OpenModelica Course Intermediate

Modelica.Fluid library description

Appendix 1. Pipe pressure loss and mass flow

Appendix 2. Partial Vessel pressure calculation formula

December 7, 2017 Shu Tanaka (Amane Ryuken Co., Ltd.)

Appendix 1. Pipe pressure loss and mass flow

(1) Overview

(2) Parameters that classify the calculation method

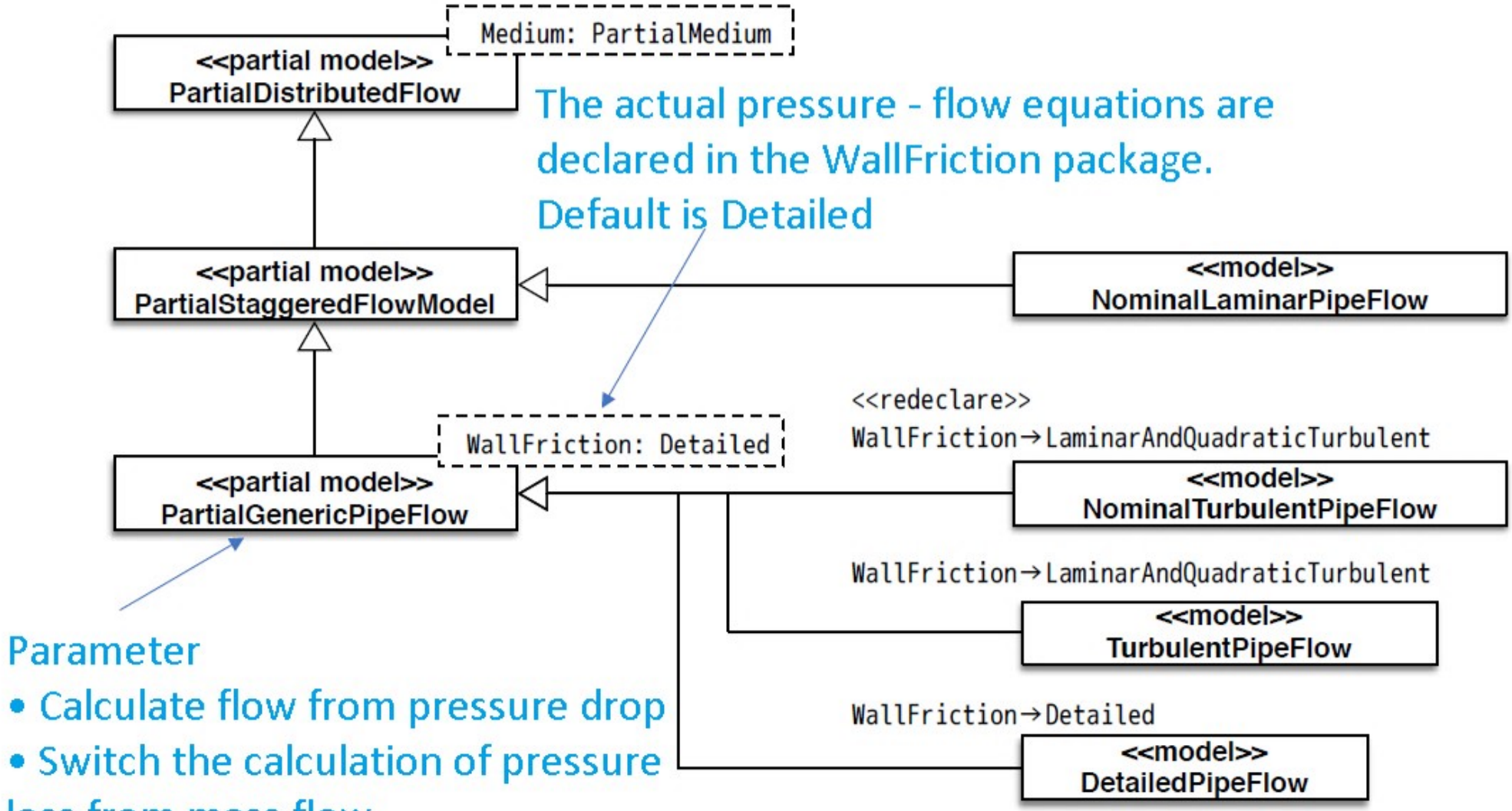
(3) homotopy method and homotopy operator

(4) WallFriction.Detailed

- massFlowRate_dp
- pressureLoss_m_flow

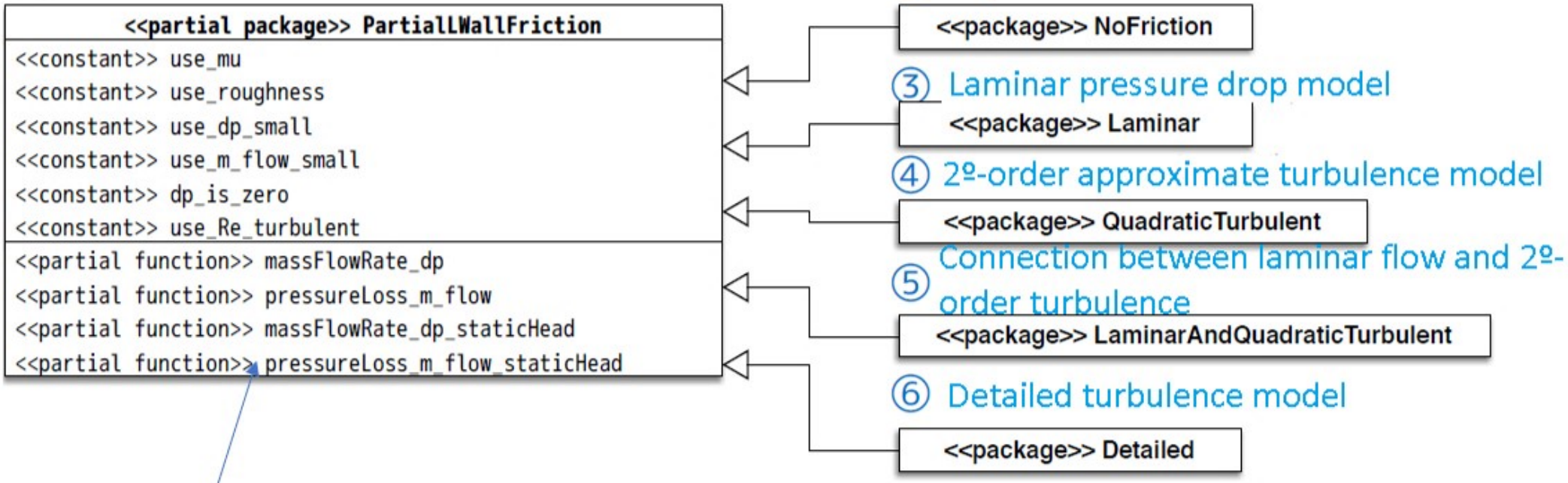
(1) Overview

The relationship between the pressure loss of the pipe and the mass flow rate is calculated by the model that inherits PartialGenericFlow.



WallFriction Package

① Interface



The pressure changes continuously when changing from forward to reverse flow

- massFlowRate_dp: Calculate mass flow from pressure loss
- pressureLoss_m_flow: Calculate the pressure loss from the mass flow rate

If it changes discontinuously or if there is a height difference

- massFlowRate_dp_staticHead: Calculate mass flow from pressure loss
- pressureLoss_m_flow_staticHead: Calculate pressure loss from mass flow

Part of Mass Flow or Pressure Drop Calculation in PartialGenericPipeFlow

```
if continuousFlowReversal then
   // simple regularization
   if from_dp and not WallFriction.dp_is_zero then
     m_flows = homotopy(
       actual=  WallFriction.massFlowRate_dp(
                  dps_fg - {g*dheights[i]*rhos_act[i] for i in 1:n-1},
                  rhos_act,
                  rhos_act,
                  mus_act,
                  mus_act,
                  pathLengths_internal,
                  diameters,
                  (crossAreas[1:n-1]+crossAreas[2:n])/2,
                  (roughnesses[1:n-1]+roughnesses[2:n])/2,
                  dp_small/(n-1),
                  Res_turbulent_internal)*nParallel,
       simplified=  m_flow_nominal/dp_nominal*(dps_fg - g*dheights*rho_nominal));
   else
     dps_fg = homotopy(
       actual=  WallFriction.pressureLoss_m_flow(
                  m_flows/nParallel,
                  rhos_act,
                  rhos_act,
                  mus_act,
                  mus_act,
                  pathLengths_internal,
                  diameters,
                  (crossAreas[1:n-1]+crossAreas[2:n])/2,
                  (roughnesses[1:n-1]+roughnesses[2:n])/2,
                  m_flow_small/nParallel,
                  Res_turbulent_internal) + {g*dheights[i]*rhos_act[i] for i in 1:n-1},
       simplified=  dp_nominal/m_flow_nominal*m_flows + g*dheights*rho_nominal);
   end if;
```

(2) Classification of calculation method using parameters

(3) Homotopy method

```
else
   // regularization for discontinuous flow reversal and static head
   if from_dp and not WallFriction.dp_is_zero then
     m_flows = homotopy(
       actual=  WallFriction.massFlowRate_dp_staticHead(
                  dps_fg,
                  rhos[1:n-1],
                  rhos[2:n],
                  mus[1:n-1],
                  mus[2:n],
                  pathLengths_internal,
                  diameters,
                  g*dheights,
                  (crossAreas[1:n-1]+crossAreas[2:n])/2,
                  (roughnesses[1:n-1]+roughnesses[2:n])/2,
                  dp_small/(n-1),
                  Res_turbulent_internal)*nParallel,
       simplified=  m_flow_nominal/dp_nominal*(dps_fg - g*dheights*rho_nominal));
   else
     dps_fg = homotopy(
       actual=  WallFriction.pressureLoss_m_flow_staticHead(
                  m_flows/nParallel,
                  rhos[1:n-1],
                  rhos[2:n],
                  mus[1:n-1],
                  mus[2:n],
                  pathLengths_internal,
                  diameters,
                  g*dheights,
                  (crossAreas[1:n-1]+crossAreas[2:n])/2,
                  (roughnesses[1:n-1]+roughnesses[2:n])/2,
                  m_flow_small/nParallel,
                  Res_turbulent_internal),
       simplified=  dp_nominal/m_flow_nominal*m_flows + g*dheights*rho_nominal);
   end if;
 end if;
```

(2) Parameters for calculation method

① from_dp

If true, calculate mass flow from pressure difference

if flase, calculate pressure difference from mass flow

Connected by momentumDynamics

Declaration part of PartialGenericPipeFlow

```
 parameter Boolean from_dp = momentumDynamics >= Types.Dynamics.SteadyStateInitial
"= true, use m_flow = f(dp), otherwise dp = f(m_flow)"  annotation( ...);
```

momentumDynamics	from_dp	initial condition
DynamicFreeInitial	false	
FixedInitial	false	m_flow_start
SteadyStateInitial	true	d(m_flow)/dt = 0
SteadyState	true	

② WallFriction.dp_is_zero

True if there is no viscous pressure loss

③ continuousFlowReversal

True if the pressure changes continuously when changing from forward to reverse

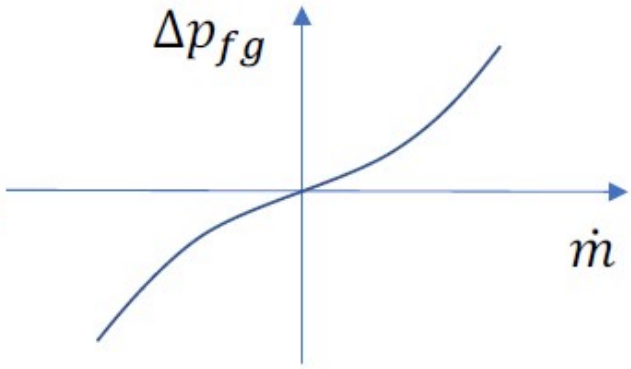
Declaration part of PartialGenericPipeFlow

```
final parameter Boolean continuousFlowReversal=
    (not useUpstreamScheme)
    or constantPressureLossCoefficient
    or not allowFlowReversal
```

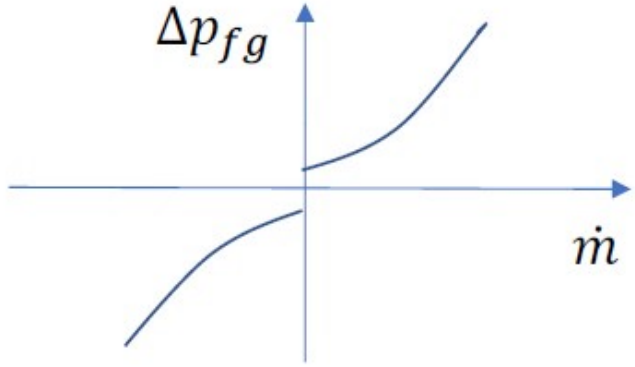
In addition, the value is true when the pressure loss coefficient is not a wind-up difference scheme
or the pressure loss coefficient is constant
or does not consider backflow.

For false, separate cases because a special regularization is required.

true



false



(3) Homotopy method and homotopy operator

Overview

· In the initialization phase of dynamic simulation, you may need to solve a large non-linear equation system with repeated calculations.
· The homotopy method is a method for solving a non-linear equation system by solving a model that simply converges, and continuously deforming the model from the model to the actual model to track the solution.

Homotopy transformation The following expression continuously changes λ from 0 to 1.

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

Modelica homotopy operator

The homotopy operator is used for a variable to be input into a non-linear equation.

Homotopy operator

$$m\_flows = homotpy(actual, simplified)$$

(4) WallFriction.Detailed

Darcy-Weisbach Equation

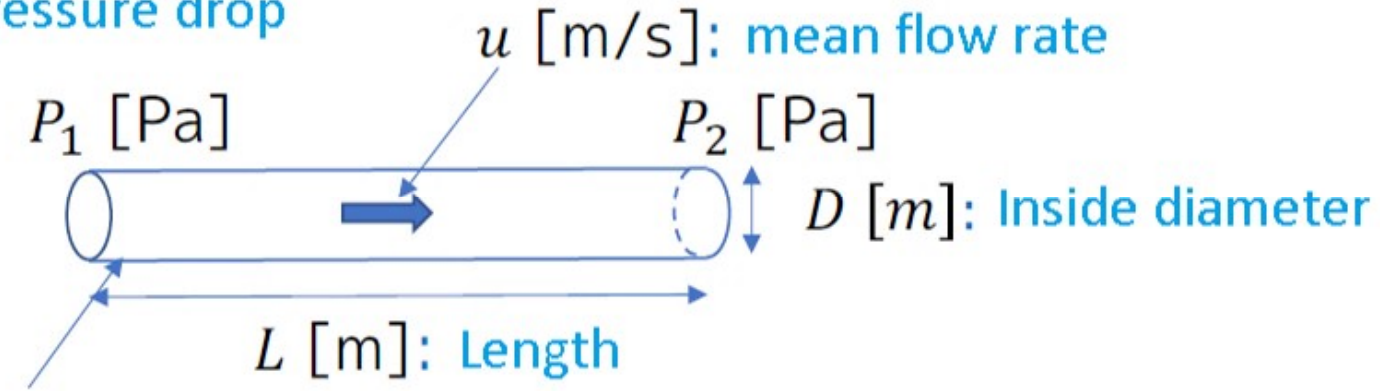
$$\Delta P = \lambda \frac{L}{D} \frac{\rho u^2}{2}$$

λ :Pipe friction coefficient (Darcy's friction factor)

ρ[kg/m3]: Density

μ[Pa.s]: Viscosity

$\Delta P = P_1 - P_2$: Pressure drop



$\Delta = \frac{\varepsilon}{D}$: Relative roughness

$Re = \frac{\rho u D}{\mu}$: Reynolds number

λ : Darcy's friction factor

$Re < Re_1$ Laminar flow

Hagen-Poiseuille flow

$$\lambda = \frac{64}{Re} \qquad u = \frac{\Delta P}{L}\frac{r^2}{8\mu}$$

$Re > Re_2$ Turbulent flow

Colebrook–White Equation

$$\frac{1}{\sqrt{\lambda}} = -2\log_{10}\left(\frac{2.51}{Re\sqrt{\lambda}} + \frac{\varepsilon/D}{3.7}\right) = -2\log_{10}\left(\frac{2.51}{Re\sqrt{\lambda}} + 0.27\Delta\right)$$

$Re_1 < Re < Re_2$ The range of transition regions between turbulent and laminar flow

$$Re_1 = 745\exp\left(if\ \Delta < 0.0065\ then\ 1\ else\ \frac{0.0065}{\Delta}\right)$$

[Samoilenko 1968; Idelchik 1994, p. 81, sect. 2.1.21]

$$Re_2 = Re_{turbulent} = 4000$$

$\lambda_2 \equiv \lambda Re^2$ Introduction

For laminar flow

$$\lambda = \frac{64}{Re} \Leftrightarrow Re = \frac{\lambda_2}{64} \quad \text{or} \quad \lambda_2 = 64\,Re$$

$Re \to 0$ So λ is diverging, but λ_2 is not

In the case of turbulence

$$\frac{1}{\sqrt{\lambda}} = -2\log_{10}\left(\frac{2.51}{Re\sqrt{\lambda}} + 0.27\Delta\right) \Leftrightarrow Re = -2\sqrt{\lambda_2}\log_{10}\left(\frac{2.51}{\sqrt{\lambda_2}} + 0.27\Delta\right)$$

The Colebrook-White expression is no longer a implicit function.

Relation between pressure and lambda

$$\Delta P = \lambda\frac{L}{D}\frac{\rho u^2}{2}, \quad Re = \frac{\rho u D}{\mu} \Leftrightarrow \lambda_2 = |\Delta p|\frac{2D^3\rho}{L\mu^2} \quad \text{または} \quad \Delta p = \pm\frac{L\mu^2}{2D^3\rho}\lambda_2$$

Relationship between Mass flow rate and Reynolds number

$$Re = \frac{\rho u D}{\mu}, \quad \dot{m} = \rho u A \quad \Leftrightarrow \dot{m} = \frac{\mu A}{D}Re \quad \text{or} \quad Re = \frac{D}{\mu A}|\dot{m}|$$

massFlowRate_dp Determine mass flow from pressure difference

① $\lambda_2 = |\Delta p| \dfrac{2D^3\rho}{L\mu^2}$

② $Re = \dfrac{\lambda_2}{64}$ First, the Reynolds number is obtained by assuming the laminar flow.

if $Re > Re_1$

$$Re = -2\sqrt{\lambda_2}\log_{10}\left(\frac{2.51}{\sqrt{\lambda_2}} + 0.27\Delta\right)$$

If it is outside the range of laminar flow, the Reynolds number is calculated assuming turbulent flow.

if $Re < Re_2$

In the transition region, the Reynolds number is calculated by an interpolation function.

$$Re = interplateRegion2(Re, Re_1, Re_2, \Delta, \lambda_2)$$

③ $\dot{m} = \pm\dfrac{\mu A}{D}Re$ A mass flow rate is calculated from the Reynolds number. The sign matches Δ%.

pressureLoss_m_flow Calculate pressure difference from mass flow rate

① $Re = \frac{D}{\mu A}|\dot{m}|$

② if $Re \leq Re_1$

$$\lambda_2 = 64\,Re$$

if $Re \geq Re_2$

$$\lambda_2 = 0.25\left(\frac{Re}{\log\left\{\frac{\Delta}{3.7} + \frac{5.74}{Re^{0.9}}\right\}}\right)^2$$

(Swamee-Jain equation)

(Approximate formula of Colebrook-White's formula)

$$\lambda = 0.25\left[\log\left\{\frac{\Delta}{3.7} + \frac{5.74}{Re^{0.9}}\right\}\right]^{-2}$$

$Re_1 < Re < Re_2$ Interpolation function for transition region

$$\lambda_2 = interplateInRegion2(Re, Re1, Re2, \Delta)$$

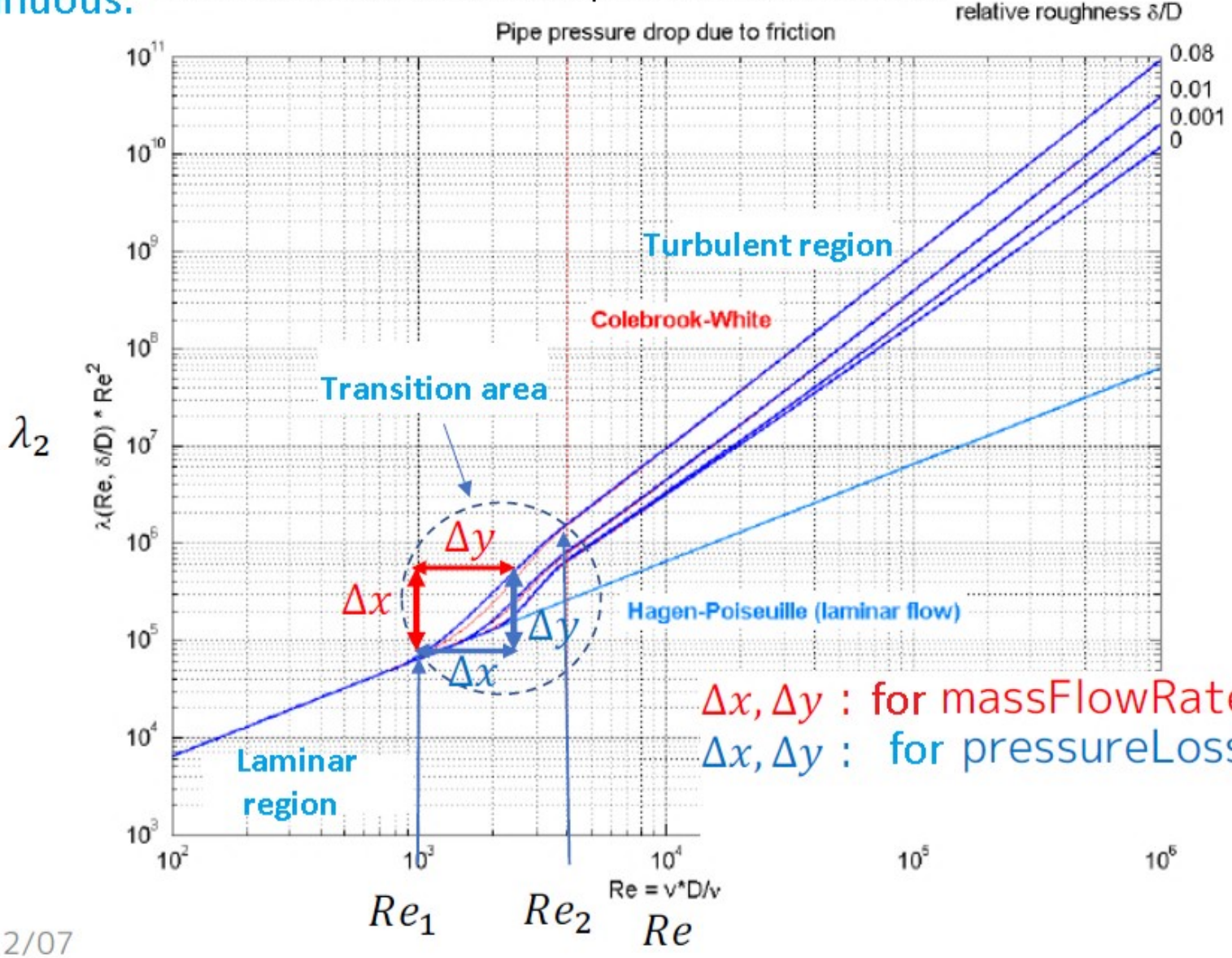
③ $\Delta p = \pm\frac{L\mu^2}{2D^3\rho}\lambda_2$ Calculate the pressure difference from λ_2

The sign matches $\dot{m}$

Concept of interpolation method for transition region

Approximate Δy with a cubic polynomial of Δx using the distances Δx and Δy on the logarithmic graph, and connect to the curves of the laminar flow region and the turbulent flow region with Re_1 and Re_2, respectively, until the 1st derivative is continuous.

From Modelica.Fluid.UsersGuid.ComponentDefinition.WallFrincion



λ_2

Re_1 Re_2 Re

$\Delta x, \Delta y$: for massFlowRate_dp

$\Delta x, \Delta y$: for pressureLoss_m_flow

Appendix 2. PartialLumpedVessel pressure equation

(1) General pressure loss coefficient and pressure flow rate relational expression

(2) Effect of correction penetration when the liquid level is close to the port height

(3) Regularization when flow rate is small (Proper flow)

(4) Calculation of nonlinear equations by homotopy method

(1) General pressure loss coefficient and pressure flow rate relational expression

Inflow

$$p_{in} + \frac{\dot{m}^2}{2\rho_{in}A_{in}^2} - \zeta_{in} \cdot \frac{\dot{m}^2}{2\rho_{in}A_{in}^2} = p_{vessel} + \frac{\dot{m}^2}{2\rho_{vessel}A_{vessel}^2}$$

Static pressure / Dynamic pressure / Pressure loss / Static pressure / Dynamic pressure

$$\Rightarrow \quad p_{in} = p_{vessel} + \left(\zeta_{in} - 1 + \frac{A_{in}^2}{A_{vessel}^2}\right)\frac{1}{2\rho_{in}A_{in}^2} \cdot \dot{m}^2$$

Outflow

$$p_{vessel} + \frac{\dot{m}^2}{2\rho_{vessel}A_{vessel}^2} - \zeta_{out} \cdot \frac{\dot{m}^2}{2\rho_{out}A_{out}^2} = p_{out} + \frac{\dot{m}^2}{2\rho_{out}A_{out}^2}$$

Static pressure / Dynamic pressure / Pressure loss / Static pressure / Dynamic pressure

$$\Rightarrow \quad p_{out} = p_{vessel} - \left(\zeta_{out} + 1 - \frac{A_{out}^2}{A_{vessel}^2}\right)\frac{1}{2\rho_{vessel}A_{out}^2}\dot{m}^2$$

Dynamic pressure $\frac{1}{2}\rho u^2 = \frac{(\rho u A)^2}{2\rho A^2} = \frac{\dot{m}^2}{2\rho A^2}$ (Kinetic energy per unit volume)

(2) Correction when the liquid level is close to the port height

When entering the container $(\dot{m}_{port} > \dot{m}_{turbulent})$ Port static pressure

$$p_{in} = p_{vessel_i} + \frac{1}{2A_i^2}\underbrace{\left(\zeta_{in_i} - 1 + \frac{A_i^2}{A_{vessel}^2}\right)\frac{1}{\rho_{in_i}}penetration_i}_{k_1} \cdot m_i^2$$

When spilling from the container $(\dot{m}_{port} < -\dot{m}_{turbulent})$ Port static pressure

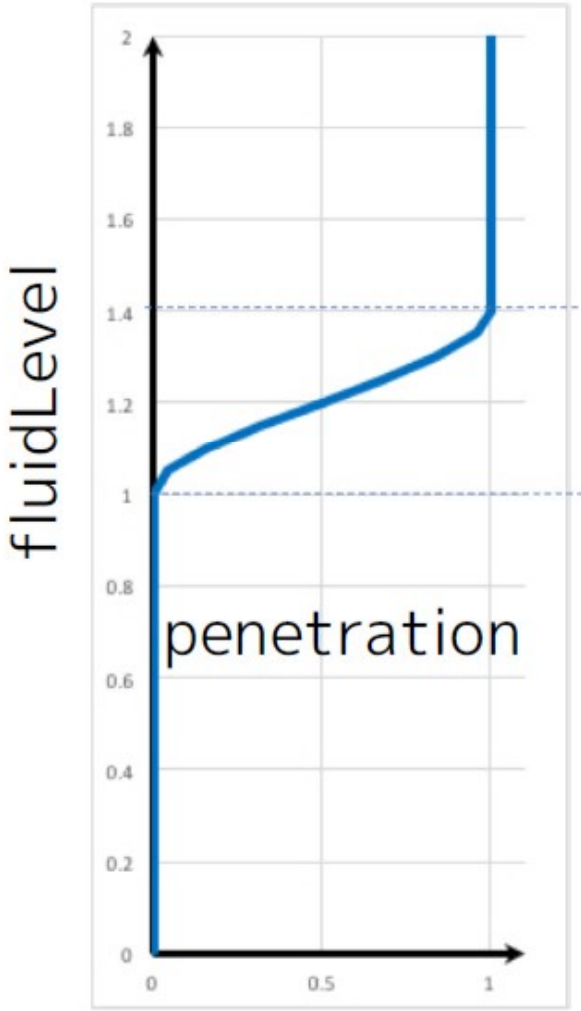
$$p_{out} = p_{vessel_i} - \frac{1}{2A_i^2}\underbrace{\left(\zeta_{out_i} + 1 - \frac{A_i^2}{A_{vessel}^2}\right)\frac{1}{\rho_{vessel}}\frac{1}{penetration_i}}_{k_2} \cdot \dot{m}_i{}^2$$

When the liquid level is below the port height + 0.2D, $penetration_i < 1$
The pressure loss at the time of inflow is small, and the pressure loss at the time of outflow is large.

Formula for penetration

$$penetration_i = \boldsymbol{regStep}(fluidLevel - height_i - 0.1D_i, 1, 1\times10^{-3}, 0.1D_i)$$

```
ports_penetration[i]
= Utilities.regStep(fluidLevel - portsData_height[i] - 0.1*portsData_diameter[i],
   1, 1e-3, 0.1*portsData_diameter[i]);
```



penetration=1.0

Port height + 0.2D

Port height

penetration=0.001

$penetration_i = 1$ when use_portsData = false

Modelica.Fluid.Utilities.regStep

A function that approximates a step function in a continuous differentiable line

$$y = \begin{cases} y1, & x > 0 \\ y2, & x \le 0 \end{cases} \Rightarrow y = \begin{cases} y1, & x > x_{small} \\ y2, & x < -x_{small} \\ f(y1, y2) & -x_{small} \le x \le x_{small} \end{cases}$$

The area of $-x_{small} < x < x_{small}$ is a quadratic that changes from y1 to y2. Approximate with polynomial f (x)

```
y := smooth(1, if x >  x_small then y1 else
               if x < -x_small then y2 else
               if x_small > 0 then
                 (x/x_small)*((x/x_small)^2 - 3)*(y2-y1)/4 + (y1+y2)/2
               else
                (y1+y2)/2);
```

(3) Regularization when flow rate is small

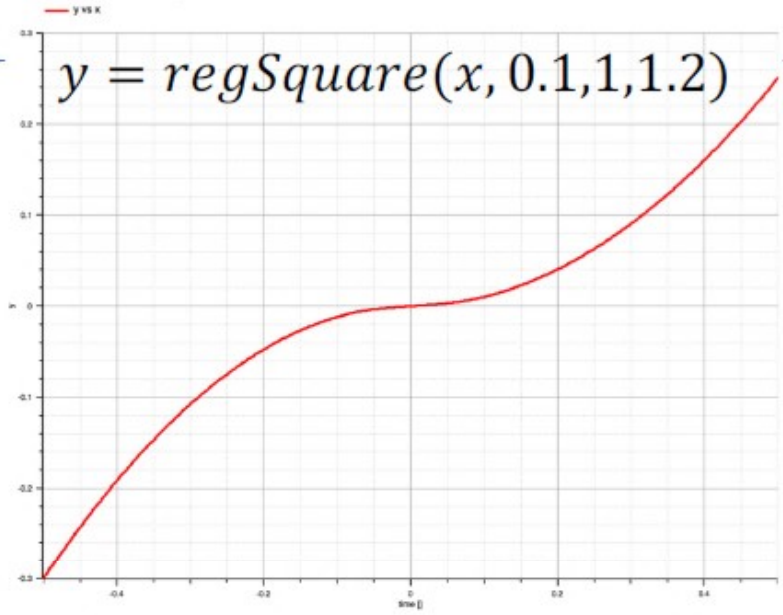
$$p_{port} = p_{vessel_i} + \frac{1}{2A_i}\text{regSqure2}(\dot{m}, \dot{m}_{turbulent}, k_1, k_2)$$

Connect p_{in} and p_{out} smoothly when $|\dot{m}_i| < \dot{m}_{turbulent_i}$

Modelica.Fluid.Utilities.regSquare2

$$y = \text{regSquare}(x, x_{small}, k_1, k_2, use_{yd0}, yd0)$$

$$y = \begin{cases} k_1 x^2, & x \geq 0, k_1 > 0 \\ -k_2 x^2, & x < 0, k_2 > 0 \end{cases}$$



Smoothly connect with $-x_{small} \leq x \leq x_{small}$

- Approximate $-x_{small} \leq x \leq 0$ and $0 \leq x \leq x_{small}$ with separate cubic polynomials.
- The derivative is not zero at $x = 0$. (The reciprocal does not become infinity.)
- The first derivative is continuous in all areas.
- If $use\_yd0 = \text{false}$ (default), the second derivative of two cubic polynomials will match at $x = 0$
- If $use\_yd0 = \text{true}$, specify the first-order fine $yd0$ at $x = 0$ as an argument.

(4) Computation of nonlinear equations by homotopy

The homotopy operator is used because the relationship between flow rate and pressure loss is non-linear.

actual model

$$p_{port} = p_{vessel} + \frac{1}{2A^2}\mathrm{regSqure2}(\dot{m}, \dot{m}_{turbulent}, k_1, k_2)$$

simplified model

$$p_{port} = p_{vessel}$$

homotopy model

Homotopy operator

$$p_{port} = homotpy\left(p_{vessel} + \frac{1}{2A^2} regSqure2(\dot{m}, \dot{m}_{turbulent}, k_1, k_2), p_{vessel}\right)$$

PartialVessel pressure calculation formula

```
// fluid flow through ports
   regularFlow[i] = fluidLevel >= portsData_height[i];
   inFlow[i]      = not regularFlow[i] and (s[i] > 0 or portsData_height[i] >= fluidLevel_max);
   if regularFlow[i] then
     // regular operation: fluidLevel is above ports[i]
     // Note: >= covers default values of zero as well
     if use_portsData then
       /* Without regularization
          ports[i].p = vessel_ps_static[i] + 0.5*ports[i].m_flow^2/portAreas[i]^2
                       * noEvent(if ports[i].m_flow>0 then zeta_in[i]/portInDensities[i] else -zeta_out[i]/medium.d);
       */

       ports[i].p = homotopy(vessel_ps_static[i] + (0.5/portAreas[i]^2*Utilities.regSquare2(ports[i].m_flow, m_flow_turbulent[i],
                                (portsData_zeta_in[i] - 1 + portAreas[i]^2/vesselArea^2)/portInDensities[i]*ports_penetration[i],
                                (portsData_zeta_out[i] + 1 - portAreas[i]^2/vesselArea^2)/medium.d/ports_penetration[i])),
                             vessel_ps_static[i]);
       /*
         // alternative formulation m_flow=f(dp); not allowing the ideal portsData_zeta_in[i]=1 though
         ports[i].m_flow = smooth(2, portAreas[i]*Utilities.regRoot2(ports[i].p - vessel_ps_static[i], dp_small,
                                  2*portInDensities[i]/portsData_zeta_in[i],
                                  2*medium.d/portsData_zeta_out[i]));
       */
     else
       ports[i].p = vessel_ps_static[i];
     end if;
     s[i] = fluidLevel - portsData_height[i];
```

Homotopy operator

regularization

k_1

k_2

penetration

- The purpose of this document is introducing Media and Fluid Libraries 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.
- Copyright and License of this document are written in the last page.

Modelica Standard Library License

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