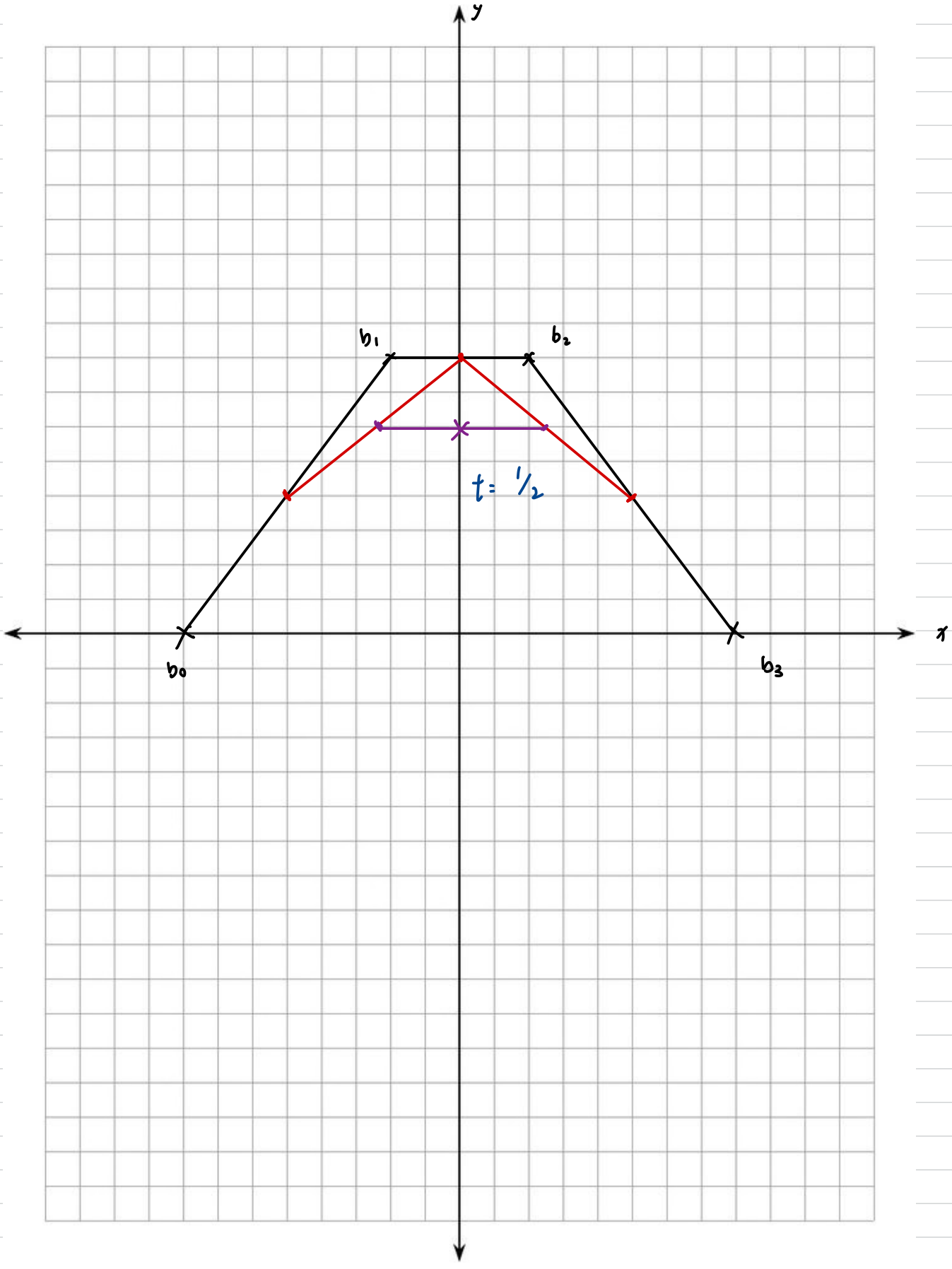
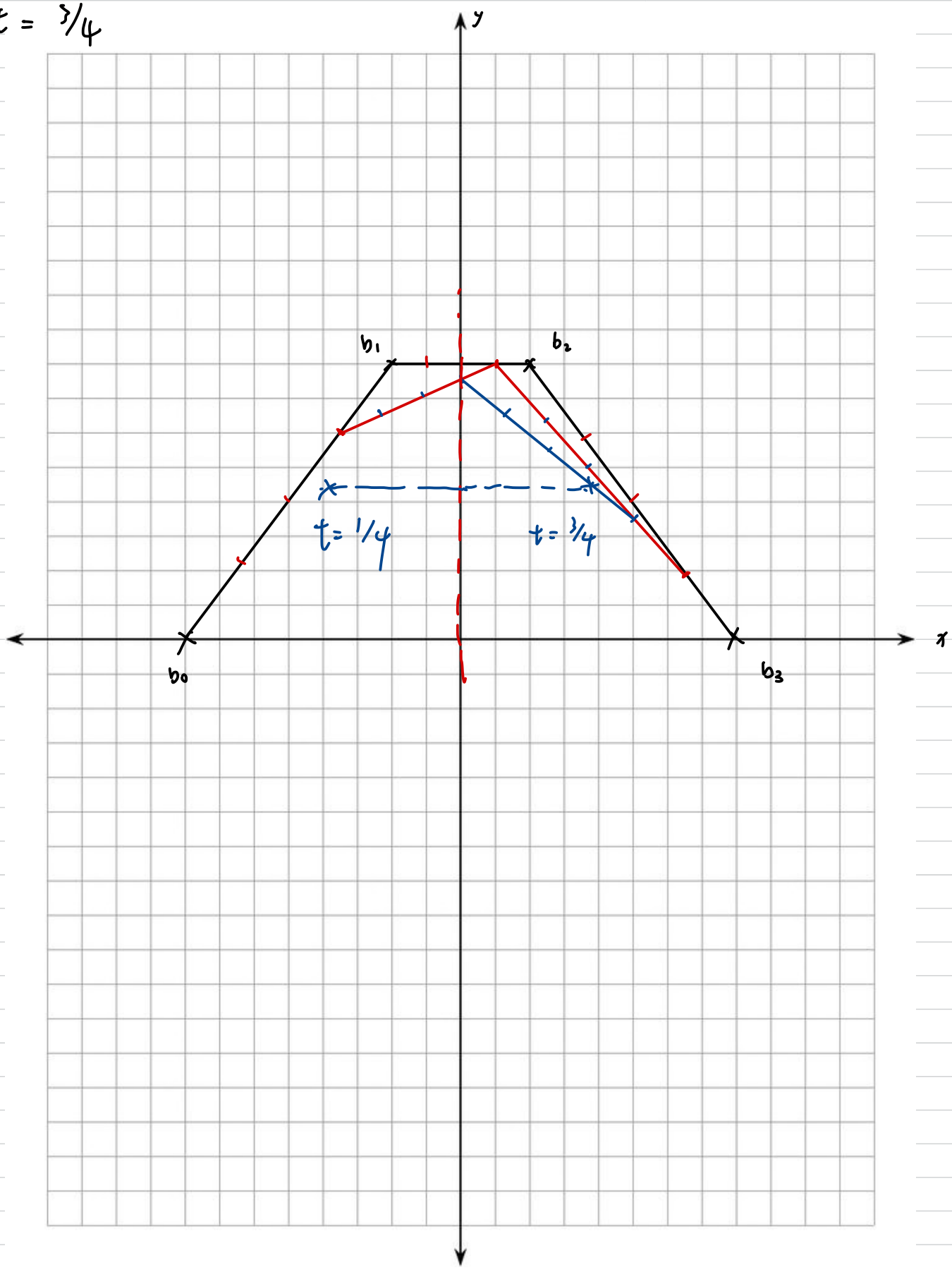
Assignment 2.1

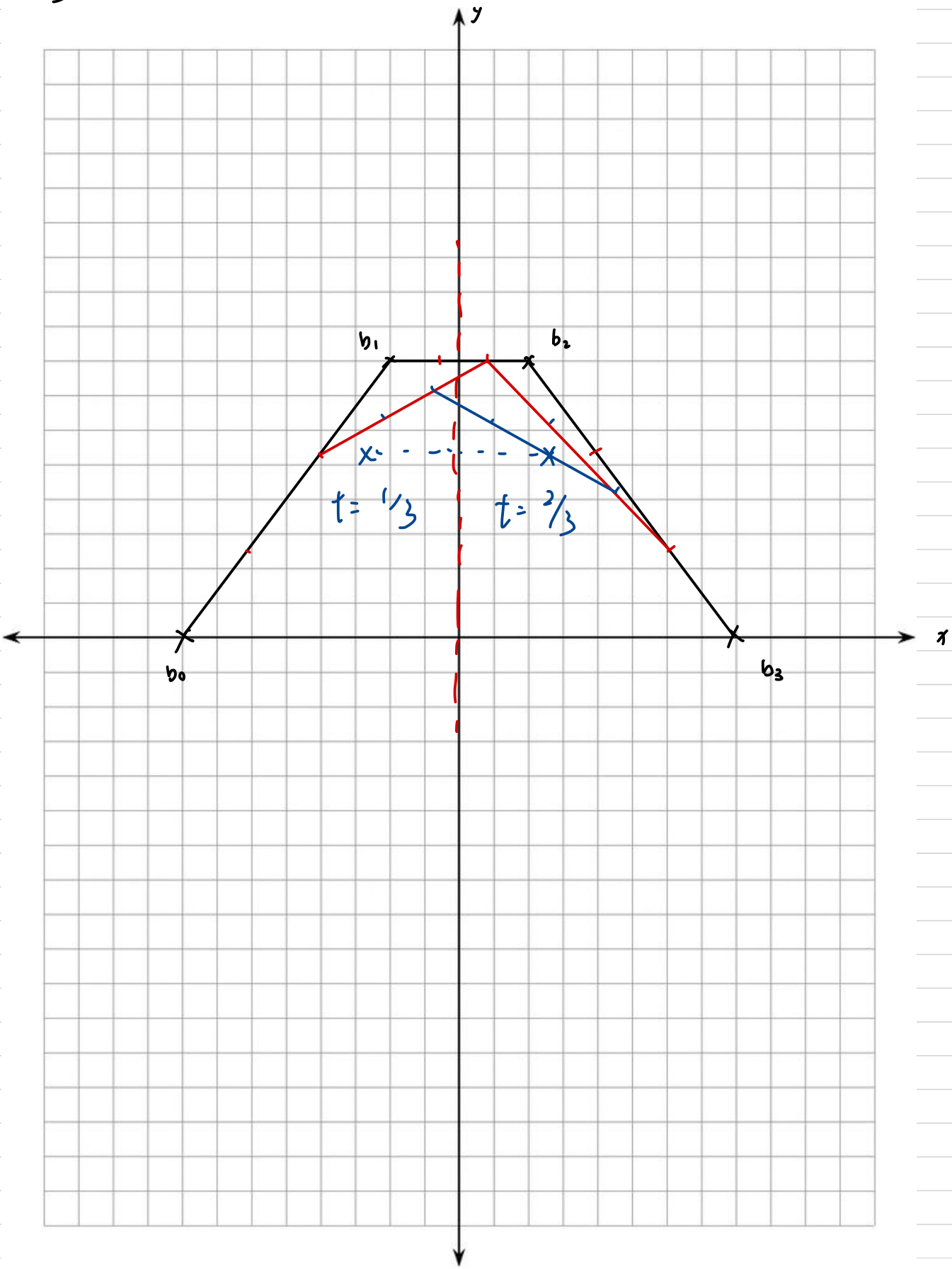
a) $\mathbf{b}_0 = \begin{pmatrix} -4 \\ 0 \end{pmatrix}, \mathbf{b}_1 = \begin{pmatrix} -1 \\ 4 \end{pmatrix}, \mathbf{b}_2 = \begin{pmatrix} 1 \\ 4 \end{pmatrix}, \mathbf{b}_3 = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$



for $t = 1/4$, $1 - t = 3/4$ since the control polygon is symmetrical, the point $t = 1/4$ should also by symmetrical to $t = 3/4$

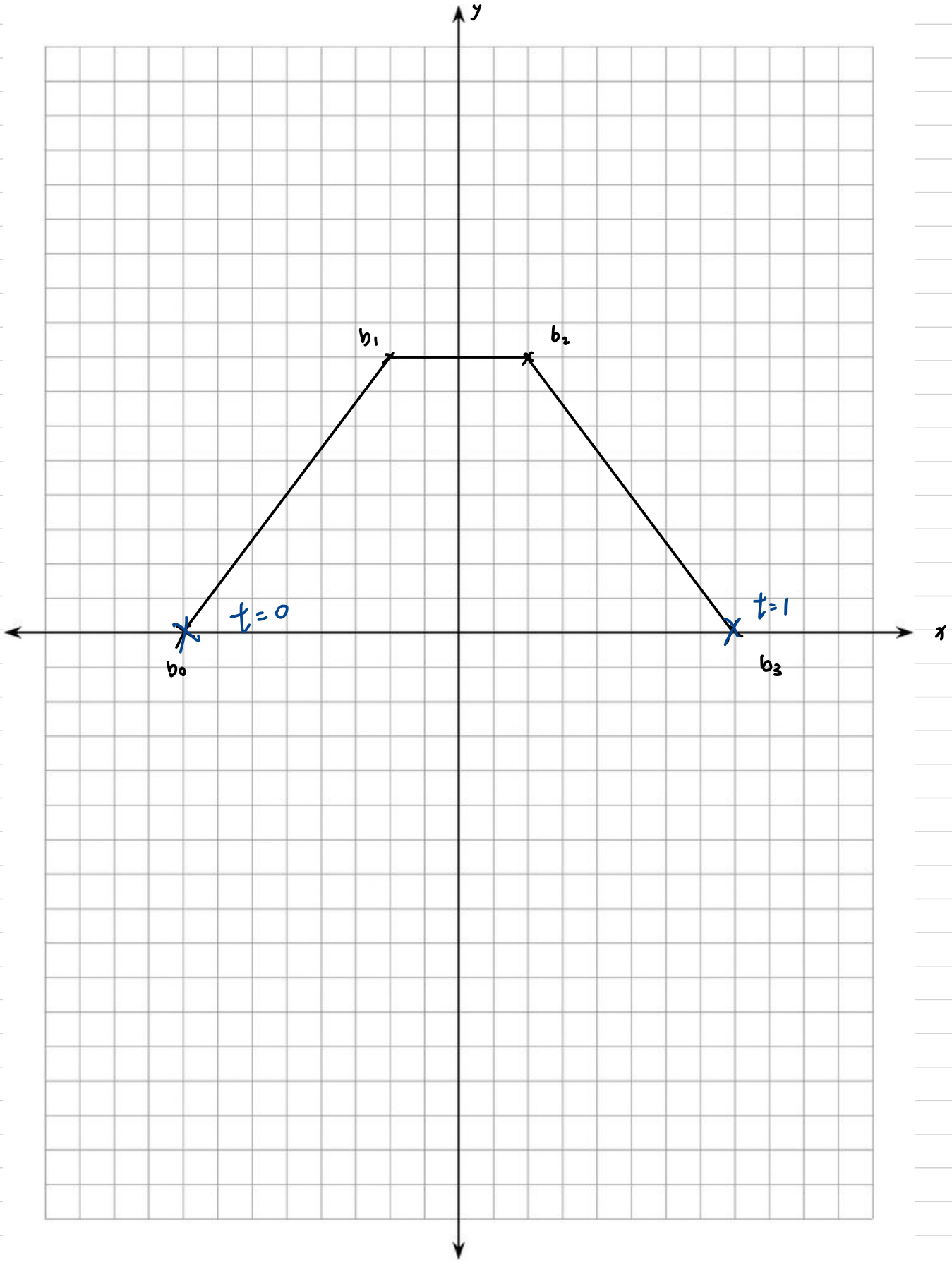


for $t = 1/3$, $1 - t = 2/3$ since the control polygon is symmetrical, the point $t = 1/3$ should also by symmetrical to $t = 2/3$

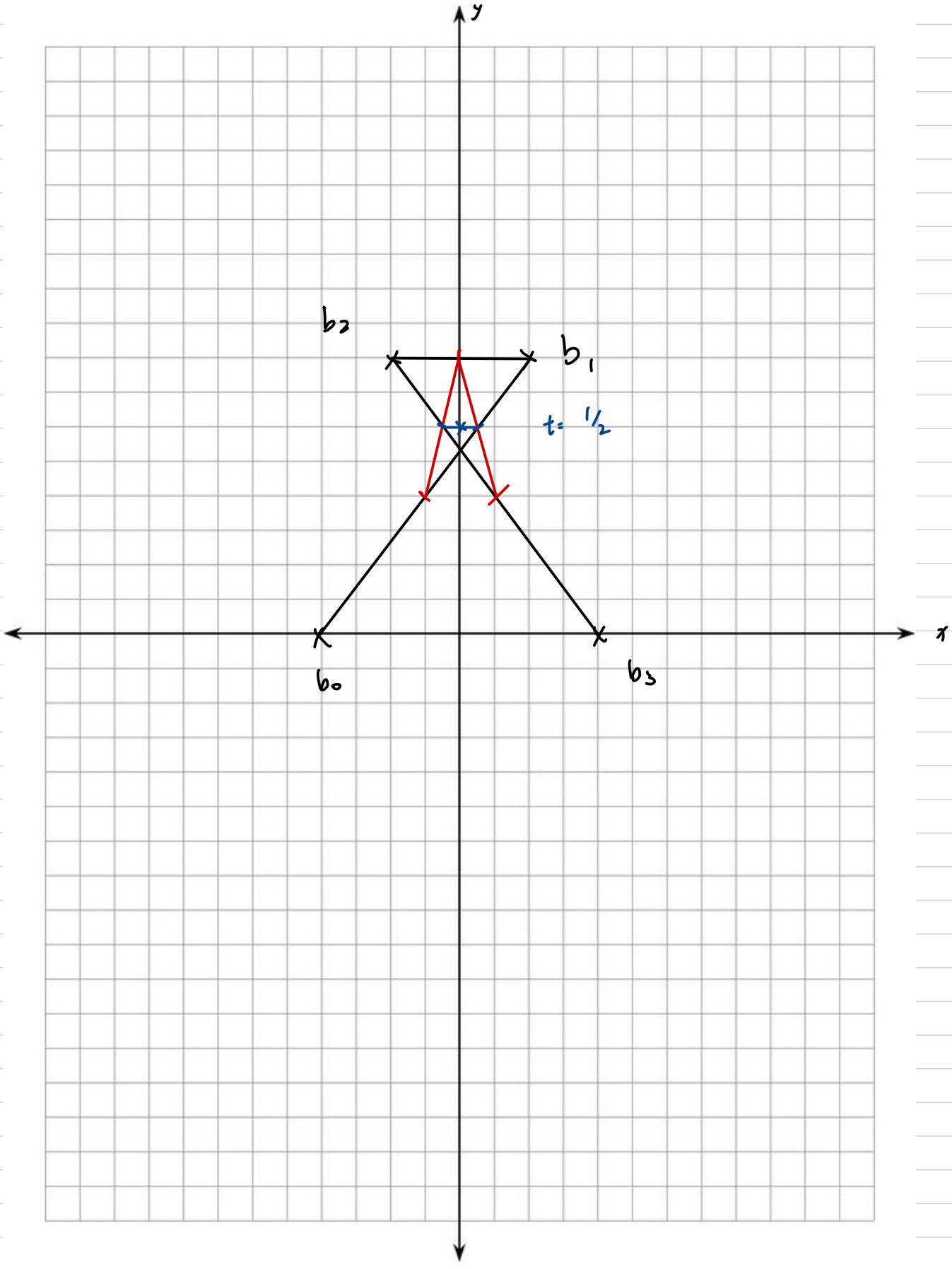


when $t = 0$ curve must go through b_0

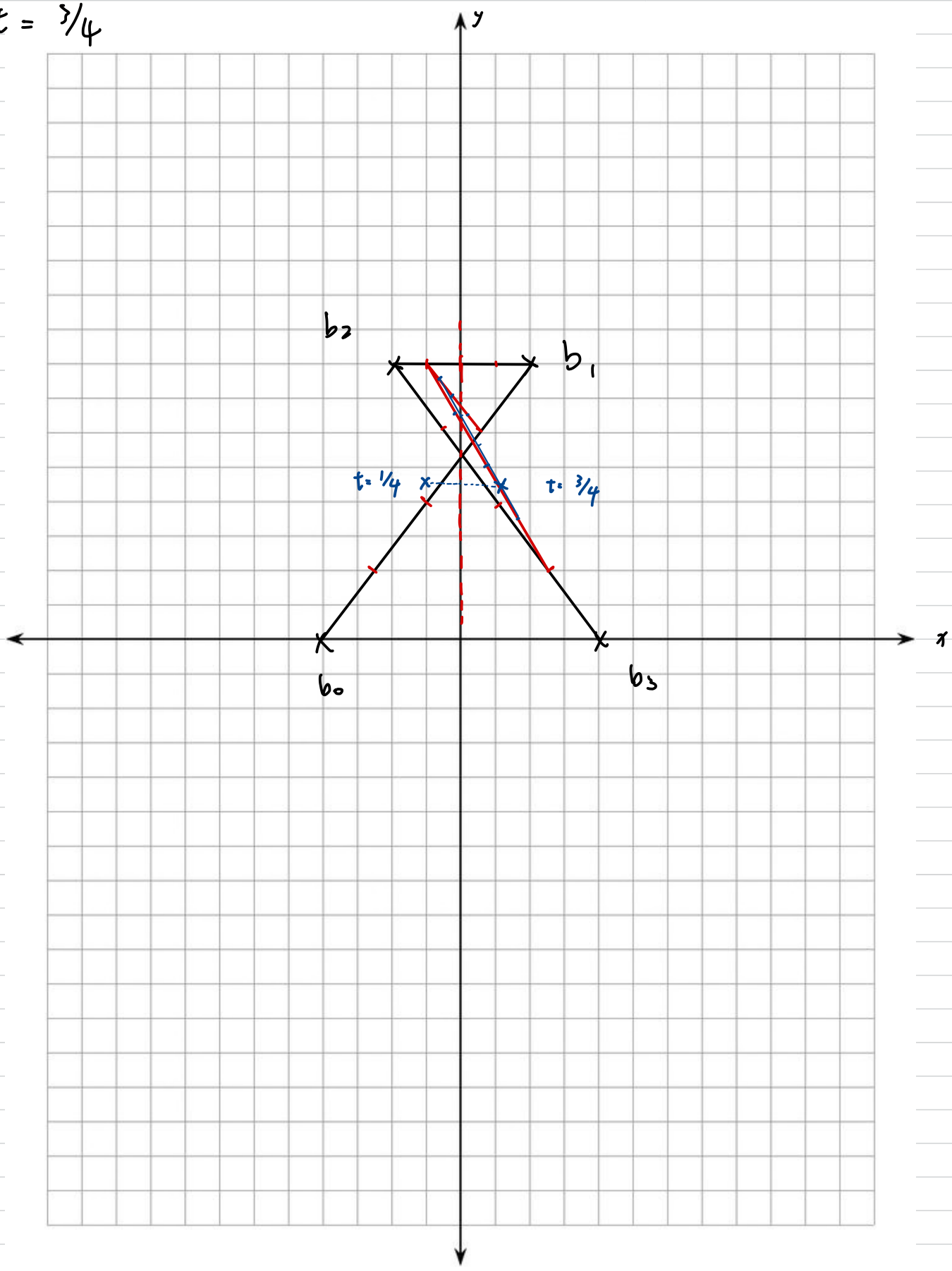
$t = 1$ curve must go through b_3



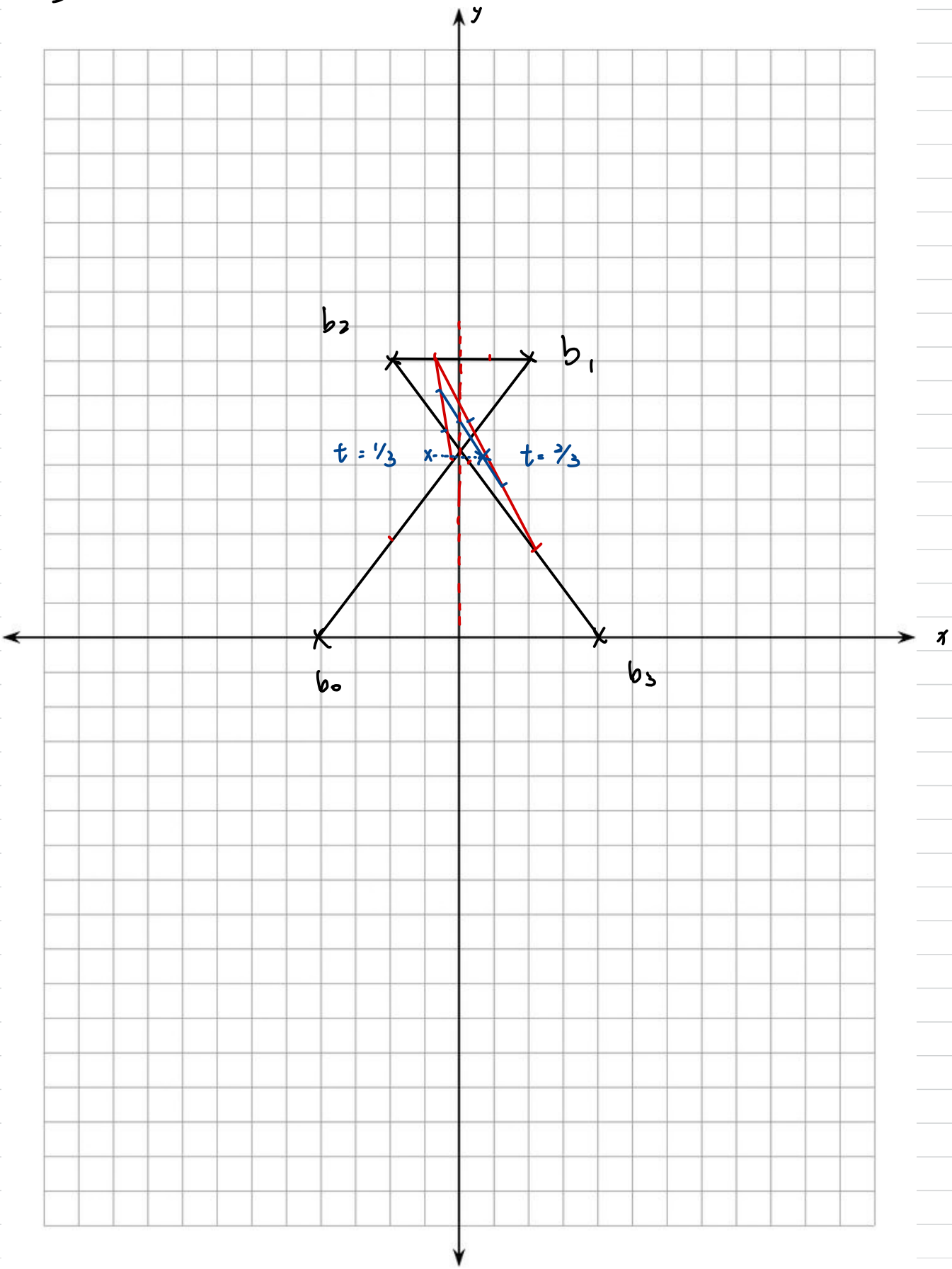
b) $\mathbf{b}_0 = \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \mathbf{b}_1 = \begin{pmatrix} 1 \\ 4 \end{pmatrix}, \mathbf{b}_2 = \begin{pmatrix} -1 \\ 4 \end{pmatrix}, \mathbf{b}_3 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$



for $t = 1/4$, $1 - t = 3/4$ since the control polygon is symmetrical, the point $t = 1/4$ should also by symmetrical to $t = 3/4$

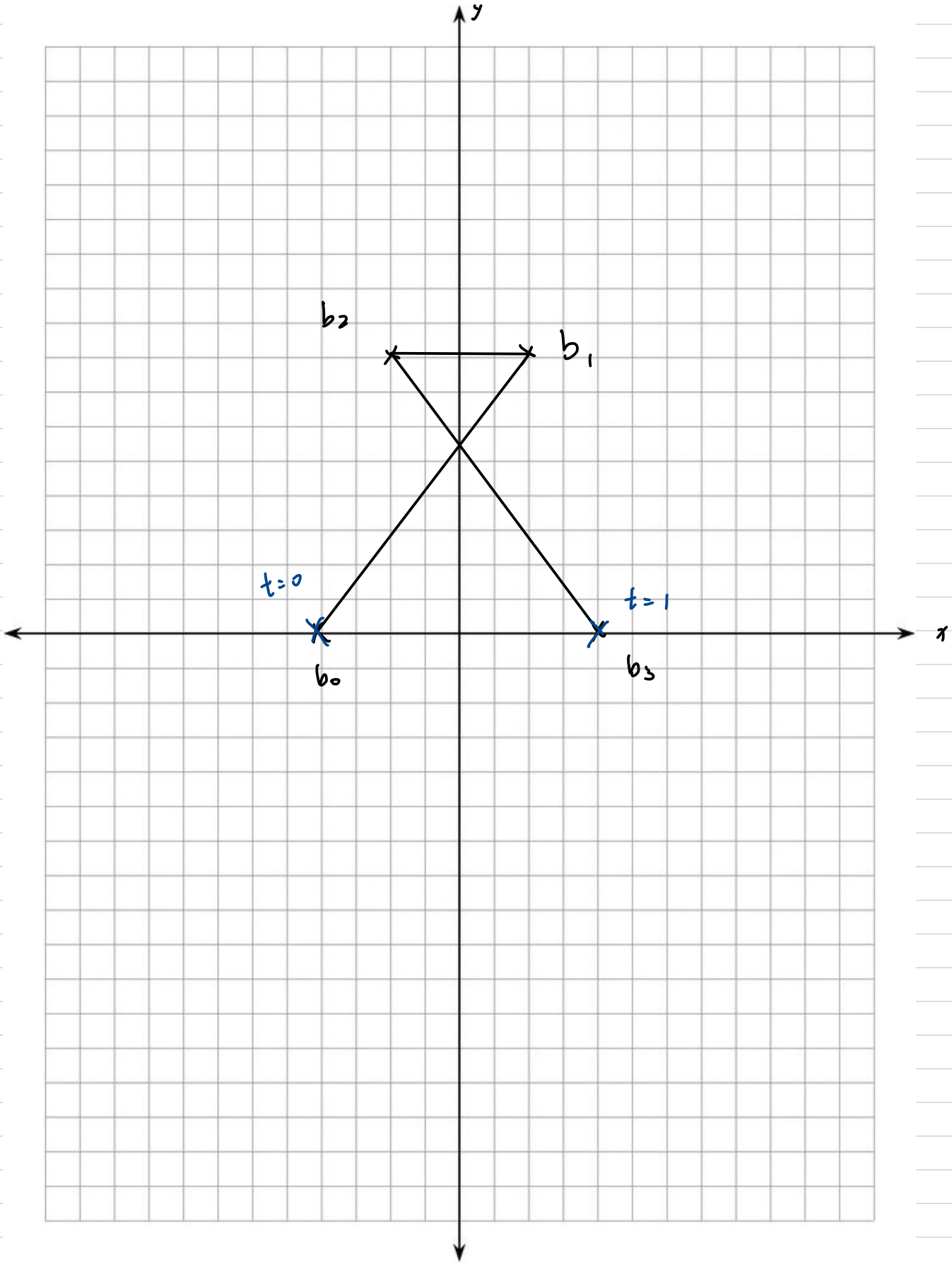


for $t = 1/3$, $1 - t = 2/3$ since the control polygon is symmetrical, the point $t = 1/3$ should also by symmetrical to $t = 2/3$

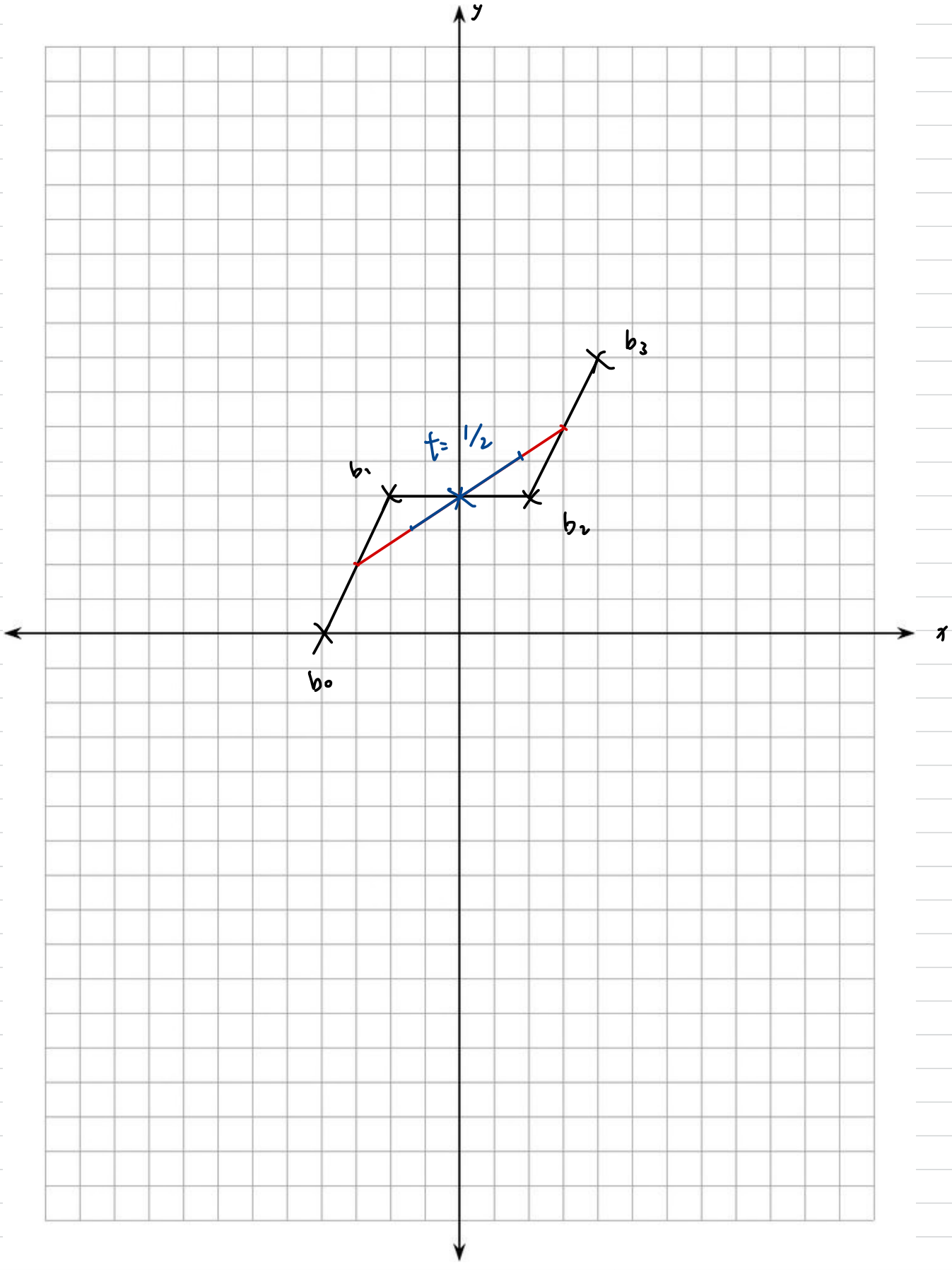


when $t = 0$ curve must go through b_0

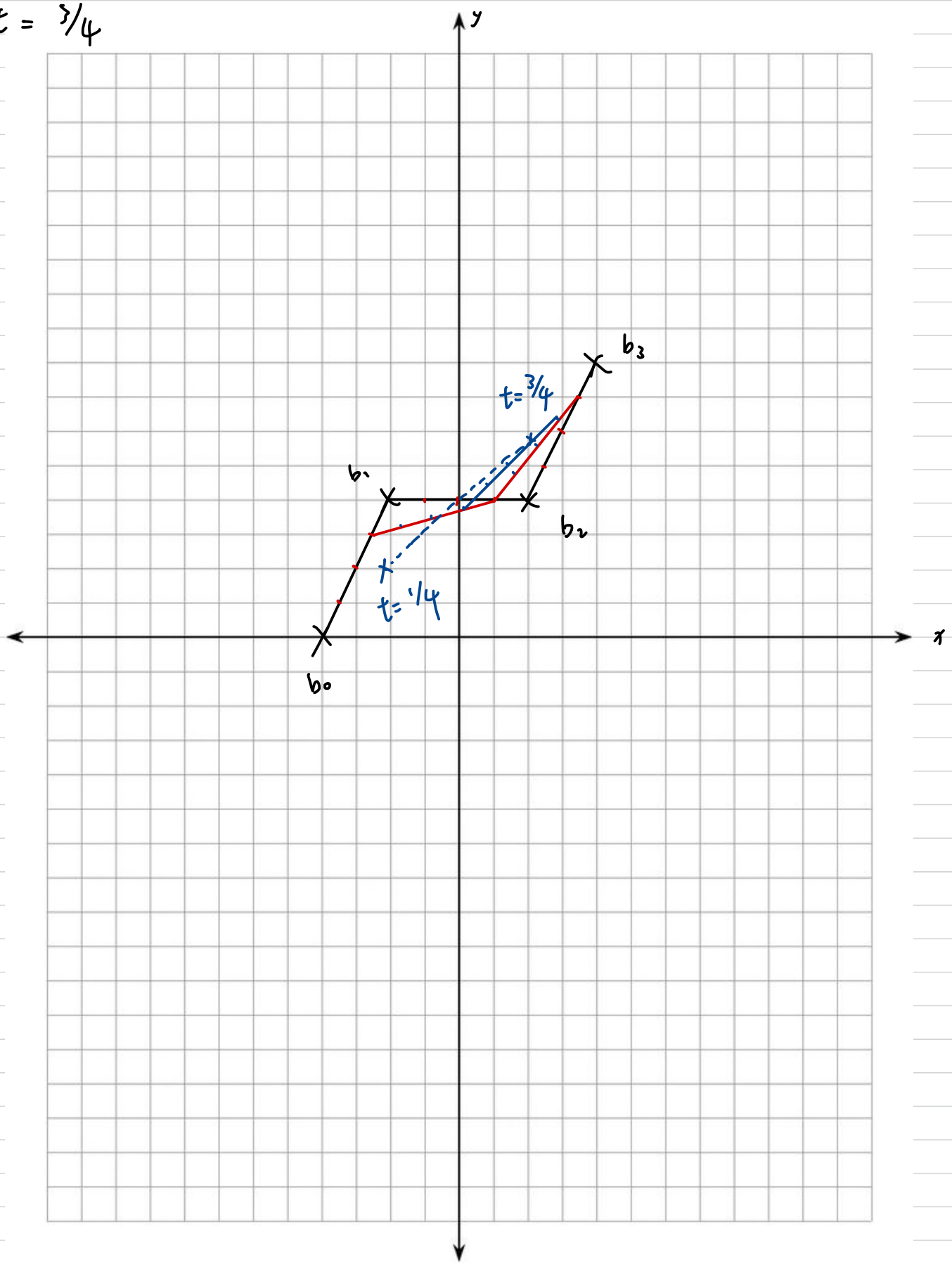
$t = 1$ curve must go through b_3



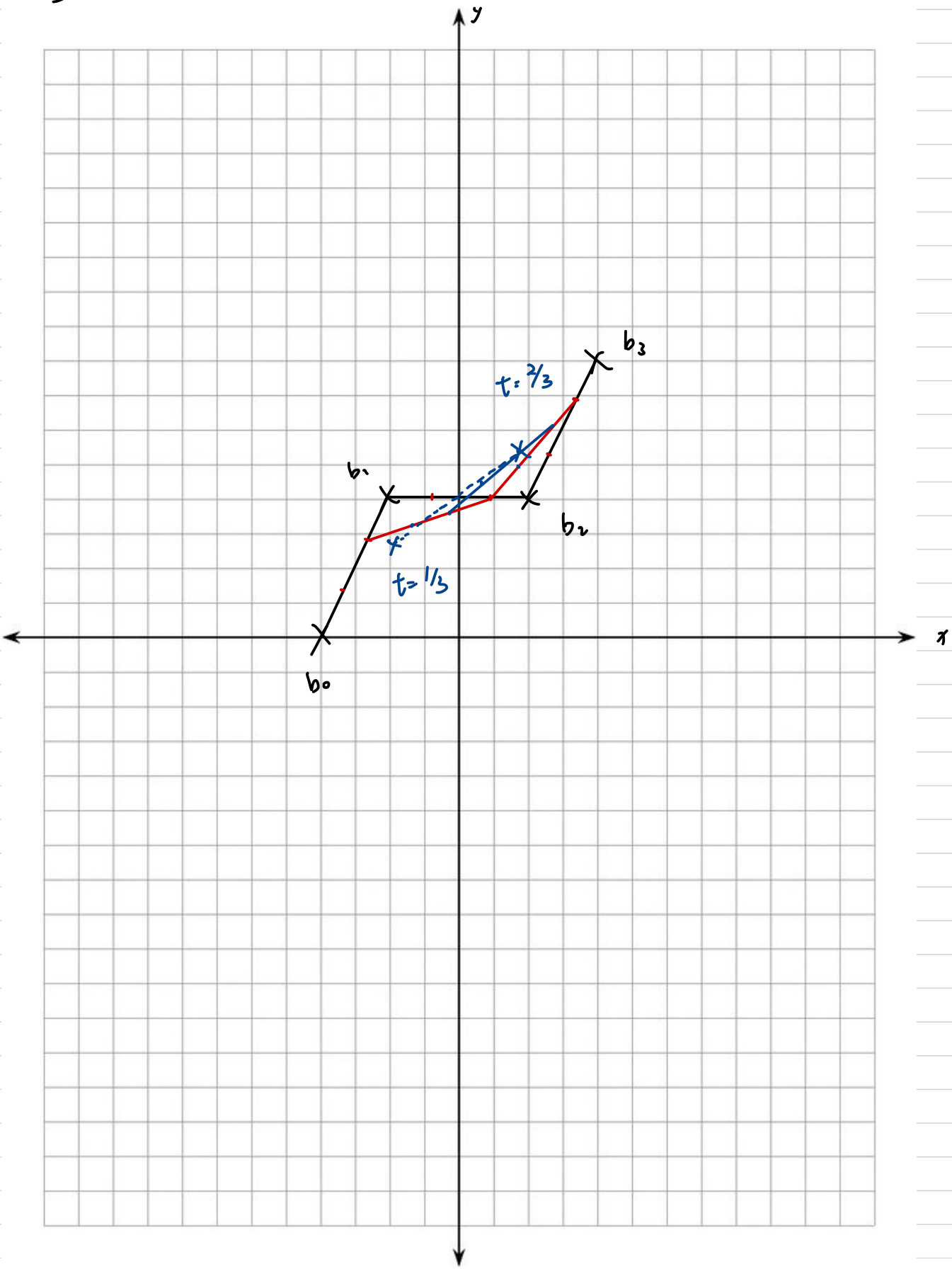
c) $\mathbf{b}_0 = \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \mathbf{b}_1 = \begin{pmatrix} -1 \\ 2 \end{pmatrix}, \mathbf{b}_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \mathbf{b}_3 = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$



for $t = 1/4$, $1 - t = 3/4$ since the control polygon is symmetrical, the point $t = 1/4$ should also by symmetrical to $t = 3/4$

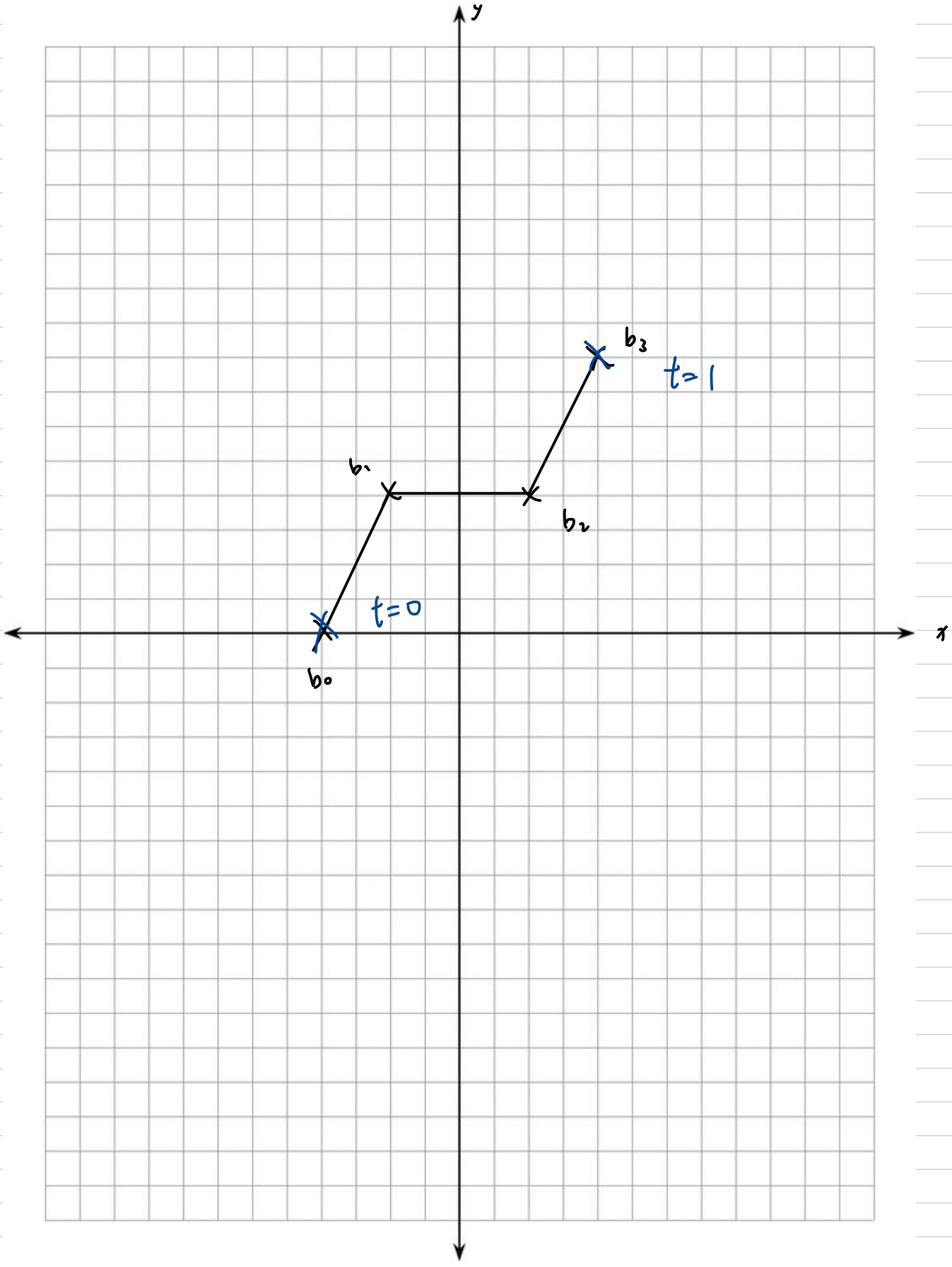


for $t = 1/3$, $1 - t = 2/3$ since the control polygon is symmetrical, the point $t = 1/3$ should also by symmetrical to $t = 2/3$



when $t = 0$ curve must go through b_0

$t = 1$ curve must go through b_3



Assignment 2.2

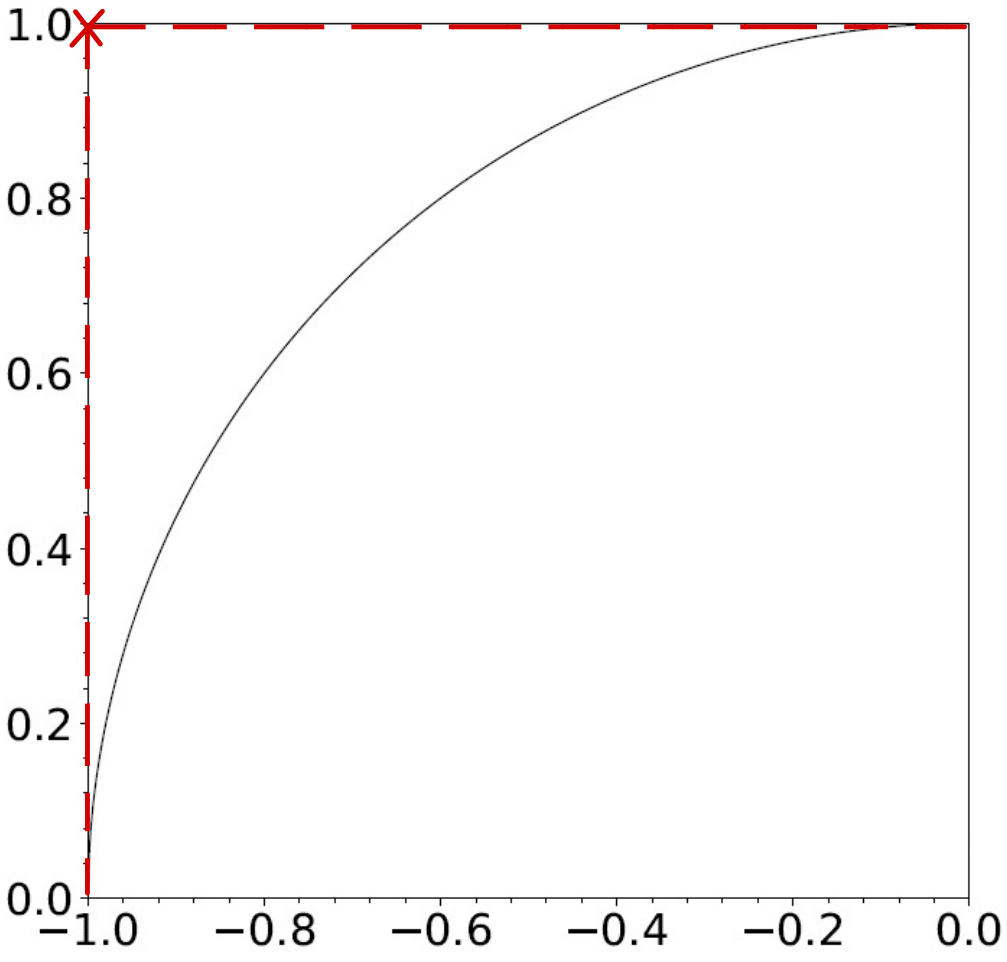
1a) Bezier curve go through b_0 and b_2

Thus, $b_0 = (-1, 0)^T$ $b_1 = (0, 1)^T$

at $t = 0$ Bezier curve must be tangential to the line $b_0 b_1$

at $t = 1$ Bezier curve must be tangential to the line $b_1 b_2$

Then if at $t = 0$ & $t = 1$ the curve be tangential to the circle. We can just take tagent line of the circle and the joint point should satisfy both requirement discussed above which is $b_1 = (-1, 1)^T$



$$F(t) = \sum_{i=0}^{m} b_i B_i^m(t)$$

3 points

$m = 2$

$$F(t) = b_0 B_0^2(t) + b_1 B_1^2(t) + b_2 B_2^2(t)$$

$$B_0^2(t) = \binom{2}{0} t^0 (1-t)^{2-0} = \frac{2!}{2!\,0!} t^0 (1-t)^2 = (1-t)^2$$

$$B_1^2(t) = \binom{2}{1} t^1 (1-t)^{2-1} = \frac{2!}{1!\,1!} t^1 (1-t)^1 = 2t(1-t)$$

$$B_2^2(t) = \binom{2}{2} t^2 (1-t)^{2-2} = \frac{2!}{0!\,2!} t^2 (1-t)^0 = t^2$$

$$F(t) = b_0 (1-t)^2 + b_1 2t(1-t) + b_2 t^2$$

$$= \binom{-1}{0} (1-t)^2 + \binom{-1}{1} 2t(1-t) + \binom{0}{1} t^2$$

$$= \begin{pmatrix} -1+2t-t^2 \\ 0 \end{pmatrix} + \begin{pmatrix} 2t^2-2t \\ 2t-2t^2 \end{pmatrix} + \begin{pmatrix} 0 \\ t^2 \end{pmatrix}$$

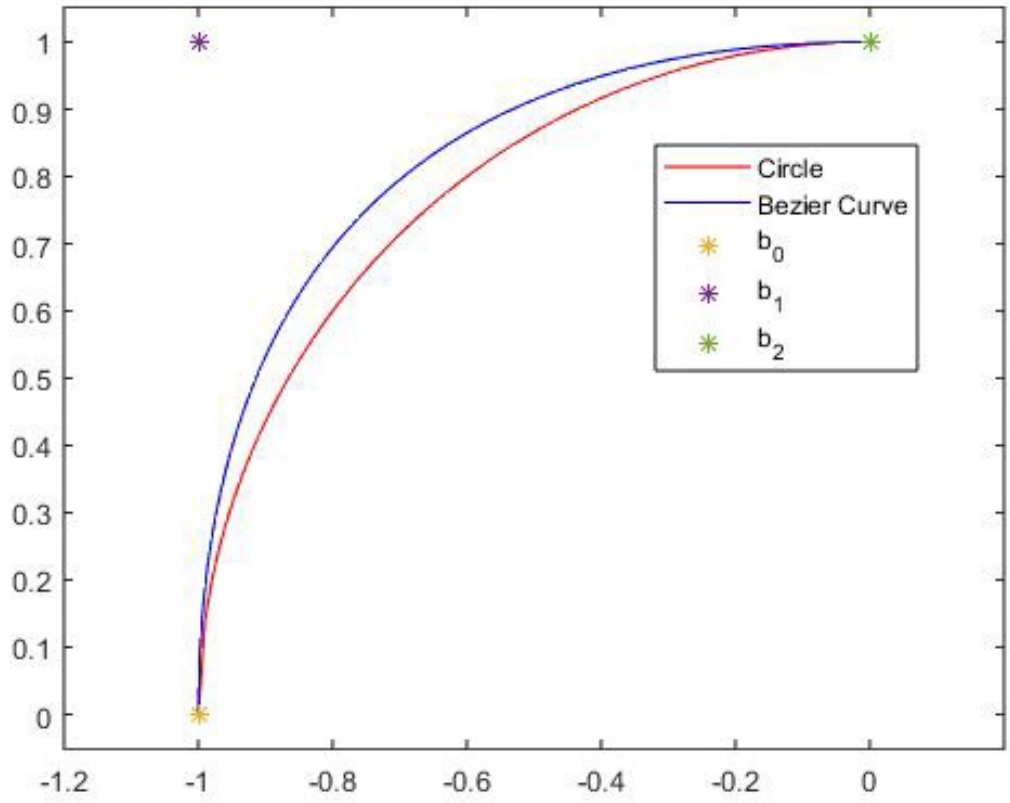
$$F(t) = \begin{pmatrix} t^2-1 \\ 2t-t^2 \end{pmatrix}$$

circle

$$C(t) = \begin{pmatrix} t \\ \sqrt{1-t^2} \end{pmatrix}$$

curves not the same

(b)




```
t = -1:0.01:0; %we sample our t from -1 to 0 with step = 0.01
xCircle = t; %our arc of the circle will begin in (-1, 0)
yCircle = sqrt(1-t.^2); %notice the '.' (dot) before ^ (to the power of).
%as t is an array/vector, we use dot to tell matlab the operation should be
%done _per_ element, not on a vector as a whole.

b1 = plot(xCircle,yCircle);
set(b1, 'color','red');

hold on
axis equal

tBezier = 0:0.01:1;
b0_x = -1;
b0_y =  0;

b1_x = -1;
b1_y =  1;

b2_x =  0;
b2_y =  1;


xBezier = b0_x*(1-tBezier).^2 + b1_x*2*tBezier.*(1-tBezier) + b2_x*tBezier.^2; %here you should write your own bezier equation
yBezier = b0_y*(1-tBezier).^2 + b1_y*2*tBezier.*(1-tBezier) + b2_y*tBezier.^2; %here you should write your own bezier equation

%xBezier = b1_x + (1-tBezier).^2*(b0_x - b1_x) + tBezier.^2*(b2_x - b1_x);
%yBezier = b1_y + (1-tBezier).^2*(b0_y - b1_y) + tBezier.^2*(b2_y - b1_y);

%xBezier = tBezier.^2-1;
%yBezier = -(1-tBezier).^2+1;

b2 = plot(xBezier,yBezier);
set(b2, 'color', 'blue');
plot(b0_x,b0_y,"*")
plot(b1_x,b1_y,"*")
plot(b2_x,b2_y,"*")
legend("Circle", "Bezier Curve","b_0","b_1","b_2")
xlim([-1.2,0.2])
hold off
```