Rudinin versio Weierstrassin funktiosta

Jatkuva funktio, jolla ei ole missään derivaattta.

17.03.08 Heikki Apiola

Liittyy Markku Halmetojan kirjoitukseen Solmun numerossa 2/2008.

Apufunktioita: Jaksollinen jatko: JJ

Aivan kuten Fourier-sarjoissa, tässä on erinomaisen hyödyllinen väline operaattori, joka suorittaa jaksollisen jatkamisen.
Hieno toteutus on peräisin Maple-gurulta: *Mike Monaganilta*

```
> restart:
> with(plots):
```

Määritellään uudet koodit varmuuden vuoksi myös tässä, vaikka ovatkin fourier.mpl:ssä (ja ohjelmat.mpl:ssä).

```
> # Jaksollinen jatko: MapleTech Vol 3 No. 3 1996 (Mike Monagan)
  #
  JJ:=proc(f,d::range)
    subs({'F'=f,'L'=lhs(d),'D'=rhs(d)-lhs(d)},
    proc(x::algebraic) local y;
      y:=floor((x-L)/D);
      F(x-y*D);
    end)
  end:

  # Esim:
  #sw:=JJ(signum,-1..1);
  #plot(sw,-4..4);
  #
```

Määritellään Rudinin funktio ϕ;

```
> phi0:=x->abs(x);
```

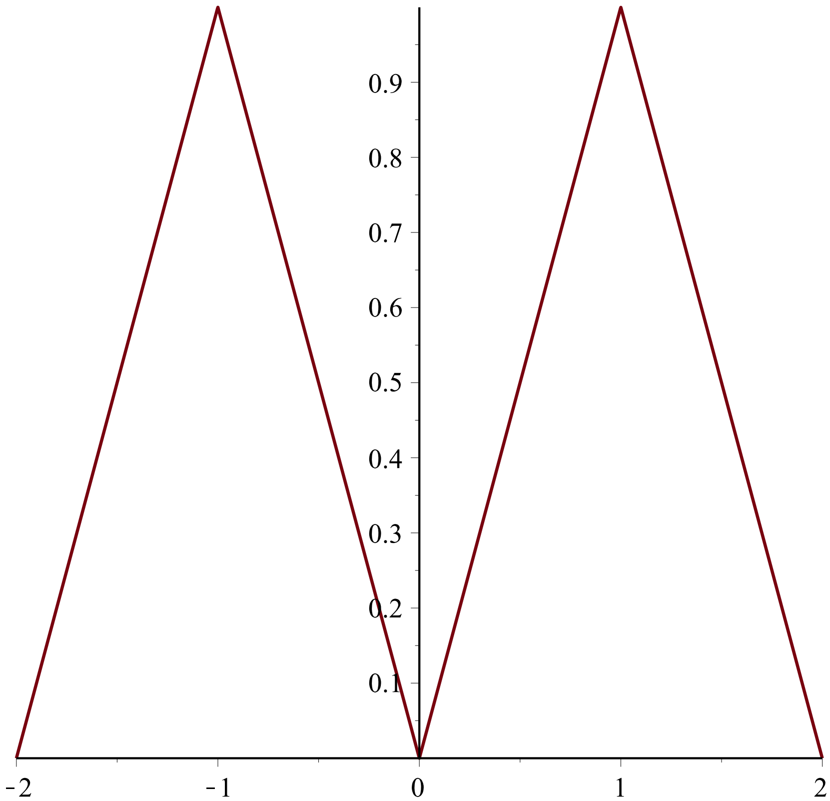
$$\phi 0 := x \to |x| \qquad \textbf{(2.1)}$$

Jatketaan jaksollisena, jaksovälinä [-1,1].

```
> phi:=JJ(phi0,-1..1);
```

$$\phi := \textbf{proc}(x::algebraic)\ \textbf{local}\ y;\ y := \text{floor}(1/2 * x + 1/2);\ phi0(x - 2 * y)\ \textbf{end proc} \qquad \textbf{(2.2)}$$

```
> plot(phi,-2..2);
```



Määritellään sarjan osasummafunktio:

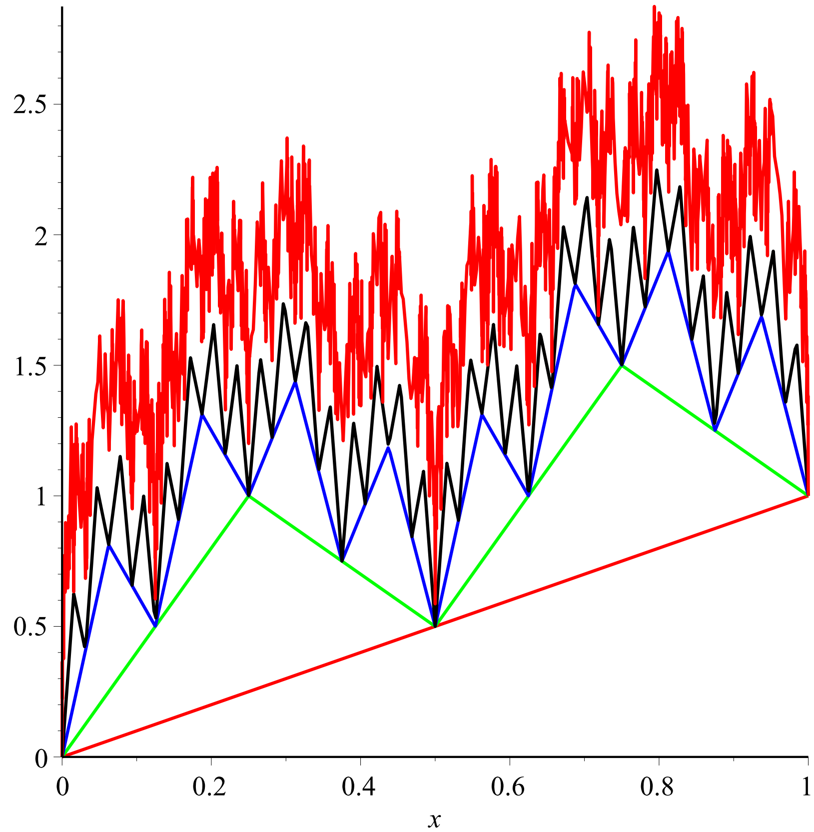
```
> osasumma:=N->sum((3/4)^k*phi(4^k*x),k=0..N);
```

$$osasumma := N \rightarrow \sum_{k=0}^{N} \left(\frac{3}{4}\right)^{k} \phi\left(4^{k} x\right) \tag{2.3}$$

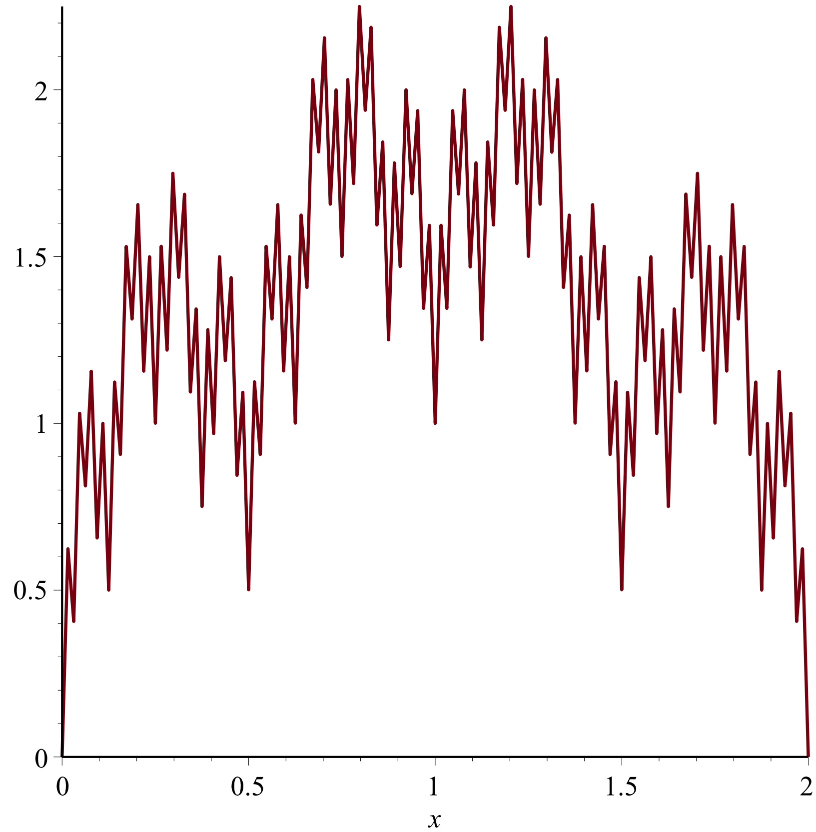
```
> osasumma(2);
```

$$\left|x - 2\,\mathrm{floor}\left(\frac{1}{2}\,x + \frac{1}{2}\right)\right| + \frac{3}{4}\left|4\,x - 2\,\mathrm{floor}\left(2\,x + \frac{1}{2}\right)\right| + \frac{9}{16}\left|16\,x - 2\,\mathrm{floor}\left(8\,x + \frac{1}{2}\right)\right| \tag{2.4}$$

```
> plot([osasumma(0),osasumma(1),osasumma(2),osasumma(3),osasumma
  (8)],x=0..1,color=[red,green,blue,black,red]);
```



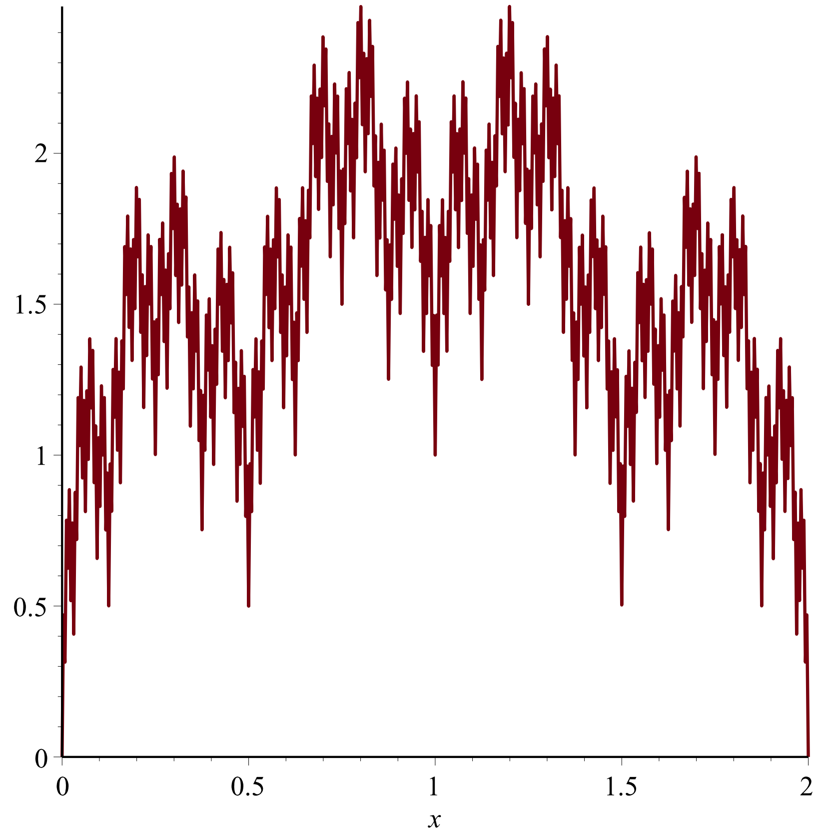

```
> plot(osasumma(3),x=0..2,numpoints=2*2*2*1800);
```




```
> 2*2*2*1800;
```

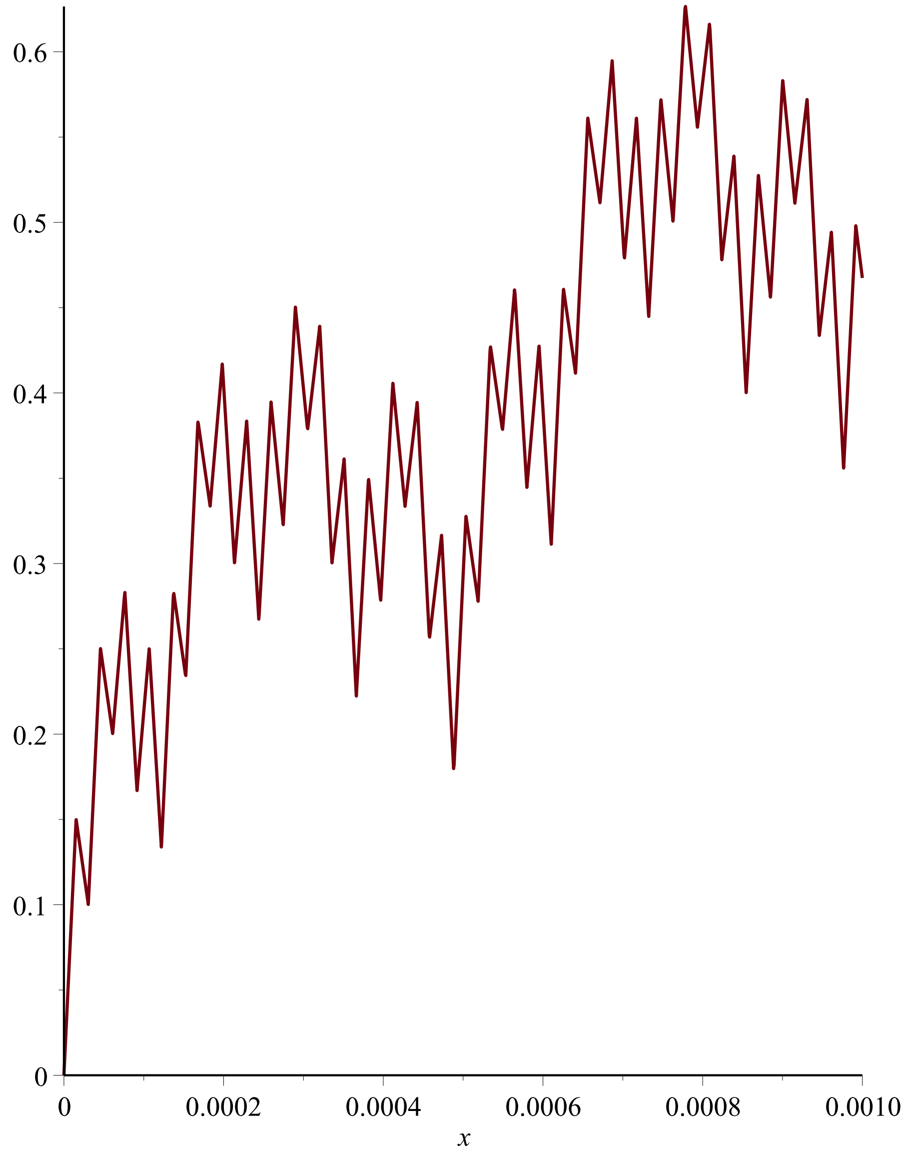
$$14400 \tag{2.5}$$

```
> plot(osasumma(4),x=0..2,numpoints=2*2*2*1800);
```



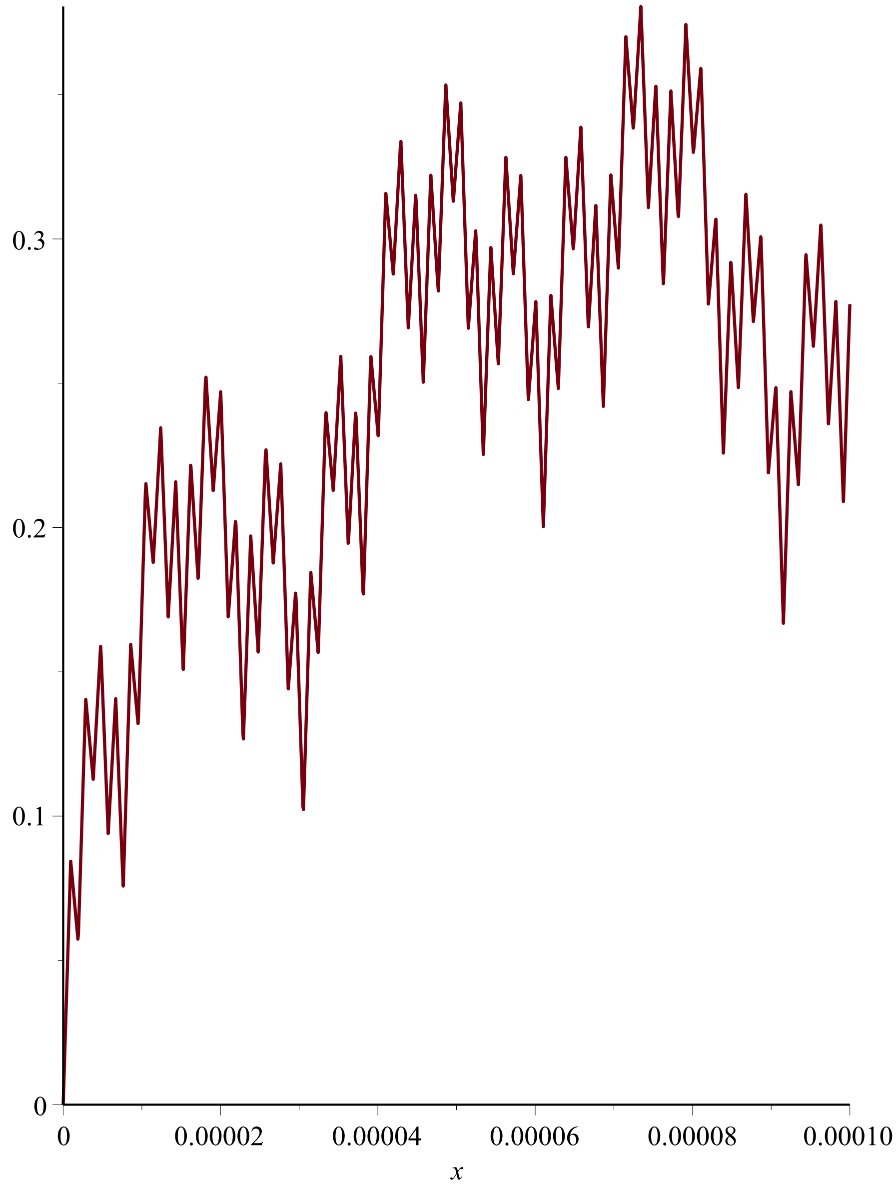
Kuvasta näkyy, miten osasummat muodostuvat. Arvolle N=8 ei kuvan resoluutio enää riitä. Lisätään laskentapisteitä ja pienennetään väliä reippaasti:

```
> plot(osasumma(8),x=0..0.001,numpoints=800);
```



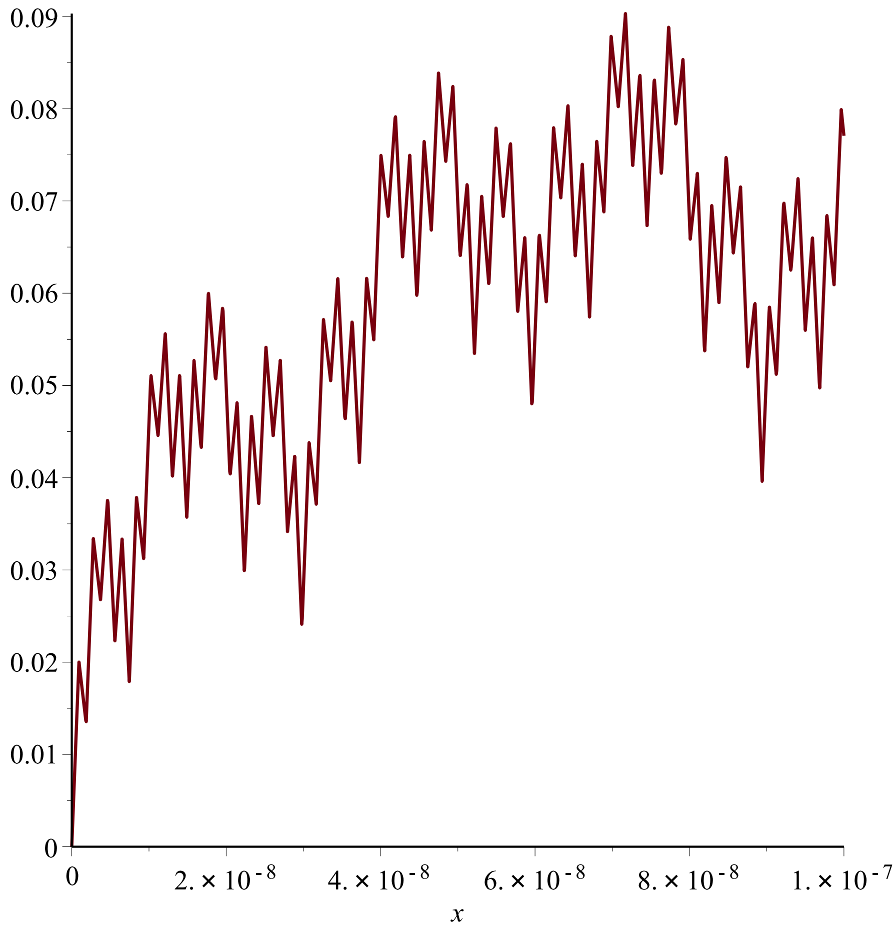
Nytpä alkaa näkyä.
Katsotaan osasummaa N=10, piennetään edelleen väliä.

```
> plot(osasumma(10),x=0..0.0001,numpoints=800);
```



Ja kerran vielä. Nyt N=15 ja väliä joudutaan pienentämään jo arvoon $10^{(-7)}$; .

```
> plot(osasumma(15),x=0..0.0000001,numpoints=800);
```



Summa summarum

On helppo uskoa, että näin voidaan jatkaa loputtomasti. Aina, kun otetaan lisää termejä, pitää lisätä laskentapisteitä
ja rajoittua yhä pienempään väliin. Kuva näyttää aina samanlaiselta. Tässä ilmenee funktion fraktaaliluonne.
Kuvan perusteella funktio näkyy jatkuvana, mutta miten pienessä ympäristössä tahansa on teräviä kulmia, joten derivaattaa
ei ole missään. (Tämä ei ole matemaattinen todistus, mutta tekee asian havainnolliseksi ja uskottavaksi.)