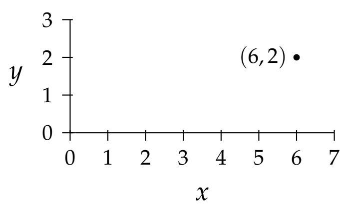
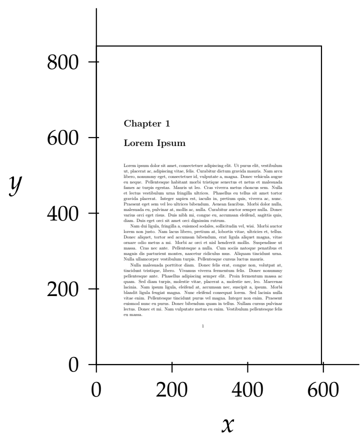
Chapter 1

Putting Marks on Paper

In this book, we shall need very little formal mathematics, but if we are considering the arrangement of letters and words and lines and pictures on the page, we shall need a way of discussing the idea of position – that is to say, *where* something is, rather than just *what* it is. Thankfully, our paper is flat and rectangular, so we can use the simple coordinates we learned in school. In other words, we just measure how far we are above the bottom left corner of the page, and how far to the right. We can write this as a pair of numbers; for example, the coordinate $(6, 2)$ is six lengths right, and two lengths up from the bottom-left of the page. It is convention to use x to denote the across part of the coordinate, and y to denote the up part. These are known as *Cartesian* coordinates, named for René Descartes (1596–1659) – the Latin form of his name is Renatus Cartesius, which is a little closer to "Cartesian". The idea was discovered independently, at about the same time, by Pierre de Fermat (1601–1665). Here is the coordinate $(6, 2)$ drawn on a little graph, with axes for x and y, and little marks on the axes to make it easier to judge position by eye:



We can assign units if we like, such as centimetres or inches, to define what these "lengths" are. In publishing, we like to use a little unit called a *point* or *pt*, which is 1/72 of an inch. This is convenient because it allows us to talk mostly using whole numbers (it is easier to talk about 450pt than about 6.319 inches). We need such small units because the items on our page are quite small and must be carefully positioned (look at the writing on this page, and see how each tiny little shape representing a character is so carefully placed) Here is how an A4 page (which is about 595 pts wide and about 842 pts tall) might look:

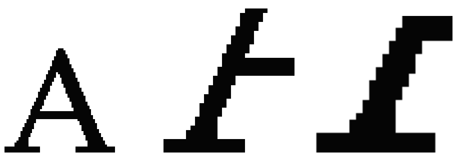


You can see that the chapter heading "Chapter 1" begins at about $(80, 630)$. Notice that the coordinates of the bottom left of the page (called the *origin*) are, of course, $(0, 0)$. The choice of the bottom left as our origin is somewhat arbitrary – one could make an argument that the top left point, with vertical positions measured downwards, is a more appropriate choice, at least in the West where we read top to bottom. Of course, one could also have the origin at the top right or bottom right, with horizontal positions measuring leftward.

We shall be using such coordinates to describe the position and shape of each part of each letter, each word, and each paragraph, as well as any drawings or photographs to be placed on the page. We will see how lines can be drawn between coordinates, and how to make the elegant curves which form the letters in a typeface.

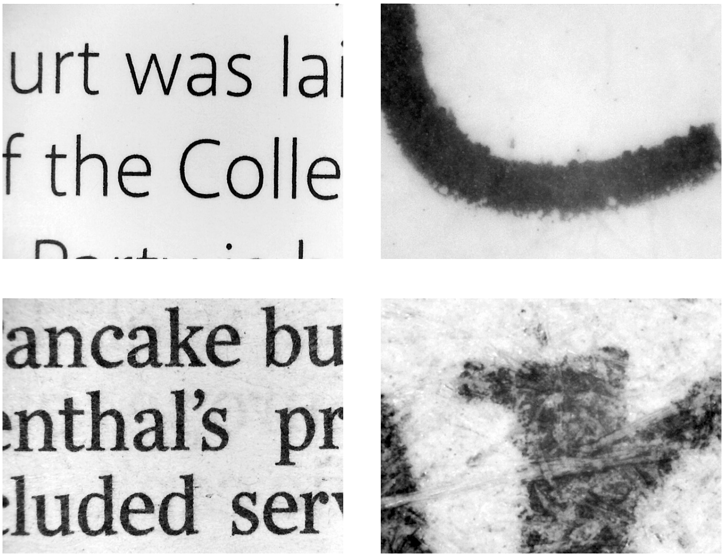
Once we have determined what shapes we wish to put on each page, we must consider the final form of our document. You may be reading this as a physical paperback book, printed and bound by very expensive equipment. You may be reading it as an electronic document (such as a PDF file) on your computer, tablet, or smartphone. Or, you may be reading it on some sort of special-purpose eBook reader. Each of these scenarios has different characteristics. Every page of the printed book is made up of hundreds of millions of little dots, each of which may be white (no ink) or black (ink). We cannot typically see the dots with the naked eye. The number of dots is known as the *resolution* (from the word "resolve"). A low resolution image is one where it is easy for the eye to resolve (that is, distinguish) the individual dots. A high resolution image has dots so small and tightly packed that the naked eye cannot distinguish them.

A high resolution printer, such as the one printing the physical copy of this book, may have as many as 600 or 1200 dots per inch (dpi); that is to say, between $600 \times 600 = 360{,}000$ and $1200 \times 1200 = 1{,}440{,}000$ dots per square inch. The screen of a computer or tablet may only have 100 to 300 dpi, but it can display many shades of grey and colours. If the resolution is too low, we see blocky images. Here is part of a capital letter A in black and white at 60 dpi, 30 dpi, and 15 dpi:



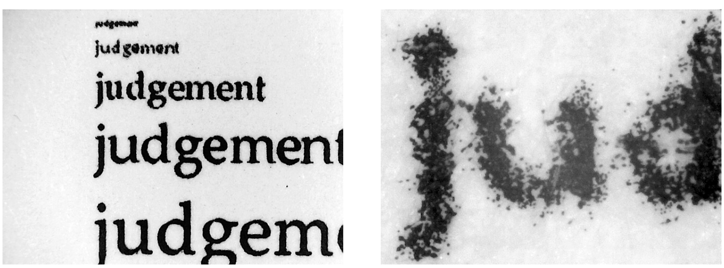
We have used square dots here, such as may be used on a modern computer screen (we call them *pixels*, which is short for "picture elements"). For viewing a page on a typical tablet computer, we might have only $2048 \times 1536 = 5{,}193{,}728$ dots on the whole screen, but they may be colours or greys, as well as black or white. When printing a book like this, we have many more dots, but only black ink. Let us say, for example, that we have a US Letter page (8.5 inches by 11 inches) and we are printing at a resolution of 1200 dpi. We have $1200 \times 1200 = 1{,}440{,}000$ dots per square inch, so we have $1200 \times 1200 \times 8.5 \times 11$, or 134,640,000 dots on the page, each of which may be black or white.

Here are some photographs, taken under a microscope, of lettering as it appears in high quality printing, and on the much lower quality, cheaper newsprint used for the daily newspaper:



The upper row shows high-resolution printing of lettering on coated paper, such as might be used for a glossy pamphlet, under a microscope at 20x magnification, and the same at 400x magnification. The lower row is standard text of the London Times printed on newsprint at 20x magnification and the same at 400x magnification.

The home or office laser printer works by using a laser to prepare a roller in such a way that a powder will adhere only to areas where the laser has not been shone. The powder (called toner) is then transferred from the roller to paper, and bonded to it by heat. The particles of toner behave rather differently from ink:

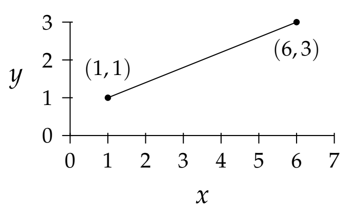


On the left is a word printed in 1pt, 2pt, 4pt, 6pt, and 8pt text under a microscope, with magnification at 20x. On the right, the

2pt word with magnification at 400x (a typeface of a given size is roughly that number of points tall, say, for its capital letters.)

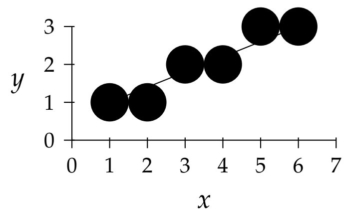
All these dots form a huge amount of information which is costly and difficult to manipulate. So, we will normally store our pages in a more structured way – some paragraphs, which are made of words, which are made of letters, which are drawn from some typeface, which is defined using lines and curves. The hundreds of millions of dots which will finally make up the page only exist temporarily as the image is printed, or placed onto the screen. (The exception, of course, is when we use photographs as part of our page – the colour of each dot is captured by the camera, and we must maintain it in that form.) Until recently the storage, communication, and manipulation of high resolution photographs was a significant problem. The storage, communication, and manipulation of high resolution video still is – imagine how many little coloured dots make up a still image, then multiply by 25 or 50 images per second for the 2 hours (7200 seconds) a feature film lasts.

We have talked only about single dots. However, we shall need lines, curves, and filled shapes to build our page. Suppose that we wish to draw a line. How can we work out which dots to paint black to represent the line? Horizontal and vertical lines seem easy – we just put ink on each dot in that row or column, for the whole length of the line. If we want a thicker line, we can ink multiple rows or columns either side of the original line. But there are many useful lines other than the horizontal and vertical ones. To begin, we shall need a way to define a line. We can just use two coordinates – those of the points at either end. For example, here is the line $(1,1)$—$(6,3)$:

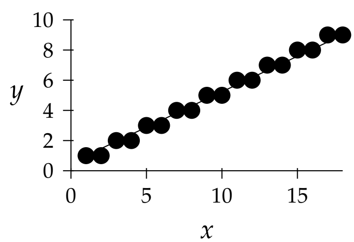


In mathematics, we would usually consider a line to be of infinite length, and so this is really a line *segment*, but we shall just call it a line. Notice that this line could equally be defined as $(6,3)$—$(1,1)$.

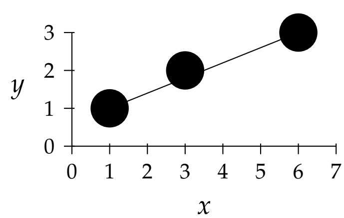
As a first strategy, let us try colouring in one dot in each column from column 1 to column 6, where the line is present. We will choose the dot whose centre is closest to the line in each case:



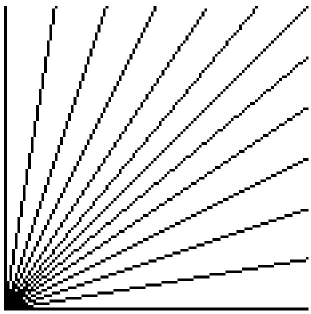
Admittedly, this does not look much like a line. But if we choose a higher resolution for a line of the same slope, and so draw more and smaller dots, we see a better approximation:



Now, you may wonder why we chose to draw one dot in each column instead of one dot in each row. For example, instead of putting one dot in each of the columns from column 1 to column 6, we might put one dot in each of the rows from row 1 to row 3, again choosing the one in that row nearest the actual line. For this shallow line, doing so would lead to a most unpleasant result:



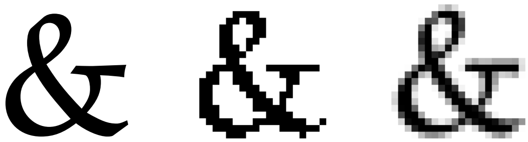
If the line is steeper than 45°, the converse is true (draw it on paper to see). So, we choose to put one black dot in each row instead of in each column in this case. Horizontal and vertical lines are simply special cases of this general method – for the vertical case we draw one dot in each row; for the horizontal case one dot in each column. For the line at exactly 45°, the two methods (row and column) produce the same result. Here is an illustration of the sorts of patterns of dots we see for lines of various slopes using this improved procedure:



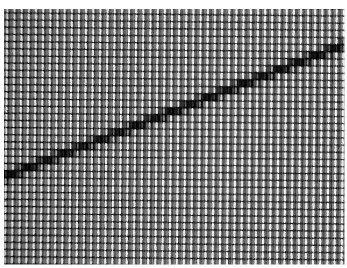
This image is 100 dots tall and wide. The results are not terribly good, for two reasons. First, at low resolutions, the individual dots are clearly visible. Moreover, the density of the lines varies. A horizontal or vertical line of length 100 will have 100 dots over its length, but the 45° line has 100 dots over a length of about 141 (the diagonal of a square with sides of length 100 is $\sqrt{2} \times 100$), and so its density of dots is lower, and it appears less dark.

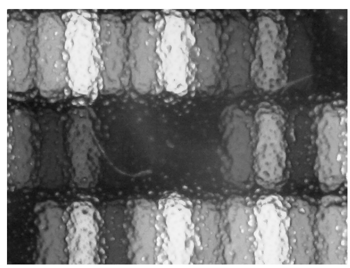
When we are using a screen, rather than paper, to display our line, we can take advantage of the ability to use more than just black and white. And so, we can use varying shades of grey: dots which are right on the line are very dark grey, dots which are just close are lighter grey. Here is a line drawn in this manner, at three scales:

We can see that the line is smoother than would otherwise be the case. If you are reading this book on an electronic device, you may be able to see this effect on the text or images with a magnifying glass. Here is another example, with a more complex, filled shape, which might be used to represent an ampersand character:



On the left is an idealised high resolution shape. In the middle, just black and white at a lower resolution. On the right, prepared for display on a screen supporting grey as well as black and white, at the same lower resolution. This use of greys is known as *antialiasing*, since the jagged edges in low resolution lines are known as *aliasing*. This term originated in the field of signal processing and is used to describe problems stemming from low-resolution versions of high-resolution signals. Here is a photograph, taken under a microscope, of such an antialiased line on a modern computer screen:

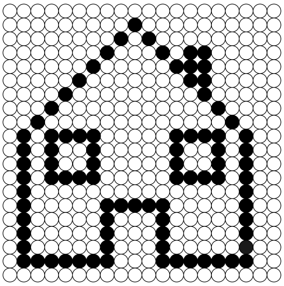




The left image is magnified 20x; the right image 400x. The rectangular shapes you can see in the images are the separate Red, Green, and Blue sub-pixels, which a monitor uses to build up all the different colours and greys it may need (the monitor makes a picture by emitting light and Red, Green, and Blue are the primary colours of light.)

What might a reasonable minimum resolution be? To simplify, let's return to the scenario where we only have black and white dots – no antialiasing. The resolution required to make the page look smooth depends on the distance at which the typical viewer sees it. For a computer screen, this might be twenty inches. For a smartphone, eight inches. For a billboard, twenty or fifty feet (if you have never walked right up to a billboard and looked at the printing, do so – it is surprisingly coarse.) The limit of the human optical system's ability to distinguish the colour of adjacent dots, or their existence or absence, is the density of light sensitive cells on the retina. At a distance of 12 inches, a density of 600 dots per inch on the printed page may be required. For a billboard, we may only need 20 or 50 dots per inch. On a screen, antialiasing allows us to use a lower resolution than we might otherwise need.

We have seen how to draw lines between points, and so we can build shapes by chaining together multiple lines. For example, the lines $(1,1)$—$(10,1)$, $(10,1)$—$(10,10)$, $(10,10)$—$(1,10)$, and $(1,10)$—$(1,1)$ form a square (you can draw it on paper if you wish). We might define this more concisely as $(1,1)$—$(10,1)$—$(10,10)$—$(1,10)$—$(1,1)$. However, if we wish to produce a filled shape (such as a letter in a word) we would have to make it up from lots of little horizontal lines or lots of little vertical ones, to make sure that every dot we wanted to be covered was covered. We should like to automate this process, so as to avoid manually specifying each part of the filled section. Consider the following child's picture of a house, made from several lines:



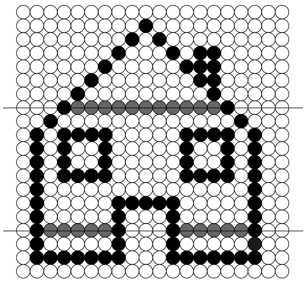
Notice that we have built three different sets of joined-up lines: one for the outline of the house, and two more, one for each window. Considering the bottom-left dot to be at $(0,0)$, they are, in fact, these sets of lines:

for the house outline
$(1,1)$—$(1,10)$—$(9,18)$—$(13,14)$—$(13,16)$—$(14,16)$—$(14,13)$—$(17,10)$—$(17,1)$—$(11,1)$—$(11,5)$—$(7,5)$—$(7,1)$—$(1,1)$

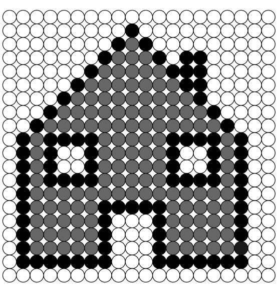
for the left window
$(3,10)$—$(6,10)$—$(6,7)$—$(3,7)$—$(3,10)$

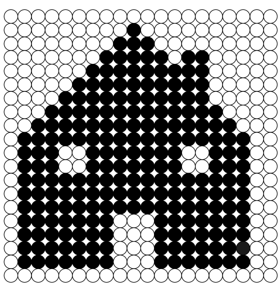
for the right window
$(12,10)$—$(15,10)$—$(15,7)$—$(12,7)$—$(12,10)$

Now, we can proceed to design a method to fill the shape. For each row of the image, we begin on the left, and proceed rightward pixel-by-pixel. If we encounter a black dot, we remember, and enter filling mode. In filling mode, we fill every dot black, until we hit another dot which was already black – then we leave filling mode. Seeing another already-black dot puts us back into filling mode, and so on.



In the image above, two lines have been highlighted. In the first, we enter the shape once at the side of the roof, fill across, and then exit it at the right hand side of the roof. In the second, we fill a section, exit the shape when we hit the doorframe, enter it again at the other doorframe – filling again – and finally exit it. If we follow this procedure for the whole image, the house is filled as expected.





The image on the left shows the new dots in grey; that on the right the final image. Notice that the windows and door did not cause a problem for our method.

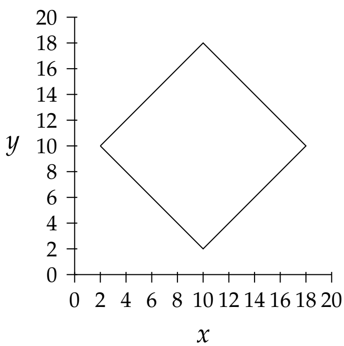
We have now looked at the very basics of how to convert descriptions of shapes into patterns of dots suitable for a printer or screen. In the next chapter, we will consider the more complicated shapes needed to draw good typefaces, which consist not only of straight lines, but also curves.

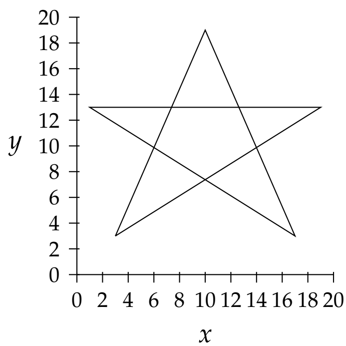
Problems

Solutions on page 147.

Grids for you to photocopy or print out have been provided on page 173. Alternatively, use graph paper or draw your own grids.

1. Give sequences of coordinates which may be used to draw these sets of lines.



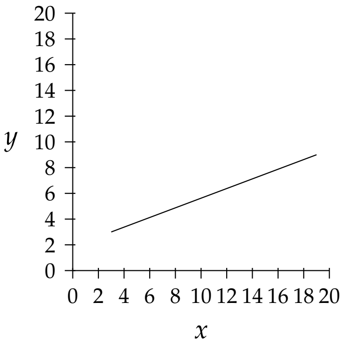


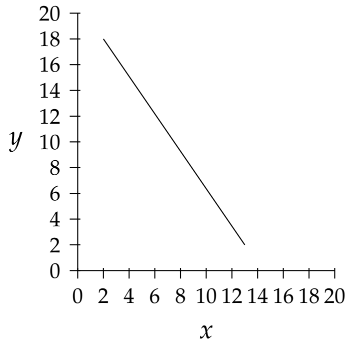
2. Draw these two sequences of coordinates on separate 20x20 grids, with lines between the points. What do they each show?

   (5,19)—(15,19)—(15,16)—(8,16)—(8,12)—(15,12)—(15,9)—(8,9)—(8,5)—(15,5)—(15,2)—(5,2)—(5,19)

   (0,5)—(10,10)—(5,0)—(10,3)—(15,0)—(10,10)—(20,5)—(17,10)—(20,15)—(10,10)—(15,20)—(10, 17)—(5, 20)—(10,10)—(0,15)—(3,10)—(0,5)

3. Given the following lines on 20x20 grids, select pixels to approximate them.





4. On 20x20 grids, choose pixels to fill in to approximate the following characters. Keep them in proportion to one another.

X t ? ©