Programming Part of the last assignment.

The goal of this program is to investigate the stability properties of Backward and Forward Euler time-stepping for the parabolic problem

$$u_t - \frac{1}{50} \Delta u = 0 \quad (x,t) \in \Omega \times (0,T)$$

$$\frac{\partial u}{\partial n} (x,t) = 0 \quad \text{on } \partial \Omega \times [0,T]$$

$$u(x) = u_0(x) \quad x \in \Omega.$$ 

Here $\Omega$ is the L shaped domain of Assignment 7. We will reuse much of the code of assignment 7.
Problem (P) differs from that studied in class in that we have a homogeneous Neumann condition. Thus, the variational formulation is: Find $U(t) : [0, T] \rightarrow H'(\Omega)$ satisfying
\[(u_t, \phi) + \frac{1}{50} D(u, \phi) = 0 \quad \text{for each} \ t \in (0, T) \text{ and } \phi \in H'(\Omega).\]

As in Assignment 7, we use an approximation space $V$ of continuous piecewise linear functions defined from triangulation coming from TRIANGLE.
(Functions in \( V \), do not satisfy any boundary conditions).

For Assignment 1, you developed routines for

1. Assembling the mass matrix denoted by \( M \) in that assignment.
2. Assembling the stiffness matrix for the Galerkin form

\[
\mathbf{C}(u,v) = \int \left( \varepsilon(u) : \varepsilon(v) \right) + \int \phi \nabla u \cdot \nabla v.
\]

For this assignment, we use the stiffness matrix denoted by \( A \) coming from \( c^2 = 1/50 \) and \( q = 0 \).
For this assignment, we take a fixed time step size, $\Delta t = 1/N$ and implement Backward and Forward Euler time stepping. In both cases, consider

\[(IV)\]

$U_0(x,y) = 1 + \cos \pi x \cos 2\pi y$

and define $C^0_j$ to be the coefficient corresponding to $I_k(u_j)$, i.e. $C^0_j = U_0(x_j)$ where $x_j$ is the $j$th node of the mesh (enumerated by TRIANGLE).

The solution of (P) with initial value $(IV)$ is

$u(x,y,t) = 1 + e^{-t/(10)} \cos \pi x \cos 2\pi y$
(Check it).

In all runs, we shall compute the approximation to \( u(t=1) \),

i.e.,

\[
W^N = \sum_{i=1}^{m} c_i \phi_i \quad (h = 1/N)
\]

and report the error

\[
E_N := \| W^N - I_h u(1) \|_{L^2(\Omega)} = \sqrt{\int_{\Omega} (c^n + d) \cdot (c^n + d)}
\]

Here \( d \) is the vector of coefficients for the interpolant of \( u(1) \),

i.e., \( d_i = u(x_i, 1) \). If you compute \( \nabla W \) for \( W^N \), then report \( \infty \) for \( E_N \).
Problem 1: Run forward Euler

for \( N = 2, 4, \ldots \) and report \( \| e_N \| \)
until you reach an \( N \) where \( \| e_N \| < 0.01 \). Do this for each of the first four meshes of assignment 7. (The forward Euler recurrence is)

\[
C^{n+1} = C^n + k \, M^{-1} A \, C^n
\]

Problem 2: Repeat Problem 1 but use Backward Euler, i.e.,

\[
C^{n+1} = \left( I + k \, M^{-1} A \right)^{-1} C^n
\]

Do not use this formula!
\[ M \left( \frac{c^{n+1} - c^n}{k} \right) + A c^{n+1} = 0 \]

\[ (M + \frac{k}{\epsilon} A) c^{n+1} = MC^n \]

Algorithm:

1. Startup (compute \( M + kA \))
2. Compute \( y = MC^n \)
3. Compute \( c^{n+1} = (M + kA) \backslash y \).