Hyperbolic Equation

Wave equation (Typical hyperbolic equation)

$$\frac{\partial^2 \phi}{\partial t^2} - u^2 \frac{\partial^2 \phi}{\partial x^2} = 0$$

By factorizing as $\left(\frac{\partial}{\partial t} - u\frac{\partial}{\partial x}\right)\left(\frac{\partial}{\partial t} + u\frac{\partial}{\partial x}\right)\phi = 0$

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

The advection equation is the simplest hyperbolic equation.

Advection Equation

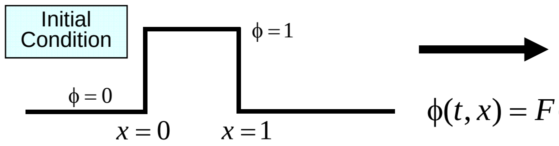
$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0$$

Analytic Solution：

$$\phi = F(x - ut)$$

$F(x)$ is an arbitrary function of x .
Confirming $f(t,x) = F(t\text{-}\Delta t, x\text{-}u\Delta t)$, we can understand the Profile $F(x)$ moves with the speed u.

● Let's solve the advection equation as an initial boundary problem.



$\phi(t, x) = F(t - \Delta t, x - u\Delta t)$

Simple
Finite Difference Method

At the time $t = t^n$ and the position $x = x_i$

■Time derivative $\frac{\partial \phi}{\partial t} \rightarrow \frac{\phi_i^{n+1} - \phi_i^n}{\Delta t}$ （Forward Difference)

■Spatial derivative $\frac{\partial \phi}{\partial x} \rightarrow \frac{\phi_{i+1}^n - \phi_{i-1}^n}{2\Delta x}$ (Center Difference)

Discretized form：

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = -u\frac{\phi_{i+1}^n - \phi_{i-1}^n}{2\Delta x}$$

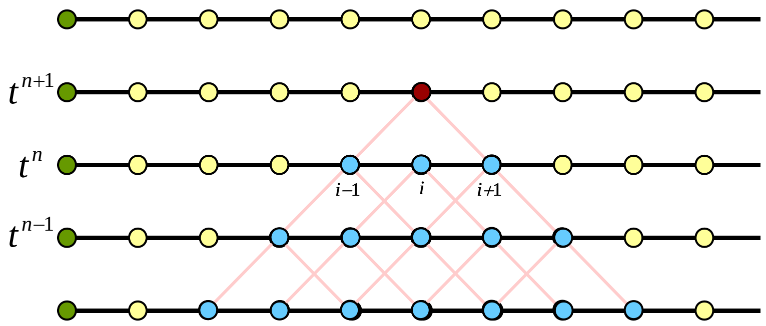
CFL Number

$$\phi_i^{n+1} = \phi_i^n - u\Delta t\frac{\phi_{i+1}^n - \phi_{i-1}^n}{2\Delta x}$$
$$= \phi_i^n - \frac{1}{2}C\left(\phi_{i+1}^n - \phi_{i-1}^n\right)$$

$C = u\Delta t/\Delta x$ is called CFL (Courant-Friedrich-Levy) number.

$$C = \frac{\text{Physical traveling distance } u\Delta t}{\text{Numerical traveling distance } \Delta x}$$

Time and Space Cone for Information travel



If C > 1, the scheme does not have enough information to obtain the value of the next time

Requirment:
The condition $C_{(CFL\ number)} \le 1$ must be statisfied.

Sample Program 3

```
#include "program1.h"
#define    N   101

int  main() {
     double f[N],  fn[N],  x[N],  cfl = 0.5;
     int    j,  icnt = 0;

     while(icnt < 100) {
          for(j=1; j < N-1; j++) {
               fn[j] = f[j] – cfl*0.5*(f[j+1] – f[j-1]);
          }
          for(j=0; j < N; j++) f[j] = fn[j];     /* updating */
     }
}
```

Source Code

For the advection equation

Stability Analysis (1/2)

von Neumann's Method

Assuming the perturbation $\phi_j^n = \delta\phi^n \, \mathrm{e}^{ik\cdot j\Delta x}$

Substituting into $\phi_j^{n+1} = \phi_j^n - \frac{1}{2}C\left(\phi_{j+1}^n - \phi_{j-1}^n\right)$

C : CFL number

$$\delta\phi^{n+1} e^{ik\cdot j\Delta x} = \delta\phi^n e^{ik\cdot j\Delta x} - \frac{1}{2}C\left(\delta\phi^n e^{ik\cdot(j+1)\Delta x} - \delta\phi^n e^{ik\cdot(j-1)\Delta x}\right)$$

For the advection equation

Stability Analysis (2/2)

$$\delta\phi^{n+1} \Big/ \delta\phi^n = 1 - \frac{1}{2}C\left(e^{ik\Delta x} - e^{-ik\Delta x}\right)$$
$$= 1 - C\, i \sin k\Delta x$$

$$\left|\delta\phi^{n+1} / \delta\phi^n\right| = \sqrt{1 + C^2 \sin^2 k\Delta x}$$

Analytically $\left|\delta\phi^{n+1} / \delta\phi^n\right| = 1$

The above formula shows $\left|\delta\phi^{n+1} / \delta\phi^n\right| \ge 1$

It is unstable to apply the forward difference in time and the center difference in space to the advection equation.

Lax Scheme（1/2）

Advection Equation $\phi_j^{n+1} = \phi_j^n - \frac{1}{2}C\left(\phi_{j+1}^n - \phi_{j-1}^n\right)$

$\phi_j^n \rightarrow \frac{1}{2}\left(\phi_{j+1}^n + \phi_{j-1}^n\right)$

$$\phi_j^{n+1} = \frac{1}{2}\left(\phi_{j+1}^n + \phi_{j-1}^n\right) - \frac{1}{2}C\left(\phi_{j+1}^n - \phi_{j-1}^n\right)$$

ϕ_j^n is replaced by the average of ϕ_{j+1}^n and ϕ_{j-1}^n

The result is expected to be diffusive.

Sample Program 4

```
#include "program1.h"
#define   N   101

int  main() {
    double f[N],  fn[N],  x[N],  cfl = 0.5;
    int    j,  icnt = 0;

    while(icnt < 100) {
        for(j=1; j < N-1; j++) {
            fn[j] = 0.5*(f[j+1] – f[j-1])
                - cfl*0.5*(f[j+1] - f[j-1]);
        }
        for(j=0; j < N; j++) f[j] = fn[j];     /* updating */
    }
}
```

Source Code

Lax Scheme（2/2）

von Neumann's stability analysis:

$$\delta\phi^{n+1}/\delta\phi^n = \cos k\Delta x - iC\sin k\Delta x$$

$$\left|\delta\phi^{n+1}/\delta\phi^n\right| = \sqrt{\cos^2 k\Delta x + C^2 \sin^2 k\Delta x}$$

It is understood that the scheme is stable for $C \le 1$

Rewriting to

$$\phi_j^{n+1} = \phi_j^n - \frac{1}{2}C\left(\phi_{j+1}^n - \phi_{j-1}^n\right) + \frac{1}{2}\left(\phi_{j+1}^n - 2\phi_j^n + \phi_{j-1}^n\right)$$

1st order upwind scheme

Since the solution of the advection equation is the profile moving with the velocity u from the upwind to the down direction, it is natural that we apply the backward finite difference to the advection term.

$$\phi_j^{n+1} = \phi_j^n - u\frac{\phi_j^n - \phi_{j-1}^n}{\Delta x}\Delta t = \phi_j^n - C\left(\phi_j^n - \phi_{j-1}^n\right)$$

Sample Program 5

```
#include "program1.h"
#define   N   101

int  main() {
      double f[N],  fn[N],  x[N],  cfl = 0.5;
      int    j,  icnt = 0;

      while(icnt < 100) {
            for(j=1; j < N-1; j++) {
                  fn[j] = f[j] – cfl*(f[j] – f[j-1]);
            }
            for(j=0; j < N; j++) f[j] = fn[j];      /* updating */
      }
}
```

Source Code

1st order upwind scheme

$$\phi_j^{n+1} = \phi_j^n - C\left(\phi_j^n - \phi_{j-1}^n\right)$$
$$= \phi_j^n - \frac{C}{2}\left(\phi_{j+1}^n - \phi_{j-1}^n\right) + \frac{C}{2}\left(\phi_{j+1}^n - 2\phi_{j-1}^n + \phi_{j-1}^n\right)$$

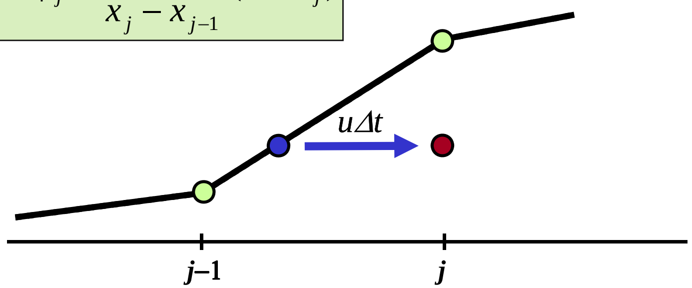
von Neumann's stability analysis:

$$\left|\delta\phi_j^{n+1}/\delta\phi_j^n\right| = \sqrt{(1 - C + C\cos k\Delta x)^2 + C^2 \sin^2 k\Delta x}$$
$$\le 1$$

Other Interpretation

Linear interpolation for the upwind region $x_{j-1} \le x \le x_j$

$$F^n(x) = \phi_j^n + \frac{\phi_j^n - \phi_{j-1}^n}{x_j - x_{j-1}}(x - x_j)$$



At the time $t^n + \Delta t$, $f_j^{n+1} = F^n(x_j - u\Delta t)$ $\quad x = x_j - u\Delta t$

Cubic Semi-Lagrangian Method

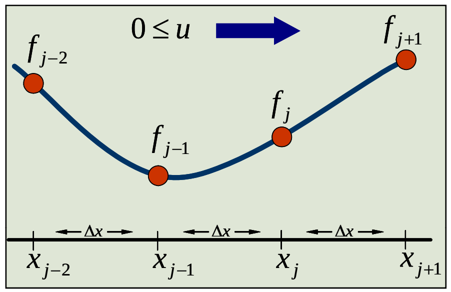
Higher-order Interpolation $F_c(x) = a(x - x_j)^3 + b(x - x_j)^2 + c(x - x_j) + f_j^n$

For upwind region,

$F_c(x_{j-1}) = f_{j-1}^n$

$F_c(x_{j-2}) = f_{j-2}^n$

$F_c(x_{j+1}) = f_{j+1}^n$



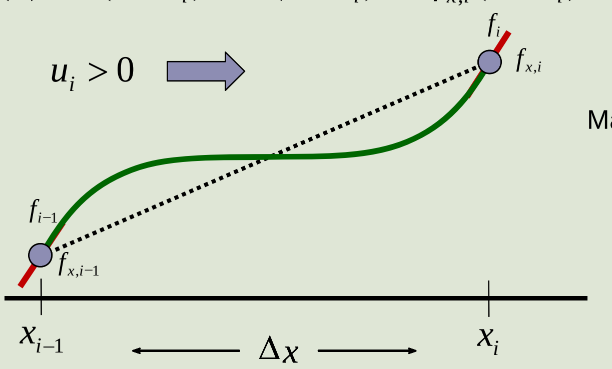
$a = \dfrac{f_{j+1}^n - 3f_j^n + 3f_{j-1}^n - f_{j-2}^n}{6\Delta x^3}$ $\quad b = \dfrac{f_{j+1}^n - 2f_j^n + f_{j-1}^n}{2\Delta x^2}$ $\quad c = \dfrac{2f_{j+1}^n + 3f_j^n - 6f_{j-1}^n + f_{j-2}^n}{6\Delta x}$

$$f_j^{n+1} = F_c^n(x_j - u\Delta t) = a\xi^3 + b\xi^2 + c\xi + f_j^n$$
$$(\xi = -u\Delta t)$$

Source Code

Hermite Interpolation

$$F_{CIP}(x) = a(x-x_i)^3 + b(x-x_i)^2 + f_{x,i}(x-x_i) + f_i$$



Matching Conditions :

$$F(x_i) = f_i$$
$$F_x(x_i) = f_{x,i}$$
$$F(x_{i-1}) = f_{i-1}$$
$$F_x(x_{i-1}) = f_{x,i-1}$$

$$a = \frac{1}{\Delta x^2}\left(f_{x,i} + f_{x,i-1}\right) - \frac{2}{\Delta x}\left(f_i - f_{i-1}\right), \quad b = \frac{1}{\Delta x}\left(2f_{x,i} + f_{x,i-1}\right) - \frac{3}{\Delta x^2}\left(f_i - f_{i-1}\right)$$

CIP Method

$t = t^{n+1}$

$$f_j^{n+1} = F_{CIP}^n(x_j - u\Delta t) = a\xi^3 + b\xi^2 + f_{x,j}^n \xi + f_j^n$$

$$f_{x,j}^{n+1} = \frac{\partial F_{CIP}^n}{\partial x}(x_j - u\Delta t) = 3a\xi^2 + 2b\xi + f_{x,j}^n$$

where $\xi = -u\Delta t$

Source Code