Chapter 3
Units and Measurement

In physics we study nature quantitatively—we describe nature with numbers. But not with numbers alone. We need to relate numbers to the physical world using measurement units. For example, you could measure the width of your desk using a pencil as in Fig. 3.1, finding that the width is 6 times the length of your pencil and 2.73 times the length of your shoe. The number alone, 6 or 2.73, does not make any sense in the physical world without the unit: a pencil or a shoe. Using a pencil or a shoe may be practical for you, but not if you want to communicate your result to someone else. Therefore we need standardized units.

3.1 Standardized Units

Here we use the standard units provided by SI[1]—the Systeme International—that provide a few precisely defined *base units*. For example, the standard unit for length in the SI system is called 1 m. To ensure that this standard is the same everywhere there is a standard meter bar, made of platinum-iridium, which is kept at the International Bureau of Weights and Measures outside Paris. Accurate copies of this bar have been sent to various national standards laboratories around the world.

All other units are derived from the base units in Table 3.1. For example, the unit for force is Newton and the unit for energy is Joule. Both are defined as combinations of the base units in the SI system:

$$1\,\text{Newton} = 1\,\text{N} = 1\,\text{kg}\,\text{ms}^{-2}, \tag{3.1}$$

$$1\,\text{Joule} = 1\,\text{J} = 1\,\text{N}\,\text{m} = 1\,\text{kg}\,\text{m}\,\text{s}^{-2}\,\text{m} = 1\,\text{kg}\,\text{m}^2\,\text{s}^{-2}. \tag{3.2}$$

[1] http://en.wikipedia.org/wiki/International_System_of_Units

A. Malthe-Sørenssen, *Elementary Mechanics Using Matlab*,
Undergraduate Lecture Notes in Physics,
DOI 10.1007/978-3-319-19587-2_3

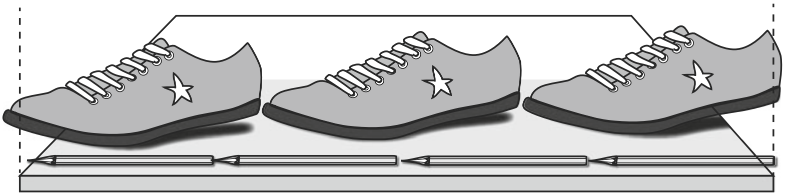

Fig. 3.1 Measuring the width of a table with a pencil and a shoe

Table 3.1 SI units and standard prefixes

SI units			Standard prefixes		
Quantity	Unit	Symbol	Factor	Prefix	Symbol
Length	Meter	m	10^{12}	Tera-	T
Time	Second	s	10^{9}	Giga-	G
Mass	Kilogram	kg	10^{6}	Mega-	M
Amount of substance	Mole	mol	10^{3}	Kilo-	k
Electric current	Ampere	A	10^{-1}	Deci-	d
Temperature	Kelvin	K	10^{-2}	Centi-	c
Luminous intensity	Candela	cd	10^{-3}	Milli-	m
			10^{-6}	Micro-	μ
			10^{-9}	Nano-	n
			10^{-12}	Pico-	p
			10^{-15}	Femto-	f

This high level of precision may seem unnecessary, but units were no joke for the NASA teams responsible for the safe landing of the Mars Climate Orbiter.[2] The Orbiter was launched in December 1998, and NASA lost contact with it on September 23rd, 1999. The Orbiter was last heard from as it approached Mars way too close to the surface. It turned out that the two groups in NASA working on the approach of the Orbiter used different measurement units. One group used English units (feet and pounds) and the other group used metric units (meters and kilograms), and they had not communicated this difference clearly. The result was devastating for the Orbiter and now serves as a warning for all of us: Keep track of your units!

Numerical Notation

Measured quantities in the SI system may be very large or very small. For example, the velocity of light in the SI system is:

$$c = 299{,}792{,}458\ \mathrm{m/s}, \tag{3.3}$$

[2] http://en.wikipedia.org/wiki/Mars_Climate_Orbiter

and the length of the E. coli bacteria is

$$l = 0.000001\,\text{m}. \tag{3.4}$$

A more practical way to represent these numbers is to use scientific notation, where we use a base of 10. The velocity of light is then:

$$c = 2.9979248 \ \times \ 10^{8}\,\text{m/s}, \tag{3.5}$$

and the length of the E. coli bacteria is

$$l = 1 \ \times \ 10^{-6}\,\text{m}. \tag{3.6}$$

We no longer need to count the number of digits to see the order of magnitude of the number. This notation is also used by matlab where we give a number by a decimal number followed by `e` and the number x of tens, corresponding to the exponent x in 10^{x}:

```
>> l = 0.000001
l =   1.0000e-006
```

Similarly, we use scientific notation when we input numbers in matlab:

```
>> u = 1.25e6
u =     1250000
```

Prefixes

To make life simpler, there are prefixes for various standard factors so that we do not have to write out the power of tens each time. Instead of writing

$$0.001\,\text{m} = 1.0 \times 10^{-3}\,\text{m}, \tag{3.7}$$

we write

$$0.001\,\text{m} = 1\,\text{millimeter} = 1\,\text{mm}. \tag{3.8}$$

Similarly, we write

$$1\,\text{nm} = 1 \times 10^{-9}\,\text{m}, \tag{3.9}$$

and the well-known kilometer:

$$1\,\text{km} = 1 \times 10^{3}\,\text{m} = 1000\,\text{m}. \tag{3.10}$$

We need a large span of prefixes because our physical world spans over a large range of scales.

3.2 Changing Units

There are many different units used to describe real world quantities: We may measure speed in meters per second, kilometers per hour or feet per second. How do we translate between them? Conversion is done by substituting equivalent quantities represented by different units.

The method is best demonstrated by example. For example, the speedometer in your car reports the speed to be 60 kilometers per hour (km/h). How can you convert this to meters per second (m/s)? This is done by replacing km with meters and hours with seconds by rewriting the units: 1 km = 1000 m and 1 h = 3600 s:

$$60\,\frac{\text{km}}{\text{h}} = 60\frac{1000\,\text{m}}{3600\,\text{s}} = 16.6\,\frac{\text{m}}{\text{s}}. \tag{3.11}$$

What about the reverse scheme? If the wind is blowing 20 meters per second (m/s), what does this correspond to in kilometers per hour (km/h)? We use the same technique as above, although we need an intermediate step to write 1 m in terms of km and 1 s in terms of hours:

$$1\,\text{km} = 1000\,\text{m} \;\Rightarrow\; \frac{1}{1000}\,\text{km} = 1\,\text{m}, \tag{3.12}$$

and similarly:

$$1\,\text{h} = 3600\,\text{s} \;\Rightarrow\; \frac{1}{3600}\,\text{h} = 1\,\text{s}. \tag{3.13}$$

We use these expressions to convert from m/s to km/h:

$$20\,\frac{\text{m}}{\text{s}} = 20\frac{\frac{1}{1000}\,\text{km}}{\frac{1}{3600}\,\text{h}} = 20\,\frac{3600}{1000}\,\frac{\text{km}}{\text{h}} = 72\,\frac{\text{km}}{\text{h}}. \tag{3.14}$$

3.3 Uncertainty and Significant Digits

Measured quantities are never exact, but have some uncertainty, which depends on the measurement method. We specify the uncertainty of a measurement by giving a range of values for the measured quantity. If a length is given as 25.0 ± 0.5 m, it means that length most probably is between $25.0 - 0.5$ m and $25.0 + 0.5$ m. (For quantities with a known distribution of values, we typically describe the range by the standard deviation of the measured quantity).

In practice we use an even briefer way of specifying the uncertainty—by only giving the *significant digits* when we write down the numerical value of a quantity. For a distance of 25.0 ± 0.5 m it does not make sense to provide many more digits for the length. Writing 25.0000 ± 0.5 m clearly overstates the precision, since we do not really know whether the number is 24.5 or 25.5 m. Instead we use the number of digits to indicate the uncertainty. We only provide the number of digits we are certain of—the significant digits. Instead of writing $d = 25.0 \pm 0.5\,\mathrm{m}$ we write $d = 25\,\mathrm{m}$. This would imply the same: we do not really know whether it is 25.5 or 24.6 m, but we know that it is not 22 m. Standard practice is that the last digit provided may be uncertain. If you write $d = 25\,\mathrm{m}$ is means that the value could be $d = 24\,\mathrm{m}$ or $d = 26\,\mathrm{m}$, and if you write $d = 25.0\,\mathrm{m}$ it means that the value could be $d = 25.05\,\mathrm{m}$, but is probably not $d = 25.15\,\mathrm{m}$.

This implied uncertainty is why you should never report the full numerical values you get from your calculator or your program. Your program returns a lot of digits—as many digits as it stores—but these digits may not be significant. You therefore need to ensure that you report only the number of significant digits. But how do you know that? You know it because the uncertainty of the result of your calculations will depend on the uncertainty of the numbers you put in. Your calculations cannot improve the uncertainty! The results you report must therefore always reflect the number of significant digits in the data you start with. And it is the number with the largest uncertainty that determines the final uncertainty.

You will learn more formally about how to handle uncertainties in complex calculations in your laboratory courses, but for now you can use the following rules of thumb for handling the number of significant digits:

Multiplication: The number with the least number of significant digits determines the number of significant digits of the result

Addition: It is the position of the decimal point in each of the numbers that determines the uncertainty.

For *addition*, the position of the decimal point and not the number of significant digits determines the uncertainty of the result:

$$3.4\,\mathrm{mm} + 10\,\mathrm{mm} = 13\,\mathrm{mm}, \tag{3.15}$$

$$1000.00\,\mathrm{m} + 5\,\mathrm{m} = 1005\,\mathrm{m}, \tag{3.16}$$

$$1\,\mathrm{km} + 10\,\mathrm{m} = 1\,\mathrm{km} + 0.010\,\mathrm{km} = 1\,\mathrm{km}, \tag{3.17}$$

$$1.000\,\mathrm{km} + 10\,\mathrm{m} = 1.000\,\mathrm{km} + 0.010\,\mathrm{km} = 1.010\,\mathrm{km}, \tag{3.18}$$

where you see how we lose small numbers due to uncertainty in the addends.

For *multiplication*, the number of significant digits in the result cannot be larger than in any of the factors:

$$2.10101\,\mathrm{N} \times 4.0\,\mathrm{m} = 8.4\,\mathrm{Nm}, \tag{3.19}$$

$$2\,\mathrm{kN} \times 4.400\,\mathrm{m} = 9\,\mathrm{Nm}, \tag{3.20}$$

$$2\,\mathrm{kN} \times 4.000\,\mathrm{m} \times 2.600\,\mathrm{s} = 2 \times 10^1\,\mathrm{Nms}. \tag{3.21}$$

3.4 Numerical Representation

Several challenges arise when we use a computer to solve physics problems. First, the computer only stores numbers and not the units. Therefore you need to keep track of the units. Second, your programs always provide more digits than the number of significant digits, so you have to keep track of the number of significant digits. Third, the computer introduces new errors by itself—errors due to the digital representation of numbers, and errors that are inherent in the algorithms we use to solve a problem.

Units on the Computer

There are no units in your program, so you need to keep track of the units yourself. If you calculate the velocity of a car driving 2.0 km in 10.0 s, you find:

```
>> x = 2.0;
>> t = 10.0;
>> v = x/t
v =     0.2000
```

It is your job to know that the units for the answer is m/s. Because you could have done exactly the same calculation with $x = 2.0$ km and $t = 10.0$ s, and the answer would instead be 0.2 km/s. Or you could have done the calculation $x = 2.0$ nm and $t = 10.0$ ms, and the result would have been $v = 0.2$ nm/ms.

How to ensure that you keep track of the units correctly? We advice you to write the units into your code when you define numbers. In your script, you could write:

```
x = 2.0; % m
t = 10.0; % s
```

This does not solve the problem, but this is a habit that makes it easier for you to spot the units.

In addition, we advice you to always use units on the axes in your plots. Indeed, we argue that a plot in physics is not complete without units on the axes. Make this a habit from the start!

Finally, as you become more proficient in numerical methods, you should also learn how to rewrite your equations in non-dimensional form.

Too Many or Too Few Digital Digits

Your program tends to return a long sequence of numbers irrespective of the number of digits you put in: It does not care about significant numbers as do we:

```
>> x = 10.0;
>> t = 3.0;
>> v = x/t
v =     3.3333
```

It is therefore always your job not to report the numbers you get out directly, but to apply the rules of the number of significant digits before you report the calculated value.

Digital Representation of Numbers

The numbers you use in your program can be of various *types* and are stored digitally in various ways.

Numbers in computers are stored in units of 4 or 8 bytes, which is the size most efficiently handled by the hardware. One byte corresponds to an 8-bit binary number. One bit is a binary number, which means that it is either 0 or 1. With 8 bits we can make 2^8 different numbers. For example, we could enumerate the numbers from 0 to $255 = 2^8 - 1$. You may use binary as a data type in your programs, but you will usually use either integers, called *integers*, or real numbers, called *floating point numbers*.

Integers

Integers are usually either represented by 4 or 8 bytes of computer memory. This means that there is a maximum number that can be represented. If you use a 4-byte integer the integers can range from $-2^{31} = -2{,}147{,}483{,}648$ to $2^{31} - 1 = 2{,}147{,}483{,}647$. However, in matlab your programming language will sort out problems if you exceed this number by automatically switching to a data type such as a 8 byte integer.[3]

Floating-Point Numbers

While floating-point numbers are supposed to represent real numbers, they also have a finite digital precision, corresponding to the number of possible values that can be spanned by the 4 or 8 bytes used to represent the floating-point number.

Floating point numbers are represented by the (significant) digits, and an exponent, using the method that we called scientific notation.

For a 4-byte floating-point number, the IEEE[4] standard determines that 23 bits are used for the digits and 9 bits are used for the exponents—including the signs for both. 4-byte floating point numbers therefore have 6–9 (significant) digits, and cover a range from -3.4×10^{38} to -1.4×10^{-45} for negative numbers, and from 1.4×10^{-45} to 3.4×10^{38} for positive numbers.

[3] In other programming languages, type switching is not standard, and you may "run out of integers" with surprising results.

[4] http://ieee.org

However, you will not encounter 4-byte numbers when programming in matlab since by default the floating-point numbers are 8-byte. In this case, numbers have 15–17 (significant) digits, and cover a range from -1.8×10^{308} to -5×10^{-324} for negative numbers, and from 5×10^{-324} to 1.8×10^{308} for positive numbers. With 8-byte floating-point numbers you will seldom encounter cases where you are limited by the numerical precision. For all practical purposes you will be limited by the number of significant digits in your measurements. In addition, you may encounter special numbers. Such as when dividing by zero:

```
>> 1.0/0.0
ans =    Inf
```

It may be surprising that matlab does not return an error when you divide by zero. Instead it return `Inf`, which acts as a number. This feature may cause problems when you try to find errors in your code. Therefore you should always look for occurrences of `Inf` when something goes wrong. While the number of significant digits usually does not pose a practical problem, we still need to remember that the floating-point numbers only have a given resolution. Not all numbers are possible to represent, and rounding errors may pop up in unexpected places. For example:

```
>> l = 11.11
l =   11.109999999999999
```

shows that even when you just enter a number, it may not be stored as exactly the same number as you entered, however, the error, the uncertainty, is very small. This has few practical consequences. But there is one practical consequence you should be aware of: You should not test if a floating-point number is equal to a particular value. For example, you are usually advised not to test if a variable is exactly equal to zero, but rather test when it is very close to zero or when the variable changes sign. For example, if you model a falling ball as a function of time, you may experience that it never reaches precisely zero height, it may be slightly above 0 at one time and then slightly below 0 at the next timestep—independently of how small you make your time steps. Therefore, be careful when checking if a value is exactly equal to zero!

Numerical Errors

We will address various numerical methods throughout this book. You should be aware that the numerical methods themselves may also introduce errors, and that these errors in some cases can be non-negligible. This may be because the methods are unsuited for the problems we address, or it may, for example, be that you use too large an integration step.

For most numerical methods from calculus, such as for numerical derivation and integration, the error from the numerical calculation depends on the size of the integration step, which usually corresponds to the time step. Typically, the error decreases with the integration step, and you can improve your error by decreasing your integration step. But only up to a certain limit, beyond which numerical rounding errors dominate. Trying to get very precise results be choosing very small integration steps can therefore sometimes lead to large errors. We will address these features in detail when we discuss the various numerical methods.

You must always take extra care when applying a numerical method to analyze your problem because:

- the method you apply may have an error that may affect the number of significant digits in your answer
- the method you apply may be unstable, producing errors that are orders of magnitude off the correct result
- you may have implemented the method incorrectly, producing a result that is significantly wrong

We therefore always use our physical intuition and insight when we analyze our results. You need to check whether the results are reasonable compared with your intuition. And you also need to check whether the results violate basic physical principles, such as the conservation laws of mass and momentum that you will learn throughout this book. However, by ensuring that you always test your methods against standardized problems with know solutions before venturing into the unknown, numerical methods provide a robust, versatile set of tools that allows us to solve practically any problem in mechanics with the precision we want.

Summary

Standard units:

- In physics all quantities are measured in physical units, and we always need to include the units when we present a quantity
- There is an international standard set of units, the SI units, which is based on a few base units: *meter*, *second*, *kilogram*, *mol*, *ampere*, *kelvin*, and *candela*
- All other units are derived from the base units. For example, the unit for force, Newton, is defined as $1\,\mathrm{N} = 1\,\mathrm{kg\,m\,s^{-2}}$.
- We use scientific notation to present numbers, by giving the base number between 0 and 10, and an exponent of 10: $998.23\,\mathrm{m} = 9.9823\,10^2\,\mathrm{m}$,
- There are standard prefixes for the most common orders of ten: so that $9.0\,10^{-9}\mathrm{s} = 9.0\,\mathrm{ns}$ (see Table 3.1).

Unit conversion:

- Units are treated as ordinary mathematical symbols
- We can convert between units by replacing equivalent units, for example, we can replace $1000\,\mathrm{m}$ by $1\,\mathrm{km}$, or we can replace $1\,\mathrm{h}$ by $3600\,\mathrm{s}$: $72\,\mathrm{km/h} = 72\,1000\,\mathrm{m}/\,(3600\,\mathrm{s}) = (72/3.6)\,\mathrm{m/s} = 20\,\mathrm{m/s}$

- We can similarly convert by replacing 1 m by $(1/1000)$ km and 1 s by $(1/3600)$ h: $20\,\text{m/s} = 20\,(1/1000)\,\text{km}/\,((1/3600)\,\text{h}) = 72\,\text{km/h}$

Uncertainty:

- Measuring a physical quantity results in uncertainty. We specify the uncertainty by a range: $x = 20.0 \pm 0.5\,\text{m}$
- Number are given so that the last digit may be uncertain. We call this number of digits the *significant number of digits*.
- If you multiply two numbers, the number of significant digits in the answer is determined by the factor with the least number of significant digits $9.8 \times 10000.0 \times 2 = 2 \times 10^5$
- If you add two numbers, the number of significant digits in the answer depends on the position of the decimal place: $1000.0 + 2 = 1002$ and $1000 + 0.1 = 1000$

Digital representation of numbers:

- Numerical quantities are without units, so you need to keep track of them
- Your answers and plots should include units
- Numbers are represented as integers or floating point number in the computer
- Floating point numbers have a limited resolution, which may lead to rounding errors
- Numerical algorithms may introduce errors that are significant—error analysis is important.

Exercises

3.1 Kilometers per hour. A car is driving at 144 km/h. Find the velocity in m/s.

3.2 Miles per hour. You car speedometer is showing both km/h and mph, miles per hour. 1 mile is 1609.34 m.
(a) If your speedometer is showing 70 km/h. What does it show in mph?
(b) If your speedometer is showing 55 mph. What does it show in km/h?

3.3 Acceleration of gravity. The acceleration of gravity is approximately $g = 9.8\,\text{m/s}^2$.
(a) Find the acceleration of gravity in feet per second squared, ft/s^2. 1 foot is 0.3048 m.
(b) Find the acceleration of gravity in kilometers per hour squared, km/h^2.

3.4 Bacterial volume. A bacteria is like a cylinder with length $4\,\mu$m, and radius $1\,\mu$m.
(a) Find its volume in μm^3.
(b) Find its volume in m^3.
(c) Find its volume in liters.

3.5 Ruler length. You have a platinum-iridium ruler that you have measured to be 0.11236 m using a high precision method. You use the ruler to measure the length of your desk, and find that the ruler fits about 20 times across the desk. What is the length of your desk?

3.6 Sphere mass and volume. A small steel sphere has a radius of 1.2 mm.
(a) What is the volume of the sphere?
(b) The density of the particular steel alloy used is $\rho = 7782\,\text{kg/m}^3$. What is the mass of the sphere?

3.7 Laserlength. You use a laser distance measurer to measure the distance from one wall to another in your house. It reads 11.2 m. As you walk across to the other wall, you see that there is a small protrusion from the wall. Using your tape measure, you find that the protrusion is 5mm high. What is the distance from the other wall to the protrusion?

3.8 Salmon speed. You have designed a special circuit to measure the swimming speed of a salmon. The circuit has a length of 62.8 m. You measure the time a salmon takes to swim one lap to be 20.6 s.
(a) What is the swimming speed of a salmon?
Your assistant insists that you would get better precision if you instead measured the time the salmon took to swim 10 rounds. You find that the salmon uses 206.0 s to swim 10 rounds.
(b) What is the speed of the salmon?
(c) Does this produce better accuracy? Can you give other examples of situations where this strategy would improve the accuracy?