The Rhythm of Hailstones

New Mexico Adventures in
Supercomputing Challenge
Final Report
April 6, 2005

Team 082
Silver High School

Team Members:
Sasha Ogas
David Potts

Teacher:
Mrs. Peggy Larisch

Project Mentors:
Mr. Berry Estes
Mrs. Roxanne Ogas

Acknowledgments

The authors would like to thank the following persons for their help and support throughout this project. They have been of great assistance and guidance. They are greatly appreciated:

- Mrs. Peggy Larisch-Teacher, Silver High School, Advanced Computer studies
- Mr. Berry F. Estes-Nuclear Engineer (Retired), Sandia National Laboratories
- Mrs. Roxanne Ogas- Coach, Silver High School, Silver Sensations Dance Team

Table of Contents

E.0 Executive Summary

The Rhythm of Hailstones is designed to create different and entertaining dance routines. This project uses the "Hailstone Theory," consisting of two equations, created by Lothar Collatz. If the initial number used in the theory is even, it is divided by 2. If the number is odd, it is multiplied by 3, then 1 is added to the product. The sequence terminates at 1 because of the division that occurs during the "Hailstone Theory". Mike Keith first applied the "Hailstone Theory" to music using the theory to compose new pieces of music. Team 082 hypothesized that the "Hailstone Theory" could be applied to dance because music and dance are similar. They both go by counts and rhythms.

Many times a choreographer is not given enough time to create a fresh dance routine or develops a type of "writer's block", causing large amounts of pressure for the choreographer. Choreographers do not have many technological resources. This project will correct this problem and allow the choreographer a way to create a new dance in a matter of minutes.

The computer program runs through a series of steps to output the final product. Initially, a number between 50 and 100 is entered into the program and is run through the "Hailstone Theory". The log of each number outputted in the theory's sequence is taken and then multiplied by the number in the logarithmic equation. The user then opens "Windows Movie Maker" and rearranges the video clips of each eight count in the dance in the order of the sequence the computer program has created.

The team has made 204 dances by using The Rhythm of Hailstones. This gives choreographers a computer program that will build new and exciting dances right away. The team members have used the Rhythm of Hailstones programs, and the project has proven that the programs as developed by the team can be applied to the choreographing of new dances involving unique rhythms.

1.0 Introduction

1.1 Purpose

The purpose of The Rhythm of Hailstones is to help choreographers create dances when they no longer have any ideas for dance routines. Using the “Hailstone Theory” and logarithms, this project will rearrange the eight counts from a dance and create many more dances. The “Hailstone Theory“ is a mathematical theory, which uses two equations to create a random sequence of numbers that terminates at one. Windows Movie Maker” is used to build a video of the dance to assist the viewer on what each move looks like.

1.2 Background Research

Dance is an art form that has become common throughout the years [1]. Even though there are many types of dance that exist today, more are yet to become popular as well. Some people say dance itself can work magic [2]. There is not a limit as to the different types of dances performed, or even the amount of dancing that occurs on a daily basis. Everyone dances. Dancing can be a baby moving around in it’s crib, a couple competing in a ballroom dancing competition, or even a shy teenager dancing alone in their room [3]. Dancing is not only about the movements occurring, but also about the music. Music contains certain accents. The choreographer, the creator of the dance, often compared to the director of a play, is in charge of choosing what movements to place when those accents occur in the music [4]. If the dance does not go along with the music, then the overall effect the choreographer is trying to portray to the audience is lost [5]. The choreographer must undergo a large amount of pressure. If the dance does not flow correctly, does not go with the music, or does not use imagination and creativity, the choreographer has failed [5].

There are still many more directions where dance can be explored. Choreographers continue to search for new ways to make the audience enthusiastic about watching their performances and their creations. As time continues, many choreographers run out of new and exciting ideas [6].

There must be something done to help choreographers overcome a type of “writer’s block”. The function of this project is to assist choreographers in creating a dance. After placing one dance into the program, the output will be phenomenal because it is a breakthrough in dance technology. The choreographer will be able to have numerous dances created by simply using one set of eight counts in the computer program making it easier to design an innovative dance.

1.3 Scope

The user selects one of four different computer programs created for the four types of dances previously selected to use as examples for the numerous dances in the world. The user inputs a number between 50 and 100 into the computer program to begin the process. The computer program uses the “Hailstone Theory” to establish whether the number is even, or whether it is odd. If the program finds the number to be even, the number is divided by 2. If the number is determined as odd, the program then multiplies the number by 3, and then has 1 added to the product. A sequence of numbers will continue until reaching 1and will always end at 1 [7]. In that sequence, each number corresponds to an 8/4 count video clip. After the sequence is displayed, the user will then use “Windows Movie Maker” to create a visual aide of the newly created dance routine.

For example, the user wants to create a Jazz routine; the user opens the Jazz computer program. The user next enters 64, which is within the 50 to 100 range. Since 64 is an even number, it is divided by 2. The result is 32. The process continues until it reaches 1. Then, the program takes log of each number in the sequence. Once the log is found, the choreographer views the sequence and places the eight-count video clips, which

corresponds to the data obtained from the C++ Program. The video clips are viewed in “Windows Movie Maker”. The user has just created a visual aide to assist them on learning the new dance routine

1.4 Computer Program

Members of the team are using a C++ computer program to create up to fifty separate dance variations based on the primary dance. With the assistance of logarithms, the previously created music-based “Hailstone Theory” has been used to rearrange the sets of 8/4 counts in dance routines. In order to see the results, “Windows Movie Maker” is included in the project to show the user what the dance actually looks like after it has been run through the computer program.

2.0 Project Proposal

2.1 Description of Project

This computer program shall use the ideas of a choreographer to create more ideas for that same person. The program will use formulas (discussed in Section 3.0) to determine which actions to take, and then randomly organize the clips from a dance. This will help ease the stress from the choreographer and allow for further advancement in the world of dance.

3.0 Analytical Methodology

3.1 Mathematical Bases

The “Hailstone Theory” is based on the following equations:

$$\begin{cases} n/2 & \text{if n is even} \\ 3(n)+1 & \text{if n is odd} \end{cases}$$

Lothar Collatz created the “Hailstone Theory”, also known as the Collatz Problem, 3n+1 problem, and the Syracuse Problem, in 1939 [7]. When a number is run through these equations, the output is a sequence of numbers that are in random order from zero to the number of eight-counts of that specific dance routine, eventually reaching one. When the sequence of numbers arrives at one, the program stops the sequence.

Along with the “Hailstone Theory”, the programs use a logarithmic function for each of the four specific dances. A logarithmic function [8] is used in The Rhythm of Hailstones to simplify the mathematical calculations. The program uses a $\log_{10}$ function. The number in front of each logarithmic function is there so the numbers in the sequence do not exceed the number of eight counts for that specific dance routine. It does this by multiplying the log by a number where the highest log output multiplied by the number before it equals the number of the last eight count of the dance. For example, if there are 38 eight counts in the dance, the choreographer does not want the program to output 44 as one of the numbers.

The logarithmic functions used in the programs are as follows:

- Jazz Program (Appendix A): $(6)*\log_{10}(n)$
- Hip Hop Program (Appendix B): $(8)*\log_{10}(n)$
- Mix Program (Appendix C): $(9.5)*\log_{10}(n)$
- Pom Program (Appendix D): $(9)*\log_{10}(n)$

Where *n* is the integer entered by the user.

3.2 Computer Applications

All four programs were created and written by Team 082. The four Rhythm of Hailstones programs are similar in structure with different outputs. The programs all include the stdlib.h, math.h, and iosphere.h source files. All four programs allow the user to enter a starting number between 50 and 100. This range of numbers was chosen because the sequences that are produced work well with the number of eight-

counts and the length of the music. The starting number is run through the formulas in a loop. Then the numbers outputted are run through the specific logarithmic function of that program. As a result, the program outputs a list of numbers that increase and decrease in a manner similar to hailstones, for example, 19, 16, 20, 18, 15, 19, 17, 14, 12, 10, 7, 5, 2, 0 (Note: $\log_{10}$ of 1 is 0). This example from the Hip Hop program used the starting number 75. The first step requires 75 to be multiplied by 3 and have 1 added to that number because the starting number of this example is odd. The final result of this equation is 226 and because 226 is even it is next divided by 2. This continues until the sequence reaches a value of 1. Then each of those numbers is run through the logarithmic function for the Hip Hop program ($(8)*\log_{10}(n)$). That is how the smaller numbers are produced. Each number in the sequence corresponds to a video clip issued to a specific eight count. If the user entered 75 in all four programs, the output sequence would be different for each program. This happens because each program has a specific logarithmic function where the number in front of the logarithmic function is different for each dance routine. The numbers differ because each dance has a different number of eight counts.

The logic for the Rhythm of Hailstones programs is simple:

- The user enters a beginning integer.
- If the integer is odd, then it is multiplied by three and adds one.
- If the integer is even, then two divides it.
- The for() loop has this process continue until the number hits one.
- Those numbers outputted are run through a logarithm.
- The program outputs those integers in a list.
- The user opens Windows Movie Maker (where the video clips are located) and put the video clips in the order of the outputted sequence.

4.0 Results

4.1 Computer Calculations

The computer program uses the “Hailstone Theory” in order to successfully create a different dance routine each time the program is run for each specific type of dance (jazz, pom, hip hop, mix) selected by Team 082. The user enters a number between 50 and 100 into the computer program, and then it is run through the ‘Hailstone Theory” (see Section 3.0). Next, a sequence is outputted and placed into the logarithmic formula (see Section 3.0) formulated by Team 082. Finally, the user is able to view the ending sequence of numbers produced by the computer program. The sequence of numbers is generated after a number is run through the “Hailstone Theory” by the computer program. (For an example see Section 3.0)

4.2 Graphs, Tables and Figures

A graph (See Appendix E), is used for the viewer to have a visual aide of what one of the sequences outputted by the computer program may look like. It is evident that an odd number is multiplied by 3 then has 1 added to the product when the sequence increases. When the sequence decreases, it is because the number is even and is divided by two. The sequence eventually terminates at 1 showing the observer that that serves as the stopping point for both the sequences and the dance routines. The sequence always stops at 1 because no matter what, the second to last number when first run through the “Hailstone Theory” is 2. The log of 2 is 1. The log of 1 is 0. That is the furthest it will go.

5.0 Conclusions

5.1 Mathematical Models

The mathematical models used in the computer program were crucial to the function of the project. They are crucial because they create the dance routines. The “Hailstone Theory” creates the random sequence of numbers that are then placed into the logarithmic function, making the numbers to end in the correct range not exceeding the number of eight counts in the original dance. No matter what number within the specified range is inputted, all the numbers in the output sequence are equal to or less than the total number of eight counts in the original dance routine.

5.2 Computer Program

The computer program is a success because it generates the output needed for this project. Team 082 is pleased to have the program working correctly and as planned. There are 51 different dance routines that can be created for each of the four dance programs in The Rhythm of Hailstones. A total of 204 different dance routines are produced using this project.

5.3 Results

The project created adds more technology to dance by employing computers in an innovative manner. The choreographer is able to turn on the computer and open a computer program that will build a fresh dance routine for them in no time. The dance routine outputted is guaranteed to be different for each initial number used in the computer program. This happens because the mathematical models used in the project output a unique sequence for each starting number entered. Choreographers can use this computer program to help construct different dance routines in a short amount of time.

5.4 Word Processor Program

Microsoft Word Processor is the program that was used to write the final report for The Rhythm of Hailstones. It kept everything in a nice fashion.

6.0 Future Work

More time being permitted, Team 082 believes The Rhythm of Hailstones computer program can be advanced by finding a way to link “Windows Movie Maker” to The Rhythm of Hailstones computer programs. By doing so, the program would automatically arrange the actual video clips of each eight count in the order of the outputted sequence of numbers. The user would be able to view the video on the spot instead of going through and creating the video themselves.

References

[1] Dance. Farlex. October 29, 2004 <http://www.thefreedictionary.com/dance>.

[2] Van Zandt, Eleanor. The Arts: Dance. Austin: Steck-Vaugh Library, 1988.

[3] Hayes, Elizabeth. Dance Composition and Production for High Schools and Colleges. New York: The Ronald Press Company, 1955.

[4] Sorell, Walter. The Dance Has Many Faces. Chicago: a capella books, 1992.

[5] McGreevy-Nichols, Susan, and Scheff Helene. Building Dances. Human Kinetics, 1995.

[6] Louis, Murray. Inside Dance. New York: St. Martin's Press, 1980.

[7] Keith, Mike. The Music of Hailstones. October 1998. September 16, 2004 <http://users.aol.com/s6sj7gt/hailmus.htm>.

[8] Math Skills Review: Logarithms. September 27, 2004 <http://www.chem.tamu.edu/class/fyp/mathrev/mr-log.html>.

Appendix A-Jazz Dance

```
#include <iostream.h>
#include <stdlib.h>
#include <math.h>

/*The Rhythm of Hailstones: Jazz. This program uses logarithms and the Hailstone
Theory to output a sequence of numbers. Each number corresponds to a video clip
for the Jazz dance routine. There are 51 different dance routines that can be
created by this program.*/

/*This is the main function of the program. It includes the Hailstone Theory
formulas and the logarithmic function.*/
//function@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
void hailstone()
{
int a;//starting number
cout<<"Enter a number between 50 and 100."<<endl;//user inputs starting number
cin>>a;
if(a<50 || a>100)//limits range
{
cout<<"The number you entered is not in the 50-100 range"<<endl;
hailstone();
}
for(int count = 0; a!=1; count++)
 {
 if(a%2==0)//checks if number is even
 {
//if number is even
a=(a/2);
 }
 else
 {
//if number is odd
a=(3*a+1);
 }
cout.setf(ios::fixed);
cout.precision(0);//limits output numbers to 0 decimal places
double f=((6)*log10(a));//logarithmic function
cout <<f<<"   ";//final output sequence
}
cout<<endl;
```

```
cout<<endl;
cout<<"Now you can match the numbers in the sequence above to the Jazz video clips
in"<<endl;
cout<<"Windows Movie Maker."<<endl;
cout<<"Then put them in the order of the sequence above and add music."<<endl;
cout<<endl;
cout<<endl;
char answer;
cout<<"Do you want to create another dance routine? (Y/N)"<<endl;
cin>> answer;
if((answer=='y')||(answer=='Y'))
hailstone();
}
//function#########################################################################
###
int main()
{
cout << "\t  !!!!!!!!!!!!!!!!!!!!  !!!!!!         !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"<<endl;
cout << "\t           !!          !!    !!                !!           !!"<<endl;
cout << "\t           !!         !!      !!              !!           !!"<<endl;
cout << "\t           !!        !!        !!             !!           !!"<<endl;
cout << "\t           !!       !!          !!            !!           !!"<<endl;
cout << "\t           !!      !!!!!!!!!!!!!!!          !!           !!"<<endl;
cout << "\t           !!     !!             !!        !!           !!"<<endl;
cout << "\t!!         !!    !!               !!      !!           !!"<<endl;
cout << "\t!!         !!   !!                 !!    !!           !!"<<endl;
cout << "\t!!         !!  !!                   !!  !!           !!"<<endl;
cout << "\t!!!!!!!!!!!!!!! !!                    !!!!!!!!!!!!!!! !!!!!!!!!!!!!!"<<endl;
cout <<endl;
cout <<endl;
hailstone();
     system("PAUSE");
     return 0;
}
```

Appendix B- Hip Hop Dance

```
#include <iostream.h>
#include <stdlib.h>
#include <math.h>

/*The Rhythm of Hailstones: Hip Hop. This program uses logarithms and the Hailstone
Theory to output a sequence of numbers. Each number corresponds to a video clip
for the Hip Hop dance routine. There are 51 different dance routines that can be
created by this program.*/
```

```
/*This is the main function of the program. It includes the Hailstone Theory
formulas and the logarithmic function.*/
//function@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@
void hailstone()
{
int a;//starting number
cout<<"Enter a number between 50 and 100."<<endl;//user inputs starting number
cin>>a;
if(a<50 || a>100)//limits range
{
cout<<"The number you entered is not in the 50-100 range"<<endl;
hailstone();
}
for(int count = 0; a!=1; count++)
 {
 if(a%2==0)//checks if the number is even
 {
//if the number is even
a=(a/2);
 }
 else
 {
//if the number is odd
a=(3*a+1);
 }
cout.setf(ios::fixed);
cout.precision(0);//limits output numbers to 0 decimal places
double f=((8)*log10(a));//logarithmic function
cout <<f<<"   ";//final output sequence
}
cout<<endl;
cout<<endl;
cout<<"Now you can match the numbers in the sequence above to the Hip Hop video
clips"<<endl;
cout<<"in Windows Movie Maker."<<endl;
cout<<"Then put them in the order of the sequence above and add music."<<endl;
cout<<endl;
cout<<endl;
char answer;
cout<<"Do you want to create another dance routine? (Y/N)"<<endl;
cin>> answer;
if((answer=='y')||(answer=='Y'))
hailstone();
}
```

```
//function#########################################################################
###
int main()
{
cout<<"\t\t\t      @@   @@ @@ @@@@@@@   "<<endl;
cout<<"\t\t\t      @@   @@ @@ @@        @@   "<<endl;
cout<<"\t\t\t      @@   @@ @@ @@        @@   "<<endl;
cout<<"\t\t\t      @@   @@ @@ @@        @@   "<<endl;
cout<<"\t\t\t      @@@@@ @@ @@@@@@@ "<<endl;
cout<<"\t\t\t      @@   @@ @@ @@        "<<endl;
cout<<"\t\t\t      @@   @@ @@ @@        "<<endl;
cout<<"\t\t\t      @@   @@ @@ @@        "<<endl;
cout<<"\t\t\t      @@   @@ @@ @@        "<<endl;
cout<<endl;
cout<<"\t\t\t  @@   @@ @@@@@@ @@@@@@@"<<endl;
cout<<"\t\t\t  @@   @@ @@    @@ @@       @@"<<endl;
cout<<"\t\t\t  @@   @@ @@    @@ @@       @@"<<endl;
cout<<"\t\t\t  @@   @@ @@    @@ @@       @@"<<endl;
cout<<"\t\t\t  @@@@@ @@    @@ @@@@@@@"<<endl;
cout<<"\t\t\t  @@   @@ @@    @@ @@"<<endl;
cout<<"\t\t\t  @@   @@ @@    @@ @@"<<endl;
cout<<"\t\t\t  @@   @@ @@    @@ @@"<<endl;
cout<<"\t\t\t  @@   @@ @@@@@@@ @@  @@"<<endl;
cout<<endl;
cout<<endl;
hailstone();
     system("PAUSE");
     return 0;
}
```

Appendix C- Mix Dance

```
#include <iostream.h>
#include <stdlib.h>
#include <math.h>
```

/*The Rhythm of Hailstones: Mix. This program uses logarithms and the Hailstone Theory to output a sequence of numbers. Each number corresponds to a video clip for the Mix dance routine. There are 51 different dance routines that can be created by this program.*/

/*This is the main function of the program. It includes the Hailstone Theory formulas and the logarithmic function.*/

```
//function@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
void hailstone()
{
int a;//starting number
cout<<"Enter a number between 50 and 100."<<endl;//user enters starting number
cin>>a;
if(a<50 || a>100)//limits range
{
cout<<"The number you entered is not in the 50-100 range"<<endl;
hailstone();
}
for(int count = 0; a!=1; count++)
 {
 if(a%2==0)//checks if number is even
 {
//if number is even
a=(a/2);
 }
 else
 {
//if number is odd
a=(3*a+1);
 }
cout.setf(ios::fixed);
cout.precision(0);//limits output numbers to 0 decimal places
double f=((9.5)*log10(a));//logarithmic function
cout <<f<<"   ";//final output sequence
}
cout<<endl;
cout<<endl;
cout<<"Now you can match the numbers in the sequence above to the Mix video clips
in"<<endl;
cout<<"Windows Movie Maker. Then put them in the order of the sequence above and
add music."<<endl;
cout<<"Then put them in the order of the sequence above and add music."<<endl;
cout<<endl;
cout<<endl;
char answer;
cout<<"Do you want to create another dance routine? (Y/N)"<<endl;
cin>> answer;
if((answer=='y')||(answer=='Y'))
hailstone();
}
//function#########################################################################
###
int main()
{
```

```
cout<<"\t\t\t $$$$     $$$$   $$  $$      $$"<<endl;
cout<<"\t\t\t$$  $$   $$  $$  $$   $$    $$"<<endl;
cout<<"\t\t\t$$   $$ $$   $$  $$    $$  $$"<<endl;
cout<<"\t\t\t$$    $$$    $$  $$     $$$$"<<endl;
cout<<"\t\t\t$$      $      $$  $$       $$"<<endl;
cout<<"\t\t\t$$             $$  $$     $$$$"<<endl;
cout<<"\t\t\t$$             $$  $$    $$  $$"<<endl;
cout<<"\t\t\t$$             $$  $$   $$    $$"<<endl;
cout<<"\t\t\t$$             $$  $$  $$      $$"<<endl;
cout<<endl;
cout<<endl;
hailstone();
     system("PAUSE");
     return 0;
}
```

Appendix D- Pom Dance

```
#include <iostream.h>
#include <stdlib.h>
#include <math.h>

/*The Rhythm of Hailstones: Pom. This program uses logarithms and the Hailstone
Theory to output a sequence of numbers. Each number corresponds to a video clip
for the Pom dance routine. There are 51 different dance routines that can be
created by this program.*/

/*This is the main function of the program. It includes the Hailstone Theory
formulas and the logarithmic function.*/
//function@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
void hailstone()
{
int a;//starting number
cout<<"Enter a number between 50 and 100."<<endl;//user inputs starting number
cin>>a;
if(a<50 || a>100)//limits range
{
cout<<"The number you entered is not in the 50-100 range"<<endl;
hailstone();
}
for(int count = 0; a!=1; count++)
 {
 if(a%2==0)//checks if number is even
 {
```

```
//if number is even
a=(a/2);
 }
 else
 {
//if number is odd
a=(3*a+1);
 }
cout.setf(ios::fixed);
cout.precision(0);//limits output numbers to 0 decimal places
double f=((9)*log10(a));//logarithmic function
cout <<f<<"   ";//final output sequence
}
cout<<endl;
cout<<endl;
cout<<"Now you can match the numbers in the sequence above to the Pom video clips
in"<<endl;
cout<<"Windows Movie Maker."<<endl;
cout<<"Then put them in the order of the sequence above and add music."<<endl;
cout<<endl;
cout<<endl;
char answer;
cout<<"Do you want to create another dance routine? (Y/N)"<<endl;
cin>> answer;
if((answer=='y')||(answer=='Y'))
hailstone();
}
//function#########################################################################
###
int main()
{
cout<<"\t\t############  ###########   ####    ####"<<endl;
cout<<"\t\t##          ##  ##        ##  ## ##  ## ##"<<endl;
cout<<"\t\t##          ##  ##        ##  ##  ## ##  ##"<<endl;
cout<<"\t\t##          ##  ##        ##  ##   ###   ##"<<endl;
cout<<"\t\t############  ##        ##  ##    #    ##"<<endl;
cout<<"\t\t##              ##        ##  ##         ##"<<endl;
cout<<"\t\t##              ##        ##  ##         ##"<<endl;
cout<<"\t\t##              ##        ##  ##         ##"<<endl;
cout<<"\t\t##              ###########  ##         ##"<<endl;
cout<<endl;
cout<<endl;
hailstone();
     system("PAUSE");
     return 0;
}
```

Appendix E-Sequence Graph