POLITECNICO
DI TORINO
Dipartimento Energia
"Galileo Ferraris"

Gas Network modelling for a multi-gas system

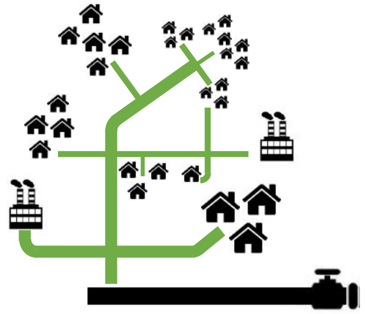
Version July 12th, 2023

Gas Network Model

Overview on Fluid-Dynamic Solver

Gas Network Model

Overview:



1D gas flow -

Transient -

Isothermal -

Multi-component -

Parameters

- Network Topology
- Pipelines Features
- Gas Profiles
- P, Composition
  at injection node

Inputs

Physical Model

Outputs

Gas Flow -
for each branch

P, Composition -
at each node

At each timestep

Equations:

Coninutiy equation

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

Flow equation

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho v^2)}{\partial x} + \frac{\partial p}{\partial x} + \frac{\lambda}{2D}\rho v|v| + \rho g \sin\alpha = 0$$

Coupled equations
to be linearized
and solved iteratively

Closure equation:

Equation of State

$$\frac{p}{\rho} = Z\frac{R_0}{MM}T$$

Z determined through **GERG-08 wide range equation of state**

Gas Network Model

Solving Strategy:

p_{inlet}^{0} ; $\boldsymbol{G_{ext}^{0}}$

initial [t=0]
steady state solution

SIMPLE algorithm adaptation

Equation of State

updates the fluid properties

Coninutiy equation

$$\frac{V_i}{c_{i,j}^2 \Delta t}(p_i^{n+1} - p_i^n) = \sum_{j=1}^{k} G_{i,j}^{n+1} - G_{ext,i}^{n+1}$$

Flow equation

LINEARIZED EQUATION

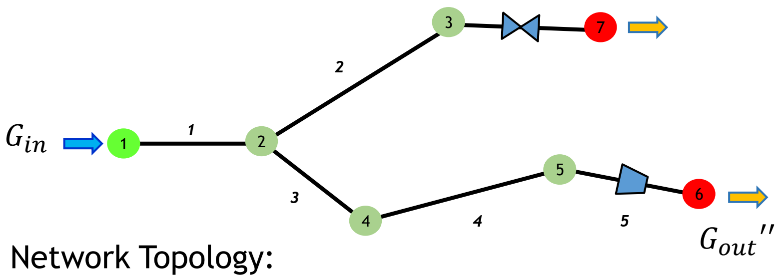
$$\Delta P_j^{(k+1),n+1} - \left(2R_{f\,j}\left|G_j^{(k),n+1}\right| + R_{I_j}\right) G_j^{(k+1),n+1}$$
$$= - R_{f_j}\left|G_j^{(k),n+1}\right| G_j^{(k),n+1} - R_{I_j} G_j^n$$

Coupled fluid-dynamic problem

$$\Delta P_j^{n+1} - \left[R_{f_j}|G_j^{n+1}|G_j^{n+1} + R_{I_j}(G_j^{n+1} - G_j^n)\right] < toll$$

no

Networkwide:



Network Topology:

Incidence Matrix
representation of the network

+1 inlet node
-1 outled node
Reading column-by-column

branches

$$\boldsymbol{X} = \begin{array}{c c} & \begin{array}{cccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} \end{array} \\ \begin{array}{c} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \\ \mathbf{6} \\ \mathbf{7} \end{array} & \begin{bmatrix} +1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \end{array}$$

nodes

Repeated for
each Timestep

Input data structure

Input Data Structure - EDGE TABLE

Non pipeline elements

nodes

branch	IN	OUT	L (km)	D (m)	epsi[mm]	COMP	REG	VAL	RES	n°Grid points
1	1	2	530.0	0.7937	0.014					53
2	2	3	1.0	0.7937	0.014	1				1
3	3	4	540.0	0.7937	0.014					54

Topological info:
Branch (edges) list and indication of inlet and oulet node

Either 0 or 1 to indicate that a specific branch is a Non-pipeline element thus the equations applied are different

Used to create a mesh within each pipe (used both for the fluiddynamic problem and more importantly, for the quality tracking)

Some additional column may be added as additional information about each pipe:

For example, to take into account the gravitational effect, if the altitude at each node is not available, then the DeltaH or the inclination angle should be indicated for each pipe

Non pipeline elements

Element Types	Description
Passive Elements	
pipe	models a section of a pipeline, basic properties are length, diameter, roughness and pipe efficiency
resistor	models passive devices that cause a local pressure drop (e.g. meters, inlet piping, coolers, heaters, scrubbers etc.)
Active Elements	
compressor	models a compressor station with generic constraints, allows the specification of a control mode of the station (e.g. outlet pressure control, inlet pressure control, flow rate control etc.)
regulator	models a pressure reduction and metering station located at the interface of two neighbouring networks with different maximum operating pressures, allows the specification of a control mode of the station (e.g. outlet pressure control, inlet pressure control, flow rate control etc.)
valve	models a valve station, which is is either opened or closed

Non-pipeline (NP) elements have not been «structurally» embedded. The structure of the models allows for their integration but a more robust and general organization of the code and the integration of the NP elements should be done

Input Data Structure - NODE TABLE

Others units my be accepted such as kg/s or Sm3/h

Vector of the molar composition of natural gas

Node name	Node	Height [m]	Gas Flow [kW]	Pressure [bar-g]	CH_4	N_2	CO_2	C_2H_6	C_3H_8	H_2	...	H_2S
Wafa	1	0	-5,203,292	**100**	85.306	3.882	1.268	6.486	2.058	0	…	0
Mellitah	2	100	0	65							...	
Mellitah_out	3	200	0	120							...	
Gela	4	150	**5,203,292**	70	85.306	3.882	1.268	6.486	2.058	0.000	…	0.000

Some additional column may be added here as well depending on the number of components of the natural gas we want to consider: For quality tracking goals I would say that the minimum is 2

Usually the initial pressure is a given value as a boundary condition

Some additional column may be added For example, for graphical representation purposes, colums for x-y coordinates should be integrated

Usually the outlet gas flow (here given in terms of thermal energy release) is a given value as a boundary condition

This should go together with the choice of the equation of state.

Sign convention:

+ if it is ***exiting*** form the network;

– If it is **entering** in the network

Matlab Functions Management

Matlab Functions Management

A_GRID CREATOR

Input PIPE
Input NODES

It creates a .mat set of data about the infrastructure to avoid to always reading the data structures on Excel when working on the same infrastructure.

Given a topology, It is structured so that it is possible to set a number of points for each pipeline so to create a more refined topology of the network.
It is useful not only to obtain much precise fluid-dynamic results but also for the preparation of the batch method for quality tracking

DATA_INPUT_Trial2023_3.mat

This is the model for the fluid-dynamic and quality tracking simulation of the network.

A_MODEL

Input Profiles

It reads from an excel file the time profiles of the set quantities at the boundary condition (pressure, mass flow, composition)

These are all functions that are called from the main script already in the preparation phase to calculate some useful quantities such as the molar mass, the mass based heating value and to convert molar concetration to mass and vice versa

MOL2MASS_CONV
MolarMassGERG
MASS_HV
MASS2MOL_CONV
...

Matlab Functions Management

A_MODEL

Steady State

This function is called to calculate the initial condition of all the network simulation that is assumed to be in steady state.

This function uses the same solving algorithm for the fluid dynamic but the time dependent feature is set to 0

ReducingParametersGERG;
PseudoCriticalPointGERG;
PropertiesGERG

EQUATION OF STATE (GERG-2008)

Other minor functions

For any pressure of the gas in the network (the temperature is assumed to be always constant), all the other properties of the gas needs to be calculated.

There are many possible Equation of State that ones can call

GERG-2008 is the **most advanced** one but also the **most computationally heavy**.

The set of functions and subfunction were translated from a VBA tool available online by Marco Cavana, generating a quite complex system of functions and subfunctions, probably not the most efficient one.

One could think of **structure the program** to **give a couple of choices** to the final user **on what is the desired Eq. Of State that is to be used**

Matlab Functions Management

A_MODEL

Steady State

This function is called to calculate the initial condition of all the network simulation that is assumed to be in steady state.

This function uses the same solving algorithm for the fluid dynamic but the time dependent feature is set to 0

ReducingParametersGERG; PseudoCriticalPointGERG; PropertiesGERG

EQUATION OF STATE (GERG-2008)

Other minor functions

OTHERS POSSIBLE ALGORITHM CAN BE IMPLEMENTED

In previous versions, the author used a different algorithm for the calculation of the steady state condition:

The **SIMPLE** algorithm adapted for the compressible fluid networks.

In general the SIMPLE algorithm is said to be very robust

We can think of implement it in the open access tool.

Matlab Functions Management

A_MODEL

Steady State

This function is called to calculate the initial condition of all the network simulation that is assumed to be in steady state.

This function uses the same solving algorithm for the fluid dynamic but the time dependent feature is set to 0

ReducingParametersGERG;
PseudoCriticalPointGERG;
PropertiesGERG

EQUATION OF STATE (GERG-2008)

Other minor functions

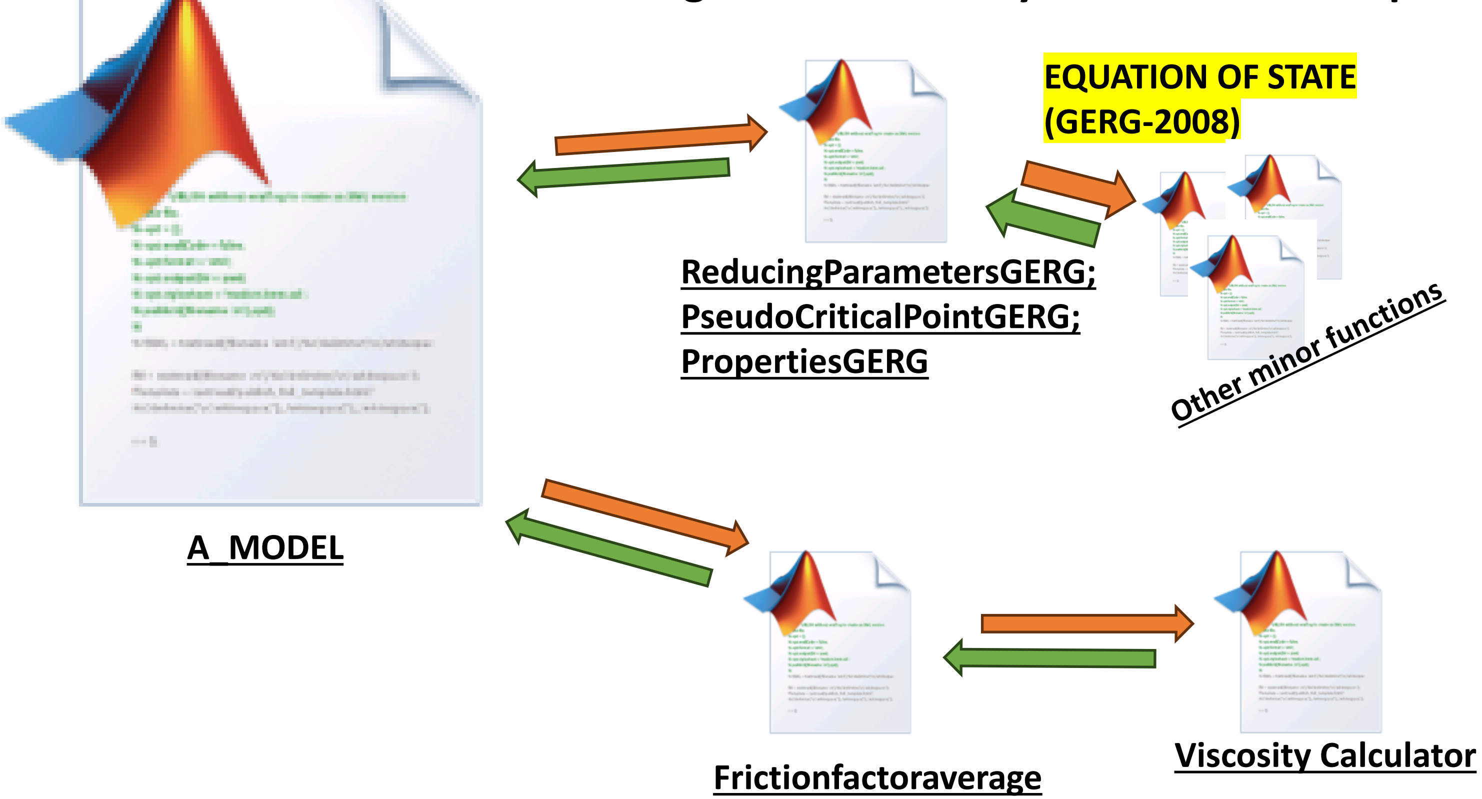
```
save('Trial2023_3_P0','p_0');
save('Trial2023_3_G0','G_0');
```

The calculation of the steady state gives
G0 = the mass flow rates at any pipeline (or pipeline section) of the network in the hypothesis that at t=1 the fluid-dynamic of the network is at steady state.
P_0 = the nodal pressure at any node of the network

These quantities are generally saved to save time when doing trials on the same network and starting from the same conditions

Matlab Functions Management

Getting into the fluid-dynamic simulation phase:

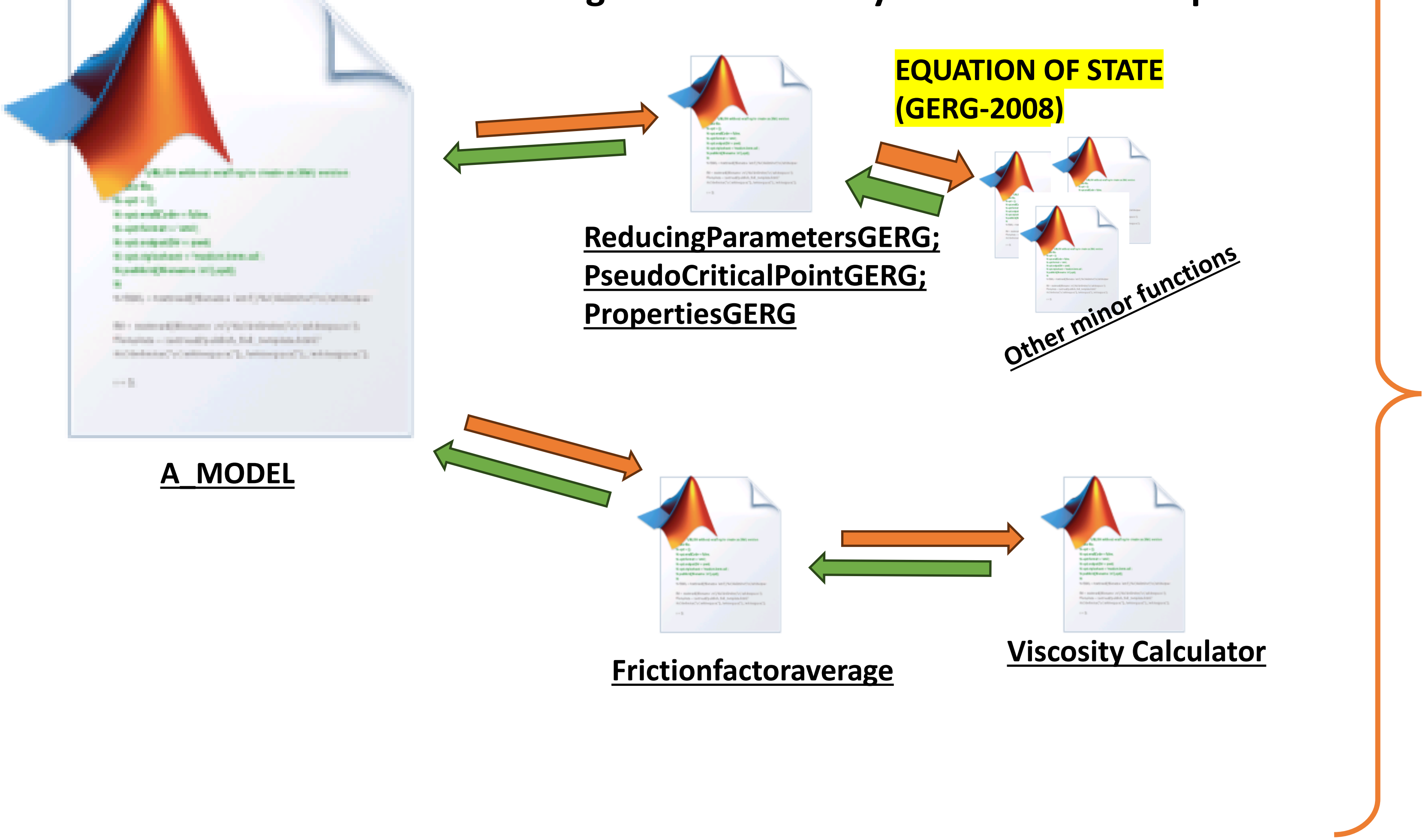


The calculation of the friction factor for each pipeline section is done calling an outside function. As it is funciton of the Reynolds Number that, in turn, in funciton of the viscosity calculator, also the viscosity is calculated through an external function as it is variable with the composition and the temperature.

The correlations for the friction factor are many and as for the equation of state, we could decide if setting up different choice.
In any case this friction factor calculator is quite light computationally speaking

Matlab Functions Management

Getting into the fluid-dynamic simulation phase:



This is the part of the program that is nested in 2/3 cycle:

1) **TIME**: if the simulation is time-dependent, any consequent instant of time is solved starting from the solution of the previous timestep (it is the «ii» cycle with reference to the matlab script)

2) **COMPOSITION**
it is cycled util the gas quality problem reaches a convergence: this to take into account the impact of the change of the gas composition on the fluid dynamic
TO BE CHECKED + TO BE CHECKED IF THIS CYCLE IS MEANINGFUL
AT THE MOMENT IS NOT INCLUDED.

3) **Convergence of the linearization of the fluid-dynamic problem**
this is the «k» cycle (with reference to the matlab script) where all what has been discussed and will be described is cycled.

Deep dive into the fluid-dynamic equations

Transient Gas Network Model

A closer look:

Coninutiy equation

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

$G = \rho v A$

$\rho = \frac{p}{ZRT}$

$c^2 = ZRT$

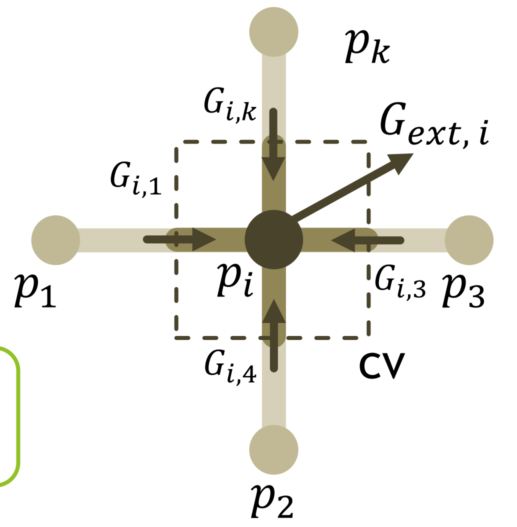
$$\frac{\partial p}{dt} = -\frac{c^2}{A}\frac{\partial G}{\partial x}$$

Control Volume integration

$$\frac{V_i}{c_{i,j}^2}\frac{dp_i}{dt} = \sum_{j=1}^{k} G_{i,j} - G_{ext,i}$$

time discretization

$$\frac{V_i}{c_{i,j}^2 \Delta t}\left(p_i^{n+1} - p_i^{n}\right) = \sum_{j=1}^{k} G_{i,j}^{n+1} - G_{ext,i}^{n+1}$$



Flow equation

$$\frac{\partial(\rho v)}{\partial t} + \cancel{\frac{\partial(\rho v^2)}{\partial x}} + \frac{\partial p}{\partial x} + \frac{\lambda}{2D}\rho v|v| + \cancel{\rho g \sin\alpha} = 0$$

creeping motion *horizontal pipelines*

$P = p^2$

$$\frac{\partial p}{\partial x} = -\frac{\lambda c^2}{2DA^2 p}|G|G - \frac{1}{A}\frac{\partial G}{\partial t}$$

Finite Difference over a pipeline

time discretization

LINEARIZATION NEEDED!

$$\Delta P_j^{n+1} = R_{f_j}\left|G_j^{n+1}\right|G_j^{n+1} + R_{I_j}\left(G_j^{n+1} - G_j^{n}\right)$$

with:

Fluid-dynamic Resistance $R_f = \frac{16\,\lambda c^2}{\pi^2 D^5 p}\Delta x$

Inertia Resistance $R_I = \frac{2\,p_m}{A}\frac{\Delta x}{\Delta t}$

Averaged over the pipeline length

- Friction factor λ → Colebrook-White correlation
- Compressibility Factor Z → GERG-2008 wide range equation of state

Detail of momentum eq. with gravitational term

$G = \rho v A$ $\quad \rho = \frac{p}{ZRT}$ $\quad c^2 = ZRT$

Flow equation

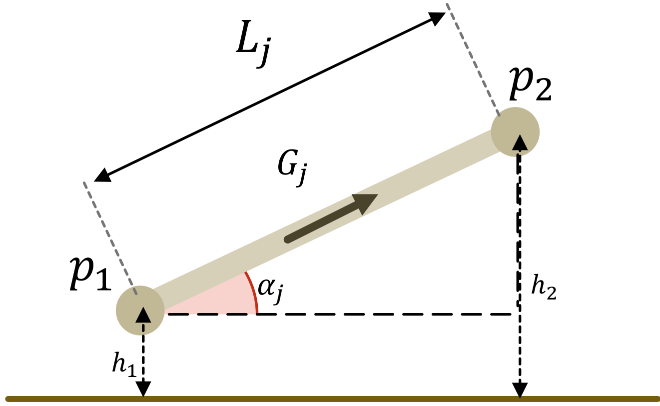
$$\frac{\partial(\rho v)}{\partial t} + \cancel{\frac{\partial(\rho v^2)}{\partial x}} + \frac{\partial p}{\partial x} + \frac{\lambda}{2D}\rho v|v| + \rho g \sin\alpha = 0$$

creeping motion

$$\frac{\partial p}{\partial x} + \frac{g \sin\alpha}{c^2} p = -\frac{\lambda c^2}{2DA^2 p}|G|G - \frac{1}{A}\frac{\partial G}{\partial t}$$

$P = p^2$

$$\frac{\partial P}{\partial x} + \frac{2g\sin\alpha}{c^2} P = -\frac{2\,p}{A}\frac{\partial G}{\partial t} - \frac{\lambda c^2}{DA^2} G|G|$$



integration of the spatial derivative:

linear and non-homogeneous differential equation

$$P_{in} - P_{out}e^{s_j} = \frac{2\,\bar{p}_j\, l_{e_j}}{A_j}\frac{\partial G_j}{\partial t} + \frac{\lambda_j \overline{c^2}_j\, l_{e_j}}{D_j A_j^2} G_j|G_j|$$

Effective length l_{e_j}

$$l_{e_j} = \begin{cases} l_j\,, & h_{in} = h_{out} \\ \dfrac{e^{s_j}-1}{s_j}\, l_j\,, & h_{in} \neq h_{out} \end{cases} \qquad s_j = \frac{2g(h_{out} - h_{in})}{\overline{c^2}_j}$$

Averaged quantities which makes possible the analytical solution

average pressure

$$\bar{p} = \frac{p_{in}^2 + p_{in}p_{out} + p_{out}^2}{p_{in} + p_{out}}$$

average speed of Sound

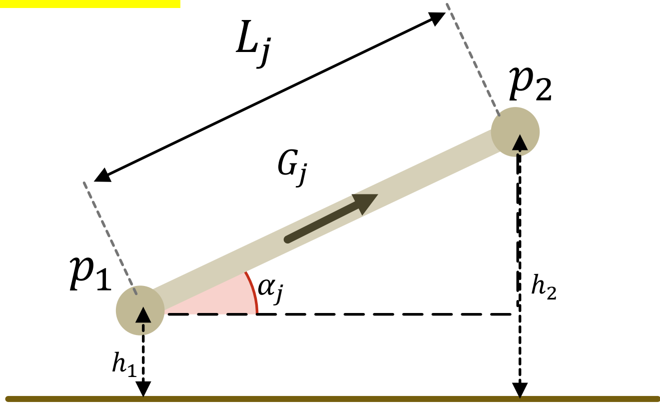
$$\overline{c^2} = Z(\bar{p}, T, [y])RT$$

Gas composition

Detail of momentum eq. with gravitational term

$$P_{in} - P_{out} e^{s_j} = \frac{2\,\bar{p}_j\, l_{e_j}}{A_j}\frac{\partial G_j}{\partial t} + \frac{\lambda_j\, \overline{c^2}_j\, l_{e_j}}{D_j A_j^2} G_j|G_j|$$

Ordinary differential equation in which time derivative (inertia term) can be trated by means of **implicit finite different scheme** (over each pipe)



$$\boldsymbol{P_{in}^{t+1} - P_{out}^{t+1} e^{s} = \frac{2\,\overline{p^{t+1}}\, l_e}{A\,\Delta t}\left(G^{t+1} - G^{t}\right) + \frac{\lambda\, \overline{c^2}\, l_e}{D A^2} G^{t+1}\left|G^{t+1}\right|}$$

$$\Delta P_j^{n+1} = R_{f_j}\left|G_j^{n+1}\right|G_j^{n+1} + R_{I_j}\left(G_j^{n+1} - G_j^{n}\right)$$

LINEARIZATION NEEDED!

with:

Fluid-dynamic Resistance $R_f = \frac{16\,\lambda c^2}{\pi^2 D^5 p}\Delta x$

Inertia Resistance $R_I = \frac{2\,p_m}{A}\frac{\Delta x}{\Delta t}$

Averaged over the pipeline length

For a generic pipe j, assuming the linearization point as $\left(\Delta P_j^{t+1}, G_j^{t+1}\right)$, the application of the linearization formula:

$$\Delta P_j^{t+1\,(k+1)} - \Delta P_j^{t+1\,(k)} = \left.\frac{d\Delta P_j^{t+1\,(k)}}{d\,G_j^{t+1\,(k)}}\right|^{(k)} \left(G_j^{t+1\,(t+1)} - G_j^{t+1\,(k)}\right)$$

LINEARIZATION FORMULA

leads to the following expression:

$$\boldsymbol{\Delta P_j^{t+1\,(k+1)} - \underbrace{\left(2R_F \cdot \left|G_j^{t+1\,(k)}\right| + R_I\right)}_{R} G_j^{t+1\,(k+1)} = -R_F \cdot \left|G_j^{t+1\,(k)}\right| G_j^{t+1\,(k)} - R_I G_j^{t\,(k)}}$$

Detail of momentum eq. with gravitational term

$$\Delta P_j^{t+1\,(k+1)} - \left(2R_F \cdot \left|G_j{}^{t+1\,(k)}\right| + R_I\right) G_j{}^{t+1\,(k+1)} = -R_F \cdot \left|G_j{}^{t+1\,(k)}\right| G_j{}^{t+1\,(k)} - R_I G_j{}^{t\,(k)}$$

R

$$\Delta P_j^{n+1} = P_{in}^{t+1} - P_{out}^{t+1} e^s = p_{in}^2{}^{t+1} - p_{out}^2{}^{t+1} e^s$$

Dividing by:

$$p_{in} + p_{out} e^{s/2}$$

$$\Delta p_j^{n+1} = p_{in}^{t+1} - p_{out}^{t+1} e^{s/2}$$

R'

$$\frac{\Delta P_j^{t+1\,(k+1)}}{p_{in}^{t+1} + p_{out}^{t+1} e^{s/2}} - \frac{\left(2R_F \cdot \left|G_j{}^{t+1\,(k)}\right| + R_I\right)}{p_{in}^{t+1} + p_{out}^{t+1} e^{s/2}} G_j{}^{t+1\,(k+1)} = \frac{-R_F \cdot \left|G_j{}^{t+1\,(k)}\right|}{p_{in}^{t+1} + p_{out}^{t+1} e^{s/2}} G_j{}^{t+1\,(k)} \frac{- R_I G_j{}^{t\,(k)}}{p_{in}^{t+1} + p_{out}^{t+1} e^{s/2}}$$

$R_F{}'$ $R_I{}'$

Matrix construction (momentum eq)

Pipeline linearized equation

$$\mathbf{A_g^t}\,\boldsymbol{p}^{t+1\,(k+1)} - \mathbf{R}\,\boldsymbol{G}^{t+1\,(k+1)} = -\,\mathbf{R_F}\left(\left|\boldsymbol{G}^{t+1\,(k)}\right| \circ \boldsymbol{G}^{t+1\,(k)}\right) - \mathbf{R_I}\,\boldsymbol{G}^{t\,(k)}$$

$\circ$ ← element by element product

Modified incidence matrix

to take into account the gravitational terms (referred to the squared pressures)

$$\mathbf{A_g} = [a_{g_{i,j}}]^{n\times b},\ a_{g_{i,j}} = \begin{cases} +1, & \textit{node i is the inlet of edge j} \\ -e^{s_j}, & \textit{node i is the outlet of edge j} \\ 0, & \textit{node i and edge j have no connections} \end{cases}$$

Pipeline linearized equation

$$\mathbf{A'^t_g}\,\underline{\boldsymbol{p}^{t+1\,(k+1)}} - \mathbf{R'}\,\underline{\boldsymbol{G}^{t+1\,(k+1)}} = -\,\mathbf{R'_F}\left(\left|\boldsymbol{G}^{t+1\,(k)}\right| \circ \boldsymbol{G}^{t+1\,(k)}\right) - \mathbf{R'_I}\,\boldsymbol{G}^{t\,(k)}$$

unknowns

Modified incidence matrix

to take into account the gravitational terms (**referred to the pressures**)

$$\mathbf{A'_g} = [a'_{g_{i,j}}]^{n\times b},\ a'_{g_{i,j}} = \begin{cases} +1, & \textit{node i is the inlet of edge j} \\ -e^{s_j/2}, & \textit{node i is the outlet of edge j} \\ 0, & \textit{node i and edge j have no connections} \end{cases}$$

$$\mathbf{R}'_F = \frac{\boldsymbol{R_F}}{|\boldsymbol{A}'_g| \cdot\ \boldsymbol{p}^{\,\mathrm{t+1},(k)}}$$

$$\mathbf{R}'_I = \frac{\boldsymbol{R_I}}{|\boldsymbol{A}'_g| \cdot\ \boldsymbol{p}^{\,\mathrm{t+1},(k)}}$$

Element by element division

$$\mathbf{R}' = 2\boldsymbol{R}'_{\boldsymbol{F}} \circ \left|\boldsymbol{G}^{t+1\,(k)}\right| + \boldsymbol{R}'_{\boldsymbol{I}}$$

$$\mathbf{abs}(\mathbf{A'_g}) = |\boldsymbol{A}'_g| = [a'_{g_{i,j}}]^{n\times b},\ a'_{g_{i,j}} = \begin{cases} +1, & \textit{node i is the inlet of edge j} \\ +e^{s_j/2}, & \textit{node i is the outlet of edge j} \\ 0, & \textit{node i and edge j have no connections} \end{cases}$$

Matrix construction (momentum eq)

Pipeline linearized equation

unknowns

R_F , R_I diagonal matrices

$$A'^{t}_{g}\, p^{t+1\,(k+1)} - \mathrm{R}'\, G^{t+1\,(k+1)} = -\,\mathrm{R}'_{\mathrm{F}}\left(\left|G^{t+1\,(k)}\right| \circ G^{t+1\,(k)}\right) - \mathrm{R}'_{\mathrm{I}}\, G^{t\,(k)}$$

b x n

n x 1

b x b

b x 1

b x b

b x 1

b x b

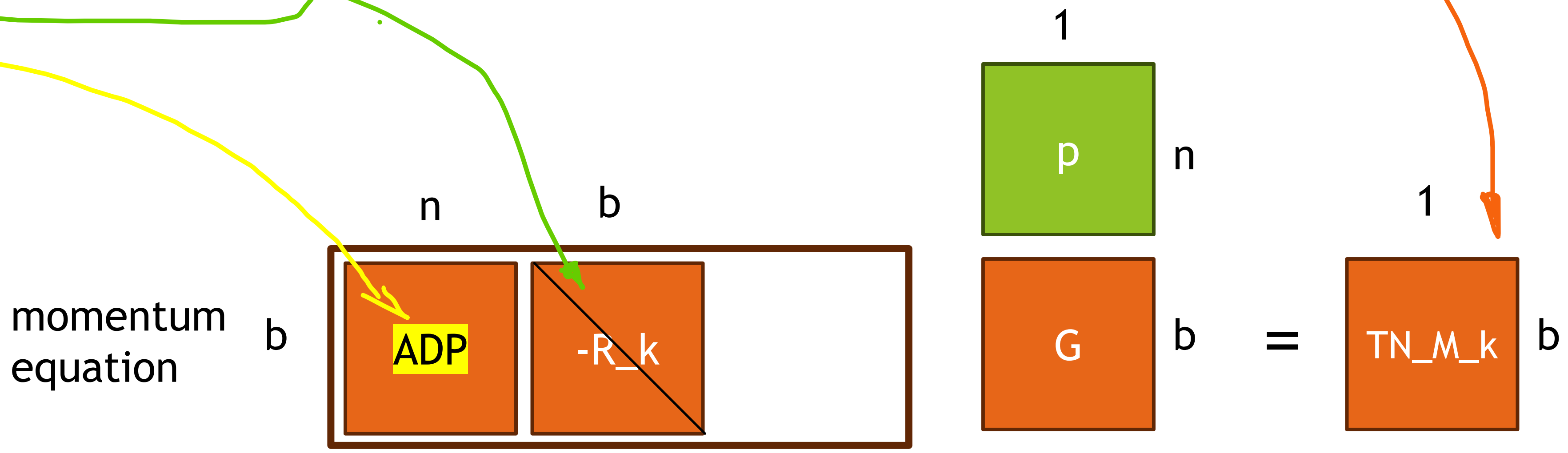
b x 1

$$\mathrm{R}'_F = \frac{R_F}{\left|A'_g\right| \cdot\ p^{\,\mathrm{t+1},(k)}}$$

$$\mathrm{R}'_I = \frac{R_I}{\left|A'_g\right| \cdot\ p^{\,\mathrm{t+1},(k)}}$$

Element by element division

$$\mathrm{R}' = 2R'_F \circ \left|G^{t+1\,(k)}\right| + R'_I$$



Matrix construction (continuity eq)

$$\frac{V_i}{c_{i,j}^2 \Delta t}(p_i^{n+1} - p_i^n) = \sum_{j=1}^{k} G_{i,j}^{n+1} - G_{ext,i}^{n+1}$$

$$\frac{V_i}{c_{i,j}^2 \Delta t} p_i^{n+1} - \sum_{j=1}^{k} G_{i,j}^{n+1} - G_{ext,i}^{n+1} = \frac{V_i}{c_{i,j}^2 \Delta t} p_i^n$$

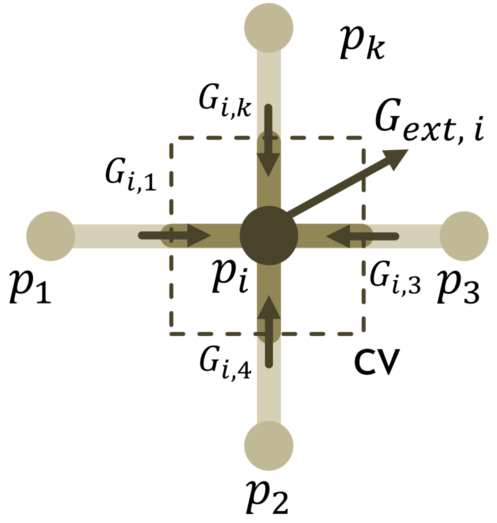
unknowns

diagonal matrix

$$\boldsymbol{\Phi} = \left[\phi_{i,i}\right]^{n\times n}, \phi_{i,i} = \frac{V_i}{c_i^2\ \Delta t}$$

n x 1 n x 1

$$\underbrace{\boldsymbol{\Phi p}^{t+1}}_{\text{n x 1}} + \mathbf{A}\ \underbrace{\boldsymbol{G}^{t+1}}_{\text{b x 1}} + \mathbf{I}\ \boldsymbol{G_{ext}}^{t+1} = \boldsymbol{\Phi p}^{t}$$



In this paper here: https://doi.org/10.1016/j.jngse.2015.11.036
They use this methodology for the boundary conditions:

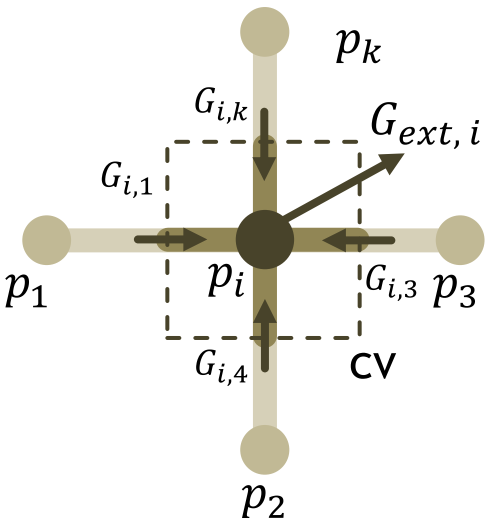
$$\boldsymbol{\Phi p}^{t+1} + \mathbf{A}\ \boldsymbol{G}^{t+1} = \boldsymbol{\Phi p}^{t} - \mathbf{0.5}\,(\boldsymbol{G_{ext}}^{t} + \boldsymbol{G_{ext}}^{t+1})$$

When I tried to implement this I was obtaining oscillating behaviours.

Matrix construction (continuity eq)

$$\frac{V_i}{c_{i,j}^2 \Delta t}(p_i^{n+1} - p_i^n) = \sum_{j=1}^{k} G_{i,j}^{n+1} - G_{ext,i}^{n+1}$$

$$\frac{V_i}{c_{i,j}^2 \Delta t} p_i^{n+1} - \sum_{j=1}^{k} G_{i,j}^{n+1} - G_{ext,i}^{n+1} = \frac{V_i}{c_{i,j}^2 \Delta t} p_i^n$$



unknowns

diagonal matrix

$$\boldsymbol{\Phi} = [\phi_{i,i}]^{n\times n}, \phi_{i,i} = \frac{V_i}{c_i^2\,\Delta t}$$

n x b

n x 1

n x 1

$$\boldsymbol{\Phi p}^{t+1} + \mathbf{A}\,\boldsymbol{G}^{t+1} + \mathbf{I}\,\boldsymbol{G_{ext}}^{t+1} = \boldsymbol{\Phi p}^{t}$$

n x n

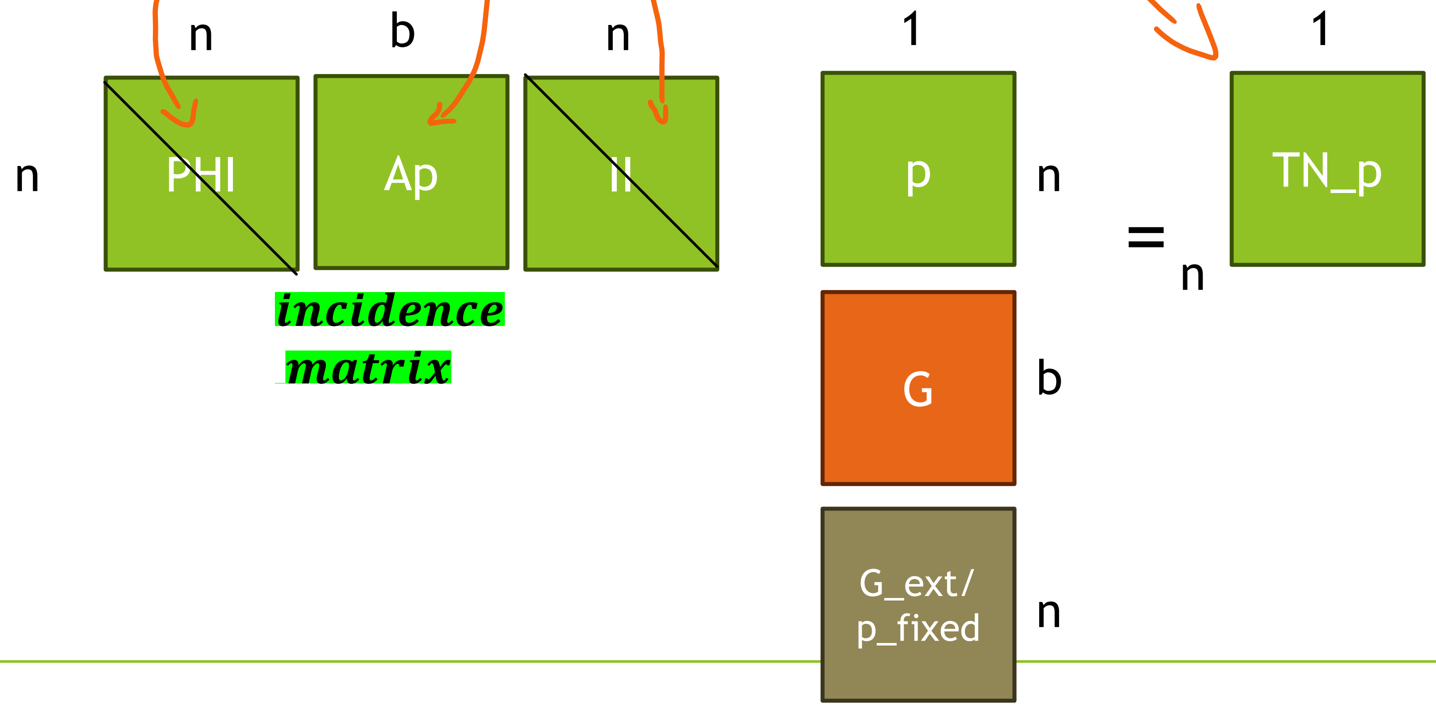
n x 1

b x 1

n x n

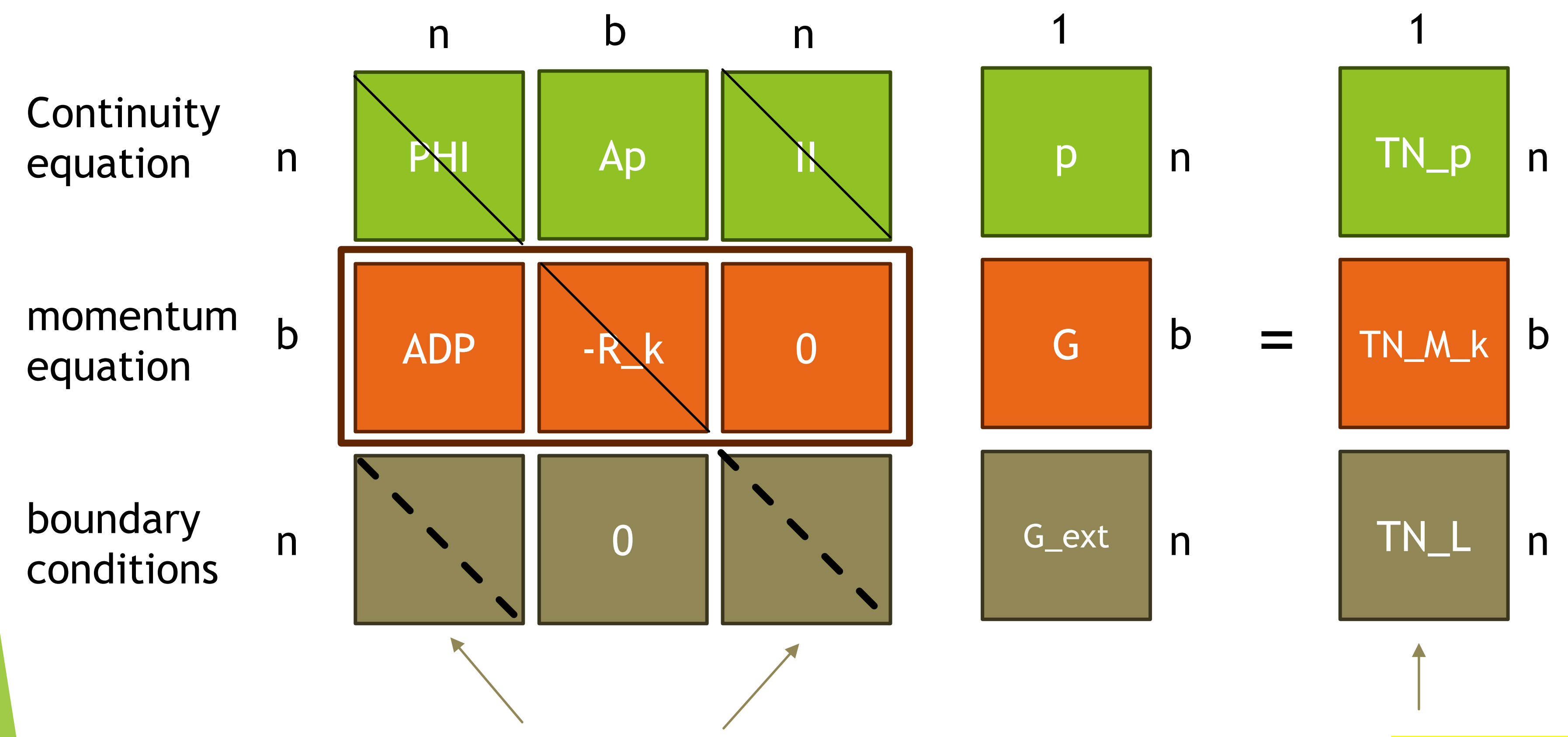
n x n

Continuity equation



incidence matrix

Matrix construction



Diagonal matrix

From matlab code

```
TN_P=PHI*p_n;
%TN_P=PHI*p_n-II./2*L_0(:,ii-1);

TN_M_k=(-Rf.*abs(G_k(:,k)).*G_k(:,k) -Ri.*G_n);

TN_L=G_ext_t(:,ii);
TN_L(1)=p_in_t(ii)
```

1 or 0 on the diagonals to fix
PRESSURE OR MASS FLOW
at the designated nodes

Assigned **pressure or mass flow** (either inlet or outlet) accrodingly

Gas Network Model - overview of the fluid-dynamic solver

Solving Strategy:

p^0_{inlet} ; $\boldsymbol{G^0_{ext}}$

initial [t=0]
steady state solution

Equation of State

updates the fluid properties

Coninutiy equation

$$\frac{V_i}{c^2_{i,j}\Delta t}(p_i^{n+1} - p_i^n) = \sum_{j=1}^{k} G_{i,j}^{n+1} - G_{ext,i}^{n+1}$$

Flow equation

LINEARIZED EQUATION

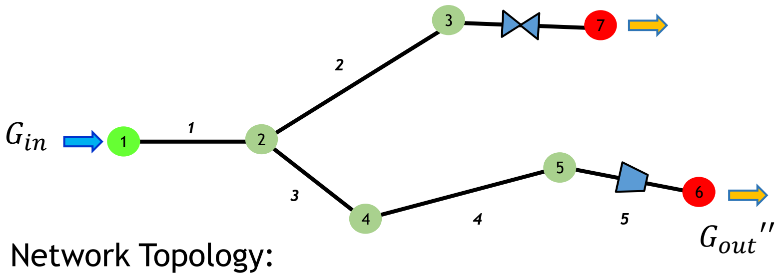
$$\Delta P_j^{(k+1),n+1} - \left(2R_{f_j}\left|G_j^{(k),n+1}\right| + R_{I_j}\right) G_j^{(k+1),n+1}$$
$$= -R_{f_j}\left|G_j^{(k),n+1}\right| G_j^{(k),n+1} - R_{I_j} G_j^n$$

Coupled fluid-dynamic problem

$$\Delta P_j^{n+1} - \left[R_{f_j}|G_j^{n+1}|G_j^{n+1} + R_{I_j}(G_j^{n+1} - G_j^n)\right] < toll$$

no

Networkwide:



Network Topology:

Incidence Matrix
representation of the network

+1 inlet node
-1 outled node
Reading column-by-column

branches

$$\boldsymbol{X} = \begin{array}{c} \\ \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \\ \mathbf{6} \\ \mathbf{7} \end{array} \begin{array}{c} \begin{array}{cccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} \end{array} \\ \begin{bmatrix} +1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \end{array}$$

nodes

Repeated for
each Timestep

Quality Tracking section

For the moment I am stopping here: in the following slide I gave to you just the overview of the method used. I need to revise more carefully the different version I created.
In any case I always found it hard to model this “batch method” and I would appreciate if we could work together, even starting from almost scratch, in order to create a tool that is much more efficient than the one I used to have.

After all, I think that the application of the batch method is, at least for the case of each pipeline, a problem that can be formalized as a “queue problem”. It needs to be coupled with conservation of mass at the intersections.

Validation

Quality Tracking section:

→ Transport of the concentration vector along the pipelines

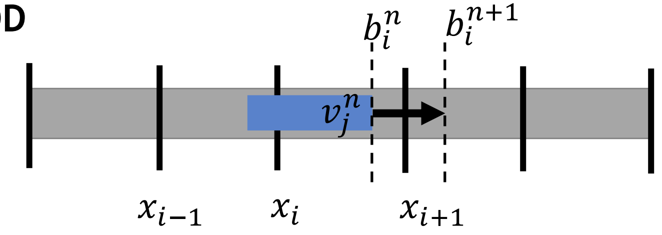
$\boxed{\text{Advective transport equation}}$

$$\frac{\partial \boldsymbol{Y}}{\partial t} + v\frac{\partial \boldsymbol{Y}}{\partial x} = 0$$

with:
- v : gas velocity;
- $\boldsymbol{Y}$: gas composition vector;

Lagrangian coordinates-based
BATCH TRACKING METHOD

$$b_i^{n+1} = b_i^n + v_j^n \Delta t$$



→ Mixing at network nodes

$\boxed{\text{Coninutiy equation}}$

for each chemical species

$$y_{c.s._i} = \left(\sum_j^{inward} G_j y_{c.s.,j} - G_{ext_i} y_{c.s.,ext_i}\right) \Bigg/ \left(\sum_j^{outward} G_j - G_{ext_i} + \frac{V_i}{c_{i,j}^2 \Delta t}\left(p_i^{n+1} - p_i^n\right)\right)$$

Solving strategy:

Convergence of fluid-dynamic problem

Coninutiy equation

applied at mixing nodes

Advective transport equation

$$b_i^{n+1} = b_i^n + v_j^n \Delta t$$

for each branch connecting two mixing nodes

$$\frac{\|Y^{new} - Y^{old}\|}{\|Y^{old}\|} < toll_c$$

no

Back to coupled fluid-dynamic problem

Repeated for
each Timestep

To be continued…