Simplified wave modelling

John C. Bancroft

ABSTRACT

Wave motion is modelled using the acoustic wave equation and implemented using MATLAB. This method requires two initial conditions that are introduced using a simple wavelet on a one dimensional propagator such as a string, spring, or wire. This model is expanded to two dimensions that illustrate plane-wave propagation, boundary effects, and Huygen's wavelets.

WAVE ON A STRING.

Assume we want to model transverse wave motion on a string (spring, wire, etc). We will compute the motion of a wavelet at equal time increments that simulate photos to evaluate the motion.

We start with a Gaussian-shaped wavelet illustrated below in Figure 1 that represents a photograph of the transverse displacement. We have chosen the width of the wavelet to be large, relative to the sample interval so that linear interpolation provides an adequate description, i.e. the sample rate is approximately ten times the maximum frequency to minimize grid dispersion. Given this information:

- What is the direction of motion?
- What will be the shape in the next photograph?

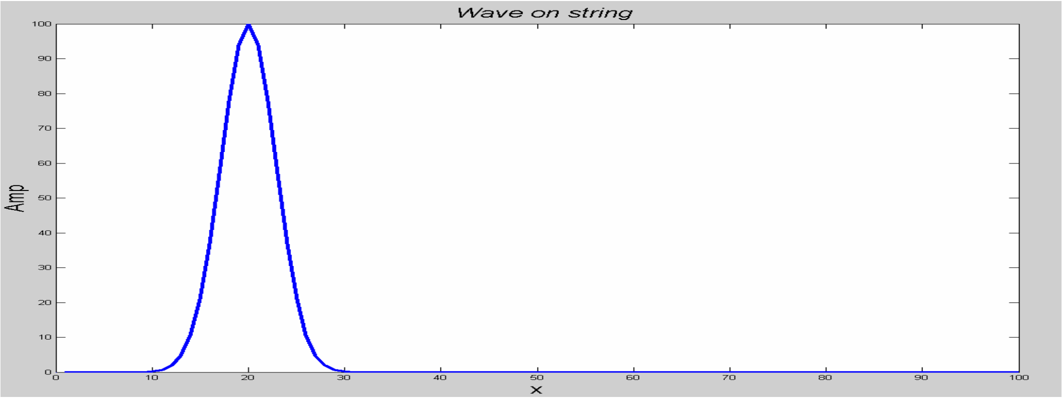


FIG 1: A representation of a Gaussian shaped wavelet on a string.

Is the wavelet in Figure 1 going to move to the left, the right or bounce up and down? We simply don't know because we don't have enough information. How do we provide that additional information? Let's examine the wave equation to find out.

The string has one distance dimension, *x*, and we define the displacement *p(x)*, at different time intervals such as *p(x, t=1)*, *p(x, t=2),* etc. It therefore becomes convenient to define a 2D array, *p(x, t)*, that defines the amplitude of the strings displacement at a given location, *x*, and time, *t*. The movement of energy on the string is governed by the one dimensional acoustic wave-equation

$$\frac{\partial^2 p(x,t)}{\partial x^2} = \frac{1}{v^2}\frac{\partial^2 p(x,t)}{\partial t^2}. \tag{1}$$

We will use the finite difference method to approximate the wave-equation. A second derivative of a function, *f(x)*, is approximated in a discrete form of *f* at position, *n*, i.e. f_n, and is approximated by

$$\frac{\partial^2 f(x)}{\partial x^2} = \frac{f_{n-1} - 2f_n + f_{n+1}}{\delta x^2}, \tag{2}$$

where the increments of f_n are δx. The finite difference equation for the wave equation becomes:

$$\frac{p_{i-1,j} - 2p_{i,j} + p_{i+1,j}}{\delta x^2} = \frac{1}{v^2}\frac{p_{i,j-1} - 2p_{i,j} + p_{i,j+1}}{\delta t^2}. \tag{3}$$

This equation can be represented in graphical form in the following figure, with the increment, *i*, representing distance to the left, and *j*, the time increment that increases vertically.

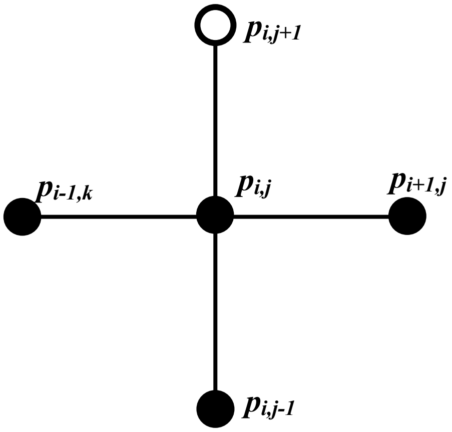


FIG 2: Finite difference model of full wave-equation.

We wish to compute the position of the string at a new time, given some initial condition. We will choose $p_{i,\,j+1}$ as the unknown value in the finite-difference equation because it is a single value at a new time, giving

$$p_{i,j+1} = 2p_{i,j} - p_{i,j-1}\frac{v^2\delta t^2}{\delta x^2}\left(p_{i-1,j} - 2p_{i,j} + p_{i+1,j}\right). \tag{4}$$

According to this equation, we need to know the position of the string at the two previous times, *j*, and *j-1*; .i.e. the two initial conditions of the string. Shown below is the finite-difference operator positioned at the at i^{th} spatial location to calculate the amplitude on the string for the third time, at *j=3*.

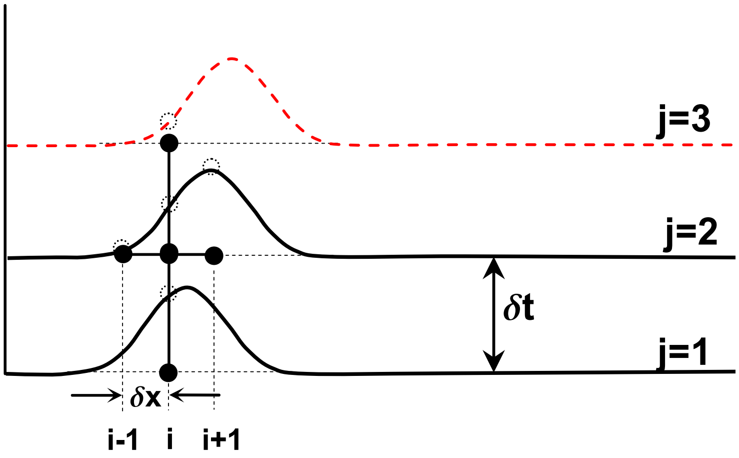
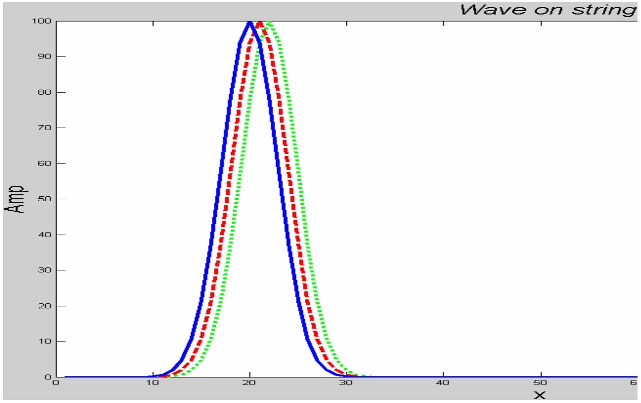


FIG 3: The first two initial conditions at *j =1* and *2* to compute the first line of samples at *j=3*.

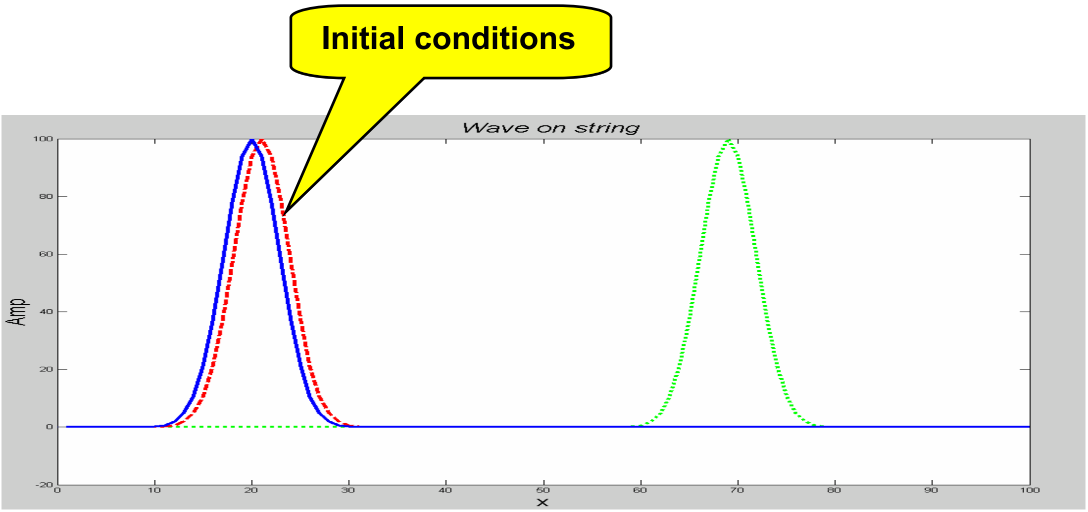
If we desire to model a wave travelling down the string, then we need to start with a wavelet on the string at time *j=1*, and the same wavelet repeated on the string at time *j=2* but moved to a new spatial location. The location on the *j=2* string is critical and must have a spatial increment defined by the wave velocity, *v*, and the time increment, *δt*. These two initial conditions would also match any two sequential photos of the wave moving down the string. They are shown above, along with the finite-difference operator, illustrating the computation of the amplitudes on the string at time, *j=3*. (The above figure has used a very large *δ*x on the operator for the purposes of illustration only).

Computer simulations are illustrated in Figure 4, with (a) showing the amplitude of a Gaussian wavelet on the string at the first two times and the resulting calculation of the wavelet at the third time, *k=3*. The amplitudes at succeeding times may be calculated from the previous two times as illustrated in (b) that shows the wavelet at a time increment, *k=50*. The initial two wavelets are also shown to demonstrate how well the amplitude and shape of the wavelet has been preserved after 48 iterations.

The complete MATLAB code for producing Figures 4 and 5 is listed in Appendix A. The part of the code that propagates the wavefield is six lines that are encircled to illustrate the simplicity. The coding method is similar to FORTRAN and does not take advantage of vector math that is available to a programmer.



a)



b)

FIG 4: The first two initial conditions and the computation of a) the next third time increment, and b) at 50 time increments later. Note the preservation of the shape and amplitude.

The full set of iterations are shown below in a three-dimensional view that shows the amplitude of the wavelet at the incremental times. This is a trivial task in MATLAB that simply displays 2D array, *p(t, x)*, in Figure 5 with the call, "`figure (3); mesh(p);`", highlighted in the program listing in appendix A. This figure has zoom and rotation features that allow the programmer to view the image from any position. The simple coding of this program, along with the powerful MATLAB graphics, provide an excellent analysis tool.

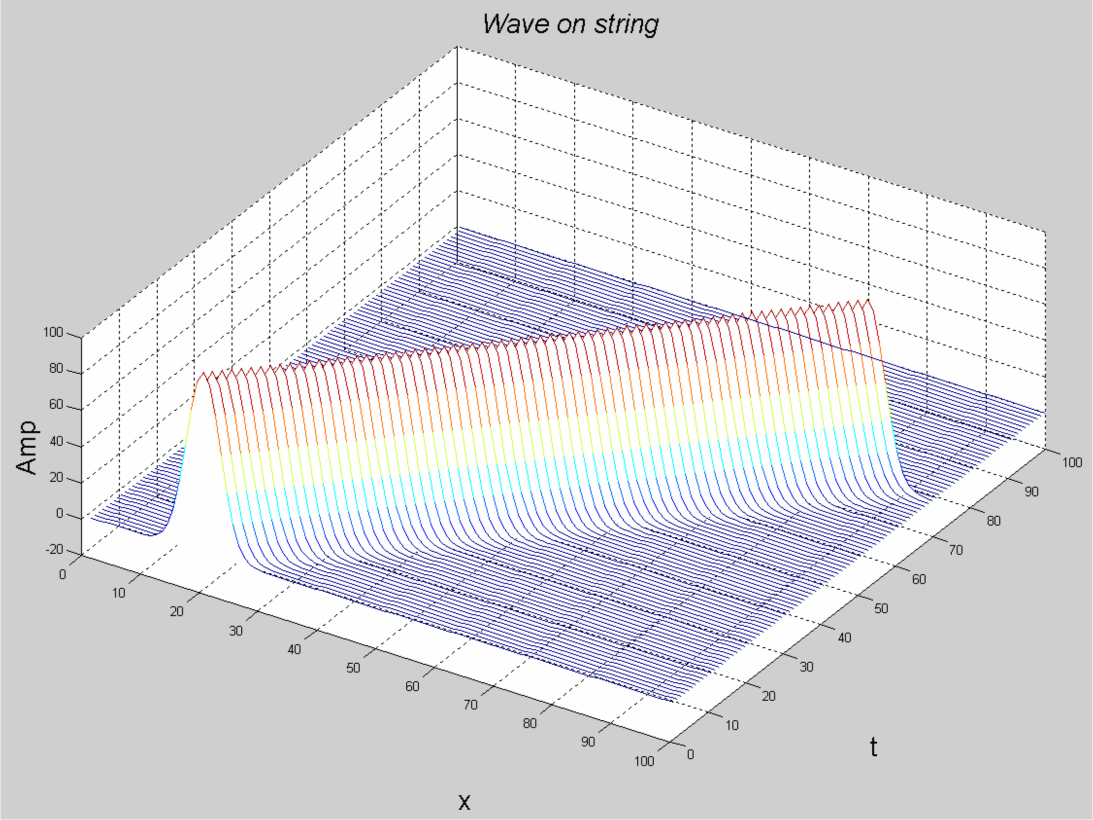
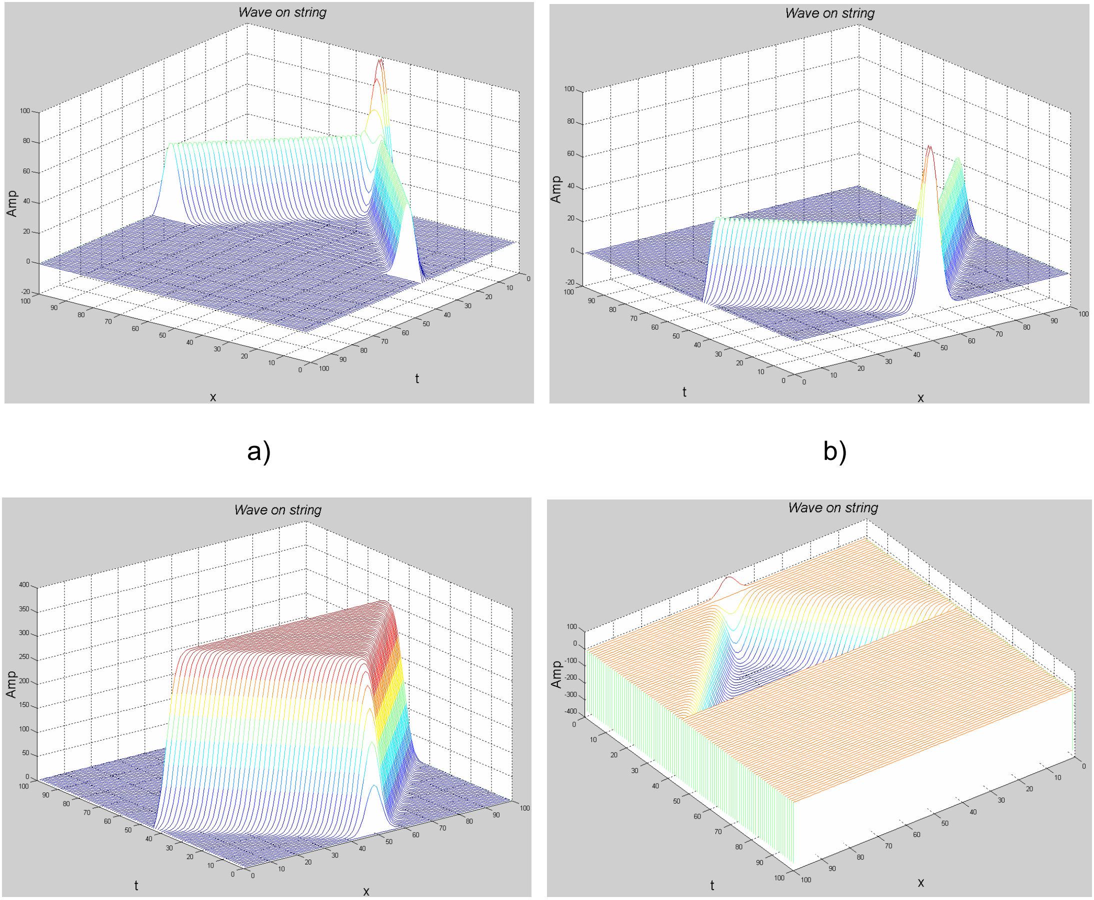


FIG 5: A full set of time images of the displacement

To illustrate the importance of the initial conditions, various configurations of the first two wavelets are shown in the following figures. Note the effect on the subsequent propagations.

- Figure 6a and b show the first two wavelets at the same amplitude and location at the centre of the string. The result is two wavelets moving away from each other. (These two images demonstrate two different perspective views of the same figure).
- Figure 6c shows the two initiating wavelets at the same location, but the second is twice the amplitude, producing an interesting deformation of the string. A similar type of deformation is obtained by zeroing the amplitude of the second wavelet as viewed in Figure 6d.

We refer to the setting of the two initial conditions or the defining the deformation of the string at the first two time intervals as “exciting” the string.



a)

b)

c)

d)

FIG 6: Various wave propagations that result when the initial conditions are varied, a) when the location and amplitude are the same, b) the same as (a) but a different view, c) with the same location but the amplitude of the second initial condition has twice the amplitude, and d) when the second initial condition is zero.

2D WAVE PROPAGATION

The propagation of energy on a 2D plane is also quite simple to program by extending the concepts of the 1D program. The two-dimensional wave equation,

$$\frac{\partial^2 p(x,z,t)}{\partial x^2}+\frac{\partial^2 p(x,z,t)}{\partial z^2}=\frac{1}{v^2}\frac{\partial^2 p(x,z,t)}{\partial t^2}, \tag{5}$$

becomes the finite difference equation,

$$\frac{p_{i-1,j,k}-2p_{i,j,k}+p_{i+1,j,k}}{\delta x^2}+\frac{p_{i,j-1,k}-2p_{i,j,k}+p_{i,j+1,k}}{\delta z^2}=\frac{1}{v^2}\frac{p_{i,j,k-1}-2p_{i,j,k}+p_{i,j,k+1}}{\delta t^2}. \tag{6}$$

Solving for the single time sample at *k+1* we get:

$$p_{i,j,k+1} = 2p_{i,j,k} - p_{i,j,k-1} + v^2\delta t^2\left(\frac{p_{i-1,j,k} - 2p_{i,j,k} + p_{i+1,j,k}}{\delta x^2} + \frac{p_{i,j-1,k} - 2p_{i,j,k} + p_{i,j+1,k}}{\delta z^2}\right). \quad (7)$$

This difference equation is illustrated in Figure 7. A plane through the central points represents time at the k^{th} level and the one sample point below this plane is a sample on the previous time layer at *k-1*. The single point to be computed (the circle) lies on the upper plane at time *k+1*.

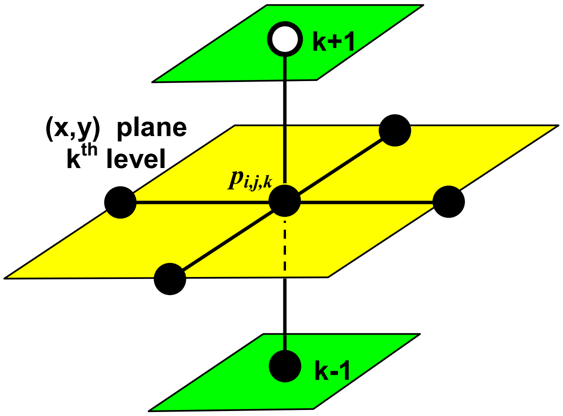
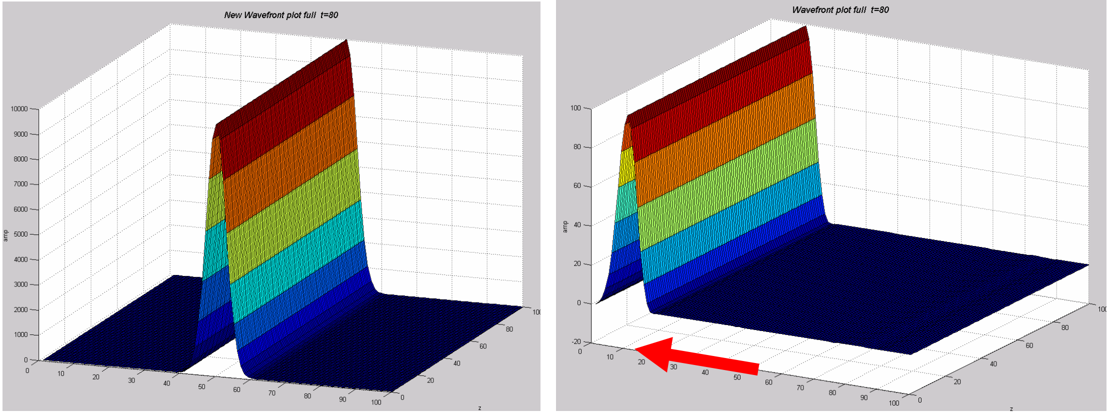


FIG 7: Schematic representation of the three-dimensional (*x, z, t*) operator.

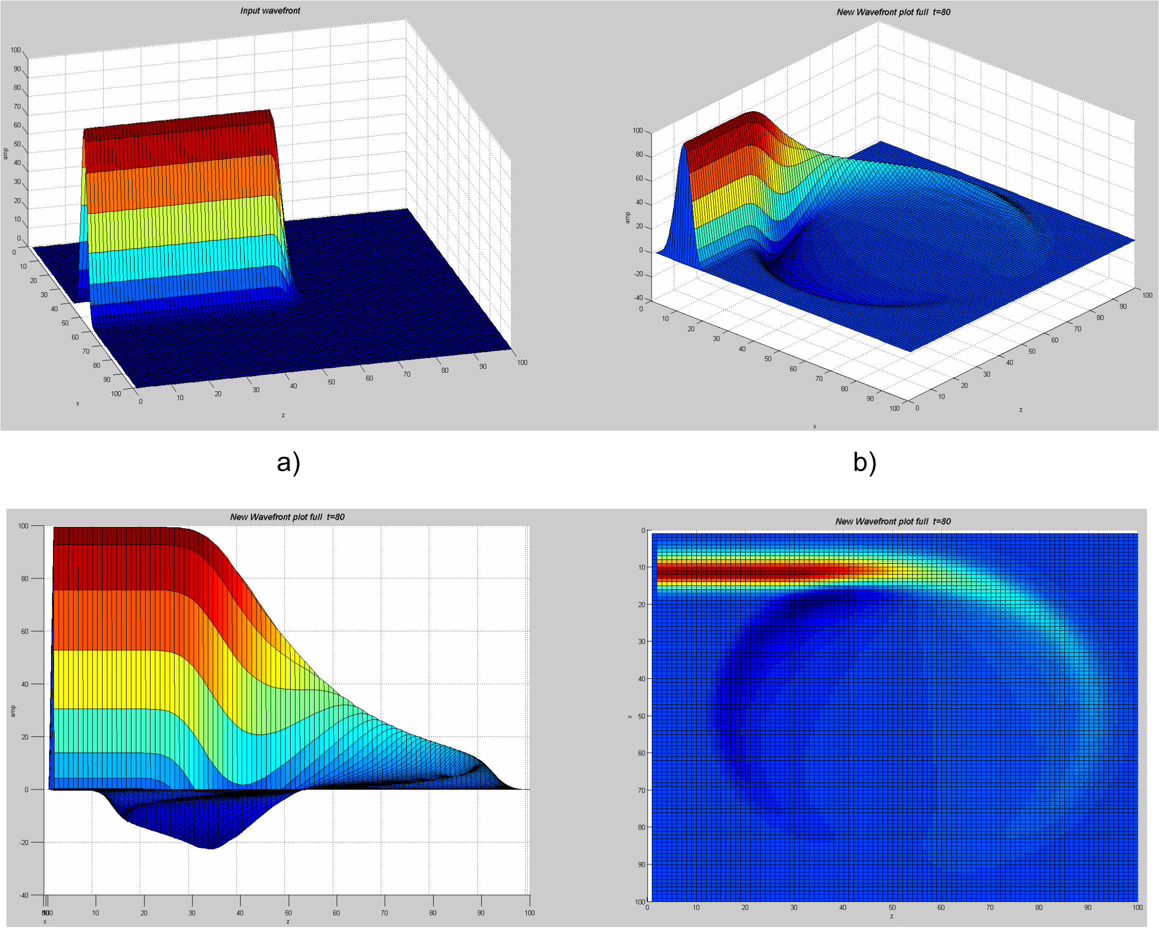
We therefore need two planes of the wavefield (at *k-1*, and *k*) to start the propagation. As in the 1D case, it is critical that the first two planes represent the desired initial conditions. In Figure 8a below, the initial wavefield starts at the centre of the plane and then, after 80 time iteration of computing the wavefield, we get the wavefield shifted to the left as evident in part (b). Note that the amplitude and shape of the wave field has been preserved.



a) b)

FIG 8: A plane wave designed to propagate to the left, a) at time zero, and b) at after 80 time increments.

Figure 9a show the an initialization of a plane-wave that only extend part way across the surface. The resulting wavefield after 80 time iterations is shown in various perspective views in panels b), c), and d). Note the unusual effect of this result as energy has propagated opposite to the direction to the main part of the wavefront.



a)

b)

c)

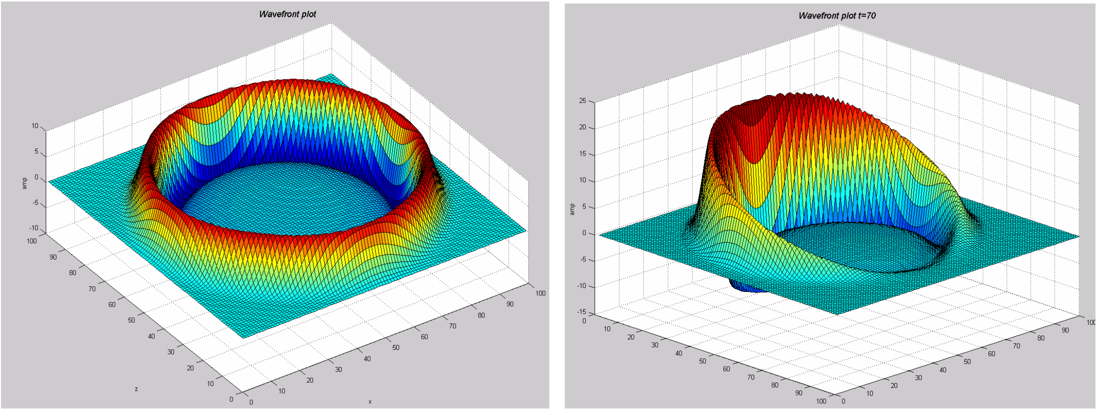
d)

FIG 9: A portion of a plane-wave is propagated 40 time increments with: a) the initial location with a rounded edge, b) a perspective view after 40 time lags, c) side view, and d) a plan view.

Figure 10 contains two images that result from a 3-D Gaussian-shaped wavelet that is circular in (x, z) as displayed in Figure 11a. The circular wavefront in part (a) in Figure 10 was excited by keeping the two excitation wavelets at the same location, while part (b) was excited by moving the second wavelet to the left with the propagation velocity. Note that even though the wavelet was initially propagated to the left, some energy also moves in the opposite direction.

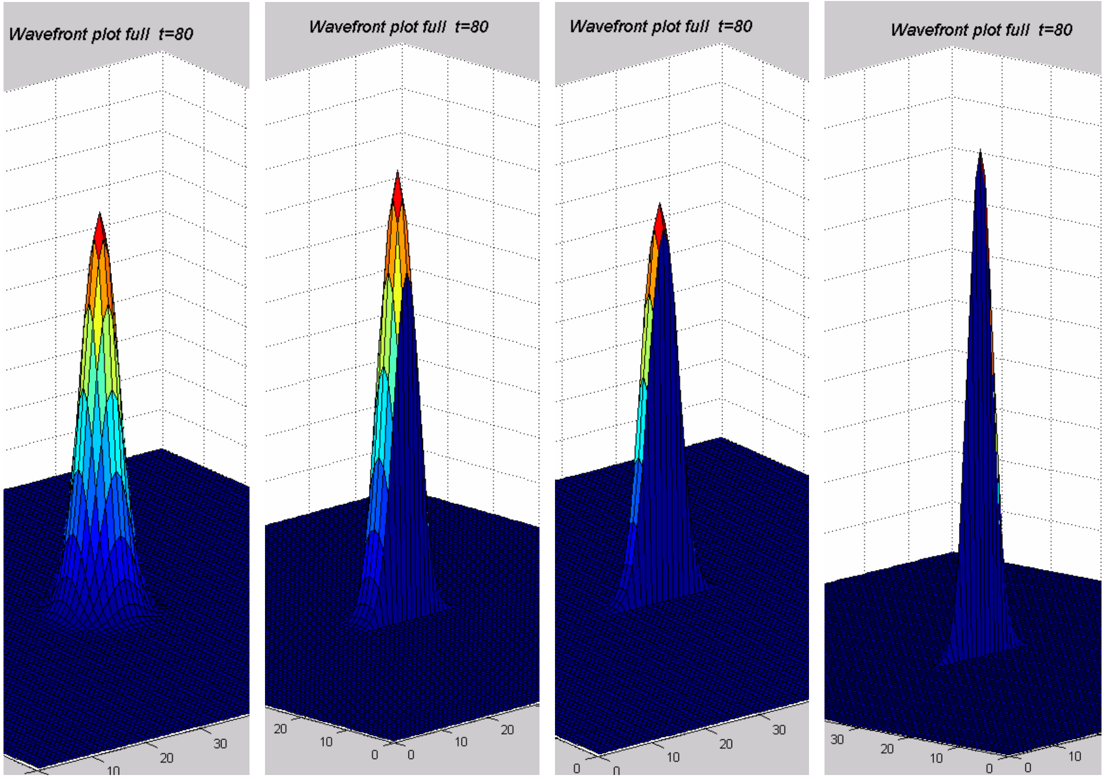
Figure 11 displays three additional wavelets in (b), (c), and (d) that are truncated to widths of five, three, and one sample. The intent of these wavelets is to represent some form of decomposition of a plane wave: i.e. part (d) is just one slice from the wavefront. This slice is definitely aliased in the direction parallel to the wavefront and we should expect some form of dispersion during propagation.

Each wavelet is propagated in the direction of the wavefront and creates the circular images on the left side of the corresponding parts of Figure 12. These circular images represent some form of Huygen's wavelets that are used to propagate wavefronts. The right side of these figures are the wavefront that are reconstructed from the Huygen's wavelets. This was achieved by assuming source points were positioned at the original location of the wavefront (i.e. a continuous line of samples) and the resulting Huygen's wavelets summed.



a) b)

FIG 10: Two wavefront responses from a Gaussian shaped wavelet at the centre with different initial conditions; with a) the initial wavelets at the same location, and b) when the second initial wavelet is shifted to the left with the medium's velocity.



a) b) c) d)

FIG 11: Wavelet sources with a) the full circular wavelet, b) truncated to five points wide, c) three points wide and d) one point wide.

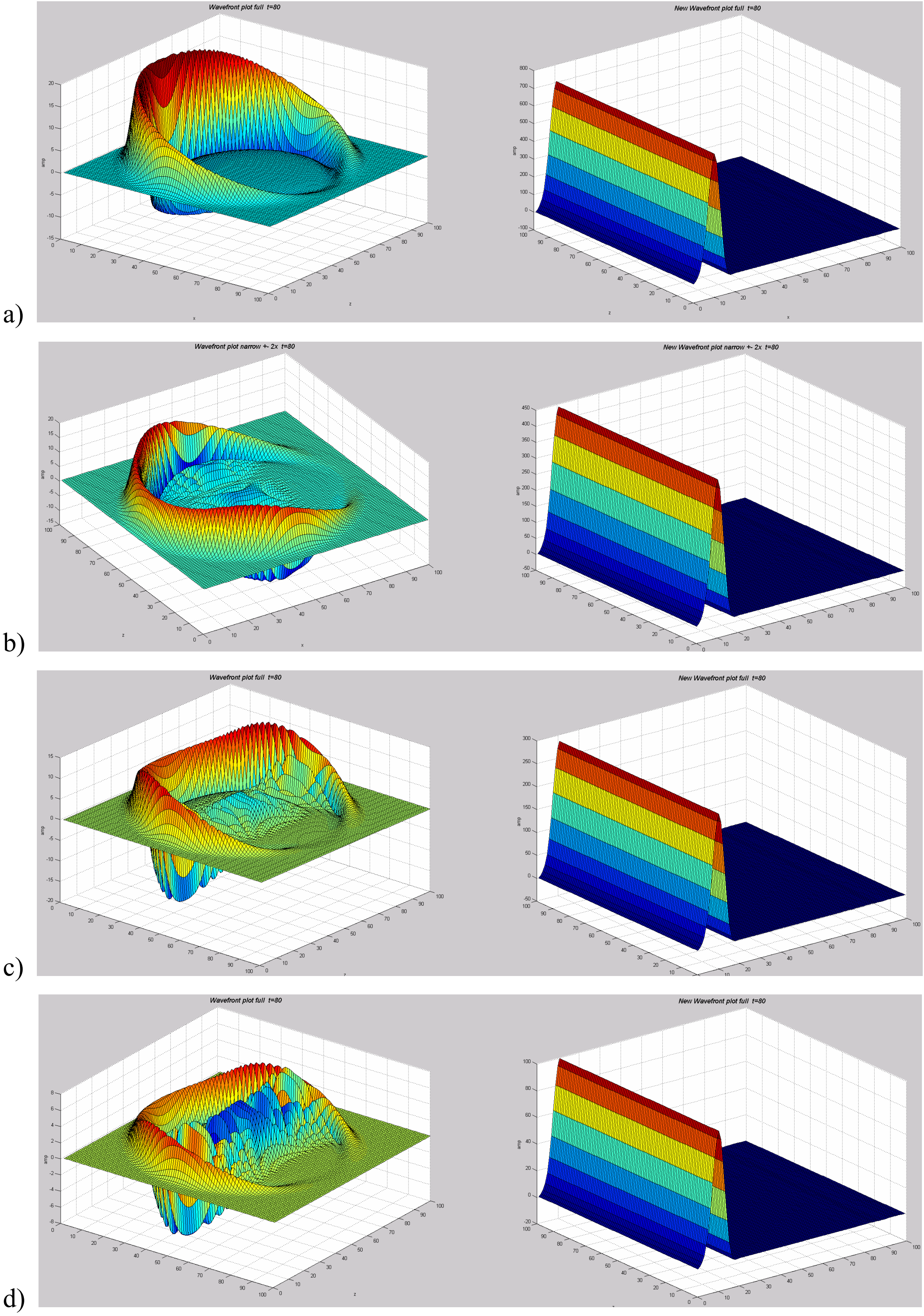


FIG 12: Huygen's wavefronts on the left side of each image when the source wavelet is: a) circular; b) 5 points wide; c) 3 points wide; and d) 1 sample wide. The right side is the corresponding reconstruction of the wavefront from an array of source points.

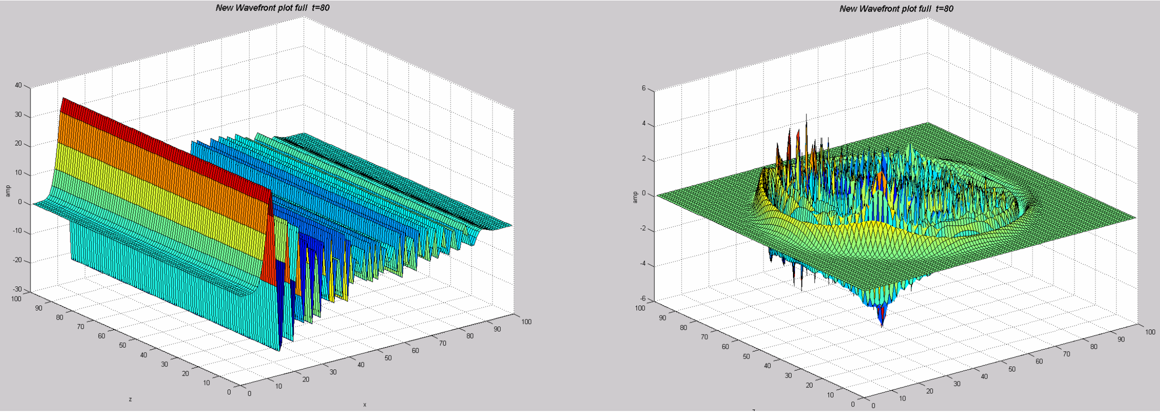


FIG 13: Reconstructed wavefront and Huygen's wavelet approximation when a single sample spike is propagated as part of a wavefront.

The results of using a single sample for excitation is illustrated in Figure 13. Aliasing in the direction of wavefield propagation causes excessive dispersion of the energy.

The non-aliased circular/Gaussian wavelet of Figure 11a was used to demonstrate the formation of the Huygen's wavelet. This is illustrated in Figure 14, which shows the propagating wavefields at time intervals of 5, 10, 20, 30, 40, 50, 60, and 80. Note the formation of the phase-shift as the amplitudes become negative, and that some energy is propagated in the reverse direction.

The Huygen's wavelet energy lies on a circular path that extends in all directions around the excitation point. The amplitude tapers to zero in the direction opposite the excitation direction, and then to a maximum in the direction of excitation or propagation. Figure 15 is a side view of the Huygen's wavelet that shows the peak amplitude has a linear slope that rises in the direction of propagation. This amplitude is described in theory as 1+cosine(θ), where θ is the angle from the direction of propagation. Also note that the phase-shift appears to be 45 degrees.

COMMENTS

The images in Figure 10 illustrate the difference in the energy radiated from an isotropic point source and that of an element on a wavefront.

The small portion of MATLAB code that propagates the 2D energy is included as Appendix B.

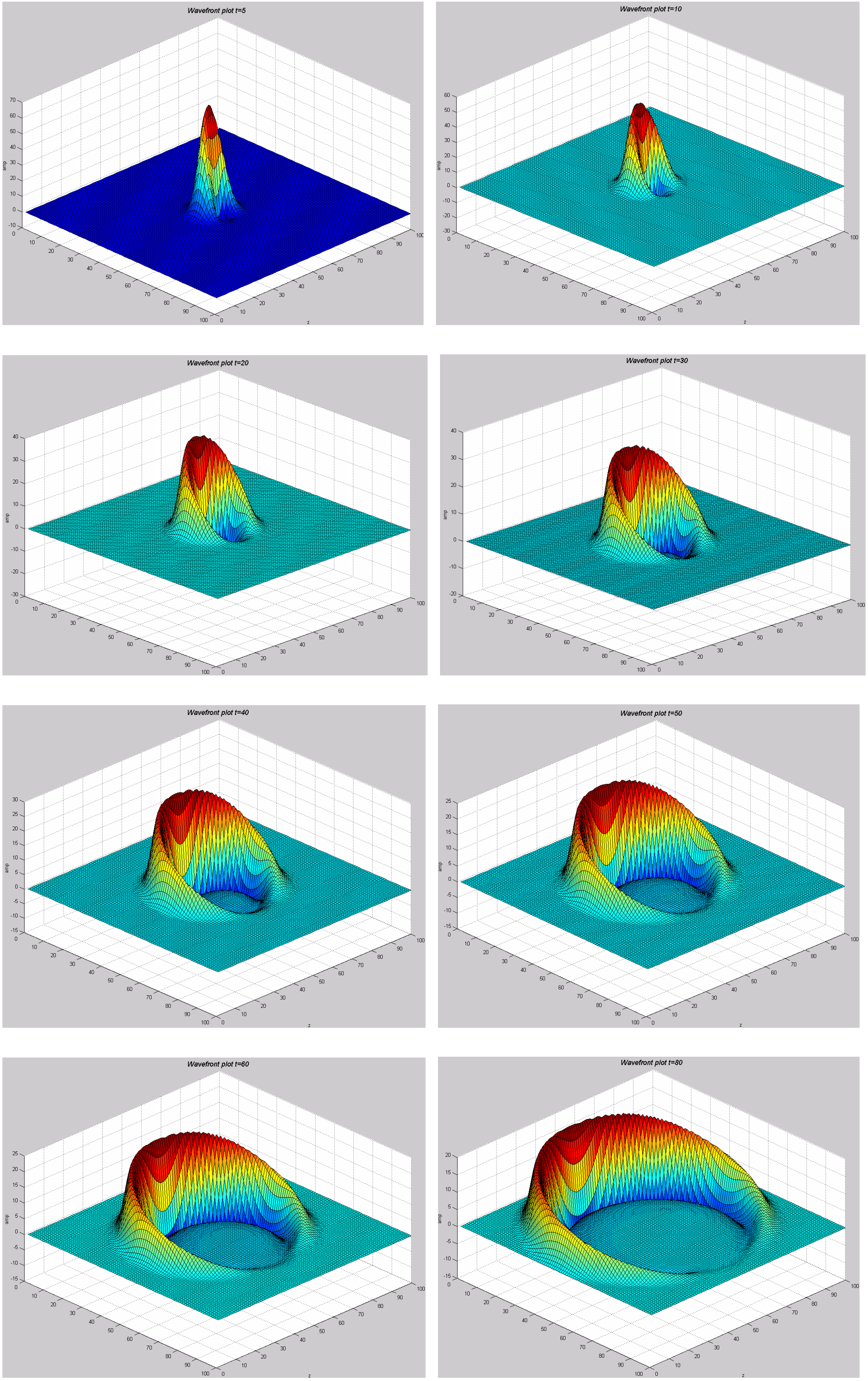


FIG 14: Formation of a Huygen’s wavelet at 5,10, 20, 30, 40, 50, 60, 80 time units.

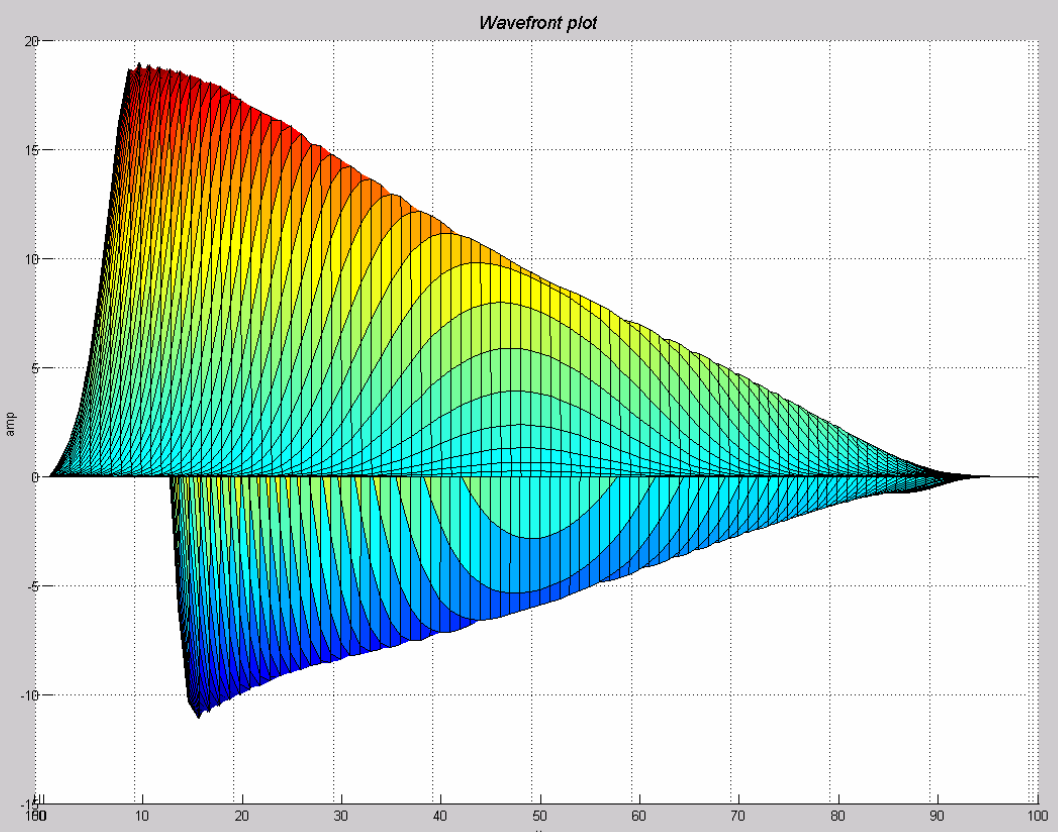


FIG 15: Side view of Huygen's wavelet displaying the amplitude.

CONCLUSIONS

Wave propagation can be illustrated using simple programs written and displayed in MATLAB.

ACKNOWLEDGEMENTS

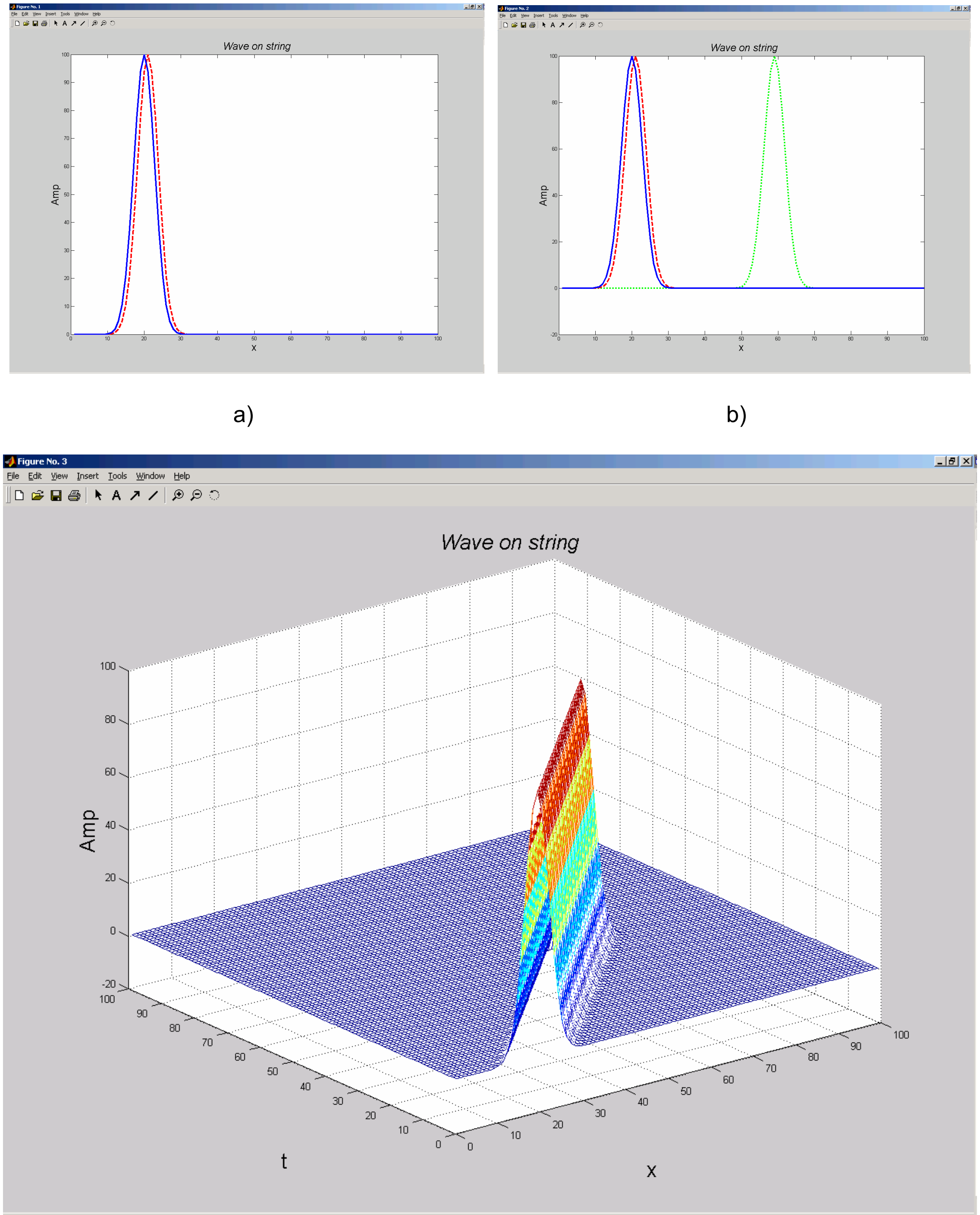
We acknowledge NSERC and the CREWES sponsors for their continued support.

APPENDIX A

MATLAB code for modelling 2D data. The two parts encircled contain the wave propagation code and the function call to plot the view of the full movement of the waveform.

```
%  Wave on string
clear
v = 1000.0; % Velocity
dx = 1.0;  % x increment
dt = 0.001; % time increment
nx = 100;  %Number of x samples
nt = 100;  %Number of z samples
```

```
p = zeros(nt,nx);  % Matrix for string
s1 = zeros(nx);   % String at time 1
s2 = zeros(nx);   % String at time 2
s40= zeros(nx);   % String at time 50
xary=zeros(nx);   % Plot axis

%  Define the position of the string at the first two times
for ix = -10:10
   p(1, ix+20 ) = 100*exp(-(ix^2 )/16.0 );
   p(2, ix+21 ) = 100*exp(-(ix^2 )/16.0 );
end

%  Loop for each time increment: limit time to prevent
distortion due to boundary reflection
for it = 3: nt-55
  for ix = 2:99
  p(it,ix) = 2*p(it-1,ix) - p( it-2, ix) +
  ((v*dt/dx)^2)*(p(it-1,ix-1)-2*p(it-1,ix)+p(it-1,ix+1));
  end
end

% Get singel copies of the two excitation arrays and one
latter array.
for ix = 1:nx
  xary(ix)=ix;
  s1(ix) = p(1, ix);
  s2(ix) = p(2, ix);
  s40(ix) = p(40, ix);
end

% Plot the data
figure (1); plot(xary,s2,'r--',xary,s1,'b','LineWidth',3);
xlabel('x','FontSize',20), ylabel('Amp','FontSize',20)
title(' \it{ Wave on string}', 'FontSize',20)

figure (2); plot(xary,s40,'g:',xary,s2,'r--
',xary,s1,'b','LineWidth',3);
xlabel('x','FontSize',20), ylabel('Amp','FontSize',20)
title(' \it{ Wave on string}', 'FontSize',20)

figure(3); mesh( p);
xlabel('x','FontSize',20), ylabel('t','FontSize',20),
zlabel('Amp','FontSize',20)
title(' \it{ Wave on string}', 'FontSize',20)
```



a)

b)

c)

FIG A1: MATLAB windows with results of the 1D modelling code showing: a) the initial conditions on a string; b) initial conditions on a string and the wave after 40 time increments; and c) all 40 time increments on the string.

APPENDIX B

Portion of MATLAB code for modelling 3D data that propagates the wave.

```
%*********************************************************
% Compute each time layer
for it = 3:ntd
it
 % Compute each x trace
 for ix = 2:nx-1
   %Compute each sample in trace
   for iz = 2:nz-1
     p = vol(ix, iz, it-1);
     ptm1 = vol(ix, iz, it-2);
     pzm1 = vol(ix, iz-1, it-1);
     pzp1 = vol(ix, iz+1, it-1);
     pxm1 = vol(ix-1, iz, it-1);
     pxp1 = vol(ix+1, iz, it-1);
       %solve wave-equation
     ptp1 = 2.0*p - ptm1 + V^2*dt^2*( (pxm1 -2*p
         +pxp1)/dx^2 + (pzm1 -2*p + pzp1)/dz^2 );
     vol(ix, iz, it) = vol(ix, iz, it) + ptp1;
   end
 end
%********************************************************
```