MTH 656/659 Computational Wave Propagation Assignment 2

Sandeep Koranne

May 7, 2017

1 Dispersion relation for the homogenous wave equation in 1D

Consider the wave equation given by:

$$u_{tt} - c^2 \Delta u = 0$$

and its numerical solution using the explicit Leapfrog scheme:

$$U_j^{n+1} = 2U_j^n - U_j^{n-1} + \nu^2 (U_{j+1}^n - 2U_j^n + U_{j-1}^n)$$

which is conditionally stable under the CFL condition $\nu = \frac{c\Delta t}{\Delta x} \leq 1$.

1.1 Compute the numerical dispersion relation of the above scheme

Proof. We can write the numerical scheme as:

$$\frac{U_j^{n+1} - 2U_j^n + U_j^{n-1}}{\Delta t^2} - c^2 \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{h^2} = 0$$

Consider a plane wave solution written as

$$u_j^n = e^{i(n\omega\Delta t - jkh)}$$

Substituting above in the numerical scheme we get:

$$\begin{aligned}
\frac{e^{i((n+1)\omega\Delta t - jkh)} - 2e^{i(n\omega\Delta t - jkh)} + e^{i((n-1)\omega\Delta t - jkh)}}{\Delta t^2} &= c^2 \frac{e^{i(n\omega\Delta t - (j+1)kh)} - 2e^{i(n\omega\Delta t - jkh)} + e^{i(n\omega\Delta t - (j-1)kh)}}{h^2} \\
e^{i(n\omega\Delta t - jkh)} \frac{e^{i\omega\Delta t} - 2 + e^{-i\omega\Delta t}}{\Delta t^2} &= e^{i(n\omega\Delta t - jkh)} c^2 \frac{e^{ikh} - 2 + e^{-ikh}}{h^2} \\
\frac{e^{i\omega\Delta t} - 2 + e^{-i\omega\Delta t}}{\Delta t^2} &= c^2 \frac{e^{ikh} - 2 + e^{-ikh}}{h^2} \\
\frac{2}{\Delta t^2}(1 - \cos\omega\Delta t) &= \frac{2c^2}{h^2}(1 - \cos kh) \\
\sin^2 \frac{\omega\Delta t}{2} &= \frac{c^2\Delta t^2}{h^2} \sin^2 \frac{kh}{2} \qquad (1)
\end{aligned}$$

Solving for ω_h using Equation 1, we get

$$\omega_h = \frac{2}{\Delta t} \arcsin(\frac{c\Delta t}{h} \sin\frac{kh}{2})$$

The numerical dispersion q_h is given by

$$q_h = \frac{\omega_w}{\omega}$$

where $\omega = ck$, substituting ω_h from above

$$q_h = \frac{2}{ck\Delta t} \arcsin(\frac{c\Delta t}{h} \sin\frac{kh}{2})$$

We expand $\arcsin(x)$ using Taylor expansion and substitute in the above

$$q_h \approx 1 - \frac{k^2h^2}{24} + \frac{c^2k^2\Delta t^2}{24} + \cdots + O((kh)^4)$$

Simplifying the above using $\nu = \frac{c\Delta t}{h}$:

$$q_h = 1 - \frac{k^2h^2}{24}(1-\nu^2) + O((kh)^4)$$

Since $\omega_h = \omega q_h = (ck)q_h$ we get the desired numerical dispersion relation

$$\omega_h^{\Delta t} = ck\Big[1 - \frac{k^2h^2}{24}(1-\nu^2) + O((kh)^4)\Big]$$

$\square$

1.2 Phase Error

Define the *Phase Error* for a numerical method to be

$$\Phi(\omega) = |\frac{\omega - \omega_h^{\Delta t}}{\omega}|,$$

where ω satisfies the exact dispersion relation $\omega = ck$. The number of points per wavelength is defined to be $N_{\text{ppw}} = \frac{\lambda}{h} = \frac{2\pi}{kh}$ where λ is the wavelength. How many points per wavelength are required to get the frequency $\omega_h^{\Delta t}$ correct to the following for $\nu = 1/\sqrt{3}$.

1.2.1 1% error

We have that

$$q_h = 1 - \frac{k^2h^2}{24}(1-\nu^2) + O((kh)^4)$$

Substituting for $N_{\text{ppw}} = \frac{2\pi}{kh}$, we get $kh = \frac{2\pi}{N_{\text{ppw}}}$. For $\nu = \frac{1}{\sqrt{3}}$, $1 - \nu^2 = \frac{2}{3}$, and the error term is

$$\frac{k^2h^2}{36} \le 0.01 \implies \frac{kh}{6} \le 0.1 \implies kh \le 0.6 \implies N_{\text{ppw}} \ge \frac{2\pi}{0.6}$$

So 11 points per wavelength should suffice.

1.2.2 0.1% error

Similarly for this error we need 34 points per wavelength.

2 Dispersion relation for homogenous wave equation in 2D

Consider the initial value problem for the two dimensional homogenous wave equation

$$u_{tt} - c^2 \Delta u = 0$$

2.1 Calculate the numerical dispersion error for the centered in time and centered in space scheme

Consider the 2D wave equation as given above and its plane wave solution:

$$u_{ij}^n = e^{i(n\omega t - k_x i h_x - k_y j h_y)}$$

For simplicity we assume that $k_x = k_h h \cos\theta$ and $k_y = k_h h \sin\theta$ and $h_x = h_y$, but this is not strictly required, and the analysis can be done similarly as shown below. Then we have the Fourier components as shown above and we can compute the numerical dispersion by substituting it in the numerical scheme for 2D which is

$$\frac{U_{ij}^{n+1} - 2U_{ij}^n + U_{ij}^{n-1}}{\Delta t^2} = c^2 \Big(\frac{U_{i+1,j}^n - 2U_{ij}^n + U_{i-1,j}^n}{h^2} + \frac{U_i, j+1 - 2U_{ij}^n + U_{i,j-1}^n}{h^2} \Big)$$

By substituting the plane wave form of the Fourier component and dividing both sides by the term U_{ij}^n we get:

$$\frac{1 - \cos\omega\Delta t}{\Delta t^2} = \frac{c^2}{h^2} \Big((1 - \cos k_h h \cos\theta) + (1 - \cos k_h h \sin\theta) \Big)$$

This can be simplified to

$$\sin^2(\frac{\omega \Delta t}{2}) = \frac{c^2 \Delta t^2}{h^2} \Big(\sin^2(\frac{k_h h \cos\theta}{2}) + \sin^2(\frac{k_h h \sin\theta}{2}) \Big)$$

as required.

2.2 Rewriting the above using Courant number

We begin by replacing $(c\Delta t)/h = \nu$ to get

$$\sin^2(\frac{\omega \Delta t}{2}) = \nu^2 \Big(\sin^2(\frac{k_h h \cos\theta}{2}) + \sin^2(\frac{k_h h \sin\theta}{2}) \Big)$$

Substituting for $N_{\text{ppw}} = \frac{2\pi}{kh}$, and noting that $\omega \Delta t$ can be written as $\frac{\pi\nu}{N_{\text{ppw}}}$, substituting above we get:

$$\sin^2(\frac{\pi\nu}{N_{\text{ppw}}}) = \nu^2 \Big(\sin^2(\frac{k_h h \cos\theta}{2}) + \sin^2(\frac{k_h h \sin\theta}{2}) \Big)$$

as required.

2.3 Solving for Normalized Numerical Phase Veilocity

For the majox axes, $\theta = [0, 90, 180, 270]$, the right hand side of the above equation can be simplified, as either $\cos\theta$ or $\sin\theta$ is zero, and the other term becomes unity. Without loss of generality, we thus get

$$\frac{1}{\nu^2}\sin^2\left(\frac{\pi\nu}{N_{\text{ppw}}}\right) = \sin^2\frac{k_h^{\Delta t}h}{2}$$

Therefore,

$$k_h^{\Delta t} = \pm 2\arcsin\frac{\sin\frac{\pi\nu}{N_{\text{ppw}}}}{\nu}$$

Similarly for the grid diagonals, $\sin\theta = \cos\theta = \frac{1}{\sqrt{2}}$, therefore

$$k_h^{\Delta t} = \pm 2\sqrt{2}\arcsin\frac{\sin\frac{\pi\nu}{N_{\text{ppw}}}}{2\nu}$$

2.4 Newton's method for numerically solving for numerical phase velocity

Recall that Newton's method for solving $f(x) = 0$, is

$$x_{\text{new}} = x_{\text{old}} - \frac{f(x)}{f'(x)}$$

till sufficient accuracy is achieved. In the case of normalized numerical phase velocity, we can write $f(x)$ (where x represents $k_h^{\Delta t}$ as

$$\left(\sin^2\left(\frac{k_h h\cos\theta}{2}\right) + \sin^2\left(\frac{k_h h\sin\theta}{2}\right)\right) - \frac{1}{\nu^2}\sin^2\left(\frac{\pi\nu}{N_{\text{ppw}}}\right)$$

Differentiating $f(x)$ we get

$$f'(x) = \frac{h\cos\theta}{2}\sin\left(\frac{k_h h\cos\theta}{2}\right)\cos\left(\frac{k_h h\cos\theta}{2}\right) + \frac{h\sin\theta}{2}\sin\left(\frac{k_h h\sin\theta}{2}\right)\cos\left(\frac{k_h h\sin\theta}{2}\right)$$

Substituting above we get the required relation, and it has to be repeated for each angle θ.

2.5 Numerical solution using the iterative procudure

Normalize h using $\lambda = 1$, then $k = 2\pi$ and

$$\frac{c_h^{\Delta t}}{c} = \frac{2\pi}{k_h^{\Delta t}}$$

Use Courant number $\nu = 0.5$ and $\tau = 10^{-6}$ and consider the angle $\theta \in [0, \pi/2]$, i.e.

$$\theta_j = \frac{(j-1)\pi}{2n}$$

where n is the number of angles considered. Produce plots for $N_{\text{ppw}} = 5, 10, 20$, and comment on the results.

Proof. Since $\nu = 0.5$, and N_{ppw} is given we can calculate h and use that in the Newton method. The implementation was completed in the Julia language and the code is shown in Listing 1.

```julia
##################################################################################
# File          : numerical_phase_velocity.jl
# Author        : Sandeep Koranne (C) 2017, All rights reserved.
# Purpose       : HW Assignment code for calculating numerical phase velocity
#               : using Newton's method
##################################################################################
# Newton method for root
# x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}
# question: how to get f'(x_n), Soln: use Finite Difference Approximation
# f'(x) = [f(x+h)-f(x-h)]/2h centered difference
function finite_difference_derivative(f,x,h)
    return ( ( f(x+h) - f(x-h) )/ (2*h) );
end

# Helper functions
function A(h,theta)
    return (0.5 * h * cos( theta ) );
end
function B(h,theta)
    return (0.5 * h * sin( theta ) );
end
function C(nu,N_PPW)
    return ( (1/(nu^2)) * ( sin( (pi*nu)/(N_PPW*1.0) )^2 ) );
end

function F(x,h,theta,nu,N_PPW)
    return ( ( sin( A(h, theta) * x)^2 ) + ( sin( B(h,theta) * x )^2 ) - C( nu, N_PPW ) );
end

function FD(x,h,theta,nu,N_PPW)
    return ( A(h,theta)*sin(2*A(h,theta)*x) + B(h,theta)*sin(2*B(h,theta)*x) );
end

# Or the explicit function derivative may be given

function NewtonRaphsonSolve( x0,theta,N_PPW,tolerance )
   nu = 0.5; # fixed for this problem
   h  = 1.0/N_PPW;
   MAX_ITERATION_COUNT = 1000;
   for i in 1:MAX_ITERATION_COUNT
        xn = x0 - F(x0,h,theta,nu,N_PPW)/FD(x0,h,theta,nu,N_PPW);
        #print("xn = "); println(xn);
        #print("x0 = "); println(x0);
        #print("f(x0) = "); println(F(x0,h,theta,nu,N_PPW));
        if( abs( xn-x0 ) < tolerance )
            #print("Solution found in : ");
            #println(i);
            return x0;
```

```
        end
        x0 = xn;
    end
end

function CalculateForAllAngles(N,N_PPW)
    THETA_MATRIX=zeros(N+1,2)
    for i in 1:N+1
        theta = (i-1)*2*pi/(N);
        THETA_MATRIX[i,1] = theta*360/(2*pi);
        THETA_MATRIX[i,2] = (2*pi)/NewtonRaphsonSolve( 2*pi, theta, N_PPW, 1e-6);
    end
    writedlm("N.txt", THETA_MATRIX, " ");
end

#sx = NewtonRaphsonSolve( 2*pi,0,5,1e-6);
#println("Newton Raphson solve: ");
#println( sx );
CalculateForAllAngles( 60, 20 );
```

Listing 1: Normalized numerical phase velocity.

□

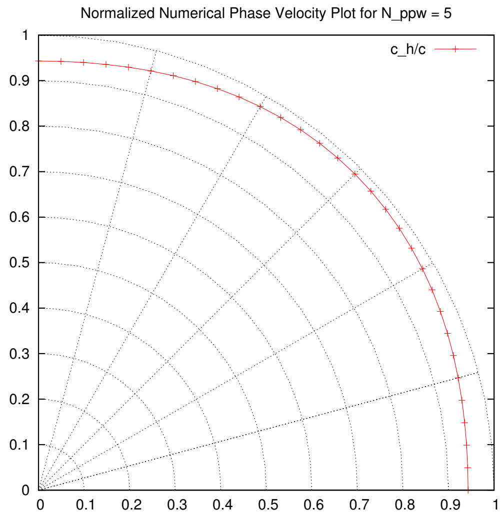
The plot for the ratio of $\frac{c_h^{\Delta t}}{c}$ is shown in Figure 1 for $N_{\text{ppw}} = 5, 10, 20$.

2.5.1 Discussion of results

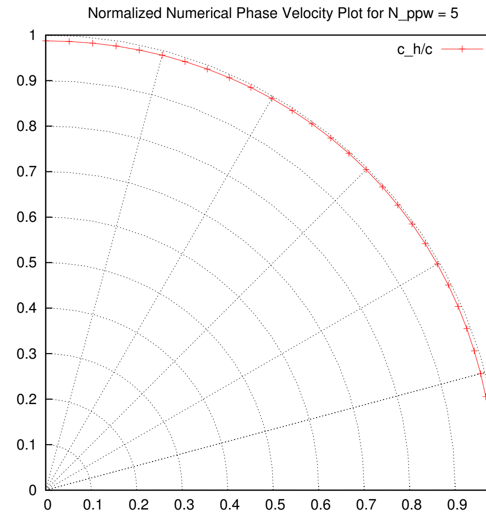
As can be seen in Figure 1, as the N_{ppw} increases the relative error between $c_h^{\Delta t}$ and c decreases rapidly. Moreover, the error is minimum for the diagonal angles, where the numerical error is cancelled to some extent due to the presence of $\sin\theta$ and $\cos\theta$. For θ close to the major axis, the $\cos\theta$ and $\sin\theta$ term is unity, so the numerical dispersion error due to $k_j^{\Delta t}$ is amplified.

2.6 Visualization of Numerical Anisotropy

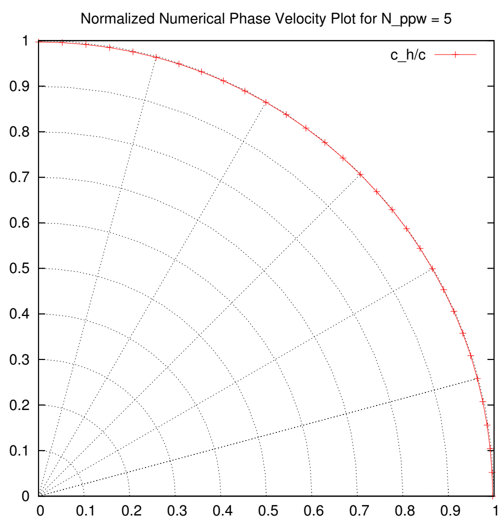
We plotted the error due to numerical dispersion for angle sweeping 2π, as performance of the program was not a problem. The resulting polar plots are shown in Figure 2.



(a) $N_{\text{ppw}} = 5$

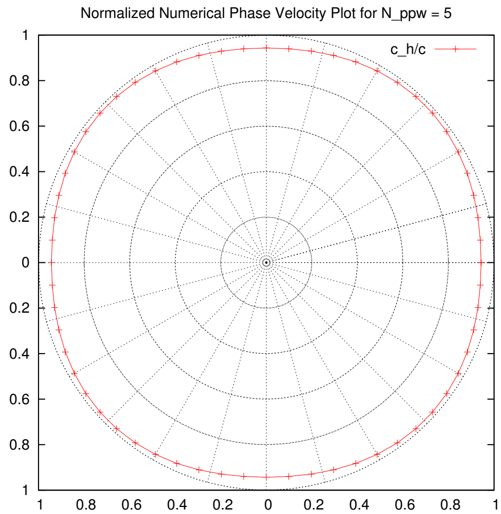


(b) $N_{\text{ppw}} = 10$

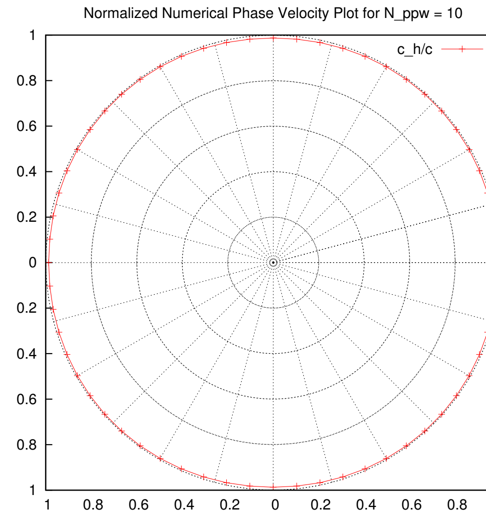


(c) $N_{\text{ppw}} = 20$

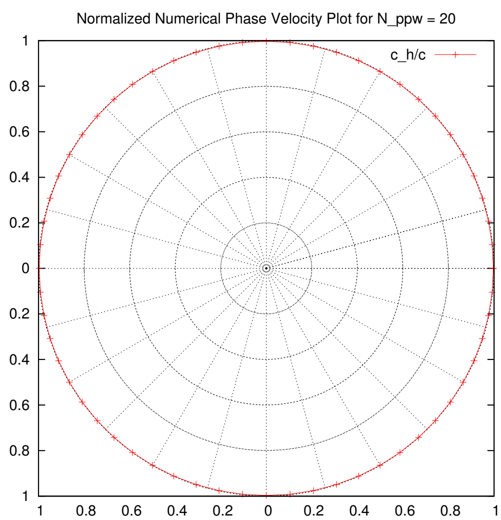
Figure 1: Normalized numerical phase velocity.



(a) $N_{\text{ppw}} = 5$



(b) $N_{\text{ppw}} = 10$



(c) $N_{\text{ppw}} = 20$

Figure 2: Normalized numerical phase velocity.

3 Construction of schemes in Heterogenous Media

Consider the initial value problem for the 1D heterogenous wave equation

$$\eta(x)\frac{\partial^2 u}{\partial t^2} - \frac{\partial}{\partial x}\Big(\gamma(x)\frac{\partial u}{\partial x}\Big) = 0$$

where $\eta(x)$ and $\gamma(x)$ are given strictly positive functions of x.
Consider the semi-discrete second order approximation in space given as

$$\frac{d^2 u_h}{dt^2} + B_h u_h = 0$$

where the operator $B_h : V_0^1 \to V_0^1$ is defined as

$$(B_h u_h)_l = \frac{1}{\eta_l h}\Big(\gamma_{l-\frac{1}{2}}\frac{u_l - u_{l-1}}{h} - \gamma_{l+\frac{1}{2}}\frac{u_{l+1} - u_l}{h}\Big)$$

with

$$\begin{aligned}\eta_l &= \frac{1}{2h}\int_{x_{l-1}}^{x_{l+1}} \eta(x)dx \\ \gamma_{l+\frac{1}{2}} &= \frac{1}{h}\int_{x_{l-1}}^{x_{l+1}} \gamma(x)dx\end{aligned}$$

$\forall u_h \in V_0^1$.

3.1 Problem 3(a)

Show that $(B_h u_h, u_h)_0 = \sum_{l\in\mathbb{Z}} \gamma_{l+\frac{1}{2}}\frac{|u_{l+1}-u_l|^2}{h}$.

Proof. We know that norm for V_0^1 is given by

$$||u_h||^2 = \sum_{l\in\mathbb{Z}} \eta_l |u_l|^2 h$$

therefore $(B_h u_h, u_h)_0$ can be written as

$$(B_h u_h, u_h)_0 = \frac{1}{\eta_l h}\sum_{l\in\mathbb{Z}} \gamma_{l+\frac{1}{2}}\eta_l \frac{|u_{l+1} - u_l|^2}{h^2} h$$

Simplifying the above expression we get

$$(B_h u_h, u_h)_0 = \frac{1}{h^2}\sum_{l\in\mathbb{Z}} \gamma_{l+\frac{1}{2}}|u_{l+1} - u_l|^2$$

as desired. □

3.2 Problem 3(b)

Show that $||B_h|| \leq \frac{4\tilde{c}^2}{h^2}$, where we define $||B_h|| = \sup_{u_h \in V_0^1} \frac{(B_h u_h, u_h)_0}{||u_h||^2}$, and $\tilde{c} = \sup_{l\in\mathbb{Z}} \sqrt{\frac{\gamma_l}{\eta_l}}$.

Proof. We begin by noting the norm equation from part (a) above

$$(B_h u_h, u_h)_0 = \frac{1}{\eta_l h} \sum_{l\in\mathbb{Z}} \gamma_{l+\frac{1}{2}} \eta_l \frac{|u_{l+1} - u_l|^2}{h^2} h$$

Since $\gamma_l = \frac{\gamma_{l+\frac{1}{2}} + \gamma_{l-\frac{1}{2}}}{2}$, we can use index translation to write the above norm as

$$(B_h u_h, u_h)_0 = \frac{2}{h^2} \sum_{l\in\mathbb{Z}} (\gamma_{l+\frac{1}{2}} |u_l|^2 + \gamma_{l-\frac{1}{2}} |u_l|^2)$$

Since the grid is infinite the two sums can be done independently and are infact equal, therefore

$$(B_h u_h, u_h)_0 = \frac{4}{h^2} \sum_{l\in\mathbb{Z}} \gamma_l |u_l|^2$$

Now we use the definition of $\tilde{c}$ to rewrite the above as

$$\begin{aligned} (B_h u_h, u_h)_0 &\leq \frac{4}{h^2} \sum_{l\in\mathbb{Z}} \frac{\gamma_l}{\eta_l} \eta_l |u_l|^2 \\ &\leq \frac{4}{h^2} \sum_{l\in\mathbb{Z}} \tilde{c}^2 ||u_l||^2 \end{aligned} \tag{2}$$

Now define $||B_h|| = \sup \frac{(B_h u_h, u_h)_0}{||u_h||^2}$, then dividing the above equation by $||u_h||^2 \neq 0$ we get

$$||B_h|| \leq \frac{4}{h^2} \tilde{c}^2$$

as required. □

3.3 Problem 3(c)

Finally show that the stability condition for the fully discrete scheme using the 3-point approximation in time as seen in class is $\frac{\tilde{c}\Delta t}{h} < 1$, which is a natural extension of the homogenous case.

Proof. We know from part (b) above that

$$||B_h|| \leq \frac{4}{h^2} \tilde{c}^2$$

Moreover we know that the sufficient condition for the positivity of $E_h^{n+\frac{1}{2}}$ is given by $\Delta t < \frac{2}{\sqrt{||B_h||}}$, substituting from above the expression for $||B_h||$, we get $\Delta t < \frac{h}{\tilde{c}}$, as required. □