

Znajdowanie zer wielomianu w postaci Beziiera za  
pomocą przycinania sześciennego

Maciej Pacut

Wrocław 2010

## 0.1 Wstęp

!!!!!!! Wstęp napisz na końcu, powinien on opisywać, co zostanie zawarte w pracy

Jakie zastosowania ma znajdowanie zer? Szukanie przecięcia krzywej z prostą. Szukanie najbliższych punktów na prostej. Niniejsza praca daje pogląd na to, czym jest szukanie zer wielomianu i jak napisać metodę szukania zer.

## 0.2 Numeryczne znajdowanie zer

### 0.2.1 Znajdowanie jednego zera

Metoda znajdowania jednego zera funkcji dostaje argumenty:

1. pewien opis funkcji  $f$ , który umożliwia obliczenie wartości tej funkcji dla dowolnego argumentu
2. przedział początkowy  $P_0$ , czyli przedział w którym szukamy zera funkcji  $f$ ; metoda nie znajdzie żadnego zera spoza tego przedziału
3. żądana dokładność  $\varepsilon$  znalezienia zera funkcji  $f$

Metoda numerycznego znajdowania jednego zera funkcji  $f$  generuje ciąg przedziałów  $\langle P_0, P_1, \dots, P_n \rangle$  o własnościach:

1. każdy przedział jest podprzedziałem poprzedniego przedziału:

$$P_k \subset P_{k-1} \quad (1 \leq k \leq n)$$

2. każdy przedział zawiera zero funkcji  $f$  z przedziału  $P_0$

Adnotacja na dole strony: odnośnie zbioru pustego. Jeśli w przedziale  $P_0$  nie ma zera funkcji  $f$  to stwierdzenie, że przedział pusty (porażka) zawiera zero funkcji  $f$  z przedziału  $P_0$  jest prawdziwe.

3. ostatni przedział  $P_n$  jest pierwszym przedziałem krótszym niż  $2\varepsilon$

Jeśli w przedziale  $P_0$  metoda nie znalazła zera, to ostatnim przedziałem w ciągu jest przedział pusty; wtedy wynikiem jest brak zer. W przeciwnym przypadku wynikiem jest środek przedziału  $P_n$ , który jest odległy od zera funkcji  $f$  o co najwyżej  $\varepsilon$ . W przypadku istnienia zera w przedziale  $P_0$ , udało się znaleźć zero funkcji  $f$  z dokładnością  $\varepsilon$ .

Przedstawiony został w powyższym schemacie pewien zbiór metod znajdowania jednego zera funkcji  $f$ . Metody w tym zbiorze różnią się realizacją funkcji generującej ciąg  $\langle P_0, P_1, \dots, P_n \rangle$ . Generowanie ciągu można opisać w postaci szukania podprzedziału  $P_{k+1}$  dla danego przedziału  $P_k$ .

Przykładem metody znajdowania jednego zera funkcji jest bisekcja. W bisekcji przedział  $P_{k+1}$  otrzymuje się dzieląc  $P_k$  na pół i jako  $P_{k+1}$  wybierając

tę połowę, w której jest możliwość wystąpienia zera. W przypadku, gdy w żadnej połowie nie ma możliwości wystąpienia zera,  $P_{k+1}$  jest przedziałem pustym i metoda kończy działanie. W przypadku, gdy w obu połowach jest możliwość wystąpienia zera, jako  $P_{k+1}$  wybierana jest dowolna z połów. W bisekcji należy określić realizację operacji określającej możliwość wystąpienia zera w danym przedziale (w danej połowie przedziału  $P_k$ ). Przykładową taką operacją jest sprawdzenie, czy funkcja  $f$  ma w końcach sprawdzanego przedziału różne znaki.

! rysunek bisekcji

W niniejszej pracy zajmujemy się inną metodą z wyżej przedstawionego zbioru metod szukania zer funkcji. Przykładem metody nie należącej do wyżej opisanego zbioru metod jest metoda Newtona. Znaczącą różnicą między metodą Newtona a metodami opisanymi wyżej jest generowanie ciągu przybliżeń zera funkcji zamiast generowania ciągu przedziałów zawierających zero.

## 0.2.2 Znajdowanie wielu zer

Zmodyfikujmy powyższy schemat znajdowania jednego zera. Zamiast generowania ciągu przedziałów, metoda szukania wielu zer generuje drzewo przedziałów. Drzewo przedziałów ma własności:

1. w korzeniu znajduje się przedział początkowy  $P_0$
2. każdy potomek jest podprzedziałem swojego rodzica
3. przedział z każdego węzła zawiera zero funkcji  $f$  !! to niekoniecznie jest prawda
4. każdy liść  $L$  jest pierwszym przedziałem na ścieżce od korzenia do  $L$ , który ma długość mniejszą niż  $2\varepsilon$

Wynikiem jest zbiór środków niepustych przedziałów znajdujących się w liściach drzewa przedziałów.

!! Rysunek drzewa przedziałów w bisekcji przy dwóch podprzedziałach, rysunek drzewa przedziałów zdegenerowanego do listy !! Podpis do rysunku: W wypadku 1 zera drzewo może się degenerować do listy, którą można traktować jako ciąg z poprzedniego schematu.

Przedstawiony został w powyższym schemacie pewien zbiór metod znajdowania jednego zera funkcji  $f$ . Metody w tym zbiorze różnią się realizacją funkcji generującej drzewo przedziałów. Generowanie drzewa można opisać w postaci operacji znajdowania zbioru podprzedziałów (potomków) dla danego przedziału (rodzica).

Przykładem metody szukania wielu zer funkcji jest modyfikacja opisanej w poprzednim podrozdziale metody bisekcji. W przypadku gdy w obu połowach dzielonego przedziału jest możliwość wystąpienia zera, obie połowy są określane jako potomkowie przedziału.

### 0.2.3 Szybkość metody znajdowania zer

Szybkość znajdowania zer jest jedną z miar jakości metody. Możemy do pomiaru szybkości użyć komputera. Można zaprogramować metodę szukania zer w pewnym ustalonym języku programowania, za pomocą ustalonego kompilatora i uruchomić ją na komputerze z ustalonym procesorem, ustaloną hierarchią pamięci, na ustalonym systemie operacyjnym, na którym pracują równolegle ustalone programy i podać na wejście programowi ustaloną funkcję, której zer szukamy i ustaloną dokładność. Czas, który upłynął od podania danych do otrzymania wyniku można traktować jako szybkość działania metody. To podejście ma pewne wady, takie jak testowanie metody na skończonej liczbie funkcji oraz brak możliwości powtórzenia eksperymentu w dokładnie tych samych warunkach. Jednak jest to pomiar łatwy do wykonania i dający pewne wskazówki co do szybkości działania metody.

Zamiast mierzenia czasu możemy zmierzyć liczbę operacji zmiennoprzecinkowych, które wykonano znajdując zera konkretnej funkcji. To podejście wyklucza wiele czynników, biorących udział w poprzednim pomiarze. Przykładowo, pomijane są realizacje operacji przydziału i zwalniania pamięci, wczytywania danych i wypisywania wyników. Zyskujemy możliwość porównywania metod znajdowania zer przez uruchamianie ich na tych samych danych, lecz na innych komputerach. Jednakże, wciąż jesteśmy ograniczeni do testowania na skończonej liczbie funkcji.

Lepsze metody zostały wymyślane. Możemy rozpatrzyć zależności między kolejnymi przedziałami dla dowolnej funkcji analizując operację znajdowania następnika przedziału w ciągu lub potomków przedziału w drzewie przedziałów. Posługujemy się pojęciem rzędu zbieżności, czyli miarą tego, ile krótszy jest następny wygenerowany przedział od poprzedniego. Oznaczając długość  $k$ -tego przedziału w ciągu jako  $e_k$  metoda ma rząd zbieżności  $p$ , gdy:

$$e_k^p \sim e_{k+1}$$

Biorąc pod uwagę, że przedziały nie mogą mieć ujemnej długości, że zależność może się stabilizować dopiero od pewnego kroku, oraz że analizowana metoda powinna być zbieżna, mamy rząd zbieżności 1 gdy:

$$\lim_{k \rightarrow \infty} \frac{e_{k+1}}{e_k} = u \quad 0 < u < 1$$

oraz rząd zbieżności  $p$ ,  $p > 1$ , gdy:

$$\lim_{k \rightarrow \infty} \frac{e_{k+1}}{e_k^p} = u \quad u > 0$$

Przykładowo. bisekcja ma rząd zbieżności 1, gdyż następny przedział jest połową poprzedniego:

$$\begin{cases} e_{k+1} = \frac{1}{2}e_k \\ \lim_{k \rightarrow \infty} \frac{e_{k+1}}{e_k} = \frac{1}{2} \end{cases} \quad (1)$$

Musimy także uwzględnić, ile operacji zostało wykonanych, aby wygenerować następny przedział. Bierzemy pod uwagę tylko niektóre operacje, tj. obliczenie wartości funkcji w punkcie czy obliczenie pochodnej, resztę operacji traktując jako tanie.

Przykładowo, jeśli porównujemy dwie metody, z której jedna wymaga obliczenia funkcji w krancach poprzedniego przedziału i pochodnej wewnątrz i następny przedział jest krótszy 3-krotnie oraz mamy metodę, która wymaga jedynie obliczenia pochodnej i przedział jest 2-krotnie mniejszy to  $3/3 = 1$  a  $2/1 = 2$  i druga metoda jest lepsza.

W przypadku wielu zer drzewo rozwijane jest tak, że w jednym kroku (dla jednej ewaluacji) rozwijana jest tylko jedna gałąź, dlatego ignorujemy kroki pracujące nad innymi zerami.

Przykładowo, bisekcja ma zbieżność liniową. O zbieżności liniowej i dodatkowemu wymaganiu na zbieżność liniową - stała musi być mniejsza od jedności, gdyż inaczej metoda byłaby rozbieżna - kolejny przedział byłby większy od poprzedniego!

## 0.3 Znajdowanie zer wielomianów

### 0.3.1 Zera wielomianów stopnia niższego niż 5

Istnieją wzory analityczne na zera wielomianów niskiego stopnia

Wzory na zera 2 i 3 napisac, 4 wspomnieć że są.

Poprawianie wzorów. Wzory Viete'a. Dodatkowy krok metody Newtona.

### 0.3.2 Zera wielomianów dowolnego stopnia

\* Wyższego stopnia należy szukać numerycznie \*\* <http://mathworld.wolfram.com/AbelsImpossibility/>

Znajdowanie zer wielomianów nie jest jednak tak ogólnym zadaniem jak znajdowanie zer funkcji ciągłych. Możemy skorzystać z wielu własności wielomianów.

Gdy mamy informację, że funkcja, której zer szukamy jest wielomianem, możemy skorzystać z własności wielomianów do: \* przyspieszenia metody \* upewnienia się, że znajdujemy wszystkie zera \*\* ograniczenie na liczbę zer rzeczywistych - stopień \*\* ustalenie pewnego przybliżenia początkowego \*\* ciągłość 0..stopień-1 pochodnych

## 0.4 Wielomiany w postaci Beziera

!! wiadomosci z ponizszego paragrafu powinny potwierdzic zrodłami albo chociaz sprawdzic ich poprawnosc w jakimś podręczniku.

Zbiór wielomianów stopnia nie większego niż  $n$  jest przestrzenią wektorową o wymiarze  $n$ . Każda przestrzeń liniowa ma nieskończoną liczbę baz. Jedną z baz wielomianów jest baza potęgowa  $1, x, x^2, x^3, \dots, x^n$ . Inną bazą jest baza Bernsteina  $B_0^n(x), \dots, B_n^n(x)$ , gdzie  $B_i^n(x) = \binom{n}{i} x^i (1-x)^{n-i}$ . Wielomian wyrażony w bazie Bernsteina nazywany jest wielomianem w postaci Beziera. Każdy wielomian w bazie potęgowej  $P(x) = \sum_{i=0}^n p_i x^i$  można przedstawić w bazie Bernsteina  $P(x) = \sum_{i=0}^n b_i B_i^n(x)$ . Zależność między wektorem  $p = \langle p_0, \dots, p_n \rangle$  a wektorem  $b = \langle b_0, \dots, b_n \rangle$  w formie  $p[m_{ij}] = b$  nazywana jest macierzą konwersji albo macierzą przejścia.

Postać Beziera ma inne własności niż postać potęgowa. W jednych zastosowaniach sprawdza się postać potęgowa, w innych postać Beziera, w innych sprawdzają się jeszcze inne bazy. Przykładowym zastosowaniem postaci Beziera jest grafika komputerowa. Zgodnie z tw. Weierstrassa (udowodnionym zresztą przy pomocy wielomianów w bazie Bernsteina) krzywa parametryczna opisana parą wielomianów może odwzorowywać dowolny kształt ciągły z dowolną dokładnością. Specjalne krzywe wielomianowe, krzywe, których wielomiany są opisane w postaci Beziera dają łatwość kształtowania krzywej. Związek współczynników Beziera wielomianów opisujących krzywą jest wizualnie powiązany z kształtem krzywej. Pozwala to ludziom nie znającym matematyki !! (bardzo niefortunne określenie) używać potężnego narzędzia jakim są krzywe parametryczne do konstrukcji kształtów. Krzywe i powierzchnie parametryczne są potężnym narzędziem, gdyż są opisane wektorowo a nie rastrowo, dzięki czemu pozwalają na łatwą (effortless, bezwysiłkową?), bez modyfikacji każdego piksela modyfikację istniejących kształtów oraz pozwalają na przekształcenia tj. przesunięcie, obrót i skalowanie.

!! rysunek wielomianu w postaci Beziera wraz z punktami kontrolnymi  
!! rysunek krzywej !! rysunek powierzchni

Powodem powiązania geometrycznego postaci Bezier z jej wykresem jest jej własność otoczki wypukłej. Krótki dowód przez kombinację wypukłą.

!! rysunek otoczki wypukłej

## 0.5 Znajdowanie zer wielomianu w postaci Beziera

Przecięcie krzywa, powierzchnia-prosta jest problemem, który sprowadza się do znajdowania zera wielomianu w formie Beziera. Problem ten występuje np. w ray-tracingu, czyli wyświetlaniu sceny oświetlonej techniką śledzenia promieni w scenie, której obiekty są powierzchniami w postaci Beziera. Krótkie przedstawienie tej redukcji.

Pomysły na szukanie zer. Pierwszy pomysł to obliczenie macierzy przej-

scia (wspomnienie, że efektywne algorytmy wymagają czasu proporcjonalnego liniowo do stopnia, gdyż macierz przejścia jest wstęgowa). Jednakże, lepsze metody istnieją. Wykorzystują one pewne własności postaci Beziera.

\* Dzielenie przedziału. Algorytm de Casteljau. Poddziedzina. Rozszerzona definicja wielomianu Beziera ze względu na dziedzinę.

Znajdowanie wielu zer. Dzielenie przedziału na więcej części, gdy może wystąpić więcej niż 1 zero w przedziale (np. gdy jest zbyt długi, czyli przedział nie kurczy się tak szybko jak by to wynikało z szybkości zbieżności metody). Tutaj, rząd zbieżności metody to 2, więc dzielimy na dwie części w przypadku, gdy przedział się nie skurczył o połowę.

Metoda bezclip. Szybkość metody bezclip. Wykorzystujemy właściwości wielomianu: poddziedzina (do nowej otoczki wypukłej) Metoda pasuje do schematu w/w.

## 0.6 Aproksymacja wielomianu wielomianem

Bez wgłębiania się w samą aproksymację? Można przedstawić aproksymację jako zagadnienie geometryczne.

Przybliżenie wielomianu wysokiego stopnia wielomianem niskiego stopnia. "Redukcja stopnia".

Aproksymując wielomian upraszczamy go, tracąc informacje (punkty krańcowe, ) ale jednocześnie zyskując możliwość łatwiejszego przetwarzania go.

Macierz redukcji. Obliczanie macierzy redukcji metodą Lewanowicza-Woźnego.

## 0.7 n-clip

Dzięki skorzystaniu z dalszych własności postaci Beziera: \* Maksymalna różnica między wielomianami dzięki łatwemu obliczaniu normy maksymalnej wielomianu różnicy. \* Podnoszenie stopnia.

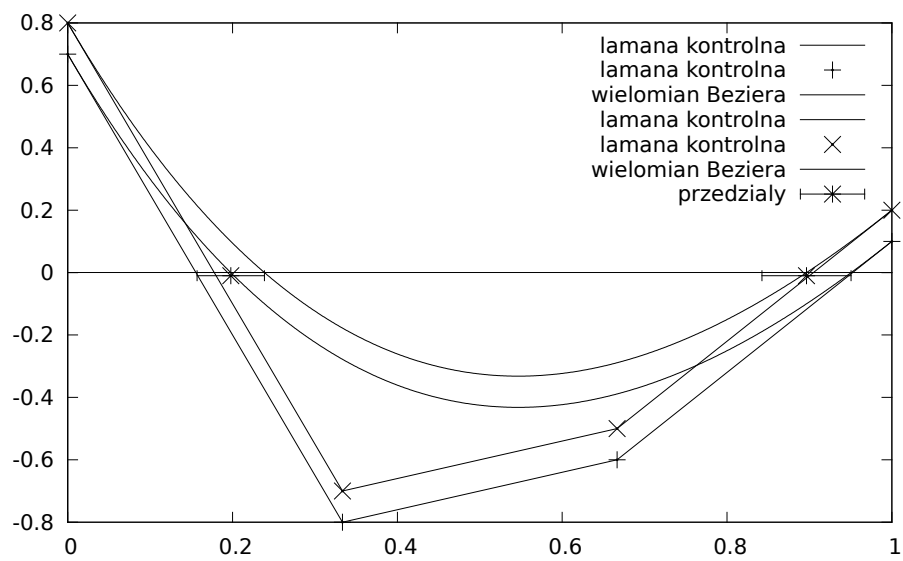
Dwa wielomiany ograniczające.

Wykorzystuje fakt, że funkcja, której zer szukamy jest wielomianem. Wykorzystujemy także pewne własności wielomianów Beziera tj. podział wielomianu w punkcie czy poddziedzina. (Potrzebne jest to przy wielokrotnym wykorzystaniu tej samej macierzy redukcji, zauważmy również, że po znalezieniu pierwszego zera nie warto szukać zer wielomianu  $P(x)/(x-r_0)$ , gdyż nie można wykorzystać tej samej macierzy redukcji)

### 0.7.1 cubicclipping

### 0.7.2 podsumowanie

!! na koncu



## 0.8 Dodatek 1: Operacje na przedziałach

- odejmowanie przedziałów od siebie