24 grudnia 2019

Paweł Golik
298868
G1

Aproksymacja trygonometryczna ciągła w przedziale $[0, 2\pi]$. Całkowanie złożoną formułą trapezów. Tablicowanie funkcji, przybliżenia i błędu w m punktach przedziału $[0, 2\pi]$ oraz obliczanie błędu średniokwadratowego w tych punktach.

Projekt nr 77

1 Wstęp

Zamierzeniem projektu jest zbadanie problemu aproksymacji trygonometrycznej ciągłej funkcji jednej zmiennej rzeczywistej o przebiegu okresowym, w m punktach przedziału $I = \{x \in [0, 2\pi]\}$ przy wykorzystaniu całkowania złożoną formułą trapezów do wyznaczenia funkcji aproksymującej *f*(x)* będącej sumą szeregu Fouriera.

W projekcie uwzględniono zagadnienie wpływu liczby punktów aproksymacji m na czas i dokładność obliczeń, tablicowanie wartości funkcji, przbliżenia oraz błędu średniokwadratowego w punktach aproksymacji. Wykreślono także poglądowe wykresy funkcji aproksymowanej oraz aproksymującej.

Algorytm przetestowany został na różnych przypadkach funkcji jednej zmiennej rzeczywistej $f : I \to \mathbf{R}$ dla różnych wartości m, od 5 do 200. Dla większości funkcji interpolacja była dokładna dla niewielkiego m, z wyjątkiem funkcji posiadających asymptoty pionowe na badanym przedziale - w punktach nieokreślonych błąd był duży, co rzutowało niekorzystnie na wartość błędu średniokwadratowego. Dla funkcji nieokresowych błąd aproksymacji był zbyt duży, aby użyta metoda miała praktyczne zastosowanie.

2 Opis użytych metod

Aproksymacja funkcji:

W praktyce czasami funkcja, którą chcemy aproksymować, jest okresowa. W przypadku dyskretnym, wartości f_i mogą być na przykład wartościami pomiarowymi pewnego zjawiska, o którym wiadomo, że ma przebieg okresowy. Wtedy do aproksymacji korzystniej jest stosować wielomiany trygonometryczne:

$$f^*(x) = a_0 + \sum_{j=1}^{m} (a_j cos(jx) + b_j sin(jx)) \tag{1}$$

Jest to $m - ta$ suma szeregu Fouriera, będąca f^* - funkcją aproksymującą.

Niech $V = L_p^2(0, 2\pi)$ będzie przestrzenią rzeczywsitą z iloczynem skalarnym:

$$< g_i, g_j >= \int_0^{2\pi} g_j(x)g_i(x)dx = \begin{cases} 0 & i \neq j, \\ 2\pi & i = j = 0 \\ \pi & i = j \neq 0 \end{cases}, \tag{2}$$

gdzie $0 <= j <= m$, $0 <= i <= m$, m - liczba punktów aproksymacji w przedziale oraz $g_i, g_j \in D$ - baza wielomianów trygonometrycznych z bazą: $\{1, sin(x), cos(x), sin(2x), cos(2x), ..., sin(mx), cos(mx)\}$, ponadto zauważmy, że D jest podprzestrzenią liniową V skończenie wymiarową oraz V posiada zdefiniowaną normę $||g_i||$ wyznaczoną przez ten iloczyn ($||g_i|| = \sqrt{< g_i, g_i >}$).

Zadanie aproksymacji funkcji $f \in V$:

Znaleźć element optymalny $f^* \in D$, taki, że:

$$||f - f^*|| = inf_{g \in D}||f - g||$$

Element $f^* \in D$ jest elementem optymalnym dla $f \in V$ względem podprzestrzeni D wtedy i tylko wtedy, gdy

$$\forall g \in D \qquad < f^* - f, g >= 0$$

.

Utworznie funkcji aproksymującej ***f*(x)*****:**

Niech $dim(D) = n$ $(n = 2m + 1)$ oraz niech $(g_1, g_2, ..., g_n)$ będzie bazą D.
Zatem f^* można zapisać jako:

$$f^* = \sum_{i=1}^{n} \alpha_i g_i, \qquad \alpha_i \in \mathbb{R}$$

Warunek $\forall g \in D \quad < f^* - f, g >= 0$ jest równoważny warunkowi
$\forall i = 1, ..., n \quad < f^* - f, g_i >= 0$, co daje układ n równań:

$$G\alpha = d = \begin{bmatrix} < g_1, g_1 > & < g_1, g_2 > & ... & < g_1, g_n > \\ < g_2, g_1 > & < g_2, g_2 > & ... & < g_2, g_n > \\ ... & ... & ... & ... \\ < g_n, g_1 > & < g_n, g_2 > & ... & < g_n, g_n > \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ ... \\ \alpha_n \end{bmatrix} = \begin{bmatrix} < f, g_1 > \\ < f, g_1 > \\ ... \\ < f, g_n > \end{bmatrix}$$

W tym przypadku funkcja aproksymująca jest kombinacją liniową funkcji z bazy: $\{1, sin(x), cos(x), sin(2x), cos(2x), ..., sin(mx), cos(mx)\}$.

Z (2) wynika, że macierz Grama (G) jest diagonalna, a rozwiązaniem układu równań normalnych jest $\alpha = [a_0, a_1, b_1, a_2, b_2, ..., a_m, b_m]^T$, gdzie:

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(x)dx, \qquad a_i = \frac{1}{\pi} \int_0^{2\pi} f(x)cos(jx)dx, \qquad b_i = \frac{1}{\pi} \int_0^{2\pi} f(x)sin(jx)dx$$

dla $i = 1, 2, ..., m$.

Całkowanie przy użyciu złożonej formuły trapezów:

Przedział całkowania $[a, b]$ dzielimy na podprzedziały $[x_k, x_{k+1}]$ $(k = 1, 2, ..., N)$ o długości $H = \frac{b-a}{N}$, przy czym $x_k = a + kH$ dla $k = 0, 1, ..., N$.

Złożony wzrór trapezów:

$$S(f) = \frac{H}{2}(f(a) + f(b) + 2\sum_{k=1}^{N-1} f(a + kH))$$

W projekcie przyjęto $N = 1000$.

Błąd aproksymacji:

$$e_u = inf_{g \in D}||f - g|| = ||f - f^*||$$

Z twierdzenia Pitagorasa:

$$e_u = \sqrt{||f||^2 - ||f^*||^2}, \qquad gdzie \ \ ||f|| > ||f^*||$$

Obliczanie błędu średniokwadratowego:

Po obliczeniu wartości błędów we wszystkich m punktach, dzięki ich tablicowaniu, łatwo możemy obliczyć błąd średniokwadratowy:

$$MSE = \frac{1}{m}\sum_{k=1}^{k=m}(f(x) - f^*(x))^2$$

3 Działanie programu:

Początkowo program wyznacza wartości kolejnych współczynników a_i, b_i potrzebnych do stworzenia funkcji aproksymującej zgodnie ze wzorem (1). Do obliczenia każdej z całek zastosowano złożoną kwadraturę trapezów (*funkcje: trapezy_a, trapezy_b, trapezy*), a następnie wyliczono wartość w m punktach funkcji aproksymującej, wartości rzeczywiste funkcji aproksymowanej oraz błąd w tych punktach z uwzględnieniem tablicowania w macierzy S, będacej macierzą $3 \times m$ (*funkcja run*).

Powyższe obliczenia przeprowadzono dla wielu wartości $m = [5, 10, 15, 30, 50, 70, 100, 200]$ przy jednoczesnym pomiarze czasu działania programu (w sek.) za pomocą wbudowanej funkcji środowiska MATLAB (*tic toc*).

Następnie rezultat badań zaprezentowano w formie tablicy S, wartości błędu średniokwadratowego dla największego m oraz wykresów:

1. funkcji aproksymowanej i aproksymującej dla każdej wartości m (*wbudowana funkcja plot*)
2. wykresu błędu i czasu trwania programu w zależności od m (*wbudowana funkcja semilog).*

W oparciu o wartości błędów aproksymacji stwierdzono o poprawnym, bądź niepoprawnym działaniu algorytmu.

4 Prezentacja przykładów oraz wyników

Działanie programu dla funkcji okresowych, dobrze określonych na przedziale I:

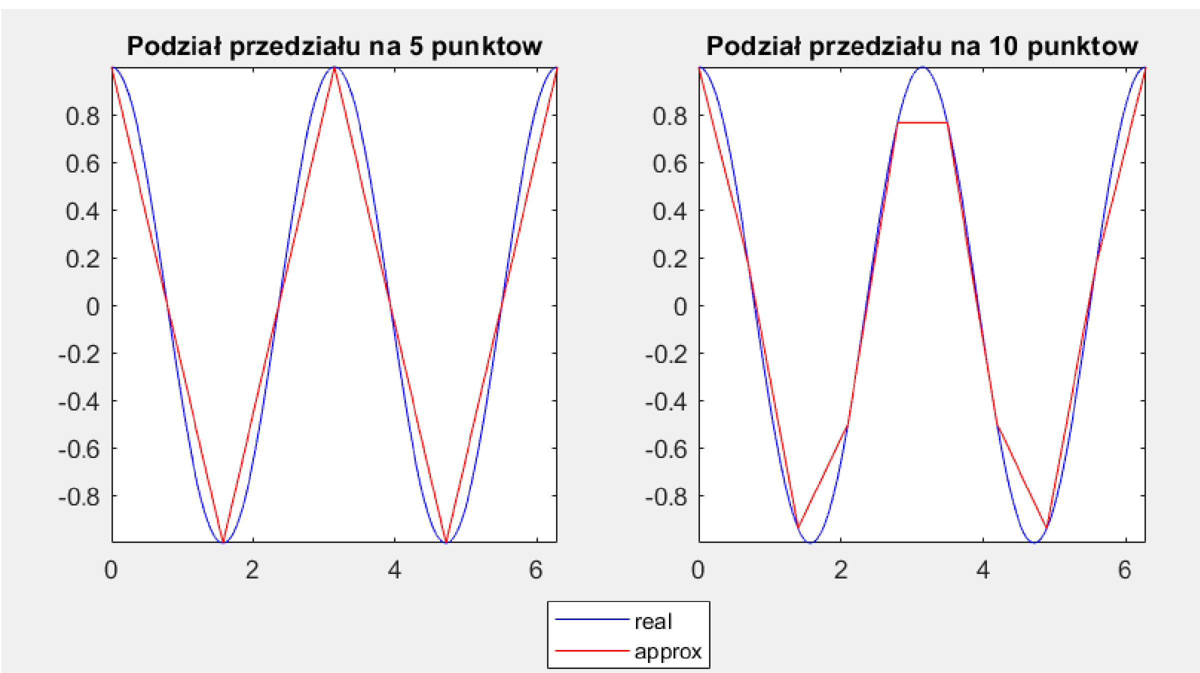
Działanie programu dla różnorodnych funkcji okresowych dobrze określonych na przedziale.

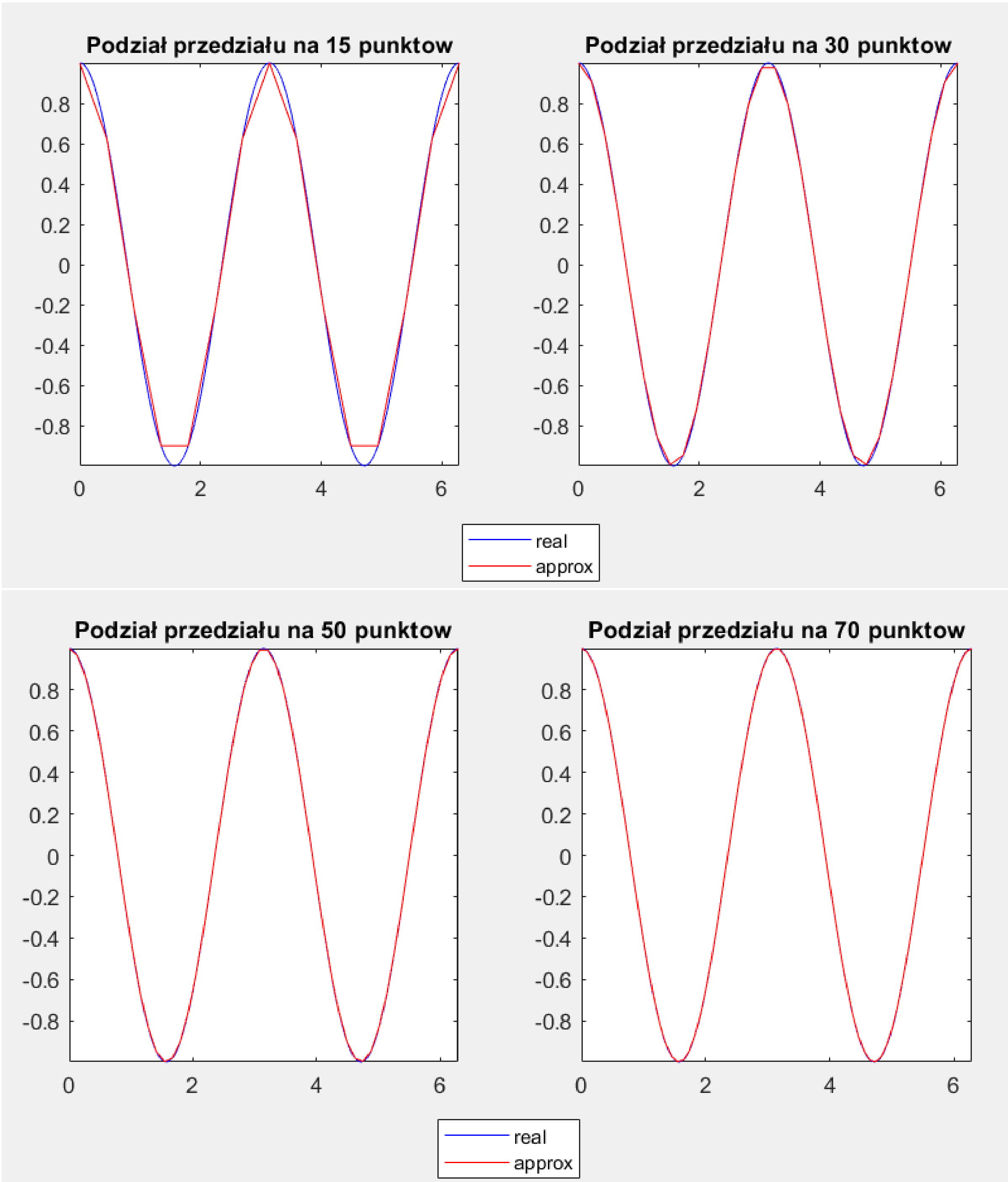
1. $f(x) = cos^4(x) - sin^4(x)$

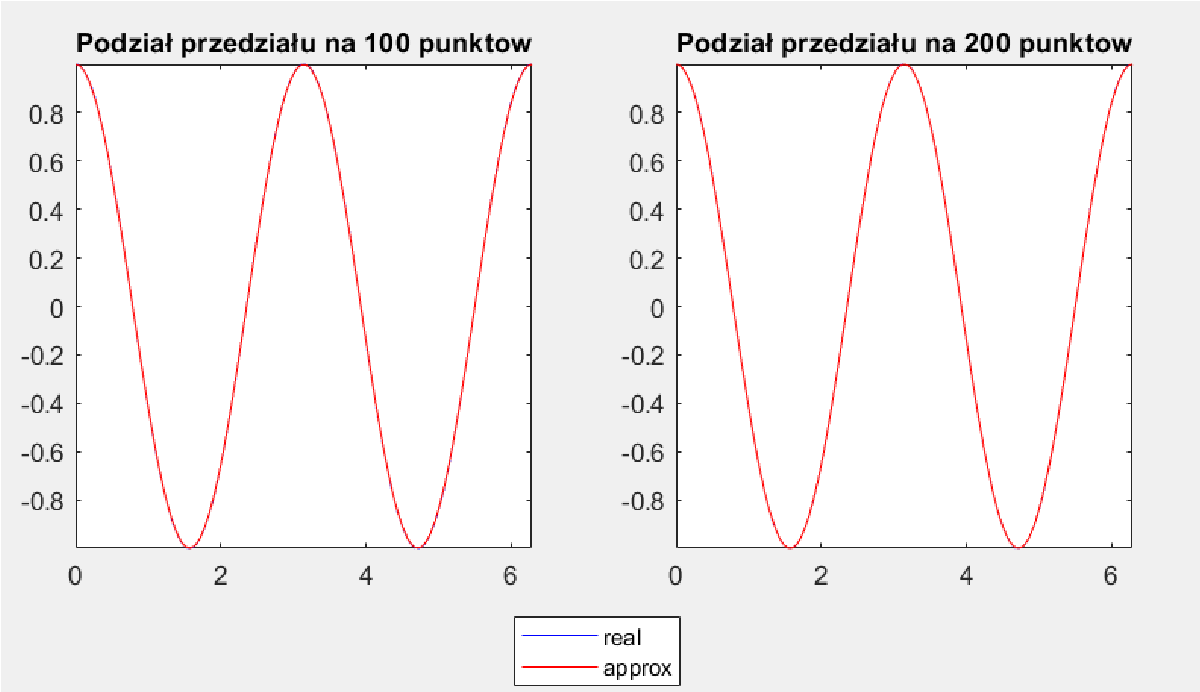
Tablicowanie wartości, przybliżenia i błędu oraz błąd średniokwadratowy:

```
Tablicowanie: [real/approx/err]

S =

    1.0000    1.0000    0.0000
    0.9980    0.9980    0.0000
    0.9920    0.9920    0.0000
    0.9821    0.9821    0.0000
    0.9683    0.9683    0.0000

Błąd średniokwadratowy:

mean_square_error =

   5.2088e-15
```

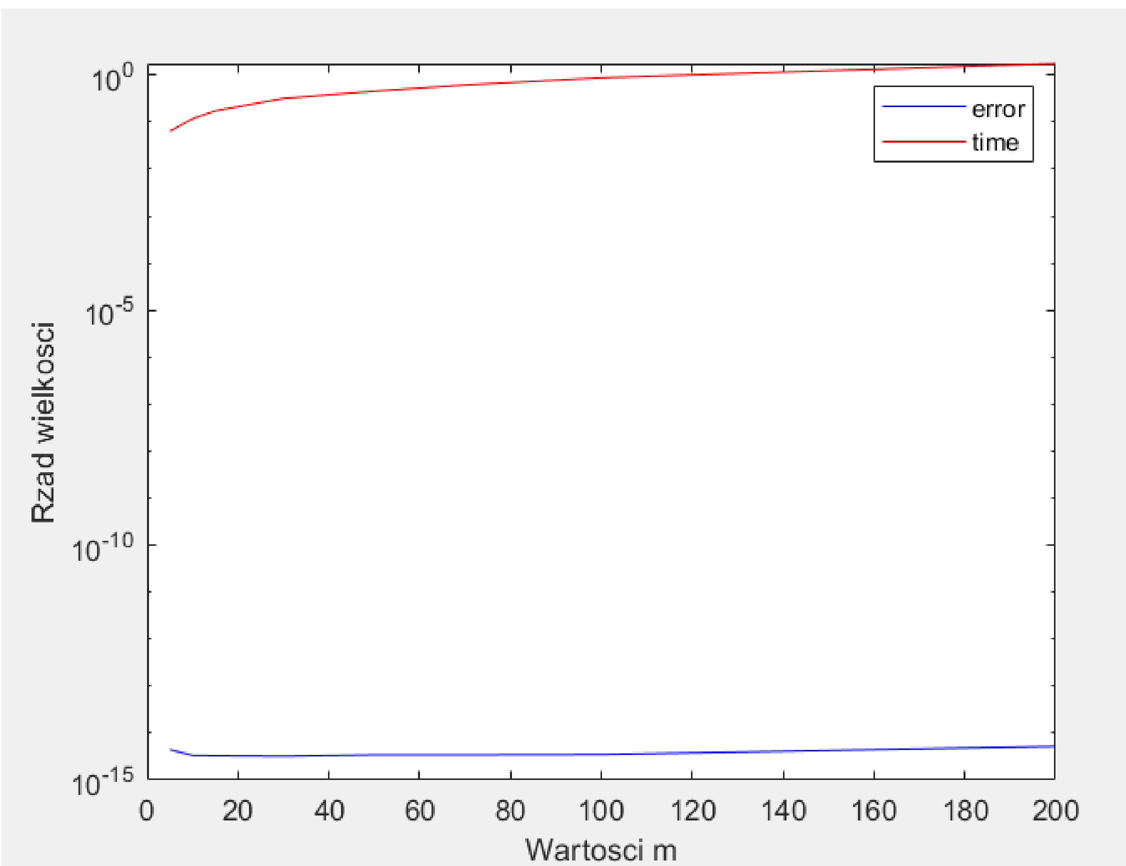
Wykresy f oraz f^* dla poszczególnych wartości m:




Podział przedziału na 15 punktow
Podział przedziału na 30 punktow
real
approx
Podział przedziału na 50 punktow
Podział przedziału na 70 punktow
real
approx



Wykres błędu oraz czasu (sek.) w zależności od liczby punktów m:

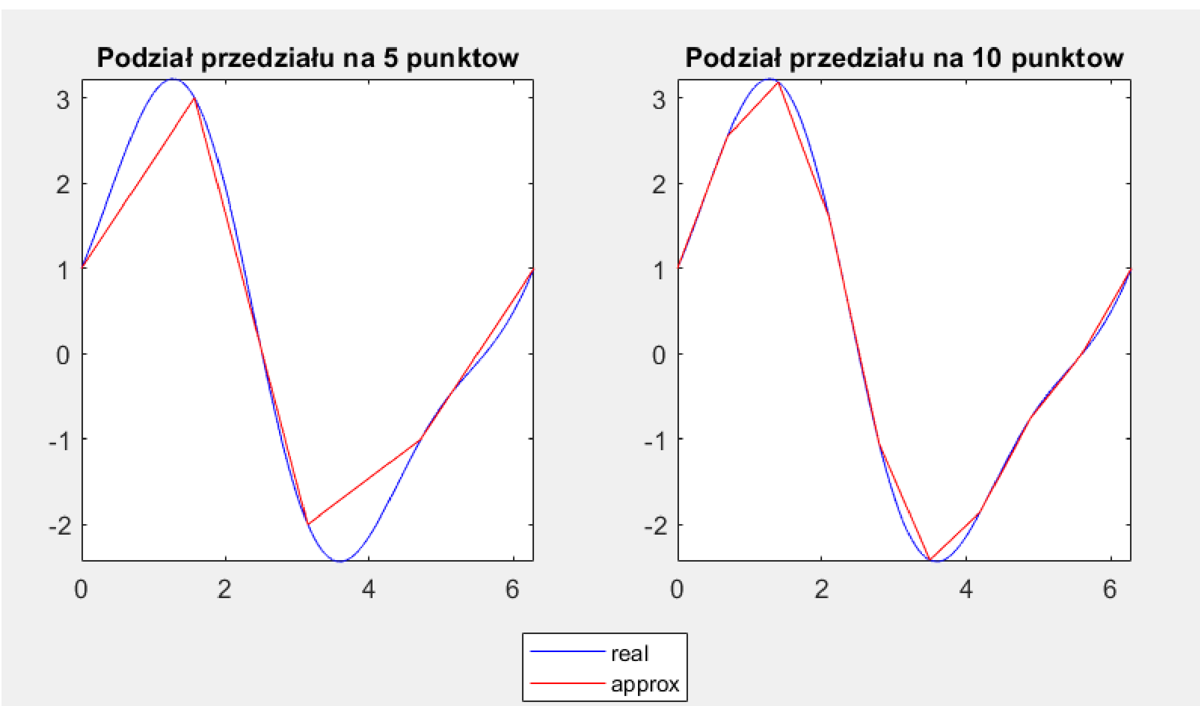


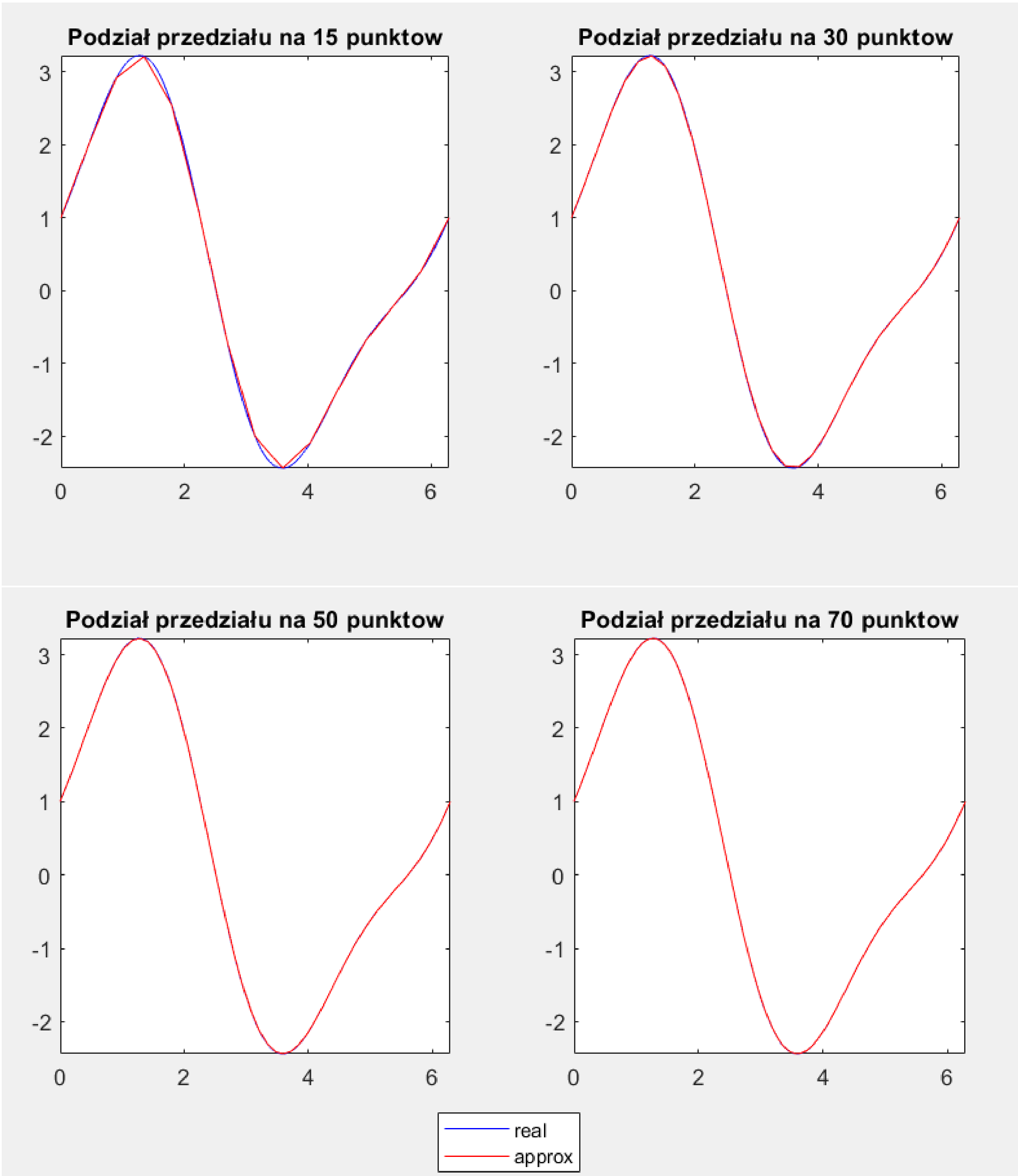
2. $f(x) = 0.25 + 1.5cos(x) + 2sin(x) - 0.75cos(2x)$

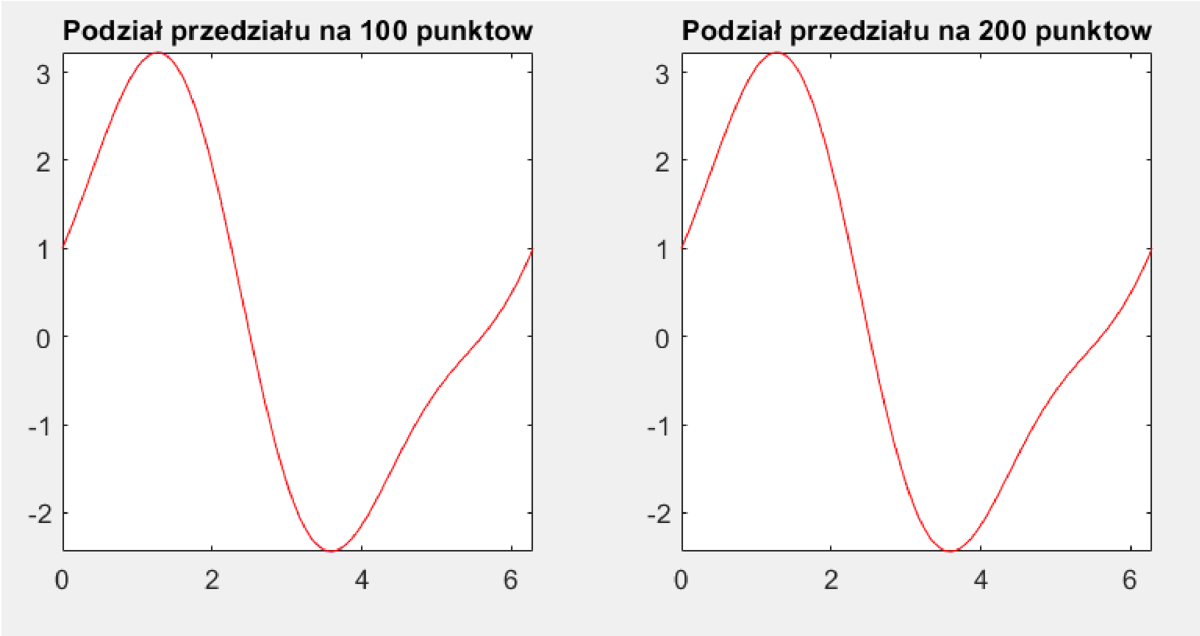
Tablicowanie wartości, przybliżenia i błędu oraz błąd średniokwadratowy:

```
Tablicowanie: [real/approx/err]

S =

    1.0000    1.0000    0.0000
    1.0639    1.0639    0.0000
    1.1292    1.1292    0.0000
    1.1959    1.1959    0.0000

Błąd średniokwadratowy:

mean_square_error =

   2.1017e-14
```

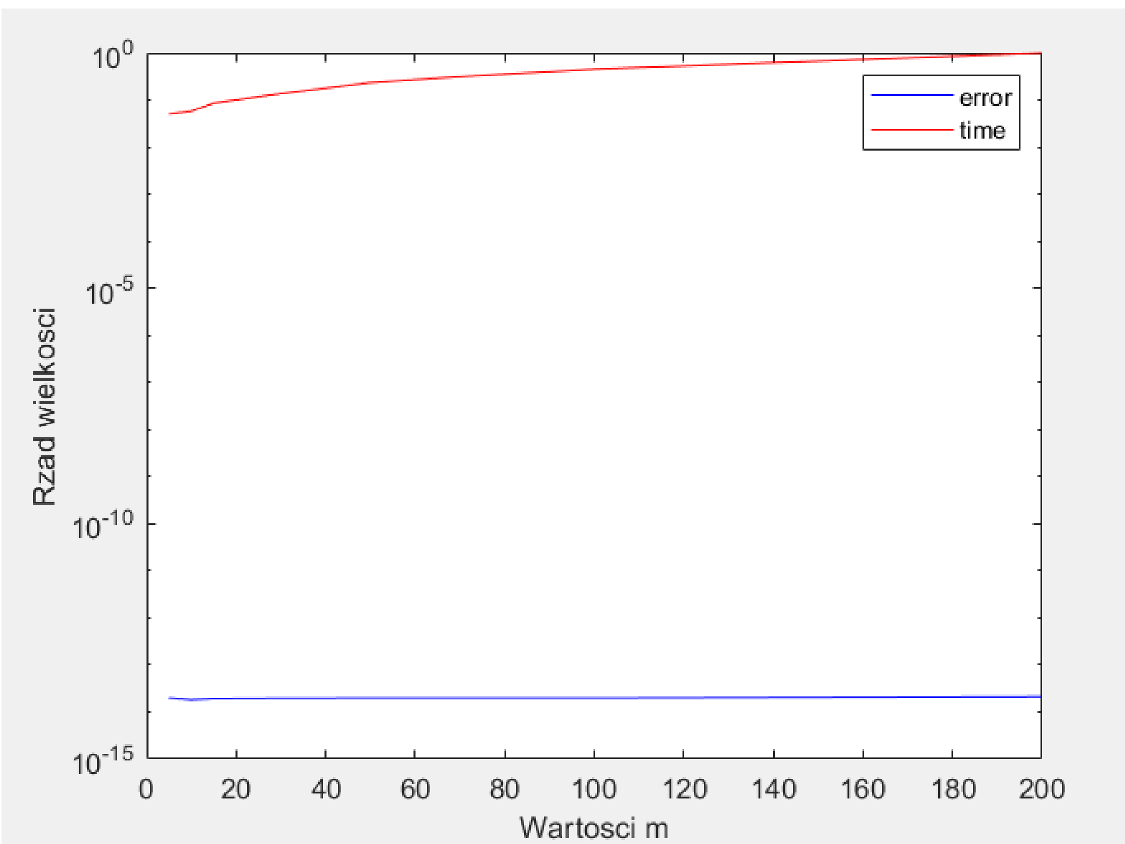
Wykresy f oraz f^* dla poszczególnych wartości m:




Podział przedziału na 15 punktow
Podział przedziału na 30 punktow
Podział przedziału na 50 punktow
Podział przedziału na 70 punktow
3
2
1
0
-1
-2
0
2
4
6
real
approx



Wykres błędu oraz czasu (sek.) w zależności od liczby punktów m:

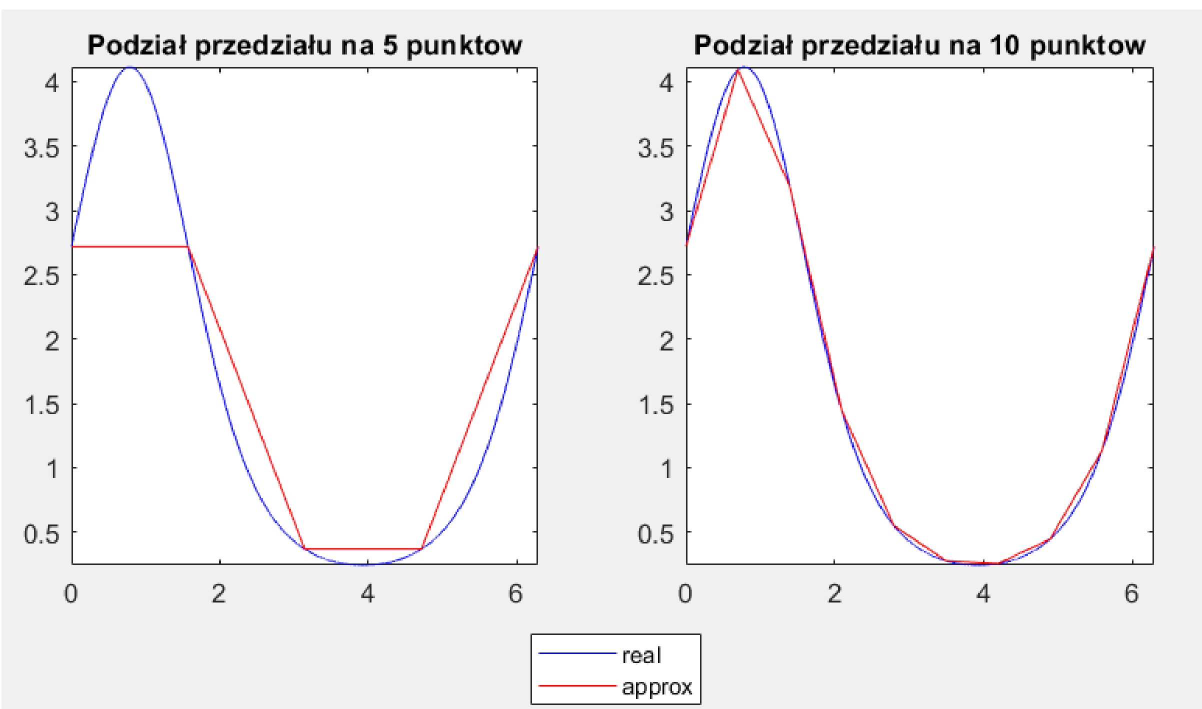


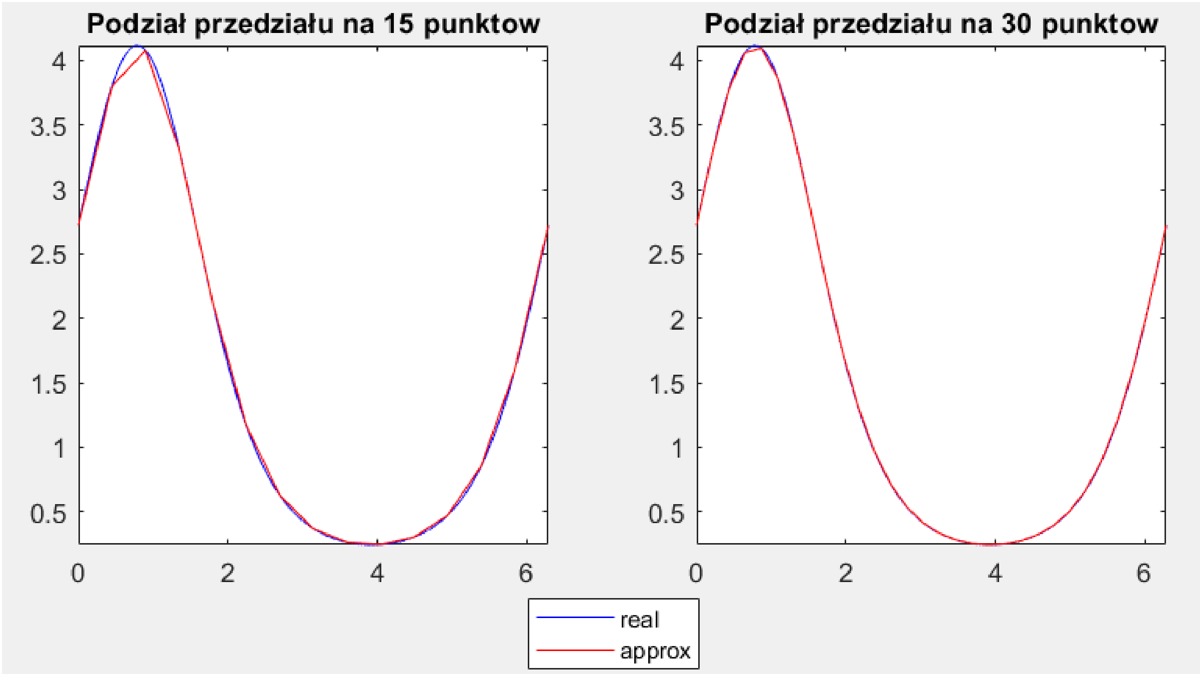
3. $f(x) = e^{sin(x)+cos(x)}$

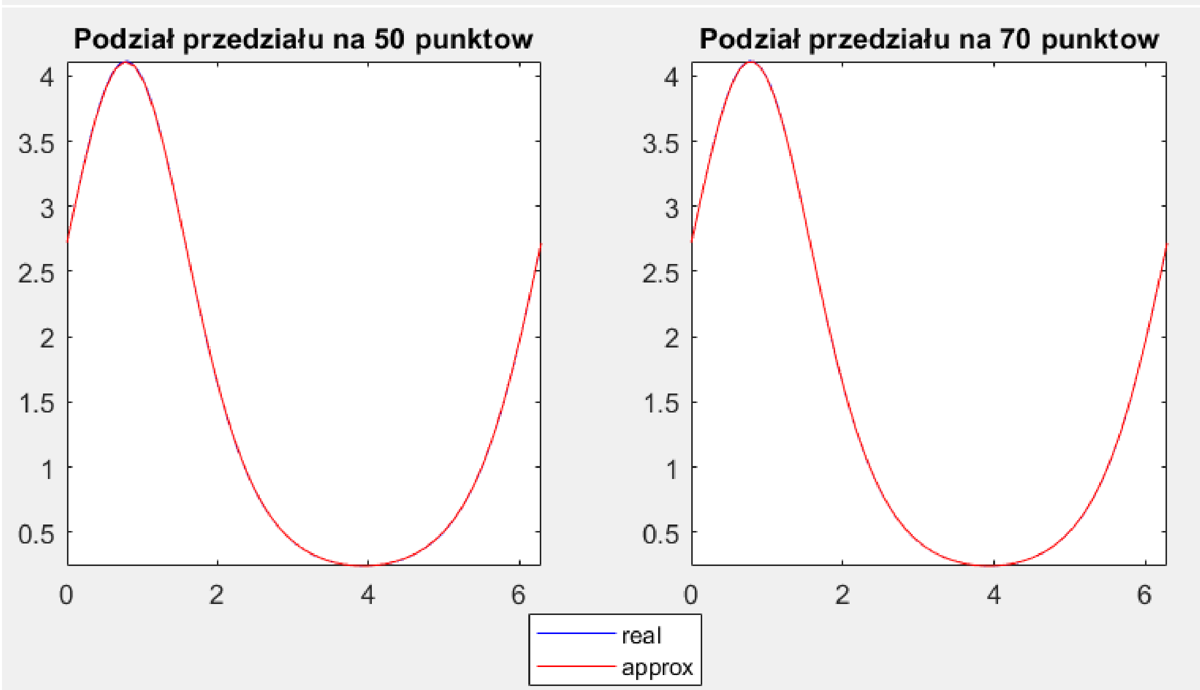
Tablicowanie wartości, przybliżenia i błędu oraz błąd średniokwadratowy:

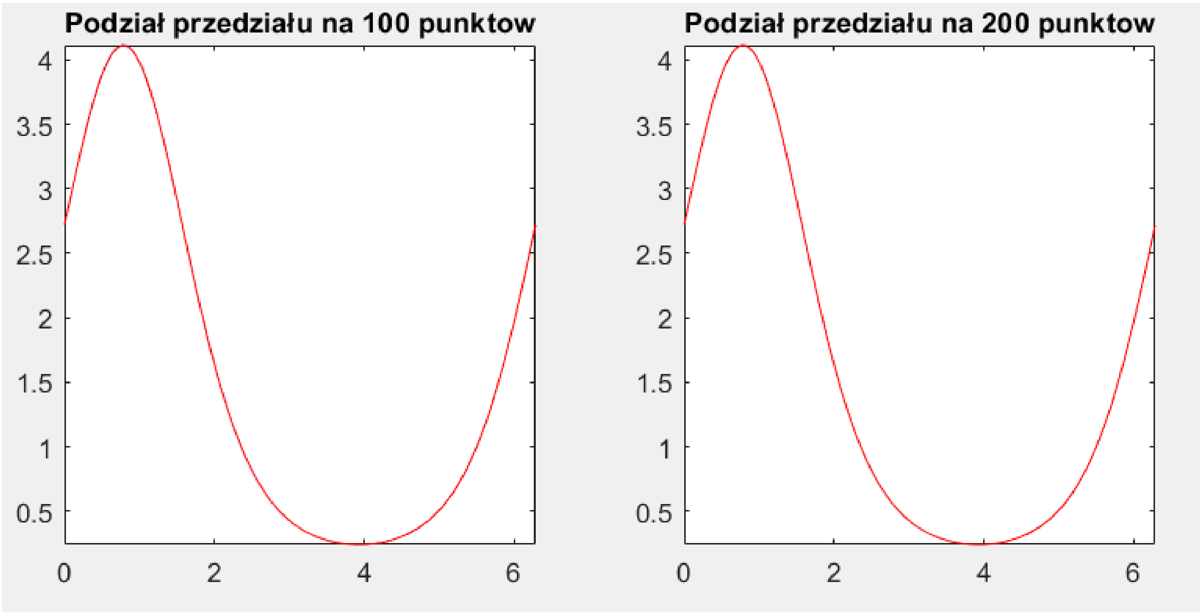
```
Tablicowanie: [real/approx/err]

S =

    2.7183    2.7183    0.0000
    2.8041    2.8041    0.0000
    2.8896    2.8896    0.0000
    2.9746    2.9746    0.0000
    3.0587    3.0587    0.0000

Błąd średniokwadratowy:

mean_square_error =

   2.4249e-14
```

Wykresy f oraz f^* dla poszczególnych wartości m:

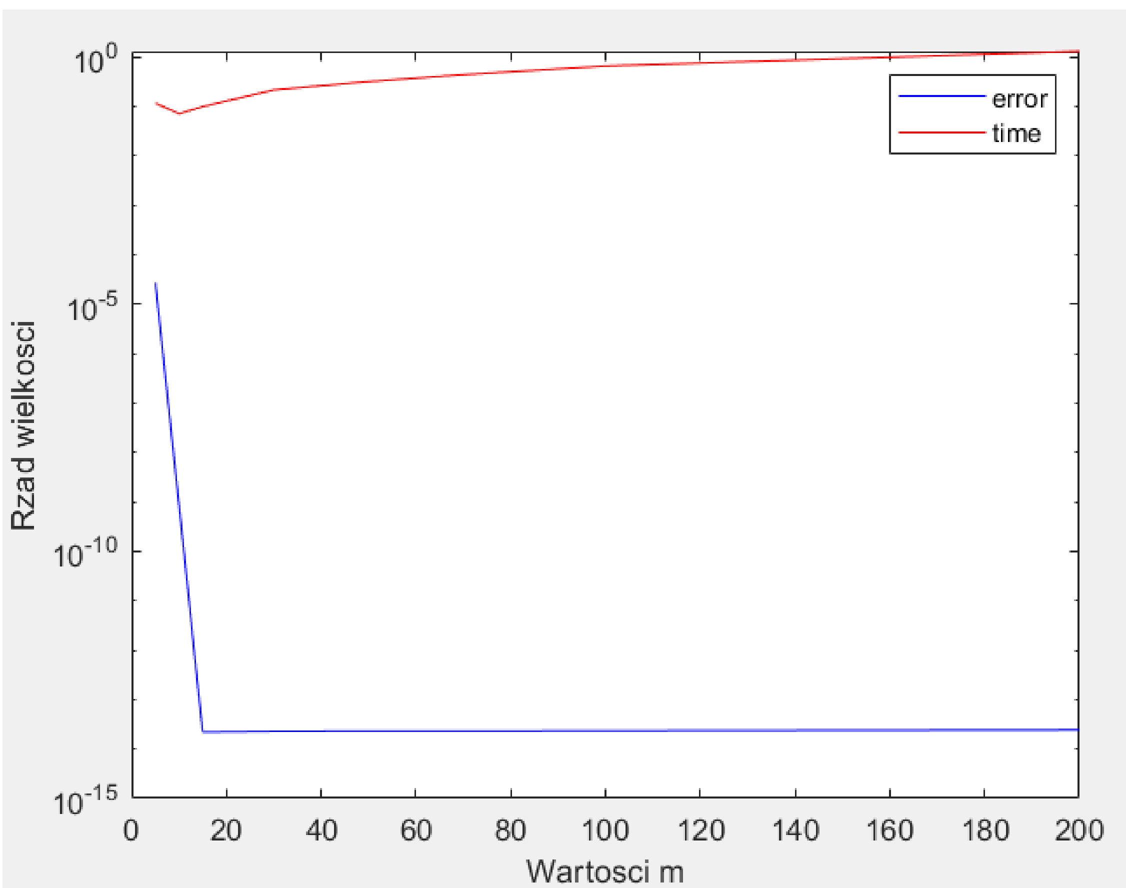



Podział przedziału na 15 punktow
Podział przedziału na 30 punktow
real
approx


Podział przedziału na 50 punktow
Podział przedziału na 70 punktow
real
approx



Wykres błędu oraz czasu (sek.) w zależności od liczby punktów m:

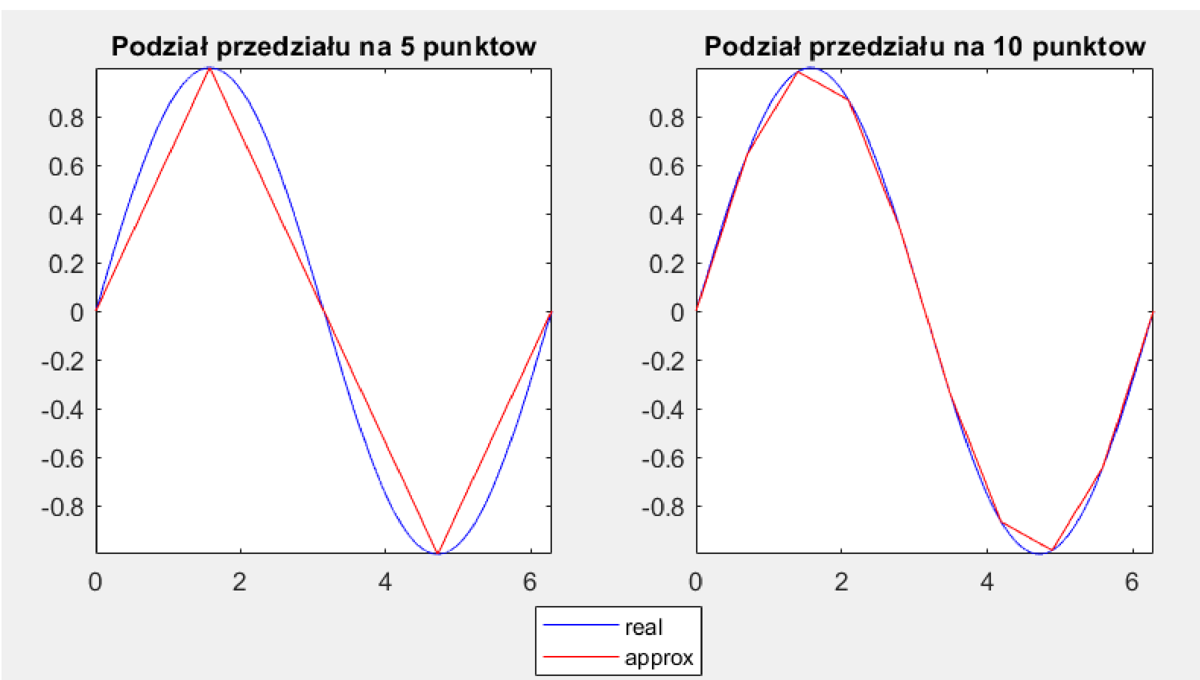


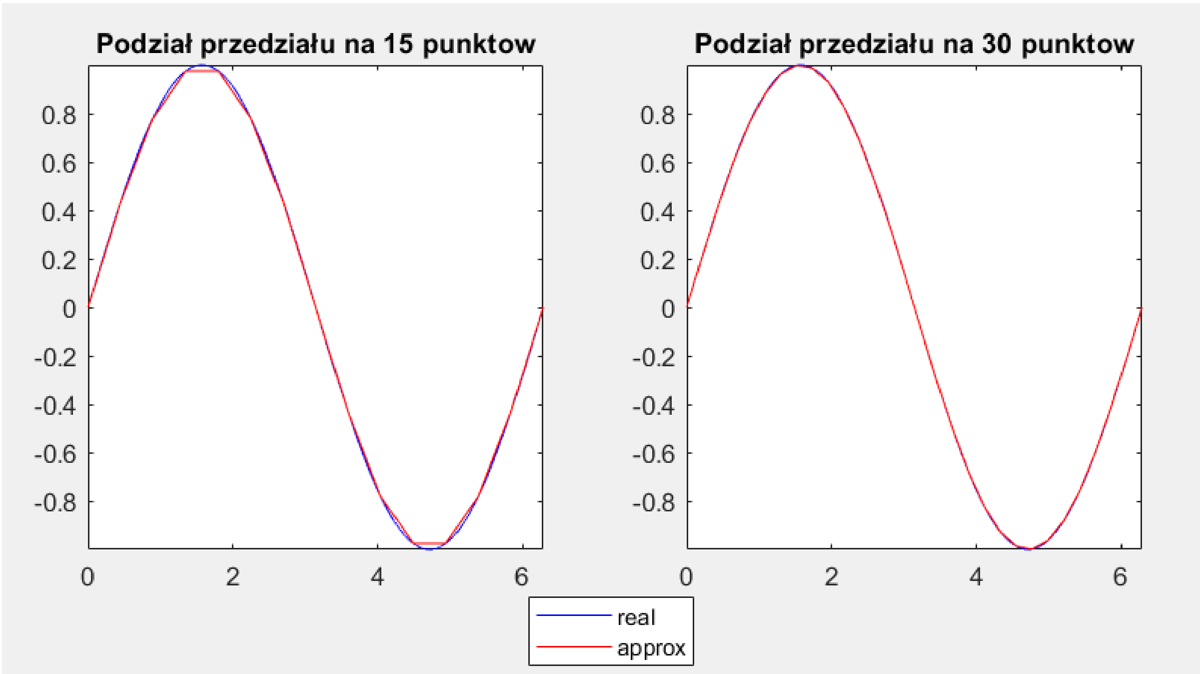
4. $f(x) = sin(x)$

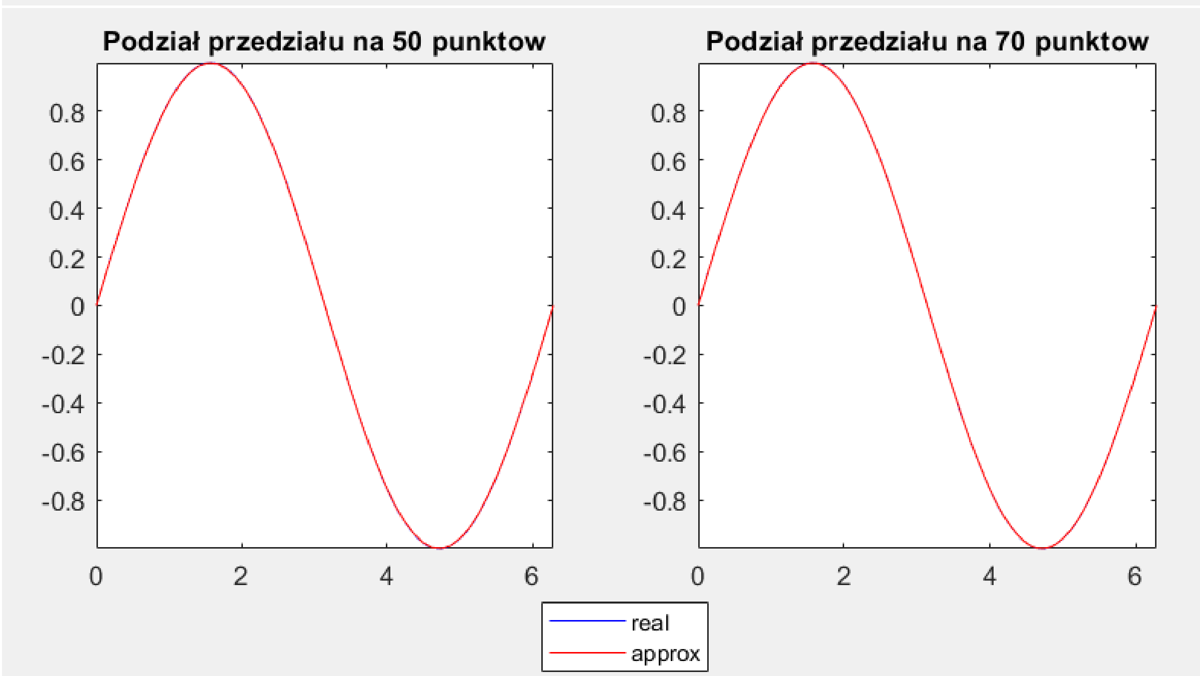
Tablicowanie wartości, przybliżenia i błędu oraz błąd średniokwadratowy:

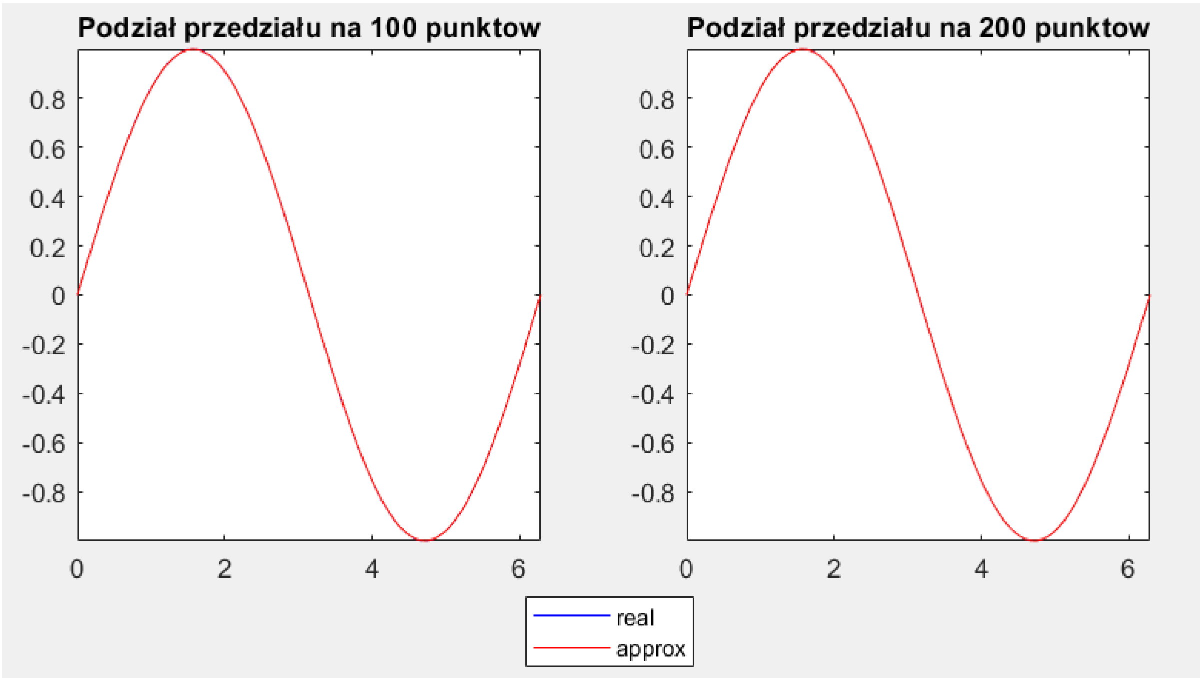
```
Tablicowanie: [real/approx/err]

S =

         0   -0.0000    0.0000
    0.0316    0.0316    0.0000
    0.0631    0.0631    0.0000
    0.0946    0.0946    0.0000
    0.1260    0.1260    0.0000

Błąd średniokwadratowy:

mean_square_error =

   4.4791e-15
```

Wykresy f oraz f^* dla poszczególnych wartości m:

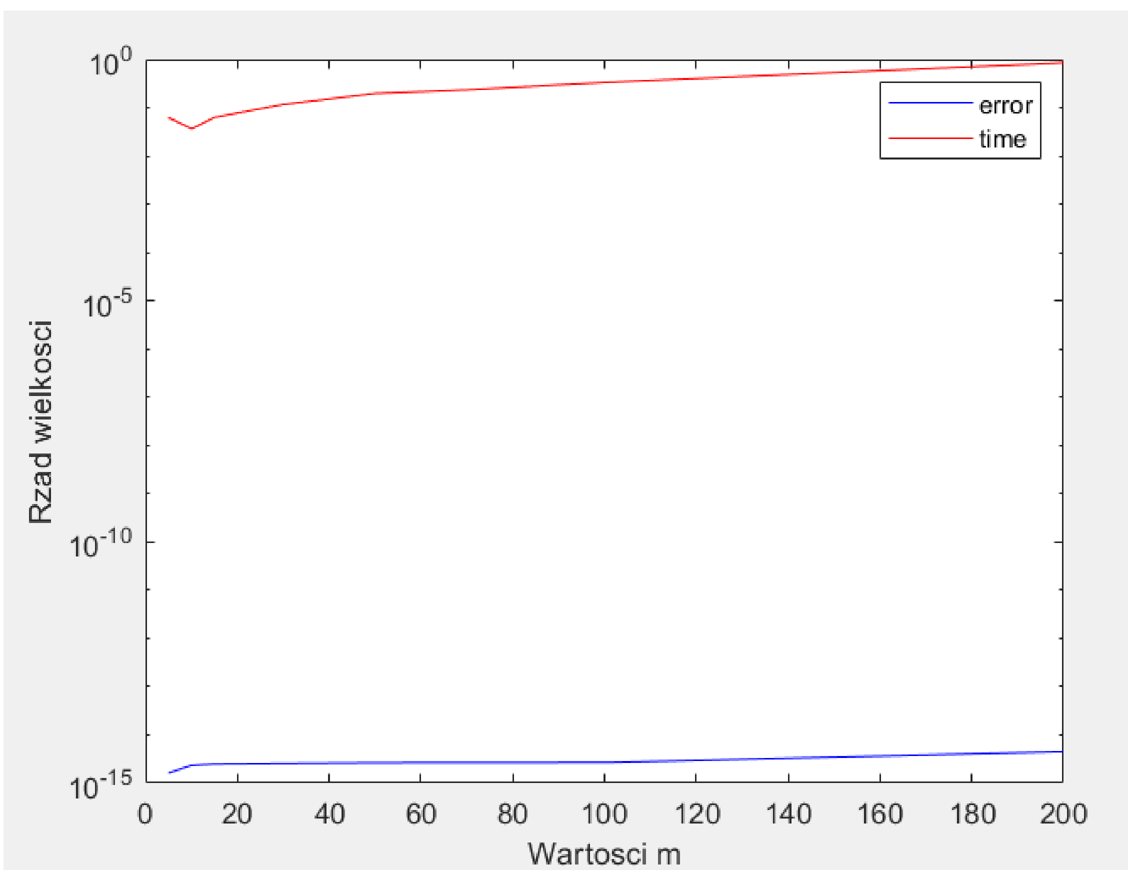



Podział przedziału na 15 punktow
Podział przedziału na 30 punktow
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
2
4
6
real
approx


Podział przedziału na 50 punktow
Podział przedziału na 70 punktow
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
2
4
6
real
approx



Wykres błędu oraz czasu (sek.) w zależności od liczby punktów m:



Działanie programu dla funkcji okresowych, posiadających asymptoty pionowe na przedziale I:

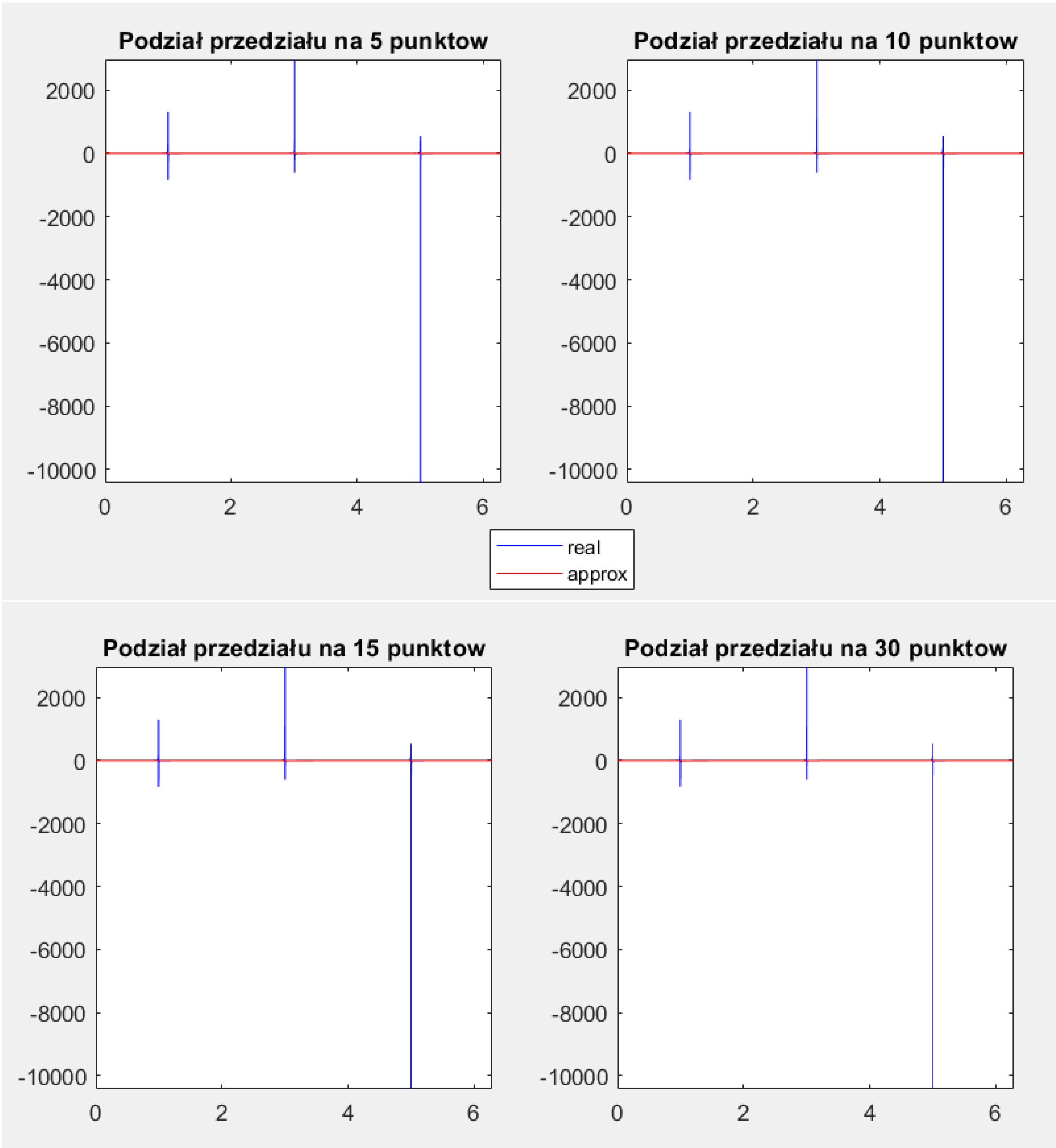
W przypadku występowania asymptoty pionowej w punkcie $x_0 \in I$, wartość aproksymacji w tym punkcie jest przybliżona z dużym niedomiarem, co rzutuje na wynik błędu średniokwadratowego.

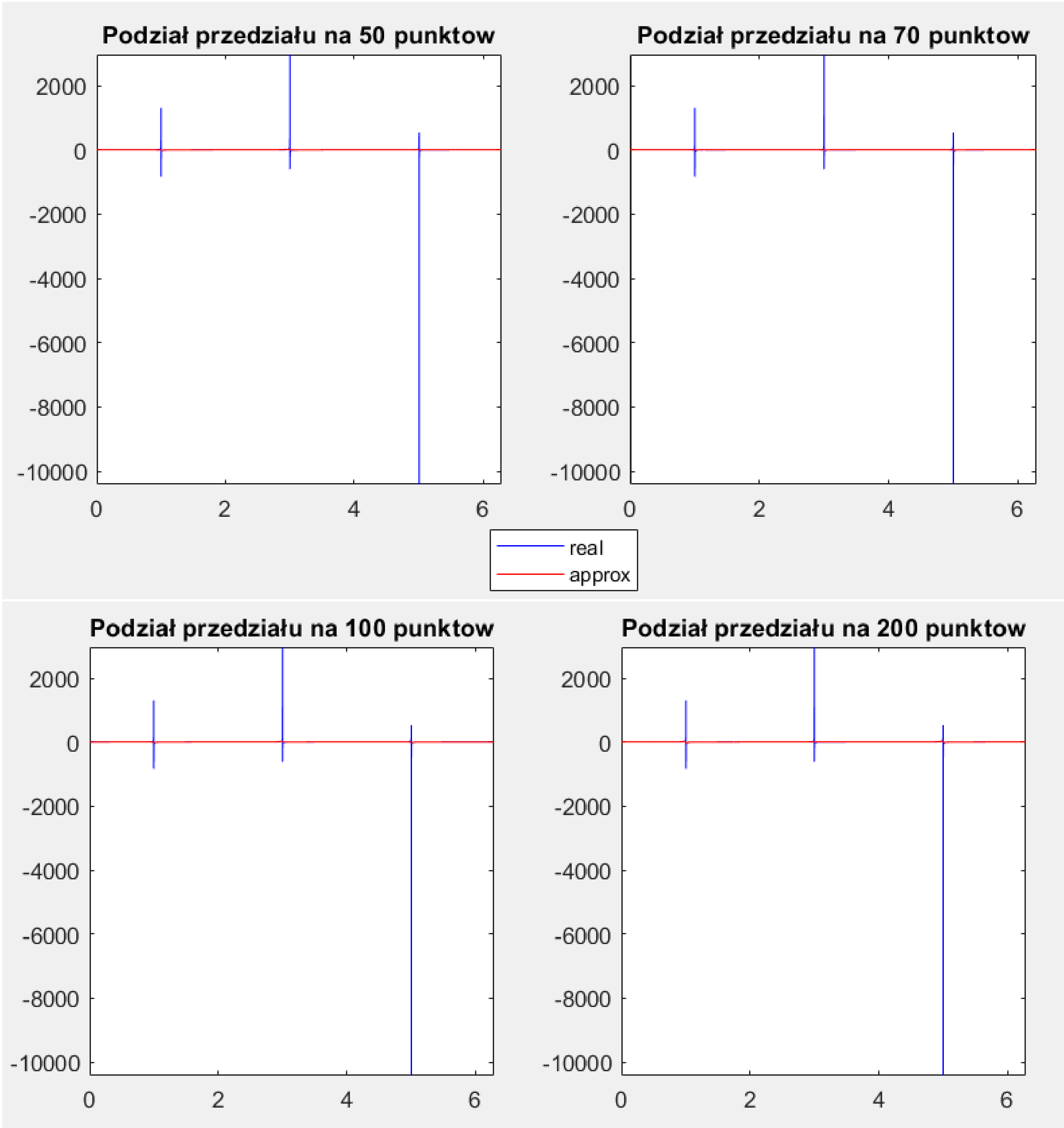
5. $f(x) = 0.5tan(\frac{\pi}{2}x)$

Tablicowanie wartości, przybliżenia i błędu oraz błąd średniokwadratowy:

```
Tablicowanie: [real/approx/err]

S =

         0    0.0883    0.0883
    0.0248    0.0133    0.0115
    0.0498    0.0403    0.0095
    0.0749    0.0713    0.0036
    0.1005    0.1040    0.0034

Błąd średniokwadratowy:

mean_square_error =

   45.5959
```

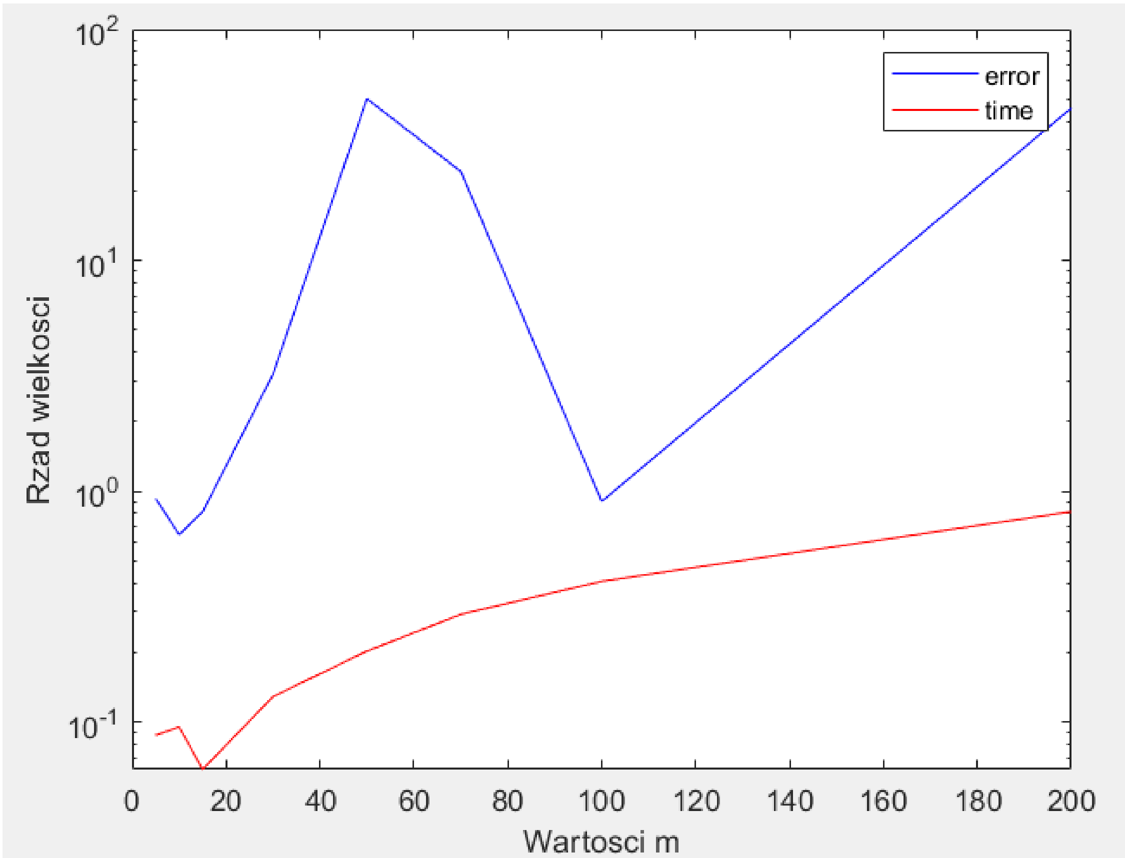
Wykresy f oraz f^* dla poszczególnych wartości m:





Podział przedziału na 50 punktow
Podział przedziału na 70 punktow
Podział przedziału na 100 punktow
Podział przedziału na 200 punktow
2000
0
-2000
-4000
-6000
-8000
-10000
0
2
4
6
real
approx

Wykres błędu oraz czasu (sek.) w zależności od liczby punktów m:



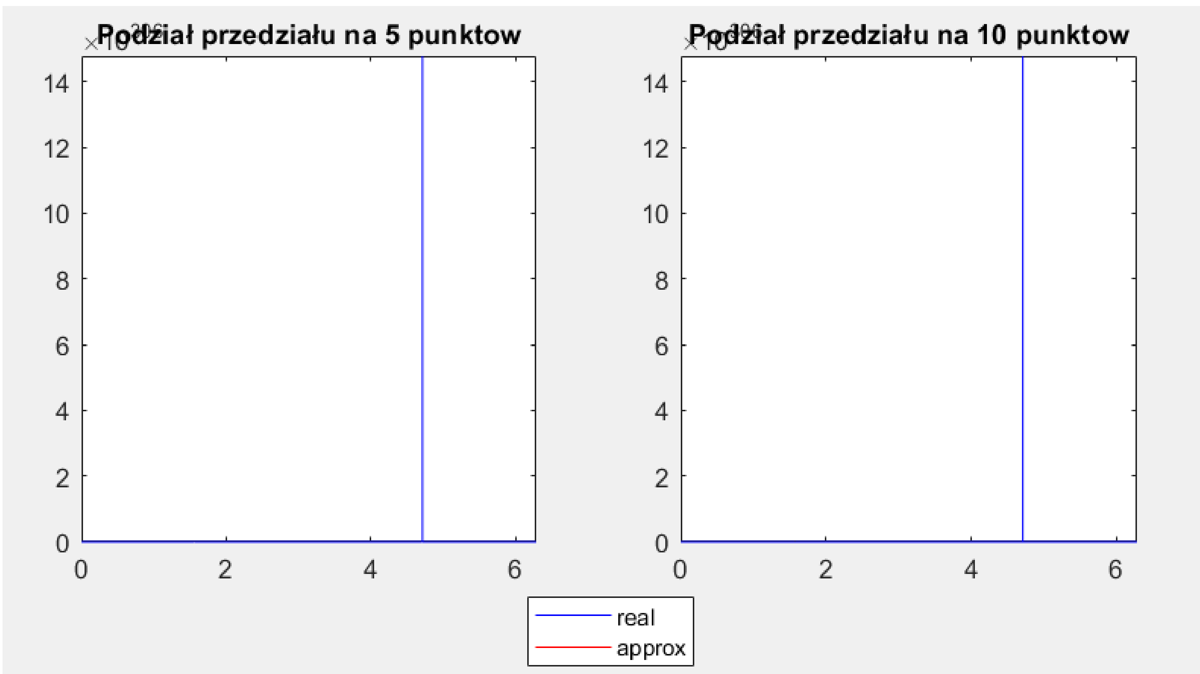
6. $f(x) = e^{tan(x)}$

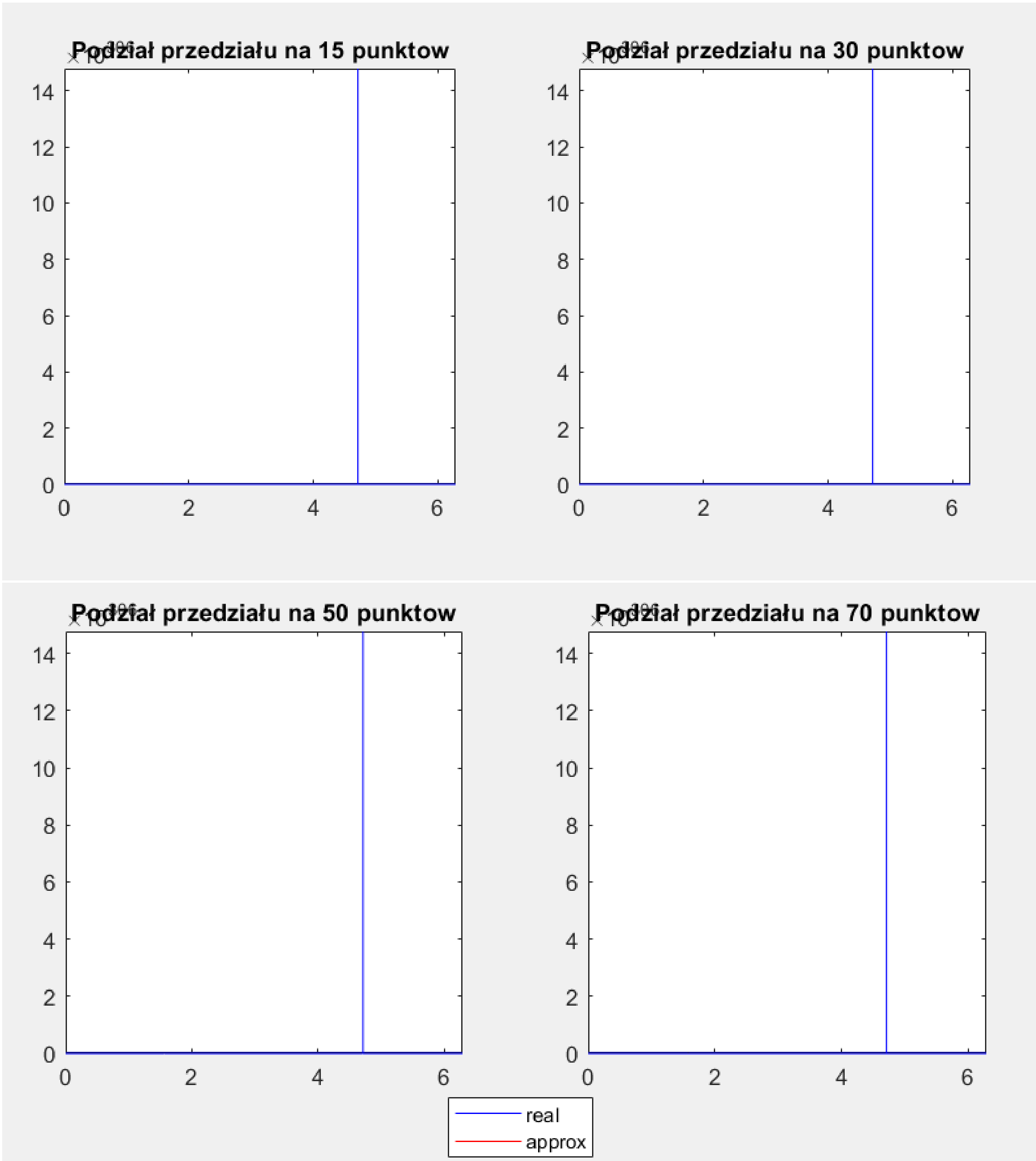
Tablicowanie wartości, przybliżenia i błędu oraz błąd średniokwadratowy:

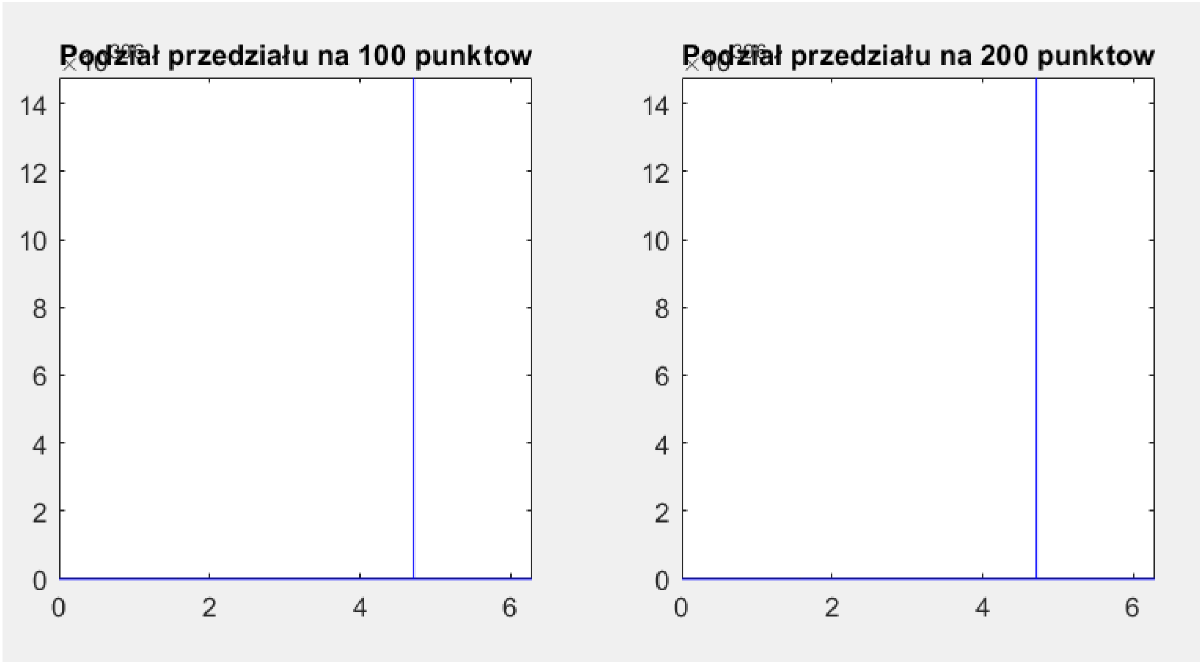
```
Tablicowanie: [real/approx/err]

S =

   1.0e+55 *

     0.0000          NaN          NaN
     0.0000          NaN          NaN
     0.0000          NaN          NaN
     0.0000          NaN          NaN
```

```
Błąd średniokwadratowy:

mean_square_error =

   NaN
```

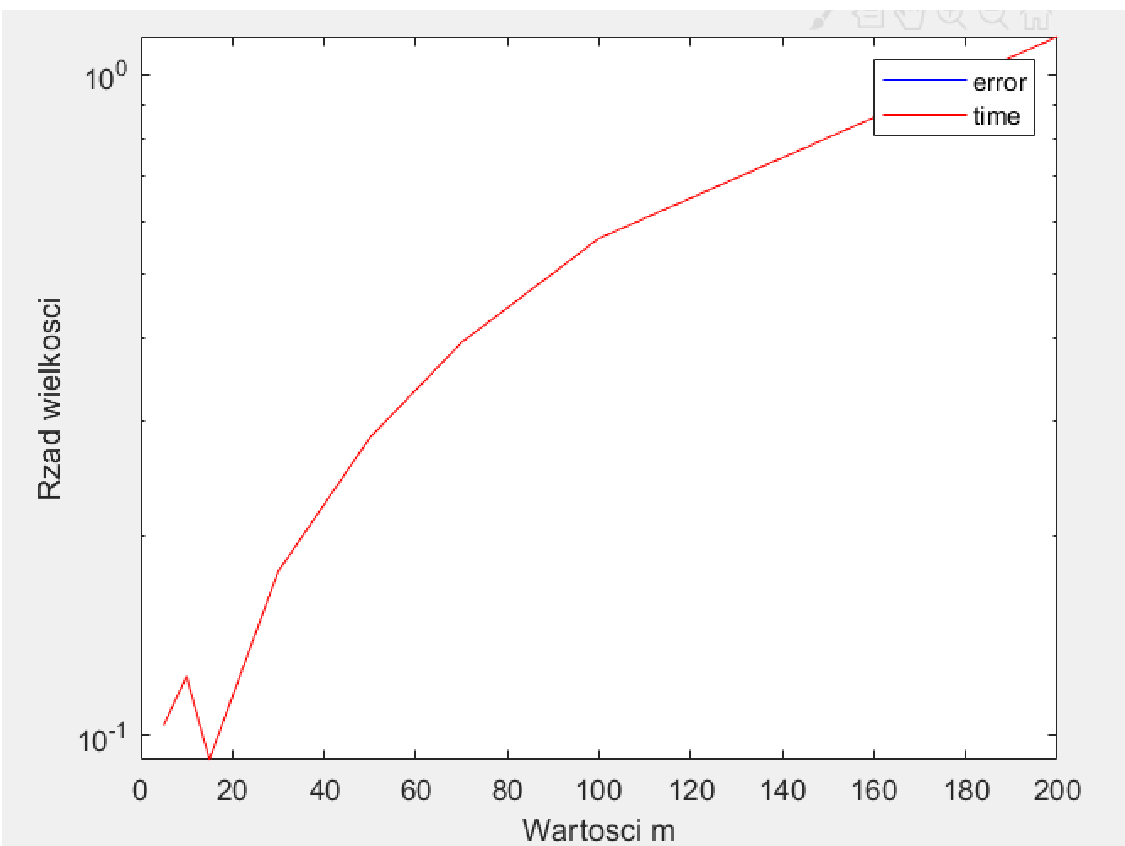
Wykresy f oraz f^* dla poszczególnych wartości m:




Podział przedziału na 15 punktow
Podział przedziału na 30 punktow
Podział przedziału na 50 punktow
Podział przedziału na 70 punktow
real
approx



Wykres błędu oraz czasu (sek.) w zależności od liczby punktów m:



Działanie programu dla funkcji nieokresowej i funkcji stałej :

W przypadku funkcji nieokresowych podany algorytm przybliża funkcję z dużym błędem. Dla funkcji stale równej 1 (przykład 7), której wzór na pierwszy rzut oka przypomina funkcję okresową, aproksymacja oscyluje wokół prostej $y = 1$ co jest bardziej widoczne dla dużej liczby punktów m - zaobserwować można skutki przybliżania funkcji za pomocą szeregu Fouriera. Dla funkcji stałej błąd jest odpowiedniego rzędu, natomiast szereg Fouriera nie radzi sobie z funkcją kwadratową (przykład 8).

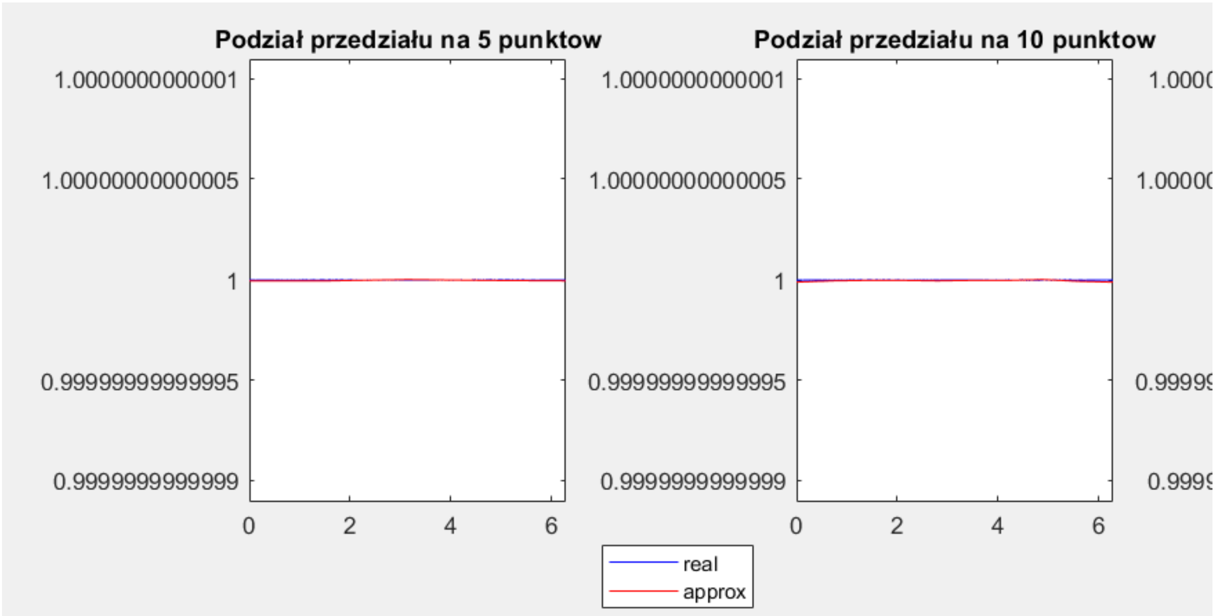
7. $f(x) = cos(2x) + 2sin^2(x)$

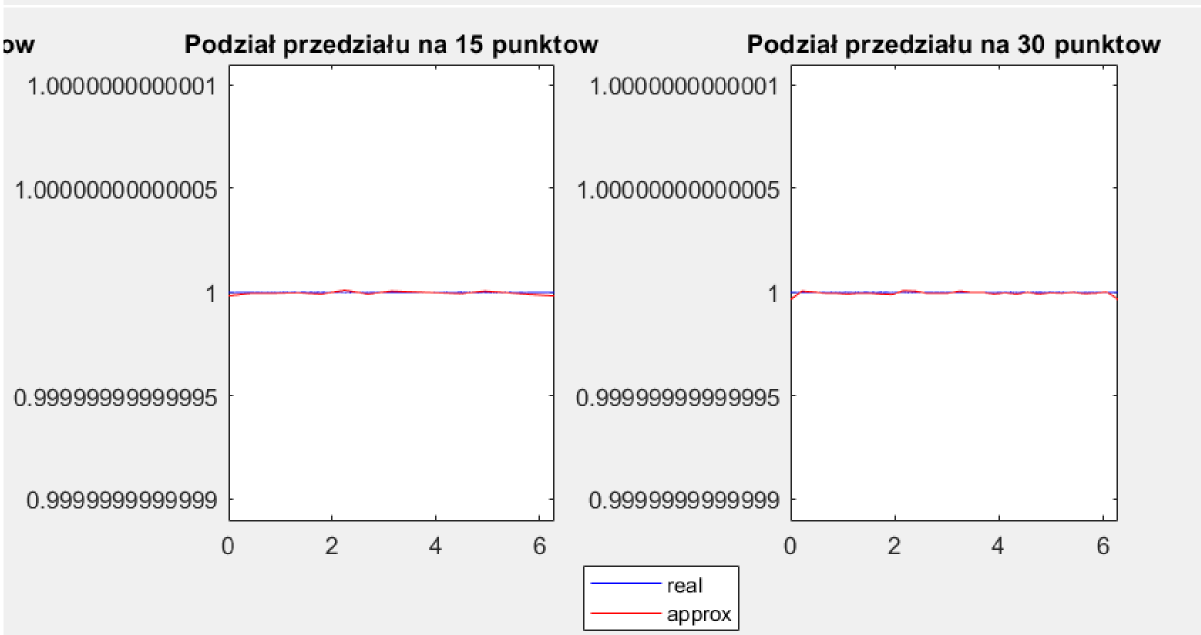
Tablicowanie wartości, przybliżenia i błędu oraz błąd średniokwadratowy:

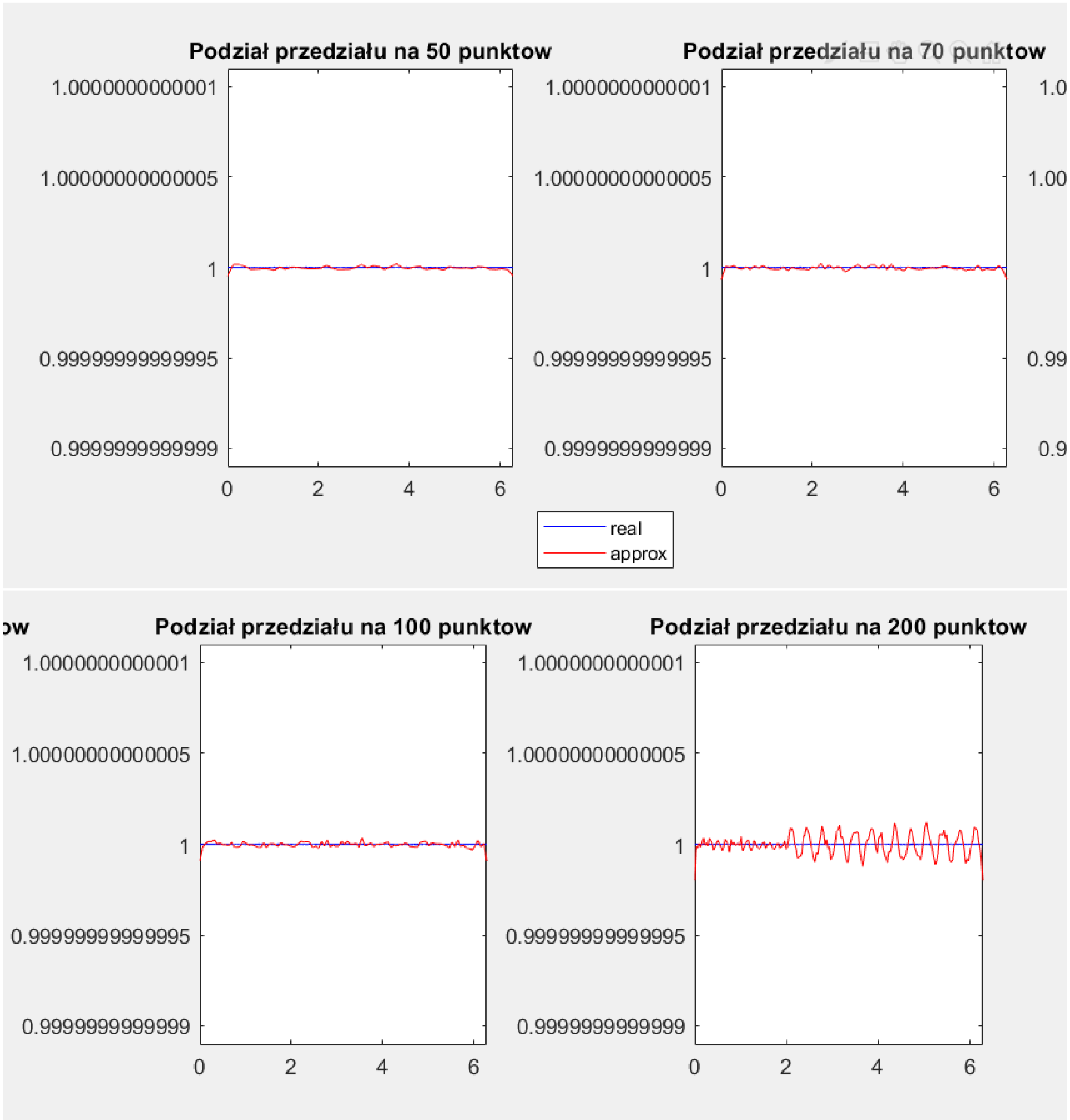
```
Tablicowanie: [real/approx/err]

S =

    1.0000    1.0000    0.0000
    1.0000    1.0000    0.0000
    1.0000    1.0000    0.0000
```

```
Błąd średniokwadratowy:

mean_square_error =

   5.6302e-15
```

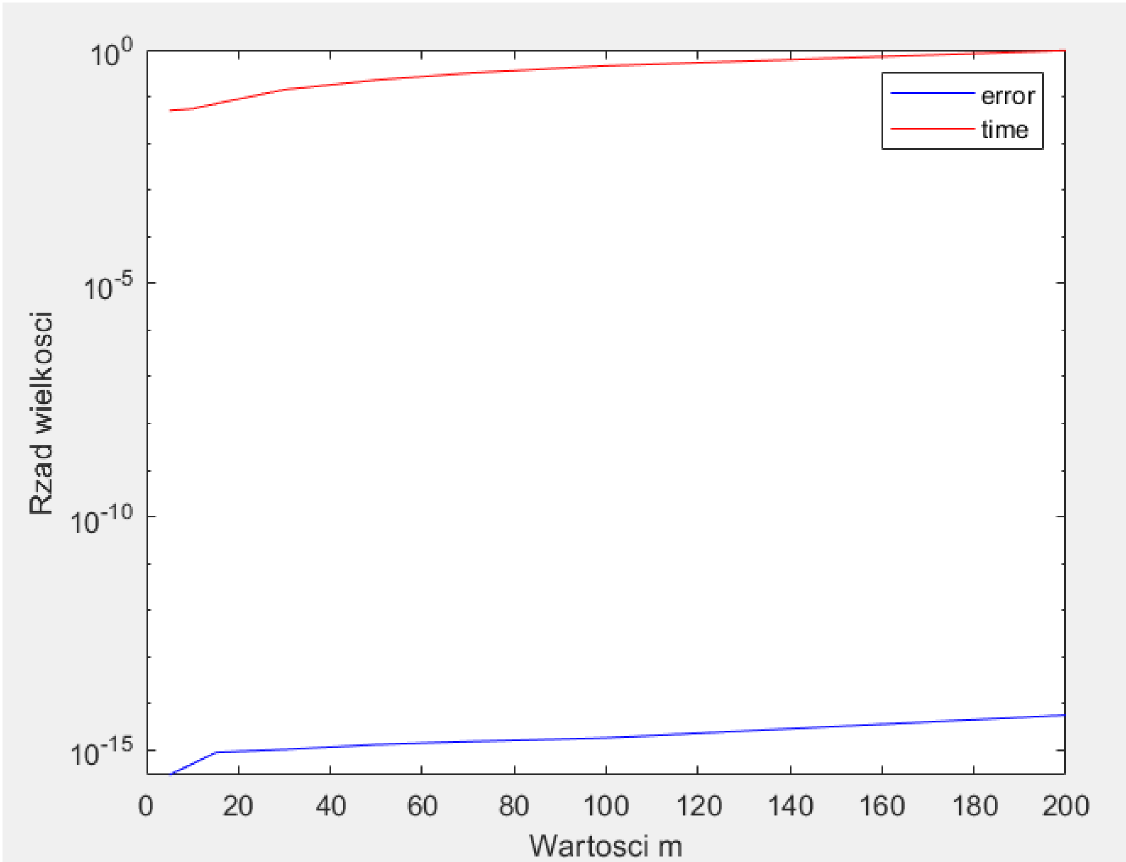
Wykresy f oraz f^* dla poszczególnych wartości m:






Podział przedziału na 50 punktow
Podział przedziału na 70 punktow
Podział przedziału na 100 punktow
Podział przedziału na 200 punktow
1.0000000000001
1.00000000000005
1
0.99999999999995
0.9999999999999
0
2
4
6
real
approx

Wykres błędu oraz czasu (sek.) w zależności od liczby punktów m:



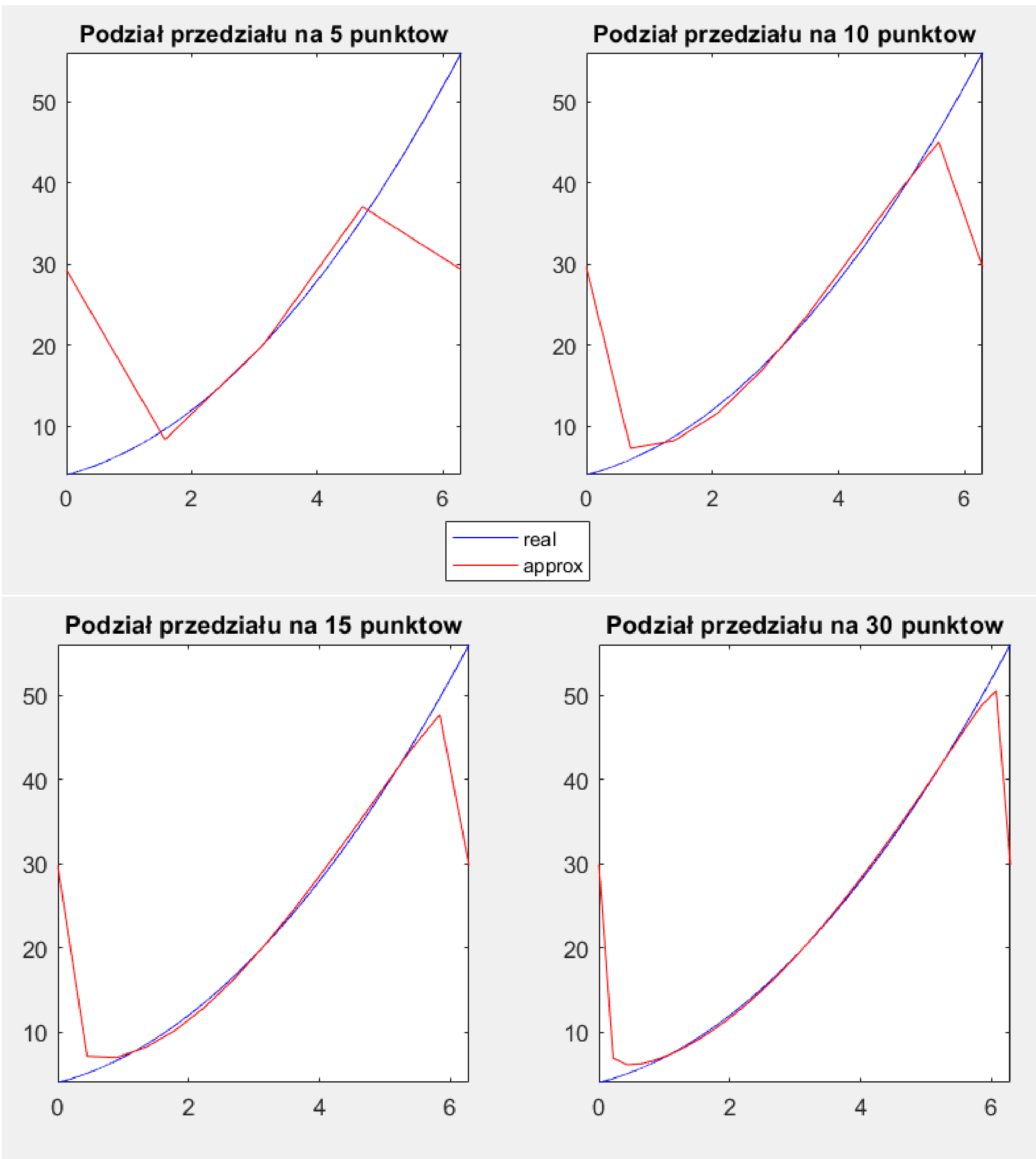
8. $f(x) = cos(2x) + 2sin^2(x)$

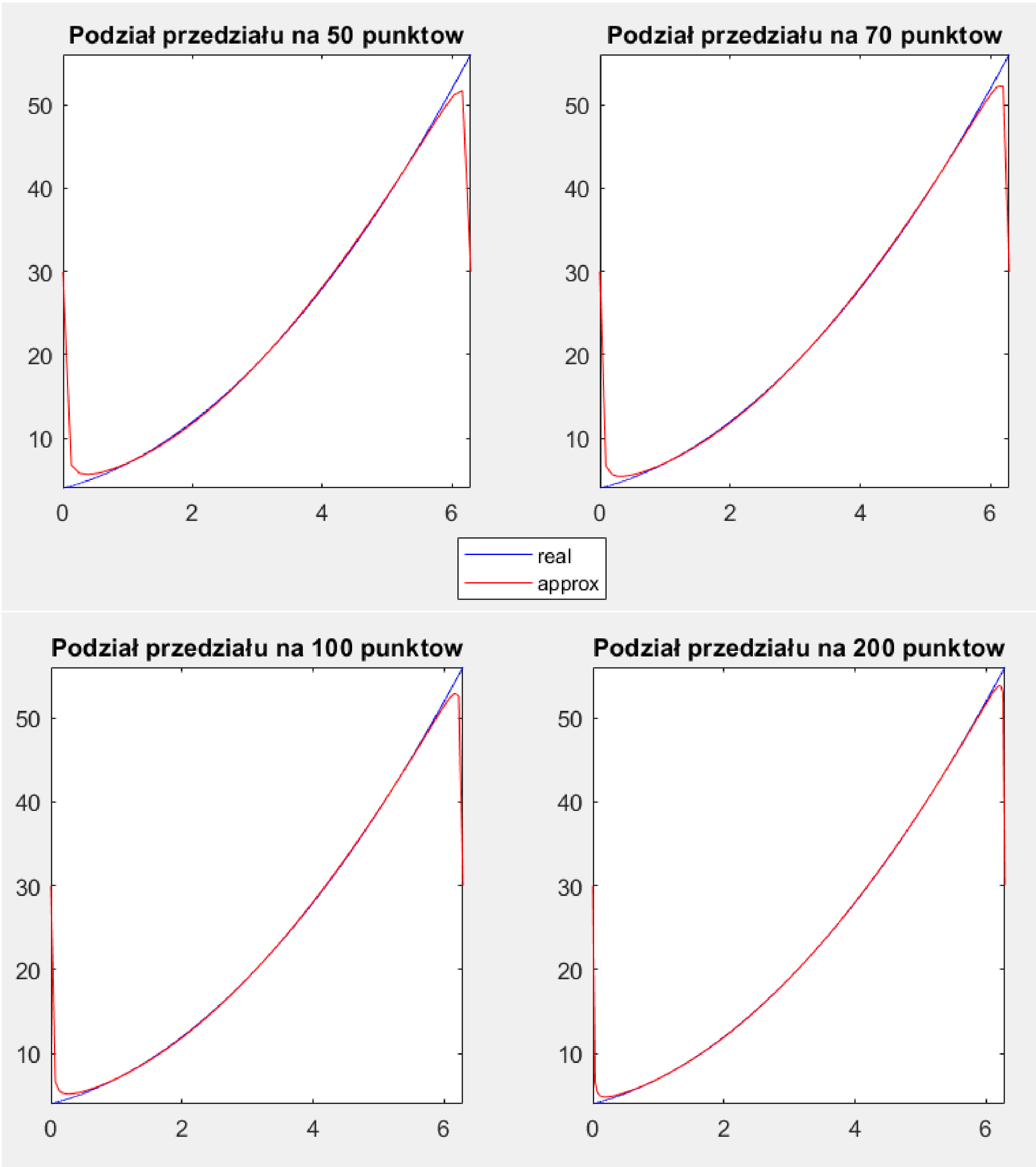
Tablicowanie wartości, przybliżenia i błędu oraz błąd średniokwadratowy:

```
Tablicowanie: [real/approx/err]

S =

    4.0000   30.0024   26.0024
    4.0641    6.5894    2.5252
    4.1303    5.4272    1.2969
    4.1984    5.0640    0.8656
    4.2685    4.9144    0.6459
    4.3407    4.8529    0.5123
```

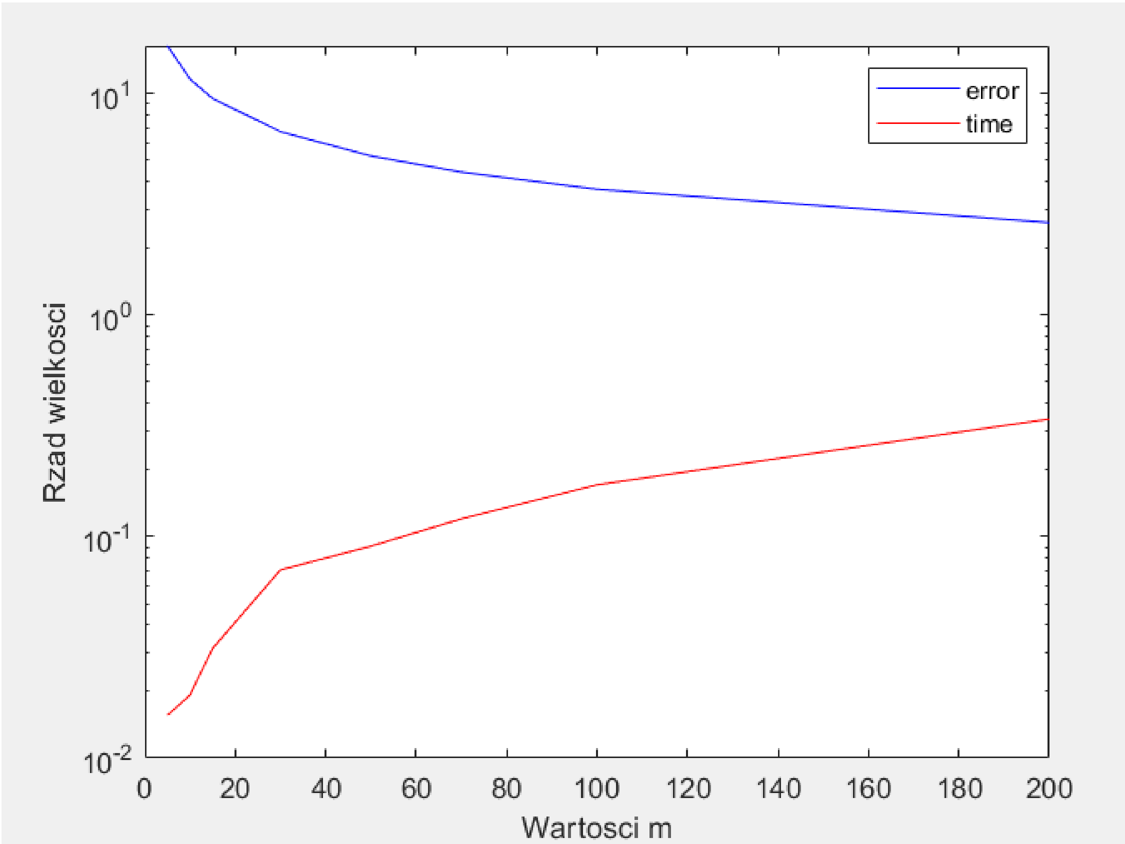
```
Błąd średniokwadratowy:

mean_square_error =

    2.6220
```

Wykresy f oraz f^* dla poszczególnych wartości m:




Podział przedziału na 50 punktow
Podział przedziału na 70 punktow
Podział przedziału na 100 punktow
Podział przedziału na 200 punktow
real
approx
0
2
4
6
10
20
30
40
50

Wykres błędu oraz czasu (sek.) w zależności od liczby punktów m:



5 Wnioski i interpretacja

W przypadku funkcji okresowych, dobrze określonych na przedziale I algorytm już dla niewielu punktów m aproksymuje wartości fukcji z dobrą dokładnością rzędu ok. 10^{-15}. Dalsze zwiększanie m prowadzi jednak do wzrostu czasu trwania działania programu, przy jednoczesnej stagnacji wartości błędu.

Dla funkcji okresowych posiadających asymptotę pionową w punkcie $x_0 \in I$ wartość aproksymacji jest niedokładna, co rzutuje na wartość błędu średniokwadratowego (wartość o charakterze globalnym) sprawiając, że jest on duży mimo dobrej aproksymacji funkcji poza punktem x_0. Przy badaniu takich funkcji, trzeba pamiętać o wyłączeniu z wyników punktów bliskich x_0.

Omawiany algorytm jest nieskuteczny dla funkcji nieokresowych, użycie szeregu Fouriera implikuje większą niedokładność przybliżenia funkcji. Warto mieć na uwadze, że często pozornie okresowe funkcje będące złożeniami funkcji trygonometrycznych mogą być na przykład funkcją stałą, tak jak wyrażenie $cos(2x) + 2sin^2(x)$ po zastosowaniu wzoru na $cos(2x)$ upraszcza się do jedynki trygonometrycznej. Dla tej funkcji aproksymacja naszym algorytmem również jest niedokładna.