Formulation of the PDE problem using finite element method

The following PDE describes the spatio-temporal evolution of ethanol concentration in the capillary assay domain:

$$
\begin{aligned}
&\frac{\partial C}{\partial t} = D\Delta C \text{ in } \Omega, \text{ for } t > 0 \\
&\frac{\partial C}{\partial S} = 0 \text{ on } \partial\Omega, \text{ for } t > 0 \\
&C = C_0 \text{ on } \Omega_c,\ C = 0 \text{ on } \Omega_p, \text{ for } t > 0
\end{aligned} \tag{1}
$$

Where D is diffusion coefficient of ethanol in water, Ω is the whole computational domain, Ω_c is the capillary domain, Ω_p is the pond domain, and $\partial\Omega$ is the surface of the whole computational domain. Nuemann natural boundary condition is applied to $\partial\Omega$, and ethanol initial concentration is set to C_0 in the capillary and to zero in the pond.

We use finite element method (FEM) to solve the PDE. Ethanol concentration in the domain Ω is approximated by $u \approx C(x, y, z, t)$, which is also referred to as a *trial function* in FEM context.

Since u varies with time, the straightforward approach is to first discretize the time derivative by a finite difference approximation, which will yield a set of stationary finite element problems. For simplicity and stability, we use Euler's implicit difference:

$$\frac{\partial u^k}{\partial t} \approx \frac{u^k - u^{k-1}}{\Delta t} = D\Delta u^k \tag{2}$$

Rearranging both sides in equation (2) and applying the initial condition I will result in equation (3):

$$
\begin{aligned}
&u^0 = I \\
&u^k - \Delta t D \Delta u^k = u^{k-1}\ , k = 1, 2, ...
\end{aligned} \tag{3}
$$

Given I, we can then solve for $u \in V$, where V is a functional space.

In order to solve for u, we then turn the problem into variational (weak) form by multiplying equation (3) by a *test function* $\nu \in \hat{V}$ where $\hat{V}$ is a functional space, and integrate the equations over Ω:

$$
\begin{aligned}
&\textstyle\int_\Omega u^0 \nu dx = \int_\Omega I \nu dx \\
&\textstyle\int_\Omega (u^k - \Delta t D \Delta u^k) \nu dx = \int_\Omega u^{k-1} \nu dx
\end{aligned} \tag{4}
$$

We then integrate the second derivative by parts and apply the boundary condition $\frac{\partial u}{\partial S} = 0$, which will result in the following equation:

$$\int_\Omega (u^k \nu + \Delta t D \nabla u^k . \nabla \nu) dx = \int_\Omega u^{k-1} \nu dx \tag{5}$$

By introducing u for u^k, we can write equation (5) in the standard notation: $a(u, \nu) = L(\nu)$:

$$
\begin{aligned}
&a_0(u, \nu) = \int_\Omega u\nu dx \\
&L_0(\nu) = \int_\Omega I\nu dx \\
&a(u, \nu) = \int_\Omega (u\nu + \Delta t D \nabla u . \nabla \nu) dx \\
&L(\nu) = \int_\Omega u^{k-1}\nu dx
\end{aligned}
\tag{6}
$$

The goal of the continuous variational problem is to find $u^0 \in V$, such that $a_0(u^0, \nu) = L_0(\nu)$ holds for all $\nu \in \hat{V}$, and to find $u^k \in V$, such that $a(u^k, \nu) = L(\nu)$ holds for all $\nu \in \hat{V}$, $k = 1, 2, ...$

Approximate continuous solutions are then calculated by restricting V and $\hat{V}$ functional spaces into finite elements in the mesh using linear Continuous Galerkin method.

A custom Python script `cap_assay.py` is then generated to employ FEniCS computational platform to set up the variational problem and solve it for the solutions. The script is available in `scripts/` directory.