Using locproj to easily estimate nonlinear local projections

Alfonso Ugarte-Ruiz

BBVA Research

Stata UK Conference, September 11, 2025

Overview

The possibilities of different types of nonlinearities in local projections are plentiful and the applications are growing quickly

1. Nonlinearities in LPs and in LOCPROJ
   - What is an IRF and a local projection?
   - Nonlinearities with LOCPROJ
2. Linear in parameters
   - Polynomial terms
   - Interaction with a categorical variable
   - Interaction with a continuous variable
3. Nonlinear in parameters
   - Binary dependent variable
   - Quantile local projections

Outline

Impulse Response Functions using Local Projections

Following [Jordà and Taylor, 2024], let y_t denote an outcome variable of interest, s_t the policy intervention variable, and let x_t denote a vector of controls variables. Formally, we define an impulse response function (IRF) as:

$$\mathcal{R}_{sh}(h) = E[y_{t+h}|s_t = s_0 + \delta; x_t] - E[y_{t+h}|s_t = s_0; x_t]; \qquad h = 0, 1, ..., H, \tag{1}$$

where s_0 denotes the value of the variable s_t without intervention and δ is the size of the intervention. The local projection of y_{t+h} on s_t can be estimated with the following regressions:

$$y_{t+h} = \alpha_h + \beta_h s_t + \gamma_h' x_t + \nu_{t+h}; \qquad h = 0, 1, ..., H \tag{2}$$

In local projections, we can deal with nonlinearities by introducing new variables that are powers of the variable s_t, interactions with categorical or continuous variables, or by using models that are nonlinear in parameters.

Outline

What does LOCPROJ do?

`locproj` reports the Impuls Response Function (IRF), together with its standard error and confidence interval, as an output matrix and through an IRF graph.

The options allow defining the desired specification in a fully automatic way or in a more explicit way, with many alternatives in between. One can write a specification in which the response variable is y and the shock variable is s as simply as:

```
.  locproj y n
```

Or equivalently as:

```
.  locproj y, shock(n)
```

There are several options to define the specification and estimation method, and to display, save, and plot the results. You can see [Ugarte, 2025] or the help file for more details.

What are the outputs after running LOCPROJ?

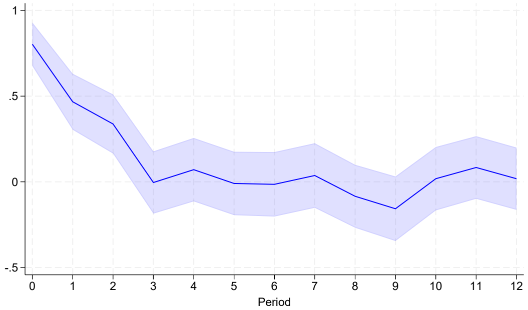
IRF Matrix

```
. locproj y, shock(n n_2) ylags(1) slags(2) controls(l(0/3).r) hor(12)

Impulse Response Function
```

	IRF	Std.Err.	IRF LOW	IRF UP
0	0.80286	0.06453	0.67570	0.93002
1	0.46729	0.08295	0.30384	0.63074
2	0.33747	0.08759	0.16486	0.51007
3	-0.00406	0.09224	-0.18581	0.17770
4	0.07093	0.09393	-0.11417	0.25603
5	-0.00958	0.09391	-0.19464	0.17547
6	-0.01456	0.09547	-0.20271	0.17358
7	0.03688	0.09565	-0.15162	0.22538
8	-0.08434	0.09323	-0.26807	0.09940
9	-0.15716	0.09561	-0.34560	0.03127
10	0.01818	0.09392	-0.16692	0.20329
11	0.08368	0.09268	-0.09898	0.26634
12	0.01808	0.09231	-0.16386	0.20003

IRF Graph



How does LOCPROJ deal with nonlinearities?

- If variables s1, s2 and s3 are introduced in the option `shock()` as: `shock(s1 s2 s3)`, `locproj` includes those three variables in the specification, includes also their lags if the option `slags()` specifies so, and then runs the expression $(s1 + s2 + s3)$ through the command `lincom`
- Directly writing a linear combination of parameters through the option `lcs()` which is equivalent to running that expression through the command lincom, e.g. `lcs(2*s1+2*2*s2+3*s3)`
- Using the option `margins`, which estimates derivatives of the response with respect to the shock variable rather than using its estimated parameter. It is equivalent to using the Stata command `margins` and `locproj` allows including its available options by using the option `mropt()`.

Outline

Quadratic terms

The most basic case of nonlinearity is when the effect of the shock can be different at different levels of the shock variable itself, such as when we include a quadratic term:

$$y_{t+h} = \alpha_h + \beta_h^a s_t + \beta_h^b s_t^2 + \gamma_h' x_t + \nu_{t+h} \tag{3}$$

Using (3) in (1), we get that in general the response depends on the initial level of the variable, s_0, and the size of the intervention, δ, in the following way:

$$\mathcal{R}_{sh}(h) = \beta_h^a(s_0 + \delta) + \beta_h^b(s_0 + \delta)^2 + \gamma_h' x_t - [\beta_h^a(s_0) + \beta_h^b(s_0)^2 + \gamma_h' x_t)]$$

$$\mathcal{R}_{sh}(h) = \beta_h^a \delta + 2\beta_h^b s_0 \delta + \beta_h^b \delta^2 \tag{4}$$

Other polynomial terms

If we had a cubic term in the specification:

$$y_{t+h} = \alpha_h + \beta_h^a s_t + \beta_h^b s_t^2 + \beta_h^c s_t^3 + \gamma_h' x_t + \nu_{t+h} \tag{5}$$

The response would be given by:

$$\mathcal{R}_{sh}(h) = \beta_h^a \delta + 2\beta_h^b s_0 \delta + \beta_h^b \delta^2 + 3\beta_h^c s_0^2 \delta + 3\beta_h^c s_0 \delta^2 + \beta_h^c \delta^3 \tag{6}$$

Example with a quadratic term

- We follow Example 5.4 in [Ugarte, 2025], where we are interested in estimating the IRF from a shock to the variable n (Growth rate of hours worked) into the variable y (Growth rate of real GDP).
- The first step is to generate a new variable equal to the square of *n*:

```
. gen n_2 = n^2
```

- Alternatively, we can use an interaction term such as `c.n#c.n`. For simplicity, we use the new generated variable `n_2`, but the result will be exactly the same if we use `c.n#c.n` instead.
- Following equation (4), if we assume $\delta = 1$ and $\delta^2 = 1$, and the initial level $s0 = 0$, then we would only need to include the two variables `n` and `n_2` inside the option `shock()`:

```
. locproj y, shock(n n_2) ylags(1) slags(2) controls(l(0/3).r) hor(12)
```

Example with a quadratic term: using the option LCS()

- However, if the initial level of the variable `n` is different than zero, i.e. $s0 \neq 0$, then we need to use the option `lcs()`.

  We write the expression in equation (4) with $\delta = 1$ and $\delta^2 = 1$, and we can use as the initial level of the variable `n` ($s0$) its estimated sample mean, which we denote *nm*:

```
.   sum n
.   scalar nm=r(mean)

.   locproj y, shock(n n_2) ylags(1) slags(2) controls(l(0/3).r) hor(12)
lcs(n*1+2*n_2*1*nm+n_2*1)
```

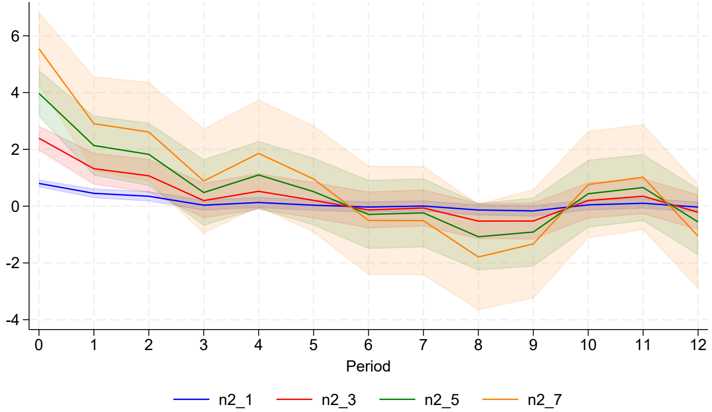
Example with a quadratic term: using the option LCS()

- If we want to estimate the IRF for other values of the variable n, for instance for $n = 3$ and $n = 5$, and thus, $n^2 = 9$ and $n^2 = 25$ we would need to change the expression that goes into the option `lcs()`:

```
. locproj y, shock(n n_2) ylags(1) slags(2) controls(l(0/3).r) hor(12)
lcs(n*3+2*n_2*3*nm+n_2*9)
. locproj y, shock(n n_2) ylags(1) slags(2) controls(l(0/3).r) hor(12)
lcs(n*5+2*n_2*5*nm+n_2*25)
```

We can graph the different IRFs using LPGRAPH

Usually, we are interested in comparing the IRF at different levels of the shock. `locproj` allows us to save the results and we can plot them with `lpgraph`



Outline

Interaction with a dummy variable (State dependent LPs)

One of the most common nonlinearities is that of a state-dependent LP, when the shock interacts with another variable that defines a state, which is usually characterized by a binary (dummy) variable $D_t \in \{0, 1\}$ that defines whether a state is active or not.

If we want to express the response as a difference with respect to the other state, our specification would be:

$$y_{t+h} = \alpha_h + \beta_h^a s_t + \beta_h^b D_t s_t + \gamma_h' x_t + \nu_{t+h} \tag{7}$$

If we want to express the response as an absolute impact at each one of the two states, our specification would be:

$$y_{t+h} = \alpha_h + \beta_h^a (D_t = 0) s_t + \beta_h^b (D_t = 1) s_t + \gamma_h' x_t + \nu_{t+h} \tag{8}$$

IRF of each state depending on specification

$\mathcal{R}_{sh}(h)$	**Equation (7)**	**Equation (8)**
Difference with other state	β_h^b	
Total response if $D_t = 1$	$\beta_h^a + \beta_h^b$	β_h^b
Total response if $D_t = 0$	β_h^a	β_h^a

Table: IRF according to specification

Interaction with a dummy variable: Introducing extra terms in the option SHOCK()

We want to specify a different reaction before and after the global financial crisis (GFC). We first generate two dummy variables. The first one `bef_gfc` is equal to one before the first quarter of 2009 and zero afterwards, while the second one `aft_gfc` is equal to one after the first quarter of 2009 and zero otherwise, and we denote the latter as D_t.

```
. gen bef_gfc = dateq<tq(2009q1)
. gen aft_gfc = dateq>=tq(2009q1)
```

- Total response if $D_t = 1$ (after GFC) according to equation (7):
  ```
  . locproj y l.y l(0/3).r, s(n c.n#c.aft_gfc) sl(2) hor(12)
  ```
- Total response if $D_t = 1$ according to equation (8):
  ```
  . locproj y l.y l(0/3).r l(0/2)(c.n#c.bef_gfc), s(c.n#c.aft_gfc) sl(2)
  hor(12)
  ```

We can play with the syntax to get the response in the other state

- Total response if $D_t = 0$ (before GFC) according to equation (7):

```
. locproj y l.y l(0/3).r l(0/2)(c.n#c.aft_gfc), s(n) sl(2) hor(12)
```

- Total response if $D_t = 0$ according to equation (8):

```
. locproj y l.y l(0/3).r l(0/2)(c.n#c.aft_gfc), s(c.n#c.bef_gfc) sl(2)
hor(12)
```

We can always verify the results by looking at the output of each step

We can display the regression output of the first two steps, $h = 0, 1$, without any other control variable to see how the coefficients of the two variables are added:

```
. locproj y, s(n c.aft_gfc#c.n) hor(1) noi noconstant
y_h(0)
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
n	.9958853	.0581427	17.13	0.000	.8819277	1.109843
c.aft_gfc#c.n	-.4854552	.1799913	-2.70	0.007	-.8382317	-.1326787

y_h(1)

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
n	.6542207	.0751751	8.70	0.000	.5068801	.8015612
c.aft_gfc#c.n	-.4244693	.2360608	-1.80	0.072	-.88714	.0382014

Impulse Response Function

	IRF	Std.Err.	IRF LOW	IRF UP
0	0.51043	0.17034	0.17489	0.84597
1	0.22975	0.22377	-0.21105	0.67055

Interaction with a dummy variable: Using the option LCS()

As the expression that we introduce in option `lcs()` must be a linear combination of variables included in the specification, we must make sure that we correctly specify each of the variables when using the Stata factor variables syntax.

- Total response if $D_t = 1$ (after GFC) according to equation (7):

```
. locproj y l(0/2)(n aft_gfc#c.n) l(0/3).r, yl(1) lcs(n+1.aft_gfc#c.n)
```

- Total response if $D_t = 1$ according to equation (8):

```
. locproj y l(0/2)(aft_gfc#c.n) l(0/3).r, yl(1) lcs(1.aft_gfc#c.n)
```

- Total response if $D_t = 0$ (before GFC) according to equation (7):

```
. locproj y l(0/2)(n aft_gfc#c.n) l(0/3).r, yl(1) lcs(n)
```

- Total response if $D_t = 0$ according to equation (8):

```
. locproj y l(0/2)(aft_gfc#c.n) l(0/3).r, yl(1) lcs(0.aft_gfc#c.n)
```

We can checkout again the output of each step in each specification

Regression outputs when using specification in eq. (7)

```
. locproj y (n aft_gfc#c.n), lcs(n+1.aft_gfc#c.n) h(1) noi nograph noconstant
y_h(0)
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
n	.9958853	.0581427	17.13	0.000	.8819277	1.109843
aft_gfc#c.n						
0	0	(empty)				
1	-.4854552	.1799913	-2.70	0.007	-.8382317	-.1326787

y_h(1)

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
n	.6542207	.0751751	8.70	0.000	.5068801	.8015612
aft_gfc#c.n						
0	0	(empty)				
1	-.4244693	.2360608	-1.80	0.072	-.88714	.0382014

Regression outputs when using specification in eq. (8)

```
. locproj y aft_gfc#c.n, lcs(1.aft_gfc#c.n) h(1) noi nograph noconstant
y_h(0)
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
aft_gfc#c.n						
0	.9958853	.0581427	17.13	0.000	.8819277	1.109843
1	.5104301	.1703417	3.00	0.003	.1765665	.8442937

y_h(1)

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
aft_gfc#c.n						
0	.6542207	.0751751	8.70	0.000	.5068801	.8015612
1	.2297513	.2237709	1.03	0.305	-.2088315	.6683342

Outline

Interaction with a continuous variable

We can interact our shock variable with a continuous variable that alters the reaction of our outcome variable.

$$y_{t+h} = \alpha_h + \beta_h^a s_t + \beta_h^b s_t * z_t + \gamma_h' x_t + \nu_{t+h} \tag{9}$$

In this case the IRF would be given by $\mathcal{R}_{sh}(h) = \beta_h^a + \beta_h^b$ if the size of the shock and the interaction variable are normalized. In a more general way,

$$\mathcal{R}_{sh}(h) = \beta_h^a \delta + \beta_h^b \delta * \theta \tag{10}$$

where δ is the size of the shock and we want to evaluate the IRF at $z_t = \theta$

Using the options LCS() and MARGINS are equivalent

This case is pretty similar to the one of a quadratic term, however now the initial level of the shock $s0$ does not intervene, although the value of the variable z does. In this example we estimate an IRF when the shock interacts with another continuous variable, e, the percent change in US exchange rate, which corresponds to the variable z in Equation (9).

- If $e = 1$, according to Equation (10), we can simply write:
```
. locproj y l.y l(0/3).r, s(n c.n#c.e) sl(2) hor(12)
```
- For any other value of e, we need to use the option `lcs()`. For instance, if we want to evaluate the IRF at $e = 3$:
```
. locproj y l.y l(0/3).r, s(n c.n#c.e) sl(2) hor(12) lcs(n + c.n#c.e*3)
```
- We would obtain exactly the same results using the option `margins` in the following way:
```
. locproj y l.y l(0/3).r, s(n c.n#c.e) sl(2) hor(12) margins mropt(atmeans
at(e=3))
```

Outline

Binary dependent variable

Local projections can also be estimated when the outcome variable is binary. We thus may be interested in the probability of an outcome at some point in the future if there is a shock to a variable today. Then Equation 1 could be redefined as follows:

$$\mathcal{R}_{s\to h} = P[y_{t+h} = 1 | s_t = s_0 + \delta; x_t] - P[y_{t+h} = 1 | s_t = s_0; x_t]; \qquad h = 0, 1, ..., H, \qquad (11)$$

`locproj` can accommodate several commands like `probit`, `logit`, `xtprobit`, `xtlogit`, `oprobit`, `ologit`, etc. Moreover, `locproj` has the option of expressing the IRF as the response of the probability of a positive outcome, exactly as expressed in 10, using the Stata command `margins`.

Using the option MARGINS to obtain marginal effects and to manage interactions with dummy variables

The option `margins` estimates the marginal effect of a unit of our shock variable s on the probability of a positive outcome of our dependent variable. We just need to change the estimation method, for instance to `xtlogit` with fixed effects, and also add the option `margins`:

```
.  locproj y l(0/2).s, margins m(xtlogit) fe
```

We can also interact the shock variable with a dummy variable D. The option margins allow us to estimate a separate IRF for each category of variable D. To do that we need to use the option `mrfvar()`. In this option we need to specify the expansion (either 0 or 1) of the categorical variable that has been interacted with our shock variable:

```
.  locproj y l(0/2).s D#c.l(0/2).s, margins m(xtlogit) fe mrfvar(1.D)
.  locproj y l(0/2).s D#c.l(0/2).s, margins m(xtlogit) fe mrfvar(0.D)
```

Outline

Quantile LPs

One possibility is that a shock may have no visible effects on the average outcome, but it may have considerable impact on the tails of its distribution.

The approach to calculate quantile local projections is parallel to the way local projections are computed at the mean. The only difference is that we are now dealing with a nonlinear model so the marginal effect of a change in the shock has to be evaluated accordingly.

In the case of `locproj` we only need to change the estimation method to `qreg`, and we can also use the IV case, `ivqreg`.

One advantage of `locproj` is that it has been adapted so that we can include lags of the dependent variable and the shock variable automatically even if the estimating method we are using does not allow time-series operators, as is the case with the commands `qreg` and `ivqreg`.

Stata quantile regression commands cannot handle time-series operators

We are going to estimate the IRF of the GDP growth rate to a shock in the monetary policy interest rate r. We want to estimate the IRF for different quantiles of the distribution of our dependent variable, using the quantile regression method `qreg`.

We want our shock variable to have an impact with a one period lag. However, the `qreg` command does not allow the use of time-series operators. Thus we first need to generate the variable r_{t-1} and using it as our shock variable:

```
.  gen lr=l.r
```

With LOCPROJ We can automatically include lags of response and shock variables

`locproj` has been adapted so that we can include lags of the dependent variable and the shock variable by using the options `ylags()` and `slags()` respectively. For instance, to include three lags of y and of r_{t-1} we can just write `yl(3)` and `sl(3)`.

However, if we want to include lags of our control variables, we do need to do it by hand, generating each one of the lagged-terms we want. In our example, we are going to introduce three lags of the variable n:

```
. gen ln=l.n
. gen l2n=l2.n
. gen l3n=l3.n
```

We only need to change the estimation method and define the percentile we want to estimate

We will estimate the IRF for three moments of the variable y distribution: its mean, the 20th percentile, the median, and the 80th percentile. In all cases we use a robust estimator of the variance-covariance matrix. For the average outcome, we use OLS:

```
. locproj y lr n ln l2n l3n, yl(3) sl(3) h(-4/12) save irfn(Mean) r nograph
```

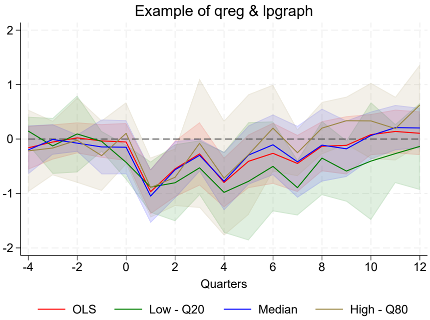
For the other moments of the distribution we use the command `qreg`:

```
. locproj y lr n ln l2n l3n, yl(3) sl(3) h(-4/12) m(qreg) q(20) nograph
save irfn(Q20) vce(r)
. locproj y lr n ln l2n l3n, yl(3) sl(3) h(-4/12) m(qreg) q(50) nograph
save irfn(Q50) vce(r)
. locproj y lr n ln l2n l3n, yl(3) sl(3) h(-4/12) m(qreg) q(80) nograph
save irfn(Q80) vce(r)
```

We can plot all the IRFs together so that we can compare the response at different percentiles

Now we can create one graph with the four IRFs plotted together, while also choosing the color of each one of the IRFs:

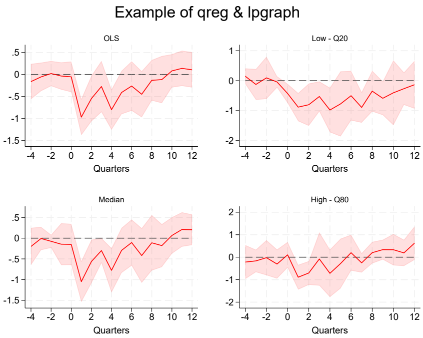
```
.  lpgraph Mean Q20 Q50 Q80, h(-4/12) tti(Quarters) lab1(OLS) lab2(Low - Q20)
lab3(Median) lab4(High - Q80) lc1(red) lc2(green) lc3(blue) lc4(brown)
title(Example of qreg & lpgraph, size(0.9)) z
```



Or we can plot them separately to have a clear look at each one of them

We can also create four individual graphs and then combine them into a single graph. To do so, we need to specify the option separate.

```
.  lpgraph Mean Q20 Q50 Q80, h(-4/12) separate nolegend tti(Quarters)
ti1(OLS) ti2(Low - Q20) ti3(Median) ti4(High - Q80) lcolor(red)
title(Example of qreg & lpgraph, size(0.9)) z
```



References

Jordà, Ò. and Taylor, A. M. (2024).
Local projections.
Federal Reserve Bank of San Francisco Working Paper Series, 2024(24).

Ugarte, A. (2025).
locproj and lpgraph: Stata commands to estimate local projections.
Working Papers 25/09, BBVA Bank, Economic Research Department.

The End