

# Modelización, Computación y Calibración Macroeconómica

José L. Torres

Universidad de Málaga

UHU, 12 de Julio 2024

- 1 Equilibrio general dinámico computable
- 2 El algoritmo de Newton-Raphson
- 3 Métodos de resolución de los modelos DSGE
- 4 Las funciones de política

# 1. Equilibrio general dinámico computable

- Modelos macroeconómicos actuales: Modelos de equilibrio general dinámicos computables.
- Sólo bajo supuestos muy restrictivos tienen soluciones explícitas y solo pueden obtenerse soluciones numéricas.
- Tienen que ser resueltos en un ordenador.
- Necesidad de disponer de formas funcionales, valores de los parámetros y valores de las variables exógenas.
- Dos métodos para calcular los parámetros:
  - Calibración
  - Estimación econométrica (Maximaverosimilitud o Bayesiana).

# 1. Equilibrio general dinámico computable

- Los modelos EGDE tienen parámetros constantes y perturbaciones de media cero.
- Cambios esperados da lugar a respuestas anticipadoras por parte de los agentes.
- Cambios anticipados forman parte del conjunto de información en función del cual se toman las decisiones.
- Los cambios estructurales se representan como una variable exógena determinista.
- Cambios no anticipados simplemente alteran el comportamiento de las variables una vez ocurren pero no entran en las expectativas.

# 1. Equilibrio general dinámico computable

- El procedimiento para resolver estos modelos contiene los siguientes pasos:
  - Calcular las condiciones de primer orden, que junto con las condiciones de factibilidad nos van a caracterizar el equilibrio.
  - Calibrar los parámetros y calcular el Estado Estacionario.
  - Log-linearizar las ecuaciones que caracterizan el equilibrio, con el objeto de hacer estas ecuaciones aproximadamente lineales en términos de su log-desviación respecto al Estado Estacionario.
  - Resolver el sistema de ecuaciones resultante para obtener las funciones de política y funciones de transición.

# 1. Equilibrio general dinámico computable

- En realidad no resolveremos numéricamente el modelo de partida (sistema de ecuaciones no lineales y dinámicas), sino que resolvemos numéricamente dos modelos:
  - Una transformación del modelo eliminando la dinámica (variables endógenas constantes a lo largo del tiempo). En este caso resolvemos un sistemas de ecuaciones estáticas, aunque las ecuaciones son no lineales. Esto nos permite obtener una solución numérica del estado estacionario.
  - Una aproximación lineal al modelo dinámico. En este caso resolveremos un sistema de ecuaciones dinámicas, pero con ecuaciones lineales a partir de la solución obtenida para el sistema anterior estático.

# 1. Equilibrio general dinámico computable

- Necesidad de utilizar un lenguaje de computación. Diferentes opciones:
  - Los primeros códigos para resolver modelos DSGE fueron elaborados utilizando lenguajes compilados: FORTRAN y C/C++. Algunos autores los siguen usando hoy en día, dado su poder de computación.
  - Posteriormente, se ha extendido el uso de lenguajes interpretados, fundamentalmente Matlab/Octave, pero también GAUSS, Mathematica y R. Más recientemente se han desarrollado códigos en Python y en Julia.
  - Desarrollo de pre-procesadores específicos para resolver modelos DSGE.
  - Para modelos sencillos podemos usar una hoja de cálculo como Excel (uso del solver del Excel o método de los valores propios).  
[www.macrodynamicsmodels.com](http://www.macrodynamicsmodels.com)

# 1. Equilibrio general dinámico computable

- Pre-procesadores:
  - Dynare. Elaborado por el CEPREMAP (Francia). Desarrollado para Matlab/Octave.
  - Es el más extendido y utilizado en la modelización DSGE.
  - Puede resolver, simular y estimar modelos DSGE o cualquier otro modelo dinámico.

# 1. Equilibrio general dinámico computable

- Alternativas a Dynare:
  - gEcon: Desarrollado en R por el Departamento de Análisis Estratégico de la Oficina del Primer Ministro de Polonia.
  - Principal características: gEcon puede resolver directamente modelos DSGE a partir del problema de maximización de los agentes.  
<http://gecon.r-forge.r-project.org>
  - IRIS: Conjunto de programas en Matlab desarrollado inicialmente por el FMI. IRIS puede resolver, simular y estimar modelos DSGE.  
<http://iris-toolbox.com>
  - YADA: Es una interface gráfica de usuario (GUI) basada en Matlab y desarrollada por el BCE. <http://www.texlips.net/yada/>

# 1. Equilibrio general dinámico computable

- Alternativas a Dynare:
  - PyMacLab: Python Macroeconomics Laboratory, es una librería en Python para resolver modelos DSGE desarrollada por Erich Scheffel. <http://pypi.python.org/pypi/pymaclab>
  - SolveDSGE for Julia. Librería desarrollada en Julia por Richard Dennis.
  - Dolo/Jolo. Desarrollado por Pablo Winart.

# 1. Equilibrio general dinámico computable

- MMDB: Volker Wieland's Macroeconomic Model Database. Se trata de un recurso web con un gran número de modelos escritos en Dynare.
- 14 Modelos DSGE calibrados.
- 26 Modelos DSGE estimados para los Estados Unidos.
- 9 Modelos DSGE estimados para la zona euro, incluyendo el modelo QUEST III.
- 7 Modelos DSGE multipaís.
- 5 Modelos DSGE estimados para otras economías.
- Posibilidad de comparar varios modelos online.

# 1. Equilibrio general dinámico computable

- Otro recurso web interesante es el International Network for DSGE modeling, fiscal and monetary policy
- [www.dsge.net](http://www.dsge.net)
- Incluye una gran cantidad de herramientas numéricas para la solución de modelos DSGE, tales como el algoritmo Anderson-Moore (AIM), las herramientas de Uhlig, y otros recursos en Matlab.

## 2. El algoritmo de Newton-Raphson

- Al resolver el modelo obtenemos un sistema de ecuaciones compuesto por las condiciones de primer orden y por las restricciones de factibilidad.
- Tendremos  $n$  ecuaciones con  $n$  incógnitas:  $F : R^n \longrightarrow R^n$ .
- El problema consiste en encontrar un vector  $\hat{x} = (\hat{x}_1, \dots, \hat{x}_n)$  de  $R^n$  tal que su imagen por  $F : R^n \longrightarrow R^n$  sea  $F(\hat{x}) = 0$ .

## 2. El algoritmo de Newton-Raphson

- Necesitamos dar al ordenador una solución inicial,  $x_0$ . A esto se le llama la "semilla". Se debe usar la información que tengamos sobre  $F$ .
- Debemos establecer un criterio de tolerancia. Valor mínimo de margen de error que consideramos aceptable.

## 2. El algoritmo de Newton-Raphson

- El objetivo que se persigue es encontrar una solución,  $x$ , tal que  $F(x)$  sea igual a cero. Para ello partimos de desarrollar la serie de Taylor de la función para un determinado valor inicial  $x_n$ :

$$F(x) = F(x_n) + F'(x_n)(x - x_n) + \frac{F''(x_n)}{2!}(x - x_n)^2 + \dots$$

- El procedimiento es iterativo y consiste en ir calculando la anterior expresión para las distintas aproximaciones resultantes. Esta expresión la evaluaríamos para diferentes valores tal que:

$$F(x_{n+1}) = F(x_n) + F'(x_n)(x_{n+1} - x_n) + \frac{F''(x_n)}{2!}(x_{n+1} - x_n)^2 + \dots$$

## 2. El algoritmo de Newton-Raphson

- La solución vendría dada por el valor que hace que la expansión de Taylor sea igual a cero:

$$F(x_n) + F'(x_n)(x_{n+1} - x_n) + \frac{F''(x_n)}{2!}(x_{n+1} - x_n)^2 + \dots = 0$$

## 2. El algoritmo de Newton-Raphson

- Si truncamos la expansión de Taylor a partir del término de grado 2, obtenemos que:

$$F(x_n) + F'(x_n)(x_{n+1} - x_n) \simeq 0$$

- La aproximación anterior es más exacta cuanto más cerca estemos de la solución. Despejando la solución obtenida resulta:

$$x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)}$$

que es a lo que se denomina la fórmula de Newton-Raphson.

## 2. El algoritmo de Newton-Raphson

- El algoritmo sería el siguiente. Comenzamos por un valor inicial de  $x_0$ . La solución que obtendríamos de aplicar el algoritmo de Newton-Raphson sería.

$$x_1 = x_0 - \frac{F(x_0)}{F'(x_0)}$$

- Si  $F(x_1)$  es cero el proceso finaliza.
- Si  $F(x_1)$  es diferente de cero entonces procedemos a una segunda iteración calculando:

$$x_2 = x_1 - \frac{F(x_1)}{F'(x_1)}$$

## 2. El algoritmo de Newton-Raphson

- De nuevo, si  $F(x_2)$  es diferente de cero entonces procedemos a una tercera iteración calculando:

$$x_3 = x_2 - \frac{F(x_2)}{F'(x_2)}$$

y así, hasta que encontremos una solución tal que el valor de la función sea cero.

## 2. El algoritmo de Newton-Raphson

- Criterio de tolerancia. Este criterio determina en cuanto se puede desviar la solución del cero absoluto. En cada iteración podemos calcular una aproximación al error relativo absoluto, que lo definimos como:

$$|\varepsilon_a| = \left( \frac{x_{n+1} - x_n}{x_n} \right) \times 100$$

- Si  $|\varepsilon_a|$  es mayor que un valor fijado a priori,  $\varepsilon$  (la tolerancia), entonces el algoritmo procede a realizar una nueva iteración. En el caso en el que el error relativo absoluto sea inferior al criterio de tolerancia, el algoritmo finaliza, dado la última iteración la solución al mismo.

## 2. El algoritmo de Newton-Raphson

- Ejemplo: En el caso en que la ecuación que queremos resolver sea lineal, el algoritmo de Newton-Raphson encuentra la solución de forma directa, ya que el error que comete es cero. Supongamos que queremos encontrar el cero para la siguiente función lineal:

$$F(x) = x - 2$$

- Oviamente la solución a la anterior función es  $x = 2$ . Imaginemos que no lo sabemos y creemos que su valor debe ser 5, ( $x_0 = 5$ ). Si aplicamos el algoritmo de Newton-Raphson entonces tendríamos que:

$$x_1 = 5 - 3 = 2$$

- Evaluando la función para dicha solución resulta:

$$F(x_0) = x_0 - 2 = 2 - 2 = 0$$

## 2. El algoritmo de Newton-Raphson

- Ejemplo: Resolver la raíz cuadrada de un número  $y$ . Dado que  $x = \sqrt{y}$ , también podemos escribirlo como  $x^2 = y$ . Esto significa que podemos resolver y encontrar los ceros para la siguiente función:

$$F(x) = x^2 - y$$

- Aplicando el algoritmo de Newton-Raphson obtenemos que:

$$x_{i+1} = x_i - \frac{x_i^2 - y}{2x_i} = \frac{2x_i^2 - x_i^2 + y}{2x_i} = \frac{1}{2} \left( x_i + \frac{y}{x_i} \right)$$

## 2. El algoritmo de Newton-Raphson

- En general tenemos un sistema de ecuaciones. Aproximando las funciones a través de la primera expansión de Taylor:

$$F(x) \approx F(\bar{x}) + J(\bar{x})(x - \bar{x})$$

donde  $J(\bar{x})$  es la matriz jacobiana de  $F$  evaluada en  $\bar{x}$ :

$$J(\bar{x}) = \begin{bmatrix} F_{11}(\bar{x}) & F_{12}(\bar{x}) & \dots & F_{1n}(\bar{x}) \\ F_{21}(\bar{x}) & F_{22}(\bar{x}) & \dots & F_{2n}(\bar{x}) \\ \dots & \dots & \dots & \dots \\ F_{n1}(\bar{x}) & F_{n2}(\bar{x}) & \dots & F_{nn}(\bar{x}) \end{bmatrix}$$

donde  $F_{ij}(\bar{x}) = \frac{\partial F_i(\bar{x})}{\partial x_j}$ .

## 2. El algoritmo de Newton-Raphson

- Teorema de Taylor: Conforme  $\bar{x}$  se acerca al valor  $x$ , los términos de orden mayor tienden a cero. Dado que estamos buscando un cero de la ecuación  $F(x)$ , la expresión anterior podemos evaluarla en  $\hat{x}$  y escribirla como:

$$\hat{x} \approx \bar{x} - J(\bar{x})^{-1}F(\bar{x})$$

## 2. El algoritmo de Newton-Raphson

- El algoritmo de Newton-Raphson funciona de la siguiente forma:
1. A partir de la semilla propuesta,  $x_0$ , evaluamos la función  $F(x_0)$  y  $J^{-1}(x_0)$ , para calcular:

$$x_1 = x_0 - J(x_0)^{-1}F(x_0)$$

2. Dado el nivel de tolerancia fijado,  $\varepsilon$ , calculamos la distancia entre  $x_0$  y  $x_1$ . Si la distancia es inferior a  $\varepsilon$ , entonces nos quedamos con  $x_0$  como la solución. En caso contrario, volvemos a repetir el paso 1 pero con el valor  $x_1$ , evaluando  $F(x_1)$  y  $J^{-1}(x_1)$ , para calcular:

$$x_2 = x_1 - J(x_1)^{-1}F(x_1)$$

3. Repetir el paso 2. Si la distancia es inferior al nivel de tolerancia repetir el paso 1, con el valor  $x_2$ , evaluando  $F(x_2)$  y  $J^{-1}(x_2)$ , para calcular:

$$x_3 = x_2 - J(x_2)^{-1}F(x_2)$$

## 2. El algoritmo de Newton-Raphson

- El algoritmo de Newton-Raphson tiene algunos problemas:
- ① Si la semilla no es buena puede que no exista convergencia o que obtengamos otra solución, en el caso en que existan soluciones múltiples.
- ② Necesidad de disponer de expresiones analíticas de todas las derivadas parciales de  $F$ .

## 2. El algoritmo de Newton-Raphson

- Ejemplo: Encontrar los ceros de la siguiente función:

$$F(x) = (x - 2)(x + 2)$$

- Fácil: directamente  $x = 2$  y  $x = -2$ .

## 2. El algoritmo de Newton-Raphson

```
% Este programa resuelve la ecuación  $F(x)=(x-2)(x+2)$ 
clear all
% Semilla  $x(1)=-10$ ;
% Tolerancia  $\text{crit}=1e-20$ ;
% Número máximo de iteraciones  $\text{maxit}=1000$ ;
for i=1:maxit
    J(i) = 2*x(i);
     $x(i+1) = x(i) - J(i)^{-1} * (x(i)-2) * (x(i)+2)$ ;
    if  $\text{abs}(x(i+1)-x(i)) < \text{crit}$ ; break; end
end
if  $i \geq \text{maxit}$ 
    sprintf('Atención: Número máximo de %g iteraciones
alcanzado', maxit)
end
sprintf('La solución es %g', x(i))
```

## 2. El algoritmo de Newton-Raphson

- Como podemos comprobar si comenzamos con una semilla negativa la solución es  $x = -2$ , pero si ponemos una semilla positiva la solución es  $x = 2$ .
- Por otra parte, construir el Jacobiano era muy fácil. Pero si tenemos una función con un número de variables muy grande, construir el Jacobiano es una tarea complicada. Para resolver este problema podemos calcular derivadas numéricas.

## 2. El algoritmo de Newton-Raphson

- Según la definición de derivada:

$$\begin{aligned} J_1(x) &= \lim_{h_1 \rightarrow 0} \frac{F(x) - F(x_1 + h_1, x_2, \dots, x_n)}{h_1} \\ J_2(x) &= \lim_{h_2 \rightarrow 0} \frac{F(x) - F(x_1, x_2 + h_2, \dots, x_n)}{h_2} \\ &\dots \\ J_n(x) &= \lim_{h_n \rightarrow 0} \frac{F(x) - F(x_1, x_2, \dots, x_n + h_n)}{h_n} \end{aligned}$$

donde  $J_j(x)$  es el vector columna con las  $n$  derivadas parciales con respecto a  $x_j$ .

## 2. El algoritmo de Newton-Raphson

- Podemos aproximar en un ordenador derivadas parciales, fijando un incremento  $h$  lo suficientemente pequeño como para escribir:

$$J_1(x) \approx \frac{F(x) - F(x_1 + h_1, x_2, \dots, x_n)}{h_1}$$

$$J_2(x) \approx \frac{F(x) - F(x_1, x_2 + h_2, \dots, x_n)}{h_2}$$

$$J_n(x) \approx \frac{\dots}{\dots} \frac{F(x) - F(x_1, x_2, \dots, x_n + h_n)}{h_n}$$

## 2. El algoritmo de Newton-Raphson

```
% Este programa resuelve la ecuación  $F(x)=(x-2)(x+2)$  pero
con aproximaciones numéricas
clear all
% Semilla
x(1)=-10;
% Tolerancia
crit=1e-20;
% Incremento
h=1e-8;
% Número máximo de iteraciones
maxit=1000;
for i=1:maxit
J(i) = 1/h*((x(i)+h-2)*(x(i)+h+2)-(x(i)-h-2)*(x(i)-h+2));
x(i+1) = x(i)-J(i)^(-1)*(x(i)-2)*(x(i)+2);
if abs(x(i+1)-x(i))<crit; break; end
end
if i>=maxit
```

## 2. El algoritmo de Newton-Raphson

- Otro ejemplo: Supongamos que tenemos que resolver la raíz cuadrada de un número  $y$ . Dado que  $x = \sqrt{y}$ , podemos escribir la siguiente función  $x^2 = y$ . Esto significa que calcular la raíz cuadrada es equivalente a encontrar el cero en la siguiente función:

$$F(x) = x^2 - y$$

- Usando la fórmula de Newton-Raphson obtenemos:

$$x_{i+1} = x_i - \frac{x_i^2 - y}{2x_i} = \frac{2x_i^2 - x_i^2 + y}{2x_i} = \frac{1}{2} \left( x_i + \frac{y}{x_i} \right)$$

- Por ejemplo, supongamos que  $y = 20$ . La solución a la raíz cuadrada sería  $x = \sqrt{20} = 4.4721$ . Imaginemos que no sabemos nada acerca de dicha solución y que suponemos que la "solución tentativa" es  $x_0 = 3$ .
- La primera iteración del algoritmo de Newton-Raphson nos daría un valor de:

$$x_1 = x_0 - \frac{x_0^2 - y}{2x_0} = \frac{1}{2} \left( x_0 + \frac{y}{x_0} \right) = \frac{1}{2} \left( 3 + \frac{20}{3} \right) = 4.8333$$

- En la siguiente iteración obtendríamos un valor de:

$$x_2 = \frac{1}{2} \left( x_1 + \frac{y}{x_1} \right) = \frac{1}{2} \left( 4.83 + \frac{20}{4.83} \right) = 4.4856$$

- En la siguiente iteración obtendríamos un valor de:

$$x_3 = \frac{1}{2} \left( x_2 + \frac{y}{x_2} \right) = \frac{1}{2} \left( 4.48 + \frac{20}{4.48} \right) = 4.4722$$

## 2. El algoritmo de Newton-Raphson

- Otro ejemplo: Resolver el siguiente sistema de ecuaciones:

$$F_1(x, z) = x - 3z + 1$$

$$F_2(x, z) = 2x + z - 5$$

El Jacobiano correspondiente a este sistema de ecuaciones es:

$$J(x, z) = \begin{bmatrix} \frac{\partial F_1(x, z)}{\partial x} & \frac{\partial F_1(x, z)}{\partial z} \\ \frac{\partial F_2(x, z)}{\partial x} & \frac{\partial F_2(x, z)}{\partial z} \end{bmatrix} = \begin{bmatrix} 1 & -3 \\ 2 & 1 \end{bmatrix}$$

## 2. El algoritmo de Newton-Raphson

- Vamos a proponer como semilla, que  $x = 1$  y  $z = 2$ , que obviamente no es la solución al anterior sistema. Si introducimos dicha semilla en el sistema de ecuaciones obtenemos que:

$$F_1(1, 2) = 1 - 3 \times 2 + 1 = -4$$

$$F_2(1, 2) = 2 \times 1 + 2 - 5 = -1$$

valores que resultan diferentes de cero, por lo que la semilla propuesta no es la solución. Si aplicamos el algoritmo de Newton-Raphson, obtenemos que:

$$\begin{bmatrix} x_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 & -3 \\ 2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} -4 \\ -1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

- Si introducimos esta nueva solución en el sistema de ecuaciones obtenemos:

$$F_1(2; 1) = 2 - 3 \times 1 + 1 = 0$$

$$F_2(2; 1) = 2 \times 2 + 1 - 5 = 0$$

### 3. Métodos de resolución de los modelos DSGE

- Una vez que tenemos una versión lineal (log-lineal) del modelo, podemos proceder a su solución.
- Resolvemos esta aproximación lineal, no el modelo original no lineal (método local).
- En la literatura encontramos diferentes métodos: Blanchard y Kahn (1980), Binder y Pesaran (1995), Uhlig (1999), Klein (2000), Sims (2001).
- Todos los métodos anteriores producen resultados muy similares (aunque con algunas diferencias).

### 3. Métodos de resolución de los modelos DSGE

- Condición de Blanchard y Kahn o condición de rango: El número de valores propios fuera del círculo unitario debe ser igual al número de variables adelantadas.
- Si la condición de rango no se satisface, entonces o bien el estado estacionario no existe o bien hay infinitos caminos para llegar al estado estacionario.

### 3. Métodos de resolución de los modelos DSGE

- Recordemos, el modelo (su aproximación log-lineal) que vamos a resolver es el siguiente:

$$\begin{aligned}\hat{y}_t &= \hat{a}_t + \alpha \hat{k}_t + (1 - \alpha) \hat{l}_t \\ [1 - \beta + (1 - \alpha)\beta\delta] \hat{c}_t &= (1 - \beta + \beta\delta) \hat{y}_t - \alpha\beta\delta \hat{i}_t \\ \hat{k}_{t+1} &= (1 - \delta) \hat{k}_t + \delta \hat{i}_t \\ \hat{y}_t - \hat{c}_t &= \left[ 1 + \frac{\gamma(1 - \alpha)}{(1 - \gamma)} \frac{1 - \beta + \beta\delta}{1 - \beta + (1 - \alpha)\beta\delta} \right] \hat{l}_t \\ E_t \hat{c}_{t+1} - \hat{c}_t &= (1 - \beta + \beta\delta) E_t \hat{y}_{t+1} - (1 - \beta + \beta\delta) E_t \hat{k}_{t+1} \\ \hat{a}_t &= \rho \hat{a}_{t-1} + \varepsilon_t\end{aligned}$$

### 3. Métodos de resolución de los modelos DSGE

- El método propuesto por Blanchard y Kahn (1980) distingue entre variables predeterminadas y no predeterminadas.
- Comenzamos con la construcción de los siguientes dos vectores de desviaciones de las variables respecto a su estado estacionario:

$$x_t = \begin{bmatrix} \hat{y}_t \\ \hat{i}_t \\ \hat{l}_t \end{bmatrix}$$

$$s_t = \begin{bmatrix} \hat{k}_t \\ \hat{c}_t \end{bmatrix}$$

donde el primer vector incluye las desviaciones de la producción, inversión y empleo, mientras que el segundo está formado por las desviaciones del stock de capital y del consumo, esto es, las variables que aparecen no solo en el periodo actual sino también en el periodo futuro.

### 3. Métodos de resolución de los modelos DSGE

- Escribimos el modelo en el espacio de estados.
- En general, el espacio de estados para un modelo DSGE adopta la siguiente forma:

$$Ay_{t+1} = By_t + C\varepsilon_{t+1}$$

- Resolviendo para  $y_{t+1}$ :

$$y_{t+1} = A^{-1}By_t + A^{-1}C\varepsilon_{t+1}$$

- Pero  $A$  puede ser no invertible. El uso de la descomposición de Schur resuelve el problema de no invertibilidad de  $A$ .

### 3. Métodos de resolución de los modelos DSGE

- El modelo vamos a escribirlo como dos sistemas de ecuaciones: El primero para las ecuaciones estáticas y el segundo para las ecuaciones dinámicas. Estos dos sistemas de ecuaciones serían:

$$Ax_t = Bs_t + C\hat{a}_t \quad (1)$$

y

$$DE_t s_{t+1} + FE_t x_{t+1} = Gs_t + Hx_t + I\hat{a}_t \quad (2)$$

### 3. Métodos de resolución de los modelos DSGE

- El primer sistema consiste en las siguiente tres ecuaciones:

$$\hat{y}_t - (1 - \alpha)\hat{l}_t = \hat{a}_t + \alpha\hat{k}_t$$

$$(1 - \beta + \beta\delta)\hat{y}_t - \alpha\beta\delta\hat{i}_t = [1 - \beta + (1 - \alpha)\beta\delta]\hat{c}_t$$

$$\hat{y}_t - \left[ 1 + \frac{\gamma(1 - \alpha)}{(1 - \gamma)} \frac{1 - \beta + \beta\delta}{1 - \beta + (1 - \alpha)\beta\delta} \right] \hat{l}_t = \hat{c}_t$$

### 3. Métodos de resolución de los modelos DSGE

- El segundo sistema consiste en las siguientes dos ecuaciones:

$$(1 - \beta + \beta\delta) E_t \hat{k}_{t+1} + E_t \hat{c}_{t+1} - (1 - \beta + \beta\delta) E_t \hat{y}_{t+1} = \hat{c}_t$$

$$E_t \hat{k}_{t+1} = (1 - \delta) \hat{k}_t + \delta \hat{i}_t$$

### 3. Métodos de resolución de los modelos DSGE

- Para simplificar la notación, vamos a definir los siguientes tres parámetros:

$$\theta = 1 - \beta + \beta\delta$$

$$\phi = 1 - \beta + (1 - \alpha)\beta\delta$$

$$\eta = 1 + \frac{\gamma(1 - \alpha)\theta}{(1 - \gamma)\phi}$$

### 3. Métodos de resolución de los modelos DSGE

- El primer sistema sería por tanto:

$$\hat{y}_t - (1 - \alpha)\hat{l}_t = \hat{a}_t + \alpha\hat{k}_t$$

$$\theta\hat{y}_t - \alpha\beta\delta\hat{i}_t = \phi\beta\delta\hat{c}_t$$

$$\hat{y}_t - \eta\hat{l}_t = \hat{c}_t$$

### 3. Métodos de resolución de los modelos DSGE

- El segundo sistema sería:

$$\theta E_t \hat{k}_{t+1} + E_t \hat{c}_{t+1} - \theta E_t \hat{y}_{t+1} = \hat{c}_t$$

$$E_t \hat{k}_{t+1} = (1 - \delta) \hat{k}_t + \delta \hat{i}_t$$

### 3. Métodos de resolución de los modelos DSGE

- Por tanto, las matrices de constantes vendrían dadas por:

$$A = \begin{bmatrix} 1 & 0 & \alpha - 1 \\ \theta & \phi - \theta & 0 \\ 1 & 0 & -\eta \end{bmatrix}$$

$$B = \begin{bmatrix} \alpha & 0 \\ 0 & \phi \\ 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

### 3. Métodos de resolución de los modelos DSGE

- y por:

$$D = \begin{bmatrix} \theta & 1 \\ 1 & 0 \end{bmatrix}$$

$$F = \begin{bmatrix} -\theta & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 0 \\ 0 & 1 - \delta \end{bmatrix}$$

$$H = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \delta & 0 \end{bmatrix}$$

### 3. Métodos de resolución de los modelos DSGE

- Finalmente, el modelo en el espacio de estados lo cerramos incorporando la expectativa de la desviación de la productividad total de los factores:

$$E_t \hat{a}_{t+j} = \rho_A^j \hat{a}_t$$

- El sistema (1) puede escribirse como:

$$x_t = A^{-1} B s_t + A^{-1} C \hat{a}_t$$

- Tomando un periodo hacia adelante resulta:

$$E_t x_{t+1} = A^{-1} B E_t s_{t+1} + A^{-1} C \rho_A \hat{a}_t$$

- Sustituyendo en el sistema (2), llegamos a:

$$(D + FA^{-1}B)E_t s_{t+1} = (G + HA^{-1}B)s_t + (HA^{-1}C - FA^{-1}C\rho_A)\hat{a}_t$$

### 3. Métodos de resolución de los modelos DSGE

- Resolviendo para las matrices, el sistema final sería:

$$E_t s_{t+1} = J s_t + M \hat{a}_t \quad (3)$$

donde:

$$J = (D + FA^{-1}B)^{-1}(G + HA^{-1}B)$$

$$M = (D + FA^{-1}B)^{-1}(HA^{-1}C - FA^{-1}C\rho_A)$$

### 3. Métodos de resolución de los modelos DSGE

- Usando la descomposición de Jordan, la matriz  $J$  la podemos descomponer como:

$$J = O^{-1}NO$$

donde:

$$N = \begin{bmatrix} N_{11} & 0 \\ 0 & N_{22} \end{bmatrix}$$

y donde:

$$O = \begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix}$$

- Obsérvese que los elementos de la diagonal de  $N$  son los valores propios de la matriz  $J$ . Para que la solución sea única, el valor de  $N_{11}$  tiene que estar dentro del círculo unitario y el valor de  $N_{22}$  fuera del círculo unitario. Las columnas de  $O^{-1}$  son los vectores propios de la matriz  $J$ .

### 3. Métodos de resolución de los modelos DSGE

- Demasiados valores propios mayores a la unidad: Solución inestable.
- Número insuficiente de valores propios mayores a la unidad: Equilibrios múltiples.
- Número de valores propios mayores que la unidad igual al número de variables forward-looking: Unicidad.

# 13. Métodos de resolución de los modelos DSGE

- Por tanto, nuestro sistema (3) podemos escribirlo como:

$$E_t s_{t+1} = \begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix}^{-1} \begin{bmatrix} N_{11} & 0 \\ 0 & N_{22} \end{bmatrix} \begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} s_t + \begin{bmatrix} M_{11} \\ M_{21} \end{bmatrix} \hat{a}_t$$

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} E_t s_{t+1} = \begin{bmatrix} N_{11} & 0 \\ 0 & N_{22} \end{bmatrix} \begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} s_t + \begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix}^{-1} \begin{bmatrix} M_{11} \\ M_{21} \end{bmatrix} \hat{a}_t$$

### 3. Métodos de resolución de los modelos DSGE

- Alternativamente, podemos escribir las siguientes ecuaciones de expectativas:

$$E_t s_{1,t+1} = N_{11} s_{1,t}^1 + Q_{11} \hat{a}_t$$

$$E_t s_{2,t+1} = N_{22} s_{2,t}^1 + Q_{21} \hat{a}_t$$

donde:

$$s_{1,t}^1 = O_{11} \hat{k}_t + O_{12} \hat{c}_t$$

$$s_{2,t}^1 = O_{21} \hat{k}_t + O_{22} \hat{c}_t$$

y donde:

$$Q = \begin{bmatrix} Q_{11} \\ Q_{21} \end{bmatrix} = O^{-1} M$$

### 3. Métodos de resolución de los modelos DSGE

- Dado que el valor de  $N_{22}$  está fuera del círculo unitario, podemos resolver para  $s_{2,t}^1$  hacia adelante:

$$s_{2,t}^1 = \frac{1}{N_{22}} E_t s_{2,t+1} - \frac{Q_{21}}{N_{22}} \hat{a}_t$$

resultando:

$$\begin{aligned} s_{2,t}^1 &= -\frac{Q_{21}}{N_{22}} \sum_{j=0}^{\infty} \left( \frac{1}{N_{22}} \right)^j E_t \hat{a}_{t+j} \\ &= -\frac{Q_{21}}{N_{22}} \sum_{j=0}^{\infty} \left( \frac{\rho_A}{N_{22}} \right)^j \hat{a}_t = \frac{Q_{21}}{\rho_A - N_{22}} \hat{a}_t \end{aligned}$$

Resolviendo para  $\hat{c}_t$  obtenemos:

$$\frac{Q_{21}}{\rho_A - N_{22}} \hat{a}_t = O_{21} \hat{k}_t + O_{22} \hat{c}_t$$

### 3. Métodos de resolución de los modelos DSGE

- A partir de la expresión anterior obtenemos la log-desviación del consumo (la función de política) como:

$$\hat{c}_t = -\frac{O_{21}}{O_{22}}\hat{k}_t + \frac{Q_{21}}{O_{22}(\rho_A - N_{22})}\hat{a}_t$$

o alternativamente:

$$\hat{c}_t = S_1\hat{k}_t + S_2\hat{a}_t$$

siendo

$$S_1 = -\frac{O_{21}}{O_{22}}$$

$$S_2 = \frac{Q_{21}}{O_{22}(\rho_A - N_{22})}$$

### 3. Métodos de resolución de los modelos DSGE

- En el caso del vector  $s_{1,t}^1$  tenemos que:

$$s_{1,t}^1 = (O_{11} + O_{12}S_1)\hat{k}_t + O_{12}S_2\hat{a}_t$$

y sustituyendo,

$$E_t s_{1,t+1}^1 = N_{11}s_{1,t}^1 + Q_{11}\hat{a}_t$$

$$E_t s_{1,t+1}^1 = N_{11} \left[ (O_{11} + O_{12}S_1)\hat{k}_t + O_{12}S_2\hat{a}_t \right] + Q_{11}\hat{a}_t$$

$$(O_{11} + O_{12}S_1)\hat{k}_{t+1} = N_{11}(O_{11} + O_{12}S_1)\hat{k}_t + (Q_{11} + O_{12}S_2(1 - \rho_A))\hat{a}_t$$

### 3. Métodos de resolución de los modelos DSGE

- a partir de la cual obtenemos la función de transición para el stock de capital:

$$\hat{k}_{t+1} = S_3 \hat{k}_t + S_4 \hat{a}_t$$

siendo:

$$S_3 = N_{11}$$

$$S_4 = \frac{Q_{11} + N_{11} O_{12} S_2 - O_{12} S_2 \rho_A}{O_{11} + O_{12} S_1}$$

### 3. Métodos de resolución de los modelos DSGE

Finalmente, volviendo al sistema inicial:

$$x_t = A^{-1}Bs_t + A^{-1}C\hat{a}_t$$

$$x_t = A^{-1}B \begin{bmatrix} \hat{k}_t \\ \hat{c}_t \end{bmatrix} + A^{-1}C\hat{a}_t$$

$$x_t = A^{-1}B \begin{bmatrix} 1 \\ S_1 \end{bmatrix} \hat{k}_t + \left[ A^{-1}C + A^{-1}B \begin{bmatrix} 0 \\ S_2 \end{bmatrix} \right] \hat{a}_t$$

### 3. Métodos de resolución de los modelos DSGE

o:

$$x_t = S_5 s_t + S_6 \hat{a}_t$$

donde:

$$S_5 = A^{-1}B \begin{bmatrix} 1 \\ S_1 \end{bmatrix}$$

$$S_6 = A^{-1}C + A^{-1}B \begin{bmatrix} 0 \\ S_2 \end{bmatrix}$$

### 3. Métodos de resolución de los modelos DSGE

- Una vez realizados los anteriores cálculos, ya tenemos la solución del modelo. Agregando los diferentes componentes, la solución al modelo es:

$$\begin{bmatrix} \widehat{k}_{t+1} \\ \widehat{a}_{t+1} \end{bmatrix} = \begin{bmatrix} S_3 & S_4 \\ 0 & \rho_A \end{bmatrix} \begin{bmatrix} \widehat{k}_t \\ \widehat{a}_t \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{bmatrix}$$

and

$$\begin{bmatrix} \widehat{y}_t \\ \widehat{i}_t \\ \widehat{l}_t \\ \widehat{c}_t \end{bmatrix} = \begin{bmatrix} S_5 & S_6 \\ S_1 & S_2 \end{bmatrix} \begin{bmatrix} \widehat{k}_t \\ \widehat{a}_t \end{bmatrix}$$

esto es, la solución está compuesta por un vector de log-desviaciones de las variables de control en función del vector de variables de estado, donde las matrices  $S_1$ ,  $S_2$ ,  $S_5$  y  $S_6$  son funciones de los parámetros del modelo  $(\alpha, \beta, \gamma, \delta, \rho_A, \sigma_A)$ .

### 3. Métodos de resolución de los modelos DSGE

- Para obtener la solución numérica, se requiere la calibración o estimación de las anteriores matrices compuestas por los parámetros estructurales del modelo.
- Dado el proceso que siguen las variables de estado (que las convierte en variables predeterminadas), podemos predecir el valor futuro de dichas variables de estado y dadas estas futuras variables de estado podemos realizar proyecciones de las variables de control en el futuro.

### 3. Métodos de resolución de los modelos DSGE

- El método propuesto por Klein (2000) es un híbrido entre el de Blanchard y Khan (1980) y el de Sims (2001). Escribiendo el primer sistema de ecuaciones un periodo hacia adelante:

$$Ax_{t+1} = Bs_{t+1} + C\rho\hat{a}_t$$

- Y sustituyendo en el segundo sistema de ecuaciones, llegamos a la misma expresión que en el método Blanchard-Kahn:

$$\begin{aligned} & DE_t s_{t+1} + (FA^{-1}B)E_t s_{t+1} + (FA^{-1}C)\rho\hat{a}_t \\ = & Gs_t + (HA^{-1}B)s_t + (HA^{-1}C + I)\hat{a}_t \end{aligned}$$

### 3. Métodos de resolución de los modelos DSGE

- Agrupando términos,

$$(D + FA^{-1}B)E_t s_{t+1} = (G + HA^{-1}B)s_t + (HA^{-1}C + I - FA^{-1}C\rho)\hat{a}_t$$

$$JE_t s_{t+1} = Ks_t + L\hat{a}_t$$

donde:

$$J = D + FA^{-1}B$$

$$K = G + HA^{-1}B$$

$$L = HA^{-1}C + I - FA^{-1}C\rho$$

### 3. Métodos de resolución de los modelos DSGE

- El método de Klein usa la descomposición generalizada de Schur. La descomposición de Schur de las matrices  $J$  y  $K$  viene dada por by:

$$QJZ = S$$

$$QKZ = T$$

donde  $Q$ ,  $Z$  son matrices unitarias y  $S$ ,  $T$  son matrices triangulares superiores con los elementos de la diagonal siendo los valores propios generalizados de  $J$  y  $K$ :

$$S = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}$$

$$T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}$$

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$

### 3. Métodos de resolución de los modelos DSGE

- Particionamos  $Z$  de forma que:

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$$

$$z_t = Z^H s_t$$

donde  $Z^H$  es la matriz traspuesta conjugada.

### 3. Métodos de resolución de los modelos DSGE

- Dado que:

$$\begin{aligned} J &= Q'SZ^H \\ K &= Q'TZ^H \end{aligned}$$

podemos reescribir nuestro sistema como:

$$\begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} E_t \begin{bmatrix} z_{1,t+1} \\ z_{2,t+1} \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} z_{1,t} \\ z_{2,t} \end{bmatrix} + \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \Xi \hat{a}_t$$

### 3. Métodos de resolución de los modelos DSGE

- El componente inestable del sistema viene dado por:

$$S_{22}E_t z_{2,t+1} = T_{22}z_{2,t} + Q_2 \Xi \hat{a}_t$$

- Iterando hacia adelante en el componente inestable del sistema obtenemos:

$$z_{2,t} = M \hat{a}_t$$

$$\text{vec}(M) = [(\rho^T \otimes S_{22}) - I_z \otimes T_{22}]^{-1} \text{vec}(Q_2 \Xi)$$

$$M = [\rho^T S_{22} - T_{22}]^{-1} Q_2 \Xi$$

- Por tanto:

$$z_{2,t} = [\rho^T S_{22} - T_{22}]^{-1} Q_2 \Xi \hat{a}_t$$

### 3. Métodos de resolución de los modelos DSGE

- El componente estable del sistema sería:

$$S_{11}E_t z_{1,t+1} + S_{12}E_t z_{2,t+1} = T_{11}z_{1,t} + T_{12}z_{2,t} + Q_1 \Xi \hat{a}_t$$

- Ahora usamos la solución del componente inestable para resolver el componente estable:

$$z_{1,t+1} = S_{11}^{-1} T_{11} z_{1,t} + S_{11}^{-1} (T_{12} M - S_{12} M \rho + Q_1 \Xi) \hat{a}_t - Z_{11}^{-1} Z_{12} M \varepsilon_{t+1}$$

- Volviendo a las variables originales del modelo, la solución puede definirse como:

$$\hat{c}_t = Z_{21} Z_{11}^{-1} \hat{k}_t + (Z_{22} - Z_{21} Z_{11}^{-1} Z_{12}) M \hat{a}_t$$

$$\begin{aligned} \hat{k}_{t+1} = & Z_{11} S_{11}^{-1} T_{11} Z_{11}^{-1} \hat{c}_t + (Z_{11} S_{11}^{-1} [T_{12} M - S_{12} M \rho + Q_1 \Xi] \\ & - Z_{11} S_{11}^{-1} T_{11} Z_{11}^{-1} Z_{12} M + Z_{12} M \rho) \hat{a}_t \end{aligned}$$

### 3. Métodos de resolución de los modelos DSGE

- Finalmente, sustituyendo obtenemos la siguiente función de transición para el stock de capital:

$$\hat{k}_{t+1} = N\hat{k}_t + O\hat{a}_t$$

donde:

$$\begin{aligned} N &= Z_{11}S_{11}^{-1}T_{11}Z_{11}^{-1}Z_{21}Z_{11}^{-1} \\ O &= (Z_{11}S_{11}^{-1}[T_{12}M - S_{12}M\rho + Q_1\Xi] \\ &\quad - Z_{11}S_{11}^{-1}T_{11}Z_{11}^{-1}Z_{12}M + Z_{12}M\rho) \\ &\quad + Z_{11}S_{11}^{-1}T_{11}Z_{11}^{-1}(Z_{22} - Z_{21}Z_{11}^{-1}Z_{12})M \end{aligned}$$

### 3. Métodos de resolución de los modelos DSGE

- La función de política del consumo sería:

$$\hat{c}_t = P\hat{k}_{t-1} + Q\hat{a}_{t-1}$$

donde

$$P = Z_{21}Z_{11}^{-1}Z_{11}S_{11}^{-1}T_{11}Z_{11}^{-1}Z_{21}Z_{11}^{-1}$$
$$Q = Z_{21}Z_{11}^{-1} \left[ \begin{array}{l} (Z_{11}S_{11}^{-1}[T_{12}M - S_{12}M\rho + Q_1\Xi] - \\ Z_{11}S_{11}^{-1}T_{11}Z_{11}^{-1}Z_{12}M + Z_{12}M\rho) \\ + Z_{11}S_{11}^{-1}T_{11}Z_{11}^{-1}(Z_{22} - Z_{21}Z_{11}^{-1}Z_{12})M \end{array} \right]$$
$$+(Z_{22} - Z_{21}Z_{11}^{-1}Z_{12})M\rho$$

## 4. Las funciones de política

- Tal y como hemos visto, resolver un modelo DSGE implica obtener las funciones desconocidas de las variables endógenas que satisfacen las condiciones de primer orden. Habitualmente las condiciones de primer orden tienen la siguiente forma general:

$$E_t f(y, x, \Omega) = 0$$

donde  $y$  es un vector de variables endógenas de control y  $x$  un vector de variables de estado (tanto variables exógenas como variables endógenas predeterminadas).

- Supongamos que  $y = g(x, \Omega)$  es la función de política desconocida.
- Previamente hemos tenido de calcular el estado estacionario.

## 4. Las funciones de política

- Lo que realmente hacemos es encontrar los coeficientes de una aproximación lineal a la función  $g(x, \Omega)$ , tal que:

$$g(x, \Omega) = g_0(\Omega) + g_1(\Omega)(x - \bar{x})$$

y

$$E_t(f(g(x, \Omega))) = 0$$

## 4. Las funciones de política

En el modelo que estamos resolviendo, esto implica encontrar los parámetros del siguiente sistema de ecuaciones recursivas de las variables endógenas como desviación respecto a su valor de equilibrio, tal que:

$$\begin{aligned}\hat{y}_t &= g_{yk}\hat{k}_{t-1} + g_{ya}\hat{a}_t \\ \hat{c}_t &= g_{ck}\hat{k}_{t-1} + g_{ca}\hat{a}_t \\ \hat{i}_t &= g_{ik}\hat{k}_{t-1} + g_{ia}\hat{a}_t \\ \hat{l}_t &= g_{lk}\hat{k}_{t-1} + g_{la}\hat{a}_t \\ \hat{k}_t &= g_{kk}\hat{k}_{t-1} + g_{ka}\hat{a}_t \\ \hat{a}_t &= \rho\hat{a}_{t-1} + \varepsilon_t\end{aligned}$$

## 4. Las funciones de política

- En el caso de que usemos una aproximación de segundo orden, las funciones de política son más complejas. Por ejemplo, para el caso del nivel de producción tendríamos:

$$\hat{y}_t = 0.5\Delta^2 + g_{y,k}\hat{k}_t + g_{y,a}\rho_A\hat{a}_t + g_{y,\varepsilon}\varepsilon_t + 0.5g_{y,y}[\hat{y}_{t-1} \otimes \hat{y}_t] + 0.5g_{\varepsilon,\varepsilon}(\varepsilon_t \otimes \varepsilon_t) + g_{y,\varepsilon}[\hat{y}_t \otimes \varepsilon_t]$$

donde  $\Delta^2$  es la matriz de varianza-covarianza de las innovaciones en los shocks.