



Contents

Press F1 at any time for Help on using this on-line User Manual.

Introduction

Copyright and Warranty

About Art Song

Shareware

Art Song Distribution

Art Song Registration

Using Art Song

■ **Installation and De-installation**

■ **Operational Overview**

■ **Compose and Save a MIDI Sequence**

■ **Load and Play a MIDI File**

■ **Load an Image**

■ **Create and Edit a Style**

■ **Create and Edit a Trace Pattern**

Some Theory

Troubleshooting Guide

Customer Assistance



Art Song™ v2 User Manual

Welcome to **Art Song**. We hope you enjoy using this program and compose lots of original music for your various multimedia, animation, or other projects that may require unique background music.

This version of Art Song is a fully-functional, copyrighted program distributed as SHAREWARE so you may thoroughly evaluate it for your particular music compositional needs. You must register to continue to use **Art Song** past the 30 day evaluation period, or to use any of the generated sequences. Please refer to the REGISTRATION section for details on how to register and its benefits.

The **Art Song** software package consists of the software program, several types of data files, and on-line documentation including this user-manual and context-sensitive help. Data files include those for generating trace patterns, defining MIDI performance and composition parameters, defining pitch scales, and some example MIDI files. See PACKING LIST to insure that you have received the complete package.

This manual will help you learn to productively use the program as quickly as possible and then serve as a general reference for the program.

Information on specific screens, commands, buttons, etc. is provided by the on-line Help.

*Pressing the **Contents** button above will take you to the manuals Table of Contents from any point in the manual. Pressing **F1** will provide you with help using this manual.*



Copyright and Warranty Information

Copyright

The **Art Song** program, data files, and user documentation are Copyright (c) 1995,1996 by Digital Expressions. ALL RIGHTS RESERVED.

Limited Warranty on Diskette

Digital Expressions warrants to the original consumer purchaser of this product, for a period of 90 days from the date of original purchase, that the diskette is free from physical defects due to faulty materials and workmanship.

Return any defective materials for a complete replacement.

Limited Warranty on Software Program and On-line Documentation

Digital Expressions warrants to the original consumer purchaser of this product, for a period of 90 days from the date of original purchase, that the software program and on-line documentation will perform substantially as described in the user manual and packaging.

Warranty Exclusions and Limitations

USERS OF ART SONG MUST ACCEPT THIS DISCLAIMER OF WARRANTY: ART SONG IS SUPPLIED AS IS. DIGITAL EXPRESSIONS EXPRESSLY AND SPECIFICALLY DISCLAIMS ANY OTHER EXPRESS OR IMPLIED WARRANTIES WITH RESPECT TO THE DISKETTE, MANUAL, OR SOFTWARE, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, AND FITNESS FOR A PARTICULAR USE. DIGITAL EXPRESSIONS SHALL NOT IN ANY CASE BE LIABLE FOR INCIDENTAL, CONSEQUENTIAL, OR OTHER INDIRECT DAMAGE ARISING FROM THE USE OR INABILITY TO USE THIS SOFTWARE, EVEN IF DIGITAL EXPRESSIONS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC RIGHTS WHICH MAY VARY FROM STATE TO STATE.

THIS AGREEMENT SHALL BE GOVERNED BY THE LAWS OF STATE OF WISCONSIN AND SHALL INURE TO THE BENEFIT OF DIGITAL EXPRESSIONS AND ANY SUCCESSORS, ADMINISTRATORS, HEIRS, AND ASSIGNS. ANY ACTION OR PROCEEDING BROUGHT BY EITHER PARTY AGAINST THE OTHER ARISING OUT OF OR RELATED TO THIS

AGREEMENT SHALL BE BROUGHT ONLY IN A STATE OR FEDERAL COURT OF COMPETENT JURISDICTION LOCATED IN OUTAGAMIE COUNTY, WISCONSIN. THE PARTIES HEREBY CONSENT TO IN PERSONAM JURISDICTION OF SAID COURTS.

Warranty Claims

Please return any defective product to:

Digital Expressions
W6400 Firelane 8
Menasha, WI 54952

Please include your name, return address, phone number, and a brief statement of the defect.



About Art Song™

General Information

Art Song is a novel algorithmic music composition program designed to help non-musicians compose unique and original MIDI sequences for use as background music in multimedia or video projects. Generated sequences follow the General MIDI instrument naming standard and can be saved as standard MIDI (Format 1) files for compatibility with most multimedia authoring platforms and music sequencing software.

Art Song is copyrighted software but is distributed as shareware to allow you a thorough evaluation of the program for your music compositional needs. If you continue to use **Art Song** after a 30-day evaluation period you must register.

Give a copy to a friend!

The **Art Song** software package consists of the software program, data files, and on-line documentation including a user-manual and context-sensitive help. Data files include the *.IFS files used for generating trace patterns, the *.STL files which define various MIDI performance, orchestration, composition parameters, and some example MIDI files (*.MID). See Packing List.

The shareware version of **Art Song** is **fully functional** and identical to the licensed version of the program with the exception of the word UNREGISTERED in the title of the main screen. The shareware version, however, is accompanied only by sufficient data files to allow complete evaluation of the program; you will receive additional Style, Scale, and Trace Patterns data files when you register.

Operational Information

Art Song produces musical sequences by tracing a path over a picture. This picture can be any uncompressed Windows bitmap file *.BMP. **Art Song can convert your favorite pictures to music!** As a path is traced, the red, green, blue (RGB) color components of the points visited along the path are converted into MIDI note information.

By changing the picture, the trace pattern, or how colors are interpreted, an infinite number of original sequences can easily be composed.

As strange as this procedure may sound, the program tends to naturally produce music suggestive of medieval, renaissance, pre-classical, or new age styles of music. These are styles of music in which the melody of the individual parts is emphasized rather than harmonic relationships between the parts. **Art Song** can also create unique rhythmic patterns useful for many background music applications.

The key to **Art Songs** unique compositional capability resides in how the path of points is generated. This path is determined by a set of mathematical functions and a calculation process called an Iterated Function System. Due to the unique characteristics of Iterated Function Systems the overall pattern traced, as well as the pattern formed by the number of times particular points are visited, are both fractals. These fractal characteristics impart a recognizable structure to the generated music.

Of course, understanding Iterated function systems and fractals is NOT necessary for the understanding and use of **Art Song**.

Because **Art Song** produces a different type of music than random or pattern-based algorithmic composition programs, it is an excellent tool to add to your arsenal of music composition tools.



Definition of Shareware

See Also: Art Song Distribution Registration

Shareware is not a type of software but a method of software distribution. Shareware distribution gives users a chance to try software before buying it. If you try a Shareware program and continue using it, you are expected to register. Individual programs differ on details -- some request registration while others require it, some specify a maximum trial period. With registration, you get anything from the simple right to continue using the software to an updated program with printed manual.

Copyright laws apply to both shareware and commercial software, and the copyright holder retains all rights, with a few specific exceptions as stated below. Shareware authors are accomplished programmers, just like commercial authors, and the programs are of comparable quality. (In both cases, there are good programs and bad ones!) The main difference is in the method of distribution. The author specifically grants the right to copy and distribute the software, either to all and sundry or to a specific group. For example, some authors require written permission before a commercial disk vendor may copy their shareware.

Shareware is a distribution method, not a type of software. You should find software that suits your needs and pocketbook, whether it's commercial or Shareware. The Shareware system makes fitting your needs easier, because you can try before you buy. And because the overhead is low, prices are low also. Shareware has the ultimate money-back guarantee -- if you don't use the product, you don't pay for it.



Art Song™ Distribution

See Also: Shareware Registration

Art Song is a "shareware program" and is provided at no charge to the user for evaluation. Feel free to share it with your friends, but please do not give it away altered or as part of another system.

The essence of "user-supported" software is to provide personal computer users with quality software without high prices, and yet to provide incentive for programmers to continue to develop new products. If you find this program useful and find that you are using **Art Song** and continue to use **Art Song** after a reasonable trial period, you must make a registration payment of \$34.95 (+ \$4 S&H) to Digital Expressions. This registration fee will license one copy for use on any one computer at any one time. You must treat this software just like a book. An example is that this software may be used by any number of people and may be freely moved from one computer location to another, so long as there is no possibility of it being used at one location while it's being used at another. Just as a book cannot be read by two different persons at the same time.

Commercial users of **Art Song** must register and pay for their copies of **Art Song** within 30 days of first use or their license is withdrawn. Site-License arrangements may be made by contacting Digital Expressions.

Anyone distributing **Art Song** for any kind of remuneration must first contact Digital Expressions at the address below for authorization. This authorization will be automatically granted to distributors recognized by the (ASP) as adhering to its guidelines for shareware distributors, and such distributors may begin offering **Art Song** immediately (However Digital Expressions must still be advised so that the distributor can be kept up-to-date with the latest version of **Art Song**).

You are encouraged to pass a copy of **Art Song** along to your friends for evaluation. Please encourage them to register their copy if they find that they can use it. All registered users will receive a copy of the latest version of the **Art Song** system plus additional benefits.



How to Register

A registration form is provided for your convenience. Just fill it out and send it along with the registration fee to obtain a licensed copy of Art Song, additional data files, and other additional benefits.

Art Song is not a free program, it is copyrighted by Digital Expressions and distributed as Shareware so you can try the program before buying it. If you continue using the program after 30 days you are expected to register.

When you register **Art Song**, you will receive:

- A licensed copy of the latest version of the program.
- A printed copy of the User Manual.
- Additional trace pattern, style, and scale files.
- Unlimited royalty-free use of generated MIDI sequences.
- Priority technical support.
- News of future product upgrades and enhancements.
- News and information on new and related products.

Registration provides a single use license

By registering you obtain a licensed copy of **Art Song** for personal use. Registration does not give you title to the software or any copyright rights. You may not use, copy, rent, lease, sell, modify, decompile, disassemble, reverse engineer, or transfer the licensed program except as provided in this agreement; see distribution. All rights not expressly granted here are reserved to Digital Expressions.

You may:

- make as many copies of the software as you need for back-up purposes.
- use the program on more than one computer, provided there is no chance that it will be used simultaneously on more than one computer.

Site License

If you intend to use **Art Song** on multiple computers having multiple users you will require a site license. Please contact us for details.



Installation and Deinstallation

Prior to beginning installation, please make sure your hardware meets the requirements.

Installing Art Song from ARTSONG2.ZIP

- Use the **Windows File Manager** to create a new directory and drag the ARTSONG2.ZIP file into it.
- Use pkZip204 to unzip the file.
- Check the expanded contents against the packing list to ensure that the software package is complete. Installation also requires the VBRUN300.DLL which can be downloaded from the same site ARTSONG2.ZIP was downloaded.
- Use the **Windows File Manager** to run SETUP.EXE from the new directory.

Installing Art Song from the Installation Disk

- Use the **Windows File Manager** to run SETUP.EXE on the installation disk.

Deinstallation of Art Song

If you followed the recommended installation instructions all the files will be located together in one directory, except MUSICLAB.DLL which will be located in the Window System Directory.

- Use the **Windows File Manager** to delete the directory containing the program and data files.



Art Song™ Operational Overview

Please refer to the Help file for additional specific information on individual buttons or menu items.

NOTE: Key program commands are accessible as both buttons and menu items with associated Short Cuts.

Images, Paths, and Styles

Art Song uses four raw materials for composition; an image, a trace pattern, a scale or set of allowed note pitches, and a style. The program starts-up with a default image but any uncompressed bitmap (*.BMP) can be used. Symmetrical images seem to work the best.

Art Song uses Iterated Function Systems to calculate a path over the image. Different paths are represented by different IFS Codes. At points along the path, the RGB information from the image is converted into MIDI note data; Style information controls how the RGB information is converted to MIDI data.

While there are other ways to generate paths, Iterated Function Systems are used because 1) an infinite variety of paths can be generated, 2) they provide a consistent, compact, relatively simple, and visual means for representing and editing paths (the IFS Code), and 3) the paths have fractal characteristics. Fractal characteristics, specifically self-similarity, impart structure to the generated sequences analogous to the familiar theme and variations structure found in much music.

An **Art Song** Style consists of MIDI performance data, orchestration information, and values for the various parameters used by the program to convert the image color information into MIDI note data. Support for up to 10 individual instrumental parts is provided in each Style.

Because **Art Song** calculates points, displays them, converts them to MIDI music data, and outputs that data, it is a computation-intensive program and will work best on a faster computer. On slower computers you may not be able to simultaneously generate and play all 10 tracks.

For even first time users, a wide variety of sequences can be composed merely by using different pictures or selecting different trace patterns (IFS Code), scales, or styles from the lists. Advanced users can edit all aspects of the IFS Codes and the Styles.

Start Up Configuration

At startup, **Art Song** loads a default image, produces a list of available trace patterns (IFS Codes), a default scale list, and a list of available Styles from the same directory in which the program is located. These lists are placed in the Trace Pattern and Style list boxes respectively. The first item in each list is

automatically loaded.

If no data files are located in the program directory, you will need to change the directory searched for data files. Click on the list box and press space-bar, or use the **Change Directory** command from the **Trace** or **Style** menus to open a window allowing selection of a directory.

Composing and Saving a MIDI sequence

In order to compose a MIDI sequence, an image, a trace pattern, and a style must be defined. The default configuration of the program at startup should meet these criteria.

Composing a sequence is as easy as pressing the button labeled Compose located in the upper center of the Main Screen, or selecting the Compose menu item from the Song Menu.

If a MIDI output device has not already been selected, you will automatically be asked to select an output device before composition begins.

During composition you may change the pitch scale either by toggling the keys on the Piano keyboard at the right of the screen which changes the root of the scale, or by selecting an entirely different scale from the list below the keyboard.

Click again on the **Compose** button to stop composition and play. A dialog box will appear asking you whether you want to save your creation.

Loading and Playing a MIDI File

Use the **Play** button to hear the currently generated sequence or a sequence loaded by selecting the **Open...** command from the **Song** Menu. Click the button again to stop.

That's all there is to it! You can select a new trace pattern or style from the respective lists or load a new image file for an almost unlimited number of compositions.

Loading an Image

A new image can be loaded by using the **Load Image...** command from the **Style** Menu or by pressing the **Image** button at the upper left of the screen.

This concludes how to perform basic operations, the following examples provide information on performing more advanced operations.



Creating and Editing the Trace Pattern

The trace pattern is determined by an IFS Code. Creating or editing the trace essentially means creating and editing the underlying IFS Code. In order to generate a path, an IFS Code must consist of at least 2 transforms.

To create a new IFS Code:

- 1) Select the **New...** Menu item (CTRL N) from the **Trace** Menu. The image area is cleared to allow viewing and editing the transforms of the IFS Code. A small window opens providing access to the transform editing tools.
- 2) Add new transforms to the IFS Code by clicking on the **Add Transform** button. The number of transforms as well as the current selected transform is displayed above the **Add** and **Delete Transform** buttons at the top of the window.
- 3) Edit IFS Code as discussed in steps 2-4 below.

To edit the IFS Code:

- 1) Click on the **Edit** button to the right of the Trace Pattern list box, or select the **Edit** menu item from the **Trace** Menu. (CTRL F). The image area is cleared to allow viewing and editing the transforms of the IFS Code. A small window opens providing access to the transform editing tools.
- 2) Each transform in the IFS Code is represented by a box. Select one of the transforms to edit by clicking the mouse in the enclosed region.
- 3) Move the transform by clicking the mouse over one of the vertices/corners and dragging, or use the **Move** buttons in the **Tools** window. The transform can also be sized or rotated using the buttons in the Tools window.
- 4) Additional transforms can be added by clicking on the the **Add Transform** button.
- 5) Remove the currently selected transform by clicking on the **Delete Transform** button.
- 4) Click on the **Test** button to view the trace path.
- 5) After completing edits, toggle editmode off either by closing the **Tools** window or clicking on the **Edit** button.



Creating and Editing the Style

The Style contains parameters defining MIDI performance, orchestration, and composition algorithms.

To create a New Style:

- 1) Select the **New** command from the **Style** Menu. This sets all style parameters to default values, and opens the **Style** Window.
- 2) The Style window contains two major sections, a Mixer section and the Instrument settings section. The Mixer section provides control to MIDI performance of the sequence and consists of Mute buttons, volume sliders, and pan sliders for each of the 10 parts. The Instrument settings provide control over the Orchestration, MIDI Channel and Patch, as well as the parameters that control the composition algorithms.
- 3) Turn on the number of Parts that are to be used. Turn a part on/off by clicking in the left most column in the instrument settings grid. A part number is visible when that part is on.
- 4) For each part that is on set the Orchestration and Algorithm values.

To Edit a Style:

- 1) Select the **Edit** command from the **Style** Menu or click on the **Edit** Styles button at the right of the Styles list. This opens the **Style** Window.
- 2) Select a part and parameter for editing by clicking in a grid cell. Information on the various parameters is available in the on-line help for the Styles Window.
- 3) **Copy** Style parameters from one part to another by clicking on a part number and dragging it to the part to receive the copy.
- 4) **Erase** the events in a part by double-clicking on the part number.



Art Song™ User Manual Version

Manual and Software by DT Strohbeen

Manual Version 1.0 for Program Release 2.0

January 24, 1996



Packing List

The program package whether downloaded or received on diskette should include the following programs:

ARTSNG20.EXE	Program file
ARTSNG20.HLP	On-line Help file
MANUAL20.HLP	User Manual
DEFAULT.SCA	Default Scale List
README.TXT	
MUSICLAB.DLL	
SETUP.EXE	Installation program
SETUP.LST	
SETUP1.EXE	
SETUPKIT.DLL	
VER.DLL	
VBRUN300.DLL	
(Not included in ARTSONG2.ZIP)	
*.IFS	IFS Code definition
files	
*.STL	Style definition file
*.MID	Example MIDI Files

The SHAREWARE package ARTSNG20.ZIP contains only a few IFS Codes and Style definition files, sufficient for evaluation.

PLEASE NOTE: INSTALLATION FROM ARTSONG2.ZIP ALSO REQUIRES VBRUN300.DLL. IF YOU DO NOT ALREADY HAVE IT, YOU CAN DOWN LOAD IT FROM YOUR SERVICE PROVIDER.

If the package you received does not contain these files it is incomplete. Please notify us.

Registered users will receive additional STYLE, TRACE PATTERN, and SCALE data files with the licensed version of Art Song.



Hardware Requirements

In order to use **Art Song** your computer should have at least a 80486 processor, Windows 3.1, and a minimum of 4M Random Access Memory; 8M RAM is recommended. Art Song is computational intensive and will work best on faster computers.

Some type of MIDI Output device such as an internal sound card or external MIDI synthesier is required. A WaveTable synthesis device is recommended for the highest quality results.

Approximately 2M of harddisk space will be needed for installation.



Troubleshooting Guide

Here is a list of frequently asked questions and potential problems. Check this list first if you experience difficulties or something just doesn't seem right.

CheckList

- 1) Are all the necessary software components present? See [Packing List](#).
- 2) Is your computer and hardware configured correctly? See [Hardware](#).
- 3) Is the program installed correctly? See [Installation](#).

Problem:

Sequences do not play.

What to Do:

- 1) Make sure that the file MUSICLAB.DLL is located either in the Windows System directory or in the same directory as the **Art Song** program.
- 2) Verify that the volume for the parts and the selected MIDI device is not set at 0.

Problem:

The instruments don't sound right or the wrong instruments are playing.

What to Do:

- 1) Select a MIDI Output device which is General MIDI compatible.
- 2) Select the MIDI Mapper as the default MIDI Out device and create a MIDI Mapper setup.
- 3) Modify the instrument Patch settings in the Style window to meet the requirements of your MIDI device.

Art Song uses instruments based on [General Midi Instrument Voice Assignments](#) which may not be supported by your MIDI device.

Problem:

The instruments don't sound very good.

What to Do:

Most Sound Cards create sounds using FM synthesis. This means they start with generated tones and combine them to approximate the sound of real instruments. The newer Wave Table Synthesis Sound cards start with samples

of real instrument sounds and sound much better. For serious work WAVE TABLE synthesis sound cards are recommended.

Problem:

An error message when loading an image file.

What to Do:

Art Song does not support compressed bitmaps. If you try to load a compressed bitmap, you will typically get an Out of System Resources message. Most paint programs will allow you to open and then resave the image in a non-compressed bitmap format.

Problem:

Composition and playback get out of sync.

What to Do:

1) Compose fewer parts at a time, dont try to compose all 10 parts at once.

2) Dont perform other functions while composing.

During composing Art Song must perform many calculations. Sometimes if you are performing other activities your hardware cannot keep up with all the program demands.

Problem:

Mute and Instrument Volume do not appear to work.

What to Do:

1) **Art Song** uses the volume change capabilities of your output device for volume and mute (volume = 0) changes. Some older sound cards do not support volume changes.

The generated sequence should be sound normal when played.

Please let us know if you experience any other difficulties so we can either fix them.



Customer Assistance

Phone: (414) 733-6863

strohbeen@aol.com

E-mail:

See Also [Warranty](#).

Save time, before you contact us check out the [Troubleshooting](#) guide.

We will be happy to help you with any questions, ideas, or problems you have with this product, even if you are not a registered user. Of course, as a registered user you will receive priority support.

We provide product support by mail, telephone, or e-mail, whatever means is most convenient for you. However, we do request that you write or use e-mail for non-critical issues.

When contacting us please provide:

- Brief description of the problem.
- Steps or program operations that lead to occurrence of the problem.
- Operating System i.e. Windows 3.1, Windows NT, Win 95...

We may also need other types of information to fully diagnose and resolve a problem:

- Basic information about your computer and its operation such as memory, other types of programs that may be running while using **Art Song**, and types of attached devices and sound cards.

When writing or phoning please remember to include your name, phone number, return address, or e-mail address so we can respond.

Write:

Digital Expressions
W6400 Firelane 8
Menasha, WI 54952

We would also like to hear from you about how we can improve **Art Song** to better meet your needs.

Art Song Registration Form

[Print Form](#)

Send To: **Digital Expressions**
W6400 Firelane 8
Menasha, WI 54952

____ REGISTRATION

Enclosed is US \$34.95 registration fee (+ \$4 S&H in US, Canada, and Mexico, or + \$8 Air Mail elsewhere), please rush me the registered version of **Art Song** and additional data files.

Name _____

Street _____

City _____ **State** _____ **ZIP** _____

E-Mail _____ **Phone** _____

The following information is **optional** but will help us better serve you.

Where did you obtain a copy of this program? _____

Musical Interests _____

How can we make this program more useful to you? _____

Any suggestions or comments? _____



References

Some additional sources of information you might find useful.

1) **Fractals Everywhere**, Michael Barnsley, Academic Press, Inc. San Diego, 1988.

This undergraduate mathematics text book provides the most comprehensive treatment of fractal geometry and Iterated Function Systems available, by the pioneer of IFS Theory.

2) **The Science of Fractal Images**, edited by H.O. Pietgen and P.H. Richter, Springer-Verlag, Berlin, 1986.

This early classic on fractal images and their generation contains a chapter on Iterated Function Systems by Michael Barnsley.

2) Michael Gogins, Iterated Functions Systems Music, *Computer Music Journal* **15**(1): 40-48. MIT Press.

An early application of Iterated Function Systems to music composition. The article describes composition based on the Measure of the IFS attractor. I used a similar IFS-based compositional method in two earlier programs, **MusicLab IFS** and **Fractal MusicLab**, for the Amiga.

Many new popular computer-graphics books provide discussions of IFS Theory and IFS Image generation. Here are just a few:

3) **Tricks of the Graphic Gurus**, Dick Oliver et al., Sams Publishing, 1993.

4) **Fractal Creations, 2nd Edition**, Tim Wegner and Bert Taylor, Waite Group Press, Inc., 1993.

This book also includes both a version of Fractint to generate your own fractal images as well as a CD-ROM full of excellent fractal images to use for composition.



Iterated Function Systems

This section provides a basic explanation of the mathematical theory underlying **Art Song** operations. Please note: This material is NOT required to understand and use the program. It is only provided for those individuals having an interest in the underlying mathematics of the program.

Throughout the discussion, an intuitive, rather than rigorous, approach is taken. Refer to one of the [references](#) for a more rigorous mathematical treatment to Iterated Function Systems and fractals.

Iteration

Understanding iteration as applied to one function or formula is fundamental to the understanding of Iterated Function Systems. The latter basically applies iteration to sets or groups of functions.

Iteration is simply the process of repeatedly calculating the **output** of a function and then using that output as the **input** to the next calculation. This process often leads to a stable limiting value; a value which reproduces itself when it is re-input into the function. This limiting value can be a solution to the mathematical problem.

The ability to start with a guess and, by using a well-defined process, achieve a problem solution makes iteration very useful. And because it is a well-defined process it is particularly suitable for solving problems with computers.

Using iteration requires making an initial estimate or guess of the problem solution. This first value is used to compute a second value, the second value is used to calculate a third, then a fourth, and so on. The successive values of iteration are called iterates. The process can be represented symbolically by: $x_{n+1} = f(x_n)$. This equation signifies that the $n+1$ th iterate is calculated from the n th iterate. $f(x)$ can be read: function of x .

[Click here for an example of iteration](#)

Iteration is not a new technique; it has been in use for a long time. An early example is [Newtons method](#) for determining the [roots](#) of [polynomial](#) functions.

When using Newton's method, we start with an initial guess for a root, and iterate until the output approaches a stable limiting value. This limiting value will be one of the roots of the function.

An interesting question naturally arises here. When multiple problem solutions exist, such as the multiple roots of polynomials, what determines whether the initial guess leads to one particular root or another? As it turns out, there is a very complex boundary between 'initial guesses' which lead to one particular root or to another different root. This complexity continues to exist even when looking at very small differences between initial values of the guess; this boundary never becomes smooth even at very small differences.

This complex boundary is an example of a fractal. Newton's method is still often used today to generate many types of fractal images.

Function Attractors

The process of iteration generates a sequence of values which can be considered to trace out a path or orbit. These orbits can exhibit very complex behaviors, even for relatively simple functions. Currently these orbits are the objects of study in such subjects as non-linear dynamics, fractal geometry, or chaos theory.

When the path followed by the successive values of the iterate approaches a stable limiting value we call that limiting value an **attractor** of the function. Solutions to many problems, such as Newton's method for roots, rely upon the existence of these stable limiting values or attractors.

Attractors do not have to be just single points, other types have been discovered. Limit cycles, for example, have an orbit that asymptotically approaches a repeating pattern of limiting values.

Attractors can even get more interesting! Many nonlinear functions when iterated will have values that wander between limits yet never visit the same point twice. Such a function is said to have a chaotic or **strange attractor**. Strange attractors have fractal dimensions because they fill a definite fraction, but not all of their respective function spaces. When the set of points of such a strange attractor is 'self-similar' under magnification, i.e. appearance does not change appreciably when magnified, it is called a fractal.

Fractals

Fractals are sets of points that have a similar appearance under different arbitrary levels of magnification. Because of the continuous detail at smaller and smaller scales, fractals can be said to be 'infinitely' detailed.

Coastlines and treelines are often cited examples of naturally occurring fractals.

We will use the 'treeline to help demonstrate the concept of 'infinite detail': From a sufficient distance, we see a treeline as a continuous 'corrugated' curve. However as we approach the treeline, we begin to see the individual outlines of trees, then individual branches, and so on. If we had been measuring the distance between two points along the original treeline, we would have found that the apparent distance increased as we approached the trees and had to measure around the individual trees, then the individual branches, then the leaves, etc. Since our treeline continues to exhibit additional detail at greater magnifications, we can imagine this to continue to occur at infinitely large magnifications hence infinite detail. (Of course natural objects do have a real limiting resolution or granularity).

Contrast the above example of a treeline with a similar experiment using a classical geometric object. For example, imagine measuring the distance around a circle, i.e. its circumference. Now imagine putting the circle and the ruler under a 'microscope' and measuring the circumference again.

Regardless of how much we magnify the edge of the circle, the measured circumference will remain the same length. Standard geometric objects are smooth at all magnification levels. This differs from fractal objects which continue to exhibit additional details at each increased magnification level.

Although fractals are complex shapes, they can often be produced by very simple processes. We saw one example, Newton's method, above.

Self-Similarity

Shapes containing parts which are small replicas of the overall shape are said to be self-similar. If the parts are exact replicas of the whole, then the shape is called strictly self-similar. Self-similar objects are usually fractals. However, not all fractals are self-similar.

The Sierpinski triangle, shown below as Figure 1, is an example of a strictly self-similar fractal. You will note that the triangle appears to be formed of three similar triangles, each of which in turn appears to be formed of three smaller similar triangles, etc. If you magnify smaller portions of the triangle, the basic 'shape' of the image will remain constant.



Figure 1 Sierpinski Triangle

It is this complex hierarchical structure of self-similar fractals that makes them interesting for music composition. Music theory teaches us that unity and coherence in music compositions is achieved by repetition and development of a smaller number of musical motifs or themes. The 'self-similar' structure of fractals appears to inherently have some of these same qualities.

Iterated Function Systems

Groups or systems of functions can be iterated as well as single functions. The principle difference between iterating one function or multiple functions is the need to select one of the multiple functions for each iteration. The selected function is then used to compute the iterate.

Although iterated function systems could include any type of function, the most widely studied systems are comprised of affine transforms. The term 'Iterated Function System' typically refers to systems of affine transforms, usually between 2 - 20, one of which is randomly chosen for each iteration.

For each iteration, one of the transforms in the system is randomly selected and used to calculate the output for that iteration. This process almost always produces self-similar fractal attractors.

Art Song provides two automatic means for assigning probabilities to the transforms for random selection during iteration. Probabilities can be 1) uniform, each transform has the same selection probability, or 2) 'determinant relative'.

For determinant relative probabilities, the probability assigned to each transform is proportional to the size of its determinant relative to the sum of all the determinants.

The image produced by iteration of an Iterated Function System is the attractor of the Iterated Function System.

IFS Code

An important requirement of Iterated Function Systems is that, on the average, the distance between successively calculated points decreases. When this convergence criteria is met we call the set of coefficients for all the transforms an **IFS Code**.

In two dimensions, the IFS Code consists of seven (7) coefficients for each affine transform. Six (6) coefficients represent the actual transform. The seventh coefficient corresponds to that transform's probability of being chosen relative to the other transforms. **Art Song** displays the values of these seven coefficients for the selected transform at the bottom of the display when editing the trace pattern.

While calculating the orbit of the IFS, it can be expected that some points will be 'hit' or visited more often than other points. The **measure** reflects this difference. When rendering an IFS image the measure is often used to color the particular points. Different colors are chosen to represent the number of times points are 'hit' during computation. In **Art Song** this measure reflects the recurrence of particular musical patterns or motifs.

The structure or overall image of the attractor is controlled by the set of the first six transform coefficients; the measure is controlled by the probabilities.

Despite the use of random iteration, theory indicates that for each IFS Code there is a unique associated attractor-image and a unique associated measure on the attractor.

MIDI (**M**usical **I**nstrument **D**igital **I**nterface) was developed in the early 1980s as a standard means for computers to communicate with electronic musical instruments. It has evolved into a means to link virtually all of the equipment used in music and video production. Standard MIDI files consist of MIDI information stored on disk.

General MIDI is a standard list of 128 different musical instruments. Using this standard allows sequence and device compatibility for selection of the most appropriate musical instrument. Before General MIDI different MIDI devices implemented their own individual instrument lists; musical sequences could sound drastically different when played on different devices.

RGB (**R**ed, **G**reen, **B**lue) represents the values for the individual components of a particular color.

Roots of Functions

For polynomial functions in one variable $f(x)$, the 'roots' are values of the variable 'x' which reduce the value of the function to zero, i.e. $f(x) = 0$.

For example the roots of $f(x) = x^2 + x - 6$ are $x = 2 \dots (2)^2 + (2) - 6 = 0$
and $x = -3 \dots (-3)^2 + (-3) - 6 = 0$.

Quadratic Function

A polynomial function of degree 2. The standard form of a quadratic function in one variable x is $f(x) = A_0 + A_1x + A_2x^2$. For example: $f(x) = 4 + 9x + 5x^2$

Polynomial Functions

An algebraic function of two or more summed terms, each term consisting of a constant multiplier and one or more variables raised generally to integral powers.

The standard form of a polynomial of degree n in one variable x is:

The A 's are constants.

Newton's Method

Assume we want to find the values of x that satisfy some polynomial function $f(x) = 0$, i.e the roots of the polynomial. Newton's method proceeds by making a rough graph of the function $y = f(x)$. We estimate the root as the point where the graph crosses the x -axis. This estimate is our first approximation, call it x_1 .

The next approximation x_2 is where the tangent to the curve at x_1 crosses the x -axis. x_3 is where the tangent to the curve at x_2 crosses the x -axis, etc. As the process continues, the values of x should continue to approach one of the roots of the function.

Tangent

The tangent to a curve is a line that touches the curve at a single point, but does not intersect (cross) the curve.

Iterating the Logistic Equation

The logistic equation is an important equation in population dynamics. It represents the population 'x' in the (n+1)th year as a function of the population in year n, i.e. where R is a constant:

Starting with an initial value of $x_1 = 0.1$ and iterating the logistic function with $R = 2$, the following series is obtained: (* = multiplication operator)



The sequence of iterate values, 0.1, 0.18, 0.295... 0.500, trace out a path or orbit. The stable limiting value, 0.5, for this particular example represents a fixed point attractor of the function.

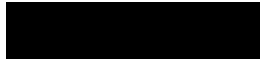
Affine Transforms

Affine transforms are functions comprised of a rotation, a scaling, and a translation. In two dimensions, the x-y plane, affine transforms can be written as:

A large black rectangular box redacting the equation that defines an affine transform in two dimensions.

The r 's are scaling factors, the θ 's are rotation angles, and the constants are translations.

The above equation is typically simplified to the following form:

A black rectangular box redacting the simplified form of the affine transform equation.

with each transform being typically represented by the six coefficients A,B,C,D,E,and F.

The determinant of the affine transform is equal to: $AD - BC$.

