

Roadmap for W7 Slides

- Model and data: W7 Slides 3-10
- Dynare setup (measurement equations, initialization): W7 Slides 11-15
- Priors and estimation command: W7 Slides 18-22
- MH details and tuning: W7 Slides 21-24 and Slide 31
- Extensions and comparison via marginal likelihood: W7 slides 32-35

Errors in all .mod files

- In Dynare, bounds come either from (i) the explicit lower/upper bounds you put in estimated_params, or (ii) implicit bounds induced by the prior + prior_trunc.
- The original .mod file sets:

```
1 | prior_trunc=1e-32
```

- Change it to `prior_trunc=0`

Why `prior_trunc = 0` Works (Even Though $0 < 1e-32$)

Key point

In Dynare, setting

```
1 | prior_trunc = 0;
```

does not mean “truncate at zero.”

It means “do not truncate the prior at all.”

So:

- `prior_trunc = 1e-32` → *extremely tight truncation* (often numerically unstable)
- `prior_trunc = 1e-10` → *mild truncation* (often fine, still working for our case)

- `prior_trunc = 0` → **truncation is switched off**

That's why `0` can work even though it is smaller than `1e-32`.

What `prior_trunc` actually does in Dynare

Dynare constructs an **effective support** for each prior by trimming tails where the prior density is “too small” relative to its peak. A useful way to think about it is:

$$\{\theta : p(\theta) \geq \text{prior_trunc} \cdot \max_{\theta} p(\theta)\}.$$

- If `prior_trunc > 0`, Dynare **numerically trims** the tails of the prior using a **relative density cutoff**.
- If `prior_trunc = 0`, Dynare **skips truncation entirely**.

So `prior_trunc` is a *relative density threshold*, **not** a lower bound on the parameter.

Why `1e-32` can break the code

In principle, `1e-32` sounds “almost zero.” In practice, it can be problematic because Dynare’s internal computations rely on:

- **double precision** floating point arithmetic,
- **log densities** (and transformations back and forth),
- numerical routines for mode-finding / Hessians, etc.

For skewed priors (e.g., Gamma; Beta near boundaries), a cutoff as tight as `1e-32` can cause:

- numerically computed truncation bounds to collapse inward, and
- small rounding error to place your initial value *just outside* the allowed interval.

That’s how you can end up with the error:

Initial value(s) of `phi_pic` are outside parameter bounds.

even when the printed starting value looks reasonable.

Why `prior_trunc = 0` is often safest

With `prior_trunc = 0`:

- Dynare relies on **explicit bounds** you set in `estimated_params`, and/or
- the **natural support** of the prior distribution (e.g., Gamma implies parameter > 0).

This is why many applied Dynare users either set:

```
1 | prior_trunc = 0;
```

or use a modest value like:

```
1 | prior_trunc = 1e-8; % or 1e-10
```

and then **always set explicit bounds** for key parameters (policy coefficients, std. devs, etc.).

Practical rule of thumb

- **Debugging:**

```
1 | prior_trunc = 0;
```

- **Production estimation (with explicit bounds):**

```
1 | prior_trunc = 1e-8 or 1e-10
```

- If your model is running fine with 1e-10 (or 1e-8), keep it. Prefer the smaller value that remains stable.
- If you see tail/bounds weirdness again, increase priortrunc (e.g., to 1e-8, 1e-6, or 1e-5) or set priortrunc=0 and impose explicit bounds in estimated_params.
- **Avoid extremely tiny values** like `1e-20` or smaller unless you have a specific reason and you have verified numerical stability.