Numerical Simulation of Pressure Surge with the Method of Characteristics

R. Fiereder

02.04.2009 Saint Petersburg, Russia

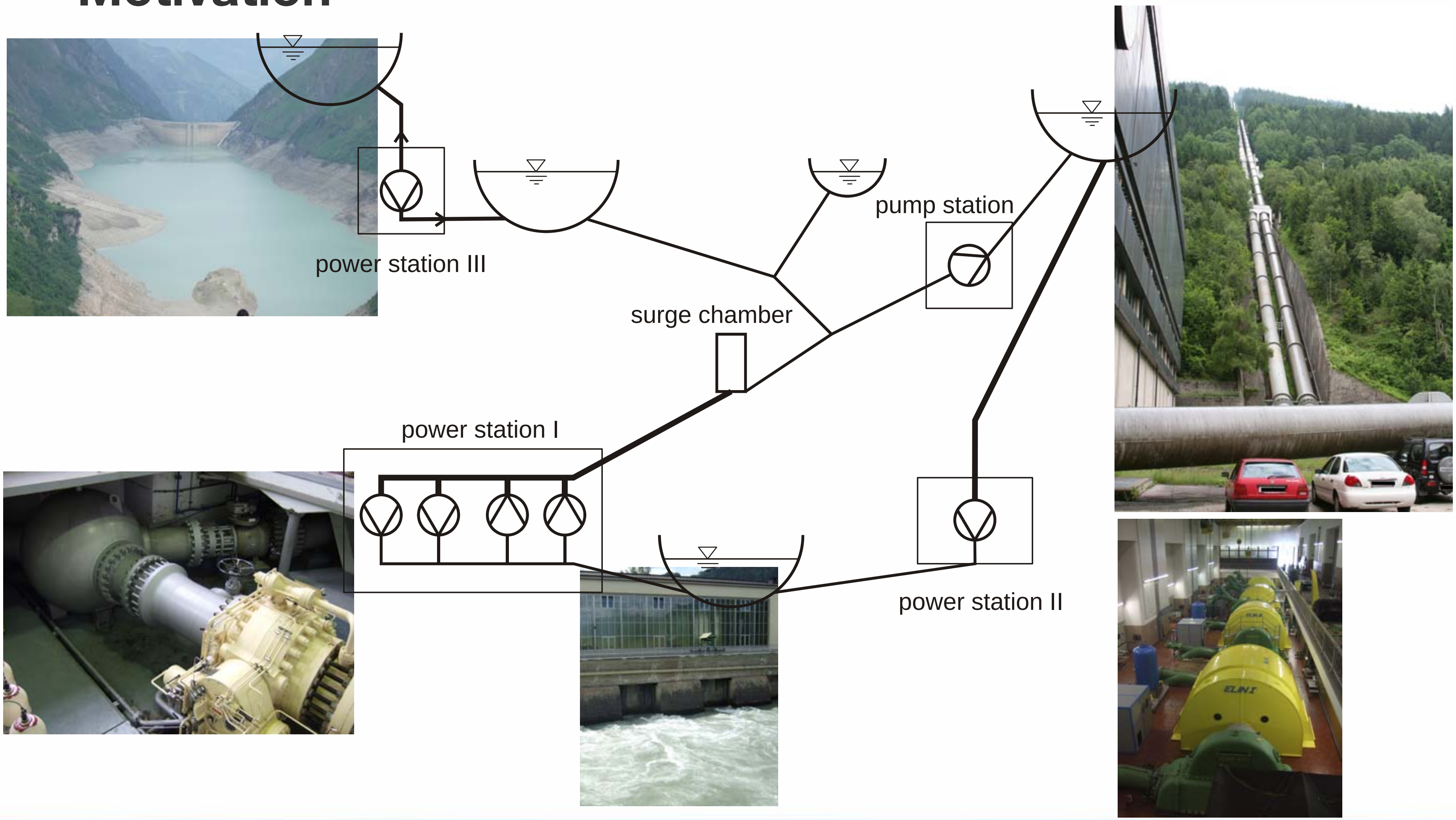
Content

- Motivation
- Governing Equations
  - Continuity Equation
  - Momentum Equation
- Method of Characteristics
- Implementation
- Boundary Conditions
- Application
  - pressure surge
  - oscillating valve
  - surge chamber oscillations
  - power plant simulation

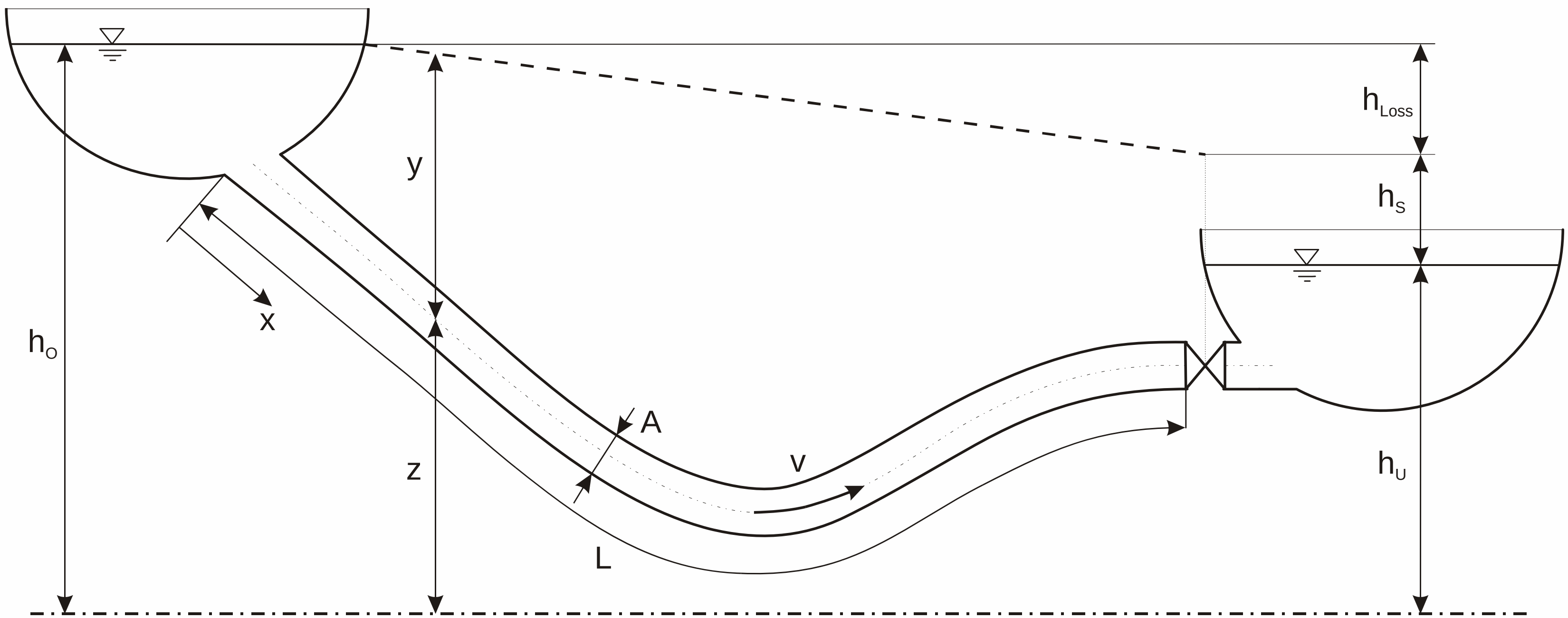
Motivation

- Simulation of interaction of plant components
- Identification of resonance cases

Motivation


power station III
surge chamber
pump station
power station I
power station II

Governing Equations



- 1D
- unsteady
- compressible

⇒

- Continuity equation
- Momentum equation

Continuity Equation

$$\frac{1}{\rho}\frac{d\rho}{dt}+\frac{1}{A}\frac{dA}{dt}+\frac{\partial v}{\partial x}=0$$

- Continuity

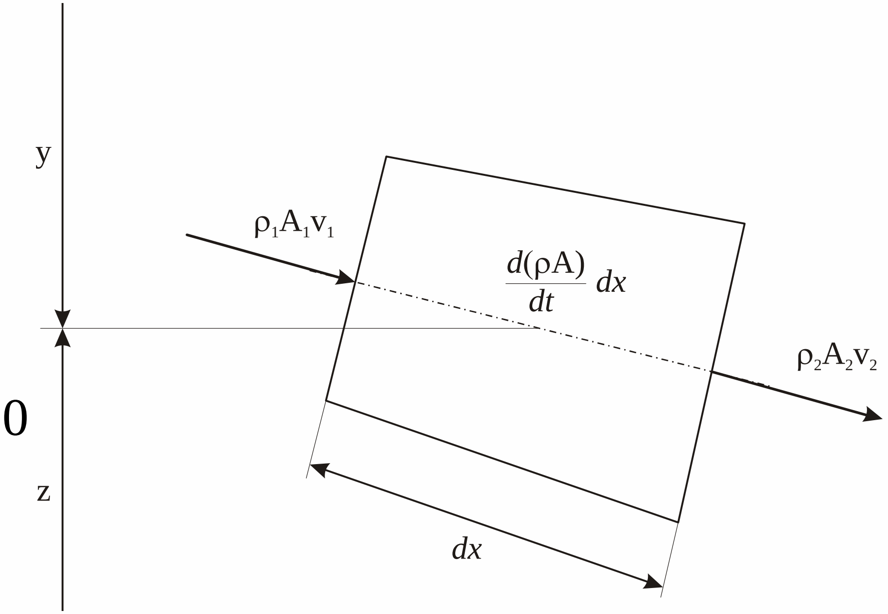
$$\frac{\partial(\rho A)}{\partial t}+\frac{\partial(\rho v A)}{\partial x}=0$$

- Product rule

$$A\frac{\partial \rho}{\partial t}+\rho\frac{\partial A}{\partial t}+Av\frac{\partial \rho}{\partial x}+\rho v\frac{\partial A}{\partial x}+\rho A\frac{\partial v}{\partial x}=0$$

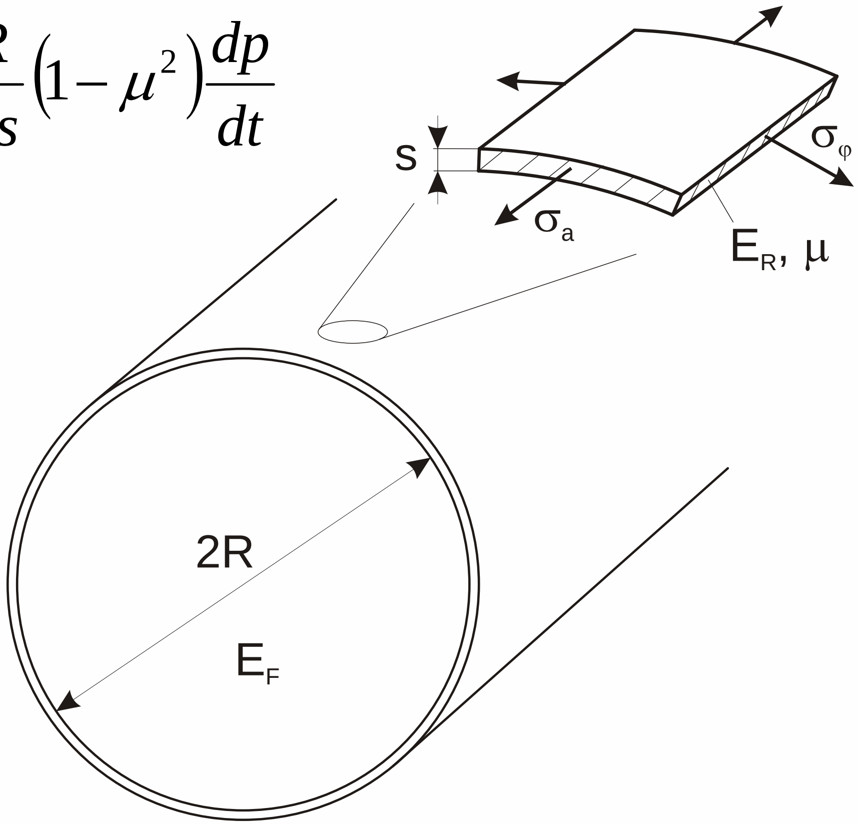
- Total differential

$$\frac{d}{dt}=\frac{\partial}{\partial t}+\frac{dx}{dt}\frac{\partial}{\partial x}$$



Continuity Equation

$$\frac{1}{A}\frac{dA}{dt} = \frac{2R}{E_R s}\left(1-\mu^2\right)\frac{dp}{dt}$$



- Circular Pipe

$$\frac{dA}{dt} = 2\pi R\frac{dR}{dt} \qquad \frac{1}{A}\frac{dA}{dt} = 2\frac{d\varepsilon_\varphi}{dt}$$

- No axial displacement / Hooke‘s law

$$\sigma_a = \mu\sigma_\varphi \qquad \varepsilon_\varphi = \frac{\sigma_\varphi - \mu\sigma_a}{E_R}$$

- s << R

$$\frac{d\sigma_\varphi}{dt} \cong \frac{R}{s}\frac{dp}{dt}$$

Continuity Equation

$$\frac{1}{\rho}\frac{d\rho}{dt} = \frac{1}{E_F}\frac{dp}{dt}$$

- Fluid compressibility
  - Hooke's law applies

$$E_F = \frac{dp}{dV/V}$$

  - Mass in Control Volume is constant

$$dm = Vd\rho + \rho dV = 0$$

$$\frac{dV}{V} = -\frac{d\rho}{\rho}$$

Continuity Equation

$$\frac{\partial p}{\partial t} + v\frac{\partial p}{\partial x} + \rho a^2 \frac{\partial v}{\partial x} = 0$$

$$\frac{dp}{dt}\underbrace{\left[\frac{1}{E_F} + \frac{2R}{E_R s}\left(1-\mu^2\right)\right]}_{\frac{1}{E_{Sys}}} + \frac{\partial v}{\partial x} = 0$$

$$a^2 = \frac{E_{Sys}}{\rho}$$

Momentum Equation

$$\frac{1}{\rho}\frac{\partial p}{\partial x} + g\frac{\partial z}{\partial x} + gJ_e + v\frac{\partial v}{\partial x} + \frac{\partial v}{\partial t} = 0$$

- Momentum conservation

$$\sum_i F_i = \frac{dI}{dt}$$

- Pressure force

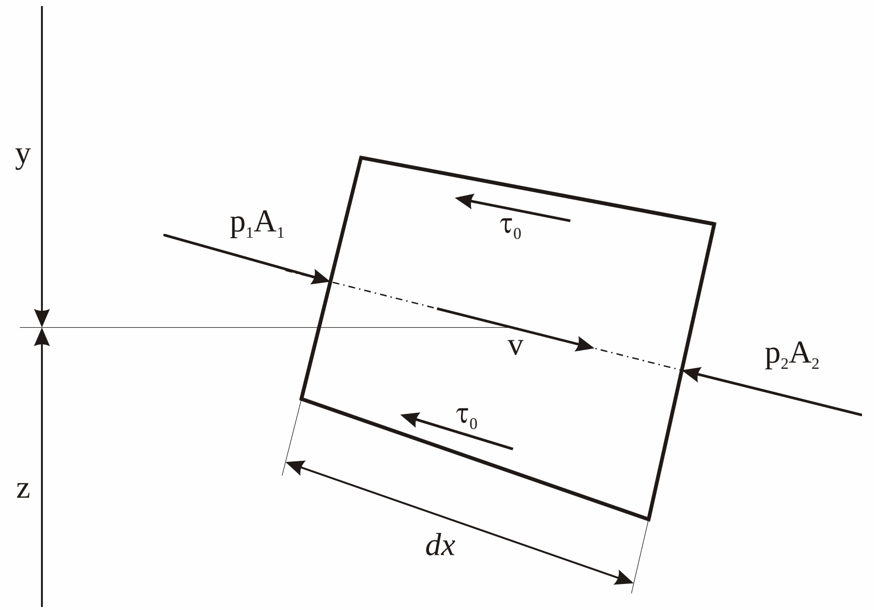
$$F_P = -A\frac{\partial p}{\partial x}dx$$

- Gravity force

$$F_g = -\rho g A\frac{\partial z}{\partial x}dx$$

- Friction force

$$F_\tau = -2\pi R\tau_0 dx$$



$$\tau_0 = -\frac{\rho\lambda|v|v}{8} \qquad \lambda = -\frac{4J_e gR}{v^2}$$

Final Set of Equations

$$\frac{\partial p}{\partial t} + v\frac{\partial p}{\partial x} + \rho a^2 \frac{\partial v}{\partial x} = 0 \qquad \frac{1}{\rho}\frac{\partial p}{\partial x} + g\frac{\partial z}{\partial x} + gJ_e + v\frac{\partial v}{\partial x} + \frac{\partial v}{\partial t} = 0$$

- First order hyperbolic differential equation system
- Two dependent variables p, v
- Two independent variables x, t
- a, ρ, J_e are parameters of the system
- Solution by means of Method of Characteristics

Method of Characteristics

- Hyperbolic differential equation system
  - Interferences propagate at a certain speed
  - Existence of constant variables along characteristic lines :

$$\frac{d}{dt}R^{\pm}\Big|_{C^{\pm}} = 0$$

Riemann Variables

$$\frac{\partial p}{\partial t} + v\frac{\partial p}{\partial x} + \rho a^2 \frac{\partial v}{\partial x} = G_1 = 0$$

$$\frac{1}{\rho}\frac{\partial p}{\partial x} + g\frac{\partial z}{\partial x} + gJ_e + v\frac{\partial v}{\partial x} + \frac{\partial v}{\partial t} = G_2 = 0$$

$$G = G_2 + \xi G_1$$

$$\left[\frac{\partial v}{\partial t} + \left(v + \xi\rho a^2\right)\frac{\partial v}{\partial x}\right] + \xi\left[\frac{\partial p}{\partial t} + \left(v + \frac{1}{\rho\xi}\right)\frac{\partial p}{\partial x}\right] = 0$$

$$\frac{dv}{dt} + \xi\frac{dp}{dt} + g\left(J_e + \frac{\partial z}{\partial x}\right) = 0$$

$$\frac{dx}{dt} = v + \xi\rho a^2 = \frac{1}{\rho\xi}$$

$$\xi = \pm\frac{1}{\rho a}$$

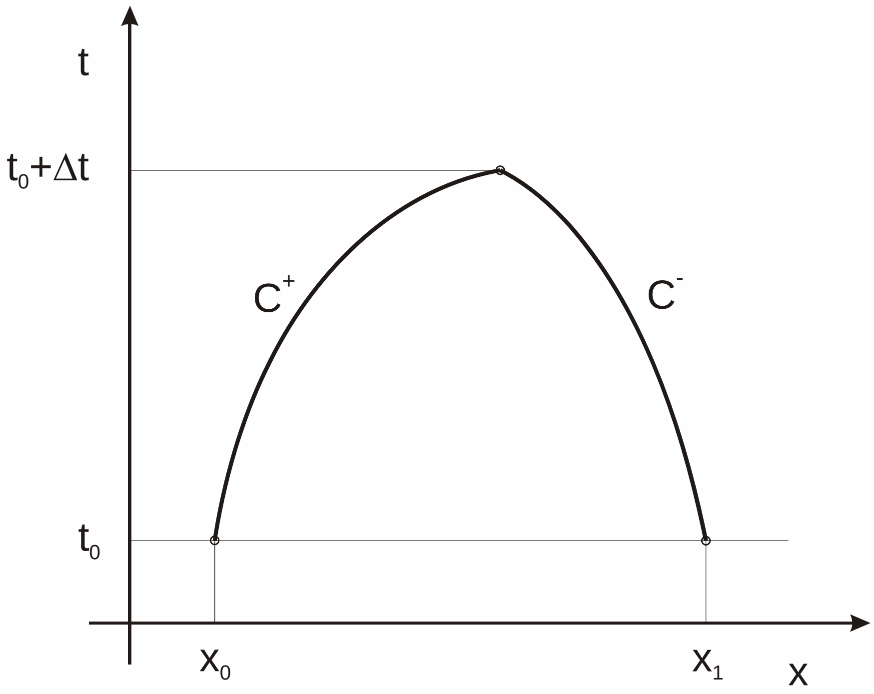
Method of Characteristics

$$\frac{dv}{dt}+\frac{1}{\rho a}\frac{dp}{dt}+g\left(J_e+\frac{\partial z}{\partial x}\right)=\frac{d}{dt}R^{+}=0$$

$$C^{+}=\frac{dx}{dt}=v+a$$

$$\frac{dv}{dt}-\frac{1}{\rho a}\frac{dp}{dt}+g\left(J_e+\frac{\partial z}{\partial x}\right)=\frac{d}{dt}R^{-}=0$$

$$C^{-}=\frac{dx}{dt}=v-a$$



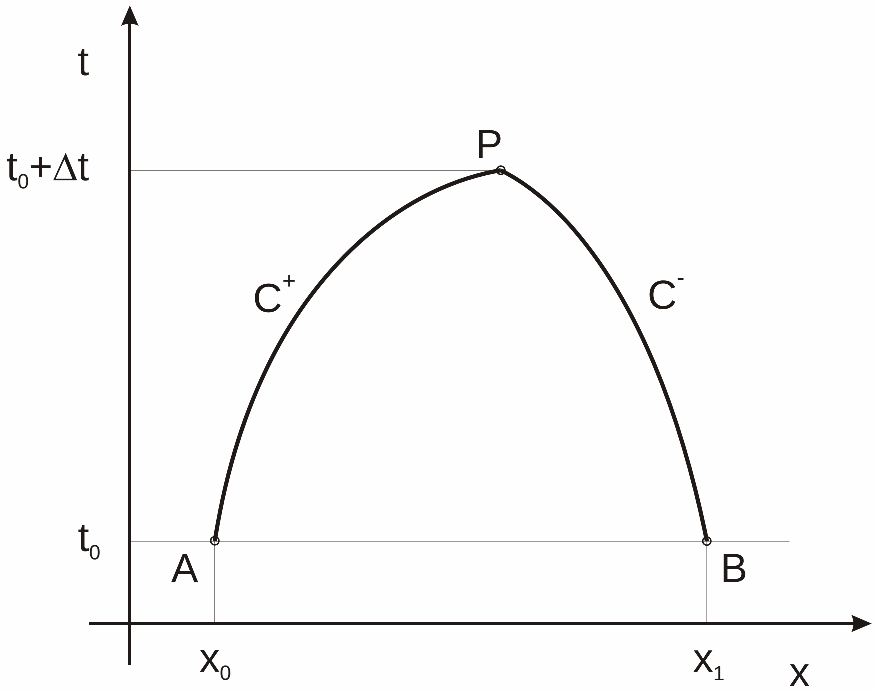
Method of Characteristics

- Calculate values in P from known values in Point A and B
  - Integrate $R^{\pm}$ along $C^{\pm}$

$$\int_A^P dv + \frac{1}{\rho a}\int_A^P dp + g\int_A^P J_e dt + g\int_A^P \frac{\partial z}{\partial x} dt = 0$$

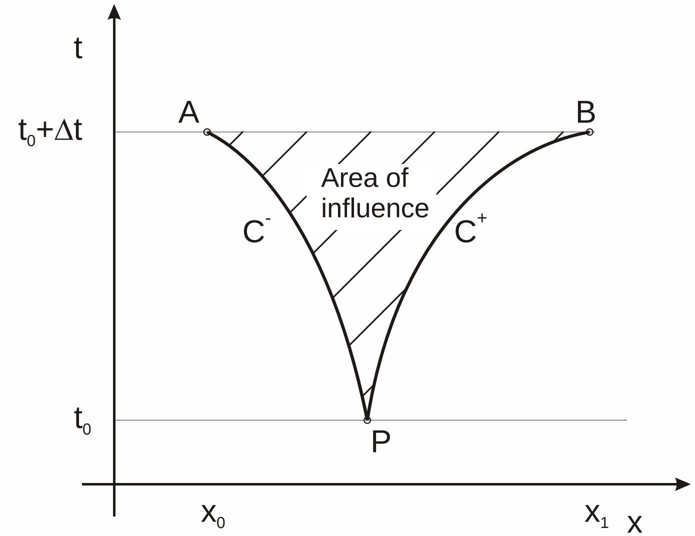
$$\int_B^P dv - \frac{1}{\rho a}\int_B^P dp + g\int_B^P J_e dt + g\int_B^P \frac{\partial z}{\partial x} dt = 0$$

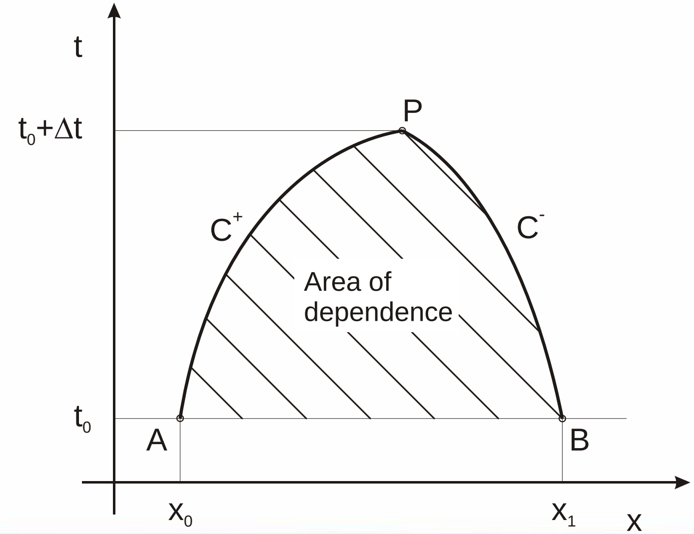
  - Two equations for two unknown variables



Method of Characteristics-Summary

- Solution is only valid on Characteristics
- Area of influence
- Area of dependence
- Characteristics separate two different solution domains





Implementation

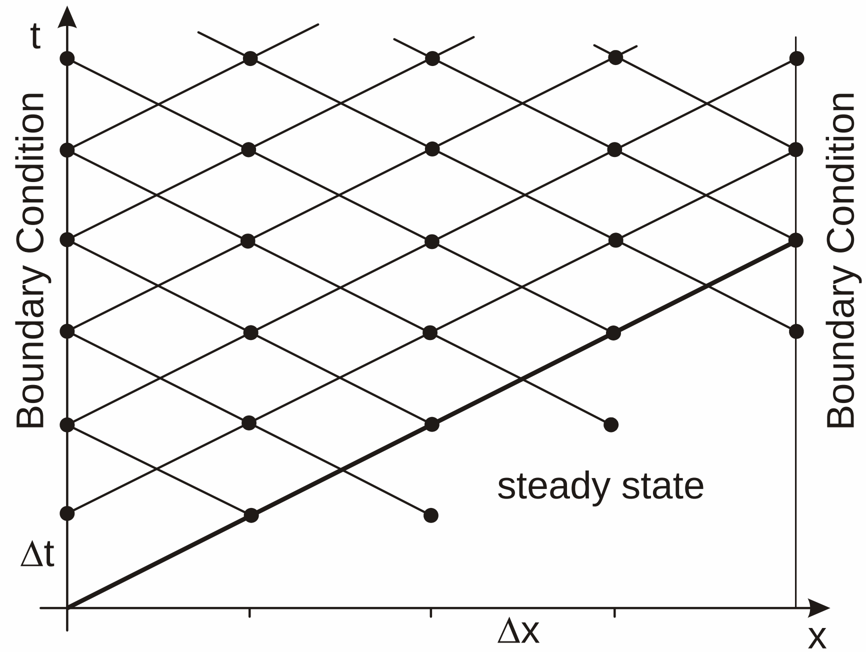
- Simplifications
  - a >> v

$$C^{-} = \frac{dx}{dt} = a \longrightarrow$$ straight lines, $\Delta x = a\Delta t$

  - velocity v is not known along C
    -> Friction term has to be approximated

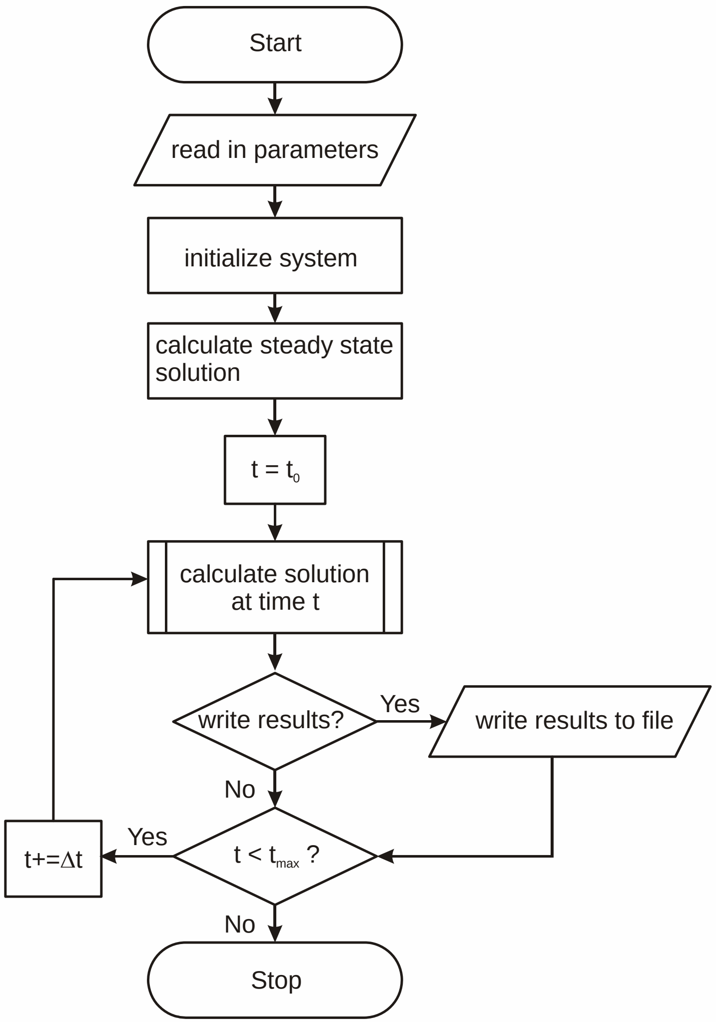
$$\int_A^P J_e dt = J\int v^2 dt \cong Jv_A|v_A|\Delta t + \mathrm{O}(\Delta t) = J_{e,A}\Delta t$$

- Required
  - Computational grid
  - Steady-state solution for the system
  - Boundary condition providing a disturbance



Implementation

- Get system parameters
- Initialize system
  - Calculate Wave velocity **a**
  - Generate calculation grid
- Calculate steady state solution
  - Stationary Hydraulics
- Calculate solution for time t



Implementation

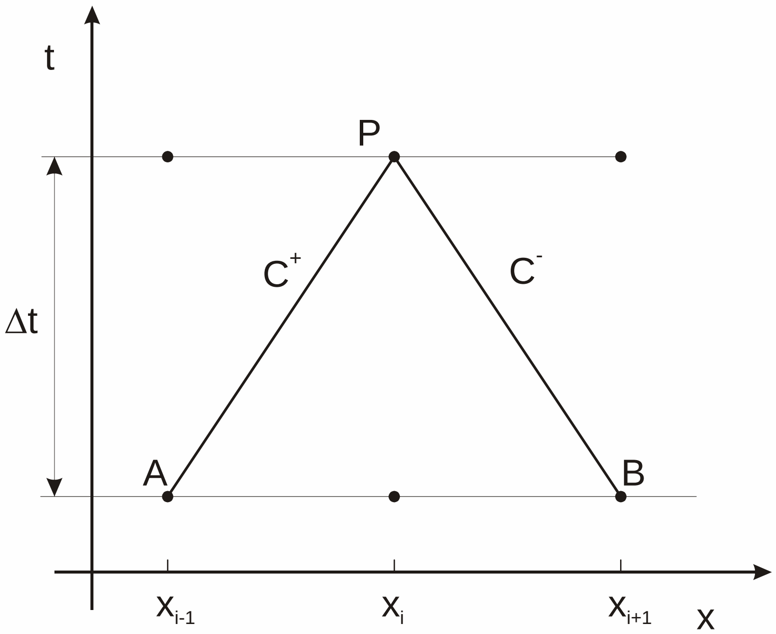
- Method for calculating new time level
  - C^+

$$(v_P - v_A) + \frac{1}{\rho a}(p_P - p_A) + gJ_{e,A}\Delta t = 0$$

$$v_P + \frac{1}{\rho a}p_P \underbrace{- v_A - \frac{1}{\rho a}p_A + gJ_{e,A}\Delta t}_{K_A^+} = 0$$

  - C^-

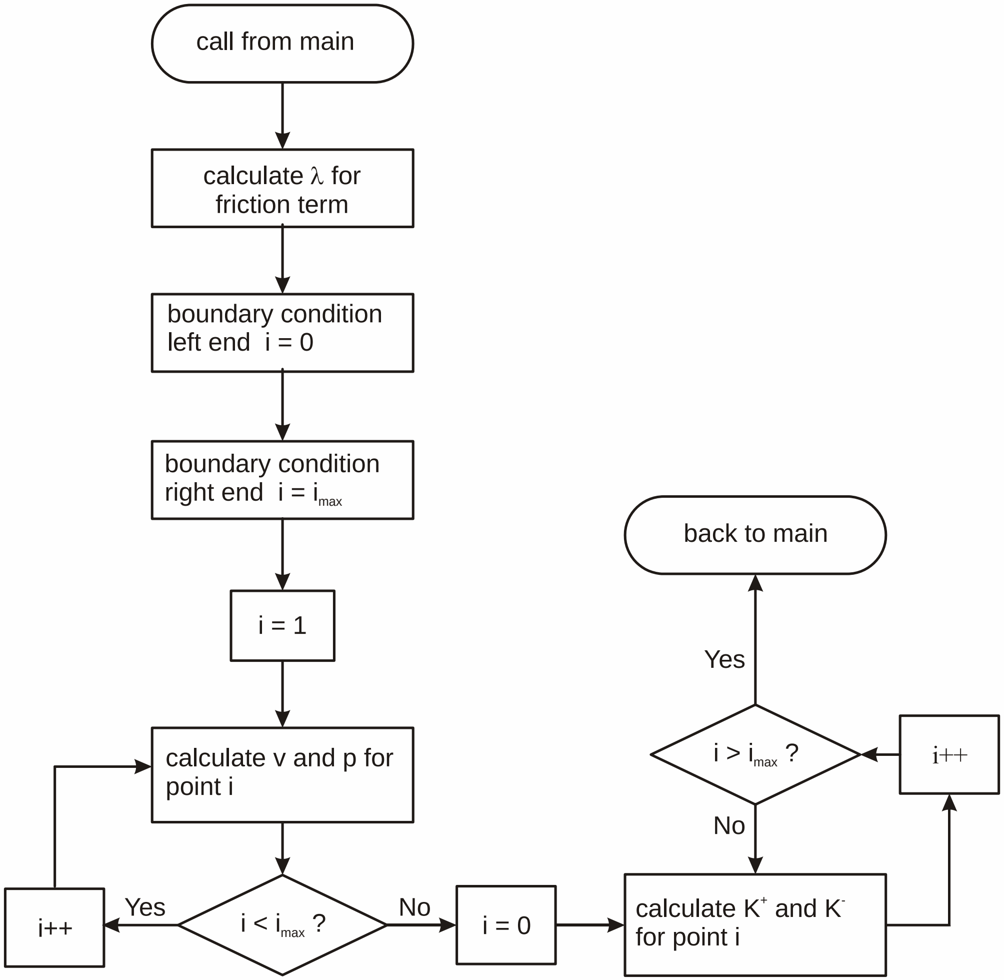
    analog



$$v_P = \frac{K_B^- - K_A^+}{2} \qquad p_P = -\frac{\rho a(K_B^- + K_A^+)}{2}$$

Implementation

- Calculate λ
- Evaluate boundary condition
- Calculate solution for inner field
- Calculate K^+ and K^-



Boundary Conditions

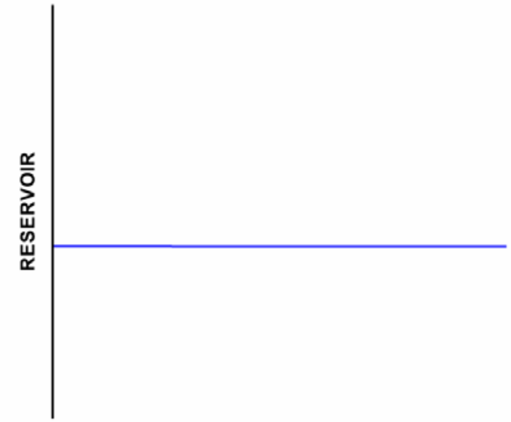
- Static
  - passive → Reservoir
  - active → Valve

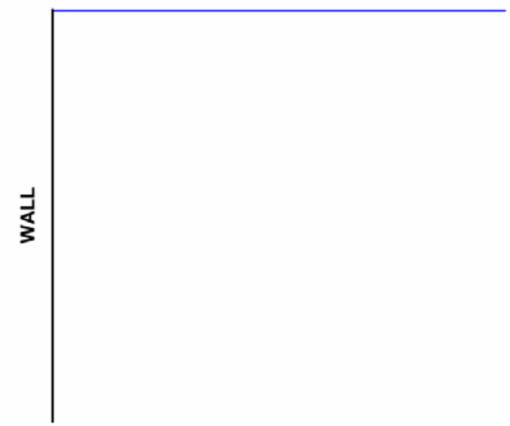
	passive	active
static	$f(p,v)$	$f(p,v,t)$
dynamic	$f(p,\frac{dp}{dt}v,\frac{dv}{dt})$	$f(p,\frac{dp}{dt}v,\frac{dv}{dt},t)$

- Dynamic
  - passive → surge chamber
  - active → Francis Turbine

Boundary Conditions

- Reservoir
  - p = const. ⟶ v is calculated
  - compression waves are reflected as expansions waves
  - expansions waves are reflected as compression waves
- Wall
  - v = 0 ⟶ p is calculated
  - waves are reflected

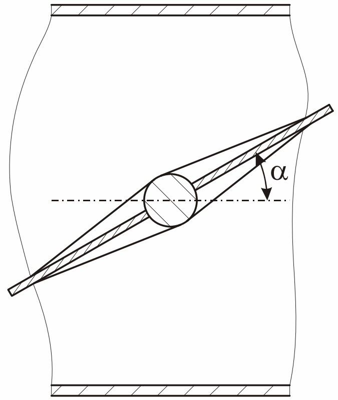


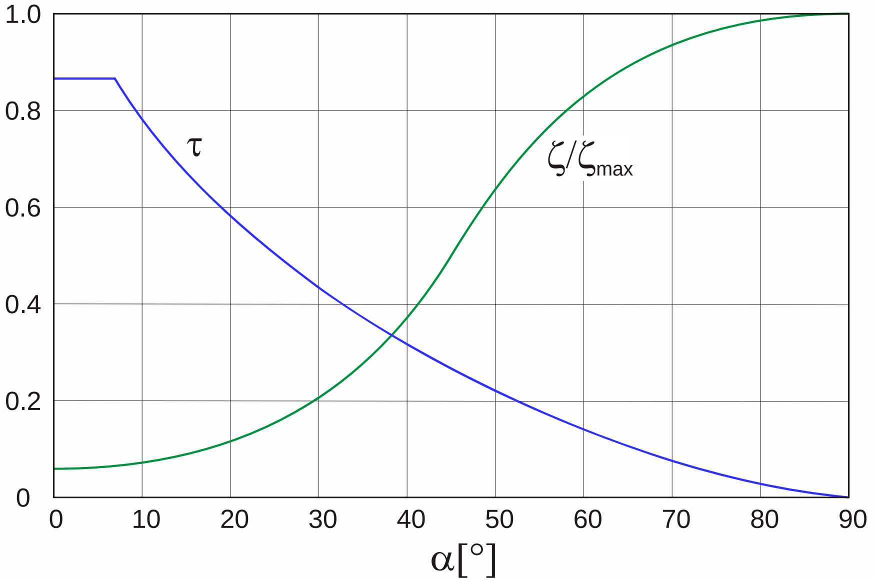


Boundary Conditions

- Valve
  - v = f(Δp, τ, ζ)
  - ζ –> loss coefficient
  - $\tau = \dfrac{A_{valve}}{A_{pipe}}$

$$v_1 = \zeta\tau\sqrt{2/\rho \cdot |p_0 - p_1|}$$

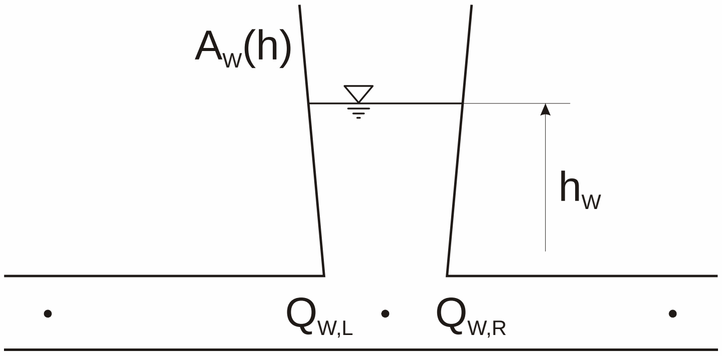




Boundary Conditions

- Surge chamber
  - p_W can not be computed directly
    $Q_W(h_w,\ldots)$
  - predictor corrector scheme
    - guess for p_W
    - calculate Q_W
    - correct p_W
    - correct Q_W

    iterate till convergence

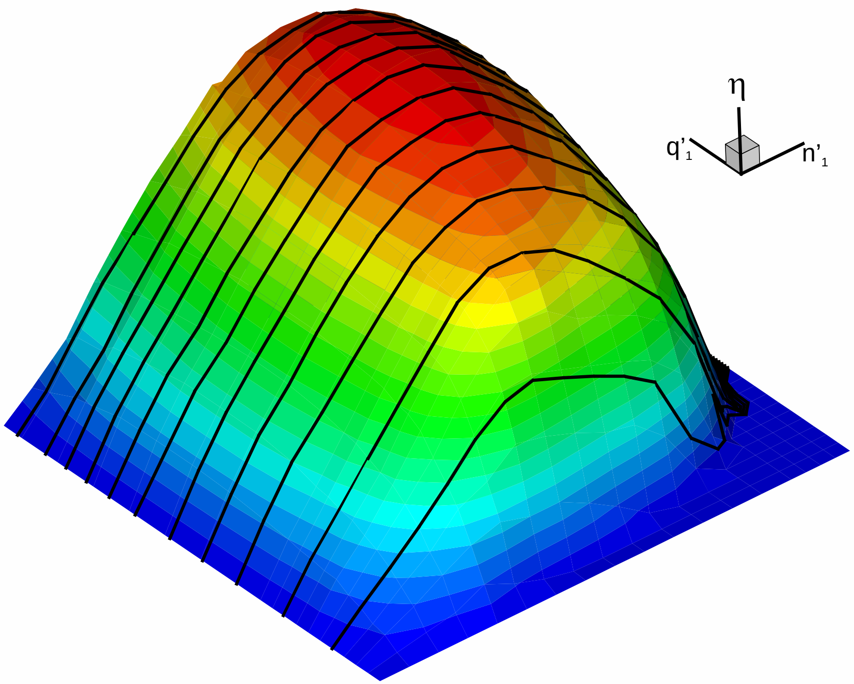


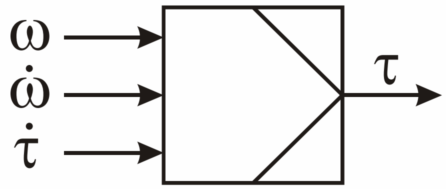
$$\frac{dh}{dt} = \frac{1}{A_W(h)}\left(Q_{W,L} - Q_{W,R}\right)$$

$$\Delta h_W = h_{W,t_0+\Delta t} - h_{W,t_0}$$

Boundary Conditions

- Turbine
  - Machine data
    - Mass moment of inertia of machine and rotating fluid
    - Hill chart
    - Generator
  - Governor
  - Power Grid





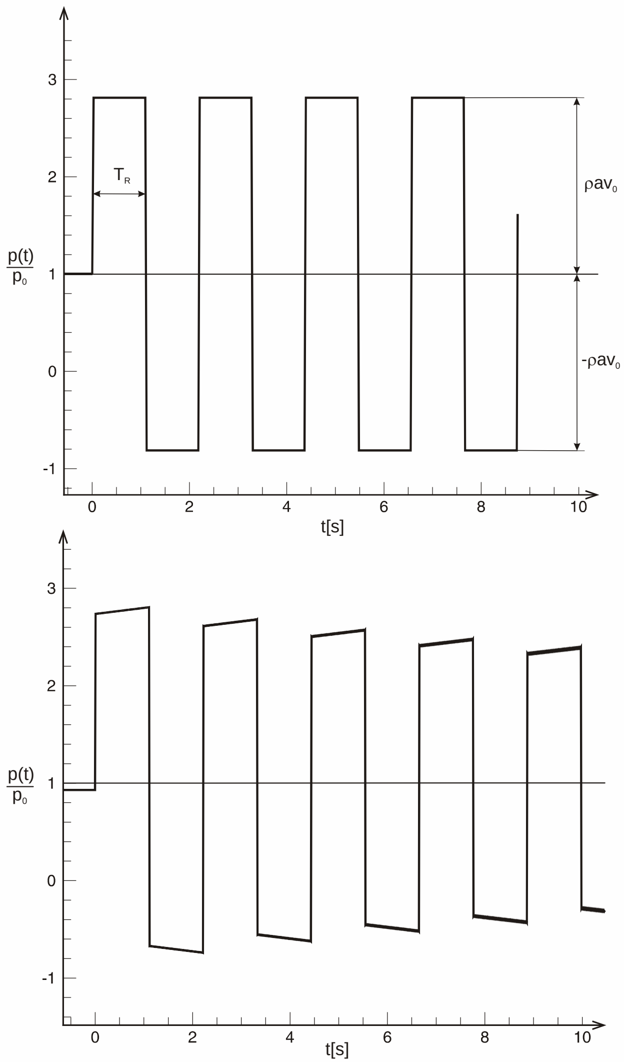
Application

PRESSURE SURGE

- Joukowski - pressure surge
  (Жуковский)

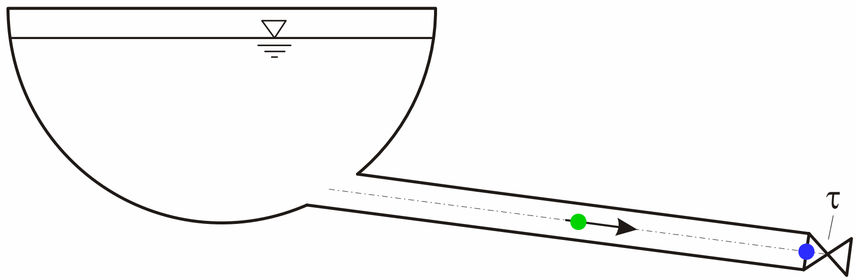
  – sudden closure of a Valve

  – kinetic energy of fluid is converted in pressure



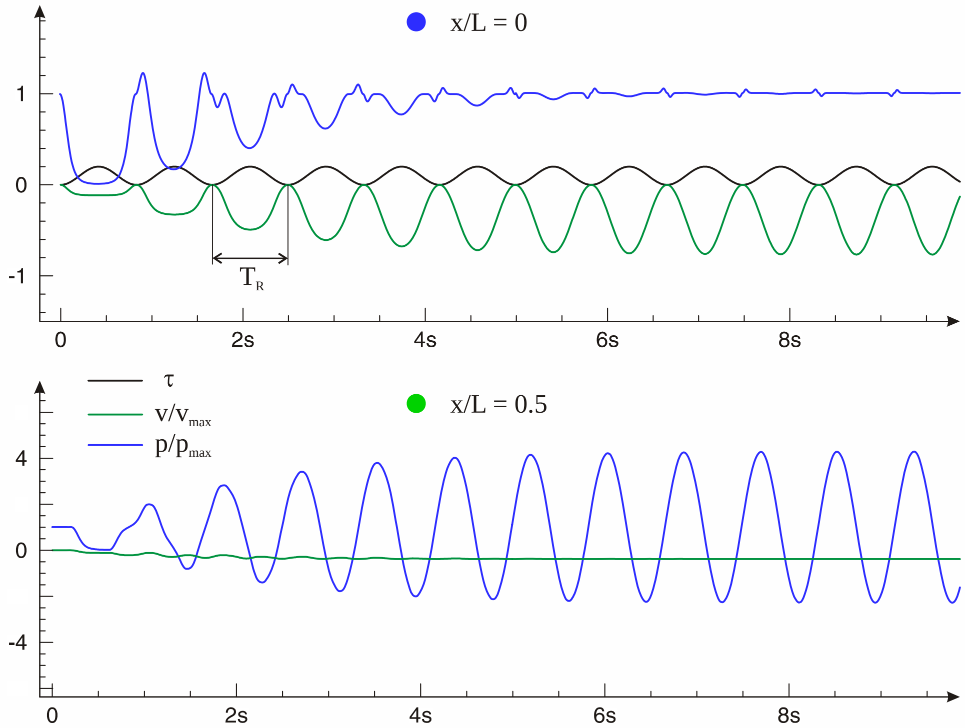
Application

OSCILLATING VALVE $T_\tau = T_R$



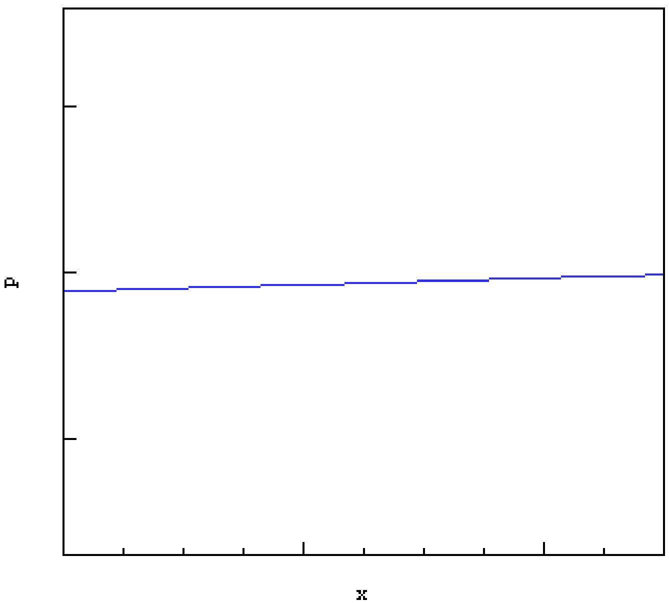
- Results
  - pressure and velocity in time at the valve ●
  - pressure and velocity in time in middle of the pipe ●

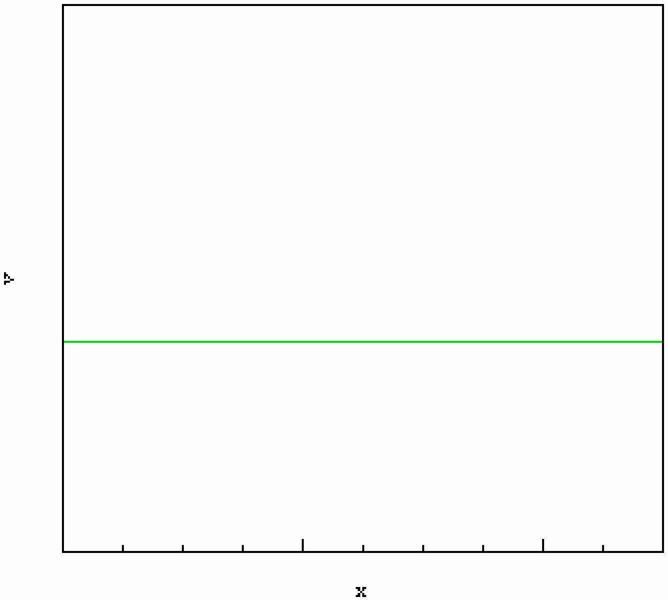
$$\tau = \frac{A_{valve}}{A_{pipe}}$$



Application

OSCILLATING VALVE $T_\tau = 2T_R$



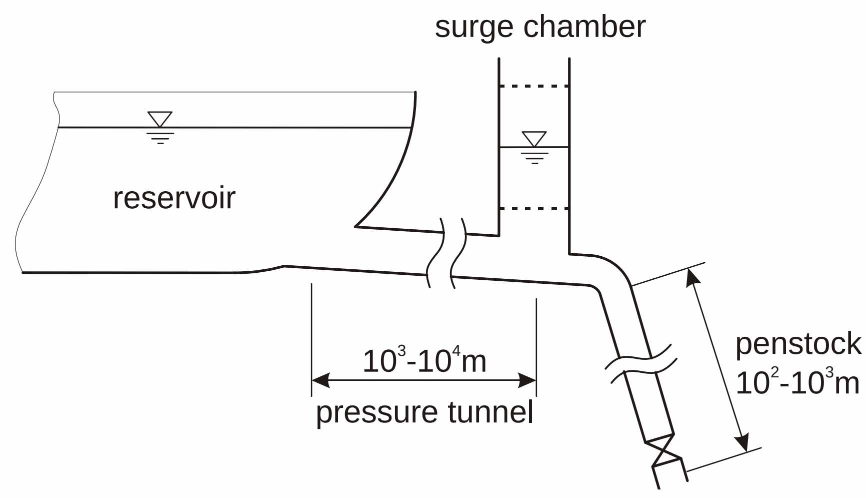


- Pressure and velocity variation along pipe
- $T_\tau = 2T_R$

Application

SURGE CHAMBER OSCILLATION

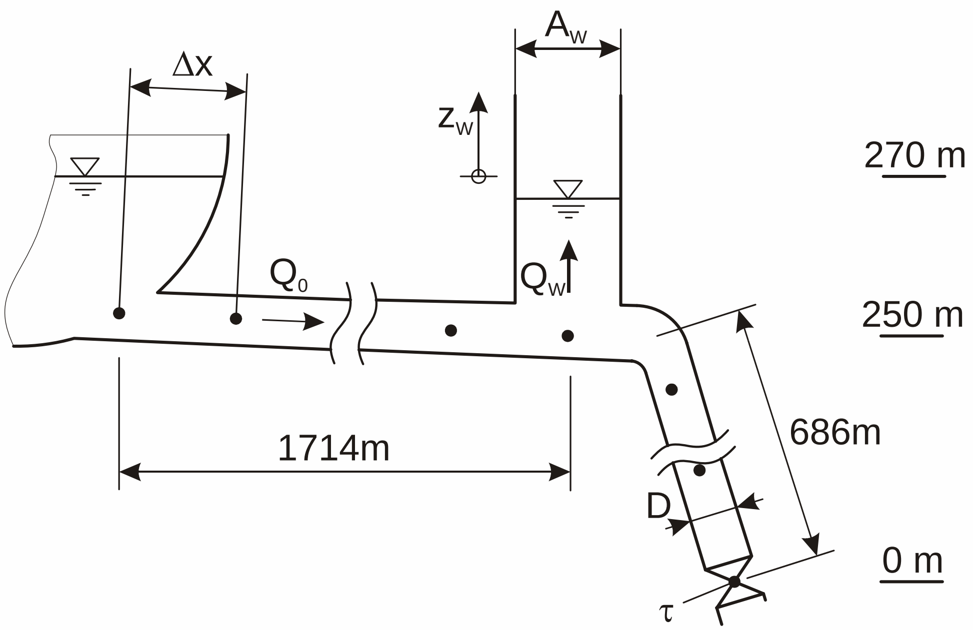
- surge chamber
  - reduces pressure fluctuations
  - hydraulic uncoupling of pressure tunnel from penstock
  - Improvement of control
- surge camber oscillation
  - Interaction of high frequent pressure waves and low frequent inertia waves



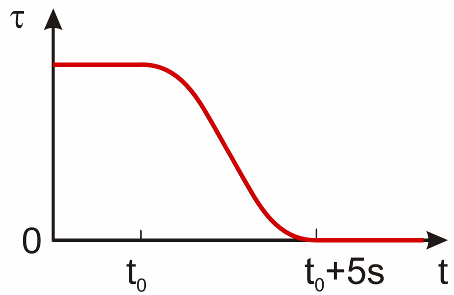
Application

SURGE CHAMBER OSCILLATION

- Model



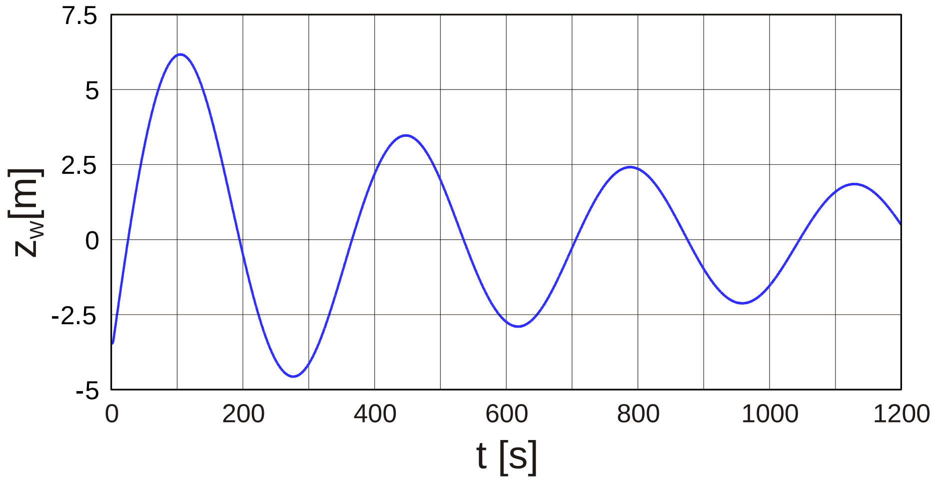
Δx = 137.2 [m]
Δt = 0.14263 [s]
a = 961.92 [m/s]
A_w = 79.0 [m^2]
D = 2.5 [m]
Q_0 = 12.25 [m^3/s]

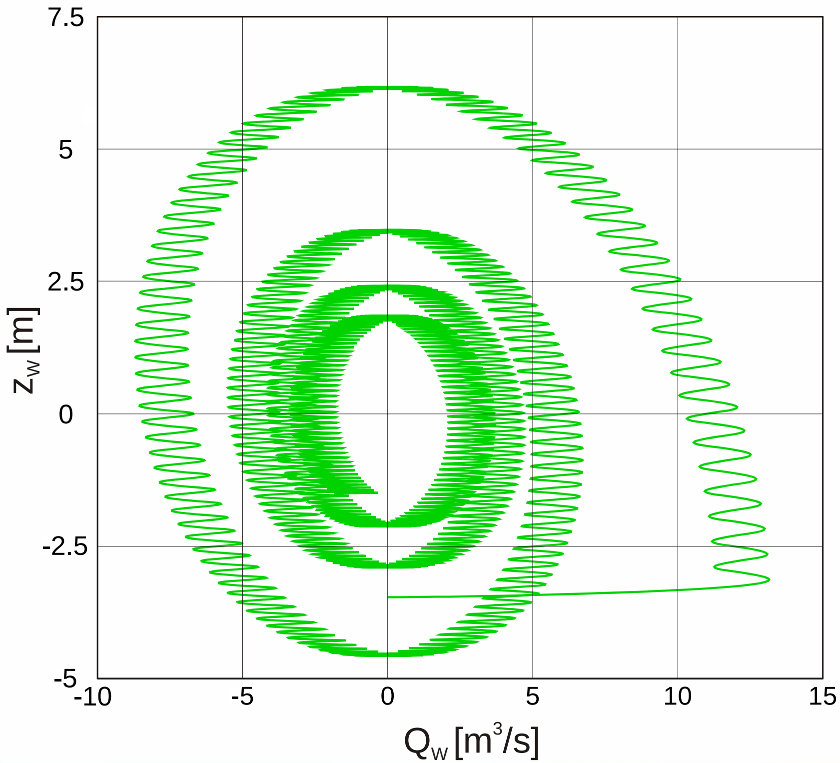


Application

SURGE CHAMBER OSCILLATION

- Results
  - Elevation of surge chamber water surface
  - Zustandsdiagramm

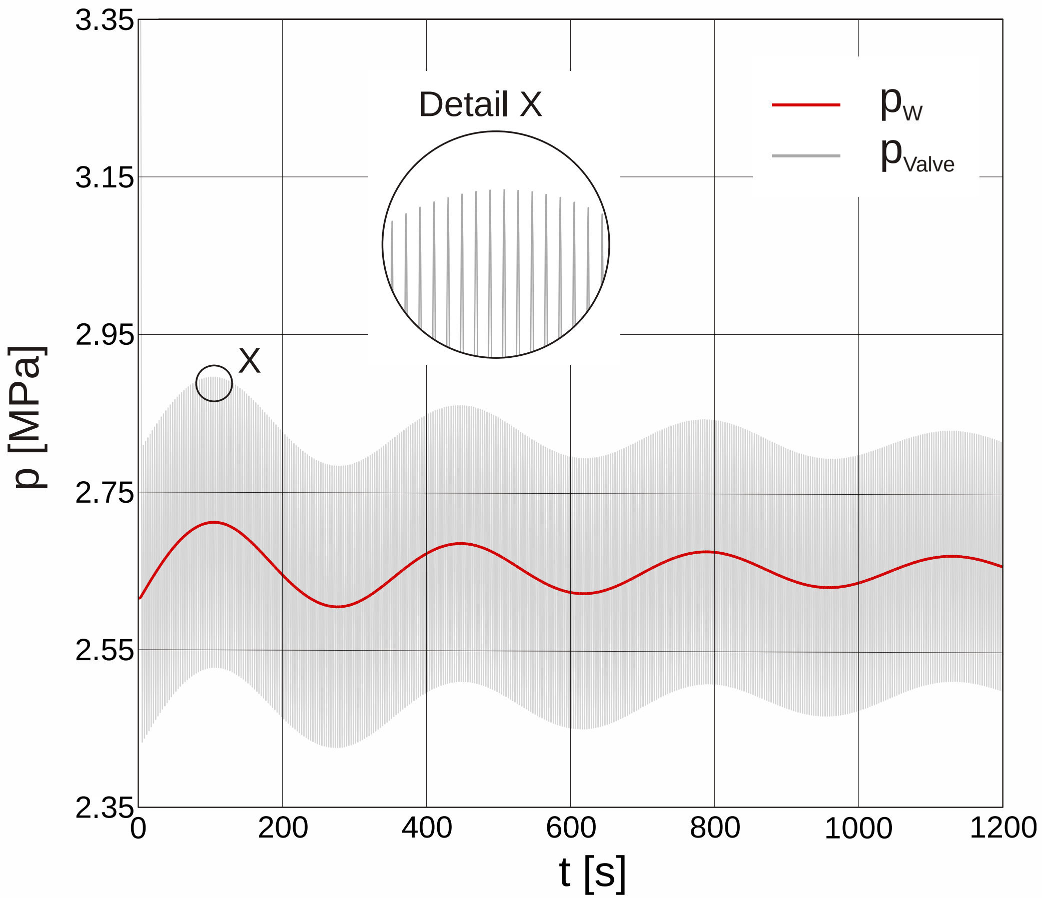




Application

SURGE CHAMBER OSCILLATION

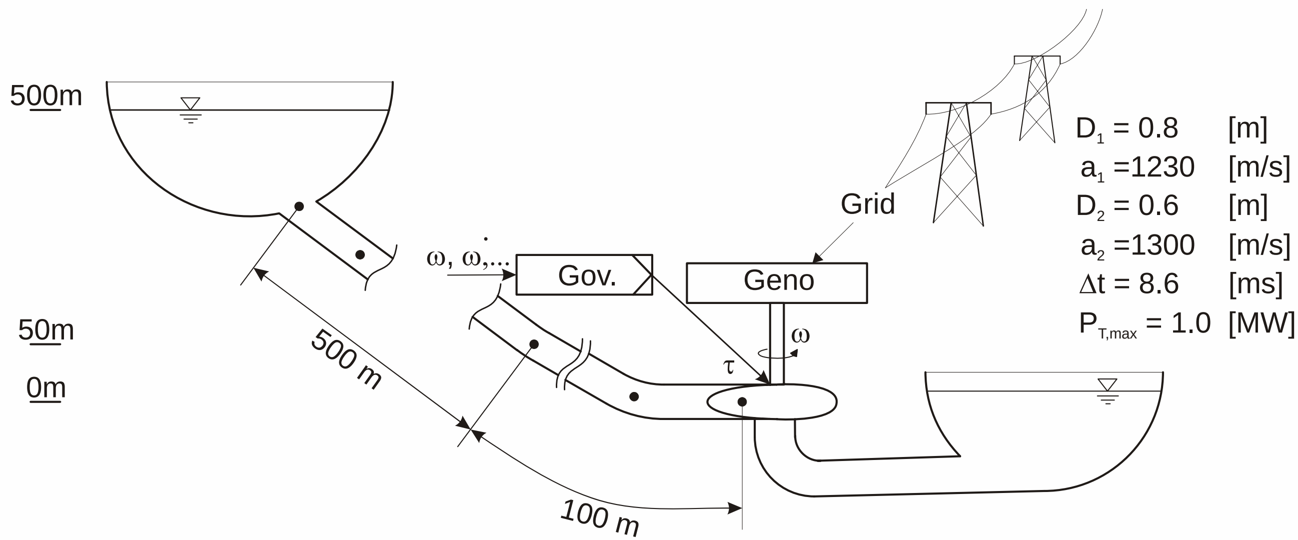
- Results
  - pressure at the lower end of the penstock
  - pressure at the upper end of the penstock



Application

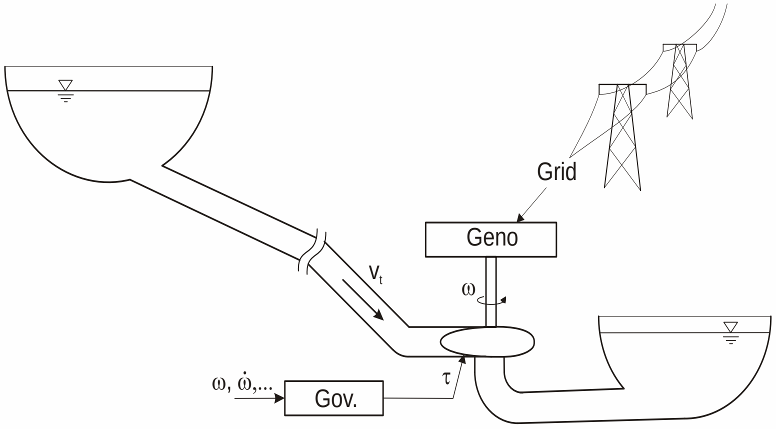
POWER PLANT

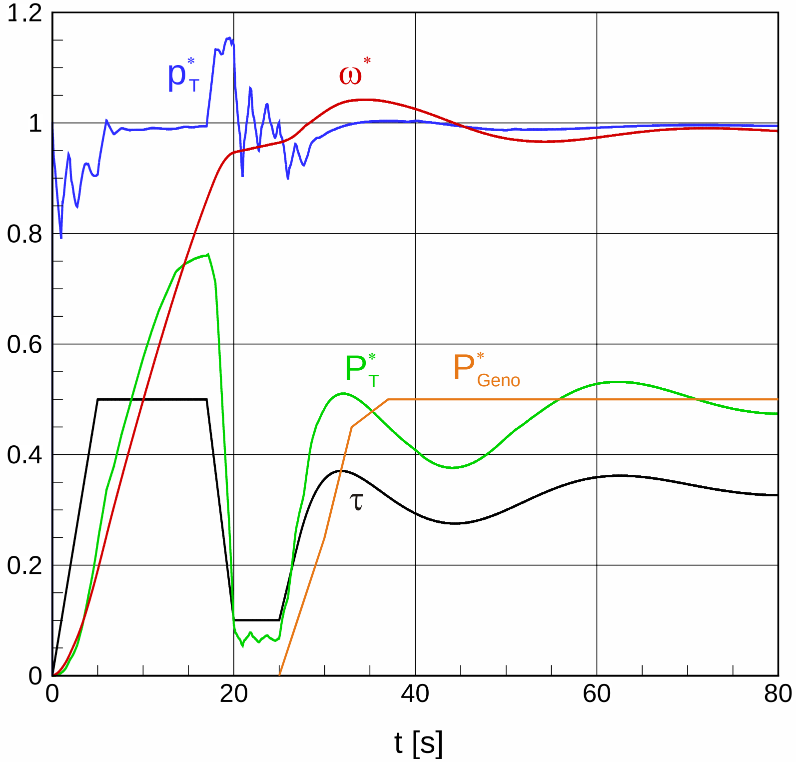
- Model
  - Simulation of plant, governor, grid and machine
  - Francis turbine
  - Isolated grid



Application

POWER PLANT

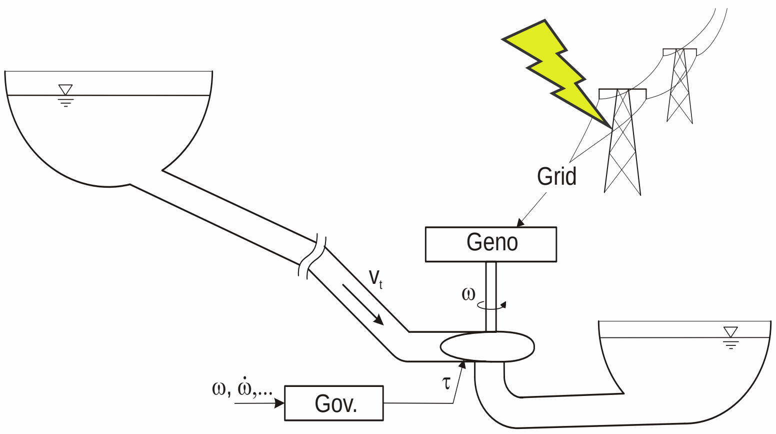


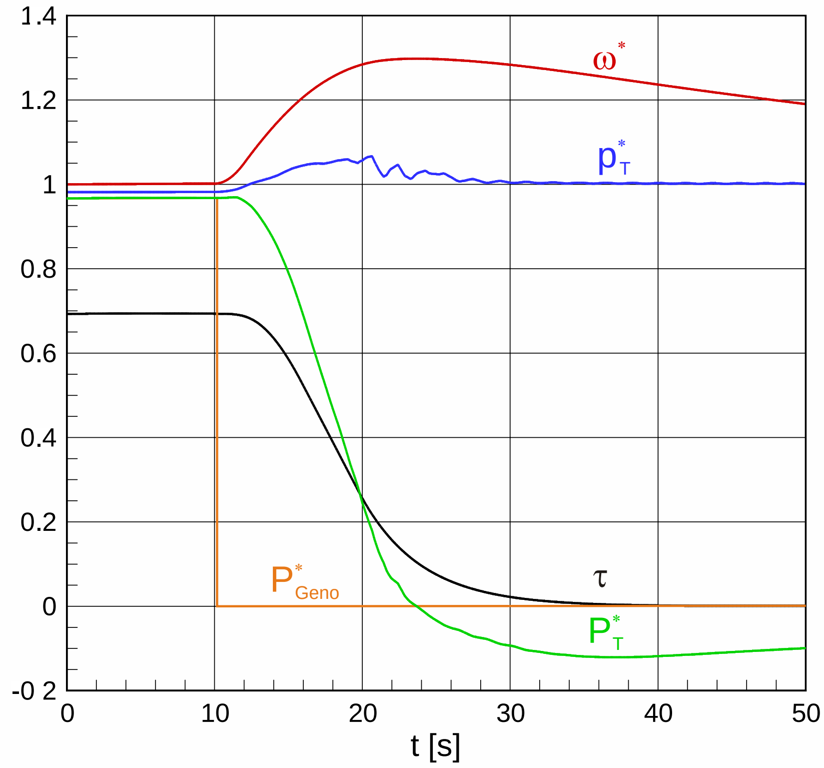


- start up
  - open loop control to overcome mass moment of inertia
  - synchronization
  - closed loop control by governor

Application

POWER PLANT



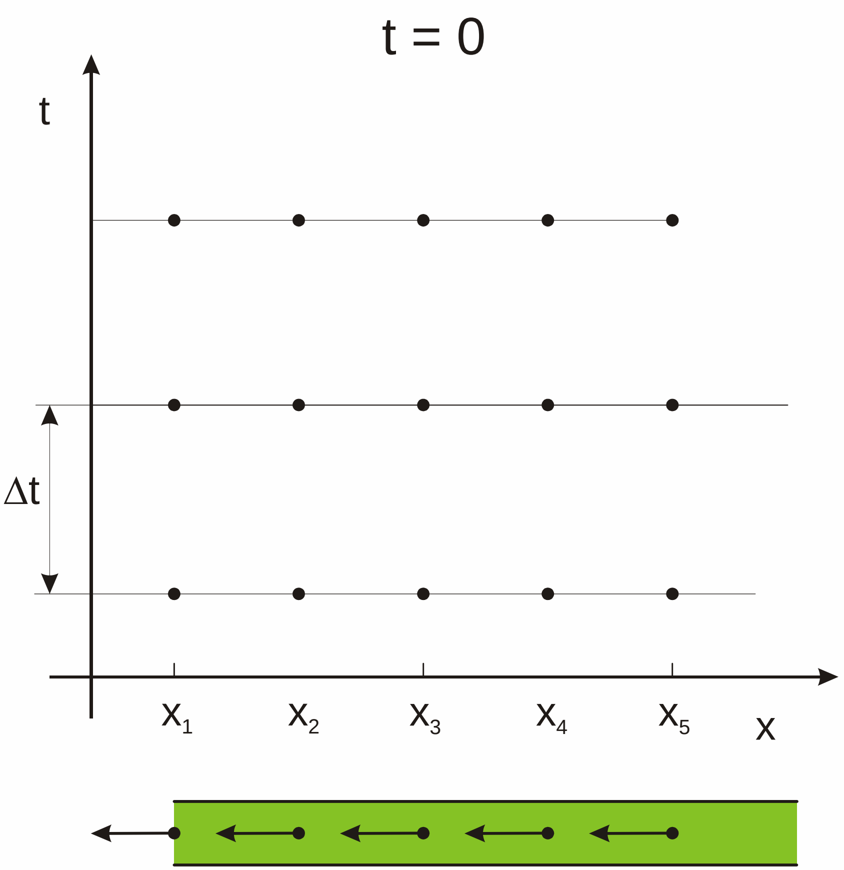


- load rejection
  - Sudden load rejection caused by failure in grid or generator

Thank you for your attention!

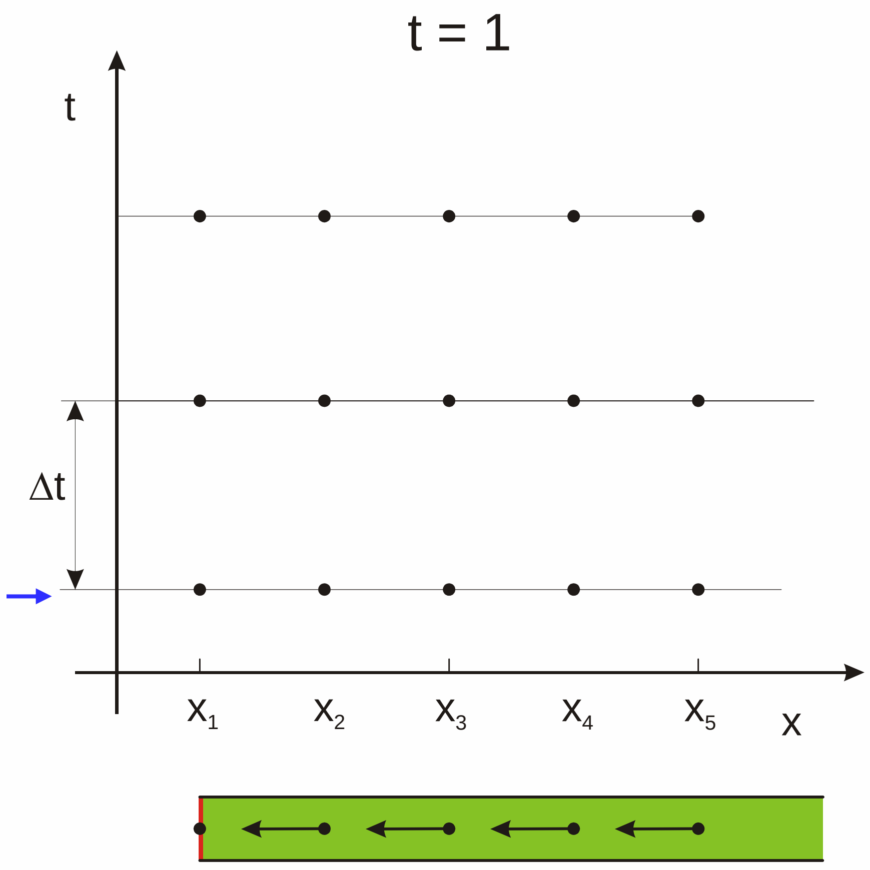
Implementation

- Method for calculating new time level
  - Initial solution
  - constant velocity
  - low pressure



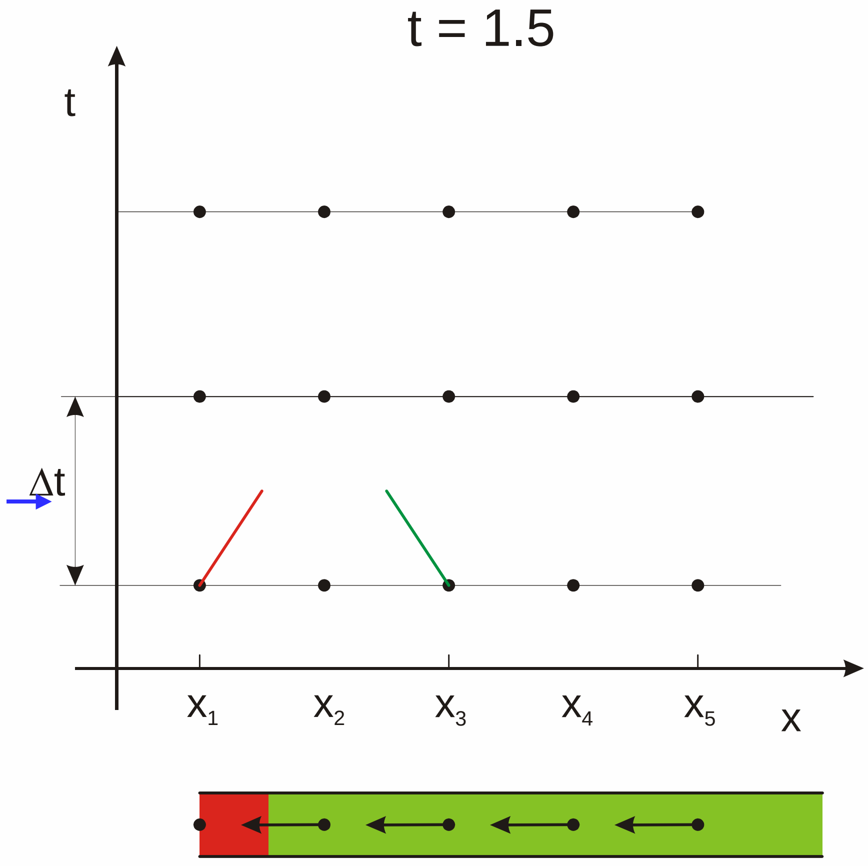
Implementation

- Method for calculating new time level
  - valve closes suddenly



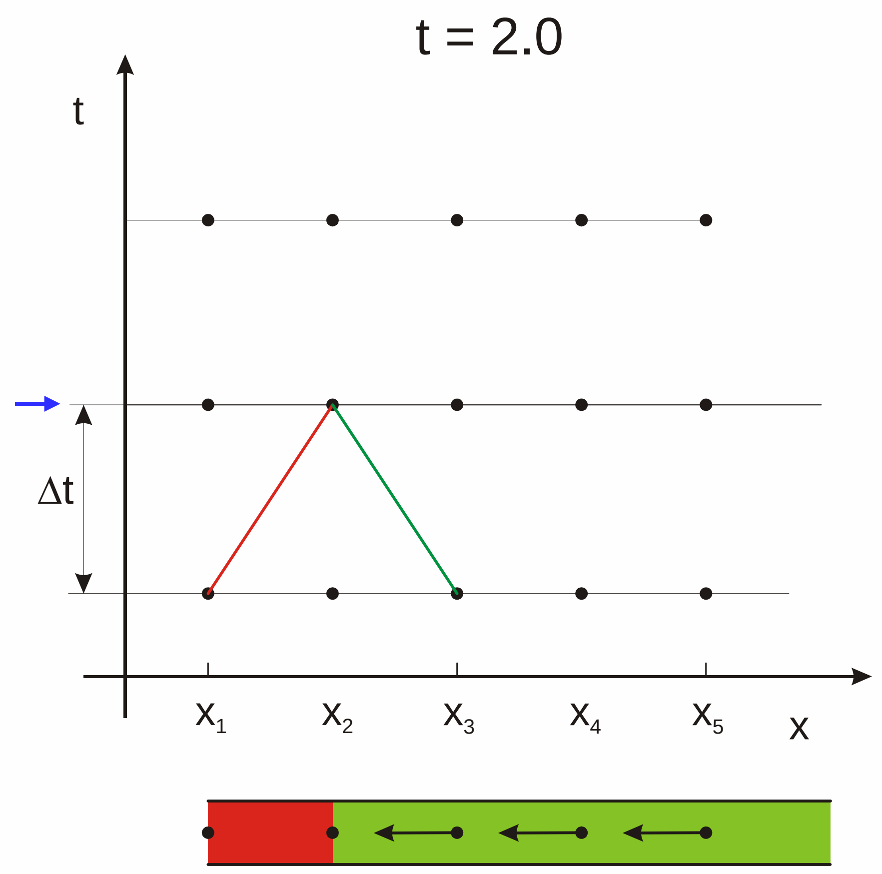
Implementation

- Method for calculating new time level
  - disturbance propagates upstream
  - half time step



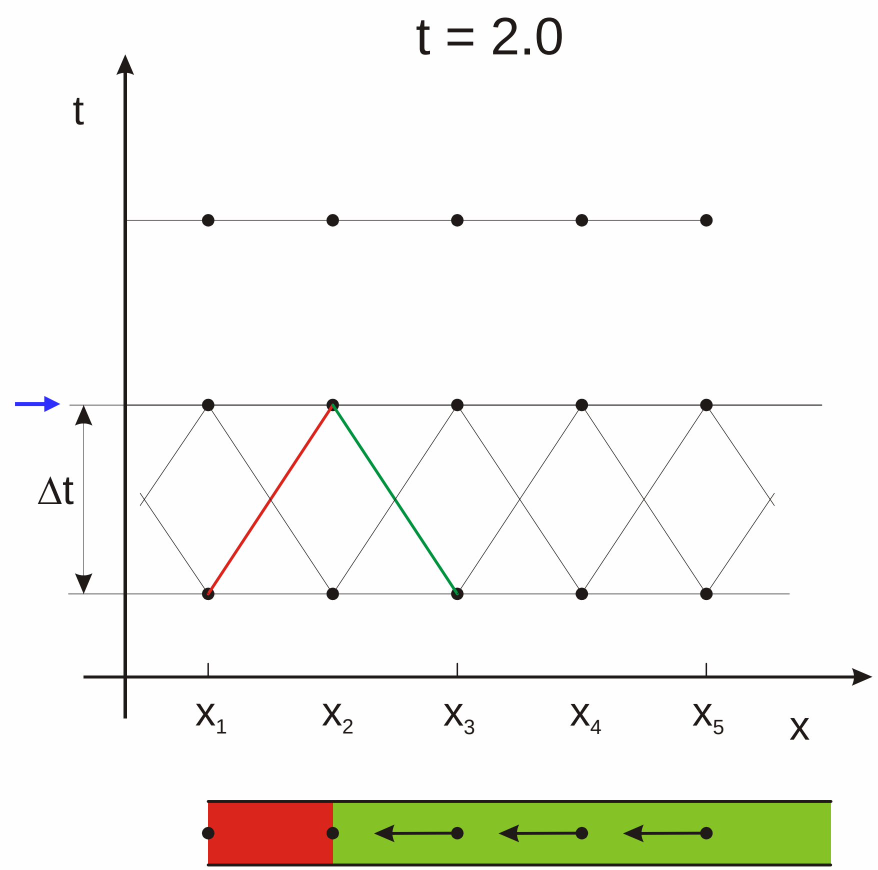
Implementation

- Method for calculating new time level
  - Next time level
  - calculate solution on intersection Point



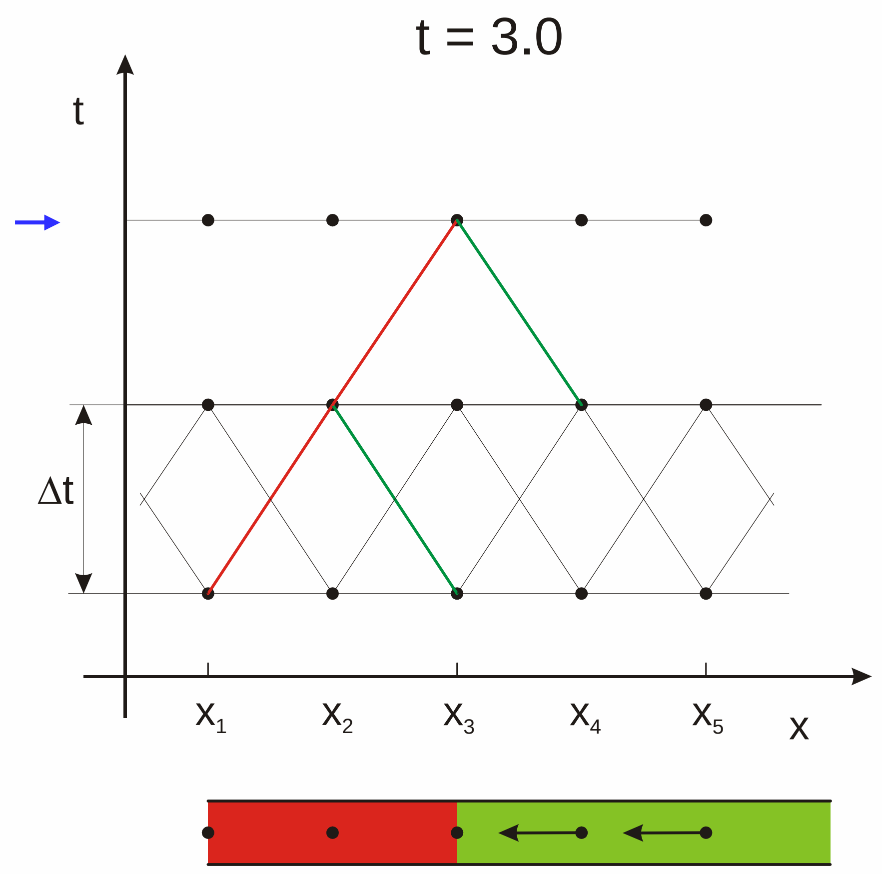
Implementation

- Method for calculating new time level



Implementation

- Method for calculating new time level



Implementation

- Method for calculating new time level