Numerical Methods

College 3: Nulpunten van Functies
(Hoofdstuk 3)

A.A.N. Ridder

Department EOR
Vrije Universiteit Amsterdam

Huispagina:
`http://personal.vu.nl/a.a.n.ridder/numprog/default.htm`

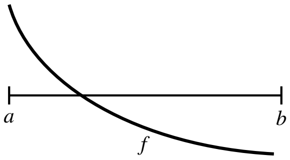
14 september 2015

Onderwerpen

Het Univariate Nulpuntprobleem

Uit de Analyse

- Gegeven $f : \mathbb{R} \to \mathbb{R}$, continu.
- Veronderstel dat $a < b$ bestaan met $f(a)f(b) < 0$.
- Pas tussenwaardestelling van continue functies toe: er is een $x \in (a, b)$ met $f(x) = 0$.



Tussenwaardestelling

Tussenwaardestelling

Zij f een continue reëelwaardige functie op het interval $[a,b]$ en γ een getal tussen $f(a)$ en $f(b)$, dus

$$f(a) \leq \gamma \leq f(b) \quad \text{indien} \quad f(a) \leq f(b);$$

of

$$f(b) \leq \gamma \leq f(a) \quad \text{indien} \quad f(b) \leq f(a).$$

Dan bestaat er een $c \in [a,b]$ met $f(c) = \gamma$.

In het speciale geval dat $\gamma = 0$ heet dit de stelling van Bolzano.

Numeriek Nulpuntprobleem

- In een EOR probleem wordt de functie vaak gegeven via vergelijkingen of modellen of data.
- We zijn geïnteresseerd in een getalsmatige (numerieke) oplossing.
- Bv, zie opdracht 1, maximum likelihood wordt vaak opgelost door een nulpunt van de afgeleide.
- Zie ook de vakken OR 1 en ME 1.

Iteratieve Methoden

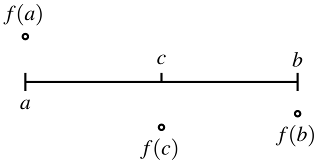
- Bijna alle numerieke *root finding* methoden benaderen een nulpunt x door een rij $(x_n)_{n=0}^{\infty}$;
- De rij wordt iteratief/recursief geconstrueerd;
- Dwz, x_n is een functie van $x_0, x_1, \ldots, x_{n-1}$.
- De meest gangbare methoden zijn:
  - *(A).* Bisectie
  - *(B).* Newton's methode (Newton-Raphson)
  - *(C).* Secant methode
  - *(D).* False position (Regula falsi)

§3.1 Bisectie

Bisectie in Woorden

- Je weet dat er een nulpunt is in het interval $[a, b]$.
- Dan is noodzakelijk dat een nulpunt in de linkerhelft $[a, (a+b)/2]$ is of in de rechterhelft $[(a+b)/2, b]$.
- Veronderstel dat je kunt beredeneren dat een nulpunt in de linkerhelft zit.
- Noem dat “halve” interval weer $[a, b]$.
- Redeneer hetzelfde vanaf het begin.
- Herhaal zolang totdat $a \approx b$, dwz $b - a < \epsilon$.
- Bijvoorbeeld $\epsilon = 10^{-8}$.

Bisectie in Figuur



- Middelpunt van $[a, b]$ is $c = (a + b)/2$.
- Je weet dat er een nulpunt ligt tussen a en c als $f(a)f(c) < 0$.

Bisectie in Matlab

Zelf programmeren.

```
function [c,fc,n] = bisectie(a0,b0,nmax,itol,ftol)
   a = a0; b = b0;
   fa = funf(a); fb = funf(b);
   if fa*fb>0
     error('verkeerde start');
   end

   n = 1; c = (a+b)/2;
   fc = funf(c); breedte = b-a;
   while n<nmax && breedte>itol && abs(fc)>ftol
     if fa*fc<0
          b = c;
     else
          a = c;
          fa = fc;
     end
     n = n+1; c = (a+b)/2;
     fc = funf(c); breedte = b-a;
   end
end
```

Toelichting van de Matlab Code

De inputargumenten zijn

- `a0,b0` is het inklemmingsinterval waarbinnen je een nulpunt denkt te weten.
- `nmax` is het maximaal aantal iteraties. Als deze bereikt wordt, stopt de uitvoering.
- `itol`: als het de breedte van het inklemmingsinterval onder deze waarde komt, stopt de uivoering.
- `ftol`: als het de absolute waarde van de functie onder deze waarde komt, stopt de uivoering.

De functie waarvan een nulpunt gezocht wordt heet in de code `funf`.

In het volledige programma wordt die als een afzonderdlijk Matlab `function` gecodeerd.

Zie het voorbeeld op slide 16.

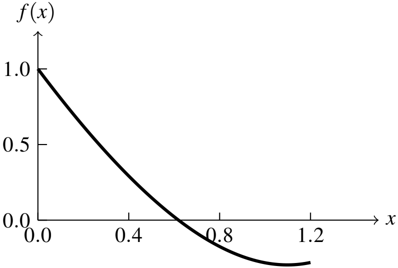
Stoppen van de Bisectie-Iteratie

Als aan minstens één van de volgende criteria is voldaan.

- $b - a < \epsilon$ (zelf gekozen);
- $|f(c)| < \epsilon$ (zelf gekozen);
- Aantal iteraties $> M$ (zelf gekozen).

Voorbeeld

- Nulpunten van $f(x) = e^x - 3x$ [zie opgave 1 van Problems 3.1].
- Eerst een inklemmingsinterval vinden.
- $f(0) = 1 > 0$;
- $f'(x) = e^x - 3 > 0$ voor $x > \ln(3) \approx 1.1$ en $f'(x) < 0$ voor $x < \ln(3)$.
- $f(\ln(3)) = 3 - 3\ln(3) < 0$.
- Concludeer: er is een (uniek!) nulpunt in $[0, \ln(3)]$.



Voorbeeld (vervolg)

- Zet $a = 0, b = 1.2$. Dan $c = 0.6$ met $f(c) = e^{0.6} - 1.8 \approx 0.0221 > 0$.
- Nulpunt zit in $[a = 0.6, 1.2 = b]$.
- $c = (a + b)/2 = 0.9; f(c) = e^{0.9} - 2.9 \approx -0.2404 < 0$.
- Nulpunt zit in $[a = 0.6, 0.9 = b]$.
- $c = (a + b)/2 = 0.75; f(c) = e^{0.75} - 2.25 \approx -0.1330 < 0$.
- Nulpunt zit in $[a = 0.6, 0.75 = b]$.
- Etc.

Matlab Programma van Voorbeeld

```
function bisvb
   start;

   function start
      a0 = 0; b0 = 1.2;
      nmax = 100; itol = 1.0e-6; ftol = 1.0e-10;
      [x,fx,n] = bisectie(a0,b0,nmax,tol);
      fprintf('gevonden nulpunt x = %.6f\n', x);
      fprintf('functiewaarde in x = %.6e\n', fx);
      fprintf('aantal iteraties   = %d\n', n);
   end

   function y = funf(x)
      y = exp(x) - 3*x;
   end

   function [c,fc,n] = bisectie(a0,b0,nmax,tol)
      % code gegeven op slide 11
   end
end
```

Ouput

```
gevonden nulpunt x = 0.619062
functiewaarde in x = -3.184608e-07
aantal iteraties   = 22
```

Analyse Bisectie

Veronderstel $r \in [a_0, b_0]$ is een nulpunt.

(a). Iteratieve methode;

(b). In n-de iteratie is $[a_n, b_n]$ een inklemmingsinterval (klemt een nulpunt in);

(c). Lengte interval

$$b_n - a_n = \tfrac{1}{2}\,(b_{n-1} - a_{n-1}) = \cdots = (\tfrac{1}{2})^n\,(b_0 - a_0) \to 0$$

(d). Noem $x_n = (a_n + b_n)/2$ de n-de benadering van nulpunt r;

(e). Omdat voor alle n

   (i). $a_n < x_n < b_n$ (zie (d).);

   (ii). $a_n < r < b_n$ (zie (b).);

   (iii). $b_n - a_n \to 0$ (zie (c).);

   geldt $\lim_{n\to\infty} x_n = r$.

(f). Fouten $e_n = x_n - r$ voldoen aan $|e_n| \approx \frac{1}{2}|e_{n-1}|$; dat wil zeggen lineaire order van convergentie.

Aantal Benodigde Iteraties

- Zie Theorem 1 blz 119-120;
- Uit
$$|e_n| \leq \tfrac{1}{2}(b_n - a_n), \quad b_n - a_n = \tfrac{1}{2}(b_{n-1} - a_{n-1})$$
Volgt
$$|e_n| \leq (b_0 - a_0)/2^{n+1}.$$
- Dus voldoende voor $|e_n| < \epsilon$ is
$$n = \left\lfloor {}^2\log \frac{b-a}{\epsilon} \right\rfloor.$$

Discussie Bisectie Methode

Voordelen:

- Eenvoudige methode;
- Inklemming van nulpunt;
- Controle over inklemmingsinterval;
- Gegarandeerde convergentie;
- Alleen expressie van de functie nodig (niet die van de afgeleide);
- In volgende iteratie is slechts één nieuwe functie-evaluatie nodig;
- Foutschatting mogelijk;

Nadelen:

- Twee startwaarden nodig die voldoen aan $f(a)f(b) < 0$;
- Lineaire convergentie (zie slide 18 en boek pag 120);

§3.2 Newton's Methode

Formule

Heet ook wel Newton-Raphson.

Aanname

f is continu differentieerbaar (in omgeving van nulpunt x) en $f'(x) \neq 0$.

Iteratieformule

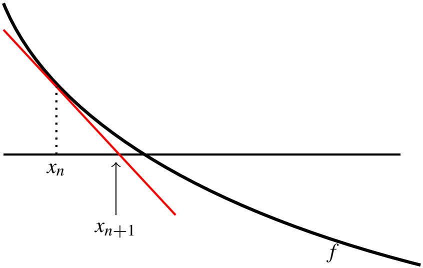
x_0 'willekeurig'; dan voor $n = 0, 1, \ldots$

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

Afleiding Newton's Iteratieformule I

Eerste manier: geometrisch.

- x_{n+1} is snijpunt van de raaklijn aan grafiek f in $(x_n, f(x_n))$ met x-as.
- Werk uit met calculus.



Afleiding Newton's Iteratieformule II

Tweede manier: analytisch. Zie college 2 van OR I.

- Lineaire Taylor benadering van $f(x_n + h)$;
- Noem $x_{n+1}^{\text{exact}} = x_n + h^{\text{exact}}$ waarbij h^{exact} zodat $f(x_{n+1}^{\text{exact}}) = 0$:

$$0 = f(x_{n+1}^{\text{exact}}) = f(x_n + h^{\text{exact}}) = f(x_n) + h^{\text{exact}} f'(\xi_n)$$

$$\Rightarrow \quad h^{\text{exact}} = -f(x_n)/f'(\xi_n).$$

- Benadering: $h = -f(x_n)/f'(x_n)$ geeft x_{n+1} zoals in interatieformule waarvoor $f(x_{n+1}) \approx 0$.

Convergentie-analyse Newton's Methode

- Zie Theorem 1 blz 129-130;
- Veronderstel dat ook f'' bestaat in omgeving van nulpunt r.
- Fouten $e_n = x_n - r$ voldoen aan (weer Taylor toepassen)

$$0 = f(x) = f(x_n - e_n) = f(x_n) - e_n f'(x_n) + \tfrac{1}{2} e_n^2 f''(\xi_n).$$

- Deel door $f'(x_n) \neq 0$; dan eerste twee termen naar links:

$$\underbrace{e_n - \frac{f(x_n)}{f'(x_n)}}_{=e_{n+1}} = \frac{1}{2} e_n^2 \frac{f''(\xi_n)}{f'(x_n)}.$$

- Conclusie: kwadratische convergentie $|e_{n+1}| \approx C|e_n|^2$.

Schatting Fout

Lemma

$$e_n \approx x_n - x_{n+1}.$$

Bewijs:

- Gebruik weer $f(x_n + h) \approx f(x_n) + hf'(x_n)$.
- Namelijk

$$0 = f(x) = f(x_n - e_n) \approx f(x_n) - e_n f'(x_n) \;\Rightarrow\; e_n \approx \frac{f(x_n)}{f'(x_n)}.$$

- Iteratieformule: $x_{n+1} = x_n - f(x_n)/f'(x_n)$,

$$\Rightarrow\; x_n - x_{n+1} = \frac{f(x_n)}{f'(x_n)} \approx e_n.$$

Discussie Newton's Methode

Voordelen:

- Als convergentie, dan snelle convergentie (kwadratisch);
- Foutschatting mogelijk;
- Slechts één startwaarde nodig, zonder restrictie (maar zie bij nadelen).

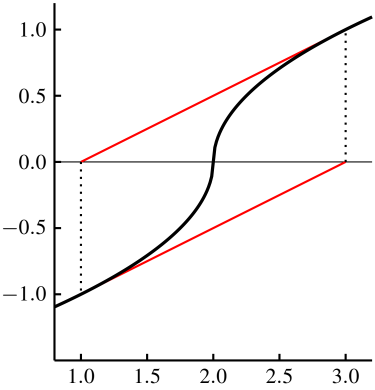
Nadelen:

- Afgeleide nodig;
- Volgende iteratie heeft twee nieuwe functie-evaluaties nodig ($f(x_n)$ en $f'(x_n)$);
- Geen inklemming;
- Geen gegarandeerde convergentie: hangt van startwaarde af [zie college 2 van OR 1, en volgende slide]

Tegenvoorbeeld

$$f(x) = \begin{cases} \sqrt{x-2}, & x \geq 2, \\ -\sqrt{2-x}, & x \leq 2. \end{cases}$$

$$f'(x) = \begin{cases} 1/2\,\sqrt{x-2}, & x > 2, \\ 1/2\,\sqrt{2-x}, & x < 2. \end{cases}$$



Dan $x_0 = 1$ geeft

$$x_1 = 1 - \frac{-1}{1/2} = 3,\ x_2 = 3 - \frac{1}{1/2} = 1, \ldots.$$

Het Goede Nieuws over Newton's Methode

Theorem 1

Zij $f : \mathbb{R} \to \mathbb{R}$ waarvoor

- r is een nulpunt, dwz $f(r) = 0$;
- f is continu differentieerbaar in omgeving van r;
- $f'(x) \neq 0$;
- f'' bestaat in omgeving van r.

Construeer de rij $(x_n)_n$ volgens Newton's iteratieformule.

Dan zijn er $\delta > 0$ en $C < \infty$ zodat als $|x_0 - r| \leq \delta$, geldt

- $\lim_{n\to\infty} x_n = x$,
- voor alle n: $|e_n| = |x_n - x| \leq \delta$,
- voor alle n: $|e_{n+1}| \leq C|e_n|^2$.

Bewijs: blz 129-130.

§3.3 Secant Methode

Formule

Afgeleide-vrije benadering van Newton's methode.

Iteratieformule

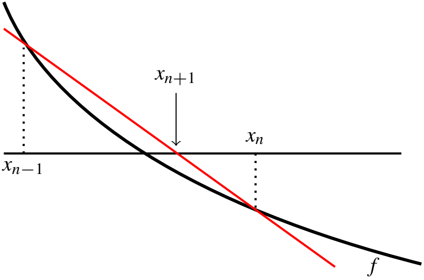
x_0 en x_1 'willekeurig'; dan voor $n = 1, 2, \ldots$

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

Afleiding Iteratieformule I

Geometrisch:

- x_{n+1} is snijpunt van de (secant) lijn door $(x_{n-1}, f(x_{n-1}))$ en $(x_n, f(x_n))$ met x-as.
- Werk uit met calculus.



Afleiding Iteratieformule II

Analytisch.

Benader de differentiaal in Newton's iteratieformule door een differentieverschil.

NB: tekens van $f(x_{n-1})$ en $f(x_n)$ hoeven niet veschillend te zijn.

Dwz $f(x_{n-1})f(x_n) > 0$ is mogelijk (zie bv de volgende iteratie van de illustratie vorige slide).

Het Goede Nieuws over Secant Methode

Stelling

Zij $f : \mathbb{R} \to \mathbb{R}$ waarvoor

- r is een nulpunt, dwz $g(r) = 0$;
- f is continu differentieerbaar in omgeving van r;
- $f'(x) \neq 0$;
- f'' bestaat in omgeving van r.

Construeer de rij $(x_n)_n$ volgens de secant iteratieformule.

Dan zijn er $\delta > 0$ en $C < \infty$ zodat als $|x_0 - r| \leq \delta$ en $|x_1 - r| \leq \delta$, geldt

- $\lim_{n\to\infty} x_n = r$,
- voor alle n: $|e_n| = |x_n - r| \leq \delta$,
- voor alle n: $|e_{n+1}| \leq C|e_n|^{\alpha}$,
- waar $\alpha = \frac{1}{2}(1 + \sqrt{5}) \approx 1.62$ de gulden snede ratio.

Bewijs: niet in dit college.

Discussie Secant Methode

Voordelen:

- Als convergentie, dan superlineair;
- Foutschatting mogelijk: $e_n \approx x_n - x_{n+1}$ analoog Newton;
- Alleen expressie van de functie nodig (niet die van de afgeleide);
- In volgende iteratie is slechts één functie-evaluatie nodig;
- Twee startwaarden nodig zonder restrictie (maar zie bij nadelen);

Nadelen:

- Geen inklemming;
- Geen gegarandeerde convergentie: hangt van startwaarden af;

Illustratie Newton en Secant

$\sqrt{2} = 1.414213562373095\cdots$ is nulpunt van $f(x) := x^2 - 2$.

- Newton:

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

- Secant:

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

Numerieke Resultaten

Achtereenvolgende benaderingen bij dezelfde start.
De onderstreepte cijfers zijn correct.
Newton convergeert kwadratisch: verdubbeling aantal correcte decimalen per iteratie!

n	Newton	secant
0	25.000000000000000	25.000000000000000
1	12.539999999999999	12.539999999999999
2	6.349744816586922	8.404368673415025
3	3.332359052069724	5.127429947360361
4	1.966267236985128	3.332359052069725
5	$\underline{1.4}$91711486846459	2.256136364657026
6	$\underline{1.41}$6226662208632	1.703187661025851
7	$\underline{1.41421}$4993136636	$\underline{1.4}$75661900863967
8	$\underline{1.414213562373}$819	$\underline{1.41}$9799538646396
9	$\underline{1.414213562373095}$	$\underline{1.414}$332109612634
10	$\underline{1.414213562373095}$	$\underline{1.414213}$796025638

Fouten bij Newton Methode

Illustratie van $e_n \approx x_n - x_{n+1}$ en van $|e_n| \approx C|e_{n-1}|^2$.

n	fout e_n	$x_n - x_{n+1}$	$\lvert e_n/e_{n-1}^2 \rvert$
0	2.3586e+001	1.2460e+001	
1	1.1126e+001	6.1903e+000	0.020000
2	4.9355e+000	3.0174e+000	0.039872
3	1.9181e+000	1.3661e+000	0.078743
4	5.5205e-001	4.7456e-001	0.150044
5	7.7498e-002	7.5485e-002	0.254289
6	2.0131e-003	2.0117e-003	0.335185
7	1.4308e-006	1.4308e-006	0.353051
8	7.2364e-013	7.2387e-013	0.353500
9	-2.2204e-016	-2.2204e-016	
10	-0.0000e+000	2.2204e-016	

Fouten bij Secant Methode

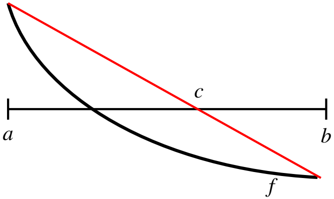
Illustratie van $e_n \approx x_n - x_{n+1}$ en van $|e_n| \approx C|e_{n-1}|^{\alpha}$.

n	fout e_n	$x_n - x_{n+1}$	$\lvert e_n/e_{n-1}^{\alpha}\rvert$
0	2.3586e+001	1.2460e+001	
1	1.1126e+001	4.1356e+000	0.066886
2	6.9902e+000	3.2769e+000	0.141739
3	3.7132e+000	1.7951e+000	0.159714
4	1.9181e+000	1.0762e+000	0.229618
5	8.4192e-001	5.5295e-001	0.293468
6	2.8897e-001	2.2753e-001	0.381743
7	6.1448e-002	5.5862e-002	0.457993
8	5.5860e-003	5.4674e-003	0.509719
9	1.1855e-004	1.1831e-004	0.523795
10	2.3365e-007	2.3364e-007	0.526196

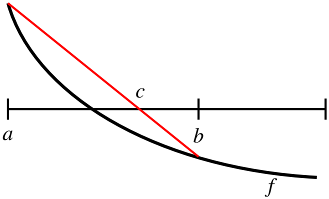
False Position Methode (Regula Falsi)

Illustratie

Combineert secant methode met een inklemming à la bisectie.



Eerste iteratie.



Tweede iteratie.

Algoritme False Position

1. Uitgangspunt: $a < b$ en $f(a)f(b) < 0$;

2. Bepaal het snijpunt c van de (secant) lijn door $(a, f(a))$ en $(b, f(b))$ met de x-as:

$$c = \frac{af(b) - bf(a)}{f(b) - f(a)}.$$

3. Als $f(a)f(c) < 0$:

–> Er is een nulpunt in (a, c).

–> Zet $b = c$ en herhaal vanaf 1.

Anders (dwz $f(c)f(b) < 0$):

–> Er is een nulpunt in (c, b).

–> Zet $a = c$ en herhaal vanaf 1.

4. Stop als minstens een van de volgende:

–> $b - a < \epsilon$ (zelf gekozen);

–> $|f(c)| < \epsilon$ (zelf gekozen);

–> Aantal iteraties $> M$ (zelf gekozen).

Discussie False Position

Voordelen:

- Inklemming van nulpunt;
- Gegarandeerde convergentie;
- Alleen expressie van de functie nodig (niet die van de afgeleide);
- In volgende iteratie is slechts één nieuwe functie-evaluatie nodig;

Nadelen:

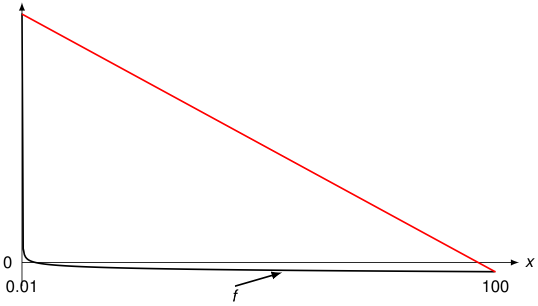
- Geen controle over inklemmingsinterval;
- Geen foutschatting mogelijk;
- Twee startwaarden nodig die voldoen aan $f(a)f(b) < 0$;
- Lineaire convergentie (zie slide 18 en boek pag 120);

Bisectie versus False Position

Zij

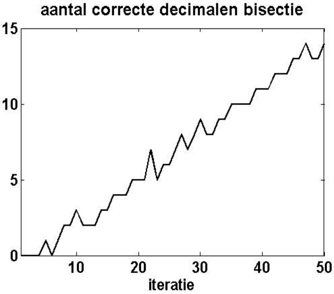
$$f(x) = \frac{1}{x} - \log x + \log 2, \quad x > 0.$$

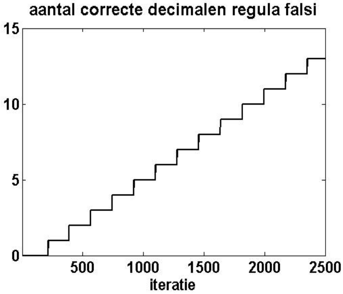
Deze functie heeft één nulpunt r in $[0.01, 100]$.



Bisectie versus False Position (vervolg)

- We beschouwen de bisectiemethode en de false posion methode om r te benaderen door een rij $(x_n)_{n=0}^{\infty}$.
- Zie onderstaande grafieken.
- Verklaar de langzamere maar monotone convergentie van de methode van false position.
- Waarom hoeft bisectie niet monotoon te convergeren?





Modificatie van False Position Methode

- Als in de false position methode een eindpunt voor een tweede keer niet zou veranderen, dan wordt de f waarde in dat punt gehalveerd;
- Heet ook wel de Illinois methode;
- Voordelen: inklemming en superlineaire convergentie

$$|e_{n+1}| \approx C|e_n|^{1.44}$$

Resultaten Illinois Methode

Voorbeeld van slide 44:

n	x_n	fout e_n	$x_n - x_{n+1}$	$\lvert e_n/e_{n-1}^{1.44}\rvert$
1	96.427087324236155	9.3584e+001	3.4139e+000	
2	93.013146539840818	9.0170e+001	6.3050e+000	0.130777
3	86.708135877284860	8.3865e+001	1.0830e+001	0.128319
4	75.877896034939525	7.3035e+001	1.6375e+001	0.124043
5	59.502996070464718	5.6660e+001	2.0172e+001	0.117434
6	39.331425560907959	3.6488e+001	1.8599e+001	0.109003
7	20.732213136355625	1.7889e+001	1.2059e+001	0.100713
8	8.672980740925043	5.8299e+000	5.3855e+000	0.091608
9	3.287463056474262	4.4440e-001	1.0459e+000	0.035093
10	2.241552587828692	-6.0151e-001	-6.6176e-001	1.933935
11	2.903311464733327	6.0252e-002	5.1914e-002	0.125274
12	2.851397189362043	8.3373e-003	1.4193e-002	0.476290
13	2.837203797826053	-5.8561e-003	-5.8669e-003	5.772030
14	2.843070690940083	1.0819e-005	1.0805e-005	0.017735
15	2.843059885813215	1.4047e-008	2.8057e-008	0.198767
16	2.843059857755793	-1.4010e-008	-1.4010e-008	
17	2.843059871766233	-4.4409e-016	-4.4409e-016	

Stelsels van Niet-lineaire Vergelijkingen

Probleemschets

- Gegeven $\boldsymbol{F} : \mathbb{R}^n \to \mathbb{R}^m$ met $n > 1$.
- Los op
$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}.$$
- Beperking: $m = n$, dwz, evenveel vergelijkingen als variabelen.
- Bijvoorbeeld $m = n = 3$ en
$$\begin{aligned} f_1(x_1, x_2, x_3) &= x_1^2 x_3 - 2x_1 x_3^3 - x_2 + 0.5 = 0 \\ f_2(x_1, x_2, x_3) &= x_1^2 + 4x_2^2 - 5x_2 x_3 + 6 = 0 \\ f_3(x_1, x_2, x_3) &= x_2 x_3 + x_1^2 x_2^2 + 8x_1 x_2 x_3 - 12 = 0 \end{aligned}$$
- Newton's methode kan hier toegepast worden, de multi-dimensionale generalizatie. Zie boek, pg 132-134, en zie college 2 van OR 1.

MATLAB

Beschikbare Functies

- `fzero`: nulpunt van continue univariate functie
- `fsolve`: oplossing van een stelsel (niet-lineaire) vergelijkingen

Codes zijn combinaties van geavanceerde algoritmes. Bv, `fzero` combineert bisectie, secant en invers kwadratische interpolatie (niet behandeld).

Zie de code van deze Matlab procedures door in het commando scherm te tikken `type fzero` respectievelijk `type fsolve`

Voorbeeld van `fsolve`: beschouw de $\boldsymbol{F} : \mathbb{R}^3 \to \mathbb{R}^3$ op slide 49.

Matlab Code

```
function nonlineqsvb

   start;

   function start
      x0 = [0;0;0]; % gok een oplossing
      x = fsolve(@(x)F(x),x0);
      fprintf('gevonden nulpunt x =\n');
      fprintf('%f\n',x);
      fprintf('\nF(x) =\n');
      fprintf('%e\n', F(x));
   end

   function y = F(x)
      y = zeros(3,1); % kolomvector
      y(1) = x(1)^2 * x(3) - 2*x(1)*x(3)^3 ...
                 - x(2) + 0.5;
      y(2) = x(1)^2 + 4*x(2)^2 - 5*x(2)*x(3) + 6;
      y(3) = x(2)*x(3) + x(1)^2 * x(2)^2 ...
                 + 8*x(1)*x(2)*x(3) - 12;
   end
end
```

Output

```
gevonden nulpunt x =
-0.035603
4.408004
3.798693

F(x) =
1.855893e-11
1.492140e-12
-1.246292e-11
```

Maar Let Op

- Veronderstel $|f'(r)| \approx 0$ in het nulpunt r.
- Oftewel, de functie is bijna vlak in de buurt van nulpunt.
- Stopcriterium $|f'(x)| < \epsilon$ kan grote relatieve fout geven.
- Voorbeeld: $f(x) = x^3 + \delta x$ met $\delta = 10^{-8}$, met nulpunt $r = 0$.
- Wegens stopcriterium $|f(x)| < 10^{-10}$ stoppen bv de bisectie en Newton procedures 'te vroeg'.

```
bisectie met start in [-0.2,0.5]    newton met start in 0.5
gevonden nulpunt x = -0.000391      gevonden nulpunt x = 0.000330
functiewaarde in x = -6.351089e-11  functiewaarde in x = 3.938954e-11
aantal iteraties   = 8              aantal iteraties   = 19
```

- Idem meerdimensionaal als $|\nabla F(r)| \approx 0$.