This script implements a numerical heat diffusion simulation using an explicit finite difference method based on Fourier's heat equation. It calculates heat distribution over time for a mesh.

The script is designed to handle a mesh composed of quadrilateral elements (2D). The mesh is created in Femap, and the data related to elements and nodes is exported to Python via .csv files.

Six CSV files are required:

- Three input files (element_input.csv, entry_nodes.csv, and entry_loads.csv) containing data from Femap.

- Three blank output files (element_output.csv, output_nodes.csv, and output_loads.csv), which will be populated after running the script.

Explanation of the physical phenomenon of heat transfer using an explicit finite difference method for approximations.

## 1. The Heat Equation

The heat equation describes how heat distributes over time in a material. In two dimensions, it is given by:

$$\frac{\partial T}{\partial t} = \alpha \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$

where:

- $T(x, y, t)$ is the **temperature** at a point $(x, y)$ at time $t$.
- $\alpha = \frac{k}{\rho c_p}$ is the **thermal diffusivity** of the material.
  - $k$ = thermal conductivity $(W/mK)$
  - $\rho$ = density $(kg/m^3)$
  - $c_p$ = specific heat capacity $(J/kgK)$
- $\frac{\partial T}{\partial t}$ represents how **temperature changes over time**.
- $\frac{\partial^2 T}{\partial x^2}$ and $\frac{\partial^2 T}{\partial y^2}$ represent **how temperature spreads in space**.

## 2. Discretization: Finite Difference Approximation

Since solving the equation analytically is difficult for complex geometries, we use a numerical approach.

Using finite differences, we approximate the derivatives as follows:

$$\frac{\partial T}{\partial t} \approx \frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t}$$

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i+1,j}^{n} - 2T_{i,j}^{n} + T_{i-1,j}^{n}}{\Delta x^2}$$

$$\frac{\partial^2 T}{\partial y^2} \approx \frac{T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}}{\Delta y^2}$$

**Explicit Method (Forward Time, Central Space)**

The script applies the explicit finite difference method, updating the temperature at each point using the formula:

$$T_{i,j}^{n+1} = T_{i,j}^{n} + \frac{\alpha \Delta t}{\Delta x^2} \left( T_{i+1,j}^{n} + T_{i-1,j}^{n} + T_{i,j+1}^{n} + T_{i,j-1}^{n} - 4T_{i,j}^{n} \right)$$

This means:

- The **new temperature** at a node $(i, j)$ at time step $n + 1$ depends on:
  - The **old temperature** at $(i, j)$ at time step $n$.
  - The **temperature of its four neighboring nodes**.
  - The term $4T_{i,j}$ ensures that the change is balanced.

## 3. The Fourier Number and Stability Condition

To ensure numerical stability, we define the Fourier number:

$$Fo = \frac{\alpha \Delta t}{\Delta x^2}$$

For the explicit method, stability requires:

$$Fo \leq 0.25$$

If this condition is not met, the solution will diverge (temperatures will oscillate wildly).

In the script:

- `d_xy` is $\Delta x$, the minimum node distance.
- `delta_t` is $\Delta t$, the time step.
- The function **raises an error** if `fourier >= 0.25`.