**Problem 1**

(1) Create a computer program that derives the average from N random numbers obtained from a uniformly random distribution of numbers in space [0 to 1]. The program must run for N = 10, 100, 1000, 10000, 100000, 1000000 random numbers. Make the graph of the mean as a function of N (where the N axis is logarithmic). Describe what conclusions you draw from the results. Use your personal ID number as the original seed (as in all subsequent problems).

(2) Create a program that performs a random walk for N = 1000 steps, in the case of systems (a) 1 dimension and (b) 2 dimensions. The program must calculate the square of the displacement, $R^2$. Run the program for 10000 runs and find the $R^2$ average.

(3) Use the previous program to determine $<R^2>$, but now every 100 steps, from 1 to 1000. Find the average for 10000 runs. Make a graph of the results. Find the optimal straight line using the least squares method. Comment on your results.

(4) Create a program that performs a random walk for N = 1000 steps, in 1 dimension, just like in problem 2. Calculate the position (R) located after the N steps, for 100000 runs. Next, prepare the distribution of the values of R. Repeat the same for N = 2000 steps. Make a graph of the two distributions for the two values of N. What conclusions do you draw from the two curves? Finally, in the same graph include the analytical solution of the distributions.

(5) Create a program in which a particle will perform a random walk for t = 1000 steps in a one-dimensional, two-dimensional and three-dimensional system. The program should calculate the <S>, where S is the number of grid sites that the particle visited at least once. You will make 10000 simulations and find 10 points (one every 100 steps, from 0 to 1000), which will be the average of 10000 simulations. Make the graph of <S> as a function of time t. Then plot and compare to the analytical results as found at the link: Sites Visited [Analytical], on the site kelifos.physics.auth.gr

(6) Create a program in which you have a 2D grid with a size of 500x500. In this grid, randomly enter a number of traps, which they will have a concentration c. Then place 1 particle in a random position in the grid and allow the particle to perform a random walk. The walk will stop when the particle happens to fall into a trap as it moves. The time that it took for the particle to be trapped is t. Do 100000 simulations, and monitor the trapping times, and make the distribution of those times. Pay attention to boundary conditions. When the particle reaches the ends of the grid, it should not be allowed to leave the grid, but remain inside, or returning to previous position, either placed circularly on the opposite side of the grid. Run the program for $c = 10^{-2}$ and $c = 10^{-3}$. Calculate the probability of survival Φ (t), that is the percentage of particles that had not fallen into a trap after t steps. Put the 2 Φ distributions in the same graph and compare your results with Rosenstock's theoretical approach, for which:
Φ (t) = (1- c) <S (t)>, where Φ (t) is the probability of survival and <S (t)> the average number of lattice positions that particle visited at least once. In the above theoretical approach, you will use the analytical expressions of <S> which you will find in the file located at the link: Sites Visited [Analytical], on the site kelifos.physics.auth.gr

**Problem 2**

Consider a square grid of 2 dimensions of size 450 x 450 and place a particle in the center.

Then create a circle with a radius of 200, centered in the square grid. Then select a new particle and place it randomly at a point on the periphery of the circle. The new particle performs a random walk until one of the following possibilities occurs: (a) occupies a position "next" to the central particle, on either side of it (b) it leaves far enough away from the original area (exits the grid). In case (a) this particle is attached to the original particle, its path stops, and so now there is an aggregation of two particles. In case (b) the particle should not leave the grid but it should be repositioned at the periphery of the circle at another random point, and continue its random walk. The process is repeated for a third, fourth particle, etc. and continues for as many particles as needed until the growing aggregate from the center of the circle touches its circumference.

Sketch the state of the system at the end of the process.

Determine the fractal dimension of the structure you created using the following method: First, select a position in the grid randomly at a maximum distance of 10 grid sites from the center of the circle. This position now becomes the center of a square with side L = 10. You count the number of sites M that are occupied by particles in this square that has dimensions 10x10. Then increase L by 10 and create a new square with the same center and side L = 20. Now calculate the new M. Continue the process by enlarging the side of the square, so that you finally have 10 such squares with the last one having side L = 100. Make the graph of L as a function of M. If this structure is fractal then the number of occupied positions M follows a relation of the form $M \sim L^{df}$, with $d_f$ being the fractal dimension of this structure. Therefore, by depicting in a log-log diagram the quantity M for the various values of L, we find the $d_f$ from the slope. We must make N runs of the process and obtain their mean. So we repeat this process N = 20 times, we keep the values of M for the N repetitions and we calculate the average of the 10 values of M. Finally, we find the dimension $d_f$ in the above way.

This model was proposed in the published paper

T.A.Witten, L.M.Sander, “Diffusion-Limited Aggregation, a Kinetic Critical Phenomenon”, Physical Review Letters vol 47, no 19, 1981

**Problem 3**

Create a two-dimensional grid of size NxN. N should take different values and will usually be in the range 100 <N <1000. The grid sites will be randomly chosen to be 0 or 1 with probability p being 1 and (1-p) being 0. Then apply the CMLT algorithm to find the full distribution of clusters. Change p from 0.1 to 0.8 initially with Δp = 0.1, but close to the critical point $p_c$ with Δp = 0.01. Find the values of the critical point $p_c$

Calculate the quantity:

$$I_{av} = \sum_{m=1}^{m_{max}} \frac{i_m\, m^2}{pN^2}$$

Now calculate the $I'_{av}$ that is the same as the $I_{av}$, but without the largest cluster. Make the graphs of Iav and I'av vs p.

All results must be averages, so they need 1000 runs.

Calculate Pmax:

$$p_{max} = \frac{m_{max}}{p\,N^2}$$

and make the graph Pmax vs p.

From the relations

$$\begin{aligned} I_{av}(p) &= k|p - p_c|^{-\gamma}, \qquad p < p_c \\ I'_{av}(p) &= k|p - p_c|^{-\gamma'}, \qquad p > p_c \\ p_{max}(p) &= k|p - p_c|^{-\beta}, \qquad p < p_c \end{aligned}$$

find the critical exponents β, γ, γ '.

The recommended grid size is 1000x1000. Recommended p values: 0.1, 0.2, 0.3, 0.4, 0.5, 0.55, 0.56, 0.57, 0.58, 0.59, 0.6, 0.61, 0.62, 0.63, 0.64, 0.64, 0.65, 0.7, 0.8

In the relation given by Iav (average size of clusters), m is the size of the cluster, $i_m$ the number of clusters with size m and the product $pN^2$ is equal to the number of "1" in the grid.

In the equation of $p_{max}$, $m_{max}$ is the size of the largest cluster.

To calculate the critical exponents, we consider only the points that are close to the $p_c$.

The theoretical values for critical exponents, in a network of infinite dimensions, are:

β = 5/36

γ = 43/18

**Problem 4**

(1) Create an Erdos-Renyi network with N = 10000 nodes. It will be a network in which each of the N nodes will have a random number of k connections. Apply the rule that between two nodes there is a probability p = 1/6 that there is a connection. Find the number of k connections of each node. It will be a table with N = 1000 values of integers. Make a plot of the distribution of k, P (k), as a function of k and calculate the mean value of k. The results should be an average of 1000 simulations. Do the same for a network with N = 100000 nodes.

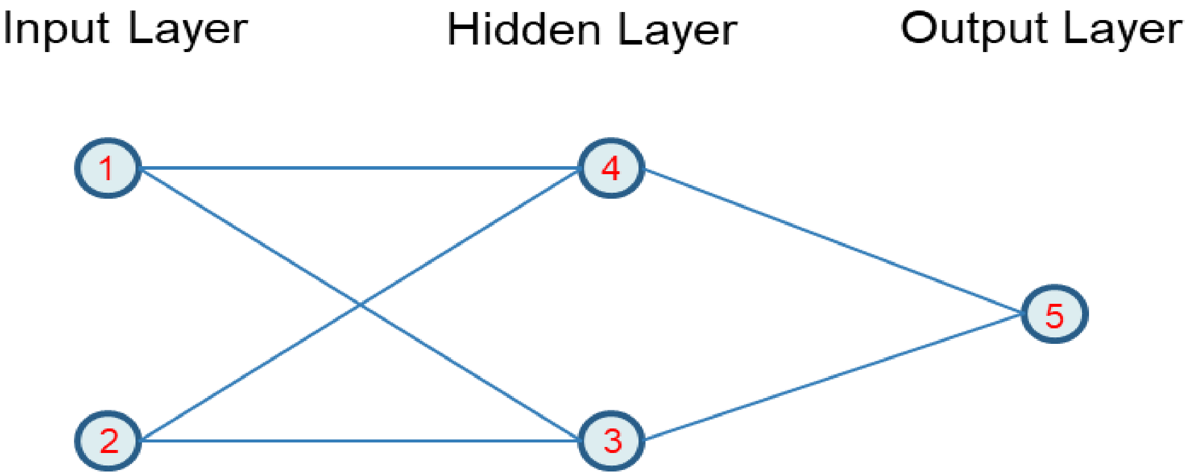
(2) Create a Small World Network with N = 1000 nodes. Initially each node will have exactly k=15 connections, i.e. there will be a total of 15x1000=15000 connections. Start redistributing the connections with probability p = 0.20. Make the plot of the distribution of k, P (k), as a function of k. The results should be an average of 1000 simulations.

(3) Create a power law network (scale-free network) with N = 10000 nodes and γ=3. Assign a random number of k connections for each node with probability $P(k)=k^{-\gamma}$. Make the graph of the distribution of k, P (k), as a function of k. Find the value of γ exactly from the simulation. Do the same for γ = 2.0 and 2.5 The results should be an average of 1000 simulations.

**Problem 5**

Create an artificial neural network (ANN) to solve the X-OR problem, using the back-propagation method. The input level consists of 2 nodes, the hidden level also consists of 2 nodes and the output level consists of one node, as in the diagram. The initial weights of the links should be randomly selected from a uniform distribution w ∈ (-1.1). The maximum acceptable error in the output (1/2|output-target|$^2$) to be considered 0.01 and the learning rate η = 0.2

The maximum number of training cycles (epochs) will be 10000. After the network training is finished, the computer code should write the final weights of all connections in a file.


Input Layer
Hidden Layer
Output Layer
1
4
5
2
3