ZADANIE NUMERYCZNE 4

- Autor: Jan Jochymczyk

Treść: **(zadanie numeryczne NUM4)**

Zadana jest macierz A , ma liczby 12 na diagonali, 8 na pierwszej pozycji nad diagonalą,

a pozostałe elementy są równe 1. Wymiar macierzy ustalamy na $N = 80$.

oraz wektor $\mathbf{b} \equiv (5, \ldots, 5)^T$. Macierz **A** ma liczby 12 na diagonali, 8 na pierwszej pozycji nad diagonalą,

a pozostałe elementy są równe 1. Wymiar macierzy ustalamy na $N = 80$.

- Rozwiąż numerycznie równanie **Ay** = **b**, stosując odpowiednią metodę. **Uwaga:** algorytm należy go

  zaimplementować samodzielnie.

- Sprawdź swój wynik przy użyciu procedur bibliotecznych lub pakietów algebry komputerowej.
- Potraktuj *N* jako zmienną i zmierz czas potrzebny do uzyskania rozwiązania w funkcji *N*. Wynik przedstaw na wykresie. Czy wykres jest zgodny z oczekiwaniami?

1. Cel zadania

Celem zadania było było numeryczne rozwiązanie równania Ay = b, wybierając najlepsza metodę. Oczywiście metoda zależy od rodzaju macierzy A. W tym zadaniu macierz A jest macierzą posiadającą liczby 12 na diagonali, 8 na pierwszej pozycji nad diagonalą, a pozostałe elementy równe jeden. Wymiar macierzy wynosi 80.

Macierz A można przedstawić jako sumę dwóch macierzy. Jedna to macierz, której elementy to same 1, a druga to macierz A – 1.

2. Zastosowanie najlepszej metody

Macierz A można przekształcić na sumę dwóch macierzy. Macierz A bez przekształcenia nie jest macierzą gęstą. Macierz A trzeba jakimś sposobem przeksztalcic do macierzy rzadkiej. Można to uzyskać poprzez rozdzielenie macierzy A na sumę dwóch macierzy, a później zastosowanie metody Shermana Morrisona do znalezenia rozwiązania równania.

3. Wyniki

Rozwiazaniem równania jest wektor:

[0.05081874647092044, 0.050818746470920495, 0.05081874647092047, 0.05081874647092052, 0.05081874647092041, 0.05081874647092058, 0.050818746470920356, 0.05081874647092066, 0.050818746470920106, 0.05081874647092108, 0.050818746470919524, 0.05081874647092194, 0.05081874647091819, 0.050818746470924075, 0.050818746470914805, 0.050818746470929294,

0.05081874647094228, 0.05081874647088616, 0.05081874647097445,
0.05081874647105364, 0.05081874647124948,
0.05081874647292675,
0.05081874647587489, 0.05081874646313486, 0.05081874648315496,
0.05081874646501132245, 0.05081874646423444944, 0.0508187465455249,
0.05081874665514788, 0.05081874618142024, 0.05081874692584937,
0.05081874759431626, 0.05081874470558426, 0.05081874924502022,
0.05081874211162085, 0.05081875332124849 4, 0.05081873570611922, 0.05081876338703667,
0.05081871988845221, 0.05081878824337052, 0.05081868082849891, 0.05081884962329722,
0.05081858437432846, 0.05081900119413651 5, 0.05081834619158099, 0.0508193754813111,
0.05081775802602087, 0.05082029974147697 6, 0.05081630561718872, 0.05082258209821314,
0.05081271905660334, 0.05082821812199029, 0.05080386244781074, 0.05084213565009288,
0.05078199204650674, 0.05087650342357064, 0.05072798554532731 4, 0.05096137078256685,
0.05059462255261909 5, 0.05117094119967974, 0.05026529761144160 6, 0.05168845182153001,
0.04945206663424831, 0.05296638621426247, 0.04744388401709731 5, 0.05612210175549964,
0.04248490245229605, 0.06391478707161591, 0.03023925409839895, 0.08315794877059712]

Wynik z biblioteki numpy:

[0.05081875 0.05081875 0.05081875 0.05081875 0.05081875 0.05081875

0.05081875 0.05081875 0.05081875 0.05081875 0.05081875 0.05081875

0.05081875 0.05081875 0.05081875 0.05081875 0.05081875 0.05081875

0.05081875 0.05081875 0.05081875 0.05081875 0.05081875 0.05081875

0.05081875 0.05081875 0.05081875 0.05081875 0.05081875 0.05081875

0.05081875 0.05081875 0.05081875 0.05081875 0.05081875 0.05081875

0.05081875 0.05081875 0.05081875 0.05081875 0.05081875 0.05081875

0.05081874 0.05081875 0.05081874 0.05081875 0.05081874 0.05081876

0.05081872 0.05081879 0.05081868 0.05081885 0.05081858 0.050819

0.05081835 0.05081938 0.05081776 0.0508203 0.05081631 0.05082258

0.05081272 0.05082822 0.05080386 0.05084214 0.05078199 0.0508765

0.05072799 0.05096137 0.05059462 0.05117094 0.0502653 0.05168845

0.04945207 0.05296639 0.04744388 0.0561221 0.0424849 0.06391479

0.03023925 0.08315795]

4. Dygresja o metodach, wynikach oraz czasach działania algorytmów.

Wyniki są niemal identyczne, ale dla algorytmu napisanego samodzielnie są bardziej precyzyjne. Czas działania algorytmów się różni. Algorytm samodzielnie zaimplementowany działa znacznie szybciej niż algorytm rozwiązywania z biblioteki numerycznej. Dzieje się tak, ponieważ algorytm napisany samodzielnie można dostosować do struktury macierzy, którą mamy podaną. Stosując „trik" podany w punkcie „2." dla dużych N jest zbawieniem przed ogromną złożonością obliczeniową oraz czasową.

5. Wykres

Wykres znajduje się w pliku „wykres.jpeg", jest zapisany w grafice rastrowej. Obrazuje zależności czasu działania programu dla samodzielnie napisanego algorytmu z czasem działania algorytmu z biblioteki numerycznej. W punkcie 4. napisane jest „dla dużych N jest zbawieniem przed ogromną złożonością obliczeniową oraz czasową", widać to na wykresie. Gdy N rośnie, to czas działania algorytmu z biblioteki numerycznej rośnie

znacznie szybciej, niż algorytm napisany samodzielnie. Wykres jest zgodny z oczekiwaniami, ponieważ algorytm napisany samodzielnie działa szybciej, niż gotowy algorytm z biblioteki numerycznej.