MACROECONOMIA IV-AYUDANTIA
Apunte Básico e Introducción a Matlab

PRIMAVERA 2020

El documento fue elaborado por:
Christian M. Huanto Q.
Luis Serrudo L.

1. Introducción a MATLAB

MATLAB es el nombre abreviado de "Matrix Laboratory". Es un programa para realizar cálculos numéricos con vectores y matrices, y por tanto se puede trabajar también con números escalares (tanto reales como complejos), con cadenas de caracteres y con otras estructuras de información más complejas.

Matlab es al mismo tiempo un entorno y un lenguaje de programación. Uno de sus puntos fuertes es que permite construir nuestras propias herramientas reutilizables. Podemos crear fácilmente nuestras propias funciones y programas especiales (conocidos como M-archivos) en código Matlab, los podemos agrupar en Toolbox (también llamadas librerías): colección especializada de M-archivos para trabajar en clases particulares de problemas.

Matlab, a parte del cálculo matricial y álgebra lineal, también puede manejar polinomios, funciones, ecuaciones diferenciales ordinarias, gráficos ...

Octave es el equivalente libre de MATLAB, y puede descargarse del siguiente enlace https://www.gnu.org/software/octave/ . Octave tambi en cuenta con una versi on on line que tiene lasventajas de no utilizar la memoria RAM del ordenador y que puede trabajarse en simult aneo un proyectocon m as personas conectadas al mismo link, esta versión esta disponible en https://octave-online.net/ .

Finalmente, GAUSS es otro entorno que tiene su propio lenguaje de programación similar al de MATLAB. Sus ventajas residen en la capacidad de optimización de código mediante vectorización y una variedad de paquetes que facilitan la programación. Se puede descargar una versión gratuita para estudiantes en https://www.aptech.com/industry-solutions/gauss-in-education/students/ .

2. El Espacio de trabajo de MATLAB

MATLAB tiene 3 ventanas principales: (1) Current folder donde se muestra la dirección de la carpeta del ordenador donde se está trabajando y es útil para organizar varios archivos que se estén utilizando en la programación, (2) Command Window que es la ventana donde se puede programar y ver los resultados de la programación y (3) Workspace que es donde se almacena la información generada mediante la programación. El espacio de trabajo de Octave y el de GAUSS son similares al de MATLAB.

Una forma de guardar la programación y optimizarla es programando en un *script*, que en matlab es un archivo *m-file* y en GAUSS un *g-file*. Para crear un nuevo script teclee ctrl+N.

3. Rápida referencia para programar en MATLAB

En la presente sección presentamos los comandos para empezar a programar en MATLAB y Octave, ya que manejan el mismo lenguaje. Para una referencia rápida de GAUSS con el mismo contenido vea https://www.aptech.com/resources/tutorials/ .

Se presentan ejemplos utilizando esta sección en el *m-file* adjunto.

3.a. Funciones / Comandos básicos

Comando	Descripción
clear;	Elimina todas las variables del workspace.
clc;	Borra todo el texto de la ventana de comandos, limpia la pantalla.

Nota: para la descripción de un comando utilizar F1 cuando el cursor está sobre el comando.

3.b. Creación de matrices

Comando	Descripción
a = 1	Añade la variable *a* (matriz 1x1) al workspace.
a = [1 2 3 4]	Genera una matriz 1x4.
X = [1 2 3; 4 5 6; 7 8 10]	Genera una matriz 3x3.
X = zeros(m,n)	Genera una matriz mxn con todos los elementos igual a cero.
X = ones(m,n)	Genera una matriz mxn con todos los elementos igual a uno.
X = randn(m,n)	Genera una matriz mxn de números aleatorios de una normal estándar.
X = rand(m,n)	Genera una matriz mxn de números aleatorios de una distribución uniforme.
B = 0:10:100	Genera un vector de una secuencia de números de la forma start:step:end.
B = eye(n)	Genera una matriz identidad nxn.

Nota: Si finaliza una instrucción con un punto y coma, MATLAB realiza el cálculo, pero suprime la visualización de la salida en la ventana de comandos.
Reglas para el nombre de variables

- Todos los nombres deben comenzar por una letra.
- El nombre de la variable máximo 63 caracteres
- El nombre solo puede estar compuesto por Letras, números y barra baja.
- Existen diferencias entre mayúsculas y minúsculas es decir A≠a
- Hay nombre de variables que no se pueden usar, para ver estas variables escriba el código **iskeyword**
- Tener cuidado al sobrescribir funciones por ejemplo: si primero ponemos la función **sin(60)** y luego llamamos a una variable **sin=30** la función de seno deja de funcionar.

3.c. Manipulación de matrices

Nota: Aparte del workspace donde se almacenan las matrices creadas, se puede utilizar el comando *whos* para visualizar lo generado.

Para guardar lo almacenado en el workspace (datos), se utiliza el comando *save myfile.mat* y para utilizar datos guardados en un archivo .mat se pueden cargar al workspace utilizando *load myfile.mat* .

Comando	Descripción
a = X(m,n)	Extrae el elemento de X ubicado en (m,n)
a = X(m,:)	Extrae todas las columnas de la fila m.
a = X(:,n)	Extrae todas las filas de la columna n.
a = X(r_start:r_end,:)	Extrae todas las columnas del rango de filas entre r_star y r_end.
a = A([a b], n)	Extrae las filas a y b de la columna n.
A = [a,b]	Concatena horizontalmente las matrices a y b.
A = [a; a]	Concatena verticalmente las matrices a y b.

3.d. Operadores

3.d.1. Operadores Elemento por Elemento

Operadores	Descripción
z = x .* y;	Multiplicación elemento por elemento
z = x ./ y;	División elemento por elemento.
z = x .^{y};	Potenciación elemento por elemento.
z = x + y;	Suma elemento por elemento.
z = x - y;	Resta elemento por elemento.

3.d.2. Operadores de matrices

Operadores	Descripción
Z = X * Y;	Multiplicación de matrices.
b = Y / X;	Resulve un sistema de ecuaciones lineales.
Z = X .*. Z;	Producto Kronecker.
Z = X';	Transpuesta de una matriz.
Z = inv(X);	Inversa de una matriz.

3.d.3. Operadores lógicos Elemento por Elemento y de Matrices

Operadores	Descripción
z = x .> y;	Mayor que Elem x Elem.
z = x .< y;	Menor que Elem x Elem.
z = x > y;	Es todo elemento en x mayor que su correspondiente elemento en y.
z = x < y;	Es todo elemento en x menor que su correspondiente elemento en y.

3.e. Generación de variables string

- Para crear variables tipo string se debe utilizar comillas para encerrar el texto que se desee guardar en una variable.
- Si el texto incluye alguna expresión que requiera usar comillas se debe utilizar doble comillas para que pueda escribirse como se desea.
- Las matrices pueden almacenar variables string, el manejo es el mismo que el visto hasta el momento.

- Si se quiere combinar texto y variables numéricas se utiliza la función de adición +.

3.f. Utilizar funciones de MATLAB

MATLAB proporciona una gran cantidad de funciones que realizan tareas computacionales. Las funciones son equivalentes a subrutinas o métodos en otros lenguajes de programación.

Para llamar a una función, como *max*, encierre sus argumentos de entrada entre paréntesis:

Función	Descripción
M = mean(A)	Almacena en M la media de los elementos de A.
S = sum(A)	Almacena en S la suma de los elementos de A.
M = max(A)	Almacena en M el valor máximo de cada columna de A.
disp(X)	Muestra el valor de la variable sin mostrar el nombre de la variable.
plot(X,Y)	Genera un gráfico de línea 2D de Y con X.
scatter(X,Y)	Genera un gráfico de dispersión 2D de Y con X.

Nota: Cuando se tiene más de un resultado de salida, se utilizan los corchetes de la siguiente manera: utilizando de ejemplo la función *max*, se puede obtener además el posicionamiento del elemento máximo de un vector A, si escribimos en la ventana de comandos

[maxA,location] = max(A)

3.g. Crear funciones en MATLAB

function [y1,...,yN] = myfun(x1,...,xM)
declara una función llamada myfun que acepta entradas x1, ..., xM y devuelve salidas y1, ..., yN. Esta declaración debe ser la primera línea ejecutable de la función. Los nombres de funciones válidos comienzan con un carácter alfabético y pueden contener letras, números o guiones bajos. Al final de la función debe terminarse la programación con *end*.

Puede guardar su función:

- En un function file que contiene solo definiciones de de la función: el nombre del archivo debe coincidir con el nombre de la primera función en el archivo.
- En un script que contiene comandos y definiciones de funciones: las funciones deben estar al final del archivo. Los script no pueden tener el mismo nombre que una función en el archivo. Las funciones se admiten en scripts en R2016b o posterior.

Ejemplos de funciones adjuntas en el m-file.

3.h. Loops y optimización de código

Muchos programas creados en lenguajes de programación pasan mucho tiempo dentro de loops (operaciones repetitivas). Muchas veces, en vez de utilizar algoritmos repetitivos que demandan mucho tiempo y memoria RAM a la computadora, se puede optimizar el código vectorizandolo. Para esto GAUSS facilita su lenguaje, pero en MATLAB también puede hacerse. Sin embargo, muchas veces se tendrán que usar loops de todas maneras, para lo cual también es bueno tener en cuenta algunos tips para optimizar el código y no demandar innecesariamente trabajo al computador, aumentando el tiempo en que corre el código.

3.h.1. Utilizar loops definidos, no condicionales

Los loops condicionales son loops que continúan iterando hasta que una declaración dada se evalúa como verdadera o falsa. Ejemplo de esto es usar *while* o *until*. Estos loops pueden resultar intuitivos y fáciles de usar, sin embargo, su equivalente utilizando *for* tiene algunas ventajas, una de ellas es que es mas veloz.

Si se comparan ambos loops, se notará que el loop for es más compacto y mantiene todo el control del loop en un solo lugar. Esto significa que escribirá menos y tampoco tendrá que buscar en el cuerpo del loop, posiblemente largo, para localizar la línea que está incrementando su contador del loop.

¿Cuánto tiempo ahorrarán los loops *for* mi programa? Es raro que la implementación del loop real ocupe una parte significativa del tiempo de ejecución total. Por lo tanto, es probable que el aumento de velocidad sea modesto. Sin embargo, el uso de loops *for* tiene muchas otras ventajas y, en casi todos los contextos, no hay desventajas. Como tal, deben usarse de forma predeterminada.

3.h.2. Evitar la concatenación de matrices en los loops

Evitar la concatenación de matrices en loops es una de las herramientas de optimización más poderosas para programar. También es sencillo de implementar en la mayoría de los casos. Comencemos con un código simple que

- Cree un vector aleatorio.
- Calcule su media.
- Guarde esta media como el siguiente elemento en el vector.

Para lograr el resultado sin concatenación, debe preasignar el vector de interés a su tamaño final. Luego, en el ciclo, establecer las filas individuales en el nuevo valor en lugar de agregar otra fila en la parte inferior.

Comparando ambos códigos, el cambio de código es bastante pequeño. Ambos también son bastante legibles. Sin embargo, el código con el vector de salida preasignado puede ejecutarse de 5 a 20 veces más rápido, según el tamaño de los vectores y la memoria disponible en su máquina.

3.h.3. Evitar trabajos innecesarios en los loops

Nota: para correr una parte seleccionada del script en vez del script entero usar F9.

Formatos de Salida de las expresiones numéricas

Comando	Descripción
format long	Número con 14 decimales
format bank	Número con 2 decimales
format short	Número con 4 decimales(default)
format +	Muestra solo el signo de los números.
format rat	Formato de números racionales
format short e	Notación científica en 4 decimales
format long e	Notación científica de 14 dígitos

4. Estática comparativa en el modelo IS-LM

4.a. IS

Del mercado de bienes, partimos de la identidad contable de uso de fondos.

$$Y = c + I + G$$

Dos críticas importantes a esto:

- Top-Bottom
- Bloques separados (insconsistentes)

El consumo

$$C = c(Y - T, r, V)$$

La inversión

$$I = I(r)$$

Derivación de la IS: Y = f(r)

$$Y = c(Y - T) + I(r) + G \tag{1}$$

Diferenciamos totalmente la expresión anterior:

$$dY = \frac{\partial C}{\partial Y}dY + \frac{\partial C}{\partial T}dT + \frac{\partial I}{\partial r}dr + dG$$

$$dY = c_1 dY - c_1 dT + I_1 dr + dG \tag{2}$$

$$dY = c_1 dY - 0 + I_1 dr + 0$$
$$(1 - c_1)dY = I_1 dr$$
$$\left.\frac{dr}{dY}\right|_{IS} = \frac{1 - c_1}{I_1} < 0$$

4.b. LM

Del mercado de dinero

$$\frac{M^S}{P} = \frac{M^D}{P} = \frac{M}{P} = L(Y, r) \tag{3}$$

Diferenciamos totalmente la expresión anterior:

$$\frac{dM}{P} - \frac{M}{P^2}dP = L_1 dY + L_2 dr \tag{4}$$

de lo que se obtiene

$$\left.\frac{dr}{dY}\right|_{LM} = -\frac{L_1}{L_2} > 0$$

4.c. Equilibrio en en el mercado del producto y del dinero

Definimos las funciones implícitas de ambos mercados, que son las ecuaciones 1 y 2.

$$Y - c(Y - T) - I(r) - G = 0$$
$$\frac{M}{P} - L(Y, r) = 0$$

Aplicamos la diferencial de todo el sistema, que son las ecuaciones 2 y 4

$$dY - c_1 dY + c_1 dT - I_1 dr - dG = 0$$
$$\frac{dM}{P} - \frac{M}{P^2}dP - L_1 dY - L_2 dr = 0$$

Ordenamos de tal manera que queden las variables endógenas a la izquierda y las exógenas a la derecha del sistema de ecuaciones.

$$dY - c_1 dY - I_1 dr = dG - c_1 dT$$
$$-L_1 dY - L_2 dr = \frac{M}{P^2}dP - \frac{dM}{P}$$

Reescribimos el sistema de forma matricial

$$\begin{bmatrix} 1-c_1 & -I_1 \\ -L_1 & -L_2 \end{bmatrix} \begin{bmatrix} dY \\ dr \end{bmatrix} = \begin{bmatrix} dG - c_1 dT \\ \frac{M}{P^2} dP - \frac{dM}{P} \end{bmatrix}$$

Dado que el determinante del Jacobiano es distinto de cero, aseguramos que existe una solución al sistema de ecuaciones.

Para resolver problemas de estática comparativa utilizamos la regla de la función implícita que es una aplicación de la resolución por Cramer del sistema construido.

Ejemplo 1: Cuál es el efecto del Gasto (pol fiscal expansiva) financiado por deuda

$$\frac{\partial Y}{\partial G} = \frac{det(J_i)}{det(J) = \Delta}$$

$$det(J_i) = \begin{vmatrix} 1 & -I_1 \\ 0 & -L_2 \end{vmatrix} = -L_2$$
$$\Delta = -L_2 (1 - c_1) - I_1 L_1 > 0$$

$$\frac{\partial Y}{\partial G} = \frac{-L_2}{-L_2 (1 - c_1) - I_1 L_1} = \frac{1}{1 - c_1 + I_1 \frac{L_1}{L_2}} > 0$$

Ejemplo 2: Cuál es el efecto del Gasto (pol fiscal expansiva) financiado por emisión monetaria

$$\left.\frac{\partial Y}{\partial G}\right|_{dG=dM} = \frac{det(J_i)}{\Delta}$$

$$det(J_i) = \begin{vmatrix} 1 & -I_1 \\ -1/P & -L_2 \end{vmatrix} = -L_2 + I_1(-1/P)$$

$$\left.\frac{\partial Y}{\partial G}\right|_{dG=dM} = \frac{-L_2 - \frac{I_1}{P}}{-L_2(1 - c_1) - I_1 L_1} > 0$$

Ejercicio propuesto. Cuál es el efecto sobre el producto de un aumento del gasto financiado con impuestos siguiendo una regla de presupuesto equilibrado?

5. Pautas y consejos para los ejercicios

Lo que tienen que hacer es básicamente lo siguiente, como saben en el modelo IS-LM las variables dependiente son Y y r por lo tanto primero hallan las expresiones para la IS-LM, es decir:

$$IS: \quad Y = f(consumo, inversion, Gastodegobierno)$$

$$IS: \quad Y = f(T, r, G)$$

Para la LM

$$LM: \quad \frac{M}{P} = f(Producto, tipodeinteres)$$

$$LM: \quad r = f(Y, \frac{M}{P})$$

5.a. Pasos para la estimación:

Paso 1: Ver si las ecuaciones son sub-identificadas, exactamente identificadas o Sobre identificadas
paso 2: Si ambas son exactamente o sobre identificadas se pueden recuperar de ambas los parametros
Paso 3: los modelos quedarían mas o menos de la siguiente forma.

$$IS: \quad Y = \beta_0 + \beta_1 T + \beta_2 r + \beta_3 G + u_t$$

$$LM: \quad r = \gamma_0 + \gamma_1 Y + \gamma_2 \frac{M}{P} + v_t$$

y encontramos la forma reducida que es solo remplazar la IS en la LM o la LM en la IS

$$\text{Forma reducida IS}: \quad Y = f(T, \frac{M}{P}, G)$$

$$\text{Forma reducida LM}: \quad r = f(T, G, \frac{M}{P})$$

Paso 4: ahora para recuperar parametros consistentes regresionan cualquiera de las dos formas reducidas y encuentro el valor esperado de la variable dependiente $\hat{r}$ o $\hat{Y}$.
Paso 5: Con cualquiera de las dos variables ahora puedo regresionar la IS o LM, por ejemplo

$$LM: \quad r = \gamma_0 + \gamma_1 \hat{Y} + \gamma_2 \frac{M}{P} + v_t$$

o tambie la IS

$$IS: \quad Y = \beta_0 + \beta_1 T + \beta_2 \hat{r} + \beta_3 G + u_t$$

Paso 6: Ahora para recuperar los parametros solo es despejar los coeficientes estimados y recuperar cada parametro, por ejemplo

$$\hat{\beta}_0 = \frac{c_0 + i_0}{1 - c1}$$

$$\hat{\beta}_1 = -\frac{c_1}{1 - c_1}$$

... etc
¿Problemas con $c_0 + i_0$? Respuesta ecuación de consumo o inversión pueden ayudarnos pero considerando $\hat{r}$ o $\hat{Y}$ segun corresponda. ejemplo $I = f(\hat{r})$ y usar esos coeficientes

5.b. Multiplicadores

Una forma alternativa de poder encontrar los multiplicadores es la siguiente

$$Y = c(Y - T) + I(r) + G$$

$$\frac{M}{P} = L(Y, r)$$

Diferenciando

$$dY = c1dY - c_1dT) + I_1dr + dG$$

$$\frac{dMP - MdP}{P^2} = L_1dY + L_2dr$$

despejo a lado derecho las variables dependientes y al lado izquierdo las independientes

$$(1 - c_1)dY - I_1dr = -c1dT + dG$$

$$L_1dY + L_2dr = \frac{1}{P}dM + \frac{M}{P^2}dP$$

Lo paso a forma matricial

$$\begin{bmatrix} 1 - c_1 & -I_1 \\ L_1 & L_2 \end{bmatrix} \begin{bmatrix} dY \\ dr \end{bmatrix} = \begin{bmatrix} -c_1 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{p} & -\frac{M}{P^2} \end{bmatrix} \begin{bmatrix} dT \\ dG \\ dM \\ dP \end{bmatrix}$$

encuentro el determinante del sistema:

$$\Delta = L_2(1 - c_1) + I_1L_1$$

Ahora para encontrar cualquier multiplicador usando el metodo de cramer por ejemplo :

$$\frac{dY}{dG} = \frac{det}{Det_sistema} = \frac{\begin{vmatrix} 1 & -I_1 \\ 0 & L_2 \end{vmatrix}}{\Delta} = \frac{L_2}{L_2((1 - c_1) + \frac{I_1L_1}{L_2})} = \frac{1}{(1 - c_1) + \frac{I_1L_1}{L_2}} > 0$$

Se recomienda que practiquen o realicen todos los multiplicadores por ejemplo $\frac{dY}{dG}$, $\frac{dY}{dT}$, $\frac{dY}{dM}$, $\frac{dY}{dP}$, $\frac{dr}{dG}$, $\frac{dr}{dT}$, $\frac{dr}{dM}$, $\frac{dr}{dP}$

5.c. Ejercicio RATEX

Encontrar la regla Optima de política monetaria y el producto dde largo plazo de la siguiente curva de oferta a la Lucas

$$y_t = \gamma[p_t - E_{t-1}p_t] + \lambda y_{t-1} + u_t \quad u_t \sim (0, \sigma_u^2)$$

y la demada de dinero

$$m_t - p_t = y_t + v_t \quad v_t \sim (0, \sigma_v^2)$$

Resolucion: Primero encontramos como fijan los precios los agentes o la expectativa del precio el dia de hoy

$$E_{t-1}p_t = E_{t-1}m_t - E_{t-1}y_t \tag{5}$$

ahora encontraremos como fijan sus expectativas sobre el producto (pero considerando como fijan las expectativas sobre el precio)

$$y_t = \gamma(m_t - y_t - v_t - E_{t-1}m_t + E_{t-1}y_t) + \lambda y_{t-1} + u_t$$

aplicando esperanza

$$E_{t-1}y_t = \overline{y} = \lambda y_{t-1} \quad \textbf{Producto de largo plazo}$$

Ahora para encontrar el valor que minimice el error cuadrático medio, y como la autoridad solo controla la emisión monetaria, debemos encontrar el m_t optimo que es solo encontrar ;el valor minimo del MSE con respecto a m_t y teniendo encuenta que no existira volatilidad $y_t = E_{t-1}y_t$

$$Min_M SE = E(y_t - y^*)^2 = E(\gamma(m_t - y_t - v_t - E_{t-1}m_t + E_{t-1}y_t) + \lambda y_{t-1} + u_t - \lambda y_{t-1})^2$$

$$\frac{\partial MSE}{\partial m_t} = 0$$

donde al final se encuentra que:

$$m_t = E_{t-1}m_t$$

Lo cual esa es **la política óptima** que debe elegir el tomador de decisiones.

6. Exogeneidad

En esta sección abordaremos la exogeneidad desde el punto de vista de la econometría inglesa. En la literatura de econometría Inglesa existen tres tipos de exogeneidad y cada una tiene una distinta manera de testearse que debe cumplir ciertas condiciones.

6.a. Exogeneidad Débil

Mayormente un primer ejercicio que se realiza en un modelo es la Inferencia y para realizar dicho ejercicio se debe cumplir que una variable x_t (Variable exógena) cumpla con exogeneidad débil, aquí uno debe pensar que el mundo no están complicado, hay que ir testeando, además no debería haber heterogeneidad si se tiene puesto muchas x_t (Variables explicativas).
Es algo que debe cumplirse sobre todo cuando condicionamos nuestro modelo con alguna variable contemporánea, por ejemplo:

$$y_t = \alpha + \beta' x_t + \gamma\ y_{t-1} + u_t$$

Dadas unas variables contemporáneas necesitamos que se cumpla con el supuesto de Exogeneidad Débil porque si no $\hat{\beta}$ será un estimador inconsistente para poder hacer inferencia robusta, por lo que uno debe testear exogeneidad débil, ya que se necesita que la marginal y la condicional sean libres de variación, si en el caso de que no exista exogeneidad (endogeneidad) uno puede utilizar variables instrumentales o ecuaciones simultaneas o rezagos de dicha variable.
Para testear la exogeneidad débil se realiza mediante un test indirecto, primero se debe correr la regresión del modelo anterior y obtener los $\hat{u}_t$, luego se procede a estimar una forma para la marginal de x_t.

$$x_t = \rho\ x_(t-1) + v_t$$

Por ejemplo para el modelo de la marginal se puede usar un modelo de serie de tiempo, pero se debe cumplir que v_t sea Ruido Blanco. Para finalizar este test indirecto uno debe regresionar $\hat{u}_t = f(x_t, y_{t-1}, \hat{v}_t)$ y debe evaluar si se cumple la siguiente hipótesis.

$$H_o : \hat{v}_t = 0 \sim \chi^2(1) \qquad \exists \text{ Exogeneidad Débil}$$

6.b. Exogeneidad Fuerte

Un segundo ejercicio es cuando uno esta interesado o queremos hacer forecast o proyecciones, en este caso uno desea ver si y_{t+h} se puede proyectar con los hechos y ver si el pasado de x_t me ayuda a predecir o proyectar y_t, en este se requiere que x_t sea fuertemente exógena para el parámetro de interés
Pasemos a ver como podemos probar que si una variable es fuertemente exógena, por ejemplo, si se tiene un modelo:

$$y_t = \alpha + \beta' x_t + u_t$$

Y además, se tiene sospechas que el modelo para x_t es

$$x_t = \gamma y_{t-1} + v_t$$

Sería ideal que ocurriera que y_t no cause en el sentido de Granger a x_t, es decir lo que se debería testear es que x_t sea fuertemente exógena (es decir x_t debe ser exógena débil y además y_t no causa a x_t).

Podría ocurrir el caso de que uno no tenga una variable que sea fuertemente exógena por lo cual eso no va a impedir que yo no pueda hacer forcast, lo que uno necesita para que se cumpla el aspecto de futuro relativo de Hendry es que el modelo sea estable y confiable.
Para evaluar si un modelo es estable o no presenta cambios de régimen la manera mas sencilla es usar un test de Chow, además la estabilidad puede generarse a partir de no-linealidades omitidas por lo cual es recomendable hacer un Test Reset. Otra forma de ver la estabilidad de un modelo es usando los famosos test de Cusum y Cusum-cuadrado. Si además el investigador como ya se menciono antes esta interesado en realizar escenarios contra-factuales este debe evaluar si x_t es súper-exógena (es decir la variable debe ser débilmente exógena y además la distribución de la condicional es estructuralmente invariante o los parámetros de la condicional son estables).

6.c. Súper Exógena

Un ultimo ejercicio que en la literatura con frecuencia se quiere realizar es ver o evaluar escenarios contra-factuales, que es muy distinto a hacer inferencia o forcast, ya que ahí asumimos que la estructura no cambia, pero cuando uno realiza escenarios contra factuales yo estoy implícitamente asumiendo que la estructura va a cambiar (es decir uno esta haciendo por ejemplo una reforma tributaria, cambio de política, etc) y de acuerdo a la teoría Lucas nos diría si cambia la estructura debería cambiar la forma en que y_t se relaciona con x_t y por lo tanto uno no podría utilizar la relación que tenia antes para decir que pasa cuando cambia x_t, En este caso Hendry[1] nos presenta una forma de testear la critica de Lucas y concluir si una variable es súper exógena. Vamos a decir que una variable x_t es súper exógena para el parámetro de interés si es que aun ante cambios en la ley de movimiento de las x_t la relación entre y_t y x_t se mantiene estable y al final hacer ejercicios contra-factuales.
Como se dijo si además el investigador como ya se menciono antes esta interesado en realizar escenarios contra-factuales este debe evaluar si x_t es súper-exógena (es decir la variable debe ser débilmente exógena y además la distribución de la condicional es estructuralmente invariante o los parámetros de la condicional son estables).
Solo por aclaración cuando uno evalúa si una variable es súper exógena es como si se estuviera testeando la Critica de Lucas, ya que como sabemos la critica de Lucas nos dice que si cambia la ley de movimiento de los estados entonces cambia la función de política o en el lenguaje econométrico nos dice que si cambia la marginal de x_t, entonces cambia la condicional de y_t en x_t. Si uno desea testear la critica de Lucas necesita que cambie la marginal de x_t, es decir uno necesita un modelo para x_t y este tiene que ser inestable, si esto se cumple uno ahora debe testear si la condicional de y_t es estable, si en el caso de que la condicional no es estable se cumple la critica de Lucas por lo que uno no puede realizar escenarios contra-factuales, pero si en el caso de que la condicional de y_t es estable y además la marginal de x_t es inestable en este caso se rechaza la critica de Lucas y mi variable x_t es súper-exógena y además puedo realizar escenarios contra-factuales.

[1] Ver el libro de Dynamic Econometrics - David F. Hendry