

# Geese honk, but they don't wave

## Sinusoidal waves

---



### CS112 Scientific Computation

Department of Computer Science  
Wellesley College

## Creating vibes with GUIs

We will build a GUI-based teaching tool to help us identify the parameters of a **sinusoidal wave**\*



\* Based on an idea suggested by Jim McClellan at Georgia Tech

Waves 14-2

## Sine functions

General form of a sine function:

$$\text{amp} * \sin(2 * \pi * \text{freq} * t + \text{phi})$$

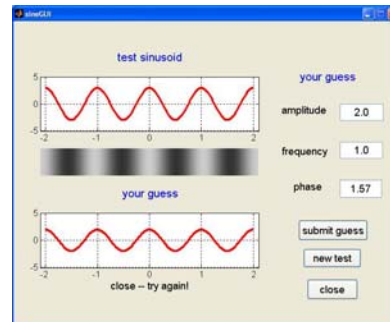
where

$t$  = time

amp = sine wave amplitude

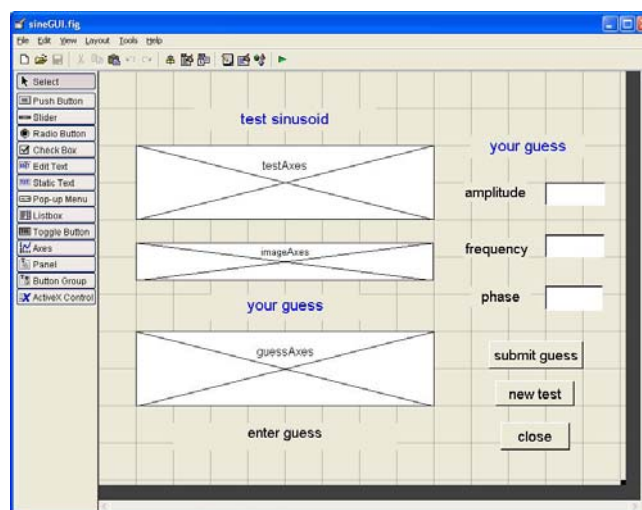
freq = frequency

phi = phase, or shift in time



Waves 14-3

## Using GUIDE's Layout Editor



Waves 14-4

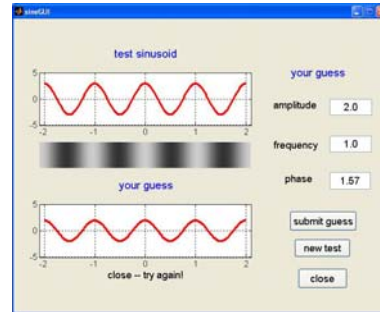
## sineGUI

Program randomly generates sinusoidal waves and displays "test" sinusoid in upper plot

A pattern of brightness that varies in a similar sinusoidal fashion is displayed as an image

User enters amplitude, frequency, and phase of sine wave

When **submit guess** button is pressed, the user's sine wave is drawn at the bottom and a message indicating correctness of guess is printed



Waves 14-5

## Parameters

The program selects three parameters from the following sets of values

**amplitude:**

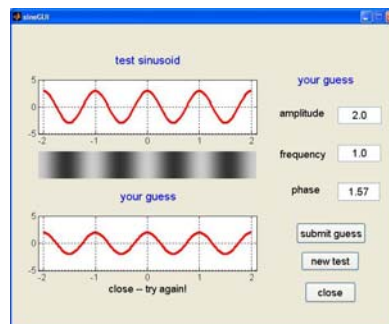
1 2 3 4 5

**frequency:**

0.25 0.5 0.75 1.0 1.5 2.5 3.0

**phase\*:**

-2.36 -1.57 -0.78 0.0 0.78  
1.57 2.36 3.14



\* Phase values are angles, in radians, corresponding to -135°, -90°, -45°, 0°, 45°, 90°, 135°, 180°

Waves 14-6

## Recall functions defined in `sineGUI.m`

`sineGUI`: top-level function at the beginning of the file that is called from the Command Window. This function initializes the GUI program and opens the GUI window. *We will not modify this function*

`sineGUI_OpeningFcn`: executed just before the GUI window is made visible. We will modify this function to set up an initial test sine wave for the program

`sineGUI.m_OutputFcn`: returns outputs to the Command Window. *We will not modify this function*

Waves 14-7

## Changes to `sineGUI_OpeningFcn`

`% --- Executes just before sineGUI is made visible.`

`function sineGUI_OpeningFcn(hObject, eventdata, handles, varargin)`

`% set up parameters and display for initial test sinusoid`

`handles.amplitude = 3;`

`handles.frequency = 1;`

`handles.phase = 1.57;`

`displaySine(handles, 1);`

Always the same  
first time around

`% Choose default command line output for sineGUI`

`handles.output = hObject;`

`% Update handles structure`

`guidata(hObject, handles);`

Don't forget to update  
global structure!

Waves 14-8

## function displaySine (then)

```
function displaySine (handles, newTest)
if newTest
    axes(handles.testAxes); ← direct display
                            to testAxes
    time = -2:0.02:2;
    sine = handles.amplitude * sin(2.0*pi*handles.frequency .* ...
                                    time+handles.phase);

    plot(time, sine, 'LineWidth', 3, 'Color', 'r');
    axis([-2.1 2.1 -5.1 5.1]), grid on
    axes(handles.imageAxes);
    image = zeros(25,201);
    for i = 1:25
        image(i,:) = sine(1,:);
    end
    imshow(image, [-5 5]);
    axes(handles.guessAxes);
    plot(0,0);
    axis([-2.0 2.0 -5.1 5.1]), grid on } clear
                                        guessAxes
                                        } set up
                                        and display
                                        image matrix
```

Waves 14-9

## function displaySine (else)

```
function displaySine (handles, newTest)
...
else
    axes(handles.guessAxes); ← direct display
                              to guessAxes
    amp = str2double(get(handles.ampBox, 'String'));
    freq = str2double(get(handles.freqBox, 'String'));
    phase = str2double(get(handles.phaseBox, 'String')); } get
                                                            user's
                                                            guesses
    time = -2:0.02:2;
    sine = amp * sin(2.0*pi*freq*time + phase);
    plot(time, sine, 'LineWidth', 3, 'Color', 'r');
    axis([-2.1 2.1 -5.1 5.1]); } calculate sine
                                and plot it in
                                guessAxes
    grid on
end
```

Waves 14-10

## Setting up for the user's next action

So far we've been concerned with responding to the user's current action. We also need to consider, *what will the user do next?*



This may involve enabling and disabling GUI components:

```
set(component, 'Enable', 'on')  
set(component, 'Enable', 'off') } Enabling or  
                                disabling a  
                                component
```

Waves 14-11

## submitButton\_Callback

```
function submitButton_Callback (hObject, eventdata, handles)  
% display sinusoid guessed by user  
displaySine(handles, 0)  
  
% check whether user's guess is correct  
if ((handles.amplitude == str2double(get(handles.ampBox, 'String'))) & ...  
    (handles.frequency == str2double(get(handles.freqBox, 'String'))) & ...  
    (handles.phase == str2double(get(handles.phaseBox, 'String'))))  
    set(handles.statusLabel, 'String', 'You got it! Try a new one!');  
    set(handles.submitButton, 'Enable', 'off'); % disable submit button  
else  
    set(handles.statusLabel, 'String', 'close -- try again!');  
end  
  
% Update handles structure  
guidata(hObject, handles);
```

Waves 14-12

## testButton\_Callback

```
function testButton_Callback(hObject, eventdata, handles)
% create parameters for new test sinusoid
handles.amplitude = selectRandom(1:5);
handles.frequency = selectRandom([0.25 0.5 0.75 1.0 1.5 2.0 2.5 3.0]);
handles.phase = selectRandom([-2.36 -1.57 -0.78 0.0 0.78 1.57 2.36 3.14]);
displaySine(handles,1); % display new test sinusoid
disp(['correct amplitude: ' num2str(handles.amplitude)]; % cheat sheet
disp(['correct frequency: ' num2str(handles.frequency)];
disp(['correct phase: ' num2str(handles.phase)];

set(handles.ampBox, 'String', ''); % clear text boxes
set(handles.freqBox, 'String', '');
set(handles.phaseBox, 'String', '');
set(handles.submitButton, 'Enable', 'on'); % enable submit button

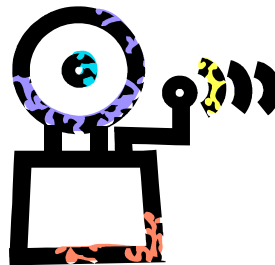
% Update handles structure
guidata(hObject, handles);
```

Waves 14-13

## Bells and whistles

Other things you may want to explore  
in a GUI for your final project:

- adding animation
- other GUI components
  - Slider
  - Listbox
  - Radio Buttons
- dialog boxes
- pull-down menus
- mouse control



Waves 14-14