# Optymalizacja – poszukiwanie ekstremów funkcji metodą złotego podziału

Michał Saturczak
Informatyka Stosowana, Akademia Górniczo-Hutnicza
Metody Numeryczne

26 kwietnia 2024

# 1 Wstęp teoretyczny

Metoda złotego podziału jest techniką optymalizacji, która pozwala na znalezienie minimum funkcji jednej zmiennej. Proces ten odbywa się poprzez iteracyjne zawężanie przedziału poszukiwań, aż do osiągnięcia założonej krańcowej szerokości przedziału. Poniżej przedstawiono kolejne kroki tej metody:

1. Rozpoczynamy od wyznaczenia przedziału, w którym rozpocznie się poszukiwanie minimum, oznaczanego jako $[x_a, x_b]$.
2. Następnie przedział zostaje podzielony na trzy części poprzez wyznaczenie dwóch punktów: $x_1 = x_a + \lambda_1(x_b - x_a)$ i $x_2 = x_a + \lambda_2(x_b - x_a)$, gdzie $\lambda_1 = \frac{r}{2}$, $\lambda_2 = r$, a $r = \frac{\sqrt{5}-1}{2}$.
3. Zawężamy przedział poprzez podstawienie jednej wartości brzegowej wartościami $x_1$ lub $x_2$: jeżeli $f(x_1) > f(x_2)$ to $x_a = x_1$, jeżeli $f(x_1) < f(x_2)$ to $x_b = x_2$.
4. Punkty 2 i 3 powtarzamy aż do osiągnięcia założonej krańcowej szerokości przedziału, czyli do momentu, kiedy $|x_b - x_a| < \varepsilon$.
5. Wynikiem końcowym jest środek otrzymanego na końcu przedziału.

Metoda złotego podziału wykorzystuje własności proporcji złotego podziału, aby optymalizować proces poszukiwania minimum. Dzięki temu jest to efektywna metoda optymalizacji, szczególnie przydatna w przypadkach, gdy potrzebujemy znaleźć minimum funkcji jednej zmiennej.

# 2 Problem

## 2.1 Rozwiązanie problemu

- Napisano program w języku C++ do rozwiązania problemów związanych ze złotym podziałem.
- Zapisano do sześciu skoroszytów dane odpowiednio dla każdego podpunktu zadania.
- Zrobiono wykresy przedstawiające lot pocisków oraz wykresy 30 iteracji logarytmu dla metody złotego podziału i trójpodziału.

## 2.2 Funkcje w języku C++ użyte do rozwiązania problemu (bez funkcji podanych w poleceniu)

```
double negative_wziuuum(double alpha)
{
  std::ofstream empty_file;
  return -wziuuum(alpha, empty_file);
}

double wziuuum_i_bum(double alfa, double distance, double elevation, std::ofstream &file)
{

  double x, y, v, Time;
  double pi, g, Cx, diam, Sm, M, T, P0;
  double dt, v_tmp;

  pi = 3.14159265358979;
  g = 9.807;

  Cx = 0.187;

  diam = 155;
  Sm = pi * std::pow((diam / 2 / 1000), 2);

  M = 48;
  v = 958.0;

  T = 273;
  P0 = 101300;

  alfa = alfa / 180 * pi;

  x = 0.0;
  y = 1.0E-5;
  Time = 0.0;

  if (file.is_open())
  {
    file << "x\ty\n";
    file << x << "\t" << y << "\n";
  }

  while (y > 0.0 && x < distance)
  {
    dt = 1.0 / v;

    x += v * std::cos(alfa) * dt;
    y += v * std::sin(alfa) * dt;

    v_tmp = v + (-F_drag(v, Cx, Sm, y, T, P0) / M - g * std::sin(alfa)) * dt;
    alfa -= g * std::cos(alfa) / v * dt;
    v = v_tmp;
    Time += dt;

    if (file.is_open())
    {
      file << x << "\t" << y << "\n";
    }
  }
```

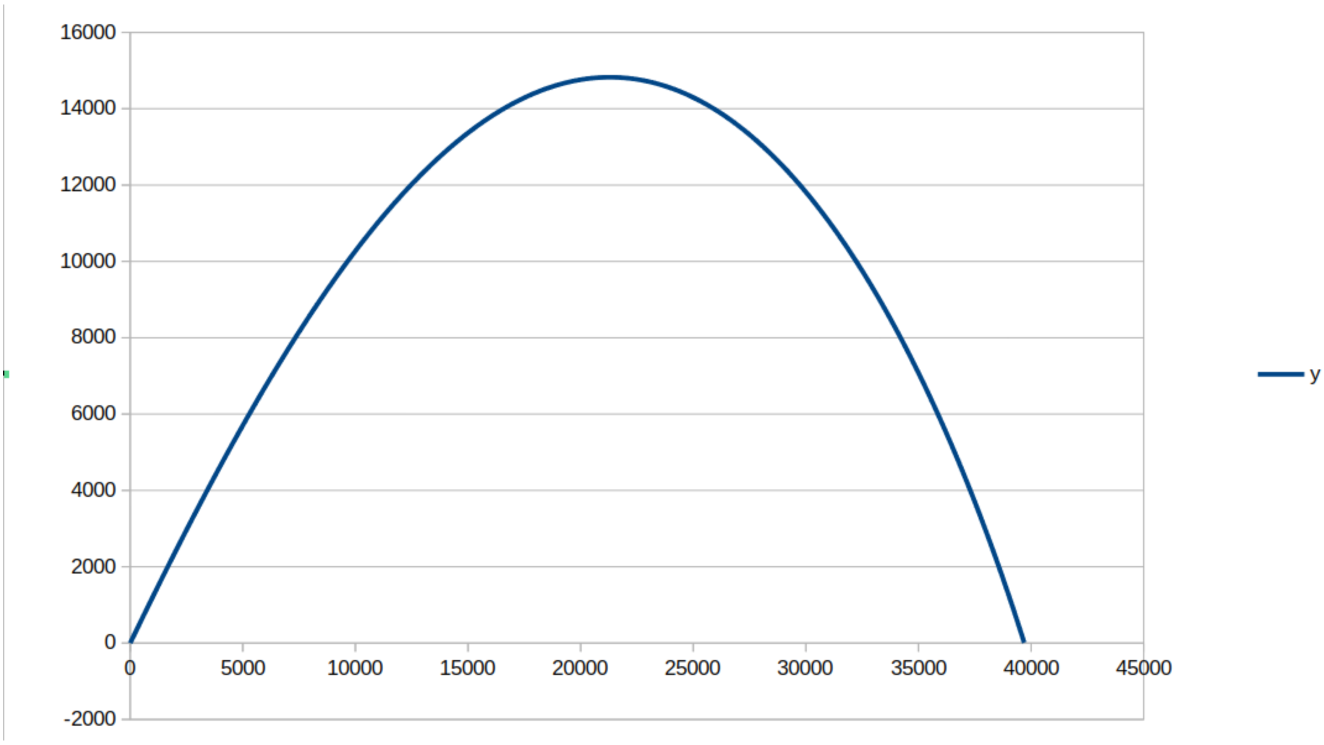
```
  return std::sqrt(std::pow(x - distance, 2) + std::pow(y - elevation, 2));
}

auto bum0 = [](double alpha)
{
  std::ofstream empty_file;
  return wziuuum_i_bum(alpha, 30000, 0, empty_file);
};

auto bum300 = [](double alpha)
{
  std::ofstream empty_file;
  return wziuuum_i_bum(alpha, 30000, 300, empty_file);
};
```

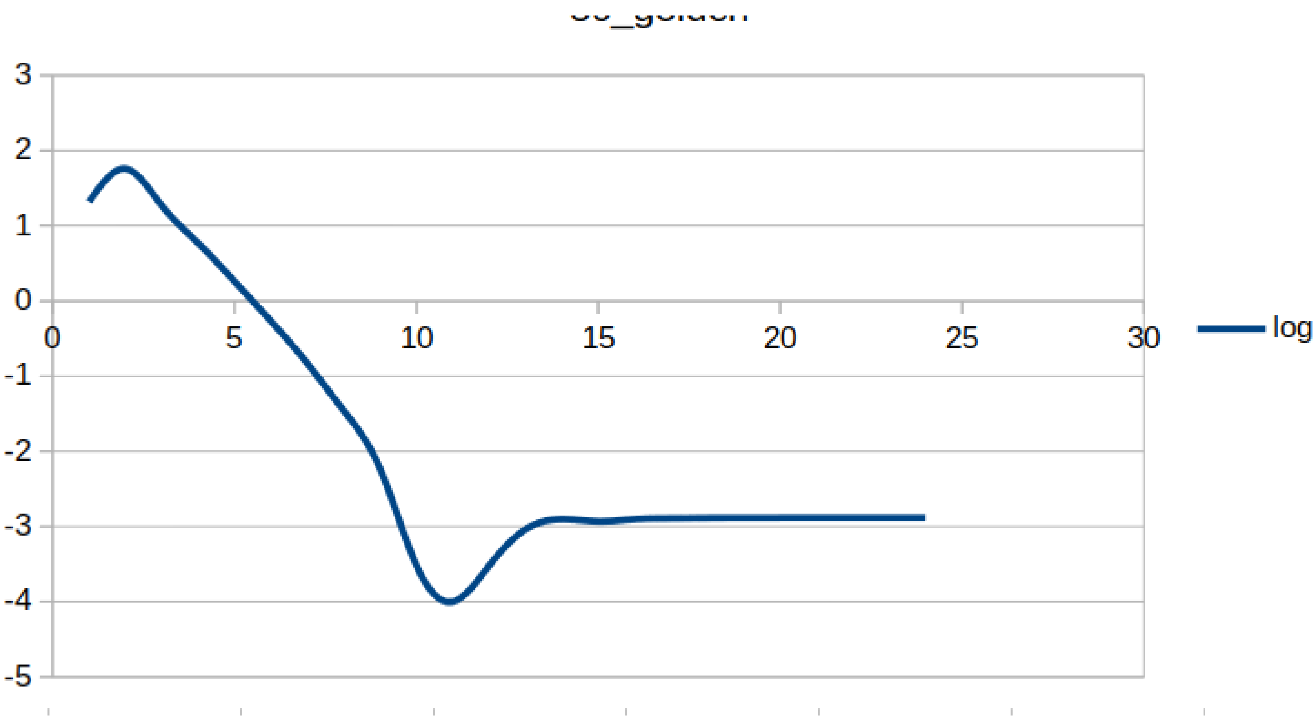
# 3 Wyniki

## 3.1 Trajektoria lotu dla maksymalnego zasięgu

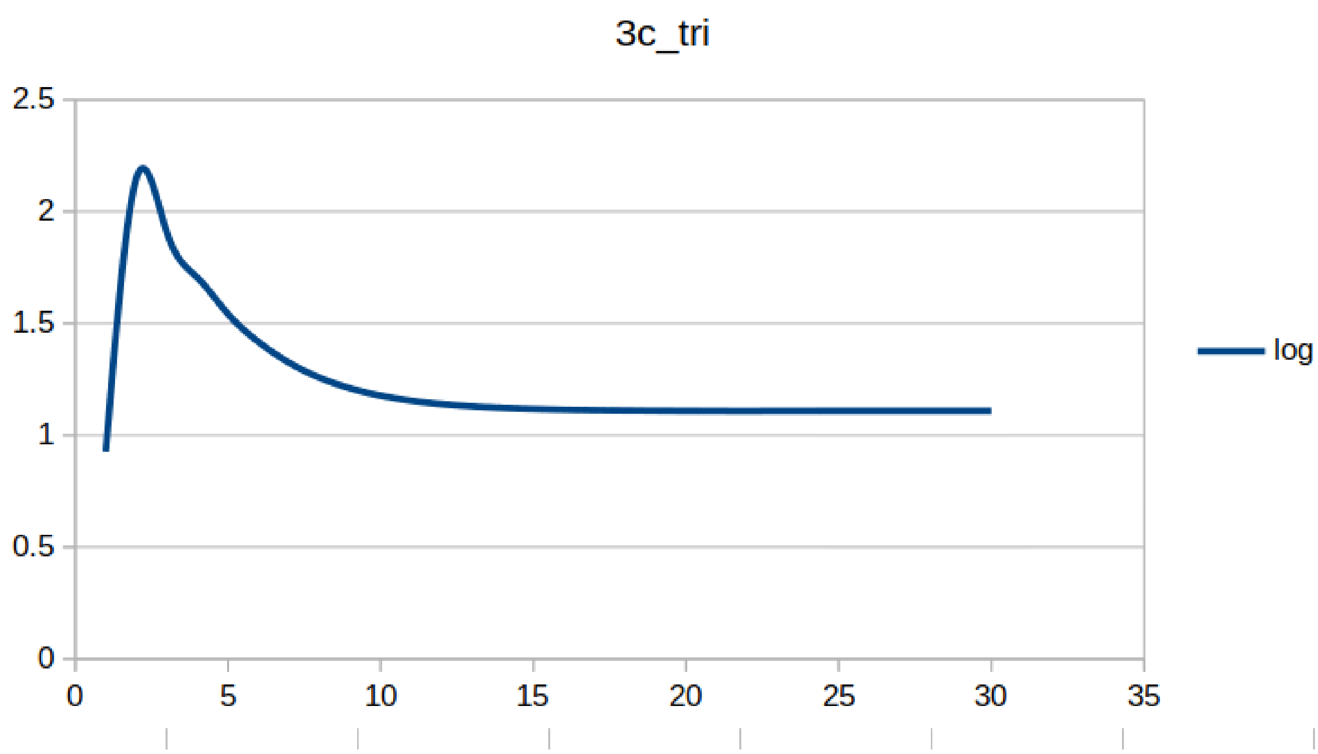


## 3.2 Porównanie metody złotego podziału i trójpodziału dla poszczególnych iteracji

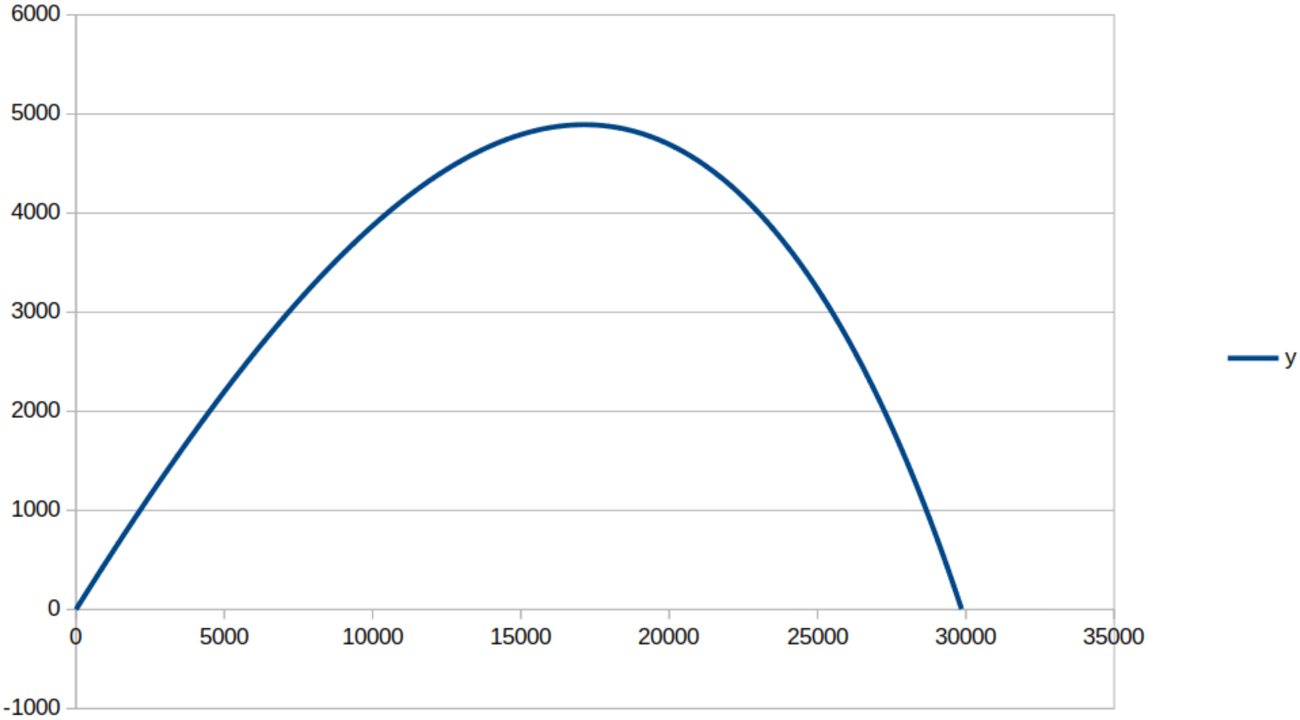
### 3.2.1 Złoty podział



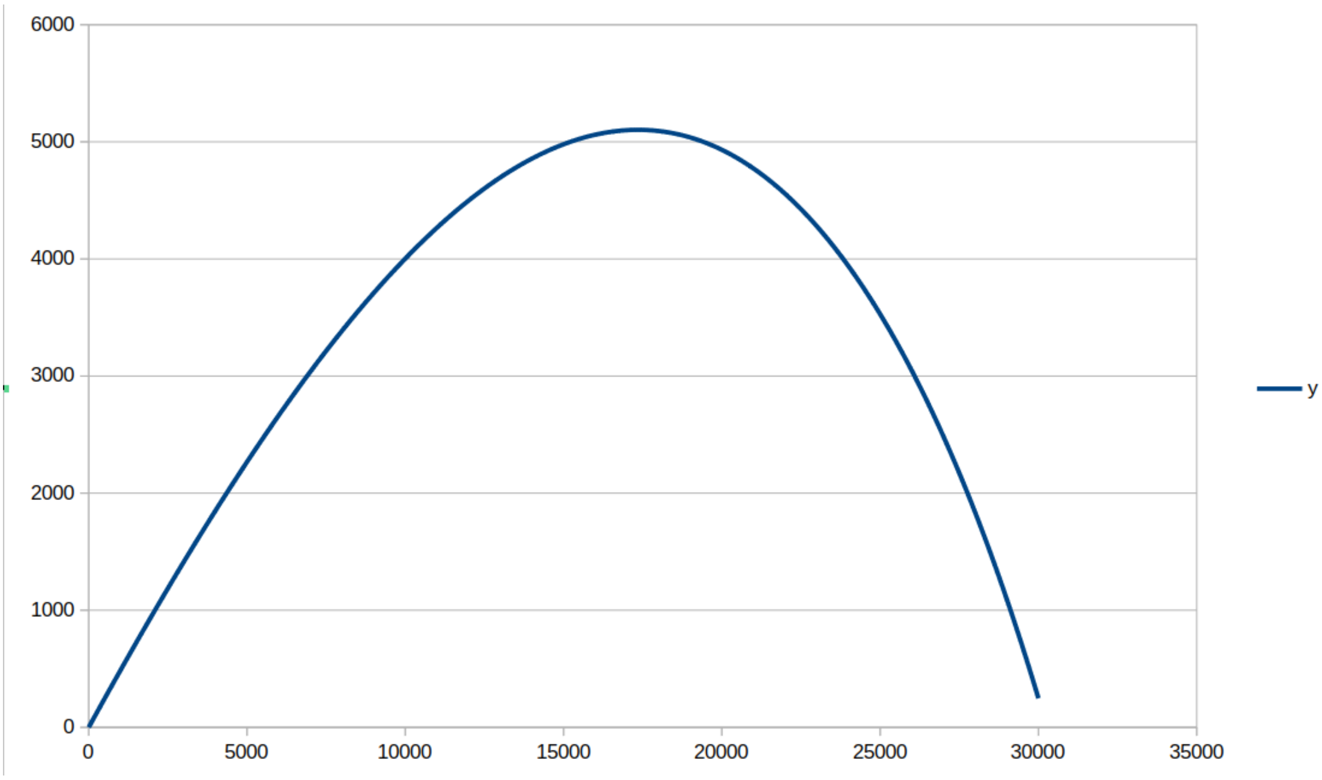

### 3.2.2 Trójpodział



### 3.3 Trajektoria lotu dla maksymalnego zasięgu dla wysokości 0




### 3.4 Trajektoria lotu dla maksymalnego zasięgu dla wysokości 300



# 4 Wnioski

Podsumowując, metoda złotego podziału okazuje się być skutecznym narzędziem do znajdowania minimum funkcji z zadaną dokładnością. Wykorzystuje ona unikalne właściwości złotego podziału, aby optymalizować proces poszukiwania minimum, co prowadzi do szybkich i efektywnych wyników.

W przypadku podziału przedziału na trzy części, obserwujemy, że liczba potrzebnych iteracji rośnie. Jest to spowodowane koniecznością dodatkowego sprawdzania i porównywania wartości funkcji w dwóch punktach, zamiast jednego. Mimo to, metoda ta nadal jest skuteczna, choć może być nieco wolniejsza w porównaniu z tradycyjnym podejściem.

Jednakże, ogólnie rzecz biorąc, metoda złotego podziału nie wymaga dużej ilości iteracji do rozwiązania danego problemu, co czyni ją szybką i skuteczną. Ta cecha jest szczególnie korzystna w przypadkach, gdy mamy do czynienia z złożonymi funkcjami lub gdy mamy ograniczone zasoby obliczeniowe.

W związku z powyższym, metoda złotego podziału jest cennym narzędziem w arsenale technik optymalizacji, które mogą być stosowane w różnych dziedzinach nauki i inżynierii.