Liczenie całki oznaczonej złożoną metodą trapezów

1. Cel oraz opis metody

Metoda całkowania numerycznego poprzez metodę złożonych trapezów ma na celu przybliżenie wartości całki oznaczonej . Metoda ta polega na podzieleniu długości przedziału przez precyzję - to określa wysokość trapezu oraz obliczenie wartości na krańcach aktualnego podprzedziału - wartości te będą jego podstawami. Minusami tej metody jest niedoszacowanie (lub w niektórych skrajnych przypadkach przeszacowanie) całki oznaczonej. Metodę tą można określić wzorem

$$\sum_{i=0}^{precyzja} \frac{(a+b)*h}{2}$$

Gdzie:

$h = \frac{(start - koniec)}{precyzja}$

$a = \text{poczatek funckji} + h * n$

$b = \text{poczatek funckji} + \text{h} * (i + 1)$

2. Opis programu

Wybrana metoda

Odwrotna notacja polska

Aby obliczyć całkę potrzebuje obliczyć wartość jej funkcji na podprzedziałach, a więc dane wprowadzone przez użytkownika muszę odpowiednio zinterpretować. Podstawowym problemem jest kolejność wykonywania działań, bez używania zewnętrznych bibliotek jednym z prostszych sposobów jest zamiana działania na odwrotną notację polską(ONP, z ang. RPN). Zastosowałem do tego algorytm Shunting yard ze względu na jego prostotę oraz ogólną niezawodność. Algorytm ten polega na sprawdzeniu czy Aby program był w stanie obliczać podstawowe funkcję matematyczne takie jak sinus, tangens, logarytm naturalny czy pierwiastek trzeba było zmodyfikować algorytm ze względu że w podstawowej wersji działa on tylko na podstawowe operacje matematyczne, w tym celu dodałem dodatkowe "wagi" operatorów dla funkcji oraz potęgowania. Do przechowywania wyrażenia ONP użyłem obiektu typu vector co ułatwiło zarządzanie pamięcią programu.

Generalnie algorytm ten można zaprezentować w następujący sposób:

Algorytm składa się z ciągu symboli, stosu operatorów oraz wyjścia

1. Jeżeli symbol to liczba → na wyjście
2. Jeżeli stos jest pusty lub symbol jest nawiasem otwierającym → na stos
3. Jeżeli symbol jest nawiasem zamykającym → wypisuj symbole ze stosu na wyjście do napotkania nawiasu otwierającego
4. Jeżeli symbol jest operatorem oraz ma priorytet wyższy niż element na górze stosu, wrzuć go na stos
5. Jeżeli symbol jest operatorem oraz ma priorytet niższy niż ten na górze stosu → wypisuj do napotkania operatora o wyższym lub równym priorytecie lub wyczerpania stosu→ wrzuć go na stos

Priorytety operatorów mają typowe priorytety dla matematycznego priorytetu w kolejności wykonywania działań.

Metoda sprawdzania czy część kodu jest zdefiniowaną funkcją matematyczną

```
map <string,double(*)(double)> initFunc(){
    map<string, double(*)(double)> func;
    func["sin"] = sin;
    func["cos"] = cos;
    func["tg"] =  tan;
    func["ctg"] = [](double x){return 1/tan(x); };
    func["pi"] = [](double){return M_PI;};
    func["exp"] = exp;
    func["log"] = log;
    func["sqrt"] = sqrt;
    return func;

}
```

W programie zdefiniowane zostały obsługiwane funkcję matematyczne, zostały one zdefiniowane w mapie przyporządkowującej danym typu string wskaźnik funkcji matematycznej Samo sprawdzanie polega na liczeniu ile znaków po sobie to litera do napotkania operatora matematycznego, wtedy program szuka w mapie tego ciągu oraz wrzuca na stos operatorów ten ciąg znaków. Mapa ma ułatwić tym samym wywołanie funkcji podczas odczytywania wartości wyrażenia ONP w zależności od jednego parametru. Sprowadziło to pisanie instrukcji switch do zadeklarowania mapy oraz wywołania funkcji poprzez bibliotekę functional oraz dodatkową funkcję:

```
double eval(function<double(double)> func, double x) {
    return func(x);
}
```

Która przyjmuje wskaźnik funkcji matematycznej func oraz parametr x(z tego względu że wszystkie funkcję są jednoparametrowe) po czym zwraca jego wartość funkcji func(x).

Ograniczenia programu z przykładami

Program oferuję obliczenie przybliżonej wartości całki oznaczonej, niemniej należy przyjąć odpowiednie założenia:

- Program zwróci wartość -inf/+inf jeżeli otrzyma funkcję która na podanym przedziale nie istnieje, tzn. nie jest ciągła - świetnym przykładem mógłby być wynik programu dla funkcji ctg(x) na przedziale <0,1>, jako że funkcja ta nie istnieje w zerze, wynik będzie wynosił inf - nieskończoność.

```
Program liczący całkę oznaczoną złożoną metodą trapezów
1-Wpisz funkcje do obliczenia całki
2-Dane testowe pobierane z pliku
1
Wpisz funkcje którą chcesz obliczyć
ctg(x)
Wpisz początek przedziału
0
Wpisz koniec przedziału
1
Wpisz precyzję(ilość trapezów)
1000
ctg(x) Całka dla podanej funkcji wynosi inf
```

- Program nie obsługuje błędów wejściowych, tzn. jeżeli między funkcją sin a jej wartością np. sin(x) wstawimy znak operacji arytmetycznej program nie zadziała, dlatego sin*(x) nie jest poprawnym zapisem.
- Program nie sprawdza czy w funkcji istnieją niezdefiniowane ciągi znaków, czyli jeżeli chcielibyśmy obliczyć całkę funkcji arg(x), program nie zadziała oraz nie pozwoli poprawić danych.

Możliwości programu

Program jest w stanie obliczyć funkcję dla zdefiniowanych funkcji matematycznych, przy zachowaniu poprawnego zapisu takiego jak:

- -2x^sin(x)
- x^6+ctg(-x^exp(10))/tg(-x^exp(10))
- ctg(x)+ln(exp(x^6))+cos(pi)-sin(x)^2.5

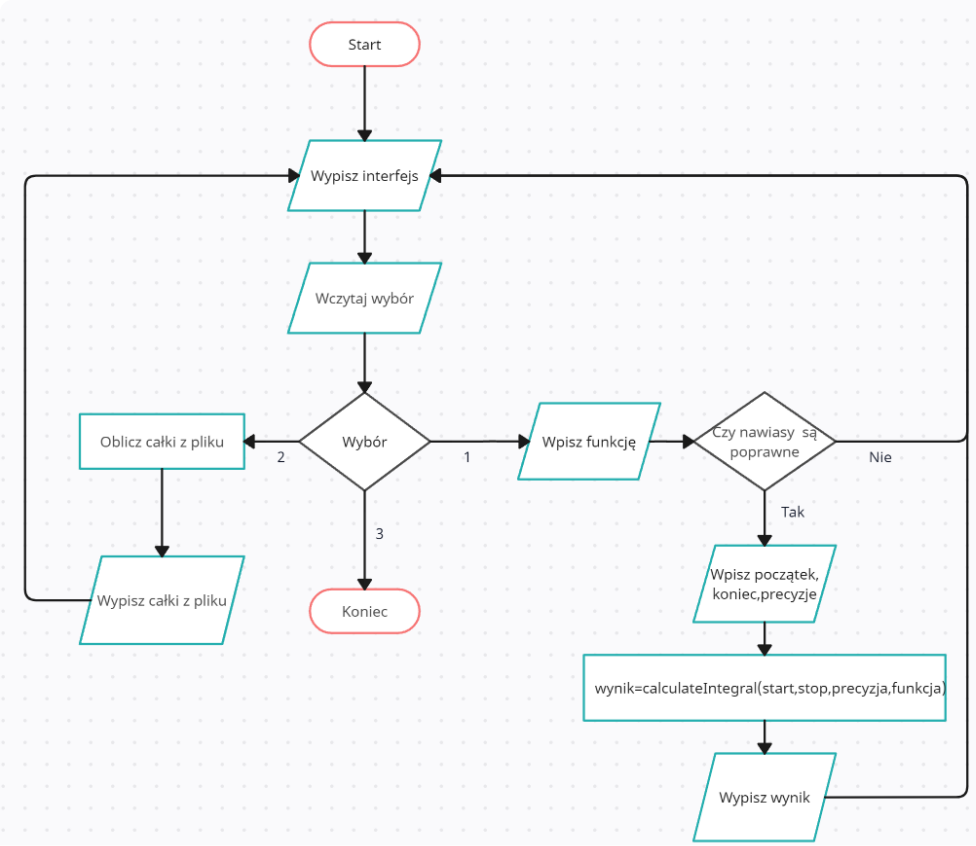
Czyli:

1. Przed funkcją lub symbolem x może stać **liczba** rzeczywista wtedy program dostawi znak mnożenia oraz "pod spodem" przeliczy wartość
2. Między funkcjami oraz między znakiem x musi stać znak operacji arytmetycznej.
3. W funkcji znak pi zostanie zamieniony na M_PI

Mając to na uwadze program obliczy przybliżoną całkę oznaczoną każdej funkcji matematycznej która jest zdefiniowana oraz na podanym przez użytkownika przedziale jej całka jest zbieżna.

Schematy blokowe:

Schemat działania całego programu



Schemat działania funkcji liczącej całkę: