

Sprawozdanie z projektu

Klasyczna metoda SOR dla równań typu $Ax = b$

Miłosz Zieliński

14.01.2024

Treść

Opis metody	3
Rozwiązanie problemu	4
Przypadki	5
Przypadek 1	5
Przypadek 2	6
Przypadek 3	7
Przypadek 4	8
Przypadek 5	10
Przypadek 6	10
Podsumowanie.....	11

Opis metody

Metoda SOR (metoda sukcesywnej nadrelaksacji) to iteracyjny sposób rozwiązywania układów równań liniowych postaci:

$$Ax = b,$$

gdzie A to macierz współczynników, x to wektor niewiadomych, a b to wektor wyrazów wolnych.

Metoda SOR jest ulepszeniem metody Gaussa-Seidla poprzez wprowadzenie parametru relaksacji ω , który ma na celu przyspieszenie zbieżności. Metoda zakłada, że rozwiązanie układu równań można iteracyjnie poprawiać korzystając z następującego wzoru:

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right),$$

Dla $i = 1, 2, \dots, n$, gdzie:

- $x_i^{(k+1)}$ to wartość i -tej zmiennej w $k + 1$ tej operacji
- a_{ii} to element diagonalny macierzy A
- $\omega \in (1, 2)$ to parametr relaksacji, jeżeli wynosiłby 1, to byłaby to zwykła metoda Gaussa-Seidla

Aby metoda była zbieżna to promień spektralny macierzy iteracyjnej T_{SOR} musi być mniejszy niż 1, a $\omega \in (0, 2)$.

Rozwiązanie problemu

W zadaniu miałem skupić się na rozwiązywaniu równań liniowych $Ax = b$, gdzie $A(n \times n)$ jest macierzą postaci

$$A = \begin{pmatrix} C & S \\ -S & C \end{pmatrix},$$

gdzie $C, S(p \times p)$ i $n = 2p$. Przy czym $C = \text{diag}(c_1, \dots, c_p)$ i $\det C \neq 0$ oraz $S = \text{diag}(s_1, \dots, s_p)$, gdzie $c_i^2 + s_i^2 = 1$, dla $i = 1, \dots, p$.

W celu rozwiązania problemu stworzyłem zestaw plików:

- `generate_matrix(c)`
Funkcja tworzy i zwraca macierz A na podstawie zadanego wektora c . Macierz A w U ma wartości dodatnie, a w L wartości ujemne, ale to gdzie są jakie nie powinno wpływać na wyniki.
- `spectral_radius(A)`
Funkcja wylicza odpowiedni promień spektralny na podstawie zadanej macierzy A i go zwraca.
- `optimal_omega(A)`
Funkcja wylicza „najlepszy” dla danej macierzy parametr ω i go zwraca.
- `sor_method(A, b, omega, tol, maxIter, xStart)`
Jest to właściwa funkcja w całej metodzie. Przyjmuje ona macierz parametrów A , wektor wyrazów wolnych, parametr ω , tolerancję błędu (warunek zakończenia obliczeń), maksymalną liczbę iteracji i wektor przybliżeń początkowych. Zwracany jest wektor wynikowy x . Pozostałem przy metodzie z dwoma pętlami, tak jak było na wykładzie, ponieważ dzięki temu miałem z jakiegoś powodu o kilka procent mniejszą liczbę iteracji.
- `plot_omega_vs_radius(A)`
Pomocnicza funkcja do narysowania wykresu
- `plot_iterations_vs_omega(A, b, tol, max_iter, xStart)`
Również funkcja pomocnicza do rysowania wykresu.
- `test`
W tym pliku generowane są wszystkie przypadki

W macierzach, które sprawdzam uwarunkowanie macierzy jest równe 1, wynik to z konstrukcji macierzy. Dlatego nie sprawdzam wskaźnika uwarunkowania w przykładach.

Przypadki

W poniższych przypadkach starałem się pokazać jak działa oraz jakie są wady metody SOR dla macierzy zadanej postaci.

Przypadek 1 (zwykły)

Na przykładzie macierzy

0.8500	0	0	0.5268	0	0
0	0.8500	0	0	0.5268	0
0	0	0.8500	0	0	0.5268
-0.5268	0	0	0.8500	0	0
0	-0.5268	0	0	0.8500	0
0	0	-0.5268	0	0	0.8500

Można zobaczyć, że metoda działa prawidłowo:

Przypadek 1

Porównanie wyników:

Index	ExactSolution	SOR_Solution	Error
1	-1.2571	-1.2571	1.6888e-07
2	-0.93391	-0.93391	2.3305e-07
3	-0.6107	-0.6107	2.9721e-07
4	3.9268	3.9268	1.4157e-07
5	5.3036	5.3036	1.9536e-07
6	6.6803	6.6803	2.4915e-07

Przypadek 2 (za duży promień spektralny)

Aby metoda była zbieżna, to promień spektralny macierzy musi być mniejszy od 1. W przypadku macierzy A ma ona w każdym wierszu tylko dwa niezerowe parametry. Aby metoda była zbieżna to macierz ta musi być zgodnie uporządkowana, czyli wszystkie elementy na diagonalu są niezerowe oraz zachodzi warunek:

$$a_{ii} > \sum_{j=1, j \neq i}^n |a_{ij}|$$

Czyli dla każdego wiersza element na diagonalu ma większą wartość od drugiego elementu w wierszu.

W przypadku rozpatrywanej macierzy dla macierzy C i S zachodzi: $c_i^2 + s_i^2 = 1$, tym samym dla

wszystkich elementów na diagonalu zachodzić musi: $c_i > \sqrt{1 - s_i^2} \rightarrow c_i > \frac{\sqrt{2}}{2}$.

W przypadku macierzy na której diagonalu znajdują się kolejne wartości:

`c = [0.98;0.75;0.85]`

Porównanie wyników:

Index	ExactSolution	SOR_Solution	Error
1	0.18401	NaN	NaN
2	-1.8072	NaN	NaN
3	-0.6107	NaN	NaN
4	4.119	NaN	NaN
5	5.0729	NaN	NaN
6	6.6803	NaN	NaN

Promień spektralny:

2.0938

Widzimy, że promień spektralny jest większy od 1, dlatego metoda nie zbiega.

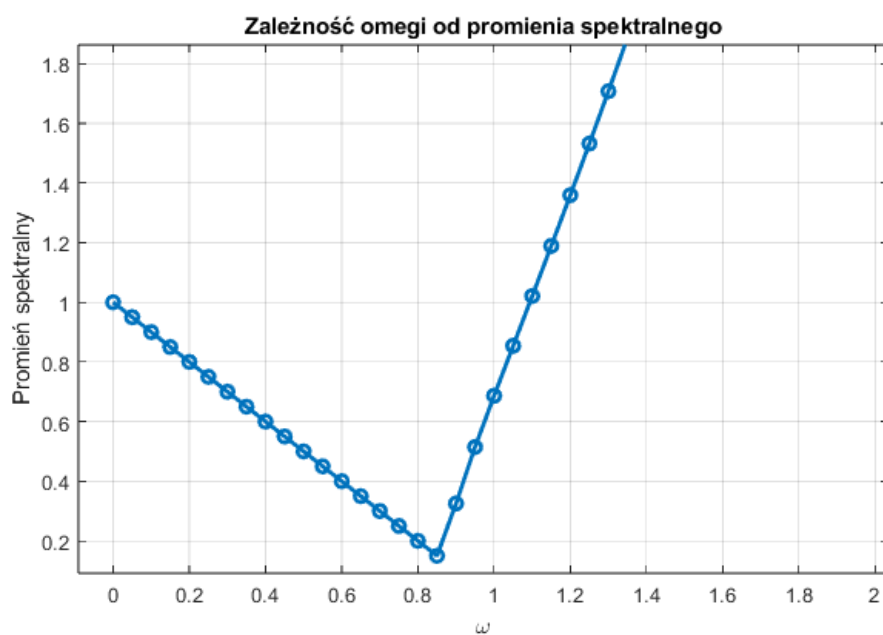
Przypadek 3 (promień spektralny, a wartość ω)

W poprzednich przypadkach do wyznaczenia ω używam funkcji `optimal_omega`. Wyznacza ona ω dzięki której promień spektralny macierzy iteracji SOR będzie najmniejszy, a tym samym metoda będzie najszybciej zbiegać. Parametr ten można wyliczyć ze wzoru:

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \mu^2}}.$$

, gdzie μ jest promieniem spektralnym macierzy iteracji Jacobiego.

źródło: <https://mst.mimuw.edu.pl/lecture.php?lecture=mo2&part=Ch5#S3.SS3>

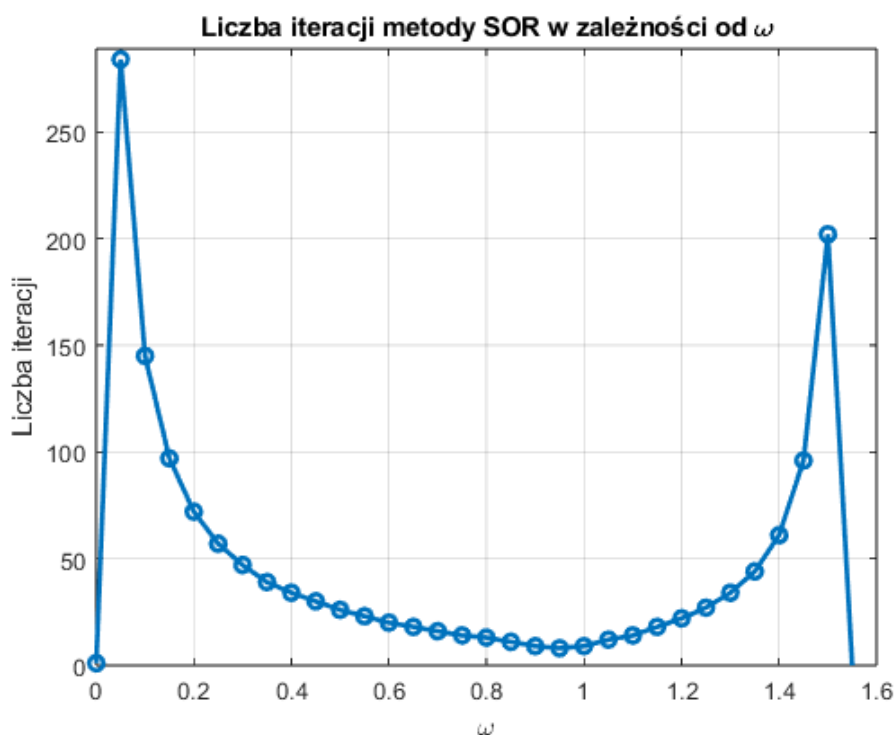


Na załączonej grafice widać, iż od pewnej ω metoda będzie rozbieżna. Dlatego należy zwrócić uwagę na dobór ω , aby funkcja nie była rozbieżna.

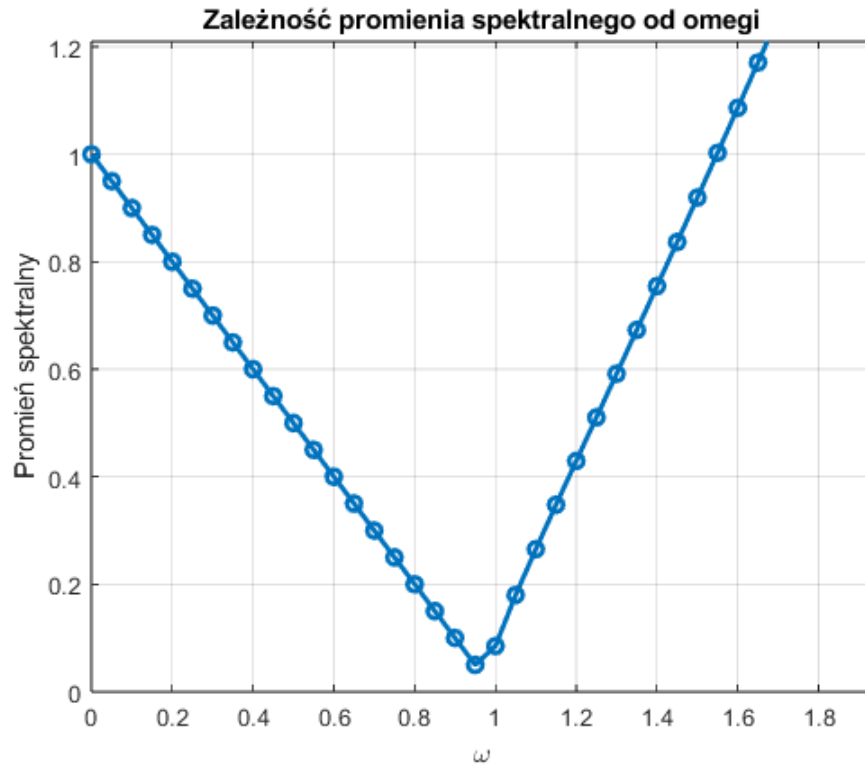
Przypadek 4 (wartość omega, a liczba iteracji)

Zależy nam, aby liczba iteracji była jak najmniejsza. Poniższy wykres przedstawia jak wartość omegi wpływa na liczbę iteracji.

(dla macierzy o diagonalnej $c = [0.96; 0.97; 0.97; 0.98; 0.99; 0.97]$)



Analizując powyższy wykres możemy wywnioskować, że liczba iteracji metody SOR jest powiązana od parametru ω . Widzimy, że od pewnego miejsca metoda jest rozbieżna (powyżej 1.5) oraz osiąga minimum iteracji dla ω wynoszącej około 0.95.



Możemy również zauważyć, że promień spektralny jest najmniejszy dla ω wynoszącej około 0.95. Szybkość zbieżności jest ściśle powiązana z promieniem spektralnym.

W momencie, gdy liczyłem optymalną ω , to wartość, która mi wychodziła nie pokrywała się z optymalną ω ą widzianą na wykresie (tam gdzie liczba iteracji najmniejsza). W przypadku innych macierzy macierz „normalnych” jakie sprawdzałem to był optymalny wynik, ale nie w przypadku podanych w treści zadania. Widać to na powyższych grafikach. Aby zachodziła metoda SOR to ω powinna być większa od 1 i tak mi wychodziło z obliczeń funkcji `optimal_omega(A)`. Nie jest to jednak zgodne z tymi wykresami i innymi przypadkami jakie analizowałem. Wydaje mi się więc, że wynika to z konstrukcji macierzy jakie miałem rozważać w tym zadaniu.

Przypadek 5 (zmiana wektora początkowego przybliżeń)

Zmiana początkowego wektora przybliżeń nie wpływa na to, czy metoda będzie zbieżna, ale może wpływać na to, ile iteracji wykona algorytm. Pokazałem to dla trzech przybliżeń początkowych, wektora zer, wektora wartości -50 i wektora wartości 50. Na grafice poniżej znajduje się także wektor rozwiązań problemu, który był liczony w tym przykładzie.

Tabela porównawcza liczby iteracji:

Start_Vector	Iterations
"zeros"	33
"-50 * ones"	35
"50 * ones"	31

Columns 1 through 11										
-5.1236	-3.5600	-2.2000	0.6911	-1.1032	-1.7682	-6.4112	18.4051	22.9200	29.3455	31.7730
Columns 12 through 14										
38.5847	42.9753	45.7919								

Widzimy, że wektor przybliżeń początkowych wpływa na liczbę iteracji. Dobranie jednak go w „ciemno” może być trudne.

Przypadek 6 (wartości na przekątnej, a liczba iteracji)

Poniżej chciałbym pokazać, że im wartości na przekątnej bardziej różnią się od pozostałych wartości (silna dominacja na przekątnej) tym algorytm będzie szybciej zbiegał.

Do stworzenia macierzy wykorzystałem macierze stworzone z następujących wektorów:

c1 = [0.99;0.98;0.98;0.98;0.99;0.99;0.98];

c2 = [0.86;0.87;0.87;0.88;0.89;0.87;0.88];

c3 = [0.76;0.79;0.77;0.795;0.79;0.79;0.79];

Można zauważyć, każdy kolejny wektor ma mniejsze wartości.

Tabela wyników dla Przypadku 6:

Case	Spectral_Radius	Condition	Iterations
"c1"	0.060406	1	8
"c2"	0.3789	1	19
"c3"	0.76587	1	66

Z tabelki widzimy, że im mniejsze wartości, tym większy promień spektralny, tym samym większa liczba iteracji. Przykłady były liczone dla tej samej ω .

Chciałem również sprawdzić, czy jest tak samo, jeżeli na przekątnej znajdują się wartości tylko ujemne. Okazuje się, że tak się nie dzieje, co jest bardzo ciekawe. Poniżej wykorzystałem te same wektory co przed chwilą, ale ze zmienionymi znakami.

Tabela wyników dla Przypadku 6 (ujemne):

Case	Spectral_Radius	Condition	Iterations
"c4"	0.12555	1	171
"c5"	0.48299	1	25
"c6"	0.90349	1	169

W tym przypadku mniejszy promień spektralny nie oznacza szybszej iteracji. Co jest bardzo ciekawe.

Podsumowanie

Metoda SOR dość dobrze rozwiązuje układy równań typu $Ax = b$ dla macierzy podanych w treści zadania. Chociaż należy pamiętać, że wartości na diagonalnej muszą być odpowiednio duże, aby promień spektralny macierzy był mniejszy od 1. Pojawiają się jednak pewne problemy z dobraniem odpowiedniej wartości ω , ponieważ przynajmniej w przypadkach analizowanych przeze mnie optymalna omega wydaje się być bardzo często mniejsza od 1, co nie do końca zgodne jest z metodą SOR i przeczy wynikowi uzyskiwanemu przez funkcję szukającą optymalną ω .

Zastosowania metody są bardzo szerokie, praktycznie w każdej dziedzinie wymagającej obliczeń dużych układów równań. Wykorzystuje się ją w rozwiązywaniu układów równań, gdy metoda Jacobiego lub Gaussa-Seidel może zbiegać za wolno. Ograniczeniem tej metody jest jednak odpowiednie dobranie parametru ω .