

# Package ‘godley’

May 8, 2026

**Type** Package

**Title** Stock-Flow-Consistent Model Simulator

**Version** 0.2.2

**Maintainer** Elżbieta Jowik <jowik.elzbieta@gmail.com>

## Description

Define, simulate, and validate stock-flow consistent (SFC) macroeconomic models. The godley R package offers tools to dynamically define model structures by adding variables and specifying governing systems of equations. With it, users can analyze how different macroeconomic structures affect key variables, perform parameter sensitivity analyses, introduce policy shocks, and visualize resulting economic scenarios. The accounting structure of SFC models follows the approach outlined in the seminal study by Godley and Lavoie (2007, ISBN:978-1-137-08599-3), ensuring a comprehensive integration of all economic flows and stocks. The algorithms implemented to solve the models are based on methodologies from KinSELLA and O'Shea (2010) <doi:10.2139/ssrn.1729205>, Peressini and Sullivan (1988, ISBN:0-387-96614-5), and contributions by Joao Macalos.

**URL** <https://gamrot.github.io/godley/>

**BugReports** <https://github.com/gamrot/godley/issues>

**License** GPL (>= 3)

**Encoding** UTF-8

**Imports** dplyr, stringr, tidyr, igraph, purrr, vctrs, rlang, rootSolve,  
plotly, magrittr, checkmate, vecsets, lubridate, data.table,  
tibble, visNetwork

**Depends** R (>= 4.1.0)

**RoxygenNote** 7.3.2

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Michał Gamrot [aut, cph],  
Iwo Augustyński [ctb],  
Julian Kacprzak [ctb],  
Elżbieta Jowik [cre, ctb]

Repository CRAN

Date/Publication 2025-02-23 23:30:06 UTC

## Contents

add_equation . . . . .	2
add_equation_single . . . . .	3
add_scenario . . . . .	3
add_shock . . . . .	4
add_variable . . . . .	5
add_variable_single . . . . .	5
change_init . . . . .	6
create_model . . . . .	6
create_sensitivity . . . . .	7
create_shock . . . . .	7
d . . . . .	8
plot_cycles . . . . .	8
plot_simulation . . . . .	9
prepare . . . . .	10
simulate_scenario . . . . .	10

**Index** **12**

---

add_equation	<i>Add equations to the model</i>
--------------	-----------------------------------

---

### Description

Add equations to the model

### Usage

```
add_equation(model, ...)
```

### Arguments

model	SFC model object
...	additional arguments passed to the function.

### Value

updated SFC model object containing added equation

---

add\_equation\_single     *Add single equation to the model*

---

**Description**

helper for add\_equation()

**Usage**

```
add_equation_single(model, equation, hidden = FALSE, desc = "")
```

**Arguments**

model	SFC model object
equation	string equation in format: 'x = y + z - a * b + (c + d) / e + f[-1]'
hidden	logical, indicates if equation should be written as hidden, defaults to FALSE
desc	string equation description

---

add\_scenario     *Add scenario to the model*

---

**Description**

Add scenario to the model

**Usage**

```
add_scenario(
  model,
  name = "expansion",
  origin = "baseline",
  origin_start = NA,
  origin_end = NA,
  shock
)
```

**Arguments**

model	SFC model object
name	string name of scenario, defaults to 'expansion'
origin	string name of origin scenario, from which the new scenario will be created, defaults to 'baseline'
origin_start	numeric period number from origin scenario from which the new scenario will begin
origin_end	numeric period number from origin scenario on which the new scenario will end
shock	shock object from create_shock() and add_shock()

**Value**

updated SFC model object containing added scenario

---

add_shock	<i>Add shock to shock object</i>
-----------	----------------------------------

---

**Description**

Add shock to shock object

**Usage**

```
add_shock(
  shock,
  variable,
  value = NA,
  rate = NA,
  absolute = NA,
  start = NA,
  end = NA,
  desc = ""
)
```

**Arguments**

shock	tibble from create_shock()
variable	string variable name
value	numeric, an explicit value or values for the variable, will be extended with last value
rate	numeric, multiplier to influence the original value of the variable
absolute	numeric, absolute value to influence the original value of the variable
start	numeric or date period number for the shock to take place, defaults to NA
end	numeric or date period number for the shock to take place, defaults to NA
desc	string variable description

**Value**

updated shock object containing added shock

---

add_variable	<i>Add variables to the model</i>
--------------	-----------------------------------

---

**Description**

Add variables to the model

**Usage**

```
add_variable(model, ...)
```

**Arguments**

model	SFC model object
...	additional arguments passed to the function.

**Value**

updated SFC model object containing added variable

---

add_variable_single	<i>Add single variable to the model</i>
---------------------	---

---

**Description**

helper for add\_variable()

**Usage**

```
add_variable_single(model, name, init = NA, desc = "")
```

**Arguments**

model	SFC model object
name	string name for added variable
init	numeric initial value, defaults to 1e-05
desc	string variable description

---

change_init	<i>Change initial value of a variable</i>
-------------	---

---

**Description**

Change initial value of a variable

**Usage**

```
change_init(model, name, value)
```

**Arguments**

model	SFC model object
name	string variable name
value	numeric value that will replace existing initial value

**Value**

updated SFC model object with new variable initial value

---

create_model	<i>Create SFC model object</i>
--------------	--------------------------------

---

**Description**

Create SFC model object

**Usage**

```
create_model(name = "SFC model", template)
```

**Arguments**

name	string name for created SFC model object
template	string name of model template chosen from: 'SIM', 'PC', 'LP', 'REG', 'OPEN', 'BMW', 'BMWK', 'DIS', 'DISINF', 'SIMEX', 'PCEX' or user created SFC model object to be used as a template

**Value**

SFC model object

---

create\_sensitivity      *Create model with sensitivity scenarios*

---

**Description**

Create model with sensitivity scenarios

**Usage**

```
create_sensitivity(model_pass, variable, lower = 0, upper = 1, step = 0.1)
```

**Arguments**

model_pass	SFC model object that will be used as a baseline for sensitivity calculation
variable	string name of the variable that will be used
lower	numeric lower bound value of the variable
upper	numeric upper bound value of the variable
step	numeric step between upper and lower bounds for the variable to take value

**Value**

SFC model object with sensitivity scenarios

---

create\_shock      *Create shock used in add\_scenario().*

---

**Description**

Create shock used in add\_scenario().

**Usage**

```
create_shock()
```

**Value**

shock object

---

d	<i>Calculate 1 order lag difference of a variable in model</i>
---	--

---

**Description**

Calculate 1 order lag difference of a variable in model

**Usage**

d(x)

**Arguments**

x                    variable name

**Details**

this is a special function to be used exclusively in model equation strings e.g. "x = d(y) + z"

**Value**

difference

---

plot_cycles	<i>Network plot of the model</i>
-------------	----------------------------------

---

**Description**

Network plot of the model

**Usage**

plot\_cycles(model, save\_file = NULL)

**Arguments**

model                SFC model object created with [create\\_model](#)  
 save\_file            name and path to save the plot as html file

**Details**

This function creates a representation of a model as a directed graph. Additionally it shows cycles in the model including these with lagged variables. Graph can be saved as html file.

**Value**

visNetwork object

**Examples**

```
model <- godley::create_model(name = "SFC model", template = "BMW")
plot_cycles(model)
```

---

plot_simulation	<i>Plot simulations of multiple variables in multiple scenarios</i>
-----------------	---

---

**Description**

Plot simulations of multiple variables in multiple scenarios

**Usage**

```
plot_simulation(
  model,
  scenario = "baseline",
  take_all = FALSE,
  from = NA,
  to = NA,
  expressions = "Y"
)
```

**Arguments**

model	SFC model object
scenario	vector of strings or single string name of scenario(s) from which take variables values, defaults to 'baseline'
take_all	logical indicating whether all scenarios containing the given scenario name string(s) should be used, defaults to FALSE
from	numeric period number from which the plot should start, defaults to maximum value
to	numeric period number on which the plot should end, defaults to minimum value
expressions	vector of strings or single string name of variable(s) expression(s) to plot, defaults to 'Y'

**Value**

plotly plot

---

prepare	<i>Make initial matrix row for baseline scenario and prepare equations for simulate_scenario()</i>
---------	--

---

**Description**

Make initial matrix row for baseline scenario and prepare equations for simulate\_scenario()

**Usage**

```
prepare(model, verbose = FALSE)
```

**Arguments**

model	SFC model object
verbose	logical to tell if additional model verbose should be displayed

**Value**

verified and prepared SFC model object

---

simulate_scenario	<i>Simulate scenario of SFC model object</i>
-------------------	--

---

**Description**

Simulate scenario of SFC model object

**Usage**

```
simulate_scenario(
  model,
  scenario,
  periods = NA,
  start_date = NA,
  method = "Gauss",
  max_iter = 350,
  tol = 1e-05,
  hidden_tol = 0.1,
  verbose = FALSE
)
```

**Arguments**

<code>model</code>	SFC model object
<code>scenario</code>	vector of strings or single string name of scenario(s) to simulate
<code>periods</code>	numeric total number of rows (periods) in the model, defaults to 100
<code>start_date</code>	character date to begin the simulation in the format "yyyy-mm-dd"
<code>method</code>	string name of method used to find solution chosen from: 'Gauss', 'Newton', defaults to 'Gauss'
<code>max_iter</code>	numeric maximum iterations allowed per period, defaults to 350
<code>tol</code>	numeric tolerance accepted to determine convergence, defaults to 1e-05
<code>hidden_tol</code>	numeric error tolerance to accept the equality of hidden equations, defaults to 0.1.
<code>verbose</code>	logical to tell if additional model verbose should be displayed

**Value**

updated model containing simulated scenario(s)

# Index

`add_equation`, 2  
`add_equation_single`, 3  
`add_scenario`, 3  
`add_shock`, 4  
`add_variable`, 5  
`add_variable_single`, 5

`change_init`, 6  
`create_model`, 6, 8  
`create_sensitivity`, 7  
`create_shock`, 7

`d`, 8

`plot_cycles`, 8  
`plot_simulation`, 9  
`prepare`, 10

`simulate_scenario`, 10