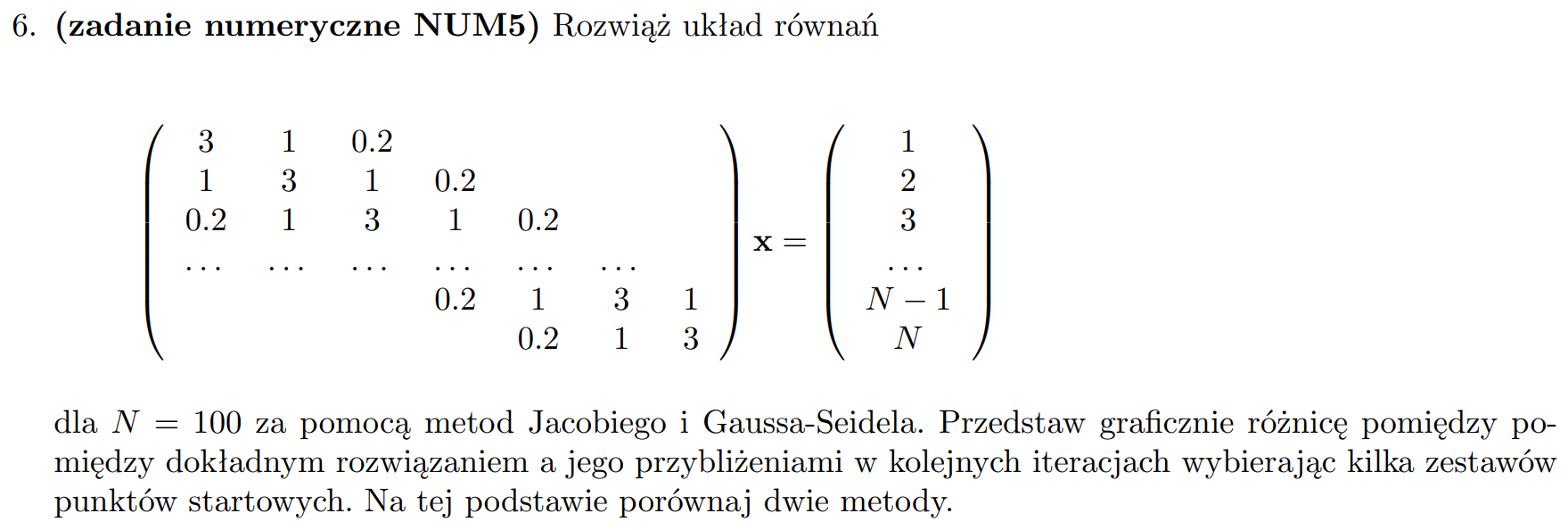
# Zadanie numeryczne 5



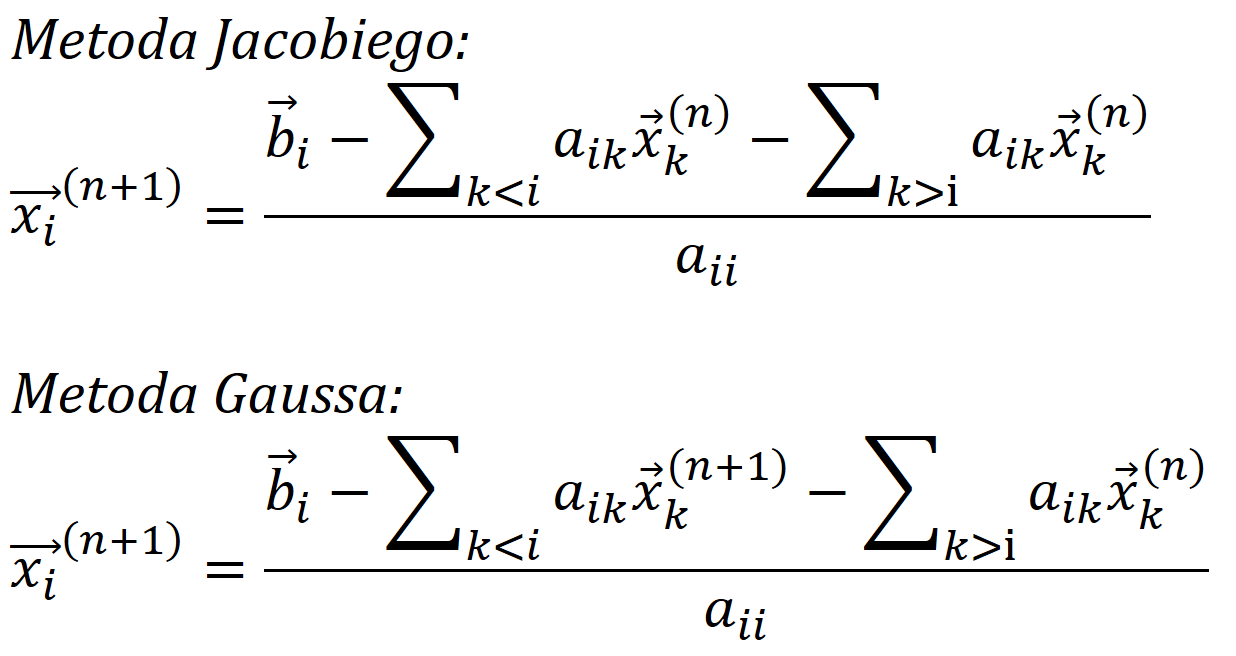
# Wprowadzenie

Celem zadania jest napisanie programu wyliczającego przybliżenie rozwiązania równania za pomocą metod iteracyjnych Jacobiego i Gaussa-Seidela oraz graficzne przedstawienie zmian w przybliżeniach w kolejnych iteracjach.

# Wynik

Problem rozwiązuje program program.py, wykorzystujący biblioteki numpy oraz matplotlib.

Do obliczenia wyniku wykorzystuje metody iteracyjne Jacobiego i Gaussa-Seidela w postaci następujących wzorów:

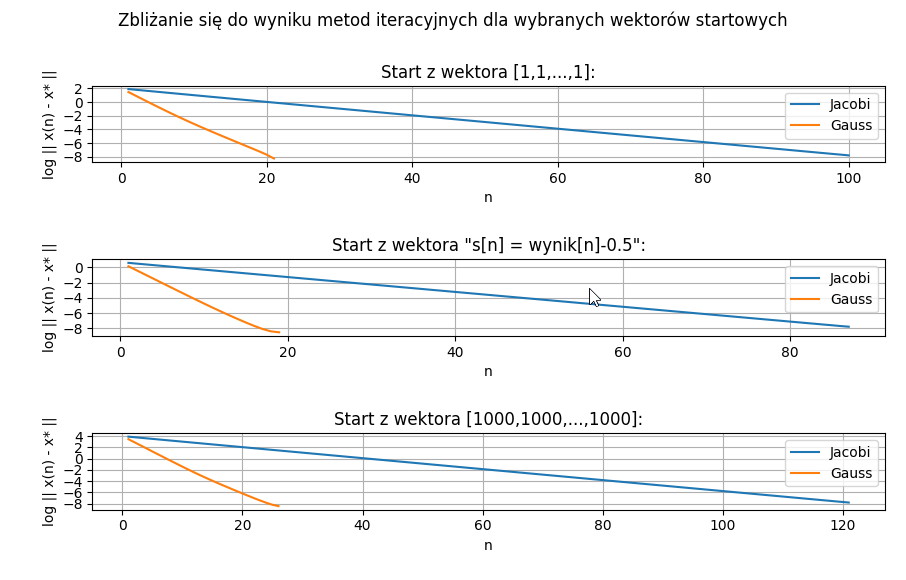


Program najpierw oblicza wynik z dokładnością 0.00000001 za pomocą metody Gaussa, gdzie warunkiem zbieżności metody jest fakt, że różnica pomiędzy normą wektora z poprzedniej i aktualnej iteracji jest mniejsza niż 0.00000001.

Przy obliczaniu wyniku, program wykorzystuje wstęgową strukturę macierzy A, której nawet nie musi trzymać w pamięci, jako że jest wartość w dowolnym punkcie można bez problemu zwrócić z funkcji. Także we wzorach iteracyjnych, zamiast pełnych sum po wszystkich elementach nad/pod diagonalą, program wykonuje tylko te operacje, dla których wie, że elementy macierzy A będą niezerowe.

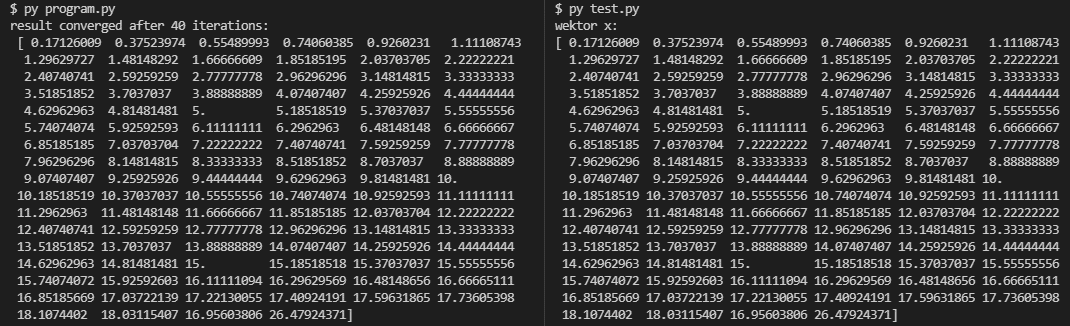
Następnie program wykonuje obie metody iteracyjne dla wektorów startowych [1,1,…,1], [1000,1000,…,1000] oraz wektora, którego każda wartość jest równa odpowiadającej jej wartości wektora rozwiązania – 0.5. W trakcie wykonywania tych metod program oblicza normę dla różnicy wektora danej iteracji od wektora wyniku i zapisuje ją w liście.

Na koniec program wyświetla wygenerowane wykresy oraz wypisuje w konsoli obliczony wektor wynikowy.



# 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 .



Jak widać, wynik zgadza się z wynikiem otrzymanym metodą iteracyjną

Na otrzymanych wykresach widać również, że niezależnie od wybranego wektora startowego, po odpowiedniej ilości iteracji otrzymany wynik jest bardzo precyzyjny. Metoda Gaussa-Seidela otrzymuje precyzyjny wynik w o wiele mniejszej liczbie iteracji.