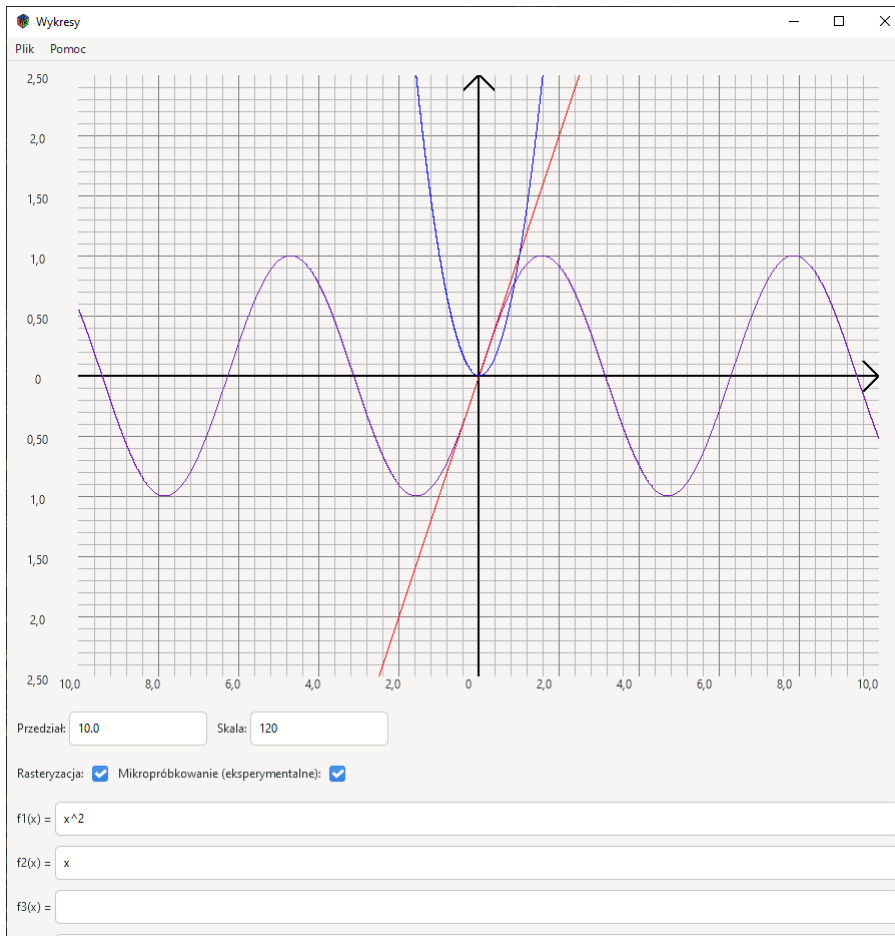


Program Wykresy - instrukcja obsługi

Wiktor Ogrodnik

7 lutego 2021

1 Opis programu



Program wykresy służy do rysowania wykresów jednej zmiennej w zadanym przedziale. Wykres rysuje się w górnej części programu po naciśnięciu przycisku wprowadź. Program posiada zaawansowany interpreter obsługujący operatory matematyczne, funkcje takie jak sinus czy logarytm, kolejność wykonywania działań, dowolną ilość nawiasów oraz stałe matematyczne. Moduł wyświetlający wykres na ekranie ma zaimplementowany system prostej rasteryzacji, dzięki czemu nawet wykresy skomplikowanych funkcji prezentują się w przystępny sposób. Wspierane systemy operacyjne to Windows 10 i Linux.

1.1 Technologie

Technologie użyte do stworzenia programu to:

- Język C11 (GCC 10.2),
- Biblioteka GTK+ 3.24,
- Program make,
- Visual Studio Code.

2 Kompilacja ze źródła i uruchamianie

2.1 Windows

- pobrać i zainstalować MSYS2, zgodnie z zaleceniami ze strony <https://www.msys2.org/>;
- pobrać i zainstalować bibliotekę GTK (GTK+ 3.24) zgodnie z instrukcją na stronie <https://www.gtk.org/docs/installations/windows/>;
- jeżeli nie są zainstalowane, przez menedżer pakietów MSYS2 pobrać kompilator mingw i program make;
- poprzez konsolę MSYS2 MinGW 64-bit (lub 32-bit) odnaleźć folder z kodem źródłowym i wpisać polecenie 'make';
- uruchomić program poleceniem ./wykresy.exe.

2.2 Linux

- poprzez wbudowany w dystrybucje menedżer pakietów pobrać i zainstalować pakiet 'libgtk-3-dev' wraz z pakietami zależnymi;
- jeżeli nie są zainstalowane, przez wbudowany w dystrybucje menedżer pakietów pobrać kompilator gcc i program make;
- w terminalu odnaleźć folder z kodem źródłowym i wpisać polecenie 'make';
- uruchomić program poleceniem ./wykresy.

3 Równania matematyczne

W tej sekcji opiszę jakie możliwości daje interpreter poleceń matematycznych wbudowany w program i jak konstruować zapytania. W programie można wpisać na raz 4 równania, które będą wyświetlane jednocześnie na wykresie.

3.1 Składnia

- liczby
- funkcje
- stałe
- zmienna
- nawiasy
- operatory binarne

3.1.1 Liczba

Liczba to stała zapisana za pomocą cyfr oraz kropki i przecinka (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0., ..). Liczby to np. 10, 2,5, 10.5, 0, 11, 123, itd. Przed liczbą może znajdować się '+' lub '-' oznaczający znak liczby, np: +10, -0.5.

3.1.2 Funkcje

Funkcje składają się z nazwy, nawiasów okrągłych oraz przyjmują jeden argument. Np.: $\sin(<\text{wyrażenie}>)$, $\log(<\text{wyrażenie}>)$, itd.

W programie dostępne są następujące funkcje:

- $\sin(x)$
- $\cos(x)$
- $\tan(x)$ - inna forma: $\text{tg}(x)$
- $\cot(x)$ - inna forma: $\text{ctg}(x)$
- $\ln(x)$ - logarym naturalny, również $\log(x)$

- $\log_2(x)$ - logarytm dwójkowy
- $\log_{10}(x)$ - logarytm dziesiętny
- \sqrt{x} - pierwiastek kwadratowy
- $\text{abs}(x)$ - wartość bezwzględna
- $\sinh(x)$ - sinus hiperboliczny
- $\cosh(x)$ - cosinus hiperboliczny
- $\tanh(x)$ - tangens hiperboliczny, również $\text{tgh}(x)$
- $\text{floor}(x)$ - część całkowita, tak zwana podłoga
- $\text{ceil}(x)$ - część całkowita (zaokrąglenie w górę), tak zwany sufit
- $\text{asin}(x)$ - arcus sinus, również $\arcsin(x)$
- $\text{acos}(x)$ - arcus cosinus, również $\arccos(x)$
- $\text{atan}(x)$ - arcus tangens, również $\arctan(x)$ i $\text{arctg}(x)$
- $\exp(x)$ - funkcja eksponentyjalna (e^x)

3.1.3 Stałe

W programie zdefiniowane są 3 specjalne stałe, można je wpisywać tak samo jak liczby:

- e - e
- π - π
- ϕ - ϕ

3.1.4 Zmienna

Jako, że program rysuje funkcje jednej zmiennej, to dostępna jest tylko jedna zmienna - 'x'. Interpreter pod zmienną x podstawia odpowiednią wartość liczbową, w zależności od części wykresu, która jest rysowana.

Uwaga! Użycie innej litery niż x jako zmiennej nie zadziała!

3.1.5 Nawiasy

Dostępne są 4 rodzaje nawiasów:

- Nawiasy okrągłe '()' - zwykłe nawiasy bez specjalnych funkcji, wyrażenie w tych nawiasach wykonywane jest w pierwszej kolejności. Dzięki nim można wpływać na kolejność wykonywania działań.
- Nawiasy typu '||' - posiadają funkcje nawiasów okrągłych, a dodatkowo obliczają wartość bezwzględną. Równoważne byłoby użycie funkcji $\text{abs}(<\text{wyrażenie}>)$.
- Nawiasy kwadratowe '[]' - oprócz tego, że wyrażenie w nawiasach wykonywane jest w pierwszej kolejności, to zwracana jest część całkowita tego wyrażenia. Równoważne byłoby użycie funkcji $\text{floor}(<\text{wyrażenie}>)$.
- Nawiasy klamrowe '{}' - wyrażenie w nawiasach wykonywane jest w pierwszej kolejności i zwracana jest część ułamkowa wyrażenia.

3.1.6 Operatory binarne

W programie jest 5 operatorów binarnych, przyjmujących dwa argumenty, jeden z prawej, drugi z lewej strony operatora.

Dostępne operatory:

- $x+y$ - dodawanie $x + y$
- $x-y$ - odejmowanie $x - y$
- $x*y$ - mnożenie $x * y$
- x/y - dzielenie x/y
- x^y - potęgowanie x^y

Operatory mają różne priorytety. Najpierw wykonywane jest potęgowanie, potem mnożenie i dzielenie, a na końcu dodawanie i odejmowanie.

Jeżeli operatory mają ten sam priorytet wykonywane są od lewej do prawej, wyjątkiem jest potęgowanie, które wykonuje się w drugą stronę.

Polecenie `'x^y^z'` jest interpretowane jako x^{y^z} . Ale `'x^y^z*a'` jest interpretowane jako $x^{y^z} \times a$. Dlatego zalecane jest stosowanie nawiasów, żeby uniknąć nieporozumień.

Czasem niewymagane jest pisanie znaku mnożenia. Np. `'2x'` interpretowane jest jako `'2 * x'`, `2(x + 3)` interpretowane jest jako `2*(x+3)`.

Jako `x` i `y` można podstawić zmienną, stałą, liczbę, wyrażenie w nawiasach lub funkcję.

3.2 Semantyka

Poprawne wyrażenia matematyczne w programie:

- Każda liczba, stała i zmienna jest poprawnym wyrażeniem
- `<wyrażenie>` jest poprawnym wyrażeniem
- `<wyrażenie> <op> <wyrażenie>` jest poprawnym wyrażeniem, gdzie `<op>` to `'+'`, `'-'`, `'*'`, `'/'` lub `'^'`
- `nazwafunkcji(<wyrażenie>)` jest poprawnym wyrażeniem
- `(<wyrażenie>)` jest poprawnym wyrażeniem (gdzie `'()'` to dowolne nawiasy)

Wszystkie wyrażenia matematyczne można skonstruować według powyższej zasady.

W zapytaniach można używać dowolnej liczby spacji lub nie używać jej wcale, wyjątkiem są spacje wewnątrz liczby, nazwy funkcji lub stałej, które są niedozwolone.

3.3 Przykłady równań

W tym dziale zawarłem konkretne przykłady funkcji jednej zmiennej i wyjaśnienie jak będą zinterpretowane przez program.

- `2 + x => 2 + x`
- `sin(x^2/10+5^2^(x)) => sin($\frac{x^2}{10} + 5^{2^x}$)`
- `cos(x)^2 - 10/x*2 => $\cos^2(x) - \frac{10}{x} * 2$`
- `cos(x)^2 - 10/(x*2) => $\cos^2(x) - \frac{10}{2x}$`
- `10 + (2 + 3) - floor(x^2)^2 => $10 + (2 + 3) - \text{floor}^2(x^2)$`

4 Ustawienia

W programie znajduje się kilka ustawień, którymi można zarządzać jak funkcje mają być wyświetlane na wykresie.

4.1 Przedział

Pole tekstowe przedział przyjmuje jedną liczbę (może być to liczba całkowita lub liczba ze skończonym rozwinięciem dziesiętnym). Po wpisaniu 10.0 funkcja będzie rysowana na przedziale od -10 do 10.

4.2 Skala

Pole tekstowe skala przyjmuje jedną liczbę (może być to liczba całkowita lub liczba ze skończonym rozwinięciem dziesiętnym). Skala odpowiada za przeciwdziedzinę na jakiej rysowana będzie funkcja. Przy skali 1.0 najmniejsza wartość jaką można zobaczyć to $f(x) = -300$, a największa $+300$, przy skali 2.0 przedział to $(-150, 150)$, przy domyślnej skali 30.0 przedział to $(-10, 10)$.

4.3 Rasteryzacja i Mikropróbkowanie

Rasteryzacja odpowiada za łączenie kropek na wykresie, tak żeby wyświetlany wykres był ciągłą linią. Opcja ta zalecana jest gdy wykres szybko rośnie lub maleje, bez rasteryzacji ciężko byłoby zaobserwować przebieg funkcji. Uwaga! Rasteryzacja w punktach nieciągłości może zachowywać się nieprzewidywalnie, np. dla wyrażenia $\text{floor}(x)$ funkcjonalność będzie rysowała pionowe linie w punktach $f(k)$, gdzie k jest liczbą całkowitą, co nie powinno mieć miejsca. Rasteryzacja ma mały wpływ na wydajność i jest włączona domyślnie.

Mikropróbkowanie działa tylko przy włączonej rasteryzacji. Funkcjonalność wykonuje dodatkowe testy w punktach nieciągłości i decyduje czy rysować linie łączącą kropki czy nie. Funkcjonalność jest eksperymentalna, ale powinna lepiej odzwierciedlać wykresy skomplikowanych funkcji. Mikropróbkowanie ma duży wpływ na wydajność i jest domyślnie wyłączone.

5 Przewodnik po kodzie programu

W programie znajdują się następujące moduły

- Inteperter poleceń matematycznych w plikach `eval.h` i `eval.c`,
- Moduł wyświetlający wykres w plikach `draw.h` i `draw.c`,
- Moduł rysujący interfejs w plikach `ui.h` i `ui.c`,
- Moduł zawierający konwertery pomiędzy typami w C w plikach `convert.h` i `convert.c`,
- Używane struktury zdefiniowane są w pliku `types.h`.

Wyjaśnienie działania poszczególnych funkcji jest wyjaśnione w komentarzach wewnątrz programu.

6 Dalszy rozwój

Po zakończeniu projektu planuję dalsze prace nad programem. Oprócz rozwoju samego kodu, gdzie planuję migrację do GTK 4 oraz wykorzystywanie XML i CSS przy rysowaniu interfejsu, zaplanowane są również następujące funkcjonalności:

- Wsparcie dla funkcji o kilku argumentach, np. logarytm dowolnej podstawy;
- Rysowanie wykresu pochodnej;
- Zapisywanie wykresu do pliku `.bmp`;
- Zapis i odczyt funkcji z pliku tekstowego;
- Parametry uruchamiania programu z konsoli;
- Menu ustawień i "Co nowego";
- Konfiguracja własnych stałych matematycznych;
- Moduł automatycznych aktualizacji.

7 Licencjonowanie

Program wykresy będzie darmowym oprogramowaniem open-source na licencji MIT. Po oddaniu projektu kod zostanie otwarty i będzie dostępny w publicznym repozytorium na GitHubie.