Zadanie 1.

Przypomnieć „szkolne” ścisłe wartości funkcji $\sin x$ w przedziale $[0, \pi/2]$. Używając ich jako tabeli wartości wyprowadzić i zaprogramować algorytm przybliżający funkcję sinus:

- za pomocą interpolacji liniowej; można użyć własnej implementacji lub gotowej, np: GNU GSL [`sin1`]
- za pomocą wielomianu interpolującego (t.j. przechodzącego przez wszystkie punkty) [`sin2`]

Określić poprzez porównanie z wbudowaną funkcją sinus maksymalny błąd absolutny i względny.

Zadanie 2.

Powtórzyć Zadanie 1 zmieniając precyzję obliczeń numerycznych. Porównać dokładność i szybkość działania

Zadanie 3.

Na przybliżenie funkcji sinus nakładamy dodatkowy warunek, aby pochodne w zerze lub/i $\pi/2$ mają być takie same jak dla funkcji sinus. Aby to osiągnąć przybliżamy funkcję sinus:

- rozwinięciem Taylora w zerze [`sin3`],
- wielomianem który równocześnie przechodzi przez wszystkie punkty, a dodatkowo jego pochodne na końcach przedziału są zadane [`sin4`].

Porównać dokładność przybliżenia z funkcjami z Zad. 1.

Zadanie 4.

Korzystając z okresowości funkcji $\sin, \cos$ rozszerzyć dziedzinę powyższych funkcji na pełny zakres liczb zmiennoprzecinkowych. Sprawdzić poprawność działania dla bardzo dużych wartości argumentów.

Zadanie 5.

Korzystając z wyniku Zad. 4 zaimplementować funkcję tangens oraz zbadać jej dokładność.

andrzej.odrzywolek@uj.edu.pl

Zadanie 6.

W Zad. 1 rozłożenie punktów w których obliczamy wartości sinusa wybrano na podstawie wiedzy szkolnej. Czy dokładność przybliżenia zależy od rozłożenia punktów przy zachowaniu ich liczby? Jeżeli tak, jakie jest optymalne ich rozłożenie?

Wskazówka: użyj aproksymacji Czybyszewa lub/i wielomianów ortogonalnych

Zadanie 7.

W pewnych sytuacjach interpolacja/aproksymacja musi oprócz zadanej dokładności spełniać pewne warunki, których złamanie grozi katastrofalnymi skutkami obliczeniowymi lub programistycznymi. Dla przykładu, analizowane wcześniej przybliżenia funkcji sinus nigdy nie powinny dać wyniku większego niż 1.

Proszę przetestować przybliżenia, czy zawsze zachowują warunek $|\sin x| \leqslant 1$ i przedyskutować jak zagwarantować np: nieujemność przybliżenia.

Zadanie 8*.

Przybliżyć funkcję sinus w przedziale $[0, \pi/2]$ za pomocą innych niż wielomiany wyrażeń, np: funkcji wymiernych (zawierających dzielenie), lub zupełnie dowolnych z pewnego zbioru funkcji i operacji (np: pierwiastkowanie, potęgowanie, logarytm).

Zadanie 9*.

Najprostszym sposobem zwiększenia dokładności aproksymacji/interpolacji funkcji sinus wydaje się zwiększenie liczby punktów/stopnia wielomianu limitowane jedynie wielkością pamięci. Proszę zbadać zachowanie się algorytmu dla coraz większej liczby punktów/stopnia wielomianu, np: w postępie geometrycznym (potęgi dwójki) w praktyce. Jak zachowuje się dokładność, szybkość działania oraz zużycie pamięci. Co w praktyce nas ogranicza?