Laboratorium 5

Podstawy Metod Komputerowych w Obliczeniach Inżynierskich
Rozwiązywanie układów równań, dekompozycje macierzy

Marek Wodziński

AGH UST

18.11.2020

Spis Treści

Zadanie 1

Zadanie 1

Napisz funkcję implementującą algorytm eliminacji Gaussa dla dowolnej macierzy kwadratowej, którą można sprowadzić do postaci górnej trójkątnej. Przetestuj funkcję dla różnych macierzy (w tym dla macierzy o dużym rozmiarze). Pamiętaj o poprawnej implementacji pivotów.

```
def gaussian_elimination(A):
    # A - input matrix
    # outputs A converted to upper triangular matrix
    pass
```

Zadanie 2

Zadanie 2

Korzystając z funkcji zaimplementowanej w Zadaniu 1 - zaimplementuj funkcję do rozwiązywania układów równań (zakładając, że układ równań posiada jedno rozwiązanie). Skorzystaj z funkcji "solve upper triangular" zaimplementowanej na zajęciach.

```
def solve_equations(A, y):
    # A - input matrix
    # y - coefficients
    # returns x, where Ax = y
    pass
```

Zadanie 3

Zadanie 3

Bazując na funkcji "solve upper triangular" zaimplementuj funkcję "solve lower triangular", a następnie wykorzystaj ją do rozwiązania przykładowych układów równań korzystając z dekompozycji LU. Jakie są zalety i dodatkowe możliwości rozwiazywania układów równań korzystajac z dekompozycji LU względem np. eliminacji Gaussa, bezpośredniego odwracania macierzy, metody pseudoinwersji?

```
def solve_lower_triangular(A, y):
    # solves for x where Ax = y
    # and A is a lower triangular matrix
    pass
```

Zadanie 4

Zadanie 4* - dla zdecydowanie "bardziej ambitnych"

Zaimplementuj własną dekompozycje SVD opartą o "power iteration method". Zignoruj przypadek gdy wartości osobliwe są bliskie 0. Porównaj działanie i czas wykonywania z funkcją SVD dostępna w bibliotece numpy/scipy.

```
def our_svd(A)
    # Calculates the SVD using the power method
    # A - input matrix
    # return U, S, Vt where USVt = A
    pass
```