Jan Doniec A=5, B=1, C=2, D=5

................................................................... ---------------------------------

(Imię i nazwisko) (A, B, C, D)

Parametry:

M = 10

N = 14

norma = 1

**Raport z Pracowni nr 2**

**Zadanie 1.**

1. Cel zadania

Celem zadania było zbadanie jak zbieżność Iteracji Prostej jest zależna od parametru n (Parametr n - wielkość macierzy), przy normie macierzowej równej 1, wykorzystując metodę iteruj\_roznica().

1. Metody

Do przeprowadzenia doświadczenia wykorzystano komputer MacBook Air z procesorem Apple M1 ze zainstalowanym środowiskiem Visual Studio Code, wykorzystując pliki zawierające klasy napisane w języku Python, udostępnione studentom podczas kursu Metody Numeryczne. Część analizy danych odbyła się również w programie Microsoft Excel.

1. Przyjęte parametry

Do wykonania zadania przyjęto poniższe parametry:

* *eps*=1.0E-10 – parametr stopu
* *alfa*=0.3 – parametr zmiennej losującej układ
* *k*=5 – liczba pomiarow dla jednej wartosci parametru
* *norma*=1

1. Przebieg doświadczenia i wyniki

W zgodzie z wybranymi parametrami, przeprowadzone zostało kilka testów kontrolnych. Podczas sesji testowej rozważono dwie różne macierze o rożnych wartościach parametru n należących od 10 do 140.

A graph with red dots and blue dots

Description automatically generated

Rys.1 – Przedstawia normy kolejnych przybliżeń dla dwóch różnych parametrów n (n=10, n=140).

Uzyskano następujące wyniki:

Dla n=10:

Norma macierzy: 1.017589

Niedokładność rozwiązania: 6.619938e-11

Dla n=140:

Norma macierzy: 1.012952

Niedokładność rozwiązania: 1.355744e-10

Na podstawie powyższych wyników można sformułować następującą hipotezę:

Niezależnie od wzrostu rozmiaru macierzy norma oscyluje w zbliżonych wartościach.

W eksperymencie wykorzystano następujące wartości n=[10,20,30,40,50,60,70,80,90,100,110,120,130,140]

Wykorzystano również funkcje badaj\_zbieznosc():

def badaj\_zbieznosc(self):

param = [10,20,30,40,50,60,70,80,90,100,110,120,130,140]

sr\_norma\_macierzy = []

sr\_niedokladnosc = []

sr\_liczba\_iteracji = []

for n in param :

u1 = uklad.Uklad(wymiar=self.n)

norma\_macierzy = 0.0

liczba\_iteracji=0.0

niedokladnosc = 0.0

iteracje = 0

while iteracje < self.k:

u1.losuj\_uklad\_symetryczny\_dodatnio\_okreslony()

test1 = iteracjaprosta.IteracjaProsta(ukl=u1)

test1.przygotuj()

norma\_D = u1.norma\_macierzy(

typ=self.norma,

macierz=test1.D

)

iter=test1.iteruj\_roznica(

norma = self.norma,

eps= self.eps

)

niedokl = test1.sprawdz\_rozwiazanie(norma=self.norma)

if iter == 0:

continue

else:

norma\_macierzy += norma\_D

niedokladnosc += niedokl

liczba\_iteracji += iter

iteracje += 1

sr\_liczba\_iteracji.append(liczba\_iteracji/self.k)

sr\_norma\_macierzy.append(norma\_macierzy / self.k)

sr\_niedokladnosc.append(niedokl / self.k)

print("Wielkosc \nmacierzy \t \t ||D|| \t Iteracje \t Niedokladnosc")

print("-------" \* 9)

for i in range(len(param)):

wyniki = f"{param[i]} \t\t\t"

wyniki += f"{sr\_norma\_macierzy[i]:.6f} \t\t"

wyniki += f"{sr\_liczba\_iteracji[i]:.2f} \t"

wyniki += f"{sr\_niedokladnosc[i]:.6e} \n"

print(wyniki)

Wynikiem jej działania dla wcześniejszych parametrów jest zwrócenie w konsoli następującej sekwencji danych:

Wielkosc

macierzy ||D|| Iteracje Niedokladnosc

-----------------------------------------------------------------------------------------

10 1.017589 14.00 6.619938e-11

20 1.022122 13.80 5.337707e-11

30 1.006595 13.60 6.349509e-11

40 1.011566 13.80 7.751698e-11

50 1.014971 14.00 3.949318e-11

60 1.038630 13.80 5.534426e-11

70 1.014687 14.00 3.800554e-11

80 1.010966 14.00 5.530306e-11

90 1.007955 13.80 5.043926e-11

100 1.017232 14.00 3.133300e-11

110 1.004543 13.80 5.700721e-11

120 1.017006 14.00 3.759572e-11

130 1.014551 13.80 3.825631e-11

140 1.012952 13.80 1.355744e-10

Zauważmy, że:

-Wartości norm niewiele się zmieniają wraz ze wzrostem rozmiaru macierzy

-Im większy rozmiar macierzy tym mniejsza niedokładność

A graph with blue lines and numbers

Description automatically generated

A graph with blue line

Description automatically generated

1. Wnioski

Przy wzroście rozmiaru macierzy n:

* Nie wydaje się, aby rozmiar macierzy miał wpływ na normę macierzy
* Wydaje się, że średnio im większa macierzy tym mniejsza niedokładność

**Zadanie 2.**

1. Cel zadania

Celem zadania było zbadanie wpływu wartości gamma na efektywność uzyskiwania rankingu PageRank metodą Iteracji Seidla oraz Metodą Potęgową przy normie równej 0. Wykorzystano do tego metodę iteruj\_roznica().

1. Metody

Do przeprowadzenia doświadczenia wykorzystano komputer MacBook Air z procesorem Apple M1 ze zainstalowanym środowiskiem Visual Studio Code, wykorzystując pliki zawierające klasy napisane w języku Python, udostępnione studentom podczas kursu Metody Numeryczne. Część analizy danych odbyła się również w programie Microsoft Excel.

1. Przyjęte parametry

Do wykonania zadania przyjęto poniższe parametry:

* eps=1.0E-10 – parametr stopu
* k=5 – liczba pomiarow dla jednej wartosci parametru
* norma=0
* n=30 – rozmiar macierzy

1. Przebieg doświadczenia i wyniki

Dla wybranych wartości parametru gamma przeprowadzono kilka testów kontrolnych. Rozważono macierz na której zastosowano dwie metody – Iteracji Seidela oraz Metodę Potęgową. Przyjęto następujące wartości parametru gamma: 0.99 oraz 0.01.

* Dla 0.99:

Średnia liczba linkow: 29.866666666666667

**Iteracja Seidela**

-Liczba iteracji: 145

-Niedokładność: 7.95043822443553e-11

**Metoda Potęgowa**

-Liczba iteracji: 5

-Niedokładność: 0.04617518826059073

* Dla 0.01:

1. Wnioski

...