Introduction to Matlab Programming
Problems

Arrays and Graphics

1 Exercises

Problem 1. Use the documentation to look up the functions: `atan` and `atan2`. Illustrate the similarity and difference between them.

Problem 2. What are the results of the following expressions and why:

$$\frac{1}{0}, \frac{0}{0}, \frac{0}{\infty}, \frac{\infty}{0}, \frac{\infty}{\infty}, \infty + \infty, \infty - \infty, 0 \cdot \infty, 1^{\infty}, 0^{0}, 0^{\infty}, \infty^{0}, \infty^{\infty}.$$

Problem 3. Let $x := 3$ and $y := 7$, using variables x and y find the value of the following expressions:

a) $\dfrac{x^2 - y^2}{x^3 y}$

b) $x^y - y^x$

c) $\sin(\pi(x+y)) - \sqrt{y-x}$

d) $\log_x(y)$

e) $e^x + e^{-x} - 2\cosh(x)$

f) $\sqrt[x]{y}$

Problem 4. Write a program, that converts from kilometers to miles by reading an input (number) from the user.

Problem 5. Explain the results of the following expressions:

$$\texttt{uint8}(5-17), \quad [\text{'M'}, 65, 84, 76, 65, 66], \quad \texttt{int8}(1-2^{10}), \quad 1 + 0.1 \times \texttt{eps} \text{ opposed to } 1 + 100 \times \texttt{eps}.$$

Problem 6. Make the following lists using only the colon operator (`:`), `linspace` and arithmetics:

a) $[0, 2, 4, 6, \dots, 20]$

b) $[11, 9, \dots, -9, -11]$

c) $[0.001, 0.01, \dots, 1000, 10000]$

d) $[1, 2, 4, 8, \dots, 1024]$

e) $[\underbrace{1, -1, 1, -1, \dots, 1, -1}_{20}]$

f) the list obtained by dividing the unit interval $[0, 1]$ to 5 equal parts (boundary points included)

g) the list obtained by dividing the interval $[0, 2\pi)$ to 7 equal parts (0 included, 2π excluded)

Problem 7. Run the following command and explain de result:

$$\texttt{char}(\texttt{cumsum}([99, 16, -10, 2, -1, -9, -33, 39, 6, -12, 8, 3, -62, 53, 12, -2])).$$

Can you do your own version?

Problem 8. Make a list X of 20 random integers: $-10 \leq X_k \leq 30$ with uniform distribution. Now, select the following entries of this list:

a) negative entries

b) entries greater than or equal to 7

c) entries greater than -5 and less than or equal to 12

d) odd entries

e) entries divisible by 3 or 7

f) entries dividing 360

Problem 9. Define the following (anonymus) functions:

a) $\mathrm{sumN}(n) := 1 + 2 + \cdots + n$

b) $\mathrm{sumN2}(n) := 1^2 + 2^2 + \cdots + n^2$

c) $\mathrm{fact}(n) := n!$

d) $\mathrm{binom}(n,k) := \binom{n}{k}$

e) $\mathrm{ln2}(n) := 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \cdots + \frac{(-1)^{n+1}}{n}$

f) $\mathrm{solQuad}(a,b) := [\frac{-a-\sqrt{a^2-4b}}{2}, \frac{-a+\sqrt{a^2-4b}}{2}]$

g) $\mathrm{cumAvr}([x_1,x_2,\ldots,x_n]) := [x_1, \frac{x_1+x_2}{2}, \ldots, \frac{x_1+x_2+\cdots+x_n}{n}]$

h) $\mathrm{first}([x_1,x_2,\ldots,x_n]) := x_1$

i) $\mathrm{rest}([x_1,x_2,\ldots,x_n]) := [x_2,x_3,\ldots,x_n]$

j) $\mathrm{take}([x_1,x_2,\ldots,x_n],m) := [x_1,x_2,\ldots,x_m]$

k) $\mathrm{drop}([x_1,x_2,\ldots,x_n],m) := [x_{m+1},x_{m+2},\ldots,x_n]$

Problem 10. Make the following matrices using only the colon operator (`:`), `diag`, `zeros`, `ones`, `eye`, `repmat`, `reshape`, `cat`, `flipdim`, `padarray` and arithmetics:

a) $\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$

e) $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

b) $\begin{bmatrix} 2 & 5 & 0 & 0 & 0 & 0 \\ 7 & 2 & 5 & 0 & 0 & 0 \\ 0 & 7 & 2 & 5 & 0 & 0 \\ 0 & 0 & 7 & 2 & 5 & 0 \\ 0 & 0 & 0 & 7 & 2 & 5 \\ 0 & 0 & 0 & 0 & 7 & 2 \end{bmatrix}$

d) $\begin{bmatrix} 1 & 4 & 7 & 10 \\ 0 & 0 & 0 & 0 \\ 2 & 5 & 8 & 11 \\ 0 & 0 & 0 & 0 \\ 3 & 6 & 9 & 12 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

f) $\begin{bmatrix} 0 & 0 & 0 & 3 & 3 & 1 & 1 \\ 0 & 0 & 0 & 3 & 3 & 1 & 1 \\ 2 & 2 & 2 & 3 & 3 & 1 & 1 \\ 2 & 2 & 2 & 3 & 3 & 5 & 5 \\ 0 & 0 & 0 & 3 & 3 & 5 & 5 \\ 0 & 0 & 0 & 3 & 3 & 5 & 5 \end{bmatrix}$

Problem 11. Make a 10×10 multiplication table. Could you do it in modulo 11 residue class?

Problem 12. Make a 5×6 matrix X of random integers $0 \le X_{ij} \le 50$, then determine the

a) maximum of each column,

b) minimum of each row,

c) sum of the rows,

d) sum of the even columns,

e) sum of the first and last columns,

f) even entries,

g) number of zeros,

h) largest entry,

i) three smallest entries.

Problem 13. Consider Table 1 containing experiment data on 30 subjects. Make a random test table, and determine the:

a) IDs of all female,

b) ID of the youngest male,

c) IDs of those scored ≥ 0.5,

d) mean and standard deviation of the scores

e) mean age of those scored ≤ 0.2,

f) gender ratio,

g) IDs of females past 35,

h) mean scores of males younger than 37,

i) gender ratio of those scored ≥ 0.7,

j) IDs of females scored ≤ 0.3 or younger than 40.

ID	Gender (1–M, 2–F)	Age (20–50)	Score (0–1)
1	2	32	0.68
2	2	28	0.78
3	1	47	0.98
4	2	45	0.43
⋮	⋮	⋮	⋮
30	1	22	0.73

Table 1: Example of the experiment data

Problem 14. Given a round table with 10 seats. Let 0 and 1 denote the empty and occupied seats, respectively. The seats are randomly taken, that is we have a random list of 0's and 1's. Count the number of (non-empty) neighbours within distance of 2 for all seats. How about if the table wasn't round?

Problem 15. Plot the following functions in one figure, but in three – vertically arranged – seperate axes:

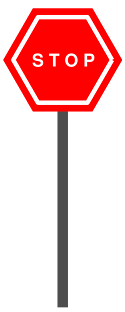
$$\frac{\sin(x)}{x},\ -10 \leq x \leq 10, \qquad \sin\left(\frac{1}{x}\right),\ -10 \leq x \leq 10, \qquad x^2 e^{-x}\cos(5x),\ 0 \leq x \leq 10.$$

Problem 16. Simulate a hundred rolls with two dice, and plot the histogram of the result.

Problem 17. Plot the pie chart of the age data in Table 1 for the age groups: 20–30–40–50, highlight the most numerous group.

Problem 18. Make 5 figures arranged in an X pattern. Make sure that your solution works with any screen resolution.

Problem 19. Draw the following pictures using graphics primitives.



Problem 20. Design your own clock face.

2 Projects

Random Walk in 2D

Consider a particle sitting at the origin $(0, 0)$. Every step the particle makes one of the moves: left, right, up or down randomly with equal probability. Plot the particle's trajectory for a few hundred steps as shown in Figure 1.

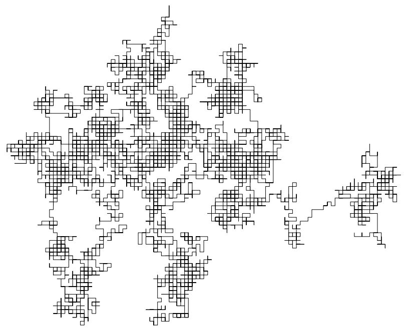


Figure 1: Trajectory of a random walk.

Monte Carlo Simulation

Consider the square having vertices $A(1, -1)$, $B(1, 1)$, $C(-1, 1)$, $D(-1, -1)$ and – inside this square – the unit disk. Choose n random points $P(x, y)$ inside the square, i.e. $-1 \leq x, y \leq 1$. Let N denote the number of points inside the disk, that is $x^2 + y^2 \leq 1$. Now, if n is large enough, then

$$\frac{N}{n} \approx \frac{A_{\text{disk}}}{A_{\text{square}}} = \frac{\pi}{4}.$$

Write a program that simulates the above process. Display the unit disk and the points, also plot the relative frequency: $\frac{N}{n}$ for all $n = 1, 2, \ldots$ as shown in Figure 2.

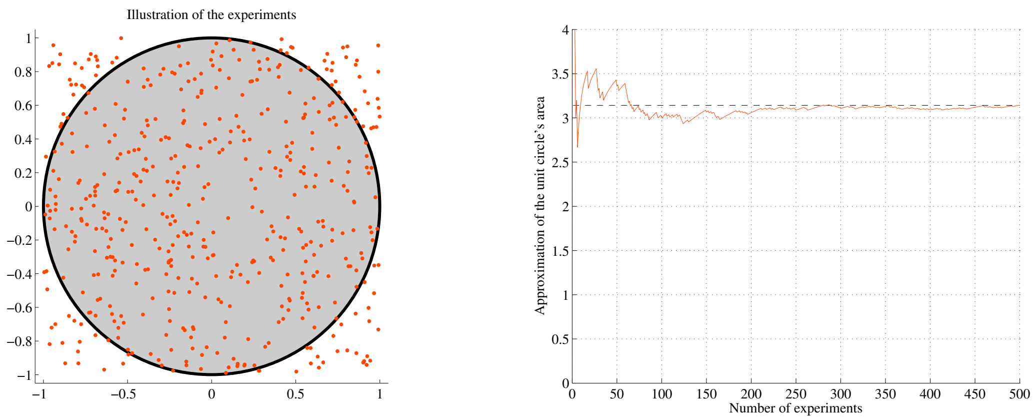


Figure 2: Results of the simulation.

Convay Game of Life

This game is played on an $n \times n$ grid of 1's and 0's, where 1 and 0 represent living and dead cells, respectively. It is clear that every cell has exactly 8 neighbours, except along the border. However, we can view these border cells, as if they had 8 neighbours, considering the cells outside the border dead. The population of cells advances to a new generation by applying the following rules for every cell:

- Any live cell with fewer than two live neighbours dies, as if caused by under-population.
- Any live cell with two or three live neighbours lives on to the next generation.
- Any live cell with more than three live neighbours dies, as if by overcrowding.
- Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

Write a program that for a given table, computes the next generation. For example:

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Look up the `image` object in the documentation, and use it to display the original and new generation in separate figures.

PageRank

This algorithm was first used by Google to order search results. According to Google:

> PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

From this perspective the internet consists of webpages and links between them (see Figure 3). The graph in

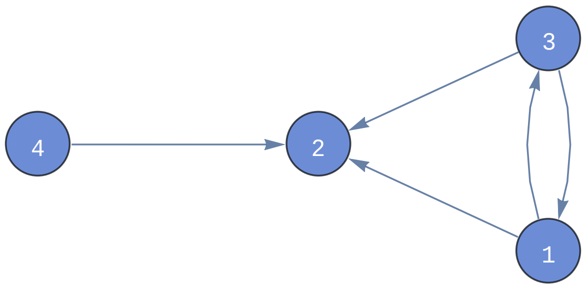


Figure 3: An example graph with $n = 4$ edges.

Figure 3 is represented as two arrays: vertecies (V) and edges (E) in the following way:

$$V = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}, \quad E = \begin{bmatrix} 1 & 1 & 3 & 3 & 4 \\ 2 & 3 & 1 & 2 & 2 \end{bmatrix}.$$

Denote the number of vertices by n, which is $n = 4$ in our example. Now we construct the adjacency matrix A of the graph, that is we have $A_{ij} = 1$ if and only if the graph has the edge $i \to j$, otherwise $A_{ij} = 0$. It could be that, some of the rows have only zero entries, which are quite problematic, therefore we need to substitute every such row with a row full of ones to get a new matrix B. This means, that if we have webpages that don't link to anywere, than we assume instead they link to everywhere. For the example graph, they look like this:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad \longrightarrow \qquad B = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The next step is to make sure that every row sums up to one, thus we need to normalize the rows of B by which we obtain the matrix M with entries

$$M_{ij} := \frac{B_{ij}}{\sum_{k=1}^{n} B_{ik}} = \frac{B_{ij}}{B_{i1} + B_{i2} + \cdots + B_{in}}.$$

Continuing the example, we have:

$$M = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

In order to regularize the problem, we need to perturb the matrix M with the matrix S, that is

$$P := \alpha M + (1 - \alpha) S,$$

where $S_{ij} := 1/n$. Google recommendation is to use $\alpha := 0.8$. The example becomes:

$$P = \frac{4}{5} \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \frac{1}{5} \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} = \frac{1}{20} \begin{bmatrix} 1 & 9 & 9 & 1 \\ 5 & 5 & 5 & 5 \\ 9 & 9 & 1 & 1 \\ 1 & 17 & 1 & 1 \end{bmatrix}$$

Now, in the last step we need to solve the linear equation $P^\top x = x$ for the unkown vector x. This x vector will contain the ranking scores for each webpage. The equation $P^\top x = x$ is equivalent to the homogeneous equation $(P^\top - I)x = 0$, where I is the $n \times n$ identity matrix. Due to the earlier regularization, we can be sure there is only one x vector satisfing this equation[1]. Finally, normalize the result, so that the ranking scores sum up to 1:

$$r := \frac{x}{\sum_{i=1}^{n} x_i}.$$

The ranking scores for the example: $r = [0.223,\ 0.426,\ 0.223,\ 0.128]$, so the ranking is $R = [2,\ 3,\ 1,\ 4]$.

Write a program that for a given V and E computes the ranking. Use randomly generated test graphs to test your program. Make sure that the test graph is a simple (directed) graph, i.e. it does not contain loops or multiple edges.

[1]It is clear that both x and let's say $1.34 \cdot x$ imply the same ranking. So actually, there are infinitely many solutions for $P^\top x = x$, but they only differ in a scalar factor, thus impling the same ranking.