Grant Application

Felipe Alves
March 23, 2017

Abstract

These notes details the proposal to port the MATLAB code accompanying **?** *"Solving heterogeneous-agent models by projection and perturbation"* to Julia language. Although recent, Julia is spreading among economists doing computational economics for it allows to solve economic models at speeds that approach performance of C/Fortran while being much easier to learn and write.

1 Introduction

A growing literature in macroeconomics now concentrates on how heterogeneity matters for our understanding of aggregate quantities dynamics and their response to economics shocks. However, models that incorporate this heterogeneity usually are computationally challenging, specially because the distributions of agents, typically an infinite dimensional object, is part of aggregate state of the economy.

Since **?** original work a bunch of algorithms on how to solve these kind of models have been proposed and tested. Most recently, **?** method has stood out for its combination of *projection and perturbation methods*. Projection methods allow for nonlinearities of the decision rules with respect to idiosyncratic variables while solving the model dynamics to aggregate shocks remains computationally efficient due to the latter.

? illustrates his method in the context of the **?** model. Translating the original code written in MATLAB to Julia will allow us to have a working example of the method in the Julia language, making the extension to more complicated models easier due to Julia's performance gain.

1.1 Method Steps

At a broad level, the solution method involves three major steps

1. Approximate infinite dimensional equilibrium objects - including the policy functions and cross-sectional distribution of individual states - by finite dimensional objects.

2. Compute the stationary equilibrium of the discrete model. This will take into account the idiosyncratic uncertainty but not the uncertainty coming from aggregate shocks. This step usually involves
    - An **outer** loop over steady-state prices.
    - An **inner** loop computing (**i**) optimal decisions conditional on prices; (**ii**) the ergodic cross-sectional distribution of the individual state variables; (**iii**) checking market clearing conditions.
3. Linearize the discrete model equations around steady-state and compute the solution of the linearized system taking into account aggregate shocks.

Each one of these steps is computationally challenging. Steps 1, 2 are somewhat now standard on the literature and are easily implemented in Julia[1]. Step 3 is perhaps the most burdensome step. Although perturbation has been used extensively in representative agent DSGE literature, available tools/methods either require the model to be in the linearized form already or when they can perform this step from the full non-linear system, as is in the case of Dynare, it is not straightforward how to specify the equilibrium conditions for our heterogeneous agent model inside it.[2]

One alternative is to linearize the system by finite differencing. However, there are good computational reasons for not doing so. An attractive alternative available in Julia is to use ForwardDiff.jl package which implements methods to take derivatives using **forward mode automatic differentiation** (AD). AD consists of a set of techniques based on the mechanical application of the chain rule to obtain derivatives of a function given as a computer program. This is used in the original code, but the implementation of automatic differentiation is done through the definition of a new class using object-oriented capabilities of MATLAB. I expect Julia's implementation to be not only more efficient but also of easier comprehension and extension.

2 PROJECT

Specifically the project would involve a creation of

- GitHub repo with code performing the three steps for a **?** economy;
- Notes on the model and detailed description of each of the steps;
- Comparison with **?** algorithm (Maybe??)

[1] QUANTECON codes and lectures offer examples on how to set up the discretization scheme and solve agent's problem in classical economic models. Different representations of the distribution of agents are discussed in chapter 6 of **?**. Finally, finding equilibrium prices may require some root-finding method, which are also available in Julia - see Roots.jl and NLsolve.jl packages.

[2] See **?** for the application of the same solution method using *Dynare* in MATLAB.

The total amount of hours expected to be devoted to the project is 50 hours.

Trip Budget

In addition, the project also predict a 4-5 day trip to **Abu Dhabi** (United Arab Emirates) to meet and work with Reiter, who will be staying in NYU Abu Dhabi. Initially, the trip is planned to occur on early November and an initial budget including tickets and accomodation is between $**1500-1800 -** prices checked for dates Nov 9-13 in *Al Manzel Hotel Apartments* on *expedia.com* on September 3.