Chapter 6. Semi-Lagrangian Methods

References:

Durran Chapter 6.
Review article by Staniforth and Cote (1991) MWR, 119, 2206-2223.

6.1. Introduction

Semi-Lagrangian (S-L for short) methods, also called quasi-Lagrangian, have been around since the early 1960's. T.N. Krishnamurti is probably the first to use S-L method. The S-L scheme becomes really popular following Andre Robert's work in the early 80's (Robert 1982) when he combined the use of S-L with semi-implicit method (SLSI) which allows for large time step size because both schemes are unconditionally stable. The SL scheme deals with the advection part, and the SI scheme deals with faster wave propagation.

Two principal reasons for using SL methods in favor of more traditional Eulerian approaches:

Major: To overcome the CFL constraint which is based on <u>stability</u>, not <u>accuracy</u> considerations. That is, from on accuracy point of view, we could live with a much larger Δt than which is required by the linear stability condition because <u>time truncation error</u> is often smaller than <u>spatial truncation error.</u>

Minor: Because SL methods solve the equations in Lagrangian form, i.e., du/dt=0 rather than the advection form, i.e., $\partial u/\partial t + \vec{V}\cdot\nabla u = 0$, there is no instability due to aliasing.

In the case of pure advection, stability depends upon the transport velocity. Since advection term does not appear in SL methods, there is <u>no time step constraint</u> for SL schemes. (However, Δt must be small enough to keep the total truncation error reasonably small. Just as with implicit schemes that are unconditionally stable, we don't want to use too large a Δt due to temporal truncation error – remember that home work problem for heat transfer?).

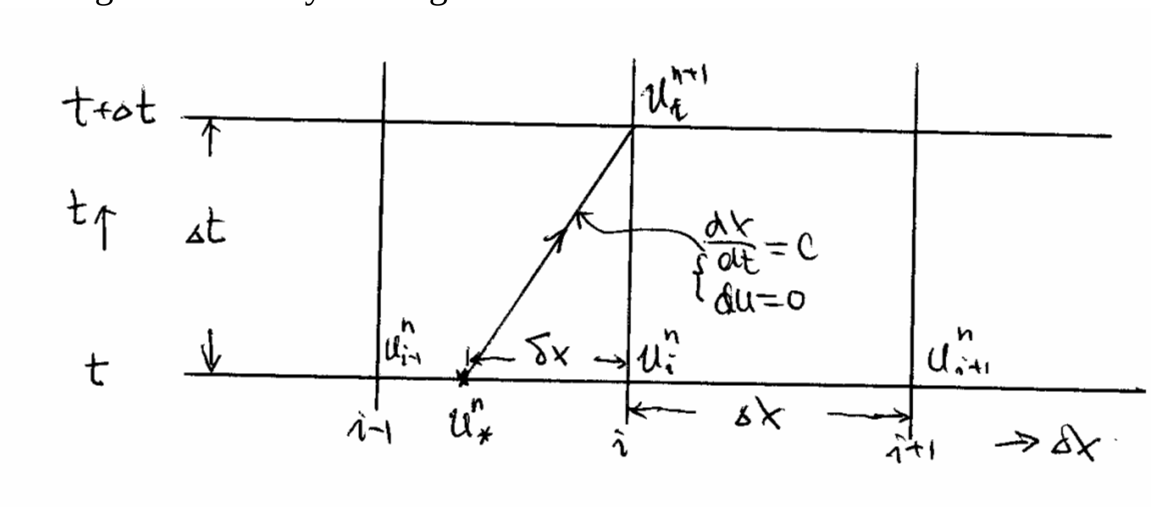
Of course other modes can be present in the problem. For example, internal gravity waves, which can impose more severe limitation on Δt unless they are treated implicitly. Hence the SL-SI methods.

Before looking at SL methods, recall the possible coordinate frameworks discussed earlier in the semester:

1. Eulerian – here, we use a fixed set of points. In the case of spatially staggered grids. Unless you use grid refinement, you always work with a fixed number of points.

2. Lagrangian – the other extreme. In this case, the mesh actually follows the fluid and can become distorted, necessitating a remaping operation from time to time.

3. Semi-Lagrangian – Here we seek to combine the two methods by moving data relative to a fixed (in space) grid. It is really nothing more than the method of characteristics:



We "sit" at point i, and realizing that $u_i^{n+1} = u_*^n$, or the value of u traced back along the characteristic path. This is because we have the compatibility equation

$$du = 0 \text{ along } dx/dt = c \text{ for a pure advection problem } \partial u/\partial t + c\partial u/\partial x = 0 .$$

Recall that by using quadratic interpolation to obtain value u_* we obtained Lax-Wendroff scheme that is stable when

$$c\, Dt \leq Dx.$$

In other words, the departure point could be no further than Dx away from point i, i.e., the scheme is stable provided than u_* was determined by interpolation not extrapolation. This means that the departure point must be surrounded by points used to determine its value.

6.2. Stability of S-L schemes

Why S-L schemes are unconditionally stable?

Consider a simple 1-D transport problem in our previous space-time diagram. If u is known at all previous times, how do we find $u(t+Dt)$? From our discussions earlier in the semester, we know that

$$u(x, t+Dt) = u(x-ct, t).$$

Consider now that arrival point $x_I = I\, Dx$ at $t + Dt = (n+1)Dt$. Then, the departure point is

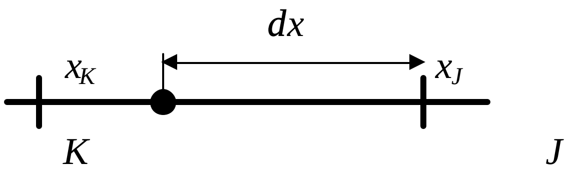
$$x_* = i\, Dx - c\, Dt.$$

Thus,

$$u(iDx,\ (n+1)Dt\ ) = u(\ x_*,\ nDt)$$

where, again, x_* needs not be a grid point.

Consider a linear Lagrange interpolation polynomial for obtaining u_* - this is simple yet non-trivial and is sufficient for our analysis. For any 2 grid points, *J* and *K*, with *J* > *K*, we can write without loss of generality



$$u(\ x_*, n\Delta t) = \frac{x_* - x_K}{x_J - x_K} u_J^n + \frac{x_J - x_*}{x_J - x_K} u_K^n\ .$$

Now if we apply a von Neumann stability analysis to this equation, using $x_* = i\ Dx - c\ Dt$ and $u(x_*, nDt) = u(iDx, (n+1)Dt\ )$, we obtain (verify for yourselves):

$$|\ l\ |^2 = 1 + \left[-1 + \left\{ -1 + \frac{2(I - s\ - J)}{K - J} \right\}^2 \right] \sin^2 \left[\frac{k\Delta x}{2}(J - K) \right]$$

where $s\ = c\Delta t / \Delta x$ is the Courant number. We want $|\lambda|^2 \le 1$ for stability, from which it follows that

$$0 \le \frac{2(I - s - J)}{K - J} \le 2$$

or equivalently, $K \le (i - \sigma) \le J$.

Multiplying through by Δx gives

$$\boxed{x_K \le x_* \le x_J}$$

for stability. Thus, stability is guaranteed for any location of the departure point in the mesh provided that *interpolation* is used to obtain u_* *instead of extrapolation*!

Note, with, for example, upstream forward scheme, we have decided to use u_{i-1} and $u_{i,}$ therefore to meet the 'interpolation' requirement, *cDt* has to be no greater than *Dx* – a consistent result.

Discussion:

- SL methods are no more accurate than their Eulerian counterparts.
- Since fewer time steps are taken, the time truncation error is actually smaller.
- The biggest drawback with SL methods is the difficulty in enforcing conservation.
- For constant flow, the time step size can be arbitrarily large so we can complete the integration of a pure advection problem in one step. The only error comes from the spatial interpolation.

6.3. Implementation of S-L schemes

Let's now look at the actual ways of applying SL method.

Consider a 3 time-level scheme applied to the equation

$$\frac{dF}{dt} = 0 .$$

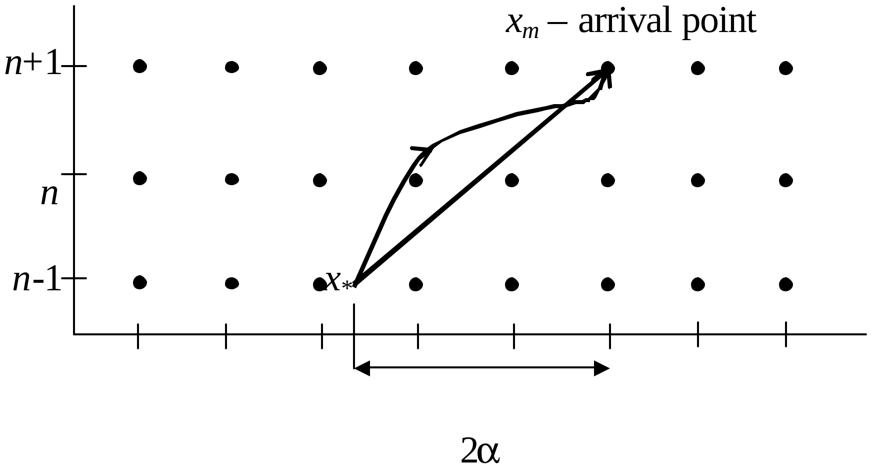
Let's write the scheme as

$$F[\ x_m(t+Dt),\ t+Dt\ ] - F[\ x_*(t\ -Dt),\ t\ -\ Dt\ ] = 0$$

where x_m = mesh point,

x_* = departure point.

Now, let a = distance a particle travels in a single time step. Clearly, it is a function of the local velocity and the Dt. Let's look at an x-t diagram



Let's assume that the trajectory is a straight line.

The major problem in determining the trajectory is that

$$\frac{dx}{dt} = u(x, t)$$

is implicit - x depends on *u* and *u* depends on *x*.

From our diagram, we have

$$F[\, x_m,\, t+Dt] - F[x_m - 2a,\, t\text{-}Dt\ ] = 0,$$

so, we need both a and *F* at the departure point (a determines the departure point).

Question: How do we find a?

$a = f(u)$
$u = dx/dt$
$x = g(a)$ → implicit problem!

When *u* = constant everywhere (uniform flow), SL methods are essentially exact for obvious reasons.

To find a, we integrate the trajectory equation over time interval [*t-Dt*, *t+Dt*] using the mid-point rule:

$$\int_{x-\Delta x}^{x+\Delta x} f(x)dx = 2\Delta x f(x) + O(\Delta x^3)$$

$$\int_{x-\Delta x}^{x+\Delta x} \frac{dx}{dt}dt = \int_{t-\Delta t}^{t+\Delta t} u\,dt$$

$$x(t+\Delta t) - x(t-\Delta t) = 2\Delta t\bar{u} + O(\Delta t^3)$$

where $\bar{u}$ is some time-averaged value of u. But, time and space are related, so let

$\bar{u} \equiv u(x-a,t)$ - i.e., u at halfway along the trajectory.

$\therefore$ $x(t+\Delta t) - x(t-\Delta t) = 2a = 2\Delta t u(x-a,t)$

→ $a = \Delta t u(x-a,t)$.

To solve this equation, we make a first guess for a (say, the a calculated based on u at current grid point), and iterate until convergence:

$$a^{v+1} = \Delta t u(x-a^{v},t).$$

Now that we have α, i.e., we found that departure point, we use interpolation to find the value of *F* at the departure point, which is the *F* at the mesh point at *t+Dt.*

Convergence: It turns out that iteration will converge if

$$\Delta t \left\| \frac{\partial u}{\partial x} \right\| < 1$$

(for 1-D problem) where || || is an L-2 norm. Therefore time step size is limited by the accurate calculation of trajectory, not by stability.

Steps of Implementation:

1. Solve the α-equation iteratively to obtain the departure point
2. Use interpolation to obtain F at the departure point
3. Replace F at the arrival (mesh) point with that computed at the departure point.

SL methods can be applied to forced problems as well, e.g., $dF/dt = G$ and to systems of equations (see review article and Durran). But SL refers only to the advecton process.

Role of Interpolation

After we find the departure point, we must use interpolation to find the value of F at this point. Because we are using interpolation, some damping or smoothing occurs and conservation is difficult to achieve. The accuracy of the interpolation process determines the accuracy and order of the S-L scheme.

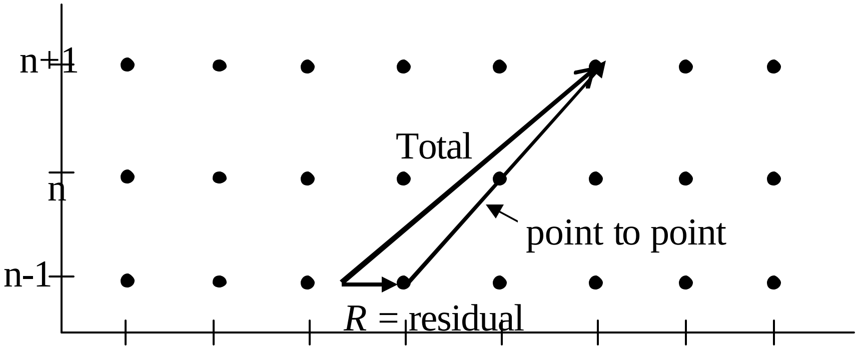
Interpolation is also expensive, especially in multiple dimensions. To overcome this, Ritchies (1986, MWR, 114, 135-146) devised a non-interpolating S-L method, which is a topic of next section.

In a model that advects many variables, such as hydrometeor types and chemical species, S-L method is attractive since all variables (on non-staggered grid) share the same trajectories, and the interpolations can share common interpolation weights.

Because the trajectory calculation is expensive, models based on S-L method usually use non-staggered grid, i.e., all variables are located at the same grid points.

6.4. Non-interpolating S-L Methods

The essence of this approach is that the trajectory vector is broken down into the sum of 2 vectors: one that extends from the arrival point backwards to the gridpoint nearest the departure point, and the other that is approximated as a residual. Schematically, we have:



The mesh-point to mesh-point vector is simple replacement since the values at grid points are always known. The residual *R* is handled using an Eulerian approach, which requires that $| u\Delta t/\Delta x | \leq 1$.

Functionally, the 2-part method works as follows.

Our governing equation is *dF/dt* = 0. Expanding (consider only 1-D case for now), we have

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

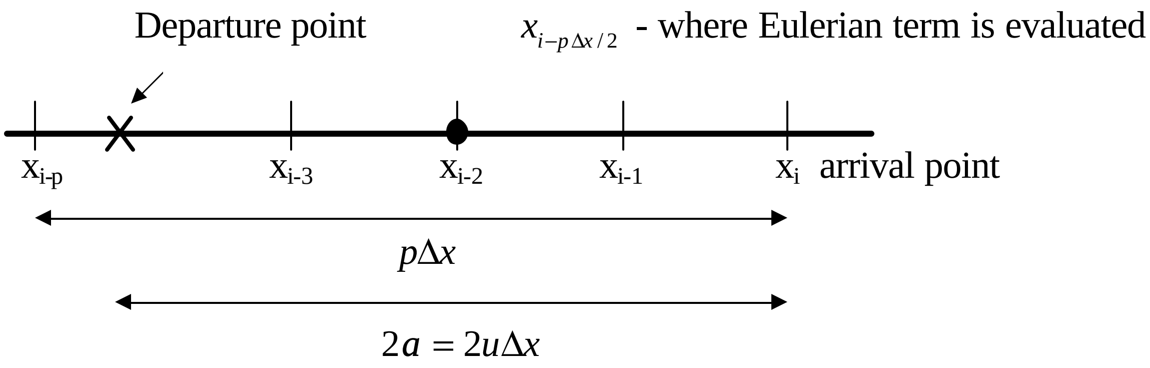
Now, add the quantity $-\frac{p\Delta x}{2\Delta t}\frac{\partial F}{\partial x}$ to both sides, where *p* is an integer.

$$\frac{\partial F}{\partial t} + \left(u - \frac{p\Delta x}{2\Delta t}\right)\frac{\partial F}{\partial x} = -\frac{p\Delta x}{2\Delta t}\frac{\partial F}{\partial x} \quad \rightarrow$$

$$\frac{\partial F}{\partial t} + \frac{p\Delta x}{2\Delta t}\frac{\partial F}{\partial x} = u'\frac{\partial F}{\partial x} \qquad \text{where} \qquad u' \equiv u - \frac{p\Delta x}{2\Delta t} .$$

The LHS now represents the change in *F* following the motion but at a speed $\frac{p\Delta x}{2\Delta t}$, which means that, over two time steps, the total displacement will be *p* grid intervals (can have *p*>1). The residual term on the RHS is simply an Eularian advection term that can be treated using standard methods.

p is chosen to give a displacement to the gridpoint nearest the departure point x - 2α, i.e., *p* is the integer nearest *2uDt/Dx*



$x_i - pDx/2$ is a grid point for p that is even
$x_i - pDx/2$ is a 1/2 grid point for p that is odd

So we have

$$F(x, t+Dt) - F(x - pDx, t - Dt) = - u' \frac{\partial F}{\partial x}(x_i - pDx/2, t).$$

Because $x - pDx$ is a gridpoint (p is an integer), no interpolation is needed to evaluate the LHS. The RHS will be evaluated <u>at</u> a grid point (p-even) or at half-point (p-odd), where in the latter case we will use average of neighboring points.

One can show that this method is stable if

$$\max\left\{2\left|\frac{u'\Delta t}{\Delta x}\right|\right\} \le 1$$

where is the residual Courant number. In general, this is much cheaper than, and of equivalent accuracy to, interpolating methods.

Note that due to the introduction of the residual term – we again have problem with nonlinear aliasing.