Replication Package: Computational Environment Analysis for Mahajan et al (2025)

Aprajit Mahajan

October 3, 2025

Contents

1 Executive Summary

We have conducted a systematic analysis of computational differences affecting standard error calculations in Tables 3-5. Our investigation reveals that **parameter estimates are robust across all computational environments** (with maximum differences $\sim 10^{-5}$). However, the **standard errors differ** because the minor differences in parameter estimates appear to be amplified through Hessian matrix calculations and this variation depends upon the precise hardware and software configuration.

Key Finding: The root cause in standard error differences is differences in Intel Math Kernel Library (MKL) Conditional Numerical Reproducibility (CNR) instruction set architecture across computing environments. **All Linux systems with MATLAB versions R2022a or earlier (that we tested) using CNR branch AVX produce numerically identical results to our original submission**, regardless of toolbox versions or MKL versions.

Critical Evidence:

- **Linux_2020b** (CNR branch AVX): Original submission results
- **Linux_2021b** (CNR branch AVX): **Identical results**
- **Linux_2022a** (CNR branch AVX): **Identical results**
- **All other systems** (CNR branch AVX2/SSE4_2): Different results but consistent within branch

Thus our original submission environment produces replicable results across any computational environment with the same instruction set architecture on Linux (with MATLAB versions R2022a or earlier that we tested). Later versions of MATLAB on Linux have deprecated Intel AVX support in the Math Kernel Library, automatically falling back to SSE4_2 instructions, which produces different numerical results. In some environments it is possible to instruct MATLAB to use AVX via environment variables set before starting MATLAB (however, this does not work with newer MATLAB versions where Intel has deprecated AVX support).

We recommend replicators use a Linux environment with MATLAB R2022a or earlier with CNR AVX. The replication package includes instructions for MATLAB to use AVX via setting environment variables in Linux environments. The package also includes hard-coded parameter estimates (corresponding to the CNR branch AVX results) in the file `benchmark_estimates.mat` in the `outputdata/` folder so that replicators can compare their environment's output to the benchmark values. Note: Environment variable forcing to use AVX instructions may not work on all systems, particularly with MATLAB versions newer than R2022a where Intel has deprecated AVX support.

2 Problem Statement

The RES data editor reported inability to replicate standard errors for Tables 3-5. Our original submission used Ubuntu 24.04 with MATLAB R2020b, but replication attempts on different systems yielded different standard errors despite nearly identical parameter estimates.

3 Systematic Testing Methodology

We implemented a comprehensive testing framework across 7 computational environments:

3.1 System Reference Key

For replication transparency, we tested on the following systems. Throughout this document, systems are identified by their OS and MATLAB version (e.g., "Linux 2020b"):

- **Linux_2020b**: Ubuntu 24.04, MATLAB R2020b (original submission environment)
- **Linux_2021b**: Ubuntu, MATLAB R2021b
- **Linux_2022a**: Ubuntu, MATLAB R2022a
- **Linux_2024b**: Ubuntu 24.04, MATLAB R2024b
- **Mac_2020b**: macOS, MATLAB R2020b
- **Mac_2024b**: macOS, MATLAB R2024b
- **Windows_2024b**: Windows, MATLAB R2024b

Internal system hostnames have been removed for privacy; the technical specifications (OS, MATLAB version, toolbox versions, and MKL information) provided throughout this document are sufficient for replication purposes.

3.2 Testing Protocol:

1. **Systematic Tests**: Full optimization on each system using identical code
2. **Fixed Parameter Tests**: Standard error computations at fixed $\hat{\theta}$ value (set equal to the submitted estimates)
3. **Toolbox Documentation**: Complete version tracking for all systems
4. **BLAS/LAPACK Analysis**: Underlying mathematical library identification

4 Key Findings

4.1 Parameter Estimates

All systems produce nearly identical parameter estimates with maximum differences at the 10^{-5} level. However, our systematic testing reveals a clear pattern for **exact replication**:

Exact Replication of Submitted Results:

- **Linux_2020b** (CNR branch AVX): Original submission results
- **Linux_2021b** (CNR branch AVX): **Identical to submission results**
- **Linux_2022a** (CNR branch AVX): **Identical to submission results**

Key: Intel MKL using AVX instruction set produces exact replication across different Linux systems for MATLAB versions R2022a or earlier.

Different from Submitted Results (but identical within group):

- **Mac systems** (CNR branch AVX2):
- **Windows systems** (CNR branch AVX2):

Single System

- **Linux_2024b** (CNR branch SSE4_2): Different results from previous two sets (Note AVX deprecation in newer MKL)

The instruction set architecture thus seems to be the determining factor for exact replication on Linux system. Operating system differences (Linux vs Mac vs Windows) create additional barriers that cannot be overcome by instruction set forcing alone.

It is worth noting that all economic conclusions remain robust across all computational environments, with differences occurring only at the level of numerical precision in optimization convergence.

4.1.1 CNR Branch explanation:

- CNR (Conditional Numerical Reproducibility) refers to Intel MKL's adaptive selection of CPU instruction sets based on available hardware capabilities. Intel MKL automatically detects the CPU at runtime and chooses the most advanced instruction set the processor supports.
- Systems may have different CNR branches because of:
  - Hardware Detection: Intel MKL queries the CPU's capabilities and selects the "best" available instruction set
  - CPU Age/Architecture: Older CPUs support fewer instruction sets than newer ones
  - Vendor Differences: Intel, AMD, and Apple Silicon have different instruction set support
  - Platform Differences: Linux, macOS, and Windows may have different capabilities
  - Different Instruction Sets: (a) SSE4 (oldest, most compatible), (b) AVX (advanced, introduced 2011) (c) AVX2 (newer, introduced 2013).
  - Why This Causes Numerical Differences: Intel MKL uses different algorithms and precision handling for each instruction set. Even though all comply with IEEE floating-point standards, the specific implementation details (rounding order, vectorization patterns, memory alignment) differ subtly between instruction sets. These tiny differences (at machine precision level) get amplified through iterative optimization algorithms and Hessian matrix calculations.

4.2 Standard Errors

Standard error differences arise because small differences in parameter estimates (at the 10^{-5} level) get amplified through the Hessian matrix calculation.

Exact Standard Error Replication Achieved:

- **Linux systems using Matlab $\leq$ R2022a with forced AVX**: Produce identical standard errors to original submission
- **Fixed parameter validation**: All systems produce identical standard errors when using the exact same parameter estimates

Implication: Standard error differences occur because different computational environments find slightly different optimization solutions (differing by ~10^{-5}), and these tiny differences get magnified when computing the Hessian matrix for standard errors.

4.2.1 Interpretation:

The issue is **optimization path sensitivity**.

1. **Different optimization paths**: Each system's numerical libraries guide the optimization algorithm along slightly different paths to the solution
2. **Convergence to nearby points**: All systems converge to nearly identical parameter estimates (differences ~10^{-5})
3. **Hessian amplification**: When computing standard errors, the Hessian matrix calculation amplifies these tiny parameter differences into larger standard error differences
4. **Proof of soundness**: When we force all systems to use identical parameter values, they compute identical standard errors

5 Technical Analysis:

5.1 Intel MKL Instruction Set Impact

5.1.1 CNR Branch Explanation:

- **CNR**: Conditional Numerical Reproducibility
- **AVX**: Advanced Vector Extensions (256-bit operations)
- **AVX2**: Enhanced Advanced Vector Extensions
- **SSE4_2**: Streaming SIMD Extensions 4.2 (128-bit operations)

5.1.2 Why This Causes Differences:

Sensitivity to Instruction Set Architecture (ISA) is documented in the literature, see e.g. Intel Corporation (2023); Blackford et al. (2002); Goldberg (1991)

1. **Floating-point precision handling** varies across instruction sets
2. **Vectorization strategies** affect how matrix operations are batched
3. **Rounding behaviors** differ in numerical linear algebra routines
4. **Optimization convergence paths** vary due to numerical precision differences

5.2 Evidence Summary

5.2.1 Computational Environment Analysis:

System	MATLAB	MKL Version	CNR Branch	Max Param Diff	Max SE Ratio
Linux_2020b	R2020b	2019.0.3	AVX	(submitted version)	(submitted version)
Linux_2021b	R2021b	2019.0.3	AVX	0	1.000
Linux_2022a	R2022a	2021.3	AVX	0	1.000
Linux_2024b	R2024b	2024.1	SSE4_2	1.66e-05	2.202
Mac_2020b	R2020b	2019.0.3	AVX2	4.85e-06	5.835
Mac_2024b	R2024b	2023.2.2	AVX2	4.85e-06	5.835
Windows_2024b	R2024b	2024.1	AVX2	4.85e-06	5.835

Note: Max Param Diff = maximum absolute difference in parameter estimates compared to Linux_2020b reference; Max SE Ratio = maximum ratio of standard errors compared to Linux_2020b reference.

Key: Linux systems with CNR branch AVX and MATLAB R2022a or earlier are expected to produce identical results (verified for R2020b, R2021b, and R2022a), regardless of toolbox versions, or MKL versions. The instruction set architecture appears to be the determining factor.

6 Implications

6.1 Robustness

- **Parameter estimates are robust**: Maximum differences at 10^{-5} level. No economic significance of differences in estimates.
- **Economic conclusions remain valid** across all computational environments
- **Model specification not an issue**: Issue is purely computational, not methodological

6.2 Standard Error Computations

- **Inference framework is correct**: Computing standard errors for a fixed point estimate yields identical standard errors across systems
- **Standard error amplification**: Small optimization differences (10^{-5}) amplify to large SE differences through Hessian sensitivity
- **Computational dependency**: Results appear to depend on underlying mathematical libraries, not MATLAB software versions

6.3 Replicability Implications:

- **Computational environment specification** is crucial for numerical reproducibility. In our case we specify Matlab versions R2022a or earlier.
- **Intel MKL instruction set** should be documented in replication packages. In our case we specify AVX. As noted above, this can be set on some systems with some versions of Matlab.

7 Conclusion

7.1 Replication:

- **Linux systems with CNR branch AVX** and Matlab $\leq$ R2022a produce identical results in our testing
- **Environment variable solution** on compatible Linux systems can use 'export MKL_ENABLE_INSTRUCTION

7.2 Practical Replication Strategy

We recommend a tiered approach for replicators:

7.2.1 Primary Method (Exact Replication):

- Linux operating system with MATLAB R2022a or earlier
- Force Intel MKL to use AVX instruction set via environment variables (as indicated above)
- This approach works across different computers and MATLAB versions

7.2.2 Alternative Method (Validation):

- Any computational environment with hard-coded parameter estimates from reference system
- Compare optimization results to benchmark values to understand system-specific differences
- Fixed parameter validation confirms identical inference capabilities across systems

7.2.3 Documentation Package:

- Provide exact parameter estimates from CNR branch AVX reference environment
- Include environment variable instructions for Linux users
- Document computational environment specifications for transparency

A Technical Implementation

A.1 Code Availability

Complete systematic testing framework available:

- `table3_col1_systematic.m`: Comprehensive system testing
- `table3_col1_fixed.m`: Fixed parameter validation
- `compare_col1_results.m`: Multi-system analysis
- `quick_toolbox_check.m`: Computational environment documentation

A.2 Replication Materials

All results reproducible using provided systematic testing code on respective computational environments.

A.3 System Specifications

System	MATLAB	OS	Arch	Java Version
Linux_2020b	2020b	Linux	glnxa64	Java 1.8.0_202-b08
Linux_2021b	2021b	Linux	glnxa64	Java 1.8.0_202-b08
Linux_2022a	2022a	Linux	glnxa64	Java 1.8.0_202-b08
Linux_2024b	2024b	Linux	glnxa64	Java 1.8.0_202-b08
Mac_2020b	2020b	Mac	maci64	Java 1.8.0_202-b08
Mac_2024b	2024b	Mac	maci64	Java 1.8.0_202-b08
Windows_2024b	2024b	Windows	win64	Java 1.8.0_202-b08

A.4 Critical Toolbox Versions

System	Optim_TB	GlobalOpt_TB	Stats_TB	Econom_TB
Linux_2020b	9.0	4.4	12.0	5.5
Linux_2021b	9.2	4.6	12.2	5.7
Linux_2022a	9.3	4.7	12.3	6.0
Linux_2024b	24.2	24.2	24.2	24.2
Mac_2020b	9.0	4.4	12.0	5.5
Mac_2024b	24.2	24.2	24.2	24.2
Windows_2024b	24.2	24.2	24.2	24.2

A.5 BLAS/LAPACK Library Information

System	BLAS Info
Linux_2020b	Intel(R) Math Kernel Library Version 2019.0.3, CNR branch AVX
Linux_2021b	Intel(R) Math Kernel Library Version 2019.0.3, CNR branch AVX
Linux_2022a	Intel(R) oneAPI Math Kernel Library Version 2021.3, CNR branch AVX
Linux_2024b	Intel(R) oneAPI Math Kernel Library Version 2024.1, CNR branch SSE4_2
Mac_2020b	Intel(R) Math Kernel Library Version 2019.0.3, CNR branch AVX2
Mac_2024b	Intel(R) oneAPI Math Kernel Library Version 2023.2.2, CNR branch AVX2
Windows_2024b	Intel(R) oneAPI Math Kernel Library Version 2024.1, CNR branch AVX2

A.6 Parameter Estimates (High Precision, α parameters scaled)

Table 1: Parameter Estimates (High Precision, α parameters scaled)

Param	Linux_2020b	Linux_2021b	Linux_2022a	Linux_2024b	Mac_2020b	Mac_2024b	Windows_2024b
δ	0.998998786871	0.998998786871	0.998998786871	0.998993934761	0.998993934532	0.998993934532	0.998993934532
β_n	0.058070730104	0.058070730104	0.058070730104	0.058082116067	0.058070582983	0.058070582983	0.058070582983
β_s	0.155020752431	0.155020752431	0.155020752431	0.155004108118	0.155023852041	0.155023852041	0.155023852041
α_1	1.113234316088	1.113234316088	1.113234316088	1.113234038095	1.113235466611	1.113235466611	1.113235466611
α_2	-1.079984496935	-1.079984496935	-1.079984496935	-1.079990259502	-1.079981182643	-1.079981182643	-1.079981182643
α_3	-1.119936882194	-1.119936882194	-1.119936882194	-1.119928189646	-1.119936079567	-1.119936079567	-1.119936079567
α_4	-0.135365940412	-0.135365940412	-0.135365940412	-0.135365853520	-0.135364647762	-0.135364647762	-0.135364647762
α_5	0.106658403956	0.106658403956	0.106658403956	0.106659502922	0.106657783379	0.106657783379	0.106657783379
α_6	-0.195874270445	-0.195874270445	-0.195874270445	-0.195874412452	-0.195873540639	-0.195873540639	-0.195873540639
α_7	0.166662585345	0.166662585345	0.166662585345	0.166664764511	0.166659924256	0.166659924256	0.166659924256
γ_1	0.404709479518	0.404709479518	0.404709479518	0.404709496545	0.404709971357	0.404709971357	0.404709971357
γ_2	-0.033596249061	-0.033596249061	-0.033596249061	-0.033595777560	-0.033596115432	-0.033596115432	-0.033596115432

A.7 Parameter Differences (from Linux_2020b reference)

Table 2: Parameter Differences (from Linux_2020b reference)

Param	Linux_2021b	Linux_2022a	Linux_2024b	Mac_2020b	Mac_2024b	Windows_2024b
δ	0.00e+00	0.00e+00	-4.85e-06	-4.85e-06	-4.85e-06	-4.85e-06
β_n	0.00e+00	0.00e+00	1.14e-05	-1.47e-07	-1.47e-07	-1.47e-07
β_s	0.00e+00	0.00e+00	-1.66e-05	3.10e-06	3.10e-06	3.10e-06
α_1	0.00e+00	0.00e+00	-2.78e-07	1.15e-06	1.15e-06	1.15e-06
α_2	0.00e+00	0.00e+00	-5.76e-06	3.31e-06	3.31e-06	3.31e-06
α_3	0.00e+00	0.00e+00	8.69e-06	8.03e-07	8.03e-07	8.03e-07
α_4	0.00e+00	0.00e+00	8.69e-08	1.29e-06	1.29e-06	1.29e-06
α_5	0.00e+00	0.00e+00	1.10e-06	-6.21e-07	-6.21e-07	-6.21e-07
α_6	0.00e+00	0.00e+00	-1.42e-07	7.30e-07	7.30e-07	7.30e-07
α_7	0.00e+00	0.00e+00	2.18e-06	-2.66e-06	-2.66e-06	-2.66e-06
γ_1	0.00e+00	0.00e+00	1.70e-08	4.92e-07	4.92e-07	4.92e-07
γ_2	0.00e+00	0.00e+00	4.72e-07	1.34e-07	1.34e-07	1.34e-07

A.8 Standard Errors

Table 3: Standard Errors Across All Systems

Param	Linux_2020b	Linux_2021b	Linux_2022a	Linux_2024b	Mac_2020b	Mac_2024b	Windows_2024b
δ	0.04385	0.04385	0.04385	0.05812	0.06398	0.06398	0.06398
β_n	0.06131	0.06131	0.06131	0.01231	0.06796	0.06796	0.06796
β_s	0.07086	0.07086	0.07086	0.05616	0.10117	0.10117	0.10117
α_1	0.00545	0.00545	0.00545	0.00341	0.00335	0.00335	0.00335
α_2	0.00287	0.00287	0.00287	0.00175	0.00384	0.00384	0.00384
α_3	0.00273	0.00273	0.00273	0.00234	0.00230	0.00230	0.00230
α_4	0.00559	0.00559	0.00559	0.00568	0.00583	0.00583	0.00583
α_5	0.00197	0.00197	0.00197	0.00435	0.01152	0.01152	0.01152
α_6	0.00611	0.00611	0.00611	0.00545	0.00837	0.00837	0.00837
α_7	0.00234	0.00234	0.00234	0.00136	0.00837	0.00837	0.00837
γ_1	0.02438	0.02438	0.02438	0.02905	0.05280	0.05280	0.05280
γ_2	0.03632	0.03632	0.03632	0.05403	0.13651	0.13651	0.13651

A.9 Standard Error Ratios (relative to Linux_2020b)

Table 4: Standard Error Ratios (relative to Linux_2020b)

Param	Linux_2021b	Linux_2022a	Linux_2024b	Mac_2020b	Mac_2024b	Windows_2024b
δ	1.000	1.000	1.325	1.459	1.459	1.459
β_n	1.000	1.000	0.201	1.108	1.108	1.108
β_s	1.000	1.000	0.793	1.428	1.428	1.428
α_1	1.000	1.000	0.625	0.614	0.614	0.614
α_2	1.000	1.000	0.610	1.338	1.338	1.338
α_3	1.000	1.000	0.856	0.842	0.842	0.842
α_4	1.000	1.000	1.017	1.043	1.043	1.043
α_5	1.000	1.000	2.202	5.835	5.835	5.835
α_6	1.000	1.000	0.892	1.371	1.371	1.371
α_7	1.000	1.000	0.581	3.583	3.583	3.583
γ_1	1.000	1.000	1.192	2.166	2.166	2.166
γ_2	1.000	1.000	1.487	3.758	3.758	3.758

A.10 Fixed Parameter Validation Results

Table 5: Fixed Parameter Validation Results (Using Exact R2020b Parameter Estimates)

Parameter	All Systems
δ	0.04385
β_n	0.06131
β_s	0.07086
α_1	0.00545
α_2	0.00287
α_3	0.00273
α_4	0.00559
α_5	0.00197
α_6	0.00611
α_7	0.00234
γ_1	0.02438
γ_2	0.03632

Fixed Parameter SE Ratios: All ratios = 1.000 across all systems when using exact R2020b parameter estimates, confirming identical inference capabilities.

A.11 Fixed Parameter Standard Errors (All Systems Using Exact R2020b Parameter Estimates)

Table 6: Standard Errors Computed Using Exact R2020b Parameter Estimates Across All Systems

Parameter	Linux_2020b	Linux_2021b	Linux_2022a	Linux_2024b	Mac_2020b	Mac_2024b	Windows_2024b
δ	0.04385	0.04385	0.04385	0.04385	0.04385	0.04385	0.04385
β_n	0.06131	0.06131	0.06131	0.06131	0.06131	0.06131	0.06131
β_s	0.07086	0.07086	0.07086	0.07086	0.07086	0.07086	0.07086
α_1	0.00545	0.00545	0.00545	0.00545	0.00545	0.00545	0.00545
α_2	0.00287	0.00287	0.00287	0.00287	0.00287	0.00287	0.00287
α_3	0.00273	0.00273	0.00273	0.00273	0.00273	0.00273	0.00273
α_4	0.00559	0.00559	0.00559	0.00559	0.00559	0.00559	0.00559
α_5	0.00197	0.00197	0.00197	0.00197	0.00197	0.00197	0.00197
α_6	0.00611	0.00611	0.00611	0.00611	0.00611	0.00611	0.00611
α_7	0.00234	0.00234	0.00234	0.00234	0.00234	0.00234	0.00234
γ_1	0.02438	0.02438	0.02438	0.02438	0.02438	0.02438	0.02438
γ_2	0.03632	0.03632	0.03632	0.03632	0.03632	0.03632	0.03632

Note: All standard errors are numerically identical across all computational environments when using exact R2020b parameter estimates.

B Replication Instructions for Linux Systems

B.1 Step-by-Step Process for Exact Replication

B.1.1 Prerequisites

- Linux operating system
- MATLAB R2022a or earlier
- Terminal access

B.1.2 Step 1: Check Current MKL Configuration

Start MATLAB and check your current Intel MKL configuration:

```
version -blas
```

Look for the CNR branch information in the output. You should see one of:

- `CNR branch AVX` → Will produce exact replication ✓
- `CNR branch AVX2` → Will produce different results
- `CNR branch SSE4_2` → Will produce different results

B.1.3 Step 2: Set Environment Variables (if needed)

If your system shows CNR branch AVX2 or SSE4_2, exit MATLAB completely and run these commands in your terminal:

```
export MKL_ENABLE_INSTRUCTIONS=AVX
export MKL_DEBUG_CPU_TYPE=5
```

B.1.4 Step 3: Start MATLAB from Terminal

From the same terminal session where you set the environment variables:

```
matlab
```

B.1.5 Step 4: Verify Configuration

In the newly started MATLAB session, verify the configuration changed:

```
version -blas
```

You should now see `CNR branch AVX` in the output.

B.1.6 Step 5: Run Replication Code

Execute the main replication script:

```
master
```

B.2 Expected Results

B.2.1 If CNR Branch AVX is Active:

- Parameter estimates: Identical to submitted results
- Standard errors: Identical to submitted results

B.2.2 If Other CNR Branches are Active:

- Parameter estimates: Nearly identical (differences ~10^{-5})
- Standard errors: Different

B.3 Troubleshooting

B.3.1 Environment Variables Don't Work

If setting environment variables doesn't change the CNR branch:

- Your MATLAB version may be too new (newer versions have deprecated AVX)
- Your system may not support AVX instruction sets
- Try using the provided hard-coded parameter estimates for validation instead

B.4 Technical Notes

- Environment variables must be set BEFORE starting MATLAB
- Changes take effect only for the MATLAB session started from that terminal
- The `master.m` script has been modified to include automatic diagnostic checks

References

BLACKFORD, L. S., A. PETITET, R. POZO, K. REMINGTON, R. C. WHALEY, J. DEMMEL, J. DONGARRA, I. DUFF, S. HAMMARLING, G. HENRY, ET AL. (2002): "An Updated Set of Basic Linear Algebra Subprograms (BLAS)," *ACM Transactions on Mathematical Software*, 28, 135–151. [6]

GOLDBERG, D. (1991): "What Every Computer Scientist Should Know About Floating-Point Arithmetic," *ACM Computing Surveys*, 23, 5–48. [6]

INTEL CORPORATION (2023): *Intel Math Kernel Library Developer Guide: Achieving Reproducible Results*, Intel Corporation, accessed January 2025. [6]