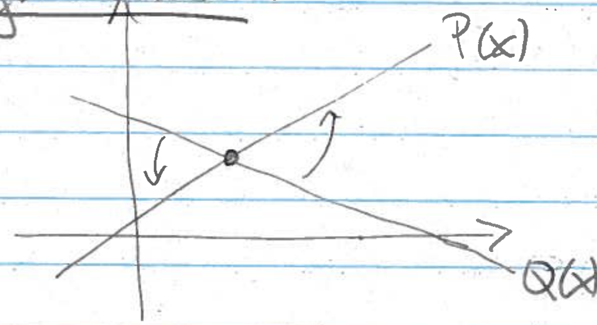
Lec. 7 Zeros of polynomials.

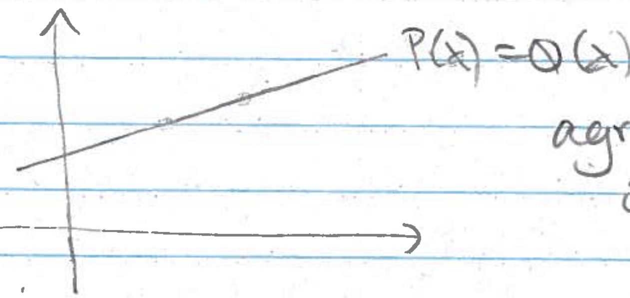
Identical Polynomials.

n=1



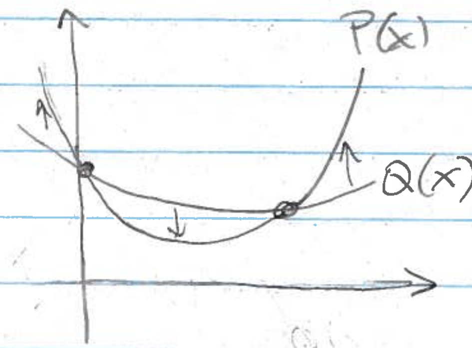
agree @ one pt

swiveling Q around as shown to make it agree with P @ a second pt, makes the lines coincident.



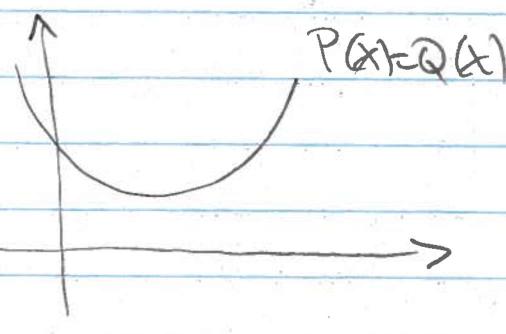
agree @ infinity of pts.

n=2



agree @ two pts

deforming Q in the way shown, in an effort to make it agree with P @ a third pt, makes it agree everywhere!



agree @ infinity of pts.

Horner's Method

Given

$$p(x) = a_0 + a_1 x + \cdots + a_n x^n,$$

we wish to evaluate $p(x_0)$. To do this we define a new seq. of constants:

$$b_n = a_n$$

$$b_{n-1} = a_{n-1} + b_n x_0$$

$$\vdots$$

$$b_0 = a_0 + b_1 x_0.$$

Then:

$$p(x_0) = b_0.$$

Pf of Horner's Method

$$p(x_0) = a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n$$

$$= a_0 + (\cdots(a_{n-2} + (a_{n-1} + \underbrace{a_n}_{} x_0) x_0) x_0 \cdots) x_0$$

$$= a_0 + (\cdots(a_{n-2} + (\underbrace{a_{n-1} + b_n x_0}_{}) x_0) x_0 \cdots) x_0$$

$$= a_0 + (\cdots(\underbrace{a_{n-2} + b_{n-1} x_0}_{}) x_0) \cdots) x_0$$

$$= a_0 + (\cdots b_{n-2} x_0) \cdots) x_0$$

$$\vdots$$

$$= a_0 + b_1 x_0$$

$$= b_0$$

But:

$$p(x) = (x - x_0)Q(x) + R(x).$$

$$\Rightarrow \quad p(x_0) = R(x_0)$$

Thus, Horners Method is a way to compute the remainder of polynomial division.

Another method for polynomial division is "synthetic division".

Example

$$P(x) = 2x^3 - 6x^2 + 2x - 1$$

Compute $P(3)$.

Sol^n

Horner's Method:

$$b_3 = a_3 = 2$$

$$b_2 = a_2 + b_3 x_0 = -6 + 2(3) = 0$$

#ops $= 2(3) = 6$

$$b_1 = a_1 + b_2 x_0 = 2 + 0(3) = 2$$

$$b_0 = a_0 + b_1 x_0 = (-1) + 2(3) = 5.$$

Synthetic division:

$$\begin{array}{c|cccc}
x_0 & x^3 & x^2 & x^1 & x^0 \\
3 & \boxed{2} & \boxed{-6} & \boxed{2} & \boxed{-1} \\
 & \downarrow & 6 & 0 & 6 \\
 & 2 & 0 & 2 & 5
\end{array}$$

#ops $= 2(3) = 6$

$$P(x) = (2x^2 + 0x + 2)(x - 3) + 5$$

$$\Rightarrow P(3) = 5.$$

Brute-force Method:

$$2(3)^3 - 6(3)^2 + 2(3) - 1$$

$$= 54 - 54 + 6 - 1$$

$$= 5.$$

$$\#ops = \underbrace{3 + 2 + 1}_{\text{muls.}} + \underbrace{3}_{\text{adds.}} = 9$$

Thus Horner's Method is an iterative and computationally efficient way to compute polynomials.