Econometrica Supplementary Material

SUPPLEMENT TO "INFERENCE BASED ON STRUCTURAL VECTOR AUTOREGRESSIONS IDENTIFIED WITH SIGN AND ZERO RESTRICTIONS: THEORY AND APPLICATIONS"

(*Econometrica*, Vol. 86, No. 2, March 2018, 685–720)

JONAS E. ARIAS
Federal Reserve Bank of Philadelphia

JUAN F. RUBIO-RAMÍREZ
Economics Department, Emory University, BBVA Research, and Federal Reserve Bank of Atlanta

DANIEL F. WAGGONER
Federal Reserve Bank of Atlanta

THIS SUPPLEMENT IS ORGANIZED AS FOLLOWS. Section I shows that using one-sided numerical derivatives can decrease computational time without compromising numerical accuracy. Section II tests for the variance of the importance sampler weights to be finite. Section III provides a step-by-step pseudo-code for using Algorithms 1 and 3 in the paper.

I. ONE-SIDED VERSUS TWO-SIDED DERIVATIVES

In Section 7 of the paper, we showed several cases where using one-sided derivatives decreases computing time considerably. We now show that, at least for the application described in Section 6 of the paper, this can be done without compromising the numerical accuracy of either the volume elements or the IRFs.

Consider Figure S.1. This figure shows the histogram of the percent difference between the volume element associated with the mapping $v_{(g\circ f_h)|\mathcal{Z}}(\mathbf{A}_0, \mathbf{A}_+)$, when computed using one-sided derivatives relative to when the same volume element is computed using two-sided derivatives. The percent difference is expressed in terms of the volume elements computed using two-sided derivatives. The figure shows that most of the percent differences are smaller than 0.1. These results suggest that it is unlikely to find cases in which the one- and two-sided derivatives substantially differ.

Next, we show that the percent differences in the volume element reported above do not affect the IRFs. Figure S.2 plots the percentage point difference between the median and the bounds of the probability bands of the IRFs obtained using one-sided derivatives and the ones obtained using two-sided derivatives. In all the cases, the percentage point difference is smaller than 0.1, and the conclusions a researcher would obtain are unchanged.

II. TESTING THE FINITE VARIANCE OF THE IMPORTANCE SAMPLER WEIGHTS

In this section, we numerically test for the variance of the importance sampler weights to be finite. In particular, we use the Wald, score, and likelihood ratio (LR) tests as described in Koopman, Shephard, and Creal (2009). We run these tests on the importance sampler weights obtained using Algorithm 3 and used in Section 6 of our paper.

Jonas E. Arias: jonas.arias@phil.frb.org
Juan F. Rubio-Ramírez: jrubior@emory.edu
Daniel F. Waggoner: daniel.f.waggoner@atl.frb.org

https://doi.org/10.3982/ECTA14468

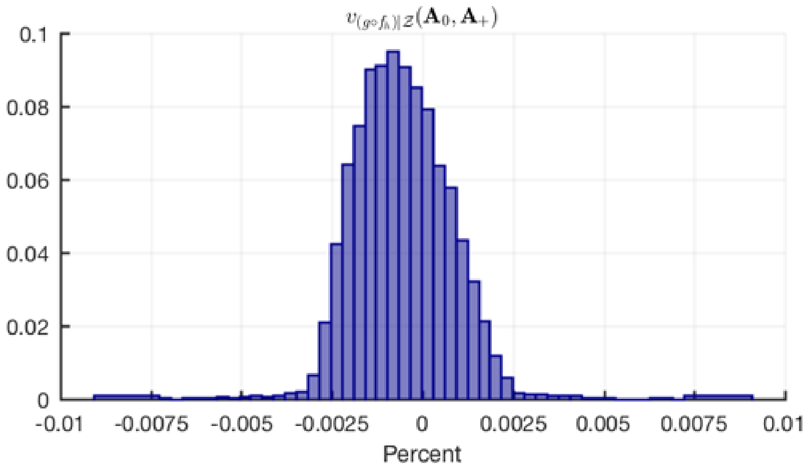

FIGURE S.1.—Volume elements comparison. The histogram shows the percent difference between the volume element $v_{(g\circ f_h)|\mathcal{Z}}(\mathbf{A}_0, \mathbf{A}_+)$ computed using one-sided relative to two-sided derivatives. The histogram includes 99 percent of the support of the distribution and it is based on 10,000 independent draws obtained using Algorithm 3.

The Wald, score, and LR tests assume that the importance sampler weights are independent draws from a Pareto distribution characterized by the shape parameter ξ. The null and alternative hypothesis for the Wald, score, and likelihood ratio tests proposed by

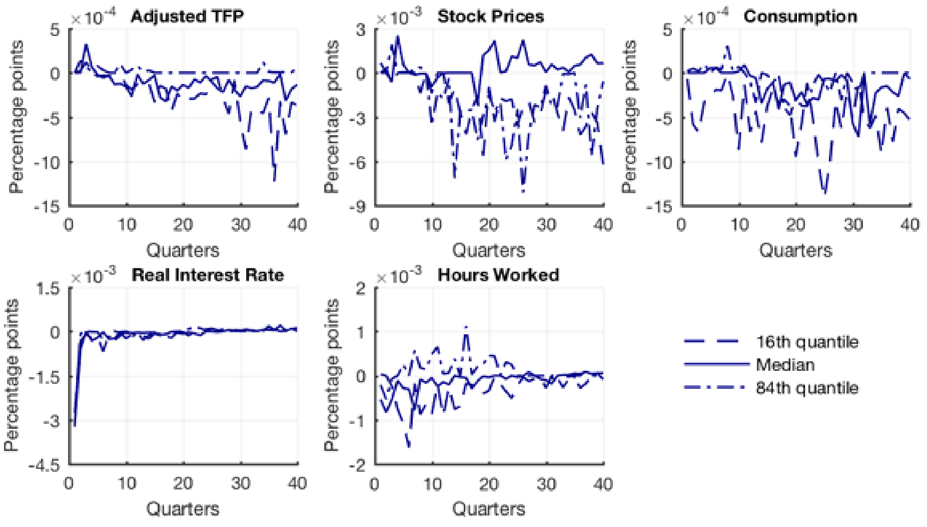

FIGURE S.2.—IRFs comparison. The solid lines show the percentage point difference between the point-wise median IRFs computed using one-sided relative to two-sided derivatives. The dashed lines and the dashed-dotted lines show the percentage point difference in point-wise 16th and 84th quantile IRFs computed using one-sided relative to two-sided derivatives, respectively. The figure is based on 10,000 independent draws obtained using Algorithm 3.

TABLE S.I
WALD, SCORE, AND LIKELIHOOD RATIO TESTS

Threshold	Largest 50%	Largest 40%	Largest 30%	Largest 10%	Largest 1%
Wald	−0.46	−0.56	−0.57	−0.67	−0.58
Score	−26.63	−19.22	−17.34	−9.78	−2.57
LR	0.00	0.00	0.00	0.00	0.00

Koopman, Shephard, and Creal (2009) are

$$H_0 : \xi = \frac{1}{2} \quad \text{and} \quad H_1 : \xi > \frac{1}{2}$$

because for $\xi > \frac{1}{2}$ the Pareto distribution variance does not exist.

We conduct these tests for several thresholds of the importance sampler weights, ranging from the largest 50% to the largest 1% of the importance sampler weights. These thresholds determine the number of importance sampler weights used to implement the tests. Table S.I shows the value of these tests for several thresholds in the aforementioned range—as shown in the first row of the table. The second row of the table shows the Wald test statistics, the third row shows the score test statistics, and the fourth row shows the LR test statistics. The 95% critical values for the Wald, score, and LR tests are 1.65, 1.65, and 2.69, respectively. None of the values displayed in Table S.I exceed these critical values. Hence, these tests indicate that the importance sampler weights have finite variance.

III. NUMERICAL ALGORITHMS

In this section, we provide a step-by-step pseudo-code for using the algorithms developed in the paper. This description is complemented by the MATLAB R2016a code that accompanies this Supplemental Material. In what follows, MATLAB's built-in functions are denoted in *italics* format while the remaining functions are denoted in `typewriter` format.

Algorithm 1 makes independent draws from the normal-generalized-normal posterior distribution over the structural parameterization conditional on the sign restrictions. The pseudo-code given below preserves the notation used in the paper unless stated otherwise.

Algorithm 1

```
1: Set $(\tilde{\nu}, \tilde{\boldsymbol{\Phi}}, \tilde{\boldsymbol{\Psi}}, \tilde{\boldsymbol{\Omega}})$ equal to the posterior parameters as described in Section 2.5 of the paper.
2: Set nd equal to the required number of independent draws from the posterior distribution over the structural parameterization conditional on the sign restrictions.
3: Let cholOmega $=$ h$(\tilde{\boldsymbol{\Omega}})'$;    ▷ The function h denotes MATLAB's chol function. We use h instead of chol because users could define h to be another differentiable decomposition as mentioned in Section 2.3 of the paper. If h $=$ chol, then cholOmega $=$ chol$(\tilde{\boldsymbol{\Omega}})'$.
4: while draw <= 10 × nd do    ▷ 10 × nd is an educated guess of the number of independent draws from the orthogonal reduced-form parameterization necessary to achieve nd independent draws from the posterior distribution over the structural parameterization conditional on the sign restrictions.
```

```
    ▷ Lines 5–7 implement Step 1 of Algorithm 1.
5:      $\boldsymbol{\Sigma} = iwishrnd(\tilde{\boldsymbol{\Phi}}, \tilde{\boldsymbol{\nu}})$;
6:      cholSigma $= \mathtt{h}(\boldsymbol{\Sigma})'$;
7:      $\mathbf{B} = kron(\text{cholSigma}, \text{cholOmega}) \times randn(\text{m} \times \text{n}, 1) + reshape(\tilde{\boldsymbol{\Psi}}, \text{m} \times \text{n}, 1)$; ▷ $\mathbf{B}$
    is a draw from a multivariate normal distribution with mean $\tilde{\boldsymbol{\Psi}}$ and variance $\boldsymbol{\Sigma} \otimes \tilde{\boldsymbol{\Omega}}$.
    ▷ Lines 8–14 implement Step 2 of Algorithm 1.
8:      $\mathbf{X} = randn(\text{n}, \text{n})$;
9:      $[\mathbf{Q}, \mathbf{R}] = qr(\mathbf{X})$;
10:     for $j = 1$ to n do
11:         if $(\mathbf{R}(j, j) < 0)$ then
12:             $\mathbf{Q}(:, j) = -\mathbf{Q}(:, j)$;        ▷ This is the normalization stated in Theorem 4.
13:         end if
14:     end for
    ▷ Lines 15–19 implement Step 3 of Algorithm 1.
15:     $x = [\text{vec}(\mathbf{B}); \text{vec}(\boldsymbol{\Sigma}); \text{vec}(\mathbf{Q})]$;                ▷ vec denotes vectorization.
16:     $y =$ f_h_inv$(x)$;      ▷ The function f_h_inv corresponds to the mapping $f_h^{-1}$
    described in Section 2.3 of the paper; see below.
17:     if Sign restrictions are satisfied then
18:         $(\mathbf{A}_0, \mathbf{A}_+) = (reshape(y(1 : \text{n}^2), \text{n}, \text{n}), reshape(y(\text{n}^2 + 1 : end), \text{m}, \text{n}))$;
19:     end if
    ▷ Line 20 implements Step 4 of Algorithm 1.
20:     draw = draw + 1;
21: end while
```

The function `f_h_inv` is defined as follows:

```
1: function f_h_inv(x)
2:     $\mathbf{B} = reshape(x(1 : \text{m} \times \text{n}), \text{m}, \text{n})$;
3:     $\boldsymbol{\Sigma} = reshape(x(\text{m} \times \text{n} + 1 : \text{m} \times \text{n} + \text{n}^2, \text{n}, \text{n})$;
4:     $\mathbf{Q} = reshape(x(\text{m} \times \text{n} + \text{n}^2 + 1 : end), \text{n}, \text{n})$;
5:     $\mathbf{A}_0 = \mathtt{h}(\boldsymbol{\Sigma})^{-1}\mathbf{Q}$;
6:     $\mathbf{A}_+ = \mathbf{B}\mathbf{A}_0$;
7:     Return $[\text{vec}(\mathbf{A}_0); \text{vec}(\mathbf{A}_+)]$;
8: end function
```

Algorithm 3 makes independent draws from the normal-generalized-normal posterior distribution over the structural parameterization conditional on the sign and zero restrictions. The pseudo-code given below preserves the notation used in the paper and in the step-by-step description of Algorithm 1 unless stated otherwise.

The implementation of Algorithm 3 relies on the auxiliary functions: `ff_h_inv`, `VolumeElement`, `LogVolumeElement`, `ff_h`, `f_h`, and `ZeroRestrictions`. These functions are described after the algorithm in the order in which they are called.

Algorithm 3

```
1: Set $(\tilde{\nu}, \tilde{\boldsymbol{\Phi}}, \tilde{\boldsymbol{\Psi}}, \tilde{\boldsymbol{\Omega}})$ equal to the posterior parameters as described in Section 2.5 of the
   paper.
2: Set nd equal to the required number of independent draws from the posterior distri-
   bution over the structural parameterization conditional on the sign and zero restric-
   tions.
```

```
 3: Let cholOmega $= \mathtt{h}(\tilde{\boldsymbol{\Omega}})'$;
 4: while draw $<= 10 \times$ nd do    ▷ $10 \times$ nd is an educated guess of the number
    of independent draws from the orthogonal reduced-form parameterization necessary
    to achieve nd independent draws from the posterior distribution over the structural
    parameterization conditional on the sign and zero restrictions.
    ▷ Lines 5–14 implement Step 1 of Algorithm 3.
 5:     $\boldsymbol{\Sigma} = iwishrnd(\tilde{\boldsymbol{\Phi}}, \tilde{\boldsymbol{\nu}})$;
 6:     cholSigma $= \mathtt{h}(\boldsymbol{\Sigma})'$;
 7:     $\mathbf{B} = kron(\text{cholSigma}, \text{cholOmega}) \times randn(\mathrm{m} \times \mathrm{n}, 1) + reshape(\tilde{\boldsymbol{\Psi}}, \mathrm{m} \times \mathrm{n}, 1)$;
 8:     for $j = 1$ to n do
 9:         $x_j = randn([\mathrm{n} + 1 - j - z_j, 1])$;
10:         $w_j = x_j / norm(x_j)$;
11:     end for
12:     $w = [w_1; \dots; w_n]$;
13:     $x = [\mathrm{vec}(\mathbf{B}); \mathrm{vec}(\boldsymbol{\Sigma}); w]$;
14:     $y = \mathtt{ff\_h\_inv}(x)$;    ▷ The function ff_h_inv denotes the inverse of the
    composite function $(g \circ f_h)$ described in Section 4.2 of the paper.
    ▷ Lines 15–20 implement Step 2 of Algorithm 3.
15:     if Sign restrictions are satisfied then
16:         $(\mathbf{A}_0, \mathbf{A}_+) = (reshape(y(1 : \mathrm{n}^2), \mathrm{n}, \mathrm{n}), reshape(y(\mathrm{n}^2 + 1 : end), \mathrm{m}, \mathrm{n}))$;
17:         Set uw $= |det(\mathbf{A}_0)|^{-(2\mathrm{n}+\mathrm{m}+1)} / \mathtt{VolumeElement}(\mathbf{A}_0, \mathbf{A}_+)$;
18:     else
19:         Set uw $= 0$;
20:     end if
    ▷ Line 21 implements Step 3 of Algorithm 3.
21:     draw $=$ draw $+ 1$;
22: end while
    ▷ Lines 23–29 implement Step 4 of Algorithm 3.
23: imp_w $=$ uw$/sum$(uw);
24: for draw=1:nd do    ▷ The effective sample size implied by imp_w should be greater
    than or equal to nd.
25:     is_draw $= randsample(1 : size(imp\_w, 1), 1, \text{true}, imp\_w)$;
26:     Set $(\mathbf{B}, \boldsymbol{\Sigma}, \mathbf{Q})$ equal to the corresponding is_draw.
27:     $x = [\mathrm{vec}(\mathbf{B}); \mathrm{vec}(\boldsymbol{\Sigma}); \mathrm{vec}(\mathbf{Q})]$;
28:     $y = \mathtt{f\_h\_inv}(x)$;
29:     $(\mathbf{A}_0, \mathbf{A}_+) = (reshape(y(1 : \mathrm{n}^2), \mathrm{n}, \mathrm{n}), reshape(y(\mathrm{n}^2 + 1 : end), \mathrm{m}, \mathrm{n}))$;
30: end for
```

```
 1: function ff_h_inv(x)
 2:     $\mathbf{B} = reshape(x(1 : \mathrm{m} \times \mathrm{n}), \mathrm{m}, \mathrm{n})$;
 3:     $\boldsymbol{\Sigma} = reshape(x(\mathrm{m} \times \mathrm{n} + 1 : \mathrm{m} \times \mathrm{n} + \mathrm{n}^2, \mathrm{n}, \mathrm{n})$;
 4:     for $j = 1$ to n do
 5:         Set $\mathtt{ZF}_j = \mathbf{Z}_j \mathbf{F}(\mathbf{B}, \boldsymbol{\Sigma}, \mathbf{I}_n)$;
 6:     end for
 7:     $w = x(\mathrm{m} \times \mathrm{n} + \mathrm{n}^2 + 1 : end)$;
 8:     $\mathbf{Q} = zeros(\mathrm{n}, \mathrm{n})$;
 9:     $k = 0$;
10:     for $j = 1$ to n do
```

```
11:         $s = size(W_j, 1)$;  ▷ The matrix $W_j$ is a fixed $(n + 1 - j - z_j) \times n$ random matrix; for $1 \le j \le n$. See Appendix A.3 for details.
12:         $w_j = w(k+1 : k+s)$;
13:         $\tilde{\mathbf{M}}_j = \begin{bmatrix} \mathbf{Q}(:, 1 : j-1)' \\ \mathtt{ZF}_j \\ W_j \end{bmatrix}$;
14:         $[\mathbf{K}, \mathbf{R}] = qr(\tilde{\mathbf{M}}_j')$;
15:         for $i = n - s + 1$ to n do
16:             if $\mathbf{R}(i, i) < 0$ then
17:                 $\mathbf{K}(:, i) = -\mathbf{K}(:, i)$;
18:             end if
19:         end for
20:         $\mathbf{K}_j = \mathbf{K}(:, n - s + 1 : n)$;
21:         $\mathbf{Q}(:, j) = \mathbf{K}_j w_j$;
22:         $k = k + s$;
23:     end for
24:     $x = [\mathrm{vec}(\mathbf{B}); \mathrm{vec}(\mathbf{\Sigma}); \mathrm{vec}(\mathbf{Q})]$;
25:     $y = \mathtt{f\_h\_inv}(x)$;
26: end function
```

```
1: function VolumeElement($\mathbf{A}_0, \mathbf{A}_+$)
2:     Return $exp$(LogVolumeElement ($\mathbf{A}_0, \mathbf{A}_+$))
3: end function
```

```
1: function LogVolumeElement($\mathbf{A}_0, \mathbf{A}_+$)
2:     Dfx = NumericalDerivative(ff_h);    ▷ The function NumericalDerivative is available in the code that accompanies this Supplemental Material.
3:     Dhx = NumericalDerivative(ZeroRestrictions);
4:     N = Dfx $\times$ $null$(Dhx);
5:     Return 0.5 $\times$ LogAbsDet(N$'$ $\times$ N);    ▷ The function LogAbsDet computes the log of the absolute value of the determinant of a square matrix.
6: end function
```

```
1: function ff_h(x)
2:     Use the function f_h to obtain ($\mathbf{B}$, $\mathbf{\Sigma}$, $\mathbf{Q}$) form $x$.
3:     for $j = 1$ to n do
4:         Set $\mathtt{ZF}_j = \mathbf{Z}_j \mathbf{F}(\mathbf{B}, \mathbf{\Sigma}, \mathbf{I}_n)$;
5:     end for
6:     $w = zeros(\sum_{j=1}^{n} n - (j - 1 + z_j), 1)$;
7:     $k = 0$;
8:     for $j = 1 :$ n do
9:         $s = size(W_j, 1)$;
10:        $\tilde{\mathbf{M}}_j = \begin{bmatrix} \mathbf{Q}(:, 1 : j-1)' \\ \mathtt{ZF}_j \\ W_j \end{bmatrix}$;
11:        $[\mathbf{K}, \mathbf{R}] = qr(\tilde{\mathbf{M}}_j')$;
```

```
12:         for $i = \mathrm{n} - s + 1$ to n do
13:             if $\mathbf{R}(i, i) < 0$ then
14:                 $\mathbf{K}(:, i) = -\mathbf{K}(:, i)$;
15:             end if
16:         end for
17:         $\mathbf{K}_j = \mathbf{K}(:, \mathrm{n} - s + 1 : \mathrm{n})$;
18:         $w(k + 1 : k + s) = \mathbf{K}_j'\mathbf{Q}(:, j)$;
19:         $k = k + s$;
20:     end for
21: end function
```

```
1: function f_h(x)
2:     $\mathbf{A}_0 = reshape(x(1 : \mathrm{n}^2), \mathrm{n}, \mathrm{n})$;
3:     $\mathbf{A}_+ = reshape(x(\mathrm{n}^2 + 1 : end), \mathrm{m}, \mathrm{n})$;
4:     $\mathbf{B} = \mathbf{A}_+\mathbf{A}_0^{-1}$;
5:     $\boldsymbol{\Sigma} = (\mathbf{A}_0\mathbf{A}_0')^{-1}$;
6:     $\mathbf{Q} = \mathtt{h}(\frac{1}{2}(\boldsymbol{\Sigma} + \boldsymbol{\Sigma}'))\mathbf{A}_0$;          ▷ See Appendix A.2 for details.
7:     Return [vec($\mathbf{B}$); vec($\boldsymbol{\Sigma}_+$); vec($\mathbf{Q}$)];
8: end function
```

```
1: function ZeroRestrictions($\mathbf{A}_0, \mathbf{A}_+$)
2:     nzeros = 0;
3:     for $j = 1 : \mathrm{n}$ do
4:         nzeros = nzeros + $size(\mathbf{Z}_j, 1)$;
5:     end for
6:     $z = zeros(\mathrm{nzeros}, 1)$;
7:     $k = 1$;
8:     for $j = 1 : \mathrm{n}$ do
9:         $s = size(\mathtt{ZF}_j, 1)$;
10:        $z(k : k + s - 1) = \mathtt{ZF}_j(:, j)$;
11:        $k = k + s$;
12:    end for
13: end function
```

REFERENCE

KOOPMAN, S. J., N. SHEPHARD, AND D. CREAL (2009): "Testing the Assumptions Behind Importance Sampling," *Journal of Econometrics*, 149, 2–11. [1,3]

Co-editor Elie Tamer handled this manuscript.

Manuscript received 9 June, 2016; final version accepted 5 December, 2017; available online 20 December, 2017.