ENERGY: A time-dependent linear refraction model for a narrow banded wave field over 2D bathymetry and currents

Documentation and User's Manual

H. Tuba Özkan-Haller
College of Oceanic and Atmospheric Sciences
Oregon State University

January 24, 2005

Contents

List of Figures

1 Introduction

This document describes a model for the solution a time-dependent linear wave model based on the wave energy equation for a narrow-banded wave field. Wave-current interaction with depth-averaged currents (allowed to be time-dependent) is included through the consideration of the conservation of waves equations along with the linear dispersion relation in the presence of currents. The three time-dependent governing equations are solved using finite difference formulations for the spatial derivatives on a regular fixed grid and a high-order time integration scheme. The model domain extends from the shoreline to an arbitrary distance offshore. This wave model was originally developed as part of a combined wave-circulation model and information of the application of this model on the development and evolution of shear instabilities of the longshore current has been documented in *Özkan-Haller and Li* [2003] and the reader is referred to this document for a detailed discussion of the numerical methods used.

This document entails a user's manual for the purpose of walking the reader through the necessary steps to use the model for the prediction of the wave field over variable bathymetry and in the presence of ambient 2D currents. The model is provided in the form of a subroutine that functions as part of the NearCom system. In the following, first the flow chart of the model is discussed and the purpose of the individual modules is clarified. The input parameters for the program are then discussed. An example application, including the necessary input and the resulting output, are presented.

2 Model Formulation

In situations where the offshore wave field varies temporally due to wave grouping the wave energy near shore is expected to vary over the same time scales. Further, in the presence of a temporally varying circulation field, the wave energy is also expected to be time-dependent, even in situations where the offshore wave field does not display any variations (in the form of wave groups). We model this variation by assuming a narrow banded incident spectrum with Rayleigh-distributed wave heights and utilizing the time dependent energy equation for the incident waves [*Phillips*, 1982].

$$\begin{aligned}\frac{\partial E}{\partial t}+\frac{\partial(E(u'+c_g\cos\theta))}{\partial x'} &+ \frac{\partial(E(v'+c_g\sin\theta))}{\partial y'}\\ &+ S_{xx}\frac{\partial u'}{\partial x'}+S_{xy}\left(\frac{\partial u'}{\partial y'}+\frac{\partial v'}{\partial x'}\right)+S_{yy}\frac{\partial v'}{\partial y'}=-\epsilon_b. \qquad (1)\end{aligned}$$

Here, the x' and y' denote a right-handed coordinate axis for wave propagation with x' pointing onshore and y' pointing alongshore. Although such a coordinate system is often utilized in wave models, it is not the most natural system for many circulation models. Hence, in this model we adopt a coordinate system with x pointing offshore and y pointing alongshore to form a right-handed coordinate system. Hence, with domain lengths in the x and y directions of L_x and L_y, respectively, we use $x' = L_x - x$ and $y' = L_y - y$. The energy of the incident waves E is given by linear wave theory as $(1/8)\rho g H_{rms}^2$, θ and c_g are the local angle of incidence (measured counterclockwise from the x'-axis) and group velocity associated with the peak frequency relative to the current, respectively. The group velocity $c_g = nc$, where $c = \sigma/k$ is the wave celerity and

$$n=\frac{1}{2}\left[1+\frac{2kd}{\sinh 2kd}\right]. \qquad (2)$$

The variables u' and v' denote the cross-shore and longshore current components in the (x', y')-coordinate system and are given by $u' = -u$ and $v' = -v$. Note that the last three terms on the left hand side of (1) represent the wave-current interaction effects. Depending on their sign these terms allow for the loss of wave energy to the currents or, alternately, for the infusion of energy from the currents to the incident waves. The parameter ϵ_b represents the wave-averaged breaking dissipation and is modeled using the formulation of *Thornton and Guza* [1983] given by

$$\epsilon_b=\frac{3\sqrt{\pi}}{16}\frac{\rho g B^3 f_p}{\gamma^4 d^5}H_{rms}^7. \qquad (3)$$

Here, f_p is the peak frequency of a narrow banded spectrum with assumed Rayleigh distributed wave heights. The coefficients used for the wave height transformation model are $B = 0.78$ and $\gamma = 0.45$.

The initial condition of the wave energy E is computed assuming steady state and utilizing information on the incoming wave field at the initial time step. For this purpose, the current version of the code uses a low-order forward stepping scheme in the x-direction.

In the presence of a time-varying current field, kinematic properties of the incident wave field will also vary in time. We model the time variation of the wavenumber vector $\mathbf{k}$ associated with the peak frequency by using the conservation of waves equations [*Dean and Dalrymple*, 1991] for the x-component k_x $(= k\cos\theta)$ and the y-component k_y $(= k\sin\theta)$ given by

$$\frac{\partial k_x}{\partial t}+\frac{\partial \omega}{\partial x'}=0, \qquad \frac{\partial k_y}{\partial t}+\frac{\partial \omega}{\partial y'}=0, \qquad (4)$$

where $\omega = 2\pi/T = 2\pi f_p$ is the absolute radial wave frequency given by

$$(\omega - k_x u' - k_y v')^2 = gk\tanh kd. \qquad (5)$$

The initial condition for the wavenumber is computed using the refraction equation

$$\nabla' \times \mathbf{k} = 0 \qquad (6)$$

for the given bathymetry. Here,

$$\boldsymbol{\nabla}'() = \frac{\partial()}{\partial x'}\mathbf{i} + \frac{\partial()}{\partial y'}\mathbf{j} \tag{7}$$

is the horizontal gradient operator in the (x', y') coordinate system. Note that if (6) is initially satisfied, it will be satisfied throughout the simulation. This can be shown by analyzing $\boldsymbol{\nabla}' \times (eqn4)$, which yields

$$\boldsymbol{\nabla}' \times \left(\frac{\partial \mathbf{k}}{\partial t} + \boldsymbol{\nabla}'\omega \right) = 0. \tag{8}$$

Since by definition $\boldsymbol{\nabla}' \times \boldsymbol{\nabla}'\omega = 0$, this implies that

$$\frac{\partial}{\partial t} \left(\boldsymbol{\nabla}' \times \mathbf{k} \right) = 0 \tag{9}$$

stating that $\boldsymbol{\nabla}' \times \mathbf{k} = 0$ will hold throughout the simulation provided it holds initially.

The modified energy equation (1) and wavenumber equations (4) account for the shoaling of the incident waves on any component of the circulation field that directly opposes them. They also account for the refraction of the incident wave crests around any local features of the circulation field, such as narrow offshore directed current jets. Note that (4) implies that any temporal variation in the wavenumber components necessitates a spatial variation in the absolute wavenumber. This is because the wave field is not assumed to adjust instantaneously to the circulation field and the effects of current accelerations are included. For a discussion of the ramifications of a spatially varying absolute frequency pleas see *Özkan-Haller and Li* [2003].

The energy equation (1) is solved in time given an initial condition and an offshore boundary condition. Periodicity or wall-boundaries can be specified in the alongshore direction. The simulations are initiated using the steady state solution for the initial wave properties offshore; therefore, the wave height at the offshore boundary is not ramped. The spatial derivatives in the energy equation are computed using 4th order finite difference derivatives.

The wavenumber equations (4) are solved in conjunction with the dispersion relationship (5). We specify the wavenumber at the offshore boundary and use periodicity or wall boundaries in the alongshore direction. Theoretically, no boundary condition is necessary at the shoreline. However, we note here that the model requires the specification of a non-zero depth at the most shoreward grid point. We utilize 4th order derivatives to compute spatial derivatives in the longshore direction. However, in the cross-shore direction we utilize an upward differencing scheme to compute the spatial derivatives.

3 User's Manual

3.1 Program Outline and Flow Chart

The source code *energy.f* requires the presence of *definitions.inc* that contains the array definitions, *constants.inc* that contains specification of constants used in the model, and *parameter.inc* where the number of grid points in the x- and y-directions are defined.

1. *Wavemodule():* Main subroutine. Calls the subroutines *incon*, and *nextime*. Also controls input and output of variables by calling subroutines *adapt_circ_sedi*, *adapt_wave*, *opens*, *readin*, and *timeout*.

2. *adapt_circ_sedi:* Called by *Wavemodule*. Converts information passed from the master program on depth-averaged current velocities, wave-averaged surface elevation, and water depth to the variables used herein.

3. *adapt_wave:* Called by *Wavemodule*. Converts information regarding radiation stresses, a linear wave estimate of the mass flux due to waves, energy dissipation due to breaking, wave height, near-bottom velocity according to linear wave theory, group velocity with respect to the current, celerity, and the magnitude of the wavenumber to be passed to the master program. A rough estimate of breaking status at every grid point is also provided.

4. *incon:* Called by *Wavemodule*. Computes the initial values of the variables. Calls *scout* to echo input conditions. Calls the subroutine *refract2d* to compute wavenumber and angle of incidence for the incident waves at the initial time step. Also calls the subroutines *ddy2fd* and *shore* to compute the energy at the initial time step and variables related to the shoreline contour at the initial time step.

5. *refract2d:* Called by *incon*. Computes the variation of the cross-shore and longshore components of the wavenumber in space at the initial time step. Also computes the related refraction of an obliquely incident wave for 2D bathymetry. Calls subroutine *ddy2fd* to compute y-derivatives contained in the refraction equations.

6. *nextime:* Called by *okmodel*. Main model module. Calls subroutine *diff* to compute all the spatial derivatives in the governing equations. Updates unknowns to the new time step and calls subroutine *rollb* to replace the values at the current time level with the newly computed time level. Then, the subroutines *aliasx* and *aliasy* or *aliasy_w* are called to reduce the occurrence of high frequency noise.

7. *diff:* Called by *nextime*. Evaluates spatial derivatives in the governing equations using 4th order finite difference formulations in the longshore direction (calling subroutine *ddyfd*) and the cross-shore direction (calling subroutine *ddxfd*) or upward finite differencing (calling subroutine *ddxfd*).

8. *rollb:* Called by *nextime*. Replaces the new and current time levels.

9. *aliasy:* Called by *nextime*. Eliminates noise in the longshore direction from the computations for alongshore periodic domains so that no aliasing occurs. The algorithm is based on the method outlined by *Shapiro* [1970]. Calls subroutine *shapd1*. **Note:** *In the version provided, no variables are being actively filtered. However, experience shows that filtering may become necessary once the wave model is allowed to interact with a wave-induced circulation model.*

10. *aliasy_w:* Called by *nextime*. Eliminates noise in the longshore direction from the computations for domains with wall-boundaries in the alongshore direction so that no aliasing occurs. The algorithm is based on the method outlined by *Shapiro* [1970]. Calls subroutine *shapd1*. **Note:** *In the version provided, no variables are being actively filtered. However, experience shows that filtering may become necessary once the wave model is allowed to interact with a wave-induced circulation model.*

11. *aliasx:* Called by *nextime*. Eliminates noise in the cross-shore direction from the computations so that no aliasing occurs. The algorithm is based on the method outlined by *Shapiro* [1970]. Calls subroutine *shapd1*. **Note:** *In the version provided, no variables are being actively filtered. However, experience shows that filtering may become necessary once the wave model is allowed to interact with a wave-induced circulation model.*

12. *shapd1:* Called by *aliasx* and *aliasy*. Computes high wavenumber noise that should be eliminated so that no aliasing occurs.

13. *ddxfd1:* Called by *diff*. Computes x-derivative of a given two-dimensional function using upward finite differencing. Used for the differentiation of the cross-shore wavenumber.

14. *ddxfd:* Called by *diff*. Computes x-derivative of a given two-dimensional function using 4th order finite difference formulations.

15. *ddyfd:* Called by *diff*. Computes y-derivative of a given two-dimensional function using 4th order finite difference formulations.

16. *ddy2fd:* Called by *incon* and *refract2d*. Computes y-derivative of a given one-dimensional function using 4th order finite difference formulations.

17. *readin:* Called by *Wavemodule*. Reads input file *inp.dat*.

18. *scout:* Called by *incon*. Checks input values and prints values of input parameters onto screen.

19. *timeout:* Called by *Wavemodule*. Outputs time series at a specified point.

20. *opens:* Called by *Wavemodule*. Opens output files for time series and snapshots of energy or the wavenumber components.

21. *shore:* Called by *incon*. Initializes derivatives related to the computational grid.

3.2 Installation Instructions

The program should be installed as part of the NearCom system. An example Makefile is included linking this wave model to a mock circulation code (*circ.f*) that generates a narrow offshore-directed accelerating rip current for the example case discussed below.

3.3 Program Input

The input file *inp.dat* contains general information about the domain and the initial conditions. This file is also where the user specifies values for several switches. The file consists of a line that states the names of the variables that need to be specified in the following line. The definitions and possible values of the parameters are:

- `dt`: Time step (sec).
- `number`: Controls the output of snapshots. Values of the unknowns are printed to a file every `number` time steps.
- `nfirst`: In some situations it might be necessary to control the first occurrence of a spatial output in addition to the frequency of output dictated by `number`. `nfirst` dictates the first time spatial output is written.
- `nloc`: Number of points where a time series output is desired. Time series at a maximum of 20 points can be output.
- `dtint`: Time interval at which time series will be saved (sec). Can be different from model time step `dt`.
- `xpos, ypos`: Cross-shore and longshore coordinate (m) of location where time series are desired. Time series are output at the specified x-location. If no gridpoint exists at that location, linear interpolation is used to approximate the values at that location. On the other hand, the code picks the grid point closest to the specified y-location and outputs the time series at that point. This location is noted in the output file. If output at more than one location is needed, state the desired x- and y-location for each point in a separate line.
- `icase` controls the type of motion to be modeled.
  - `icase=1` and `icase=2`: No longer supported.
  - `icase=3`: Steady or unsteady waves, integration of the only energy equation is performed.
  - `icase=4`: Steady or unsteady waves, integration of the energy equation and wavenumber equations are performed.

  If no wave refraction around currents is required, choose `icease=3`. Otherwise, use `icase=4`.
- `iwall` determines the type of lateral boundary condition.
  - `iwall=1`: Periodic boundaries
  - `iwall=2`: Wall boundaries
- `iwc` determines if wave-current interaction is included.
  - `iwc=0`: Wave-current interaction excluded.

– iwc=1: Wave-current interaction included.

- lx: Domain length in the x-direction in (m).
- ly: Domain length in the y-direction in (m).
- wh: Wave height or rms wave height of incident waves at the offshore boundary (m).
- fbar: Frequency or peak frequency of incident waves (Hz).
- th0: Angle of incidence (°) of incident waves at the offshore boundary located at x=lx. Note that this angle is measured counterclockwise from the x'-axis (see discussion is Section 2).
- order_sw: The order of the wall boundary conditions at the alongshore boundaries. order_sw=2 is recommended.

3.4 Program Output

For the present version all output files are in ASCII. Although this inhibits optimum performance to a degree, it simplifies applications across different platforms. All output files are formatted to contain one column. MATLAB files are provided that show how post-processing of the files can be carried out.

The output files containing information about the grid points are:

- *y.dat*: Location of grid points in the longshore direction.
- *phi.dat*: Location of grid points in the cross-shore direction.

Snapshots of the flow field are saved in the following files. The files are in column format.

- *zeta.dat*: Snapshots of the cross-shore locations of the grid points.
- *hh.dat*: Snapshots of water depth h.
- *waveh.dat*: Snapshots of the incident wave height H_{rms}.
- *u.dat*: Snapshots of the cross-shore component of the current u. Note that this should be equal to the current provided by a circulation module.
- *v.dat*: Snapshots of the cross-shore component of the current u. Note that this should be equal to the current provided by a circulation module.
- *kx.dat*: Snapshots of the the cross-shore component of the incident wave wavenumber k_x.
- *ky.dat*: Snapshots of the the longshore component of the incident wave wavenumber k_x.
- *sig.dat*: Snapshots of the absolute frequency of the incident waves.
- *waveh.dat*: Snapshots of the incident wave height H_{rms}.

Time series at up to 20 locations are saved in column format, where each column corresponds to a location. The cross-shore and longshore coordinates of the location are noted in the first two rows of the output file.

- *wht.dat*:Time series of wave height H_{rms} at the physical locations specified in *inp.dat*.
- *kxt.dat*:Time series of the cross-shore wavenumber component k_x of the incident waves at the physical locations specified in *inp.dat*.
- *kyt.dat*:Time series of the longshore wavenumber component k_y of the incident waves at the physical locations specified in *inp.dat*.
- *sigt.dat*:Time series of the absolute frequency at the physical locations specified in *inp.dat*.

Finally, the file *ok.status* contains a log of the screen output.

4 Example Calculation: A wave field interacting with an accelerating jet

The program operates in two distinct modes. The chosen mode is identified using the switch `icase` in *inp.dat*. In the first mode (indicated by `icase=3`), computations are carried out accounting for a time varying incident wave field only. In this mode of operation, the wavenumber field in the domain is computed once in the beginning of the simulation is is assumed to be constant in time. Hence, no integration of the wavenumber equations is performed. Therefore, wave-current interaction due to only wave shoaling on the circulation is considered if `iwc=1` is chosen.

The more comprehensive mode (`icase=4`) involves the integration of all three equations stated in Section 2. In this mode, the time-dependent energy equation and the wavenumber equations are integrated in tandem. The effects of wave-current interaction due to wave shoaling and wave refraction on currents can be considered if `iwc=1`.

Below we provide an example case for a wave field propagating over an accelerating offshore directed jet. In order to repeat this case, choose $nx = 64$ and $ny = 128$ in *parameter.inc*, then compile as stated above. For this example an incoming wave with an rms wave height of 0.25m at a period of 6.25s at 10° incidence is specified at the offshore boundary of a barred beach profile. The offshore boundary is located at a water depth of 6m.

The input file *inp.dat* is as follows:

```
dt    number  nfirst
0.1   100   0
nloc  dtint                 # output at points
1     1.
xpos  ypos
75.    100.
icase iwall  iwc
4     1      1
lx   ly
225.  200.
wh    fbar  th0     # incident wave field at off-shore bndry
0.25    0.16   10.
order_sw   ( = 1 for 1st order; = 2 for 2nd order; = 4 for 4th order)
2
```

The resulting output can be viewed utilizing the MATLAB code *wave_en.m* given in the Appendix. The resulting figures are shown below. The specified current accelerates between $t = 50$s and $t = 150$s, and the effects of wave shoaling and refraction over opposing currents are evident. For further discussion of this example case, the reader is referred to *Özkan-Haller and Li* [2003] where a similar test case is discussed in detail.

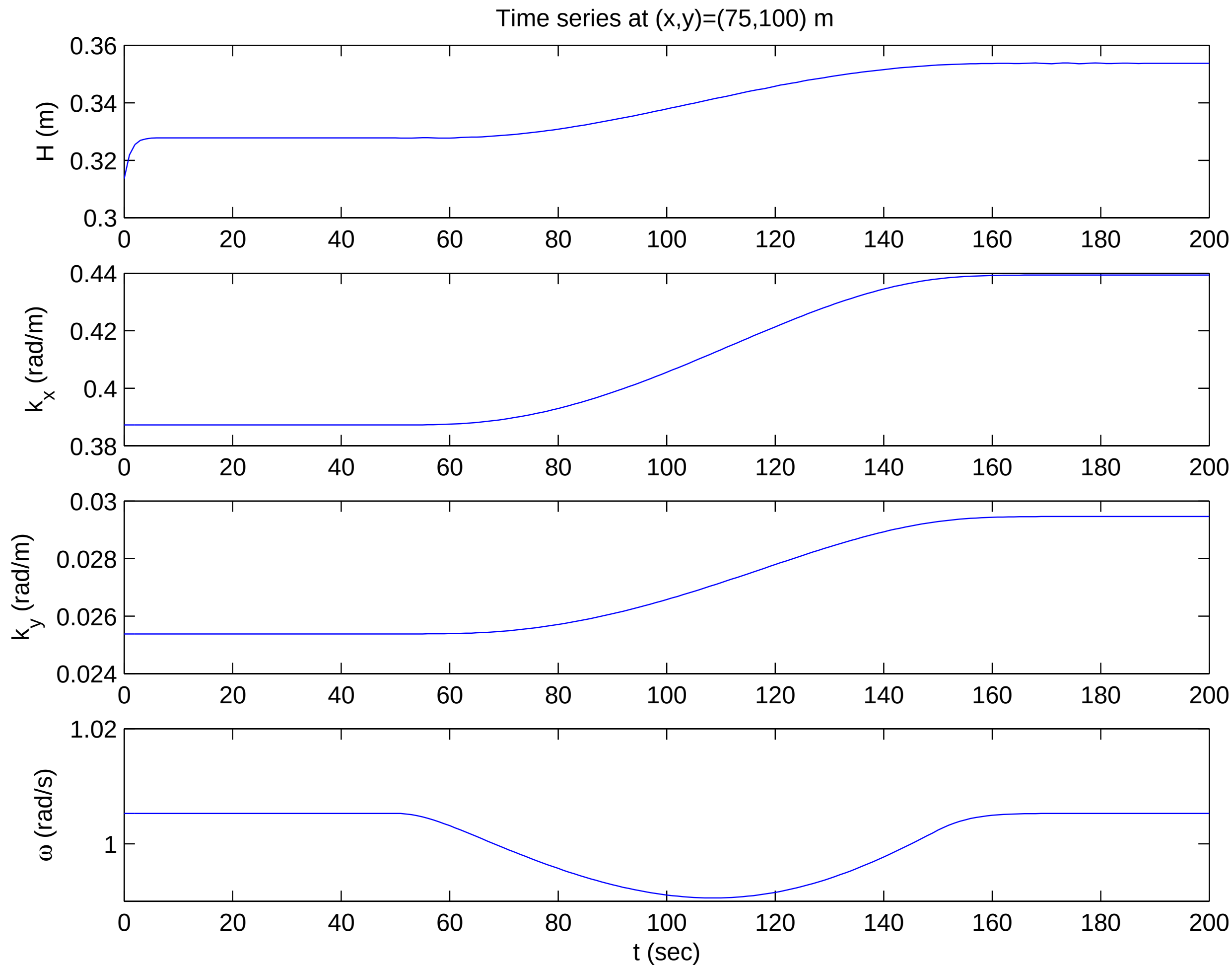

Figure 1: Time series of H_{rms}, k_x, k_y, and ω at the location specified in *inp.dat*.

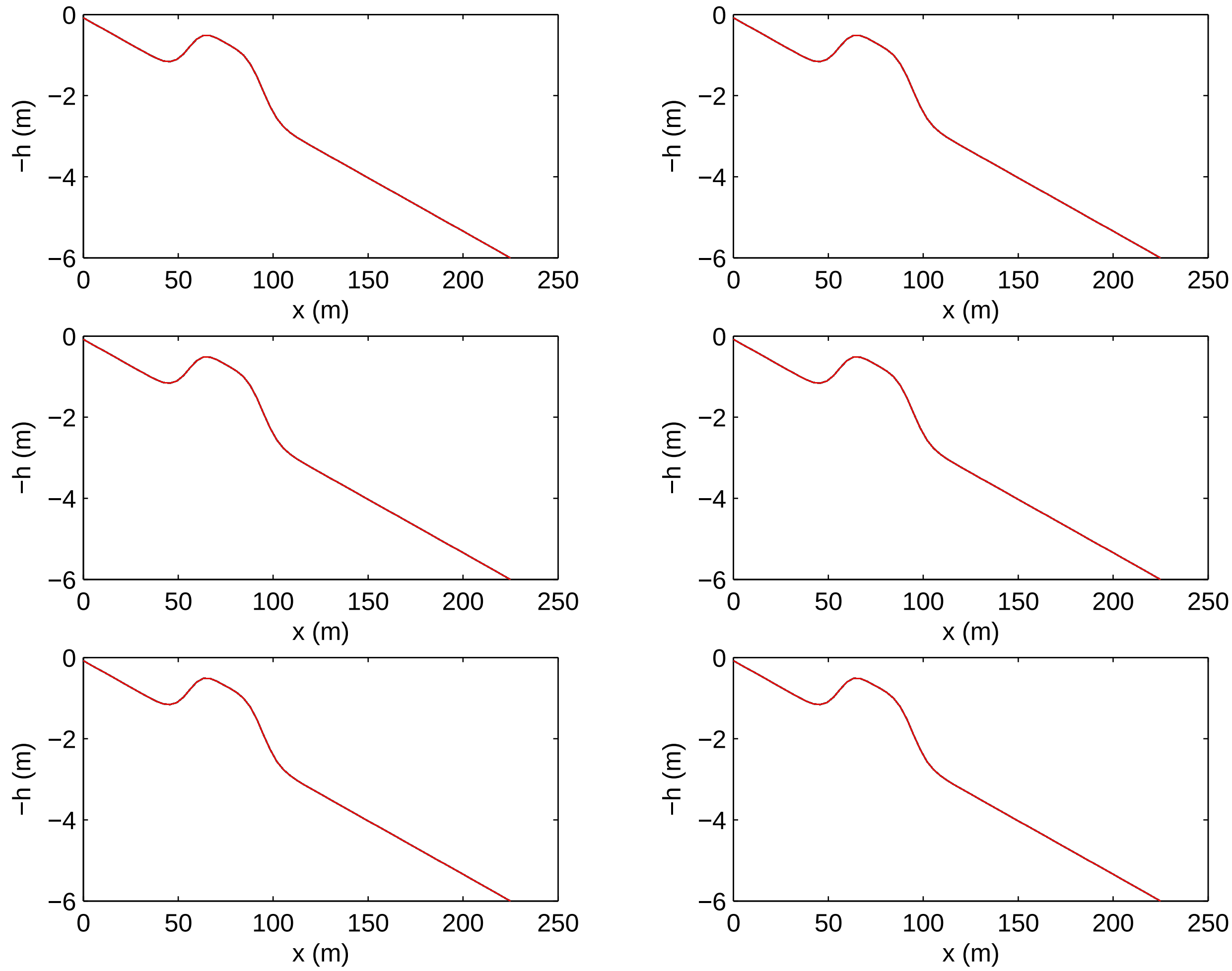

Figure 2: Snapshots of the water depth h at the first time level `nfirst` and at subsequent times determined by the output interval `number` specified in *inp.dat*.

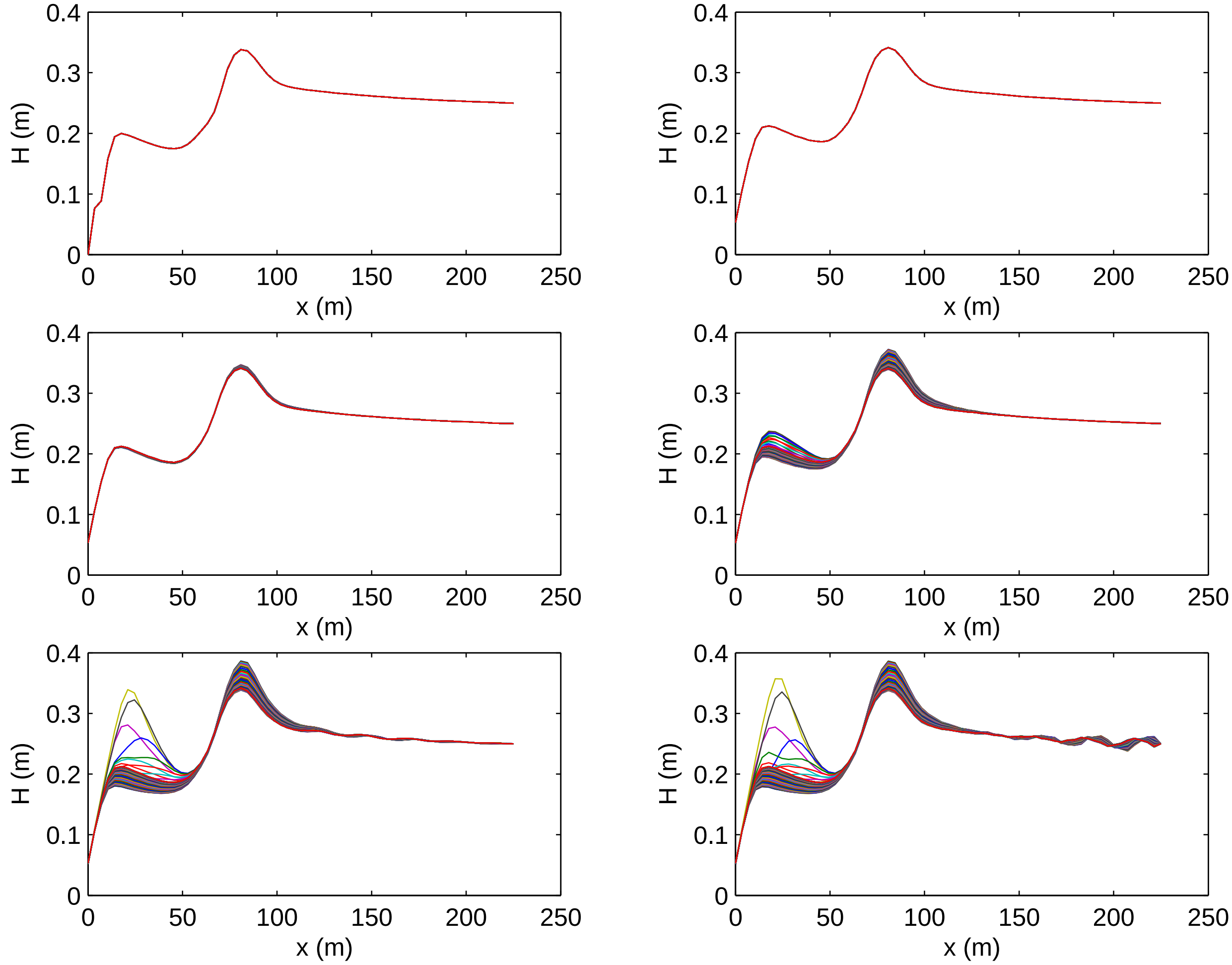

Figure 3: Snapshots of the wave height H_{rms} at the first time level `nfirst` and at subsequent times determined by the output interval `number` specified in *inp.dat*.

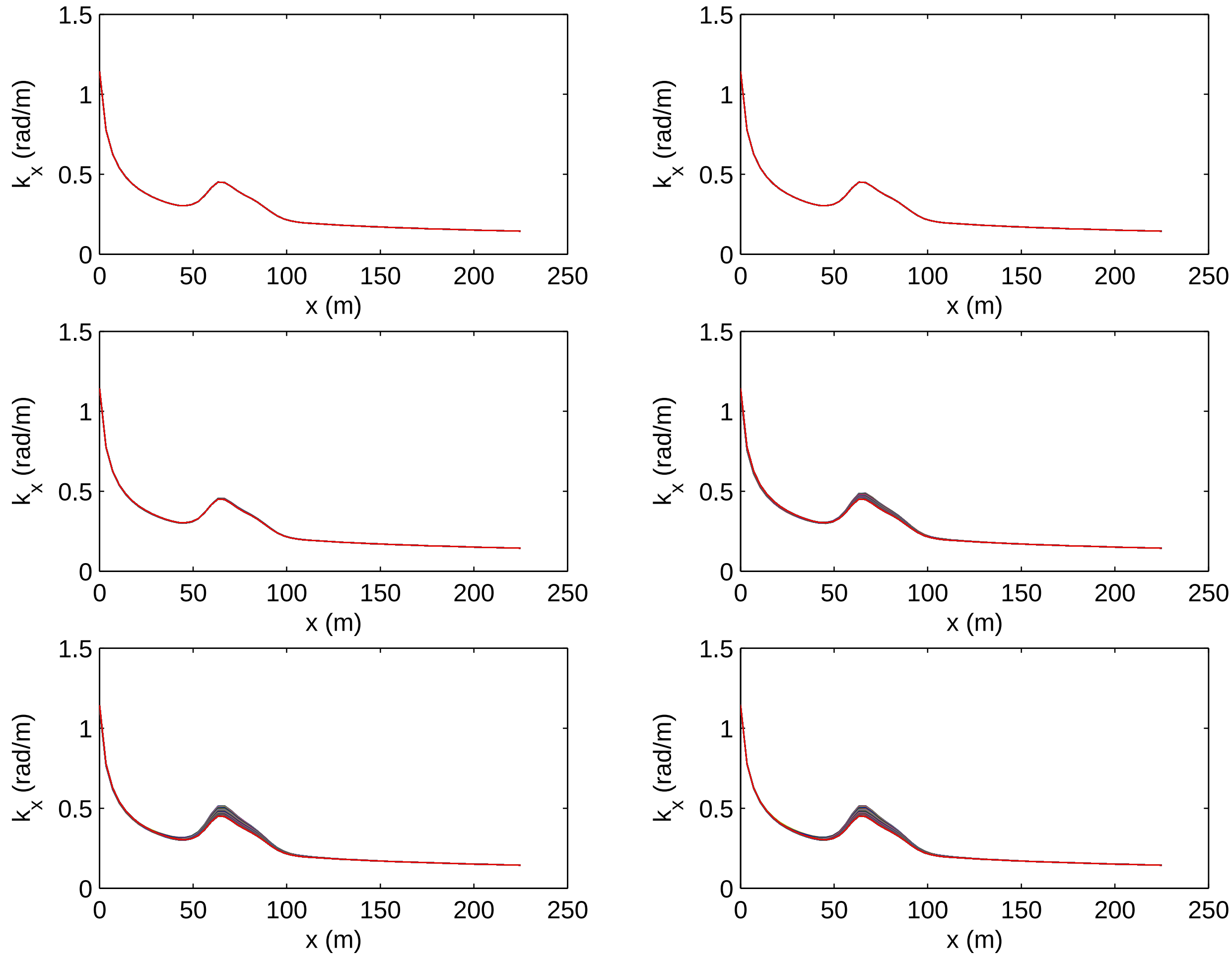

Figure 4: Snapshots of the cross-shore component of the wavenumber k_x at the first time level `nfirst` and at subsequent times determined by the output interval `number` specified in *inp.dat*.

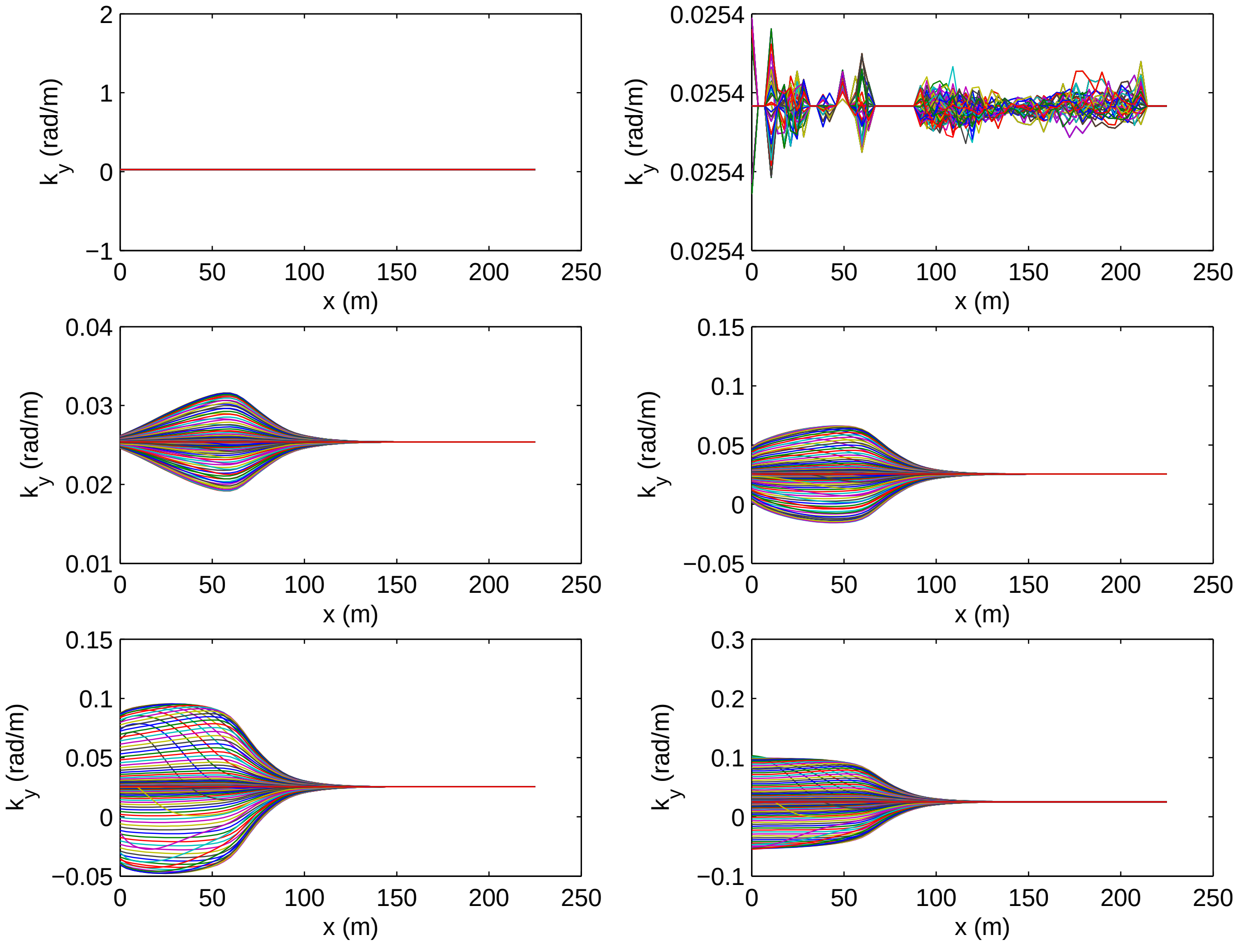

Figure 5: Snapshots of the the longshore component of the wavenumber k_y at the first time level `nfirst` and at subsequent times determined by the output interval `number` specified in *inp.dat*.

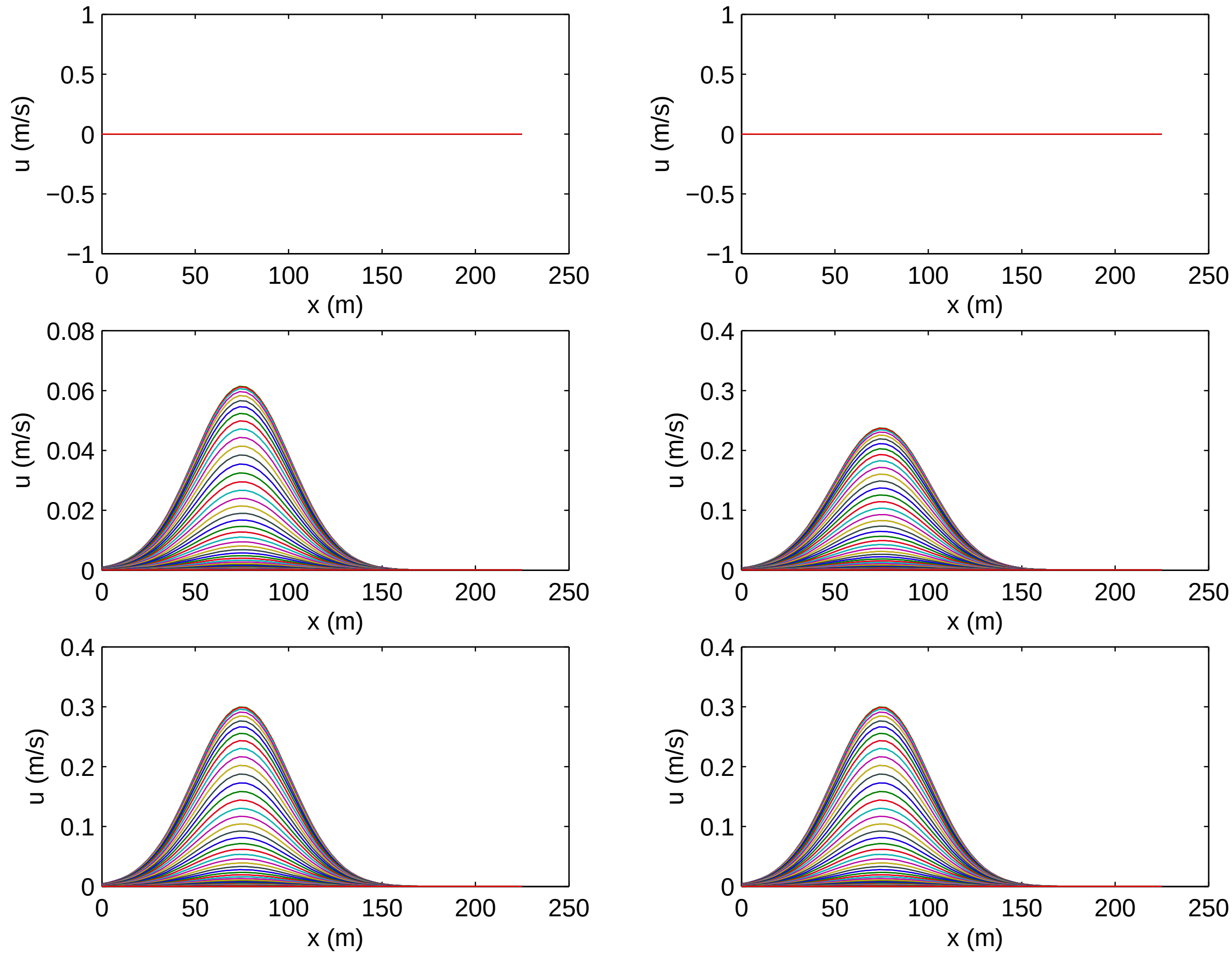

Figure 6: Snapshots of the the cross-shore current u (passed from the circulation module) at the first time level `nfirst` and at subsequent times determined by the output interval `number` specified in *inp.dat*.

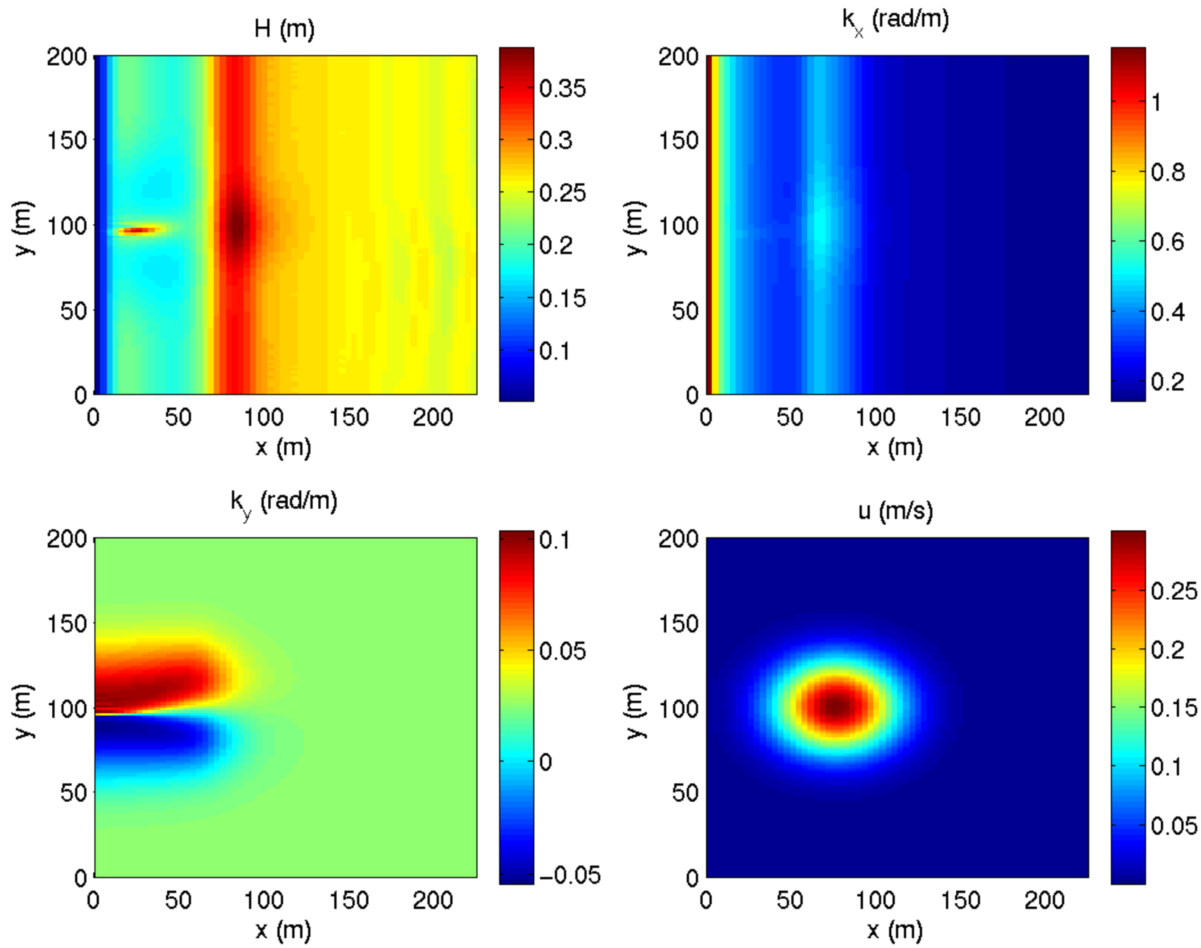

Figure 7: Snapshots of wave height H_{rms}, wavenumber components k_x and k_y, and cross-shore current u at the final time step.

5 References

Dean, R.G., and R.A. Dalrymple, *Water Wave Mechanics for Engineers and Scientists*, World Scientific, New Jersey, 1991.

Özkan-Haller, H.T., and Y. Li, , *J. of Geophys. Res.*, Effects of wave-current interaction on shear instabilities of longshore currents, *108(C5)*, 3139, 2003.

Phillips, O.M., *The Dynamics of the Upper Ocean*, Cambridge Univ. Press, New York, 1982.

Shapiro, R., Smoothing, filtering and boundary effects, *Reviews of Geophys. and Space Phys.*, *8*, 359–387, 1970.

Thornton, E.B., and R.T. Guza, Transformation of wave height distribution, *J. of Geophys. Res.*, *88*, 5925–5938, 1983.

6 Appendix

The MATLAB code *wave_en.m* used to plot the figures shown in Section 4 is given below.

```
load wht.dat
load kxt.dat
load kyt.dat
load sigt.dat

load inp.mt

figure(1)
dt=inp(5);
xloc=inp(6);
yloc=inp(7);
t=(0:dt:(length(wht)-3)*dt);
subplot(4,1,1),plot(t,wht(3:end)),ylabel('H (m)')
eval(['title(''Time series at (x,y)=(',int2str(xloc),',',int2str(yloc),') m '')'])
subplot(4,1,2),plot(t,kxt(3:end)),ylabel('k_x (rad/m)')
subplot(4,1,3),plot(t,kyt(3:end)),ylabel('k_y (rad/m)')
subplot(4,1,4),plot(t,sigt(3:end)),ylabel('\omega (rad/s)')
xlabel('t (sec)')

print -depsc test_fig1.eps

load zeta.dat

load kx.dat
load ky.dat
load waveh.dat
load hh.dat
load u.dat


load phi.dat
load y.dat
mt=length(phi);
myt=length(y);
[PHI,Y]=meshgrid(phi,y);

n=length(zeta);
nt=n/myt;
z=reshape(zeta,myt,nt)';
kx=reshape(kx,myt,nt)';
ky=reshape(ky,myt,nt)';
wh=reshape(waveh,myt,nt)';
h=reshape(hh,myt,nt)';
u=reshape(u,myt,nt)';


k=sqrt(kx.^2+ky.^2);

aa=size(z);
m=aa(1);
n=m/mt;
```

```
%ii=(14:2:n);
ii=(1:round(n/6):n);


orient portrait
for i=1:length(ii)
si=int2str(ii(i));
ns=((ii(i)-1)*mt)+1;
ne=(ii(i)*mt);
eval(['z',si,'(1:mt,1:myt)=z(ns:ne,1:myt);'])
eval(['kx',si,'(1:mt,1:myt)=kx(ns:ne,1:myt);'])
eval(['ky',si,'(1:mt,1:myt)=ky(ns:ne,1:myt);'])
eval(['k',si,'(1:mt,1:myt)=k(ns:ne,1:myt);'])

eval(['wh',si,'(1:mt,1:myt)=wh(ns:ne,1:myt);'])
eval(['h',si,'(1:mt,1:myt)=h(ns:ne,1:myt);'])
eval(['u',si,'(1:mt,1:myt)=u(ns:ne,1:myt);'])

end

figure(2)
in=0;
for i=1:1:length(ii)
in=in+1;
si=int2str(ii(i));
subplot(3,2,in)
eval(['plot(z',si,',-h',si,')']);
xlabel('x (m)'),ylabel('-h (m)')
end

print -depsc test_fig2.eps

figure(3)
in=0;
for i=1:1:length(ii)
in=in+1;
si=int2str(ii(i));
subplot(3,2,in)
eval(['plot(z',si,',wh',si,')']);
xlabel('x (m)'),ylabel('H (m)')
end

print -depsc test_fig3.eps

figure(4)
in=0;
for i=1:1:length(ii)
in=in+1;
si=int2str(ii(i));
subplot(3,2,in)
eval(['plot(z',si,',kx',si,')']);
xlabel('x (m)'),ylabel('k_x (rad/m)')
end
```

```
print -depsc test_fig4.eps

figure(5)
in=0;
for i=1:1:length(ii)
in=in+1;
si=int2str(ii(i));
subplot(3,2,in)
eval(['plot(z',si,',ky',si,')']);
xlabel('x (m)'),ylabel('k_y (rad/m)')
end

print -depsc test_fig5.eps

figure(6)
in=0;
for i=1:1:length(ii)
in=in+1;
si=int2str(ii(i));
subplot(3,2,in)
eval(['plot(z',si,',u',si,')']);
xlabel('x (m)'),ylabel('u (m/s)')
end

print -depsc test_fig6.eps

figure(7)
%Snapshots at the last step

subplot(2,2,1)
pcolor(z21',y,wh21'),
shading flat, axis equal, axis tight
xlabel('x (m)'),ylabel('y (m)')
title('H (m)')
colorbar

subplot(2,2,2)
pcolor(z21',y,kx21'),
shading flat, axis equal, axis tight
xlabel('x (m)'),ylabel('y (m)')
title('k_x (rad/m)')
colorbar

subplot(2,2,3)
pcolor(z21',y,ky21'),
shading flat, axis equal, axis tight
xlabel('x (m)'),ylabel('y (m)')
title('k_y (rad/m)')
colorbar

subplot(2,2,4)
pcolor(z21',y,u21'),
shading flat, axis equal, axis tight
xlabel('x (m)'),ylabel('y (m)')
```

```
title('u (m/s)')
colorbar

print -depsc test_fig7.eps
```