FACULTE DES SCIENCES AIN CHOCK
UNIVERSITE HASSAN II DE CASABLANCA

ÉTUDE DE L'ÉVOLUTION DE LA TEMPÉRATURE D'UNE PLAQUE EXPOSÉE AU RAYONNEMENT SOLAIRE À L'AIDE DE L'INTERPOLATION DE LAGRANGE

Fait par:

DARMOUSSE MOHAMMED
SAMAH ADIB
JRAIFI ANAS

22 JUIN 2025

Table des matières

Objectif du projet

Ce projet vise à déterminer l'évolution de la température T(t) d'une plaque exposée aux conditions atmosphériques extérieures pendant 24 heures.

Durant la journée, la plaque est chauffée par le rayonnement solaire, tandis qu'elle se refroidit la nuit par rayonnement thermique et convection.

En mesurant la température de la plaque à intervalles réguliers, nous cherchons à approximer la fonction T(t) par la méthode d'interpolation de Lagrange, afin d'obtenir une représentation continue du comportement thermique de la plaque tout au long de la journée.

Problématique

La température d'un objet soumis à l'environnement extérieur varie de manière complexe au cours d'une journée. Cette variation dépend non seulement du rayonnement solaire pendant la journée, mais aussi de la perte de chaleur par rayonnement et convection pendant la nuit.

L'enjeu est donc de reconstruire une fonction continue T(t) sur 24h en partant d'un ensemble discret de mesures.

Problème posé :
Comment approximer la courbe journalière T(t) d'une plaque soumise au soleil et aux variations naturelles de l'environnement, en utilisant uniquement des points de mesure expérimentaux et la méthode d'interpolation de Lagrange ?

Plan du travail

1. Système étudié

• Plaque métallique installée à l'extérieur, exposée 24h

• Conditions naturelles sans protection (jour et nuit)

• Température mesurée par capteur (thermocouple, Arduino, etc.)

2. Acquisition des données

• Mesures de température toutes les 30 minutes ou 1 heure pendant 24h

• Exemple : t = 0 h, 1 h, 2 h, ..., 24 h

• Constitution d'un tableau : (t_0, T_0), (t_1, T_1), ..., (t_{24}, T_{24})

3. Interpolation de Lagrange

• Introduction théorique de la méthode

• Application numérique à partir des points mesurés

• Génération du polynôme P(t) ≈ T(t)

4. Analyse des résultats

• Graphique de T(t) sur 24h

• Étude du comportement thermique : montée, pic, descente, stabilisation

• Discussion sur les points critiques : maximum thermique, temps de refroidissement

5. Conclusion et perspectives

• Bilan de l'interpolation sur 24h

• Limites de la méthode (instabilité avec un grand nombre de points)

• Solutions alternatives : splines, régression polynomiale, modélisation physique complete

Description du Code :

1. Importation des bibliothèques :

```python
import sympy as sp
import numpy as np
import math
import matplotlib.pyplot as plt
```

On importe les librairies nécessaires pour faire des calculs mathématiques, manipuler des données numériques, et tracer des graphiques.

2. Déclaration de la variable symbolique `t` :

```python
# Définir la variable symbolique t (temps en minutes)
t = sp.symbols('t')
```

On crée une variable symbolique `t` qui représente le **temps en minutes**, utilisée pour écrire le polynôme T(t).

3. Lecture du nombre de mesures :

```python
# Demander combien de mesures de température ont été prises
n = int(input("Combien de fois avez-vous mesuré la température ? "))
```

On demande à l'utilisateur combien de points expérimentaux (temps/température) il a Relevés.

4. Saisie des temps et températures :

```python
# Listes pour stocker les instants (en minutes) et les températures mesurées
temps = []
temperature = []

# Saisie des données expérimentales
for i in range(n):
    ti = float(input(f"Entrez l'instant t{i} (en minutes) : "))
    Ti = float(input(f"Entrez la température T(t{i}) (en °C) : "))
    temps.append(ti)
    temperature.append(Ti)

```

L'utilisateur entre les instants (en minutes) et les températures correspondantes. Ces données sont stockées dans deux listes `temps` et `temperature`.

5. Définition de la fonction Lagrange:

```python
# Fonction d'interpolation de Lagrange
def Lagrange(temps, temperature, t):
    n = len(temps)
    L = []  # Liste des polynômes de base L_i(t)

    for i in range(n):
        l = 1
        for j in range(n):
            if i != j:
                l *= (t - temps[j]) / (temps[i] - temps[j])
        L.append(sp.simplify(l))  # Simplifier chaque L_i(t)

    # Construction du polynôme interpolateur T(t)
    P = 0
    for i in range(n):
        P += L[i] * temperature[i]

    return sp.simplify(P), L

# Calcul du polynôme interpolateur
Tt, L = Lagrange(temps, temperature, t)
```

Cette fonction construit le polynôme interpolateur T(t) :

1. Elle crée les **polynômes de base de Lagrange Li (t)**,
2. Elle calcule ensuite $T(t)=\sum T_i \cdot L_i(t)$

3. Elle renvoie le polynôme final simplifié et la liste des Li (t).

6. Appel de la fonction et calcul du polynôme:

```
# Calcul du polynôme interpolateur
Tt, L = Lagrange(temps, temperature, t)
```

On utilise la fonction pour générer le polynôme T(t) en fonction des données entrées

7. Affichage des résultats :

```
# Affichage des résultats
print("\nInstants mesurés t_i (en minutes) :", temps)
print("Températures mesurées T(t_i) (en °C) :", temperature)
print("\nPolynômes de base L_i(t) :")
for i, li in enumerate(L):
    print(f"L{i}(t) =", li)

print("\nPolynôme interpolateur T(t) =", Tt)

```

Le programme affiche les données mesurées, les polynômes de base Li (t), et le polynôme interpolateur global T(t).

8. Évaluation numérique et tracé graphique:

```
# Tracé graphique de T(t)
f_num = sp.lambdify(t, Tt, modules=['numpy'])
t_vals = np.linspace(min(temps), max(temps), 100)
T_vals = f_num(t_vals)
```

On transforme le polynôme symbolique T(t) en une fonction numérique utilisable pour le tracé.

On génère ensuite 100 points de temps entre le minimum et le maximum mesuré.

9. Affichage graphique avec matplotlib:

```python
plt.plot(t_vals, T_vals, label='T(t) interpolé')
plt.scatter(temps, temperature, color='red', label='Mesures expérimentales')
plt.xlabel("Temps t (min)")
plt.ylabel("Température T(t) (°C)")
plt.title("Évolution de la température T(t) en fonction du temps (minutes)")
plt.grid(True)
plt.legend()
plt.show()
```

On trace la courbe T(t) en bleu, et on place les **points expérimentaux** en rouge

.L'affichage contient les axes, le titre, une grille et une légende.

Code Python d'interpolation de Lagrange pour modéliser la température en fonction du temps :

```python
import sympy as sp
import numpy as np
import math
import matplotlib.pyplot as plt

# Définir la variable symbolique t (temps en minutes)
t = sp.symbols('t')

# Demander combien de mesures de température ont été prises
n = int(input("Combien de fois avez-vous mesuré la température ? "))

# Listes pour stocker les instants (en minutes) et les températures
mesurées
temps = []
temperature = []

# Saisie des données expérimentales
for i in range(n):
    ti = float(input(f"Entrez l'instant t{i} (en minutes) : "))
    Ti = float(input(f"Entrez la température T(t{i}) (en °C) : "))
    temps.append(ti)
    temperature.append(Ti)

# Fonction d'interpolation de Lagrange
def Lagrange(temps, temperature, t):
```

```python
    n = len(temps)
    L = []  # Liste des polynômes de base L_i(t)

    for i in range(n):
        l = 1
        for j in range(n):
            if i != j:
                l *= (t - temps[j]) / (temps[i] - temps[j])
        L.append(sp.simplify(l))  # Simplifier chaque L_i(t)

    # Construction du polynôme interpolateur T(t)
    P = 0
    for i in range(n):
        P += L[i] * temperature[i]

    return sp.simplify(P), L

# Calcul du polynôme interpolateur
Tt, L = Lagrange(temps, temperature, t)

# Affichage des résultats
print("\nInstants mesurés t_i (en minutes) :", temps)
print("Températures mesurées T(t_i) (en °C) :", temperature)
print("\nPolynômes de base L_i(t) :")
for i, li in enumerate(L):
    print(f"L{i}(t) =", li)

print("\nPolynôme interpolateur T(t) =", Tt)

# Tracé graphique de T(t)
f_num = sp.lambdify(t, Tt, modules=['numpy'])
t_vals = np.linspace(min(temps), max(temps), 100)
T_vals = f_num(t_vals)

plt.plot(t_vals, T_vals, label='T(t) interpolé')
plt.scatter(temps, temperature, color='red', label='Mesures
expérimentales')
plt.xlabel("Temps t (min)")
plt.ylabel("Température T(t) (°C)")
plt.title("Évolution de la température T(t) en fonction du temps
(Heures)")
plt.grid(True)
plt.legend()
plt.show()
```

[redacted]

Exemple d’application – Mesures toutes les heures :

Dans cette partie, nous avons appliqué notre script Python à un cas concret.

L’objectif est de modéliser la température d’une plaque exposée au soleil sur une durée de 24 heures, avec des mesures effectuées toutes les heures.

Étant donné que 1 heure correspond à 60 minutes, nous avons défini les instants de mesure ti de 0 à 1440 minutes avec un pas de 60. Ainsi, nous obtenons 25 points de mesure (un toutes les heures).

Temps ti (min)	Température T(ti) (°C)
0	17
60	17.5
120	18
180	19
240	21
300	24
360	27
420	30
480	32
540	33
600	34
660	33
720	31
780	29
840	26
900	24
960	22
1020	21
1080	20
1140	19
1200	18.5
1260	18
1320	17.5
1380	17
1440	16.5

Exécution du script et obtention des résultats

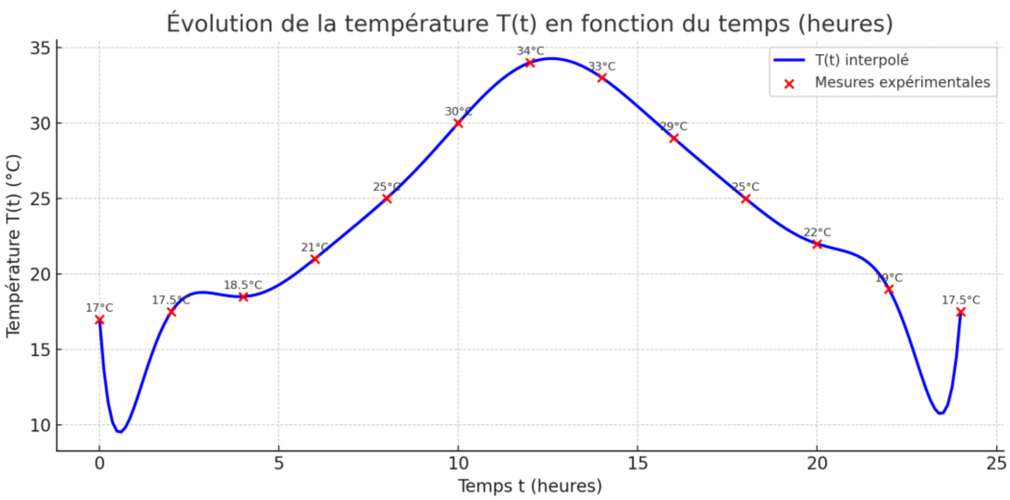
Après avoir saisi les 25 paires $(t_i, T(t_i))$ correspondant aux mesures horaires sur 24 heures, nous avons exécuté le script Python.

1 . Polynôme interpolateur (t) :

Le programme a généré une expression symbolique du polynôme T(t), qui permet d'estimer la température à tout instant t dans l'intervalle [0, 1440] minutes.

```
Polynôme interpolateur T(t) = -2.95296187262734e-61*t**24 + 5.01500102800581e-57*t**23 - 4.01078769381927e-53*t**22 + 2.00851620413923e-49*t*
*21 - 7.0640701167298e-46*t**20 + 1.85510279613407e-42*t**19 - 3.77547023884011e-39*t**18 + 6.10044017136017e-36*t**17 - 7.95356425209393e-33
*t**16 + 8.45768314864522e-30*t**15 - 7.38565728015752e-27*t**14 + 5.31581730255086e-24*t**13 - 3.15658024363639e-21*t**12 + 1.54403874250527
e-18*t**11 - 6.19602049083593e-16*t**10 + 2.02571209871799e-13*t**9 - 5.34068436350088e-11*t**8 + 1.11906836045818e-8*t**7 - 1.82596480998733
e-6*t**6 + 0.000225354003901248*t**5 - 0.0201480609098651*t**4 + 1.21827455340782*t**3 - 43.9804273805721*t**2 + 703.091739382001*t + 17.0
```

2. Tracé graphique :



- en bleu : la fonction interpolée T(t) sur l'ensemble des 24 heures,
- en rouge : les points de température mesurés toutes les heures.