Supplemental Appendices for

Information matrix tests for multinomial logit models

Dante Amengual

CEMFI, Casado del Alisal 5, E-28014 Madrid, Spain

<amengual@cemfi.es>

Gabriele Fiorentini

Università di Firenze and RCEA, Viale Morgagni 59, I-50134 Firenze, Italy

<gabriele.fiorentini@unifi.it>

Enrique Sentana

CEMFI, Casado del Alisal 5, E-28014 Madrid, Spain

<sentana@cemfi.es>

April 2024, revised December 2024

A. Proofs

A.1. Proof or Proposition 1

Note that

$$\frac{\partial p_j(\mathbf{z};\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j} = \frac{1}{\left(\sum_{\ell=1}^{K} e^{\boldsymbol{\beta}_\ell'\mathbf{z}}\right)^2}\left[e^{\boldsymbol{\beta}_j'\mathbf{z}}\left(\sum_{\ell=1}^{K} e^{\boldsymbol{\beta}_\ell'\mathbf{z}}\right) - e^{2\boldsymbol{\beta}_j'\mathbf{z}}\right]\mathbf{z} = p_j(\mathbf{z};\boldsymbol{\beta})[1-p_j(\mathbf{z};\boldsymbol{\beta})]\mathbf{z},$$

while

$$\frac{\partial p_k(\mathbf{z};\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j} = \frac{-e^{\boldsymbol{\beta}_k'\mathbf{z}}e^{\boldsymbol{\beta}_j'\mathbf{z}}}{\left(\sum_{s=1}^{K} e^{\boldsymbol{\beta}_s'\mathbf{z}}\right)^2}\mathbf{z} = -p_k(\mathbf{z};\boldsymbol{\beta})p_j(\mathbf{z};\boldsymbol{\beta})\mathbf{z}$$

when $k \neq j$. Interestingly, these expressions coincide with $\mathbf{z}$ times the conditional variance of ξ_j given $\mathbf{z}$ and the conditional covariance between ξ_j and ξ_k given $\mathbf{z}$, respectively.

To derive the score, it is convenient to re-write both expressions together as

$$\frac{\partial p_k(\mathbf{z};\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j} = p_k(\mathbf{z};\boldsymbol{\beta})[I(j=k) - p_j(\mathbf{z};\boldsymbol{\beta})]\mathbf{z},$$

where $I(.)$ is the usual indicator function. The contribution of a single observation to the log-likelihood function (ignoring constant terms) will be

$$\ln f(\boldsymbol{\xi},\mathbf{z};\boldsymbol{\beta}) = \sum_{k=1}^{K} \xi_k \ln p_k(\mathbf{z};\boldsymbol{\beta}).$$

Hence, the score with respect to $\boldsymbol{\beta}_j$ $(j = 2,\ldots,K)$ will be given by

$$\mathbf{s}_j(\boldsymbol{\xi},\mathbf{z};\boldsymbol{\beta}) = \frac{\partial \ln f(\boldsymbol{\xi},\mathbf{z};\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j} = \sum_{k=1}^{K} \xi_k[I(j=k) - p_j(\mathbf{z};\boldsymbol{\beta})]\mathbf{z} = [\xi_j - p_j(\mathbf{z};\boldsymbol{\beta})]\mathbf{z} = u_j(\xi_j,\mathbf{z};\boldsymbol{\beta})\mathbf{z},$$

where $u_j(\xi_j,\mathbf{z};\boldsymbol{\beta}) = \xi_j - p_j(\mathbf{z};\boldsymbol{\beta})$. Thus, we can write the first-order conditions together as (3). From here, the second derivatives will be

$$\begin{aligned}
\mathbf{h}_{jj}(\boldsymbol{\xi},\mathbf{z};\boldsymbol{\beta}) &= \frac{\partial^2 \ln f(\boldsymbol{\xi},\mathbf{z};\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j \partial \boldsymbol{\beta}_j'} = -p_j(\mathbf{z};\boldsymbol{\beta})[1-p_j(\mathbf{z};\boldsymbol{\beta})]\mathbf{z}\mathbf{z}' \quad \text{and} \\
\mathbf{h}_{j\ell}(\boldsymbol{\xi},\mathbf{z};\boldsymbol{\beta}) &= \frac{\partial^2 \ln f(\boldsymbol{\xi},\mathbf{z};\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j \partial \boldsymbol{\beta}_\ell'} = p_j(\mathbf{z};\boldsymbol{\beta})p_\ell(\mathbf{z};\boldsymbol{\beta})\mathbf{z}\mathbf{z}',
\end{aligned}$$

whence (4) follows. Therefore, we will have that

$$\begin{aligned}
\frac{\partial \ln f(\boldsymbol{\xi},\mathbf{z};\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j}\frac{\partial \ln f(\boldsymbol{\xi},\mathbf{z};\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j'} + \frac{\partial \ln f(\boldsymbol{\xi},\mathbf{z};\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j \partial \boldsymbol{\beta}_j'} &= \{u_j^2(\xi_j,\mathbf{z};\boldsymbol{\beta}) - p_j(\mathbf{z};\boldsymbol{\beta})[1-p_j(\mathbf{z};\boldsymbol{\beta})]\}\mathbf{z}\mathbf{z}' \text{ while} \\
\frac{\partial \ln f(\boldsymbol{\xi},\mathbf{z};\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j}\frac{\partial \ln f_i(\boldsymbol{\xi},\mathbf{z};\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_\ell'} + \frac{\partial \ln f(\boldsymbol{\xi},\mathbf{z};\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j \partial \boldsymbol{\beta}_\ell'} &= [u_j(\xi_j,\mathbf{z};\boldsymbol{\beta})u_\ell(\xi_\ell,\mathbf{z};\boldsymbol{\beta}) + p_j(\mathbf{z};\boldsymbol{\beta})p_\ell(\mathbf{z};\boldsymbol{\beta})]\mathbf{z}\mathbf{z}',
\end{aligned}$$

where we have used the fact that $\xi_j^2 = \xi_j$ and $\xi_j\xi_\ell = 0$. Therefore, we can write the influence functions corresponding to the information matrix equality in matrix notation as (5), the advantage of using of *vech* instead of *vec* being that we easily eliminate the duplicated influence functions that appear in (4) and the outer product of (3), thereby avoiding generalised inverses and providing the right number of degrees of freedom.

The reminder statements in the second part of the proposition follow directly from Chesher (1983) and Lancaster (1984) given the *i.i.d.* nature of the sample. □

A.2. Proof of Proposition 2

We can expand the quantities that appear in $cov(\mathbf{m}_{j\ell}, \mathbf{m}_{j'\ell})$ as follows:

$$
\begin{aligned}
E(m_{jj}^2|\mathbf{z}) &= E\{[u_j^2 - p_{ji}(1-p_{ji})]^2|\mathbf{z}\} \;=\; E(u_j^4|\mathbf{z}) - 2p_j(1-p_j)E(u_j^2|\mathbf{z}) + p_j^2(1-p_j)^2, \\
E(m_{j\ell}^2|\mathbf{z}) &= E[(u_j u_\ell + p_j p_\ell)^2|\mathbf{z}] \;=\; E(u_j^2 u_\ell^2|\mathbf{z}) - 2p_j p_\ell E(u_j u_\ell|\mathbf{z}) + p_j^2 p_\ell^2, \\
E(m_{jj} m_{j'j'}|\mathbf{z}) &= E\{[u_j^2 - p_j(1-p_j)][u_{j'}^2 - p_{j'}(1-p_{j'})]|\mathbf{z}\} \\
&= E(u_j^2 u_{j'}^2|\mathbf{z}) - p_j(1-p_j)E(u_{j'}^2|\mathbf{z}) - p_{j'}(1-p_{j'})E(u_j^2|\mathbf{z}) + p_j(1-p_j)p_{j'}(1-p_{j'}), \\
E(m_{jj} m_{j\ell}|\mathbf{z}) &= E\{[(u_j^2 - p_j(1-p_j)](u_j u_\ell + p_j p_\ell)|\mathbf{z}\} \\
&= E(u_j^3 u_\ell|\mathbf{z}) + p_j p_\ell E(u_j^2|\mathbf{z}) - p_j(1-p_j)E(u_j u_\ell|\mathbf{z}) - p_j^2(1-p_j)p_\ell, \\
E(m_{jj} m_{j'\ell}|\mathbf{w}) &= E\{[u_j^2 - p_{ji}(1-p_{ji})](u_{j'} u_\ell + p_{j'} p_\ell)|\mathbf{z}\} \\
&= E(u_j^2 u_{j'} u_\ell|\mathbf{z}) + p_{j'} p_\ell E(u_j^2|\mathbf{z}) - p_j(1-p_j)E(u_{j'} u_\ell|\mathbf{z})] - p_j(1-p_j)p_{j'} p_\ell, \\
E(m_{j\ell} m_{j'\ell}|\mathbf{z}) &= E(u_j u_\ell + p_j p_\ell)(u_{j'} u_\ell + p_{j'} p_\ell)|\mathbf{z}] \\
&= E(u_\ell^2 u_j u_{j'}|\mathbf{z}) + p_j p_\ell p_{j'} p_\ell + p_{j'} p_\ell E(u_j u_\ell|\mathbf{z})] + p_j p_\ell E(u_{j'} u_\ell|\mathbf{z}) \quad \text{and} \\
E(m_{j\ell} m_{j'\ell'}|\mathbf{z}) &= E[(u_j u_\ell + p_j p_\ell)(u_{j'} u_{\ell'} + p_{j'} p_{\ell'})|\mathbf{z}] \\
&= E(u_j u_\ell u_{j'} u_{\ell'}|\mathbf{z}) + p_j p_\ell p_{j'} p_{\ell'} + p_j p_\ell E(u_{j'} u_{\ell'}|\mathbf{z}) + p_{j'} p_{\ell'} E(u_j u_\ell|\mathbf{z}).
\end{aligned}
$$

Then, if we use the formulae for the fourth-order centered moments of the multinomial distribution in Ouimet (2021), namely

$$
\begin{aligned}
E(u_j^4) &= (1-p_j)p_j[1 - 3(1-p_j)p_j], \\
E(u_j^3 u_\ell) &= -p_j[1 - 3(1-p_j)p_j]p_\ell, \\
E(u_j^2 u_\ell^2) &= p_j p_\ell(p_j + p_\ell - 3p_j p_\ell), \\
E(u_j^2 u_{j'} u_\ell) &= p_j(1-p_j)p_{j'} p_\ell \quad \text{and} \\
E(u_j u_\ell u_{j'} u_{\ell'}) &= -3p_j p_\ell p_{j'} p_{\ell'},
\end{aligned}
$$

we obtain the expressions in part a) of the lemma.

Doing the same with the expressions in $cov(\mathbf{m}_{j\ell}, \mathbf{s}_{j'})$:

$$\begin{aligned} E(m_{jj}u_j|\mathbf{z}) &= E\{[u_j^2 - p_j(1-p_j)]u_j|\mathbf{w}\} \;=\; E(u_j^3|\mathbf{z}) - p_j(1-p_j)E(u_j|\mathbf{z}), \\ E(m_{jj}u_{j'}|\mathbf{z}) &= E\{[u_j^2 - p_j(1-p_j)]u_{j'}|\mathbf{z}\} \;=\; E(u_j^2 u_{j'}|\mathbf{z}) - p_j(1-p_j)E(u_{j'}|\mathbf{z}), \\ E(m_{j\ell}u_j|\mathbf{z}) &= E[(u_j u_\ell + p_j p_\ell)u_j|\mathbf{z}] \;=\; E(u_j^2 u_\ell|\mathbf{z}) + p_j p_\ell E(u_j|\mathbf{z}) \quad \text{and} \\ E(m_{j\ell}u_{j'}|\mathbf{z}) &= cov(m_{j\ell}, u_{j'}|\mathbf{z}) \;=\; E[(u_j u_\ell + p_j p_\ell)u_{j'}|\mathbf{z}] \;=\; E(u_j u_\ell u_{j'}|\mathbf{z}) + p_j p_\ell E(u_{j'}|\mathbf{z}), \end{aligned}$$

and using the formulae for the third-order centered moments of the multinomial distribution in Ouimet (2021),

$$\begin{aligned} E(u_j^3) &= p_j(1-p_j)(1-2p_j), \\ E(u_j^2 u_\ell) &= p_j(1-2p_j)p_\ell \\ E(u_j u_\ell u_{j'}) &= 2p_j p_\ell p_{j'}, \end{aligned}$$

we obtain the expressions in part b) of the lemma.

Finally, the expression for the information matrix follows from its definition. □

B. The IM test for the conditional multinomial logit model

In the conditional multinomial logit model we can write the choice probabilities, which are a function of some m regressors $\mathbf{w}'_{ki} = (w_{1ki}, \dots, w_{mki})$ that vary across n individuals indexed by i and K alternatives indexed by k, as

$$p_{ki}(\boldsymbol{\alpha}, \mathbf{w}_{1i}, \dots, \mathbf{w}_{Ki}) = P(\xi_{ki} = 1|\mathbf{w}_{1i}, \dots, \mathbf{w}_{Ki}) = \frac{e^{\boldsymbol{\alpha}'\mathbf{w}_{ki}}}{\sum_{\ell=1}^{K} e^{\boldsymbol{\alpha}'\mathbf{w}_{\ell i}}}, k = 1, \dots, K, \qquad \text{(B1)}$$

where $\boldsymbol{\alpha}$ is an $m \times 1$ vector of coefficients that does not depend on the alternatives. For our purposes, it is convenient to re-write these probabilities as

$$p_{ki}(\boldsymbol{\alpha}, \mathbf{w}_i) = \frac{e^{\boldsymbol{\alpha}'(\mathbf{w}_{ki} - \mathbf{w}_{1i})}}{\sum_{\ell=1}^{K} e^{\boldsymbol{\alpha}'(\mathbf{w}_{\ell i} - \mathbf{w}_{1i})}}, \quad k = 1, \dots, K$$

by dividing both the numerator and denominator of (B1) by $e^{\boldsymbol{\alpha}'\mathbf{w}_{1i}}$ so that

$$p_{1i}(\boldsymbol{\alpha}, \mathbf{w}_i) = \frac{1}{1 + \sum_{\ell=2}^{K} e^{\boldsymbol{\alpha}'(\mathbf{w}_{\ell i} - \mathbf{w}_{1i})}}$$

As explained in footnote 4 of chapter 2 of Maddala (1983), it is possible to write this conditional multinomial logit model as a restricted multinomial logit model. The following simple example with three choices and two regressors illustrates the required transformation:

$$
\begin{aligned}
p_{1i}(\boldsymbol{\alpha}, \mathbf{w}_i) &= \frac{e^{\alpha_1 w_{11i} + \alpha_2 w_{21i}}}{\sum_{\ell=1}^{3} e^{\alpha_1 w_{1\ell i} + \alpha_2 w_{2\ell i}}} = \frac{1}{1 + \sum_{\ell=2}^{3} e^{\alpha_1 (w_{1\ell i} - w_{11i}) + \alpha_2 (w_{2\ell i} - w_{21i})}} \\
p_{2i}(\boldsymbol{\alpha}, \mathbf{w}_i) &= \frac{e^{\alpha_1 w_{12i} + \alpha_2 w_{22i}}}{\sum_{\ell=1}^{3} e^{\alpha_1 w_{1\ell i} + \alpha_2 w_{2\ell i}}} = \frac{e^{\alpha_1 (w_{12i} - w_{11i}) + \alpha_2 (w_{22i} - w_{21i})}}{1 + \sum_{\ell=2}^{3} e^{\alpha_1 (w_{1\ell i} - w_{11i}) + \alpha_2 (w_{2\ell i} - w_{21i})}} \\
p_{3i}(\boldsymbol{\alpha}, \mathbf{w}_i) &= \frac{e^{\alpha_1 w_{13i} + \alpha_2 w_{23i}}}{\sum_{\ell=1}^{3} e^{\alpha_1 w_{1\ell i} + \alpha_2 w_{2\ell i}}} = \frac{e^{\alpha_1 (w_{13i} - w_{11i}) + \alpha_2 (w_{23i} - w_{21i})}}{1 + \sum_{\ell=2}^{3} e^{\alpha_1 (w_{1\ell i} - w_{11i}) + \alpha_2 (w_{2\ell i} - w_{21i})}}
\end{aligned}
$$

As expected, the three probabilities add up to 1.

More generally, define $\mathbf{w}_i' = [(\mathbf{w}_{2i} - \mathbf{w}_{1i})', \ldots, (\mathbf{w}_{ki} - \mathbf{w}_{1i})', \ldots, (\mathbf{w}_{Ki} - \mathbf{w}_{1i})']$ and $\boldsymbol{\beta}_k'(\boldsymbol{\alpha}) = (\mathbf{0}', \ldots, \boldsymbol{\alpha}', \ldots, \mathbf{0}')$ for $k = 2, \ldots, K$, which in this trinomial example are simply

$$
\begin{aligned}
\mathbf{w}_i' &= (w_{12i} - w_{11i}, w_{22i} - w_{21i}; w_{13i} - w_{11i}, w_{23i} - w_{21i}) \\
\boldsymbol{\beta}_2'(\boldsymbol{\alpha}) &= (\alpha_1, \alpha_2; 0, 0) \\
\boldsymbol{\beta}_3'(\boldsymbol{\alpha}) &= (0, 0; \alpha_1, \alpha_2)
\end{aligned}
$$

As we shall see below, the crucial ingredient to obtain the expressions for the IM test for the conditional multinomial logit model from the IM test for the multinomial logit model in Propositions 1 and 2 are the Jacobians of $\boldsymbol{\beta}_k'(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$, which in our trinomial example are given by

$$
\begin{aligned}
\frac{\partial \boldsymbol{\beta}_2'(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \\
\frac{\partial \boldsymbol{\beta}_3'(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} &= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.
\end{aligned}
$$

Let $u_{ki} = \xi_{ki} - p_{ki}$ denote the generalised residuals for $k = 2, \ldots, K$. We know that the contribution of observation i to score in the trinomial logit model is given by

$$
\begin{aligned}
\frac{\partial l_i}{\partial \boldsymbol{\beta}_2'(\boldsymbol{\alpha})} &= u_2 \mathbf{w}_i' = u_2 (w_{12i} - w_{11i}, w_{22i} - w_{21i}; w_{13i} - w_{11i}, w_{23i} - w_{21i}) \\
\frac{\partial l_i}{\partial \boldsymbol{\beta}_3'(\boldsymbol{\alpha})} &= u_3 \mathbf{w}_i' = u_3 (w_{12i} - w_{11i}, w_{22i} - w_{21i}; w_{13i} - w_{11i}, w_{23i} - w_{21i})
\end{aligned}
$$

As a result, the chain rule for first derivatives immediately implies that the contribution of observation i to the score of the conditional multinomial logit model will be given by

$$
\begin{aligned}
\frac{\partial l_i}{\partial \boldsymbol{\alpha}'} &= \frac{\partial l_i}{\partial \boldsymbol{\beta}_2'(\boldsymbol{\alpha})} \frac{\partial \boldsymbol{\beta}_2'(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} + \frac{\partial l_i}{\partial \boldsymbol{\beta}_3'(\boldsymbol{\alpha})} \frac{\partial \boldsymbol{\beta}_3'(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = u_2 \mathbf{w}_i' \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} + u_3 \mathbf{w}_i' \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \\
&= u_2 (w_{12i} - w_{11i}, w_{22i} - w_{21i}) + u_3 (w_{13i} - w_{11i}, w_{2\ell i} - w_{21i}).
\end{aligned}
$$

If we realise that $\xi_{1i} = 1 - \xi_{2i} - \xi_{3i}$ and $p_{1i} = 1 - p_{2i} - p_{3i}$, then we can define $u_{1i} = \xi_{1i} - p_{1i} = -(u_{2i} + u_{3i})$ so that we can write

$$\frac{\partial l_i}{\partial \boldsymbol{\alpha}'} = u_1(w_{11i}, w_{21i}) + u_2(w_{12i}, w_{22i}) + u_3(w_{13i}, w_{23i}),$$

which coincides with the expression we would obtain by working directly with (B1). Entirely analogous expressions apply for general K. In this respect, it is important to mention that Cameron and Trivedi (2005) write the score as follows:

$$\frac{\partial l_i}{\partial \boldsymbol{\alpha}'} = \sum_{k=1}^{K} \xi_{ki}(\mathbf{w}'_{ki} - \bar{\mathbf{w}}'_i) \tag{B2}$$

where

$$\bar{\mathbf{w}}'_i = \sum_{\ell=1}^{K} p_{\ell i}\mathbf{w}'_{\ell i}.$$

But we can always write (B2) as

$$\begin{aligned}\frac{\partial l_i}{\partial \boldsymbol{\alpha}'} &= \sum_{k=1}^{K} \xi_{ki}\left(\mathbf{w}'_{ki} - \sum_{\ell=1}^{K} p_{\ell i}\mathbf{w}'_{\ell i}\right) = \sum_{k=1}^{K} \xi_{ki}\mathbf{w}'_{ki} - \sum_{\ell=1}^{K} p_{\ell i}\mathbf{w}'_{\ell i}\sum_{k=1}^{K}\xi_{ki} \\ &= \sum_{k=1}^{K} \xi_{ki}\mathbf{w}'_{ki} - \sum_{\ell=1}^{K} p_{\ell i}\mathbf{w}'_{\ell i} = \sum_{k=1}^{K} u_{ki}\mathbf{w}'_{ki}\end{aligned}$$

because $\sum_{k=1}^{K} \xi_{ki} = 1$ and the addition operation satisfies the commutative property.

We can also use the chain rule for second derivatives in theorem 6.9 of Magnus and Neudecker (2019) to obtain the Hessian of the log-likelihood function. In principle, there would be two terms. The first one simply premultiplies the Hessian of the multinomial logit model with respect to $\boldsymbol{\beta}_2(\boldsymbol{\alpha})$ and $\boldsymbol{\beta}_3(\boldsymbol{\alpha})$ by the Jacobian matrix

$$\left[\frac{\partial \boldsymbol{\beta}'_2(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}, \frac{\partial \boldsymbol{\beta}'_3(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}\right] \tag{B3}$$

and postmultiplies it by its transpose. The second term, on the other hand, requires the product of the Kronecker product of the score times the identity matrix with the second-order Jacobian

$$\frac{\partial}{\partial \boldsymbol{\alpha}'} vec\left[\frac{\partial \boldsymbol{\beta}'_2(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}, \frac{\partial \boldsymbol{\beta}'_3(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}\right].$$

But since the mappings from $\boldsymbol{\alpha}$ to $\boldsymbol{\beta}_2(\boldsymbol{\alpha})$ and $\boldsymbol{\beta}_3(\boldsymbol{\alpha})$ are both linear, this second Jacobian is identically 0, so we only need the first term. As a consequence, the sum of the Hessian and the outer product of the score of the conditional multinomial model can be obtained by simply premultiplying the sum of the Hessian and outer product of the score of the multinomial logit model by the Jacobian (B3) and postmultiplying it by its transpose.

Again, the same argument applies for general K. Consequently, we can combine the Jacobian matrix with the theoretical expressions for the conditional variances and covariances of the different influence functions in our Proposition 2 to obtain the expressions for the asymptotic covariance matrices for the conditional multinomial model.

C. Monte Carlo simulations: design and additional results

C.1. Design

For each DGP, we always include an intercept and either one or two standard normal uncorrelated explanatory variables. Following Horowitz (1994), we keep the explanatory variables $\mathbf{z}_i$, $i = 1, \ldots, N$ fixed in repeated samples. Nevertheless, we minimise the effects of the specific draws of these regressors by using the standard normal quantile function to generate them inverting a grid of points equally spaced over the unit interval - from $1/(2N)$ to $1 - 1/(2N)$. In the case of two non-constant regressors, we randomly permute each of them separately to ensure their independence, and additionally conduct a Cholesky decomposition to make them exactly orthogonal in the sample.

More importantly, we choose the $\boldsymbol{\beta}'s$ so that in simulated samples of five million observations they provide roughly balanced frequencies across categories and reasonable values for the pseudo-R^2's proposed by Cragg and Uhler (1970) and McFadden (1974), which we denote as R^2_{CU} and R^2_{MF}, respectively. Specifically, we consider under the null:

DGP A $K=3$, $L=2$: We pick $\boldsymbol{\beta}_2 = (-1, -2)'$ and $\boldsymbol{\beta}_3 = (-1, 2)'$ so that the average frequencies are 0.36, 0.32 and 0.32, with $R^2_{MF} = 0.34$ and $R^2_{CU} = 0.14$. As the coefficient sign does not alter the explanatory power of the z's, the two binary logits have $R^2_{MF} = 0.26$ and $R^2_{CU} = 0.15$.

DGP B $K = 3$, $L = 3$: We pick $\boldsymbol{\beta}_2 = (-1, -2, 2)'$ and $\boldsymbol{\beta}_3 = (-1, 2, -1)'$ so that the average frequencies are 0.28, 0.36 and 0.36, with $R^2_{MF} = 0.45$ and $R^2_{CU} = 0.21$, and $R^2_{MF} = 0.35$ and $R^2_{CU} = 0.21$ for the binary logits.

DGP C $K = 5$, $L = 2$: We pick $\boldsymbol{\beta}_2 = (-1, -2)'$, $\boldsymbol{\beta}_3 = (-1, 2)'$, $\boldsymbol{\beta}_4 = (-2, -4)'$ and $\boldsymbol{\beta}_5 = (-2, 4)'$ so that the average frequencies are 0.24, 0.14, 0.14, 0.24 and 0.24, with $R^2_{MF} = 0.37$ and $R^2_{CU} = 0.10$. Once again, the sign of the coefficient does not alter the explanatory power of the z's, so that the two binary logits involving (ξ_1, ξ_2) and (ξ_1, ξ_3) are such that $R^2_{MF} = 0.15$ and $R^2_{CU} = 0.08$, while those for (ξ_1, ξ_4) and (ξ_1, ξ_5) have $R^2_{MF} = 0.51$ and $R^2_{CU} = 0.34$.

DGP D $K = 5$, $L = 3$: We pick $\boldsymbol{\beta}_2 = (-1, -2, 2)'$, $\boldsymbol{\beta}_3 = (-1, 2, -2)'$, $\boldsymbol{\beta}_4 = (-2, -4, 4)'$ and $\boldsymbol{\beta}_5 = (-2, 4, -4)'$ so that the average frequencies are 0.18, 0.11, 0.11, 0.30 and 0.30, with $R^2_{MF}=0.47$ and $R^2_{CU}=0.16$. In turn, the two binary logits for (ξ_1,ξ_2) and (ξ_1,ξ_3) have $R^2_{MF}=0.18$ and $R^2_{CU}=0.10$, while those for (ξ_1,ξ_4) and (ξ_1,ξ_5) have $R^2_{MF}=0.59$ and $R^2_{CU}=0.43$.

As for the alternatives, we consider:

DGP *a* For the second half of the sample, we replace the slopes of z_1 by -6 and 4 when $K=3$, and -4, 6, 4 and 0 when $K=5$.

DGP *b* We perturb the $K-1$ slopes of z_1 by 3ϵ, with ϵ obtained by the standard normal quantile function to a grid of points equally spaced ranging from $1/(2N)$ to $1-1/(2N)$.

DGP *c* We draw samples from the following nested logit models:

For $K=3$, we set
$\Pr(\xi_1=1|\mathbf{z})=e^{\boldsymbol{\beta}_2'\mathbf{z}}/(1+e^{\boldsymbol{\beta}_2'\mathbf{z}})$,
$\Pr(\xi_2=1|\mathbf{z})=\Pr(\xi_1=0|\mathbf{z})/(1+e^{\boldsymbol{\gamma}_2'\mathbf{z}})$,
$\Pr(\xi_3=1|\mathbf{z})=\Pr(\xi_1=0|\mathbf{z})e^{\boldsymbol{\gamma}_2'\mathbf{z}}/(1+e^{\boldsymbol{\gamma}_2'\mathbf{z}})$,
where, if $L=2$, $\boldsymbol{\beta}_2=(1,-2)'$, $\boldsymbol{\gamma}_2=(-3,5)$ and, if $L=3$, $\boldsymbol{\beta}_2=(1.5,2,-2)'$, $\boldsymbol{\gamma}_2=(6,5,-5)$.

For $K=5$, we set
$\Pr(\xi_1=1|\mathbf{z})=1/(1+e^{\boldsymbol{\beta}_2'\mathbf{z}}+e^{\boldsymbol{\beta}_3'\mathbf{z}})$,
$\Pr(\xi_2=1|\mathbf{z})=e^{\boldsymbol{\beta}_2'\mathbf{z}}/\left[(1+e^{\boldsymbol{\beta}_2'\mathbf{z}}+e^{\boldsymbol{\beta}_3'\mathbf{z}})(1+e^{\boldsymbol{\gamma}_2'\mathbf{z}})\right]$,
$\Pr(\xi_3=1|\mathbf{z})=e^{\boldsymbol{\beta}_2'\mathbf{z}}e^{\boldsymbol{\gamma}_2'\mathbf{z}}/\left[(1+e^{\boldsymbol{\beta}_2'\mathbf{z}}+e^{\boldsymbol{\beta}_3'\mathbf{z}})(1+e^{\boldsymbol{\gamma}_2'\mathbf{z}})\right]$,
$\Pr(\xi_4=1|\mathbf{z})=e^{\boldsymbol{\beta}_3'\mathbf{z}}/\left[(1+e^{\boldsymbol{\beta}_2'\mathbf{z}}+e^{\boldsymbol{\beta}_3'\mathbf{z}})(1+e^{\boldsymbol{\delta}_2'\mathbf{z}})\right]$,
$\Pr(\xi_5=1|\mathbf{z})=e^{\boldsymbol{\beta}_3'\mathbf{z}}e^{\boldsymbol{\delta}_2'\mathbf{z}}/\left[(1+e^{\boldsymbol{\beta}_2'\mathbf{z}}+e^{\boldsymbol{\beta}_3'\mathbf{z}})(1+e^{\boldsymbol{\delta}_2'\mathbf{z}})\right]$,
where, if $L=2$, $\boldsymbol{\beta}_2=(-2,-2)'$, $\boldsymbol{\beta}_3=(-2,2)'$, $\boldsymbol{\gamma}_2=(-8,8)$, $\boldsymbol{\delta}_2=(8,-8)$ and, if $L=3$, $\boldsymbol{\beta}_2=(-2,-2,-1)'$, $\boldsymbol{\beta}_3=(-2,-2,-1)'$, $\boldsymbol{\gamma}_2=(-8,8,1)$, $\boldsymbol{\delta}_2=(8,-8,1)$.

C.2. Additional results for the binary logit model

In Table A1 below we report the same figures as in Table 1 but for the binary logits for models with three categories. The results for models with five categories are available upon request. Not surprisingly, the same pattern is obtained regarding the massive overrejection of the OPS version of the test when relying on asymptotic critical values. Interestingly, the overrejection of the CM test at the 1% level becomes more moderate, likely due to small number of degrees of freedom of the corresponding asymptotic distribution, namely $L(L+1)/2$. Once again, the parametric bootstrap corrects the size distortions for all the sample sizes we consider. Similarly, in Table A2 below we report the rejection rates under the different alternatives that we consider for the binary logit models when there are three categories. As expected, the power figures indicate the same pattern as in Table 2, but with less power.

C.3. Additional references

Cameron, A.C. and Trivedi, P.K. (2005): *Microeconometrics: methods and applications*, Cambridge University Press

Cragg, S. G. and Uhler, R. (1970): "The demand for automobiles", *Canadian Journal of Economics*, 3, 386–406.

Maddala, G.S. (1983): *Limited-dependent and qualitative variables in econometrics*, Cambridge University Press.

Magnus, J.R. and Neudecker, H. (2019): *Matrix differential calculus with applications in statistics and econometrics*, 3rd edition, Wiley.

Ouimet, F. (2021): "General formulas for the central and non-central moments of the multinomial distribution", *Stats* 4, 18–27.

Table A1: (Binary) logit IM tests: Size properties (for three categories)

Panel A: Two explanatory variables: $\mathbf{z} = (1, z)'$ with $z \sim i.i.d.\ N(0, 1)$

	Asymptotic critical values							Bootstraped critical values					
	OPS			CM				OPS			IM		
Sample size	10%	5%	1%	10%	5%	1%	Sample size	10%	5%	1%	10%	5%	1%
	(ξ_1, ξ_2)							(ξ_1, ξ_2)					
125	63.18	55.31	38.40	7.02	4.42	2.33	125	4.81	1.61	0.09	9.46	4.67	1.12
500	41.00	34.26	24.39	8.49	5.21	2.47	500	9.63	4.55	0.76	9.66	5.02	1.12
2,000	26.20	19.85	12.38	9.57	5.75	2.14	2,000	10.37	5.25	1.06	10.05	5.28	1.19
	(ξ_1, ξ_3)							(ξ_1, ξ_3)					
125	62.37	54.16	37.90	7.24	4.70	2.51	125	4.92	1.40	0.11	9.55	4.86	1.21
500	41.51	34.73	24.97	8.47	5.13	2.36	500	9.57	4.80	0.67	9.70	4.78	0.96
2,000	24.73	18.63	11.33	9.26	5.51	1.85	2,000	9.54	4.63	1.01	9.86	4.95	0.90

Panel B: Three explanatory variables: $\mathbf{z} = (1, z_1, z_2)'$ with $(z_1, z_2) \sim i.i.d.\ N(\mathbf{0}, \mathbf{I}_2)$

	Asymptotic critical values							Bootstraped critical values					
	OPS			CM				OPS			IM		
Sample size	10%	5%	1%	10%	5%	1%	Sample size	10%	5%	1%	10%	5%	1%
	(ξ_1, ξ_2)							(ξ_1, ξ_2)					
125	77.80	70.61	53.37	7.51	4.68	2.25	125	3.37	1.00	0.11	9.44	4.62	0.96
500	58.54	51.25	39.26	9.45	5.64	2.47	500	9.63	4.37	0.63	10.11	5.18	1.04
2,000	32.10	26.48	16.46	9.44	5.48	1.88	2,000	10.01	5.31	1.01	9.29	4.67	0.99
	(ξ_1, ξ_3)							(ξ_1, ξ_3)					
125	86.94	80.87	63.79	8.47	5.54	2.58	125	4.16	1.23	0.07	10.37	5.31	1.11
500	56.83	49.86	37.42	8.84	5.34	2.05	500	9.86	4.93	0.79	9.56	4.75	0.97
2,000	32.76	25.89	16.76	10.80	6.08	1.86	2,000	10.61	5.34	0.88	10.64	5.31	0.98

Notes: Monte Carlo rejection rates based on 10,000 replications. OPS refers to the version of the statistic proposed by Chesher (1983) and Lancaster (1984), while CM to the feasible version that makes use of the theoretical expressions in Proposition 2 replacing the true parameter values by their MLEs and unconditional expectations by sample averages. Rejection rates in the right subpanels are based on the asymptotic distribution in Proposition 1 while the left ones on a parametric bootstrap procedure in which we simulate $B = 99$ samples from the model estimated under the null. See Supplemental Appendix C.1 for details about the DGPs.

Table A2: (Binary) logit IM tests: Power properties (for three categories)

Panel A: Alternative hypothesis: Heterogeneous Gaussian β_{i2}

	Two regressors							Three regressors					
	(ξ_1, ξ_2)			(ξ_1, ξ_3)				(ξ_1, ξ_2)			(ξ_1, ξ_3)		
Sample size	10%	5%	1%	10%	5%	1%	Sample size	10%	5%	1%	10%	5%	1%
125	62.36	45.04	13.20	7.28	4.04	0.76	125	49.20	30.12	5.60	8.72	4.52	0.88
500	98.40	95.60	78.64	9.88	5.16	0.84	500	63.48	46.20	7.28	10.60	5.92	1.28
2,000	100.00	100.00	100.00	10.24	5.76	1.36	2,000	100.00	100.00	99.32	10.44	4.72	0.88

Panel B: Alternative hypothesis: Group heterogeneity in β_{i2}

	Two regressors							Three regressors					
	(ξ_1, ξ_2)			(ξ_1, ξ_3)				(ξ_1, ξ_2)			(ξ_1, ξ_3)		
Sample size	10%	5%	1%	10%	5%	1%	Sample size	10%	5%	1%	10%	5%	1%
125	21.48	12.80	3.16	26.64	18.60	5.20	125	17.04	8.48	1.44	35.56	21.04	3.88
500	28.28	18.20	4.48	45.36	29.76	5.96	500	29.04	15.44	2.52	52.72	38.20	11.00
2,000	34.84	24.36	6.68	73.60	61.28	25.36	2,000	40.32	28.16	7.76	94.72	89.24	54.60

Panel C: Alternative hypothesis: Nested logit

	Two regressors							Three regressors					
	(ξ_1, ξ_2)			(ξ_1, ξ_3)				(ξ_1, ξ_2)			(ξ_1, ξ_3)		
Sample size	10%	5%	1%	10%	5%	1%	Sample size	10%	5%	1%	10%	5%	1%
125	36.64	18.92	2.32	24.60	15.24	3.68	125	22.11	14.72	7.00	27.12	14.40	1.84
500	95.60	86.92	32.80	61.32	44.60	11.48	500	40.64	28.08	8.88	82.52	69.76	29.36
2,000	100.00	100.00	99.64	99.32	97.44	67.80	2,000	90.16	80.12	39.08	100.00	100.00	98.60

Notes: Monte Carlo rejection rates based on 2,500 replications. Results for the feasible version of the IM test that makes use of the theoretical expressions in Proposition 2 replacing the true parameter values by their MLEs and unconditional expectations by sample averages. Rejection rates are based on a parametric bootstrap procedure in which we simulate $B = 99$ samples from the model estimated under the null. See Supplemental Appendix C.1 for details about the DGPs.

Matlab Code

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% IMtest main function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [statS,statT] = IMtest(beta,Y,W)

%% Input
% beta     L x (NM-1) matrix of MNlogit parameters (excluding the normalizing zeros)
% Y        Nx1 vector of categorical observations Y(i)=k  k=1,...,K
% W        NxL matrix of regressors (including the constant)

%% Output
% statS    IM statistics based on sample covariance of influence functions
% statT    IM statistics based on theoretical covariance of influence functions

warning off;
[NW,NMm] = size(beta); NM = NMm + 1;
if NM == 2
    Y = Y + 1;
end
N = length(Y);
b = [zeros(NW,1),beta]';
PPki = exp(W*b')./sum(exp(W*b'),2);
YY = zeros(length(Y),NM);
for ii = 1:NM
    YY(:,ii) = (Y == ii);
end
% Computing regressors and their sample covariance matrix
XX = zeros(N,NW*(NW+1));
XX(:,1:NW) = W;
for i=1:NW
        XX(:,i*NW+1:(i+1)*NW) = W(:,i)*ones(1,NW).*W(:,1:end);
end
aux = zeros(NW,NW);
% Eliminating redundant columns and rows of kron(Wt,Wt')
temp = 1;
for i= 1:NW
    for j= 1:NW
        aux(j,i) = temp;
        temp = temp + 1;
    end
end
aux2 = vech(aux)' + NW;
WWvech = XX(:,aux2);    % WWvech = [XX(:,1:NW),XX(:,aux2)];
WWall = XX(:,NW+1:end);
%% Contribution to the score, Hessian and IM infl. functions (i = 1,...,N)
ScoreJ = zeros(NM*NW,1);
HessJ = zeros(NM*NW,NM*NW);
IMmom = zeros(NM*NW,NM*NW);
for ii = 1:NM
    % Score
    ScoreJ(1+(ii-1)*NW:ii*NW) = mean((YY(:,ii) - PPki(:,ii).*ones(1,NW)).*W);
    % Hessian
```

```
    tempHess = mean(-(PPki(:,ii).*(1 - PPki(:,ii))*ones(1,NW^2)).*WWall);
    HessJ(1+(ii-1)*NW:ii*NW,1+(ii-1)*NW:ii*NW) = reshape(tempHess,NW,NW);
    % IM influence functions
    tempIM = mean(((YY(:,ii) - PPki(:,ii)).^2 - (PPki(:,ii).*...
               (1 - PPki(:,ii)))).*ones(1,NW^2)).*WWall);
    IMmom(1+(ii-1)*NW:ii*NW,1+(ii-1)*NW:ii*NW) = reshape(tempIM,NW,NW);
    for jj = ii+1:NM
        % Hessian
        tempHess = mean((PPki(:,ii).*PPki(:,jj))*ones(1,NW^2).*WWall);
        HessJ(1+(ii-1)*NW:ii*NW,1+(jj-1)*NW:jj*NW) = reshape(tempHess,NW,NW);
        HessJ(1+(jj-1)*NW:jj*NW,1+(ii-1)*NW:ii*NW) = reshape(tempHess,NW,NW);
        % IM influence functions
        tempIM = mean((((YY(:,ii) - PPki(:,ii)).*(YY(:,jj) - PPki(:,jj))...
               + PPki(:,ii).*PPki(:,jj)).*ones(1,NW^2)).*WWall);
        IMmom(1+(ii-1)*NW:ii*NW,1+(jj-1)*NW:jj*NW) = reshape(tempIM,NW,NW);
        IMmom(1+(jj-1)*NW:jj*NW,1+(ii-1)*NW:ii*NW) = reshape(tempIM,NW,NW);
    end
end
%% Sample covariances for everything
ScoreJN = zeros(N,NM*NW);
HessJN = zeros(N,NM*NW,NM*NW);
for ii = 1:NM
    % Score
    ScoreJN(:,1+(ii-1)*NW:ii*NW) = (YY(:,ii) - PPki(:,ii).*ones(1,NW)).*W;
    % Hessian
    tempHess = -(PPki(:,ii).*(1 - PPki(:,ii))*ones(1,NW^2)).*WWall;
    HessJN(:,1+(ii-1)*NW:ii*NW,1+(ii-1)*NW:ii*NW) = reshape(tempHess,N,NW,NW);
    for jj = ii+1:NM
        % Hessian
        tempHess = (PPki(:,ii).*PPki(:,jj))*ones(1,NW^2).*WWall;
        HessJN(:,1+(ii-1)*NW:ii*NW,1+(jj-1)*NW:jj*NW) = reshape(tempHess,N,NW,NW);
        HessJN(:,1+(jj-1)*NW:jj*NW,1+(ii-1)*NW:ii*NW) = reshape(tempHess,N,NW,NW);
    end
end
% Computing the information matrix
ScoreJN = ScoreJN(:,1+NW:NM*NW);   % InfMat1 = ScoreJN'*ScoreJN/N;
HessJN = HessJN(:,1+NW:NM*NW,1+NW:NM*NW);
InfMat = - reshape(mean(reshape(HessJN,N,(NM-1)^2*NW^2)),(NM-1)*NW,(NM-1)*NW);
IMmomN = zeros(N,(NM-1)*NM/2*NW*(NW+1)/2);
index = 0;
for ii = 2:NM
    % IM influence functions
    tempIM = ((YY(:,ii) - PPki(:,ii)).^2 - (PPki(:,ii).*...
               (1 - PPki(:,ii)))).*ones(1,NW*(NW+1)/2)).*WWvech;
    IMmomN(:,index+1:index+NW*(NW+1)/2) = tempIM;
    index = index + NW*(NW+1)/2;
    for jj = ii+1:NM
        tempIM = (((YY(:,ii) - PPki(:,ii)).*(YY(:,jj) - PPki(:,jj))...
               + PPki(:,ii).*PPki(:,jj)).*ones(1,NW*(NW+1)/2)).*WWvech;
        IMmomN(:,index+1:index+NW*(NW+1)/2) = tempIM;
        index = index + NW*(NW+1)/2;
    end
end
```

```
% Computing the test statistic
IMmomMean = mean(IMmomN);
VarScoreS = ScoreJN'*ScoreJN/N;
VarIMmomS = IMmomN'*IMmomN/N;
CovScoreIMmomS = IMmomN'*ScoreJN/N;
%% Using LIE for Covariance matrices
CovIMmomScoreT = zeros((NM-1)*NM/2*NW*(NW+1)/2,NW*(NM-1));
index = 0;
for ii = 2:NM
    for jj = ii:NM
        index = index + NW*(NW+1)/2;
        for kk = 2:NM % index for score
            if (ii == jj) && (ii == kk)
                p1 = PPki(:,ii);
                temp = p1.*(1 - 3*p1 + 2*p1.^2);
            elseif (ii == jj) && (kk ~= ii) && (kk ~= jj)
                p1 = PPki(:,ii);
                p2 = PPki(:,kk);
                temp = p1.*(-1 + 2*p1).*p2;
            elseif (ii == kk) && (jj ~= ii) && (jj ~= kk)
                p1 = PPki(:,ii);
                p2 = PPki(:,jj);
                temp = p1.*(-1 + 2*p1).*p2;
            elseif (jj == kk) && (ii ~= jj) && (ii ~= kk)
                p1 = PPki(:,jj);
                p2 = PPki(:,ii);
                temp = p1.*(-1 + 2*p1).*p2;
            else
                p1 = PPki(:,ii);
                p2 = PPki(:,jj);
                p3 = PPki(:,kk);
                temp = 2*p1.*p2.*p3;
            end
            for hh = 1:NW
                CovIMmomScoreT(index+1-NW*(NW+1)/2:index,...
                                (kk-2)*NW+hh) = mean(WWvech'.*...
                                (ones(NW*(NW+1)/2,1)).*(temp'.*W(:,hh)'),2);
            end
        end
    end
end
CovIMmomIMmomT = zeros((NM-1)*NM/2*NW*(NW+1)/2,(NM-1)*NM/2*NW*(NW+1)/2);
indexrow = 0;
for ii = 2:NM
    for jj = ii:NM
        indexrow = indexrow + NW*(NW+1)/2;
        indexcol = 0;
        for iip = 2:NM
            for jjp = iip:NM
                indexcol = indexcol + NW*(NW+1)/2;

                % Fourth order, only one
                if (ii == jj) && (ii == iip) && (ii == jjp)
                    p1 = PPki(:,ii);
                    temp = -4*p1.^4 + 8*p1.^3 - 5*p1.^2 + p1;
                % (Two) second order (variance)
                elseif ((ii == iip) && (jj == jjp)) && (ii ~= jj) ||...
```

```
                   ((ii == jjp) && (iip == jj)) && (ii ~= iip)
                p1 = PPki(:,ii);
                p2 = PPki(:,jj);
                temp= p1.^2.*p2 + p1.*p2.^2 - 4*p1.^2.*p2.^2;
            % (Two) second order (covariance)
            elseif ((ii == jj) && (iip == jjp)) && (ii ~= iip)
                p1 = PPki(:,ii);
                p2 = PPki(:,iip);
                temp= -p1.*p2 + 2*p1.^2.*p2 + 2*p1.*p2.^2 - 4*p1.^2.*p2.^2;
            % Third order one, linear another one (j,j) & (j,j')
            elseif (ii == jj) && (ii == iip) && (jjp ~= ii)
                p1 = PPki(:,ii);
                p2 = PPki(:,jjp);
                temp = p2.*(4*p1.^2 - 4*p1.^3 - p1);
            elseif (ii == jj) && (ii == jjp) && (iip ~= ii)
                p1 = PPki(:,ii);
                p2 = PPki(:,iip);
                temp = p2.*(4*p1.^2 - 4*p1.^3 - p1);
            elseif (iip == jjp) && (ii == iip) && (jj ~= ii)
                p1 = PPki(:,iip);
                p2 = PPki(:,jj);
                temp = p2.*(4*p1.^2 - 4*p1.^3 - p1);
            elseif (iip == jjp) && (jj == iip) && (jj ~= ii)
                p1 = PPki(:,iip);
                p2 = PPki(:,ii);
                temp = p2.*(4*p1.^2 - 4*p1.^3 - p1);
            % Second order one, linear another one (j,j) & (j',l)
            elseif (ii == jj) && (ii ~= iip) && (jj ~= jjp)
                p1 = PPki(:,ii);
                p2 = PPki(:,iip);
                p3 = PPki(:,jjp);
                temp = 2*p1.*p2.*p3 - 4*p1.^2.*p2.*p3;
            elseif (iip == jjp) && (iip ~= ii) && (jjp ~= jj)
                p1 = PPki(:,iip);
                p2 = PPki(:,ii);
                p3 = PPki(:,jj);
                temp = 2*p1.*p2.*p3 - 4*p1.^2.*p2.*p3;
            % Second order one, linear another one (j,l) & (j,l')
            elseif (ii == iip) && (jj ~= ii) && (jjp ~= iip)
                p1 = PPki(:,ii);
                p2 = PPki(:,jj);
                p3 = PPki(:,jjp);
                temp = p1.*p2.*p3 - 4*p1.^2.*p2.*p3;
            elseif (ii == jjp) && (jj ~= ii) && (iip ~= jjp)
                p1 = PPki(:,ii);
                p2 = PPki(:,jj);
                p3 = PPki(:,iip);
                temp = p1.*p2.*p3 - 4*p1.^2.*p2.*p3;
            elseif (jj == jjp) && (ii ~= jj) && (iip ~= jjp)
                p1 = PPki(:,jj);
                p2 = PPki(:,ii);
                p3 = PPki(:,iip);
                temp = p1.*p2.*p3 - 4*p1.^2.*p2.*p3;
            elseif (jj == iip) && (ii ~= jj) && (iip ~= jjp)
                p1 = PPki(:,jj);
                p2 = PPki(:,ii);
```

```
                    p3 = PPki(:,jjp);
                    temp = p1.*p2.*p3 - 4*p1.^2.*p2.*p3;

                % All linear (j,l) & (j',l')
                else
                    p1 = PPki(:,ii);
                    p2 = PPki(:,jj);
                    p3 = PPki(:,iip);
                    p4 = PPki(:,jjp);
                    temp = -4*p1.*p2.*p3.*p4;
                end
                for hh = 1:NW*(NW+1)/2
                    CovIMmomIMmomT(indexrow+1-NW*(NW+1)/2:indexrow,...
                                indexcol+1-NW*(NW+1)/2+hh-1) = mean(WWvech
                                    '.*...
                                (ones(NW*(NW+1)/2,1)).*(temp'.*WWvech(:,hh)')
                                    ,2);
                end
            end
        end
    end
end
VarIMmomSthetaS = VarIMmomS - CovScoreIMmomS*(VarScoreS\CovScoreIMmomS');
statS = N*IMmomMean*(VarIMmomSthetaS\IMmomMean');
VarIMmomSthetaT = CovIMmomIMmomT - CovIMmomScoreT*(InfMat\CovIMmomScoreT');
statT = N*IMmomMean*(VarIMmomSthetaT\IMmomMean');
end
```