Metody numeryczne II

Poszukiwanie ekstremów funkcji

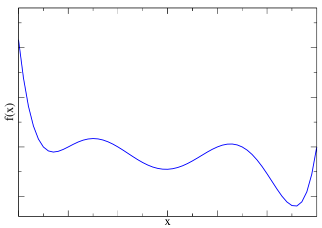
Janusz Szwabiński

szwabin@ift.uni.wroc.pl

Poszukiwanie ekstremów funkcji

1. Funkcje jednej zmiennej
   - metoda złotego podziału
   - interpolacja paraboliczna i metoda Brenta
   - poszukiwanie minimów z wykorzystaniem pierwszych pochodnych
2. Funkcje wielu zmiennych
   - metoda symplex
   - metoda Powella
   - metody oparte na gradientach
3. Minima globalne

Sformułowanie zagadnienia



Metoda złotego podziału

Jeżeli dla $a < b < c$ zachodzi

$$f(b) < f(a)$$

oraz

$$f(b) < f(c)$$

$\Rightarrow$ funkcja $f(x)$ ma w przedziale (a, c) minimum (o ile nie jest osobliwa)

$\Rightarrow$ potrzeba trzech punktów do zlokalizowania minimum

Wybieramy nowy punkt $x \in (b, c)$ (lub $x \in (a, b)$)

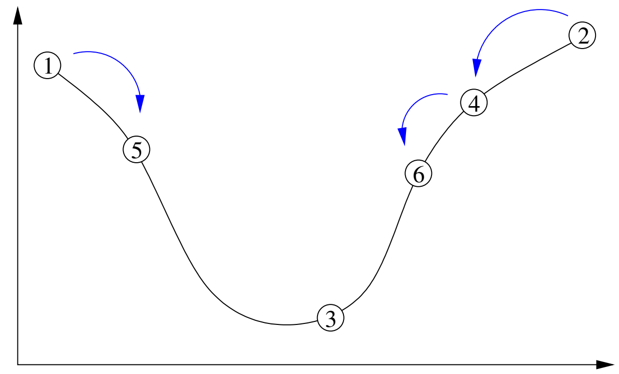
- $f(b) < f(x)$

  $\Rightarrow$ minimum leży w przedziale (a, x)

- $f(b) > f(x)$

  $\Rightarrow$ minimum leży w przedziale (b, c)

$\Rightarrow$ punkt środkowy trypletów (a, b, x) lub (b, x, c) jest najlepszym przybliżeniem minimum osiągniętym do tej pory



Załóżmy, że minimum leży w punkcie b oraz, że udało nam się go zlokalizować w małym przedziale

$$(1-\epsilon)b < b < (1+\epsilon)b$$

gdzie ϵ to liczba rzędu jednostki maszynowej (np. $3 * 10^{-8}$ w pojedynczej precyzji)
W otoczeniu punktu b funkcja dana jest wzorem

$$f(x) \simeq f(b) + \frac{1}{2} f''(b)(x-b)^2$$

Chcemy, aby w ostatnim równaniu drugi wyraz był rzędu $\epsilon f(b)$:

$$|x-b| < \sqrt{\epsilon}|b| \sqrt{\frac{2|f(b)|}{b^2 f''(b)}}$$

Dla większości funkcji wartość

$$\sqrt{\frac{2|f(b)|}{b^2 f''(b)}} \sim 1$$

Stąd

$$|x-b| < \sqrt{\epsilon} b$$

$\Rightarrow$ nie ma raczej sensu pytać o otoczenie minimum mniejsze niż $\sqrt{\epsilon} b$

Przypuśćmy, że b jest ułamkiem w odległości między a i c, tzn.

$$\frac{b-a}{c-a} = w, \quad \frac{c-b}{c-a} = 1-w$$

Załóżmy ponadto, że

$$\frac{x-b}{c-a} = z$$

⇒ następny przedział, w którym znajduje się minimum, będzie miał długość $w + z$ albo $1 - w$

Chcemy zminimalizować najgorszą możliwość:

$$z = 1 - 2w$$

Stąd wynika

$$|b-a| = |x-c|$$

⇒ x będzie symetryczny do b względem środka przedziału i będzie leżał w dłuższym z podprzedziałów

Jeżeli wybraliśmy z optymalnym w bierzącym kroku, to w poprzednim zrobiliśmy to samo dla w

⇒ x powinien być tym samym ułamkiem odległości między b i c (jeśli to jest większy podprzedział), co b jest między a i c

Stąd

$$\frac{z}{1-w} = w \;\Rightarrow\; w^2 - 3w + 1 = 0 \;\Rightarrow\; w = \frac{3-\sqrt{5}}{2} \approx 0,38197$$

⇒ punkt środkowy b optymalnego przedziału (a, b, c) zawierającego minimum leży w odległości ułamkowej $0,38197$ od jednego z końców i $0,61803$ od drugiego

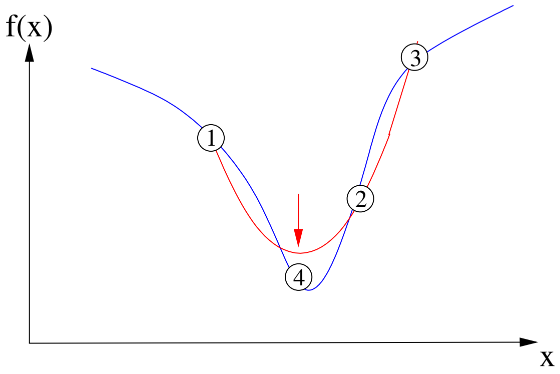
⇒ liczba cyfr znaczących w wyniku rośnie liniowo z liczbą dodatkowych kroków

Interpolacja paraboliczna i metoda Brenta

Funkcja wystarczająco gładka

⇒ w pobliżu minimum funkcja kwadratowa jest dobrym przybliżeniem

⇒ parabola dopasowana do dowolnych trzech punktów powinna doprowadzić nas w jednym kroku do minimum albo przynajmniej w jego bliskie otoczenie



Minimum paraboli przechodzącej przez trzy dowolne (niewspółliniowe) punkty a, b i c

$$x = b - \frac{1}{2}\frac{(b-a)^2[f(b)-f(c)] - (b-c)^2[f(b)-f(a)]}{(b-a)[f(b)-f(c)] - (b-c)[f(b)-f(a)]}$$

Wada: wzór opisuje zarówno minimum jak i maksimum paraboli

$\Rightarrow$ w pobliże minimum dochodzimy innym algorytmem

Metoda Brenta:

- w każdym kroku tej metody zapamiętujemy sześć punktów a, b, u, v, w i x
  - minimum leży w przedziale (a, b)
  - x to najlepsze do tej pory przybliżenie minimum
  - w punkcie w funkcja ma drugą co do wartości najmniejszą wartość
  - v - poprzednia wartość w
  - u to punkt, w którym wartość funkcji była wyliczona ostatnio

- interpolację kwadratową przeprowadzamy dla punktów x, v i w
- wynik interpolacji zostaje zaakceptowany, jeśli:
  - parabola leży wewnątrz przedziału (a, b)
  - interpolacja pociąga za sobą przesunięcie od x do nowego punktu, które jest mniejsze niż połowa takiego przesunięcia w poprzednim kroku.
- jeśli interpolacja będzie nie do przyjęcia, „włączamy” metodę złotego podziału

Metody wykorzystujące pierwsze pochodne

Teoretycznie możemy po prostu szukać pierwiastków pierwszej pochodnej, ignorując zupełnie informację zawartą w wartościach funkcji

Wady

- jak odróżnić maksima od minimów (bez wyliczania drugiej pochodnej)
- dokąd „pójść” z warunków początkowych, które pokazują spadek pochodnych w jednym (lub obu) punkcie zewnętrznym w kierunku wyprowadzającym po za przedział określoności funkcji

Inna możliwość

- jeżeli minimum znajduje się w przedziale (a, b, c), obliczenie pochodnej w punkcie b wskazuje jednoznacznie, czy punkt x leży w (a, b) czy w (b, c)
- wartości pochodnej w b i w poprzednim punkcie x są ekstrapolowane do zera metodą siecznych (odwrotna interpolacja liniowa)

Metoda symplex

Definicja Sympleks to figura geometryczna, składająca się w N wymiarach z $N + 1$ punktów (wierzchołków)

- trójkąt w dwóch wymiarach
- czworościan (niekoniecznie regularny) w trzech wymiarach
- traktując dowolny punkt niezdegenerowanego sympleksu jako początek, inne punkty definiują kierunki wektorów rozpinających N–wymiarową przestrzeń wektorową

Wartości początkowe

- potrzebujemy $N + 1$ punktów
- oznaczając jeden z nich przez $\vec{P}_0$, inne określamy ze wzoru

$$\vec{P}_i = \vec{P}_0 + \lambda \vec{e}_i$$

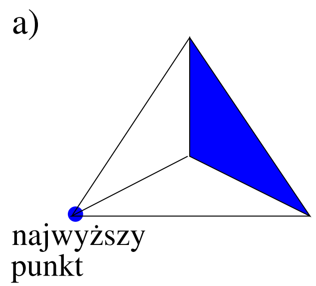
gdzie $\vec{e}_i$ - wektor bazowy

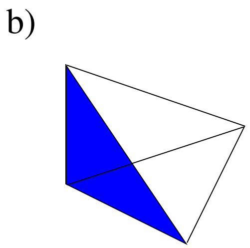
λ - charakterystyczna skala problemu

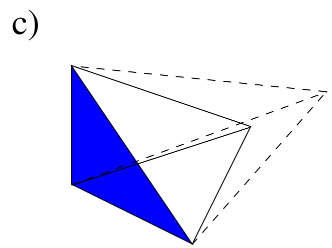
Definicja Najwyższym punktem sympleksu nazywamy wierzchołek, w którym wartość funkcji będzie największa. Analogicznie określamy punkt najniższy sympleksu

Deformacje sympleksu

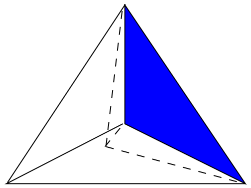
- odbicia (z zachowaniem objętości)
- odbicie i rozciąganie
- kontrakcja







d)



$\Rightarrow$ sympleks porusza się w kierunku najbliższego minimum i zapada się w sobie po jego osiągnięciu

Metoda Powella

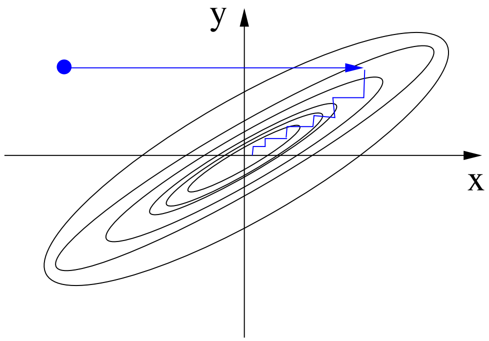
Startujemy w punkcie $\vec{P}_0$ i przemieszczamy się w kierunku $\vec{n}$

$\Rightarrow$ minimum funkcji $f(\vec{P})$ wzdłuż $\vec{n}$ możemy znaleźć za pomocą metod dla funkcji jednej zmiennej

$\Rightarrow$ chcemy znaleźć minimum funkcji wielu zmiennych wykonując szereg jednowymiarowych minimalizacji

Wektory jednostkowe $\vec{e}_1, \ldots, \vec{e}_N$ jako zbiór kierunków

- zadowalające dla pewnej klasy funkcji, ale...
- ...mało wydajne



Przy wyborze zbioru kierunków powinniśmy zwrócić uwagę na dwie rzeczy:

- zbór powinien zawierać kilka, które szybko doprowadzą nas w pobliże minimum
- minimalizacja w jednym kierunku nie powinna psuć osiągniętej już minimalizacji w innym.

Minimalizujemy funkcję wzdłuż $\vec{u}$

$\Rightarrow$ gradient funkcji w minimum (liniowym) musi być prostopadły do $\vec{u}$

Rozwijamy $f(\vec{x})$ w szereg względem początku układu $\vec{P}$

$$\begin{aligned} f(\vec{x}) &= f(\vec{P}) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \dots \\ &\simeq c - \vec{b}\vec{x} + \frac{1}{2}\vec{x}\mathbf{A}\vec{x} \end{aligned}$$

$$c \equiv f(\vec{P}), \quad \vec{b} \equiv -\nabla f|_{\vec{P}}, \quad [\mathbf{A}]_{i,j} \equiv \frac{\partial^2 f}{\partial x_i \partial x_j}$$

Gradient funkcji ma postać

$$\nabla f = \mathbf{A}\vec{x} - \vec{b}$$

$\Rightarrow$ minimum występuje w punkcie będącym rozwiązaniem układu

$$\mathbf{A}\vec{x} = \vec{b}$$

Poruszając się w pewnym kierunku, spowodujemy zmianę gradientu

$$\delta(\nabla f) = \mathbf{A}(\delta\vec{x})$$

Aby minimalizacja wzdłuż $\vec{v}$ nie zepsuła minimalizacji wzdłuż $\vec{u}$, gradient musi pozostać prostopadły do $\vec{u}$

$$0 = \vec{u}\delta(\nabla f) = \vec{u}\mathbf{A}\vec{v}$$

$\Rightarrow$ $\vec{u}$ i $\vec{v}$ są sprzężone

Inicjalizujemy zbiór kierunków $\vec{u}_i$ wektorami bazowymi

$$\vec{u}_i = \vec{e}_i, \quad i = 1, \dots, N$$

Procedura bazowa:

- zapamiętaj pozycję wyjściową jako $\vec{P}_0$
- dla $i = 1, \dots, N$, przesuń $\vec{P}_{i-1}$ do minimum wzdłuż kierunku $\vec{u}_i$ i oznacz ten punkt jako $\vec{P}_i$
- dla $i = 1, \dots, N-1$ przyjmij $\vec{u}_i \leftarrow \vec{u}_{i+1}$
- przyjmij $\vec{u}_N \leftarrow \vec{P}_N - \vec{P}_0$
- przesuń $\vec{P}_N$ do minimum wzdłuż $\vec{u}_N$ i nazwij ten punkt $\vec{P}_0$

⇒ dla formy kwadratowej k iteracji produkuje zbiór kierunków, którego k ostatnich elementów jest wzajemnie ze sobą sprzężonych

⇒ N iteracji procedury bazowej ($N(N+1)$ minimalizacji liniowych) powinno dokładnie zminimalizować formę kwadratową

Wada

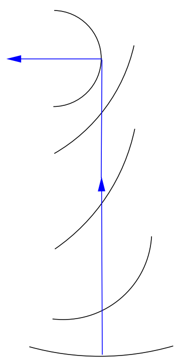
- odrzucenie w każdym kroku $\vec{u}_1$ na rzecz $\vec{P}_N - \vec{P}_0$ zmierza do wytworzenia kierunków liniowo zależnych

⇒ reinicjalizuj zbiór kierunków $\vec{u}_i$ po każdych N lub $N+1$ iteracjach procedury bazowej

Metoda najszybszego spadku

- wybierz punkt początkowy $\vec{P}_0$
- przejdź od $\vec{P}_i$ do $\vec{P}_{i+1}$ minimalizując funkcję wzdłuż kierunku najszybszego spadku, tzn.

$$-\nabla f(\vec{P}_i)$$



⇒ metoda wymusza skręty pod kątem prostym

⇒ mała wydajność (ale duża stabilność!)

Metoda gradientów sprzężonych

Załóżmy ponownie, że naszą funkcję można przybliżyć formą

$$f(\vec{x}) \simeq c - \vec{b}\vec{x} + \frac{1}{2}\vec{x}\mathbf{A}\vec{x}$$

Startując od pewnego wektora $\vec{g}_0$, generujemy dwa ciągi wektorów,

$$\begin{aligned} \vec{g}_{i+1} &= \vec{g}_i - \lambda_i \mathbf{A}\vec{h}_i, \quad i = 0, 1, 2, \ldots \\ \vec{h}_{i+1} &= \vec{g}_{i+1} + \gamma_i \vec{h}_i, \quad \vec{h}_0 = \vec{g}_0 \end{aligned}$$

Chcemy, aby wektory spełniały warunki ortogonalności i sprzężenia

$$\begin{aligned}\vec{g}_i\vec{g}_j &= 0\\ \vec{h}_i\mathbf{A}\vec{h}_j &= 0,\;\; j<i\\ \vec{g}_i\vec{h}_j &= 0\end{aligned}$$

Stąd wynika

$$\begin{aligned}\lambda_i &= \frac{\vec{g}_i\vec{g}_i}{\vec{h}_i\mathbf{A}\vec{h}_i} = \frac{\vec{g}_i\vec{h}_i}{\vec{h}_i\mathbf{A}\vec{h}_i}\\ \gamma_i &= \frac{\vec{g}_{i+1}\vec{g}_{i+1}}{\vec{g}_i\vec{g}_i}\end{aligned}$$

Twierdzenie 1 *Niech*

$$\vec{g}_i = -\nabla f(\vec{P}_i)$$

dla pewnego punktu $\vec{P}_i$, w którym funkcja f ma postać formy kwadratowej. Przypuśćmy dalej, że kontynuujemy poszukiwanie minimum w kierunku $\vec{h}_i$ oraz, że nowe minimum ulokowane jest w punkcie $\vec{P}_{i+1}$. Wówczas zachodzi

$$\vec{g}_{i+1} = -\nabla f(\vec{P}_{i+1})$$

przy czym $\vec{g}_{i+1}$ jest taki sam, jakby był skonstruowany za pomocą

$$\vec{g}_{i+1} = \vec{g}_i - \lambda_i\mathbf{A}\vec{h}_i,\;\; i=0,1,2,\ldots$$

Dowód Z równania

$$\nabla f = \mathbf{A}\vec{x} - \vec{b}$$

wynika

$$\vec{g}_i = -\mathbf{A}\vec{P}_i + \vec{b}$$

a zatem

$$\vec{g}_{i+1} = -\mathbf{A}(\vec{P}_i + \lambda\vec{h}_i) + \vec{b} = \vec{g}_i - \lambda\mathbf{A}\vec{h}_i$$

z λ dobranym tak, aby osiągnąć minimum liniowe. Ale w minimum tym zachodzi

$$\vec{h}_i\nabla f = -\vec{h}_i\vec{g}_{i+1} = 0$$

Dowód (ciąg dalszy)
Rozwiązując ostanie dwa równania względem λ, otrzymamy

$$\lambda_i = \frac{\vec{g}_i\vec{g}_i}{\vec{h}_i\mathbf{A}\vec{h}_i} = \frac{\vec{g}_i\vec{h}_i}{\vec{h}_i\mathbf{A}\vec{h}_i}$$

co kończy dowód ∎

Wyżarzanie symulowane

Analogia z termodynamiką

- wolno schładzany system znajduje stan o minimalnej energii
- energia systemu w równowadze termicznej w temperaturze T jest rozłożona pomiędzy różne stany energetyczne z rozkładem prawdopodobieństwa Boltzmanna

$$P(E) \sim \exp\left(-\frac{E}{kT}\right)$$

$\Rightarrow$ nawet w niskich temperaturach system może znajdować się w stanie o wysokiej energii

$\Rightarrow$ istnieje możliwość, że system „wyjdzie" z lokalnego minimum i znajdzie inne (lepsze)

Algorytm Metropolisa: symulowany system zmienia konfigurację z energii E_1 do E_2 z prawdopodobieństwem

$$p = \min\left[1, \exp\left(-\frac{E_2 - E_1}{kT}\right)\right]$$

Niezbędne elementy (w systemach innych niż termodynamiczne)

- opis możliwych konfiguracji systemu
- generator losowych zmian w konfiguracji
- funkcja E (odpowiednik energii), której minimum to cel obliczeń
- parametr T (odpowiednik temperatury) oraz reguły, w jaki sposób parametr ten zmieniany jest od dużych do małych wartości.

Zagadnienie komiwojażera:

- odwiedzamy N miast o pozycjach (x_i, y_i), a następnie wracamy do miasta początkowego
- każde miasto ma być odwiedzone tylko raz
- trasa możliwie najkrótsza

Problem NP–zupełny

$\Rightarrow$ czas znalezienia dokładnego rozwiązania rośnie z N jak

$$\exp[const \times N]$$

$\Rightarrow$ koszty obliczeniowe są bardzo duże

$\Rightarrow$ dużo minimów lokalnych

Cel praktyczny: znaleźć minimum, które nawet jeśli nie jest absolutne, nie może być już dużo bardziej poprawione

⇒ wyżarzanie pozwala znaleźć takie minimum z nakładem obliczeń N^{γ}, gdzie γ to pewna mała liczba

1. *Konfiguracja*: miasta są ponumerowane indeksem $i = 1, \ldots, N$ i każde z nich ma współrzędne (x_i, y_i). Konfiguracje to permutacje numerów miast
2. *Zmiany konfiguracji*:
   - część trasy usunięta i zastąpiona tymi samymi miastami w odwrotnej kolejności
   - część trasy usunięta i wstawiona między inne dwa miasta

3. *„Energia”* to po prostu długość całej trasy

$$E = L = \sum_{i=1}^{N} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$

4. Generujemy najpierw losowo różne konfiguracje i używamy ich do określenia zmian ΔE, z którymi będziemy mieli do czynienia przy przejściu od kroku do kroku. Wybieramy początkowe T dużo większe od największego ΔE, a następnie zmniejszamy T o maksymalnie 10% w każdym kroku. Utrzymujemy T stałe dla $100N$ rekonfiguracji lub $10N$ udanych rekonfiguracji (w zależności od tego, co nastąpi wcześniej)

Minimum „zwykłej” funkcji $f(\vec{x})$

1. Konfiguracja to po prostu punkt $\vec{x}$
2. Zmiana konfiguracji to losowe przesunięcie od $\vec{x}$ do $\vec{x} + \delta\vec{x}$
3. Energia to wartość $f(\vec{x})$
4. Parametr kontrolny definiuje się podobnie, jak w przypadku minimalizacji kombinatoryjnej

Metody dyfuzyjne

Wyobraźmy sobie metodę, której każdy krok polega na częściowym „zasypywaniu” minimów lokalnych

⇒ po pewnej liczbie kroków, powinna zostać jedynie reszta minimum globalnego

⇒ do jego znalezienia wystarczą metody lokalne

Funkcję jednej zmiennej $f(x)$ poddajemy następującemu ciągowi transformacji:

$$
\begin{aligned}
f_0(x) & \\
f_1(x) &= f_0(x) + \Delta t \frac{\mathrm{d}^2}{\mathrm{d}x^2} f_0(x) \\
&\vdots \\
f_{n+1}(x) &= f_n(x) + \Delta t \frac{\mathrm{d}^2}{\mathrm{d}x^2} f_n(x)
\end{aligned}
$$

przy czym Δt jest małą, dodatnią liczbą

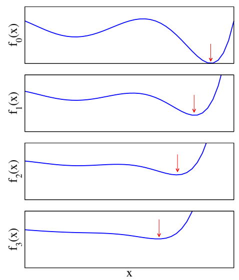
⇒ minima w każdym kroku będą „płytsze”

⇒ maksima będą niższe

⇒ w punktach przegięcia funkcja pozostaje bez zmian

Przykład Rozważmy funkcję

$$f(x) = (x-3)(x-1.5)(x-1)(x+1)(x+2)(x+2,5)$$



Wady

- pozycja minimum w każdym kroku przesuwa się w porównaniu z położeniem oryginalnym
- konieczność wyliczania drugich pochodnych

Rozwiązanie (pierwszego problemu)

- obliczamy minimum funkcji $f_4(x)$ i traktując to jako wartość startową, szukamy minimum funkcji $f_3(x)$ itd.
- możliwe, jeżeli krok od f_i do f_{i+1} nie będzie zbyt duży

Z równania

$$f_{n+1}(x) = f_n(x) + \Delta t \frac{\mathrm{d}^2}{\mathrm{d}x^2} f_n(x)$$

wynika

$$\frac{f_{n+1}(x) - f_n(x)}{\Delta t} = \frac{\mathrm{d}^2 f_n(x)}{\mathrm{d}x^2}$$

W granicy $\Delta t \to 0$ otrzymamy

$$\frac{\mathrm{d} f_n(t)}{\mathrm{d}t} = \frac{\mathrm{d}^2 f_n(x)}{\mathrm{d}x^2}$$

czyli jednorodne równanie dyfuzji ze stałą dyfuzji równą 1 oraz warunkiem początkowym

$$f(x, t = 0) = f_0(x)$$

W d wymiarach

$$\frac{\mathrm{d} f_n(\vec{x}, t)}{\mathrm{d}t} = \Delta_d f(\vec{x}, t)$$

Możliwe rozwiązanie tego równania ma postać

$$f(\vec{x}, t) = \frac{1}{(2\sqrt{\pi t})^d} \int \mathrm{d}^d x' f_0(\vec{x}') \exp\left[-\frac{|\vec{x} - \vec{x}'|^2}{4t}\right]$$