1. Aliasing:

   (a) Create a sine wave with frequency 60Hz as follows:

      i. Set $f = 60$. Set $t_{\min} = -0.05, \quad t_{\max} = 0.05$. Create a linearly spaced interval $t = [t_{\min}, t_{\max}]$ with 400 node points.
      ii. Create $x = \cos(2\pi f t)$
      iii. Plot t against x.

   (b) Subsample this function with sampling frequency 800 Hz as follows:

      i. Set $T = 1/800$, $n_{\min} = \lceil t_{\min}/T \rceil$, $n_{\max} = \lfloor t_{\max}/T \rfloor$. Create a linearly spaced interval $n = [n_{\min}, n_{max}]$ with interval 1.
      ii. Create $x_1 = \cos(2\pi f n T)$
      iii. Plot t against x_1. This is an oversampled version of x.

   (c) Create another oversampled version of x with sampling frequency 400 Hz.

   (d) Create another critically sampled version of x with sampling frequency 120 Hz.

   (e) Create another undersampled version of x with sampling frequency 70 Hz. How well do these functions represent the original function x.

2. Plasma Effect: Plasma effects are often used to create wobbly animations in demo products like screen savers. Using mathematical functions like sine, tan, etc, is one way of creating such effects.

   (a) Create a grid with $x, y \in [-\pi, \pi]$. Plot the function $f(x, y) = \sin(5 * x)$. This an over simplified plasma.

   (b) Now Plot the function $f(x, y) = \left(\sin^2(3x) + \sin^2(3y)\right)/2$. You can try changing the powers of sine and the frequency to see how it evolves.

   (c) Plot $f(x, y) = \left|\cos\left(20\left(\cos^2(20) + \sin^2(20)\right)\right)\right|$

   (d) Now create a colour image with the following parameters

$$
\begin{aligned}
r &= \cos(x) + \sin(y) \\
\text{red} &= \cos(yr) \\
\text{green} &= \cos(xyr) \\
\text{blue} &= \sin(xr).
\end{aligned}
$$

   Take the absolute value of the result to avoid working with negative numbers.

   (e) Try to generate 2 colour plasma effects on your own.

   (f) Now animate a plasma sequence using the following sequence

$$
\begin{aligned}
v1 &= \sin(10x + 5t) \\
v2 &= \sin(10(x \sin(t/2) + y \cos(t/3)) + 5t) \\
cx &= x + .5 \sin(t/5) \\
cy &= y + .5 \cos(t/3) \\
v3 &= \sin\left(\sqrt{100(cx^2 + cy^2) + 1} + t\right) \\
v &= v1 + v2 + v3
\end{aligned}
$$

   where t is the animation time. Show the image as a colour image using the following colour component

$$
\begin{aligned}
r &= 1 \\
g &= \cos(\pi v) \\
b &= \sin(\pi v)
\end{aligned}
$$

(g) Create a meshgrid with $x, y \in [-\pi, \pi]$. with 100 elements in the x and y direction. Visualize the image

$$z = \sin\left(i\sqrt{x^2 + y^2}\right), \quad i = 1, 2, \ldots$$

i. At what value of i does the image no more display concentric circles. In this case, i masquerades as the frequency of the sin curve.

(h) Increase the number of samples in the meshgrid to $n * 50,\ n = 3, 4, \ldots, 16$. For each n, visualize the image z and determine at what values of i the image stops displaying concentric circles.

(i) Plot n against i.

(j) The Nyquist Sampling theory, one of the major theories that determine how we sample continuous functions states that "an image can be reconstructed from its samples without error if the number of samples is at least twice the highest frequency." Can you relate the this with the phenomena in (f), (g), (h)?

3. Iterated Function Systems (Deterministic Algorithms): Affine maps of the form

$$f(x) = Lx + b$$

where L is a linear map given by a matrix and $b \in \mathbb{R}^2$where the linear map is a contraction map can be used to produce some very stunning curves. This project uses affine maps to create some graphics.

(a) Use $L_j = \begin{pmatrix} 1/3 & 0 \\ 0 & 1/3 \end{pmatrix}$, $(1 \le j \le 5)$ and $b_1 = \begin{pmatrix} 1/3 \\ 0 \end{pmatrix}$, $b_2 = \begin{pmatrix} 0 \\ 1/3 \end{pmatrix}$, $b_3 = \begin{pmatrix} 1/3 \\ 1/3 \end{pmatrix}$, $b_4 = \begin{pmatrix} 2/3 \\ 1/3 \end{pmatrix}$, $b_5 = \begin{pmatrix} 1/3 \\ 2/3 \end{pmatrix}$ for each iteration i, create the transformation A_i by applying these transformation to A_{i-1}. Begin with A_0 as the boundary of unit square.

(b) The next IFS consists of the affine mappings f_j $(1 \le j \le 4)$ given by $L_1 = \begin{pmatrix} 1/3 & 0 \\ 0 & 1/3 \end{pmatrix}$, $b_1 = \begin{pmatrix} 1/3 \\ 0 \end{pmatrix}$, $L_2 = \begin{pmatrix} 0 & -1/3 \\ 1/3 & 0 \end{pmatrix}$, $b_2 = \begin{pmatrix} 1/3 \\ 0 \end{pmatrix}$, $L_3 = \begin{pmatrix} 0 & 1/3 \\ -1/3 & 0 \end{pmatrix}$, $b_3 = \begin{pmatrix} 2/3 \\ 1/3 \end{pmatrix}$, $L_4 = \begin{pmatrix} 1/3 & 0 \\ 0 & 2/3 \end{pmatrix}$, $b_4 = \begin{pmatrix} 1/3 \\ 1/3 \end{pmatrix}$. Take A_0 as the half of the unit square below the diagonal off the origin.

(c) Perform (b) using the boundary of the unit square.

(d) Now consider the affine mappings f_j $(1 \le j \le 3)$ given by $L_1 = \begin{pmatrix} 1/3 & 0 \\ 0 & 1/3 \end{pmatrix}$, $b_1 = \begin{pmatrix} 1/3 \\ 2/3 \end{pmatrix}$, $L_2 = \begin{pmatrix} 0 & -1/3 \\ 1 & 0 \end{pmatrix}$, $b_2 = \begin{pmatrix} 1/3 \\ 0 \end{pmatrix}$, $L_3 = \begin{pmatrix} 0 & 1/3 \\ -1 & 0 \end{pmatrix}$, $b_3 = \begin{pmatrix} 2/3 \\ 1 \end{pmatrix}$. Take A_0 as the half of the unit square below the diagonal.

(e) Repeat (d) using the unit square. In all questions, show at least A_0, A_1, A_3, A_6, A_{10}.

(f) Try the affine map f_j $(1 \le j \le 3)$ given by $L_1 = \begin{pmatrix} 0.387 & 0.430 \\ 0.430 & -0.387 \end{pmatrix}$, $b_1 = \begin{pmatrix} 0.176 \\ 0.522 \end{pmatrix}$, $L_2 = \begin{pmatrix} 0.441 & -0.091 \\ -0.009 & -0.322 \end{pmatrix}$, $b_2 = \begin{pmatrix} 0.342 \\ 0.506 \end{pmatrix}$, $L_3 = \begin{pmatrix} -0.468 & 0.020 \\ -0.113 & 0.015 \end{pmatrix}$, $b_3 = \begin{pmatrix} 0.320 \\ 0.400 \end{pmatrix}$. Start with A_0 as the unit square.

4. Iterated Function Systems: Probabilistic Algorithms. These IFS use affine transformations of the form

$$f(x, y) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

where each transformation is selected based on a probability p.

(a) The Transformation for the Barnsley fern are given in the following table

w	a	b	c	d	e	f	p
f_1	0	0	0	0.16	0	0	0.01
f_2	0.85	0.04	-0.04	0.85	0	1.6	0.85
f_3	0.2	-0.26	0.23	0.22	0	1.6	0.07
f_4	-0.15	0.28	0.26	0.24	0	0.44	0.07

i. Plot the fern from the given transformation above using 1000, 30000, 50000 points starting from the point (0.5,0.5).

(b) Change the Transformations to the following

w	a	b	c	d	e	f	p
f_1	0	0	0	0.16	0	0	0.1
f_2	0.2	-0.26	0.23	0.22	0	1.6	0.08
f_3	-0.15	0.28	0.26	0.24	0	0.44	0.08
f_4	0.75	0.04	-0.04	0.85	0	1.6	0.74

(c) Next, try the following transformation

w	a	b	c	d	e	f	p
f_1	0.5	0	0	0.5	1	1	0.33
f_2	0.5	0	0	0.5	1	50	0.33
f_3	0.5	0	0	0.5	50	50	0.34

begin at the point $(0,0)$. This is the Sierpinski Triangle. Change the probabilities to

i. $p_1 = 0.2$, $p_2 = 0.46$, $p_3 = 0.34$

ii. $p_1 = 0.1$, $p_2 = 0.56$, $p_3 = 0.34$

iii. How many iterations are needed by the different probabilities to produce a good Sierpinski triangle.

(d)

w	a	b	c	d	e	f	p
f_1	0	0	0	0.5	0	0	0.05
f_2	0.42	-0.42	0.42	0.42	0	0.2	0.4
f_3	0.42	0.42	-0.42	0.42	0	0.2	0.4
f_4	0.1	0	0	0.1	0	0.2	0.15

Begin at $(0,0)$.

(e) Change the transformation to this new transformation:

w	a	b	c	d	e	f	p
f_1	0.1950	-0.4880	0.3440	0.4430	0.4431	0.2452	0.2
f_2	0.4620	0.4140	-0.2520	0.3610	0.2511	0.5692	0.2
f_3	-0.6370	0	0	0.5010	0.8562	0.2512	0.2
f_4	-0.0350	0.0700	-0.4690	0.0220	0.4884	0.5069	0.2
f_5	-0.0580	-0.0700	0.4530	-0.1110	0.5976	0.0969	0.2

5. Playing with images III Nonlinear Processing: This project experiments with noise removal techniques on images.

(a) a.png is a test image used for image processing. b.png is the image a.png corrupted with noise. One way of approximating the real image a.png is to blur the image. To do this

i. Take the 3×3 matrix of ones scaled such that the sum of the elements is 1, ie $\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$.

Convolve the image b.png with the matrix to reduce the noise in the image.

ii. Perform the same operation using 5×5, 7×7, 9×9, 11×11 and 13×13 matrices. Show the resulting images.

(b) The Gaussian Filter with standard deviation σ is given by the equation

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

i. In a 501×501 image, visualize Gaussian filters with σ =2, 10,15,20,30,5070,100.

ii. Create a 5×5 Gaussian filter $f(x,y)$ with $\sigma = 2$. Scale this matrix to have sum 1. Use this image to smooth the image b.png.

(c) The image c.tif is an image corrupted by another kind of noise. What is the difference between the two kinds of noise?

i. Filter this image with the filter in (a)-i and (b)-ii.

ii. Replace each pixel in the image with the median of its 9 neighbours. What is the result.

(d) For the images d.tif and e.tif, perform the following operations:

i. For each pixel in the images, replace the value of the pixel with the maximum value of the pixel with the 8-neighbours around it.

ii. For each pixel in the images, replace the value of the pixel with the minimum value of the pixel with the 8-neighbours around it.

iii. Explain why these operations work for one kind of noise and not the other.

6. Playing with Images 2: The dataset for this project is Lena.tif

(a) One way of removing noise from an image is to solve the heat equation

$$\frac{\partial u}{\partial t} = -\nabla^2 u.$$

This solution can be approximated by the Gaussian convolution $g \star u$ where g is the Gaussian function or a smoothing filter and u is the original image. One can denoise an image in discrete terms by using the simple averaging filter

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Convolution here is averaging each element with its neighbouring elements and summing. Perform this operation on the Lena image and display the result. (Note: It is advisable to change the image from uint8 to float and scale from [0,255] to [0,1] before filtering).

(b) The heat equation is a diffusion process. Perform repeated convolution on the image with the filter in (a) until the image intensity becomes homogeneous. Show the evolution of the image.

(c) Another smoothing filter much closer to the Gaussian is the binomial filter formed by taking the outer product of the matrix $[1, 2, 1]$ scaled to sum to 1. This is called the binomial filter because the matrix elements are taken from the Pascal's triangle.

i. Use this filter to filter the Lena image 3 times.

ii. Use another filter formed from the level 6 Pascal's triangle $[1, 6, 15, 20, 15, 6, 1]$. Is there any conclusion to be drawn from the two operations. The operations performed above are called low pass filtering.

(d) It is possible to find the derivative of the function u as the derivative with respect to x given as $\frac{\partial u}{\partial x}$ and $\frac{\partial u}{\partial y}$ for the derivative with respect to y. These filters are approximated by

$$\frac{\partial u}{\partial x} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } \frac{\partial u}{\partial y} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

i. Use convolution to find $\frac{\partial u}{\partial x}$ and $\frac{\partial u}{\partial y}$ for u =Lena.tif

ii. Find the gradient magnitude by performing tha calculation $\sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2}$

(e) Smooth Lena.tif using the level 4 Pascal's triangle $[1, 4, 6, 4, 1]$ and perform the calculations in (d) on the result. What is the result?

7. Playing with Images: (Note: An 8-bit image has maximum value 255 and minimum 0.)

(a) For the image a.tif, perform some arithmetic operations:

i. add 50 to the value of each element
ii. subtract 50 from the value of each element.
iii. divide the image by 2
iv. multiply the image by 2

(b) Consider the image b.tif. This is called a low contrast image in image processing.

i. Using python, find the minimum and maximum values in the image.
ii. Display the histogram of the image.
iii. Determine a linear equation to extend the image's contrast such that the minimum value maps to 0 and the maximum value maps to 255. This is called contrast stretching.
iv. Display the histogram of the new image.

(c) Consider the image c.tif. This is an MRI image of a fractured human spine. The image can be corrected using the power law or gamma transformation

$$y = cx^{\gamma}$$

where c and γ are positive constants. Visualize the function for $c = 1$ and $\gamma = 0.04, 0.10, 0.20, 0.40, 0.67, 1, 1.5, 2.5, 5.0, 10.0, 25.0$. What will be the effect of applying the gamma transformation with the specified γ on an image.

i. Apply the gamma transformation to the image b.tif with $c = 1$ and $\gamma = 0.6, 0.4, 0.3$.
ii. Apply the gamma transformation to the image c.tif with $c = 1$ and $\gamma = 3.0, 4.0, 5.0$.

8. The Julia set is a fractal named after the french mathematician Gaston Julia whi investigated their properties. You are to investigate the Julia Set and visualize it. Take as your complex space the region $[-2, 2] \times [-2, 2]\, i$

(a) The first fractal you will investigate is the fractal

$$f_c(z) = z^2 + c \quad \text{for}$$

i. $c = 1 - \varphi$ where φ is the golden ratio
ii. $c = (\varphi - 2) + (\varphi - 1)\, i = -0.4 + 0.6i$
iii. $c = 0.8 + 0.156i$
iv. $c = 0.285 + 0.1i$

(b) Investigate the Julia set on the following functions:

i. $f(z) = z^4 + 0.484$
ii. $f(z) = z * e^z + 0.04$
iii. $f(z) = \sqrt{\sinh(z^2)}$+0.065+0.122i
iv. $f(z) = \left(\frac{z^2 + z}{\ln(z)}\right) + 0.268 + 0.060i$

9. Consider a $(2n+1)\times(2n+1)$ square that is centred at the origin and has been covered with 1×1 tiles. A robot is placed on the centre tile and proceeds to hop from tile to tile according to a set of very simple rules. In particular, if (x_c, y_c) is the current robot location, then

   - With probability $p_N = .25$, it moves to its "north" neighbour. Its new location is then $(x_c, y_c + 1)$.
   - With probability $p_E = .25$, it moves to its "east" neighbour. Its new location is then $(x_c + 1, y_c)$.
   - With probability $p_S = .25$, it moves to its "south" neighbour. Its new location is then $(x_c, y_c - 1)$.
   - With probability $p_W = .25$, it moves to its "west" neighbour. Its new location is then $(x_c - 1, y_c)$. The hopping continues until the robot reaches an edge tile. A simulation of this variety is called a random walk. Construct a random walk for $n = 10$.

   (a) For a given n, what is the average number of hops required for the robot to reach the boundary?

   (b) Given that x and y define the robot's trajectory, print the number of times that the robot moved in each of the north, south, east and west direction.

   (c) Modify your random walk to allow movement in four additional directions, NE, SE, SW, NW, all eight directions are equally likely to be taken.

   (d) What is the average number of times that the robot revisits the origin? Produce a bar plot that displays these expected values for $n = 5, 10, 15, 20, 25$.

   (e) We conjecture that the robot is more likely to exit near the middle of an edge than near a corner. Produce a bar plot that sheds light on this conjecture.

   (f) We have so far assumed that each of the directional probabilities is the same. Generalize the random walk so t he user inputs the probabilities. Explore the correlation between the directional probabilities and the exit edge.

10. Let R be a rectangular plate with vertices at $(0,0), (6,0), (6,4), (0,4)$ and assume that its temperature distribution is given by

$$T_{plate}(x,y) = 100e^{-.4\left((x-1)^2+.7(y-3)^2\right)} + 80e^{-.2\left(2(x-5)^2+1.5(y-1)^2\right)},$$

   the idea is to track how the plate cools when all of a sudden, the boundary temperature is reduced to zero and held constant.

   (a) Assume a simple discrete model that can be used to simulate how the temperature changes on an $m \times n$ grid of points that has been superimposed on the region. At the start, the temperature at each interior point is prescribed by the function $T_{plate}(x,y)$. The temperature on the boundary mesh is set to zero.

   (b) Simulate the cooling process at times $t = \triangle t, 2\triangle t, 3\triangle t, \ldots$ according the following rule:
   The temperature at an interior mesh point at time $(k+1)\triangle t$ is the average of the temperature at its four neighbour mesh points at the time $k\triangle t$.

   (c) What is the difference between running the average in the source image and keeping the average in a new destination image?

   (d) How does the simulation change is the eight neighbours are used instead of four.

   (e) Assume that simulation terminates as soon as the maximum temperature is less than one-tenth the maximum temperature at the start of the simulation. How does the number of iterations depend on the mesh size parameter m and n?

   (f) What happens to the simulation when weighted averaging is used (example triangle averaging)?

11. (a) Assume you have a site that allows passwords of length 3 of uppercase letters. How many different passwords can be created? show samples.

   (b) Show how the number of password possibilities increase as the length of the passwords are increased between 3 and 15.

(c) Add lowercase letters to the problems above. How do the possibilities increase.

(d) Add numbers and the symbols +, -, _, @, .

(e) Simulate brute force password hacking for (a). How long does it take?

(f) Try the same for alphanumeric characters and the special symbols in (d). How long does it take to hack a password system?

(g) Increase the problem complexity using passwords of length 4, 5, and 6(if possible). Compare the time it takes.

(h) Is it possible to do the same for passwords of length 6-8 using your laptops?

12. Cellular Automata: Cellular automata is used in different branches of science to model complexity for example in organisms. An example of 2 dimensional cellular automata is a 2d grid of cells with each cell having one of two states on or off, black or white with each cell having limited local knowledge. In a series of cycles, each cell decides its next state based on the current states of its surrounding cells (from Matt Pearson, Generative art: a practical guide using processing, Manning publications, 2011).

   (a) Implement the Game of life CA using the following rules

      i. If a live cell has two or three live neighbours, it continues to live. Otherwise it dies either of lonliness or overcrowding.

      ii. If a dead cell has exactly three live neighbours, a miracle occurs; it comes back to life.

   (b) Modify the CA in (a) using a neighbouhood of radius 2.

   (c) The stepping stone cellular automata has the following rule: "Choose a number between 0 and 1; this will be the update probability for all cells. For each cell in the array, generate a random number between 0 and 1 at every time step. If the random number generated for the given cell is higher than the update probability, the color of the cell changes to that of one of its neighbours selected uniformly at random. (Neighbour is defined as the four orthogonally adjacent cells: north, east, south, west)."

      i. What is the evolutionary behaviour of this CA?

      ii. What happens when we begin with a grid of random colors?

      iii. What if we extend the definition of neighbour to include the north-east, north-west, south-east, south-west?

      iv. Now take a picture as the initial grid and discuss what happens in time.

   (d) Another CA type is Vichniac Vote. In this CA evolution, each cell is susceptible to peer pressure. It looks to its neighbours to observe the current state. If it's state is in the majority, it remains unchanged. It it is in the minority, it changes. The cell uses its own state as a tie breaker.

   (e) Brian's brain is a three-state cellular automata. The states mimic the behaviour of brain neurons. The states are of, fire, rest. The rules are:

      i. If the state is firing, the next state is resting.

      ii. If the state is resting, the next state is off.

      iii. If the state is off and exactly two states are firing, the state becomes firing.

   (f) Waves (averaging) is an example of CA with continuous behaviour. The state can vary across a range of values. The rules are as follows:

      i. If the average of the neighbouring states is 255, the state becomes 0.

      ii. If the average of the neighbouring states is zero, the state becomes 255.

      iii. Otherwise,

$$\text{new state} = \text{current state} + \text{neighbourhood average} - \text{previous state value}$$

      iv. If the new state goes over 255, clip to 255. If the new state goes under 0, clip to 0.

13. The following statistical values have important applications in image restoration.

    (a) Mean: $f(x,y) = \dfrac{1}{mn}\sum_{(r,c)\in W} g(r,c)$

    (b) Geometric Mean: $f(x,y) = \left[\prod_{(r,c)\in W} g(r,c)\right]^{1/mn}$

    (c) Harmonic Mean: $f(x,y) = \dfrac{mn}{\sum_{(r,c)\in W}\left(\dfrac{1}{g(r,c)}\right)}$

    (d) Median: $f(x,y) = \text{median}\left\{g(r,c)|\,(r,c)\in W\right\}$

    (e) Min filter: $f(x,y) = \min\left\{g(r,c)|\,(r,c)\in W\right\}$

    (f) Max filter: $f(x,y) = \max\left\{g(r,c)|\,(r,c)\in W\right\}$

    (g) Standard deviation: $f(x,y) = \sqrt{\dfrac{1}{mn-1}\sum_{(r,c)\in W}\left[g(r,c) - \dfrac{1}{mn-1}\sum_{(r,c)\in W} g\left(r,c\right)\right]^2}$

    Implement these filters in image restoration and explain how they work.

14. Heat dissipation:
    A rectangular plate has been warmed by a pair of heat sources, one at (1,3) and the other at (5,1). The temperature drops exponentially away from these points and is modeled by

$$T_{\text{plate}}(x,y) = 100e^{-.4((x-1)^2+.7(y-3)^2)} + 80e^{-.2(2(x-5)^2+1.5(y-1)^2)}$$

    (a) Develop an intuition about the variation of T_{plate} across the plate by displaying a contour plate.

    (b) Let R be a rectangular plate with vertices at (0,0), (6,0), (6,4), (0,4) and assume the temperature is given in the equation above. The aim is to track how the plate cools when all of a sudden the boundary temperature is reduced to 0 and held constant.

        i. Model the cooling process by using pixel averaging. That is for every pixel not on the boundary, We average the pixel with its 8-neighbours

*	*	*
*	*	*
*	*	*

        ii. Create an approximation of the Gaussian by creating an outer product of a level two Pascal's triangle entry with itself

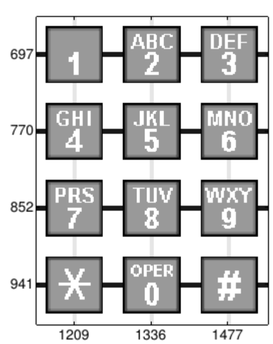
        iii. Create a new cooling model by doing a sum of products with the Gaussian above and the pixel with its 8-neighbours.

        iv. Animate the cooling process in both scenarios.

15. How rapidly the function $y(t) = \sin(2\pi\omega t)$ varies with time depends upon the frequency ω. Elementary sinusoids such as these can be scaled and combined to produce more complicated signals

$$y = a_1\sin(2\pi\omega_1 t) + a_2\sin(2\pi\omega_2 t).$$

    This principle is at work when you use a touch-tone telephone. Each button-push generates a signal that is the sum of two elementary sinusoids. See figure below.



Corresponding the rows 1 through 4 of the touch tone pad are frequencies 697, 770, 852, 941, while frequencies 1209, 1336, 1477 are associated with columns 1 through 3. When a button is depressed, a signal is formed by averaging the two sinusoids whose frequencies are identified by the button's row/column location as

$$y = \frac{1}{2}\sin(2\pi\omega_{\text{row}}t) + \frac{1}{2}\sin(2\pi\omega_{\text{col}}t)$$

is generated. Each button therefore has its own unique fingerprint. These waveforms might be visually similar but are mathematically very different.

(a) Play and plot the sound generated by each button press for a quarter of a second. Use the sampling rate fs = 32768.

(b) Assume you receive a vector that encodes the sound for a particular touch tone. The challenge is deciphering what button was pressed. Since each key is very different from the other, there is little correlation between them. One way to measure the correlation between them is to compute the cosine of the angle between them

$$\cos_{xy} = \frac{\left|\sum_{i=0}^{n-1} x_i y_i\right|}{\sqrt{\sum_{i=0}^{n-1} x_i^2}\sqrt{\sum_{i=0}^{n-1} y_i^2}}$$

which is the consequence of the dot product $(\cdot)$ between two vectors. If $\cos_{xy}$ is close to 0 then the the frequencies do not correlate. $\cos_{xy}$ close to 1 means the frequencies match. Develop a function to identify the buttons pressed from a given sound.

(c) Using $\cos_{xy}$ above, find the pairwise correlation between the frequencies given above.

(d) What happens to the deciphering process if the received signal gets corrupted by noise. Write a function that corrupts the signal with some noise explain what happens.

(e) The frequencies in touch tone dialing were chosen to facilitate the deciphering process. Change the frequencies to 200, 400, 600, 800 for the rows and 900, 1100, 1300 for the columns and try (c) again.

16. Exploring Pascal's triangle: This project seeks to understand the complexity of Pascal's triangle.

(a) In a 500×500 array, generate the first 500 Pascal's triangle entries.

(b) What do these numbers represent.

(c) What is the relationship between the non-one numbers on a row and the first number on a row whose 1st element (0th element is 1) is prime.

(d) Show all odd entries in Pascal's triangle.

(e) What patterns do you get when you take Pascal's triangle mod:

i. 3
ii. 4
iii. 5
iv. 6
v. 7
vi. 8
vii. 9
viii. 10
ix. 11
x. 12

17. In the complex plane, the *Heighway dragon* is the limit of the iterated function system(IFS) defined by the following transformations

$$\begin{aligned} f_1(z) &= \frac{(1+i)z}{2} \\ f_2(z) &= 1 - \frac{(1-i)z}{2} \end{aligned}$$

with the initial set of points $S_0 = \{0, 1\}$.

(a) Using an initial value of $z_0 = 0$ perform a few iterations of the two transformations above and store the results of each iteration in an array *zvals*. Plot the $\Im(zvals)$ versus $\Re(zvals)$ and investigate your results after each iteration. What happens to the line segments?

(b) The twindragon can be constructed by placing two Heighway dragon curves back-to-back. It is the limit of the following IFS

$$\begin{aligned} f_1(z) &= \frac{(1+i)z}{2} \\ f_2(z) &= 1 - \frac{(1+i)z}{2} \end{aligned}$$

with the initial set of points $S_0 = \{0, 1, 1-i\}$

(c) The *terdragon* is the limit of the following IFS

$$\begin{aligned} f_1(z) =& \lambda z \\ f_2(z) =& \frac{i}{\sqrt{3}} z + \lambda \\ f_3(z) =& \lambda z + \lambda^* \end{aligned}$$

where $\lambda = \frac{1}{2} - \frac{i}{2\sqrt{3}}$ and $\lambda^* = \frac{1}{2} + \frac{i}{2\sqrt{3}}$. Repeat the procedure in (i) and explain your results.

18. (a) Read the Wikipedia entry for "The Random Walk Model" and the Simple Random Walk on $\mathbb{Z}$.

(b) Write a function that produces a simple 1-D random walk with a given number of steps. In the simplest case, at each step a decision is made on whether to move up or down with equal probability. What happens when the probability is not equal?

(c) The random walk in (b) can be upscaled by adding the distance travelled.

(d) Generate 10 such models and plot them on the same canvas.

(e) What kinds of real world data can this be used to model.

(f) Plot a smoothed version using a five day moving average.

(g) Stock watchers sometimes use moving averages to determine whether to buy or sell stocks. Create a 5000 step random walk and use it to model the stock market, showing the effects or moving averages in the buy/sell decision.

19. Basic background subtraction

(a) Create a discrete signal. It can be a sine wave and add noise to this signal. One way of removing noise is the application of the central limit theorem. Investigate how this works.

(b) You wll be provided with a collection of images extracted from a video. You will use your experience in (a) to perform background subtraction on the images.