Description Remarks and examples References Also see

Description

In this entry, we introduce DSGE models and the two commands that estimate the parameters of DSGE models, dsgenl and dsge. For Bayesian estimation, see [BAYES] **bayes: dsge** and [BAYES] **bayes: dsgenl**. We begin with an overview of DSGE models. We then illustrate the process of DSGE modeling by working an example. In this example, we describe a model in its original nonlinear form and estimate its parameters using dsgenl. We also write the model in a corresponding linearized form and estimate the parameters using dsge. We conclude by showing how this example fits into the general DSGE modeling framework.

Remarks and examples

Remarks are presented under the following headings:

Introduction to DSGE models An example: A nonlinear DSGE model Writing down nonlinear DSGEs Data preparation Specifying the model to dsgenl Parameter estimation and interpretation of nonlinear DSGEs An example: A linear DSGE model Writing down linearized DSGEs Specifying the model to dsge Parameter estimation and interpretation of linear DSGEs Postestimation Policy and transition matrices Impulse responses Forecasts Structural and reduced forms of DSGE models

Introduction to DSGE models

DSGE models are models for multiple time series used in macroeconomics and finance. These models are systems of equations that are motivated by economic theory and in which expectations of future values of variables play an important role. Because these models come from theory, the parameters of these models can typically be directly interpreted in terms of the motivating theory. DSGE models are used for macroeconomic policy analysis and forecasting.

In DSGE models, individuals' actions are summarized by decision rules that take the form of nonlinear systems of dynamic equations. These decision rules often come from dynamic stochastic optimization problems. A DSGE model consists of these decision rules, plus any aggregation conditions, resource or budget constraints, and stochastic processes for exogenous variables.

Because the model's equations are the solution to dynamic optimization problems, model equations can feature expectations of future variables. These expectations are endogenous. In DSGE models, expectations of future values of variables correspond to their conditional mean as implied by the model. In other words, individuals' expectations of future values are correct, on average. Such expectations are said to be model-consistent expectations or rational expectations.

There are three kinds of variables in DSGE models: control variables, state variables, and shocks. The terminology is taken from the state-space and optimal control literatures. In DSGE models, the concepts of exogeneity and endogeneity are understood relative to a time period. A state variable is fixed, or exogenous, in a given time period. The system of equations then determines the value of the state variable one period in the future. On the other hand, the system of equations determines the value of a control variable in the current time period. Control variables in a DSGE model can be either observed or unobserved. State variables are always unobserved.

DSGE models can be written in multiple forms. The model may consist of equations that are nonlinear both in the variables and in the parameters. Such a model is said to be nonlinear. A DSGE model is said to be linear, or linearized, when the model equations are linear in the variables. Linear models may still be nonlinear in their parameters.

Nonlinear DSGE models are commonly linearized prior to analysis. In Stata, you have two choices for handling the linearization of a nonlinear model. You can write your model in its nonlinear form and use the dsgenl command. dsgenl will take a linear approximation to the nonlinear model for you. Alternatively, you can derive the linear form of the model and use the dsge command. Although linearizing the equations yourself requires extra effort, there can be advantages. Linearized DSGE models are easier to manipulate than nonlinear ones. For instance, it is easier to include additional lag terms in a linearized model than in a nonlinear one.

After linearization, the DSGE model must be solved prior to estimation. In any analysis of simultaneous equations systems, to solve a model means to write the model's endogenous variables as functions of its exogenous variables. In DSGE models, the analogous concept is to write the model's control variables in terms of its state variables. The model's solution consists of a system of equations relating the control variables to the state variables and a system of equations describing the evolution of the state variables over time. The solution to a DSGE model thus takes the form of a state-space model. The solution to a DSGE model is a crucial object for both estimation and analyses after estimation. Both the likelihood function and the impulse–response functions are formed from the model solution.

dsgenl solves and estimates the parameters of nonlinear DSGE models. It linearizes the model by taking a first-order approximation to the model's equations at the model's steady state, solves the linearized model, and estimates the parameters of the model using the linearized solution.

dsge solves and estimates the parameters of linearized DSGE models.

For Bayesian estimation, see [DSGE] Intro 9. bayes: dsge fits Bayesian linear DSGE models, and bayes: dsgen1 fits Bayesian nonlinear DSGE models.

General introductions to DSGE modeling are available in Ljungqvist and Sargent (2018) and Woodford (2003). Canova (2007) and DeJong and Dave (2011) describe parameter estimation using DSGE models. Fernández-Villaverde, Rubio-Ramírez, and Schorfheide (2016) provide an overview of solution and estimation strategies for nonlinear DSGE models.

An example: A nonlinear DSGE model

Writing down nonlinear DSGEs

Consider the following nonlinear model, similar to that in Woodford (2003, chap. 4). The model consists of equations that describe the behavior of households, firms, and a central bank. Interactions among these actors produce a model of inflation, output growth, and the interest rate. Models of this type are popular in academic and policy settings and are used to describe monetary policy.

Household optimization generates an equation that relates current output Y_t to the expected value of a function of tomorrow's output Y_{t+1} , tomorrow's inflation Π_{t+1} , and the current nominal interest rate R_t ,

$$\frac{1}{Y_t} = \beta E_t \left(\frac{1}{Y_{t+1}} \frac{R_t}{\Pi_{t+1}} \right) \tag{1}$$

where β is a parameter that captures households' willingness to delay consumption.

Optimization by firms generates an equation that links the current deviation of inflation from its steady state, $\Pi_t - \Pi$, to the expected value of the deviation of inflation from its steady state in the future, $E_t (\Pi_{t+1} - \Pi)$, and to the ratio of actual output, Y_t , to the natural level of output, Z_t ,

$$(\Pi_t - \Pi) + \frac{1}{\phi} = \phi\left(\frac{Y_t}{Z_t}\right) + \beta E_t \left(\Pi_{t+1} - \Pi\right)$$
(2)

where ϕ is a parameter linked to the pricing decision of firms. Firms are not affected by inflation per se; they are affected only by deviations of inflation from its steady-state value.

Finally, there is an equation that describes central bank policy. The central bank adjusts the interest rate in response to inflation and, perhaps, to other factors that we leave unmodeled. The equation for the central bank policy is

$$\frac{R_t}{R} = \left(\frac{\Pi_t}{\Pi}\right)^{1/\beta} U_t \tag{3}$$

where R is the steady-state value of the interest rate and U_t is a state variable that captures all movements in the interest rate that are not driven by inflation.

We make a few modifications to the model equations before estimating the parameters. Woodford (2003) rewrites the model in (1)–(3) by defining $X_t = Y_t/Z_t$ as the output gap. We do the same. Substituting in X_t gives us the three-equation system

$$1 = \beta E_t \left(\frac{X_t}{X_{t+1}} \frac{1}{G_t} \frac{R_t}{\Pi_{t+1}} \right)$$
(4a)

$$(\Pi_t - \Pi) + \frac{1}{\phi} = \phi X_t + \beta E_t \left(\Pi_{t+1} - \Pi \right)$$
(5a)

$$\frac{R_t}{R} = \left(\frac{\Pi_t}{\Pi}\right)^{1/\beta} U_t \tag{6a}$$

where $G_t = Z_{t+1}/Z_t$ is a state variable that captures changes in Z_t .

The model is completed by adding equations describing the evolution of the state variables G_t and U_t . We write them as autoregressive processes in logarithms,

$$\ln(G_{t+1}) = \rho_g \ln(G_t) + \xi_{t+1}$$
(7a)

$$\ln(U_{t+1}) = \rho_u \ln(U_t) + e_{t+1} \tag{8a}$$

The variables ξ_{t+1} and ϵ_{t+1} are shocks to the state variables. We now have a complete nonlinear DSGE model. To estimate its parameters, we need to get some data.

Data preparation

There are three requirements of the data used with dsgenl and dsge. First, the data must be tsset so that Stata understands the time structure of your data. Second, the series in your model must have zero mean. This requirement is actually handled for you. Before solving the models, these commands first transform variables into differences from their steady state or, with dsgenl, percent deviations from their steady state. The third requirement is that variables in your data must be weakly stationary.

We have data on the price level and the nominal interest rate in rates.dta. These data were obtained from the Federal Reserve Economic Database (FRED), which contains many macroeconomic and financial time series; see [D] **import fred**. We can type tsset to see that dateq has already been set as the quarterly time variable in this dataset.

```
. use https://www.stata-press.com/data/r19/rates2
(Federal Reserve Economic Data - St. Louis Fed, 2017-02-10)
. tsset
Time variable: dateq, 1947q1 to 2017q1
        Delta: 1 quarter
. describe
Contains data from https://www.stata-press.com/data/r19/rates2.dta
                          281
Observations:
                                                Federal Reserve Economic Data -
                                                  St. Louis Fed, 2017-02-10
    Variables:
                            5
                                                26 Apr 2024 21:22
Variable
              Storage
                         Display
                                    Value
                                                Variable label
                                    label
   name
                 type
                         format
datestr
                str10
                         %-10s
                                                Observation date
                                                Numeric (daily) date
daten
                int
                         %td
gdpdef
                                                GDP deflator GDPDEF
                float
                         %9.0g
                float
                         %9.0g
                                                Federal funds rate FEDFUNDS
r
dateq
                int
                         %tq
                                                Quarterly date
```

```
Sorted by: dateq
```

The dataset includes the price-level variable gdpdef. But the model is written in terms of the inflation rate. For quarterly data, the inflation rate is conventionally obtained as 400 times the difference in log of the price level. Therefore, we begin by generating an inflation rate variable p by using the L. lag operator.

```
. generate p = 400*(ln(gdpdef) - ln(L.gdpdef))
(2 missing values generated)
. label variable p "Inflation rate"
```

We now have inflation and interest rates.

Specifying the model to dsgenl

Equations (4a)-(8a) are the nonlinear DSGE model. The command to estimate the parameters of the system (4a)-(8a) is

. dsgen	l (1 = {beta}*(x/F.x)*(1/g)*(r/F.p))	111
	(1/{phi} + (p-1) = {phi}*x + {beta}*(F.p-1))	111
	$({beta}*r = p^{(1/{beta})*u})$	111
	$(\ln(F.u) = {rhou}*\ln(u))$	111
	$(\ln(F.g) = {rhog}*\ln(g)),$	111
	exostate(u, g) observed(p, r) unobserved(x)	

Each equation is bound in parentheses. The equations in the command look similar to the equations we wrote down. The equations can be written in nearly any form that is logically equivalent to the model equations. In addition to providing the equations, you must specify which variable plays which role in the model through the options. The option exostate() lists all exogenous state variables, those that are subject to shocks. The option observed() lists observed control variables. The option unobserved() lists unobserved (latent) control variables. A fourth option, endostate(), is available for endogenous state variables, those that are not subject to shocks. Each variable must be listed in one and only one of the options. The number of exogenous state variables must be the same as the number of observed control variables.

Parameters are bound in curly braces; this is how Stata distinguishes between parameters and variables. Parameters can appear nonlinearly, they can appear more than once, and they can appear in multiple equations. A parameter repeated across equations is considered the same parameter; for example, in the above model, beta appears in each of the first three equations, and it is assumed that there is only one parameter beta to be estimated. These cross-equation restrictions are common in DSGE models.

The operator F. can be applied to both state and control variables to denote the expected future value of a variable. For example, we typed F. x to include x_{t+1} in the model. However, unlike F. used in other time-series applications, the value of x_{t+1} is not determined from the value of x at time t + 1 in your data. Instead, it is an expectation of the value of x at time t + 1 based on your model. In fact, the entire model can be written in terms of expectations.

Notice that in (4a) and (5a), the E_t operator appears. This is the statistical expectation operator. It refers to expected value of the term in brackets conditional on information known at time t. In dsgenl, this expectation operator is assumed in front of each equation. Hence, the dsgenl equation

is interpreted as

$$E_t\left(1-\beta\frac{X_t}{X_{t+1}}\frac{1}{G_t}\frac{R_t}{\Pi_{t+1}}\right)=0$$

which is equivalent to what we wrote in (4a).

Parameter estimation and interpretation of nonlinear DSGEs

The model's parameters are estimated by taking a linear approximation to the model's equations, solving the linearized system for its state-space form, and estimating the parameters of the state-space model by maximum likelihood.

. dsgenl (1 =	{beta}*(x/F.x	:)*(1/g)*(r/	F.p))							
> (1/{]	$(1/{phi} + (p-1) = {phi}*x + {beta}*(F.p-1))$									
$({beta}*r = p^{(1/{beta})*u})$										
> (ln(H	$(ln(F.u) = {rhou}*ln(u))$									
> (ln(H	F.g) = {rhog}*	ln(g)),								
> exost	tate(u g) obse	rved(p r) u	nobserved	d(x) nolog	5					
Solving at ini	itial paramete	r vector								
Checking ident	ification									
First-order DS	SGE model									
Sample: 1954q3 Log likelihood	8 thru 2016q4 1 = -768.09383	ł			Number of	obs = 250				
	Coefficient	Std. err.	Z	P> z	[95% conf.	interval]				
/structural										
beta	.5112917	.075773	6.75	0.000	.3627794	.659804				
phi	5.895233	1.652618	3.57	0.000	2.656161	9.134304				
rhou	.6989186	.0449146	15.56	0.000	.6108877	.7869496				
rhog	.9556408	.0181346	52.70	0.000	.9200975	.991184				
sd(e.u)	2.317573	.2987328			1.732068	2.903079				
sd(e.g)	.6147308	.0973072			.4240122	.8054493				
. 0.										

We see header information describing the sample date range, the number of observations, and the log likelihood.

In the estimation table, we see estimates of model parameters. Some of these parameters have a structural interpretation. Parameter beta plays two roles in the model: it is the discount factor in the household and firm equations, and its inverse captures the degree to which the central bank responds to inflation in the interest rate equation. Its point estimate is about one half, so a point estimate for its inverse is about 2.

We use nlcom to obtain an estimate of the parameter $1/\beta$ directly.

. nlcom 1/_b[1 _nl_1:	beta] 1/_b[beta]					
	Coefficient	Std. err.	z	P> z	[95% conf.	interval]
nl_1	1.955831	.2898524	6.75	0.000	1.387731	2.523931

Values for this parameter in the literature are typically around 1.5. Our estimated value is about 2, but the common value of 1.5 lies within the confidence interval.

The phi parameter is a measure of price adjustment. The remaining four parameters describe the evolution of the model's exogenous state variables. Both state variables are highly persistent, with persistence parameters of 0.70 and 0.96 on u and g, respectively. The estimated standard deviations of the shocks are also displayed. The shock to the state variable u is denoted e.u and has standard deviation 2.32. The shock to the state variable g is denoted e.g and has standard deviation 0.61.

An example: A linear DSGE model

Writing down linearized DSGEs

The model in (4a)-(8a) is nonlinear. To use dsge, we first write the model in its corresponding linearized form, which we show below. Throughout this manual, we use lowercase letters to denote deviations of variables from the steady state. The linearized versions of (4a)-(8a) are

$$x_t = E_t x_{t+1} - (r_t - E_t \pi_{t+1} - g_t)$$
(4b)

$$\pi_t = \beta E_t \pi_{t+1} + \kappa x_t \tag{5b}$$

$$r_t = \frac{1}{\beta}\pi_t + u_t \tag{6b}$$

$$u_{t+1} = \rho_u u_t + \epsilon_{t+1} \tag{7b}$$

$$g_{t+1} = \rho_q g_t + \xi_{t+1} \tag{8b}$$

The new parameter κ is a function of the underlying parameter ϕ from (5a).

We have implicitly constrained some of the coefficients; for example, the coefficient on the interest rate in the output equation is constrained to -1.

Specifying the model to dsge

The dsge command to estimate the parameters in the system of (4b)–(8b) is

•	dsge	(p	=	{beta}*F.p + {kappa}*x)	111
		(x	=	F.x - (r - F.p - g), unobserved)	
		(r	=	(1/{beta})*p + u)	
		(F.u	=	{rhou}*u, state)	///
		(F.g	=	{rhog}*g, state)	

As with dsgen1, each equation is bound in parentheses. The equations look almost identical to the system in (4b)–(8b). Because a model has as many variables as it has equations, each variable will appear on the left-hand side of one and only one equation but can appear on the right-hand side of as many equations as we like. Equations may be specified in any order.

We identify types of variables differently with dsge than we did with dsgen1. Equation options that appear within the parentheses indicate which type of variable appears on the left-hand side of the equation. Equations specified without any options are equations for observed control variables. We add the unobserved option to specify that a left-hand-side variable is an unobserved control variable. We use the state option to identify state variables.

In our model, p and r are the observed control variables p_t and r_t , and their equations are specified without options. The output gap, x_t , is an unobserved control variable, so we specify the equation for x using the unobserved option.

u and g are the state variables u_t and g_t , and option state is included when we type their equations. Recall that state variables are fixed in the current period, so we specify how they evolve through time by modeling the one-period lead—hence, the F. on the left-hand side of each state equation. The state equations specify how the state variable evolves as a function of the current state variables and, possibly, the control variables.

The shocks ϵ_t and ξ_t enter the system through the state equations of their corresponding variable. By default, a shock is attached to each state equation. The number of shocks must be the same as the number of observed control variables in the model.

When we typed

. dsge ... (F.u = {rhou}*u, state) ...

the underlying equation is what we wrote in (7b). If a state variable is treated as deterministic in your model, then it will not have a shock. For example, capital accumulation is often treated as deterministic. To include an equation for a state variable without a shock, we would include the noshock option within the equation.

As with dsgen1, the operator F. can be applied to both state and control variables to denote the expected future value of a variable. For example, we can type F.x to specify x_{t+1} . Unlike F. used in other time-series applications, the value of x_{t+1} is not taken from the value of x at time t + 1 in your data. Instead, it is an expectation of the value of x at time t + 1 based on your model. dsge interprets equations with F. operators in the following manner. The equation

. dsge (p = {beta}*F.p + {kappa}*x) ...

is interpreted as

 $E_t \left(p_t - \beta p_{t+1} - \kappa x_t \right) = 0$

which is equivalent to (4b) above. In a linear DSGE model, it is immaterial whether the expectation operator is thought of as applying to a variable, or to an equation, or indeed to the entire system of equations jointly.

The parameters we want to estimate are bound in curly braces.

For more details on the dsge syntax, see [DSGE] Intro 2.

Parameter estimation and interpretation of linear DSGEs

We estimate the parameters of the model in (4b)–(8b). These equations are much discussed in the monetary economics literature. Equation (4b) is known as the output-gap Euler equation. Equation (5b) is known as a New Keynesian Phillips Curve, and the parameter κ is known as the slope of the Phillips curve. In New Keynesian models, prices depend on output, and κ is a measure of that dependence. Equation (6a) is known as a Taylor rule, after Taylor (1993). The coefficient on inflation in a Taylor rule is a commonly discussed parameter. β has two roles in the model above. It relates current inflation deviations to expected future inflation deviations, and it relates interest rate deviations to inflation deviations.

```
. dsge (p = {beta}*F.p + {kappa}*x)
       (x = F.x - (r - F.p - g), unobserved)
>
>
       (r = (1/{beta})*p + u)
>
       (F.u = {rhou}*u, state)
       (F.g = {rhog}*g, state)
(setting technique to bfgs)
Iteration 0: Log likelihood = -13931.564
Iteration 1: Log likelihood = -1301.5118
                                            (backed up)
Iteration 2: Log likelihood = -1039.6984
                                           (backed up)
Iteration 3: Log likelihood = -905.70867
                                            (backed up)
Iteration 4: Log likelihood = -842.76867
                                           (backed up)
(switching technique to nr)
Iteration 5: Log likelihood = -812.04209
                                            (backed up)
Iteration 6: Log likelihood = -787.29942
Iteration 7:
             Log likelihood = -777.50219
Iteration 8: Log likelihood = -768.84849
Iteration 9: Log likelihood = -768.13605
Iteration 10: Log likelihood = -768.0951
Iteration 11: Log likelihood = -768.09383
Iteration 12: Log likelihood = -768.09383
DSGE model
Sample: 1954q3 thru 2016q4
                                                            Number of obs = 250
Log likelihood = -768.09383
               Coefficient Std. err.
                                                P>|z|
                                                           [95% conf. interval]
                                           z
/structural
                 .5112878
                            .0757909
                                         6.75
                                                0.000
                                                           .3627403
                                                                       .6598353
        beta
       kappa
                 .1696301
                            .0475493
                                         3.57
                                                0.000
                                                           .0764353
                                                                       .2628249
                                        15.56
                                                0.000
                                                           .6108785
       rhou
                 .6989185
                            .0449192
                                                                       .7869585
       rhog
                 .9556407
                            .0181342
                                        52.70
                                                0.000
                                                           .9200984
                                                                       .9911831
                            .2988025
      sd(e.u)
                  2.31759
                                                           1.731948
                                                                       2.903232
```

Two of the parameters have structural interpretations. The parameter kappa is the slope of the Phillips curve. Theory predicts that this parameter will be positive, and indeed our estimate is positive.

.4239761

.8054938

The parameter beta is the inverse of the coefficient on inflation in the interest rate equation. We can obtain an estimate of $1/\beta$, which is interpreted as the degree to which the central bank responds to movements in inflation, by using nlcom.

. nlcom 1/_b[beta]							
_nl_1:	_nl_1: 1/_b[beta]						
	Coefficient	Std. err.	z	P> z	[95% conf.	interval]	
_nl_1	1.955846	.2899255	6.75	0.000	1.387602	2.524089	

A typical value for $1/\beta$ in the literature is 1.5. Our point estimate is around 2.

.0973277

Postestimation

sd(e.g)

.614735

Several tools are available to help you analyze your model after you have estimated its parameters. These tools are available after using either dsgenl or dsge. Below, we continue with the model we fit using dsge. The results would be the same if you used these commands after dsgenl.

Policy and transition matrices

The matrix of parameters in the state-space form that specifies how the state variables affect the control variables is known as the policy matrix. Each policy matrix parameter is the effect of a one-unit shock to a state variable on a control variable.

We use estat policy to view these results.

```
. estat policy
Policy matrix
```

		C						
		D	Delta-method					
		Coefficient	std. err.	z	P> z	[95% conf	. interval]	
р								
	u	4170858	.0389324	-10.71	0.000	4933918	3407798	
	g	.8818832	.2330568	3.78	0.000	.4251002	1.338666	
x								
	u	-1.580149	.3926325	-4.02	0.000	-2.349694	8106032	
	g	2.658658	.9045254	2.94	0.003	.8858213	4.431496	
r								
	u	.1842446	.0567979	3.24	0.001	.0729228	.2955665	
	g	1.724827	.2210255	7.80	0.000	1.291625	2.158029	

Results are listed equation by equation. The first block is the policy equation for inflation p and writes it as a function of the state variables alone. A unit shock to the state u reduces inflation by an estimated 0.417, and a unit shock to g raises inflation by an estimated 0.882.

The matrix of parameters that specifies the dynamic process for the state variables is known as the state transition matrix. The state transition equation relates the future values of the state variables to their values in the current period. Each state transition matrix parameter is the effect of a one-unit shock to a state variable on its one-period-ahead mean.

Transition matrix of state variables Delta-method Coefficient std. err. z P> z [95% conf. interval F.u u .6989185 .0449192 15.56 0.000 .6108785 .786958 g 1.11e-16 F.g u 0 (omitted) . .9556407 .0181342 52.70 0.000 .9200984 .991183	. estat	transi	ition					
Delta-method Coefficient std. err. z P> z [95% conf. interval F.u u .6989185 .0449192 15.56 0.000 .6108785 .786958 g 1.11e-16 F.g u 0 (omitted) .9556407 .0181342 52.70 0.000 .9200984 .991183	Transit	ion mat	trix of state	variables				
F.u g 1.11e-16			I Coefficient)elta-method std. err.	Z	P> z	[95% conf.	interval]
u .6989185 .0449192 15.56 0.000 .6108785 .786958 g 1.11e-16 F.g u 0 (omitted) .95556407 .0181342 52.70 0.000 .9200984 .991183	F.u							
g 1.11e-16		u	.6989185	.0449192	15.56	0.000	.6108785	.7869585
F.g u 0 (omitted) g .9556407 .0181342 52.70 0.000 .9200984 .991183		g	1.11e-16	•	•	•	•	•
u 0 (omitted) g .9556407 .0181342 52.70 0.000 .9200984 .991183	F.g							
g .9556407 .0181342 52.70 0.000 .9200984 .991183		u	0	(omitted)				
		g	.9556407	.0181342	52.70	0.000	.9200984	.9911831

Note: Standard errors reported as missing for constrained transition matrix values.

Both state variables are modeled as autoregressive processes, so the results in estat transition repeat the estimates of rhou and rhog from the dsge output. In this case, the other entries in the state transition matrix are 0 or differ from 0 only because of lack of numerical precision. In more complicated models, such as those in which a state equation depends on a control variable, the state transition matrix will contain new information about that state variable.

Impulse responses

The state-space form allows us to trace the path of a control or state in response to a shock to a state. This path is known as an impulse-response function (IRF). irf after dsge or dsgenl estimates IRFs, and it puts the named set of estimates into an .irf file, whose results can be displayed using irf graph or irf table.

To graph the IRF, we first create the nkirf.irf file and set it as the active .irf file using the irf set command.

```
. irf set nkirf.irf
(file nkirf.irf created)
(file nkirf.irf now active)
```

Next, we use irf create to estimate a complete set of impulse responses based on our dsge command. A complete set of impulse responses is an impulse to each shock and the response of each state and control variable to that impulse. For the model in this example, irf create generates an impulse to e.u and e.g, then stores the response to that impulse on p, x, r, g, and u. The results are stored in the nkirf.irf file.

```
. irf create model1
(file nkirf.irf updated)
```

We now use irf graph to plot the impulse responses. The impulse() and response() options control which impulse and which responses are chosen. To view the response of p, x, r, and u to a shock to u, we type

```
. irf graph irf, impulse(u) response(x p r u) byopts(yrescale)
```



Graphs by irfname, impulse variable, and response variable

The state variable u models movements in the interest rate that occur for reasons other than the feedback between inflation and the interest rate. A shock to u is effectively a surprise increase in the interest rate, and the IRF traces out how this shock causes temporary decreases to inflation (top-left graph) and to the output gap (bottom-right graph).

Forecasts

The forecast suite of commands produces dynamic forecasts from the fitted model.

We first store the dsge estimation results.

. estimates store dsge_est

We use tsappend() to extend the dataset by 3 years, or 12 quarters.

. tsappend, add(12)

To set up a forecast, we perform three steps. First, we type forecast create to initialize a new forecasting model, which we name dsgemodel.

```
. forecast create dsgemodel
Forecast model dsgemodel started.
```

Next, we add the estimates from dsge to the forecasting model using forecast estimates.

```
. forecast estimates dsge_est
Added estimation results from dsge.
Forecast model dsgemodel now contains 2 endogenous variables.
```

This command adds the estimates stored in dsge_est to the model dsgemodel. Finally, we produce dynamic forecasts beginning in the first quarter of 2017 using forecast solve. The prefix(d1_) option specifies the d1_ prefix that will be given to the variables created by forecast. We also request that dynamic forecasts begin in the first quarter of 2017 with the begin(tq(2017q1)) option.

```
. forecast solve, prefix(d1_) begin(tq(2017q1))
Computing dynamic forecasts for model dsgemodel.
Starting period:
                       2017a1
                       2020a1
Ending period:
Forecast prefix: d1_
2017q1: .....
2017q2:
          . . . . . . . . . . . . .
2017q3:
          . . . . . . . . . . . . . .
2017q4:
           . . . . . . . . . . . . . . .
2018q1:
           . . . . . . . . . . . . . .
2018q2:
           . . . . . . . . . . . . .
2018q3:
            . . . . . . . . . . . . .
2018q4:
            . . . . . . . . . . . . .
2019q1:
            . . . . . . . . . . . . .
2019q2:
            . . . . . . . . . .
2019q3:
           . . . . . . . . . . . . .
2019q4:
           . . . . . . . . . . . . .
2020q1:
           . . . . . . . . . . . . .
Forecast 2 variables spanning 13 periods.
```

The dynamic forecast begins in the first quarter of 2017, so all forecasts are out of sample.

We can graph the forecast for inflation d1_p using tsline.

. tsline d1_p if tin(2010q1, 2021q1), tline(2017q1)

The model forecasts that inflation will smoothly return to its long-run value, the sample mean.

We can also begin the forecast during a time period for which observations are available. Specifying the begin(tq(2014q1)) option produces dynamic forecasts beginning in the first quarter of 2014, so we can compare the forecast for 2014–2016 with the actual observations over that period.

. forecast solve, prefix(d2_) begin(tq(2014q1)) Computing dynamic forecasts for model dsgemodel. Starting period: 2014q1 Ending period: 2020q1 Forecast prefix: d2_ 2014q1: 2014q2: 2014q3: . . . 2014q4: . . . 2015q1: 2015q2: 2015q3: 2015q4: 2016q1: 2016q2: 2016q3: 2016q4: 2017q1: 2017q2: 2017q3: 2017q4: 2018q1: 2018q2: 2018q3: 2018q4: 2019q1: 2019q2: 2019q3: 2019q4: 2020q1: Forecast 2 variables spanning 25 periods.



We then plot both the observed inflation and the forecast.

The forecast captures the general upward trend in inflation from 2014–2016, but it does not predict the variation in inflation around the upward trend.

Structural and reduced forms of DSGE models

Now that we have worked an example, we show how it fits in the more general formulation of DSGE models. The model in (4a)–(8a) is an example of a nonlinear DSGE model. In general, a nonlinear DSGE model can be expressed as

$$E_t \left\{ \mathbf{f}(\mathbf{x}_{t+1}, \mathbf{y}_{t+1}, \mathbf{x}_t, \mathbf{y}_t, \boldsymbol{\theta}) \right\} = \mathbf{0}$$
(9)

where **f** is a vector of equations, \mathbf{x}_t is a vector of state variables, \mathbf{y}_t is a vector of control variables, and $\boldsymbol{\theta}$ denotes the vector of structural parameters. To solve a DSGE model means to write it in state-space form. In general, the state-space form of a nonlinear DSGE model is nonlinear. dsgen1 takes a linear approximation to the model equations to solve for a linear approximation to the state-space form.

The model in (4b)–(8b) is an example of a linearized DSGE model. In general, a linearized DSGE model can be expressed as

$$\mathbf{A}_0 \mathbf{y}_t = \mathbf{A}_1 E_t (\mathbf{y}_{t+1}) + \mathbf{A}_2 \mathbf{y}_t + \mathbf{A}_3 \mathbf{x}_t$$
(10)

$$\mathbf{B}_{0}\mathbf{x}_{t+1} = \mathbf{B}_{1}E_{t}(\mathbf{y}_{t+1}) + \mathbf{B}_{2}\mathbf{y}_{t} + \mathbf{B}_{3}\mathbf{x}_{t} + \mathbf{C}\boldsymbol{\epsilon}_{t+1}$$
(11)

where as before \mathbf{y}_t is a vector of control variables, \mathbf{x}_t is a vector of state variables, and $\boldsymbol{\epsilon}_t$ is a vector of shocks. \mathbf{A}_0 through \mathbf{A}_3 and \mathbf{B}_0 through \mathbf{B}_3 are matrices of parameters. We require that \mathbf{A}_0 and \mathbf{B}_0 be diagonal matrices. The entries in all of these matrices are functions of the structural parameters, which we denote by vector $\boldsymbol{\theta}$. Economic theory places restrictions on the entries in the matrices. \mathbf{C} is a selection matrix that determines which state variables are subject to shocks.

The reduced form of a DSGE model expresses the control variables as functions of the state variables alone and specifies how state variables evolve over time. The reduced form is a state-space model. The state-space form of the model is given by

$$\mathbf{y}_t = \mathbf{G}\mathbf{x}_t \tag{12}$$

$$\mathbf{x}_{t+1} = \mathbf{H}\mathbf{x}_t + \mathbf{M}\boldsymbol{\epsilon}_{t+1} \tag{13}$$

where \mathbf{y}_t is a vector of control variables, \mathbf{x}_t is a vector of state variables, and $\boldsymbol{\epsilon}_t$ is a vector of shocks. **G** is the policy matrix, and **H** is the state transition matrix. **M** is diagonal and contains the standard deviations of the shocks.

 \mathbf{y}_t is partitioned into observed and unobserved controls, $\mathbf{y}_t = (\mathbf{y}_{1,t}, \mathbf{y}_{2,t})$. The observed control variables are related to the control variables by the equation

$$\mathbf{y}_{1,t} = \mathbf{D}\mathbf{y}_t$$

where \mathbf{D} is a selection matrix. Only observed control variables play a role in estimation. The number of observed control variables must be the same as the number of state equations that include shocks.

You specify a model of the form (9) to dsgenl and a model of the form (10)-(11) to dsge. Many models require some manipulation to fit into the structure in (9) and (10); see [DSGE] Intro 4 for details. Postestimation commands estat policy and estat transition will display the policy and transition matrices in (12) and (13), respectively.

References

Canova, F. 2007. Methods for Applied Macroeconomic Research. Princeton, NJ: Princeton University Press.

- DeJong, D. N., and C. Dave. 2011. Structural Macroeconometrics. 2nd ed. Princeton, NJ: Princeton University Press. https://doi.org/10.2307/j.ctt7srm7.
- Fernández-Villaverde, J., J. F. Rubio-Ramírez, and F. Schorfheide. 2016. "Solution and estimation methods for DSGE models". In *Handbook of Macroeconomics*, edited by J. B. Taylor and H. Uhlig, vol. 2A: 527–724. Amsterdam: North-Holland. https://doi.org/10.1016/bs.hesmac.2016.03.006.
- Ljungqvist, L., and T. J. Sargent. 2018. Recursive Macroeconomic Theory. 4th ed. Cambridge, MA: MIT Press.
- Taylor, J. B. 1993. Discretion versus policy rules in practice. Carnegie-Rochester Conference Series on Public Policy 39: 195–214. https://doi.org/10.1016/0167-2231(93)90009-L.
- Woodford, M. 2003. Interest and Prices: Foundations of a Theory of Monetary Policy. Princeton, NJ: Princeton University Press.

Also see

- [DSGE] dsge Linear dynamic stochastic general equilibrium models
- [DSGE] dsge postestimation Postestimation tools for dsge
- [DSGE] dsgenl Nonlinear dynamic stochastic general equilibrium models
- [DSGE] dsgenl postestimation Postestimation tools for dsgenl
- [DSGE] Intro 9 Bayesian estimation
- [TS] forecast Econometric model forecasting
- [TS] irf Create and analyze IRFs, dynamic-multiplier functions, and FEVDs
- [TS] sspace State-space models

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on citing Stata documentation.