



Phydra
import phydra

[Library]

XSO

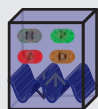
import xso

[Framework]

Models



Chemostat



Slab

Components

State variables



Fluxes



Forcings



Component

← **build**

Variable types

@xso.component
class Component:

```
var1 = xso.variable(...)
var2 = xso.variable(...)
par = xso.parameter(...)
fx = xso.forcing(setup_func='fx_setup')
```

```
def fx_setup(self, ...):
    return forcing
```

```
@xso.flux
def flux_func(self, var1, var2, par, fx):
    return var1 * var2 + par / fx
```

State Variable

Parameter

Forcing

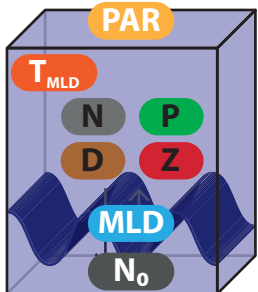
setup function

Flux

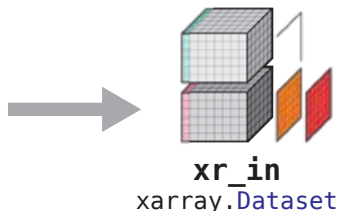
flux function

1 Create or adapt model object
slab_npzd = xso.create({'*components'})

4 Store output
xr_out.to_netcdf()

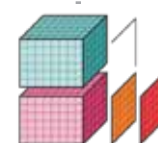


slab_npzd
xs.Model



xr_in
xarray.Dataset

3 Run model
xr_out = xr_in.xsimlab.run(
 model=slab_npzd)



xr_out
xarray.Dataset

2 Setup model & choose solver
xr_in = xso.setup(
 model=slab_npzd, solver, time,
 *input_vars, *output_vars)

5 Analyse & visualise output
xr_out.P_value.plot()
xarray matplotlib

