

Wstęp

Przez optymalizację będziemy rozumieć wyznaczanie ekstremum (maksimum lub minimum) funkcji rzeczywistej dowolnej liczby zmiennych. Zmienne od których zależy funkcja mogą być ograniczone do pewnego obszaru (optymalizacja z ograniczeniami). Jeżeli zakres tych zmiennych jest dowolny mamy do czynienia z optymalizacją bez ograniczeń. Ponieważ wyznaczanie maksimum dla funkcji f jest tym samym co wyznaczanie minimum dla funkcji $-f$, więc w rozważaniach można się ograniczyć tylko do przypadku wyznaczania minimum.

Matematyczne sformułowanie problemu (w wersji dla minimum) jest następujące: dana jest funkcja (zwana czasami *funkcją celu*)

$$f: \mathbb{R}^n \supset \Omega \rightarrow \mathbb{R}, \quad (0.1)$$

określona na pewnym podzbiorze $\Omega \subset \mathbb{R}^n$. Szukamy takiego punktu $x^* = (x_1^*, \dots, x_n^*) \in \Omega$, że

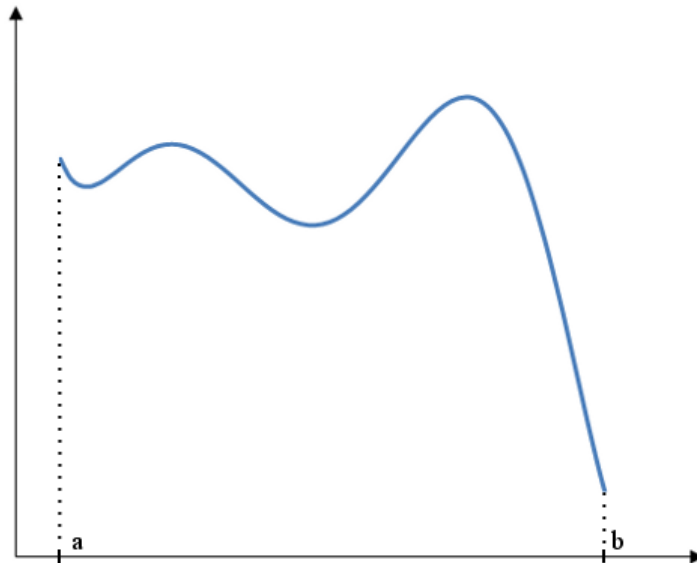
$$f(x^*) = \min_{\Omega} f = \min \{f(x) : x \in \Omega\}. \quad (0.2)$$

Powyższe sformułowanie oznacza, że dla $x^* \in \Omega$ funkcja osiąga *minimum globalne* w dziedzinie Ω . Oprócz takiego ekstremum mogą występować ekstrema lokalne (czyli największe lub najmniejsze wartości w pewnym otoczeniu punktu $x^* \in \Omega$). Z punktu widzenia zastosowań najczęściej jesteśmy zainteresowani znalezieniem minimum globalnego. Jest to problem trudniejszy niż znajdowanie tylko minimum lokalnego. (Formalnie ekstremum globalne jest jednym z ekstremów lokalnych). Większość metod optymalizacji (w szczególności te, które opiszemy dalej) znajduje zazwyczaj minima (maksima) lokalne. Używając powyższych oznaczeń możemy powiedzieć, że zagadnienie optymalizacji bez ograniczeń występuje wtedy, gdy $\Omega = \mathbb{R}^n$, czyli dla funkcji $f: \mathbb{R}^n \rightarrow \mathbb{R}$. Najprostszym przypadkiem jest funkcja kwadratowa, np. $f(x) = 2x^2 - 2x + 3$ ma minimum globalne dla $x^* = 1/2$, w którym osiąga wartość $f(x^*) = 5/2$. Oznacza to, że

$$f(x) \geq 5/2 \quad \text{dla } x \in \mathbb{R}.$$

Zauważmy jednak, że funkcja ta, której dziedzina $\Omega = \mathbb{R}$, nie posiada w ogóle maksimum (ani globalnego, ani lokalnego), gdyż wykresem jest parabola o ramionach skierowanych do góry.

Na Rys. 1 jest pokazany przykład wykresu funkcji $f: [a, b] \rightarrow \mathbb{R}$, czyli $\Omega = [a, b]$. Zauważmy, że funkcja ta posiada trzy minima lokalne, ale jedno z nich osiąga na prawym końcu, $x_{min}^* = b$, przy czym jest to jednocześnie minimum globalne. Podobnie posiada ona trzy maksima, ale maksimum globalne jest osiągnięte wewnątrz dziedziny $\Omega = [a, b]$.



Rys. 1. Funkcja jednej zmiennej, która ma trzy minima lokalne oraz trzy maksima lokalne (włączając punkty na końcach przedziału). Maksimum globalne jest osiągnięte wewnątrz przedziału, ale minimum globalne jest w prawym końcu (na brzegu dziedziny).

Jeżeli funkcja $f = f(x)$ oraz dziedzina $\Omega \subset \mathbb{R}^n$ mają pewną szczególną postać, to używane są specyficzne procedury znajdowania minimum. Można tu wyróżnić następujące kategorie: (1) funkcje jednej zmiennej; (2) funkcje liniowe wielu zmiennych, zbiór $\Omega \subset \mathbb{R}^n$ zdefiniowany przy pomocy nierówności liniowych – zagadnienie *programowania liniowego*; (3) funkcje kwadratowe wielu zmiennych. W pozostałych przypadkach należy stosować ogólne metody optymalizacji. Możemy je podzielić na gradientowe i bezpośrednie. Metody gradientowe używają pochodnej funkcji f , którą można utożsamiać z gradientem, ∇f , natomiast metody bezpośrednie odwołują się jedynie do wartościowania samej funkcji (na przykład metoda Neldera–Meada).

Optymalizacja funkcji jednej zmiennej

Jednym ze sposobów znajdowania minimum funkcji $f:[a,b] \rightarrow \mathbb{R}$ jest rozwiązywanie równania $f'(x)=0$, które jest warunkiem koniecznym na istnienie ekstremum we wnętrzu przedziału $[a,b]$. Oczywiście metodę tę można stosować, gdy funkcja jest różniczkowalna i najlepiej w sytuacji, gdy możemy ją obliczyć wzorem. Ponadto należy pamiętać, że daje ona w zasadzie tylko punkty „podejrzane” o ekstremum, czego klasycznym przykładem jest funkcja $f(x)=x^3$, dla której równanie $f'(x)=2x^2=0$ daje $x=0$. Nie jest to jednak punkt, w którym funkcja ta osiąga ekstremum.

W praktyce najbardziej skuteczne algorytmy znajdowania minimum dla funkcji jednej zmiennej to metody bezpośrednie. Ze względu też na specyfikę jednowymiarową, nie wszystkie te metody można uogólnić na przypadki funkcji wielu zmiennych $f:\mathbb{R}^n \rightarrow \mathbb{R}$.

Metody podziału

Przykładowe algorytmy opisane w tej sekcji dają ciąg zbieżny do minimum pod warunkiem, że funkcja f jest *unimodalna*. W innych przypadkach ciąg nie musi być zbieżny.

Definicja. Funkcja $f : [a, b] \rightarrow \mathbb{R}$ jest *unimodalna*, gdy (i) jest ciągła; (ii) istnieje punkt $x^* \in (a, b)$ taki, że f jest malejąca dla $x < x^*$ oraz rosnąca dla $x > x^*$.

Dalej zajmujemy się takimi wariantami metod podziału, w których w każdym kroku iteracji wybieramy dwa punkty do obliczeń. W ogólnym przypadku możemy wybierać według ustalonego schematu k punktów $a < t_1 < \dots < t_k < b$, a następnie porównywać wartości $f(t_i)$ oraz $f(t_{i+1})$ dla $i = 1, \dots, k$ tak, aby wybrać podprzedział, który zawiera szukany element optymalny x^* .

Na przykład, metoda podziału z wyborem dwóch punktów $c, d \in (a, b)$, $c < d$, polega na porównaniu wartości $f(c)$, $f(d)$, a następnie wybraniu odpowiedniego przedziału, $[a, d]$ lub $[c, b]$ zawierającego element optymalny x^* , do kolejnej takiej samej iteracji. Mianowicie

jeżeli $f(c) < f(d)$, to wybieramy $[a, d]$,

jeżeli $f(c) > f(d)$, to wybieramy $[c, b]$.

W ten sposób otaczamy szukany element coraz mniejszymi przedziałami, czyli lokalizujemy go z coraz większą dokładnością. Oczywiście wyboru przedziału $[c, d]$ możemy dokonywać na wiele sposobów, i w ten sposób powstają różne szczegółowe warianty metody podziału.

Metoda podziału na trzy części

Jako punkty podziału (dwa) wybieramy $t_1 < t_2 \in (a, b)$ tak, aby odległości pomiędzy a, t_1, t_2, b były równe (inaczej odcinki: $[a, t_1]$, $[t_1, t_2]$, $[t_2, b]$ mają tę samą długość). Zatem

$$t_1 = a + \frac{1}{3}(b - a), \quad t_2 = a + \frac{2}{3}(b - a).$$



Rys. 2. Układ punktów do metody podziału na trzy (równe) części.

Łatwo teraz opisać algorytm wyznaczania kolejnego przedziału zawierającego poszukiwany element optymalny $x^* = x_{min}$.

jeżeli $f(t_1) \leq f(t_2)$, to $x_{min} \in [a, t_2]$,

jeżeli $f(t_1) > f(t_2)$, to $x_{min} \in [t_1, b]$.

W każdej iteracji następuje zmniejszenie długości przedziału $2/3$ razy. Po n iteracjach długość przedziału, w którym znajduje się minimum wynosi $(2/3)^n(b - a)$. Prosta implementacja tego algorytmu jest przedstawiona poniżej.

```
int main() {  
    double a, b, t1, t2, x_min, y1, y2, eps;  
  
    eps = 1E-8; a = 0.0; b = 4.0;  
  
    do {  
        t1 = a + (1.0/3.0)*(b-a);  
        t2 = a + (2.0/3.0)*(b-a);  
  
        if (f(t1) <= f(t2)) b = t2;  
        else a = t1;  
  
    } while (0.5*fabs(b-a) > eps);  
  
    x_min = (a + b)/2.0;  
}
```

Rys. 3. Metoda podziału na trzy części – implementacja w języku C/C++. Fragment z programu „optimization 1D - one third points.cpp”.

Przykład 1. Znajdziemy numerycznie minimum funkcji określonej wzorem

$$f(x) = x^3 + 1,5x^2 - 6x + 1.$$

Obliczając pochodną i przyrównując od zera mamy $f'(x) = 3x^2 + 3x - 6 = 0$. Rozwiązując otrzymamy $x_1 = -2$, $x_2 = 1$. Ponieważ $f''(x) = 6x + 3$, więc $f''(-2) = -9$ i $f''(1) = +9$. Stąd dla $x = x_1$ mamy maksimum, a dla $x = x_2$ minimum. Jeżeli uruchomimy program z Rys. 3, gdzie początkowy przedział $[a, b] = [0, 4]$, to uzyskamy wynik



```
CA: C:\Users\szyszkina\Documents\Skrzynka\cpp\optimization 1D - one third points.exe  
x_min = 1  
f(x_min) = -2.5  
Press any key to continue . . .
```

Tak więc minimum, $x_{min} = 1$, zostało poprawnie zlokalizowane.

Ogólnie przy metodach podziału z wyborem dwóch punktów zauważamy pewne podobieństwo do metody bisekcji (dla równania $f(x) = 0$), ale tym razem *na ogół* dokonujemy dwóch wartościowań w każdym kroku iteracji (wybieramy dwa punkty, $c = t_1$ i $d = t_2$ w kroku iteracji, a następnie obliczamy $f(t_1)$, $f(t_2)$). Użyte sformułowanie „na ogół” sugeruje, że istnieje metoda, która ten problem – dwóch wartościowań – w jakiś sposób omija. Jest to o tyle istotne, że w przypadku algorytmów optymalizacji (podobnie jest dla równań nieliniowych), najbardziej kosztowną operacją jest właśnie wartościowanie funkcji f . Okazuje się, że przez odpowiedni wybór podziału, jeden z punktów z aktualnej iteracji pokryje się z odpowiednim punktem z iteracji kolejnej, a zatem nie trzeba będzie wykonywać drugiego wartościowania, bo będzie już znane z poprzedniego kroku! W ten sposób uzyskamy *metodę złotego podziału*. Nazwa „złoty podział” bierze się stąd, że proporcja, która to gwarantuje, $(d - a)/(b - a)$, jest słynnym *złotym podziałem*.¹

¹ Złotym podziałem interesowali się już matematycy w starożytnej Grecji. Określili oni, że *złota proporcja* oznacza taki podział całości na dwie mniejsze części, że stosunek większej części do całości jest taki sam,

Metoda złotego podziału

Założmy, że elementy $c, d \in (a, b)$ są określone przez wzory

$$\begin{aligned}t_1 &= a + \alpha^2(b - a), \\t_2 &= a + \alpha(b - a),\end{aligned}\tag{0.3}$$

gdzie $0 < \alpha < 1$. Chcemy tak dobrać współczynnik α , aby w kolejnym kroku iteracji, niezależnie od tego czy wybierzemy przedział $[a, t_2]$ czy $[t_1, b]$, jeden z punktów nowego przedziału pokrył się z którymś z poprzedniego. Prowadzi to do następujących dodatkowych równości

$$\begin{aligned}t_1 &= a + \alpha(t_2 - a), \\t_2 &= t_1 + \alpha^2(b - t_1).\end{aligned}\tag{0.4}$$

Z zależności wyrażonych wzorami (0.3) i (0.4) uzyskujemy następujące równanie na współczynnik α

$$\alpha^2 + \alpha + 1 = 0.\tag{0.5}$$

Dodatnim pierwiastkiem tego równania jest $\alpha = \frac{-1 + \sqrt{5}}{2} \approx 0,618033989$. Tak więc metodę złotego podziału możemy opisać następująco:

Dane wejściowe: funkcja unimodalna $f : [a, b] \rightarrow \mathbb{R}$, dokładność $\varepsilon > 0$.

1) Inicjalizacja:

$$\alpha = (\sqrt{5} - 1)/2, t_1 = a + \alpha^2(b - a), t_2 = a + \alpha(b - a), y_1 = f(t_1), y_2 = f(t_2)$$

2) **dopóki** $|b - a|/2 > \varepsilon$ **wykonuj**

$$\text{jeżeli } y_1 < y_2, \text{ to } b = t_2, t_2 = t_1, t_1 = a + \alpha^2(b - a), y_2 = y_1, y_1 = f(t_1)$$

$$\text{w przeciwnym razie } a = t_1, t_1 = t_2, t_2 = a + \alpha(b - a), y_1 = y_2, y_2 = f(t_2)$$

3) wypisz $x_{\min} = (a + b)/2$

Przykład realizacji tej metody w języku C/C++ jest podany na listingu poniżej. Jest on fragmentem programu *optimization 1D – golden section.cpp*.

jak stosunek mniejszej do większej. Dla odcinka o długości a oznacza to podział punktem x , który spełnia równość

$$\frac{x}{a} = \frac{a-x}{x},$$

co daje równanie $x^2 + ax + a^2 = 0$, zatem proporcja $\alpha = x/a$ spełnia równanie $\alpha^2 + \alpha + 1 = 0$. Ponieważ geometria i proporcje zajmowały poczesne miejsce w greckiej filozofii, skutkowało to m.in. uznaniem złotej proporcji za kanon piękna. Np. w rzeźbach Fidiasza czy Praksytelesa talia dzieli sylwetkę ludzką właśnie w złotym stosunku.

```
const double alpha = 0.618033989;  
const double alpha2= alpha*alpha;  
  
int main() {  
    double a, b, t1, t2, x_min, y1, y2, eps;  
  
    eps = 1E-8; a = 0.0; b = 2.0;  
  
    t1 = a + alpha2*(b-a);  
    t2 = a + alpha*(b-a);  
    y1 = f(t1); y2 = f(t2);  
    do {  
        if (y1 < y2) { b = t2; t2 = t1; t1 = a + alpha2*(b-a); y2 = y1; y1 = f(t1); }  
        else { a = t1; t1 = t2; t2 = a + alpha*(b-a); y1 = y2; y2 = f(t2); }  
    } while (0.5*fabs(b-a) > eps);  
  
    x_min = (a + b)/2.0;  
    cout << "x_min = " << x_min << "\n";  
    cout << "f(x_min) = " << f(x_min) << "\n";  
    system("pause");  
}
```

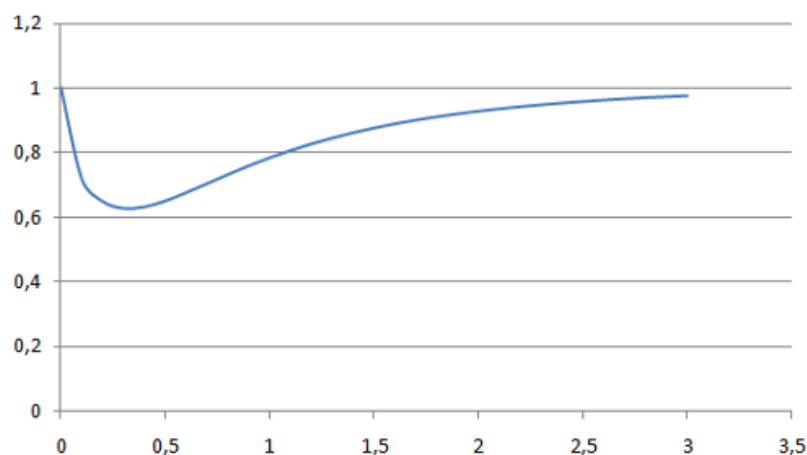
Przykład 2. Znajdziemy numerycznie minimum funkcji określonej wzorem

$$f(x) = 1 - \frac{\sqrt{x}e^{-x}}{1 + \ln(1 + x^2)}.$$

Z wykresu funkcji — Rys. 4 — widać, że do obliczeń można wybrać przedział $[0, 2]$. Po uruchomieniu programu uzyskamy wynik

```
C:\skrypt\optimization 1D - golden section.exe  
x_min = 0.325729  
f(x_min) = 0.625681  
Press any key to continue . . .
```

Zatem przybliżona wartość $x_{min} \approx 0,32573$. W punkcie tym funkcja osiąga minimum $f_{min} \approx 0,62568$.



Rys. 4. Wykres funkcji z przykładu. Sugeruje on, że jest ona umimodalna na przedziale $[0, \infty)$. Widać, że dobrym przedziałem początkowym do procedury podziału może być $[0, 2]$.

Optymalizacja funkcji wielu zmiennych

Koncepcyjnie najprostsze wydają się metody najszybszego spadku. Można je sformułować następująco: dla danego punktu startowego $x^{(0)} \in \mathbb{R}^n$, obliczmy kolejne punkty wg schematu

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}, \quad k = 0, 1, 2, \dots \quad (0.6)$$

gdzie $d^{(k)} \in \mathbb{R}^n$ jest odpowiednio dobieranym kierunkiem (wektorem) oraz $\alpha_k > 0$ jest dodatnim parametrem (czasami zwanym długością kroku, ang. *step size*). Podstawowy pomysł w powyższym schemacie polega na tym, aby w każdym kroku iteracji (0.6) poruszać się w kierunku, w którym funkcja f maleje. Ponieważ kierunek największego wzrostu funkcji jest wyznaczony przez gradient, $\nabla f(x^{(k+1)})$, więc naturalne jest, aby szukać kierunku przeciwnego do tego gradientu (czyli $-\nabla f(x^{(k+1)})$, kierunek największego spadku) lub ogólniej kierunku, który jest pod kątem ostrym do $-\nabla f(x^{(k)})$ (lub jak kto woli: pod kątem rozwartym względem $\nabla f(x^{(k)})$). Tak więc żądanie na $d^{(k)} \in \mathbb{R}^n$ jest następujące

$$d^{(k)} \cdot \nabla f(x^{(k)}) < 0. \quad (0.7)$$

(Ściśle rzecz biorąc w warunku (0.7) należy rozróżnić dwa przypadki: (i) $\nabla f(x^{(k)}) \neq 0$, wtedy warunek (0.7) może być stosowany; (ii) $\nabla f(x^{(k)}) = 0$, wtedy kładziemy $d^{(k)} = 0$, gdyż warunek (0.7) nie ma tu zastosowania. W tym przypadku zachodzi też podejrzenie, że dla $x = x^{(k)}$ jest ekstremum, bo pochodna w tym punkcie znika, $f'(x^{(k)}) = \nabla f(x^{(k)}) = 0$.)

Różne warianty metody spadku powstają teraz przez konkretny sposób wyboru $d^{(k)}$ spełniającego warunek (0.7). W szczególności można spotkać następujące metody:

- *Metoda gradientu* (inaczej zwana *metodą największego spadku*, (ang. *steepest descent method*):

$$d^{(k)} = -\nabla f(x^{(k)}). \quad (0.8)$$

- *Metody gradientu sprzężonego* (ang. *conjugate gradient methods*):²

$$d^{(k)} = -\nabla f(x^{(k)}) + \beta_k d^{(k-1)}, \quad k = 1, 2, 3, \dots \quad (0.9)$$

gdzie współczynniki $\beta_k \in \mathbb{R}$ dobiera się w taki sposób, aby kierunki $\{d^{(k)}\}$ były wzajemnie prostopadłe w odpowiednio wybranym iloczynie skalarny. Zauważmy, że schemat (0.9) zawiera informację o kierunku z poprzedniego kroku, $d^{(k-1)}$. Mamy tu więc *sprzężenie*

² Historycznie nazwa *metoda gradientu sprzężonego* odnosi się do pewnego algorytmu rozwiązywania układu równań liniowych, $Ax = b$, gdzie $A \in \mathbb{R}^{n \times n}$ (macierz $n \times n$). Oryginalna metoda – opracowana w 1952 przez Hestenesa i Stiefela – dotyczyła układów z macierzą A symetryczną dodatnio określoną. Związek pomiędzy układem $Ax = b$ a problemem optymalizacji jest następujący: Jeżeli macierz A symetryczna dodatnio określona, to punkt minimum funkcji kwadratowej $f(x) = (1/2)x \cdot Ax - b \cdot x$ jest rozwiązaniem wspomnianego układu. Pewne idee z tej klasycznej metody mogą być przeniesione do problemów optymalizacji nieliniowej i tę grupę metod powinno określać się mianem *nieliniowe metody gradientu sprzężonego*.

kierunku aktualnego z kierunkiem poprzednim – stąd nazwa *metoda gradientu sprzężonego*. Ponieważ może być tutaj wiele strategii wyboru współczynnika β_k , więc w gruncie rzeczy jest to dość szeroka klasa metod. Dwie są tutaj szczególnie znane i cenione: *metoda Fletchera-Reevesa* oraz *metoda Polaka-Ribière’a*.

- (i) Metoda Fletchera-Reevesa: współczynniki β_k do wzoru (0.9) są w tej metodzie określone następująco

$$\beta_k = \frac{\nabla f(x^{(k)}) \cdot \nabla f(x^{(k)})}{\nabla f(x^{(k-1)}) \cdot \nabla f(x^{(k-1)})} = \frac{\|\nabla f(x^{(k)})\|_2^2}{\|\nabla f(x^{(k-1)})\|_2^2}, \quad (0.10)$$

gdzie jak zwykle kropka $(\cdot)$ oznacza iloczyn skalarny w $\mathbb{R}^n$, a symbol $\|x\|_2$ normę euklidesową: $x = (x_1, \dots, x_n), y = (y_1, \dots, y_n) \in \mathbb{R}^n$, to $x \cdot y = \sum_{i=1}^n x_i y_i$, $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$.

- (ii) Metoda Polaka-Ribière’a: współczynniki β_k do wzoru (0.9) są w tej metodzie określone następująco

$$\beta_k = \frac{\nabla f(x^{(k)}) \cdot (\nabla f(x^{(k)}) - \nabla f(x^{(k-1)}))}{\nabla f(x^{(k-1)}) \cdot \nabla f(x^{(k-1)})} = \frac{\|\nabla f(x^{(k)})\|_2^2 - \nabla f(x^{(k)}) \cdot \nabla f(x^{(k-1)})}{\|\nabla f(x^{(k-1)})\|_2^2}. \quad (0.11)$$

▪ *Metoda Newtona:*

$$d^{(k)} = -H^{-1}(x^{(k)}) \nabla f(x^{(k)}), \quad (0.12)$$

gdzie $H(x)$ jest tzw. *hesjanem* funkcji f w punkcie x . W tym przypadku zakładamy, że funkcja f jest klasy C^2 (istnieje druga pochodna i jest ciągła). Hesjan jest macierzą kwadratową $n \times n$, której elementami są pochodne cząstkowe drugiego rzędu,

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}.$$

Wybór $d^{(k)}$ nie wystarczy do kompletnego opisu metody spadku, gdyż pozostaje jeszcze problem wyznaczenia współczynnika α_k w relacji (0.6). Chcemy przy tym, aby zachodził warunek

$$f(x^{(k)} + \alpha_k d^{(k)}) < f(x^{(k)}), \quad (0.13)$$

ale z jednocześnie, aby α_k nie był zbyt mały (co prowadziłoby do zbyt wolnej zbieżności). W zasadzie ogólna strategia obliczania α_k może sprowadzać się do następującego problemu minimalizacji w jednym wymiarze

$$\text{szukamy } \alpha \text{ takiego, że } \varphi(\alpha) = f(x^{(k)} + \alpha d^{(k)}) \text{ jest zminimalizowana.} \quad (0.14)$$

Niestety w przypadku dowolnej funkcji $f: \mathbb{R}^n \rightarrow \mathbb{R}$, funkcja $\varphi: \mathbb{R} \rightarrow \mathbb{R}$ zdefiniowana powyżej jest na ogół nieliniowa i problem minimalizacji określony w (0.14) należy rozwiązywać

odpowiednią jednowymiarową procedurą numeryczną. Jedną z możliwości jest technika opierająca się na aproksymacji funkcji $\varphi(\alpha) = f(x^{(k)} + \alpha d^{(k)})$ wielomianem interpolacyjnym, a następnie minimalizacji tego wielomianu (jednej zmiennej α). Przykładem takiego postępowania jest *metoda Powella*, gdzie używa się interpolacji kwadratowej.

Przykład 3. Wykorzystamy metodę największego spadku do zilustrowania jednego kroku iteracji znajdowania minimum funkcji

$$f(x, y) = x^4 + y^4 + 2x^2y^2 - 4x + 1.$$

Punkt startowy jest dany: $(x^{(0)}, y^{(0)}) = (1, 25; 1, 25)$.

Dla testu obliczmy najpierw wartość funkcji w punkcie $(x^{(0)}, y^{(0)})$:

$$f(x^{(0)}, y^{(0)}) = f(1, 25; 1, 25) = 7,7656.$$

Opieramy się na iteracji $x^{(k+1)} = x^{(k)} + \alpha d^{(k)}$, gdzie $\alpha > 0$ wybieramy tak, aby funkcja $\varphi(\alpha) = f(x^{(k)} + \alpha d^{(k)})$ była możliwie najmniejsza. Obliczamy gradient:

$$\frac{\partial f}{\partial x}(x, y) = 4x^3 + 4xy^2 - 4, \quad \frac{\partial f}{\partial y}(x, y) = 4y^3 + 4x^2y,$$

czyli $\nabla f(x, y) = (4x^3 + 4xy^2 - 4, 4y^3 + 4x^2y)$. Zatem

$$\nabla f(x^{(0)}, y^{(0)}) = \nabla f(1, 25; 1, 25) = (11, 625; 15, 625),$$

$$d^{(0)} = -\nabla f(x^{(0)}, y^{(0)}) = (-11, 625; -15, 625).$$

Można by teraz obliczyć funkcję $\varphi(\alpha)$ podstawiając

$$\varphi(\alpha) = f(x^{(0)} + \alpha d^{(0)}) = f(1, 25 - 11, 625\alpha; 1, 25 - 15, 625\alpha)$$

a następnie wykorzystać wtórna $f(x, y)$, aby uzyskać analityczną postać $\varphi(\alpha)$, którą następnie różniczkujemy względem α i pochodną przyrównujemy do zera, $\varphi'(\alpha) = 0$. W ten sposób możemy znaleźć minimum (ale nie koniecznie!) pomocniczej funkcji $\varphi(\alpha)$ i wyznaczyć $\alpha = \alpha_k$. W praktyce jednak będzie trochę prościej rachunkowo, gdy najpierw policzymy $\varphi'(\alpha)$

$$\varphi'(\alpha) = \frac{d}{d\alpha} f(x^{(k)} + \alpha d^{(k)}) = d^{(k)} \cdot \nabla f(x^{(k)} + \alpha d^{(k)}),$$

i do tego wyrażenia podstawimy punkt $(x^{(0)}, y^{(0)}) = (1, 25; 1, 25)$. Otrzymujemy

$$\nabla f(x^{(0)}, y^{(0)}) =$$

$$= (-17636, 6(\alpha - 0, 035)(\alpha^2 - 0, 252\alpha + 0, 019), -23705(\alpha - 0, 08)(\alpha^2 - 0, 179\alpha + 0, 0082)),$$

$$\varphi'(\alpha) = d^{(0)} \cdot \nabla f(x^{(0)}, y^{(0)}) = d_1^{(0)} \frac{\partial f}{\partial x}(x^{(0)}, y^{(0)}) + d_2^{(0)} \frac{\partial f}{\partial y}(x^{(0)}, y^{(0)}) =$$

$$= 575417(\alpha - 0, 048)(\alpha^2 - 0, 223\alpha + 0, 014) = 0.$$

Równanie $\varphi'(\alpha) = 0$, czyli $(\alpha - 0,048)(\alpha^2 - 0,223\alpha + 0,014) = 0$ ma tylko jeden pierwiastki $\alpha = 0,048$. (Równanie $\alpha^2 - 0,223\alpha + 0,014 = 0$ nie ma pierwiastków, gdyż $\Delta = -0,00594 < 0$.) Widać też, że pochodna $\varphi'(\alpha)$ zmienia znak z ujemnego na dodatni w punkcie $\alpha = 0,048$, więc jest to minimum. Dlatego przyjmujemy $\alpha^{(0)} = 0,048$. Obliczamy nowy punkt

$$(x^{(1)}, y^{(1)}) = (x^{(0)}, y^{(0)}) + \alpha^{(0)} d^{(0)} = (1,25; 1,25) + 0,048(-11,625; -15,625) = (0,693; 0,501).$$

Wartość funkcji w nowym punkcie $(x^{(1)}, y^{(1)})$:

$$f(x^{(1)}, y^{(1)}) = f(0,693; 0,501) = 0,763.$$

Widzimy, że wartość funkcji w nowym punkcie jest istotnie mniejsza. Kontynuując dalej tę metodę otrzymamy kolejne punkty. Wyniki są przedstawione w Tabeli 1.

Tabela 1. Punkty otrzymane metodą największego spadku dla funkcji z przykładu. Ostatnia kolumna pokazuje wartości funkcji w kolejnych punktach.

k	$x^{(k)}$	$y^{(k)}$	$f(x^{(k)}, y^{(k)})$
0	1,250	1,250	7,765
1	0,693	0,501	0,765
2	1,046	0,239	0,141
3	0,949	0,185	0,036
4	1,014	0,061	0,008