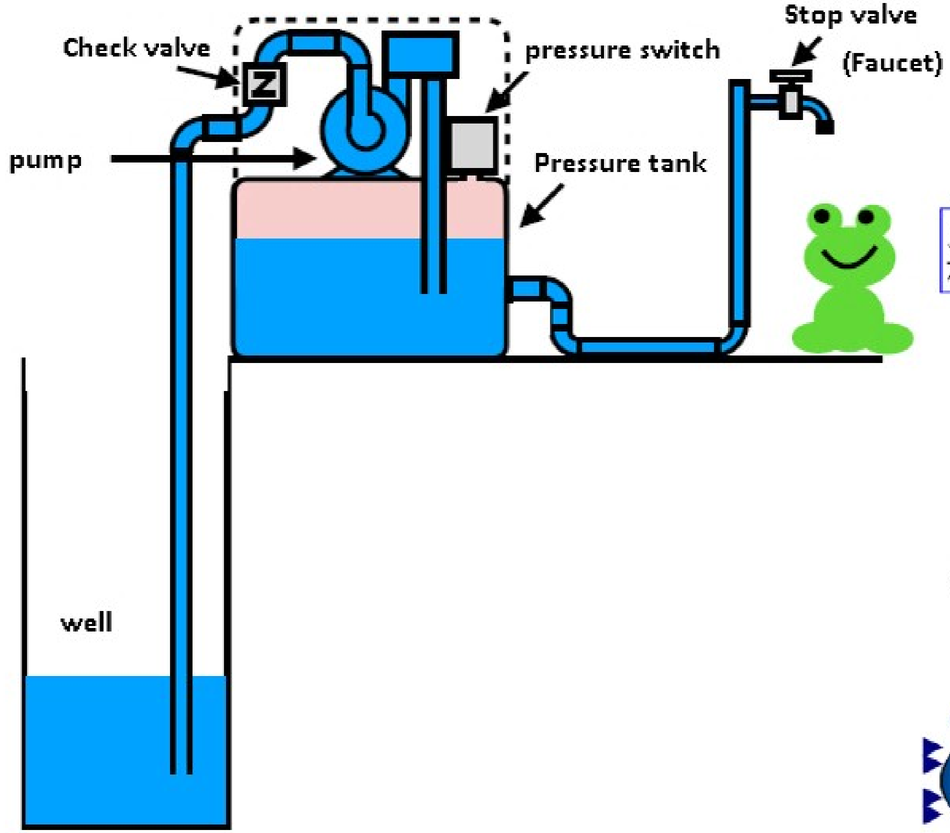
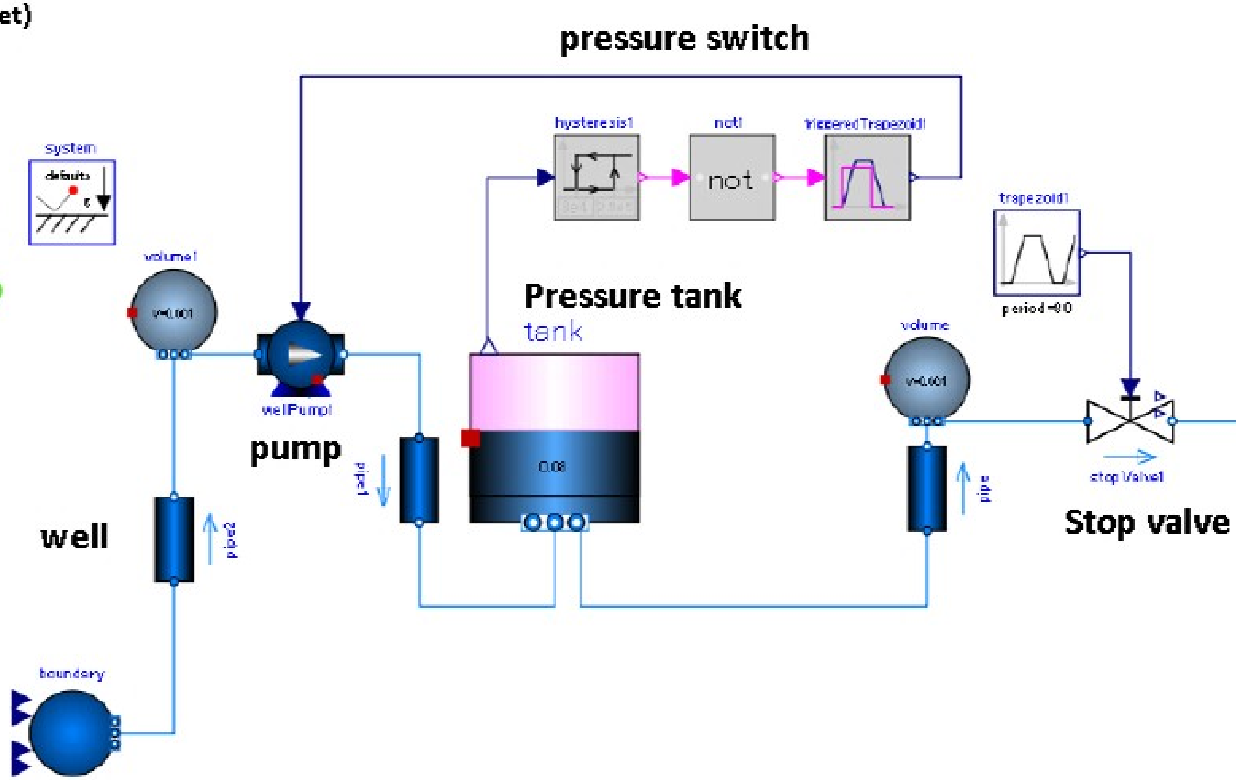
Hands-on
OpenModelica
ter supply system model for shallow well pum





2019/12/14 16th Modellca Library Japanese Stud

Session finback

Well pump water supply system model (1)

We will model a pressure tank type shallow well pump and prototype a model that pumps well water.

- Pump
- Stop valve (faucet)
- Pressure tank
- Pressure switch (pump on / off control device)
- Piping systems (wells, pipes)

This combination is modeled.

Part 1 models the **pump** and **stop valve.**

参考資料: Modelica.Fluid.Valves About liquid valve models
https://www.amane.to/archives/442

Modelica.Fluid.Machines Centrifugal pump model
https://www.amane.to/archives/377

Today's schedule

Shallow well pump water supply system

Stop valve

- StopValve Creating a model
- StopValveTest1 Unit test
- StopValveTest2 Stop valve and open tank

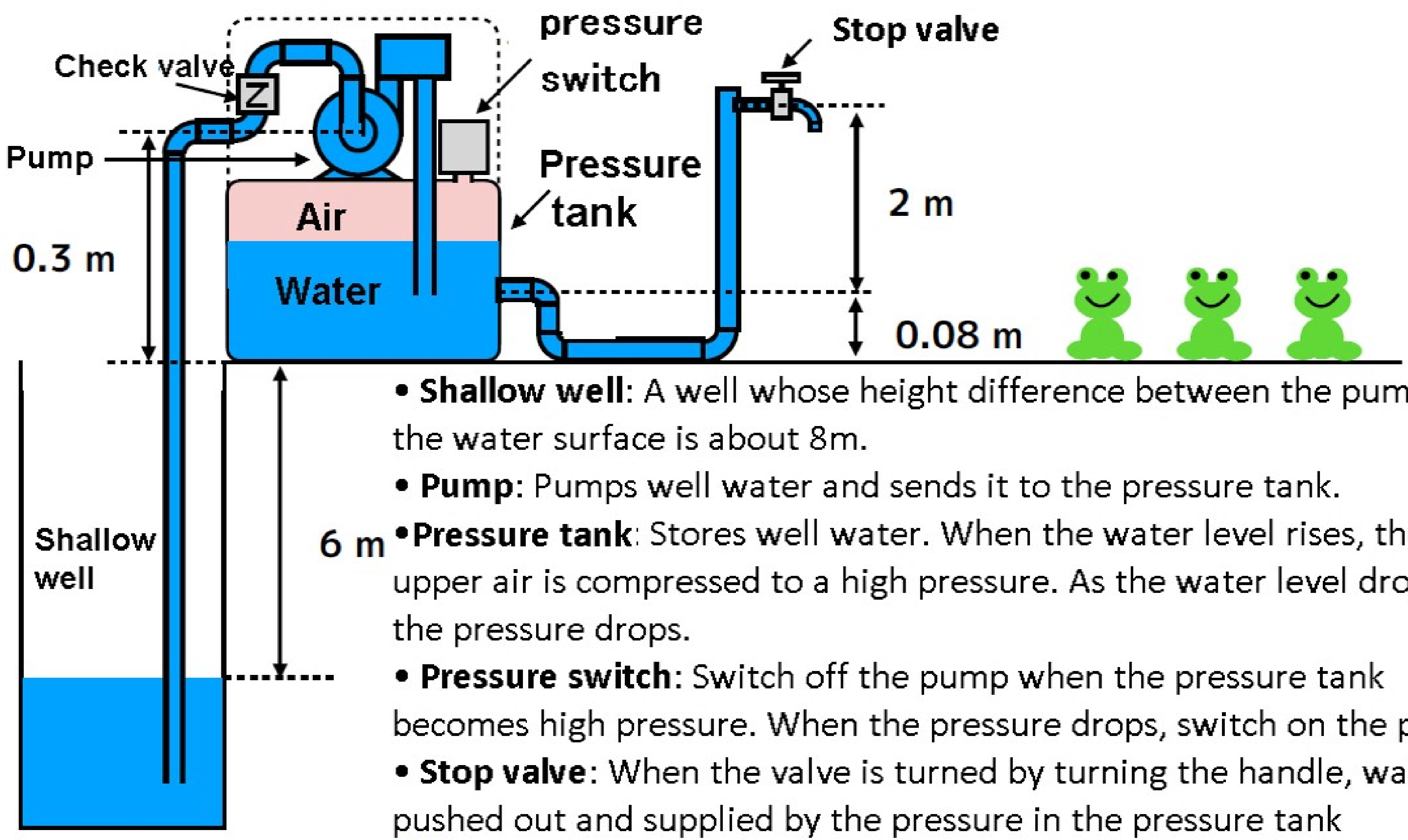
Pump

- (head curve)
- (power curve)

- WellPump Creating a model
- WellPumpTest1 Unit test
- WellPumpTest2 Pump + open tank + stop valve

Conclusion

(Shallow well pumping system)



- **Shallow well**: A well whose height difference between the pump and the water surface is about 8m.
- **Pump**: Pumps well water and sends it to the pressure tank.
- **Pressure tank**: Stores well water. When the water level rises, the upper air is compressed to a high pressure. As the water level drops, the pressure drops.
- **Pressure switch**: Switch off the pump when the pressure tank becomes high pressure. When the pressure drops, switch on the pump.
- **Stop valve**: When the valve is turned by turning the handle, water is pushed out and supplied by the pressure in the pressure tank

Reference https://as76.net/life/a_tank_p.php など

(Stop Valve)

Water tap valve

Reference Materials https://www.seikatu-cb.com/suidou/sknow.html

LLC Wright https://www.walight.jp/2017/07/01/ Saving water-same thing when fully opened /

Environment ministry https://www.mhlw.go.jp/topics/bukyoku/kenkou/suido/jouhou/kankyou/dl/090729-1d.pdf

(The original story is the Tokyo Waterworks Bureau "Environmental Report (2007 version)")

Pressure difference	0.1	Mpa	100000	Pa	Δp
Max. flow rate	22.1	L/min	0.000368333	m3/s	q
Density	998.233	kg/m3	998.233	kg/m3	ρ
Av			3.68008E-05	m2	A_v
Mass flow			0.367682488	kg/s	$\dot{m}$

$$A_v = \frac{q}{\sqrt{\frac{\Delta p}{\rho}}}$$

$$\dot{m} = \rho q$$

Flow rate traced from the figure in the reference material

Please refer at your own risk!

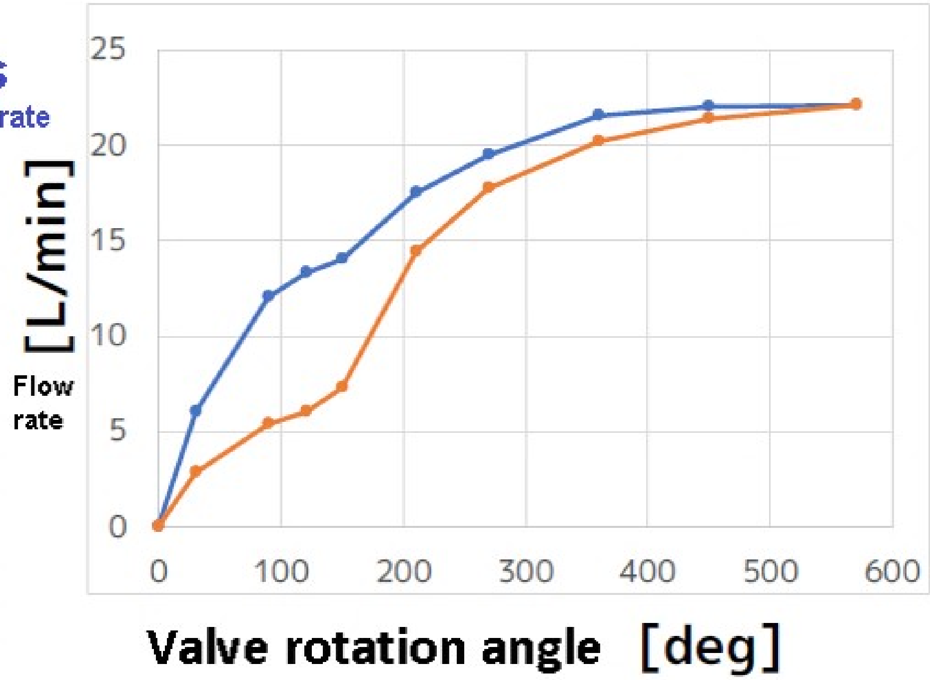
Angle and flow

Angle	Usual	Water saving
deg	L/min	L/min
0	0	0
30	6	2.8
90	12	5.4
120	13.3	6
150	14	7.3
210	17.5	14.4
270	19.5	17.7
360	21.5	20.2
450	22	21.4
570	22.1	22.1

Inherent flow characteristics

Normalized by the maximum value of angle and flow rate

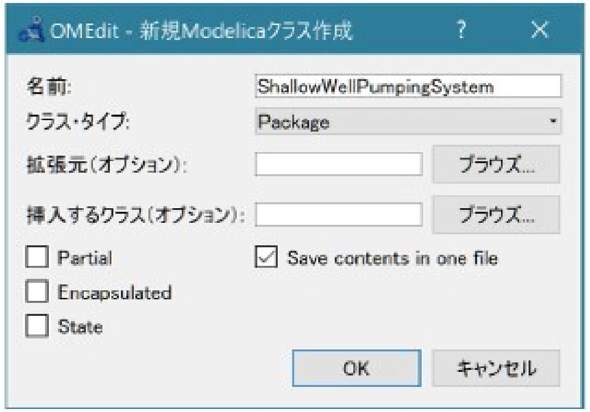
Opening	Usual	Water saving
0	0	0
0.05263158	0.27149321	0.12669683
0.15789474	0.54298643	0.24434389
0.21052632	0.60180995	0.27149321
0.26315789	0.63348416	0.33031674
0.36842105	0.7918552	0.65158371
0.47368421	0.88235294	0.80090498
0.63157895	0.97285068	0.91402715
0.78947368	0.99547511	0.96832579
1	1	1



StopValve Creating a model

Create a StopVale by duplicating Modelica.Fluid.Valves.ValveIncompressible (valve model for liquids) and modify it to fit the specific flow characteristics in CombiTable1D.

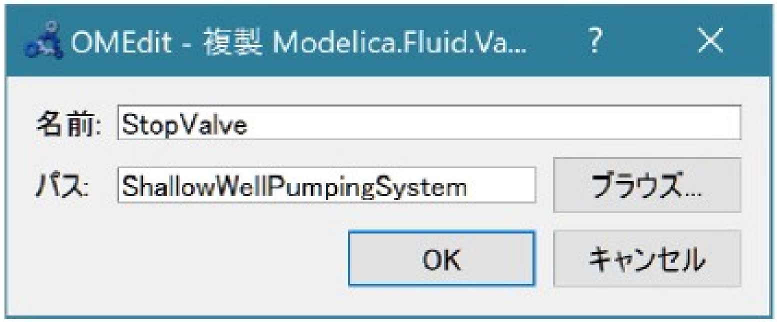
① File> New Modelica Class



Name: ShallowWellPumpingSystem

Class type: Package

② Right-click Modelica.Fluid.Valves.ValveIncompressible in the library browser and select "Duplicate".

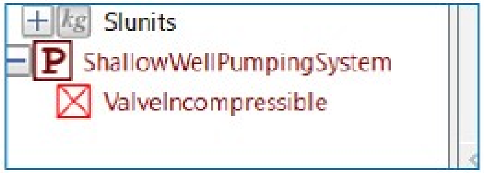


Name: StopValve

Path : ShallowWellPumpingSystem

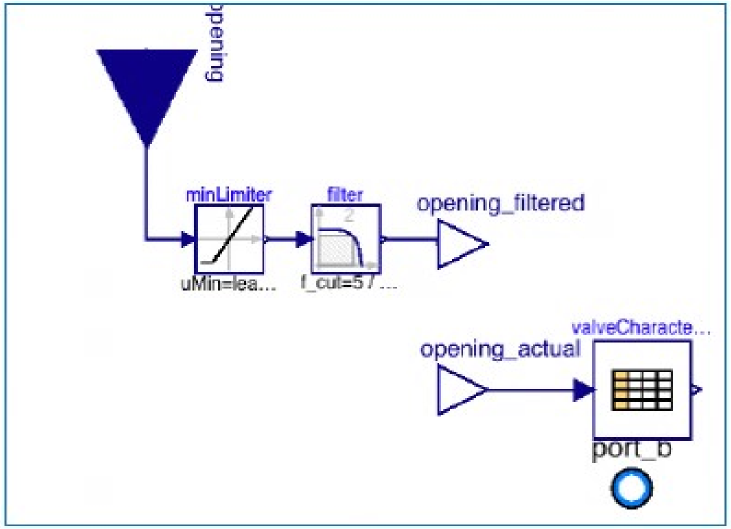
③ Edit the source code of StopValve in text view.

```
model StopValve "Valve for (almost) incompressible fluids"
  extends Modelica.Fluid.Valves.BaseClasses.PartialValve(
    CvData = Modelica.Fluid.Types.CvTypes.Av,
    Av = 3.68008e-5,
    dp_nominal = 1e5,
    m_flow_nominal = 0.3);
  import Modelica.Fluid.Types.CvTypes;
  import Modelica.Constants.pi;
  import SI = Modelica.SIunits;
  import Modelica.Fluid.Utilities;
  constant SI.ReynoldsNumber Re_turbulent = 4000 "cf. straight pipe for fully open valve --
dp_turbulent increases for closing valve” ;
```



In the state where it is duplicated, a red X is attached in the library browser, but it can be corrected by adjusting the scope with extends... or import statement.

④ Paste Modelica.Blocks.Tables.CombiTable1D in the diagram view. Change the name to valveCharacteristicTable.



⑤ Connect the opening_actual and valveCharacteristicTable inputs.

⑥ In text view

valveCharacteristicTable Then, the specific flow characteristic data of the normal frame is set. Set the completion method to smoothness.

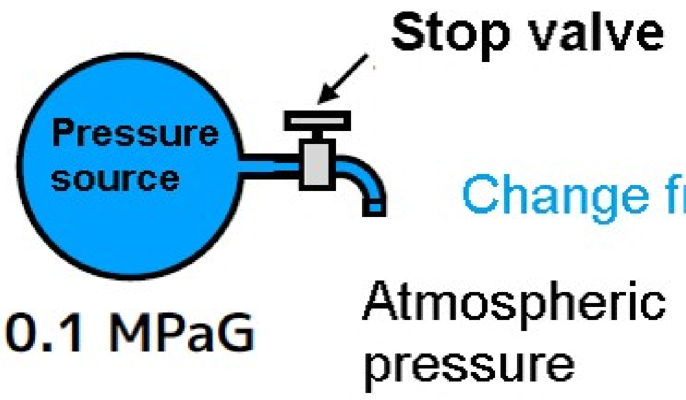
```
  Modelica.Blocks.Tables.CombiTable1D valveCharacteristicTable(
    smoothness = Modelica.Blocks.Types.Smoothness.MonotoneContinuousDerivative1,
    table = [
        0, 0; 0.052631579, 0.271493213; 0.157894737, 0.542986425;
        0.210526316, 0.601809955; 0.263157895, 0.633484163; 0.368421053, 0.791855204;
        0.473684211, 0.882352941; 0.631578947, 0.972850679; 0.789473684, 0.995475113; 1, 1]
    ) annotation( …);
```

⑦ Modify equation to use valveCharacteristicTable.

```
equation
  connect(valveCharacteristicTable.u[1], opening_actual) annotation( …);
// m_flow = valveCharacteristic(opening)*Av*sqrt(d)*sqrt(dp);
// relativeFlowCoefficient = valveCharacteristic(opening_actual);
  relativeFlowCoefficient = valveCharacteristicTable.y[1];
  if checkValve then
```

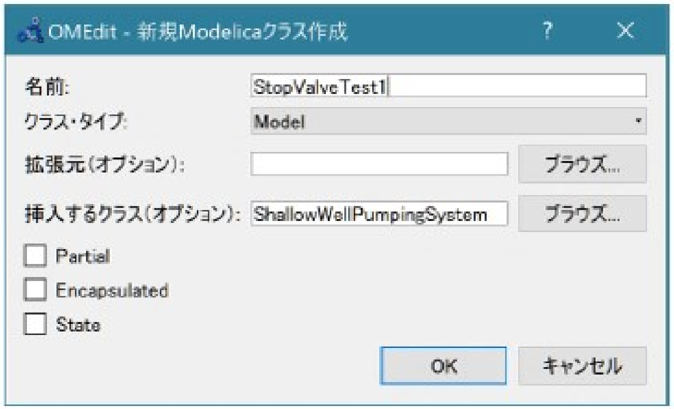
StopValveTest1 Unit test

Create a model to test the operation of the stop valve



Change from fully closed to fully open in 10 seconds.

① **Right-click ShallowWellPumpingSystem in the library browser, select "New Modelica Class", and create StopValveTest1.**

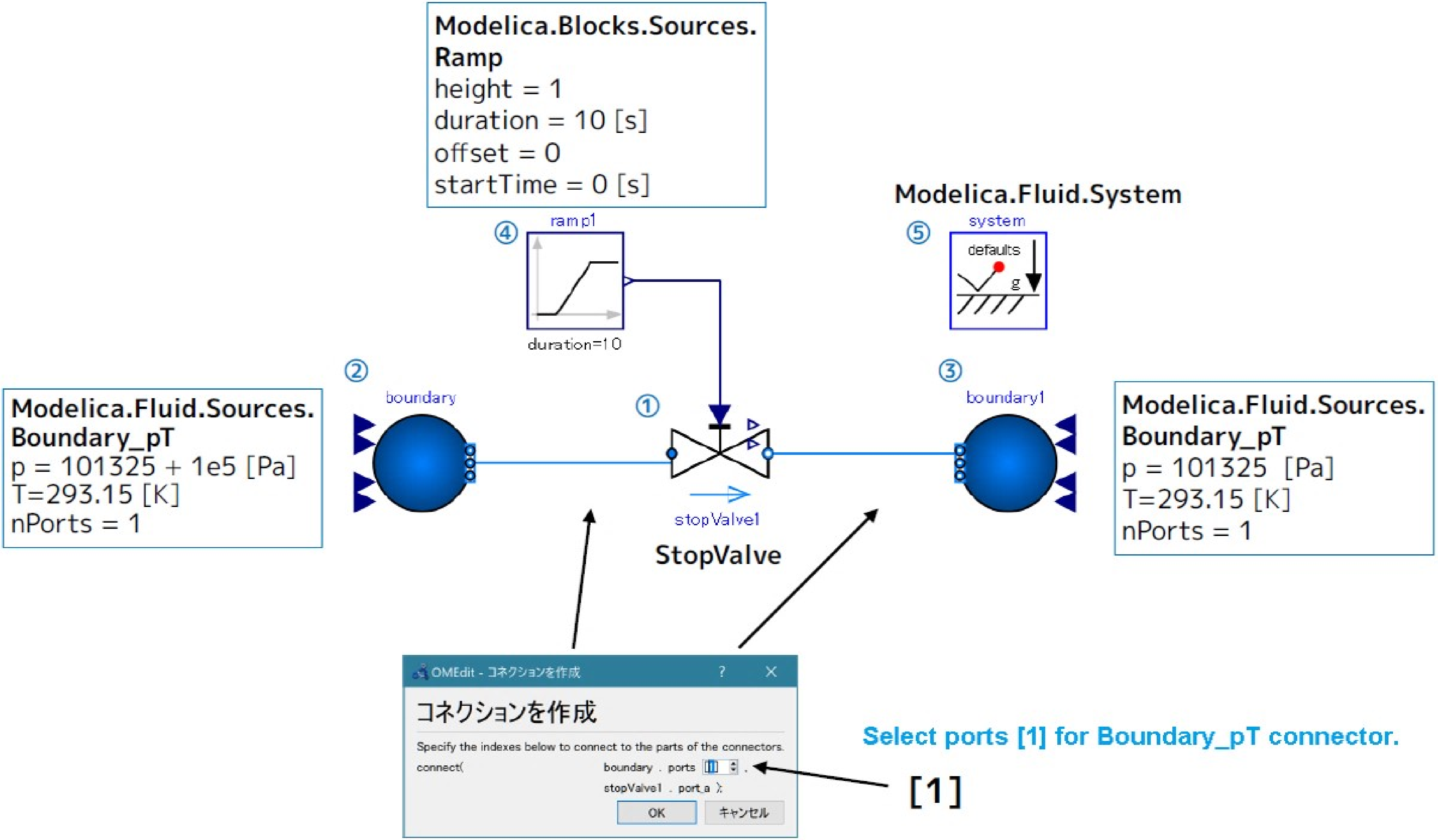


Name: StopValveTest1

Class type : Model

The class to insert : ShallowWellPumpingSystem

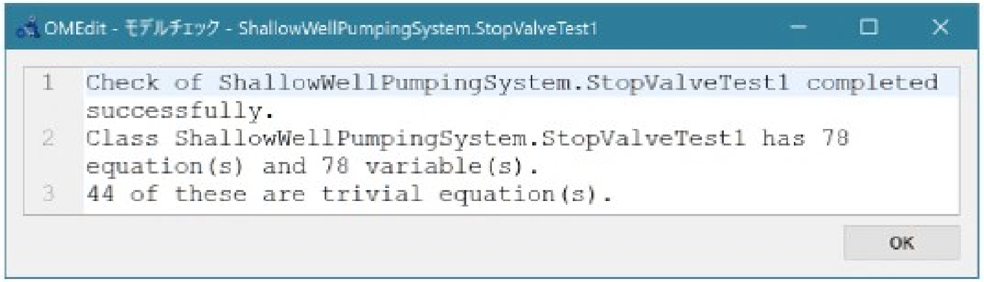
② Drag and drop the part model from the library browser to the diagram view and set the parameters.



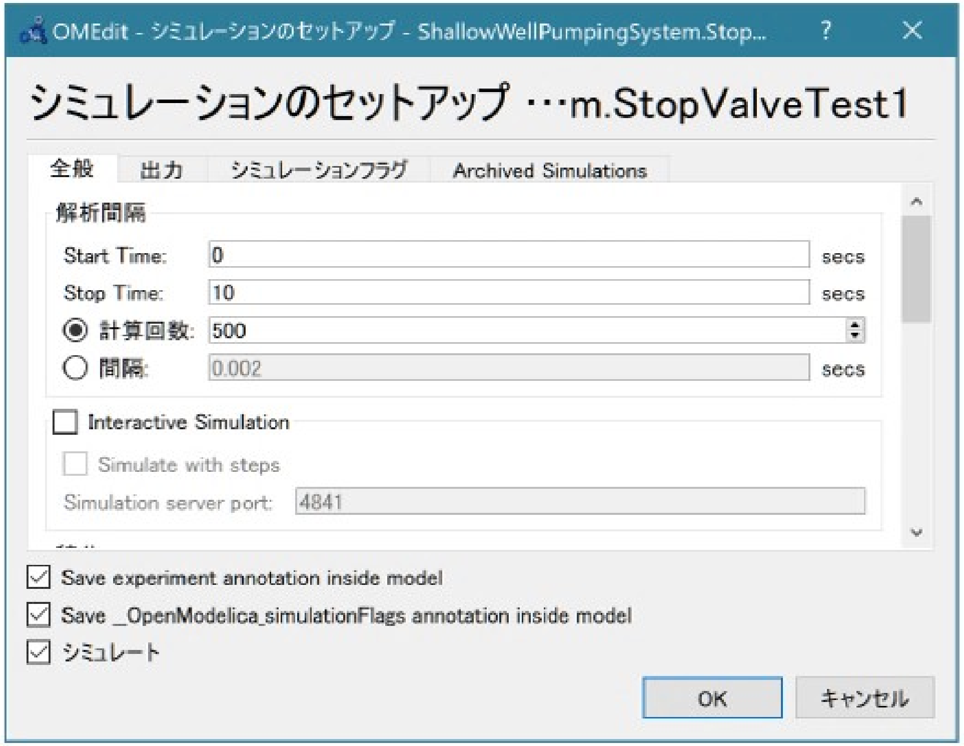
③Set the Media (fluid model) in the text view

```
model StopValveTest1
  replaceable package Medium = Modelica.Media.Water.StandardWater;
  StopValve stopValve1(redeclare package Medium = Medium) annotation( ⋯);          ①
  Modelica.Fluid.Sources.Boundary_pT boundary(                                      ②
    redeclare package Medium = Medium, T = 293.15, nPorts = 1, p = 101325 + 1e5) annotation( ⋯);
  Modelica.Fluid.Sources.Boundary_pT boundary1(                                     ③
    redeclare package Medium = Medium, nPorts = 1, p = 101325) annotation( ⋯);
  Modelica.Blocks.Sources.Ramp ramp1(duration = 10) annotation( ⋯);                ④
  inner Modelica.Fluid.System system annotation( ⋯);                               ⑤
equation
  connect(ramp1.y, stopValve1.opening) annotation( ⋯);
  connect(stopValve1.port_b, boundary1.ports[1]) annotation( ⋯);
  connect(boundary.ports[1], stopValve1.port_a) annotation( ⋯);
  annotation( ⋯);
end StopValveTest1;
```

④ Simulation> Model Check

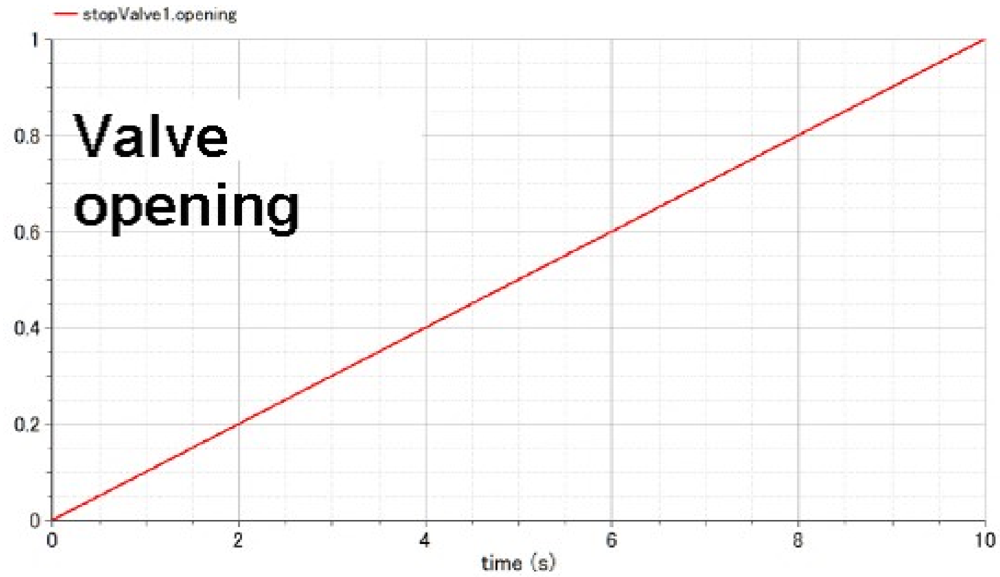


⑤ Simulation> Simulation setup

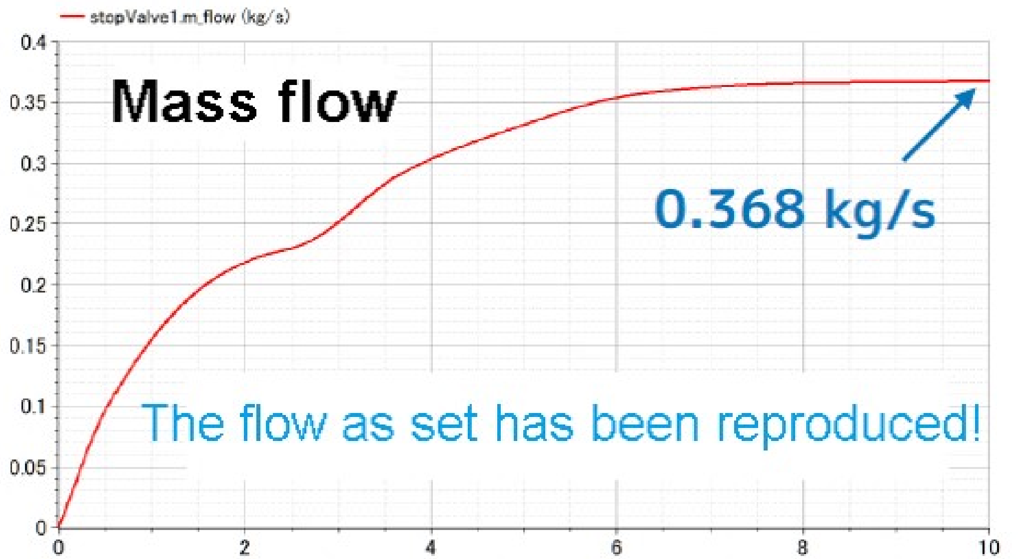


Stop Time = 10 [secs]

Simulation result



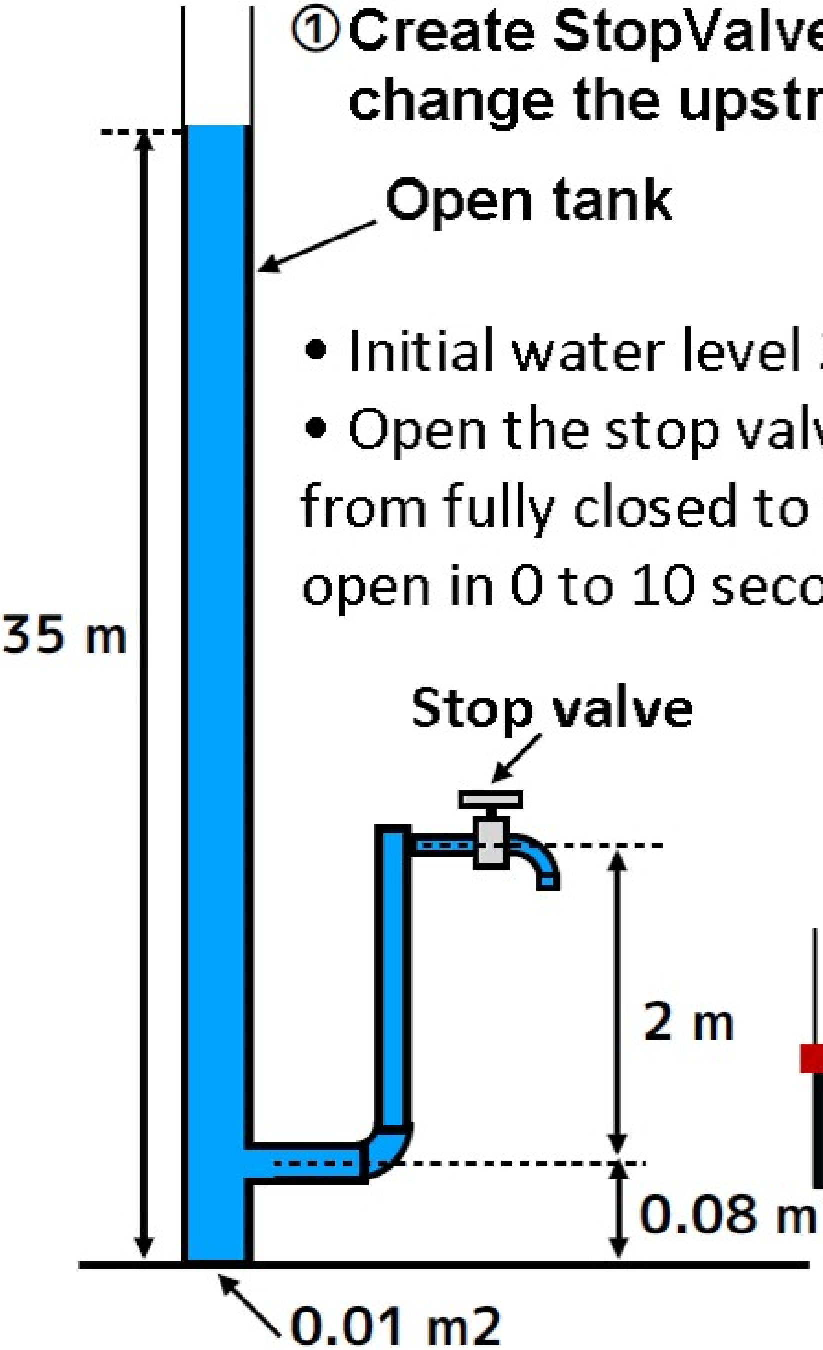
time



time

StopValveTest2 Stop valve and open tank

①Create StopValveTest2 by duplicating StopValveTest1 and change the upstream parts as follows.

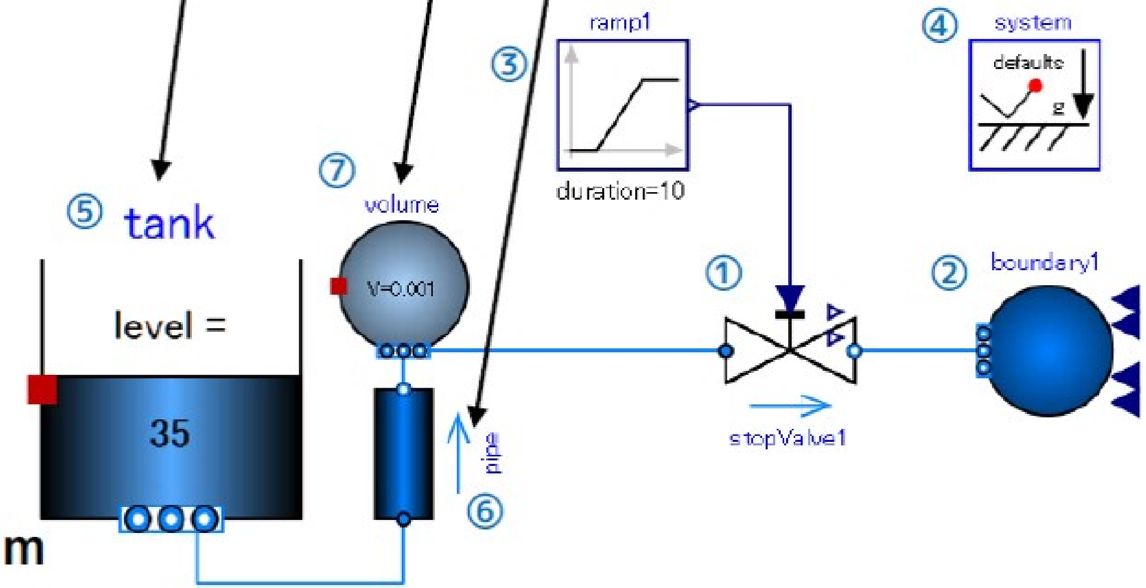


- Initial water level 35 m
- Open the stop valve from fully closed to fully open in 0 to 10 seconds

tank
Modelica.Fluid.Vessels.OpenTank

volume
Modelica.Fluid.Vessels.ClosedVolume

pipe
Modelica.Fluid.Pipes.StaticPipe



② Set the parameters of the changed part.

tank

height = 40 [m]
crossArea = 0.01 [m2]
nPorts = 1
use_portsData = true
portsData ={Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(diameter = 0.015, height = 0.08)}
energyDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial
massDynamics =Modelica.Fluid.Types.Dynamics.FixedInitial
level_start = 35 [m]

volume

V = 0.001 [m3]
nPorts = 2
use_portsData = false
energyDynamics = Modelia.Fluid.Types.Dynamics.FixedInitial
massDynamics = Modelia.Fluid.Types.Dynamics.FixedInitial

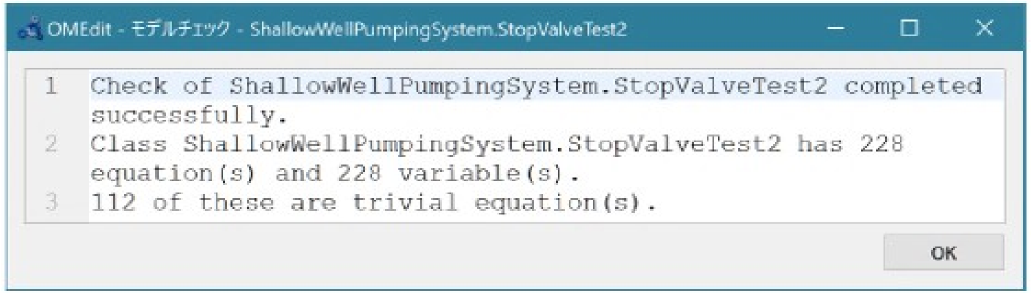
pipe

length = 2 [m]
diameter = 0.015 [m]
height_ab = 2 [m]

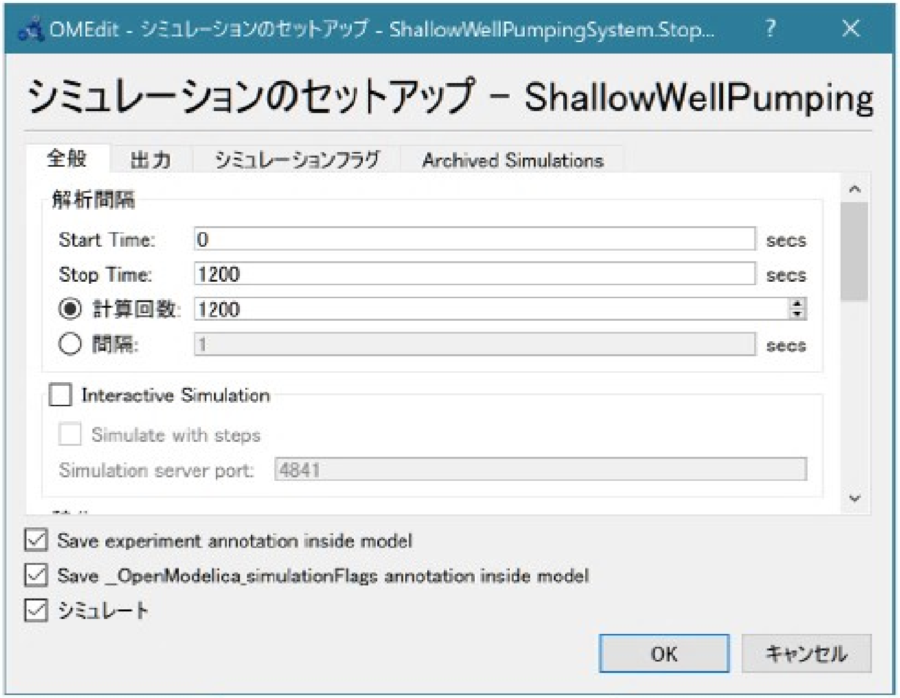
③ Set Media (fluid model) in text view.

```
model StopValveTest2
  replaceable package Medium = Modelica.Media.Water.StandardWater;
  ShallowWellPumpingSystem.StopValve stopValve1(redeclare package Medium = Medium) annotation( ⋯);   ①
  Modelica.Fluid.Sources.Boundary_pT boundary1(                                                      ②
    redeclare package Medium = Medium, T = 293.15, nPorts = 1, p = 101325) annotation( ⋯);
   Modelica.Blocks.Sources.Ramp ramp1(                                                               ③
     duration = 10, height = 1, offset = 0, startTime = 0) annotation( ⋯);
  inner Modelica.Fluid.System system annotation( ⋯);                                                 ④
  Modelica.Fluid.Vessels.OpenTank tank(                                                              ⑤
    redeclare package Medium = Medium, crossArea = 0.01,
    energyDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial,
    height = 40, level_start = 35,
    massDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial,
    nPorts = 1,
    portsData = {Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(diameter = 0.015, height = 0.08)},
    use_portsData = true)  annotation( ⋯);
  Modelica.Fluid.Pipes.StaticPipe pipe(                                                              ⑥
    redeclare package Medium = Medium, diameter = 0.015, height_ab = 2, length = 2)  annotation( ⋯);
  Modelica.Fluid.Vessels.ClosedVolume volume(                                                        ⑦
    redeclare package Medium = Medium,
    V = 0.001,energyDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial,
    massDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial, nPorts = 2, use_portsData = false)
annotation( ⋯);
equation
  connect(pipe.port_b, volume.ports[1]) annotation( ⋯);
  connect(volume.ports[2], stopValve1.port_a) annotation( ⋯);
  connect(tank.ports[1], pipe.port_a) annotation( ⋯);
  connect(ramp1.y, stopValve1.opening) annotation( ⋯);
   connect(stopValve1.port_b, boundary1.ports[1]) annotation(  ⋯);
  annotation( ⋯);
end StopValveTest2;
```

④ Simulation> Model Check



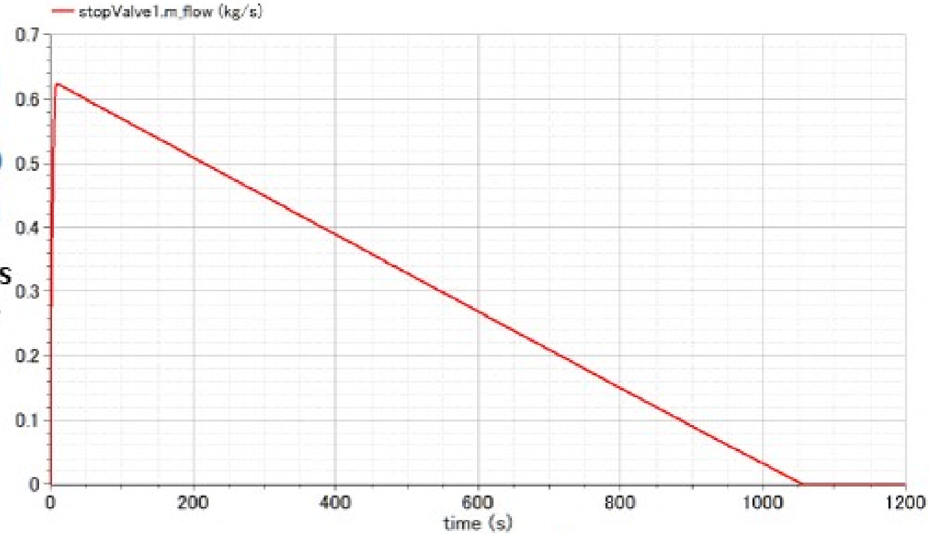
⑤ Simulation> Simulation setup



Start Time: 0 [s]
Stop Time: 1200 [s]
Number of calculation : 1200

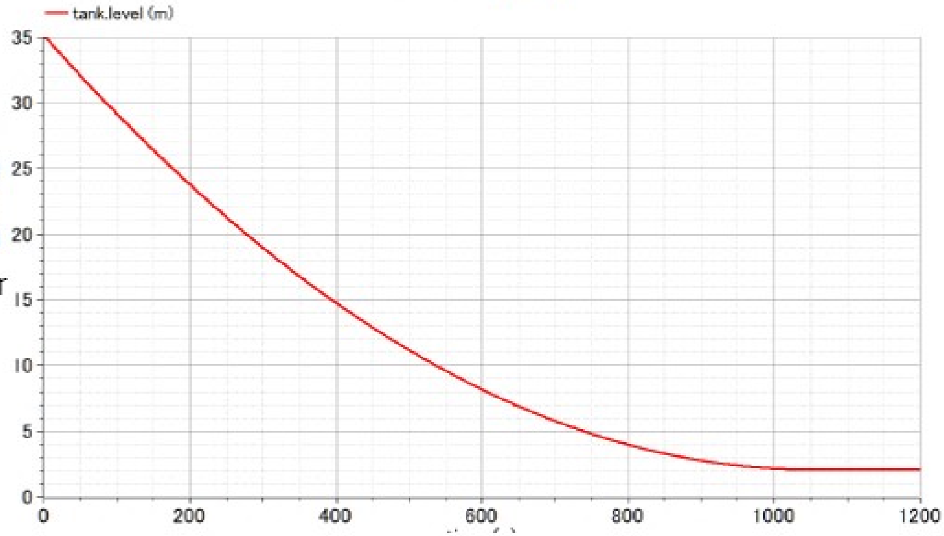
Simulation result

[kg/s]
Mass flow



time[s]

[m]
Water level



time[s]

As the water flows out of the stop valve, the water level drops and the flow

Pump

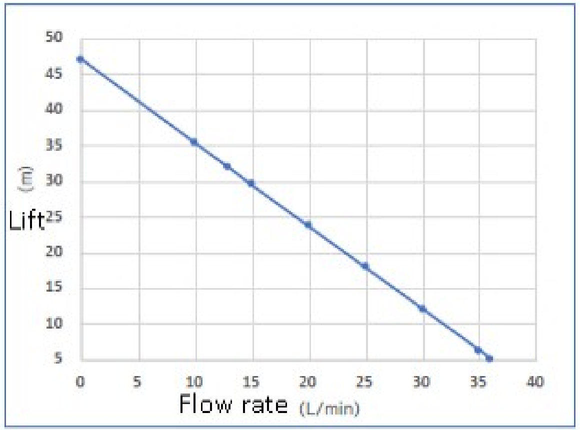
Reference EBARA CORPORATION 25HPO5.25S https://product-standard-pump.ebara.com/product/detail/P030766

Pump power (motor)

frequency	f	50	Hz
№ of poles	p	2	
Speed	n	3000	rpm

$$n = \frac{120f}{p}$$

ρ	density (25℃)	997.062	kg/m3
g	Gravitational acceleration	9.80665	m/s2



Pump specifications

Power supply	Motor nominal output	Allowable indentation height	Maximum suction height	Water supply	Lift	Starting pressure	Suction flange	Discharge flange
V	W	m	m	L/min	m	kPa	mm	mm
100	250	2	-8	30	12	140/200	25	25

Flow rate	Lift ($head$)
L/min	m
0.000	47.000
10.000	35.333
12.857	32.000
15.000	29.500
20.000	23.667
25.000	17.833
30.000	12.000
35.000	6.167
36.000	5.000
40.286	0.000

Flow rate	Flow rate (q)	Boost (Δp)	Water power (W_h)	Shaft power (W_m)	Efficiency (η)
m3/min	m3/s	Pa	W	W	-
0.0000E+00	0.0000E+00	4.5956E+05	0.000	100.000	0.00000
1.0000E-02	1.6667E-04	3.4548E+05	57.581	141.667	0.40645
1.2857E-02	2.1429E-04	3.1289E+05	67.048	153.571	0.43659
1.5000E-02	2.5000E-04	2.8845E+05	72.112	162.500	0.44376
2.0000E-02	3.3333E-04	2.3141E+05	77.136	183.333	0.42074
2.5000E-02	4.1667E-04	1.7437E+05	72.655	204.167	0.35586
3.0000E-02	5.0000E-04	1.1733E+05	58.667	225.000	0.26074
3.5000E-02	5.8333E-04	6.0297E+04	35.173	245.833	0.14308
3.6000E-02	6.0000E-04	4.8889E+04	29.334	250.000	0.11733
4.0286E-02	6.7143E-04	0.0000E+00	0.000	267.857	0.00000

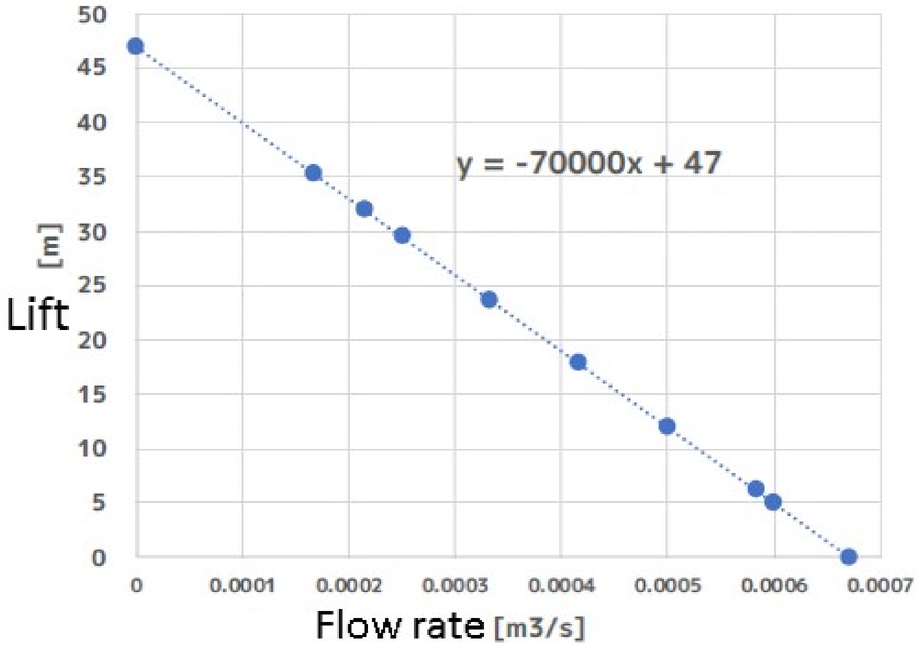
$$\Delta p = \rho g \cdot head$$

$$W_h = \Delta p \cdot q$$

$$\eta = \frac{W_h}{W_m}$$

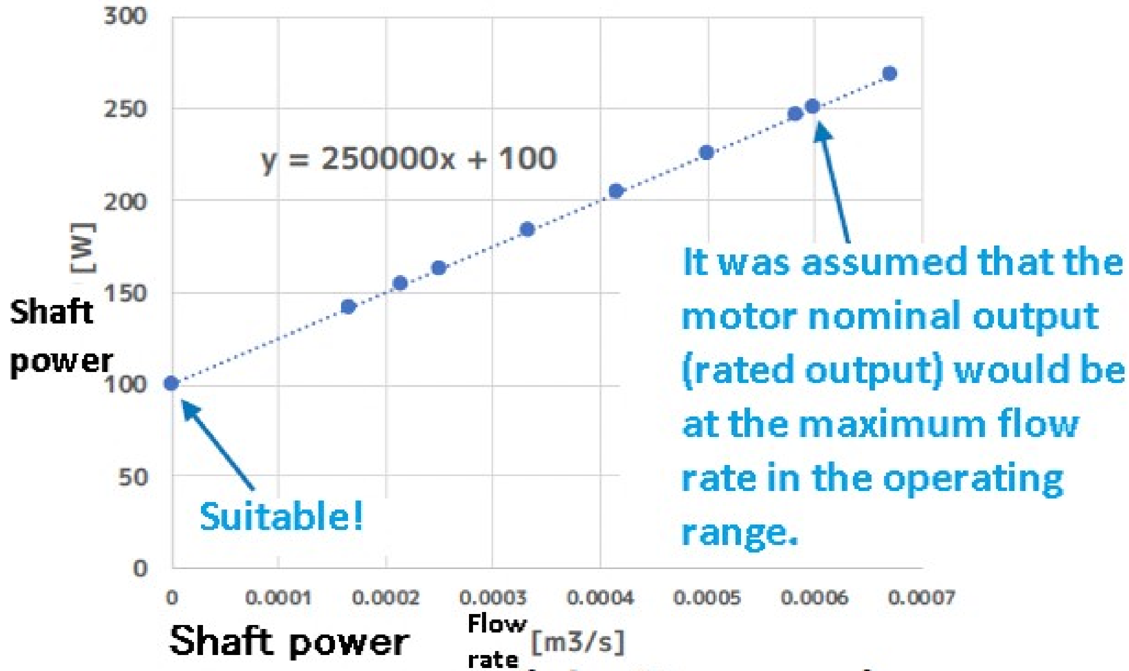
Operating range

Shaft power was created appropriately assuming that the maximum value of shaft power in the operating range is the motor nominal output

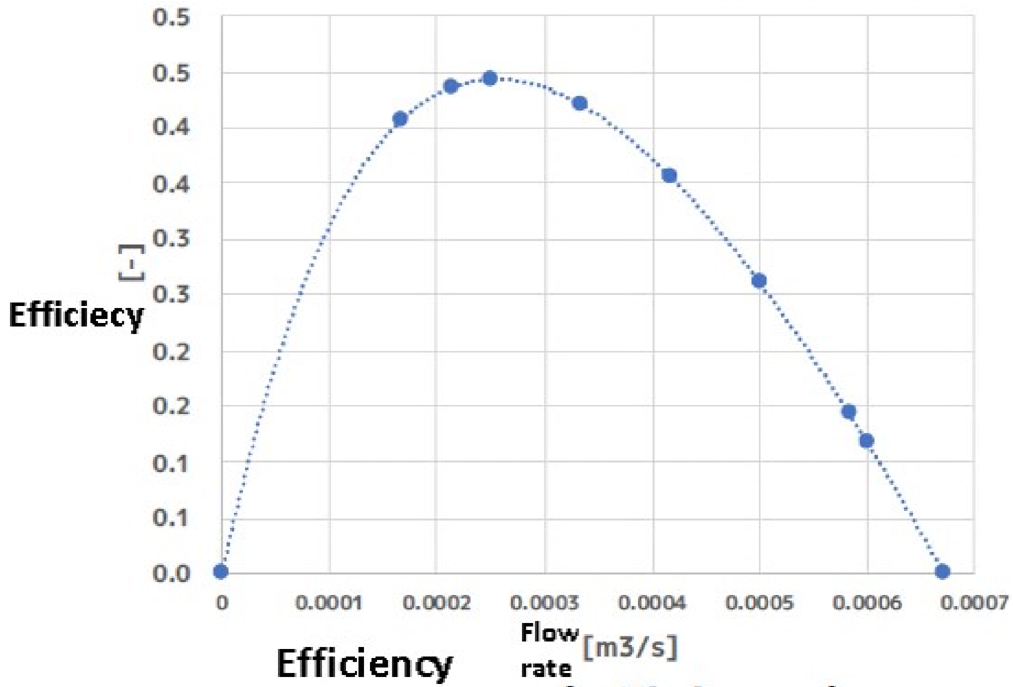


Lift characteristics(head)

The formula was calculated using the approximation curve function of Excel.



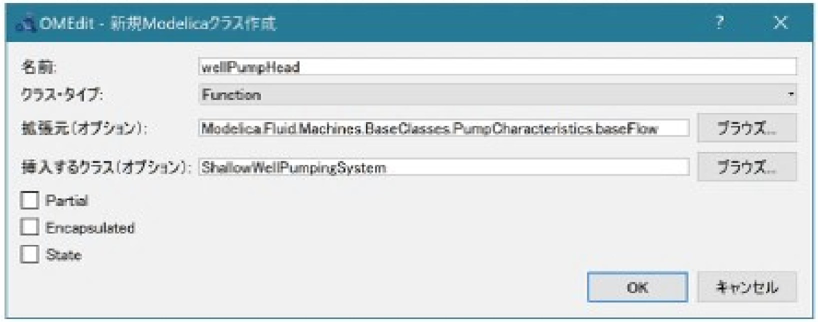
Shaft power characteristics (shaft power)



Efficiency characteristics (efficiency)

Lift characteristics (head curve)

① Right-click ShallowWellPumpingSystem in the library browser and select New Modelica Class.



The class to insert:

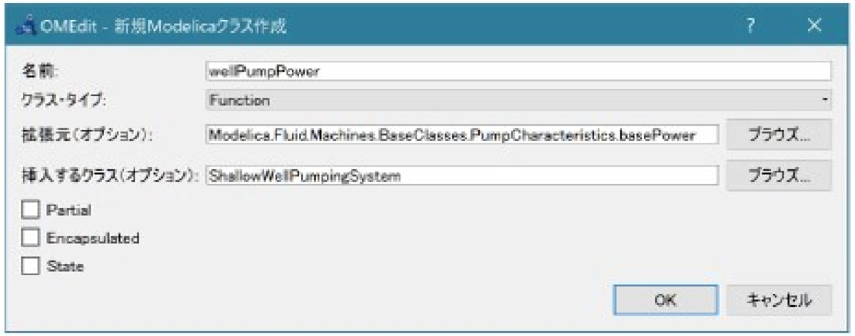
Name: wellPumpHead
Class type: Function
Expansion source: Modelica.Fluid.Machines.BaseClasses.PumpCharacteristic.baseFlow
The class to insert: ShallowWellPumpingSystem

② In the text view, enter the calculation formula for the head characteristic.

```
function wellPumpHead
  extends Modelica.Fluid.Machines.BaseClasses.PumpCharacteristics.baseFlow;
algorithm
  head := (-70000.0 * V_flow) + 47.0;
end wellPumpHead;
```

Shaft power characteristics (shaft power curve)

①Right-click SallowWellPumpingSystem in the library browser and select New Modelica Class.



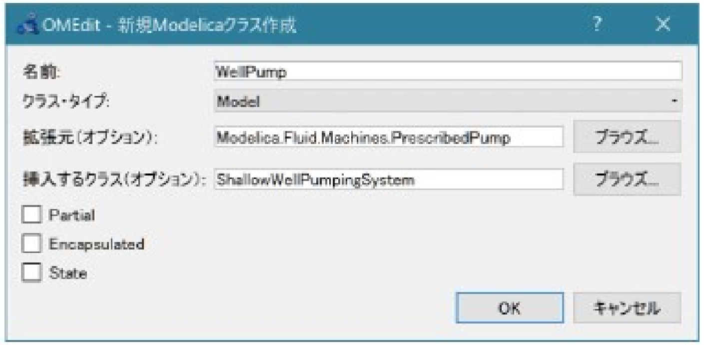
Name: wellPumpPower
Class type: Function
Expansion source: Modelica.Fluid.Machines.BaseClasses.PumpCharacteristic.basePower
The class to insert: ShallowWellPumpingSystem

②In the text view, enter the calculation formula for the shaft power characteristics.

```
function wellPumpPower
  extends Modelica.Fluid.Machines.BaseClasses.PumpCharacteristics.basePower;
algorithm
  consumption := 250000 * V_flow + 100;
end wellPumpPower;
```

WellPump Creating a model

① Right-click ShallowWellPumpingSystem in the library browser and select New Modelica Class.



Name: WellPump
Class type: Model
Expansion source: Modelica.Fluid.Machines.PrescribedPump
Class to insert: ShallowWellPumpingSystem

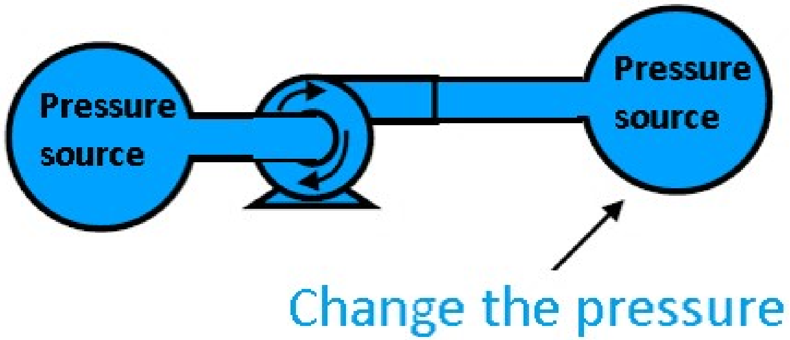
② In the text view, set flowCharacteristic, poworCharacteristic, parameters.

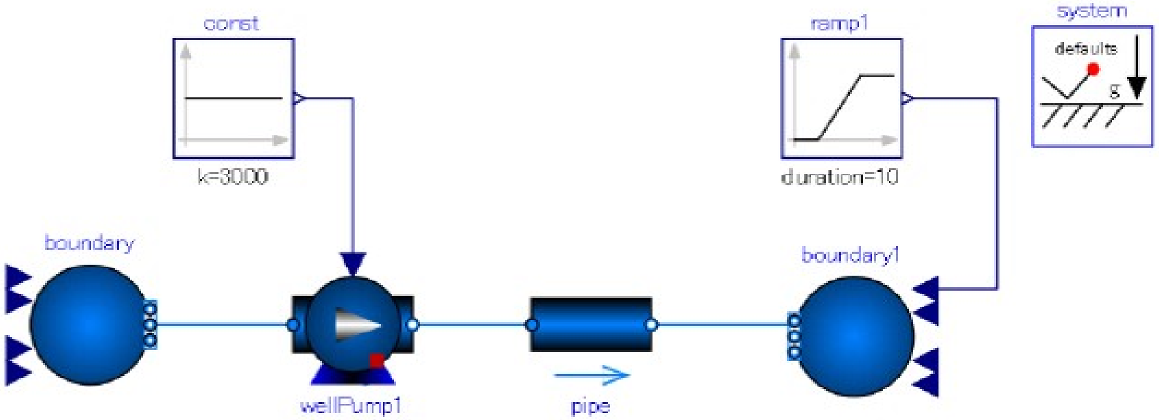
```
Version:1.0 StartHTML:0000000107 EndHTML:0000005158 StartFragment:0000000471 EndFragment:0000005120
model WellPump
  extends Modelica.Fluid.Machines.PrescribedPump(
    redeclare function flowCharacteristic = wellPumpHead,
    redeclare function powerCharacteristic = wellPumpPower,
    N_nominal = 3000, checkValve = true, m_flow_start = 0.4, use_N_in = true,
    use_powerCharacteristic = true, V = 0.001,
    energyDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial,
    massDynamics = Modelica.Fluid.Types.Dynamics.SteadyStateInitial);
equation

end WellPump
```

Setting of pump characteristics

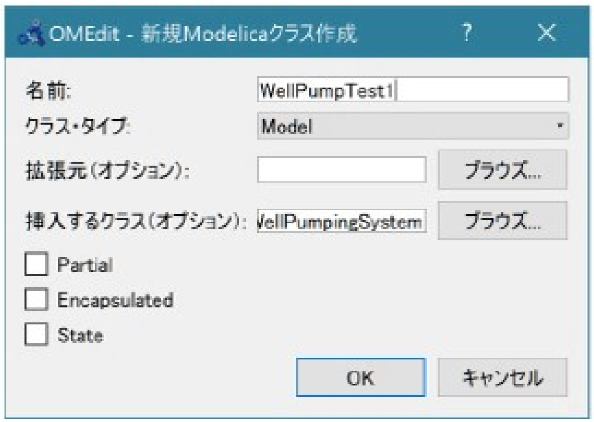
WellPumpTest1 Unit test

Creating a WellPump test model





① Right-click ShallowWellPumpingSystem in the Library Browser and select New Modelica Class



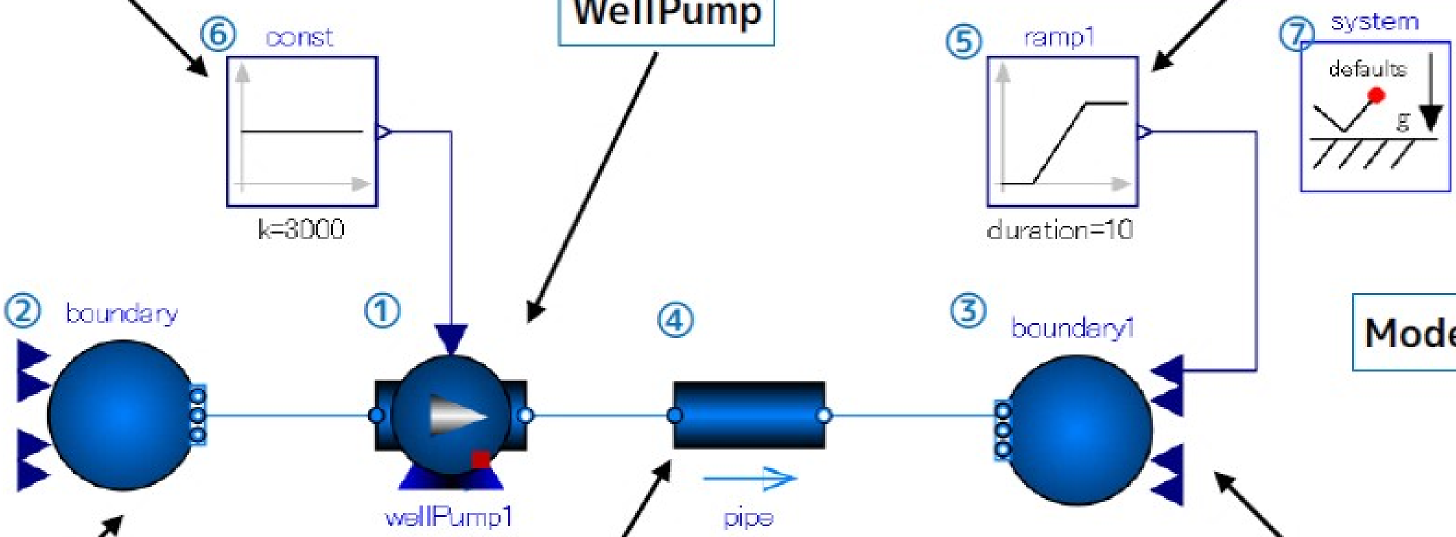
Name: WellPumpTest
Class type: Model
Class to insert: ShallowWellPumpingSystem

② Place parts in the diagram view and set parameters.

Modelica.Blocks.Sources.Constant
k = 3000

WellPump

Modelica.Blocks.Sources.Rampt
height = 400000
duration = 10
offset = 120000
startTime = 0



Modelica.Fluid.System

Modelica.Fluid.Sources.Boundary_pT
p =101325 [Pa]
T = 293.15 [K]
nPorts = 1

Modelica.Fluid.Pipes.StaticPipe
length = 1 [m]
diameter = 0.025 [m]
height_ab = 0 [m]

Modelica.Fluid.Sources.Boundary_pT
use_p_in = true
nPorts = 1

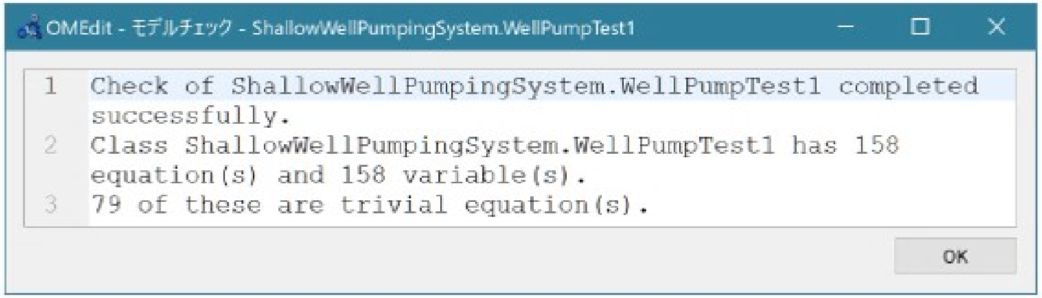
Select ports [1] for Boundary_pT connector.

[1]

③ Edit the source code in the text view and set the Media (fluid model).

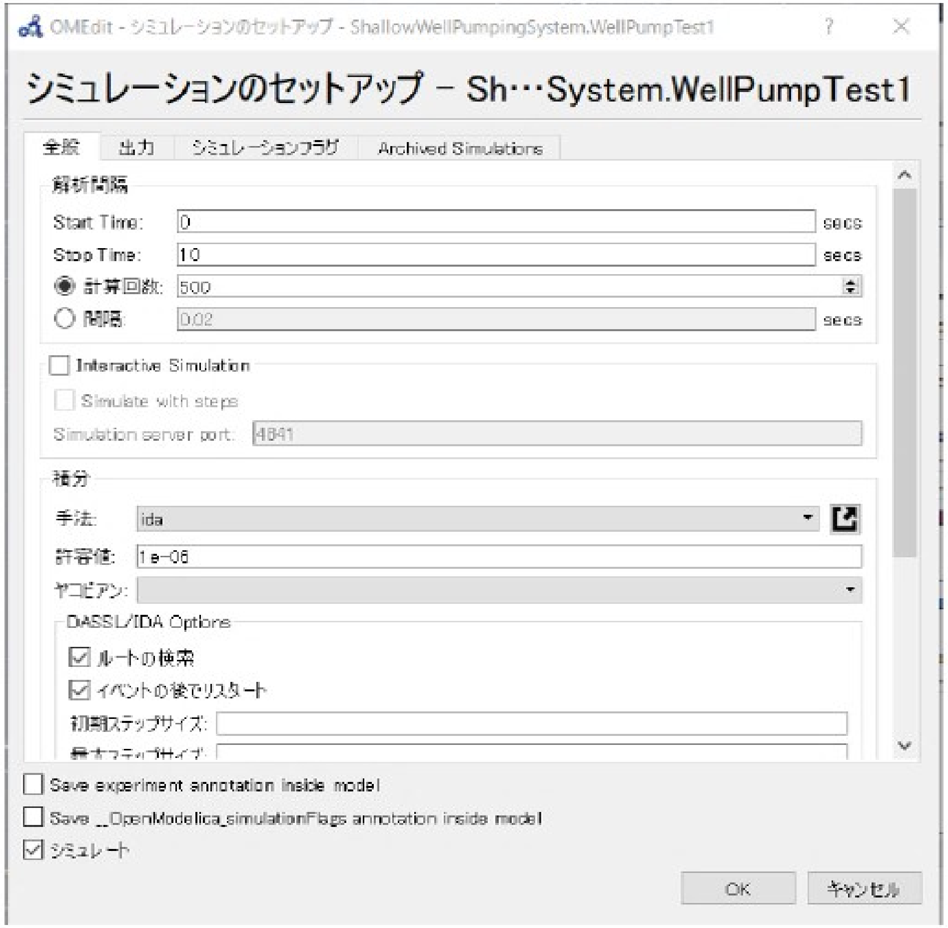
```
model WellPumpTest1
  replaceable package Medium = Modelica.Media.Water.StandardWater;
  ShallowWellPumpingSystem.WellPump wellPump1(redeclare package Medium = Medium) annotation( …);   ①
  Modelica.Fluid.Sources.Boundary_pT boundary(                                                       ②
    redeclare package Medium = Medium, T = 293.15, nPorts = 1, p = 101325)  annotation( …);
  Modelica.Fluid.Sources.Boundary_pT boundary1(                                                      ③
    redeclare package Medium = Medium, nPorts = 1, use_p_in = true)  annotation( …);
  Modelica.Fluid.Pipes.StaticPipe pipe(                                                              ④
    redeclare package Medium = Medium, diameter = 0.025, length = 1)  annotation( …);
  Modelica.Blocks.Sources.Ramp ramp1(                                                                ⑤
    duration = 10, height = 400000, offset = 120000, startTime = 0)  annotation( …);               ⑥
  Modelica.Blocks.Sources.Constant const(k = 3000)  annotation( …);
  inner Modelica.Fluid.System system annotation( …);                                                 ⑦
equation
  connect(const.y, wellPump1.N_in) annotation( …);
  connect(ramp1.y, boundary1.p_in) annotation( …);
  connect(pipe.port_b, boundary1.ports[1]) annotation( …);
  connect(wellPump1.port_b, pipe.port_a) annotation( …);
  connect(boundary.ports[1], wellPump1.port_a) annotation( …);
  annotation( …);
end WellPumpTest1;
```

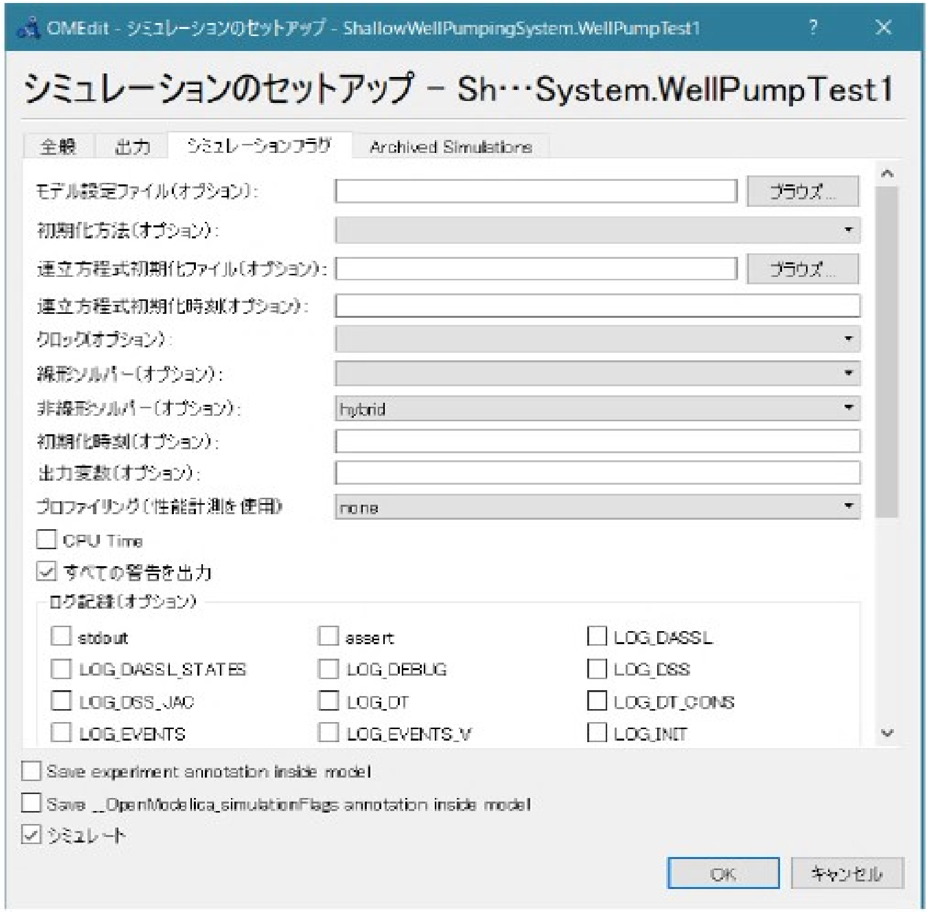
④ Simulation> Model Check



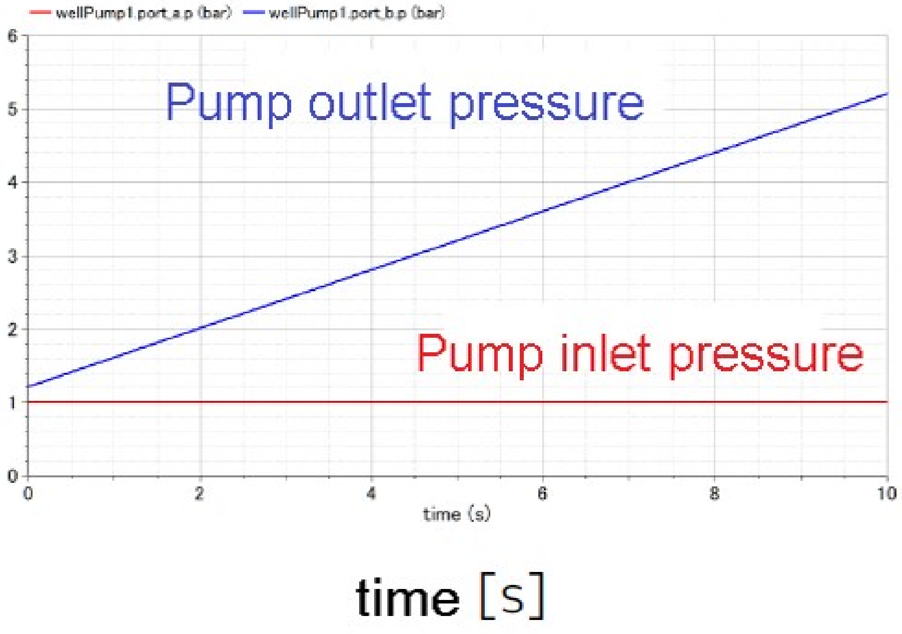
Start Time = 0 [s]
Sop Time = 10 [s]
Number of calculations = 500
Method: ida
Non-linear solver options: hybrid

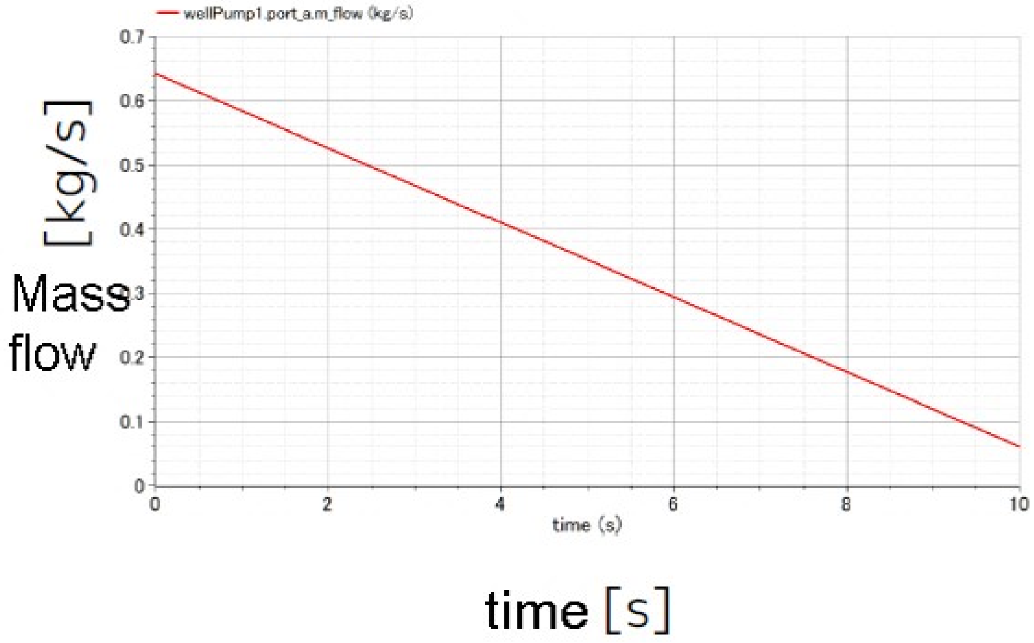
⑤ Simulation> Simulation setup





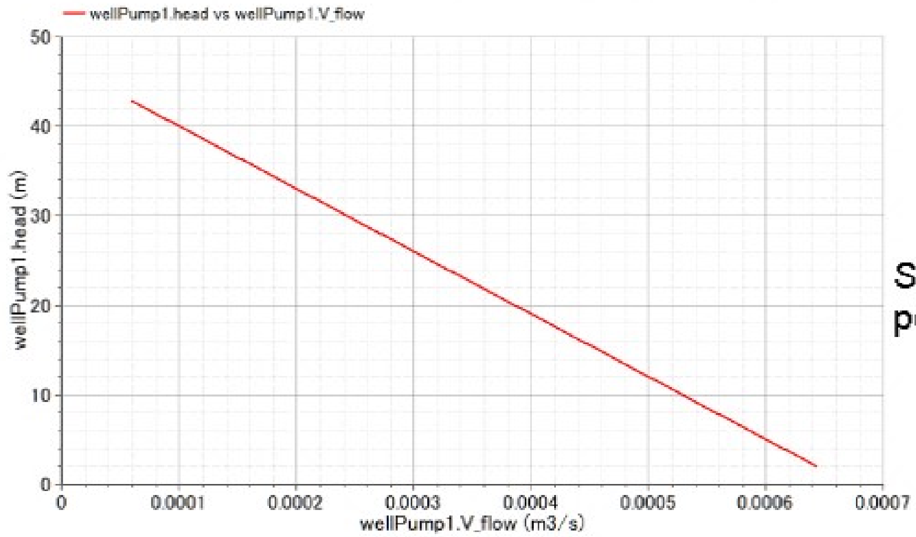
Simulation result





Simulation results

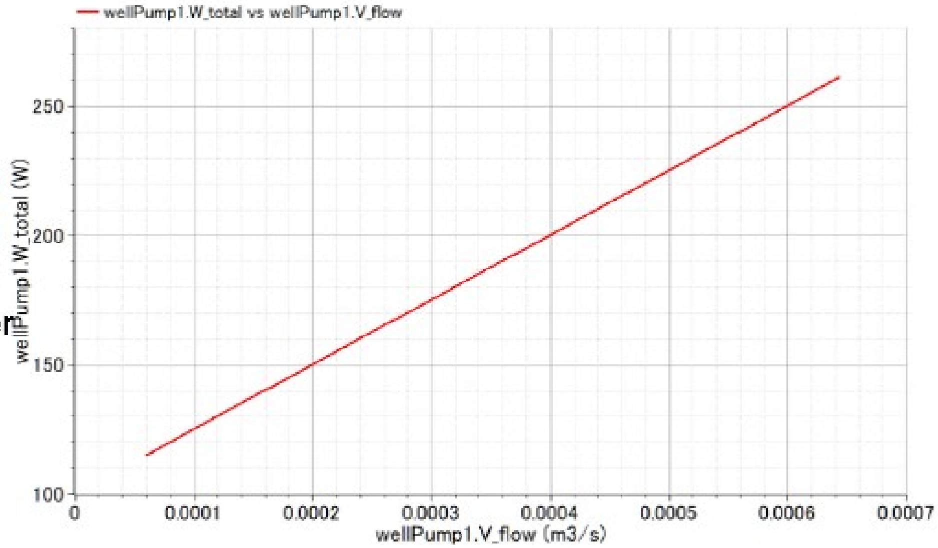
Lift [m]



Flow rate [m3/s]

Lift characteristics

Shaft power [W]

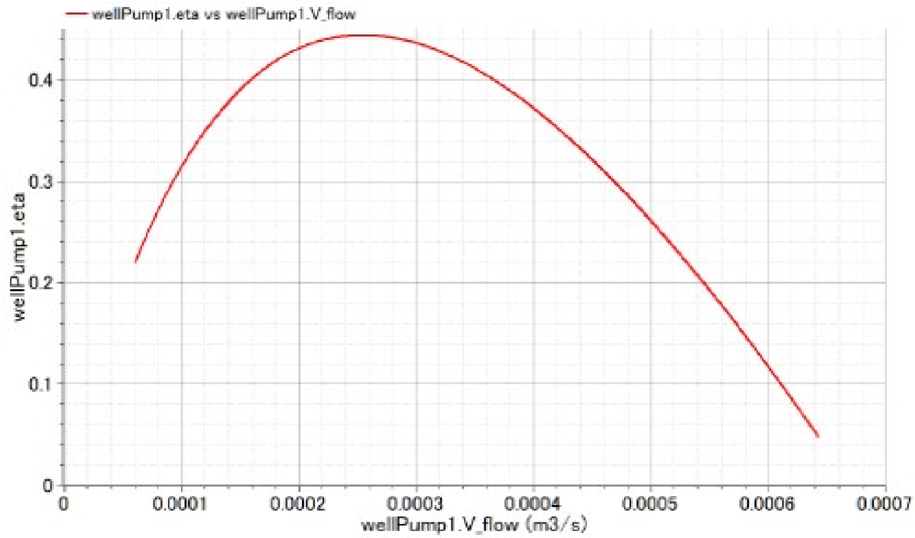


Flow rate [m3/s]

Shaft power characteristics

Pump performance was reproduced !

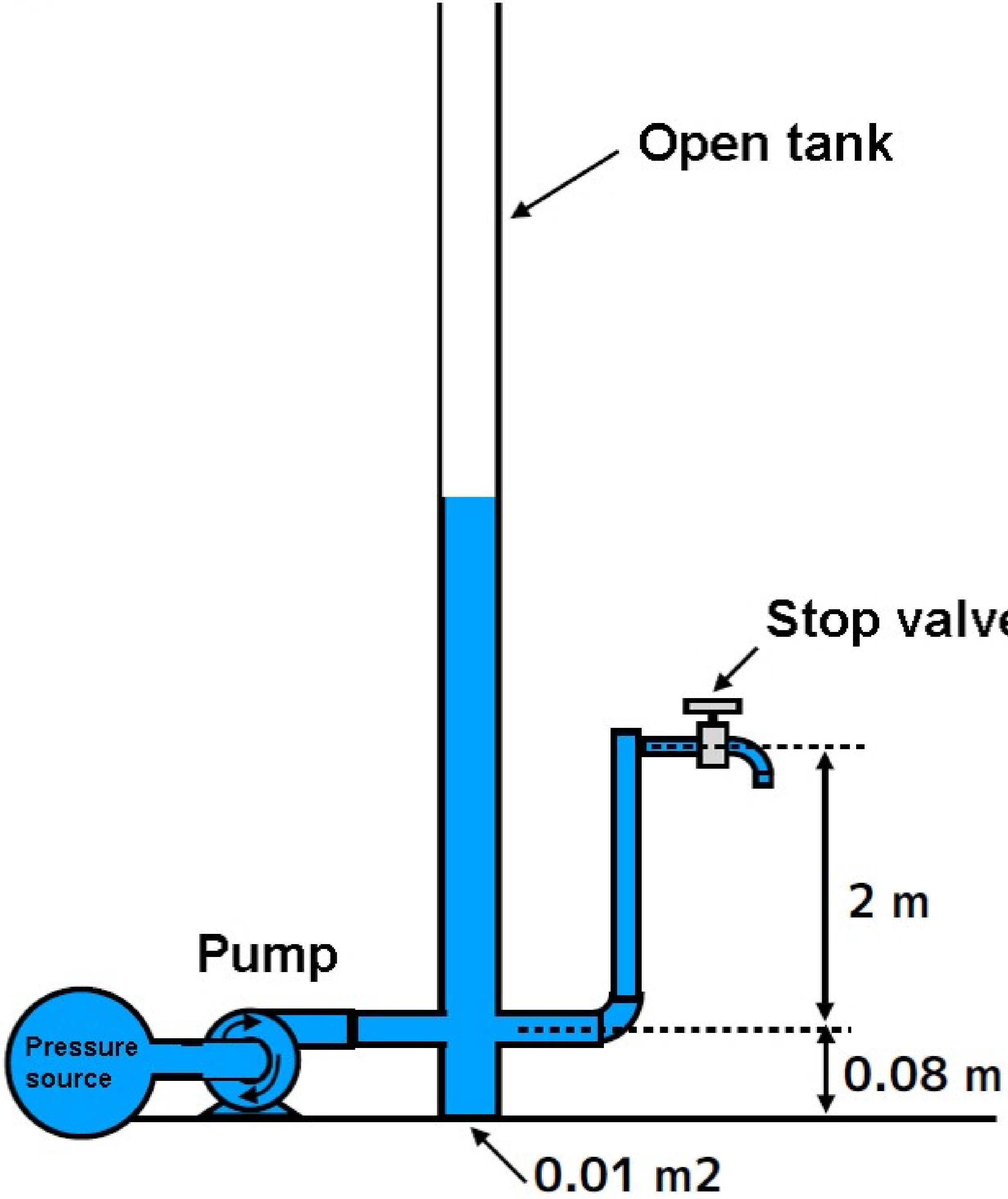
Efficiency [-]



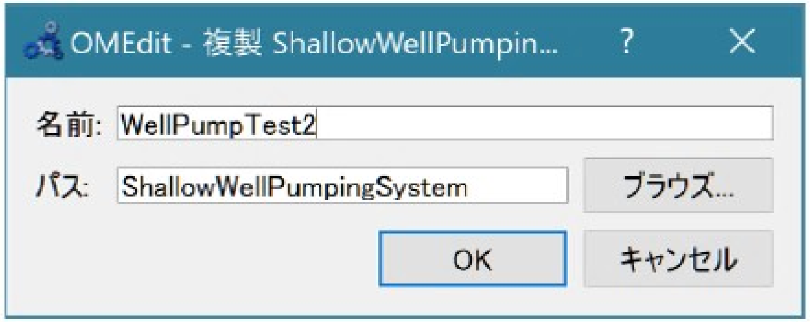
Flow rate [m3/s]

Efficiency characteristics

WellPumpTest 2 Pump + open tank + stop valve



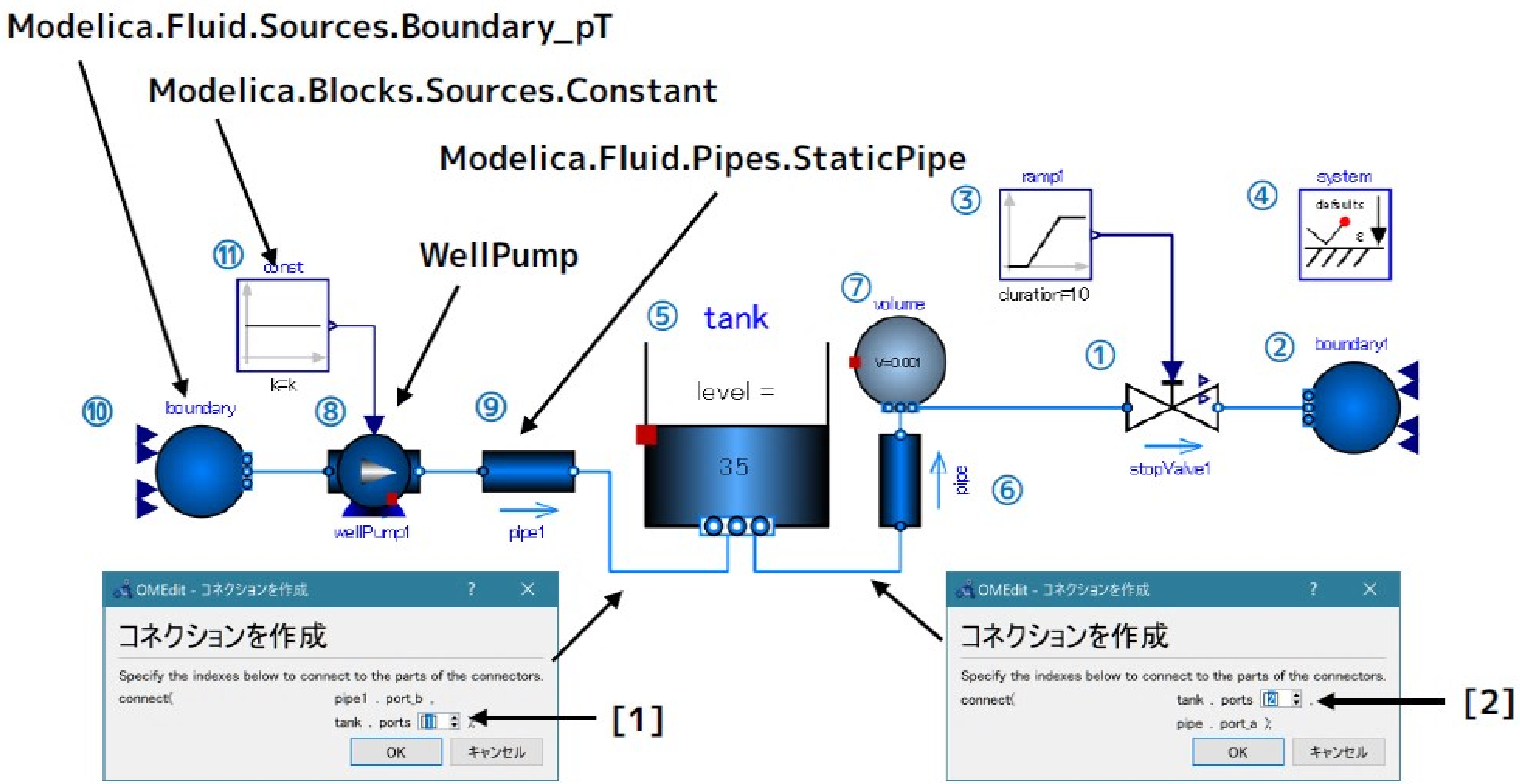
① Right-click StopValveTest2 and select Duplicate.



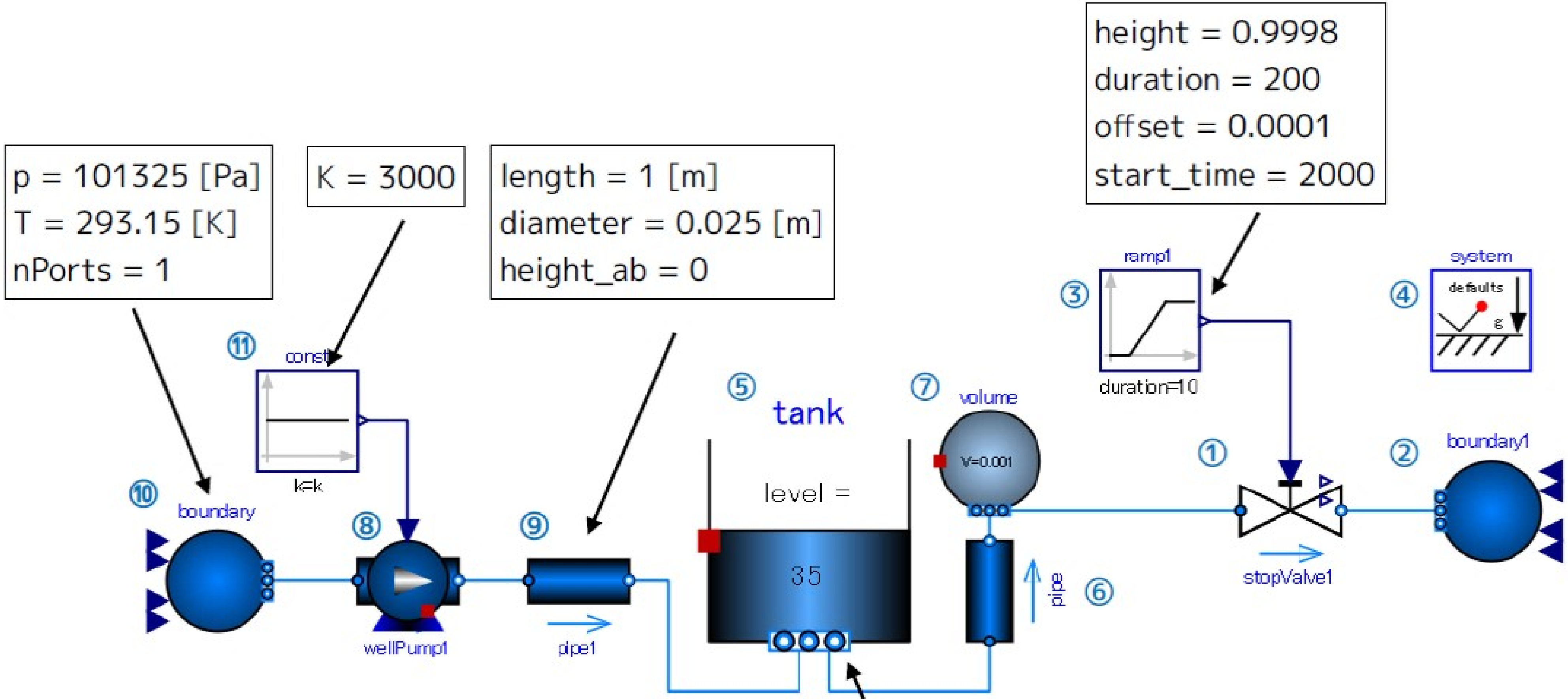
Name: WellPumpTest2

Path: ShallowWellPumpingSystem

② Place components and connect.



③ Set parameters



Modify the parameters in the text view referring to the next slide.

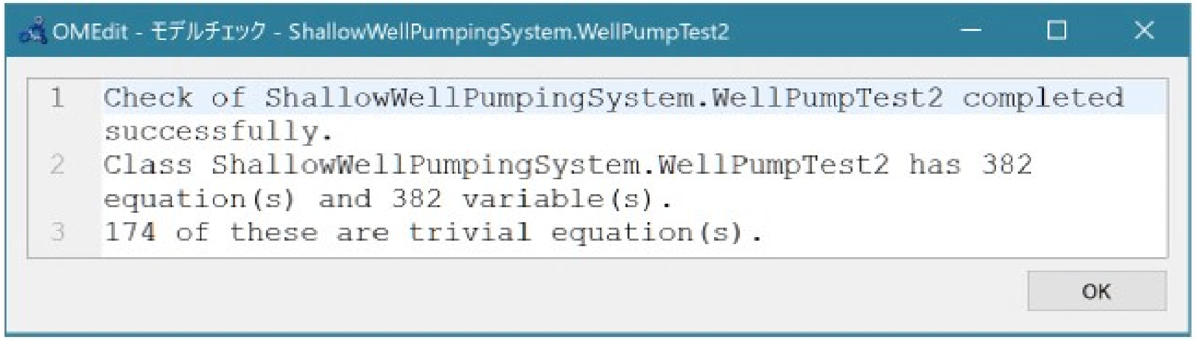
④ Set the Media (fluid model) in the text view.

Modify level_start, nPorts and portsData of OpenTank.

```
model WellPumpTest2
  replaceable package Medium = Modelica.Media.Water.StandardWater;
  ShallowWellPumpingSystem.StopValve stopValve1(redeclare package Medium = Medium) annotation( …);   ①
  Modelica.Fluid.Sources.Boundary_pT boundary1(                                                     ②
    redeclare package Medium = Medium, T = 293.15, nPorts = 1, p = 101325) annotation( …);
  Modelica.Blocks.Sources.Ramp ramp1(duration = 10, height = 1, offset = 0, startTime = 2000)     ③
    annotation( …);
  inner Modelica.Fluid.System system annotation( …);                                                ④
  Modelica.Fluid.Vessels.OpenTank tank(                                                             ⑤
    redeclare package Medium = Medium,
    crossArea = 0.01,
    energyDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial,
    height = 50, level_start = 0,
    massDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial,
    nPorts = 2,
    portsData = {
     Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(diameter = 0.025, height = 0.08),
     Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(diameter = 0.015, height = 0.08)
     },
    use_portsData = true) annotation( …);
  Modelica.Fluid.Pipes.StaticPipe pipe(                                                             ⑥
    redeclare package Medium = Medium, diameter = 0.015, height_ab = 2, length = 2) annotation( …);
```

```
 Modelica.Fluid.Vessels.ClosedVolume volume(                                          ⑦
    redeclare package Medium = Medium, V = 0.001,
    energyDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial,
    massDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial,
    nPorts = 2, use_portsData = false) annotation( …);
  ShallowWellPumpingSystem.WellPump wellPump1(redeclare package Medium = Medium) annotation( …);   ⑧
  Modelica.Fluid.Pipes.StaticPipe pipe1(                                              ⑨
    redeclare package Medium = Medium, diameter = 0.025, height_ab = 0, length = 1)  annotation( …);
  Modelica.Fluid.Sources.Boundary_pT boundary(                                        ⑩
    redeclare package Medium = Medium, T = 293.15, nPorts = 1, p = 101325)  annotation( …);
  Modelica.Blocks.Sources.Constant const(k = 3000)  annotation( …);                   ⑪
equation
  connect(const.y, wellPump1.N_in) annotation( …);
  connect(ramp1.y, stopValve1.opening) annotation( …);
  connect(tank.ports[2], pipe.port_a) annotation( …);
  connect(pipe1.port_b, tank.ports[1]) annotation( …);
  connect(boundary.ports[1], wellPump1.port_a) annotation( …);
  connect(wellPump1.port_b, pipe1.port_a) annotation( …);
  connect(pipe.port_b, volume.ports[1]) annotation( …);
  connect(volume.ports[2], stopValve1.port_a) annotation( …);
  connect(stopValve1.port_b, boundary1.ports[1]) annotation( …);
  annotation( …);
end WellPumpTest2;
```

⑤Simulation> Model Check



Start Time = 0 [s]

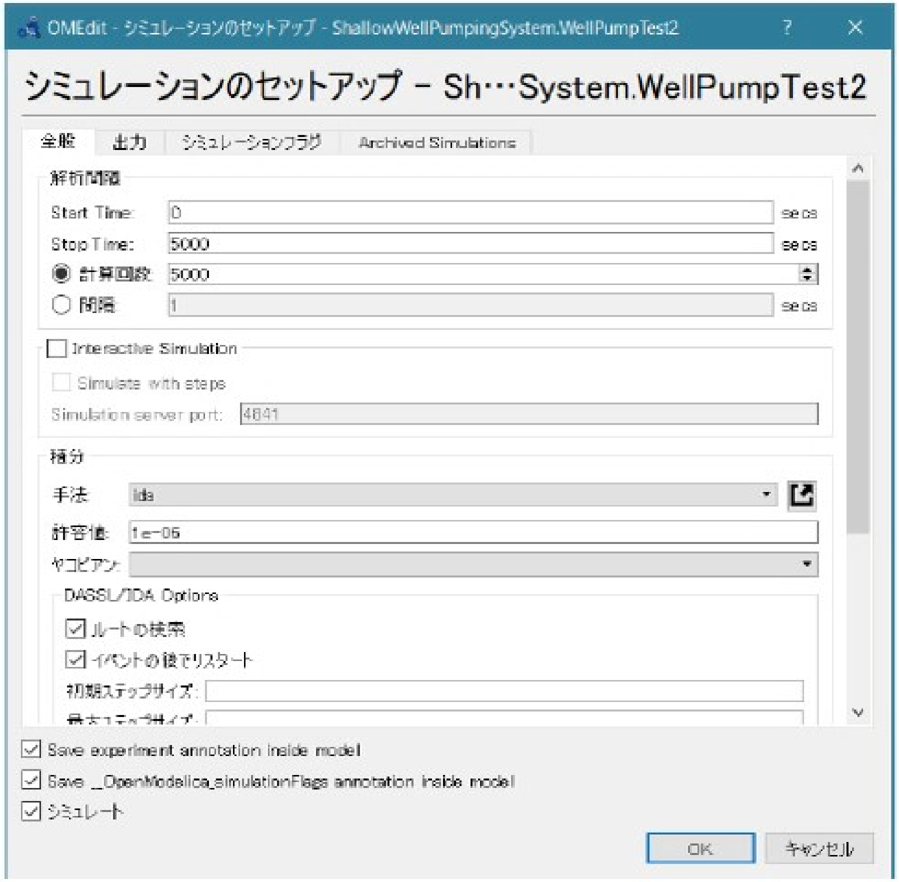
Stop Time = 5000 [s]

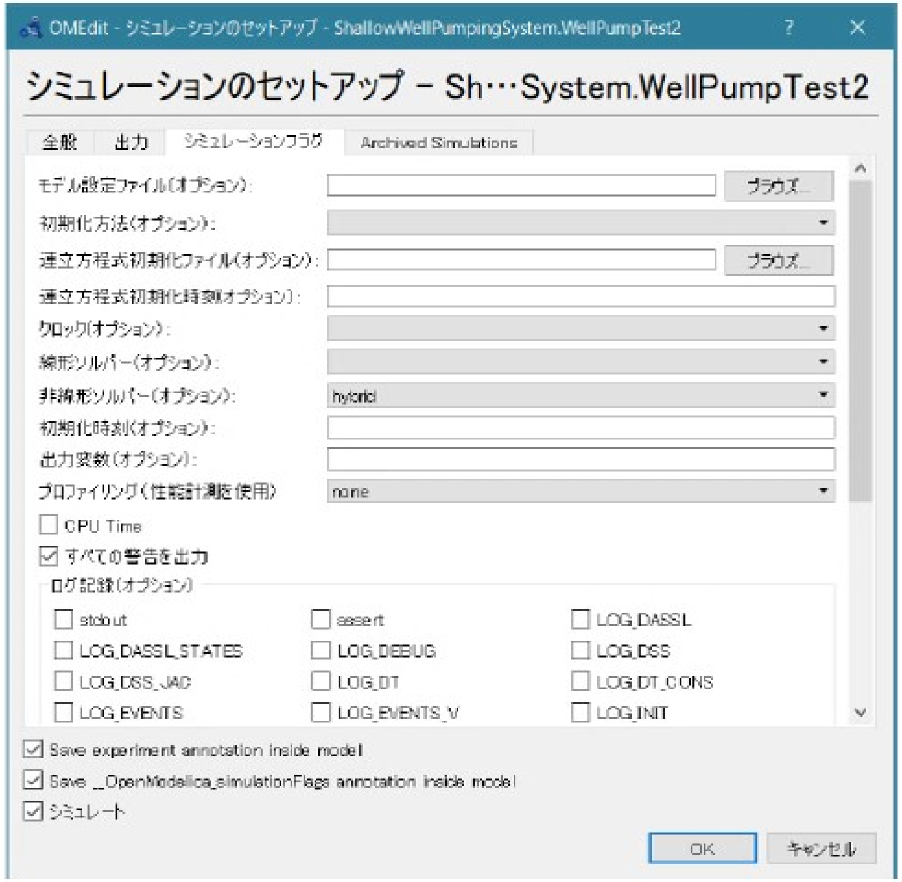
Number of calculations = 5000

Integration method: ida

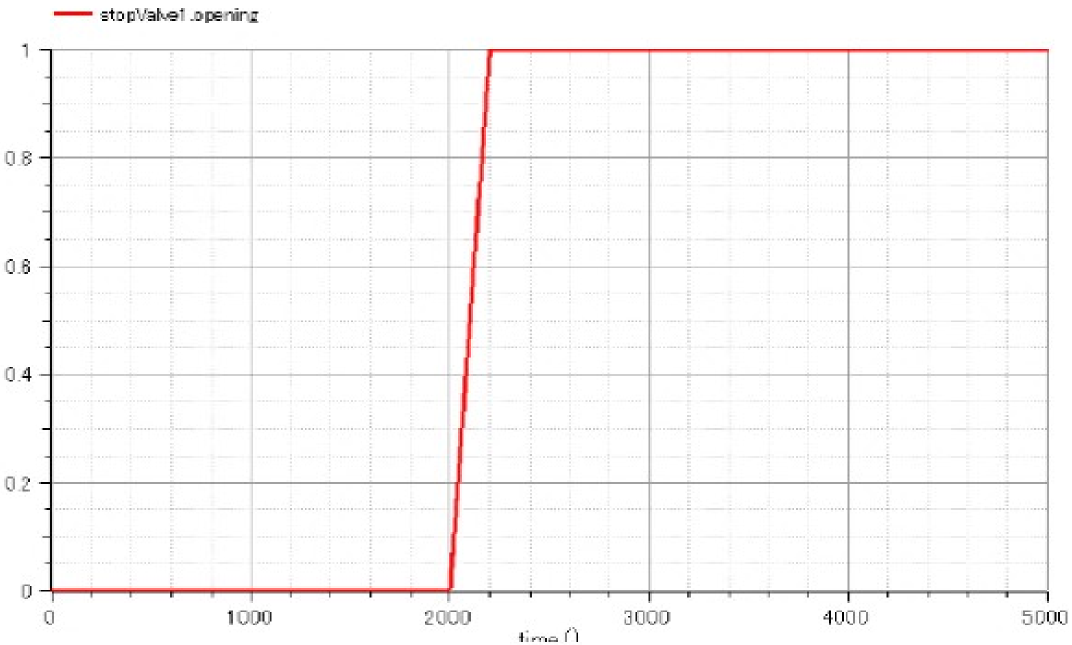
Nonlinear solver: hybrid

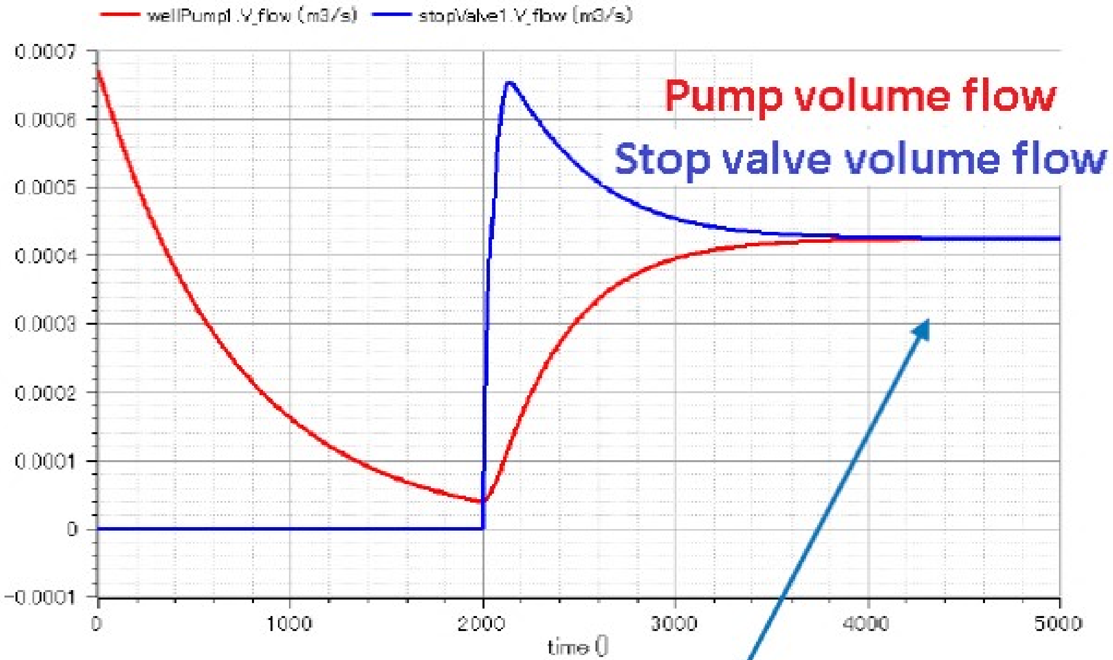
⑥Simulation> Simulation setup



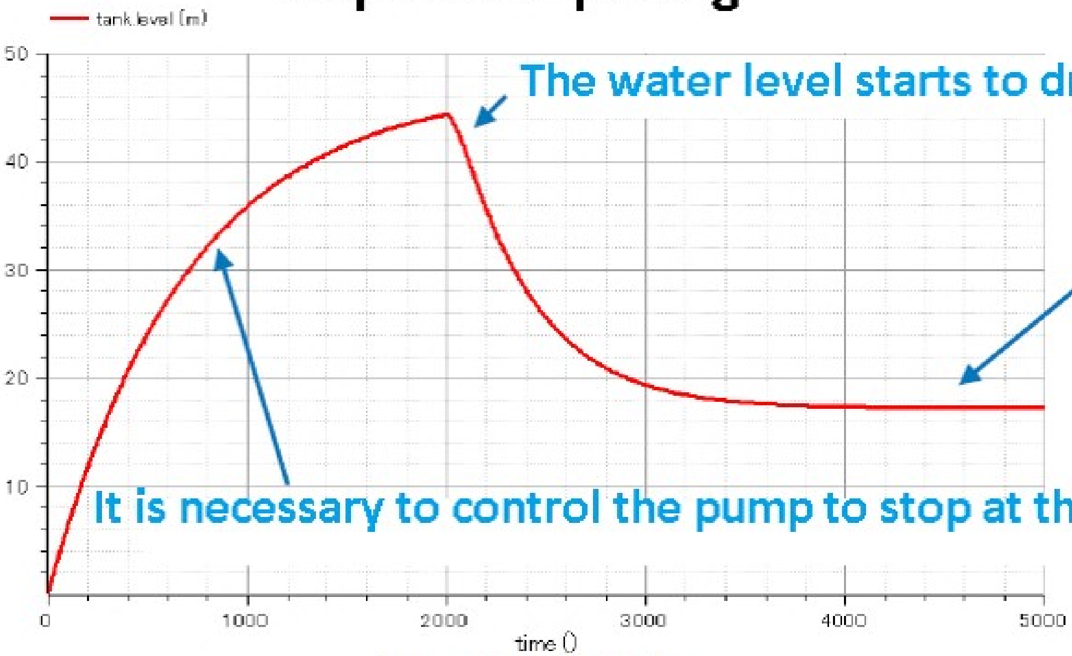


Simulation result





Stop valve opening



タンクの水位

Conclusion

- The model of the stop valve and the pump was created, and a unit test was performed.
- A test model with a pump, open tank and stop valve connected was created. It was confirmed that this model was in a steady state at the water level where the flow rate of the pump and the flow rate of the stop valve were balanced.

Future tasks

- Model the pipes in the well using StaticPie.
- Modify OpenTank to create a pressure tank model.
- Model pressure switches with components in Modelica.Blocks.
- Combine the above to complete a shallow well pump supply system model.