

Instructions for reproducing figures and tables in “Fiscal Multipliers at the Zero Lower Bound: The Role of Policy Inertia”

by Taisuke Nakata (taisuke.nakata@frb.gov or taisuke.nakata@gmail.com)

Date created: March 31, 2017

1 Overview

In this file, I describe how to generate figures and tables in the paper and the online appendix. In general, to replicate each figure in the main text, find its folder with its respective figure number or table number. Also, one should have a license to use IMSL Fortran library to run Fortran files properly. Fortran codes were compiled using Intel Fortran Compiler in Linux. Matlab codes can be run in both Windows and Linux. All Fortran files should not take any longer than 5 minutes, and most of them will take less than a minute to complete their runs. All MATLAB files will create figures in less than 30 seconds.

Basic Structure of the "Codes" folder:

Each folder is named after its respective figure(s) or table(s). In folders for figures, Fortran files yield .dat files which are used as inputs for MATLAB files to create figures. In folders for tables, Fortran files yield numerical results that are reported in tables. Detailed step-by-step instructions for creating each figure and table are found below in sections 2 and 3.

2 Main text

Figure 1: Policy Inertia and the Government Spending Multiplier

1. Run NL_2SMS_TShock.f90.
2. figure1_10yr.m file will create FMs_NL_2SMS_ZShock_10yr_colorless.eps (figure 1) by reading in the following .dat files, created from running the Fortran codes:
 - (a) irf_G_MODEL_delshock_gshock_NITR_corr_FM_NL_10yr.dat,
irf_G_MODEL_delshock_NITR_corr_FM_NL_10yr.dat,
irf_G_MODEL_gshock_NITR_corr_FM_NL_10yr.dat.

Figure 2: IRFs at the ZLB—with and without policy inertia—

1. Run NL_2SMS_TShock_IRF8Q.f90.

2. figure2.m file will create IRF8Q_colorless.eps by reading in the following .dat files, created from running the Fortran codes:

- (a) irf_G_MODEL_delshock_gshock_NITR_corr_FM_NL_IRF8Q.dat,
irf_G_MODEL_delshock_NITR_corr_FM_NL_IRF8Q.dat,
irf_G_MODEL_gshock_NITR_corr_FM_NL_IRF8Q.dat.

Figure 3: Policy Inertia and Expected ZLB Duration

1. Run NL_2SMS_TShock_EZLBD.f90.
2. figure3.m file will create EZLBD.eps by reading in the following .dat files, created from running the Fortran codes:
 - cRHO_r_EZLBD.dat,
cRHO_r_ShockDur_ZLBD.dat.

Table 2: Multipliers and Policy Inertia: At and Away from the ZLB

1. Run NL_2SMS_TShock_1Qpvgm.f90, NL_2SMS_TShock_4Qpvgm.f90,
and NL_2SMS_TShock_40Qpvgm.f90.
2. PVGM.m file will produce PVGM.mat by reading in the following .dat files, created from running the above Fortran codes:
 - (a) irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_1Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_4Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_40Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_1Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_4Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_40Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_1Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_4Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_40Qpvgm.dat.

Table 3: Multipliers at the ZLB: Alternative Policy Rules

1. First two rows of the table are directly from the first two rows of the table 2.
2. Alternative Inertial Rule ($\rho_r = 0.85$)
 - (a) Run NL_2SMS_TShock_AIR_1Qpvgm.f90, NL_2SMS_TShock_AIR_4Qpvgm.f90,
and NL_2SMS_TShock_AIR_40Qpvgm.f90.

(b) PVGM.m file will produce PVGM.mat by reading in the following .dat files, created from running the above Fortran codes:

- i. irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_AIR_1Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_AIR_4Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_AIR_40Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_AIR_1Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_AIR_4Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_AIR_40Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_AIR_1Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_AIR_4Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_AIR_40Qpvgm.dat.

3. Price-Level Targeting ($\phi_p = 1$)

(a) Run NL_2SMS_TShock_PLT_1Qpvgm.f90, NL_2SMS_TShock_PLT_4Qpvgm.f90, and NL_2SMS_TShock_PLT_40Qpvgm.f90.

(b) PVGM.m file will produce PVGM.mat by reading in the following .dat files, created from running the above Fortran codes:

- i. irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_PLT_1Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_PLT_4Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_PLT_40Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_PLT_1Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_PLT_4Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_PLT_40Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_PLT_1Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_PLT_4Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_PLT_40Qpvgm.dat.

4. Reifschneider-Williams Rule ($\alpha_Z = 1$)

(a) Run NL_2SMS_TShock_RWR_1Qpvgm.f90, NL_2SMS_TShock_RWR_4Qpvgm.f90, and NL_2SMS_TShock_RWR_40Qpvgm.f90.

(b) PVGM.m file will produce PVGM.mat by reading in the following .dat files, created from running the above Fortran codes:

- i. irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_RWR_1Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_RWR_4Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_RWR_40Qpvgm.dat,

irf_G_MODEL_delshock_NITR_e_corr_FM_NL_RWR_1Qpvgm.dat,
 irf_G_MODEL_delshock_NITR_e_corr_FM_NL_RWR_4Qpvgm.dat,
 irf_G_MODEL_delshock_NITR_e_corr_FM_NL_RWR_40Qpvgm.dat,
 irf_G_MODEL_gshock_NITR_e_corr_FM_NL_RWR_1Qpvgm.dat,
 irf_G_MODEL_gshock_NITR_e_corr_FM_NL_RWR_4Qpvgm.dat,
 irf_G_MODEL_gshock_NITR_e_corr_FM_NL_RWR_40Qpvgm.dat.

Table 4: Multipliers at the ZLB: Sensitivity Analyses

1. Alternative Inertial Rule ($\rho_r = 0.85$)

- (a) Run NL_2SMS_TShock_AIR_1Qpvgm.f90, NL_2SMS_TShock_AIR_4Qpvgm.f90, and NL_2SMS_TShock_AIR_40Qpvgm.f90.
- (b) PVGM.m file will produce PVGM.mat by reading in the following .dat files, created from running the above Fortran codes:
 - i. irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_AIR_1Qpvgm.dat,
 irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_AIR_4Qpvgm.dat,
 irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_AIR_40Qpvgm.dat,
 irf_G_MODEL_delshock_NITR_e_corr_FM_NL_AIR_1Qpvgm.dat,
 irf_G_MODEL_delshock_NITR_e_corr_FM_NL_AIR_4Qpvgm.dat,
 irf_G_MODEL_delshock_NITR_e_corr_FM_NL_AIR_40Qpvgm.dat,
 irf_G_MODEL_gshock_NITR_e_corr_FM_NL_AIR_1Qpvgm.dat,
 irf_G_MODEL_gshock_NITR_e_corr_FM_NL_AIR_4Qpvgm.dat,
 irf_G_MODEL_gshock_NITR_e_corr_FM_NL_AIR_40Qpvgm.dat.

2. Price-Level Targeting ($\phi_p = 1$)

- (a) Run NL_2SMS_TShock_PLT_1Qpvgm.f90, NL_2SMS_TShock_PLT_4Qpvgm.f90, and NL_2SMS_TShock_PLT_40Qpvgm.f90.
- (b) PVGM.m file will produce PVGM.mat by reading in the following .dat files, created from running the above Fortran codes:
 - i. irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_PLT_1Qpvgm.dat,
 irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_PLT_4Qpvgm.dat,
 irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_PLT_40Qpvgm.dat,
 irf_G_MODEL_delshock_NITR_e_corr_FM_NL_PLT_1Qpvgm.dat,
 irf_G_MODEL_delshock_NITR_e_corr_FM_NL_PLT_4Qpvgm.dat,
 irf_G_MODEL_delshock_NITR_e_corr_FM_NL_PLT_40Qpvgm.dat,
 irf_G_MODEL_gshock_NITR_e_corr_FM_NL_PLT_1Qpvgm.dat,

irf_G_MODEL_gshock_NITR_e_corr_FM_NL_PLT_4Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_PLT_40Qpvgm.dat.

3. Reifschneider-Williams Rule ($\alpha_Z = 1$)

- (a) Run NL_2SMS_TShock_RWR_1Qpvgm.f90, NL_2SMS_TShock_RWR_4Qpvgm.f90, and NL_2SMS_TShock_RWR_40Qpvgm.f90.
- (b) PVGM.m file will produce PVGM.mat by reading in the following .dat files, created from running the above Fortran codes:
 - i. irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_RWR_1Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_RWR_4Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_RWR_40Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_RWR_1Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_RWR_4Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_RWR_40Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_RWR_1Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_RWR_4Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_RWR_40Qpvgm.dat.

Table 5: Multipliers at the ZLB: Extended Models

1. There are 5 extended models: Model with Consumption Habits, with Price Indexation, with Sticky Wage, with Capital, and with Hand-to-Mouth Households. For each model, find the folder name that corresponds to each model and do the following to compute multipliers in the table:
 - (a) Run NL_2SMS_TShock_1Qpvgm.f90, NL_2SMS_TShock_4Qpvgm.f90, and NL_2SMS_TShock_40Qpvgm.f90.
 - (b) PVGM.m file will produce PVGM.mat by reading in the following .dat files, created from running the above Fortran codes:
 - i. irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_1Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_4Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_40Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_1Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_4Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_40Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_1Qpvgm.dat,

irf_G_MODEL_gshock_NITR_e_corr_FM_NL_4Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_40Qpvgm.dat.

3 Online appendix

Table C.2: Multipliers and Policy Inertia: At and Away from the ZLB

1. Semi-Loglinear Model

- (a) Run NL_2SMS_TShock_1Qpvgm.f90, NL_2SMS_TShock_4Qpvgm.f90,
and NL_2SMS_TShock_40Qpvgm.f90.
- (b) PVGM.m file will produce PVGM.mat by reading in the following .dat files, created
from running the above Fortran codes:
 - i. irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_1Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_4Qpvgm.dat,
irf_G_MODEL_delshock_gshock_NITR_e_corr_FM_NL_40Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_1Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_4Qpvgm.dat,
irf_G_MODEL_delshock_NITR_e_corr_FM_NL_40Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_1Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_4Qpvgm.dat,
irf_G_MODEL_gshock_NITR_e_corr_FM_NL_40Qpvgm.dat.

Figure D.1: IRFs away from the ZLB—with and without policy inertia—

1. Run NL_2SMS_TShock_IRF8Q.f90.
2. figure_awayZLB.m file will create IRF8Q_awayZLB.eps by reading in the following .dat
files, created from running the Fortran codes:
 - (a) irf_G_MODEL_delshock_gshock_NITR_corr_FM_NL_IRF8Q.dat,
irf_G_MODEL_delshock_NITR_corr_FM_NL_IRF8Q.dat,
irf_G_MODEL_gshock_NITR_corr_FM_NL_IRF8Q.dat.

Table E.1: Multipliers at the ZLB: Alternative Policy Rules

1. Same as Table 3. See above.

Figure E.1: IRFs at the ZLB with Three Alternative Policy Rules

1. There are 3 folders in this folder:
 - (a) “AltInertialRule” — stands for Alternative Inertial Rule.
 - (b) “PLT” — stands for Price-Level Targeting.
 - (c) “RWRule” — Reifschneider-Williams (2000) Rule.
2. In each folder, there is one Fortran code
3. Run this code. Then 3 .dat files will be produced
4. IRF_AltRules_extended.m file will create IRF_AltRules_extended.eps by reading in the .dat files in these 3 folders.

Table F.1: Multipliers at the ZLB: Sensitivity Analyses

1. Same as Table 4. See above.

Table G.1: Multipliers at the ZLB: Extended Models

1. Same as Table 5. See above.

Figure H.1: Present Value Multipliers

1. Run NL_2SMS_TShock.f90.
2. figure_extended.m file will create FMs_NL_2SMS_ZShock_extended.eps by reading in the following .dat files, created from running the Fortran codes:
 - (a) irf_G_MODEL_delshock_gshock_NITR_corr_FM_NL.dat,
 irf_G_MODEL_delshock_gshock_NITR_corr_FM_NL_1yr.dat,
 irf_G_MODEL_delshock_gshock_NITR_corr_FM_NL_10yr.dat,
 irf_G_MODEL_delshock_NITR_corr_FM_NL.dat,
 irf_G_MODEL_delshock_NITR_corr_FM_NL_1yr.dat,
 irf_G_MODEL_delshock_NITR_corr_FM_NL_10yr.dat,
 irf_G_MODEL_gshock_NITR_corr_FM_NL.dat,
 irf_G_MODEL_gshock_NITR_corr_FM_NL_1yr.dat,
 irf_G_MODEL_gshock_NITR_corr_FM_NL_10yr.dat.