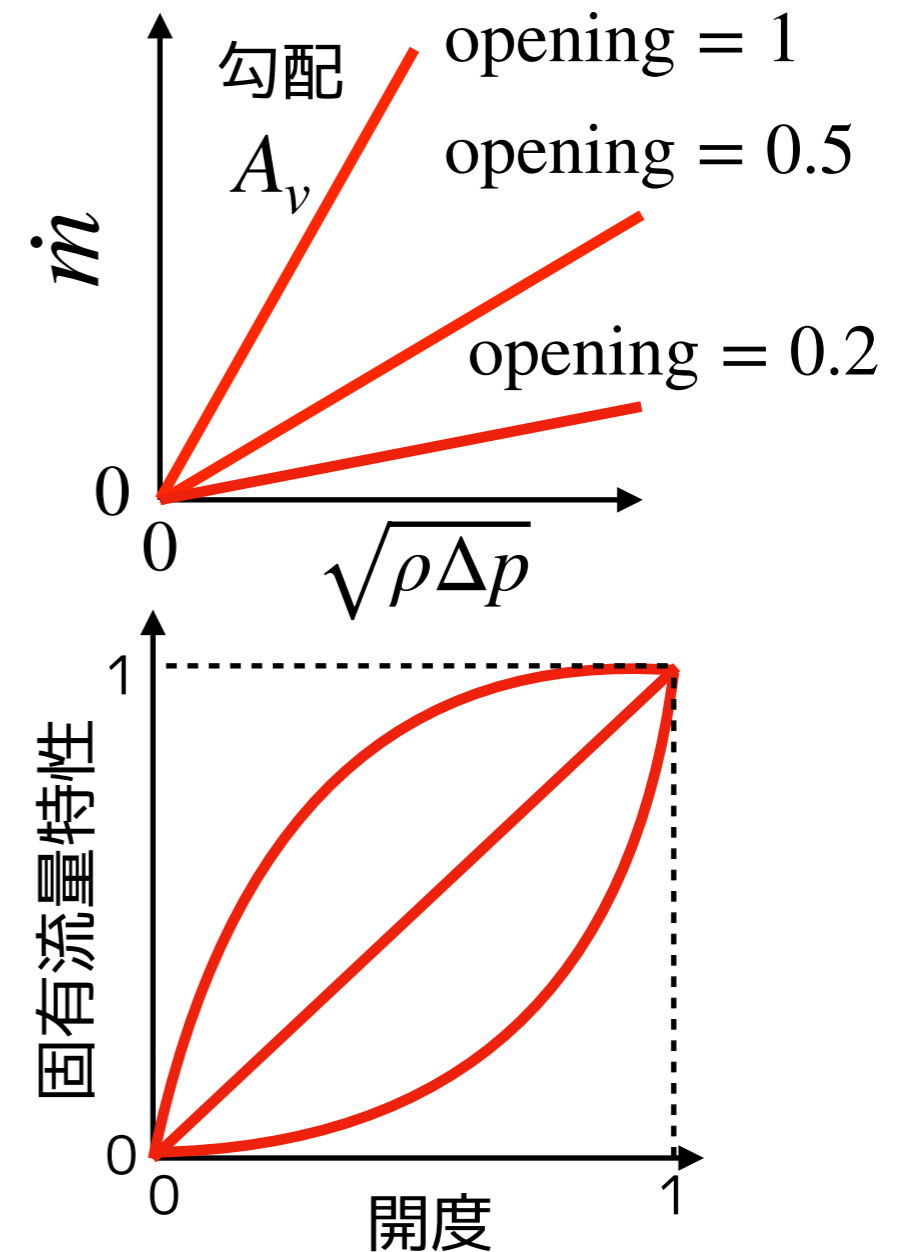
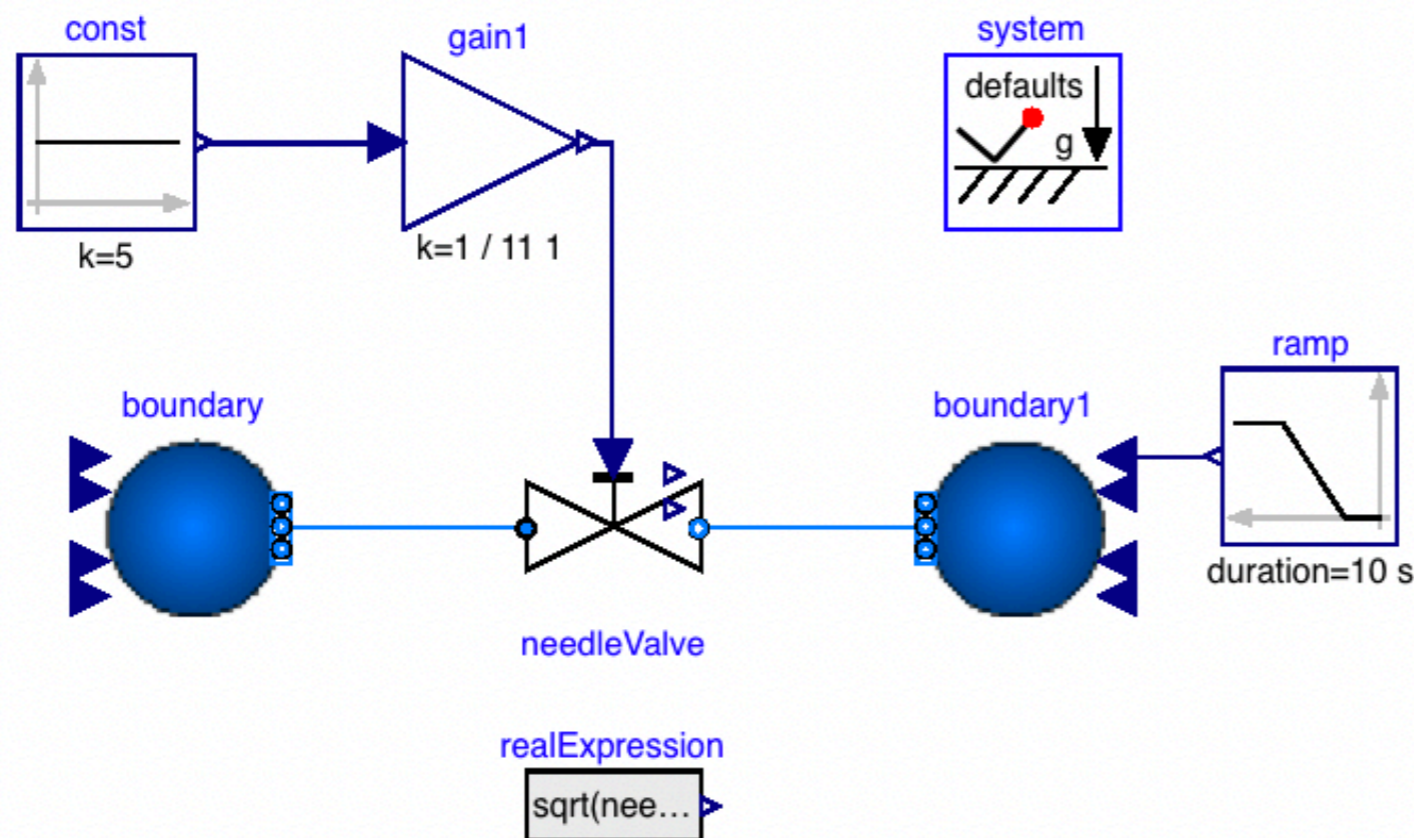


# Modelica.Fluid.Valvesについて(2)

液体用バルブ ValveIncompressible, ValveVaporizing



第14回 Modelica ライブラリ勉強会

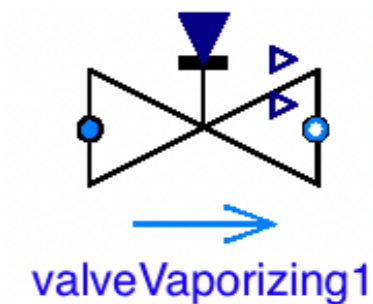
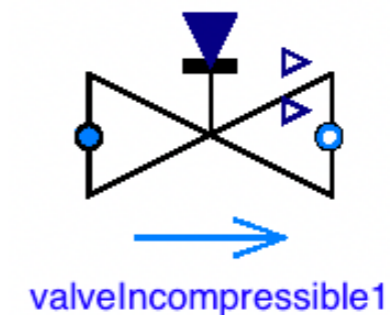
2019/09/28 finback

# 液体用バルブモデル

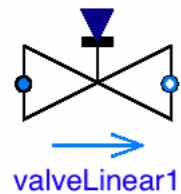
Modelica Standard Library (MSL) の Modelica.Fluid.Valves に含まれる **ValveIncompressible** と **ValveVaporizing** は液体用のバルブのモデルです。ValveIncompressible は、**流量係数** ( $A_v$ ,  $C_v$ ,  $K_v$  など) と **固有流量特性** を使用して圧力差と流量の関係を計算します。ValveVaporizing は、バルブ内部の液体の蒸発をモデル化します。さらに **圧力回復特性 (FL)** が必要となり、Medium に **2 相流体** を設定する必要があります。

## Contents

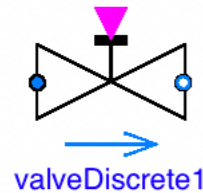
- [Modelica.Fluid.Valves のバルブモデル](#)
- [継承関係](#)
- [ValveIncompressible 液体用バルブ](#)
- [ValveVaporizing 内部で液体が蒸発するバルブ](#)
- [Regularization\(流れの適正化\)](#)
- [ホモトピー変換](#)
- [開度のフィルタリング](#)
- [例題 ValveExample2](#)
- [まとめ](#)



# Modelica.Fluid.Valves のバルブモデル

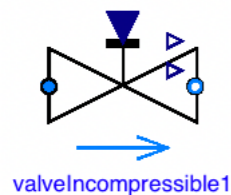


**ValveLinear**  
流量と開度・圧力差が比例する

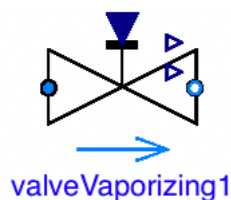


**ValveDiscrete**  
流量と圧力差が比例する  
全開と全閉のみ

簡易バルブモデル

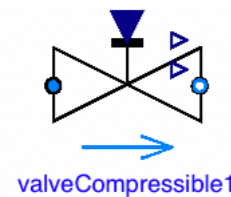


**ValveIncompressible**  
液体用バルブ



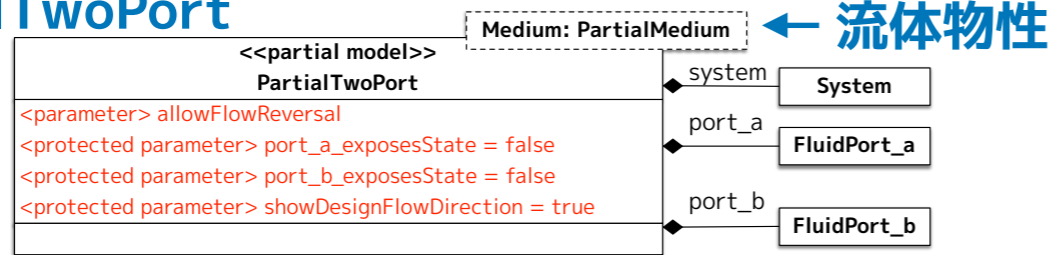
**ValveVaporizing**  
内部で液体が蒸発するバルブ

バルブ選定用の  
性能データを用いたモデル

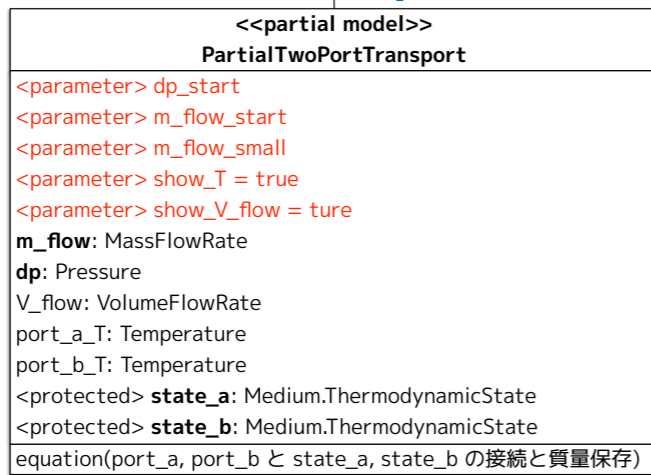


**ValveCompressible**  
気体用バルブ

## PartialTwoPort



## PartialTwoPortTransport

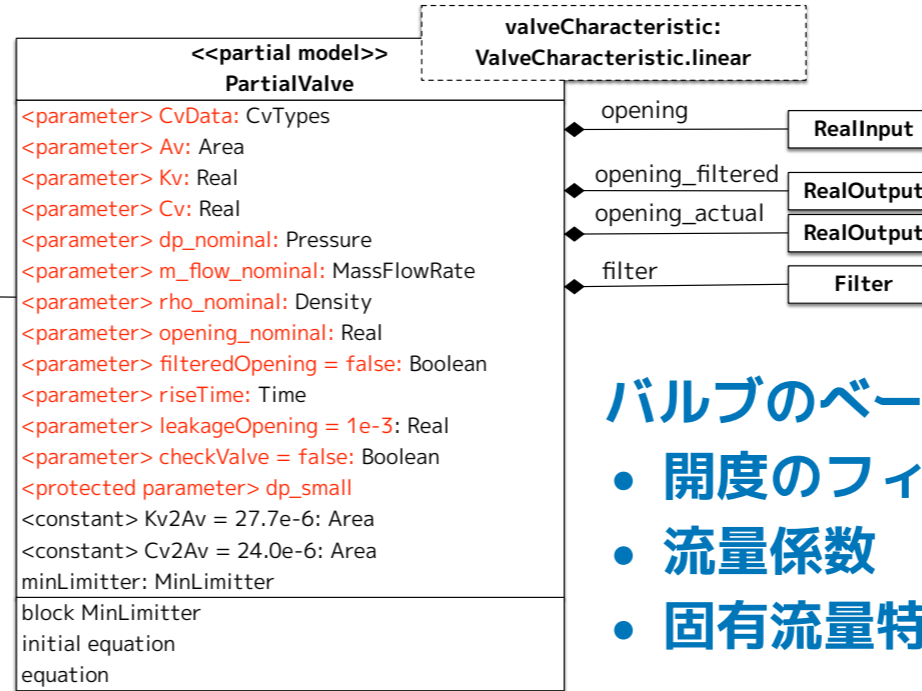


出入口があり

内部に流体を蓄えないモデル

# モデルの継承関係

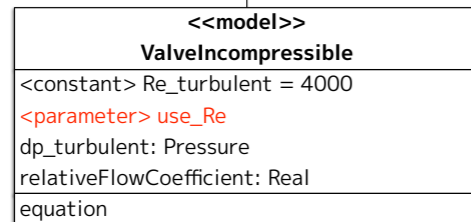
固有流量特性



バルブのベースモデル

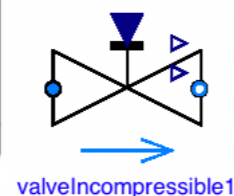
- 開度のフィルタ設定
- 流量係数
- 固有流量特性

圧力回復特性



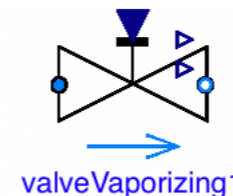
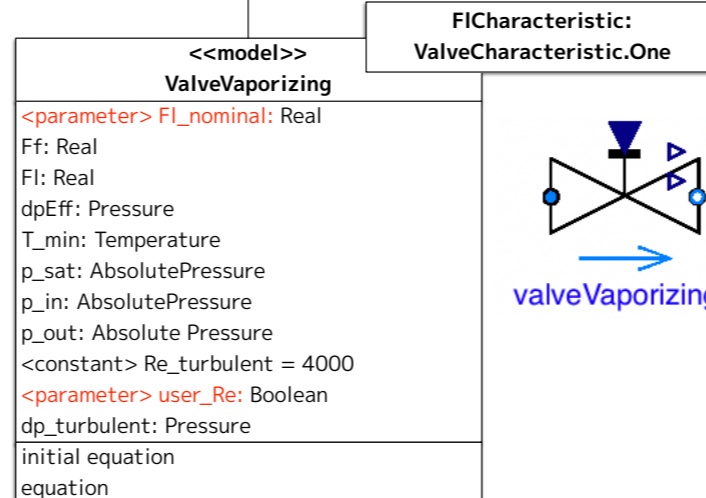
ValveIncompressible

- 液体用バルブ

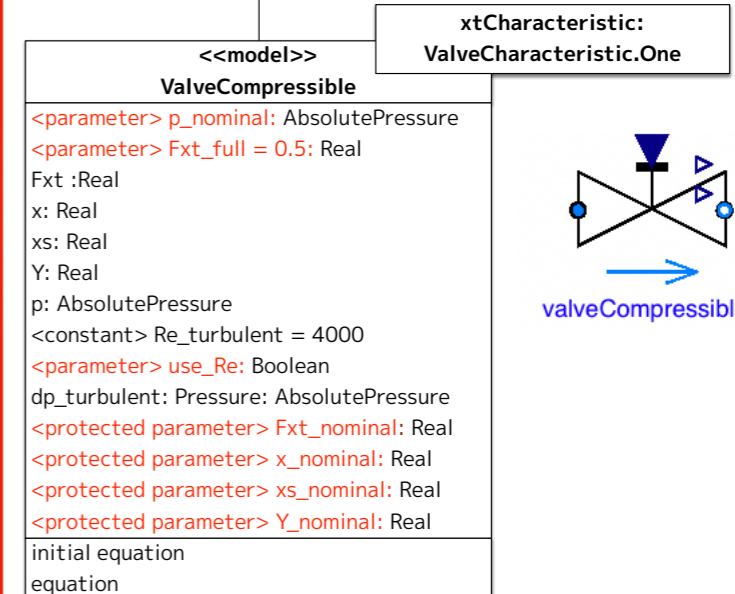


ValveVaporizing

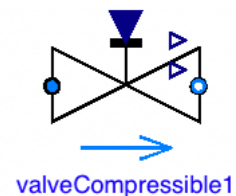
- 内部で液体が蒸発するバルブ



臨界圧力降下比特性

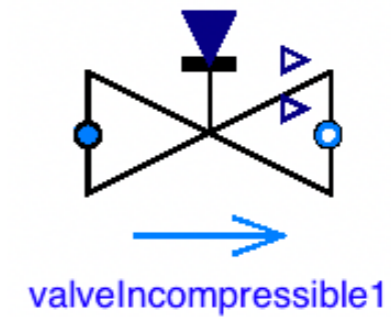


ValveCompressible 気体用バルブ



# ValveIncompressible 液体用バルブ

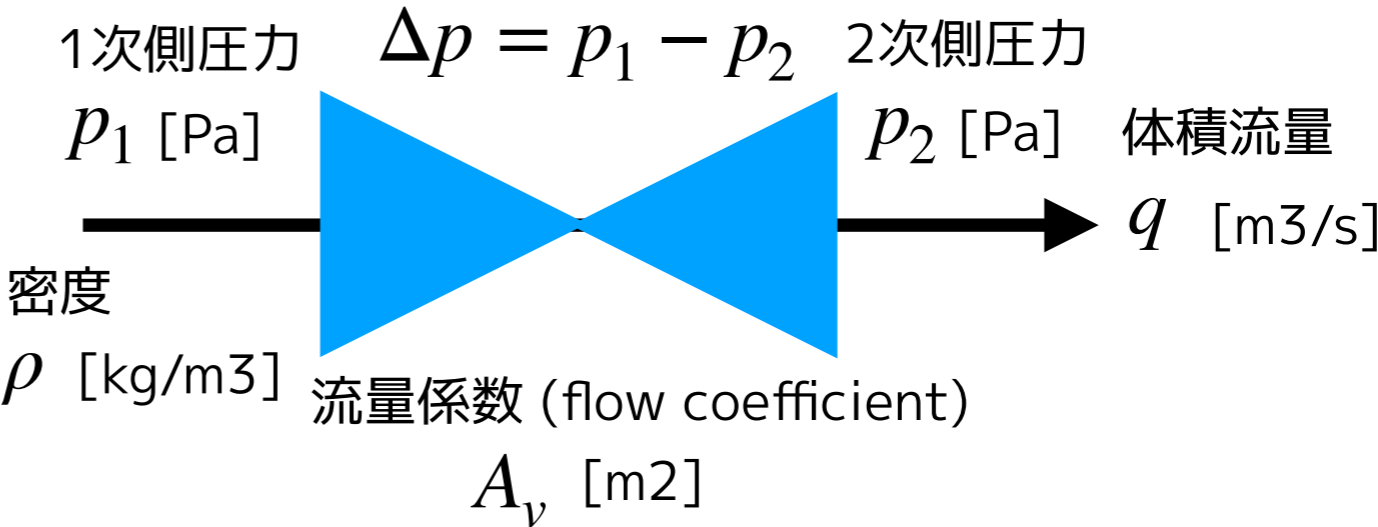
- 全開時の流量計算式
- 実用されている流量係数
- 流量係数の設定
- 開度を変えた場合の流量係数
- Modelica.Fluid.Valves の固有流量特性のモデル
- バルブ内部の流れ
- 流量係数、有効最縮流部断面積、圧力回復係数、圧力損失係数の関係



全開時の流量計算式

流量係数 (flow coefficient)

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



実用されている流量係数

	SI 単位	ヨーロッパ	アメリカ
流量計算式	$q = A_v \sqrt{\frac{\Delta p}{\rho}}$	$q = K_v \sqrt{\frac{\Delta p}{G}}$	$q = C_v \sqrt{\frac{\Delta p}{G}}$
全開時の流量係数 (flow coefficient)	$A_v$ [m <sup>2</sup> ]	$K_v = \frac{10^6}{27.7} A_v$	$C_v = \frac{10^6}{24} A_v$
体積流量 (volumetric flow rate)	$q$ [m <sup>3</sup> /s]	$q$ [m <sup>3</sup> /h],	$q$ [USG/min],
圧力降下 (pressure drop)	$\Delta p$ [Pa]	$\Delta p$ [bar]	$\Delta p$ [psi]
密度(density) 比重(specific gravity)	$\rho$ [kg/m <sup>3</sup> ]	$G = \rho / \rho_0$	$G = \rho / \rho_0$

$\rho$  をかけると

$$\dot{m} = A_v \sqrt{\rho \Delta p}$$

$A_v$  は面積の単位。

$K_v, C_v$  は、複雑な単位で、値は  $\rho_0$  のとり方に依存する。

$\rho_0 = 999.007$  [kg/m<sup>3</sup>]  
15.56°Cの水の密度 (60°F)

Fluid.UserGuide.ComponentDefinition.ValveCharacteristics には cold water at 4 °C とあるが間違い？

## 流量係数(flow coefficient)の設定

### ① パラメータ CvData で、流量係数の設定方法を選択する。

```
parameter Modelica.Fluid.Types.CvTypes CvData=Modelica.Fluid.Types.CvTypes.OpPoint
"Selection of flow coefficient"
```

#### 選択肢 (Modelica.Fluid.Types.CvTypes)

```
type CvTypes = enumeration(
  Av "Av (metric) flow coefficient",      Av を設定する
  Kv "Kv (metric) flow coefficient",      Kv を設定する
  Cv "Cv (US) flow coefficient",          Cv を設定する
  OpPoint "Av defined by operating point") 流量などの基準値から計算する(デフォルト)
"Enumeration to define the choice of valve flow coefficient" annotation (···);
</html>));
```

### ② CvData = OpPoint (デフォルト) を選択した場合、m\_flow\_nominal, rhod\_nominal, dp\_nominal, opening\_nominal などから Av が計算される。

### ③ CvData = Av を 選択した場合、パラメータ Av を上書きして設定する。

```
parameter SI.Area Av(
  fixed= CvData == Modelica.Fluid.Types.CvTypes.Av,
  start=m_flow_nominal/(sqrt(rho_nominal*dp_nominal))*valveCharacteristic(opening_nominal))
  "Av (metric) flow coefficient"
```

デフォルト CvData = OpPoint の場合

$$A_v = \frac{m\_flow\_nominal}{\sqrt{\rho\_nominal \cdot \Delta p\_nominal}} valveCharacteristic(opening\_nominal)$$

④ CvData = Kv または CvData = Cv を 選択した場合、パラメータ Kv またはパラメータ Cv を設定する。Av は Cv または Kv から自動的に計算される。

```
initial equation
  if CvData == CvTypes.Kv then
    Av = Kv*Kv2Av "Unit conversion";
  elseif CvData == CvTypes.Cv then
    Av = Cv*Cv2Av "Unit conversion";
  end if;
```

← CvData = Kv, Cv のときは  $A_v$  に変換する。

```
public
  constant SI.Area Kv2Av = 27.7e-6 "Conversion factor";
  constant SI.Area Cv2Av = 24.0e-6 "Conversion factor";
```

## 開度を変えた場合の流量計算式

流量係数は開度とともに変化する。

固有流量特性, 相対流量係数  
(relative flow coefficient)

全開時の流量係数  
(flow coefficient)

$$\dot{m} = \rho q = (\text{relativeFlowCoefficient} \cdot A_v) \sqrt{\rho \Delta p}$$

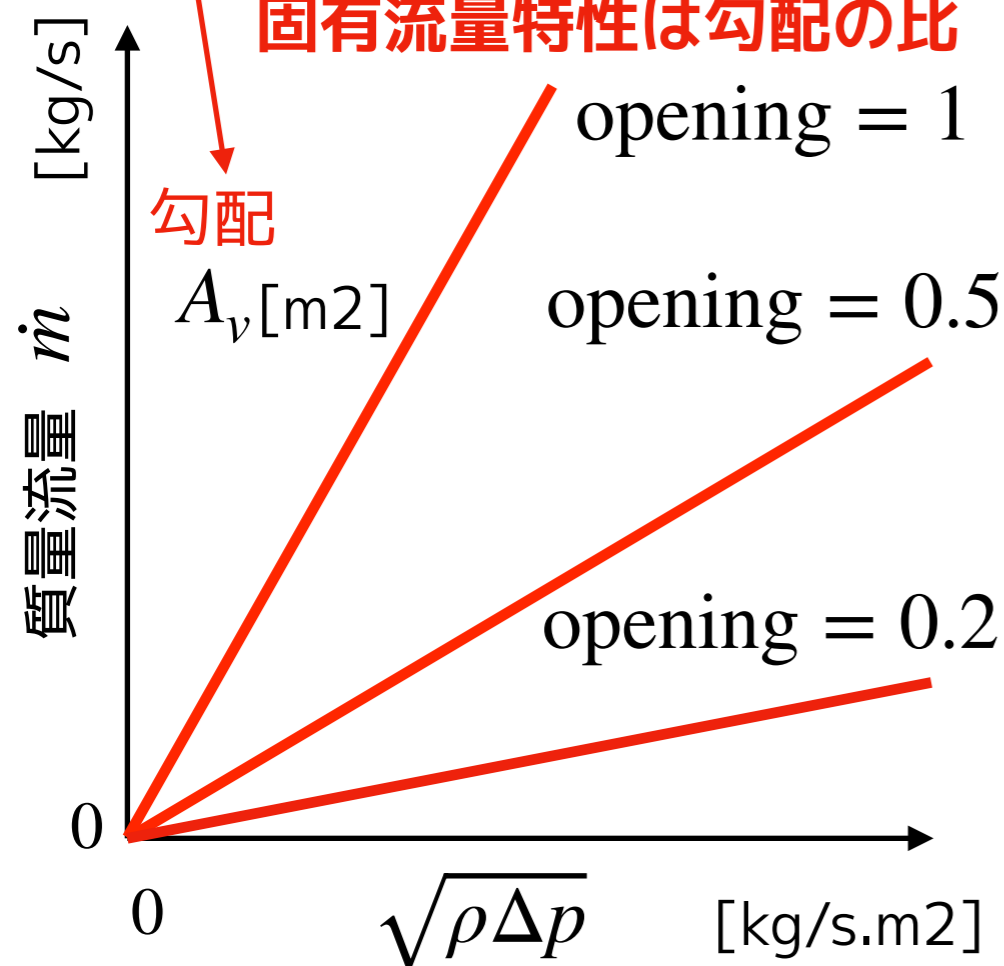
$$\text{relativeFlowCoefficient} = \text{valveCharacteristic}(\text{opening})$$

全開時に対する流量係数の比

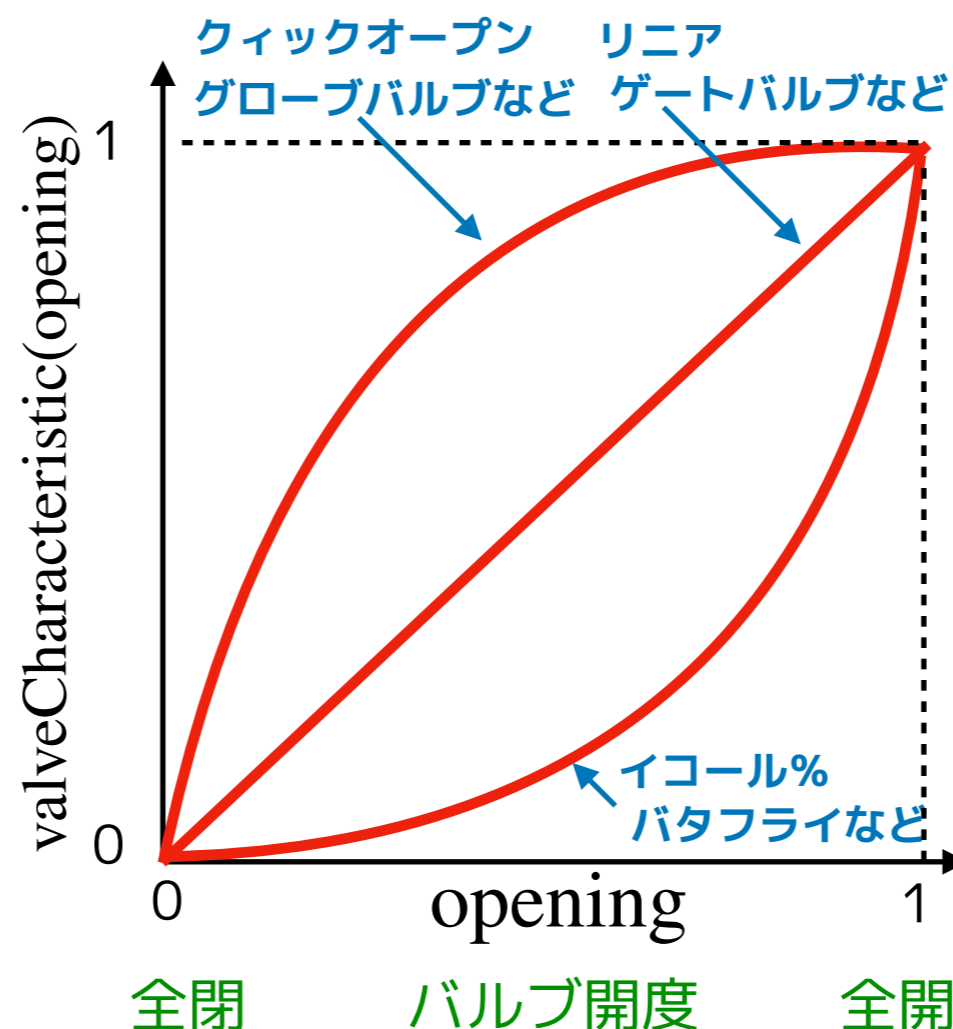
バルブ開度

流量係数は勾配

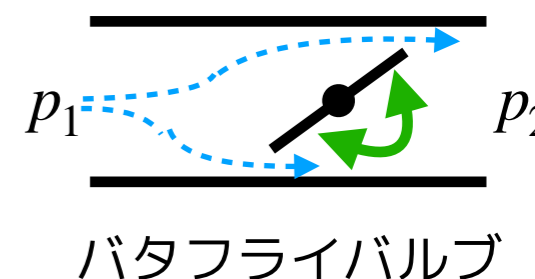
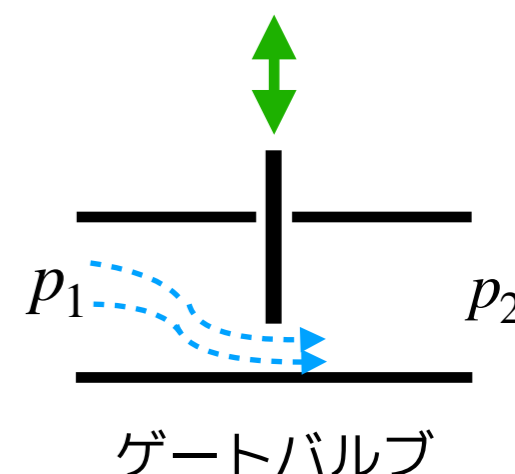
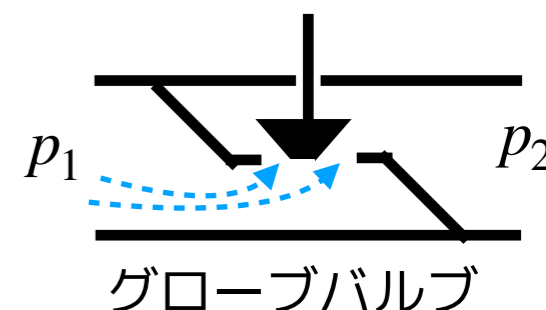
固有流量特性は勾配の比



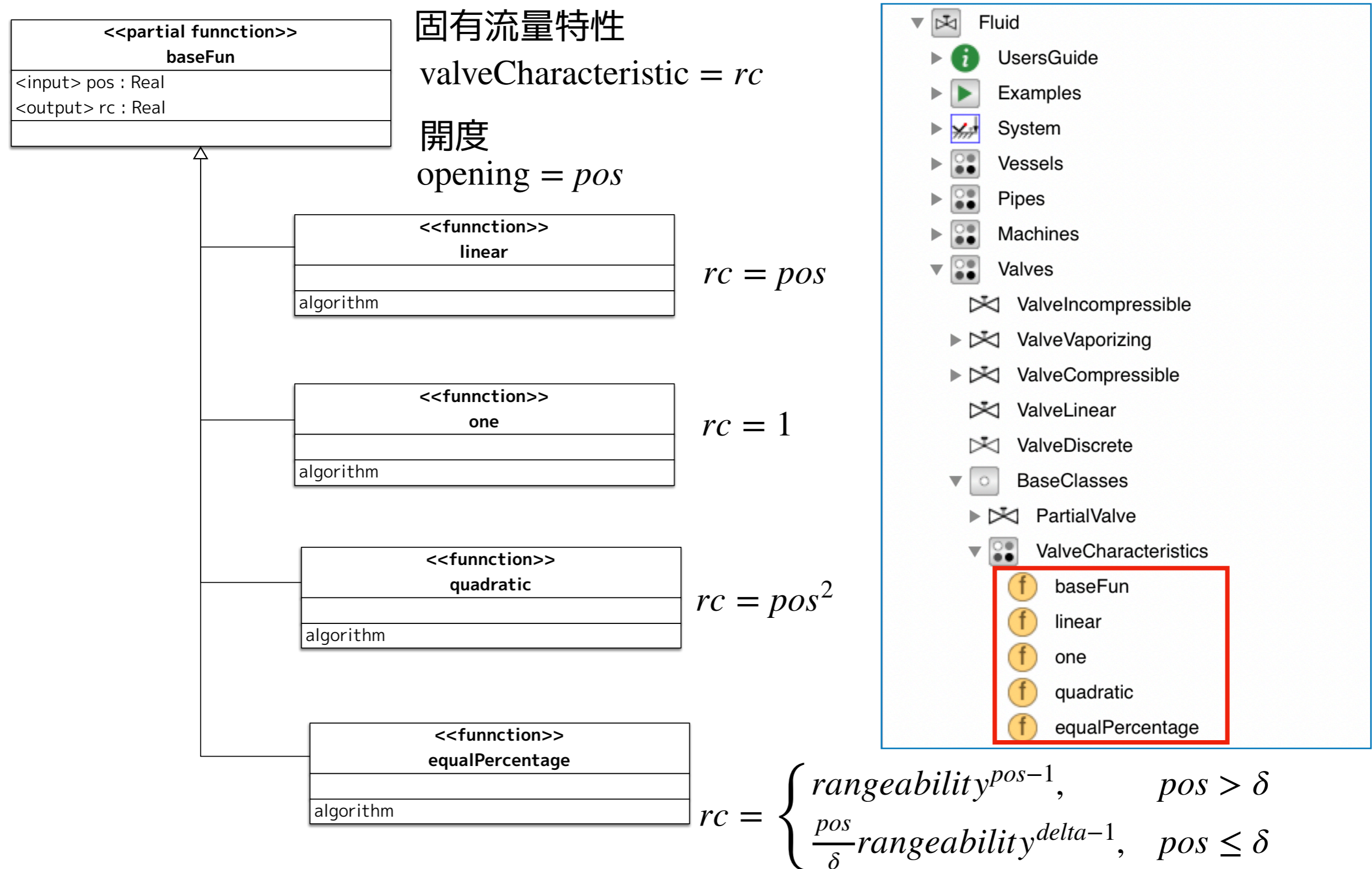
固有流量特性



バルブ開度  
opening



# Modelica.Fluid.Valves で用意されてる固有流量特性のモデル



## 固有流量特性 (relative flow coefficient) の設定

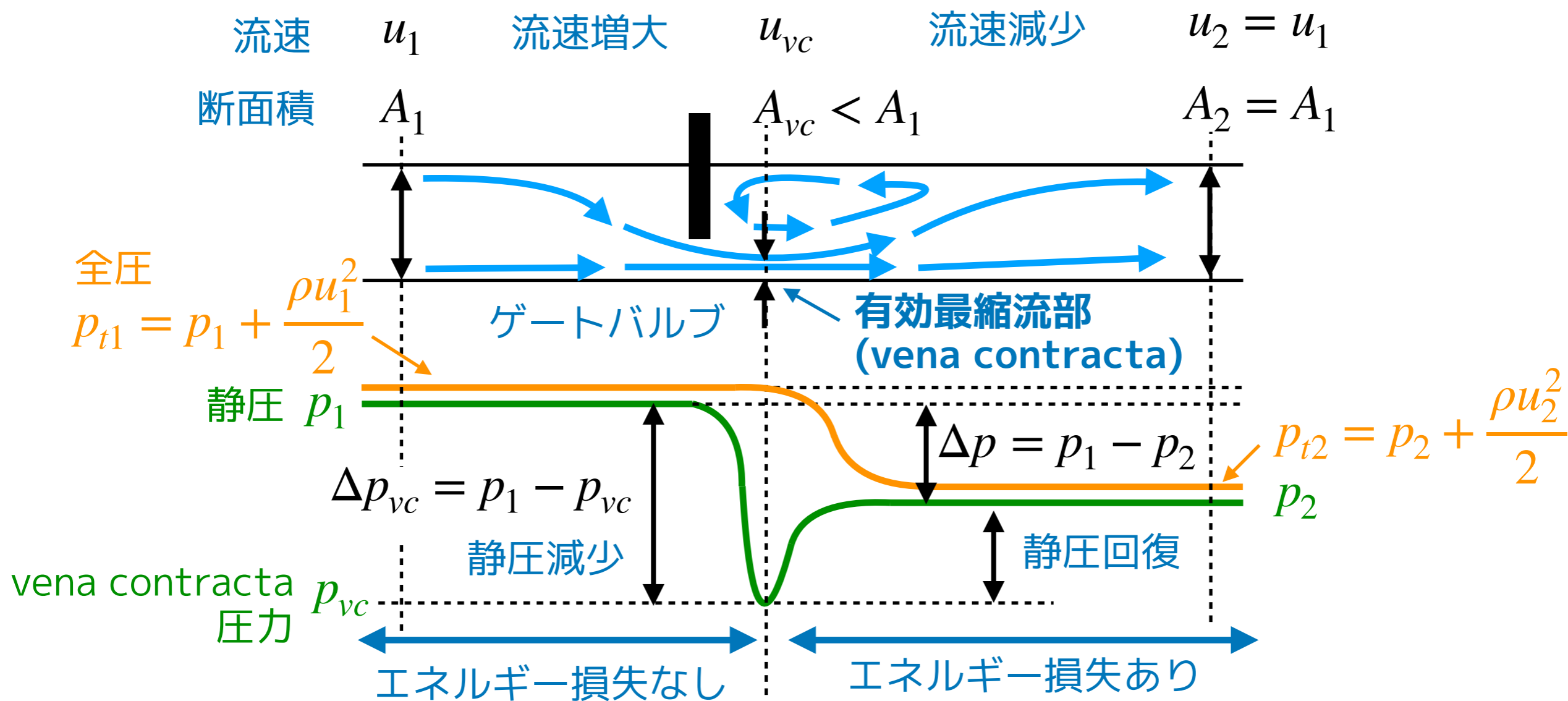
### PartialValve

```
replaceable function valveCharacteristic =  
    Modelica.Fluid.Valves.BaseClasses.ValveCharacteristics.linear  
constrainedby  
    Modelica.Fluid.Valves.BaseClasses.ValveCharacteristics.baseFun  
    "Inherent flow characteristic"  
annotation(choicesAllMatching=true);
```

この function を redeclare することによって  
固有流量特性を定義する。

## バルブ内部の流れ

(1) 流量保存式 (flow rate valance)  $\dot{m} = \rho A_1 u_1 = \rho A_{vc} u_{vc} = \rho A_2 u_2$



(2) ベルヌーイの定理 (Bernoulli's principle)

$$p_{t1} = p_1 + \frac{\rho u_1^2}{2} = p_{vc} + \frac{\rho u_{vc}^2}{2}$$

(3) 圧力回復係数 (pressure recovery factor)

$$F_L \equiv \sqrt{\frac{p_1 - p_2}{p_1 - p_{vc}}} = \sqrt{\frac{\Delta p}{\Delta p_{vc}}}$$

## 流量係数、有効最縮流部断面積、圧力回復係数、圧力損失係数の関係

$$\begin{aligned}
 \Delta p &= \overset{\text{(3) 圧力回復係数}}{\downarrow} F_L^2 \Delta p_{vc} = F_L^2 (p_1 - p_{vc}) = \overset{\text{(2) ベルヌーイの定理}}{\downarrow} F_L^2 \left( \frac{\rho u_{vc}^2}{2} - \frac{\rho u_1^2}{2} \right) = \overset{\text{(1) 流量保存式}}{\downarrow} F_L^2 \frac{\rho u_1^2}{2} \frac{A_1^2}{A_{vc}^2} \left( 1 - \frac{A_{vc}^2}{A_1^2} \right) \\
 \therefore q = u_1 A_1 &= \sqrt{\frac{2}{1 - \frac{A_{vc}^2}{A_1^2}} \frac{A_{vc}}{F_L}} \sqrt{\frac{\Delta p}{\rho}} = A_v \sqrt{\frac{\Delta p}{\rho}} \Leftrightarrow \frac{\rho u_1^2}{2} = \frac{1}{2} \frac{A_v^2}{A_1^2} \Delta p
 \end{aligned}$$

### 流量係数

vena contracta 断面積

$$A_v = \sqrt{\frac{2}{1 - \frac{A_{vc}^2}{A_1^2}} \frac{A_{vc}}{F_L}} \sim \frac{\sqrt{2} A_{vc}}{F_L} \text{ [m}^2\text{]}$$

圧力回復係数

### 圧力損失係数

入口断面積

$$K_i \equiv \frac{p_{t1} - p_{t2}}{\frac{\rho u_1^2}{2}} = \frac{\Delta p}{\frac{1}{2} \frac{A_v^2}{A_1^2} \Delta p} = \frac{2 A_1^2}{A_v^2}$$

バルブ前後の全圧差

動圧

流量係数

- 流量係数は、縮流と圧力回復のバランスを表す。
- 流量係数は、**圧力損失係数**と**基準断面積**からも計算できる。

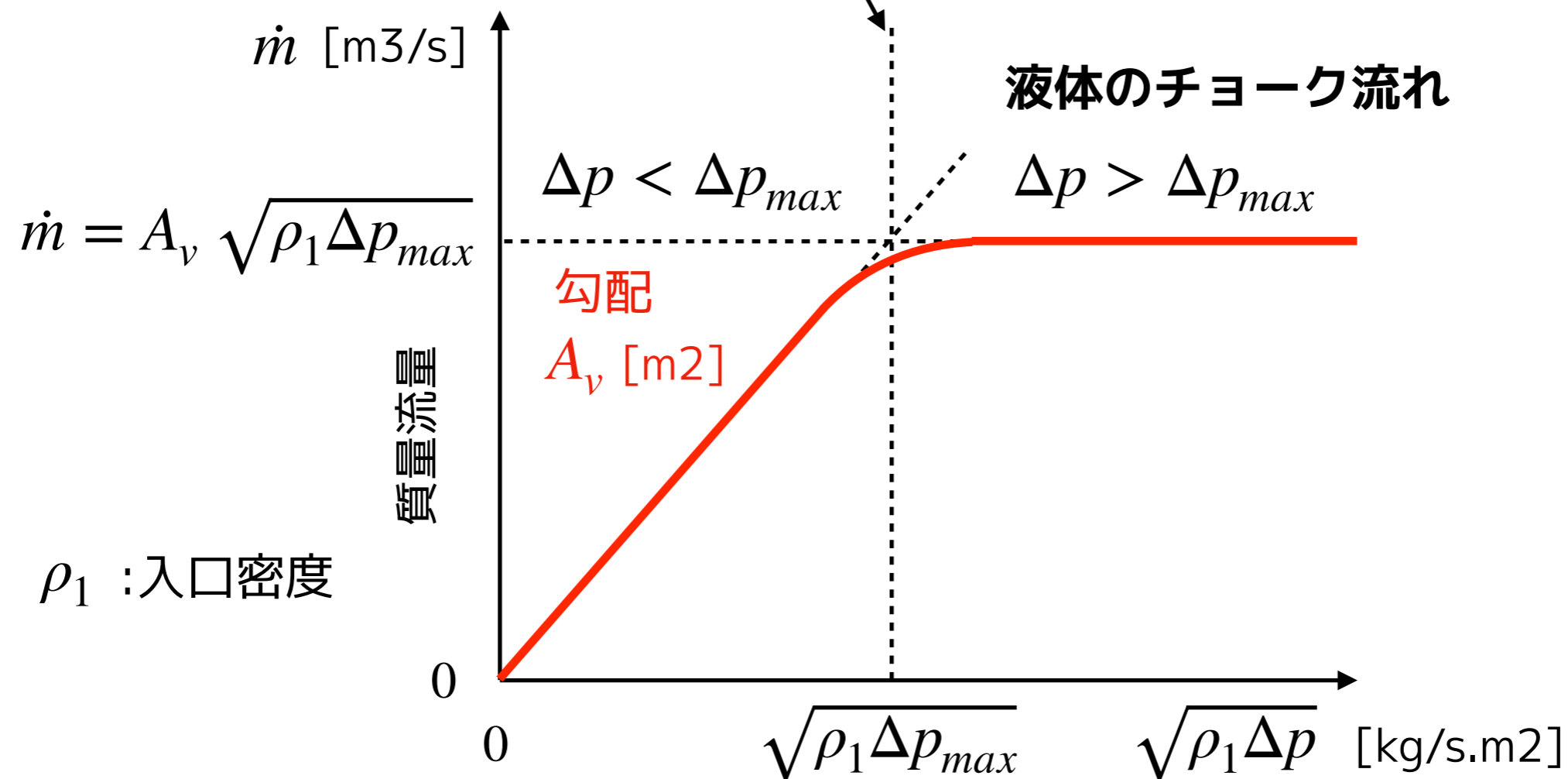
# ValveVaporizing 内部で液体が蒸発するバルブ

- 液体のチョーク流れ
- 設計上の最大許容圧力効果と圧力回復係数
- 開度を変えた場合の流量計算式
- 圧力回復係数の求め方

## 液体のチョーク流れ (liquid choked flow)

$\Delta p$  が  $\Delta p_{max}$  に達すると vena contracta で気泡が発生して流量が増えなくなる。

$\Delta p = \Delta p_{max}$  : 設計上の最大許容圧力降下  
(maximum allowable sizing pressure drop)



## 設計上の最大許容圧力降下と圧力回復係数

設計上の最大許容圧力降下  
(maximum allowable sizing pressure drop)

$$\Delta p_{max} = F_L^2(p_1 - F_F p_{sat1})$$

vena contracta の圧力が  $p_{vc} = F_F p_{sat1}$   
となるときの差圧  $\Delta p$

圧力回復係数

$$F_L \equiv \sqrt{\frac{p_1 - p_2}{p_1 - p_{vc}}} = \sqrt{\frac{\Delta p}{\Delta p_{vc}}}$$

臨界圧力比係数

(liquid critical pressure ratio factor)

$$F_F = 0.96 - 0.28 \sqrt{\frac{p_{sat1}}{p_{crit}}}$$

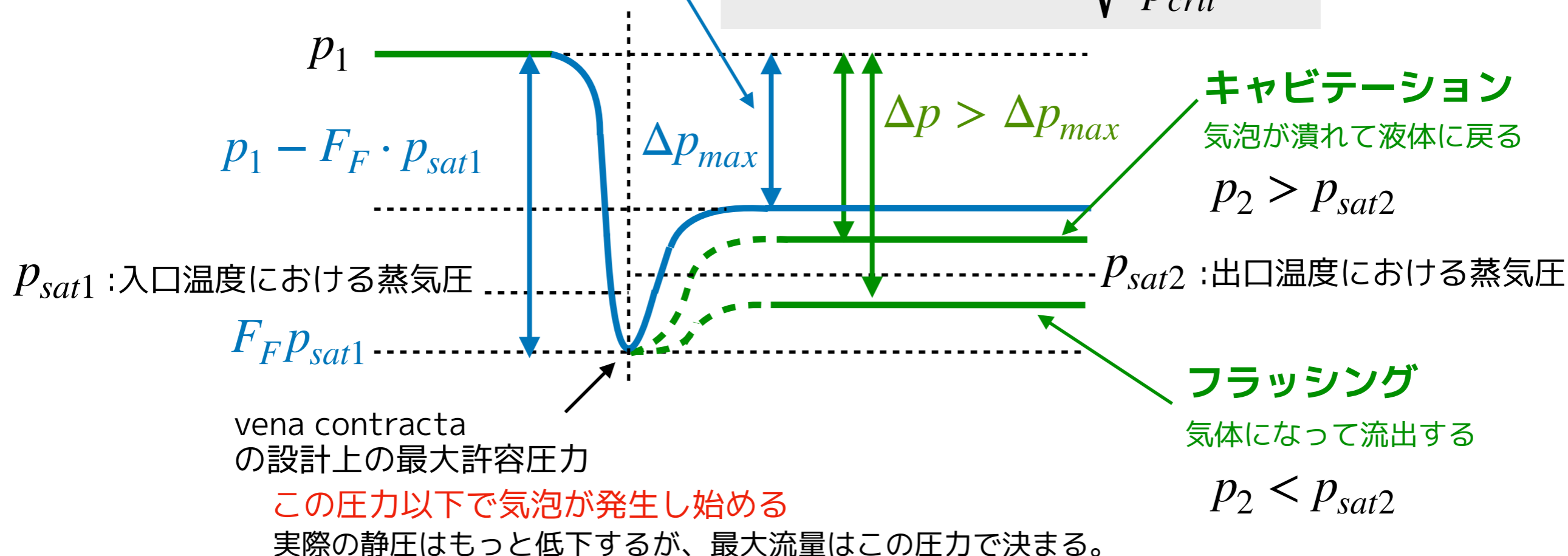
$p_1$  入口圧力

$p_2$  出口圧力

$p_{vc}$  vena contract  
圧力

$$\Delta p = p_1 - p_2$$

$$\Delta p_{vc} = p_1 - p_{vc}$$



## 開度を変えた場合の流量計算式

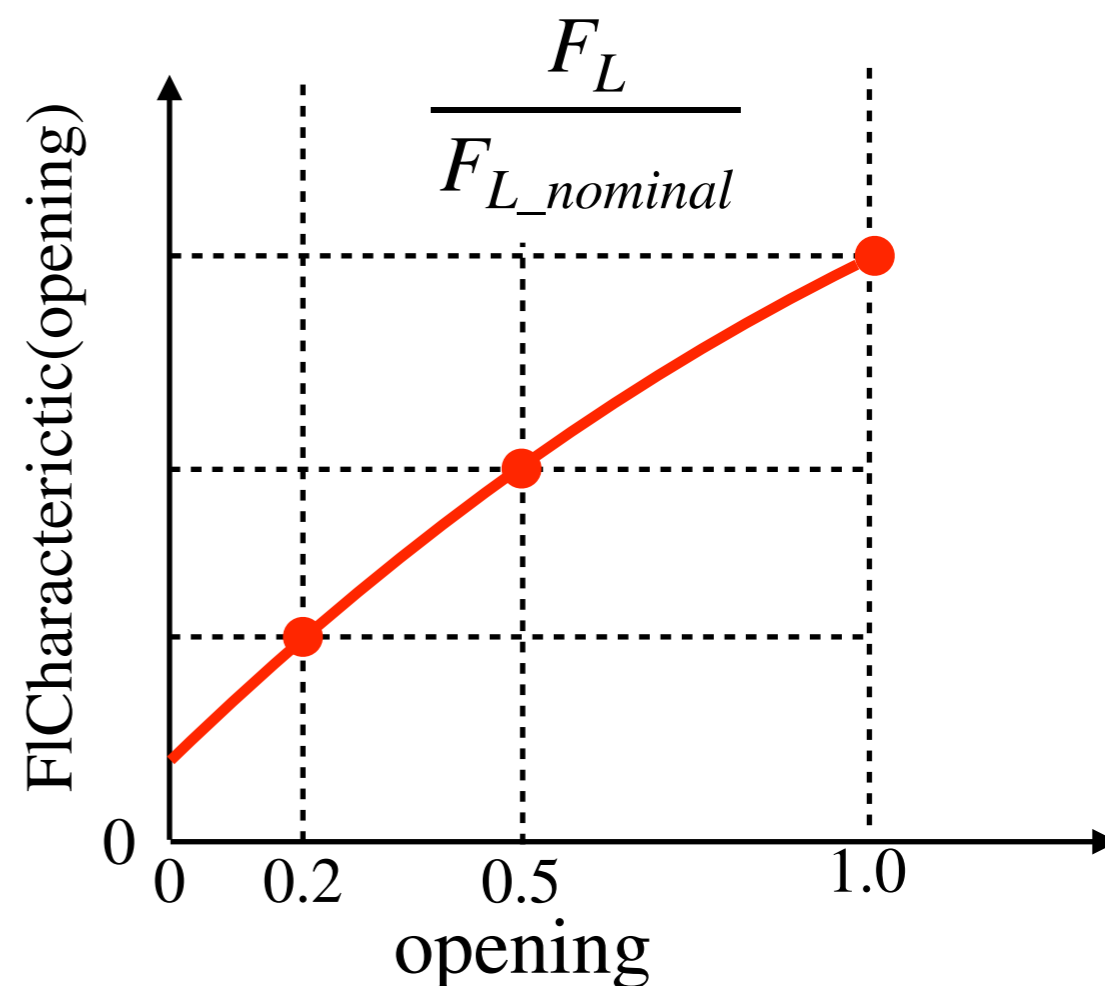
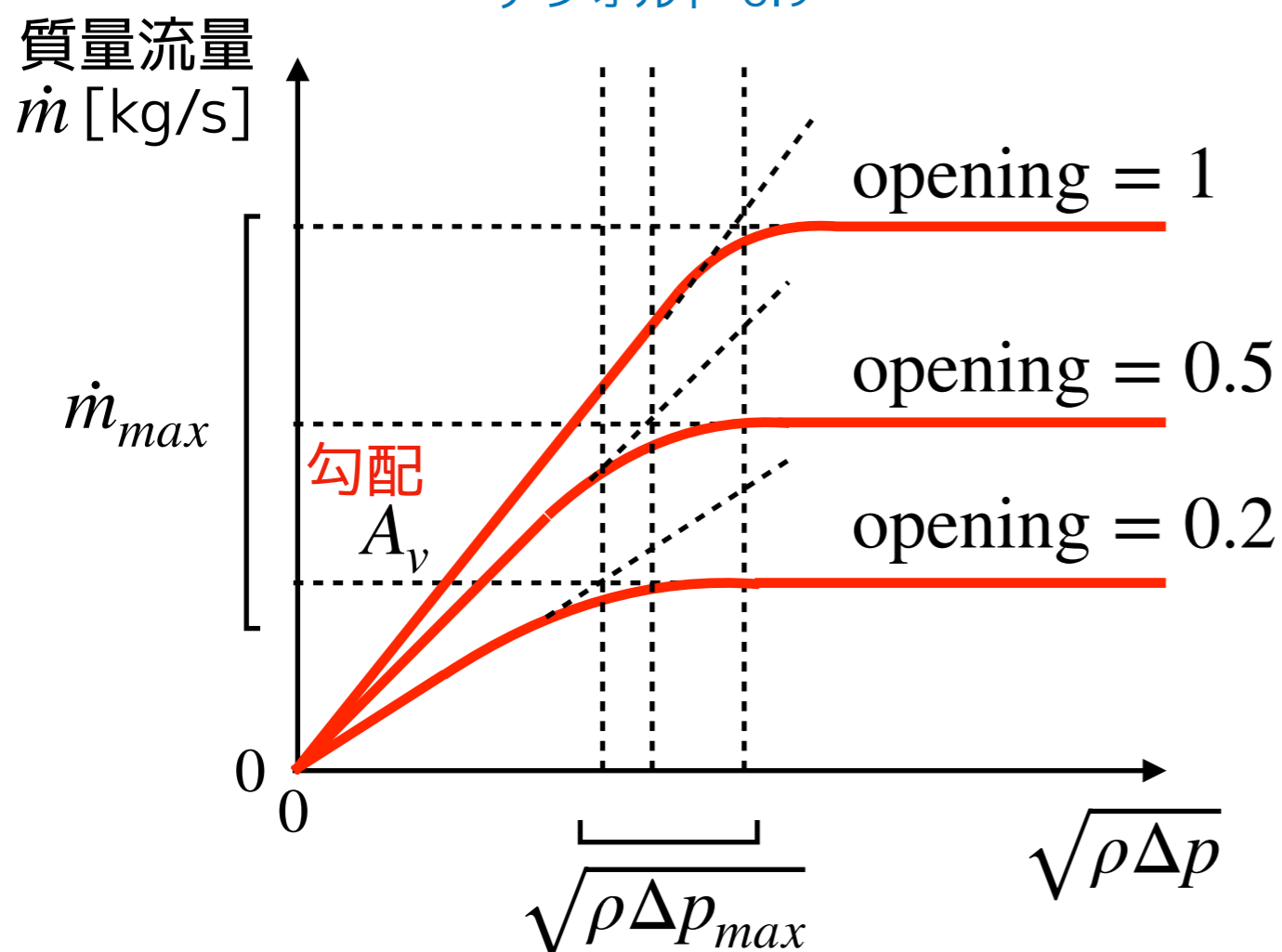
$$\dot{m} = (\text{valveCharacteristic}(\text{opening}) \cdot A_v) \sqrt{\rho \Delta p_{eff}}$$

$$\Delta p_{eff} = \begin{cases} \Delta p, & \Delta p \leq \Delta p_{max} \\ \Delta p_{max}, & \Delta p > \Delta p_{max} \end{cases} \quad \text{実効的圧力降下 (effective pressure drop)}$$

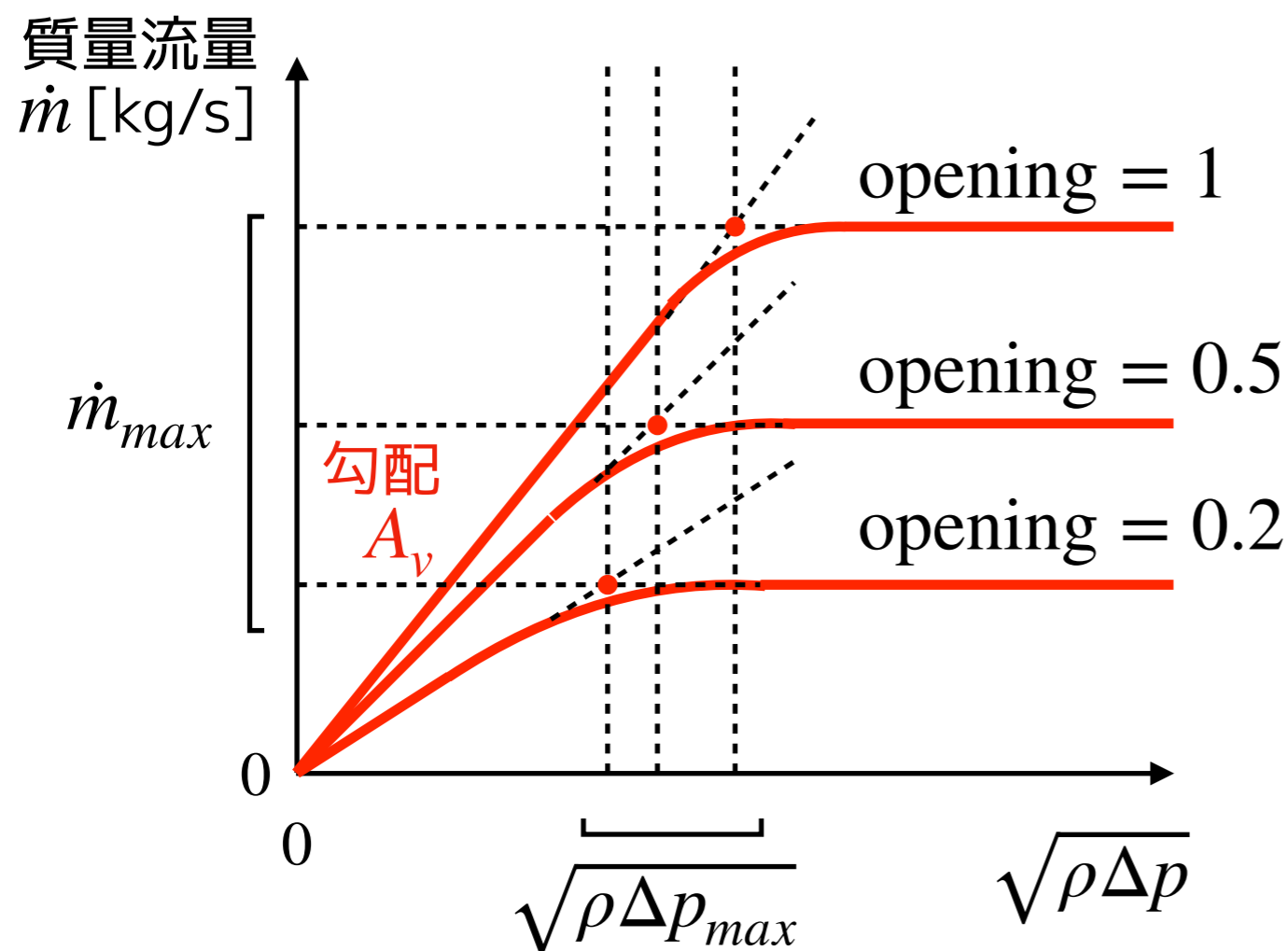
$$\Delta p_{max} = F_L^2 (p_1 - F_F \cdot p_{sat1}) \quad \text{設計上の最大許容圧力降下 (maximum allowable sizing pressure drop)}$$

$$F_L = F_{L\_nominal} \cdot \text{FlCharacteristic}(\text{opening}) \quad \text{圧力回復係数 (pressure recovery coefficient)}$$

デフォルト 0.9



## 圧力回復係数の求め方



$$F_L \equiv \sqrt{\frac{p_1 - p_2}{p_1 - p_{vc}}} = \sqrt{\frac{\Delta p}{\Delta p_{vc}}}$$

$p_1$  [Pa]: 入口圧力

$p_2$  [Pa]: 出口圧力

$\rho$  : 入口密度

$p_{sat1}$  : 入口温度における蒸気圧

臨界圧力比係数

liquid critical pressure ratio factor

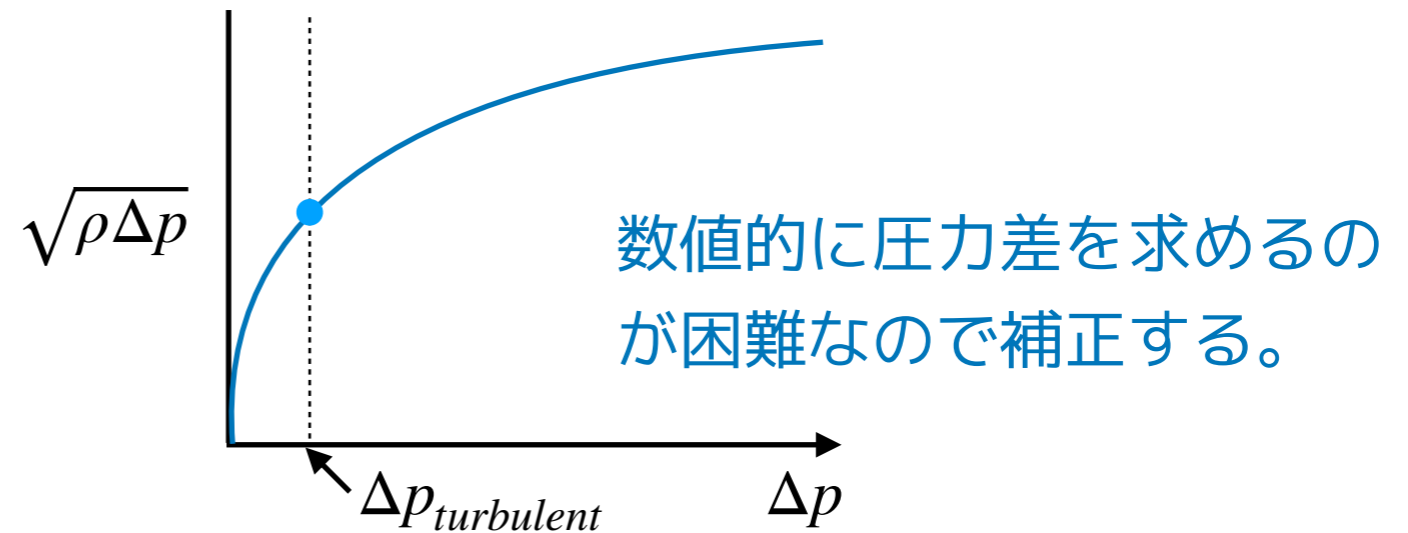
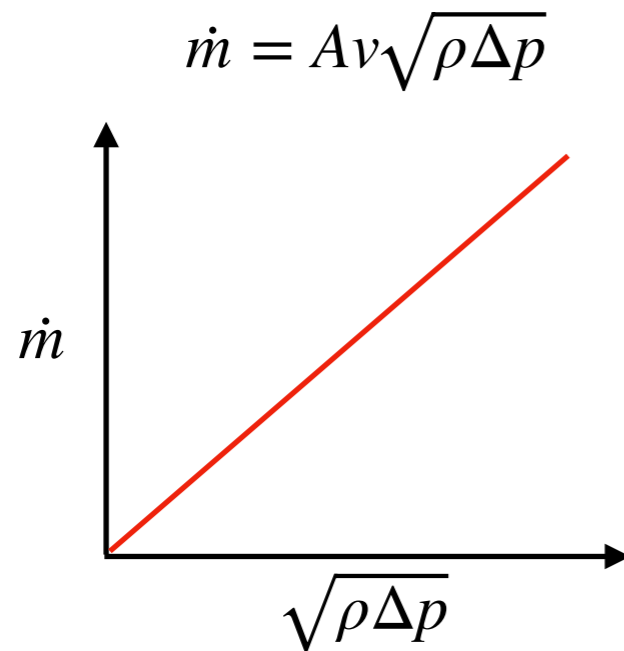
$$F_F = 0.96 - 0.28 \sqrt{\frac{p_{sat1}}{p_c}}$$

- $p_1$  を固定して  $\Delta p = p_1 - p_2$  を変化させる実験を行い、開度毎に  $\Delta p_{max}$  または  $\dot{m}_{max}$  を測定する。
- 以下の式より  $F_L$  を計算する。

$$\dot{m}_{max} = (\text{valveCharacteristic}(\text{opening}) \cdot A_v) \sqrt{\rho(p_1 - F_F p_{sat1})} \cdot F_L$$

$$\sqrt{\rho \Delta p_{max}} = \sqrt{\rho(p_1 - F_F p_{sat1})} \cdot F_L$$

# Regularization 流れの適正化



$|\Delta p| < \Delta p_{turbulent}$  のとき

- $\sqrt{\rho\Delta p}$  の勾配が発散してしまう。
- 流れは乱流から層流に変化する。

**➡ 発散しないように  $\sqrt{\rho\Delta p}$  を補正する。**

- 逆止弁機能あり (checkValve = true)
- 逆流あり (checkValve = false, allowFlowReversal = true)
- 逆流なし (checkValve = false, allowFlowReversal = false)

## Regularizationの範囲

$$|\Delta p| < \Delta p_{turbulent} \quad \text{乱流から層流に繊維する圧力差}$$

## PartialValve

$$\Delta p_{small} = \begin{cases} \frac{\Delta p_{nominal}}{\dot{m}_{nominal}} \cdot \dot{m}_{small}, & use\_Re = true \\ system.dp\_small, & use\_Re = false \end{cases}$$

## ValveIncompressible, ValveVaporizing, ValveCompressible

$$\Delta p_{turbulent} = \begin{cases} \Delta p_{small}, & use\_Re = false \\ \max \left( \Delta p_{small}, \frac{(\mu_a + \mu_b)^2 \frac{\pi}{8} Re_{turbulent}^2}{\{\max(\text{relativeFlowCoefficient}(\text{opening}), 0.001) A_v\}(\rho_a + \rho_b)} \right), & use\_Re = true \end{cases}$$

$$\frac{1}{A_v} \frac{(\mu_a + \mu_b)^2}{\rho_a + \rho_b} \frac{\pi}{8} Re_{turbulent}^2 = \frac{1}{A_v} \frac{\pi D^2}{4} \frac{\rho_a + \rho_b}{2} u_{turbulent}^2 = \frac{2A}{A_v} \frac{\bar{\rho} u_{turbulent}^2}{2}$$

代入 ↑

$$Re_{turbulent} \equiv \frac{\bar{\rho} u D}{\bar{\mu}} = \frac{(\rho_a + \rho_b) u_{turbulent} D}{\mu_a + \mu_b} = 4000$$

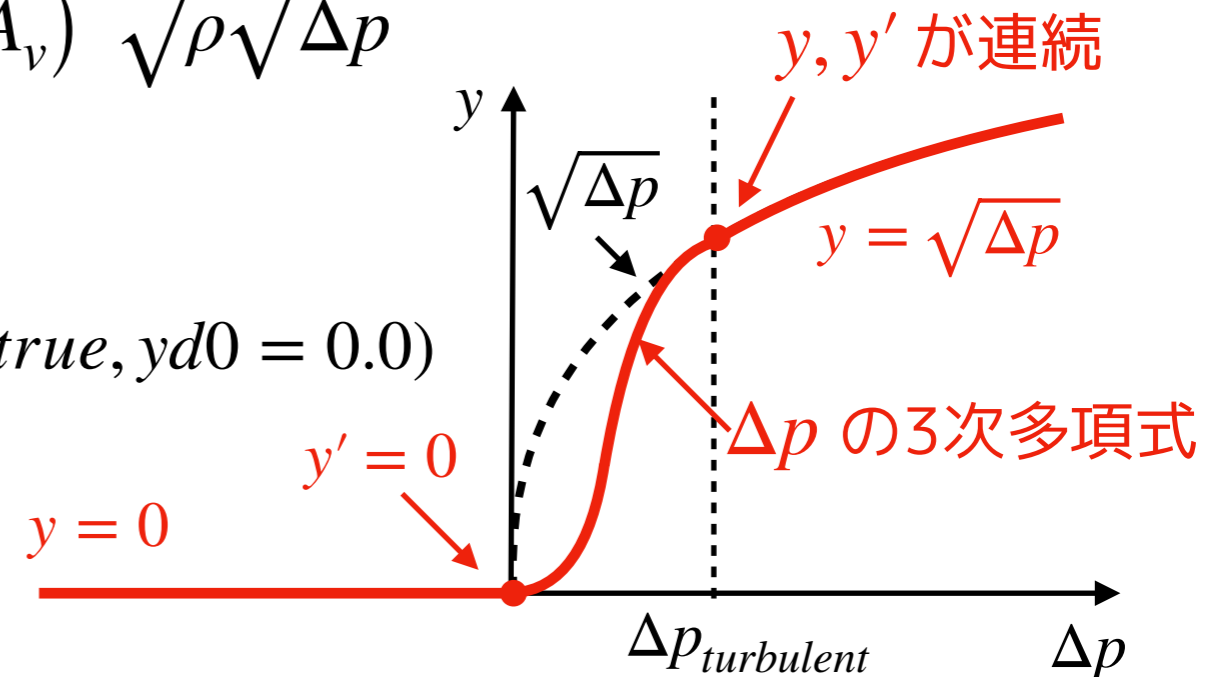
A 断面積

## 逆止弁機能あり(checkValve = true)

$$\dot{m} = \rho q = (\text{valveCharacteristic}(\text{opening}) A_v) \sqrt{\rho} \sqrt{\Delta p}$$

$\sqrt{\Delta p}$  を次の関数で近似する。

$$y = \text{regRoot2}(\Delta p, \Delta p_{\text{turbulent}}, 1.0, 0.0, \text{use\_yd0} = \text{true}, \text{yd0} = 0.0)$$



$$y = \text{regRoot2}(x, x_{\text{small}}, k1, k2, \text{use\_yd0}, \text{yd0})$$

$$y = \begin{cases} \sqrt{k_1 x}, & x \geq 0 \\ -\sqrt{k_2 x}, & x < 0 \end{cases}$$

の領域  $-x_{\text{small}} < x < x_{\text{small}}$  を、2つの3次多項式 (1つは  $-x_{\text{small}} < x < 0$ , 1つは  $0 \leq x < x_{\text{small}}$ ) で近似し、以下を満たすようにする。

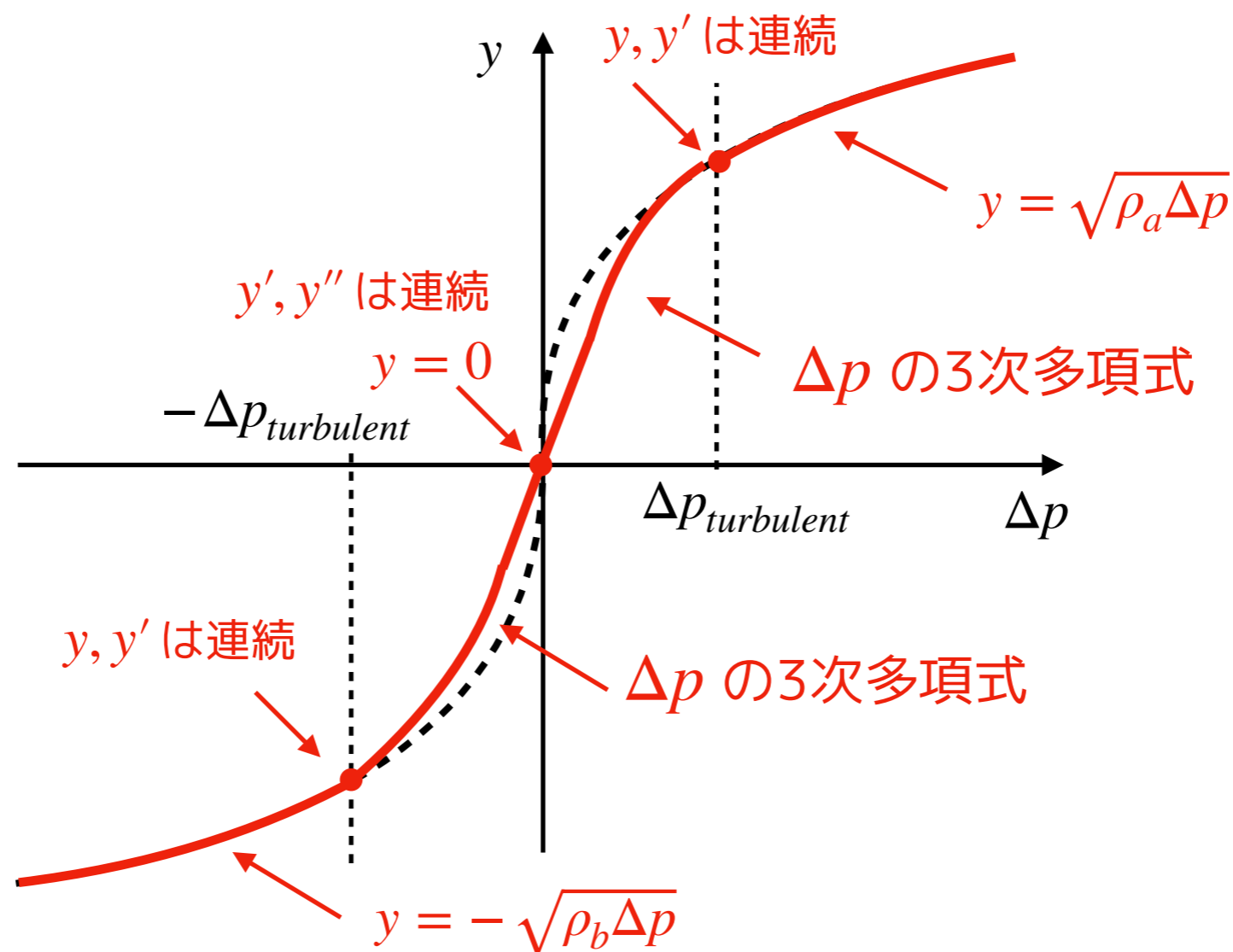
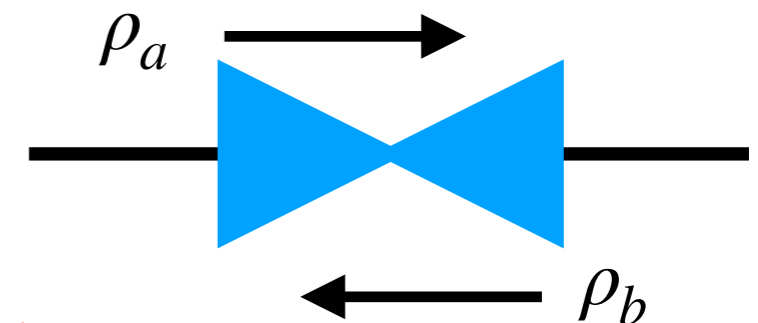
- $x = 0$  での微分が有限になる。
- 関数は、全ての領域で、1階微分まで連続となる。
- $\text{use\_yd0} = \text{false}$  にすると  $x = 0$  における2階微分が一致する。  $\text{use\_yd0} = \text{true}$  のときは  $x = 0$  における微分を  $\text{yd0}$  で指定する。多項式が単調増大になるように、  $\text{yd0}$  は必要に応じて修正される。

逆流あり(checkValve = false, allowFlowReversal = true)

$$\dot{m} = \rho q = (\text{valveCharacteristic}(\text{opening}) A_v) \sqrt{\rho \Delta p}$$

$\sqrt{\rho \Delta p}$  を次の関数で近似する。

$$y = \text{regRoot2}(\Delta p, \Delta p_{\text{turbulent}}, \rho_a, \rho_b)$$



逆流なし(checkValve = false, allowFlowReversal = false)

$$\dot{m} = \rho q = (\text{valveCharacteristic}(\text{opening}) A_v) \sqrt{\rho} \sqrt{\Delta p}$$

$\sqrt{\Delta p}$  を次の関数で近似する。逆流を考慮しないので  $\Delta p > 0$  のみを考える。

$$y = \text{regRoot}(\Delta p, \Delta p_{\text{turbulent}}) = \frac{\Delta p}{(\Delta p^2 + \Delta p_{\text{turbulent}}^2)^{\frac{1}{4}}}$$

$$y = \text{regRoot}(x, \text{delta})$$

有限の微分値を持つ非対称平方根近似関数

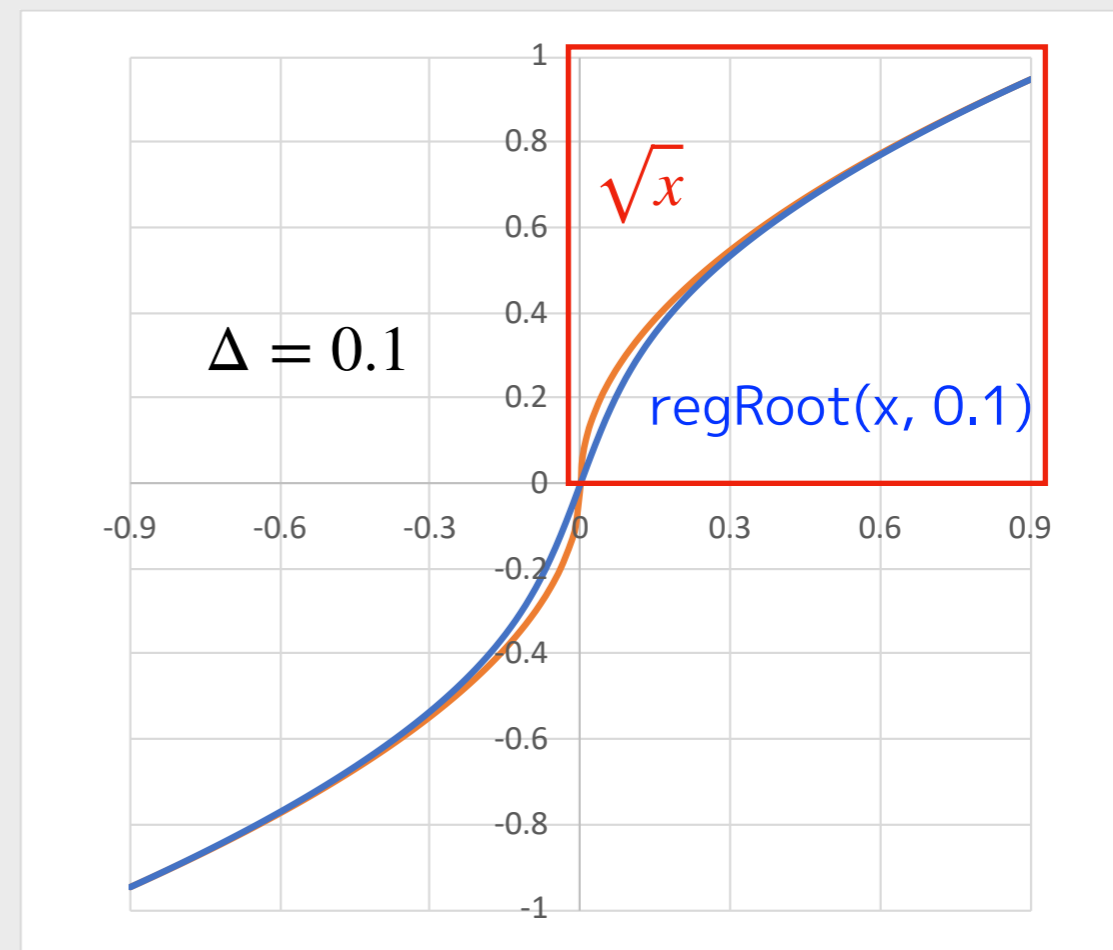
$$y = \frac{x}{(x^2 + \Delta^2)^{0.25}}$$

$$x \gg \Delta$$

$$y = \frac{x}{\sqrt{x}} = \sqrt{x}$$

$$x \ll \Delta$$

$$y = \frac{x}{\sqrt{\Delta}}$$




# ホモトピー変換 (homotopy transformation)

$$\lambda \cdot actual + (1 - \lambda) \cdot simplified$$

- 動的シミュレーションの初期化フェーズでは、大きな非線形方程式系を繰り返し計算で解く必要がある場合がある。
- ホモトピー法は、まず簡単に収束するモデル (simplified model) を解き、このモデルから実際のモデル(actual model) までモデルを連続的に変形させて解を追跡することによって非線形方程式系を解く方法である。

ホモトピーオペレータ


$$m_{flow} = \text{homotopy}(actual, simplified)$$

actual model

$$\text{valveCharacteristic}(\text{opening}) A_v \sqrt{\rho \Delta p}$$

simplified model

$$\text{valveCharacteristic}(\text{opening}) \cdot m_{flow\_nominal} \frac{\Delta p}{\Delta p_{nominal}}$$

# ValveIncompressible の方程式のホモトピー変換と regularization

ホモトピー変換

```
equation
  // m_flow = valveCharacteristic(opening)*Av*sqrt(d)*sqrt(dp);

  relativeFlowCoefficient = valveCharacteristic(opening_actual);
  if checkValve then
    m_flow = homotopy(relativeFlowCoefficient*Av*sqrt(Medium.density(state_a))*
                      Utilities.regRoot2(dp,dp_turbulent,1.0,0.0,use_yd0=true,yd0=0.0),
                      relativeFlowCoefficient*m_flow_nominal*dp/dp_nominal);
    /* In Modelica 3.1 (Disadvantage: Unnecessary event at dp=0, and instead of smooth=2)
    m_flow = valveCharacteristic(opening)*Av*sqrt(Medium.density(state_a))*
              (if dp>=0 then Utilities.regRoot(dp, dp_turbulent) else 0);
    */
  elseif not allowFlowReversal then
    m_flow = homotopy(relativeFlowCoefficient*Av*sqrt(Medium.density(state_a))*
                      Utilities.regRoot(dp, dp_turbulent),
                      relativeFlowCoefficient*m_flow_nominal*dp/dp_nominal);
  else
    m_flow = homotopy(relativeFlowCoefficient*Av*
                      Utilities.regRoot2(dp,dp_turbulent,Medium.density(state_a),Medium.density(state_b)),
                      relativeFlowCoefficient*m_flow_nominal*dp/dp_nominal);
    /* In Modelica 3.1 (Disadvantage: Unnecessary event at dp=0, and instead of smooth=2)
    m_flow = smooth(0, Utilities.regRoot(dp, dp_turbulent)*(if dp>=0 then sqrt(Medium.density(state_a))
    else sqrt(Medium.density(state_b))));
    */
end equation
```

regularization

逆止弁機能

逆流なし

逆流あり

# ValveVaporizing の方程式のホモトピー変換と regularization

equation

```

p_in = port_a.p;
p_out = port_b.p;
T_in = Medium.temperature(state_a);
p_sat = Medium.saturationPressure(T_in);
Ff = 0.96 - 0.28*sqrt(p_sat/Medium.fluidConstants[1].criticalPressure);
Fl = Fl_nominal*FlCharacteristic(opening_actual);
dpEff = if p_out < (1 - Fl^2)*p_in + Ff*Fl^2*p_sat then
    Fl^2*(p_in - Ff*p_sat) else dp
    "Effective pressure drop, accounting for possible choked conditions";
// m_flow = valveCharacteristic(opening)*Av*sqrt(d)*sqrt(dpEff);
if checkValve then
    m_flow = homotopy(valveCharacteristic(opening_actual)*Av*sqrt(Medium.density(state_a))*
        Utilities.regRoot2(dpEff,dp_turbulent,1.0,0.0,use_yd0=true,yd0=0.0),
        valveCharacteristic(opening_actual)*m_flow_nominal*dp/dp_nominal);
    /* In Modelica 3.1 (Disadvantage: Unnecessary event at dpEff=0, and instead of smooth=2)
m_flow = valveCharacteristic(opening)*Av*sqrt(Medium.density(state_a))*
    (if dpEff>=0 then Utilities.regRoot(dpEff, dp_turbulent) else 0);
*/
elseif not allowFlowReversal then
    m_flow = homotopy(valveCharacteristic(opening_actual)*Av*sqrt(Medium.density(state_a))*
        Utilities.regRoot(dpEff, dp_turbulent),
        valveCharacteristic(opening_actual)*m_flow_nominal*dp/dp_nominal);
else
    m_flow = homotopy(valveCharacteristic(opening_actual)*Av*
        Utilities.regRoot2(dpEff,dp_turbulent,Medium.density(state_a),Medium.density(state_b)),
        valveCharacteristic(opening_actual)*m_flow_nominal*dp/dp_nominal);
    /* In Modelica 3.1 (Disadvantage: Unnecessary event at dp=0, and instead of smooth=2)
m_flow = valveCharacteristic(opening)*Av*
    smooth(0, Utilities.regRoot(dpEff, dp_turbulent)*(if dpEff>=0 then sqrt(Medium.density(state_a)) else
sqrt(Medium.density(state_b))));
*/
end if;

annotation ( ...);
end ValveVaporizing;
    
```

ホモトピー変換

regularization

逆止弁機能

逆流なし

逆流あり

# 開度のフィルタリング

(開度の急変をさけるオプション)

filteredOpening = true のとき有効

riseTime 遅延

leakageOpening 開度の下限値

カットオフ周波数  $f_{\text{cut}} = 5/(2 \cdot \pi \cdot \text{riseTime})$

## PartialValve

filter (second order, criticalDamping)

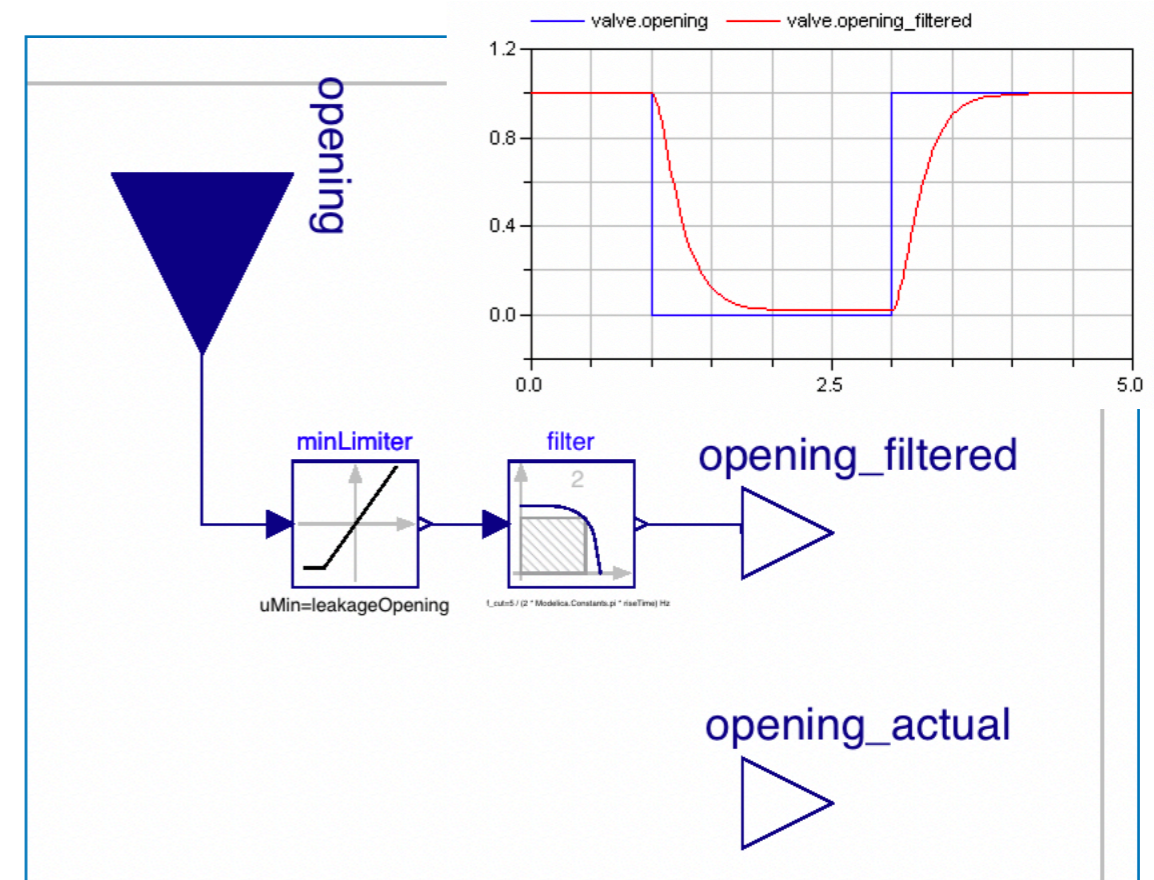
```
Modelica.Blocks.Continuous.Filter filter(order=2, f_cut=5/(2*Modelica.Constants.pi
    *riseTime)) if filteredOpening
    annotation (Placement(transformation(extent={{34,44},{48,58}}))));
```

## minLimiter

```
block MinLimiter "Limit the signal above a threshold"
    parameter Real uMin=0 "Lower limit of input signal";
    extends Modelica.Blocks.Interfaces.SISO;

    equation
        y = smooth(0, noEvent( if u < uMin then uMin else u));
        annotation ( ... );
    end MinLimiter;
```

```
MinLimiter minLimiter(uMin=leakageOpening)
    annotation (Placement(transformation(extent={{10,44},{24,58}}))));
```



## 例題 ValveExample2

NeedleValveTest1

NeedleValveTest2

ButterflyValveTest1

ButterflyValveTest2

WaterValveTest1

ValveIncompressible の例題

ValveVaporizing の例題

ValveIncompressible の改造

固有流量特性を CombiTable1D を使って設定する。

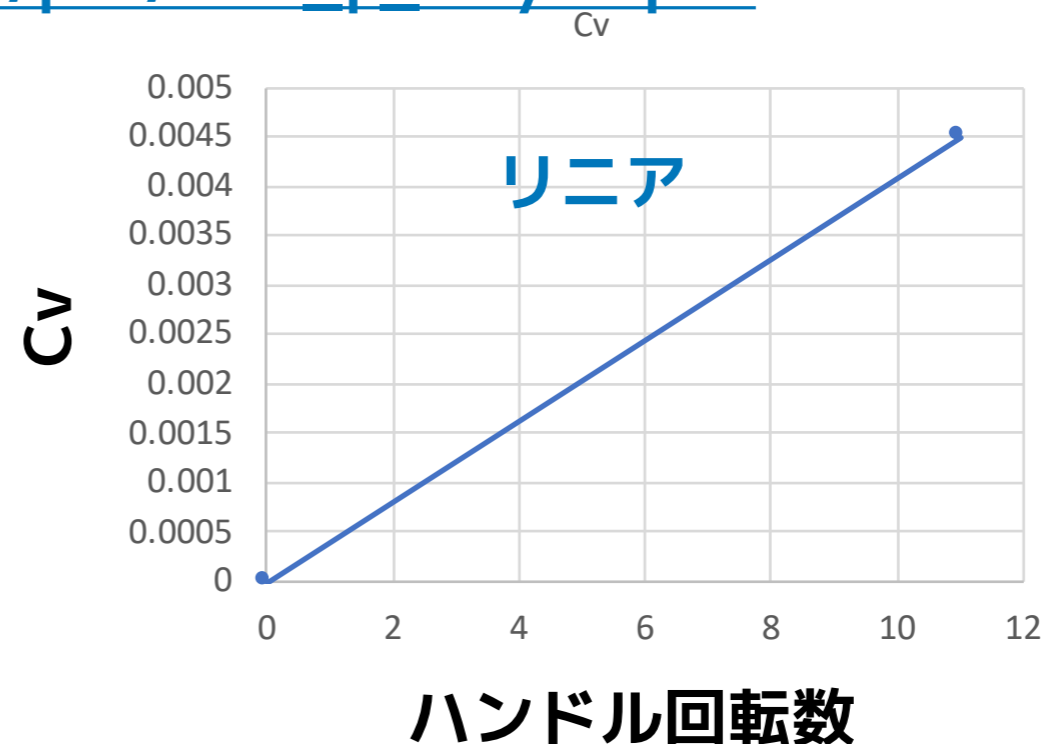
# NeedleValveTest1

参考 株式会社フジキン, 微量流量調整バルブ

[https://www.fujikin.co.jp/support/pdf/821\\_p\\_biryou.pdf](https://www.fujikin.co.jp/support/pdf/821_p_biryou.pdf)

## ニードルバルブ（流量調整バルブ）

呼び径	6.35	
入口内径	6	mm
Cv	0.0045	



## 流量特性

回転数	Cv
0	0
11	0.0045

## 全開時の運転条件

Av	1.08E-07	m2
水密度(20℃)	998.233	kg/m3
差圧	5.00E+06	Pa
質量流量	7.63E-03	kg/s

$$Av = 24 \times 10^6 Cv$$

$$\rho$$

$$\Delta p$$

$$\dot{m} = Av \sqrt{\rho \Delta p}$$

## テストモデル

圧力差を固定して、開度を全閉から全開まで変化させる。

### ハンドル回転数

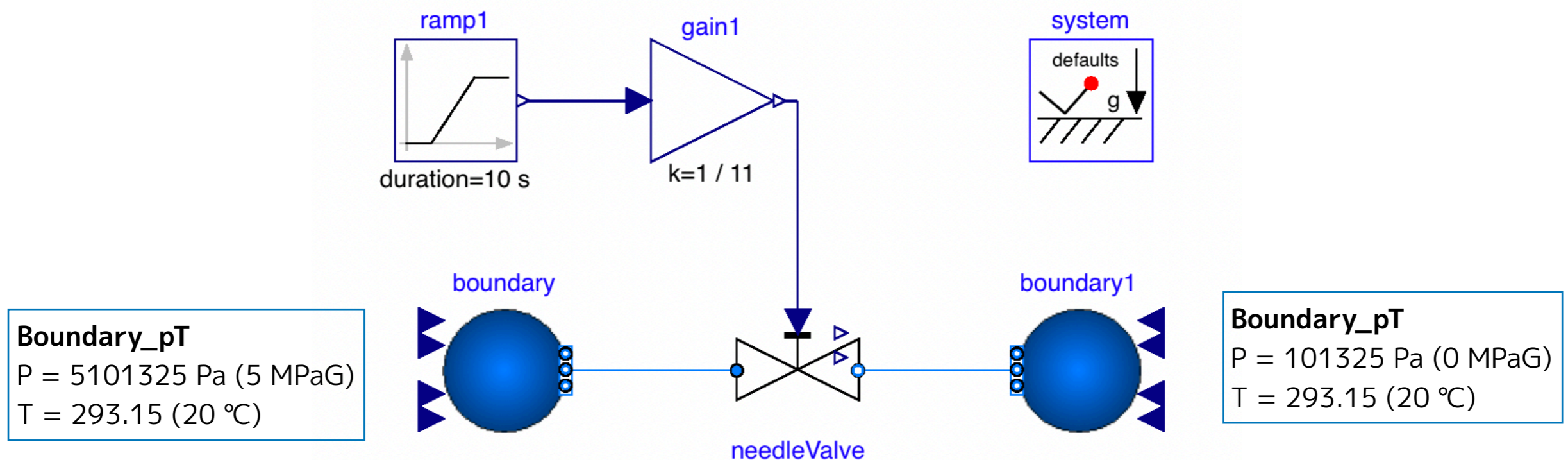
**Modelica.Blocks.Sources.Ramp**  
 height = 11  
 duration = 10 s  
 offset = 0  
 startTime = 0 s

### 作動流体

Medium = Modelica.Media.Water.StandardWater

### バルブ開度

**Modelica.Blocks.Math.Gain**  
 k = 1/11



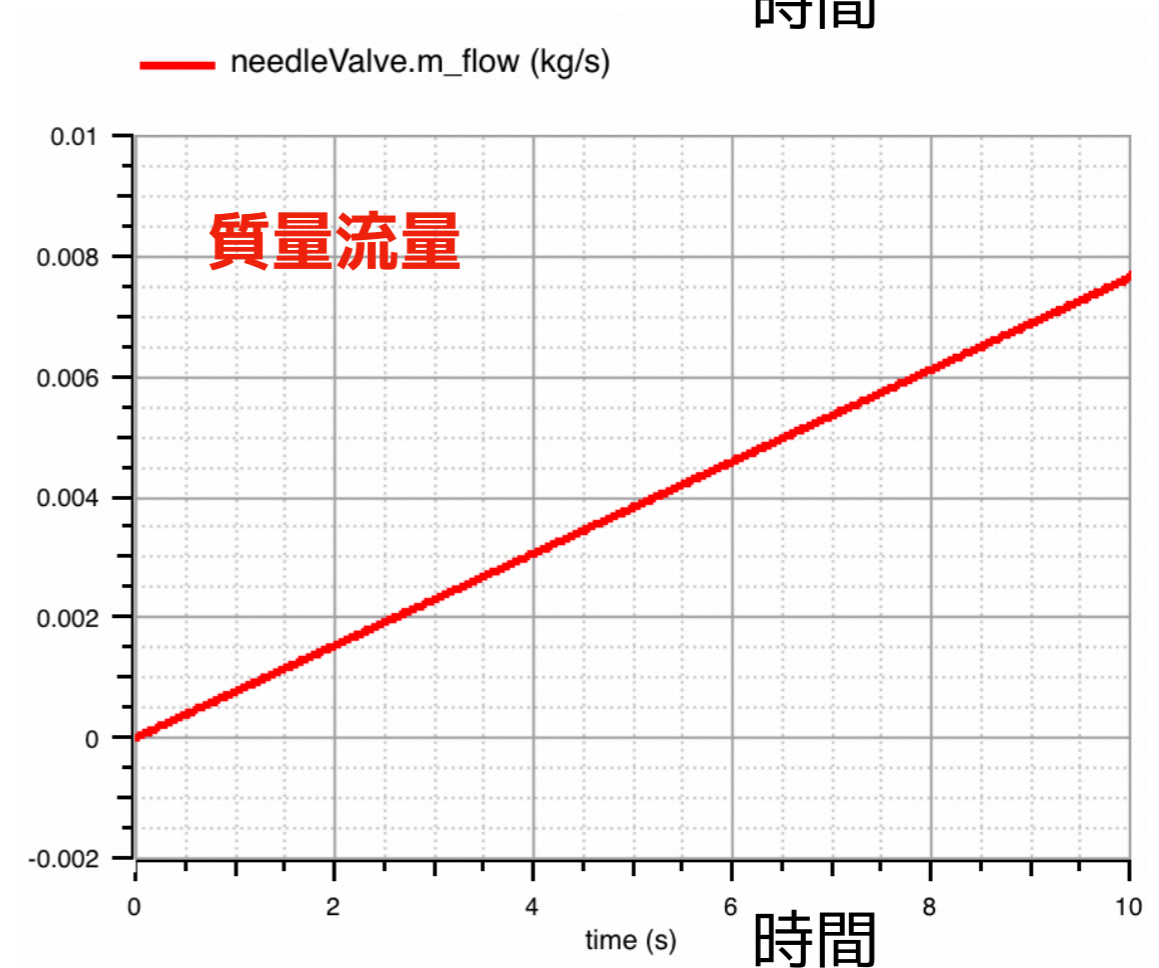
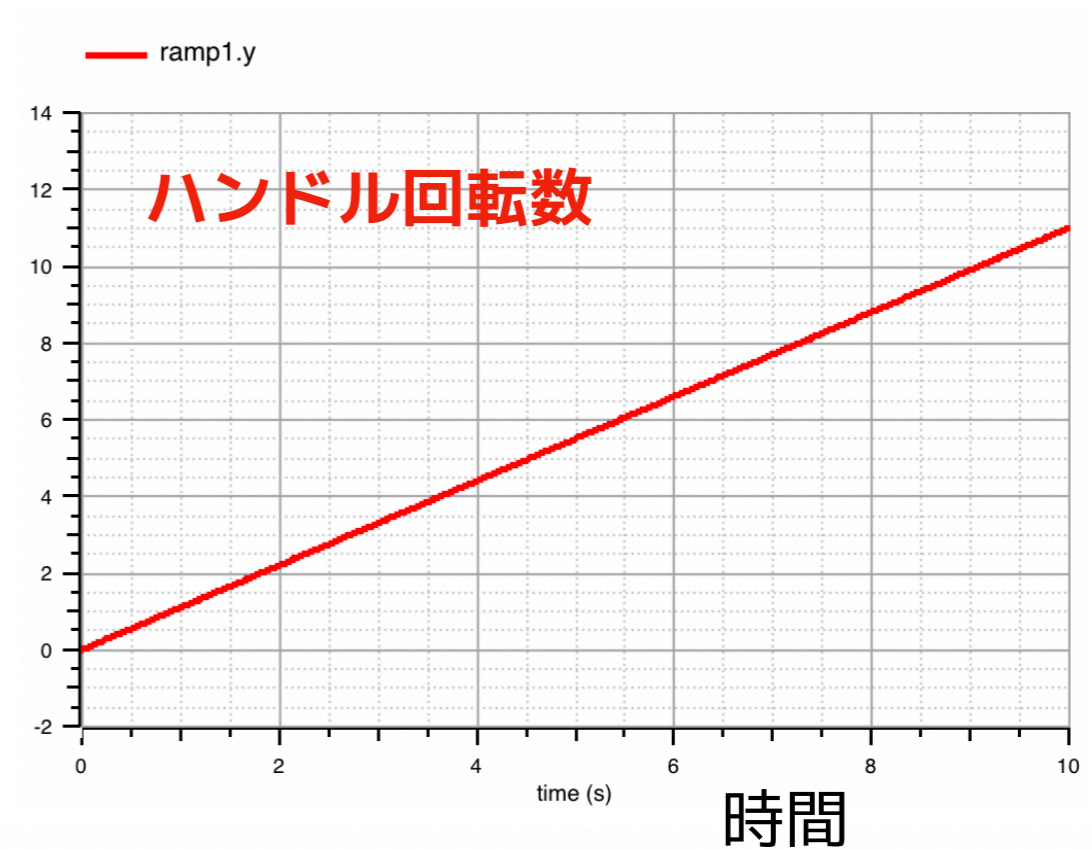
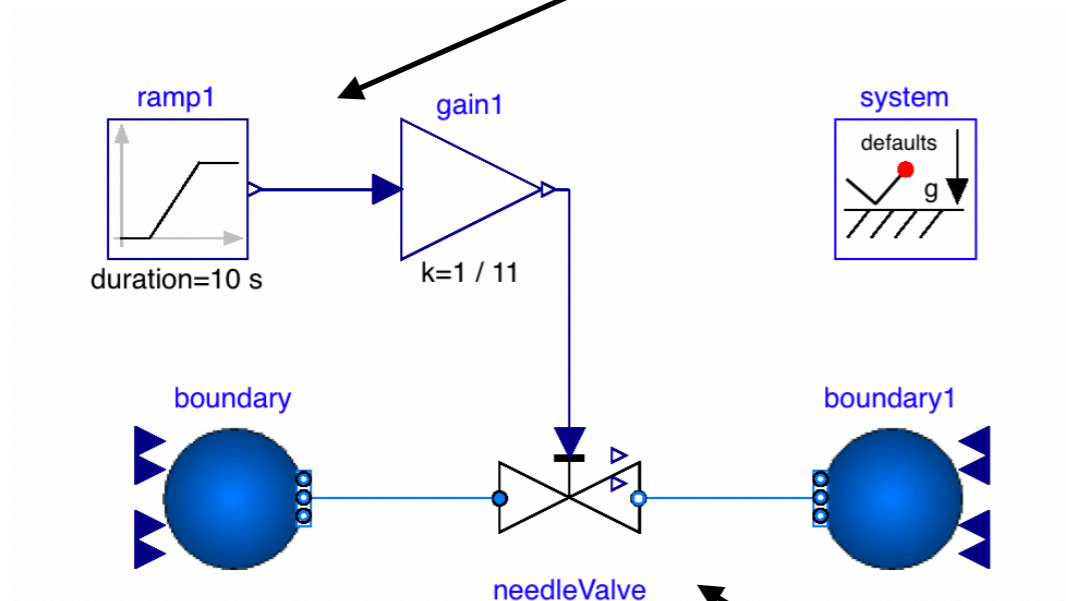
### ValveIncompressible

CvData = Modelica.Fluid.Types.CvTypes.Cv  
 Cv = 0.0045  
 dp\_nominal = 5e+06 [Pa]  
 m\_flow\_nominal = 7.63e-3 [kg/s]

## テストモデルのソースコード

```
model NeedleValveTest1
  replaceable package Medium = Modelica.Media.Water.StandardWater;
  Modelica.Fluid.Valves.ValveIncompressible needleValve(redeclare package Medium = Medium,
    Cv = 0.0045, CvData = Modelica.Fluid.Types.CvTypes.Cv,
    dp_nominal = 5.10132e+6, m_flow_nominal = 7.63e-3) annotation( ...);
  Modelica.Fluid.Sources.Boundary_pT boundary(redeclare package Medium = Medium,
    T = 293.15, nPorts = 1, p = 5101325) 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 = 11, offset = 0, startTime = 0) annotation( ...);
  Modelica.Blocks.Math.Gain gain1(k = 1 / 11) annotation( ...);
  inner Modelica.Fluid.System system annotation( ...);
equation
  connect(gain1.y, needleValve.opening) annotation( ...);
  connect(ramp1.y, gain1.u) annotation( ...);
  connect(boundary1.ports[1], needleValve.port_b) annotation( ...);
  connect(boundary.ports[1], needleValve.port_a) annotation( ...);
  annotation( ...);
end NeedleValveTest1;
```

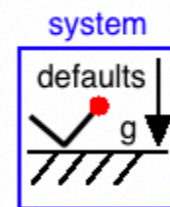
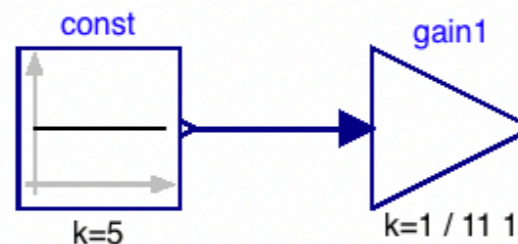
# シミュレーション結果



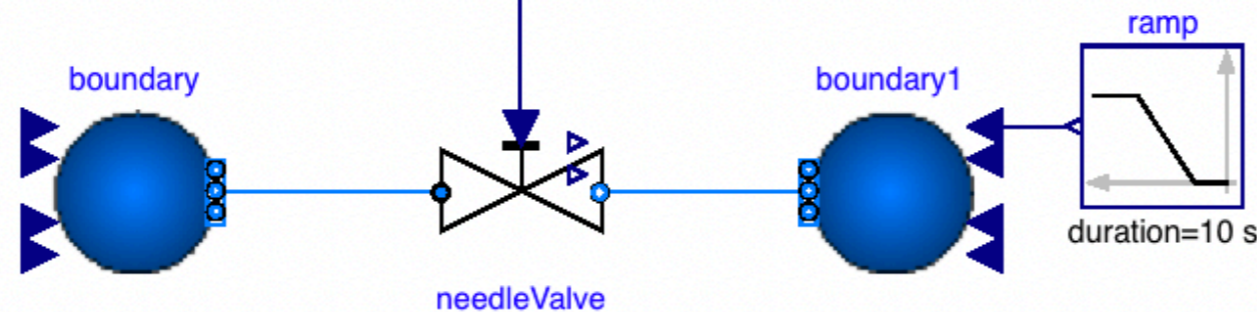
# NeedleValveTest2

開度を固定して出口の圧力を下げ、  
圧力差を増大させる。

**Modelica.Blocks.Sources.Constant**  
k = 5



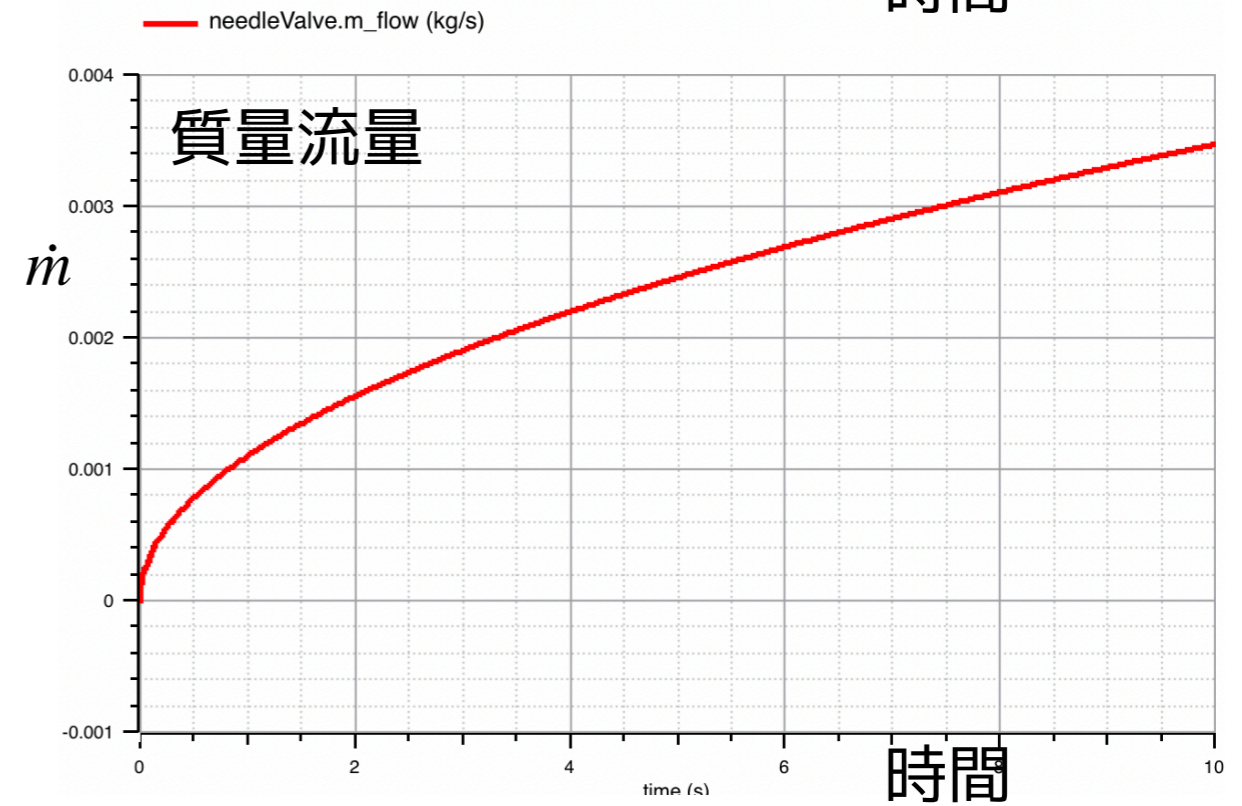
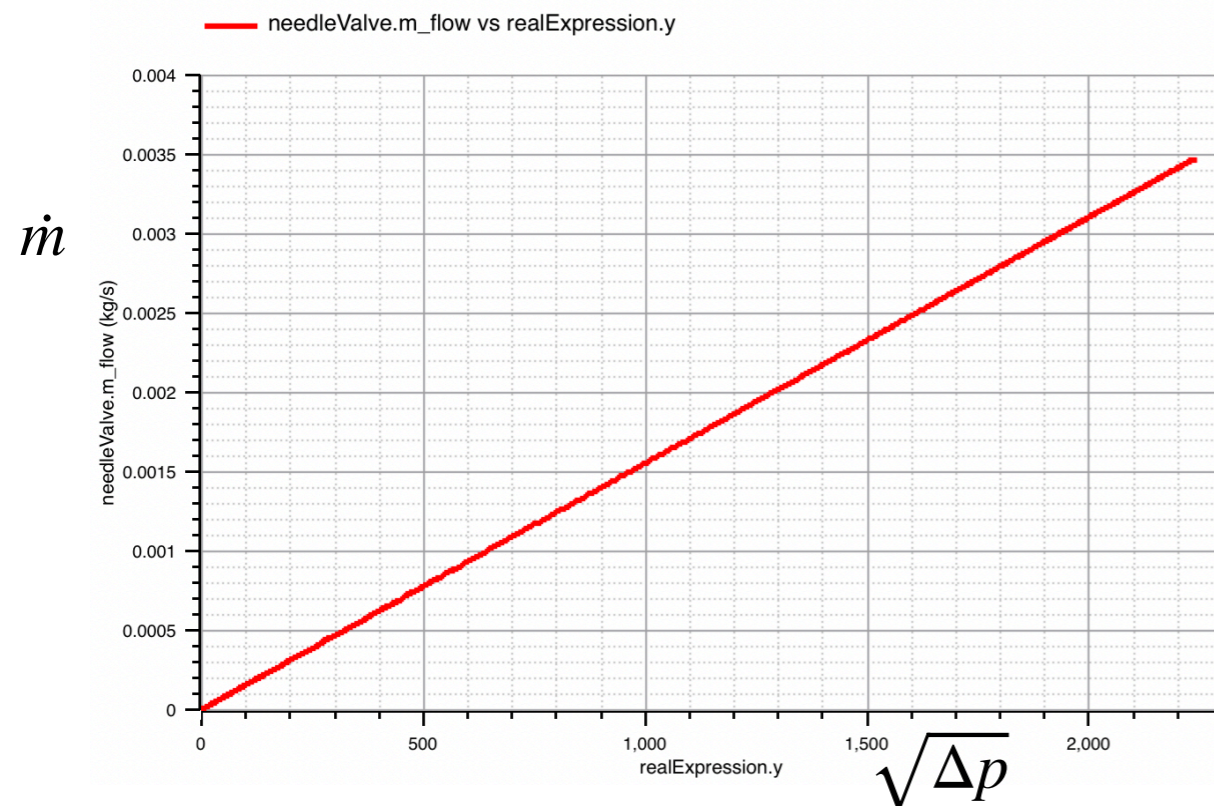
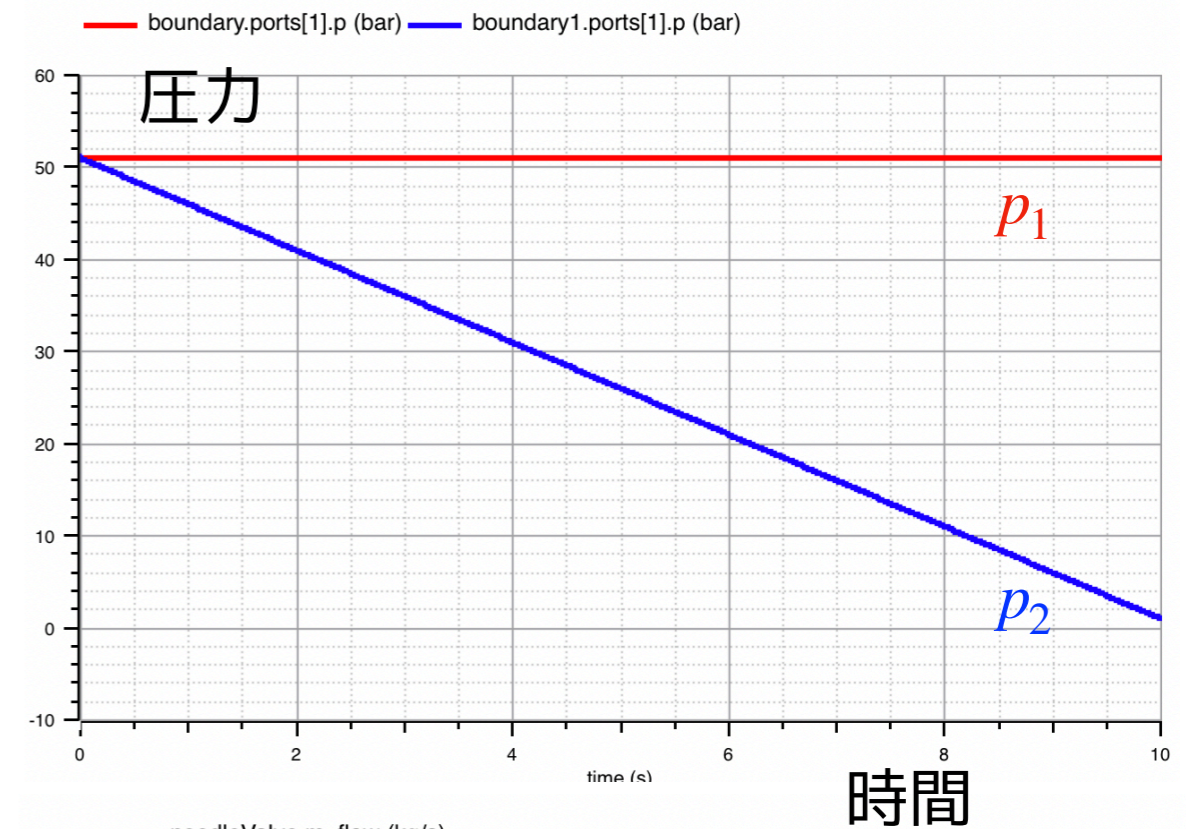
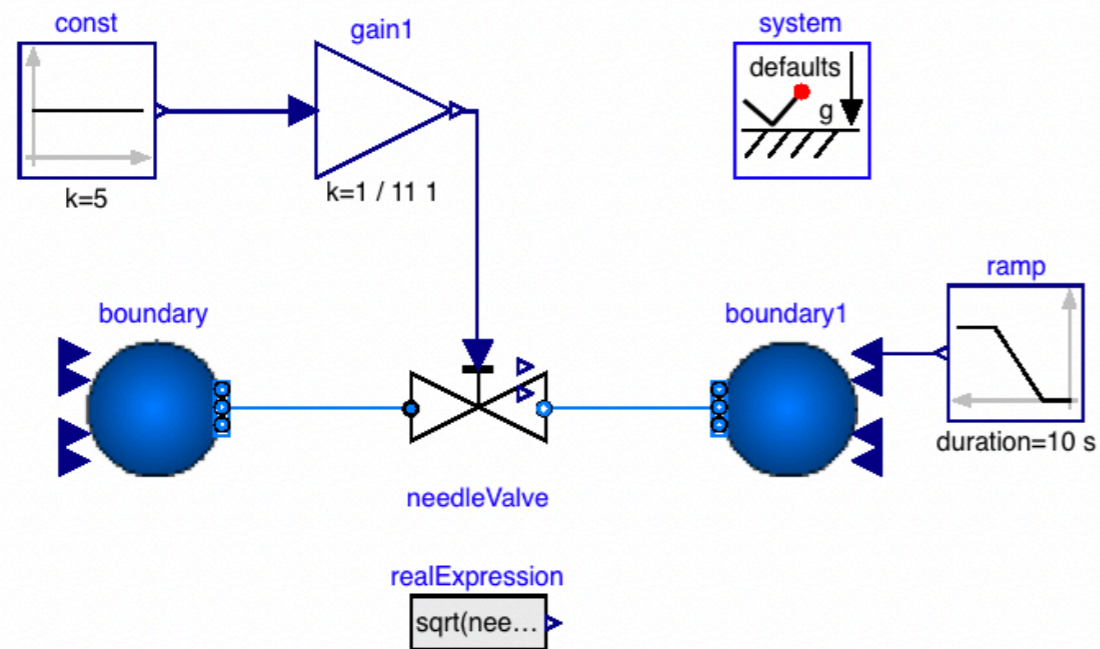
**Modelica.Blocks.Sources.Ramp**  
height = -5000000  
duration = 10 s  
offset = 5101325  
startTime = 0 s



**realExpression**  
sqrt(nee...)

**realExpression**  
k = sqrt(needleValve.dp)

# シミュレーション結果



# ButterflyValveTest1

参考 巴バルブ ロータリーコントロールバルブ

<http://www.tomoevalve.com/product/file/1505/313efe030c.pdf>

## バタフライバルブ

呼び径		150	mm
内径	d	143	mm

## 全開時の運転条件

断面積	A	0.016060607	m <sup>2</sup>
許容流速	u	5	m/s
水密度	ρ	998.233	kg/
質量流量	m	80.16113975	kg/s

流量係数	Cv	790	
	Av	0.01896	m <sup>2</sup>
圧力差	Δp	17906.84928	Pa
上流圧力	p1	119231.8493	Pa
下流圧力	p2	101325	Pa

## 流量特性

角度(deg)	Cv	FL
10	26	0.85
20	72	0.8
30	140	0.78
40	239	0.76
50	375	0.73
60	555	0.71
70	790	0.7

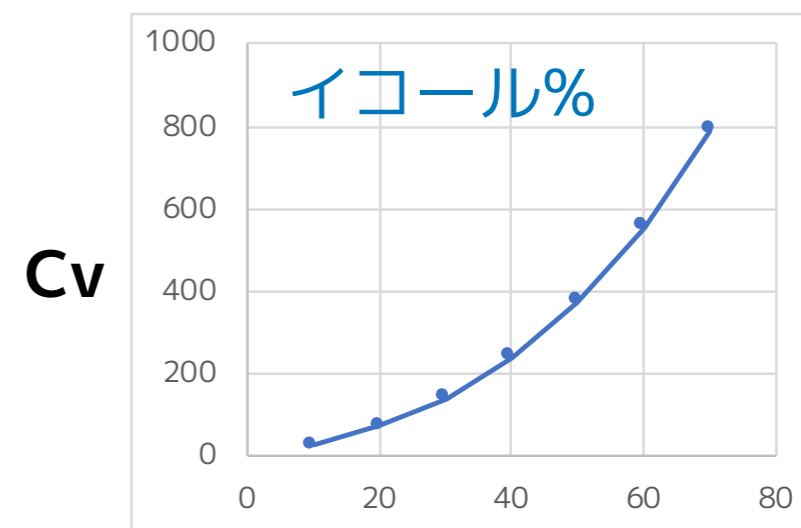
$$A = \frac{\pi d^2}{4}$$

$$Av = 24 \times 10^6 C_v$$

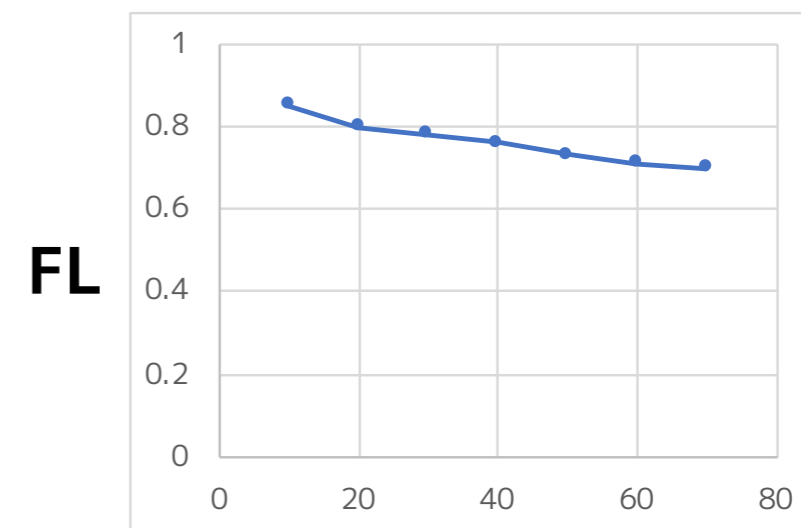
$$\dot{m} = \rho u A$$

$$\Delta p = \frac{1}{\rho} \left( \frac{\dot{m}}{Av} \right)^2$$

$$p_1 = p_2 + \Delta p$$



角度[deg]



角度[deg]

# 固有流量特性

角度	Cv	正規化した角度 開度	正規化した Cv 固有流量特性
0.01		0.000142857	
1		0.014285714	
5		0.071428571	
10	26	0.142857143	0.032911392
15		0.214285714	
20	72	0.285714286	0.091139241
25		0.357142857	
30	140	0.428571429	0.17721519
40	239	0.571428571	0.302531646
50	375	0.714285714	0.474683544
60	555	0.857142857	0.702531646
70	790	1	1

開度	固有流量特性	Cv
0.00014286	6.89979E-05	0.054508313
0.01428571	0.006899786	5.45083131
0.07142857	0.034498932	27.25415655
0.14285714	0.068997865	54.5083131
0.21428571	0.103496797	81.76246965
0.28571429	0.137995729	109.0166262
0.35714286	0.172494662	136.2707828
0.42857143	0.209732037	165.6883094
0.57142857	0.309919271	244.8362239
0.71428571	0.457965105	361.7924327
0.85714286	0.676731191	534.6176408
1	1	790

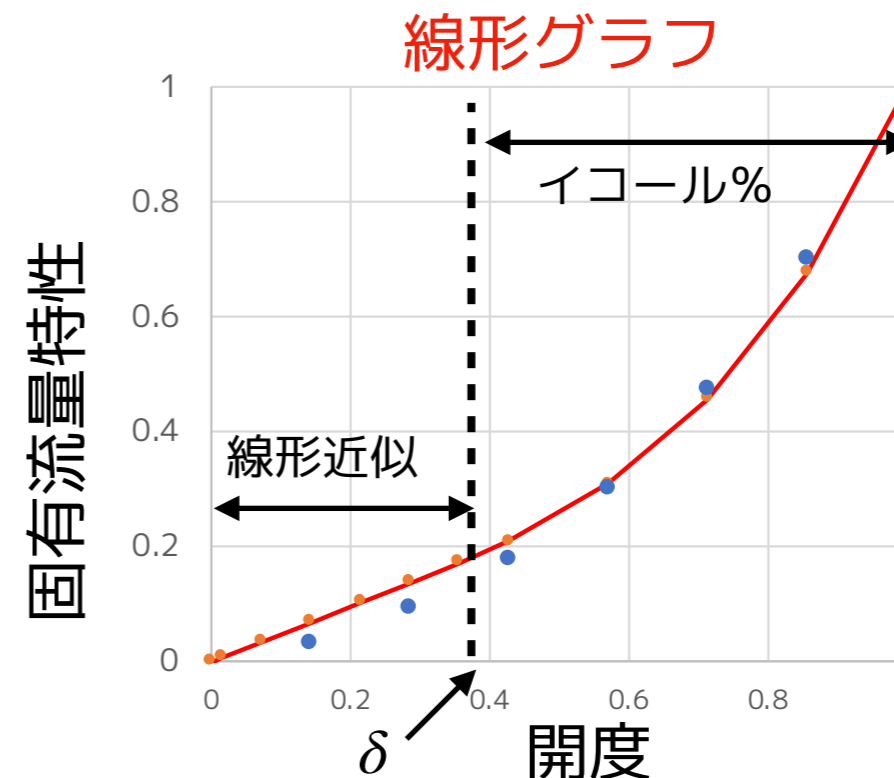
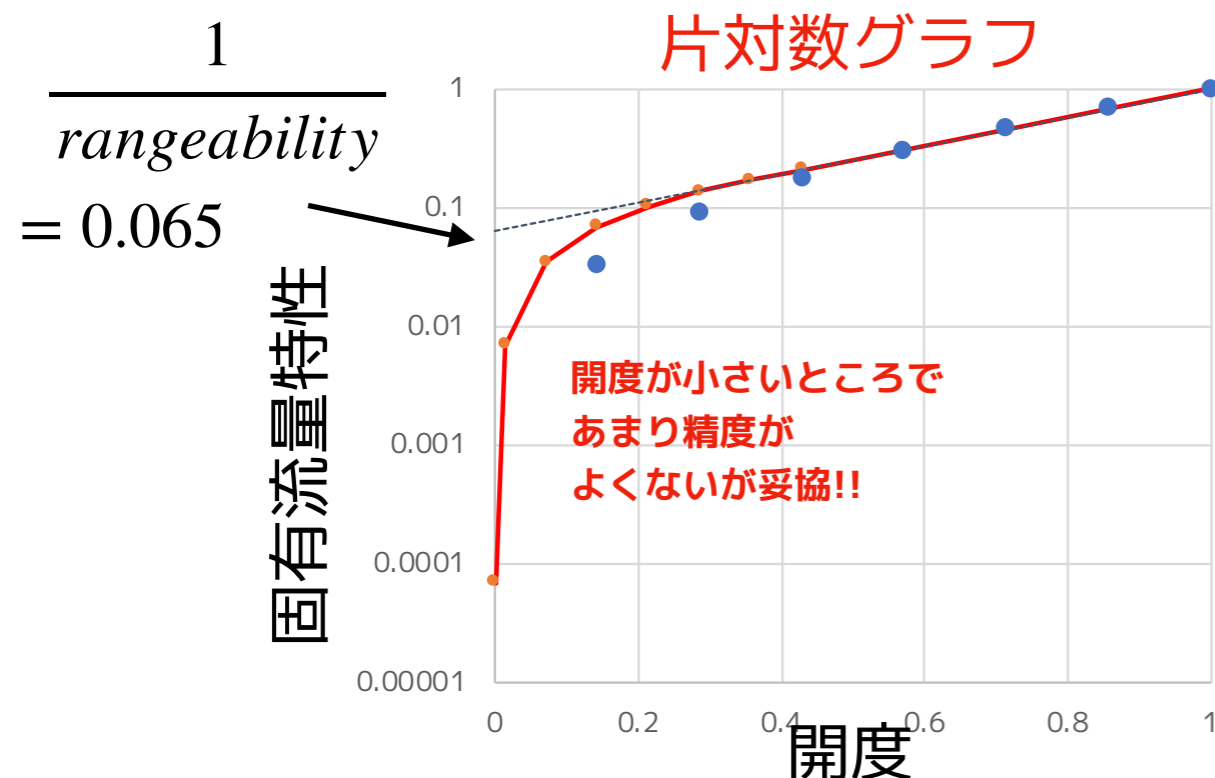
## イコール% によるフィッティングの計算式

$$rc = \begin{cases} rangeability^{pos-1}, & pos > \delta \\ \frac{pos}{\delta} rangeability^{\delta-1}, & pos \leq \delta \end{cases}$$

## グラフより以下を調整する

$$rangeability = \frac{1}{0.065} = 15.3846$$

$$\delta = 0.37$$



# 圧力回復特性

fl\_nominal

最大角度で正規化する

最大値が1になるように正規化する

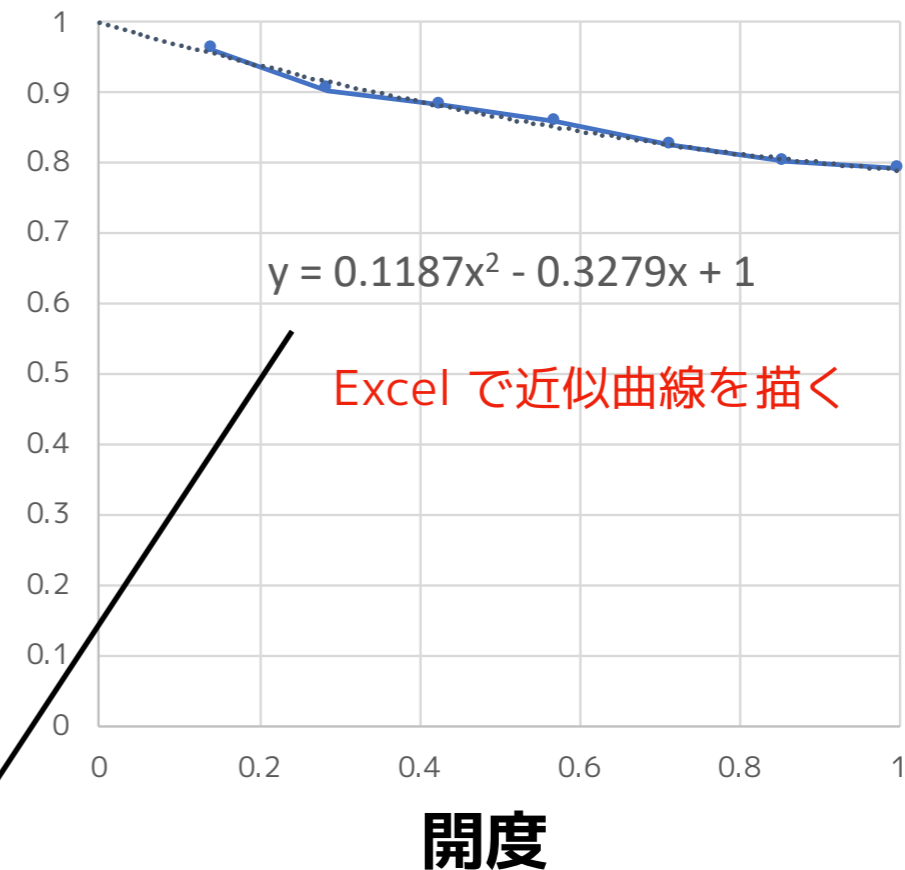
角度	FL	開度	FICharacteristic
0	0.8843	0	1
10	0.85	0.142857143	0.961212258
20	0.8	0.285714286	0.904670361
30	0.78	0.428571429	0.882053602
40	0.76	0.571428571	0.859436843
50	0.73	0.714285714	0.825511704
60	0.71	0.857142857	0.802894945
70	0.7	1	0.791586566

正規化した開度とFLの関係から  
Excel で近似曲線を描き、  
function を作成する。

fl\_butterfly

```
function fl_butterfly
  extends Modelica.Fluid.Valves.BaseClasses.ValveCharacteristics.baseFun;
algorithm
  rc := (0.1187 * pos - 0.3279) * pos + 1.0;
end fl_butterfly;
```

FICharacteristic



## テストモデル

圧力差を固定して、開度を全閉から全開まで変化させる。

### Modelica.Blocks.Sources.Ramp

height = 69.9  
duration = 10  
offset = 0  
startTime=0

### Modelica.Blocks.Math.Gain

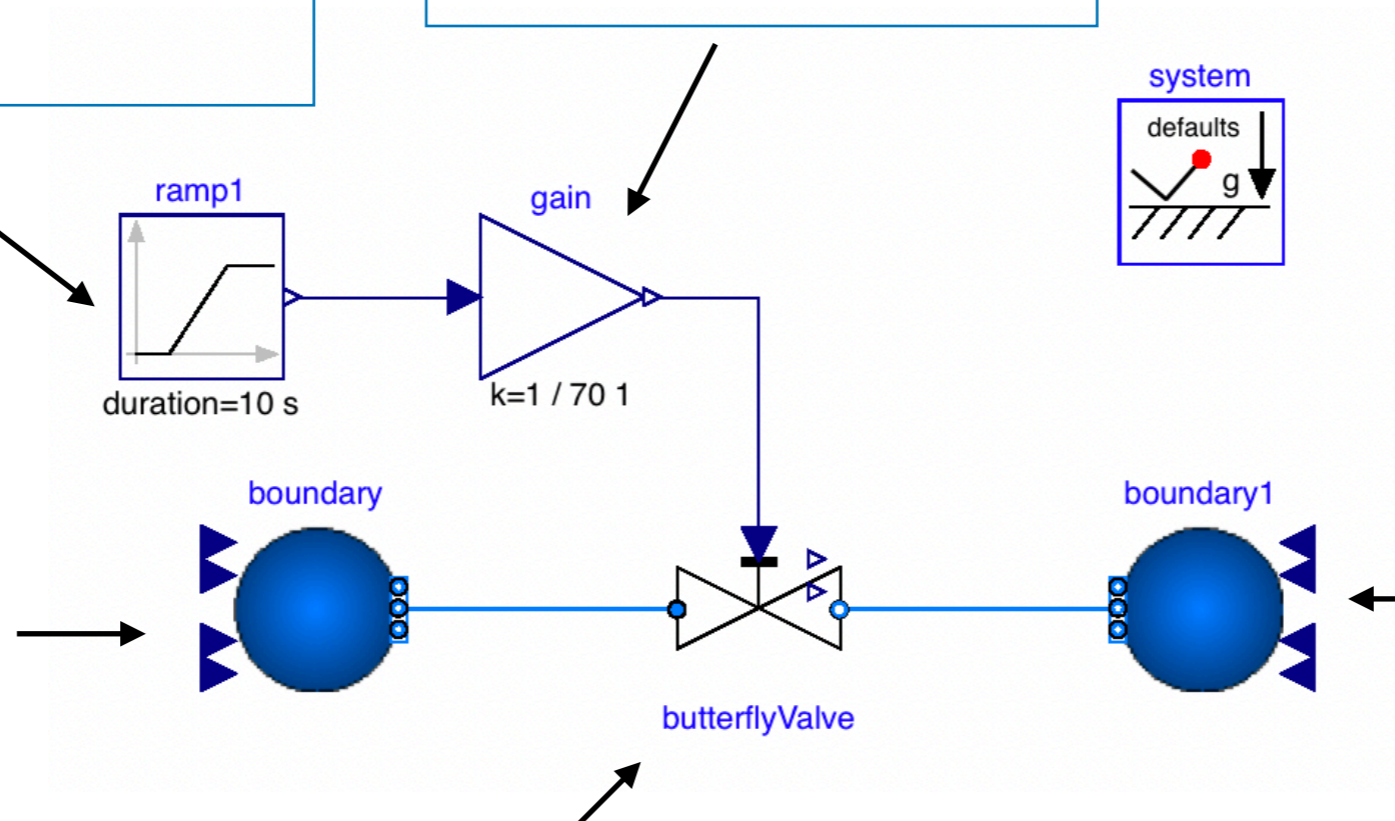
k = 1/70

### Boudary\_pT

p = 119231.849  
T=293.15

### Bondary\_pT

p=101325  
T=293.15



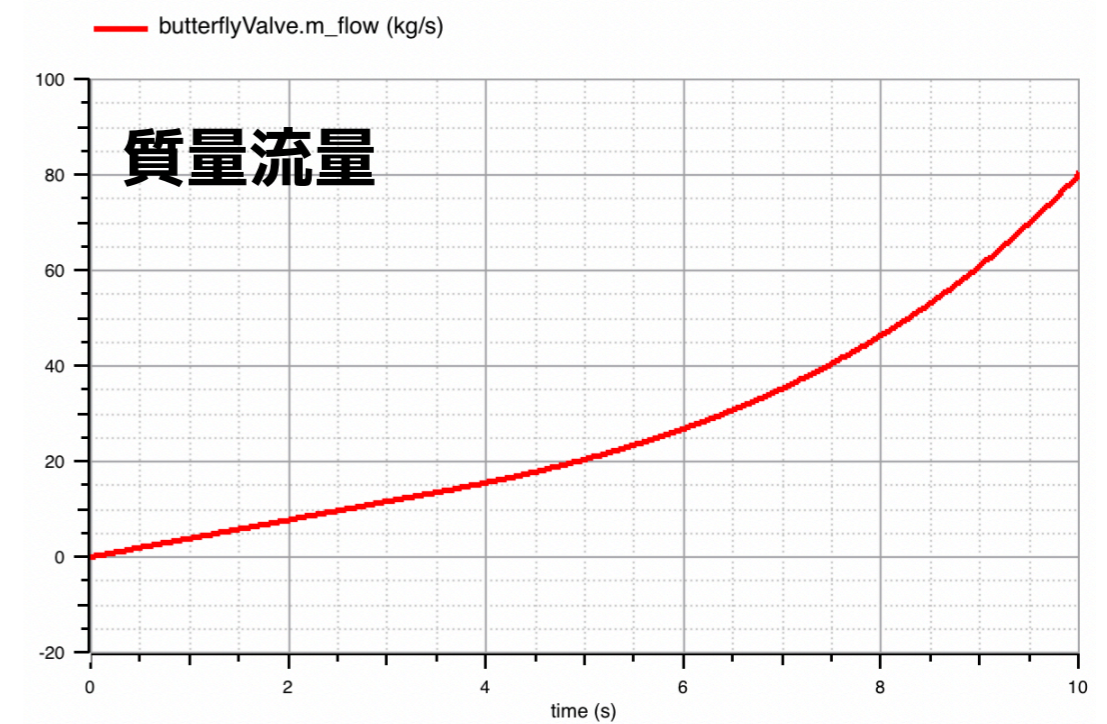
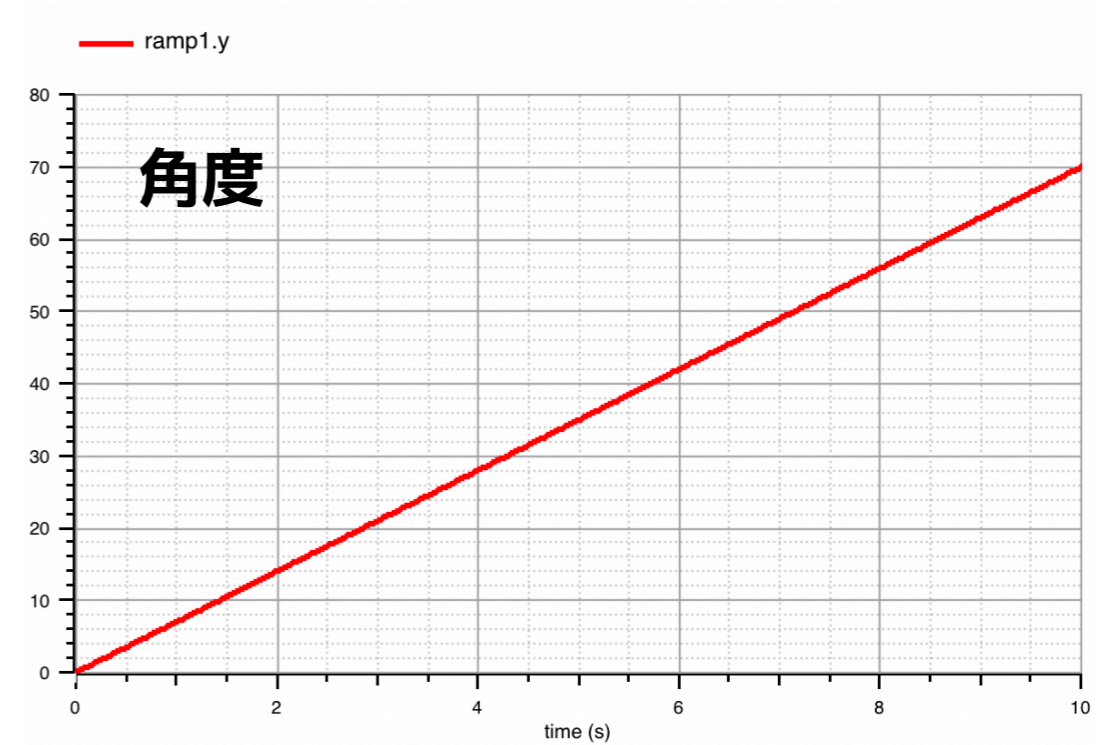
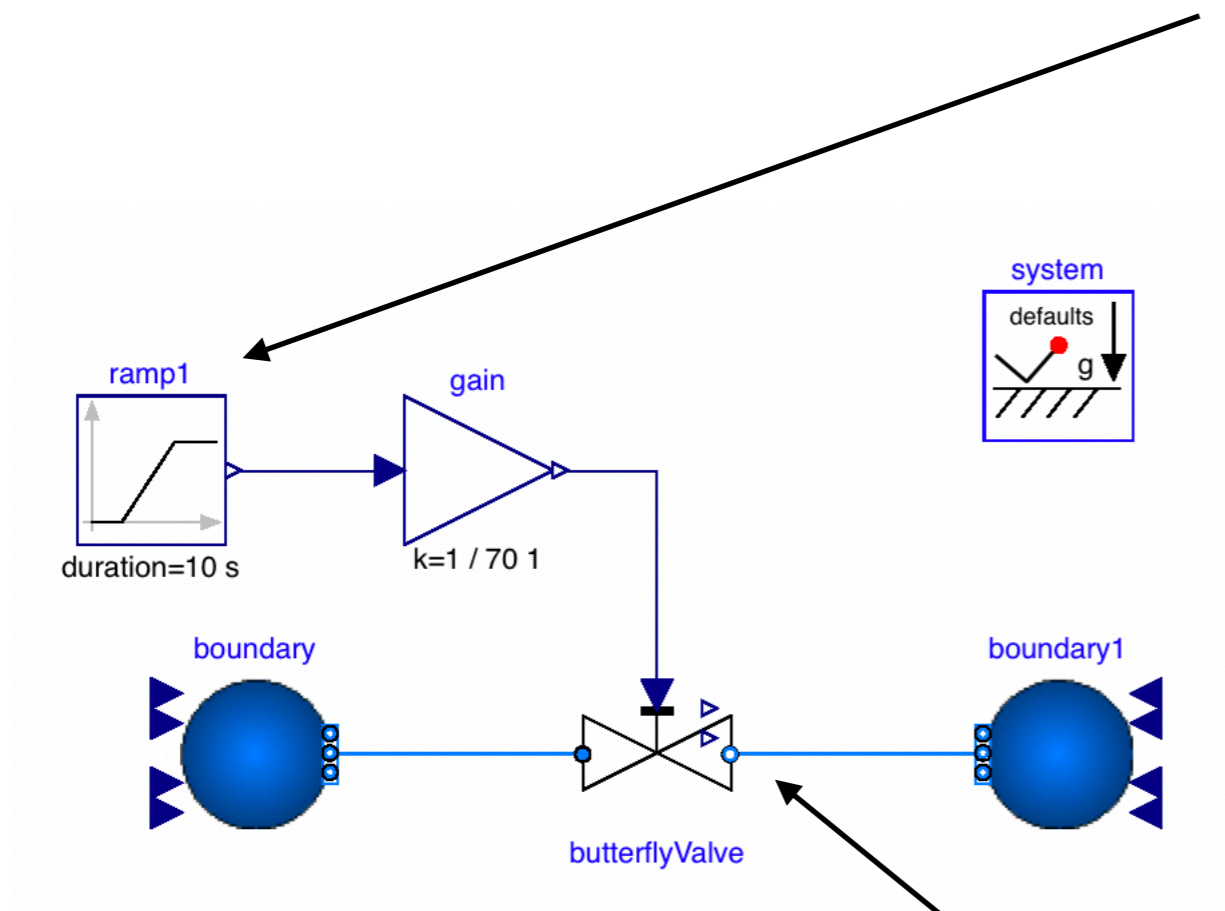
### ValveVaporizing

CvData = Modelica.Fluid.Types.CvTypes.Cv  
Cv 0 = 790  
dp\_nominal = 0.18 [bar]  
m\_flow\_nominal = 80 [kg/s]

## テストモデルのソースコード

```
model ButterflyValveTest
  replaceable package Medium = Modelica.Media.Water.StandardWater;
  Modelica.Fluid.Valves.ValveVaporizing butterflyValve(
    redeclare package Medium = Medium,
    redeclare function valveCharacteristic = Modelica.Fluid.Valves.BaseClasses.ValveCharacteristics.equalPercentage(
      rangeability = 15.38461538, delta = 0.37),
    redeclare function flCharacteristic = fl_butterfly,
    Cv = 790, CvData = Modelica.Fluid.Types.CvTypes.Cv, Fl_nominal = 0.8843, dp_nominal = 18000, m_flow_nominal = 80)
    annotation( ...);
  Modelica.Fluid.Sources.Boundary_pT boundary(
    redeclare package Medium = Medium, T = 293.15, nPorts = 1, p = 119231.849) 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 = 69.9, offset = 0, startTime = 0) annotation( ...);
  inner Modelica.Fluid.System system annotation( ...);
  Modelica.Blocks.Math.Gain gain(k = 1 / 70) annotation( ...);
equation
  connect(butterflyValve.port_b, boundary1.ports[1]) annotation( ...);
  connect(boundary.ports[1], butterflyValve.port_a) annotation( ...);
  connect(gain.y, butterflyValve.opening) annotation( ...);
  connect(ramp1.y, gain.u) annotation( ...);
  annotation( ...);
end ButterflyValveTest;
```

# シミュレーション結果

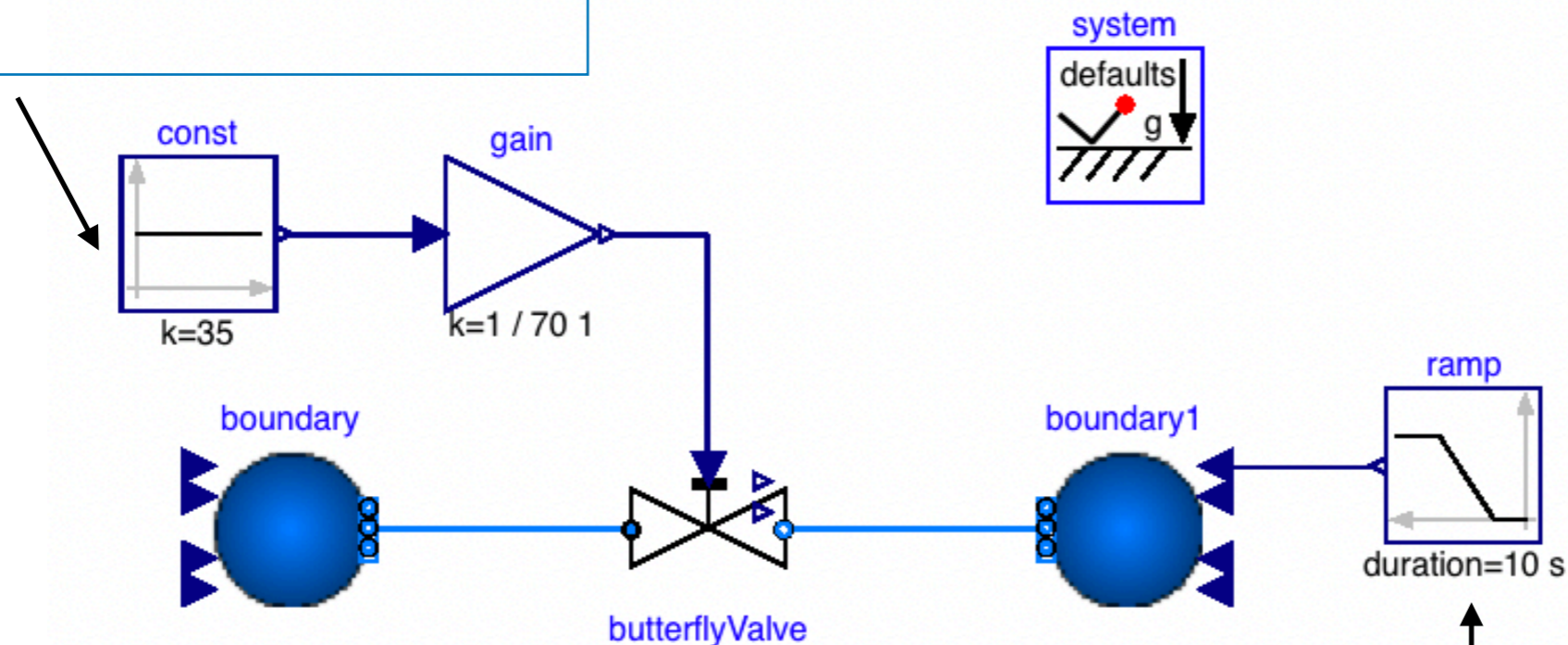


# ButterflyValveTest2

角度を35°に固定して、出口の圧力を変化させる。

**Modelica.Blocks.Sources.Constant**

$k = 35$



**realExpression**

$\text{sqrt}(\text{bu} \dots)$

**Modelica.Blocks.Sources.RealExpression**

$y = \text{sqrt}(\text{butterflyValve.dp})$

**Modelica.Blocks.Sources.Ramp**

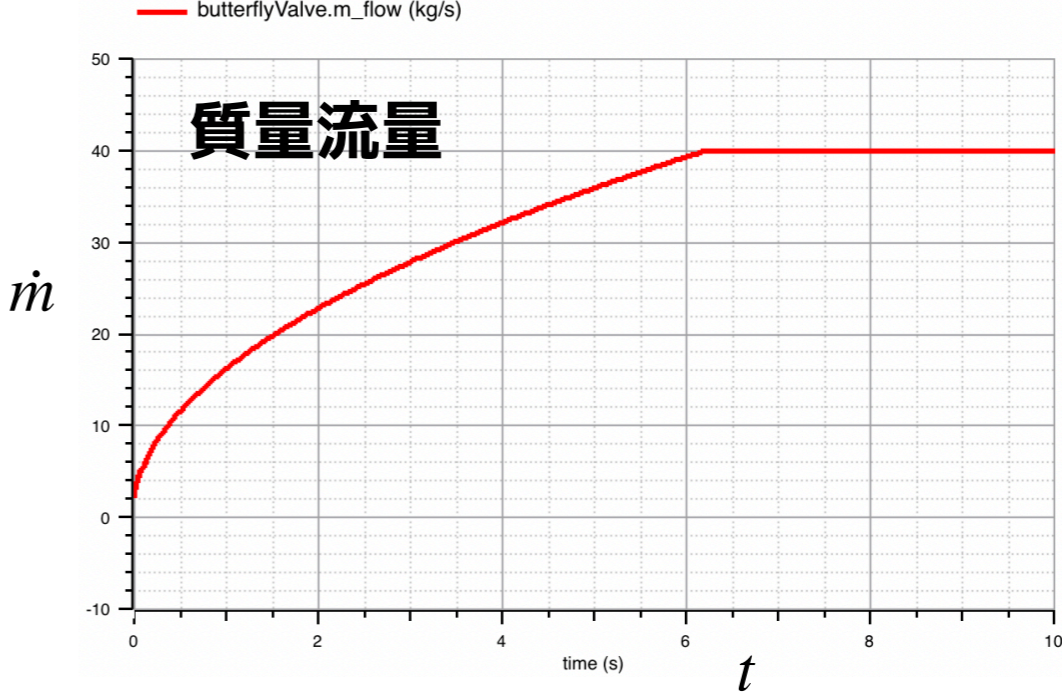
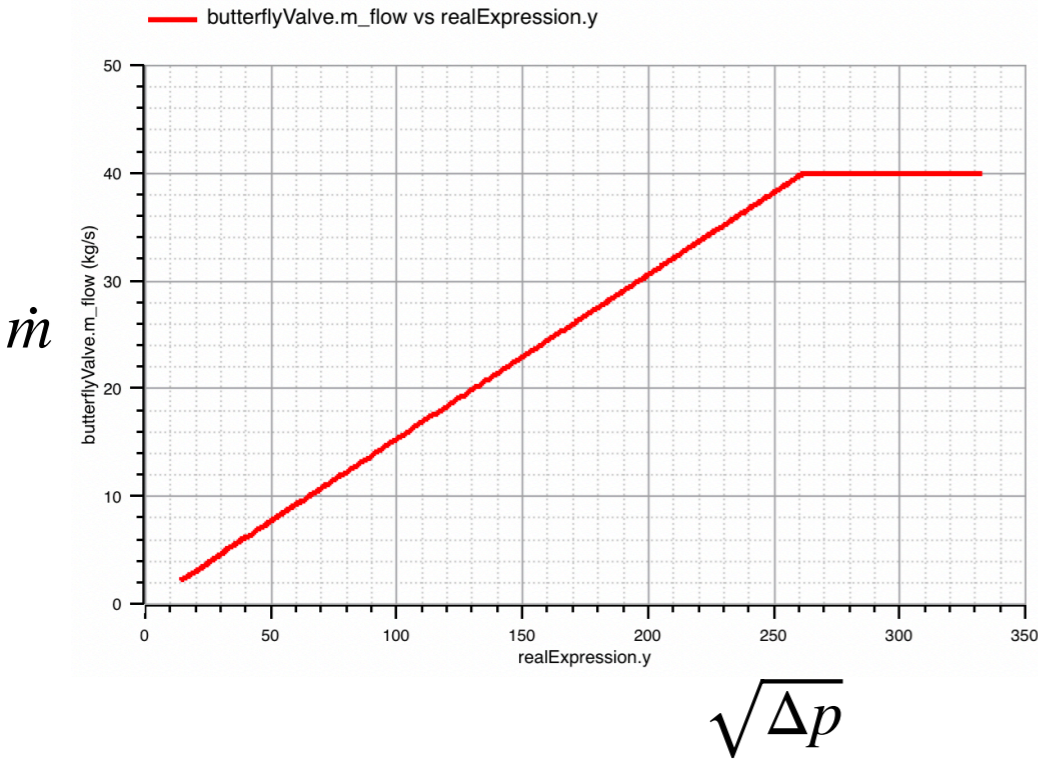
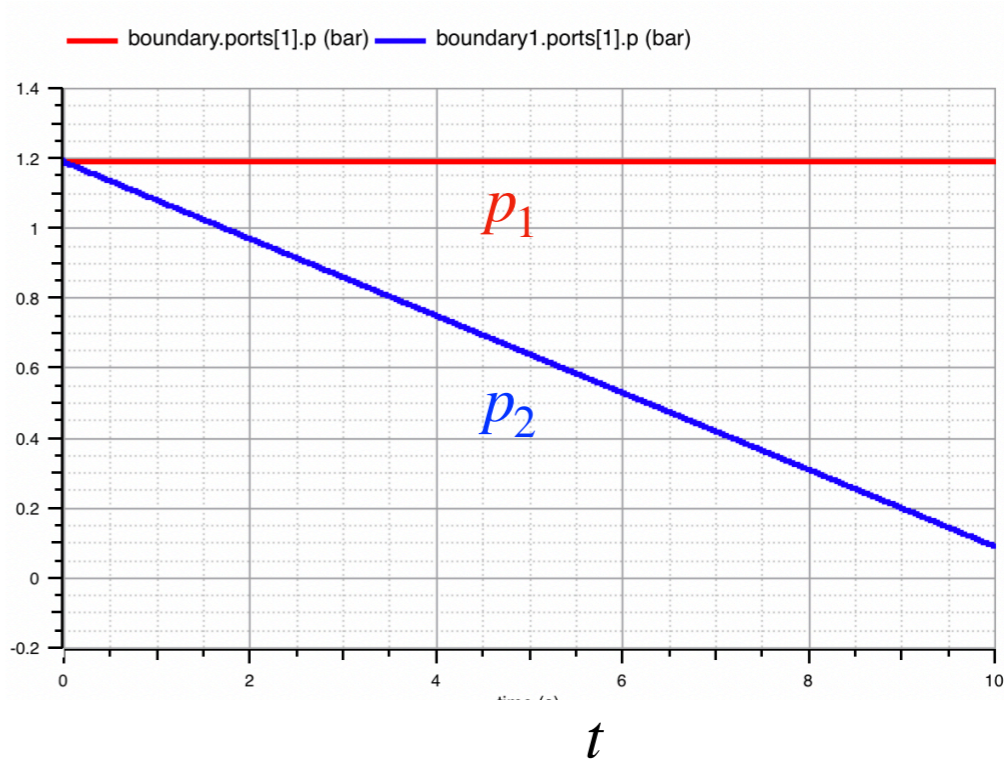
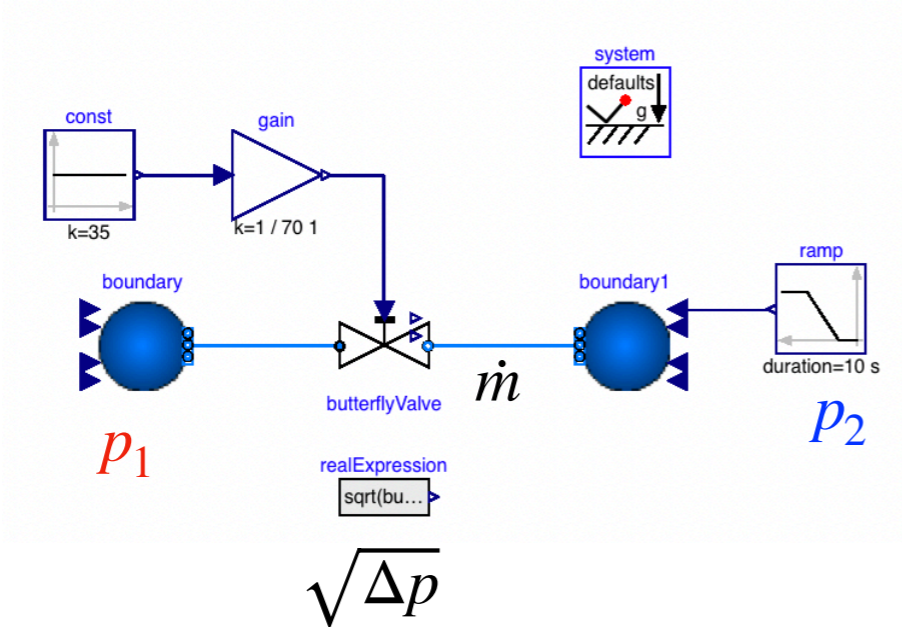
height = -110000

duration = 10 s

offset = 119000

startTime = 0

シミュレーション結果



# WaterValveTest1

## 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	m3/s
密度	998.233	kg/m3	998.233	kg/m3
Av			3.68008E-05	m2
質量流量			0.367682488	kg/s

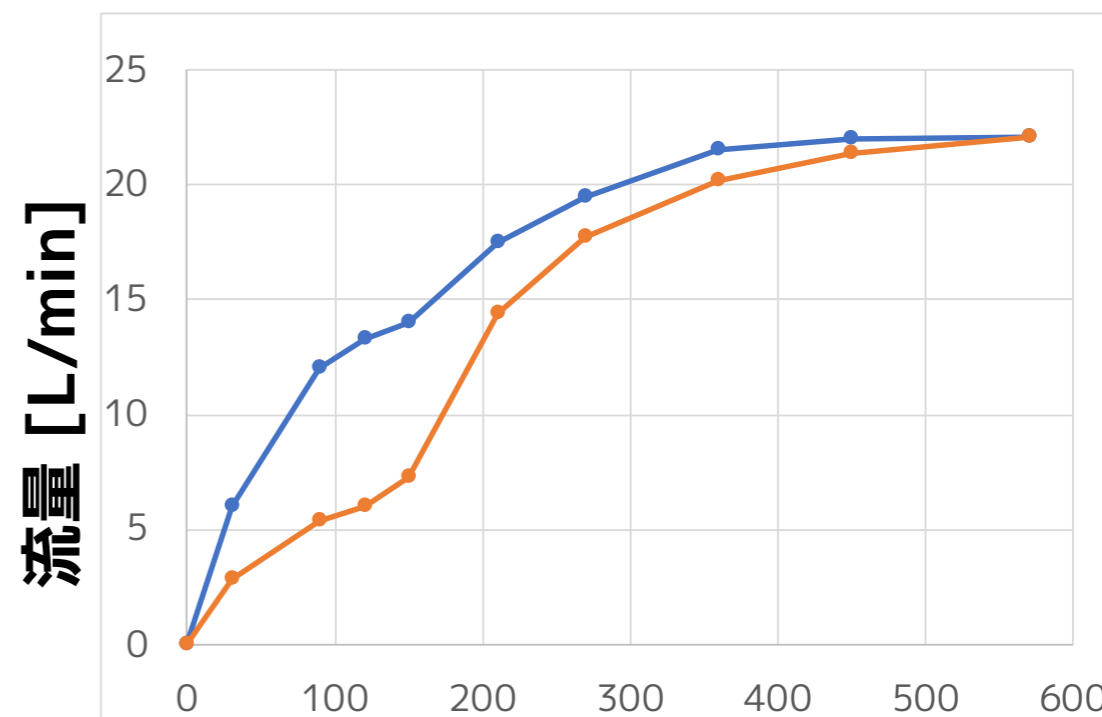
$$A_v = \frac{q}{\sqrt{\frac{\Delta p}{\rho}}}$$

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！

	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

角度や流量の最大値で正規化

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

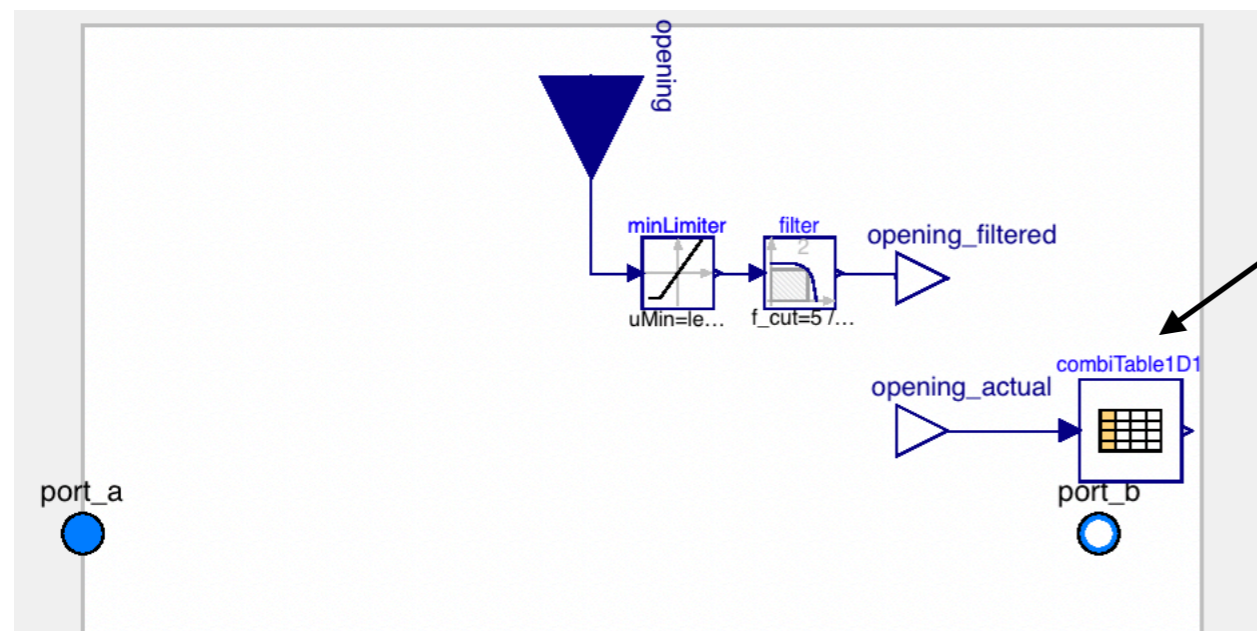


バルブの回転角度 [deg]

## ValveIncompressible の改造

Excel 等で近似曲線を作るのが困難なので  
CombiTable1D を使用して固有流量特性のフィッティングを行う。

### ValveWater1



① ValveIncompressible を複製して  
ValveWater1 を作成し、  
CombiTable1Dを貼り付ける

## ②ValveWater1のソースコードを編集する。

### スコープを調整する。

```
model ValveWater1 "Valve for (almost) incompressible fluids"
  extends Modelica.Fluid.Valves.BaseClasses.PartialValve;
  import Modelica.Fluid.Types.CvTypes;
  import Modelica.Constants.pi;
  import Modelica.Fluid.Utilities;
  import SI = Modelica.SIunits;
  constant SI.ReynoldsNumber Re_turbulent = ...
```

### CombiTable1Dの設定を行う。

```
Modelica.Blocks.Tables.CombiTable1D combiTable1D1(
  smoothness = Modelica.Blocks.Types.Smoothness.ContinuousDerivative,
  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( ...);
```

← 普通コマのデータ

### CombiTable1Dを接続する。

```
equation
  connect(opening_actual, combiTable1D1.u[1]) annotation( ...);
  // m_flow = valveCharacteristic(opening)*Av*sqrt(d)*sqrt(dp);
  // relativeFlowCoefficient = valveCharacteristic(opening_actual);
  relativeFlowCoefficient = combiTable1D1.y[1];
  ...
```

③ValveWatwer1を複製してValveWater2を作成して編集する。

```
Modelica.Blocks.Tables.CombiTable1D combiTable1D1(  
  smoothness = Modelica.Blocks.Types.Smoothness.ContinuousDerivative,  
  table = [  
    0, 0; 0.052631579, 0.126696833;  
    0.157894737, 0.244343891; 0.210526316, 0.271493213;  
    0.263157895, 0.330316742; 0.368421053, 0.65158371;  
    0.473684211, 0.800904977; 0.631578947, 0.914027149;  
    0.789473684, 0.968325792; 1, 1  
  ]  
) annotation( ...);
```

← 節水コマのデータ

## テストモデル

**Modelica.Blocks.Sources.Ramp**  
 height = 569.9 [deg]  
 duration = 10 [s]  
 offset = 0.0  
 startTime = 0

**Modelica.Blocks.Math.Gain**  
 $k = 1/570$

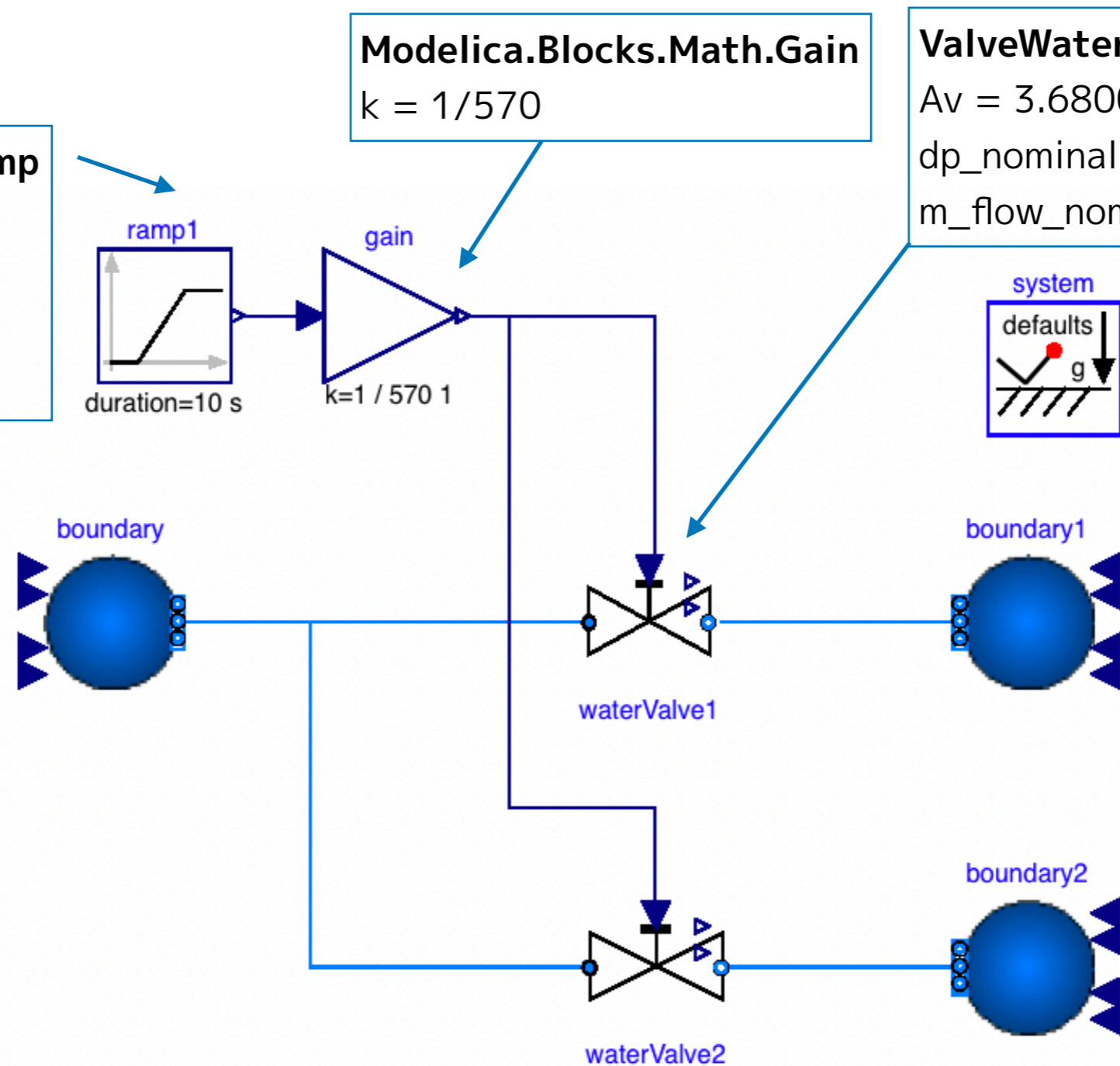
**ValveWater1**  
 $Av = 3.68008e-5$   
 $dp\_nominal = 1$  [bar]  
 $m\_flow\_nominal = 0.03$  [kg/s]

**Boundary\_pT**  
 $p = 201325$  [Pa]  
 $T = 293.15$  [K]

**Boundary\_pT**  
 $p = 101325$  [Pa]  
 $T = 293.15$  [K]

**Boundary\_pT**  
 $p = 101325$  [Pa]  
 $T = 293.15$  [K]

**ValveWater1**  
 $Av = 3.68008e-5$   
 $dp\_nominal = 1$  [bar]  
 $m\_flow\_nominal = 0.03$  [kg/s]

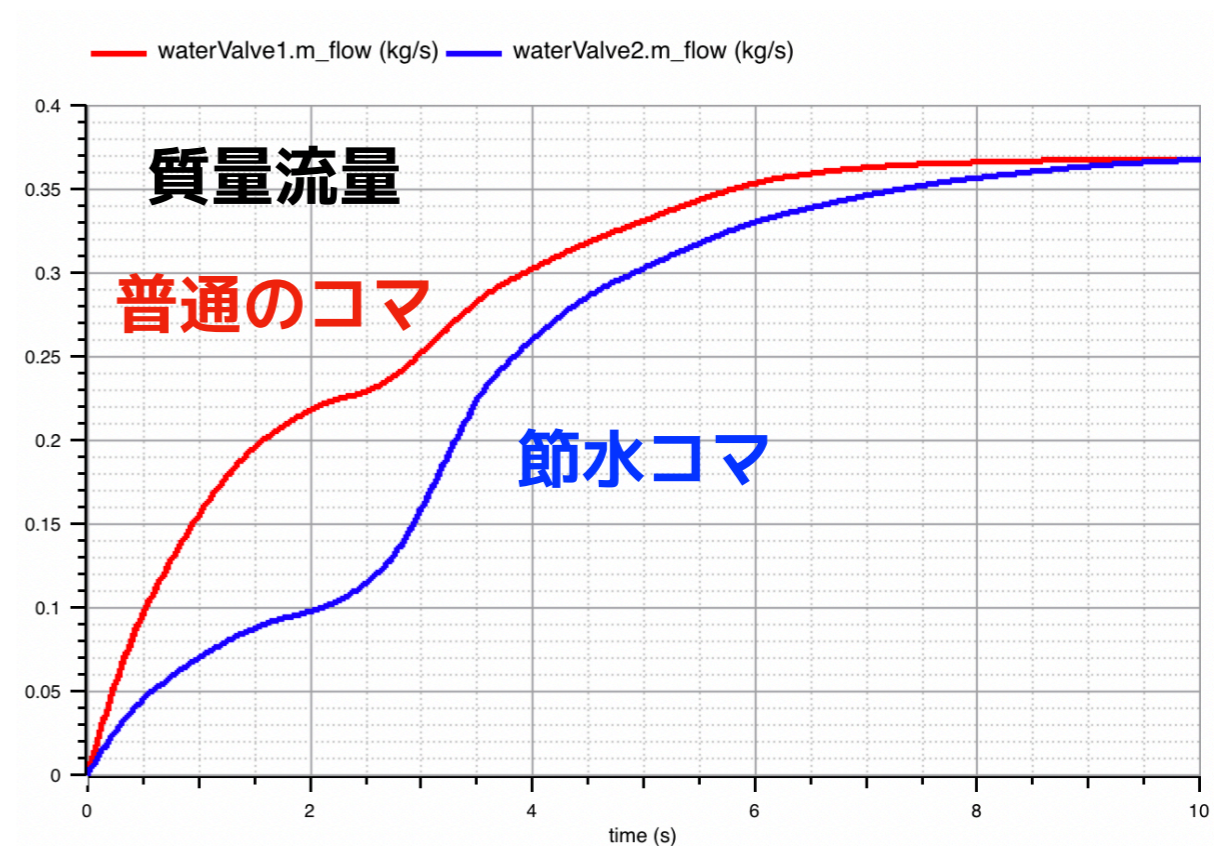
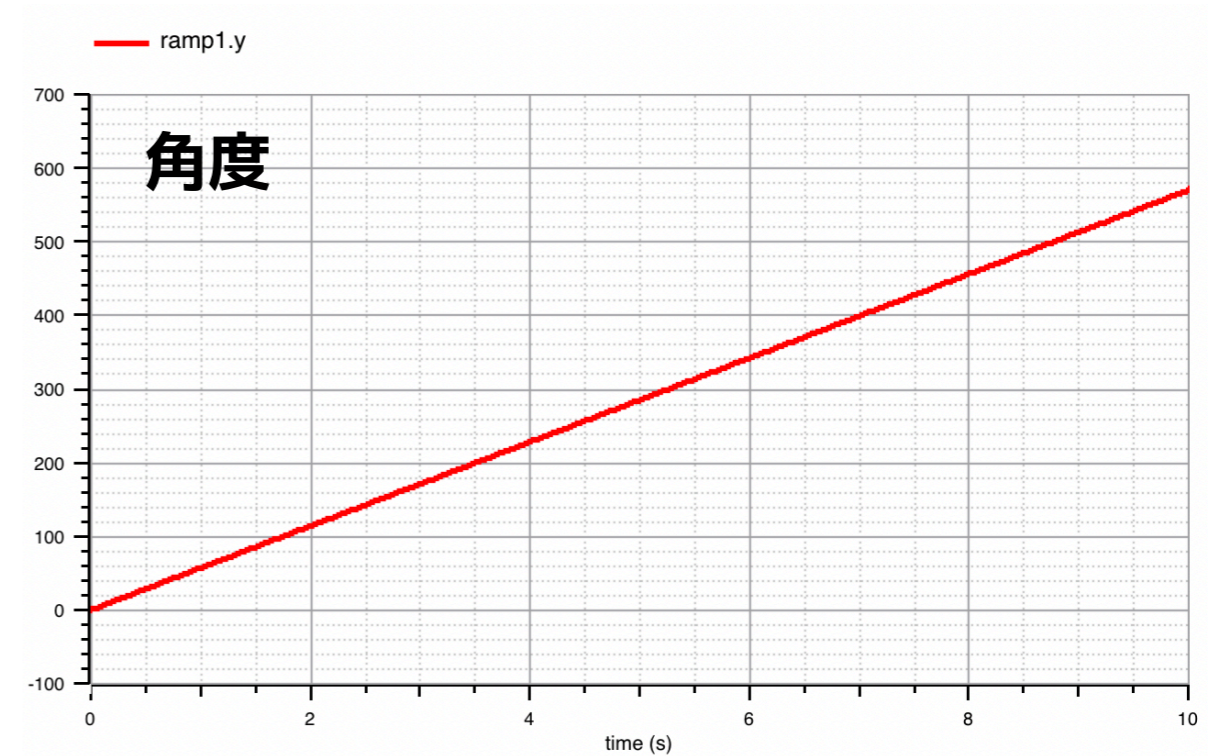
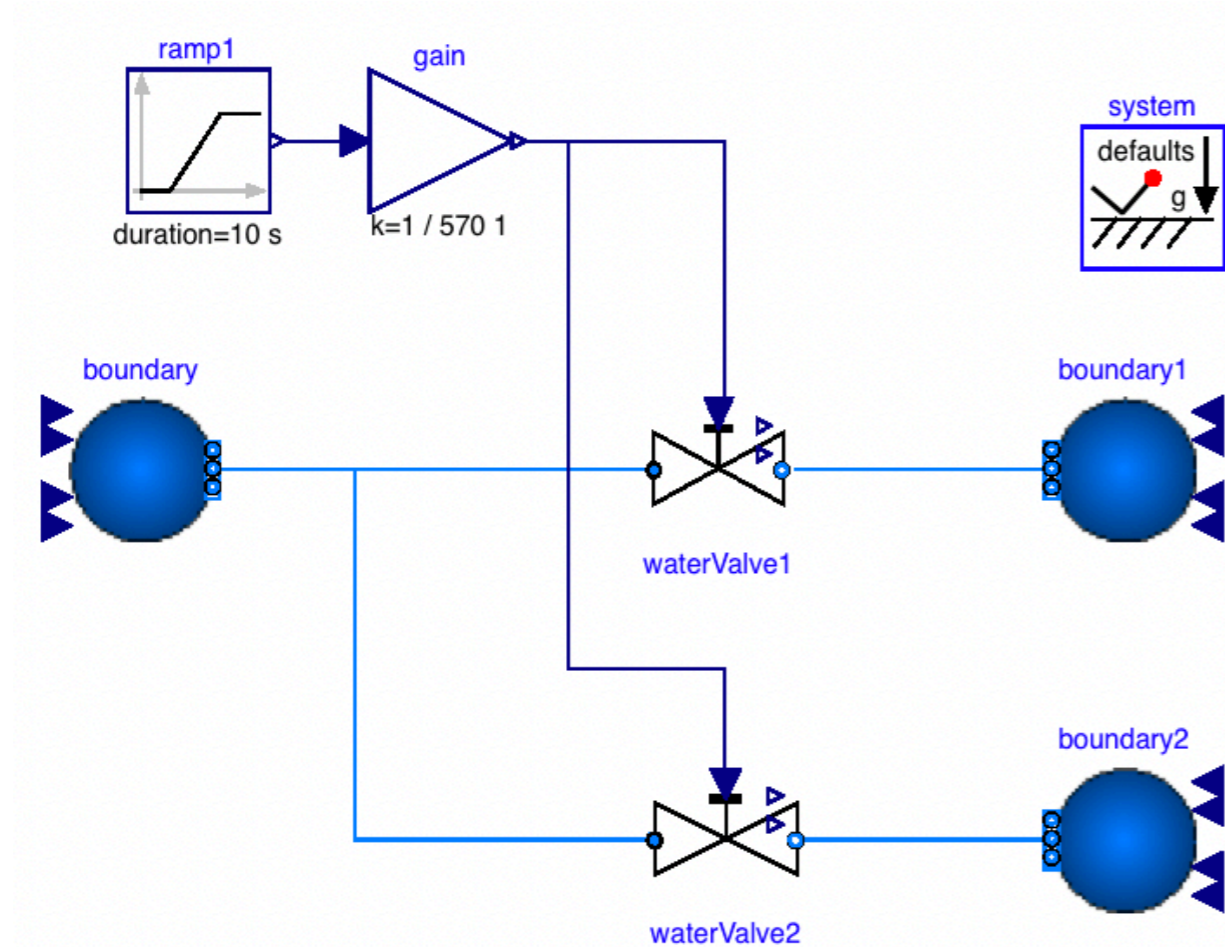


圧力差を固定して、開度を全閉から全開まで変化させる。

## テストモデルのソースコード

```
model WaterValveTest1
  replaceable package Medium = Modelica.Media.Water.StandardWater;
  ValveExample2.ValveWater1 waterValve1(
    redeclare package Medium = Medium, Av = 3.68008e-5,
    CvData = Modelica.Fluid.Types.CvTypes.Av, dp_nominal = 100000, m_flow_nominal = 0.03) annotation( ...);
  Modelica.Fluid.Sources.Boundary_pT boundary(
    redeclare package Medium = Medium, T = 293.15, nPorts = 2, p = 201325) 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 = 569.99, offset = 0.0, startTime = 0) annotation( ...);
  inner Modelica.Fluid.System system annotation( ...);
  ValveWater2 waterValve2(
    redeclare package Medium = Medium, Av = 3.68008e-5,
    CvData = Modelica.Fluid.Types.CvTypes.Av, dp_nominal = 100000, m_flow_nominal = 0.03) annotation( ...);
  Modelica.Fluid.Sources.Boundary_pT boundary2(
    redeclare package Medium = Medium, T = 293.15, nPorts = 1, p = 101325) annotation( ...);
  Modelica.Blocks.Math.Gain gain(k = 1 / 570) annotation( ...);
equation
  connect(boundary.ports[2], waterValve2.port_a) annotation( ...);
  connect(boundary.ports[1], waterValve1.port_a) annotation( ...);
  connect(waterValve2.port_b, boundary2.ports[1]) annotation( ...);
  connect(waterValve1.port_b, boundary1.ports[1]) annotation( ...);
  connect(ramp1.y, gain.u) annotation( ...);
  connect(gain.y, waterValve1.opening) annotation( ...);
  connect(gain.y, waterValve2.opening) annotation( ...);
  annotation( ...);
end WaterValveTest1;
```

# シミュレーション結果



## まとめ

- MSLに含まれる液体用のバルブモデルである ValveIncompressible と ValveVaporizing についてモデルの計算式や使用方法を調査した。
- ValveIncompressible は、実験やバルブメーカーなどから得られる流量係数と固有流量特性を使用してモデル化を行う。
- ValveVaporizing は、液体の流れに加えてバルブ内の液体の圧力が低下して蒸発する流れをモデル化する。そのため、作動流体として2相流体を使用し、流量係数と固有流量特性の他に圧力回復特性の設定が必要となる。

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

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

- The purpose of this document is introducing the valve models of Modelica.Fluid which are included in the Modelica Standard Library (MSL). This document uses libraries, software, figures, and documents included in MSL, and those modifications. Licenses and copyrights of those are written in next page.
- 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>

# Modelica Standard Library License

<https://github.com/modelica/ModelicaStandardLibrary/blob/master/LICENSE>

BSD 3-Clause License

Copyright (c) 1998-2019, Modelica Association and contributors  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.