Zadanie numeryczne 5

Igor Tyszer

Grudzień 2021

1. Wstęp

Chcemy rozwiązać poniższy układ równań:

$$\begin{pmatrix} 3 & 1 & 0.2 & & & & \\ 1 & 3 & 1 & 0.2 & & & \\ 0.2 & 1 & 3 & 1 & 0.2 & & \\ \dots & \dots & \dots & \dots & \dots & \dots & \\ & & & 0.2 & 1 & 3 & 1 \\ & & & & 0.2 & 1 & 3 \end{pmatrix} \text{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ \dots \\ N-1 \\ N \end{pmatrix}$$

za pomocą metod Jacobiego i Gaussa-Seidela. Zostanie również przedstawiona graficzna różnica pomiędzy dokładnym rozwiązaniem a jego przybliżeniami w kolejnych iteracjach poprzez wybranie kilka zestawów punktów startowych. Na tej podstawie porównane zostaną obie metody.

2. Metoda Jacobiego i metoda Gaussa-Seidela.

Obie metody są metodami iteracyjnymi. W metodach iteracyjnych rozwiązanie dokładne otrzymuje się, teoretycznie, w granicy nieskończenie wiele kroków – w praktyce liczymy na to, że po skończonej (i niewielkiej) liczbie kroków zbliżymy się do wyniku ścisłego w granicach błędu zaokrąglenia.

Metoda Jacobiego

$x_i^{(k+1)} = (b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{N} a_{ij}\, x_j^{(k)}) \,/\, a_{ii}$

Górny indeks $x^{(k)}$ oznacza, że jest to przybliżenie w k-tym kroku.
Zauważmy, że w tej metodzie nie wykorzystuje się najnowszych przybliżeń. Na przykład obliczając $x_2^{(k+1)}$ korzystamy z $x_1^{(k)}$, mimo iż znane jest już wówczas $x_1^{(k+1)}$.

Metodę tę łatwo można zrównoleglić. Ulepszenie to stosuje metoda Gaussa-Seidela.

$$x_i^{(k+1)} = (b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{N} a_{ij}\, x_j^{(k)}) \,/\, a_{ii}$$

Dla numerycznej efektywności obu metod numerycznych, w swoim programie uwzględniłem rzadką strukturę macierzy podanej w zadaniu, która posiada pięć diagonali, a reszta to elementy zerowe.

3. Wyniki i wykresy

Poniżej przedstawiony jest wynik równania dla obu metod, a także wykresy prezentujące graficznie różnicę pomiędzy wybranymi rozwiązaniami, a jego przybliżeniami w kolejnych iteracjach w zależności od wyboru zestawu punktu startowych.

Wynik dokładny, uzyskany za pomocą biblioteki numerycznej:

[0.1712600924915493 0.3752397374515707 0.5548999253689071
 0.740603848924158 0.9260230950961975 1.1110874263605377
 1.2962972717646515 1.481482921363533 1.666666089813675
 1.8518519452948385 2.037037047738371 2.2222222114536554
 2.4074074103683 2.592592592366967 2.7777777776340296
 2.9629629630301957 3.148148148135447 3.3333333333327193
 3.5185185185196883 3.7037037037033413 3.8888888888889266
 4.0740740740740895 4.25925925925925 4.444444444444447
 4.62962962962963 4.814814814814814 5.000000000000001
 5.185185185185184 5.37037037037037 5.555555555555556
 5.740740740740741 5.925925925925926 6.111111111111111
 6.296296296296297 6.481481481481482 6.666666666666668
 6.851851851851851 7.037037037037037 7.2222222222222205
 7.407407407407407 7.5925925925925934 7.777777777777778
 7.962962962962963 8.14814814814815 8.333333333333332
 8.518518518518517 8.703703703703704 8.888888888888888
 9.074074074074076 9.25925925925926 9.444444444444446
 9.629629629629628 9.814814814814813 10.000000000000002
10.185185185185183 10.370370370370372 10.555555555555555
10.740740740740742 10.925925925925926 11.11111111111111
11.296296296296296 11.48148148148148 11.666666666666666
11.851851851851851 12.037037037037036 12.222222222222223
12.407407407407407 12.592592592592592 12.77777777777778
12.962962962962964 13.148148148148149 13.333333333333332
13.518518518518515 13.703703703703695 13.888888888888957
14.07407407407391 14.259259259259084 14.444444444447468

14.629629629617883 14.814814814829209 15.000000000089301
15.185185184578442 15.37037037200435 15.555555556025894
15.740740716823789 15.92592603089244 16.111110941047105
16.296295691233045 16.481486556842203 16.666651111165365
16.85185669432779 17.037221385436535 17.221300545231433
17.409241909490532 17.59631865256542 17.73605398322119
18.10744020272868 18.031154065721086 16.95603806179297
26.479243708354268]

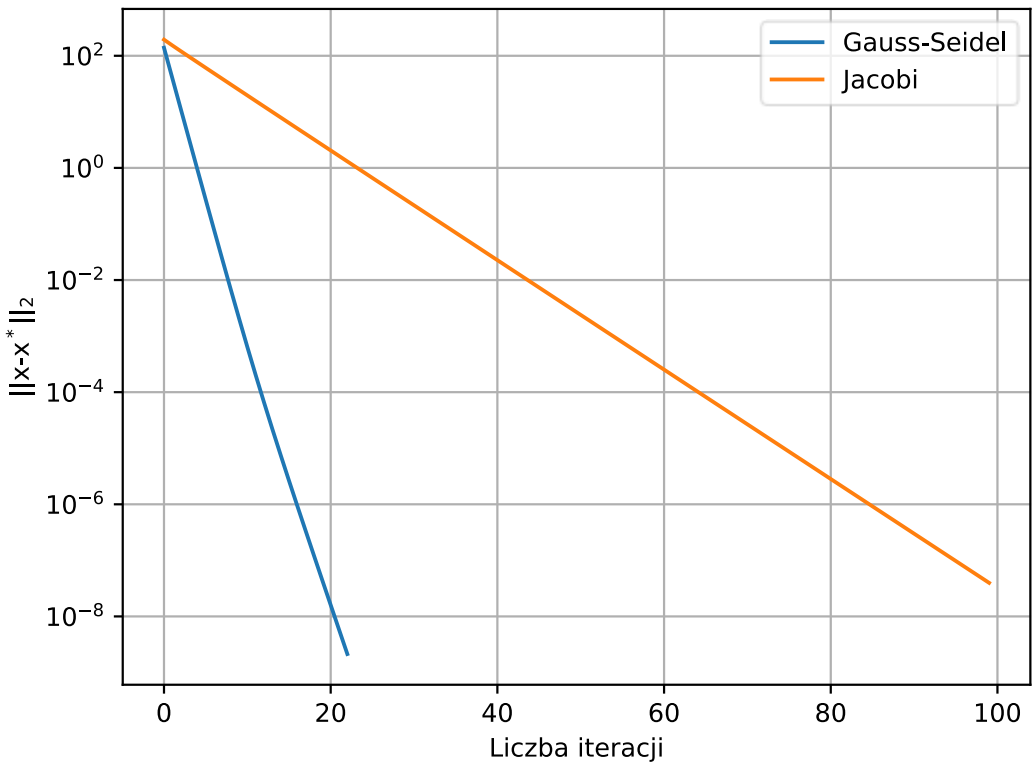
Wynik uzyskany za pomocą iteracyjnej metody Jacobiego(dla 100 iteracji):

[0.1712600924486809 0.3752397373720857 0.5548999252518895
0.7406038487697334 0.926023094904339 1.1110874261312387
1.2962972714979 1.4814829210593163 1.6666660894719791
1.8518519449156514 2.037037047321674 2.222222210999437
2.4074074098765403 2.5925925918376547 2.7777777770671475
2.9629629624257294 3.1481481474933837 3.333333332653043
3.5185185178023892 3.703703702948405 3.888888888096344
4.074074073243847 4.259259258391344 4.444444443538866
4.629629628686367 4.814814813833865 4.999999998981356
5.185185184128843 5.370370369276325 5.555555554423804
5.7407407395712795 5.9259259247187535 6.111111109866225
6.296296295013693 6.481481480161161 6.666666665308628
6.851851850456097 7.03703703560356 7.222222220751026
7.407407405898488 7.592592591045952 7.7777777761934175
7.962962961340881 8.148148146488348 8.333333331635815
8.518518516783287 8.703703701930758 8.888888887078235
9.074074072225718 9.259259257373207 9.444444442520712
9.629629627668235 9.814814812815781 9.99999999796336
10.185185183110983 10.370370368258667 10.555555553406434
10.740740738554303 10.92592592370231 11.11111110885051
11.296296293998935 11.48148147914767 11.666666664296791
11.851851849446405 12.03703703459663 12.222222219747627
12.40740740489957 12.592592590052675 12.777777775207197
12.962962960363422 13.148148145521695 13.3333333306824
13.518518515845967 13.70370370101287 13.888888886183759
14.074074071358794 14.259259256539158 14.444444441728498
14.62962962690633 14.814814812132285 14.99999999741493
15.185185181935305 15.370370369401817 15.55555555347403
15.74074071433329 15.92592602847453 16.11111093871348
16.296295688995723 16.481486554713413 16.66665110915738
16.851856692452785 17.037221383706378 17.221300543657538
17.40924190808366 17.596318651335476 17.736053982177094
18.107440201878173 18.031154065070485 16.956038061348
26.479243708113245]

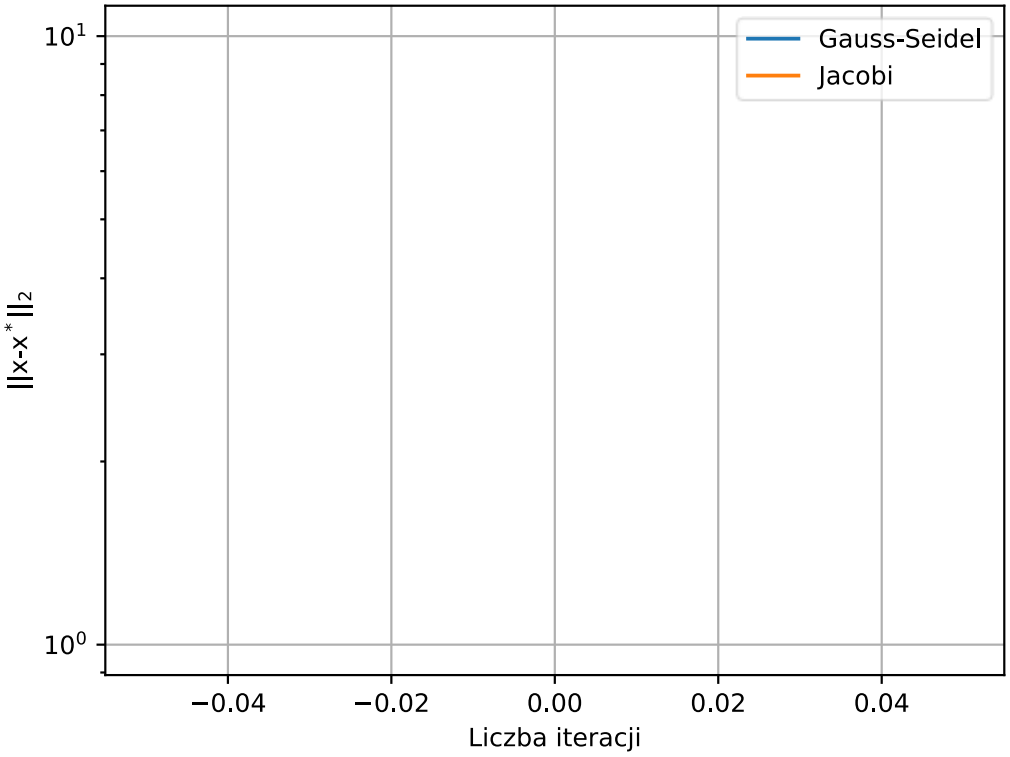
Wynik uzyskany za pomocą iteracyjnej metody Gaussa-Seidela(dla 100 iteracji):

[0.1712600924915493 0.3752397374515706 0.5548999253689074
 0.7406038489241576 0.9260230950961977 1.1110874263605375
 1.2962972717646517 1.481482921363533   1.6666660898136743
 1.8518519452948397 2.0370370477383695 2.222222211453657
 2.407407410368299   2.5925925923669677 2.777777777634029
 2.9629629630301952 3.148148148135448   3.3333333333327193
 3.518518518519688   3.7037037037033413 3.8888888888889284
 4.074074074074087   4.2592592592592515 4.444444444444446
 4.62962962962963   4.814814814814814   4.999999999999999
 5.185185185185186   5.37037037037037   5.5555555555555545
 5.740740740740741   5.9259259259259265 6.11111111111111
 6.296296296296298   6.481481481481481   6.666666666666667
 6.85185185185185   7.037037037037037   7.222222222222222
 7.407407407407407   7.592592592592594   7.777777777777778
 7.962962962962963   8.148148148148147   8.333333333333334
 8.518518518518519   8.703703703703704   8.888888888888891
 9.074074074074073   9.259259259259261   9.444444444444445
 9.629629629629628   9.814814814814817 10.000000000000002
10.185185185185187 10.370370370370368 10.555555555555557
10.74074074074074   10.925925925925924 11.111111111111112
11.296296296296296 11.481481481481481 11.666666666666666
11.851851851851853 12.037037037037036 12.222222222222221
12.407407407407407 12.592592592592593 12.777777777777777
12.962962962962964 13.148148148148147 13.333333333333334
13.518518518518517 13.70370370370369   13.888888888888957
14.074074074073911 14.259259259259082 14.444444444447475
14.629629629617876 14.814814814829212 15.0000000000893
15.185185184578444 15.370370372004347 15.555555556025894
15.740740716823792 15.925926030892441 16.111110941047105
16.296295691233045 16.481486556842206 16.666651111165365
16.85185669432779   17.037221385436535 17.221300545231433
17.409241909490536 17.59631865256542   17.736053983221193
18.10744020272868   18.031154065721086 16.956038061792967
26.47924370835427]

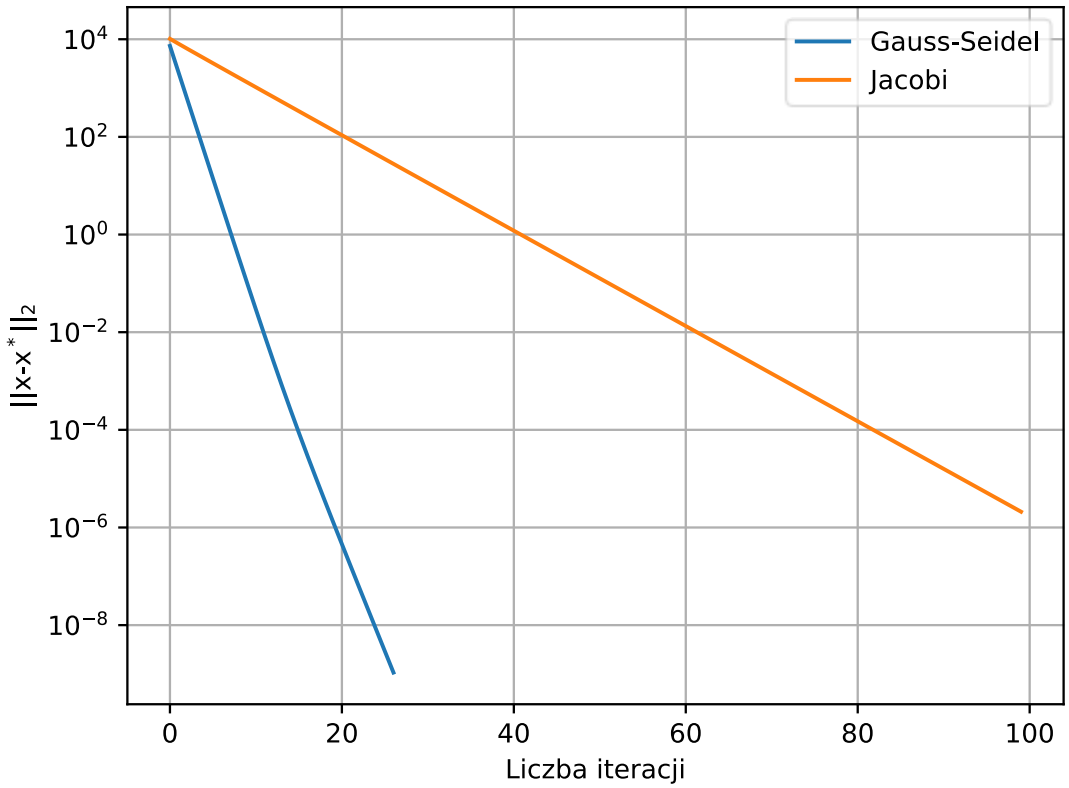
Wykres dla x składającego się z punktów zerowych(x = [0, 0, ..., 0])



Wykres dla x będącego dokładnym rozwiązaniem:



Wykres dla x = [10, 20, 30, …, 1000]:



Powyższe wykresy wyraźnie pokazują, że metoda Gaussa-Seidela jest około dwa razy efektywniejsza niż metoda Jacobiego. Używając metody Gaussa-Seidela zbieżność do oczekiwanego wyniku otrzymujemy przy o wiele mniejszej liczbie iteracji niż używając metody Jacobiego. Gdy za punkty startowe weźmiemy wektor z dokładnym rozwiązaniem, wtedy algorytm kończy się przy pierwszej iteracji, ponieważ błąd jest mniejszy niż podany w funkcji epsilon(10e-10).
W pozostałych dwóch grafikach, wykres dla obu metod stopniowo maleje z każdą iteracją, czyli rozwiązanie z każdą kolejną iteracją co raz bardziej zbiega do oczekiwanego rozwiązania, aż zostanie osiągnięty limit iteracji lub błąd będzie mniejszy od epsilon.