Lecture 3: Numerical Schemes for Time Integrations -- Solving Initial Value Problems

All the numerical time integrations are constructed based on finite difference numerical schemes. FFT and Cubic Spline become useless in the numerical time integration processes. The numerical time integration schemes can be classified into the following two categories:

Explicit Scheme:

- The future information are determined based on the present and the past information
- Easy to program, easy to blowout!
- To avoid blowout, one have to choose shorter time step.
- A short time step means more CPU time

Implicit Scheme:

- The future information are determined based on the future, the present, and the past information
- Difficult to program and/or require more memory
- Stable in large time step.
- As a result, implicit scheme can save CPU time

3.1. Examples of Explicit Scheme

Examples of explicit time integration schemes include Euler method, Runge-Kutta method, the Adams' open Formula, which is also called Adams-Bashforth Formula, and the Lax-Wendroff scheme. Table 3.1. lists the numerical schemes of Euler method and Runge-Kutta method. Table 3.2 lists Adams' open formulae at different orders of accuracy, which will be discussed in the subsection 3.3. The Lax-Wendroff scheme to be discussed in this subsection is commonly used in the fluid simulations.

Table 3.1. Explicit time integrations and their corresponding spatial integrations

<table>
<tr><th>The spatial integrations based on finite differences scheme</th><th>The explicit time integrations</th></tr>
<tr><td>$\dfrac{dy(x)}{dx} = f(x)$, $h = \Delta x$</td><td>$\dfrac{dy(t)}{dt} = f(t,y)$, $h = \Delta t$</td></tr>
<tr><td>1st order integration
$y_{i+1} = y_i + hf_i + O(h^2 f^{(1)})$</td><td>1st order explicit scheme: Euler method
$y^{n+1} = y^n + hf(t^n, y^n) + O(h^2 f^{(1)})$</td></tr>
<tr><td>2nd order integration
Trapezoidal rule
$y_{i+1} = y_i + h\dfrac{f_{i+1} + f_i}{2} + O(h^3 f^{(2)})$</td><td>2nd order Runge-Kutta method
(an explicit scheme)
$(y^*)^{n+\frac{1}{2}} = y^n + \dfrac{h}{2} f(t^n, y^n)$
$y^{n+1} = y^n + hf(t^{n+\frac{1}{2}}, (y^*)^{n+\frac{1}{2}}) + O(h^3 f^{(2)})$</td></tr>
<tr><td>4th order integration
Simpson's rule
$y_{i+1} = y_i$
$+ h(\frac{1}{6} f_i + \frac{4}{6} f_{i+(1/2)} + \frac{1}{6} f_{i+1})$
$+ O(h^5 f^{(4)})$</td><td rowspan="2">4th order Runge-Kutta method
(an explicit scheme)
$(y^*)^{n+\frac{1}{2}} = y^n + \dfrac{h}{2} f(t^n, y^n)$
$(y^{**})^{n+\frac{1}{2}} = y^n + \dfrac{h}{2} f(t^{n+\frac{1}{2}}, (y^*)^{n+\frac{1}{2}})$
$(y^{***})^{n+1} = y^n + hf(t^{n+\frac{1}{2}}, (y^{**})^{n+\frac{1}{2}})$
$y^{n+1} = y^n + h[\frac{1}{6} f(t^n, y^n)$
$+ \frac{2}{6} f(t^{n+\frac{1}{2}}, (y^*)^{n+\frac{1}{2}})$
$+ \frac{2}{6} f(t^{n+\frac{1}{2}}, (y^{**})^{n+\frac{1}{2}})$
$+ \frac{1}{6} f(t^{n+1}, (y^{***})^{n+1})] + O(h^5 f^{(4)})$</td></tr>
<tr><td>3rd order integration
Simpson's $\frac{3}{8}$ rule
$y_{i+1} = y_i$
$+ \frac{h}{3}(\frac{3}{8} f_i + \frac{9}{8} f_{i+\frac{1}{3}} + \frac{9}{8} f_{i+\frac{2}{3}} + \frac{3}{8} f_{i+1})$
$+ O(h^5 f^{(4)})$</td></tr>
</table>

Exercise 3.1.

Solve proton's trajectory in a uniform magnetic field $\mathbf{B} = \mathbf{e}_z B_0$ and electric field $\mathbf{E} = \mathbf{e}_y E_0$ by means of (i) Euler method, (ii) 2nd order Runge-Kutta method, and (iii) 4th order Runge-Kutta method, where $\mathbf{e}_y$ and $\mathbf{e}_z$ are the unit vectors along the y and z directions, respectively. To achieve the same degrees of accuracy, different size of time steps should be used in different numerical schemes. Solve this problem for 100 gyro periods with the following three different initial conditions:

Case 1: $\mathbf{x}(t=0)=0$ and $\mathbf{v}(t=0)=0$

Case 2: $\mathbf{x}(t=0)=0$ and $\mathbf{v}(t=0)=(2.5E_0/B_0)\mathbf{e}_x$

Case 3: $\mathbf{x}(t=0)=0$ and $\mathbf{v}(t=0)=(0.5E_0/B_0)\mathbf{e}_x$

Plot proton's trajectory in both x-y space and in v_x-v_y space.

Compare your numerical results with the theoretical solutions.

The Lax-Wendroff scheme is an explicit scheme. It is good for solving fluid equations with absence of diffusion or dissipation terms. A set of one-dimensional fluid equations without dissipation or diffusion terms (example of such set of fluid equation is the 1-D ideal-MHD equation as shown in Table 3.2), can be written in the following conservative form

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}(\mathbf{U})}{\partial x}=0$$

which can be solved numerically by the second order Lax-Wendroff scheme.

Step 1:

$$\mathbf{U}_{i+\frac{1}{2}}^{n+\frac{1}{2}}=\frac{\mathbf{U}_{i+1}^{n}+\mathbf{U}_{i}^{n}}{2}-\frac{\Delta t}{2\Delta x}\left[\mathbf{F}(\mathbf{U}_{i+1}^{n})-\mathbf{F}(\mathbf{U}_{i}^{n})\right]$$

Step 2:

$$\mathbf{U}_{i}^{n+1}=\mathbf{U}_{i}^{n}-\frac{\Delta t}{\Delta x}[\mathbf{F}(\mathbf{U}_{i+\frac{1}{2}}^{n+\frac{1}{2}})-\mathbf{F}(\mathbf{U}_{i-\frac{1}{2}}^{n+\frac{1}{2}})]$$

Additional examples and advanced discussion on using Lax-Wendroff scheme to solve fluid equations can be found in the book by Richtmyer and Morton (1967).

Exercise 3.2.

Using the second order Lax-Wendroff scheme to solve Korteweg-deVries (KdV) equation

$$\frac{\partial V}{\partial t}+(C_0+V)\frac{\partial V}{\partial x}+\alpha\frac{\partial^3 V}{\partial x^3}=0$$

with uniform boundary condition and a given initial profile $V(x,t=0)$ with a bump at center of the simulation domain. Plot evolutions of spatial profile $V(x,t)$. You can normalize your velocity field by C_0. Study the following two cases: one for $\alpha>0$, and the other for $\alpha<0$.

Table 3.2. The magnetohydrodymic equations in the conservative form

<table>
<tr><td>The mass continuity equation
$\frac{\partial}{\partial t}\rho+\nabla\cdot(\rho\mathbf{V})=0$</td><td rowspan="5">1-D MHD equations
$\frac{\partial}{\partial t}(\rho)+\frac{\partial}{\partial x}(\rho V_x)=0$
$\frac{\partial}{\partial t}(\rho V_x)+\frac{\partial}{\partial x}(\rho V_x^2+p+\frac{B_y^2+B_z^2}{2\mu_0})=0$
$\frac{\partial}{\partial t}(\rho V_y)+\frac{\partial}{\partial x}(\rho V_x V_y-\frac{B_x B_y}{\mu_0})=0$
$\frac{\partial}{\partial t}(\rho V_z)+\frac{\partial}{\partial x}(\rho V_x V_z-\frac{B_x B_z}{\mu_0})=0$
$\frac{\partial}{\partial t}(\frac{1}{2}\rho V^2+\frac{3}{2}p+\frac{B^2}{2\mu_0})$
$+\frac{\partial}{\partial x}[(\frac{1}{2}\rho V^2+\frac{5}{2}p+\frac{B^2}{\mu_0})V_x-\frac{B_x(\mathbf{B}\cdot\mathbf{V})}{\mu_0}]=0$
$\frac{\partial}{\partial t}(B_y)-\frac{\partial}{\partial x}(E_z)=0$
$\frac{\partial}{\partial t}(B_z)+\frac{\partial}{\partial x}(E_y)=0$</td></tr>
<tr><td>The momentum equation
$\frac{\partial}{\partial t}(\rho\mathbf{V})+\nabla\cdot[\rho\mathbf{V}\mathbf{V}+\mathbf{P}+\frac{\mathbf{1}B^2}{2\mu_0}-\frac{\mathbf{B}\mathbf{B}}{\mu_0}]=0$</td></tr>
<tr><td>The energy equation
$\frac{\partial}{\partial t}(\frac{1}{2}\rho V^2+\frac{3}{2}p+\frac{B^2}{2\mu_0})$
$+\nabla\cdot[(\frac{1}{2}\rho V^2+\frac{3}{2}p)\mathbf{V}+\mathbf{P}\cdot\mathbf{V}+\mathbf{q}+\frac{\mathbf{E}\times\mathbf{B}}{\mu_0}]=0$</td></tr>
<tr><td>The MHD Ohm's law
$\mathbf{E}+\mathbf{V}\times\mathbf{B}=0$</td></tr>
<tr><td>Maxwell's equations
$\nabla\cdot\mathbf{B}=0$
$\frac{\partial}{\partial t}\mathbf{B}=-\nabla\times\mathbf{E}$
$\nabla\times\mathbf{B}=\mu_0\mathbf{J}$</td></tr>
</table>

3.2. Examples of Implicit Scheme

Consider a charge particle moving in a uniform strong magnetic field. Momentum equation of this charge particle is

$$\frac{d\mathbf{v}(t)}{dt}=\frac{q}{m}\mathbf{v}(t)\times\mathbf{B}_0 \tag{3.1}$$

The following numerical scheme is an implicit scheme of Eq. (3.1)

$$\mathbf{v}^{n+1}=\mathbf{v}^n+\Delta t\frac{q}{m}\frac{\mathbf{v}^n+\mathbf{v}^{n+1}}{2}\times\mathbf{B}_0 \tag{3.2}$$

Exercise 3.3.

Solve Eq. (3.2) to obtain v_x^{n+1}, v_y^{n+1} and v_z^{n+1} for a given set of v_x^n, v_y^n, v_z^n, B_{0x}, B_{0y}, and B_{0z}.

Exercise 3.4.

Solve proton's trajectory in Exercise 3.1 by means of the implicit scheme discussed this section.

In addition to the gyro motion, the diffusion equation is another type of differential equation, which *should* be solved by an implicit scheme.

The diffusion equation

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2} \tag{3.3}$$

can be solved numerically by one of the following implicit schemes.

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{\kappa}{(\Delta x)^2}\frac{1}{2}[(T_{i+1}^n - 2T_i^n + T_{i-1}^n) + (T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1})] \tag{3.4}$$

or

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{\kappa}{(\Delta x)^2}(T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}) \tag{3.5}$$

or

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{\kappa}{(\Delta x)^2}[(1-\lambda)(T_{i+1}^n - 2T_i^n + T_{i-1}^n) + \lambda(T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1})] \tag{3.6}$$

where $0 < \lambda < 1$. For $\lambda = 1/2$, Eq. (3.6) is reduced to Eq. (3.4). For $\lambda = 1$, Eq. (3.6) is reduced to Eq. (3.5). Eq. (3.4) can be written as

$$-\alpha T_{i-1}^{n+1} + (1+2\alpha)T_i^{n+1} - \alpha T_{i+1}^{n+1} = \alpha T_{i-1}^n + (1-2\alpha)T_i^n + \alpha T_{i+1}^n \tag{3.7}$$

where $\alpha = \dfrac{\kappa \Delta t}{2(\Delta x)^2}$

For given boundary conditions $T(x=0) = T_0$, and $T(x = N_x \Delta x) = T_{N_x}$, Eq. (3.7) can be rewritten in the following tri-diagonal matrix form:

$$\begin{pmatrix} (1+2\alpha) & -\alpha & 0 & \ldots & 0 \\ -\alpha & (1+2\alpha) & -\alpha & \ldots & 0 \\ | & & & & | \\ 0 & \ldots & -\alpha & (1+2\alpha) & -\alpha \\ 0 & \ldots & 0 & -\alpha & (1+2\alpha) \end{pmatrix} \begin{pmatrix} T_1^{n+1} \\ T_2^{n+1} \\ | \\ T_{N_x-2}^{n+1} \\ T_{N_x-1}^{n+1} \end{pmatrix} = \begin{pmatrix} 2\alpha T_0 + (1-2\alpha)T_1^n + \alpha T_2^n \\ \alpha T_1^n + (1-2\alpha)T_2^n + \alpha T_3^n \\ | \\ \alpha T_{N_x-3}^n + (1-2\alpha)T_{N_x-2}^n + \alpha T_{N_x-1}^n \\ \alpha T_{N_x-2}^n + (1-2\alpha)T_{N_x-1}^n + 2\alpha T_{N_x} \end{pmatrix}$$

Exercise 3.5.

Write a subroutine, using Gauss elimination method to solve ($x_1, x_2, ..., x_n$) in the following tri-diagonal set of equations. Limit number of arrays used in your program. There should be no more than five $n \times 1$ arrays used in your program.

$$\begin{pmatrix} b_1 & c_1 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & \dots & 0 \\ | & & & & | \\ 0 & \dots & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & \dots & 0 & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ | \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ | \\ r_{n-1} \\ r_n \end{pmatrix}$$

Examples on how to solve tri-diagonal set of equations numerically can be found at Press et al. (1988).

Exercise 3.6.

Write a program to solve diffusion equation (3.3) for a given initial condition and boundary conditions. Plot evolution of spatial profile $T(x,t)$.

Adams' close formula, which is also called Adams-Moulton formula, is also an implicit scheme. Table 3.4 lists Adams' close formulae at different orders of accuracy, which will be discussed in the next subsection 3.3.

3.3. Predictor-Corrector Method Based on Adams Formula

Predictor-Corrector method is an easy-to-program implicit scheme, but require more memory than the corresponding explicit scheme. We use the Adams' formula to construct the Predictor-Corrector simulation scheme (e.g., Shampine and Gordon, 1975; Press et al., 1988). Tables 3.3 and 3.4 list the Adams' open and close formulae, respectively, at different orders of accuracy. Proof of these Adams' formulae can be found in advanced mathematics textbooks (e.g., Hildebrand, 1976). A 4th order Predictor-Corrector method (e.g., Shampine and Gordon, 1975) is summarized in Table 3.5.

Exercise 3.7.

Using the forth order Predictor-Corrector method described in Table 3.4 to solve Korteweg-deVries (KdV) equation in Exercise 3.2 and proton's trajectory in Exercise 3.4.

Table 3.3. Adams' Open Formulae (also called Adams-Bashforth Formula)

Order of Accuracy	Solving $dy/dt = f$ or $\partial y/\partial t = f$ explicitly with $h = \Delta t$
1st	$y^{n+1} = y^n + h[f^n] + O(h^2 f^{(1)})$
2nd	$y^{n+1} = y^n + h[\frac{3}{2} f^n - \frac{1}{2} f^{n-1}] + O(h^3 f^{(2)})$
3rd	$y^{n+1} = y^n + h[\frac{23}{12} f^n - \frac{16}{12} f^{n-1} + \frac{5}{12} f^{n-2}] + O(h^4 f^{(3)})$
4th	$y^{n+1} = y^n + h[\frac{55}{24} f^n - \frac{59}{24} f^{n-1} + \frac{37}{24} f^{n-2} - \frac{9}{24} f^{n-3}] + O(h^5 f^{(4)})$
5th	$y^{n+1} = y^n$ $+ h[\frac{1901}{720} f^n - \frac{2774}{720} f^{n-1} + \frac{2616}{720} f^{n-2} - \frac{1274}{720} f^{n-3} + \frac{251}{720} f^{n-4}] + O(h^6 f^{(5)})$
6th	$y^{n+1} = y^n$ $+ h[\frac{4277}{1440} f^n - \frac{7923}{1440} f^{n-1} + \frac{9982}{1440} f^{n-2} - \frac{7298}{1440} f^{n-3} + \frac{2877}{1440} f^{n-4} - \frac{475}{1440} f^{n-5}]$ $+ O(h^7 f^{(6)})$

Table 3.4. Adams' Close Formulae (also called Adams-Moulton Formula)

Order of Accuracy	Solving $dy/dt = f$ or $\partial y/\partial t = f$ implicitly with $h = \Delta t$
1st	$y^{n+1} = y^n + h[f^{n+1}] + O(h^2 f^{(1)})$
2nd	$y^{n+1} = y^n + h[\frac{1}{2} f^{n+1} + \frac{1}{2} f^n] + O(h^3 f^{(2)})$
3rd	$y^{n+1} = y^n + h[\frac{5}{12} f^{n+1} + \frac{8}{12} f^n - \frac{1}{12} f^{n-1}] + O(h^4 f^{(3)})$
4th	$y^{n+1} = y^n + h[\frac{9}{24} f^{n+1} + \frac{19}{24} f^n - \frac{5}{24} f^{n-1} + \frac{1}{24} f^{n-2}] + O(h^5 f^{(4)})$
5th	$y^{n+1} = y^n$ $+ h[\frac{251}{720} f^{n+1} + \frac{646}{720} f^n - \frac{264}{720} f^{n-1} + \frac{106}{720} f^{n-2} - \frac{19}{720} f^{n-3}] + O(h^6 f^{(5)})$
6th	$y^{n+1} = y^n$ $+ h[\frac{475}{1440} f^{n+1} + \frac{1427}{1440} f^n - \frac{798}{1440} f^{n-1} + \frac{482}{1440} f^{n-2} - \frac{173}{1440} f^{n-3} + \frac{27}{1440} f^{n-4}]$ $+ O(h^7 f^{(6)})$

Table 3.5. Procedure of the 4th order Predictor-Corrector Method

Initial Steps	Using 4th order Runge-Kutta method to obtain y^1, y^2, and y^3 from y^0.
Predicting Step	Using 4th order Adams Open Formula to predict y^4 from y^0, y^1, y^2, and y^3.
Correcting Steps	Using 4th order Adams Close Formula to correct y^4 from y^1, y^2, y^3, and the predicted y^4 (or corrected y^4 of the last iteration).
	Repeat the correcting step for several times or until *the iteration converges*. [The condition of convergence in an iteration scheme will be discussed in the next section (Section 5).]
. . . .	Repeat the *Predicting* and *Correcting Steps* to advance y from y^4 to y^n

References

Hildebrand, F. B., *Advanced Calculus for Applications, 2nd edition,* Prentice-Hall, Inc., Englewood, Cliffs, New Jersey, 1976.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes (in C or in FORTRAN and Pascal), Cambridg*e University Press, Cambridge, 1988.

Richtmyer, R. D., and K. W. Morton, *Difference Methods for Initial-Value Problems,* 2nd edition, John Wiley & Sons, Inc., 1967.

Shampine, L. F., and M. K. Gordon, *Computer Solution of Ordinary Differential Equation: the Initial Value Problem,* W. H. Freeman and Company, San Francisco, 1975.

System/360 Scientific Subroutine Package Version III, Programmer's Manual, 5th edition, IBM, New York, 1970.