# Impulse Response Function

Xiangyang Li

March 28, 2016

## 1 Definition

### 1.1 Conditional IRF

An "impulse response function" (henceforth IRF) is the change in the current and expected future values of a random variable conditional on the realization of uncertainty at some point in time. Formally:

$$\begin{aligned} IRF_t(j) &= E\left(y_{t+j}\left|\Omega_{t-1}, \epsilon_t \neq 0\right.\right) - E\left(y_{t+j}\left|\Omega_{t-1}, \epsilon_t = 0\right.\right) \\ &= E_t\left(y_{t+j}\right) - E_{t-1}\left(y_{t+j}\right) \end{aligned} \tag{1}$$

where $j = 0, 1, 2, \cdots$, $\epsilon_t$ is exogenous shock. The second equality is just a reduced form of the first equality. $E$ is the mathematical expectation operator and $\Omega_t$ is the information set of time $t$. The IRF, in essence, considers the impact of shock on the expectation of future variables. The IRF in eq(1) is often referred as **conditional IRF** since it conditional on information set $\Omega_{t-1}$ and time $t$ shock $\epsilon_t$.

### 1.2 Linear Examples

Suppose we have ARMA(p,q) process

$$y_t = a + \sum_{j=1}^{p} \rho_j y_{t-j} + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j}$$

where $\{\epsilon_t\}$ is a white noise series. For simple life here, let's assume that $p = q = 1$

$$y_t = \overbrace{a + \rho_1 y_{t-1} + \theta_1 \epsilon_{t-1}}^{\Omega_{t-1}} + \epsilon_t$$

$$\begin{aligned} IRF_t(0) &= \underbrace{E_t\left(y_t\right)}_{y_t} - \underbrace{E_{t-1}\left(y_t\right)}_{a+\rho_1 y_{t-1}+\theta_1\epsilon_{t-1}} \\ &= \epsilon_t \end{aligned}$$

As we can see that the realized shock $\epsilon_{t-1}$ is canceled out and the average effect of symmetric shocks is always 0. The only thing we need care about is the impulsion $\epsilon_t$.

Then one-period-out IRF of $y_t$ is

$$\begin{aligned} IRF_t(1) &= \underbrace{E_t(y_{t+1})}_{a+\rho_1 y_t+\theta_1\epsilon_t} - \underbrace{E_{t-1}(y_{t+1})}_{a+\rho_1 y_t} \\ &= \theta_1\epsilon_t \end{aligned}$$

and in general $IRF_t(2) = \rho_1\theta_1\epsilon_t$, $IRF_t(j) = \rho_1^{j-1}\theta_1\epsilon_t, j > 0$.

## 1.3 Nonlinear Example

Suppose we have 2nd order approximation equation like

$$y_t = \rho y_{t-1} + \alpha y_{t-1}^2 + \epsilon_t$$

then

$$E_t(y_t) - E_{t-1}(y_t) = \epsilon_t$$

One-period-out IRF:

$$y_{t+1} = \rho\left(\rho y_{t-1} + \alpha y_{t-1}^2 + \epsilon_t\right) + \alpha\left(\rho y_{t-1} + \alpha y_{t-1}^2 + \epsilon_t\right)^2 + \epsilon_{t+1}$$

then

$$E_t(y_{t+1}) = \rho\left(\rho y_{t-1} + \alpha y_{t-1}^2 + \epsilon_t\right) + \alpha\left(\left(\rho y_{t-1} + \alpha y_{t-1}^2\right)^2 + \epsilon_t^2 + 2\left(\rho y_{t-1} + \alpha y_{t-1}^2\right)\epsilon_t\right)$$

$$E_{t-1}(y_{t+1}) = \rho\left(\rho y_{t-1} + \alpha y_{t-1}^2\right) + \alpha\left(\left(\rho y_{t-1} + \alpha y_{t-1}^2\right)^2\right)$$

Hence

$$IRF_t(1) = \rho\epsilon_t + \alpha\epsilon_t^2 + 2\alpha\left(\rho y_{t-1} + \alpha y_{t-1}^2\right)\epsilon_t \tag{2}$$

The IRF is much more complicated than the linear example. In the nonlinear example, the one-period-out IRF is a function of $t-1$ information set $\Omega_{t-1}$. In linear case, it is not even a function of $\Omega_{t-1}$ since it is canceled out and become irrelevant.

In practice, it is too hard to compute IRFs by analytic formulas when equations are nonlinear. We need leverage the simulation technology. The following is the algorithm. Let's elaborate the algorithm using the nonlinear example[1]:

1. From a random number generator, draw:

$$\epsilon_{t+1}^{(1)}, \epsilon_{t+2}^{(1)}, \cdots, \epsilon_{t+T}^{(1)}$$

[1]Note that calculation conditioned on $y_t$ or the initial conditions. Hence, we still need specify or fix a value for it before IRF is numerically obtained, for example, the non-stochastic steady state.

2. Using the stochastic equation, $\rho y_{t-1} + \alpha y_{t-1}^2$ and the given $\epsilon_t$ to compute (by pruning)
$$y_{t+1}^{(1)}, y_{t+2}^{(1)}, \cdots, y_{t+T}^{(1)}$$
3. Repeat this, over and over again, $H$ times (usually big), to get
$$y_{t+1}^{(1)}, y_{t+2}^{(1)}, \cdots, y_{t+T}^{(1)}$$
$$\cdots$$
$$y_{t+1}^{(H)}, y_{t+2}^{(H)}, \cdots, y_{t+T}^{(H)}$$
4. Finally,
$$E_t\left(y_{t+j}\right) = \frac{1}{H}\sum_{i=1}^{H} y_{t+j}^{(i)},\ j = 1, 2, \cdots T$$
5. Repeat the preceding steps to compute
$$E_{t-1}\left(y_{t+j}\right) = \frac{1}{H}\sum_{i=1}^{H} y_{t+j}^{(i)},\ j = 1, 2, \cdots T$$
except set $\epsilon_t = 0$.

### 1.4 Unconditional IRF

As we have seen in previous subsections, conditional IRF requires specifying the information set, $\Omega_{t-1}$. As shown in eq(2), $y_{t-1}$ should be specified before we could get $IRF_t(1)$. But how do we choose $y_{t-1}$? The non-stochastic steady state or stochastic mean? This is not commonly discussed in part because with linear examples it is irrelevant. Hence, Unconditional mean comes into play.

From the definition (1), we see that conditional IRF is a function of time $t-1$ information set $\Omega_{t-1}$. From this point, we can say that conditional IRF itself is a random variable. Statistically speaking, we could report its mean as unconditional IRF by dropping some warming periods[2] so as to the initial state dependence will be minimized if not eliminated. This is exactly what Dynare has done. Let's see how Dynare has done to solve this issue.

## 2 Dynare Implementation

### 2.1 The Algorithm

Suppose we have time 0, time $t$ and time $T$ where $T > t$ and $t$ itself is large. For example $t = 100$.

[2]By dropping warming periods, we try to isolate the dependence of our simulation on initial states

1. From a random number generator, draw $H$ sets of shocks but no need to draw $\epsilon_t$ :
$$\epsilon_0^{(1)}, \epsilon_1^{(1)}, \cdots, \epsilon_{t-1}^{(1)}, \epsilon_{t+1}^{(1)}, \epsilon_{t+2}^{(1)}, \cdots, \epsilon_{t+T}^{(1)}$$
$$\cdots$$
$$\epsilon_0^{(H)}, \epsilon_1^{(H)}, \cdots, \epsilon_{t-1}^{(H)}, \epsilon_{t+1}^{(H)}, \epsilon_{t+2}^{(H)}, \cdots, \epsilon_{t+T}^{(H)}$$

2. Using $\epsilon_t = \sigma_\epsilon$ together with above $H$ sets of shocks to compute (by pruning)
$$y_0^{(1,1)}, y_1^{(1,1)}, \cdots, y_{t-1,}^{(1,1)}, y_{t,}^{(1,1)} y_{t+1}^{(1,1)}, y_{t+2}^{(1,1)}, \cdots, y_{t+T}^{(1,1)}$$
$$\cdots$$
$$y_0^{(H,1)}, y_1^{(H,1)}, \cdots, y_{t-1,}^{(H,1)}, y_{t,}^{(H,1)} y_{t+1}^{(H,1)}, y_{t+2}^{(H,1)}, \cdots, y_{t+T}^{(H,1)}$$

3. Using $\epsilon_t = 0$ together with above $H$ sets of shocks to compute (by pruning)
$$y_0^{(1,0)}, y_1^{(1,0)}, \cdots, y_{t-1,}^{(1,0)}, y_{t,}^{(1,0)} y_{t+1}^{(1,0)}, y_{t+2}^{(1,0)}, \cdots, y_{t+T}^{(1,0)}$$
$$\cdots$$
$$y_0^{(H,0)}, y_1^{(H,0)}, \cdots, y_{t-1,}^{(H,0)}, y_{t,}^{(H,0)} y_{t+1}^{(H,0)}, y_{t+2}^{(H,0)}, \cdots, y_{t+T}^{(H,0)}$$

4. Then the first $t$ periods is dropped in each replication or simulation (burn-in).

5. Finally,
$$IRF = \frac{1}{H}\sum_{i=1}^{H}\left(y_{t+j}^{(i,1)} - y_{t+j}^{(i,0)}\right),\ j = 1, 2, \cdots T$$

## 2.2 The Implementation

You instruct Dynare to do the preceding calculation by setting two options in **stoch_simul** command:

- $H$ is set by including the option $replic = H$;
- $T$ is set by including $irf = T$.
- $t$ is set by including $drop = t$

Setting this two options will automatically trig Dynare to compute IRF.The specific steps to implement are:

1. Draw a series of random shocks for 140 periods (usually the white noise)
2. Perform simulation Y1
3. Add one standard deviation to the simulated series for shock e in period 101

4. Perform simulation Y2
5. The IRF for this experiment is equal to Y2-Y1
6. Results obtained in 5) is affected by idiosyncratic shocks other than the one impulse in period 101. In order to average over the effect of these idiosyncratic shocks, we perform 50 replications of steps 1) to 5) and report the average.

There are few points needed to be pointed out for above implementation:

1. Initial 100 warming periods is the default value of option **drop**
2. 40 periods for the IRF is the default value of option **irf**
3. 50 replications is the default value of option **replic**

By following this procedure, it average over the influence of the state point at which the impulse occurs and of future shocks in periods after the impulse. So, Dynare reports indeed an average IRF.

## 3 Customization

Let's look at the IRF.m file in Dynare[3]. We have commented few lines in **for** loop.

```
function y = irf(dr, e1, long, drop, replic, iorder)

% function y = irf(dr, e1, long, drop, replic, iorder)
% Computes impulse response functions
%
% INPUTS
%    dr:     structure of decisions rules for stochastic simulations
%    e1:     exogenous variables value in time 1 after one shock
%    long:   number of periods of simulation
%    drop:   truncation (in order 2)
%    replic: number of replications (in order 2)
%    iorder: first or second order approximation
%
% OUTPUTS
%    y:      impulse response matrix
%
% SPECIAL REQUIREMENTS
%    none

% Copyright (C) 2003-2010 Dynare Team
```

[3]IRF.m lies at like "C:\dynare\4.4.3\matlab" where C is the drive symbol and 4.4.3 is the version of the Dynare in use

```
%
% This file is part of Dynare.
%
% Dynare is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%
% Dynare is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
% GNU General Public License for more details.
%
% You should have received a copy of the GNU General Public License
% along with Dynare.  If not, see <http://www.gnu.org/licenses/>.

global M_ oo_ options_


if M_.maximum_lag >= 1
    temps = repmat(dr.ys,1,M_.maximum_lag);
else
    temps = zeros(M_.endo_nbr, 1); % Dummy values for purely forward models
end
y       = 0;

if iorder == 1
    y1 = repmat(dr.ys,1,long);
    ex2 = zeros(long,M_.exo_nbr);
    ex2(1,:) = e1';
    y2 = simult_(temps,dr,ex2,iorder);
    y = y2(:,M_.maximum_lag+1:end)-y1;
else
    % eliminate shocks with 0 variance
    i_exo_var = setdiff([1:M_.exo_nbr],find(diag(M_.Sigma_e) == 0 ));
    nxs = length(i_exo_var);
    ex1 = zeros(long+drop,M_.exo_nbr);
    ex2 = ex1;
    chol_S = chol(M_.Sigma_e(i_exo_var,i_exo_var));
    for j = 1: replic
        %drawing shocks from normal distriubtion;
        %where chol_S denotes the standard dev. matrix of shocks
        ex1(:,i_exo_var) = randn(long+drop,nxs)*chol_S;
        ex2 = ex1;

        %add one standard deviation at time t;
        ex2(drop+1,:) = ex2(drop+1,:)+e1';

        %simulation using the decision rule and shocks
        y1 = simult_(temps,dr,ex1,iorder);
        y2 = simult_(temps,dr,ex2,iorder);

        %IRFs are added up
        y = y+(y2(:,M_.maximum_lag+drop+1:end)-y1(:,M_.maximum_lag+drop+1:end));
    end
    y=y/replic;
end
```

There are at least two points that we can DIY in calculation IRFs[4].

1. Conduct simulation with bigger shocks. In standard simulation, only one standard deviation is added. Here if you want to simulate with 2 standard deviations, then you could simply modify the codes like:

   ```
   ex2(drop+1,:) = ex2(drop+1,:)+2*e1';
   ```

2. Conduct simulation with negative shocks. Again, in standard simulation, only positive shock is added at time $t+1$. You could simulate your model using negative shocks by simply replace the plus sign with minus sign, like[5]:

   ```
   ex2(drop+1,:) = ex2(drop+1,:) - e1';
   ```

I recommend that you should read the code line by line. This will let you fully understand what is going on behind help you to realize what you want to do.

[4]I strongly recommend that you backup you irf.m file before you modify it.

[5]There is another way to simulate with negative shock $\epsilon_t$: you can use $A_t = \rho A_{t-1} - \epsilon_t$ in your equilibrium conditions.