

**Algorytmy numeryczne**  
**Zadanie 2**  
**Paulina Żurawska i Paweł Szczupak**  
**grupa 1, Aplikacje internetowe i bazy danych**

Zadanie polegało na zaimplementowaniu algorytmu eliminacji Gaussa w następujących wariantach:

1. bez wyboru elementu podstawowego
2. z częściowym wyborem elementu podstawowego
3. z pełnym wyborem elementu podstawowego.

Program wykonujący powyższe algorytmy został wykonany i skompilowany w języku C# (wersja 6.0) w Visual Studio, na komputerze Lenovo 700-15ISK o pamięci 8GB (SO-DIMM DDR4, 2133MHz) i procesorze Intel Core i5-6300HQ (4 rdzenie, od 2.3 GHz do 3.2 GHz, 6MB cache). Testy zostały przeprowadzone na dwóch typach: float i double.

**Poprawność implementacji**

Aby ukazać poprawność implementacji poniżej zostały zamieszczone wyniki mnożenia macierzy i wektorów poprzez program, jak i poprzez stronę WolframAlpha. Kolejno są: macierz \* wektor = wektor.

**Wynik przedstawiony poprzez Program:**

```
-0.02778625 0.5970612  
-0.315094 -0.8264008
```

```
-1.579965 -0.1200674
```

```
-0.02778625 0.5970612
```

**Wynik przedstawiony poprzez WolframAlpha:**

Input interpretation: Result:

$$\begin{pmatrix} -0.02778625 & 0.5970612 \\ -0.315094 & -0.8264008 \end{pmatrix} \cdot \begin{pmatrix} -1.579965 \\ -0.1200674 \end{pmatrix} = \begin{pmatrix} -0.0277863 \\ 0.597061 \end{pmatrix}$$

**Hipotezy**

(w poniższej tabelce zamieszczone są przykładowe dane wygenerowane przez program potwierdzające odpowiedzi)

**Hipoteza1:** Dla dowolnego ustalonego rozmiaru macierzy czas działania metody Gaussa w kolejnych wersjach (1, 2, 3) rośnie.

Prawda.

Dla dowolnego ustalonego rozmiaru macierzy czas działania metody Gaussa w kolejnych wersjach (1, 2, 3) rośnie. Jest to spowodowane poprzez coraz bardziej skomplikowane rozwiązania, które zostały zastosowane do odtworzenia algorytmów.

**Hipoteza2:** Dla dowolnego ustalonego rozmiaru macierzy błąd uzyskanego wyniku metody Gaussa w kolejnych wersjach (1, 2, 3) maleje.

Prawda.

Dla dowolnego ustalonego rozmiaru macierzy błąd uzyskanego wyniku metody Gaussa w kolejnych wersjach (1, 2, 3) maleje. Dzieje się tak, ponieważ kolejno wersje algorytmów (1, 2, 3) dają coraz dokładniejsze wyniki.

**Hipoteza3:** Użycie własnej arytmetyki na ułamkach zapewnia bezbłędne wyniki niezależnie od wariantu metody Gaussa i rozmiaru macierzy.

---

## Pytania

**Pytanie1:** Jak zależy dokładność obliczeń (błąd) od rozmiaru macierzy dla dwóch wybranych przez Ciebie wariantów metody Gaussa gdy obliczenia prowadzone są na typie podwójnej precyzji (TD)?

Dla algorytmu Gaussa z częściowym wyborem elementu podstawowego oraz z pełnym wyborem elementu podstawowego, błąd, od macierzy większej jak 1000 dla float i double zwiększa się.

**Pytanie2:** Jak przy wybranym przez Ciebie wariancie metody Gaussa zależy czas działania algorytmu od rozmiaru macierzy i różnych typów?

Dla algorytmu Gaussa z częściowym wyborem elementu podstawowego czas działania algorytmu zwiększa się wraz z wzrostem rozmiaru macierzy. Dla typu double algorytm wykonuje się zawsze trochę dłużej niż dla typu float.

Typ	Rozmiar macierzy	Gauss bez wyboru elementu podstawowego: czas	Gauss bez wyboru elementu podstawowego: błąd	Gauss z częściowym wyborem elementu podstawowego: czas wykonania	Gauss z częściowym wyborem elementu podstawowego: błąd	Gauss z pełnym wyborem elementu podstawowego: czas	Gauss z pełnym wyborem elementu podstawowego: błąd
Float	300	00:00:14:221	0.222689440063579	00:00:14:505	8.80919101255984E-05	00:00:33:80	2.65021473389027E-05
Double	300	00:00:14:307	5.04396524547701E-11	00:00:14:497	1.59872445546023E-13	00:00:34:148	8.70414851306123E-14
Float	500	00:00:16:640	0.0085823177448745	00:00:17:56	8.26958733384231E-05	00:00:43:206	4.55427213328363E-05
Double	500	00:00:17:167	1.77781345200856E-10	00:00:18:286	6.57252030578093E-13	00:00:43:818	1.06581410364015E-13
Float	750	00:00:54:345	0.00755160430205137	00:00:54:540	4.47321392389455E-05	00:02:18:709	3.8897563001683E-06
Double	750	00:00:55:182	5.21538368047914E-11	00:00:55:578	2.43360886997834E-13	00:02:23:110	4.79616346638068E-14
Float	1000	00:02:03:259	0.000264675201663067	00:02:04:259	7.47808564760533E-05	00:05:28:432	5.04139637413914E-06
Double	1000	00:02:11:22	2.03215222427389E-12	00:02:11:60	1.88293824976427E-13	00:05:41:456	8.5265128291212E-14
Float	1250	00:04:18:465	16.0061246656135	00:04:59:77	0.000109305417112182	00:11:18:162	0.000108717569975018
Double	1250	00:04:32:682	1.77720949068316E-09	00:05:00:160	2.55795384873636E-13	00:11:37:916	4.12114786740858E-13
Float	1500	00:06:36:428	0.102927785217712	00:06:37:890	0.000149622852937625	00:18:02:564	0.0012502738431408
Double	1500	00:07:01:26	2.73914224635519E-11	00:07:05:111	9.9475983006414E-13	0:18:13:685	5.91846538655227E-13
Float	1750	00:10:45:725	4.88335849622245	00:10:59:781	0.00148874475524252	00:38:00:885	0.0013930512167162
Double	1750	00:15:29:198	1.89476878631467E-09	00:15:34:149	9.98514039074886E-13	00:39:50:257	6.70006239588838E-13

## Zakres prac

Paulina Żurawska – Implementacja klas generycznych

Paweł Szczupak - Implementacja algorytmów

Wspólnie – Sprawozdanie, testy