Odwrotna metoda potęgowa

Macierz trójdiagonalna i rozkład Householdera

Tomasz Pawlaczyk
nr albumu xxxxxx

Wydział MiNI
Politechnika Warszawska

Projekt 2 – zadanie 14

23 stycznia 2026, czwartek

Treść zadania

Odwrotna metoda potęgowa z normowaniem dla macierzy trójdiagonalnej. Celem jest znalezienie wartości własnej macierzy $A \in \mathbb{R}^{n \times n}$ leżącej najbliżej zadanej wartości μ.

Do rozwiązywania układów równań należy użyć rozkładu macierzy $A - \mu I$ wyznaczonego za pomocą dwuwierszowych odbić Householdera.

Zakładamy, że:

- $1 \leq n \leq 200\,000$.
- nie wolno korzystać z funkcji `sparse` oraz `spdiags`.

Zastosowane metody numeryczne

Ogólne działanie metody

Wprowadzenie:

- Celem jest przybliżenie pojedynczej wartości własnej macierzy trójdiagonalnej.
- Szukamy wartości własnej najbliższej zadanej liczbie μ.
- Wykorzystujemy odwrotną metodę potęgową z przesunięciem.

Przesunięcie spektrum:

$$B = A - \mu I$$

Rozkład Householdera macierzy $B = A - \mu I$

Program realizuje rozkład macierzy trójdiagonalnej $B = A - \mu I$ do postaci $QB = R$, gdzie Q jest iloczynem dwuwierszowych odbić Householdera, a R jest macierzą górnotrójkątną o wąskim paśmie.
Rozkład wykonywany jest krok po kroku: w każdym kroku zerowany jest jeden element pod przekątną, a modyfikowany jest jedynie lokalny fragment macierzy, co zapewnia niską złożoność obliczeniową.

Macierz R przechowywana jest pasmowo w postaci trzech wektorów r_0, r_1, r_2 i ma strukturę:

$$R = \begin{bmatrix} r_0(1) & r_1(1) & r_2(1) & 0 & \cdots \\ 0 & r_0(2) & r_1(2) & r_2(2) & \ddots \\ 0 & 0 & r_0(3) & r_1(3) & \ddots \\ \vdots & & \ddots & \ddots & \ddots \end{bmatrix}$$

gdzie $r_0(i)$ oznacza element diagonalny $R_{i,i}$, $r_1(i)$ element $R_{i,i+1}$, a $r_2(i)$ element $R_{i,i+2}$.

Reprezentacja macierzy Q — odbicia Householdera

Macierz ortogonalna Q nie jest przechowywana jawnie. Jest ona iloczynem lokalnych odbić Householdera:

$$Q = H_{n-1} \cdots H_2 H_1.$$

Pojedyncze odbicie ma postać $H_i = I - \beta_i u_i u_i^T$, gdzie wektor Householdera u_i ma tylko dwie niezerowe składowe:

$$u_i = \begin{bmatrix} 0 \\ \vdots \\ v_1^{(i)} \\ v_2^{(i)} \\ 0 \\ \vdots \end{bmatrix}, \qquad \beta_i = \frac{2}{u_i^T u_i}.$$

W implementacji przechowywane są wyłącznie wartości $v_1^{(i)}$, $v_2^{(i)}$ oraz współczynnik β_i, bez jawnego tworzenia macierzy Q.

Konstrukcja wektora Householdera (2D)

W każdym kroku rozkładu rozważany jest lokalny wektor z kolumny i:

$$x = \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r_0(i) \\ \mathrm{pod}(i) \end{bmatrix}.$$

Celem jest wyzerowanie drugiej składowej za pomocą odbicia Householdera, tak aby

$$Hx = \begin{bmatrix} \pm\sqrt{a^2 + b^2} \\ 0 \end{bmatrix}.$$

Wektor Householdera definiowany jest jako

$$u = \begin{bmatrix} v_1^{(i)} \\ v_2^{(i)} \end{bmatrix} = \begin{bmatrix} a + \mathrm{sign}(a)\sqrt{a^2 + b^2} \\ b \end{bmatrix}.$$

Taka konstrukcja zapewnia stabilność numeryczną oraz lokalność operacji.

Lokalny fragment macierzy i aktualizacja

W kroku i przetwarzany jest lokalny fragment macierzy obejmujący kolumny $i, i+1, i+2$:

$$\begin{bmatrix} r_0(i) & r_1(i) & 0 \\ \mathrm{pod}(i) & r_0(i+1) & r_1(i+1) \end{bmatrix}.$$

Wykonywane są rzuty na wektor Householdera u_i:

$$s_1 = u_i^T \begin{bmatrix} r_0(i) \\ \mathrm{pod}(i) \end{bmatrix}, \quad s_2 = u_i^T \begin{bmatrix} r_1(i) \\ r_0(i+1) \end{bmatrix}, \quad s_3 = u_i^T \begin{bmatrix} 0 \\ r_1(i+1) \end{bmatrix}.$$

Efekt działania:

- s_1 zeruje element pod przekątną,
- s_2 aktualizuje $r_1(i)$ oraz $r_0(i+1)$,
- s_3 tworzy $r_2(i)$ i modyfikuje $r_1(i+1)$.

Zastosowanie macierzy Q w `householder_uklad`

Macierz ortogonalna Q nie jest tworzona jawnie. Jest zapisana pośrednio jako iloczyn odbić Householdera:

$$Q = H_{n-1} \cdots H_1, \qquad H_i = I - \beta_i u_i u_i^T,$$

gdzie wektor odbicia ma tylko dwie niezerowe składowe

$$u_i = \begin{bmatrix} v_1^{(i)} \\ v_2^{(i)} \end{bmatrix}.$$

Działanie pojedynczego odbicia (lokalne):

$$t = \beta_i \left(v_1^{(i)} y_i + v_2^{(i)} y_{i+1} \right),$$

$$y_i \leftarrow y_i - v_1^{(i)} t, \qquad y_{i+1} \leftarrow y_{i+1} - v_2^{(i)} t.$$

Modyfikowane są wyłącznie pozycje i oraz $i+1$ wektora.
Efekt pętli: po wykonaniu wszystkich odbić otrzymujemy

$$y = Q^* b,$$

bez jawnego konstruowania macierzy Q.

Podstawienie wsteczne

Wejście: macierz górnotrójkątna R zapisana pasmowo jako wektory r_0, r_1, r_2 oraz wektor $y = Qb$.
Wyjście: wektor x spełniający układ równań $Rx = y$.

Algorytm rozwiązuje układ z macierzą górnotrójkątną, zaczynając od ostatniej niewiadomej i przechodząc w górę. Każdy element x_i obliczany jest z już znanych wartości x_{i+1} oraz x_{i+2}. Dzięki strukturze pasmowej w każdym kroku modyfikowane są tylko 2–3 elementy.

Bezpośrednie wzory:

$$x_n = \frac{y_n}{r_0(n)}, \qquad x_{n-1} = \frac{y_{n-1} - r_1(n-1)x_n}{r_0(n-1)},$$

$$x_i = \frac{y_i - r_1(i)x_{i+1} - r_2(i)x_{i+2}}{r_0(i)}.$$

Złożoność obliczeniowa algorytmu wynosi $O(n)$, przy niskim zużyciu pamięci.

Użyty estymator błędu zbieżności

W programie stosowany jest estymator postaci

$$\left\| (\tilde{x}_i^{(k+1)})^{-1} \, |\tilde{x}_i^{(k+1)}| \, \tilde{x}^{(k+1)} - (\tilde{x}_i^{(k)})^{-1} \, |\tilde{x}_i^{(k)}| \, \tilde{x}^{(k)} \right\|.$$

Gdzie:

- $\tilde{x}^{(k)}, \tilde{x}^{(k+1)}$ — kolejne przybliżenia wektora własnego,
- i — indeks składowej o największej wartości bezwzględnej w $\tilde{x}^{(k)}$,
- $\tilde{x}_i^{(k)}$ — i-ta składowa wektora $\tilde{x}^{(k)}$.

Estymator eliminuje potencjalne zmiany znaku wektora własnego pomiędzy iteracjami i mierzy rzeczywistą zmianę jego kierunku.

Testy częściowe

Test częściowy 1 — pojedynczy krok Householdera

Cel testu: weryfikacja pojedynczego kroku Householdera

Macierz wejściowa B:

$$B = \begin{bmatrix} 4 & 3 & 0 & 0 & 0 \\ -1 & 3 & 1 & 0 & 0 \\ 0 & 2 & 5 & -1 & 0 \\ 0 & 0 & -2 & 6 & 2 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

Macierz po jednym kroku Householdera $H_1 B$:

$$B_1 = \begin{bmatrix} -4.1231 & -2.1828 & 0.2425 & 0 & 0 \\ -0.0000 & 3.6380 & 0.9701 & 0 & 0 \\ 0 & 2 & 5 & -1 & 0 \\ 0 & 0 & -2 & 6 & 2 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

Sprawdzenie kluczowe:

$$B(2,1) = -1.00, \qquad B_1(2,1) = -2.22 \cdot 10^{-16}.$$

Wniosek: element $(2,1)$ został wyzerowany numerycznie — test poprawny.

Test częściowy 2 — rozkład Householdera (QR)

Cel testu: sprawdzenie poprawności rozkładu oraz ortogonalności macierzy Q.

Macierz wejściowa B:

$$\begin{bmatrix} 4 & 3 & 0 & 0 & 0 \\ -1 & 3 & 1 & 0 & 0 \\ 0 & 2 & 5 & -1 & 0 \\ 0 & 0 & -2 & 6 & 2 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

Macierz R po rozkładzie:

$$\begin{bmatrix} -4.1231 & -2.1828 & 0.2425 & 0 & 0 \\ 0 & -4.1515 & -3.2589 & 0.4817 & 0 \\ 0 & 0 & -4.3955 & 3.5104 & 0.9100 \\ 0 & 0 & 0 & -5.0443 & -2.1421 \\ 0 & 0 & 0 & 0 & 1.6072 \end{bmatrix}$$

Wyniki:

$$\|Q^T Q - I\| = 2.34 \cdot 10^{-16}, \quad \|QR - B\| = 1.62 \cdot 10^{-15}, \quad \|R_{\text{dol}}\| = 0.$$

Wniosek: rozkład poprawny — Q ortogonalna, R górnotrójkątna.

Test częściowy 3 — podstawienie wsteczne

Cel testu: sprawdzenie poprawności rozwiązania układu $Rx = y$ metodą podstawienia wstecznego.

Dwa sposoby rozwiązania:

- rozwiązanie dokładne x_{exact} obliczone ręcznie,
- rozwiązanie numeryczne x_{num} wyznaczone przez program.

Wyniki:

$$x_{\text{exact}} = (1, -1, 2, 0, -1), \qquad x_{\text{num}} = (1, -1, 2, 0, -1).$$

$$\|x_{\text{num}} - x_{\text{exact}}\| = 0.$$

Wniosek: podstawienie wsteczne działa poprawnie — program dokładnie odtwarza rozwiązanie układu trójpasmowego, bez błędu numerycznego.

Test częściowy 4 — odbicia Householdera

Cel testu: weryfikacja poprawności zastosowania odbicia Householdera bez jawnego tworzenia macierzy Q.

Dwa sposoby obliczeń:

- wektorowo — lokalna aktualizacja dwóch składowych wektora,
- macierzowo — jawne obliczenie Hb.

Wyniki:

$$y_{\text{wektor}} = (-2, 1, 3, 4, 5), \qquad y_{\text{macierz}} = (-2, 1, 3, 4, 5).$$

$$\|y - y_{\text{exact}}\| = 0.$$

Wniosek: lokalne odbicia Householdera działają identycznie jak jawna macierz, co potwierdza poprawność i efektywność implementacji.

Testy poprawności

Test 1 — macierz Toeplitza (Laplace 1D)

Cel testu: weryfikacja poprawności obliczeń poprzez porównanie z wartością własną znaną ze wzoru analitycznego.

Dla macierzy Laplace'a 1D zastosowano wzór

$$\lambda_k = 2 - 2\cos\left(\frac{k\pi}{n+1}\right),$$

przy $n = 100$, $k = 25$ oraz przesunięciu μ bliskim λ_k.

Wyniki:

$$\lambda_{\text{dokl}} = 0.574832071704986,$$

$$\lambda_{\text{num}} = 0.574832071704986,$$

$$|\lambda_{\text{dokl}} - \lambda_{\text{num}}| = 1.11 \cdot 10^{-16}, \qquad \|Av - \lambda_{\text{num}} v\| = 3.59 \cdot 10^{-16}.$$

Wniosek: zgodność z rozwiązaniem analitycznym do precyzji maszynowej potwierdza poprawność implementacji metody.

Test 2 — macierz diagonalna z małym sprzężeniem

Cel testu: sprawdzenie stabilności rozwiązywania układów liniowych oraz zachowania odwrotnej metody potęgowej dla macierzy bliskiej diagonalnej.

Macierz ma postać diagonalną z niewielkimi elementami pozadiagonalnymi rzędu $\varepsilon = 10^{-6}$. Szukana jest wartość własna najbliższa przesunięciu μ.

Wyniki:

$$\lambda_{\text{dokl}} = 37.0, \qquad \lambda_{\text{num}} = 37.0,$$

$$|\lambda_{\text{dokl}} - \lambda_{\text{num}}| = 0, \qquad \|Av - \lambda_{\text{num}} v\| = 3.57 \cdot 10^{-17}.$$

Wniosek: metoda zachowuje pełną dokładność numeryczną i pozostaje stabilna nawet przy bardzo słabym sprzężeniu.

Test 3 — Laplace 1D, $n = 200\,000$

Cel testu: weryfikacja poprawności i wydajności algorytmu dla maksymalnego dopuszczalnego rozmiaru macierzy n = 200 000.

Test wykonano dla macierzy Laplace'a 1D, z przesunięciem μ bliskim wartości własnej danej wzorem analitycznym.

Wyniki:

$$\lambda_{\text{wzór}} = 5.8578088406190432 \cdot 10^{-1},$$

$$\lambda_{\text{num}} = 5.8578088406190454 \cdot 10^{-1},$$

$$\text{błąd} = 2.22 \cdot 10^{-16}, \qquad \text{residuum} = 2.76 \cdot 10^{-16}.$$

Zgodność kierunku wektora własnego:

$$\frac{|x^T v_{\text{analytic}}|}{\|x\| \, \|v_{\text{analytic}}\|} = 0.9999999999999997.$$

Wniosek: algorytm zachowuje dokładność maszynową, poprawny wektor własny nawet dla $n = 200\,000$.

Test 4 — macierz diagonalna, wartości ujemne

Cel testu: sprawdzenie, czy metoda wybiera wartość własną najbliższą zadanemu przesunięciu w sytuacji, gdy wartość własna jest ujemna.

Wyniki:

$$\lambda_{\text{wzór}} = -3, \qquad \mu = -2.8, \qquad \lambda_{\text{num}} = -3,$$

$$\text{błąd} = 0, \qquad \text{residuum} = 1.40 \cdot 10^{-15}.$$

Wniosek: odwrotna metoda potęgowa poprawnie wybiera wartość własną najbliższą μ.

Test 5 — macierz rzędu ε (dokładność)

Cel testu: weryfikacja poprawności odwrotnej metody potęgowej dla macierzy o bardzo małych elementach $O(\varepsilon)$.

Skala elementów: $\varepsilon = 10^{-6}$, przesunięcie μ dobrane w pobliżu jednej z wartości własnych.

Porównanie z `eig()`:

$$\lambda_{\text{eig}} = 2.7907478172376863 \cdot 10^{-6},$$

$$\lambda_{\text{num}} = 2.7907478172376880 \cdot 10^{-6},$$

$$\text{różnica} = 1.69 \cdot 10^{-21}.$$

Poprawność:

$$\|Av - \lambda v\| = 3.33 \cdot 10^{-21}, \qquad \text{błąd względny} = 6.07 \cdot 10^{-16}.$$

Test 5 — macierz rzędu ε

Cel testu: weryfikacja poprawności odwrotnej metody potęgowej dla macierzy trójdiagonalnej o elementach rzędu precyzji maszynowej.

Skala elementów: $O(\varepsilon) = 2.22 \cdot 10^{-16}$, przesunięcie $\mu = 5.247728 \cdot 10^{-16}$.

Wyniki numeryczne:

$$\lambda_{\text{eig}} = 6.1967049652393558 \cdot 10^{-16},$$

$$\lambda_{\text{num}} = 6.1967049652393587 \cdot 10^{-16}, \qquad \text{różnica} = 2.96 \cdot 10^{-31}.$$

Sprawdzenie poprawności:

$$\|Av - \lambda v\| = 8.22 \cdot 10^{-31}, \quad \text{błąd względny} = 4.77 \cdot 10^{-16},$$

uwarunkowanie macierzy: $\operatorname{condest}(A - \mu I) = 7.73 \cdot 10^{1}$

Skalowanie względne:

$$\lambda/\varepsilon = 2.790748, \qquad \text{residuum}/\varepsilon = 3.70 \cdot 10^{-15}.$$

Test 5 — stabilność (różne wektory startowe)

Cel testu: sprawdzenie niezależności wyniku od wyboru wektora startowego.

λ	residuum
$6.1967 \cdot 10^{-16}$	$8.2240 \cdot 10^{-31}$
$6.1967 \cdot 10^{-16}$	$7.5650 \cdot 10^{-31}$
$6.1967 \cdot 10^{-16}$	$8.2608 \cdot 10^{-31}$
$6.1967 \cdot 10^{-16}$	$8.2608 \cdot 10^{-31}$
$6.1967 \cdot 10^{-16}$	$8.9211 \cdot 10^{-31}$

Wniosek: wynik jest stabilny numerycznie i niezależny od wektora startowego, nawet dla macierzy o elementach rzędu precyzji maszynowej.

Test 6 — zespolone wartości własne

Cel testu: sprawdzenie działania odwrotnej metody potęgowej dla macierzy niesymetrycznej o zespolonym widmie.

Wybrano macierz trójdiagonalną spełniającą warunek $a \cdot c < 0$, co gwarantuje wystąpienie zespolonych wartości własnych. Zastosowano zespolone przesunięcie μ bliskie jednej z nich.

Wyniki:

$$\lambda_{\text{eig}} = 1.000000 + 4.756624\,i, \qquad \lambda_{\text{num}} = 1.000000 + 4.756624\,i$$

$$\text{błąd względny} = 3.72 \cdot 10^{-16}, \quad \text{estymator błędu} = 4.80 \cdot 10^{-15}$$

$$\|Av - \lambda v\| = 1.84 \cdot 10^{-15}.$$

Test 6 — stabilność dla zespolnych wartości własnych

Cel testu: sprawdzenie stabilności metody względem różnych wektorów startowych.

λ_{num}
$1.000000 + 4.756624\,i$
$1.000000 + 4.756624\,i$
$1.000000 + 4.756624\,i$
$1.000000 + 4.756624\,i$
$1.000000 + 4.756624\,i$

Wniosek: algorytm poprawnie obsługuje zespolone wartości własne i daje stabilny wynik niezależnie od wektora początkowego.

Testy numeryczne

Numeryczny test 1 — wpływ przesunięcia μ

Cel testu: zbadanie wpływu przesunięcia $\mu = \lambda + \delta$ na zbieżność odwrotnej metody potęgowej i wybór właściwej wartości własnej.

Oczekiwana wartość własna:

$$\lambda_{50} = 0.4774106137879128.$$

δ	μ	$\lambda_{\text{znaleziona}}$	indeks
10^{-1}	0.57741	0.58802	55
10^{-2}	0.48741	0.49661	51
10^{-3}	0.47841	0.47741	50

Wniosek: im mniejsze δ, tym metoda wybiera wartość własną bliższą oczekiwanej.

Numeryczny test 2 — zbieżność i czas

Cel testu: zbadanie wpływu przesunięcia μ na czas zbieżności odwrotnej metody potęgowej.

Do testu użyto **dużej, wcześniej wygenerowanej** macierzy trójdiagonalnej o rozmiarze

$$n = 1000,$$

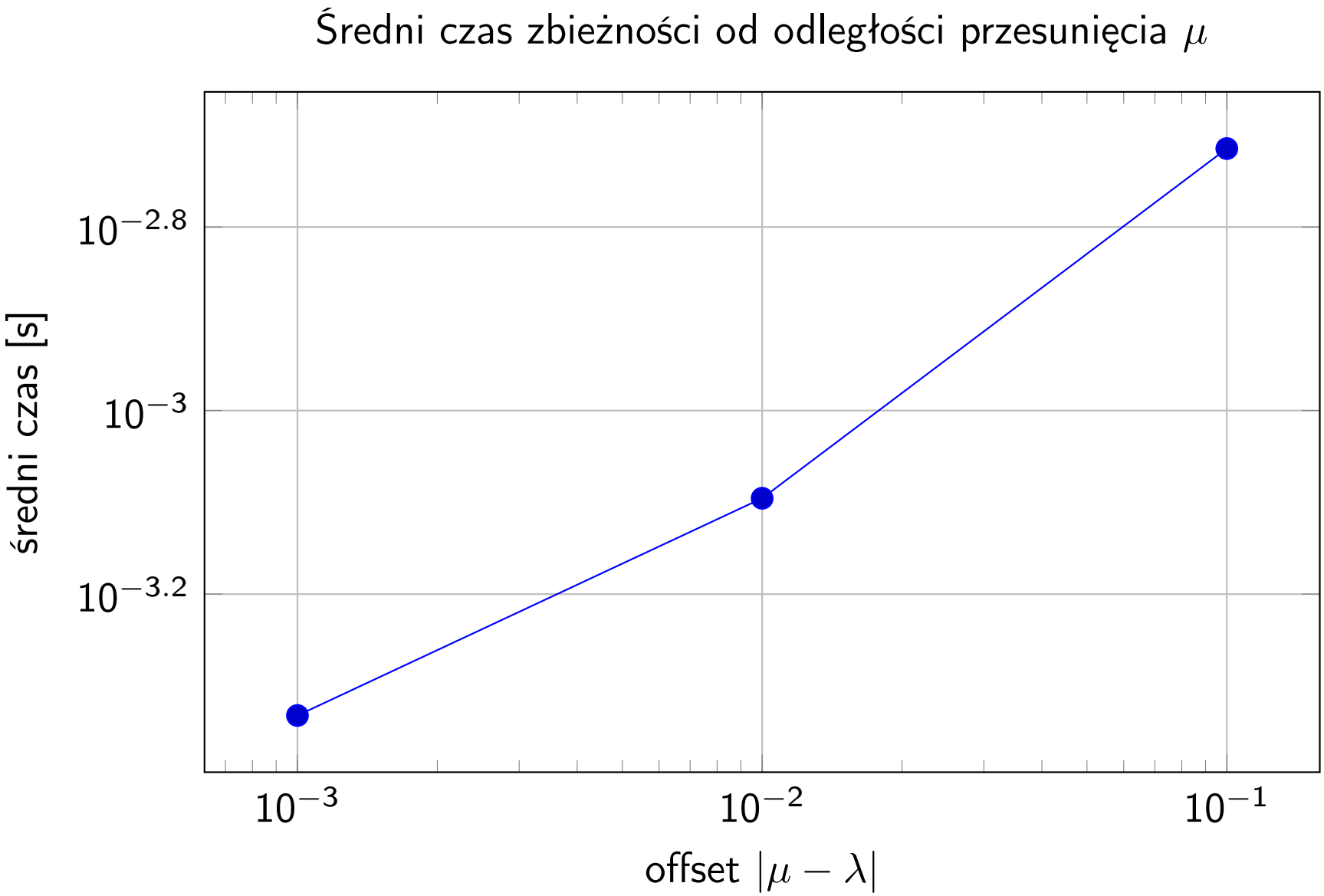
z jedną wyraźnie odizolowaną wartością własną

$$\lambda = 22.024275374571 9639.$$

Wyniki średnie (10 powtórzeń):

offset $\lvert\mu - \lambda\rvert$	$\lambda_{\text{śr}}$	czas [s]
10^{-1}	22.024	$1.93 \cdot 10^{-3}$
10^{-2}	22.024	$8.03 \cdot 10^{-4}$
10^{-3}	22.024	$4.65 \cdot 10^{-4}$

Numeryczny test 2 — wpływ przesunięcia μ na czas



Numeryczny test 3 — przypadek graniczny

Cel testu: sprawdzenie zachowania odwrotnej metody potęgowej dla poprawnego i niepoprawnego przesunięcia μ w przypadku macierzy diagonalnej.

Rozważana macierz ma wartości własne

$$10,\ 20,\ \ldots,\ 120,\ 130,\ldots$$

a szukana wartość własna wynosi $\lambda = 120$.

Wyniki:

- $\mu = 122$ (blisko λ):

$$\lambda_{\text{num}} = 120, \quad \text{błąd} = 0, \quad \text{residuum} = 8.49 \cdot 10^{-15}.$$

- $\mu = 125$ (środek między 120 i 130):

$$\lambda_{\text{num}} \approx 129.67, \quad \text{błąd} \approx 9.67, \quad \text{residuum} \approx 1.79.$$

Wniosek: metoda działa poprawnie tylko wtedy, gdy przesunięcie μ jednoznacznie wskazuje najbliższą wartość własną — zgodnie z teorią.

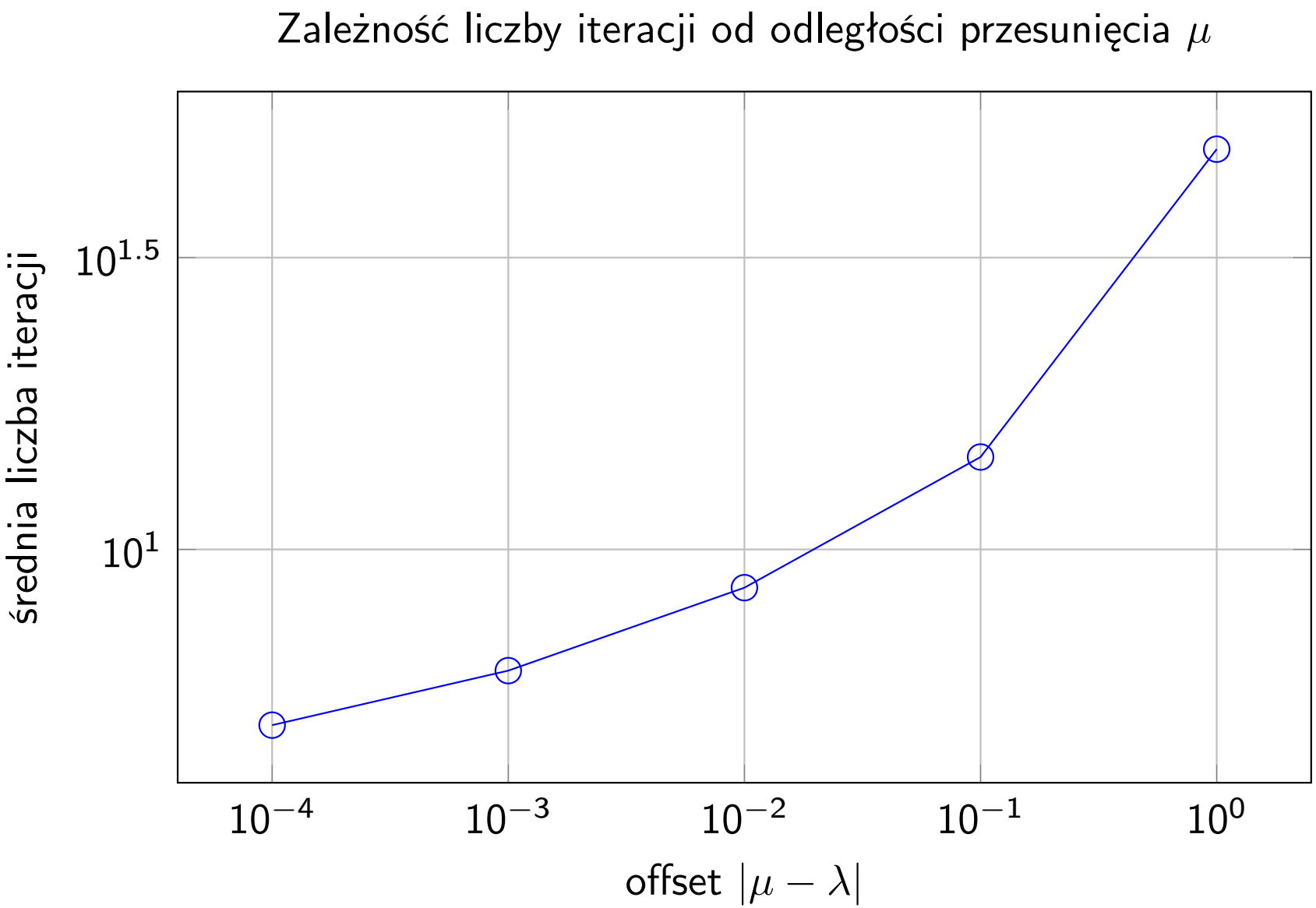
Numeryczny test 4 — liczba iteracji

Cel testu: zbadanie wpływu odległości przesunięcia $|\mu - \lambda|$ na liczbę iteracji odwrotnej metody potęgowej.

Test wykonano dla dużej, wcześniej wygenerowanej macierzy trójdiagonalnej ($n = 1000$) z odizolowaną wartością własną $\lambda = 22.024275374571963 9$.

offset $\lvert\mu - \lambda\rvert$	$\lambda_{\text{śr}}$	iteracje (śr.)
1	22.024	48.5
10^{-1}	22.024	14.4
10^{-2}	22.024	8.6
10^{-3}	22.024	6.2
10^{-4}	22.024	5.0

Numeryczny test 4 — wykres zbieżności



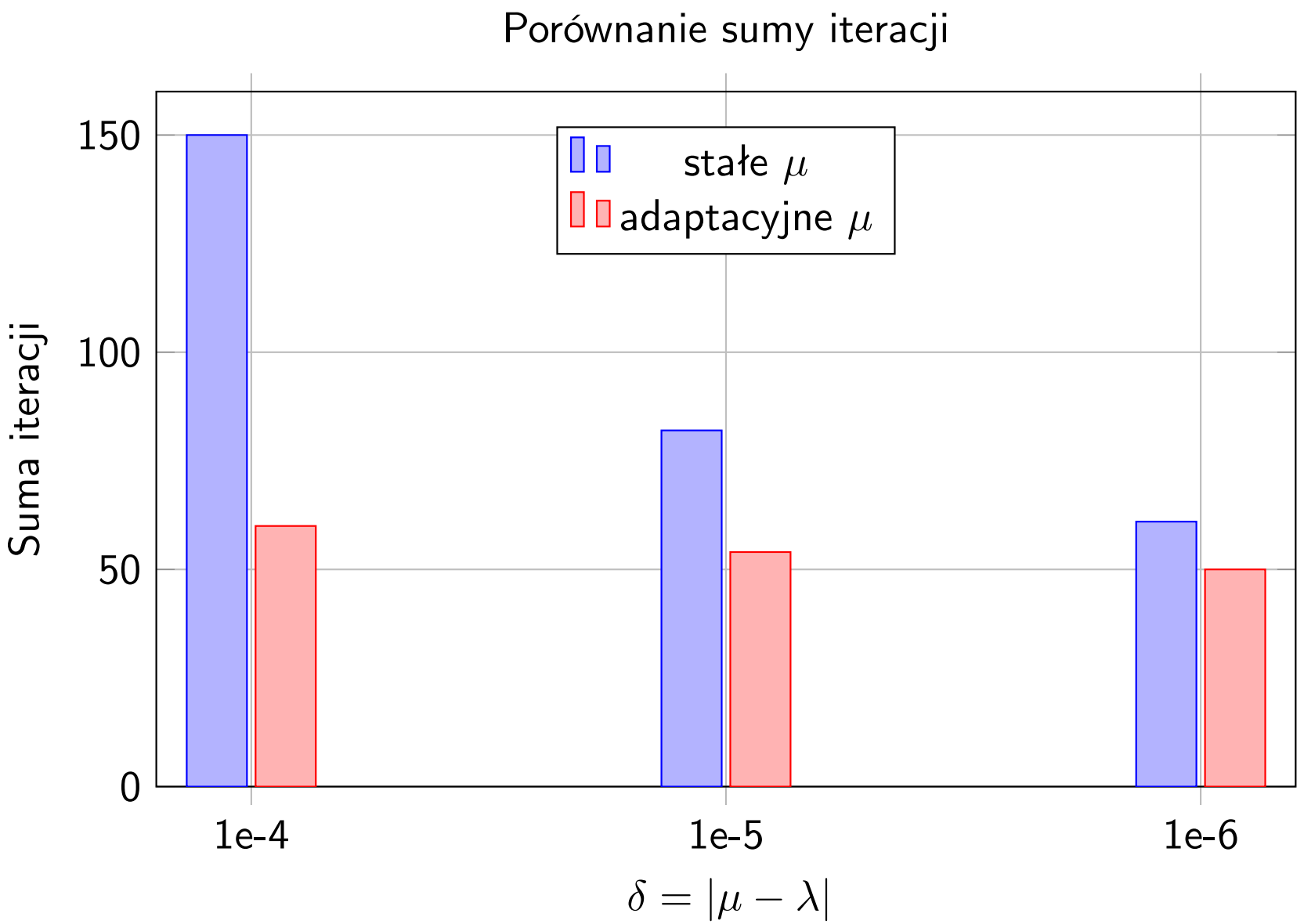
Numeryczny test 5 — porównanie metod

Cel testu: porównanie klasycznej odwrotnej metody potęgowej ze stałym przesunięciem μ oraz wariantu z iteracyjną aktualizacją parametru μ.

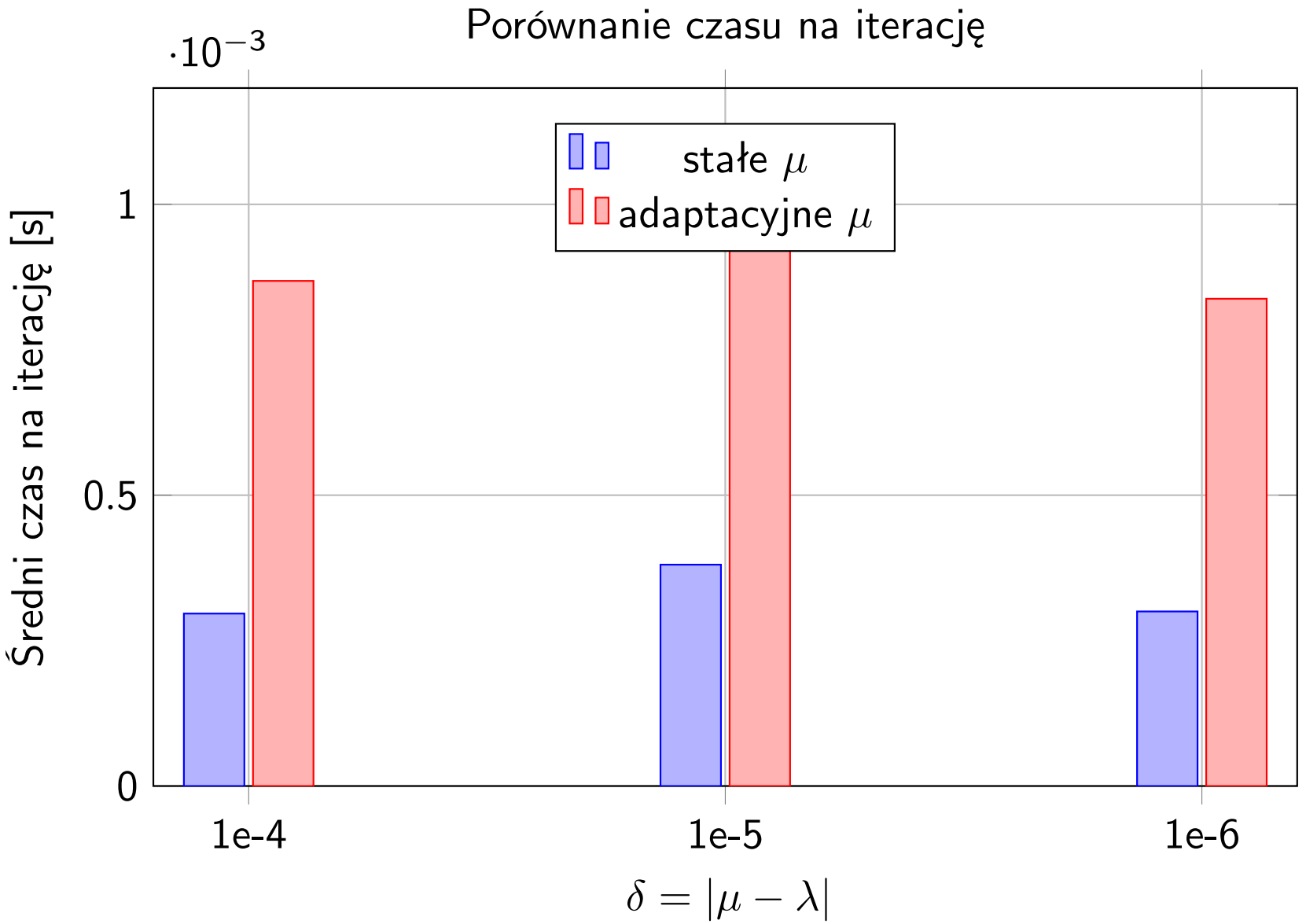
Test wykonano dla macierzy Laplace'a 1D o rozmiarze $n = 5000$, dla kilku wartości $\delta = |\mu - \lambda|$.

δ	iteracje (stałe μ)	czas/iter. [s]	iteracje (adapt. μ)	czas/iter. [s]
10^{-4}	150	$2.97 \cdot 10^{-4}$	60	$8.68 \cdot 10^{-4}$
10^{-5}	82	$3.81 \cdot 10^{-4}$	54	$9.43 \cdot 10^{-4}$
10^{-6}	61	$3.00 \cdot 10^{-4}$	50	$8.38 \cdot 10^{-4}$

Numeryczny test 5 — suma iteracji



Numeryczny test 5 — czas na iterację



Numeryczny test 5 — porównanie skuteczności

Opis: porównanie całkowitego czasu obliczeń pokazuje, która metoda jest faktycznie szybsza w praktyce.

δ	czas stałe μ [s]	czas adapt. μ [s]	zysk względny	zwycięzca
10^{-4}	0.0629	0.0692	-0.100	stała
10^{-5}	0.0217	0.0458	-1.113	stała
10^{-6}	0.0172	0.0394	-1.290	stała

Wniosek: we wszystkich przypadkach metoda ze stałym μ okazuje się szybsza mimo większej liczby iteracji.

Dziękuję za uwagę