

A contribuição de Max Mathews para a Música Computadorizada

Marcos Luis Duarte¹
USP/MESTRADO
SIMPOM: *Sonologia*
bluetriumph@gmail.com

Resumo: A síntese sonora digital teve seu início quando Max Mathews, engenheiro do *Bell Laboratories*, foi estimulado por seu chefe a dedicar-se exclusivamente a produzir programas que tornassem possível a composição de música através de computadores. As pesquisas de Mathews culminaram com a série de programas para síntese sonora e sequenciamento conhecidos como *MUSIC N*. Dentre estes, o *MUSIC III* trouxe a maior novidade da família: o conceito de unidade geradora: uma função pré-programada que simulava o comportamento de módulos eletrônicos presentes em sintetizadores analógicos. Através das unidades geradoras era possível simular osciladores, filtros, envelopes de ataque e decaimento, dando ao usuário a oportunidade de conectar os distintos blocos para então gerar um som específico. Era possível também implementar técnicas conhecidas de síntese sonora como a aditiva, a subtrativa, dentre outras mais complexas, como a síntese FM. Dentre todos os programas da família *MUSIC* desenvolvidos por Mathews, porém, o mais importante foi o *MUSIC V* que, dentre outras melhorias, foi implementado através da linguagem *FORTRAN* para os computadores da linha IBM360, mais rápidos, estáveis e, principalmente, presentes em diversas instituições da época. Isto permitiu que o *MUSIC V* fosse executado em diferentes lugares, divulgando o conceito de síntese sonora digital e, com isso, expandindo o número de pesquisadores interessados. Este processo permitiu que outras versões do programa fossem escritas em centros de pesquisas como, por exemplo, o da Universidade de Princeton e de Stanford, dando origem a novas versões, melhoradas e adaptadas aos computadores mais modernos. Esta linha de sucessivas melhorias estendeu-se até o atual *Csound*, amplamente utilizado em performances de música eletroacústica até os dias atuais.

Palavras-chave: Max Mathews; Síntese sonora digital; Programação; *MUSIC N*; Sonologia.

The Contribution of Max Mathews for Computer Music

Abstract: The digital sound synthesis began when Max Mathews, a Bell Laboratories engineer, was encouraged by his boss to exclusively produce programs that would make possible the music composition through computers. The Mathews' research culminated in the range of programs for sound synthesis and sequencing known as *MUSIC N*. Among these, the *MUSIC III* brought the biggest news of the family: the concept unit generators: a pre-programmed function that simulated the behavior of oscillators present in analog synthesizers. Through unit generators was possible to simulate oscillators, filters, attack and decay envelopes, giving the users the opportunity to connect different blocks and then generate a

¹ Orientador: Rodolfo Coelho de Souza. Agência de fomento de bolsa: CAPES.

specific sound. It was also possible to implement techniques known like the additive sound synthesis, the subtractive, among other more complex, such as the Frequency Modulation synthesis. Among all MUSIC family programs developed by Mathews, however, the most important was the MUSIC V which, among other improvements, was implemented by FORTRAN language for IBM360 computers series, which was faster, stable and present in diverse institutions at the time. Thus, the MUSIC V was performed in different places, spreading the concept of digital sound synthesis and thereby expanding the number of interested researchers. This process allowed other versions of the program, written in research centers such as the University of Princeton and Stanford, allowing new versions, improved and adapted to modern computers. This line of successive improvements culminated with current Csound program, widely used in performances of electroacoustic music today.

Keywords: Max Mathews; Digital sound synthesis; Programming; MUSIC N; Sonology.

Introdução

O campo da música computadorizada (*computer music*) foi inicialmente concebido com a síntese sonora. Na década de 1950, o trabalho de Max Mathews, engenheiro eletrônico, abriu caminho para a síntese sonora digital. Na década de 1970, Mathews foi o pioneiro em *live performance* baseada em computador. Colaborou com diversos artistas e compositores como John Cage, Edgard Varèse, Stockhausen, dentre outros. Seus estudos influenciaram gerações de pesquisadores e entusiastas da área de síntese sonora digital, como John Chowning e Jean-Claude Risset. O trabalho de Mathews foi o ponto inicial que marcou a produção de som através de computadores. Posteriormente, Mathews ficou conhecido como o pai da *computer music*.

Os programas desenvolvidos por Mathews (*MUSIC I - V*) por volta da década de 1960 foram as primeiras linguagens de programação para a produção de som. O livro por ele escrito, *Technology of Computer Music (M.I.T. Press, 1969)*, influenciou gerações de pesquisadores e entusiastas da área. Nele, Mathews apresenta os conceitos físicos e matemáticos por trás da síntese sonora, além de um manual com diversos exemplos de implementação de instrumentos através do programa *MUSIC V*.

1. A matemática básica do som

Na abordagem de Mathews, foi decisivo o modelo do som como função matemática. Em termos gerais, a função mais genérica que podemos imaginar é $y=f(x)$, onde o valor de y é dado pelo resultado da ação da função f sobre o parâmetro x . Mathews propôs a digitalização da onda sonora, uma onda mecânica que, para se propagar, é dependente do

meio em que se encontra (o ar, por exemplo), às funções matemáticas. Mais especificamente, o som é fruto da variação da pressão do ar através do tempo.

As características subjetivas do som, ou seja, como ele 'soa', depende da maneira específica de como a pressão varia. Por exemplo, um som com uma altura definida como Lá, acima do Dó central, possui uma variação periódica da pressão que se repete 440 vezes por segundo. Uma pressão constante é lida como silêncio. (MATHEWS, 1969, p. 2).

Por causa disso, a onda sonora pode ser escrita matematicamente como uma função digitalizada da pressão. Ou seja, a função f foi definida como p (de pressão) e a incógnita x como t (de tempo), resultando na expressão $som=p(t)$. Tal relação descreve a variação da pressão à medida que o tempo passa – e o resultado físico disso é o que entendemos por som.

2. A série de programas *MUSIC N*

A caminhada de Max Mathews teve início em 1954, logo após concluir seu doutorado em engenharia elétrica no *M.I.T. (Massachusetts Institute of Technology)*, quando então foi contratado pela *Bell Telephone Laboratories*, empresa que liderava as pesquisas em acústica e telefonia na época. Lá, Mathews foi o coordenador da pesquisa em acústica e comportamento, permanecendo na empresa até 1985. Inicialmente, trabalhou na codificação de voz para transmissão a longas distâncias, como o cabo submarino para a Europa. Segundo Mathews, “foi um processo bem caro, demorado e desencorajador, pois a fala é bastante complexa (DAYAL, 2011)”. Apesar disso, “mostrou-se um processo de pesquisa muito bem sucedido. (...) É hoje o processo de comunicação universal; quase todos os sinais de comunicação atuais são codificados digitalmente.” (DAYAL, 2011)

O chefe de Mathews na *Bell Labs*, John Pierce, era pianista e um entusiasta da música. Certa vez, no intervalo de um concerto em que foram juntos, Pierce perguntou: “Max, se você pode obter o som de um computador, seria possível então escrever um programa para sintetizar música no computador?” (DAYAL, 2011). Com este “impulso” e a aprovação de seu chefe, Mathews decidiu então mudar o foco de sua pesquisa. Ao invés de codificação de voz, passou a considerar a ideia de fazer música através do computador. Mais tarde, sobre as dificuldades e soluções encontradas durante o processo de pesquisa da síntese sonora digital, escreveu que

Os dois problemas fundamentais na síntese sonora são (1) a grande quantidade de dados necessários para especificar uma função de pressão – daí a necessidade de um programa de computador bastante rápido e eficiente – e (2) a necessidade de uma

linguagem simples e poderosa capaz de descrever a complexa sequência de sons. Nossa solução para estes problemas envolve três princípios: (1) funções armazenadas para agilizar a computação, (2) blocos de construção da unidade geradora para proporcionar grande flexibilidade aos instrumentos sintetizadores e (3) o conceito de nota musical para descrever sequências sonoras. (MATHEWS, 1969, p. 34).

Com a evolução de seus estudos no campo da análise, síntese e percepção de timbres sonoros, mostrou-se necessária alguma ferramenta que, a partir dos computadores digitais, traduzisse as funções matemáticas e produzisse, como resultado final, o som. A partir desta necessidade, Mathews passou a se preocupar em criar os programas capazes de transformar em realidade física, auditiva, os conceitos matemáticos estudados.

O avanço das pesquisas em computação musical culminou, em 1957, com a linguagem de programação *MUSIC I*, pioneira na síntese de áudio digital. Escrita em *Assembler* (linguagem de programação que não possui os meios de definição de estruturas de alto nível; por isso suas rotinas possuem uma estrutura bem próxima à chamada linguagem de máquina, com seus acumuladores, endereçamentos diretos à memória, e assim por diante), estava limitada às restrições técnicas do computador IBM704. Podia-se controlar a frequência, amplitude e a duração das notas, que eram produzidas apenas por simples formas de ondas triangulares e monofônicas. Embora o som produzido “não seria considerado música pelas pessoas” (DAYAL, 2011) por ser bastante rudimentar e gerar apenas uma voz, uma forma de onda e um único timbre, a linguagem de fato produzia tons, notas e escalas. Em 1957, Max Mathews e Newman Guttman criaram uma peça de 17 segundos usando o *MUSIC I*, chamada *The silver scale*. Esta peça é considerada a primeira composição gerada a partir do computador (ROADS, 1996, p.89). No mesmo ano foi também composta outra peça de cerca de 1 minuto, chamada *Pitch Variations* (ROADS, 1996, p. 89).

Seguiu-se o *MUSIC II*, que conseguiu gerar até quatro vozes. Era também capaz de gerar 16 formas de ondas através da introdução de um oscilador que lia os dados de uma tabela pré-gravada (*wavetables*). Por isso, o *MUSIC II* era mais versátil e funcional que a versão anterior, apesar de ainda utilizar o *Assembler*. Outra característica que contribuiu para uma melhor performance do *MUSIC II* é que foi escrito a partir dos computadores IBM7094, à base de transistores, ou seja, com um processamento mais rápido que os antigos computadores a válvula.

O grande avanço veio com o *MUSIC III* (1960), onde foi introduzido o conceito totalmente novo de unidade geradora para síntese sonora. Esta ideia simulava os módulos eletrônicos, como os utilizados nos sintetizadores analógicos Moog, surgidos na mesma época.

Uma unidade geradora é essencialmente uma função discreta pré-construída dentro do programa, que simula osciladores, filtros, envelopes e assim dá a possibilidade de conectar estas múltiplas unidades para gerarem um som específico. Ou seja, é um bloco ou módulo fundamental usado para construir uma variedade de algoritmos geradores de som. Cada unidade geradora aceita parâmetros numéricos e/ou sinais de áudio como entrada e produzem um sinal de saída. Com isso, a programação modular estruturava o programa a partir de unidades menores que poderiam ser reutilizadas. Foi também adicionada uma seção separada, chamada de “partitura”, onde os sons gerados podiam ser arranjados musicalmente em ordem cronológica. A técnica da unidade geradora foi estendida às linguagens *MUSIC IV* e *MUSIC V*.

O *MUSIC IV* (1963) foi uma colaboração entre Mathews e Joan Miller. Era uma versão mais completa que o *MUSIC III*, mas com poucas novidades. Foi apenas uma nova implementação, devido à mudança do computador da empresa, segundo admite o próprio Mathews: “*MUSIC IV* foi apenas uma resposta à mudança na linguagem e no computador. Então, em essência, o *MUSIC IV* não era musicalmente mais poderoso que o *MUSIC III*” (FUCHS, 1988, p.39). Algumas outras versões do *MUSIC IV* foram criadas: uma na Universidade de Princeton, (*MUSIC IVB*) e outra no *Argonne Labs* (*MUSIC IVBF*). Estas versões foram construídas com a linguagem *FORTRAN*, ao invés do *Assembler*.

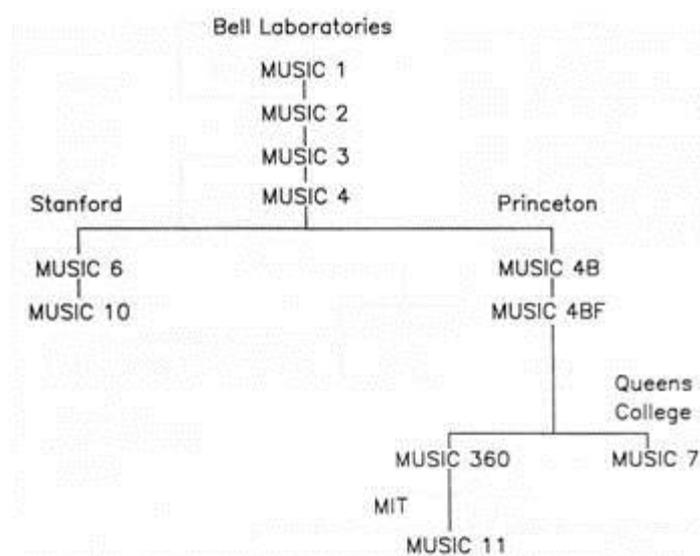


Fig. 1: Árvore da família de programas de síntese sonora baseados no MUSIC N (DODGE & JERSE, 1997, p. 17).

Para o *MUSIC V* (1967), Mathews também utilizou o *FORTRAN* especificamente para a linha IBM360 de computadores, mais rápidos e estáveis. Com esta mudança, o programa poderia rodar em qualquer outro IBM360 fora da *Bell Labs*.

O último programa que escrevi, o *MUSIC V*, ficou pronto em 1967. Este foi meu último programa porque eu o escrevi em *FORTRAN*. O *FORTRAN* ainda é usado hoje, ainda vai muito bem de saúde, então é possível recompilar os códigos fontes para as novas gerações de computadores. Vercoe (criador do Csound) escreveu para o IBM360 e, quando esta linha de computadores morreu, escreveu outro programa chamado *MUSIC II* para o PDP-11; quando este também morreu, escreveu ainda um novo programa na linguagem C chamado *CSound*. Este, novamente, é um compilador e ainda uma linguagem viva. É, na verdade, a principal linguagem de hoje. Dessa forma, ele não teve que escrever nenhum outro programa. (DAYAL, 2011).

As unidades geradoras do *MUSIC V* incluíam oscilador, filtro, adicionador, multiplicador, gerador numérico aleatório (randômico) e gerador de envelope. Os módulos básicos de processamento de sinais e síntese poderiam ser interligados para criar sons interessantes. Técnicas como a aditiva, subtrativa e síntese não linear (como síntese FM) poderiam ser implementadas facilmente com esses módulos. Eles eram similares em função aos módulos utilizados nos sintetizadores analógicos da época, como os VCOs (*Voltage-Controlled Oscillators*), VCAs (*Voltage-Controlled Amplifiers*) e os VCFs (*Voltage-Controlled Filters*). Os instrumentos no *MUSIC V* eram escritos como uma rede de unidades geradoras interligadas. A chamada para um instrumento era essencialmente uma sub-rotina que passava os argumentos (*pfields*) necessários ao instrumento. Uma partitura do *MUSIC V* era uma sequência ordenada de chamadas de instrumentos.

A figura 2 ilustra o processo pelo qual os programas de síntese sonora da família *MUSIC N* produzem som. No primeiro passo (*STEP 1^a*) o usuário pode criar uma orquestra através da definição de instrumentos. Cada instrumento é designado para produzir um tipo particular de som e possui parâmetros que controlam várias características do som, como a duração, a dinâmica e a frequência. O segundo passo (*STEP 1b*) permite a inserção dos parâmetros particulares da partitura, de acordo com o escopo do compilador. Estes dois passos completam a entrada de dados necessária para a produção do som. Os passos 2 e 3 (*STEP 2^a, 2b e 3*) são executados pelo computador, sem a intervenção do usuário. No passo 2 (*STEP 2*), as definições de instrumentos são traduzidas para linguagem de máquina e os dados presentes na partitura são compilados para que esteja no formato apropriado para o processamento do passo 3 (*STEP 3*). É no passo 3 que o computador efetivamente “toca” a partitura, criando o som (DODGE & JERSE, 1997, p. 17).

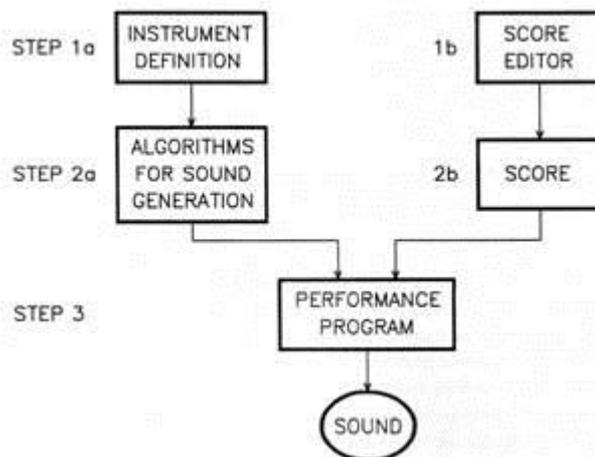


Fig. 2: Processo através do qual as linguagens da família *MUSIC N* produzem som (DODGE & JERSE, 1997, p. 18).

A figura 3, a seguir, apresenta a estrutura básica de um programa escrito no *MUSIC V*. No diagrama de blocos da unidade geradora (Fig. 3^a), a entrada da esquerda de cada oscilador é o controle de amplitude e a entrada da direita é o controle de frequência. Neste exemplo, o parâmetro P5 controla a amplitude, P6 define a duração da nota e P7 sua frequência. O oscilador superior funciona como o gerador do envelope da amplitude (Fig. 3b) para o oscilador de baixo. A função F1 (envelope da amplitude) desenha uma forma trapezoidal e F2 é a forma de onda resultante, que define o timbre da nota.

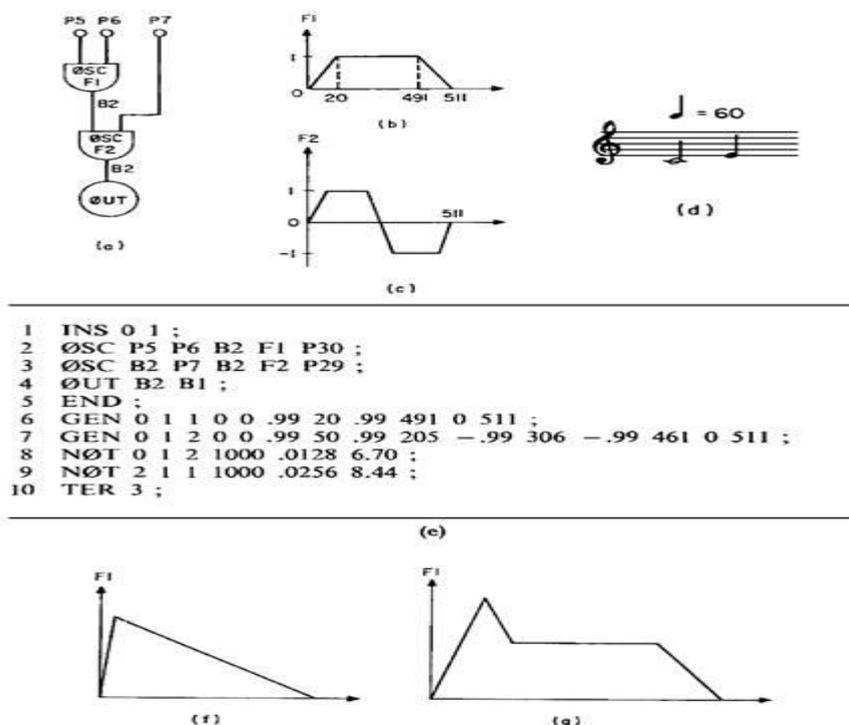


Fig. 3: Instrumento escrito com o *MUSIC V*: (a) diagrama de bloco; (b) função envelope; (c) função da forma de onda; (d) partitura convencional; (e) partitura do computador; (f) envelope típico do piano; (g) envelope típico de instrumento de metal (MATHEWS, 1969, p. 54).

Nos anos seguintes ao surgimento do *MUSIC V*, um grande número de centros de pesquisa com acesso a grandes computadores e conversores de som ampliaram os programas de síntese sonora e música computadorizada de diversas maneiras. Na Universidade de Stanford (CCRMA), por exemplo, descendentes do *MUSIC V*, como o *Mus10*, introduziu a possibilidade de nomear variáveis, entre outras melhorias para a definição dos instrumentos. O *Mus10* possuía também mais unidades geradoras embutidas, funções lineares por partes para uso como parâmetros de envelopes e um compilador de instrumento (MORRILL, 1977, p. 32). Outros descendentes do *MUSIC V* apareceram em inúmeras universidades e centros de pesquisas espalhados pelo mundo (vide Tab.1).

Mathews convenceu a *Bell Labs* a deixar os códigos do *MUSIC V* sem direitos autorais. A consequência disto é que foi um dos primeiros programas de código aberto a surgir. Este gesto garantiu a longevidade do programa, responsável direto pelo surgimento do atual *CSound*. O programa *MUSIC V* marcou o fim do envolvimento de Mathews com a série de programas que passou a ser conhecida como *MUSIC N*, programas que foram os modelos para todos os futuros programas de síntese sonora.

Ano	Versão	Local	Autor
1957	<i>MUSIC I</i>	<i>Bell Labs (New York)</i>	Max Mathews
1958	<i>MUSIC II</i>	<i>Bell Labs (New York)</i>	Max Mathews
1960	<i>MUSIC III</i>	<i>Bell Labs (New York)</i>	Max Mathews
1963	<i>MUSIC IV</i>	<i>Bell Labs (New York)</i>	Max Mathews, Joan Miller
1963	<i>MUSIC IVB</i>	<i>Princeton University</i>	Hubert Howe, Godfrey Winham
1965	<i>MUSIC IVF</i>	<i>Argonne Labs (Chicago)</i>	Arthur Roberts
1966	<i>MUSIC IVBF</i>	<i>Princeton University</i>	Hubert Howe, Godfrey Winham
1966	<i>MUSIC 6</i>	<i>Stanford University</i>	Dave Poole
1968	<i>MUSIC V</i>	<i>Bell Labs (New York)</i>	Max Mathews
1969	<i>MUSIC 360</i>	<i>Princeton University</i>	Barry Vercoe
1969	<i>MUSIC 10</i>	<i>Stanford University</i>	John Chowning, James Moorer
1970	<i>MUSIC 7</i>	<i>Queen's College (New York)</i>	Hubert Howe, Godfrey Winham
1973	<i>MUSIC 11</i>	<i>M.I.T.</i>	Barry Vercoe
1977	<i>Mus 10</i>	<i>Stanford University</i>	Leland Smith, John Tovar
1980	<i>CMusic</i>	<i>University of California</i>	Richard Moore
1984	<i>Cmix</i>	<i>Princeton University</i>	Paul Lansky
1985	<i>MUSIC 4C</i>	<i>University of Illinois</i>	James Beauchamp , Scott Aurenz
1986	<i>CSound</i>	<i>M.I.T.</i>	Barry Vercoe

Tab. 1: Família de programas MUSIC N e seus derivados.

Referências

DODGE, C., & JERSE, T. A. *Computer Music: Synthesis, Composition, and Performance*. 2nd ed. New York: Schirmer Books, 1997.

DAYAL, Geeta. *Max Mathews (1926-2011)*. Entrevista realizada em maio de 2011, disponível em <http://blog.frieze.com/max-mathews/> - acesso em 06/03/2014.

FUCHS, Mathias. Computer Music Languages... and the Real World. *Leonardo*. Supplemental Issue, Cambridge, MA, Vol. 1, Electronic Art (p. 39-42), 1988.

MATHEWS, M. V. *The Technology of Computer Music*. Cambridge, MA: MIT Press, 1969.

MORRILL, D. Trumpet Algorithms for Computer Compositions. In: ROADS, C., & STRAWN, J. (Eds.). *Foundations of Computer Music*. Cambridge, MA: MIT Press, 1987, (p. 30-44).

ROADS, C. *The Computer Music Tutorial*. Cambridge, MA: MIT Press, 1996.