Piotr Barasiński   
Patryk Fiałkowski  
  
Optymalizacja funkcji wielu zmiennych metodami bezgradientowymi.

# Wstęp

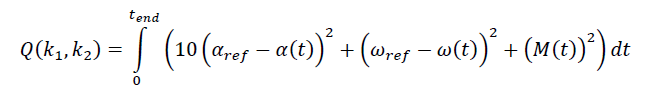
Celem ćwiczeń było zapoznanie się z metodami bezgradientowymi, Hooke’a-Jeevesa oraz Rosenbrocka, i ich implementacji do rozwiązania problemu optymalizacji.

# Przebieg ćwiczenia

Ćwiczenie było podzielone na dwa etapy, część testową w której należało wykonać 100 wywołań funkcji Hooke’a-Jeevesa oraz Rosenbrocka dla 3 różnych długości kroków, gdzie funkcja celu to:



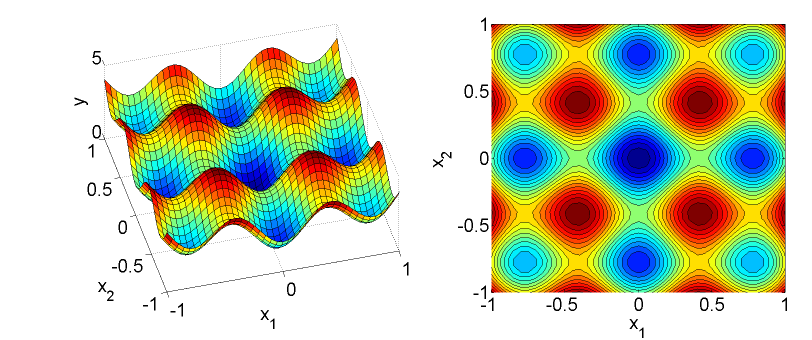
oraz część rzeczywistą, gdzie wywołując metodę Hooke’a-Jeevesa i Rosenbrocka szukaliśmy wartości współczynników wzmocnienia dających minimum dla funkcjonału:



# Część testowa

Wykonanie po 100 wyników funkcji optymalizacyjnych z losowego punktu funkcji dla   
3 róznych kroków dających nam łącznie 300 wyników umożliwiły wyprowadzenie średnich wyników dla metod, które pozwoliły ustalić ich szybkość i dokładność. W naszym sprawozdaniu przyjęliśmy długość kroków 0.25, 0.5 i 0.7. Porównując ilość wywołań między metodami Hooke’a-Jeevesa oraz Rosenbrocka od razu widać, że ta druga jest szybsza dla mniejszej długości kroku, natomiast ilość wywołań funkcji celu dla Hooke’a-Jeevesa jest bardzo zbliżona dla różnych długościach.

Dalej globalne minimum zostało przez ustalone nas ustalone na bazie wykresu funkcji:

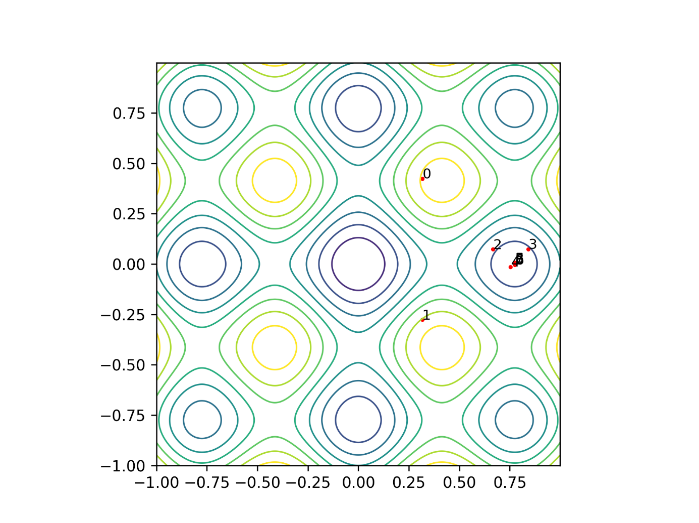


Widzimy, że minimum globalne wynosi x1=0, x2=0, y=0.

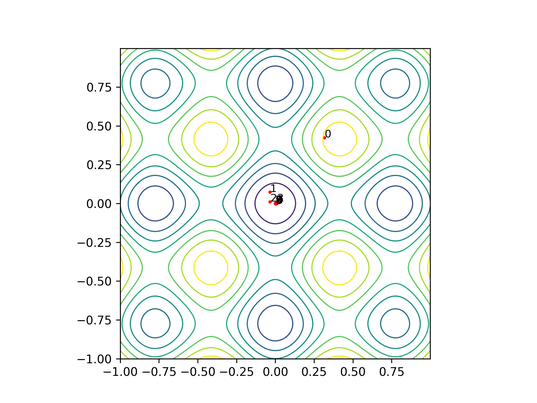
Przyjęliśmy założenie przybliżonych wartości na bazie epsilona z naszego kodu do wyznaczania minimu globalnych z wyników naszej optymalziacji. Obie funkcje przy większym kroku wskazywały więcej globalnych minimum, w wypadku wszystkich kroków różnica w liczbie odnajdywanych minimum globalnych była nieznaczna.

Wykres poziomnic dla korku s = 0.7:

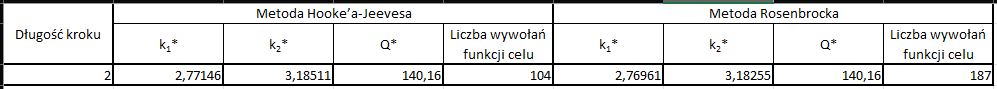
Hook-Jeves



Rosenberg



Wykorzystując wcześniej przygotowane metody można je zastosować do przeprowadzenia optymalizacji w celu odnalezienia pola pozycji oraz prędkości ramienia robota. W naszych obliczeniach funkcjonał przyjął wartość 140,16.

Warto zwrócić uwagę, że wartość szukanego k1 i k2 nieznacznie różną się pomiędzy metodami.  


Sam wykres symulacji prezentuje położenie kątowe ramienia (alfa) i prędkość kątowa ramienia (omega) ramienia robota w czasie 100s z krokiem czasowym 0,1s, dając nam 1000 wyników.

Jak widzimy dla obu metod wartości się niemal pokrywają, gdyż oba algorytmy szukają wspólnego minimum dla funkcjonału z podobną skutecznością.

# Wnioski

Przetestowane metody zostały zastosowane do optymalizacji parametrów ramienia robota, co ilustruje praktyczne zastosowanie algorytmów optymalizacji w dziedzinie robotyki. Obie metody prowadzą do zbliżonych wyników, co potwierdza ich skuteczność.

Porównując obie techniki optymalizacji funkcji wielu zmiennych głównym czynnikiem je różniącym jest prędkość obliczeń w stosunku do długości kroku. Funkcja Hook'a-Jeevesa jest stabilniejsza jeśli chodzi o ilość wywołań, jednocześnie dając podobną ilość minimum globalnych dla funkcji testowej, również obie funkcje podają nam bardzo zbliżone wyniki dla naszego problemu rzeczywistego.

Metoda Hook'a-Jeevesa to bardziej prosty i bezpośredni sposób optymalizacji, podczas gdy metoda Rosenbrocka wykorzystuje bardziej zaawansowane informacje o funkcji celu, co może prowadzić do bardziej efektywnego i szybszego znajdowania minimum przy odpowiednich parametrach, np. długości kroku.

Wyniki naszej analizy ukazują, że wybór odpowiedniej długości kroku ma istotny wpływ na skuteczność i szybkość algorytmów. Optymalny wybór długości kroku może być kluczowy dla efektywnej optymalizacji.

# Kod :

Funkcje optymalizacji z pliku opt\_alg.cpp

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Funkcje z user\_func.cpp:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Plik main:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie