Programmes Convection

Script Calcul Rayleigh

```
clear all
close all

alpha = 6e-4;  % K-1 dilatation th
g     = 9.81;  % m.s-2 accelaration p
Cp     = 2e3;  % J.kg-1.K-1 capacité calorifique massique
rho   = 9e2;   % kg.m-3 masse vol.
k     = 0.1;   % W.m-1.K-1 cond. thermique
v     = 5e-4;  % m2.s-1   visc. cinématique

Rac = 1700;     % valeur critique minimale de Ra (pour lambda = 2)

kappa = k/rho/Cp; % diffusivité thermiqe
mu = rho*v;       % viscosité dynamique

Ttop = 20;
Tbot = 21:1:121;
DT   = Tbot-Ttop    % difference de temperature
h    = [1:1:20]*1e-2; % hauteur en m

[DTg,hg] = meshgrid(DT,h);

Ra = alpha*g/kappa/mu.*DTg.*hg.^3;

Rcrist = 6e-2; % en m
hcrist = 6e-2  % en m
volcrist = pi * Rcrist^2*hcrist;



hf = figure(1);
hold on
[~,c1] = contour(DTg,hg*100,log10(Ra));
c1.LevelList = 1:1:5;
c1.LineColor = [0 0 0]; % noir en RVB
c1.ShowText = 'on';
[~,c2] = contour(DTg,hg*100,log10(Ra));
c2.LevelList = log10(Rac);
c2.LineColor = [1 0 0]; % rouge en RVB
xlabel('Delta T [K]')
ylabel('h [cm]')
legend([c1, c2],{'Rayleigh number';'Critical value'})
hold off
```

Script calcul visc

```
clear all
close all

filename = '\Photos Vidéos\PICT0010.MOV';
full_video = VideoReader(filename);

t_beg = 17;
t_end = 18;

vid_Height = full_video.Height;
```

```
vid_Width = full_video.Width;
vid_Rate = full_video.FrameRate;
x_lim = [325 425];
y_lim = [50 475];

ind_beg = round(t_beg * vid_Rate);
ind_end = round(t_end * vid_Rate)+1;
t = 0:1/vid_Rate:(ind_end-ind_beg)/vid_Rate;

s = struct('cdata',zeros(vid_Height,vid_Width,3,'uint8'),...
    'colormap',[]);
convfact = 22.2; % distance en centimètre entre les graduation 450 et 50
k = 1;
while hasFrame(full_video)
    s(k).cdata = readFrame(full_video);
    k = k+1;
end

for i = ind_beg:ind_end
    k = i-ind_beg+1;
    im = s(i).cdata;
    hf(k) = figure(k);
    hold on
        axis tight
        axis ij
        axis equal
        axis ([x_lim y_lim])
        image(im)
        if k == 1
            [x_orig,y_orig,~] = ginput(1); %cliquer sur 450
            [x_base,y_base,~] = ginput(1); % cliquer sur 50
            [x(k),y(k),~] = ginput(1);
            plot(x_orig,y_orig,'b+')
            plot(x_base,y_base,'g+')
            plot(x(k),y(k),'r+')
        else
            [x(k),y(k),~] = ginput(1);
            plot(x(k),y(k),'r+')

        end

    hold off

end
d = ((y-y_base)*22.2)/(y_orig-y_base);
    plot(t,d,'ko')

% hf_1 = figure(1);
% hold on
% subplot(1,2,1)
%     hold on
%         axis ij
%         axis tight
%         image(s(ind_beg).cdata)
%     hold off
% subplot(1,2,2)
%     hold on
%         axis ij
%         axis tight
%         image(s(ind_end).cdata)
%     hold off
% hold off
```

Script capteur T dallas

```
// First we include the libraries
#include <OneWire.h>
#include <DallasTemperature.h>
/********************************************************************/
// Data wire is plugged into pin 2 on the Arduino
#define ONE_WIRE_BUS 2
/********************************************************************/
// Setup a oneWire instance to communicate with any OneWire devices
// (not just Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);
/********************************************************************/
// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);
/********************************************************************/
void setup(void)
{
 // start serial port
 Serial.begin(9600);
 // Start up the library
 sensors.begin();
}
void loop(void)
{
 // call sensors.requestTemperatures() to issue a global temperature
 // request to all devices on the bus
/********************************************************************/
 sensors.requestTemperatures(); // Send the command to get temperature
readings
/********************************************************************/
 Serial.print(sensors.getTempCByIndex(0)); // renvoie la premiere
temperature lue
 Serial.print(" , ");                      // met une virgule entre les 2
valeurs
 Serial.print(sensors.getTempCByIndex(1)); // renvoie la 2e temperature
 Serial.print(" , ");
 Serial.println(sensors.getTempCByIndex(2)); // renvoie la 3e temperature
 delay(200);
}
```

Script recupT.m

```
% pour lire des donnees de temperature de plusieurs capteurs DS18B20
% attention au format de sortie des données ici : XX.XX , YY.YY ... c'est
% dans le sketch Arduino.

clear all
close all
baud_rate = 9600;
n_capts = 3 % nombre de capteurs de temperature
instrreset % pour remettre a zero la liste des instruments connectes
```

```
s = serial('COM3', 'BaudRate', baud_rate); % crée un objet "port"
fopen(s)

i = 0;
T_C = zeros(1e4,n_capts);
t = zeros(1e4,1);
t_rec = 2000; % temps d'enregistrement en s


hf = figure(1)
hold on
axis([0 t_rec 20 120])
couleurs =[1 0 0; 0 1 0; 0 0 1];
for j = 1:n_capts % créer les courbes pour les capteurs
    an(j) =
animatedline('MarkerEdgeColor',couleurs(:,j),'LineStyle','none','Marker','+'
);
end

tic
while toc < t_rec
    i = i + 1;
    % lire la temperature sur le port
    val = fscanf(s);
    t(i) = toc;
    for j = 1:n_capts
        ind_beg = 1+(j-1)*8; % pour lire la bonne section de la reponse
        T_C(i,j) = str2num(val(ind_beg:ind_beg + 5)); % lit les 5 digits de
la temperature
        addpoints(an(j), t(i), T_C(i,j))
        drawnow
        disp(['time ',num2str(t(i))])
    end
end
hold off

T_C = T_C(1:i,:);
t = t(1:i);

fclose(s)
```