# User manual for the deterministic part of the PFL wave model

Mathias Klahn[1], Per A. Madsen[1] and David R. Fuhrman[1]

[1]*Department of Mechanical Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark*

## Contents

# 1 Introduction

This document describes the structure of the deterministic part of the PFL wave model. This part can be used to simulate steady nonlinear waves, the formation of crescent waves, the time evolution of a Gaussian hump which initially is at rest and, finally, the broad banded focusing events studied experimentally by Johannessen & Swan [4].

In the numerical simulations, a cartesian coordinate system is adopted with the horizontal $(x, y)$-plane located at the still water level and the $z$-axis pointing vertically upwards. The wave fields are discretized on a horizontal domain of size $L_x \times L_y$, and the water depth $h$ can take any finite value. The rapid solution of the Laplace problem inherent to the potential flow formalism is made possible by using, among other techniques, an artificial boundary condition, whose vertical position is $-b$, where $b$ is some positive length (typically satisfying $b/h \ll 1$). For a full description of the method and validation related to the four types of waves in the current release of the code, the reader is referred to the paper by Klahn et al. [5]. The remainder of this document simply serves the purpose of outlining the structure of the code and, perhaps most importantly, how it can be used.

# 2 Structure of the deterministic part

The overall structure of the deterministic part of the code is illustrated in figure 1. The code consists of a directory containing the source code and four simulation directories; one for each type of deterministic wave simulation. The simulation directories all have the same structure, which is illustrated for a generic case in figure 2. The simulation directories all contain a simulation program, which initializes the relevant wave field, integrates it in time and writes the result of the computation to file in the folder `OutputData`. The simulation directories also contain an analysis program, with which the files in `OutputData` can be read and analyzed. In the following, we elaborate on the structure of the simulation and analysis programs, and we describe the most important variables used in the programs.

## 2.1 Structure of the simulation programs

The simulation programs for steady nonlinear waves, the formation of crescent waves and the Gaussian hump initially at rest are all divided into six sections. These are:

1. Addition of the path of the source directory to the search path.

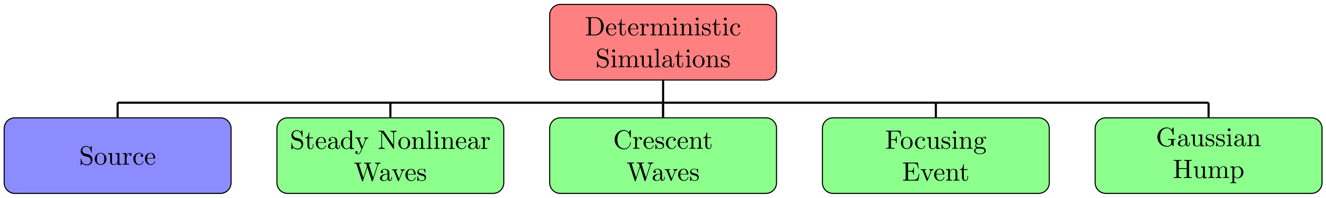

Figure 1: The overall structure of the code for the deterministic simulations. The blue source directory only contains subroutines, while the green simulation directories all contain files for the simulation of wave fields and the subsequent analysis.

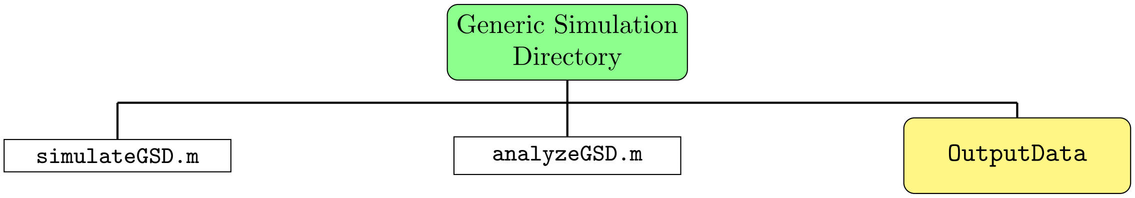

Figure 2: The structure of a generic simulation directory for the deterministic simulations. The white boxes represent Matlab files for the simulation and subsequent analysis of the relevant wave field. Output files are send to the yellow directory named `OutputData`.

2. Definition of non-dimensional physical and computational parameters.

3. Definition of fundamental dimensional physical scales (in SI units) and calculation of all other relevant dimensional quantities based on the non-dimensional quantities defined earlier.

4. Initialization of the surface elevation and the surface potential. At this stage, the vertical position of the artificial boundary is also calculated.

5. Calculation of time independent vectors and matrices which, among other things, are used for differentiation and preconditioning during the time integration.

6. Time integration of the wave field over the desired time interval and export of the results to the directory `OutputData`.

The structure of the simulation program for the focusing event is almost identical to this. The only difference is that the focusing event is directly defined in terms of dimensional variables, and the simulation program therefore skips section 2 above (its section 2 is the definition of dimensional physical parameters and non-dimensional computational parameters). The user of the simulation programs only needs to consider sections 1 and 2 and, if a different output is desired, also section 6.

## 2.2 Structure of the analysis programs

Like the simulation programs, the analysis programs are divided into sections. The number of sections varies with the type of wave field. Nevertheless, the first four sections of the analysis programs for steady nonlinear waves, crescent waves and the Gaussian hump at rest are conceptually the same. These four sections are:

1. Addition of the path of the source directory to the search path.

2. Definition of non-dimensional physical and computational parameters.

3. Definition of fundamental dimensional physical scales (in SI units) and calculation of all other relevant dimensional quantities based on the non-dimensional quantities defined earlier.

4. Loading the simulation results from file from the folder `OutputData`.

The definition of the non-dimensional variables is needed, as the files in the output directory are named after these to distinguish different cases. It should be noted that not all parameters are used in the file names by default. As such, the user should make sure that his/her parameter choices do not lead to different simulations but identical file names. The sections 5, 6,... perform the data analysis. The structure of the analysis program for the focusing event is similar to that of the other analysis programs, but it again skips section 2 above.

## 2.3 Important variables in the programs

Knowing the most important variables of the programs is always of great value to the user. Moreover, the current simulation and analysis programs will most certainly not do exactly what the user wants them to do, and for that reason it may also be important to know how to manipulate the variables. In this section we therefore very briefly describe the key variables in the programs and how these can be manipulated.

The horizontal dimensions of the computational domain is always discretized using `2*Nx` equidistant grid points in the $x$-direction and `2*Ny` equidistant grid points in the $y$-direction. The surface elevation and surface potential are represented by the column vectors `eta` and `Phis`, respectively, which both have length `2*Nx*2*Ny`. In the simulation programs, these quantities are combined into the single variable `y = [eta; Phis]`, and the values of `eta` and `Phis` at the `(nx,ny)`th grid point can thus be extracted using the lines of code

```
>> etanxny = y(nx+2*Nx*(ny-1))
>> Phisnxny = y(nx+2*Nx*(ny+2*Ny-1))
```

where 1 ≤ `nx` ≤ `2*Nx` and 1 ≤ `ny` ≤ `2*Ny`.

The vertical dimension of the computational domain (which is only relevant when solving the Laplace equation or otherwise dealing with quantities in the bulk of the water) is always discretized using `Ns+1` grid points which are denoted `s1`, `s2`, ..., `sNs+1`. These grid points are the quadrature points of a Legendre-Gauss-Lobatto quadrature of order `Ns`, and are ordered such that the first grid point in the vertical is `s1 = -b` (i.e. the location of the artificial boundary condition) and such that for the horizontal grid point with index `(nx,ny)` the last grid point in the vertical is `sNs+1 = etanxny`. The solution of the Laplace equation is denoted by `F` and is calculated in the source program `compws.m`. To extract its value at the `(nx,ny,ns)`th grid point one needs the line of code

```
>> Fnxnyns = F(nx+2*Nx(ny-1)+2*Nx*2*Ny*(ns-1))
```

where 1 ≤ `nx` ≤ `2*Nx`, 1 ≤ `ny` ≤ `2*Ny` and 1 ≤ `ns` ≤ `2*Ns`.

To differentiate the two-dimensional objects `eta` and `Phis` and the three-dimensional object `F` with respect to the spatial coordinates, one can use the vectors and matrices `Dx1`, `Dx2`, `Dy1`, `Dy2`, `Ds1` and `Ds2`. These are computed in section 5 of the simulation programs (section 4 for the focusing event). To calculate the values of the first $x$- and $y$-derivatives of `eta` on the horizontal grid, the lines of code

```
>> etax = reshape(ifft(Dx1.*fft(reshape(y(1:2*Nx*2*Ny),[2*Nx,2*Ny]))),[2*Nx*2*Ny,1])
>> etay = reshape(ifft(fft(reshape(y(1:2*Nx*2*Ny),[2*Nx,2*Ny]),2*Ny,2).*Dy1,2*Ny,2),[2*Nx*2*Ny,1])
```

should be executed. Note that the multiplication is done pointwise with the "`.*`"-operator. To calculate the second $x$- and $y$-derivatives of `eta`, `Dx1` and `Dy1` are simply replaced by `Dx2` and `Dy2`. Moreover, to calculate the $x$- and $y$-derivatives of `Phis`, the vector `y(1:2*Nx*2*Ny)` should be replaced by the vector `y(2*Nx*2*Ny+1:2*2*Nx*2*Ny)`. The velocities in the bulk of the water are obtained by differentiating the potential. This calculation is, however, a little tricky since the program works with the potential in the transformed coordinates, i.e. `F`, which is defined as $F(x, y, s(x, y, z)) = \Phi(x, y, z)$, where $\Phi(x, y, z)$ is the velocity potential in Cartesian coordinates (see eq. 13 of Klahn et al. for the definition of $s$). To calculate the components of the fluid velocity one first executes the lines of code

```
>> Fx  = reshape(ifft(Dx1.*fft(reshape(F,[2*Nx,2*Ny*(Ns+1)]))),[2*Nx*2*Ny*(Ns+1),1])
>> Fy  = reshape(ifft(fft(reshape(F,[2*Nx,2*Ny,Ns+1]),2*Ny,2).*Dy1, 2*Ny, 2),[2*Nx*2*Ny*(Ns+1),1])
>> Fs  = reshape(reshape(F,[2*Nx*2*Ny,Ns+1])*Ds1,[2*Nx*2*Ny*(Ns+1),1])
```

after which the fluid velocity in the $x$-direction is obtained using the line of code

```
>> u  = reshape(Fx-reshape(((etax./beta).*reshape(Fs,[2*Nx*2*Ny,Ns+1])).*s1Vec,
                [2*Nx*2*Ny*(Ns+1),1]), [2*Nx,2*Ny,Ns+1]),
```

the fluid velocity in the $y$-direction is obtained using the line of code

```
>> v  = reshape(Fy-reshape(((etay./beta).*reshape(Fs,[2*Nx*2*Ny,Ns+1])).*s1Vec,
                [2*Nx*2*Ny*(Ns+1),1]), [2*Nx, 2*Ny,Ns+1]),
```

and, finally, the fluid velocity in the $z$-direction is obtained using the line of code

```
>> w  = reshape(((2./beta).*reshape(Fs,[2*Nx*2*Ny,Ns+1])), [2*Nx,2*Ny,Ns+1]).
```

In these lines of code `beta = b+eta` and `s1Vec = sGrid+1`. It should be noted that `u`, `v` and `w` only contain the values of the fluid velocity at the grid points within the computational domain whose extent in the vertical dimension is limited to $-b < z < \eta$. For an example of how to calculate the fluid velocities for depths greater than $b$, the user of the program is referred to the subroutine `compVelProfile.m` in the source directory.

# 3 How to execute the deterministic simulations

As described in the previous section, the execution of the simulation and analysis programs requires that the user modifies these slightly. The path of the source code should be added to the search path should be added in the programs' section 1, and the physical and computational parameters should be declared in the programs' section 2. In the following, we explain the meaning of the (non)-dimensional parameters that the user must specify in order to execute the simulation programs. We also illustrate the output of the analysis programs.

## 3.1 Steady Nonlinear Waves

Steady nonlinear waves can be simulated by executing the program `simulateSNW.m`. For these simulations the horizontal domain is assumed to have dimensions $L_x = L_y = \lambda$, where $\lambda$ is the wavelength of the steady nonlinear wave, and the artificial boundary condition is chosen to be located at $z = -b = -\max(\eta)$, where $\eta$ is the surface elevation of the steady nonlinear wave and the maximum is taken over all grid points in the horizontal plane. To execute the program, the user must specify the following physical non-dimensional parameters in the program's section 2:

- `kh`: The non-dimensional water depth $kh$, where $k$ is the wave number and $h$ is the water depth.
- `relSteep`: The steepness of the wave relative to the maximum steepness $0.1401 \tanh\big(0.8863kh\big)$ as described by Battjes [1].
- `NPeriod`: The wave is integrated in time for the time interval `[0, NPeriod*T]`, where `T` is the wave period.

In addition, the user must specify the non-dimensional computational parameters (also in the program's section 2):

- `Nx`, `Ny` and `Ns`: The spatial resolution of the wave field. `2*Nx` grid points are used in the $x$-direction, `2*Ny` grid points are used in the $y$-direction, and `Ns+1` grid points are used in the vertical dimension. Note that the value of `Ns` is relevant only for the solution of the Laplace problem,
- `NStep`: The number of time steps every period.
- `dampStrat` and `dampCoef`: If `dampStrat = 1` the time integration is not damped. If, however, `dampStrat = 2` the time integration is damped in accordance with eq. (4) of Klahn et al. [5]. The parameter `dampCoef` equals the parameter $\alpha$ in eq. (4) of Klahn et al.
- `NSSGW`: The number of grid points used by the SSGW routine (see Clamond & Dutykh [2], [3]). Note that this number should be large (10,000 is typically a good number), as the initial condition is interpolated from the result SSGW (which is defined on a non-equidistant grid) to an equidistant grid.
- `epsGMRES`: The relative tolerance to which the set of linear equations corresponding to the discretized Laplace problem is solved by GMRES. Note that setting `epsGMRES = 1E-14` corresponds to solving the system of equations to double precision.

During the time integration, the wave motion is animated and the time in units of wave periods is written to the terminal. Once the time integration has finished, the simulation program writes the surface elevation and surface potential of the initial wave field (`y0`) and the final wave field (`y`) as well as the wave period (`T`) to file in the folder `OutputData`.

To analyze the results, the user should execute the analysis program `analyzeSNW.m`. To do so, the path to the source directory should be specified in section 1 and the same non-dimensional parameters as above should be declared in section 2. The analysis program for steady nonlinear waves in the current release of the code can be used to 1) compare the initial surface elevation with the final surface elevation and to 2) compute the vertical profile of the horizontal and vertical fluid velocity at a user-specified horizontal location. To that end, the user should specify the value of the dimensionless horizontal position (`xPos`, `yPos`) (with `xPos` being measured in units of `Lx` and `yPos` being measured in units of `Ly` such that they are both numbers between 0 and 1) in section 2 of the analysis program. Examples of output of the analysis program are shown in figure 3 for a wave with `kh = 1`, `relSteep = 0.8`, `NPeriod = 10`, `xPos = 0.25`, `yPos = 0.37`, `Nx = 32`, `Nx = 1`, `Ns = 20`, `NStep = 100`, `dampstrat = 2`, `dampCoef = 0.7`, `NSSGW = 10000` and `epsGMRES = 1E-9`.

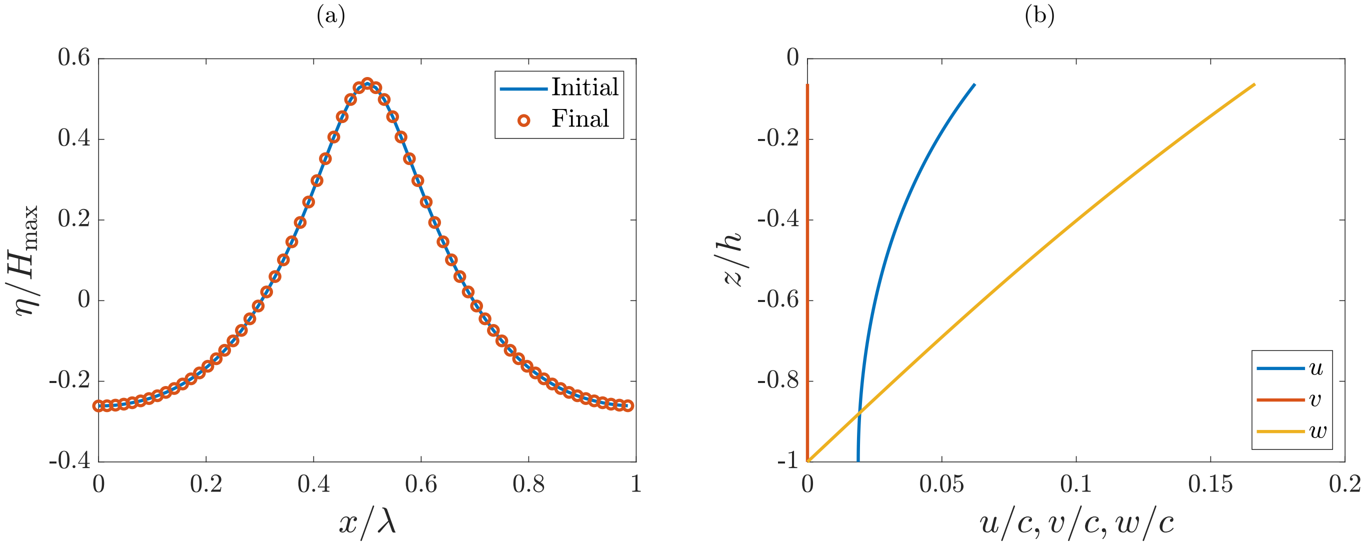

Figure 3: The output of `analyzeSNW.m` when used with the parameters `kh = 1`, `relSteep = 0.8`, `NPeriod = 10`, `xPos = 0.25`, `yPos = 0.37`, `Nx = 32`, `Nx = 1`, `Ns = 20`, `NStep = 100`, `dampstrat = 2`, `dampCoef = 0.7`, `NSSGW = 10000` and `epsGMRES = 1E-9`. (a) A comparison of the final surface elevation to the initial surface elevation ($\lambda$ is the wavelength and $H_{\max} = 0.1401\lambda \tanh\left(0.8863kh\right)$ is the maximum wave height). (b) The vertical profiles of the horizontal ($u$) and vertical ($w$) velocities ($z$ is the vertical coordinate with $z = 0$ at the still water level and $c = \lambda/T$ is the nonlinear wave celerity).

### 3.2 Formation of Crescent Waves

The formation of crescent waves can be simulated using the program `simulateCW.m`. These simulations are carried out by initializing a steady nonlinear wave (the so-called fundamental wave) and adding a small disturbance to it (see eq. (29) of Klahn et al. [5]) which grows with time. To execute the program, the user must specify the following physical non-dimensional parameters in the program's section 2:

- `kh`: The non-dimensional water depth $kh$, where $k$ is the wave number of the fundamental wave and $h$ is the water depth.
- `steep`: The steepness $H/\lambda$ (where $H$ is the wave height and $\lambda$ is the wavelength) of the fundamental wave.
- `wlX` and `wlY`: The size of the horizontal domain is `wlX*lambda` in the $x$-direction and `wlY*lambda/kq` (see below for the meaning of `kq`) in the $y$-direction.
- `NPeriod`: The wave field is integrated in time for the time interval `[0, NPeriod*T]`, where `T` is the period of the fundamental wave.
- `epsDist`: The scaling of the amplitude of the disturbance. This parameter corresponds to $\varepsilon$ in eq. (29) of Klahn et al. [5].
- `kp` and `kq`: The wave numbers of the disturbance (in units of $k$) in the $x$- and $y$-directions, respectively.

Moreover, the user must specify the non-dimensional computational parameters (also in the program's section 2):

- `Nx`, `Ny` and `Ns`: The spatial resolution of the wave field. `2*Nx` grid points are used in the $x$-direction, `2*Ny` grid points are used in the $y$-direction, and `Ns+1` grid points are used in the vertical dimension. Note that the value of `Ns` is relevant only for the solution of the Laplace problem,
- `NStep`: The number of time steps every period,
- `dampStrat` and `dampCoef`: If `dampStrat = 1` the time integration is not damped. If, however, `dampStrat = 2` the time integration is damped in accordance with eq. (4) of Klahn et al. [5]. The parameter `dampCoef` corresponds to the parameter $\alpha$ in eq. (4) of Klahn et al.
- `NSSGW`: The number of grid points used by the SSGW routine (see Clamond & Dutykh [2], [3]). Note that this number should be large (10,000 is typically a good number), as the initial condition is interpolated from the result SSGW (which is on a non-equidistant grid) to an equidistant grid,

- epsGMRES: The relative tolerance to which the set of linear equations corresponding to the discretized Laplace problem is solved by GMRES. Note that setting epsGMRES = 1E-14 corresponds to solving the system of equations to double precision,
- bParam: The vertical position of the artificial boundary condition is chosen as (1+bParam)*max(eta), where eta is the surface elevation of the fundamental wave.

During the time integration, the wave motion is animated and the time in units of periods of the fundamental wave is written to the terminal. Once the time integration has finished, the simulation program writes the time evolution of the surface elevation at $y = L_y/2$ (centerElevation), the time evolution of the most unstable Fourier coefficient of the surface elevation (coefTimeSeries) and the period of the fundamental wave (T) to file in the folder OutputData. Using the analysis program analyzeCW.m these results can be used to compute the geometrical ratios $h_{11}/h_{12}$, $h_{11}/h_{21}$ and $h_{21}/h_{22}$ (see figure 10 of Klahn et al. [5] for the definition of these quantities) and to compare the growth rate of the most unstable Fourier component of the surface elevation to that predicted by the analysis of McLean [6]. Examples of output of the analysis program are shown in figure 4 for a wave field with kh = 2*pi, steep = 0.105, NPeriod = 50, wlX = 4, wlY = 2, epsDist = 0.0001, kp = 3/2, kq = 1.23, Nx = 64, Nx = 16, Ns = 8, NStep = 50, dampstrat = 2, dampCoef = 0.5, NSSGW = 10000, epsGMRES = 1E-9 and bParam = 0.1.

Although the program can be executed for any finite water depth and wave steepness, the user should be aware that these quantities by default are defined through the values kh = 2*pi and steep = 0.105, respectively. Moreover, the user should know that the non-dimensional wave numbers of the disturbance are by default chosen as kp = 3/2 and kq = 1.23, respectively. These default values come from the fact that the analysis program (i.e. analyzeCW.m) only knows the theoretically expected growth rate (see McLean [6]) of the most unstable Fourier coefficient of the surface elevation for these parameters. If, say, a different water depth is used, the comparison of the simulated growth rate to the theoretical prediction will not make sense.

### 3.3 A Gaussian hump initially at rest

The time evolution of a Gaussian hump which initially is at rest can be simulated using the program simulateSNW.m. For these simulations the horizontal domain is assumed to have dimensions $L_x = L_y = L$. The initial condition for the surface elevation is obtained by evaluating the function

$$\eta(x, y, t) = H \exp\left(-\frac{(x - L/2)^2 + (y - L/2)^2}{\sigma^2}\right) \tag{1}$$

while the potential at the surface is simply set to a constant value such that the water initially is at rest. Length is measured in units of the horizontal domain size (i.e. $L$) and time is measured in units of $T_0 = \sigma/(gh)^{1/2}$. To execute the program, the user must specify the following physical non-dimensional parameters the program's section 2:

- hTilde: The water depth in units of the horizontal domain size $L$.
- sigmaTilde: The characteristic width, $\sigma$, of the Gaussian function (1) in units of the horizontal domain size $L$.
- HTilde: The height, $H$, of the Gaussian function (1) in units of the horizontal domain size $L$.
- NPeriod: The system is integrated in time for the time interval [0, NPeriod*T0], where T0 is the fundamental unit of time defined above.

In addition, the user must specify the non-dimensional computational parameters (also in the program's section 2):

- Nx, Ny and Ns: The spatial resolution of the wave field. 2*Nx grid points are used in the $x$-direction, 2*Ny grid points are used in the $y$-direction, and Ns+1 grid points are used in the vertical dimension. Note that the value of Ns is relevant only for the solution of the Laplace problem,
- NStep: The number of time steps per fundamental unit of time (T0). Note that a relatively small number of steps such as 10, say, is enough to obtain a fine temporal resolution.
- dampStrat and dampCoef: If dampStrat = 1 the time integration is not damped. If, however, dampStrat = 2 the time integration is damped in accordance with eq. (4) of Klahn et al. [5]. The parameter dampCoef corresponds to the parameter $\alpha$ in eq. (4) of Klahn et al.
- epsGMRES: The relative tolerance to which the set of linear equations corresponding to the discretized Laplace problem is solved by GMRES. Note that setting epsGMRES = 1E-14 corresponds to solving the system of equations to double precision.

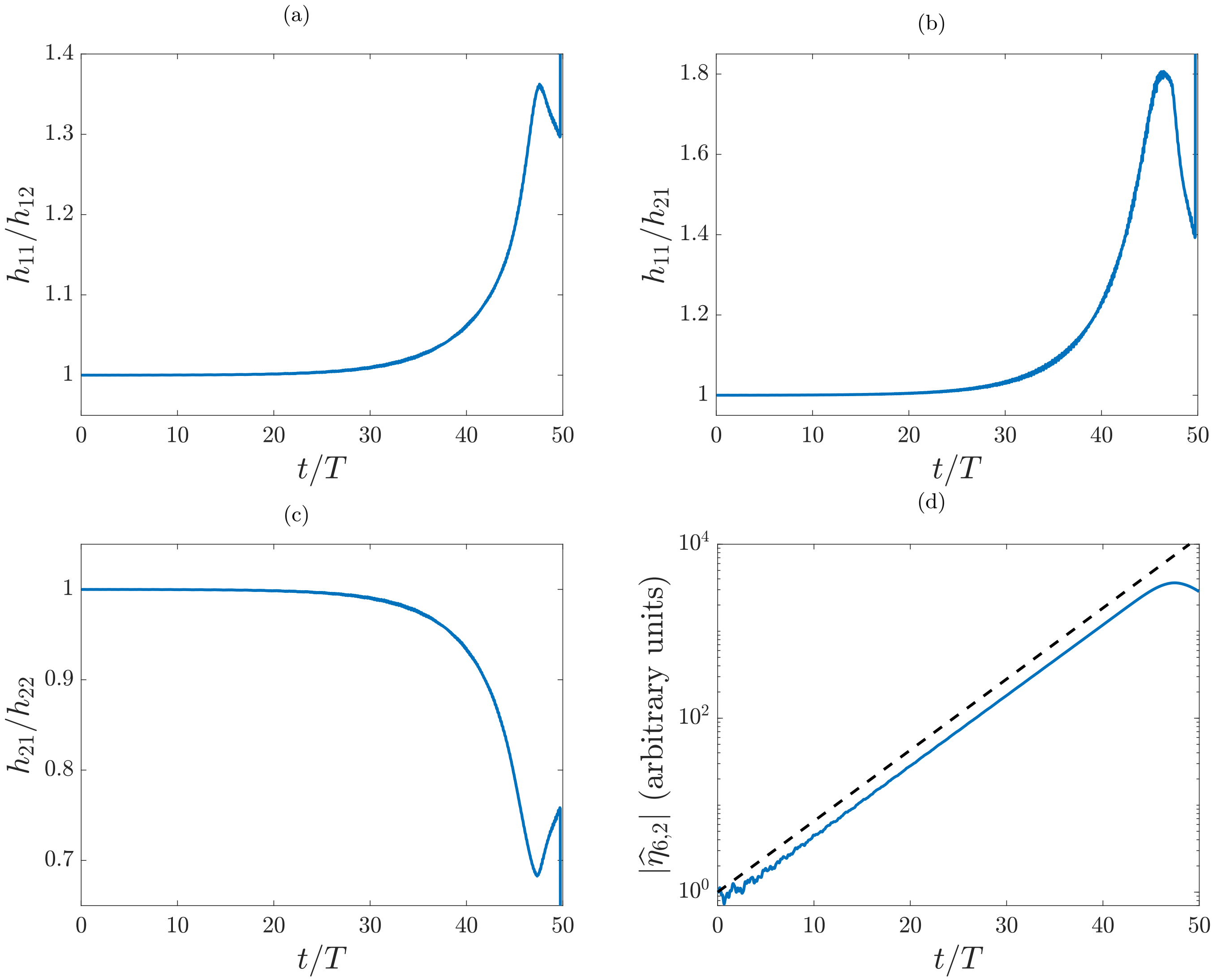

Figure 4: The output of `analyzeCW.m` when used with the parameters `kh = 2*pi`, `steep = 0.105`, `NPeriod = 50`, `wlX = 4`, `wlY = 2`, `epsDist = 0.0001`, `kp = 3/2`, `kq = 1.23`, `Nx = 64`, `Nx = 16`, `Ns = 8`, `NStep = 50`, `dampstrat = 2`, `dampCoef = 0.5`, `NSSGW = 10000`, `epsGMRES = 1E-9` and `bParam = 0.1`. (a) The ratio $h_{11}/h_{12}$. (b) The ratio $h_{11}/h_{21}$. (c) The ratio $h_{21}/h_{22}$. (d) A comparison of the growth of the most unstable Fourier coefficient (in this case the coefficient with indices $(6, 2)$) as predicted by the simulation and the analysis of McLean [6].

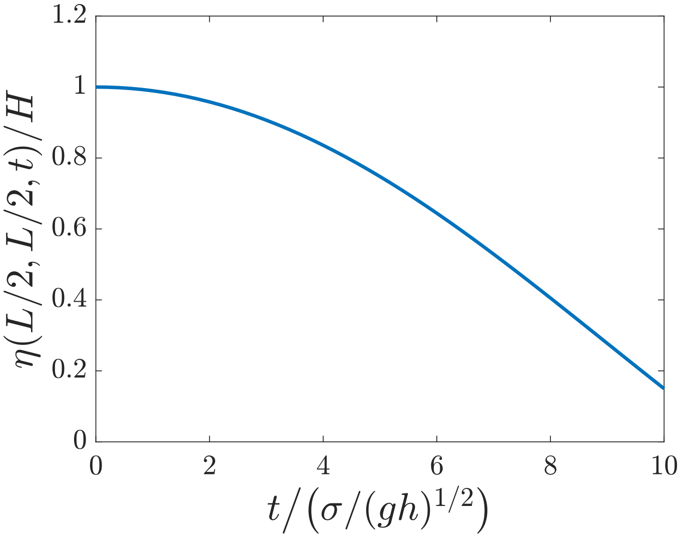

Figure 5: The output of `analyzeGH.m` when used with the parameters `hTilde = 1`, `sigmaTilde = 0.02`, `HTilde = 2^(-6)`, `NPeriod = 10`, `Nx = 64`, `Ny = 64`, `Ns = 10`, `NStep = 5`, `dampstrat = 1` and `epsGMRES = 1E-9`. The figure shows the time evolution of the surface elevation at $(x, y) = (L/2, L/2)$ as a function of time.

During the time integration, the wave motion is animated and the time in units of `T0` is written to the terminal. Once the time integration has finished, the simulation program writes the time evolution of the surface elevation at the position $(x, y) = (L/2, L/2)$ (`centerElevation`) as well as a list of discrete times (`time`) to file in the folder `OutputData`.

To analyze the results, the user should execute the analysis program, `analyzeSNW.m`. To do so, the path to the source directory should be specified in section 1 and the same non-dimensional parameters as above should be declared in section 2. The analysis program for the Gaussian hump illustrates the time evolution of the surface elevation at $(x, y) = (L/2, L/2)$. An example of the output of the analysis program is shown in figure 5 for the parameters `hTilde = 1`, `sigmaTilde = 0.02`, `HTilde = 2^(-6)`, `NPeriod = 10`, `Nx = 64`, `Ny = 64`, `Ns = 10`, `NStep = 5`, `dampstrat = 1` and `epsGMRES = 1E-9`.

### 3.4 Focusing Event

The simulation of a focusing event can be performed by executing the file `simulateFE.m` in the directory `FocusingEvent`. The code is set up to simulate the broad banded cases with $s = 4$ of Johannessen & Swan [4], who carried out a large number of experiments with focusing waves. As noted in the introduction to this section, the simulation of the focusing event deviates from the other simulations by the fact that all fundamental physical quantities are defined in SI units for this case. These quantities are specified in section 2 of `simulateFE` and are:

- `Lx` and `Ly`: The absolute size of the horizontal domain.
- `h`: The water depth.
- `g`: The gravitational acceleration.
- `A`: The focusing amplitude as predicted by first-order theory.
- `x0` and `y0`: The horizontal position of the focus as predicted by first-order theory.
- `b`: The vertical position of the artificial boundary condition.
- `t0` and `tf`: The initial and final time of the simulation relative to the time of focus as predicted by first-order theory. In order words, the time integration is carried in the time interval `[t0, tf]`.
- `deltat`: The time step.

In addition to these, the user must specify the non-dimensional computational parameters (also in the program's section 2):

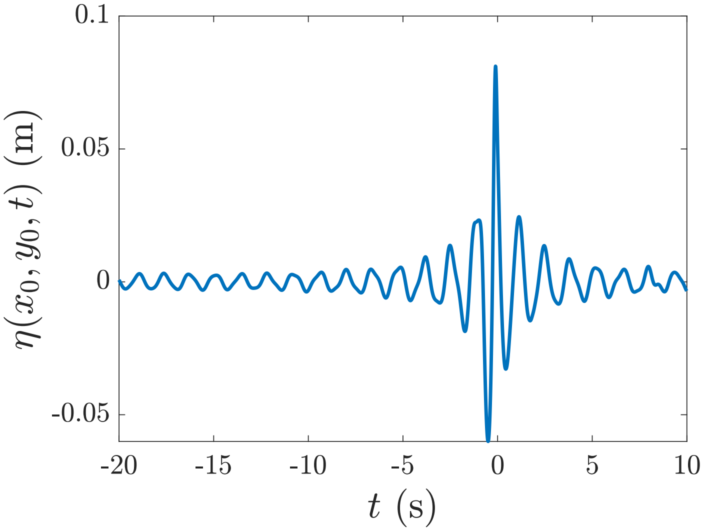

Figure 6: The output of `analyzeFE.m` when used with the parameters `Lx = 25.6`, `Ly = 25.6`, `g = 9.81`, `h = 1.2`, `A = 0.078`, `x0 = 20`, `y0 = Ly/2`, `b = 0.12`, `t0 = -20`, `tf = 10`, `deltat = 0.01`, `Nx = 128`, `Ny = 128`, `Ns = 10`, `dampstrat = 2`, `dampCoef = 0.7` and `epsGMRES = 1E-9`. The figure shows the time evolution of the surface elevation at the focus position $(x_0, y_0)$ as a function of time.

- `Nx`, `Ny` and `Ns`: The spatial resolution of the wave field. `2*Nx` grid points are used in the $x$-direction, `2*Ny` grid points are used in the $y$-direction, and `Ns+1` grid points are used in the vertical dimension. Note that the value of `Ns` is relevant only for the solution of the Laplace problem,
- `dampStrat` and `dampCoef`: If `dampStrat = 1` the time integration is not damped. If, however, `dampStrat = 2` the time integration is damped in accordance with eq. (4) of Klahn et al. [5]. The parameter `dampCoef` corresponds to the parameter $\alpha$ in eq. (4) of Klahn et al.
- `epsGMRES`: The relative tolerance to which the set of linear equations corresponding to the discretized Laplace problem is solved by GMRES. Note that setting `epsGMRES = 1E-14` corresponds to solving the system of equations to double precision.

During the time integration, the wave motion is animated and the time in units of seconds is written to the terminal. Once the time integration has finished, the simulation program writes the time evolution of the surface elevation at the focus position $(x_0, y_0)$ (`focusElevation`) as well as a list of discrete times (`time`) to file in the folder `OutputData`.

To analyze the results, the user should execute the analysis program, `analyzeFE.m`. This program illustrates the time evolution of the surface elevation at the focus point $(x_0, y_0)$. An example of the output of the program is shown in figure 6 for the parameters `Lx = 25.6`, `Ly = 25.6`, `g = 9.81`, `h = 1.2`, `A = 0.078`, `x0 = 20`, `y0 = Ly/2`, `b = 0.12`, `t0 = -20`, `tf = 10`, `deltat = 0.01`, `Nx = 128`, `Ny = 128`, `Ns = 10`, `dampstrat = 2`, `dampCoef = 0.7` and `epsGMRES = 1E-9`.

### Acknowledgement

The authors gratefully acknowledge the funding received from Centre for Oil and Gas - DTU/Danish Hydrocarbon Research and Technology Centre (DHRTC). Project name: DEWIOS. Project ID: FL_110.

## References

[1] BATTJES, J. A. 1974 Surf similarity. *Proceedings of the 14th Coastal Engineering Conference, Copenhagen, ASCE*, 466-480.

[2] CLAMOND, D. & DUTYKH, D. 2017 SSGW.m Matlab file exchange. URL
`https://www.mathworks.com/matlabcentral/fileexchange/61499-surface-gravity-waves`

[3] CLAMOND, D. & DUTYKH, D. 2018 Accurate fast computation of steady two-dimensional surface gravity waves in arbitrary depth. *J. Fluid Mech.* **844**, 491-518.

[4] JOHANNESSEN, T. B. & SWAN, C. 2001 A laboratory study of the focusing of transient and directionally spread surface water waves. *Proc. R. Soc. Lond. A* **457**, 971-1006.

[5] KLAHN, M., MADSEN, P.A. & FUHRMAN, D. R. 2021 Simulation of three-dimensional nonlinear water waves using a pseudospectral volumetric method with an artificial boundary condition. *Int. J. Numer. Meth. Fluids* (In production).

[6] MCLEAN, J. W. 1982 Instabilities of finite-amplitude waves. *J. Fluid Mech.* **114**, 315-330.