INF5620 -
First compulsory project

Arnfinn Mihle Paulsrud, Mathilde Nygaard Kamperud, Fredrik Eksaa Pettersen

October 23, 2012

1 Abstract

The project addresses the two-dimensional, standard, linear wave equation, with damping.

$$\frac{\partial^2 u}{\partial t^2} + b\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(q(x,y)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(q(x,y)\frac{\partial u}{\partial y}\right) + f(x,y,t) \tag{1}$$

We have solved this wave equation by the finite difference method. First we implemented the problem by scalar computation to get a working program, and when we were pretty sure our program worked we vectorized the problem for speed. We had three tests we could run to verify our code; we contructed a test case with constant solution, we made an exact 1D solution (plug wave) and we manufactured a standing wave. The latter test does not give an exact solution of the discrete equations, so we had to do an empirical analysis of the convergence. The expected error to the standing wave is

$$E = Ch^2 \tag{2}$$

where C s a constant chosen to be compatible with the stability criterion, and h is a common discretization parameter to be varied.
We used mayavi to visualize the 2D solution in 3D-plots.
The program can be used to simulate waves on water, and we chose to do this while looking at different sea-bottom shapes. We experimented a little as well, and ended up with a simulation of rain on a pond. The rain was made by letting the source term be a Gauss-function with random position and size.

2 mathematical problem

$$\frac{\partial^2 u}{\partial t^2} + b\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(q(x,y)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(q(x,y)\frac{\partial u}{\partial y}\right) + f(x,y,t) \tag{3}$$

The way we solve this numerically is by discretization

$$\frac{\partial^2 u}{\partial t^2} \rightarrow \frac{u^{n+1}_{i,j} - 2u^n_{i,j} + u^{n-1}_{i,j}}{\Delta t^2} \tag{4}$$

$$b\frac{\partial u}{\partial t} \rightarrow b\frac{u^{n+1}_{i,j} - u^{n-1}_{i,j}}{2\Delta t} \tag{5}$$

$$\begin{aligned}\frac{\partial}{\partial x}\left(q(x,y)\frac{\partial u}{\partial x}\right) &\rightarrow \frac{\left(q(x,y)\frac{\partial u}{\partial x}\right)_{i+\frac{1}{2},j} - \left(q(x,y)\frac{\partial u}{\partial x}\right)_{i-\frac{1}{2},j}}{\Delta x}\\ &= \frac{q(x,y)^n_{i+\frac{1}{2},j}\frac{u^n_{i+1,j}-u^n_{i,j}}{\Delta x} - q(x,y)^n_{i-\frac{1}{2},j}\frac{u^n_{i,j}-u^n_{i-1,j}}{\Delta x}}{\Delta x}\\ &= \frac{1}{\Delta x^2}\left(q(x,y)^n_{i+\frac{1}{2},j}\left(u^n_{i+1,j} - u^n_{i,j}\right) - q(x,y)^n_{i-\frac{1}{2},j}\left(u^n_{i,j} - u^n_{i-1,j}\right)\right)\end{aligned} \tag{6}$$

We can find a solution for $q(x,y)^n_{i+\frac{1}{2},j}$ and $q(x,y)^n_{i-\frac{1}{2},j}$ by taking the average between $q(x,y)^n_{i+1,j}$ and $q(x,y)^n_{i,j}$ and between $q(x,y)^n_{i,j}$ and $q(x,y)^n_{i-1,j}$.

$$q(x,y)^n_{i+\frac{1}{2},j} \approx \frac{q(x,y)^n_{i+1,j} + q(x,y)^n_{i,j}}{2} \tag{7a}$$

and

$$q(x,y)^n_{i-\frac{1}{2},j} \approx \frac{q(x,y)^n_{i,j} + q(x,y)^n_{i-1,j}}{2} \tag{8a}$$

This gives us

$$\begin{aligned}\frac{\partial}{\partial x}\left(q(x,y)\frac{\partial u}{\partial x}\right) &\to \frac{1}{\Delta x^2}\left(\frac{q(x,y)^n_{i+1,j} + q(x,y)^n_{i,j}}{2}\left(u^n_{i+1,j} - u^n_{i,j}\right) - \frac{q(x,y)^n_{i,j} + q(x,y)^n_{i-1,j}}{2}\left(u^n_{i,j} - u^n_{i-1,j}\right)\right)\\ &= \frac{1}{2\Delta x^2}\left(\left(q(x,y)^n_{i+1,j} + q(x,y)^n_{i,j}\right)\left(u^n_{i+1,j} - u^n_{i,j}\right) - \left(q(x,y)^n_{i,j} + q(x,y)^n_{i-1,j}\right)\left(u^n_{i,j} - u^n_{i-1,j}\right)\right)\end{aligned} \tag{9}$$

By doing the same for $\left(q(x,y)\frac{\partial u}{\partial y}\right)$, we get

$$\begin{aligned}&\frac{\partial}{\partial y}\left(q(x,y)\frac{\partial u}{\partial y}\right)\\ &\qquad\to \frac{1}{2\Delta y^2}\left(\left(q(x,y)^n_{i,j+1} + q(x,y)^n_{i,j}\right)\left(u^n_{i,j+1} - u^n_{i,j}\right) - \left(q(x,y)^n_{i,j} + q(x,y)^n_{i,j-1}\right)\left(u^n_{i,j} - u^n_{i,j-1}\right)\right)\end{aligned} \tag{10}$$

The last part of eq.(3) becomes

$$f(x,y,t) \to f(x,y,t)^n_{i,j} \tag{11}$$

If we now combine our discretized equations we get a numerically solvable wave equation

$$\begin{aligned}&\frac{u^{n+1}_{i,j} - 2u^n_{i,j} + u^{n-1}_{i,j}}{\Delta t^2} + b\frac{u^{n+1}_{i,j} - u^{n-1}_{i,j}}{2\Delta t} =\\ &\qquad \frac{1}{2\Delta x^2}\left(\left(q(x,y)^n_{i+1,j} + q(x,y)^n_{i,j}\right)\left(u^n_{i+1,j} - u^n_{i,j}\right) - \left(q(x,y)^n_{i,j} + q(x,y)^n_{i-1,j}\right)\left(u^n_{i,j} - u^n_{i-1,j}\right)\right)\\ &\qquad + \frac{1}{2\Delta y^2}\left(\left(q(x,y)^n_{i,j+1} + q(x,y)^n_{i,j}\right)\left(u^n_{i,j+1} - u^n_{i,j}\right) - \left(q(x,y)^n_{i,j} + q(x,y)^n_{i,j-1}\right)\left(u^n_{i,j} - u^n_{i,j-1}\right)\right)\\ &\qquad + f(x,y,t)^n_{i,j}\end{aligned}$$

We would now like to solve this equation for $u^{n+1}_{i,j}$

$$\begin{aligned}&u^{n+1}_{i,j} - 2u^n_{i,j} + u^{n-1}_{i,j} + \frac{b\Delta t}{2}\left(u^{n+1}_{i,j} - u^{n-1}_{i,j}\right) =\\ &\qquad \frac{\Delta t^2}{2\Delta x^2}\left(\left(q(x,y)^n_{i+1,j} + q(x,y)^n_{i,j}\right)\left(u^n_{i+1,j} - u^n_{i,j}\right) - \left(q(x,y)^n_{i,j} + q(x,y)^n_{i-1,j}\right)\left(u^n_{i,j} - u^n_{i-1,j}\right)\right)\\ &\qquad + \frac{\Delta t^2}{2\Delta y^2}\left(\left(q(x,y)^n_{i,j+1} + q(x,y)^n_{i,j}\right)\left(u^n_{i,j+1} - u^n_{i,j}\right) - \left(q(x,y)^n_{i,j} + q(x,y)^n_{i,j-1}\right)\left(u^n_{i,j} - u^n_{i,j-1}\right)\right)\\ &\qquad + \Delta t^2 f(x,y,t)^n_{i,j}\end{aligned}$$

$$
\begin{aligned}
&u_{i,j}^{n+1}\left(1+\frac{b\Delta t}{2}\right) = \\
&\qquad \frac{\Delta t^2}{2\Delta x^2}\left(\left(q(x,y)_{i+1,j}^n + q(x,y)_{i,j}^n\right)\left(u_{i+1,j}^n - u_{i,j}^n\right) - \left(q(x,y)_{i,j}^n + q(x,y)_{i-1,j}^n\right)\left(u_{i,j}^n - u_{i-1,j}^n\right)\right) \\
&\qquad + \frac{\Delta t^2}{2\Delta y^2}\left(\left(q(x,y)_{i,j+1}^n + q(x,y)_{i,j}^n\right)\left(u_{i,j+1}^n - u_{i,j}^n\right) - \left(q(x,y)_{i,j}^n + q(x,y)_{i,j-1}^n\right)\left(u_{i,j}^n - u_{i,j-1}^n\right)\right) \\
&\qquad + \Delta t^2 f(x,y,t)_{i,j}^n + 2u_{i,j}^n + u_{i,j}^{n-1}\left(\frac{b\Delta t}{2} - 1\right)
\end{aligned}
$$

We end up with the discretized equation

$$
\begin{aligned}
&u_{i,j}^{n+1} = \\
&\qquad \tfrac{\Delta t^2}{2\Delta x^2\left(1+\frac{b\Delta t}{2}\right)}\left(\left(q(x,y)_{i+1,j}^n + q(x,y)_{i,j}^n\right)\left(u_{i+1,j}^n - u_{i,j}^n\right) - \left(q(x,y)_{i,j}^n + q(x,y)_{i-1,j}^n\right)\left(u_{i,j}^n - u_{i-1,j}^n\right)\right) \\
&\qquad + \tfrac{\Delta t^2}{2\Delta y^2\left(1+\frac{b\Delta t}{2}\right)}\left(\left(q(x,y)_{i,j+1}^n + q(x,y)_{i,j}^n\right)\left(u_{i,j+1}^n - u_{i,j}^n\right) - \left(q(x,y)_{i,j}^n + q(x,y)_{i,j-1}^n\right)\left(u_{i,j}^n - u_{i,j-1}^n\right)\right) \\
&\qquad + \tfrac{\Delta t^2}{1+\frac{b\Delta t}{2}} f(x,y,t)_{i,j}^n + 2u_{i,j}^n + u_{i,j}^{n-1}\tfrac{1+\frac{b\Delta t}{2}}{\frac{b\Delta t}{2}-1}
\end{aligned}
\tag{12}
$$

3 Boundary condition

If we take a look at out numerical equation, we can see that well get a problem when we get to the boundary $i = 0, \quad i = L_x, \quad j = 0$ and $j = L_y$
We have been given the boundary condition $\frac{\partial u}{\partial \eta} = 0$

$$\frac{\partial u}{\partial \eta} = 0 \rightarrow [D_{2x}u]_{i,j}^n = 0 \tag{13}$$

When we are on the boundary $x = 0 \Rightarrow i = 0$ we get

$$[D_{2x}u]_{i,j}^n = \frac{u_{i+1,j}^n - u_{i-1,j}^n}{2\Delta x} = 0 \tag{14}$$

$$
\begin{aligned}
\frac{u_{1,j}^n - u_{-1,j}^n}{2\Delta x} &= 0 \\
\rightarrow u_{1,j}^n &= u_{-1,j}^n
\end{aligned}
$$

When we are on the boundary $y = 0 \Rightarrow j = 0$ we get

$$[D_{2x}u]_{i,j}^n = \frac{u_{i+1,j}^n - u_{i-1,j}^n}{2\Delta x} = 0 \tag{15}$$

$$
\begin{aligned}
\frac{u_{1,j}^n - u_{-1,j}^n}{2\Delta x} &= 0 \\
\rightarrow u_{1,j}^n &= u_{-1,j}^n
\end{aligned}
$$

4 Initial condition

If we take a look at eq.(**??**) we can see that we are going to get a problem with $u_{i,j}^{n-1}$ in our first time step. We can solve this by looking at our initial condition. If we assume that the velocity at $t = 0$ is zero, $u'(x, y, 0) = 0$ then we get

$$\frac{\partial u}{\partial t} = 0 \rightarrow [D_t u]_{i,j}^n = V(x,y) \tag{16}$$

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n-1}}{2\Delta t}=V(x,y) \tag{17}$$

For $t=0 \rightarrow n=0$ we get

$$\frac{u_{i,j}^{1}-u_{i,j}^{-1}}{2\Delta t}=V(x,y) \tag{18}$$

$$u_{i,j}^{1}=u_{i,j}^{-1}+2V(x,y)\Delta t \tag{19}$$

$$u_{i,j}^{-1}=u_{i,j}^{1}-2V(x,y)\Delta t \tag{20}$$

This solves our problem. If we now use $n=0$ in eq.(**??**) we get

$$\begin{aligned}
&u_{i,j}^{1}\left(1+\frac{b\Delta t}{2}\right)=\\
&\qquad\frac{\Delta t^2}{2\Delta x^2}\left(\left(q(x,y)_{i+1,j}^0+q(x,y)_{i,j}^0\right)\left(u_{i+1,j}^0-u_{i,j}^0\right)-\left(q(x,y)_{i,j}^0+q(x,y)_{i-1,j}^0\right)\left(u_{i,j}^0-u_{i-1,j}^0\right)\right)\\
&\qquad+\frac{\Delta t^2}{2\Delta y^2}\left(\left(q(x,y)_{i,j+1}^0+q(x,y)_{i,j}^0\right)\left(u_{i,j+1}^0-u_{i,j}^0\right)-\left(q(x,y)_{i,j}^0+q(x,y)_{i,j-1}^0\right)\left(u_{i,j}^0-u_{i,j-1}^0\right)\right)\\
&\qquad+\Delta t^2 f(x,y,t)_{i,j}^0+2u_{i,j}^0+u_{i,j}^{-1}\left(\frac{b\Delta t}{2}-1\right)
\end{aligned}$$

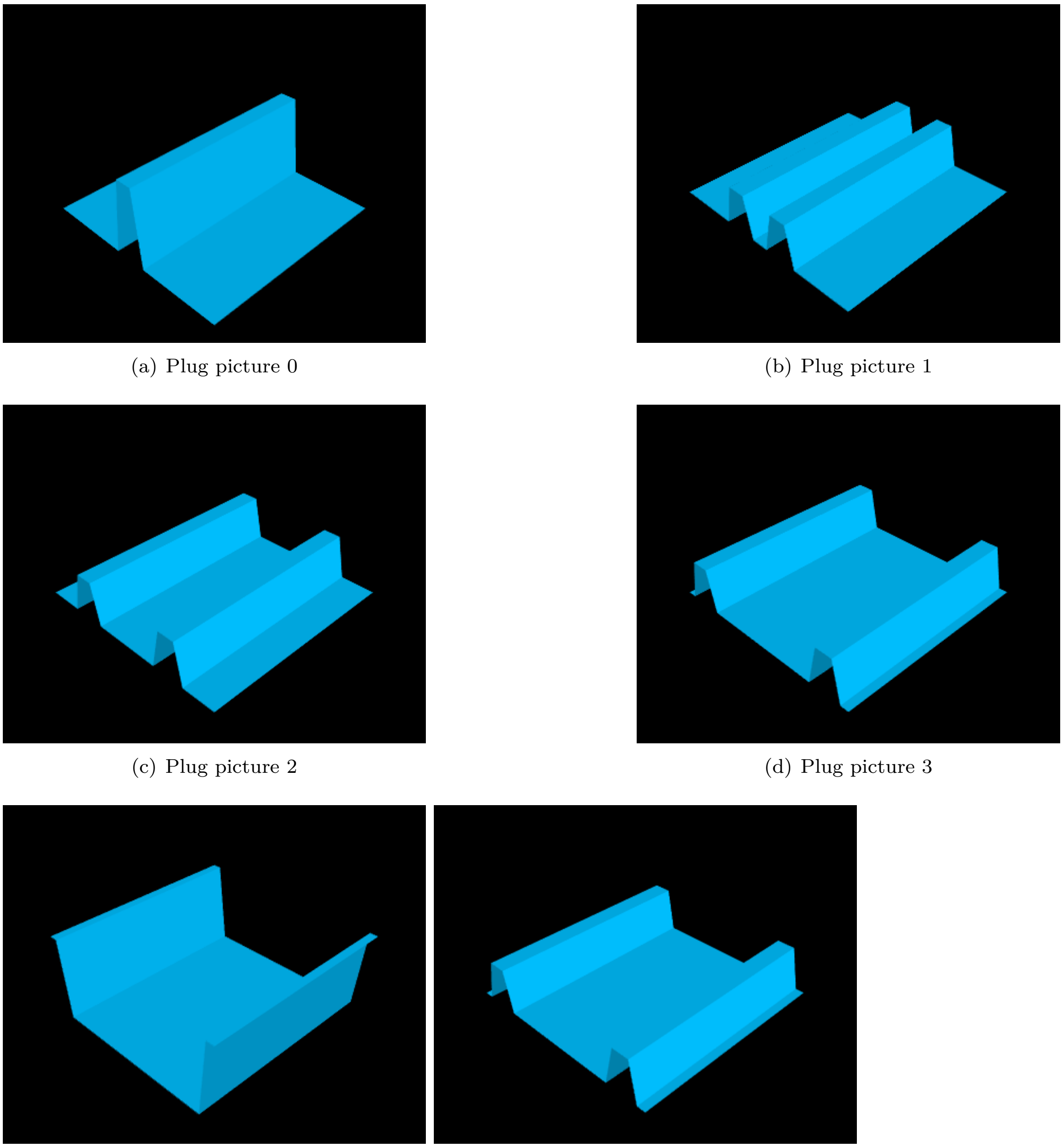
If we now use $u_{i,j}^{1}=u_{i,j}^{-1}$ we get

$$\begin{aligned}
&u_{i,j}^{1}=\\
&\quad\frac{\Delta t^2}{4\Delta x^2}\left(\left(q(x,y)_{i+1,j}^0+q(x,y)_{i,j}^0\right)\left(u_{i+1,j}^0-u_{i,j}^0\right)-\left(q(x,y)_{i,j}^0+q(x,y)_{i-1,j}^0\right)\left(u_{i,j}^0-u_{i-1,j}^0\right)\right)\\
&\quad+\frac{\Delta t^2}{4\Delta y^2}\left(\left(q(x,y)_{i,j+1}^0+q(x,y)_{i,j}^0\right)\left(u_{i,j+1}^0-u_{i,j}^0\right)-\left(q(x,y)_{i,j}^0+q(x,y)_{i,j-1}^0\right)\left(u_{i,j}^0-u_{i,j-1}^0\right)\right)\\
&\quad+\frac{\Delta t^2}{2}f(x,y,t)_{i,j}^0+u_{i,j}^0-V(x,y)\Delta t\left(\frac{b\Delta t}{2}-1\right)
\end{aligned}$$

5 Manufactured solutions

5.1 Exact 1D solution

A simple 1D square or plug wave should propagate witth exact plug shape when $c\Delta t\Delta x=1$. We choose c = 1, and set $\Delta t=\Delta x$, and use a plug as our inital condition. The plug splits in half and the two parts propagates in opposite direction as seen in figure (1).



(a) Plug picture 0

(b) Plug picture 1

(c) Plug picture 2

(d) Plug picture 3

(e) Plug picture 4

(f) Plug picture 5

Figure 1: The plug shape wave - first 6 pictures of a movie called plug.gif

We see that the plug propagates exactly how we wanted it to. The movie of the plug has the name plug.gif.

5.2 Standing wave

As a test of the program, we manufactured a solution for constant q (this means that the velocity of the waves is constant over the domain). The wanted solution is a standing wave given in eq. (21).

$$u(x,y,t) = e^{-bt}\cos(\frac{m_x\pi}{L_x})\cos(\frac{m_y\pi}{L_y})\cos(\omega t) \tag{21}$$

m_x and m_y are arbitrary integers that decides how many wavetops we end up with on our domain. The parameter ω is the frequency, and has to be chosen to fit the numerical solution.

To manufactor this solution we start by letting the initial condition be the exact standing wave at a time t=0. We need to fit the source term f(x,y,t) so that eq. (21) is a solution to our wave equation (eq. 3), and find a suitable initial velocity V(x,y).

5.2.1 Finding f(x,y,t)

Choose some $q(x,y) = A$, $A \neq 0$ eq.(3) becomes

$$\frac{\partial^2 u}{\partial t^2} + v\frac{\partial u}{\partial t} = A\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(x,y,t) \tag{22}$$

If we now insert eq.(21) into eq.(22) we get

$$\begin{aligned}
\frac{\partial u}{\partial t} &= -bu(x,y,t) - \omega \underbrace{\frac{\sin(\omega t)}{\cos(\omega t)}}_{\tan(\omega t)} u(x,y,t) \\
\frac{\partial^2 u}{\partial t^2} &= (b^2 - \omega^2)u(x,y,t) + 2\omega b \underbrace{\frac{\sin(\omega t)}{\cos(\omega t)}}_{\tan(\omega t)} u(x,y,t) \\
\frac{\partial^2 u}{\partial x^2} &= -\left(\frac{m_x \pi}{L_x}\right)^2 u(x,y,t) \\
\frac{\partial^2 u}{\partial y^2} &= -\left(\frac{m_y \pi}{L_y}\right)^2 u(x,y,t)
\end{aligned}$$

$$\Rightarrow \left[(b^2 - \omega^2) + 2\omega b\tan(\omega t) - b^2 - b\omega\tan(\omega t)\right] u(x,y,t) = -A\left[\left(\frac{m_x\pi}{L_x}\right)^2 + \left(\frac{m_y\pi}{L_y}\right)^2\right] u(x,y,t) + f(x,y,t)$$

$$\Rightarrow f(x,y,t) = \left[\omega b\tan(\omega t) - \omega^2 + A\pi^2\left(\left(\frac{m_x}{L_x}\right)^2 + \left(\frac{m_y}{L_y}\right)^2\right)\right] u(x,y,t)$$

So now we have our source term:

$$f(x,y,t) = \left[\omega b\tan(\omega t) - \omega^2 + A\pi^2\left(\left(\frac{m_x}{L_x}\right)^2 + \left(\frac{m_y}{L_y}\right)^2\right)\right] u(x,y,t) \tag{23}$$

5.2.2 Finding V(x,y)

The initial velocity is given by the exact solution (the initial condition) by

$$V(x,y) = \frac{\partial u(x,y,t)}{\partial t}|_{t=0}$$

We already found the time derivative of u in the latter subsection (finding f(x,y,t)),

$$V(x,y) = \frac{\partial u(x,y,t)}{\partial t}|_{t=0} = -(\omega tan(\omega t) + b)u(x,y,t)|_{t=0} = -bu(x,y,t=0) \tag{24}$$

5.2.3 The results

We have found all the expressions we need to implement the standing wave. Figure (2) shows the initial condition (the exact solution at t=0).

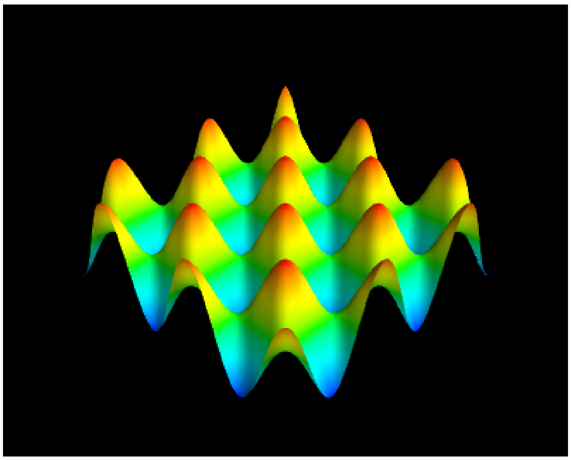


Figure 2: The exact solution at t=0, also the initial condition

Figure (3) is the first 6 pictures of two movies (FYLL INN NAVN PÃĚ FILMENE), where one shows the exact solution and the other shows the numerical solution. The numerically found solution looks good when you look at that movie alone, but when you compare it to the exact solution you see that it is a bit off. Picture (e),(f),(g) and (h) are typical examples of the difference between the numerical and exact solution. This is not very good, so lets analyse the results.
The error E is assumed to behave like

$$E = C_t \Delta t^2 + C_x \Delta x^2 + C_y \Delta y^2 \tag{25}$$

choose $\Delta t = F_t h$, $\Delta x = F_x h$ and $\Delta y = F_y h$, where F_t, F_x, F_y are freely chosen constants factors compatible with the stability criterion. The error can then be expressed as

$$E = Ch^2 \tag{26}$$

where $C = C_x F_t^2 + C_y F_x^2 + C_t F_t^2$. This means that E/h^2 should be approximately constant.
We chose $F_t = 1$, meaning $h = \Delta t$. The stability criterion says that $\Delta x = \Delta t\sqrt{2} = h\sqrt{2} \Rightarrow F_x = \sqrt{2}$. The same argumentation give $F_y = \sqrt{2}$. Our program automaticly sets $\Delta x = \Delta t\sqrt{2}$ and $\Delta y = \Delta t\sqrt{2}$, so all we need to do is run the program for different Δt and see how the error behaves. Below is the results of our analysis

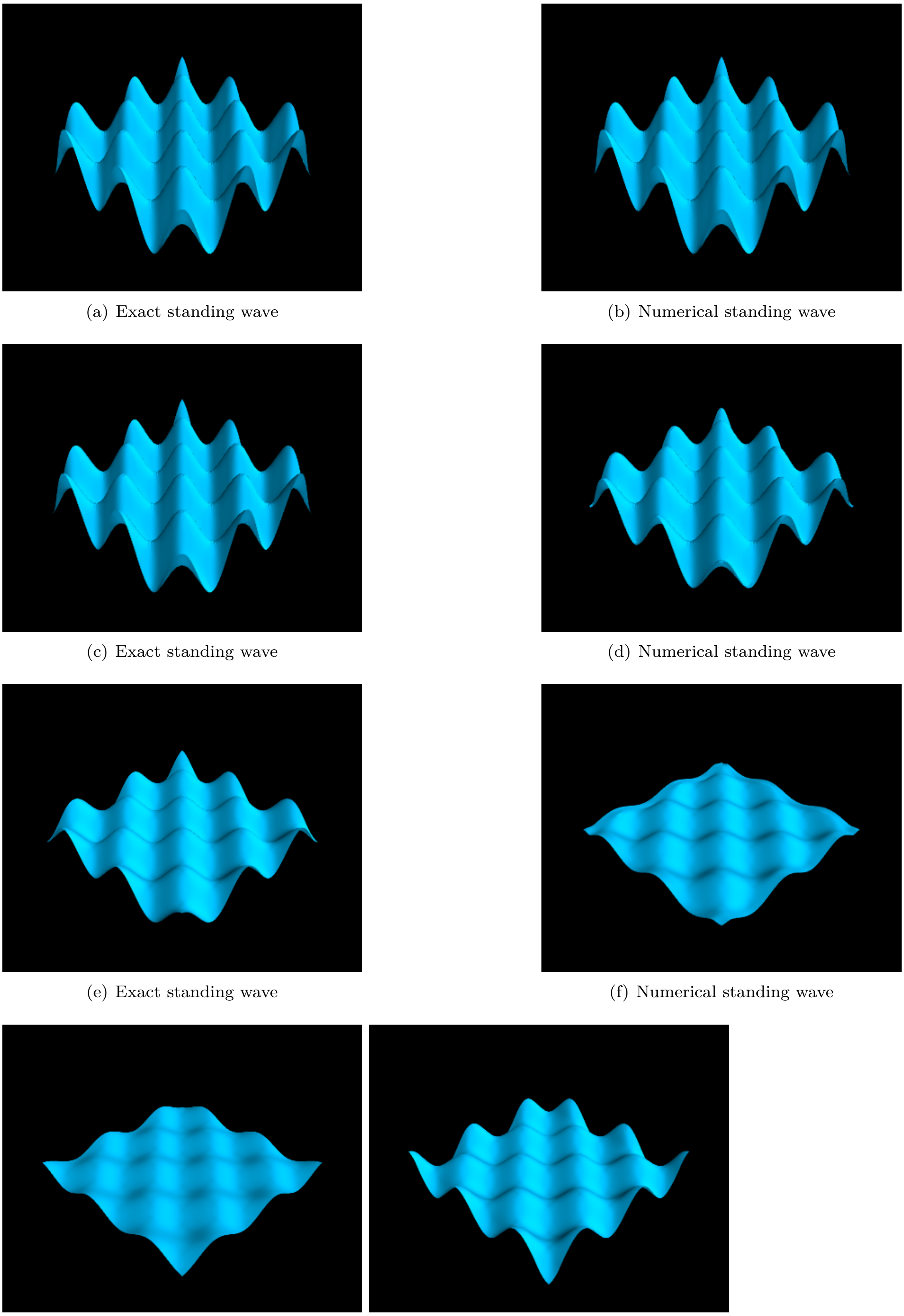
```
error for dt=1.00: 2.7598
E/h**2: 2.7598
error for dt=0.50: 0.7953
E/h**2: 3.1810
error for dt=0.20: 0.5349
E/h**2: 13.3727
error for dt=0.10: 0.2529
E/h**2: 25.2943
error for dt=0.02: 0.2987
E/h**2: 746.7569
error for dt=0.01: 0.3175
E/h**2: 3175.3953
```

The absolute error decreases from $\Delta t = 1$ to $\Delta t = 0.02$, and after that it starts increasing again. We see that E/h^2 is clearly not constant, it increases fast as Δt decreases. This tells us something is wrong with our program, and we have spendt many hours trying to find the source of this error, unsuccesfully.

6 Effect of various hill shapes

The velocity of waves on water is determined by the water depth. The q(x,y) function in our wave equation is

$$q(x, y) = gH(x, y) \tag{27}$$



(a) Exact standing wave

(b) Numerical standing wave

(c) Exact standing wave

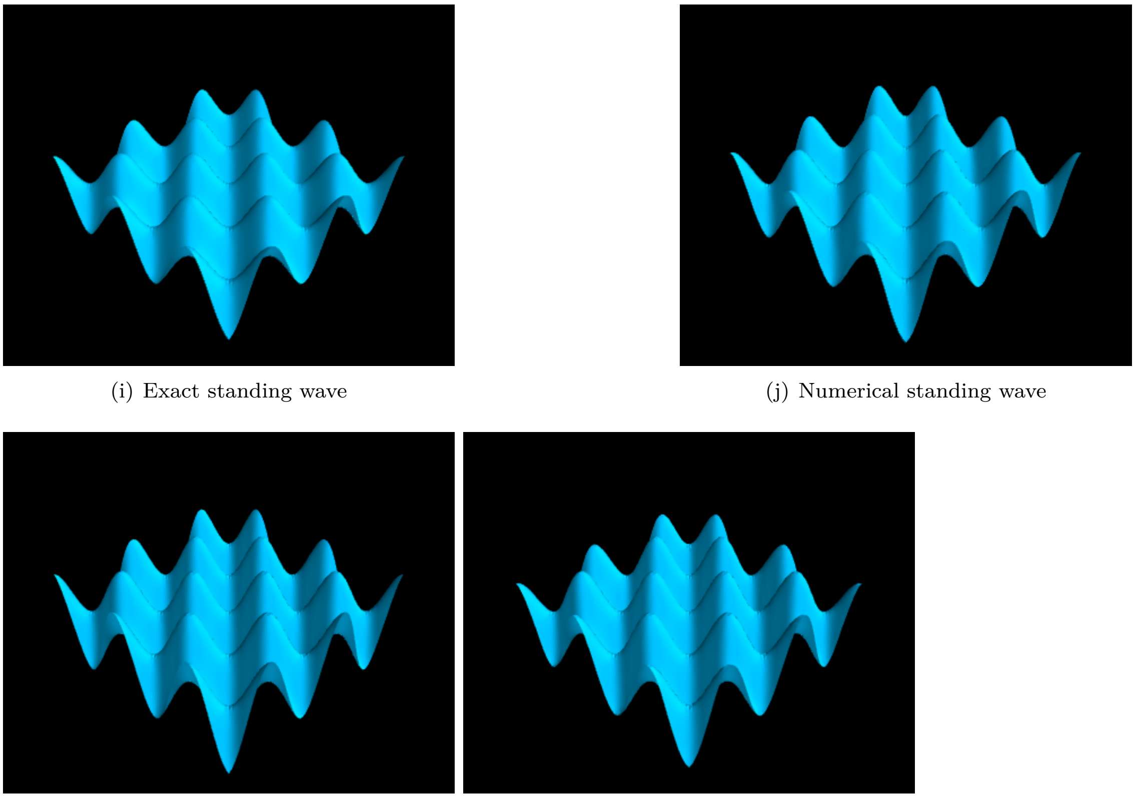
(d) Numerical standing wave

(e) Exact standing wave

(f) Numerical standing wave

(g) Exact standing wave

(h) Numerical standing wave



(i) Exact standing wave

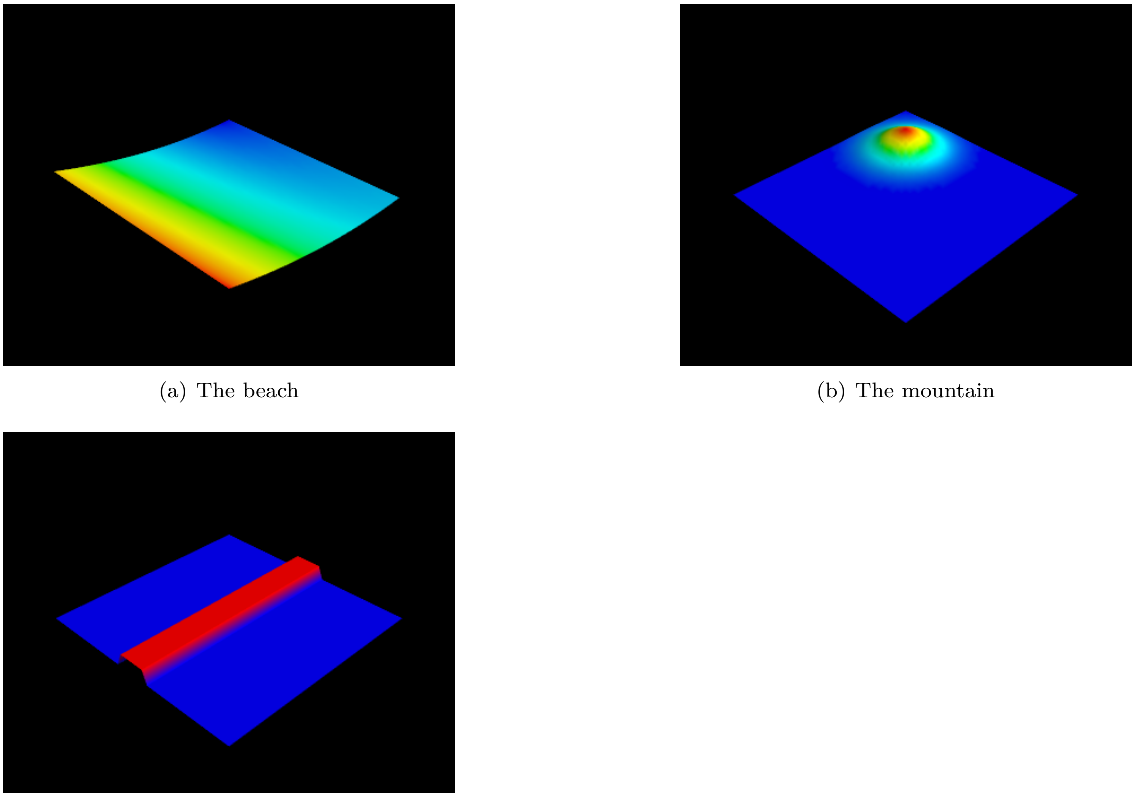
(j) Numerical standing wave

(k) Exact standing wave

(l) Numerical standing wave

Figure 3: This is the first 6 pictures of two movies, one showing the exact solution and the other showing the numerically found solution.

where $H(x, y)$ is the water depth when the surface is flat. We looked at three different sea bottoms; a beach shaped bottom, a bottom with an undersea mountain (gauss-form) and a box shape similar to the plug wave.



(a) The beach

(b) The mountain

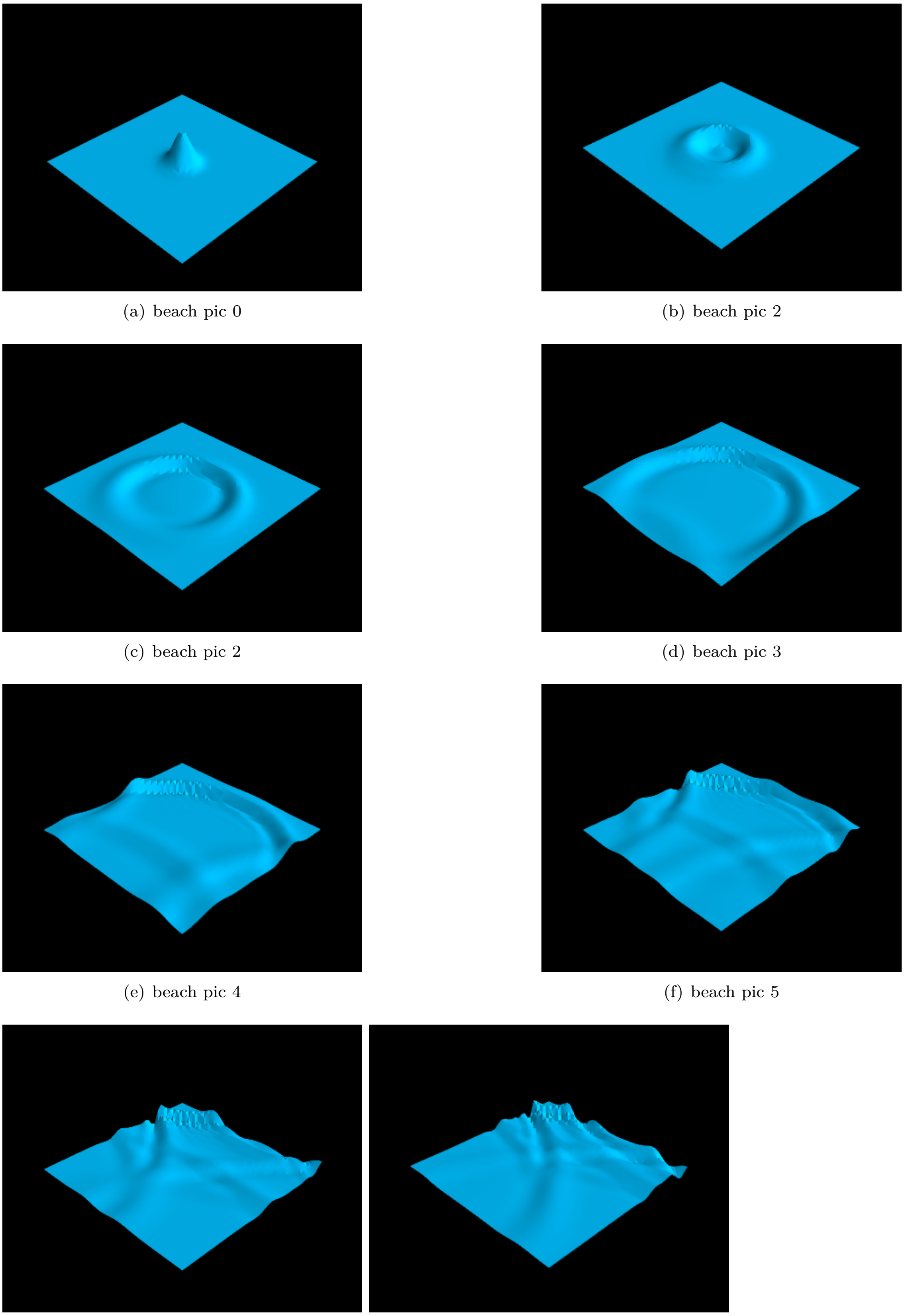
(c) The box

Figure 4: The different sea bottoms

Figure (4) show the three different sea bottoms we tested for.

6.1 The beach

We used a gauss for out initial condition, and figure (5) show the time evolution of the waves for the beach-shaped sea bottom. We see that there is some noise when the wave moves towards the corner in the back. This is a bit weird since the sea bottom actually is a quite smooth surface. The reason that noise chose to be only in that corner is because this is where the depth of the water is smallest. We believe there is a bug in our program, so maybe we can blame the noise on that bug.



(a) beach pic 0

(b) beach pic 2

(c) beach pic 2

(d) beach pic 3

(e) beach pic 4

(f) beach pic 5

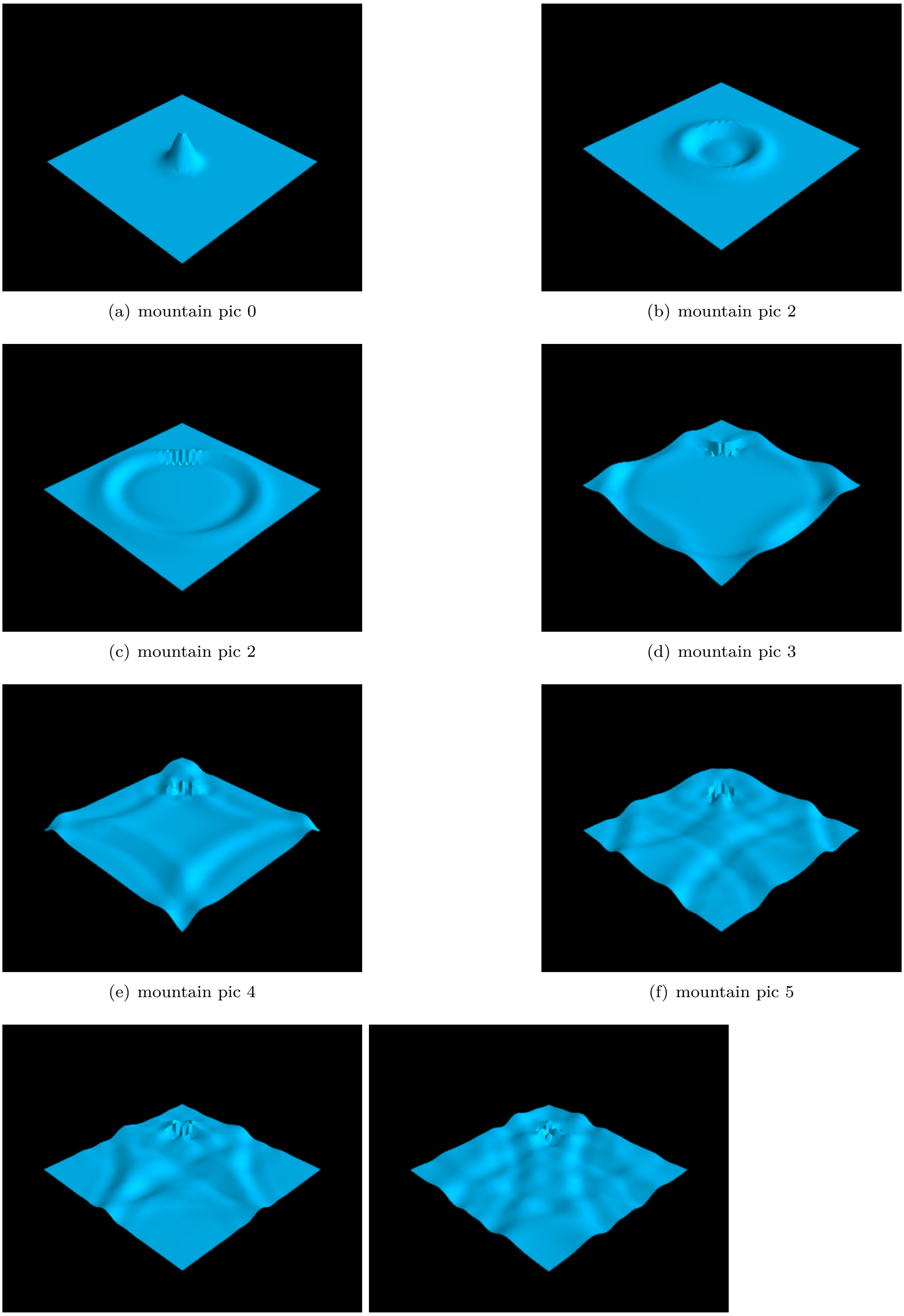
(g) beach pic 6

(h) beach pic 7

Figure 5: The waves on a beach

6.2 The mountain

Again we start out with a gauss-wave, but this time the sea bottom has a mountain in the corner. Figure (6) shows the time evolution of the gauss wave over this type of bottom. On the latest plots in the time evolution ((d) - (h)) we clearly see that there is a mountain disrupting the waves in the back corner. The waves over the mountain is not smooth (there is noise there), as one might expect.



(a) mountain pic 0

(b) mountain pic 2

(c) mountain pic 2

(d) mountain pic 3

(e) mountain pic 4

(f) mountain pic 5

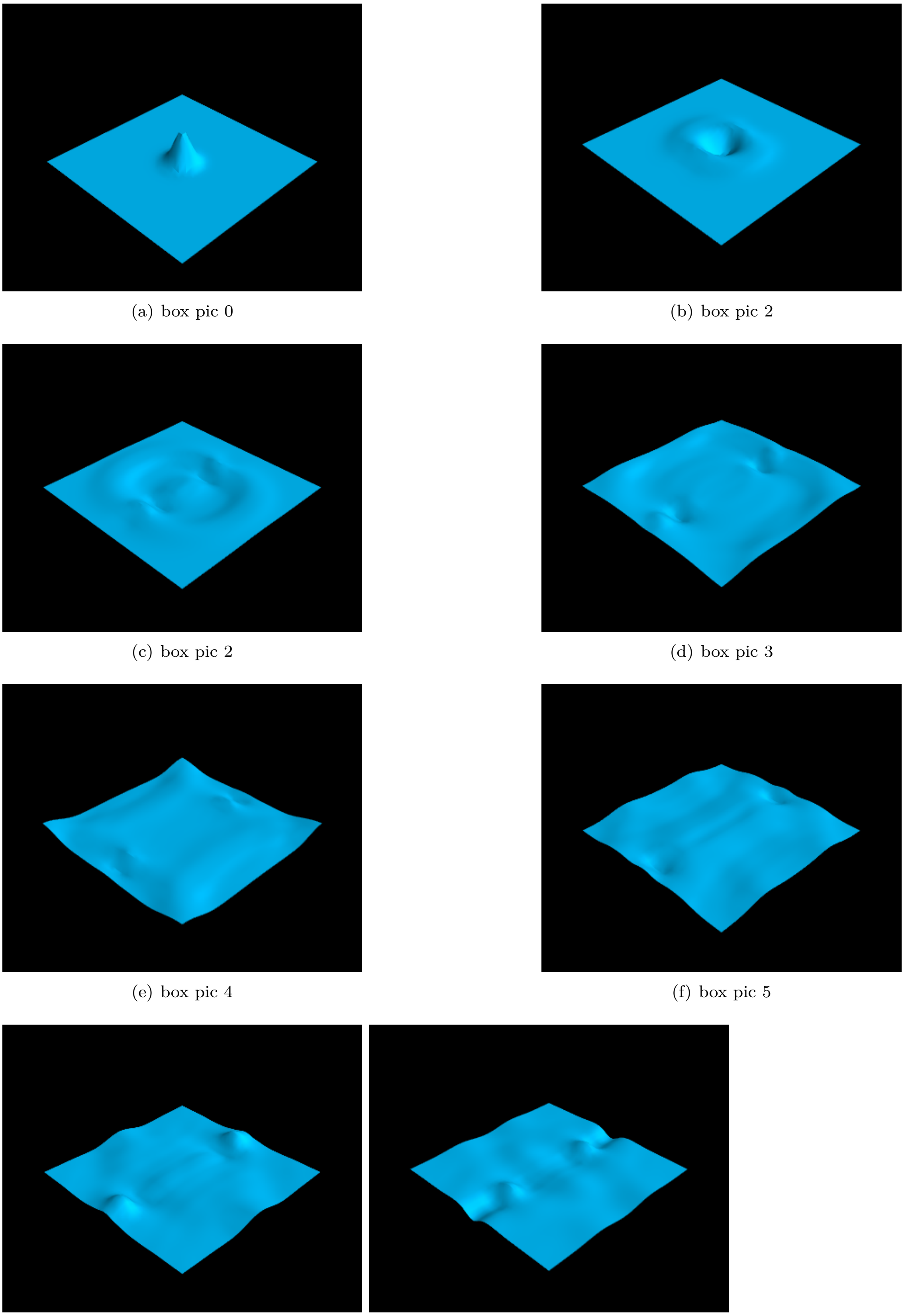
(g) mountain pic 6

(h) mountain pic 7

Figure 6: The waves over a mountain

6.3 The box

The gauss wave is the initial condition again, but now the bottom has a box shape. Figure (7) shows the time evolution of the gauss wave over this bottom. The disruption of the wave does mach the shape of the box, and there does not seem to be much noise, even though the velocity of the wave is abruptly changed when the wave crosses the box. The smooth gaussian mountain on the other hand made quite a bit of noise and the smoothest shapes of all the bottoms, the beach



(a) box pic 0

(b) box pic 2

(c) box pic 2

(d) box pic 3

(e) box pic 4

(f) box pic 5

(g) box pic 6

(h) box pic 7

Figure 7: The waves over a box

7 Rain on a pond