Supplement : Numerically Verifying that y satisfies (GR)

We recognize that the proof of Theorem 5.1 is long. The difficulty is because the theorem is stated for arbitrary parameters $\left(n, m, NI, I, \Sigma_{vv}, \Sigma_{\epsilon\epsilon} \right)$. On the other hand, once numerical values of the parameters are given, it is easy to verify that the random vector y satisfies (GR). The goal of this supplement is to provide a code that does exactly this. We start with some ground work, then we use a random number generator to pick the parameters, then we show that (GR) holds.

(GR) includes 4 identities. The first identity is: $\forall\, k$ **in** $NI, E\left[v \middle| y, s_k \right] = E\left[v \middle| y \right]$. Because of joint normality, this identity is equivalent to

$$\forall\, k \in NI, \mu_v + \begin{bmatrix} cov(v,y) & cov\left(v, s_k\right) \end{bmatrix} \begin{bmatrix} cov(y,y) & cov\left(y, s_k\right) \\ cov\left(s_k, y\right) & cov\left(s_k, s_k\right) \end{bmatrix}^{-1} \begin{bmatrix} y - Ey \\ s_k - Es_k \end{bmatrix} = \mu_v + cov(v,y)cov(y,y)^{-1}(y - Ey)$$

To simplify, we assume $|NI| > 1$, so the random vector $\begin{bmatrix} y \\ s_k \end{bmatrix}$ is non-degenerate and its covariance matrix is invertible. The identity holds for every realization of $\begin{bmatrix} y \\ s_k \end{bmatrix}$ if and only if

(A)

$$\forall\, k \in NI, \begin{bmatrix} cov(v,y) & cov\left(v, s_k\right) \end{bmatrix} \begin{bmatrix} cov(y,y) & cov\left(y, s_k\right) \\ cov\left(s_k, y\right) & cov\left(s_k, s_k\right) \end{bmatrix}^{-1} = \begin{bmatrix} cov(v,y)cov(y,y)^{-1} & 0_{n \times n} \end{bmatrix}$$

The second identity in (GR) is: $\forall\, k$ **in** $NI, cov\left(v \middle| y, s_k \right) = cov\left(v \middle| y \right)$. Because of joint normality, this identity is equivalent to

$$\forall\, k \in NI, cov(v,v) - \begin{bmatrix} cov(v,y) & cov\left(v, s_k\right) \end{bmatrix} \begin{bmatrix} cov(y,y) & cov\left(y, s_k\right) \\ cov\left(s_k, y\right) & cov\left(s_k, s_k\right) \end{bmatrix}^{-1} \begin{bmatrix} cov(y,v) \\ cov\left(s_k, v\right) \end{bmatrix} = cov(v,v) - cov(v,y)cov(y,y)^{-1}cov(y,v)$$

But this second identity is just multiplying equation (A) from the right by the vector $\begin{bmatrix} cov(y,v) \\ cov\left(s_k, v\right) \end{bmatrix}$. Thus, if (A) holds, then the two conditions in (GR) that correspond to the set of non-index investors hold.

Next, we show that the two conditions in (GR) that correspond to the set of index investors are reduced to one cndition.

The third identity in (GR) is $\forall\ k$ **in** $I, E\left[1'v|y, s_k\right] = E\left[1'v|y\right]$. Again, joint normality implies that this condition is equivalent to

$$\forall\ k \in I, 1'\mu_v + 1'\begin{bmatrix} cov(v,y) & cov(v,s_k) \end{bmatrix} \begin{bmatrix} cov(y,y) & cov(y,s_k) \\ cov(s_k,y) & cov(s_k,s_k) \end{bmatrix}^{-1} \begin{bmatrix} y - Ey \\ s_k - Es_k \end{bmatrix} = 1'\mu_v + 1' cov(v,y) cov(y,y)^{-1}(y - Ey)$$

To simplify, we assume $|I| > 1$, so the random vector $\begin{bmatrix} y \\ s_k \end{bmatrix}$ is non-degenerate and its covariance matrix is invertible. The identity holds for every realization of $\begin{bmatrix} y \\ s_k \end{bmatrix}$ if and only if

(B)

$$\forall\ k \in I, 1'\begin{bmatrix} cov(v,y) & cov(v,s_k) \end{bmatrix} \begin{bmatrix} cov(y,y) & cov(y,s_k) \\ cov(s_k,y) & cov(s_k,s_k) \end{bmatrix}^{-1} = \begin{bmatrix} 1' cov(v,y) cov(y,y)^{-1} & 0_{1 \times n} \end{bmatrix}$$

The forth and last identity in (GR) is $\forall\ k \in I, cov\left(1'v|y, s_k\right) = cov\left(1'v|y\right)$. Thanks to joint normality, this identity is equivalent to

$$\forall\ k \in I, 1' cov(v,v)1 - 1'\begin{bmatrix} cov(v,y) & cov(v,s_k) \end{bmatrix} \begin{bmatrix} cov(y,y) & cov(y,s_k) \\ cov(s_k,y) & cov(s_k,s_k) \end{bmatrix}^{-1} \begin{bmatrix} cov(y,v) \\ cov(s_k,v) \end{bmatrix} 1 = 1' cov(v,v)1 - 1' cov(v,y) cov(y,y)^{-1} cov(y,v)1$$

But this second identity is already embeded in equation (B). It is just multiplying (B) from the right by the $(2n+1) \times 1$ vector $\begin{bmatrix} cov(y,v) \\ cov(s_k,v) \end{bmatrix} 1$.

We conclude that to check that y satisfies (GR) (under the simplifying assumtion that $|NI| > 1$ and $|I| > 1$), we only need to check Equations (A) and (B).

We want to avoid inverting large matrices. So we multply (A) from the right by

$$\begin{bmatrix} cov(y,y) & cov(y,s_k) \\ cov(s_k,y) & cov(s_k,s_k) \end{bmatrix}$$ and we simplify to get that (A) is equivalent to

$$\forall\ k \in NI,\quad \begin{bmatrix} cov(v,y) & cov(v,s_k) \end{bmatrix} = \begin{bmatrix} cov(v,y) & cov(v,y)\cdot cov(y,y)^{-1}\cdot cov(y,s_k) \end{bmatrix}$$

matching terms, we concludes that to verify (A) we only need to verify

(AA)

$$cov(v,s_k) = cov(v,y)\cdot cov(y,y)^{-1}\cdot cov(y,s_k)$$

We repeat the same for condition (B). We multiply on the right by $$\begin{bmatrix} cov(y,y) & cov(y,s_k) \\ cov(s_k,y) & cov(s_k,s_k) \end{bmatrix}$$ and we simplify to get that (B) is equivalent to

$$\forall\ k \in I,\quad \begin{bmatrix} 1'\,cov(v,y) & 1'\,cov(v,s_k) \end{bmatrix} = \begin{bmatrix} 1'\,cov(v,y) & 1'\,cov(v,y)\cdot cov(y,y)^{-1}\cdot cov(y,s_k) \end{bmatrix}$$

matching terms, we concludes that to verify (B) we only need to verify

(BB)

$$1'\,cov(v,s_k) = 1'\,cov(v,y)\cdot cov(y,y)^{-1}\cdot cov(y,s_k)$$

Now, before we verify (AA) and (BB), we need to express each term that shows up in (AA)and (BB) in terms of the exogenous parametrs of the model, wich are $cov(v,v)$ and $cov(\epsilon,\epsilon)$. The term

$cov(v,s_k) = cov(v,v)$, for both $k \in NI$ and $k \in I$.

$cov(v,y)$ is given in Appendix B of the paper (Equation B.2)

$cov(y,y)$ is given in Appendix B of the paper (Equation B.7)

$cov(y,s_k)$ is given in Appendix B of the paper: Equation B.4 when $k \in NI$ and Equation B.5 when $k \in I$.

Exogenous Parameters

Steps	Results (Maple Code)
Restart Maple, and set the random seed. (without an argument to randomize(), the seed is taken from the clock. Without a call to randomize(), the seed is the default Maple seed).	*restart* *randomize*(1234) 1234 **(1.1)**
Number of assets: Number of Investors: Number of non-indexers (between 2 and m-2):	$n := RandomTools[Generate](integer(range = 2..100))$ $n := 49$ **(1.2)** $m := RandomTools[Generate](integer(range = 4..10000))$ $m := 8919$ **(1.3)** $NI := RandomTools[Generate](integer(range = 2..m - 2))$ $NI := 1320$ **(1.4)**
Load Maple's bulit in LinearAlegbra package.	*with*(*LinearAlgebra*) :
A routine to generate a random positive definite matrix Maple's Random Matrix populates entries with integers between -99 and 99. To use random numbers that are Gaussian, comment Line 5 and uncomment Line 6.	1 `generatRandomPositiveDefiniteMatrix:= proc(nn` 2 `  local L, Sigma, found;` 3 `  found:= false;` 4 `  while not found do` 5 `    L:= RandomMatrix(nn, nn,shape=triangular` 6 `    #L:= Matrix(n,RandomTools:-Generate('dis` 7 `    Sigma:= L.Transpose(L); #If L is non-sin` 8 `    found:= IsDefinite(Sigma, query = 'posit` 9 `  end do;` 10 `  return Sigma;` 11 `end proc:`

Test the routine	$generatRandomPositiveDefiniteMatrix(4)$ $\begin{bmatrix} 2025 & -675 & -4140 & 1215 \\ -675 & 10026 & -1491 & -6147 \\ -4140 & -1491 & 14930 & -8152 \\ 1215 & -6147 & -8152 & 21093 \end{bmatrix}$	**(1.5)**
Set the exogenous covariance matrices	$\Sigma_{vv} := generatRandomPositiveDefiniteMatrix(n)$ $\Sigma_{vv} := \begin{bmatrix} 49\ x\ 49\ Matrix \\ Data\ Type:\ anything \\ Storage:\ rectangular \\ Order:\ Fortran\_order \end{bmatrix}$	**(1.6)**
	$\Sigma_{\epsilon\epsilon} := generatRandomPositiveDefiniteMatrix(n)$ $\Sigma_{\epsilon\epsilon} := \begin{bmatrix} 49\ x\ 49\ Matrix \\ Data\ Type:\ anything \\ Storage:\ rectangular \\ Order:\ Fortran\_order \end{bmatrix}$	**(1.7)**
	$one := Matrix(n, 1, 1);$ $one := \begin{bmatrix} 49\ x\ 1\ Matrix \\ Data\ Type:\ anything \\ Storage:\ rectangular \\ Order:\ Fortran\_order \end{bmatrix}$	**(1.8)**
Regardless what the index k is, $\Sigma_{vs} = \Sigma_{vv}$, and $\Sigma_{ss} = \Sigma_{vv} + m \cdot \Sigma_{\epsilon\epsilon}$	$\Sigma_{vs} := \Sigma_{vv}$ $\Sigma_{vs} := \begin{bmatrix} 49\ x\ 49\ Matrix \\ Data\ Type:\ anything \\ Storage:\ rectangular \\ Order:\ Fortran\_order \end{bmatrix}$	**(1.9)**
	$\Sigma_{ss} := \Sigma_{vv} + m \cdot \Sigma_{\epsilon\epsilon}$ $\Sigma_{ss} := \begin{bmatrix} 49\ x\ 49\ Matrix \\ Data\ Type:\ anything \\ Storage:\ rectangular \\ Order:\ Fortran\_order \end{bmatrix}$	**(1.10)**

Endogenous Parameters

Steps	Results (Maple Code)
g is Equation (12) in the paper	$g := \left(\Sigma_{vv} + \Sigma_{\epsilon\epsilon}\right)^{-1}.\Sigma_{vv}.one$ $g := \begin{bmatrix} 49 \text{ x } 1 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$ **(2.1)**
Σ_{vy} is Equation (B.2) in the paper. We use Maple's ability to bulid a matrix from submatrices	$\Sigma_{vy} := Matrix\left(\left[\Sigma_{vv}, \Sigma_{vv}.g\right]\right)$ $\Sigma_{vy} := \begin{bmatrix} 49 \text{ x } 50 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$ **(2.2)**
Σ_{yy} is given in Equation (B.7) in the paper.	$\Sigma_{yy} := Matrix\left(\left[\left[\Sigma_{vv} + \frac{m}{NI}\cdot\Sigma_{\epsilon\epsilon}, \Sigma_{vv}.one\right], \left[Transpose(one).\Sigma_{vv}, Transpose(one).\Sigma_{vv}.g\right]\right]\right)$ $\Sigma_{yy} := \begin{bmatrix} 50 \text{ x } 50 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$ **(2.3)**

Verifying Equation (AA)

Steps	Results (Maple code)
When $k \in NI$, Σ_{ys} is given in Equation (B.4) in the Appendix B in the paper.	$\Sigma_{ys} := Matrix\left(\left[\left[\Sigma_{vv} + \frac{m}{NI}\cdot\Sigma_{\epsilon\epsilon}\right], \left[Transpose(one).\Sigma_{vv}\right]\right]\right)$ $\Sigma_{ys} := \begin{bmatrix} 50 \text{ x } 49 \text{ Matrix} \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix}$ **(3.1)**
We check identity (AA) holds. Maple compares two matrices using the Equal command and returns true of the two matrices are equal. This verifies (AA).	$Equal\left(\Sigma_{vs}, \Sigma_{vy}.\Sigma_{yy}^{-1}.\Sigma_{ys}\right)$ $true$ **(3.2)**

Verifying Equation (BB)

Steps	Results
When $k \in I$, Σ_{ys} is given in Equation (B.5) in the Appendix B in the paper.	$\Sigma_{ys} := Matrix\left(\left[\left[\Sigma_{vv}\right], \left[Transpose(one).\Sigma_{vv}\right]\right]\right)$ $\Sigma_{ys} := \begin{bmatrix} 50 \text{ x } 49 \text{ Matrix} \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix}$ **(4.1)**
We check the identity (BB). Maple compares two matrices using the Equal command and returns true of the two matrices are equal. This verifies (BB).	$Equal\left(Transpose(one).\Sigma_{vs}, Transpose(one).\Sigma_{vy}.\Sigma_{yy}^{-1}.\Sigma_{ys}\right)$ $true$ **(4.2)**