Table of Contents

[Main.m 2](#_Toc165374685)

[Velocity-driven 2](#_Toc165374686)

[generateP.m 2](#_Toc165374687)

[calculateQ.m 3](#_Toc165374688)

[calculateSR.m 4](#_Toc165374689)

[calculateVisco.m 4](#_Toc165374690)

[generateVreal.m 5](#_Toc165374691)

[validateReynolds.m 6](#_Toc165374692)

[calculateReq.m 6](#_Toc165374693)

[calculateReqError.m 6](#_Toc165374694)

[calculatePrequired.m 6](#_Toc165374695)

[tools 7](#_Toc165374696)

[validUserInput.m 7](#_Toc165374697)

[readMaterial.m 7](#_Toc165374698)

## Main.m

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% main script

% Last updated : 2021-04-02

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% Description : General script that talks to other scripts, queries the

% database and interacts with the user. Also provides user feedback.

%

% Instructions:

% This script must be used with the materialsDB.xls database.

% Follow the instructions in the Matlab console.

%

% Authors: Jean-François Chauvette, David Brzeski

% Date: 2020-05-29

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Import path of necessary folders: Velocity-driven (equations) and Tools (graphs)
2. Initialize variables
   1. debug\_mode = 1; % To print in the console all the intermediate values for calculation
   2. graph\_mode = 'P'; % Determines which graph to plot.
      1. % P = Pressure vs. Speed
      2. % V = Viscosity vs. Shear rate
      3. % S = Printing Speed vs. nozzle ID number
      4. % Q = Mass flow rate vs. printing speed
3. Multinozzle geometry
   1. Number of nozzles (alpha)
   2. Measured Diameter of Every nozzle
   3. D(2,:)= error on measured Diameter
4. V: array of desired printing speed, 50 mm/s to 250 mm/s
5. Opens the material database, retrieves rheological info
6. Initialize the size of all matrixes for plotting, based on the size of V
7. Compute overall P for desired nozzle exit speed using **generateP.m,**  with data from the materials database.
8. Compute true velocity and flow rate for nozzle true applied pressure for each P/v combination, using **generateVreal.m**.
9. Filament diameter calculation (extruded from the nozzle), using volumetric flow (Q) divided by printing speed (V)
10. Generate plot based on step b.b

## Velocity-driven

### generateP.m

function [P,eta,SR,Q,deta,dP,dRi,dSR] = generateP(rho, v, D, L, n, K, eta\_0, eta\_inf, tau\_0, lambda, a, P\_amb, debug\_mode)

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% generateP.m

% Last updated : 2020-06-13

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% This function's purpose is to regroup all the necessary function calls in

% order to calculate the required pressure for a given material and nozzle

% exit velocity. It validates the Reynold numbers. Finally, it returns the

% required Pressure along with Viscosities and Shear rates arrays for plots.

%

% Author: Jean-François Chauvette

% Date: June 13, 2020

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Uses **calculateQ.m** to get flow for every nozzle (for cylindrical nozzle)
2. Sums a) for an overall flow
3. If debuf\_mode enabled, prints flow for every nozzle and overall
4. Uses **calculateSR.m** to get shear rate for every nozzle (formula not dependant on nozzle shape)
5. Applies Rabinovitch correction on shear rate (for non-Newtonian fluid)
6. Uses **calculateVisco.m** to calculate the apparent viscosity inside every nozzle.
7. Uses **validateReynolds.m** to get properties of the flow (laminar, turbulent, position)
8. If flow is laminar
   1. Uses **calculateReq.m** to calculate equivalent hydraulic resistance as a sum of resistance in all nozzles
   2. Uses **calculateReqError.m** to get the error on the hydraulic resistance
   3. Uses **calculatePrequired.m** to get required pressures for the speed and resistance.
9. Nan of not laminar

### calculateQ.m

function [Q,dQ,Q\_eq] = calculateQ(D,v)

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%calculateQ.m is the function used to obtain the volumetric flow rate thru

%several nozzle.

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%The output is a flow rate array for each of the nozzles (Q). The inputs

%are the nozzle diameter array (D), the desired speed for all nozzles(v).

%Author: David Brzeski, Jean-François Chauvette

%Date: June 13, 2020

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if isnumeric(D) && isnumeric(v)

area = pi().\*0.25.\*D(1,:).^2;

Q = area(1,:).\*v;

dQ = pi\*0.5.\*D(1,:).\*D(2,:).\*v;

end

end

1. Makes sure input is numeric.
2. Calculates area of each nozzle with measured diameter (1D matrix of length of nozzle number)
3. Calculates volumetric flow for each requested speed (2D matrix of number of nozzle x number of required speeds)
4. Calculates error of flow, using diameter and error on diameter time speed.

### calculateSR.m

function [SR,dSR] = calculateSR(Q,D,v)

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%calculateSR.m is the function used to obtain the shear rate inside

%multiple nozzles.

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%The output is the shear rate array (SR). The inputs are the flow rate

%array (Q), the nozzle diameter array (D)

%Author: David Brzeski, Jean-François Chauvette

%Date: June 13, 2020

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if isnumeric(Q) && isnumeric(D)

SR = 32.\*Q./(pi.\*D(1,:).^3);

dSR = 8\*v.\*D(2,:)./D(1,:).^2;

end

end

1. Uses cylindrical nozzle formula to calculate shear rate inside nozzle, with flow and nozzle diameter
2. Calculates error

### calculateVisco.m

function [eta,deta] = calculateVisco(SR,n,K,eta\_inf,eta\_0,tau\_0,lambda,a,debug\_mode,dSR)

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%calculateVisco.m is the function used to obtain the apparent viscosity

%inside a every nozzle, depending on the material's bahaviour law.

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%The output is the apparent viscosity array (eta). The inputs are the shear

%rate (SR), the viscosity index (n), the consistency index (K), the

%infinite viscosity (eta\_inf), the rest-state viscosity (eta\_0), the creep

%factor (tau\_0, the relaxation time (lambda) and the Carreau model exponent (a).

%Author: David Brzeski, Jean-François Chauvette

%Date: June 13, 2020

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%for debugging purposes, the following errors are defined (later = Excel):

%Note: delta eta not yet defined for Carreau, Bingham, & Herschell-Bulkley

%(extended)models

* Calculate viscosity, using a model depending on a few parameters:
  + Sisko model *if n~=0 && K~=0 && eta\_inf~= 0 && eta\_0==0 && tau\_0==0 && lambda==0 && a==0*
  + Newtonian model *if n==1 && K==0 && eta\_inf~=0 && eta\_0==0 && tau\_0==0 && lambda==0 && a==0*
  + Ostwald-de-Waele model (pure power law) *if n~=0 && K~=0 && eta\_inf==0 && eta\_0==0 && tau\_0==0 && lambda==0 && a==0*
  + Carreau model *if n~=0 && K==0 && eta\_inf~=0 && eta\_0~=0 && tau\_0==0 && lambda~=0 && a~=0*
  + Bingham model *if n==1 && K==0 && eta\_inf~=0 && eta\_0==0 && tau\_0~=0 && lambda==0 && a==0*
  + Herschell-Bulkley extended model *if n~=0 && K~=0 && eta\_inf~=0 && eta\_0==0 && tau\_0~=0 && lambda==0 && a==0*
  + Herschell-Bulkley model *if n~=0 && K~=0 && eta\_inf==0 && eta\_0==0 && tau\_0~=0 && lambda==0 && a==0*

### generateVreal.m

function [v\_real, Q\_real,dv\_real,Q\_theo] = generateVreal(P, dP, Q, rho, v, D, L, n, K, eta\_0, eta\_inf, tau\_0, lambda, a, P\_amb, debug\_mode)

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% generateVreal.m

% Last updated : 2021-10-27

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%

%

% Author: Jean-François Chauvette, David Brzeski

% Date: June 13, 2020

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. As presented in the Article, generates real V on every nozzle by pressure equilibrium and hydraulic resistance.
2. Iterates until convergence on flow criteria (Q)
3. Uses
   1. calculateSR.m
   2. calculateVisco.m
   3. calculateReq.m
   4. calculateReqError.m

### validateReynolds.m

function [typeEcoul,Re]= validateReynolds(rho, v, D, eta, debug\_mode)

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%validateReynolds.m is the function used to validate whether a laminar flow

%is occuring in the the nozzles of the robot. Upon validation, the Hagen-

%Poiseuille viscosity formulation may be used. If any of the flow rates are

%in the transition zone or turbulent, the function returns the position of

%nozzles having non laminar flow.

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%The output is the flow type, 0=laminar, 1=transition zone, 2=turbulent.

%pos is the position array of nozzles having non laminar flow.

%Function valid only for the Extended Herschell-Bulkley, Herschell-Bulkley,

%Sisko, Ostwald-de-Waele, Bingham and Newtonian models.

%Author: David Brzeski, Jean-François Chauvette

%Date: June 13, 2020

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### calculateReq.m

function [R\_eq,Ri] = calculateReq(eta,L,D)

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%calculateReq.m is the function used to obtain the equivalent hydraulic

%resistance of several nozzles in parallel.

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%The output is the equivalent hydraulic resistance (R\_eq). The inputs are

%the apparent viscosity array (eta), the nozzle length array (L), the

%nozzle diameter array (D).

%Author: David Brzeski, Jean-François Chauvette

%Date: June 13, 2020

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Calculate equivalent hydraulic resistance, as a sum of resistance in every nozzle.
2. For conical, change of equation required\*

\*\*verify if it is only for cylindrical nozzle\*\*

### calculateReqError.m

function [ReqError,dRi] = calculateReqError(R\_eq,Ri,alpha,D,L,eta,deta)

1. Calculates error of hydraulic resistance

### calculatePrequired.m

function [P] = calculatePrequired(R\_eq,Q\_eq,P\_amb)

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%calculatePrequired.m is the function used to obtain the required pressure

%to extrude material through the equivalent flow resistance network

%caracterized by the nozzles in parallel.

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%The output is the required pressure (P).

%The inputs are the equivalent flow resistance (R\_eq), the equivalent total

%flow rate (Q\_eq) and the ambiant pressure (P\_amb).

%Author: David Brzeski, Jean-François Chauvette

%Date: June 13, 2020

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if isnumeric(R\_eq) && isnumeric(Q\_eq) && isnumeric(P\_amb)

P = (R\_eq\*Q\_eq+P\_amb);

end

## tools

### validUserInput.m

function userInput = validUserInput(sentence,type,varargin)

% validUserInput

% Last updated : 2020-06-05

%

% Description : This function ask the user a choice of answer related to

% the sentenced passed in the parameter sentence. The valid choices can be

% multiple or single and are contained in the vararagin cell. The function

% keeps asking the input if it is invalid.

%

% Inputs:

% sentence : the question asked to the user.

% type : validation type

% 0 : choice of answers (strings)

% 1 : positive numeric

% varargin : the choices of valid answers.

%

% Outputs:

% userInput : the valid user input.

%

% Author: Jean-François Chauvette

% Date: 2019-01-19

### readMaterial.m

function [rho, w, f, n, k, eta\_inf, eta\_0, tau\_0, lambda, a] = readMaterial(file, sheet)

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%readMaterial.m is the function used to extract the informations about a

%specific material from the material database

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%The output is a cell containing the material properties.

%The inputs are the database file, the sheet name (material)

%Author: David Brzeski, Jean-François Chauvette

%Date: 2020-05-25

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% Lecture des données

[~, ~, raw] = xlsread(file, sheet, 'A1:C10');

% Attribution des données aux variables

rho = raw{1,2};

w = raw{2,2};

f = raw{3,2};

n = raw{4,2};

k = raw{5,2};

eta\_inf = raw{6,2};

eta\_0 = raw{7,2};

tau\_0 = raw{8,2};

lambda = raw{9,2};

a = raw{10,2};

end