Algorithmen des wissenschaftlichen Rechnens II
Übungsblatt 4
Hierarchische Basis

```
> restart:
  with(plots):
  with(LinearAlgebra):
Warning, the name changecoords has been redefined
```

Ein paar generelle Optionen zum Plotten:

```
> poptions := thickness=2;
  poptions2d := thickness=2,
                axes=boxed,
                labels=["x1", "x2", ""],
                orientation=[-120,45];
```

$$poptions := thickness = 2$$

$$poptions2d := thickness = 2,\ axes = boxed,\ labels = [\text{"x1"}, \text{"x2"}, \text{""}],\ orientation = [-120, 45]$$

Basisfunktionen erzeugen

Eine Funktion zum Erzeugen der Hut-Basisfunktionen $\phi_{l,i}(x)$, in Abhängigkeit von Level l und Index i:

```
> phi_li := (l,i) -> unapply(
                       piecewise(x <= (i-1)/2^l, 0,
                                 x <= i/2^l,     2^l*x+1-i,
                                 x <= (i+1)/2^l, i+1-2^l*x,
                                 0),
                     x);
```

$$phi\_li := (l, i) \to \text{unapply}\left(\text{piecewise}\left(x \le \frac{i-1}{2^l}, 0, x \le \frac{i}{2^l}, 2^l x + 1 - i, x \le \frac{i+1}{2^l}, i + 1 - 2^l x, 0\right), x\right)$$

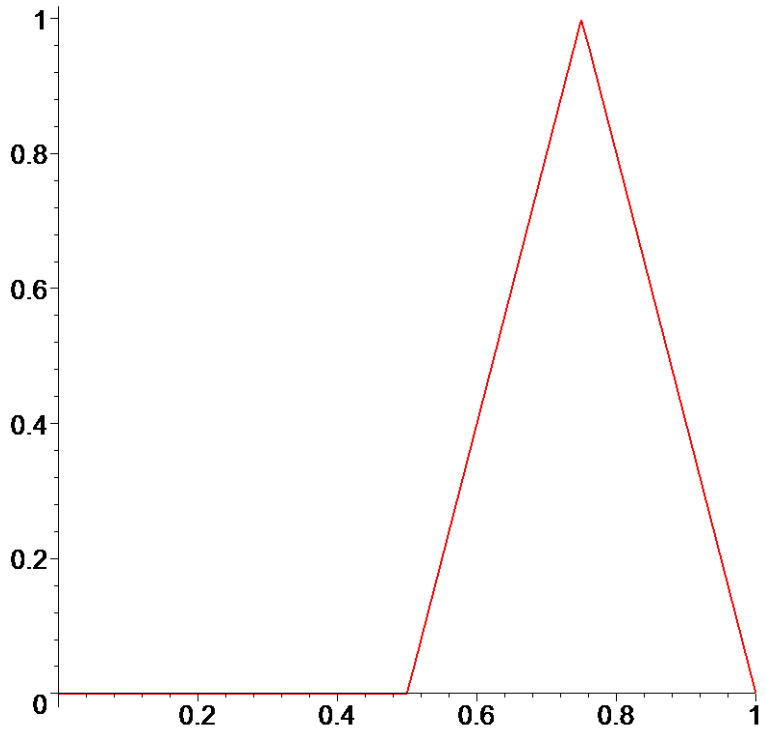
Nachher wird es praktischer sein, die Parameter in eine Liste $[l, i]$ zu verpacken, und Folgendes zu verwenden:

```
> phi_ind := ind -> phi_li(ind[1], ind[2]);
```

$$phi\_ind := ind \to \text{phi\_li}(ind_1, ind_2)$$

```
> phi_ind([2,3]);
> plot(%, 0..1, poptions);
```

$$x \to \text{piecewise}\left(x \le \frac{1}{2}, 0, x \le \frac{3}{4}, 4x - 2, x \le 1, 4 - 4x, 0\right)$$



Nun das ganze *d*-dimensional. Level *lv* und Index *iv* sind nun *d*-komponentige Listen.

```
> phi_LI := (lv,iv) -> unapply(

  mul(phi_li(lv[j],iv[j])(x||j),j=1..nops(lv)),
                              [seq(x||j, j=1..nops(lv))]);
```

$$phi\_LI := (lv, iv) \rightarrow \text{unapply}(\text{mul}(\text{phi\_li}(lv_j, iv_j)(x||j), j = 1 \,..\, \text{nops}(lv)), [\text{seq}(x||j, j = 1 \,..\, \text{nops}(lv))])$$

Auch die bekommen wir später als Paare:

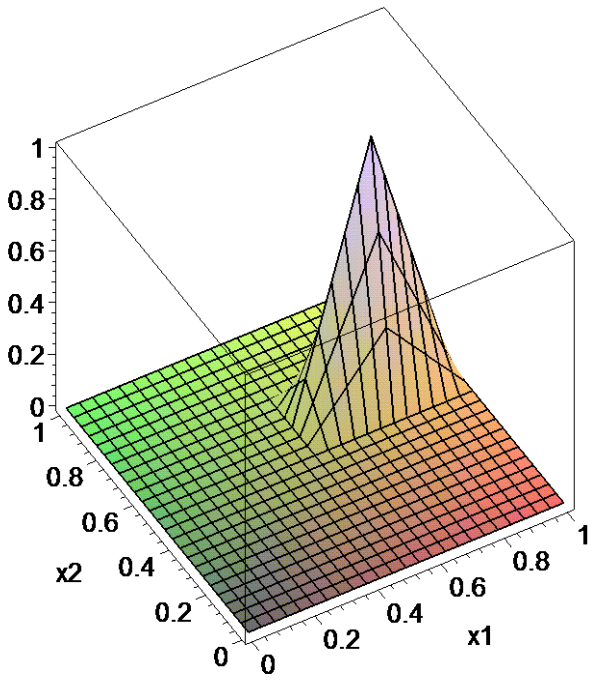
```
> phi_IND := indv -> phi_LI(indv[1], indv[2]);
```

$$phi\_IND := indv \rightarrow \text{phi\_LI}(indv_1, indv_2)$$

```
> phi_IND([[2,3],[3,5]]);
> plot3d(%, 0..1, 0..1, poptions2d);
```

$$(x1, x2) \rightarrow \text{piecewise}\left(x1 \le \frac{1}{2}, 0, x1 \le \frac{3}{4}, 4\,x1 - 2, x1 \le 1, 4 - 4\,x1, 0\right)$$

$$\text{piecewise}\left(x2 \le \frac{1}{2}, 0, x2 \le \frac{5}{8}, 8\,x2 - 4, x2 \le \frac{3}{4}, 6 - 8\,x2, 0\right)$$



>

>

Eindimensionales Hierarchisieren

Basis erzeugen

Die Basis von V_n wird repräsentiert durch eine Liste von Level-Index-Paaren [l,i].
Zunächst für ein Level:

```
> hier_basis_level := l -> [seq([l,2*i-1],i=1..2^l/2)];
```

$$hier\_basis\_level := l \rightarrow \left[\mathrm{seq}\left([l,\ 2\,i-1],\ i = 1\ ..\ \frac{1}{2}\,2^{l} \right) \right]$$

Und nun eine Schleife über die Level:

```
> hier_basis := n  -> [seq(op(hier_basis_level(l)),l=1..n)];
```

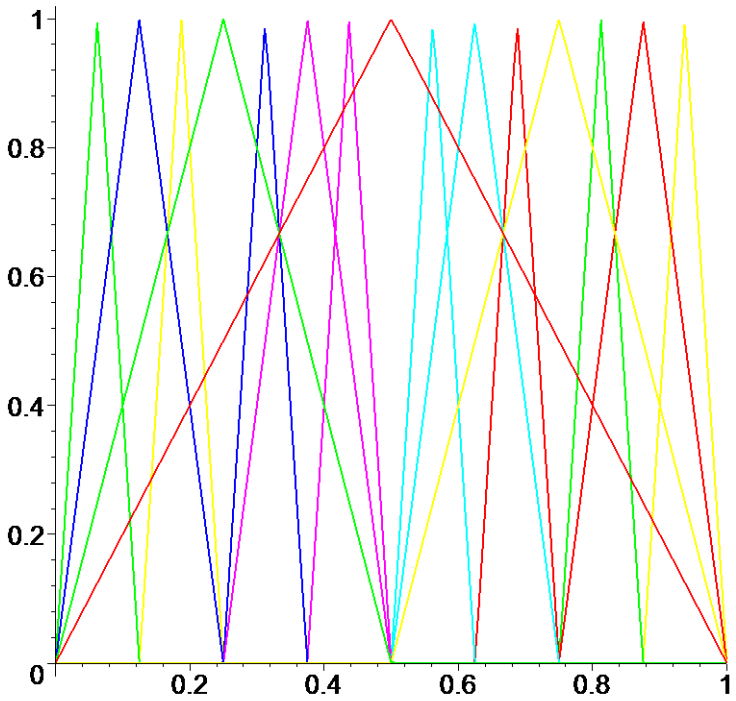
$$hier\_basis := n \rightarrow [\mathrm{seq}(\mathrm{op}(\mathrm{hier\_basis\_level}(l)),\ l = 1\ ..\ n)]$$

```
> hier_basis(3);
```

$$[[1,\ 1],\ [2,\ 1],\ [2,\ 3],\ [3,\ 1],\ [3,\ 3],\ [3,\ 5],\ [3,\ 7]]$$

Die Basisfunkionen selber bekommen wir, indem wir phi_ind auf die Level/Index-Liste anwenden:

```
> plot(map(phi_ind,hier_basis(4)),0..1,poptions);
```




```
>
```

Beispielfunktion

```
> u_test := x -> 4*x*(1-x);
```

$$u\_test := x \to 4\,x\,(1-x)$$

```
>
```

Überschuss ausrechnen (Differenzenformel)

```
ueberschuss1 := (u, l, i) ->
u(2^(-l)*i)-(u(2^(-l)*(i-1))+u(2^(-l)*(i+1)))/2;
```

$$ueberschuss1 := (u, l, i) \to u(2^{(-l)}\,i) - \frac{1}{2}\,u(2^{(-l)}\,(i-1)) - \frac{1}{2}\,u(2^{(-l)}\,(i+1))$$

```
> ueberschuss1_ind := (u,ind) -> ueberschuss1(u, ind[1],
  ind[2]);
```

$$ueberschuss1\_ind := (u, ind) \to \mathrm{ueberschuss1}(u, ind_1, ind_2)$$

```
> seq(ueberschuss1_ind(u_test, ind), ind=hier_basis(4));
```

$$1, \frac{1}{4}, \frac{1}{4}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}$$

```
>
```

Überschuss ausrechnen (Integral)

```
> l2sp := (f,g) -> int(f(x)*g(x),x=0..1);
```

$$l2sp := (f, g) \to \int_0^1 f(x)\,g(x)\,dx$$

```
> l2norm := f -> sqrt(l2sp(f,f));
```

$$l2norm := f \to \sqrt{\mathrm{l2sp}(f, f)}$$

```
> ueberschuss2 := (u, l, i) -> -2^(-l)/2*l2sp(D[1,1](u),
```

```
   phi_li(l,i));
```

$$ueberschuss2 := (u, l, i) \to -\frac{1}{2}\, 2^{(-l)}\, \mathrm{l2sp}(\mathrm{D}_{1,1}(u), \mathrm{phi\_li}(l, i))$$

```
> ueberschuss2_ind := (u,ind) -> ueberschuss2(u, ind[1],
  ind[2]);
```

$$ueberschuss2\_ind := (u, ind) \to \mathrm{ueberschuss2}(u, ind_1, ind_2)$$

```
> seq(ueberschuss2_ind(u_test, ind), ind=hier_basis(4));
```

$$1, \frac{1}{4}, \frac{1}{4}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}$$

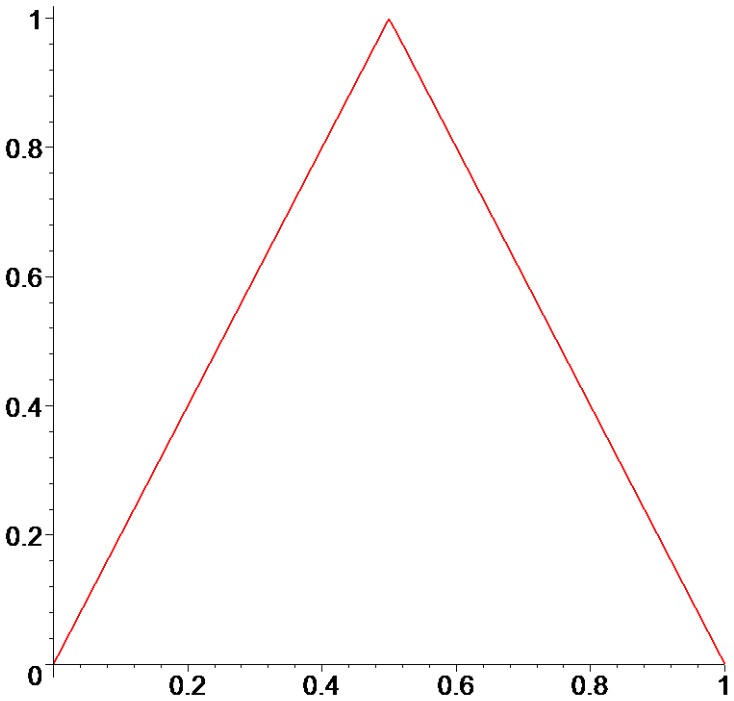
```
>
```

Hierarchische Inkremente

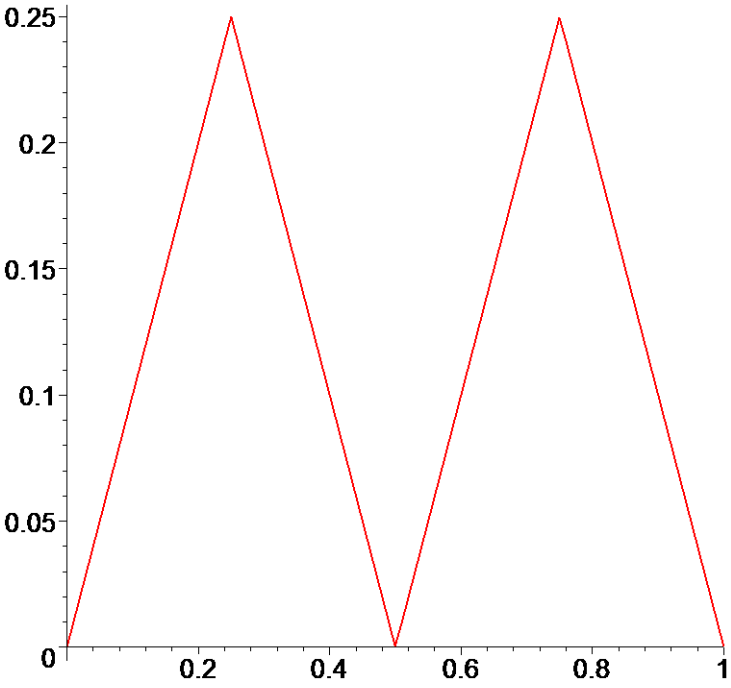
```
> wl := (u, l) -> add(
                        ueberschuss1_ind(u, ind)*phi_ind(ind),
                        ind=hier_basis_level(l));
```

$$wl := (u, l) \to \mathrm{add}(\mathrm{ueberschuss1\_ind}(u, ind)\ \mathrm{phi\_ind}(ind), ind = \mathrm{hier\_basis\_level}(l))$$

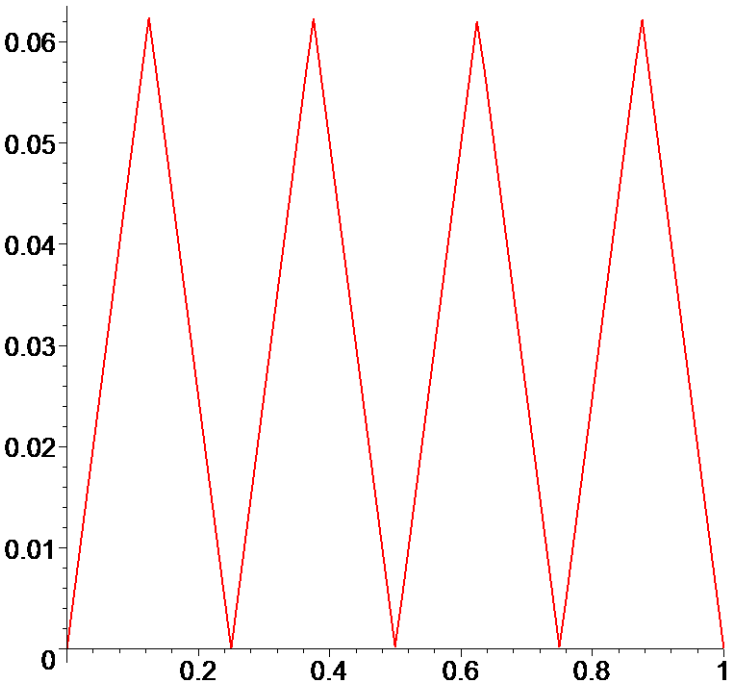
```
> for l from 1 to 3 do
     plot(wl(u_test,l), 0..1,poptions);
     print(l2norm(wl(u_test,l)))
  end do;
```



$$\frac{\sqrt{3}}{3}$$



$$\frac{\sqrt{3}}{12}$$



$$\frac{\sqrt{3}}{48}$$

>

>

Zweidimensionales Hierarchisieren

Basis erzeugen

Hier ist die Basis eine Liste von Level-Index-Paar-Paaren [l1,l2],[i1,i2]].
Zunächst für ein Level:

```
> hier_basis_level := L  -> [seq(seq(

  [[L[1],L[2]],[2*i1-1,2*i2-1]],

  i1=1..2^L[1]/2),i2=1..2^L[2]/2)];
```

hier_basis_level :=

$$L \to \left[\mathrm{seq}\left(\mathrm{seq}\left([[L_1, L_2], [2\,i1 - 1, 2\,i2 - 1]], i1 = 1 .. \frac{1}{2} 2^{L_1} \right), i2 = 1 .. \frac{1}{2} 2^{L_2} \right) \right]$$

Und nun eine Schleife über die Level:

```
> hier_basis := n -> [seq(seq(
                           op(hier_basis_level([l+1-l2,l2])),
                      l2=1..l),l=1..n)];
```

hier_basis :=

$$n \to [\mathrm{seq}(\mathrm{seq}(\mathrm{op}(\mathrm{hier\_basis\_level}([l + 1 - l2, l2])), l2 = 1 .. l), l = 1 .. n)]$$

```
> hier_basis(2);
```

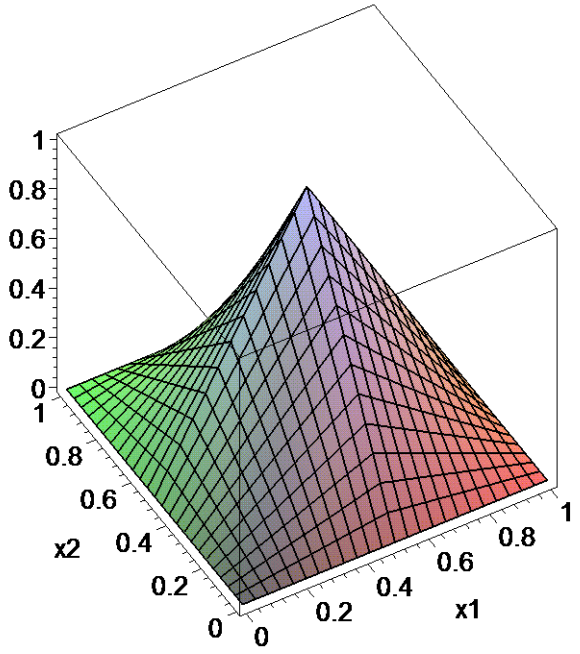
$$[[[1, 1], [1, 1]], [[2, 1], [1, 1]], [[2, 1], [3, 1]], [[1, 2], [1, 1]], [[1, 2], [1, 3]]]$$

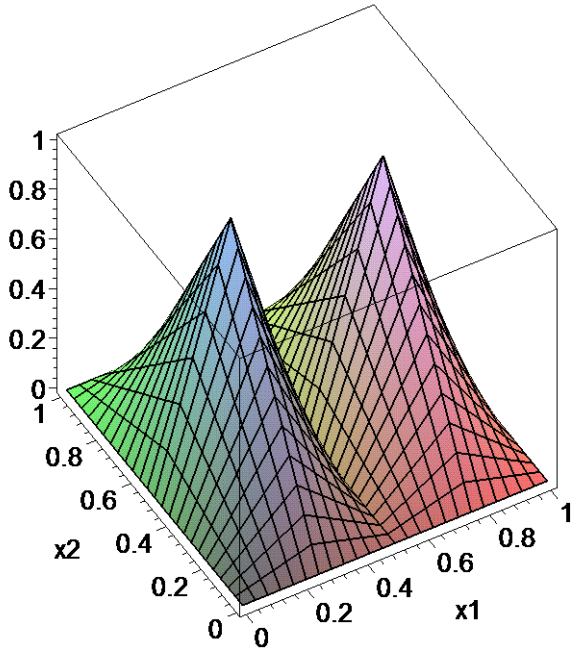
```
> printlevel := 2;
```

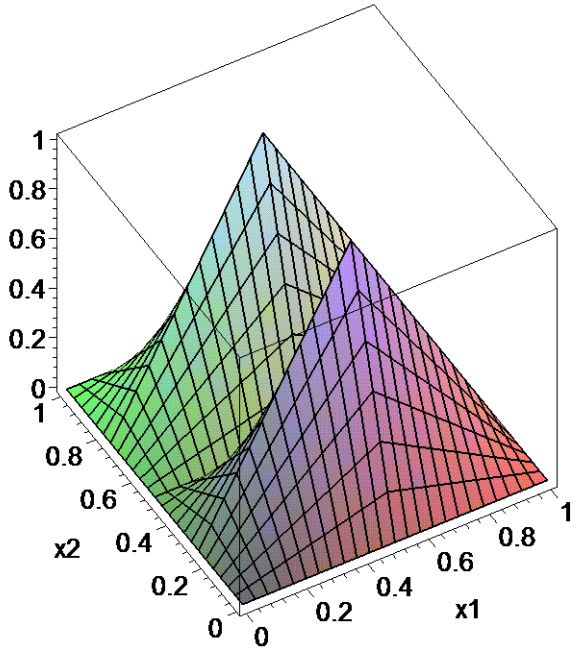
printlevel := 2

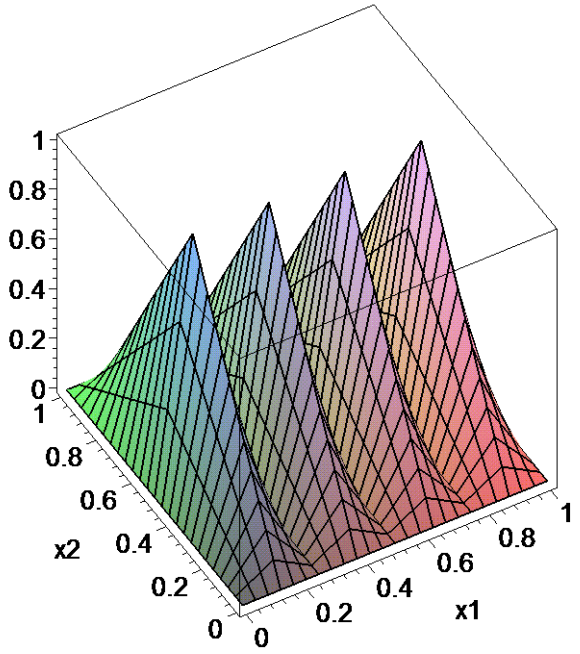
Die Basisfunkionen selber bekommen wir, indem wir phi_ind auf die Level/Index-Liste anwenden:

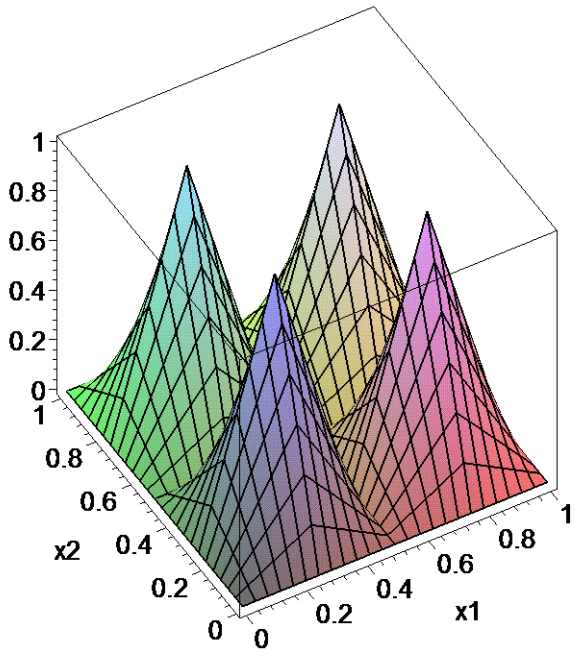
```
> for l from 1 to 3 do
    for l2 from 1 to l do
      plot3d(map(phi_IND,hier_basis_level([l+1-l2,l2])),

             0..1,0..1,poptions2d)
    end do
  end do;
```

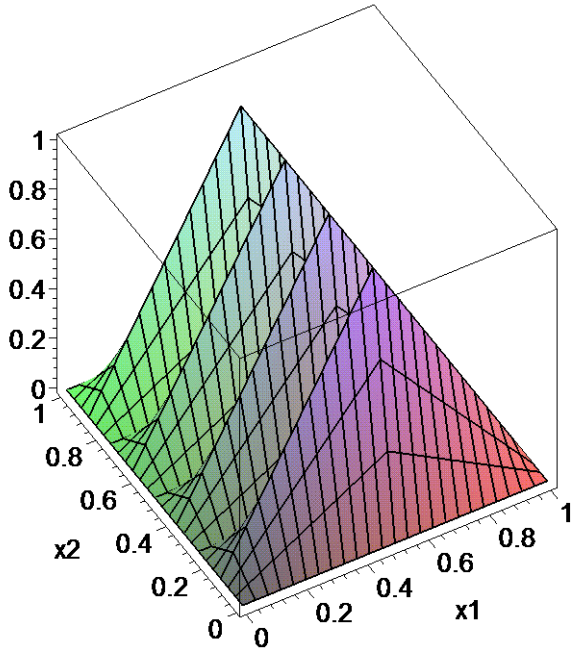

1
0.8
0.6
0.4
0.2
0
x2
0
0.2
0.4
0.6
0.8
1
x1


1
0.8
0.6
0.4
0.2
0
x2
0
0.2
0.4
0.6
0.8
1
x1


1
0.8
0.6
0.4
0.2
0
x2
0
0.2
0.4
0.6
0.8
1
x1


1
0.8
0.6
0.4
0.2
0
x2
0
0.2
0.4
0.6
0.8
1
x1






```
>
```

Beispielfunktion

```
> u_test := (x1,x2)->16*x1*(1-x1)*x2*(1-x2);
```

$$u\_test := (x1, x2) \rightarrow 16\, x1\, (1 - x1)\, x2\, (1 - x2)$$

```
>
```

Überschuss ausrechnen (Differenzenformel)

```
> ueberschuss1 := proc(u, lv, iv)
      local l1,l2,i1,i2,h1,h2,x1,x2,v1;
      l1 := lv[1];   l2 := lv[2];
```

```
    i1 := iv[1];   i2 := iv[2];
    h1 := 2^(-l1); h2 := 2^(-l2);
    x1 := h1*i1;   x2 := h2*i2;
    v1 := x -> u(x1,x) - (u(x1-h1,x)+u(x1+h1,x))/2;
    return v1(x2) - (v1(x2-h2)+v1(x2+h2))/2;
end proc:
```

```
> ueberschuss1_ind := (u,ind) -> ueberschuss1(u, ind[1],
  ind[2]);
```

$$ueberschuss1\_ind := (u, ind) \rightarrow \text{ueberschuss1}(u, ind_1, ind_2)$$

```
> seq(ueberschuss1_ind(u_test, ind),
      ind=hier_basis(4));
```

$$1, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}$$
$$\frac{1}{64}, \frac{1}{64}$$
$$\frac{1}{64}$$

```
>
```

Überschuss ausrechnen (Integral)

```
> l2sp := (f,g) ->
  int(int(f(x1,x2)*g(x1,x2),x1=0..1),x2=0..1);
```

$$l2sp := (f, g) \rightarrow \int_0^1 \int_0^1 f(x1, x2)\, g(x1, x2)\, dx1\, dx2$$

```
> l2norm := f -> sqrt(l2sp(f,f));
```

$$l2norm := f \rightarrow \sqrt{\text{l2sp}(f, f)}$$

```
> ueberschuss2 := (u, lv, iv) ->
                  2^(-lv[1]-lv[2])/4*
                  l2sp(D[1,1,2,2](u), phi_LI(lv,iv));
```

$$ueberschuss2 := (u, lv, iv) \rightarrow \frac{1}{4} 2^{(-lv_1 - lv_2)}\, \text{l2sp}(\text{D}_{1,1,2,2}(u), \text{phi\_LI}(lv, iv))$$

```
> ueberschuss2_ind := (u,ind) -> ueberschuss2(u, ind[1],
  ind[2]);
```

$$ueberschuss2\_ind := (u, ind) \rightarrow \text{ueberschuss2}(u, ind_1, ind_2)$$

```
> seq(ueberschuss2_ind(u_test, ind), ind=hier_basis(4));
```

$$1, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}$$
$$\frac{1}{64}, \frac{1}{64}$$
$$\frac{1}{64}$$

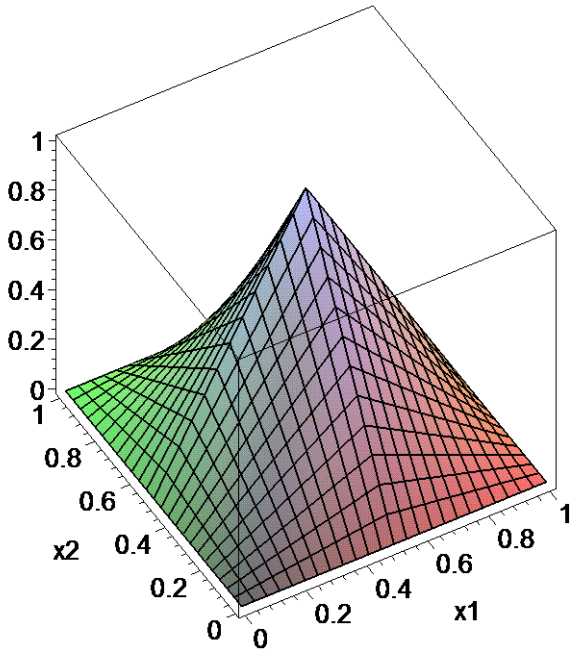
```
>
```

Hierarchische Inkremente

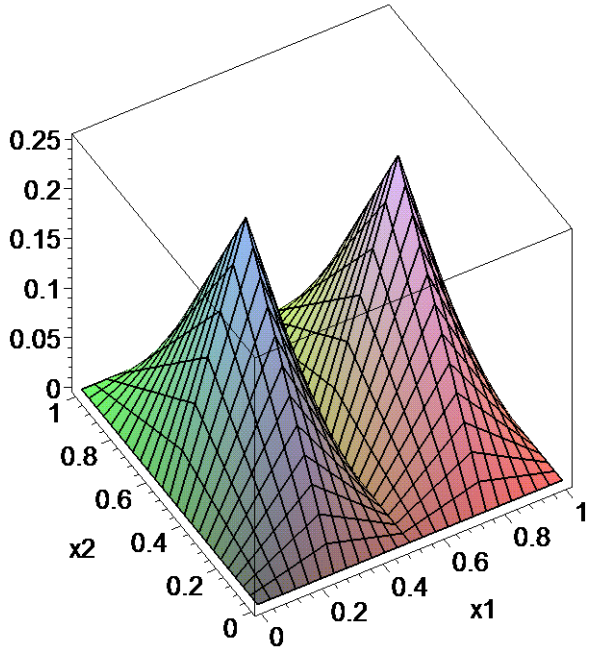
```
> wl := (u, l) -> add(
                        ueberschuss1_ind(u, ind)*phi_IND(ind),
                        ind=hier_basis_level(l));
```

$$wl := (u, l) \rightarrow \mathrm{add}(\mathrm{ueberschuss1\_ind}(u, ind)\ \mathrm{phi\_IND}(ind), ind = \mathrm{hier\_basis\_level}(l))$$

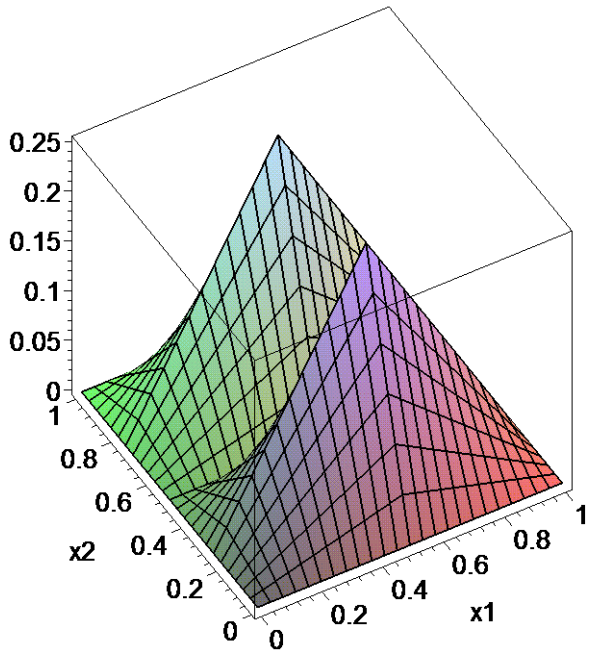
```
> for l from 1 to 3 do
     for l2 from 1 to l do
        plot3d(wl(u_test,[l+1-l2,l2]),
  0..1,0..1,poptions2d);
        print(l2norm(wl(u_test,[l+1-l2,l2])))
     end do
  end do;
```



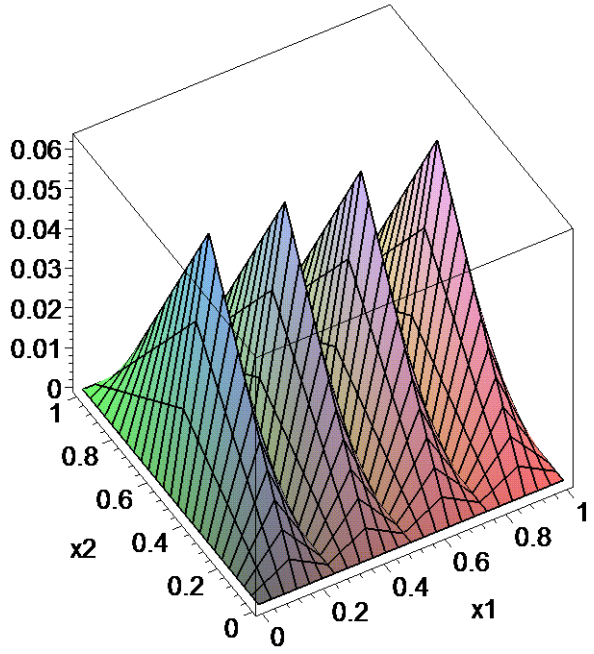
$$\frac{1}{3}$$



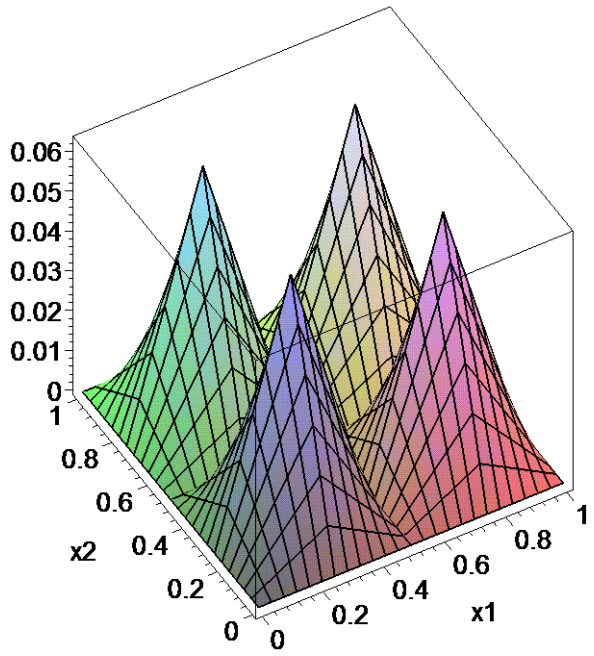
$\frac{1}{12}$



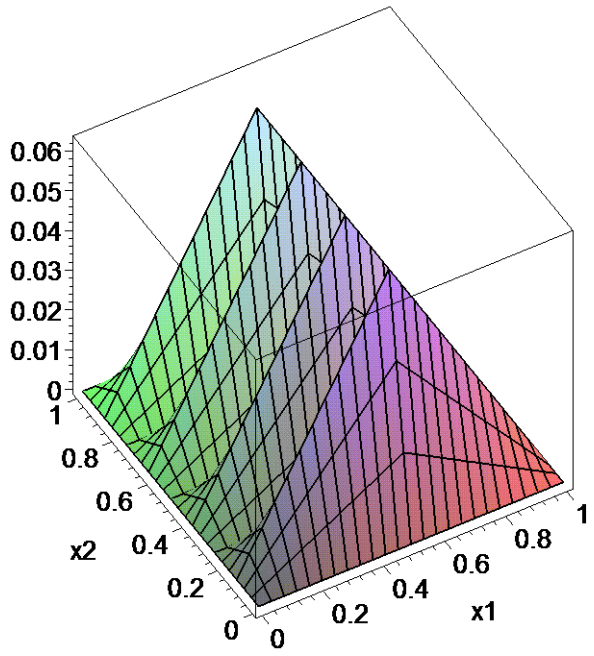
$\frac{1}{12}$



$$\frac{1}{48}$$



$$\frac{1}{48}$$



$$\frac{1}{48}$$