01208583 Robotics

LECTURE 2 : OPTIMIZATION BASICS

Dr. Varodom Toochinda

Dept. of Mechanical Engineering

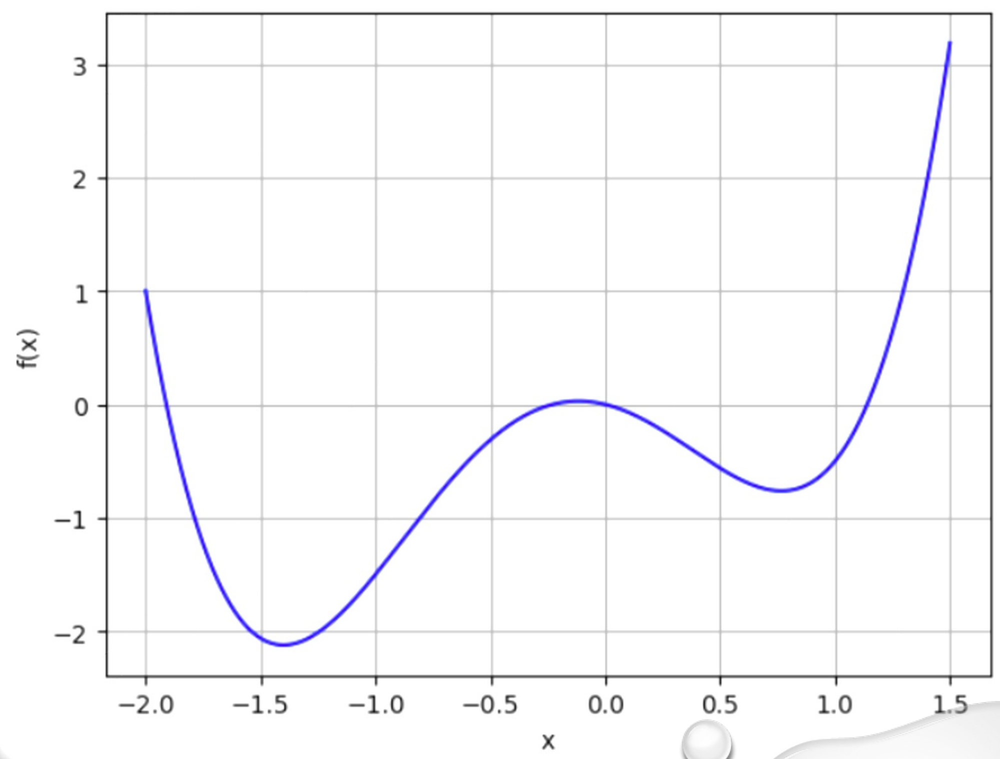
Kasetsart University

Topics

- Necessary condition for minimization
- Newton's method for root finding
- Apply Newton's method to minimization problem
- Regularization
- Line search method
- Minimization problem with equality constraints
- Minimization problem with inequality constraints

อ่านเนื้อหาโดยละเอียดและรันโค้ดสำหรับเอาต์พุตในสไลด์นี้
ได้จาก chapter2.ipynb

$$f(x) = x^4 + x^3 - 2x^2 - 0.5x$$



$$\frac{\partial f(x)}{\partial x} = 0$$

เงื่อนไขจำเป็นสำหรับค่าต่ำสุด

การหาค่ารากโดยวิธีนิวตัน

ในการหาจุดที่ค่าความชันเป็นศูนย์ สามารถประยุกต์ใช้ *วิธีนิวตัน (Newton's method)* ที่ใช้ในการวิเคราะห์เชิงเลขสำหรับหาค่ารากของฟังก์ชันจำนวนจริง ซึ่งก็คือจุดที่ฟังก์ชันมีค่าเป็นศูนย์ นิยามเชิงคณิตศาสตร์คือ สำหรับฟังก์ชัน $f(x)$ หาค่าของ x^* ที่ทำให้ $f(x^*) = 0$

ในบริบทของระบบควบคุมที่เป็นเนื้อหาหลักของหนังสือ อาจเปรียบเทียบได้กับ *จุดสมดุล (equilibrium)* ของพลวัตในระบบเวลาต่อเนื่อง หรือหากพิจารณาใน *ระบบเวลาวิยุต (discrete-time)* จุดสมดุลที่เสถียรเรียกว่า *จุดตรึง (fixed point)* ที่ทำให้ $f(x^*) = x^*$ สังเกตว่าทั้งสองรูปแบบมีความสัมพันธ์กัน เพราะเราสามารถแปลงปัญหาจุดตรึงเป็นการหาค่ารากได้โดยเขียนอยู่ในรูป $f(x^*) - x^* = 0$

การหาค่าราก โดยวิธีนิวตัน

วิธีการของนิวตันเริ่มต้นจากการประมาณค่าเชิงเส้นฟังก์ชัน $f(x)$

$$f(x+\Delta x) \approx f(x) + \frac{\partial f(x)}{\partial x}\Delta x \qquad (2.3)$$

ตั้งให้การประมาณค่านี้เท่ากับศูนย์

$$f(x) + \frac{\partial f(x)}{\partial x}\Delta x = 0 \qquad (2.4)$$

หาคำตอบ Δx ได้ดังนี้

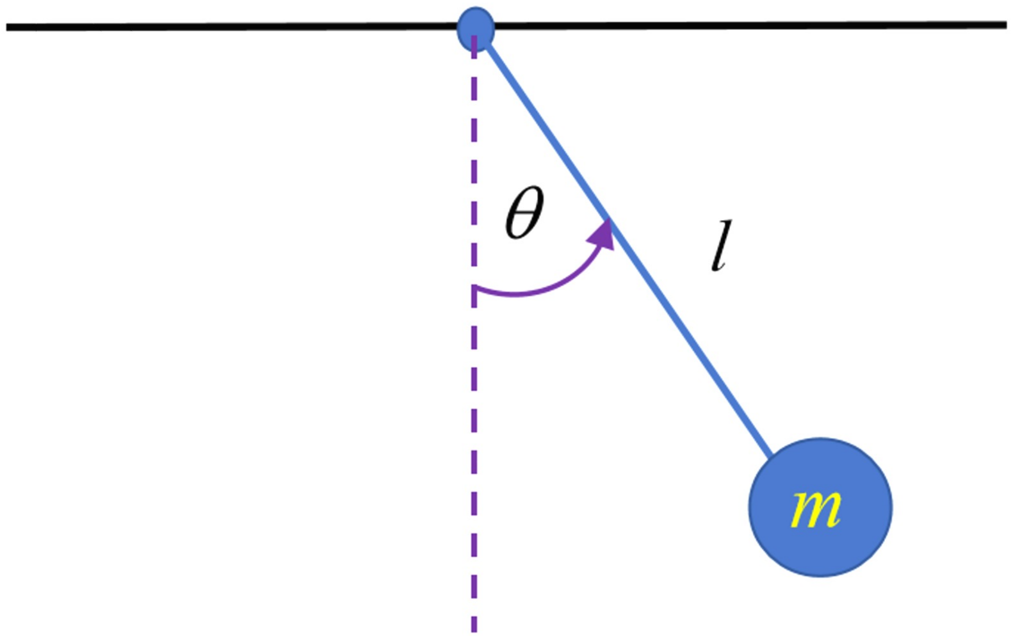
$$\Delta x = -\left(\frac{\partial f(x)}{\partial x}\right)^{-1} f(x) \qquad (2.5)$$

การหาค่ารากโดยวิธีนิวตัน

- คาดเดาค่าเริ่มต้น x
- คำนวณขั้นการเปลี่ยนค่า Δx จาก (2.5)
- ปรับค่า $x \leftarrow x + \Delta x$
- วนซ้ำจนลู่เข้า คือค่า $f(x)$ เข้าสู่ศูนย์โดยมีค่าผิดพลาดอยู่ในพิกัดที่ต้องการ

ตัวอย่าง 2.2 พลวัตลูกตุ้ม

$$ml^2\ddot{\theta} + mgl\ sin(\theta) = 0$$



รูปที่ 2.2 ลูกตุ้มสำหรับตัวอย่าง 2.2

เมื่อกำหนดสถานะ $x = [\theta \ \ \dot{\theta}]^T$ จะเขียนอยู่ในรูปเมทริกซ์ได้เป็น

$$\dot{x} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ -\frac{g}{l} sin(\theta) \end{bmatrix}$$

ตัวอย่าง 2.2 พลวัตลูกตุ้ม

หากต้องการจำลองพลวัตของลูกตุ้มบนคอมพิวเตอร์ จะต้องแปลงระบบให้อยู่ในรูปเวลาวิยุตซึ่งทำได้หลายวิธี วิธีการหนึ่งที่จะแสดงในตัวอย่างนี้โดยการหาค่ารากหรือวิธีจุดตรึงเรียกว่า *วิธีออยเลอร์ย้อนหลัง (backward Euler method)* เขียนได้ดังนี้

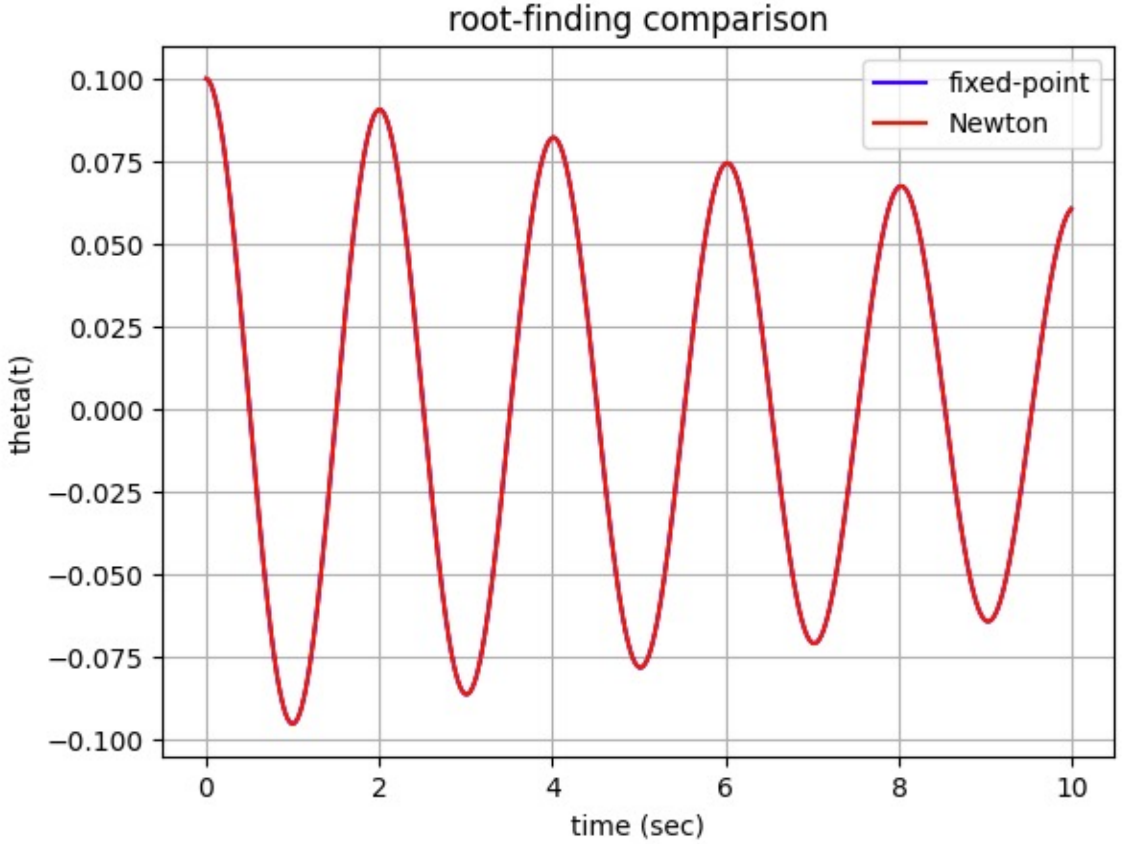
$$x_{k+1} = x_k + hf(x_{k+1}) \tag{2.8}$$

โดย h คือ*ค่าขั้นเวลา (time step)* สังเกตว่าค่าของสถานะ x_{k+1} ใน (2.5) ขึ้นกับ $f(x_{k+1})$ ซึ่งไม่สามารถคำนวณได้โดยตรง แต่เมื่อจัดรูปใหม่เป็น

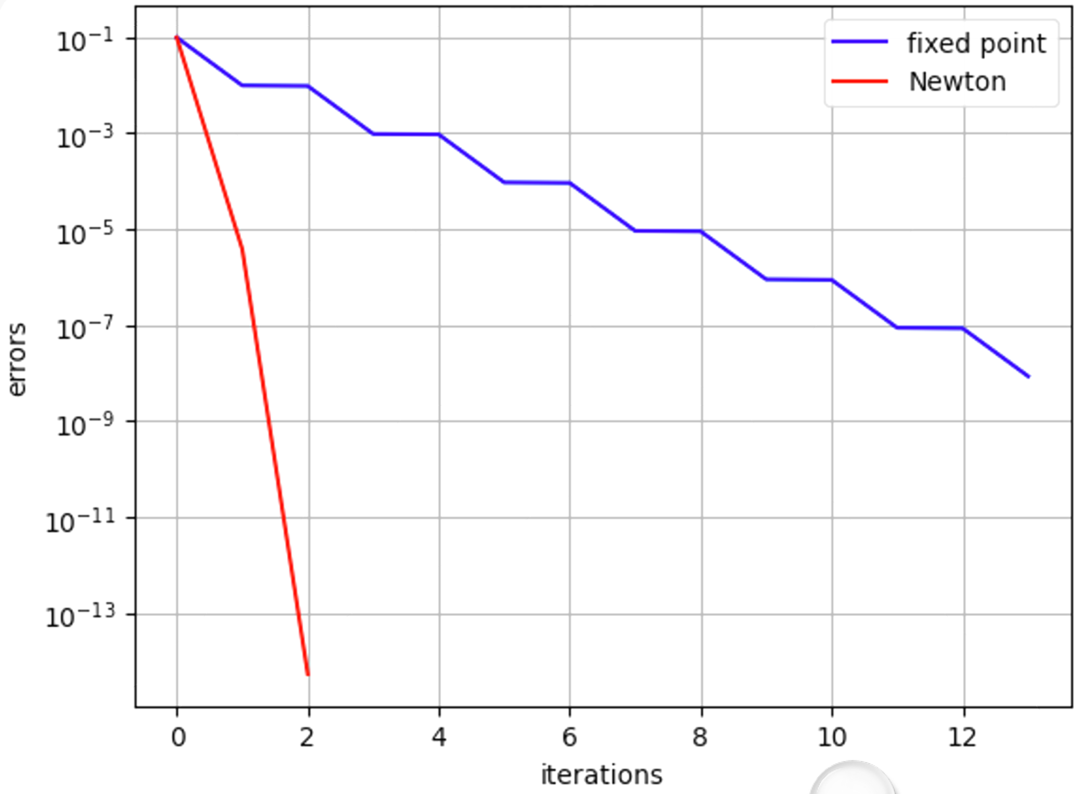
$$x_k + hf(x_{k+1}) - x_{k+1} = 0 \tag{2.9}$$

จะกลายเป็นโจทย์ที่สามารถใช้วิธีหาค่ารากหรือวิธีจุดตรึงได้ จะใช้ทั้งสองวิธีเพื่อเปรียบเทียบกัน

ผลการเปรียบเทียบ
วิธีจุดตรึง
และวิธีนิวตัน



เปรียบเทียบความเร็วในการลู่เข้าระหว่างวิธีจุดตรึงและวิธีนิวตัน



ใช้วิธีหาค่าราก ในการหาค่าต่ำสุดของฟังก์ชัน

จากสมการ (2.2) ในหัวข้อ 2.1.1 ได้กล่าวถึงเงื่อนไขจำเป็นของค่าต่ำสุดของฟังก์ชันมูลค่า คือต้องมีค่าอนุพันธ์เท่ากับศูนย์ ดังนั้นเราสามารถประยุกต์ใช้วิธีการหาค่ารากในการหาค่าต่ำสุดของฟังก์ชัน โดยพิจารณา $\nabla f(x) = \frac{\partial f(x)}{\partial x}$ และหาค่าของ x ที่ทำให้ $\nabla f(x) = 0$

จากโจทย์ปัญหาการหาค่าต่ำสุด

$$\min_x f(x), \quad f(x) : \mathbb{R}^n \to \mathbb{R} \tag{2.10}$$

เมื่อดำเนินการตามขั้นตอนของวิธีนิวตัน จะได้ว่า

$$\nabla f(x + \Delta x) \approx \nabla f(x) + \frac{\partial}{\partial x}(\nabla f(x))\Delta x = 0 \tag{2.11}$$

นิยาม $\nabla^2 f(x) = \frac{\partial}{\partial x}(\nabla f(x))$ แก้สมการ (2.11) เพื่อหาคำตอบ Δx

$$\Delta x = -(\nabla^2 f(x))^{-1} \nabla f(x) \tag{2.12}$$

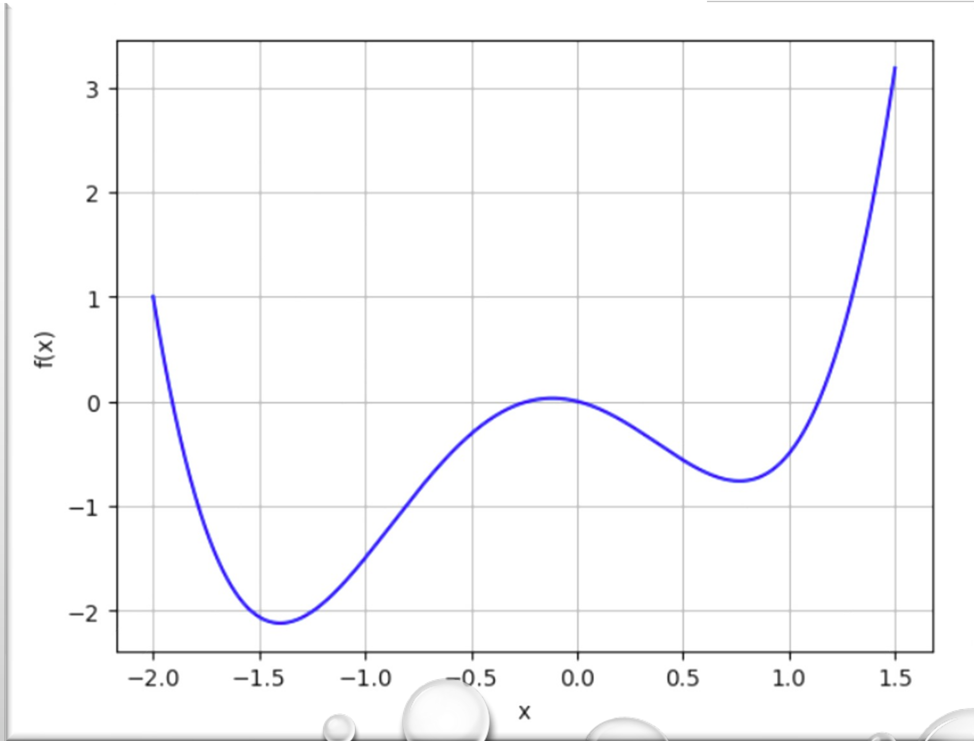
อัลกอริทึมหาค่าต่ำสุดโดยวิธีนิวตัน

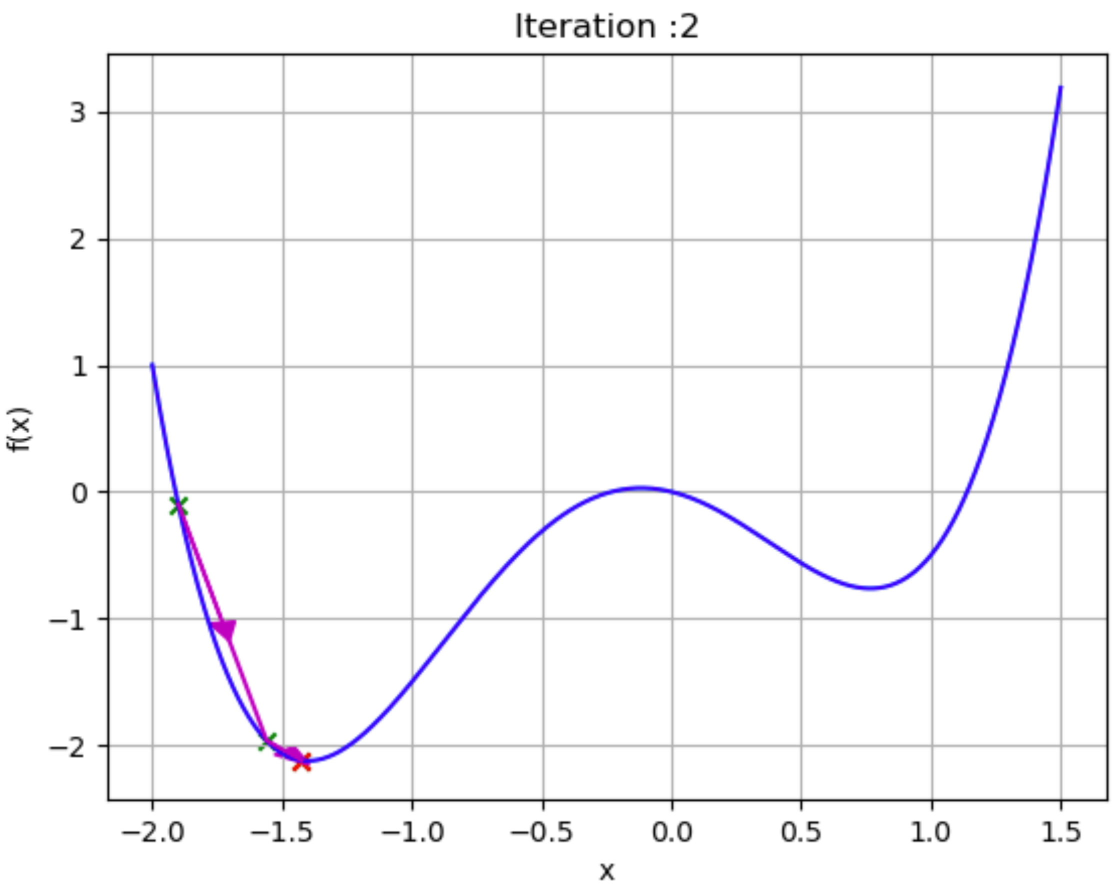
- คาดเดาค่าเริ่มต้น x
- คำนวณขั้นการเปลี่ยนค่า Δx จาก (2.12)
- ปรับค่า $x \leftarrow x + \Delta x$
- วนซ้ำจนลู่เข้า คือค่า $\nabla f(x)$ เข้าสู่ศูนย์

ตัวอย่าง 2.3 ทดลองหาค่าต่ำสุดของฟังก์ชันในตัวอย่าง 2.1

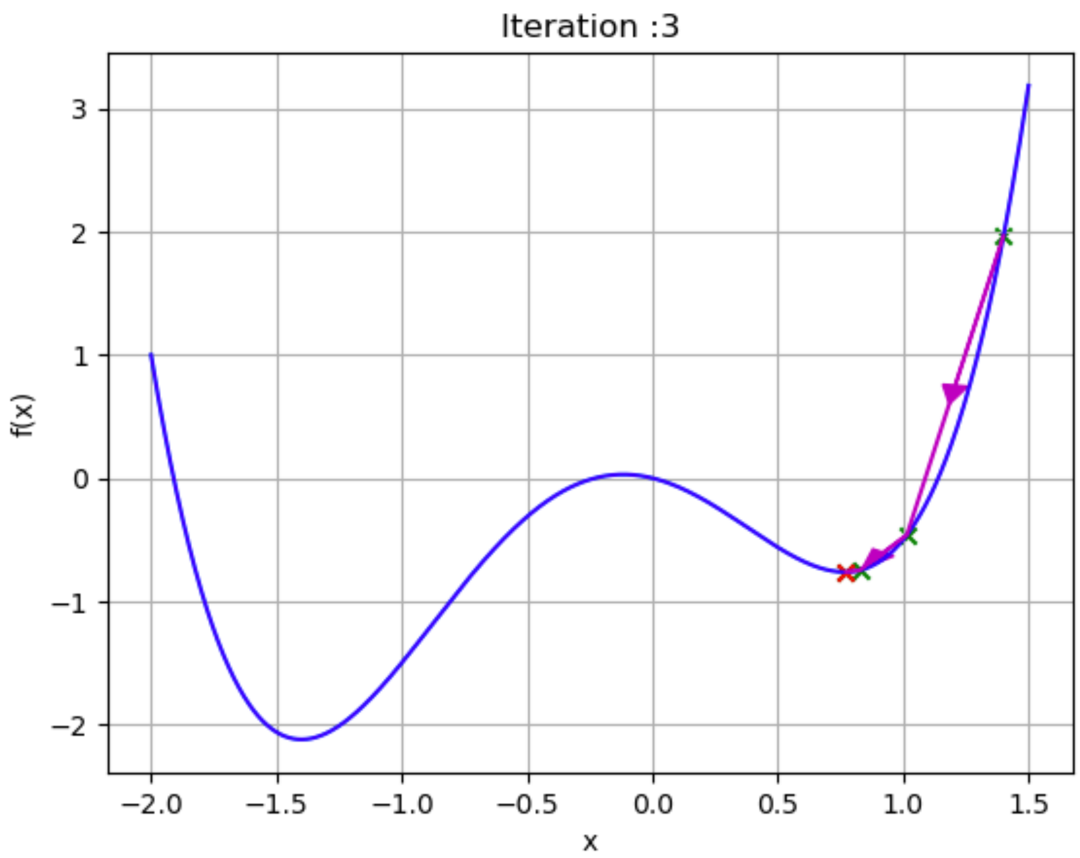
$$f(x) = x^4 + x^3 - 2x^2 - 0.5x$$

- รันโค้ดได้จาก chapter2.ipynb

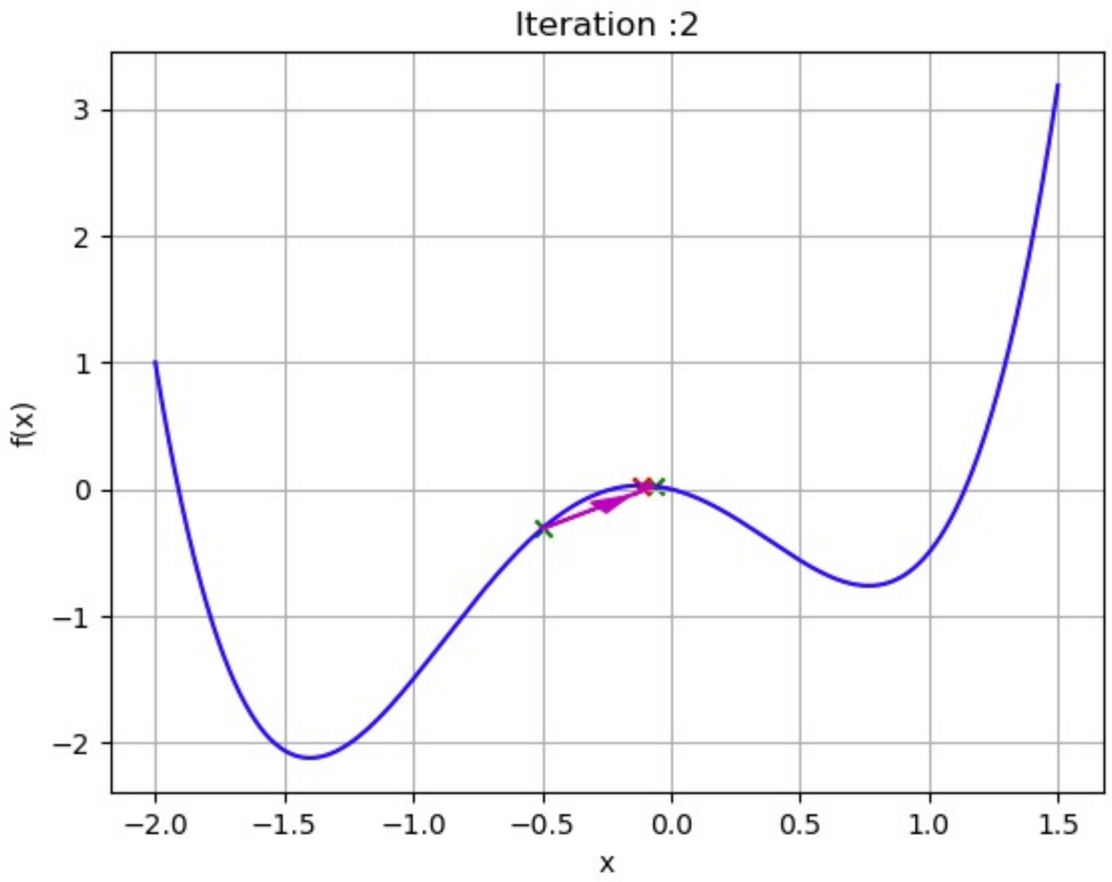




รูปที่ 2.6 การลู่เข้าสู่ค่าต่ำสุดวงกว้างจากค่าเริ่มต้น $x = -1.9$



รูปที่ 2.7 การลู่เข้าสู่ค่าต่ำสุดเฉพาะที่จากค่าเริ่มต้น $x = 1.4$



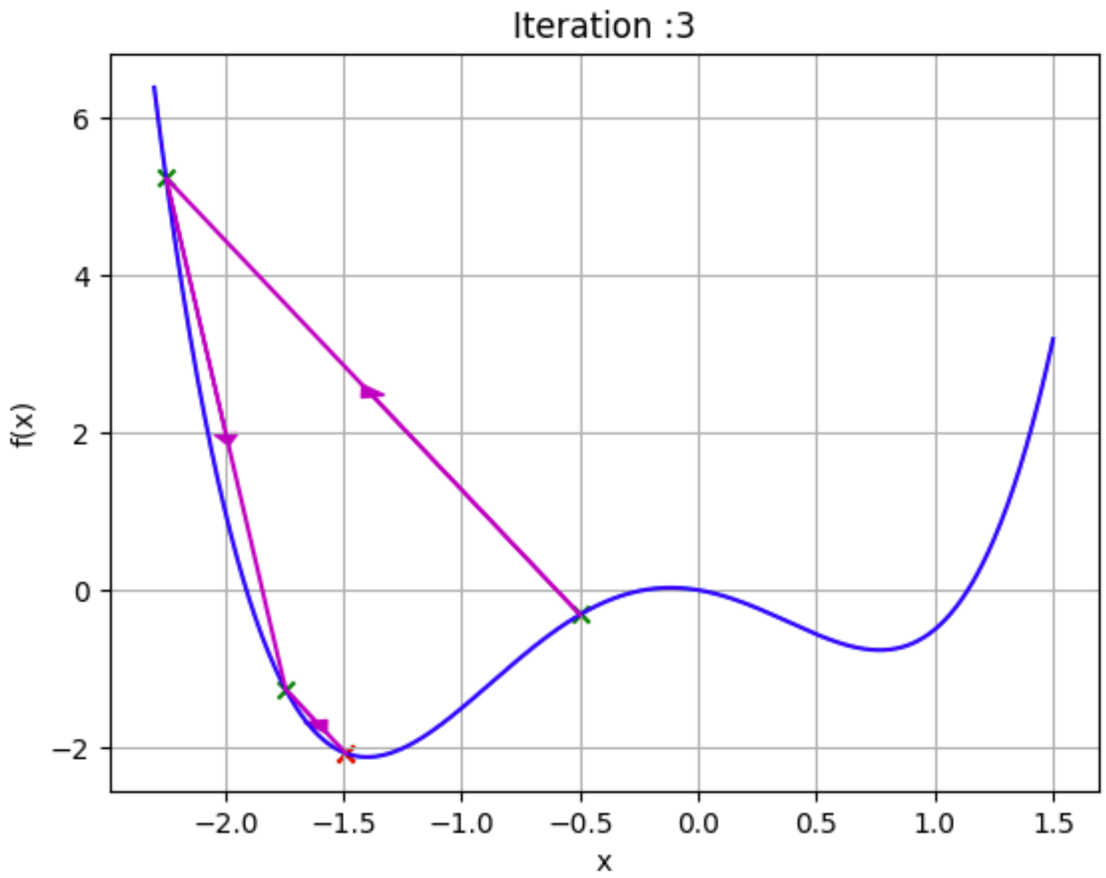
รูปที่ 2.7 การลู่เข้าสู่ค่าสูงสุดเฉพาะที่จากค่าเริ่มต้น $x = -0.5$

วิธีการเรกูลาร์ไรเซชัน (REGULARIZATION)

- คาดเดาค่าเริ่มต้น x
- นิยามเมทริกซ์ H โดยกำหนดค่า $H = \nabla^2 f(x)$
- สร้างเงื่อนไขวนซ้ำ (while) หาก H ไม่เป็นบวกแน่นอน ปรับค่า $H \leftarrow H + \beta I$ โดยค่าสเกลาร์ $\beta > 0$ คือค่าไฮเปอร์พารามิเตอร์ที่เลือกตามความเหมาะสม
- คำนวณขั้นการเปลี่ยนค่า Δx จาก

$$\Delta x = -H^{-1} \Delta f(x) \tag{2.13}$$

- ปรับค่า $x \leftarrow x + \Delta x$
- วนซ้ำจนลู่เข้า คือค่า $\nabla f(x)$ เข้าสู่ศูนย์



รูปที่ 2.8 การลู่เข้าสู่ค่าต่ำสุดโดยวิธีการเรกูลาร์ไรเซชัน

การค้นหาตามเส้น (LINE SEARCH)

จากตัวอย่าง 2.4 จะเห็นว่าในขั้นแรกการปรับค่า x ทำให้ค่า $f(x)$ มากกว่าเดิม ซึ่งสำหรับวิธีการหาค่าต่ำสุดแล้วเป็นสิ่งไม่พึงประสงค์ เพราะนอกจากทำให้ลู่เข้าสู่คำตอบช้าแล้ว อาจส่งผลเสียอื่นเช่นทำให้ออกจากย่านการดึงดูด (region of attraction) หรือโดดข้ามไปมาโดยไม่ลู่เข้าเลยก็ได้ การเพิ่มวิธีการค้นหาตามเส้น (line search) สามารถช่วยให้คำตอบลู่เข้าสู่ค่าต่ำสุดได้เร็วขึ้น แนวคิดคือตรวจสอบค่า $f(x+\Delta x)$ และ ย้อนรอย (backtrack) คือลดขนาดของ Δx จนกระทั่งได้ค่าของฟังก์ชันมูลค่าลดลง

มีหลายวิธีการที่สามารถทำได้โดยอาศัยแนวทางนี้ วิธีที่จะนำเสนอในหัวข้อนี้ใช้ เงื่อนไขของอามิโฮ (Armijo's condition) [2] ซึ่งเป็นวิธีที่ได้ผลดีในกรณีทั่วไป หลักการคือวนซ้ำเพื่อตรวจสอบเงื่อนไข

$$f(x+\alpha\Delta x) > f(x) + b\alpha\nabla f(x)^T\Delta x \tag{2.14}$$

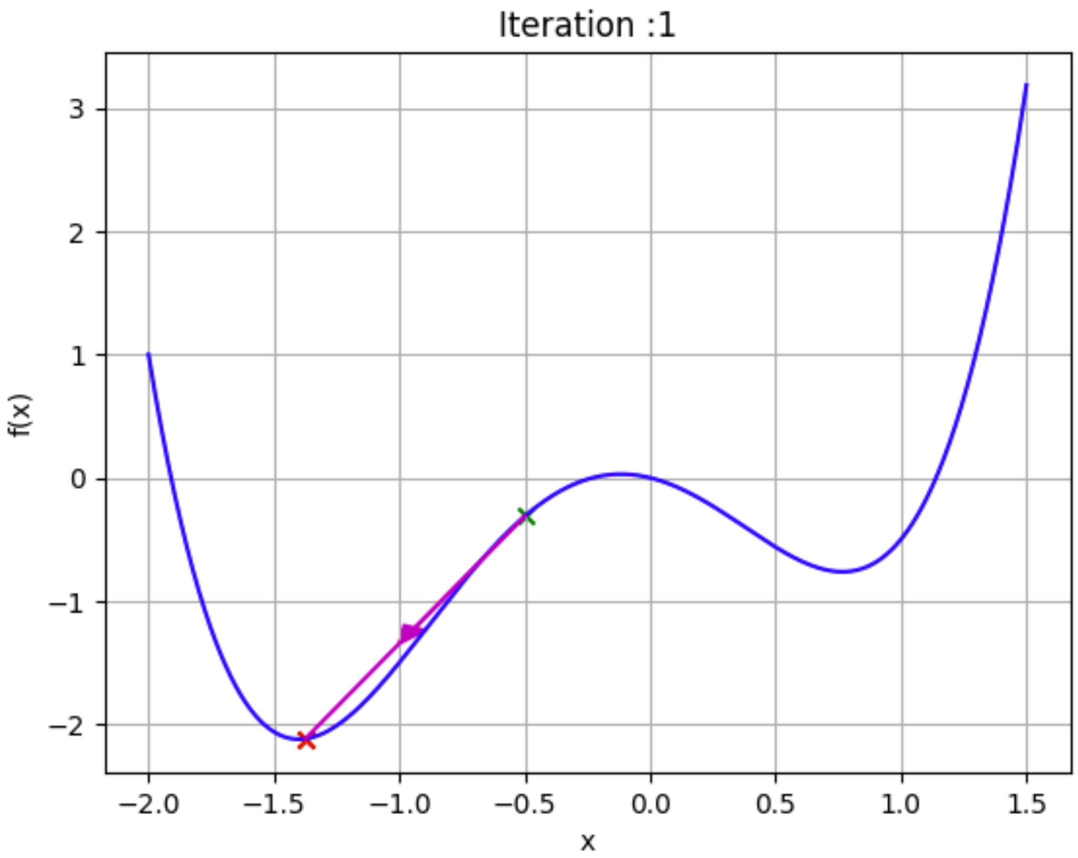
หากเป็นจริง ลดค่า $\alpha \leftarrow d\alpha, \quad d<1$ จนกว่าเงื่อนไขเป็นเท็จ พจน์สุดท้ายทางด้ายขวาของ (2.14) อธิบายได้ว่าเป็นค่าการลดที่คาดหวังจากการประมาณเชิงเส้นของฟังก์ชัน โดย b คือค่าความคลาดเคลื่อนยินยอม (tolerance) โดยทั่วไปกำหนดค่าในช่วง $0.1-10^{-4}$ ส่วนค่า $d<1$ คือตัวประกอบการลดค่า อาจใช้ค่าเช่น 0.5

อัลอกริทึมของนิวตันหลังการทำเรกูลาร์ไลเซชันถูกปรับเปลี่ยนเป็นดังนี้

- คาดเดาค่าเริ่มต้น x
- นิยามเมทริกซ์ H โดยกำหนดค่า $H = \nabla^2 f(x)$
- สร้างเงื่อนไขวนซ้ำ (while) หาก H ไม่เป็นบวกแน่นอน ปรับค่า $H \leftarrow H + \beta I$ โดยค่าสเกลาร์ $\beta > 0$ คือค่าไฮเปอร์พารามิเตอร์ที่เลือกตามความเหมาะสม
- คำนวณขั้นการเปลี่ยนค่า Δx ตาม (2.13)

- เพิ่มตัวแปร $b \in [10^{-4}, 10^{-1}], \quad d < 1$
- สร้างเงื่อนไขวนซ้ำเพื่อตรวจสอบ (2.14) หากเป็นจริง ลดค่า $\alpha \leftarrow d\alpha$ จนกว่าเงื่อนไขเป็นเท็จ

- ปรับค่า $x \leftarrow x + \Delta x$
- วนซ้ำจนลู่เข้า คือค่า $\nabla f(x)$ เข้าสู่ศูนย์



รูปที่ 2.9 ปรับปรุงการลู่เข้าสู่ค่าต่ำสุดโดยวิธีการค้นหาตามเส้น

ปัญหาการหาค่าเหมาะที่สุดแบบมีเงื่อนไข

เงื่อนไขบังคับแบบสมการ (equality constraints)

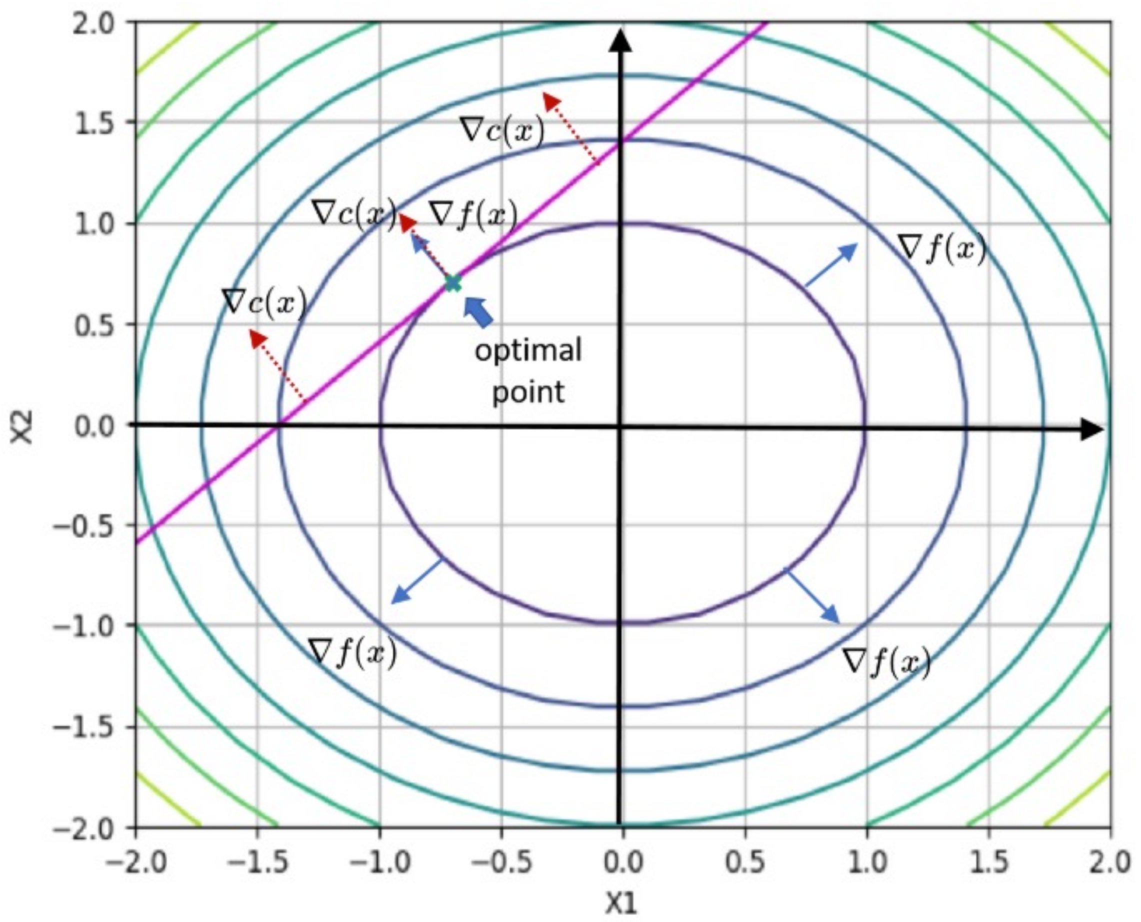
เงื่อนไขบังคับแบบสมการ (inequality constraints)

เงื่อนไขบังคับแบบสมการ

รูปแบบทางคณิตศาสตร์สำหรับการหาค่าเหมาะที่สุดที่มีเงื่อนไขบังคับแบบสมการเขียนได้ดังนี้

$$\min_{x} f(x), \quad f(x) : \mathbb{R}^n \to \mathbb{R}$$

$$s.t. \quad c(x) = 0, \quad c(x) : \mathbb{R}^n \to \mathbb{R}^m \qquad (2.15)$$



รูปที่ 2.10 ตัวอย่างการหาค่าเหมาะที่สุดบนเงื่อนไขสมการในระนาบ 2 มิติ

เงื่อนไขจำเป็น

ที่จุด x^* ที่ทำให้ $f(x^*)$ มีค่าต่ำสุดบนเงื่อนไขบังคับ เราสามารถสรุปเงื่อนไขจำเป็นได้ดังนี้

1. ต้องการ $\nabla f(x)|_{x^*} = 0$ ในทิศทางที่ x เคลื่อนที่จาก x^* ได้อิสระสอดคล้องกับเงื่อนไขบังคับ จากตัวอย่างในรูป 2.10 คือ $\nabla f(x)|_{x^*} = 0$ ในทิศทางตามเส้นตรง ซึ่งจากสามัญสำนึกของเราคือสิ่งที่ควรจะเป็น เพราะสมมุติว่าค่า x^* ทำให้ $f(x^*)$ มีค่าต่ำสุด หาก $\nabla f(x)|_{x^*} \neq 0$ บนเส้นตรงที่ x^* สามารถเคลื่อนที่ได้ การเคลื่อนที่ในทิศทาง $-\nabla f(x)$ ส่งผลทำให้ค่า $f(x) < f(x^*)$
   ขัดแย้งกับสมมุติฐานที่ว่าค่า x^* ทำให้ $f(x^*)$ มีค่าต่ำสุด
2. ค่าของ x ต้องสอดคล้องกับเงื่อนไขบังคับ $c(x) = 0$

จะได้ว่าที่จุดคำตอบ คือจุดที่เส้นโค้งระดับสัมผัสกับเส้นตรง เกรเดียนต์ $\nabla f(x)$ และ $\nabla c(x)$ จำเป็นต้องขนานกัน

ณ จุดคำตอบ องค์ประกอบใดๆ ที่ไม่เท่ากับศูนย์ของ $\nabla f(x)$ จะต้องตั้งฉากกับเงื่อนไขบังคับ $c(x) = 0$ (ในมิติที่สูงขึ้นจะเป็น surface หรือ manifold) หรือเขียนเป็นสมการคณิตศาสตร์ได้ว่า

$$\nabla f(x) + \lambda \nabla c(x) = 0 \tag{2.16}$$

ค่า λ มีชื่อเรียกว่า ตัวคูณลากรานจ์ (Lagrange multiplier) หรือ ตัวแปรคู่กัน (dual variable) จาก (2.16) มองอีกมุมหนึ่งได้ว่า λ เป็นค่าสเกลาร์ที่ปรับให้เวกเตอร์ $\nabla f(x)$ และ $\nabla c(x)$ มีขนาดเท่ากัน

ในกรณีทั่วไปที่เงื่อนไขบังคับเป็นเวกเตอร์ สามารถเขียนได้เป็น

$$\nabla f(x) + \lambda^T \nabla c(x) = 0, \quad \lambda \in \mathbb{R}^m \tag{2.17}$$

จากเงื่อนไขของเกรเดียนต์ใน (2.17) เพื่อความเหมาะสมในการจัดรูปแบบปัญหา จะนิยาม *ลากรานเจียน (Lagrangian)*

$$L(x, \lambda) = f(x) + \lambda^T c(x) \tag{2.18}$$

ใช้สัญกรณ์ $\nabla_x L$ และ $\nabla_\lambda L$ แทน $\frac{\partial L}{\partial x}$ และ $\frac{\partial L}{\partial \lambda}$ ตามลำดับ จะได้ว่า

$$\nabla_x L(x, \lambda) = \nabla f(x) + \left(\frac{\partial c(x)}{\partial x}\right)^T \lambda = 0 \tag{2.19}$$

$$\nabla_\lambda L(x, \lambda) = c(x) = 0 \tag{2.20}$$

ต้องการประยุกต์ใช้วิธีนิวตันในการแก้ปัญหา ดังนั้นจึงดำเนินการต่อตามแนวทางเดิม คือประมาณค่าเชิงเส้นโดยการกระจายเทเลอร์และให้เท่ากับศูนย์

$$\nabla_x L(x+\Delta x, \lambda+\Delta\lambda) \approx \nabla_x L(x,\lambda) + \frac{\partial^2 L}{\partial x^2}\Delta x + \frac{\partial^2 L}{\partial x \partial \lambda}\Delta\lambda = 0 \tag{2.21}$$

$$\nabla_x L(x+\Delta x, \lambda+\Delta\lambda) \approx c(x) + \frac{\partial c(x)}{\partial x}\Delta x = 0 \tag{2.22}$$

สังเกตว่าเมื่อใช้สมการ (2.20) จะได้ว่า $\frac{\partial^2 L}{\partial x \partial \lambda} = \left(\frac{\partial c(x)}{\partial x}\right)^T$ แทนค่าในพจน์สุดท้ายทางด้านซ้ายของ (2.21) และจัดรูป (2.21), (2.22) ให้อยู่ในรูปสมการเมทริกซ์

$$\begin{bmatrix} \frac{\partial^2 L}{\partial x^2} & \left(\frac{\partial c(x)}{\partial x}\right)^T \\ \frac{\partial c(x)}{\partial x} & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -\nabla_x L(x,\lambda) \\ -c(x) \end{bmatrix} \tag{2.23}$$

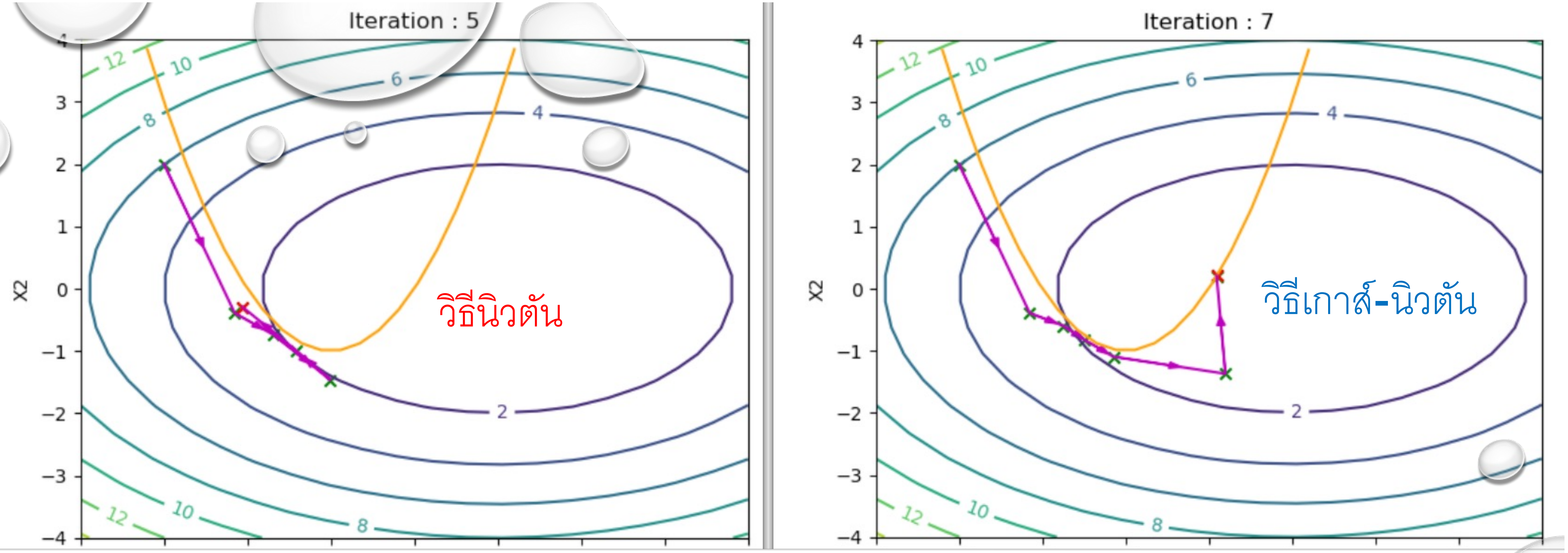
ซึ่งสามารถใช้ในอัลกอริทึมคำนวณการเปลี่ยนแปลงในแต่ละขั้นของตัวแปร x และ λ ก่อนที่จะแสดงตัวอย่าง จะชี้ประเด็นหนึ่งที่อาจสร้างปัญหาในการคำนวณและการหลีกเลี่ยง

2.2.2 วิธีการเกาส์-นิวตัน

เมื่อพิจารณาสมาชิก (1,1) ของเมทริกซ์ในสมการ (2.23) กระจายออกได้เป็นดังนี้

$$\frac{\partial^2 L}{\partial x^2} = \nabla^2 f(x) + \frac{\partial}{\partial x}\left[\left(\frac{\partial c(x)}{\partial x}\right)^T \lambda\right] \tag{2.24}$$

ปัญหาอยู่ที่พจน์สุดท้ายทางด้านขวา ซึ่งเป็นอนุพันธ์ย่อยของ $c(x)$ เทียบกับ x ซ้อน 2 ครั้ง ในกรณีที่ $c(x)$ เป็นเวกเตอร์ อนุพันธ์ย่อยชั้นแรกให้ผลลัพธ์เป็นเมทริกซ์ เมื่อหาอนุพันธ์ย่อยอีกครั้งจะได้ผลลัพธ์เป็นเทนเซอร์ แม้ว่าสามารถสร้างได้โดยภาษาคอมพิวเตอร์ในปัจจุบัน แต่จะมีความซับซ้อนในการคำนวณมากโดยเฉพาะหากข้อมูลมีขนาดใหญ่ ดังนั้นในทางปฏิบัติมักพบว่าพจน์นี้ถูกละทิ้งไป วิธีที่ทำให้ง่ายขึ้นนี้มีชื่อเรียกว่า เกาส์-นิวตัน (Gauss-Newton) ซึ่งแม้ว่าอาจลู่เข้าช้ากว่าวิธีนิวตัน แต่การคำนวณในแต่ละรอบจะน้อยกว่า ซึ่งสุดท้ายอาจพบว่าใช้เวลารวมน้อยกว่าวิธีนิวตัน



ตัวอย่าง 2.6

- รันโค้ดได้จาก chapter2.ipynb

2.2.3 เงื่อนไขบังคับแบบอสมการ

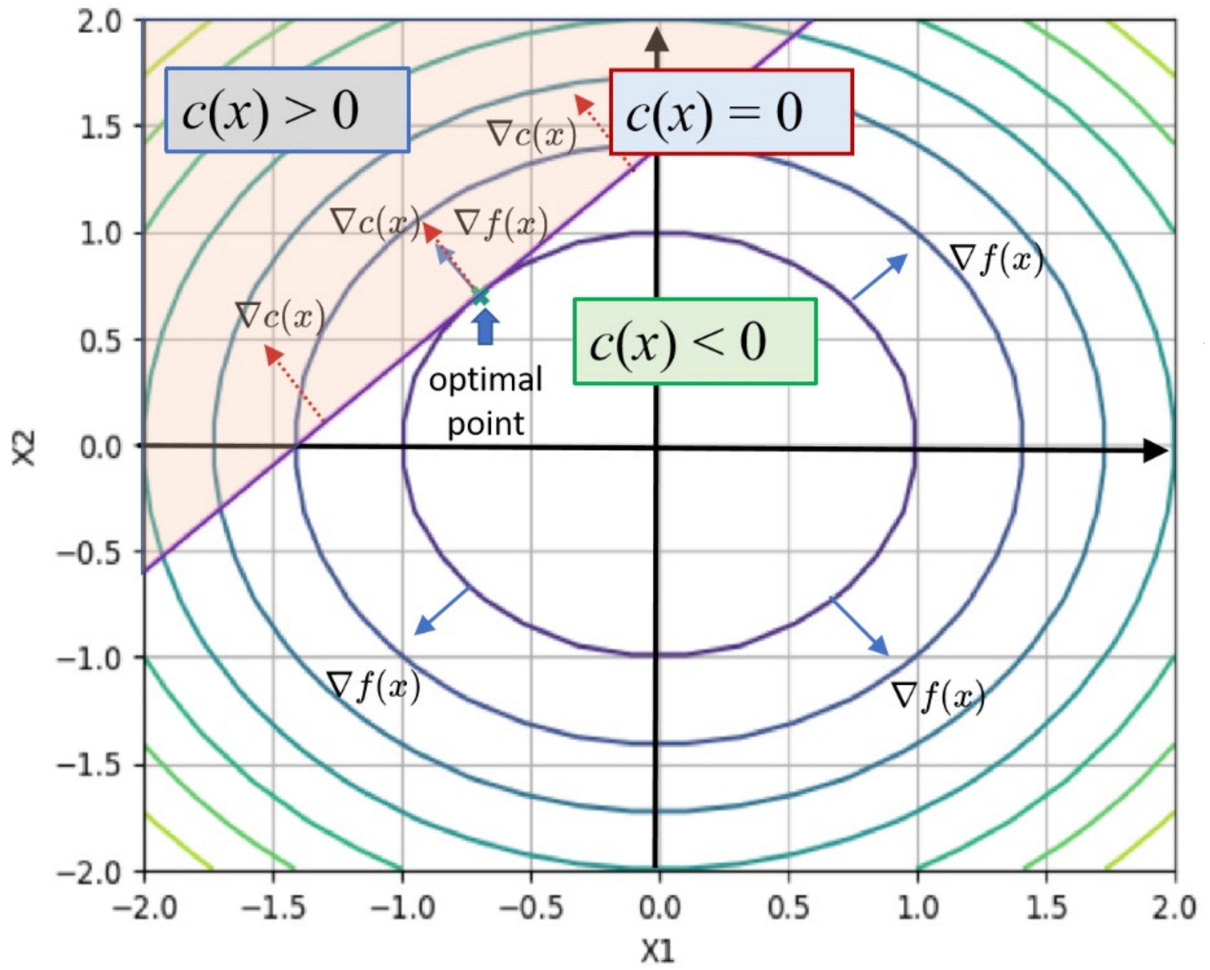
เมื่อเราเข้าใจวิธีการแก้ปัญหาต่ำสุดที่มีเงื่อนไขบังคับแบบสมการแล้ว ในหัวข้อนี้จะขยายไปยังกรณีเงื่อนไขบังคับแบบอสมการ ซึ่งเกิดขึ้นเสมอในทางปฏิบัติ กับโจทย์ปัญหาที่มีการจำกัดค่าของตัวแปร เช่นเอาต์พุตของตัวควบคุมไม่เกินจากค่าของแรงดันไฟเลี้ยง แรงบิดของข้อต่อหุ่นยนต์ขึ้นกับขนาดของมอเตอร์ ตำแหน่งของโดรนถูกจำกัดเพื่อมิให้ชนสิ่งกีดขวาง หรือเพดานของจำนวนเงินที่จัดสรรให้กับแต่ละกองทุน

รูปแบบทางคณิตศาสตร์สำหรับการหาค่าเหมาะที่สุดที่มีเงื่อนไขบังคับแบบอสมการเขียนได้ดังนี้

$$\min_x f(x), \quad f(x) : \mathbb{R}^n \to \mathbb{R}$$

$$s.t. \;\; c(x) \geq 0, \quad c(x) : \mathbb{R}^n \to \mathbb{R}^m \tag{2.25}$$

ส่วนที่เปลี่ยนแปลงจากเงื่อนไขบังคับแบบสมการคือเงื่อนไข $c(x) \geq 0$ หรือในบางตำราอาจใช้ $c(x) \leq 0$ ซึ่งสมนัยกันโดยเปลี่ยนเครื่องหมายของ $c(x)$ แม้ว่าใน (2.25) ส่วนของเงื่อนไขจะรวมกรณี $c(x) = 0$ อยู่ด้วยแล้ว แต่ในทางปฏิบัตินิยมเขียนเงื่อนไขสมการและอสมการแยกจากกัน ใน (2.25) เราเขียนเพียงเงื่อนไขอสมการเพื่อความชัดเจนในการอธิบาย ในกรณีที่โจทย์มีเงื่อนไขทั้งสองรูปแบบ เพียงแต่รวมวิธีการในหัวข้อนี้และหัวข้อก่อนหน้าเข้าด้วยกันเท่านั้น



รูปที่ 2.15 การหาค่าเหมาะที่สุดแบบมีเงื่อนไขอสมการในระนาบ 2 มิติ

หัวใจสำคัญสำหรับปัญหาการหาค่าต่ำสุดแบบมีเงื่อนไขอสมการ (2.25) เรียกว่าเงื่อนไข KKT [3,4] (Karush-Kuhn-Tucker conditions) ประกอบด้วย 4 สมการดังนี้

$$\nabla f(x) - \left(\frac{\partial c}{\partial x}\right)^T \lambda = 0 \tag{S}$$

$$c(x) \geq 0 \tag{PF}$$

$$\lambda \geq 0 \tag{DF}$$

$$\lambda \odot c(x) = 0 \tag{CS}$$

- S : Stationarity (ความคงที่)
- PF : Primal feasibility (ความเป็นไปได้ของตัวแปรไพรมัล)
- DF : Dual feasibility (ความเป็นไปได้ของตัวแปรดูอัล)
- CS : Complementary slackness (ความหย่อนเติมเต็ม)

2.2.4 วิธีเลือกเซตที่แอ็กทิฟ

วิธีเลือกเซตที่แอ็กทิฟ (active set method) กล่าวได้ว่าเป็นวิธีการที่ตรงไปตรงมาที่สุดในการหาคำตอบต่ำที่สุดบนเงื่อนไขบังคับอสมการคือเขียนโค้ดการวนซ้ำภายนอกเพื่อเลือกหรือคาดเดาว่าเงื่อนไขบังคับตัวใดแอ็กทิฟและเปิดสวิตช์ตัวแปรลากรานจ์สำหรับเงื่อนไขนั้น ทำให้กลายเป็นปัญหาเงื่อนไขบังคับสมการ ซึ่งเราทราบวิธีการหาคำตอบแล้วจากหัวข้อก่อนหน้านี้ จะเห็นว่าวิธีการนี้มีประสิทธิภาพมากหากสมมุติว่าเรามีเทพพยากรณ์ที่มีความแม่นยำสามารถบอกได้ว่าเงื่อนไขใดที่แอ็กทิฟ ซึ่งในหลายงานที่เกี่ยวข้องกับระบบควบคุมเช่น MPC สามารถคาดเดาได้อย่างค่อนข้างแม่นยำ

ผลการคาดเดาเงื่อนไขมี 3 กรณีดังนี้

1. การคาดเดาแม่นยำ ค่าที่ได้เป็นคำตอบที่ถูกต้อง
2. คาดเดาว่าเงื่อนไขแอ็กทิฟ แต่ความจริงไม่แอ็กทิฟ คือคาดว่าคำตอบอยู่บนเงื่อนไขสมการแต่ค่าจริงอยู่ภายใน ค่า λ จะได้เป็นค่าลบ ซึ่งละเมิด (DF) การที่ได้ $\lambda < 0$ เสมือนว่าต้องการดึงคำตอบจริงเข้าหาเส้นขอบเขต คือคำตอบที่เราคาดเดา
3. คาดเดาว่าเงื่อนไขไม่แอ็กทิฟ แต่ความจริงแอ็กทิฟ คือคาดว่าคำตอบอยู่ภายใน $c(x) \geq 0$ แต่ค่าจริงอยู่บนเงื่อนไขสมการ $c(x) = 0$ จะได้ $c(x)$ เป็นค่าลบ ซึ่งละเมิด (PF)

ผลการละเมิดในข้อ 2. และ 3. ช่วยให้เราทราบว่าการคาดเดาไม่ถูกต้อง

อย่างไรก็ตาม หากเงื่อนไขบังคับมีมากและเราไม่มีตัวช่วยในการคาดเดาได้แม่นยำ แต่ใช้วิธีการวนซ้ำทั้งหมดเพื่อตรวจสอบว่าเงื่อนไขใดแอ็กทิฟ วิธีการนี้จะเสียเวลามากจนไม่สามารถใช้งานได้ดี ดังนั้นสรุปได้ว่าไม่ใช่วิธีการที่เหมาะสมสำหรับกรณีทั่วไป

2.2.5 วิธีการลงโทษ

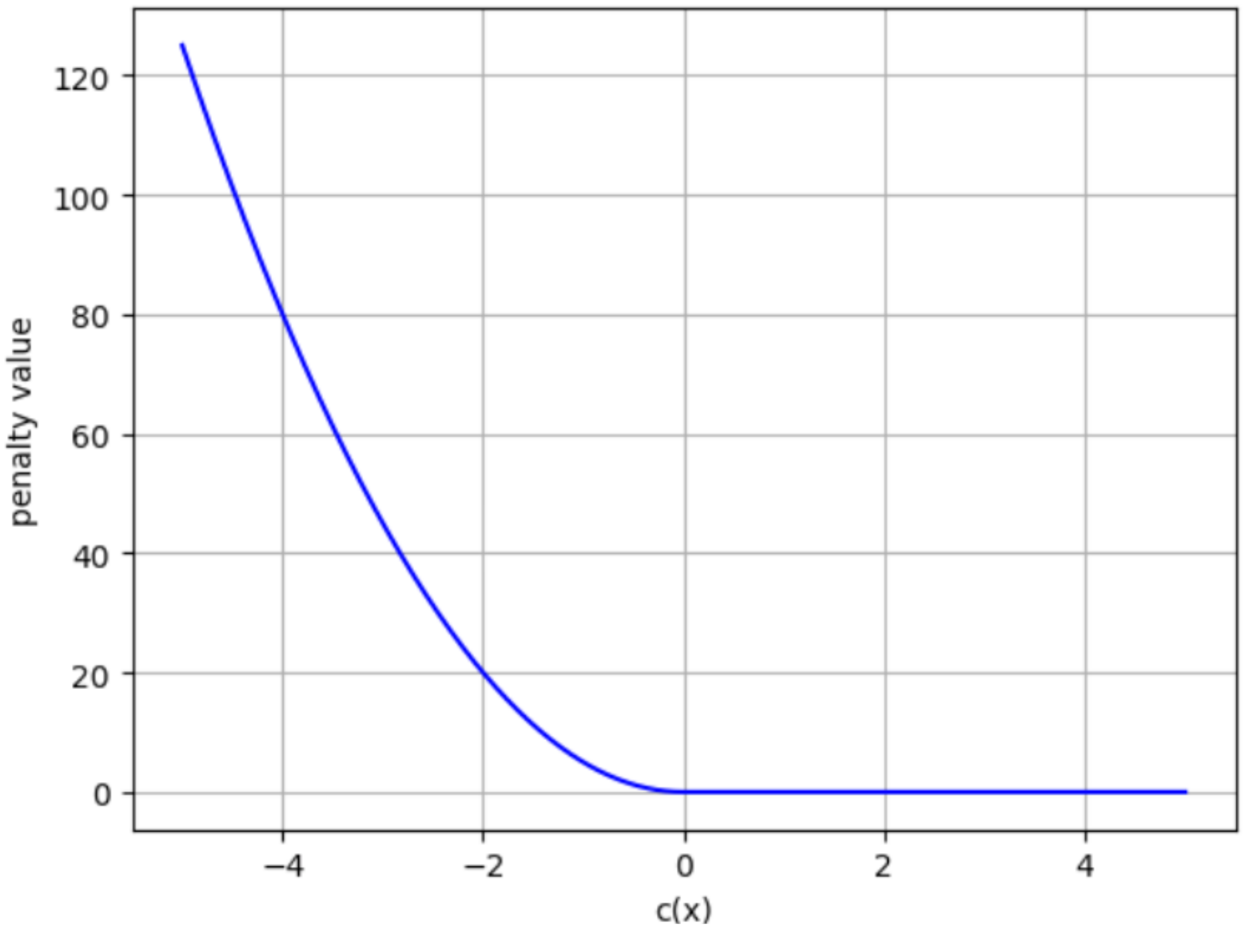
หลักการของวิธีการลงโทษ (penalty method) คือเปลี่ยนเงื่อนไขบังคับเป็นพจน์มูลค่าที่ให้โทษเมื่อเกิดการละเมิด

$$\min_{x} \; f(x) + \frac{\rho}{2} \left[min(0, c(x))\right]^2 \tag{2.26}$$

โดยพจน์ที่เพิ่มเข้ามาจะเป็นศูนย์เมื่อ $c(x) \geq 0$ แต่จะเพิ่มค่าของฟังก์ชันมูลค่าเมื่อ x ละเมิดเงื่อนไขบังคับ

หมายเหตุ : เพื่อความชัดเจนในการอธิบายเราพิจารณาเพียงเงื่อนไขบังคับเดียวใน (2.26) หากมีหลายเงื่อนไขบังคับ พจน์ลงโทษจะอยู่ในรูปผลรวมของแต่ละเงื่อนไข

กราฟของพจน์ให้โทษเทียบกับ $c(x)$ เมื่อเลือก $\rho = 10$ มีลักษณะดังรูปที่ 2.16



รูปที่ 2.16 กราฟของพจน์ให้โทษ $\frac{\rho}{2}\left[min(0,c(x))\right]^2$

2.2.6 วิธีลากรานเจียนแต่งเติม

* วิธีนี้บรรยายใน CMU16-745 ปี ค.ศ. 2024

จากปัญหาที่เกิดขึ้นกับวิธีการลงโทษ ทางแก้หนึ่งที่นิยมกันเรียกว่า *วิธีลากรานเจียนแต่งเติม (augmented Lagrangian)* ต่อไปจะเรียกย่อว่าวิธี AL หลักการคือประมาณค่า λ จากพจน์ให้โทษในแต่ละรอบการวนซ้ำ วิธีการที่ใกล้เคียงกันมีชื่อเรียกว่า ADMM (Alternating Direction Method of Multipliers)

เริ่มต้นจากการนิยามฟังก์ชันลากรานเจียนแต่งเติม

$$L_\rho(x, \tilde{\lambda}) \triangleq f(x) - \tilde{\lambda}^T c(x) + \frac{\rho}{2}\left[min(0, c(x))\right]^2 \tag{2.27}$$

และใช้เป็นฟังก์ชันข้อมูลรวมที่ต้องการลดค่า

$$\underset{x}{min} \;\; L_\rho(x, \tilde{\lambda}) \tag{2.28}$$

เมื่อใช้วิธีการนิวตัน จะได้ความสัมพันธ์

$$\frac{\partial f(x)}{\partial x} - \tilde{\lambda}\frac{\partial c(x)}{\partial x} + \rho c(x)^T \frac{\partial c(x)}{\partial x} = \frac{\partial f(x)}{\partial x} + (-\tilde{\lambda} + \rho c(x))^T \frac{\partial c(x)}{\partial x} = 0 \tag{2.29}$$

จาก (2.29) จะเห็นได้ชัดเจนว่า $\tilde{\lambda}$ และ $\rho c(x)$ ทำหน้าที่ในการให้โทษกับฟังก์ชันมูลค่าหมือนกัน ดังนั้นในแต่ละรอบการวนซ้ำ เราสามารถถ่ายค่าน้ำหนักการให้โทษมายัง $\tilde{\lambda}$

$$\tilde{\lambda} \leftarrow \tilde{\lambda} - \rho c(x) \tag{2.30}$$

โดยการถ่ายค่านี้จะกระทำเฉพาะเงื่อนไขบังคับที่แอ็กทิฟ (ถูกละเมิด) เนื่องจากพจน์ให้โทษเป็นศูนย์หาก (PF) ในเงื่อนไข KKT เป็นจริง

หมายเหตุ : สังเกตเครื่องหมายในการถ่ายค่าจะตรงข้ามกับ (2.29) ทั้งนี้เพื่อสอดคล้องกับ (PF) และ (DF) ในเงื่อนไข KKT กล่าวคือ เมื่อมีการละเมิดเงื่อนไข ค่า $\rho c(x)$ จะเป็นค่าลบ แต่ (DF) บังคับว่า $\lambda \geq 0$

ดังนั้นเค้าโครงของอัลกอริทึม AL จะเป็นดังนี้

- กำหนดค่า $\tilde{\lambda} = 0$ และค่า ρ น้อยๆ เช่น 1.0
- วนซ้ำขั้นตอน 1 - 3 ด้านล่างนี้จนกว่าจะลู่เข้า
  - 1. หาค่าต่ำสุดตามสมการ (2.28) (ขั้นตอนนี้จะมีการวนซ้ำภายในจนกว่าจะลู่เข้าสู่คำตอบ)
  - 2. ปรับค่า $\tilde{\lambda} \leftarrow max(0, \tilde{\lambda} - \rho c(x))$ โดยมีการจำกัดค่าเป็นเฉพาะค่าบวกตามเงื่อนไข (DF)
  - 3. เพิ่มค่า $\rho \leftarrow \alpha\rho$ โดยค่าที่นิยมใช้คือ $\alpha \approx 10$

ข้อดีของวิธีการ AL สรุปได้ดังนี้

- แก้ไขปัญหาเชิงเลขในวิธีการลงโทษ โดยอัลกอริทึม AL จะลู่เข้าได้โดยไม่ทำให้ ρ มีค่าสูงมากเข้าสู่อนันต์
- มีข้อจำกัดน้อย ใช้งานได้ดีกับปัญหาที่ไม่เป็นแบบคอนเวกซ์

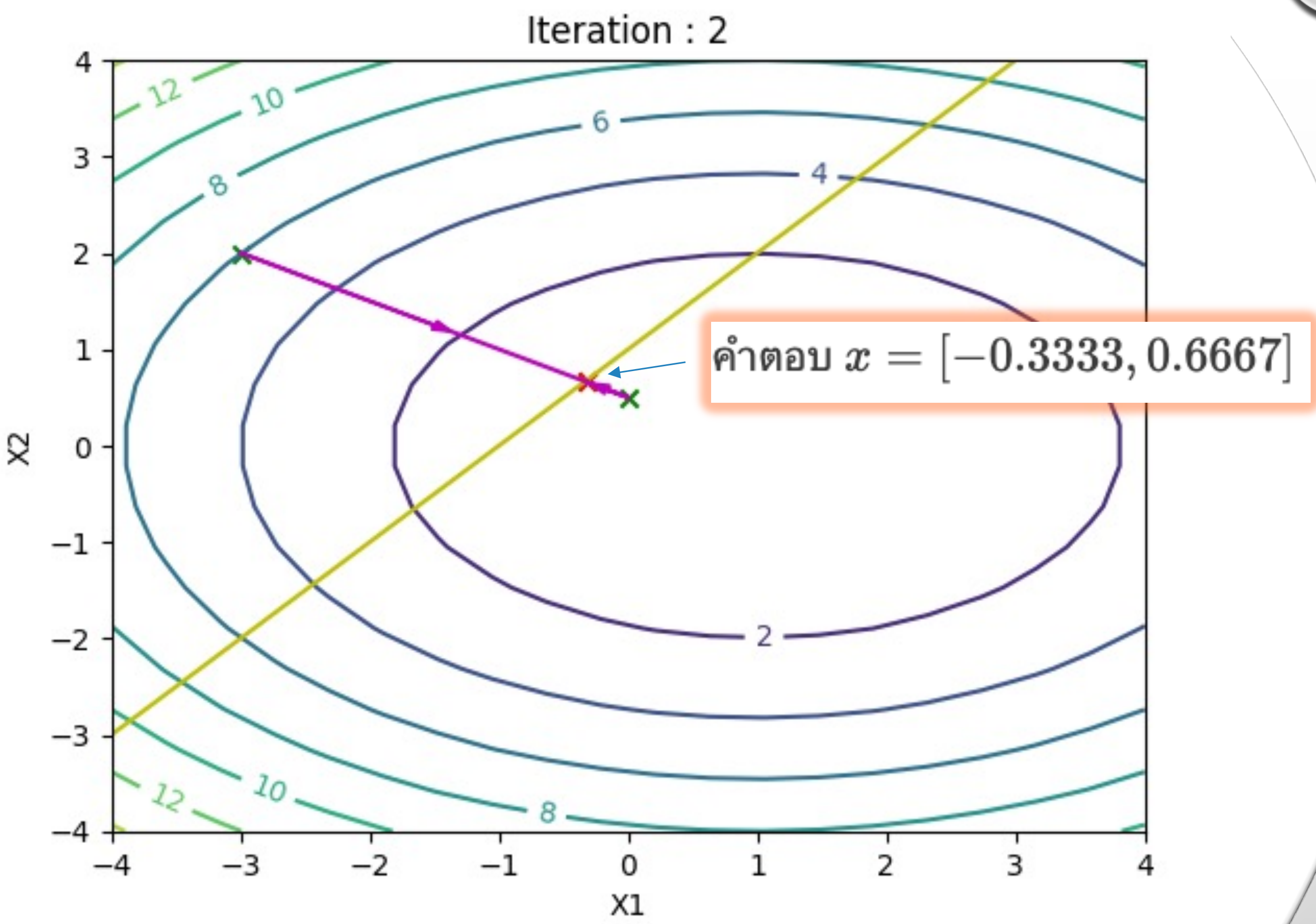
ตัวอย่าง 2.7 ในตัวอย่างนี้จะทดลองใช้วิธี AL หาคำตอบของปัญหาการโปรแกรมกำลังสอง (Quadratic Program) ต่อไปจะเรียกโดยย่อว่า QP เมื่อพิจารณาเฉพาะเงื่อนไขบังคับแบบอสมการ ปัญหา QP แบบคอนเวกซ์มีรูปแบบดังนี้

$$\min_{x} \ \frac{1}{2}x^T Q x + q^T x, \ \ Q \succ 0$$

$$s.t. \ \ Ax \geq b \tag{2.31}$$

QP เป็นปัญหาที่มีประโยชน์ในระบบควบคุม โดยเฉพาะในกรณีที่เป็นแบบคอนเวกซ์ จะสามารถหาคำตอบได้เร็ว (อัตรา kHz) ในตัวอย่างนี้จะสร้างโจทย์ QP อย่างง่าย สามารถเขียนโค้ดอนุพันธ์ของฟังก์ชันมูลค่าและเงื่อนไขโดยไม่ต้องใช้การหาอนุพันธ์อัตโนมัติ ดังนั้นจึงไม่จำเป็นต้องใช้แพ็กเกจเช่น Drake หรือ JAX

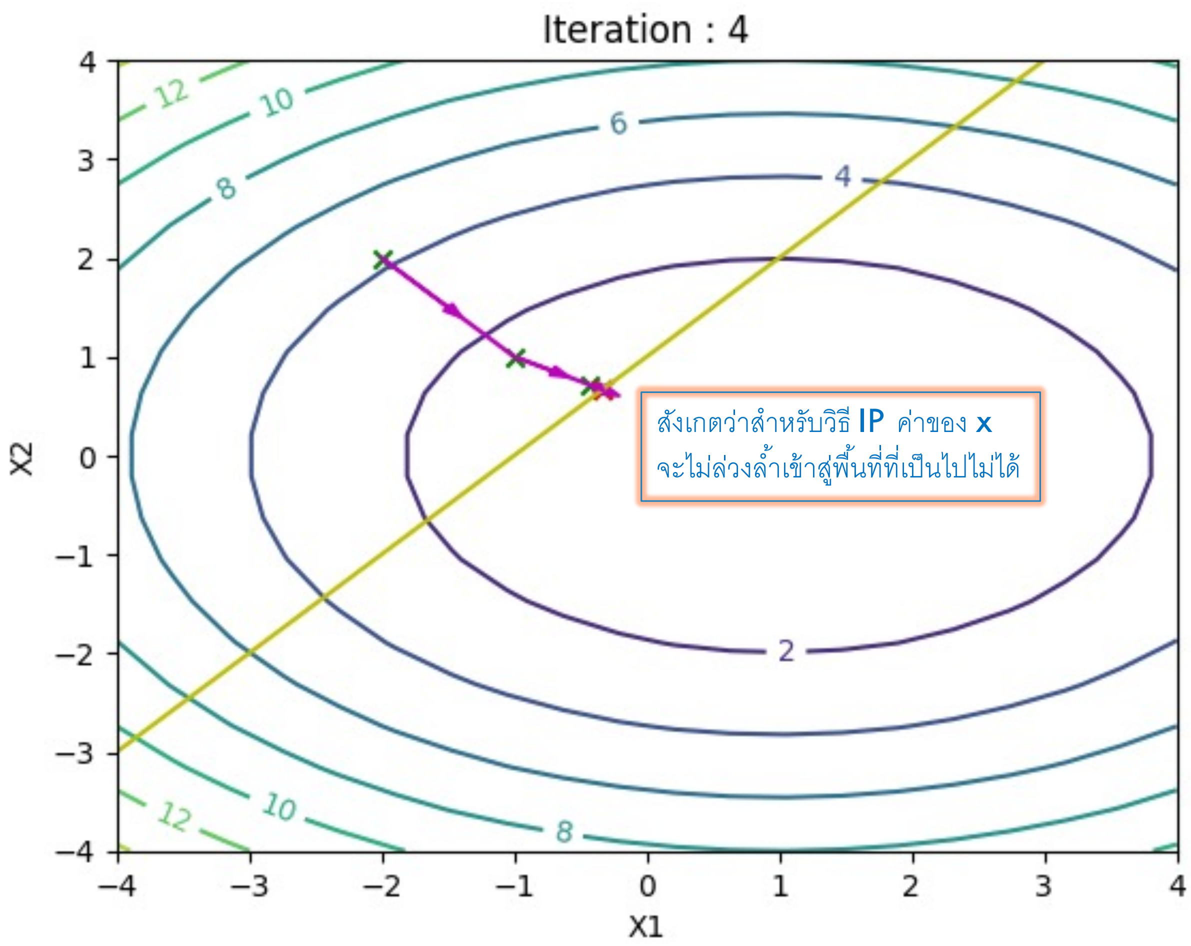
- รันโค้ดได้จาก chapter2.ipynb



รูปที่ 2.19 การลู่เข้าสู่คำตอบของปัญหา QP ในตัวอย่าง 2.7

วิธีจุดภายใน (Interior-point method)

- ใช้งานได้ดีกับปัญหาคอนเวกซ์ กำหนดขอบเขตเวลาการลู่เข้าได้ ทำให้เหมาะสมกับการทำงานแบบเรียลไทม์
- ใช้ในตัวหาคำตอบชื่อ IPOPT (Interior Point OPTmizer)
- เนื่องจากมีรายละเอียดค่อนข้างมาก ศึกษาเพิ่มเติมได้จาก chapter2.ipynb



รูปที่ 2.23 การลู่เข้าสู่คำตอบของวิธี IP

ปรับปรุงการหาค่าเหมาะที่สุดแบบมีเงื่อนไข

เรกูลาร์ไรเซชัน

การค้นหาตามเส้น

- ศึกษารายละเอียดได้จาก chapter2.ipynb

บรรณานุกรม

1. Z. Manchester et.al. 16-745 Optimal Control & Reinforcement Learning, Course materials, Carnegie Mellon University. 2024,2025.

2. L. Armijo. Minimization of funcionts having Libschitz coninuous first partial derivative. Pacific J. Math. 16(1):1-3. 1966.

3. W. Karush. Minima of Functions of Several Variables with Inequalities as Side Constraints. (M.Sc. thesis). Dept. of Mathematics, Univ. of Chicago, Chicago, Illinois. 1939.

4. H.W. Kuhn and A.W. Tucker. Nonlinear Programming. Proceedings of 2nd Berkeley Symposium. Berkeley. Univ. of California Press. pp. 481-492. 1951.

5. A. Wächter and L. T. Biegler. On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming, Mathematical Programming 106(1), pp. 25-57, 2006.

6. F. Permenter. Log-domain interior-point methods for convex quadratic progrmming. Optimization Letter 17 pp. 1613-1631. 2023.