

Wstęp do metod numerycznych

11. Minimalizacja: funkcje jednej zmiennej

P. F. Góra

<http://th-www.if.uj.edu.pl/zfs/gora/>

2015

Lokalna minimalizacja ciągła

Minimalizacja funkcji jest jedną z najważniejszych operacji wykonywanych numerycznie. W niniejszym wykładzie zajmiemy się wyłącznie poszukiwaniem minimów lokalnych (nie globalnych) funkcji *ciągłych*, których argument także zmienia się w sposób ciągły. Tak poszukiwanie minimów globalnych, jak i minimalizacja dyskretna (zmienna (lub zmienne w przypadku wielowymiarowym) może przyjmować tylko pewne wartości punktowe) są znacznie trudniejszymi zagadnieniami, wykraczającymi poza ramy niniejszego wykładu.

Niekiedy zdarza się, że bardziej interesuje nas znalezienie maksimum, nie minimum. Jednak jeśli funkcja $g(x)$ ma jakimś punkcie maksimum, które chcemy znaleźć, to funkcja $f(x) = -g(x)$ ma w tym samym punkcie minimum.

W przeciwieństwie do omawianego poprzednio rozwiązywania równań algebraicznych, do stwierdzenia, że funkcja osiąga gdzieś minimum, nie wystarczy pojedyncze obliczenie funkcji. Wartość funkcji musimy porównywać z uprzednio policzonymi wartościami w innych punktach. Procedurę poszukiwania minimum musimy skończyć najpóźniej wtedy, gdy zmiana wartości funkcji staje się porównywalna z zadaną tolerancją, w szczególności, z dokładnością, z jaką prowadzimy obliczenia.

Jak dokładnie możemy zlokalizować minimum?

Niech $f(x)$ ma minimum w punkcie x_0 . Rozwijając w szereg Taylora otrzymujemy

$$f(x_0 + \delta) \simeq f(x_0) + \underbrace{f'(x_0)}_{=0} \delta + \frac{1}{2} f''(x_0) \delta^2 \quad (1)$$

Jeżeli $|f(x_0 + \delta) - f(x_0)| \leq \epsilon$, gdzie ϵ jest zadaną dokładnością, musimy przerwać obliczenia. Z rozwinięcia (1) widać natychmiast, że w takim wypadku

$$|\delta| \sim \sqrt{\epsilon}. \quad (2)$$

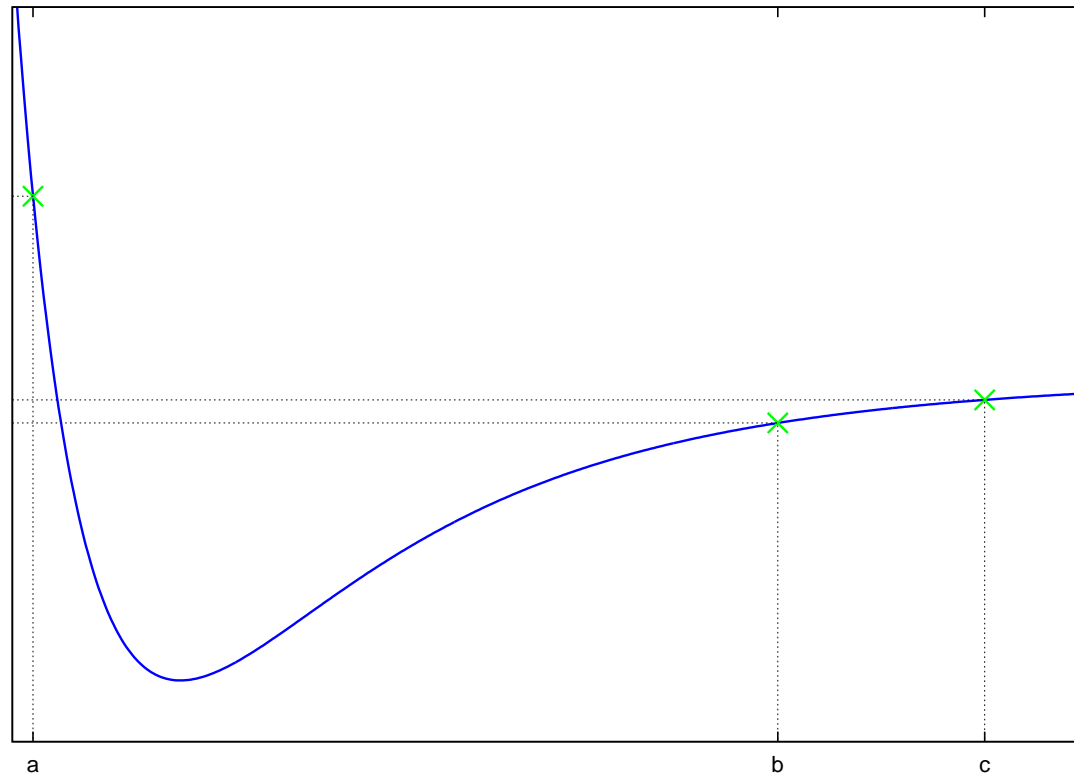
W szczególności widać zatem, że **minimum jesteśmy w stanie zlokalizować z dokładnością co najwyżej równą pierwiastkowi z numerycznej precyzji obliczeń.**

Wstępna lokalizacja minimum

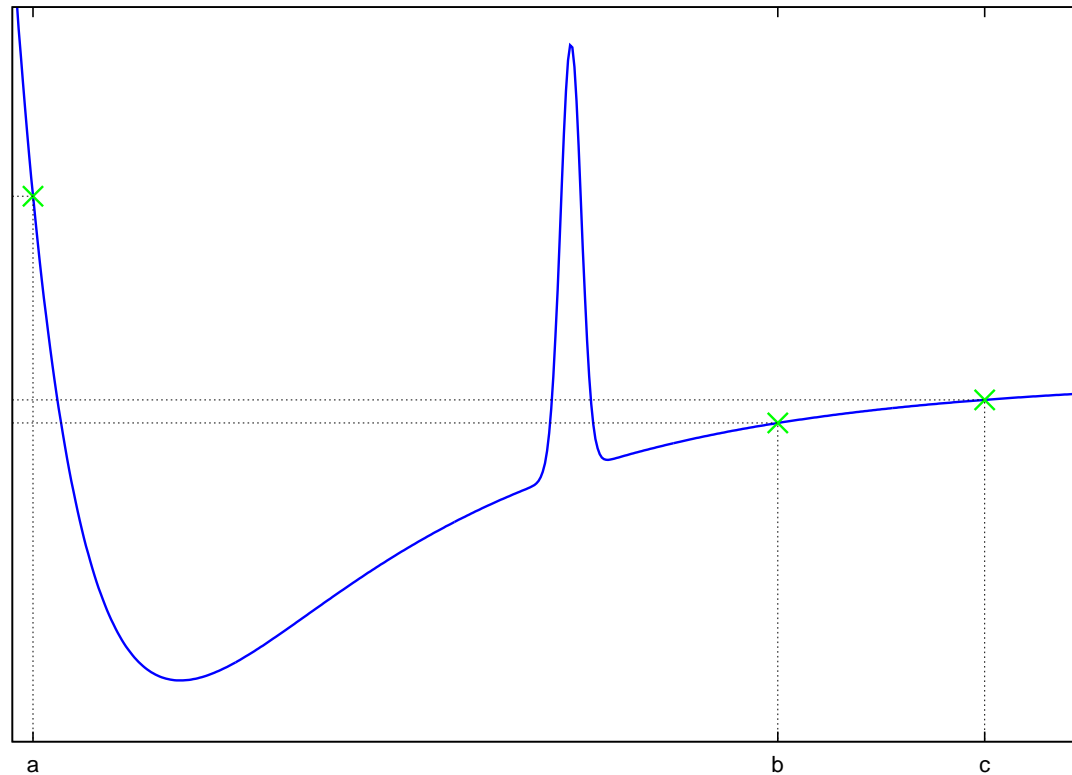
Mówimy, że trzy punkty (a, b, c) ograniczają (lokalizują) minimum funkcji $f(x)$, jeżeli

$$a < b < c: f(a) > f(b), f(c) > f(b) \quad (3)$$

Jak znaleźć takie punkty? Obliczamy wartość funkcji w dwu punktach; jeżeli nie zachodzą **bardzo szczególne** okoliczności, wyznaczają one lokalny kierunek spadku funkcji. Wyznaczamy trzeci punkt idąc w kierunku spadku, o taką samą odległość, jaka dzieliła punkty początkowe. Jeśli warunek (3) nie jest spełniony, *podwajamy krok*, zawsze biorąc pod uwagę dwa ostatnio obliczone punkty. Uwaga: Trzeba założyć maksymalną dopuszczalną liczbę kroków (lub maksymalną dopuszczalną wielkość kroku), aby zabezpieczyć się przed nieskończoną iteracją, gdybyśmy znaleźli się na monotonicznie malejącej gałęzi funkcji.



Trójka punktów (a, b, c) wyznaczająca przybliżone położenie minimum



Strategia z początkową trójką (a, b, c) znajdzie *jedno* z *dwóch* minimów tej funkcji, leżące w przedziale $[a, c]$.

Strategia

Gdy mamy trójkę (a, b, c) spełniającą warunki (3), wybieramy pewien punkt d , $a < d < c$, $d \neq b$ jako nowy punkt wewnętrzny. Obliczamy $f(d)$. Dalsze postępowanie zależy od tego, jak wartość $f(d)$ ma się do wartości $f(b)$.

$f(d) < f(b)$	$f(d) > f(b)$
Jeżeli $d < b$, to $c = b$ $b = d$	Jeżeli $d < b$, to $a = d$
Jeżeli $d > b$, to $a = b$ $b = d$	Jeżeli $d > b$, to $c = d$

(4)

“Przesuwając” punkty, analogicznie “przesuwany” też obliczone wartości funkcji (na przykład $c = b \Rightarrow f_c = f_b \equiv f(b)$), żeby nie trzeba było ich ponownie obliczać.

W każdym przypadku otrzymujemy *nową* trójkę (a, b, c) spełniającą warunki (3), a zatem możemy iterować całą procedurę. Długość przedziału zawierającego minimum jest za każdym razem mniejsza niż w poprzednim kroku iteracji. Iterację kończymy, gdy $|c - a| < \tau(|b| + |d|)$, gdzie τ jest tolerancją i należy wziąć “stare” b .

Jeżeli tylko funkcja jest ciągła w przedziale $[a, c]$ i ma w nim minimum, ta strategia zbiegnie się do tego minimum (jeżeli jest tam więcej niż jedno minimum, zbiegnie się do jednego z nich). Pozostaje tylko ustalić **jak wybrać punkt wewnętrzny d ?**

Metoda złotego podziału

W metodzie złotego podziału punktu d szukamy w *większym* z dwóch podprzedziałów $[a, b]$ i $[b, c]$. Okazuje się, że aby zminimalizować kłopoty wynikające z “pecha”, iż szukane minimum będzie w wielu krokach iteracji leżeć w krótszym z podprzedziałów, poszczególne przedziały powinny spełniać **złotą proporcję**: długość większego podprzedziału ma się do długości całego przedziału tak, jak długość mniejszego do długości większego. Zakładając, że podprzedział $[a, b]$ jest dłuższy, powinniśmy mieć

$$\frac{|b - a|}{|c - a|} = \frac{|c - b|}{|b - a|} \quad (5)$$

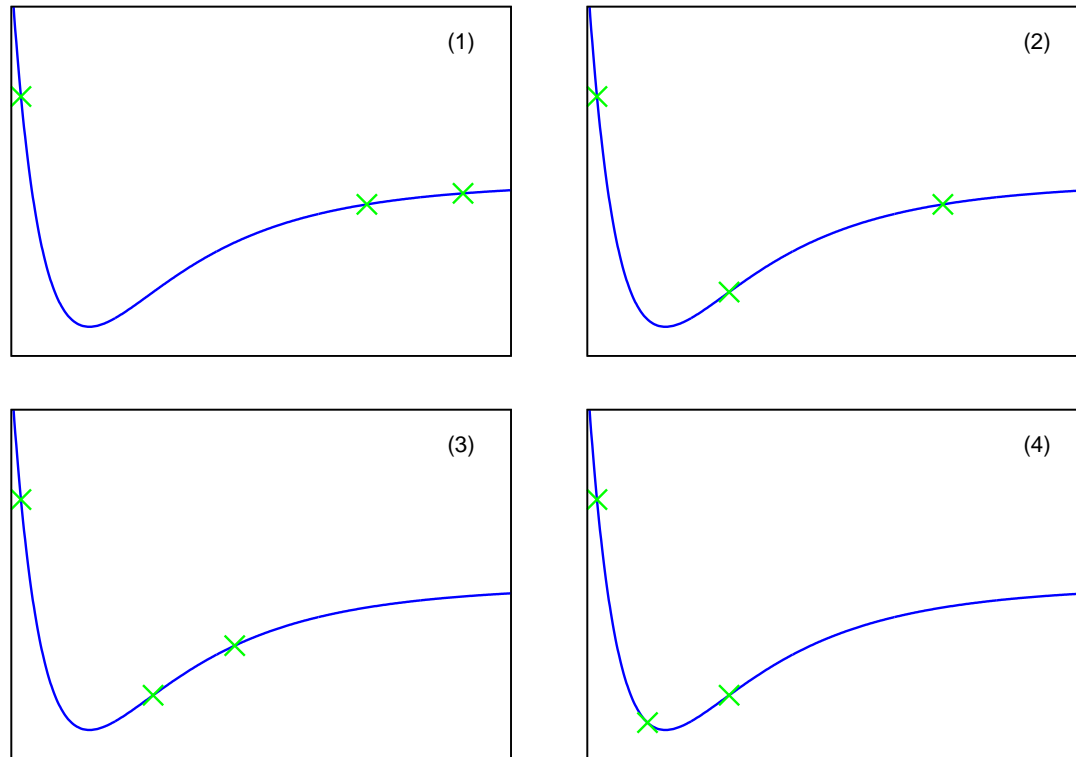
Choć początkowa trójka (a, b, c) zapewne nie spełnia “złotej proporcji”, szybko do niej dochodzimy, postępując jak następuje:

- Jeżeli $|b - a| > |c - b|$, $d = a + w \cdot |b - a|$,
- Jeżeli $|b - a| < |c - b|$, $d = b + w \cdot |c - b|$,

gdzie $w = \frac{3 - \sqrt{5}}{2} \simeq 0.381966$, po czym zawężamy przedział stosując (4).

Metoda złotego podziału tak naprawdę nie korzysta z wartości funkcji, a tylko z porównania co jest większe od czego. Metoda ta jest zbieżna liniowo.

Przykład



Cztery kroki metody złotego podziału

Interpolacja paraboliczna i metoda Brenta

Metodą lepszą od złotego podziału, gdyż efektywnie wykorzystującą wartości funkcji, jest interpolacja paraboliczna: Przez trzy punkty $(a, f_a \equiv f(a))$, (b, f_b) , (c, f_c) przeprowadzamy parabolę, a jako punkt d bierzemy jej minimum:

$$d = \frac{1}{2} \cdot \frac{a^2(f_c - f_b) + b^2(f_a - f_c) + c^2(f_b - f_a)}{a(f_c - f_b) + b(f_a - f_c) + c(f_b - f_a)} \quad (6)$$

dalej zaś postępujemy zgodnie z (4).

Uzasadnieniem dla stosowania tej metody jest fakt, iż w pobliżu minimum funkcja powinna z dobrym przybliżeniem zgadzać się ze swoim rozwinięciem w szereg Taylora do drugiego rzędu. Dzięki temu metoda ta może być zbieżna szybciej, niż liniowo, ale mogą się zdarzyć kłopoty, zwłaszcza, jeśli początkowo funkcja nie przypomina paraboli.

W związku z tym stosujemy **metodę Brenta**:

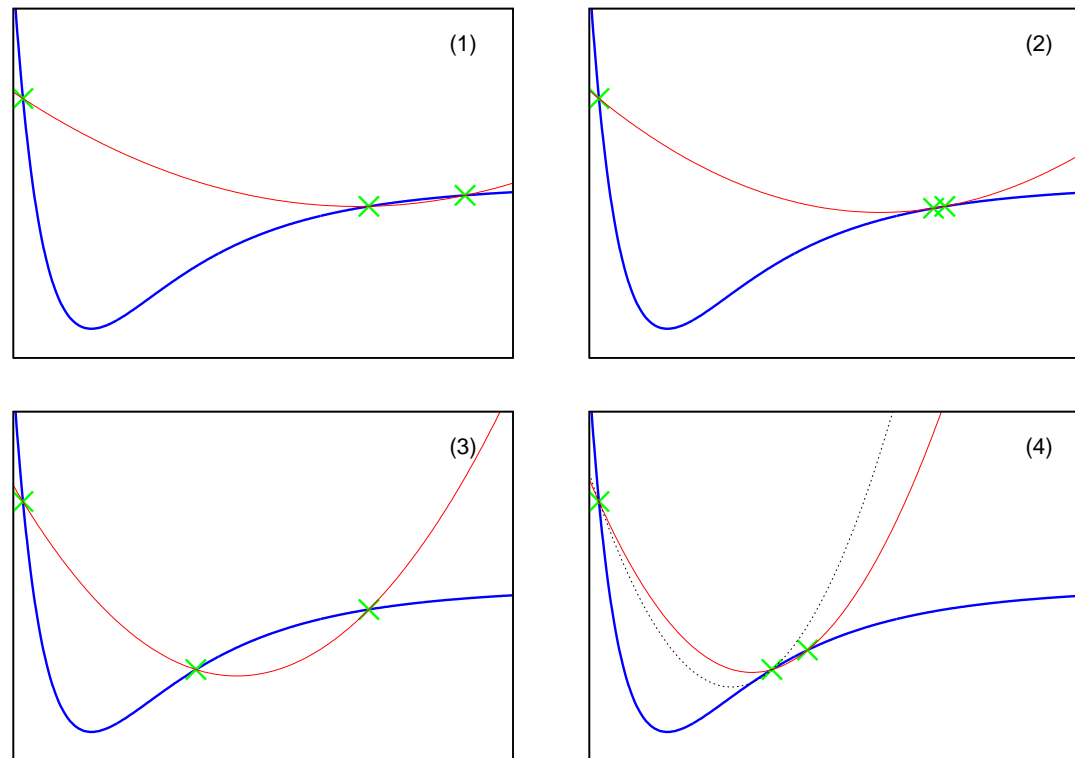
- Mając trójkę (a, b, c) , spełniającą (3), obliczamy d zgodnie ze wzorem (6).
- Akceptujemy tak obliczone d , jeżeli
 1. d leży wewnątrz przedziału, $a < d < c$, oraz
 2. Rozmiar nowego przedziału, wyznaczonego zgodnie z (4), jest mniejszy niż połowa przedziału w *przedostatniej* iteracji.
- Jeżeli nie akceptujemy nowego d , dokonujemy *bisekcji* przedziału $[a, c]$ i jako d bierzemy jego środek, a dalej postępujemy zgodnie z (4).

Powyższa metoda Brenta jest obecnie uważana za **podstawową metodę poszukiwania minimów** funkcji jednowymiarowych.

Uwaga: Dlaczego bisekcja?

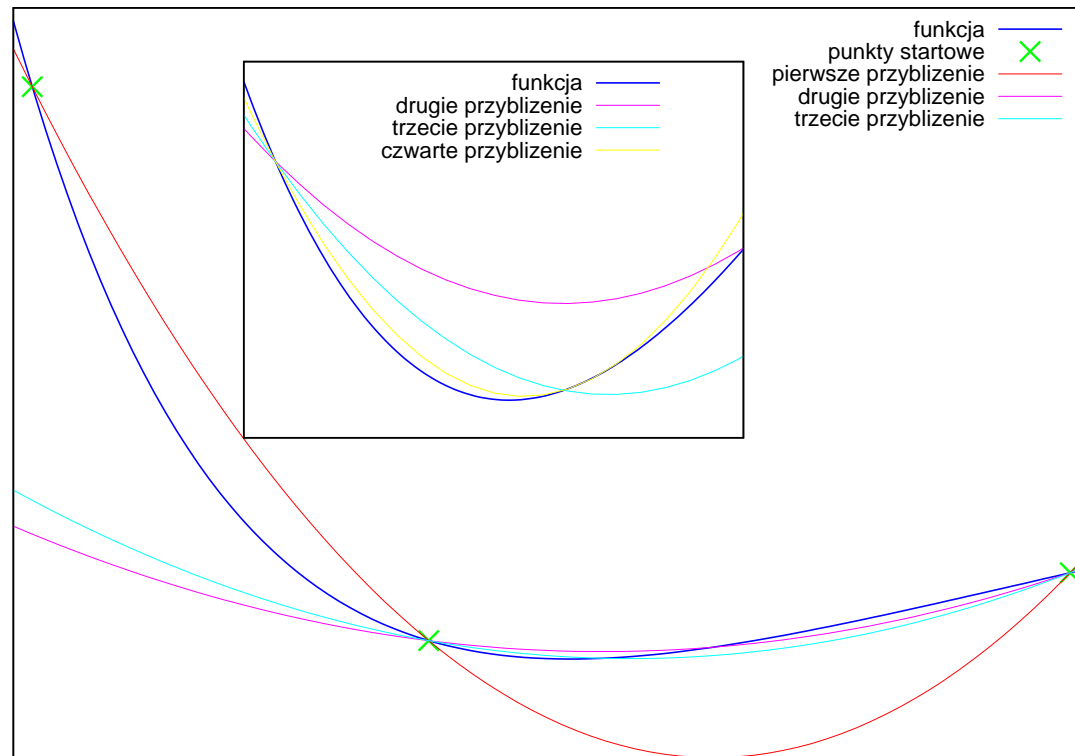
Można się zastanawiać, dlaczego w przypadku stagnacji interpolacji parabolicznej zaleca się stosowanie jednego kroku bisekcji, nie zaś jednego kroku omawianej wcześniej metody złotego podziału. Odpowiedź jest następująca: Stagnacja interpolacji parabolicznej występuje, gdy nowy punkt d (minimum paraboli) wypada bardzo blisko któregoś krańca przedziału. Z drugiej strony, metoda złotego podziału zawsze wybiera punkt w większym z dwu podprzedziałów. Gdy zaś jeden z podprzedziałów jest bardzo krótki, drugi pokrywa się niemalże z całym przedziałem, wyznaczany w metodzie bisekcji punkt środkowy na pewno znajdzie się w większym podprzedziale, a metoda bisekcji jest *odrobinę* prostsza od złotego podziału.

Przykład



Cztery kroki metody Brenta. W kroku trzecim dokonano *bisekcji*. Na ostatnim panelu linia przerywana pokazuje następną parabolę. Zbieżność jest na tym etapie powolna, gdyż początkowe przybliżenie paraboliczne jest bardzo dalekie od prawdziwego kształtu funkcji.

Uwaga!



Metoda Brenta **znacznie** przyspiesza zbieżność, gdy funkcję da się dobrze przybliżyć za pomocą paraboli.
Wstawka na rysunku pokazuje powiększenie okolic minimum.
To samo można powiedzieć o metodzie wykorzystującej pochodne, omówionej poniżej.

Wykorzystanie pochodnych

Jeżeli funkcja f jest różniczkowalna i jeżeli obliczenie jej pochodnej jest łatwe, możemy posłużyć się pochodną dla przyspieszenia zbieżności. W przypadku funkcji jednej zmiennej, **w przeciwieństwie do funkcji wielu zmiennych**, korzystanie z pochodnych nie powoduje **znacznego** przyspieszenia — niekiedy, zwłaszcza daleko od minimum, metoda zachowuje się *gorzej*, niż metoda złotego podziału.

Najprostszą metodą jest dokonanie interpolacji liniowej pochodnych* w *skrajnych* punktach przedziału†. Jako punkt d bierzemy miejsce zerowe funkcji interpolującej pochodną:

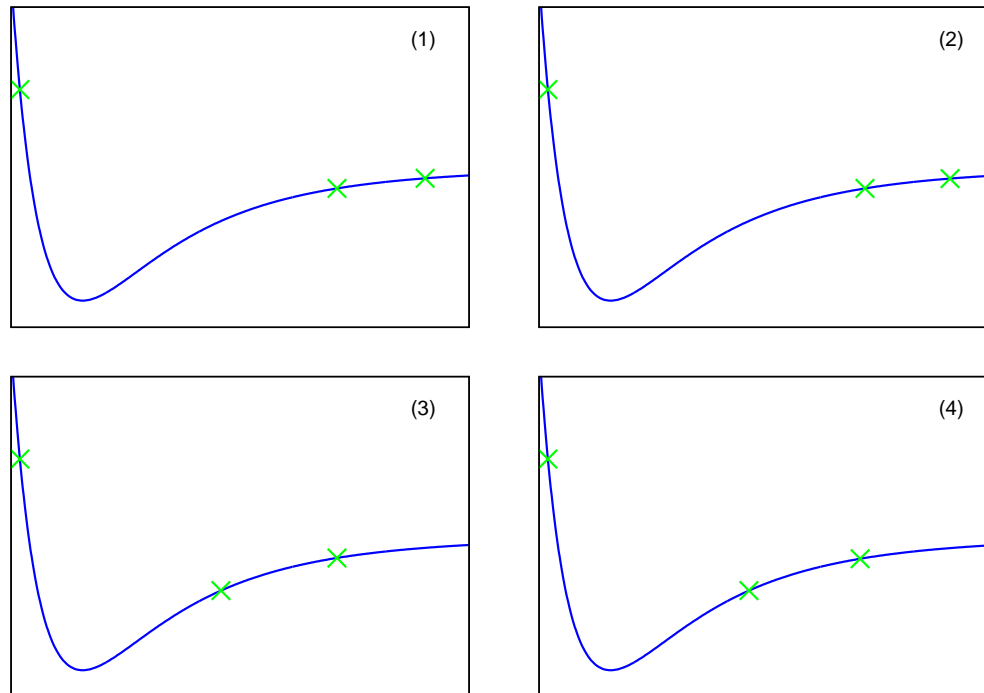
$$d = \frac{af'_c - cf'_a}{f'_c - f'_a}, \quad (7)$$

*Założenie, że pochodną można przybliżyć funkcją liniową jest równoważne założeniu, że funkcję można przybliżyć parabolą, jednak **numeryczne** wyniki mogą być różne — na niekorzyść metody wykorzystującej pochodne.

†Przy założeniu, że w skrajnych punktach pochodna ma przeciwny znak. Jeśli założenie to nie jest spełnione, wyinterpolowane miejsce zerowe pochodnej wypadnie poza przedziałem.

po czym, jak poprzednio, zawężamy przedział posługując się warunkami (4). Jeżeli metoda wpada w stagnację, wykonujemy *bisekcję* przedziału. Numeryczne własności tej metody są na ogół gorsze niż metody Brenta.

Przykład



Cztery kroki metody wykorzystującej pochodną. Skrajny prawy punkt w kroku drugim jest prawie taki sam, jak początkowy skrajny punkt, trzeba więc wykonać bisekcję (trzeci panel). Sytuacja powtarza się w kolejnym kroku. Metoda, dla tego przybliżenia początkowego, jest wolniej zbieżna niż metoda złotego podziału.

Wykorzystanie interpolacji Hermite'a

Można też użyć bardziej złożonej metody wykorzystującej pochodne: Korzystając z interpolacji Hermite'a, konstruujemy wielomian trzeciego stopnia zgadzający się z badaną funkcją i jej pochodną w skrajnych punktach przedziału, po czym jako punkt d bierzemy minimum tego wielomianu leżące w przedziale $[a, c]$, o ile takowe istnieje. Metoda ta może mieć podobne wady, co uprzednio omawiana metoda, a w dodatku cechuje się wyraźnie większą złożonością obliczeniową.

Metoda Brenta i metody wykorzystujące pochodne przyspieszają zbieżność jeżeli funkcja daje się dobrze przybliżyć za pomocą rozwinięcia Taylora do rzędu drugiego.