Algorithmen des wissenschaftlichen Rechnens II
Übungsblatt 5
Diskrepanz

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

Warning, the assigned name arrow now has a global binding
```

Ein paar generelle Optionen zum Plotten:

```
> poptions := thickness=3,
              symbol=DIAMOND, symbolsize=20,
              labeldirections=[HORIZONTAL,VERTICAL],
              font=[HELVETICA,20]:
```

Ein Gitter ist hier eine Liste von Paaren [xi,yi] mit ganzzahligen 0<=xi<=N1 und 0<=yi<=N2 für die Gitterpunkte [xi/N1,yi/N2].

Jaja, im Angabenblatt steht "<" statt "<=" und dort ist N1=N2=N, aber diese kleine Verallgemeinerung wird uns ein weiteres Beispiel (die Halton-Folge) mit schön kleiner Diskrepanz erlauben.

```
>
```

So kann man ein Gitter malen:

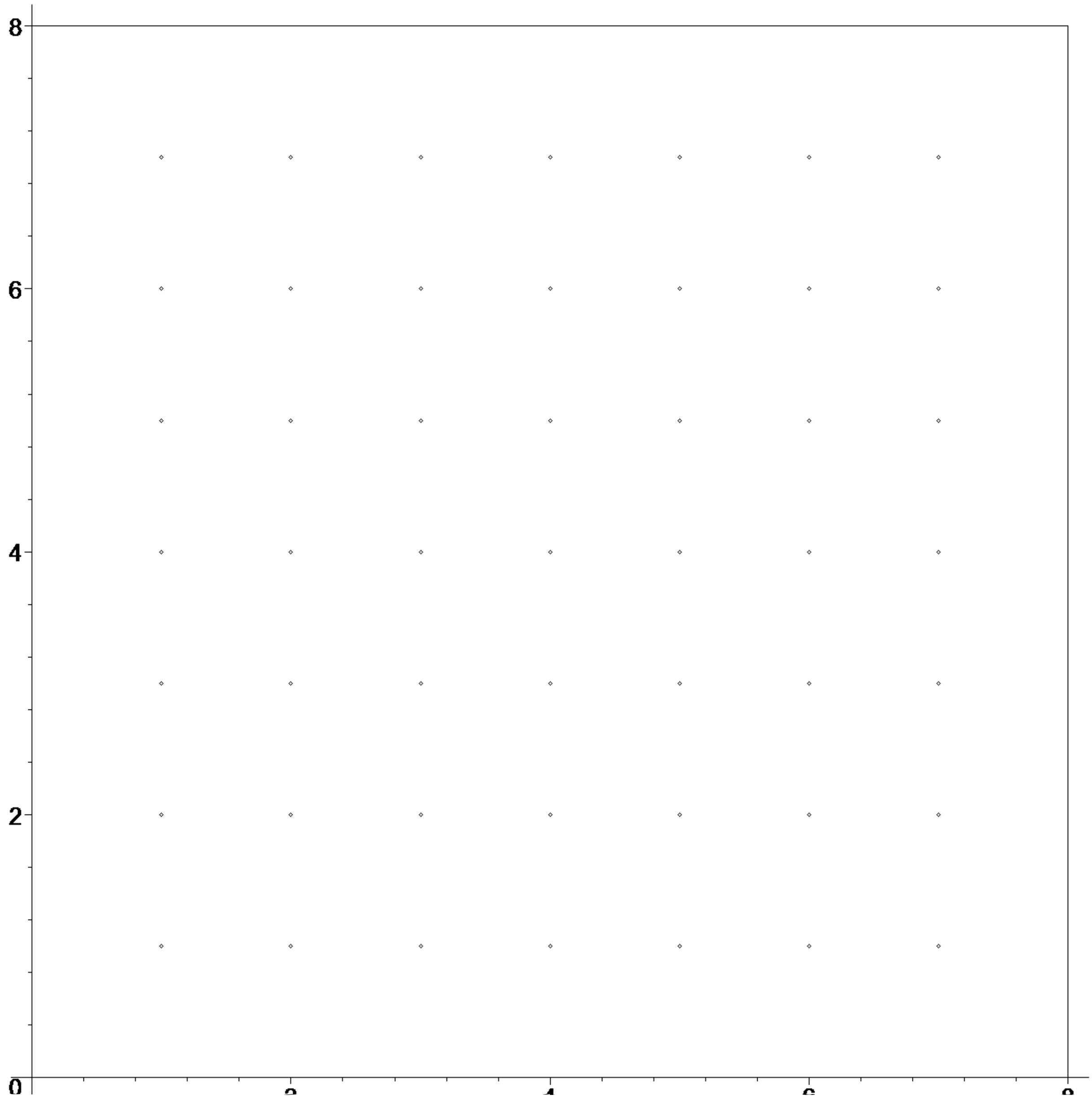
```
> bild := proc(x, N1, N2)
    display(rectangle([0,0],[N1,N2]),
            pointplot(x),
            scaling=CONSTRAINED)
  end proc:
```

Gitter erzeugen

Das ist das vollte Gitter der Tiefe n (Maschenweite h=2^(-n)). Nachher wird es praktisch sein, wenn die Funktion die Rastergrößen N1 und N2 gleich mitliefert, daher tut sie es.

```
> fg := n -> ([seq(seq([i,j],i=1..2^n-1),j=1..2^n-1)],2^n,2^n):
> fg(2);
> bild(fg(3));
```

$$[[1,1],[2,1],[3,1],[1,2],[2,2],[3,2],[1,3],[2,3],[3,3]],4,4$$

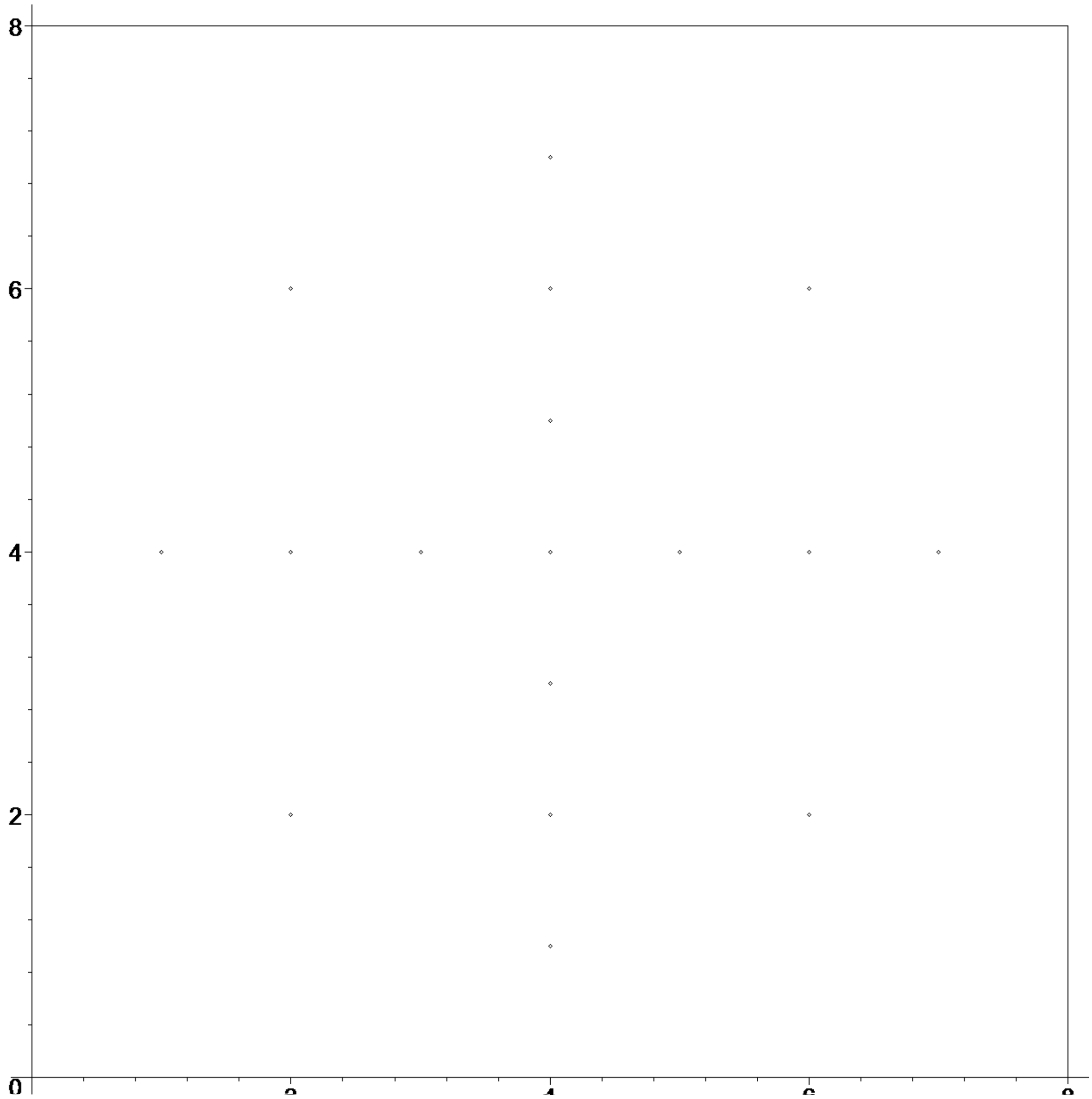


Analog das dünne Gitter der Tiefe n:

```
> dg := n -> ([seq(seq(

  seq(seq([(2*i1-1)*2^(n-l1),(2*i2-1)*2^(n-l2)],
                        i1=1..2^l1/2),i2=1..2^l2/2),
             l1=1..n+1-l2), l2=1..n)], 2^n, 2^n):
> dg(2);
> bild(dg(3));
```

$$[[2, 2], [1, 2], [3, 2], [2, 1], [2, 3]], 4, 4$$

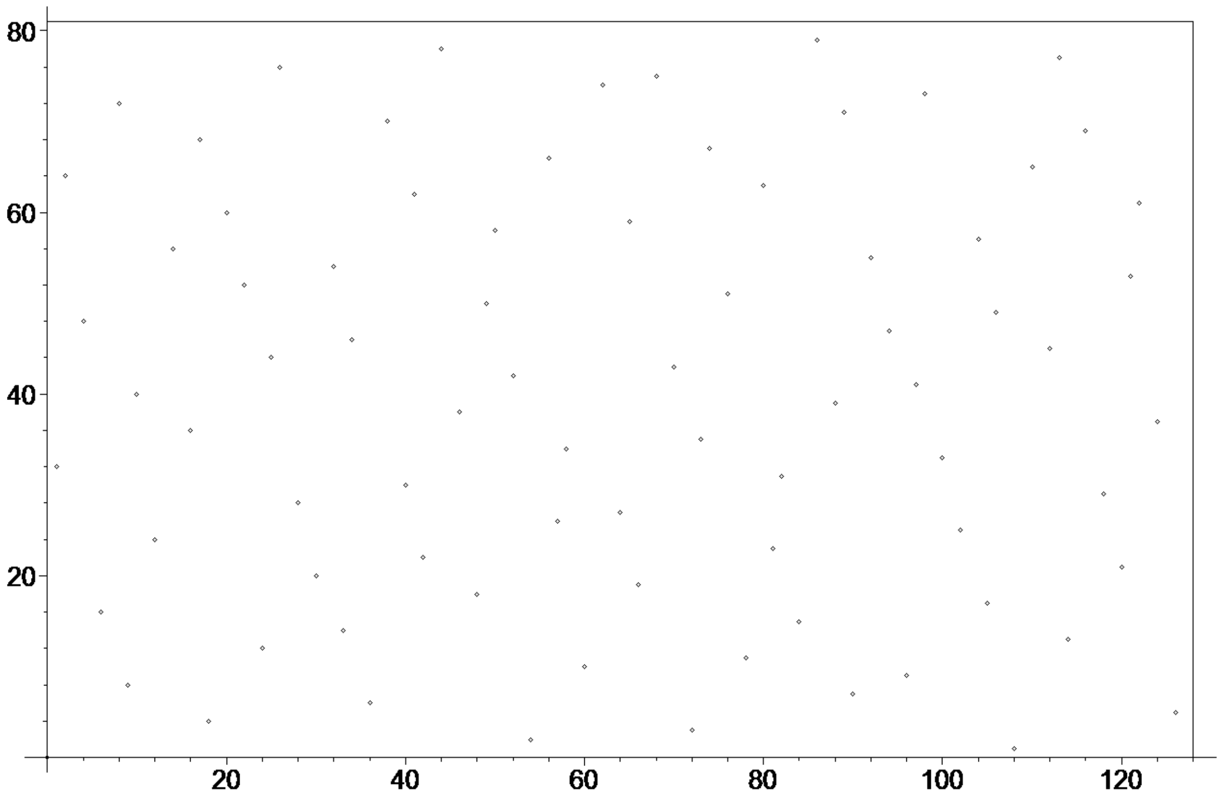


Nun noch die Halton-Folge (bzw. die ersten k Glieder) zur Basis b1=2, b2=3. Was das genau ist, ist egal, hier nur soviel: es ist eine Punktemenge, die für Quadraturverfahren wie wir sie hier betrachten, günstig ist.

```
> phi := proc(n,b)
     local l,i;
     l := convert(n, base, b);
     add(l[i]*b^(-i),i=1..nops(l))
  end proc:
> halton := proc(k)
    local b1,b2,N1,N2;
    b1 := 2;
    b2 := 3;
    N1 := b1^(ceil(log[b1](k)));
    N2 := b2^(ceil(log[b2](k)));
    return [seq([phi(i-1,b1)*N1,phi(i-1,b2)*N2],i=1..k)],
           N1, N2
  end proc:
```

```
> halton(17);
> bild(halton(80));
```

[[0, 0], [16, 9], [8, 18], [24, 3], [4, 12], [20, 21], [12, 6], [28, 15], [2, 24], [18, 1], [10, 10], [26, 19], [6, 4], [22, 13], [14, 22], [30, 7], [1, 16]], 32, 27




```
>
```

Diskrepanz zählen

Nun bestimmen wir die Diskrepanz "mit Gewalt", d.h. wir zählen für jeden Rasterpunkt [i/N1,j/N2], i=0..N1, j=0..N2, wie viele Gitterpunkte "links unterhalb" des Punktes liegen und berechnen die Differenz des Anteils an der Gitterpunkt-Gesamtzahl und der Rechtecksfläche (i/N1)*(j/N2).
Wenn wir einmal mit "<" zählen und einmal mit "<=", können wir sicher sein, die maximal mögliche Differenz zu erwischen (wieso?).
Als Ergebnis geben für die Auswertung wir eine Liste mit drei Komponenten zurück:

- Die Gesamtzahl der Gitterpunkte

- Die Diskrepanz (also die maximal gefundene Differenz)
- eine Liste aus dem (genauer: aus einem) Punkt, in dem die maximale Differenz gefunden wurde, die aus der Quadraturformel geschätzte Größe des zugehörigen Recheckes und dessen tatsächliche Größe (der Betrag der Differenz von beidem ist also die Diskrepanz des Gitters)

```
> diskrepanz := proc(x::list([nonnegint,nonnegint]),
  N1::posint, N2::posint)
    local i, j, anz_lt, anz_leq, xi, xj, d_lt, d_leq, d_max,
  x_max;
    global p_erg;
    d_max := 0;
    for i from 0 to N1 do for j from 0 to N2 do
       xi := [i,j];
       anz_lt := 0;
       anz_leq := 0;
       for xj in x do
          if xj[1]<xi[1] and xj[2]<xi[2] then
             anz_lt := anz_lt +1
          end if;
          if xj[1]<=xi[1] and xj[2]<=xi[2] then
             anz_leq := anz_leq +1
          end if
       end do;
       d_lt := abs(xi[1]*xi[2]/(N1*N2)-anz_lt/nops(x));
       if d_lt>d_max then
          d_max := d_lt;
          x_max := [xi, anz_lt/nops(x), xi[1]*xi[2]/(N1*N2)]
       end if;
       d_leq := abs(xi[1]*xi[2]/(N1*N2)-anz_leq/nops(x));
       if d_leq>d_max then
          d_max := d_leq;
          x_max := [xi, anz_leq/nops(x), xi[1]*xi[2]/(N1*N2)]
       end if;
    end do; end do;
    return [nops(x), d_max, x_max]
  end proc:
>
```

Auswertung

Ein paar volle Gitter:

```
> df := [seq(diskrepanz(fg(n)),n=1..6)];
```

$$df := \left[\left[1, \frac{3}{4}, \left[[1, 1], 1, \frac{1}{4}\right]\right], \left[9, \frac{7}{16}, \left[[3, 3], 1, \frac{9}{16}\right]\right], \left[49, \frac{15}{64}, \left[[7, 7], 1, \frac{49}{64}\right]\right], \left[225, \frac{31}{256}, \left[[15, 15], 1, \frac{225}{256}\right]\right], \left[961, \frac{63}{1024}, \left[[31, 31], 1, \frac{961}{1024}\right]\right],\right.$$

$$\left[3969, \frac{127}{4096}, \left[[63, 63], 1, \frac{3969}{4096}\right]\right]\right]$$

Ein paar dünne Gitter:

```
> dd := [seq(diskrepanz(dg(n)),n=1..7)];
```

$$dd := \left[\left[1, \frac{3}{4}, \left[[1, 1], 1, \frac{1}{4}\right]\right], \left[5, \frac{7}{16}, \left[[3, 3], 1, \frac{9}{16}\right]\right], \left[17, \frac{87}{272}, \left[[6, 6], \frac{15}{17}, \frac{9}{16}\right]\right], \left[49, \frac{183}{784}, \left[[12, 12], \frac{39}{49}, \frac{9}{16}\right]\right], \left[129, \frac{125}{688}, \left[[24, 24], \frac{32}{43}, \frac{9}{16}\right]\right], \left[321, \frac{253}{1712}, \left[[48, 48], \frac{76}{107}, \frac{9}{16}\right]\right], \left[769, \frac{1527}{12304}, \left[[96, 96], \frac{528}{769}, \frac{9}{16}\right]\right]\right]$$

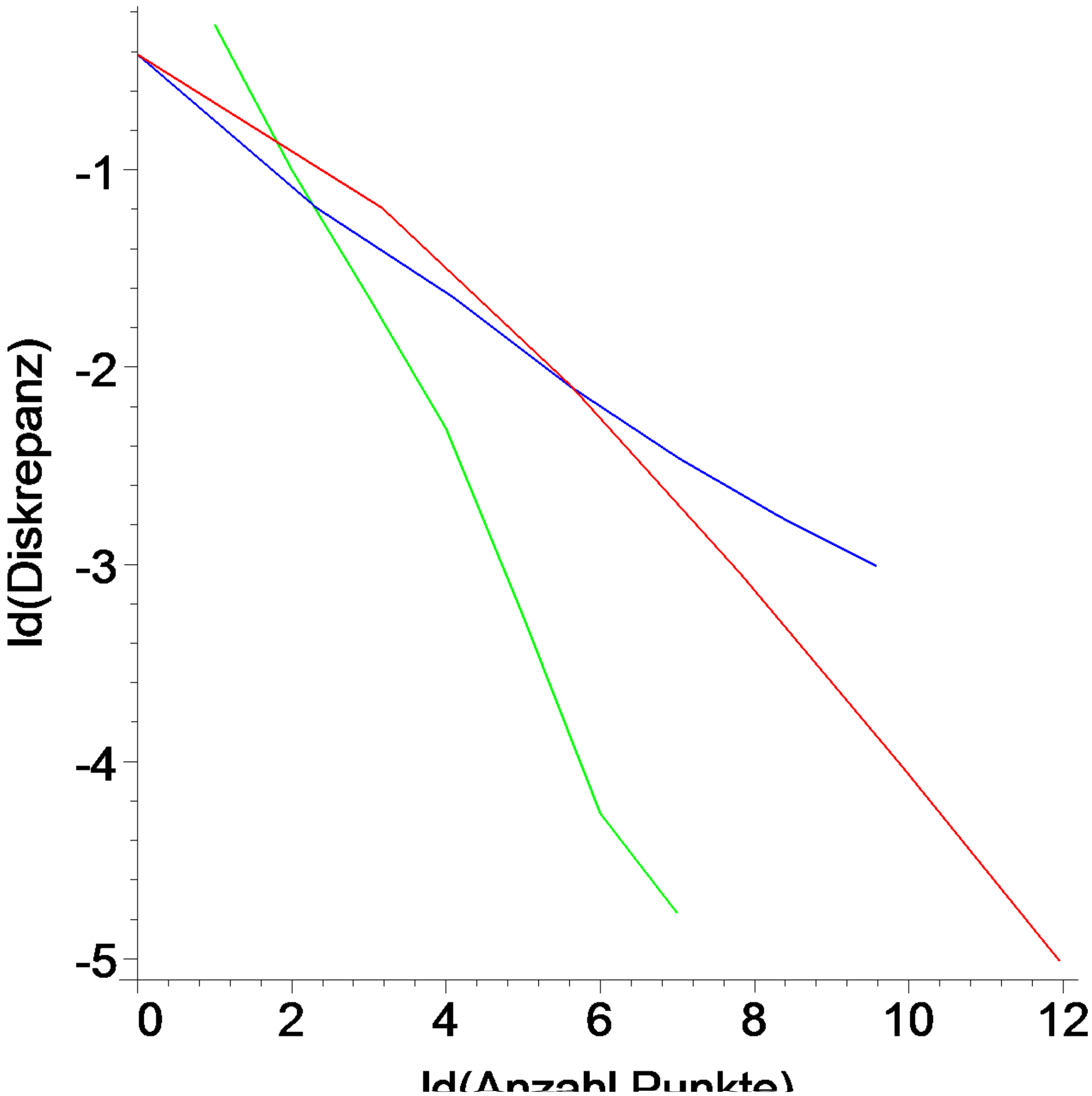
Und ein paar Gitter der Halton-Folge:

```
> dh := [seq(diskrepanz(halton(2^n)),n=1..7)];
```

$$dh := \left[\left[2, \frac{5}{6}, \left[[1, 1], 1, \frac{1}{6}\right]\right], \left[4, \frac{1}{2}, \left[[3, 3], \frac{3}{4}, \frac{1}{4}\right]\right], \left[8, \frac{23}{72}, \left[[7, 7], 1, \frac{49}{72}\right]\right], \left[16, \frac{29}{144}, \left[[12, 13], \frac{9}{16}, \frac{13}{36}\right]\right], \left[32, \frac{5}{48}, \left[[18, 30], \frac{5}{16}, \frac{5}{24}\right]\right], \left[64, \frac{5}{96}, \left[[54, 58], \frac{21}{32}, \frac{29}{48}\right]\right], \left[128, \frac{95}{2592}, \left[[93, 121], \frac{51}{128}, \frac{3751}{10368}\right]\right]\right]$$

Zum Vergleich zeichnen wir sie alle in ein Schaubild (Volles Gitter: Rot, dünnes Gitter: Blau, Halton-Folge Grün), immer der Zweierlogarithmus der Diskrepanz aufgetragen gegen den Zweierlogarithmus der Anzahl Gitterpunkte.

```
> pf := pointplot([seq([log[2](i[1]),log[2](i[2])],i=df)],
                  color=RED,connect=true,poptions):
  pd := pointplot([seq([log[2](i[1]),log[2](i[2])],i=dd)],
                  color=BLUE,connect=true,poptions):
> ph := pointplot([seq([log[2](i[1]),log[2](i[2])],i=dh)],
                  color=GREEN,connect=true,poptions):
  display([pf,pd,ph],
          labels=["ld(Anzahl
  Punkte)","ld(Diskrepanz)"],poptions);
```



Wir sehen: das dünne Gitter hat eine sehr große (Stern-)Diskrepanz, ist für Quadraturformeln dieser Bauart also ungeeignet.
Brauchbare Dünngitter-Quadraturformeln bekommt man erst, indem man den Auswertestellen passende Gewichte gibt (Archimedes und die Kombinationstechnik machen das z.B.). Dann sind anders definierte Diskrepanzen relevant, bei denen das dünne Gitter gut abschneidet.

>