บทที่ 2
การวิเคราะห์เชิงตัวเลขสำหรับการหาราก และการแก้สมการ

หน้า

บทที่ 2
การวิเคราะห์เชิงตัวเลขสำหรับการหาราก และการแก้สมการ

การหารากของ $f(x)$ นั่นคือการ solve สมการ $f(x) = 0$

2.1 วิธีการหารากของ $f(x)$

Slope Methods: 1) วิธีของนิวตัน-ราฟสัน (Newton-Raphson(NR) Method)
2) วิธีเซแคนต์ (Secant(SEC) Method)

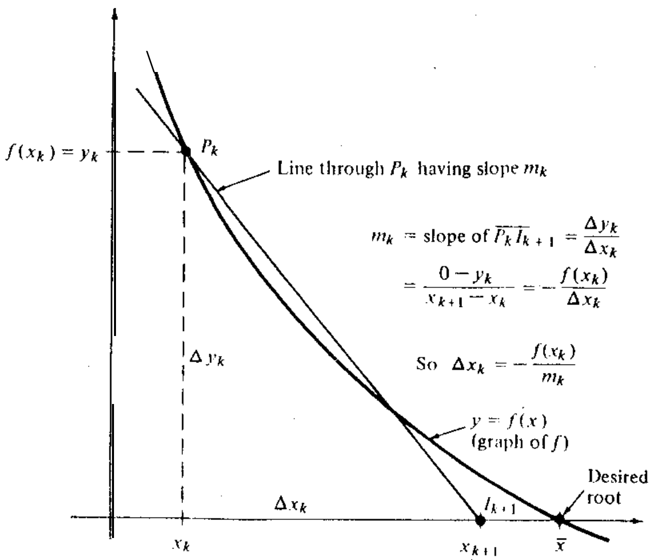
2.1A Slope Hethod Strategy

สมมุติว่า x_k เป็น current approximation ของรากที่ต้องการคือ $\bar{x}$

ให้ P_k เป็นจุด (x_k, y_k) โดยที่ $y_k = f(x_k)$

m_k เป็น non-zero number ที่แทน slope ของ curve $y = f(x)$ near the point P_k

x_{k+1} เป็น x-intercept ของเส้นตรงผ่านจุด P_k ซึ่งมี slope m_k



รูป 2.1-1 รูปแสดง Slope Method

จากรูปแสดงว่า $x_{k+1} = x_k + \Delta x_k$ ...(1)

โดยที่ $\Delta x_k = -y_k/m_k = -f(x_k)/m_k$ ซึ่งเป็นตัวกำหนดขั้นตอนวิธีของการทำซ้ำ

$\Delta x_k = x_{k+1} - x_k \doteq$ ความคลาดเคลื่อนของ x_k ...(2)

Algorithm: Slope Method (Special Cases: NR and SEC)

Purpose: To find a root $\bar{x}$ of $f(x)$, that is, to solve $f(x) = 0$ for $\bar{x}$.

{ *initialize*}
GET *MaxIt*, (maximum number of iterations]
NumSig, (desired number of accurate significant digits}
x_0 {an initial guess of $\bar{x}$}
$Xprev \leftarrow x_0$; $Yprev \leftarrow f(x_0)$; $RelTol \leftarrow 10^{-NumSig}$

{ *iterate*)
DO FOR $k = 1$ TO *MaxIt* **UNTIL termination test** is satisfied
BEGIN
{form Slope) *Slope* ← (formula involving *Xprev* and *Yprev*)
DeltaX ← − *Yprev/Slope*
X ← *Xprev* + *DeltaX*; *Y* ← *f(X)*
OUTPUT (*k, X, Y, DeltaX*)
(update) *Xprev* ← *X*; *Yprev* ← *Y*
[termination test: $|DeltaX| \le RelTol * |X|$ *or* $Yprev = 0$}
END

IF termination test succeeded
THEN OUTPUT (X approximates $\bar{x}$ to *NumSig* significant digits.)
ELSE OUTPUT (*MaxIt* iterations failed to yield a suitable *X*.)

รูป 2.1-2 รหัสเทียมสำหรับ **Slope Method**

2.1B วิธีของนิวตัน-ราฟสัน (Newton-Raphson Method): $m_k = m_{tan}$

ถ้า $f'(X)$ หาค่าได้

m_k = tangent slope ที่จุด $P_k(x_k, f(x_k))$

จากวิชาแคลคูลัส

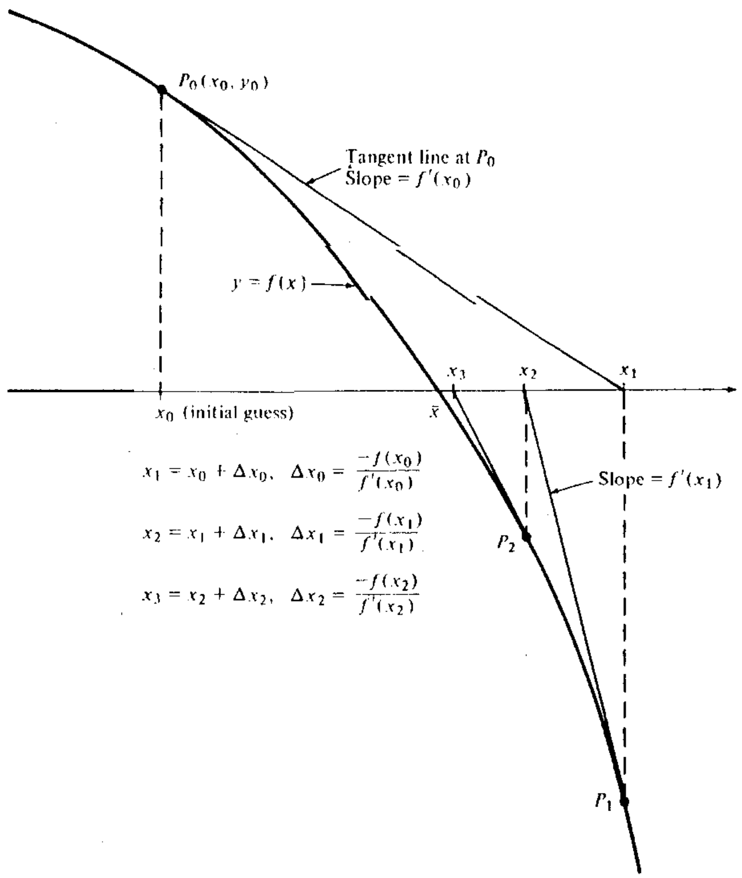
$$m_{tan} = f'(x_k) = \text{อนุพันธ์ของ } f \text{ ที่ } x_k$$

จาก (1) ใช้

$$m_k = m_{tan}$$

$$\boxed{\begin{aligned}(NR)\quad x_{k+1} = x_k + \Delta x_k, \text{ โดยที่ } \Delta x_k &= -f(x_k)/m_{tan} \\ &= -f(x_k)/f'(x_k)\end{aligned}} \quad \ldots(3)$$

สูตร (3) นี้คือขั้นของการทำซ้ำของวิธีของนิวตัน-ราฟสัน (หรือเรียกว่า วิธีของนิวตัน ก็ได้) สำหรับหารากของ f(x) หรือจากวิธีที่แสดงได้ตามรูปต่อไปนี้ เราอาจเรียก วิธีของนิวตัน-ราฟสัน ว่า **Method of Tangent**



รูป 2.1-3 รูปแสดงวิธีของนิวตัน-ราฟสัน

ตัวอย่าง

a. จงแสดงการใช้วิธีของนิวตัน-ราฟสัน เพื่อหา $\sqrt[n]{c}$ สำหรับ c > 0 และ n > 0

b. จงใช้วิธีของนิวตัน-ราฟสัน เพื่อหา $\sqrt{78.8}$ to 6s.

(หมายเหตุ: $\sqrt{78.8} \doteq 8.8769364$)

Solution

a. สำหรับ c ใดๆที่มากกว่า 0, $\sqrt[n]{c}$ คือรากบวกซึ่งมีเพียงรูปเดียวของ

$$f(x) = x^n - c \qquad \ldots(4)$$

(จาก 2.1A) สำหรับ f(x) ใน (4) $f'(x) = nx^{n-1}$

ดังนั้นโดยวิธีของนิวตัน-ราฟสัน

$$x_{k+1} = x_k + \Delta x_k \text{ โดยที่ } \Delta x_k = -(x_k^n - c)/nx_k^{n-1} \qquad \ldots(5)$$

b. แทน n = 2 ใน (5)

$$x_{k+1} = x_k + \Delta x_k \text{ โดยที่ } \Delta x_k = (c - x_k^2)/2x_k \qquad \ldots(6)$$

จาก (6) ใช้ c = 78.8 และ ค่าเดาเริ่มต้นที่ไม่ดี $x_0 = 14$

k = 0: $\Delta x_0 = (78.8-14^2)/2(14) \doteq -4.185714$;

ดังนั้น $x_1 = 14 - 4.185714 = 9.814286$

k = 1: $\Delta x_1 = (78.8-x_1^2)/2x_1 \doteq -0.892587$;

ดังนั้น $x_2 = 9.814286 - 0.892587 = 8.921699$

k = 2: $\Delta x_2 = (78.8-x_2^2)/2x_2 \doteq -0.044650$;

ดังนั้น $x_3 = 8.921699 - 0.044650 = 8.877049$

k = 3: $\Delta x_3 = (78.8-x_3^2)/2x_3 \doteq -0.000113$;

ดังนั้น $x_4 = 8.877049 - 0.000113 = 8.876936$

เพราะว่า $|\Delta x_3| \doteq 0.0001$ เราสรุปว่า x_3 มีความแม่นยำประมาณ 4d(นั่นคือ 5s) จากการพิจารณาอัตราการลดลงอย่างรวดเร็วของ $|\Delta x_k|$ เข้าหาศูนย์

x_4 ควรจะมีความแม่นยำถึง 6s จริงๆแล้วมีความแม่นยำถึง 7s ตามที่ได้แสดงไว้

ถ้าเราแทน Δx_k ใน (6) เราจะได้สูตรสำหรับการทำซ้ำดังนี้

$$x_{k+1} = x_k+(78.8-x_k^2)/2x_k = (x_k^2+78.8)/2x_k = g(x_k) \qquad ...(7)$$

ขั้นตอนวิธีของวิธีของนิวตัน-ราฟสันลู่เข้าแบบอันดับสองเข้าสู่ $\bar{x} = \sqrt{78.8}$ เมื่อ $f(x) = x^2 - 78.8$

โดยทั่วๆไปวิธีการทำซ้ำโดยใช้วิธีของนิวตัน-ราฟสันนั้นมักจะทำให้เกิดการลู่เข้าแบบอันดับสอง นั่นคือจะทำให้

$$(x_{k+1} - x_k) \approx C(x_k - x_{k-1})^2, \quad C = \text{ค่าคงที่} \qquad ...(8$$

เมื่อรากที่ต้องการ ($\tilde{x}$) คือ รากเชิงเดียว (simple root) ของ f

2.1C การจัดรูปสมการใหม่เพื่อทำให้วิธีของนิวตัน-ราฟสัน ง่ายต่อการนำไปใช้

สมมุติเราต้องการรากบวกสองรากที่เล็กที่สุดของ

$$f(x) = e^{-x} \sec x - 1 \qquad . \ (9)$$

โดยให้มีความแม่นยำถึง 7s

$$f(x) = 0$$

$$e^{-x} \sec x - 1 = 0 \iff e^{-x} = \cos x$$

$$\iff e^{-x} - \cos x = 0$$

นั่นคือรากของ f(x) ที่กำหนดให้ใน (9) จะเหมือนรากของ

$$f(x) = e^{-x} - \cos x \qquad ...(10)$$

จาก (10) เราอาจหา $f'(x)$ ได้ง่ายกว่าจาก (9)

$\Delta x_k = -f(x_k)/f'(x_k)$ โดยที่ $f'(x_k) = \sin x_k - e^{-x_k}$

เราจะใช้ $\bar{x}_1 \approx 1.3$ และ $\bar{x}_2 \approx 3\pi/2$ เป็นค่าเดาเริ่มต้นสองค่า สำหรับหารากที่ต้องการทั้งสอง ทั้งนี้โดยพิจารณาจากตัวอย่างของเทคนิคต่อไปนี้

<u>**ตัวอย่าง**</u> จงหาค่าเดาเริ่มต้น (initial guesses) สำหรับขั้นตอนวิธีของการทำซ้ำ

ในการหารากของฟังก์ชัน

และจงประมาณรากทั้งหมดของ $f(x) = e^{-x} \sec x - 1$

<u>**Solution**</u> เราจะเขียน $f(x) = 0$ ในรูป $g(x) = h(x)$

แล้ววาด curve ของ $g(x)$ และ $h(x)$ จุดตัดของ $g(x)$ และ $h(x)$

จะเป็นค่าประมาณของรากของ $f(x)$

สมการ $f(x) = 0$ จะมีรากเหมือนกับสมการต่อไปนี้

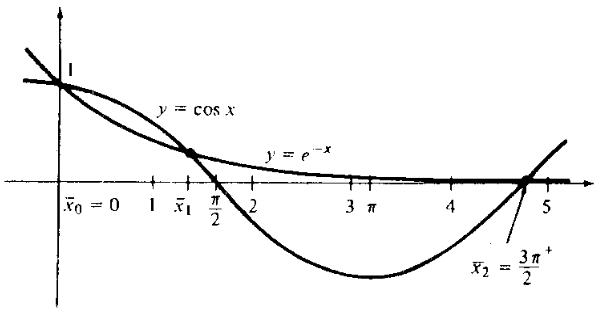
$e^{-x} \sec x = 1$ หรือ $e^{x} \cos x = 1$ หรือ $e^{x} = \sec x$ หรือ $e^{-x} = \cos x$

เราจะเลือก $g(x) = e^{-x}$ และ $h(x) = \cos x$ ซึ่งง่ายที่สุดในการวาด

รูป 2.1-4 แสดงว่า $f(x) = e^{-x} \sec x - 1$ มีรากมากมาย

รากที่เล็กที่สุดคือ $\bar{x}_0 = 0$ รากตัวถัดไปคือ $\bar{x}_1 \approx 1.3$ และตัวอื่น ๆ คือ

$\bar{x}_k \approx (2k-1)\pi/2$ สำหรับ $k = 2, 3, 4, \ldots$



รูป 2.1-4 รูปแสดงการประมาณค่ารากของ $e^{-x} \sec x - 1$

หมายเหตุ เราไม่จำเป็นต้องใช้ ค่าเดาเริ่มต้น ที่มีความแม่นยำเกิน 2s

จากการใช้ $\bar{x}_1 \approx 1.3$ และ $\bar{x}_2 \approx 3\pi/2$ เป็น ค่าเดาเริ่มต้นของวิธีของนิวตัน-ราฟสัน เมื่อทำซ้ำไปเพียง 2 ครั้งทั้งสองกรณี เราได้ค่า $\bar{x}_1$ และ $\bar{x}_2$ แม่นยำถึง 7s(จริง ๆ แล้วได้ถึง 8s) และคอลัมน์ Δx_k แสดงว่า

*the **number** of accurate **decimal places(roughly)** doubles **with** each iteration once x_k gets close to $\bar{x}$*

การลดลงอย่างรวดเร็วของ Δx_k เป็นลักษณะของการลู่เข้าแบบอันดับสอง เมื่อใช้ค่าเดาเริ่มต้นที่ดี

ตาราง 2.1-1* การทำซ้ำโดยวิธี NR สำหรับการหาราก 2 ราก ของ $f(x) = e^{-x} - \cos x$

k	x_k	$y_k = f(x_k)$	$\Delta x_k = \dfrac{-y_k}{f'(x_k)}$	$x_{k+1} = x_k + \Delta x_k$
0	1.3	0.50329644E−2	-0.0072833	1.2927167
1	1.2927167	0.14403021E-4	−0.0000210	1.2926957
0	4.7123890	0.89832910E−2	0.0089033	4.7212923
1	4.7212923	0.46291601E-b	0.0000005	4.7212928

ทั้งนี้ถ้าลองใช้ x_0 ตัวเดิม และทำกับ f(x) ใน (9) จะพบว่าทำได้ยากกว่าทำกับ f(x) ใน(10) การลองทำตามที่กล่าวจะแสดงให้เห็นว่า ไม่ว่าจะใช้วิธีการอย่างใด การเขียน f(x)=0 เสียใหม่ให้อยู่ในรูปที่ง่าย จะประหยัดพลังงานในการแก้ปัญหาได้อย่างมาก (และจะลดโอกาสของการเกิดความคลาดเคลื่อนจากการกระทำของคนลงมาก)

2.1D วิธีเซแคนต์ (SEC Method): $m_k = m_{sec}$

ในวิธีของนิวตัน-ราฟสัน m_k ใน general slope method formula

$$x_{k+1} = x_k + \Delta x_k \text{ โดยที่ } \Delta x_k = -f(x_k)/m_k$$

คือ exact slope ณ จุด $P_k(x_k, y_k)$

และต้องหา $f'(x)$ และคำนวณ $m_{tan} = f'(x_k)$ เพิ่มจาก $f(x_k)$ ในแต่ละครั้งของการทำซ้ำ

ในการหาค่าและแทนค่า $f'(x)$ เมื่อ $f(x) = x^n - c$ หรือ $f(x) = e^{-x} - \cos x$ นั้นยังไม่ยุ่งยาก แต่ปัญหาเช่น $f(x) = \sin\sqrt{\sec x + x^3 e^{5x/\tan x}} - e^{-x}$ นั้นจะเป็นงานที่ยุ่งยากมาก

สำหรับ $f(x)$ ที่ยุ่งยากลักษณะนี้ slope method ที่เราต้องการคือ วิธีที่จะทำให้หาค่า m_k จาก $f(x)$ อย่างเดียว(ไม่ใช่จาก $f'(x)$)เช่นหา m_k จาก current point $P_k(x_k, f(x_k))$ และ preceding current point $P_{k-1}(x_{k-1}, f(x_{k-1}))$ เพื่อหาค่า secant slope

$$m_{sec} = [f(x_k) - f(x_{k-1})]/(x_k - x_{k-1}) = (y_k - y_{k-1})/\Delta x_{k-1}$$

ใช้ $m_k = m_{sec}$ ในสูตรของ general slope method

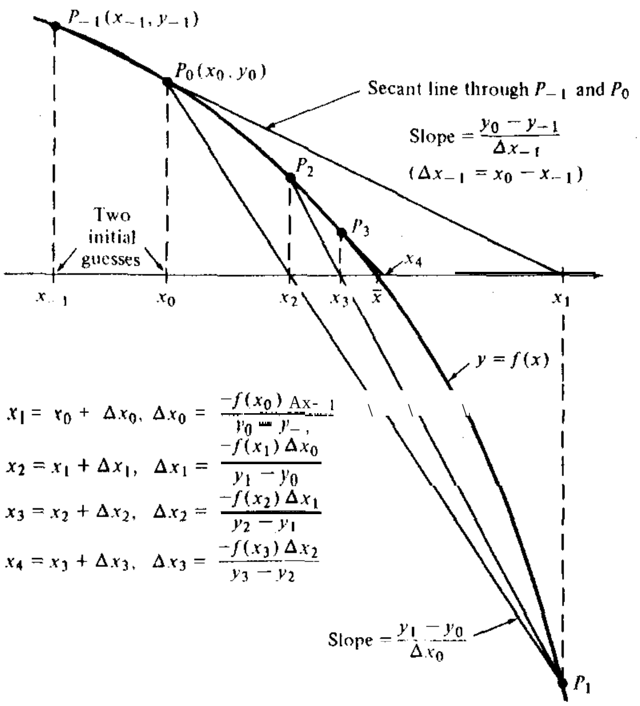
$$\boxed{\begin{aligned} \text{(SEC)}\quad x_{k+1} = x_k + \Delta x_k, \text{ โดยที่ } \Delta x_k &= -f(x_k)/m_{sec} \\ &= -y_k\, \Delta x_{k-1}/(y_k - y_{k-1}) \end{aligned}}$$

ดังนั้นวิธีเซแคนต์ต้องใช้ค่าเดาเริ่มต้น 2 ค่า

รูปต่อไปแสดงการหาค่า x_1, x_2, x_3, และ x_4 จาก ค่าเดาเริ่มต้น x_0 และ x_{-1} ถ้า x_{k-1} และ x_k ใกล้ $\bar{x}$ ทั้ง 2 ค่าแล้ว

m_{sec} (เส้นผ่านจุด P_{k-1} และ P_k) $\approx$ m_{tan} (ณ จุด P_k)

แสดงว่าวิธีเซแคนต์ จะลู่เข้าเร็วพอ ๆ กับวิธีของนิวตัน-ราฟสัน ดังจะแสดงในตัวอย่างถัดไป



รูป 2.1-5 รูปแสดงวิธีเซแคนต์

<u>**ตัวอย่าง**</u> จงใช้วิธีเซแคนต์ เพื่อหา $\sqrt{78.8}$ ให้ได้ 6s โดยใช้ $x_{-1}=14.1$ และ $x_0=14.0$ จากตัวอย่างที่ผ่านมา เราพบว่า $\bar{x} = \sqrt{78.8}$ คือรากบวกของ $f(x) = x^2 - 78.8$

จากวิธีเซแคนต์

$$\Delta x_{-1} = x_0 - x_{-1} = 14.0 - 14.1 = -0.1$$ เราได้

k=0: $\Delta x_0 = -y_0 \Delta x_{-1} / (y_0 - y_{-1})$

$= -117.20(-0.1)/(117.20-120.01) \doteq -4.170818$

ดังนั้น $x_1 = x_0 + \Delta x_0 = 14.0 - 4.170818 = 9.829182$

$k=1$: $\Delta x_1 = -y_1 \Delta x_0/(y_1 - y_0)$

$= -17.81282(-4.170818)/(17.81282-117.20)$

$\doteq -0.747521$

ดังนั้น $x_2 = x_1 + \Delta x_1 = 9.829182-0.747521 = 9.061661$

$k=2$: $\Delta x_1 = -y_2 \Delta x_1/(y_2 - y_1)$

$= -3.676567(-0.747521)/(3.676567-17.81282)$

$\doteq -0.194416$

ดังนั้น $x_3 = x_2 + \Delta x_2 = 9.081661-0.194416 = 6.687245$

และคำนวณต่อไปอีกดังแสดงในตารางถัดไป

ตาราง 2.1-2 การทำซ้ำโดยวิธี SEC สำหรับการหาราก 1 ราก ของ $f(x) = x^2 - 78.8$

k	x_k	$y_k = f(x_k)$	$\Delta x_k = \dfrac{-y_k \, \Delta x_{k-1}}{y_k - y_{k-1}}$	$x_{k+1} = x_k + \Delta x_k$
-1	14.1	120.01		
0	14.0	117.20	-4.170818	9.829182
1	9.829182	17.81282	-0.747521	9.081661
2	9.081661	3.676567	-0.194416	8.887245
3	8.887245	0.183124	-0.010191	8.877054
4	8.877054	0.002088	-0.000117	8.876937

เพราะว่า $\left|\Delta x_4\right| = 0.000117 \approx 10^{-4}$

ดังนั้น $x_4 = 8.877054$ จึงมีความแม่นยำประมาณ 4d (นั่นคือ 5s)

และเพราะว่า $\left|\Delta x_k\right|$ มีค่าเข้าใกล้ศูนย์อย่างรวดเร็ว $x_5 = 8.876937$ ควรจะมีความแม่นยำถึง 6s (ความจริง x_5 มีความแม่นยำถึง 6s จริง ๆ) เมื่อเปรียบเทียบกับการใช้วิธีของนิวตัน-ราฟสัน โดยใช้ x_0 ตัวเดียวกัน จะพบว่าวิธีเซแคนต์ จะใช้จำนวนครั้งของการทำซ้ำมากกว่าวิธีของนิวตัน-ราฟสัน อยู่ประมาณ 25% เพื่อให้ได้ความแม่นยำถึง 6s โดยทั่ว ๆ ไปแล้ววิธีเซแคนต์ จะลู่เข้าเร็วกว่าแบบเชิงเส้น แต่ไม่เร็วเท่ากับแบบอันดับสอง เมื่อ $\bar{x}$ เป็นรากเชิงเดียว

2.1E การหาค่าเดาเริ่มต้น (Initial Guesses) สำหรับวิธีเซแคนต์

จาก 2.1C ได้กล่าวถึงวิธีการหาค่าเดาเริ่มต้นมาแล้ว ส่วนในการหาค่าเดาเริ่มต้นตัวที่สองสำหรับวิธีเซแคนต์ นั้นหาได้โดยการทำให้ ค่าเดาเริ่มต้นตัวแรกเปลี่ยนค่าไปเล็กน้อย เช่น ค่าเดาเริ่มต้นตัวแรก $x_{-1} = 14.0$

ค่าเดาเริ่มต้นตัวที่สอง $x_o = 14.0+0.1 = 14.1$

ตัวอย่าง จงใช้วิธีเซแคนต์ เพื่อหารากบวกสองรากที่เล็กที่สุดของ(to 7s)

ของ $f(x) = e^{-x} - \cos x$

Solution จากรูปใน 2.1C เราจะเห็นว่า $\bar{x}_1 \approx 1.3$ และ $\bar{x}_2 \approx 3\pi/2$

สำหรับ $\bar{x}_1$ เราใช้ $x_{-1} = 1.2$ และ $x_o = 1.3$

ดังนั้น $f(x_{-1}) \doteq -0.06116$ และ

$f(x_o) \doteq 0.00503$

สำหรับ $\bar{x}_2$ เราใช้ $x_{-1} = 3\pi/2 + 0.1$ และ $x_o = 3\pi/2$

ดังนั้น

$f(x_{-1}) \doteq -0.09170$

$f(x_o) \doteq 0.008983$

ใช้การทำซ้ำโดยวิธีเซแคนต์ (จนได้ 7s accuracy) ได้ผลดังนี้

ตาราง 2.1-3 การทำซ้ำโดยวิธี SEC สำหรับการหาราก 2 ราก ของ $f(x) = e^{-x} - \cos x$

k	x_k	$y_k = f(x_k)$	$\Delta x_k = \dfrac{-y_k \Delta x_{k-1}}{y_k - y_{k-1}}$	$x_{k+1} = x_k + \Delta x_k$
−1	1.2	−0.6116354E−1		
0	1.3	0.5032964E−2	−0.0076031	1.2923969
1	1.2923969	−0.2052559E−3	0.0002980	1.2926949
2	1.2926949	−0.5943326E−6	0.0000008	1.2926957
−1	4.8123890	0.9170500E−1		
0	4.7123890	0.8983291E−2	0.0089219	4.7213109
1	4.7213109	−0.1826462E−4	−0.0000181	4.7212928

การลู่เข้าของวิธีเซแคนต์เกือบเร็วเท่ากับการลู่เข้าแบบอันดับสองของวิธีของนิวตัน-ราฟสัน

ในวิธีเซแคนต์ ถ้า x_k และ x_{k-1} อยู่ใกล้ $\bar{x}$ แล้ว

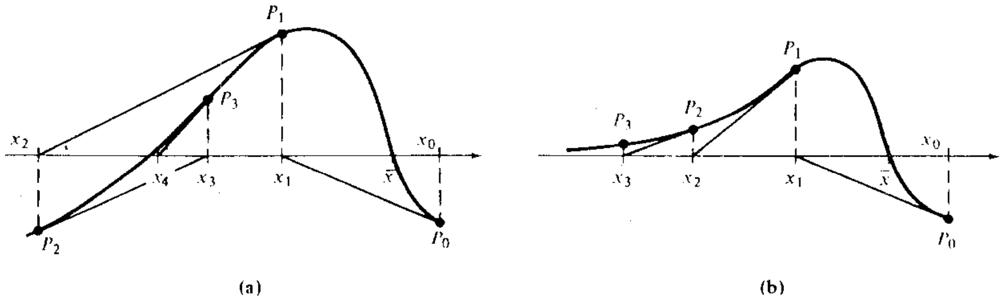
$$\left|x_{k+1} - x_k\right| \approx C\left|x_k - x_{k-1}\right|^{1.618}, \quad C = \text{Constant}$$

และเรียกการเกิดข้างต้นว่า **การลู่เข้าที่เกือบเป็นแบบอันดับสอง (almost quadratic convergence)**

2.1F <u>สิ่งที่อาจเกิดขึ้นกับวิธีของนิวตัน-ราฟสัน และ วิธีเซแคนต์</u>

ในบางเหตุการณ์ slope methods อาจจะลู่เข้าไปยังรากตัวที่เราไม่ตั้งใจหรืออาจจะไม่ลู่เข้าเลย

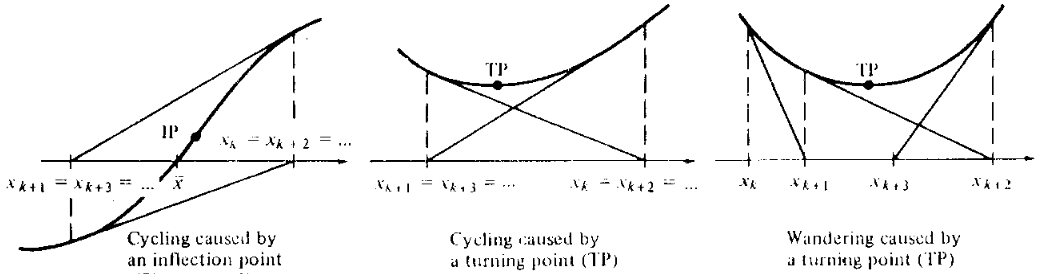
จากรูปต่อไปนี้แสดง ผลที่อาจเกิดการใช้ค่าเดาเริ่มต้นที่ไม่ดีสำหรับวิธีของนิวตัน-ราฟสัน



รูป 2.1-6 ผลที่อาจเกิดจากการใช้ค่าเดาเริ่มต้นที่ไม่ดีสำหรับวิธีของนิวตัน-ราฟสัน

(a) และ (b) เกิดขึ้นเพราะ **tangent line** ที่ $P_o(x_o, f(x_o))$ เกือบจะเป็นเส้นแนวราบ (horizontal) [i.e., $m_o = f'(x_o) \approx 0$] ผลที่ตามมาคือ x-intercept (x_1) จะอยู่ไกลจากทั้ง x_o และ $\bar{x}$ ที่ต้องการ

เหตุการณ์ดังกล่าวนี้เรียกว่า **Overshoot** และอาจเกิดเมื่อค่าเริ่มต้น m_{sec}-->0 เช่นกัน แต่อย่างไรก็ดี m_{sec} ใช้ P_{k-1} ร่วมกับ P_k ซึ่งจะทำให้ได้ x_e อยู่ใกล้ x_o ถึงแม้ว่า x_1 overshoots



Cycling caused by an inflection point (IP) near ($\bar{x}$, 0).

(a)

Cycling caused by a turning point (TP) near the x-axis.

(b)

Wandering caused by a turning point (TP) near the x-axis.

(c)

รูป 2.1-7 การเกิด Cycling และ Wandering เมื่อใช้วิธีของนิวตัน-ราฟสัน

ภาพ (b) และ (c) อาจเกิดกับวิธีเซแคนต์ เช่นกัน

จากภาพทั้งสองสรุปได้ว่า

1. การใช้ค่าเดาเริ่มต้นที่ดีหรือแม่นยำนั้นเป็นสิ่งสำคัญ เมื่อจะหารากที่อยู่ใกล้ turning point (TP)

2. การกำหนดขีดจำกัดบน (upper limit) ของการทำซ้ำไว้ (12-25 ครั้ง) เป็นสิ่งจำเป็นอย่างยิ่งในโปรแกรมเมื่อใช้ slope method ไม่ว่าจะเป็นวิธีใด

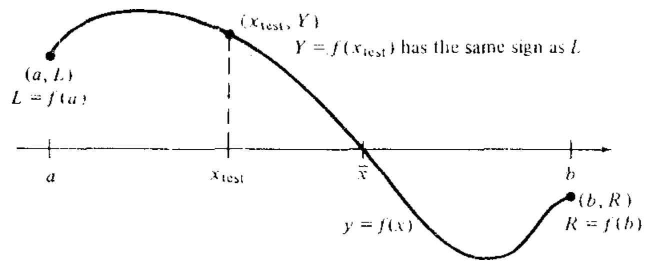
2.10 Bracketing Methods

วิธีนี้ถูกเรียกว่า Bracketing Methods เพราะ bracket $\bar{x}$ ไว้ระหว่างจุดปลาย (endpoints) ของ close interval [a,b]

Bracketing Methods ทุกวิธีจะเริ่มจากช่วงเริ่มต้น [a,b] โดยที่

$L = f(a)$ และ $R = f(b)$ ต้องมีเครื่องหมายตรงกันข้าม

ถ้าสมมุติว่า f เป็นฟังก์ชันต่อเนื่องแล้ว จะต้องมีอย่างน้อย 1 ราก $\bar{x}$ อยู่ระหว่าง a และ b



รูป 2.1-8 วิธี Bracketing

เพื่อทำให้คำอธิบายง่าย เราสมมุติว่าจะมีเพียง 1 รากเท่านั้นในช่วง [a,b] เราจะหาขั้นตอนวิธีเพื่อที่จะเคลื่อน a และ b เข้าหากันโดยที่ $\bar{x}$ จะต้องอยู่ระหว่าง a และ b เช่นเดิม

วิธีการคือ หา **a test value** x_{test} ใน open interval (a,b) แล้วแทน [a,b] ด้วย [a,x_{test}] หรือ [x_{test},b] ทั้งนี้แล้วแต่ว่า $\bar{x}$ จะอยู่ในช่วงใด การแทนค่านั้นทำได้ง่าย โดยการดูเครื่องหมายของ y = f(x_{test})

IF sign(Y)=sign(L) THEN replace (a,L) by (x_{test},Y)

ELSE {sign(Y)=sign(L)} replace (b,R) by **(x_{test},Y)**

จาก sign(Y) = sign(L) (ทั้งคู่เป็น +) ดังนั้นเราจะแทน (a,L) ด้วย (x_{test},Y)

ขบวนการทำซ้ำกับช่วง [a,b] ที่สั้นลงกว่าเดิม จนกว่า Y = 0 หรือ x_{test}**'s สองค่าที่อยู่ติดกัน มีค่าใกล้เคียงกันอย่างพอเพียง (sufficiently close)**

Algorithm: Bracketing Method [Special Cases: BIS and FP]

Purpose: To find a root $\bar{x}$ of f(x) in an interval $[a, b]$ for which $L = f(a)$ and R = f(b) have opposite sign.

{*initialize*}
GET *a*, 6. *MaxIt, NumSig*
L - f(o); R - f(b); RelTol ← $10^{-NumSig}$
IF sign(l) = sign(R) THEN STOP (method cannot be used)
Xprev ← *6* (this ensures that the initial *DeltaX* ≠ 0}

{*iterate:* Repeatedly form X (= x_{test}) in the *open* interval (a, b)}
DO FOR k = **1** TO *MaxIt* UNTIL **termination test** is satisfied
BEGIN
(form *X*} X ← (formula for x_{test} involving *a*, 6, *L, R)*
DeltaX ← *X* − *Xprev*; *Y - f(X)*
OUTPUT (*k, X, Y, DeltaX*) (i.e., *k*. x_k, y_k, Δx_{k-1}}
{**update**}
IF sign(Y) = sign(l)
THEN [move *a*} BEGIN *a* ← X; *L* ← *Y* **END**
ELSE [move *b*} BEGIN *b* ← *X*; *R* ← *Y* END
Xprev ← *X*
[termination test: |*DeltaX*| ≤ *RelTol**|*X*| or *Y* = 0}
END

IF **termination test** succeeded
THEN OUTPUT (X approximates $\bar{x}$ to *NumSig* significant digits.)
ELSE OUTPUT (*MaxIt* iterations failed to yield a suitable X.)

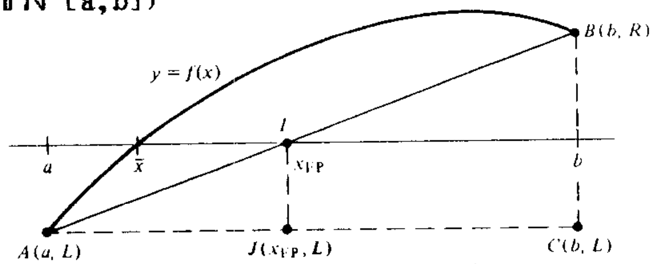
รูป 2.1-9 รหัสเทียมสำหรับ Bracketing Method

Bracketing methods ทุกวิธีมีลักษณะที่ร่วมกันคือ วิธีการจะใช้ได้ถ้า f(x) ที่อยู่คนละข้างของ $\bar{x}$ มีเครื่องหมายต่างกันเท่านั้น

2 วิธีที่นิยมคือ

$x_{test} = x_{lid} = (a+b)/2$	**(Bisection (BIS) Method)**
	หรือ Interval Halving Method
$x_{test} = x_{FP} = (bL-aR)/(L-R)$	**(False Position (FP) Method)**
	หรือ Regula Falsi Method

ใน FP method x_{test} คือ x-intercept of the secant line จาก (a,L) ถึง (b,R) ดังรูป (ส่วนใน BIS method x_{test} คือจุดแบ่งครึ่งช่วง [a,b])



By similar triangles ($\Delta AIJ \sim \Delta ABC$)

$$\frac{\overline{AJ}}{\overline{IJ}} = \frac{\overline{AC}}{\overline{BC}}, \text{ that is, } \frac{x_{FP} - a}{0 - L} = \frac{b - a}{R - L}$$

Solving:

$$x_{FP} = a + \frac{L}{L - R}(b - a) = \frac{bL - aR}{L - R}$$

รูป 2.1-10 แสดงที่มาของสูตรสำหรับ x_{FP}

ตัวอย่าง จงทำซ้ำ 8 ครั้งโดยใช้ (1) BIS method

(2) FP method

$f(x) = x^2 - 70.8$,a = 6, b = 12

Solution ตาราง 2.1-4 การทำซ้ำโดยวิธี (a) BIS และ (b) FP สำหรับ $f(x) = x^2 - 78.8$

a = 6, $L = f(a) = -42.8$ (Sign of L is ⊖)

$b = 12$, $R = f(b) = 65.2$ (Sign of R is ⊕)

(a) Bisection Method

k	a	b	$x_k = \frac{a+b}{2}$	$Y = f(x_k)$	$\Delta x_{k-1} = x_k - x_{k-1}$
1	6.	12.0	9.0	2.2	—
2	6.	9.0	1.5	-22.55	- 1.5
3	7.5	9.0	8.25	- 10.7375	0.75
4	8.25	9.0	8.625	-4.409375	0.375
5	8.625	9.0	8.8125	- 1.139844	0.1875
6	8.8125	9.0	8.90625	0.5212891	0.09375
7	8.8125	8.90625	8.859375	-0.3114746	-0.046875
8	8.859375	8.90625	8.882813	0.1043579	0.023438

(b) False Position Method

k	a	b	$x_k = \frac{bL - aR}{L - R}$	$Y = f(x_k)$	$\Delta x_{k-1} = x_k - x_{k-1}$
1	6.0	12.0	8.377778	-8.612840	—
2	8.377778	12.0	8.800436	- 1.352323	0.422658
3	8.800436	12.0	8.865450	-0.2037901	0.065014
4	8.865450	12.0	8.875217	−0.3051934E−1	0.009767
5	8.875217	12.0	8.876679	−0.4566260E−2	0.001462
6	8.876679	12.0	8.876898	−0.6831018E−3	0.000219
7	8.876898	12.0	8.87693 1	−0.1021884E−3	0.000033
8	8.87693 I	12.0	8.876936	−0.1528675E−4	0.000005

จากตารางข้างบน x_k คือ x_{test} ตัวที่ k

พิจารณา Δx คอลัมน์ เราพบว่า

สำหรับ BIS: $\left|\Delta x_k\right| / \left|\Delta x_{k-1}\right| = 1/2$ (i.e., $\Delta x_k = \pm 1/2\ \Delta x_{k-1}$)

สำหรับ FP: $\Delta x_k / \Delta x_{k-1} \approx$ Constant (i.e., $\Delta x_k \approx C\Delta x_{k-1}$)

โดยที่ $C \approx 0.15$

แสดงการลู่เข้าแบบเชิงเส้นไปยังรากเชิงเดียว $\bar{x} = \sqrt{78.8}$ ซึ่งช้ากว่าวิธีของนิวตัน-ราฟสัน หรือ วิธีเซแคนต์มาก

2.1H สรุปะการเปรียบเทียบวิธีต่าง ๆ ในการหาราก

ในสถานการณ์ส่วนมาก วิธีเซแคนต์นั้นมีประสิทธิภาพดีพอ ๆ กับวิธี BIS หรือวิธี FP ดังนั้นในการเปรียบเทียบจะทำการเปรียบเทียบวิธีของนิวตัน-ราฟสันและวิธีเซแคนต์เท่านั้น

Accuracy and Rate of Convergence

สำหรับรากเชิงเดียว (simple root) วิธีของนิวตัน-ราฟสันมักจะลู่เข้าแบบอันดับสอง โดยที่ความแม่นยำเพิ่มเป็นเท่าตัวทุกหนึ่งหรือสองครั้งของการทำซ้ำ ส่วนวิธีเซแคนต์นั้นการลู่เข้าเกือบเป็นแบบอันดับสองโดยที่ต้องการจำนวนการทำซ้ำมากกว่าวิธีของนิวตัน-ราฟสัน 25% จึงจะได้ความแม่นยำเท่ากันทั้งนี้โดยใช้ x_o เดียวกัน

ทั้งวิธีของนิวตัน-ราฟสัน และวิธีเซแคนต์ จะลู่เข้าแบบเชิงเส้นเมื่อหารากค่าซ้ำ (multiple roots)

Robustness

ทั้ง 2 วิธีอาจจะล้มเหลวที่จะลู่เข้าสู่รากที่ต้องการถ้า slope ใกล้ x_o มีค่าใกล้ศูนย์ เนื่องจาก m_{sec} บางส่วนขึ้นอยู่กับค่าจากการทำซ้ำครั้งหนึ่งก่อนหน้านั้น

ปัญหาของ overshoot จึงรุนแรงน้อยกว่าวิธีของนิวตัน-ราฟสัน จากเหตุผลเดียวกัน วิธีเซแคนต์จึงเกิด "cycle" หรือ "wander" น้อยกว่าวิธีของนิวตัน-ราฟสัน

Ease of Hand Calculating

ถ้า f(x) ไม่ซับซ้อนมาก เรามักจะหาและพยายามทำรูป f(x)/f'(x) ให้ง่ายแล้วใช้วิธีของนิวตัน-ราฟสัน สำหรับ f(x) ที่ดูยุ่งยากซับซ้อน เราควรใช้วิธีเซแคนต์

Ease of Programming

โปรแกรมสำหรับวิธีเซแคนต์ นั้นมีความยาวเท่า ๆ กับวิธีของนิวตัน-ราฟสัน โปรแกรมสำหรับวิธีของนิวตัน-ราฟสันนั้นมีขั้นตอนการ update ค่าน้อยกว่าแต่ต้องกำหนดฟังก์ชันสำหรับ f(x) และ f'(x)

Ease of Using a Program

โปรแกรมสำหรับวิธีเซแคนต์นั้นใช้ได้ง่ายกว่า โดยเฉพาะอย่างยิ่งเมื่อ f(x) ยุ่งยาก ทั้งนี้เพราะวิธีนี้ไม่ต้องหา f'(x) แล้วกำหนดไว้ในโปรแกรม

2.1I การหารากโดยใช้คอมพิวเตอร์

ซับรูทีน SECANT เพื่อหารากของ f(x) โดยวิธีเซแคนต์ ตามขั้นตอนวิธีใน 2.1A

```
C ⊠ ⊠ ⊠ • ★ ⊠ ★ ⊠ ★ CALSEC • ⊠ ⊠ ★ ★ • ⊠ • ★ ★ ⊠ ⊠
C INTERACTIVE CALLING PROGRAM FOR SUBROUTINE SECANT
      EXTERNAL F3
      DATA IW, IR, MAXIT /5, 5, 201
C
      WRITE(IW,1)
    1 FORMAT('0ENTER DESIRED # SIGNIFICANT DIGITS, IOUT (0 - 2)')
      READ(IR,*) NUMSIG, IOUT
      WRITE(IW,2)
    2 FORMAT(' ENTER TWO INITIAL GUESSES (SECOND CLOSER IF POSSIBLE)')
      READ(IR,*) XPREV, X
C
      CALL SECANT(F3, NUMSIG, MAXIT, XPREV, X, Y, IOUT, IEXIT, IW)
C
      STOP
      CND

      FUNCTION F3(X)
C     FOR EXAMPLE OF SECTION 2.1E
         F3 = SIN(X)/COS(X) - X*EXP(-X*X)
      RETURN
      FND
```

รูป 2.1-11 โปรแกรมเรียกใช้ซับรูทีน SECANT

```
      SUBROUTINE SECANT(F,NUMSIG,MAXIT,XPREV,X,Y,IOUT,IEXIT,IW)
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - C
C THIS SUBROUTINE FINDS ROOTS OF F(X) USING THE SECANT METHOD.       C
C                                                                    C
C IN THE PARAMETER LIST, THE USER PROVIDES:                          C
C            F = NAME OF SUBPROGRAM FOR FUNCTION F(X)                C
C       NUMSIG = DESIRED NUMBER OF SIGNIFICANT DIGITS                C
C        MAXIT = MAX NUMBER OF ITERATIONS                            C
C     XPREV, X = TWO INITIAL GUESSES                                 C
C         IOUT = 0 TO SUPPRESS ALL OUTPUT (TO DEVICE IW)             C
C                1 TO OUTPUT FINAL RESULTS ONLY                      C
C                2 TO OUTPUT DETAILS FOR EACH ITERATION              C
C                                                                    C
C THE SUBROUTINE RETURNS:                                            C
C            x = CURRENT X WHEN TERMINATION OCCURRED                 C
C            Y = F(X)                                                C
C        IEXIT = 1 IF A ZERO SLOPE WAS ENCOUNTERED                   C
C                2 IF A ZERO FUNCTION VALUE WAS ENCOUNTERED          C
C                3 IF CONVERGENCE APPARENTLY OCCURRED                C
C                4 IF CONVERGENCE DIDN'T APPEAR TO OCCUR             C
C                                                                    C
C SUBPROGRAM NAME F MUST GE DECLARED EXTERNAL IN CALLING PROGRAM     C
C - - - - - - - - - - - - - - - - - - - - - VERSION 1: 5/1/81 - -    C
C INITIALIZE:
        RELTOL = 10.**(-NUMSIG)
        YPREV = F(XPREV)
        Y = F(X)
C
        IF CIOUT .GT. 1) WRITE(IW,5) XPREV, YPREV, X, Y
    5   FORMAT(' SECANT METHOD     X(0)=SECOND INITIAL GUESS',/,
     &         '0 K',6X,'X=XK',10X,'Y=F(X)',8X,'X-XPREV',/,
     &         ' -1 ",F12.7,4X,E12.5, '/ ,' 0 ',F12.7,4X,E12.5)
C
C ITERATE (SECANT METHOD):
      IEXIT = 1
      DO 100 K=l, MAXIT
C
C      **FORM DELTAX = -Y/SLOPE (PROVIDED SLOPE IS NOT ZERO)
         SLOPE = (Y-YPREV)/(X-XPREV)
         IF (SLOPE .EQ. 0.) GOT0 200
         DELTAX = -Y/SLOPE
C
C      **UPDATE X AND Y
         XPREV = X
         YPREV = Y
         X = XPREV + DELTAX
         Y = F(X)
C
         IF (IOUT .GT. 1) WRITE(IW,6) K, X, Y, DELTAX
    6    FORMAT(I3, 1X, F12.7, 4X, E12.5, 3X, E12.5)
C
C      **CONVERGENCE TESTS
         IF (Y .EQ. 0.) IEXIT = 2
         IF (ABS(DELTAX) .LE. RELTOL*ABS(X)) IEXIT = 3
         IF (IEXIT .GT. 1) GOT0 200
  100 CONTINUE
      IEXIT = 4
C
  200 IF CIOUT .EQ. 0) RETURN
      IF (IEXIT .EQ. 1) WRITE(IW,1) x
      IF (IEXIT .EQ. 2) WRITE(IW,2) X
      IF CIEXIT .EQ. 3) WRITE(IW,3) X, NUMSIG
      IF CIEXIT .EQ. 4) WRITE(IW,4) MAXIT
    1 FORMAT('OSLOPE=O WHEN X=',F12.7,'. ITERATION DISCONTINUED')
    2 FORMAT('OCOMPUTED F(',F12.7,') = ZERO. ITERATION DISCONTINUED')
    3 FORMAT('OX=',F12.7,' APPEARS ACCURATE TO ',I1,'S')
    4 FORMAT('ODESIRED ACCURACY IS NOT EVIDENT IN',I3,'ITERATIONS')
      RETURN
      END
```

รูป 2.1-12 ซับรูทีน SECANT

2.2 การหารากของ Real Polynomials: วิธีของแบร์สโตว์ (Bairstow's Method)

2.2A การลดรูปของ Real Polynomials โดยการใช้ตัวประกอบกำลังสอง (Quadratic Factors)

ถ้า $p(x)$ เป็น real polynomial (นั่นคือส.ป.ส.ทั้งหมดของ $p(x)$ เป็นเลขจำนวนจริง) แล้วเราไม่ต้องใช้การคำนวณเชิงซ้อน ทั้งนี้เพราะรากเชิงซ้อนของ real polynomials จะอยู่ในรูปค่าสังยุค (conjugate pairs) (นั่นคือ $\alpha \pm i\beta$)

real quadratic $q(x) = (x-\alpha-i\beta)(x-\alpha+i\beta)$

$= (x-\alpha)^2+\beta^2$ เป็น real factor ของ $p(x)$

นั่นคือมี real polynomial $Q(x)$ ที่ทำให้ $p(x) = q(x)Q(x)$
โดยที่ $Q(x)$ มี degree = [degree ของ $p(x)$] - 2

เมื่อ $n>=3$ รากทั้งหมดของ real polynomial $p(x)$ อาจหาโดยการใช้ **Quadratic deflation algorithm** ดังนี้

```
Algorithm: Quadratic Deflation

Purpose: To find all roots of a real polynomial p(x) = a1 x^n + ... + an x + a(n+1)

(initialize)
GET n, a1, ..., a(n+1)   {a1 ≠ 0}

(iterate)
DO WHILE n ≥ 2
  BEGIN
  {factor} Find a real quadratic q(x) such that p(x) = q(x)Q(x)
  (solve) Final the two roots r1, r2 of q(x) by the quadratic formula
  OUTPUT (Roots: r1, r2)
  (deflate) BEGIN p(x) ← Q(x); n ← n - 2 END
  END

IF n = 1 THEN BEGIN LastRoot ← -a2 / a1; OUTPUT (Root: LastRoot) END
```

รูป 2.2-1 รหัสเทียมสำหรับ Quadratic Deflation

ขั้น {solve} นั้นทำได้โดยการเรียกใช้ ซับรูทีน QROOTS (ใน 1.5C)
ส่วนขั้น {factor} จะใช้วิธีการของแบร์สโตว์ (Bairstow's method)
ซึ่งมีรากฐานมาจาก Synthetic Division Algorithm

2.2B การหารแบบสังเคราะห์ (Synthetic Division) ด้วย q(x)

ถ้า $p(x) = a_1x^n + a_2x^{n-1} + \ldots + a_nx + a_{n+1} \quad (a_1 \neq 0)$ ...(1)

ถูกหารด้วย real quadratic $q(x)$

โดยที่ $q(x) = x^2 - rx - s$ ...(2)

ผลลัพธ์คือ $Q(x)$ ซึ่งมี degree=(n-2) และ Remainder $R(x)$

ซึ่งมี degree <= 1

นั่นคือ $p(x)/q(x) = Q(x) + R(x)/q(x)$

หรือ $p(x) = q(x)Q(x) + R(x)$ ...(3)

$$p(x) = \underbrace{(x^2 - rx - s)}_{q(x)}\underbrace{(b_1x^{n-2} + b_2x^{n-3} + \ldots + b_{n-2}x + b_{n-1})}_{Q(x)} + \underbrace{b_n(x-r) + b_{n+1}}_{R(x)} \qquad \ldots(4)$$

กระจาย (4) แล้วเทียบกับส.ป.ส.ของ (1) ได้ผลดังตารางนี้

Power of x	Equated Coefficients	Solving for b_i
x^n	$a_1 = b_1$	$b_1 = a_1$
x^{n-1}	$a_2 = b_2 - rb_1$	$b_2 = a_2 + rb_1$
x^{n-2}	$a_3 = b_3 - rb_2 - sb_1$	$b_3 = a_3 + rb_2 + sb_1$
x^{n-3}	$a_4 = b_4 - rb_3 - sb_2$	$b_4 = a_4 + rb_3 + sb_2$
x	$a_n = b_n - rb_{n-1} - sb_{n-2}$	$b_n = a_n + rb_{n-1} + sb_{n-2}$
constant	$a_{n+1} = b_{n+1} - rb_n - sb_{n-1}$	$b_{n+1} = a_{n+1} + rb_n + sb_{n-1}$

...(5)

ค่าของ b_i's ซึ่งเป็นตัวกำหนด $Q(x)$ และ $R(x)$ นั้นอาจหาได้อย่างสังเคราะห์ [นั่นคือใช้เพียงค่าของ r และ s และส.ป.ส.ของ p(x) เท่านั้น]

ตัวอย่าง

$$\frac{2x^6 - 4x^5 + x^3 - 40x}{x^2 - x + 4} = (2x^4 - 2x^3 - 10x^2 - x + 39) + \frac{3(x-1) - 153}{x^2 - x + 4} \quad \ldots(6)$$

(a)	a_1	a_2	a_3	a_4	$\cdots$	a_{n-1}	a_n	a_{n+1}
$\times r>$		$+b_1 r$	$+b_2 r$	$+b_3 r$	$\cdots$	$+b_{n-2} r$	$+b_{n-1} r$	$+b_n r$
$\times s>$			$+b_1 s$	$+b_2 s$	$\cdots$	$+b_{n-3} s$	$+b_{n-2} s$	$+b_{n-1} s$
	b_1	b_2	b_3	b_4		b_{n-1}	b_n	b_{n+1}
(b) a_i's>	2	-4	0	1	0	-40	0	
$\times 1>$		2	-2	-10	-1	39	3	
$\times(-4)>$			-8	8	40	4	-156	
b_i's>	2	-2	-10	-1	+39	+3	-153	

รูป 2.2-Z Quadratic Synthetic Division: (a) $p(x)/(x^2-rx-s)$
(b) $(2x^6 -4x^5 +x^3 -40x)/(x^2 -x +4)$

2.2C วิธีของแบร์สโตว์ (Bairstow's Method)

จุดประสงค์คือหา q(x) ซึ่งเป็นตัวประกอบของ p(x) และทำให้ R(x) = 0 นั่นคือต้องการ $R(x) = b_n(x-r)+b_{n+1} = 0$ แสดงว่าเราต้องหา r และ s ที่ทำให้ b_n และ $b_{n+1} = 0$ ...(7)

ปัญหานี้ไม่ง่าย (not a trivail) เพราะจะเห็นจาก (5) ว่า $b_2,\ldots,b_{n+1}$ นั้นขึ้นอยู่กับ b's ที่นำหน้ามา และเป็นฟังก์ชันของ r และ s

ให้ $\bar{r}$ และ $\bar{s}$ เป็นค่าที่ทำให้ (7) จริง

สมมุติให้ r และ s เป็นค่าประมาณของ $\bar{r}$ และ $\bar{s}$

ถ้า $dr = \bar{r} - r$ และ $ds = \bar{s} - s$ ...(8)

ทั้ง 2 ตัวมีค่าน้อย และพิจารณาว่า b_n และ b_{n+1} เป็นฟังก์ชันของ (r,s) เราจะใช้ total differentials ของ b_n และ b_{n+1} เพื่อหาค่าประมาณ b_n และ b_{n+1}

$$0 = b_n(\bar{r},\bar{s}) = b_n(r+dr,s+ds) \approx b_n(r,s)+(\partial b_n/\partial r)dr+(\partial b_n/\partial s)ds \quad \ldots (9a)$$

$$0 = b_{n+1}(\bar{r},\bar{s}) = b_{n+1}(r+dr,s+ds) \approx b_{n+1}(r,s)+(\partial b_{n+1}/\partial r)dr +(\partial b_{n+1}/\partial s)ds \quad \ldots (9b)$$

จาก (8) $r_{k+1} = r_k \ t \ dr$, และ $s_{k+1} = s_k \ t \ ds_k$ ...(10)

ควรจะเป็นค่าประมาณของ $\bar{r}$ และ $\bar{s}$ ได้ดีกว่า r_k และ s_k

เพื่อที่จะทำให้เราสามารถใช้ iteration(10) เราทราบต้องค่าของ partial derivitives 4 ค่าใน (9) สิ่งที่แบร์สโตว์สังเกตเห็นคือถ้าทำการหารแบบสังเคราะห์ด้วย $q(x) = x^2 - rx - s$ ซึ่งเราทำโดยการแทน a_i's ด้วย b_i's แล้วหา $c_1,\ldots,c_n$ ดังต่อไปนี้

	b_1	b_2	b_3 . . .	b_n	b_{n+1}
x r		$+c_1r$	$+c_2r$	$+c_{n-1}r$	
x s			$+c_1s$	$+c_{n-2}s$	
	c_1	c_2	c_3 . . .	c_n	not needed

...(11)

แล้ว partial derivatives ใน (9) อาจหาได้จาก

$$\partial b_n/\partial r = c_{n-1}, \ \partial b_n/\partial s = c_{n-2}, \ \partial b_{n+1}/\partial r = c_n,$$

และ $ab_{n+1}/\partial s = c_{n-1}$ ...(12)

จาก (12) และ (9) เราจะเห็นว่า dr, และ ds, ใน (10) หาได้จาก การแก้ระบบสมการเชิงเส้นต่อไปนี้

$c_{n-1}dr + c_{n-2}ds = -b_n$

$c_n dr \quad + c_{n-1}ds = -b_{n+1}$ ...(13)

ดังนั้น $dr_k = (b_n c_{n-1} - b_{n+1} c_{n-2})/(c_n c_{n-2} - c_{n-1}^2)$

และ $ds_k = (b_{n+1} c_{n-1} - b_n c_n)/(c_n c_{n-2} - c_{n-1}^2)$ ...(14)

ผลการทำซ้ำสำหรับการหา q(x) เราเรียกว่าวิธีของแบร์สโตว์ ค่าเดาเริ่มต้นของ r_o และ s_o คือ 0 หรืออาจใช้วิธีดังต่อไปนี้

สำหรับรากค่าสูง ลองใช้ $r_o = -a_2/a_1$ และ $s_o = -a_3/a_1$ ...(15a)

สำหรับรากค่าต่ำ (ถ้า $a_{n-1} \neq 0$)

ลอง $r_o = -a_n/a_{n-1}$ และ $s_o = -a_{n+1}/a_{n-1}$ ...(15b)

ทั้งนี้เพราะรากค่าสูง $\bar{x}$ จะสอดคล้องกับ

$0 = p(\bar{x}) \approx (a_1\bar{x}^2 + a_2\bar{x} + a_3)\bar{x}^{n-2}$ ดังนั้น $\bar{x}^2 + (a_2/a_1)\bar{x} + (a_3/a_1) \approx 0$

และทำนองเดียวกัน รากที่เล็กมาก ๆ $\bar{x}$ จะสอดคล้องกับ

$a_{n-1}\bar{x}^2 + a_n\bar{x} + a_{n+1} \approx 0$

ขั้นตอนวิธีสำหรับวิธีของแบร์สโตว์และตัวอย่างที่ใช้วิธีของแบร์สโตว์เพื่อหารากของสมการ polynomial degree 5 คือ

Algorithm: Bairstow's Method *

Purpose: To find a quadratic factor $q(x) = x^2 - rx - s$ of an nth degree polynomial
$$p(x) = a_1x^n + \cdots + a_nx + a_{n+1} \ (a_1 \neq 0).$$

{initialize}

GET n, a_1, a_2, , a_{n+1}, (parameters of p(x))
$MaxIt$, $NumSig$, (termination parameters)
r_0, s_0 (initial guesses of r, s)

$b_1 \leftarrow a_1$; $c_1 \leftarrow b_1$; $r \leftarrow r_0$; $s \leftarrow s_0$; $Tol \leftarrow 10^{-NumSig}$

(iterate)

DO FOR $k = 1$ TO $MaxIt$ UNTIL **termination test** is satisfied
BEGIN [get remaining b's and c's by quadratic synthetic division)
$b_2 \leftarrow a_2 + b_1r$; $c_2 \leftarrow b_2 + c_1r$
DO FOR $i = 3$ TO $n + 1$
BEGIN
$b_i \leftarrow a_i + rb_{i-1} + sb_{i-2}$ [This is (5)]
$c_i \leftarrow b_i + rc_{i-1} + sc_{i-2}$ [This is (12))
END
{solve]

$$\left.\begin{aligned} Det &\leftarrow c_nc_{n-2} - c_{n-1}^2 \\ dr &\leftarrow (b_nc_{n-1} \quad b_{n+1}c_{n-2})/Det \\ ds &\leftarrow (b_{n+1}c_{n-1} - b_nc_n)/Det \end{aligned}\right\} \text{[This is (14))}$$

$r \leftarrow r + dr$; $s \leftarrow s + ds$ [This is (10))
OUTPUT (r, dr, s, ds)
[termination test: $|dr| \leqslant Tol * \max(1, |r|)$ **and** $|ds| \leqslant Tol * \max(1, |s|)$]
END

IF **termination test succeeded** {i.e. ($|\rho_r|$ or $|\epsilon_r| \leqslant Tol$) and ($|\rho_s|$ or $|\epsilon_s| \leqslant Tol$)}
THEN OUTPUT ($q(x) = x^2 - rx - s$ is a factor of p(x), and
$Q(x) = b_1x^{n-2} + + b_{n-2}x + b_{n-1}$ i s $p(x)/q(x)$.)
(In this case the p(x) $\leftarrow$ Q(x) step of Quadratic **Deflation** is simply)
{ DO FOR $i = 2$ TO $n - 1$ $a_i \leftarrow b_i$ }
ELSE OUTPUT (Convergence is not apparent in $MaxIt$ iterations.)

รูป 2.2-3 รหัสเทียมสำหรับวิธีของแบร์สโตว์

ตัวอย่าง* จงหารากทั้งหมดของ 5th-degree polynomial

$$p(x) = x^5 - 4.5x^4 \text{ t } 4.55x^3 \text{ t } 2.675x^2 - 3.3x - 1.4375 \qquad \ldots(16)$$

โดยใช้วิธีของแบร์สโตว์ ให้หารากที่มีขนาดเล็กที่สุดก่อน

Solution โดยใช้ (15b) เราจะเริ่มด้วยค่าเดาสำหรับ r และ s ดังนี้

r_o = -[(-3.3)/2.675] = 1.233645 และ

s_o = -[(-1.4375)/2.675] = 0.5373832

ค่าของ c_i's ซึ่งจะใช้ในการหา dr_o และ ds_o นั้นได้จากการทำการหารแบบสังเคราะห์ 2 ครั้งดังนี้

a_i's:	1	-4.5	+4.55	+2.675	-3.3	-1.4375
×1.233645>		t1.233645	-4.029522	+1.305025	+2.744540	+0.016057
×0.5373832>			+0.537383	-1.755284	+0.568477	+1.195538
b_i's:	1	-3.266355	+1.057861	+2.224740	+0.013016	-0.225904
×1.233645>		+1.233645	-2.507643	-1.125576	+0.008414	
×0.5373832>			+0.537383	-1.092344	-0.490308	
c_i's:	I	-2.032710	-0.912398	+0.006820	-0.468877	

โดยใช้ (14) ดังนั้น

$$dr_0 = \frac{(0.013016)(0.006820) - (-0.225904)(-0.912398)}{(-0.468877)(-0.912398) - (0.006820)^2} = \frac{-0.2060261}{0.4277564} = -0.4816435$$

$$ds_0 = \frac{(0.006820)(-0.225904) - (-0.468877)(0.013016)}{-0.4277564} = +0.01066556$$

$r_1 = r_o \text{ t } dr_o$ = 1.233645 - 0.4816435 = 0.752001

$s_1 = s_0 + ds_o$ = 0.5373632 t 0.01066556 = 0.5480488

ทำซ้ำต่อไปเรื่อย ๆ ได้ผลในตารางต่อไปนี้

<u>ตาราง 2.2-1</u> การลู่เข้าของวิธีของแบร์สโตว์

k	Ar	AS	r	s
0	—		1.233645	0.5373832
1	-0.4816435	0.0106656	0.752001	0.5480488
2	0.1478756	0.2232268	0.899877	0.7712756
3	0.523493 1	0.1512589	1.423370	0.9225345
4	-0.0828045	-0.7647468	1.340565	0.1577877
5	0.6070840	0.3324581	1.948274	0.4902458
6	-2.8041133	-0.6506271	-0.855840	−0.1603809
7	-0.1514431	-0.0790732	− 1.007283	-0.23945 12
8	0.766207E−2	−0.990590E−2	0.999620	-0.2493600
9	0.3797788-3	−0.640298E−3	−1.000000	−0.2500003
10	0.262E−6	0.3218E−6	−1.000000	−0.2500000

จากคอลัมน์ Δr และ Δs ในตารางข้างต้น แสดงว่าการลู่เข้า (convergence) เป็นแบบอันดับสอง เมื่อ r_k และ s_k อยู่ใกล้ $\bar{r} = -1$ และ $\bar{s} = -0.25$ (นั่นคือเมื่อ $k >= 7$) อย่างไรก็ดี ทั้ง r_k และ s_k เปลี่ยนไปมา (wandered) ในการทำซ้ำ 6 ครั้งแรก

ดังนั้นตัวประกอบกำลังสองคือ $q(x) = x^2 - (-1)x - (-0.25)$

$$= (x + 0.5)^2 \qquad \ldots(17)$$

ซึ่งมีราก 2 รากที่เหมือนกันคือ -0.5 (-0.5 คือรากซ้ำสองหรือ double root) รากทั้ง 2 รากข้างต้นเป็นราก 2 รากที่เล็กที่สุดของ p(x) ด้วย

ทำการหาร p(x) ด้วย $q(x) = x^2$ t x t 0.25 จะได้

a_i's)	1	-4.5	+4.55	t2.675	-3.3	-1.4375
x(-1))		-1.0	t5.5	-9.8	t5.75	0
x(-0.25))			-0.25	t1.375	-2.45	t1.4375
b_i's)	1	-5.5	+9.8	-5.75	0	0

ดังนั้น $p(x) = (x+0.5)^2 Q(x)$

โดยที่ $Q(x) = x^3 - 5.5x^2 + 9.8x - 5.75$ ...(18)

จาก (15b) ค่าเดาเริ่มต้นสำหรับหาการรากที่เล็กที่สุดของ $Q(x)$ คือ

$$r_o = -(9.8)/(-5.5) \doteq 1.781818$$

และ $s_o = -(-5.75)/(-5.5) \doteq -1.045455$ ...(19)

ใช้ r_o และ s_o ใน (19) สำหรับใช้วิธีของแบร์สโตว์กับ $Q(x)$ ได้ผลในตารางต่อไปนี้

ตาราง 2.2-2 วิธีของแบร์สโตว์สำหรับ p(x) ที่ถูกลดรูปลงแล้ว

k	Δr	ΔS	r	s
0	-	-	1.7818180	-1.0454550
1	.9900225E+00	-.2123772E+00	2.7718410	-1.2578320
2	.2243017E+01	-.1078122E+01	5.0148570	-2.3359550
3	.4447243E+00	-.7045599E+01	5.4595810	-9.3815540
4	-.5982121E+00	.3044031E+01	4.8613690	-6.3375230
5	-.4490317E+00	.1538286E+01	4.4123370	-4.7992360
6	-.3166079E+00	.8509890E+00	4.0957290	-3.9482470
7	-.9467953E+00	.2448020E+01	3.1489340	-1.5002270
8	.9065881E-01	-.9687552E+00	3.2395930	-2.4689820
9	-.6083820E+00	.5875002E+00	2.6312110	-1.8814820
10	.3234949E+00	-.2932734E+00	2.9547060	-2.1747560
11	.5034394E-01	-.1252603E+00	3.0050500	-2.3000160
12	-.5123975E-02	.7923925E-04	2.9999260	-2.2999370
13	.7459542E-04	-.6339898E-04	3.0000000	-2.3000000
14	-.2270622E-07	.1135291E-07	3.0000000	-2.3000000

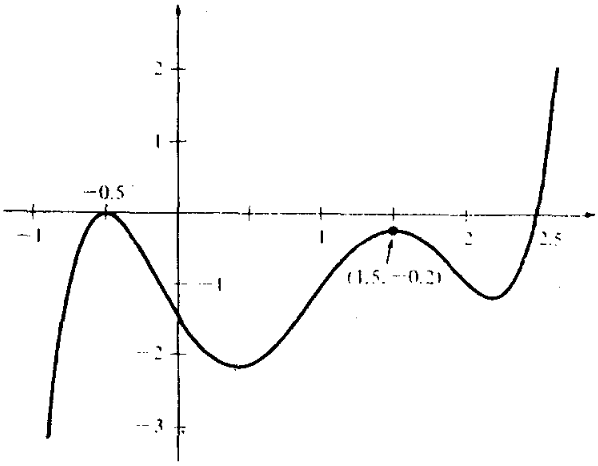
ดังนั้น $q(x) = x^2 - (3)x - (-2.3) = (x-1.5)^2 + 0.05$ ...(20)

$q(x)$ มีรากเป็นค่าสังยุค (complex conjugate roots) $1.5 \pm \sqrt{0.05}\, i$ หาร $p(x)$ ซึ่งถูกลดรูปลงแล้ว [i.e., $Q(x)$ ใน (18)] ด้วย $q(x)$ ใน (20) ได้ผลลัพธ์คือ $x-2.5$

ดังนั้นรากทั้ง 5 ของ $p(x)$ ใน (16) คือ

$-0.5,\ -0.5,\ 1.5+\sqrt{0.05}\, i,\ 1.5-\sqrt{0.05}\, i,\ 2.5$ ...(21)

กราฟของ $p(x)$ ดังแสดงในรูป



รูป 2.2-4 กราฟของ p(x)

แบบฝึกหัดบทที่ 2

2.1 Let $f(x)=x^5-6x^4+9x^3$, $\bar{x}=0$, $x_0=-0.3$ and for (SEC) $x_{-1}=-0.4$

(a) Find x_1, x_2, x_3 and x_4 by (NR). Verify that the convergence to x is linear (i.e., $\Delta x_k \approx C\Delta x_{k-1}$)

(b) Repeat part a) for x_1, x_2, x_3, and x_4 obtain. using (SEC).

2.2 Do (a) and (b) of Exercise 2.1 uith $f(x)=x^5-6x^4+9x^3$ again, but with $\bar{x}=3$, $x_0=3.06$ and for (SEC) $x_{-1}=3.1$.

2.3 Perform two iterations of Bairstou's method as indicated.

(a) $p(x)=x^4-5x^3+8.5x^2-6x+2$, $r_0=1$, $s_0=0$.

(b) $p(x)=2x^4+5x^3-5x^2-5x+3$, $r_0=-3$, $s_0=-3$.

Given r_8 =-2.5 and s_8 =1.5 find $Q(x)$ then roots of $p(x)$ = 0.

(c) $p(x)=x^5+x^4-14.8x^3+23.4x^2-12.6x+2$, $r_0=2$, $s_0=-2$.

(d) $p(x)=x^4-x^3+x^2+2$, $r_0=2$, $s_0=-1$.

Given r_7 =2 and s_7 =-2 find $Q(x)$ then roots of $p(x)$ = 0.

2.4 The polynomial $p(x)=x^4-10x^3+35x^2-50x+24$ has roots x = 1, 2, 3, 4

(a) Starting with r_0 = 3.2, s_0 = -2.2 Cto find $q(x)=(x-1)(x-2)]$ Use Bairstou's method to find r_k and s_k until successive iterates agree to 3s. Deflate $p(x)$ and find the error in the roots of the deflated $p(x)$ (whose roots, if exact would be 3 and 4).

(b) Starting with $r_0 = 7.3$, $s_0 = -12.3$ [to find $q(x) = (x-3)(x-4)$] Use Bairstow's method to find r_k and s_k until successive iterates agree to 3s. Deflate $p(x)$ and find the error in the roots of the deflated $p(x)$ (whose roots, if exact would be 1 and 2).

คำตอบ

		x_1	x_2	x_3	x_4
2.1	(a)	-0.205714	-0.139956	-0.094650	-0.063731
	(b)	-0.234046	-0.178752	-0.136971	-0.104386

		x_1	x_2	x_3	x_4
2.2	(a)	3.03086	3.01566	3.00769	3.00396
	(b)	3.03882	3.02410	3.01506	3.00936

		r_1	r_2	r_3	r_4
2.3	(a)	4/3	-1/3	-1.12397	-0.332415
	(b)	1.93099	-1.29774	1.84501	-0.937337