Brandon Swatek - zadanie numeryczne nr 1

In [2]:
```python
import numpy as np


def tridiagonal_matrix_alg(a, b, c, d):

    n = len(d)
    x = np.zeros(n, float)

    c_prim = np.zeros(n-1, float)
    d_prim = np.zeros(n, float)
    c_prim[0] = c[0]/b[0]
    d_prim[0] = d[0]/b[0]
    for i in range(1, n-1):
        c_prim[i] = c[i]/(b[i] - a[i-1]*c_prim[i-1])
    for i in range(1, n):
        d_prim[i] = (d[i] - a[i-1]*d_prim[i-1])/(b[i] - a[i-1]*c_prim[i-1])

    x[n-1] = d_prim[n-1]
    for i in range(n-1, 0, -1):
        x[i-1] = d_prim[i-1] - c_prim[i-1]*x[i]
    return x


if __name__ == '__main__':
    aa = np.array([1, 1, 1, 1, 1, 1])
    bb = np.array([4, 4, 4, 4, 4, 4, 4])
    cc = np.array([1, 1, 1, 1, 1, 1])
    dd = np.array([1, 2, 3, 4, 5, 6, 7])
    print("x elements: ")
    print(tridiagonal_matrix_alg(aa, bb, cc, dd))
```

```
x elements:
[0.1667894  0.33284242 0.50184094 0.65979381 0.8589838  0.90427099
 1.52393225]
```

Metoda: Metoda Thomasa dla macierzy trójprzekątniowych / trójdiagonalnych (Tridiagonal Matrix Algorithm). Jest to uproszczona forma metody eliminacji Gaussa.

Dane wejścia: Wektor bb to właściwa diagonala, aa i cc to pozostałe niezerowe przekątne. Razem, obrazują macierz z polecenia. Wektor dd to wektor znajdujący się po prawej stronie równania.

Opis metody: Metodę można stosować dla tzw. macierzy wstęgowych czyli inaczej trójprzekątniowych. Algorytm ma liniową złożoność O(N). Przy macierzy wejściowej N x N, współczynniki potrzebne do rachunków można zapisać na dwóch wektorach o długości N, tutaj to wektory c_prim oraz d_prim. Wektory współczynników c_prim i d_prim wypełniane są wartościami liczonymi indukcyjnie, eliminując nasz wektor aa (w środku algorytmu odpowiednio 'a'). Później stosuje się tzw. backsubstitution aby otrzymywać kolejno elementy szukanego wektora x.