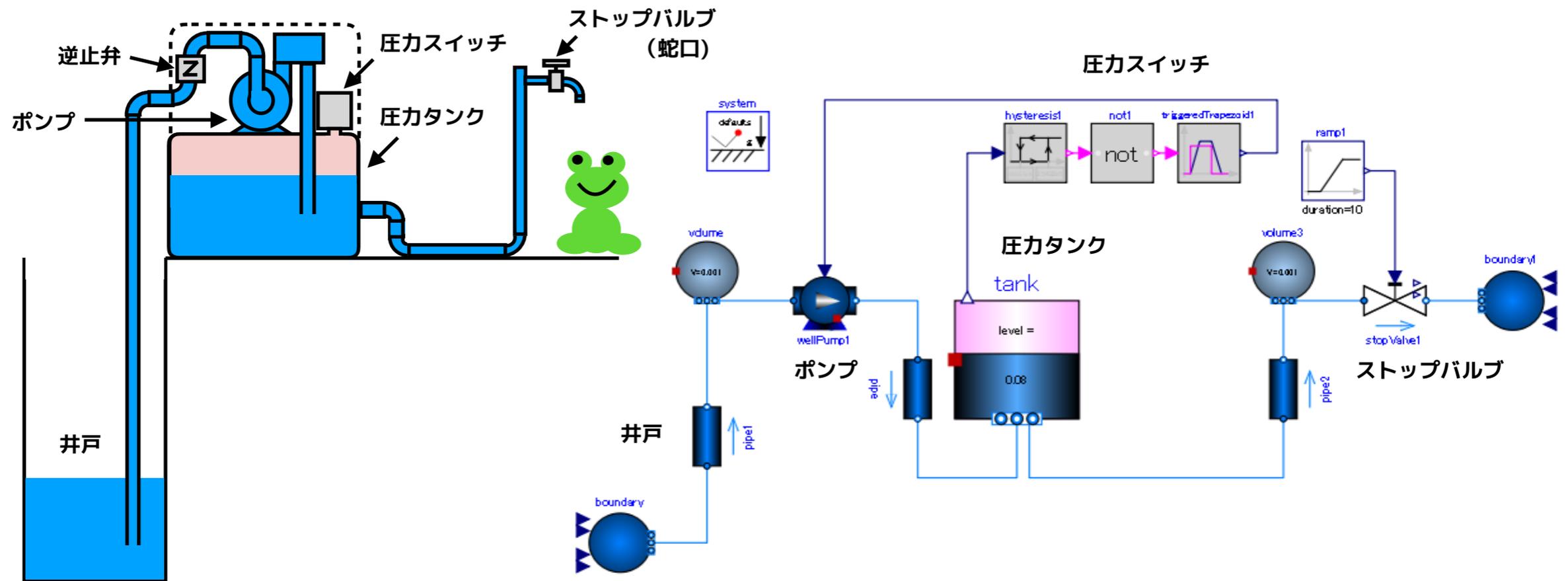


# ハンズオン

## OpenModelica による

### 浅井戸ポンプ給水システムモデル (第1回)



2019/11/16 第15回 Modelica ライブラリ勉強会  
finback

## OpenModelica による浅井戸ポンプ給水システムモデル（第1回）

圧力タンク式の浅井戸ポンプをモデル化して、井戸水を汲み上げて給水するモデルを試作します。

- ポンプ
- ストップバルブ（蛇口）
- 圧力タンク
- 圧力スイッチ（ポンプのオンオフ制御装置）
- 配管系（井戸、パイプ）

などをモデル化して組み合わせます。

第1回は、**ポンプ**と**ストップバルブ**をモデル化します。

参考資料: Modelica.Fluid.Valves の液体用バルブモデルについて  
<https://www.amane.to/archives/442>

Modelica.Fluid.Machines の遠心ポンプモデルについて  
<https://www.amane.to/archives/377>

# 本日のメニュー

## 浅井戸ポンプ給水システム

### ストップバルブ (stop valve)

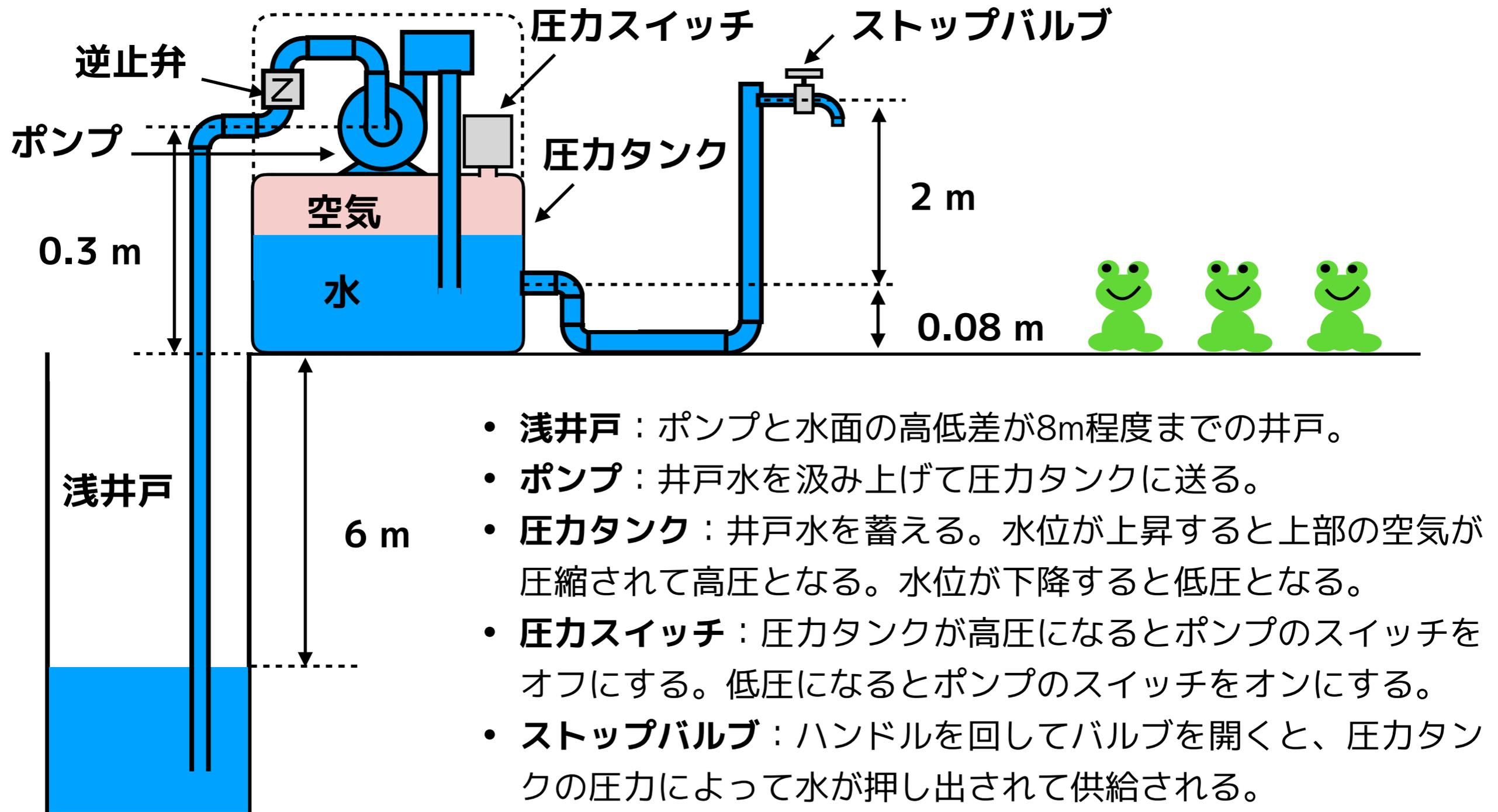
- StopValve モデルの作成
- StopValveTest1 単体テスト
- StopValveTest2 ストップバルブとオープンタンク

### ポンプ

- 揚程特性 (head curve)
- 軸動力特性 (power curve)
- WellPump モデルの作成
- WellPumpTest1 単体テスト
- WellPumpTest2 ポンプ+オープンタンク+ストップバルブ

### まとめ

# 浅井戸ポンプ給水システム (Shallow well pumping system)



参考 [https://as76.net/life/a\\_tank\\_p.php](https://as76.net/life/a_tank_p.php) など

# ストップバルブ (Stop Valve)

## 水道蛇口のバルブ

参考資料 生活の知恵袋 <https://www.seikatu-cb.com/suidou/sknow.html>

合同会社ワライト <https://www.walight.jp/2017/07/01/節水コマ-全開にしたら同じこと/>

環境省 <https://www.mhlw.go.jp/topics/bukyoku/kenkou/suido/jouhou/kankyuu/dl/090729-1d.pdf>

(元ネタは東京都水道局「環境報告書（平成19年度版）」)

圧力差	0.1	Mpa	100000	Pa
最大流量	22.1	L/min	0.000368333	m <sup>3</sup> /s
密度	998.233	kg/m <sup>3</sup>	998.233	kg/m <sup>3</sup>
Av			3.68008E-05	m <sup>2</sup>
質量流量			0.367682488	kg/s

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

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

参考資料の図からトレースした流量  
自己責任で参考にしてね！

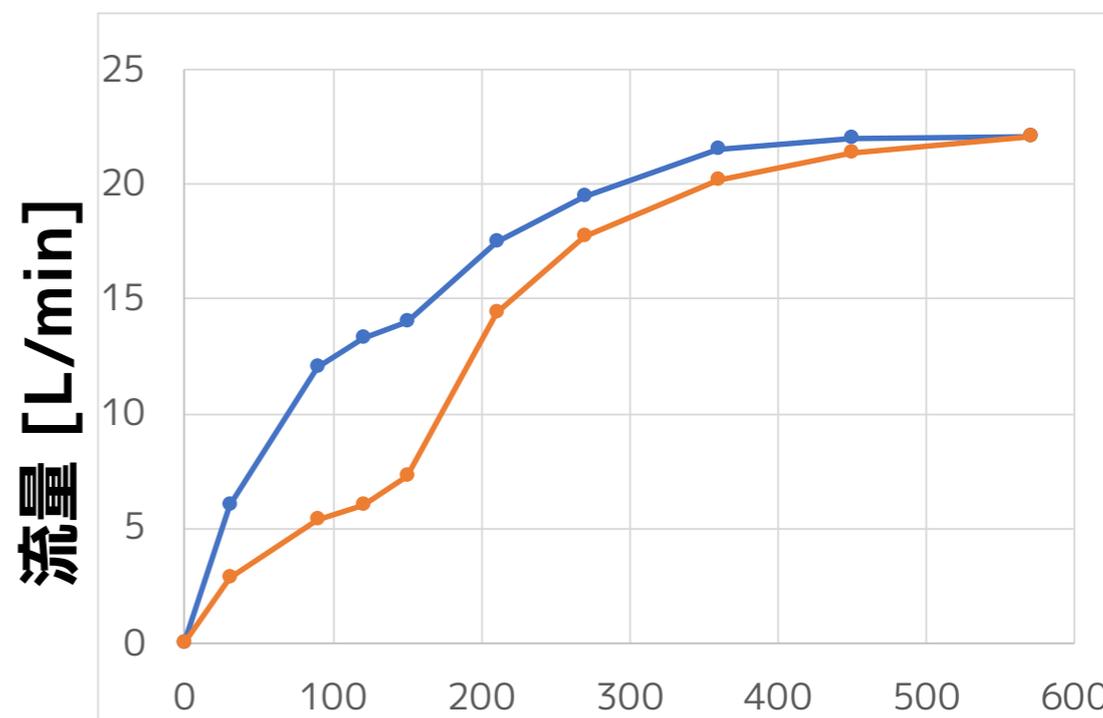
### 角度と流量

### 固有流量特性

角度や流量の最大値で正規化

角度	普通	節水コマ
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

開度	普通のコマ	節水コマ
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

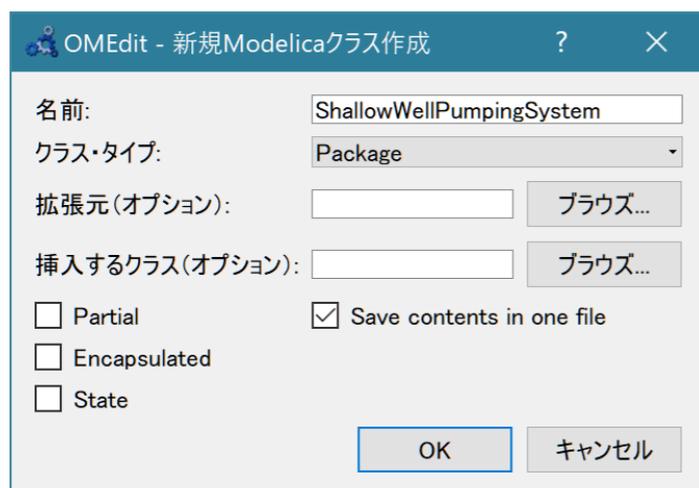


バルブの回転角度 [deg]

# StopValve モデルの作成

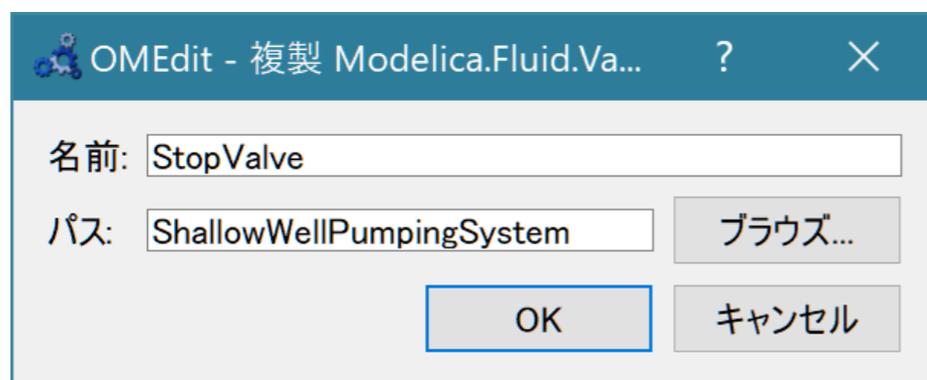
Modelica.Fluid.Valves.ValveIncompressible（液体用のバルブモデル）を複製して StopVale を作成し、CombiTable1D で固有流量特性のフィッティングができるように改造する。

## ① ファイル > Modelica クラス新規作成



名前: ShallowWellPumpingSystem  
クラス・タイプ: Package

## ② ライブラリブラウザの Modelica.Fluid.Valves.ValveIncompressible を右クリックし、「複製」を選択する。



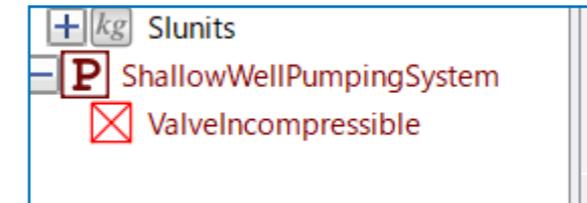
名前: StopValve  
パス: ShallowWellPumpingSystem

## ③ テキストビューで、StopValve のソースコードを編集する。

```

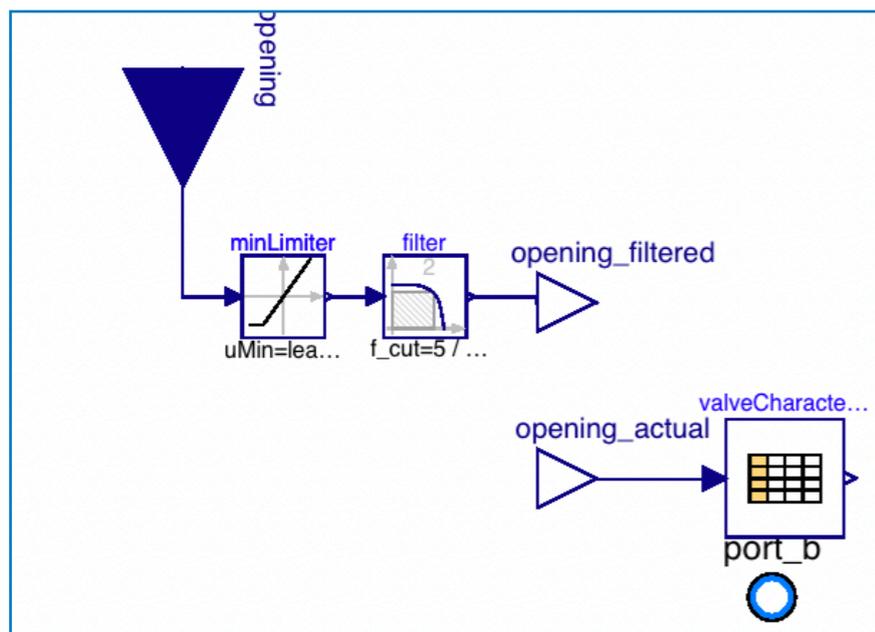
model StopValve "Valve for (almost) incompressible fluids"
  extends Modelica.Fluid.Valves.BaseClasses.PartialValve(
    CvData = Modelica.Fluid.Types.CvTypes.Av,
    Av = 3.68008e-5,
    dp_nominal = 1e5,
    m_flow_nominal = 0.3);
  import Modelica.Fluid.Types.CvTypes;
  import Modelica.Constants.pi;
  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" ;

```



複製したままの状態ではライブラリブラウザで赤い×がつきますが extends ... や import 文でスコープを調整すると直ります。

## ④ ダイアグラムビューに、Modelica.Blocks.Tables.CombiTable1D を貼り付ける。名前を valveCharacteristicTable にする。



## ⑤ opening\_actual と valveCharacteristicTable の入力を接続する。



[1]

## ⑥ テキストビューで

valveCharacteristicTable に、普通コマの固有流量特性データを設定する。  
smoothness に補完方法を設定する。

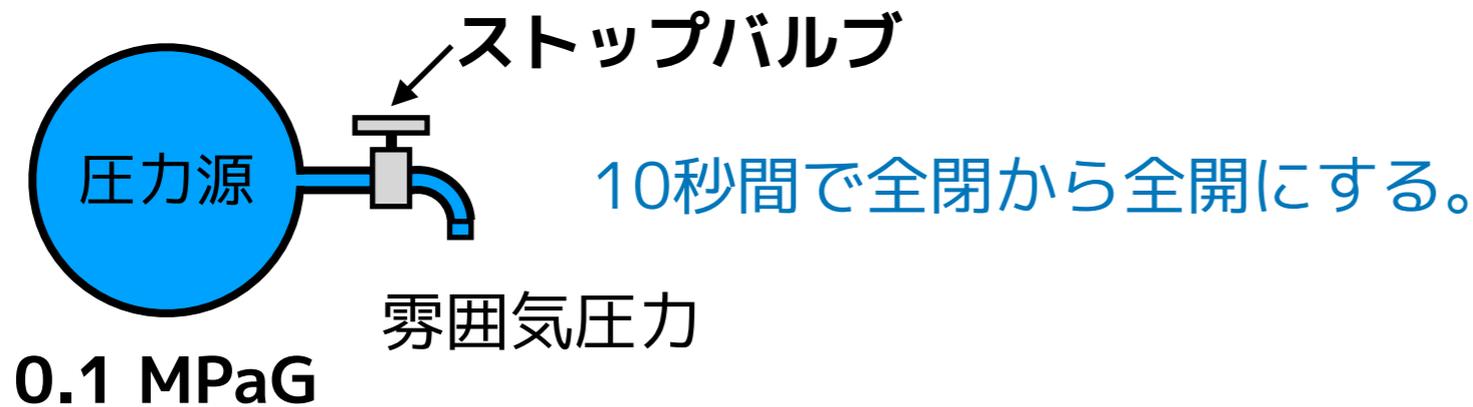
```
Modelica.Blocks.Tables.CombiTable1D valveCharacteristicTable(  
  smoothness = Modelica.Blocks.Types.Smoothness.MonotoneContinuousDerivative1,  
  table = [  
    0, 0; 0.052631579, 0.271493213; 0.157894737, 0.542986425;  
    0.210526316, 0.601809955; 0.263157895, 0.633484163; 0.368421053, 0.791855204;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

## ⑦ valveCharacteristicTable を使うように equation を修正する。

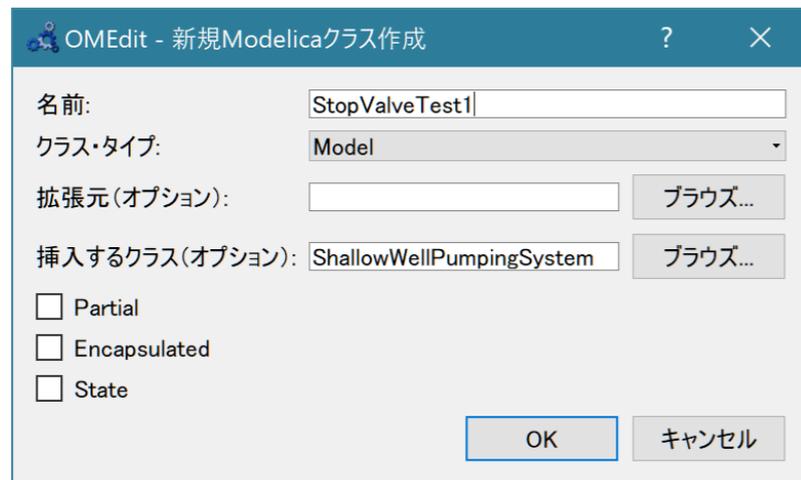
```
equation  
  connect(valveCharacteristicTable.u[1], opening_actual) annotation( ...);  
  // m_flow = valveCharacteristic(opening)*Av*sqrt(d)*sqrt(dp);  
  // relativeFlowCoefficient = valveCharacteristic(opening_actual);  
  relativeFlowCoefficient = valveCharacteristicTable.y[1];  
  if checkValve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

# StopValveTest1 単体テスト

ストップバルブの動作をテストするモデルを作る。



- ① ライブラリブラウザの ShallowWellPumpingSystem を右クリックし、「Modelica クラス新規作成」を選択し、StopValveTest1 を作成する。

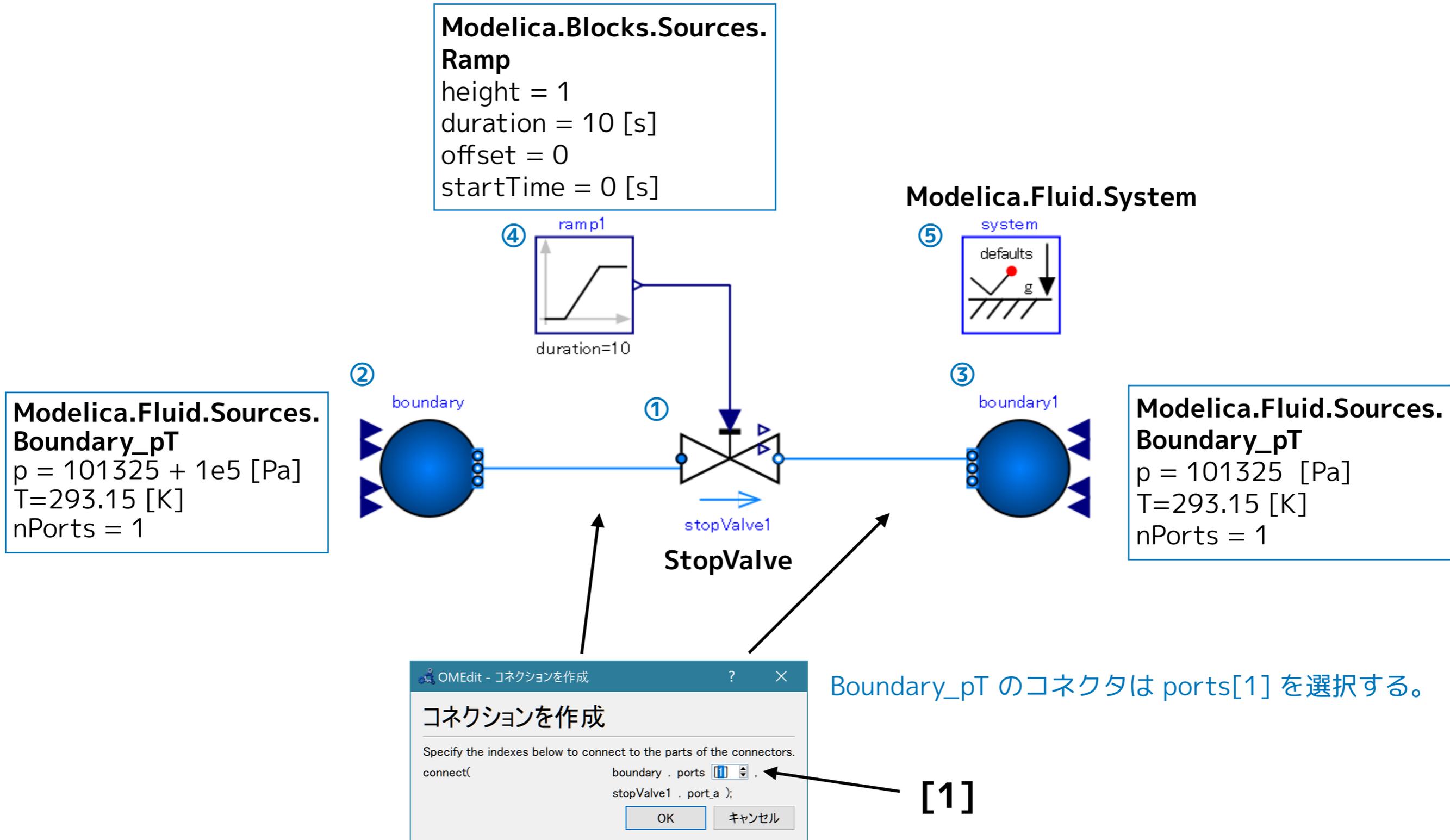


名前: StopValveTest1

クラス・タイプ: Model

挿入するクラス: ShallowWellPumpingSystem

② ダイアグラムビューに、ライブラリブラウザから部品モデルをドラッグ・アンド・ドロップして、パラメータを設定する。



## ③ テキストビューで、Media（流体モデル）の設定を行う。

```
model StopValveTest1
  replaceable package Medium = Modelica.Media.Water.StandardWater;
  StopValve stopValve1(redeclare package Medium = Medium) annotation( ...);
  Modelica.Fluid.Sources.Boundary_pT boundary(
    redeclare package Medium = Medium, T = 293.15, nPorts = 1, p = 101325 + 1e5) annotation( ...);
  Modelica.Fluid.Sources.Boundary_pT boundary1(
    redeclare package Medium = Medium, nPorts = 1, p = 101325) annotation( ...);
  Modelica.Blocks.Sources.Ramp ramp1(duration = 10) annotation( ...);
  inner Modelica.Fluid.System system annotation( ...);
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;
```

①

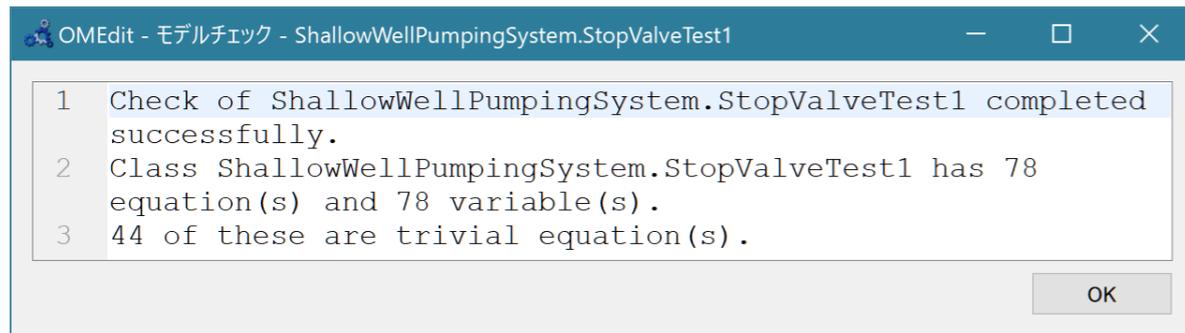
②

③

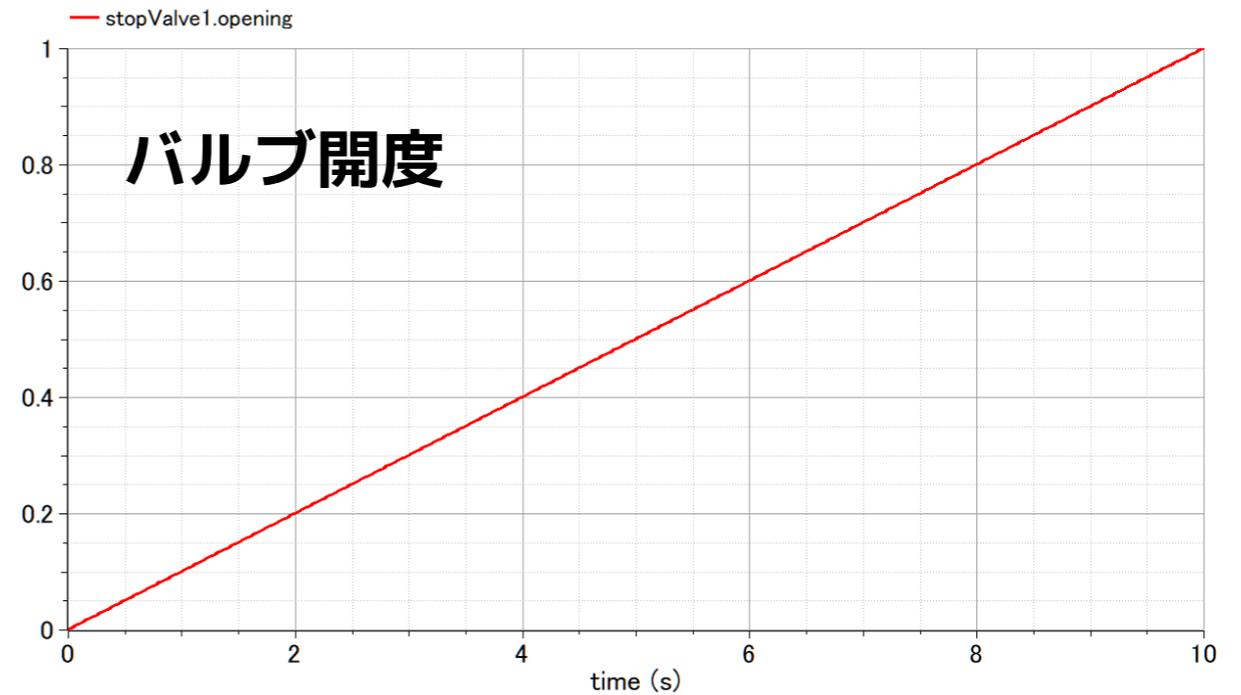
④

⑤

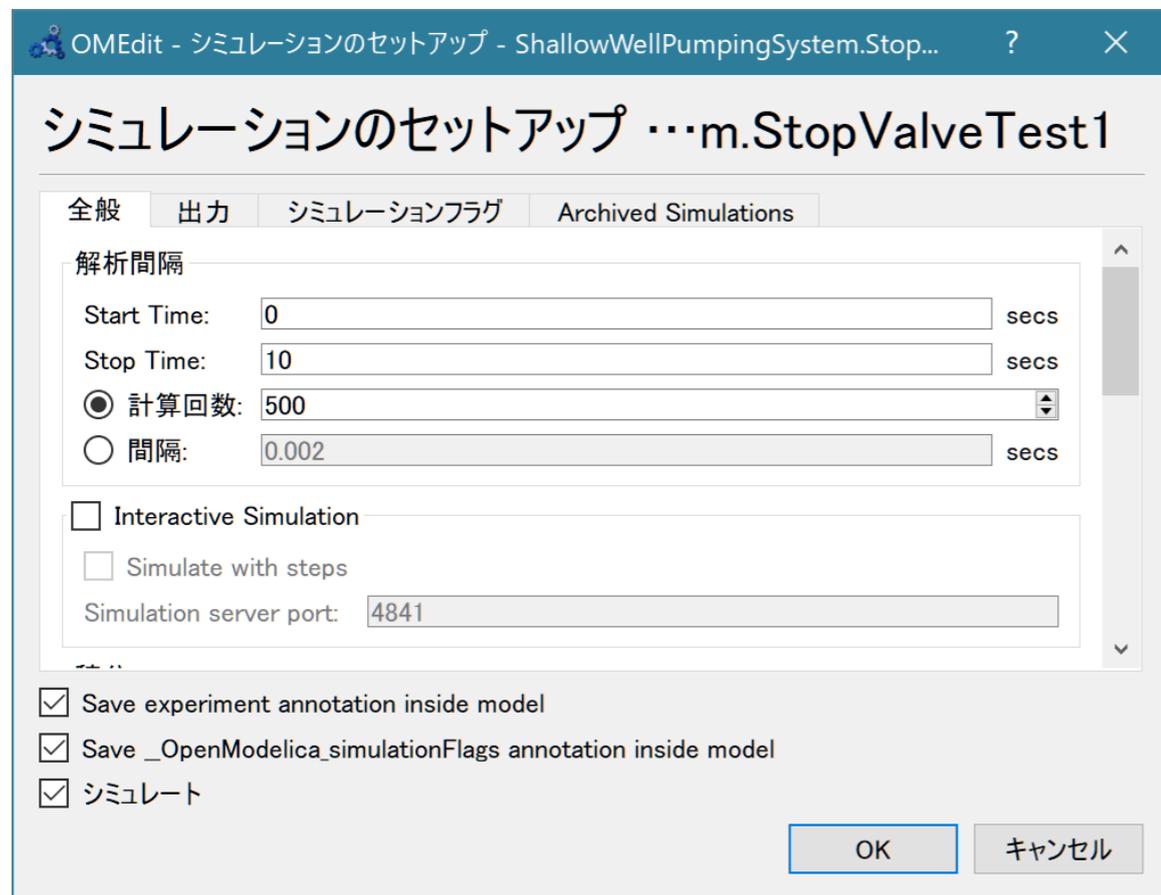
### ④ シミュレーション>モデルチェック



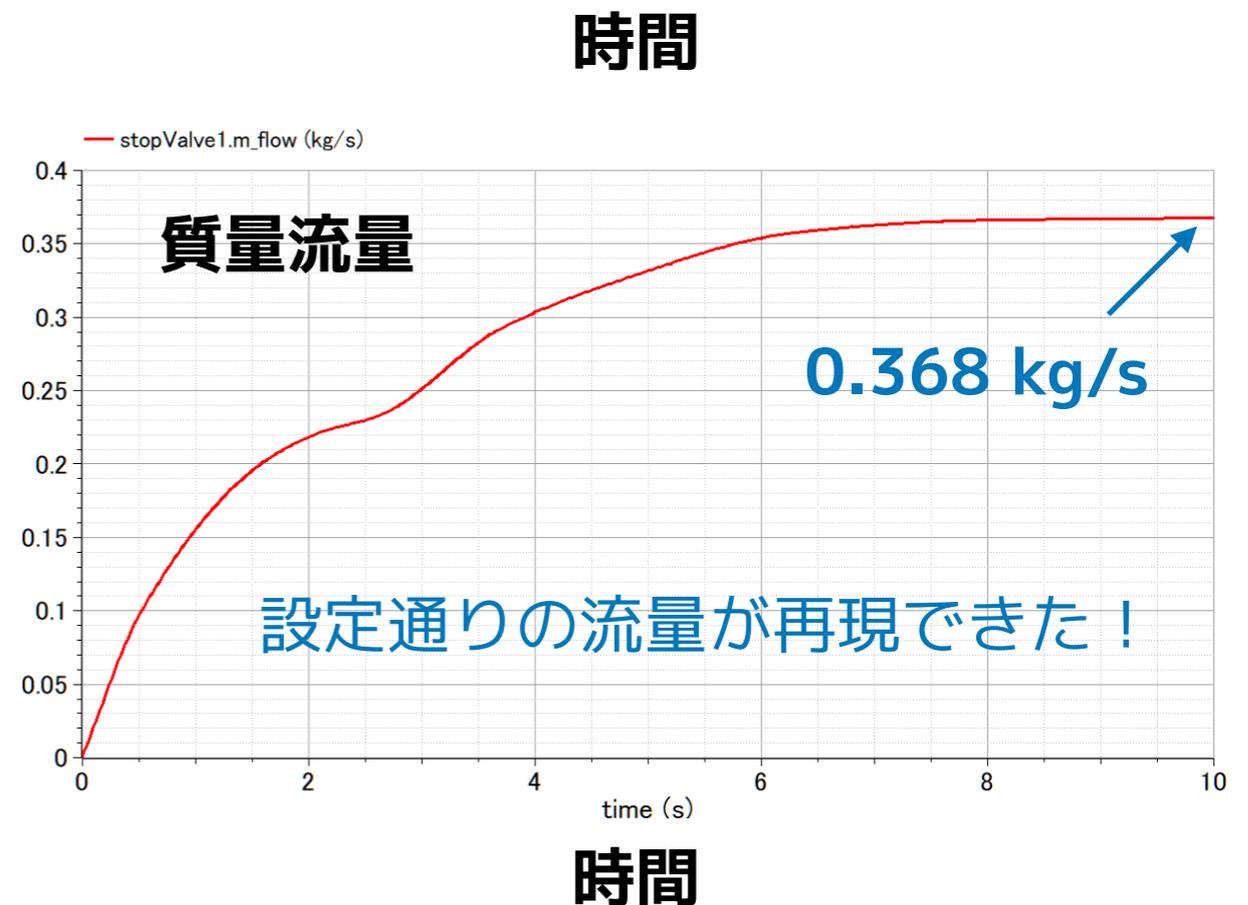
### シミュレーション結果



### ⑤ シミュレーション>シミュレーションのセットアップ



Stop Time = 10 [secs]



# StopValveTest2 ストップバルブとオープンタンク

① StopValveTest1 を複製して StopValveTest2 を作成し、上流の部品を次のように変更する。

## オープンタンク

- 初期水位 35 m
- 0秒から10秒でストップバルブを全閉から全開にする。

## ストップバルブ

## tank

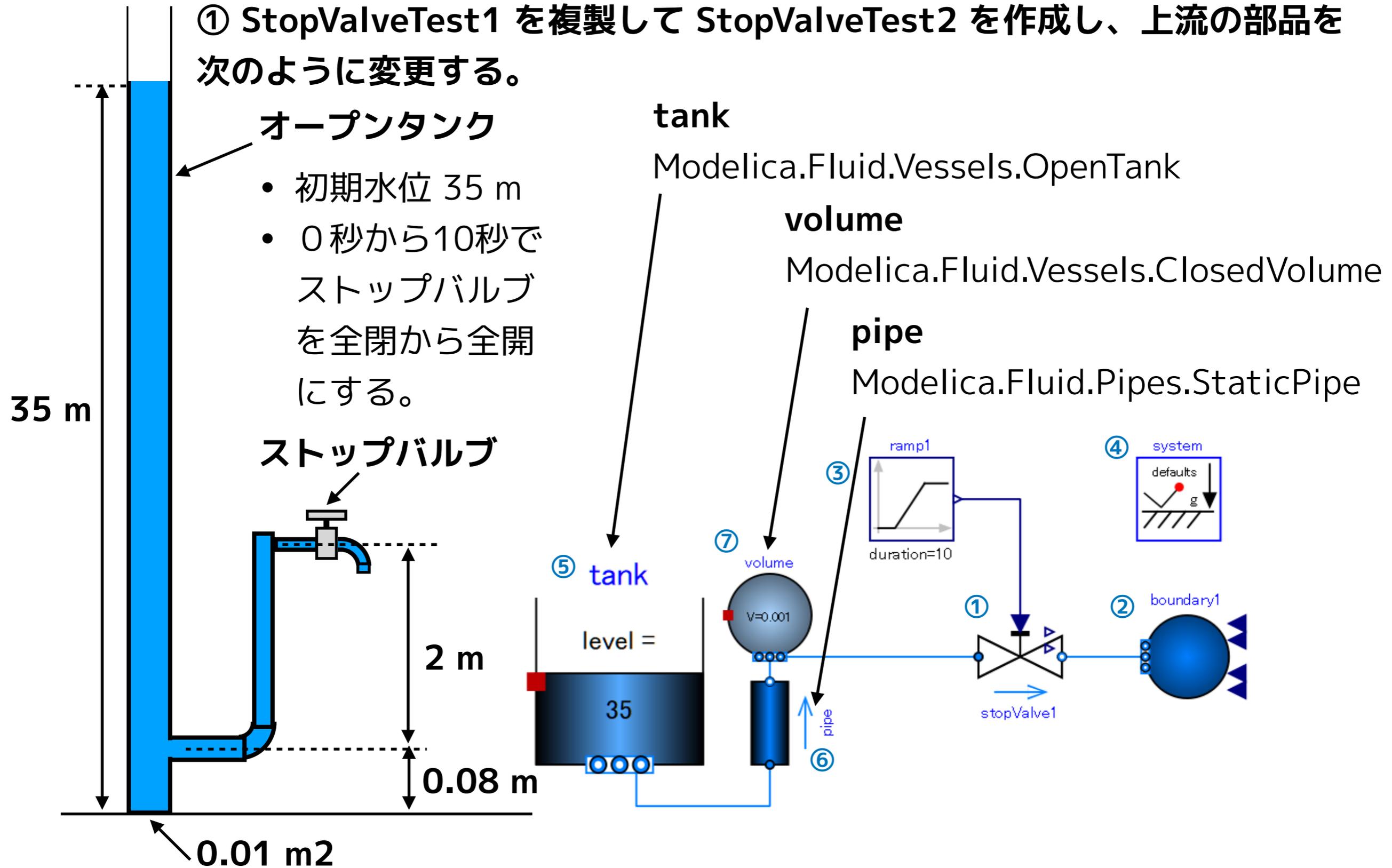
Modelica.Fluid.Vessels.OpenTank

## volume

Modelica.Fluid.Vessels.ClosedVolume

## pipe

Modelica.Fluid.Pipes.StaticPipe



## ② 変更した部品のパラメータを設定する。

### 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 = Modelica.Fluid.Types.Dynamics.FixedInitial
massDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial
```

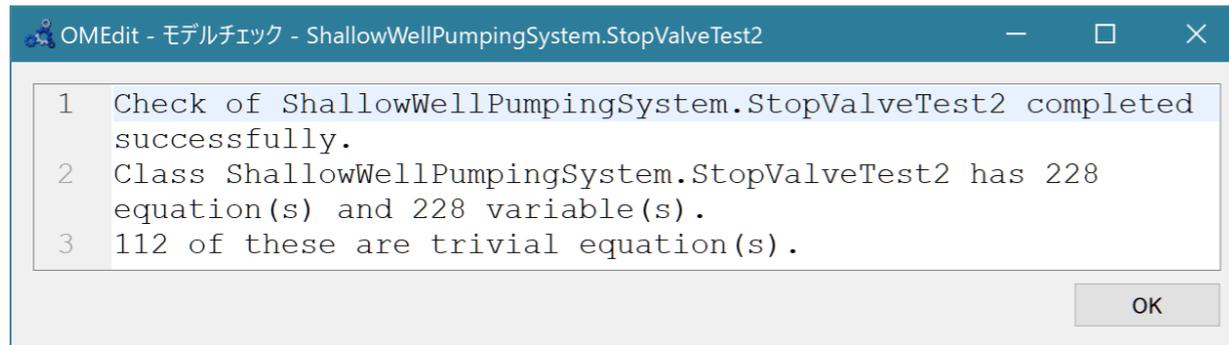
### pipe

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

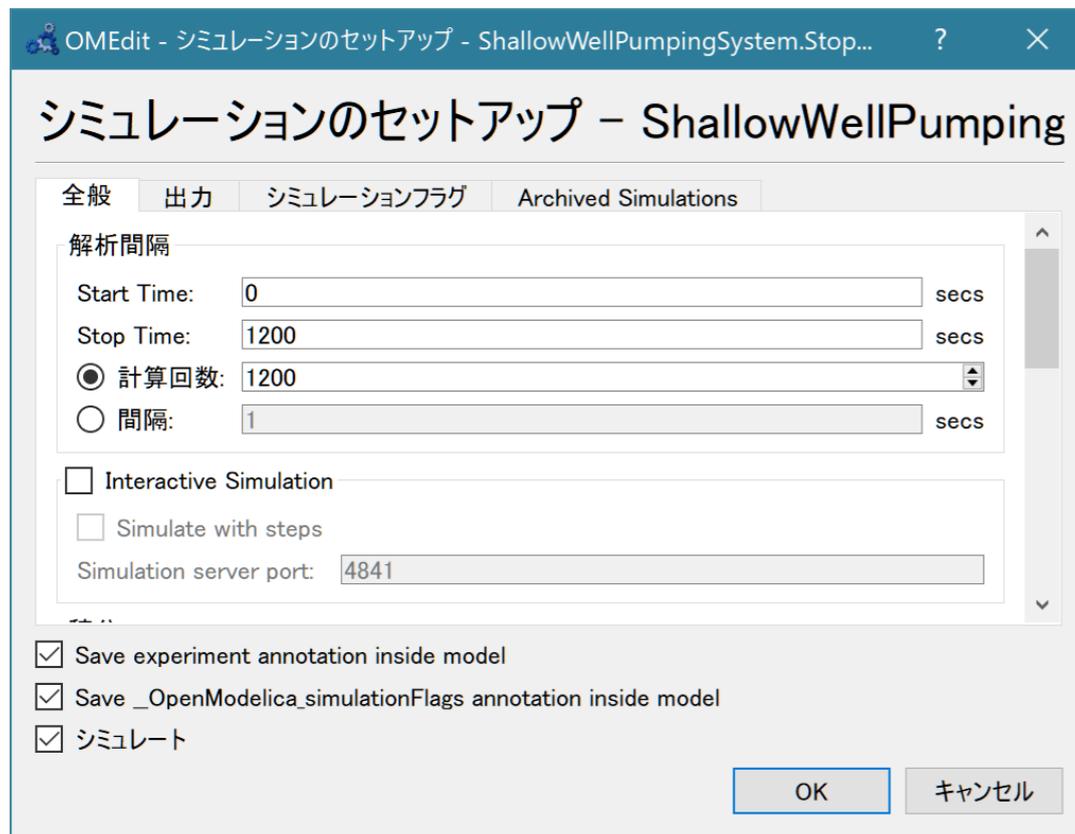
### ③ テキストビューで Media(流体モデル)を設定する。

```
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 シミュレーション>モデルチェック

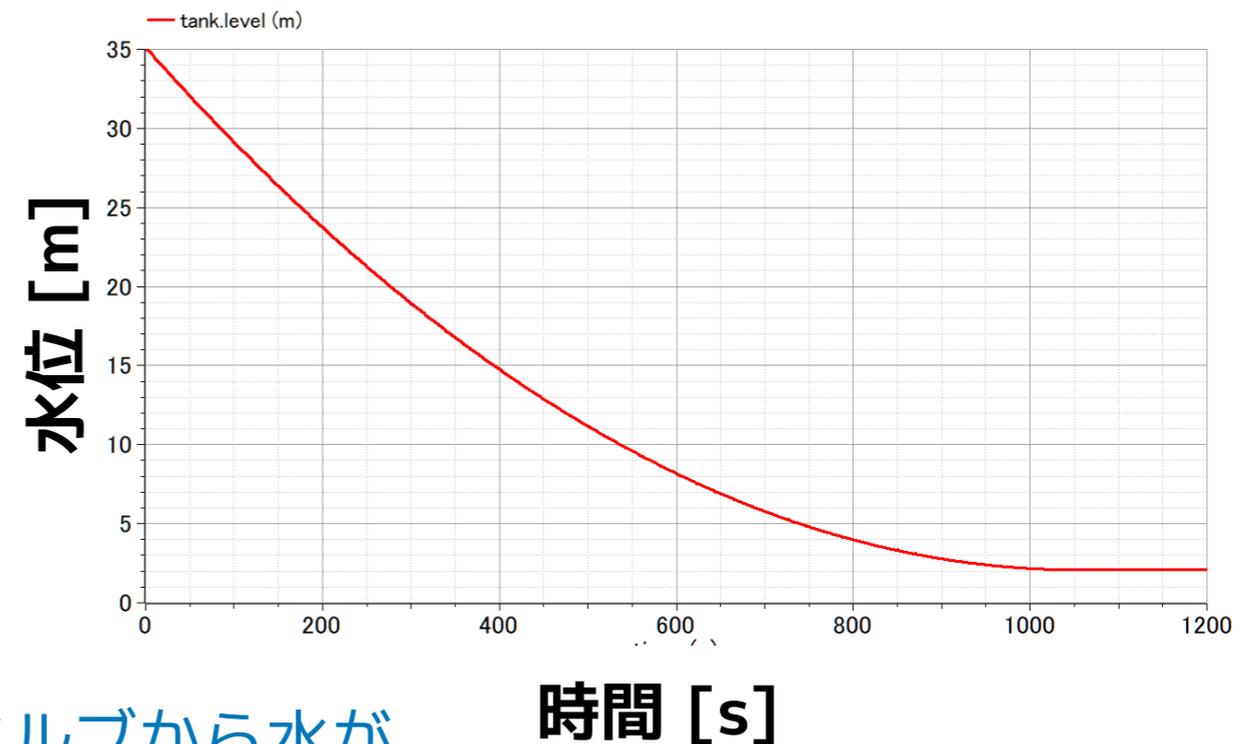
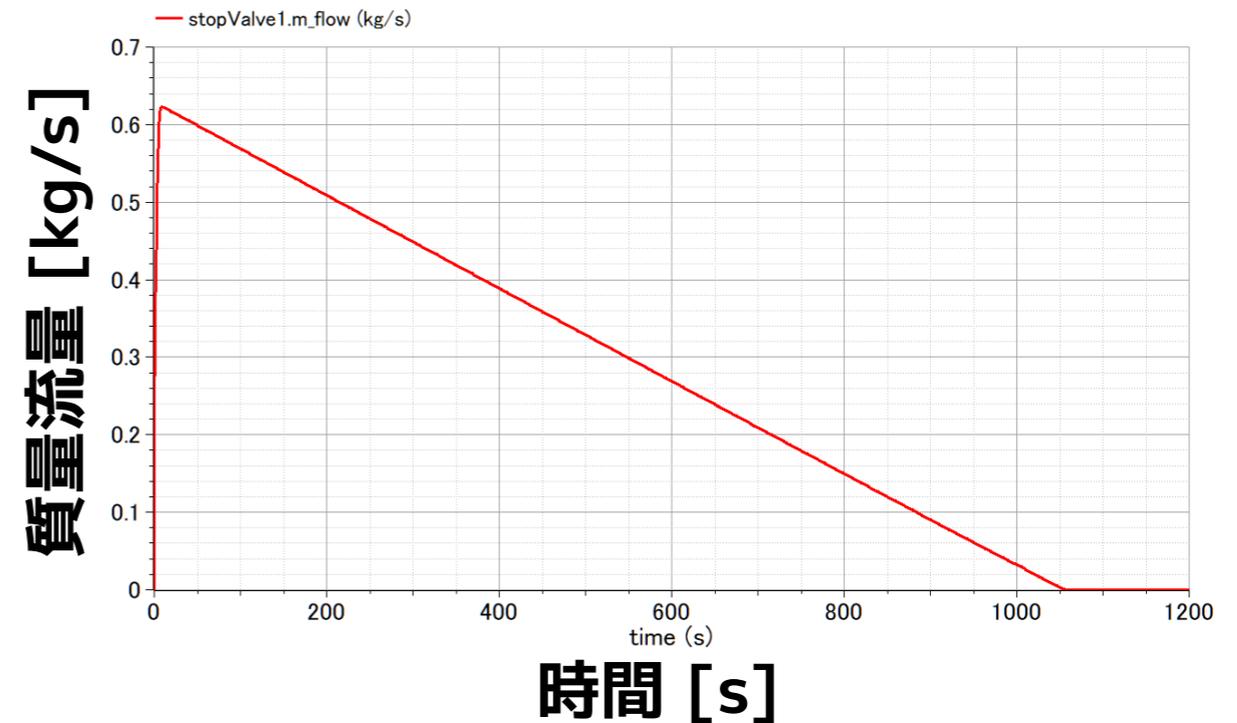


### ⑤ シミュレーション>シミュレーションのセットアップ



Start Time: 0 [s]  
 Stop Time: 1200 [s]  
 計算回数: 1200

### シミュレーション結果



ストップバルブから水が流出するとともに水位が低下し、流量も減少する。

# ポンプ

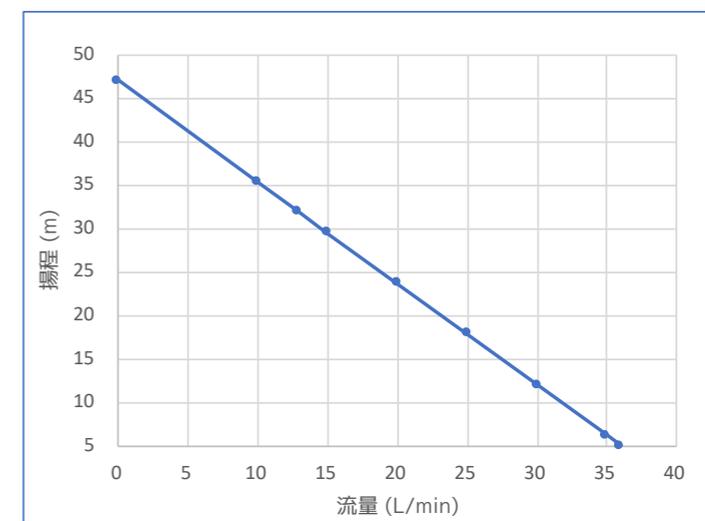
参考 荏原製作所 25HPO5.25S <https://product-standard-pump.ebara.com/product/detail/P030766>

## ポンプ動力（モーター）

周波数	f	50	Hz
極数	p	2	
回転数	n	3000	rpm

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

$\rho$	密度(25°C)	997.062	kg/m <sup>3</sup>
$g$	重力加速度	9.80665	m/s <sup>2</sup>



## ポンプ仕様

電源	電動機 呼び出力	許容押込 高さ	最大吸上 高さ	給水量	揚程	始動圧力	吸込 フランジ	吐出 フランジ
V	W	m	m	L/min	m	kPa	mm	mm
100	<b>250</b>	2	-8	<b>30</b>	<b>12</b>	140/200	25	25

<i>head</i>		<i>q</i>	$\Delta p$	$W_h$	$W_m$	$\eta$
流量	揚程	流量	昇圧	水動力	軸動力	効率
L/min	m	m <sup>3</sup> /min	Pa	W	W	-
<b>0.000</b>	<b>47.000</b>	0.0000E+00	4.5956E+05	0.000	100.000	0.00000
10.000	35.333	1.0000E-02	3.4548E+05	57.581	141.667	0.40645
12.857	<b>32.000</b>	1.2857E-02	3.1289E+05	67.048	153.571	0.43659
15.000	29.500	1.5000E-02	2.8845E+05	72.112	162.500	0.44376
20.000	23.667	2.0000E-02	2.3141E+05	77.136	183.333	0.42074
25.000	17.833	2.5000E-02	1.7437E+05	72.655	204.167	0.35586
<b>30.000</b>	<b>12.000</b>	3.0000E-02	1.1733E+05	58.667	225.000	0.26074
35.000	6.167	3.5000E-02	6.0297E+04	35.173	245.833	0.14308
<b>36.000</b>	<b>5.000</b>	3.6000E-02	4.8889E+04	29.334	<b>250.000</b>	0.11733
40.286	0.000	4.0286E-02	0.0000E+00	0.000	267.857	0.00000

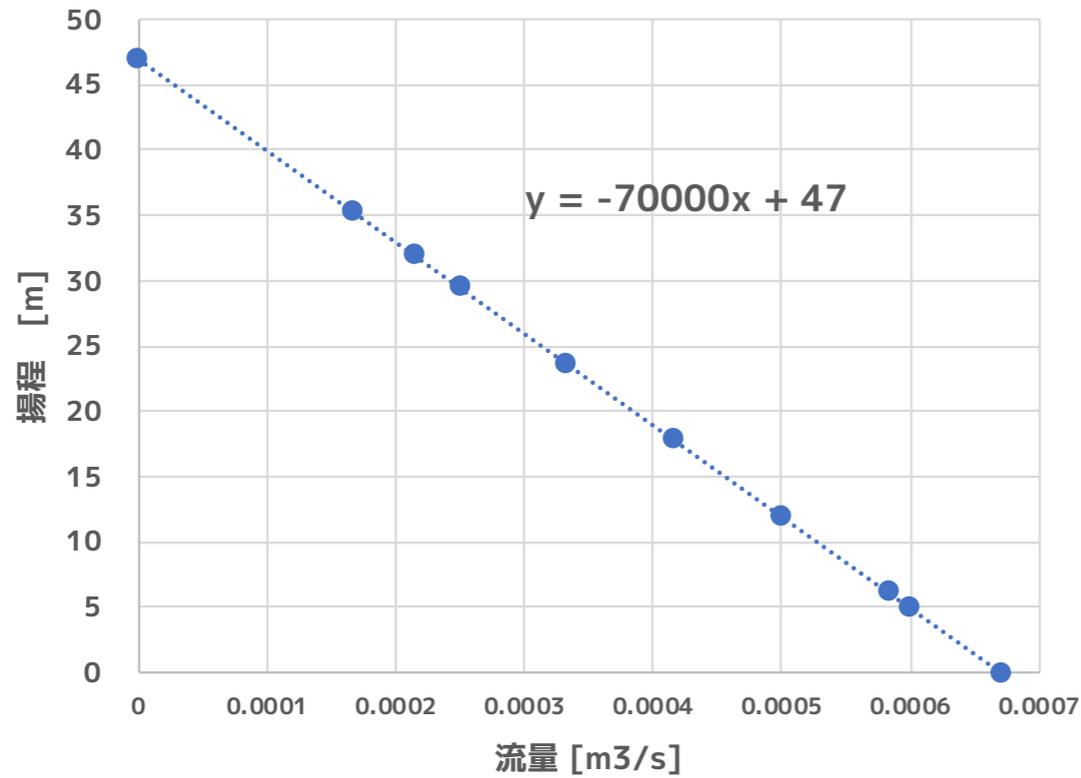
$$\Delta p = \rho g \cdot head$$

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

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

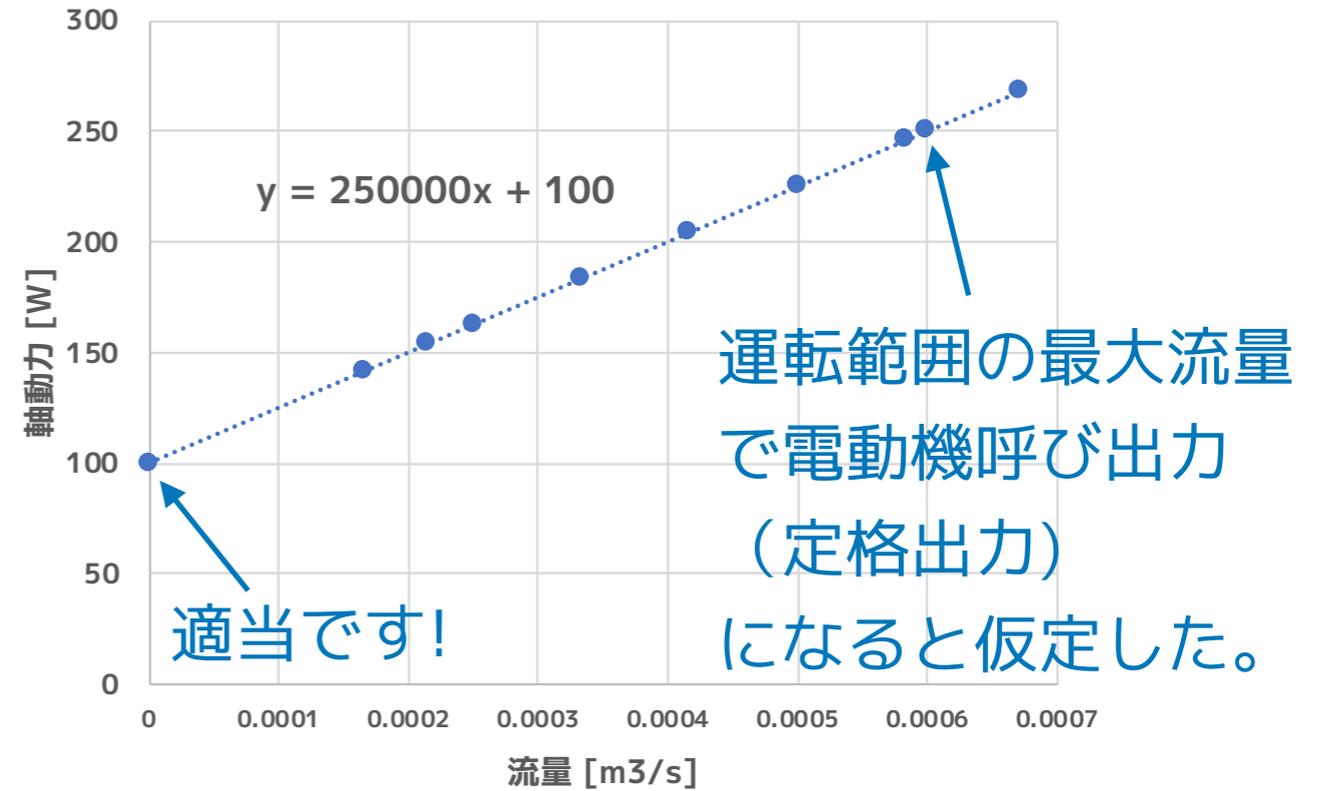
運転範囲

軸動力は、運転範囲の軸動力の最大値が電動機呼び出力になると仮定して適当に作成した

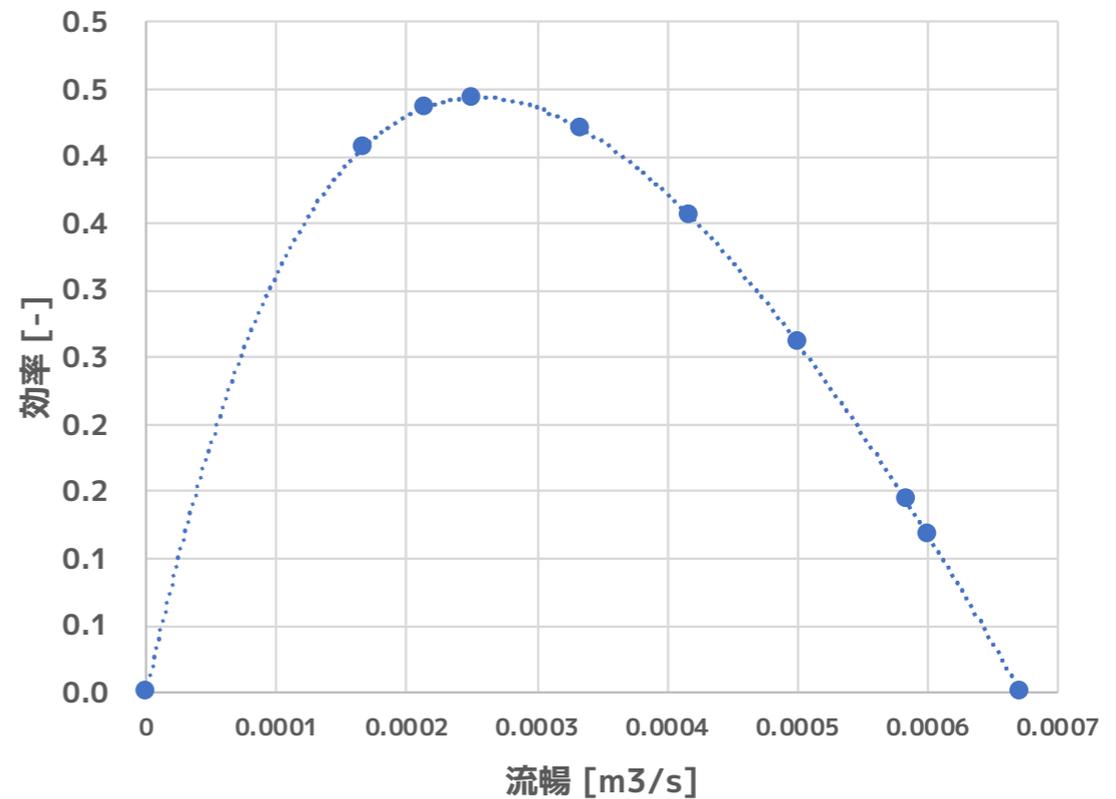


揚程特性 (head)

Excel の近似曲線機能で  
計算式を求めた。



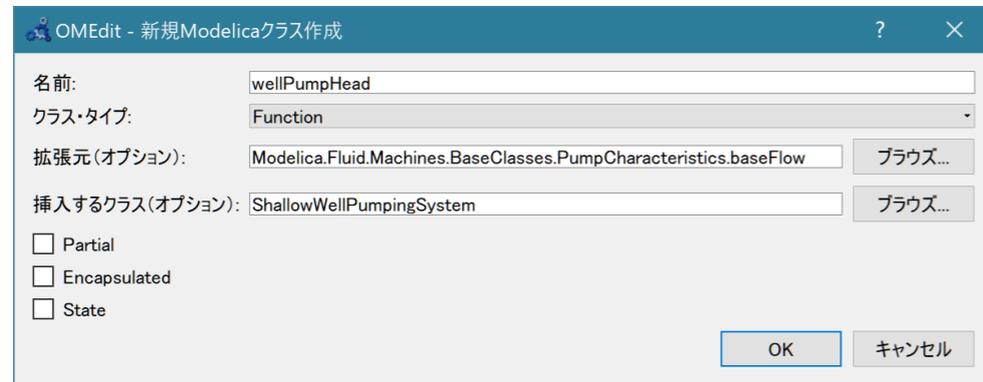
軸動力特性 (shaft power)



効率特性 (efficiency)

# 揚程特性 (head curve)

① ライブラリブラウザの ShallowWellPumpingSystem を右クリックして Modelica クラス新規作成を選択する。



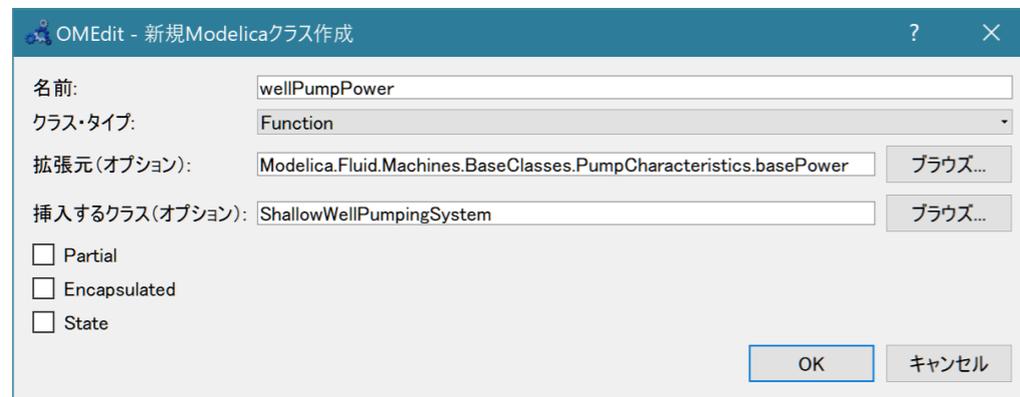
名前: wellPumpHead  
クラス・タイプ: Function  
拡張元: Modelica.Fluid.Machines.BaseClasses.PumpCharacteristic.baseFlow  
挿入するクラス: ShallowWellPumpingSystem

② テキストビューで、揚程特性の計算式を入力する。

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

# 軸動力特性 (shaft power curve)

① ライブラリブラウザの ShallowWellPumpingSystem を右クリックして Modelica クラス新規作成を選択する。



名前: wellPumpPower

クラス・タイプ: Function

拡張元: Modelica.Fluid.Machines.BaseClasses.PumpCharacteristic.basePower

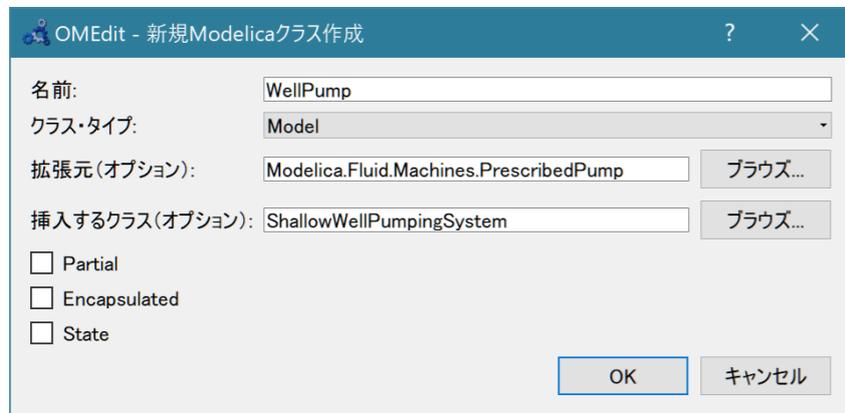
挿入するクラス: ShallowWellPumpingSystem

② テキストビューで、軸動力特性の計算式を入力する。

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

# WellPump モデルの作成

① ライブラリブラウザの ShallowWellPumpingSystem を右クリックして Modelica クラス新規作成を選択する。



名前: WellPump  
クラス・タイプ: Model  
拡張元: Modelica.Fluid.Machines.PrescribedPump  
挿入するクラス: ShallowWellPumpingSystem

② テキストビューで、flowCharacteristic, powerCharacteristic, パラメータを設定する。

```
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);
```

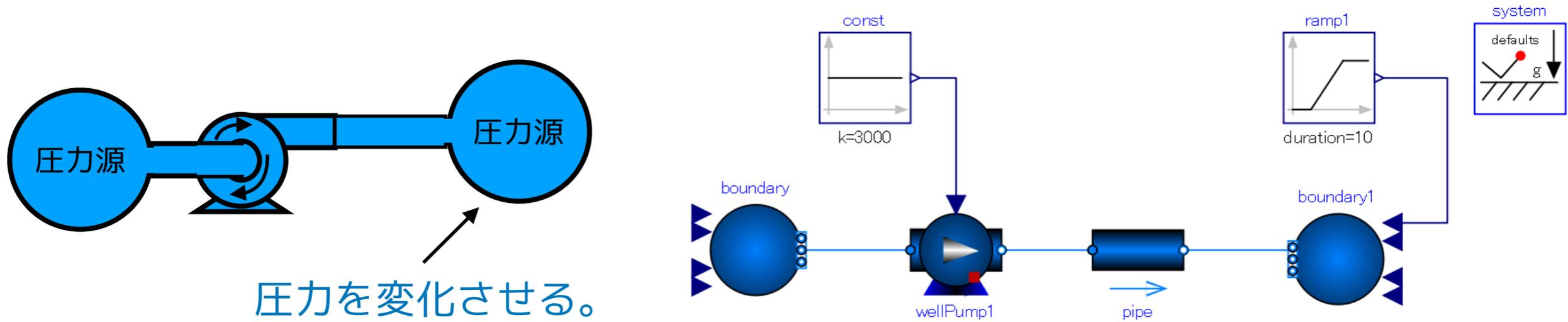
```
equation
```

```
end WellPump
```

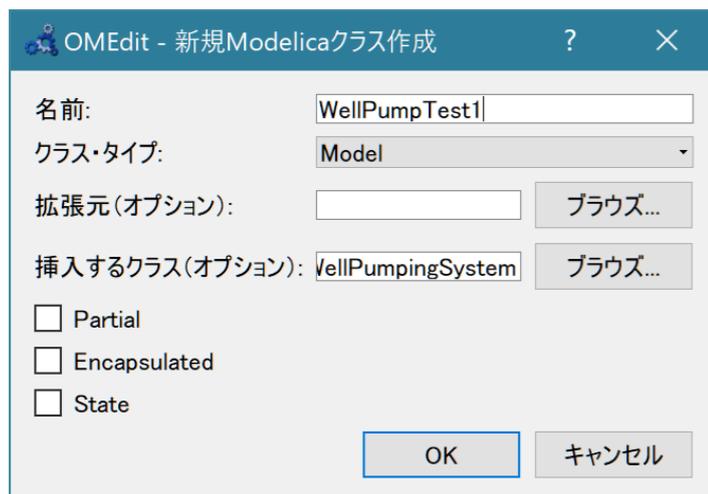
ポンプ特性の設定

# WellPumpTest1 単体テスト

WellPump のテスト用モデルを作成する。

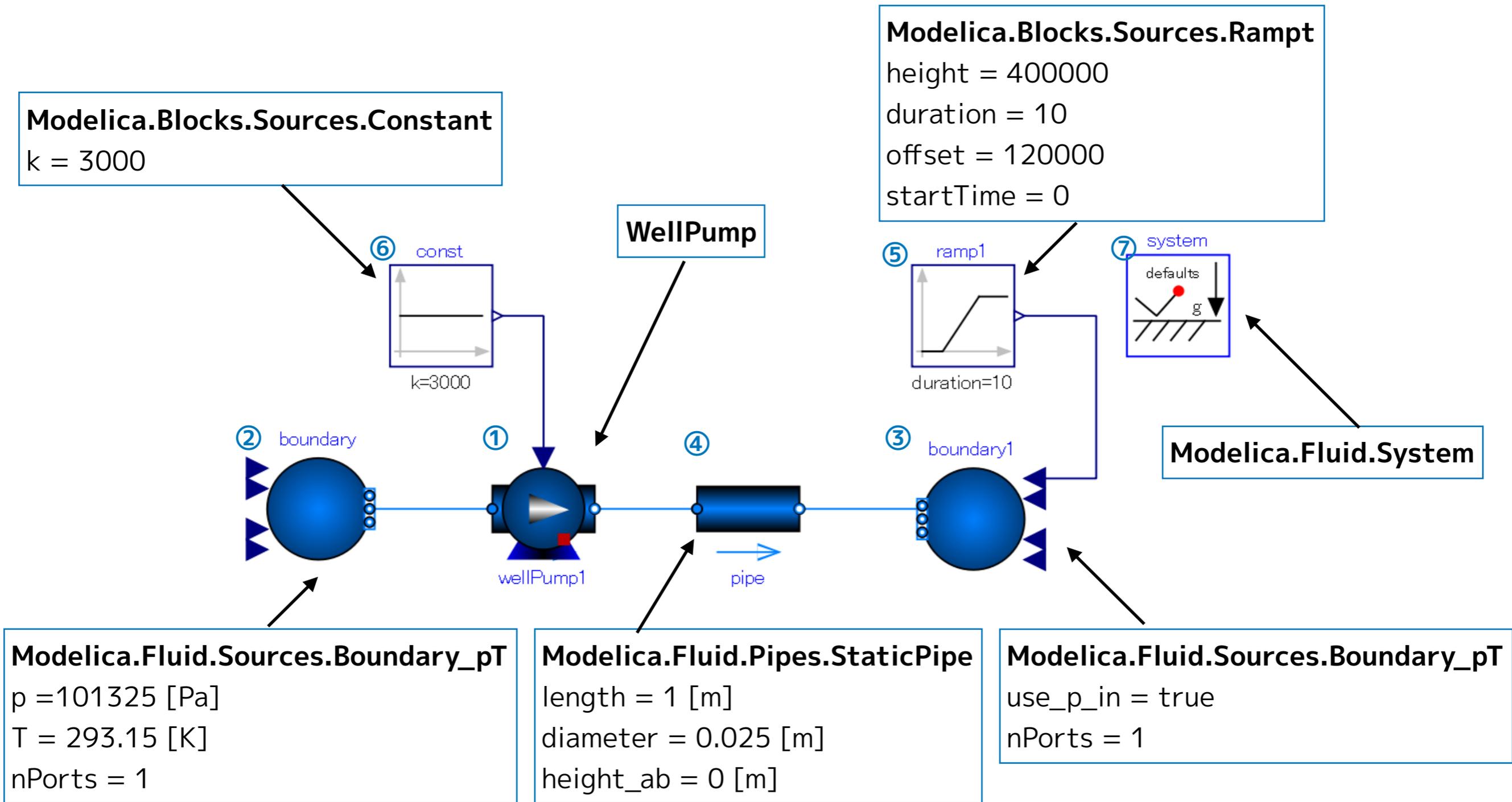


① ライブラリブラウザの ShallowWellPumpingSystem を右クリックして Modelica クラス新規作成を選択する。



名前: WellPumpTest  
クラス・タイプ: Model  
挿入するクラス: ShallowWellPumpingSystem

② ダイアグラムビューに部品を配置して、パラメータを設定する。



Boundary\_pT のコネクタは ports[1] を選択する。

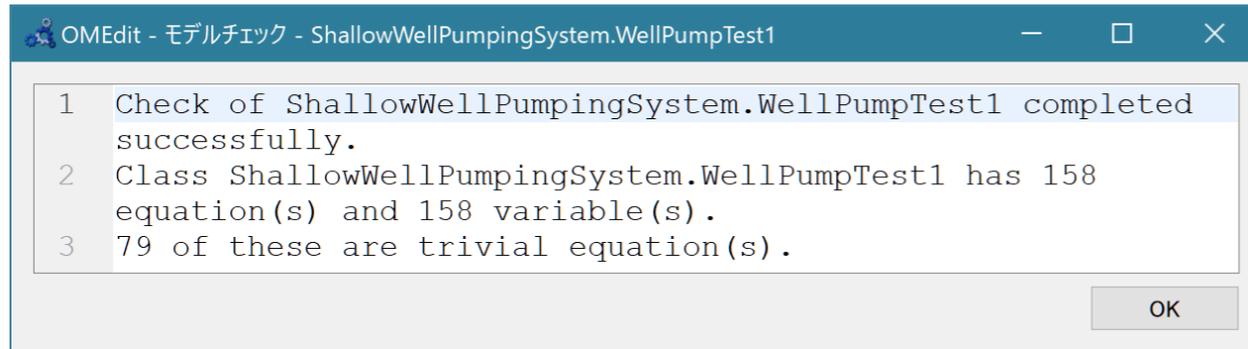


[1]

## ③ テキストビューでソースコードを編集して、Media（流体モデル）を設定する。

```
model WellPumpTest1
  replaceable package Medium = Modelica.Media.Water.StandardWater;
  ShallowWellPumpingSystem.WellPump wellPump1(redeclare package Medium = Medium) annotation( ...); ①
  Modelica.Fluid.Sources.Boundary_pT boundary(
    redeclare package Medium = Medium, T = 293.15, nPorts = 1, p = 101325) annotation( ...); ②
  Modelica.Fluid.Sources.Boundary_pT boundary1(
    redeclare package Medium = Medium, nPorts = 1, use_p_in = true) annotation( ...); ③
  Modelica.Fluid.Pipes.StaticPipe pipe(
    redeclare package Medium = Medium, diameter = 0.025, length = 1) annotation( ...); ④
  Modelica.Blocks.Sources.Ramp ramp1(
    duration = 10, height = 400000, offset = 120000, startTime = 0) annotation( ...); ⑤
  Modelica.Blocks.Sources.Constant const(k = 3000) annotation( ...); ⑥
  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 シミュレーション>モデルチェック



Start Time = 0 [s]

Sop Time = 10 [s]

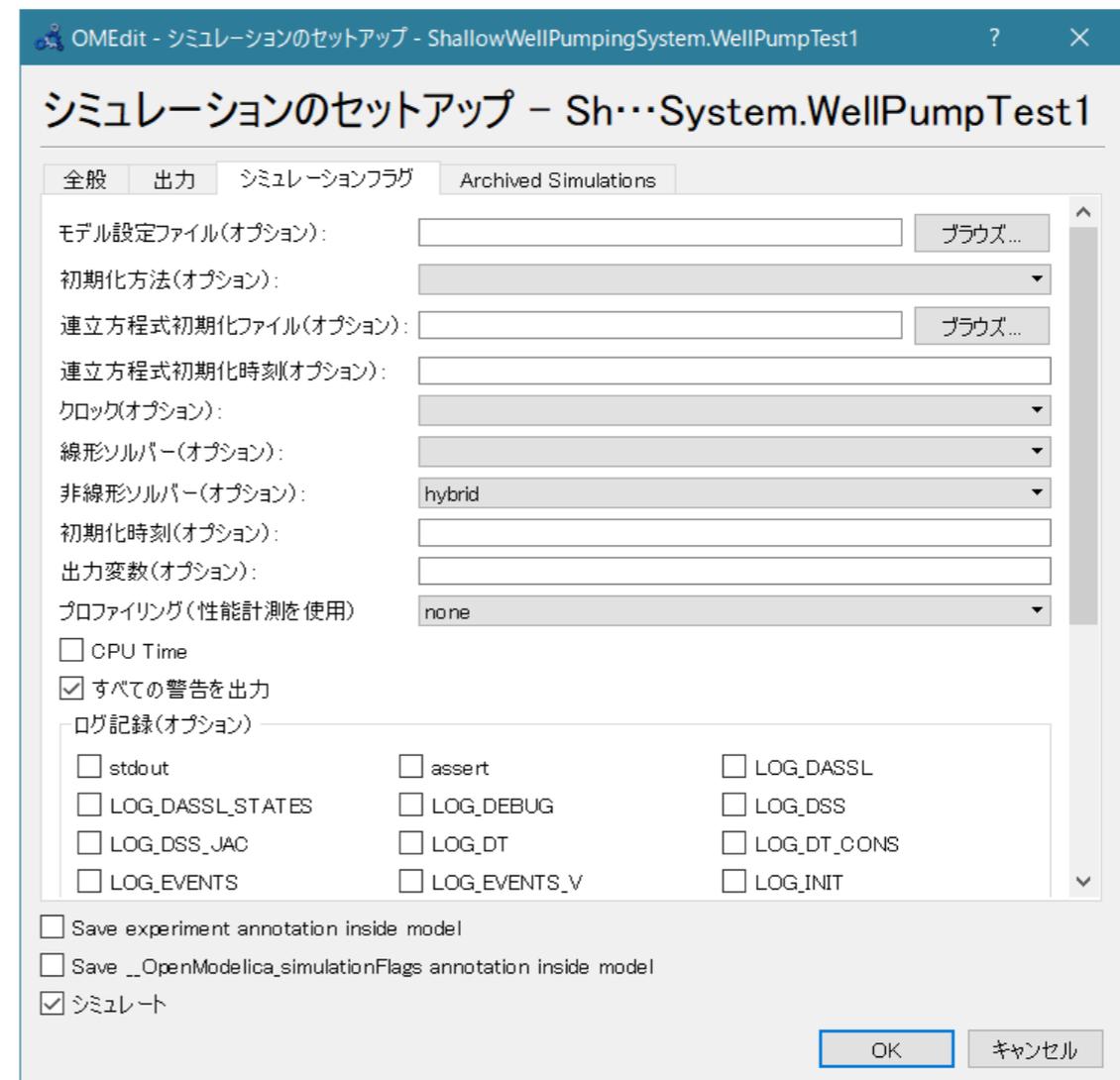
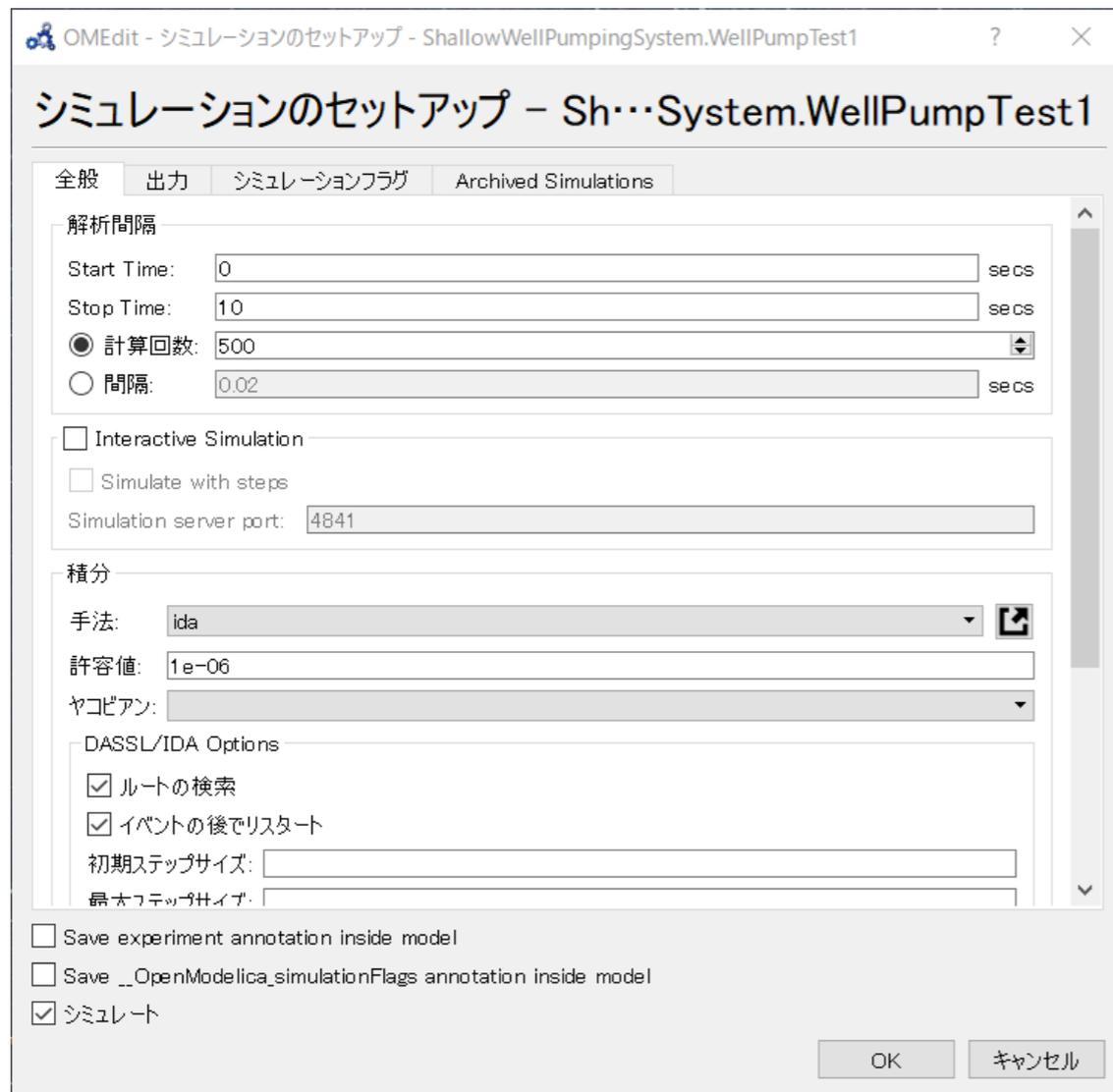
計算回数 = 500

手法: ida

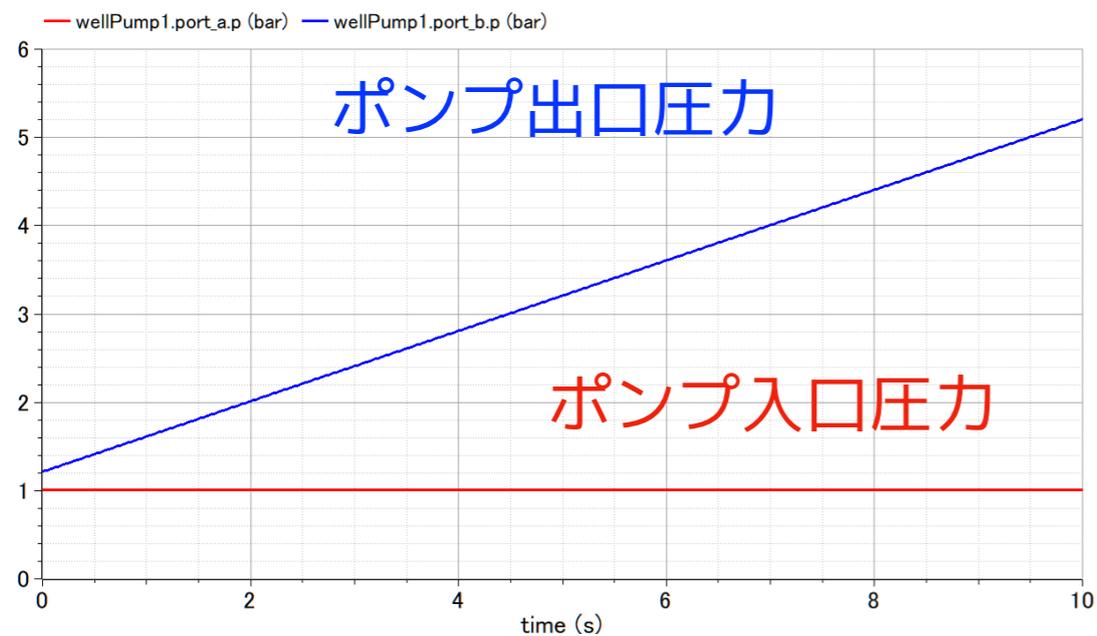
非線形ソルバーオプション: hybrid

## ⑤ シミュレーション>

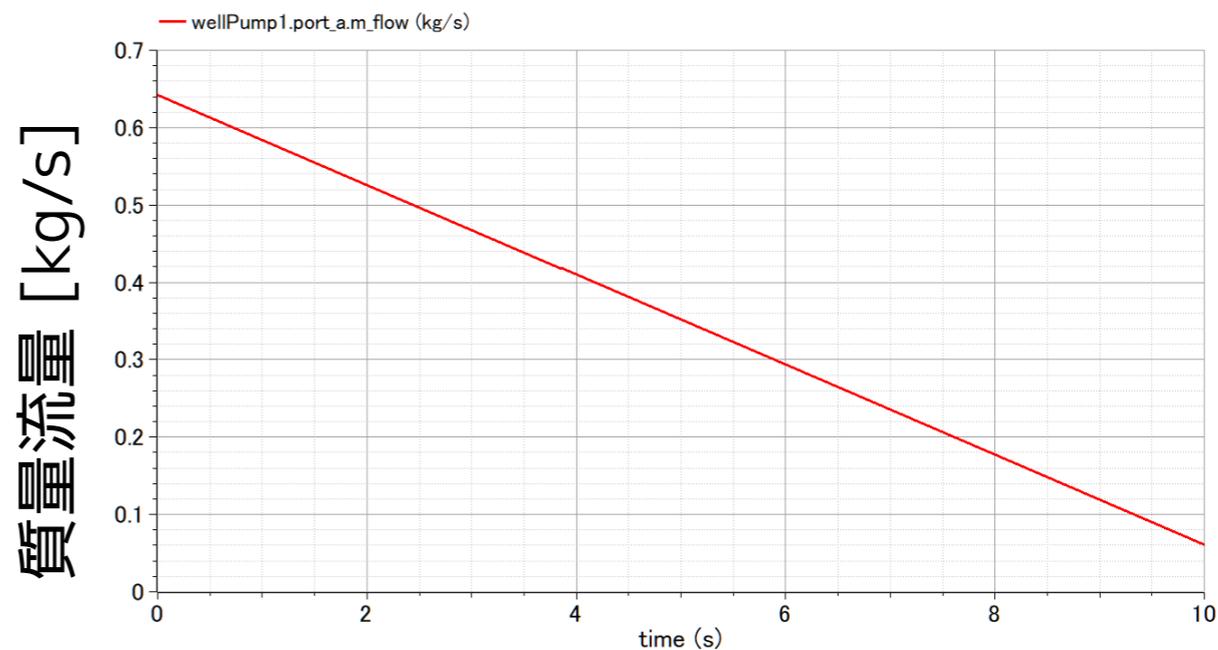
## シミュレーションのセットアップ



# シミュレーション結果

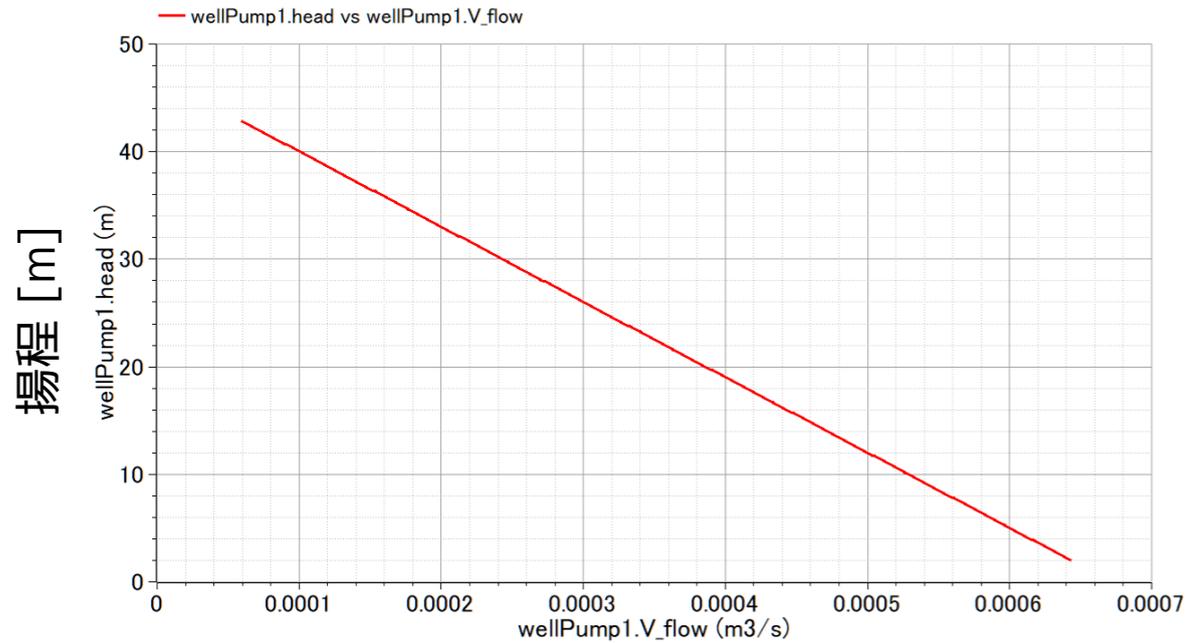


時間 [s]

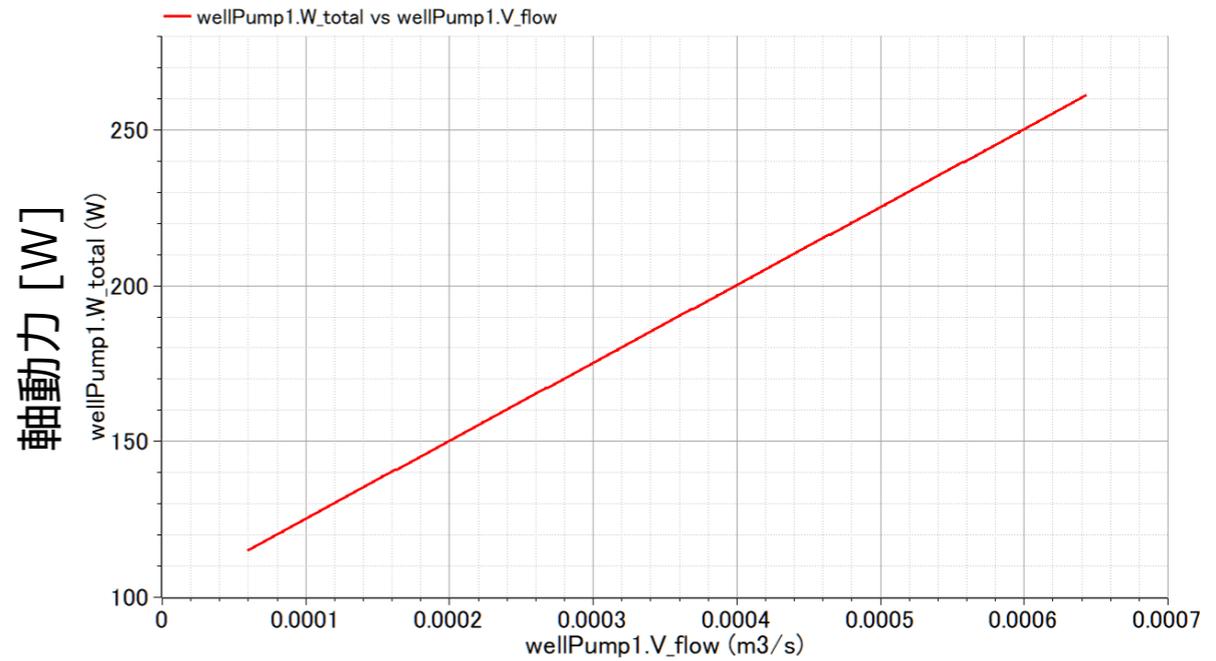


時間 [s]

# シミュレーションの結果

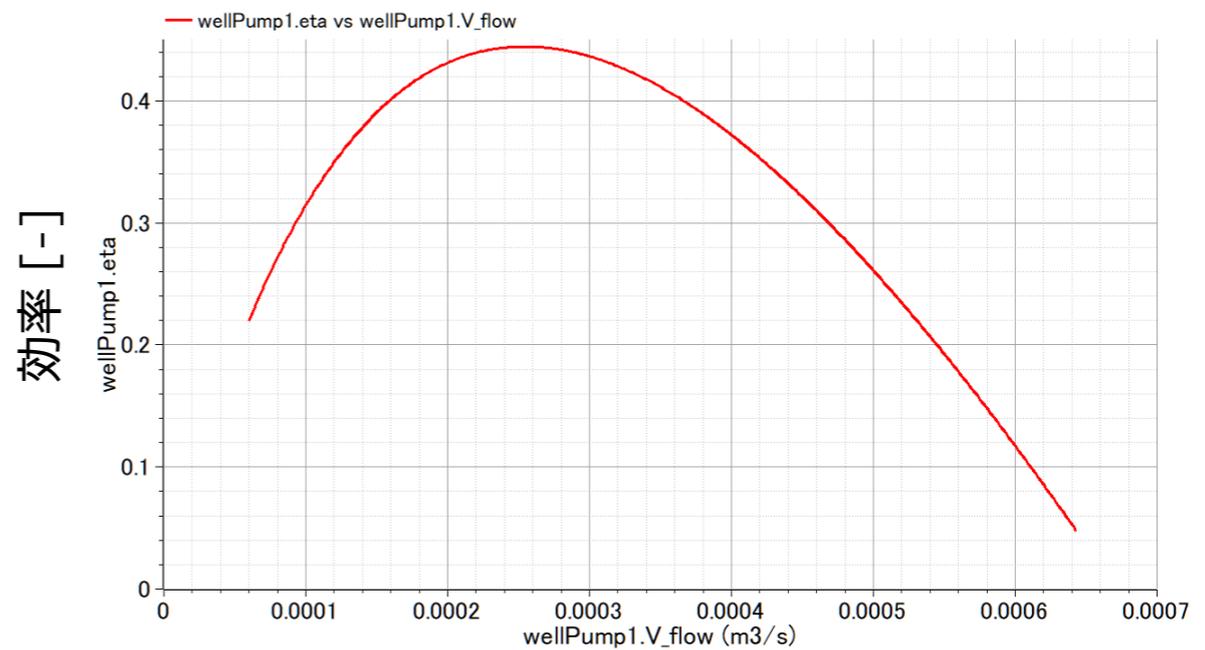


体積流量 [m<sup>3</sup>/s]  
揚程特性



体積流量 [m<sup>3</sup>/s]  
軸動力特性

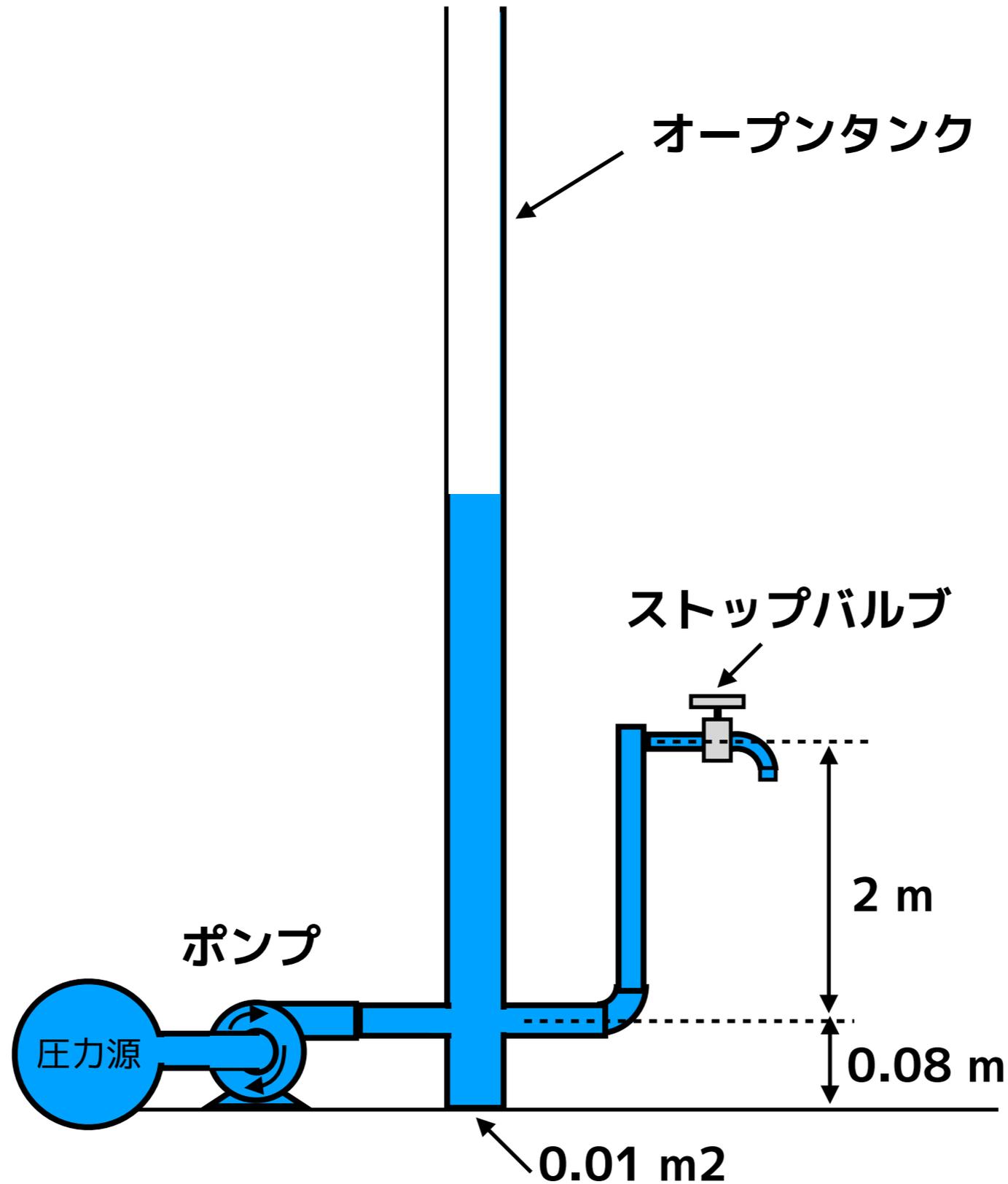
ポンプ性能が再現できた！



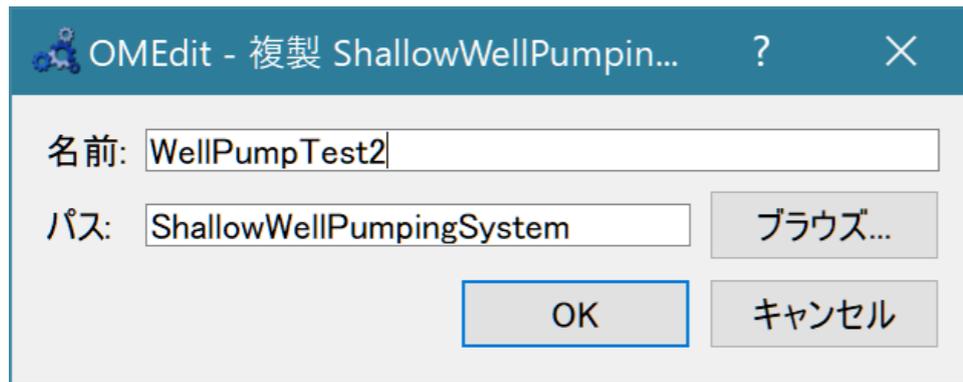
体積流量 [m<sup>3</sup>/s]  
効率特性

# WellPumpTest 2

## ポンプ+オープンタンク+ストップバルブ



① StopValveTest2 を右クリックして、「複製」を選択する。



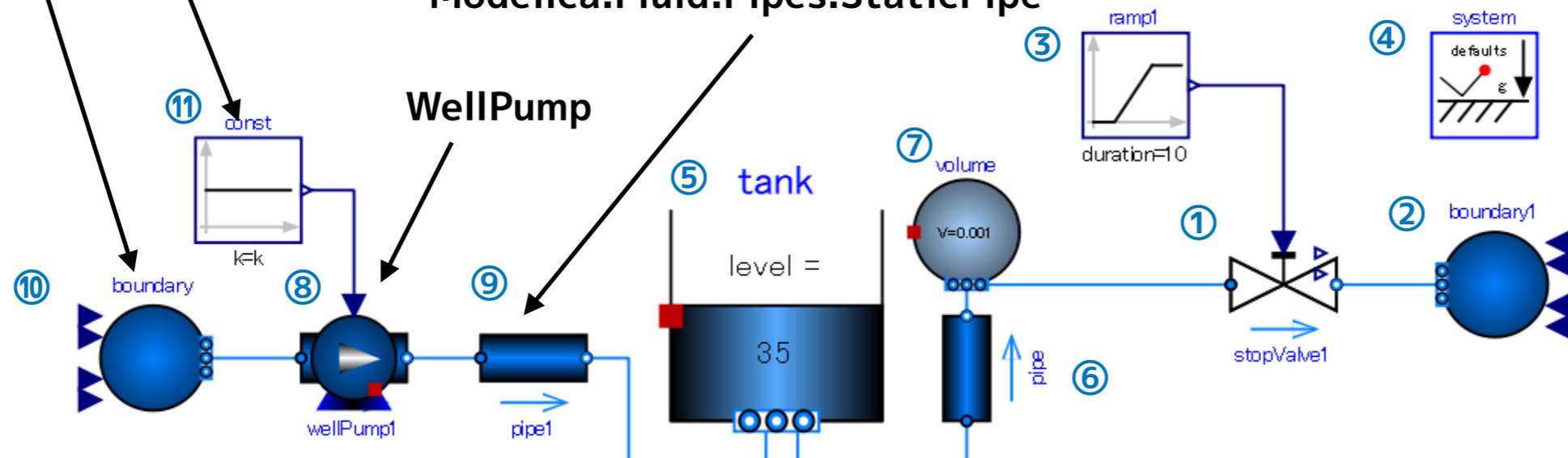
名前: WellPumpTest2  
パス: ShallowWellPumpingSystem

② 部品を配置して接続する。

Modelica.Fluid.Sources.Boundary\_pT

Modelica.Blocks.Sources.Constant

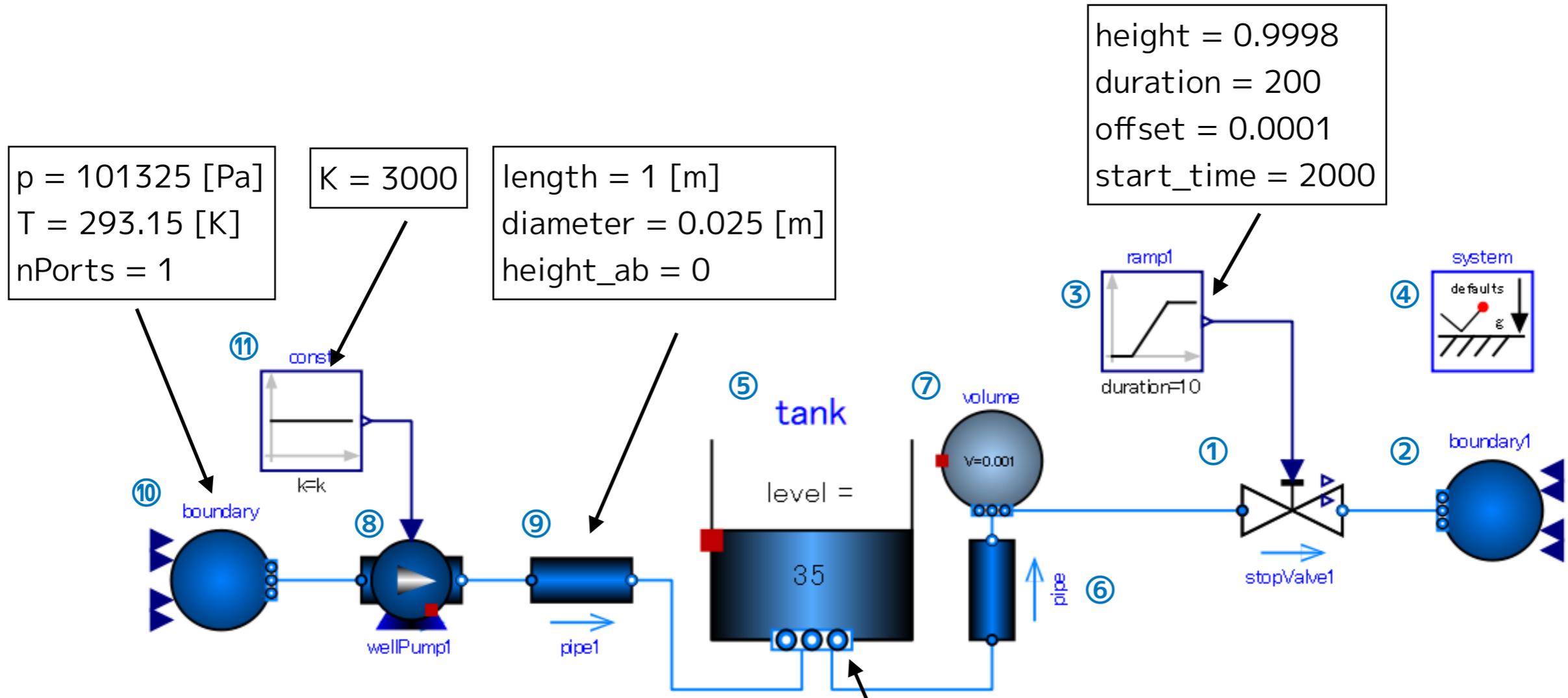
Modelica.Fluid.Pipes.StaticPipe



[1]

[2]

③ パラメータを設定する。



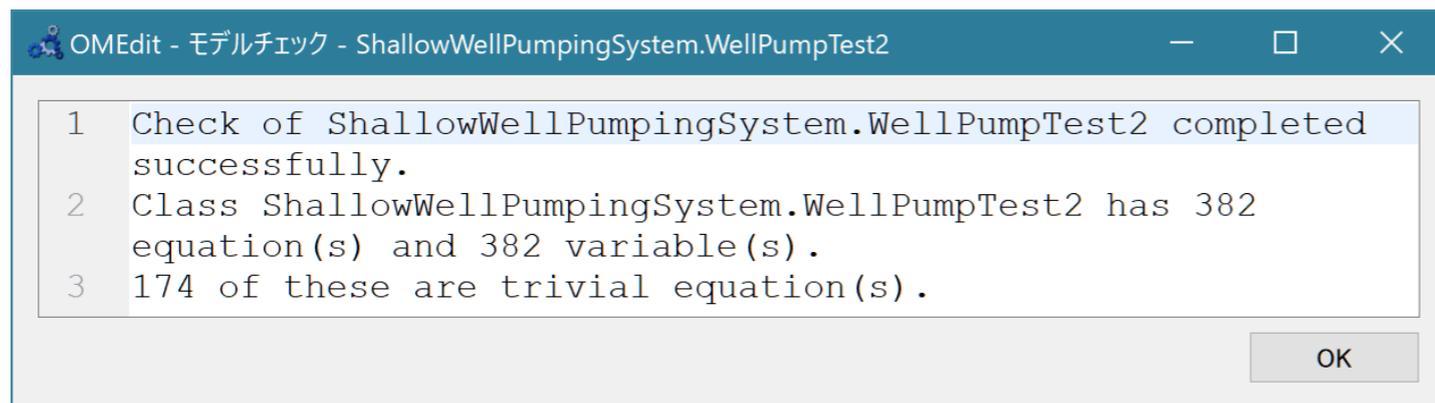
次スライドを参考に  
テキストビューでパラメータを修正する。

④ テキストビューで、Media (流体モデル) の設定を行う。  
OpenTankの level\_start, nPorts, portsData も修正する。

```
model WellPumpTest2
  replaceable package Medium = Modelica.Media.Water.StandardWater;
  ShallowWellPumpingSystem.StopValve stopValve1(redeclare package Medium = Medium) annotation( Ⓚ);
  Modelica.Fluid.Sources.Boundary_pT boundary1( Ⓛ);
    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) Ⓛ);
    annotation( Ⓛ);
  inner Modelica.Fluid.System system annotation( Ⓛ);
  Modelica.Fluid.Vessels.OpenTank tank( Ⓛ);
    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( Ⓛ);
    redeclare package Medium = Medium, diameter = 0.015, height_ab = 2, length = 2) annotation( Ⓛ);
```

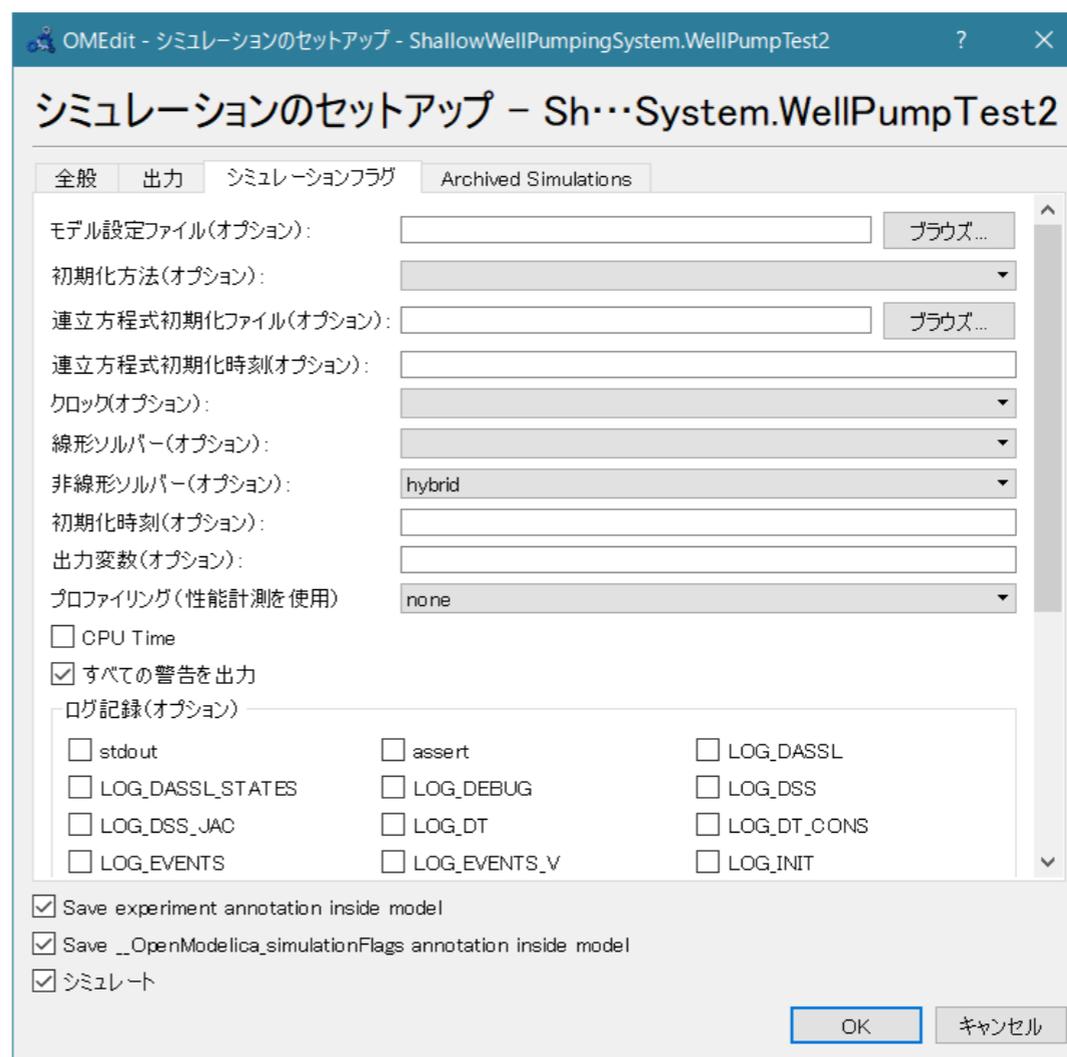
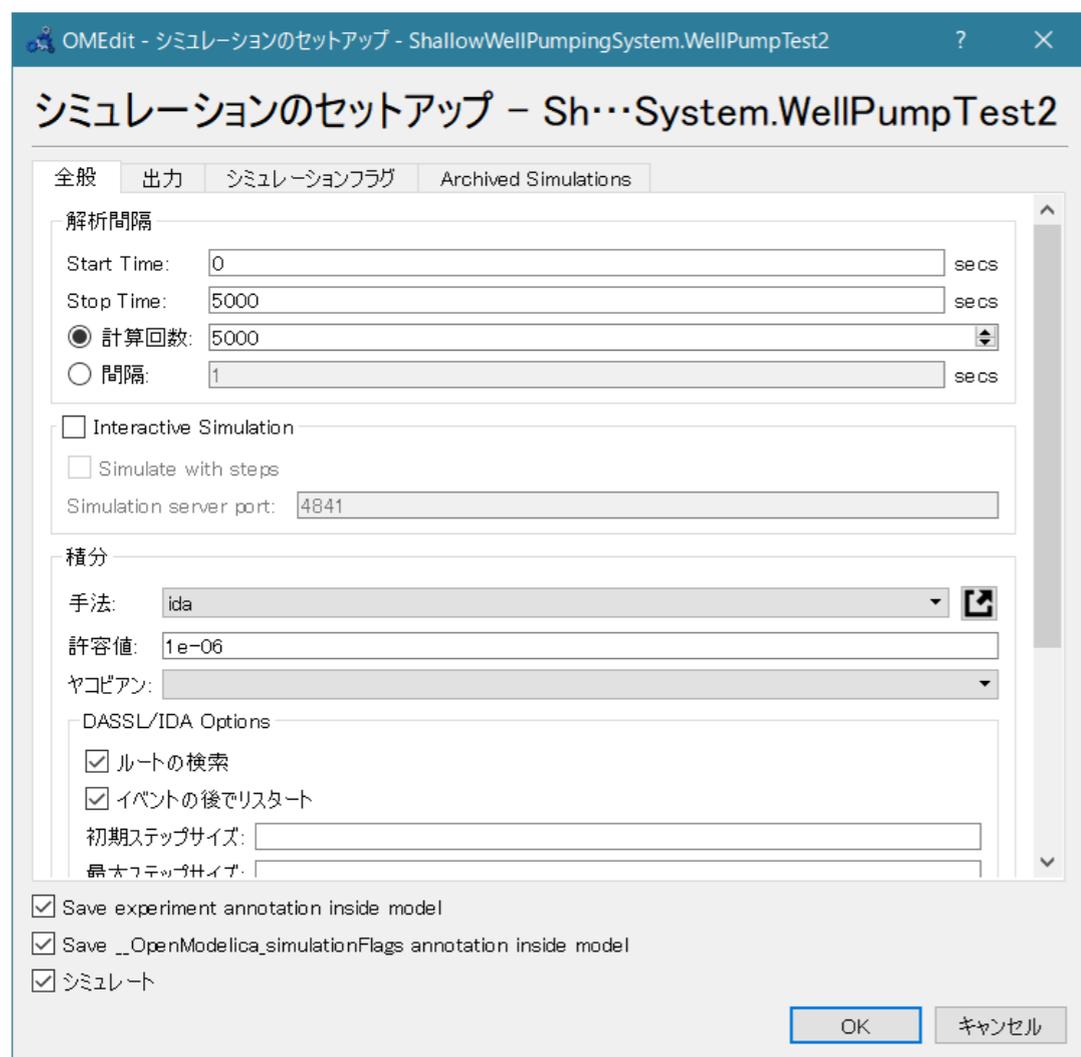
```
Modelica.Fluid.Vessels.ClosedVolume volume( ⑦
  redeclare package Medium = Medium, V = 0.001,
  energyDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial,
  massDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial,
  nPorts = 2, use_portsData = false) annotation( ...);
ShallowWellPumpingSystem.WellPump 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;
```

### ⑤ シミュレーション>モデルチェック

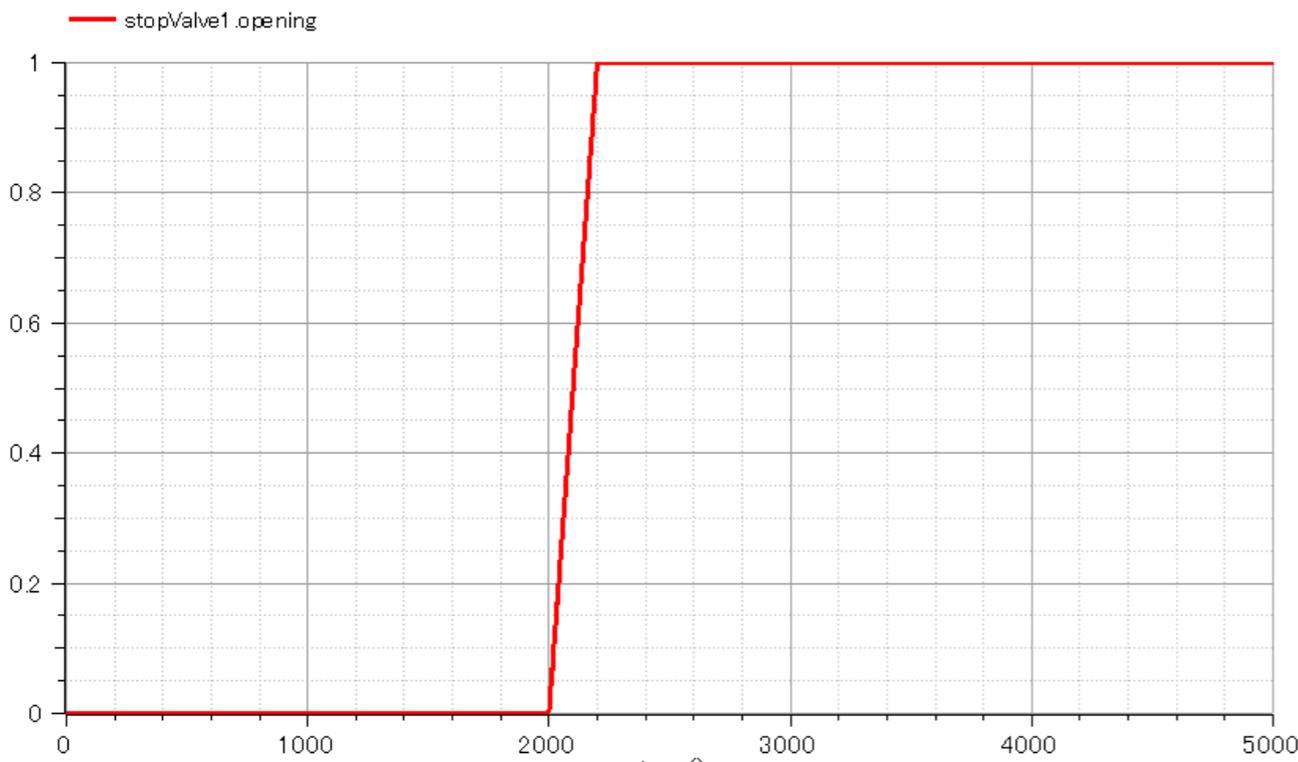


Start Time = 0 [s]  
 Stop Time = 5000 [s]  
 計算回数 = 5000  
 積分手法: ida  
 非線形ソルバー: hybrid

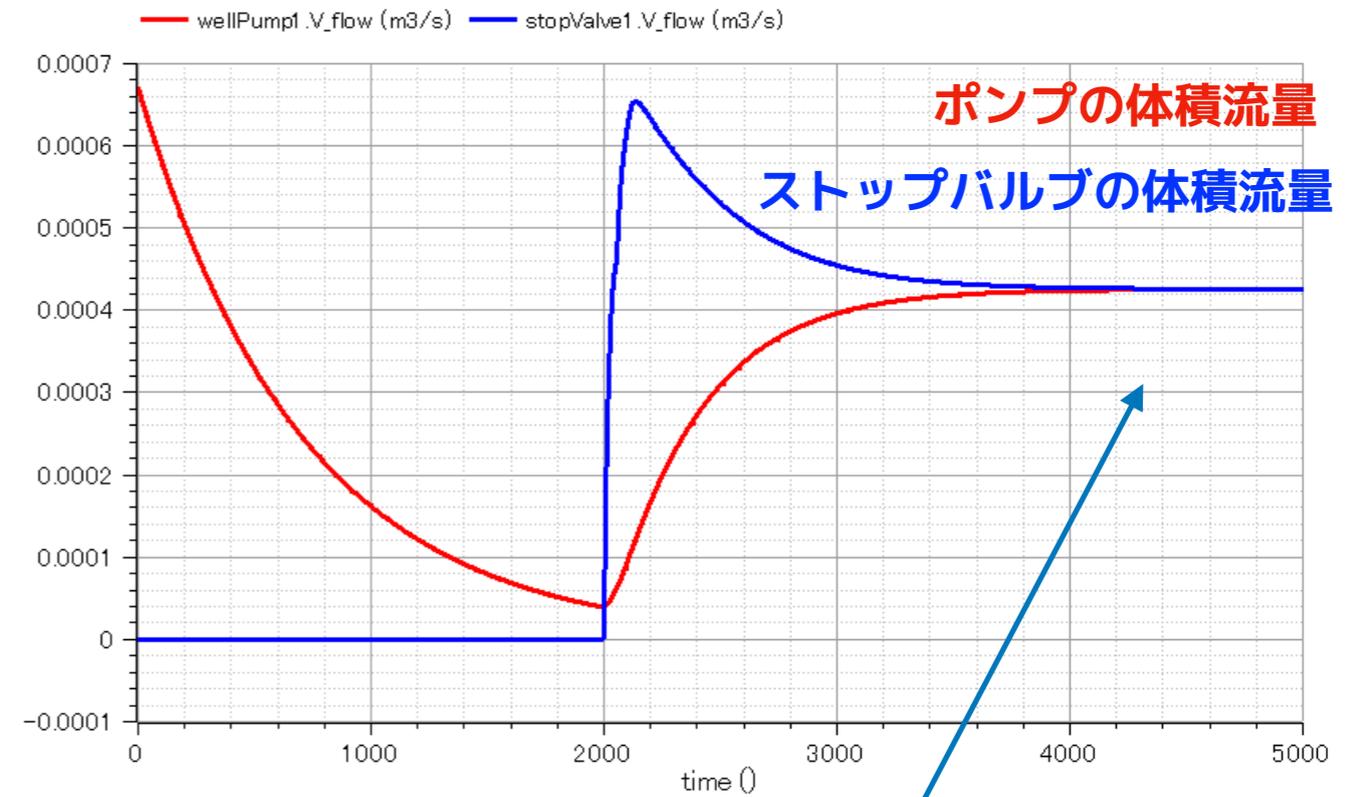
### ⑥ シミュレーション>シミュレーションのセットアップ



# シミュレーション結果

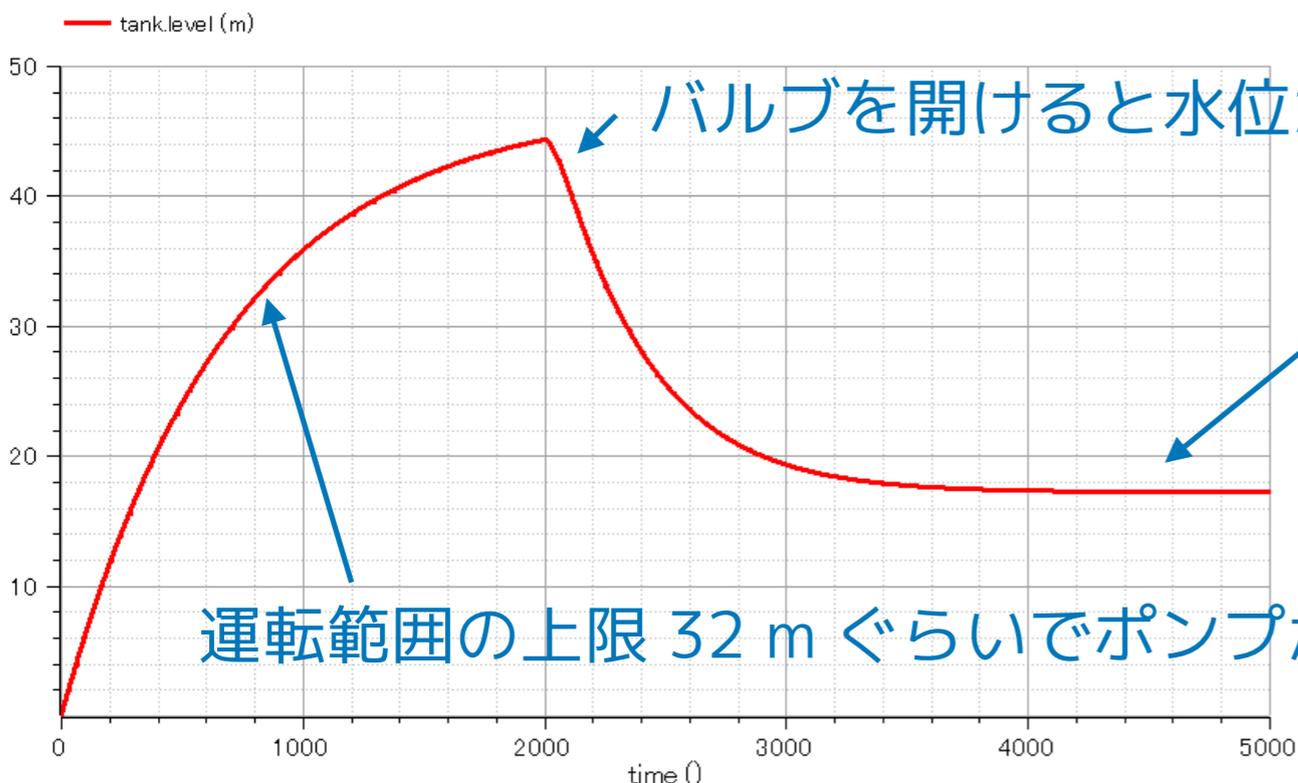


### ストップバルブの開度



ポンプの体積流量

ストップバルブの体積流量



### タンクの水位

バルブを開けると水位が下がり始める。

ポンプ流量のとバルブの流量が一致する  
水位で定常になる。

運転範囲の上限 32 m ぐらいでポンプが止まるような制御が必要！

## まとめ

- ストップバルブとポンプのモデルを作成して単体テストを行い、それぞれの性能を再現することができた。
- ポンプ、オープンタンク、ストップバルブを接続したテストモデルを作成した。ポンプの流量とストップバルブの流量が釣り合う水位で、このモデルが定常状態となることを確認した。

## 今後のメニュー

- StaticPie を使用して井戸内のパイプをモデル化する。
- OpenTank を改造して圧力タンクモデルを作る。
- Modelica.Blocks のコンポーネントで圧力スイッチをモデル化する。
- 以上を組み合わせて浅井戸ポンプ給水システムモデルを完成させる。

**Licensed by Amane Tanaka under the Modelica License 2**

**Copyright(c) 2019, Amane Tanaka**

**This document is free and the use is completely at your own risk; it can be redistributed and/or modified under the terms of the Modelica license 2, see the license conditions (including the disclaimer of warranty) at <http://www.modelica.org/licenses/ModelicaLicense2>**