Iterativ beregning af Rodapproximationer.

Jacob Nielsen

I det følgende forklares med udgangspunkt i binomialformlen algoritmer til beregning af approxomationer til kvadratrødder og kubikrødder. Grund ideen bag beregningerne har været brugt i tidsrummet fra år 1000 og sikkert tidligere og til fremkomsten af regnemaskiner. Algoritmens styrke er, at den kan udføres alene ved brug af de fire regningsarter, og at rodapproximationerne konvergerer hurtigt mod den rigtige værdi.

Binomialformlen:

$$(a+b)^n = \sum_{r=0}^{r=n} K_{(n,r)} \cdot a^r \cdot b^{n-r}\ , \quad K_{(n,r)} = \frac{n!}{r! \cdot (n-r)!}\ , \quad 0! \equiv 1$$

$$(a+b)^2 = 1 \cdot b^2 + 2 \cdot a \cdot b + 1 \cdot a^2$$

$$(a+b)^4 = b^4 + 4 \cdot a \cdot b^4 + 6 \cdot a^2 \cdot b^2 + 4 \cdot b \cdot a^4 + a^4$$

Binomialkoefficienterne kan hurtigt findes ved hjælp af Pascals trekant[1].

Bevis for binomialformlen:

Det forudsættes bekendt, at antallet af måder at vælge r elementer ud af n er lig med K(n,r)[2].
$(a+b)^n$ består er et produkt af n identiske faktorer (a+b). Hvert led i produktet indeholder et led fra hver faktor, som kan være enten et a eller et b. Alle led har således formen $a^r\ b^{n-r}$. De r parenteser, der skal bidrage med et a kan udvælges blandt de n faktorer på K(n,r) måder.
Ser vi f.eks. på $(a+b)^4$, så er der bidrag af formen $a^2\ b^2$. De parenteser, der bidrager med faktoren a kan vælges på K(4,2) = 6 måder blandt de fire parenteser. Koefficienten til $a^2\ b^2$ er således lig med seks. De seks muligheder er vist nedenfor.

$$(a+b)\cdot(a+b)\cdot(a+b)\cdot(a+b) = a^2 b^2$$

$$(a+b)\cdot(a+b)\cdot(a+b)\cdot(a+b) = a^2 b^2$$

$$(a+b)\cdot(a+b)\cdot(a+b)\cdot(a+b) = a^2 b^2$$

$$(a+b)\cdot(a+b)\cdot(a+b)\cdot(a+b) = a^2 b^2$$

$$(a+b)\cdot(a+b)\cdot(a+b)\cdot(a+b) = a^2 b^2$$

$$(a+b)\cdot(a+b)\cdot(a+b)\cdot(a+b) = a^2 b^2$$

---

[1] Der findes f.eks. En udmærket beskrivelse af Pascals trekant i Wikipedia.

[2] Et bevis for formlen til beregning af K(n,r) findes i mange bøger om sanssynlighedsregning og kombinatorik.

Kvadratrodsapproximationer

Den sidste linje i beregningerne nedenfor viser, hvordan den næste approximation til kvadtatrodden af et tal beregnes ved at lægge korrektionen korr til den forgående rodapproximation rapp.
Næst sidste linje viser, hvordan korrektionerne beregnes.
Undervejs smides andenordensledet i korrektionen væk. Dette vil i stigende grad være en god tilnærmelse efterhånden som iterationsprocessen skrider frem. Bilag 1 er en beregning med tal, der svarer til beregningen med symboler nedenfor.

$$Tal - Rapp^2 = \mathrm{Re}\,st$$
$$(Rapp + Korr)^2 = Tal$$
$$Rapp^2 + 2Rapp \cdot Korr + Korr^2 = Tal$$
$$Rapp^2 + 2Rapp \cdot Korr \approx Tal$$
$$\Updownarrow$$
$$2Rapp \cdot Korr \approx Tal - Rapp^2 = \mathrm{Re}\,st$$
$$\Updownarrow$$
$$\boxed{Korr \approx \frac{\mathrm{Re}\,st}{2Rapp}}$$
$$Rapp_{n+1} = Rapp_n + Korr_n$$

Kubikrodsapproximationer

Formlen til beregning af korrektionen ved iterativ beregning af kubikrødder findes på samme måde ved at smide led af højere orden end en i korrektionen væk.

$$Tal - Rapp^3 = \mathrm{Re}\,st$$
$$(Rapp + Korr)^3 = Tal \;\Leftrightarrow$$
$$Rapp^3 + 3Rapp^2 \cdot Korr + 3Rapp \cdot Korr^2 + Korr^3 = Tal$$
$$Rapp^3 + 3Rapp^2 \cdot Korr \approx Tal$$
$$\Updownarrow$$
$$3Rapp^2 \cdot Korr \approx Tal - Rapp^3 = \mathrm{Re}\,st$$
$$\Updownarrow$$
$$\boxed{Korr \approx \frac{\mathrm{Re}\,st}{3Rapp^2}}$$
$$Rapp_{n+1} = Rapp_n + Korr_n$$

Bilag 2 er indeholder eksempler på beregning af kubikrødder med denne metode.

Korrektionens nøjagtighed

Nu undersøges **størrelsen af fejlen**, der begås **ved at bortkaste andenordensleddet** i korrektionen. Den eksakte værdi af korrektionen betegnes nedenfor med Korr.ex.

$$Tal - Rapp^2 = Rest$$

$$(Rapp + Korr)^2 = Tal \quad \Leftrightarrow$$

$$Rapp^2 + 2Rapp \cdot Korr + Korr^2 = Tal \quad \Leftrightarrow$$

$$2Rapp \cdot Korr.ex + Korr.ex^2 = Tal - Rapp^2 = Rest$$

****

$$Rapp^2 + 2Rapp \cdot Korr \approx Tal$$

$$\Updownarrow$$

$$2Rapp \cdot Korr \approx Tal - Rapp^2 = Rest$$

$$\frac{2Rapp \cdot Korr.ex + Korr.ex^2}{2Rapp \cdot Korr} = 1 \quad \Leftrightarrow \quad \frac{Korr.ex}{Korr} + \frac{Korr.ex^2}{2Rapp \cdot Korr} = 1 \quad \Leftrightarrow$$

$$Korr.ex + \frac{Korr.ex^2}{2Rapp} = Korr \quad \Leftrightarrow$$

$$\boxed{fejl\ i\ procent = \frac{Korr - Korr.ex}{Korr.ex} * 100\% = \frac{Korr.ex * 50}{Rapp}\%}$$

I linjen under stjernerne er andenordensleddet fra tredje ligning fra oven bortkastet. Vi får nu en rød og en blå ligning, hvor højresiderne er ens. Når ligningerne divideres med hinanden bliver højresiden således lig med en.

Betragt nu resultatet, der er indrammet. Fejlen bliver mindre og mindre efterhånden som vi kommer tættere på den korrekte værdi af kvadratroden; idet korrektionen bliver mindre og Rappe ændrer sig ikke meget. Hvis vi opnår at korrektionen er under en tiendedel af rodapproximationen, så bliver fejlen mindre end 5 %.

Kan det gå helt galt, eller virker metoden altid ? Helt galt går det kun, hvis vores korrektion bliver nul eller får det forkerte fortegn, så nærmer vi os ikke den rigtige værdi af kvadratroden. Det kræver imidlertid, at fejlen bliver større end 100 %. Det sker kun, hvis Korr.ex > 2 Rapp. Det svarer til, at vi gætter en værdi af kvadratroden, der er mindre end en tredjedel af den korrekte værdi. Metoden er således temmelig robust.

Bilag 1 *iterativ beregning af kvadratrødder*

Nedenfor beregnes en tilnærmelse til kvadratroden af 6708.

tal:=6708

rapp1:=75 (første approximation til kvadratroden)

korr.eksakt1:=$\sqrt{6708}-75$ ▸ 6.90238 (den præcise korrektion til vores gæt er 6,90238)

Nu beregnes den tilnærmede korrektion, som det gøres i iterationen.

rest1:=**tal**−**rapp1**^{2} ▸ 1083

korr1:=$\dfrac{\textbf{rest1}}{2\cdot \textbf{rapp1}}$ ▸ 7.22

rapp2:=**rapp1**+**korr1** ▸ 82.22 (næste tilnærmelse til $\sqrt{6708}$)

rest2:=**tal**−**rapp2**^{2} ▸ -52.1284

korr2:=$\dfrac{\textbf{rest2}}{2\cdot \textbf{rapp2}}$ ▸ -0.317006

rapp3:=**rapp2**+**korr2** ▸ 81.903

$\sqrt{6708}$ ▸ 81.9024

Tredje tilnærmelse til kvadratroden af 6708 stemmer overens med den rigtige værdi på de fire første betydende cifre.

Bilag 2 *iterativ beregning af kubikrødder*

	A	B n	C n.itredje	D rapp	E rest	F korr
=			='n^3			
1	Tallet	1.E0	1.E0	3.E1	7.567E3	2.80259E0
2	3.4567E4	2.E0	8.E0	3.28026E1	-7.2892...	-2.2581E-1
3		3.E0	2.7E1	3.25768E1	-5.0063...	-1.57246E-3
4		4.E0	6.4E1	3.25752E1	-2.4164...	-7.59077E-8
5		5.E0	1.25E2	3.25752E1	1.E-9	3.14126E-13

Tabellen viser beregningen af kubikroden af 34567. I søjle C opløftes de første hele tal i tredje. Vi ser at $3^3 = 27$ og dermed er $30^3 = 27\ 10^3 = 27000$. $40^3 = 64\ 10^3 = 64000$. Kubikroden af 34567 ligger derfor mellem 30 og 40. Det er baggrunden for at det første gæt, som er markeret med grønt i feltet D1 er 30.

Bemærk, at allerede efter tre iterationer er kubikroden bestemt med meget stor nøjagtighed.

I tabellen nedenfor ses, at hvis vi starter med det ret dårlige gæt 20, så kræver det kun tre iterationer mere at opnå den samme nøjagtighed. Algoritmen konvergeret hurtigt, og den er forholdsvis robust over for dårlige gæt som startværdier.

	A	B n	C n.itredje	D rapp	E rest	F korr
=			='n^3			
1	Tallet	1.E0	1.E0	2.E1	2.6567E4	2.21392E1
2	3.4567E4	2.E0	8.E0	4.21392E1	-4.02599E4	-7.55753E0
3		3.E0	2.7E1	3.45816E1	-6.78883E3	-1.89227E0
4		4.E0	6.4E1	3.26894E1	-3.64701E2	-1.13764E-1
5		5.E0	1.25E2	3.25756E1	-1.26774E0	-3.9822E-4
6		6.E0	2.16E2	3.25752E1	-1.5496E-5	-4.8677E-9