Ilustrasi Pembuatan Kurva Bezier

en.wikipedia.org/wiki/Bézier_curve

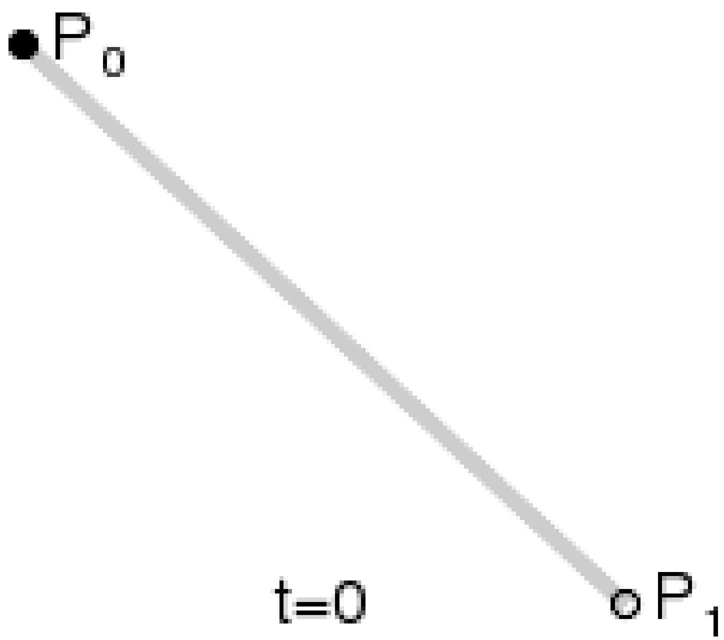
Linear Bézier curves

- Given points $\mathbf{P}_0$ and $\mathbf{P}_1$, a linear Bézier curve is simply a straight line between those two points. The curve is given by

$$\mathbf{B}(t) = \mathbf{P}_0 + t(\mathbf{P}_1 - \mathbf{P}_0) = (1-t)\mathbf{P}_0 + t\mathbf{P}_1 \ , t \in [0,1]$$

- and is equivalent to linear interpolation.

Ilustrasi: kurva linier



- The *t* in the function for a linear Bézier curve can be thought of as describing how far **B**(*t*) is from $\mathbf{P}_0$ to $\mathbf{P}_1$.
- For example when *t=0.25*, **B**(*t*) is one quarter of the way from point $\mathbf{P}_0$ to $\mathbf{P}_1$.
- As *t* varies from 0 to 1, **B**(*t*) describes a curved line from $\mathbf{P}_0$ to $\mathbf{P}_1$.

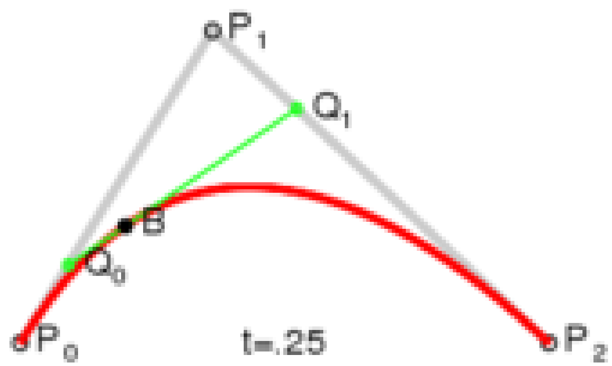
Quadratic Bézier curves

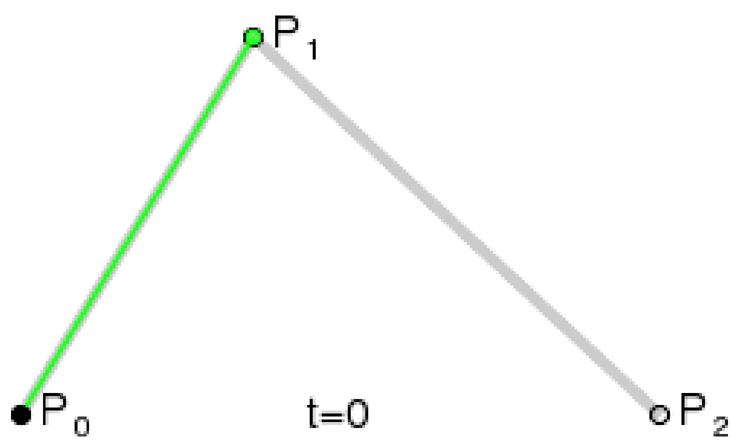
- A quadratic Bézier curve is the path traced by the function **B**(t), given points $\mathbf{P}_0$, $\mathbf{P}_1$, and $\mathbf{P}_2$,

$$\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2(1-t)t\mathbf{P}_1 + t^2\mathbf{P}_2 \ , t \in [0,1].$$

- A quadratic Bézier curve is also a parabolic segment.

Ilustrasi: kurva kuadratik





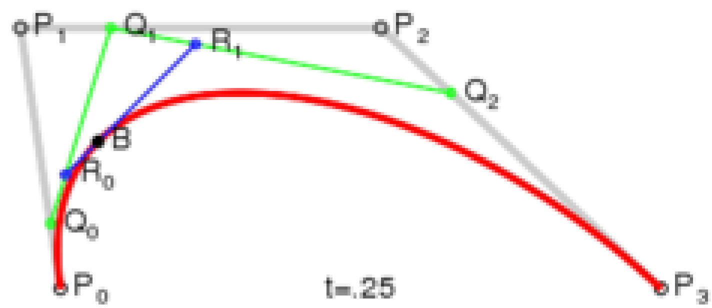
- For quadratic Bézier curves one can construct intermediate points $\mathbf{Q}_0$ and $\mathbf{Q}_1$ such that as t varies from 0 to 1:
- Point $\mathbf{Q}_0$ varies from $\mathbf{P}_0$ to $\mathbf{P}_1$ and describes a linear Bézier curve.
- Point $\mathbf{Q}_1$ varies from $\mathbf{P}_1$ to $\mathbf{P}_2$ and describes a linear Bézier curve.
- Point $\mathbf{B}(t)$ varies from $\mathbf{Q}_0$ to $\mathbf{Q}_1$ and describes a quadratic Bézier curve.

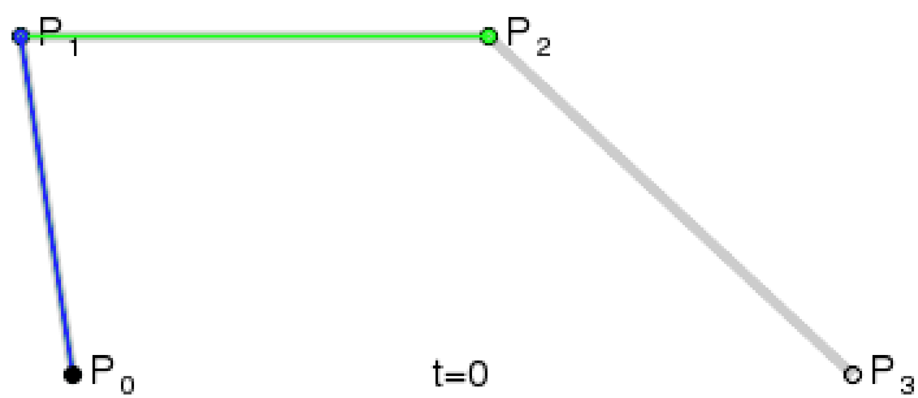
Cubic Bézier curves

- Four points $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$ in the plane or in three-dimensional space define a cubic Bézier curve.
- The curve starts at $\mathbf{P}_0$ going toward $\mathbf{P}_1$ and arrives at $\mathbf{P}_3$ coming from the direction of $\mathbf{P}_2$. Usually, it will not pass through $\mathbf{P}_1$ or $\mathbf{P}_2$; these points are only there to provide directional information. The distance between $\mathbf{P}_0$ and $\mathbf{P}_1$ determines "how long" the curve moves into direction $\mathbf{P}_2$ before turning towards $\mathbf{P}_3$.
- The parametric form of the curve is:

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3 \text{ , } t \in [0,1].$$

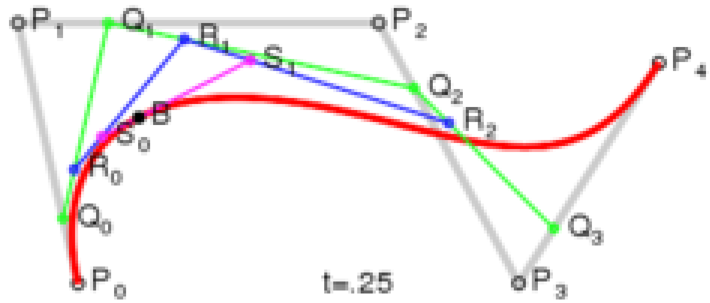
Ilustrasi : Kurva Cubic

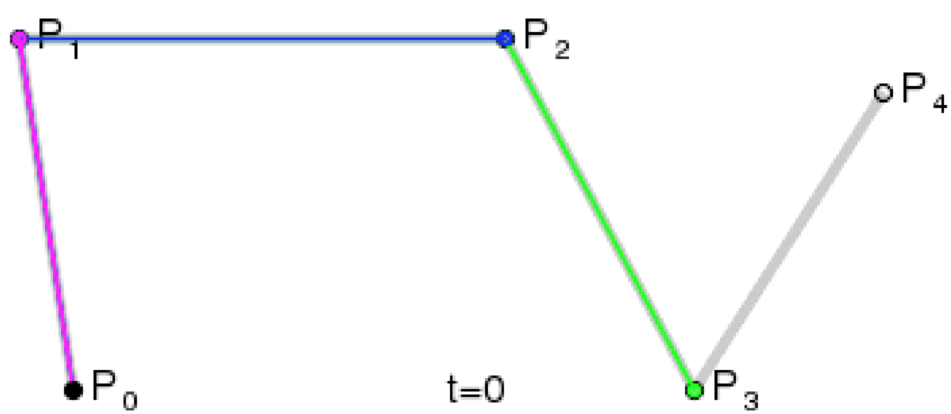




- For higher-order curves one needs correspondingly more intermediate points. For cubic curves one can construct intermediate points $\mathbf{Q}_0$,$\mathbf{Q}_1$ & $\mathbf{Q}_2$ that describe linear Bézier curves, and points $\mathbf{R}_0$ & $\mathbf{R}_1$ that describe quadratic Bézier curves

Ilustrasi: Fourth-order curve

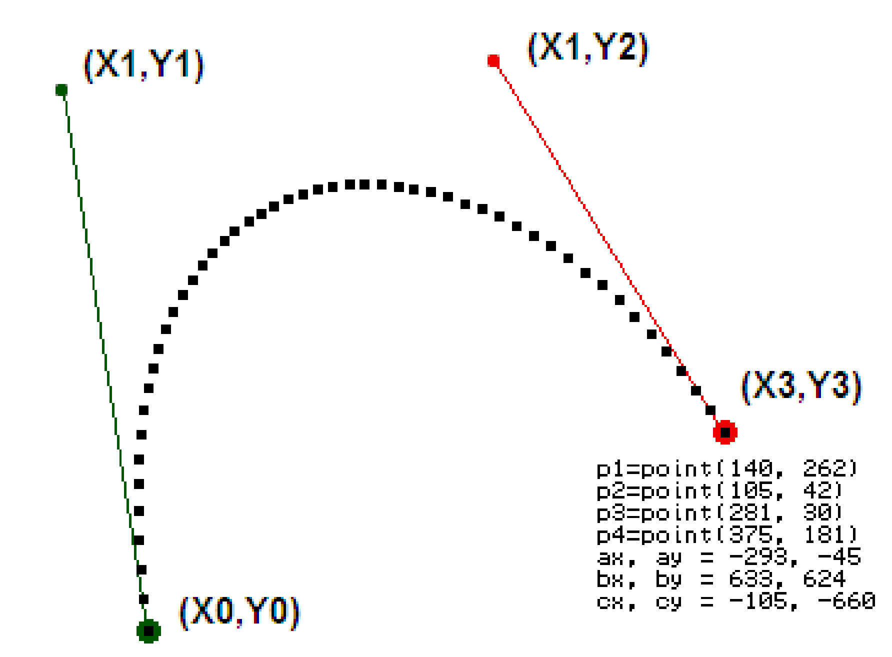




- For fourth-order curves one can construct intermediate points $\mathbf{Q}_0$, $\mathbf{Q}_1$, $\mathbf{Q}_2$ & $\mathbf{Q}_3$ that describe linear Bézier curves, points $\mathbf{R}_0$, $\mathbf{R}_1$ & $\mathbf{R}_2$ that describe quadratic Bézier curves, and points $\mathbf{S}_0$ & $\mathbf{S}_1$ that describe cubic Bézier curves

The Math Behind the Bezier Curve

www.moshplant.com/direct-or/bezier/math.html



- A cubic Bezier curve is defined by four points.
- Two are *endpoints*.
  - (x_0,y_0) is the *origin* endpoint.
  - (x_3,y_3) is the *destination* endpoint.
- The points (x_1,y_1) and (x_2,y_2) are *control points*.

- Two equations define the points on the curve. Both are evaluated for an arbitrary number of values of *t* between 0 and 1.
- As increasing values for *t* are supplied to the equations, the point defined by $x(t),y(t)$ moves from the origin to the destination.

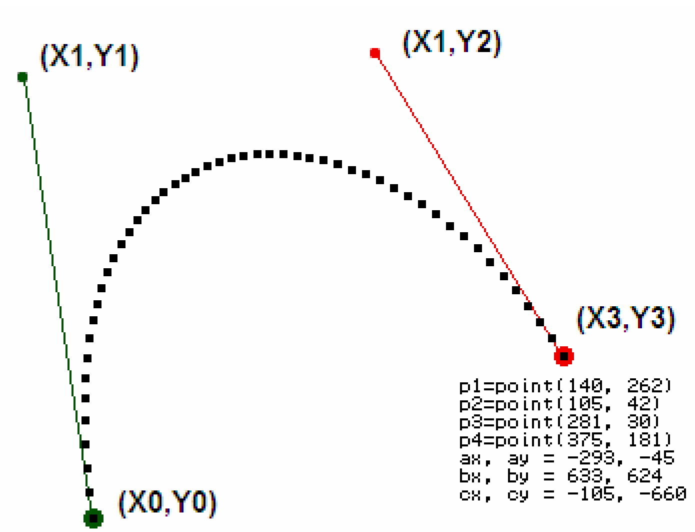
- This is how the equations are defined in Adobe's PostScript references.

$$x(t) = a_x t^3 + b_x t^2 + c_x t + x_0$$

$$x_1 = x_0 + c_x / 3$$
$$x_2 = x_1 + (c_x + b_x) / 3$$
$$x_3 = x_0 + c_x + b_x + a_x$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + y_0$$

$$y_1 = y_0 + c_y / 3$$
$$y_2 = y_1 + (c_y + b_y) / 3$$
$$y_3 = y_0 + c_y + b_y + a_y$$



- This method of definition can be reverse-engineered, so:

$$c_x = 3\ (x_1 - x_0)$$
$$b_x = 3\ (x_2 - x_1) - c_x$$
$$a_x = x_3 - x_0 - c_x - b_x$$

$$c_y = 3\ (y_1 - y_0)$$
$$b_y = 3\ (y_2 - y_1) - c_y$$
$$a_y = y_3 - y_0 - c_y - b_y$$