Catam Project Report
PartII Additional Projects
(July 2020 Edition)

Freud Fan

29/04/2021

- Phase and Group Velocity
- Programs

1 Phase and Group Velocity

Given the Klein-Gordon equation

$$u_{tt} - c_0^2 u_{xx} = -q^2 u, \quad c_0 > 0,\ q \geq 0, \tag{1}$$

w.l.o.g., take $c_0 = 1^*$.

1.1 Question 1

Fourier transform (1) w.r.t x,

$$\tilde{u_{tt}} + k^2 \tilde{u} + q^2 \tilde{u} = 0.$$

Try solutions of the form $\tilde{u} = e^{\lambda t}$ to get $\lambda^2 + k^2 + q^2 = 0$, i.e., $\lambda = \pm i\sqrt{q^2 + k^2}$. Hence

$$\tilde{u}(k,t) = A(k)e^{-i\sqrt{q^2+k^2}t} + B(k)e^{i\sqrt{q^2+k^2}t}.$$

Substitute in i.c.'s $u(x,0) = f(x)$, $u_t(x,0) = 0$,

$$\begin{aligned}
\tilde{u}(k,0) = \tilde{f}(k) &\Rightarrow A(k) + B(k) = \tilde{f}(k) \\
\tilde{u}_t(k,0) = 0 &\Rightarrow i\sqrt{q^2+k^2}(A(k) - B(k)) = 0,
\end{aligned}$$

i.e., $A(k) = B(k) = \tilde{f}(k)/2$. Apply inverse Fourier transform to $\tilde{u}$,

$$\begin{aligned}
u(x,t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{ikx}\tilde{u}(k,t)dk = \frac{1}{4\pi}\int_{-\infty}^{\infty} \tilde{f}(k)e^{ikx-i\sqrt{q^2+k^2}t}dk + \frac{1}{4\pi}\int_{-\infty}^{\infty} \tilde{f}(k)e^{ikx+i\sqrt{q^2+k^2}t}dk \\
&= \frac{1}{4\pi}\int_{-\infty}^{\infty} \tilde{f}(k)e^{ikx-i\Omega(k)t}dk + complex\ conjugate
\end{aligned}$$

where $\Omega(k) = \sqrt{q^2+k^2}$. The last equality comes from the evenness of $\tilde{f(k)}^\dagger$. For fixed $q > 0$, phase velocity $v_p = \Omega(k)/k = \sqrt{q^2+k^2}/k \to \pm 1$ as $k \to \pm\infty$, $v_p \to \pm\infty$

---

*For any $c_0' > 0$, if $u(x,t) = g(x - c_0't)$ (to be checked a-posteriori) is a solution for (1) with parameters c_0' and q', then $u(x,t) = g(x - c_0t)$ is a solution for (1) with parameters c_0 and q, by noticing that the solution has a symmetry of translation depending on the parameters with i.c.'s invariant. In the case of **Question 4**, there is an extra parameter $\omega_0 c_0'/c_0 = \omega_0' \to \omega_0$ in the b.c.'s.

†Given $f(x)$ even and real,

$$\tilde{f}(-k) = \tilde{f}(k)^* = \int_{-\infty}^{\infty} f(x)e^{ikx}dx = \int_{-\infty}^{\infty} f(-x)e^{-ikx}dx = \int_{-\infty}^{\infty} f(x)e^{-ikx}dx = \tilde{f}(k)$$

$$\frac{1}{4\pi}\int_{-\infty}^{\infty} \tilde{f}(k)e^{ikx+i\sqrt{q^2+k^2}t}dk = \frac{1}{4\pi}\int_{-\infty}^{\infty} \tilde{f}(-k)e^{-ikx+i\sqrt{q^2+k^2}t}dk = \left(\frac{1}{4\pi}\int_{-\infty}^{\infty} \tilde{f}(k)e^{ikx-i\sqrt{q^2+k^2}t}dk\right)^*$$

as $k \to 0^{\pm}$. Group velocity $v_g = \Omega'(k) = k/\sqrt{q^2+k^2} = 1/v_p \to \pm 1$ as $k \to \pm\infty$, $v_g = 0$ when $k = 0$. For $q = 0$, $v_g = 1/v_p = k/|k| = 1$ when $k > 0$, and -1 when $k < 0$, not defined for $k = 0$ due to different limits when approached from 0^+ or 0^-. Fig.1 gives a sketch of the two velocities when $q > 0$. The phase velocity always has larger magnitude than the group velocity in this case.

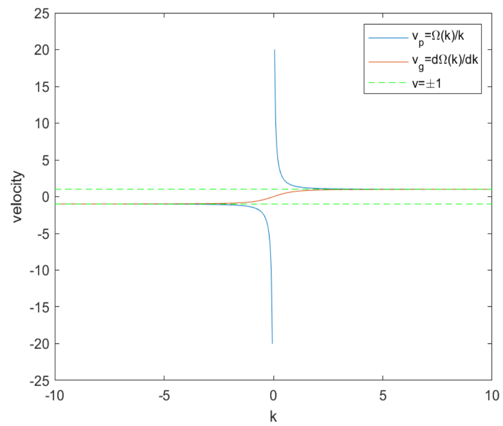

Figure 1: Phase velocity $\Omega(k)/k$ and group velocity $\Omega'(k)$ against k

Physically, this means the wave propagates at the **phase velocity**, which denpends on the wavenumber k ($k > 0$). The wave propagates faster with smaller wavenumber, and propagates at constant velocity when it has a large wavenumber. The larger the group velocity, the lower the phase velocity (in this case).

1.2 Question 2

The **stationary phase method** states that the asymptotic approximation of a function I(x) in the form

$$I(x) = \int_a^b f(k)e^{ix\phi(k)}dk, \quad -\infty \le a < b \le +\infty$$

as $|x| \to \infty$ is mainly contributed by the neighbourhood of k around where $\phi'(k) = 0$ (and possibly end points a and b) due to a cancellation of sinusoids with rapidly varying phase.

Applying this method we find the asymptotic of u(x,t) with $\phi(k) = kV - \Omega(k)$, by a substitution of $V \equiv x/t$ and noticing that $\phi'(k) = V - \Omega'(k) = 0$ when k is such that $\Omega'(k_0) = V$. Assume $\tilde{f}(k_0) \neq 0$ and $\phi''(k_0) < 0$,

$$\begin{aligned}
u(x,t)\big|_{x=Vt} &= \frac{1}{4\pi}\int_{-\infty}^{\infty} \tilde{f}(k)e^{it(kV-\Omega(k))}dk \quad + \ cx\ conjugate \\
&\sim \frac{1}{4\pi}\int_{k_0-\epsilon}^{k_0+\epsilon} \tilde{f}(k)e^{it\phi(k)}dk \quad + \ cx\ conjugate \\
&\sim \frac{1}{4\pi}\int_{k_0-\epsilon}^{k_0+\epsilon} \tilde{f}(k_0)e^{it(\phi(k_0)+\frac{1}{2}\phi''(k_0)(k-k_0^2))}dk \quad + \ cx\ conjugate \\
&= \frac{e^{it\phi(k_0)}\tilde{f}(k_0)}{2\pi\sqrt{-2t\phi''(k_0)}}\int_{-\sqrt{-t\phi''(k_0)\epsilon/2}}^{\sqrt{-t\phi''(k_0)\epsilon/2}} e^{-is^2}ds \quad + \ cx\ conjugate \\
&\sim \frac{e^{it\phi(k_0)}\tilde{f}(k_0)}{2\pi\sqrt{-2t\phi''(k_0)}}\int_{-\infty}^{\infty} e^{-is^2}ds \quad + \ cx\ conjugate \\
&= \frac{e^{it\phi(k_0)}\tilde{f}(k_0)}{2\sqrt{-2\pi\phi''(k_0)t}}e^{-\frac{\pi}{4}i} \quad + \ cx\ conjugate \\
&= \frac{|\tilde{f}(k_0)|}{\sqrt{2\pi\Omega''(k_0)t}} cos\left(k_0x - \Omega(k_0)t + arg\tilde{f}(k_0) - \frac{\pi}{4}\right)
\end{aligned}$$

where the first '$\sim$' is from integration by parts, as other terms integrating from $-\infty$ to $k_0-\epsilon$ and from $k_0+\epsilon$ to ∞ is $\mathcal{O}(1/|t|)$; the second '$\sim$' is from Taylor expansion; the second '$=$' is from substitution $s = \sqrt{-t\phi''(k_0)/2}(k-k_0)$, given $\phi''(k_0) = -\Omega''(k_0) = -q^2/(q^2+k_0^2)^{3/2} < 0$; the third '$\sim$' is from Localization Principle; the last '$=$' is by setting $\tilde{f}(k_0) = |\tilde{f}(k_0)|e^{iarg\tilde{f}(k_0)}$.

Now, $\Omega'(k_0) = k_0/\sqrt{q^2+k_0^2} = V \equiv x/t \Rightarrow k_0 = qx/\sqrt{t^2-x^2}$ (for $|x| < t$ and w.l.o.g. $k_0 > 0$ since $\tilde{f}(k)$ even in k), $\Omega''(k_0) = q^2/(q^2+k_0^2)^{3/2} = (t^2-x^2)^{3/2}/(t^3q)$. Hence

$$u(x,t)\big|_{x\equiv Vt} \sim \frac{q^{1/2}t}{\sqrt{2\pi}(t^2-x^2)^{3/4}}\tilde{f}\left(\frac{qx}{\sqrt{t^2-x^2}}\right) cos\left(q\sqrt{t^2-x^2}+\pi/4\right), \quad |x| < t. \quad (2)$$

Physically, we are looking at a point which is moving in time and we see the solution $u(x,t)$ on the line $x \equiv Vt$. $u(x,t)$ is a superposition of waves with initial data e^{ikx}, each associated with an $\tilde{f}(k)$, the Fourier transform of some initial data. The **group velocity** is the propagation velocity of the envelope of the wave packet whose velocity is dominated by the wave with the spatial frequency (wavenumber) k_0 ('carrier frequency') such that $\Omega'(k_0) = V$, and is indeed $\Omega'(k)$ from above formula. Waves inside the wave packet propagates at their phase velocity.

1.3 Question 3

We aim to solve (1) with i.c.'s

$$u(x,0) = f(x) = (1-x^2)^2 \mathbb{I}_{\{|x|\leq 1\}}, \quad u_t(x,0) = 0. \tag{3}$$

1.3.1 Solving for u(x,t) in the initial-value problem: Programming Task

A program solving for u using the centred-difference approximation[‡] is listed on page 16, named **Q3fds(xmax,tmax,hx,ht,q,Er)**. $xmax$, $tmax$ are the range we are integrating, hx, ht are the step lengths, Er is the tolerance we accept.

In the grid, x, t range from 0 to 50, tolerance set to be 0.0001. Results in Fig.2-10.

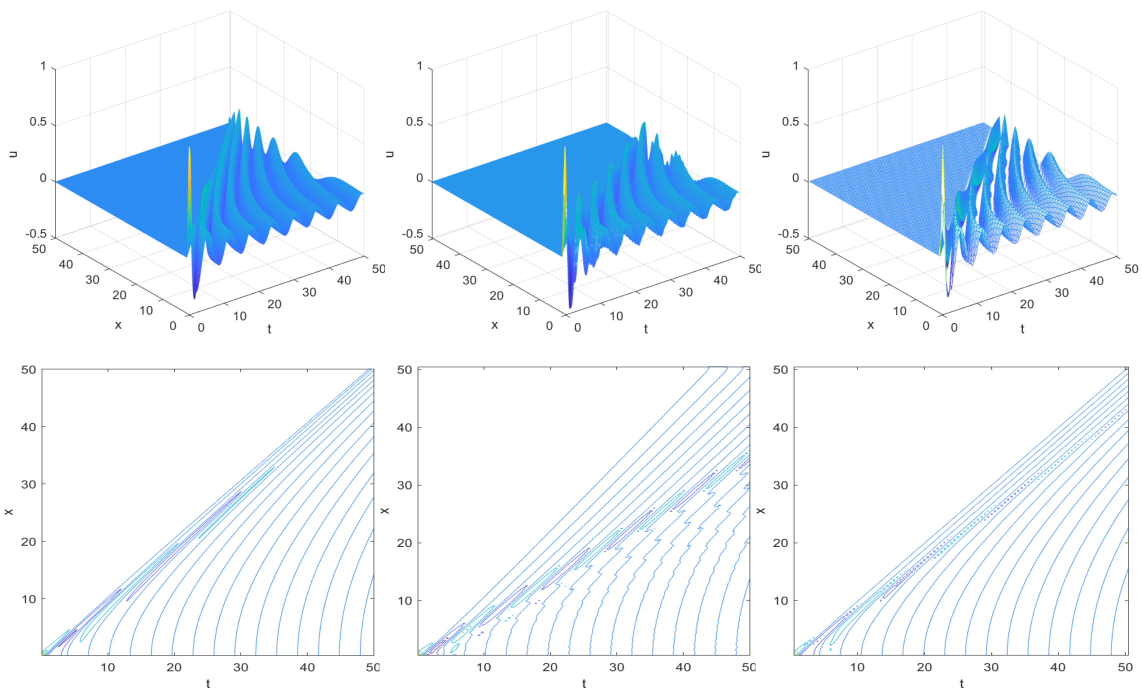

Figure 2: Mesh and contour plots of u for $q = 1$, $(\Delta t, \Delta x) = (0.05, 0.05)$, $(0.05, 0.5)$, $(0.5, 0.5)$

[‡]Given i.c. $u_t(x,0) = 0$, by setting $x = i\Delta x$, $t = j\Delta t$, we have

$$0 = u_t(x,0) = \lim_{\Delta t \to 0} \frac{u(i\Delta x, \Delta t) - u(i\Delta x, -\Delta t)}{2\Delta t} \approx \lim_{\Delta t \to 0} \frac{u_i^1 - u_i^{-1}}{2\Delta t} \quad \Rightarrow \quad u_i^1 = u_i^{-1}.$$

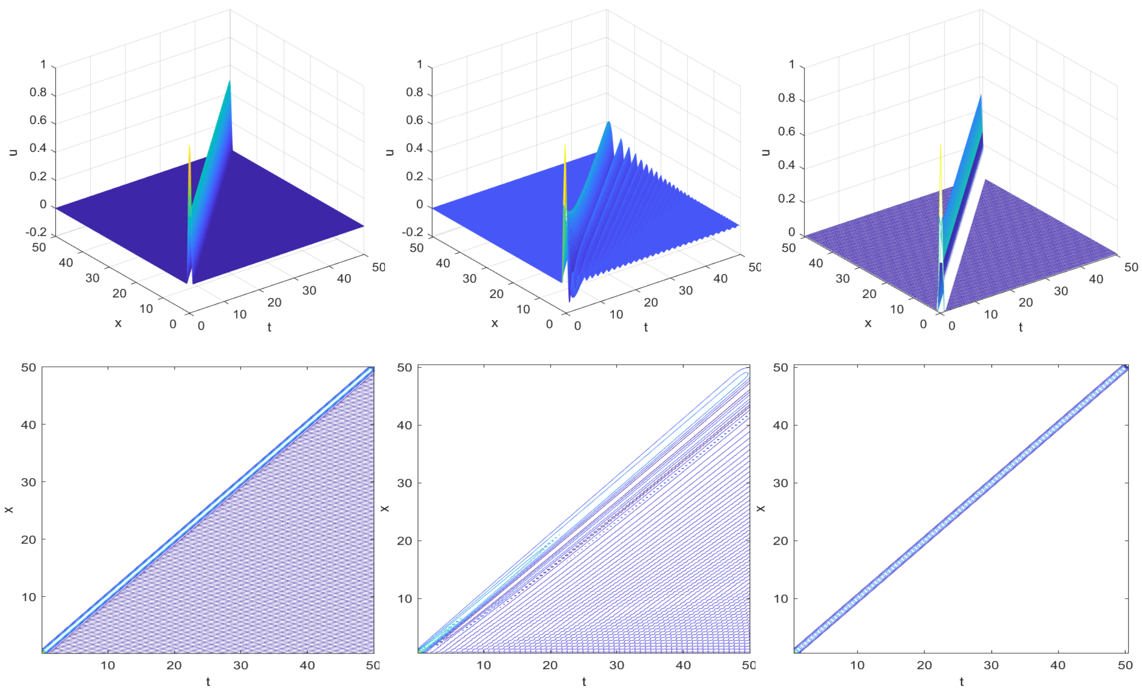

Figure 3: Mesh and contour plots of u for $q = 0$, $(\Delta t, \Delta x) = (0.05, 0.05)$, $(0.05, 0.5)$, $(0.5, 0.5)$

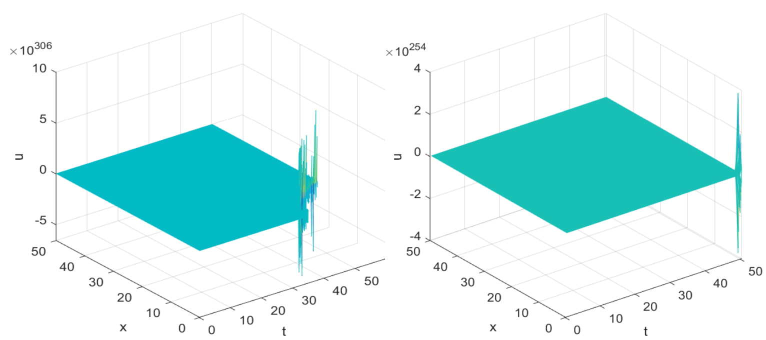

Figure 4: Mesh plots of u for $q = 1$ (similar for $q = 0$), $(\Delta t, \Delta x) = (0.2, 0.05)$, $(0.5, 0.05)$

The numerical scheme is

- stable, as the errors eventually fall within tolerance, i.e., the error for a cell generally decays as computation proceeds. Below are selected values of errors along with the iteration **Q3fds(10,10,0.5,0.5,1,0.0001)**, i.e., variables range from 0 to 10 with step lengths 0.5 and tolerance 0.0001,

Iteration times	Error
6441	3.981771547741017e+04
6450	2.812020678311800e+03
6454	1.099517819378291e+02
6455	66.363943418591617
6456	8.028561917444918
6457	4.516066078562766
6458	0

Table 1: Errors along with the iteration.

- accurate, if $\Delta t/\Delta x \leq 1$, but inaccurate if $\Delta t/\Delta x > 1$ (see Fig.4 for a blowing up), and the accuracy becomes even poorer as the ratio decreases to just above 1 or as Δx decreases. See Tbl.2.

Δx	$\Delta t/\Delta x$	Percentage error%
0.05	2	1e+305
0.05	10	1e+253
0.05	20	1e+154
0.1	2	1e+279
0.1	10	1e+124
1	2	1e+22
1	10	1e+10

Table 2: Accuracy of the scheme w.r.t. Δx and $\Delta t/\Delta x$.

To see accuracy, for the $q = 0$ case, compare the solution u_{fd} from the numerical scheme **Q3fds(50,50,0.05,0.05,0,0.0001)** with the exact solution

$$u(x,t) = \frac{1}{2}((f(x-t)+f(x+t)) = \frac{1}{2}((1-(x-t)^2)^2\mathbb{I}_{\{|x-t|\leq 1\}}+(1-(x+t)^2)^2\mathbb{I}_{\{|x+t|\leq 1\}}) \tag{4}$$

whose graph is in Fig.5, a lot similar to Fig.3(i).

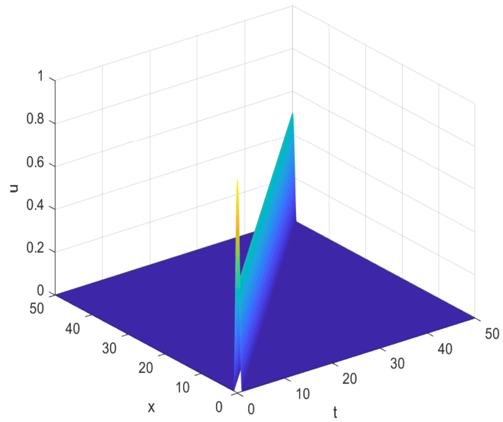

Figure 5: Exact solutions $u(x,t)$ from (3)

See Tbl.3 for selected results from a range of x and t.

(t, x)	u_{fd}	u	Percentage error%
(0.55,0.15)	0.48285	0.482850000000000	0
(7.7,7.1)	0.2048	0.204800000000000	0
(1.85,0.85)	-2.6645e-15	0	2.664500000000000e-13
(29.5,17.6)	-4.2188e-15	0	4.218800000000000e-13
(35,34.35)	0.16675	0.166753125000001	3.125000000991918e-04
(40.8,20.15)	6.6613e-16	0	6.661300000000001e-14

Table 3: A selection of points of numerical solutions u_{fd} and exact solutions u ($q = 0$).

Therefore very high accuracy when $|x - t| < 1$, high accuracy up to rounding errors when $|x - t| \geq 1$ for the $q = 0$ case.

For the $q = 1$ case, compare the solution u_{fd} from the numerical scheme **Q3fds(50,50,0.05,0.05,1,0.0001)** with the exact solution

$$\begin{aligned} u(x,t) &= \frac{1}{2}\left((f(x-t) + f(x+t) - qt\int_{-\pi/2}^{\pi/2} J_1(qtcos\theta)f(x+tsin\theta)d\theta\right) \qquad (5) \\ &= \frac{1}{2}((1-(x-t)^2)^2\mathbb{I}_{\{|x-t|\leq 1\}} + (1-(x+t)^2)^2\mathbb{I}_{\{|x+t|\leq 1\}} \\ &\quad - qt\int_{max\{-\pi/2,-sin^{-1}((1+x)/t)\}}^{min\{\pi/2,sin^{-1}((1-x)/t)\}} J_1(qtcos\theta)(1-(x+tsin\theta)^2)^2 d\theta) \;\; given\ sin^{-1}\ real. \end{aligned}$$

and the asymptotic expansion u_{ae} in (2), whose graph is in Fig.6, a lot similar to Fig.2(i).

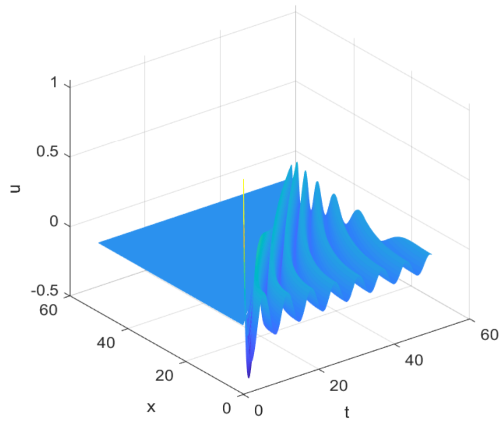

Figure 6: Asymptotic expansion $u(x,t)$ from (2)

See Tbl.4 for selected results from a range of x and t.

(t,x)	u_{fd}	u	Percentage error%
(0.3,0.15)	0.75563	0.75551	0.01200
(25.2,25)	0.026712	0.030277	0.356500
(26.45,26.4)	0.010296	0.0070613	0.3234700
(29.2,1)	0.010737	0.010496	0.024100
(30.4,16.95)	0.06199	0.061784	0.020600

Table 4: A selection of points of numerical solutions u_{fd} and exact solutions u ($q = 1$).

Therefore good accuracy of the numerical scheme, especially when $|x - t| > 1$, for the $q = 1$ case.

In the above section, we found $\tilde{f}(k)$ by Fourier transform,

$$\begin{aligned}
\tilde{f}(k) &= \int_{-\infty}^{\infty} f(x)e^{-ikx}dx = \int_{-1}^{1}(1-x^2)^2 cos(kx)dx \\
&= -\frac{16}{k^3}sink - \frac{48}{k^4}cosk + \frac{24}{k^4}\int_{-1}^{1} cos(kx)dx \\
&= -\frac{16}{k^3}sink - \frac{48}{k^4}cosk + \frac{48}{k^5}sink
\end{aligned}$$

where the second '=' comes from the oddness of $sin(kx)$ and the third '=' comes from integration by parts. Plots in Fig.7.

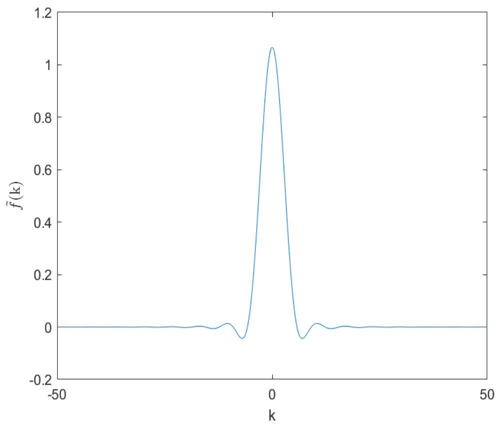

Figure 7: $\tilde{f}(k)$

A comparison of solutions from numerical schemes for the $q = 0$ case and a superposition of solutions from numerical schemes and asymptotic approximations (2) for the $q = 1$ case under various t also plotted in Fig.8 and Fig.9. See below.

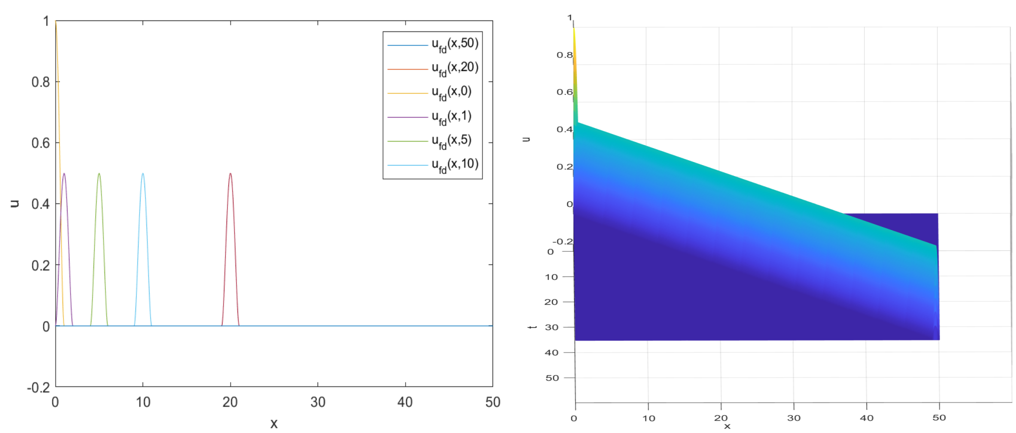

Figure 8: u_{fd} against x for various t=(0,1,5,10,20,50) and mesh(u_{fd}) along x direction (q=0)

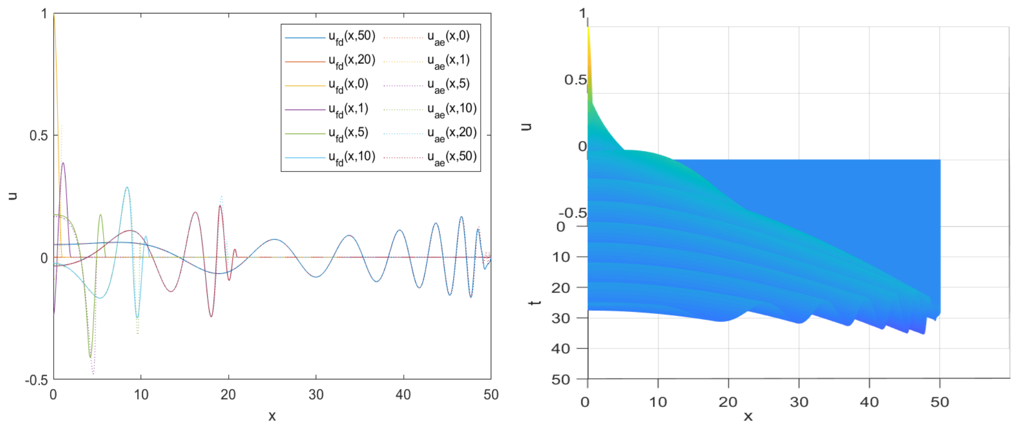

Figure 9: u_{fd}, u_{ae} against x for various t=(0,1,5,10,20,50) (zoom to see clearer) and mesh(u_{fd}) along x direction (q=1)

The asymptotic approximation is accurate when t is large but has slight deviation when t is small (see Fig.9) as expected. The solution u_{fd} at t=0 confirmed by the i.c.'s for both q=0 and q=1.

Both cases (q=0 and q=1) give bumps at the value x=t (q=0) or many bumps with less spacing as approaching x=t (q=1). u goes to zero soon after the bump (q=0) or at the last bump (q=1) (as shown by u_{ae}, despite a bit delay according to u_{fd}). I.e., **the bump moves in time marching direction**. In addition, both cases have their largest bumps at t=0.

In the case q=0, each bump (all towards up, so more like a pulse) has the same magnitude (except for the first half) and duration, i.e., the wave is travelling **undistorted**, with a symmetric pulse shape about the value of time the solution locates. The group velocity is equal to the phase velocity (**non-dispersive**) and the dispersion relation is reduced to $\Omega(k) = k$ ($k > 0$), i.e. proportional and linear in k. In the case q=1, bumps in one packet have various but smaller magnitudes towards up and down connecting one another (so more like oscillations in the packet), with changing frequency higher as time goes on. The outline connecting the local maximums or local minimums is sinusoidal-like, as indicated by the approximated solution in (2). This makes the solution a **Gaussian wave packet** moving forward in time marching direction while broadening, i.e., **distorted**.

When q is very small but non-zero, expect fewer oscillations with larger magnitudes and eventually only one of them left at the x value which equals the time when the solution is sliced, consistent with our interpretation of the group velocity, whereas now the group velocity is close to the phase velocity both with magnitude $\approx$

1 (possibly in opposite directions), i.e., the wave is almost undistorted as time goes on (non-dispersive). See Fig.10.

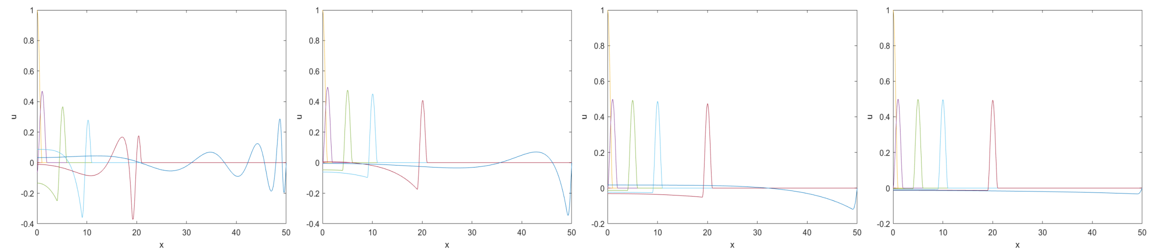

Figure 10: u_{fd} against x under various t for q=0.5, 0.2, 0.1, 0.05

1.4 Question 4

We now switch to solve (1) for $x, t > 0$ with i.c.'s and b.c.

$$u(x,0) = u_t(x,0) = 0 \tag{6}$$
$$u(0,t) = sin(\omega_0 t). \tag{7}$$

1.4.1 Solving for u(x,t) in the signalling problem: Programming Task

A modified program solving for u is listed on page 16, named **Q4fds(xmax,tmax,hx, ht,q,w0,Er)**. q, $w0$ are the parameters we need to input.

To find the analytic solution for $q = 0$, Laplace transform (1) w.r.t t and insert i.c's (6),

$$0 = \hat{u}_{tt} - \hat{u}_{xx}(x,p) = p^2\hat{u}(x,p) - pu(x,0) - u_p(x,0) - \hat{u}_{xx}(x,p) = p^2\hat{u}(x,p) - \hat{u}_{xx}(x,p).$$

Try solutions of the form $\hat{u} = e^{\beta x}$ to get $p^2 - \beta^2 = 0$, i.e., $\beta = \pm p$. Hence

$$\hat{u}(x,p) = \hat{C}(p)e^{px} + \hat{D}(p)e^{-px}.$$

Use convolution theorem to get

$$u(x,t) = \int_{-\infty}^{\infty} C(t-\tau)\delta(\tau + x) + D(t-\tau)\delta(\tau - x)\ d\tau = C(t+x) + D(t-x).$$

Let u_c be a complementary solution satisfying homogeneous b.c.'s and i.c's. Then

$$\begin{aligned} u(x,0) = 0 \quad &\Rightarrow \quad C(x) + D(-x) = 0 \\ u_t(x,0) = 0 \quad &\Rightarrow \quad C'(x) + D'(-x) = 0 \quad \Rightarrow \quad C(x) - D(-x) = const. \\ u(0,t) = 0 \quad &\Rightarrow \quad C(t) + D(t) = 0 \end{aligned}$$

I.e., $C(t+x) = -D(t-x) = const.$. Hence $u_c = 0$. Now try a particular solution $u_p(x,t) = H(t-x)sin(\omega_0(t-x))$ such that u_p solves (1) with homogeneous i.c's and b.c. (7). Check

$$\begin{aligned} u_p(x,0) &= -H(-x)sin(\omega_0 x) = 0 && \forall x > 0 \\ (u_p)_t(x,0) &= -\delta(-x)sin(\omega_0 x) + \omega_0 H(-x)cos(\omega_0 x) = 0 && \forall x > 0 \\ u_p(0,t) &= H(t)sin(\omega_0 t) = sin(\omega_0 t) && \forall t > 0 \end{aligned}$$

Hence for q=0 the general solution u satisfying required i.c.'s and b.c.'s is

$$u(x,t) = u_c(x,t) + u_p(x,t) = \begin{cases} H(t-x)sin(\omega_0(t-x)) & t \neq x \\ 0 & t = x \end{cases}$$

I.e.,

$$u(x,t) = sin(\omega_0(t-x))\mathbb{I}_{\{t>x\}}. \tag{8}$$

Plot this exact solution (8) with $\omega_0 = 0.9$ in Fig.11.

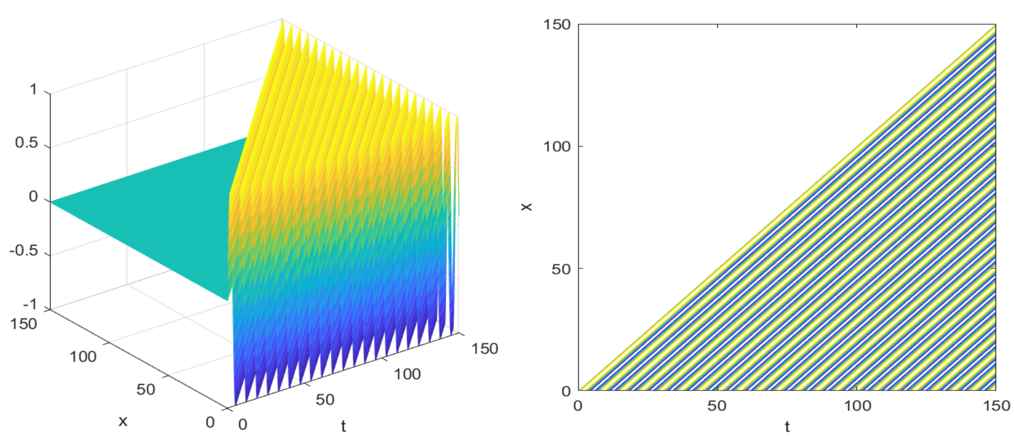

Figure 11: Mesh and contour plots of the exact solution u from (8) with $\omega_0 = 0.9$

Compare this solution with numerical solutions generated by the finite-difference scheme[§], a selection of which is shown in Fig.12, where the domain of x ($0 \leq x \leq 150$)

[§]Also checked the graph for q=0 to be the same as Fig.11.

is chosen such that the graph illustrates the general feature of the solution clear and is large enough to catch the evolution of it in the x-direction as parameters vary (given $t_{max} = 150$, expect interesting behavior of solutions only in the region $|x| < t$, similar to (8)), and the step lengths ($\Delta t = \Delta x = 0.2$) are chosen according to the desirable group velocity in order to build stable results in appearance while keep the processing time bearable.

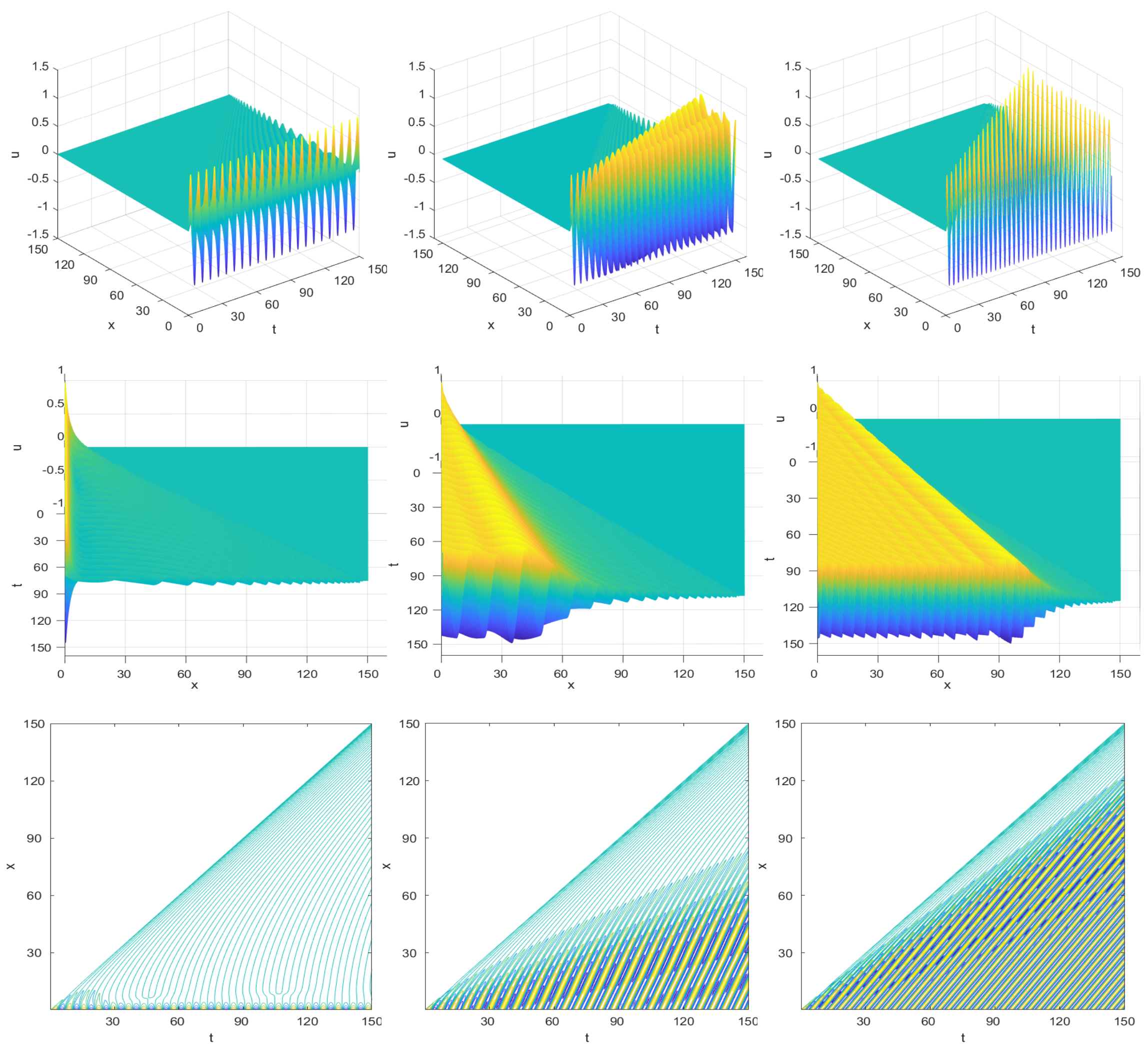

Figure 12: Mesh and contour plots of u_{fd} for $q = 1$ with $\omega_0 = 0.9, 1.1, 1.5$

From contour plots in Fig.12, we can see thin, stretched circles along the contours where which u is constant. The crests and troughs form an alternating sequence along the t direction, indicating sinusoidal solutions $-$ at any fixed instant of time, the solution varies sinusoidally along the x-axis, whereas at any fixed location on the x-axis the solution varies sinusoidally with time. As ω_0 increases, level curves of u become parallel to the $x = t$ direction, the contour plots becomes similar to Fig.11(ii), i.e., the numerical solution becomes a lot similar to the exact solution when $q = 0$, except for the denser level curves.

Note that when $q = 1$, $k_0 = \pm V/\sqrt{1-V^2}$. Given different ω_0, slices of the numerical solution u_{fd} against x at various t are plotted in Fig.13. In all three cases, the width of the wave packet increases in time. When $|x| > t$, the solution is exponentially small due to monotonicity of ϕ in (2). The group velocity is the same given fixed step lengths. (In the case when the group velocity $v_g \approx (150-100)/(150-100) = 1$, $k_0 \to \infty$, the wave is not heavily distorted in its general shape.) For the case

- $\omega_0 = 0.9 < 1$, the height of the wave packet is small due to strong cancellation between the sinusoid waves $-$ waves travel out of phase.
- $\omega_0 = 1.1, 1.5 > 1$, sinusoid waves cancel only at the end of the wave packet $-$ waves from the initial pulse travel at different phase velocities. As ω_0 increases, higher frequency results in narrower width of the bumps as well as smaller ranged cancellation.

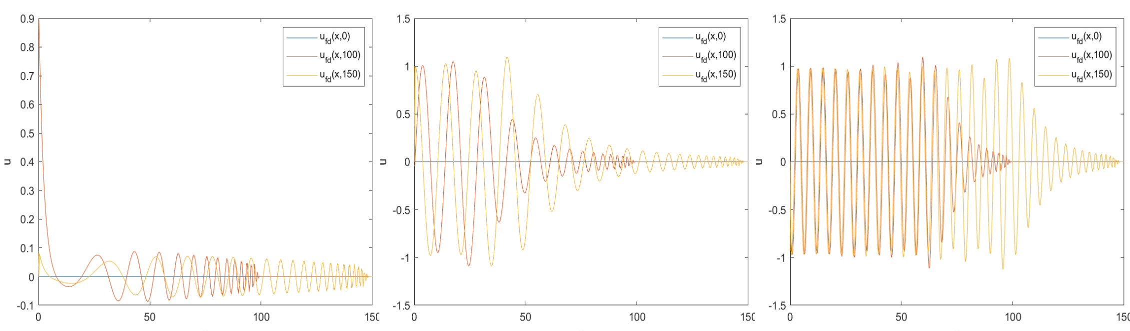

Figure 13: u_{fd} against x under various t=(0,100,150) for $\omega_0 = 0.9, 1.1, 1.5$ (q=1)

Similarly, in terms of phase velocity, plot u_{fd} against t at fixed x with $\omega_0 = 0.9, 1.1, 1.5$ in Fig.14. As ω_0 increases, the magnitude of the waves grows faster in time. This means that waves within one wave packet becomes in phase as time goes on, i.e., their phase velocities becomes similar.

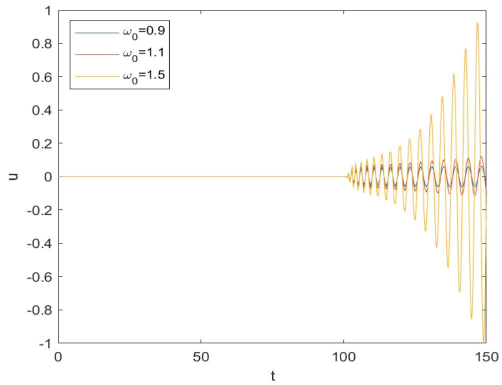

Figure 14: u_{fd} against t under x=100 for $\omega_0 = 0.9, 1.1, 1.5$ (q=1)

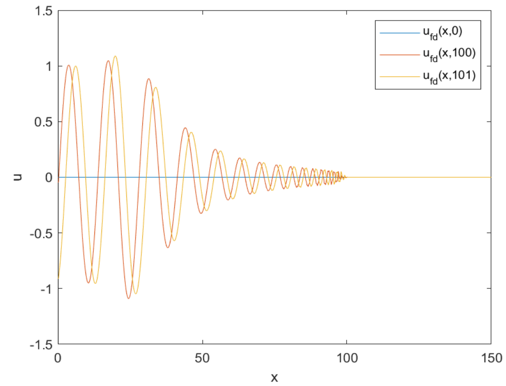

Figure 15: u_{fd} against x under t=(0,100,101) for $\omega_0 = 1.1$ (q=1)

In addition, can see from Fig.15 that the highest peak inside a wave packet move relatively to the right, indicating the phase velocity being in the same direction of the group velocity.

2 Programs

Note: Some programs listed on this pdf have *'return'* added after excessively long texts, which needs to be removed before testing.

2.1 Question 1

2.1.1 Velocities sketch

```
k=-10:0.05:10;q=1;
a=zeros(1,length(k))+1;b=zeros(1,length(k))-1;
vp=(q^2+k.^2).^(1/2)./k;
vg=k./(q^2+k.^2).^(1/2);
plot(k,vp,k,vg',k,a,'g--',k,b,'g--')
xlabel('k');ylabel('velocity');
legend('v_p=\Omega(k)/k','v_g=d\Omega(k)/dk','v=\pm1')
```

2.2 Question 3

2.2.1 Q3fds(xmax,tmax,hx,ht,q,Er)

```
function Q3fds(xmax,tmax,hx,ht,q,Er)
% hx=delta x, ht=delta t, Er=accuracy required to end the scheme
% extra 2's are added to Ux, Ut entries due to -1,0 entries
% e.g. Q3fds(50,50,0.05,0.05,1,0.0001)
Lx=xmax/hx+1;Lt=tmax/ht+1;
U0=zeros(Lx+1,Lt+1);
for i=2:Lx
    if (i-2)*hx<=1
        U0(i,2)=(1-((i-2)*hx)^2)^2;
    else
        U0(i,2)=0;
    end
end
for j=1:Lt+1
    U0(Lx+1,j)=0;
end
U0(1,2)=U0(3,2);
```

```
for i=2:Lx
    U0(i,1)=(U0(i+1,2)-2*U0(i,2)+U0(i-1,2))*ht^2/(2*hx^2)-
    q^2*ht^2/4*(U0(i+1,2)+U0(i-1,2))+U0(i,2);
    U0(i,3)=U0(i,1);
end
U0(1,1)=U0(3,1);U0(1,3)=U0(3,3);

U=U0;er=1;n=0;
while er>Er
    er=0;
    for j=3:Lt
        for i=2:Lx
            U(i,j+1)=ht^2*((U0(i+1,j)-2*U0(i,j)+U0(i-1,j))/(hx^2)-
            q^2*(U0(i+1,j)+U0(i-1,j))/2)+2*U0(i,j)-U0(i,j-1);
            d=abs(U(i,j+1)-U0(i,j+1));n=n+1;
            er=max(er,d);
        end
        U(1,j+1)=U(3,j+1);
    end
    U0=U;
end
U=U(2:Lx+1,2:Lt+1);
T=0:ht:tmax;
X=0:hx:xmax;
subplot(1,2,1)
mesh(U)% change to mesh(T,X,U) where needed
set(gca,'XTick',0:tmax/(ht*5):tmax/ht);
set(gca,'YTick',0:xmax/(hx*5):xmax/hx);
set(gca,'XTicklabel',{0:tmax/5:tmax});
set(gca,'YTicklabel',{0:xmax/5:xmax});
xlabel('t');ylabel('x');zlabel('u');
subplot(1,2,2)
contour(U)
xlabel('t');ylabel('x')
set(gca,'XTick',0:tmax/(ht*5):tmax/ht);
set(gca,'YTick',0:xmax/(hx*5):xmax/hx);
set(gca,'XTicklabel',{0:tmax/5:tmax});
```

```
set(gca,'YTicklabel',{0:xmax/5:xmax});
end
```

2.2.2 Plot $\tilde{f}(k)$

```
k=-50:0.05:50;
f=-16*sin(k)./k.^3-48*cos(k)./k.^4+48*sin(k)./k.^5;
plot(k,f)
xlabel('k');ylabel('$\tilde{f}(k)$','Interpreter','latex')
```

2.2.3 Plot u_{ae} in (2) with q=1, u in (4) with q=0 and u in (5) with q=1

```
%plot u(x,t) in (2), q=1
xmax=50;tmax=50;hx=0.05;ht=0.05;q=1;
x=0:hx:xmax;t=0:ht:tmax;
[X,T]=meshgrid(x,t);L=length(x);
U=zeros(L);
for i=1:L
    for j=1:L
        k=q*x(i)/(t(j)^2-x(i)^2)^(1/2);
        f=-16*sin(k)/k^3-48*cos(k)/k^4+48*sin(k)/k^5;
        U(i,j)=q^(1/2)*t(j)*f*cos(q*(t(j)^2-x(i)^2)^(1/2)+pi/4)/((2*pi)^
        (1/2)*(t(j)^2-x(i)^2)^(3/4))*(abs(x(i))<t(j));
    end
end
subplot(1,2,1)
mesh(t,x,U)
xlabel('t');ylabel('x');zlabel('u');
subplot(1,2,2)
contour(t,x,U)
xlabel('t');ylabel('x')

%plot u(x,t) in (4), q=0
xmax=50;tmax=50;hx=0.05;ht=0.05;
x=0:hx:xmax;t=0:ht:tmax;
[X,T]=meshgrid(x,t);L=length(x);
U=zeros(L);
```

```
for i=1:L
    for j=1:L
        U(i,j)=((1-(x(i)-t(j))^2)^2*(abs(x(i)-t(j))<=1)+(1-(x(i)+t(j))^2)^
        2*(abs(x(i)+t(j))<=1))/2;
    end
end
mesh(U)
set(gca,'XTick',0:tmax/(ht*5):tmax/ht);
set(gca,'YTick',0:xmax/(hx*5):xmax/hx);
set(gca,'XTicklabel',{0:tmax/5:tmax});
set(gca,'YTicklabel',{0:xmax/5:xmax});
xlabel('t');ylabel('x');zlabel('u');

%plot u(x,t) in (5), q=1 using Bessel function
%at specific point
q=1;t=25.2;x=25;
syms y
a1=(abs(x-t)<=1);
a2=(abs(x+t)<=1);
if imag(-asin((1+x)/t))==0
    b1=max(-pi/2,-asin((1+x)/t));
else
    b1=-pi/2;
end
if imag(asin((1-x)/t))==0
    b2=min(pi/2,asin((1-x)/t));
else
    b2=pi/2;
end
J=int(besselj(1,q*t*cos(y))*(1-(x+t*sin(y))^2)^2,b1,b2);
U=((1-(x-t)^2)^2*a1+(1-(x+t)^2)^2*a2-q*t*J)/2
plot(x,U,'*')
```

2.2.4 Q3fdsii to produce slices u against x

```
function Q3fdsii(xmax,tmax,hx,ht,q,T,Er)
% plot finite scheme with u against x w.r.t. different t
% hx=delta x, ht=delta t, Er=accuracy required to end the scheme
```

```
% extra 2 and extra 3 added to Ux, Ut entries since -1,0,
% (and for Ut eliminating the last 0 column)
% e.g. Q3fdsii(50,50,0.05,0.05,1,[0,1,5,10,20,50],0.0001)
Lx=xmax/hx+1;Lt=tmax/ht+2;
U0=zeros(Lx+1,Lt+1);
for i=2:Lx
    if (i-2)*hx<=1
        U0(i,2)=(1-((i-2)*hx)^2)^2;
    else
        U0(i,2)=0;
    end
end
for j=1:Lt+1
    U0(Lx+1,j)=0;
end
U0(1,2)=U0(3,2);

for i=2:Lx
    U0(i,1)=(U0(i+1,2)-2*U0(i,2)+U0(i-1,2))*ht^2/(2*hx^2)-q^2*ht^2/4*
    (U0(i+1,2)+U0(i-1,2))+U0(i,2);
    U0(i,3)=U0(i,1);
end
U0(1,1)=U0(3,1);U0(1,3)=U0(3,3);

U=U0;er=1;n=0;
while er>Er
    er=0;
    for j=3:Lt
        for i=2:Lx
            U(i,j+1)=ht^2*((U0(i+1,j)-2*U0(i,j)+U0(i-1,j))/(hx^2)-
            q^2*(U0(i+1,j)+U0(i-1,j))/2)+2*U0(i,j)-U0(i,j-1);
            d=abs(U(i,j+1)-U0(i,j+1));n=n+1;
            er=max(er,d);
        end
        U(1,j+1)=U(3,j+1);
    end
    U0=U;
end
```

```
U=U(2:Lx+1,2:Lt+1);Us=zeros(Lx,length(T));X=0:hx:xmax;
for k=1:length(T)
    Ts=T(k)/ht+1;K=q*X./(Ts^2-X.^2).^(1/2);
    Us(:,k)=U(:,Ts);
    plot(X,Us);hold on;
end
legend('u(x,T(1))','u(x,T(2))','u(x,T(3))','u(x,T(4))','u(x,T(5))',
'u(x,T(6))');
%edit the legend later after checking which is which
%OR after changing length of T
xlabel('x');ylabel('u');
end
```

2.2.5 Plot u_{ae} in (2) with q=1, u in (4) with q=0 and u in (5) with q=1

```
\subsection{Question 4}
\subsubsection{Q4(L,x)}
\begin{lstlisting}
function Q4(L,x)
% L for S_L and for R_{L,L}
% e.g. Q4(20,1)
XX=0:L;R=zeros(1,L+1);R(1)=1;S=R;i=1;j=1;k=1;
while i<=L
    c=factorial(i)*(-x)^i;
    S(i+1)=S(i)+c; i=i+1;
end
while k<=L
% content
i=1; C=zeros(1,2*k+1); C(1)=1;
while i<=2*k
    C(i+1)=factorial(i)*(-1)^i;
    i=i+1;
end
i=1; A=zeros(k);
while i<=k
    A(1:k,i)=C(k+2-i:2*k+1-i).';
    i=i+1;
end
```

```
Q=mldivide(A,-C(k+2:2*k+1).');
i=1; B=zeros(k+1,k);
while i<=k
     B(i+1:k+1,i)=C(1:k+1-i);
     i=i+1;
end
P=(C(1:k+1).'+B*Q).';
X=zeros(1,k+1);
i=0;
while i<=k
     X(i+1)=x^i;
     i=i+1;
end
R(j+1)=(P*X.')/(1+X(2:k+1)*Q);
j=j+1; k=k+1;
end
% end of content
plot(XX,R,XX,S)
xlabel('L');ylabel('estimates');
legend('R_{L,L}','S_L')
end
```

2.2.6 Plot slices u_{ae} in (2) against x

```
function Q3u_x(t)
% t in T=[0,1,5,10,20,50]
% use this for superposition with u_{fd}
q=1;xmax=50;hx=0.05;
x=0:hx:xmax;
U=zeros(1,length(x));
for i=1:length(x)
    k=q*x(i)/(t^2-x(i)^2)^(1/2);
    f=-16*sin(k)/k^3-48*cos(k)/k^4+48*sin(k)/k^5;
    U(i)=q^(1/2)*t*f*cos(q*(t^2-x(i)^2)^(1/2)+pi/4)/((2*pi)^(1/2)*
    (t^2-x(i)^2)^(3/4))*(abs(x(i))<t);
end
plot(x,U,':')
end
```

2.3 Question 4

2.3.1 Q4fds(xmax,tmax,hx,ht,q,w0,Er)

```
function Q4fds(xmax,tmax,hx,ht,q,w0,Er)
% hx=delta x, ht=delta t, Er=accuracy required to end the scheme
% extra 2 and extra 2 added to Ux, Ut entries due to -1,0 entries
% e.g. Q4fds(150,150,0.2,0.2,1,1.1,0.0001)
Lx=xmax/hx+1;Lt=tmax/ht+1;
U0=zeros(Lx+1,Lt+1);
for j=2:Lt+1
    U0(2,j)=sin(w0*(j-2)*ht);
end
for i=1:Lx+1
    U0(i,2)=0;
end
U0(2,1)=U0(2,3);

for i=2:Lx
    U0(i,1)=(U0(i+1,2)-2*U0(i,2)+U0(i-1,2))*ht^2/(2*hx^2)-q^2*ht^2/4*
    (U0(i+1,2)+U0(i-1,2))+U0(i,2);
    U0(i,3)=U0(i,1);
end
for j=3:Lt
    U0(1,j)=(hx^2*(U0(2,j+1)-2*U0(2,j)+U0(2,j-1))+2*U0(2,j)*ht^2)/
    (2*ht^2-(q*ht*hx)^2);
    U0(3,j)=U0(1,j);
end
U0(1,1)=U0(3,1);

U=U0;er=1;
while er>Er
    er=0;
    for j=3:Lt
        for i=4:Lx
            U(i,j+1)=ht^2*((U0(i+1,j)-2*U0(i,j)+U0(i-1,j))/(hx^2)-
            q^2*(U0(i+1,j)+U0(i-1,j))/2)+2*U0(i,j)-U0(i,j-1);
            d=abs(U(i,j+1)-U0(i,j+1));
            er=max(er,d);
```

```
        end
    end
    U0=U;
end
U=U(2:Lx+1,2:Lt+1);
T=0:ht:tmax;
X=0:hx:xmax;
subplot(1,2,1)
mesh(U)% change to mmesh(T,X,U) where needed
set(gca,'XTick',0:tmax/(ht*5):tmax/ht);
set(gca,'YTick',0:xmax/(hx*5):xmax/hx);
set(gca,'XTicklabel',{0:tmax/5:tmax});
set(gca,'YTicklabel',{0:xmax/5:xmax});
xlabel('t');ylabel('x');zlabel('u');
subplot(1,2,2)
contour(U)
xlabel('t');ylabel('x')
set(gca,'XTick',0:tmax/(ht*5):tmax/ht);
set(gca,'YTick',0:xmax/(hx*5):xmax/hx);
set(gca,'XTicklabel',{0:tmax/5:tmax});
set(gca,'YTicklabel',{0:xmax/5:xmax});
end
```

2.3.2 Plot u in (8) with $\omega_0 = 0.9$

```
xmax=150;tmax=150;hx=0.2;ht=0.2;
x=0:hx:xmax;t=0:ht:tmax;w0=0.9;
[T,X]=meshgrid(t,x);L=length(x);
U=zeros(L);
for i=1:L
    for j=1:L
      U(i,j)=sin(w0*(t(j)-x(i)))*(t(j)-x(i)>0);
    end
end
subplot(1,2,1)
mesh(T,X,U)
xlabel('t');ylabel('x');
subplot(1,2,2)
```

```
contour(T,X,U)
xlabel('t');ylabel('x');
```

2.3.3 Q4fdsii to produce slices u against x

```
function Q4fdsii(xmax,tmax,hx,ht,q,w0,T,Er)
% hx=delta x, ht=delta t, Er=accuracy required to end the scheme
% extra 2 and extra 3 added to Ux, Ut entries since -1,0,
% (and for Ut eliminating the last 0 column)
% e.g.  Q4fdsii(150,150,0.2,0.2,1,1.1,[0,100,150],0.0001)
Lx=xmax/hx+1;Lt=tmax/ht+1;
U0=zeros(Lx+1,Lt+1);
for j=2:Lt+1
    U0(2,j)=sin(w0*(j-2)*ht);
end
for i=1:Lx+1
    U0(i,2)=0;
end
U0(2,1)=U0(2,3);

for i=2:Lx
    U0(i,1)=(U0(i+1,2)-2*U0(i,2)+U0(i-1,2))*ht^2/(2*hx^2)-q^2*ht^2/4*
    (U0(i+1,2)+U0(i-1,2))+U0(i,2);
    U0(i,3)=U0(i,1);
end
for j=3:Lt
    U0(1,j)=(hx^2*(U0(2,j+1)-2*U0(2,j)+U0(2,j-1))+2*U0(2,j)*ht^2)/
    (2*ht^2-(q*ht*hx)^2);
    U0(3,j)=U0(1,j);
end
U0(1,1)=U0(3,1);

U=U0;er=1;
while er>Er
    er=0;
    for j=3:Lt
        for i=4:Lx
            U(i,j+1)=ht^2*((U0(i+1,j)-2*U0(i,j)+U0(i-1,j))/(hx^2)-
```

```
            q^2*(U0(i+1,j)+U0(i-1,j))/2)+2*U0(i,j)-U0(i,j-1);
            d=abs(U(i,j+1)-U0(i,j+1));
            er=max(er,d);
        end
    end
    U0=U;
end
U=U(2:Lx+1,2:Lt+1);Us=zeros(Lx,length(T));X=0:hx:xmax;
Ts1=T(1)/ht+1;Us(:,1)=U(:,Ts1);%edit this if change T
Ts2=T(2)/ht+1;Us(:,2)=U(:,Ts2);
Ts3=T(3)/ht+1;Us(:,3)=U(:,Ts3);
plot(X,Us(:,1),X,Us(:,2),X,Us(:,3))
legend('u_{fd}(x,T(1))','u_{fd}(x,T(2))','u_{fd}(x,T(3))')
%edit the legend later after checking which is which
%OR after changing length of T
xlabel('x');ylabel('u');
end
```