Analysis of Semi-Lagrangian advection using finite difference methods

Michael Adriaens
Supervisors:
Jason Frank (Mathematical Sciences)
Michiel van den Broeke (Climate Physics)
Sander Tijn (KNMI)

A thesis presented for the degree of
Master of Science

Universiteit Utrecht

Mathematical Sciences and Climate Physics
Utrecht University

In collaboration with

Royal Dutch Meteorological Institute (KMNI)

Utrecht, The Netherlands
28-08-2023

Abstract

The advective terms in several numerical weather models have been integrated with Semi-Lagrangian methods for decades. Their stability allows for larger time steps than explicit finite difference methods, while experiencing less dispersion than implicit finite difference methods. In the case of the one dimensional advection equation, it turns out that for uniform and constant velocity, the Semi-Lagrangian method is equivalent to a shifted finite difference method. The degree of polynomial interpolation determines the truncation of the Taylor series and accuracy of the finite difference approximations. The shift is necessary to account for arbitrarily large time steps without losing stability as occurs for explicit finite difference methods. For linear interpolation, the integration is guarenteed to be stable for non-uniform and non-constant velocity, while higher order interpolation can in theory experience instability. The instabilities occasionally observed in the HARMONIE operational code are a part of the departure point problem. This thesis proves the existence and uniqueness of and convergence to these departure points. In areas with large vertical velocity gradients, the convergence of fixed point iteration can be too slow, resulting in nonphysical crossing of characteristics and therefore negative model layer heights. This could be the key to finding a criterion to prevent runtime errors occurring and improving reliability of numerical weather prediction.

Contents

1 Introduction

Historical context
What is the thing people talk about when they have no conversation topics? That's right, weather. The weather always influences our lives. Whether that is through being the reason for our holiday destinations, the reason plans get cancelled or the reason people need to flee their home. Historically people have always been fascinated by and have admired the weather, first thinking it is due to gods and later learning about its physics. Predicting weather has for a long time seemed impossible. In the late 19th century, the equations that govern the flow in the atmosphere were known, but no solution could be calculated.

In 1913, Lewis F. Richardson proposed the idea to use current observations of the state of the atmosphere to calculate how the state changes over a certain time interval. He gathered all the observations available at that time and used his finite difference method to predict the state in 6 hours time. He did these calculations by hand and it took him more than 6 weeks. As promising as it sounded, the prediction was not accurate. The numerical result showed a decrease in surface pressure of 145 hPa, which was luckily not observed (Richardson, 1922). In spite of this error, Richardson believed that the technique could be successfully applied with more observational coverage. And someday he hoped that the time to compute the prediction would become smaller than the time interval over which the calculation is done. This is essential for numerical weather prediction to be useful.

In the following years, difference equations started to be researched. A very important analysis was done by Courant, Friedrichs and Lewy in 1928. They showed for hyperbolic differential equation, that the analytical domain of dependence should be contained in the numerical domain of dependence. This puts a limit on the ratio between spatial and temporal intervals. On a fixed grid, there is a limit on the size of the time step that can be used for the numerical integration (Courant et al., 1928). It was this limit that Richardson had exceeded which, unbeknown to him at the time, created an unstable numerical prediction. Because of this limit, a lot of smaller time steps needed to be calculated to complete the prediction in a stable way. More time steps required more computational power than was available at the time. This computational limit was overcome when the first big computers were created and in 1950 the first numerical weather prediction was made. More specifically, in 1950 the first prediction was made where the forecast time period was at least equal to the computational time. This meant the turning point, where numerical weather forecasting could really have a function as an predictor of weather, the realisation of Richardson's dream.

Prior research
The first numerical weather predictions treated advection with Eulerian finite difference methods, subject to the aforementioned stability limit. Eulerian

means that the advection is seen from a Eulerian frame of reference, where local changes in some scalar depend on the flow to and from this point in space. These changes are then calculated on a stationary grid, where finite difference are used to approximate the derivatives. Another method would be to look from the frame of a fluid parcel and follow this parcel along its trajectory. This would be advection in a Lagrangian frame of reference, where the state of the air parcel is assumed not to change along its trajectory. A starting grid of air parcels would only change their positions, not the value of the scalar inside this parcel. These values are constant, so unconditionally stable. The only challenge is the calculation of trajectories. Lagrangian method were already shown in 1955 to not be very useful, as an ordered starting grid will quickly become very irregular (Welander, 1955). The win of the unconditional stability came at the cost of lost accuracy and added complexity. Around this time, a lot of research was being done on a 'hybrid' method. This method combined the regularity of a stationary grid of Eulerian methods and the greater stability of Lagrangian methods. This hybrid method is called a Semi-Lagrangian method and early research showed promising results (Fjørtoft, 1955; Wiin-Nielsen, 1959). It entails resetting the 'flowing' grid of the Lagrangian method to an ordered grid after every time step.

Applying the Semi-Lagrangian technique to meteorological equation was first done in 1962 by Krishnamurti and in subsequent years further improved (Krishnamurti, 1962; Sawyer, 1963; Leith, 1965). The next step was combining Semi-Lagrangian treatment of advection with semi-implicit methods for the gravitational oscillations present in the atmosphere (Robert, 1981; Robert, 1982). These oscillations were a big stability issue with explicit methods, limiting the time steps taken. The Semi-Lagrangian semi-implicit method was applied to many different problems and they all showed an improvement in stability and accuracy over finite difference methods (Temperton & Staniforth, 1987; Côté & Staniforth, 1988; Bermejo & Staniforth, 1992). A review was published in 1991 by Staniforth, discussing all the advancements on Semi-Lagrangian methods and summarising the current knowledge on accuracy and efficiency (Staniforth & Côté, 1991). It was shown that using Semi-Lagrangian semi-implicit methods for numerical weather forecasting could reduce the computation time for a 10 day forecast from 24 to about 4 hours, about 6 times as fast.(Simmons, 1991). This allowed for a higher resolution and longer forecasting range. The main improvement was the increase in time step that could be taken. The limit found by Courant, Friedrichs and Lewy in 1928 could be avoided.

In 2005 the programme HIRLAM (High Resolution Limited Area Modelling) was founded as a collaboration between numerous European meteorological institutes. In 2008 they created the HARMONIE model (HIRLAM ALADIN Research on Mesoscale Operational NWP In Euromed) together with ALADIN from MeteoFrance and the ECMWF (European Centre for Medium-Range Weather Forecasts). This model is still in use at KNMI (Koninklijk Nederlands Meteorologisch Instituut).

The problem
As the Semi-Lagrangian method is understood to be unconditionally stable when applied to the advection problem, it should not fail due to rounding or truncation errors. Nevertheless, occasionally the forecast produced by HARMONIE at KNMI throws an exception. The computations are interrupted and have to be restarted from zero. To prevent a recurrence, the time step is reduced and the forecast is recalculated. If the problem occurs too late in the forecast, it is not always possible to rerun the model before the scheduled release of the forecast. Understanding the reason for the failure could prevent these problems.

In this thesis we study the stability of Semi-Lagrangian methods. What makes Semi-Lagrangian methods so useful for advection problems? Can we understand their advantage over Eulerian finite difference methods by finding a direct relation between the two?

Overview of contents
This thesis will first start with a summary of Eulerian finite difference methods for solving the one dimensional advection problem. More specifically, we will discuss their consistency, stability and convergence. Next, the Semi-Lagrangian method is explained, with both the finding of the departure point and interpolation of the value at the departure point. For the iteration of finding the departure point, we will prove the existence of such a point for bounded velocity fields, discuss the uniqueness and convergence of the iteration and show an example of the method used in HARMONIE (SETTLS). Then we will discuss how the Semi-Lagrangian methods can be understood as Eulerian finite difference methods. We will show that in that context, the advantage of Semi-Lagrangian methods can be explained.

2 Eulerian advection

Change of a scalar field $u(t, \vec{x})$, $t \in \mathbb{R}$ and $\vec{x} \in \mathbb{D}$, is governed by the continuity equation

$$\frac{\partial u}{\partial t} + \vec{\nabla} \cdot (u\vec{v}) = f,$$

where $\vec{v}(t, \vec{x})$ is the velocity field of the fluid, $f(t, \vec{x})$ is the creation or removal of u locally and $\vec{\nabla}\cdot$ is the divergence operator on $\mathbb{D}$. The equation describes that the change of u in time is determined by the flow of u and local changes in u.

For example, when u is a mass density of some quantity in the atmosphere, u is assumed to be only transported. That would mean the following holds

$$f = 0.$$

With that, the continuity equation turns into the advection equation

$$\frac{\partial u}{\partial t} + \vec{\nabla} \cdot (u\vec{v}) = 0.$$

Now the local change of u in a volume of air is only due to transport with the velocity $\vec{v}$. When the velocity is divergence free, i.e. $\vec{\nabla} \cdot \vec{v}$, the advection equation becomes

$$\frac{\partial u}{\partial t} + \vec{v} \cdot \vec{\nabla} u = 0.$$

This is often the assumption in atmospheric models, because air in the atmosphere can be assumed to be incompressible. In one dimension, u and v depend only on t and x. I will apply the discussed techniques to the one dimensional advection equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(uv) = 0.$$

Often techniques are easier to apply and prove when the velocity is constant and uniform, then denoted by $v(t, x) = c \in \mathbb{R}$. That changes the advection equation to:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0.$$

We will also consider a domain with length L and periodic boundary conditions $u(t, 0) = u(t, L)$ and if applicable $v(t, 0) = v(t, L)$. Furthermore, the initial condition will be denoted by $u(0, x) = g(x)$.

2.1 Analytical solution

The advection equation is a hyperbolic partial differential equation (PDE). One such category of PDEs are the quasi-linear PDEs.

Definition 2.1 (Quasi-linear partial differential equation)**.** A PDE is called a *quasi-linear* PDE of order k if the highest order derivatives are order k and the terms of order k occur as linear combinations with coefficients depending only on terms of order $i < k$. Take for example a function $u(x, y)$. A first order quasi-linear PDE will be of the form

$$a(x, y, u)\frac{\partial u}{\partial x} + b(x, y, u)\frac{\partial u}{\partial y} + c(x, y, u) = 0$$

2.1.1 General velocity

The advection equation is a quasi-linear partial differential equation of the variable $u(t, x)$. It would even be considered fully linear. It has $a(t, x, u) = 1$, $b(t, x, u) = v(t, x)$ and $c(t, x, u) = 0$:

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0. \tag{1}$$

All quasi-linear PDEs can be solved by method of characteristics. This method transforms the PDE into a set of ordinary differential equations (ODEs). For a quasi-linear PDE of order k, the characteristic curve can be parameterised with k parameters. The one dimensional advection equation is a first order PDE, so the characteristic lines can be parameterised with parameter s. The change of u with s is given by

$$\frac{du}{ds} = \frac{\partial u}{\partial t}\frac{dt}{ds} + \frac{\partial u}{\partial x}\frac{dx}{ds}. \tag{2}$$

Comparing equation (2) to equation (1) gives the following system of ODEs:

$$\begin{cases} \frac{dt}{ds} = 1, & t(0) = 0 \\ \frac{dx}{ds} = v(t, x), & x(0) = x_0 \\ \frac{du}{ds} = 0, & u(0) = g(x_0) \end{cases}$$

The first equation yields $t(s) = s$. The second one is a bit more tricky, with its result being

$$x(s) = x_0 + \int_0^s v(t(\sigma), x(\sigma))d\sigma = x_0 + \int_0^s v(\sigma, x(\sigma))d\sigma$$

The lines of x depending on t are called the characteristic lines and are given by

$$x(t) = x_0 + \int_0^t v(\tau, x(\tau))d\tau$$

Because the derivative of u along the characteristics is zero, u is conserved along these lines. The solution for $x(t, x)$ is given by

$$\begin{aligned} u(s) &= g(x_0) \\ u(t, x) &= g\left(x - \int_0^t v(\tau, x(\tau)) d\tau\right) \end{aligned}$$

The problem is, that this solution is not explicit. To find the solution for a point (t, x), one has to find the trajectory $x(t)$ first, which generally has no analytic solution. So this has to be somehow approximated and then the characteristic has to be followed back to the initial condition.

Another problem could be the crossing of different characteristic lines, making the solution not unique. In the case that v is divergence free, the exact characteristic lines will never cross. With a velocity field with non-zero divergence or when approximating the characteristic lines, this becomes a possibility.

2.1.2 Uniform constant velocity

When the velocity is uniform and constant, the solution will be analytic. Assuming $v(t, x) = c \in \mathbb{R}$:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0.$$

Again, comparing this equation to equation (2) gives the following system of ODEs:

$$\begin{cases} \frac{dt}{ds} = 1, & t(0) = 0 \\ \frac{dx}{ds} = c, & x(0) = x_0 \\ \frac{du}{ds} = 0, & u(0) = g(x_0) \end{cases} \rightarrow \begin{cases} t(s) = s \\ x(s) = x_0 + cs \\ u(s) = g(x_0) \end{cases} \rightarrow u(t, x) = g(x - ct)$$

Here $u(0, x) = g(x)$ is the initial condition. This shows that the solution of the advection equation is the initial condition shifted along characteristic lines $x(t) = x_0 + ct$. In this case the characteristic lines are straight and parallel lines, visualised in figure 1.

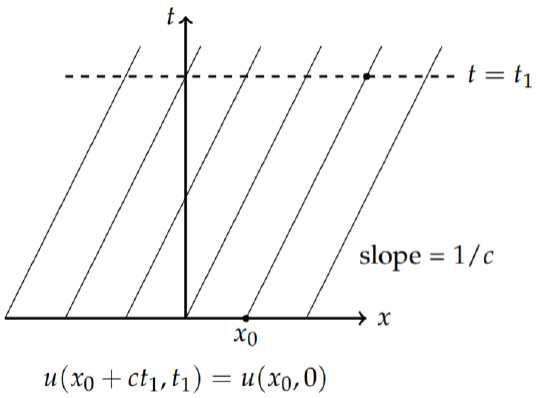


Figure 1: The characteristic lines, with slope $1/c$ in the xt-plane. The initial condition at $(x,t) = (x_0, 0)$, will equal the solution along the characteristic line though that point (Herman, 2015)

.

2.1.3 Conservation of mass and energy

The advection equation with periodic boundary conditions conserves both mass and energy, denoted respectively by

$$\int_0^L u(t,x)dx \qquad \text{and} \qquad \int_0^L u^2(t,x)dx.$$

Moreover, every higher power of u is also conserved. The following derivation shows the conservation of mass:

$$\begin{aligned}
\frac{\partial}{\partial t}\int_0^L u dx &= \int_0^L \frac{\partial u}{\partial t} dx \\
&= -\int_0^L \frac{\partial}{\partial x}(uv)\, dx \\
&= (u(t,0)v(t,0) - u(t,L)v(t,L)) \\
&= 0.
\end{aligned}$$

And for the energy the derivation is

$$
\begin{aligned}
\frac{\partial}{\partial t}\int_0^L u^2 dx &= \int_0^L \frac{\partial}{\partial t}\left(u^2\right) dx \\
&= 2\int_0^L u\frac{\partial u}{\partial t} dx \\
&= -2\int_0^L u\frac{\partial}{\partial x}\left(uv\right) dx \\
&= -2\int_0^L uv\frac{\partial u}{\partial x} dx - 2\int_0^L u^2\frac{\partial v}{\partial x} dx \\
&= -\int_0^L v\frac{\partial}{\partial x}\left(u^2\right) dx - 2\int_0^L u^2\frac{\partial v}{\partial x} dx \\
&= \left(u^2(t,0)v(t,0) - u^2(t,L)v(t,L)\right) - \int_0^L u^2\frac{\partial v}{\partial x} dx \\
&= -\int_0^L u^2\frac{\partial v}{\partial x} dx.
\end{aligned}
$$

In the case that the velocity is uniform, so there is no horizontal gradient, there is energy conservation

$$
\begin{aligned}
\frac{\partial}{\partial t}\int_0^L u^2 dx &= -\int_0^L u^2\frac{\partial v}{\partial x} dx \\
&= 0.
\end{aligned}
$$

A constant velocity is not necessary for energy conservation. That the energy is conserved in the case of advection with uniform velocity makes sense. Whatever initial condition one starts with, it will be transported at a certain velocity everywhere, but never change shape, u does not change along the characteristics. When the velocity only depends on time, the characteristics are no longer straight lines, but their horizontal distance will be constant.

2.2 Numerical solution

Without solving a differential equation analytically, a solution can be found numerically. A difference equation is a numerical scheme on a discrete domain

$$
t^n = n\Delta t, \quad n = 0, 1, 2, \ldots \quad \text{and} \quad x_j = j\Delta x, \quad j = 0, \ldots, d-1,
$$

with $d\Delta x = L$. This is used to approximate a differential equation on a continuous domain. The goal is to find a solution u_j^n of the difference equation that approximates the solution $u(t^n, x_j)$ to the differential equation on the gridpoints, or $u_j^n \to u(t^n, x_j)$ as $\Delta t, \Delta x \to 0$. Important techniques include finite

difference methods and finite volume methods for the spatial differences. To numerically integrate the PDE forward in time, implicit or explicit methods can be used, or a combination of both. A general time integration can be written as

$$\vec{u}^{n+1} = f(\vec{u}^{n+1}, \vec{u}^{n}, \vec{u}^{n-1}, \vec{u}^{n-2}, \dots)$$

If f only depends on previous time steps, then the method is explicit. Also depending on the function value at the current time step $\vec{u}^{n+1}$ is called implicit. Semi-implicit will mean that f only depends on $\vec{u}^{n+1}$ linearly. For implicit methods, an extra step needs to be carried out to calculate $\vec{u}^{n+1}$. Most often this explicit expression cannot be found, so some root finding algorithm is used. Explicit methods can be evaluated directly, but suffer from instability as Δt gets too large.

This paper will focus on explicit finite difference methods. The next sections will summarise how to evaluate these methods. First, the difference methods should be consistent with the differential equation. Next, the time step integration should be stable. Both ensure that the solution of the difference equation converges to the solution of the differential equation, or $u_j^n \rightarrow u(t^n, x_j)$ as $\Delta t, \Delta x \rightarrow 0$.

A lot of theory can be visualised with numerical simulations. The numerical simulations are done on a grid of 100 nodes. The initial condition will be an array of all 0 and one node will have a value of 1. The mass and energy are defined as the L^1-norm and L^2-norm of $\vec{u}$ respectively. This will make the initial mass and energy both 1. Then $\frac{c\Delta t}{\Delta x}$ is varied and simulated for 100 time steps. The results of different tests will be showed throughout this thesis.

2.2.1 Consistency

The first thing that is necessary for convergence is consistency.

Definition 2.2 (Consistency)**.** A finite difference equation is called *consistent* with a differential equation, if by reducing the time step size and the spacial grid spacing the truncation error could be made to approach zero.

Consider solving the one dimensional advection equation with an explicit finite difference method. One such method could be upwind Euler forward (assuming $c > 0$):

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \rightarrow \frac{u_j^{n+1} - u_j^n}{\Delta t} + c\frac{u_j^n - u_{j-1}^n}{\Delta x} = 0$$

According to the Taylor expansions of all the terms

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \left.\frac{\partial u}{\partial t}\right|_{(t^n, x_j)} + \mathcal{O}(\Delta t)$$
$$\frac{u_j^n - u_{j-1}^n}{\Delta x} = \left.\frac{\partial u}{\partial x}\right|_{(t^n, x_j)} + \mathcal{O}(\Delta x)$$

Multiplying the second equation with c and adding the first equation yields

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + c\frac{u_j^n - u_{j-1}^n}{\Delta x} = \left.\frac{\partial u}{\partial t}\right|_{(t^n, x_j)} + c\left.\frac{\partial u}{\partial x}\right|_{(t^n, x_j)} + \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x)$$

So this shows that

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + c\frac{u_j^n - u_{j-1}^n}{\Delta x} \to \left.\frac{\partial u}{\partial t}\right|_{(t^n, x_j)} + c\left.\frac{\partial u}{\partial x}\right|_{(t^n, x_j)} \quad \text{as } \Delta t, \Delta x \to 0$$

So the finite difference method is consistent with the differential equation. These derivations are similar for a general velocity profile. It replaces all u with uv and removes c, but is still consistent with the advection equation.

2.2.2 Stability

Another important property of numerical integration methods is stability. The definition of stability depends on the context used. We will describe stability in two ways: Numerical stability and linear stability of maps. The first one is stability in the sense that when a rounding error is made, it will be damped and not increase in the following time steps. The second one is the linear stability analysis of maps, particularly maps that project a vector in $\mathbb{R}^d$ to another vector in $\mathbb{R}^d$. These maps have fixed points and solutions can converge to or diverge from this fixed point, defining stable or unstable fixed points.

Numerical stability
When a differential equation and the used difference equation are both linear and the domain has periodic boundary, Von Neumann stability analysis can be used (Charney et al., 1950). It assumes that calculations have finite precision, so numerical errors will occur. These errors are the difference between the exact numerical solution and the one obtained with finite precision arithmetic. Finite precision arithmetic is used by computers, that have to store values with finite precision, numerical errors occur. As the domain is periodic, the solution will be too and the numerical error will as well (Fletchet, 1991).

Definition 2.3 (Linear partial differential equation)**.** A *linear partial differential equation* is a differential equation that is linear in the governing function

and its derivatives. Take for example a function $u(x, y)$. A first order linear PDE will be of the form

$$a(x,y)\frac{\partial u}{\partial x} + b(x,y)\frac{\partial u}{\partial y} + c(x,y)u + d(x,y) = 0.$$

The essence of Von Neumann stability analysis is looking at the Fourier decomposition of this error. As the domain is periodic, this will be an exact decomposition. In one dimension, the Fourier modes will have wave number $k_m = \frac{\pi m}{L}$, with L the length of the domain. Due to the difference equation being linear with constant coefficients, each Fourier mode will satisfy the finite difference equation as well. Let's again look at upwind Euler forward difference scheme applied to the advection equation in one dimension.

$$u_j^{n+1} = \left(1 - \frac{c\Delta t}{\Delta x}\right) u_j^n + \frac{c\Delta t}{\Delta x} u_{j-1}^n. \tag{3}$$

A general mode has the following form:

$$\epsilon(x,t) = E(t)e^{ik_m x}.$$

Here E is the amplitude of the mode at a certain time t. For now this is assumed to be unknown. This yields

$$\begin{aligned}
\epsilon_j^{n+1} &= E((n+1)\Delta t)e^{ik_m j\Delta x}, \\
\epsilon_j^n &= E(n\Delta t)e^{ik_m j\Delta x}, \\
\epsilon_{j-1}^n &= E(n\Delta t)e^{ik_m (j-1)\Delta x}.
\end{aligned}$$

Plugging this ansatz into equation (3) yields

$$\begin{aligned}
E((n+1)\Delta t)e^{ik_m j\Delta x} &= \left(1 - \frac{c\Delta t}{\Delta x}\right) E(n\Delta t)e^{ik_m j\Delta x} + \frac{c\Delta t}{\Delta x} E(n\Delta t)e^{ik_m (j-1)\Delta x} \\
\frac{E((n+1)\Delta t)}{E(n\Delta t)} &= \left(1 - \frac{c\Delta t}{\Delta x}\right) + \frac{c\Delta t}{\Delta x} e^{-ik_m \Delta x} \\
\frac{E((n+1)\Delta t)}{E(n\Delta t)} &= \left(1 - \frac{c\Delta t}{\Delta x} + \frac{c\Delta t}{\Delta x}\cos\left(ik_m\Delta x\right)\right) - i\frac{c\Delta t}{\Delta x}\sin\left(ik_m\Delta x\right)
\end{aligned}$$

For the numerical error to stay bounded, the amplitude of the waves should not increase after every time step. That means the following condition needs to hold:

$$\left|\frac{E((n+1)\Delta t)}{E(n\Delta t)}\right| \leq 1.$$

This corresponds to

$$\sqrt{\left(1-\frac{c\Delta t}{\Delta x}\right)^2+\frac{c\Delta t}{\Delta x}^2+2\left(1-\frac{c\Delta t}{\Delta x}\right)\frac{c\Delta t}{\Delta x}\cos\left(\kappa\Delta x\right)}\leq 1$$

This is satisfied for all k_m if

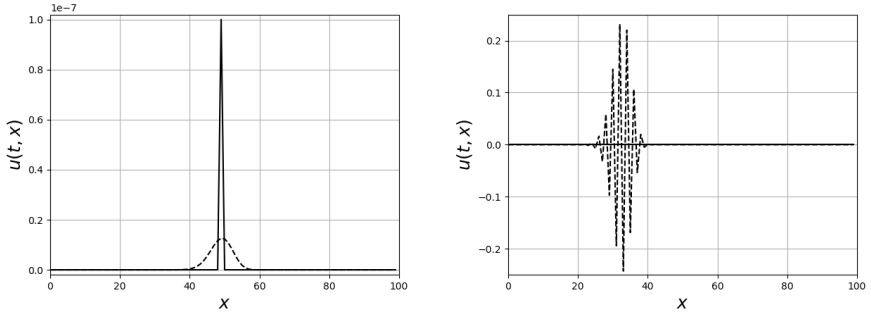
$$0\leq\frac{c\Delta t}{\Delta x}\leq 1. \tag{4}$$

It means this method can be stable when $c\geq 0$ and $\Delta t\leq\frac{\Delta x}{c}$, but will always be unstable when this limit is exceeded.

To show the validity of this analysis, consider the simulation where the initial condition is given by

$$\vec{u}^0=\begin{bmatrix}0\\ \vdots\\ 0\\ 1e-7\\ 0\\ \vdots\\ 0\end{bmatrix}$$

So a zero vector, but due to some small error, either computational or something else, one of the nodes gets the value $1e-7$. The Von Neumann stability region given by (4) would indicate where such an error would grow in size and where it would decay. Figure 2 shows a simulation of transport of this error over one round trip through the domain.



(a) $\frac{c\Delta t}{\Delta x}=0.9$ (b) $\frac{c\Delta t}{\Delta x}=1.1$

Figure 2: Simulation of error propagation over one full period of the domain. The unbroken line is the inital condition and the dashed line is the value of u after one full revolution.

Figure 2a has the result of the simulation for $\frac{c\Delta t}{\Delta x} = 0.9$. The initial error has shrunk in size and has diffused, which is expected for a stable finite difference advection scheme. Figure 2b shows the results for $\frac{c\Delta t}{\Delta x} = 0.9$. There the initial error has grown and also spread to neighbouring nodes. The order of the error has grown by order of $1e7$.

Stability of maps

When the numerical scheme is applied to all grid points simultaneously, it will be denoted by a matrix vector multiplication. For example, equation (3) would turn into

$$\vec{u}^{n+1} = A\vec{u}^n, \qquad A = I - \frac{c\Delta t}{\Delta x}\begin{bmatrix} 1 & 0 & \cdots & 0 & -1 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & & -1 & 1 & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}. \tag{5}$$

When this map A is applied to $\vec{u}^n$, the L^2-norm of $\vec{u}^n$ is required not to grow. Otherwise, applying this map over and over, lets the energy in the system grow without bound. The non-growth condition corresponds to

$$\frac{||\vec{u}^{n+1}||_2}{||\vec{u}^n||_2} \leq 1, \quad \forall \vec{u} \in \mathbb{R}^d.$$

This corresponds to the spectral radius of A being less or equal 1, or for the eigenvalues of A: $|\lambda_j| \leq 1$ for all $j = 0, \ldots, d-1$. For the eigenvalues on the unit circle, the requirement is that they are semisimple. The shape of this matrix allows for easy evaluation of the eigenvalues. Due to the periodic boundary condition and uniform velocity, the resulting matrix A of any applied finite difference method will be circulant.

Definition 2.4 (Circulant matrix)**.** A $d \times d$ matrix is called *circulant* if it is of the form

$$M = \begin{bmatrix} b_0 & b_1 & \cdots & b_{d-2} & b_{d-1} \\ b_{d-1} & b_0 & b_1 & & b_{d-2} \\ \vdots & b_{d-1} & b_0 & \ddots & \vdots \\ b_2 & & \ddots & \ddots & b_1 \\ b_1 & b_2 & \cdots & b_{d-1} & b_0 \end{bmatrix}.$$

Let $\vec{b}$ be the vector corresponding to the first row of M with entries $b_0, b_1, \ldots, b_{d-1} \in \mathbb{C}$. Then every row is this same vector, but shifted.

These matrices have explicitely known eigenvectors and eigenvalues, shown in the following theorem.

Theorem 2.5. *The normalized eigenvectors v_k of a circulant matrix are given by*

$$v_k = \frac{1}{\sqrt{d}}\left(1, \omega^k, \omega^{2k}, \ldots, \omega^{(d-1)k}\right),$$

where ω is a primary root of unity, i.e.

$$\omega = e^{\frac{2\pi i}{d}}.$$

The eigenvalues λ_k that correspond to the eigenvectors v_k are given by

$$\begin{aligned}\lambda_k &= \sum_{m=0}^{d-1} b_m \omega^{mk} \\ &= b_0 + b_1 e^{\frac{2\pi i}{d}k} + \cdots + b_{d-1} e^{\frac{2\pi i}{d}(d-1)k}\end{aligned}$$

Proof. Proof can be found in (Davis, 1979). □

Remark. Notice that the eigenvectors of circulant matrices don't depend on the entries of the matrix, therefore all circulant matrices have the same eigenvectors. Also notice that all eigenvalues and eigenvectors are distinct, so all eigenvalues are always semisimple.

Using this theorem, the eigenvalues of A are given by

$$\lambda_k = 1 - \frac{c\Delta t}{\Delta x} + \frac{c\Delta t}{\Delta x} e^{\frac{2\pi i}{d}(d-1)k}.$$

This yields

$$\begin{aligned}|\lambda_k|^2 &= \left(1 - \frac{c\Delta t}{\Delta x} + \frac{c\Delta t}{\Delta x}\cos\left(\frac{2\pi(d-1)k}{d}\right)\right)^2 + \left(\frac{c\Delta t}{\Delta x}\sin\left(\frac{2\pi(d-1)k}{d}\right)\right)^2 \\ &= \left(1 - \frac{c\Delta t}{\Delta x}\right)^2 + \left(\frac{c\Delta t}{\Delta x}\right)^2 + 2\left(1 - \frac{c\Delta t}{\Delta x}\right)\frac{c\Delta t}{\Delta x}\cos\left(\frac{2\pi(d-1)k}{d}\right)\end{aligned}$$

When

$$0 \leq \frac{c\Delta t}{\Delta x} \leq 1, \tag{6}$$

then

$$\left(1 - \frac{c\Delta t}{\Delta x}\right)\frac{c\Delta t}{\Delta x} \geq 0.$$

That means

$$
\begin{aligned}
|\lambda_k|^2 &= \left(1-\frac{c\Delta t}{\Delta x}\right)^2 + \left(\frac{c\Delta t}{\Delta x}\right)^2 + 2\left(1-\frac{c\Delta t}{\Delta x}\right)\frac{c\Delta t}{\Delta x}\cos\left(\frac{2\pi(d-1)k}{d}\right) \\
&\leq \left(1-\frac{c\Delta t}{\Delta x}\right)^2 + \left(\frac{c\Delta t}{\Delta x}\right)^2 + 2\left(1-\frac{c\Delta t}{\Delta x}\right)\frac{c\Delta t}{\Delta x} \\
&= \left(1-\frac{c\Delta t}{\Delta x}+\frac{c\Delta t}{\Delta x}\right)^2 \\
&= 1
\end{aligned}
$$

So all the eigenvalues always are either inside or on the unit circle when equation (6) is satisfied. For values outside of this range we have

$$
\left(1-\frac{c\Delta t}{\Delta x}\right)\frac{c\Delta t}{\Delta x} < 0.
$$

Remark that $d \geq 2$ for this method to be used. Take $k = 1$, then

$$
\frac{2\pi(d-1)}{d} \neq 2\pi n, \qquad n \in \mathbb{N}.
$$

So that means that

$$
\begin{aligned}
|\lambda_1|^2 &= \left(1-\frac{c\Delta t}{\Delta x}\right)^2 + \left(\frac{c\Delta t}{\Delta x}\right)^2 + 2\left(1-\frac{c\Delta t}{\Delta x}\right)\frac{c\Delta t}{\Delta x}\cos\left(\frac{2\pi(d-1)}{d}\right) \\
&> \left(1-\frac{c\Delta t}{\Delta x}\right)^2 + \left(\frac{c\Delta t}{\Delta x}\right)^2 + 2\left(1-\frac{c\Delta t}{\Delta x}\right)\frac{c\Delta t}{\Delta x} \\
&= \left(1-\frac{c\Delta t}{\Delta x}+\frac{c\Delta t}{\Delta x}\right)^2 \\
&= 1.
\end{aligned}
$$

In conclusion, because all the eigenvalues are semisimple, the map of equation (5) is stable for

$$
0 \leq \frac{c\Delta t}{\Delta x} \leq 1,
$$

which is the same region from the Von Neumann stability analysis. In this case the two analyses show the same results, because the eigenvectors are the Fourier modes. So there is a connection between the norm of the eigenvalues and the amplitudes of Fourier modes in the von Neumann analysis.

Another way to show this is to notice that all the eigenvalues

$$
\lambda_k = 1 - \frac{c\Delta t}{\Delta x} + \frac{c\Delta t}{\Delta x} e^{\frac{2\pi i}{d}(d-1)k}
$$

lie on a circle in the complex plane with centre and radius given respectively by

$$m_c = 1 - \frac{c\Delta t}{\Delta x}, \qquad r_c = \frac{c\Delta t}{\Delta x}.$$

This circle of eigenvalues is contained in the unit circle for

$$0 \leq \frac{c\Delta t}{\Delta x} \leq 1,$$

and exceeds the unit circle outside of this range. This is visualised in figure 3. For values outside of the stable range, the eigenvalues exceed the unit circle, either to to the left of 1, for $\frac{c\Delta t}{\Delta x} > 1$ or to the right of 1, for $\frac{c\Delta t}{\Delta x} < 0$.

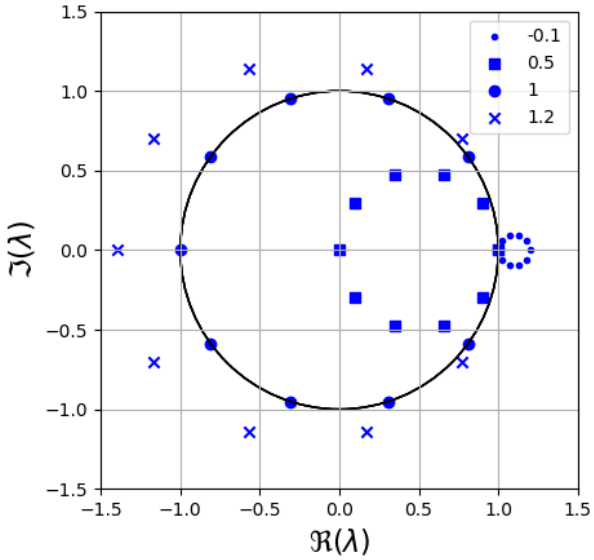


Figure 3: Eigenvalues of the map corresponding to integrating the advection equation with upwind Euler forward. There are 10 gridpoints and the value of $\frac{c\Delta t}{\Delta x}$ is varied between -0.1, 0.5, 1 and 1.2. The eigenvalues are plotted next to the unit circle indicated in black.

To see this for more values of $\frac{c\Delta t}{\Delta x}$, figure 4 shows the biggest absolute value, or spectral radius, of the matrix. It shows that the spectral radius is equal to 1 in the expected region and bigger than 1 outside of this region. This shows where the map is stable and unstable.

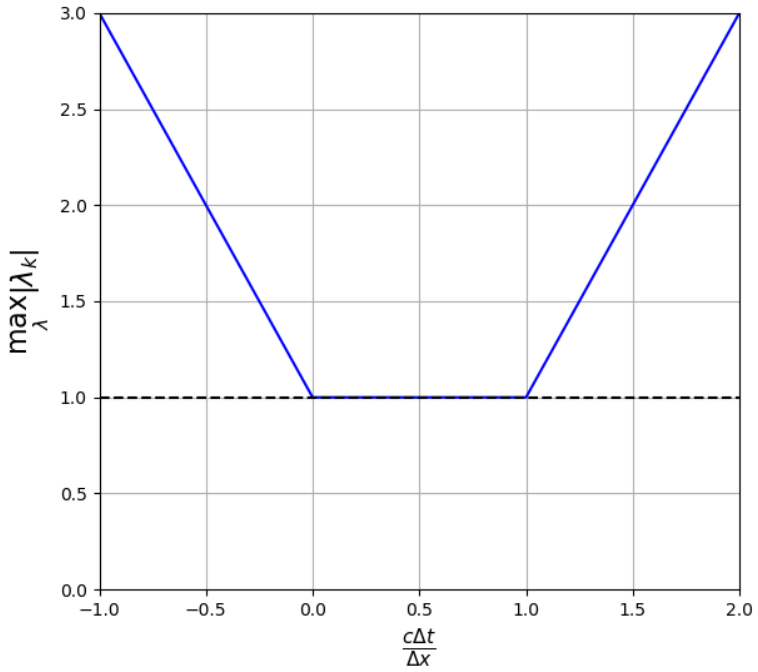


Figure 4: The biggest absolute value of all the eigenvalues of the map. This is the map from applying the upwind Euler forward to the advection equation. This value is plotted against the value of $\frac{c\Delta t}{\Delta x}$, varying between -1 and 2.

2.2.3 Convergence

In 1956, Lax and Richtmyer showed that for well-posed linear initial value problem and a difference equation that is consistent with it, stability is a necessary and sufficient condition for the numerical solution of the difference equation to converge to the analytical solution of the differential equation (Lax & Richtmyer, 1956). The advection equation with uniform constant velocity is such a problem and the consistency and stability of the upwind Euler forward method are shown in the section before. Therefore, the numerical solution of that finite difference equation will converge to the solution of the advection equation for

$$0 \leq \frac{c\Delta t}{\Delta x} \leq 1.$$

CFL-condition
For a numerical solution of a finite difference approximation to converge to the analytical solution of the hyperbolic differential equation, it is a necessary condition for the analytical domain of dependence to be contained in the numerical

domain of dependence (Courant et al., 1928). This condition is called the CFL-condition. The authors of this paper showed that when this is reversed, the solution cannot converge. This reversal means that the solution of the differential equation depends on information outside of the domain of dependence of the difference equation. This information cannot also influence the solution of the difference equation, as it lies outside of its domain of dependence. The solution of the difference equation misses information necessary for it to converge to the analytical solution as the time step size and grid spacing both approach zero.

Definition 2.6 (Domain of dependence)**.** The *analytical domain of dependence* of a point (t, x) in space-time is the domain in space-time that influences its value, i.e. all the point necessary for the determination of its value. The *numerical domain of dependence* of a grid point (t_{n+1}, x_j) in space-time is the smallest domain in space-time that includes the grid points influencing it.

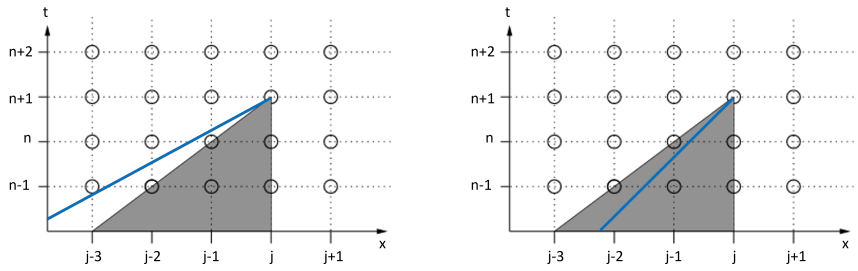
Let's continue with the one-dimensional advection equation. The point (x_j, t_{n+1}) has a different domain of dependence for the differential equation and the difference equation. The first one is equal to the characteristic line through (x_j, t_{n+1}): $t(x) = (x - x_j)/c + t_{n+1}$. The second one is a little more difficult to see. The value of u_j^{n+1} in

$$u_j^{n+1} = \left(1 - \frac{c\Delta t}{\Delta x}\right) u_j^n + \frac{c\Delta t}{\Delta x} u_{j-1}^n$$

depends on the value of u_j^n and u_{j-1}^n. These two in turn depend on u_j^{n-1}, u_{j-1}^{n-1} and u_{j-2}^{n-1} via the same equation. This will continue until the initial condition is reached. The numerical domain of dependence is now the triangle containing all these points, i.e. the lower triangle between the lines

$$x(t) = x_j \qquad \text{and} \qquad x(t) = x_j + (t - t^n)\frac{\Delta x}{\Delta t}$$

Figure 5a and 5b show the CFL-condition not satisfied and satisfied respectively. Figure 5a shows the characteristic line, or analytical domain of dependence, to be outside of the numerical domain of dependence. Figure 5b shows it to be inside.



(a) CFL condition not satisfied.

(b) CFL condition satisfied.

Figure 5: The CFL-condition for the one dimensional advection equation with uniform constant velocity $c > 0$. The analytical domain of dependence is visualised with a blue line and the numerical domain of dependence is shown as the greyed out area (Willcox & Wang, n.d.).

The CFL condition in this specific case can be written as

$$0 \leq c \leq \frac{\Delta x}{\Delta t} \quad \rightarrow \quad 0 \leq \frac{c\Delta t}{\Delta x} \leq 1.$$

So for one dimensional advection with uniform constant positive velocity, numerically integrated with upwind Euler forward, the CFL condition covers the same region as the stability region. This immediately confirms that the convergence is lost outside of the CFL region, as stability is a necessary condition for convergence.

Furthermore, is seems that in this case the stability is a sufficient condition for convergence. This result is expected, because this is a well-posed linear initial value problem.

2.2.4 Conservation of mass and energy

In the discrete case, the mass and energy of the system are represented respectively by

$$\sum_{j=0}^{d-1} u_j \equiv ||\vec{u}||_1 \qquad \text{and} \qquad \sum_{j=0}^{d-1} u_j^2 \equiv ||\vec{u}||_2^2.$$

First using equation (3), the mass can be evaluated to be

$$\begin{aligned}
\sum_{j=0}^{d-1} u_j^{n+1} &= \left(1 - \frac{c\Delta t}{\Delta x}\right) \sum_{j=0}^{d-1} u_j^n + \frac{c\Delta t}{\Delta x} \sum_{j=0}^{d-1} u_{j-1}^n \\
&= \left(1 - \frac{c\Delta t}{\Delta x}\right) \sum_{j=0}^{d-1} u_j^n + \frac{c\Delta t}{\Delta x} \sum_{j=0}^{d-1} u_j^n \\
&= \left(1 - \frac{c\Delta t}{\Delta x} + \frac{c\Delta t}{\Delta x}\right) \sum_{j=0}^{d-1} u_j^n \\
&= \sum_{j=0}^{d-1} u_j^n.
\end{aligned}$$

The second equality follows from the periodic boundary conditions. This shows that the mass stays the same from time step to time step. Just as in the analytical case, the mass is conserved.

Figure 6 is the result of simulations of equation (3) for different values of $\frac{c\Delta t}{\Delta x}$.

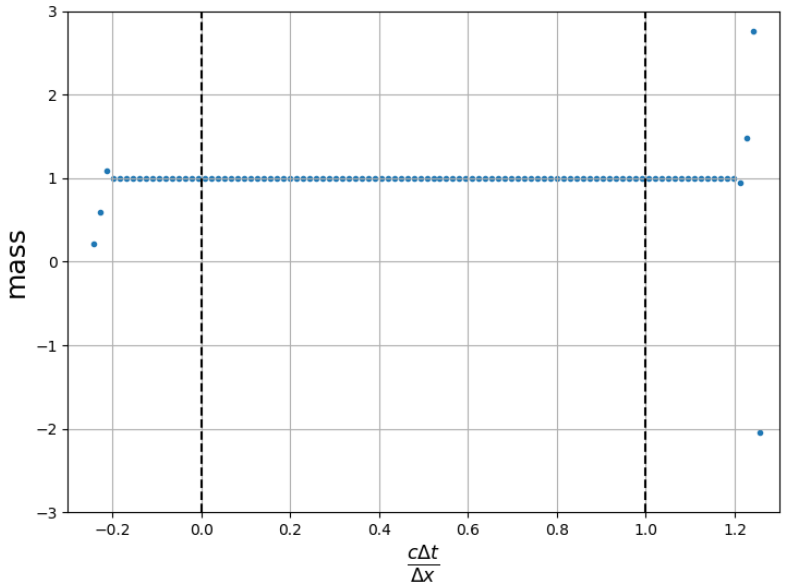


Figure 6: Mass of $\vec{u}$ after 100 time steps for different values of $\frac{c\Delta t}{\Delta x}$. The boundaries of the stable region are indicated by the dashed black lines. Values of $\frac{c\Delta t}{\Delta x}$ without a corresponding mass value means that these points were too big in size to plot on a reasonably scaled plot.

In the region $-0.2 \leq \frac{c\Delta t}{\Delta x} \leq 1.2$, the mass seems to be conserved after 100 time steps. Outside of this region, the mass changed. In theory, the numerical method should be mass conserving, but the numerical instability changes that. Due to rounding errors blowing up, the mass can still increase or decrease. In figure 7, the results can be seen for 1000 time steps, where the region of stability further coincides with the region of mass conservation. The rounding errors now also have had time to blow up in the regions just left of the left dashed line and right of the right dashed line.

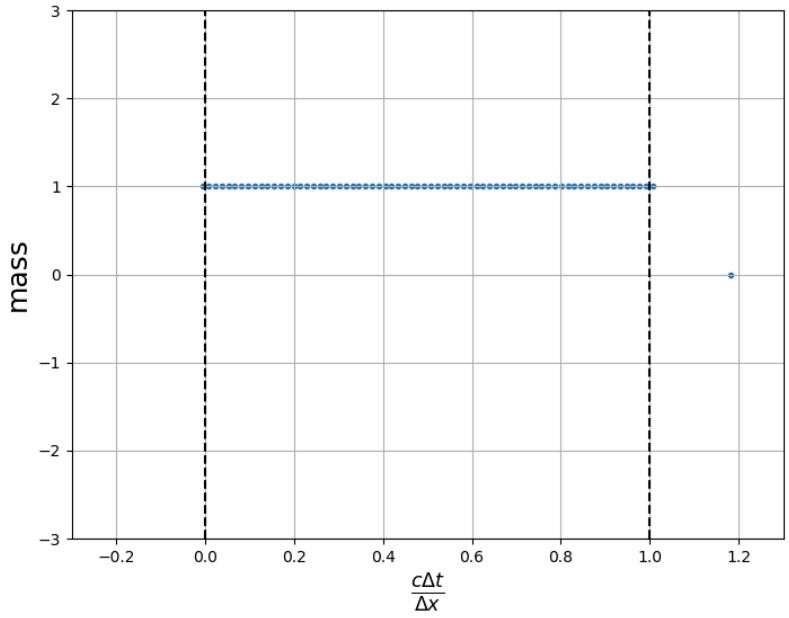


Figure 7: Mass of $\vec{u}$ after 1000 time steps for different values of $\frac{c\Delta t}{\Delta x}$. The boundaries of the stable region are indicated by the dashed black lines. Values of $\frac{c\Delta t}{\Delta x}$ without a corresponding mass value means that these points were too big in size to plot on a reasonably scaled plot.

The energy of $\vec{u}^{n+1}$, according to equation (3), is given by

$$\begin{aligned}
\sum_{j=0}^{d-1}\left(u_j^{n+1}\right)^2 &= \sum_{j=0}^{d-1}\left(\left(1-\frac{c\Delta t}{\Delta x}\right)u_j^n + \frac{c\Delta t}{\Delta x}u_{j-1}^n\right)^2 \\
&= \sum_{j=0}^{d-1}\left(\left(1-\frac{c\Delta t}{\Delta x}\right)^2\left(u_j^n\right)^2 + \left(\frac{c\Delta t}{\Delta x}\right)^2\left(u_{j-1}^n\right)^2\right) \\
&\quad + 2\sum_{j=0}^{d-1}\left(\left(1-\frac{c\Delta t}{\Delta x}\right)\frac{c\Delta t}{\Delta x}u_j^n u_{j-1}^n\right) \\
&= \left(\left(1-\frac{c\Delta t}{\Delta x}\right)^2 + \left(\frac{c\Delta t}{\Delta x}\right)^2\right)\sum_{j=0}^{d-1}\left(u_j^n\right)^2 \\
&\quad + 2\left(1-\frac{c\Delta t}{\Delta x}\right)\frac{c\Delta t}{\Delta x}\sum_{j=0}^{d-1}u_j^n u_{j-1}^n \\
\frac{\sum_{j=0}^{d-1}\left(u_j^{n+1}\right)^2}{\sum_{j=0}^{d-1}\left(u_j^n\right)^2} &= \left(1-\frac{c\Delta t}{\Delta x}\right)^2 + \left(\frac{c\Delta t}{\Delta x}\right)^2 + 2\left(1-\frac{c\Delta t}{\Delta x}\right)\frac{c\Delta t}{\Delta x}\frac{\sum_{j=0}^{d-1}u_j^n u_{j-1}^n}{\sum_{j=0}^{d-1}\left(u_j^n\right)^2}
\end{aligned}$$

So energy is generally not conserved. The last equation is always equal to 1 for a constant $\vec{u}$, which is trivial. In the case of non constant data, the ratio between sums in the last term can be understood as a ratio between autocorrelation with lag 1 and lag 0. Because this ratio is always between -1 and 1, the energy is non-increasing for all $\vec{u}$ if

$$0 \leq \frac{c\Delta t}{\Delta x} \leq 1.$$

It will be perfectly energy conserving if

$$\frac{c\Delta t}{\Delta x} = 0 \quad \vee \quad \frac{c\Delta t}{\Delta x} = 1.$$

The first one is trivial and the second case is the case where the information is shifted one node along every time step. This energy region can be seen in figure 8.

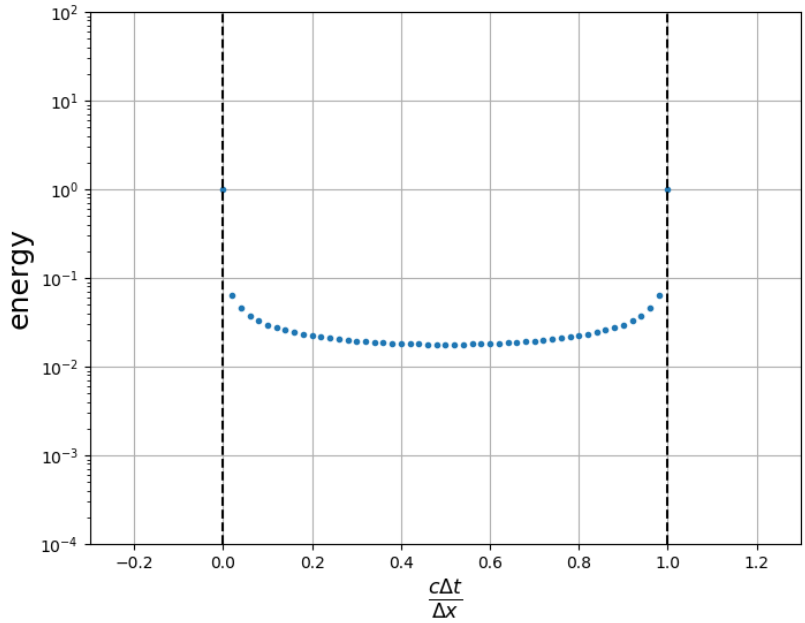


Figure 8: Energy of $\vec{u}$ after 1000 time steps with $\Delta t = 1$, $\Delta x = 1$ and values of c varying between -0.3 and 1.3. The result is plotted on a logarithmic scale for the different values of $\frac{c\Delta t}{\Delta x}$. The boundaries of the stable region are indicated by the dashed black lines. Data points outside of this stable region have a too large value of the energy to be plotted.

The energy decreases inside of this region, the energy is conserved at the boundaries and the energy increases without bound outside of this region.

3 Semi-Lagrangian advection

Instead of fixing the coordinate frame and calculating the local change of a scalar due to its transport, one could also follow a parcel of air containing this scalar quantity. In this way there is no flow in and out of the parcel, so the amount of the scalar in our parcel remains constant. This way only the 'route' this parcel takes needs to be calculated. This trajectory then fully determines the advection.

Definition 3.1. Lagrangian advection equation] The advection of some property $u(t, \vec{x})$ in a velocity field $\vec{v}(t, \vec{x})$ is governed by the advection equation in Lagrangian form

$$\frac{du}{dt} = 0, \quad \frac{d\vec{x}}{dt} = \vec{v}\,(t, \vec{x})\,. \tag{7}$$

Here $\frac{d}{dt} = \frac{\partial}{\partial t} + \vec{v} \cdot \vec{\nabla}$ is the material derivative, defining the link between the Lagrangian and the Eulerian frame of reference.

The assumption in this equation is that u undergoes material advection, i.e. is constant in an air parcel. The material derivative of u is zero, meaning constant inside the followed air parcel. Using Lagrangian methods, integrating this equation on a finite grid will conserve the initial condition but change the coordinates of the air parcels. So the grid will morph according to the velocity field. Now the method needs to keep track of changing coordinates, which is computationally very heavy. Furthermore, the grid very quickly becomes chaotic.

The way the Lagrangian method can be used effectively is by combining it with Eulerian principles. That is what are called Semi-Lagrangian methods. Instead of starting at the departure points on a grid, these methods start with the arrival points on a grid. This means, that to find the value of u at an arrival point $\vec{x}_A$, one needs to find the value of u along the same trajectory Δt time units earlier. This point is then called the departure point $\vec{x}_D$. Finding the value of u at $\vec{x}_A$ consists of two problems: 1. Finding the departure point $\vec{x}_D$ and 2. Determining the value of u at $\vec{x}_D$ and setting that as the new value of u at $\vec{x}_A$.

3.1 Determining the departure point

Determining the departure point at time t from an arrival point x_A at time $t + \Delta t$ is done by solving the following equation:

$$\vec{x}_D = \vec{x}_A - \int_t^{t+\Delta t} \vec{v}(\tau, \vec{x}(\tau))d\tau.$$

The average of $\vec{v}(\tau, \vec{x}(\tau))$ in $\tau \in [t, t + \Delta t]$ can be defined as

$$\bar{\vec{v}}(t, \Delta t) = \frac{1}{\Delta t} \int_t^{t+\Delta t} \vec{v}(\tau, \vec{x}(\tau))d\tau.$$

So then the departure point problem is the solution to

$$\vec{x}_D = \vec{x}_A - \Delta t \vec{\tilde{v}}(t, \Delta t).$$

This average can be approximated by various choices of function, but not every choice will be as accurate or easy to calculate. In the analytical case, the average is calculated along the trajectory, but that is generally unknown. Approximations will use the value of $\vec{v}$ at certain t and $\vec{x}$. With an unknown trajectory, the value of $\vec{v}$ is usually taken at $\vec{x_D}$ and $\vec{x_A}$. The time at which the values are taken is chosen a priori and the arrival point is also already known. The value of $\vec{\tilde{v}}$ in a single departure point problem then only depends on the unknown departure point itself: $\vec{\tilde{v}}(\vec{x}_D)$.

3.1.1 Existence

The existence of a departure point is important for all applications of Semi-Lagrangian methods (i.e. numerical weather prediction). A departure point $\vec{x}_D \in \mathbb{R}^d$ should be a solution of the following equation with $\vec{x}, \vec{x}_A, \vec{\tilde{v}} \in \mathbb{R}^d$:

$$\vec{x} = \vec{x}_A - \Delta t \vec{\tilde{v}}\left(\vec{x}\right).$$

This would be the same as finding the roots of

$$f_i\left(x_1, \ldots, x_d\right) = \frac{1}{\Delta t}\left(x_i - x_{A,i}\right) + \tilde{v}_i\left(x_1, \ldots, x_d\right) = 0, \quad \forall i = 1, \ldots, d.$$

where

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}, \quad \vec{x}_A = \begin{bmatrix} x_{A,1} \\ \vdots \\ x_{A,d} \end{bmatrix} \quad \text{and} \quad \vec{\tilde{v}} = \begin{bmatrix} \tilde{v}_1 \\ \vdots \\ \tilde{v}_d \end{bmatrix}.$$

Theorem 3.2 (Poincaré–Miranda)**.** *Consider the rectangle* $\mathfrak{R} \in \mathbb{R}^d$

$$\mathfrak{R} = [a_1, b_1] \times \ldots \times [a_d, b_d],$$

and consider d *continuous functions of* d *variables,* $f_1, \ldots, f_d$*. Assume that for each variable* x_i *the function* f_i *is non-positive when* $x_i = a_i$ *and non-negative when* $x_i = b_i$*. Then there exists a point* $\boldsymbol{x} \in \mathfrak{R}$ *such that* $f_i(\boldsymbol{x}) = 0$ *for all* $i = 1, \ldots, d$*.*

Proof. Proof can be found in (Miranda, 1940). □

Corollary 3.2.1. *Suppose the velocity field* $\vec{\tilde{v}}$ *is bounded in each component, i.e. there exist* $\tilde{v}_{i,min}$ *and* $\tilde{v}_{i,max}$ *such that* $\tilde{v}_{i,min} \leq \tilde{v}_i \leq \tilde{v}_{i,max}$ *for all* $i = 1, \ldots, d$*, then there exists at least one solution* $\vec{x} \in [x_{A,1} - \Delta t \tilde{v}_{1,max}, x_{A,1} - \Delta t \tilde{v}_{1,min}] \times \ldots \times [x_{A,d} - \Delta t \tilde{v}_{d,max}, x_{A,d} - \Delta t \tilde{v}_{d,min}]$ *to the departure point problem.*

Proof. Take $\mathfrak{R} = [x_{A,1} - \Delta t \tilde{v}_{1,max}, x_{A,1} - \Delta t \tilde{v}_{1,min}] \times \ldots \times [x_{A,d} - \Delta t \tilde{v}_{d,max}, x_{A,d} - \Delta t \tilde{v}_{d,min}]$. We also have

$$f_i(x_1, \ldots, x_d) = \frac{1}{\Delta t}(x_i - x_{A,i}) + \tilde{v}_i(x_1, \ldots, x_d), \quad \forall i = 1, \ldots, d.$$

This gives for $x_i = x_{A,i} - \Delta t \tilde{v}_{i,max}$:

$$\begin{aligned} f_i(x_1, \ldots, x_{A,i} - \Delta t \tilde{v}_{i,max}, \ldots, x_d) &= -\tilde{v}_{i,max} + \tilde{v}_i(x_1, \ldots, x_d) \\ &\leq 0, \quad \forall i = 1, \ldots, d. \end{aligned}$$

And for $x_i = x_{A,i} - \Delta t \tilde{v}_{i,min}$:

$$f_i(x_1, \ldots, x_{A,i} - \Delta t \tilde{v}_{i,min}, \ldots, x_d) = -\tilde{v}_{i,min} + \tilde{v}_i(x_1, \ldots, x_d) \geq 0, \quad \forall i = 1, \ldots, d.$$

Now according to the Poincaré-Miranda theorem, there is at least one $\vec{x} \in \mathfrak{R}$ such that $f_i(\vec{x}) = 0$ for all $i = 1, \ldots, d$, which is a solution to the departure point problem. □

3.1.2 Uniqueness

The existence of a departure point doesn't necessarily proof uniqueness of this departure point. In the previous part we showed that there is at least 1 solution to the departure point problem whenever the velocity is bounded in every dimension. Let's say there is at least one departure point $\vec{x}_D$. When there is an $i \in \{1, \ldots, d\}$ such that

$$\max_{j=1,\ldots,d} \left. \frac{\partial \tilde{v}_i}{\partial x_j} \right|_{\vec{x}_D} < -\frac{1}{\Delta t},$$

then there are always at least 2 more departure points. This also means that if

$$\max_{\vec{x} \in \mathbb{D}} \max_{j=1,\ldots,d} \left. \frac{\partial \tilde{v}_i}{\partial x_j} \right|_{\vec{x}} > -\frac{1}{\Delta t},$$

then the departure point is unique in $\mathbb{D}$.

3.1.3 Convergence

Calculating the departure point can be done in multiple ways. If $\vec{\tilde{v}}$ can be expressed as a single function, there is a chance that the departure point can be found analytically. For example, if $\vec{\tilde{v}} = \vec{x}$, then the departure point is given by

$$\vec{x}_D = \frac{1}{1 + \Delta t} \vec{x}_A$$

More generally, the departure point cannot be found analytically. Normally the velocity is only known on a discrete grid, so there will not be an expression

possible for a function. In all the cases where the departure point cannot be found analytically, other methods need to be used. The most used method is that of the iterative method. This entails the following:

$$\vec{x}_{k+1} = \vec{x}_A - \Delta t \vec{\vec{v}}\left(\vec{x}_k\right),$$

where $\vec{x}_0 = \vec{x}_A$ can be used as initial guess. Given convergence of this method, as $k \to \infty$, we have $\vec{x}_k \to \vec{x}_D$.

Theorem 3.3. *If a fixed point iteration is given by*

$$\vec{x}_{k+1} = \vec{F}\left(\vec{x}_k\right),$$

with F a matrix operator, and there exists a $\sigma < 1$ such that

$$||J_{\vec{F}}\left(\vec{x}\right)|| \leq \sigma, \quad \forall \vec{x} \in \mathbb{D},$$

then the fixed point iteration converges to fixed point $\vec{x}^*$ for any initial guess $\vec{x}_0 \in \mathbb{D}$. Here J is the Jacobian matrix and $||\cdot||$ is a natural matrix norm.

Remark. As the spectral radius is bounded by every matrix norm, the matrix norm in this theorem could also be replaced by the spectral radius of the Jacobian matrix. This way the eigenvalues can be analysed instead of the matrix norm.

In the departure point problem, initial condition $\vec{x}_0 \in \mathbb{D}$ converges to the fixed point if

$$\rho\left(J_{\vec{v}}\left(\vec{x}\right)\right) \leq ||J_{\vec{v}}\left(\vec{x}\right)|| < \frac{1}{\Delta t}, \quad \forall \vec{x} \in \mathbb{D}.$$

Notice that this is a similar condition as for uniqueness of the departure point. This condition is often checked for in numerical weather simulations, so it will be used as an upper limit on the time step.

3.1.4 SETTLS

To use the fixed point iteration to find the departure point $\vec{x}_D$, an approximation for the average velocity along the trajectory $\vec{\vec{v}}$ has to be used. The HARMONIE model at KNMI uses the so called SETTLS scheme.

Definition 3.4 (SETTLS)**.** To calculate the departure point of the Lagrangian trajectory integration, the SETTLS (Stable Extrapolation Two Time Level Scheme) scheme uses the iterative scheme

$$\begin{cases} \vec{x}^0 & = \vec{x}_A - \frac{\Delta t}{2}\left(3\vec{v}(t, \vec{x}_A) - \vec{v}(t - \Delta t, \vec{x}_A)\right) \\ \vec{x}^{k+1} & = \vec{x}_A - \frac{\Delta t}{2}\left(2\vec{v}(t, \vec{x}^k) - \vec{v}(t - \Delta t, \vec{x}^k) + \vec{v}(t, \vec{x}_A)\right) \qquad \text{k} \geq 0 \end{cases} \tag{8}$$

Where $\vec{x}_A$ is the arrival point at time $t + \Delta t$ and $\vec{x}^{k+1}$ is an approximation of the departure point $\vec{x}_D$ at time t after k iterations. Here $\vec{v}(t, \vec{x}^k)$ and $\vec{v}(t - \Delta t, \vec{x}^k)$ will be interpolated from values at grid points around $\vec{x}^k$.

Existence
The SETTLS scheme uses the following approximation to $\vec{\tilde{v}}$:

$$\begin{aligned}\vec{\tilde{v}}\left(\vec{x}\right) &= \frac{1}{2}\left(\vec{v}\left(t+\Delta t,\vec{x}\right)+\vec{v}(t,\vec{x}_A)\right)\\ &= \frac{1}{2}\left(2\vec{v}(t,\vec{x})-\vec{v}(t-\Delta t,\vec{x})+\vec{v}(t,\vec{x}_A)\right)\end{aligned}$$

As $\vec{v}$ is bounded at all time t, this equation for $\vec{\tilde{v}}$ is also bounded. Therefore, there exists a departure point that can be found with the SETTLS scheme.

Uniqueness
A departure point $\vec{x}_D \in \mathbb{D}$ of the SETTLS scheme is unique in $\mathbb{D}$ when

$$\max_{\vec{x}\in\mathbb{D}} \max_{j=1,\ldots,d} \left.\frac{\partial \tilde{v}_i}{\partial x_j}\right|_{\vec{x}} > -\frac{1}{\Delta t},$$
$$\max_{\vec{x}\in\mathbb{D}} \max_{j=1,\ldots,d} \left(\left.\frac{\partial v_i}{\partial x_j}\right|_{(t,\vec{x})} - \frac{1}{2}\left.\frac{\partial v_i}{\partial x_j}\right|_{(t-\Delta t,\vec{x})} \right) > -\frac{1}{\Delta t}.$$

In one dimension this means

$$\max_{\vec{x}\in\mathbb{D}} \left(\left.\frac{\partial v}{\partial x}\right|_{(t,\vec{x})} - \frac{1}{2}\left.\frac{\partial v}{\partial x}\right|_{(t-\Delta t,\vec{x})} \right) > -\frac{1}{\Delta t}.$$

Convergence
The fixed point iteration using the SETTLS scheme always converges to a departure point $\vec{x}_D \in \mathbb{D}$ when

$$||J_{\vec{v}}\left(\vec{x}\right)|| < \frac{1}{\Delta t}, \quad \forall \vec{x} \in \mathbb{D}.$$

Which in one dimension reads

$$\left| \left.\frac{\partial v}{\partial x}\right|_{(t,\vec{x})} - \frac{1}{2}\left.\frac{\partial v}{\partial x}\right|_{(t-\Delta t,\vec{x})} \right| < \frac{1}{\Delta t}, \quad \forall \vec{x} \in \mathbb{D},$$
$$-\frac{1}{\Delta t} < \left.\frac{\partial v}{\partial x}\right|_{(t,\vec{x})} - \frac{1}{2}\left.\frac{\partial v}{\partial x}\right|_{(t-\Delta t,\vec{x})} < \frac{1}{\Delta t}, \quad \forall \vec{x} \in \mathbb{D}.$$

This interval is smaller than the interval for uniqueness. Should the criterion for convergence be met, then the fixed point is always unique.

Another way to prove this is by introducing

$$x^k = x_A - \Delta x^k, \qquad x_D = x_A - \Delta x^*$$

The SETTLS scheme can then be written as

$$\begin{aligned} x_A - \Delta x^{k+1} &= x_A - \frac{\Delta t}{2}\left(2v\left(t, x_A - \Delta x^k\right) - v\left(t - \Delta t, x_A - \Delta x^k\right) + v\left(t, x_A\right)\right) \\ \Delta x^{k+1} &= \frac{\Delta t}{2}\left(2v\left(t, x_A - \Delta x^k\right) - v\left(t - \Delta t, x_A - \Delta x^k\right) + v\left(t, x_A\right)\right) \end{aligned} \tag{9}$$

And for the departure point we have

$$\begin{aligned} x_A - \Delta x^* &= x_A - \frac{\Delta t}{2}\left(2v\left(t, x_A - \Delta x^*\right) - v\left(t - \Delta t, x_A - \Delta x^k\right) + v\left(t, x_A\right)\right) \\ \Delta x^* &= \frac{\Delta t}{2}\left(2v\left(t, x_A - \Delta x^*\right) - v\left(t - \Delta t, x_A - \Delta x^k\right) + v\left(t, x_A\right)\right) \end{aligned} \tag{10}$$

The Taylor series expansions of $v(t, x_A - \Delta x^k)$ and $v(t - \Delta t, x_A - \Delta x^k)$ around $x = x_A - \Delta x^*$ are given respectively by

$$\begin{aligned} v(t, x_A - \Delta x^k) &= v(t, x_A - \Delta x^*) + (\Delta x^* - \Delta x^k)\left.\frac{\partial v}{\partial x}\right|_{(t,\chi_1)}, \\ v(t - \Delta t, x_A - \Delta x^k) &= v(t - \Delta t, x_A - \Delta x^*) + (\Delta x^* - \Delta x^k)\left.\frac{\partial v}{\partial x}\right|_{(t-\Delta t,\chi_2)}. \end{aligned}$$

Here χ_1 and χ_2 are the values between $x_A - \Delta x^k$ and $x_A - \Delta x^*$ for which these expansions are exact. Plugging these equations into (9) yields

$$\begin{aligned} \Delta x^{k+1} = \frac{\Delta t}{2}\Bigg(&2v(t, x_A - \Delta x^*) + 2(\Delta x^* - \Delta x^k)\left.\frac{\partial v}{\partial x}\right|_{(t,\chi_1)} \\ &- v(t - \Delta t, x_A - \Delta x^*) - (\Delta x^* - \Delta x^k)\left.\frac{\partial v}{\partial x}\right|_{(t-\Delta t,\chi_2)} + v\left(t, x_A\right)\Bigg). \end{aligned}$$

Using equation (10) this can be rewritten to

$$\Delta x^{k+1} = \Delta x^* + \frac{\Delta t}{2}\left(2(\Delta x^* - \Delta x^k)\left.\frac{\partial v}{\partial x}\right|_{(t,\chi_1)} - (\Delta x^* - \Delta x^k)\left.\frac{\partial v}{\partial x}\right|_{(t-\Delta t,\chi_2)}\right).$$

Now introduce $\epsilon^k = \Delta x^* - \Delta x^k$, so the equation becomes

$$-\epsilon^{k+1} = \frac{\Delta t}{2}\left(2\epsilon^k\left.\frac{\partial v}{\partial x}\right|_{(t,\chi_1)} - \epsilon^k\left.\frac{\partial v}{\partial x}\right|_{(t-\Delta t,\chi_2)}\right)$$

The absolute value of this ϵ has to shrink every iteration for the scheme to converge:

$$\frac{\left|\epsilon^{k+1}\right|}{|\epsilon^k|} = \frac{\Delta t}{2}\left| 2\left.\frac{\partial v}{\partial x}\right|_{(t,\chi_1)} - \left.\frac{\partial v}{\partial x}\right|_{(t-\Delta t,\chi_2)} \right| < 1$$

$$\left| \left.\frac{\partial v}{\partial x}\right|_{(t,\chi_1)} - \frac{1}{2}\left.\frac{\partial v}{\partial x}\right|_{(t-\Delta t,\chi_2)} \right| < \frac{1}{\Delta t}$$

This inequality is a bit less strict then via the Jacobian matrix, as χ_1 and χ_2 are some unique value.

3.2 Interpolation

When the departure point is found, chances are that it will not exactly coincide with a grid point. The value at the departure point has to be interpolated with the values surrounding the departure point. The more points used, the more accurate the interpolation becomes. For example, zero-th order interpolation would be the value of the nearest point, first order would be a linear function through the direct neighbours, etc. The interpolation technique used for finding the value at the departure point would be characterised by spline interpolation. The location of the departure point determine what interpolation nodes are used. So the interpolation polynomial is distinct on each grid segment (x_j, x_{j+1}).

One of the most well known formulas for finding the interpolation polynomial, uses Lagrange polynomials. Assume we want to find the value of u at the departure point x_D in one dimension. The interpolation data $\{u_j\}$ are known at interpolation nodes $\{x_j\}$, for $j = 0, \ldots, m$. The value of $u(x_D)$ is then given by

$$u(x_D) = \sum_{j=0}^{m} l_j(x_D)u_j, \qquad l_j(x) = \prod_{\substack{i=0 \\ i\neq j}}^{m} \frac{(x - x_i)}{(x_j - x_i)},$$

where l_j are called the Lagrange basis polynomials. The polynomial $l_j(x)$ takes value 1 at x_j and 0 at all the other nodes. For problems in more dimensions, other interpolation techniques may be used.

4 An interpretation of Semi-Lagrangian methods as finite difference methods

Integrating the advection equation using Semi-Langrangian methods is not bound by stability criteria like the explicit Eulerian finite difference methods. Time steps may be arbitrarily large, with the only downside being the accuracy. We have seen that convergence of the fixed point iteration for the departure point may impose a time step size restriction, e.g. the SETTLS method only converges for time steps smaller than the reciprocal of the velocity gradient. This is typically in the range of $1e4$ seconds. Alternative root finding methods, such as the bisection method, have no such restriction. For HARMONIE with a grid spacing of 2.5km and typical maximum atmospheric velocities of 100 kilometers an hour, the CFL-condition implies a step size of 90 seconds. This shows the advantage of Semi-Lagrangian methods over explicit Eulerian finite difference methods. In this section we try to explain these advantages from the perspective of Eulerian finite difference methods. By interpreting the Semi-Lagrangian methods as Eulerian methods and applying the analysis that is normally used on Eulerian methods. The Semi-Lagrangian methods allow larger time steps like implicit finite difference methods, but don't suffer from large dispersion errors like the implicit methods.

4.1 Consistency

First we will show that the Semi-Lagrangian scheme can be viewed as a Eulerian finite difference method. To make this argument, the scheme needs to be consistent with the Eulerian advection equation. Recall both advection equations, Eulerian and Lagrangian respectively.

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{11}$$

$$\frac{du}{dt} = 0, \qquad \frac{dx}{dt} = c \tag{12}$$

We give some examples of Semi-Lagrangian schemes rewritten as finite difference schemes:

Example 4.1 (First order)**.** Consider the departure point corresponding to arrival point x_j in the case of uniform velocity:

$$x_D = x_j - c\Delta t$$

In the Lagrangian frame, the solution of the advection equation is as follows:

$$u_j^{n+1} = l(t^n, x_D),$$

where l is the interpolated value of u at time t^n, using Lagrange polynomials. To linearly interpolate the value of u at this departure point, one needs to know

between which two gridpoints x_D is situated. Introduce p, which is calculated to be

$$p = \left\lfloor \frac{c\Delta t}{\Delta x} \right\rfloor.$$

Then x_D is between x_{j-p-1} and x_{j-p}. Linearly interpolating gives

$$\begin{aligned} l(t, x_D) &= \frac{x_D - x_{j-p-1}}{x_{j-p} - x_{j-p-1}} u^n_{j-p} + \frac{x_D - x_{j-p}}{x_{j-p-1} - x_{j-p}} u^n_{j-p-1} \\ &= \frac{x_j - c\Delta t - x_{j-p-1}}{x_{j-p} - x_{j-p-1}} u^n_{j-p} + \frac{x_j - c\Delta t - x_{j-p}}{x_{j-p-1} - x_{j-p}} u^n_{j-p-1} \\ &= \frac{(p+1)\Delta x - c\Delta t}{\Delta x} u^n_{j-p} + \frac{p\Delta x - c\Delta t}{-\Delta x} u^n_{j-p-1} \\ &= \left(1 - \frac{c\Delta t}{\Delta x} + p\right) u^n_{j-p} + \left(\frac{c\Delta t}{\Delta x} - p\right) u^n_{j-p-1} \end{aligned}$$

Which means the solution is given by

$$\begin{aligned} u^{n+1}_j &= l(t, x_D) \\ &= \left(1 - \frac{c\Delta t}{\Delta x} + p\right) u^n_{j-p} + \left(\frac{c\Delta t}{\Delta x} - p\right) u^n_{j-p-1} \end{aligned}$$

This can be rewritten as

$$u^{n+1}_j = u^n_{j-p} + (p\Delta x - c\Delta t)\frac{u^n_{j-p} - u^n_{j-p-1}}{\Delta x} \tag{13}$$

Introducing the reduced velocity $\tilde{c}$

$$\tilde{c} = c - \frac{p\Delta x}{\Delta t},$$

equation (13) is given by

$$\begin{aligned} u^{n+1}_j &= u^n_{j-p} - \tilde{c}\Delta t \frac{u^n_{j-p} - u^n_{j-p-1}}{\Delta x}, \\ \frac{u^{n+1}_j - u^n_{j-p}}{\Delta t} &= -\tilde{c}\frac{u^n_{j-p} - u^n_{j-p-1}}{\Delta x} \end{aligned}$$

This is the upwind Euler forward methods for $p = 0$, as then the reduced velocity is the same as the original velocity. For the case where $p > 0$, the method is the same, but using shifted nodes and a reduced velocity.

Example 4.2 (Second order)**.** Again, the departure point corresponding to arrival point x_j in the case of uniform velocity is given by

$$x_D = x_j - c\Delta t$$

In the Lagrangian frame, the solution of the advection equation is as follows:

$$u_j^{n+1} = l(t, x_D),$$

where l is the interpolated value of u using Lagrange polynomials. To quadratically interpolate the value of u at this departure point, one needs to know between which two gridpoints x_D is situated. In this case, p is

$$p = \left\lfloor \frac{c\Delta t}{\Delta x} \right\rfloor$$

Then x_D is between x_{j-p-1} and x_{j-p}. For the quadratic interpolation, also include x_{j-p-2}. Linearly interpolating with Lagrange interpolation polynomials gives

$$\begin{aligned}
l(t, x_D) = & \frac{(x_D - x_{j-p-2})(x_D - x_{j-p-1})}{(x_{j-p} - x_{j-p-2})(x_{j-p} - x_{j-p-1})} u_{j-p}^n \\
& + \frac{(x_D - x_{j-p-2})(x_D - x_{j-p})}{(x_{j-p-1} - x_{j-p-2})(x_{j-p-1} - x_{j-p})} u_{j-p-1}^n \\
& + \frac{(x_D - x_{j-p-1})(x_D - x_{j-p})}{(x_{j-p-2} - x_{j-p-1})(x_{j-p-2} - x_{j-p})} u_{j-p-2}^n \\
= & \frac{(x_j - c\Delta t - x_{j-p-2})(x_j - c\Delta t - x_{j-p-1})}{(x_{j-p} - x_{j-p-2})(x_{j-p} - x_{j-p-1})} u_{j-p}^n \\
& + \frac{(x_j - c\Delta t - x_{j-p-2})(x_j - c\Delta t - x_{j-p})}{(x_{j-p-1} - x_{j-p-2})(x_{j-p-1} - x_{j-p})} u_{j-p-1}^n \\
& + \frac{(x_j - c\Delta t - x_{j-p-1})(x_j - c\Delta t - x_{j-p})}{(x_{j-p-2} - x_{j-p-1})(x_{j-p-2} - x_{j-p})} u_{j-p-2}^n \\
= & \frac{((p+2)\Delta x - c\Delta t)((p+1)\Delta x - c\Delta t)}{2\Delta x^2} u_{j-p}^n \\
& + \frac{((p+2)\Delta x - c\Delta t)(p\Delta x - c\Delta t)}{-\Delta x^2} u_{j-p-1}^n \\
& + \frac{((p+1)\Delta x - c\Delta t)(p\Delta x - c\Delta t)}{2\Delta x^2} u_{j-p-2}^n
\end{aligned}$$

$$\begin{aligned}
u_j^{n+1} = & u_{j-p}^n + (p\Delta x - c\Delta t) \frac{3u_{j-p}^n - 4u_{j-p-1}^n + u_{j-p-2}^n}{2\Delta x} \\
& + \frac{(p\Delta x - c\Delta t)^2}{2} \frac{u_{j-p}^n - 2u_{j-p-1}^n + u_{j-p-2}^n}{\Delta x^2}
\end{aligned} \tag{14}$$

And again with the reduced velocity $\tilde{c}$ it reads

$$
\begin{aligned}
u_j^{n+1} = \quad & u_{j-p}^n - \tilde{c}\Delta t \frac{3u_{j-p}^n - 4u_{j-p-1}^n + u_{j-p-2}^n}{2\Delta x} \\
& + \frac{(\tilde{c}\Delta t)^2}{2} \frac{u_{j-p}^n - 2u_{j-p-1}^n + u_{j-p-2}^n}{\Delta x^2}
\end{aligned}
$$

This does not seem the same as a finite difference method for the Eulerian advection equation, but the second term reads like a 2nd order forward approximation to the first derivative. Furthermore, the last term reads like a second order centered approximation to the second derivative.

Both examples are similar in result, with only the order of accuracy increasing and the number of terms. To understand these results, expand the solution to the Lagrangian advection equation with a Taylor expansion:

$$
\begin{aligned}
u_j^{n+1} &= l(t, x_D) \\
&= l(t, x_j - c\Delta t) \\
&= l(t, x_{j-p} - \tilde{c}\Delta t) \\
&= l(t, x_{j-p}) - \tilde{c}\Delta t l'(t, x_{j-p}) + \frac{(-\tilde{c}\Delta t)^2}{2} l''(t, x_{j-p}) + \dots \\
&= \sum_{k=0}^{\infty} \frac{(-\tilde{c}\Delta t)^k}{k!} l^{(k)}(t, x_{j-p})
\end{aligned}
\tag{15}
$$

Here $l^{(k)}$ is the k-th derivative of the Lagrange polynomial used for the interpolation. The degree of l is equal to the order of interpolation used. For linear interpolation, l is also linear. For quadratic interpolation, it too is quadratic. The derivatives of l are zero for k larger than the order of interpolation. Take for example linear interpolation, which is first order, then

$$
l^{(k)} \equiv 0, \quad \text{for } k > 1.
$$

This reasoning also explains why the quadratic interpolation yielded an additional term of higher order in $\tilde{c}\Delta t$ than the linear interpolation.

In the continuous case, the value of u at the departure point can also be expanded around x_{j-p}.

$$
\begin{aligned}
u(t, x_D) &= u(t, x_j - c\Delta t) \\
&= u(t, x_{j-p} - \tilde{c}\Delta t) \\
&= u(t, x_{j-p}) + (-\tilde{c}\Delta t) \left.\frac{\partial u}{\partial x}\right|_{(t, x_{j-p})} + \frac{(-\tilde{c}\Delta t)^2}{2} \left.\frac{\partial^2 u}{\partial x^2}\right|_{(t, x_{j-p})} + \dots \\
&= \sum_{k=0}^{\infty} \frac{(-\tilde{c}\Delta t)^k}{k!} \left.\frac{\partial^{(k)} u}{\partial x^{(k)}}\right|_{(t, x_{j-p})}
\end{aligned}
$$

The Taylor series can be compactly written as

$$\begin{aligned} u(t, x_D) &= e^{(p\Delta x - c\Delta t)\frac{\partial}{\partial x}} u(t, x_j) \\ &= e^{\tilde{c}\Delta t \frac{\partial}{\partial x}} u(t, x_{j-p}) \end{aligned}$$

This notation is exact when u is an analytic function. When m-th degree interpolation is used, the Semi-Lagrangian method can be interpreted as a truncation of the exponential to order m. In particular, each derivative operator $\frac{\partial^{(k)}}{\partial x^{(k)}}$ is approximated by a consistent finite difference operator, i.e. $l^{(k)}$ is consistent with the k-th derivative of u. If m is the order of interpolation used, then the error is at least order $m + 1 - k$.

Assume interpolation of order m, with interpolation nodes $\{x_{j+n_i}\}$ and interpolation values $\{u(t, x_{j+n_i})\}$, with $i = 0, \ldots, m$. In other words a list of $m + 1$ distinct points to use for the m-th degree Lagrange polynomial. The Taylor expansions of $u(t, x_j + n_i)$ about x_j are

$$\begin{cases} u(t, x_{j+n_0}) & = u(t, x_j) + n_0\Delta x \left.\frac{\partial u}{\partial x}\right|_{(t,x_j)} + \cdots + \frac{(n_0\Delta x)^m}{m!} \left.\frac{\partial^{(m)} u}{\partial x^{(m)}}\right|_{(t,x_j)} + \mathcal{O}\left(\Delta x^{m+1}\right) \\ u(t, x_{j+n_1}) & = u(t, x_j) + n_1\Delta x \left.\frac{\partial u}{\partial x}\right|_{(t,x_j)} + \cdots + \frac{(n_1\Delta x)^m}{m!} \left.\frac{\partial^{(m)} u}{\partial x^{(m)}}\right|_{(t,x_j)} + \mathcal{O}\left(\Delta x^{m+1}\right) \\ & \vdots \\ u(t, x_{j+n_m}) & = u(t, x_j) + n_m\Delta x \left.\frac{\partial u}{\partial x}\right|_{(t,x_j)} + \cdots + \frac{(n_m\Delta x)^m}{m!} \left.\frac{\partial^{(m)} u}{\partial x^{(m)}}\right|_{(t,x_j)} + \mathcal{O}\left(\Delta x^{m+1}\right) \end{cases}$$

To approximate the k-th derivative with these points, consider a linear combination of these expansions with coefficients a_i. The following system of equations for all $r = 0, \ldots, m - 1$ needs to be solved

$$n_0^r a_0 + \cdots + n_m^r a_m = \begin{cases} r!, & \text{if} \quad r = k, \\ 0, & \text{if} \quad r \neq k. \end{cases} \tag{16}$$

This is equivalent to

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ n_0 & n_1 & \cdots & n_m \\ n_0^2 & n_1^2 & \cdots & n_m^2 \\ \vdots & \vdots & \vdots & \vdots \\ n_0^{m-1} & n_1^{m-1} & \cdots & n_m^{m-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = k!\vec{e}_k \tag{17}$$

The matrix on the left is a Vandermonde matrix, known to be nonsingular if the nodes n_i are distinct (Klinger, 1967). Furthermore, $\vec{e}_k$ is the k-th standard basis vector. The equations (16) imply.

$$a_0 u(t, x_{j+n_0}) + \cdots + a_m u(t, x_{j+n_m}) = \Delta x^k \left.\frac{\partial^{(k)} u}{\partial x^{(k)}}\right|_{(t,x_j)} + \mathcal{O}\left(\Delta x^{m+1}\right)$$

So, the solution of equation (17) are the coefficients for the finite difference equation that approximates the k-th derivative of u:

$$\frac{a_0 u(t, x_{j+n_0}) + \cdots + a_m u(t, x_{j+n_m})}{\Delta x^k} = \left.\frac{\partial^{(k)} u}{\partial x^{(k)}}\right|_{(t,x_j)} + \mathcal{O}\left(\Delta x^{m+1-k}\right)$$

The error could also be of higher order. More specifically, if q is the smallest integer greater than m satisfying $n_0^q a_0 + \cdots + n_m^q a_m \neq 0$, then the error is of order Δx^{q-k}. Here $q - k \geq m + 1 - k$.

Example 4.3. Take the points x_{j-1} and x_{j+1}. Let us use the value of u at these points to calculate $u(t, x_j)$ and the first derivative at this same point. This means $n_0 = -1$ and $n_1 = 1$. The equations to solve are

$$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} a_0' \\ a_1' \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

These two equations yield the following results:

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}, \qquad \begin{bmatrix} a_0' \\ a_1' \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ \frac{1}{2} \end{bmatrix}.$$

Furthermore, $a_0^2 + a_1^2 \neq 0$, $a_0'^2 + a_1'^2 = 0$ and $a_0'^3 + a_1'^3 \neq 0$. This means the approximations have error of Δx^{2-0} and Δx^{3-1} respectively.

$$\begin{aligned} \frac{u(t, x_{j-1}) + u(t, x_{j+1})}{2} &= u(t, x_j) + \mathcal{O}\left(\Delta x^2\right) \\ \frac{-u(t, x_{j-1}) + u(t, x_{j+1})}{2} &= \left.\frac{\partial u}{\partial x}\right|_{(t,x_j)} + \mathcal{O}\left(\Delta x^2\right) \end{aligned}$$

The fact that the first derivative is calculated with a centered method, ensures this higher order of accuracy.

Example 4.4. Let's try to find a finite difference method for the third derivative in (t, x_j) using $\{x_{j-4}, x_{j-1}, x_j, x_{j+2}, x_{j+3}\}$, $(m = 4)$.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & 0 & 2 & 3 \\ 16 & 1 & 0 & 4 & 9 \\ -64 & -1 & 0 & 8 & 27 \\ 256 & 1 & 0 & 16 & 81 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 6 \\ 0 \end{bmatrix}.$$

These equations yield the following result:

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 \end{bmatrix} = \begin{bmatrix} -\frac{1}{21} & \frac{1}{6} & 0 & -\frac{1}{3} & \frac{3}{14} \end{bmatrix}.$$

Furthermore, $-1024a_0 - a_1 + 32a_3 + 243a_4 \neq 0$. This means the approximation has an error of Δx^{4+1-3}.

$$\frac{-2u(t,x_{j-4})+7u(t,x_{j-1})-14u(t,x_{j+2})+9u(t,x_{j+3})}{42\Delta x^3}=\left.\frac{\partial^3 u}{\partial x^3}\right|_{(t,x_j)}+\mathcal{O}\left(\Delta x^2\right)$$

This is the only finite difference formula possible with these points and has second order accuracy.

Equations (13) and (14) can be understood as follows. Equation (13) uses two points, is therefore linear, so had a Taylor expansion up to the first derivative. This first derivative is approximated by the finite difference equation that can be made with the two points. Equation (14) utilises three points, is therefore quadratic, and the last term has an approximation for the second derivative. In conclusion, the Semi-Lagrangian method is consistent with the Taylor approximation up to one order higher than the order of interpolation used.

4.2 Stability

The Semi-Lagrangian method is shown to be equal to some finite difference method that is consistent with a Taylor expansion around the departure point. The next question is if this method is stable. In the sense of Von Neumann numerical stability analysis and stability of the resulting map.

Numerical stability
Recall that in the Eulerian frame, using first order upwind Euler forward yielded

$$u_j^{n+1}=\left(1-\frac{c\Delta t}{\Delta x}\right)u_j^n+\frac{c\Delta t}{\Delta x}u_{j-1}^n \tag{18}$$

And recall the formula that was the result of the Semi-Lagrangian method with linear interpolation:

$$u_j^{n+1}=\left(1-\frac{c\Delta t}{\Delta x}+p\right)u_{j-p}^n+\left(\frac{c\Delta t}{\Delta x}-p\right)u_{j-p-1}^n,\qquad p=\left\lfloor\frac{c\Delta t}{\Delta x}\right\rfloor \tag{19}$$

For $p=0$ this is the same as equation (18), indicating the similarity in both methods. Again introduce the Fourier mode of the error:

$$\epsilon(x,t)=E(t)e^{ik_m x}$$

Plugging this general mode into equation (19) yields

$$
\begin{aligned}
E((n+1)\Delta t)e^{ik_m j\Delta x} &= \left(1 - \frac{c\Delta t}{\Delta x} + p\right) E(n\Delta t)e^{ik_m(j-p)\Delta x} \\
&\quad + \left(\frac{c\Delta t}{\Delta x} - p\right) E(n\Delta t)e^{ik_m(j-p-1)\Delta x} \\
E((n+1)\Delta t) &= \left(1 - \frac{c\Delta t}{\Delta x} + p\right) E(n\Delta t)e^{-ik_m p\Delta x} \\
&\quad + \left(\frac{c\Delta t}{\Delta x} - p\right) E(n\Delta t)e^{-ik_m(p+1)\Delta x} \\
\frac{E((n+1)\Delta t)}{E(n\Delta t)} &= \left(1 - \frac{c\Delta t}{\Delta x} + p\right) e^{-ik_m p\Delta x} \\
&\quad + \left(\frac{c\Delta t}{\Delta x} - p\right) e^{-ik_m(p+1)\Delta x} \\
&= \left(1 - \frac{c\Delta t}{\Delta x} + p\right) \cos\left(k_m p\Delta x\right) + \left(\frac{c\Delta t}{\Delta x} - p\right) \cos\left(k_m(p+1)\Delta x\right) \\
&\quad - i\left(1 - \frac{c\Delta t}{\Delta x} + p\right) \sin\left(k_m p\Delta x\right) - i\left(\frac{c\Delta t}{\Delta x} - p\right) \sin\left(k_m(p+1)\Delta x\right)
\end{aligned}
$$

The norm of this ratio is given by

$$
\begin{aligned}
\left|\frac{E((n+1)\Delta t)}{E(n\Delta t)}\right|^2 &= \left(1 - \frac{c\Delta t}{\Delta x} + p\right)^2 + \left(\frac{c\Delta t}{\Delta x} - p\right)^2 \\
&\quad + 2\left(1 - \frac{c\Delta t}{\Delta x} + p\right)\left(\frac{c\Delta t}{\Delta x} - p\right) \cos\left(k_m \Delta x\right)
\end{aligned}
$$

Analogous to the Eulerian case, for this norm to be less or equal to 1 yields

$$
p \leq \frac{c\Delta t}{\Delta x} \leq p + 1.
$$

For $p = 0$, this is the same stability region as before in the Eulerian frame. Even more similarity can be seen when the *reduced velocity* $\tilde{c}$, given by

$$
\tilde{c} = c - \frac{p\Delta x}{\Delta t},
$$

is plugged into equation (4.1). This yields

$$
u_j^{n+1} = \left(1 - \frac{\tilde{c}\Delta t}{\Delta x}\right) u_{j-p}^n + \left(\frac{\tilde{c}\Delta t}{\Delta x}\right) u_{j-p-1}^n.
$$

Here now

$$
0 \leq \frac{\tilde{c}\Delta t}{\Delta x} \leq 1,
$$

so the reduced velocity also satisfies the stability criterion from before. The only difference is that the two values used to calculate u_j^{n+1} have shifted to u_{j-p}^n and u_{j-p-1}^n instead of u_j^n and u_{j-1}^n. This corresponds to shifting the domain $p\Delta x$ to the left, such that the reduced velocity satisfies the stability condition of the case without the shift.

Stability of maps
Rewriting equation (19) as a vector matrix equation gives

$$\vec{u}^{n+1} = AS^p\vec{u}^n, \qquad A = I - \frac{\tilde{c}\Delta t}{\Delta x}\begin{bmatrix} 1 & 0 & \cdots & 0 & -1 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & & -1 & 1 & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}$$

Here S is the left shift matrix given by

$$S = \begin{bmatrix} 0 & \cdots & 0 & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$$

Remark. Both A and S are circulant matrices.

Remark. The corresponding vectors of circulant matrices A and S are given by

$$\vec{b}_A = \begin{bmatrix} 1 - \frac{c\Delta t}{\Delta x} + p \\ 0 \\ \vdots \\ 0 \\ \frac{c\Delta t}{\Delta x} - p \end{bmatrix}, \qquad \vec{b}_S = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

The following lemma is well-known:

Lemma 4.5. *Let A and B be circulant matrices of equal dimension. The eigenvalues of the matrix AB are given by the product of eigenvalues from A and B.*

Proof. Let A and B be circulant matrices, with eigenvalues α_k, β_k respectively. The eigenvectors v_k of circulant matrices are independent of the matrix elements, so they are the same for every circulant matrix. This yields $Av_k = \alpha_k v_k$ and $Bv_k = \beta_k v_k$. The eigenvalues of the matrix AB, the result of the matrix multiplication, are given by

$$ABv_k = A\beta_k v_k = \beta_k A v_k = \beta_k \alpha_k v_k$$

So the eigenvalues are equal to $\beta_k\alpha_k$, the product of eigenvalues from A and B. □

Lemma 4.6. *Shifting all the elements of a circulant matrix with the shift matrix S, rotates the eigenvalues in the complex plane.*

Proof. Let A be a circulant matrix with eigenvalues α_k corresponding to eigenvectors v_k. Let S be a shift matrix. This entails that S is a circulant matrix with only one non-zero element of $\vec{b}$, $b_m = 1$. The eigenvalues μ_k of S corresponding to eigenvectors v_k are given by

$$\mu_k = b_m e^{\frac{2\pi i}{d} mk} = e^{i\phi_k}, \qquad \phi_k = \frac{2\pi mk}{d}, \qquad k = 0, 1, \ldots, m-1.$$

Lemma 4.5 says that the eigenvalues of AS are equal to $\alpha_k \mu_k = \alpha_k e^{i\phi_k}$. This shows that the eigenvalues of A are rotated in the complex plane when multiplied by S. Specifically

$$|\alpha_k \mu_k| = |\alpha_k e^{i\phi_k}| = |\alpha_k|, \qquad arg(\alpha_k \mu_k) = arg(\alpha_k e^{i\phi_k}) = arg(\alpha_k) + \phi_k$$

□

Because the shift matrix only rotates the eigenvalues, it does not change their absolute value and in turn will not change the spectral radius. This means

$$\rho\left(AS^p\right) = \rho\left(A\right)$$

The stability of the map is determined by the spectral radius of AS^p. As this is the same as the spectral radius of A, the stability region is the same as in the Eulerian case.

Non-uniform velocity

For non-uniform velocity $\vec{v}$, the reduced velocity is not the same for every arrival point. The shift p is also dependent on the velocity, so it can vary. Now, both the matrix A is not circulant anymore, as the shift needed to make the method consistent cannot be written as S^p. This loss of the circulant properties, means the map can now become unstable for interpolation.

For the map to be stable, the spectral radius should be less than or equal to unity. This spectral radius is bound by every matrix norm induced by a vector norm. Polynomial interpolation always has the property that

$$\sum_{i=0}^{m} a_i = 1$$

In the case of linear interpolation, the interpolation coefficients also sum to 1 in absolute value:

$$\sum_{i=0}^{m} |a_i| = 1$$

The ∞-norm of a matrix is given by

$$\max_{0 \leq j \leq m} \sum_{i=0}^{m} |a_{ij}|.$$

In other words, this norm is the maximum value of all absolute row sums. For linear interpolation we get

$$\rho(\tilde{A}) \leq ||\tilde{A}||_\infty = 1,$$

meaning the map is always stable. The reason this holds, is because of the maximum principle. If there exist u_{min} and u_{max} such that all the values $\{u_j^n\}$ satisfy

$$u_{min} \leq u_j^n \leq u_{max}, \qquad \forall j = 0, \ldots, d-1.$$

Then all the linear interpolated values $l(x)$ satisfy

$$u_{min} \leq l(x) \leq u_{max}, \qquad \forall x \in \mathbb{D}.$$

Consequently

$$||\vec{u}^{n+1}||_\infty \leq ||\vec{u}^n||_\infty,$$

which is the same as the ∞-norm of the matrix being less than or equal to unity. For higher order interpolation, this maximum principle does not hold. Examples can easily be constructed where the spectral radius is bigger than 1 for a non-uniform velocity and higher order interpolation.

4.3 Convergence

For convergence, again both consistency and stability is needed. The consistency condition in this case is that the Semi-Lagrangian method used is consistent with a truncated Taylor approximation using the reduced velocity $\tilde{c}$. For Semi-Lagrangian methods this means the reduced departure point is given by $\tilde{x}_D = x_j - \tilde{c}\Delta t$. So the departure point is shifted back to the adjacent interval to the arrival point in steps of Δx.

The resulting departure point will be interpolated, where this interpolation will equal to some finite difference method of a truncated Taylor series. This finite difference method has a certain stability region, with the reduced departure point in it. With the use of the shift matrix, the method is shifted to the original departure point. The stability region remains unchanged, only shifted along to the departure point. But the stability of the method is the same, whether it is

applied to the reduced departure point or shifted to the original departure point.

As the method is used to find a solution with the original velocity, the shift is necessary for consistency. First it was applied to the reduced departure point, so the resulting finite difference method is only consistent with the advection equation with reduced velocity. The shift makes the method consistent with the original advection equation.

Using a Semi-Lagrangian method to solve the advection equation with uniform constant velocity can be understood as a finite difference technique:

1. Calculate the reduced velocity via

$$\tilde{c} = c - \frac{p\Delta x}{\Delta t}, \qquad p = \left\lfloor \frac{c\Delta t}{\Delta x} \right\rfloor .$$

2. Interpret the solution to the advection equation as an interpolation around the departure point

$$\begin{aligned} u(t^{n+1}, x_j) &= u(t^n, x_D) \\ &= u(t^n, x_j - c\Delta t) \\ &= u(t^n, x_{j-p} + p\Delta x - c\Delta t) \\ &= u(t^n, x_{j-p} - \tilde{c}\Delta t) \\ &= e^{-\tilde{c}\Delta t \frac{\partial}{\partial x}} u(t^n, x_{j-p}) \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} \left(-\tilde{c}\Delta t \frac{\partial}{\partial x} \right)^k u(t^n, x_{j-p}) \\ &= u(t^n, x_{j-p}) - \tilde{c}\Delta t \left. \frac{\partial u}{\partial x} \right|_{(t^n, x_{j-p})} + \frac{(\tilde{c}\Delta t)^2}{2} \left. \frac{\partial^2 u}{\partial x^2} \right|_{(t^n, x_{j-p})} - \dots . \end{aligned}$$

3. Choose the number of interpolation nodes $\{x_{j-p+n_i}\}$ and corresponding interpolation values $\{u^n_{j-p+n_i}\}$ for $i = 0, \dots, m$, based on the order of accuracy m required. This will determine the truncation of the Taylor approximation up to $k = m$.

4. Use these nodes for finite difference methods consistent with the terms in the exponential expansion. To find the coefficients $\{a_i\}$, $\{b_i\}$, etc., each time a system of equation with a Vandermonde matrix has to be solved. The resulting finite difference methods will be given by

$$
\begin{aligned}
u(t^n, x_{j-p}) &= a_0 u^n_{j-p+n_0} + \cdots + a_m u^n_{j-p+n_m} + \mathcal{O}\left(\Delta x^{m+1}\right) \\
\left.\frac{\partial u}{\partial x}\right|_{(t^n, x_{j-p})} &= \frac{b_0 u^n_{j-p+n_0} + \cdots + b_m u^n_{j-p+n_m}}{\Delta x} + \mathcal{O}\left(\Delta x^m\right) \\
&\vdots \\
\left.\frac{\partial^{(m)} u}{\partial x^{(m)}}\right|_{(t^n, x_{j-p})} &= \frac{c_0 u^n_{j-p+n_0} + \cdots + c_m u^n_{j-p+n_m}}{\Delta x^m} + \mathcal{O}\left(\Delta x\right)
\end{aligned}
$$

This will result in the solution

$$
\begin{aligned}
u^{n+1}_j = {} & a_0 u^n_{j-p+n_0} + \cdots + a_m u^n_{j-p+n_m} \\
& - \frac{\tilde{c}\Delta t}{\Delta x}\left(b_0 u^n_{j-p+n_0} + \cdots + b_m u^n_{j-p+n_m}\right) \\
& + \ldots \\
& + \frac{1}{m!}\left(\frac{-\tilde{c}\Delta t}{\Delta x}\right)^m \left(c_0 u^n_{j-p+n_0} + \cdots + c_m u^n_{j-p+n_m}\right)
\end{aligned}
$$

It can be written as a map

$$
\begin{aligned}
\vec{u}^{n+1} &= D_0 S^p \vec{u}^n - \frac{\tilde{c}\Delta t}{\Delta x} D_1 S^p \vec{u}^n + \cdots + \frac{1}{m!}\left(\frac{-\tilde{c}\Delta t}{\Delta x}\right)^m D_m S^p \vec{u}^n \\
&= \sum_{k=0}^{m} \frac{1}{k!}\left(\frac{-\tilde{c}\Delta t}{\Delta x}\right)^k D_k S^p \vec{u}^n \\
&= \tilde{A} S^p \vec{u}^n
\end{aligned}
$$

5. These finite differences are then shifted p grid points to be consistent with the advection equation with the original velocity. So in total the advection equation, solved with the Semi-Lagrangian method with m-th order interpolation, can be interpreted as the following finite difference method.

$$
\begin{aligned}
\frac{du}{dt} &= 0 \\
&\Updownarrow \\
u(t^{n+1}, x_j) &= u(t^n, x_D) \\
&\Updownarrow \\
u(t^{n+1}, x_j) &= \sum_{k=0}^{m} \frac{1}{k!} \left(-\tilde{c}\Delta t \frac{\partial}{\partial x} \right)^k u(t^n, x_{j-p}) \\
&\Updownarrow \\
u_j^{n+1} &= a_0 u_{j-p+n_0}^n + \cdots + a_m u_{j-p+n_m}^n \\
&\quad + \frac{b_0 u_{j-p+n_0}^n + \cdots + b_m u_{j-p+n_m}^n}{\Delta x} \\
&\quad + \dots \\
&\quad + \frac{c_0 u_{j-p+n_0}^n + \cdots + c_m u_{j-p+n_m}^n}{\Delta x^m} \\
&\Updownarrow \\
\vec{u}^{n+1} &= \tilde{A} S^p \vec{u}^n
\end{aligned}
$$

The knowledge about consistency and stability in the Eulerian frame can be applied to the Semi-Lagrangian method. Therefore, in the case of uniform constant velocity, the Semi-Lagrangian method is both consistent and stable when seen as a finite difference method. This means the numerical solution of the Semi-Lagrangian method will converge to the analytical solution of the advection equation as $\Delta t, \Delta x \to 0$.

In the case of the upwind Euler forward finite difference scheme, the CFL-condition was given by

$$0 \leq \frac{c\Delta t}{\Delta x} \leq 1.$$

Increasing the time step size, increasing the velocity or reducing the grid spacing could all result in this limit being exceeded. That would result in the loss of convergence due to numerical instability. The first order Semi-Lagrangian method can be understood as calculating the reduced velocity, that satisfies the CFL-condition again, and then shifting the numerical domain of dependence so that it covers the analytical domain of dependence again. This is visualised in figure 9. For values of $\frac{c\Delta t}{\Delta x}$ larger than unity, the original CFL-condition is exceeded. With the reduced velocity, p is now calculated to be 1, which means the numerical domain of dependence should be shifted to cover the analytical domain of dependence again. This new numerical domain of dependence is seen as the shaded blue area. The CFL-condition is now again satisfied, for otherwise unstable values of $\frac{c\Delta t}{\Delta x}$.

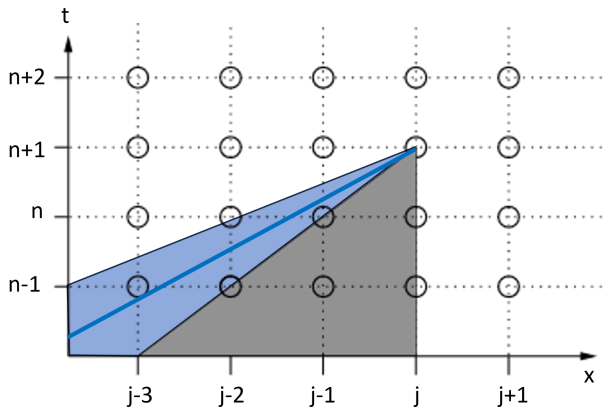


Figure 9: The shifted CFL-condition visualised for the Semi-Lagrangian method with linear interpolation. The grey area is the numerical domain of dependence for $p = 0$, which means the method is the same as the upwind Euler forward method. The blue region is the numerical domain of dependence for $p = 1$, which now covers the blue line indicating the analytical domain of dependence.

5 Discussion

Rewriting the Semi-Lagrangian method as an finite difference method helped us understand why they allow for larger time steps while still maintaining stability. For uniform and constant velocity, determining the departure point is exact and the polynomial interpolation used can be understood as a truncation of the Taylor series up to the degree of the polynomial used for the interpolation. These terms in the Taylor series are equal to some finite difference approximation of the corresponding derivative. The use of circulant shift matrices, allows for shifting of the stability region to always include the departure point, while being consistent with a finite difference scheme around a reduced departure point.

Non-uniform, non-constant velocity
The connection between Eulerian finite difference methods and Semi-Lagrangian methods can be used to study their stability. For a non-uniform velocity, linear interpolation can be shown to be stable as long as the departure point is between the interpolation nodes. When using an interpolation polynomial of higher degree, the matrix used for the integration map does not necessarily have spectral radius smaller or equal to unity. This does not mean that using higher order interpolation in for example atmospheric conditions can quickly yield unstable numerical integration. The value of the interpolant can exceed all interpolation values, but this will not occur every time step at the same location. The main reason for this are the advection of the velocity itself and the viscosity of the transportation fluid. The velocity will not remain constant in atmospheric conditions, therefore changing the integration map every time step. Large velocity gradients are counteracted by the viscosity of the fluid, diffusing these gradients.

Future research could mainly focus on what the effect of the changing velocity profile over time is on the stability of the integration map. Single time steps could be done with a matrix whose spectral radius is bigger than unity. The velocity profile has changed for the next time steps, dampening the instability created by the unstable time step. Incorporating an equation for the change in velocity profile, such as the viscous Burgers' equation, could be combined with the Semi-Lagrangian advection, to show stability over more integration steps. The analysis of this thesis is also only applied to one-dimensional advection, but weather prediction is a three-dimensional problem. The connection of these higher dimensional problems could maybe also be connected to finite difference methods.

Analysis of the problem of HARMONIE
We used the theoretical analysis from the thesis to study some recurring failures in the HARMONIE weather prediction code used by KNMI. Analysing the errors given by the HARMONIE model when the model crashes, leads us to determine where the problem originates. The log files show a lot of warnings about the Semi-Lagrangian departure point being underground, below the low-

est model layer. These are warnings, because the model has code that shifts these departure points back above ground. This same log then shows an error message about an invalid floating point operation. These are typically square roots of negative numbers, or divisions by zero. In this code, analysis of the errors origin shows us that the problem is a square root of a negative number. In calculations regarding the condensation of water in the air, the square root of the model layer thickness is used. This thickness is calculated using the difference between the vertical coordinate of a grid point and the one below. The layer thickness can become negative if the height of a gridpoint is higher than the one above or lower than the one below. The problem seems to be in the determination of the departure point. The HARMONIE model uses the SETTLS fixed point iteration with 2 iterations. Increasing this number of iterations or changing the fixed point iteration to a predictor-corrector method both sometimes help to fix the problem. This is another indication that the departure point problem is the thing that crashes the model run.

This paper proved that the departure point always exists. Model runs of HARMONIE are run with time steps of 75 seconds, while typical vertical velocity gradients allow time steps in the order of 10^4. For the used time step size, the departure point is always unique and the fixed point iteration converges to the departure point. Moreover, this unique departure point is monotone, so two departure points corresponding to two arrival points will never change order. This leaves only one possible explanation for the error. If the fixed point iteration converges very slowly, the limited number of iterations can result in the observed switch. Some converging fixed point iterations can jump around the fixed point, converging very slowly. Two neighbouring arrival points, both converging slowly to their departure point and jumping around it during the iterations, can yield a change in order at the truncation of the iterations. The higher the iteration count, the smaller truncation error to the fixed point. The HARMONIE operational model only uses 2 iterations, so this effect can be significant enough. The rate of convergence depends on the ratio between the velocity gradient and the time step. The bigger the gradient, the slower the convergence is and the larger the timestep, the slower the convergence is.

The model run fails in an area with large changes in surface elevation. The wind following the surface will be changing from horizontal to vertical, meaning a large vertical velocity gradient. This causes the both the warnings of trajectories going underground, as the error resulting in termination of the run. As the vertical velocity gradient is large, the convergence will be slow, increasing the chance for two point to switch in vertical order.

6 Conclusion

Rewriting the Semi-Lagrangian method as a finite difference method helped us understand its advantages. The numerical domain of dependence can be shifted to allow arbitrarily large time steps. The stability is no longer lost for larger time steps. The only limit on the time step comes from the determination of the departure point. Using fixed point iteration poses a limitation on the time step, as the uniqueness and convergence depend on the velocity gradient and the time step used. For typical atmospheric conditions, this limit is not a problem. The problem of the HARMONIE model seems to originate in the truncation of the fixed point iteration. Regions with large vertical velocity gradients can slow the convergence of the fixed point iteration and cause a crossing of characteristic lines when it is truncated. Vertical crossing of characteristics create negative layer heights in the model, causing a floating point exception.

The analysis of this thesis is only possible for one dimensional advection with uniform and constant velocity. Future research could try to find a similar connection to finite difference methods for higher dimensional advection or problems with a non-uniform and non-constant velocity. For the non-uniform and non-constant velocity, the transport of the velocity itself and viscosity of the transportation fluid should be taken into account.

References

Bermejo, R., & Staniforth, A. (1992). The conversion of semi-lagrangian advection schemes to quasi-monotone schemes. *Monthly Weather Review*, *120*(11), 2622–2632.

Charney, J. G., Fjörtoft, R., & Neumann, J. v. (1950). Numerical integration of the barotropic vorticity equation. *Tellus*, *2*(4), 237–254.

Côté, J., & Staniforth, A. (1988). A two-time-level semi-lagrangian semi-implicit scheme for spectral models. *Monthly weather review*, *116*(10), 2003–2012.

Courant, R., Friedrichs, K., & Lewy, H. (1928). Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische annalen*, *100*(1), 32–74.

Davis, P. J. (1979). *Circulant matrices* (Vol. 2). Wiley New York.

Fjørtoft, R. (1955). On the use of space-smoothing in physical weather forecasting. *Tellus*, *7*(4), 462–480.

Fletchet, C. A. (1991). Computational techniques for fluid dynamics; vol 1.

Herman, R. L. (2015). Introduction to partial differential equations. *North Carolina, NC, USA: RL Herman*.

Klinger, A. (1967). The vandermonde matrix. *The American Mathematical Monthly*, *74*(5), 571–574.

Krishnamurti, T. N. (1962). Numerical integration of primitive equations by a quasi-lagrangian advective scheme. *Journal of Applied Meteorology (1962-1982)*, 508–521.

Lax, P. D., & Richtmyer, R. D. (1956). Survey of the stability of linear finite difference equations. *Communications on pure and applied mathematics*, *9*(2), 267–293.

Leith, C. E. (1965). Numerical simulation of the earth's atmosphere, methods. *Comput. Phys.*, *4*, 1–28.

Miranda, C. (1940). Un'osservazione su un teorema di brouwer. *Bollettino dell'Unione Matematica Italiana*, *3*(2), 5–7.

Richardson, L. F. (1922). *Weather prediction by numerical process*. University Press.

Robert, A. (1981). A stable numerical integration scheme for the primitive meteorological equations. *Atmosphere-Ocean*, *19*(1), 35–46.

Robert, A. (1982). A semi-lagrangian and semi-implicit numerical integration scheme for the primitive meteorological equations. *Journal of the Meteorological Society of Japan. Ser. II*, *60*(1), 319–325.

Sawyer, J. S. (1963). A semi-lagrangian method of solving the vorticity advection equation. *Tellus*, *15*(4), 336–342.

Simmons, A. (1991). Development of a high resolution, semi-lagrangian version of the ecmwf forecast model. *Numerical Methods in Atmospheric Models*, *2*, 281–324.

Staniforth, A., & Côté, J. (1991). Semi-lagrangian integration schemes for atmospheric models—a review. *Monthly weather review*, *119*(9), 2206–2223.

Temperton, C., & Staniforth, A. (1987). An efficient two-time-level semi-lagrangian semi-implicit integration scheme. *Quarterly Journal of the Royal Meteorological Society*, *113*(477), 1025–1039.

Welander, P. (1955). Studies on the general development of motion in a two-dimensional, ideal fluid. *Tellus*, *7*(2), 141–156.

Wiin-Nielsen, A. (1959). On the application of trajectory methods in numerical forecasting. *Tellus*, *11*(2), 180–196.

Willcox, K., & Wang, Q. (n.d.). 2.6 upwinding and the cfl condition. https://mitocw.ups.edu.ec/courses/aeronautics-and-astronautics/16-90-computational-methods-in-aerospace-engineering-spring-2014/numerical-methods-for-partial-differential-equations/upwinding-and-the-cfl-condition/1690r-the-cfl-condition/