Newtons_Law_of_Cooling_.py

```python
# Визначення температури об'єкта з часом при охолодженні від 400 °C до нормальних умов (20 °C)
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# Вхідні дані:
t_start, t_stop, t10, t30 = 0, 100, 10, 30  # Час початку, закінчення та для визначення температури, хв.
T0, Tn = 400, 20  # Початкова температура та температура при нормальних умовах °C.
a, step_number = 0.058, 1000  # Коефіцієнт тепловіддачі, Вт/м²/К та кількість кроків.


def sinter_ode_fun(T, t):  # Локальна функція що імплементує Закón Ньютона — Ріхмана  1 usage  ± Serhii Miroshnychenko
    return - a * (T - Tn)


t_range = np.linspace(t_start, t_stop, step_number)  # Змінна для інтервалу часу t=0 до 100 хвилин
T_sol = odeint(sinter_ode_fun, T0, t_range)  # Виклик odeint, щоб розв'язати диференціальне рівняння
T10 = T_sol[np.abs(t_range - t10).argmin()][0]  # Пошук температури через 10 хвилин
T30 = T_sol[np.abs(t_range - t30).argmin()][0]  # Пошук температури через 30 хвилин
plt.plot( *args: t_range, T_sol, 'r', label='зміна температури від часу')  # Графік охолодження
plt.scatter(t10, T10, color='red'); plt.scatter(t30, T30, color='red')
plt.text(t10 + 2, T10,  s: f'{T10:.1f} °C - температура через {t10} хв', color='red', ha='left')
plt.text(t30 + 2, T30,  s: f'{T30:.1f} °C - температура через {t30} хв', color='red', ha='left')
plt.axhline(y=T10, xmin=t_start, xmax=t10 / t_stop, color='red', linestyle='dotted')
plt.axvline(x=t10, ymin=t_start, ymax=(T10 - Tn) / (T0 - Tn), color='red', linestyle='dotted')
plt.axhline(y=T30, xmin=0, xmax=t30 / t_stop, color='red', linestyle='dotted')
plt.axvline(x=t30, ymin=0, ymax=0.01 + (T30 - Tn) / (T0 - Tn), color='red', linestyle='dotted')
plt.title("Охолодження об'єкта (Python)"); plt.legend()  # Опис та налаштування графіка
plt.xlabel('час (хвилини)'); plt.ylabel('температура (°C)'); plt.xlim(left=t_start, right=t_stop); plt.show()

```

Newtons_Law_of_Cooling_.m

```matlab
% Визначення температури об'єкта з часом при охолодженні від 400 C до нормальних умов (20 °C) і візуалізац
% Вхідні дані:
t_start = 0; t_stop = 100; t10 = 10; t30 = 30;  % Час початку, закінчення та для визначення температури відповідно, хв.
T0 = 400; step_number = 1000; % Початкова температура, °C та кількість кроків.

tRange = linspace(t_start, t_stop, step_number); % змінна для інтервалу часу t=0 до 100 хвилин
[tSol, TSol] = ode45(@sinterODEfun, tRange, T0); % Виклик ode45, щоб розв'язати диференціальне рівняння
[~, idx10] = min(abs(tSol - t10)); T10 = TSol(idx10); % Пошук температури через 10 хвилин
[~, idx30] = min(abs(tSol - t30)); T30 = TSol(idx30); % Пошук температури через 30 хвилин

plot(tSol, TSol, "r", 'DisplayName', 'зміна температури від часу') % Графік охолодження від 400 °C до нормальних умов
hold on; scatter(t10, T10, 'r', 'filled'); scatter(t30, T30, 'r', 'filled');
text(t10+2, T10, sprintf('%.1f °C - температура через 10 хв', T10), 'Color', 'red', 'HorizontalAlignment', 'left');
text(t30+2, T30, sprintf('%.1f °C - температура через 30 хв', T30), 'Color', 'red', 'HorizontalAlignment', 'left');
plot([t_start t10], [T10 T10], 'r--', 'LineWidth', 0.7); plot([t10 t10], [t_start T10], 'r--', 'LineWidth', 0.7);
plot([t_start t30], [T30 T30], 'r--', 'LineWidth', 0.7); plot([t30 t30], [t_start T30], 'r--', 'LineWidth', 0.7);
title("Охолодження об'єкта (Matlab)") % Опис та налаштування графіка
legend("зміна температури від часу"); xlabel("час (хвилини)"); ylabel("температура (°C)")
xlim([t_start t_stop]); ylim([t_start T0]); hold off;

function dTdt = sinterODEfun(t,T) % Локальна функція що імплементує Закón Ньютона — Ріхмана
    a = 0.058;  % Коефіцієнт тепловіддачі, Вт/м²/К.
    Tn = 20;  % Температура при нормальних умовах, °C.
    dTdt = - a * (T - Tn);
end
```