

# PSE: Congélation du sol

## Méthodes et protocoles

Alice WU  
Alvin SIA  
Delin JIA

### I. Dissolution du sel dans l'eau

#### Matériel :

Agitateur magnétique IKA  
Thermocouple National Instruments NI USB-TC01  
Bécher de 100 mL  
Barreau aimanté (avec une diamètre de 8mm, longueur de 3.5cm)  
Eau du robinet  
Sel de types variables (NaCl, KCl, sucre)  
Scotch  
Balance

#### Logiciel:

MATLAB, Excel

#### Protocole :

En connaissant l'enthalpie de dissolution de chaque sel dans l'eau, nous pouvons en déduire l'abaissement de température de l'eau. Nous commençons donc par dissoudre chacun leur tour les sels choisis dans de l'eau du robinet à la même température (à température ambiante autour 25 °C) sous une même agitation de 500 tours par minute, 5g de sel dans 100mL d'eau. Nous utilisons un thermocouple qui peut afficher la température à l'écran de l'ordinateur afin de mesurer son abaissement à l'aide de Matlab. Nous avons ensuite comparé les valeurs expérimentales d'abaissement de température pour des masses variables de sel sur Excel afin de choisir un abaissement maximal de la température de l'eau.

### II. Refroidissement du sol

#### Matériel :

Caméra thermique optris PI  
Thermocouple National Instruments NI USB-TC01  
Sable à l'humidité variable de dimension homogène (de 1% à 15%)  
Papier filtre  
KCl  
Bloc de glace (composé de l'eau du robinet)  
Moule (cylindre)  
Boite calorifugée  
Scotch

#### Logiciel:

MATLAB, Excel

#### Protocole :

Nous avons opté pour un sol modélisé avec un sable, pour pouvoir refaire les expériences et doser la quantité d'eau dans notre sol. Afin de congeler le sol, nous avons conçu un montage: Un bloc de sable, une feuille de papier filtre pour éviter que trop de sel ne pénètre dans le sol, une couche de sel et un bloc de glace.

Nous utilisons d'abord le moule pour fabriquer le bloc de sable de 300g et de glace (pour obtenir des expériences reproductibles). Pour chaque essai, nous insérons la sonde du thermocouple dans le sable avec une profondeur de 1 cm et déposons le sel de 40g sur la papier filtre. Tout le montage est calorifugé. Nous attendons 30 à 40min minutes pour chaque essai. Nous avons effectué une dizaine d'essais en faisant varier l'humidité du sable (soit la quantité d'eau ajoutée au sable sec). Toutes les données sont obtenues à l'aide de MATLAB(voir annexe 1). Enfin nous traçons la courbe de la température en fonction du temps sur Excel.

### **III. DSC**

#### Matériel :

DSC Q200

Sable

Capsule

Sable KCl

#### Protocole :

Nous avons réalisé une DSC (Differential Scanning Calorimetry) qui mesure la variation du flux thermique en fonction de la température. Les pics observés correspondent à la solidification exothermique de l'eau dans le sable.

### **IV. Mesure de la résistance du sable**

#### Matériel:

Arduino Uno

Resistance de 11,6 MOhm

Scotch

KCl

Sable

#### Logiciel:

Arduino, Matlab, Excel

#### Protocole:

En théorie, si l'eau présente aux interstices des grains de sables gèle, la résistance du milieu augmente. Nous ajoutons donc le matériel de mesure suivant au protocole II : un Arduino Uno à l'aide d'un pont diviseur de tension, on en déduit la résistance du sol. Nous insérons le sonde pour mesurer la résistance dans le sable avec une profondeur de 1 cm avec la sonde de thermocouple. A l'aide du programme écrit dans Arduino, nous pouvons obtenir les données et les analyser sur Excel (voir annexe 2).

## Annexe

### 1. Acquisition de la température (code MATLAB)

```
%% Discover Devices that Support Thermocouples
% To discover a device that supports Thermocouples, click the name of the
% device in the list in the Command window, or access the device in the
% array returned by Idaq.getDevicesI command. This example uses a NI 9213
% device. This is a 16 channel thermocouple module and is device 6 in our
% system.
devices = daq.getDevices
devices(1)

%% Add a Thermocouple Channel
% Create a session, and add an analog input channel with IThermocoupleI
% measurement type and change the IRatel to four scans per second.
%
s = daq.createSession('ni');
addAnalogInputChannel(s,'Dev1',0, 'Thermocouple');
s.Rate = 0

%% Configure Channel Properties
% Many properties are configured on channels individually. You can access
% channels through the IChannelsI property, and see a list of properties
% and possible values by using the IsetI command.
tc = s.Channels(1);
set(tc)

%%
% In this example, set the thermocouple type to K and units to Kelvin. Make
% sure you match the thermocouple type to your sensor configuration.
tc.ThermocoupleType = 'T';
tc.Units = 'Celsius';

%%
% For a quick summary of the channel type
tc

%% Start the Acquisition
% Use the IstartForegroundI command to start the acquisition.

i=1
t0=tic
for i=1:900
[data,time] = s.inputSingleScan();
temperature(i)=data;
temps(i)=toc(t0);
A(i,:)=[temps(i),temperature(i)];
disp(A(i,:));
pause(2);
end

figure
plot(temps, temperature)
xlabel('Time (secs)');
```

```
ylabel('Temperature (Celsius)');  
grid()
```

## 2. Acquisition de la résistance (code Arduino)

```
int analogPin = 0;  
  
float Vout = 0;  
float Vref = 5;  
  
float Vin = 5;  
  
float R1 = 11600000;  
float R2 = 0;  
  
float buffer = 0;  
unsigned long time;  
  
float Techant=2000;  
float nbPointMoy=150;  
  
void setup()  
{  
  Serial.begin(9600);  
  Vout = 1.0*analogRead(analogPin)/1024*Vref;  
  Serial.print("fenetre de moyennage (ms)= ");  
  Serial.println(nbPointMoy*Techant);  
  Serial.print("Techant (ms)= ");  
  Serial.println(Techant);  
}  
  
void loop()  
{  
  float Vout_new = 1.0*analogRead(analogPin)/1024*Vref;  
  
  Vout = (Vout_new+(nbPointMoy-1)*Vout)/nbPointMoy;  
  
  R2 = ( Vin/Vout -1 ) * R1;  
  
  //Serial.print("Vout: ");  
  //Serial.println(Vout);  
  
  // Serial.print("R2: ");  
  time = millis();  
  Serial.print(time);  
  Serial.print(" ");  
  
  Serial.println(R2);  
  
  delay(Techant);  
}
```