Assignment 2

CASPER BRESDAHL, whs715, University of Copenhagen, Denmark

1 INTRODUCTION

This week we will first introduce the theory behind advection and mean curvature flow and how we can discretize these. We will then look at the first experiment looking into how advection without any boundary issues behave compared to advection where boundaries will play a role. We will then look at the second experiment where we will compare three different boundary conditions and how these change the end result when applying mean curvature flow to a signed distance field. At the end we will conclude on our results.

2 THEORY

2.1 Advection

Advection is the movement of some quantity via the flow of a fluid. The partial differential equation

$$\frac{\partial \phi}{\partial t} = -(\mathbf{u} \cdot \Delta)\phi \tag{1}$$

Where ϕ is a scalar field and $\mathbf{u}$ is a velocity field, is often used when dealing with advection terms. The idea when working with advection is, we treat each node in our grid as a particle, and we then trace each particle back in time and copy its value to the present time step. However, this tracing back in time is not free from issue, as we might trace a particle to outside our grid. If this happens we will have to interpolate the point back on the boundary, preferably by finding the intersection point between the boundary and the 'trace back'. When looking at the math we find that $\phi(t)$ changes the same way whether we take a particle approach or grid approach. If we define $\frac{D\phi(\mathbf{x},t)}{Dt} = k$, we have:

$$k = (\mathbf{u} \cdot \Delta)\phi + \frac{\partial \phi}{\partial t}$$

Solving $\frac{D\phi(\mathbf{x},t)}{Dt} = k$ using a backwards difference approximation (implicit first order time integration) we find:

$$\phi^t = \phi^{t-\Delta t} + \Delta t k$$

By moving the right hand side of Equation 1 to the left side, we find $k = 0$ and we are left with $\phi^t = \phi^{t-\Delta t}$. To find the location in the grid of $\phi^{t-\Delta t}$ we can use a backwards finite difference approximation:

$$\frac{\partial \mathbf{x}}{\partial t} = \frac{\mathbf{x}^t - \mathbf{x}^{t-\Delta t}}{\Delta t} = \mathbf{u}$$

Solving for node location at time $t - \Delta t$ we have $\mathbf{x}^{t-\Delta t} = \mathbf{x}^t - \Delta t \mathbf{u}$. As we need to interpolate $\mathbf{x}^{t-\Delta t}$ onto our grid we 'smooth' ϕ and we thus loose or dissipate ϕ.

2.2 Mean curvature flow

When applying mean curvature flow, we smooth our object along the mean curvature normal direction which means we minimize the surface area. Mean curvature flow is given by the partial differential equation

$$\frac{\partial \phi}{\partial t} = \nabla \cdot \frac{\nabla \phi}{||\nabla \phi||}$$

Where we initialize ϕ as a signed distance field and define the right hand side as κ. We can now rewrite κ using $\nabla \cdot (a\mathbf{v}) = \nabla a \cdot \mathbf{v} + a\nabla \cdot \mathbf{v}$ and get:

$$\begin{aligned}\kappa &= \nabla\phi \cdot \nabla \left(\frac{1}{||\nabla \phi||} \right) + \frac{1}{||\nabla \phi||} \nabla \cdot \phi \\ &= \nabla\phi \cdot \nabla \left(\left(\nabla\phi^T \nabla\phi \right)^{-\frac{1}{2}} \right) + \frac{\nabla^2 \phi}{||\nabla \phi||}\end{aligned}$$

Using the chain rule we can further deduce:

$$\nabla \left(\left(\nabla\phi^T \nabla\phi \right)^{-\frac{1}{2}} \right) = -\frac{1}{2} \frac{1}{||\nabla \phi||^3} \cdot 2(\nabla(\nabla\phi))\nabla\phi$$

Using the Hessian $\mathbf{H} = \nabla(\nabla\phi)$ and the trace of the Hessian $\nabla^2\phi = \mathrm{tr}(\mathbf{H})$ we have:

$$\begin{aligned}\kappa &= \frac{\mathrm{tr}(\mathbf{H})}{||\nabla\phi||} - \frac{\nabla\phi^T \mathbf{H} \nabla\phi}{||\nabla\phi||^3} \\ &= \frac{\nabla\phi^T \nabla\phi \mathrm{tr}(\mathbf{H}) - \nabla\phi^T \mathbf{H} \nabla\phi}{||\nabla\phi||^3}\end{aligned}$$

using central difference approximations we find that the approximation k of κ at some node in our grid can be found by:

$$k_{i,j} = \frac{(D_x\phi_{i,j})^2 D_{yy}\phi_{i,j} + (D_y\phi_{i,j})^2 D_{xx}\phi_{i,j} - 2D_{xy}\phi_{i,j} D_x\phi_{i,j} D_y\phi_{i,j}}{\sqrt{(D_x\phi_{i,j})^2 + (D_y\phi_{i,j})^2}} \tag{2}$$

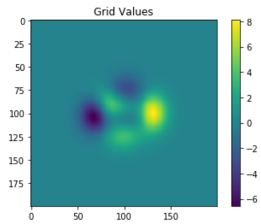
Where D_x is the first order approximation in the x direction, D_y in the y direction, D_{xx} is the second order approximation in the x direction, D_{yy} in the y direction and D_{xy} being the second order approximation in both the x and y direction. Using k we can now approximate the mean curvature at each node in our grid and construct the update scheme $\phi_{i,j}^{t+1} = \phi_{i,j}^t + \Delta t k_{i,j}$. However, some numerical considerations needs to be taken. For instance the denominator in Equation 2 might go to zero which is unrealistic in a signed distance field where it would be approximately 1 everywhere. We can remedy this by adding a small epsilon in the denominator, or by redefining the denominator to be equal to 1 if the actual value goes below $\frac{1}{2}$. We cab also clamp our k to avoid it going to infinity, as the maximum mean curvature we can find is $\frac{1}{\max(\Delta x, \Delta y)}$.

3 EXPERIMENT 1 - HOW BOUNDARIES AFFECT ADVECTION WITH SEMI LAGRANGIAN TIME INTEGRATION

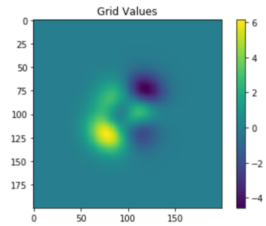
In this experiment we would like to examine how the boundaries affect our advection. In this week's notebook we have worked with the peaks functions, and used advection with semi Lagrangian time integration to turn the scalar field 360 degrees around. This can be

Author's address: Casper Bresdahl, whs715, University of Copenhagen, Copenhagen, Denmark, whs715@alumni.ku.dk.

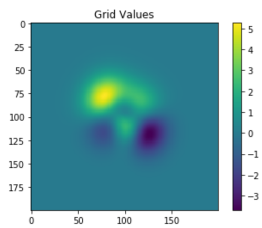
seen in Figure 1 and we will use this as a baseline comparison for our experiment.



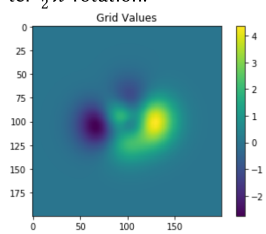
(a) Image of peaks function before advection.



(b) Image of peaks function after $\frac{1}{2}\pi$ rotation.



(c) Image of peaks function after $1\frac{1}{1}\pi$ rotation.



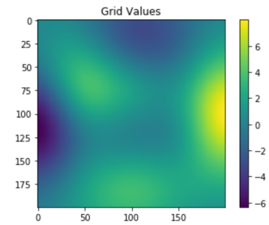
(d) Image of peaks function after 2π rotation.

Fig. 1. Advection without any boundary issues. The grid is defined over the domain -5 to 5 with 200 grid nodes along each axis, and with $\Delta t = 0.0005$.

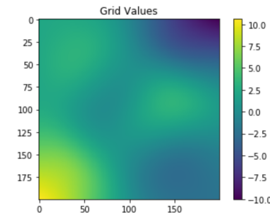
As we see this is no perfection advection, and we have an error of 0.21, meaning we loose about 21% mass. We also note we are not having any issues with boundaries, as the peaks rotate in the middle of the grid without hitting the boundaries. We can now shrink the grid such that when rotating the peaks we will move outside the grid. The previous grid was defined over the domain -5 to 5, we now shrink this to -1.5 to 1.5. In Figure 2 we see the results of shrinking the grid. When comparing Figure 1a and Figure 2a we do not see too much of a difference except we have 'zoomed in'. However, comparing Figure 1b with Figure 2b and Figure 1c with Figure 2c we see that the 'shapes' are similar, but that the peaks have become a lot larger in the corners of Figure 2b and Figure 2c and similarly the big canyon have becomes much deeper. This is to the extend that the smaller peaks in the experiment images have grown almost as large as the original large peak. We also see that the small canyon in the lower right corner of Figure 2b and lower left corner of Figure 2c have been somewhat leveled out. When comparing Figure 1d and Figure 2d we see pretty much the same shapes but we again see the scaling of the peaks have changed, although not as drastically as for the intermediate rotations.

3.1 Discussion of results

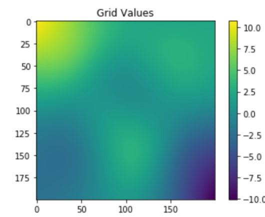
We see when we shrink the domain that instead of the peaks and canyons going towards 0 they actually become bigger. As they rotate, they move outside our grid, and thus we need to interpolate them back in. In this process we compute the new grid value based on an interpolation to the nearest grid cell. The fact we move outside the grid causes us to lose information about the peaks and canyons and because we use the neighbourhood around the nearest grid cell we also a slight error in the computed value as it actually should have been computed at a slightly different location. These are probably the causes we see a different behaviour of the peaks and canyons compared to when we do not have any boundary issues. The interpolation used to get the results for this experiment is the interpolation function in this week's notebook.



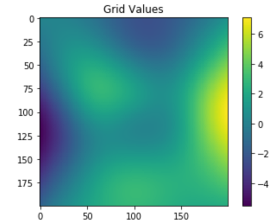
(a) Image of peaks function before advection.



(b) Image of peaks function after $\frac{1}{2}\pi$ rotation.



(c) Image of peaks function after $1\frac{1}{1}\pi$ rotation.



(d) Image of peaks function after 2π rotation.

Fig. 2. Advection with boundary issues. The grid is defined over the domain -1.5 to 1.5 with 200 grid nodes along each axis, and with $\Delta t = 0.0005$.

4 EXPERIMENT 2 - THE EFFECT OF DIFFERENT BOUNDARY CONDITIONS WHEN APPLYING MEAN CURVATURE FLOW ON SIGNED DISTANCE FIELDS.

In this experiment we would like to examine how different boundary conditions influence the end result of applying mean curvature flow to a signed distance field. When deriving the mean curvature flow we run into boundary issues as we need the values surrounding a grid cell to compute its value. One way to deal with this is to pad the grid with an extra 'layer' such that when we compute the values inside the grid, we can use these 'extra' values to avoid out of bound errors. However, which values should we pad with? In Figure 3 we have depicted a signed distance field and three boundary conditions we will look at in turn.

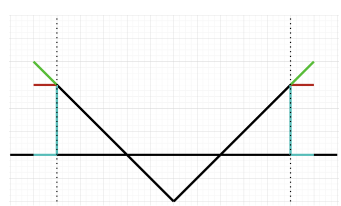
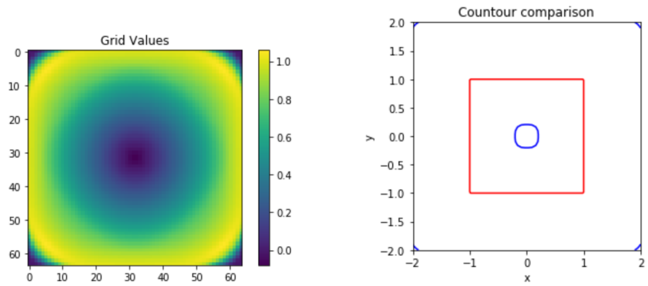


Fig. 3. A signed distance field depicted in black along three different choices for padding to handle boundary conditions.

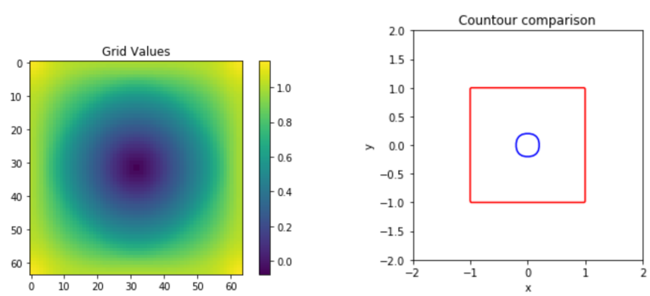
In blue we have a Dirichlet boundary condition where the chosen constant is zero. As we see in Figure 3 this gives a very abrupt change in the curvature at the boundary. Simply padding our signed distance field from this week's notebook with zeros gives us the following results seen in Figure 4.



(a) (b)

Fig. 4. Padding our signed distance field with zeros before applying mean curvature flow. (A) Image of the grid after applying mean curvature flow. (B) Contours of the grid. Red contours are from before applying mean curvature flow, and blue contours are from after.

We here note how the corners of the resulting grid have values close to zero but the rest of the grid seems to have increasing values when we move from the center towards the boundaries. In Figure 4b we note the slight oval shape of the center blue contour, and how we can see the corner contours.
The red boundary condition from Figure 3 is a von Neumann boundary condition, that is we have $\frac{\partial \phi}{\partial x} = 0$. When discretizing this by using central difference approximations, we find we can implement this by mirroring the values inside the grid along the border to the padded layer. Hence we see in Figure 3 the red boundary condition stays constant for the last value in the signed distance field. The results of using a von Neumann boundary condition can be seen in Figure 5.



(a) (b)

Fig. 5. Padding our signed distance field with mirrored values before applying mean curvature flow. (A) Image of the grid after applying mean curvature flow. (B) Contours of the grid. Red contours are from before applying mean curvature flow, and blue contours are from after.

We here note how the values of the grid always increase as we move from the center to the boundaries, and how the blue contour of Figure 5b is completely circular and is the only contour after applying the mean curvature flow.
The last boundary condition is depicted in green in Figure 3 and is $\frac{\partial^2 \phi}{\partial x^2} = 0$. As depicted in Figure 3 this would represent the continuation of the signed distance field as if there were no boundary. I have not been implementing this boundary condition as the discretization of $\frac{\partial^2 \phi}{\partial x^2} = 0$ quickly becomes rather cumbersome. However, this boundary condition will be included in the following discussion of the results.

4.1 Discussion of results

A natural question is which boundary condition yields the best results? To answer this, we again look at **??**. When computing values for our grid we would prefer never to have a boundary, and so naturally using $\frac{\partial \phi}{\partial x} = 0$ as our boundary condition would give us this effect, as this would 'continue our signed distance field'. However, it is cumbersome to implement and a much simpler solution is the von Neumann boundary condition. Although it does not 'continue' the function, it does not make an abrupt change, and when we add together the nodes in the neighbourhood around a boundary node, it does not 'do anything unexpected'. We can compare this to the Dirichlet boundary condition which to some extend could be described as 'removing' the node value. That is, when we add together the neighbourhood we are gonna miss a term which is roughly as big as the value of the boundary node. This is why probably why in our results we see that the corners of the mean curvature flow using Dirichlet boundary conditions are bending downwards whereas the results when using von Neumann are much more 'continuous'. We thus find $\frac{\partial \phi}{\partial x} = 0$ would be the best boundary condition, but because it is hard to implement the von Neumann boundary condition is a good backup.

5 CONCLUSION

We end by concluding that advection without any boundary issues behaves differently than advection where we are affected by boundaries. This is due to the loss of information as our peaks move outside the boundaries, and due to how we interpolate back onto our grid, where if we interpolate back to the nearest grid cell might end up computing the value the cell in a slightly wrong location. We also conclude that using different boundary conditions when applying mean curvature flow can give vastly different end results. We find using $\frac{\partial \phi}{\partial x} = 0$ would give the most natural boundary and thus the best results, however, can be hard to implement. Thus using von Neumann boundary conditions is a good alternative as it does not make any abrupt change around the neighbourhood which a Dirichlet boundary condition would.