Title

stata.com

dsgenl — Nonlinear dynamic stochastic general equilibrium models

Description

`dsgenl` fits nonlinear dynamic stochastic general equilibrium (DSGE) models to multiple time series via perturbation methods. DSGE models are systems of equations that are motivated by economic theory. In these systems, expectations of future values of variables can affect the current values. The parameters of these models are often directly interpretable in terms of economic theory.

Menu

Statistics > Multivariate time series > Dynamic stochastic general equilibrium (DSGE) models > Nonlinear DSGE models

Syntax

```
dsgenl (lexp_1 = rexp_1) [(lexp_2 = rexp_2) ...] [if] [in],
    observed(varlist) exostate(namelist) [options]
```

rexp_j and *lexp_j* are substitutable expressions on the right-hand side and left-hand side of equation *j*. These are Stata expressions that include scalars, variables, and parameters to be estimated. The variables may be state variables, observed control variables, and unobserved control variables. The lead operator, `F.`, can be applied to a variable so that `F.`*varname* refers to the expected value of *varname* in the next time period. No other time-series operators are allowed; see [DSGE] **Intro 4** for information on including additional lags and leads. Parameters to be estimated are enclosed in curly braces; `{beta}` is an example of a parameter. Initial values for parameters are given by including an equal sign and the initial value inside the braces; for example, `{beta=2}` sets the initial value for `beta` to 2. See *Specifying the system of nonlinear equations* in [DSGE] **Intro 2**.

options	Description
Main	
* observed(*varlist*)	list of observed control variables
unobserved(*namelist*)	list of unobserved control variables
* exostate(*namelist*)	list of exogenous state variables
endostate(*namelist*)	list of endogenous state variables
constraints(*constraints*)	apply specified linear constraints
linearapprox	take a linear, rather than log-linear, approximation
noidencheck	do not check for parameter identification
solve	return model solution at initial values
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim or robust
Reporting	
level(#)	set confidence level; default is level(95)
nocnsreport	do not display constraints
display_options	control columns and column formats and line width
Maximization	
maximize_options	control the maximization process
Advanced	
nodemean	do not demean data prior to estimation
post	force posting of estimation results in the event of errors caused by lack of identification or stability
idtolerance(#)	set tolerance used for identification check; seldom used
steadytolerance(#)	set tolerance used for convergence in steady-state calculations; seldom used
steadyinit(*matrix*)	set initial values for steady state; seldom used
coeflegend	display legend instead of statistics

* observed() and exostate() are required.

You must tsset your data before using dsgenl; see [TS] **tsset**.

bayes and collect are allowed; see [U] **11.1.10 Prefix commands**. For more details, see [BAYES] **bayes: dsgenl**.

coeflegend does not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Main

observed(*varlist*) specifies which variables that appear in the model equations are observed control variables. observed() is required.

unobserved(*namelist*) specifies which variables that appear in the model equations are unobserved control variables. The names must be valid Stata names but need not exist in your dataset.

exostate(*namelist*) specifies which variables that appear in the model equations are exogenous state variables. Exogenous state variables are subject to shocks. The names must be valid Stata names but need not exist in your dataset. There must be the same number of exogenous state variables as there are observed control variables in your model. exostate() is required.

`endostate(`*namelist*`)` specifies which variables that appear in the model equations are endogenous state variables. Endogenous state variables are not subject to shocks. The names must be valid Stata names but need not exist in your dataset.

`constraints(`*constraints*`)`; see [R] **Estimation options**.

`linearapprox` specifies that a linear approximation be applied to the DSGE model rather than a log-linear approximation. In either case, an approximation is applied. In a log-linear approximation, variables are interpreted as percentage deviations from steady state. In a linear approximation, variables are interpreted as unit deviations from steady state.

`noidencheck` skips the check that the parameters are identified at the initial values. Models that are not structurally identified can still converge, thereby producing meaningless results that only appear to have meaning; thus, care should be taken in specifying this option. See [DSGE] **Intro 6** for details.

`solve` puts the model into state-space form at the initial parameter values. No standard errors are produced.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`) and that are robust to some kinds of misspecification (`robust`); see [R] ***vce_option***.

Reporting

`level(#)`, `nocnsreport`; see [R] **Estimation options**.

display_options: `noci`, `nopvalues`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`; see [R] **Estimation options**.

Maximization

maximize_options: `difficult`, `technique(`*algorithm_spec*`)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, and `from(`*init_specs*`)`; see [R] **Maximize**.

Advanced

`nodemean` prevents `dsgenl` from removing the mean of the observed control variables prior to estimation.

`post` causes `dsgenl` to post the parameter vector into `e()`, even in the event of errors that arise from checking stability conditions or identification.

`idtolerance(#)` specifies the tolerance used in the identification diagnostic. The default is `idtolerance(1e-6)`.

`steadytolerance(#)` specifies the tolerance used in calculations of the model's steady state. The default is `steadytolerance(1e-17)`.

`steadyinit(`*matrix*`)` specifies a vector of initial values for use in finding the steady state.

The following option is available with `dsgenl` but is not shown in the dialog box:

`coeflegend`; see [R] **Estimation options**.

Remarks

For an introduction to what dsgenl can do and how to use it, see [DSGE] **Intro 1**. It is highly recommended that you read the introduction first.

For examples of dsgenl, see the examples of classic DSGE models in [DSGE] **Intro 3d**, [DSGE] **Intro 3e**, and [DSGE] **Intro 3f**.

Stored results

dsgenl stores the following in e():

Scalars

e(N)	number of observations
e(k)	number of parameters
e(k_eq)	number of equations in e(b)
e(k_dv)	number of dependent variables
e(k_state)	number of state equations
e(k_control)	number of control equations
e(k_shock)	number of shocks
e(k_observed)	number of observed control equations
e(k_stable)	number of stable eigenvalues
e(ll)	log likelihood
e(tmin)	minimum time in sample
e(tmax)	maximum time in sample
e(rank)	rank of VCE
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	dsgenl
e(cmdline)	command as typed
e(depvar)	unoperated names of dependent variables
e(state)	unoperated names of state variables
e(shock)	unoperated names of state variables subject to shocks
e(control)	unoperated names of control variables
e(observed)	unoperated names of observed control variables
e(title)	title in estimation output
e(tvar)	variable denoting time within groups
e(tmins)	formatted minimum time
e(tmaxs)	formatted maximum time
e(tsfmt)	format for the current time variable
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(opt)	type of optimization
e(method)	likelihood method
e(idencheck)	passed, failed, or skipped
e(technique)	maximization technique
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginsnotok)	predictions disallowed by margins

Matrices

e(b)	parameter vector
e(Cns)	constraints matrix
e(ilog)	iteration log (up to 20 iterations)
e(eigenvalues)	generalized eigenvalues
e(gradient)	gradient vector
e(shock_coeff)	estimated shock coefficient matrix
e(transition)	estimated state transition matrix
e(policy)	estimated policy matrix

`e(steady)`	estimated steady-state vector
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

A DSGE model is a system of equations in which the values of the control variables $\mathbf{y}_t$ and of the state variables $\mathbf{x}_t$ depend on their values in previous periods and their one-period-ahead rational expectations; see [DSGE] **Intro 1** for an introduction to control variables and state variables. The structural form of a nonlinear DSGE model is

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

where $\mathbf{f}$ is an $n\times 1$ vector of functions, the n_y variables $\mathbf{y}_t$ are control variables, and the $n_x\times 1$ variables $\mathbf{x}_t$ are state variables. The parameter $\boldsymbol{\theta}$ contains the model's structural parameters.

Fernández-Villaverde, Rubio-Ramírez, and Schorfheide (2016) contain an introduction to the solution and estimation of such models. `dsgenl` proceeds by linearizing (1) around the steady state and then solving the resulting linear system.

The steady state is found by solving the deterministic system of equations,

$$\mathbf{f}(\mathbf{x},\mathbf{y},\mathbf{x},\mathbf{y};\boldsymbol{\theta})=\mathbf{0}$$

which is n equations in n unknowns. The location of the steady state is a vector $(\overline{\mathbf{x}},\overline{\mathbf{y}})$, each element of which may depend on the structural parameters.

Once the steady state has been found, the vector of functions $\mathbf{f}$ is linearized around that point, yielding

$$E_t\left\{\mathbf{F}_1\mathbf{x}_{t+1}+\mathbf{F}_2\mathbf{y}_{t+1}+\mathbf{F}_3\mathbf{x}_t+\mathbf{F}_4\mathbf{y}_t\right\}=\mathbf{0} \tag{2}$$

Equation (2) is in the form of a linearized DSGE model, and the remaining solution techniques are the same as for `dsge`; see *Methods and formulas* in [DSGE] **dsge**. Parameter estimation proceeds on the basis of the linearized equation shown as (2).

Reference

Fernández-Villaverde, J., J. F. Rubio-Ramírez, and F. Schorfheide. 2016. Solution and estimation methods for DSGE models. In Vol. 2A of *Handbook of Macroeconomics*, ed. J. B. Taylor and H. Uhlig, chap. 9, 527–724. Amsterdam: North-Holland. https://doi.org/10.1016/bs.hesmac.2016.03.006.

Also see

[DSGE] **dsgenl postestimation** — Postestimation tools for dsgenl

[DSGE] **Intro 2** — Learning the syntax

[BAYES] **bayes: dsgenl** — Bayesian nonlinear dynamic stochastic general equilibrium models

[TS] **sspace** — State-space models

[TS] **tsset** — Declare data to be time-series data

[TS] **var** — Vector autoregressive models

[U] **20 Estimation and postestimation commands**