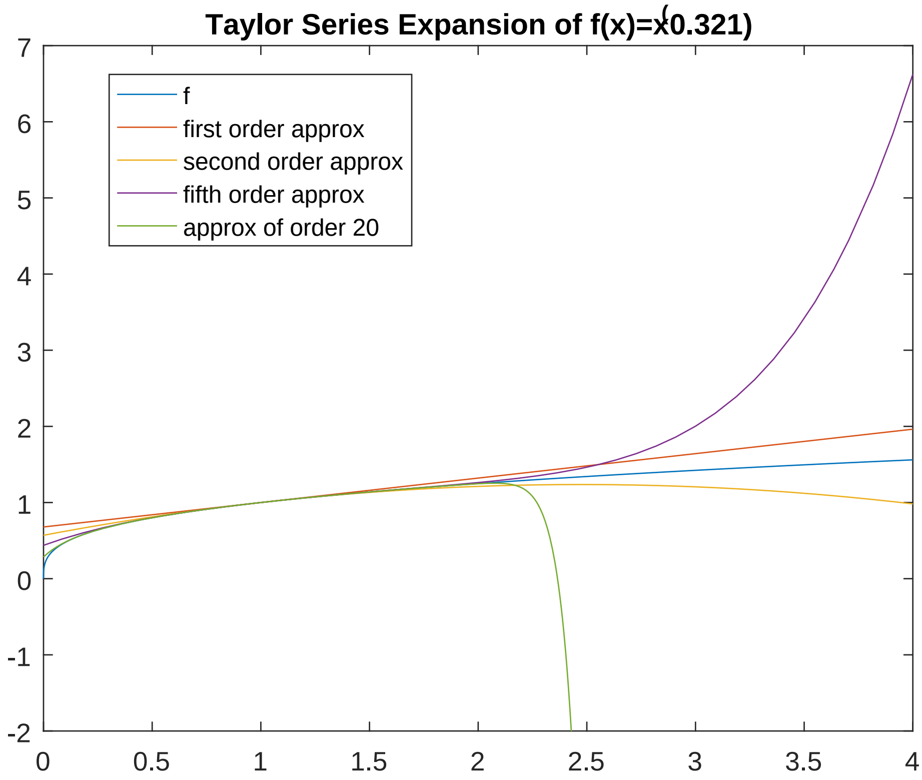
Quantitative Macro HW 2

Alexander Wurdinger

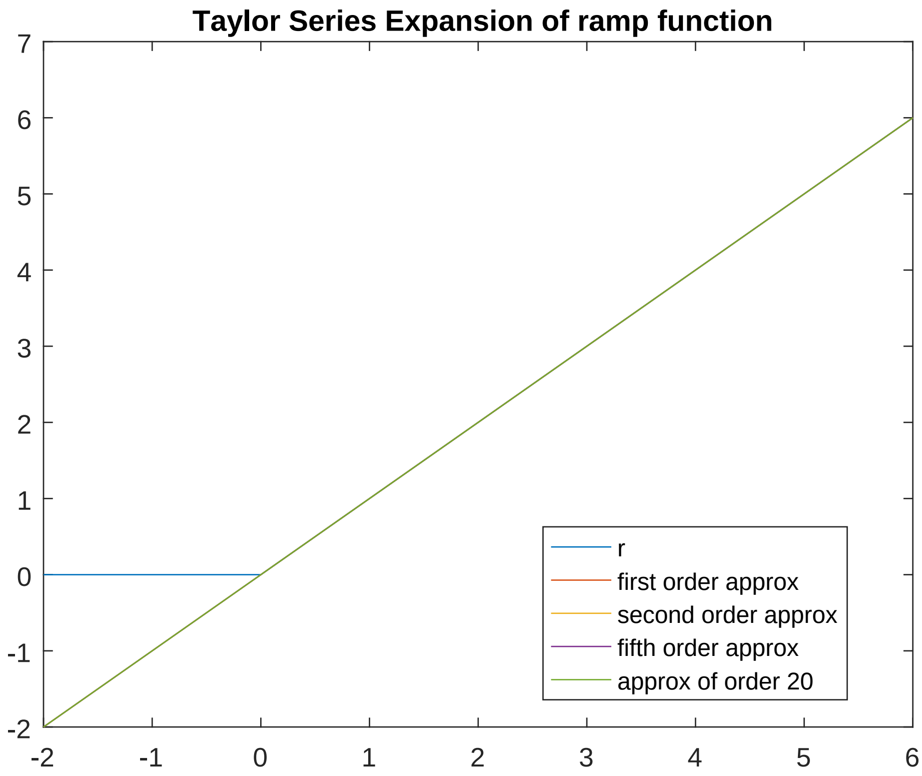
September 30, 2019

Q 1



Approximating the function $f(X) = x^{0.321}$ with taylor series around $\hat{x} = 1$ it shows that using smaller order approximations gives more accurate results than higher order ones in the given domain (0,4).

Q 2



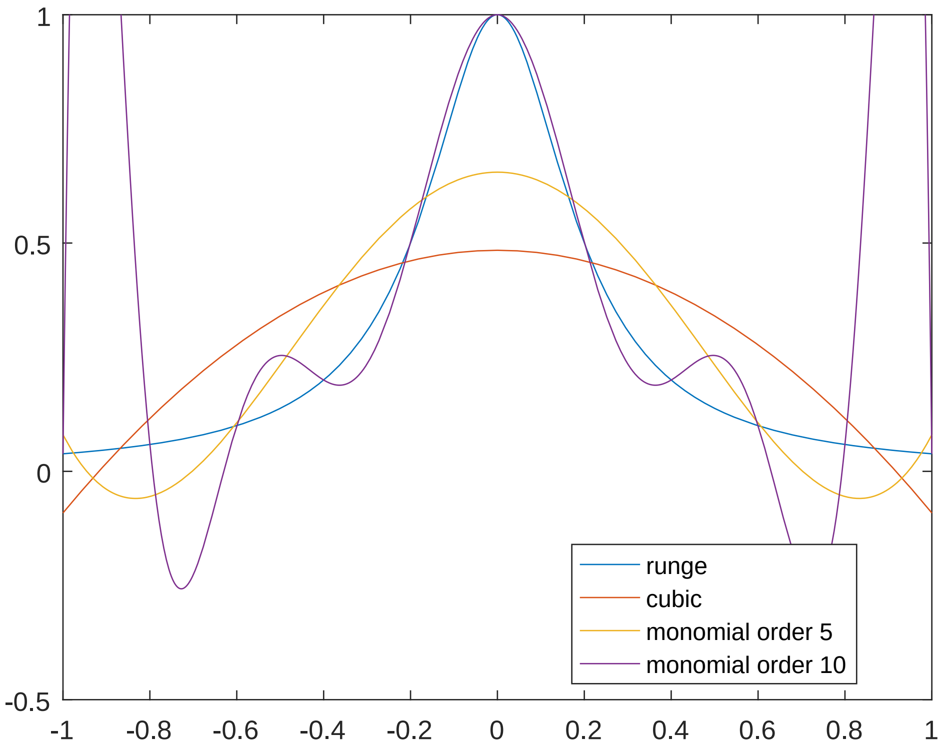
Here, all approximations of all orders give the same result. They meet the function exactly up until the kink at $x = 0$. Obviously when approximating to the right side of the kink, one cannot represent the flat line to the left of the kink with a taylor approximation, as the derivatives used don't coincide with the movement of the function before the kink.

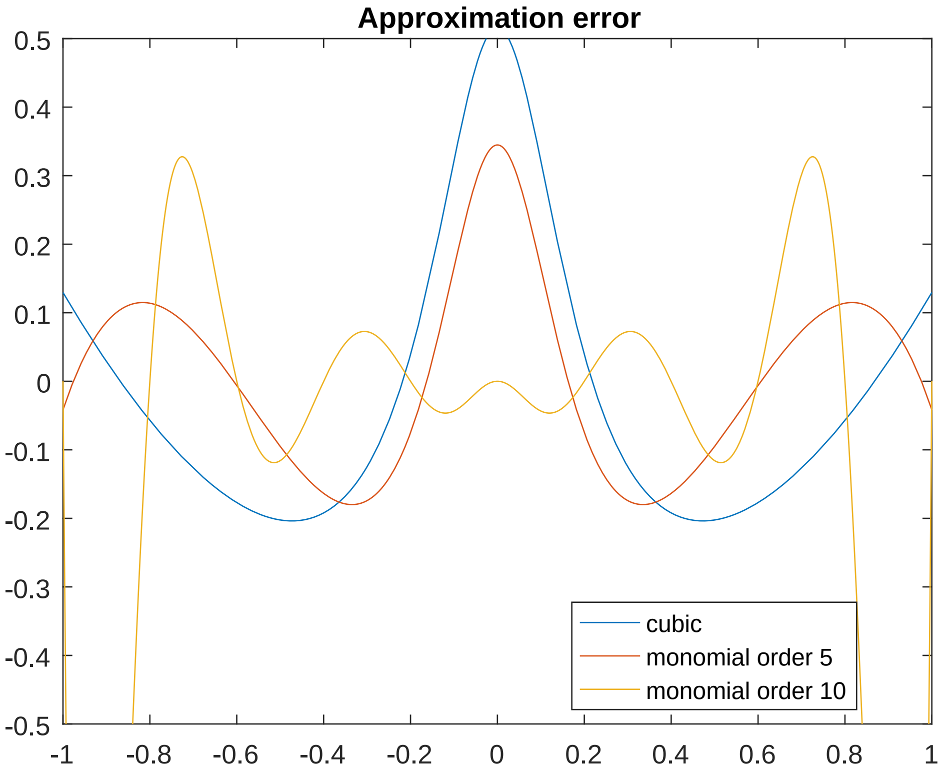
Q 3

In the following all approximation errors are (mistakenly) calculated as "real function value - approximated function value". So the resulting error graphs are flipped in comparison to the graphs in the slides !

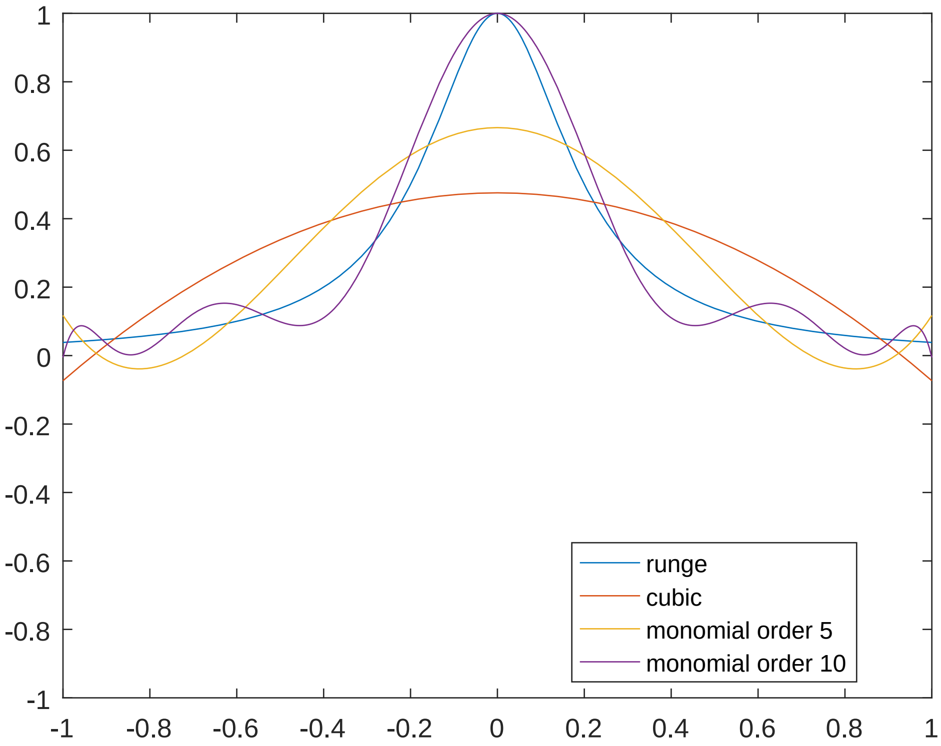
Runge function

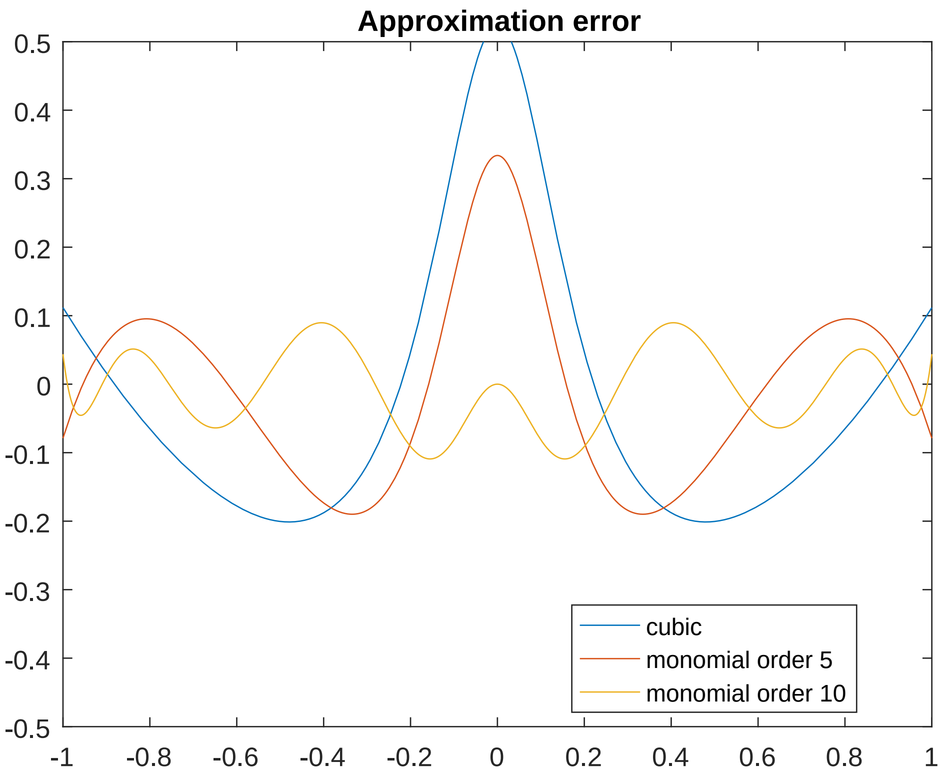
Evenly spaced nodes and cubic polynomial



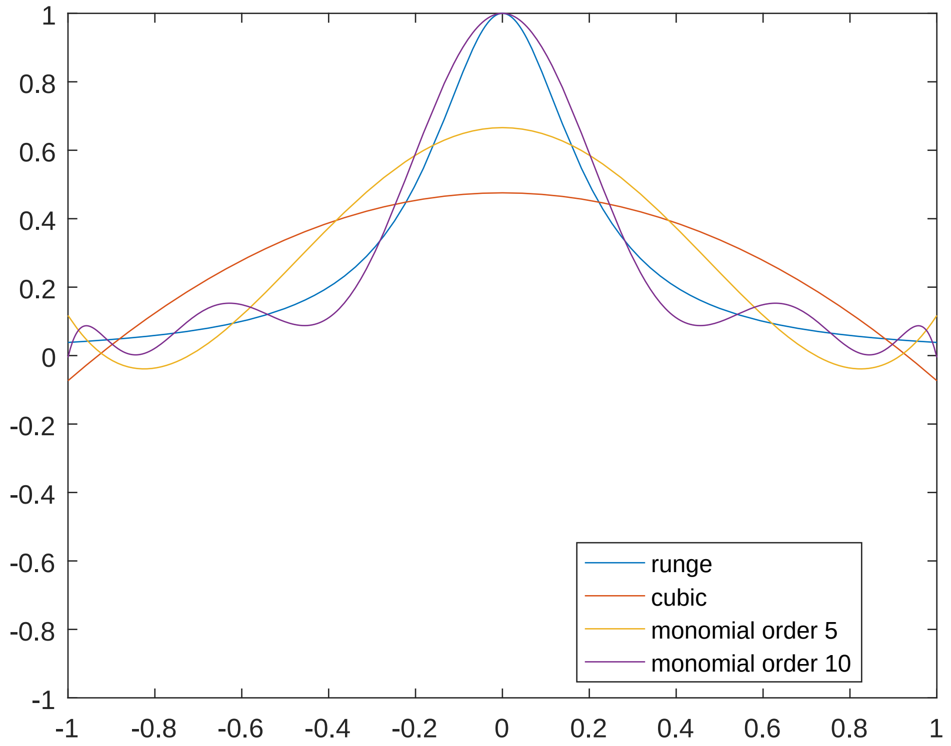


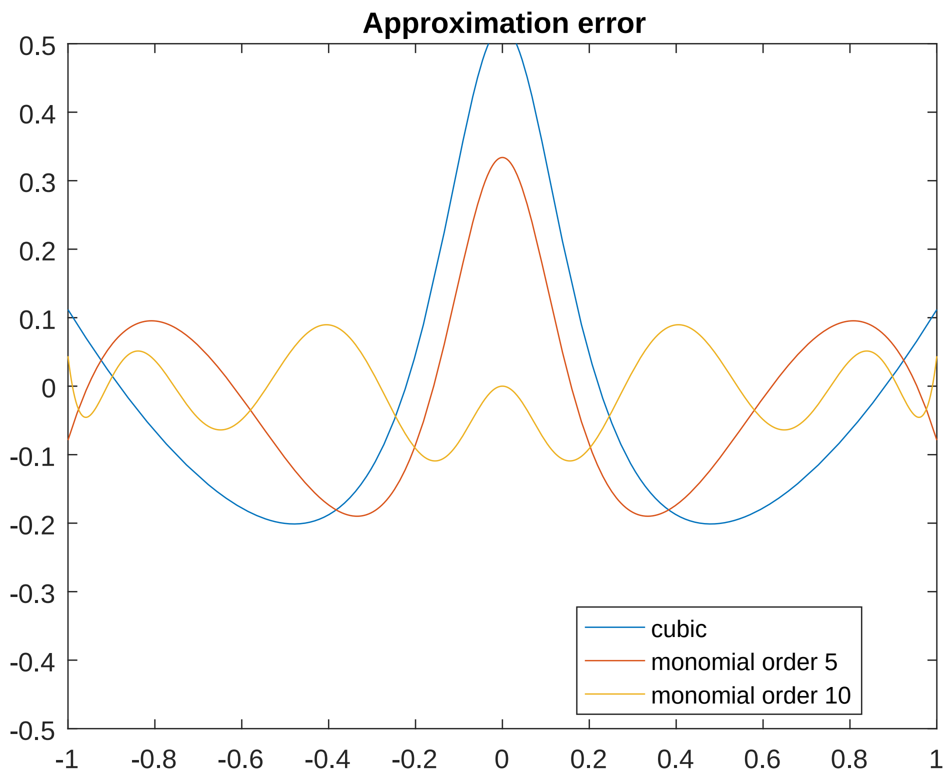
Chebyshev nodes and cubic polynomial





Chebyshev approximation algorithm

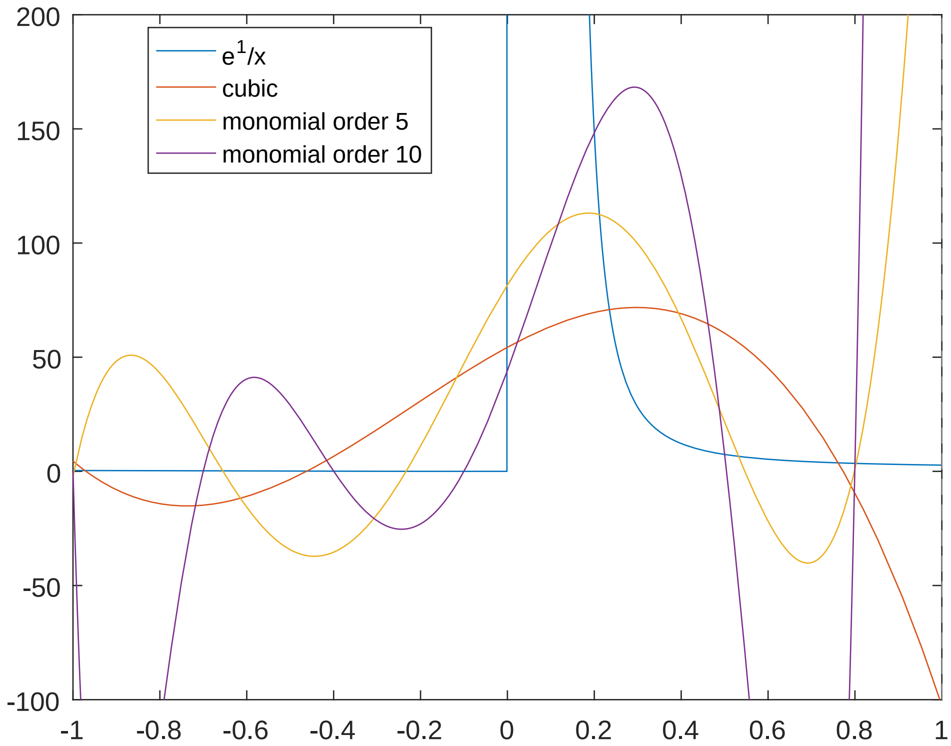


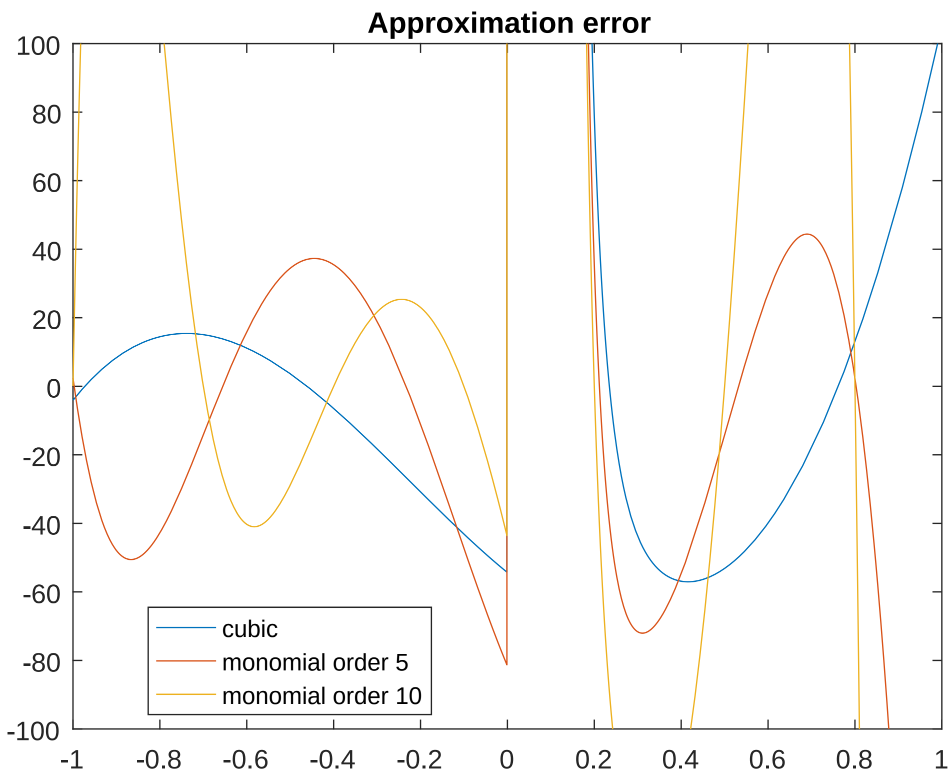


While around zero all approximation algorithm make the same (high) mistake, on the tails of the function in the the interval the chebyshev nodes perform better. This is not surprising as this way of estimating puts more weight on the tails be spacing the nodes closer together. Moreover for the runge function it seems to be unimportant when using chebyshev modes if one uses chebyshev or standard polynomials as they produce the same outcome. All improvments are made by using the chebyshev nodes.

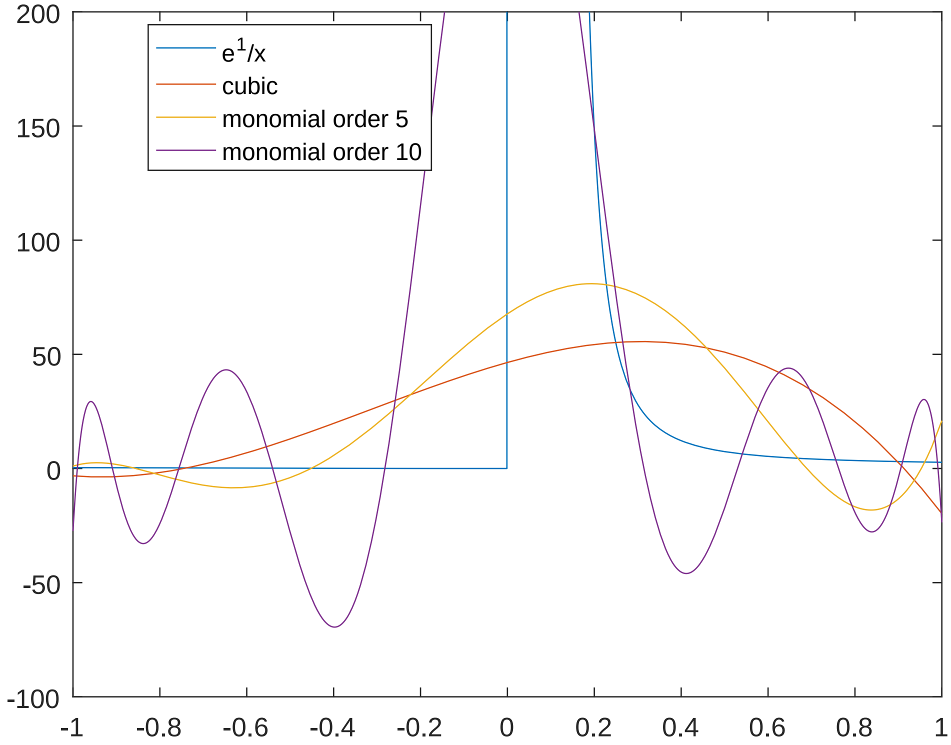
Exponential function

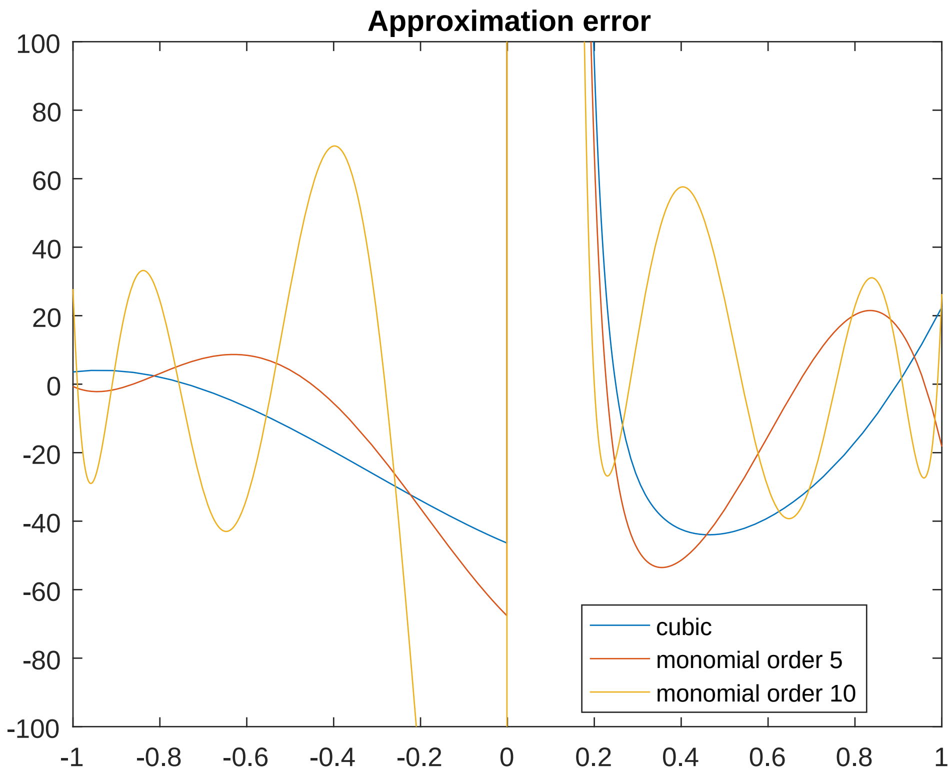
Evenly spaced nodes and cubic polynomial



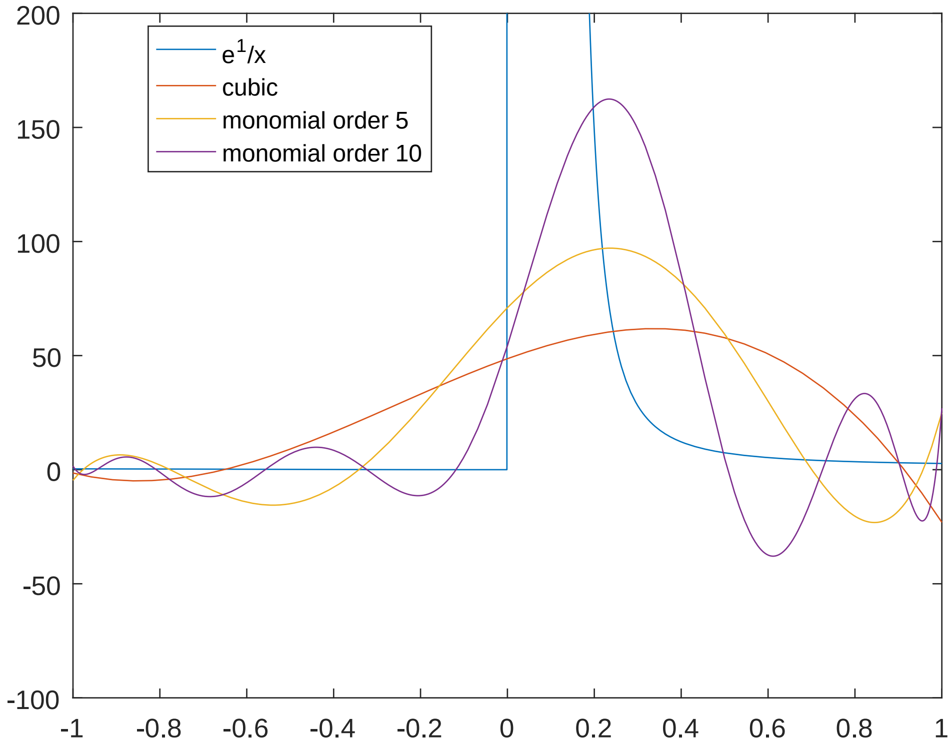


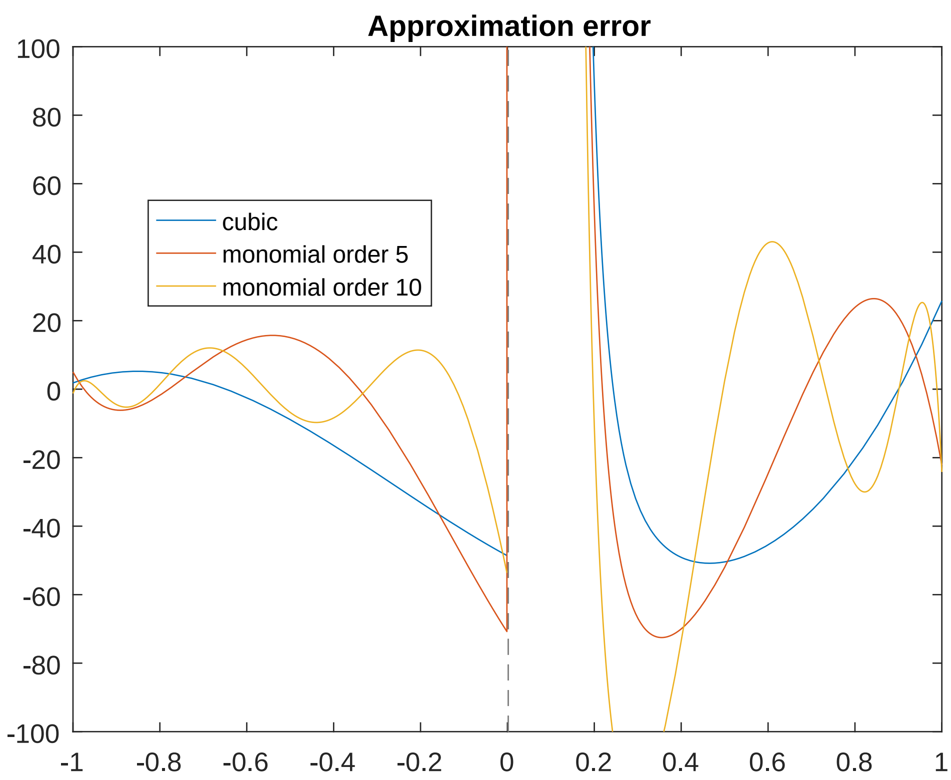
Chebyshev nodes and cubic polynomial





Chebyshev approximation algorithm

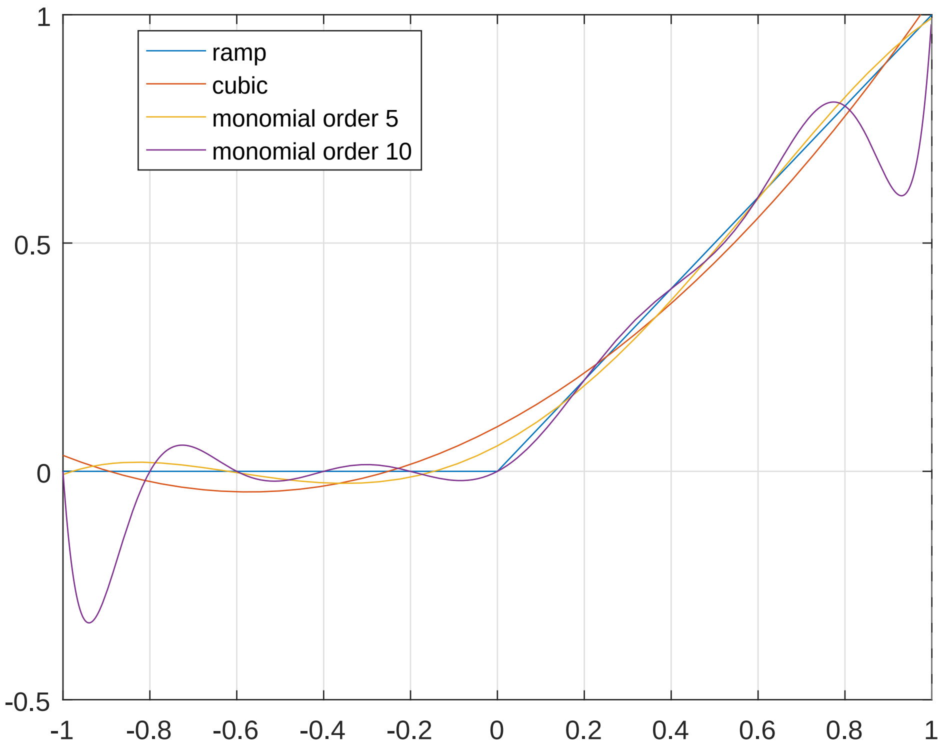


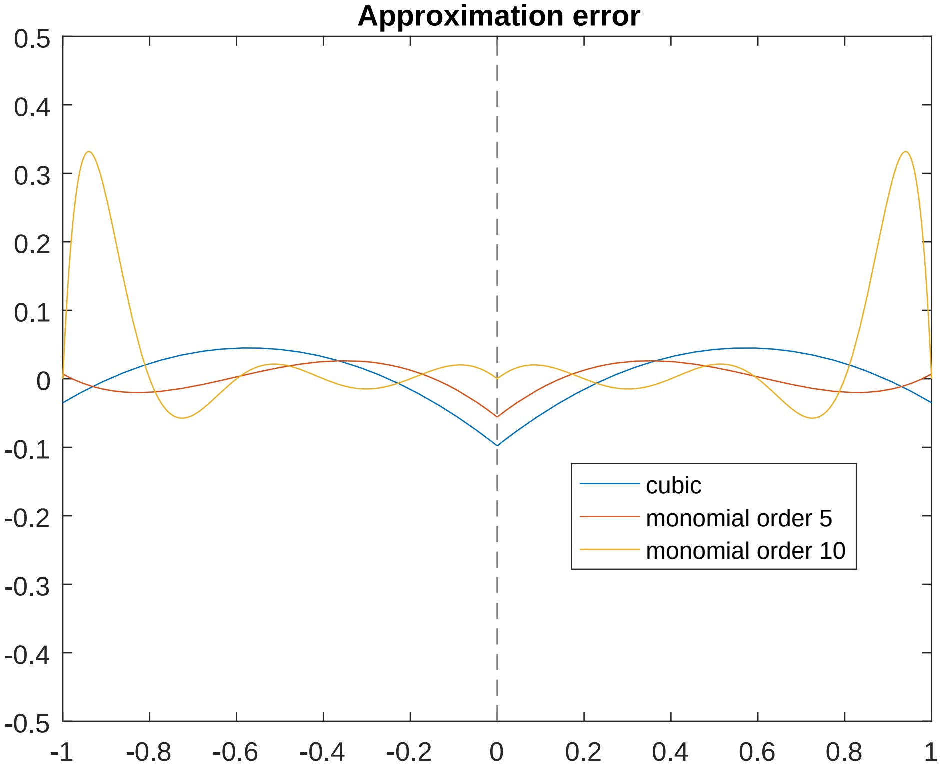


All approximation algorithms equally do not seem to be able to reproduce the behaviour of the function when x goes to zero from the left and f(x) goes to infinity. Again using chebyshev nodes improves the approximation at the tails. This time also using the chebyshev polynomial further reduces the error !

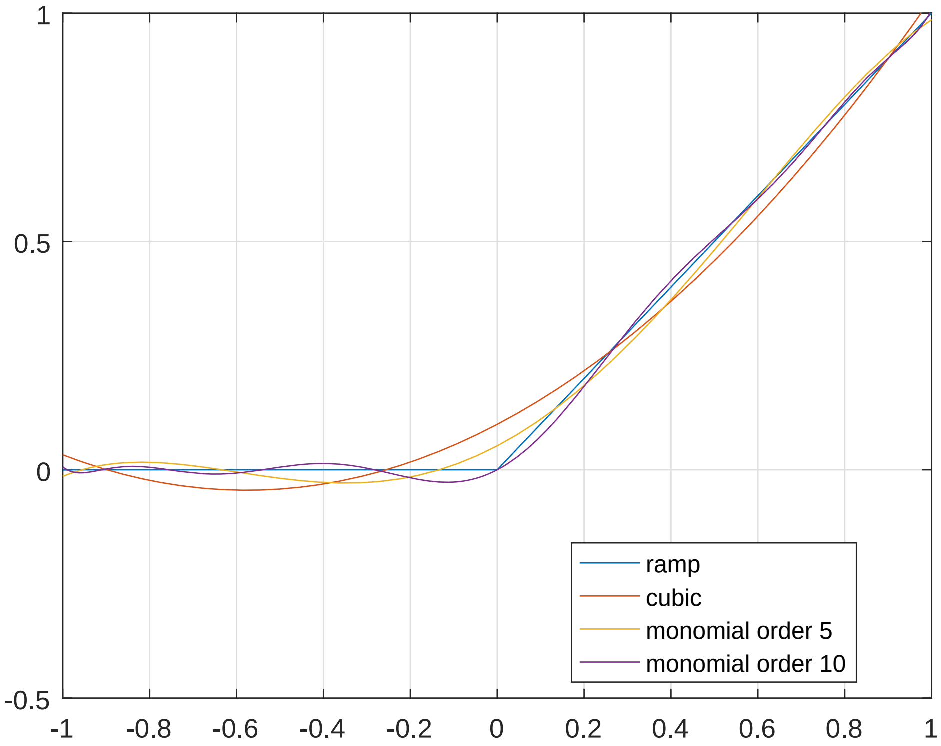
Ramp function

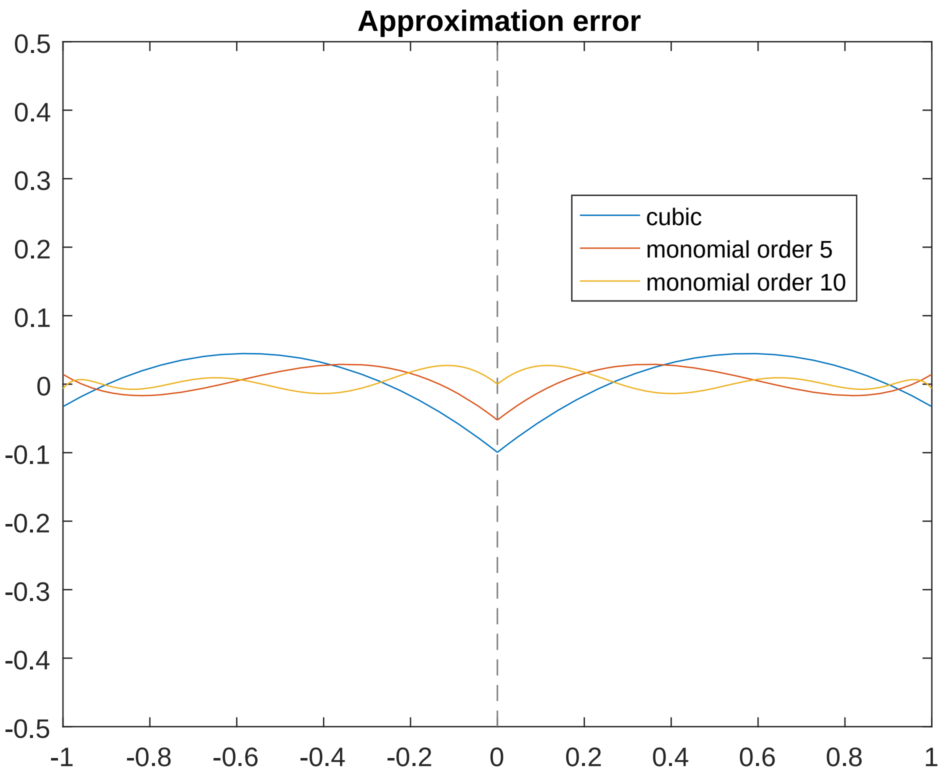
Evenly spaced nodes and cubic polynomial



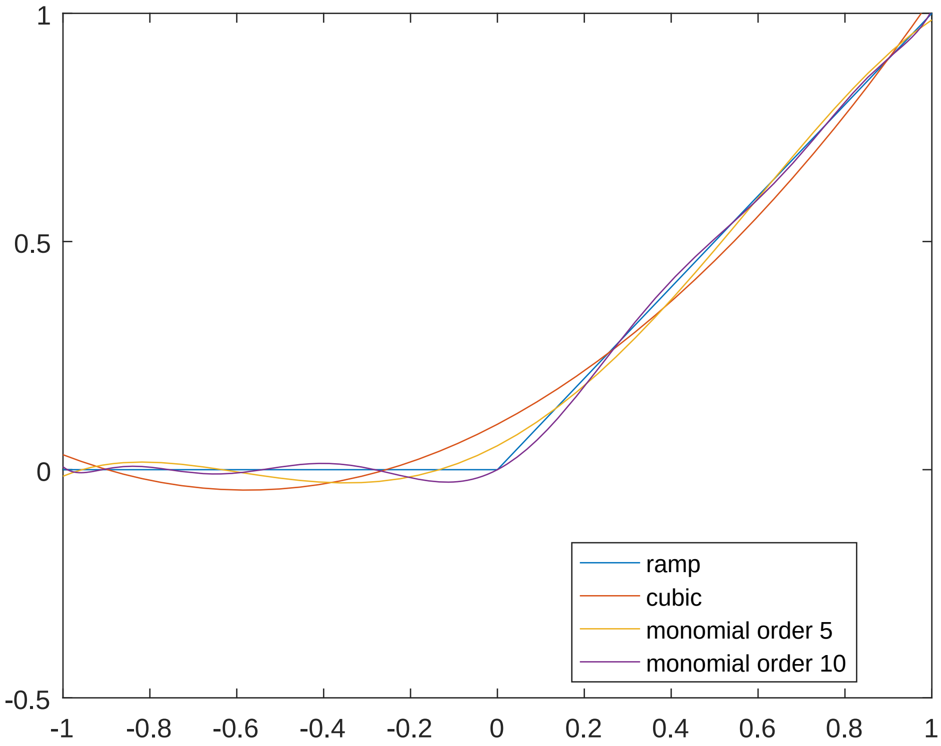


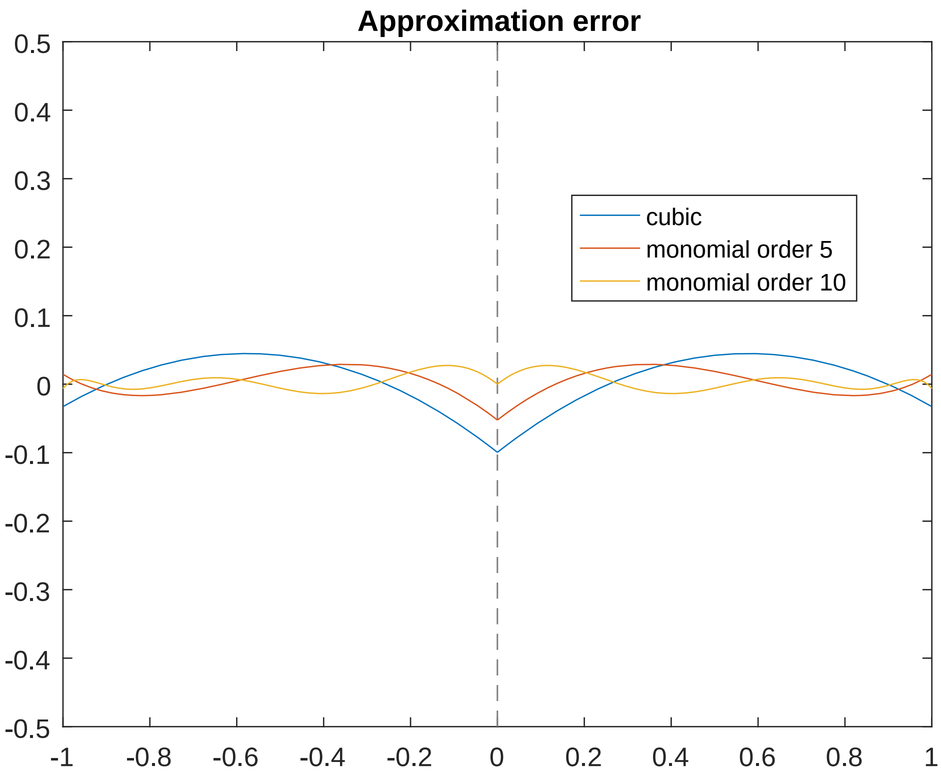
Chebyshev nodes and cubic polynomial





Chebyshev approximation algorithm





For the ramp function the biggest difference of the three alternative approximation algorithms is a bigger error at the tails when using evenly spaced nodes for higher order approximations. For lower order approximations all three algorithms yield the same results.

CES production function

ES

$$f(k,h) = ((1-\alpha)k^{\frac{\sigma-1}{\sigma}} + \alpha h^{\frac{\sigma-1}{\sigma}})^{\frac{1}{\sigma-1}}$$

The fraction of the derivatives of the CES function with respect to each input yields the MRS, i.e.:

$$MRS = \frac{1-\alpha}{\alpha}(\frac{h}{k})^{\frac{1}{\sigma}}$$

We define $MRS = \theta$ so that $\frac{h}{k} = \frac{\alpha}{1-\alpha}\theta^{\sigma}$ Following the definition of the elasticity of substitution we get

$$ES = \frac{dln(\frac{h}{k})}{dlnMRS} = \frac{d\frac{\alpha}{1-\alpha}\theta^{\sigma}}{d\theta}\frac{\theta}{\frac{\alpha}{1-\alpha}\theta^{\sigma}} = \frac{\sigma\theta^{\sigma-1}\theta}{\theta^{\sigma}} = \sigma$$

Labour share

Assuming competitive markets, the labour share of income can be calculated as

$$LS = \frac{rK}{y} = \alpha(\frac{l}{Y})^{1-\frac{1}{\sigma}}$$

Approximation

The m file contains my best try in approximating the CES function with a 2D chebyshev algorithm. I calculated wieghts and the corresponding polynomials. As i could not move further from there I cannot display any other results here. Except a plot of the isoquants of the CES function