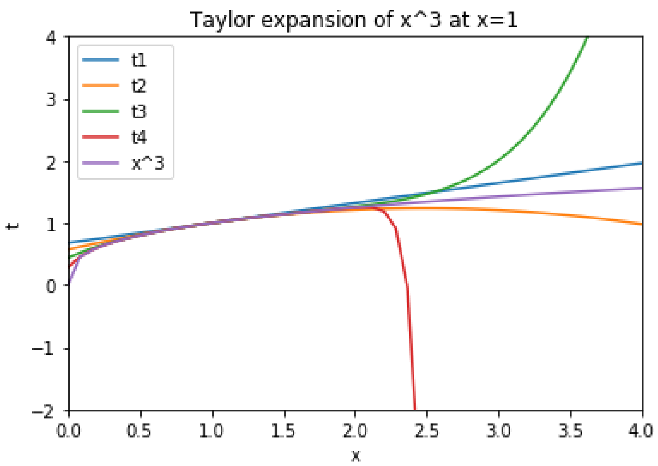
QUANT MACRO: PROBLEM SET 2

Mario Serrano

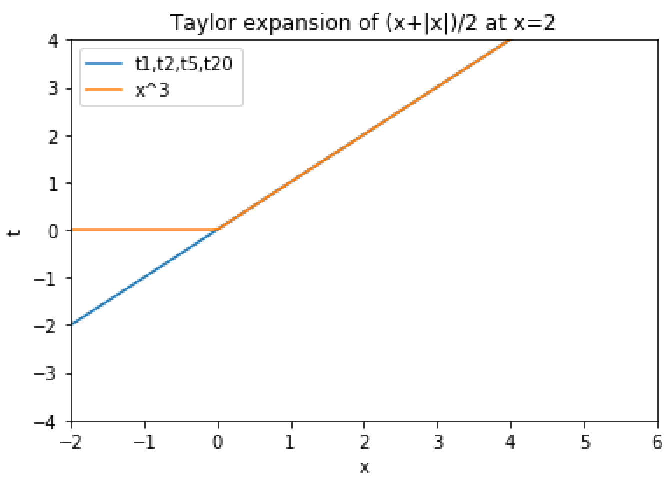
QUESTION 1: FUNCTION APPROXIMATION, UNIVARIATE

1. Approximate f(x) = x .321 with a Taylor series around x = 1. Compare your approximation over the domain (0,4). Compare when you use up to 1, 2, 5 and 20 order approximations. Discuss your results.



As we see, Taylor expansions are good approximations around x = 1, since we are dealing with a local approximation method. However, as we deviate from this point, approximation function became less and lees accurate. Moreover, high order approximations seems to be less useful to estimate the function away from x = 1 than low order. Therefore, to raise the order of the approximation could improve the accuracy around 1, but it is not useful for the whole function.

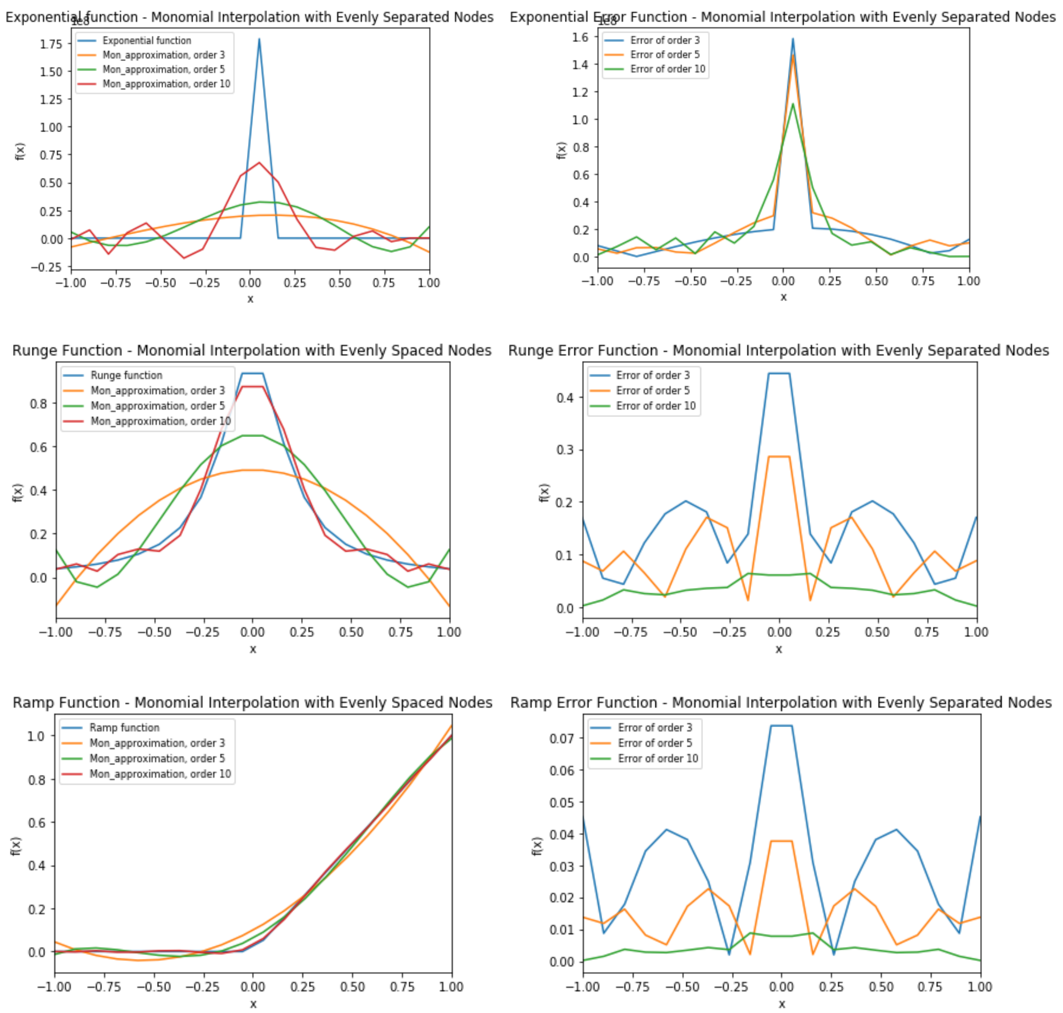
2. Approximate the ramp function with a Taylor series around x = 2. Compare your approximation over the domain (0,6). Compare when you use up to 1; 2; 5 and 20 order approximations. Discuss your results



In this case the Taylor expansion seems to be a perfect approximation to the function, for values higher than 0. This is because in this domain the original function is linear, so the Taylor expansion can replicate it perfectly. However, in x = 0 there is a kink point and the slope of the function change drastically. The Taylor expansion cannot deal with this problem, so as the function take values away from the kink point, the approximation becomes less and less accuracy. Due to this problem, it is not useful to elevate the order of the expansion, if we want to approximate the function in this part of the domain.

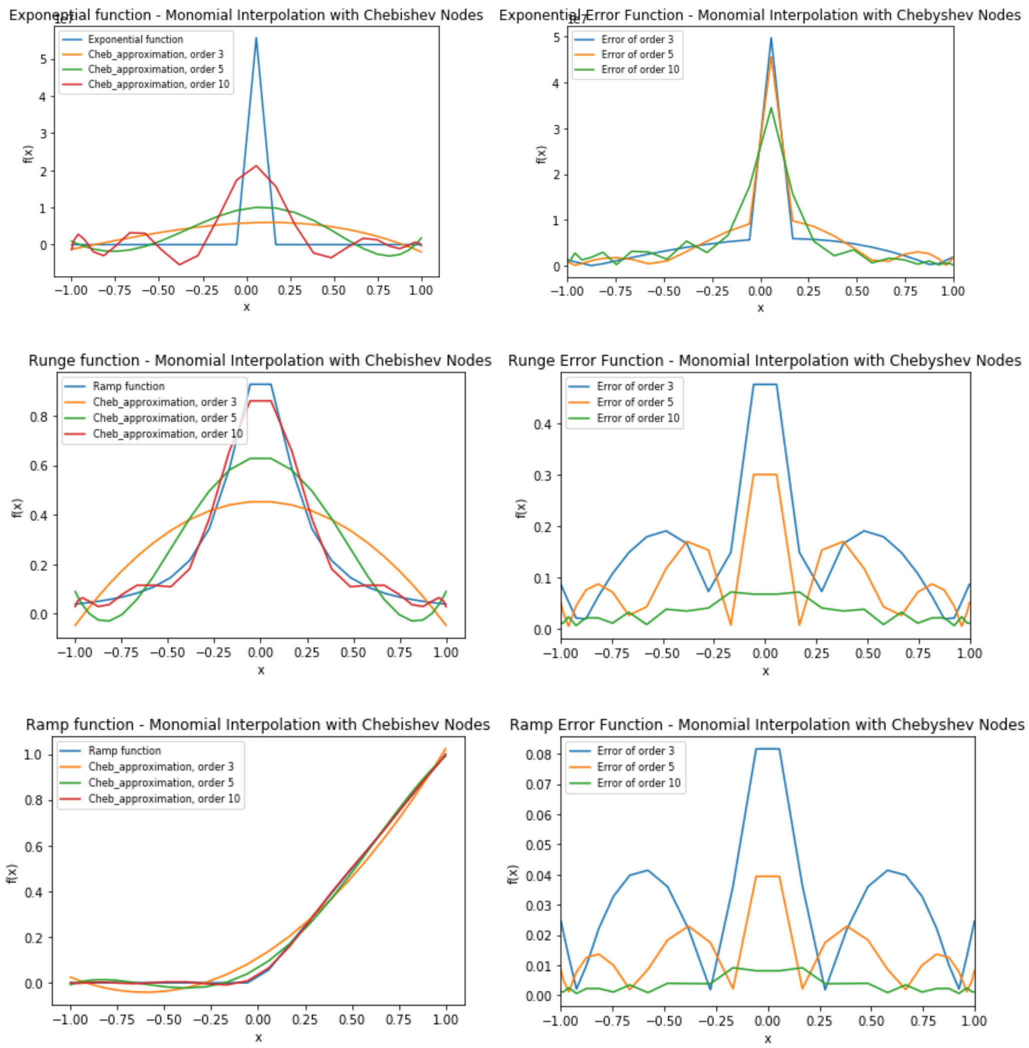
3. Approximate these three functions: e ^(1/x), the runge function, and the ramp function for the domain [-1; 1] with:

3.1. Evenly spaced interpolation nodes and a cubic polynomial. Redo with monomials of order 5 and 10. Plot the exact function and the three approximations in the same graph. Provide an additional plot that reports the errors as the distance between the exact function and the approximantions.



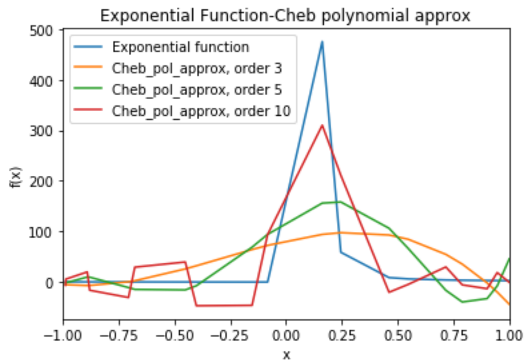
With monomial approximation with evenly separated nodes we can see that the approximation at the extreme points is quite poor. This is because monomial bases are not orthogonal. Moreover, as you increase the number of nodes the function is less an less well approximated, as we can see in the error graphic of the Runge function.

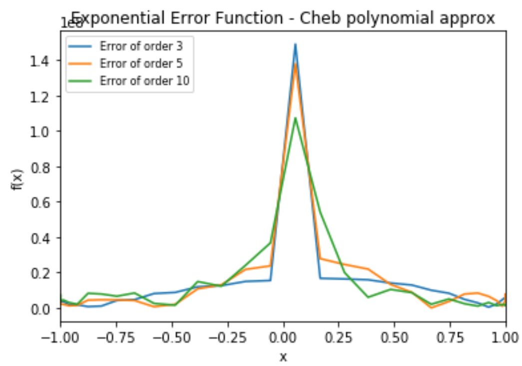
3.2. Chebyshev interpolation nodes and a cubic polynomial. Redo with monomials of order 5 and 10. Plot the exact function and the three approximations in the same graph. Provide an additional plot that reports the errors as the distance between the exact function and the approximand.

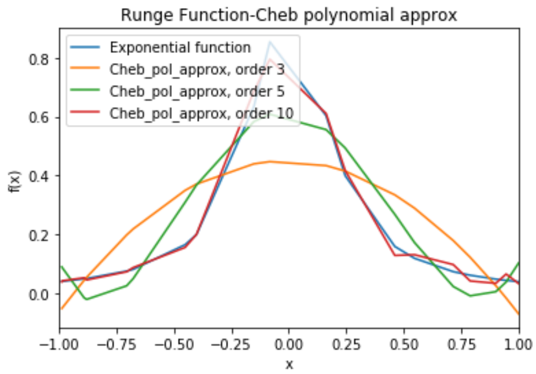


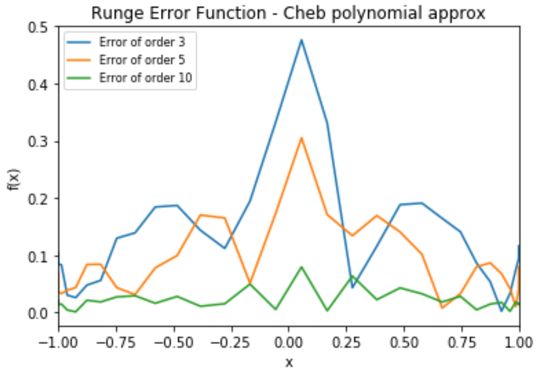
Now, we are approximating with monomial interpolation at Chebyshev nodes. We should see that this kind of approximation is more accurate at extreme points, that is, the approximation error at these points is lower. This is because the Chebyshev roots are closer to one another in the extreme points and more spaced at the central ones. However I obtain the same graphics as before, probably due to an error in my code that I could not solve.

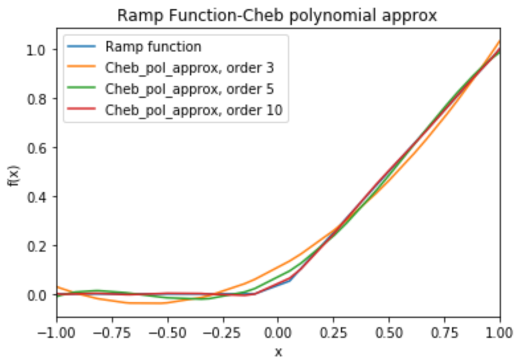
3.3 Chebyshev interpolation nodes and Chebyshev polynomial of order 3, 5 and 10. How does it compare to the previous results? Report your approximation and errors.

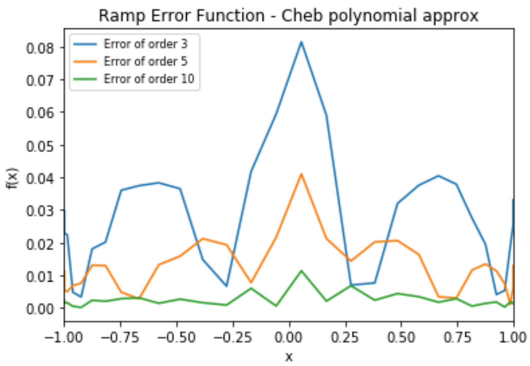












In this third part we are doing Chebychev polynomial interpolation with Chebychev roots. We can see that the approximations are smoother. However, when we have kinks or any singularity of the function, as we have in the first or third functions for instance, Chebyshev polynomials does not approximate better those points.

A conclusion, we can say that for approximate the whole function it is better to use Chebyshev nodes, specially for the three method. However, we have to take into account that these three methods are not accurate to deal with kinks.

QUESTION 2: FUNCTION APPROXIMATION, MULTIVARIATE

Consider the following CES function f(k,h) = $[(1-\alpha)k^{\sigma-1/\sigma} + \alpha h^{\sigma-1/\sigma}]$^σ/(σ-1)where σ is the elasticity of substitution (ES) between capital and labor and α is a relative input share parameter. Set σ = 0,5, α = 0,25, k E [0, 10] and h E [0, 10]. Do the following items:

- **Show that σ is the ES**

Elasticity of substitution can be defined as:

σ = d ln (h/k)/d ln (f_k / f_h)

Marginal productivities are:

- MPH = $\frac{df(k,h)}{dh} = \frac{\sigma}{\sigma-1}\left[(1-\alpha)k^{\sigma-1/\sigma} + \alpha h^{\sigma-1/\sigma}\right]^{\sigma/(\sigma - 1)} \frac{(1-\alpha)(\sigma-1)}{\sigma} k^{-1/\sigma}$

- MPK = $\frac{df(k,h)}{dk} = \frac{\sigma}{\sigma-1}\left[(1-\alpha)k^{\sigma-\frac{1}{\sigma}} + \alpha h^{\sigma-\frac{1}{\sigma}}\right]^{\sigma/(\sigma - 1)} \frac{\alpha(\sigma-1)}{\sigma} h^{-\frac{1}{\sigma}}$

We compute the denominator and take logarithms:

$$\frac{df(k,h)}{dk} \Big/ \frac{df(k,h)}{dh} = \frac{(1-\alpha)h^{1/\sigma}}{\sigma\, k^{1/\sigma}} = \frac{(1-\alpha)}{\sigma}\left(\frac{h}{k}\right)^{1/\sigma} = \frac{MPK}{MPH}$$

$$\log\left(\frac{MPK}{MPH}\right) = \log\left(\frac{1-\alpha}{\sigma}\right) + \frac{1}{\sigma}\log\left(\frac{h}{k}\right)$$

So, taking derivatives with respect to ln (h/k) we obtain:

$$\varepsilon_{kh} = \sigma$$

Therefore, we prove that sigma is the elasticity of substitution.

- **Compute labor share for an economy with that CES production function assuming factor inputs face competitive markets**

The labor share can be defined as:

$$s = \frac{MPH\ h}{f(k,\ h)}$$

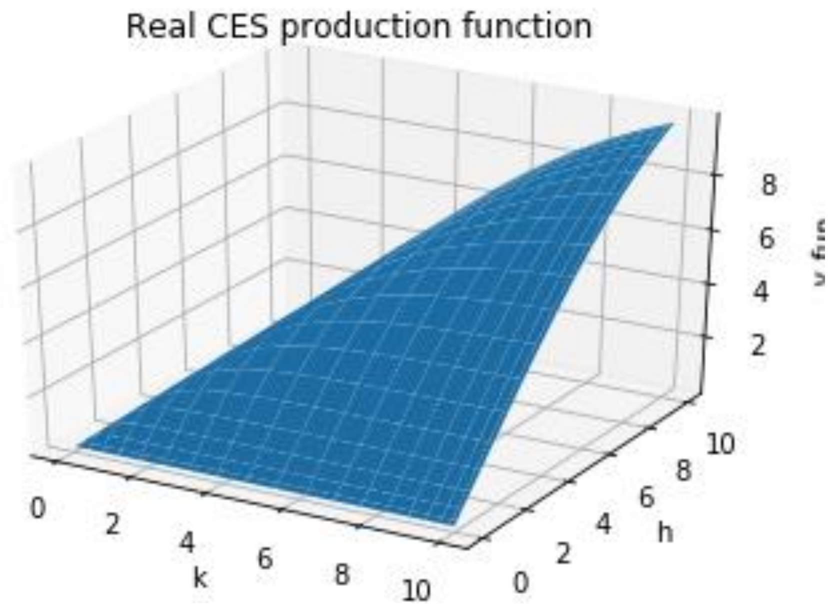
Therefore:

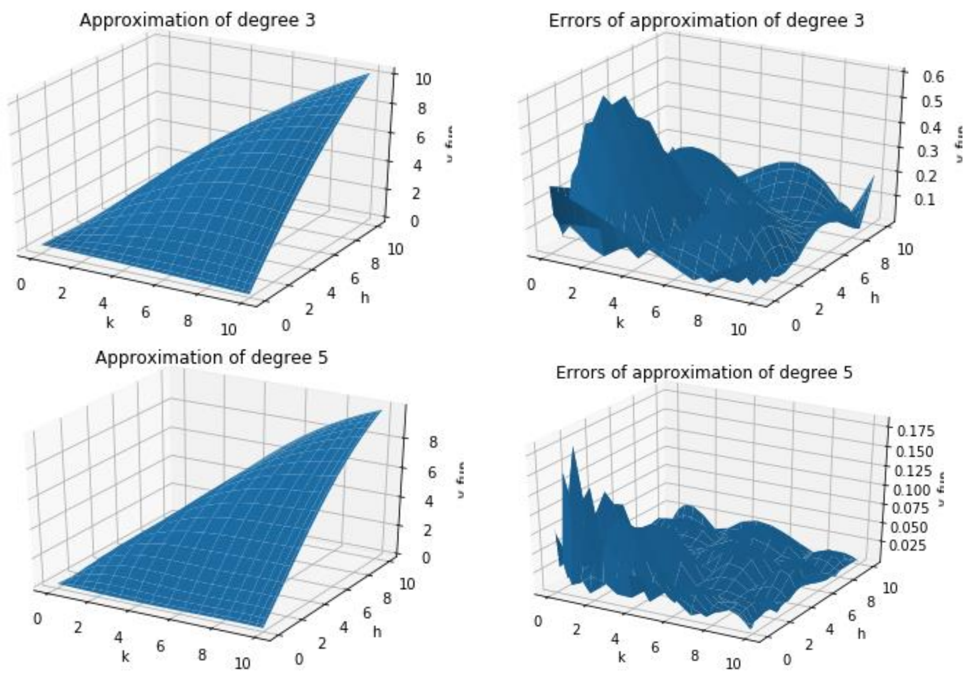
$$\text{MPH h} = \alpha h\left[(1-\alpha)k^{\sigma-1/\sigma} + \alpha h^{\sigma-1/\sigma}\right]^{\sigma/(\sigma - 1)} k^{-1/\sigma}$$

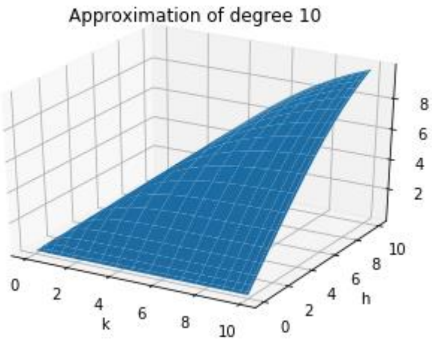
Solving for s:

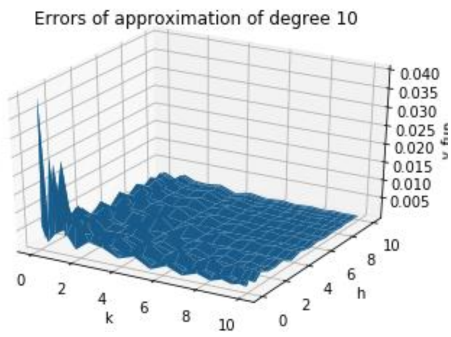
$$s = \frac{1}{3(k^{-3}+1)}$$

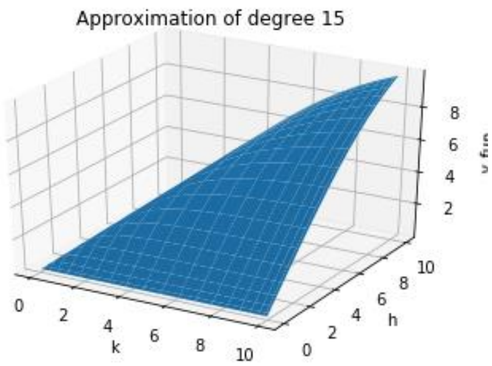
- **Approximate f(k,h) using a 2-dimensional Chebyshev regression algorithm. Fix the number of nodes to be 20 and try Cheby polynomials that go from degree 3 to 15. For each case, plot the exact function and the approximation (vertical axis) in the (k,h) space.**

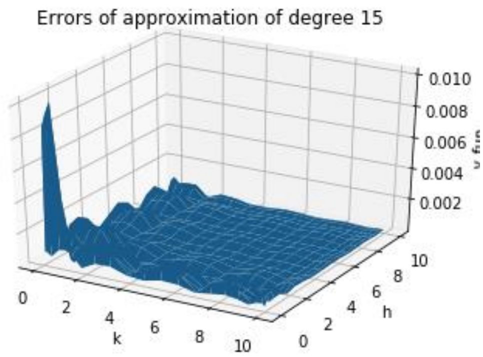




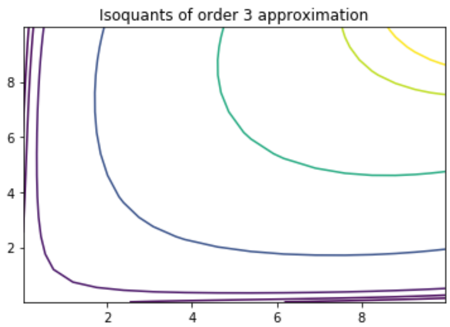


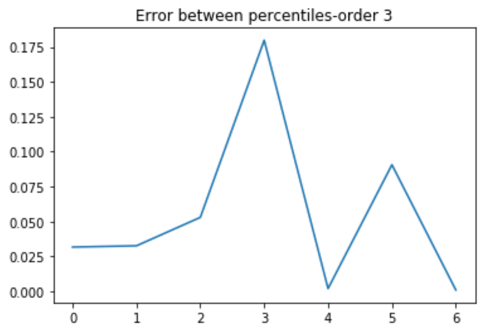


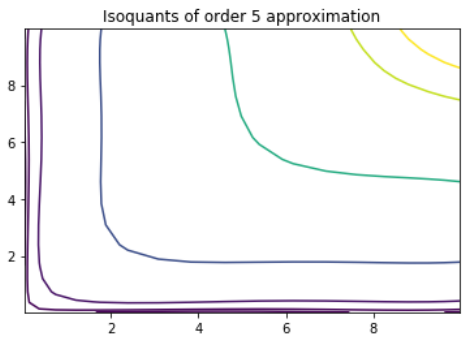


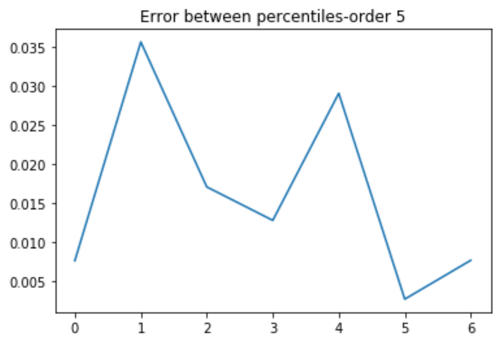


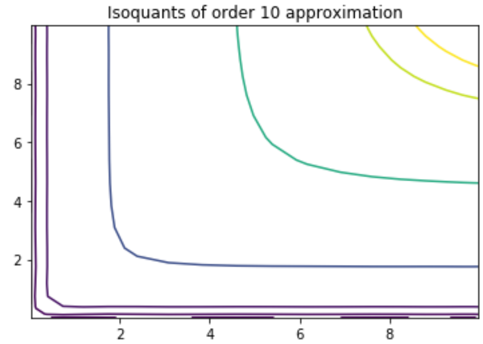
- **Plot the exact isoquants associated with the percentiles 5, 10, 25, 50, 75, 90 and 95 of output. Use your approximation to plot the isoquants of your approximation. Plot the associated errors per each of these isoquant.**

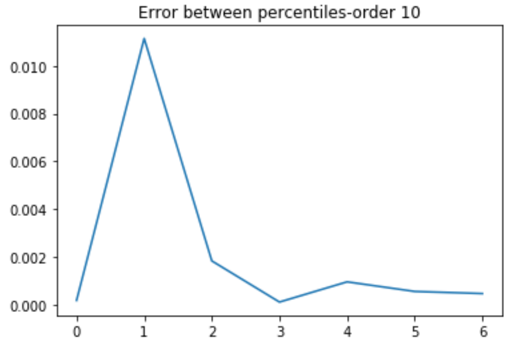


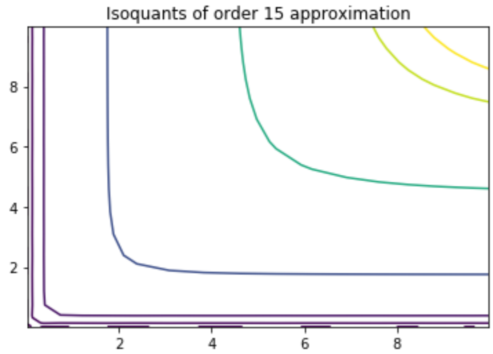


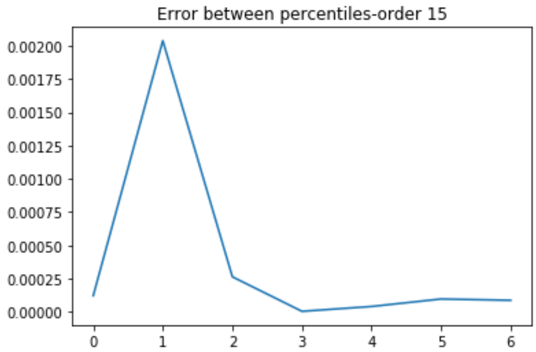





Isoquants of order 10 approximation
2
4
6
8
2
4
6
8


Error between percentiles-order 10
0.000
0.002
0.004
0.006
0.008
0.010
0
1
2
3
4
5
6


Isoquants of order 15 approximation
2
4
6
8
2
4
6
8


Error between percentiles-order 15
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
0.00200
0
1
2
3
4
5
6