

Dynare 手册

陈普 翻译

2023 年 2 月 17 日

4.7 初始和终端条件

对于多数练习而言，提供初值（或终值）条件是必须的，因为你必须对非线性求解器提供一个初始猜测（译者注：解非线性方程组必须提供初值）。本节描述为该目的而使用的语句。

很多时候（确定性或随机性求解时）需要计算非线性模型的稳态：`initval`用来设定非线性求解器的初值，命令`resid`用来计算使用该初值的非线性方程的残差。

在完美预期模式下，前向模型要求 Dynare 提供初值和终值条件。通常初值和终值条件都是静态均衡下的取值，但也不必然。

一个典型的应用就是，在 0 时期的一个经济体，它在第 1 期产生一个冲击，然后追踪其回到初始状态的轨迹。为达此目的，需要`initval`和`shocks`（见 Shock on exogenous variables 一节）。

另一个应用就是研究经济体从 0 时期任意一个初始值如何收敛至均衡。此时，命令`histval`允许在模拟开始之前，对有滞后值的变量设定不同的历史初值。基于 Dynare 的设计，此时`initval`就用来设定终值了。

4.7.1 语块：initval

Block: `initval(OPTIONS...);`

`initval`块有两个目的：一是在完美预期模拟的情景下提供非线性求解器一个猜测值，二是在完美预期和随机模拟情景下提供稳态计算的猜测值。在完美预期模拟中，它也会被用来声明初值和终值条件，这取决于是否存在`histval`和`endval`语块。由于完美预期情景下`histval`和`endval`的存在带给`initval`语块的交互含义，我们强烈推荐，在运行`perfect_foresight_setup`之后和运行`perfect_foresight_solver`之前，检查结构体变量`oo_.endo_simul`和`oo_.exo_simul`，看它们是否存储了的你想要的值。当变量存在前导和滞后时，这些结构体的子域会在第一列/行存储滞后变量的历史值，以及最后一列/行存储前导变量的终值。

`initval`语块用`end`结束，它包含的代码行形式如下：

`VARIABLE_NAME=EXPRESSION;`

在确定性（即完美预期）模型中

首先，该语块提供的值会填满存储内生和外生变量的`oo_.endo_simul`和`oo_.exo_simul`变量。若无其他语块存在，意味着它也填充了这些矩阵的最后一列/行，相当于它提供了所有内生变量和外生变量的初值和终值条件。同时它也提供了求解器计算中间模拟期变量的初值。若存在`histval`语块（因此不会存在`endval`语块），该语块将通过设置`oo_.endo_simul`和`oo_.exo_simul`的第一列/行，而提供（或重写）状态变量（滞后）的历史值。这意味着当`histval`语块存在时，`initval`语块仅仅设置了前导变量的终值，并为完美预期下的求解器提供了计算的初值。

考虑到`initval`的各种功能，通常需要在`initval`语块中为所有内生变量提供数值。求解器需要一个可行的初始值，初值和终值条件对于滞后和前导变量而言必须要提供。如果某些内生或外生变量没有

在`initval`语块中设置，0 值会被默认设置，意识到这一点很重要。因为当用`varexo`设置外生变量时，如TFP 是不允许取值为 0 的。

注意到如果`initval`语块之后立刻跟着`steady`命令，它的语义会有轻微改变。`steady`命令会假设外生变量取`initval`语块中的值不变，并计算模型中所有内生变量的稳态值。基于该外生变量取值的内生变量稳态值存储在`oo_.endo_simul`中，它不仅取代了求解器要求的初值角色，也取代了历史条件和终值条件的角色。因此，`initval`模块之后紧跟`steady`命令等价于外生变量取`initval`中的值，内生变量为条件于该外生变量的稳态值。

在随机模型中

`initval`的主要目的是在稳态计算中提供非线性求解器一个初始猜测值。注意，如果`initval`之后并未紧跟`steady`命令，稳态计算也会被接下来的命令（`stoch_simul, estimation...`）触发。

因此，当对`ramsey_model`的计算提供了一个具有解析解的稳态文件时，`initval`允许对该稳态设定一个初始的工具值（`initial instrument value`）。

没有必要对外生随机变量初值声明为 0，它仅仅是一个可能的值而已。

后续计算的稳态（不是初值，要用初值请使用历史值）用来作为所有时期在模拟开始前的初始条件，这包括三种可能的随机模拟模式：

- `stoch_simul`，如果`periods`被设定；
- `forecast`作为预测计算的初始点；
- `conditional_forecast`作为条件预测的初始点。

要从一个不是稳态而是某个特殊的值开始模拟，请使用`histval`。

选项

`all_values_required`

当至少有一个内生或外生变量没在`initval`语块中设置时，将输出错误并停止处理`.mod` 文件。

例子

```
initval;
c = 1.2;
k = 12;
x = 1;
end;

steady;
```

4.7.2 语块：endval

`endval(OPTION...);`

`endval`语块用`end`结束，它包含的代码行形式如下：

`VARIABLE_NAME=EXPRESSION;`

`endval`语块只在确定性模型中有意义，且不能与`histval`语块一起使用。与`initval`命令一样，它将在语块内的值填满存储内生和外生变量的`oo_.endo_simul`和`oo_.exo_simul`变量。如果没有`initval`语块，它将填充整个矩阵，也就是说，因为它也填充了这些矩阵的第一列/行和最后一列/行，因此它提供了所有内生变量和外生变量的初值和终值。此外，因为它也填充了中间模拟期，相当于也提供了求解器对这些中间模拟期的一个初始猜测值。

如果`initval`语块存在，`initval`将提供变量（如果有状态变量或者滞后变量）的历史值，`endval`填充矩阵余下部分，也就是说，`endval`将提供：1）进入模型的前导变量的终端条件；2）完美预期求解器所有模拟期中所有内生变量的初始猜测值。

注意，如果某些内生或外生变量没有在`endval`语块中提及，它们的值由最后那个`initval`或`steady`命令（如果有）提供。此时遗漏的变量不会自动填充为 0。再一次强烈推荐在运行`perfect_foresight_setup`之后，运行`perfect_foresight_solver`之前，检查结构体变量`oo_.endo_simul`和`oo_.exo_simul`，里面存储的是不是你想要的值。

与`initval`一样，如果`endval`之后立刻跟着`steady`命令，它的语义会发生轻微变化。`steady`命令会基于`endval`中外生变量的值不变，计算模型中所有内生变量的稳态值。这些条件于外生变量的稳态值会被写进`oo_.endo_simul`，从而不仅提供了求解器的初值，也提供了历史值和终值。因此，正式地说，`endval`语块紧跟`steady`等价于`endval`设定外生变量值，内生变量设为相应的稳态值。

选项

`all_values_required`

见前面。

例子

```
var c k;
varexo x;

model;
c + k - aa*x*k(-1)^alph - (1-delt)*k(-1);
c^(-gam) - (1+bet)^(-1)*(aa*alph*x(+1)*k^(alph-1) + 1 - delt)*c(+1)^(-gam);
end;

initval;
c = 1.2;
k = 12;
x = 1;
end;

steady;

endval;
c = 2;
k = 20;
x = 2;
end;

steady;

perfect_foresight_setup(periods=200);
perfect_foresight_solver;
```

本例中，要解决的就是给定外生技术水平 x ，求消费和资本在 $t = 1$ 至 $T = 200$ 的最优路径。 c 是前向变量， k 是纯后向变量（状态变量），外生变量 x 在物质资本的期望收益中带有前导。

最初的均衡稳态值基于 $x=1$ 计算，终值则基于 $x=2$ 计算。`initval`语块设置了 k 的初始条件（因为它仅是一个后向变量），`endval`语块设置了 c 的终端条件（因为它仅是一个前向变量）。完美预期求解器的初始猜

测值由endval语块给出。看下面更多的细节。

例子

```

var c k;
varexo x;

model;
c + k - aa*x*k(-1)^alph - (1-delt)*k(-1);
c^(-gam) - (1+bet)^(-1)*(aa*alph*x(+1)*k^(alph-1) + 1 - delt)*c(+1)^(-gam);
end;

initval;
k = 12;
end;

endval;
c = 2;
x = 1.1;
end;

perfect_foresight_setup(periods=200);
perfect_foresight_solver;

```

本例中，没有steady命令，因此，initval和endval语块精确设定了相应条件。我们需要c和x的终端条件，因为它们都有前导，也需要k的初始条件，因为它有滞后。

在endval语块中设置x=1.1，若无shocks语块，则意味着技术在 $t = 1$ 时为 1.1，并永久保持，因为除了第一列外，endval填充了oo_.endo_simul和oo_.exo_simul所有元素。第一列存储了初始值，初始值若没有设置，则initval语块会将其默认设为 0。

因为资本的运动是后向的，我们需要k在 $t = 0$ 期的初值。由于endval的存在，这不能通过histval语块完成，但必须在initval语块中设定。类似地，因为欧拉方程是前向的，我们需要 $t = 201$ 期的c的终值，它在endval语块中设置。

如上所见，没有必要在initval语块中设定c和x，也没有必要再endval语块中设置k，它们对结果都没有影响。因为第一期的优化问题就是基于前定的由 $t = 0$ 期而来的资本存量k以及当前和未来的技术水平x，选择 $t = 1$ 期的c,k，此时 $t = 0$ 期的c和x的值就没有起作用。类似地， $t = 200$ 期的c,k选择也不依赖于 $t = 201$ 期的k。如欧拉方程所示，该选择（译者注：消费的选择）仅依赖于未来的消费c和技术水平x以及当期的资本存量，并不包含未来的资本存量k。直觉就是这些没起作用的变量分别是 $t = 0$ 和 $t = 201$ 期优化问题的结果，它们不需要在 $t = 1$ 和 $t = 200$ 期进行建模。

例子

```

initval;
c = 1.2;
k = 12;
x = 1;
end;

endval;
c = 2;
k = 20;
x = 1.1;

```

`end;`

本例中，提供了前向变量 x 和 c 的初值，也提供了后向变量的 k 的终值。如前所述，这些值不会影响模拟的结果。**Dynare** 将把它们作为给定，基本上是假设存在某些外生和状态变量使得该选择成为均衡值（基本上是未设定时期，即 $t < 0$ 和 $t > 201$ 期的初始或终端条件）。

上面这个例子表明看待`initval`和`endval`之后使用`steady`的另一种方式。不是说模拟范围之前或之后的隐含未设定条件必须要符合那些语块中内生变量的初值和终值条件，而是`steady`设定的 $t < 0$ 和 $t > 201$ 的条件等于`initval`和`endval`语块中外生变量决定的内生变量的稳态值。这些内生变量在 $t = 0$ 和 $t = 201$ 期被设为相应的稳态均衡值。

在`initval`中设定的 $t = 0$ 期的 c 和在`endval`中设定的 $t = 201$ 期的 k ，在绘制内生变量的模拟向量时具有重要的含义。即，`oo_.endo_simul`的行：该向量将包含初值和终值条件，因此本例中有 202 期。当你对外向变量设定任意的初值，对后向变量设定任意的终值，这些变量将离内生决定的 $t = 1$ 和 $t = 200$ 期的内生变量的值很远。当 $t = 0$ 期和 $t = 201$ 期的值无关于 $0 < t < 201$ 期的动态路径时，它们也许会导致一个看起来很奇怪的大跳跃。在上例中，当你使用`rplot`或手动绘制`oo_.endo_val`时，消费将在 $t = 0$ 至 $t = 1$ 时呈现出一个大跳跃，资本也会在 $t = 200$ 至 $t = 201$ 时呈现一个大跳跃。

4.7.3 语块: `histval`

```
histval(OPTION...);
```

在确定性完美预期情景中

模型中有变量滞后期超过一期，`histval`语块可以对这些状态变量设定不同的历史初值。此时，`initval`语块仅设定终值和求解器的初值。注意`histval`语块不设置非状态变量。

`histval`语块用`end`结束，它包含的代码行形式如下：

```
VARIABLE_NAME=EXPRESSION;
```

`EXPRESS`可以是任意返回数值的有效表达式，也能包含已经初始化的变量名。

Dynare 的传统是时期 1 是模拟的第一期。在时间上回溯，模拟开始前一期是时期 0，然后是-1，等等。

没有在`histval`语块中初始化的状态变量将在 0 期以及之前时期被设为 0。注意，`histval`语块后不能跟`steady`。

例子

```
model;
x=1.5*x(-1)-0.6*x(-2)+epsilon;
log(c)=0.5*x+0.5*log(c(+1));
end;
```

```
histval;
x(0)=-1;
x(-1)=0.2;
end;
```

```
initval;
c=1;
x=1;
end;
```

本例中, `histval` 用来设定内生变量 `x` 的两个滞后的历史值, 存储在 `oo_.endo_simul` 的第一列。 `initval` 语块用来设置前向变量 `c` 的终端条件, 存储在 `oo_.endo_simul` 的最后一列。而且, `initval` 语块定义了完美求解器要求的内生变量 `c` 和 `k` 的初值。

在随机模拟情景中

随机模拟情景中, `histval` 允许设置状态空间模拟的初始点。与完美预期模拟一样, 未设置的变量将被设为 0。而且, 当仅有状态变量进入递归的政策函数时, 所有控制变量的设定值将被忽略。它们在如下情况中被使用,

- 设定了 `periods` 选项的 `stoch_simul`。注意, 这仅会影响模拟的初始值, 而不会影响脉冲响应函数。当使用 `loglinear` 选项时, `histval` 语块将仍取未对数化的初始值。
- 在 `forecast` 中进行预测时会作为初始点。当使用 `loglinear` 选项时, `histval` 语块将仍取未对数化的初始值。
- 在校准了的模型中使用 `conditional_forecast` 会被作为条件预测计算的初始点。当使用 `loglinear` 选项时, `histval` 语块将仍取未对数化的初始值。
- 在 `Ramsey policy` 中, 计划者的目标函数是基于内生状态变量计算的, 因此它也会设定内生状态变量 (包括滞后的外生变量) 值。注意, 对应计划者的拉格朗日乘子的初值不能被设定 (见 `evaluate_planner_objective`)。

选项

`all_values_required`

见前面。

例子

```
var x y;
varexo e;

model;
x = y(-1)^alpha*y(-2)^(1-alpha)+e;
end;

initval;
x = 1;
y = 1;
e = 0.5;
end;

steady;

histval;
y(0) = 1.1;
y(-1) = 0.9;
end;

stoch_simul(periods=100);
```

4.7.4 命令 resid

该命令将基于最后的initval或endval语块（或你提供的稳态文件，见 **Steady state**）中给定的内生变量的值，呈现模型静态方程的残差。

4.8 外生变量的冲击

在确定性情景下，当人们想研究一个均衡到另一个均衡的转移路径时，等价于分析一个永久冲击的效果。这在 **Dynare** 中可以通过使用合适的initval和endval来完成。

另一个典型的实验就是研究暂时性冲击的效果，即在暂时性冲击之后系统又回到原来的均衡（如果系统是稳定的）。暂时性冲击是模型中一个或多个外生变量的临时变化。临时性冲击可在命令shocks中设置。

在随机性情景下，每期的外生变量都将取随机值。在 **Dynare** 中，这些随机值遵循零均值的正态分布，使用者可以设定这些冲击的波动幅度。冲击的方差协方差矩阵中的非零元素也能在shocks中设置。或者，整个矩阵可用Sigma_e变量直接设定。

如果外生变量冲击的方差设为 0，该变量会出现在政策和转移函数的报告中，但在脉冲响应函数和矩计算中不会被使用。设定方差为 0 是去除外生冲击的一个便捷方式。

注意，缺省情况下，如果在.mod文件中存在多个shocks或mshocks语块，它们会被累加：语块中声明的所有冲击都会被考虑。然而，如果shocks或mshocks语块包含overwrite选项，它将替代之前所有的shocks和mshocks语块。

4.8.1 语块 shocks;

Block:shocks(overwrite);

见上面关于overwrite选项的含义。

确定性情景

对于确定性模拟，语块shocks设定外生变量的临时变化。永久性变化用endval语块。

语块应该包含如下一个或多个“三行组”：

```
var VARIABLE_NAME;
periods INTEGER[:INTEGER] [[,] INTEGER[:INTEGER]]...;
values DOUBLE | (EXPRESSION) [[,] DOUBLE | (EXPRESSION) ]...;
```

可以设定冲击持续多期并随时间而变化。periods关键词接受多个时期或时期范围的设置，但必须与values关键词包含的冲击值的数目相匹配。注意periods关键词对日期范围的设置也能与values关键词的单值匹配。如果values是个标量，则该值将应用到整个时期范围。如果values是向量，则其元素的数目必须与时期范围的期数匹配。

注意，冲击值不严格限制为数值常数：任意表达都是允许的，但你必须用小括号把它们括住。

periods可行的范围从 0 至在perfect_foresight_setup中periods设置的数值。

警告：注意内生模拟期中的第一期是数值 1。因此，对第 0 期冲击值的设定可能会与initval或endval（取决于当时的情景）中初值设定产生冲突（即会重写，或被重写）。使用者应该确认在perfect_foresight_setup之后oo_.exo_simul是否有着正确设置。

例子（标量值）

```
shocks;
var e;
```

```

periods 1;
values 0.5;

var u;
periods 4:5;
values 0;

var v;
periods 4:5 6 7:9;
values 1 1.1 0.9;

var w;
periods 1 2;
values (1+p) (exp(z));
end;

```

例子（向量值）

```

xx = [1.2; 1.3; 1];

shocks;
var e;
periods 1:3;
values (xx);
end;

```

随机情景

在随机模拟中，`shocks`语块设定外生变量冲击协方差矩阵的非零元素。你可以使用如下形式：

- 设定外生变量的标准差。
`var VARIABLE_NAME; stderr EXPRESSION;`
- 设定外生变量的方差。
`var VARIABLE_NAME = EXPRESSION;`
- 设定两个外生变量的协方差。
`var VARIABLE_NAME, VARIABLE_NAME = EXPRESSION;`
- 设定两个外生变量的相关系数。`corr VARIABLE_NAME, VARIABLE_NAME = EXPRESSION;4.8.`

若是在估计模型，那你也可以设置内生变量的方差和协方差：此时，设置的值被当做这些内生变量度量误差的校准值。这要求在`shocks`语块之前用`varobs`命令进行设置。

例子

```

shocks;
var e = 0.000081;
var u; stderr 0.009;
corr e, u = 0.8;
var v, w = 2;
end;

```

随机最优政策情景

当在`ramsey_model`或`discretionary_poicy`情景下计算条件福利时，福利是以计划者在第一期做出选择时拥有的状态变量为条件的。第一期的信息集包括外生变量冲击各自的实现。因此，他们已知的值可用完美预期的语法设定。注意 1) 所有非第 1 期的其他值都将被忽略。2) 滞后冲击值（比如新闻冲击）在`histval`中设置。

例子

```
shocks;
var u; stderr 0.008;

var u;
periods 1;
values 1;
end;
```

确定性和随机冲击的混合

代理人若在模拟开始时就知道未来的外生改变,那么也可以混合确定性和随机性冲击。此时,`stoch_simul`计算的是给状态空间添加了未来信息时的理性预期解,然后`forecast`计算的是条件于初始条件和未来信息的模拟。

例子

```
varexo_det tau;
varexo e;
...

shocks;
var e; stderr 0.01;

var tau;
periods 1:9;
values -0.15;
end;

stoch_simul(irf=0);
forecast;
```

4.8.2 语块 `mshocks`

```
Block:mshocks(overwrite);
```

该语块的目的类似于确定性情景下的`shocks`，只是此时给定的数值以乘法意义看待。比如，某时刻某外生变量冲击取值 1.05，该值为在稳态（在最后一个`initval`或`endval`后给出）之上 5%。

语法类似于确定性情景下的`shocks`。

该命令仅在以下两种情况下有意义：

- 确定性情景下，外生变量有非零稳态
- 随机情景下，确定性外生变量有非零稳态

选项`overwrite`的意义可看前面。