VFI Toolkit: Pseudocodes

Robert Kirkby*

August 28, 2023

Keywords: Value function iteration, Semi-endogenous shocks.

JEL Classification: E00; C68; C63; C62

*Kirkby: Victoria University of Wellington. Please address all correspondence about this article to Robert Kirkby at <robertdkirkby@gmail.com>, robertdkirkby.com, vfitoolkit.com.

Contents

1 Introduction

This document presents pseudocodes for various commands from VFI Toolkit. The actual implementations in VFI Toolkit often involve parallelized versions of the psuedocodes presented here, and also typically have options to use more-or-less parallelization (versus looping).

This document is a work-in-progress. Please request pseudocode for any specific command you want to see it for (either by email or on discourse.vfitoolkit.com).

Throughout I use the standard notation of VFI Toolkit: so d is a decision variable, a is endogenous state (and a' the next-period endogenous state), z is a markov exogenous state, e is an i.i.d. exogenous state, u is an i.i.d. shock that occurs between periods. V is the value function. For finite horizon problems there are N_j periods, indexed by j.

In many psuedocodes $g(a,z)$ denotes a policy, with $g^d(a,z)$ being the policy for the decision variables and $g^{a'}(a,z)$ being the policy for the next-period endogenous state.

2 Value Function Iteration: Solving for V and Policy

2.1 Infinite Horizon: decision variable, markov shock

Starting from an initial guess for the value function, we iterate on the value function until convergence. To speed things up Howards improvement is used.

Declare initial value V_0.
Declare iteration count $n = 0$.
while $||V_n - V_{n-1}|| >$ 'tolerance constant'
  Increment n. Let $V_{old} = V_{n-1}$.
  for A ll values z
    Calculate $E[V_{old}(a', z')]$
    for A ll values of a
      Calculate $V_n(a,z) = max_{d,a' \in D(a,z)} F(d,a',a,z) + \beta E[V_{old}(a',z')]$
      ... and keep the $argmax$ $g_n(a,z)$.
    end for
  end for
  if U seHowards==1
    for n Howards number of times
      Update $V_n(a,z) = F(g_n^d(a,z), g_n^{a'}(a,z), a, z) + \beta E[V_n(g_n^{a'}(a,z)', z')]$
    end for
  end if
end while

return V_n, g_n

Note that by making the outer-loop over z we are able to reduce the number of times we compute the expectations (which depend on z and not on a).

Howards improvement algorithm is to use the current policy to evaluate/update the value function a couple of times. This is faster as it skips the maximization step (which is computationally the most costly) and because the policy function typically is 'pointing' in the direction of the solution, so a few cheap computations to move further in this direction is worthwhile. $UseHowards$ is implemented as conditions under which Howards improvement algorithm should be used (which can be controlled using $vfoptions$. $nHowards$ is set to 80 by default (based on a trying out a bunch of different values, this seemed to be best at speeding up the 'average' problem). In practice, we avoid using Howards improvement algorithm for the first few iterations (as the initial guess is likely poor, and so Howards does not much help), and also stop using Howards improvement algorithm once we are within one order of magnitude of convergence of the value function.[1]

A couple of hints to how this could be done better (the kind of improvements that work for just about any algorithm, rather than things like using endogenous grid method instead of pure discretization): ideally it would use the improved versions of Howards developed in Rendahl (2022), Bakota and Kredler (2022) and Phelan and Eslami (2022) (I've not tried these). And it would use relative VFI, or even endogenous VFI (for these later two, they were tried but don't work in VFI Toolkit because of the pure discretization); Bray (2017).

2.2 Finite Horizon: decision variable, markov shock

Just some simple backward iteration. Let Θ be all the parameters of the model, and let θ_j be the values of those parameters which are relevant in period j.[2]

Solve for final period, N_j: $V_{N_j}(a,z) = max_{d,a' \in D(a,z)} F(d,a',a,z)$
... and keep the argmax g_{N_j}.
for j count backward from $N_j - 1$ to 1
  Get θ_j from Θ
  for A ll values z
    Calculate $E[V_{j+1}(a',z')]$
    for A ll values of a
      Calculate $V_j(a,z) = max_{d,a' \in D(a,z)} F(d,a',a,z) + \beta E[V_{j+1}(a',z')]$

[1] This is because otherwise Howards leads to very small errors. You won't notice them normally, but when trying to compute a transition path between stationary equilibrium they are problematic as the transition won't use Howards and so tries to go to a very slightly different place. If you are thinking 'but I saw the proof in Sargent & Stachurski that Howards is fine and gives same answer' the trick is that the last line of their proof assumes that Howards evaluates the 'policy greedy' value function, but in practice you will never do this.

[2] So if a parameter is independent of age, then it is just in θ_j as is. If a parameter depends on age, then only the value relevant to age j is in θ_j.

... and keep the *argmax* $g_j(a, z)$.

end for

end for

end for

return $[V_1, V_2, ..., V_{N_j}]$, $[g_1, g_2, ..., g_{N_j}]$

If you were writing custom code for a specific problem, you would likely try to take advantage of things like monotonicity, but because VFI Toolkit is trying to solve generic problems it just uses this simple robust approach.

References

Ivo Bakota and Matthias Kredler. Continuous-time speed for discrete-time models: A markov-chain approximation method. MEA Discussion Papers, 2022. doi: https://dx.doi.org/10.2139/ssrn.4155499.

Robert Bray. Markov decision processes with exogenous variables. Management Science, 2017. doi: https://doi.org/10.1287/mnsc.2018.3158.

Robert Kirkby. VFI toolkit, v2. Zenodo, 2022. doi: https://doi.org/10.5281/zenodo.8136790.

Thomas Phelan and Keyvan Eslami. Applications of markov chain approximation methods to optimal control problems in economics. Journal of Economic Dynamics and Control, 2022. doi: https://doi.org/10.1016/j.jedc.2022.104437.

Pontus Rendahl. Continuous vs. discrete time: Some computational insights. Journal of Economic Dynamics and Control, 2022. doi: https://doi.org/10.1016/j.jedc.2022.104522.