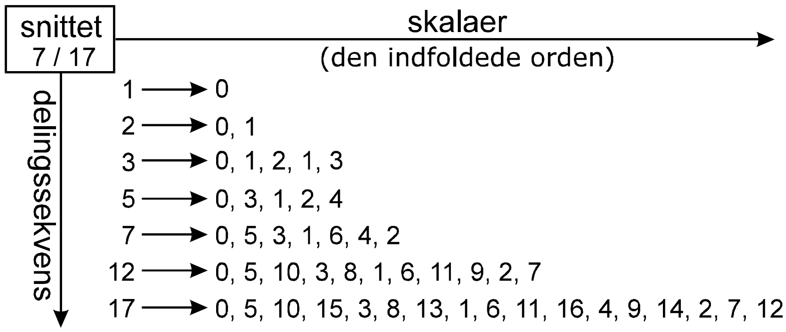
Dette er den tredje af fem artikler under den fælles overskrift
Studier på grundlag af programmet SKALAGENERATOREN
(forfatter: Jørgen Erichsen)

3. Om skalamønstrene og "den indfoldede orden"

Lad os begynde dette kapitel med et kort resumé. Vi har set, hvordan det gyldne snit giver anledning til dannelsen af et system af strukturer, der i denne forbindelse kaldes skalaer og hvis udvikling følger Fibonacci-rækken. Vi har videre set, at at *ethvert* snit giver anledning til dannelsen af et sådant system af skalaer, og at udviklingen i hvert enkelt tilfælde følger en unik talrække, der i denne forbindelse kaldes delingssekvensen.

Sammenhængen mellem snit, delingssekvens og skalaer er grafisk anskueliggjort i fig.3.1, idet jeg endnu engang har valgt snittet 7/17 som eksempel:



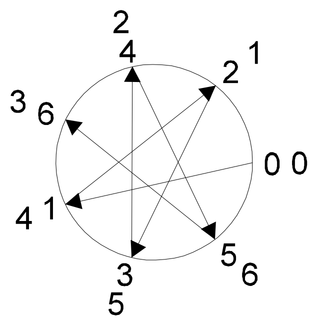
Figur 3.1

Skalaerne er her repræsenteret ved deres *indfoldede orden* – et begreb der blev forklaret i kap.1. Jeg vil nu gennemgå en algoritme, der sætter os i stand til at finde den indfoldede orden for en given skala, når blot vi kender snittet samt antallet af elementer i skalaen. I algoritmen indgår division med rest også kaldet *modulodivision*. I nogle situationer og ikke mindst i forbindelse med programmering er det ikke kvotienten, men resten vi er interesseret i, og vi foretager da en modulodivision. Handler det f.eks. om at finde resten, når vi dividerer 12 med 7, skrives det således:

$$12 \bmod 7 = 5$$

(udtales "12 modulo 7 er lig med 5"). Samme rest får vi, når vi dividerer hvert af tallene 5, 19, 26, 33, 40 osv. med 7, og man siger at disse tal er kongruente modulo 7.

Betragt nu den følgende tegning, der fremstiller et delingsforløb baseret på snittet 4/7. Den indre kreds af tal angiver den rækkefølge, hvori delingen finder sted, dvs. den indfoldede orden; den ydre kreds af tal er en nummerering af delingerne regnet fra begyndelsen og i modsat retning af urviserens bevægelse:



Figur 3.4

Det vi nu skal lægge mærke til er, at der er en sammenhæng mellem de to rækker af tal. Ganger vi nemlig tallene i den indre kreds med snittet 4/7, vil resten i hvert tilfælde være identisk med det tilsvarende tal i den ydre kreds – eller som det udtrykkes i modulo-terminologi:

$$
\begin{aligned}
0 \cdot 4 \bmod 7 &= 0 \\
2 \cdot 4 \bmod 7 &= 1 \\
4 \cdot 4 \bmod 7 &= 2 \\
6 \cdot 4 \bmod 7 &= 3 \\
1 \cdot 4 \bmod 7 &= 4 \\
3 \cdot 4 \bmod 7 &= 5 \\
5 \cdot 4 \bmod 7 &= 6
\end{aligned}
$$

Idet vi symbolisererer et tal i den indfoldede orden (den indre kreds af tal) ved bogstavet q og et tal i den numeriske rækkefølge (den ydre kreds af tal) ved bogstavet n, går opgaven imidlertid ud på at besvare spørgsmålet: ”Hvilken værdi har q på den n’te plads?” Opgaven kan kun løses ved, at vi afprøver alle potentielle muligheder og stopper op, når vi har fundet den rigtige. I dette tilfælde skal vi for hver af de syv pladser gennemfører alle syv modulodivisioner, frem til det punkt, hvor resten er identisk med nummeret på den pågældende plads. Det giver en masse regnearbejde, men det er ganske uden betydning, når vi tager computeren til hjælp. Formuleret i et simpelt programmeringssprog er fremgangsmåden (algoritmen) denne:

```
FOR n = 0 TO 6
  FOR q = 0 TO 6
    IF q * 4 MOD 7 = n THEN EXIT FOR
  NEXT q
  PRINT q
NEXT n
```

Vi har her to løkker, hvor den ene er indesluttet i den anden. Begge løkker er af typen For-Next, dvs. de gennemløber et antal i forvejen fastlagte værdier, her fra 0 til 6. Lad os som eksempel sige, at vi i den yderste løkke er nået frem til 5. plads. Nu afprøves i den inderste løkke hver af de syv værdier som q kan have. Afprøvningen består i at udføre den før omtalt modulodivision. Når den som resultat giver n (udtrykt i kodeordene IF . . . THEN), har vi fundet den indfoldede orden på den pågældende plads; vi kan nu forlade den inderste løkke (udtrykt i kodeordene EXIT FOR), hvorefter resultet skrives ud (kodeordet PRINT q). Dermed er vi tilbage i den ydre løkke, og den samme procedure gentages med den næste værdi af n. Som slutresultat vil vi på skærmen have stående skalaens indfoldede orden:

```
0  2  4  6  1  3  5
```

Ved hjælp af en tredje løkke, som omslutter de to andre, kan vi få programmet til at udskrive den den indfoldede orden for *alle* snit, hvor nævneren er 7 (altså 1/7, 2/7 . . . 6/7), og ved at indskyde en passende kode på det sted, hvor der skiftes til en ny tæller, kan vi få det hele sat op i form af en matrice, dvs. en opstilling i rækker og kolonner. Tælleren kalder vi som sædvanlig b; linieskiftet er her blot antydet ved sætningen ”skift til ny linie”:

```
FOR b = 1 TO 6
   FOR n = 0 TO 6
      FOR q = 0 TO 6
        IF q * b MOD 7 = n THEN EXIT FOR
      NEXT q
      PRINT q
   NEXT n
   (skift til ny linie)
NEXT b
```

På skærmen står der nu:

0	1	2	3	4	5	6
0	4	1	5	2	6	3
0	5	3	1	6	4	2
0	2	4	6	1	3	5
0	3	6	2	5	1	4
0	6	5	4	3	2	1

I fig.2.7 blev samtlige delingsforløb, hvor snittet er rationelt, og hvor nævneren er mindre end 13 anskueliggjort grafisk. Den følgende opstilling viser den indfoldede orden for hver af disse delingsforløb:

Den indfoldede orden
for alle snit fra 1/2 til 11/12
(sml. med fig.2.7)

2-deling

0	1

3-deling

0	1	2
0	2	1

4-deling

0	1	2	3
0	·	1	·
0	3	2	1

5-deling

0	1	2	3	4
0	3	1	4	2
0	2	4	1	3
0	4	3	2	1

6-deling

0	1	2	3	4	5
0	·	1	·	2	·
0	·	·	1	·	·
0	·	2	·	1	·
0	5	4	3	2	1

7-deling

0	1	2	3	4	5	6
0	4	1	5	2	6	3
0	5	3	1	6	4	2
0	2	4	6	1	3	5
0	3	6	2	5	1	4
0	6	5	4	3	2	1

8-deling

0	1	2	3	4	5	6	7
0	·	1	·	2	·	3	·
0	3	6	1	4	7	2	5
0	·	·	·	1	·	·	·
0	5	2	7	4	1	6	3
0	·	3	·	2	·	1	·
0	7	6	5	4	3	2	1

9-deling

0	1	2	3	4	5	6	7	8
0	5	1	6	2	7	3	8	4
0	·	·	1	·	·	2	·	·
0	7	5	3	1	8	6	4	2
0	2	4	6	8	1	3	5	7
0	·	·	2	·	·	1	·	·
0	4	8	3	7	2	6	1	5
0	8	7	6	5	4	3	2	1

10-deling

0	1	2	3	4	5	6	7	8	9
0	·	1	·	2	·	3	·	4	·
0	7	4	1	8	5	2	9	6	3
0	·	3	·	1	·	4	·	2	·
0	·	·	·	·	1	·	·	·	·
0	·	2	·	4	·	1	·	3	·
0	3	6	9	2	5	8	1	4	7
0	·	4	·	3	·	2	·	1	·
0	9	8	7	6	5	4	3	2	1

11-deling

0	1	2	3	4	5	6	7	8	9	10
0	6	1	7	2	8	3	9	4	10	5
0	4	8	1	5	9	2	6	10	3	7
0	3	6	9	1	4	7	10	2	5	8
0	9	7	5	3	1	10	8	6	4	2
0	2	4	6	8	10	1	3	5	7	9
0	8	5	2	10	7	4	1	9	6	3
0	7	3	10	6	2	9	5	1	8	4
0	5	10	4	9	3	8	2	7	1	6
0	10	9	8	7	6	5	4	3	2	1

12-deling

0	1	2	3	4	5	6	7	8	9	10	11
0	·	1	·	2	·	3	·	4	·	5	·
0	·	·	1	·	·	2	·	·	3	·	·
0	·	·	·	1	·	·	·	2	·	·	·
0	5	10	3	8	1	6	11	4	9	2	7
0	·	·	·	·	·	1	·	·	·	·	·
0	7	2	9	4	11	6	1	8	3	10	5
0	·	·	·	2	·	·	·	1	·	·	·
0	·	·	3	·	·	2	·	·	1	·	·
0	·	5	·	4	·	3	·	2	·	1	·
0	11	10	9	8	7	6	5	4	3	2	1

Sammenligner vi med fig.2.4, er det tydeligt hvordan polygonernes individuelle struktur og indbyrdes symmetri afspejler sig i matricerne. De delinger der bortfalder, fordi tæller og nævner kan forkortes, er markeret ved en prik.

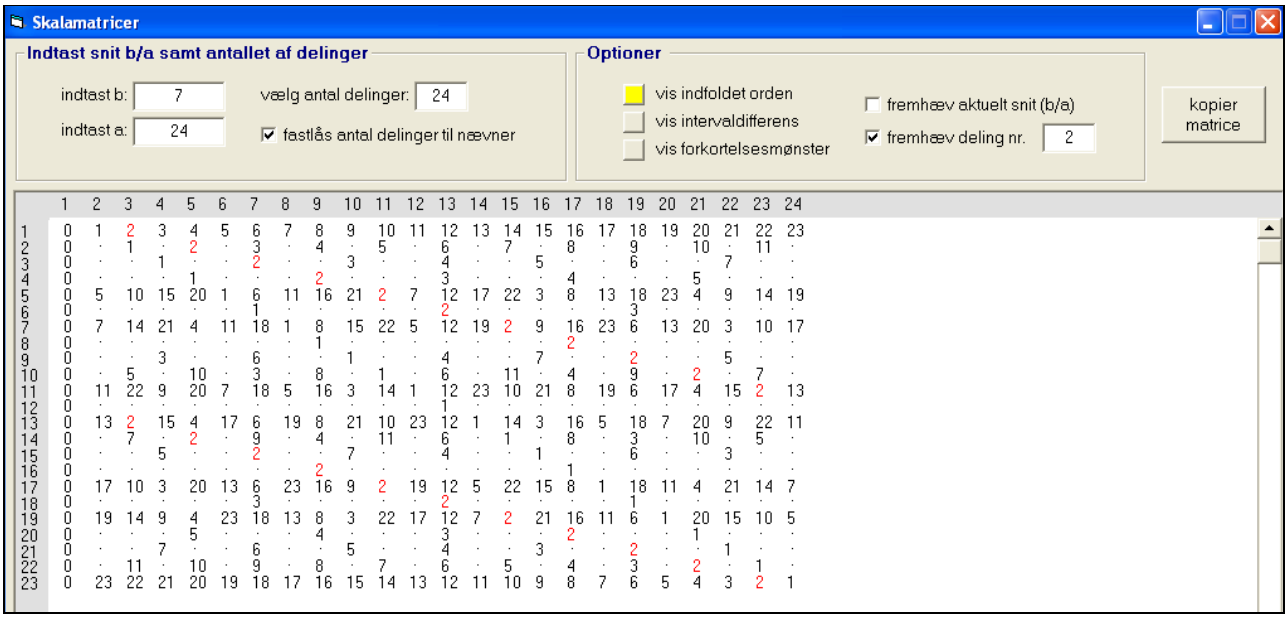
Hvad angår den indbyrdes forbindelse mellem rækkerne i den enkelte matrice, så springer det umiddelbart i øjnene, at 1-tallerne danner en diagonal linie ned gennem matricen. Men ser vi nøjere efter, viser det sig, at noget lignende er tilfældet med de øvrige tal; blot er linien her ikke diagonal, men hældningen aftager i takt med at tallet vokser. I den næste opstilling gentages matricen for 11-deling, idet 2-tallerne er fremhævet i opstillingen til venstre, og 3-tallerne er fremhævet i opstillingen til højre:

0	1	**2**	3	4	5	6	7	8	9	10
0	6	1	7	**2**	8	3	9	4	10	5
0	4	8	1	5	9	**2**	6	10	3	7
0	3	6	9	1	4	7	10	**2**	5	8
0	9	7	5	3	1	10	8	6	4	**2**
0	**2**	4	6	8	10	1	3	5	7	9
0	8	5	**2**	10	7	4	1	9	6	3
0	7	3	10	6	**2**	9	5	1	8	4
0	5	10	4	9	3	8	**2**	7	1	6
0	10	9	8	7	6	5	4	3	**2**	1

0	1	2	**3**	4	5	6	7	8	9	10
0	6	1	7	2	8	**3**	9	4	10	5
0	4	8	1	5	9	2	6	10	**3**	7
0	**3**	6	9	1	4	7	10	2	5	8
0	9	7	5	**3**	1	10	8	6	4	2
0	2	4	6	8	10	1	**3**	5	7	9
0	8	5	2	10	7	4	1	9	6	**3**
0	7	**3**	10	6	2	9	5	1	8	4
0	5	10	4	9	**3**	8	2	7	1	6
0	10	9	8	7	6	5	4	**3**	2	1

Hvis man forestiller sig at matricen vikles rundt om en cylinder, så vil 1-tallerne danne en skruelinje, som når én gang rundt om cylinderen, 2-tallerne vil danne en skruelinje, som når to gange rundt om cylinderen, 3-tallerne vil danne en skruelinje, som når tre gange rundt om cylinderen – og så fremdeles. Hver enkelt række gentages her i et cirkulært periodisk forløb; men man kan også forestille sig, at det samme er tilfældet med kolonnerne – nemlig ved at cylinderen forvandles til en torus (en ring).

Jeg har skrevet et specielt program, SKALAMATRICER, som grundlæggende er baseret på den algoritme, der er gennemgået ovenfor. Den næste illustration, fig.4.2, viser hvordan brugerfladen ser ud.



Figur 3.5

Øverst til venstre ser vi de samme indtastningsbokse for snittet og for antallet af delinger, som vi kender fra SKALAGENERATOREN. Desuden kan antallet af delinger fastlåses til nævneren. Af praktiske grunde er nævnerens maksimale størrelse sat til 99. Allerede før vi når hertil begynder genereringen af matricen at tage mærkbar tid (det skyldes dog mere opsætningen på skærmen, end det skyldes det egentlige regnearbejde), og desuden vokser matricens udstrækning i både bredde og højde, så det kun er et udsnit ad gangen, der kan ses på skærmen (bemærk rullepanelerne til højre og forneden). Til højre er der tre valg: Vi kan få vist (1) den indfoldede orden, (2) de tilsvarende intervaldifferenser, (3) forkortelsesmønstret. Hvad intervaldifferenser angår, er det let at indse, at der blot skal indskydes en simpel subtraktionsprocedure i programkoden, for at vi kan få disse beregnet og udskrevet. Hvad forkortelsesmønstret betyder, bliver forklaret lidt senere. Uanset hvilken tæller vi har indtastet, vises resultatet for *alle* tællere mellem 1 og a–1, men vi kan vælge at få fremhævet det indtastede snit. Vi kan også vælge at få fremhævet et bestemt tal i den indfoldede orden – i eksemplet er det tallet 2 (fremhævet med rødt)

Vælger vi at få vist intervaldifferenserne, får vi dette resultat at se (her er tillige valgt at få fremhævet det indtastede snit, 7/24):

		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-23
2		1	·	1	·	1	·	1	·	1	·	1	·	1	·	1	·	1	·	1	·	1	·	-11
3		·	1	·	·	1	·	·	1	·	·	1	·	·	1	·	·	1	·	·	1	·	·	-7
4		·	·	1	·	·	·	1	·	·	·	1	·	·	·	1	·	·	·	1	·	·	·	-5
5		5	5	5	-19	5	5	5	5	-19	5	5	5	5	-19	5	5	5	5	-19	5	5	5	-19
6		·	·	·	·	1	·	·	·	·	·	1	·	·	·	·	·	1	·	·	·	·	·	-3
7		7	7	-17	7	7	-17	7	7	7	-17	7	7	-17	7	7	7	-17	7	7	-17	7	7	-17
8		·	·	·	·	·	·	1	·	·	·	·	·	·	·	1	·	·	·	·	·	·	·	-2
9		·	3	·	·	3	·	·	-5	·	·	3	·	·	3	·	·	-5	·	·	3	·	·	-5
10		5	·	5	·	-7	·	5	·	-7	·	5	·	5	·	-7	·	5	·	-7	·	5	·	-7
11	1	11	-13	11	-13	11	-13	11	-13	11	-13	11	11	-13	11	-13	11	-13	11	-13	11	-13	11	-13
12		·	·	·	·	·	·	·	·	·	·	1	·	·	·	·	·	·	·	·	·	·	·	-1
13	3	-11	13	-11	13	-11	13	-11	13	-11	13	-11	-11	13	-11	13	-11	13	-11	13	-11	13	-11	-11
14		7	·	-5	·	7	·	-5	·	7	·	-5	·	-5	·	7	·	-5	·	7	·	-5	·	-5
15		·	5	·	·	-3	·	·	5	·	·	-3	·	·	-3	·	·	5	·	·	-3	·	·	-3
16		·	·	·	·	·	·	2	·	·	·	·	·	·	·	-1	·	·	·	·	·	·	·	-1
17	7	-7	-7	17	-7	-7	17	-7	-7	-7	17	-7	-7	17	-7	-7	-7	17	-7	-7	17	-7	-7	-7
18		·	·	·	·	3	·	·	·	·	·	-1	·	·	·	·	·	-1	·	·	·	·	·	-1
19	9	-5	-5	-5	19	-5	-5	-5	-5	19	-5	-5	-5	-5	19	-5	-5	-5	-5	19	-5	-5	-5	-5
20		·	·	5	·	·	·	-1	·	·	·	-1	·	·	·	-1	·	·	·	-1	·	·	·	-1
21		·	7	·	·	-1	·	·	-1	·	·	-1	·	·	-1	·	·	-1	·	·	-1	·	·	-1
22		11	·	-1	·	-1	·	-1	·	-1	·	-1	·	-1	·	-1	·	-1	·	-1	·	-1	·	-1
23	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

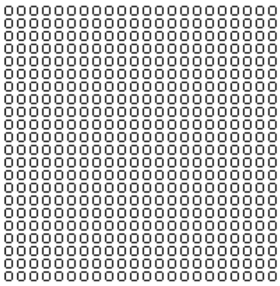
Figur 3.6

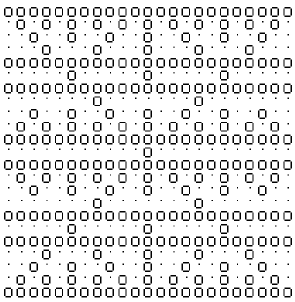
Vælger vi som den tredje mulighed at få vist forkortelsesmønstret, bliver alle tal erstattet af nuller. Dermed fremkommer et geometrisk mønster, der endnu tydeligere fremhæver de steder, hvor tæller og nævner forkortes. Forkortelsesmønstret, der egentlig mere skal forstås som et matematisk kuriosum, ser i det aktuelle tilfælde sådan ud:

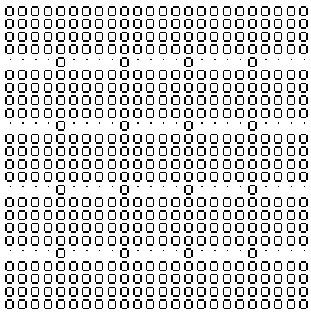
```
00000000000000000000000
.0.0.0.0.0.0.0.0.0.0.0.
..0..0..0..0..0..0..0..
...0...0...0...0...0...
00000000000000000000000
.....0.....0.....0.....
00000000000000000000000
.......0.......0.......
..0..0..0..0..0..0..0..
.0.0.0.0.0.0.0.0.0.0.0.
00000000000000000000000
...........0...........
00000000000000000000000
.0.0.0.0.0.0.0.0.0.0.0.
..0..0..0..0..0..0..0..
.......0.......0.......
00000000000000000000000
.....0.....0.....0.....
00000000000000000000000
...0...0...0...0...0...
..0..0..0..0..0..0..0..
.0.0.0.0.0.0.0.0.0.0.0.
00000000000000000000000
```

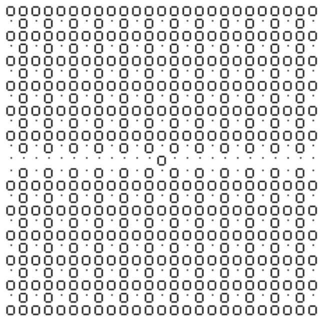
Figur 3.7

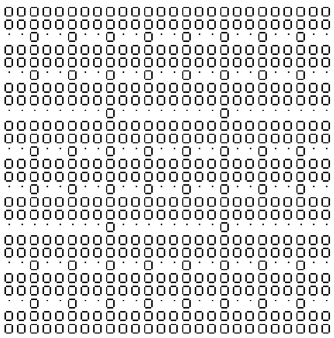
Selv om forkortelsesmønstret vel mere må betragtes som et kuriosum, er det ganske fascinerende at følge, hvordan det ændrer sig fra den ene nævner til den anden. I de næste figurer kan vi således følge, hvordan forkortelsesmønsteret udvikler sig mellem nævnerne 23 og 29. I ydertilfældene afslører mønstrets maksimale tæthed, at der er tale om primtal, hvorimod de mere eller mindre gennemhullede mønstre i de mellemliggende tilfælde særdeles tydeligt afslører, hvor tæller og nævner kan forkortes. 24 har som allerede nævnt særlig mange forkortelsesmuligheder, hvorimod 25 kun har en enkelt – men jeg vil i øvrigt overlade den aritmetiske tolkning af mønstrene til læseren.

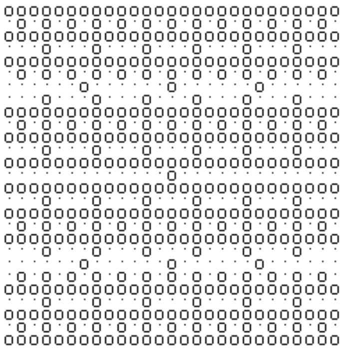

23

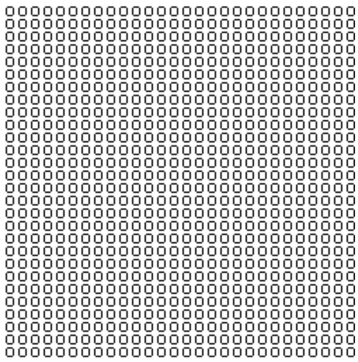

24


25


26


27


28


29

Man vil også kunne tolke de markerede tal som nodeværdier og de tomme pladser som pauser. Idet det vandrette forløb på sædvanlig vis identificeres med tidsforløbet, kan hele matricen tolkes som et partitur, og måske kan musikeren allerede for sit indre øre høre et rytmisk samspil mellem fra 23 til 29 instrumenter – et samspil der strækker sig fra fuldkommen monotoni til de mest komplicerede polyrytmer!

I øvrigt minder mønstrene også om de hulkort, der brugtes til Jacquard-vævning (en teknik til vævning af komplicerede mønstre), før denne blev afløst af computerstyret vævning.