Piotr Barasiński   
Patryk Fiałkowski  
  
Optymalizacja z ograniczeniami funkcji wielu zmiennych metodami bezgradientowymi.

# Wstęp

Celem ćwiczeń było zapoznanie się z metodami bezgradientowymi, Neldera-Meada oraz funkcją kary, i ich implementacji do rozwiązania problemu optymalizacji.

# Przebieg ćwiczenia

Ćwiczenie było podzielone na dwa etapy, część testową w której należało wykonać 100 wywołań funkcji Neldera-Meada dla zewnętrznej oraz wewnętrznej funkcji kary dla 3 różnych parametrów a, gdzie funkcja celu to:

Obraz zawierający Czcionka, diagram, tekst, linia

Opis wygenerowany automatycznie

oraz część rzeczywistą, gdzie wywołując metodę Neldera-Meada z zewnątrzną funkcją kary, szukaliśmy wartości przemieszczenia oraz prędkości piłki w celu symulacji trajektorii jej lotu:

Obraz zawierający tekst, diagram, linia, Wykres

Opis wygenerowany automatycznie

# Część testowa

Wykonanie po 100 wyników funkcji optymalizacyjnych z losowego punktu funkcji dla   
3 różnych wartości a dających nam łącznie 300 wyników umożliwiły wyprowadzenie średnich wyników dla metod, które pozwoliły ustalić ich szybkość i dokładność. Wartości **a** zostały z góry założone w treści zadania. Porównując zastosowanie zewnętrznej oraz wewnętrznej funkcji kart dla metody Neldera-Meada widzimy, że zewnętrzna funkcja kary lepiej wypada przy mniejszej wartości a, natomiast wewnętrzna przy tym zbliżonej 5.

Wyniki z również ukazują nam znaczną różnicę pomiędzy wynikami w y, gdyż jest to mocno powiązane z wynikiem funkcji kary. Wewnętrzna funkcja uwzględnia odwrotność funkcji ograniczających, które często są bliskie zeru, stąd otrzymujemy niskie wartości dla S(x1, x2) co wpływa na wynik funkcji y.

# Przypadek rzeczywisty

Wykorzystując wcześniej przygotowaną metodę z zewnętrzna funkcją kary możemy ją zastosować do przeprowadzenia optymalizacji w celu wykonania symulacji trajektorii lotu rzuconej piłki. Przez to, że zawsze startujemy z losowo wybranych punktów wyniki będą różniły się miedzy wywołaniami.

A white rectangular box with black text

Description automatically generated

Poniżej wykres toru lotu piłki z symulacji:

A graph with a green line and red line

Description automatically generated

Jak widzimy x wzrasta, a y spada, co jest zgodne z grawitacją i omawianym w instrukcji efektem Magnusam, w wyniku którego piłka spadając zmienia również położenie w osi x, ponadto osie przecinają się mniej więcej w punkcie 5,5.

# Wnioski

Metoda Neldera-Meada (inaczej znana jako algorytm simpleksowy) jest techniką optymalizacyjną używaną do minimalizacji funkcji wielu zmiennych bez użycia gradientów. W tym kontekście funkcje kary są stosowane do monitorowania i kary za naruszanie ograniczeń zadania optymalizacji. Zadaniem funkcji kar jest ukaranie rozwiązań, które prowadzą do niespełnienia warunków ograniczeń, ale nie jest bezpośrednio związana z funkcją celu (funkcją, którą próbujemy zminimalizować). Naruszenie ograniczeń jest karane poprzez dodanie do funkcji celu pewnej kary, proporcjonalnej do stopnia naruszenia.

Różnica między zewnętrzną a wewnętrzną funkcją kary w metodzie Neldera-Meada polega na tym, czy funkcja kary jest oddzielna od funkcji celu (zewnętrzna) czy też zintegrowana z nią (wewnętrzna). Oba podejścia są stosowane w sytuacjach, gdzie występują ograniczenia zadania optymalizacyjnego.

W naszym wypadku funkcje kary mogą się różnić dokładnością i szybkością, w zakresie szybkości:

# Odpowiedź na to, która jest szybsza, może zależeć od implementacji i charakterystyki konkretnego algorytmu optymalizacyjnego.

# W naszym wypadku wydajność funkcji kary zależała w znacznym stopniu od podanego a, dla wyższych a znaczne szybszym rozwiązaniem była funkcja zewnętrzna kary.

# W praktyce, wybór między zewnętrzną a wewnętrzną funkcją kary zależy od specyfiki problemu, dostępnych danych. Często konieczne może się okazać eksperymentalne porównanie obu podejść w kontekście konkretnego problemu, aby określić, które jest bardziej efektywne.

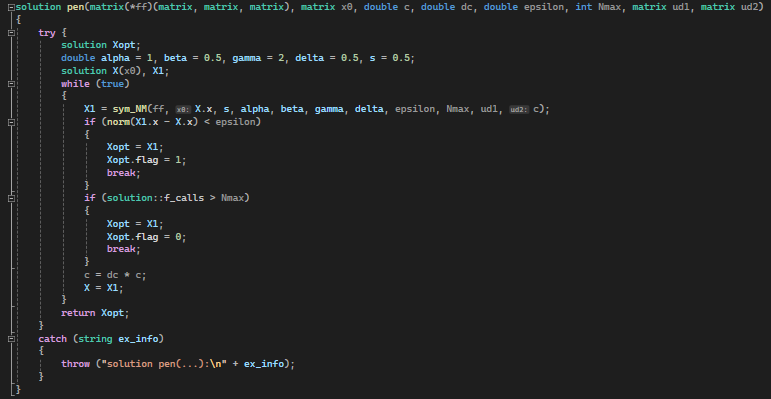
W naszym wypadku dla funkcji testowej ważny był efekt w postaci wyniku dla r, który decydował o tym, czy funkcja znalazła się w zakresie minimum globalnego (których w naszej funkcji jest nieskończenie wiele, ale ogranicza nasz zakres na bazie funkcji ograniczeń).

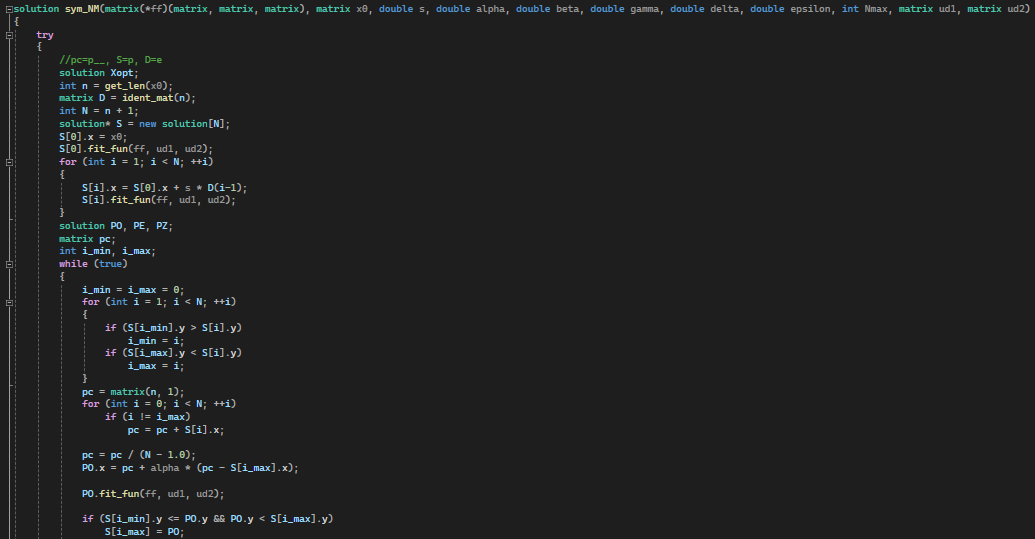
Dla obu funkcji kar wynik w tym zakresie był podobny, ale przy wyższym „a” znacząco lepiej w efektywności wyglądała funkcja zewnętrzna kary, dodatkowo nakładała również znacznie mniejszy wynik kary na naszą funkcję co widać na bazie y.

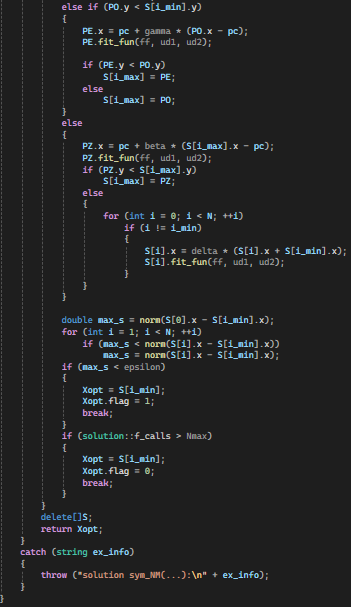
W wypadku rzeczywistego problemu wynik oraz wykres z symulacji wykazuje, że funkcja faktycznie spełnia swoje zadanie i w sposób prawidłowy oblicza ruch piłki. Trajektoria lotu piłki zmienia się w zależności od losowo przyjętej prędkości początkowej oraz rotacji, startując zawsze z ruchu x=0, y=100, czego efektem są różnice pomiędzy wywołaniami funkcji, w zależności od losowych atrybutów piłka osiąga inny wynik dla x end, końcowy y zawsze musi wynosić 0.

# Kod :

Funkcje optymalizacji z pliku opt\_alg.cpp:

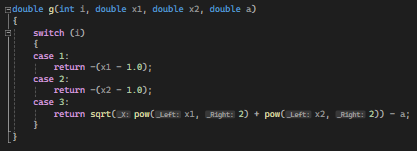






Funkcje z user\_func.cpp:

Funkcja warunków:

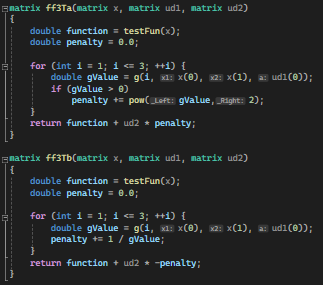


Funkcja testowa:

Obraz zawierający zrzut ekranu

Opis wygenerowany automatycznie

Funkcje kary dodawane do wyniku funkcji testowej:



Funkcja , z której pobieramy macierz z wynikami do symulacji:

Obraz zawierający tekst, elektronika, zrzut ekranu, wyświetlacz

Opis wygenerowany automatycznie

Plik main:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie