CMIS Hand-in 2: Finite Difference Methods 2

Nikolai Plambech Nielsen
lpk331@alumni.ku.dk
Niels Bohr Institute, University of Copenhagen

1 INTRODUCTION

2 SEMI-LAGRANGIAN IMPLICIT TIME INTEGRATION: SOLVING THE ADVECTION EQUATION

In this problem we encounter the advection equation:

$$\frac{D\phi}{Dt} = (\mathbf{u}\cdot\nabla)\phi + \frac{\partial\phi}{\partial t} \tag{1}$$

Where $D\phi/Dt$ is the particle derivative of the scalar field ϕ, and $\mathbf{u}$ is the associated velocity field. We set $D\phi/Dt = 0$ such that no dissipation occurs for the system:

$$\frac{\partial\phi}{\partial t} = -(\mathbf{u}\cdot\nabla)\phi \tag{2}$$

The advection equation describes the flow of some scalar field ϕ in a velocity field $\mathbf{u}$. This could for example describe the motion of food colouring in a pool of water. Then ϕ would be the density of the food colouring molecules, whilst $\mathbf{u}$ would be the flow of the water in the pool.

The idea behind semi-lagrangian time integration is to use this property of the advection equation, and directly use the values of the scalar field at the previous times step. Specifically we treat our grid points as particles floating along the vector field. At a previous time, the grid point particles (to first order) were at

$$\mathbf{x}^{t-\Delta t} = \mathbf{x}^t - \Delta t\mathbf{u} \tag{3}$$

we then use the the value of the field at these points as the value for *phi* on the grid points, for the next time step:

$$\phi(\mathbf{x}^t) \leftarrow \phi(\mathbf{x}^t - \Delta t\mathbf{u}) \tag{4}$$

The problem then, is that we are not guaranteed, that the grid point particles hit a grid point position, and as such we might not know the value of $\phi(\mathbf{x}^t - \Delta t\mathbf{u})$. To get around this we use a bilinear interpolation between the four nearest grid points to $\mathbf{x}^{t-\Delta t}$, and use this interpolated value as the true value.

For this problem we use a velocity field of $\mathbf{u} = (y, -x)^T$, corresponding to constant circular motion in the clockwise direction. For the scalar field we use a sum of Gaussian functions. If we choose the width of the Gaussians to be sufficiently small, and a large domain, virtually all the points of ϕ that are non-zero will be nowhere near the boundary of the domain. This ensures that a bilinear interpolation always has enough points to use for the interpolation.

With this vector field, we expect the scalar field to rotate in time, with a period of 2π. We can then perform an (approximate) full rotation in N steps by defining $\Delta t = 2\pi/N$. It is only approximate due to the truncation error present when discretising the system.

2.1 Experiments

For this problem I perform two experiments. In the first I want to test how the rotation depends on the time step, and in the second I test how much bleeding occurs, again as a function of the time step and grid spacing.

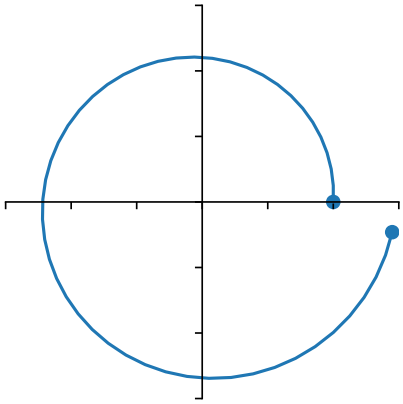

Figure 1: Typical trajectory of a grid point after N iterations of semi-Lagrangian time integration (if the grid point was continually interpreted as a particle).

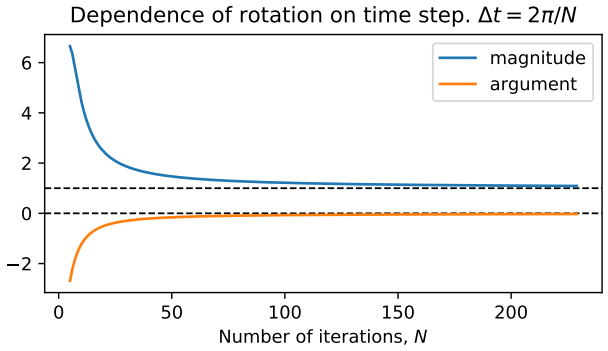

Figure 2: Argument and ratio of magnitudes calculated as a function of the time step size Δt. As Δt decreases, the system approaches a full revolution, corresponding to an argument of 0 and a magnitude ratio of 1.

To test the rotation, I take a point, say $\mathbf{x} = (2, 0)^T$ and backtrace N times to get $\mathbf{x}'$, corresponding to one rotation of the whole grid, and compare the magnitude and argument of $\mathbf{x}'$ to that of $\mathbf{x}$. An example of the trajectory is seen in figure 1, with $\Delta t = \pi/25 \approx 0.126$. In figure 2 the ratio of magnitudes $||\mathbf{x}'||/||\mathbf{x}||$ and the argument of x' (atan2(x'_1, x'_2)) is calculated for a range of simulations. As seen in the figure, the values tend to 1 and 0 respectively, corresponding

N	atan2(x_1', x_2')	$\|\|\mathbf{x}'\|\|/\|\|\mathbf{x}\|\|$
74	-0.997	1.300
207	-0.0323	1.090

Table 1: Number of iterations per revolution needed to achieve an accuracy greater than 0.1 in both argument and ratios of magnitudes in the Semi-lagrangian time integration.

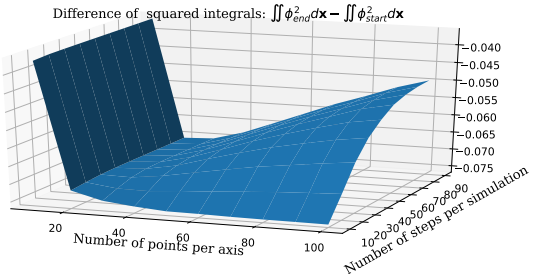

Figure 3: $\Delta\phi^2$ for a range of different number of iterations and number of grid points. Note $\Delta t \propto N_t, \Delta_x \propto N_x$.

to:

$$\lim_{\Delta t \to 0} \mathbf{x}' = \mathbf{x}. \tag{5}$$

To number of iterations per rotation needed to achieve an accuracy of 0.1 in the argument and ratios of magnitude are tabulated in table 1.

Since we have no source term in our equation, and we set $D\phi/Dt = 0$ the integral of ϕ cannot increase. If our simulation is faithful, then we would expect the integral of ϕ stay constant in time. A way to measure the faithfulness of the simulation would then be to calculate

$$\Delta\phi^2 = \iint \phi^2_{\text{end}} - \phi^2_{\text{start}} dx\, dy \approx \sum\sum \phi^2_{\text{end}} - \phi^2_{\text{start}} \Delta x \Delta y \tag{6}$$

for a range of grid spacings and time step sizes. One could also calculate the squared difference integral, but since we already know that the system never completes a full rotation (except in the limit $\Delta t \to 0$), the actual error calculated would not be attributable to the interpolation only, but a mix of the approximate full rotation and the interpolation. By calculating integral of the difference of the squares instead, we eliminate the dependency on the only approximate rotation.

With this measure of error we also expect $\lim_{\Delta x, \Delta t \to 0} \Delta\phi^2 = 0$. The results are seen in figure 3. Notice the sign on the z-axis. The difference does decrease as Δt and Δx are decreased, but seems to taper off towards some value. There is of course the sharp decrease in error when $N_x = 10$, but this is a pathological case, because $\Delta x = 2$, and I am calculating the error for a Gaussian profile with $\sigma_x = \sigma_y = 1$. There simply is not enough points to sample the function accurately.

Lastly we view the evolution of the system with time, seen in figure 4. And indeed, the scalar field rotates.

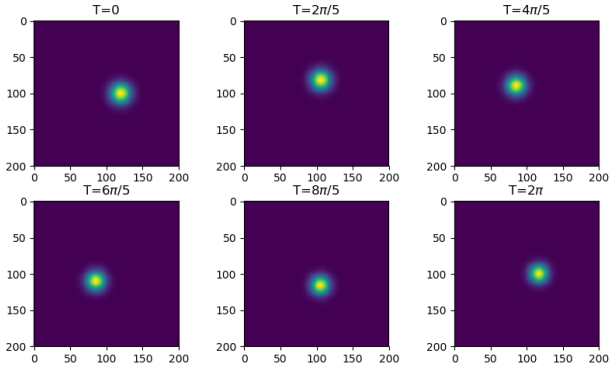

Figure 4: Evolution of a Gaussian profile for $\mathbf{u} = (y, -x)^T$, with $\Delta t = \pi/50$. Shown after n iterations, with $n \in \{0, 20, 40, 60, 80, 100\}$.

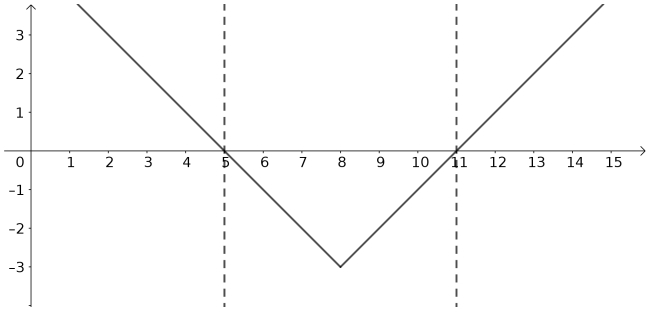

Figure 5: A signed distance field in one dimension, with the inside as $x \in [5, 11]$.

3 MEAN CURVATURE FLOW WITH FINITE DIFFERENCE METHODS

Next we turn our attention to solving the mean curvature flow equation for some signed distance field. Mean curvature flow is analogous to wrapping a tight string around an isosurface of a scalar field and then tightening the string such that it follows the isosurface at a higher value. The equation is given by

$$\frac{\partial\phi}{\partial t} = \nabla \cdot \frac{\nabla\phi}{||\nabla\phi||} = \kappa \tag{7}$$

The right hand side of the equation can be rewritten as

$$\kappa = \frac{\nabla\phi^T \nabla\phi \operatorname{tr} H - \nabla\phi^T H \nabla\phi}{||\nabla\phi||^3} \tag{8}$$

with H being the Hessian matrix.

3.1 Signed Distance Fields

A signed distance field is a scalar field describing the signed euclidian distance from each pixel of a black and white, binary image, to the nearest border between black and white, in units of pixels. The sign is such that white pixels are counted as inside, with a negative distance to the border. A one dimensional is seen in figure 5, where the inside is defined to be $x \in [5, 11]$. For points sufficiently close to a border will have a gradient of approximately 1.

3.2 Discretisation

For the domain we employ a central difference approximation for each of the different spatial derivatives. This gives

$$\kappa \approx k = \frac{(D_x\phi)^2 D_{yy}\phi + (D_y\phi)^2 D_{xx}\phi - 2D_{xy}\phi D_x\phi D_y\phi}{\left[(D_x\phi)^2 + (D_y\phi)^2\right]^{3/2}} \tag{9}$$

where D is the central difference approximation, where the subscripts denote which variables are differentiated with respect to. For example the central difference approximation for the second order derivative with respect to x:

$$D_{xx}\phi_{ij} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{\Delta x^2}. \tag{10}$$

Some problems with this equation would be at or around points where the norm of the gradient disappears. This would result in possible division by zero, or just a very large value of k.

A conservative maximum bound for the mean curvature is given by the inverse of the grid spacing: $k_{\max} = \max(\Delta x, \Delta y)^{-1}$. To remedy values that are too large one can clamp the value of k to between $\pm k_{\max}$ by

$$k \leftarrow \max(-k_{\max}, \min(k, k_{\max})). \tag{11}$$

For the temporal derivative we employ a first order forward difference to integrate in time:

$$\phi_{ij}^{t+\Delta t} = \phi_{ij}^{t} + \Delta t k_{ij}, \tag{12}$$

for some small Δt.

This numerical scheme is conditionally stable, since too large a time step will lead to numerical blow up of the solution. The condition is called the CFL condition, and is given by

$$\Delta t \leq \frac{h}{2k_{\max}}, \quad h = \min(\Delta x, \Delta y) \tag{13}$$

In our case, $\Delta x = \Delta y = 1$, and $\Delta t \leq 1/2$.

3.3 Boundary conditions

For a signed distance field, the boundary nodes will not have a fixed value, and as such a von Neumann boundary condition is preferable to a Dirichlet boundary condition. In general, the signed distance field will have a gradient of ± 1 in each direction, so long as the node is sufficiently far away from the midpoint of a region. Since midpoints cannot occur at the boundary, it makes sense to extend the derivative to the boundary. For the left boundary this would mean $d\phi_{1,j}/dx = d\phi 2_j/dx$. This is implemented using ghost nodes and a central difference approximation:

$$\frac{\partial\phi_{1,j}}{\partial x} = \frac{\partial\phi_{2,j}}{\partial x} \quad \Rightarrow \quad \frac{\phi_{2,j} - \phi_{0,j}}{2\Delta x} = \frac{\phi_{3,j} - \phi_{1,j}}{2\Delta x} \tag{14}$$

Giving the updating formula for ghost nodes on the left boundary:

$$\phi_{0,j} = \phi_{1,j} + \phi_{2,j} - \phi_{3,j}. \tag{15}$$

3.4 Experiments

The mean curvature flow equation should conserve the signed distance field properties. To test this I convert ϕ back to a black and white image after the simulation using the mapping

$$\text{BW} = \begin{cases} 1 & \phi < 0 \\ 0 & \phi \geq 0 \end{cases} \tag{16}$$

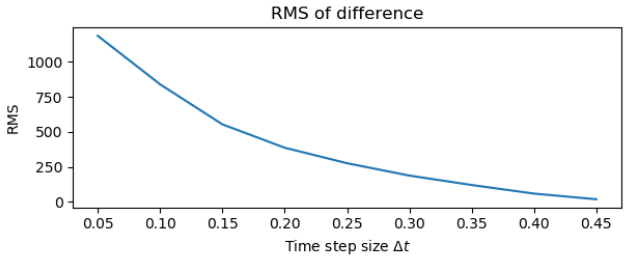

Figure 6: RMS between ϕ_{end} and $\phi_{\text{reconstructed}}$ as a function of Δt, with $T = 500$.

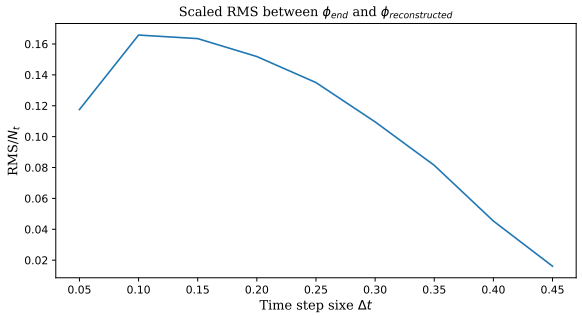

Figure 7: RMS between ϕ_{end} and $\phi_{\text{reconstructed}}$ as a function of Δt. Scaled by the number of iterations: $N_t = T/\Delta t$, for $T = 500$.

This mapping erases all information from ϕ, other than which parts of the field were inside and outside. Then I use the `bw2phi` function supplied by Kenny to generate the signed distance field from this new black and white image, introducing information to the system again. To quantify the error I calculate the RMS of the residual between the two images over the domain:

$$\text{Res} = \sum_{j=0}^{N}\sum_{i=1}^{N}(\phi_{\text{end}} - \phi_{\text{reconstructed}})^2/N^2 \tag{17}$$

This measurement should yield 0 if the simulation is a faithful representation of the mean curvature flow equation. This is tested on the sample image supplied by Kenny, resembling a flipped U, for a range of time step sizes (while keeping the "absolute time" unchanged, such that the end images should be equal across simulations). We of course expect the residual to trend towards zero as the time step is decreased. The results are shown in figure 6.

As seen in the figure, the residual actually increases as Δt is decreased, contrary to the expectation. This could however be a result of the number of iterations performed. For $\Delta t = 0.45$ a total of 1112 iterations were completed, while for $\Delta t = 0.05$ 10000 iterations were completed. Since more iterations equals more calculations, each of which includes truncation errors, it is natural to expect a higher residual as more iterations are performed. If we then also scale the residual by the number of iterations we get the error dependence on Δt, per iteration. This is seen in figure 7.

The error still increases as Δt decreases, until some critical value of $\Delta t \approx 0.1$ is reached. I would have liked to investigate this further, if time permitted.

Lastly, in figure 8, the signed distance field can be seen at a variety of times, with $\Delta t = 0.25$, showing the time evolution of ϕ. As expected we see the contour of $\phi = 0$ rounding at the corners and shrinking.

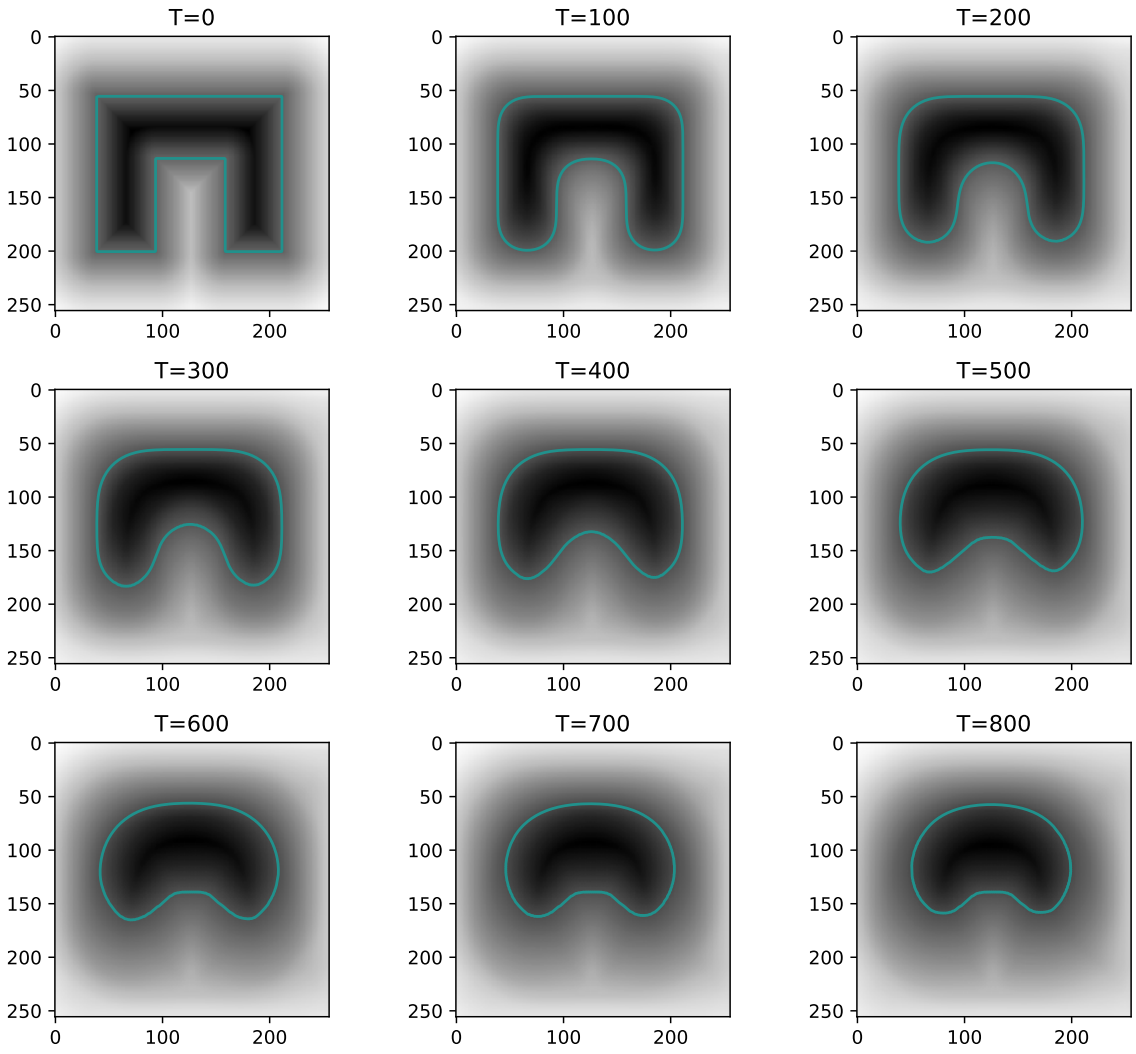

Figure 8: Evolution of the signed distance field generated from the sample image. $\Delta t = 0.25$.