Numerical Partial Differential Equations

Homework I

Name : 정 상 만 **Student Number** : 20190310290

Exercise # 1.1.1

Consider the initial value problem for the equation

$$u_t + a u_x = f(t,x)$$

with $u(0,x) = 0$ and

$$f(t,x) = \begin{cases} 1 & x \geq 0, \\ 0 & otherwise. \end{cases}$$

Assume that a is positive. Show that the solution is given by

$$u(t,x) = \begin{cases} 0 & x \leq 0, \\ x/a & x \geq 0 \text{ and } x - at \leq 0, \\ t & x \geq 0 \text{ and } x - at \geq 0. \end{cases}$$

Solution)

First, we consider the characteristic curve, which is a represented line for the predicted direction of the solution. Before solving this problem, recall the method in the textbook.

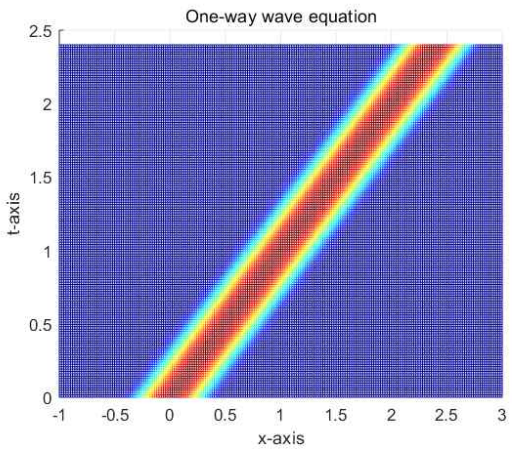


Figure 1. The characteristic curve ($x - at = 0$) of the exact solution of one-way wave equation at $a = 1$.

Given the equation is defined by $u_t + a u_x = f(t,x)$. Let us define the mapping $(t,x) \mapsto (\tau, \xi)$ that $t = \tau$, $\xi = x - at$, then their inverses are $\tau = t$, $x = \xi + a\tau$, and define $u(t,x) = \tilde{u}(\tau, \xi)$, then we can take the partial derivative of τ by

$$\frac{\partial \tilde{u}}{\partial \tau} = u_t(1) + u_x(a) = u_t + a u_x = f(t,x).$$

In this equation, it can be the ordinary differential equation. Hence, take the integration to both sides from 0 to τ, then

$$\int_0^\tau \frac{\partial \tilde{u}}{\partial \sigma}\, d\sigma = \int_0^\tau f(\sigma, \xi + a\sigma)\, d\sigma \;\Rightarrow\; \tilde{u}(\tau,\xi) - \tilde{u}(0,\xi) = \int_0^\tau f(\sigma, \xi + a\sigma)\, d\sigma.$$

Since $\tilde{u}(0,\xi) = 0$ by the assumption, we obtain $\tilde{u}(\tau,\xi) = \int_0^\tau f(\sigma, \xi + a\sigma)\, d\sigma$, and since $u(t,x) = \tilde{u}(\tau,\xi)$, finally we can write $u(t,x) = \int_0^t f(s, x - a(t-s))\, ds$ ($\because\ \xi + a\sigma = (x - at) + as$).

Now, use the range of the equation u and the function f,

CASE 1.
If $x \le 0$, then $f = 0$ and thus $u(t,x) = 0$ (in this case, $x = 0 \Rightarrow f = 1$ but $u = 0$).

CASE 2.

If $x \ge 0$ and $x - at \le 0$, let assume that $x - at + as \le 0$, then $x - at \le as$ and so $\frac{x}{a} - t \le s \Rightarrow t - \frac{x}{a} \ge s$. Note that $0 \le s \le t$. Thus, we can let the integral as the partial form from 0 to $t - x/a$ and from $t - x/a$ to t since x/a is positive. This can be written by

$$u(t,x) = \int_0^{t-x/a} f\, ds + \int_{t-x/a}^t f\, ds \;\Rightarrow\; u(t,x) = \int_0^{t-x/a} 0\, ds + \int_{t-x/a}^t 1\, ds.$$

In this form, let $x^* = x - a(t-s)$, then $f(s,x^*) = 0$ since $x^* \le 0 \;\Leftrightarrow\; t - \frac{x}{a} \ge s$, and also $f(s,x^*) = 1$ since $x^* \ge 0 \Leftrightarrow t - \frac{x}{a} \le s$. By calculation of these, we obtained $u = t - t + x/a = x/a$.

CASE 3.
Finally, $x \ge 0$ and $x - at \ge 0$, that is, $x - at + as \ge 0$, then f is easily calculated by 1 (consider $x^* = x - a(t-s)$). Thus, it can be calculated by

$$u(t,x) = \int_0^t f(s, x - a(t-s))\, ds = \int_0^t 1\, ds = t.$$

Therefore, we obtain the solution of the equation as follows.

$$u(t,x) = \begin{cases} 0 & x \le 0, \\ x/a & x \ge 0 \text{ and } x - at \le 0, \\ t & x \ge 0 \text{ and } x - at \ge 0. \end{cases}$$

■

Exercise # 1.2.1

Consider system $(1.2.2)$ on the interval $[0,1]$, with a equal to 0 and b equal to 1 and with the boundary conditions u^1 equal to 0 at the left and u^1 equal to 1 at the right boundary. Show that if the initial data are given by $u^1(0,x)=x$ and $u^2(0,x)=1$, then the solution is $u^1(t,x)=x$ and $u^2(t,x)=1-t$ for all (t,x) with $0 \le x \le 1$ and $0 \le t$.

Solution)

Note that the system $(1.2.2)$ is given by

$$\begin{pmatrix} u^1 \\ u^2 \end{pmatrix}_t + \begin{pmatrix} a & b \\ b & a \end{pmatrix} \begin{pmatrix} u^1 \\ u^2 \end{pmatrix}_x = 0, \; a=0, b=1 \text{ on the interval } [0,1]$$

with the boundary : $u^1(t,0)=0$ & $u^1(t,1)=1$ and with the initial : $u_0^1(x)=x$ & $u_0^2(x)=1$.

Then, $\begin{pmatrix} a & b \\ b & a \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = A$ and the eigenvalues of A is $\lambda = \pm 1$ since $|A-\lambda I| = \lambda^2 - 1 = 0$ and the matrix consist of the eigenvectors is $P = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. Thus we want to take $\Lambda = PAP^{-1}$, multiplying P to the system $Pu_t + PAu_x = 0 \;\Rightarrow\; Pu_t + PAP^{-1}u_x = 0 \;\Rightarrow$ This system can be rewritten as

$$\begin{pmatrix} u^1+u^2 \\ u^1-u^2 \end{pmatrix}_t + \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} u^1+u^2 \\ u^1-u^2 \end{pmatrix}_x = 0.$$

Since the characteristic curve of the one-way wave equation is $x-at$ and the eigenvalues of A are the characteristic speeds, so we obtain the two families of characteristic curve that are given by

$$\xi_1 = x-t \text{ and } \xi_2 = x+t \text{ at } \lambda = 1, \; -1, \text{ respectively.}$$

Now let $w^1 = u^1+u^2$ and let $w^2 = u^1-u^2$, then the inverse relations are

$$u^1 = \frac{w^1+w^2}{2} \text{ and } u^2 = \frac{w^1-w^2}{2}.$$

The equations satisfied by w^1 and w^2 are $w_t^1 + w_x^1 = 0$ & $w_t^2 - w_x^2 = 0$. Also we can define the new initial conditions and boundary conditions of w's. Since $w^1 = u^1+u^2$ and $w^2 = u^1-u^2$, we obtain:

$$w^1(0,x) = x+1 \text{ and } w^2(0,x) = x-1$$

and obtain the boundary conditions using the inverse relations:

$$w^1(t,0) = -w^2(t,0) + 2u^1(t,0) = -w^2(t,0) \text{ and } w^2(t,1) = -w^1(t,1) + 2.$$

Now, consider the regions for the characteristics in order to determine the solution in the interior.

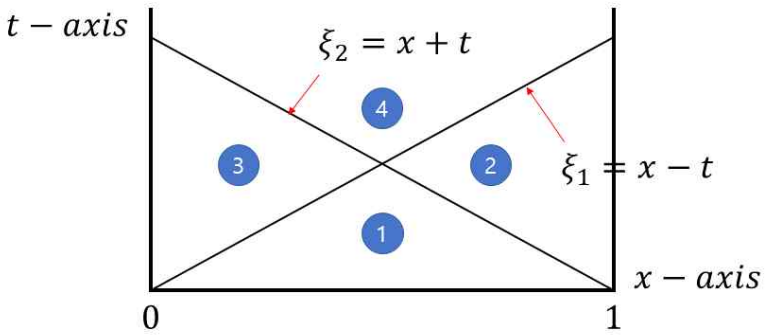


Figure 2. The characteristic curves and the interior regions assigned the curve.

REGION 1.

In region 1, the solution is determined by the initial conditions. Thus, using the characteristics and the initial data we obtain

$$w^1(t,x) = w^1(0,x-t) = x-t+1 \text{ and } w^2(t,x) = w^2(0,x+t) = x+t-1$$

and then, using the inverse relations, we have

$$u^1(t,x) = \frac{w^1(t,x)+w^2(t,x)}{2} = x \text{ and } u^2(t,x) = \frac{w^1(t,x)-w^2(t,x)}{2} = 1-t.$$

REGION 2.

In region 2, the solution of w^1 also determined as the case of region 1, but the solution of w^2 need to consider the boundary condition at $x=1$. Hence, the boundary condition of w^2 at $x=1$ is:

$$w^2(t,1) = -w^1(t,1)+2 = t-2+2 = t \text{ since } w^1(t,x) = w^1(0,x-t) = x-t+1.$$

Thus, $w^2(t,x) = w^2(x+t-1,1) = x+t-1$ and therefore,

$$u^1(t,x) = \frac{w^1(t,x)+w^2(t,x)}{2} = x \text{ and } u^2(t,x) = \frac{w^1(t,x)-w^2(t,x)}{2} = 1-t.$$

REGION 3.

In region 3, the solution of w^2 also determined as the case of region 1, but the solution of w^1 need to consider the boundary condition at $x=0$. Hence, the boundary condition of w^1 at $x=0$ is:

$$w^1(t,0) = -w^2(t,0) = -t+1 \text{ since } w^2(t,x) = x+t-1 \text{ in region 1.}$$

Thus, $w^1(t,x) = w^1(t-x,0) = -t+x+1$ and therefore,

$$u^1(t,x) = \frac{w^1(t,x)+w^2(t,x)}{2} = x \text{ and } u^2(t,x) = \frac{w^1(t,x)-w^2(t,x)}{2} = 1-t.$$

■

Exercise # 1.3.1

For values of x in the interval $[-1,3]$ and t in $[0,2.4]$, solve the one-way wave equation

$$u_t + u_x = 0$$

with the initial data

$$u(0,x) = \begin{cases} \cos^2 \pi x & |x| \le 1/2, \\ 0 & otherwise. \end{cases}$$

and the boundary data $u(t,-1) = 0$.

Use the following four schemes for $h = 1/10,\ 1/20$, and $1/40$.

(a) Forward-time backward-space scheme with $\lambda = 0.8$.

(b) Forward-time central-space scheme with $\lambda = 0.8$.

(c) Lax-Friedrichs scheme with $\lambda = 0.8$ and $\lambda = 1.6$.

(d) Leapfrog scheme with $\lambda = 0.8$.

For schemes (b), (c), and (d), at the right boundary use the condition $v_M^{n+1} = v_{M-1}^{n+1}$ where $x_M = 3$. For scheme (d) use scheme (b) to compute the solution at $n = 1$.

For each scheme determine whether the scheme is a useful or useless scheme. For the purposes of this exercise only, a scheme will be useless if $|v_m^n|$ is greater than 5 for any value of m and n. It will be regarded as a useful scheme if the solution looks like a reasonable approximation to the solution of the differential equations. Graph or plot several solutions at the last time they were computed. What do you notice about the "blow-up time" for the useless schemes as the mesh size decreases? Is there a pattern to these solutions? For the useful cases, how does the error decrease as the mesh decreases; i.e., as h decreases by one-half, by how much does the error decrease?

Solution)

By (1.3.2) in the textbook, we obtained the formula of the equation as follows.

$$v_m^{n+1} = v_m^n + \frac{a}{2}\lambda(v_{m+1}^n - v_{m-1}^n),\ \lambda = \frac{k}{h},\ a = 1. \qquad \text{(FTBS)}$$

Let $\lambda = 0.8$ and let $\Delta x = h = 1/10,\ 1/20,\ 1/40$, then $\Delta t = k = h\lambda = 0.08,\ 0.04,\ 0.02$. Similarly, other schemes are written as follows.

$$v_m^{n+1} = v_m^n - \frac{1}{2}a\lambda(v_{m+1}^n - v_{m-1}^n) \qquad \text{(FTCS)}$$

$$v_m^{n+1} = v_m^{n-1} - a\lambda(v_{m+1}^n - v_{m-1}^n) \qquad \text{(Leapfrog)}$$

$$v_m^{n+1} = \frac{1}{2}(v_{m+1}^n + v_{m-1}^n) - \frac{1}{2}a\lambda(v_{m+1}^n - v_{m-1}^n) \qquad \text{(Lax-Friedrichs)}$$

Now we can calculate the numerical solutions of the equation and their errors. Refer to the below.

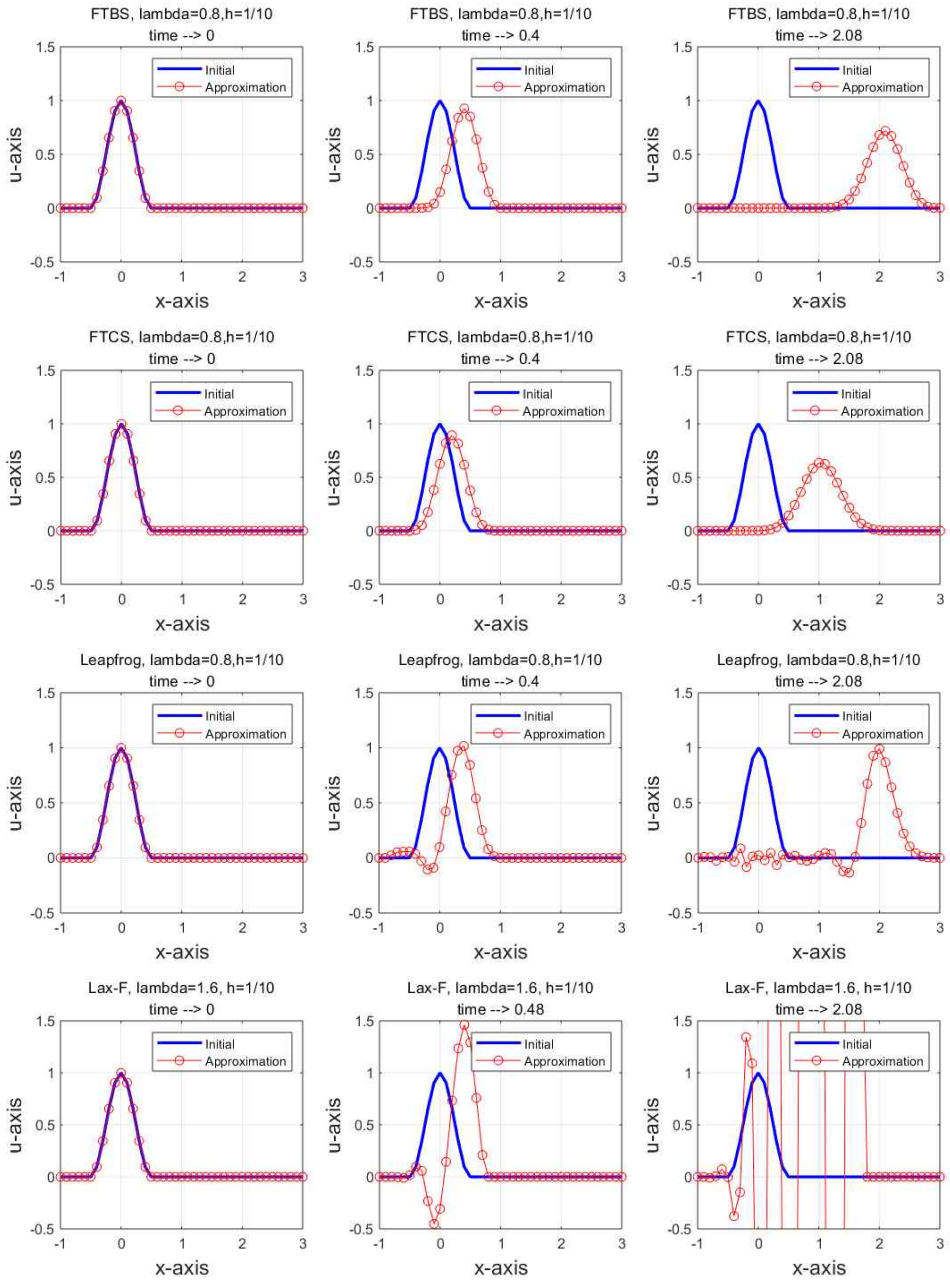


Figure 3. Comparison of each finite difference scheme at time. The bottom graph (Lax-Friedrichs) was blown up at $\lambda = 1.6$.

We set the value of h is $1/10$ and the factor $\lambda = 0.8$ for all schemes except Lax-Friedrichs scheme, with the same h. In Figure 2, we can find the fact that the Lax-Friedrichs scheme do not maintain the form of the solution and blow up at the high speed for t: this t is called 'blow-up time'.

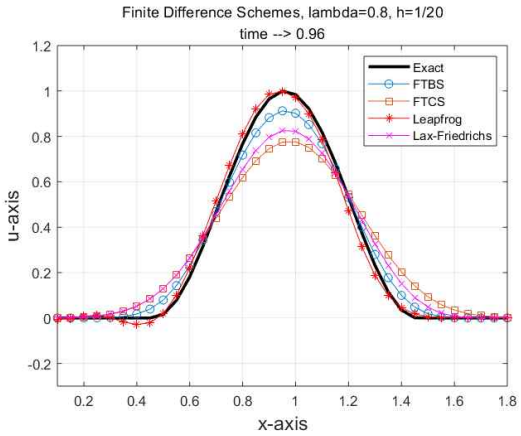


Figure 4. Comparison of each finite difference scheme with $h = 1/20$ and $\lambda = 0.8$, at time $t = 0.96$.

The step size of space is increasing, then the difference of each scheme is decreasing, but Leapfrog scheme always dominants for any step size of h in this problem.

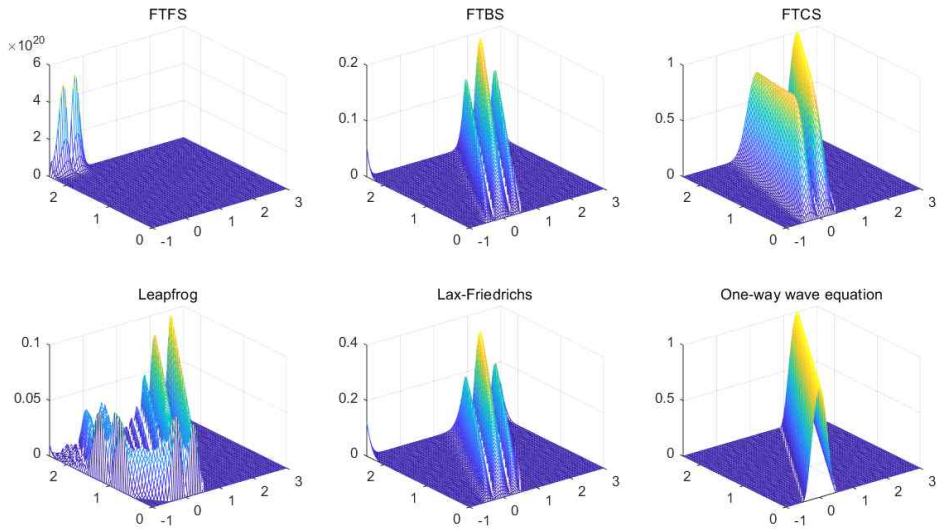


Figure 5. Comparison of the true error for each schemes, $h = 1/20$, $\lambda = 0.8$.

■

Exercise # 1.4.1

Show that the forward-time central-space scheme:

$$\frac{v_m^{n+1}-v_m^n}{k}+a\frac{v_{m+1}^n-v_{m-1}^n}{2h}=0$$

is consistent with equation $u_t+au_x=0$.

Solution)

For $u_t+au_x=0 \Leftrightarrow Pu=0$, $P=\frac{\partial}{\partial t}+a\frac{\partial}{\partial x}$, define the function $\phi(t,x)$ that is smooth from definition 1.4.2, such that $P\phi=\phi_t+a\phi_x$. For the FTCS scheme, the difference operator $P_{k,h}$ is given as follows.

$$P_{k,h}\phi=\frac{\phi_m^{n+1}-\phi_m^n}{k}+a\frac{\phi_{m+1}^n-\phi_{m-1}^n}{2h},\ \phi_m^n=\phi(nk,mh).$$

We begin with the Taylor series of the function ϕ in t and x about (t_n,x_m). We have that

$$\phi_m^{n+1}=\phi_m^n+k\phi_t+\frac{1}{2}k^2\phi_{tt}+O(k^3),$$

$$\phi_{m+1}^n=\phi_m^n+h\phi_x+\frac{1}{2}h^2\phi_{xx}+O(h^3),$$

$$\phi_{m-1}^n=\phi_m^n-h\phi_x+\frac{1}{2}h^2\phi_{xx}+O(h^3),$$

$$\begin{aligned}P_{k,h}\phi&=\frac{1}{k}\phi_m^n+\phi_t+\frac{1}{2}k\phi_{tt}-\frac{1}{k}\phi_m^n+O(k^3)+\frac{a}{2h}\phi_m^n+\frac{a}{2}\phi_x+\frac{a}{4}h\phi_{xx}-\frac{a}{2h}\phi_m^n+\frac{a}{2}\phi_x+\frac{a}{4}h\phi_{xx}+O(h^3)\\&=\phi_t+\frac{1}{2}k\phi_{tt}+a\phi_x+\frac{a}{2}h\phi_{xx}+O(k^3+h^3).\end{aligned}$$

Subtracting from $P\phi$ to $P_{k,h}\phi$ results in:

$$P\phi-P_{k,h}\phi=-\frac{1}{2}k\phi_{tt}-\frac{a}{2}h\phi_{xx}+O(k^3+h^3)\to 0 \text{ as } k,h\to 0.$$

Therefore, by definition 1.4.2, forward-time central-space scheme is consistent.

■

Exercise # 1.4.2

Show that the leapfrog scheme:

$$\frac{v_m^{n+1}-v_m^{n-1}}{2k}+a\frac{v_{m+1}^n-v_{m-1}^n}{2h}=0$$

is consistent with equation $u_t+au_x=0$.

Solution)

For $u_t+au_x=0 \Leftrightarrow Pu=0$, $P=\frac{\partial}{\partial t}+a\frac{\partial}{\partial x}$, define the function $\phi(t,x)$ that is smooth from definition 1.4.2, such that $P\phi=\phi_t+a\phi_x$. For the leapfrog scheme, the difference operator $P_{k,h}$ is given as follows.

$$P_{k,h}\phi=\frac{\phi_m^{n+1}-\phi_m^{n-1}}{2k}+a\frac{\phi_{m+1}^n-\phi_{m-1}^n}{2h},\ \phi_m^n=\phi(nk,mh).$$

We begin with the Taylor series of the function ϕ in t and x about (t_n,x_m). We have that

$$\phi_m^{n+1}=\phi_m^n+k\phi_t+\frac{1}{2}k^2\phi_{tt}+\frac{1}{6}k^3\phi_{ttt}+O(k^4),$$

$$\phi_m^{n-1}=\phi_m^n-k\phi_t+\frac{1}{2}k^2\phi_{tt}-\frac{1}{6}k^3\phi_{ttt}+O(k^4),$$

$$\phi_{m+1}^n=\phi_m^n+h\phi_x+\frac{1}{2}h^2\phi_{xx}+\frac{1}{6}h^3\phi_{xxx}+O(h^4),$$

$$\phi_{m-1}^n=\phi_m^n-h\phi_x+\frac{1}{2}h^2\phi_{xx}-\frac{1}{6}h^3\phi_{xxx}+O(h^4),$$

$$P_{k,h}\phi=\frac{1}{2k}\left(2k\phi_t+\frac{1}{3}k^3\phi_{ttt}\right)+\frac{a}{2h}\left(2h\phi_x+\frac{1}{3}h^3\phi_{xxx}\right)+O(k^3+h^3).$$

Subtracting from $P\phi$ to $P_{k,h}\phi$ results in:

$$P\phi-P_{k,h}\phi=\frac{1}{6}k^2\phi_{ttt}+\frac{a}{6}h^2\phi_{xxx}+O(k^3+h^3)\to 0 \text{ as } k,h\to 0.$$

Therefore, by definition 1.4.2, leapfrog scheme is consistent.

■

Exercise # 1.5.1

Show that the schemes of the form

$$v_m^{n+1} = \alpha v_{m+1}^n + \beta v_{m-1}^n$$

are stable if $|\alpha| + |\beta|$ is less than or equal to 1. Conclude that the Lax-Friedrichs scheme:

$$\frac{v_m^{n+1} - 1/2(v_{m+1}^n + v_{m-1}^n)}{k} + a\frac{v_{m+1}^n - v_{m-1}^n}{2h} = 0$$

is stable if $|a\lambda|$ is less than or equal to 1.

Solution)

$$\begin{aligned}
\sum_{m=-\infty}^{\infty} \left|v_m^{n+1}\right|^2 &= \sum_{m=-\infty}^{\infty} \left|\alpha v_{m+1}^n + \beta v_{m-1}^n\right|^2 \\
&\le \sum_{m=-\infty}^{\infty} |\alpha|^2 \left|v_{m+1}^n\right|^2 + 2|\alpha||\beta| \left|v_{m+1}^n\right| \left|v_{m-1}^n\right| + |\beta|^2 \left|v_{m-1}^n\right|^2 \\
&\le \sum_{m=-\infty}^{\infty} |\alpha|^2 \left|v_{m+1}^n\right|^2 + |\alpha||\beta| \left(\left|v_{m+1}^n\right|^2 + \left|v_{m-1}^n\right|^2\right) + |\beta|^2 \left|v_{m-1}^n\right|^2 \quad \cdots \quad (*)
\end{aligned}$$

Since $2xy \le x^2 + y^2$ and the sum can be split over the terms with index m and those with index $m+1$ and the index can be shifted so that all terms have the index m:

$$\begin{aligned}
(*) &= \sum_{m=-\infty}^{\infty} |\alpha|^2 \left|v_{m+1}^n\right|^2 + |\alpha||\beta| \left|v_{m+1}^n\right|^2 + \sum_{m=-\infty}^{\infty} |\alpha||\beta| \left|v_{m-1}^n\right|^2 + |\beta|^2 \left|v_{m-1}^n\right|^2 \\
&= \sum_{m=-\infty}^{\infty} |\alpha|^2 \left|v_m^n\right|^2 + |\alpha||\beta| \left|v_m^n\right|^2 + \sum_{m=-\infty}^{\infty} |\alpha||\beta| \left|v_m^n\right|^2 + |\beta|^2 \left|v_m^n\right|^2 \\
&= \sum_{m-\infty}^{\infty} \left(|\alpha|^2 + 2|\alpha||\beta| + |\beta|^2\right) \left|v_m^n\right|^2 \\
&= (|\alpha| + |\beta|)^2 \sum_{m=-\infty}^{\infty} \left|v_m^n\right|^2
\end{aligned}$$

$$\sum_{m=-\infty}^{\infty} \left|v_m^{n+1}\right|^2 \le (|\alpha| + |\beta|)^2 \sum_{m=-\infty}^{\infty} \left|v_m^n\right|^2$$

Thus, $\Rightarrow \sum_{m=-\infty}^{\infty} \left|v_m^n\right|^2 \le (|\alpha| + |\beta|)^{2n} \sum_{m=-\infty}^{\infty} \left|v_m^0\right|^2, \ \forall\, n.$

Hence, if $|\alpha| + |\beta| \le 1$, the scheme will be stable. The Lax-Friedrichs scheme can be written by

$$v_m^{n+1} = 1/2(v_{m+1}^n + v_{m-1}^n) - 1/2a\lambda(v_{m+1}^n - v_{m-1}^n) = 1/2(1 - a\lambda)v_{m+1}^n + 1/2(1 + a\lambda)v_{m-1}^n.$$

Therefore, $|\alpha + \beta| = |a\lambda| \le |\alpha| + |\beta| = |a\lambda - 1| + |a\lambda + 1| \le 1$ if the scheme is stable.

■

Exercise # 1.5.2

By multiplying the leapfrog scheme by $v_m^{n+1}+v_m^{n-1}$ and summing over all values of m, obtain the relation

$$\sum_{m=-\infty}^{\infty}\left|v_m^{n+1}\right|^2+\left|v_m^n\right|^2+a\lambda(v_m^{n+1}v_{m+1}^n-v_{m+1}^{n+1}v_m^n)=\sum_{m-\infty}^{\infty}\left|v_m^n\right|^2+\left|v_m^{n-1}\right|^2+a\lambda\left(v_m^nv_{m+1}^{n-1}-v_{m+1}^nv_m^{n-1}\right).$$

Show that the leapfrog scheme:

$$\frac{v_m^{n+1}-v_m^{n-1}}{2k}+a\frac{v_{m+1}^n-v_{m-1}^n}{2h}=0$$

is stable for $|a\lambda|<1$.

Solution)

The formula for the leapfrog scheme is derived by $v_m^{n+1}=v_m^{n-1}+a\lambda(v_{m+1}^n-v_{m-1}^n)$. Hence,

$$\begin{aligned}\sum_{m=-\infty}^{\infty}\left|v_m^{n+1}\right|^2&=\sum_{m=-\infty}^{\infty}\left|v_m^{n-1}+a\lambda(v_{m+1}^n-v_{m-1}^n)\right|^2\\&\le\sum_{m-\infty}^{\infty}\left|v_m^{n-1}\right|^2+2a\lambda\left|(v_m^{n-1}v_{m+1}^n-v_m^{n-1}v_{m-1}^n)\right|+a^2\lambda^2\left|v_{m+1}^n-v_{m-1}^n\right|^2\\&\le\sum_{m=-\infty}^{\infty}\left|v_m^{n-1}\right|^2+|a||\lambda|\left(\left|v_m^{n-1}\right|^2+\left|v_{m+1}^n-v_{m-1}^n\right|^2\right)+|a|^2|\lambda|^2\left|v_{m+1}^n-v_{m-1}^n\right|^2\cdots(*)\end{aligned}$$

Since $2xy\le x^2+y^2$ and the sum can be split over the terms with index m and those with index $m+1$ and the index can be shifted so that all terms have the index m:

$$\begin{aligned}(*)&=\sum_{m=-\infty}^{\infty}\left|v_m^{n-1}\right|^2+|a||\lambda|\left|v_m^{n-1}\right|^2+|a||\lambda|\left|v_{m+1}^n-v_{m-1}^n\right|^2+|a|^2|\lambda|^2\left|v_{m+1}^n-v_{m-1}^n\right|^2\\&=\sum_{m-\infty}^{\infty}\left|v_m^{n-1}\right|^2+|a||\lambda|\left|v_m^{n-1}\right|^2+\sum_{m=-\infty}^{\infty}|a||\lambda|\left|v_{m+1}^n-v_{m-1}^n\right|^2+|a|^2|\lambda|^2\left|v_{m+1}^n-v_{m-1}^n\right|^2\\(*)&=\sum_{m-\infty}^{\infty}\left|v_m^{n-1}\right|^2+|a||\lambda|\left|v_m^{n-1}\right|^2+\sum_{m=-\infty}^{\infty}|a||\lambda|\left|v_m^n-v_m^n\right|^2+|a|^2|\lambda|^2\left|v_m^n-v_m^n\right|^2\\&=\sum_{m-\infty}^{\infty}\left|v_m^{n-1}\right|^2+|a||\lambda|\left|v_m^{n-1}\right|^2\\&=(1+a\lambda)\sum_{m-\infty}^{\infty}\left|v_m^{n-1}\right|^2\end{aligned}$$

By the assumption,

$$\Rightarrow\sum_{m=-\infty}^{\infty}\left|v_m^{n+1}\right|^2+\left|v_m^n\right|^2\le(1+a\lambda)\sum_{m=-\infty}^{\infty}\left|v_m^n\right|^2+\left|v_m^{n-1}\right|^2=(1+a\lambda)\sum_{m=-\infty}^{\infty}\left|v_m^1\right|^2+\left|v_m^0\right|^2$$

$$\Rightarrow\sum_{m=-\infty}^{\infty}\left|v_m^{n+1}\right|^2\le(1+a\lambda)\sum_{m=-\infty}^{\infty}\left|v_m^1\right|^2\text{ and }\sum_{m=-\infty}^{\infty}\left|v_m^n\right|^2\le(1+a\lambda)\sum_{m=-\infty}^{\infty}\left|v_m^0\right|^2$$

← 이 부분 다시 확인 ← refer to the textbook, Chp 4. Stability for Multistep Schemes.

■

Exercise # 1.6.1

Show that the following modified Lax-Friedrichs scheme for the one-way wave equation, $u_t + au_x = f$, given by

$$v_m^{n+1} = \frac{1}{2}(v_{m+1}^n + v_{m-1}^n) - \frac{a\lambda}{1+(a\lambda)^2}(v_{m+1}^n - v_{m-1}^n) + kf_m^n$$

is stable for all values of λ. Discuss the relation of this explicit and unconditionally stable scheme to Theorem 1.6.2.

Solution)

$$\begin{aligned} \left|v_m^{n+1}\right|^2 &\le \left|\frac{1}{2}(v_{m+1}^n + v_{n-1}^m) - \frac{a\lambda}{1+(a\lambda)^2}(v_{m+1}^n - v_{m-1}^n) + kf_m^n\right|^2 \\ &= \left|v_{m+1}^n\left(\frac{1}{2} - \frac{a\lambda}{1+(a\lambda)^2}\right) + v_{m-1}^n\left(\frac{1}{2} + \frac{a\lambda}{1+(a\lambda)^2}\right) + kf_m^n\right|^2 \end{aligned}$$

Let $\left(\frac{1}{2} - \frac{a\lambda}{1+(a\lambda)^2}\right) = \alpha$ and $\left(\frac{1}{2} + \frac{a\lambda}{1+(a\lambda)^2}\right) = \beta$ and $f_m^n = 0$, finally we can consider the formula $v_m^{n+1} = \alpha v_{m+1}^n + \beta v_{m-1}^n$ and thus, by exercise 1.5.1, the scheme is stable if $|\alpha| + |\beta| \le 1$.

Since $\alpha + \beta = 1$, this scheme satisfied the condition $|\alpha + \beta| \le |\alpha| + |\beta| \le 1$ for every λ.
Thus, this scheme is explicit and unconditionally stable scheme.

Theorem 1.6.2 says there are no explicit, unconditionally stable, consistent finite difference schemes for hyperbolic systems of partial differential equations. Hence we now focus on the condition for **consistency**. So, check the consistency for this scheme.

Let us define the smooth function ϕ be a test function for the scheme and for the difference operator $P = \frac{\partial}{\partial t} + a\frac{\partial}{\partial x}$, we can set $P\phi = \phi_t + a\phi_x$. Use the Taylor expansion, we obtain:

$$\begin{aligned} \phi_m^{n+1} &= \phi_m^n + k\phi_t + \frac{1}{2}k^2\phi_{tt} + \frac{1}{6}k^3\phi_{ttt} + O(k^4) \\ \phi_{m+1}^n &= \phi_m^n + h\phi_x + \frac{1}{2}h^2\phi_{xx} + \frac{1}{6}h^3\phi_{xxx} + O(h^4) \\ \phi_{m-1}^n &= \phi_m^n - h\phi_x + \frac{1}{2}h^2\phi_{xx} - \frac{1}{6}h^3\phi_{xxx} + O(h^4) \end{aligned}$$

For the Lax-Friedrichs scheme, put the function ϕ then we can write

$$P_{k,h}\phi = \phi_m^{n+1} - 1/2(\phi_{m+1}^n + \phi_{m-1}^n) + \frac{a\lambda}{1+(a\lambda)^2}(\phi_{m+1}^n - \phi_{m-1}^n).$$

But, put the derived expansion, there is k or h term for ϕ_t or ϕ_x since the factor $a\lambda/(1+(a\lambda)^2)$.
Thus, $P\phi - P_{k,h}\phi$ does not converges to 0 as $k,h \to 0$ & if it possible, $hk^{-1} \to 0$.
Therefore, this scheme is not consistency.

■

Exercise # 2.1.4

Use an argument similar to that used in $(2.1.11)$ to show that the initial value problem for the equation $u_t = u_{xxx}$ is well-posed.

Solution)

$$\frac{\partial u}{\partial x} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{iwx} iw\, \hat{u}(w)\, dw \;\Rightarrow\; \frac{\partial \hat{u}}{\partial x} = iw\hat{u}(w) \text{ by the inversion transform.}$$

Use the above, we obtain

$$\hat{u}_{xxx} = (iw)^3 \hat{u}(w) = -iw^3 \hat{u}(w) \;\Rightarrow\; \hat{u}_t = \hat{u}_{xxx} = -iw^3 \hat{u}(w).$$

By the textbook page 43, the solution of equation can be $\hat{u}(t,w) = e^{-iw^3 t} \hat{u}_0(w)$.

Thus, using $(2.1.11)$, then

$$\begin{aligned} \int_{-\infty}^{\infty} |u(t,x)|^2 dx &= \int_{-\infty}^{\infty} \left|\hat{u}(t,w)\right|^2 dw \\ &= \int_{-\infty}^{\infty} \left|e^{-iw^3 t} \hat{u}_0(w)\right|^2 dw \\ &= \int_{-\infty}^{\infty} \left|\hat{u}_0(w)\right|^2 dw \\ &= \int_{-\infty}^{\infty} \left|u_0(x)\right|^2 dx \end{aligned}$$

since $\left|e^{i\theta}\right| = 1$ and Parseal's relations ($\int_{-\infty}^{\infty} |u(x)|^2 dx = \int_{-\infty}^{\infty} \left|\hat{u}(w)\right|^2 dw$). Let $C_T = 1$, then the above satisfy the definition of well-posed (def.1.5.2). Thus, the equation $u_t = u_{xxx}$ is well-posed.

■

Exercise # 2.1.5

Use an argument similar to that used in $(2.1.11)$ to show that the initial value problem for the equation $u_t + u_x + bu = 0$ is well-posed.

Solution)

$$\frac{\partial u}{\partial x} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{iwx} iw\, \hat{u}(w)\, dw \;\Rightarrow\; \frac{\partial \hat{u}}{\partial x} = iw\hat{u}(w) \text{ by the inversion transform.}$$

Use the above, we can write

$$\hat{u}_t = -(iw + b)\hat{u}(w).$$

By the textbook page 43, the solution of equation can be $\hat{u}(t,w) = e^{-(iw+b)t}\hat{u}_0(w)$.

Thus, using $(2.1.11)$, then

$$\begin{aligned}
\int_{-\infty}^{\infty} |u(t,x)|^2 dx &= \int_{-\infty}^{\infty} |\hat{u}(t,w)|^2 dw \\
&= \int_{-\infty}^{\infty} \left| e^{-(iw+b)t}\hat{u}_0(w) \right|^2 dw \\
&= e^{-bt} \int_{-\infty}^{\infty} \left| \hat{u}_0(w) \right|^2 dw \\
&= e^{-bt} \int_{-\infty}^{\infty} |u_0(x)|^2 dx
\end{aligned}$$

since $|e^{i\theta}| = 1$ and Parseal's relations ($\int_{-\infty}^{\infty} |u(x)|^2 dx = \int_{-\infty}^{\infty} |\hat{u}(w)|^2 dw$). Let $C_T = e^{-bt}$, then the above satisfy the definition of well-posed (def.1.5.2). Thus, the equation $u_t = u_{xxx}$ is well-posed.

■

Exercise # 2.2.1

Show that the backward-time central-space scheme

$$\frac{v_m^{n+1}-v_m^n}{k}+a\frac{v_{m+1}^{n+1}-v_{m-1}^{n+1}}{2h}=0$$

is consistent with equation $u_t+au_x=0$ and is unconditionally stable.

Solution)

1. Consistency

Let the smooth function ϕ that $P\phi=\phi_t+a\phi_x$ and $P_{k,h}\phi=\frac{\phi_m^{n+1}-\phi_m^n}{k}+a\frac{\phi_{m+1}^{n+1}-\phi_{m-1}^{n+1}}{2h}$. Then using the Taylor expansion, we can write

$$\phi_m^{n+1}=\phi_m^n+k\phi_t+\frac{1}{2}k^2\phi_{tt}+\frac{1}{6}k^3\phi_{ttt}+O(k^4)$$
$$\phi_{m+1}^{n+1}=\phi_m^n+h\phi_x+\frac{1}{2}h^2\phi_{xx}+\frac{1}{6}h^3\phi_{xxx}+O(h^4)$$
$$\phi_{m-1}^{n+1}=\phi_m^n-h\phi_x+\frac{1}{2}h^2\phi_{xx}-\frac{1}{6}h^3\phi_{xxx}+O(h^4)$$

Thus, $P_{k,h}\phi$ can be written as

$$\begin{aligned}P_{k,h}\phi&=\phi_t+\frac{1}{2}k\phi_{tt}+\frac{1}{6}k^2\phi_{ttt}+O(k^3)+a\phi_x+ah\phi_{xx}+\frac{1}{6}ah^2\phi_{xxx}+O(h^3)\\&=\phi_t+\frac{1}{2}k\phi_{tt}+\frac{1}{6}k^2\phi_{ttt}+a\phi_x+ah\phi_{xx}+\frac{1}{6}ah^2\phi_{xxx}+O(k^3+h^3)\end{aligned}$$

Therefore, $P\phi-P_{k,h}\phi\to 0$ as $t,h\to 0$. $\therefore$ this scheme is consistent.

2. Unconditionally stable.

Consider $v_m^n=g^ne^{iw\theta}$, then the scheme is

$$\frac{g^{n+1}e^{im\theta}-g^ne^{im\theta}}{k}+a\frac{g^{n+1}e^{i(m+1)\theta}-g^{n+1}e^{i(m-1)\theta}}{2h}=0\ \Rightarrow\ g^ne^{im\theta}\left\{\frac{g-1}{k}+a\frac{ge^{i\theta}-ge^{-i\theta}}{2h}\right\}=0$$

Then, $g-1+\frac{a}{2}\lambda(g(2isin(\theta))\ \Rightarrow\ g(\frac{a}{2}\lambda(2isin(\theta)))=1\ \Rightarrow\ g(\theta)=\frac{1}{a\lambda isin(\theta)}$. Clearly, $|g(\theta)|\le 1$. Thus, by Theorem, this scheme is stable. (unconditionally)

■

Exercise # 2.3.2

Use the unstable forward-time central-space scheme $(1.3.3)$ for $u_t + u_x = 0$ with the following two sets of initial data on the interval $[-1,3]$ for $0 \le t \le 4$:

(a) $u_0(x) = \begin{cases} 1-|x| & |x| \le 1 \\ 0 & otherwise \end{cases}$

(b) $u_0(x) = \sin x$.

Use a grid spacing of 0.1 and λ equal to 0.8. Demonstrate that the instability is evident sooner with the less smooth initial data (a) than it is for the smooth data (b). Show that the growth in the instability for each case is approximately $|g(\pi/2)|$. For (a) use the boundary condition $u(t,-1) = 0$, and for (b) use the boundary condition $u(t,-1) = -\sin(1+t)$. Use $v_M^{n+1} = v_{M-1}^{n+1}$ at the right boundary.

Solution)

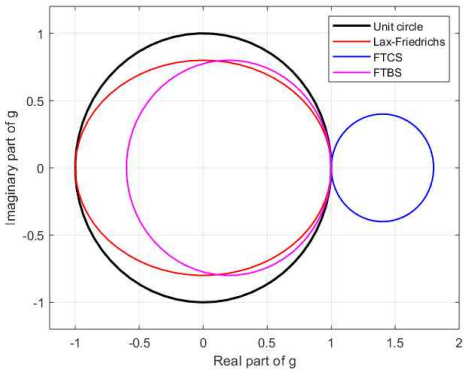


Figure 6. Von neumann stability region for the schemes

Given the initial condition (a) and (b), $\lambda = 0.8$, $a = 1$, $h = 0.1$, the results of less smooth initial condition is

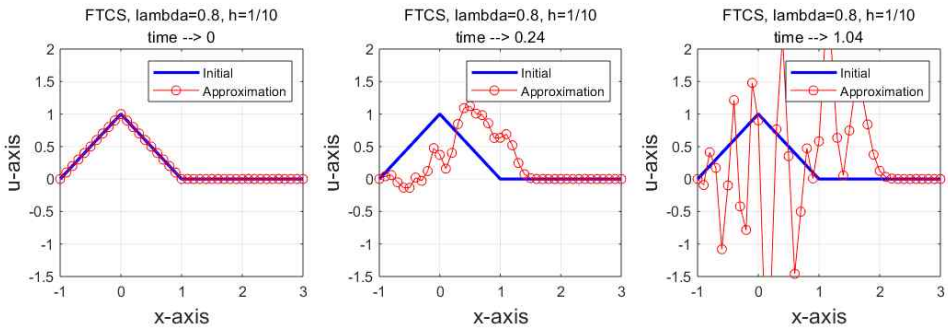


Figure 7. Numerical solution for the less smooth initial condition $u(t,-1) = 0$.

The results of smooth initial condition is

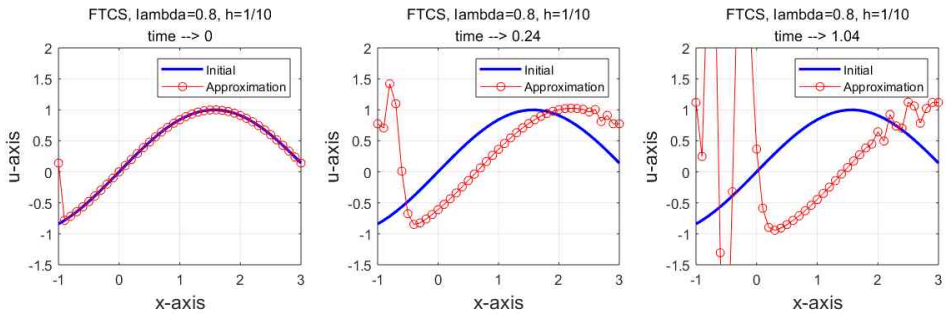


Figure 8. Numerical solution for the smooth initial condition $u(t,-1) = -\sin(1+t)$.

■

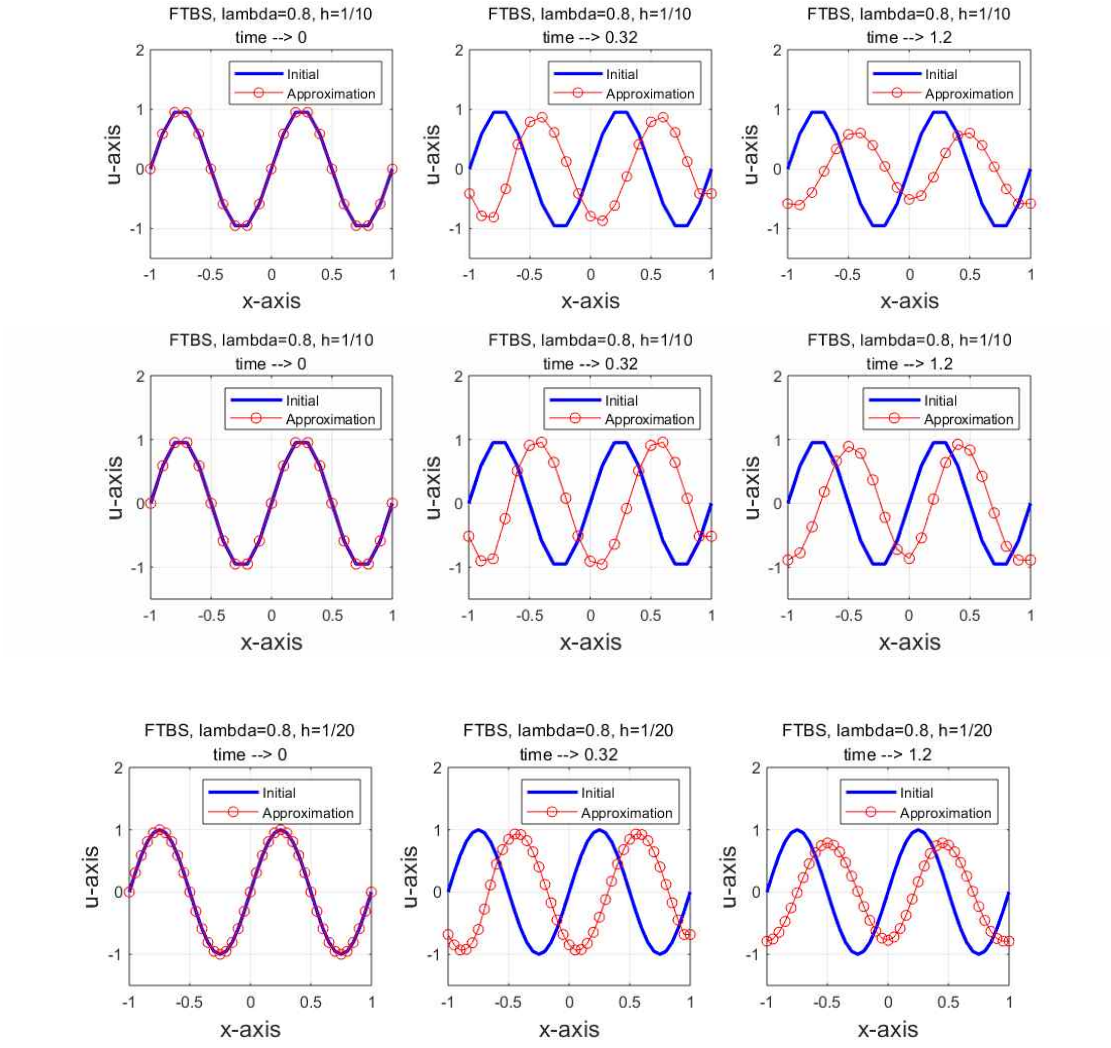
Exercise # 3.1.2

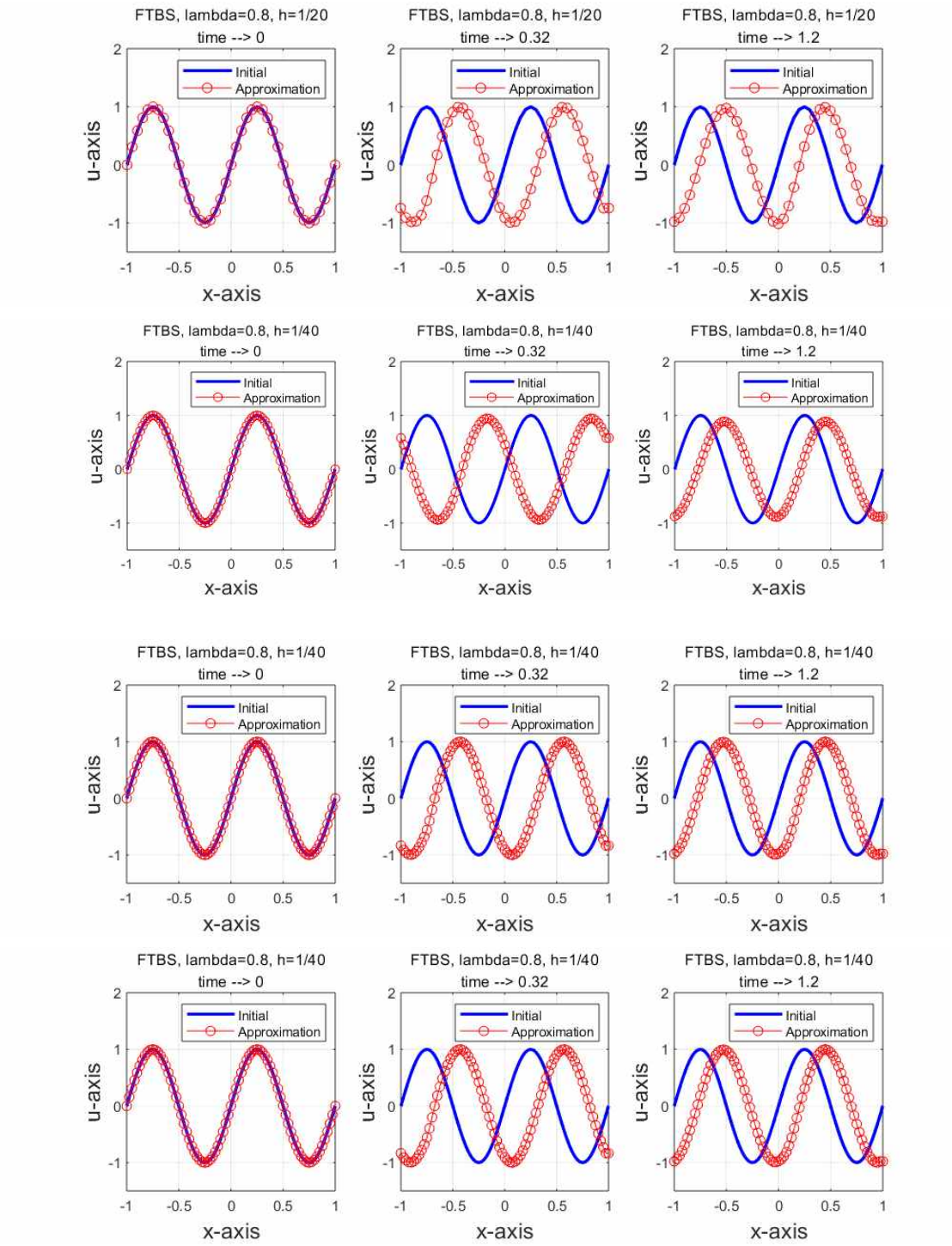
Solve $u_t + u_x = 0, \quad -1 \le x \le 1, \quad 0 \le t \le 1.2$ with $u(0,x) = \sin 2\pi x$ and periodicity, i.e., $u(t,1) = u(t,-1)$. Use two methods:

(a) Forward-time backward-space with $\lambda = 0.8$,
(b) Lax-Wendroff with $\lambda = 0.8$.

Demonstrate the first-order accuracy of the solution of (a) and the second-order accuracy of the solution of (b) using $h = \frac{1}{10}, \frac{1}{20}, \frac{1}{40}$ and $\frac{1}{80}$. Measure the error in the L^2 norm $(3.1.24)$ and the maximum norm. (In the error computation, do not sum both grid points at $x = -1$ and $x = 1$ as separate points.)

Solution)




FTBS, lambda=0.8, h=1/20
time --> 0
FTBS, lambda=0.8, h=1/20
time --> 0.32
FTBS, lambda=0.8, h=1/20
time --> 1.2
FTBS, lambda=0.8, h=1/40
time --> 0
FTBS, lambda=0.8, h=1/40
time --> 0.32
FTBS, lambda=0.8, h=1/40
time --> 1.2
FTBS, lambda=0.8, h=1/40
time --> 0
FTBS, lambda=0.8, h=1/40
time --> 0.32
FTBS, lambda=0.8, h=1/40
time --> 1.2
FTBS, lambda=0.8, h=1/40
time --> 0
FTBS, lambda=0.8, h=1/40
time --> 0.32
FTBS, lambda=0.8, h=1/40
time --> 1.2
Initial
Approximation
u-axis
x-axis

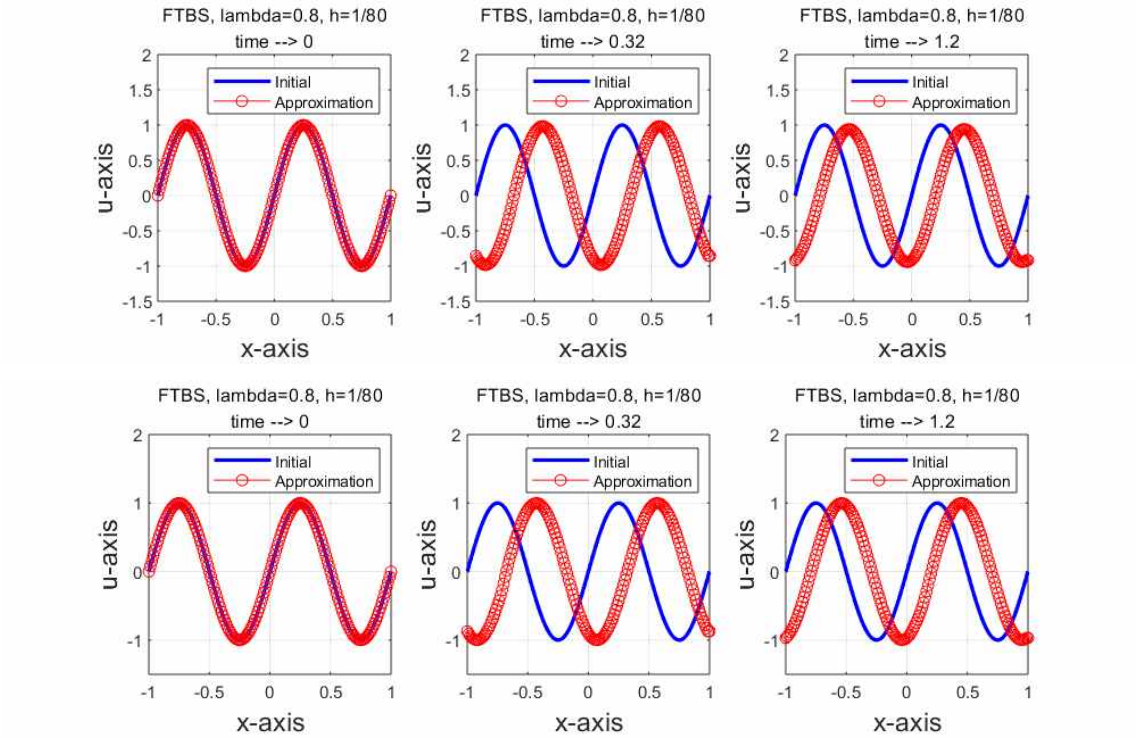


Figure 9. Comparison between two scheme using different space step size.

In the command window (MATLAB), we obtain

	Forward t bacward x		Lax-Wendroff	
h	Error	Order	Error	Order
1/10	3.955e-01		1.731e-01	
1/20	2.158e-01	0.874	4.447e-02	1.960
1/40	1.130e-01	0.934	1.116e-02	1.994
1/80	5.783e-02	0.966	2.792e-03	1.999
1/160	2.926e-02	0.983	6.978e-04	2.000

■

Exercise # 3.2.2

Show that the backward-time central-space scheme

$$\frac{v_m^{n+1}-v_m^n}{k}+a\frac{v_{m+1}^{n+1}-v_{m-1}^{n+1}}{2h}=0$$

is consistent with equation $u_t+au_x=0$ and is unconditionally stable.

Solution)

This problem is the same as exercise # 2.2.1.

Unconditionally stable.

Consider $v_m^n=g^n e^{iw\theta}$, then the scheme is

$$\frac{g^{n+1}e^{im\theta}-g^n e^{im\theta}}{k}+a\frac{g^{n+1}e^{i(m+1)\theta}-g^{n+1}e^{i(m-1)\theta}}{2h}=0 \Rightarrow g^n e^{im\theta}\left\{\frac{g-1}{k}+a\frac{ge^{i\theta}-ge^{-i\theta}}{2h}\right\}=0$$

Then, $g-1+\frac{a}{2}\lambda(g(2isin(\theta)) \Rightarrow g(\frac{a}{2}\lambda(2isin(\theta)))=1 \Rightarrow g(\theta)=\frac{1}{a\lambda isin(\theta)}$. Clearly, $|g(\theta)| \le 1$.

Thus, by Theorem, this scheme is stable. (unconditionally)

■

Exercise # 3.4.1

Solve the initial-boundary value problem $(1.2.1)$

$$u_t + au_x = 0 \text{ with } 0 \le x \le 1,\ t \ge 0.$$

with the leapfrog scheme and the following boundary conditions. Use $a = 1$. Only (d) should give good results. Why?

(a) At $x = 0$, specify $u(t,0)$; at $x = 1$, use boundary condition $(3.4.1b)$.
(b) At $x = 0$, specify $u(t,0)$; at $x = 1$, specify $u(t,1) = 0$.
(c) At $x = 0$, use boundary condition $(3.4.1b)$; at $x = 1$, use $(3.4.1c)$.
(d) At $x = 0$, specify $u(t,0)$; at $x = 1$, use boundary condition $(3.4.1c)$.

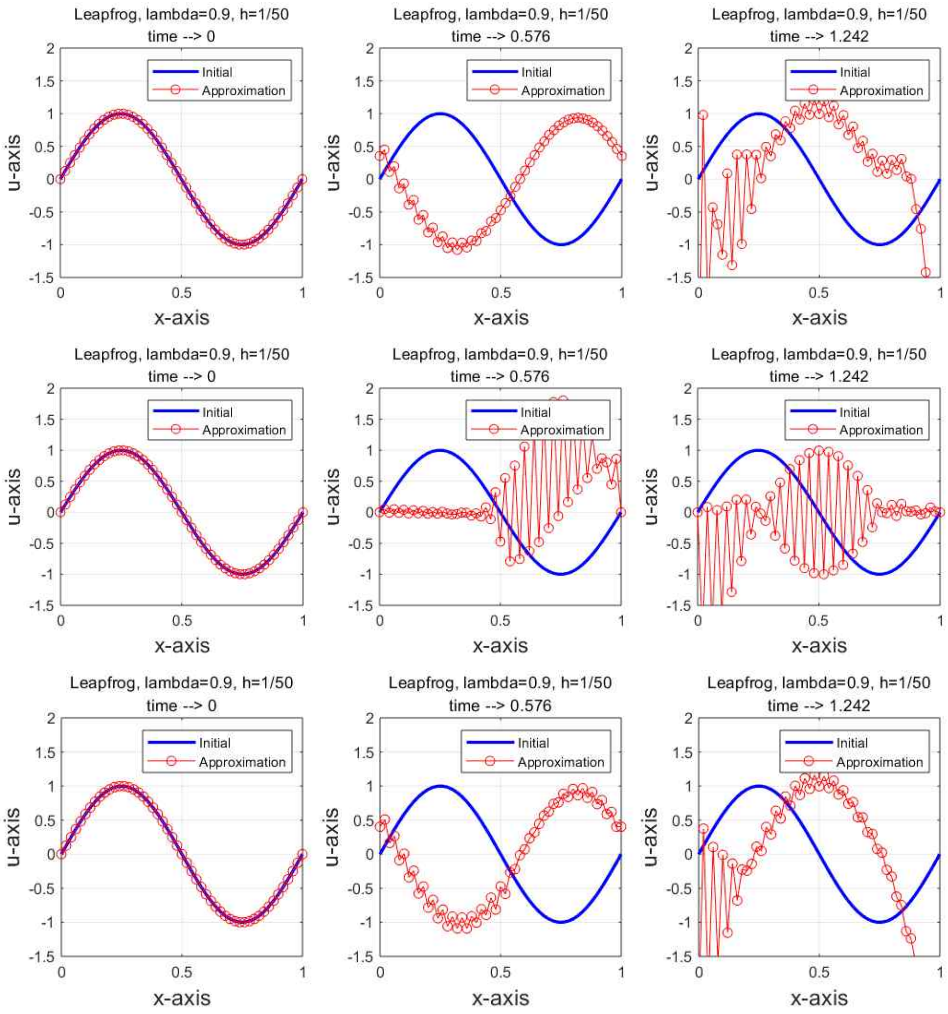
Solution)

Given the numerical boundary conditions in our textbook are

$$v_M^{n+1} = 2v_{M-1}^{n+1} - v_{M-2}^{n+1} \qquad (3.4.1b)$$

$$v_M^{n+1} = v_{M-1}^{n} \qquad (3.4.1c)$$

We take the left boundary condition specify $u(t,0) = 0$ except to specific comments.



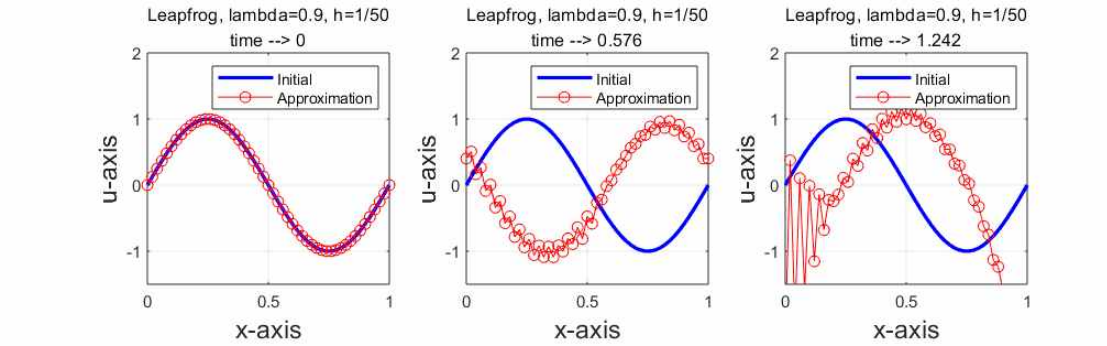


Figure 10. Comparison of the different boundary conditions using leapfrog scheme.

In solving initial-boundary value problems such as (1.2.1) by finite difference schemes, we must use the boundary conditions required by the partial differential equation in order to determine the finite difference solution. Many schemes also require additional boundary conditions, called numerical boundary conditions, to determine the solution uniquely.

The analysis of the stability of a problem involving both initial data and boundary conditions is done by considering the several parts. First, the scheme must be stable for the initial value problem considered on an unbounded domain. This is done with von Neumann analysis. The stability of the boundary conditions is done for each boundary separately. Conditions at one boundary cannot have a significantly ameliorating effect on an unstable boundary condition at the other boundary. As the preceding examples show, a boundary condition may be stable or unstable depending on the scheme with which it is being used.

One final comment should be made on this topic. In solving initial-boundary value problems by finite differences, it is best to distinguish clearly between those boundary conditions required by the partial differential equation and the numerical boundary conditions. By making this distinction, we can avoid solving overdetermined or underdetermined partial differential equation initial-boundary value problems.

** **You (for myself) think you need to see in detail, refer to the textbook (FDS and PDE).**

■

MATLAB CODE FOR THIS HOMEWORK

OOA_2ndgraph.m

```
%% Ch.3 Order of accuracy : 2-dimensional graph

for i=1:n
    % the graph of (x,u) plane for each time t>=0.
    plot(x,u0,'b','LineWidth',2);
    hold on
    plot(x,u(i,:),'r-o');
    hold off
    ylim([-1.5 1.5])
    xlabel('x-axis','FontSize',14)
    ylabel('u-axis','FontSize',14)
    legend('Initial','Approximation')
    title({'The Finite Difference Scheme Approximation of PDE';
        ['time --> ',num2str(t(i))]});
    grid on
    %    F(i)=getframe;
    pause(0.01);
end

%  movie(F)
```

OOA_choose_the_scheme.m

```
%%  Ch.3 Order of accuracy : approximation using FDS formulas

disp('***1.FTFS, 2.FTBS, 3.FTCS, 4.Leapfrog, 5.Lax-Friedrichs, 6.Lax-Wendroff***')
disp('***You do not want to see the graph, enter any number except 1~6.***')
k=input('Enter one of the schemes : 1,2,3,4,5 and 6   ');
switch k
    case 1
        OOA_FTFS
%         figure(k)
         OOA_2ndgraph
    case 2
        OOA_FTBS
%         figure(k)
         OOA_2ndgraph
    case 3
        OOA_FTCS
%         figure(k)
         OOA_2ndgraph
    case 4
        OOA_Leapfrog
%         figure(k)
         OOA_2ndgraph
    case 5
        OOA_LaxFriedrichs
%         figure(k)
         OOA_2ndgraph
    case 6
        OOA_LaxWendroff
%         figure(k)
         OOA_2ndgraph
    otherwise
        disp('You must enter one of the number 1,2,3,4,5 and 6.')
end
```

OOA_compare_schemegraph.m

```
%% Ch.3 Order of accuracy : subplot of 2-dimensional graph

% the graph of (x,u) plane for each time t>=0.
plot(x,u_exact(25,:),'k','LineWidth',2.2);
hold on
plot(x,u2(25,:),'-o');
plot(x,u3(50,:),'-s');
plot(x,u4(25,:),'r-*');
plot(x,u5(25,:),'m-x');
hold off
xlim([0.1 1.8])
ylim([-0.3 1.2])
xlabel('x-axis','FontSize',14)
ylabel('u-axis','FontSize',14)
legend('Exact','FTBS','FTCS','Leapfrog','Lax-Friedrichs')
title({'Finite Difference Schemes, lambda=0.8, h=1/10';
    ['time --> ',num2str(t(25))]});
grid on
```

OOA_exact1.m

```
%% Ch.3 Order of accuracy : exact solution 1 of the equation.
clc
for i=1:n % consider the grid axis for u(:,:)
    for j=1:m % consider the grid axis for u(:,:)
        % exact sol : u(t,x)=u(x-at), a=1. Thus x(i)-t(j).
        if (abs(x(j)-a*t(i))<=1)
            u_e(i,j)=sin(2*pi*(x(j)-a*t(i)));
        else
            u_e(i,j)=0;
        end
    end
end
```

OOA_exact2.m

```
%% Ch.3 Order of accuracy : exact solution 2 of the equation.
clc
for j=1:n % consider the grid axis for u(:,:)
    for i=1:m % consider the grid axis for u(:,:)
         % exact sol : u(t,x)=u(x-at), a=1. Thus x(i)-t(j).
        if (abs(x(i)-a*t(j))<=1/2)
            u(j,i)=1-2*abs(x(i)-a*t(j));
        else
            u(j,i)=0;
        end
    end
end
```

OOA_exact3.m

```
%% Ch.3 Order of accuracy : exact solution 3 of the equation.
clc
for i=1:n % consider the grid axis for u(:,:)
    for j=1:m % consider the grid axis for u(:,:)
        % exact sol : u(t,x)=u(x-at), a=1. Thus x(i)-t(j).
        if (abs(x(j)-a*t(i))<=1/2)
            u(i,j)=cos(pi*(x(j)-a*t(i)))^2;
        else
            u(i,j)=0;
        end
    end
end
```

OOA_exact4.m

```
%% Ch.3 Order of accuracy : exact solution 4 of the equation.
clc
for i=1:n % consider the grid axis for u(:,:)
    for j=1:m % consider the grid axis for u(:,:)
        % exact sol : u(t,x)=u(x-at), a=1. Thus x(i)-t(j).
        if (abs(x(j)-a*t(i))<=1)
            u(i,j)=1-abs(x(j)-a*t(i));
        else
            u(i,j)=0;
        end
    end
end
```

OOA_exact5.m

```
%% Ch.3 Order of accuracy : exact solution 5 of the equation.
clc
for i=1:n % consider the grid axis for u(:,:)
    for j=1:m % consider the grid axis for u(:,:)
        % exact sol : u(t,x)=u(x-at), a=1. Thus x(i)-t(j).
            u(i,j)=sin(x(j)-a*t(i));
    end
end
```

OOA_exact6.m

```
%% Ch.3 Order of accuracy : exact solution 6 of the equation.
clc
for i=1:n % consider the grid axis for u(:,:)
    for j=1:m % consider the grid axis for u(:,:)
        % exact sol : u(t,x)=u(x-at), a=1. Thus x(i)-t(j).
            u(i,j)=sin(2*pi*(x(j)-a*t(i)));
    end
end
```

OOA_FTBS.m

```
%% Ch.3 Order of accuracy : FTBS scheme
clc
for i=1:n-1
    for j=2:m
        % Forward-time backward-space scheme
        u(i+1,j)=u(i,j)-a*lambda*(u(i,j)-u(i,j-1));
    end
    u(i+1,1)=0; % boundary, at x=-1.
    u(i+1,m)=u(i+1,m-1); % boundary, at x=1
    u(:,1)=u(:,m); % periodic condition
end
```

OOA_FTCS.m

```
%% Ch.3 Order of accuracy : FTCS scheme
clc
for i=1:n-1
    for j=2:m-1
        % Forward-time central-space scheme
        u(i+1,j)=u(i,j)-a*lambda/2*(u(i,j+1)-u(i,j-1));
    end
    %     u(i+1,1)=0; % boundary, at x=-1.
    u(i+1,1)=-sin(1+x(j)); % boundary, at x=-1, exercise 2.3.2 (b).
    u(i+1,m)=u(i+1,m-1); % boundary, at x=3
    u(:,1)=u(:,m); % periodic condition
end
```

OOA_FTFS.m

```
%% Ch.3 Order of accuracy : FTFS scheme
clc
for i=1:n-1
    for j=1:m-1
        % Forward-time forward-space scheme
        u(i+1,j)=u(i,j)-a*lambda*(u(i,j+1)-u(i,j));
    end
    u(i+1,1)=0; % boundary, at x=-1.
    u(i+1,m)=u(i+1,m-1); % boundary, at x=1
    u(:,1)=u(:,m); % periodic condition
end
```

OOA_LaxFriedrichs.m

```
%% Ch.3 Order of accuracy : Lax-Friedrichs scheme
clc
for i=1:n-1
    for j=2:m-1
        % Lax-Friedrichs scheme
        u(i+1,j)=0.5*(u(i,j+1)+u(i,j-1))-0.5*a*lambda*(u(i,j+1)-u(i,j-1));
    end
    u(i+1,1)=0; % boundary, at x=-1.
    u(i+1,m)=u(i+1,m-1); % boundary, at x=1
    u(:,1)=u(:,m); % periodic condition
end
```

OOA_LaxWendroff.m

```
%% Ch.3 Order of accuracy : Lax-Wendroff scheme
clc
for i=1:n-1
    for j=2:m-1
        % Lax-Wendroff scheme

u(i+1,j)=u(i,j)-a*lambda/2*(u(i,j+1)-u(i,j-1))+a^2*lambda^2/2*(u(i,j+1)-2*u(i,j)+u(i,j-1
));
    end
    u(i+1,1)=0; % boundary, at x=-1.
    u(i+1,m)=u(i+1,m-1); % boundary, at x=1
    u(:,1)=u(:,m); % periodic condition
end
```

OOA_Leapfrog.m

```
%% Ch.3 Order of accuracy : Leapfrog scheme
clc

for j=2:m
    u(2,j)=u(1,j)-a*lambda/2*(u(1,j)-u(1,j-1));
end
for i=2:n-1
    for j=2:m-1
        % Leapfrog scheme
        u(i+1,j)=u(i-1,j)-a*lambda*(u(i,j+1)-u(i,j-1));
    end
%     u(i+1,1)=0; % boundary, at x=-1.
%     u(i+1,m)=u(i+1,m-1); % boundary, at x=1

    % (a)
%     u(i+1,1)=0; % boundary, at x=-1.
%     u(i+1,m)=2*u(i+1,m-1)-u(i+1,m-2); % boundary, at x=1

    % (b)
%     u(i+1,1)=0; % boundary, at x=-1.
%     u(i+1,m)=0; % boundary, at x=1

    % (c)
%     u(i+1,1)=2*u(i+1,m-1)-u(i+1,m-2); % boundary, at x=-1.
%     u(i+1,m)=u(i+1,m-1); % boundary, at x=1

    % (d)
    u(i+1,1)=0; % boundary, at x=-1.
    u(i+1,m)=u(i+1,m-1); % boundary, at x=1

 u(:,1)=u(:,m); % periodic condition
end
```

OOA_main.m

```
%% Ch.3 Order of accuracy : fundamentals
clc, clear

% parameters setting
global h
% Given h is 1/10, 1/20, 1/40, 1/80, 1/160.
h=1/50; % determine the step size
OOA_parameters

% load the approximations of FDS
OOA_FTFS
u1=u;
OOA_FTBS
u2=u;
OOA_FTCS
u3=u;
OOA_Leapfrog
u4=u;
OOA_LaxFriedrichs
u5=u;
OOA_LaxWendroff
u6=u;

% load the exact solution
OOA_exact6
u_exact=u;

%% Ch.3 Order of accuracy : Calculate the error

% norm of h equals to sqrt(h*sum(abs(u_exact(t,:)-u_approx(t,:)).^2))
% Error_1=@(t,h) sqrt(h)*norm(u_exact(t,:)-u2(t,:),2);
% Error_2=@(t,h) sqrt(h)*norm(u_exact(t,:)-u4(t,:),2);
% Error_3=@(t,h) sqrt(h)*norm(u_exact(t,:)-u6(t,:),2);
% Error_4=@(t,h) sqrt(h)*norm(u_exact(t,:)-u5(t,:),2);

% Error_FTCS=@(t,h) sqrt(h)*norm(u_exact(t,:)-u3(t,:),2);

% find the time = 5.4
% temp=find(t < 3.3);
% t_index=temp(length(temp))+1;

% test
% FTBS_error=Error_1(t_index,h)
% Leapfrog_error=Error_2(t_index,h)
% Lax_Wendroff_error=Error_3(t_index,h)
% Lax_Friedrichs_error=Error_4(t_index,h)

% Error_FTCS(t_index,h)
```

```
% 2-dimensional graph
% You must enter all of the number 1,2,3,4, 5 and 6 respectively.
% 1.FTFS, 2.FTBS, 3.FTCS, 4.Leapfrog, 5.Lax-Friedrichs, 6.Lax-Wendroff
OOA_choose_the_scheme

% mesh graph of the error
% OOA_meshgraph
```

OOA_meshgraph.m

```
% Error computations mesh graph
figure(2)
subplot(2,3,1)
Err_1=abs(u_exact-u1); %FTFS
mesh(x,t,Err_1)
title('FTFS')
subplot(2,3,2)
Err_2=abs(u_exact-u2); %FTBS
mesh(x,t,Err_2)
title('FTBS')
subplot(2,3,3)
Err_3=abs(u_exact-u3); %FTCS
mesh(x,t,Err_3)
title('FTCS')
subplot(2,3,4)
Err_4=abs(u_exact-u4); %Leapfrog
mesh(x,t,Err_4)
title('Leapfrog')
subplot(2,3,5)
Err_5=abs(u_exact-u5); %Lax-Friedrichs
mesh(x,t,Err_5)
title('Lax-Friedrichs')
subplot(2,3,6)
mesh(x,t,u_exact)
title('One-way wave equation') % Exact solution
```

OOA_parameters.m

```
%% Ch.3 Order of accuracy : parameters setting
% Let the speed of propagetion a is 1.
% Given PDE is the one-way wave equation.

% Define the Variables we need
a=1;
global h
%h=1/10; % step size of x, i.e. h=dx
lambda=0.9; % lambda=k/h
k=lambda*h; % k=dt
t=0:k:1.5; % initial value of t is 0.
x=0:h:1; % boundary value of u, u(t,-1)=0 & u(t,3)=0
n=length(t);
m=length(x);
u=zeros(n,m); % finite difference grid
u0=zeros(1,m); % initial frame of grid

% Initial-Boundary Problem Function u0

% initial 1
% for i=1:m
%  if (abs(x(i))<=1)
%      u0(i)= sin(2*pi*x(i)); % u(0,x)=u0(x)
%  else
%      u0(i)=0;
%  end
% end
% u(1,:)=u0; % Initial value setting

% initial 2
% for i=1:m % consider the grid axis for u(:,:)
%     if (abs(x(i))<=1/2)
%         u0(i)=1-2*abs(x(i));
%     else
%         u0(i)=0;
%     end
% end
% u(1,:)=u0; % Initial value setting

% % initial 3
% for i=1:m
%  if (abs(x(i))<=1/2)
%      u0(i)= cos(pi*x(i))^2; % u(0,x)=u0(x)
%  else
%      u0(i)=0;
%  end
% end
% u(1,:)=u0; % Initial value setting
```

```
% initial 4
% for i=1:m
%  if (abs(x(i))<=1)
%     u0(i)= 1-abs(x(i)); % u(0,x)=u0(x)
%  else
%     u0(i)=0;
%  end
% end
% u(1,:)=u0; % Initial value setting

% initial 5
% for i=1:m % consider the grid axis for u(:,:)
%        % exact sol : u(t,x)=u(x-at), a=1. Thus x(i)-t(j).
%           u0(i)=sin(x(i));
% end
% u(1,:)=u0; % Initial value setting

% initial 6
for i=1:m % consider the grid axis for u(:,:)
       % exact sol : u(t,x)=u(x-at), a=1. Thus x(i)-t(j).
          u0(i)=sin(2*pi*x(i));
end
u(1,:)=u0; % Initial value setting
```

OOA_schemes.m

```
%% Ch.3 Order of accuracy : finite difference schemes
clc
k=input('Enter one of the schemes :
1(FTFS),2(FTBS),3(LeapFrog),4(Lax-F),5(Lax-W),6(FTCS)');
for i=2:n-1
    for j=2:m-1
        switch k
            case 1
                % Forward-time forward-space scheme
                 u(i+1,j)=u(i,j)-a*lambda*(u(i,j+1)-u(i,j));
            case 2
                % Forward-time backward-space scheme
                 u(i+1,j)=u(i,j)-a*lambda*(u(i,j)-u(i,j-1));
            case 3
                 % Leapfrog scheme
                 u(i+1,j)=u(i-1,j)-a*lambda*(u(i,j+1)-u(i,j-1));
            case 4
                 % Lax-Friedrichs scheme
                 u(i+1,j)=0.5*(u(i,j+1)+u(i,j-1))-0.5*a*lambda*(u(i,j+1)-u(i,j-1));
            case 5
                % Lax-Wendroff scheme

u(i+1,j)=u(i,j)-a*lambda/2*(u(i,j+1)-u(i,j-1))+a^2*lambda^2/2*(u(i,j+1)-2*u(i,j)+u(i,j-1
));
            case 6
                 % Forward-time central-space scheme
                 u(i+1,j)=u(i,j)-a*lambda/2*(u(i,j)-u(i,j-1));
            otherwise
                disp('You must put one of the number 1, 2, 3, 4, 5 and 6.')
        end
    end
%       u(i+1,1)=0; % boundary, at x=-1.
%       u(i+1,m)=u(i+1,m-1); % boundary, at x=1
     u(:,1)=u(:,m); % periodic condition
end
```

OOA_subplot_2ndgraph.m

```
%% Ch.3 Order of accuracy : subplot of 2-dimensional graph

% the graph of (x,u) plane for each time t>=0.
subplot(1,3,1)
plot(x,u0,'b','LineWidth',2);
hold on
plot(x,u(1,:),'r-o');

ylim([-1.5 2])
xlabel('x-axis','FontSize',14)
ylabel('u-axis','FontSize',14)
legend('Initial','Approximation')
title({'Leapfrog, lambda=0.9, h=1/50';
    ['time --> ',num2str(t(1))]});
grid on
subplot(1,3,2)
plot(x,u0,'b','LineWidth',2);
hold on
plot(x,u(33,:),'r-o');

ylim([-1.5 2])
xlabel('x-axis','FontSize',14)
ylabel('u-axis','FontSize',14)
legend('Initial','Approximation')
title({'Leapfrog, lambda=0.9, h=1/50';
    ['time --> ',num2str(t(33))]});
grid on
subplot(1,3,3)
plot(x,u0,'b','LineWidth',2);
hold on
plot(x,u(70,:),'r-o');

ylim([-1.5 2])
xlabel('x-axis','FontSize',14)
ylabel('u-axis','FontSize',14)
legend('Initial','Approximation')
title({'Leapfrog, lambda=0.9, h=1/50';
    ['time --> ',num2str(t(70))]});
grid on
```

OOA_Von_neumann.m

```
%% Stability region for the schemes
clc
% Set the functions g & the variables
g1=@(x) cos(x)-sqrt(-1)*a*lambda*sin(x); % Lax
g2=@(x) 1-1/2*a*lambda*(cos(x)+sqrt(-1)*sin(x)-1); % FTCS
g3=@(x) 1-a*lambda+a*lambda*cos(x)-sqrt(-1)*a*lambda*sin(x); % FTBS
x=-1:0.1:7;
a=1;
lambda=0.8;

% the norm of g2(pi/2)
norm(g2(pi/2))

% Graph of the Von neumann stability region (amplication factor).
unit=cos(x)+sqrt(-1)*sin(x);
plot(real(unit),imag(unit),'k','LineWidth',1.8)
xlabel('Real part of g')
ylabel('Imaginary part of g')
grid on
xlim([-1.2 2])
ylim([-1.2 1.2])
hold on
plot(real(g1(x)),imag(g1(x)),'r','LineWidth',1.2)
plot(real(g2(x)),imag(g2(x)),'b','LineWidth',1.2)
plot(real(g3(x)),imag(g3(x)),'m','LineWidth',1.2)
legend('Unit circle','Lax-Friedrichs','FTCS','FTBS')
```

Table_3_1_1.m

```
clc
clear
close all
method=1;
a=1;
h_vec=[1/10 1/20 1/40 1/80 1/160];
Nh=length(h_vec);
error=zeros(Nh,4);
for hsize=1:Nh
    h=h_vec(hsize);
    lambda=0.9;
    Ft=5.4; % final time
    k=h*lambda;
    x=-1:h:1;
    t=0:k:Ft;
    Nx=length(x);
    Nt=length(t);

    %initial condition
    u0=zeros(1,Nx);
    for i=1:Nx
        if x(i)>=-1 && x(i)<=1
            u0(i)=1-sin(2*pi*x(i));
        end
    end

    u_ex=zeros(Nt,Nx);
    for j=1:Nt
        for i=1:Nx
            u_ex(j,i)=1-sin(2*pi*(x(i)-a*t(j)));
        end
    end

    u2=zeros(Nt,Nx); % forward-time backward-space
    u4=zeros(Nt,Nx); % leapfrog

    u2(1,:)=u0(:);
    u4(1,:)=u0(:);
    for j=1:Nt-1 % forward-time backward-space
        for i=1:Nx
            if i==1
                u2(j+1,i)=u2(j,i)-a*lambda*(u2(j,i)-u2(j,Nx-1));
            else
                u2(j+1,i)=u2(j,i)-a*lambda*(u2(j,i)-u2(j,i-1));
            end
        end
    end

    for j=1:Nt-1 % leapfrog
```

```
        for i=1:Nx
            if j==1
                if i==1 % first time step for leapfrog (FTCS)
                    u4(j+1,i)=u4(j,i)-1/2*a*lambda*(u4(j,i+1)-u4(j,Nx-1));
                elseif i==Nx
                    u4(j+1,i)=u4(j,i)-1/2*a*lambda*(u4(j,2)-u4(j,i-1));
                else
                    u4(j+1,i)=u4(j,i)-1/2*a*lambda*(u4(j,i+1)-u4(j,i-1));
                end
            else
                if i==1
                    u4(j+1,i)=u4(j-1,i)-a*lambda*(u4(j,i+1)-u4(j,Nx-1));
                elseif i==Nx
                    u4(j+1,i)=u4(j-1,i)-a*lambda*(u4(j,2)-u4(j,i-1));
                else
                    u4(j+1,i)=u4(j-1,i)-a*lambda*(u4(j,i+1)-u4(j,i-1));
                end
            end
        end
    end
    error(hsize,1)=sqrt(h*sum((u2(end,:)-u_ex(end,:)).^2));
    error(hsize,3)=sqrt(h*sum((u4(end,:)-u_ex(end,:)).^2));
     if hsize~=1
         error(hsize,2)=log(error(hsize-1,1)/error(hsize,1))/log(2);
         error(hsize,4)=log(error(hsize-1,3)/error(hsize,3))/log(2);
     end
end

%% error table
h_string={'1/10 ', '1/20 ','1/40 ','1/80 ','1/160'};

fprintf('-------------------------------------------------------\n')
fprintf('       | Forward t bacward x  |   Leapfrog scheme    |\n')
fprintf('-------------------------------------------------------\n')
fprintf('    h   |    Error   | Order  |    Error   | Order  |\n')
fprintf('-------------------------------------------------------\n')
for i=1:Nh
    if i==1
        fprintf(' %s  | %1.3e  |        | %1.3e  |        |\n',...
            h_string{i},error(i,1),error(i,3));
    else
        fprintf(' %s  | %1.3e  | %1.3f  | %1.3e  | %1.3f  |\n',...
          h_string{i},error(i,1),error(i,2),error(i,3),error(i,4));
    end
end
fprintf('-------------------------------------------------------\n')
```

Table_3_1_2.m

```
clc
clear
close all
method=1;
a=1;
h_vec=[1/10 1/20 1/40 1/80 1/160];
Nh=length(h_vec);
error=zeros(Nh,4);
for hsize=1:Nh
    h=h_vec(hsize);
    lambda=0.9;
    Ft=5.4; % final time
    k=h*lambda;
    x=-1:h:6;
    t=0:k:Ft;
    Nx=length(x);
    Nt=length(t);

    %initial condition
    u0=zeros(1,Nx);
    for i=1:Nx
        if x(i)>=-1/2 && x(i)<=1/2
            u0(i)=1-2*abs(x(i));
        end
    end

    u_ex=zeros(Nt,Nx);
    for j=1:Nt
        for i=1:Nx
            if x(i)>=-1/2+a*t(j) && x(i)<=1/2+a*t(j)
            u_ex(j,i)=1-2*abs(x(i)-a*t(j));
            end
        end
    end


    u2=zeros(Nt,Nx); % Lax-Wendroff
    u4=zeros(Nt,Nx); % Lax-Friderichs

    u2(1,:)=u0(:);
    u4(1,:)=u0(:);

    for j=1:Nt-1
        for i=1:Nx
            if i==1

u2(j+1,i)=u2(j,i)-a*lambda/2*(u2(j,i+1)-u2(j,Nx-1))+a^2*lambda^2/2*(u2(j,i+1)-2*u2(j,i)+
u2(j,Nx-1));
              u4(j+1,i)=1/2*(u4(j,i+1)+u4(j,Nx-1))-1/2*a*lambda*(u4(j,i+1)-u4(j,Nx-1));
```

```
            elseif i==Nx

u2(j+1,i)=u2(j,i)-a*lambda/2*(u2(j,2)-u2(j,i-1))+a^2*lambda^2/2*(u2(j,2)-2*u2(j,i)+u2(j,
i-1));
                u4(j+1,i)=1/2*(u4(j,2)+u4(j,i-1))-1/2*a*lambda*(u4(j,2)-u4(j,i-1));
            else

u2(j+1,i)=u2(j,i)-a*lambda/2*(u2(j,i+1)-u2(j,i-1))+a^2*lambda^2/2*(u2(j,i+1)-2*u2(j,i)+u
2(j,i-1));
                u4(j+1,i)=1/2*(u4(j,i+1)+u4(j,i-1))-1/2*a*lambda*(u4(j,i+1)-u4(j,i-1));
            end
        end
    end
    error(hsize,1)=sqrt(h*sum((u2(end,:)-u_ex(end,:)).^2));
    error(hsize,3)=sqrt(h*sum((u4(end,:)-u_ex(end,:)).^2));
    if hsize~=1
        error(hsize,2)=log(error(hsize-1,1)/error(hsize,1))/log(2);
        error(hsize,4)=log(error(hsize-1,3)/error(hsize,3))/log(2);
    end
end

%% error table
h_string={'1/10 ', '1/20 ','1/40 ','1/80 ','1/160'};

fprintf('-------------------------------------------------------\n')
fprintf('       |    Lax-Wendroff    |   Lax-Friedrichs    |\n')
fprintf('-------------------------------------------------------\n')
fprintf('    h   |   Error   | Order |   Error   | Order  |\n')
fprintf('-------------------------------------------------------\n')
for i=1:Nh
    if i==1
        fprintf(' %s  | %1.3e  |        | %1.3e  |        |\n',...
            h_string{i},error(i,1),error(i,3));
    else
        fprintf(' %s  | %1.3e  | %1.3f  | %1.3e  | %1.3f  |\n',...
          h_string{i},error(i,1),error(i,2),error(i,3),error(i,4));
    end
end
fprintf('-------------------------------------------------------\n')
```

Implicit_BTBS.m

```
clc

% BTBS
a=1; h=1/30; lambda=1; k=lambda*h; t=0:k:1; x=-2:h:3;

C1=-a*lambda; C2=1+a*lambda;
n=length(t); m=length(x);

u1(n,m)=0;
for i=1:length(x)
    if (1-abs(x(i))>0)
        u0(i)=1-abs(x(i));
    else
        u0(i)=0;
    end
end
u1(1,:)=u0;
A=eye(m);
A=A*C2;
tmp=C1*ones(m-1,1);
lowA=diag(tmp,-1);
A=A+lowA;
for i = 1:n-1
    u1(i+1,:)=A\u1(i,:)';
end
% mesh(x,t,u)
```

Implicit_CN.m

```
 clc

% Crank-Nicolson scheme
a=1; h=1/30; lambda=1; k=lambda*h; t=0:k:1; x=-2:h:3;

C1=-1/4*a*lambda; C2=1/4*a*lambda;
n=length(t); m=length(x);
u2(n,m)=0;

for i=1:length(x)
    if (1-abs(x(i))>0)
        u0(i)=1-abs(x(i));
    else
        u0(i)=0;
    end
end
u2(1,:)=u0;
A=eye(m);
tmp1=C2*ones(m-1,1);
tmp2=C1*ones(m-1,1);
```

```
lowA=diag(tmp2,-1);
uppA=diag(tmp1,1);
A=A+lowA+uppA;

b=zeros(1,m);
for i = 1:n-1
    b(1)=(1+a/4*lambda)*u2(i,1)-a/4*lambda*u2(i,2);
    for j=2:m-1
        b(j)=a/4*lambda*u2(i,j-1)+u2(i,j)-a/4*lambda*u2(i,j+1);
    end
    b(m)=(1-a/4*lambda)*u2(i,m)+a/4*lambda*u2(i,m-1);
    u2(i+1,:)=A\b';
end

% mesh(x,t,u)
```

Compare_two_Implicit_scheme.m

```
clear, clc

Implicit_BTBS
Implicit_CN

% exact solution 1
for i=1:n
    for j=1:m
        if (1-abs(x(j)-a*t(i))>0)
            u_e1(i,j)=1-abs(x(j)-a*t(i));
        else
            u_e1(i,j)=0;
        end
    end
end

% plots
figure(1)
plot(x,u_e1(n,:),'k','LineWidth',1.2);
ylim([-0.5 1.5])
grid on
hold on
plot(x,u1(n,:),'r-o',x,u2(n,:),'g-o');
legend('Exact','FTBS','Crank-Nicolson')
hold off
```