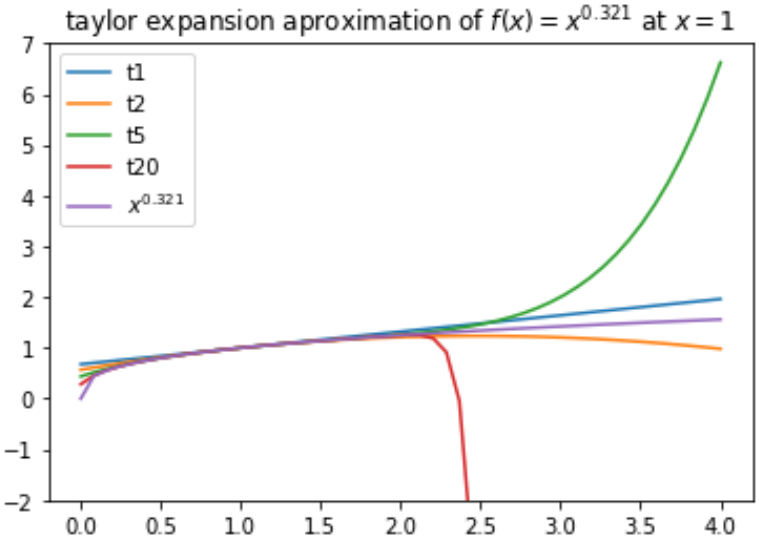
I did this problem set in collaboration with **Germán Sánchez Arce** and **María Gonzalez**.

Question 1. Function Approximation: Univariate

1-Approximation of exponential:

Approximate f(x) = $x^{0.321}$ with a Taylor series around x = 1, of order 1, 2, 5 and 20:



We can see that any taylor aproximation is quite good for values near to 1

Nevertheless, the more far away we go from 1. the worse is the aproximation

Beside if error will grow up strongly when we increase the degrees of the taylor

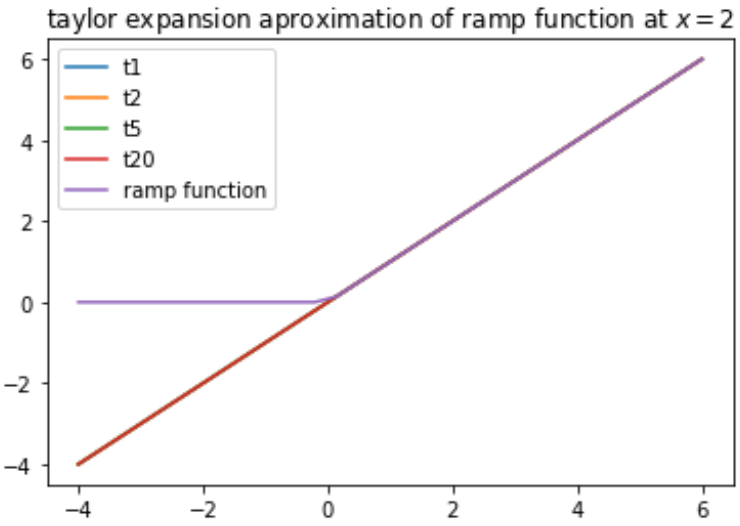
at the same time that we go away from 1.

Hence, may be, it is a good thing to use more degrees in order to aproximate a point of f

But in this case is not a good idea increase the degrees if what we want is aproximate

the whole function

2-Aproximation of ramp function:

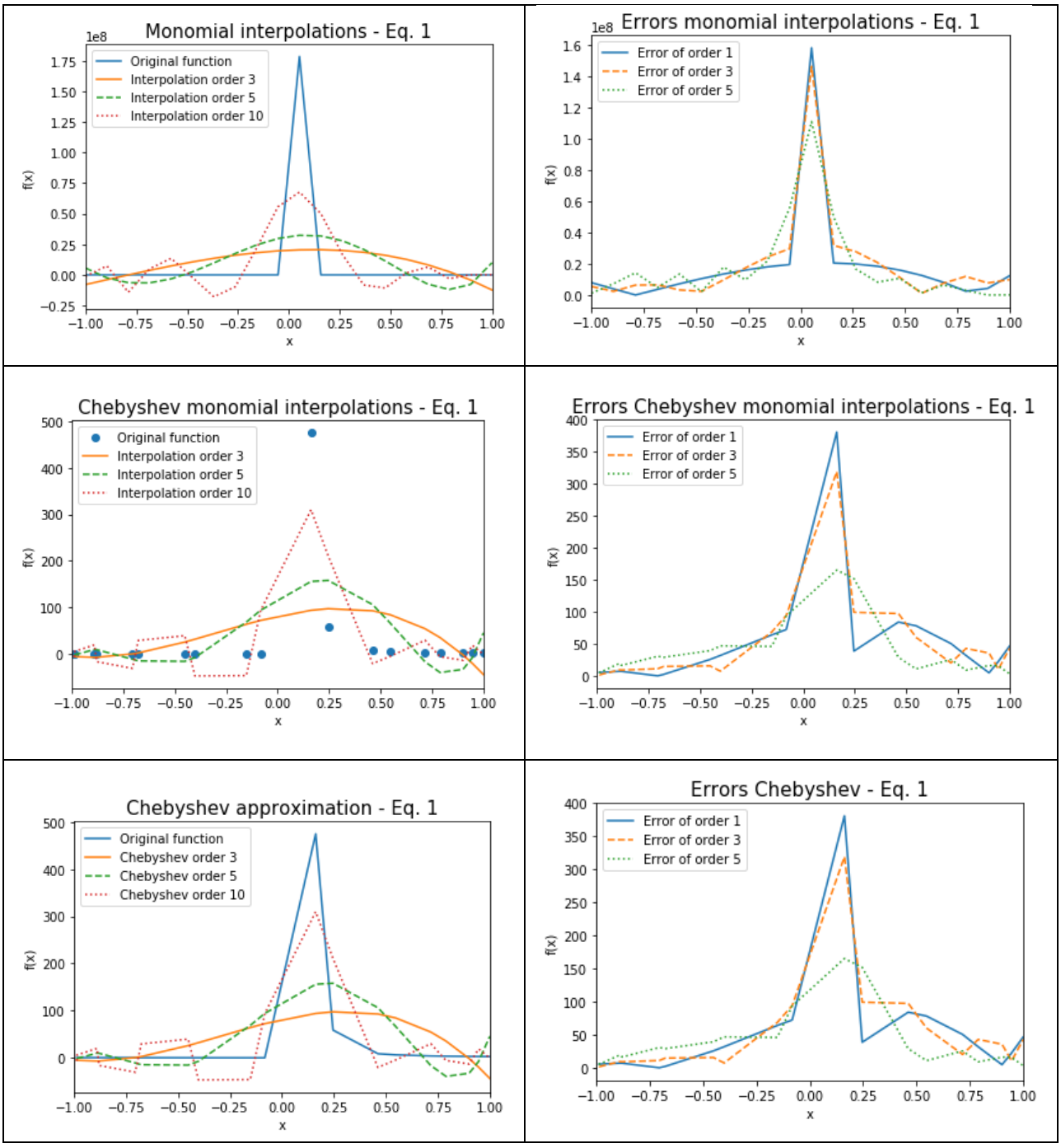


We can see that any taylor aproximation is perfect for values near 2

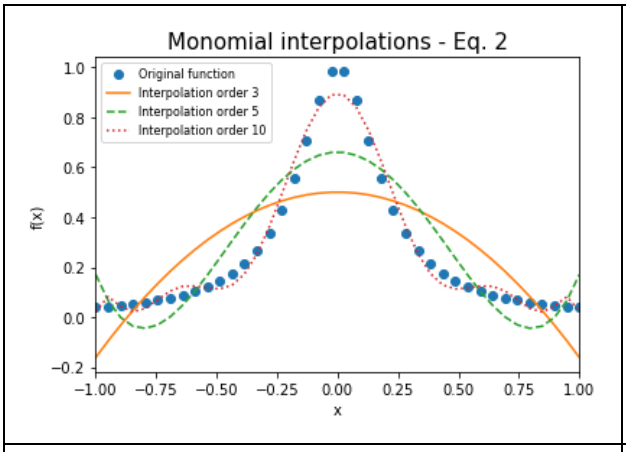
Nevertheless, since we have a sudden kink in x = 0, we have that taylor aproximations no matter the degree they are will generate and increasing error of aproximation, since the actual function is slope 0 and all the taylor expansions are slope 1.

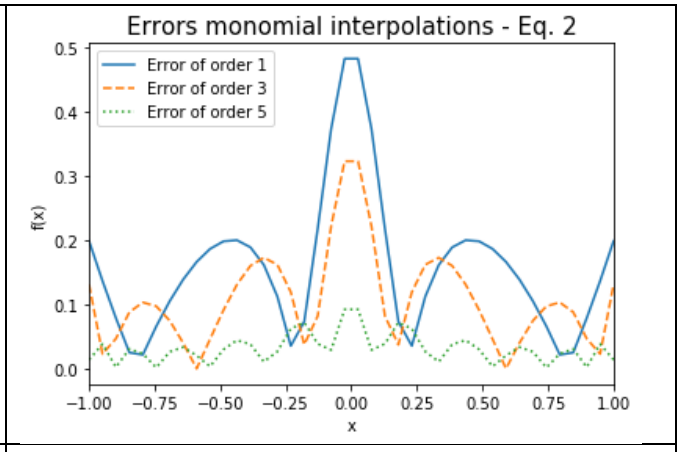
3- Interpolations:

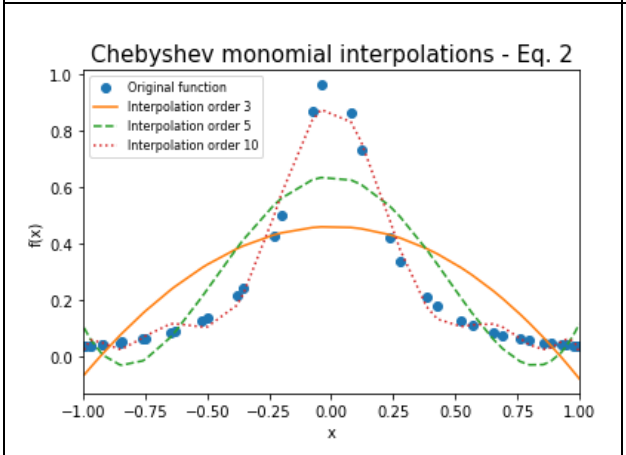
First function: $e^{1/x}$

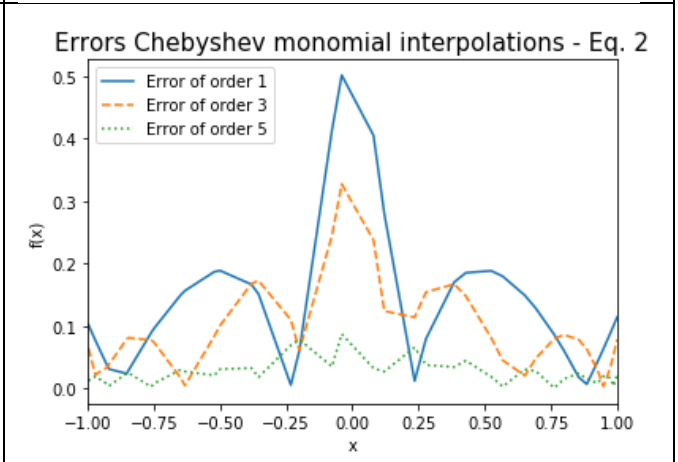


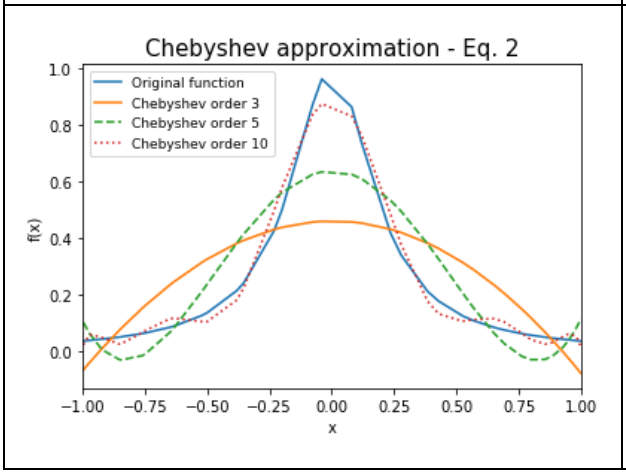
Second function: Runge function

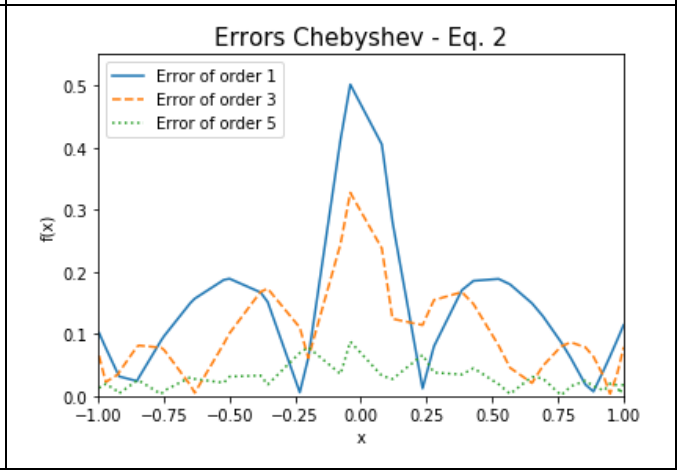




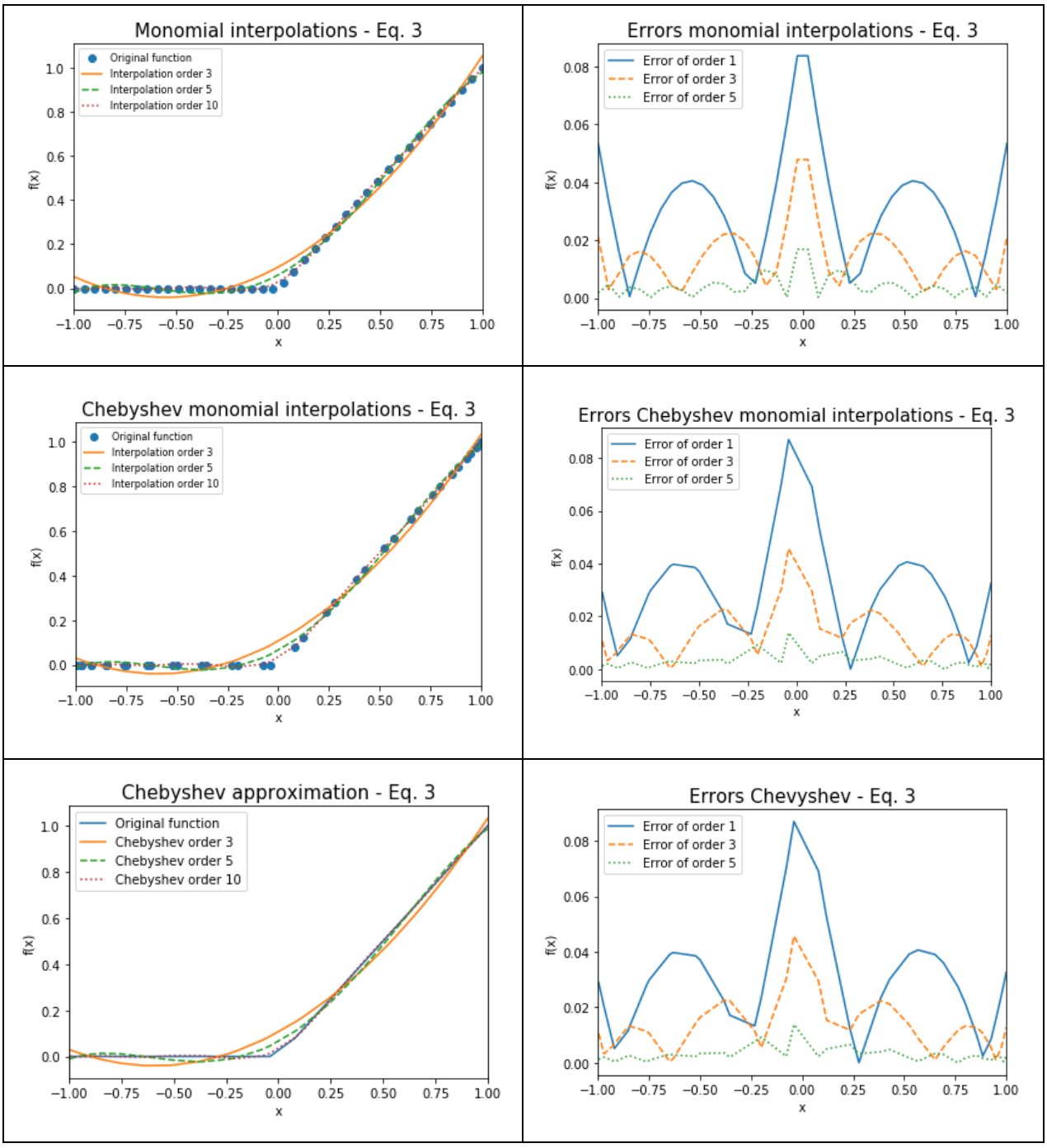








Third function: ramp function



The first approximation is done with evenly-spaced interpolation nodes and monomials of order 3, 5 and 10. The conclusion of this kind of approximation is that, when we increase the order of the polynomial, less informative it is, this is because monomial bases are not orthogonal. Moreover, what we can see is that in the extreme points, the approximation error is bigger than in the next two kinds of approximations.

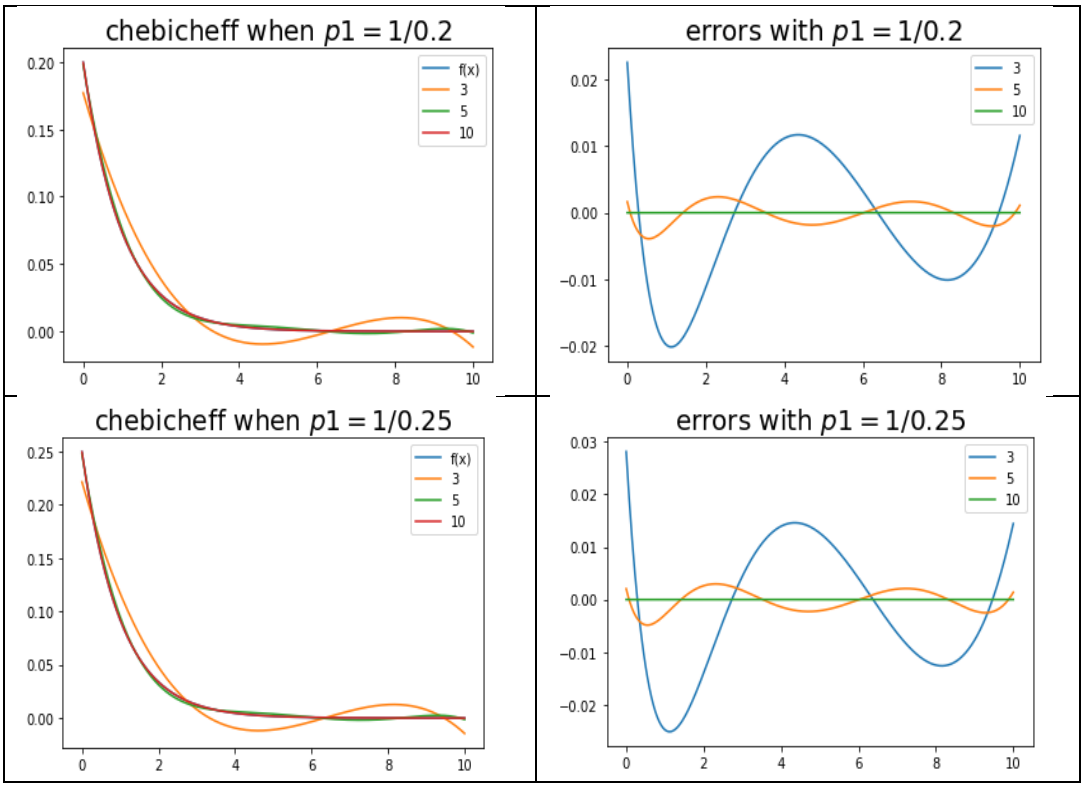
The second approximation is done also with monomials of order 3, 5 and 10, but with the Chebyshev nodes, that is, not evenly-spaced interpolation nodes. The main result we obtain with this approximation is that the errors at the extreme points are smaller than in the previous procedure. This is because the Chebyshev roots are closer to one another in the

extreme points and more spaced at the center ones. In fact, what we can see is that polynomials exhibit equioscillant errors if we interpolate using the Chebyshev nodes.

In the third approximation we use, as in the previous one, the Chebyshev roots, but now those points are interpolated with the Chebyshev polynomial instead of the monomials procedure, which is supposed to be a smoother approximation. 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.

In the last two cases, we observe that the approximation errors are quite similar and smaller than the error generated with the first one. This means that the Chebyshev nodes are important in order to have a better global approximation of the function, but if we have to decide what we would use, we'll end up with the last one (because it has some nice properties like the orthogonality of the basis), always taking into account that, since it is a smooth approximation, all the kinks and singularities the function could have will not be well-approximated.

4- Exercice of approximating a probability function:



Question 2. Function Approximation: Multivariate

1. Show that σ is the elasticity of substitution:

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

First, we need to calculate the marginal productivity of labor (MPL) and the marginal productivity of capital (MPK):

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

Dividing both marginal productivities we get:

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

Now, taking logarithms:

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

Since we are looking for the marginal rate of substitution of capital and labor, we need to take derivative in (1) with respect to $\log\left(\frac{k}{h}\right)$ (or w.r.t. $\log\left(\frac{h}{k}\right)$ and take the inverse). The result we obtain is:

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

2. Obtain the labor share of an economy with CES production function.

We know that labor share is given by:

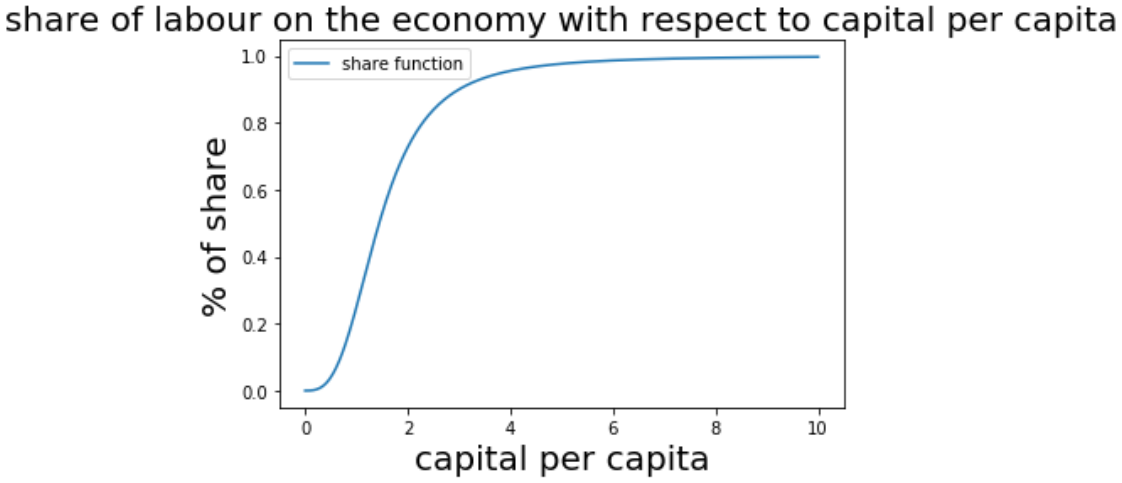
$$s = \frac{MPH\ h}{f(k,\ h)} \qquad (2)$$

Therefore:

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

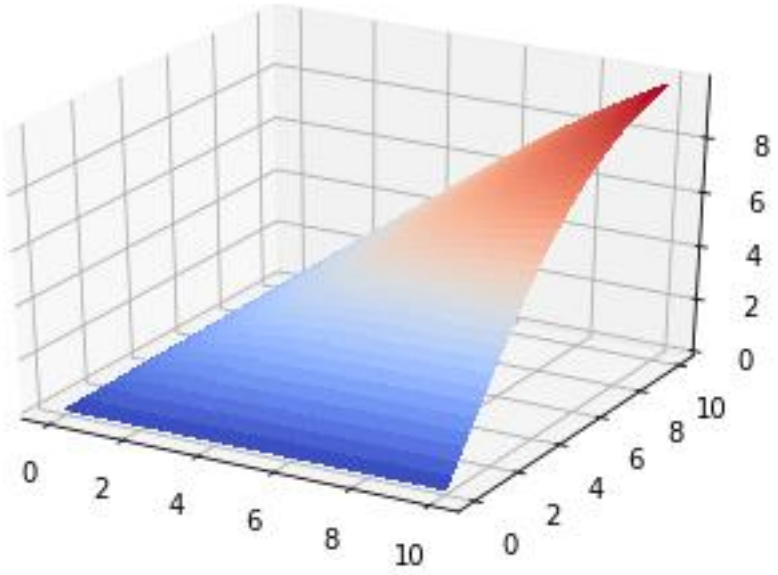
Call A to $(1-\alpha)k^{\sigma-1/\sigma} + \alpha h^{\sigma-1/\sigma}$. Following (2), we end up to the following result:

$$s = \frac{1}{A}\alpha h^{\sigma-1/\sigma} = \frac{1}{3(k^{-3}+1)}$$



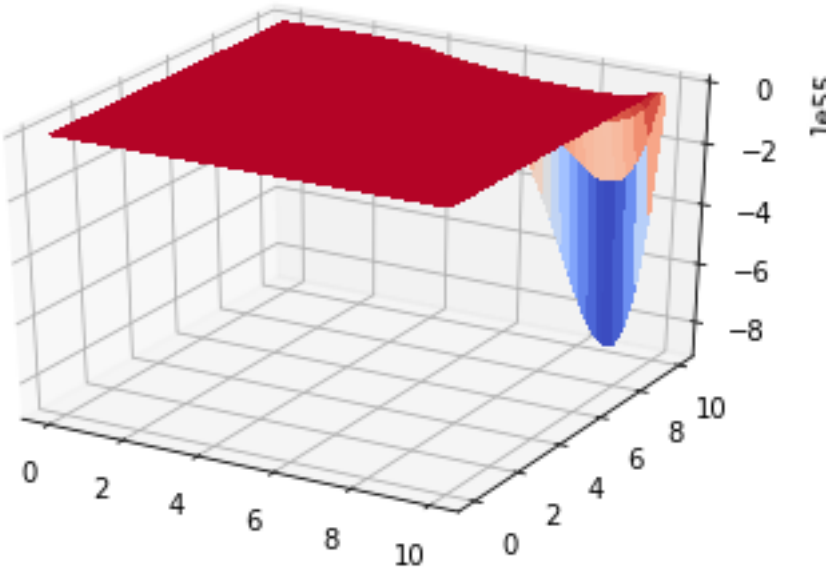
3. Use chebycheff polynomial to approximate a CES function:

CES FUNCTION WITH LABOUR AND CAPITAL AS INPUTS.



X exes is Labour, Y exes is capital and Z exes is Output.

APROXIMATION OF CES FUNCTION WITH CHEBYCHEFF POLYNOMIAL INTERPOLATION AND CHEBYCHEFF NODES.



X exes is Labour, Y exes is capital and Z exes is Output.

Obviously it is not a correct approximation, I will explain my steps to clarify what I tried to code, to show that my intuition was not really far a way of what I should have done.

I wanted first to code a function where you give a data of x, a data of y, and a degree n and the function give a vector with all the possible combinations of the family of chebycheff polynomials. For example, this vector with a second degree would be: A = [1, x, y, xy] where x and y are the vector of data, and the matrix A is dimension N x 4 where N is the total of observations. The code of this part is in the lines 619-644 (I define functions to make the vectors of chebycheff polynomial), line 647-656 (I create the grid of chebycheff nodes, and I substitute this data into the functions, getting the matrix of data), line 657-660 (I substitute chebycheff grid into CES function getting an endogenous variable, and I use this endougenous var and the matrix of data and I make an OLS estimation getting the parameter p). 662-669 (I use this parameters and a grid in the space X-Y to fed the function chebval2d that give us the value of the chebycheff polynomial for that domain).

4. **Isoquantes:**