Visual \parshape Editor for TeX

To try out the project, go to parshape.com or checkout the code at github.com/tmscer/parshape. See this file's source at github.com/tmscer/parshape/blob/master/doc/parshape.tex.

The problem

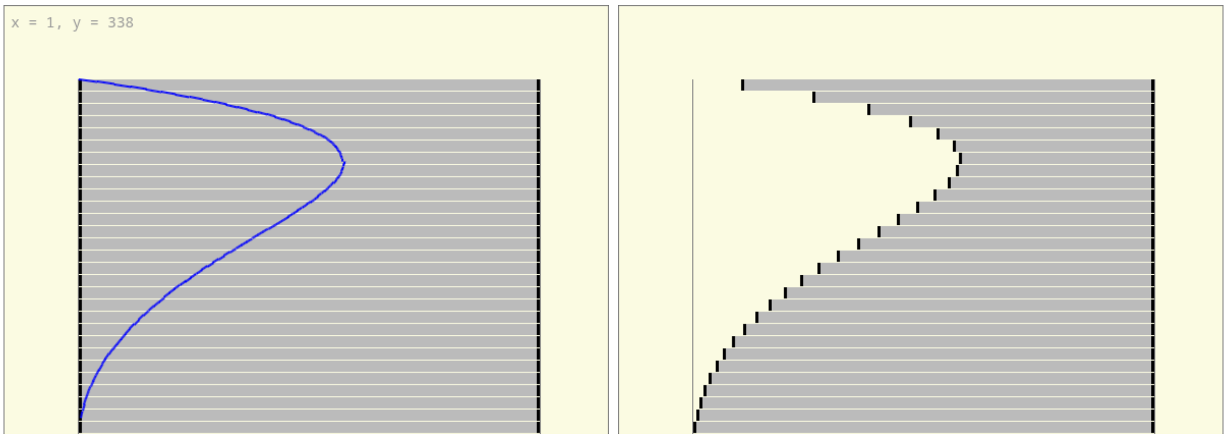
`\parshape` allows a TeX user to shape their paragraph. The following block of text was shaped using `\parshape 3 2cm 12cm 4cm 320pt 2cm 11cm`:

Notice the first line is moved by **2cm** to the right and is **12cm** long.
The second line is moved by **4cm** to the right and is **400pt** long.
The third line is moved by **2cm** to the right and is **11cm** long.

Creating just these 3 lines took the author quite a while. Trying to fit a paragraph around a more complex shape like an image, table or some weird shape is cumbersome and time consuming. Other software like MS Word help the user by being a visual tool where users get immediate feedback. TeX, of course, isn't that.

The solution

Below you can see two versions of the `<Canvas>` component of the web-based solution.



On the left is the *before* state. After the user *right clicks* at the bottom end of the blue Bézier curve. The new state shown on the right will emerge. A similar edit can be made to the right edge of every line of the paragraph and with other geometric shapes.

Configuration

The web app accepts several TeX parameters to mimick a page. To get their values, insert the left column of the following snippet into your source file. The right column has values of this document.

```
\tt{
pagewidth: \the\pagewidth \nl              pagewidth: 597.50787pt
pageheight: \the\pageheight \nl            pageheight: 845.04684pt
baselineskip: \the\baselineskip \nl        baselineskip: 13.0pt
lineskip: \the\lineskip \nl                lineskip: 1.0pt
hsize: \the\hsize \nl                      hsize: 231.84843pt
hoffset: \the\hoffset \nl                  hoffset: 56.9055pt
voffset: \the\voffset \nl                  voffset: 56.9055pt
pdfhorigin: \the\pdfhorigin \nl            pdfhorigin: 0.0pt
pdfvorigin: \the\pdfvorigin \nl            pdfvorigin: 0.0pt
}
```

An Example

Here's how a paragraph edited using the editor previously shown above can look like with some extra edits to the right edge of the paragraph:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis ullamcorper enim et
dignissim feugiat. Cras blandit tincidunt augue, id maximus velit
venenatis vel. Nam volutpat arcu in dignissim varius.
Integer faucibus tristique ex, in egestas
lectus posuere in. Maecenas at
tellus tortor. Sed euismod
id lorem dictum tincidunt.
Aenean lobortis non tellus
a efficitur. Aliquam id lectus
sit amet ex tincidunt euismod.
Quisque a condimentum arcu. Donec
et dignissim quam. Praesent hendrerit arcu
sit amet ultrices laoreet. Quisque eu ex ut ante
suscipit porttitor. Mauris erat libero, pulvinar non ante
sed, ultrices fringilla purus. Praesent lacus ligula, eleifend vel urna
ut, dictum porta metus. Nunc a urna eu ipsum ultrices commodo id non
purus. Fusce volutpat auctor dolor, ut fringilla risus ultrices eget. Duis
a aliquet dui. Nunc tempus auctor dapibus. Aliquam aliquam, metus at
blandit scelerisque, elit tellus sodales ligula, at auctor tellus urna nec
justo. Donec ullamcorper luctus nulla. Nunc rutrum, nulla id ultrices
molestie, quam ex placerat nisi, ac consequat nisi urna lacinia ante.
Ut pharetra aliquam lacus et dignissim. Quisque feugiat tristique
maximus. Vivamus eget auctor diam. Proin consectetur ante vel
sapien hendrerit finibus. Nunc mattis erat vitae sem eleifend, varius tempus
odio sollicitudin. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras
massa odio, dictum eget hendrerit ut, eleifend quis nulla. Aenean vehicula, nunc vel
ullamcorper scelerisque, magna urna feugiat purus, vel facilisis eros dui ac neque. Nam eget
mollis risus, ac gravida nunc. Suspendisse fermentum nibh ex, non suscipit mauris placerat id.
Nulla lobortis risus neque, non euismod neque posuere vel. Praesent a diam in nisi efficitur pulvinar.
Suspendisse cursus leo nec felis ultrices, et tempor tortor maximus. Duis tincidunt ultricies magna
non suscipit. Phasellus interdum dapibus varius. Aliquam pellentesque elit diam, vel molestie lectus
condimentum vel.

Supported Geometric Shapes and Their Math

The editor currently supports three geometric shapes: lines, half-circles and Bézier curves. Line has a variant called *snap-angle line* which forces the line's angle to be a multiple of 15°. Intersection with paragraph lines is always calculated for the middle of the paragraph according to primitive register `\baselineskip`.

The coordinate system has its origin in the left top corner of where the text can start. X-axis is positive to the right and y-axis is positive in the downward direction.

Line

This is the most simple geometric shape. Given two points, it is possible to get an equation of the form $y = ax + b,\ a,b \in \mathbb{R}$. Then, for every paragraph line, which are all horizontal of the form $y = \mathit{baselineskip} \cdot (i + 0.5)$, we can solve for x provided the drawn line isn't vertical. In all cases bounding box of the drawn line is checked in order to avoid the change of paragraph lines outside it.

Snap-angle Line

The equation remains the same as for the normal line. However, we have to recalculate the second placed point such that the line's angle is a multiple of 15° degrees or $\frac{\pi}{12}$ radians. We calculate the line's angle using using $\theta = \tan^{-1} a$ and round it. After that we *rotate* the second placed point using trigonometric function cos for x-coordinates sin for y-coordiantes.

Bézier Curve

For $n+1$ points, coordinates of *some* point on the Bézier curve would be:

$$\vec{B_t} = t^0(1-t)^n \cdot \mathbf{P_0} + t^1(1-t)^{n-1}\binom{n}{1} \cdot \mathbf{P_1} + \cdots + t^k(1-t)^{n-k}\binom{n}{k} \cdot \mathbf{P_k} + \cdots + t^n(1-t)^0\mathbf{P_n}$$

where $\mathbf{P_i}$ are placed points and $k \in \mathbb{N}_0$ is an iterator from 0 to n. Parameter t is unique for each point and it is from the interval $[0,1]$. The more accurate the approximation needs to be, the smaller t_{step} should be.

Intersection algorithm for Bézier curves does the following: for every paragraph line, intersect with all lines between adjacent points of the Bézier curve approximation and choose the leftmost or rightmost based on whether we're shaping the left edge or the right edge of the paragraph. Essentially we're reusing the previous algorithm for lines several times.

Half-circles

There are four cases to handle: 1) we're shaping the *left* edge with the *left* half a full circle, 2) we're shaping the *left* edge with the *right* half a full circle, 3) we're shaping the *right* edge with the *left* half a full circle, 4) we're shaping the *right* edge with the *right* half a full circle. That can be achieved by knowing how to intersect a full circle with a horizontal paragraph line and then choosing which edge of the paragraph to change and choosing whether to use the left or the right half-circle.

Equation for a circle with an origin (x_0, y_0) has the form

$$(x-x_0)^2 + (y-y_0)^2 = r^2$$

where r stands for radius. Given a paragraph line $y = c,\ c \in \mathbb{R}$, we can solve for x:

$$\begin{aligned}(x-x_0)^2 + (c-y_0)^2 &= r^2\\ x^2 - 2x_0 + x_0^2 + (c-y_0)^2 &= r^2\\ x^2 - 2x_0x + (x_0^2 + (c-y_0)^2 - r^2) &= 0\end{aligned}$$

That's nothing more than an ordinary quadratic equation. When determinant $D < 0$, the circle has no intersection with the current paragraph line. When determinant $D = 0$, there is only one intersection. When determinant $D > 0$, there are two intersections and we have to choose based on which half-circle we're using.