Center for Computer Research in Music and Acoustics February 1987

Department of Music
Report No. STAN-M-37

THE COMPUTER AS A MUSICAL INSTRUMENT

by

Max V. Mathews and John R. Pierce

This is a reprint of an article that appeared in **Scientific American** in February 1987, Vol. 255, No. 2. It is a general introductory article to computer music written by the founders of the field.

CCRMA
DEPARTMENT OF MUSIC
Stanford University
Stanford, California 94305

an Article from SCIENTIFIC AMERICAN

FEBRUARY, 1987 VOL. 255 NO. 2

The Computer as a Musical Instrument

If a computer generates the right sequence of numbers, any sound—including some never before heard—can be produced. Because of its versatility, such digital sound synthesis has found a place in music

by Max V. Mathews and John R. Pierce

The French composer and conductor Pierre Boulez startled American concertgoers last year when he toured the country with an ensemble that included a computer and an array of electronic sound-modifying devices. Boulez' works for orchestra and computer-generated sound integrate digital electronic equipment in an active, sound-producing role that goes far beyond the more established role of such equipment in the recording and playback of symphonic music. Although computers are not yet a part of every symphony orchestra, digital sound synthesizers are already serving as alternatives to traditional instruments in the production of sound tracks for film and television, and they are quickly becoming the instruments of choice in popular music.

Behind all these efforts to incorporate the digital synthesis of complex sounds into music lies some pioneering work on the computer processing of sound, with which we were privileged to be involved, that began in the 1950's at the Bell Telephone Laboratories. We were originally drawn to the computer as a sound-analyzing and sound-producing device while investigating the factors that contribute to the efficient transmission of speech through telephone lines. It soon became clear to us that the quality of sound is of great importance not only in speech but also in music, and we enthusiastically began to study the production of musical sounds.

Although our first attempts to produce musical sounds from a computer were disappointing, the electronic instruments and computer programs that eventually evolved from these initial attempts are now sophisticated enough to have significant impact in several music-related areas. First, the instruments are of considerable commercial importance to the music-recording industry since they can efficiently generate music or sound effects that can be readily synchronized with action depicted in a motion picture or on television. Second, they provide a virtually limitless universe of sounds through which composers and performers can express their thoughts and feelings. Finally, and perhaps most important, they can deepen understanding of the particular patterns of sound called "music."

Reduced to its essential physical nature, sound is no more than a pressure fluctuation in the air. As such it can be expressed graphically by means of a waveform: a plot of how the ambient air pressure varies as a function of time. How sound is perceived, that is, whether it is pleasant or unpleasant, depends specifically on the way the various features of the pressure fluctuations are translated into nerve impulses in the ear and on the way the nerve impulses are then interpreted subjectively by the brain.

Sounds that are heard as having a definite pitch have waveforms that exhibit a nearly periodic variation in pressure. The pitch of a sound corresponds directly to the variation's frequency of repetition. For example, a pressure variation that repeats itself 440 times per second is perceived by a musician as a tone with a definite pitch: A above middle C.

Most sounds one hears originate from the vibrations induced in common, everyday objects: human vocal cords through which air is exhaled, violin strings that are bowed and automobiles that collide—to name a few. Sounds can also be generated from the vibrations set up in a loudspeaker by varying the voltage of its electrical input. Indeed, as any audiophile knows, an excellent reproduction of a given sound can be elicited from a good loudspeaker if an accurate voltage analogue of the sound's pressure function is applied to the speaker.

A mathematical verity known as the sampling theorem states that any waveform made up of multiple components of various frequencies can be exactly described by a sequence of numbers that give the value of the waveform's amplitude at a rate determined by the waveform's bandwidth, or range of component frequencies (conventionally expressed in hertz, or cycles per second). Specifically, the rate at which the numbers must be generated is equal to twice the bandwidth of the waveform. It is this theorem, proved by Claude E. Shannon of Bell Laboratories in 1948, that underlies all digital recording, processing and generation of sound.

The sampling theorem implies that one second of a sound whose bandwidth is 20,000 hertz (which spans the range of frequencies audible to the human ear) can be exactly recorded if 40,000 numbers, called samples, corresponding to the evenly spaced, instantaneous values of the sound wave's pressure amplitude (or the voltage analogue thereof) are collected during that one second. Conversely, if the appropriate 40,000 sample values per second can be summoned, any perceptible sound could conceivably be produced in all its acoustic intricacy. The compact disc, on which about 40,000 samples per second of sound are encoded and stored as points of varying reflectance, is an example of such a storage-and-retrieval system.

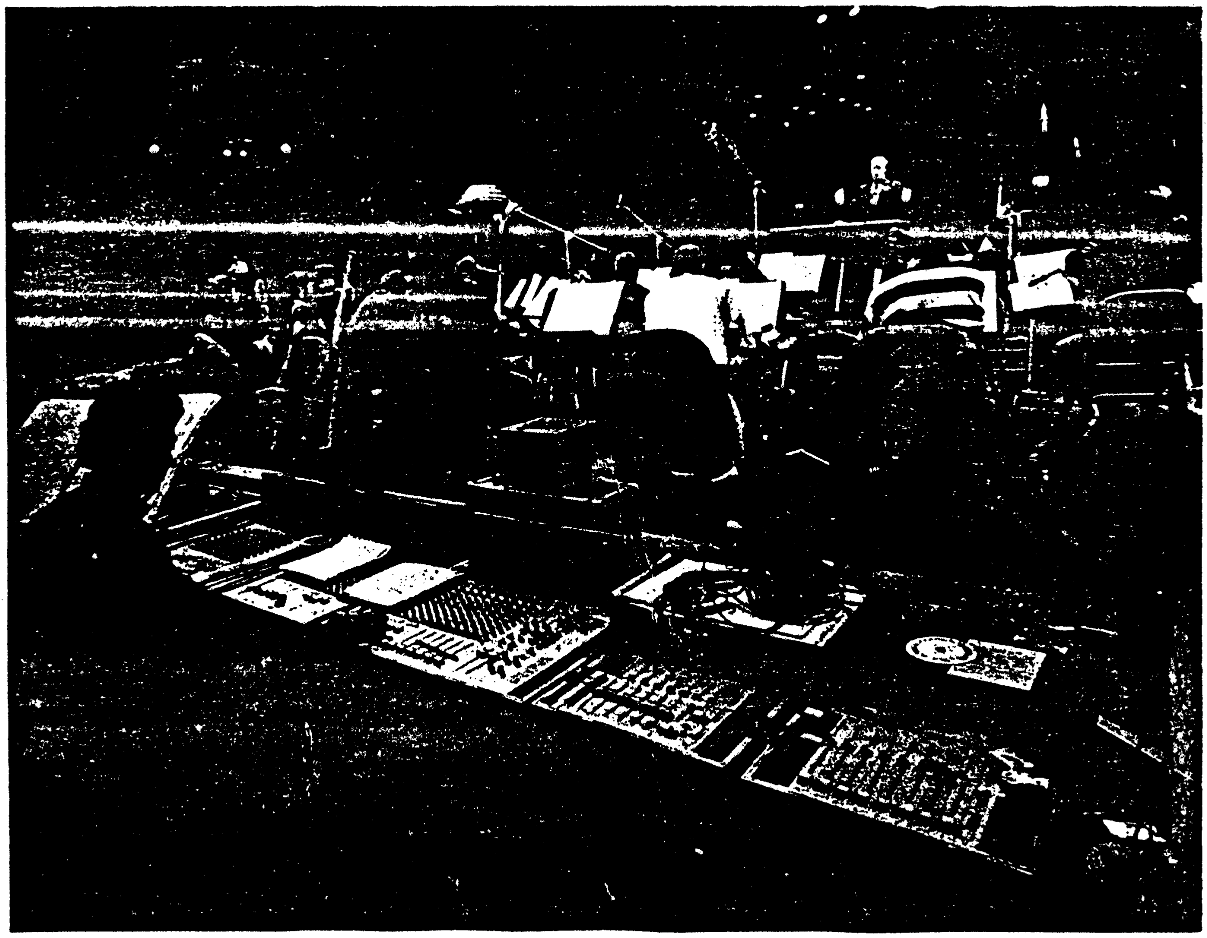
Another way to store and retrieve such huge quantities of numbers for sound-generation purposes is afforded by digital microprocessors, such as those found in computers. Converting numbers in a computer into voltages—an essential step in digital sound processing—can be accomplished easily by means of analog-to-digital converters, which translate an electrical signal into a sequence of numbers proportional to the signal's voltage, and by digital-to-analog converters, which perform the reverse process. The succession of discrete voltage pulses that a digital-to-analog converter produces from a sequence of amplitude samples is generally "smoothed" into a continuous waveform by a special filter before the electrical signal is amplified and broadcast through a loudspeaker system.

For practical reasons both computers and compact discs express amplitude samples in terms of binary numbers. The cur.ent standard is to specify a sample by means of 16 bits, or binary digits. This allows the amplitude of a sound waveform to be divided into 65,536 discrete levels. (In practice, half of these binary numbers are used to represent positive amplitude samples and half are used to represent negative amplitude samples.) This range of amplitude levels is not enough to achieve a reproduction of either rock or symphonic concerts that is absolutely free of perceptible noise, but the resultant sound fidelity is certainly very much better than analog recordings on standard phonograph discs or magnetic tapes.

Our initial efforts to apply the sampling theorem to generate sound by means of a computer were met with an unfortunate surprise: we seemed to be able to synthesize only unmusical sounds rather than beautiful musical sounds. Those early efforts consisted primarily in converting patterns of numbers representing simple waveforms (such as sinusoid or sawtooth) into sounds, and the sounds we thus produced tended to be either bland or buzzy and "electronic."

The problem was not that the computer we worked with was inherently limited in the sound waves it could produce (although there were some practical constraints in terms of money and time). The problem was rather that no one knew what constituted a waveform the human ear would perceive as a beautiful musical sound. A sound's characteristic quality is called its timbre, or tone "color." Although we were able to generate sounds of a given pitch and loudness from the computer, sounds of a pleasing timbre proved to be difficult to generate.

The modest amount of available literature on the physics of sounds of traditional instruments was not of much help. It turned out to be not only incomplete but in many cases blatantly wrong. For example, although the literature emphasized the description of a musical tone in terms of its steady state, or the middle part of its waveform, it soon became clear to us that the waveform's beginning (called the attack) and end (called the decay) were



DIGITAL SOUND PROCESSING plays an active, integral part along with the instrumentalists of the Ensemble InterContemporain in a rehearsal performance of Pierre Boulez' composition *Répons*. Technicians (*foreground*) control the real-time, or instantaneous, processing of sound from an electronic work station, taking their cues from Boulez himself (*background*), who is conducting.

more important. The contour of a waveform, drawn by connecting crest to crest and trough to trough, is called its envelope. An envelope that has an abrupt, steep attack followed by a gently sloped decay results in a sound of plucked or struck timbre, regardless of the waveform's steady-state fluctuations within its envelope.

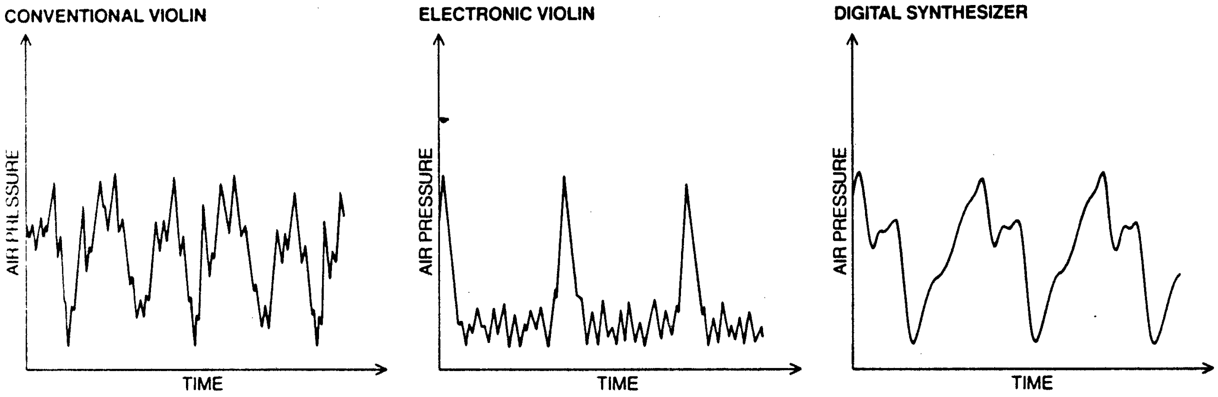
It was also thought that knowledge of the relative amplitudes of the various frequency components of a musical tone's steady state was sufficient to characterize the tone's timbre. A frequency spectrum of a musical tone shows that its waveform contains not just one component at the frequency corresponding to the pitch (called the fundamental frequency) but several components whose frequencies are generally, although not always, whole-number multiples of the fundamental frequency. These components are called partial tones, or partials for short, and can be numerically ranked according to their frequency. Hence the fundamental frequency is the first partial, the component with the next-higher frequency is the second partial and so on.

The traditional view of timbre, for instance, held that the clarinet's sound was completely determined by a spectrum in which the frequencies of all partials were odd-number multiples of the fundamental frequency. Although this property of the clarinet's frequency spectrum is to a certain extent true, we found that it is not enough to characterize the instrument's timbre.

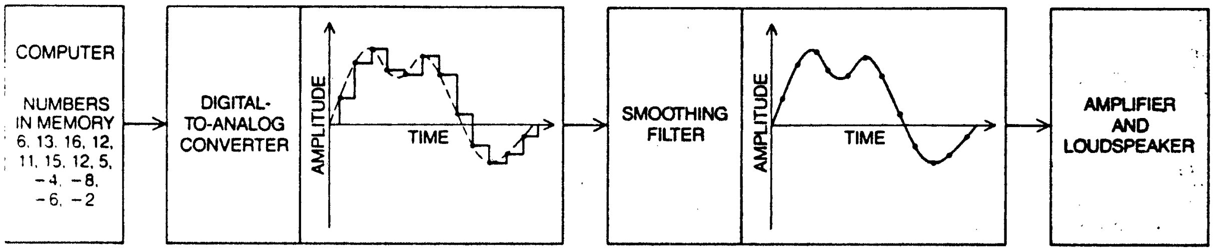
It has taken much detailed study to understand the timbral aspects of sound and to dispel some of the antiquated, misleading generalities. Fortunately the computer itself proved to be a powerful tool for studying musical timbres. A striking demonstration of how to analyze and synthesize a sound of good timbre is the early work done at Bell Laboratories in 1965 by the French composer and physicist Jean-Claude Risset.

Risset sought to synthesize good brass-instrument tones based on what he had read in the existing literature about the relative amplitudes of a trumpet's partials. He programmed a computer to generate numbers that would correspond to the amplitude samples of what he thought would be a trumpetlike waveform. When he converted the numbers into sound, however, he found that the resultant sound was not at all like that of a trumpet.

Risset then recorded real trumpet tones and analyzed their spectra with a computer. He found that a trumpet spectrum changed during the playing of the tone and that the high-frequency partials had much larger amplitudes in the middle of the tone than in the attack and decay parts of the tone. By synthesizing a sound whose high-frequency partials build up slowly during the sound's attack and reach their maximum amplitude during the tone's steady state, he succeeded in producing tones the average listener could not distinguish from recorded trumpet tones.



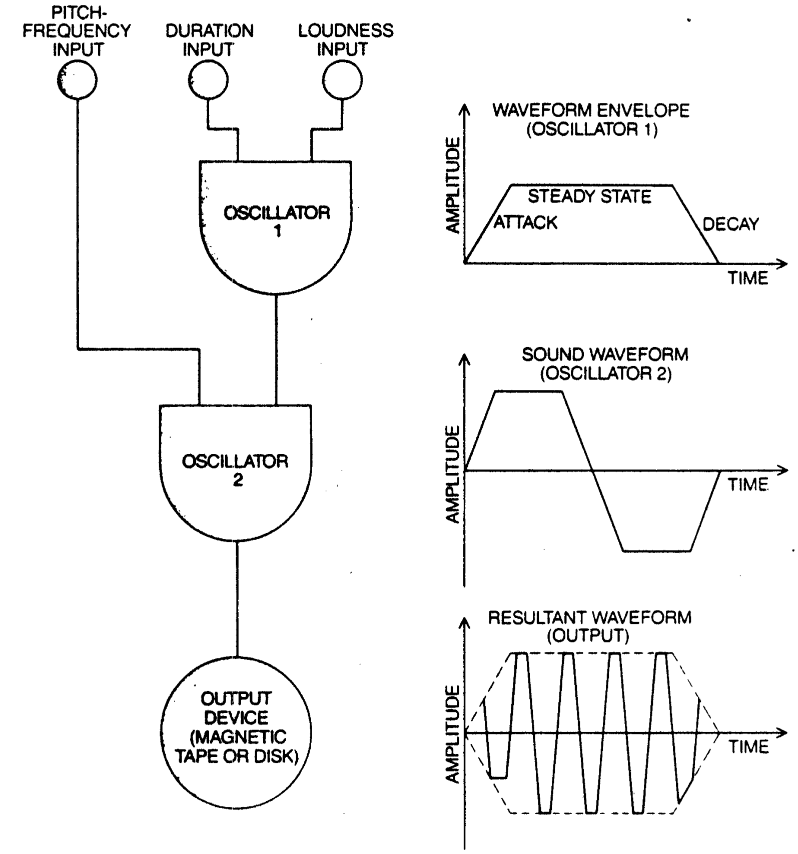
WAVEFORMS show the way a particular sound causes the ambient air pressure to fluctuate. The sound from a conventional violin (*left*) is characterized by a complex periodic waveform. An electronic violin (*middle*) built by one of the authors (Mathews) converts the motion of a bowed metal string into an electrical signal that is then filtered to yield a simple violinlike waveform. A popular digital sound synthesizer (*right*) can mimic the waveform of an actual violin sound more closely than the electronic violin can.


COMPUTER
NUMBERS IN MEMORY 6, 13, 16, 12, 11, 15, 12, 5, −4, −8, −6, −2
DIGITAL-TO-ANALOG CONVERTER
AMPLITUDE
TIME
SMOOTHING FILTER
AMPLITUDE
TIME
AMPLIFIER AND LOUDSPEAKER

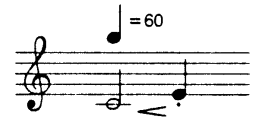
SAMPLING THEOREM, which posits that any waveform made up of components of various frequencies can be exactly described by a sequence of numbers, underlies all digital processing of sound. The numbers, called samples, are proportional to the instantaneous amplitude of the waveform, and the minimum sampling rate is twice the waveform's bandwidth (the range of component frequencies). Samples stored in a computer's memory can be converted into voltages proportional to each sample's value. The discrete voltages can then be "smoothed" into a continuous signal that is amplified and sent to a loudspeaker, where it is converted into sound. Sound synthesis involves programming a computer to produce the appropriate sequence of numbers for a given sound.

In brief, Risset discovered that the steady-state frequency spectrum is not an adequate description of a timbre. In order to synthesize a musical sound of a particular timbre one must know how the sound's frequency spectrum changes when a tone is played, that is, how the various partials build up at the beginning of the tone and how they die away at the end. Moreover, he found that different partials followed different time courses and that these were critically important to the listener. Hence a steady-state frequency spectrum shaped by a single envelope rarely suffices to reproduce the sound of conventional instruments.

Risset's work followed a methodology known as analysis by synthesis, which has led to great progress in understanding the timbre of sounds from traditional instruments. The analysis-by-synthesis methodology begins with an analysis of a sound—often by means of a computer, which can break down the sound into individual frequency components and determine each component's envelope. Such analysis usually gives a plethora of detail, from which the experimenter must select those features he believes are important in producing the sound's characteristic timbre. The investigator then formulates a hypothesis that gives a simple, physical description of the sound and tests the hypothesis by synthesizing a sound based on it. The hypothesis is evaluated by comparing the synthesized sound with the original sound. If the two are indistinguishable to the listener, the experimenter has succeeded in finding those details that effectively characterize a particular class of sounds.



STARTING TIME (SECONDS)	DURATION (SECONDS)	LOUDNESS (ARBITRARY UNITS)	PITCH FREQUENCY (HERTZ)
0.0	1.5	1	262
2.0	.25	5	325



COMPUTER "INSTRUMENT" is constructed from so-called unit generators in the sound-synthesis program Music V, written by one of the authors (Mathews). Unit generators are subprograms whose numerical inputs and outputs can be interconnected. The most important unit generator is the oscillator. Every time an oscillator is cycled it generates a series of numbers that correspond to a preselected waveform. The output waveform's amplitude and the frequency of the waveform-generating cycles are determined by the oscillator's two inputs. The amplitude input of a pitch-determining oscillator often is the output of another oscillator that controls the sound's envelope. The envelope determines the sound's attack (how quickly it builds up), its steady state (its middle part) and its decay (how quickly it fades away). An instrument thus constructed is "played" by means of note lists (*bottom left*): computer instructions that specify essentially the same information that a note on a musical staff (*bottom right*) conveys to the performer.

In order to synthesize sounds, whether for analysis or for musical purposes, one must have computer software that can efficiently generate the sequence of binary numbers representing the successive samples of a waveform. The software has to be simple to work with and yet sophisticated enough to allow the incorporation of virtually any sound-timbre hypothesis into a waveform.

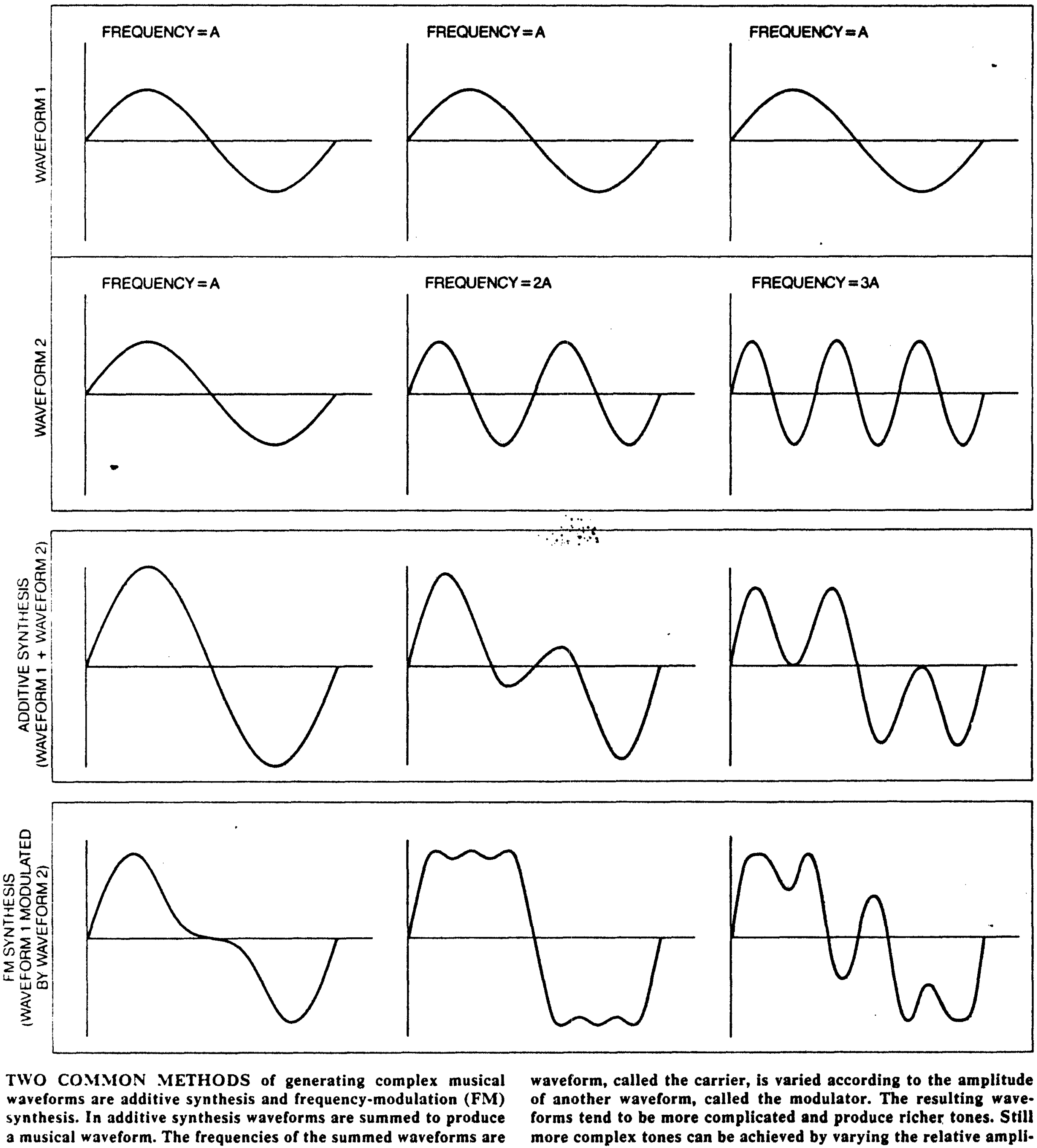
During the late 1950's and early 1960's one of us (Mathews) wrote such software. One of these computer programs, Music V, contains the basic concepts that are still to be found in the software employed in computer-music centers throughout the world. These include a number of simple program building blocks, called unit generators, to give the musician flexibility in creating unique instrumental sounds; stored tables of numbers to generate efficiently certain waveforms on command, and lists of notes (instead of the conventional music notation) to specify what is to be played.

A composer can construct his own musical "instruments" by connecting unit generators in a variety of ways. In some respects unit generators are computer simulations of the electronic devices in early analog synthesizers, which the user interconnected with patch cords. Whereas the modular devices in an analog synthesizer manipulate electric voltages, the unit generators of Music V manipulate numbers. The most important unit generators are the oscillator, the adder and the multiplier.

Every time an oscillator is cycled it generates a series of numbers that corresponds to a waveform drawn from a catalogue of possible waveforms stored in sampled form in the computer memory. Such waveforms can take on a variety of shapes: sinusoidal, square and sawtooth, among others. Stored waveforms greatly increase the efficiency of an oscillator because the computer need only look up samples of the waveform rather than having to calculate them. In addition the stored waveform determines some aspects of timbre.

Two input variables control the oscillator: one determines the amplitude of its output and the other determines the frequency of waveform-generating cycles. Both inputs can change with time, allowing a rise or fall in amplitude and frequency. The amplitude input of a sound-producing oscillator is often the output of another oscillator that functions as an envelope generator to control the sound's attack and decay. As we mentioned above, attack and decay functions also have an important influence on the timbre of a computer-generated sound.

The adder and multiplier unit generators, as their names imply, respectively calculate the sum and the product of two input numbers. The adder can sum the outputs of two sinusoidal oscillators that have been "tuned" to be the instrument's partials. It can also add to the input that specifies the frequency of an oscillator a small, regularly changing number. In this way a vibrato effect can be simulated. The



TWO COMMON METHODS of generating complex musical waveforms are additive synthesis and frequency-modulation (FM) synthesis. In additive synthesis waveforms are summed to produce a musical waveform. The frequencies of the summed waveforms are generally whole-number multiples of the pitch-determining frequency, corresponding to a musical sound's multiple frequency components, called partials. In FM synthesis the frequency of a waveform, called the carrier, is varied according to the amplitude of another waveform, called the modulator. The resulting waveforms tend to be more complicated and produce richer tones. Still more complex tones can be achieved by varying the relative amplitudes and phases of the carrier and modulator during the course of a tone. Since FM synthesis requires fewer waveforms to produce richer musical tones, it is more popular than additive synthesis.

multiplier can be used in several ways: as a volume control (by multiplying a given factor with an oscillator's amplitude-setting input) or as a pitch transposer (by multiplying a given factor with an oscillator's frequency-setting input). Unit generators that supply random numbers are also frequently used to make both noiselike sounds and small random fluctuations in the pitch and amplitude of generated tones in order to make them sound livelier and less machinelike.

The composer "creates" an instrument at the start of a Music V program by selecting a set of unit generators and specifying the interconnections among all their numerical inputs and outputs. At least one output must go to a digital-to-analog converter, which converts the binary samples into a form that can then be played through a loudspeaker. Designing an instrument is a highly creative function that is available to a composer of computer music; in contrast, a composer who relies on traditional instruments to interpret his work can expect only traditional sounds. But how does one "compose" a piece of music using such a program? It is done by means of note lists.

A note list is a computer instruction that specifies essentially the same information a note on the staff of a musical score conveys to the performer. It specifies when a note is to be played, its duration, what instrument it is to be played on, its pitch and its loudness. In addition special information that regulates the timbre of the instrument is often included. Note lists do not look like notes on musical staffs, of course; they are letters and numbers that the computer interprets as input for the sample-producing "instruments" made up of unit generators.

Several general strategies can be employed to synthesize complex sounds. Risset's work, for example, made use of a technique called additive synthesis or summation of partials. In this technique the individual partials of a given sound are synthesized separately, allowing each to have its own independent frequency and envelope, before they are added together to achieve a synthesized version of the sound. Hence by means of additive synthesis the slightly inharmonic partials (partials whose frequencies are not whole-number multiples of the fundamental frequency) of the piano or the prominent inharmonic partials of bells and drums can be closely duplicated. Additive synthesis is the most general way of synthesizing timbres.

Although additive synthesis is powerful, it is also expensive and slow. Timbres have many partials, and if each one is generated separately, a great deal of computation is required. Also, if each partial follows a different course, a great deal of control information may be needed to generate their envelopes. Many musicians have therefore sought short cuts that could generate timbres comparable to those generated by additive synthesis, but with less toil. One of the notable short cuts is frequency-modulated synthesis, invented by John M. Chowning of Stanford University. It is the technique applied in the most popular digital synthesizers today.

Frequency modulation (FM) is usually thought of as a radio-communications technique for transmitting information by modulating, or varying, the frequency of a high-frequency signal (called the carrier) with a low-frequency information signal (called the modulator). Chowning's technique relies on carriers and modulators that have either identical frequencies or frequencies of the same order of magnitude. Frequency relations between carrier and modulator of this kind are avoided in FM radio transmissions because they would fruitlessly spread the information signal over a huge bandwidth. In the case of musical waveforms this spreading of the signal's frequency spectrum can be fruitfully applied as timbral enrichment.

An FM instrument is a little more complicated to understand than an additive-synthesis instrument. At least two oscillators—a carrier oscillator and a modulation oscillator—are necessary in FM synthesis. Both oscillators usually generate simple sinusoidal waveforms, whose attack and decay shapes are controlled by envelope generators. Essentially the frequency of the carrier-wave oscillator is continuously shifted by an amount proportional to the amplitude of the waveform generated by the modulation oscillator. Hence the frequency of the carrier is no longer constant; it is the sum of the average carrier frequency and the continuously varying output of the modulation oscillator.

If the average carrier frequency and the modulation frequency are the same, it turns out that the fundamental period of the frequency-modulated wave will be the same as that of the unmodified carrier wave. What has changed, however, is the carrier's waveform. It can be shown that as the amplitude of the modulation signal is increased, the number and strength of higher-frequency harmonic partials in the carrier waveform increase.

Suppose the modulator's envelope has a flatter attack than that of the carrier. In this case the high-frequency partials will build up slowly to their steady-state amplitude values. This is just what is needed for a brass-instrument timbre. Actually the computer program we have just described produces a passable brass timbre, and it requires only two oscillators and two envelope generators. In contrast, synthesizing a comparable brass tone by means of the additive-synthesis technique typically requires 10 separate oscillators and 10 separate envelope generators.

If the frequency of the modulator is not equal to that of the carrier, the partials in the frequency-modulated signal are inharmonic. The frequency spectrum of the sound consists of a cluster of partials centered on the carrier frequency and spaced apart from one another by an amount equal to the modulation frequency. If the modulation frequency is very low, a dense set of partials forms and in general a harsh, dissonant timbre is the result. If the modulation frequency is greater than the carrier, the inharmonic partials are spread out to yield a percussive timbre.

Although FM cannot produce any arbitrarily specified frequency spectrum, Chowning showed that a great array of musically interesting sounds could be efficiently produced merely by choosing the right frequencies and the right envelopes for the carrier and modulator. Moreover, the basic FM sound can be further enriched by simply adding together several FM waveforms. The Yamaha DX-7 synthesizer, which is based on FM synthesis, has six oscillators for each of 16 simultaneous voices. These oscillators are often grouped into three carrier-modulation pairs whose resultant FM signals are then summed.

Additive and FM synthesis are early methods of producing musical sounds by means of a computer that are still applied today. More recently developed approaches employ digital hardware designed specifically for musical purposes. Several synthesizer manufacturers, for example, have exploited the dramatic drop in the cost of memory chips to store copies of actual instrumental waveforms in sampled form. For each instrumental sound the waveform's attack and a part of its steady state or decay must be stored. Tones of various pitches are produced by speeding up or slowing down the rate at which the stored waveforms are strung together. The tonal quality and duration can be adjusted by averaging and smoothly fitting together different parts of the stored waveforms.

Although digitally stored, "natural" waveforms can produce "natural" sounds, the waveforms cannot adequately capture all the loudness, pitch and timbre nuances in the phrasing of a musical passage. Hence much work is still spent on improving the "naturalness" of synthesized sounds, including those constructed from basic waveforms, such as sawtooth waves.

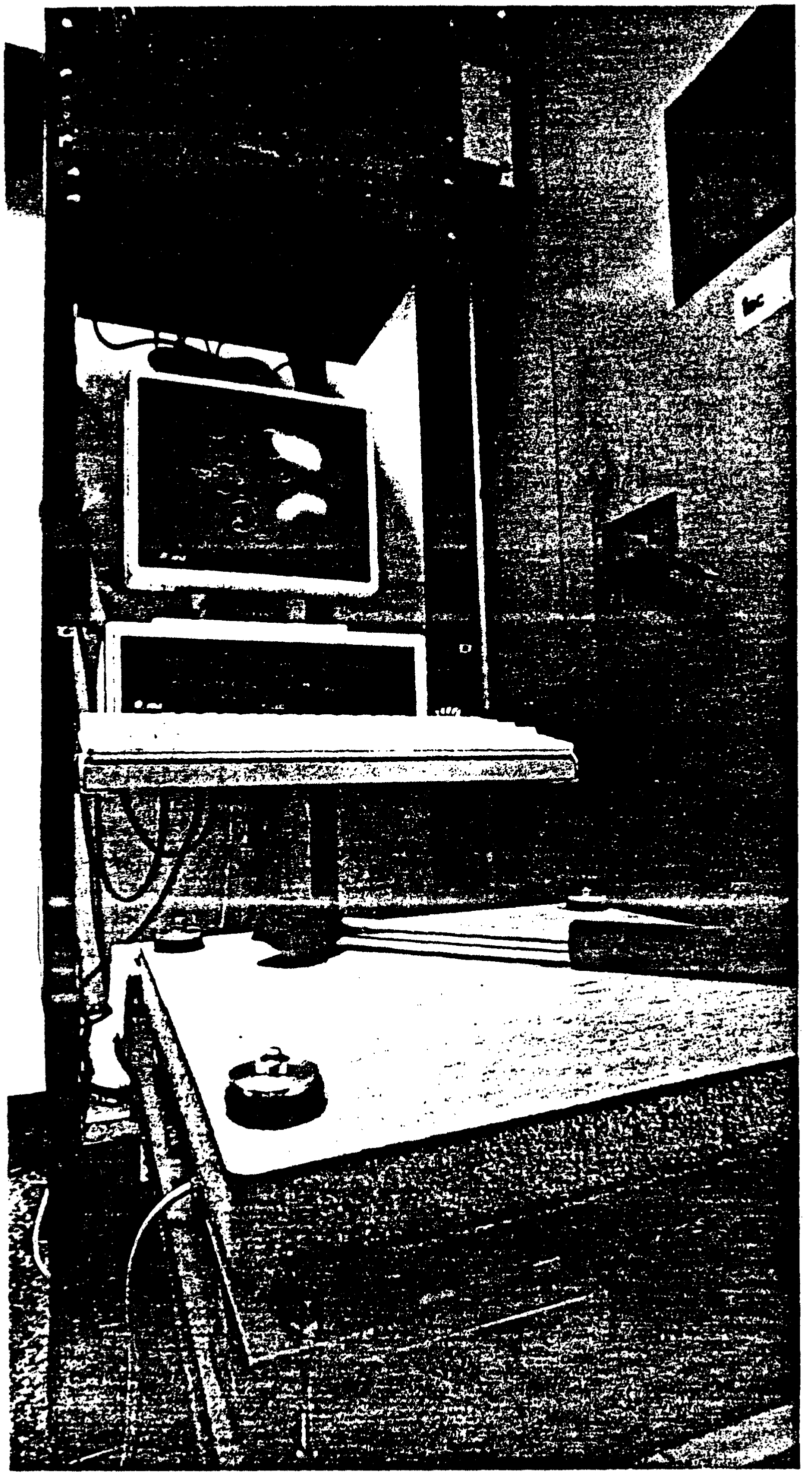
A sawtooth wave is similar to the waveform of the sound produced by bowing a violin string, yet when the sawtooth wave is played through loudspeakers, it sounds unpleasantly buzzy. The difference between a violin and a loudspeaker subjected to a sawtoothlike waveform is that the body of the violin naturally favors certain frequencies (the resonant frequencies) and damps others. Similarly, the human singing voice originates in vibrations of the vocal folds, which in themselves produce a nondescript sound. The timbre of the voice is formed almost entirely by the resonances of the vocal tract.

These principles of musical sound were understood at the beginning of computer music, but they were difficult to apply. New developments, primarily related to the cost and speed of integrated circuits, now make it possible to simulate by computer the characteristic resonances of natural sound-producing systems.

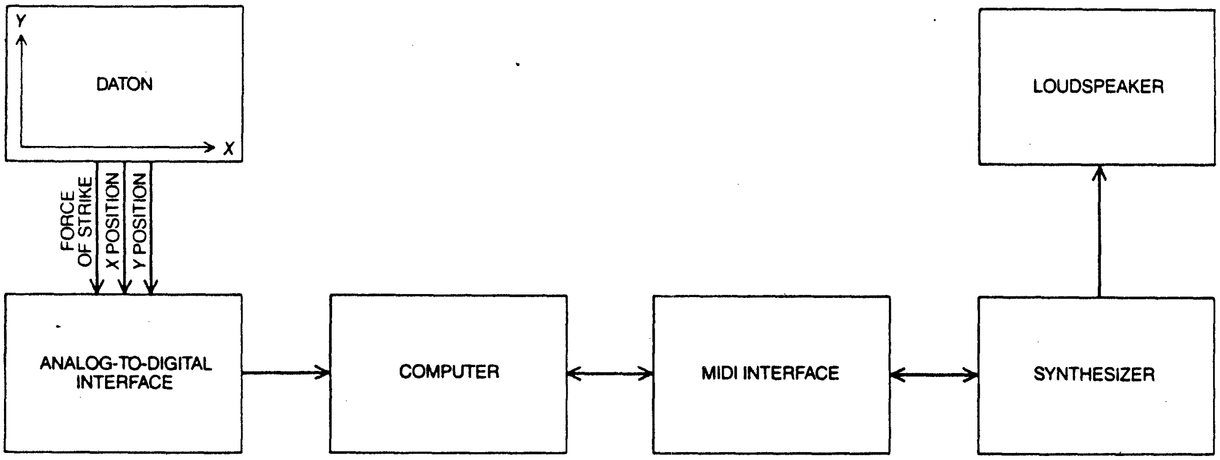
Sound synthesis based on this notion has two aspects: a general excitation, such as that provided by a sawtooth waveform, and a subsequent oscillatory decay of several major resonances. The key to getting a computer to produce a violinlike sound from a sawtooth waveform therefore lies in the simulation of the oscillatory decay of various frequency components. The computer can do this by solving linear difference equations, the discrete analogues of linear differential equations. The solution of a linear difference equation is a sum of damped sinusoids, and these can simulate the decaying oscillations of a violin soundboard or of the vocal tract.

The general problem with this type of synthesis is that it takes a lot of computing to solve a difference equation. Carver A. Mead and John C. Wawrzynek of the California Institute of Technology have recently manufactured integrated-circuit chips specifically designed for musical synthesis by the solution of linear difference equations. If such chips become available commercially, synthesis through this approach may leap forward.

Today there is no lack of strategies for the digital generation of complex and satisfying musical sounds. The chief problem with most strategies is the amount of computation nec-



"INTELLIGENT" MUSICAL INSTRUMENT in the laboratory of one of the authors (Mathews) consists of a personal computer hooked up to a custom-made sensor and a commercially available digital sound synthesizer. The instrument enables one to "conduct" a programmed piece of music: tapping with a cushioned hammer on the rectangular "daton," or drum baton (*foreground*), sets the tempo; the placement of the hammer taps controls the instrumental balance and loudness. On the computer screen above the daton a series of strikes are displayed: a circle whose radius is proportional to the force of the strike indicates where the hammer hit the daton surface. The pair of black boxes on the rack above the computer are the sound synthesizer and a digital sound-processing device.



MUSICAL-INSTRUMENT DIGITAL INTERFACE (MIDI) is a protocol for digitally coding musical data that was adopted by manufacturers of electronic instruments a few years ago. It allows a synthesizer to communicate with a computer. In the daton instrument (*see illustration on preceding page*) analog electrical signals that specify where and how hard the daton surface was hit are converted into binary numbers and passed on to a computer. The computer combines this information with the notes that have been stored in its memory and sends the combined information in MIDI form to the synthesizer, which then plays the appropriate tones.

essary to produce sounds with a lush timbre. A rough estimate is a million operations (multiplications and additions) per second of sound per instrument for a total of from 10 to 20 million operations per second of sound in a large work. The prodigious computations involved in synthesizing complex sounds have prevented the effective generation of sounds by general-purpose computers in real time—that is, without the need to record the sound samples at a low speed before they can be heard at normal speed.

Early computers did not have nearly enough capacity to synthesize music in real time. Instead composers synthesized their pieces slowly and deliberately, recording the sound samples on digital magnetic tapes and listening to the result on playback. Much fine music was created in this way, and it still may be the best way to make records and fill sound tracks, because a composer can apply a great degree of control in creating, evaluating and revising his musical scores.

Nevertheless, the recording process eliminates one kind of musician—the "live" performer. Interpretation of performance nuance must be written into the score by the composer; otherwise it will be missing. Also missing is the active pleasure of playing music, which is very important to both professional and amateur performers; listeners to such music may also regret the lack of a chance to play along vicariously with musicians at concerts.

Although general-purpose computers having enough power for real-time musical capability now exist, they are generally too expensive and cumbersome to serve as practical musical instruments in the concert hall or the home. The solution certainly lies in the production of efficient, special-purpose chips, such as those in the Yamaha synthesizers and those made in experimental form by Mead and Wawrzynek. In fact, digital musical instruments based on such chips are in many cases less expensive than some traditional acoustic instruments.

In this connection, the establishment of a standard musical-instrument digital interface (MIDI) protocol among manufacturers of commercial electronic instruments is good news for computer musicians: it allows a computer to be hooked up to such instruments, endowing them with some "intelligence." MIDI was originally intended to standardize the transmission of control information between various brands of synthesizers. Pressing a key on the keyboard of a MIDI-capable synthesizer not only causes a tone to be played but also transmits some data bits on an output cable that identify which key was pressed and how hard it was struck. A synthesizer can also have a MIDI input cable. If it receives key-play information through this cable, it will play a tone exactly as though one of its own keys had been pressed. In principle, anything that can be done on a synthesizer can be locally controlled by sensors on the machine (such as keys, buttons or knobs) or remotely controlled through MIDI.

Although the creators of MIDI may not have intended it to serve as the mode of communication between a computer and a synthesizer, it works well in this role. It certainly makes life beautiful for people who love to play with computers and synthesizers: an ordinary personal computer has plenty of power for most music-control purposes, since the waveform generation is taken care of by specialized circuits in the synthesizer itself. The essential parts of such an intelligent instrument are the synthesizer, a sensor on which the performer plays, the computer and, last but far from least, the software that ties all the components together.

Although the methods we have described for synthesizing sounds have been used to imitate traditional instruments, digital electronic equipment can just as easily create entirely new classes of sounds. More important, a sampled waveform—whether it is digitally recorded from a "real" instrument or manufactured in a computer—lends itself to easy manipulation. By means of digital sound processing a particular sound can be readily transformed into a completely different sound. For example, the frequency spectra and envelopes characterizing waveforms of human speech can be molded so that they sound like, say, a lion's roar.

Further study of musical sound by means of computers will certainly lead to more accurate imitations of traditional instruments and to the devising of means for the subtle and rapid control of their sound qualities, which is so important in actual musical performance. Also, computers have a key role in elucidating the subjective response that sounds elicit. This is particularly important for the modern composer because he or she is no longer limited to arranging sounds that can be produced from conventional instruments; it is now possible to call forth any sound that is imaginable—and even some that are not.