

# Programowanie i metody numeryczne

## zestaw zadań 14

### Wartości własne, wartości szczególne ( Rozkład SVD)

#### *Transformata Fouriera*

Opracowanie *Jędrzej Wardyn*

10 czerwca 2024

## 1 Wartości własne metodą iteracyjną

Napisz program, którym znajdziesz dominującą wartość własną dość prostej macierzy:

$$\begin{pmatrix} -2 & -4 & 2 \\ -2 & 1 & 2 \\ 4 & 2 & 5 \end{pmatrix}$$

Rozwiązanie ręczne

Poradnik do metody [kliknijcie w X tam po prawej żeby zamknąć tam co wyskakuje]

## 2 Rozkład SVD: wartości szczególne (singular values)

Pisanie algorytmu do SVD jest zbyt dużym zadaniem więc na potrzeby zadania powinno się skorzystać z jakiejś biblioteki.

Sprawdźcie czy biblioteka Eigen jest dostępna dla C++ poprzez dodanie

```
#include<Eigen/Dense>
```

I kompilację programu typu "Hello World!". Jeśli nie jest, to proponuję skorzystać z biblioteki w python'a `import numpy as np`, która powinna być dostępna.

Wykorzystując metodę SVD znajdź wartości osobliwe układu równań:

$$A = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}$$

Rozwiązanie krok po kroku

Kalkulator do SVD online dla porównania.

Dokładniej na temat metody SVD

### 3 (Jeśli pakiet FFTW dostępny) Wykorzystanie transformaty Fouriera do rozwiązania przewodnictwa cieplnego w 1D

Celem zadania jest zrozumienie jak wykorzystać bibliotekę taką jak FFTW.

Cała przestrzeń jest dana przez  $0 < x < 1$ , a warunek początkowy to  $u(x, t = 0) = 1$  dla obszaru  $0.4 < x < 0.6$ , a w wyniku przepływu ciepła ten prostokąt powinien zmienić się na coś w rodzaju krzywej rozchodzącej się na boki. Wykorzystaj funkcje z FFTW:

```
fftw_malloc
fftw_plan_dft_1d
fftw_execute
```

Na samym końcu dodaj czyszczenie:

```
// Cleanup
fftw_destroy_plan(forward_plan);
fftw_destroy_plan(backward_plan);
fftw_free(in);
fftw_free(out);
delete[] wavefunction;
```

Dla porównania możesz zobaczyć jak to zrobiono w pythonie: Przewiń filmik do części z kodem w pythonie lub skorzystaj z kodów ze strony. Kod w pythonie [cały pakiet], konkretny plik 1, konkretny plik 2.

### 4 Podstawowa (z definicji) transformata Fouriera

(autor: M.M.)

Napisz program, który wczyta do dwóch vectorów dane z zewnętrznego pliku o dwóch kolumnach (w pliku znajdują się dane typu  $(x, f(x))$ , gdzie  $f(x)$  jest sumą sinusów o różnych częstościach), a następnie obliczy transformatę Fouriera  $f(x)$  (drugiej kolumny). Dla zestawu danych  $(x_0, x_1, \dots, x_{N-1})$  mamy otrzymać nowyzestaw:

$$F_k = \sum_{n=0}^{N-1} f_n e^{-\frac{2\pi}{N} i k n} \quad 0 \leq k < N$$

Kwadrat modułu  $F_k$  nazywamy spektrum:  $\|F_k\|^2 = (\text{Re}F_k)^2 + (\text{Im}F_k)^2$ .

przedziału początkowych danych, a  $dx$  to odległość między kolejnymi punktami (np.  $dx = x_1 - x_0$ ). Zidentyfikuj ("na oko") częstości funkcji  $f(x)$ . Napisz klasę Fourier.

- Jakie zmienne powinna mieć ta klasa?
- konstruktor: konstruktor powinien przyjmować jeden argument- string ze ścieżką i nazwą pliku z danymi. Następnie ma wczytywać dane do dwóch vectorów.
- metoda DFT(), która wyznaczy  $F_k$ , spectrum i poprawne częstości dla osi poziomej. Wywołaj metodę na końcu konstruktora
- Napisz kod w pythonie do wyrysowania wykresu danych wejściowych oraz wykresu wyniku z transformaty Fouriera, wykorzystaj skalę logarymiczną w osi Y.

[ZOBACZ KOLEJNĄ STRONĘ DLA KODU]

Skorzystaj ze zdefiniowanego typu complex, przykład:

```
#include<vector>
#include<complex>
int main(){
complex<float> arg(0,0),val(15,0), I(0,1);
vector<complex<float>> dft;
}
```

Pamiętaj o kastowaniu zmiennych typu float na zmienne typu complex przy wyznaczaniu wykładnika exponensa. Przykładowy program do zapisywania danych z i do pliku może wyglądać następująco:

```
#include<iostream>
#include<vector>
#include<fstream>
#include<sstream>
#include<string>
using namespace std;
int main(){
ofstream file("output.dat");
if(file.is_open()){
cout<<"start!\n";
for(int i =0;i<10;i++){
file<<i<<" "<<i*i<<" "<<i*i*i<<"\n";
}
}
file.close();
int a,b,c;
ifstream file1("output.dat");
string line;
while(getline(file1,line)){
istringstream row{line};
row>>a>>b>>c;
cout<<b<<endl;
}
}
```