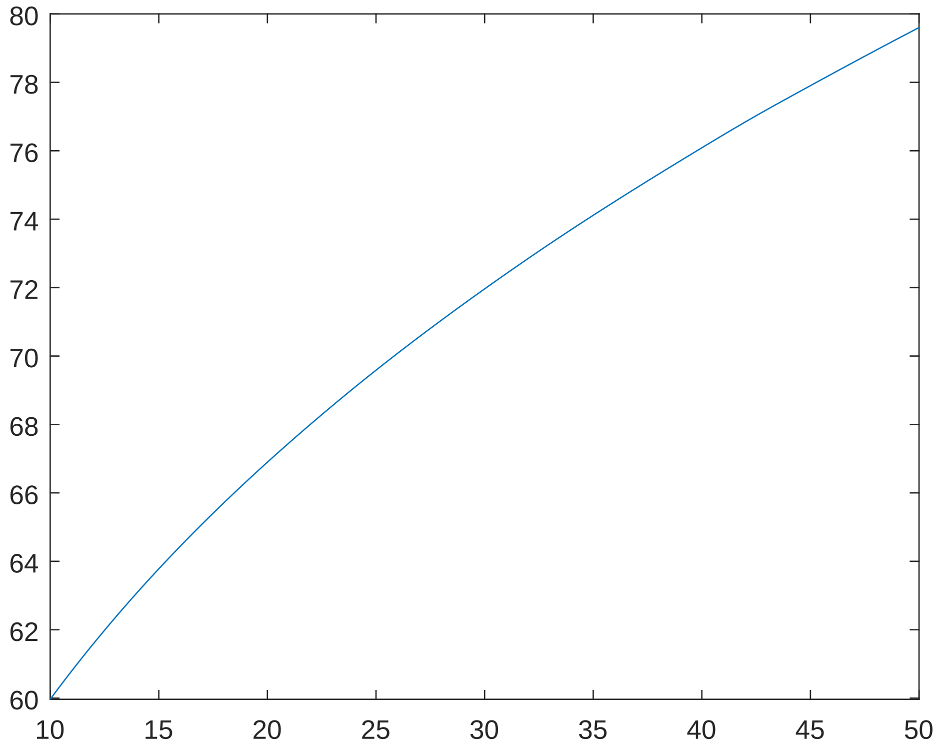
Quantitative Macro HW 3

Alexander Wurdinger

October 17, 2019

Q1

brute force



Doing all the steps described in the slides, I arrive 184 iterations at the plotted value function. The process took 0.117834 seconds. seconds.

Part b and c

Adding the described variation I arrive at the same value function that attains a fixed point. Number of iterations are also equivalent, but time needed to arrive at the fixed point is 0.280981 seconds. seconds respective 0.269514 seconds.

So while in theory those variations should speed up the process they seem to slow it down. For this specific problem this not to surprising as while in a to attain χ only two matrices have to be added, arguably a really fast process in matlab, in b and c χ is calculated in a loop.

If there are more grid points to be evaluated the speeding up processes might work, as then there are less entries per row to be evaluated in the max function when using the concavity or no maximization at all when using the monotonicity of the policy functions.

Part d

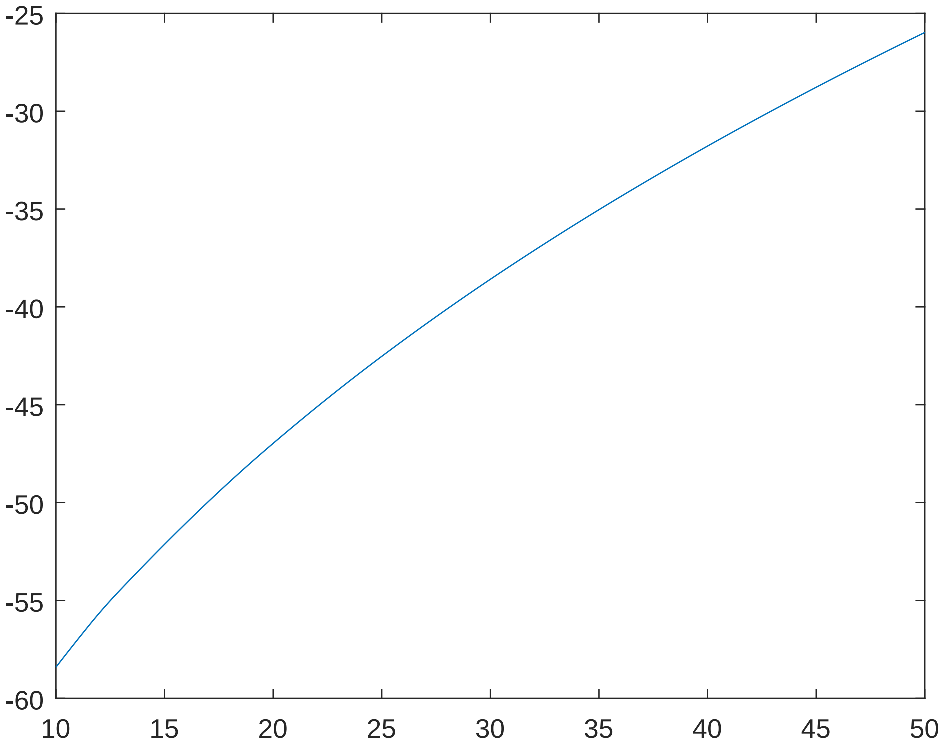
As using local search without using the monotonicity of the policy function does not seem really useful, I implemented both variants at the same time and the same equilibrium value function was achieved in the same amounts of iterations as before. Time spend was 0.273461 seconds. So faster then only using monotonicity.

It has to be noted that several runs of the program revealed that the bounds have to be set quite high (102) to avoid them being binding.

Part e,f

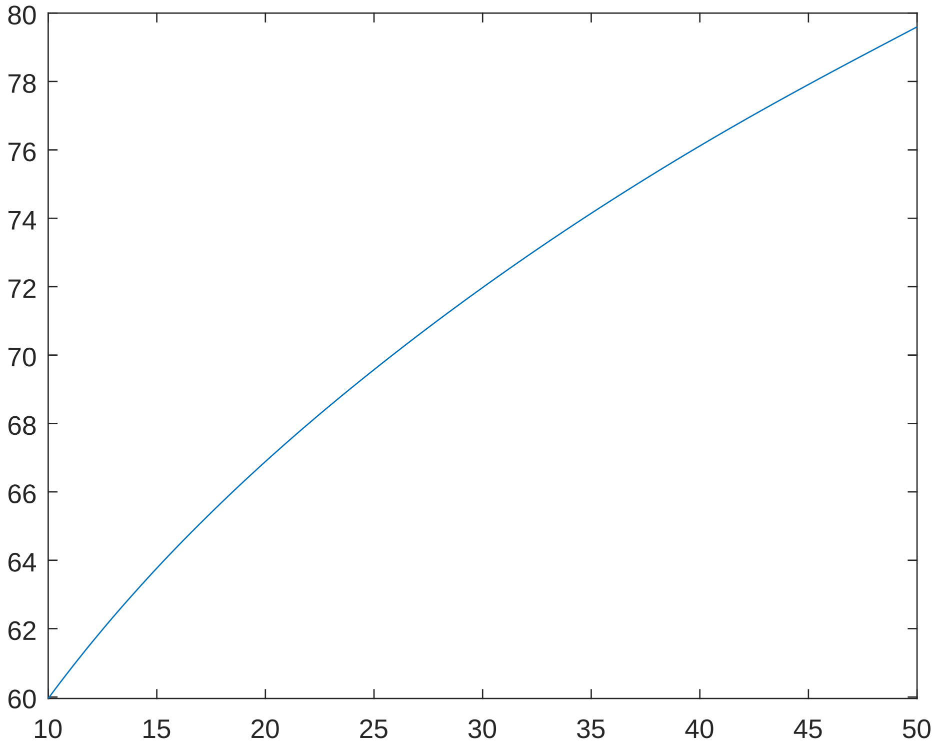
work in progress.

Q2



Same procedure as in Q1 with added labour supply.

Q3



As there are still problems with my chebychev approximation algorithm. I used a polynomial (degree 5) interpolation to approximate the value function in each iteration. The value function then converges. The whole process needs 6.373861 seconds and this therefore so far the slowest option in this example.