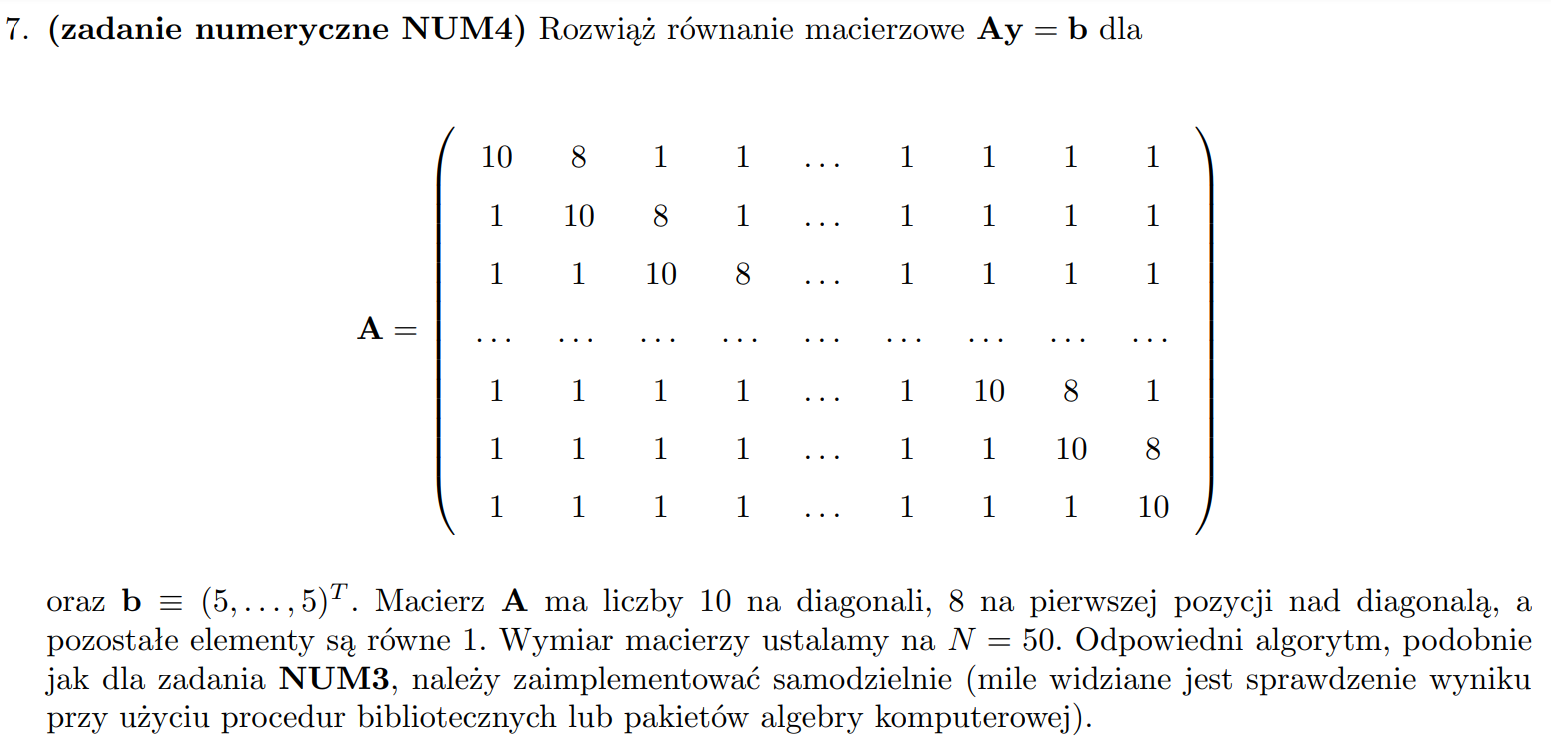
# Zadanie numeryczne 4



# Wprowadzenie

Celem zadania jest napisanie programu rozwiązującego równanie ze skomplikowaną macierzą w bardzo dobrej złożoności obliczeniowej. W tym celu powinien zostać wykorzystany wzór Shermana-Morrisona, pozwalający rozbić obliczanie na operacje rzędu .

# Wynik

Problem rozwiązuje program program.py, który nie wykorzystuje żadnych dodatkowych bibliotek, ponad bazowe możliwości pythona.

Jako że po rozbiciu zapisaniu macierzy wejściowej w formie , gdzie wektory i są równe , otrzymujemy macierz A która ma na diagonali same 9 i w pierwszym pasie nad diagonalą same 7, rozkład LU jest nie potrzebny, jako że dana macierz jest macierzą trójkątną górną, więc dla takiego A: , a więc wszystkie działania tą macierzą na wektory można od razu wykonywać poprzez „backward substitution”.

Program nie przechowuje macierzy U w pamięci, używa funkcji pomocniczej, która po prostu zwraca odpowiednią wartość macierzy U na podstawie otrzymanych indeksów. Tak samo program nie przechowuje w pamięci wektora , ani , ani , jako że wszystkie ich wartości są takie same.

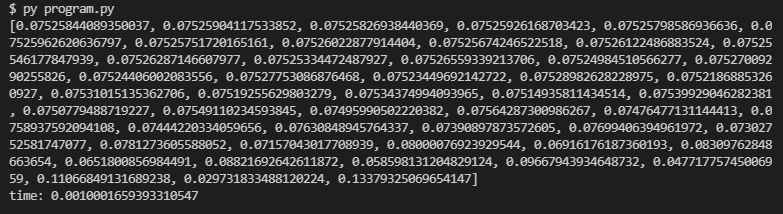
Najpierw program oblicza i , rozwiązując równania i .

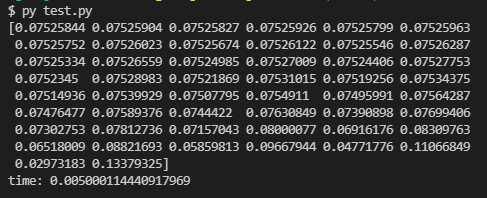
Następnie program oblicza elementy równania wyprowadzonego ze wzoru Shermana-Morrisona co daje wynik:

(0.07525844089350037, 0.07525904117533852, 0.07525826938440369, 0.07525926168703423, 0.07525798586936636, 0.07525962620636797, 0.07525751720165161, 0.07526022877914404, 0.07525674246522518, 0.07526122486883524, 0.07525546177847939, 0.07526287146607977, 0.07525334472487927, 0.07526559339213706, 0.07524984510566277, 0.07527009290255826, 0.07524406002083556, 0.07527753086876468, 0.07523449692142722, 0.07528982628228975, 0.07521868853260927, 0.07531015135362706, 0.07519255629803279, 0.07534374994093965, 0.07514935811434514, 0.07539929046282381, 0.0750779488719227, 0.07549110234593845, 0.07495990502220382, 0.07564287300986267, 0.07476477131144413, 0.0758937592094108, 0.07444220334059656, 0.07630848945764337, 0.07390897873572605, 0.07699406394961972, 0.07302752581747077, 0.0781273605588052, 0.07157043017708939, 0.08000076923929544, 0.06916176187360193, 0.08309762848663654, 0.0651800856984491, 0.08821692642611872, 0.058598131204829124, 0.09667943934648732, 0.04771775745006959, 0.11066849131689238, 0.029731833488120224, 0.13379325069654147) ᵀ

# Dyskusja wyników

W celu weryfikacji wyników napisałem dodatkowy program test.py, który po wygenerowaniu wejściowej macierzy A, wykorzystuje funkcję biblioteczną numpy.linalg.solve do wyliczenia wektora :





Mimo niższej precyzji rozwiązania bibliotekowego widać, że wyniki są zgodne.

Oprócz tego oba programy mierzą swój czas wykonania, dzięki czemu widać, że program.py działa około 5 razy szybciej.