

Stochastic Debt Sustainability Analysis

Carlo A. Favero*

April 3, 2025

Contents

1	SDSA	2
1.1	The Blanchard Book Model	3
1.1.1	Calibration	3
1.1.2	Simulation	3
1.1.3	A code in R for SDSA using the Blanchard model	5
1.2	An Application to Italian Data	8
1.3	A code in R for SDSA on Italian Data	9
	References	12

*Bocconi University & Innocenzo Gasparini Institute for Economic Research (IGIER) & Centre for Economic Policy Research (CEPR), Bocconi University, Department of Economics via Roentgen 1, 20136 Milano, Italy, carlo.favero@unibocconi.it. Website: <https://mypage.unibocconi.eu/carloambrogiofavero/>.
Lecture Notes for the course Fiscal Macroeconomics 2024/25.

1 SDSA

Consider once again the government’s intertemporal budget constraint. To simulate the future debt-to-GDP ratio, we must complement this equation with the key drivers of debt dynamics: the primary surplus, the average cost of debt financing, the economy’s growth rate, and stock-flow adjustments. Since all these variables are subject to uncertainty, debt dynamics are inherently uncertain as well.

This leads to an operational definition of debt sustainability, as provided in (Blanchard, 2023):

“Debt is sustainable if the probability that debt is on an exploding trajectory n years out is small.”

Debt sustainability, therefore, is a probabilistic concept that must be assessed through **Stochastic Debt Sustainability Analysis (SDSA)**. SDSA simulates the evolution of the debt-to-GDP ratio while accounting for uncertainty, providing the full distribution of projected debt paths over time.

Like any simulation, SDSA can be conducted under different scenarios: a **baseline scenario**, which assumes the continuation of an existing fiscal policy rule, and an **alternative scenario**, where the fiscal policy rule is modified. To implement the analysis, one must:

1. Choose the time horizon n .
2. Specify the equations governing the dynamics of all relevant variables.
3. Compute the full distribution of the debt-to-GDP ratio at each horizon under both scenarios.

Debt sustainability can then be evaluated by assessing the probability that the debt-to-GDP ratio exceeds a chosen upper bound within the given time frame. The simulation under the baseline and the alternate scenarios allows to assess the effect of different policies on debt sustainability.

1.1 The Blanchard Book Model

Blanchard illustrates SDSA in chapter 4 of his book by considering the following system to determine the debt to GDP dynamics :

$$\begin{aligned}
R_t^{av} - g_t &= x_t + u_t \\
x_t &= x_{t-1} + e_t^x, \quad e_t^x \sim N(0, s_x), \quad x_0 = 0.0 \\
u_t &= a_u + e_t^u, \\
s_t &= (1 - c)a_s + c[(R_t^{av} - g_t)b_{t-1}] + e_t^s \quad 0 \leq c \leq 1 \\
b_t &= b_{t-1} + (R_t^{av} - g_t)b_{t-1} - s_t \\
e_t^x &\sim N(0, s_x), \quad e_t^u \sim N(0, s_u), \quad e_t^s \sim N(0, s_s)
\end{aligned}$$

e^s, e^x and e^u are uncorrelated.

The first equation captures the notion that expected future $R_t^{av} - g_t$ is constant in absence of shocks but it may either increase or decrease over time, permanently (because of the component x , which has an infinite memory of all shocks hitting it) or temporarily (because of the component u , which is white noise and it has no memory of all shocks hitting it). The primary surplus is on weighted average of two rules: a constant surplus rule and the debt stabilizing rules, but it is also affected by a shock e_t^s . the baseline and the alternate scenarios are determined by different choices of the weight c . The IGBC is linearized by using the approximation $\frac{1+R_t^{av}}{1+g_t} \approx (1 + R_t^{av} - g_t)$, so that the specification of separates processes for R_t^{av} and g_t is not necessary.

1.1.1 Calibration

The permanent component of the gap between the average cost of financing the debt is initialized at 0, $x_0 = 0$ and the debt to gdp ratio is initialized at 1, $b_0 = 1$.

Assume $s_x = 0.3\%$. The standard deviation of x , n periods ahead, call it $\sigma(x_n)$, is equal to $\sqrt{n} \cdot s_x$. If the horizon of the simulation n is 10 years, then $\sigma(x_n) = 3.3 \cdot s_x = 1\%$ and the probability that the permanent component of $R_t^{av} - g_t, x_t$, increases by 2% or more over the 10 years is equal to 2.5%. Furthermore, $a_u = -0.02$ and $s_u = 1\%, a_s = -0.02$

1.1.2 Simulation

The debt dynamics is simulated 10000 times with an horizon of ten periods and the distribution of difference between the debt to gdp ratio ten-year ahead and the current one is computed, and reported in a figure for different choices of the c parameter. The following

graph is obtai

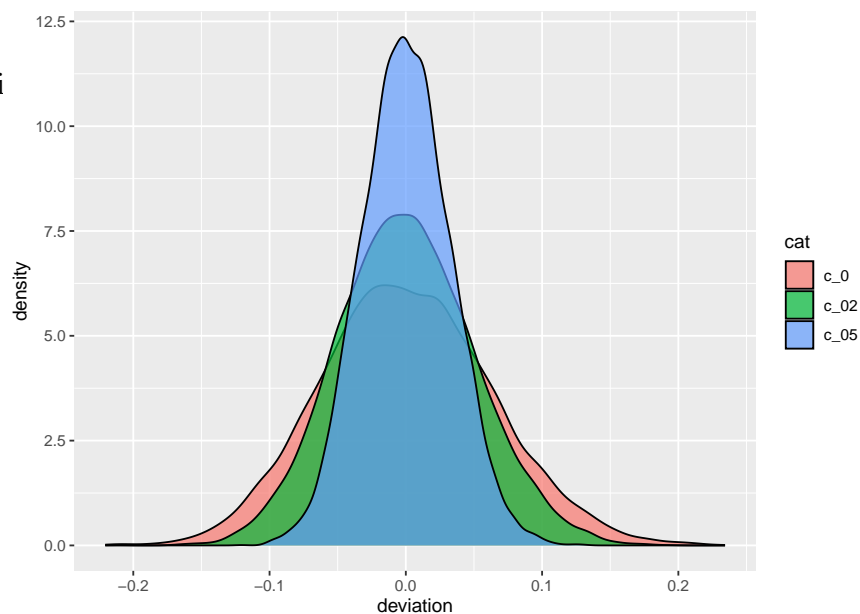


Figure 1: Ten-year ahead distributions of deviations of debt/GDP from the initial value =1

The 98th percentiles of the deviations of ten-year ahead debt/gdp from 1 are respectively 0.133, when $c=0$, 0.106, when $c=0.2$, and 0.667, when $c=0.5$.

Here is a list of potentially **interesting questions**

- What is the distribution of $R_t^{av} - g_t, x_t$ and of its permanent and temporary components ?
- What is the pattern over time of the average debt to gdp ratio ? What explains it ?

1.1.3 A code in R for SDSA using the Blanchard model

```
1 # Clear environment and set working directory
2 rm(list = ls())
3 setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
4
5 # Load necessary package
6 listofpackages <- c("ggplot2","dplyr","readxl","rlist")
7 for (j in listofpackages) {
8   if (sum(installed.packages()[, 1] == j) == 0) {
9     install.packages(j)
10  }
11  library(j, character.only = TRUE)
12 }
13
14
15 ### Blanchard Replication-----
16 num_sim <- 10000
17 n_periods <- 10
18 c_list <- c(0,0.2,0.5)
19 b_mat_list <- list()
20 for (c in c_list){
21   b_mat <- matrix(nrow = num_sim, ncol = n_periods)
22   for (i in 1:num_sim){
23     s_x = 0.003
24     s_u = 0.01
25     s_s = 0.01 #Not given
26     a_u = -0.02
27     a_s = -0.02 #a_u = a_s, implies that under certainty, the debt ratio would
                remain constant.
28
29     e_x = numeric(n_periods)
30     e_u = numeric(n_periods)
31     x = numeric(n_periods)
32     u = numeric(n_periods)
33     rg_diff = numeric(n_periods)
34     s <- numeric(n_periods)
35     b <- numeric(n_periods)
36     d <- numeric(n_periods)
37
38     b[1] = 1
39     x[1] = 0
40     rg_diff[1] = -0.02
41     for (t in 2:n_periods){
42       e_x[t] = rnorm(1,mean = 0, sd = s_x)
43       x[t] = x[t-1] + e_x[t]
44       e_u[t] = rnorm(1, mean = 0, sd = s_u)
```

```

45     u[t] = a_u + e_u[t]
46     rg_diff[t] <- x[t] + u[t]
47     e_s <- rnorm(1, mean = 0, sd = s_s)
48     #s[t] <- a_s + e_s + c*rg_diff[t]*b[t-1]
49     s[t] <- (1-c)*(a_s + e_s) + c*rg_diff[t]*b[t-1]
50     b[t] <- b[t-1] + rg_diff[t]*b[t-1] - s[t]
51   }
52   b_mat[i,] <- b
53 }
54 b_mat_list <- list.append(b_mat_list, b_mat)
55 }
56
57 # Create a date series starting from 2024
58 start_date <- as.Date("2024-01-01")
59 dates <- seq.Date(from = start_date, by = "year", length.out = n_periods)
60
61 plot_list <- list()
62 for (b_mat in b_mat_list){
63   b_mean <- apply(b_mat, 2, mean)
64   b_sd <- apply(b_mat, 2, sd)
65   b_ub <- b_mean + 2*b_sd
66   b_lb <- b_mean - 2*b_sd
67
68   # Create a data frame for plotting
69   data <- data.frame(
70     Date = dates,
71     b_t_mean = b_mean,
72     b_t_lb = b_lb,
73     b_t_ub = b_ub
74   )
75
76   p <- ggplot(data, aes(x = Date)) +
77     geom_ribbon(aes(ymin = b_t_lb, ymax = b_t_ub), fill = "lightblue", alpha =
78       0.5) +
79     geom_line(aes(y = b_t_mean), size = 0.5, col = "blue") +
80     labs(x = "Date", y = "Debt Ratio", title = "Mean Debt Ratio with Upper and
81       Lower Bounds Over Time")
82   plot_list <- list.append(plot_list, p)
83 }
84
85 b_mat <- b_mat_list[[1]]
86 c_0 <- data.frame(deviation = b_mat[,n_periods] - b_mat[,1], cat =
87   rep("c_0", num_sim))
88
89 b_mat <- b_mat_list[[2]]
90 c_02 <- data.frame(deviation = b_mat[,n_periods] - b_mat[,1], cat =
91   rep("c_02", num_sim))

```

```
88 b_mat <- b_mat_list[[3]]
89 c_05 <- data.frame(deviation = b_mat[,n_periods] - b_mat[,1], cat =
    rep("c_05",num_sim))
90 debt_exp <- do.call('rbind', list(c_0,c_02,c_05))
91 ggplot(debt_exp, aes(deviation, fill = cat)) +
92   geom_density(alpha = 0.7)
```

1.2 An Application to Italian Data

The model considered in the previous section is extremely simplified and not calibrated to any real macroeconomic data. A first step toward a more realistic model can be taken by fitting to the data from Italy a version of the model for debt dynamics introduced in the first lectures.

$$g_t = \mu_1 + \epsilon_t^1 \quad (1)$$

$$R_t^{av} = \mu_2 + \rho R_{t-1}^{av} + \epsilon_t^2 \quad (2)$$

$$sfa_t = \mu_3 + \epsilon_t^3 \quad (3)$$

$$d_t = -((R_t^{av} - g_t)/(1 + g_t)) * b_{t-1} \quad (4)$$

$$b_t = \frac{1 + R_t^{av}}{1 + g_t} b_{t-1} + d_t + sfa_t \quad (5)$$

In this case the government follows a debt stabilizing rule without considering the Stock Flow Adjustment. There are three sources of uncertainty: the fluctuations of nominal growth around its mean, the innovations in the autoregressive process for the average cost of financing the debt and the fluctuations of the ratio of SFA to GDP around its mean. Calibration here is implemented by estimating the parameters $\mu_1, \mu_2, \mu_3, \rho$ using a sample of annual Italian data from 1995 to 2022. Once the unknown parameters fitted residuals $\epsilon_t^1, \epsilon_t^2, \epsilon_t^3$ become also available and stochastic simulations are then obtained via the so called bootstrapping methods in the following steps.

- put the three residuals into a matrix
- by randomly extracting rows from the residuals matrix create a sample of the three residuals of the length of the simulation horizon
- generate artificial series for $g_t, R_t^{av}, sfa_t, d_t, b_t$ over the simulation horizons
- replicate the above process n times to obtain the distribution of all variables at all horizons

Differently from the Monte-Carlo methods used in the first illustrative example, the Bootstrap methods does not impose any form on the distribution of residuals. Moreover, the correlation structure of the three residuals is preserved by drawing rows of the matrix of residuals rather than the elements from the three vectors of residuals individually. The following Rcode illustrates the implementation of SDSA using this procedure.

1.3 A code in R for SDSA on Italian Data

```
1 # Clear environment and set working directory
2 rm(list = ls())
3 setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
4
5 # Load necessary package
6 listofpackages <- c("ggplot2","dplyr","readxl","rlist")
7 for (j in listofpackages) {
8   if (sum(installed.packages()[, 1] == j) == 0) {
9     install.packages(j)
10   }
11   library(j, character.only = TRUE)
12 }
13
14
15
16 ### SDSA with bootstrap applied to Italian Data-----
17 IT_DB <- read_xlsx("data/DB.xlsx") %>% filter(MS == "IT (Italy)")
18 IT_DB$Debt_lag <- lag(IT_DB$`Gvt Debt`)
19 IT_DB$Avg_cost <- (IT_DB$`Primary Surplus` - IT_DB$Surplus)/IT_DB$Debt_lag
20 IT_DB$sfa <- IT_DB$SFA/IT_DB$GDP
21 names(IT_DB)[4] <- "Debt"
22 names(IT_DB)[8] <- "Growth"
23 mod_growth <- lm(Growth ~ 1, data = IT_DB)
24 mu_1 <- coef(mod_growth)
25 resid_g <- resid(mod_growth)[-1]
26 mod_r <- lm(Avg_cost ~ lag(Avg_cost), data = IT_DB)
27 resid_r <- resid(mod_r)
28 mu_2 <- coef(mod_r)[1]
29 rho <- coef(mod_r)[2]
30 mod_sfa <- lm(sfa ~ 1, data = IT_DB)
31 mu_3 <- coef(mod_sfa)
32 resid_sfa <- resid(mod_sfa)[c(-1,-2)]
33 e_t <- cbind(resid_g,resid_r,resid_sfa)
34
35
36 num_sim <- 10000
37 n_periods <- 10
38 b_mat <- matrix(nrow = num_sim, ncol = n_periods)
39 for (i in 1:num_sim){
40   # Initialization
41   R_av<- numeric(n_periods)
42   g<- numeric(n_periods)
43   sfa <- numeric(n_periods)
44   b <- numeric(n_periods)
45   d <- numeric(n_periods)
```

```

46   sfa <- numeric(n_periods)
47   disc_b<- numeric(n_periods)
48   bstar <- numeric(n_periods)
49   bstar <- rep(0.6, n_periods)
50
51   # Initial values for b_t and d_t
52   b[1] <- IT_DB$Debt[nrow(IT_DB)]/IT_DB$GDP[nrow(IT_DB)]
53   disc_b[1]<-b[1] # discounted b set equal to b in the first period
54   d[1] <- -0.03    # Initial value for d_1 (could be different)
55   sfa[1] <- 0
56   R_av[1] <- IT_DB$Avg_cost[nrow(IT_DB)]
57   for (t in 2:n_periods) {
58     e_bs <- e_t[sample(1:nrow(e_t),1),]
59     g[t] <- mu_1 + e_bs[1]
60     R_av[t] <- mu_2 + rho*R_av[t-1] + e_bs[2]
61     sfa[t] <- mu_3 + e_bs[3]
62     d[t] <- -((R_av[t] - g[t]) / (1 + g[t])) * b[t-1] #-((R_av[t] - g[t]) / (1 +
        R_av[t])) * b[t-1]
63     b[t] <- ((1 + R_av[t]) / (1 + g[t])) * b[t-1] + d[t] + sfa[t]
64     disc_b[t]<- b[t]*((1 + g[t]) / (1 + R_av[t]))^t
65   }
66   b_mat[i,] <- b
67 }
68
69 b_mean <- apply(b_mat, 2, mean)
70 b_sd <- apply(b_mat,2, sd)
71 b_ub <- b_mean + 2*b_sd
72 b_lb <- b_mean - 2*b_sd
73
74 # Create a date series starting from 2024
75 start_date <- as.Date("2024-01-01")
76 dates <- seq.Date(from = start_date, by = "year", length.out = n_periods)
77
78 # Create a data frame for plotting
79 data <- data.frame(
80   Date = dates,
81   b_t_mean = b_mean,
82   b_t_lb = b_lb,
83   b_t_ub = b_ub
84 )
85
86 ggplot(data, aes(x = Date)) +
87   geom_ribbon(aes(ymin = b_t_lb, ymax = b_t_ub), fill = "lightblue", alpha =
      0.5) +
88   geom_line(aes(y = b_t_mean), size = 0.5, col = "blue") +
89   labs(x = "Date", y = "Debt Ratio", title = "Mean Debt Ratio with Upper and
      Lower Bounds Over Time")

```

```
90
91 debt_exp <- b_mat[,n_periods] - b_mat[,1]
92 hist(debt_exp, freq = FALSE, breaks = 50)
```

References

Blanchard, O. (2023). *Fiscal policy under low interest rates*. MIT Press.