# Testing Macro Models Using Indirect Inference

David Meenagh & Yongdeng Xu[1]

# 1. Introduction

Indirect Inference provides a classical statistical inferential framework for testing a model. The aim is to compare the performance of the auxiliary model estimated on the simulated data derived from the model, with the performance of the auxiliary model when estimated from the actual data. In practice we use a VAR as the auxiliary model, but you could also use IRFs and moments. If the structural model is correct then its predictions about the time series properties of the data should match those based on actual data. We choose a VAR as the auxiliary model because the solution to a log-linearised DSGE model can be represented as a restricted VARMA model, and this can be closely represented by a VAR. A level VAR can be used if the shocks are stationary. In what follows we do not assume that the data is stationary; however if you wish only to use stationary data then you may ignore the remarks below about trends and tests of non stationarity of the error processes.

# 2. Model Evaluation by Indirect Inference

The method of evaluating a model by Indirect Inference is carefully explained for users in Le et al. (2016) which should be cited when using any of these programmes. The method was introduced and refined in a series of papers referred to there. The criterion we use when evaluating the model is the Wald test of the differences between the vector of relevant VAR coefficients from simulated and actual data. If the DSGE model is correct then it should produce simulated data that is similar to the actual data, and therefore the VAR estimates on the simulated data will not be significantly different from the VAR estimates on the actual data. From the actual data we get the VAR parameters $\beta^a$, and from the simulations we get N sets of VAR parameters $\beta^i$ ( for i = 1::N), from which we perform the relevant calculations. The Wald statistic that we calculate is:

$$W = (\beta^a - \bar{\beta})'\Omega^{-1}(\beta^a - \bar{\beta}) \tag{1}$$

where $\bar{\beta} = E(\beta^i) = \frac{1}{N}\sum_{i=1}^{N}\beta^i$ and $\Omega = \mathrm{cov}(\beta^i - \bar{\beta}) = \frac{1}{N}\sum_{i=1}^{N}(\beta^i - \bar{\beta})(\beta^i - \bar{\beta})'$ .

In essence we are measuring the distance the actual VAR parameters are from the average of the simulated VAR parameters.

## 2.1 *Implementation of the Wald test by bootstrapping*

Suppose the DSGE model is

$$A_0 E_t y_{t+1} = A_1 y_t + z_t \tag{2}$$

$$z_t = D z_{t-1} + E\varepsilon_t$$

The DSGE model is solved by Dynare (Juillard, 2001). The solved reduced form is

[1] Corresponding Author: xuy16@cf.ac.uk, Cardiff Business School, Cardiff University, Aberconway Building, Colum Drive, Cardiff, CF10 3EU, UK

$$x_t = Ax_{t-1} + B\varepsilon_t \quad (3)$$

where $x_t = (y_t, z_t, a_t)'$, $a_t$ are the auxiliary variables. The coefficients $A$ and $B$ are derived from (2)[2].

The following steps summarise how to implement the Wald test by bootstrapping:

**Step 1:** Calculate the residuals and innovations of the economic model conditional on the data and parameters.

**Step 2:** Derive the simulated data by bootstrapping

**Step 3:** Compute the Wald statistic.

**Step 1:** Calculating the model residuals ( $z_t$ ) and innovations ( $\varepsilon_t$ ).

The number of independent structural residuals is taken to be less than or equal to the number of endogenous variables. Using the data and the parameters we can calculate the structural residuals. If the equation does not have any expectations then the residuals are simply backed out from the equation and the data. If the equation has expectations in it we need to estimate the expected values. To do this we use the robust instrumental variables methods of McCallum (1976) and Wickens (1982), with the lagged endogenous data as instruments. In practice we estimate a VAR of all the expected variables and use this to calculate the expectations. In some DSGE models many of the structural residuals are assumed to be generated by autoregressive processes. If they are, then we need to estimate them. After re-estimation of AR coefficients, we can calculate model innovations. We call this method 'LIML'. This procedure is implemented by the `Get_Res_LIML()` function.

```
[residual,inno,rho_est] = GetRes_LIML(fname,act_data,inx_expect,inx_eqs);
```

Or if we obtained the AR coefficients from calibration or estimation (as e.g. in SW(2007) model), we can get the model innovations directly from the solved reduced form. We call this method the 'exact method'. This procedure is implemented by `Get_Res_Exact()` function.

```
[residual,inno,rho_est] = GetRes_Exact(fname,act_data,[]);
```

The details of two methods are explained in the next section.

**Step 2:** Simulating the data. Once we have the model innovations, we can simulate the data by bootstrapping these innovations. We bootstrap by time vector to preserve any simultaneity between them, and solve the resulting model using Dynare. More specifically, the bootstrapped data $x_t$ is obtained from equation (3). To obtain the N bootstrapped simulations that we need we repeat this process, drawing each sample independently. This procedure is implemented the `Boots_data()` function.

Another type of bootstrap is the parametric bootstrap. That is to say if we know the error distribution (i.e. normal distribution with known variance), we can also bootstrap the data from Monte Carlo simulation. There is an option `type` that you can choose to use parametric or residual bootstrap.

```
type=2;                 %type=1, boostrap from unknown shocks
                        %type=2, boostrap from known shocks (MC simulation)
```

[2] In dynare, A=oo_.dr.ghx; B=oo_.dr.ghu;

```
boots_data = Boots_data(fname,act_data,inno,nboot,A,B,stv,type);
```

**Step 3:** Compute the Wald statistic. We estimate the auxiliary model — a VAR(1) — using both the actual data and the N samples of simulated data. We then calculate the Wald statistic using equation (1). The bootstrap distribution of the Wald statistic can be found by substituting each $\beta^i$ for $\beta^a$ in Equation (1).

The choice of variables and the order of the VAR is up to you. The Wald test is a strict test, so increasing the order of the VAR makes the test more stringent; hence in practice we use a VAR(1). You can use all the variables in the VAR, or a subset of variables to see what combinations of parameters the model can fit.

For the model to fit the data at the 95% confidence level we want the Wald statistic for the actual data to be less than the 95th percentile of the Wald statistics from the simulated data. The Wald statistics from the simulated data come from a $\chi^2$ distribution with degrees of freedom equal to k-1, where k is the number of parameters in $\beta$ .

To make it easier to understand whether the model has not been rejected by the data we transform the Wald for the actual data into a t-statistic using the formula and scale it so that if the Wald was equal to the 95th percentile from the simulated data we would get a Transformed Wald of 1.645.

$$T = 1.648\left(\frac{\sqrt{2w^a} - \sqrt{2k-1}}{\sqrt{2w^{0.95}} - \sqrt{2k-1}}\right) \tag{4}$$

where $w^a$ is the Wald statistic on the actual data and $w^{0.95}$ is the Wald statistic for the 95th percentile of the simulated data.

This procedure is implemented by `Wald_stationary()` function.

```
[pvalue, Wald, Trans_Wald] =
Wald_stationary(act_data(var_no,:),boots_data(var_no,:,:),var_order,var_var
iance);
```

## *Remark: IIW test when shocks are non-stationary*

After we get model residuals $z_t$ , we would like to know if the shocks are stationary. The ADF test is used. Empirical work on the SW model finds that most of the variables are stationary, except the productivity shock (Meenagh et al. , 2012)). For nonstatioanry shocks, we consider the following autoregressive process

$$\Delta z_t = \rho \Delta z_{t-1} + \varepsilon_t \tag{5}$$

And we re-estimate this error process and get the model innovation $\varepsilon_t$ for productivity shocks.[3] For other stationary shocks, we use an AR(1) process in levels, and get the model innovation as usual.

[3] If you find any other non-stationary shocks when you implement the test, you can use same error process. But you had better not rely only on ADF test. Make your own judgement and do not use too many nonstationary shocks. A suggestion is that you only consider nonstationarity for productivity. There is often ambivalence in the tests for stationarity of the shocks and in this case the deciding factor can be the Wald test for the overall model including the assumed status of the shocks.

After that, we modify the error process in “fname.mod” file and update the AR coefficients. And then we run “dynare fname.mod” again to get “A” and “B” matrix and bootstrapped the data from equation (3).

You could add any trend terms found in the errors to the simulated data manually. But in the Wald test, we are normally only interested in the dynamic properties of the data and not in the trend terms. So it is not necessary to add trend terms to the simulated data. The bootstrapped data from equation (3) maintains the dynamic properties of the model. Trend terms can be included in the VAR estimated on the data; then the trend coefficients are ignored in the Wald.

The choice of auxiliary equation follows Davidson et al.(2010) and Meenagh et al.(2012).

To use these methods on non-stationary data we need to reduce them to stationarity. This we do by assuming that the variables are cointegrated with a set of exogenous non-stationary variables, so that the residuals are stationary. We then difference the data and write the relationships as a Vector Error Correction Mechanism, as we now explain.

We suppose that in the class of structural models in which we are interested as potential candidates for the true model the endogenous variable vector $y_t$ can be written in linearised form as a function of lagged y, a vector of exogenous variables $x_t$, $z_t$ and of errors $\varepsilon_t$.

$$y_t = f(y_{t-1}, x_t, z_t, \varepsilon_t) \tag{6}$$

Now we assume that $x_t$ are non-stationary, I(1), variables with drift trends (which may be zero); that $z_t$ are I(0) with deterministic trends (that may be zero) and that $\varepsilon_t$ are exogenous variables defined as before. Thus there are cointegrating relationships in the model that define the ‘trend’ values of y as linear functions of the ‘trends’ in these exogenous variables or $A\bar{y}_t = B\bar{x}_t + C\bar{z}_t$ where for example if $\Delta x_t = \rho \Delta x_{t-1} + d + \varepsilon_{1t}$ then $\bar{x}_t = x_t + \frac{\rho}{1-\rho}\Delta x_t + dt$; we note also that $z_t = c + et + b(L)\varepsilon_{2t}$. Hence $\bar{y}_t = A^{-1}(Bx_t + Cz_t + ft)$. We now define the VECM as:

$$\Delta y_t = C\varepsilon_{1t} + D\varepsilon_{2t} + Ev_t - \Gamma(y_{t-1} - \bar{y}_{t-1}) \tag{7}$$

We can rewrite this as a VAR in the levels of $y_t$, augmented by the arguments of $\bar{y}_t$:

$$\begin{aligned} y_t &= (I - \Gamma)y_{t-1} + \Gamma\bar{y}_{t-1} + \eta_t \\ &= (I - \Gamma)y_{t-1} + \Gamma A^{-1}\left[B\bar{x}_{t-1} + C(c + et) + ft\right] + \eta_t \\ &= Fy_{t-1} + G\bar{x}_{t-1} + ht + \eta_t + cons \end{aligned} \tag{8}$$

where $\eta_t = A^{-1}[C\varepsilon_{1t} + D\varepsilon_{2t} + Ev_t]$.It should be noted that ‘cons’ includes dummy constants for outliers in the errors – we interpret these as effects of one-off events such as strikes. This is our auxiliary equation in the indirect inference testing procedure. We estimate it both on the data and on the data simulated from the model bootstraps. It allows us to test whether the model can capture the relationships in the data; we focus on the matrices F in practice. [4]
This IIW test procedure is implemented by the following function:

[4] A necessary condition for the stationarity of the VECM arguments is that $y_t$ is cointegrated with the elements of $\bar{y}_t$ both in the data and in the bootstrap simulations; we check for this and report if it is not satisfied, as this would invalidate the tests.

```
[pvalue, Wald, Trans_Wald ] =
Wald_nonstationary(act_data(var_no,:),boots_data(var_no,:,:),act_nonstatRes
id,boots_nonstatResid,rho_nonstat,var_order,var_variance);
```

# 3. Details of how to get model residuals and innovations

To get model residuals $\hat{\varepsilon}_t$, there are two ways, the LIML and exact method.

## 3.1 *LIML – when shock AR coefficients are unknown*

Under the LIML method, we only need to know the structural parameters. We do not require knowing the shock process and error distribution. We then get the model residuals from LIML. Suppose the model is

$$A_0 E_t y_{t+1} = A_1 y_t + z_t \tag{9}$$

$$z_t = ?\, z_{t-1} + ?\, \varepsilon_t \tag{10}$$

where $y_{t+1}$ are endogenous variables[5] and $z_t$ are model residuals which may be represented by the VAR, $\varepsilon_{t+1}$ are shock innovations, and are exogenous variables.

Then we get model shocks from

$$z_t = A_0 E_t y_{t+1} - A_1 y_t \tag{11}$$

where $E_t y_{t+1}$ is estimated from LIML. If the equation has expectations in it we need to estimate the expected values. To do this we use the robust instrumental variables methods of McCallum (1976) and Wickens (1982), with the lagged endogenous data as instruments. In practice we estimate a VAR of all the expected variables and use this to calculate the expectations.

In implement the method, we make use dynare function *"fname_dynamic.m"* . When we run *"dynare fname.mod"*, Dynare also produces a *"fname_dynamic.m"* file. This is a function that generates "lhs-rhs" of equation (9). To get model residuals $z_t$, we input $A_0$, $A_1$, $E_t y_{t+1}$, $y_t$ and let $z_t$ to be zero. Then the "lhs-rhs" in equation (9) are the model residuals.

After we get the model residuals $z_t$, we may need to determine the stationarity of the residuals and the structure of the shock process. The default process is AR(1). We then re-estimate the AR(1) process, get AR coefficients and model innovation $\varepsilon_t$. Note that you may want specify your own error process (e.g. ARMA), re-estimate it and get the model innovation $\varepsilon_t$. To do so, you need to amend this function manually.

This procedure is implemented by `GetRes_LIML()` function.

```
[residual,inno,rho_est] = GetRes_LIML(fname,act_data,inx_expect,inx_eqs);
```

[5] Note if there are leads which are bigger or equal than 2 (for example, $E_t y_{1,t+2}$), you need to define a new variable (for example, $y_{1lead} = E_t y_{1,t+1}$) and get $y_{1lead}$ from LIML. $y_{1lead}$ is treated as observed.

After re-estimation, you need to update the error process in dynare and run *"dynare fname.mod"* to get A and B matrices.

## 3.2 *Exact method – when shock AR coefficients are known*

In the exact method, we know the structure of shock process. Suppose it follows an AR(X) process and we know the AR coefficients. We can obtain the model errors from the observed data and model parameters exactly,

For example, suppose the DSGE model is

$$A_0 E_t y_{t+1} = A_1 y_t + z_t \tag{12}$$

$$z_t = D_3 z_{t-1} + E_2 \varepsilon_t$$

The reduced form is

$$x_t = A x_{t-1} + B \varepsilon_t \tag{13}$$

where $x_t = (y_t, z_t, a_t)'$, $a_t$ are the auxiliary variables. The coefficient matrices $A$ and $B$ are derived from (12)

Re-writing it in declaration order[6] :

$$x_t = D x_{t-1} + E \varepsilon_t \tag{14}$$

Or

$$\begin{pmatrix} y_t \\ z_t \end{pmatrix} = \begin{pmatrix} D_1 & D_2 \\ 0 & D_3 \end{pmatrix} \begin{pmatrix} y_{t-1} \\ z_{t-1} \end{pmatrix} + \begin{pmatrix} E_1 \\ E_2 \end{pmatrix} \varepsilon_t \tag{15}$$

In Dynare,

$$\begin{aligned} (y_t - D_1 y_{t-1}) &= D_2 z_{t-1} + E_1 \varepsilon_t \\ &= E_1 E_2^{-1} (E_2 E_1^{-1} D_2 z_{t-1} + E_2 \varepsilon_t) \\ &= E_1 E_2^{-1} z_t \end{aligned} \tag{16}$$

since $E_2 E_1^{-1} D_2 = D_3$ in Dynare. So

$$z_t = (E_1 E_2^{-1})^{-1} (y_t - D_1 y_{t-1}) \tag{17}$$

If the number of independent structural residuals is equal to the number of endogenous variables, $E_1$ is a squared matrix, so the residuals are obtained through the above equation. If the number of independent structural residuals is less than the number of endogenous variables, we make use of part of the endogenous variables and part of $E_1$, which makes $E_1$ a squared matrix, and obtain the residuals through the above equation.

---

[6] where
```
E=B(oo_.dr.inv_order_var,:);
D=[zeros(M_.endo_nbr,M_.nstatic)
A(:,1:M_.nspred)zeros(M_.endo_nbr,M_.nfwrd)];
D=D(oo_.dr.inv_order_var,oo_.dr.inv_order_var);
```

Then, the model innovations are

$$\varepsilon_t = E_2^{-1}(z_t - D_3 z_{t-1}) \tag{18}$$

We can also estimate rhos through the exact method. The exact method is conducted through iteration. We start with a set of rhos, most easily derived from LIML. Get a new set of residuals and rhos from the equations below; repeat until convergence

$$z_t = \hat{R} z_{t-1} + \hat{\varepsilon}_t \tag{19}$$

This procedure is implemented by `Get_Res_Exact()` function.

```
[residual,inno,rho_est] = GetRes_Exact(fname,act_data,[]);
```

# 4. Examples

Two examples, Smets-Wonters (2007) NK model and NK 3-equation model (used by Le et al,2011; Liu and Minford, 2014).

Smets-Wonters model: sw_st.mod

NK 3-equation NK model: NK3eq_st.mod

**Step1: Calculate the model residuals and innovations**.

dynare sw_st.mod;

*LIML method*

[residual,inno,rho_est,nst_inx] = GetRes_LIML(fname,act_data,inx_expect,inx_eqs)

Input:

- fname: fname= M_.fname;.
- act_data: k*T matrix
- ind_lead=[1:7,13:14]; % The variables you used to generate E_t[y_(t+1)] by LIML
- ind_eq=[5 2 1 3 10 13 14]; % Select Equations that contains model residuals, in the order that you declare the residuals in your .mod file

Output:

- Residual: is the structure residuals; k*T matrix
- Innovation: model innovations, exogenous variables ; k*T matrix
- rho_hat: estimated AR coefficients for structure residuals;
- nst_inx: index of nonstationary shocks if there are.

*Exact method*

[residual,inno,rho_est] = GetRes_Exact(fname,act_data,[]);

**Step 2: Derive the simulated data by bootstrapping**

boots_data = Boots_data(fname,act_data,inno,nboot,A,B,stv,type);

Input:

- nboot: number of bootstraps.
- A: oo_.dr.ghx
- B: oo_.dr.ghx
- inno: model innovation
- type: Type=1: residual bootstrap; type =2 parametric simulation

Output:

- boots_data: simulated data k*T*nboot matrix

**Step 3: Compute the Wald statistic.**

[ pvalue, Wald, Trans_Wald] =
Wald(act_data(var_no,:),Boots_data(var_no,:,:),var_order,var_variance);

Input:

- var_no=[1 5 7]; % Choice of variables in the Wald calculation
- var_order=1; % Order of Var in the Wald caculation
- var_variance=1; % var_variance=1 ;including the volatility of shocks

Output:

- Wald: Wald statistics
- Trans_Wald: Transformed Wald

Step3: Nonstatioanry case

```
[Wald, Trans_Mdis_norm ] =
Wald_nonstationary(act_data(var_no,:),boots_data(var_no,:,:),act_nonstatRes
id,boots_nonstatResid,rho_nonstat,var_order,var_variance);
```

Input:

- var_no=[1 5 7]; % Choice of variables in the Wald calculation
- var_order=1; % Order of Var in the Wald caculation
- var_variance=1; % var_variance=1 ;including the volatility of shocks
- act_nonstatResid: % $x_t$ for actual data;
- boots_nonstatResid % $x_t$ for bootstrap data;
- rho_nonstat % rhos

Output:

- Wald: Wald statistics
- Trans_Mdis_norm: Transformed Wald

# 5. The power of the II Wald test

The power of the IIW test is studied by Le et.al (2016).

They examine the power of the Wald test by positing a variety of false models, increasing in their order of falseness. We generate the falseness by introducing a rising degree of numerical mis-specification for the model parameters. Thus we construct a False DSGE model whose parameters were moved x% away from their true values in both directions in an alternating manner (even-numbered parameters positive, odd ones negative); similarly, we alter the higher moments of the error processes (standard deviation) by the same +/ − x%. We may think of this False Model as having been proposed as potentially 'true' following previous calibration or estimation of the original model.

The transformed Wald is calculated each time. The power of the test is the probability of rejecting a false model by the data (or the probability that Transformed Wald is bigger than 1.645).

```
s=1:1000
[pvalue(s), Wald(s), Trans_Wald(s)] =
Wald_stationary(act_data(var_no,:),boots_data(var_no,:,:),var_order,var_var
iance);

power=mean(Trans_Wald>1.645);
```

# 6. Model Estimation by II

As mentioned earlier, the Wald statistic measures the distance between the data and the model. Therefore to estimate the model parameters we can use any minimising algorithm to minimise the Wald for the actual data. The function to minimise takes the coefficients as an input and then does Steps 1–3above, giving the Wald as the output.

```
[ Trans_Wald ] = CalcWald(act_data,fname, coef, LIML_option, var_option)
```

The function `CalcWald` includes the three functions that calculate Wald statistics as stated in section 2(`GetRes_Exact()`,`Boots_data()`,`Wald_stationary()`) . The input is actual data and starting coefficients and output is the transformed Wald .

The `simulannealbnd` algorithm supplied in Matlab is suggested, as it has been found to find global minima. In practice it is better to minimise the Transformed Wald because it is easier to see if we have found a set of parameters where the model is not rejected, as we are just looking to see if we found a Transformed Wald less than 1.645.

```
II_coef=simulannealbnd(@(II_coef)
CalcWald(act_data,fname,II_coef,LIML_option,var_option),coef,lb,ub,options);
```

# References


[1] Davidson, J, Meenagh, D, Minford, P & Wickens, M. 2010. Why crises happen- nonstationary macroeconomics. Working Paper E2010/13, Cardiff Economics Working Papers, Cardiff University

[2]Juillard, M. (2001). Dynare: a program for the simulation of rational expectations models. Computing in economics and finance 213. Society for Computational Economics.

[3]Le, V.P.M., Meenagh, D., Minford, P., Wickens, M., 2011. How much nominal rigidity is there in the US economy — testing a New Keynesian model using indirect inference. Journal of Economic Dynamics and Control 35(12), 2078–2104.

[4] Le, V.P.M., Meenagh, D., Minford, P., Wickens, M., Xu, Y, 2016. Testing macro models by indirect inference: a survey for users. Open Economic Review, 27, 1-38

[5] Liu, C. and Minford, P., 2014. How important is the credit channel? An empirical study of the US banking crisis. Journal of Banking and Finance, 41, 119-134.

[6]McCallum, B. T. (1976). Rational expectations and the natural rate hypothesis: some consistent estimates. Econometrica, 44, 43–52.

[7] Meenagh, D., Minford, P. and Wickens M.R., 2012. Testing macroeconomic models by indirect inference on unfiltered data, Cardiff Working Paper No E2012/17, Cardiff University, Cardiff Business School, Economics Section; also CEPR discussion paper no 9058, CEPR, London.

[8] Smets, F., Wouters, R., 2007. Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach. American Economic Review, 97, 586--606.

[9]Wickens, M.R., 1982. The efficient estimation of econometric models with rational expectations. Review of Economic Studies, 49, 55–67.