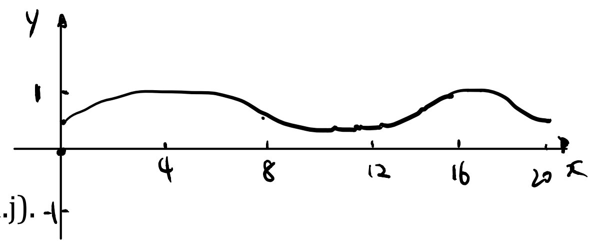
CS 5390/7390 Fall 2020 HW 3 – Sampling, Interpolation, and Quantization

## PART A: SineWaves (40 points)

Submit all files in parts A and Parts B in one zip folder in Canvas.

1) Consider a sine function used to define the intensity of each pixel across a 512x512 floating point image, varying only with column. How would you define the intensity for a single element f(i,j) in a such a way that the sine way completes **64 complete cycles** across the full length of the image?

a) Sketch this as a one dimensional slice of the image with the horizontal axis labeled as pixel value and the vertical axis the amplitude of the intensity ranging from [0,1]




b) Write the code to set a for a single pixel f(i.j).

c) Considering the same 512 x 512 image, what is the maximum number of cycles that could be used to assure that the sine wave frequency can be reconstructed from the image?

Sketch what this would look like and explain your reasoning.

From what I understood of the question: the min number of pixels need for a cycle is 2. So the max number of cycles in a 512 x 512 image is 512/2 = 256 cycles.

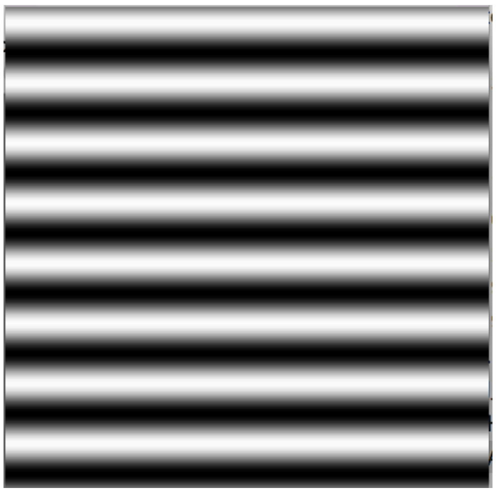
2) Write a program **SineWave.cpp** to and display the following NxN images using OpenCV. Upload your code and screenshots of our display.

CS 5390/7390 Fall 2020 HW 3 – Sampling, Interpolation, and Quantization

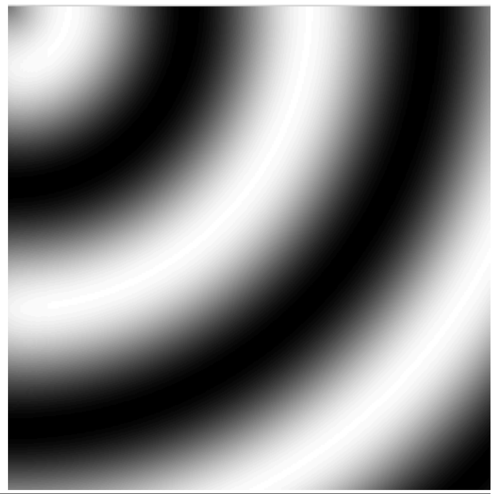
5390 section – do parts A and B. 7390 and Honors – A,B,C and D

Make your image any size such $N = 2^k$, where k is an integer greater than 5.

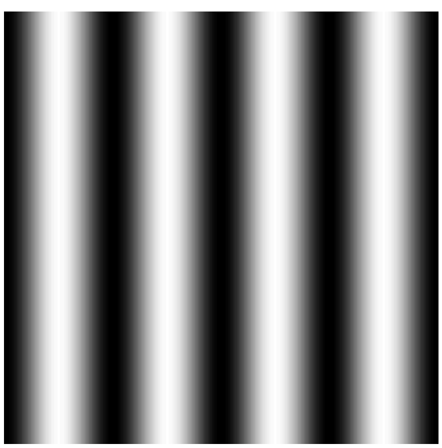
**A** (Value of row 0 is 0.5)



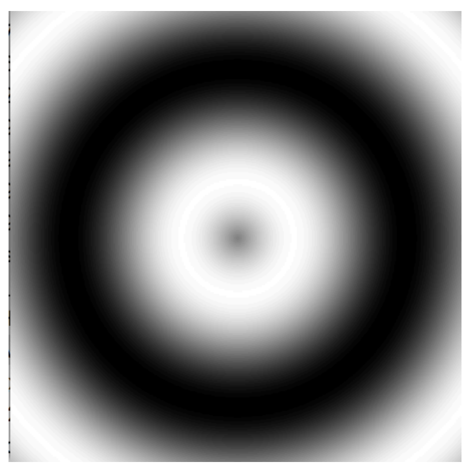
**B** (Value of (0,0) is 0.5)



**C** (Value of column 0 is 0)



**D** (Value of center point is 0.5)



## PART B: Programming problems using OpenCV C++ (60 points)

Submit all files in parts A and Parts B in one zip folder in Canvas.

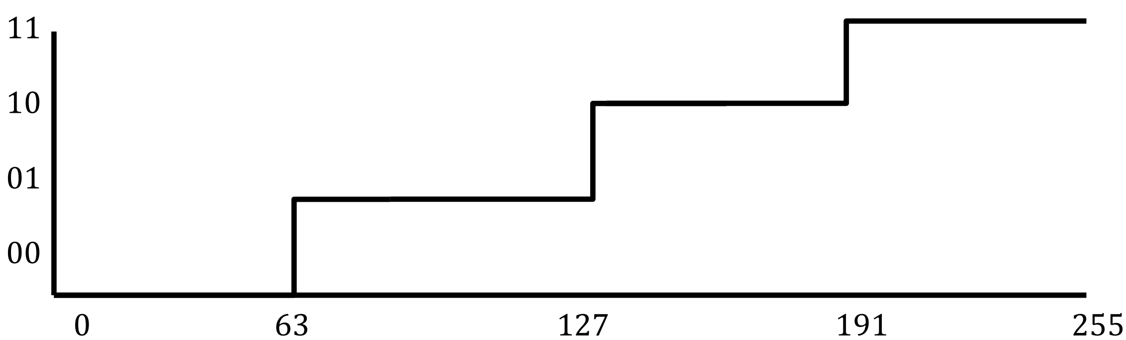
**1A) (5390 and 7390 sections)** Create a program, **ColorRamp.cpp**, to generate a two tone color ramp image of arbitrary dimensions, MxN, so that the amount of one color plan is linearly interpolated across the entire width. Use linear interpolation

Let the color in the first column be $C_0 = (0,0,0)$ and let the color in the last column be $C_{N-1} = (255,0,0)$. That is, compute $C_i = \alpha\, C_0 + (1 - \alpha)\, C_{N-1}$

**1B) 7390 Requirement ONLY.** Create a **TwoToneColorRamp.cpp** so that the color varies smoothly horizontally and vertically in two colors. Use bilinear interpolation to vary across the plane.

2) Create a program, **AddStructuredNoise.cpp**, to generate and display a new 256x256 gray scale image that adds a sinusoidal image to flowersgray.tiff. HINT: Use techniques you used Part A, problem 2A

3) Create a program, **QuantizeThis.cpp**, choose a MxN 8-bit color image, where $256 \leq M \leq 1024$ and $256 \leq N \leq 1024$, and create a second image in which each color plane is separately quantized to a **2, 3 or 4 bits** representation. Save your image to a new png file using imwrite. Upload the source code, your input image and the png you created. Note – consider this to be the implementation of a tonal mapping function that maps an input (horizontal axis) to discrete outputs (vertical axis) as a step function.




4A) Develop an interactive tool **SineWaveExplorer.cpp** for constructing a single sinusoidal wave grating. Have one slider control the horizontal frequency, one control the phase shift and one control the amplitude.

4B) (7390 and Honors section only requirement) extend 4A to have a fourth slider that controls vertical frequency to result in various orientations of the grating.