Quantitative Macroeconomics: Homework 2

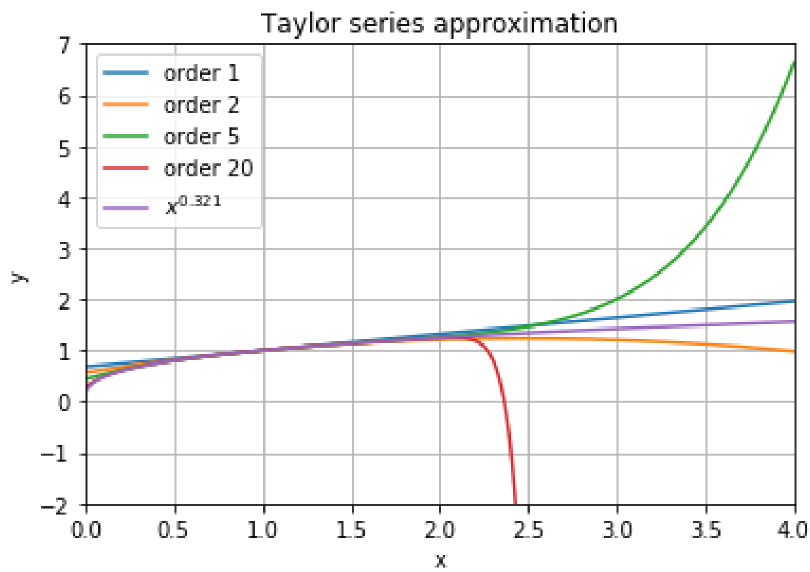
Marta Oliva Riera

1 Univariate function approximation

(1) Taylor series approximation of $x^{0.321}$

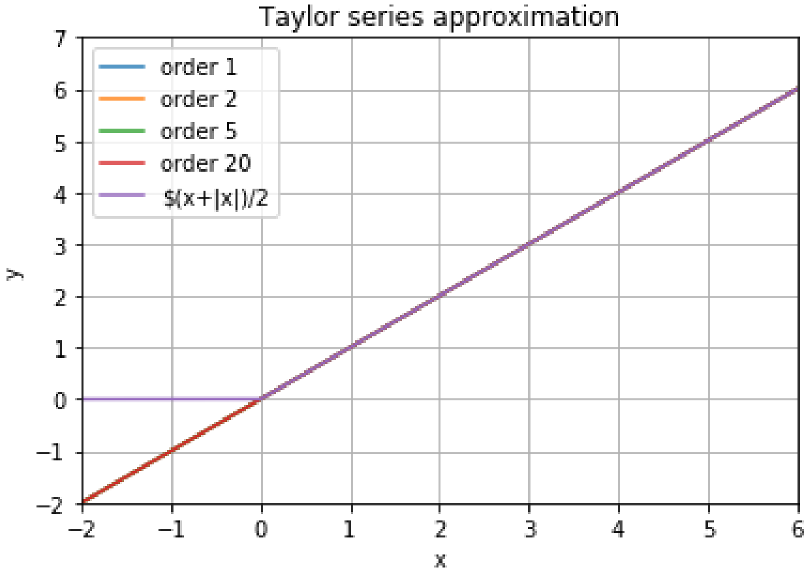
In the graph below are my results from approximating $x^{0.321}$ with Taylor expansion of order 1, 2, 5 and 20 around $\bar{x} = 1$. All of the approximations are very close to the real function around the specified point, but they deviate increasingly as the values of x get larger, which means the error of our approximation increases. We can see that the approximation of order 1 seems to be the closest to the real values in the domain $x \in [0, 4]$, while both the approximations of order 5 and 20 deviate greatly for $x > 3$ and $x > 2$ respectively.

Taylor expansion is a method of local approximation, and it cannot reliably approximate $f(x)$ at any point farther away from $\bar{x}$ than any singular point of the function. In this case, there is a singularity at $x = 0$, and that is why for $x > 2$ all approximations using Taylor start having large errors.



(2) Taylor series approximation of the ramp function

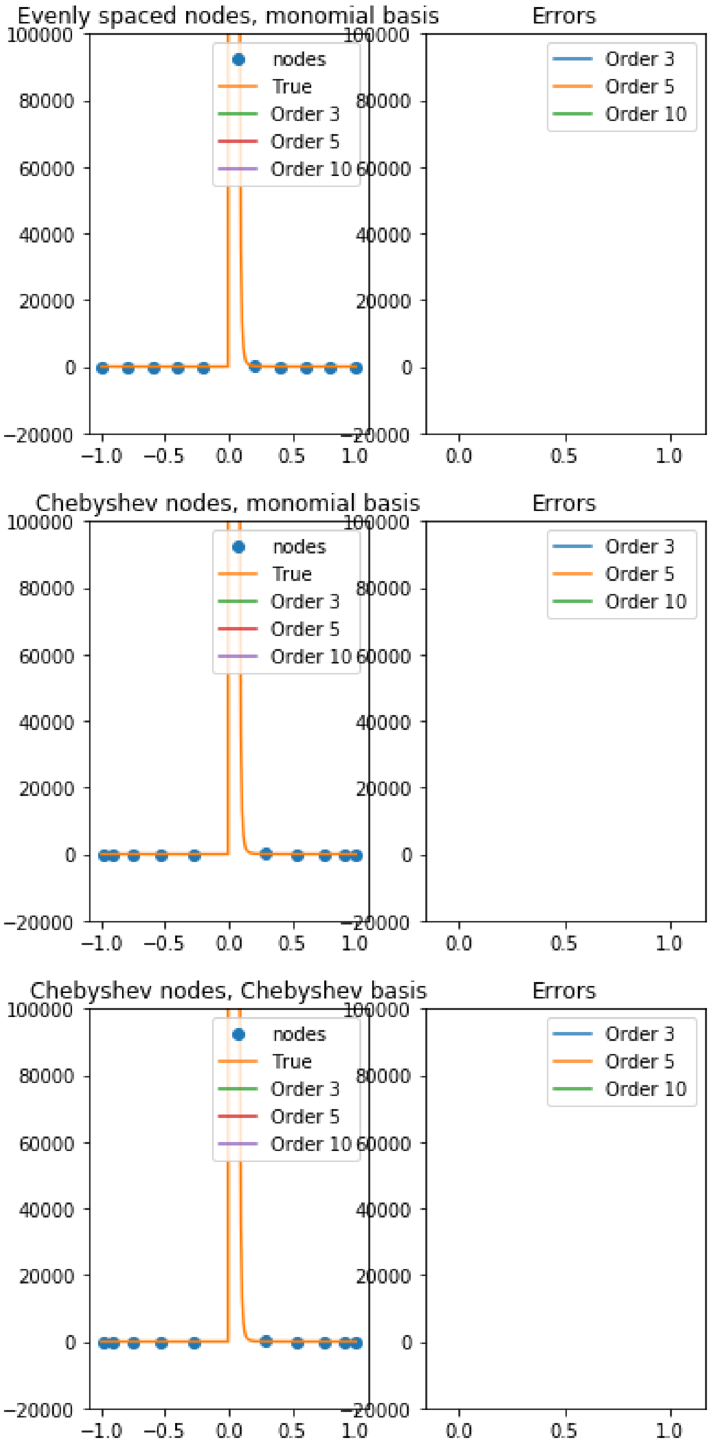
These are the Taylor approximations of the ramp function ($f(x) = \frac{x+|x|}{2}$) at $\bar{x} = 1$, of orders 1, 2, 5 and 20. All orders have the exact same shape: a straight line which is equal to the true function for $x > 0$ but misses the kink at $x = 0$. The kink cannot be represented by the approximations because the function is not differentiable at $x = 0$.



(3) Approximate an exponential function, the runge and the ramp function with different global methods

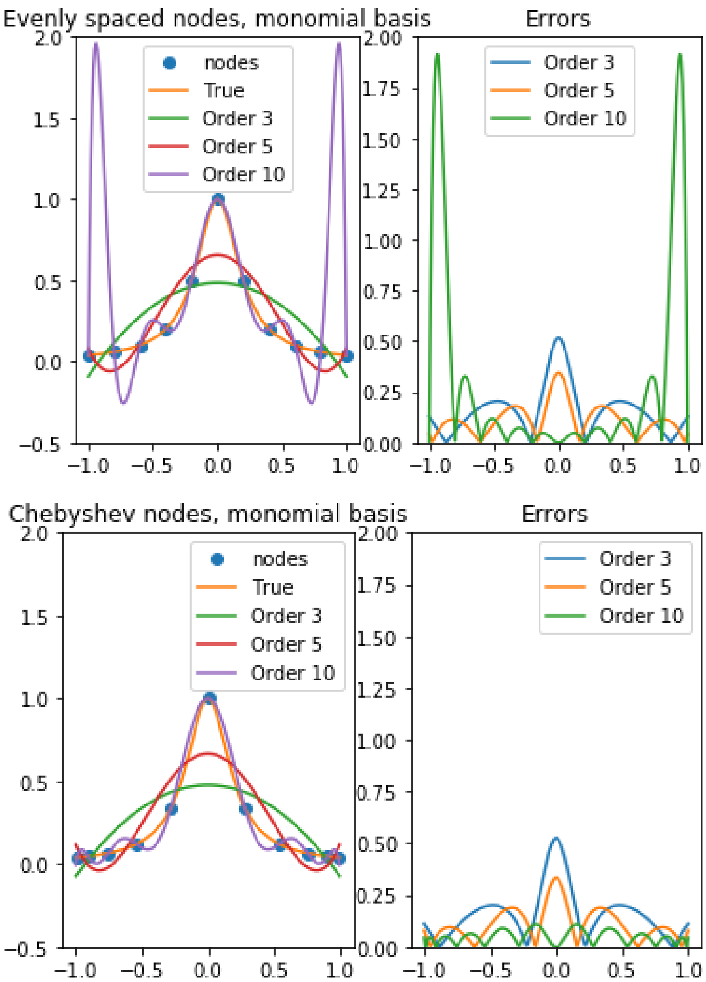
EXPONENTIAL FUNCTION:

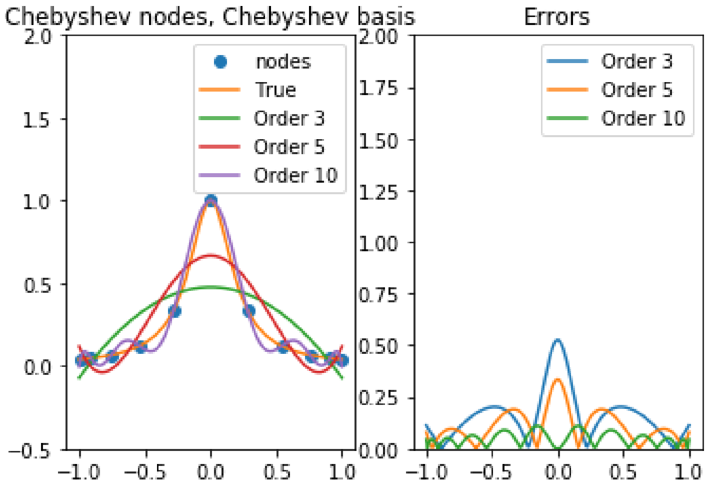
Below are the plots with my results from interpolating the exponential function $f(x) = e^{\frac{1}{x}}$. My results are not correct because it appears as if all interpolations (in both types of nodes and basis) are not computed at all or they are exactly equal to the function, resulting in no error. I could not find the error in my code, but I suspect it results from the discontinuity of the function at $x = 0$, as that's the only difference between this and the other two functions, for which the code works.


Evenly spaced nodes, monomial basis
Errors
Chebyshev nodes, monomial basis
Errors
Chebyshev nodes, Chebyshev basis
Errors
nodes
True
Order 3
Order 5
Order 10
100000
80000
60000
40000
20000
0
−20000
−1.0
−0.5
0.0
0.5
1.0

RUNGE FUNCTION:

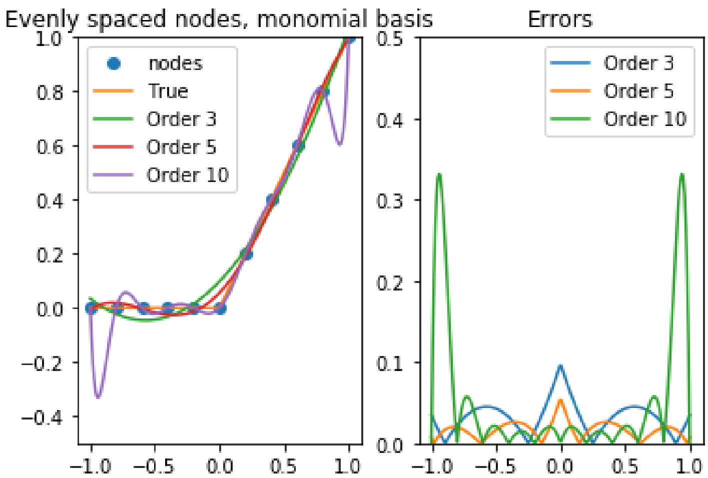
Below are the interpolations of the runge function, $f(x) = \frac{1}{1+25x^2}$. Using evenly spaced nodes and monomial basis results in large errors at the tails, specially noticeable in the order 10 interpolation. Except for the large error at the tails, the order tail interpolation seems to be the closest to the true runge function. Using Chebyshev nodes instead improves the interpolation error at the tails, as can be seen in the second and third rows of plots. However, there does not seem to be much difference in using monomial basis or Chebyshev polynomials, as those interpolations look almost the same.

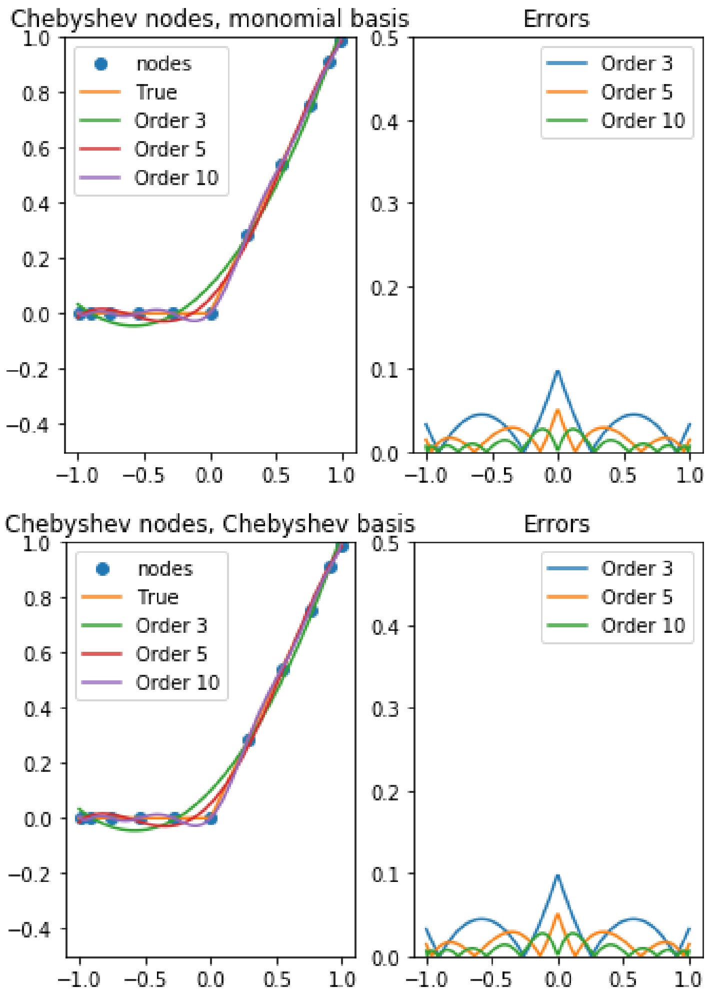




RAMP FUNCTION:

Here are the different interpolations of the ramp function, $f(x) = \frac{x+|x|}{2}$. Using evenly spaced nodes results in larger errors in the tails of the interpolation, especially in the order 10 one. The rest of orders have quite small errors. When using Chebyshev nodes instead of evenly spaced ones, the large errors at the tails fade away (as Chebyshev nodes are more concentrated at the tails) and the approximations are closer to the true function. Both approximations with Chebyshev nodes (one using monomial basis and the other using Chebyshev polynomials) have quite small errors. These are noticeably closer to the true function than the taylor approximation in the previous question was, as now they are able to better approximate the function for values of the domain below the kink.





2 Multivariate function approximation

Constant elasticity of substitution function:

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

Show analytically that σ is the elasticity of substitution:

The elasticity of substitution for our function is defined in the following way:

$$\sigma_{hk} = \frac{\partial ln(h/k)}{\partial ln MRTS_{kh}} = \frac{\partial ln(h/k)}{\partial ln \frac{\partial f(k,h)}{\partial k} / \frac{\partial f(k,h)}{\partial h}}$$

where $MRTS_{kh} = \frac{1-\alpha}{\alpha}(\frac{k}{h})^{\frac{-1}{\sigma}} = \frac{1-\alpha}{\alpha}(\frac{h}{k})^{\frac{1}{\sigma}}$, such that:

$$\sigma_{hk} = \frac{\partial ln(h/k)}{\partial ln \frac{1-\alpha}{\alpha}(\frac{h}{k})^{\frac{1}{\sigma}}} = \frac{\partial ln(h/k)}{\partial \frac{1}{\sigma}\frac{1-\alpha}{\alpha} ln(\frac{h}{k})}$$

Compute labour share for an economy with that CES production function assuming factor inputs face competitive markets:

The labour share is computed as workers' income over total production ($LS = \frac{hw}{f(k,h)}$). In a competitive market for labour, wages are determined as $\frac{\partial f(k,h)}{\partial h}$, such that the labour share is the following:

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

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

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

Approximate the CES function using a 2-dimensional Chebyshev regression algorithm:

I set up the code to do this but was unable to obtain results, as there are some errors in my code which I have not been able to fix.