NUM3

Cel:

Obliczenie macierzy $y = A^{-1}x$ uważając na fakt, iż macierz A jest rzadka.

Opis ćwiczenia:

1. Rozkład LU
2. Zastosowanie forward_substitution (Podstawienie w przód)
3. Zastosowanie backward_substitution(Podstawienie wstecz)
4. Obliczenie wyznacznika macierzy

Teoria:

Rozkład LU: to metoda numeryczna, która polega na rozłożeniu macierzy kwadratowej A na iloczyn dwóch macierzy trójkątnych: dolnej L (Lower) i górnej U (Upper).

Macierz L jest macierzą trójkątną dolną, co oznacza, że wszystkie wyrazy leżące ponad główną przekątną są równe zero. Wyrazy na przekątnej głównej macierzy L są równe 1.

Macierz U jest macierzą trójkątną górną, co oznacza, że wszystkie wyrazy leżące poniżej przekątnej głównej są równe zero

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1N} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2N} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3N} \\ \dots & \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & a_{N3} & \dots & a_{NN} \end{bmatrix}}_{\mathbf{A}} = \underbrace{\begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \dots & \dots & \dots & \dots & \\ l_{N1} & l_{N2} & l_{N3} & \dots & 1 \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1N} \\ & u_{22} & u_{23} & \dots & u_{2N} \\ & & u_{33} & \dots & u_{3N} \\ & & & \dots & \dots \\ & & & \dots & u_{NN} \end{bmatrix}}_{\mathbf{U}}$$

W przypadku macierzy rzadkich, gdzie większość elementów ma wartość zero, efektywna faktoryzacja LU może być znacznie przyspieszona poprzez wykorzystanie własności strukturalnych macierzy. W szczególności, jeśli macierz ma tylko cztery niezerowe diagonale, można to wykorzystać do zastosowania faktoryzacji LU w czasie liniowym. W tradycyjnym algorytmie faktoryzacji LU, potrzebujemy iterować przez wszystkie elementy macierzy, co ma złożoność czasową $O(n^3)$. Jednak w przypadku, gdy macierz ma strukturę rzadką, można znacznie przyspieszyć ten proces. Jeśli znasz strukturę macierzy i wiesz, że ma tylko cztery niezerowe diagonale, można zoptymalizować proces faktoryzacji LU poprzez pominięcie zerowych elementów. Dzięki temu unikamy iteracji przez wiele zerowych elementów, co znacząco obniża złożoność czasową. Doolittle jest jednym z algorytmów faktoryzacji LU, który można zastosować w tym przypadku. W efekcie można osiągnąć złożoność czasową O(n), co jest znaczną poprawą w porównaniu do złożoności klasycznego problemu algorytmu faktoryzacji LU. Dodatkowo, warto zauważyć, że wykorzystanie algorytmu Doolittle'a w kontekście macierzy rzadkich nie tylko przyspiesza obliczenia, ale również znacznie redukuje zużycie pamięci, co jest kluczowe w przypadku dużych macierzy.

Algorytm Doolitle'a i jego jawne wzory

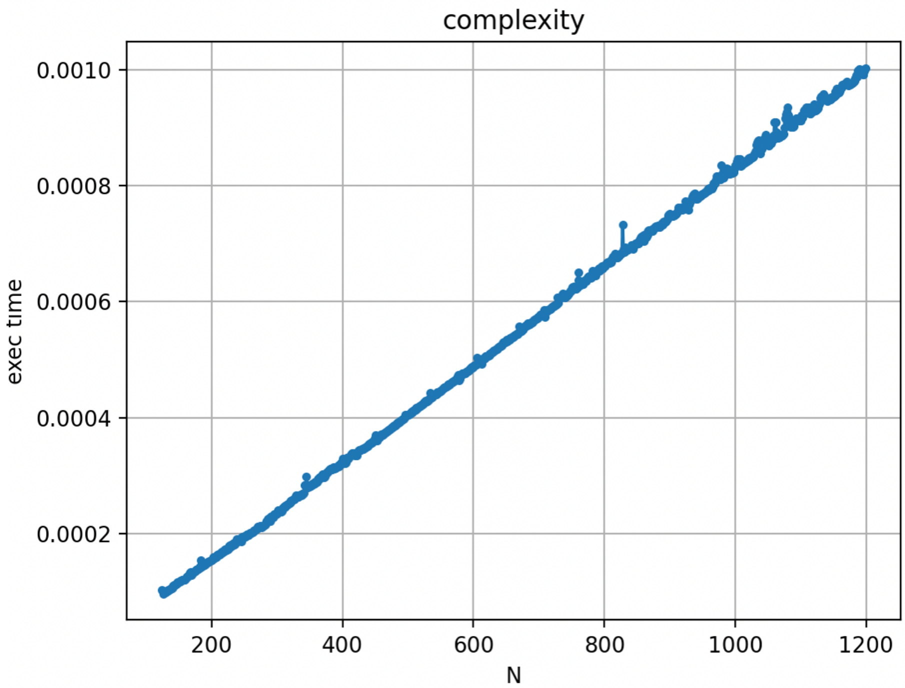
$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}$$

$$l_{ij} = \frac{1}{u_{jj}} (a_{ij} - \sum_{k=1}^{j-1} (l_{ik} u_{kj})$$

Wyniki/Wnioski:

Własna implementacja działa znacząco krócej od tej rozwiązywanej przez pakiet NUMPY, ponadto wyniki są niemal identyczne(błąd typu 10e-15).

Wykres złożoności, jak widać jest ona liniowa.



Rozwiązanie:

```
Wyznacznik jest równy:
 6141973498.857843
```

y = [0.448700827728733, 1.4132732869357947, 2.1348778535462736, 2.8690132654396248, 3.5914885705595205, 4.311604959915503, 5.029827173723323, 5.747011462584994, 6.463503693914558, 7.179525964548697, 7.8952125968955915, 8.610651859797315, 9.325903619364162, 10.041009954626537, 10.756001271894783, 11.470900080737994, 12.185723394049369, 12.900484305313817, 13.615193053266005, 14.329857758065131, 15.044484941235352, 15.759079899902147, 16.473646980766127, 17.18818978377055, 17.902711315621058, 18.617214106977666, 19.331700302956037, 20.046171733762908, 20.76062997036838, 21.47507636878333, 22.189512105570603, 22.903938206548368, 23.6183555701598, 24.332764986629712, 25.047167153767735, 25.761562690082965, 26.475952145728595, 27.190336011683936, 27.904714727496145, 28.61908868783829, 29.33345824808956, 30.047823729103516, 30.762185421298863, 31.476543588182352, 32.19089846939369, 32.90525028334641, 33.61959922952588, 34.33394549049516, 35.048289233651346, 35.762630612767495, 36.4769697693505, 37.19130683383959, 37.90564192666726, 38.6199751592003, 39.33430663457682, 40.04863644845216, 40.762964689665075, 41.47729144083417, 42.19161677889273, 42.905940775569235, 43.62026349782013, 44.33458500822004, 45.04890536531427, 45.763224623938, 46.47754283550541, 47.19186004827236, 47.90617630757509, 48.62049165604775, 49.334806133820685, 50.04911977870149, 50.76343262634067, 51.47774471038327, 52.192056062607854, 52.9063667130542, 53.62067669014054, 54.33498602077156, 55.04929473043772, 55.76360284330703, 56.47791038230969, 57.192217369216245, 57.906523824710035, 58.62082976845425, 59.335135219153926, 60.049440194613666, 60.763744711791205, 61.47804878684707, 62.192352435190934, 62.90665567152471, 63.62095850988271, 64.3352609636691, 65.04956304569288, 65.76386476820053, 66.47816614290662, 67.19246718102224, 67.90676789328191, 68.62106828996849, 69.33536838093679, 70.0496681756356, 70.76396768312829, 71.47826691211242, 72.19256587093781, 72.90686456762393, 73.62116300987596, 74.33546120510012, 75.04975916041795, 75.76405688267984, 76.47835437847795, 77.19265165415811, 77.90694871583132, 78.62124556938441, 79.33554222049035, 80.04983867461782, 80.76413493704023, 81.47843101284448, 82.19272690693916, 82.90702262406211, 83.62131816878798, 84.33561354553508, 85.04990875857204, 85.76420381203626, 86.47849870460276, 87.19279247126808, 87.90778524137869, 88.68203579310355]