HW 2 Solutions

1. Eq. 4.8 is $u_i^n = \lambda^n A e^{jki\Delta x}$

Eq. 4.16 is:

$$u_i^{n+1} = u_i^n - \frac{c\Delta t}{\Delta x}\left(u_{i+1}^n - u_i^n\right)$$

$$\lambda^{n+1} \cancel{A} e^{jki\Delta x} = \lambda^n \cancel{A} e^{jki\Delta x} - \frac{c\Delta t}{\Delta x}\left(\lambda^n \cancel{A} e^{jk(i+1)\Delta x} - \lambda^n \cancel{A} e^{jki\Delta x}\right)$$

$$\rightarrow \lambda^{n+1} \cancel{e^{jki\Delta x}} = \lambda^n\left[\cancel{e^{jki\Delta x}} - \frac{c\Delta t}{\Delta x}\left(e^{jk\cancel{(i+1)}\Delta x} - \cancel{e^{jki\Delta x}}\right)\right]$$

$$\rightarrow \frac{\lambda^{n+1}}{\lambda^n} = 1 - \frac{c\Delta t}{\Delta x}\left(\underbrace{e^{jk\Delta x}}_{\text{Oscillates b/w }[0,1]} - 1\right)$$

For $e^{jk\Delta x} = 1$

$$\lambda = \frac{\lambda^{n+1}}{\lambda^n} = 1 - \frac{c\Delta t}{\Delta x}(1-1) = 1$$

Stable for all conditions

For $e^{jk\Delta x} = 0$

$$\lambda = 1 - \frac{c\Delta t}{\Delta x}(0-1)$$

$$= 1 + \frac{c\Delta t}{\Delta x}$$

For $\lambda < 1$, $\frac{c\Delta t}{\Delta x} < 0$ Upper bound

For $\lambda > 0$, $\frac{c\Delta t}{\Delta x} > -1$ Lower bound

So CFL criterion is

$$\boxed{-1 \leq \frac{c\Delta t}{\Delta x} \leq 0}$$

Values of $e^{jk\Delta x}$ between $[0, 1]$ are less stringent, so above is the necessary condition.

2. Eq. 4.64 is

$$u_i^{n+1} = \tilde{u}_i^{n-1} - \frac{c\Delta t}{\Delta x}\left(u_{i+1}^n - u_{i-1}^n\right)$$

Eq. 4.65

$$\tilde{u}_i^{n-1} = u_i^{n-1} + \gamma\left(u_i^n - 2u_i^{n-1} + u_i^{n-2}\right)$$

$$\rightarrow u_i^{n+1} = u_i^{n-1} + \gamma\left(u_i^n - 2u_i^{n-1} + u_i^{n-2}\right) - \frac{c\Delta t}{\Delta x}\left(u_{i+1}^n - u_{i-1}^n\right)$$

Eq 4.8 is $u_i^n = \lambda^n A e^{jki\Delta x}$

$$\rightarrow \lambda^{n+1} \cancel{Ae^{jki\Delta x}} = \lambda^{n-1} \cancel{Ae^{jki\Delta x}} +$$

$\gamma\left(\lambda^n \cancel{Ae^{jki\Delta x}}\right.$ [redacted] $\left. - 2\lambda^{n-1} \cancel{Ae^{jki\Delta x}} + \lambda^{n-2} \cancel{Ae^{jki\Delta x}}\right)$

$$- \frac{c\Delta t}{\Delta x}\left(\lambda^n \cancel{A} \underbrace{e^{jk(i+1)\Delta x}}_{e^{jk\Delta x}} - \lambda^n \cancel{A} \underbrace{e^{jk(i-1)\Delta x}}_{e^{-jk\Delta x}}\right)$$

$$\rightarrow \lambda^{n+1} = \lambda^{n-1} + \gamma\left(\lambda^n - 2\lambda^{n-1} + \lambda^{n-2}\right) - \frac{c\Delta t}{\Delta x}\left(\lambda^n e^{jk\Delta x} - \lambda^n e^{-jk\Delta x}\right)$$

But $e^{jk\Delta x} - e^{-jk\Delta x} = 2j\sin k\Delta x$
(Eq. 4.23)

$$\rightarrow \lambda^{n+1} = \lambda^{n-1} + \gamma\left(\lambda^{n} - 2\lambda^{n-1} + \lambda^{n-2}\right)$$
$$-2\lambda^{n}\frac{c\Delta t}{\Delta x}j\sin k\Delta x$$

Divide by λ^n

$$\rightarrow \lambda = \lambda^{-1} + \gamma\left(1 - 2\lambda^{-1} + \lambda^{-2}\right) - 2\frac{c\Delta t}{\Delta x}j\sin k\Delta x$$

$$\sigma = \frac{c\Delta t}{\Delta x}\sin k\Delta x \quad \text{(Eq. 4.31)}$$

$$\lambda = \lambda^{-1} + \gamma - 2\gamma\lambda^{-1} + \gamma\lambda^{-2} - 2j\sigma$$

Multiply by λ^2

$$\lambda^3 = \lambda + \gamma\lambda^2 - 2\gamma\lambda + \gamma - 2j\sigma\lambda^2$$

$$\lambda^3 = \lambda^2\left(\gamma - 2j\sigma\right) + \lambda\left(1 - 2\gamma\right) + \gamma$$

$$\boxed{\lambda^3 - \left(\gamma - 2j\sigma\right)\lambda^2 - \left(1 - 2\gamma\right)\lambda - \gamma = 0}$$

For a cubic polynomial

$$ax^3 + bx^2 + cx + d = 0$$

Let $P = b^2 - 3ac$

$$Q = 2b^3 - 9abc + 27a^2 d$$

Then $C = \sqrt[3]{\dfrac{Q \pm \sqrt{Q^2 - 4P^3}}{2}}$

Then $x = \dfrac{-1}{3a}\left(b + C + \dfrac{P}{C}\right)$

For

$$\lambda^3 - (\gamma - 2j\sigma)\lambda^2 - (1-2\gamma)\lambda - \gamma = 0$$

$$P = -(\gamma - 2j\sigma) + 3(1-2\gamma)\lambda$$

$$Q = -2(\gamma - 2j\sigma)^3 - 9(\gamma - 2j\sigma)(1-2\gamma)$$
$$-27\gamma$$

<u>Program & plot</u>

$$C = \sqrt[3]{\frac{Q \pm \sqrt{Q^2 - 4P^3}}{2}}$$

$$P = -(y - 2j\sigma) + 3(1-2y)\lambda$$

$$Q = -2(y-2j\sigma)^3 - 9(y-2j\sigma)(1-2y) - 27y$$

$$Q^2 = \left[-2(y-2j\sigma)^3 - 9(y-2j\sigma)(1-2y) - 27y\right] \cdot \left[-2(y-2j\sigma)^3 - 9(y-2j\sigma)(1-2y) - 27y\right]$$

$$= 4(y-2j\sigma)^6 + 18(y-2j\sigma)^4(1-2y)$$
$$+ 54y(y-2j\sigma)^3 + 18(y-2j\sigma)^4(1-2y)$$
$$+ 81(y-2j\sigma)^2(1-2y)^2 + 243y(y-2j\sigma)(1-2y)$$
$$+ 54y(y-2j\sigma)^3 + 243y(y-2j\sigma)(1-2y)$$
$$+ 729y^2$$

3. a) Main points:

- Compiled languages like Fortran are faster than interpretative languages like Python.

- The outline of a program (pg 67)

- Apply any filters only after <u>all</u> intermediate values have been computed.

- Avoid time dimensions in your arrays to conserve memory.

- Place common calculations such as $\Delta x / \Delta t$ into a variable for speed

- Array calculations (i.e. implicit loops) are faster than explicit loops.

- Avoid if-statements inside of loops when possible.

3. b) Grading my SW model.

Pros	Cons
• Options at top	• Line 35: Use array ops instead of loop
• Helper function for repeated code.	• Anything output related. The csv is a mess
• R-A filter done after spatial loop done	• Line 97: Should convert output interval to a model step interval and actually do the check. ↳ Corrected in problem 4 solution.
	• Helper function causes multiple grid loops to be done.

4. See attached figures.

5.

a) Derive analytical solution:

$$x = x_0 \cos\left(\sqrt{\frac{k}{m}}\, t\right)$$

$$v = \frac{dx}{dt} = -x_0 \sqrt{\frac{k}{m}} \sin\left(\sqrt{\frac{k}{m}}\, t\right)$$

See attached plots

b) Euler's Method

$$\frac{d^2 x}{dt^2} + \frac{k}{m} x = 0$$

$$\rightarrow \frac{dv}{dt} = -\frac{k}{m} x$$

$$\rightarrow \boxed{v_{i+1} = v_i - \Delta t \frac{k}{m} x_i}$$

2^{nd}-order for position:

$$x_{i+1} = 2x_i - x_{i-1} - \Delta t^2 \frac{k}{m} x_i$$

c) A-B

$$\boxed{v_{i+1} = v_i - \frac{\Delta t}{12}\frac{k}{m}\left(23x_i - 16x_{i-1} + 5x_{i-2}\right)}$$

x is same as before.