Homework 6

In this homework you will solve two different sets of equations in two spatial dimensions.

- **2D Reaction–Diffusion equation:** You will solve the equation

$$\partial_t c - D(\partial_x^2 c + \partial_y^2 c) = c(1-c). \tag{1}$$

  To do this, you will write a class `ReactionDiffusion2D` in `equations.py`. This class will be passed a `FieldSystem` containing the field c, a diffusivity D, and the second derivative operators in the x and y directions, `dcdx2` and `dcdy2`, respectively.

  The class will have a `step` method which will take a timestep of size dt. You should use operator splitting so the diffusion terms are treated implicitly. Your timestepping method should be second order accurate. For the diffusion terms, use the `CrankNicolson` scheme, and for the reaction term use the `PredictorCorrector` scheme.

  In addition to taking a timestep, the `ReactionDiffusion2D` class should also store `self.t` and `self.iter` attributes which update every timestep. This will be used to decide when the simulation is over.

- **2D Viscous Burgers equation:** You will solve the equations

$$\partial_t u + u\partial_x u + v\partial_y u - \nu(\partial_x^2 + \partial_y^2)u = 0, \tag{2}$$
$$\partial_t v + u\partial_x v + v\partial_y v - \nu(\partial_x^2 + \partial_y^2)v = 0. \tag{3}$$

  As in the Reaction–Diffusion problem, you will be writing a class `ViscousBurgers2D` in `equations.py`. This class will be passed a `FieldSystem` $X = (u, v)$, the viscosity ν, and the convergence order of the spatial derivatives. In this problem, you will need to make the appropriate spatial derivative operators that you need to solve the equation. As above, your algorithm should use `Crank-Nicolson` for the diffusion terms and the `PredictorCorrector` scheme for the advection terms. Your algorithm should be second order accurate in time. You will also need to update and store `self.t` and `self.iter` in the class.