淺水波之計算方法-作業2

[redacted]

1 成果及解析

首先介紹線性淺水波方程式：

$$\begin{cases} \frac{\partial \eta}{\partial t} + \frac{\partial hU}{\partial x} = 0 \\ \frac{\partial U}{\partial t} = -g\frac{\partial \eta}{\partial x} \end{cases} \tag{1}$$

其中，η為自由液面的起伏，h為靜水深度，U為水平方向平均速度，g為重力加速度。本次作業主要以這個方程組在空間上及時間上進行離散化，代入初始條件計算$t=0$的η和U，再由Lax-Friedrichs方法求得他們的後續發展。以下依照我打的程式邏輯，敘述各個步驟使用的方程式。

1. 首先由題目提供的初始條件，得到:

$$\begin{cases} \eta(x,0) = Hsech^2(K(x - C\cdot 0)), K = \frac{1}{h}\sqrt{\frac{3H}{4h}} \\ U(x,0) = \frac{\eta(x,0)}{h}\sqrt{gh} \end{cases} \tag{2}$$

   其中，H為波高，C為波速（大約等於$\sqrt{gh}$），K為有效波數。

2. 接著，將(1)經由Lax-Friedrichs方法，得到離散化的方程式：

$$\begin{cases} \eta_i^{(n+1)} = \frac{\eta_{i+1}^{(n)} + \eta_{i-1}^{(n)}}{2} - \frac{\Delta t}{2\Delta x}[U_{i+1}^{(n)}h_{i+1} - U_{i-1}^{(n)}h_{i-1}] \\ U_i^{(n+1)} = \frac{U_{i+1}^{(n)} + U_{i-1}^{(n)}}{2} - \frac{\Delta t}{2\Delta x}g[\eta_{i+1}^{(n)} - \eta_{i-1}^{(n)}] \end{cases} \tag{3}$$

   其中，(n)代表時間點，i代表空間上的點，Δx代表每個點的間距，Δt代表每次測量的間隔時間。

   Δx與Δt之間的關係如下：

$$\Delta t = C_{CFL}\frac{\Delta x}{\sqrt{gh_0}} \tag{4}$$

   其中，C_{CFL}為$Courant\ number$，h_0為最大水深。

3. 最後，使用邊界條件求兩端的η和U。這次使用的是牆的邊界條件，也就是在兩端的U必須為0，U對t的微分也必須是0。因此，我們得到以下兩式：

$$\begin{cases} \eta_{i=0} = \eta_{i=2} \\ U_{i=0} = 0 \end{cases} \tag{5}$$

   同理可推得另外一端的η和U。

下一頁開始為我跑完程式得到的結果。

1.1 畫出初始條件

由(2)畫出初始的η和U，結果如Figure 1。（我使用第二題x的範圍，也就是$-12 \leq x \leq 24$(m)。）

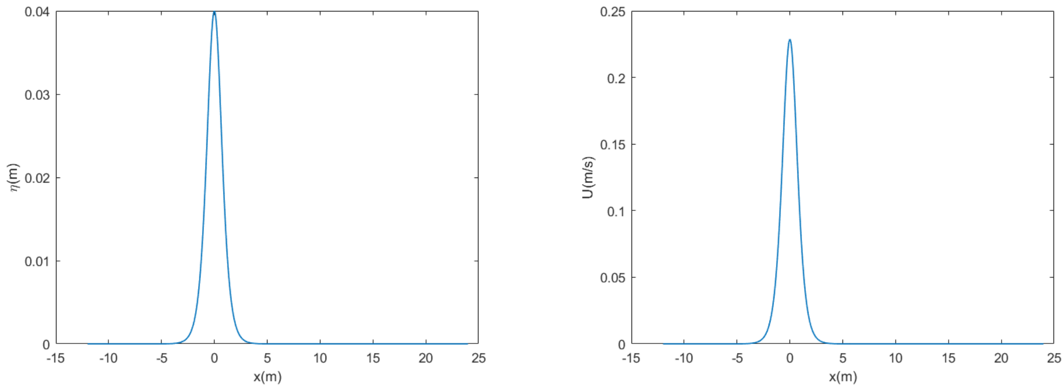


Figure 1: $t = 0$ 時的η及U

1.2 解出淺水波方程式

用Matlab解出$0 \leq t \leq 6.95$(s)時的η及U。我使用的Δx為0.06m。程式碼如附錄。

1.3 與解析解進行比較

將最後一個時間點的η與解析解的η畫在同一張圖上，結果如Figure 2。由於上題使用的Δx為0.06m，經由(4)得到Δt為0.0315s，因此這裡畫的時間點為$t = 6.9250$s。從Figure 2可以看出，做出來的結果和解析解仍然有一段差距。

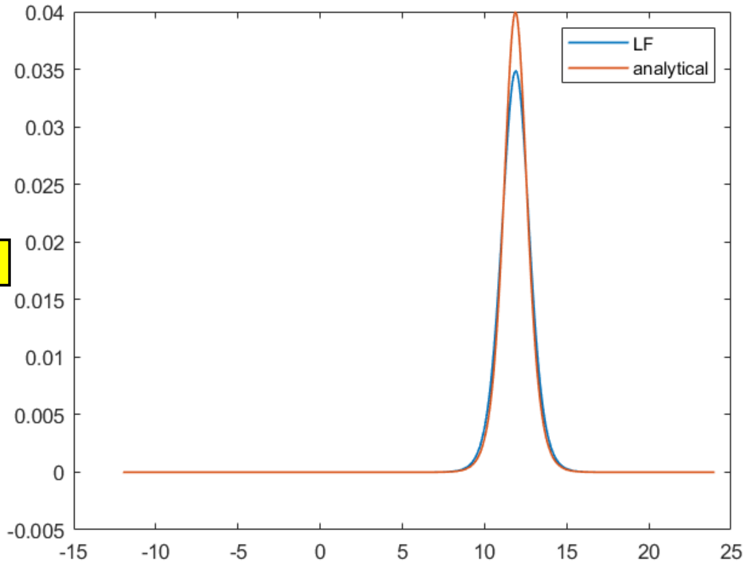


Figure 2: 數值解及解析解

1.4 使用其他Δx進行比較

我使用Δx=0.03m及Δx=0.12m來進行比較，兩者分別是原Δx的0.5倍及2倍。結果如Figure 3，可以看出取的Δx愈小，數值解愈接近解析解。

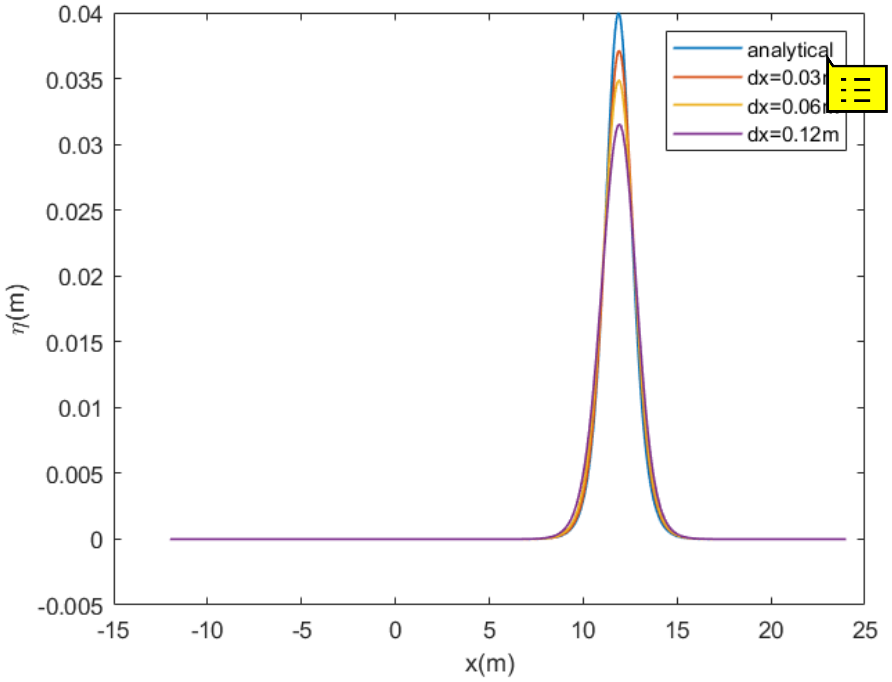
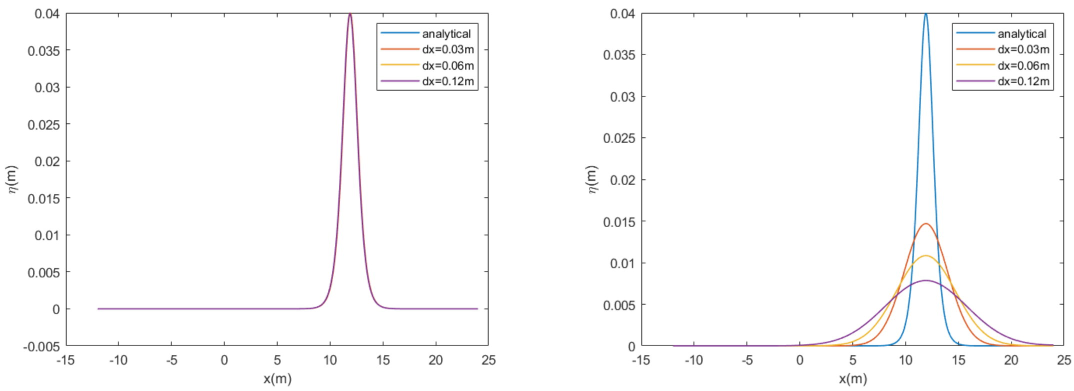


Figure 3: 不同Δx的數值解與解析解

1.4.1 試試其他C_{CFL}

因為老師講義有提到，所以我就試試看將C_{CFL}分別代入1與0.1，結果如Figure (4(a))和Figure (4(b))。C_{CFL}=1時，代不同Δx跑出來的數值解都和解析解一模一樣，但C_{CFL}=0.1時就差得非常多。



(a) C_{CFL}=1

(b) C_{CFL}=0.1

Figure 4: 不同C_{CFL}下的η

1.5 畫出L^2-norm

L^2-norm代表數值解與解析解之間的誤差，其定義如下：

$$L^2 - norm = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\eta_{i,num} - \eta_{i,true})^2} \tag{6}$$

其中，N為離散化後空間上點的數量，$\eta_{i,num}$為第i個點用數值方法算出的值，$\eta_{i,true}$為第i個點代入解析解的值。對於此作業使用的Lax-Friedrichs方法，Δx應正比於L^2-norm。我用前幾題的數據進行畫圖，結果如Figure (5)。由圖可知，Δx大約正比於L^2-norm，也就代表此數值方法為線性收斂。

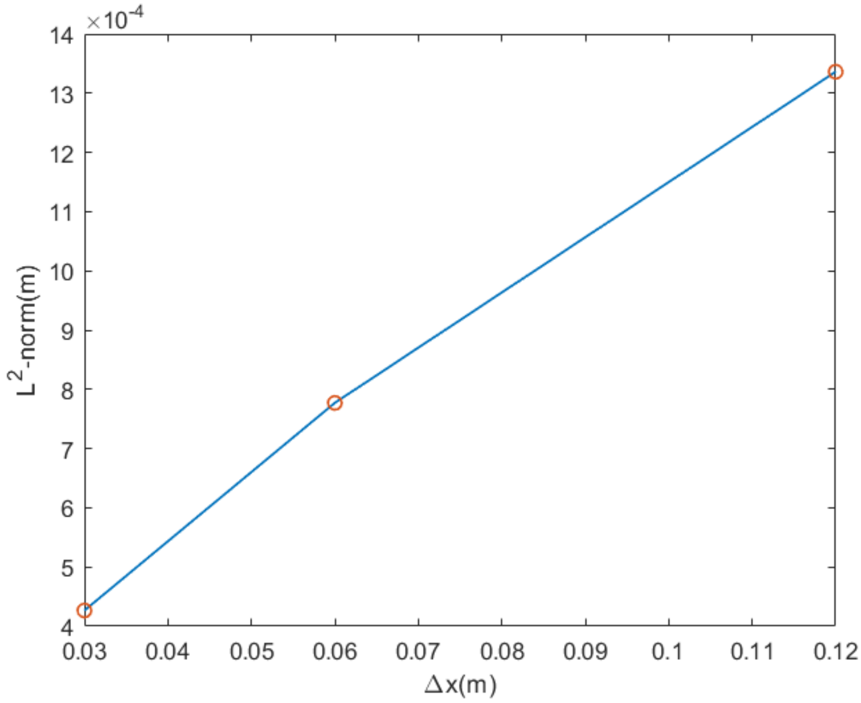


Figure 5: Δx對L^2-norm進行作圖

1.6 心得

第一次打這種分析結果的報告，如果哪裡講的不夠，希望老師可以告訴我。這也是我第一次用LaTeX打報告，格式已經盡量照老師講義打得一樣了，但是方程式感覺有點擠就是了。還有就是希望老師下次作業可以給多一點時間，不然一個禮拜要打code和打報告有點趕（自己研究LaTeX研究到有點崩潰）。

2 附錄：程式碼

```matlab
%% Assignment#2
[eta1,u1,x1,t1]=LF(0.06);
%% plot initial conditions
figure(1);
plot(x1,eta1(1,2:end-1),'LineWidth',1);
xlabel('x(m)');
ylabel('\eta(m)');
set(gca,'YTick',0:0.01:0.04);

figure(2);
plot(x1,u1(1,2:end-1),'LineWidth',1);
xlabel('x(m)');
ylabel('U(m/s)');
%% compare result with analytical solution and different step size
h=0.3;
H=0.04;
g=9.81;
K=sqrt(3*H/4/h)/h;
C=sqrt(g*h);
etaa=zeros(1,size(x1,2));
for i=1:size(x1,2)
    etaa(1,i)=H*(sech(K*(x1(1,i)-C*t1(1,end)))).^2;
end
[eta2,u2,x2,t2]=LF(0.03);
[eta3,u3,x3,t3]=LF(0.12);
figure(3);
plot(x1,etaa,'LineWidth',1);
hold on
plot(x2,eta2(end,2:end-1),'LineWidth',1);
plot(x1,eta1(end,2:end-1),'LineWidth',1);
plot(x3,eta3(end,2:end-1),'LineWidth',1);
hold off
legend('analytical','dx=0.03m','dx=0.06m','dx=0.12m');
xlabel('x(m)');
ylabel('\eta(m)');
%% L2-norm
L1=L2norm(eta1,x1,t1);
L2=L2norm(eta2,x2,t2);
L3=L2norm(eta3,x3,t3);
dx=[0.03,0.06,0.12];
L=[L2,L1,L3];
figure(4);
plot(dx,L,'LineWidth',1);
hold on
plot(dx,L,'o','LineWidth',1);
hold off
xlabel('\Deltax(m)');
ylabel('L^{2}-norm(m)');

function [eta,u,x,t]=LF(dx)
%% set parameters
h=0.3;
H=0.04;
cfl=0.9;
g=9.81;
```

```
K=sqrt(3*H/4/h)/h;
C=sqrt(g*h);
%% construct matrix
dt=dx*cfl/sqrt(g*h);
x=-12:dx:24;
t=0:dt:floor(6.95/dt)*dt;
nx=(24-(-12))/dx+1;
nt=floor(6.95/dt)+1;
eta=zeros(nt,nx+2);
u=zeros(nt,nx+2);
%% initial conditions
for i=1:nx
    eta(1,i+1)=H*(sech(K*(x(1,i)-C*t(1,1)))).^2;
    u(1,i+1)=eta(1,i+1)/h*sqrt(g*h);
end
%% boundary conditions
eta(1,1)=eta(1,3);
eta(1,nx+2)=eta(1,nx);
%% L-F method
for n=1:nt-1
    for i=1:nx
        eta(n+1,i+1)=(eta(n,i+2)+eta(n,i))/2-dt/(2*dx)*(u(n,i+2)*h-u(n,i)*h);
        u(n+1,i+1)=(u(n,i+2)+u(n,i))/2-dt/(2*dx)*g*(eta(n,i+2)-eta(n,i));
    end
    eta(n+1,1)=eta(n+1,3);
    eta(n+1,nx+2)=eta(n+1,nx);
end
end

function LL=L2norm(eta,x,t)
%% set parameters
h=0.3;
H=0.04;
g=9.81;
K=sqrt(3*H/4/h)/h;
C=sqrt(g*h);
%% calculate eta-true and L^2-norm
etat=zeros(1,size(x,2));
sum=0;
for i=1:size(x,2)
    etat(i)=H*(sech(K*(x(1,i)-C*t(1,end)))).^2;
    sum=sum+(eta(size(t,2),i+1)-etat(i)).^2;
end
LL=sqrt(sum/size(x,2));
end
```