $t = 0$ , and at location  $b$  five seconds later,” which would amount to giving  $x(0) = a$  and  $x(5 \text{ s}) = b$ . This latter case is called a “boundary value” formulation of the problem of motion. We can provide more exotic inputs, the position at time  $t_0$  and velocity at time  $T$ . In the end, the general constants of integration can be set with a variety of physical inputs – when we write a solution for the simple harmonic oscillator:

$$x(t) = A \cos(\omega t) + B \sin(\omega t) \quad (2.3)$$

we can set  $A$  and  $B$  however we like. You should try inputting a few different flavors to see how the story goes.

---

<sup>1</sup>Exact is an interesting term here – what we typically mean is “writeable in terms of familiar functions”. Most ODEs that come from physical problems have solutions, and then at the very least, can be written as an infinite (convergent) sum. But think about what a familiar function like cosine really is – how would you find the value  $\cos(.1984)$ ? What does a calculator or computer do when you input a radian argument like .1984. My point is that even functions that are easy to write down, like  $\cos(\sqrt{2}t)$ , still require some sort of numerical evaluation.

Aside from linear problems, there are a variety of forces that we can consider that are more difficult, and for which only numerical solution will provide trajectories – consider the following two dimensional example.

### Example

A charged particle enters a region of magnetic field as shown in Figure 2.1. Let  $\mathbf{r}(t) = x(t)\hat{\mathbf{x}} + y(t)\hat{\mathbf{y}}$  describe the location of the particle with respect to the origin. Then the Lorentz force law tells us that the magnetic field exerts a force of the form:  $\mathbf{F} = q\dot{\mathbf{r}}(t) \times \mathbf{B}$ . For the configuration shown below (with a magnetic field pointing out of the page, and the particle moving in the plane of the page),  $\ddot{\mathbf{r}}(t) = \frac{q}{m}\dot{\mathbf{r}}(t) \times \mathbf{B}$  gives a pair of equations:

$$\begin{aligned}\ddot{x} &= \frac{q}{m}\dot{y}B(r) \\ \ddot{y} &= -\frac{q}{m}\dot{x}B(r).\end{aligned}\tag{2.4}$$

Looks simple enough, and yet ...

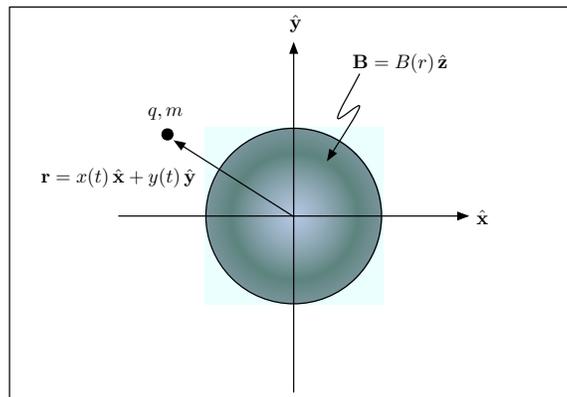


Figure 2.1: A particle enters a region with non-uniform magnetic field (pointing into the page).

So we see that adding multiple dimensions can make our problem more

difficult (read “interesting”), which brings us to.

### 2.1.2 Multiple Particles

Think of the electrostatic force acting on a particle (particle 1, with mass  $m_1$  and charge  $q_1$ ) due to a second particle (particle 2, with mass  $m_2$  and charge  $q_2$ ):

$$\mathbf{F}_{12} = \frac{q_1 q_2}{4 \pi \epsilon_0 r_{12}^2} \hat{\mathbf{r}}_{12} \quad \text{with } \mathbf{r}_{12} = \mathbf{r}_1 - \mathbf{r}_2. \quad (2.5)$$

Now if both particles are free to move, we have a six-dimensional problem to solve – three coordinates for the first particle, three for the second:

$$\begin{aligned} \ddot{\mathbf{r}}_1 &= \frac{q_1 q_2}{4 \pi \epsilon_0 m_1 r_{12}^2} \hat{\mathbf{r}}_{12} \\ \ddot{\mathbf{r}}_2 &= -\frac{1}{m_2} \mathbf{F}_{12}. \end{aligned} \quad (2.6)$$

For a realistic problem involving a configuration of  $n$  charges, we would write the equation of motion for the  $i^{\text{th}}$  particle as:

$$\ddot{\mathbf{r}}_i = \frac{1}{m_i} \sum_{\substack{j=1 \\ j \neq i}}^n \frac{q_i q_j}{4 \pi \epsilon_0 r_{ij}^2} \hat{\mathbf{r}}_{ij}, \quad (2.7)$$

and each particle has three components to fully determine its motion, for a total of  $3n$  “degrees of freedom”.

#### Example

We can pose the multiple-particle problem in terms of a pair potential:  $U(r)$ , provided by the physical problem of interest. Many interactions can be described through such a function, electrostatics and gravity being the most famous (and, tantalizingly, identical in form). Given  $U(r)$ , we can write the force on particle 1 due to particle 2 as:

$$\mathbf{F}_{12} = -U'(r) \hat{\mathbf{r}}_{12} \quad \text{with } \mathbf{r}_{12} = \mathbf{r}_1 - \mathbf{r}_2 \quad (2.8)$$

for separation vector  $\mathbf{r}_{12}$  pointing from particle 2 to particle 1. Then the equations of motion, from Newton's second law, are

$$\ddot{\mathbf{r}}_i = -\frac{1}{m_i} \sum_{\substack{j=1 \\ j \neq i}}^n U'(r_{ij}) \hat{\mathbf{r}}_{ij} \quad \text{for } i = 1, 2, \dots, n. \quad (2.9)$$

### 2.1.3 Problems that are Coming Up

Aside from the problems we have discussed already, there is a lot of additional physics to be had from second order ODEs – in one-dimension (either because of intrinsic interest, or because of symmetry reduction), Schrödinger's equation governing the quantum mechanical wave-function  $\psi(x)$  can be written as:

$$-\frac{\hbar^2}{2m} \frac{d^2\psi(x)}{dx^2} + V(x)\psi(x) = E\psi(x) \quad (2.10)$$

where  $\hbar = 1.05 \times 10^{-34}$  Js (note its size),  $m$  is the mass of the particle,  $V(x)$  is the one-dimensional potential governing the classical motion of the particle, and  $E$  is the energy of the particle. Now the quantum mechanical interpretation of  $\psi(x)$  is in terms of spatial probabilities. In words: "The probability of finding a particle of mass  $m$ , moving under the influence of  $V(x)$ , within  $dx$  of the point  $x$  is  $\psi^*(x)\psi(x)dx$ ." That means, incidentally, that the integral:  $\int_{-\infty}^{\infty} \psi^*(x)\psi(x)dx = ?$ .

#### Example

As an example that we can use immediately, we'll develop the one-dimensional form of Newton's second law for special relativistic dynamics. You know, in classical mechanics, that total energy is conserved for a particle moving with speed  $v$ :

$$E = \frac{1}{2} m v^2 + U(x) \quad (2.11)$$

where  $m$  is the mass of the particle,  $U(x)$  is the potential in which the particle is moving,  $\frac{1}{2}$  is half, and  $E$  is the total energy. By taking the time-derivative of both sides of this equation, we recover Newton's second law for "conservative" forces:

$$0 = m v \dot{v} + U'(x) v \longrightarrow m \ddot{x} = -U'(x) \equiv F(x). \quad (2.12)$$

In special relativity, you will learn that the kinetic energy combines with the "rest energy" ( $m c^2$ ), to form a term replacing the kinetic energy in (2.11) that looks like:  $\frac{m c^2}{\sqrt{1 - \frac{v^2}{c^2}}}$ , so that in special relativity, conservation of energy reads:

$$E = \frac{m c^2}{\sqrt{1 - \frac{v^2}{c^2}}} + U(x). \quad (2.13)$$

You should check (by Taylor expansion) that this conservation statement reduces to the usual one when  $v \ll c$ . If we differentiate both sides of this equation with respect to time, we get a relativistically-correct form of Newton's second law that reads:

$$m \ddot{x} = -U' \left( 1 - \frac{\dot{x}^2}{c^2} \right)^{3/2}, \quad (2.14)$$

and this can be used to solve problems in which particles move very quickly (compared to  $c$ , which in its usual units is  $c \approx 3 \times 10^8$  m/s – a large number). It is also, even for simple  $U'(x)$ , much more difficult to solve than the usual, classical expression.

## 2.2 Nondimensionalization

Before we generate the method, let's think about some actual numbers in the equations we have set up. On a computer, the smallest number that can be represented is called "machine epsilon", and is generally around  $10^{-13}$  (meaning thirteen digits, you can represent large exponents, like  $10^{-400}$ , but you don't get to specify all four hundred digits). That means you have a finite resolution for "0", and anything smaller than machine epsilon

is indistinguishable from zero. Given the awkward numerical values set by our choice of units, Newton's second law can involve computationally unwieldy numerical values – for example, the force between two equal, one kilogram masses separated by one meter, from the gravitational force law, is:  $F \sim 10^{-11}$  N, which is too small for safe use. The other direction (large numbers) is similarly limited, you get around thirteen digits there as well, so the force associated with two equal one Coulomb charges separated by one meter,  $F \sim 10^{10}$  N, is unacceptable in the other direction. In both of these cases, the problem is our choice of Newtons as the unit of force – and the associated physical constants  $G$  and  $\epsilon_0$ .

So it is important to soak up as many irrelevant physical constants into your variables as possible *prior* to putting the problem on a computer. In the case of Newton's second law, you have objects with dimensions of length, time, and force (mass-times-length-over-time-squared). What you ideally would like to do is identify natural length, time, and force scales in the problem and re-define dimensionless quantities in terms of them, so that, for example:  $\mathbf{r} = r_0 \mathbf{q}$  with  $r_0$  a length, and  $\mathbf{q}$  dimensionless,  $t = t_0 s$  for dimensionless  $s$  and fixed  $t_0$ , and perhaps  $U = U_0 V$  for dimensionless  $V$  and a natural energy scale  $U_0$ .

As a concrete example of this process, take two particles in one dimension that interact gravitationally: they start from an initial separation  $r_0$  at rest, and move under the influence of the potential  $U = -\frac{G m_1 m_2}{r_{12}}$  for  $G = 6.67 \times 10^{-11}$  N m<sup>2</sup>/kg<sup>2</sup>. There is a natural length,  $r_0$ , the initial separation, and a natural energy scale,  $U_0 \equiv \frac{G m_1 m_2}{r_0}$ , the total energy of the system, at that separation. Then we could write:

$$\frac{d^2}{ds^2} q(s) = -\frac{G m_1 m_2 t_0}{r_0 q_{12}^2 r_0} = -\frac{U_0 t_0^2}{q_{12}^2 r_0}, \quad (2.15)$$

and require that  $t_0 = \sqrt{\frac{r_0}{U_0}}$ , then our equation of motion becomes:

$$q''(s) = -\frac{1}{q_{12}^2} \quad (2.16)$$

and we have a procedure for re-introducing the dimensions as needed (at the end). Notice that all quantities are now “of order one”, since  $q_{12}$  is initially just 1, and therefore  $q''(s)$ , the dimensionless acceleration, is also of order 1 at  $s = 0$ . We want a concrete, dimensionless way to talk about small numbers, and in this case, we know that  $q(s) \ll 1$  is small, and  $q(s) \gg 1$

is large. If we did not form these ratios from the start, our work would be plagued by peanut gallery questions of the form: “Is one meter large or small?”.

## 2.3 The Verlet Method

Without further adieu, we go through the steps required to generate our first numerical method. We’ll do this in the concrete context of Newton’s second law, but the expressions end up being relevant to any second order ODE, so keep your eye out for the generalization.

### 2.3.1 Taylor Expansion

The key observation is that, in the vicinity of  $t$ , the value of  $x(t + \Delta t)$  (for small  $\Delta t^2$ ) can be estimated to arbitrary accuracy by Taylor expansion:

$$x(t + \Delta t) = x(t) + \dot{x}(t) \Delta t + \frac{1}{2} \ddot{x}(t) \Delta t^2 + \frac{1}{6} \dddot{x}(t) \Delta t^3 + \frac{1}{24} \dots + \dots, \quad (2.17)$$

and similarly for  $x(t - \Delta t)$  (just swap the sign of all odd  $\Delta t$  terms):

$$x(t - \Delta t) = x(t) - \dot{x}(t) \Delta t + \frac{1}{2} \ddot{x}(t) \Delta t^2 - \frac{1}{6} \dddot{x}(t) \Delta t^3 + \frac{1}{24} \dots + \dots, \quad (2.18)$$

and then adding these two together gives:

$$x(t + \Delta t) + x(t - \Delta t) = 2x(t) + \ddot{x}(t) \Delta t^2 + \frac{1}{12} \dddot{x}(t) \Delta t^4. \quad (2.19)$$

From Newton’s second law, we know that  $\ddot{x}(t)$  is given, that’s just  $F(t)/m$ , so that we could write this equation (once a force has been provided) as

$$\boxed{x(t + \Delta t) = 2x(t) - x(t - \Delta t) + \frac{F(t)}{m} \Delta t^2 + O(\Delta t^4)}. \quad (2.20)$$

Here, we use  $O(\Delta t^4)$  to indicate that the next order corrective term scales like  $\Delta t^4$ . How might we use this observation? Well, suppose we had a value for position at times  $t$  and  $t - \Delta t$ , and we knew the force at time  $t$ , then we could estimate (very well) the position at time  $t + \Delta t$ . Given a pair of starting positions, we could even chain together successive estimates.

<sup>2</sup>Small compared to what?

### 2.3.2 Grid

Define an equally spaced grid of temporal values,  $t_j = j \Delta t$ , for  $j = 1 \rightarrow N$ ,  $\Delta t$  fixed. Given a function of  $t$ , say  $f(t)$ , we can “project” the function onto our grid by evaluating the function at the gridpoints, and we typically define a special notation:  $f_j \equiv f(t_j)$  for this process. So  $f(t)$  is a continuous function, and  $\{f_j\}_{j=1}^N$  the values of that function on the temporal grid. What we’re going to do is use (2.20) to take a pair of given values, for  $x_0$  and  $x_{-1}$  (the value of position at times 0 and  $-\Delta t$ , respectively), and generate  $x_1$  from them – then with  $x_1$  and  $x_0$ , we’ll generate  $x_2$ , and so on, out to some given ending time. We can write (2.20), applied to a function on the grid, as:

$$x_{j+1} = 2x_j - x_{j-1} + \frac{F_j}{m} \Delta t^2. \quad (2.21)$$

We make an error in this approximation, since we are not including the additional  $\Delta t^4$  term. But it’s clear that given two starting values, we can generate a string of approximations to  $x(t)$  at the temporal gridpoints.

### 2.3.3 Method

There is a slightly subtle point, and I don’t want to make too much of it beyond mentioning it – given some sort of starting values and a force, we can march a sequence of  $x_j$  forward in “time” (i.e. on the temporal grid), and while we will refer to that solution as the set  $\{x_j\}_{j=1}^N$ , how do these values compare to the projection of the *true* solution (which we don’t even have, in most cases), which is also denoted  $\{x_j\}_{j=1}^N$  – this is a subtle and tricky question, requiring careful analysis on a case-by-case basis – its answer depends both on the force, and the type of numerical method we have in mind. As I say, though, it is a problem for later on.

Now, given two “initial values”,  $x_{-1}$  and  $x_0$ , and a continuous function, the force,  $F(t)$  (and remember,  $F(t)$  is shorthand for  $F(x, t)$ ), we have the following algorithm:

$$\begin{aligned} &\text{for } j = 0, j \leq N - 1, j = j + 1 \\ &\quad x_{j+1} = 2x_j - x_{j-1} + \frac{F_j}{m} \Delta t^2 \\ &\text{end} \end{aligned} \quad (2.22)$$

This will do it, as long as we understand  $F_j \equiv F(x_j, t_j)$ , so that we evaluate the force at a known value of  $x$  (remember, coming into each step of the loop, we have access to all position approximations up to and including  $x_j$ ).

The only issue to work out is the initial conditions – the most typical physical specification is  $x(0)$  and  $v(0)$ , the initial position and velocity. How do we turn this into values for  $x(-\Delta t)$  and  $x(0)$ ? The easy one is  $x_0 = x(0)$ , and for  $v(0)$ , think of a grid approximation to  $v(0)$ :

$$v_0 = \frac{x_0 - x_{-1}}{\Delta t} \longrightarrow x_{-1} = x_0 - \Delta t v_0. \quad (2.23)$$

This setup will work, and we can use the equality on the left to generate values of  $v_j = \frac{x_j - x_{j-1}}{\Delta t}$  if our force depends on the velocity at time level  $j$ . The initial condition is sub-optimal, since we are using a less accurate approximation to the velocity than our approximation to the acceleration (see (2.19)), but again, pragmatism is our guide – after all, we only make this inaccurate approximation once<sup>3</sup>.

### 2.3.4 Testing

We have a method, and we will check it against the exact solutions that we can generate<sup>4</sup>. But what should we do for numerical solutions that have no corresponding  $x(t)$ ? That will, presumably, be the primary utility of the Verlet integrator – what guiding principles can we use to ensure that the method works, and the results are physically relevant?

For conservative systems, forces derivable from a potential, one good check is energy conservation. We construct, at each time level, the energy:

$$E_j = \frac{1}{2} m \left[ \frac{x_{j+1} - x_{j-1}}{2 \Delta t} \right]^2 + U(x_j), \quad (2.24)$$

and we can track this quantity (which should be constant). Note that we are using a more accurate approximation to the speed in this expression, that's to keep the energy value itself accurate. The gross behavior of this quantity should be relatively constant, at the very least, bounded. If it is not, that could be an indication that, for example, the time step is too large.

---

<sup>3</sup>But don't say I didn't warn you. It is possible to make a more accurate initial approximation, I'll leave that to you.

<sup>4</sup>To ensure that it is correctly implemented.

**Lab**

In this lab, we will solve a few different ODEs numerically – the first problem ensures that your solver is working correctly – check with the lab instructor early to verify that the solver is correct. Use the spaces provided to answer any questions, and make any relevant observations.

**Problem 2.1**

For the simple oscillator, take  $\omega = 2\pi$  1/s, and solve  $\ddot{x} = -\omega^2 x$  with  $x(0) = 1$ ,  $\dot{x}(0) = 0$ . Use a time step of  $\Delta t = 1/100$  s, and plot your result together with the correct answer for  $t = 0 \rightarrow 1$  s (have your instructor check this plot). Using  $m = 1$  kg, calculate the energy at each point, and compute the following measure of error:

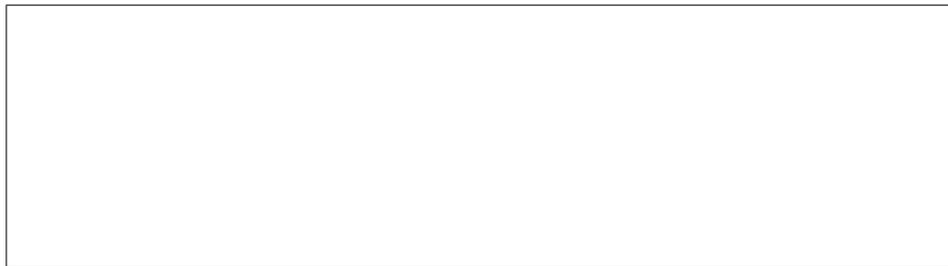
$$\Delta E = \frac{\max(E_j) - \min(E_j)}{E} \quad (2.25)$$

where  $\max/\min(E_j)$  refers to the maximum/minimum numerical value of the energy (over a 1 s interval), and  $E$  is the theoretical energy (compute it separately, and write its value below)

**Problem 2.2**

For a particle in two dimensions, we need a Verlet solver for  $x$  and  $y$  separately. Write one that takes a force function, and returns a list of the form  $\{\{x_1, y_1\}, \{x_2, y_2\}, \dots, \{x_N, y_N\}\}$ .

a. Solve the pair (2.4) using  $q = 1$  C,  $m = 2$  kg, and  $B(r) = 1$  T. We know that there exists circular motion associated with a constant magnetic field – starting at  $x(0) = 0$ ,  $y(0) = 1$  m,  $\dot{y}(0) = 0$ , find the correct initial component of velocity in the  $x$ -direction to sustain motion at radius 1 m – write your initial condition below. Solve for the motion of the particle using  $\Delta t = 1/100$  s – your solution should be circular (this provides a check of your method).

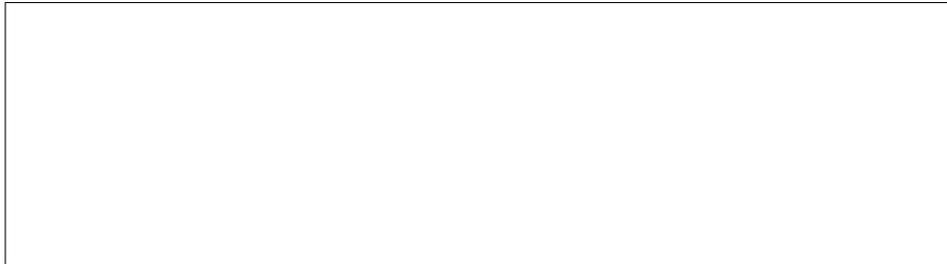


b. Using  $q = 1$  C,  $m = 1$  kg, and  $B(r) = (1 \text{ T}) e^{-r}$  with  $x(0) = 1$  m,  $y(0) = 0$  m,  $\dot{x}(0) = -0.1$  m/s,  $\dot{y}(0) = 0$  m/s, sketch the motion of the test particle for  $\Delta t = 1/100$  s, and  $N = 10000$  steps below (you can use the built-in function `ListLinePlot` to plot the pairs  $\{x, y\}$  directly from the output of your function):



**Problem 2.3**

Using the conversion between  $\{x, y, z\}$  and  $\{r, \theta, \phi\}$ , make a table of particle locations placed on a sphere of radius  $r = 1$  – use a step size of  $\Delta\theta = \pi/20$ , and  $\Delta\phi = 2\pi/10$ . Omit the particles at  $\theta = 0, \pi$ , and take  $\phi = 2\pi$ , but *not* zero (otherwise, you're putting two particles at the same location, and that's a Coulomb's law no-no). Use the provided function `ViewParticles` to ensure that your particles are correctly located on a sphere. How many particles are there?

**Problem 2.4**

The particles in your sphere will interact electrically according to the potential:  $U = \frac{q_1 q_2}{4\pi\epsilon_0 r_{12}}$ . Write a function `Force` that takes as input the list of particles `PList`, and outputs a table of the three-dimensional forces felt by each particle. Using  $\epsilon_0 = 1$ , and  $q = 1$  C for all charges, use your configuration from the previous problem to find the force on particle number 14 (write the vector position of particle 14 as well):



**Problem 2.5**

Put it all together, write a Verlet routine that loops through a list of particles, updating each one's location – use  $\epsilon_0 = 1$ ,  $m = 1$  kg,  $q = 1$  C for all particles, and a time step  $\Delta t = 1/100$ . For output, (call it `output`) give a table of position tables (so that `output[[1]]` returns a table of particle positions for  $t = \Delta t$ ). Use the provided function `ViewParticles` to generate a movie of the motion and get your lab instructor to take a look. Roughly how many time steps does it take for the sphere's radius to double?

