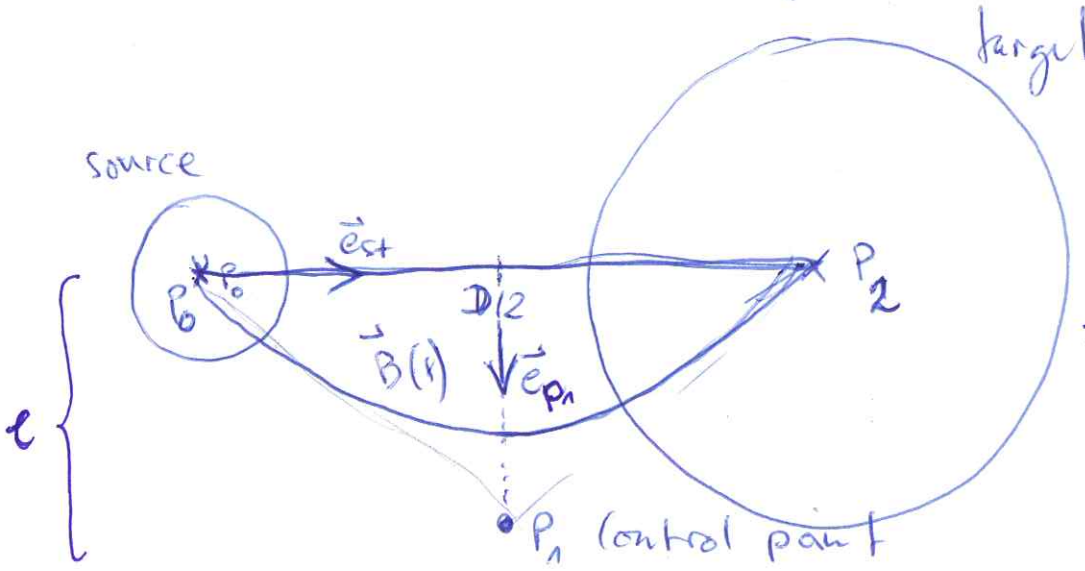
So basically the idea is to generate curved edges using Bézier curves (quadratic ones). Look at this picture



$$\vec{e}_{st} = \frac{\vec{P}_2 - \vec{P}_0}{\|\vec{P}_2 - \vec{P}_0\|}$$

$$\vec{e}_{P_1} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \vec{e}_{st}$$

$$\vec{P}_1 = \vec{P}_0 + \frac{D}{2}\vec{e}_{st} + t\,\vec{e}_{P_1}$$

First, we construct $\vec{e}_c$ and P_1, then we have the Bézier curve

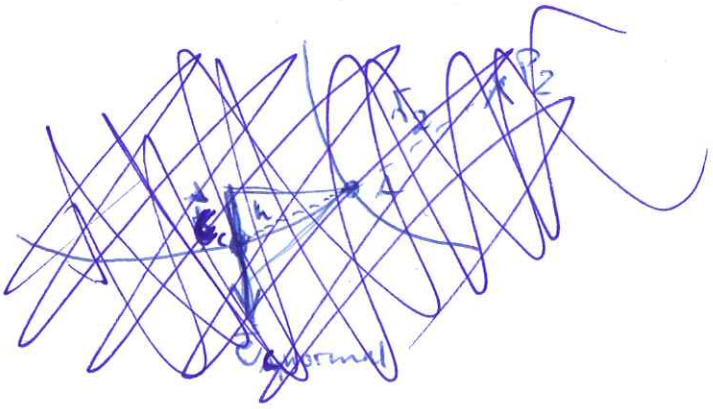
$$\vec{B}(t) = \vec{P}_1 + (1-t)^2(\vec{P}_0 - \vec{P}_1) + t^2(\vec{P}_2 - \vec{P}_1)$$

with derivative

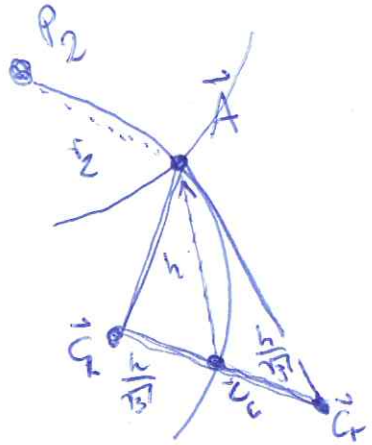
$$\vec{B}'(t) = 2(1-t)(\vec{P}_1 - \vec{P}_0) + 2t(\vec{P}_2 - \vec{P}_1)$$

with $t \in [0,1]$

We want to construct the arrow head like so



Beza The arrow head is drawn like this:



1) Use Newton-Raphson to find point A

⇒ find the zero of | initial guess $t = 0.9$

$$f_A(t) = r_2^2 - \|\vec{B}(t) - \vec{P}_2\|^2$$

with derivative

$$f_A'(t) = -2\,(\vec{B}(t) - \vec{P}_2) \cdot \vec{B}'(t)$$

⇒ finds t_A, then $\vec{A} = \vec{B}(t_A)$

2) Use Newton-Raphson to find point $\vec{C}_c$

find the zero of

$$f_c(t) = h^2 - \|\vec{B}(t) - \vec{A}\|^2$$

with derivative

$$f_c'(t) = -2\,(\vec{B}(t) - \vec{A}) \cdot \vec{B}'(t)$$

⇒ finds t_c, then $\vec{C}_c = \vec{B}(t_c)$

! good initial guess
! $t = 1 - \frac{r_2 + h}{L}$
! where L is the total
! length of the
! Bézier curve over $[0,1]$
! (see complicated function)

find normal vector at this point

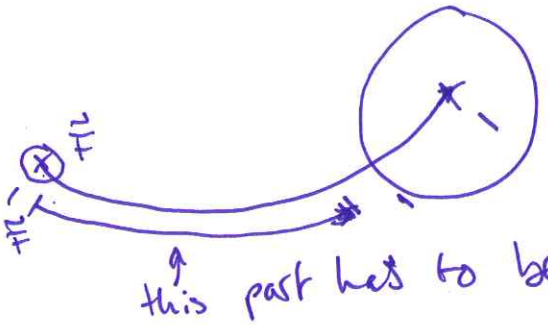
$$\vec{e}_c = \vec{B}'(t_c) \,/\, \|\vec{B}'(t_c)\|$$

then $\vec{e}_{c,\text{normal}} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \vec{e}_c$

and thus $\vec{c}_r = \frac{h}{\sqrt{3}}\,\vec{e}_{c,\text{normal}} + \vec{C}_c$

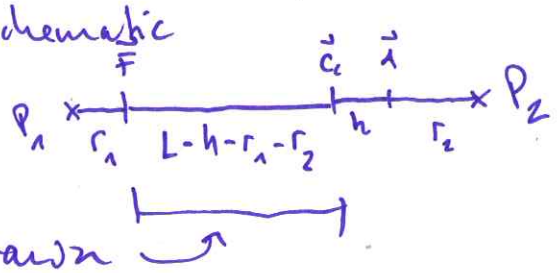
$\vec{c}_l = -\frac{h}{\sqrt{3}}\,\vec{e}_{c,\text{normal}} + \vec{C}_c$

• Now we have to draw the rest of the arrow



this part has to be drawn

schematic



simple but only approximate solution:

tell canvas to only draw this part by doing

ctx. moveTo(P_{0x}, P_{0y})

ctx. quadraticCurveTo(P_{1x}, P_{1y}, P_{2x}, P_{2y})

ctx. setLineDash([0, r_1, $L-h-r_1-r_2$, $h+r_2$])

• this is an approximation because if the line is curved, then the length of the parts in the circles and the arrow head are longer than r_1, r_2, or h, respectively

• one has to compute the total length L

• Alternatively, compute the point ~~of the source~~ $\vec{F}$ where the arrow begins at the source node, or begin at $\vec{P}_1$, then find a method to spline a Bézier curve between the start of the source arrow and point $\vec{c}_c$

⇒ turns out this works better, who would've thought