Fortran Code for "Worker Selection, Hiring, and Vacancies

Ismail Baydur
ADA University and CERGE-EI

April 13, 2016

1 Overview

This document explains the content of source files for the Fortran code used for numerical calculations in "Worker Selection, Hiring, and Vacancies". There are two separate files for each model: one for calibration and steady state simulations, and another one for policy analysis. Below I describe both of the source code files for the worker selection model. The source codes for the standard DMP and the directed search models have the same structure. They only differ with respect to the subroutines that describe the firm's optimization problem. Therefore, the explanations for the worker selection model also applies to these models.

The main source code must be supplemented with two other files: bobyqa.f and ssort.f. The first one of these files contains an algorithm developed by Michael J. D. Powell and is used to solve bound constrained optimization problems by quadratic approximation.[1] The algorithm is powerful in that it does not use derivatives of the objective function to solve this type of optimization problems. The source code can be freely downloaded from the following link:

http://mat.uc.pt/ zhang/software.html

The second file contains a selection sort algorithm written by John Mahaffy and it is used to organize simulated data in the main files. The original code can be freely downloaded from the following link:

http://www.personal.psu.edu/jhm/f90/examples/sort/sort1.f

All of these source codes are compiled using GNU Fortran compiler on a computer running Apple's operating system OS X (v. 10.9.5).

[1] Hence, the name for the solver is BOBYQA.

2 Calibration and Simulation

Calibration and steady state simulation results for the worker selection model can be obtained by compiling the source file called "ws_calib.f90" (along with bobyqa.f and sort.f files) and running the executable file. This main file contains a module for globally declared parameters and arrays, the main program, and many other subroutines that are called by the main program. The description of the global parameters are described in the module.

The main program starts with creating the grid for quadrature by calling the subroutine "lgwt". This grid is used to evaluate the Beta function that goes into the recruitment wages. It is then followed by initialization of the policy functions, the envelope condition, the stationary distribution and the grid for employment. In the current version, these are initialized at some random values. If the reader wants to use the numbers from a previous run, s/he can uncomment the lines that immediately follows initialization comments and read the files written for each variable. These files are automatically created after the first run of the program. The next step is to calculate the curvature parameter of the matching function. This is done by calling the subroutine "zbrent", which uses Brent's method to solve for the curvature parameter in the matching function for given values of market tightness and job finding probability.

The actual calibration is performed by calling the solver "bobyqa". The input function must be described in a subroutine called "calfun". In this subroutine, the first step is to initialize the model parameters to be calibrated. Then, the productivity grid and the transition matrix for idiosyncratic shocks are calculated using Tauchen's method. The next step is to solve the optimization problem of the firm and simulate the stationary distribution. Sum of squared errors, a measure of distance between the model and the data, is printed on the screen at the end of this subroutine. The solver iterates over the model parameter values until the sum of squared errors is equal to zero.

Finally, the main program simulates the model to generate monthly data and calculates vacancy yields, hires rates, and separation rates in the cross-section. The program ends with writing the output of the simulation exercises on file.

3 Policy Analysis

The policy analysis for the worker selection model is performed using the "ws_policy.f90" file. This program must also be compiled with the solver and sorting source files. As in the main file for the calibration section, the main file in this section also starts with a module containing the model parameters and some auxiliary variables. The model parameters are entered as fixed parameters and directly copied from the calibration results. As before, the policy functions, the envelope condition, the stationary distribution and the grid for employment are initialized.

The next step is to set the policy parameters equal to zero and recalculate the value of leisure, b, and calibrate the fixed entry cost, c_e, by iterating over the firm's value function. This step ensures better precision in the calculations that follow. Then, one can change the value of the policy parameters and calculate the new equilibrium by calling "solve_oq". The equilibrium is calculated by obtaining equilibrium values for the search value of unemployment, Ω, and the job finding probability, q, which together satisfy the free entry condition and are consistent with the worker's value function. To speed up the grid search process, I first calculate the upper and lower bounds for Ω and q, and then call "bobyqa" once again to solve for the equilibrium.

Finally, I simulate the model to obtain the aggregate and cross-sectional effects of the policy changes. The program ends with writing the output from policy exercises on file.

4 Miscellaneous Issues

- The source files for the standard DMP and directed search models are as follows: dmp_calib.f90, dmp_policy.f90, ds_calib.f90, and ds_policy.f90.
- The equilibrium in the directed search model is characterized solely by calculating the job search value, Ω. Hence, the Fortran code for the policy exercise uses "zbrent" rather than "bobyqa".

References

[1] G. Tauchen (1986): "Finite State Markov-Chain Approximations to Univariate and Vector Autoregressions", *Economics Letters*, vol. 20, pp. 177-181.