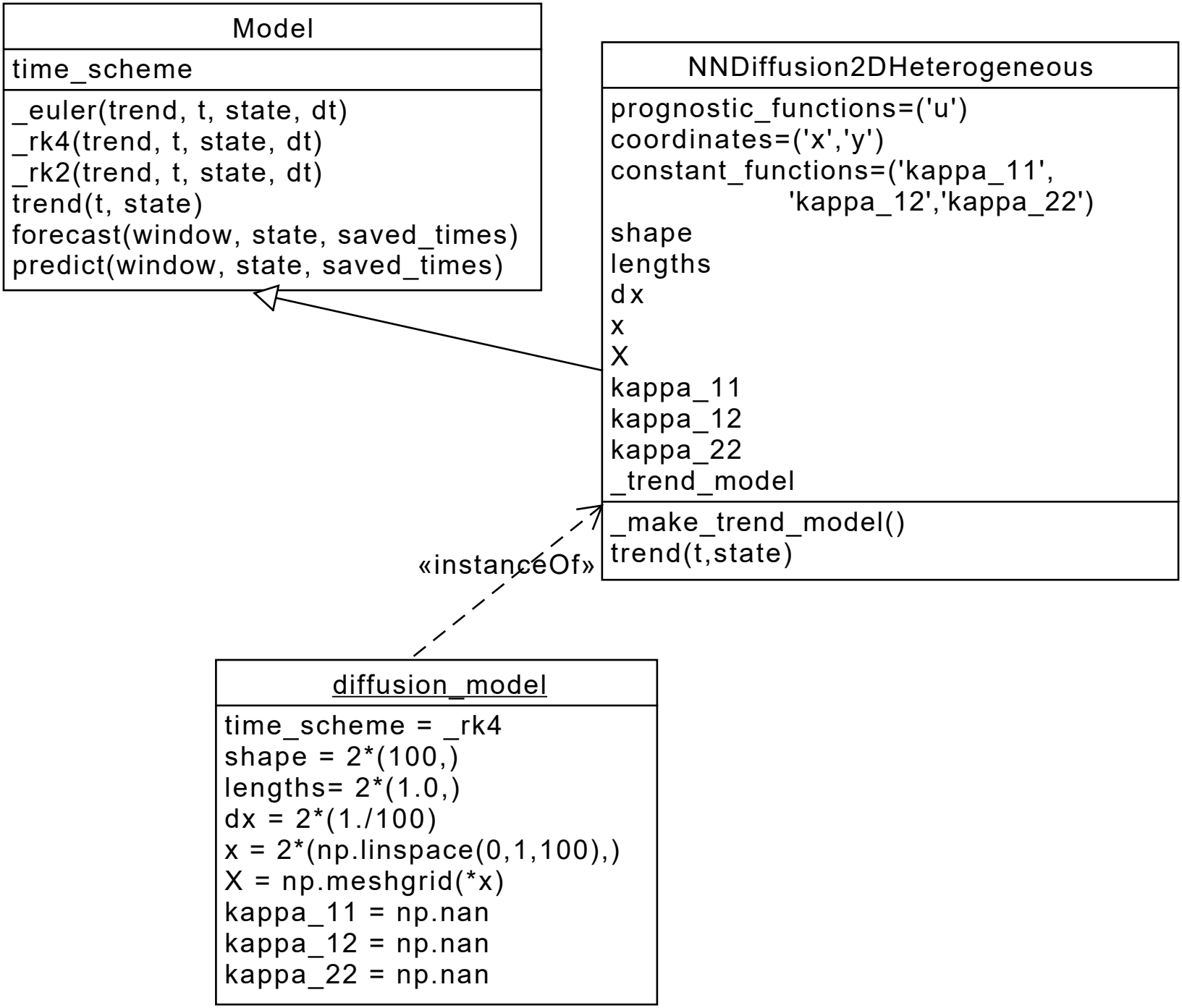
Model
time_scheme
_euler(trend, t, state, dt)
_rk4(trend, t, state, dt)
_rk2(trend, t, state, dt)
trend(t, state)
forecast(window, state, saved_times)
predict(window, state, saved_times)
NNDiffusion2DHeterogeneous
prognostic_functions=('u')
coordinates=('x','y')
constant_functions=('kappa_11',
'kappa_12','kappa_22')
shape
lengths
dx
x
X
kappa_11
kappa_12
kappa_22
_trend_model
_make_trend_model()
trend(t,state)
«instanceOf»
diffusion_model
time_scheme = _rk4
shape = 2*(100,)
lengths= 2*(1.0,)
dx = 2*(1./100)
x = 2*(np.linspace(0,1,100),)
X = np.meshgrid(*x)
kappa_11 = np.nan
kappa_12 = np.nan
kappa_22 = np.nan