שיטות איטרטיביות

האלגוריתם של *Heron*

נחשב לדוגמא $f(a)=\sqrt{a}$. ישנן כמה דרכים לחשב זאת.

1. *Heron's Algorithm* :

$$x_0=b$$

$$x_{n+1}=\frac{1}{2}\left(x_n+\frac{a}{x_n}\right)$$

ניקח את $a=2$ לדוגמא. אזי :

$$x_0=1$$

$$x_1=\frac{1}{2}(1+2)=\frac{1}{2}\cdot 3=1.5$$

$$x_2=\frac{1}{2}\left(1.5+\frac{2}{1.5}\right)=1.4\ldots$$

החישובים מתכנסים בסופו של דבר ל- $\sqrt{2}$.

2. לפי טור טיילור : $f(x+h)=f(x)+f'(x)\cdot(x-h)+\ldots+\frac{f^{(k)}(x)}{k!}(x-h)^k+o(x-h)^k$

3. $\sqrt{2}=1+\cfrac{1}{2+\cfrac{1}{2+\cfrac{1}{2+\cfrac{1}{\ddots}}}}$

כשבודקים עדיפות אלגוריתמים שתוצאתם זהה , ראשית נבדוק את הדיוק. אם יש דיוק זהה בשניהם נבדוק את מספר הפעולות וכו'.

נשאל כעת : <u>אילו מהאלגוריתמים עדיף.</u>

1. מי נותן דיוק מרבי
2. אפקטיביות הפעולות
3. זמן

הדרכים הטובות ביותר הן 1,3 (מינימום פעולות ומקסימום דיוק). הדבר נובע מבעייתיות הנגזרת במספרים קטנים (לפי טיילור). אם נציב $h\sim 0$ במשוואה $F(x,h)=\frac{f(x+h)-f(h)}{h}$ אז נקבל שגיאה גדולה (בגלל חילוק ב-0).

נבדוק את יעילות האלגוריתם *polyval (...)* ב-*Matlab* :

* הפקודה $flops$ סופרת את מספר הפעולות האריתמטיות. $flops(0)$ מאפס את המונה.

* הפקודה $polyder(p)$ מחשבת את מקדמי הנגזרת.

$>> flops(0)$

$>> p = [1,2,3,4,5,6]$

$>> polyval(p, 0.5678)$

$>> flops$

45

ערך הפולינום

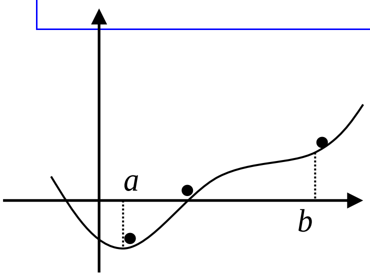
קיבלנו שהאלגוריתם לקח 45 פעולות, כלומר הוא לא האלגוריתם היעיל ביותר. אפשר לפתור את האלגוריתם הזה ע״י 10 פעולות בלבד(*Horner scheme*)

כעת, נבדוק מדוע באלגוריתם של *Heron* $x_n(a) \xrightarrow{\infty} \sqrt{a}$.

$$\Delta_{\sqrt{a},n+1} = x_{n+1} - \sqrt{a} = \frac{1}{2x_n}\left(x_n^2 + a\right) - \sqrt{a} = \frac{1}{2x_n}\left(x_n - \sqrt{a}\right)^2 = \frac{1}{2x_n}\cdot\Delta^2_{\sqrt{a},n}$$

ולכן השגיאה יורדת. אחרי 5-6 שלבים נצפה לקירוב טוב מאוד , אולם חשוב גם לקחת נקודת התחלה טובה.

אלגוריתמים לפתרון $\boldsymbol{f(x_1,\ldots,x_n)=0}$



<u>שיטה ראשונה: שיטת החצייה (Bisection)</u>

a,b הן נקודות התחלה. בין שתיהן נמצא הפתרון האמיתי.

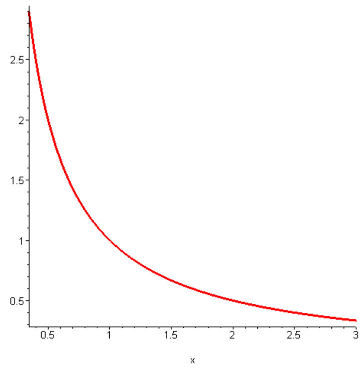
ניקח שתי נקודות כך ש- $f(a)\cdot f(b)<0$. נניח בה״כ ש- $f(b)>0,\ f(a)<0$. נגדיר את c להיות הממוצע בין שתי הנק'.

כלומר $c=\dfrac{a+b}{2}$. ואז:

$$if\ \ |f(c)|<eps \quad Exit$$
$$else,\ c = {a+b}/{2}$$
$$if\ \ f(c)>0,\ b=c$$
$$else\ a=c$$

נבדוק כמה פעולות צריך:

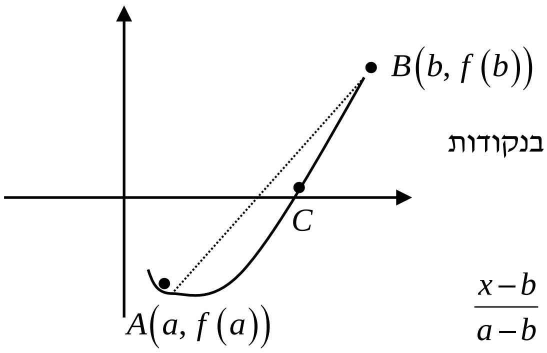
$$\left.\begin{array}{l} n=0: \|[a,b]\| = \rho \\ n=1: \|[a,b]\| = \rho/2 \end{array}\right. \Rightarrow 2^n = \frac{\rho}{eps} \Rightarrow \boxed{n = \log_2\left(\frac{\rho}{eps}\right)}$$



השגיאה בתוצאה $\Delta_n \le \dfrac{\rho}{2^n}$

n

שיטה שנייה: <u>שיטת המיתר *(Secant)*</u>

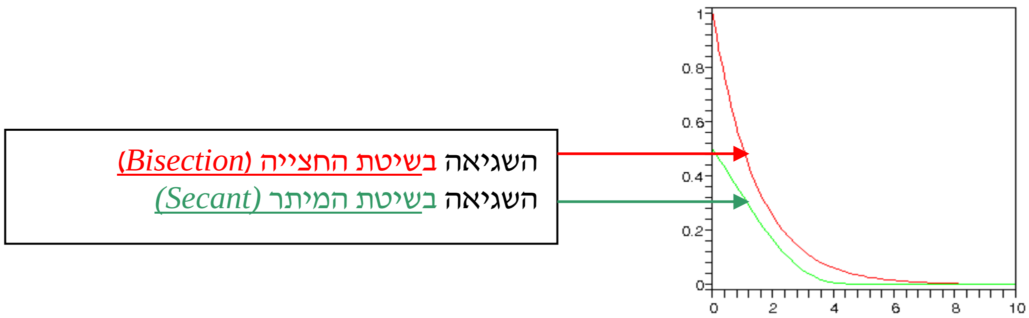


בוחרים a ו-b כך ש- $f(a)\cdot f(b)<0$. מחברים ישר AB וממשיכים כמו בשיטת החציה (על המיתר).
שיטה זו טובה יותר. מתחשבים במבנה הפונקציה ולא בנקודות אקראיות.
נבדוק כמה פעולות צריך לבצע כדי לקבל את c:

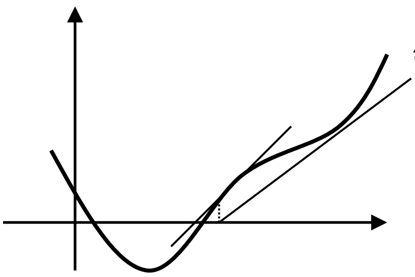
$$\frac{x-b}{a-b}=\frac{\overset{=0}{y}-f(b)}{f(a)-f(b)}$$

$$c=b-\frac{f(b)\cdot(a-b)}{f(a)-f(b)}\Rightarrow$$ נדרשות רק 5 פעולות

משפט. השגיאה Δ_n בשיטת המיתר מקיימת את התנאי $\Delta_{n+1}\le C\Delta_n^{\frac{1}{2}(1+\sqrt{5})}$



שיטה שלישית: <u>שיטת המשיק- *Newton –Raphson*</u>



בוחרים נקודה על הגרף, ומעבירים משיק. בנקודת החיתוך של המשיק עם ציר ה-x, בונים מהגרף משיק נוסף וכו'.

$$\underset{=0}{y}-f(x_n)=m(x-x_n)=f'(x_n)(x-x_n)$$

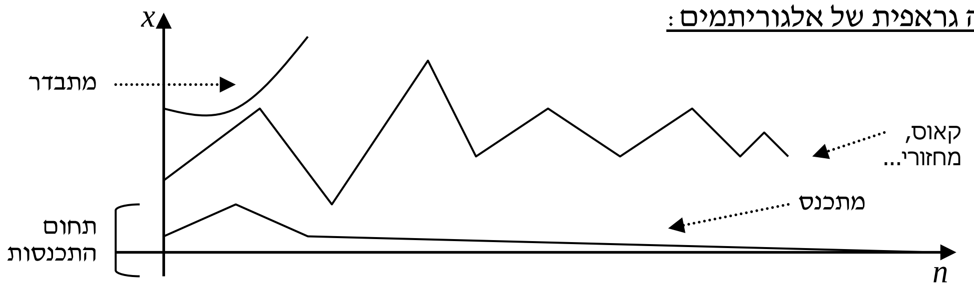
$$-f(x_n)=f'(x_n)(x_{n+1}-x_n)\Rightarrow\boxed{x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}}$$

$\Leftarrow$ זו השיטה היעילה ביותר מבין השלוש.

איטרציה פשוטה היא כזו המשתמשת באיבר קודם אחד, כלומר, $x_{n+1}=\varphi(x_n)$.

סדרת פיבונצ'י לדוגמא, היא אינה פשוטה, כי האיבר הכללי מיוצג כך: $x_{n+2}=x_{n+1}+x_n$

<u>המחשה גראפית של אלגוריתמים</u> :



הגרף האמצעי נקרא גם <u>מתכנס כאוטי</u>- סדר המחזור גדול מדי, כמו נניח $x_n = \left(-1\right)^n + \frac{1}{n}$.

$\Leftarrow$ המחשב מחשב מספרים בצורה שונה.

<u>לדוגמא</u> : $\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n \begin{cases} e \\ 1 \end{cases}$. המחשב ייתן 1 כאשר $\frac{1}{n} < ulp$ כי הוא יזניח את ערכו.

נסכם את השיטות :

1. שיטות ישירות (קרי עץ פעולות)- השגיאה רק גדלה
2. שיטות איטרטיביות- לפעמים אפשר להקטין את השגיאה עד כדי ε
3. איטרציות פשוטות – אם התהליך $x_{n+1} = \varphi\left(x_n\right)$ מתכנס אז שגיאה יורדת.

<u>הגדרה</u> : נקודה $x = x^*$ של אלגוריתם איטרציות פשוטות תיקרא נקודת שבת
באנגלית : $Fixed\ Point$ אם

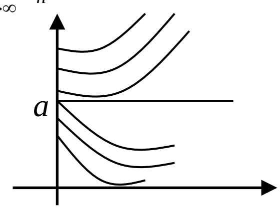
הפלט = קלט כלומר : $x^* = \varphi(x^*)$

באיטרציות פשוטות, אם נגיע פעם אחת לנקודת שבת, אז נישאר שם לתמיד.

מתאי ההתכנסות לאותה נקודה x^* בטוחה?

<u>הגדרה:</u> נקודת שבת תקרא נקודת שבת יציבה אם $\lim_{n\to\infty} x_n = a$: $\exists\varepsilon > 0, \forall\left\|x_0 - a\right\| < \varepsilon$.

לדוגמא :



כאן נקודת שבת אינה יציבה!

כיצד מגדרים ההתכנסות לנקודה שבת x^* ?

<u>הגדרה:</u> נקודת שבת x^* תקרא נקודת שבת יציבה אם סדר התכנסות r וקצב התכנסות q אם

$$\left\|x_{n+1} - x^*\right\| < q\left\|x_n - x^*\right\|^r$$

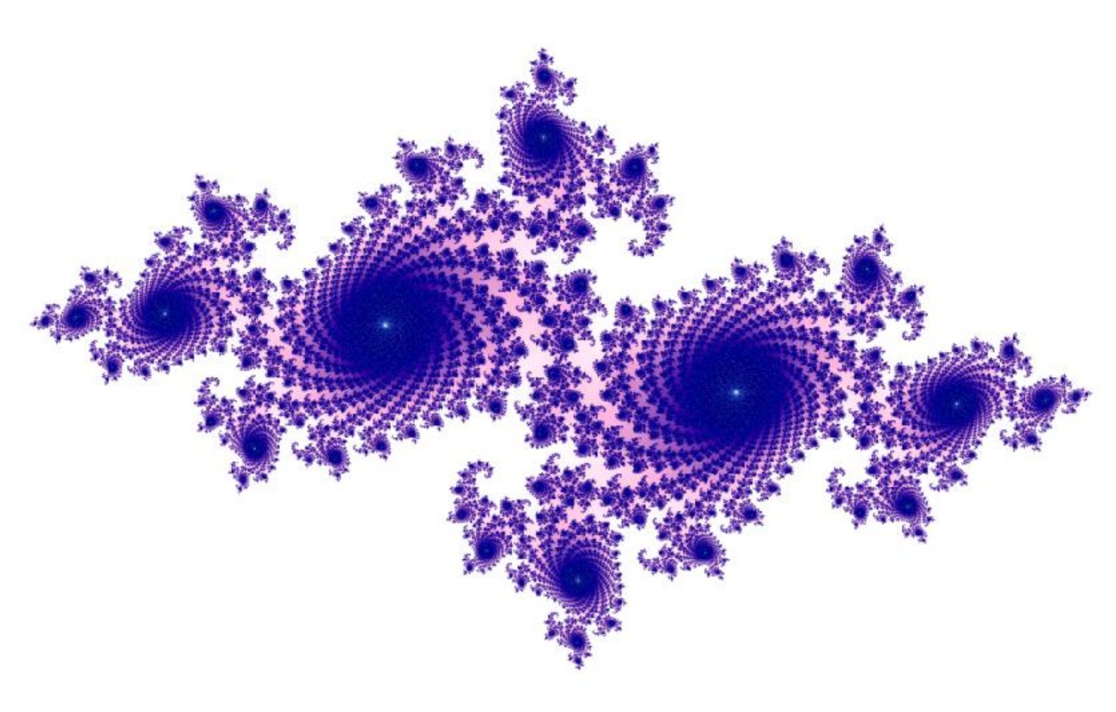
הגדרה: נגדיר <u>תחום ג'וליה</u> *(Julia set)* של איטרציה $x_{n+1} = \varphi(x_n)$ להיות:

$$J_K = \left\{ X \middle| \forall x_0 \in X, |x_n| \le K, n \ge 0 \right\}$$

באופן דומה נגדיר: $J_{K,N} = \left\{ X \middle| \forall x_0 \in X, |x_n| \le K, 0 \le n \le N \right\}$

לדוגמא: עבור $x_{n+1} = x_n^2$ תחום ג'וליה יהיה $J_K = \{x; |x| \le 1\}$. התחום זה האלגוריתם מעביר בתוכו. ולכן: $R \setminus J_K = \{x; |x| > 1\}$ הוא התחום שבו <u>האלגוריתם מתבדר</u>.

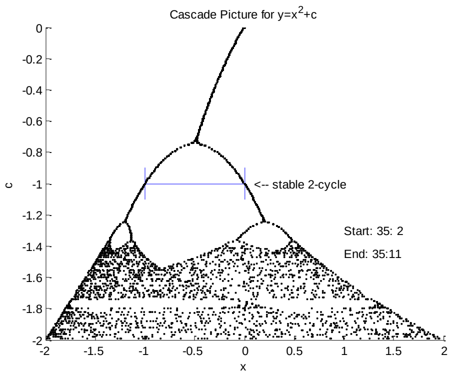
הערה: $J = \{\varnothing\}$ הוא תחום שלא חוסמים לכל *n*. (לדוגמה לאיטרציה $x_{n+1} = x_n^2 + 1$)



From Wikipedia, the free encyclopedia

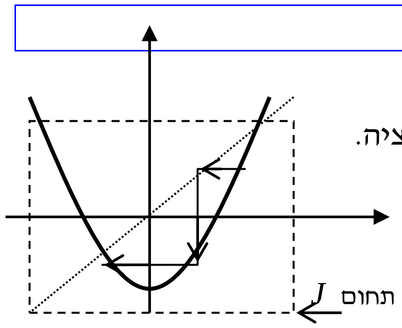
חקר אלגוריתמים איטרטיביים

אלגוריתם תמונת אשד



בעזרת אלגוריתם זה, ניתן לחקור אלגוריתמים אחרים.
דרך פעולתו : בוחרים נקודות התחלתיות מסוימות
עושים אתה האיטרציה כמה פעמים.
מאחר כך ממשיכים היטרציות ומציירים
תוצאות וכך ממשיכים למספר הנקודות התחלתיות.

אלגוריתם *Analyzer*



בעזרת אלגוריתם זה, ניתן לחקור אלגוריתמים איטרציה פשוטה.
דרך פעולתו : בוחרים נקודה מסוימת על גרף הפונקציה של איטרציה.
מן הנקודה מעבירים קו מקביל לציר x, עד שמגיעים לנקודה בה
$x=y$. משם, שוברים את הקו ב- 90° ומעבירים מקביל לציר y,
עד הנקודה בה מגיעים לנקודה על הגרף, מותחים משם קו
מקביל לציר x עד ש- $x=y$ וכו'.
אם מתחילים איטרציה בנקודה מסוימת ואחרי מספר איטרציות
מתקרבים לנקודת השבת, אזי האלגוריתם מתכנס ונקודות השבת הן יציבות.

$$(1)\ \left(x_n,\varphi\left(x_n\right)\right)\xrightarrow{line}\left(\varphi\left(x_n\right),\varphi\left(x_n\right)\right)=\left(x_{n+1},x_{n+1}\right)$$

$$(2)\ \left(x_{n+1},x_{n+1}\right)\xrightarrow{line}\left(x_{n+1},\varphi\left(x_{n+1}\right)\right)$$

יש שני סוגים של תחומי ג'וליה באיטרציה פולינומים :
1. קבעצה לא קשירה לגמרי *(totally disconnected set) Kantor set*
2. תחום קומפקטי.

מיון של נקודות שבת באמצעות כלים אלגברים

<u>תזכורת :</u> נקודת שבת היא פתרון של המשוואה $x=\varphi(x)$. נגזרת של φ היא רציפה.
נניח ש- $x=a$ היא נקודת שבת.
אם $|\varphi'(a)|>1$ אז נקודת השבת היא בלתי יציבה *repellent fixed point*
אם $|\varphi'(a)|<1$ אז נקודת השבת היא נקודת שבת יציבה, *attractive fixed point*

אם $|\varphi'(a)|=0$ אז נקודת השבת היא נקודת שבת יציבה במיוחד, *super attractive fixed point*
אם $|\varphi'(a)|=1$ אז לא ניתן לדעת לגבי נקודת השבת בלי חקירה נוספת, *indifferent fixed point*

דוגמא 1 :

$$x_{n+1}=x_n^2-2$$
$$x_1^*=2 \qquad x_2^*=-1$$
$$\varphi'(x)=\left(x^2-2\right)'=2x$$
$$\left|\varphi'\left(x_1^*\right)\right|=\varphi'(2)=4>1\Rightarrow unstable$$
$$\left|\varphi'\left(x_2^*\right)\right|=\left|\varphi'(-1)\right|=|-2|>1\Rightarrow unstable$$

כאן, תחום ג׳וליה הוא $J=[-2,2]$, יש לבדוק!

דוגמא 2 :

$$x_{n+1}=\frac{1}{2}\left(x_n+\frac{2}{x_n}\right)$$
$$\varphi(x)=\frac{1}{2}x+\frac{1}{x}=x\Rightarrow x^2=2\Rightarrow \boxed{x=\pm\sqrt{2}}$$
$$\varphi'(x)=\frac{1}{2}-\frac{1}{x^2}$$
$$\varphi'\left(\pm\sqrt{2}\right)=0\Rightarrow stable!$$

כאן, תחום ג׳וליה הוא $J=R\setminus\{0\}$

התנהגויות אלגוריתמים שונים (לדוגמה בסביבת 0):

$$(1)\quad x_n=\frac{1}{n}\to 0$$

$$(2)\quad x_n=\frac{1}{2^n}\to 0$$

$$(3)\quad x_n=\frac{1}{n!}\to 0$$

$$(4)\quad x_n=\frac{1}{2^{2^n}}\to 0$$

ההבדל היחידי הוא מהירות התכנסות האלגוריתמים.
האלגוריתם הרביעי לדוגמא, תהיה פחות מ ε אחרי 4 פעולות בלבד.
$\left(2^{-16}<\varepsilon\right)$

<u>כיצד נמדוד את ההתכנסות?</u>

נמיין את סדר וקצב ההתכנסות *(order and rate of convergence)* ע"י q,k :
אם אפשר להוכיח שלסדרה x_n שמתכנסת ל a
לכול $n>N$, $\|x_{n+1}-a\|\le q\|x_n-a\|^k$ או אם $\varepsilon_{n+1}\le q\varepsilon_n^{\,k}$, $\varepsilon_n=\|x_n-a\|$

$$(1)\quad \varepsilon_{n+1}=\frac{1}{n+1}\cdot\frac{n}{n}=\underbrace{\frac{n}{n+1}}_{q}\cdot\varepsilon_n$$

$$(2)\quad \varepsilon_{n+1}=\frac{1}{2^{n+1}}=\frac{1}{2}\cdot\frac{1}{2^n}=\frac{1}{2}\varepsilon_n$$

$$(3)\quad \varepsilon_{n+1}=\frac{1}{(n+1)n!}=\frac{1}{n+1}\cdot\varepsilon_n$$

$$(4)\quad \varepsilon_{n+1}=\frac{1}{2^{2^{n+1}}}=\frac{1}{\left(2^{2^n}\right)^2}=\varepsilon_n^2$$

case	k	q
(1)	1	~ 1
(2)	1	$1/2$
(3)	1	~ 0
(4)	2	1

אם חזקת השגיאה גדולה יותר, אז האלגוריתם מתכנס מהר יותר. אחרת, מסתכלים על q.
אם $k>1$, אז הנקודה תיקרא *super-attractive* כלומר, אטרקטיבית במיוחד.
נבדוק לפי זאת את האלגוריתמים של השיטות האחרונות שלמדנו לפתירת $f\left(x_1,\ldots,x_n\right)$
1. שיטת ניוטון-רפסון :

$$x_{n+1}-\sqrt{a}=\frac{1}{2x_n}\left(x_n-\sqrt{a}\right)^2$$

$$\Rightarrow q\sim\frac{1}{2\sqrt{a}}\qquad k=2$$

אטרקטיבית
במיוחד

2. שיטת החצייה : $\varepsilon_{n+1}\le\frac{1}{2}\varepsilon_n$. ולכן היא אינה אטרקטיבית במיוחד.
נוכיח את השגיאה :

$$\begin{cases}x_{n+1}=\varphi\left(x_n\right)\\ x^*=\varphi\left(x^*\right)\end{cases}\Rightarrow x_{n+1}-x^*=\varphi\left(x_n\right)-\varphi\left(x^*\right)$$

$$\varepsilon_{n+1}=\Delta_{n+1,x^*}=\left|x_{n+1}-x^*\right|=\left|\varphi\left(x_n\right)-\varphi\left(x^*\right)\right|\underset{\forall x^{**}\in\left[x_n,x^*\right]}{=}\left|\varphi'\left(x^{**}\right)\right|\cdot\left|x_n-x^*\right|=$$

$$=\left|\varphi'\left(x^{**}\right)\cdot\left(x_n-x^*\right)\right|$$

אם הנגזרת הראשונה שווה ל-0, ניקח $\varphi''(x^{**})$ וכך הלאה. ולכן, ניתן לכתוב זאת כנוסחת טיילור:

$$\left|\varphi'(x^*)\cdot(x_n-x^*)+\frac{\varphi''(x^*)}{2!}\cdot(x_n-x^*)+\ldots+\frac{\varphi^{(m)}(x^*)}{m!}\cdot(x_n-x^*)\right|$$

איבר ראשון השונה מ-0

סה"כ נקבל: $\varepsilon_{n+1}\le q\cdot\varepsilon_n^m$ $\left(q=\max\left\{\frac{\varphi^{(m)}(x^*)}{m}\cdot(x_n-x^*)\right\}\right)$

ולכן כדי שנוכל לבדוק מהו k, נגזור מספר פעמים את הפונקציה. אם הפונקציה גזירה אז $q=\max\limits_{x\in ox^*}\frac{\varphi^{(k)}(x)}{k!}$, בסביבה של נקודת שבת: $OA_{x^*}=\{x^*-\varepsilon_1,x^*+\varepsilon_2\}$.

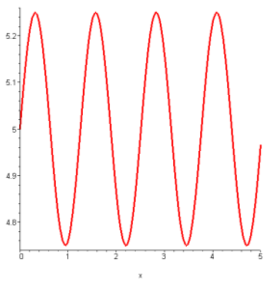
<u>הגדרה:</u> הקבוצה $A_{x^*}=\{x_0 \mid x_n\to x^*\}$ יקרא תחום בר-משיכה $(attractor)$ של הנקודה x^*. אם האלגוריתם מתכנס אז יש לנקודה תחום בר-משיכה לא רייק.

הערה: אם הנקודה יציבה, אז ניתן למין אותה ע"י כלים הבודקים מהירות של התכנסות.

הערה: כל תחום בר-משיכה שייך לתחום גיוליה.

<u>הגדרה:</u> נגדיר מחזור שבת מסדר 2 אם $y_n=\varphi(\varphi(y_n))=\varphi^{O2}(y_n)$

המחשה:



הערה: כל נקודות השבת נמצאות במחזור השבת מכול הסדר.

דוגמא: נמצא האם יש מחזור מסדר 2 באלגוריתם $x_{n+1}=x_n^2-2$:

$$x_{n+1}=x_n^2-2$$

$$\varphi(x)=x^2-2\Rightarrow\varphi(\varphi(x))=(x^2-2)^2-2=x^4-4x^2+2$$

$$y_{n+1}=y_n^4-4y^4+2$$

כעת אנו צריכים לבדוק מהם הפתרונות עבור $y=y^4-4y^2+2$. $y_2^*=-1$ $y_1^*=2$ הם פתרונות של המערכת, ולכן אנו יודעים שהפולינום הנ"ל מתחלק ב- y^2-y-2.

$$\begin{array}{r}
y^2+y-1 \\
y^2-y-2\overline{\big)\,y^4-4y^2-y+2} \\
-\underline{y^4-y^3-2y^2} \\
y^3-2y^2-y+2 \\
-\underline{y^3-y^2-2y} \\
-y^2+y+2
\end{array}$$

נקבל 2 פתרונות נוספים: $\frac{-1\pm\sqrt{5}}{2}$. כל 4 הפתרונות מקיימים את המשוואה $y=y^4-4y^2+2$,

כלומר מתקיים: $x_{2n+2}^*=\varphi(x_{2n}^*)=x_{2n}^*$. זוהי התכנסות מחזורי מסדר 2.

חזרה לפתרון משוואה $f(x)=0$

1. לפי שיטת ניוטון רפסון, נפתור את המשוואה ע״י פתרון של $x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}$. כדי למצוא נקודת שבת נפתור את $x=x-\frac{f(x)}{f'(x)}$.

דוגמא: $x^3-x=0$.

$$x^3-x=0 \Leftrightarrow x_{n+1}=x_n-\frac{x_n^3-x_n}{3x_n^2-1}=\frac{2x_n^3}{3x_n^2-1}$$

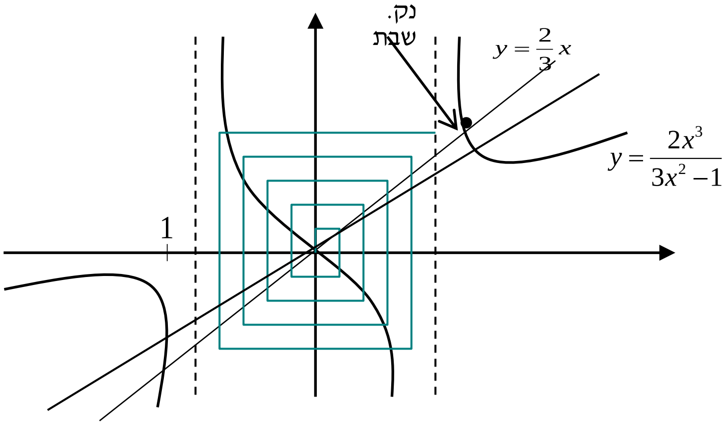
$$\varphi(x)=\frac{2x^3}{3x^2-1}\Rightarrow \varphi'(0)=\varphi'(\pm 1)=0$$

ולכן קיבלנו שיש נקודות שבת ב $x=0,\pm 1$. הנגזרת היא $2-3x^2$.

עבור $x=\pm 1$, $|2-3x^2|=1$ ולכן הנגזרת אינה יציבה! עבור $x=0$, $|2-3x^2|=2$ ולכן היא אינה יציבה גם כן.

אם נתחיל ב- $x_0=\frac{1}{\sqrt{3}}$, האלגוריתם יתבדר מכוון שנקודה זו אינה נמצאת בתחום ג׳וליה.

<u>המחשה גראפית:</u>



$J\not\ni \frac{1}{\sqrt{3}}, x_1, x_2, \dots, x_n, x_{n+1}, \dots$, אחרת נגיע למצב בו נצטרך לחלק ב-0.

מסלול של נקודה a או באנגלית *(orbit a)* זה התחום $O(a)=\{x_0 \mid \exists n: x_n=a\}$

דוגמה: מסלול של נקודה $\pm\frac{1}{\sqrt{3}}$ או התחום $O_{\pm\frac{1}{\sqrt{3}}}$ הוא תחום חשוד עבור שיטת ניוטון רפסון. הוא תחום בו האלגוריתם מתבדר. ולכן, רצוי מלכתחילה לבחור ניחוש התחלתי קרוב ככל האפשר לנקודת השבת. להשוות, מסלול של נקודה ± 1 שייך ל *attractor*.

$$f(x)=0 \Leftrightarrow x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}$$

$$f'(x^*)\neq 0 \Rightarrow x^*=x^*-\frac{f'(x^*)}{f'(x_n)}\Rightarrow f(x^*)=0$$

נחשב את השגיאה $\Delta_{x^*,n+1}$:

$$\Delta_{x*,n+1}=\left|x_{n+1}-x*\right|=\left|x_n-x*-\frac{f(x_n)-f(x*)}{f'(x_n)}\right|= \qquad \left(Q=\max_{x\in x*}\left|\frac{f''(x)}{\partial f'(x)}\right|\right)$$

$$=\left|x_n-x*-\frac{f'(x*)(x_n-x*)+f''(\tilde{x})\frac{(x_n-x*)^2}{2}}{f'(x_n)}\right|=\left|1-\frac{f'(x*)}{f'(x_n)}\right|\Delta_{x*,n}+Q\cdot(x_n-x*)^2$$

ולכן : $\frac{f''(\tilde{\tilde{x}})(x_n-x*)}{f'(x_n)}\le\left|\frac{f'(x_n)-f'(x*)}{f'(x_n)}\right|$, ואז נקבל ששיטת ניוטון-רפסון היא מסדר 2.

(כאשר $f'(x*)\neq 0$)

נניח ש- $f^{(k)}(x*)\neq 0$, $f'(x*)=0,\dots,f^{(k-1)}(x*)=0$ אז אפשר לרשום :

$$f(x)=(x-x*)^k\cdot g(x) \qquad (g'(x*)\neq 0)$$

$$f'(x)=k(x-x*)^{k-1}\cdot g(x)+(x-x*)^k\cdot g'(x)$$

$$\Rightarrow\frac{f(x)}{f'(x)}=\frac{(x_n-x*)^k\cdot g(x_n)}{k(x_n-x*)^{k-1}\cdot g(x_n)+(x_n-x*)^k\cdot g'(x_n)}=\frac{(x_n-x*)\cdot g(x_n)}{k\cdot g(x_n)+(x_n-x*)\cdot g'(x_n)}$$

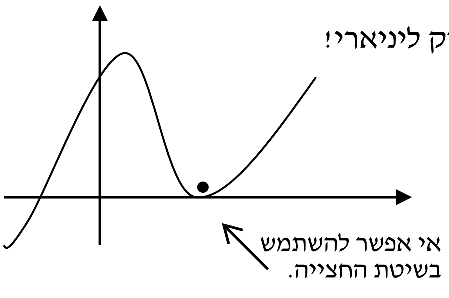
$$x_{n+1}-x*=x_n-x*-\frac{(x_n-x*)\cdot g(x_n)}{k\cdot g(x_n)+(x_n-x*)\cdot g'(x_n)}$$

$$\left|x_{n+1}-x*\right|\le\left|1-\frac{1}{k}\right|\cdot\left|x_n-x*\right|$$

סדר $=1$, $k=1$. קצב $q<1\Leftarrow 1-\frac{1}{k}=$

אלגוריתם ניוטון רפסון לא טוב כ"כ, כי הוא רק ליניארי!

גרף של פונקציה עם שורש מריבוי 2 :



$x^2-\varepsilon=0\Rightarrow x_{n+1}=x_n-\frac{x_n^2-\varepsilon}{2x_n}=\frac{1}{2}\left(x_n+\frac{\varepsilon}{x_n}\right)$ $\rightarrow$ זה לא ליניארי!

$$x_n=c_1\lambda_1^n+c_2\lambda_2^n\Leftrightarrow x_{n+1}=ax_n+b\Rightarrow x_n=c\cdot\lambda_n$$

$$c\lambda^{n+1}=a\cdot c\cdot\lambda^n+b\cdot c\cdot\lambda^{n-1}\Rightarrow\lambda^2=a\lambda+b$$

אלגוריתם זה מתכנס אם $|\lambda|<1$. במקרה זה נקודת השבת תהיה 0, והיא יציבה אם יש שני שורשים.

איכות השיטה : כיצד נשווה בין אלגוריתמים שונים?

1. כלליות *universality*
2. דיוק *accuracy*
3. קריטריונים נוספים : א. מהירות
ב. סיבוכיו
ג. חיסכון

לדוגמא, *N-R* יותר כללי מ- *Heron*, אבל *Heron* יותר טוב מבחינת דיוק: תחום ג'וליה הוא כמעט כל הציר! למרות לזות *N-R* הוא איכותי בהרבה מכיוון לכלליותו.

בשימוש בשיטה *N-R* נדרש קירוב התחלתי טוב (ע"מ להבטיח התכנסות) ולכן מבחינה מעשית, נהוג להשתמש בה רק לאחר שמקבלים הערכה ראשונית ע"י שימוש בשיטות מסדר ראשון, למשל שיטת החצייה שהיא בטוחה אך לא מהירה.

ע"י הקריטריונים הללו ניתן להשוות בין אלגוריתמים שונים ואז ניתן להתכנס במהירות יחסית יותר גבוהה לפתרון המבוקש.

שיפור התכנסות

שיטת איטקן להאצת קצב התכנסות (Aitken's delta-squared process)

ננוח שאלגוריתם איטרטיבי הוא לא טוב כ"כ, כי הוא ליניארי בסביבה נקודה שבת $x \approx x^*$

$$\varepsilon_{n+1} = \Delta_{n+1,x^*} = \left|x_{n+1} - x^*\right| = \left|\varphi(x_n) - \varphi(x^*)\right| \underset{\forall x^{**}\in[x_n,x^*]}{=} \left|\varphi'(x^{**})\right| \cdot \left|x_n - x^*\right| \approx$$

$$\approx C\left|(x_n - x^*)\right| = C\varepsilon_n, \quad 0 < C < 1.$$

ועבור n מספיק גדול,

$$\frac{\varepsilon_{n+1}}{\varepsilon_n} \approx \frac{\varepsilon_n}{\varepsilon_{n-1}} \approx C \Rightarrow$$

$$\Rightarrow \frac{x_{n+1} - x^*}{x_n - x^*} \approx \frac{x_n - x^*}{x_{n-1} - x^*} \Rightarrow \exists y_n, \quad \Rightarrow$$

$$\frac{x_{n+1} - y_{n+1}}{x_n - y_{n+1}} \approx \frac{x_n - y_{n+1}}{x_{n-1} - y_{n+1}} \Rightarrow y_{n+1} = \frac{x_{n-1}x_{n+1} - x_n^2}{x_{n+1} - 2x_n + x_{n-1}} \Rightarrow$$

$$\boxed{y_{n+1} = x_{n+1} - \frac{(x_{n+1} - x_n)^2}{x_{n+1} - 2x_n + x_{n-1}}}$$

<u>הסבר לנוסחה של איטקן:</u> עבור כל שלשה קירובי עוקבים $\{x_{n+1}, x_n, x_{n-1}\}$ של אלגוריתם ליניארי, ניתן לחשב איבר מתוקן y_{n+1} אשר בדרך כלל יהווה קירוב טוב יותר לעומת תהליך x_n.

<u>דוגמא:</u>

המהירות ההתכנסות בכל 4 אלגוריתמים הנ"ל היא ליניארית :

$$\left.\begin{array}{ll} (1) & x_n = \dfrac{1}{n} \to 0, \quad \dfrac{x_{n+1}}{x_n} = \dfrac{n}{n+1} \approx 1 \\ (2) & x_n = \dfrac{1}{n^2} \to 0, \quad \dfrac{x_{n+1}}{x_n} = \left(\dfrac{n}{n+1}\right)^2 \approx 1 \\ (3) & x_n = \dfrac{1}{n!} \to 0, \quad \dfrac{x_{n+1}}{x_n} = \dfrac{1}{n+1} \approx 0 \\ (4) & x_n = \dfrac{1}{2^n} \to 0, \quad \dfrac{x_{n+1}}{x_n} = \dfrac{1}{2} \end{array}\right\} \qquad y_{n+1} = x_{n+1} - \frac{(x_{n+1}-x_n)^2}{x_{n+1}-2x_n+x_{n-1}}$$

ואז מקבלים באופן שקול

$$(1) \qquad y_n = \frac{1}{2(n-1)} \sim \frac{1}{2n}$$

$$(2) \qquad y_n = \frac{2n^2-4n+1}{(n-1)^2(6n^4-24n^3+34n^2-20n+4)} \sim \frac{1}{3n^4}$$

$$(3) \qquad y_n = \frac{1}{(n^2-3n+1)(n-1)!} \sim \frac{1}{(n+1)!}$$

$$(4) \qquad y_n = 0$$

שיטת איטקן Δ^2 מאיצה את קצב ההתכנסות ואותה רמת דיוק מושגת לאחר פחות איטרציות.

רקורסיה ליניארי

יחס רקורסיה ליניארי הוא - $f(n) = c_1\, f(n-1) + c_2\, f(n-2) + \ldots + c_k\, f(n-k)$

הפונקציה $f(n)$ המקיימת **יחס רקורסיה ליניארי** עבור כל n שלם אי שלילי, נקרת **פתרון הרקורסיה** למציאת נוסחה מפורשת עבור $f(n)$, מציבים $f(n) = \lambda^n$ ל יחס רקורסיה ונבדוק, מתי הפונקציה $f(n) = \lambda^n$ מקיימת תנאי רקורסיה. מקבלים :

$$\lambda^n = c_1\, \lambda^{n-1} + c_2\, \lambda^{n-2} + \ldots + c_k\, \lambda^{n-k}$$

נצמצם ב- λ^{n-k} ונקבל משווה אופייני הבא :

$$\lambda^k - c_1\, \lambda^{k-1} - c_2\, \lambda^{k-2} - \ldots - c_k = 0$$

נניח ש $\lambda_1, \lambda_2, \ldots, \lambda_k$ הם שורשים פשוטים של משווה אופייני אז

$$f(n) = a_1\, \lambda_1^n + a_2\, \lambda_2^n + + \ldots + a_k\, \lambda_k^n$$

פקודה בMAPLE לפתרון רקורסיה ליניארי היא $rsolve()$
פקודות בMAPLE לפתרון המשוואה $f(x) = 0$ הן *fsolve(), fzero(),*