3 Dimensional Water Wave Equation with Surface Tension Simulation and Numerical Estimates of Dispersive Decay

Pak Kau Lim [1,2] and J. Douglas Wright[3]

Abstract

By solving and linearizing the Euler equation with surface tension, gravity, and characterized by irrotational vortex, this allows us to simulate the water wave using Fast Fourier Transform in MATLAB. The numerical estimates of linear dispersive decay is obtained and compared to the analytical estimates proven by Prof. Daniel Spirn and Prof. J. Douglas Wright using oscillatory integral methods. A practical reason in the investigation of the dispersive decay is to shed light in the complex nature of tsunami.

1 Introduction

1.1 Water Wave Equation

The physical model of water wave described in this article is governed by the Euler's equation under the assumptions of incompressible, irrotational flow, with uniform density, and negligible viscosity and wind. The formulation of the Euler's equation and usual model for surface tension under the above assumptions is derived by Segur and Henderson et al:

$$\partial_t \phi + \frac{1}{2}|\nabla \phi|^2 + \frac{P}{\rho} + gz = 0 \tag{1}$$

On $z = \eta(x, y, z, t)$,

$$P = -\tau \left[\nabla \cdot \left(\frac{\nabla \eta}{\sqrt{1+|\nabla \eta|^2}}\right)\right] \tag{2}$$

provided by Segur and Henderms that $|\nabla \eta|^2$ is negligible.

ϕ is the velocity potential results from Stokes' Theorem under irrotational flow. P represents pressure. Both g and τ represent the gravitational constant and surface tension constant respectively. By fixing a Euclidean coordinate on an initially calm water surface free of ripples, we have η as the elevation of the surface of water wave, where $h = h(x, y)$ is the elevation of sea floor.

Combining equations 1 and 2 for $\nabla \phi = 0$ and $z = 0$, this yields the general water wave equation

$$\phi_t + g\eta = \frac{\tau}{\rho}(\eta_{xx} + \eta_{yy}) \tag{3}$$

[1] *Department of Mathematics and Statistics, Boston University, MA* 02215

[2] *Department of Mechanical Engineering, Drexel University, Philadelphia, PA* 19104

[3] *Department of Mathematics, Drexel University, Philadelphia, PA* 19104

1.2 Solving Laplace's Equation

Suppose

$$\phi = \phi(x, y, z) \tag{4}$$

given boundary conditions that $\Delta\phi = 0$ when $x, y \rightarrow \infty$, $z = -h, or\ z = 0$ with φ as the values of initial condition, such that

$$\phi_z(x, y, -h) = 0 \tag{5}$$

$$\phi(x, y, 0) = \varphi(x, y, 0) \tag{6}$$

these boundary conditions eliminate the effect of reflection. Taking the Fourier Transform, F_{xy}, of equation 4 with respect to xy for $x, y \in (-\infty, \infty)$ gives

$$-k^2\hat{\phi} + \hat{\phi}_{zz} = 0 \tag{7}$$

where $k^2 = \omega^2 + \upsilon^2$, ω and υ are the frequencies in the domain of Fourier Transform with respect to x and y. Now equation 7 reduced to an ODE which has solution with coefficients terms A and B as $k > 0$

$$\hat{\phi}(\omega, \upsilon, z) = A(k)\cosh(kz) + B(k)\sinh(kz) \tag{8}$$

According to the above boundary equation, it suggests the computation of the partial derivative of $\hat{\phi}(\omega, \upsilon, z)$ along the z direction; and takes the Fourier Transform of both boundary and initial condition equations. With appropriate substitution, it is not difficult to solve the coefficient terms. The Inverse Fourier Transform of $\hat{\phi}(\omega, \upsilon, z)$ yields the solution

$$\phi(x, y, z) = F_{xy}{}^{-1}[(\cosh(kz) + \tanh(kh) \cdot \sinh(kz)) \cdot \hat{\varphi}(x, y, 0)](x, y) \tag{9}$$

For ϕ_z on $z = 0$, the initial condition becomes

$$\phi(x, y, 0) = F_{xy}{}^{-1}[k \cdot \tanh(kh) \cdot \hat{\varphi}(x, y, 0)](x, y) \tag{10}$$

1.3 Solving Water Wave Equation

With the results from II, we'll now be able to solve the water wave equations analytically in the Fourier Transform domain. Provide the general water wave equation with boundary condition on $z = 0$:

$$\eta_t = \phi_z \tag{11}$$

where $\eta = \eta(x, y, t)$ and $\varphi = \varphi(x, y, t)$. The Fourier Transform of equation 11 and the results of part II yield

$$\hat{\eta}_{tt} = k \cdot \tanh(kh) \cdot \hat{\varphi}_t \tag{12}$$

The Fourier Transform of the general water wave equation is

$$\hat{\phi}_t + g\hat{\eta} = -\frac{\tau}{\rho}(\omega^2 + \upsilon^2)\hat{\eta} \tag{13}$$

Let $\omega = \xi_1$, $\upsilon = \xi_2$, $k = |\xi|$, and $T = \frac{\tau}{\rho}$; combine equations 12 and 13, this yields

$$\hat{\eta}_{tt} = -\Lambda^2(|\xi|) \cdot \tanh(|\xi|h) \cdot \hat{\eta} \tag{14}$$

where $\Lambda^2(|\xi|) = |\xi|(g + T|\xi|^2)$. Note that $\lim_{h\to\infty} \tanh(|\xi|h) = \begin{cases} 1, & \xi > 0 \\ -1, & \xi < 0 \end{cases}$.

Since $\xi > 0$, we have $\hat{\eta}_{tt} = -\Lambda^2(|\xi|) \cdot \hat{\eta}$. This is again an ODE with solution

$$\hat{\eta}(\xi_1, \xi_2, t) = A(\xi_1, \xi_2)\cos(\Lambda(|\xi|)t) + B(\xi_1, \xi_2)\sin(\Lambda(|\xi|)t) \tag{15}$$

By introducing initial conditions $\eta(x, y, 0) = \eta_o(x, y)$ and $\eta_t(x, y, 0) = \eta_1(x, y)$, the coefficient terms in equation 15 can be solved by first converting these conditions in the domain of Fourier Transform $\hat{\eta}(\xi_1, \xi_2, 0) = \hat{\eta}_o(\xi_1, \xi_2)$ and $\hat{\eta}_t(\xi_1, \xi_2, 0) = \hat{\eta}_1(\xi_1, \xi_2)$. Then apply appropriate substitution. The resulting solution of water wave equation is

$$\hat{\eta}(\xi_1, \xi_2) = \hat{\eta}_o(\xi_1, \xi_2) \cdot \cos(\Lambda(|\xi|)t) \tag{16}$$

2 Linearization

We will use inverse Fourier Transform on equation 16 to compute the evolution of η over a 100 time frame on a 256×256 mesh grid. A problem we encounter, when using the Fast Fourier Transform ($fft2$) in MATLAB, is the difficulty to compute $\Lambda(|\xi|)$ directly. This is because MATLAB does not store explicit numeric values of ξ in the archive. We circumvent this problem by discretizing the two-variable Fourier Transform and Inverse Fourier Transform. This would allow us to extract the numeric values of ξ and compute Λ explicitly.

2.1 Discretization of Two-variable Fourier Transform

The two-variable Fourier Transform can be linearized using Riemann Sum by first approximating on a finite domain:

$$\iint_{-\infty}^{\infty} f(\bar{x})e^{-i\bar{x}\cdot\bar{k}}d\|\bar{x}\| \approx \int_{-L}^{L}\int_{-L}^{L} f(\bar{x})e^{-i\bar{x}\cdot\bar{k}}d\|\bar{x}\|$$

$$\approx \sum_{m=1}^{M}\sum_{n=1}^{N} f\left(x_{m,n}\right)e^{-ix_{m,n}\cdot k_{\xi_1,\xi_2}}\Delta x_m \Delta x_n \tag{17}$$

with $\bar{x} = (x_1, x_2)$, $\bar{k} = (k_1, k_2) \in \mathbb{R}^2$. We discretize $\bar{x}$ and $\bar{k}$ by

i. letting some arbitrary points $x_{m,n} \in [-L, L]$ and $k_{\xi_1,\xi_2} \in \left[-\frac{N\pi}{2L}, \frac{N\pi}{2L}\right]$.
ii. setting N=M so that we will have a square grid with $m, n \in [1{:}\,1{:}\,N\,]$ and $\xi_1, \xi_2 \in [1{:}\,1{:}\,N\,]$.
iii. $\Delta x_m = \Delta x_n = \Delta x = \frac{2L}{N}$ and $\Delta k_1 = \Delta k_2 = \Delta k = \frac{\pi}{L}$, such that $\Delta x \Delta k = \frac{2\pi}{N}$
iv. $x_{m,n} = \bar{m}(\Delta x) + (-L, L)$ and $k_{\xi_1,\xi_2} = \bar{\xi}(\Delta k) + (-l, -l)$ where $\bar{m} = (m, n)$, $\bar{\xi} = (\xi_1, \xi_2)$, and $l = \frac{N\pi}{2L}$

By letting $g(m, n) = f(\bar{m}(\Delta x) + (-L, L))$ and expanding the index of the exponent in equation 17,

$$\hat{f}(\bar{\xi}) = (\Delta x)^2 e^{-i\left(\bar{\xi}(\Delta k)+(-l,-l)\right)\cdot(-L,-L)}\hat{g}(\bar{\xi}) \tag{18}$$

where

$$\hat{g}(\bar{\xi}) = \sum_{m=1}^{M}\sum_{n=1}^{N} g(m, n)e^{-i\bar{m}\cdot(-l,-l)(\Delta x)}e^{-i\bar{m}\cdot\bar{\xi}(\Delta x \Delta k)} \tag{19}$$

2.2 Discretization of Two-variable Inverse Fourier Transform

By definition, the Inverse Fourier Transform of $\hat{f}(\bar{\xi})$ is

$$f(x,y) = \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} \hat{f}(\bar{\xi})e^{i\bar{x}\cdot\bar{k}}\, d\|\bar{k}\|$$

$$\approx \frac{1}{4\pi^2}\iint_{-l}^{l} \hat{f}(\bar{\xi})e^{i\bar{x}\cdot\bar{k}}\, d\|\bar{k}\|$$

$$\approx \frac{\Delta k^2}{4\pi^2}\sum_{\xi_1=1}^{N}\sum_{\xi_2=1}^{N} \hat{f}(\bar{\xi})e^{i\bar{m}\cdot\bar{\xi}(\Delta x\Delta k)+i\bar{m}(\Delta x)\cdot(-l,-l)+i\bar{\xi}(\Delta k)\cdot(-L,-L)+i(-L,-L)\cdot(-l,-l)} \tag{20}$$

Substituting equation 18 into equation 20, this yields

$$f(x,y) = e^{i(\Delta x)\bar{m}\cdot(-l,-l)}\left(\frac{1}{N^2}\sum_{\xi_1=1}^{N}\sum_{\xi_2=1}^{N} \hat{g}(\bar{\xi})e^{i\bar{m}\cdot\bar{\xi}(\Delta x\Delta k)}\right) \tag{21}$$

2.3 Linearization of η

MATLAB sees $fft2$ of $f(\bar{x})$ as

$$fft2(f(\bar{x})) = \sum_{m=1}^{M}\sum_{n=1}^{N} g(m,n)e^{-i\bar{m}\cdot\bar{\xi}(\Delta x\Delta k)}$$

In order to implement the $fft2$ properly, we rearrange equation 19 by letting

$$G(m,n) = g(m,n)e^{-i\bar{m}\cdot(-l,-l)(\Delta x)}\cdot e^{-i(-L,-L)\cdot(-l,-l)+i(-L,-L)\cdot(-l,-l)}$$

$$= f(\bar{x})e^{-i(\bar{m}(\Delta x)+(-L,-L))\cdot(-l,-l)}\cdot e^{i(2Ll)}$$

$$= f(x_1,x_2)e^{i\cdot l\cdot((x_1+x_2)+2L)}$$

such that

$$\hat{g}(\bar{\xi}) = \sum_{m=1}^{M}\sum_{n=1}^{N} G(m,n)e^{-i\bar{m}\cdot\bar{\xi}(\frac{2\pi}{N})} \tag{22}$$

let $A = e^{-i\bar{k}\cdot(-L,-L)}$, the Fourier Transform of $f(\bar{x})$ is further simplified into

$$\hat{f}(\bar{\xi}) = (\Delta x)^2\cdot A\cdot\hat{g}(\bar{\xi}) \tag{23}$$

Using equation 23, we are now ready to linearize the solution of water wave given by equation 16. Since the geometry of a droplet can be approximated by the delta function $e^{-\pi({x_1}^2+{x_2}^2)}$, we take $f(x_1,x_2) = e^{-\pi({x_1}^2+{x_2}^2)}$ in $G(m,n)$ to be our initial function to yield

$$\hat{\eta}_o(\xi_1,\xi_2) = (\Delta x)^2\cdot A\cdot\hat{g}(\bar{\xi}).$$

As a result

$$\hat{\eta}(\xi_1, \xi_2) = (\Delta x)^2 \cdot A \cdot \hat{\hat{g}}(\bar{\xi}) \tag{24}$$

where $\hat{\hat{g}}(\bar{\xi}) = \hat{g}(\bar{\xi}) \cdot \cos(\Lambda(|\xi|)t)$.

Adopting the linearized Inverse Fourier Transform, $\eta(x_1, x_2)$ equals to equation 20 with $\hat{f}(\bar{\xi})$ replaced by $\hat{\eta}(\bar{\xi})$. A short computation will then yield the linearized water wave solution

$$\eta(x_1, x_2) = e^{i(\Delta x)\bar{m}\cdot(-l,-l)} \left(\frac{1}{N^2} \sum_{\xi_1=1}^{N} \sum_{\xi_2=1}^{N} \hat{\hat{g}}(\bar{\xi}) e^{i\bar{m}\cdot\bar{\xi}(\Delta x \Delta k)} \right) \tag{25}$$

3 Simulations

Coding in MATLAB will adopt the formulation of $G(m,n)$, $\hat{\hat{g}}(\bar{\xi})$, and equation 25. $\hat{g}(\bar{\xi})$ is compute by $fft2$, and the double summation in equation 25 is evaluated by $ifft2$. Note that both of $fft2$ and $ifft2$ are implementations of Fast Fourier Transform and Inverse Fast Fourier Transform in MATLAB.

Algorithm

```
 % Perform Evolution of Water Wave solution
for t=1:nframe
    % %%%%%%%%%%%Discretize time step.%%%%%%%%%%%%%%%%
    T_step=(t-1)./dt;
    % %%%%%%%%%%%Compute g_^.%%%%%%%%%%%%%%%%%%%%%%%%%
    [x1,x2]=meshgrid([-L:h:L-h]);
    %fgen is the initial condition function
    y=fgen(x1,x2);
    %Computation of =G(m,n):=Gmn
    Gmn=fgen(x1,x2).*exp(i.*l.*((x1+x2)+2.*L));
    Gmn_shift=fftshift(Gmn);
    %Compute fft2 of Gmn_shift gives g^:=g_head
    g_head=fft2(Gmn_shift);
    % %%%%%%%%%%Compute the time term:%%%%%%%%%%%%%%%%
    [z1,z2]=meshgrid([1:1:N]);
    zz1=((z1-1)*k)-mean(mean(z1-1)*k);
    zz2=((z2-1)*k)-mean(mean(z2-1)*k);
    azz=sqrt(zz1.^2 + zz2.^2);
    Time=cos(sqrt(g*azz + tau*azz.^3)*T_step);
    % %%%%%%%%%%Compute g_double_^:=g_double_head%%%%%
    g_double_head=Time.*g_head;
    % %%%%%%%%%%Inverse Fourier Transform%%%%%%%%%%%%
    [m,n]=meshgrid([1:1:N]);
    B=exp(-i.*h.*l.*(m+n));
    ETA=B.*ifft2(g_double_head);
    eta=ifftshift(ETA);
    surf(x1,x2,real(eta));
    axis([-wf wf -wf wf -hf hf]);
drawnow;
W(:,t)=getframe(gcf);
end;
```

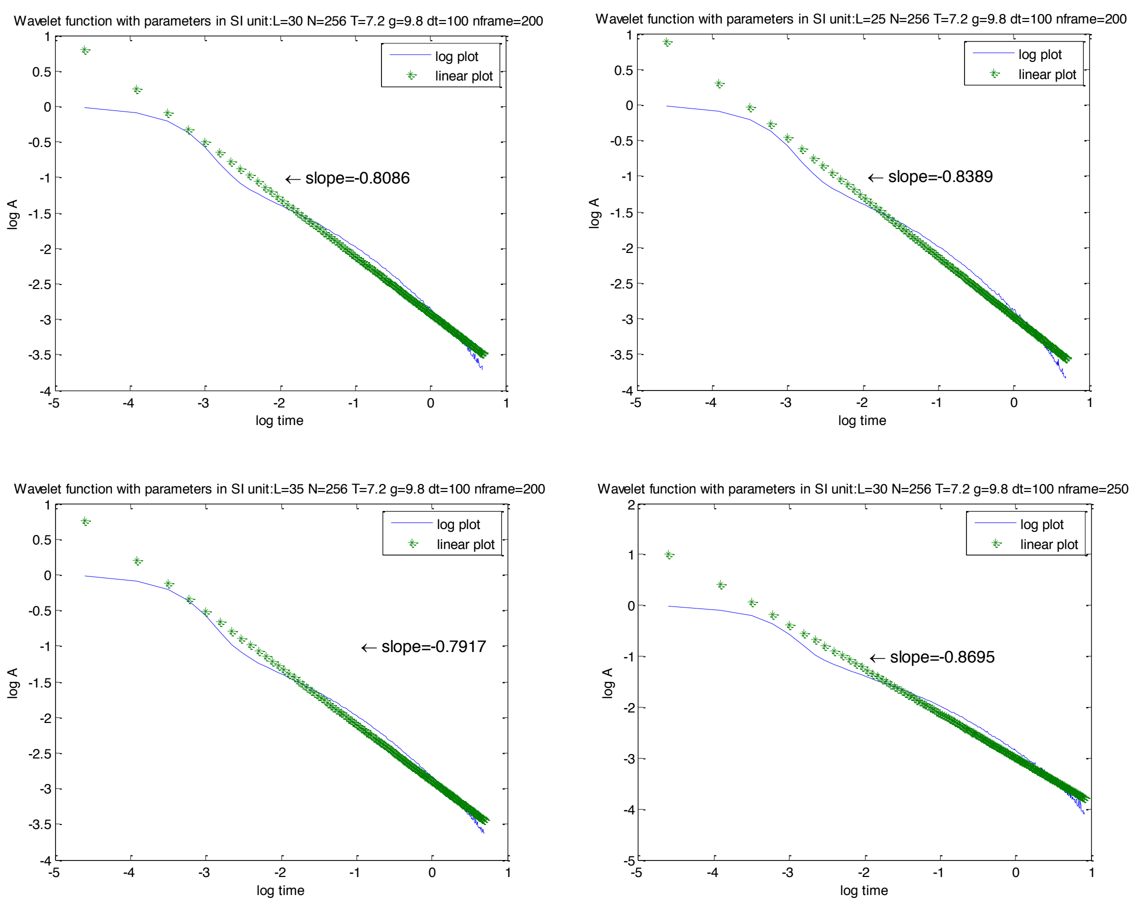
4 Results

The validity of the simulation was verified by comparing the numerical result of wave velocity and wave length to the actual water wave motion captured by our collaborator at the CCMA computing laboratory in Penn State University.
Spirn and Wright proved in their paper that

$$\max_{\bar{x}\in\mathbb{R}^2}\{\eta(\bar{x},t)\} \approx Ct^{\gamma} \tag{26}$$

γ is the rate of linear dispersive decay and C is a constant. Let $\mathrm{H} = \max_{\bar{x}\in\mathbb{R}^2}\{\eta(\bar{x},t)\}$ and take the logarithm of equation 26: $\log H = \gamma \log t + \log C$. The linear slope of the logarithm graph will then be the estimates of dispersive decay. The output data of the simulation allows us to extract the coordinates of crests. The fixed values of parameters are given as $T = 7.2$, $g = 9.8$, where the values of L, N, $nframe$, and dt are adjustable to control the resolution of simulations. Numerical estimates are shown in the following figures. Our numerical values of γ is fluctuated between -0.8 which is closed to the analytic values of $\gamma = -5/6$ proven by Spirn and Wright. The accuracy of the estimates is affected by the limited resolution of the simulation and the approximated initial geometry of the droplet using delta function.

Figure 1

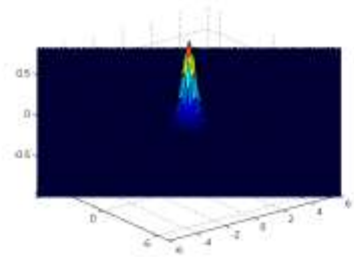


References

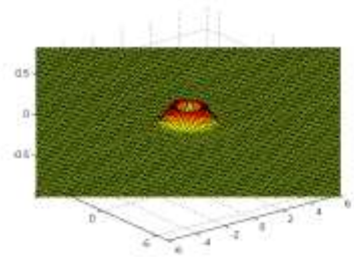
[1] Daniel Spirn, and J. Douglas Wright, Linear Dispersive Decay Estimates for the 3+1 Dimensional Water Wave Equation with Surface Tension. Canadian Mathematical Bulletin doi:10.4153/CMB-2011-057-3

[2] Harvey Segur, Diane Henerson, Water Waves-theory and experiment. CBMS Regional Conference. May 13-18, 2008.

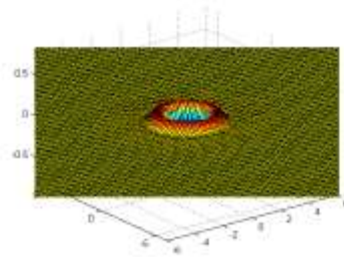
Simulations



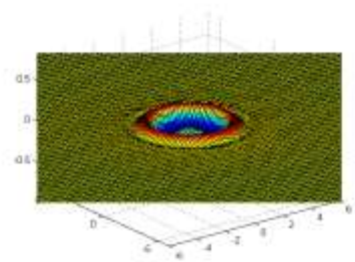
t=1



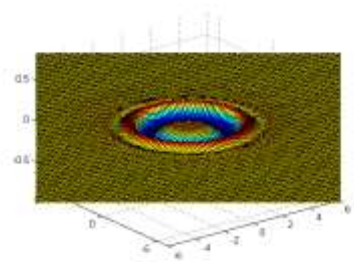
t=10



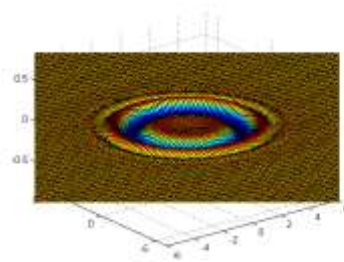
t=20



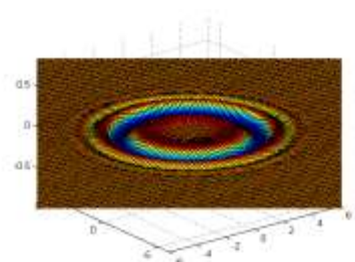
t=30



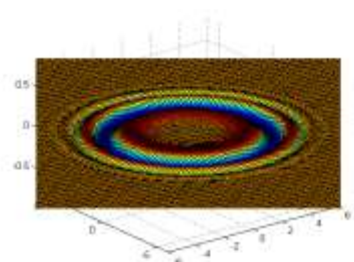
t=40



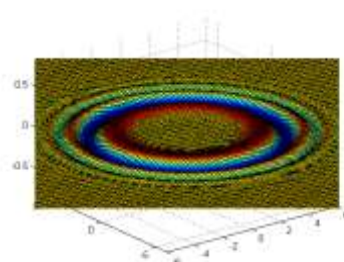
t=50



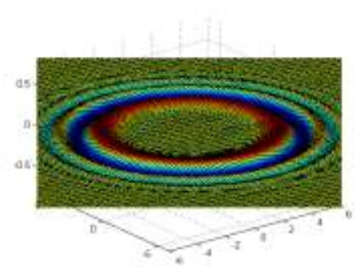
t=60



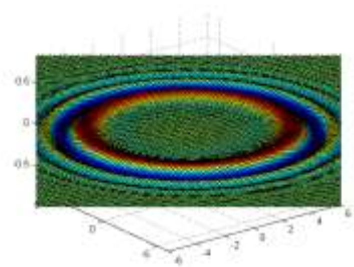
t=70



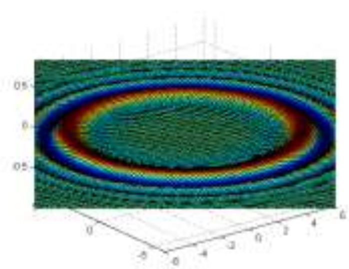
t=80



t=90



t=100



t=110