CS381: Numerical Computation & Softwares

Root finding

Jakramate Bootkrajang

Department of Computer Science

Chiang Mai University

Introduction

- ในบทนี้เราจะศึกษาการหารากของสมการ
- สมการอาจเป็น
  - เชิงเส้น เช่น $f(x) = ax + b$
  - ไม่เชิงเส้น เช่น $f(x) = ax^2 + bx + c$
- คำว่ารากของสมการ (roots) หรือคำตอบของสมการก็คือ ค่า x^* ที่ทำให้ $f(x^*) = 0$

Introduction

- หากสมการของเราเป็นสมการกำลังสอง $f(x) = ax^2 + bx + c$
- เราทราบว่ามีสูตรลัดในการหารากทั้งหมดของสมการดังกล่าว

$$x^*_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- กรณีนี้โชคดีว่าเราสามารถหา closed form formula ในการหาคำตอบได้
- ซึ่งสูตรลัดดังกล่าวมีไม่ครบทุกๆสมการ (อาจมีสำหรับกำลัง 3 และ 4) แต่หากกำลังมากกว่า 5 ก็จะไม่มีสูตรลัดมาช่วย
- ดังนั้นเราจึงต้องใช้วิธีทาง numerical มาช่วยประมาณค่ารากของสมการ

Why do we care ?

- เราอาจจะตั้งคำถามว่าทำไมเราจะต้องสนใจค่า x^* ที่ทำให้ $f(x) = 0$ ด้วย
- ก็เพราะว่าเราสามารถแปลงการหาค่าของฟังก์ชันให้ได้ค่าที่เราต้องการให้อยู่ในรูปแบบข้างต้น
- หากเราต้องการหา $g(x) = 4$ เราสามารถมองปัญหาข้างบนในรูปของ

$$
\begin{aligned}
g(x) - 4 &= 0 \\
f(x) &= 0
\end{aligned}
$$

- โดยที่เรานิยาม $f(x) = g(x) - 4$
- ในการทำ optimisation เราอาจต้องการหา critical point หรือจุดที่ $f'(x) = 0$

Problem statement

- กำหนดฟังก์ชัน $f(x)$ เราต้องการจะหาค่า x ที่ทำให้สมการ $f(x) = 0$ เป็นจริง
- สมการ $f(x) = 0$ อาจไม่มีคำตอบ นั่นคือไม่มีค่า x ที่ทำให้สมการเป็นจริง
- แต่นั่นไม่ใช่ประเด็นที่เราสนใจ เราสนใจแต่กรณีที่ สมการนั้นมีคำตอบ
- ในทางคณิตศาสตร์แล้วรากที่หาได้ อาจเป็นจำนวนจริงหรือจำนวนเชิงซ้อน แต่เราจะสนใจเฉพาะรากที่เป็นจำนวนจริง
- สมการบางสมการอาจมีรากมากกว่า 1 ตัว วิธีที่เราจะศึกษาจะไม่สามารถบอกได้ว่าจะมีรากทั้งหมดกี่ตัว แต่จะหาให้เจอ 1 ในนั้นเท่านั้น

Root finding algorithms

- อัลกอริทึมที่เราสนใจอยู่ในกลุ่มที่เรียกว่า iterative method
- คือการเริ่มจากการ “เดา” คำตอบเริ่มต้น แล้วค่อยๆปรับปรุงคำตอบไปเรื่อยๆจนกว่าจะได้ค่าที่พอใจ
- อัลกอริทึมที่เราจะศึกษาในบทนี้คือ
  - Bisection method
  - Newton’s method
  - Secant method

General guideline, where is the root?

- ก่อนจะเริ่มทำ เราจำเป็นจะต้องรู้คร่าวๆว่าเราจะหาคำตอบในช่วงไหน
- วิธีพื้นฐานคือการลองวาดกราฟของฟังก์ชัน แล้วดูช่วงที่น่าจะเจอคำตอบ (กราฟตัดกับแกน y)
  - การวาดกราฟอาจไม่ง่าย เมื่อมิติสูงขึ้นเรื่อยๆ
- วิธีที่สองอาจเป็นการหาจุด a, b ที่ซึ่ง $f(a) > 0$ และ $f(b) < 0$ ถ้าหาก $f(x)$ เป็น continuous function เราจะได้ว่ารากของสมการจะต้องอยู่ในช่วง (a, b)

Bisection method

Bisection

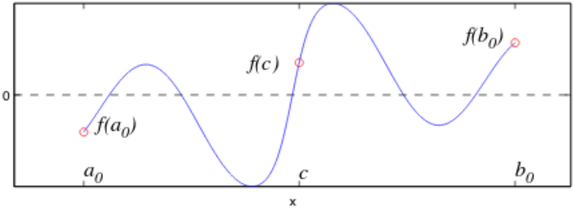
- เป็นวิธีที่เราน่าจะคิดถึงเป็นอันดับแรกในการเริ่มแก้ปัญหาการหารากของสมการ
- หลักการคือ เราพยายามจะหารากของฟังก์ชัน $f(x)$ ซึ่งเป็น continuous function บทช่วง $[a, b]$
- ที่ซึ่ง $f(a) > 0$ และ $f(b) < 0$
- เราทราบแน่นอนว่า จะต้องมีค่าหนึ่งในช่วง $[a, b]$ ที่ทำให้ $f() = 0$
- วิธี Bisection method จะเจอค่านี้ในที่สุด

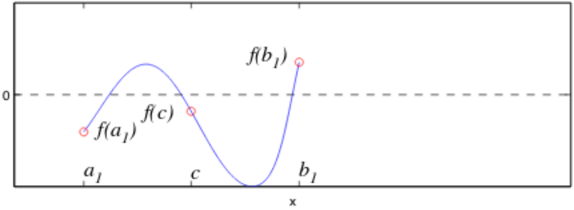
Bisection

- อันดับแรก เราแบ่งช่วง $[a, b]$ ออกเป็นสองช่วงที่จุด $c = \frac{a+b}{2}$
- ซึ่งจะได้ช่วงสองช่วงใหม่คือ $[a, c]$ และ $[c, b]$
- เราจะเลือกเก็บช่วง $[a, c]$ ไว้ หาก $f(a)f(c) < 0$ ไม่เช่นนั้นก็เก็บ ช่วง $[c, b]$ ไว้
- แล้วทำไปเรื่อยๆจนกว่าจะได้ $f(c) \approx 0$

Bisection

สอง iteration แรกของ bisection method





Convergence of Bisection method

- กำหนดให้ r คือคำตอบของสมการ
- นิยาม $|r - c_n|$ เป็นความคลาดเคลื่อน (หรือ error) ของค่าประมาณคำตอบที่ iteration n
- เราจะได้ว่า rate of convergence ของ Bisection method เป็น linear

$$|r - c_n| \leq 2^{-(n+1)}(b_0 - a_0)$$

- นั่นคือ c_n ที่ได้จาก Bisection method จะเข้าใกล้คำตอบด้วย rate เท่ากับ $\frac{1}{2}$

Newton method

Newton’s method

- Newton’s method เป็นวิธีในการหารากของสมการที่ไม่ซับซ้อน ใช้งานได้ดี และถูกนำไปใช้บ่อยในการหารากของสมการ
- วิธีนี้โดยส่วนใหญ่แล้วจะ converge เข้าหาคำตอบ แต่อาจจะไม่ทุกกรณี
- ซึ่งเราจะต้องทราบข้อจำกัดนี้ด้วย

The construction

- สมมติว่ามีฟังก์ชัน $f(x)$ ที่มีคำตอบเป็น r.
- เริ่มต้นเดาค่าของคำตอบด้วย x_0
- ให้ $l(x)$ แทนเส้นตรงที่สัมผัส $f(x)$ ที่จุด x_0 (tangent line) ซึ่งสามารถแสดงได้โดย

$$
\begin{aligned}
l(x) &= f'(x_0)(x - x_0) + f(x_0) \\
l(x) - f(x_0) &= f'(x_0)(x - x_0)
\end{aligned}
$$

The construction

- เราจะใช้จุดที่ $l(x)$ ตัดกับ x-axis เป็นค่าประมาณของ r ในรอบถัดไป
- หากให้ชื่อจุดตัดนั้นว่า x_1, เราจะได้ว่า

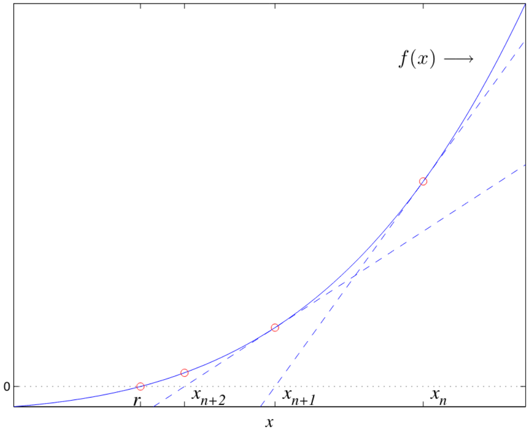
$$
\begin{aligned}
0 - f(x_0) &= f'(x_0)(x_1 - x_0) \\
x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}
\end{aligned}
$$

- จากสมการข้างต้นทำให้เราได้สมการในการปรับปรุงค่าประมาณคำตอบของ Newton's method (Newton-Raphson method) เป็น

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Newton iterations

สอง iteration แรกของ Newton method



Newton does not always converge

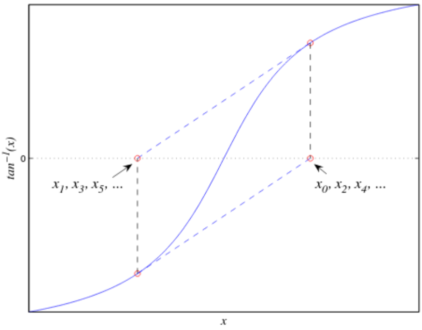


Figure 2.4: Newton's method does not always converge. In this case, the starting point is a fixed point of Newton's method iterated twice

หมายเหตุ Fixed point คือจุดที่ $x = f(x)$

Convergence of Bisection method

- กำหนดให้ e_n คือค่า error ใน iteration ที่ n
- ทฤษฎีกล่าวไว้ว่า rate of convergence ของ Newton's method เป็น quadratic

$$e_{n+1} \approx c e_n^2$$

Secant method

- เราทราบแล้วว่า Newton's method มีหลักในการปรับปรุงค่าประมาณของคำตอบโดย

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- ถ้าหากในกรณีที่เราไม่สามารถหาอนุพันธ์ของ f ได้ (อาจหาไม่ได้ หรือซับซ้อนเกินไป) เราก็จะไม่สามารถใช้ Newton's method ได้

- แต่เราอาจแก้ปัญหาโดยประมาณค่า $f'(x)$ แทน โดยวิธี

$$f'(x) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

The secant method

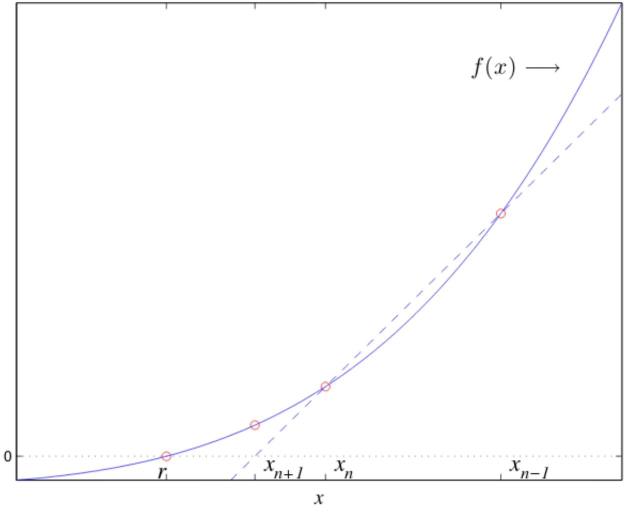
- ทำให้เราได้วิธีที่เรียกว่า Secant method

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

- สังเกตว่าหากใช้วิธี Secant method เราจะต้องเดาค่าเริ่มต้นสองจุด

Secant iterations

x_{n+1} ถูกประมาณโดยใช้ x_n และ x_{n-1}



Convergence of Secant method

- กำหนดให้ความคลาดเคลื่อนของคำตอบ ที่ n-th iteration เป็น

$$e_n = x_n - r$$

- ทฤษฎีกล่าวไว้ว่า rate of convergence ของ secant method เป็น **superlinear** (เร็วกว่า linear แต่ช้ากว่า quadratic), นั่นคือ

$$e_{n+1} \approx |e_n|^{\alpha} \quad \text{โดยที่} \quad \alpha = \frac{1+\sqrt{5}}{2}$$

References

- Introduction to numerical analysis by Doron Levy (บทที่ 2)