1. (a) **Solution**: The CFL condition is a necessary condition but not a sufficient condition for stability.

   (b) i. **Solution**: We first need the following Taylor series

$$\begin{aligned}
\phi_j^{n+1} &= \phi + \Delta t\dot{\phi} + \frac{\Delta t^2}{2}\ddot{\phi} + O(\Delta t^3) \\
\phi_{j-1}^n &= \phi - \Delta x\phi' + \frac{\Delta x^2}{2}\phi'' + O(\Delta x^3) \\
\phi_{j+1}^n &= \phi + \Delta x\phi' + \frac{\Delta x^2}{2}\phi'' + O(\Delta x^3)
\end{aligned}$$

where $\phi = \phi_j^n$ and the subscripts have been dropped for clarity. Thus we have that

$$\frac{\phi_j^{n+1} - \frac{1}{2}(\phi_{j+1}^n + \phi_{j-1}^n)}{\Delta t} = \dot{\phi} + \frac{\Delta t}{2}\ddot{\phi} + \frac{\Delta x^2}{2\Delta t}\phi'' + O(\Delta t^2) + O(\Delta x^3/\Delta t)$$

the other term is simply a second order finite difference approximation

$$\frac{\phi_{j+1}^n - \phi_{j-1}^n}{2\Delta x} = \phi' + \frac{\Delta x^2}{6}\phi''' + O(\Delta x^3)$$

thus plugging in the scheme and using the fact that $\dot{\phi} + c\phi' = 0$ we obtain an error of the form

$$\frac{\Delta t}{2}\ddot{\phi} + \frac{\Delta x^2}{6}c\phi''' + \frac{\Delta x^2}{2\Delta t} + O(\Delta t^2) + O(\Delta x^3/\Delta t) + O(\Delta x^3)$$

thus as $\Delta t, \Delta x \to 0$ we will get a consistent relation provided that $\Delta x^2/2\Delta t \to 0$, i.e. Δx^2 goes to zero faster then Δt. If we do not have this, i.e. Δt goes to zero faster then Δx^2, that term will not go to zero and instead approach some constant and thus we do not have a consitent solution. Thus halving Δx each timestep is not a problem. However, halving Δt will eventually case problems.

   ii. **Solution**: To find the stability of the scheme we preform von Neumann stability analysis. First re-writing the scheme as follows

$$\phi_j^{n+1} = (1/2 - \alpha/2)\phi_{j+1}^n + (1/2 + \alpha/2)\phi_{j-1}^n$$

with $\alpha = c\Delta t/\Delta x$. Performing von Neumann analysis we immediatley obtain an amplification factor of the form

$$\begin{aligned}
A &= (1/2 - \alpha/2)e^{ik\Delta x} + (1/2 + \alpha/2)e^{-ik\Delta x} \\
&= \cos(k\Delta x) - i\alpha\sin(k\Delta x)
\end{aligned}$$

we require that $|A| \leq 1$

$$\sqrt{\cos(k\Delta x)^2 + \alpha^2\sin(k\Delta x)} \leq 1$$

we could analyse the various possible values of sine and cosine but we instead recall that $\cos^2 x + \sin^2 x = 1$. Thus if $\alpha = 1$ we satisfy this identity. If $\alpha^2 < 1$ we would have $\cos^2 x + \alpha^2\sin^2 x \leq 1$ while if it is greater, we would have the sum being greater then one. Thus we require that

$$\alpha^2 \leq 1 \Rightarrow \frac{c\Delta t}{\Delta x} \leq 1$$

2. (a) **Solution**: The scheme is given by

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

To find the discrete dispersion relationship we sub in a plane wave

$$\phi_j^n = e^{i(kj\Delta x - \omega n \Delta t)}$$

writing the frequency as $\omega = \omega_R + i\omega_I$ we immediately obtain

$$\phi_j^n = e^{\omega_i n \Delta t} e^{i(kj\Delta x - \omega_R n \Delta t)}$$

so we require that $\omega_I \leq 0$ lest the solution grow in time and become unstable. Subbing in the fisrt relationship into the above scheme yields

$$\left(\frac{e^{-i\omega\Delta t} - e^{i\omega\Delta t}}{2\Delta t}\right)\phi_j^n = -c\left(\frac{e^{ik\Delta x} - e^{-ik\Delta x}}{2\Delta x}\right)\phi_j^n$$

which we can re-write as

$$\sin(\omega\Delta t) = \frac{c\Delta t}{\Delta x}\sin(k\Delta x)$$

Clearly if $|\alpha| = |c\Delta t/\Delta x| \leq 1, \omega$ will be real since sine is bounded by $[-1,1]$. We can see there exist two solutions given the periodic nature of the sine function. If $\alpha > 1$ we cannot possibly have ω being real since sine is bounded by $[-1,1]$, however if ω is complex, sine is entire and thus can take on values greater than 1. Now writing $\omega = \omega_R + i\omega_I$ we have that

$$\sin(\omega_R \Delta t + i\omega_I \Delta t) = \alpha \sin(k\Delta x)$$

expanding this out we obtain

$$\sin(\omega_R\Delta t)\cosh(\omega_I\Delta t) + i\cos(\omega_R\Delta t)\sinh(\omega_R\Delta t) = \alpha\sin(k\Delta x)$$

since the right side is always real, $\omega_R\Delta t = \pm\pi/2$ thus we have that

$$\pm\cosh(\omega_I\Delta t) = \alpha\sin(k\Delta x)$$

however since cosh is an even function, if ω_I is a solution, $-\omega_I$ is also a solution. Thus there exists an $\omega_I > 0$ and hence the method is unstable if $|\alpha| > 1$.

(b) **Solution**: To derive the modified equation we have from above (where again the indices of ϕ_j^n have been dropped for clarity)

$$\begin{aligned}
\phi_j^{n+1} &= \phi + \Delta t\dot{\phi} + \frac{\Delta t^2}{2}\ddot{\phi} + \frac{\Delta t^3}{6}\dddot{\phi} + \frac{\Delta t^4}{4!}\ddddot{\phi} + O(\Delta t^5)\\
\phi_j^{n-1} &= \phi - \Delta t\dot{\phi} + \frac{\Delta t^2}{2}\ddot{\phi} - \frac{\Delta t^3}{6}\dddot{\phi} + \frac{\Delta t^4}{4!}\ddddot{\phi} + O(\Delta t^5)\\
\phi_{j-1}^{n} &= \phi - \Delta x\phi' + \frac{\Delta x^2}{2}\phi'' - \frac{\Delta x^3}{6}\phi''' + \frac{\Delta t^4}{4!}\phi'''' + O(\Delta x^5)\\
\phi_{j+1}^{n} &= \phi + \Delta x\phi' + \frac{\Delta x^2}{2}\phi'' + \frac{\Delta x^3}{6}\phi''' + \frac{\Delta t^4}{4!}\phi'''' + O(\Delta x^5)
\end{aligned}$$

plugging this into the above leap frog scheme we find that

$$\begin{aligned}
\frac{\phi_j^{n+1} - \phi_j^{n-1}}{2\Delta t} &= \dot{\phi} + \frac{\Delta t^2}{6}\dddot{\phi} + O(\Delta t^4)\\
c\frac{\phi_{j+1}^{n} - \phi_{j-1}^{n}}{2\Delta x} &= c\phi' + c\frac{\Delta x^2}{6}\phi''' + O(\Delta x^4)
\end{aligned}$$

so the modified equation is thus

$$\frac{\partial\phi}{\partial t} + c\frac{\partial\phi}{\partial x} = -\frac{\Delta t^2}{6}\frac{\partial^3\phi}{\partial t^3} - c\frac{\Delta x^2}{6}\frac{\partial^3\phi}{\partial x^3} + O(\Delta x^4) + O(\Delta t^4)$$

3. (a) **Solution**: The scheme can be written as

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=\frac{u_{i+1}^n-2u_i^n+u_{i-1}^n}{\Delta x^2}$$

which can be re-written as

$$u_i^{n+1}=(1-2\alpha)u_i^n+\alpha u_{i+1}^n+\alpha u_{i-1}^n$$

with $\alpha=\Delta t/\Delta x^2$. The boundary conditions are $u(0,t)=u(1,t)=0$. See `q2a.m` for implementation. As can be seen from the graphs, the solution diverges at $\Delta t=0.00004$ while converges for $\Delta t=0.00003, 0.000015$. This can easily be explained by appealing to the amplication factor. Performing von Neumann stability analysis on the above, we immediately require that

$$A=|1-2\alpha+2\alpha\cos(k\Delta x)|\leq 1$$

taking the extreme values of the cosine we obtain that if $\cos(k\Delta x)=1, |1|\leq 1$ and if $\cos(k\Delta x)=-1$ we obtain that $\alpha\leq 1/2$. Thus we have a CFL condition on the $\Delta x,\Delta t$. It is easy to verify that if $\Delta t=0.00004$ we will not satisfy the above inequality and hence the amplification factor is greater then 1 and thus we will obtain a divergent solution.

(b) **Solution**: We will use the above scheme, however this time we have a Neumann boundary condition that $u_x(0,t)=-1$. We choose a second order finite difference approximation, i.e.

$$u_x(0,t)=\frac{u_1^n-u_{-1}^n}{2\Delta x}=-1$$

solving for u_{-1}^n we obtain that

$$u_{-1}^n=u_1^n+2\Delta x$$

thus the above scheme becomes, at $i=0$

$$u_0^{n+1}=(1-2\alpha)u_0^n+2\alpha u_1^n+2\alpha\Delta x$$

The above formulation can be written in the form

$$u^{n+1}=Bx^n+b$$

and see `q2b.m` for implementation. As can be seen from the graphs, the solution, at long times we find that the solution approaches a straight line. This result makes sense because the steady state solution of the heat equation, with the given boundary conditions is trivially given by $u(x)=x-1$. This agrees with the above result.

(c) **Solution**: We now apply the Crank-Nicolson method on the second derivative term. The scheme can be written as

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=\frac{1}{2}\left(\frac{\partial^2 u^{n+1}}{\partial x^2}+\frac{\partial^2 u^n}{\partial x^2}\right)$$

where, upon expanding out the terms on the right as above and gathering terms we obtain

$$u_i^{n+1}(1+2\alpha)-\alpha u_{i+1}^{n+1}-\alpha u_{i-1}^{n+1}=\alpha u_{i+1}^n+\alpha u_{i-1}^n+(1-2\alpha)u_i^n$$

Additionally, we must pay attention to the boundary condition that $u_x(0,t)=-1$. Applying the same technique as above, substituting in $u_{-1}^{n+1,n}$ we obtain

$$u_0^{n+1}(1+2\alpha)-2\alpha u_1^{n+1}-2\Delta x\alpha=2\alpha u_1^n+2\Delta x\alpha+(1-2\alpha)u_0^n$$

The modification of scheme is that we now have

$$Au^{n+1}=Bu^n+b\Rightarrow u^{n+1}=A^{-1}Bx^n+A^{-1}b$$

here given that the matrices are small and constant it is quicker to just initially calculate the inverse products. See `q2c.m` for the implementation. Unlike Method 1, the Crank-Nicolson is stable for all the above time steps. This is because the amplification factor is given by

$$A = \left|\frac{1 - 2\alpha + 2\alpha\cos(k\Delta x)}{1 + 2\alpha - 2\alpha\cos(k\Delta x)}\right| \leq 1$$

which is stable for all α. For $\cos(k\Delta x) = 1$ we have that $|1| \leq 1$ and for $\cos(k\Delta x) = -1$ we have that

$$A = \left|\frac{1 - 4\alpha}{1 + 4\alpha}\right| \leq 1$$

which is always less then 1. Thus the amplification factor will not grow. This is demonstrated in the attached figures.

(d) **Solution**: The equation of motion is given as

$$\frac{\partial u}{\partial t} = \frac{\partial k}{\partial x}\frac{\partial u}{\partial x} + k\frac{\partial^2 u}{\partial x^2}$$

We compute the derivative of k as follows

$$\frac{\partial k}{\partial x} = \frac{k_{i+1} - k_{i-1}}{2\Delta x}$$

Given that the boundary conditions are zero, we do not need to worry about the derivatives of k at $x = 0, 1$ because the function is 0 there. Applying the Crank-Nicolson method we have that

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{1}{2}\left(k_i'\frac{\partial u^{n+1}}{\partial x} + k_i\frac{\partial^2 u^{n+1}}{\partial x^2} + k_i'\frac{\partial u^n}{\partial x} + k_i\frac{\partial^2 u^n}{\partial x^2}\right)$$

approximating the u_x as a second order finite difference, we obtain the following scheme, after a lot of algebra

$$(1 + 2\beta)u_i^{n+1} - (\alpha + \beta)u_{i+1}^{n+1} + (\alpha - \beta)u_{i-1}^{n+1} = (\alpha + \beta)u_{i+1}^{n+1} + (1 - 2\beta)u_i^n + (-\alpha + \beta)u_{i-1}^n$$

where $\alpha = \Delta k_i'/4\Delta x, \beta = k_i\Delta t/2\Delta x^2$. See `q2d.m` for implementation. As can be seen from the plot the addition of a non-constant diffusivity changes the distribution of heat in the rod immensely. Since the diffusivity is non-linear the evolution of heat is now non-linear. For example, the right side of the rod is more resistant to heat transfer and thus it takes longer to cool when compared with the constant diffusivity. Thus the original symmetry of the problem is also broken.

4. (a) **Solution**: The upwind scheme is given by

$$\phi_j^{n+1} = (1 - \alpha)\phi_j^n + \alpha\phi_{j-1}^n$$

with $\alpha = \Delta t/\Delta x$. We also have periodic boundary conditions such that $\phi(0) = \phi(1)$. This implemented in `q4a.m`. For $\Delta t = 0.0156$ we have that the solution diverges which makes sense since $\Delta t/\Delta x > 1$ so it diverges. We see that as we decrease Δt we have that the solution becomes more diffusive.

(b) **Solution**: For the generating code, see `q4b.m`. From the graph attached we can see that as Δx gets smaller, the error goes at Δx. Similarly, since $\Delta x = 2\Delta t$ the error also goes as Δt. The reason the error does not go as Δx for large values of Δx is the error is dominated by other terms. However as we get smaller the error becomes dominated by Δx.

(c) **Solution**: See `q4c.m` for code. For the narrower peak we see that as we decrease Δt we have more dampening and dispersion. This is because as we get a narrower peak, the Fourier transform of the function becomes broader and there are more wavenumbers. For a delta function all frequncies are present. Thus there will be smaller and smaller waves sampled as Δt increase and they will satisfy a dispersive relationship. This can be seen in the plot.

(d) **Solution**: See `q4d.m`. The exact solution can be modelled by the fact that if $x < t$ then we have 0 and if $x > t$ we will have $1 - cos(x - t)$. In other words we can write the solution as $u(x,t) = H(x-t)(1-cos(x-t))$ where $H(x-t)$ is the Heaviside step function. Unfortunately I wasn't able to get the code working properly and the attached graphs showing a wildely oscillating function. I believe this is due to the fact that leapfrog samples every other point and hence, some points are zero and others are not. Thus somtimes we will have two positive points contributing and other times not, thus leading to an oscillating function.