Title

stata.com

intro 5 — Stability conditions

Description

We can estimate only the parameters of models that we can solve, and we can solve only models that satisfy a stability condition. This entry discusses this condition, how it affects estimation in practice, and how to find initial values that satisfy it.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Why we care about stability
What if the initial values are not saddle-path stable?

Why we care about stability

DSGE models are dynamic systems subject to random shocks. DSGE models that do not spiral out of control or converge to a single point when shocked are said to be "saddle-path stable". We can solve only saddle-path stable DSGE models, and we can estimate only the parameters of models we can solve.

The parameter values determine whether a DSGE model is saddle-path stable. Most DSGE models are saddle-path stable for some parameter values and not for other values. As we discussed in *How to write down a DSGE* in [DSGE] **intro 1**, the structural form of a DSGE model is

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

$$\mathbf{B}_0\mathbf{x}_{t+1} = \mathbf{B}_1 E_t(\mathbf{y}_{t+1}) + \mathbf{B}_2\mathbf{y}_t + \mathbf{B}_3\mathbf{x}_t + \mathbf{C}\boldsymbol{\epsilon}_{t+1} \tag{2}$$

Given this structural form, the values in the $\mathbf{A}_i$ and $\mathbf{B}_j$ matrices determine saddle-path stability.

In the process of solving the structural form in (1) and (2) for the state space, the **??** (**??**) solver computes the generalized eigenvalues of a matrix formed from the values in the $\mathbf{A}_i$ and $\mathbf{B}_j$ matrices. An eigenvalue is said to be stable when its absolute value is less than 1. The model is saddle-path stable when the number of stable eigenvalues equals the number of states in the model.

Once the maximization algorithm gets started, saddle-path stability usually is not an issue. Problems with saddle-path stability usually occur when trying to get the maximization process started. The initial values for the maximization algorithm must imply saddle-path stability because we cannot solve a DSGE model for parameter values that do not imply saddle-path stability. When the default initial values do not imply saddle-path stability, `dsge` searches for saddle-path-stable parameters, but there is no guarantee that it will find any.

`dsge` uses a series of small positive numbers as the default initial values for the structural parameters that appear in the $\mathbf{A}$ and $\mathbf{B}$ matrices. If you know that your model is not saddle-path stable when a parameter, say, β, is set to 0.1, you should use the `from()` option to specify an initial value that does imply saddle-path stability. Alternatively, you could reparameterize the model so that $\beta = 0.1$ does imply saddle-path stability, but this solution tends to be more work than changing the initial value.

In rare cases, when the true parameters are close to values that are not saddle-path stable, the maximum likelihood estimator will suffer from convergence problems. Better initial values can help find the solution in these cases.

What if the initial values are not saddle-path stable?

If the initial values do not imply a saddle-path stable solution and `dsge` cannot find any such values, you will see

```
. dsge ...
model is not saddle-path stable at current parameter values
    The number of stable eigenvalues is greater than the number of state
    variables.
  (output omitted)
```

or

```
. dsge ...
model is not saddle-path stable at current parameter values
    The number of stable eigenvalues is less than the number of state
    variables.
  (output omitted)
```

In either case, you should specify the `solve` and `noidencheck` options and subsequently use `estat stable` to find the problem. `solve` specifies that `dsge` only solve the model and not proceed with estimation, and `noidencheck` suppresses the check of whether the parameters are identified. `estat stable` will display the eigenvalues so that you can determine which ones are saddle-path stable and which ones are not. The trick is to change the starting values so that the number of stable eigenvalues equals the number of states in the model. Once you have determined which parameters are problematic, you can use the `from()` option to change the initial values to ones that do not have this problem.

Textbooks like those by **??** (**??**), **??** (**??**), **??** (**??**), and **??** (**??**) are full of examples in which some structural parameters must be either greater than 1 or less than 1 for the model to be saddle-path stable. Similarly, some parameters must be either greater than or less than 0 for the model to be saddle-path stable. Moving one structural parameter over one of these boundaries and looking at how the change affects the number of stable eigenvalues can help find a vector of parameters that yields a saddle-path stable solution.

That the model that you are trying to fit is likely related to other models in the literature or in the textbooks suggests another strategy. Any such references probably contain some discussion of the set of values that yield saddle-path stable solutions.

In especially difficult cases, you can use the strategy of solving the problem for a smaller model and building back up. You temporarily remove equations from the model, find a parameter vector that yields a saddle-path stable solution for the smaller model, and then progressively add back equations as you find initial values that yield saddle-path stable solutions.

▷ Example 1: Finding saddle-path stable initial values

Equations (1)–(6) model the observed control variable (inflation) p_t, the unobserved control variable (output gap) y_t, and the observed control variable (interest rate) r_t as functions of the states u_t and z_t. Lz_{t+1} is an auxiliary state that allows z_t to be a second-order process instead of a first-order process.

$$p_t = (1/\gamma)E_t(p_{t+1}) + \kappa y_t \tag{3}$$
$$y_t = E_t(y_{t+1}) - \{r_t - E_t(p_{t+1}) - z_t\} \tag{4}$$
$$r_t = \gamma p_t + u_t \tag{5}$$
$$u_{t+1} = \rho_u u_t + \xi_{t+1} \tag{6}$$
$$z_{t+1} = \rho_{z1} z_t + \rho_{z2} L z_t + \epsilon_{t+1} \tag{7}$$
$$Lz_{t+1} = z_t \tag{8}$$

We try to fit this model using default starting values below.

```
. use http://www.stata-press.com/data/r15/usmacro2
(Federal Reserve Economic Data - St. Louis Fed, 2017-01-15)
. dsge (p    = (1/{gamma})*E(F.p) + {kappa}*y)
>      (y    = E(F.y) -(r - E(f.p) - z), unobserved)
>      (r    = {gamma}*p + u)
>      (F.u  = {rho_u}*u, state)
>      (F.z  = {rho_z1}*z + {rho_z2}*Lz, state)
>      (F.Lz = z, state noshock)
model is not saddle-path stable at current parameter values
    The number of stable eigenvalues is greater than the number of state
    variables.
r(498);
```

The default initial values specify a model that is not saddle-path stable. We specify options `solve` and `noidencheck`, which cause the command to attempt only to solve the model and to skip the identification check.

```
. dsge (p    = (1/{gamma})*E(F.p) + {kappa}*y)
>      (y    = E(F.y) -(r - E(f.p) - z), unobserved)
>      (r    = {gamma}*p + u)
>      (F.u  = {rho_u}*u, state)
>      (F.z  = {rho_z1}*z + {rho_z2}*Lz, state)
>      (F.Lz = z, state noshock),
>      solve noidencheck
DSGE model
Sample: 1955q1 - 2015q4                         Number of obs     =        244
Log likelihood =          .
```

	OIM				
	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]
/structural					
gamma	.21	.	.	.	. .
kappa	.22	.	.	.	. .
rho_u	.23	.	.	.	. .
rho_z1	.24	.	.	.	. .
rho_z2	.25	.	.	.	. .
sd(e.u)	1	.			. .
sd(e.z)	1	.			. .

```
Note: Skipped identification check.
Warning: Model cannot be solved at current parameter values. Current values
         imply a model that is not saddle-path stable.
```

`dsge` displays the initial values used in the attempted solution, and it displays a warning message indicating that the model cannot be solved at these values because they imply a model that is not saddle-path stable.

We use `estat stable` to look at the eigenvalues implied by the initial values.

```
. estat stable
Stability results
```

	Eigenvalues
stable	-.3942
stable	.6342
stable	.21
stable	.23
unstable	2.605e+16
unstable	1.046

```
The process is not saddle-path stable.
The process is saddle-path stable when there are 3 stable eigenvalues and 3
    unstable eigenvalues.
```

The model is saddle-path stable when the number of stable eigenvalues equals the number of states in the model. The initial values yield four stable eigenvalues, but there are only three states in the model. So the model is not saddle-path stable at these values.

The initial value for `gamma` in the output from the failed solution attempt was 0.21. From (3), we see that the coefficient on expected future inflation is greater than one when `gamma` is less than one. We suspect that this relationship could be causing the lack of saddle-path stability. Below we use option `from()` to specify 1.2 as an initial value for `gamma` and attempt to solve the model.

```
. dsge (p    = (1/{gamma})*E(F.p) + {kappa}*y)
>      (y    = E(F.y) -(r - E(f.p) - z), unobserved)
>      (r    = {gamma}*p + u)
>      (F.u  = {rho_u}*u, state)
>      (F.z  = {rho_z1}*z + {rho_z2}*Lz, state)
>      (F.Lz = z, state noshock),
>      solve noidencheck from(gamma=1.2)
DSGE model
Sample: 1955q1 - 2015q4                         Number of obs     =        244
Log likelihood = -1504.7564
```

	OIM				
	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]
/structural					
gamma	1.2	.	.	.	. .
kappa	.22	.	.	.	. .
rho_u	.23	.	.	.	. .
rho_z1	.24	.	.	.	. .
rho_z2	.25	.	.	.	. .
sd(e.u)	1	.			. .
sd(e.z)	1	.			. .

```
Note: Skipped identification check.
Note: Model solved at specified parameters.
```

There is no warning message, so the parameter values yielded a saddle-path stable solution. As a final illustration of this point, we display the eigenvalues below.

```
. estat stable

Stability results
```

	Eigenvalues
stable	-.3942
stable	.6342
stable	.23
unstable	5.380e+16
unstable	1.2
unstable	1.264

```
The process is saddle-path stable.
```

We could now proceed with estimation.

◁

Also see

[DSGE] **intro 6** — Identification

[DSGE] **intro 7** — Convergence problems

[DSGE] **estat stable** — Check stability of system