Zadanie Numeryczne 4 – Kacper Drużdżel

a) Algorytm Gaussa-Seidela
b) Metoda gradientów sprzężonych

a)

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

W metodzie Gaussa-Seidela rozpatrujemy powyższy wzór.

Algorytm ogólny jest następujący:

Wybierz początkowe przybliżenie $\mathbf{x}^{(0)}$
for k := 1 **step** 1 **until** *oczekiwane przybliżenie* **do**
    for i := 1 **step** 1 **until** n **do**
        $\sigma = 0$
        for j := 1 **step** 1 **until** i-1 **do**
            $\sigma = \sigma + a_{ij}x_j^{(k)}$
        end (j-for)
        for j := i+1 **step** 1 **until** n **do**
            $\sigma = \sigma + a_{ij}x_j^{(k-1)}$
        end (j-for)
        $x_i^{(k)} = \frac{(b_i - \sigma)}{a_{ii}}$
    end (i-for)
    sprawdź, czy osiągnięto oczekiwane przybliżenie
end (k-for)
$\mathbf{x} \approx \mathbf{x}^{(k)}$

b) Metoda gradientów sprzężonych

Wykorzystałem algorytm podany na wykładzie

$\mathbf{A} \in \mathbb{R}^{N \times N}$ symetryczna, dodatnio określona, $\mathbf{x}_1$ — początkowe przybliżenie rozwiązania równania (12), $0 < \varepsilon \ll 1$.

$$
\begin{array}{l}
\mathbf{r}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_1, \mathbf{p}_1 = \mathbf{r}_1 \\
\textbf{while } \|\mathbf{r}_k\| > \varepsilon \\
\qquad \begin{aligned}
\alpha_k &= \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{A}\mathbf{p}_k} \\
\mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k \\
\beta_k &= \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k} \\
\mathbf{p}_{k+1} &= \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{p}_k
\end{aligned} \\
\textbf{end}
\end{array}
\tag{13}
$$

Z każdą iteracją wynik staje się dokładniejszy.

Uruchomienie programów:

G++ -o zad4a zad4a.cpp

G++ -o zad4b zad4b.cpp

Następnie:

./zad4a

./zad4b