Multiscale calculus

1 The Dadim operators

A Dadim is a computational representation of multi-dimensional function. Each Dadim consists of three components. One is a rule for evaluation, one for translation and one for dilatation. Three operators are defined to access these features.
- The operator triggers the evaluation.
- T translates the Dadim by an integer value.
- Z refines the Dadim by a factor of two

```
> da:=(dadim)-> dadim[1];
> trans:=(dadim, dir, len)-> dadim[2](dir, len);
> zoom:=(dadim, dir)-> dadim[3](dir);
```

$$da := dadim \to dadim_1$$

$$trans := (dadim,\ dir,\ len) \to dadim_2(dir,\ len)$$

$$zoom := (dadim,\ dir) \to dadim_3(dir)$$

2 Constants

The constant Dadim alwayes evaluates to the same value, no matter how often it is translated or dilatated.

$\mathrm{f}(x) = c$

2.1 Definition

cons(c) = c

T cons(c) = cons(c)

Z cons(c) = cons(c)

```
> cons:=c->[c, (dir, len) -> cons(c), (dir) -> cons(c)];
```

$$cons := c \to [c,\ (dir,\ len) \to \mathrm{cons}(c),\ dir \to \mathrm{cons}(c)]$$

2.2 Examples

This example generates a constant Dadim with a value of c

```
> cons(c);
```

$$[c,\ (dir,\ len) \to \mathrm{cons}(c),\ dir \to \mathrm{cons}(c)]$$

```
> da(cons(c));
```

$$c$$

```
> trans(trans(zoom(trans(cons(c),dir,1),0),dir,1),dir,1);
```

$$[c,\ (dir,\ len) \to \mathrm{cons}(c),\ dir \to \mathrm{cons}(c)]$$

```
> da(%);
```

$$c$$

2.3 Addition

```
> cons(4)+cons(1);
```

$$[5,\ ((dir,\ len) \to \mathrm{cons}(1)) + ((dir,\ len) \to \mathrm{cons}(4)),\ (dir \to \mathrm{cons}(1)) + (dir \to \mathrm{cons}(4))]$$

```
> da(%);
```

$$5$$

3 Variables

Variables always evaluate to the distance a dadim has been translated. The directed variable only counts directs into one direction.

f$(x) = x$

3.1 Definition

x = 0

T x = x + 1

Z x = x/2

```
> var:=()->[0, (dir, len)-> var() + cons(len), dir->var()/2];
```

$$var := () \to [0,\ (dir,\ len) \to \mathrm{var}() + \mathrm{cons}(len),\ dir \to \frac{1}{2}\,\mathrm{var}()]$$

```
> da(trans(var(), dir, 4));
```

$$4$$

3.2 Direction Filter

(dadim)_i = dadim

Ti (dadim)_i = (**T**i dadim) _i

Zi (dadim)_i = (**Z**i dadim)_i

Tj (dadim)_i = Zj (dadim)_i = (dadim)_i for i /= j

```
> directed:=(dadim,mdir)->[
> da(dadim),
> (dir,len)->directed(
> piecewise(dir=mdir,
> trans(dadim, dir, len),
> dadim),
> mdir),
> (dir)->directed(
> piecewise(dir=mdir,
> zoom(dadim,dir),
> dadim),
> mdir)];
```

$$directed := (dadim,\ mdir) \to [\mathrm{da}(dadim),$$
$$(dir,\ len) \to \mathrm{directed}(\mathrm{piecewise}(dir = mdir,\ \mathrm{trans}(dadim,\ dir,\ len),\ dadim),\ mdir),$$
$$dir \to \mathrm{directed}(\mathrm{piecewise}(dir = mdir,\ \mathrm{zoom}(dadim,\ dir),\ dadim),\ mdir)]$$

```
> x0:=directed(var(),0):
> x1:=directed(var(),1):
> x2:=directed(var(),2):
```

3.3 Examples

The variable var(0) counts the number of translation in direction 0.

```
> da(x0);
```

$$0$$

```
> da(trans(x0, 0, 2));
```

$$2$$

```
> da(trans(x0, 1, 2));
```

$$0$$

```
> da(trans(zoom(x0, 0),0 ,3));
```

$$\frac{3}{2}$$

4 Plotting

4.1 Sampling funcion values

```
> sample:=(xdadim, ydadim, dir, n)->
> piecewise(n=1,
> [[da(xdadim), da(ydadim)]],
> [[da(xdadim), da(ydadim)],
> op(sample(trans(xdadim, dir, 1), trans(ydadim, dir, 1), dir,
> n-1))]);
```

$sample := (xdadim, ydadim, dir, n) \to \text{piecewise}(n = 1, [[\text{da}(xdadim), \text{da}(ydadim)]], [$
$[\text{da}(xdadim), \text{da}(ydadim)],$
$\text{op}(\text{sample}(\text{trans}(xdadim, dir, 1), \text{trans}(ydadim, dir, 1), dir, n - 1))])$

```
> sample(x0, cons(1), 0, 4);
```

$$[[0, 1], [1, 1], [2, 1], [3, 1]]$$

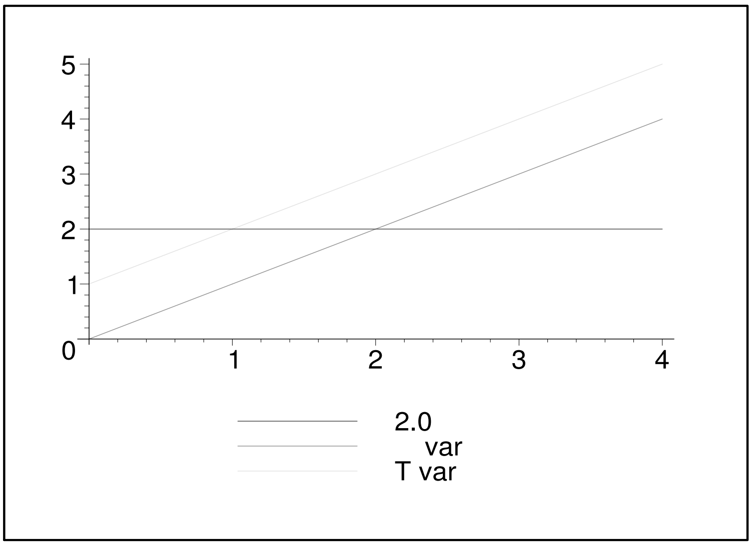
4.2 First Plots

```
> f0:= sample(x0, cons(2.0), 0, 5);
> f1:= sample(x0, x0, 0, 5);
> f2:= sample(x0, trans(x0, 0 ,1), 0, 5);
> plot([f0, f1, f2], legend=["2.0", "    var","T var"]);
```

$$f0 := [[0, 2.0], [1, 2.0], [2, 2.0], [3, 2.0], [4, 2.0]]$$

$$f1 := [[0, 0], [1, 1], [2, 2], [3, 3], [4, 4]]$$

$$f2 := [[0, 1], [1, 2], [2, 3], [3, 4], [4, 5]]$$



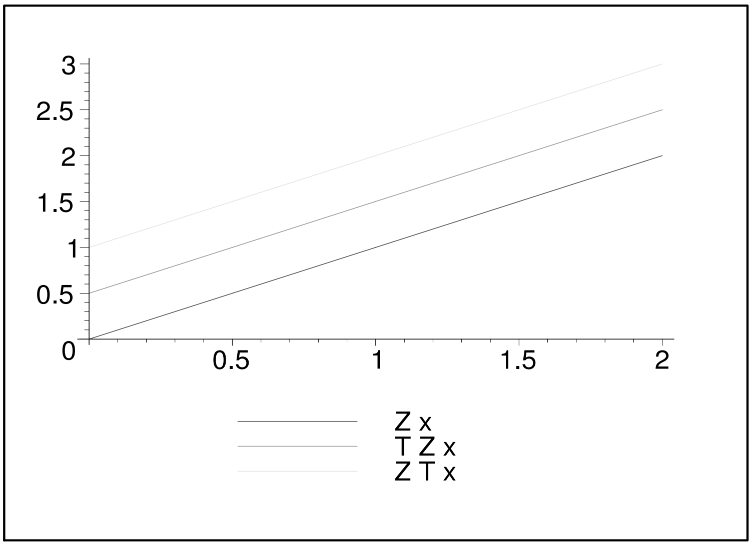
4.3 Plotting scales

```
> f1:= sample(zoom(x0, 0), zoom(x0,0), 0, 5);
> f2:= sample(zoom(x0, 0), trans(zoom(x0, 0), 0 ,1), 0, 5);
> f3:= sample(zoom(x0, 0), zoom(trans(x0, 0, 1), 0), 0, 5);
> plot([f1, f2, f3], legend=["Z x", "T Z x" ,"Z T x"]);
```

$$f1 := [[0, 0], [\frac{1}{2}, \frac{1}{2}], [1, 1], [\frac{3}{2}, \frac{3}{2}], [2, 2]]$$

$$f2 := [[0, \frac{1}{2}], [\frac{1}{2}, 1], [1, \frac{3}{2}], [\frac{3}{2}, 2], [2, \frac{5}{2}]]$$

$$f3 := [[0, 1], [\frac{1}{2}, \frac{3}{2}], [1, 2], [\frac{3}{2}, \frac{5}{2}], [2, 3]]$$



4.4 Plotting in 3D

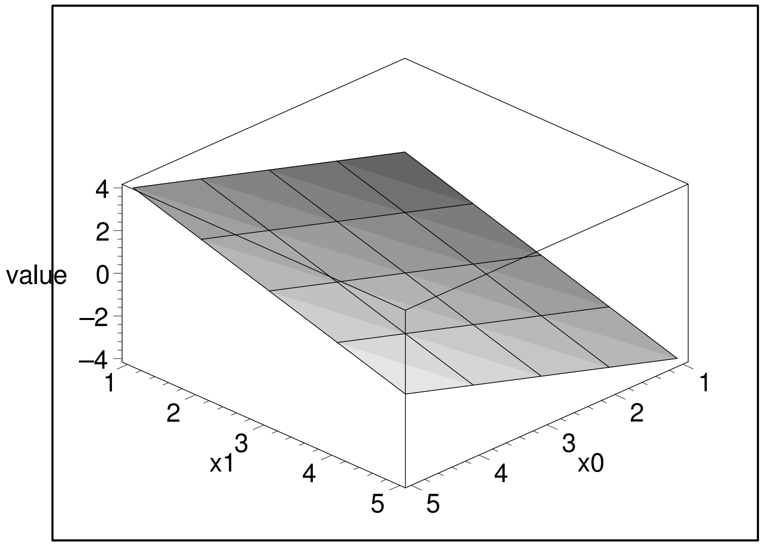
```
> sample3d:= (dadim, n, m) -> Matrix(n, m, (i,j)->da(trans(trans(dadim,
> 0, i-1), 1, j-1)));
```

$sample3d :=$
$(dadim, n, m) \to \text{Matrix}(n, m, (i, j) \to \text{da}(\text{trans}(\text{trans}(dadim, 0, i-1), 1, j-1)))$

```
> dadam:=x0-x1:
> M0:=sample3d(dadam, 5, 5);
```

$$M0 := \begin{bmatrix} 0 & -1 & -2 & -3 & -4 \\ 1 & 0 & -1 & -2 & -3 \\ 2 & 1 & 0 & -1 & -2 \\ 3 & 2 & 1 & 0 & -1 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$

```
> plot3d((i,j)->M0[i,j], 1..5, 1..5, grid=[5,5], axes=boxed,
> labels=["x0","x1","value"]);
```



5 Multiplication

$(f\,g)(x) = \mathrm{f}(x)\,\mathrm{g}(x)$

5.1 Definition

f(x1, .., xn) = f(x1, ..., xn)

T f(x1, .., xn) = f(T x1, .., T xn)

Z f(x1, ...,xn) = f(**Z**x1, .., Z xn)

```
> multop:=(arg,f)->[
> f(op(map(da, arg))),
> (dir,len)->multop(map(a->trans(a, dir, len), arg),f),
> (dir)->multop(map(a->zoom(a,dir), arg), f)];
```

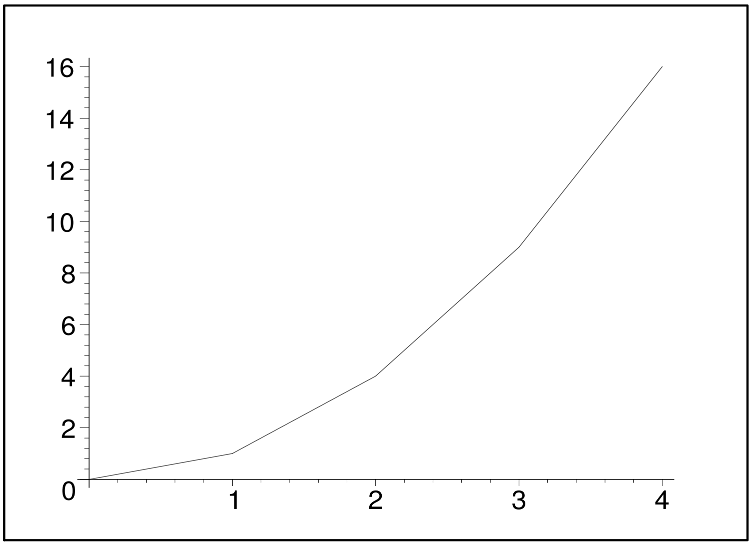
$$multop := (arg,\, f) \to [f(\mathrm{op}(\mathrm{map}(da,\, arg))),\\ (dir,\, len) \to \mathrm{multop}(\mathrm{map}(a \to \mathrm{trans}(a,\, dir,\, len),\, arg),\, f),\\ dir \to \mathrm{multop}(\mathrm{map}(a \to \mathrm{zoom}(a,\, dir),\, arg),\, f)]$$

```
> mult:=(d1,d2)->multop([d1, d2], (x,y)->x*y);
```

$$mult := (d1,\, d2) \to \mathrm{multop}([d1,\, d2],\, (x,\, y) \to x\,y)$$

5.2 A parabola

```
> plot(sample(x0, mult(x0,x0), 0, 5));
```



5.3 Example in 2D

```
> sample3d(mult(x0, x1), 5,5);
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 2 & 4 & 6 & 8 \\ 0 & 3 & 6 & 9 & 12 \\ 0 & 4 & 8 & 12 & 16 \end{bmatrix}$$

5.4 Other Functions

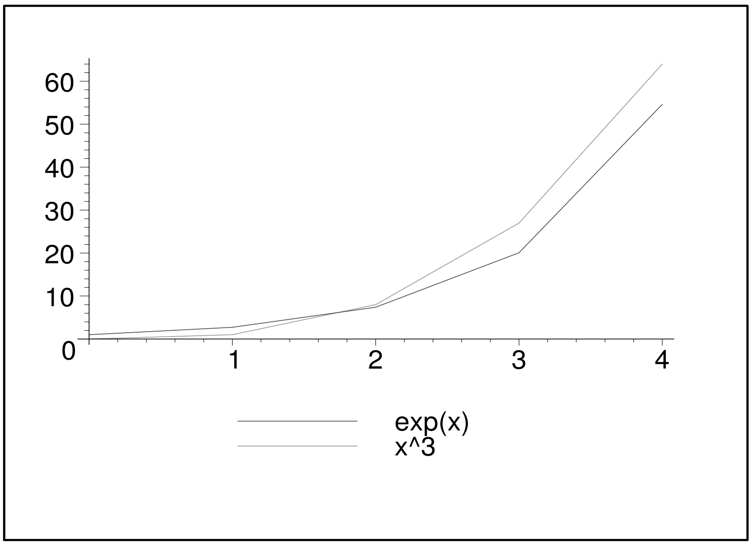
```
> dlog:= (dadim)->multop([dadim], x->log(x));
> dexp:= (dadim)->multop([dadim], x->exp(x));
> power:= (dadim1, dadim2)->multop([dadim1, dadim2], (x,y)->x^y);
```

$$dlog := dadim \to \mathrm{multop}([dadim], \log)$$

$$dexp := dadim \to \mathrm{multop}([dadim], \exp)$$

$$power := (dadim1, dadim2) \to \mathrm{multop}([dadim1, dadim2], (x, y) \to x^y)$$

```
> plot([sample(x0, dexp(x0), 0, 5),
> sample(x0, power(x0, cons(3)), 0, 5)],
> legend=["exp(x)","x^3"]);
```



6 Delta Function

$\int \delta(x-p)\, dx = 1$

$\delta(x-p) = 0$
for $x \neq p$

6.1 Definition

$\delta_p = \begin{cases} 1 & p = 0 \\ 0 & otherwise \end{cases}$

$\mathbf{T}\ \delta_p = \delta_{p+1}$

$\mathbf{Z}\ \delta_p = 2\, \delta_{2\, p}$

```
> delta_:=(p)->[
> piecewise(p=0, 1, 0),
> (dir, len) -> delta_(p-len,dir),
> dir-> 2*delta_(2*p,dir)];
```

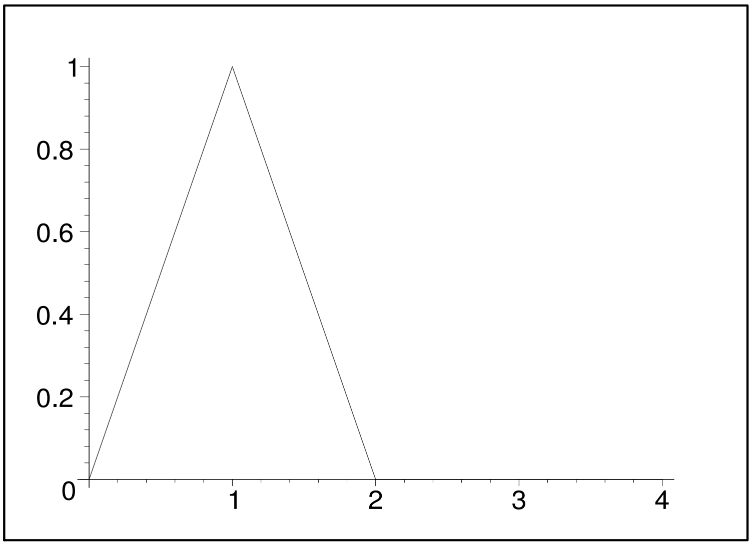
$delta\_ := p \rightarrow$
$[\mathrm{piecewise}(p = 0,\, 1,\, 0),\, (dir,\, len) \rightarrow \mathrm{delta\_}(p - len,\, dir),\, dir \rightarrow 2\,\mathrm{delta\_}(2\, p,\, dir)]$

```
> delta:=(pos,mdir)->directed(delta_(pos),mdir);
```

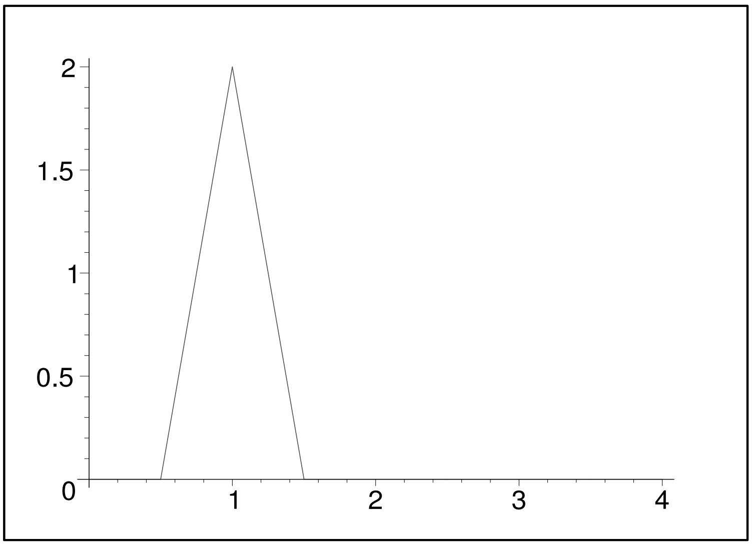
$$\delta := (pos,\, mdir) \rightarrow \mathrm{directed}(\mathrm{delta\_}(pos),\, mdir)$$

6.2 Plots on various scales

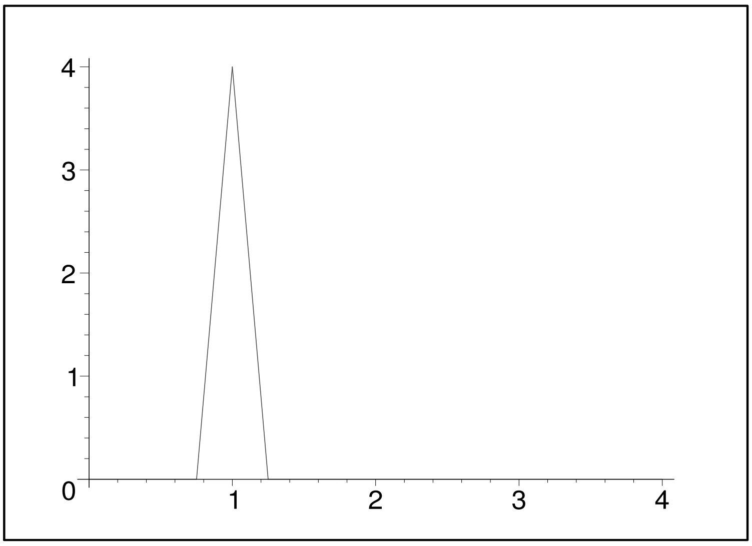
```
>  plot(sample(x0, delta(1,0), 0, 5));
```




```
>  plot(sample(zoom(x0,0), zoom(delta(1,0),0),0 ,9));
```




```
>  plot(sample(zoom(zoom(x0,0),0), zoom(zoom(delta(1,0),0),0),0,17));
```



7 Hat operator

$(H\,f)(x) = \int_{x-1}^{x} \mathrm{f}(t)\,dt$

7.1 Definition

H =

T H = H **T**

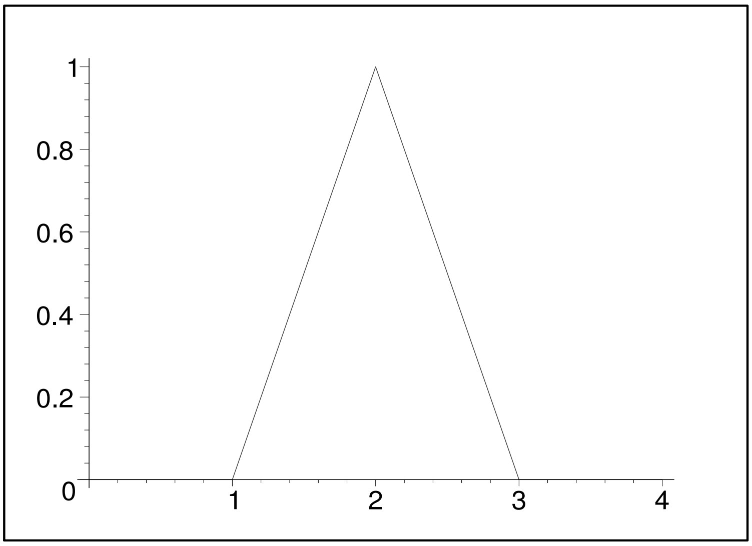
Z H = (H **Z** + H **T** **Z**)/2

```
> hat:=(dadim, mdir)->[
> da(dadim),
> (dir, len) -> hat(trans(dadim, dir, len), mdir),
> (dir) ->
> piecewise(dir=mdir,
> (hat(zoom(dadim, dir) + trans(zoom(dadim,dir), dir, +1),dir))/2,
> hat(zoom(dadim, dir), mdir))];
```

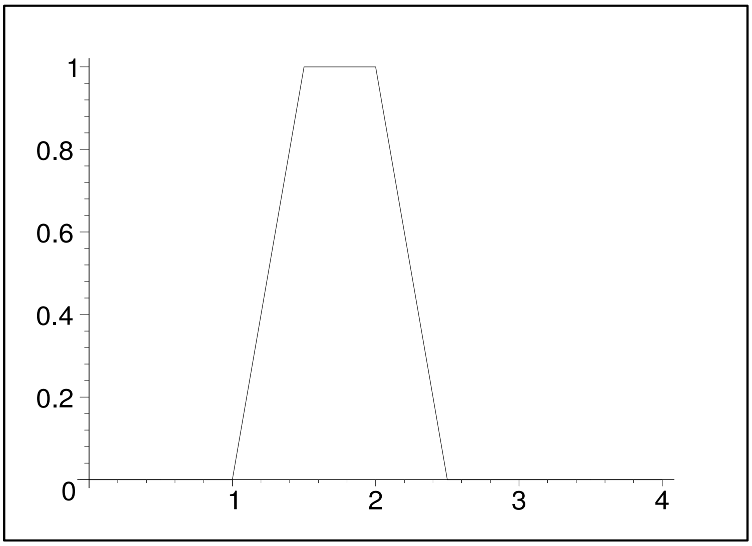
$hat := (dadim,\, mdir) \to [\mathrm{da}(dadim),\, (dir,\, len) \to \mathrm{hat}(\mathrm{trans}(dadim,\, dir,\, len),\, mdir),\, dir \to$
$\mathrm{piecewise}(dir = mdir,$
$\frac{1}{2}\,\mathrm{hat}(\mathrm{zoom}(dadim,\, dir) + \mathrm{trans}(\mathrm{zoom}(dadim,\, dir),\, dir,\, 1),\, dir),$
$\mathrm{hat}(\mathrm{zoom}(dadim,\, dir),\, mdir))]$

7.2 Examples

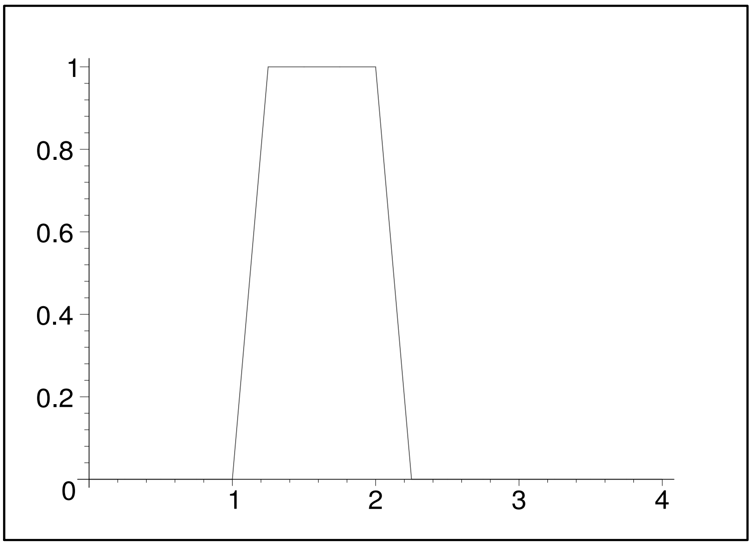
```
> dadam:=hat(delta(2,0),0):
> plot(sample(x0, dadam, 0, 5));
```




```
> plot(sample(zoom(x0,0), zoom(dadam,0), 0, 9));
```



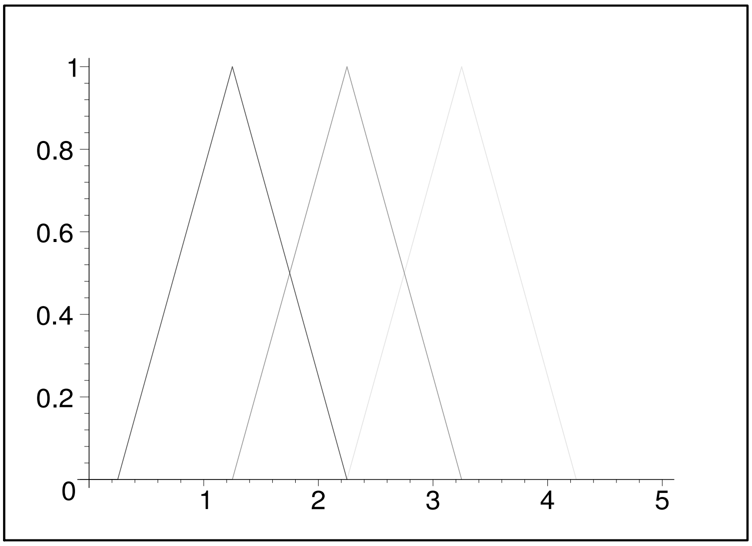

```
> plot(sample(zoom(zoom(x0,0),0), zoom(zoom(dadam,0),0),0,17));
```



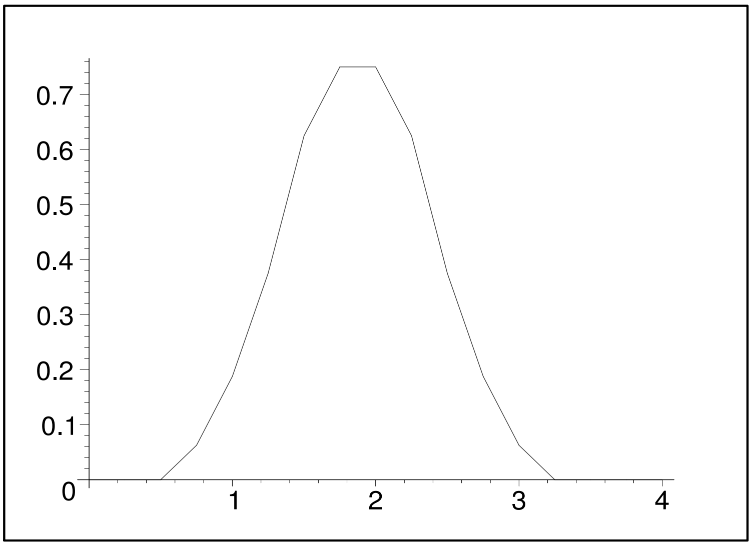
7.3 B-Spline basis functions

B-Spline basis function can be created by repeated application of the hat operator.

```
> dadam:=hat(hat(delta(2,0),0),0):
> plot([sample(zoom(zoom(x0,0),0), zoom(zoom(dadam,0),0), 0,10),
> sample(zoom(zoom(x0,0),0),
> zoom(zoom(trans(dadam,0,-1),0),0),0,14),
> sample(zoom(zoom(x0,0),0),
> zoom(zoom(trans(dadam,0,-2),0),0),0,21)]);
```

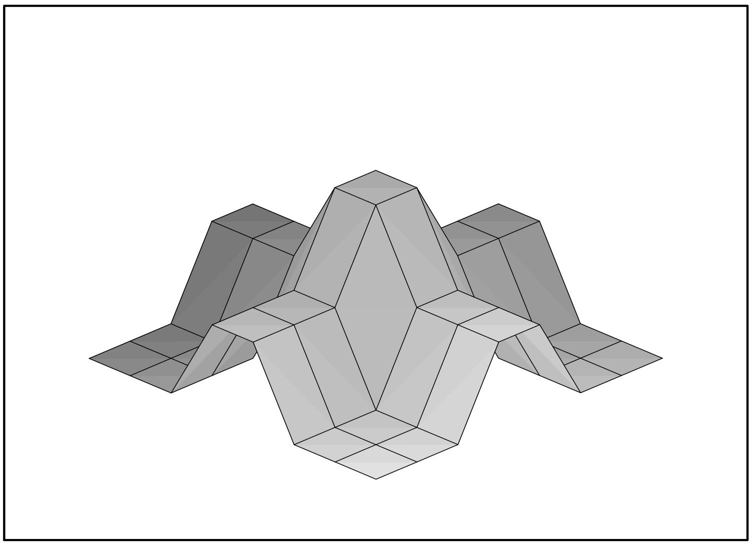


```
> dadam:=hat(hat(hat(delta(3,0),0),0),0):
> plot(sample(zoom(zoom(x0,0),0), zoom(zoom(dadam,0),0),0,17));
```



7.4 Hat operator in two dimensions

```
> dadam:=hat(hat(delta(2,0)+delta(2,1),0),1):
> M0:=sample3d(zoom(zoom(dadam,1),0), 8,8):
> plot3d((i,j)->M0[i,j], 1..8, 1..8, grid=[8,8]);
```



8 Differentiation

$(\Delta\, f)(x) = \frac{d}{dx}\,\mathrm{f}(x)$

8.1 Differential operator

Δ
$=$ **T** -

T $\Delta = \Delta$ T

Z $\Delta = 2\ \Delta$ **Z**

```
> dif:=(dadim, mdir) -> [
> da(trans(dadim, mdir, 1)) - da(dadim),
> (dir, len) -> dif(trans(dadim, dir, len), mdir),
> dir -> piecewise(mdir=dir, 2*dif(zoom(dadim, dir),dir),
> dif(zoom(dadim,dir),mdir))];
```

$$dif := (dadim,\ mdir) \to [\mathrm{da}(\mathrm{trans}(dadim,\ mdir,\ 1)) - \mathrm{da}(dadim),$$
$$(dir,\ len) \to \mathrm{dif}(\mathrm{trans}(dadim,\ dir,\ len),\ mdir),\ dir \to$$
$$\mathrm{piecewise}(mdir = dir,\ 2\,\mathrm{dif}(\mathrm{zoom}(dadim,\ dir),\ dir),\ \mathrm{dif}(\mathrm{zoom}(dadim,\ dir),\ mdir))]$$

```
> Diff(dadim, x);
```

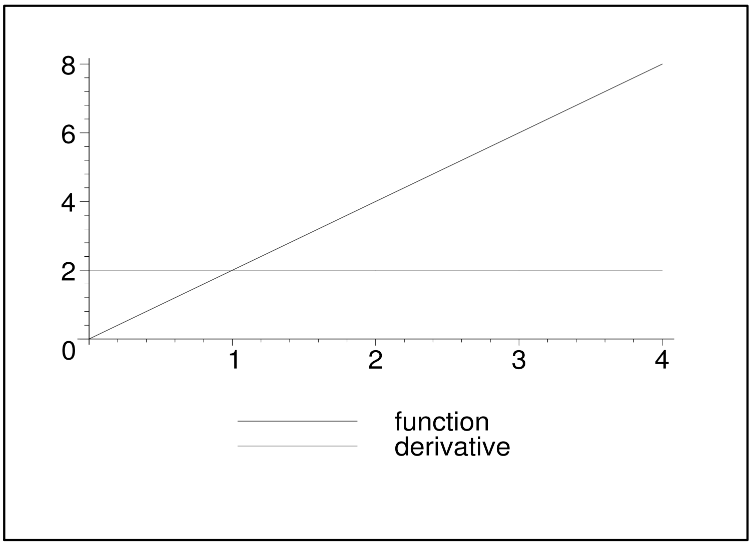
$$\frac{\partial}{\partial x}\, dadim$$

8.2 Derivative of a linear function

```
> 2*x; diff(%,x);
```

$$2\,x$$
$$2$$

```
> dadam:= 2*x0:
> plot([sample(x0, dadam, 0, 5),
> sample(x0, dif(dadam,0),0,5)], legend=["function",
> "derivative"]);
```

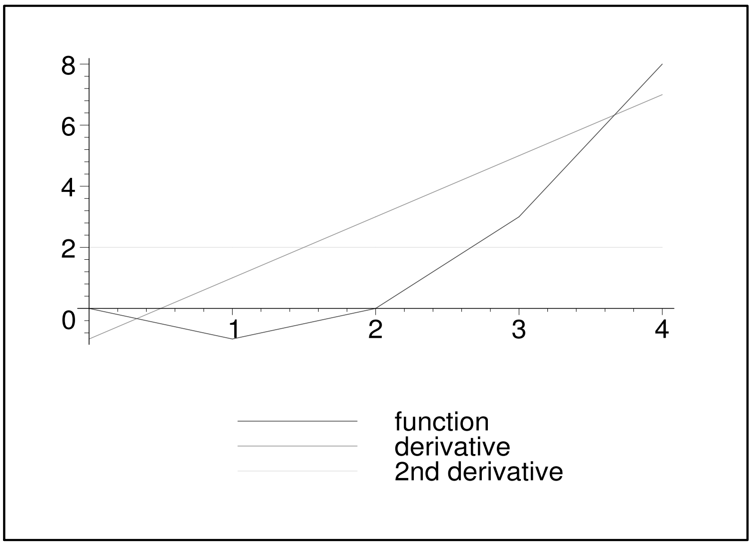


8.3 Derivaties of polynomials

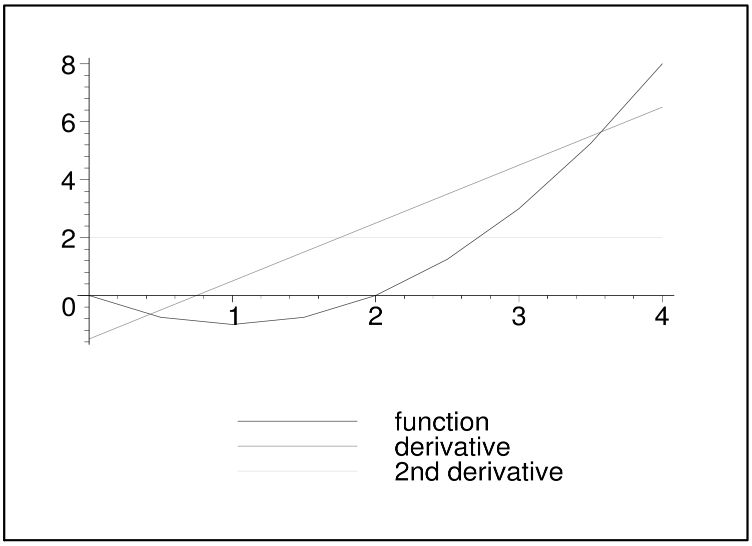
```
> x^2 - 2*x;
> diff(%,x);
> diff(%,x);
```

$$x^2 - 2\,x$$
$$2\,x - 2$$
$$2$$

```
> dadam:=mult(x0,x0) - 2*x0:
> plot([sample(x0, dadam,0,5),
> sample(x0, dif(dadam,0),0,5),
> sample(x0, dif(dif(dadam,0),0),0,5)],
> legend=["function", "derivative", "2nd derivative"]);
```




```
>  dadam:=mult(x0, x0) - mult(cons(2), x0):
>  plot([
>  sample(zoom(x0,0), zoom(dadam,0),0,9),
>  sample(zoom(x0,0), zoom(dif(dadam,0),0),0,9),
>  sample(zoom(x0,0), zoom(dif(dif(dadam,0),0),0),0,9)],
>  legend=["function", "derivative", "2nd derivative"]);
```



8.4 Product rule

Product rule:

$\Delta(fg) = f\,\Delta\,g + g\,\Delta\,f + h\,\Delta\,f\,\Delta\,g$

```
> dadam:=trans(dif(dadam,0),0, 1):
> da(dadam);
```

$$1$$

```
> da(zoom(dadam,0));
```

$$\frac{1}{2}$$

```
> da(zoom(zoom(dadam,0),0));
```

$$\frac{1}{4}$$

9 Integration

$(I\,f)(x) = \int \mathrm{f}(x)\,dx$

9.1 Integration operator

$\int dadim\,dx$
$= dadim$

T $\int dadim\,dx = H\;dadim + \int dadim\,dx$

Z $\int dadim\,dx = \frac{1\int Z\;dadim\;dx}{2}$

```
> integ0:=(dadim, mdir)->[
> 0, (dir, len)->
> piecewise(dir=mdir,
> integ0(dadim,dir)+len*hat(dadim,dir),
> integ(trans(dadim, dir, len), mdir)),
> (dir) ->
> piecewise(dir=mdir,
> mult(cons(1/2), integ0(zoom(dadim, dir), mdir)),
> integ0(zoom(dadim, dir, mdir)))];
```

$$integ0 := (dadim,\ mdir) \to [0, (dir,\ len) \to \mathrm{piecewise}(dir = mdir, \mathrm{integ0}(dadim,\ dir) + len\,\mathrm{hat}(dadim,\ dir), \mathrm{integ}(\mathrm{trans}(dadim,\ dir,\ len),\ mdir)), dir \to \mathrm{piecewise}(dir = mdir, \mathrm{mult}(\mathrm{cons}(\frac{1}{2}), \mathrm{integ0}(\mathrm{zoom}(dadim,\ dir),\ mdir)), \mathrm{integ0}(\mathrm{zoom}(dadim,\ dir,\ mdir)))]$$

```
> stepwise:=(dadim)->[
> da(dadim),
> (dir, len) -> piecewise(
> len>0, trans(stepwise(trans(dadim, dir, 1)), dir, len-1),
> len<0, trans(stepwise(trans(dadim, dir, -1)), dir, len+1),
> stepwise(dadim)),
> dir-> stepwise(zoom(dadim,dir))];
```

$$stepwise := dadim \rightarrow [\mathrm{da}(dadim), (dir,\ len) \rightarrow \mathrm{piecewise}(0 < len,$$
$$\mathrm{trans}(\mathrm{stepwise}(\mathrm{trans}(dadim,\ dir,\ 1)),\ dir,\ len - 1),\ len < 0,$$
$$\mathrm{trans}(\mathrm{stepwise}(\mathrm{trans}(dadim,\ dir,\ -1)),\ dir,\ len + 1),\ \mathrm{stepwise}(dadim)),$$
$$dir \rightarrow \mathrm{stepwise}(\mathrm{zoom}(dadim,\ dir))]$$

```
> integ:= (dadim, dir)->stepwise(integ0(dadim,dir));
```

$$integ := (dadim,\ dir) \rightarrow \mathrm{stepwise}(\mathrm{integ0}(dadim,\ dir))$$

```
> Int(dadim,x);
```

$$\int dadim\, dx$$

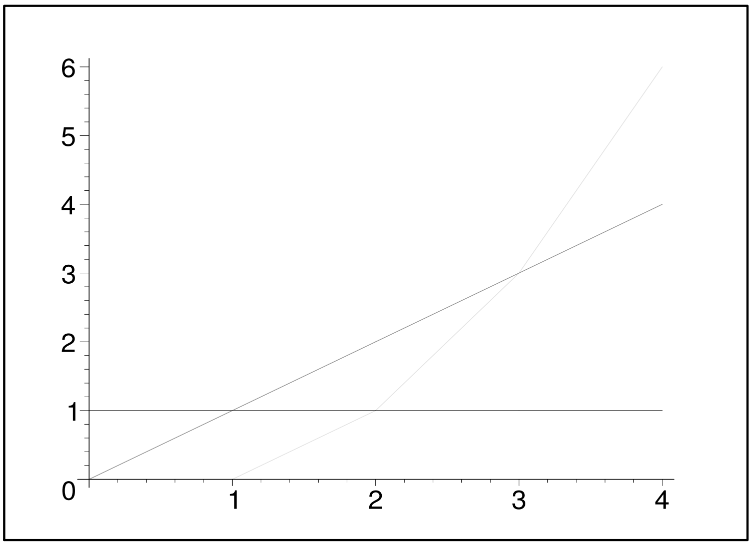
9.2 Examples

```
> 1; int(%,x); int(%,x);
```

$$1$$
$$x$$
$$\frac{x^2}{2}$$

```
> dadam:=cons(1):
> f0:=sample(x0, dadam, 0, 5);
> f1:=sample(x0, integ(dadam, 0), 0, 5);
> f2:=sample(x0, integ(integ(dadam,0),0),0,5);
> plot([f0, f1, f2]);
```

$$f0 := [[0,\ 1],\ [1,\ 1],\ [2,\ 1],\ [3,\ 1],\ [4,\ 1]]$$
$$f1 := [[0,\ 0],\ [1,\ 1],\ [2,\ 2],\ [3,\ 3],\ [4,\ 4]]$$
$$f2 := [[0,\ 0],\ [1,\ 0],\ [2,\ 1],\ [3,\ 3],\ [4,\ 6]]$$

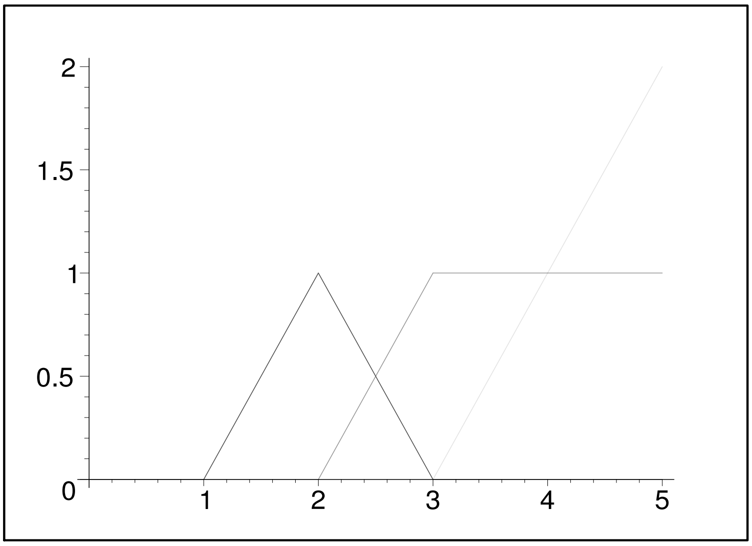


```
>  dadam:=delta(2,0):
>  f0:=sample(x0, dadam,0,6);
>  f1:=sample(x0, integ(dadam, 0), 0, 6);
>  f2:=sample(x0, integ(integ(dadam,0),0),0,6);
>  plot([f0, f1, f2]);
```

$$f0 := [[0, 0], [1, 0], [2, 1], [3, 0], [4, 0], [5, 0]]$$

$$f1 := [[0, 0], [1, 0], [2, 0], [3, 1], [4, 1], [5, 1]]$$

$$f2 := [[0, 0], [1, 0], [2, 0], [3, 0], [4, 1], [5, 2]]$$



9.3 Inversity of integration and differentiation

$x^2 + 1$

```
>  dadam:=mult(x0,x0)+cons(1):
>  sample(x0,dadam, 0,5);
```

$$[[0, 1], [1, 2], [2, 5], [3, 10], [4, 17]]$$

$\int \frac{d}{dx}\left(x^2 + 1\right) dx = x^2$

```
>  sample(x0,integ(dif(dadam,0),0), 0,5);
```

$$[[0, 0], [1, 1], [2, 4], [3, 9], [4, 16]]$$

$\frac{d^2}{dx^2}\left(\int\int x^2 + 1\, dx\, dx\right) = x^2 + 1$

```
>  sample(x0, dif(dif(integ(integ(dadam,0),0),0),0), 0,5);
```

$$[[0, 1], [1, 2], [2, 5], [3, 10], [4, 17]]$$

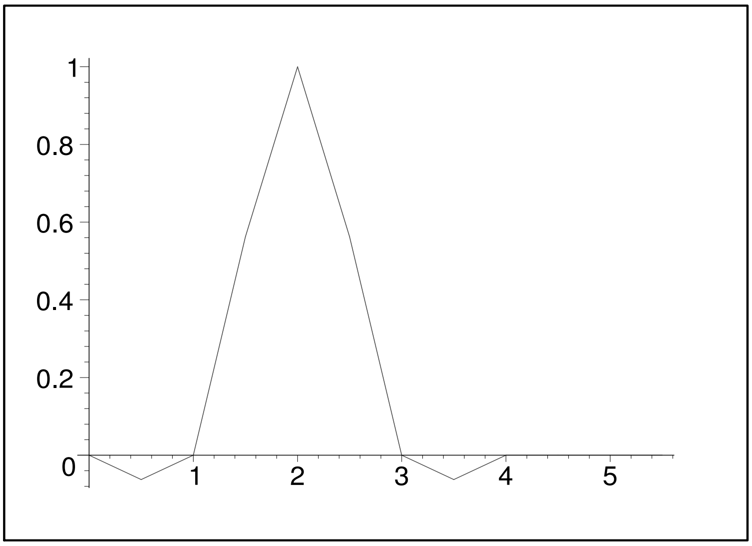
10 Wavelets

10.1 Subdivision

```
> predict_:=(dadim, mdir,state)->[
> piecewise(state=0,
> da(dadim),
> -1/16*da(trans(dadim,mdir,-1))+9/16*da(trans(dadim,mdir,0))+
> 9/16*da(trans(dadim,mdir,1))-1/16*da(trans(dadim,mdir,2))),
> (dir,len) -> piecewise(dir=mdir,
> predict_(trans(dadim, dir, floor((state+len)/2)), dir, (state+len)
> mod 2),
> predict_(trans(dadim, dir, len), mdir, state)),
> (dir) -> piecewise(dir=mdir,
> predict_(zoom(dadim, dir), dir, 0),
> predict_(zoom(dadim, dir), mdir, state))]:
> predict:= (dadim, dir) -> predict_(dadim, dir, 0):
```

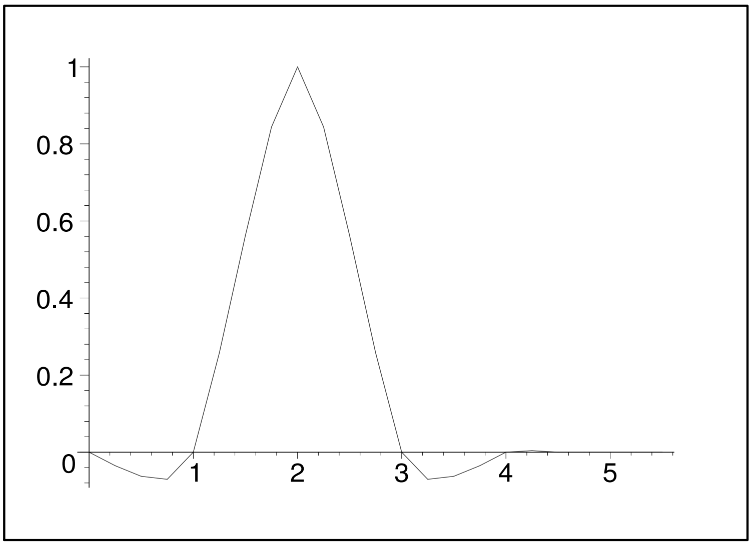
```
> dadam:= delta(2,0):
```

```
> plot(sample(zoom(x0,0),predict(dadam,0),0, 12));
```

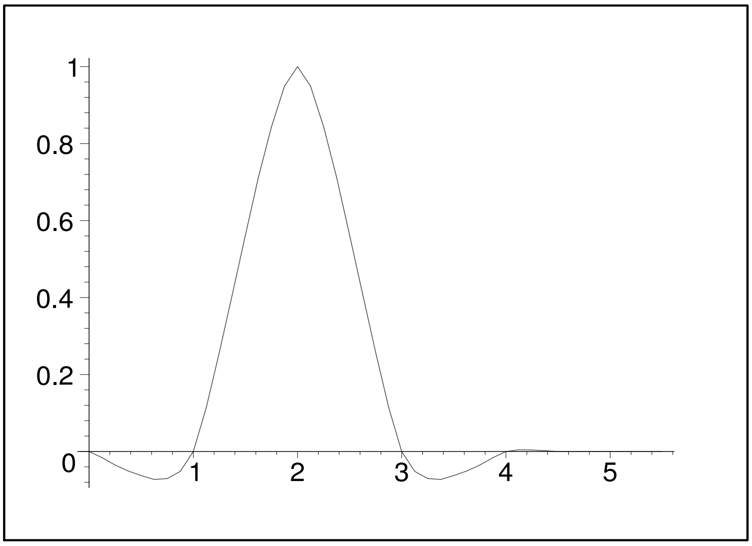


10.2 Constructing the Scaling Function

```
> dadam:=delta(2,0):
> plot(sample(zoom(zoom(x0,0),0), predict(predict(dadam,0),0),0,23));
```

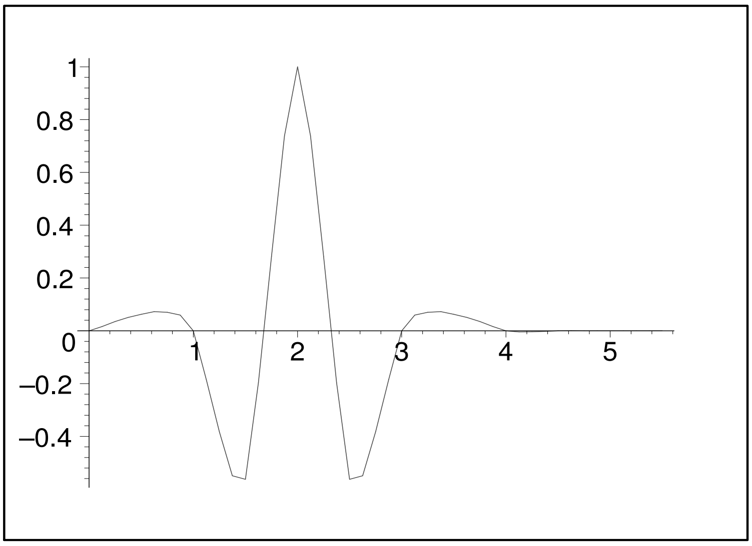



```
> zzzx0:=zoom(zoom(zoom(x0,0),0),0):
> plot(sample(zzzx0, predict(predict(predict(dadam,0),0),0),0,45));
```



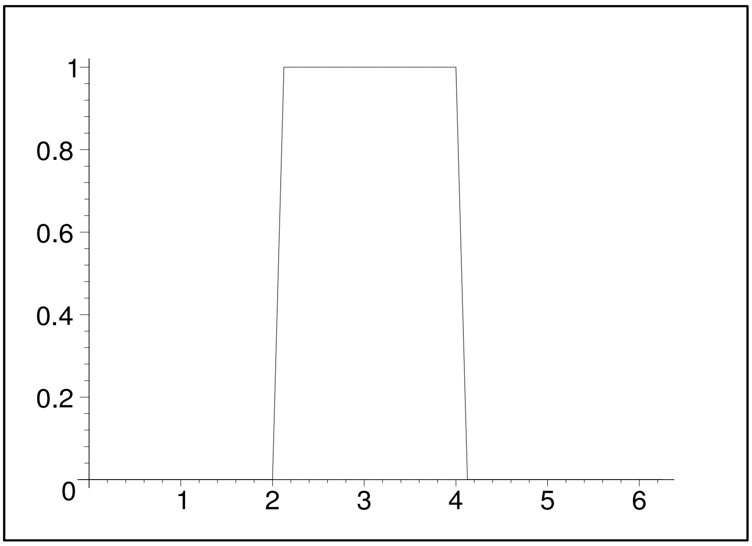
10.3 Constructing the Wavlelet

```
> f0:=predict(predict(zoom(dadam,0)-predict(dadam,0),0),0):
> plot(sample(zzzx0, f0, 0, 45));
```

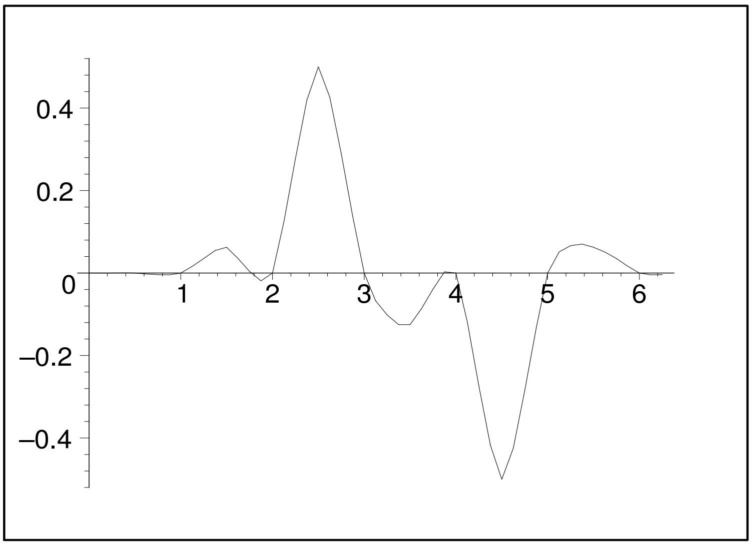


10.4 Filterbank Algorithm

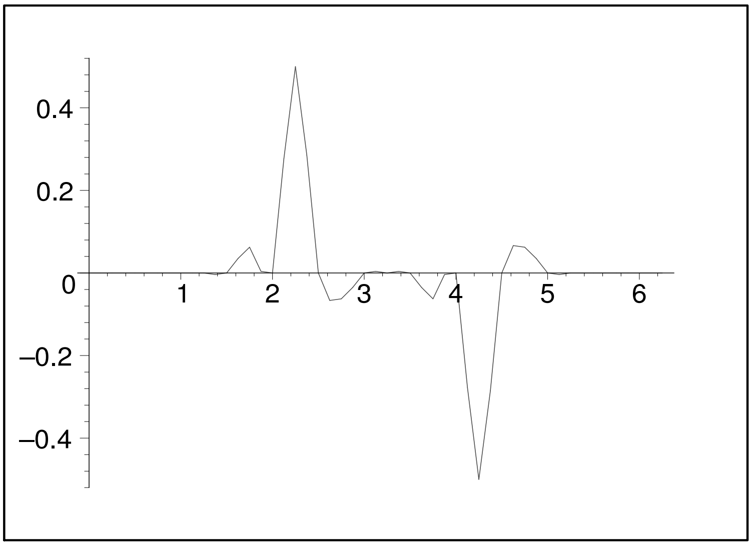
```
> dadam:=zoom(hat(delta(2,0),0),0):
> plot(sample(zzzx0, zoom(zoom(zoom(dadam,0),0),0),0,51));
```



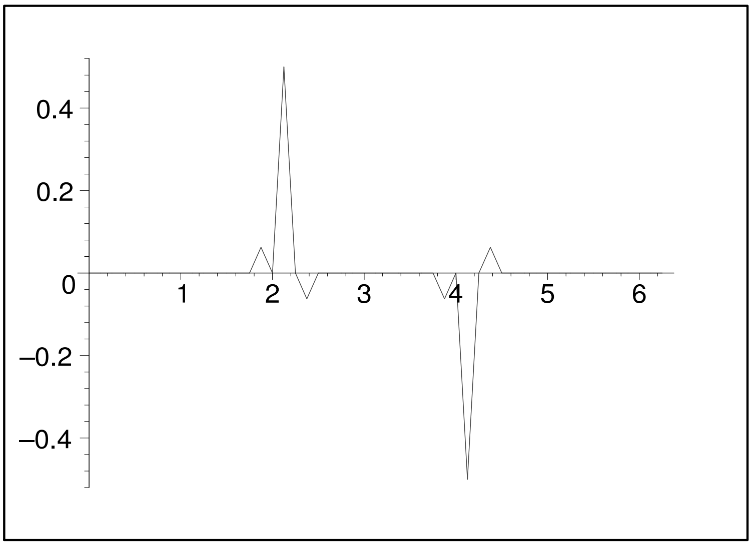

```
> f0:=predict(predict(zoom(dadam,0)-predict(dadam,0),0),0):
> plot(sample(zzzx0, f0, 0, 51));
```




```
> f1:=predict(zoom(zoom(dadam,0),0)-predict(zoom(dadam,0),0),0):
> plot(sample(zzzx0, f1, 0, 51));
```




```
> f2:=zoom(zoom(zoom(dadam,0),0),0)-predict(zoom(zoom(dadam,0),0),0):
> plot(sample(zzzx0, f2, 0, 51));
```



11 Diffusion

Convolute a function with the normal distribution.

$(B\,f)(x) = \int \mathrm{f}(x)\,\mathrm{gauss}(x-t)\,dt$

11.1 Blur operater

$\mathrm{B} = \frac{1}{4}\,T^{(-1)} + \frac{1}{2} + \frac{1}{4}\,T$

$\mathbf{T}\,\mathrm{B} = \mathrm{B}\,\mathbf{T}$

$\mathbf{Z}\,\mathrm{B} = \mathrm{B\;B\;B\;B}\,\mathbf{Z}$

```
> blur:=(dadim, mdir) -> [
> 0.25*da(trans(dadim,mdir,-1))+0.5*da(dadim)+0.25*da(trans(dadim,mdir,1
> )),
> (dir,len) -> blur(trans(dadim, dir, len), mdir),
> dir -> piecewise(dir=mdir,
> blur(blur(blur(blur(zoom(dadim,dir),dir),dir),dir),dir),
> blur(zoom(dadim, dir), mdir))];
```

$\mathit{blur} := (\mathit{dadim},\ \mathit{mdir}) \to [$
$0.25\,\mathrm{da}(\mathrm{trans}(\mathit{dadim},\ \mathit{mdir},\ -1)) + 0.5\,\mathrm{da}(\mathit{dadim}) + 0.25\,\mathrm{da}(\mathrm{trans}(\mathit{dadim},\ \mathit{mdir},\ 1)),$
$(\mathit{dir},\ \mathit{len}) \to \mathrm{blur}(\mathrm{trans}(\mathit{dadim},\ \mathit{dir},\ \mathit{len}),\ \mathit{mdir}),\ \mathit{dir} \to \mathrm{piecewise}(\mathit{dir} = \mathit{mdir},$
$\mathrm{blur}(\mathrm{blur}(\mathrm{blur}(\mathrm{blur}(\mathrm{zoom}(\mathit{dadim},\ \mathit{dir}),\ \mathit{dir}),\ \mathit{dir}),\ \mathit{dir}),\ \mathit{dir}),$
$\mathrm{blur}(\mathrm{zoom}(\mathit{dadim},\ \mathit{dir}),\ \mathit{mdir}))]$

```
> centerofmass:=(dadim, scale, dir)->integ(mult(scale,dadim), dir);
> expect:=centerofmass;
```

$$centerofmass := (dadim,\ scale,\ dir) \to \mathrm{integ}(\mathrm{mult}(scale,\ dadim),\ dir)$$

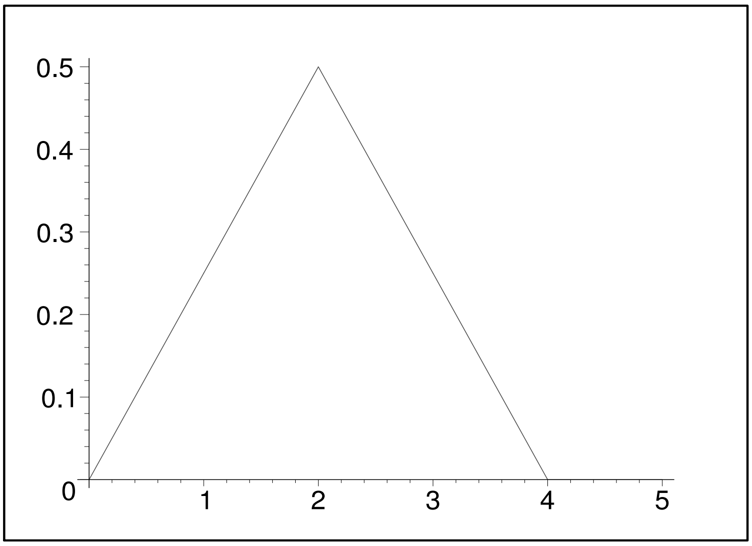
$$expect := centerofmass$$

```
> variance:=(dadim, scale, dir)->
> expect(dadim, power(scale,cons(2)), dir)-
> power(expect(dadim, scale, dir),cons(2));
```

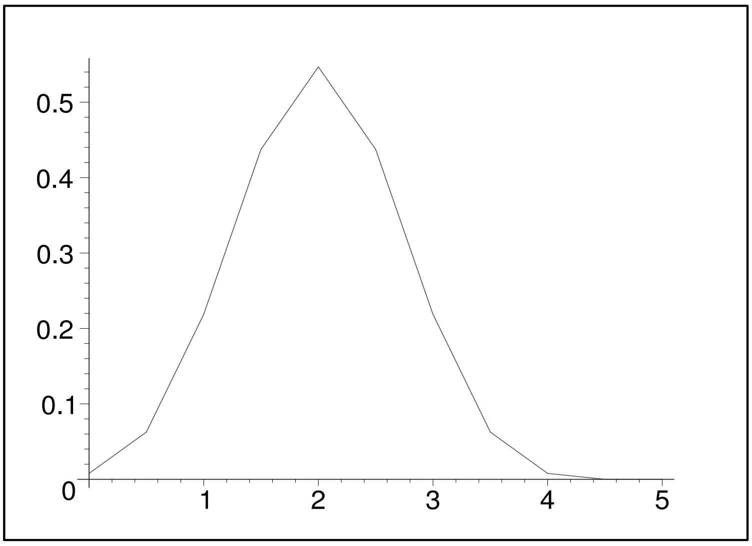
$$variance := (dadim,\ scale,\ dir) \to \mathrm{expect}(dadim,\ \mathrm{power}(scale,\ \mathrm{cons}(2)),\ dir) - \mathrm{power}(\mathrm{expect}(dadim,\ scale,\ dir),\ \mathrm{cons}(2))$$

11.2 Bell Curve:

```
> dadam:=blur(delta(2,0),0):
> plot(sample(x0, dadam, 0, 6));
```




```
> plot(sample(zoom(x0,0), zoom(dadam,0),0,11));
```




```
> da(trans(centerofmass(dadam,var(0), 0),0 ,5));
```

$$2.00$$

```
> da(trans(variance(dadam, var(0), 0), 0, 5));
```

$$0.5000$$

11.3 Logarithmic scale

```
> logblurmap0:=(h,v)->v;
> logblurmap1:=h->p;
> logblur := proc (dadim, mdir, v, h)
> [ proc ()
> local p,v1,t:
> p:=logblurmap1(h);
> t:=logblurmap0(h,v);
> t*((1-p)^2*da(trans(dadim,mdir,1)) +
> 2*p*(1-p)*da(dadim) +
> p^2*da(trans(dadim,mdir,-1))) +
> (1-t)*da(dadim);
> end proc(),
> proc (dir, len)
> logblur(trans(dadim,dir,len),mdir,v,h)
> end proc,
> proc (dir)
> piecewise(dir=mdir,
> logblur(
> logblur(
> logblur(
> logblur(
> zoom(dadim,dir), mdir, v/4, h/2
> ), mdir, v/4, h/2
> ),mdir, v/4, h/2
> ), mdir, v/4, h/2),
> logblur(zoom(dadim,dir),mdir,v,h))
> end proc]
> end proc;
```

$$logblurmap0 := (h, v) \to v$$

$$logblurmap1 := h \to p$$

$logblur :=$ **proc**$(dadim, mdir, v, h)$
[(**proc**()
local p, $v1$, t;
$p :=$ logblurmap1(h) ;
$t :=$ logblurmap0(h, v) ;
$t * ((1-p)^2 * \mathrm{da}(\mathrm{trans}(dadim, mdir, 1)) + 2 * p * (1-p) * \mathrm{da}(dadim)$
$+ p^2 * \mathrm{da}(\mathrm{trans}(dadim, mdir, -1))) + (1-t) * \mathrm{da}(dadim)$
end proc)(), **proc**(dir, len) logblur(trans$(dadim, dir, len)$, $mdir$, v, h) **end proc**,
proc(dir)
piecewise($dir = mdir$, logblur(logblur(
logblur(logblur(zoom($dadim$, dir), $mdir$, $1/4 * v$, $1/2 * h$), $mdir$, $1/4 * v$, $1/2 * h$), $mdir$,
$1/4 * v$, $1/2 * h$), $mdir$, $1/4 * v$, $1/2 * h$), logblur(zoom($dadim$, dir), $mdir$, v, h))
end proc]
end proc

```
> h:=r;
> scale1:=mult(cons(h),x0)-cons(2*h):
> sample(x0,scale1, 0,5);
```

$$h := r$$

$$[[0, -2\,r], [1, -r], [2, 0], [3, r], [4, 2\,r]]$$

```
> scale2:=dexp(scale1):
> sample(x0,scale2, 0,5);
```

$$[[0, e^{(-2\,r)}], [1, e^{(-r)}], [2, 1], [3, e^r], [4, e^{(2\,r)}]]$$

```
> dadam:=logblur(delta(2,0), 0, t, h):
> simplify(sample(x0,dadam,0,5));
```

$$[[0, 0], [1, t\,(-1+p)^2], [2, 2\,t\,p - 2\,t\,p^2 + 1 - t], [3, t\,p^2], [4, 0]]$$

```
> m1:=trans(centerofmass(dadam, scale2 ,0), 0, 5):
> mu:=simplify(da(m1));
```

$$\mu := 2\,t\,p - 2\,t\,p^2 + 1 - t + e^{(-r)}\,t - 2\,e^{(-r)}\,t\,p + e^{(-r)}\,t\,p^2 + e^r\,t\,p^2$$

```
> res0:=simplify([solve(subs(t=1,mu)=1, p)]);
```

$$\mathit{res0} := \left[\frac{1}{e^{(\frac{r}{2})}+1},\, -\frac{1}{-1+e^{(\frac{r}{2})}}\right]$$

```
> res0select:=piecewise(evalf(subs(r=1, res0[1]))>0, res0[1],
> res0[2]);
```

$$\mathit{res0select} := \frac{1}{e^{(\frac{r}{2})}+1}$$

```
> logblurmap1:=h->subs(r=h, res0select);
```

$$\mathit{logblurmap1} := h \to \mathrm{subs}(r = h,\, \mathit{res0select})$$

```
> dadam:= logblur(delta(2,0), 0, t, h):
> simplify(sample(x0,dadam,0,5));
```

$$\left[[0, 0],\, \left[1,\, \frac{t\,e^r}{(e^{(\frac{r}{2})}+1)^2}\right],\, \left[2,\, \frac{e^r + 2\,e^{(\frac{r}{2})} + 1 - e^r\,t - t}{(e^{(\frac{r}{2})}+1)^2}\right],\, \left[3,\, \frac{t}{(e^{(\frac{r}{2})}+1)^2}\right],\, [4, 0]\right]$$

```
> m1log:=trans(centerofmass(dadam, scale1, 0), 0, 5):
> mu:=simplify(da(m1log));
```

$$\mu := -\frac{r\,t\,(-1+e^{(\frac{r}{2})})}{e^{(\frac{r}{2})}+1}$$

```
> m2log:=trans(variance(dadam, scale1, 0), 0, 5):
> sigma:=simplify(da(m2log));
```

$$\sigma := -\frac{r^2\,t\,(-e^r - 1 + e^r\,t - 2\,t\,e^{(\frac{r}{2})} + t)}{(e^{(\frac{r}{2})}+1)^2}$$

```
> res:=simplify([solve(sigma=vola,t)]);
```

$$res := \left[\frac{1}{2}\,\frac{r\,e^r + r + \sqrt{r^2\,e^{(2\,r)} + 2\,r^2\,e^r + r^2 - 4\,e^{(2\,r)}\,vola + 8\,vola\,e^r - 4\,vola}}{(e^r - 2\,e^{(\frac{r}{2})} + 1)\,r},\right.$$

$$\left.\frac{1}{2}\,\frac{r\,e^r + r - \sqrt{r^2\,e^{(2\,r)} + 2\,r^2\,e^r + r^2 - 4\,e^{(2\,r)}\,vola + 8\,vola\,e^r - 4\,vola}}{(e^r - 2\,e^{(\frac{r}{2})} + 1)\,r}\right]$$

```
> weight:=piecewise(subs({r=1,vola=1},res[1]-res[2])<0, res[1],
> res[2]);
```

$$weight := \frac{1}{2}\,\frac{r\,e^r + r - \sqrt{r^2\,e^{(2\,r)} + 2\,r^2\,e^r + r^2 - 4\,e^{(2\,r)}\,vola + 8\,vola\,e^r - 4\,vola}}{(e^r - 2\,e^{(\frac{r}{2})} + 1)\,r}$$

```
> logblurmap0:=(h,v)->subs({vola=v,r=h},weight);
```

$$logblurmap0 := (h,\,v) \to \mathrm{subs}(\{r = h,\,vola = v\},\,weight)$$

```
> s:=1:
> v:=0.4:
> evalf(logblurmap0(s, v));
> evalf(logblurmap0(s/2, v/4));
```

$$0.8333237430$$

$$0.8076858686$$

```
> dadam:=logblur(delta(2,0), 0, vola, 1):
> m2log:=trans(variance(dadam, var(0), 0), 0, 5):
> sigma:=simplify(da(m2log));
```

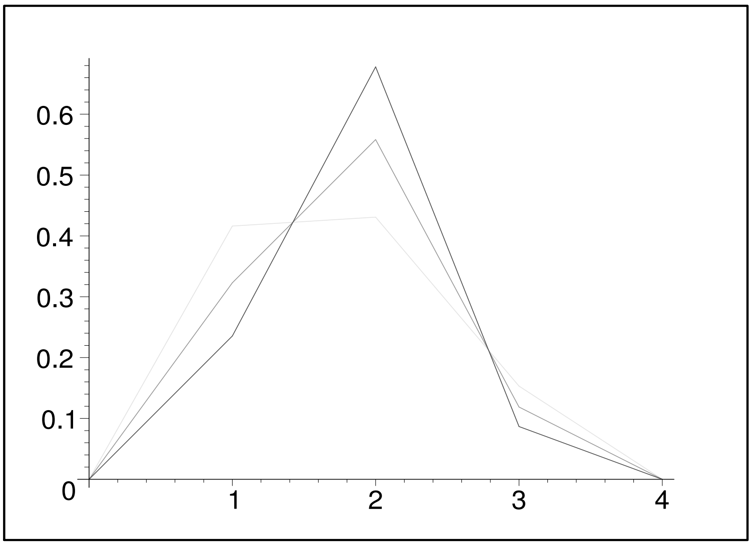
$$\sigma := vola$$

```
> sigma2:=simplify(da(zoom(m2log,0)));
```

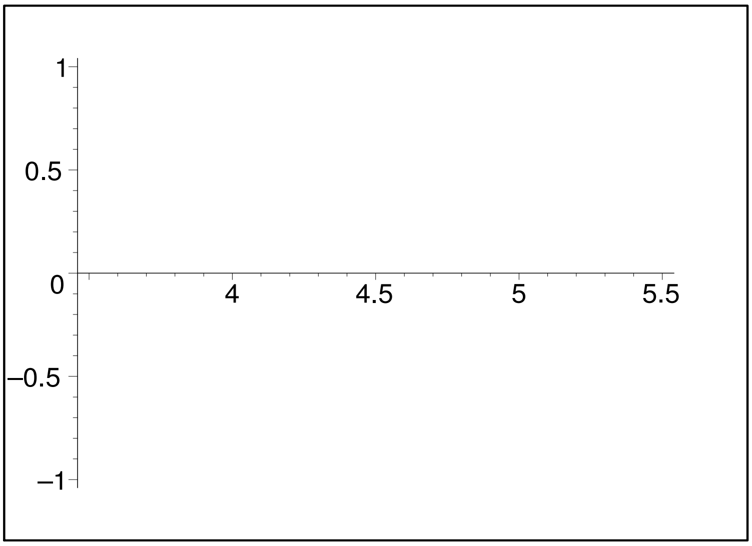
$$\sigma 2 := vola$$

```
> myblur:=(dadim,v)->logblur(dadim, 0, v, 1);
> plot([
> sample(x0,myblur(delta(2,0), 0.3), 0,5),
> sample(x0,myblur(delta(2,0), 0.4), 0,5),
> sample(x0,myblur(delta(2,0), 0.5), 0,5)]);
```

$$myblur := (dadim,\,v) \to \mathrm{logblur}(dadim,\,0,\,v,\,1)$$




```
> plot(sample(zoom(x0,0),zoom(myblur(delta(2,0), 0.4),0),0,10));
```



12 Moments

$\int x^n \, \mathrm{f}(x) \, dx$

12.1 Moment definition

```
> moment:=(dadim, n, dir)-> integ(mult(power(var(dir), cons(n)),dadim),
> dir);;
```

$$moment := (dadim,\ n,\ dir) \to \text{integ}(\text{mult}(\text{power}(\text{var}(dir),\ \text{cons}(n)),\ dadim),\ dir)$$

12.2 Moments of the bell curve

```
> dadam:= blur(delta(2,0),0):
```

The first moment:

```
> m1:=trans(moment(dadam, 1, 0), 0, 10):
> da(m1);
```

$$2.00$$

```
> da(zoom(m1,0));
```

$$2.000000000$$

The second moment:

```
> m2:=trans(moment(dadam, 2, 0), 0, 10):
> da(m2);
```

$$4.50$$

```
> da(zoom(m2,0));
```

$$4.500000000$$

The third moment:

```
> m3:=trans(moment(dadam, 3, 0), 0, 10):
> da(m3);
```

$$11.00$$

```
> da(zoom(m3,0));
```

$$11.00000000$$

The forth moment:

```
> m4:=trans(moment(dadam, 4, 0), 0, 10):
> da(m4);
```

$$28.50$$

```
> da(zoom(m4,0));
```

$$28.68750000$$

12.3 Moments of the hat function

The hat function has 0 stable moments:

```
> dadam:=hat(delta(1,0),0):
> m1:=trans(moment(dadam, 1, 0),0, 10):
> da(m1);
```

$$1$$

```
> da(zoom(m1,0));
```

$$\frac{3}{4}$$

```
> hat2:=(dadim) -> [
> (da(dadim)+da(trans(dadim,0,1)))/2,
> (dir,len) -> hat2(trans(dadim,dir,len)),
> (dir)->mult(cons(1/2),
> hat2(zoom(dadim,dir))+hat2(trans(zoom(dadim,dir),dir,1)))];
```

$$hat2 := dadim \to [\frac{1}{2}\,\mathrm{da}(dadim) + \frac{1}{2}\,\mathrm{da}(\mathrm{trans}(dadim,\,0,\,1)),$$
$$(dir,\,len) \to \mathrm{hat2}(\mathrm{trans}(dadim,\,dir,\,len)),\, dir \to$$
$$\mathrm{mult}(\mathrm{cons}(\frac{1}{2}),\,\mathrm{hat2}(\mathrm{zoom}(dadim,\,dir)) + \mathrm{hat2}(\mathrm{trans}(\mathrm{zoom}(dadim,\,dir),\,dir,\,1)))]$$

```
> dadam:=hat2(delta(1,0),0):
> m1:=trans(moment(dadam, 1, 0), 0, 10):
> da(m1);
```

$$\frac{1}{2}$$

```
> da(zoom(m1,0));
```

$$\frac{1}{2}$$

13 Hilbert Curve

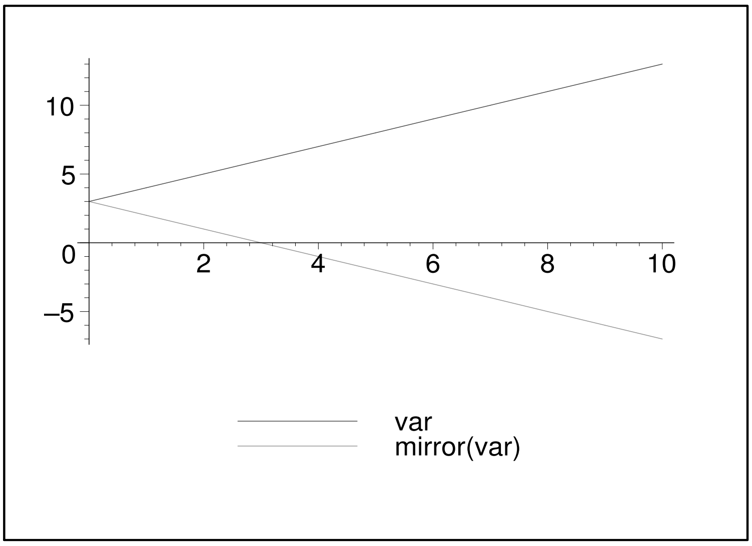
$(M\,f)(x) = \mathrm{f}(-x)$

13.1 Mirror operator

```
> mirror:=(dadim, mdir) -> [
> da(dadim),
> (dir, len) -> piecewise(dir=mdir,
> mirror(trans(dadim, dir, -len), mdir),
> mirror(trans(dadim, dir, len), mdir)),
> dir -> mirror(zoom(dadim,dir), mdir)];
```

$$mirror := (dadim,\,mdir) \to [\mathrm{da}(dadim), (dir,\,len) \to \mathrm{piecewise}(dir = mdir,$$
$$\mathrm{mirror}(\mathrm{trans}(dadim,\,dir,\,-len),\,mdir),\,\mathrm{mirror}(\mathrm{trans}(dadim,\,dir,\,len),\,mdir)),$$
$$dir \to \mathrm{mirror}(\mathrm{zoom}(dadim,\,dir),\,mdir)]$$

```
> dadam:=x0+cons(3):
> plot([sample(x0,dadam,0,11), sample(x0,mirror(dadam, 0),0,11)],
> legend=["var", "mirror(var)"]);
```



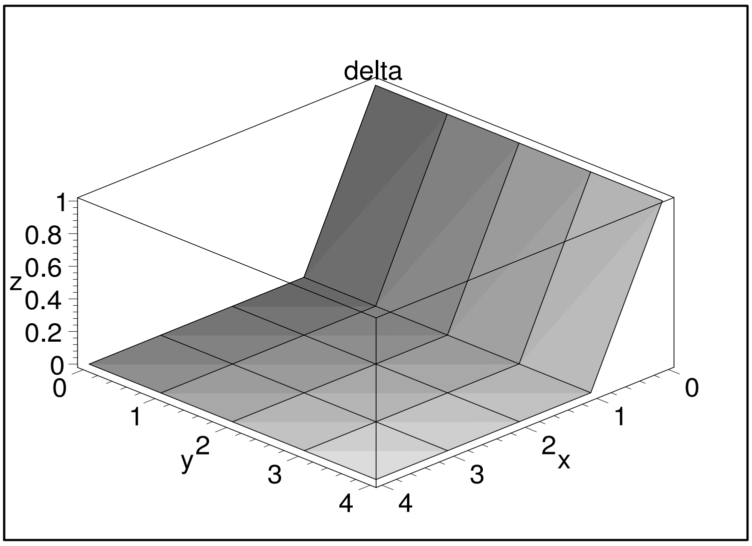
$(R\,f)(x,\,y) = \mathrm{f}(y,\,-x)$

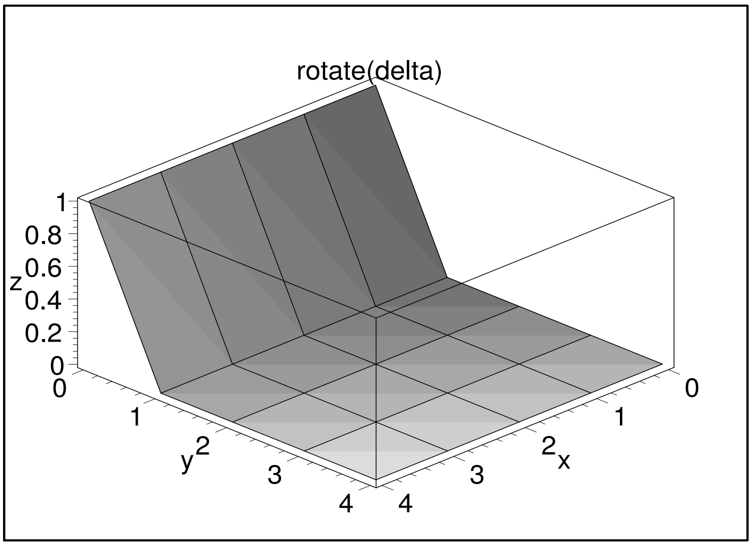
13.2 Rotate operator

```
> rotate:=(dadim) -> [
> da(dadim),
> (dir, len) -> piecewise(
> dir=0, rotate(trans(dadim, 1, len)),
> dir=1, rotate(trans(dadim, 0, -len)),
> rotate(trans(dadim, dir, len))),
> dir -> piecewise(
> dir=0, rotate(zoom(dadim, 1)),
> dir=1, rotate(zoom(dadim, 0)),
> rotate(zoom(dadim, dir)))];
```

$rotate := dadim \rightarrow [\mathrm{da}(dadim), (dir,\,len) \rightarrow \mathrm{piecewise}(dir = 0, \mathrm{rotate}(\mathrm{trans}(dadim,\,1,\,len)), dir = 1, \mathrm{rotate}(\mathrm{trans}(dadim,\,0,\,-len)), \mathrm{rotate}(\mathrm{trans}(dadim,\,dir,\,len))), dir \rightarrow \mathrm{piecewise}(dir = 0, \mathrm{rotate}(\mathrm{zoom}(dadim,\,1)),\; dir = 1, \mathrm{rotate}(\mathrm{zoom}(dadim,\,0)), \mathrm{rotate}(\mathrm{zoom}(dadim,\,dir)))]$

```
> dadam:= delta(0,0):
> M0:= sample3d(dadam,5,5):
> M1:= sample3d(rotate(dadam),5,5):
> plot3d((i,j)->M0[i+1,j+1], 0..4, 0..4, grid=[5,5], axes=boxed,
> title="delta", labels=["x","y","z"]);
> plot3d((i,j)->M1[i+1,j+1], 0..4, 0..4, grid=[5,5], axes=boxed,
> title="rotate(delta)",labels=["x","y","z"]);
```





13.3 Hilbert Curve operator

Li =

Th L0 = L1 My R **T**x

Th L1 = L2 **Tx**

Th L2 = L3 R My **T**y^-1

Th L3 = L0 R R **T**h

Zh Li = L0 My R Li **Z**x **Z**y

```
> hilbert:= (dadim, state) -> [
> da(dadim),
> (dir, len) ->
> piecewise(dir=2,
> piecewise(
> len=1,
> piecewise(
> state=0, hilbert(mirror(rotate(trans(dadim, 0, 1)), 1), 1),
> state=1, hilbert(trans(dadim, 0, 1), 2),
> state=2, hilbert(rotate(mirror(trans(dadim, 1, -1), 1)), 3),
> state=3, hilbert(rotate(rotate(trans(dadim, 2, 1))), 0)),
> len=-1,
> piecewise(
> state=0, hilbert(trans(rotate(rotate(dadim)), 2, -1),3),
> state=1,
> hilbert(trans(rotate(rotate(rotate(mirror(dadim,1)))),0,-1),
0),
> state=2, hilbert(trans(dadim, 0, -1), 1),
> state=3,
> hilbert(trans(mirror(rotate(rotate(rotate(dadim))),1),1,1), 2))
> ),
> hilbert(trans(dadim, dir, len), state)),
> dir ->
> piecewise(dir=2,
> hilbert(mirror(rotate(hilbert(zoom(zoom(dadim,0),1),state)),1),0),
> hilbert(zoom(dadim, dir), state))];
```

$$\mathit{hilbert} := (\mathit{dadim},\ \mathit{state}) \to [\mathrm{da}(\mathit{dadim}), (\mathit{dir},\ \mathit{len}) \to \mathrm{piecewise}(\mathit{dir} = 2, \mathrm{piecewise}(\mathit{len} = 1,$$
$$\mathrm{piecewise}(\mathit{state} = 0,\ \mathrm{hilbert}(\mathrm{mirror}(\mathrm{rotate}(\mathrm{trans}(\mathit{dadim},\ 0,\ 1)),\ 1),\ 1),\ \mathit{state} = 1,$$
$$\mathrm{hilbert}(\mathrm{trans}(\mathit{dadim},\ 0,\ 1),\ 2),\ \mathit{state} = 2,$$
$$\mathrm{hilbert}(\mathrm{rotate}(\mathrm{mirror}(\mathrm{trans}(\mathit{dadim},\ 1,\ -1),\ 1)),\ 3),\ \mathit{state} = 3,$$
$$\mathrm{hilbert}(\mathrm{rotate}(\mathrm{rotate}(\mathrm{trans}(\mathit{dadim},\ 2,\ 1))),\ 0)),\ \mathit{len} = -1, \mathrm{piecewise}(\mathit{state} = 0,$$
$$\mathrm{hilbert}(\mathrm{trans}(\mathrm{rotate}(\mathrm{rotate}(\mathit{dadim})),\ 2,\ -1),\ 3),\ \mathit{state} = 1,$$
$$\mathrm{hilbert}(\mathrm{trans}(\mathrm{rotate}(\mathrm{rotate}(\mathrm{rotate}(\mathrm{mirror}(\mathit{dadim},\ 1)))),\ 0,\ -1),\ 0),\ \mathit{state} = 2,$$
$$\mathrm{hilbert}(\mathrm{trans}(\mathit{dadim},\ 0,\ -1),\ 1),\ \mathit{state} = 3,$$
$$\mathrm{hilbert}(\mathrm{trans}(\mathrm{mirror}(\mathrm{rotate}(\mathrm{rotate}(\mathrm{rotate}(\mathit{dadim}))),\ 1),\ 1,\ 1),\ 2))),$$
$$\mathrm{hilbert}(\mathrm{trans}(\mathit{dadim},\ \mathit{dir},\ \mathit{len}),\ \mathit{state})),\ \mathit{dir} \to \mathrm{piecewise}(\mathit{dir} = 2,$$
$$\mathrm{hilbert}(\mathrm{mirror}(\mathrm{rotate}(\mathrm{hilbert}(\mathrm{zoom}(\mathrm{zoom}(\mathit{dadim},\ 0),\ 1),\ \mathit{state})),\ 1),\ 0),$$
$$\mathrm{hilbert}(\mathrm{zoom}(\mathit{dadim},\ \mathit{dir}),\ \mathit{state}))]$$

```
> hx:=stepwise(hilbert(x0, 0)):
> hy:=stepwise(hilbert(x1, 0)):
> [da(hx), da(hy)];
```

$$[0,\ 0]$$

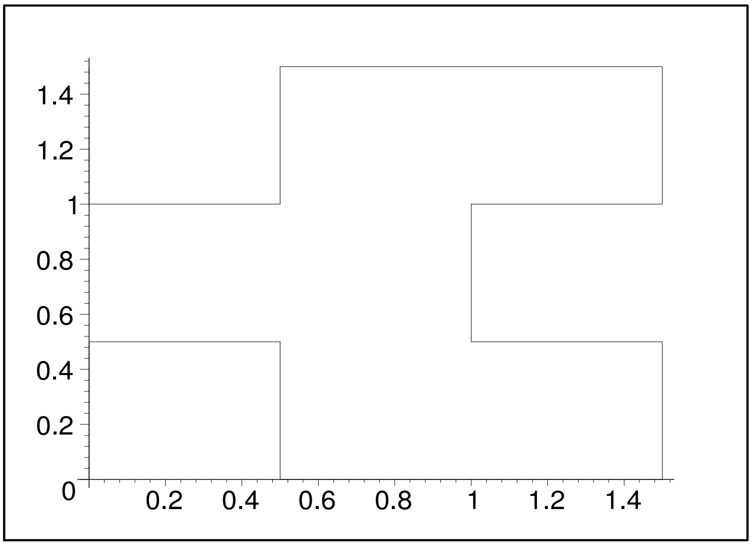
```
> sample(hx, hy, 2, 4);
```

$$[[0, 0], [1, 0], [1, 1], [0, 1]]$$

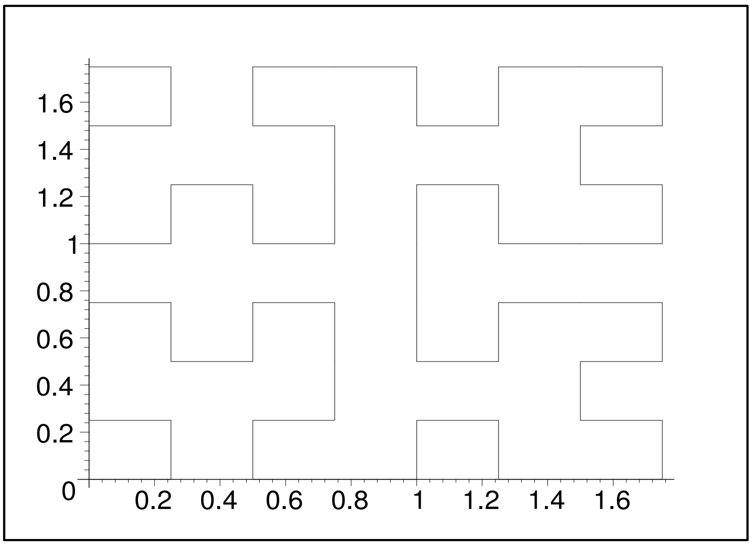
```
> sample(mirror(trans(hx,2,3),2), mirror(trans(hy,2,3),2), 2, 4);
```

$$[[0, 1], [1, 1], [1, 0], [0, 0]]$$

```
> plot(sample(zoom(hx,2), zoom(hy,2), 2, 16));
```

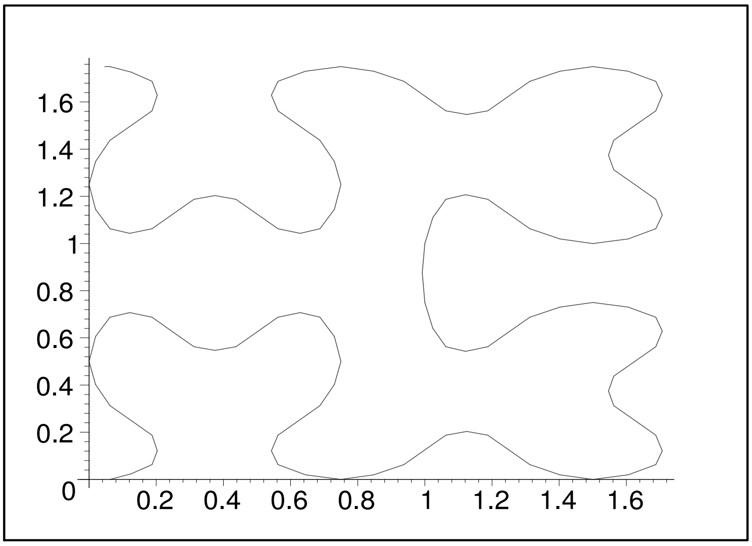



```
> plot(sample(zoom(zoom(hx,2),2), zoom(zoom(hy,2),2),2,64));
```

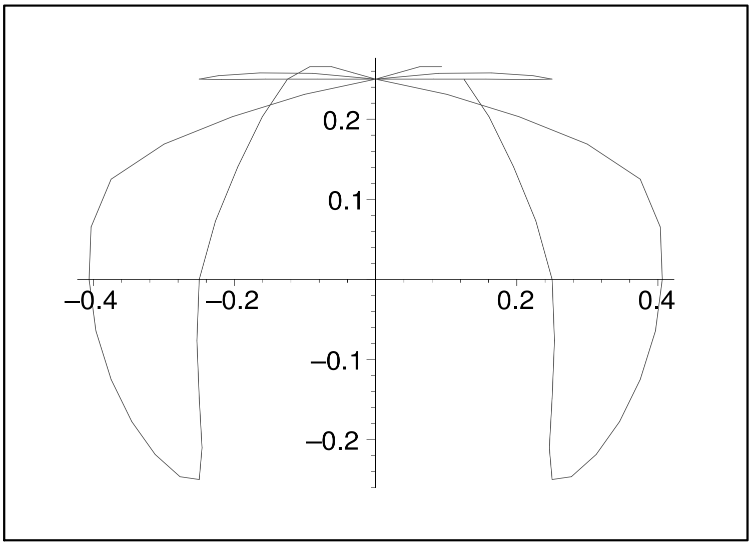


13.4 Variations

```
> plot(sample(
> predict(blur(zoom(zoom(hx,2),2),2),2),
> predict(blur(zoom(zoom(hy,2),2),2),2),2,128));
```




```
> plot(sample(
> predict(predict(dif(blur(zoom(hx,2),2),2),2),2),
> predict(predict(dif(blur(zoom(hy,2),2),2),2),2),2,64));
```

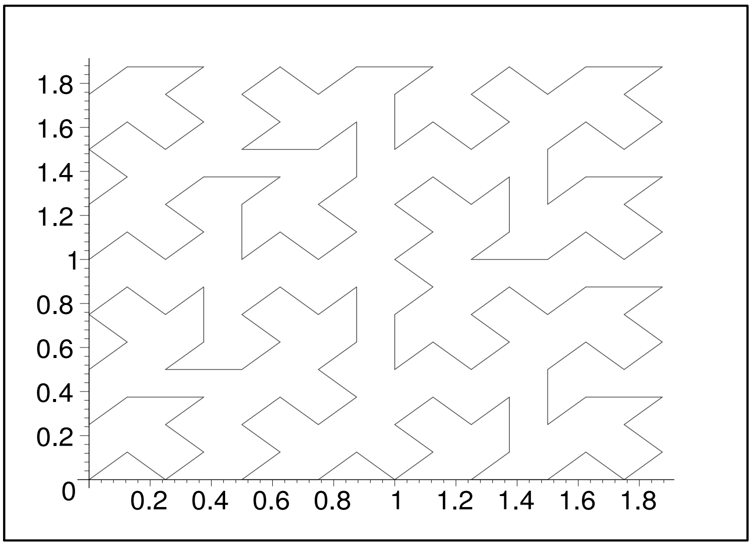


Z* =

T Z* = Z* **T** **T**

Z Z* = Id

```
> invzoom:=(dadim,mdir) -> [
> da(dadim),
> (dir,len)-> piecewise(dir=mdir,
> invzoom(trans(dadim,dir,2*len),dir),
> invzoom(trans(dadim,dir,len)),mdir),
> (dir) -> piecewise(dir=mdir,
> dadim,
> invzoom(zoom(dadim,dir),mdir))]:
> plot(sample(
> invzoom(zoom(zoom(zoom(hx,2),2),2),2),
> invzoom(zoom(zoom(zoom(hy,2),2),2),2),2,128));
```



14 Sheer

$(S^a f)(x, y) = \mathrm{f}(x + a, y)$

14.1 Definition

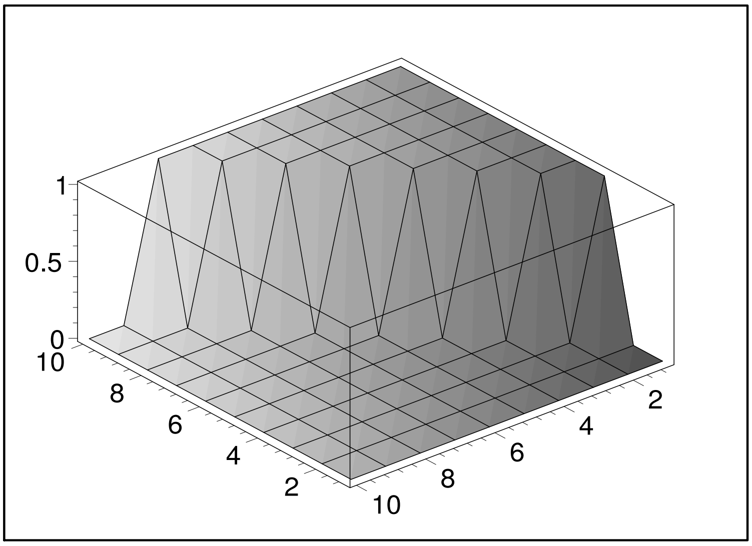
```
> dashift:=(dadim, mdir, mlen)->
> (floor(mlen+1)-mlen)*
> da(trans(dadim, mdir, floor(mlen))) +
> (mlen-floor(mlen))*
> da(trans(dadim, mdir, floor(mlen+1)));
```

$$dashift := (dadim,\ mdir,\ mlen) \rightarrow$$
$$(\mathrm{floor}(mlen+1) - mlen)\,\mathrm{da}(\mathrm{trans}(dadim,\ mdir,\ \mathrm{floor}(mlen)))$$
$$+\,(mlen - \mathrm{floor}(mlen))\,\mathrm{da}(\mathrm{trans}(dadim,\ mdir,\ \mathrm{floor}(mlen+1)))$$

```
> sheer:=(dadim, mdir, field)->[
> dashift(dadim,mdir,da(field)),
> (dir,len)->sheer(trans(dadim,dir,len), mdir, trans(field, dir,
> len)),
> dir -> sheer(zoom(dadim,dir), mdir, zoom(field,dir))];
```

$$sheer := (dadim,\ mdir,\ field) \rightarrow [\mathrm{dashift}(dadim,\ mdir,\ \mathrm{da}(field)),$$
$$(dir,\ len) \rightarrow \mathrm{sheer}(\mathrm{trans}(dadim,\ dir,\ len),\ mdir,\ \mathrm{trans}(field,\ dir,\ len)),$$
$$dir \rightarrow \mathrm{sheer}(\mathrm{zoom}(dadim,\ dir),\ mdir,\ \mathrm{zoom}(field,\ dir))]$$

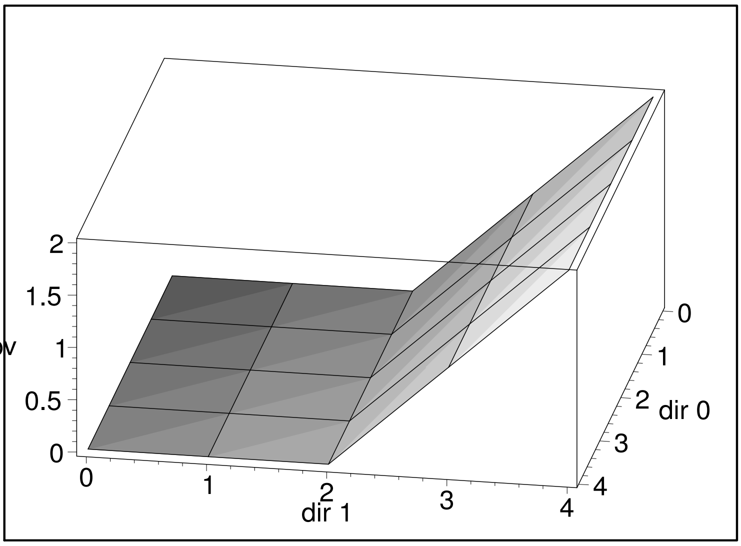
```
> dadam:=sheer(integ(delta(1,0),0),0,-x1):
> M0:=sample3d(dadam,10,10):
> plot3d((i,j)->M0[i,j],1..10,1..10,grid=[10,10],axes=boxed,orientation=
> [140,40]);
```



14.2 Asian option

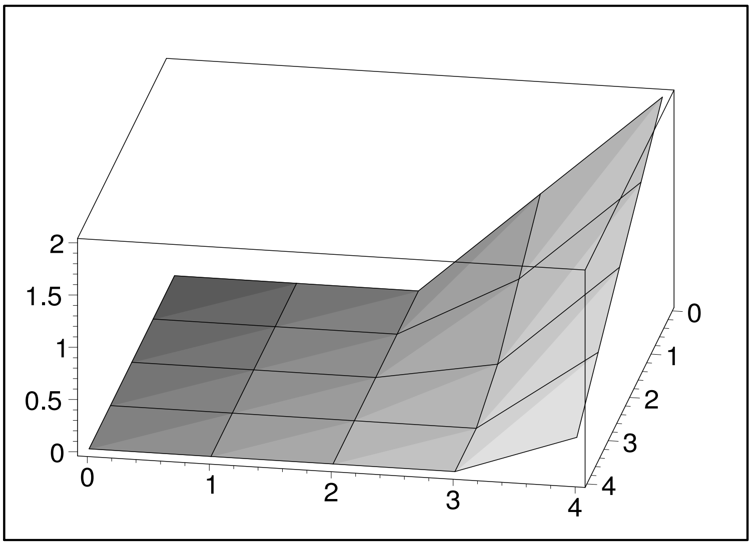
```
> pv:=multop([x1-cons(2),cons(0)], max):
> M0:=sample3d(pv, 5, 5);
> plot3d((i,j)->M0[i+1,j+1], 0..4, 0..4, grid=[5,5],
> axes=boxed,orientation=[10,50],labels=["dir 0","dir 1","pv"]);
```

$$M0 := \begin{bmatrix} 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$



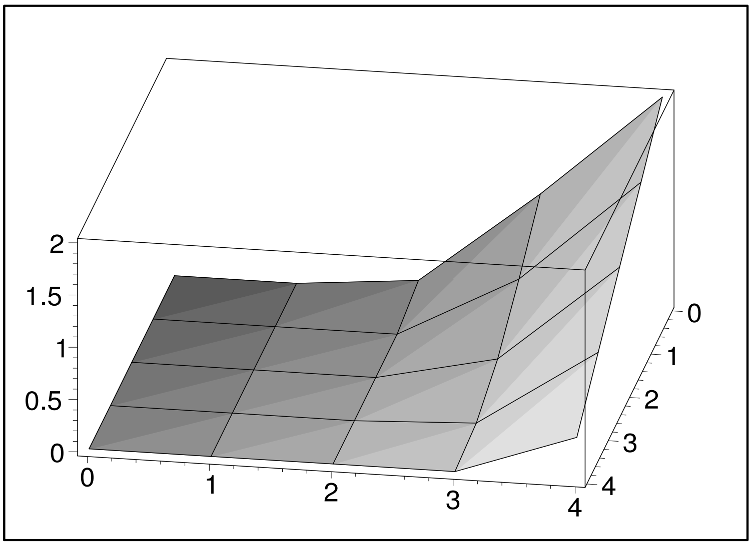

```
> pv1:=sheer(pv, 1, -0.4*x0):
> M1:=sample3d(pv1,5,5);
> plot3d((i,j)->M1[i+1,j+1], 0..4, 0..4, grid=[5,5], axes=boxed,
> orientation=[10,50]);
```

$$M1 := \begin{bmatrix} 0. & 0. & 0. & 1. & 2. \\ 0. & 0. & 0. & 0.6 & 1.6 \\ 0. & 0. & 0. & 0.2 & 1.2 \\ 0. & 0. & 0. & 0. & 0.8 \\ 0. & 0. & 0. & 0. & 0.4 \end{bmatrix}$$




```
> pv2:=blur(pv1, 0):
> M2:=sample3d(pv2,5,5);
> plot3d((i,j)->M2[i+1,j+1], 0..4, 0..4, grid=[5,5], axes=boxed,
> orientation=[10,50]);
```

$$M2 := \begin{bmatrix} 0. & 0. & 0.100 & 1.000 & 2.000 \\ 0. & 0. & 0. & 0.600 & 1.600 \\ 0. & 0. & 0. & 0.250 & 1.200 \\ 0. & 0. & 0. & 0.050 & 0.800 \\ 0. & 0. & 0. & 0. & 0.400 \end{bmatrix}$$



Dadim - Multiscale Calculus (www.dadim.de)

Stefan Dirnstorfer