Economics 202A
Macroeconomics
Fall 2022

Problem Set 7
Due by 11:59pm on November 8, 2022
Please upload solutions to Gradescope

You are allowed to work in groups. But please write up your own solutions. Please note that copying from old solutions constitutes plagiarism. Please write up your solutions as clearly as possible. Your grade will be reduced if your solution is unreasonably difficult to follow.

This is the first of three problem sets on numerical methods in this segment of the course. The second two numerical problem sets (problem sets 8 and 9) will ask you to solve two versions of the consumption-savings model we will be discussing in class. This problem set is a bit of a warm up. It covers some basic building blocks useful for numerical analysis. For those of you with a limited background in coding, just getting up to speed on using Matlab (or whatever other language you choose) will take some time. Sometimes the warm up is the hardest part.

This problem set and the other two on numerical methods are designed to complement lectures given by your GSI in sections. These lectures will cover parts of Alisdair McKay's numerical analysis notes, which are available here:

`https://alisdairmckay.com/Notes/NumericalCrashCourse/index.html#`.

and here

`https://alisdairmckay.com/Notes/HetAgents/index.html`

For those new to numerical analysis and value function iteration, I recommend that you follow closely the steps laid out in these notes. I have written the numerical problem sets as a set of step-by-step instructions assuming that you are basing your work on McKay's notes and code. However, you are welcome to solve the problems without reference to these steps and this code. If you do, please describe the steps you take and report the relevant code.

Function Approximation

Let's approximate the function $\log(x + y)$ using a set of polynomials.

a. Write a version of McKay's `PolyBasis` function for this problem. Use a second order polynomial basis.

b. Write a version of McKay's `PolyGetCoeff` function for this problem.

c. Create your main program and in this program create a grid for x and y. Do this in several steps. First, create separate grids for x and separately for y. Then, take a Cartesian product of the two.

Use 5 grid points for each dimension. Have the range of the grid be $[0.1, 2.0]$ for each dimension and have the grid points be evenly spaced. You can use the matlab functions `linespace` and `meshgrid` to do this. Reshape the outputs from `meshgrid` into two long vectors. The steps are described in

https://alisdairmckay.com/Notes/NumericalCrashCourse/VFI.html

Make sure you understand how Matlab stores and references arrays. This is also discussed in McKay's notes.

d. Calculate the true values of the function $\log(x + y)$ on the grid you created in part (c) – i.e., for each point on that grid – and place the results in a vector Z.

e. Use your `PolyGetCoeff` function to calculate the basis coefficients b for your polynomial approximation of $\log(x + y)$

f. Create a new "evaluation" grid for x and y. Follow the same steps as when you created the earlier grid, except that this time have 49 points for each dimension. Have these points be evenly spaced, but on a slightly bigger range $[0.1, 2.5]$.

g. Calculate the true value of the function $\log(x + y)$ on the evaluation grid and place the results in a vector `Zeval_true`. Use the basis coefficients b from part (e) and the `PolyBasis` function to calculate your polynomial approximation of the function $\log(x+y)$ on the evaluation grid and place the results in a vector `Zeval_approx`.

h. Plot `Zeval_true` and `Zeval_approx`. (Matlab makes nice 3D plots that you can rotate to explore the quality of the approximation. The relevant matlab function is `surf`.)

i. Calculate the maximum absolute value of the difference between the function (i.e., `Zeval_true`) and your approximation of the function (i.e., `Zeval_approx`). Do this both for the range [0.1,2.0] (i.e., the range of interpolation), and also for the full range including the part of the range where you are extrapolating (i.e., where you are beyond the range of points used to to calculate the basis coefficients b).

j. Plot the difference between `Zeval_true` and `Zeval_approx`.

k. Repeat this approximation exercise – i.e., parts (c) through (j) – with 15 and 35 points on each dimension of the grid used to calculate b. Plot these approximations together on the same figure.

l. Discuss what you have learned about the quality of polynomial approximations. (One paragraph.)

m. Redo this problem – i.e., parts (a) through (l) – for the function

$$f(x) = \begin{cases} 0 & \text{if} \quad x < 2 \\ (x-2)^{1/2} & \text{if} \quad x \geq 2 \end{cases}$$

on the range [0,4]. Again plot the three resulting approximations together on the same figure.