

# Metody numeryczne

## 12. Minimalizacja: funkcje wielu zmiennych

P. F. Góra

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

2021

Minimalizacja funkcji wielu — niekiedy **bardzo** wielu — zmiennych jest jednym z podstawowych zastosowań metod numerycznych. Okazuje się, że są **dwie strategie postępowania**: jedna jest właściwa, gdy jesteśmy **daleko od poszukiwanego minimum**, druga, gdy jesteśmy **blisko poszukiwanego minimum**. Żadna z tych strategii **nie gwarantuje**, że znajdziemy minimum globalne, ale jedynie, że lokalne.

Skąd wiemy, czy jesteśmy daleko, czy blisko minimum? Jeżeli nie mamy jakiegoś **matematycznego** kryterium pozwalającego nam to zweryfikować, zakładamy, że jesteśmy **daleko**.

W całym wykładzie rozważamy funkcje  $f : \mathbb{R}^N \rightarrow \mathbb{R}$ , odpowiednio wiele razy różniczkowalne.

Podstawową zasadą algorytmów minimalizacyjnych jest, że **idziemy zawsze “w dół”, w stronę malejących wartości funkcji**. To sprawia, że minimalizacja jest numerycznie prostsza od ogólnego problemu rozwiązywania układów równań algebraicznych (formalnie, matematycznie, znalezienie minimum oznacza rozwiązanie równania  $\nabla f = 0$ ), ale oznacza też, że przedstawione tu algorytmy znajdują jedynie **minimum lokalne**.

## Daleko od minimum: metoda najszybszego spadku

Daleko od minimum staramy się wyłącznie o to, aby w kolejnych krokach naszej procedury wartości funkcji malały. Ponieważ gradient funkcji wielu zmiennych pokazuje kierunek jej najszybszego wzrostu, kierunek przeciwny pokazuje kierunek jej najszybszego spadku. A ponieważ zależy nam, aby funkcja malała tak szybko, jak to możliwe, podążamy w tym właśnie kierunku. Otrzymujemy w ten sposób **algorytm najszybszego spadku** (ang. *steepest descent* lub *gradient descent*):

Niech aktualne położenie wynosi  $\mathbf{x}_k$ ,  $f_k = f(\mathbf{x}_k)$ .

1.

$$\mathbf{p}_k = -\nabla f(\mathbf{x}_k) \quad (1a)$$

2.  $\gamma = \gamma_{\text{start}}$

3.

$$\tilde{\mathbf{x}} = \mathbf{x}_k + \gamma \mathbf{p}_k \quad (1b)$$

$$\tilde{f} = f(\tilde{\mathbf{x}}) \quad (1c)$$

4. Jeżeli  $\tilde{f} < f_k$

- $\mathbf{x}_{k+1} = \tilde{\mathbf{x}}$ ,  $f_{k+1} = \tilde{f}$
- $k = k + 1$
- goto 1

5. Jeżeli  $\tilde{f} \geq f_k$

- opcjonalnie: zmniejsz  $\gamma$ , potem goto 3
- STOP

Ta metoda jest nieprecyzyjna, nie dostarcza kryterium czy jesteśmy blisko minimum i w zamyśle służy do szybkiego przemieszczenia się w okolice minimum. Często — zwłaszcza gdy ciąg kolejnych wartości  $f_k$  maleje *co-raz wolniej*, co interpretujemy w ten sposób, że krajobraz się “wyplaszcza” — jest to *jedyna* metoda, jakiej się w praktyce używa. Należy jednak pamiętać, że jej wyniki na ogół są nieprecyzyjne i jeżeli nam na lepszym, bardziej dokładnym zlokalizowaniu minimum, powinniśmy użyć jakichś innych metod, przynajmniej w końcowej fazie minimalizacji, pod warunkiem, że nie będzie to zbyt kosztowne ze względu na wymiarowość problemu.

## Z punktu widzenia Big Data...

W ciągu ostatnich kilkunastu lat rozmiary dostępnych zbiorów danych rosną *szybciej* niż rośnie prędkość procesorów (nawet po uwzględnieniu paralelizacji i obliczeń na GPU). Z tego punktu widzenia możliwości uczenia maszynowego są ograniczone raczej przez możliwości obliczeniowe niż przez dostępne zbiory danych. Tę klasę problemów zwyczajowo określa się jako *Big Data*.

Uczenie maszynowe bardzo często oznacza konieczność minimalizowania funkcji

$$Q(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^N Q_i(\mathbf{X}; x_i, y_i) \quad (2)$$

gdzie

- $\mathbf{X}$  jest wielowymiarową zmienną, po której minimalizujemy — wektorem estymatorów, których wartości chcemy znaleźć. Tego typu problemy pojawiają się w zagadnieniu najmniejszych kwadratów.
- $\forall i: Q_i \geq 0$ ,
- $Q_i$  mają taką samą postać funkcyjną, różnią się tylko elementami  $x_i, y_i$  (elementami zbioru uczącego),
- $N \gg 1$  ( $N$  jest zazwyczaj **BARDZO** duże, rzędu kilkudziesięciu, niekiedy kilkuset tysięcy).



Problem (2) można rozwiązać za pomocą metody najszybszego spadku, którejś z bardziej precyzyjnych metod minimalizacji lub, jeżeli mowa o liniowym zagadnieniu najmniejszych kwadratów, za pomocą przybliżonego rozwiązywania nadokreślonych układów równań przy pomocy SVD. Jednak dla **bardzo dużych  $N$** , czyli dla bardzo dużych zbiorów uczących, inne podejście może być bardziej efektywne.

Punktem wyjścia jest metoda najszybszego spadku: W każdym kroku poruszamy w kierunku minus gradientu funkcji  $Q(\mathbf{X})$ , teoretycznie aż do osiągnięcia minimum kierunkowego.

Jak jednak pamiętamy, **daleko od minimum nie minimalizujemy**, tylko poruszamy z arbitralnie dobranym krokiem w kierunku ujemnego gradientu.

Dlatego zamiast minimalizować, w  $n$ -tym kroku iteracji przyjmujemy

$$\begin{aligned}\mathbf{X}_{n+1} &= \mathbf{X}_n - \gamma \cdot \nabla_{\mathbf{X}} Q(\mathbf{X})|_{\mathbf{X}_n} \\ &= \mathbf{X}_n - \gamma \cdot \frac{1}{N} \sum_{i=1}^N \nabla_{\mathbf{X}} Q_i(\mathbf{X}; x_i, y_i)|_{\mathbf{X}_n} .\end{aligned}\quad (3)$$

$\gamma = \text{const}$  i jest nazywane *prędkością uczenia* (ang. *learning rate*).

**Uwaga:** Przyjęcie stałego kroku nie jest aż tak naiwne, jak by się to mogło wydawać. Jeśli rozpatrzmy układ równań różniczkowych typu

$$\frac{dy}{dx} = f(y) \quad (4)$$

to najprostszą (zdecydowanie nie najlepszą, ale najprostszą i najszybszą) metodą jego numerycznego rozwiązywania jest *metoda Eulera*

$$y_{n+1} = y_n + h \cdot f(y_n), \quad (5)$$

gdzie  $h$  jest stałym krokiem narastania zmiennej niezależnej  $x$ . Metoda najszybszego spadku (3) ze stałym krokiem  $\gamma$  ma taką samą postać, co metoda Eulera, po utożsamieniu  $f \equiv -\nabla Q$ .

## Stochastic Gradient Descent

Jeśli w (3)  $N \gg 1$ , czas obliczania sumy gradientów cząstkowych  $\sum_i^N \nabla Q_i$  może być bardzo znaczny. Zarazem wyrażenie  $\frac{1}{N} \sum_i^N \nabla Q_i$  ma postać średniego gradientu cząstkowego. Jeśli założymy, że **poszczególne elementy tej sumy nie odbiegają zbytnio od średniej**, możemy średni, kosztowny w wyliczaniu gradient, zastąpić *losowo wybranym* gradientem cząstkowym. Otrzymujemy zatem algorytm ***Stochastic Gradient Descent***:

- wybierz losowo  $k \in \{1, 2, \dots, N\}$
- przyjmij

$$\mathbf{X}_{n+1} = \mathbf{X}_n - \gamma \nabla_{\mathbf{X}} Q_k(\mathbf{X}; x_k, y_k) |_{\mathbf{X}_n} \quad (6)$$

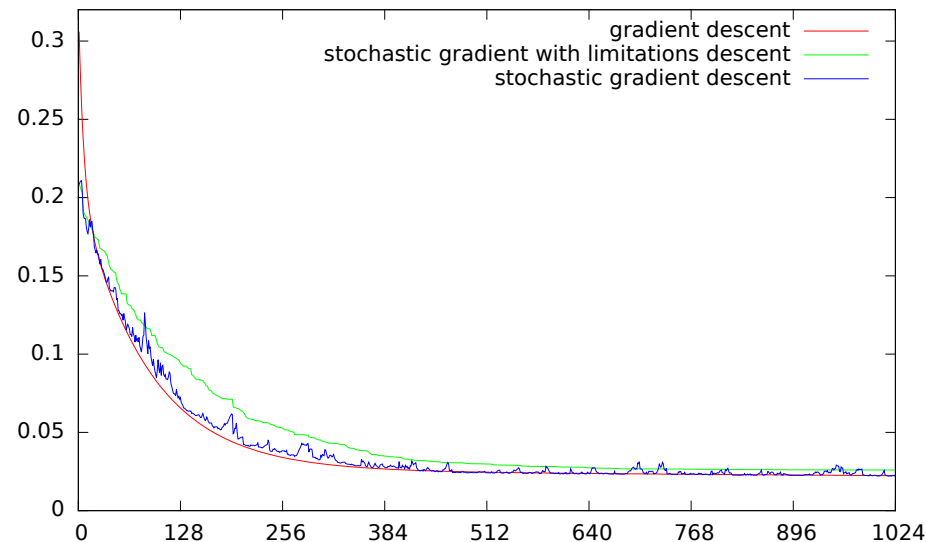
- **Dodatkowo akceptujemy powyższy krok tylko jeżeli  $Q(\mathbf{X}_{n+1}) < Q(\mathbf{X}_n)$ . Jeśli ta nierówność nie zachodzi, nie wykonujemy kroku, ale losujemy nowe  $k$ . Ten wariant nazwiemy *Stochastic Gradient Descent z ograniczeniem*.**

Iterację kończymy albo po wykonaniu z góry określonej liczby kroków, albo — częściej — gdy różnica  $|Q(\mathbf{X}_n) - Q(\mathbf{X}_{n+1})| < \varepsilon$ , gdzie  $\varepsilon$  jest ustaloną tolerancją.

Wartość *prędkości uczenia*  $\gamma$  oraz tolerancję  $\varepsilon$  dobieramy “eksperymentalnie”, to znaczy kierując się znajomością natury problemu i doświadczeniem uzyskanym przy rozwiązywaniu podobnych problemów.

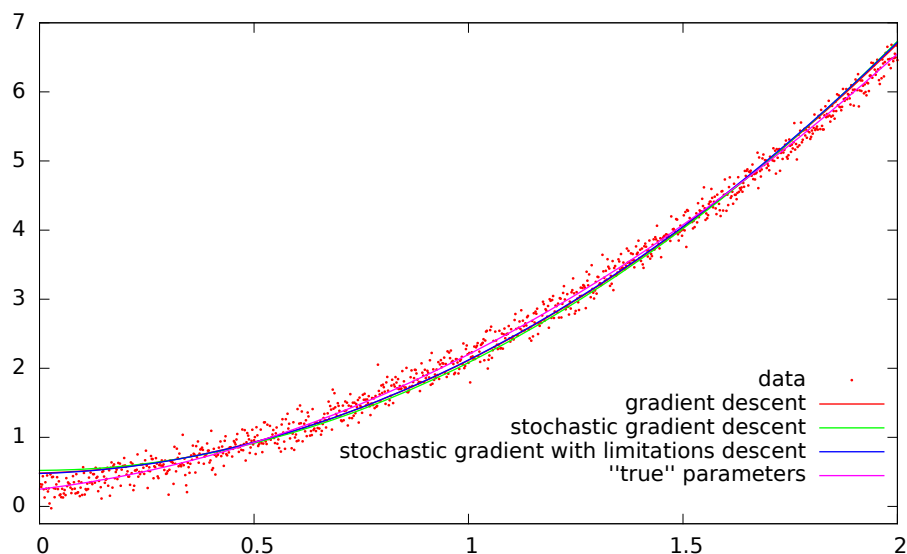
Okazuje się, że metoda *Stochastic Gradient Descent* działa zdumiewająco dobrze dla bardzo dużych zbiorów uczących. Co ciekawe, przyjęcie warunku, iż *wartość funkcji  $Q$  musi maleć po wykonaniu kroku* na ogół *pogarsza* wyniki: Od czasu do czasu można/trzeba wykonać krok “w złą stronę”.

## Przykład



Wygenerowano  $N = 1024$  punktów  $y_i = ax_i^2 + bx_i + c + \xi_i$ , gdzie  $\{\xi_i\}_{i=1}^N$  są liczbami o rozkładzie normalnym, natomiast  $x_i \in [0, 2]$ . Rysunek przedstawia kolejne wartości funkcji  $Q(a, b, c)$  uzyskane w metodzie najszybszego spadku (czerwony), w metodzie *Stochastic Gradient Descent* (niebieski) oraz *Stochastic Gradient Descent* z ograniczeniami (zielony).  $\gamma = 1/128$ . W przypadku *Stochastic Gradient Descent* wartość minimalizowanej funkcji niekiedy rośnie, ale po dostatecznie dużej liczbie iteracji wyniki

tej metody są nieco *lepsze*, niż dla metody z ograniczeniem. Co ważniejsze, wyniki są porównywalne z wynikami uzyskanymi w obliczeniowo droższej metodzie najszybszego spadku.



Dopasowane krzywe są niemalże nieodróżnialne w obszarze odpowiadającym danym wejściowym (zawartości zbioru uczącego).

## Zastosowanie metody Newtona

Formalnie znalezienie minimum funkcji wielu zmiennych jest równoważne z rozwiązaniem równania

$$\nabla f = 0 \quad (7)$$

Można spróbować rozwiązywać to równanie za pomocą wielowymiarowej metody Newtona. W kroku Newtonowskim pojawiłaby się macierz pochodnych cząstkowych wektora  $\nabla f$ ; jest to oczywiście hessjan, czyli macierz *drugich* pochodnych cząstkowych funkcji  $f$ :

$$\mathbf{H}_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}, \quad i, j = 1, \dots, N \quad (8)$$

Ostatecznie metoda Newtona miałaby postać

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}^{-1}(\mathbf{x}_k) \nabla f|_{\mathbf{x}_k} \quad (9)$$

Tak hessjan, jak i gradient należy wyliczać w bieżącym punkcie  $\mathbf{x}_k$ .



## Problemy z metodą Newtona

Metoda Newtona w postaci (9) nastrocza kilka problemów:

- Dla  $N \gg 1$  zastosowanie metody Newtona jest **monstrualnie** kosztowne
- Nawet dla  $N$  rzędu kilku-kilkunastu, wielokrotne wyliczanie pochodnych cząstkowych, a następnie rozwiązywanie układu równań  $\mathbf{H}\mathbf{z} = -\nabla f$ , może być drogie i uciążliwe
- Jeżeli minimalizowana funkcja **nie** jest formą kwadratową (a na ogół nie jest!), krok Newtonowski może być zbyt długi; metoda może wtedy “przestrzelić” nad poszukiwanym minimum
- **Warunkiem koniecznym** na to, aby iteracja (9) prowadziła do zmniejszania się wartości funkcji, jest symetria i **dodatnia określoność** wszystkich napotykanym po drodze hessjanów. Istotnie, aby po wykonaniu

kroku Newtonowskiego minimalizowana funkcja malała, musi zachodzić

$$(\mathbf{x}_{k+1} - \mathbf{x}_k)^T \nabla f|_{\mathbf{x}_k} < 0 \quad (10a)$$

— iloczyn skalarny gradientu funkcji i przesunięcia, czyli rzut przesunięcia na gradient, jest ujemny, co oznacza, że przesunięcie zostało wykonane w kierunku *malejących* wartości funkcji. Po formalnym rozwiązaniu (9) ze względu na  $\nabla f|_{\mathbf{x}_k}$  i wstawieniu wyniku do powyższego równania, daje to

$$- (\mathbf{x}_{k+1} - \mathbf{x}_k)^T \mathbf{H}(\mathbf{x}_{k+1} - \mathbf{x}_k) < 0 \quad (10b)$$

co jest spełnione, jeżeli  $\mathbf{H}$  jest dodatnio określone.

## Dodatnia określoność hessjanu

Możemy bezpiecznie założyć, że jeśli minimalizowana funkcja jest dostatecznie gładka, hessjan jest w każdym punkcie symetryczny. Niestety, nie można tego założyć odnośnie do dodatniej określoności.

W minimum hessjan jest dodatnio określony. Na podstawie ciągłości drugich pochodnych wnioskujemy, że musi istnieć pewne **otoczenie minimum, w którym hessjan jest dodatnio określony**. Jednak daleko od minimum hessjan dodatnio określony być nie musi. Możemy traktować **dodatnią określoność hessjanu** jako matematyczny warunek tego, że jesteśmy **blisko minimum**.

## Metoda Levenberga-Marquardta

Daleko od minimum stosowanie metody najszybszego spadku ma sens. Jak jednak zobaczymy, metoda ta nie najlepiej nadaje się do precyzyjnego określenia położenia minimum. Chcielibyśmy skonstruować kryterium pozwalające stwierdzić, czy jesteśmy już na tyle blisko minimum, że warto/można przełączyć się na metody bardziej precyzyjne, jeśli zachodzi taka potrzeba.

Metodą taką jest metoda Levenberga-Marquardta. **Zaleca** się jest stosowanie, gdy

- wymiarowość problemu nie jest bardzo duża,  $N \sim 100$  co najwyżej
- obliczanie drugich pochodnych jest możliwe i nie jest zbyt uciążliwe
- gdy koszt obliczania drugich pochodnych **nie** jest wielki ze względu na strukturę problemu (jak w Big Data, gdzie wymiarowość problemu

— liczba dopasowywanych parametrów — może być niewielka, ale minimalizowana funkcja jest sumą **BARDZO** wielu składników).

Metoda Levenberga-Marquardta wywodzi się z metody Newtona. Daleko od minimum hessjan nie musi być nawet dodatnio określony, co powoduje, iż krok newtonowski wcale nie musi prowadzić do spadku wartości funkcji (por. (10b)). My jednak chcemy aby wartość funkcji w kolejnych krokach spadała. Zmodyfikujmy więc hessjan:

$$\widetilde{\mathbf{H}}_{ii} = (1 + \lambda) \frac{\partial^2 f}{\partial x_i^2}, \quad (11a)$$

$$\widetilde{\mathbf{H}}_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}, \quad i \neq j, \quad (11b)$$

przy czym  $\lambda \geq 0$ .

Zauważmy, że zachodzi jedna z dwu możliwych sytuacji: (i) jeśli znajdujemy się w basenie atrakcji minimum, wówczas dla odpowiednio dużego  $\lambda$  macierz (11) stanie się dodatnio określona lub też (ii) jeśli dla żadnego dodatniego  $\lambda$  macierz (11) nie staje się dodatnio określona, oznacza to, że znajdujemy się na monotonicznej gałęzi funkcji, poza basenem atrakcji minimum.

Rozpoczynamy z jakimś niewielkim  $\lambda$ , na przykład  $\lambda = \lambda_0 = 2^{-10} = 1/1024$ . Przypuśćmy, iż aktualnym przybliżeniem minimum jest punkt  $\mathbf{x}_k$ . Dostajemy zatem...

## Algorytm Levenberga-Marquardta

1. Oblicz  $\nabla f|_{\mathbf{x}_k}$ .
2. Oblicz  $\widetilde{\mathbf{H}}(\mathbf{x}_k)$ .
3. Oblicz

$$\mathbf{x}_{\text{test}} = \mathbf{x}_k - \widetilde{\mathbf{H}}^{-1}(\mathbf{x}_k) \nabla f|_{\mathbf{x}_k} . \quad (12)$$

4. Jeżeli  $f(\mathbf{x}_{\text{test}}) > f(\mathbf{x}_k)$ , to
  - (a)  $\lambda \rightarrow 8\lambda$  (można też powiększać o inny znaczny czynnik).
  - (b) Idź do punktu 2.
5. Jeżeli  $f(\mathbf{x}_{\text{test}}) < f(\mathbf{x}_k)$ , to
  - (a)  $\lambda \rightarrow \lambda/8$  (można też zmniejszać o inny znaczny czynnik).
  - (b)  $\mathbf{x}_{k+1} = \mathbf{x}_{\text{test}}$ .
  - (c) Idź do punktu 1.

## Komentarz

Daleko od minimum *nie minimalizujemy* (nie poszukujemy minimów kierunkowych), a jedynie *podążamy w kierunku malejących wartości funkcji*.

Dodatkowo, jeśli  $\lambda > \lambda_{\max} \gg 1$ , uznajemy, iż znajdujemy się poza basenem atrakcji minimum i algorytm zawodzi. Jeśli natomiast  $\lambda < \lambda_{\min} \ll 1$ , macierz  $\widetilde{\mathbf{H}}$  jest w praktyce równa hessjanowi, który jest dodatnio określony, o czym z kolei świadczy fakt, że uzyskujemy malejące wartości funkcji  $f$ . Modyfikacja (11) przestaje być potrzebna, możemy za to przerzucić się na metody właściwe dla bliskich okolic minimum.

Ponadto w celu przyspieszenia obliczeń, jeżeli  $f(\mathbf{x}_{\text{test}}) < f(\mathbf{x}_k)$ , możemy *chwilowo* zrezygnować ze zmniejszania  $\lambda$  i modyfikowania  $\widetilde{\mathbf{H}}$  i przeprowadzić kilka kroków z tą samą macierzą, a więc korzystając z tej samej faktoryzacji.



Zauważmy, iż przypadek  $\lambda \gg 1$  oznacza, iż jesteśmy daleko od minimum. Z drugiej strony jeśli  $\lambda \gg 1$ , macierz  $\widetilde{\mathbf{H}}$  staje się w praktyce diagonalna, a zatem

$$\begin{aligned} \mathbf{x}_{\text{test}} &\simeq \mathbf{x}_k - (1 + \lambda)^{-1} \text{diag} \left\{ \left( \frac{\partial^2 f}{\partial x_1^2} \right)^{-1}, \left( \frac{\partial^2 f}{\partial x_2^2} \right)^{-1}, \dots, \left( \frac{\partial^2 f}{\partial x_N^2} \right)^{-1} \right\} \nabla f|_{\mathbf{x}_k} \\ &\simeq \mathbf{x}_i - \text{const} \cdot \nabla f|_{\mathbf{x}_k}, \end{aligned} \quad (13)$$

o ile drugie pochodne cząstkowe w poszczególnych kierunkach nie różnią się znacznie od siebie. Widać, iż daleko od minimum, gdzie warunek zachowujący raz osiągnięte minima kierunkowe nie zachodzi, algorytm Levenberga-Marquardta zachowuje się prawie jak metoda najszybszego spadku.

## Strategia precyzyjnej minimalizacji wielowymiarowej

Jeżeli jesteśmy dostatecznie blisko minimum — co stwierdzamy albo na podstawie dodatniej określoności hessjanu, albo wiedząc, że krajobraz daleko od minimum (minimów) ma “strukturę lejka” (*funnel structure*), to znaczy dla dużych wartości argumentów funkcja szybko maleje i ryzyko rozbieżności jest niewielkie — można, a niekiedy wręcz trzeba, zastosować metody, które pozwolą na bardziej precyzyjną, a przy tym możliwie szybką lokalizację minimum. Metody te wykorzystują poznane już metody minimalizowania funkcji jednej zmiennej. Przedstawimy strategię poszukiwania (lokalnego) minimum tej funkcji w postaci [ciągu minimalizacji jednowymiarowych](#).

1. Aktualnym przybliżeniem minimum jest punkt  $\mathbf{x}_k$ .
2. Wybieramy pewien kierunek poszukiwań  $\mathbf{p}_k$ .
3. Konstruujemy funkcję  $g_k: \mathbb{R} \rightarrow \mathbb{R}$

$$g_k(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{p}_k). \quad (14)$$

Zauważmy, że funkcja  $g_k(\alpha)$  **jest funkcją jednej zmiennej**. Geometrycznie jest to “ślad”  $N+1$ -wymiarowego wykresu funkcji  $f(\mathbf{x})$  przeciętego płaszczyzną zawierającą punkt  $\mathbf{x}_k$  i wektor  $\mathbf{p}_k$ .

4. Znanymi metodami jednowymiarowymi znajdujemy  $\alpha_{\min}$  takie, że funkcja (14) osiąga minimum. Jest to **minimum kierunkowe** funkcji  $f$ .
- 5.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_{\min} \mathbf{p}_k. \quad (15)$$

6. *goto* 1.

## Przykład

Niech  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ ,  $f(x_1, x_2) = 2.5x_1^2 + x_1x_2 + x_2^2 - x_1 - x_2$ . Przypuśćmy, że dany jest punkt  $\mathbf{x}_k = (1, 2)$  oraz kierunek poszukiwań  $\mathbf{p}_k = [-1, 1]^T$ . Wówczas

$$\begin{aligned} g_k(\alpha) &= f(\mathbf{x}_k + \alpha\mathbf{p}_k) \\ &= 2.5(1 - \alpha)^2 + (1 - \alpha)(2 + \alpha) + (2 + \alpha)^2 - (1 - \alpha) - (2 + \alpha) \\ &= 2.5\alpha^2 - 2\alpha + 5.5 \end{aligned} \quad (16)$$

Jest to funkcja jednej zmiennej,  $\alpha$ . Osiąga ona minimum dla  $\alpha = \frac{2}{5}$ . Zatem *następnym* punktem będzie

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{2}{5}\mathbf{p}_k = \left(1 - \frac{2}{5}, 2 + \frac{2}{5}\right) = (0.6, 2.4) \quad (17)$$

Wystarczy teraz wybrać kolejny kierunek poszukiwań  $\mathbf{p}_{k+1}$  etc.

## Jak wybierać kierunki poszukiwań?

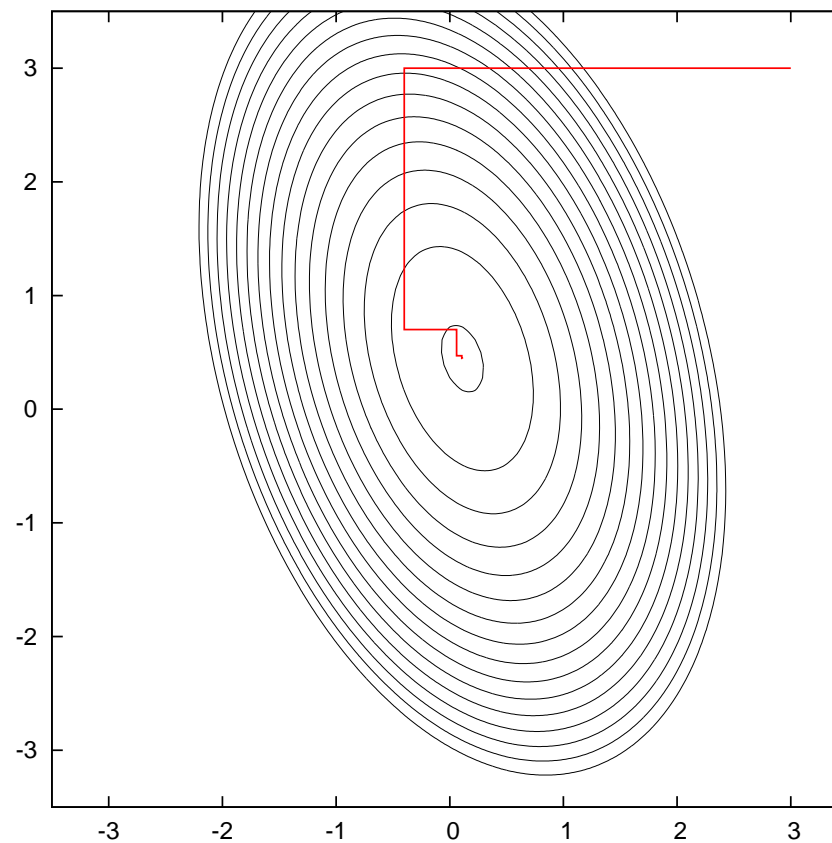
Cały problem sprowadza się zatem do wyboru odpowiedniej sekwencji kolejnych kierunków  $\{p_k\}$ .

Następujące strategie wyboru kierunków poszukiwań są bardzo popularne:

- Minimalizacji po współrzędnych — kolejnymi kierunkami poszukiwań są kierunki równoległe do kolejnych osi układu współrzędnych.
- *Metoda najszybszego spadku*, w której kierunek poszukiwań pokrywa się z minus gradientem minimalizowanej funkcji w aktualnym punkcie.

Okazuje się, że jeśli jesteśmy blisko minimum, nie są to dobre pomysły, gdyż prowadzą do wielu drobnych kroków, które częściowo likwidują efekty osiągnięte w krokach poprzednich. Dlaczego?

## Przykład — minimalizacja po współrzędnych

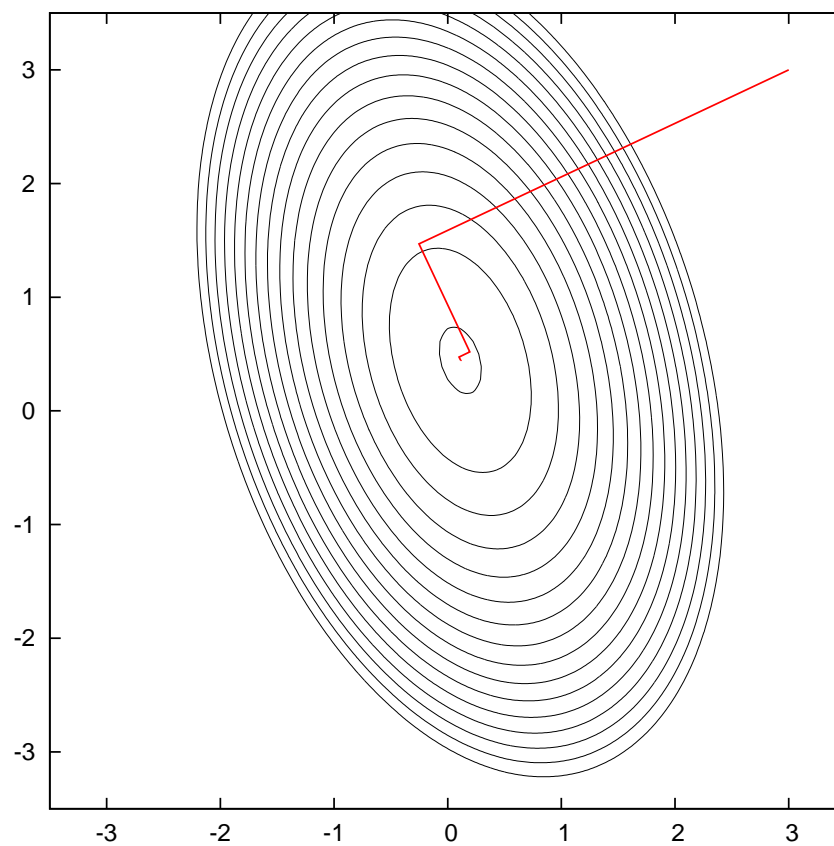


Dużo małych kroków. Osie układu nijak się mają do “naturalnej” geometrii minimalizowanej funkcji.

W metodzie najszybszego spadku co prawda realizujemy najważniejszy pomysł na minimalizację — **zawsze podążamy w kierunku malejących wartości** (czyli kierunkiem poszukiwań jest **minus gradient funkcji w aktualnym punkcie**) — ale czasami poszukiwanie minimów kierunkowych bywa niepotrzebnie kosztowne. W dodatku, jeśli jesteśmy już blisko minimum, okazuje się, że metoda najszybszego spadku także prowadzi do konieczności wykonywania wielu małych kroków, które częściowo niwelują efekty osiągnięte w krokach poprzednich.

Przy precyzyjnej lokalizacji minimum, **główny**, największy koszt obliczeniowy ponosimy wykonując wiele drobnych kroków końcowych, już w bezpośrednim sąsiedztwie minimum. Gdyby udało nam się zredukować liczbę tych kroków, odnieśliśmyby znaczny zysk na kosztach obliczeniowych.

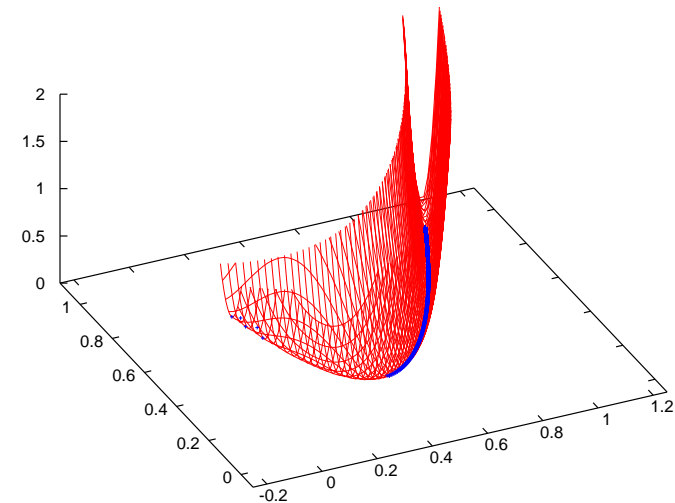
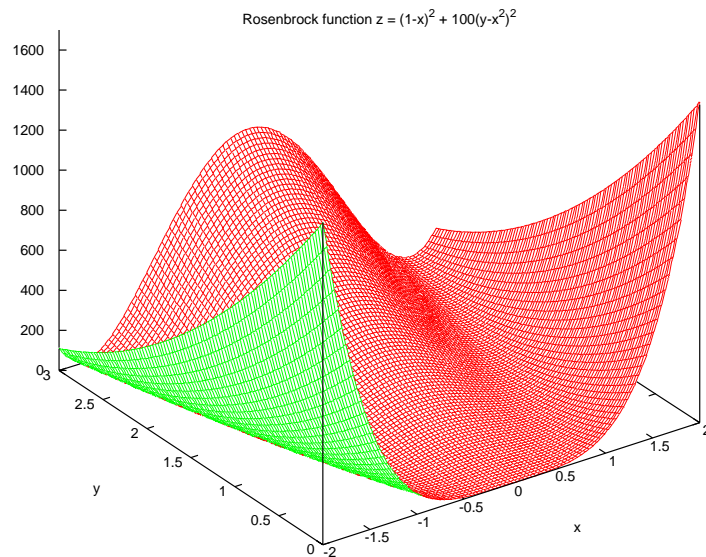
## Przykład — metoda najszybszego spadku



*Wygląda lepiej, ale kroków prawie tyle samo*



## Funkcja Rosenbrocka



Funkcja Rosenbrocka  $f(x, y) = (1 - x)^2 + 100(y - x^2)^2$  jest przykładem funkcji, którą trudno zminimalizować. W klasycznym zadaniu minimalizuje się tę funkcję startując z punktu  $(-3, -4)$ . Wartość funkcji w tym punkcie wynosi 16916, zaś gradient to  $[-15608, -2600]^T$ , należy zatem poruszać się *w kierunku* minus gradientu, ale *o bardzo niewielki ułamek* jego długości. Prawy panel pokazuje końcowy przebieg (niezbyt udanej) minimalizacji funkcji Rosenbrocka za pomocą metody najszybszego spadku.

## W pobliżu minimum

Znajdźmy warunek na to, aby  $f$  osiągała minimum kierunkowe, czyli aby  $g_k$  osiągała minimum:

$$\frac{dg_k}{d\alpha} = \sum_i \frac{\partial f}{\partial x_i} (\mathbf{p}_k)_i = \left( \nabla f |_{\mathbf{x}=\mathbf{x}_{\min}} \right)^T \mathbf{p}_k = 0. \quad (18)$$

**W minimum kierunkowym gradient funkcji jest prostopadły do kierunku poszukiwań.** Zatem w metodzie najszybszego spadku kierunek poszukiwań (lokalny kierunek minimalizacji) co prawda zaczyna się prostopadle do poziomnic funkcji, ale kończy się *stycznie* do poziomnic. Natomiast w minimalizacji po współrzędnych kolejne kierunki poszukiwań, czyli — tutaj — kolejne współrzędne, nie zależą od kształtu minimalizowanej funkcji; taka strategia nie może być optymalna.

## Przybliżenie formy kwadratowej

Przypuśćmy, że jesteśmy dostatecznie blisko minimum. Rozwijamy minimalizowaną funkcję w szereg Taylora wokół minimum i otrzymujemy

$$f(\mathbf{x}) \simeq \frac{1}{2}\mathbf{x}^T \mathbf{A}\mathbf{x} - \mathbf{b}^T \mathbf{x} + f_0, \quad (19)$$

gdzie  $\mathbf{A}$  jest macierzą drugich pochodnych cząstkowych (hessjanem) obliczanym w minimum. Z definicji minimum, macierz ta jest dodatnio określona, jeśli zaś funkcja jest dostatecznie gładka, macierz ta jest symetryczna. Zatem w pobliżu minimum, funkcja w przybliżeniu zachowuje się jak dodatnio określona forma kwadratowa.

## Gradienty sprzężone

W przybliżeniu (19) gradient funkcji  $f$  w punkcie  $\mathbf{x}_k$  wynosi

$$\nabla f|_{\mathbf{x}=\mathbf{x}_k} = \mathbf{A}\mathbf{x}_k - \mathbf{b}. \quad (20)$$

Kolejne poszukiwania odbywają się w kierunku  $\mathbf{p}_{k+1}$ . Gradient funkcji w pewnym nowym punkcie  $\mathbf{x} = \mathbf{x}_k + \alpha\mathbf{p}_{k+1}$  wynosi

$$\nabla f|_{\mathbf{x}} = \mathbf{A}\mathbf{x}_k + \alpha\mathbf{A}\mathbf{p}_{k+1} - \mathbf{b}. \quad (21)$$

*Zmiana gradientu* wynosi

$$\delta(\nabla f) = \alpha\mathbf{A}\mathbf{p}_{k+1}. \quad (22)$$

Punkt  $\mathbf{x}_k$  jest minimum kierunkowym w kierunku  $\mathbf{p}_k$ , a więc gradient funkcji w tym punkcie spełnia warunek (18). *Jeżeli chcemy aby poszukiwania w nowym kierunku nie zepsuły minimum kierunkowego w kierunku  $\mathbf{p}_k$ , zmiana gradientu musi być prostopadła do starego kierunku poszukiwań,  $\delta(\nabla f)^T \mathbf{p}_k = 0$ .* Tak jednak musi być dla *wszystkich* poprzednich kierunków, nie chcemy bowiem naruszyć żadnego z poprzednich minimów kierunkowych. A zatem

$$\mathbf{p}_i^T \mathbf{A} \mathbf{p}_j = 0, \quad i > j. \quad (23)$$

Metodę wybierania kierunków poszukiwań spełniających (23) nazywamy *metodą gradientów sprzężonych*.

## Jak się obejść bez hessjanu?

Z jednego z poprzednich wykładów znamy *algebraiczną* metodę gradientów sprzężonych, wydaje się zatem, iż moglibyśmy jej użyć do skonstruowania ciągu wektorów  $\{\mathbf{p}_k\}$ . Niestety, nie możemy, **nie znamy bowiem macierzy  $\mathbf{A}$** , czyli hessjanu w minimum. Czy możemy się bez tego obejść?

**Twierdzenie 1.** *Niech  $f$  ma postać (19) i niech  $\mathbf{r}_k = -\nabla f|_{\mathbf{x}_k}$ . Z punktu  $\mathbf{x}_k$  idziemy w kierunku  $\mathbf{p}_k$  do punktu, w którym  $f$  osiąga minimum kierunkowe. Oznaczmy ten punkt  $\mathbf{x}_{k+1}$ . Wówczas  $\mathbf{r}_{k+1} = -\nabla f|_{\mathbf{x}_{k+1}}$  jest **tym samym** wektorem, który zostałby skonstruowany w algebraicznej metodzie gradientów sprzężonych.*

*Dowód.* Na podstawie równania (20),  $\mathbf{r}_k = -\mathbf{A}\mathbf{x}_k + \mathbf{b}$  oraz

$$\mathbf{r}_{k+1} = -\mathbf{A}(\mathbf{x}_k + \alpha\mathbf{p}_k) + \mathbf{b} = \mathbf{r}_k - \alpha\mathbf{A}\mathbf{p}_k. \quad (24)$$

W minimum kierunkowym  $\mathbf{p}_k^T \nabla f|_{\mathbf{x}_{k+1}} = -\mathbf{p}_k^T \mathbf{r}_{k+1} = 0$  (por. (18)). Wobec tego mnożąc równanie (24) lewostronnie przez  $\mathbf{p}_k^T$ , otrzymujemy

$$\alpha = \frac{\mathbf{p}_k^T \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{A}\mathbf{p}_k}. \quad (25)$$

Ponieważ w algebraicznej metodzie gradientów sprzężonych  $\mathbf{r}_k^T \mathbf{p}_k = \mathbf{r}_k^T \mathbf{r}_k$ , otrzymujemy **dokładnie takie samo**  $\alpha$  jak we wzorach na metodę algebraiczną, co kończy dowód. □

## Algorytm gradientów sprzężonych

Rozpoczynamy w pewnym punkcie  $\mathbf{x}_1$ . Bierzemy  $\mathbf{r}_1 = \mathbf{p}_1 = -\nabla f|_{\mathbf{x}_1}$ .

1. Będąc w punkcie  $\mathbf{x}_k$ , dokonujemy minimalizacji kierunkowej w kierunku  $\mathbf{p}_k$ ; osiągamy punkt  $\mathbf{x}_{k+1}$ .
2. Obliczamy  $\mathbf{r}_{k+1} = -\nabla f|_{\mathbf{x}_{k+1}}$ .
3. Obliczamy (jak w metodzie algebraicznej)

$$\beta = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k}. \quad (26)$$

4. Obliczamy (jak w metodzie algebraicznej)  $\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta \mathbf{p}_k$ .



W metodzie gradientów sprzężonych te kroki, które *wymagałyby* znajomości hessjanu w minimum, *zastępujemy* minimalizacją kierunkową, natomiast te kroki, które *nie wymagają* znajomości hessjanu, wykonujemy tak samo, jak w algebraicznej metodzie gradientów sprzężonych.

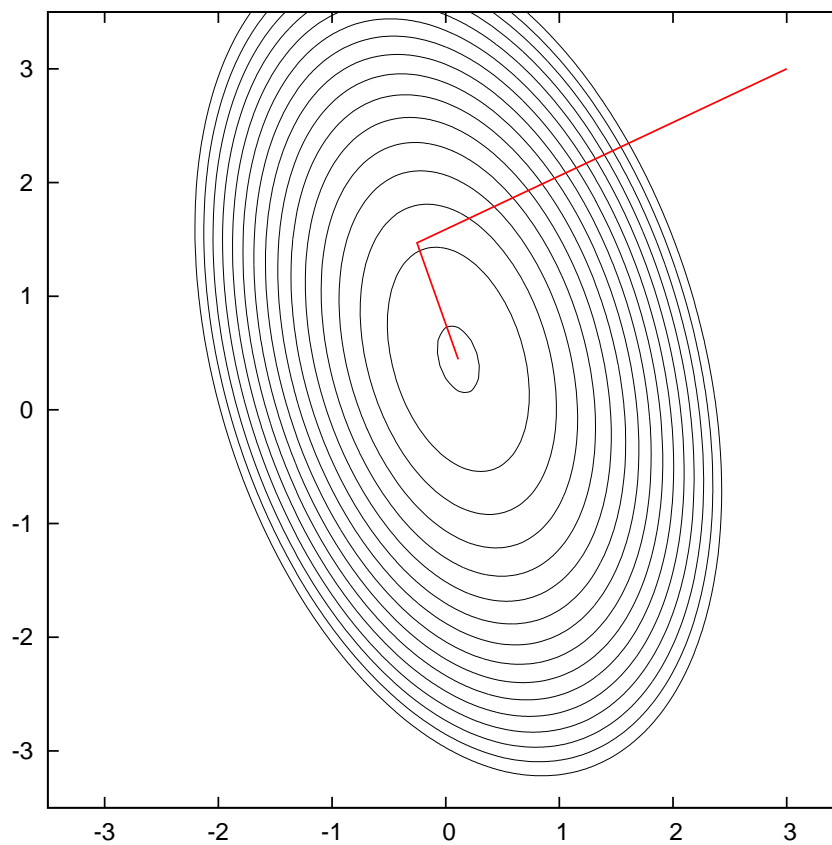
Twierdzenie ze strony 38 gwarantuje, że *formalnie* daje to to samo, co algebraiczna metoda gradientów sprzężonych.

Zamiast używać równania (26), można skorzystać z

$$\beta = \frac{\mathbf{r}_{k+1}^T (\mathbf{r}_{k+1} - \mathbf{r}_k)}{\mathbf{r}_k^T \mathbf{r}_k}. \quad (27)$$

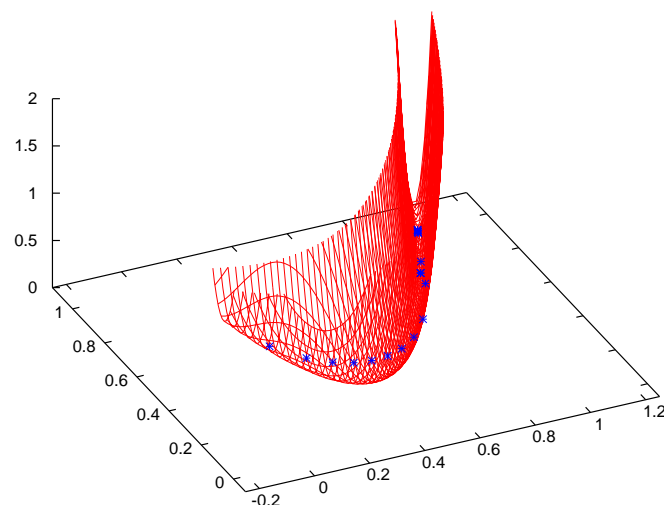
Jeżeli funkcja  $f$  ma *ściśle* postać (19), nie ma to znaczenia, gdyż  $\mathbf{r}_{k+1}^T \mathbf{r}_k = 0$ . Ponieważ jednak  $f$  jest tylko w przybliżeniu formą kwadratową, (27) może przyspieszyć obliczenia gdy grozi stagnacja.

## Przykład — metoda gradientów sprzężonych



Tylko dwa kroki! Drugi krok nie jest prostopadły do pierwszego, ale jest z nim sprzężony.

## Przykład — funkcja Rosenbrocka



Zastosowanie algorytmu gradientów sprzężonych do minimalizacji funkcji Rosenbrocka. Widać *znaczne* przyspieszenie (mniej punktów pośrednich!) w stosunku do przedstawianej powyżej metody najszybszego spadku.

## Metoda zmiennej metryki

Powróćmy do metody Newtona (9). Jeżeli **hessjan jest dodatnio określony**, krok Newtona prowadzi do spadku wartości funkcji — patrz (10b) — moglibyśmy więc użyć metody Newtona do precyzyjnego określania położenia minimum, gdy już znajdziemy się w jego otoczeniu. Dla funkcji **nie** będących w dobrym przybliżeniu formą kwadratową (19), może to, mimo dodatniej określoności hessjanu, oznaczać konieczność wykonania *wielu* drobnych kroków w końcowej fazie minimalizacji. **Praktyczną** trudność może stanowić konieczność wielokrotnego obliczania hessjanu, jeżeli obliczanie drugich pochodnych cząstkowych jest kosztowne lub kłopotliwe, w skrajnych przypadkach wręcz niemożliwe.

W takim wypadku warto jest rozważyć skorzystanie z **metody zmiennej metryki**: Zamiast korzystać z prawdziwego hessjanu obliczanego w każdym kroku, tworzymy **dodatnio określonych przybliżeń hessjanu**. Startujemy z jakiegoś  $x_0$ . Jako początkowe przybliżenie hessjanu  $H_0$  bierzemy jakąś sensowną macierz symetryczną, dodatnio określoną — jeśli nie mamy lepszego pomysłu, może to być macierz jednostkowa, natomiast jeśli do punktu  $x_0$  doszliśmy korzystając z metody Levenberga-Marquarda, naturalne (i zalecane) jest przyjęcie  $H_0 = \tilde{H}$ , gdzie ta ostatnia macierz to przybliżony hessjan obliczony w ostatnim kroku metody Levenberga-Marquarda. Następnie wykonujemy iterację:

1. Rozwiązujemy równanie  $\mathbf{H}_k \mathbf{p}_k = -\nabla f|_{\mathbf{x}_k}$ .
2. Przeprowadzamy minimalizację kierunkową funkcji jednowymiarowej  $f(\mathbf{x}_k + \alpha \mathbf{p}_k)$ . Niech minimum kierunkowemu odpowiada wartość  $\alpha_k$ .
3.  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$ .
4. Wyliczamy  $\mathbf{y}_k = \nabla f|_{\mathbf{x}_{k+1}} - \nabla f|_{\mathbf{x}_k}$ .
5. Wyliczamy nowe przybliżenie  $\mathbf{H}_{k+1}$ , *na przykład* za pomocą formuły BFGS:

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\alpha_k \mathbf{p}_k^T \mathbf{y}_k} - \frac{\nabla f|_{\mathbf{x}_k} (\nabla f|_{\mathbf{x}_k})^T}{\mathbf{p}_k^T \mathbf{H}_k \mathbf{p}_k}. \quad (28)$$

Kończymy gdy  $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \varepsilon$ , gdzie  $\varepsilon$  jest zadaną tolerancją.

## Własności metody zmiennej metryki

Można pokazać, że jeżeli  $\mathbf{H}_0$  jest symetryczna i dodatnio określona, także następne  $\mathbf{H}_k$  mają te własności.

Dodatkowo, jeśli  $f$  jest formą kwadratową (19),  $\mathbf{H}_{i \geq N+1} \equiv \mathbf{H}_{\min}$  w arytmetyce dokładnej. W ogólności ciąg  $\mathbf{H}_k$  *nie musi* być zbieżny do “prawdziwego” hessjanu, nawet jeśli metoda daje zadowalające numerycznie przybliżenie minimum.

W pewnym uproszczeniu można powiedzieć, że metoda zmiennej metryki ma się do metody gradientów sprzężonych tak, jak metoda Broydena ma się do wielowymiarowej metody Newtona (litera “B” w akronimie BFGS oznacza właśnie Broydena).



## Metoda Powella

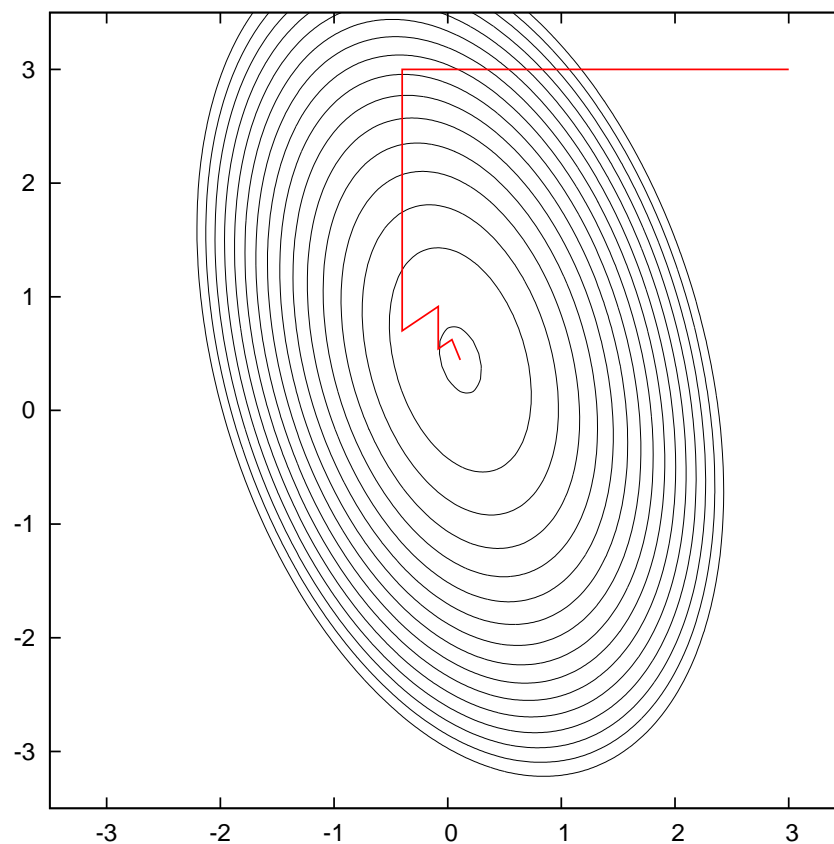
Wszystkie omówione wyżej metody wymagały obliczania pochodnych (gradientu) funkcji  $f(\mathbf{x})$ . Co jednak zrobić, jeśli obliczenie pochodnych jest kosztowne, niemożliwe lub funkcja jest nieróżniczkowalna? Zasadnicza strategia postępowania — minimalizacja kierunkowa, wybór nowego kierunku etc — pozostaje w mocy, zmienia się tylko sposób wyboru kolejnych kierunków. Metoda Powella polega na konstrukcji kierunków, które z czasem, po wielu iteracjach, stają się sprzężone.

Inicjalizujemy biorąc  $\{\mathbf{p}_i\}_{i=1}^N = \{\mathbf{e}_i\}_{i=1}^N$ , gdzie  $\{\mathbf{e}_i\}_{i=1}^N$  są kolejnymi wektorami (innymi słowy, zaczynamy od minimalizacji po współrzędnych). Następnie

1. Znajdujemy się w pewnym punkcie  $\mathbf{X}_0$ .
2. Minimalizujemy wzdłuż kolejnych kierunków  $\mathbf{p}_i$ , osiągając kolejno punkty  $\mathbf{X}_i$ .
3. Dla  $i = 1, \dots, N - 1$ :  $\mathbf{p}_i = \mathbf{p}_{i+1}$ .
4.  $\mathbf{p}_N = \mathbf{X}_N - \mathbf{X}_0$ .
5. Minimalizujemy wzdłuż (nowego)  $\mathbf{p}_N$ , oznaczając znaleziony punkt przez  $\mathbf{X}_0$ .
6. GOTO 1

Jeśli badana funkcja jest rozwijalna w szereg Taylora wokół minimum, po  $N$  iteracjach powyższej procedury kierunki  $\{\mathbf{p}_i\}_{i=1}^N$  stają się sprzężone.

## Przykład — metoda Powella



Mniej kroków niż w minimalizacji po współrzędnych. W większej liczbie wymiarów byłoby *jeszcze lepiej*.

## Minima wyższych rzędów

W powyższych rozważaniach zakładaliśmy, że w pobliżu minimum funkcja zachowuje się w przybliżeniu jak forma kwadratowa. Co jednak, gdy minimum jest wyższego rzędu, to znaczy gdy najniższy nieznikający rząd rozwinięcia Taylora jest rzędu czwartego (szóstego, ósmego, ...), jak w przykładzie funkcji  $f(x, y) = x^4 + y^4$ . Takie sytuacje **zdarzają** się w praktyce, na tyle jednak rzadko, że nie warto dla nich opracowywać osobnych metod. Korzystamy wówczas z metod już poznanych, właściwych dla minimów rzędu drugiego. Należy jednak mieć świadomość, że zbieżność w procedurze precyzyjnej lokalizacji minimum będzie wolniejsza, samo zaś minimum będziemy mogli wyznaczyć mniej dokładnie. Wynika to z faktu, że w okolicach minimum wyższego rzędu funkcja jest bardziej “płaska”.