AM225: Assignment 5 (due 5 PM, April 30)

1. **Simulating paper marbling.** Figure 1 shows an example of paper marbling, a type of craft activity for making colorful swirling designs. Paper marbling is done by filling a shallow tray with water, adding layers of oil-based paints to the surface, and then using the water motion to mix and deform the paint colors. After this, a sheet of paper is laid on top of the tray, to transfer the paint pattern. Here are some Youtube videos showing paper marbling in action: A, B, C, D. **The goal of this question is to produce your own paper marbling pattern, by extending the incompressible fluid simulation that was introduced in the lectures.** The end result should roughly resemble Fig. 1.

   As discussed the lectures, the simulation code in `am225_examples/5a_fluid_sim` simulates the Navier–Stokes equations

   $$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} \tag{1}$$

   subject to the incompressibility constraint

   $$\nabla \cdot \mathbf{u} = 0. \tag{2}$$

   Here $\mathbf{u}(\mathbf{x}, t)$ is the velocity, $p(\mathbf{x}, t)$ is the pressure, ρ is fluid density, and μ is the dynamic viscosity. To begin, extend the fluid simulation code so that it also simulates a three-component vector field $\mathbf{c}(\mathbf{x}, t) = (R(\mathbf{x}, t), G(\mathbf{x}, t), B(\mathbf{x}, t))$ representing the red, green, and blue color channels. The field values should be stored at the grid cell centers, in the same position as the velocity. Use the convention that 1 represents the maximum color channel value.[1] The vector field should satisfy the hyperbolic conservation law[2]

   $$\frac{\partial \mathbf{c}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{c} = \mathbf{0}. \tag{3}$$

   You can use any appropriate method to simulate Eq. 3 but one good choice is to use an explicit Euler step for the time derivative, and the upwinded ENO method for the spatial derivative. The fluid simulation code already contains routines to perform the ENO derivative for evaluating the $(\mathbf{u} \cdot \nabla)\mathbf{u}$ term appearing in Eq. 1, and thus you can co-opt this routine for your purposes.

   To visualize your results, an example program `wpng_example.cc` is provided that will take a three-component 2D array and convert it to a PNG image.[3]

   (a) As a warm-up and validation of your code, consider the domain $[-1, 1]^2$ with a 256×256 grid using non-periodic boundary conditions and the default initial velocity in the code. Use the initial condition

   $$\mathbf{c}(\mathbf{x}, t) = \begin{cases} (1, 1, 1) & \text{if } \lfloor 6x + 6 \rfloor + \lfloor 6y + 6 \rfloor) \text{ is even,} \\ (0.2, 0.4, 0.9) & \text{otherwise.} \end{cases} \tag{4}$$

---

[1] For example $\mathbf{c} = (1, 1, 1)$ corresponds to white.

[2] Note that since $\nabla \cdot \mathbf{u} = 0$ this is equivalent to $\partial c_j / \partial t + \nabla \cdot (c_j \mathbf{u}) = 0$ for each color channel j.

[3] As discussed in the lectures, this requires that you have libpng installed. This is usually a standard package in package management systems.

Here $\lfloor \cdot \rfloor$ is the floor function. This corresponds to a blue and white checkerboard pattern. Simulate from $t = 0$ to $t = 0.5$ and make a PNG image of the color pattern at $t = 0.5$.

(b) Simulate your own paper marbling pattern. Use a computational grid of 640×640 or larger, and either periodic or non-periodic boundary conditions. A possible approach is as follows:

- Initialize $\mathbf{c}(\mathbf{x}, 0)$ to contain a simple pattern using several colors.
- Simulate a small duration of fluid flow to deform the colors.
- Add another layer of colors to the $\mathbf{c}$ field, such as localized circles or ellipses.
- Simulate a second fluid flow to apply additional mixing.

Note that you may wish to avoid too much distortion in the fluid flow, which will blur the colors into each other. However, some mixing is inevitable. Increasing the resolution will reduce the amount of blurring. The fluid code is quite efficient, and you only need to run for a short simulation duration. Hence it is reasonable that you use a 1024×1024 grid or larger if you wish.

Chris encourages you to think about aesthetically pleasing colors and designs. Chris plans to make a montage of the submitted patterns, to be posted in the homework solutions.[4]

(c) **Optional.** If you watch the Youtube videos, you will notice that as a droplet of paint is added, it spreads out and pushes the surrouding colors away. A way to approximately model the injection of paint at a location is to change Eq. 2 to

$$\nabla \cdot \mathbf{u} = \alpha \delta(\mathbf{x} - \mathbf{x}_c) - \frac{\alpha}{A} \tag{5}$$

where δ is the Dirac delta function, α is a constant, and A is the size of the simulation domain. The second term in Eq. 5 ensures that the total amount of fluid is conserved.[5] Modify the code to simulate this process. Rather than use a perfect delta function, you should spread its influence over a small number of grid cells. The $\mathbf{c}$ field should be fixed to the injected paint color on these grid cells.

(d) **Optional.** As mentioned above you will likely observe some blurring of color boundaries, even when using the ENO method. Chris and his research group have recently been developing a simulation method called the *reference map technique*,[6] which provides one approach for mitigating this issue. Rather than advect the colors directly, we instead introduce a two-component vector field $\boldsymbol{\xi}(\mathbf{x}, t)$ called the *reference map*. This field is initialized to $\boldsymbol{\xi}(\mathbf{x}, 0) = \mathbf{x}$, and satisfies the advection equation

$$\frac{\partial \boldsymbol{\xi}}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\xi} = \mathbf{0}. \tag{6}$$

Consider simulating the deformation of a color pattern from $t = 0$ to $t = T$. The field $\boldsymbol{\xi}$ tracks the deformation of the fluid: the value of $\boldsymbol{\xi}(\mathbf{x}, T)$ gives the position in the fluid at

[4] The patterns in the montage will be anonymous. If for any reason you wish to opt out of this, please indicate this on your submitted solutions.

[5] Note that if the total amount of fluid isn't conserved, the elliptic problem is unsolvable.

[6] C. H. Rycroft, C.-H. Wu, Y. Yu, and K. Kamrin, J. Fluid Mech. **898**, A9 (2021). [Link]

$t = 0$ that is located at $\mathbf{x}$ at $t = T$. Modify the code to simulate Eq. 6. instead of Eq. 3. You will need to choose appropriate boundary conditions for $\boldsymbol{\xi}$. Once $\boldsymbol{\xi}(\mathbf{x}, T)$ is known, the color at time $t = T$ can be found using

$$\mathbf{c}(\mathbf{x}, T) = \mathbf{c}(\boldsymbol{\xi}(\mathbf{x}, T), 0). \tag{7}$$

Use Eq. 7 to find the color at time T, and compare the amount of blurring to just using a direct solve of Eq. 3.[7]

[7] At first glance, it may not be obvious why Eq. 6 will minimize the blurring when compared to Eq. 3, since both are transport equations. However, since $\boldsymbol{\xi}$ is a smooth field, it is much less susceptible to blurring than $\mathbf{c}$. Hence, this procedure results in much better defined color boundaries.

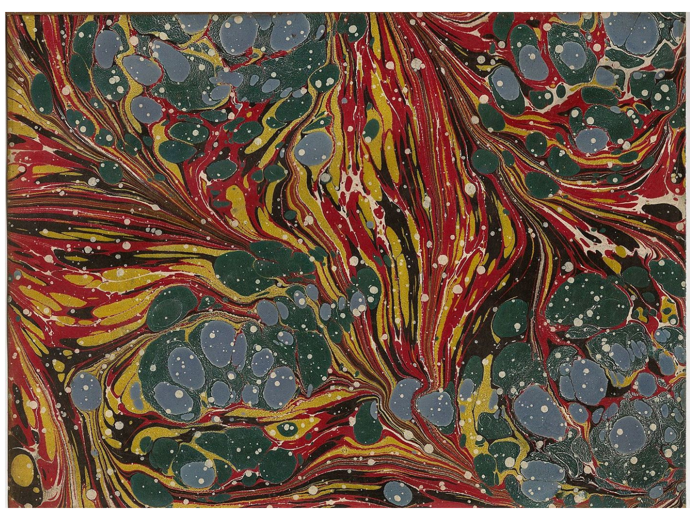

Figure 1: An example of a paper marbling pattern taken from a cover to the book *Encyclopædia Brittanica*. (Image from Wikipedia.)