Markovkedjor, musikteori och automatgenererad musik

Edvin Tengroth, Herman Efraimsson, Ludvig Wadenbäck & Lukas Petersson
Handledare: Tomas Persson

februari 2022

Sammanfattning

Att beskriva ljud och musik med matematik har under lång tid varit intressant och idag har vi en bra förståelse för vad som gör att vissa ljud och melodier låter bra och andra sämre. Vi kommer i denna rapport utforska delar av musikteori och matematik för att generera musik med målet att musiken ska låta bra. Själva genereringen sker med hjälp av Markovkedjor.

1 Introduktion

Varför låter vissa ljud och melodier bra och kan denna egenskap beskrivas matematiskt? En person som tidigt intresserade sig för denna fråga var Pythagoras som enligt legenden hörde harmoni i hammarslagen när han en dag gick förbi en smedja. Under sin livstid undersökte han denna harmoni mellan toner och utvecklade bland annat en metod för att stämma ett instrument. Lite närmare i tiden kom även Leonhard Euler att få ett intresse för harmoni mellan toner som han beskrev utförligt i brev till en tysk prinsessa [4]. Med tiden blev vår förståelse för musik och harmoni bättre och en ny fråga kunde ställas. Kan vi med hjälp av denna förståelse om ljud generera musik?

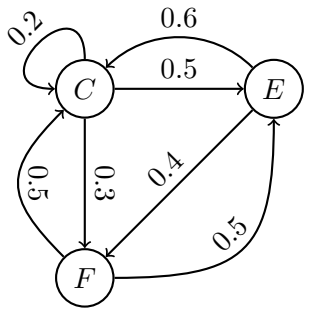
I den här rapporten ska vi utforska den frågan. Musiken kommer geneneras med hjälp av Markovkedjor som låter oss vikta hur vi väljer toner och melodier. Hur vi viktar valen kommer bestämmas genom att successivt introducera idéer från musikteori och med matematik försöka använda denna teori för att vikta valen bättre. Rapporten kommer därför vara uppdelad i iterationer. Varje iteration, efter första musikresultatet, kommer börja med ny musikteori, sedan kommer ett stycke med ny matematik och i slutet av varje iteration använder vi oss av det vi lärt oss för att generera en ny, förhoppningsvis bättre låtande låt.

2 Teoretisk grund Markovkedjor

En stokastisk variabel X betecknar resultatet av ett slumpförsök med reellvärda utfall, där de olika utfallen har bestämda sannolikheter. Antag nu en mängd av stokastiska variabler $\{X(t); t \in T\}$ vilket kallas en stokastisk process. Om vi låter indexmängden T vara diskret tid kan vi skriva denna sekvens som $\{X_0, X_1, X_2, \dots\}$ där X_t är tillståndet vid tiden t. Antag nu en sådan stokastisk process med egenskapen att tillståndet X_{t+1} endast beror på X_t och därmed ej på de föregående tillstånden $X_0, X_1, \dots, X_{t-1}$. En sådan process kallas en Markovkedja. En Markovkedja är därmed en stokastisk process där nästa steg i processen endast är beroende av det nuvarande tillståndet. Den typen av Markovkedja vi kommer använda är även tidshomogen, det vill säga att sannolikheterna för de olika utfallen inte förändras med tiden.

För att beskriva en tidshomogen Markovkedja behöver vi veta två saker, de olika värden som vårt tillstånd X_i kan anta och sannolikheterna för vårt nästa steg. Denna information kan representeras med hjälp av en riktad graf. En riktad graf är ett par (V, E), där V är en ändlig mängd och $E \subseteq V \times V$ [5]. Vi kallar vår mängd V för noder och mängden E för kanter. För att representera Markovkedjor låter vi de värden som ett tillstånd kan anta vara våra noder och sannolikheterna mellan dessa vara våra riktade kanter. Varje steg i Markovkedjan blir nu en förflyttning från en nod till en annan, där sannolikheten att en förflyttning sker till en specifik nod bestäms av värdet på den korresponderande riktade kanten. Notera att start- och slutnoden i en förflyttning kan vara samma nod.

En sådan riktad graf kan även representeras med hjälp av en anslutningsmatris. Detta är en kvadratisk matris där elementet a_{ij} i anslutningsmatrisen beskriver sannolikheten att vi förflyttar oss från noden v_i till v_j. I figur 1 representeras en enkel Markovkedja som en graf och även grafens anslutningsmatris. Markovkedjans tillstånd är de tre noterna C, E, F och pilarna beskriver sannolikheten att vi förflyttar oss mellan de olika tillstånden. Vi kan även se att om vi summerar alla de pilar som löper från en nod så blir resultatet 1.

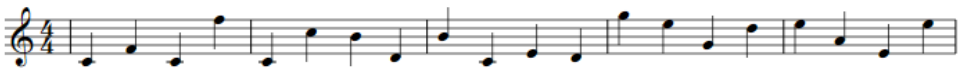


$$\begin{array}{c} \\ C \\ E \\ F \end{array} \begin{array}{c} \begin{array}{ccc} C & E & F \end{array} \\ \begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.6 & 0 & 0.4 \\ 0.5 & 0.5 & 0 \end{pmatrix} \end{array}$$

Figur 1: Exempel på en Markovkedja representerad som en graf samt grafens anslutningsmatris.

3 Första musikresultatet

Låt oss nu försöka generera en låt med hjälp av en Markovkedja. För att göra detta behöver vi bestämma vilka noter vi använder, vilken not vi börjar på och sannolikheterna till vår anslutningsmatris. Vi låter våra noder vara C-dur skalan och då vi ännu inte har en idé om vad som låter bra tillsammans så låter vi sannolikheterna vara homogena, därmed rör vi oss helt slumpmässigt mellan noter. Efter att ha valt en godtycklig startnot kan vi nu börja generera en simpel låt med vår Markovkedja. I varje steg av genereringen väljer vi vår nästa not beroende på den nuvarande notens rad i anslutningsmatrisen. I praktiken sker detta genom att först välja ett tal p_1, där $0 \leq p_1 \leq 1$, så att sannolikheten att p_1 ligger i intervallet $[0, x]$ är x. Denna egenskap kommer i resten av rapporten kallas slumpmässig. Låt sedan v_i vara vår nuvarande not som representeras av raden a_i i anslutningsmatrisen. Jämför nu radens första värde a_{i1} med p_1, om $p_1 \leq a_{i1}$ så väljer vi noten v_1 som vår nästa not och om $p_1 > a_{i1}$ väljer vi ett nytt värde $p_2 = p_1 - a_{i1}$ och repeterar föregående beräkning med a_{i2}. Denna algoritm fortsätter tills vi hittar ett värde a_{ij} som uppfyller $p_i \leq a_{ij}$ och väljer då den motsvarande noten v_j som nästa not. Vi upprepar denna process vid varje nytt steg och får därifrån låten i figur 2.



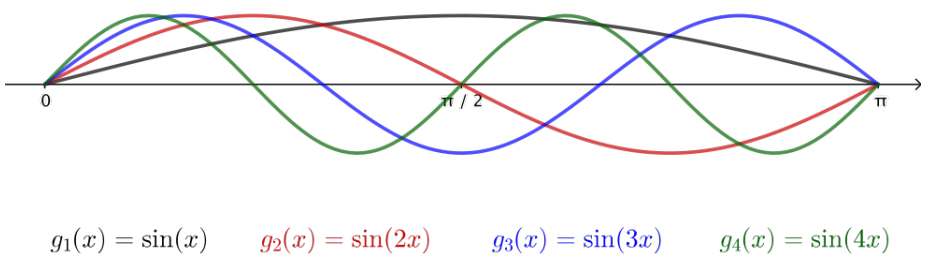
Figur 2: Enkelt musikstycke. Tryck här för att lyssna.

4 Inledande musikteori

Tonarten man spelar i är det som bestämmer vilka noter man kan spela. Den kan också kallas för en skala. En av de vanligaste är C-dur, vilket också är den som vi använder för att skapa den autogenererade musiken. Det betyder att vi kan spela noterna C, D, E, F, G, A och B. Senare kommer vi även generera vissa svarta noter till de ackordens skalor, vilket tas upp mer detaljerat senare.

I våra genererade låtar använder vi piano-ljud. Pianon stäms så att noten A över det mittersta C:et har frekvensen 440 Hz. Detta A kallas A4 och C:et kallas oftast för C4. Efter att A4 har stämts, kan resten av pianot stämmas. Detta görs så att det är samma proportion mellan frekvensen på alla två på varandra följande halvnoter. Det finns 12 halvnoter och därför är denna proportion $2^{1/12}$ så att A5 blir 880 Hz och A3 blir 220 Hz. Majoriteten av noterna får därför en irrationell frekvens som måste approximeras när man stämmer ett piano.

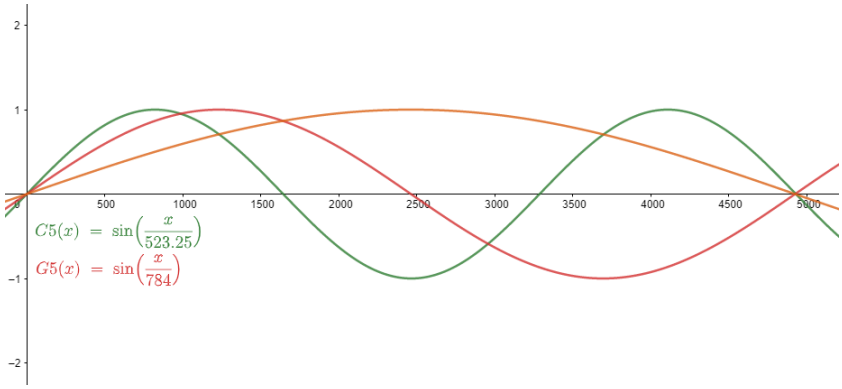
Eftersom ett piano är ett stränginstrument, kommer tonerna från de stående vågorna som strängen skapar. Grundtonen kommer från vågen med noder i strängens ändar. Överton nummer n kommer från vågen med n noder emellan strängens ändar. Övertonernas frekvens ges av $f = n \cdot f_0$, där f_0 är grundtonens frekvens.



Figur 3: Sinuskurvor som visar övertoner till en grundton.

När man spelar flera toner vill man gärna att deras övertoner ska sammanfalla för att det ska låta harmoniskt och trevligt. För att olika toners övertoner ska sammanfalla behöver grundtonernas frekvens ha ett förhållande som är väldigt nära ett bråk med låga tal i täljare och nämnare, exempelvis 3/2 eller 5/4. Förhållandet 3/2 betyder att den tredje övertonen för grundtonen kommer sammanfalla med den andra övertonen för den nya grundtonen, vilket därför låter väldigt harmoniskt. Det är dock ganska sällan som tonerna kompletterar varandra så väl, vilket är anledningen till att de toner som gör det har fått speciella namn.

En av dessa toner är tersen, som är den tredje tonen i en skala. Den låter väldigt harmonisk, eftersom förhållandet mellan den och grundtonen är $2^{\frac{4}{12}} \approx 1,2599$, vilket är väldigt nära $5/4$. I vårt fall är tersen noten E. En annan av dessa toner är kvinten, som är den femte tonen i en skala. Likt, tersen låter den också harmonisk, eftersom förhållandet där är $2^{\frac{7}{12}} \approx 1,498$ som ligger nära $3/2$. I C-dur är den femte tonen G.



Figur 4: Kvinten med tillhörande gemensam överton.

När två toner har ett förhållande som är långt ifrån ett rationellt bråk med låga siffror skapas lätt dissonans istället för konsonans. Om noterna har en liknande frekvens som är förskjuten med cirka 10 Hz skapas väldigt stor dissonans [1, s. 137]. Detta är för att övertonerna också ligger förskjutna en multipel av frekvensskillnaden och kommer således inte sammanfalla.

5 Rationella approximationer

Förhållandet mellan frekvenserna av en ton och en annan ton k steg bort bestäms som nämnt tidigare av $2^{\frac{k}{12}}$ där k är ett naturligt tal. Förhållandet är ett heltal när $k \equiv 0 \pmod{12}$ och är annars irrationellt. Eftersom två toner harmoniserar om förhållandet mellan deras frekvenser ligger nära ett rationellt tal med liten nämnare måste de irrationella talen approximeras. I fallet när förhållandet, $2^{\frac{k}{12}}$, är irrationellt behövs alltså en algoritm som kan approximera det irrationella talet med hjälp av ett rationellt tal. I fallet när förhållandet är ett heltal vill vi att samma algoritm ger tillbaka samma heltal. Vi kommer använda en algoritm som med hjälp av kedjebråk löser båda fallen.

Ett kedjebråk är ett bråk på följande form:

$$c_0 + \cfrac{1}{c_1 + \cfrac{1}{c_2 + \cfrac{1}{\ldots + \cfrac{1}{c_n}}}}, \quad c_0, c_1, c_2, \ldots, c_n \in \mathbb{N}$$

som kan representeras av en lista, $[c_0; c_1, c_2, \ldots, c_n]$ där $c_i > 0$ för alla $i > 0$ [7, s.xiii]. Med hjälp av följande algoritm kan vi skriva varje reellt tal som ett kedjebråk [2, s.4–5]: Låt $X = [c_0; c_1, c_2, ..., c_n]$ vara ett rationellt tal. Då gäller det att c_0 är heltalsdelen av X. Sätt sedan $X_1 = \frac{1}{X - c_0}$, då gäller att

$$X_1 = c_1 + \cfrac{1}{c_2 + \cfrac{1}{\ldots + \cfrac{1}{c_n}}}$$

och c_1 är heltalsdelen av X_1. På det här sättet kan c_n rekursivt bestämmas:

$$\begin{aligned} &c_0 \text{ är heltalsdelen av} X, && X_1 = \frac{1}{X - c_0}, \\ &c_n \text{ är heltalsdelen av} X_n, && X_n = \frac{1}{X_n - c_n}. \end{aligned}$$

Då X är rationellt kommer det finnas ett X_n som är ett heltal så att $c_n = X_n$. Då blir X_{n+1} ej definierat och alla koefficienterna $c_0, c_1, c_2, \ldots, c_n$ är då bestämda. Om X istället är ett irrationellt tal går det ej att hitta ett X_n som saknar decimaldel och det kommer därför alltid gå att bestämma fler koefficienter till kedjebråket. I detta fallet beskrivs X som det oändliga kedjebråket:

$$\lim_{n \to \infty} [c_0, c_1, c_2, \ldots, c_n] = [c_0; c_1, c_2, \ldots].$$

Om X är rationellt går det att från det ändliga kedjebråket bestämma X på formen $\frac{p}{q}$ där $|X - \frac{p}{q}| = 0$ och p, q är heltal. I detta fall går det att utvärdera hur pass bra två frekvenser med förhållandet X harmoniserar endast genom att kolla på hur stor nämnaren är. Eftersom förhållandet mellan frekvenserna beror på $2^{\frac{k}{12}}$ som endast är rationellt då k är en multipel av tolv och då bestäms av $2^n, n \in \mathbb{Z}$ gäller det att nämnaren i detta fallet alltid kommer att vara 1. I fallet då X är irrationellt går detta inte att göra då kedjebråket är oändligt och två tal p, q ej går att bestämmas så att $|X - \frac{p}{q}| = 0$.

Låt $C = [c_0; c_1, c_2, \ldots]$ vara det oändliga kedjebråk som fås av ovanstående algoritm då X är det irrationella tal som approximeras. Då C innehåller oändligt många koefficienter är det omöjligt att bestämma C som ett bråk.

Vi låter därför $C_n = [c_0; c_1, \ldots, c_n]$ beteckna kedjebråket C fram till och med koefficienten c_n. Problematiken är nu att bestämma n så att $C_n = \frac{p}{q}$ där q är liten samtidigt som $|X - \frac{p}{q}|$ är litet. Det gäller att approximationen blir bättre för större n, alltså att $|X - C_{n+1}| < |X - C_n|$ [8, s.44]. Vi vill därför undersöka om det är möjligt för nämnaren av C_{n+1} att vara mindre än nämnaren C_n eftersom det då gäller att C_{n+1} är en bättre kandidat med avseende på båda kriterierna jämfört med C_n.

För att undersöka detta måste vi ha en algoritm för att få fram två relativt prima tal p, q så att $C_n = \frac{p}{q}$. Låt

$$\begin{aligned} A_{-1} &= 0, & A_0 &= b_0, \\ B_{-1} &= 0, & B_0 &= 1, \end{aligned}$$

och definiera rekursivt A_n, B_n som

$$A_n = b_n A_{n-1} + A_{n-2}, \quad B_n = b_n B_{n-1} + B_{n-2}$$

då gäller det för alla $n \geq 0$, att $C_n = \frac{A_n}{B_n} = \frac{p}{q}$ [6, s.5]. Eftersom A_n och B_n är växande kan q endast bli mindre om A_n och B_n har gemensamma faktorer det vill säga inte är relativt prima. Två heltal x, y som är relativt prima kan skrivas på formen $ax + by = 1$, där a och b är heltal [7, s.75]. Låt oss nu undersöka om A_n och B_n är relativt prima. Den rekursiva formeln för att beräkna A_n och B_n kan beskrivas med hjälp av multiplikation av matriser [6, s.6].

$$\begin{bmatrix} 1 & b_0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & b_1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & b_2 \end{bmatrix} \cdots \begin{bmatrix} 0 & 1 \\ 1 & b_n \end{bmatrix} = \begin{bmatrix} A_{n-1} & A_n \\ B_{n-1} & B_n \end{bmatrix}. \tag{1}$$

Genom att beräkna determinanterna i ekvation 1 får vi

$$A_n B_{n-1} - A_{n-1} B_n = (-1)^{n-1}.$$

För udda respektive jämna n gäller det då:

$$\begin{aligned} A_n B_{n-1} - A_{n-1} B_n &= 1, \\ A_{n-1} B_n - A_n B_{n-1} &= 1. \end{aligned}$$

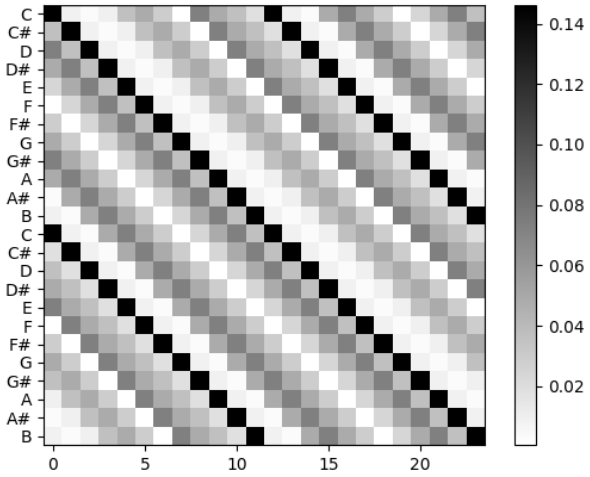
Eftersom vi vet att A_n, B_n, A_{n-1} och B_{n-1} är heltal så gäller det därför att A_n, B_n är relativt prima och det kommer aldrig gälla att nämnaren av C_{n+1} är mindre än nämnaren av C_n.

Eftersom approximationen av ett irrationellt tal blir bättre och bättre samtidigt som nämnaren av bråket växer för varje iteration av algoritmen behöver vi välja antalet iterationer så att approximationen är bra men bråket litet. Vi väljer att utvärdera approximationen efter 3 iterationer, alltså C_2. Vi viktar

sedan hur bra approximationen $C_2 = \frac{p}{q}$ av $2^{\frac{k}{12}}$ är med hjälp av följande uttryck:

$$\frac{1}{|2^{\frac{k}{12}} - \frac{p}{q}|^2 + q}.$$

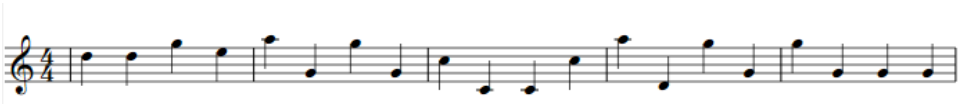
Detta eftersom vi vill att värdet av uttrycket ska bli större då approximationen blir bättre samt när q är litet. Genom att beräkna C_2 för $2^{\frac{k}{12}}$ för alla $0 \leq k \leq 23$, vikta approximationen med ovanstående uttryck och sen normera kan matrisen i figur 5 skapas.



Figur 5: Matris där position i, j anger hur stor sannolikheten är att en ton j steg bort spelas om föregående ton var i. Y-axeln sträcker sig över två oktaver.

6 Andra musikresultatet

Genom att använda matrisen från figur 5 som anslutningsmatris till vår markovkedja kan följande musikstycke genereras:



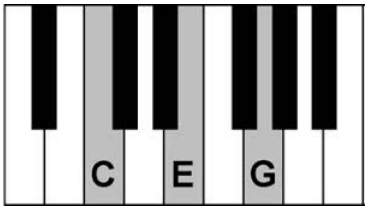
Figur 6: Musikstycke med förbättrad matris. Tryck här för att lyssna.

Om man jämför musiken från figur 6 med musiken från figur 2 ser musiken från figur 6 ut att följa mer av ett mönster jämfört med figur 2 där tonerna ser ut att hoppa mer. Det är ännu tydligare när man lyssnar på stycket att

det är en stor förbättring. Eftersom vi inte spelar några ackord till denna musiken har vi valt att spela i tonartens skala, alltså C-durskalan. I C-skalan spelas endast det vita tangenterna och därför spelas bara vita tangenter i figur 6.

7 Ackord

Ett ackord är tre eller fler noter som spelas samtidigt. De vanligaste ackorden kallas treklang. En viss nots treklang byggs upp av noten själv plus dess ters och kvint. Ett ackord ger fyllighet till musiken och spelas ofta tillsammans med en melodi. Melodin som spelas till ett ackord använder ofta noter från ackordets skala enligt musikteoretiska normer. Till exempel bör melodin som spelas tillsammans med ett D-moll ackord bestå av noter från D-moll skalan.



Figur 7: C-dur är ett treklangackord med C som grundton.

I en viss tonart kommer samtliga treklanger i dess skala låta bra i följd. En låt har ofta definierat en ackordföljd, en följd av ackord (ofta med storlek 4) som repeteras under låten. Vi såg i föregående stycke att det finns regler för vilka toner som låter bra i följd, men det finns inga motsvarande regler för vilka ackord som låter bra i följd. Det finns dock vissa ackordföljder som är vanligare än andra. I populärmusik är följden 1,5,6,4 mycket vanlig, där 1 representerar treklangen med tonartens grundton som grundton. På motsvarande sätt representerar 4 treklangen med den fjärde tonen i tonartens skala som grundton, och så vidare.

I detta projekt har vi valt ackordföljden 1,5,6,4. Eftersom vi spelar i tonarten C-dur motsvarar detta ackorden C, G, Am, F. När vi väljer melodi har vi tidigare använt en markovkedja där nästa ton baseras på förhållandet till nuvarande ton. Vårt nya kriterium för tonerna i melodin är att de måste passa bra ihop med ackordet som spelas. Dessa två sannolikheter kan kombineras på olika sätt. Antingen genom att addera dem, eller genom att multiplicera dem, och sedan normera. En hög sannolikhet multiplicerat med en sannolikhet på 0 blir 0, medan den resulterade sannolikheten fortfarande är ganska hög om man istället kombinerar dem med addition. Multiplikation är därför en ”hårdare” teknik.

Generellt gäller att två sannolikheter p och q adderas när p **eller** q ska ske. De multipliceras när p **och** q ska ske. Till exempel: Sannolikheten att man med en sexsidig tärning slår 1 eller 6 är $\frac{1}{6} + \frac{1}{6} = \frac{1}{3}$, sannolikheten att slå 1 och 6 i följd är $\frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36}$. I vårt fall vill vi att noten ska fungera bra med föregående not **och** med ackordet. Vi väljer därför att multiplicera.

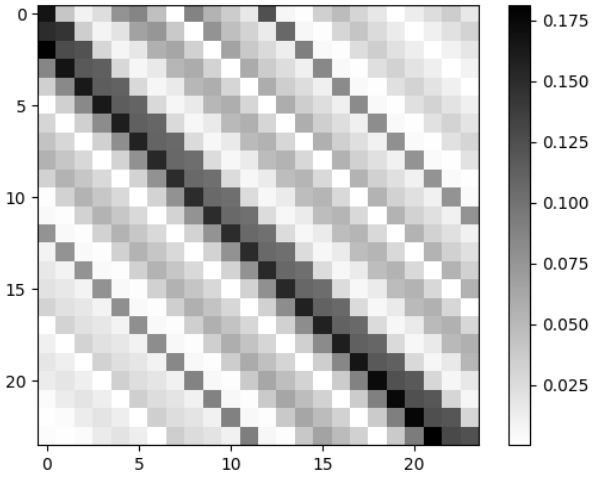
Om man anser att den ena sannolikheten är viktigare än den andra kan man vikta dem olika när man kombinerar dem. Detta kan göras additivt genom att multiplicera sannolikheterna med konstanter $c_1 \cdot p \cdot c_2 \cdot q$. Alternativt multiplikativt genom att höja sannolikheterna med en exponent, $p^{c_1} \cdot q^{c_2}$. I vårt fall blir additiv viktning meningslös eftersom alla nollskiljda sannolikheter i ackord matrisen är samma. Vi ser heller ingen musikalisk anledning att multiplikativt vikta.

8 Intervallängd

Längden på frekvensintervallet mellan två efterföljande noter har stor påverkan på hur vi upplever musiken. För det mesta är dessa intervall mycket korta, en eller två semitoner, men ibland kan de vara längre. För att implementera detta beteende i den genererade musiken skapar vi ytterligare en stokastisk matris D. Elementen i D har (innan normering) värdena

$$D[i][j] = \frac{1}{|i - j| + c},$$

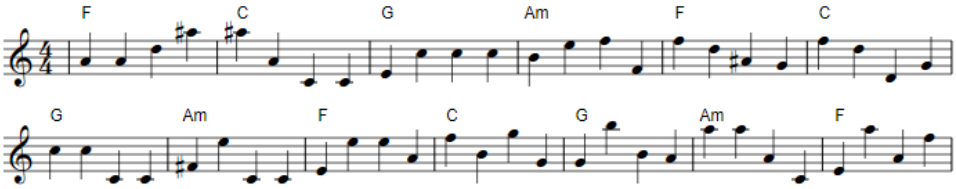
där i och j är numreringen på noterna vi mäter intervallängd mellan. På så sätt får långa interval lägre sannolikhet än korta intervall. Konstanten $c > 0$ är en utjämningskonstant som hindrar division med 0. Ett större värde på c ökar sannolikheterna att intervalet är långt. I vår implementation har c valts till $1,4$. Implementationen av denna nya stokastiska matris liknar implementationen av matrisen för ackord. Matrisen multipliceras elementvis med de andra stokastiska matriserna.



Figur 8: Den slutgiltiga matrisen för markovkedjan innan noter utanför ackordets skala har filtrerats ut.

9 Tredje musikresultatet

Precis som tidigare har vi genererat noterna genom att traversera en markovkedja där noderna är noter från två oktaver. Till skillnad från tidigare musikresultat bygger markovkejan nu inte bara på den relativa frekvensens rationella approximation, utan också på storleken på intervallängden till tidigare not och ackordet som spelas. Som tidigare beskrivet har ackordföljden valts till C, G, Am, F. Ackorden spelas vid varje ny takt.



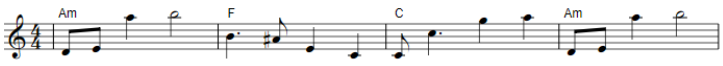
Figur 9: Musikstycke med ackord och intervallängd. Tryck här för att lyssna.

I figur 9 ser vi resultaten från dessa implementationer. Det första ackordet har slumpmässigt valts till F och sedan följs ackordföljden. Det är inte uppenbart från figuren att intervallängden har blivit kortare, men eftersom att de baseras på slump behöver inte ett exempel nödvändigtvis visa det. Slutligen ser vi att svarta tangenter nu tillåts att spela. Detta eftersom noterna väljs i ackordets skala, till skillnad från föregående musikresultat där de valdes ur tonartens skala.

10 Fraser

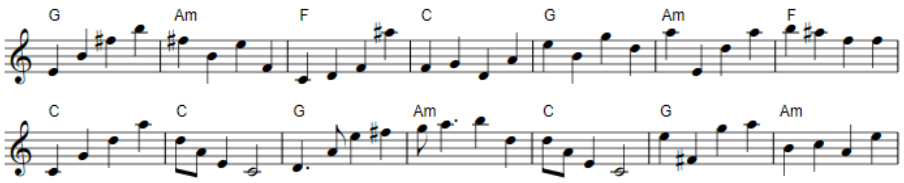
Det som kallas fraser inom musikteorin är notoriskt svårdefinierat. En fras ska på något sätt förmedla en musikalisk känsla eller tanke. Det finns alltså nästan inga kriterier som objektivt kan bestämma vad som är en fras och vad som inte är det. Oftast är frasen fyra takter lång, men även det kan variera. Låtar som saknar fraser saknar också identitet, eftersom fraserna är det som ger känslan i musiken. Alltså, även om det vi gjort innan låter ”bra” saknar det karaktär. Det svåra med att generera fraser, till skillnad från det vi gjort tidigare, är att frasen ska ha en identitet, det kan alltså inte bara vara en samling noter som låter bra tillsammans.

För att uppnå detta, skapade vi en markovkedja som baseras på de två föregående noterna. Den nya markovkedjan gör att frasen får ett mönster i sin struktur, vilket förhoppningsvis skapar en musikalisk känsla. Den genererade frasen består av fyra takter, där den sista alltid är identisk till den första. Förutom det, gjorde vi så att frasen innehåller andra takter än resten av sången. Än så länge har vi endast genererat takter som består av fyra fjärdedelsnoter, men frasen består istället av tre olika takttyper. Den första och sista takten består av två åttondelar, en fjärdedel samt en halvnot. De två takterna i mitten består istället av en tre-åttondelarsnot, en åttondel och två fjärdedelar.



Figur 10: Exempel på en fras.

I figur 10 kan vi tydligt se att noterna föredrar att fortsätta uppåt om de redan stiger, respektive nedåt om de redan faller. Figuren nedan visar hur en fras kan se ut i en låt. Det är tydligt var frasen är, eftersom det endast är där halvnoter, tre-åttondelsnoter och åttondelsnoter kan förekomma. Det finns även en tydlig skillnad i strukturen där frasen är, eftersom den har uppenbart mönster i sina upp- och nedgångar, vilket den övriga låten inte har.



Figur 11: Exempel på fras i låt.

Frasens implementation speglar mycket av den genereringen som redan utförs. Matriserna som redan finns återanvänds för skapandet av frasen, med små förändringar i matrisen som viktar baserat på avstånd (se kapitel 7). Skillnaden är att vi istället använder två olika matriser; den ena används för stigande noter, medan den andra för fallande noter. Dessa konstrueras så att noterna under respektive över den nuvarande noten får avsevärt lägre chans att väljas genom en symmetrisk viktning. Det betyder att noterna som är ovanför noten i den stigande matrisen multipliceras med vikten, medan noterna under istället divideras med vikten och motsvarande för den fallande matrisen.

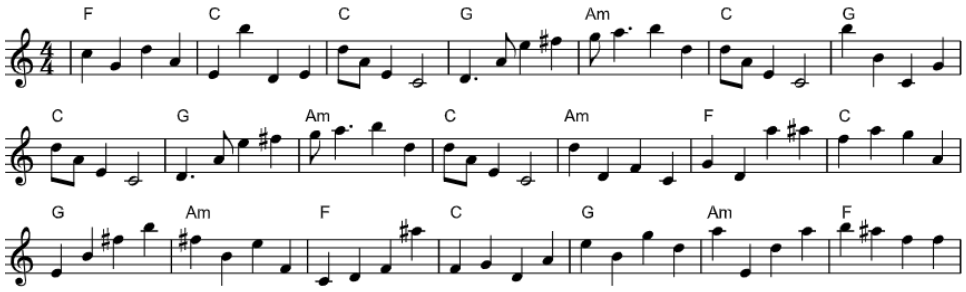
Vi utforskade två olika sätt att inkludera frasen i låten men i båda varianterna började vi att generera frasen och därefter generera låten. Den enklare varianten är att i början av varje takt slumpa fram om frasen ska väljas istället för en vanlig takt. Konsekvenser av detta är att låten får olika längd baserat på hur ofta frasen inkluderas i låten och frasen inte alltid passar bra med den tidigare noten. Det är dock enklare att bestämma den exakta frekvensen av frasen i låten, till skillnad från den mer avancerade metoden.

Den något mer avancerade varianten är att istället lägga till en ny rad och kolonn som representerar frasen i matrisen så att den får storleken 25x25. Kolonnen representerar varje nots chans att välja frasen, vilket ska vara den samma som chansen att välja den första noten i frasen. Raden är chansen att välja andra noter efter frasen, vilket bör vara identisk med raden för den sista noten i frasen, förutom att chansen att välja frasen igen rimligen är noll. Konsekvenser av den mer avancerade varianten är att frasen spelas mycket mer sällan, ty det måste både vara en ny takt och den föregående noten måste passa med den första noten i frasen. Ett sätt att lösa detta som vi inte utforskade är att göra modifieringar i frasen medan resten av låten genereras. Det skulle göra att frasen kan spelas oftare och att den blir något mer intressant.

Vi valde att använda den enklare varianten i resterande delen av låtskapandet, eftersom den är enklare att modifiera, även om den mer avancerade metoden antagligen skulle gett ett bättre ljud i slutändan. Den automatiskt genererade frasen saknar tanken bakom sig som en traditionellt komponerad fras hade haft. Den innehåller dock struktur och bryter upp den relativt monotona låten. Frasens utseende gör att den blir tydlig även när låten spelas upp, vilket skapar en avsevärt mer intressant låt än tidigare.

11 Slutgiltigt musikresultat

Efter att ha kombinerat vår tredje iteration med fraserna kan vi nu skapa något längre musikstycken som trots längden inte blir tråkiga, se bilaga 1.



Figur 12: Musikstycke från genererad låt, del av bilagan. Tryck här för att lyssna på hela bilagan.

Vi kan modifiera frekvensen av frasen i låten genom att öka eller minska sannolikheten att frasen tas med. Vi bestämde oss för att $\frac{1}{10}$ verkar bäst för att göra låten balanserad.

12 Avslutande ord

Sammanfattningsvis har vi sett att musik kan genereras med hjälp av markovkedjor. Att utvärdera resultaten är komplicerat eftersom musikkvalité är subjektivt. Kvalitén är dock inte sådan att det är något vi skulle vilja lyssna på. Detta kan vara ett resultat av antagandet i markovkedjan att en not enbart beror på föregående not, alltså bara ett steg bak i historiken. Intuitivt bör noter bero på fler steg bak i historiken. Modern musikgenereringsteknik bygger ofta på LSTM, ett neuralt nätverk där historik propageras många tidssteg [3].

En förbättring som kan göras är angående hur många iterationer genom algoritmen vi gör för att bestämma en approximation. För tillfället är antalet iterationer detsamma för varje tal som approximeras. En bra idé hade varit att variera detta antalet iterationer på det sättet att iterationen bryts när en stor koefficient hittas. Detta eftersom en stor koefficient inte förbättrar approximationen mycket men ökar storleken på q väsentligt. Viktningen av våra approximationer hade varit annorlunda, matrisen i figur 5 hade därav sett annorlunda ut och den resulterande musik hade kanske låtit bättre.

För fortsatt arbete bör taktförändringar utforskas, så att det blir större variation i låten. Även ackordförändringar, så som valsstil, eller liknande, bör tas hänsyn till för att göra så att låten känns mer flytande och föränderlig.

Vi kan även konstatera att frasen bör implementeras på ett snyggare sätt, som bättre anpassas till musikstyckets helhet. Det stora problemet med våra automatiskt genererade låtar är att de antingen känns väldigt monotona eller att de blir kaotiska. Lösningarna ovan är antagligen tillräckligt för att minska detta problem avsevärt.

Referenser

[1] Benson, D.J. (2007) *Music: A Mathematical Offering*. Cambridge: Cambridge University Press. https://www.cambridge.org/9780521619998

[2] Bugeaud, Y: *Approximation by Algebraic Numbers*. Cambridge University Press 2004

[3] Carnovalini, C. (2020) *Computational Creativity and Music Generation Systems: An Introduction to the State of the Art*

[4] Euler, L: *Letters of Euler to a German princess, on different subjects in physics and philosophy*. Murray and Highley Press 1802

[5] Grimaldi, R: *Discrete and Combinatorial Mathematics, 5th edition*. Pearson Education Press 2014

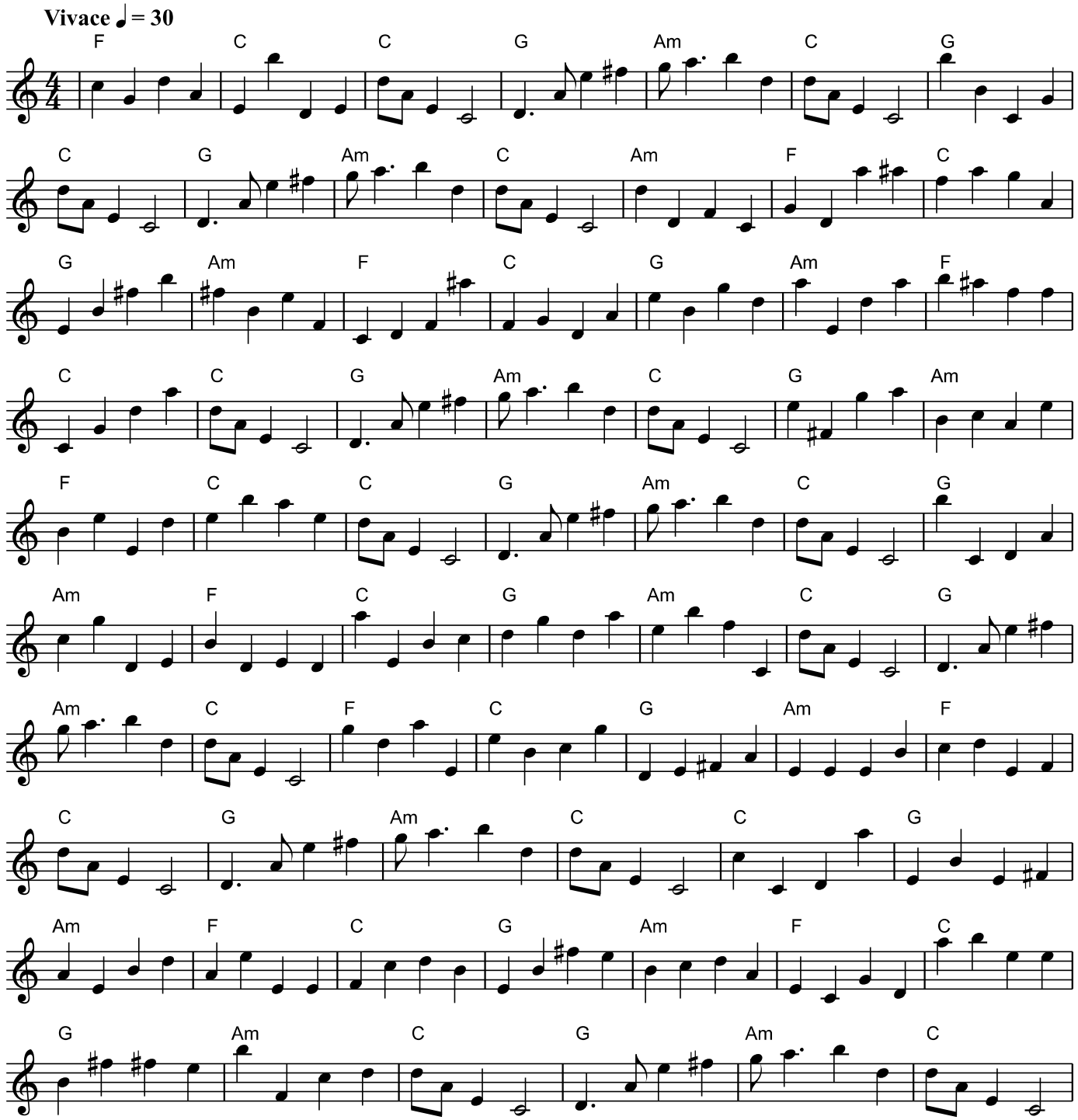
[6] Persson, T: *Topics in Analysis*. https://www.maths.lth.se/~tomasp/kurser/filer/tia.pdf (hämtad 2022–02–21)

[7] Simonson, A: *Exploring Continued Fractions: From the Integers to Solar Eclipses*. MAA press 2019

[8] Steuding, J: *Diophantine Analysis*. Chapman & Hall/CRC 2005

Markov with phrases

markov.py



Tryck för att lyssna.