

# Podstawy teorii rakietowej oraz struktura aplikacji w Haskell

Krzysztof Magnuszewski

# Wprowadzenie do teorii rakietowej

- **Podstawowe zasady:** Teoria rakietowa opiera się na trzech zasadach dynamiki Newtona, w szczególności na trzeciej zasadzie: Dla każdej akcji istnieje równa i przeciwnie skierowana reakcja.
- **Znaczenie w dzisiejszych czasach:** Znajomość zasad ruchu rakietowego jest kluczowa dla współczesnych lotów kosmicznych, eksploracji przestrzeni kosmicznej oraz rozwoju technologii rakietowych.



Photo by Chris Boyer on Unsplash

# Podstawowe wzory i ich zastosowania

- **Równanie Ciołkowskiego:**  $\Delta v = v_e \ln \frac{m_0}{m_f}$

gdzie:  $\Delta v$ - zmiana prędkości rakiety,  
 $v_e$ - efektywna prędkość wylotowa,  
 $m_0$ - masa początkowa,  
 $m_f$ - masa końcowa.

- **Znaczenie wzoru:** Równanie to pozwala obliczyć wymaganą ilość paliwa, uwzględniając masę rakiety oraz specyfikację silnika.

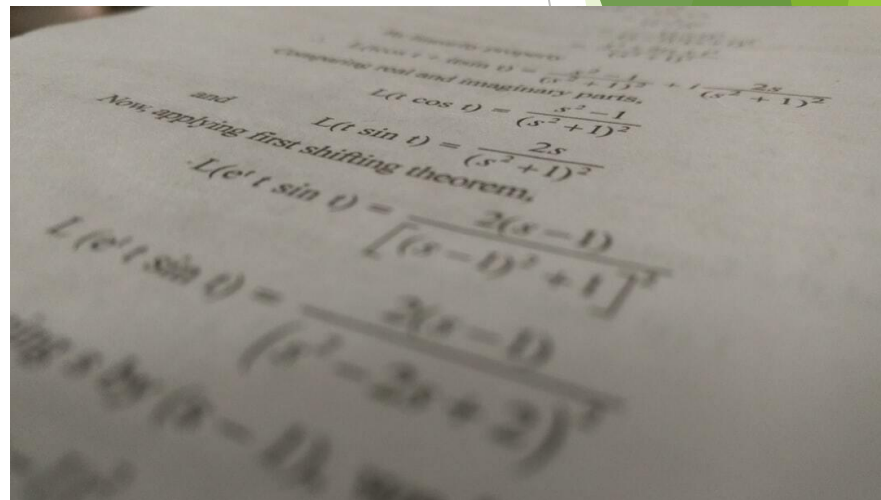


Photo by Ravi Koriya on Unsplash

# Struktura programu

- **Moduł główny:** Zarządza GUI i obsługą zdarzeń użytkownika. Importuje funkcje z innych modułów.
- **RocketThrust:** Zawiera funkcje do obliczeń związanych z ciągiem rakietowym i zużyciem paliwa.
- **GraphGenerator:** Generuje wykresy wizualizujące wyniki obliczeń.

```
rocket-thrust-calculator/  
├── app/  
│   └── Main.hs  
├── src/  
│   ├── RocketThrust.hs  
│   └── GraphGenerator.hs  
└── rocket-thrust-calculator.cabal
```

# Kluczowe funkcje i moduły

- **obliczPaliwo:** Funkcja z modułu RocketThrust, która oblicza zapotrzebowanie na paliwo na podstawie masy, ISP i delta V.

```
obliczPaliwo :: Double -> Double -> Double -> Double -> Double
obliczPaliwo masa isp deltaV szerokoscGeograficzna =
    masa * (1 - exp (-deltaV / efektywnaPredkoscWyloczniajaca isp przyspieszenieGrawitacyjne))
```

- **dostosujDeltaV:** Funkcja z modułu RocketThrust, która dostosowuje wartość delta V uwzględniając obrót Ziemi.

```
dostosujDeltaV :: Double -> Double -> Double
dostosujDeltaV deltaV szerokoscGeograficzna =
    deltaV - (predkoscObrotuZiemi * cos (szerokoscGeograficzna * pi / 180))
```

- **generujWykres:** Funkcja z modułu GraphGenerator, która generuje wykres prędkości rakiety w zależności od wysokości.

```
generujWykres :: [(Double, Double)] -> IO ()
generujWykres punkty = toFile def "predkosc_vs_wysokosc.png" $ do
    layout_title .= "Prędkość Rakiety vs Wysokość"
    plot (line "predkosc" [punkty])
```

# Narzędzia i biblioteki

- **Threepenny-GUI:** Biblioteka GUI dla języka Haskell, która umożliwia tworzenie interaktywnych aplikacji webowych.
- **Chart:** Biblioteka służąca do generowania różnych rodzajów wykresów w języku Haskell.
- **Data.Default.Class:** Biblioteka do ustawiania wartości domyślnych w Haskell.
- **Containers:** Biblioteka zawierająca różne struktury danych, takie jak listy, mapy, zbory, etc.
- **Cairo:** Biblioteka graficzna, która umożliwia rysowanie grafiki wektorowej.
- **Text:** Biblioteka służąca do manipulacji i obsługi tekstów w języku Haskell.

↓ **MSYS:** Zestaw narzędzi, które dostarczają podobne środowisko do Unixa w systemie Windows. Zapewnia konsolę, narzędzia do budowy oprogramowania, takie jak make, oraz biblioteki i narzędzia programistyczne.

↓ **Cabal:** Narzędzie do budowania i zarządzania pakietami w języku Haskell. Umożliwia m.in. budowanie i instalację pakietów.

↓ **Notepad++:** Popularny edytor tekstu dla systemu Windows.

↓ **CMD:** Klasyczne narzędzie w systemach Windows służące do wykonywania poleceń wiersza poleceń.

# Uruchomienie i działanie aplikacji

Aplikację uruchamiamy za pomocą pliku .exe, który znajduje się w :  
dist-newstyle\build\x86\_64-windows\ghc-9.8.1\rocket-thrust-calculator-0.1.0.0\x\rocket-thrust-calculator\build\rocket-thrust-calculator\  
po zbudowaniu projektu komendą: cabal build, a następnie użyciu  
komendy: cabal run w cmd lub innym narzędziu. Można też użyć STACK.

```
Administrator: Wiersz polecenia - cabal run
C:\KalkulatorStartuRakiety-main>cabal build
Up to date
C:\KalkulatorStartuRakiety-main>cabal run
Listening on http://127.0.0.1:8023
```

Następnie wchodzimy na podany adres w przeglądarce internetowej

< > ↺ VPN 🌐 127.0.0.1:8023

Masa (kg)

ISP (s)

Delta V (m/s)

Przylądek Canaveral

Oblicz

Po wejściu na podany adres powinien nam się wyświetlić formularz po którego wypełnieniu i kliknięciu przycisku oblicz wyświetli nam się wynik - ilość wymaganego paliwa do startu rakiety.

10000

300

5000

Port Kosmiczny Gujarat

Oblicz

Wymagane paliwo: 6894.95886100096 kg

Gdy już mamy wynik w głównym folderze aplikacji pojawi się plik predkosc\_vs\_wysokosc.png z wykresem przedstawiającym wynik.

