Тоон аргууд

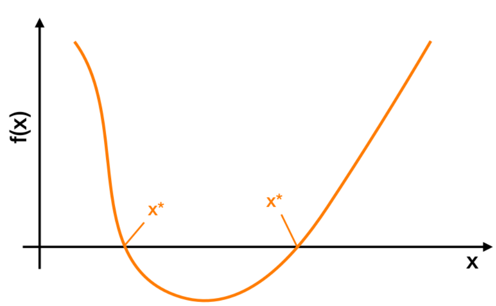
25 февраля 2020 г.

1 Шугаман Бус Систем Тэгшитгэл

Бид бүхэн N хувьсагчтай N шугаман тэгшитгэлийн систем $A\vec{x} = \vec{b}$ -ийн шийдийг хэрхэн олох талаар үзсэн. Энд A нь $N \times N$, харин $\vec{x}, \vec{b}$ нь N хэмжээст. Түүнээс гадна дата өгөгдөлийн хувьд хэрхэн шугаман тэгшитгэлийн шийд $\vec{x}$-ийг олохыг бид үзсэн. Гэвч N шугаман бус тэгшитгэлийн $f_i(\vec{x})$ шийдийг олох нь хялбар биш юм. Шугаман бус систем тэгшитгэлийг $f_i(\vec{x}^*) = 0$ хангах $\vec{x}^*$-ийг $(i = 1, \dots, N; \vec{x} = (x_1, \dots, x_N))$ олох нь зорилго юм. Шугамна бус тэгшитгэлийг доорх байдлаар илэрхийлж болно,

$$
\begin{aligned}
f_1(\vec{x}) &= f_1(x_1, x_2, \dots x_N) = 0 \\
f_2(\vec{x}) &= f_2(x_1, x_2, \dots x_N) = 0 \\
&\vdots \\
f_N(\vec{x}) &= f_N(x_1, x_2, \dots x_N) = 0.
\end{aligned} \tag{1}
$$

Ерөнхий тохиолдолд $\vec{F}(\vec{x}) = 0$ гэж бичдэг. Шугаман бус тэгшитгэлийн системийн шийдийг тооцох нь язгуур олох болдлогын нэгэн төрөл юм. Өөр үгээр илэрхийлбэл $f(x)$ функцын графикийг байгуулан, фукцын x тэнхлэгийг дайрдаг x^* утгыг олох юм.



1.1 Мэдрэмж болон нөхцөл

Хэрэв тэгшитгэлийн шийд оршин байдаг бол, шийдийг нь ямар нэгэн алгоритм хэрэглэн олох боломжтой юу?. Тэгвэл тэгшитгэл $f(x) = 0$-ийг хангадаг ойролцоо шийд нь $\tilde{x}$ харин бодит шийд нь x^* гэж үзээ. Энэ тохиолдолд $\| f(\tilde{x}) \| \approx 0$ буюу $\| \tilde{x} - x^* \| \approx 0$. Бодит шийдийг яг таг олох боломжгүй байдаг бөгөөд харин ойролцоогоор $f(\tilde{x}) \approx 0$-ийн шийд нь $\tilde{x}$ гэж авч үзнэ.

Функц нь Well-Conditioned үед оролтын өгөгдөлийг бага хэмжээгээр өөрчилөхөд үр дүн гаралт мөн бага өөрчлөгддөг. Эсрэг тохиолдол болох ill-conditioned үед оролтын өгөгдөлийг бага өөрчилөхөд гаралтын өгөгдөлийн өөрчилөлт их байдаг.

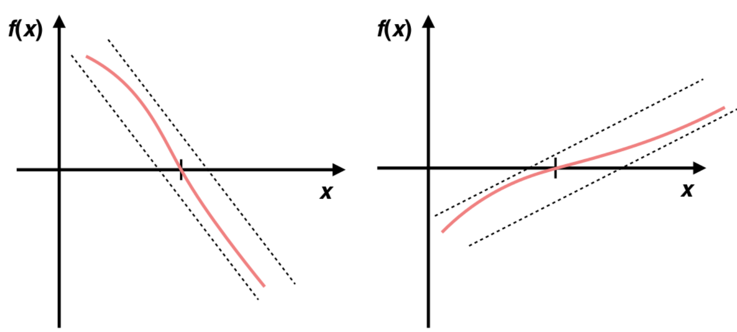
Функцыг ill- эсвэл well-conditioned хэмжихийн тулд **нөхцөлийн тоо, k**-г тодорхойлдог $k = |\delta y|/|\delta x|$, энд оролтын утгын өөрчилөлт δx, гаралтын үр дүнгийн өөрчилөлт $\delta y = f(x + \delta x) - f(x)$. Хэрэв $f(x + \delta x)$-ийг Тейлорын цуваанд задалбал; $\delta y = (f(x) + f^{'}(x)\delta x) - f(x) = f^{'}(x)\delta x$. **нөхцөлийн тоо, k**;

$$k = |f^{'}(x)| \tag{2}$$

болно. Энэ нь зөвхөн x өгөгдсөн үеп биелдэг бөгөөд бидний тохиолдолд үл мэдэгдэх хувьсагч ба x^* ойролцоо шийд нь мэдэгдэж байгаа. Тиймээс нөхцөлийн тоог доорх байдлаар тодорхойлно,

$$k = \frac{1}{|f^{'}(x^*)|}. \tag{3}$$

Хэрэв **нөхцөлийн тоо k** бага байвал тухайн функцын шийдийг well-conditioned гэж хэлдэг. Энд $|f^{'}(x^*)|$ бага үед функцын шийд нь Well-Conditioned, эсрэг тохиолдолд ill-Conditioned (тангент нь хэвтээ) гэнэ. Графикаар эхнийх нь Well-Conditioned, сүүлийнх ill-Conditioned тохиолдолууд.



1.2 Нийлэлтийн Хурд

Шугаман бус тэгшитгэлийн шийдийг олохын тулд давталтын аргыг ашигладаг. Анхны таамагласан утга $x^{(0)}$ нь боломжит шийдүүдийн дарааллыг үүсгэдэг, $x^{(1)}, x^{(2)}, x^{(3)}, \ldots x^{(k)}, \ldots$; буюу $\{x^{(k)}\}_{k=0}^{\infty}$ гэж тэмдэглэдэг. Нийлэлтийн хурдыг тодорхойлохын тулд алдааны утгыг ашигладаг, $\{x^{(k)}\}^{k\to\infty} \to x^*$ үед

$$\lim_{k\to\infty} \frac{|E^{(k+1)}|}{|E^{(k)}|^r} = C, \quad E^{(k)} = x^{(k)} - x^*. \tag{4}$$

r нь нийлэлтийн хурдны эрэмбэ. $r = 1$ шугаман нийлэлтийн хурд, $r = 2$ квадрат нийлэлтийн хурд.

2 Bisection method

$N = 1$ үед $f(x)$-ийн шийд x^* нь $f(x^*) = 0$ хангана гэж үзэе. x^* нь $f(x)$-ийн язгуур юм. $f(x^*) = 0$ хангадаг x^*-ийг олох нэгэн энгийн алгоритм нь **BISECTION** арга юм. **BISECTION** арга нь доорх байдлаар тодорхойлогддог: $[a, b]$ завсарт $f(x)$ функцын уламжлал оршин байдаг, тасралтгүй буюу $f(a)f(b) < 0$ хангадаг бол, $[a, b]$ завсарт $f(x^*) = 0$ хангах шийд x^* оршино гэж үзвэл,

1. $[a, b]$ завсарын дундаж цэгийг сонгоно: $x_1 = a + (b - a)/2 = (a + b)/2$.
    - Хэрэв $f(x^{(1)}) = 0$ бол, $x^* = x^{(1)}$ төгсөв.
    - Хэрэв $f(x^{(1)}) \neq 0$ бол, $f(x^{(1)})$ нь $f(a)$-тэй ижил тэмдэгтэй эсвэл $f(b)$-тэй.
        - Хэрэв $f(x^{(1)})$ нь $f(a)$-тэй ижил тэмдэгтэй бол, шинэ завсар $[x^{(1)}, b]$ гэж тодорхойлно.
        - Хэрэв $f(x^{(1)})$ нь $f(b)$-тэй ижил тэмдэгтэй бол, шинэ завсар $[a, x^{(1)}]$ гэж тодорхойлно.
    - Шинэ завсарт дахин дээрх тооцоог давтна.

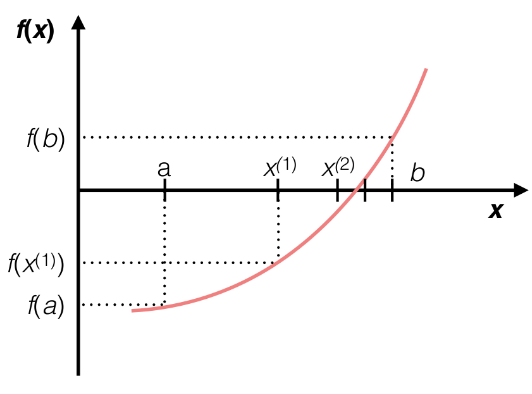


Рис. 1: BISECTION арга.

Algorithm 1 BISECTION АРГА

ОРОЛТ: (a, b) завсар, алдааны доод заат tol, хамгийн их давталтын тоо N.

ГАРАЛТ: ойролцоо шийд $x^{(k)}$ эсвэл шийд олдоогүй nan.

$k = 1$

$FA = f(a)$

while $k \leq N$ **do**

  $x^{(k)} = (a + b)/2;$

  $Fx = f(x^{(k)}).$

  if $Fx = 0$ эсвэл $(b - a)/2 < tol$ **then**

    Гаралт: $x^{(k)}$ ▷ Тэгшитгэлийн шийдийг гаралтанд өгөх.

    Алгоритм Зогсох.

  $k = k + 1$

  if $sign(FA) = sign(Fx)$ **then**

    $a = x^{(k)}$

    FA=Fx.

  else

    $b = x^{(k)}$ ▷ Шинэ завсар тодорхойлно.

ГАРАЛТ: нийт N давталтын турш шийд олдоогүй nan.

Алгоритм Зогсох.

```python
import numpy as np
import math as m
```

```python
def fnc(x):
    return 2*x**3-x**2+x-1
```

```python
N_max=14
tol=10**(-4)
a=0
b=1
k=1
FA=fnc(a)
```

```python
while k<=N_max:
    xk=(a+b)/2
    Fx=fnc(xk)
    if Fx==0 or (b-a)/2<tol:
        print(xk)
        break
    k=k+1
    if FA*Fx>0:
        a=xk
        FA=Fx
    else:
        b=xk
#print('nan')
```

```
0.73895263671875
```

Рис. 2: BISECTION аргад зориулсан Python алгоритм.