

# Metoda Sukcesywnej Nadrelaksacji (SOR)

## Implementacja praktyczna

### Wstęp

Poniższa implementacja praktyczna stanowi uzupełnienie teoretycznych rozważań na temat metod relaksacji i sukcesywnej nadrelaksacji dla rozwiązywania układów równań liniowych. Wykorzystuje ona kod omówiony dokładnie w III sprawozdaniu z Metod Numerycznych (metody numerycznego rozwiązywania układów równań liniowych). Dla uproszczenia w tym sprawozdaniu termin *metoda SOR (sukcesywnych nadrelaksacji)* użyłem zgodnie z potocznym znaczeniem jako nazwę samej metody, choć bardziej precyzyjna nazwa to *metoda relaksacji* (SOR jest szczególnym jej przypadkiem).

### Przydatne narzędzia

Poniżej umieściłem garść funkcji potrzebnych do implementacji metody Gaussa-Seidla i metody SOR. Wszystkie te funkcje zostały dokładnie omówione w trzecim sprawozdaniu.

$$\text{blad\_rozwiązania}(A, b, x) := A^{-1}b - x$$

Łączenie i rozdzielanie macierzy:

$$\text{sklej\_macierze\_poziomo}(\text{lhs}, \text{rhs}) := \begin{cases} \text{return } (-1) & \text{if } \text{rows}(\text{lhs}) \neq \text{rows}(\text{rhs}) \\ \text{for } i \in 0.. \text{cols}(\text{rhs}) - 1 \\ \quad \text{lhs}^{\langle \text{cols}(\text{lhs})+i \rangle} \leftarrow \text{rhs}^{\langle i \rangle} \\ \text{return lhs} \end{cases}$$

$$\text{wydziel\_macierz\_lewa}(\text{lhs}, \text{leftCols}) := \begin{cases} \text{for } i \in 0.. \text{leftCols} - 1 \\ \quad \text{leftMatrix}^{\langle i \rangle} \leftarrow \text{lhs}^{\langle i \rangle} \\ \text{return leftMatrix} \end{cases}$$

$$\text{wydziel\_macierz\_prawa}(\text{lhs}, \text{leftCols}) := \begin{cases} \text{for } i \in \text{leftCols}.. \text{cols}(\text{lhs}) - 1 \\ \quad \text{rightMatrix}^{\langle i-\text{leftCols} \rangle} \leftarrow \text{lhs}^{\langle i \rangle} \\ \text{return rightMatrix} \end{cases}$$

Wydzielenie macierzy poddiagonalnej, diagonalnej i naddiagonalnej:

$$\text{wydziel\_L}(A) := \begin{cases} \text{for } i \in 0.. \text{rows}(A) - 1 \\ \quad \text{for } j \in 0.. \text{cols}(A) - 1 \\ \quad \quad \begin{cases} L_{i,j} \leftarrow A_{i,j} & \text{if } i > j \\ L_{i,j} \leftarrow 0 & \text{otherwise} \end{cases} \\ \text{return L} \end{cases}$$

```

wydziel_D(A) :=
  for i ∈ 0..rows(A) - 1
    for j ∈ 0..cols(A) - 1
      Di,j ← Ai,j if i = j
    return D
wydziel_U(A) :=
  for i ∈ 0..rows(A) - 1
    for j ∈ 0..cols(A) - 1
      Ui,j ← Ai,j if i < j
      Ui,j ← 0 otherwise
    return U

```

Rodzina metod iteracyjnych - funkcje obliczające rozwiązania :

```
iter_krok(B, c, xs) := B·xs + c
```

```

iter_rozwiaz(B, c, xs, ε, nmax) :=
  xn ← iter_krok(B, c, xs)
  licznik ← 0
  while licznik < nmax ∨ (nmax < 0 ∧ max(|xn - xs|) > ε)
    xs ← xn
    xn ← iter_krok(B, c, xs)
    licznik ← licznik + 1
  return sklej_macierze_poziomo(xn, xn-0 + licznik)

```

Promień spektralny macierzy:

$$\kappa(\arg) := \max(|\text{eigenvals}(\arg)|)$$

## Rozpatrywane przykłady

Aby dokonać wiarogodnego porównania metod Gaussa-Seidla i metody SOR rozpatrzyłem kilka układów równań:

- Znanе ze sprawozdania III:

$$A_1 := \begin{pmatrix} 2 & 5 & 0 \\ 0 & 2 & 2 \\ 1 & 0 & 9 \end{pmatrix} \quad b_1 := \begin{pmatrix} 36 \\ 30 \\ 84 \end{pmatrix} \quad x_1 := A_1^{-1} b_1 = \begin{pmatrix} 3 \\ 6 \\ 9 \end{pmatrix}$$

- Z wykładu J. Sikory (Politechnika Warszawska):

$$A_2 := \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix} \quad b_2 := \begin{pmatrix} 2 \\ 6 \\ 2 \end{pmatrix} \quad x_2 := A_2^{-1} b_2 = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

- Wymyślone przeze mnie dla zabawy:

$$A_3 := \begin{pmatrix} 5 & 5 & 2 \\ 2 & 3 & -4 \\ 0 & 9 & 1 \end{pmatrix} \quad b_3 := \begin{pmatrix} 60 \\ 33 \\ 8 \end{pmatrix} \quad x_3 := A_3^{-1} b_3 = \begin{pmatrix} 11.588 \\ 1.072 \\ -1.652 \end{pmatrix}$$

Ponadto przyjąłem tolerancję:  $\varepsilon := 10^{-6}$

## Rozwiązania

Przed przystąpieniem do obliczeń należy najpierw sprawdzić warunki zbieżności dla używanych metod. Wykorzystałem metodę Gaussa-Seidla oraz metodę SOR dla współczynnika relaksacji równemu 1.1 na wszystkich trzech przykładach.

Rozkład macierzy współczynników na składowe:

$$\begin{aligned} L_1 &:= \text{wydziel\_L}(A_1) & L_2 &:= \text{wydziel\_L}(A_2) & L_3 &:= \text{wydziel\_L}(A_3) \\ D_1 &:= \text{wydziel\_D}(A_1) & D_2 &:= \text{wydziel\_D}(A_2) & D_3 &:= \text{wydziel\_D}(A_3) \\ U_1 &:= \text{wydziel\_U}(A_1) & U_2 &:= \text{wydziel\_U}(A_2) & U_3 &:= \text{wydziel\_U}(A_3) \end{aligned}$$

$$L_1 + D_1 + U_1 - A_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad L_2 + D_2 + U_2 - A_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$L_3 + D_3 + U_3 - A_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{Zgodnie z oczekiwaniami rozkład dokonał się prawidłowo.}$$

Macierz Gaussa-Seidla ma postać:  $M_{gs} = -(D + L)^{-1}U$

$$M_{gs}(L, D, U) := -(D + L)^{-1}U$$

Macierz dla metody SOR ma postać:  $M_{sor} = (D + \omega L)^{-1}(-\omega U + (1-\omega)D)$

$$M_{sor}(L, D, U, \omega) := (D + \omega \cdot L)^{-1}[-\omega \cdot U + (1 - \omega)D]$$

Na podstawie rozważań teoretycznych spodziewamy się, że dla współczynnika relaksacji równego 1 macierz metody SOR będzie równa macierzy metody Gaussa-Seidla. Poniższe sprawdzenia potwierdzają ten wniosek:

$$M_{gs}(L_1, D_1, U_1) - M_{sor}(L_1, D_1, U_1, 1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$M_{gs}(L_2, D_2, U_2) - M_{sor}(L_2, D_2, U_2, 1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$M_{gs}(L_3, D_3, U_3) - M_{sor}(L_3, D_3, U_3, 1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Omawiana rodzina metod iteracyjnych posiada dwa warunki konieczne do prawidłowego

ich działania:

- (*warunek konieczny poprawności*) Macierz współczynników nie może mieć zer na diagonalu
- (*warunek konieczny zbieżności*) Promień spektralny macierzy danej metody musi być mniejszy od 1

Pierwszy warunek jest spełniony dla wszystkich trzech rozpatrywanych układów równań. Poniżej sprawdziłem drugi warunek dla metody Gaussa-Seidla i metody SOR o  $\omega=1.1$ :

$$\begin{aligned} \kappa(\text{Mgs}(L_1, D_1, U_1)) &= 0.527 & \kappa(\text{Msor}(L_1, D_1, U_1, 1.1)) &= 0.632 \\ \kappa(\text{Mgs}(L_2, D_2, U_2)) &= 0.125 & \kappa(\text{Msor}(L_2, D_2, U_2, 1.1)) &= 0.1 \\ \kappa(\text{Mgs}(L_3, D_3, U_3)) &= 13.733 & \kappa(\text{Msor}(L_3, D_3, U_3, 1.1)) &= 17.126 \end{aligned}$$

Pierwsze dwa układy spełniają warunek zbieżności, trzeci natomiast nie. Nie znaczy to jednak, że jest on bezużyteczny. Problem trzeciego układu omówiłem trochę dalej.

Potrzebujemy jeszcze macierzy współczynników c:

$$\begin{aligned} c_{\text{gs}}(L, D, U, b) &:= (D + L)^{-1} b \\ c_{\text{sor}}(L, D, U, \omega, b) &:= (D + \omega L)^{-1} \cdot \omega \cdot b \end{aligned}$$

Rozwiązania pierwszych dwóch układów metodami Gaussa-Seidla i SOR (pierwsze przybliżenie dla wszystkich obliczeń w tym sprawozdaniu to wektor zerowy):

$$\begin{aligned} x_{\text{gs1}} &:= \text{iter\_rozwarz}(\text{Mgs}(L_1, D_1, U_1), c_{\text{gs}}(L_1, D_1, U_1, b_1), b_1 \cdot 0, \epsilon, -1) \\ x_{\text{gs2}} &:= \text{iter\_rozwarz}(\text{Mgs}(L_2, D_2, U_2), c_{\text{gs}}(L_2, D_2, U_2, b_2), b_2 \cdot 0, \epsilon, -1) \\ x_{\text{sor1}} &:= \text{iter\_rozwarz}(\text{Msor}(L_1, D_1, U_1, 1.1), c_{\text{sor}}(L_1, D_1, U_1, 1.1, b_1), b_1 \cdot 0, \epsilon, -1) \\ x_{\text{sor2}} &:= \text{iter\_rozwarz}(\text{Msor}(L_2, D_2, U_2, 1.1), c_{\text{sor}}(L_2, D_2, U_2, 1.1, b_2), b_2 \cdot 0, \epsilon, -1) \end{aligned}$$

$$\begin{aligned} x_{\text{gs1}} &= \begin{pmatrix} 3 & 29 \\ 6 & 29 \\ 9 & 29 \end{pmatrix} & x_{\text{gs2}} &= \begin{pmatrix} 1 & 8 \\ 2 & 8 \\ 1 & 8 \end{pmatrix} \\ x_{\text{sor1}} &= \begin{pmatrix} 3 & 38 \\ 6 & 38 \\ 9 & 38 \end{pmatrix} & x_{\text{sor2}} &= \begin{pmatrix} 1 & 7 \\ 2 & 7 \\ 1 & 7 \end{pmatrix} \end{aligned}$$

Jak widzimy, metoda SOR nie zawsze wykazuje lepszą zbieżność od zwykłej metody Gaussa-Seidla. W przypadku pierwszym metoda Gaussa-Seidla znalazła rozwiązanie szybciej. W drugim przypadku to metoda SOR wykazała się lepszą zbieżnością.

Ciekawym zagadnieniem jest porównanie szybkości zbieżności w zależności od parametru  $\omega$ . Zanim jednak to zanalizujemy, warto wrócić do trzeciego układu równań. Sprawdziłem promień spektralny macierzy SOR dla innego współczynnika relaksacji:

$$\kappa(\text{Msor}(L_3, D_3, U_3, 0.1)) = 0.909$$

Taka wartość pozwala na uzyskanie zbieżności metody SOR. Obliczyłem więc rozwiązanie:

$x_{\text{sor3}} := \text{iter\_rozwi\az}(M_{\text{sor}}(L_3, D_3, U_3, 0.1), c_{\text{sor}}(L_3, D_3, U_3, 0.1, b_3), b_3 \cdot 0, \epsilon, -1)$

$$x_{\text{sor3}} = \begin{pmatrix} 11.588 & 147 \\ 1.072 & 147 \\ -1.652 & 147 \end{pmatrix} \quad x_3 = \begin{pmatrix} 11.588 \\ 1.072 \\ -1.652 \end{pmatrix}$$

**Uzyskaliśmy rozwiązanie dla układu, przy którym metoda Gaussa-Seidla nie była zbieżna!**

Porównajmy więc na koniec zbieżności metody SOR dla wszystkich trzech równań w zależności od współczynnika relaksacji na przedziale (0, 2) (z rozważań teoretycznych wiemy, że w tym przedziale możemy oczekiwać zbieżności metody). Poniższa funkcja wyodrębnia informację o liczbie iteracji:

$\text{sor}(L, D, U, \omega, b, \epsilon) := \begin{cases} \text{wynik} \leftarrow \text{iter\_rozwi\az}(M_{\text{sor}}(L, D, U, \omega), c_{\text{sor}}(L, D, U, \omega, b), b \cdot 0, \epsilon, -1) \\ \text{return wynik}_{0,1} \end{cases}$

Sprawdzenie działania:

$\text{sor}(L_1, D_1, U_1, 1.1, b_1, \epsilon) = 38$        $\text{sor}(L_2, D_2, U_2, 1.1, b_2, \epsilon) = 7$

Badany zakres zbieżności:

$\omega_{\text{range}} := 0.05, 0.1 \dots 1.99$

Poniższy kod analizuje ilość iteracji metody SOR w zależności od współczynnika relaksacji w podanym zakresie:

```
itrs_1 := | licznik ← 0
          | for i ∈ ω_range
          |   (break) if †(Msor(L_1, D_1, U_1, i)) ≥ 1 ∨ licznik > 10000
          |   wynik_licznik,0 ← i
          |   wynik_licznik,1 ← sor(L_1, D_1, U_1, i, b_1, ε)
          |   wynik_licznik,2 ← †(Msor(L_1, D_1, U_1, i))
          |   licznik ← licznik + 1
          | return wynik
```

```
itrs_2 := | licznik ← 0
          | for i ∈ ω_range
          |   (break) if †(Msor(L_2, D_2, U_2, i)) ≥ 1 ∨ licznik > 10000
          |   wynik_licznik,0 ← i
          |   wynik_licznik,1 ← sor(L_2, D_2, U_2, i, b_2, ε)
          |   wynik_licznik,2 ← †(Msor(L_2, D_2, U_2, i))
          |   licznik ← licznik + 1
          | return wynik
```

```

itrs_3 := | licznik ← 0
          | for i ∈ ω_range
          |   (break) if †(Msor(L_3,D_3,U_3,i)) ≥ 1 ∨ licznik > 10000
          |   wyniklicznik,0 ← i
          |   wyniklicznik,1 ← sor(L_3,D_3,U_3,i,b_3,ε)
          |   wyniklicznik,2 ← †(Msor(L_3,D_3,U_3,i))
          |   licznik ← licznik + 1
          | return wynik
    
```

itrs\_1 =

|    | 0    | 1   | 2     |
|----|------|-----|-------|
| 0  | 0.05 | 372 | 0.966 |
| 1  | 0.1  | 213 | 0.932 |
| 2  | 0.15 | 142 | 0.899 |
| 3  | 0.2  | 105 | 0.865 |
| 4  | 0.25 | 83  | 0.831 |
| 5  | 0.3  | 67  | 0.797 |
| 6  | 0.35 | 56  | 0.763 |
| 7  | 0.4  | 48  | 0.729 |
| 8  | 0.45 | 46  | 0.695 |
| 9  | 0.5  | 41  | 0.662 |
| 10 | 0.55 | 36  | 0.629 |
| 11 | 0.6  | 32  | 0.598 |
| 12 | 0.65 | 31  | 0.567 |
| 13 | 0.7  | 28  | 0.539 |
| 14 | 0.75 | 25  | 0.514 |
| 15 | 0.8  | 25  | ...   |

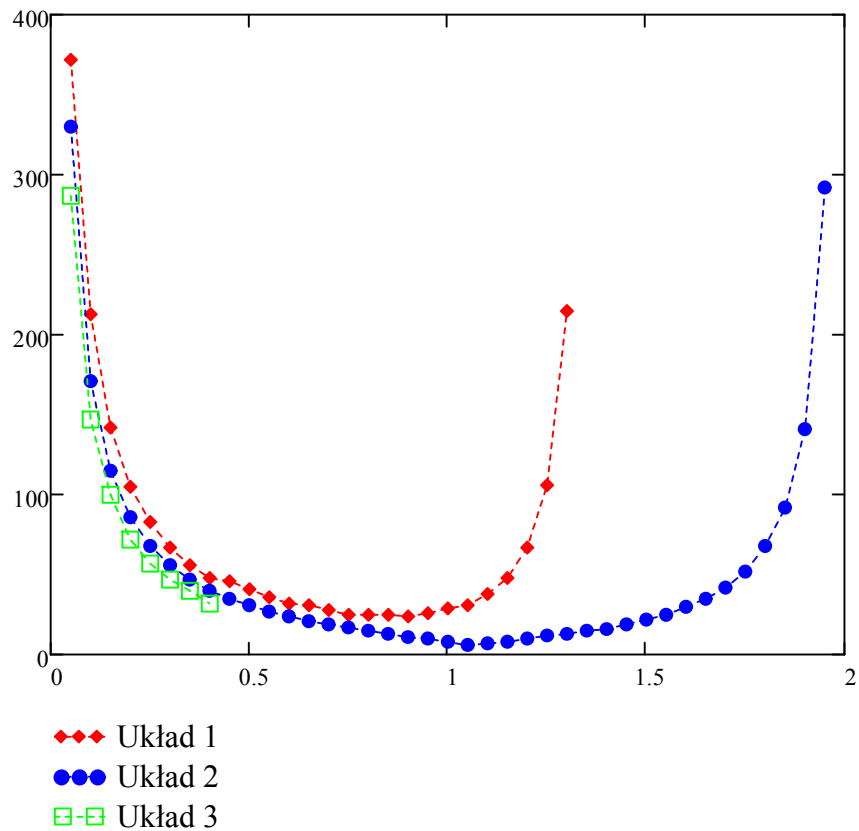
itrs\_2 =

|    | 0    | 1   | 2     |
|----|------|-----|-------|
| 0  | 0.05 | 330 | 0.967 |
| 1  | 0.1  | 171 | 0.934 |
| 2  | 0.15 | 115 | 0.9   |
| 3  | 0.2  | 86  | 0.866 |
| 4  | 0.25 | 68  | 0.831 |
| 5  | 0.3  | 56  | 0.795 |
| 6  | 0.35 | 47  | 0.758 |
| 7  | 0.4  | 40  | 0.72  |
| 8  | 0.45 | 35  | 0.681 |
| 9  | 0.5  | 31  | 0.642 |
| 10 | 0.55 | 27  | 0.601 |
| 11 | 0.6  | 24  | 0.559 |
| 12 | 0.65 | 21  | 0.515 |
| 13 | 0.7  | 19  | 0.47  |
| 14 | 0.75 | 17  | 0.422 |
| 15 | 0.8  | 15  | 0.373 |
| 16 | 0.85 | 13  | 0.32  |
| 17 | 0.9  | 11  | 0.263 |
| 18 | 0.95 | 10  | ...   |

itrs\_3 =

|      |     |       |
|------|-----|-------|
| 0.05 | 287 | 0.955 |
| 0.1  | 147 | 0.909 |
| 0.15 | 100 | 0.863 |
| 0.2  | 72  | 0.817 |
| 0.25 | 57  | 0.77  |
| 0.3  | 47  | 0.722 |
| 0.35 | 40  | 0.674 |
| 0.4  | 32  | 0.625 |

Poniższy wykres przedstawia liczbę iteracji koniecznych do osiągnięcia zadanej dokładności (oś y) w zależności od współczynnika relaksacji ω (oś x). Dalsze opisanie wykresu nie było możliwe z powodu błędów w programie Mathcad.



## Wnioski

Na podstawie przeprowadzonej analizy i powyższego wykresu udało się potwierdzić wnioski i obserwacje wysnute w części teoretycznej.

- Metoda Sukcesywnej Nadrelaksacji (SOR) może być dla danego układu równań szybciej lub wolniej zbieżna do rozwiązania niż podstawowa metoda Gaussa-Seidla; wszystko zależy od doboru współczynnika relaksacji w stosunku do układu równań.
- Nie wszystkie układy równań są zbieżne na całym przedziale dopuszczalnego współczynnika relaksacji (przedział  $(0, 2)$ ).
- Odpowiedni dobór współczynnika relaksacji pozwala nam uzyskać zbieżność dla układów, które nie spełniają warunku zbieżności w podstawowej metodzie Gaussa-Seidla