# Instrukcje ćwiczenia 1 MOWNIT.

## 1 Przygotowanie

### 1.1 Dane

- *f1(x)* oraz *f2(x)* to nr funkcji elementarnych z instukcji _1:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\arctan x$ | $\cos x$ | $\sin x$ | $\tan x$ | $\arccos x$ | $\arcsin x$ | $x^3$ | $e^x$ | $\ln x$ | $\cos x$ | $\sqrt{x}$ | $\sqrt[3]{x}$ | $x^2$ | $\cot x$ |

(Przykładowo: prowadzący wskazuje cyfry 2 i 9. Funkcja może być postaci $f(x) = \cos x \cdot \ln x$.)

  Funkcje na pewno mogą być połączone za pomocą operacji mnożenia "*" oraz prawdopodobnie innych operacji np. "+"

- *a,b,c,d* - wpisuje dokładnie to co dostaliśmy od profesora.

**Komentarz:**

1. Całe ćwiczenie 1 - 2 polega na storzeniu funckji *f* z podanych funkcji elementarnych.
2. Określamy przedział gładkości (funkcja ma być określona, bez osobliwości, ciągła ) *f* -> jakieś *x0* i *xn*
3. W pierwszym odpalamy funkcje *intpara.m* oraz *inttria.m* dla n = 10, 20, 50, 90. Wyniki (8 - 4 dla intpara i 4 dla inttria) rysujemy wykres x =log_10(błąd całkowania) y = log_10(n) oraz ustalamy stopnie bierzności (tangensy kierunkowe wykresów)
4. Błąd całowania obliczamy odejmując faktyczny wynik całkowania (użycie której kolwiek funkcji z n = 1000 lub wolfram)

# 2. Zadania

## 2.1 Zadanie 1 - 2

### 2.1.1 Instrukcja

1. **Zadanie 1**

2. Odpalamy matlaba/octave

3. Otwieramy Zad1_2.m

4. Odpalamy

5. Podajemy input vol 1:

   - *f* funkcja zadana od profesora zgodna z 1.1 w formacie matlabowym np.

   ```
   acos(x) * exp(x)
   ```

   - *x0* początek przedziału całkowania

   - *xN* koniec przedziału całkowania
   - Punkty [*x0*, *xN*] określają przedział całkowanie gdzie funkcja *f* musi być okeślona, bez osobliwości, ciągła oraz mieć miejsce zerowe jeśli go nie ma trzeba wymusić to poprzez dodanie stałej.

6. Program wypluje nam odpowiedni wykres do pliku zad1.png

7. W konsoli programu wyświetlą się informacje o współczynnikach zbierzności wykresów.

8. Wykres oraz wpsółczynniki wraz z funkcją i przedziałem należy umieścić w sprawozdaniu do ćwiczeń **(Koniec zadania 1)**.

9. **Zadanie 2**

10. W kolejnym kroku program poprosi nas o parametry związane z zadaniem drugim:

    - *eps* - wielkość akceptowalnego błędu
    - *x0* - punkt początkowy w algorytmie Newtona najczęściej 0, 1, początkek przedziału lub koniec
    - *df* - pochodna funkcji *f* polecam użyć wolframu bądź samamu wyliczyć, zgodna z matlabem.
    - *a* - początek przedziału dla algorytmu bisekcji oraz algorytmie siecznych
    - *b* - koniec przedziału dla algorytmu bisekcjii oraz algorytmie siecznych
    - **funckcja w przedziale *[a, b]* musi posiadać miejsce zerowe**

11. Program następnie wypluje plik zad2.png w którym znajdują się odpowiednie wykresy które należy umieścić w sprawozdaniu.

12. Program w konsoli wyświetli parametry wprowadzone dla zadania 2, je również należy dodać do sprawozdania.

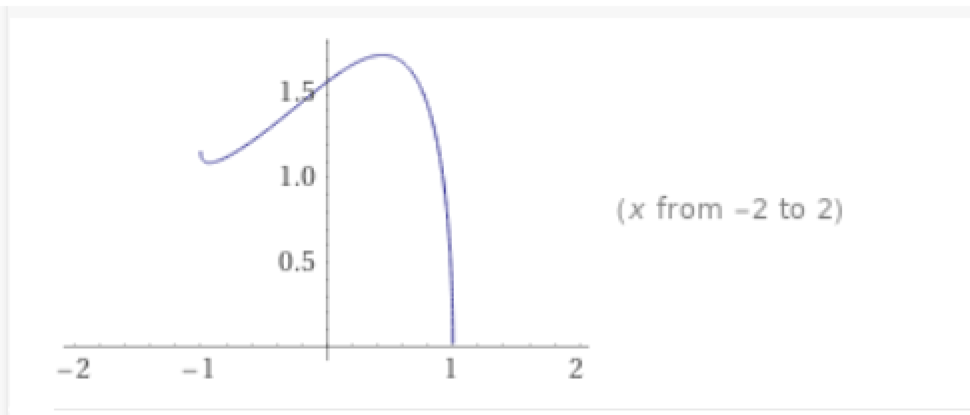
### 2.1.2 Przykład:

1. **Zadanie 1**

2. Dla przykładowej funkcji:

```
acos(x) * exp(x)
```

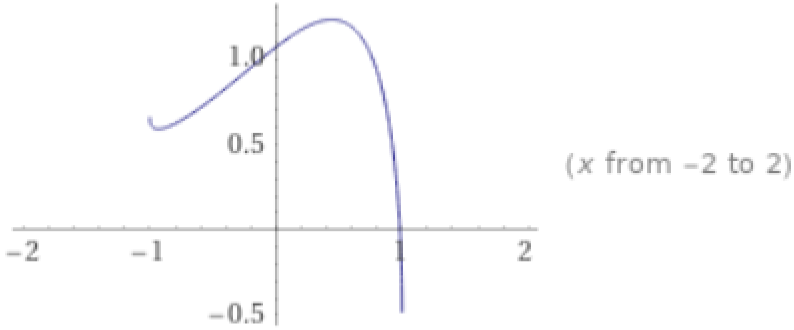
3. Wyznaczamy x0 oraz xN poprzez wpisanie funkcji do wolframu i analizie



4. Zauważamy że w przedziale od -0.9 do 1 funkcja jest ciągła jednak nie posiada miejsc zerowych 😦

5. Obniżamy funkcję poprzez odejmowanie 0.5 i otrzymujemy:

   ```
   acos(x) * exp(x) - 0.5
   ```

6. W doborze przedziału dobrze jest od razu pozbyć się wartości dla których pochodna ma wartość 0. Pochodną naszej funkcji jest (z wolframu)

   ```
   exp(x)*(-1/sqrt(1-pow2(x))+acos(x))
   ```

   dla 0 pochodna będzie mieć wartość 0 dlatego wyznaczamy przedział *[x0, xN]* jako *[0.1, 0.9]*

7. Odpalamy program wpisujemy funkcję *f*=acos(x) * exp(x) - 0.5, oraz zakres *[0.1, 0.9]*

8. dostajemy wykres zad1.png

9. w konsoli mamy tangesy kierunkowe z wykresu

10. **Zadanie 2**

11. Program prosi o dane:

    - x1 - najlepiej dać początek przedziału
    - df - wyliczona całka z wolframa exp(x)*(-1/sqrt(1-pow2(x))+acos(x))
    - przedział *[a, b]* w teori można podać normalny ale też można przepisać liczby *[x0, xN]*

12. Progam wygeneruje wykres zad2.png

13. program w konsoli wyświeli parametry

14. **Koniec Zadania 2**

## 3.1 Zadanie 3

### 3.1.1 Instrukcje

1. Dobieramy do *a,b,c,d* resztę macierzy, nasza wartość diagonalna ma być większa niż suma reszty liczb w rzędzie:

2. Macierz nie może być charakterystyczna to znaczy nie może być jednostkowa czy posiadać np. same 2
3. Odpalamy Zad3_4.m i wpisujemy macież w postaci:

$$A = \begin{bmatrix} 13 & 1 & 2 & 3 \\ 1 & 11 & 2 & 3 \\ 2 & 2 & -10 & 1 \\ 3 & 3 & 1 & 19 \end{bmatrix}.$$

```
[13,1,2,3;1,11,2,3;2,2,-10,1;3,3,1,19]
```

4. program wypluje w konsoli wszystkie potrzebne dane

## 4.1 Zadanie 4

1. Zadanie 4 używa tej samej macierzy co w przypadku zadania 3 dlatego oba zadania wykonają się po sobie
2. Należy podać wektor b 1x4 (każda linijka będzie innym wierszem) np.

```
1
2
3
4
```

3. Następnie będziemy musieli podać wektor wyniku równania

```
R = [0.0263419; 0.110835; -0.10835; 0.67992]
```

4. Najlepiej zdobyć wynik posługując się wolframem. Wpisujemy do niego formułe

```
LinearSolve[{{A11, A12, A13, A14},{A21, A22, A23, A24},{A31, A32, A33, A34},
{A41, A42, A43, A44}}, {b1,b2,b3,b4}]
np.
LinearSolve[{{13, 3, -3, 0},{1, -19, 0, 6},{0, 6, -9, 2},{1, 1, 2, 6}},
{1,2,3,4}]
```

5. Następnie program zapiszę nam na dyski obraz wykresu zad4.png
6. Koniec Ćw_1 Yey 😄