PELEG BAR SAPIR

PHYSICS SIMULATIONS (LECTURE NOTES)

1

Introduction

1.1 Why are Simulations Used?

Text

1.2 Python

1.2.1 General

Text

1.2.2 NumPy

Text

1.3 A Bit About Git

Text

1.4 Some Mathematical Background

1.4.1 Radians

In the context of trigonometry in mathematics and physics, we most commonly use **radians** in place of **degrees**. This sometimes causes some confusion with newer students, so I find it appropriate to briefly discuss the what and why of radians and their usage.

Degrees are defined as a measurement of **angles**. That is, for historical and practical reasons we define a full rotation as 360°. The main problem with using angles is that they are units by themselves, i.e. they differ from lengths. This can cause some consistency issues with regards to units. Consider for example the following equation for a **centripetal force**:

$$F = m\frac{v^2}{r}, \tag{1.1}$$

where m is the mass of the rotationg object, v^2 is the square of the linear velocity the object experiences in each time instance, and r is the distance of the rotating object from the center of rotation (Figure 1.1).

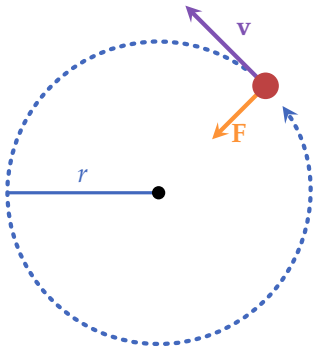

Figure 1.1: Object with mass m in perfect cicular motion with distance r around a central point. The force acting on the object is directed towards the center and has value $F = m\frac{v^2}{r}$.

In the standard SI scheme, F has units of $[\mathrm{N}]$ ("Newtons"), i.e. $\left[\mathrm{kg\,m\,s^{-1}}\right]$. We know that m has units of $[\mathrm{kg}]$, v of $\left[\mathrm{m\,s^{-1}}\right]$ and r units of $[\mathrm{m}]$ - so both sides of the equation have the same units.

However, nothing prevents us from measuring the rotation associated with $\frac{v^2}{r}$ directly using angular velocity ω - which can be defined as the amounth of degrees of rotation per time T if the rotation is uniform:

$$\omega = \frac{\theta}{T}, \tag{1.2}$$

or more generally as the time derivative of the angle,

$$\omega = \frac{\mathrm{d}\theta}{\mathrm{d}t} = \dot{\theta}. \tag{1.3}$$

If we measure the angle θ in degrees, we get that the units of angular velocity is $\left[^\circ\,\mathrm{s^{-1}}\right]$. If we then want to use ω in the centripetal force equation, we must somehow cancel the degrees unit to get the proper units of $[\mathrm{N}]$. This obviously leads to a somewhat cumbersome equation.

A much better approach is to measure the *ratio* between the length of an arc created by our angle at a radius R and the same radius itself (Figure 1.2). This gives a unitless measure which we call *radians*, and can be either ignored in the unit calculation (since it is unitless), or simply denoted $[\mathrm{rad}]$.

Another way to view radians is that they measure an arc length in units of the radius (not necessarily a unit radius). If we scale the radius by any scalar, an arc length of the same angle will scale by exactly the same scalar, and so the amount of radians we measure for the arc length won't change (the ratio stays the same, as both the numerator and denominator are scaled by the same number).

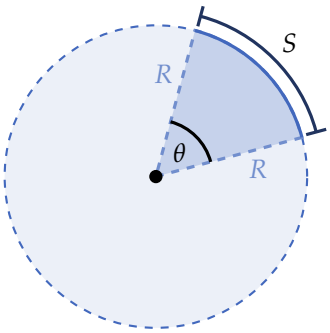

Figure 1.2: Measuring angles using radians. We express the length of S in units of R, e.g. if $S = 1$ then the arc length is equal to R, if $S = \pi$ the arc length is half that of the entire circumference of the circle, etc. The ratio S/R is constant for any real positive value of R, and thus can be used to uniquely describe the angle θ.

We can easily replace any measurement of angles in degrees by a corresponding measurement in radians, since any amount of degrees has a 1-to-1 correspondence with a single value in radians. Let us find this correspondence: since there are 2π radii in a full circle, and a full angle is $360°$, we see that each radii length of an arc corresponds to $\frac{360°}{2\pi} = \frac{180°}{\pi} \approx 57.2958°$.

Now, using the radian unit to measure an angle θ, we get that ω has units of $\left[\text{rad}\,\text{s}^{-1}\right]$, or simply $\left[\text{s}^{-1}\right]$. In turn, if we raise ω to the 2nd power and multiply it by the distance r between the rotating object and center of rotation, and the mass m of the object, we get the units $\left[\text{kg}\,\text{m}\,\text{s}^{-1}\right]$, which is exactly the excplicit form of Newtons. Indeed, using this perspective, Equation 1.1 can be written as

$$F = m\omega^2 r. \tag{1.4}$$

In my mind, this is simply beautiful.

Let's finish with some example correspondences between degrees and radians, which can be seen in Table 1.1 below.

[°]	[rad]
0	0
30	$\frac{\pi}{6}$
45	$\frac{\pi}{4}$
60	$\frac{\pi}{3}$
90	$\frac{\pi}{2}$
120	$\frac{2\pi}{3}$
180	π
270	$\frac{3\pi}{2}$
360	2π

Table 1.1: Example correspondences between measurements in degrees and radians.

1.4.2 *Taylor Series*

For some functions, it is rather easy to calculate their values at some point x_0: for example, given a polynomial function

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n, \tag{1.5}$$

it is rather easy to calculate its value at any real point x: all the operations that we need to use are addition and multiplication of real numbers, and raising real numbers to an integer power (which in principle can be implemented as repeated multiplications). Modern computers calculate such operations at the rate of billions of times a second.

Note 1.1 Floating point representation

For the sake of simplicity, I'm ignoring the entire topic of floating point numbers and relevant questions of precision.

Example 1.1 Calculating a value of a simple polynomial

Given the polynomial $P(x) = 3x^2 - 2x + 7$, we can easily calculate its value at, say, $x = 5$:

$$P(x = 5) = 3 \cdot 5^2 - 2 \cdot 5 + 7 = 3 \cdot 25 - 10 + 7 = 72.$$

For other functions, on the other hand, it is not that easy to calculate their values at most points. For example, consider the exponential function $f(x) = e^x$. We know precisely exactly one real value of the function: at $x = 0$, the function returns 1. But for any other value of x, we don't really know the value of the function. That is, we know all values *symbolicaly* (for example, $f(1) = e$, $f(2) = e^2$, etc.) - but not in explicit form.

However, given that the function behaves nicely enough (we'll discuss in a moment what that means), we can *approximate* its value to whatever percision we wish, using a method called the **Taylor series** of the function. The "price" we pay for greater percision is simply performing more calculations.

The basic idea of a Taylor series of a function $f(x)$ is that we approximate the function by adding higher and higher derivatives of the same function, at some point a for which we know precisely the value of the function and its derivatives to any order. This might sound a bit abstract, so let's explore this process using an example function - again, the exponential $f(x) = e^x$. As mentioned, we only really know one value of the function precisely, namely at $a = 0$: $e^0 = 1$.

We can therefore start approximating e^x simply as e^0. This is obviously a very impercise approximation, but the important thing

is that if we look at a very close neighborhood of $x = 0$, the approximation is not *that* bad: consider, for example $x_0 = 0.0001$. Our approximation gives $e^{x_0} \approx 1$, which is not far from the more precise value $e^{x_0} = 1.000100005$ (the value here is shown up to the ninth decimal). Of course, the closer we get to $a = 0$, the better our approximation gets: for example, with $x_1 = 0.00001$, we get $e^{x_1} \approx 1$ again, where in reality $e^{x_1} = 1.000010000$ (also shown here up to the ninth decimal). On the other hand, if we as we get farther away from $x = 0$, the approximation becomes worse and worse, as seen in Table 1.2 below.

x	e^x (exact)	Δ (error)
0.000	1.000000000	0.000000000
0.001	1.001000500	0.001000500
0.010	1.010050167	0.010050167
0.100	1.105170918	0.105170918
0.500	1.648721271	0.648721271
0.510	1.665291195	0.665291195
0.520	1.682027650	0.682027650
1.000	2.718281828	1.718281828
1.100	3.004166024	2.004166024
1.500	4.481689070	3.481689070
2.000	7.389056099	6.389056099

Table 1.2: Zero order Taylor series approximation of e^x.

Now, let's take this a step further: since the derivative of a function at a point tells us how the functions changes close to the point, we can use this information to improve our approximation by adding the value of the first derivative of e^x at $x = 0$, which is also 1. In fact, this is true for any order derivative of e^x, since $\frac{d^n}{dx^n}e^x = e^x$ for any $n \in \mathbb{N}$. Since the derivative changes with the value of x, we will multiply it by x in the approximation. Thus we get

$$e^x \approx 1 + 1 \cdot x = 1 + x, \tag{1.6}$$

which we call the **first order** approximation of the function e^x. Table 1.3 below shows the same values from Table 1.2 for x, but using the first order approximation for e^x.

By examining Table 1.3 it's clear that $1 + x$ is a much better approximation for e^x than just 1. Of course, we can take a further step: the second derivative of a function at a point tells us how the change in the function itself changes around that point. We can take the second derivative of e^x (which is by itslef e^x), again substitute $x = 0$, get $e^0 = 1$, and use this with x^2 to apperoximate e^x even better. In fact, let's already take this idea to its logical conclusion: we'll simply use the infinite power series in x, i.e. $1, x, x^2, x^3, \ldots, x^n, \ldots$ and

x	$1+x$	e^x (exact)	Δ (error)
0.000	1.000000000	1.000000000	0.000000000
0.001	1.001000000	1.001000500	0.000000500
0.010	1.010000000	1.010050167	0.000050167
0.100	1.100000000	1.105170918	0.005170918
0.500	1.500000000	1.648721271	0.148721271
0.510	1.510000000	1.665291195	0.155291195
0.520	1.520000000	1.682027650	0.162027650
1.000	2.000000000	2.718281828	0.718281828
1.100	2.100000000	3.004166024	0.904166024
1.500	2.500000000	4.481689070	1.981689070
2.000	3.000000000	7.389056099	4.389056099

Table 1.3: First order Taylor series approximation of e^x.

attach to each x^k the value given by $\frac{d^n}{dx^n}e^x\Big|_{x=0} = 1$, so we get

$$e^x \approx 1 + x + x^2 + x^3 + \cdots + x^n + \cdots = \sum_{n=0}^{\infty} x^n. \tag{1.7}$$

This, however doesn't quite work. The coefficients of each term have to be adjusted. By assuming that at the limit where $n \to \infty$ the approximation should be with zero error, we get that for each n,

$$\frac{d^n}{dx^n}e^x = \frac{d^n}{dx^n}\left(1 + x + x^2 + \dots\right) \equiv s(x). \tag{1.8}$$

Calculating the general n-th derivative of Equation 1.7 is rather easy. Let's examine the case for $n = 3$: the first three derivatives of Equation 1.7 are

$$\begin{aligned}
\tfrac{d}{dx}s(x) &= 1 + 2x + 3x^2 + 4x^3 + \dots, \\
\tfrac{d^2}{dx^2}s(x) &= 2 + 6x + 12x^2 + \dots, \\
\tfrac{d^3}{dx^3}s(x) &= 6 + 24x + \dots,
\end{aligned}$$

and if we subtitute $x = 0$ we get... TBW!

The most general polynomial expansion of a smooth function $f(x)$ is an infinite polynomial in x:

$$P(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots \tag{1.9}$$

If $P(x) = f(x)$ then all their derivatives are the equal, i.e.

$$\begin{aligned}
f'(x) &= P'(x) = a_1 + 2a_2x + 3a_3x^2 + 4a_4x^3 + \dots \\
f''(x) &= P''(x) = 2a_2 + (2\cdot 3)a_3x + (4\cdot 3)a_4x^2 + \dots, \\
f'''(x) &= P'''(x) = (2\cdot 3)a_3 + (4\cdot 3\cdot 2)a_4x + \dots, \\
f''''(x) &= P''''(x) = (4\cdot 3\cdot 2)a_4 + \dots, \\
&\vdots \\
f^{(n)}(x) &= P^{(n)}(x) = n!a_n + \dots, \\
&\vdots
\end{aligned} \tag{1.10}$$

Subtituting $x = 0$ into the equality we get

$$\begin{aligned}
f'(0) &= P'(0) = a_1 + \cancel{2a_2x} + \cancel{3a_3x^2} + \cancel{4a_4x^3} + \cancel{\dots} \\
f''(0) &= P''(0) = 2a_2 + \cancel{(2\cdot 3)a_3x} + \cancel{(4\cdot 3)a_4x^2} + \cancel{\dots}, \\
f'''(0) &= P'''(0) = (2\cdot 3)a_3 + \cancel{(4\cdot 3\cdot 2)a_4x} + \cancel{\dots}, \\
f''''(0) &= P''''(0) = (4\cdot 3\cdot 2)a_4 + \cancel{\dots}, \\
&\vdots \\
f^{(n)}(0) &= P^{(n)}(x) = n!a_n + \cancel{\dots}, \\
&\vdots
\end{aligned} \tag{1.11}$$

or more succinctly:

$$a_n = \frac{f^n(x)}{n!}. \tag{1.12}$$

Therefore, the full taylor polynomial for $f(x)$ is

$$\begin{aligned}
P(x) &= f(0) + f'(0)x + \frac{f''(0)}{2}x^2 + \frac{f'''(0)}{3!}x^3 + \frac{f''''(0)}{4!}x^4 + \dots \\
&= \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!}x^n.
\end{aligned} \tag{1.13}$$

If we take only the first N terms in $P(x)$ we get

$$P_N(x) = \sum_{n=0}^{N} \frac{f^{(n)}(x)}{n!}x^n. \tag{1.14}$$

Note 1.2 $(N+1)$-th derivative an N-th partial Taylor sum

Since the N-th partial sum of any taylor series $P(x)$ is

$$P_N(x) = \sum_{n=0}^{N} \frac{f^{(n)}(x)}{n!}x^n,$$

all terms except $\frac{a_N}{N!}$ cancel out by the N-derivative. And at the $(N+1)$-derivative this term cancels out as well, since its a constant. Therefore from its $(N+1)$-th term and on, the partial sum $P_N(x)$ reduces to zero:

$$P^{(k>N)}(x) = 0.$$

!

Since in practice we can't use the full Taylor series to approximate a function at a point, we must intelligently choose how many terms we want to use to get a "good enough" approximation - that is, we want that the differemce between our polynomial approximation is the actual function to be no more than some value Δ:

$$\left|f(a) - P_n(a)\right| \leq \Delta. \tag{1.15}$$

We can construct a **remainder function** which gives the difference between the actual function value and out n-th degree approximation at every point on the interval $I = [0, a]$:

$$R_n(x) = f(x) - P_n(x). \tag{1.16}$$

Thanks to Lagrange and other great mathematicians, the following always holds for the remainder function: if the the $n+1$-th derivative of the original function is bounded on the open interval $(0, a)$, i.e.

$$\left|f^{(n+1)}(x)\right| \leq M, \tag{1.17}$$

then there's a finite value M such that

$$\left|R_n(x)\right| \leq \left|\frac{Mx^{n+1}}{(n+1)!}\right|. \tag{1.18}$$

This is called the **Lagrange error bound**. No proof is provided for it in this course (TBW: add reference to one?).

Example 1.2 Lagrange error bound

TBW

1.4.3 *Projections, rejections and the dot product*

Every vector $\mathbf{a} \in \mathbb{R}^n$ can be decomposed into two components: one in the direction of another vector $\mathbf{b}$, and one orthogonal to $\mathbf{b}$. These components are called, respectively, the **projection** of $\mathbf{a}$ onto $\mathbf{b}$ and the **rejection** of $\mathbf{a}$ from $\mathbf{b}$. In $\mathbb{R}^3$ the same procedure can be applied to projecting a vector onto- and rejecting it from a plane, since every plane in $\mathbb{R}^3$ has a single normal vector up to a sign (Figure 1.3).

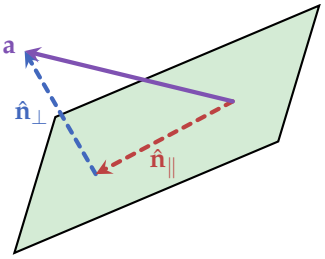

Figure 1.3: Decomposing the vector $\mathbf{a}$ into two components in respect to a plane: one parallel to the plane ($\mathbf{a}_{\parallel}$) and one orthogonal to it ($\mathbf{a}_{\perp}$). These are also called the projection and rejection of $\mathbf{a}$ on the plane, respectively.

The projection of a vector onto another vector gives rise to an important operation between two vectors: the **dot product**: given a

vector $\mathbf{a}$, its projection on the vector $\mathbf{b}$ is

$$\text{proj}_{\mathbf{b}}\mathbf{a} = |\mathbf{a}|\cos(\theta), \tag{1.19}$$

where θ is the angle between the vectors. See Figure 1.4 for a visual representation. We then define the dot product between the two vectors as

$$\langle \mathbf{a}, \mathbf{b} \rangle = |\mathbf{b}|\,\text{proj}_{\mathbf{b}}\mathbf{a} = |\mathbf{b}|\,|\mathbf{a}|\cos(\theta). \tag{1.20}$$

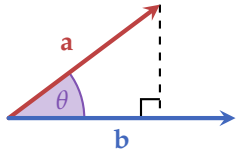

Figure 1.4: Projection of $\mathbf{a}$ onto $\mathbf{b}$: no matter how many dimensions we use, we can always rotate our view such that we look at the plane spanned by both vectors, and $\mathbf{b}$ lies horizontally. In this way it's easy to see why the projection of $\mathbf{a}$ onto $\mathbf{b}$ is $|\mathbf{a}|\cos(\theta)$: it's simply the definitions of the cosine function ("side next to the angle divided by the hypotenuse").

It is of course convenient to have a way to calculate the dot product component-wise. To find such form, we use the same two vectors $\mathbf{a}$ and $\mathbf{b}$ from before, and their difference $\mathbf{c} = \mathbf{a} - \mathbf{b}$ (Figure 1.5).

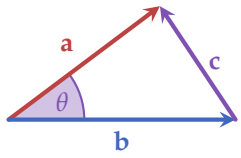

Figure 1.5: Text text text

Thanks to trigonometry we know that the following relation holds ("the law of cosines"):

$$\begin{aligned} |\mathbf{c}|^2 &= |\mathbf{a}|^2 + |\mathbf{b}|^2 - 2\,|\mathbf{a}|\,|\mathbf{b}|\cos(\theta) \\ &= |\mathbf{a}|^2 + |\mathbf{b}|^2 - 2\langle \mathbf{a}, \mathbf{b} \rangle, \end{aligned} \tag{1.21}$$

which we can rearrange into

$$\langle \mathbf{a}, \mathbf{b} \rangle = \frac{1}{2}\left(|\mathbf{a}|^2 + |\mathbf{b}|^2 - |\mathbf{c}|^2 \right). \tag{1.22}$$

In $\mathbb{R}^2$ the component version of the vectors are

$$\mathbf{a} = \begin{bmatrix} a_x \\ a_y \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_x \\ b_y \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} a_x - b_x \\ a_y - b_y \end{bmatrix}, \tag{1.23}$$

and Equation 1.22 becomes

$$\begin{aligned} \langle \mathbf{a}, \mathbf{b} \rangle &= \frac{1}{2}\left(a_x^2 + a_y^2 + b_x^2 + b_y^2 - (a_x - b_x)^2 - \left(a_y - b_y\right)^2 \right) \\ &= \frac{1}{2}\left(\cancel{a_x^2} + \cancel{a_y^2} + \cancel{b_x^2} + \cancel{b_y^2} - \cancel{a_x^2} + 2a_xb_x - \cancel{b_x^2} - \cancel{a_y^2} + 2a_yb_y + \cancel{b_y^2} \right) \\ &= a_xb_x + a_yb_y. \end{aligned} \tag{1.24}$$

In general, we can apply the same procedure in $\mathbb{R}^n$: the components of the vectors are

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} a_1 - b_1 \\ a_2 - b_2 \\ \vdots \\ a_n - b_n \end{bmatrix}, \tag{1.25}$$

and Equation 1.22 becomes

$$\begin{aligned}\langle \mathbf{a}, \mathbf{b} \rangle &= \frac{1}{2}\Big(a_1^2 + a_2^2 + \cdots + a_n^2 + b_1^2 + b_2^2 + \cdots + b_n^2 - (a_1 - b_1)^2 \\ &\quad - (a_2 - b_2)^2 - \cdots - (a_n - b_n)^2\Big) \\ &= \frac{1}{2}\Big(\cancel{a_1^2} + \cancel{a_2^2} + \cdots + \cancel{a_n^2} + \cancel{b_1^2} + \cancel{b_2^2} + \cdots + \cancel{b_n^2} - \cancel{a_1^2} + \cancel{2}a_1 b_1 - \\ &\quad \cancel{b_1^2} - \cancel{a_2^2} + \cancel{2}a_2 b_2 + \cancel{b_2^2} + \cdots + \cancel{a_n^2} + \cancel{2}a_n b_n + \cancel{b_n^2}\Big) \\ &= a_1 b_1 + a_2 b_2 + \cdots + a_n b_n \\ &= \sum_{i=1}^{n} a_i b_i.\end{aligned} \tag{1.26}$$

Let us now explore some key properties of the dot product (the reader is encouraged to prove items 2 and 3):

1. Since the product of real numbers is commutative (i.e. $ab = ba$), the dot product is also commutative: if we exchange v_i and w_i in Equation 1.26 the result stays the same.

2. The dot product is distributive over vector addition: $\langle \mathbf{a}, \mathbf{b} + \mathbf{c} \rangle = \langle \mathbf{a}, \mathbf{b} \rangle + \langle \mathbf{a}, \mathbf{c} \rangle$.

3. We can take out scalars from within the product: given the scalars ϵ_1 and ϵ_2, $\langle \epsilon_1 \mathbf{a}, \epsilon_2 \mathbf{b} \rangle = \epsilon_1 \epsilon_2 \langle \mathbf{a}, \mathbf{b} \rangle$.

4. The angle between two non-zero orthogonal vectors $\mathbf{a}$ and $\mathbf{b}$ is $\theta = \frac{\pi}{2}$, and therefore $\langle \mathbf{a}, \mathbf{b} \rangle = |\mathbf{a}|\,|\mathbf{b}| \cos\left(\frac{\pi}{2}\right) = 0$ (since $\cos\left(\frac{\pi}{2}\right) = 0$). This works in the other way around as well: if the dot product of two non-zero vectors is zero, then they are orthogonal.

5. We can't cancel vectors in the dot product the same we do with real numbers: consider three vectors $\mathbf{a} \neq \mathbf{0}$, $\mathbf{b}$, $\mathbf{c}$ such that $\langle \mathbf{a}, \mathbf{b} \rangle = \langle \mathbf{a}, \mathbf{c} \rangle$. We can redistribute the equation to give $\langle \mathbf{a}, \mathbf{b} - \mathbf{c} \rangle = 0$, which means that $\mathbf{a}$ and $\mathbf{b} - \mathbf{c}$ are orthogonal, and doesn't imply that $\mathbf{b} - \mathbf{c} = \mathbf{0}$ (i.e. $\mathbf{b}$ and $\mathbf{c}$ might not be equal).

2

Mechanics

2.1 *Preface*

Text text text

2.2 *Pendulum*

2.2.1 *Simple pendulum*

Text text text.

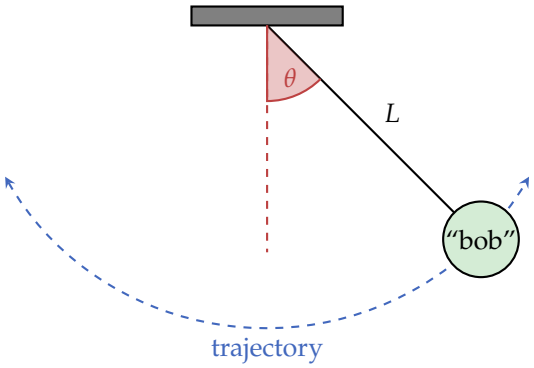

Figure 2.1: A simple pendulum. TBW: add more info.

Using force analysis we can derive an equation of motion for the bob (see Figure 2.2): since the rod can't change its length (it's always L), the only variable quantity is the angle θ, and the bob's trajectory is a circle. Any force acting in a radial direction to the trajectory must be counter-balanced (otherwise there will be some acceleration - and therefore motion - in that direction). We are therefore left with only a tangental force, with magnitude

$$F = -mg\sin(\theta), \tag{2.1}$$

the minus sign here is chosen to represent that gravity acts in the negative y direction (i.e. "down").

Applying Newton's second law we get that

$$F = ma = -mg\sin(\theta), \tag{2.2}$$

i.e.

$$a = -g\sin(\theta). \tag{2.3}$$

Now we see that the minus sign also makes sense physically, as it shows that the acceleration is always in the opposite direction to the angle (which is negative to the left and positive to the right).

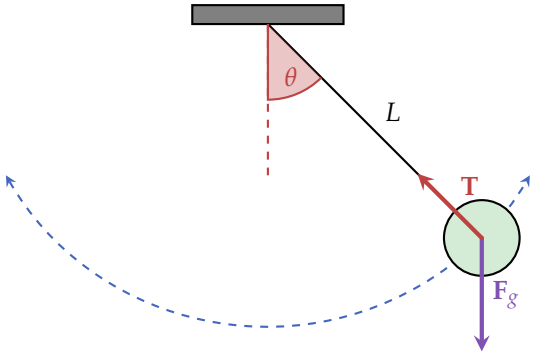

Figure 2.2: Forces acting on a simple pendulum. TBW: Force compnents.

The tangental position s of the bob can be calculated from the angle θ by

$$s = L\theta \tag{2.4}$$

(recall that θ is given in radians), and therefore the tangental velocity is

$$v = \dot{s} = L\dot{\theta}, \tag{2.5}$$

and the acceleration is therefore

$$a = \dot{v} = \ddot{s} = L\ddot{\theta}. \tag{2.6}$$

Since we know that $a = -g\sin(\theta)$, we get

$$L\ddot{\theta} = -g\sin(\theta), \tag{2.7}$$

and by moving the rhs term to the left and divide by l we get

$$\ddot{\theta} + \frac{g}{L}\sin(\theta) = 0. \tag{2.8}$$

This is a differential equation without analytical solution. We will therefore take two approaches: (1) use an approximation to yield an analytical solution, and (2) solve the equation numerically.

2.2.2 *Small-angle approximation*

The Taylor series expansion of $\sin(x)$ around $x = 0$ is

$$\sin(x) = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \ldots \tag{2.9}$$

We can therefore approximate $\sin(x)$ as x for small values of x:

$$\sin(x) \approx x. \tag{2.10}$$

This is known as the "small-angle approximation" of the sine function. By using this approximation we get that the (analytically) unsolvable Equation 2.8 reduces to

$$\ddot{\theta} + \frac{g}{L}\theta = 0, \tag{2.11}$$

for which we have an exact solution:

$$\theta(t) = A\cos\left(\omega t + \phi\right), \tag{2.12}$$

where $\omega = \sqrt{\frac{g}{L}}$. The parameters A and ϕ depend on the initial conditions (i.e. angle and tangental velocity).

TBW: resonance freq, how it looks like (+phase space), python plot?

2.2.3 *Numerical integration*

Another approach to solve Equation 2.8 is doing so numerically, i.e. essentially running a computer simulation. While computers can carry out many calculations per second (in the order of billions, in fact) - they are limited to performing descrete calculations. That is to say, a numerical calculation is also an approximation. However, unlike the small-angle approximation, in principle we can improve the approximation indefinitely, although in practice this is of course impossible.

Let us use a rather naive approach to numerically approximating Equation 2.8: instead of viewing θ as a continuous function of time t, we instead define t to only have equally spaced discrete values $t_0, t_1, t_2, \ldots$ such that

$$t_n = t_0 + n\Delta t, \tag{2.13}$$

or phrased differently: we look at the values t_i of t at intervals Δt starting from t_0 (see Figure 2.3).

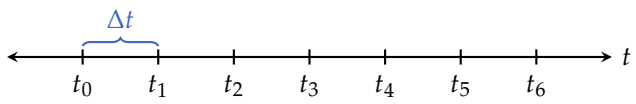

Figure 2.3: Discrete time steps.

In this discrete view, a function of t can also only be evaulated at discrete points. For example, the angle θ of the bob as a function of time is a discrete function taking the values $\theta_0, \theta_1, \theta_2, \ldots$, where each θ_i corresponds to the time t_i.

How does the discrete angular velocity look like in this scheme? It's worth looking at the definition of angular velocity in the continuous case:

$$\omega(t) = \dot{\theta}(t) = \frac{\mathrm{d}\theta(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{\theta\left(t + \Delta t\right) - \theta(t)}{\Delta t}. \tag{2.14}$$

To descritize, we can replace $\theta(t)$ with $\theta_{t_i} = \theta_i$, and therefore $\theta(t+\Delta t)$ with $\theta_{t_i+\Delta t} = \theta_{i+1}$. The limit $\lim_{\Delta t \to 0}$ is pretty meaningless, since the smallest Δt possible in the discrete case is the Δt we chose to discretize out time steps. Therefore, we get that the discrete version of $\omega(t)$ is

$$\omega_i = \frac{\theta_{i+1} - \theta_i}{\Delta t}, \tag{2.15}$$

which is nothing more than saying that the angular velocity at time t_i is simply the difference between the angle at time t_{i+1} and the angle at time t_i, divided by the time difference Δt.

Equation 2.15 gives us a powerful tool: if we only know the current angle θ_i of the bob and its current velocity ω_i, then its next position θ_{i+1} is a simple rearrangement of the equation:

$$\theta_{i+1} = \theta_i + \omega_i \Delta t. \tag{2.16}$$

The same logic can be applied to the angular acceleration $\alpha(t)$: in the continuous version its given by

$$\alpha(t) = \dot{\omega(t)} = \frac{\mathrm{d}\omega(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{\omega(t+\Delta t) - \omega(t)}{\Delta t}, \tag{2.17}$$

and thus the discrete version is

$$\alpha_i = \frac{\omega_{i+1} - \omega_i}{\Delta t}, \tag{2.18}$$

and recovering the angular velocity ω_{i+1} from the angular velocity ω_i and the angular acceleration α_i is done by

$$\omega_{i+1} = \omega_i + \alpha_i \Delta t. \tag{2.19}$$

Since we use the inverse of the differentiation operation (in the discrete sense) to recover the quantity we're after, this scheme is known as a **numerical integration**. Specifically, this kind of numerical integration is called the **Forward-Euler** method, and it is generally considered unfavourable due to the fast rate with which it gains errors, and is rarely used in practice.

However, for the sake of simplicity of writing our first simulations, we will not discuss these issues now, nor will we generalize the method and present better ones - both of which we will do later in the course. Instead, for now we will continue with using this method to devise a numerical integration scheme for a simple pendulum.

In the case of the pendulum, recall that the acceleration at time t is $\ddot{\theta}(t) = -\frac{g}{L}\sin(\theta)$ (Equation 2.8), and therefore we can discretize it as

$$\alpha_{i+1} = -\frac{g}{L}\sin(\theta_i). \tag{2.20}$$

We then use the forward Euler method to get the angular velocity and angle, as seen in Equation 2.19 and Equation 2.16, respectively.

2.2.4 *Damped oscillation*

A slightly more realistic model of a pendulum also considers the way the pendulum loses energy over time (e.g. via friction). This can be modelled by adding a force which resists the angular velocity (see also Figure 2.4):

$$F_d = -bL\dot{\theta}, \tag{2.21}$$

where b is simply a parameter which adjusts how strong the damping is.

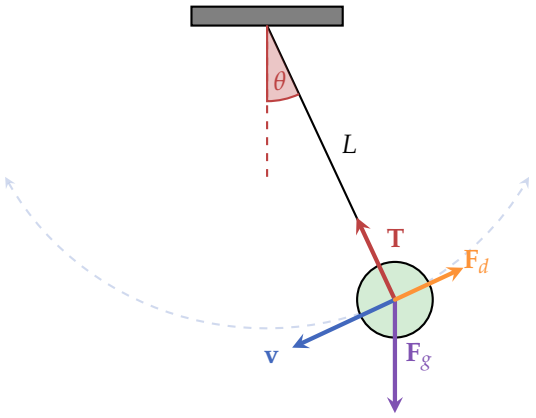

Figure 2.4: Forces on a pendulum including a damping force.

Using Newton's second law, recalling that $a = L\ddot{\theta}$ (Equation 2.6), we get

$$mL\ddot{\theta} = F_g + F_d = -mg\sin(\theta) - bL\dot{\theta}. \tag{2.22}$$

It's common to use $\beta = \frac{b}{2m}$ and of course $\omega_0 = \sqrt{\frac{g}{L}}$, which together with some rearrangement gives us

$$\ddot{\theta} = -2\beta\dot{\theta} - \omega_0^2\sin(\theta). \tag{2.23}$$

2.2.5 *Double pendulum*

An interesting system arises when we take a simple pendulum and attach another simple pendulum to it (as seen in Figure 2.5): unlike a simple pendulum, this system exhibits very complex dynamics which is sensitive to the initial conditions - i.e. it is **chaotic**, meaning that small changes in the initial conditions evolve into big differences in the dynamics of the system.

To analyze the behaviour of the system we again seek to find a differntial equation which will describe the system completely. Unfortunately, this differential equation is rather complicated and is composed of two coupled differential equations. We will only present the equation here but not how to derive it - this will be added in an appendix.

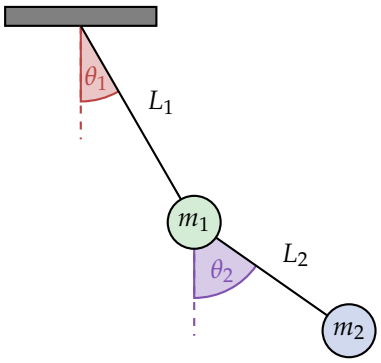

Figure 2.5: A double pendulum. TBW: add more info.

The differential equations governing the dynamics of the system are:

$$0 = (m_1 + m_2)\, L_1 \ddot{\theta}_1 + m_2 L_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 L_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + (m_1 + m_2)\, g \sin(\theta_1), \tag{2.24}$$

and

$$0 = L_2 \ddot{\theta}_2 + L_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - L_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + g \sin(\theta_2). \tag{2.25}$$

To help us get excplicit expressions for $\ddot{\theta}_1$ and $\ddot{\theta}_2$ we can define the following quantities:

$$\alpha = \frac{m_2}{m_1}, \quad \beta = \frac{L_2}{L_1}, \quad \gamma = \frac{g}{L_1}, \quad \Delta\theta = \theta_1 - \theta_2. \tag{2.26}$$

we then get the following "beautiful" expressions for $\ddot{\theta}_1$ and $\ddot{\theta}_2$:

$$\ddot{\theta}_1 = -\frac{(1+\alpha)\,\gamma \sin(\theta_1) + \alpha\beta\dot{\theta}_2^2 \sin(\Delta\theta) + \alpha \cos(\Delta\theta)\left[\dot{\theta}_1^2 \sin(\Delta\theta) - \gamma \sin(\theta_2)\right]}{1 + \alpha \sin^2(\Delta\theta)},$$

$$\ddot{\theta}_2 = \frac{(1+\alpha)\left(\dot{\theta}_1^2 \sin(\Delta\theta) - \gamma \sin(\theta_2) + \cos(\Delta\theta)\left[(1+\alpha)\,\gamma \sin(\theta_1) + \alpha\beta\dot{\theta}_2^2 \sin(\Delta\theta)\right]\right)}{\beta\left[1 + \alpha \sin^2(\Delta\theta)\right]} \tag{2.27}$$

Discretizing Equation 2.27 can then be done as following: first,

1. α, β and γ are defined as in Equation 2.26.

2. Set $\Delta\theta_k = \theta_{1,k} - \theta_{2,k}$.

3. For compactness, we set $N_k = 1 + \alpha \sin^2(\Delta\theta_k)$, $\mathrm{S}(x) = \sin(x)$, $\mathrm{C}(x) = \cos(x)$ and $A = 1 + \alpha$.

Then, we get

$$\ddot{\theta}_{1,i} = -\frac{A\gamma \mathrm{S}(\theta_{1,i-1}) + \alpha\beta\dot{\theta}^2_{2,i-1}\mathrm{S}\left(\Delta\theta_{i-1}\right) + \alpha\mathrm{C}\left(\Delta\theta_{i-1}\right)\left[\dot{\theta}^2_{1,i-1}\mathrm{S}\left(\Delta\theta_{i-1}\right) - \gamma\mathrm{S}\left(\theta_{2,i-1}\right)\right]}{N_{i-1}},$$
$$\ddot{\theta}_{2,i} = \frac{A\left(\dot{\theta}^2_{1,i-1}\mathrm{S}\left(\Delta\theta_{i-1}\right) - \gamma\mathrm{S}\left(\theta_{2,i-1}\right) + \mathrm{C}\left(\Delta\theta_{i-1}\right)\left[A\gamma\mathrm{S}\left(\theta_{1,i-1}\right) + \alpha\beta\dot{\theta}^2_{2,i-1}\mathrm{S}\left(\Delta\theta_{i-1}\right)\right]\right)}{\beta N_{i-1}}. \tag{2.28}$$

(lovely, isn't it?)

3

Thermodynamics

3.1 Preface

Text text text

3.2 Ideal gas

3.2.1 Theory

There are several commonly used models for the behaviour of gasses. A very simple yet powerful one is the **ideal gas** model: it describes gas particles as being perfect spheres which move around in an enclosed container and undergo elastic collisions with other particles and the walls of the container. In common conditions such as atmospheric pressure and temperatures around 300 [K], gasses such as helium (He), argon (Ar), nitrogen (N_2), oxygen (O_2) and carbon dioxide (CO_2) behave like ideal gasses (assuming no chemical reations take place). However, the ideal gas model fails under high pressures, low temperatures, chemical interactions and some physical processes such as adsorption or multipolar interactions.

The principle equation describing an ideal gas is the **ideal gas law**:

$$PV = nRT, \tag{3.1}$$

where (SI units in parentheses):

- P is the pressure of the gas (Pa),
- V is the volume of the container (m^3),
- n is the amount of gas (mol),
- R is the **gas constant**, $R = 8.314 \left[\mathrm{J\,K^{-1}\,mol^{-1}}\right]$, and

- T is the temperature of the gas (K).

Maxwell-Boltzman distribution

Mean free path

Temperature from energy

3.2.2 *Simulating an ideal gas using perfectly elastic spheres*

Text text text

Sphere-wall collision

An elastic collision between a particle and a wall causes the particle's velocity to flip in the direction of the wall's normal (Figure 3.1).

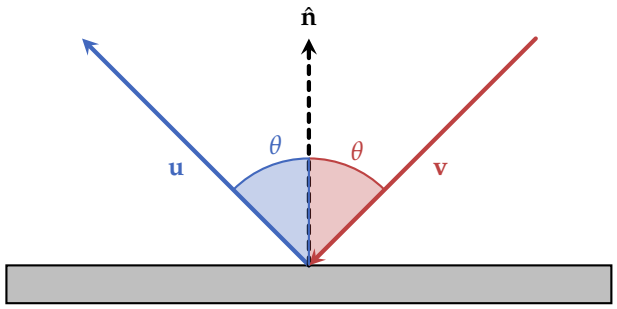

Figure 3.1: Collision of a particle and a wall. The particle bounces in such a way that the component of its velocity $\mathbf{v}$ in the direction of the wall's normal $\hat{\mathbf{n}}$ is flipped. The resulting velocity $\mathbf{u}$ has the same angle to $\hat{\mathbf{n}}$ as $\mathbf{v}$ does.

The component of $\mathbf{v}$ in the direction of $\hat{\mathbf{n}}$ is

$$\mathbf{v}_{\parallel} = \langle \mathbf{v}, \hat{\mathbf{n}} \rangle \hat{\mathbf{n}}. \tag{3.2}$$

Therefore, the component of $\mathbf{v}$ orthogonal to $\hat{\mathbf{n}}$ is

$$\mathbf{v}_{\perp} = \mathbf{v} - \mathbf{v}_{\parallel} = \mathbf{v} - \langle \mathbf{v}, \hat{\mathbf{n}} \rangle \hat{\mathbf{n}}. \tag{3.3}$$

In the case of $\mathbf{u}$, the orthogonal component is the same as that of $\mathbf{v}$, but the parallel component is inverted:

$$\begin{aligned} \mathbf{u}_{\perp} &= \mathbf{v}_{\perp}, \\ \mathbf{u}_{\parallel} &= -\mathbf{v}_{\parallel}. \end{aligned} \tag{3.4}$$

and altogether we get

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_{\perp} + \mathbf{u}_{\parallel} \\ &= \mathbf{v} - \langle \mathbf{v}, \hat{\mathbf{n}} \rangle \hat{\mathbf{n}} - \langle \mathbf{v}, \hat{\mathbf{n}} \rangle \hat{\mathbf{n}} \\ &= \mathbf{v} - 2\langle \mathbf{v}, \hat{\mathbf{n}} \rangle \hat{\mathbf{n}}. \end{aligned} \tag{3.5}$$

Example 3.1 A simple sanity check

To make a simple validation of this equation, let's examine the case for a collision with a wall which is oriented in the xy-plane (i.e. its normal points in the z-direction): given the

continuing on the next page

continuing from the previous page

velocity $\mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$, Equation 3.5 becomes

$$\begin{aligned}
\mathbf{u} &= \mathbf{v} - 2\langle \mathbf{v}, \hat{\mathbf{n}} \rangle \hat{\mathbf{n}} \\
&= \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} - 2\left\langle \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\rangle \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} - 2(\cancel{v_x \cdot 0} + \cancel{v_y \cdot 0} + v_z \cdot 1) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} - 2v_z \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} - 2 \begin{bmatrix} 0 \\ 0 \\ v_z \end{bmatrix} \\
&= \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -2v_z \end{bmatrix} \\
&= \begin{bmatrix} v_x \\ v_y \\ -v_z \end{bmatrix},
\end{aligned}$$

as expected.

Note 3.1 Studying advice

The reader is encouraged to repeat the above calculation for the cases of walls oriented in the xz- and yz-planes.

When a sphere collides with a wall, the distance d from the center of the sphere to the wall is exactly r, the radius of the sphere. The distance between a point $\boldsymbol{P}$ and a plane $\boldsymbol{\pi}$ can be defined as follows: start from $\boldsymbol{P}$ and move along the direction $\pm\hat{\mathbf{n}}$ (where $\hat{\mathbf{n}}$ is the normal to the plane) until the plane is intersected at a point $\boldsymbol{B}$. The distance between $\boldsymbol{P}$ and $\boldsymbol{\pi}$ is then the length $\boldsymbol{PB}$ (Figure 3.2).

If we know a point $\boldsymbol{A}$ on the plane, then we can the vector $\mathbf{c}$ as starting at $\boldsymbol{A}$ and ending at $\boldsymbol{P}$ (i.e. corresponding to the line segment $\overline{\boldsymbol{AP}}$). The projection of $\mathbf{c}$ onto $\hat{\mathbf{n}}$ then gives the length $\boldsymbol{PM}$ (Figure 3.3).

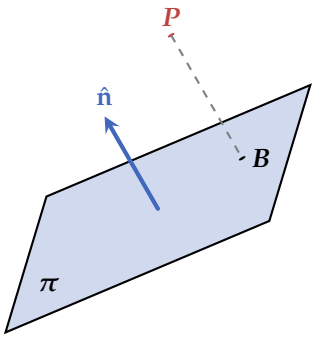

Figure 3.2: The distance between a point P and a plane π. The distance from P to the plane π is the length PB.

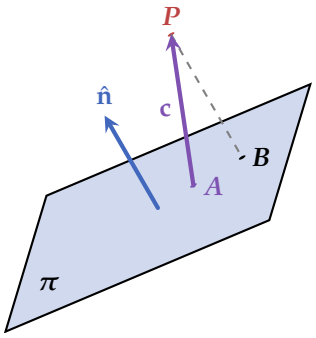

Figure 3.3: The vector $\mathbf{c}$ connects a known point on the plane A and the point P. Its projection on $\hat{\mathbf{n}}$ gives the distance PB.

Since $\hat{\mathbf{n}}$ is a unit vector, the projection of $\mathbf{c}$ onto it is simply the dot product between the two vectors, i.e. the distance d of point P and the plane π is

$$d = \langle \mathbf{c}, \hat{\mathbf{n}} \rangle. \tag{3.6}$$

Example 3.2 Distance of a point and a plane - sanity check

Let $P = (2, -4, 3)$. Let's calculate its distance to the three principle planes xy, xz and yz. We will use the origin as the point A, since all of these planes contain it. In each case, $\mathbf{c}$ will have as components the respective coordinates of P.

1. The normal to the xy-plane is $\hat{\mathbf{z}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$. The dot product $\langle \mathbf{c}, \hat{\mathbf{n}} \rangle$ is therefore $d = \cancel{2 \cdot 0} + \cancel{(-4) \cdot 0} + 3 \cdot 1 = 3$.
2. The normal to the xz-plane is $\hat{\mathbf{y}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$. The dot product $\langle \mathbf{c}, \hat{\mathbf{n}} \rangle$ is therefore $d = \cancel{2 \cdot 0} + (-4) \cdot 1 + \cancel{3 \cdot 0} = -4$.
3. The normal to the yz-plane is $\hat{\mathbf{x}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$. The dot product $\langle \mathbf{c}, \hat{\mathbf{n}} \rangle$ is therefore $d = 2 \cdot 1 + \cancel{(-4) \cdot 0} + \cancel{3 \cdot 0} = 2$.

As expected, the distance we get from P to each of the principle planes in $\mathbb{R}^3$ using Equation 3.6 is exactly the respective coordinate of P.

Sphere-sphere collision

Consider two solid spheres which have a single point of contact A. Let $m_1, r_1, \mathbf{x}_1$ and $\mathbf{v}_1$ be the mass, radius, position and velocity of the first sphere, and $m_2, r_2, \mathbf{x}_2$ and $\mathbf{v}_2$ the respective quantities for the second sphere (Figure 3.4). The line connecting the centers of the two spheres is in the direction $\hat{\mathbf{n}}$ (without loss of generality let us assume that the normal vector $\hat{\mathbf{n}}$ points from $\mathbf{x}_1$ towards $\mathbf{x}_2$). The unit vector $\hat{\mathbf{t}}$ is orthogonal to $\hat{\mathbf{n}}$ (and without loss of generality we will assume that it is oriented counter-clockwise from $\hat{\mathbf{n}}$).

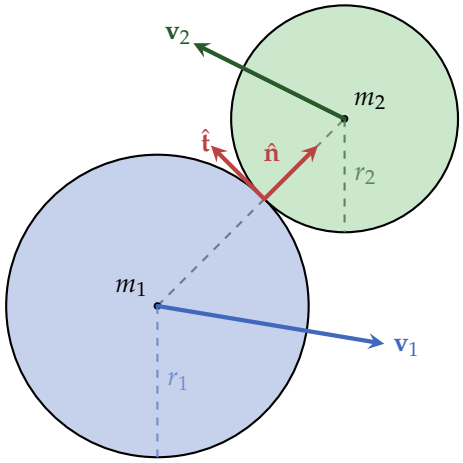

Figure 3.4: Text text text

Note that we did not define a coordinate system, nor the number of dimensions d for the problem. The only restriction is that $d \geq 2$.

Conservation of momentum means that the velocities of the spheres following the collision, $\mathbf{u}_1, \mathbf{u}_2$, are related to their velocities before the collision by

$$\begin{aligned} & m_1\mathbf{v}_1 + m_2\mathbf{v}_2 = m_1\mathbf{u}_1 + m_2\mathbf{u}_2 \\ \Rightarrow \quad & m_1\left(\mathbf{v}_1 - \mathbf{u}_1\right) = m_2\left(\mathbf{u}_2 - \mathbf{v}_2\right). \end{aligned} \tag{3.7}$$

Conservation of energy means that the velocities are also related by

$$\begin{aligned} & \frac{1}{2}m_1\|\mathbf{v}_1\|^2 + \frac{1}{2}m_2\|\mathbf{v}_2\|^2 = \frac{1}{2}m_1\|\mathbf{u}_1\|^2 + \frac{1}{2}m_2\|\mathbf{u}_2\|^2 \\ \Rightarrow \quad & m_1\left(\|\mathbf{v}_1\|^2 - \|\mathbf{u}_1\|^2\right) = m_2\left(\|\mathbf{u}_2\|^2 - \|\mathbf{v}_2\|^2\right). \end{aligned} \tag{3.8}$$

However, the forces involved in the collision can not have a component in the $\hat{\mathbf{t}}$ direction, and are limited to only point in the $\hat{\mathbf{n}}$ direction. Therefore, we can reduce the problem to this direction only by projecting all velocities involved in the problem on $\hat{\mathbf{n}}$, i.e. Equation 3.7 becomes

$$m_1\langle\mathbf{v}_1, \hat{\mathbf{n}}\rangle + m_2\langle\mathbf{v}_2, \hat{\mathbf{n}}\rangle = m_1\langle\mathbf{u}_1, \hat{\mathbf{n}}\rangle + m_2\langle\mathbf{u}_2, \hat{\mathbf{n}}\rangle. \tag{3.9}$$

TEXT TEXT TEXT

$$\begin{aligned}\mathbf{u}_1 &= \mathbf{v}_1 - \frac{2m_2}{m_1+m_2}\langle \mathbf{v}_1 - \mathbf{v}_2, \hat{\mathbf{n}}\rangle \hat{\mathbf{n}}, \\ \mathbf{u}_2 &= \mathbf{v}_2 + \frac{2m_1}{m_1+m_2}\langle \mathbf{v}_1 - \mathbf{v}_2, \hat{\mathbf{n}}\rangle \hat{\mathbf{n}}.\end{aligned} \tag{3.10}$$

To avoid reduntant calculations, we can factor out the common quantity of both velocities:

$$K = \frac{2}{m_1+m_2}\langle \mathbf{v}_1 - \mathbf{v}_2, \hat{\mathbf{n}}\rangle \hat{\mathbf{n}}, \tag{3.11}$$

yielding

$$\begin{aligned}\mathbf{u}_1 &= \mathbf{v}_1 - Km_2, \\ \mathbf{u}_2 &= \mathbf{v}_2 + Km_1.\end{aligned} \tag{3.12}$$

Reducing collision test complexity

Much like in the case of gravitational simulations, the number of calculations needed to resolve collisions between N particles is $N(N-1) = N^2 - N$, which grows like $\mathcal{O}\left(N^2\right)$. And also like in that case, we would like to reduce this complexity. One possible method is using a quad- or oct-tree as in the Barnes-Hutt algorithm mentioned in (REF!), but in the case of perfectly spherical particles which only experience instantaneous elastic collisions, there are much simpler method.

The first and simplest of these methods is the **Axis-Aligned Bounding Box**, or simply **AABB**. In the 2-dimensional case, a **bounding box** (BB) of an object is the minimal rectangular area surrounding it completely (i.e. that contains the entire object within it). An AABB of an object is (as the name suggests) a BB with sides parallel to the 2 axes (Figure 3.5).

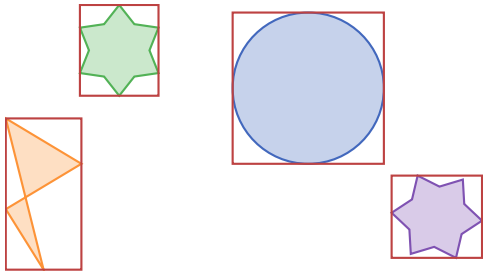

Figure 3.5: Axis-Aligned Bounding Boxes (AABBs) for different objects in 2-dimensions.

In the case of a circle of radius r, its AABB is a square of side $2r$ with the sides parallel to the x and y axes, respectively, and sharing its center with the circle. Therefore, given a circle of radius r centered at (x, y), its AABB is the square with the two opposing corners

$$\begin{aligned}\mathbf{c}_{\text{LL}} &= (x-r, y-r), \\ \mathbf{c}_{\text{UR}} &= (x+r, y+r),\end{aligned} \tag{3.13}$$

where LL stands for "Lower-Left" and UR stands for "Upper-Right".

When two objects collide, their AABBs necessarily overlap (Figure 3.6). This means that if we know that two circles have no overlap in their AABBs, we can skip the check whether they collide or not, as it is entirely redundant. Luckily, checking overlaps in AABBs can be very fast compared to checking even the simplest collision, and this way we gain advantage in computation speed. Let's see how it is done.

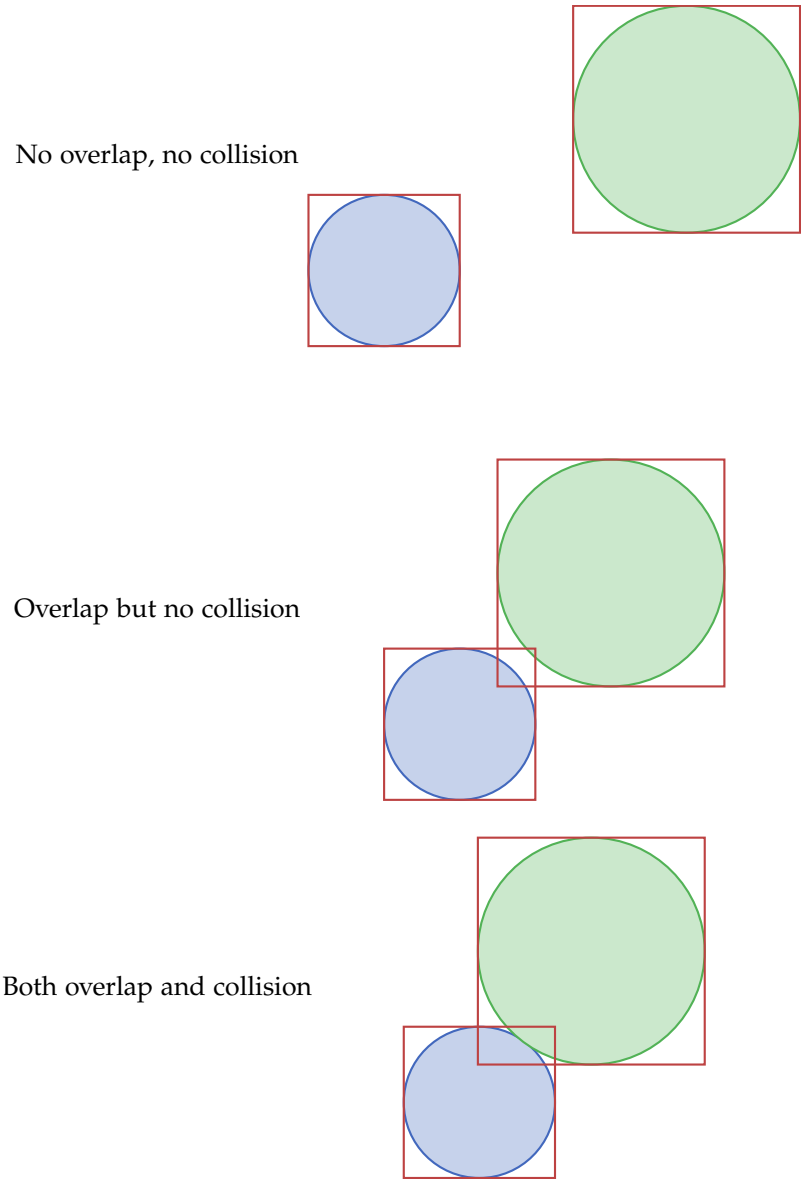

Figure 3.6: AABBs of twho circles. There can be three different relations between the circles in regards to their AABBS - no overlap between the AABBs and no collision between the circles (top), and overlap of the AABBs but no collision (middle), and an overlap with a collision (bottom). A collision between the circles necessarily means that there is also an overlap between the AABBs.

We first examine the conditions for an overlap in a single dimension: given two intervals $I_1 = [a, b]$ and $I_2 = [c, d]$, let's assume without loss of generality that a is the smallest number of the four quantities a, b, c and d. The condition for overlap between the two intervals is simply $c < b$, i.e. that the left-most point of I_2 is to the *left* of the right-most point of I_1 (Figure 3.7).

We can consider an AABB across each axis to be an interval: if the lower-left and upper-right corners of the AABB are at $\mathbf{c}_{\mathrm{LL}} = (x_1, y_1)$

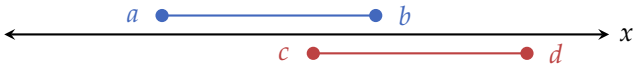

Figure 3.7: For two intervals $(a, b), (c, d)$ (assuming $a < b, c, d$) to overlap, the condition $c < b$ must be fulfilled.

and $\mathbf{c}_{\text{UR}} = (x_2, y_2)$, respectively, then the vertical sides of the AABB are represented as an interval $I_{\text{horiz}} = [x_1, x_2]$, and the horizontal sides by $I_{\text{vert}} = [y_1, y_2]$ (Figure 3.8).

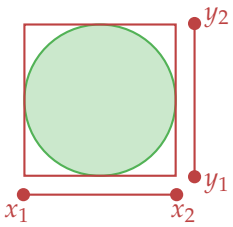

Figure 3.8: The sides of an AABB as two intervals $I_{\text{horiz}} = [x_1, x_2]$ and $I_{\text{vert}} = [y_1, y_2]$.

Two AABBs overlap **if and only if** both their horizontal and vertical intervals overlap (Figure 3.9)

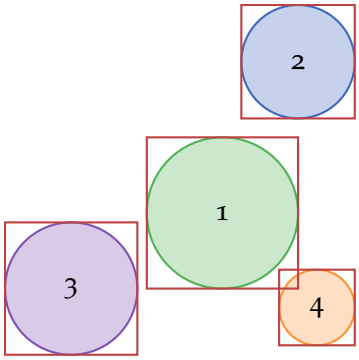

Figure 3.9: Four circles with different overlaps between their AABBs: 1-2 and 2-4 overlap horizontally only. 1-3 and 3-4 overlap vertically only. 1-4 overlap in both directions.

3.3 *Brownian dynamics*