



# שיטת חישובית באופטימיזציה

## 046197

סמסטר אביב תשפ"א 2021

מגיש:

204397368

עדו יצחקאל

לעגין

הו  $f: \mathbb{R} \rightarrow \mathbb{R}$  ג'רואה ועפלה כ-קיטpig.

כפי שראנו לאלה

$$\min_x \{f(x) | x \geq 0\}$$

בז' נאנו ידיה של הוליך ה-  
יונכיר זיה.

בז' נאנו ידיה של הוליך ה-  
 $x=0$  ב-  $f'(0) \leq 0$ .

הוכחה

הוכיחו שקיים  $x^*$  כ-הנקודה קיימת אוניביז'ן  
Slater (קיום קיון). כיוון שקיים  
 $(x^*, f(x^*))$  הוכח (בנוסף לכך) KKT

: מילויים:

$$L(x, \mu) = f(x) - \mu x$$

$$\begin{aligned} \text{stationarity} \quad \frac{\partial L}{\partial x}(x, \mu) &= f'(x) - \mu = 0 \\ &\Rightarrow f'(x) = \mu. \end{aligned}$$

C-S:

KKT מילויים יוגה ה- $(0, f(0))$  ה- $\mu = f'(0)$

$$\begin{aligned} C-S: \mu \cdot x|_{x=0} &= \mu \cdot 0 = 0 & \mu = f'(0) \geq 0 & x=0 \geq 0 \\ && \text{dual feasibility} & \text{primal} \end{aligned}$$

אם הוגדר  $x^* = 0$ ;  $\mu^* = f(0)$  אז לא תימצא נקודה קיימת על הגרף פרמי.

בפועל הדוגמא היא מילוי של נקודה קיימת בפונקציית  $f(x) = \frac{1}{2}x^2 - 1$ .

$$y > 0$$

$$f(y) = f(0) + y \cdot f'(0) \geq f(0)$$

אם  $y > 0$  אז  $x=0$  לא קיימת נקודה סידור.

$f'(x)=0 \Rightarrow x=0 \geq 0 \Rightarrow f'(0) \leq 0$  ו- 2

ו-1 פרמי

$(x, \mu=0)$  נובעת מוקהה

C-S:  $0 \cdot x_{\geq 0} = 0$        $x \geq 0$        $\mu=0 \geq 0$       stationary:

primal      dual

$$\frac{\partial L}{\partial x} = f'(x) - \mu = 0$$

אם  $\mu=0$  אז  $f'(x) = 0$  מוקהה כטיגת  $f(x) = 0$

ו-KKT

בפועל הדוגמא היא מילוי של נקודה סידור

אם זו אונטיה מוקהה אז  $f(x) = 0$  מוקהה

ו-1 מוקהן נזיר. משום שנקה מילוי

אנו מילוי מוקהן ה-2 מילוי.

2.5.2:

וניה כפער הילו נקעה הטענה:

$$(P) \quad \min_x \quad x^T Q x$$

$$\text{s.t.} \quad x^T Q x + b^T x \geq 1$$

$$C^T x \geq 0$$

$$b, c \in \mathbb{R}^n \quad \text{পি} \quad R^{xx} \in Q > 0 \quad \text{ככל}$$

ל. הינה כ פ נושא.

הוכחה:  
 $Q > 0 - l$  ו  $f(x)$  מינימלית נכון ל  $x^T Q x + b^T x \geq 1$   
 אמצע הטענה כהה:  $f(x) \leq f(x_0)$   $\forall x \in \mathbb{R}^n$

$$x^T Q x + b^T x \geq 1 \Rightarrow -x^T Q x - b^T x + 1 \leq 0$$

הוכחה כבש נושא:

$$Q = I ; b = 0 \Rightarrow -x^2 + 1 \leq 0 \\ (1-x)(1+x) \leq 0$$

כמ"ד  $x \in [-1, 1]$

$$\frac{l}{2}(-1) + (l - \frac{l}{2})(1) : \text{קמיה גיאומטרית ג'ע} \quad -\frac{l}{2} + \frac{l}{2} = 0 \quad \& \quad x \leq -1 \cup x \geq 1$$

הנעה טה להלן מושג:

גיאומטריה

## נימוק הוכחה נורמלית ②

$$\min x^T Q x$$

$$\text{s.t. } -x^T Q x - b^T x + l \leq 0$$

$$-C^T x \leq 0$$

נימוק הוכחה נורמלית (בנוסף ל- $\lambda_1 = 0$ )

$$L(x, \lambda_1, \lambda_2) = x^T Q x - \lambda_1 x^T Q x - \lambda_1 b^T x + \lambda_1 - \lambda_2 C^T x$$

$$= (l - \lambda_1) x^T Q x - (\lambda_1 b + \lambda_2 c)^T x + \lambda_1$$

נימוק הוכחה נורמלית (בנוסף ל- $\lambda_1 = 0$ )

$$\frac{\partial L(x, \lambda_1, \lambda_2)}{\partial x} = 2(l - \lambda_1) Q x - (\lambda_1 b + \lambda_2 c) = 0$$

$$x = Q^{-1} \frac{l}{2(l - \lambda_1)} (\lambda_1 b + \lambda_2 c)$$

$$g(\lambda_1, \lambda_2) = \cancel{(l - \lambda_1)} \frac{l}{2(l - \lambda_1)} (\lambda_1 b + \lambda_2 c)^T Q^{-1} Q Q^{-1} \frac{(\lambda_1 b + \lambda_2 c)}{2(l - \lambda_1)}$$

$$= -(\lambda_1 b + \lambda_2 c)^T \cdot \frac{Q^{-1}}{2(l - \lambda_1)} (\lambda_1 b + \lambda_2 c) + \lambda_1$$

$$= -\frac{(\lambda_1 b + \lambda_2 c)^T Q^{-1} (\lambda_1 b + \lambda_2 c)}{4(l - \lambda_1)} + \lambda_1$$

$$\lambda_1 = l \quad \lambda_2 = r$$

when  $\lambda_2 = r$  we have

$$L(x, \lambda_2) = -(b + \lambda_2 c)^T x + l$$

$$\text{if } \lambda_2 < 0 \quad L(x, \lambda_2) < 0$$

$$\text{then } L(x, \lambda_2) \rightarrow -\infty \text{ as } x \rightarrow \infty$$

$$\text{if } \lambda_2 \geq 0 \quad -(b + \lambda_2 c)^T x \geq 0$$

$$\text{if } \lambda_2 = 0 \quad \text{then } x = 0$$

$$\text{if } \lambda_2 > 0 \quad \text{then } x = -\frac{b}{c}$$

$$\text{else } \lambda_2 = -\frac{b}{c}$$

for  $\lambda_1$  we have

$$q(\lambda_1, \lambda_2) = \begin{cases} \lambda_1 - \frac{\lambda_1 b^T Q^{-1} (\lambda_1 b + \lambda_2 c)}{4(l-\lambda_1)} & 0 \leq \lambda_1 < l, \lambda_2 \geq 0 \\ l & \lambda_1 = l, \lambda_2 = -\frac{b}{c} \\ -\infty & \text{else.} \end{cases}$$

$$\text{Max } q(\lambda_1, \lambda_2)$$

s.t.

$$0 \leq \lambda_1 \leq l, \lambda_2 \geq 0 \quad \text{if } \lambda_1 = l, \lambda_2 = -\frac{b}{c}$$

$$\text{לעומת } \beta = b^T Q^{-1} b \quad \text{לעומת } \beta = \frac{\beta}{2}$$

בנוסף לכך נזקק גם גורם נגדי

$$\frac{2 - \beta - \sqrt{\beta(\beta+4)}}{2}$$

בכך

המקרה השני:

$$P^* \geq q(\lambda_1, \lambda_2) \quad \forall (\lambda_1, \lambda_2) \in \text{dom}(q)$$

במקרה השלישי ( $\lambda_1 < 1$ ;  $\lambda_2 = 0$ ) נוכיח כי

$$q(\lambda_1, 0) = -\frac{\lambda_1 b^T Q^{-1} \lambda_1 b}{4(1-\lambda_1)} + \lambda_1 = \frac{\beta}{4} \cdot \frac{-\lambda_1^2}{(1-\lambda_1)} + \lambda_1$$

$$\max q(\lambda_1, 0) \Rightarrow \frac{\partial q(\lambda_1)}{\partial \lambda_1} = \frac{\beta}{4} \cdot \left( \frac{-2\lambda_1(1-\lambda_1) + \lambda_1^2 - 1}{(1-\lambda_1)^2} \right) + 1$$

$$\frac{\beta}{4} \left( \frac{-2\lambda_1 + 2\lambda_1^2 - \lambda_1^2}{(1-\lambda_1)^2} \right) + 1 = \frac{\beta}{4} \left( \frac{\lambda_1^2 - 2\lambda_1}{(1-\lambda_1)^2} \right) + 1$$

$$\cancel{\frac{\beta}{4} \left( \frac{\lambda_1^2 - 2\lambda_1}{(1-\lambda_1)^2} \right) + 1 = 0}$$

$$\frac{\lambda_1^2 \beta - 2\beta \lambda_1 + 4(\ell - \lambda_1)^2}{4(\ell - \lambda_1)^2} = 0$$

$$\frac{\beta \lambda_1^2 - 2\beta \lambda_1 + 4 - 8\lambda_1 + 4\lambda_1^2}{4(\ell - \lambda_1)^2}$$

$\Leftrightarrow$

$$(\ell + \beta) \lambda_1^2 - 2(\beta + 4) \lambda_1 + 4 = 0$$

$$\begin{aligned}\lambda_{1,2} &= \frac{2(\beta + 4) \pm \sqrt{4(\beta + 4)^2 - 4 \cdot 4(4 + \beta)}}{2(\beta + 4)} \\ &= \frac{2(\beta + 4) \pm \sqrt{(\beta + 4)(4\beta + 16 - 16)}}{2(\beta + 4)}\end{aligned}$$

$$= \ell \pm \frac{\sqrt{(\beta + 4)\beta}}{\beta + 4} = \ell \pm \sqrt{\frac{\beta}{\beta + 4}}$$

$0 \leq \lambda_1 < 1$  נסיעה ירידת פולינום יישול נכון

$$\lambda_1 = \ell - \sqrt{\frac{\beta}{\beta + 4}}$$

$q_r(\lambda_1) \rightarrow 0$

$$q_r\left(\ell - \sqrt{\frac{\beta}{\beta + 4}}\right) = \frac{\beta}{4} \cdot \frac{-\left(\ell - \sqrt{\frac{\beta}{\beta + 4}}\right)^2}{\left(\ell - 1 + \sqrt{\frac{\beta}{\beta + 4}}\right)} + \ell - \sqrt{\frac{\beta}{\beta + 4}}$$

$$\begin{aligned}
&= -\frac{\beta}{4} \cdot \frac{\left(\ell - 2\sqrt{\frac{\beta}{\beta+4}} + \frac{\beta}{\beta+4}\right)}{\frac{\sqrt{\beta}}{\sqrt{\beta+4}}} + 1 - \sqrt{\frac{\beta}{\beta+4}} \\
&= -\frac{\ell}{4} \cdot \sqrt{\beta+4} \sqrt{\beta} \left(\ell - 2\sqrt{\frac{\beta}{\beta+4}} + \frac{\beta}{\beta+4}\right) + 1 - \sqrt{\frac{\beta}{\beta+4}} \\
&= -\frac{\sqrt{\beta(\beta+4)}}{4} + \frac{\beta}{2} - \frac{\beta\sqrt{\beta}}{4\sqrt{\beta+4}} + 1 - \sqrt{\frac{\beta}{\beta+4}} \\
&= \frac{-\sqrt{\beta}(\beta+4) + 2\beta\sqrt{\beta+4} - \beta\sqrt{\beta} + 4\sqrt{\beta+4} - 4\sqrt{\beta}}{4\sqrt{\beta+4}} \\
&= \frac{-\sqrt{\beta}(\beta+4 + 2\beta + 4) - (2\beta + 4)\sqrt{\beta+4}}{4\sqrt{\beta+4}} \\
&= \frac{-\sqrt{\beta}2(\beta+4) + 2(\beta+2)\sqrt{\beta+4}}{4\sqrt{\beta+4}} \\
&= \frac{\beta+2 - \sqrt{\beta(\beta+4)}}{2}
\end{aligned}$$

$(\lambda_1, \lambda_2) \in \mathbb{R}^2$   $\Rightarrow \lambda_1 = 1 - \sqrt{\frac{\beta}{\beta+4}} < 1 - 1 \quad \lambda_2 = 0 \quad \text{ר'ז}$   $p_1$   
 $\exists (m, r) \in \mathbb{N}^2 \quad q = \frac{\beta+2 - \sqrt{\beta(\beta+4)}}{2} \quad p, r \in \mathbb{N}$

$$p^* \geq q \quad \blacksquare$$

3.5.8

הוכיחו  $a, b \in \mathbb{R}^n$  ו-  $\mathbb{R}^{n \times n} \ni Q \succ 0$  כך  
הנימוק כהירות  $x^* \in \mathbb{R}^n$  נקבע:

$$\min_x \frac{x^T Q x}{x^T a}$$

(ג) כ. הוכחה א.מ.ר  
Slater י.ג. מינימום

הוכחה

$$\text{s.t. } x^T b = c \\ x^T a > 0$$

$$\alpha = a^T Q^{-1} a$$

$$\beta = b^T Q^{-1} b$$

$$r = a^T Q^{-1} b = b^T Q^{-1} a$$

$$\alpha\beta > r^2$$

הוכחה כ. הוכחה ב. מינימום

לבונן (לעומת)  $y = x^T a$  מינימום כלי

$$\min_{x,y} \frac{x^T Q x}{y}$$

$$x^T b = c \quad y = x^T a$$

$$y > 0$$

(לפי קיון)

$$Q \geq 0 \text{ קיינה } \Leftrightarrow \text{וכך } g(x) = x^T Q x \geq 0 \quad (1)$$

-ב נזכיר כי  $y > 0$  כי  $y \in \mathbb{R}$

$$h(x, y) = y \cdot g\left(\frac{x}{y}\right) = y \cdot \frac{x^T Q x}{y} = \frac{x^T Q x}{y}$$

קיינה. אך פונקציית  $y$  היא פונקציה רציפה.

(בנוסף ל 2. ו 3. מינימום ו מקסימום)

$$\begin{aligned} L(x, \mu_1, \mu_2) &= \frac{\ell}{2} x^T Q x + \mu_1 (x^T b - c) + \mu_2 (x^T a - y) \\ &= \frac{\ell}{2} x^T Q x + (\mu_1 b + \mu_2 a) - \mu_1 c - \mu_2 y \end{aligned}$$

למי פונקציה רציפה כ.א. וכאן קיינה.

$$\frac{\partial L}{\partial x} = \frac{\ell}{2} Q x + (\mu_1 b + \mu_2 a) = 0$$

$$x^* = -\frac{\ell}{2} Q^{-1} (\mu_1 b + \mu_2 a).$$

$$g(y) = -\frac{\ell}{2}(\mu_1 c + \mu_2 y)$$

$$= -\frac{\ell}{2} \cdot \left( \frac{(2ry - 2cr)\kappa}{y(\alpha\beta - r^2)} + \frac{y(-2y\beta + 2cr)}{y(\alpha\beta - r^2)} \right)$$

$$= -\frac{\ell}{2} \cdot \left( \frac{2rcy - 2c^2\alpha - 2y^2\beta + 2ycr}{y(\alpha\beta - r^2)} \right)$$

$$= \frac{y^2\beta - 2cr^2y + c^2\alpha}{y(\alpha\beta - r^2)}$$

הנתקה מינימלית

$$\min_y g(y)$$

$$y > 0$$

נבדק כי  $y \in \mathbb{R}$  הוא מינימום

$$g'(y) = \frac{(2y\beta - 2cr)\kappa y(\alpha\beta - r^2) - (y^2\beta - 2cr^2y + c^2\alpha)(\alpha\beta - r^2)}{y^2(\alpha\beta - r^2)^2} = 0$$

$$\frac{2y^2\beta - 2cr^2y - y^2\beta + 2cr^2y - c^2\alpha}{y^2(\alpha\beta - r^2)} = 0$$

$$\Rightarrow y^2\beta - c^2\alpha = 0 \quad \Rightarrow \quad y^2 = \frac{c^2\alpha}{\beta}$$

$$y > 0 \Rightarrow y^* = C \sqrt{\frac{\alpha}{\beta}}$$

O. כיד:

לפנינו מושג  $y = x^\alpha$  ו $\alpha < 0$

+ הנקודה  $(x_0, y_0)$  מוגדרת

$y(x)$  נא שתהה בנקודה  $x_0$  ו $y_0$  מוגדרות

$y > 0$  -& למשתנה  $x$  מוגדרות

$$y^* = C \sqrt{\frac{\alpha}{\beta}} \quad \text{ו-} \quad (x_0, y_0) \in$$

$$\Rightarrow x^{*\alpha} = C \sqrt{\frac{\alpha}{\beta}}$$

-4 sin r

## וְעַתָּה כִּי־יְהוָה אֱלֹהִים :

$$\min_x \left\{ \frac{1}{3} x^3 - ax^2 \right\}$$

for every  $a$ ,  $x, a > 0$

ג'נָה:

סְבִּירָה וְשִׁלְוחָה

$$f'(x) = \frac{3}{3} x^2 - a ; f'(x) = 0 \Rightarrow x^2 - a = 0$$

$$x = \pm \sqrt{a}$$

لذلك  $c - \alpha < x$

$$f''(x) = 2x \Big|_{\sqrt{a}} = 2\sqrt{a} > 0$$

PINJIN pī

$$x_k = x_{k-1} - f''(x_{k-1})^{-1} f'(x_{k-1})$$

$$= x_{k-1} - \frac{l}{2x_{k-1}} \cdot (x_{k-1}^2 - a)$$

$$= \frac{\alpha x_{k-1}^2 - x_{k-1}^2 + a}{2x_{k-1}} = \frac{x_{k-1}^2 + a}{2x_{k-1}}$$

$$x_{k-1} > x_k > \sqrt{a} \quad \text{by} \quad x_{k-1} > \sqrt{a} \quad \text{pt 3}$$

ו כה:

$$x_k = \frac{x_{k-1}^2 + a}{2x_{k-1}} < x_{k-1} \quad / \cdot 2x_{k-1}$$

<=>

$$x_{k-1}^2 + a < 2x_{k-1}^2$$

$$\angle \Rightarrow a < x_{k-1}^2$$

$$\Leftrightarrow \sqrt{a} < x_{k-1}$$

$$x_k < x_{k-1} \quad \text{if} \quad x_{k-1} > \sqrt{\alpha} \quad \text{or} \quad y^*_k$$

$$\frac{x_{k-1}^2 + a}{2x_{k-1}} > \sqrt{a} \Leftrightarrow x_{k-1}^2 + a > 2x_{k-1} \cdot \sqrt{a}$$

$$\Leftrightarrow x_{k-1}^2 - 2x_{k-1} \sqrt{a} + a > 0$$

$$\Leftrightarrow (x_{k-1} - \sqrt{a})^2 > 0$$

$$f_k = \frac{f_{k-1}^2}{2(l+f_{k-1})}$$

$$(l + \sqrt{k}) \cancel{\sqrt{a}} = \frac{(l + \sqrt{k_{-1}})^2 \cancel{\sqrt{a}}^2 + d}{2(l + \sqrt{k_{-1}}) \cancel{\sqrt{a}}}$$

$$l + S_{1k} = \frac{l + 2\sqrt{k_{k-1}} + S_{k-1}^2 + 1}{2(l + \sqrt{k_{k-1}})} = \frac{2(l + \sqrt{k_{k-1}}) + S_{k-1}^2}{2(l + \sqrt{k_{k-1}})}$$

$$l + \delta_k = l + \frac{\delta_{k-1}^2}{2(l + \delta_{k-1})} \Rightarrow \delta_k = \frac{\delta_{k-1}^2}{2(l + \delta_{k-1})}$$

$$\delta_k \leq \frac{\delta_{k-1}}{2} \quad \text{ר'ו 5}$$

$$\frac{\delta_{k-1}^2}{2(l + \delta_{k-1})} \leq \frac{\delta_{k-1}}{2}$$

$$\Leftrightarrow \delta_{k-1}^2 \leq \delta_{k-1} + \delta_{k-1}^2$$

$$\Leftrightarrow \delta_{k-1} \geq 0$$

$\delta_{k-1} > \sqrt{a} - l$  מגדיר  $\delta_{k-1} \geq 0$  כ ר'ל,

$$(l + \delta_{k-1})\sqrt{a} > \sqrt{a} \Leftrightarrow l + \delta_{k-1} > 1 \Leftrightarrow \delta_{k-1} > 0.$$

$$x_k = (l + \delta_k)\sqrt{a} \leq (l + \frac{\delta_{k-1}}{2})\sqrt{a} \leq \dots \leq (l + \frac{\delta_0}{2^k})\sqrt{a}$$

$$\xrightarrow{k \rightarrow \infty} (l + 0)\sqrt{a} = \sqrt{a}!$$

$$\delta_k \leq \frac{\delta_{k-1}^2}{2} \quad \text{ר'ו 6}$$

$$\frac{\delta_{k-1}^2}{2(l + \delta_{k-1})} \leq \frac{\delta_{k-1}^2}{2} \Leftrightarrow \frac{l}{l + \delta_{k-1}} \leq 1 \Leftrightarrow \delta_{k-1} \geq 0$$

7. עין ט.  $x_0 = \alpha$  מינימום מקומי  $\alpha = 2$

(2), (1) מינימום מקומי  $\alpha$  ב- $x_0$  ו- $x_1$   
 מינימום מקומי  $\alpha$  ב- $x_0$  ו- $x_2$  (3)-1  
 מינימום מקומי  $\alpha$  ב- $x_1$  ו- $x_2$  50 & 213 מטר

$$x_0 = \alpha = \sqrt{2}(\delta_0 + 1)$$

$$\sqrt{2} = \delta_0 + 1$$

$$\Rightarrow \delta_0 = \sqrt{2} - 1$$

$x_k$  מ-2.02.7

$$x_k - \sqrt{\alpha} \leq 10^{-50}$$

$$\Rightarrow \sqrt{\alpha}(\delta_{k+1}) - \sqrt{\alpha} \leq 10^{-50}$$

$$\sqrt{\alpha} \delta_k \leq 10^{-50}$$

$$\delta_k \leq \frac{10^{-50}}{\sqrt{\alpha}} = \frac{10^{-50}}{\sqrt{2}}$$

$$\delta_k \leq \frac{\delta_{k-1}}{2} \leq \frac{\delta_{k-2}}{4} \leq \dots \leq \frac{\delta_0}{2^k}$$

$$\frac{\delta_0}{2^k} \leq \frac{10^{-50}}{\sqrt{2}} \Leftrightarrow \frac{\sqrt{2}(\sqrt{2}-1)}{10^{-50}} \leq 2^k \quad k \geq 165.3$$

$$k = 166.$$

: (3) נס

$$\delta_k \leq \frac{\delta_{k-1}^2}{2} \leq \left( \frac{\delta_{k-2}^2}{2} \right)^2 = \frac{\delta_{k-2}^{2^2}}{2^3} \leq \dots \leq 2^{-2^{k-1}} \cdot \delta_0^{2^k}.$$

$$2^{-2^{k-1}} \cdot \delta_0^{2^k} \leq \frac{10^{-50}}{\sqrt{2}} / \log_2$$

$$(-2^{k-1} + 2^k \log(\delta_0)) \leq -50 \log(10) - \log(\sqrt{2})$$

$$(-2^{k-1} + 2^k \log(\delta_0)) \leq -50 \log(10) - \frac{k}{2} - 1$$

$$2^k \geq \frac{50 \log(10) - 1.5}{1 - \log(\sqrt{2}) - 1}$$

$$2^k \geq 72.46$$

$$k \geq 6.2$$

$$\boxed{k = 7}$$

ריבוע 3.17

```

1 import numpy as np
2
3 a = np.sqrt(2)
4
5
6 def _stop_condition(current_delta):
7     global a
8     return current_delta <= (1e-50) / a
9
10
11 def _next_delta_1(current_delta):
12     return (current_delta ** 2) / (2 * (1 + current_delta))
13
14
15 def _next_delta_2(current_delta):
16     return current_delta / 2
17
18
19 def _next_delta_3(current_delta):
20     return (current_delta ** 2) / 2
21
22
23 def _evaluations(update_function, delta_0):
24     k = 0
25     current_delta = delta_0
26     while not _stop_condition(current_delta):
27         k += 1
28         current_delta = update_function(current_delta)
29     return k
30
31
32 if __name__ == '__main__':
33     delta_0 = a - 1
34     print("1st expression reached bound in {} iterations".format(_evaluations(_next_delta_1, delta_0)))
35     print("2nd expression reached bound in {} iterations".format(_evaluations(_next_delta_2, delta_0)))
36     print("3rd expression reached bound in {} iterations".format(_evaluations(_next_delta_3, delta_0)))
37

```

פלט, מספר איטרציות לכל סוף:

```

1st expression reached bound in 7 iterations
2nd expression reached bound in 166 iterations
3rd expression reached bound in 7 iterations

```

5.5.2

$$(P_1) \min_{x,y} \{x^2 + y^2 + 5x - 10y \mid |x - y/2 - 2| \leq 1\}$$

$$(P_2) \min_{x,y} \{ (x-3)^2 + y^2 + 3x - 2y \mid \begin{array}{l} (x-1)^2 + (y-2) \leq 9; \\ x \geq 2, y \geq 0 \end{array}\}$$

הלו כ הוכח קאיין

לוכיח:

שכ"ז  $\exists$  גłów הוכח  $\forall P_1$  הוכח כי  
כל חירז  $\in \mathbb{R}^2$  נמצ

$$f_1(x,y) = x^2 + y^2 + 5x - 10y$$

$$\nabla f_1(x,y) = \begin{bmatrix} \nabla_x f_1(x,y) \\ \nabla_y f_1(x,y) \end{bmatrix} = \begin{bmatrix} 2x+5 \\ 2y-10 \end{bmatrix}$$

$$\nabla^2 f_1(x,y) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \succ 0$$

כל  $x, y \in \mathbb{R}^2$  מתקיים.

$$\Rightarrow \nabla^2 f_1(x,y) \succ 0$$

ולכן הוכח  $P_1$  קאיין.

הנעה (P<sub>2</sub>):

לעומת הולמת קניין ב-10 כינזים  
ב-10 כינזים ו-100 נספחים  
קנץ וזה כמו הולמת.

$$f_2(x, y) = (x-3)^2 + y^2 + 3x - 2y$$

$$\nabla f_2(x, y) = \begin{bmatrix} 2(x-3) + 3 \\ 2y - 2 \end{bmatrix}$$

$$\nabla^2 f_2(x, y) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

הנעה (P<sub>1</sub>) PD גורמי נספחים  
ולא מושג הולמת

## שאלה 5 סעיפים :2,3

לוד:

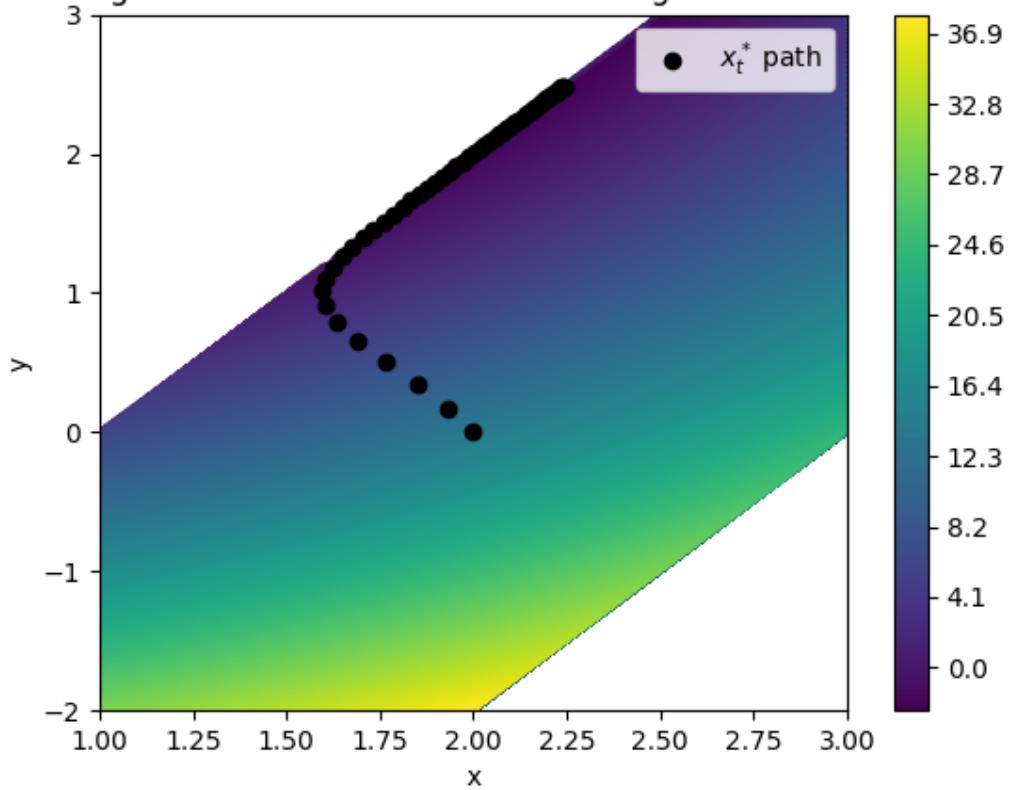
```

1 import numpy as np
2 from matplotlib import pyplot as plt
3
4
5 class MinimizationProblem:
6     def __init__(self, name: str, number_constraints):
7         self.name: str = name
8         self.number_constraints: int = number_constraints
9         self.target_func = None
10        self.dLdx = self.dLdy = self.ddLdxdy = self.ddLddx = self.ddLddy = None
11        self.constraints = None
12        self.starts_vector = None
13
14    def gradient(self, x, y, t):
15        return np.array([self.dLdx(x, y, t), self.dLdy(x, y, t)])
16
17    def hessian(self, x, y, t):
18        hess = np.array([[self.dLdxdx(x, y, t), self.ddLdxdy(x, y, t)], [self.ddLdxdy(x, y, t), self.ddLddy(x, y, t)]])
19        assert hess.shape == (2, 2)
20        return hess
21
22
23    def solve_inner(current_x_vec, t, p:MinimizationProblem, num_iter=5):
24        x_vec = current_x_vec
25        for i in range(num_iter):
26            x_vec = x_vec - 0.01 * np.linalg.inv(p.hessian(x_vec[0], x_vec[1], t)) @ p.gradient(x_vec[0], x_vec[1], t)
27        return x_vec
28
29
30    def solve_outer(mu, t_0, epsilon, p:MinimizationProblem):
31        x_vecs = [p.starts_vector]
32        t = t_0
33        while p.number_constraints / t > epsilon:
34            next_vec = solve_inner(x_vecs[-1], t, p)
35            x_vecs.append(next_vec)
36            t *= mu
37        return x_vecs
38
39
40    def init_p1():
41        p_1_problem = MinimizationProblem("P1", 2)
42        p_1_problem.constraints = lambda x, y: abs(x - 0.5 * y - 2) <= 1
43        p_1_problem.target_func = lambda x, y: (x ** 2) + (y ** 2) + 5 * x - 10 * y
44        p_1_problem.dLdx = lambda x, y, t: t * (2 * x + 5) - 1 / ((x - 0.5 * y - 3) - 1 / ((x - 0.5 * y - 1))
45        p_1_problem.ddLdxdx = lambda x, y, t: 2 * t + 1 / ((x - 0.5 * y - 3) ** 2) + 1 / ((x - 0.5 * y - 1) ** 2)
46        p_1_problem.dLdy = lambda x, y, t: t * (2 * y - 10) + 0.5 / ((x - 0.5 * y - 3) + 0.5 / ((x - 0.5 * y - 1))
47        p_1_problem.ddLddy = lambda x, y, t: 2 * t + 0.25 / ((x - 0.5 * y - 3) ** 2) + 0.25 / ((x - 0.5 * y - 1) ** 2)
48        p_1_problem.ddLdxdy = lambda x, y, t: -0.5 / ((x - 0.5 * y - 3) ** 2) - 0.5 / ((x - 0.5 * y - 1) ** 2)
49        p_1_problem.starts_vector = np.array([2, 0])
50        return p_1_problem
51
52
53    def init_p2():
54        p_2_problem = MinimizationProblem("P2", 3)
55        p_2_problem.constraints = lambda x, y: ((x - 1) ** 2 + (y - 2) ** 2 <= 9) * (x >= 2) * (y >= 0)
56        p_2_problem.target_func = lambda x, y: ((x - 3) ** 2) + (y ** 2) + 3 * x - 2 * y
57        p_2_problem.dLdx = lambda x, y, t: t * (2 * x - 3) + 2 * (x - 1) / (-((x - 1) ** 2) - ((y - 2) ** 2) + 9) - 1 / (x - 2)
58        p_2_problem.dLdy = lambda x, y, t: t * (2 * y - 2) + 2 * (y - 2) / (-((x - 1) ** 2) - ((y - 2) ** 2) + 9) - 1 / y
59        p_2_problem.ddLdxdy = lambda x, y, t: 4 * (x - 1) * (y - 2) / (((x - 1) ** 2) - ((y - 2) ** 2) + 9) ** 2
60        p_2_problem.ddLdxdx = lambda x, y, t: 2 * t + 2 / (-((x - 1) ** 2) - ((y - 2) ** 2) + 9) + 4 * ((x - 1) ** 2) / (
61            -((x - 1) ** 2) - ((y - 2) ** 2) + 9) ** 2
62        p_2_problem.ddLddy = lambda x, y, t: 2 * t + 2 / (-((x - 1) ** 2) - ((y - 2) ** 2) + 9) + 4 * ((y - 2) ** 2) / (
63            -((x - 1) ** 2) - ((y - 2) ** 2) + 9) ** 2
64        p_2_problem.starts_vector = np.array([3, 1])
65
66        return p_2_problem
67
68
69    if __name__ == '__main__':
70        p_1 = init_p1()
71        p_2 = init_p2()
72
73        mu = 1.5
74        t0 = 1
75        epsilon = 1e-10
76        n = 500
77        x_values_p1 = solve_outer(mu, t0, epsilon, p_1)
78        x_values_p2 = solve_outer(mu, t0, epsilon, p_2)
79        plot_cont(p_1, xlims=[1, 3], ylims=[-2, 3], x_values=x_values_p1, fig_path="p1.png")
80        plot_cont(p_2, xlims=[1.5, 4], ylims=[-0.5, 5], x_values=x_values_p2, fig_path="p2.png")
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

```

פלט של הגרפים:

Solving P1 with Newton's Method and Log Barrier Function



Solving P2 with Newton's Method and Log Barrier Function

