

## ASSIGNMENT 02: A NEW WAY OF THINKING

1. Create the following vectors:

$$\mathbf{u} = [-4, -2, 0, 2, 4] \quad (1)$$

$$\mathbf{v} = \left[0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi\right] \quad (2)$$

2. Calculate  $10!$  using `prod()` and store it into variable `f`.

3. Create the following matrices:

(a)

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3)$$

**Hint:** Look at the documentation for the `zeros()` function.

(b)

$$\mathbf{B} = \begin{bmatrix} 1 & 9 & 2 & 10 \\ 3 & 11 & 4 & 12 \\ 5 & 13 & 6 & 14 \\ 7 & 15 & 8 & 16 \end{bmatrix} \quad (4)$$

**Hint:** MATLAB stores matrices in column major order. An intermediate `reshape()` with a transpose can give you a “nice” matrix to play with.

4. We can “perfectly”<sup>1</sup> represent a square wave with a Fourier series like so:

$$a_n = 2n + 1 \quad (5)$$

$$s = \sum_{n=0}^{\infty} \frac{\sin(a_n t)}{a_n} \quad (6)$$

Plot the resultant approximation for  $n \in [0, 50]$  over 1000 evenly spaced points for  $t \in [-\pi, \pi]$ . Assuming you stored your approximation in `s` and values for  $t$  in `t`, you can plot your solution with `plot(t, s)`.

**Hint:** Use broadcasting to create a matrix with your  $a_n$  and  $t$  values so that you can efficiently `sum()` over your different sine waves!

---

<sup>1</sup>Well, not quite due to the Gibbs phenomenon.