Errata for Phelan and Eslami (2022)

Thomas Phelan
Federal Reserve Bank of Cleveland

March 22, 2023

This document discusses errata for Phelan and Eslami (2022). None of the following affect the description of the algorithm or the figures produced in the text but do affect the reported run times. However, these are already subject to some idiosyncratic variation across runs, which is why we simply emphasized in Phelan and Eslami (2022) that MPFI can increase the speed of convergence relative to PFI by "over an order of magnitude". As the following tables show, for the corrected code this conclusion remains true and relative speeds are largely unchanged. I have updated the code on the github page linked to the published paper[1] and I have moved the original code into a separate repository for transparency.[2] In this note I describe here the problems and the changes to the run times. If you have any questions please email me at tom.phelan@clev.frb.org.

(i) **Erronenous boundary probability in IFPs.** In the IFPs, the code sometimes did not have zero up transitions at the upper boundary for log income due to a rounding problem. The original code defined grids in z_i with $N_i - 1$ points between $\underline{z}_i + \Delta_i$ and $\overline{z}_i - \Delta_i$ inclusive. Volatility was defined to be nonzero if $\underline{z}_i + \Delta_i < z_i < \underline{z}_i + (N_i - 1)\Delta_i$, and so sometimes failed to vanish at the upper boundary when Δ_i was a recurring decimal. This occurred for the 3D problem with grid (90,30,30), where $\overline{z}_i - \Delta_i = \underline{z}_i + (N_i - 1)\Delta_i = 0.74\overline{6}$ and python declared $\overline{z}_i - \Delta_i < \underline{z}_i + (N_i - 1)\Delta_i$ to be true.

Since the transition probabilities didn't always sum to unity, for this grid the original code produced an erroneous upwards spike in consumption at points near the boundary where the solution became inaccurate.[3] Correcting the code has little effect on relative run times because the error was consistent across PFI and MPFI and still leads to a well-defined control problem: it is equivalent to assuming that for the highest income level the consumer discounts at a higher rate or faces a probability

[1] Found at https://github.com/tphelanECON/EslamiPhelan_MCA

[2] Found at https://github.com/tphelanECON/EslamiPhelan_JEDC_2022.

[3] This necessitates no change in the text though as this output was not used in the paper.

of death. The corrected code defines volatilities with a statement of the form (self.jj > 0) * (self.jj $< N_z - 2$) for the grid indices self.jj to avoid any comparison of floats.

(ii) **Upper bound in IFPs.** For a sufficiently high upper bound on assets the state constraint will not bind when the discount rate exceeds the interest rate. In the original code this upper bound was sometimes too small to ensure this. For both the 2D and 3D IFPs the upper bound was chosen to be $\overline{a} = 60$, which did not bind for the 2D IFPs but did bind for the 3D IFPs when $z_1 + z_2$ exceeded roughly 1.066. Since the stationary distribution of $z_1 + z_2$ is normal with mean zero and variance $\sqrt{2}\nu \approx 0.28$, this occurs less than 0.05% of the time for the continuous-state process. The corrected code uses a higher upper bound ($\overline{a} \approx 170$) at which the agent wishes to dissave.

Table 1 and Table 3 record the run times for the 3D problem exactly as they appear in the paper. Table 2 and Table 4 record the run times (average of ten runs, same parameters as the 3D problem in the paper) for the corrected code. In Phelan and Eslami (2022) we emphasized that MPFI can easily be an order of magnitude faster than PFI for the same discretization, and this remains true for the corrected code. Table 1 and Table 2 shows that for this set of runs the corrected code appears to be slightly faster for MPFI for all grids, and that the relative speeds of PFI and MPFI remain largely unchanged. This is to be expected because the higher upper bound for assets implies that the timestep is now larger, which increases the speed of convergence. Similar comments apply to the comparison between the generalized algorithms given in Table 3 and Table 4, although here there is no obvious relationship (at least to us) between the speed of convergence and changes in the grid size for assets. The run times in Table 4 for the corrected code are sometimes faster and sometimes slower than the original code, depending on the grid and relaxation, and the MPFI with $k = 200$ remains over an order of magnitude faster than PFI.

Grid size	PFI	VFI	$k = 10$	$k = 50$	$k = 100$	$k = 200$
(45, 15, 15)	0.762	7.410	0.847	0.249	0.195	0.158
(60, 20, 20)	4.828	23.362	2.652	0.797	0.540	0.420
(75, 25, 25)	20.045	58.819	6.635	1.944	1.237	1.088
(90, 30, 30)	68.306	110.348	12.535	3.757	2.432	2.062

Table 1: Time until convergence (original code): MPFI

Grid size	PFI	VFI	$k = 10$	$k = 50$	$k = 100$	$k = 200$
(45, 15, 15)	0.633	6.189	0.690	0.201	0.157	0.132
(60, 20, 20)	4.352	19.687	2.207	0.631	0.454	0.363
(75, 25, 25)	21.547	48.187	5.447	1.584	1.041	0.816
(90, 30, 30)	62.523	102.525	11.645	3.370	2.291	1.861

Table 2: Time until convergence (corrected code): MPFI

Grid size	PFI	$k = 0$	$k = 10$	$k = 50$	$k = 100$	$k = 200$
(45, 15, 15)	0.902	9.763	1.103	0.363	0.281	0.330
(60, 20, 20)	4.924	33.238	3.650	1.136	0.868	0.722
(75, 25, 25)	21.632	94.989	10.582	2.956	1.997	1.752
(90, 30, 30)	67.251	203.795	22.390	6.432	4.112	3.398

Table 3: Time until convergence (original code): Generalized MPFI

Grid size	PFI	$k = 0$	$k = 10$	$k = 50$	$k = 100$	$k = 200$
(45, 15, 15)	0.752	11.049	1.233	0.385	0.296	0.271
(60, 20, 20)	5.431	39.245	4.318	1.243	0.806	0.761
(75, 25, 25)	20.293	102.429	11.306	2.976	1.981	1.431
(90, 30, 30)	70.851	229.303	25.217	6.854	4.129	2.862

Table 4: Time until convergence (corrected code): Generalized MPFI

References

Phelan, T. and Eslami, K. Applications of Markov chain approximation methods to optimal control problems in economics. *Journal of Economic Dynamics and Control*, 143:104437, October 2022. doi:10.1016/j.jedc.2022.104437.