



Visualization of groundwater pollution using computer graphics  
by James S Palakovich

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in  
Computer Science  
Montana State University  
© Copyright by James S Palakovich (1990)

**Abstract:**

Water from subsurface systems comprises the largest majority of all the earth's drinking water, yet common waste disposal practices are continuously degrading the quality of our water resources. The contamination problem of subsurface water systems has been obvious for some time, although recently the problem has become progressively worse. In order to maintain the quality of our drinking water, it is necessary to develop techniques for extracting contaminating substances from the groundwater. When developing these extraction techniques, the predominant problem is visualizing the groundwater system. In this paper we describe a system that has been developed, using three dimensional computer graphics, to display models of groundwater systems including the watertable surface, the ground surface, and the pollution plume. We will show some examples of output using superficial data for a groundwater pollution system.

**VISUALIZATION OF GROUNDWATER POLLUTION  
USING COMPUTER GRAPHICS**

by

James S. Palakovich

A thesis submitted in partial fulfillment  
of the requirements for the degree

of

Master of Science

in

Computer Science

**MONTANA STATE UNIVERSITY**  
Bozeman, Montana

November 1990

1378  
P1225

-ii-

**APPROVAL**

of this thesis submitted by

James S. Palakovich

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

Nov 27th 90  
Date

J. Dembigl Stanley  
Chairperson, Graduate Committee

Approved for the Major Department

Nov 27th 90  
Date

J. Dembigl Stanley  
Head, Major Department

Approved for the College of Graduate Studies

November 28, 1990  
Date

Henry Parsons  
Graduate Dean

### STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgements of source are made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major Professor, or in his absence, by the Dean of Libraries when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature

James L. Paluszak

Date

November 27, 1990

## TABLE OF CONTENTS

	Page
LIST OF FIGURES .....	v
ABSTRACT .....	vi
CHAPTER 1	
INTRODUCTION .....	1
CHAPTER 2	
SIMULATION MODEL .....	5
Modeling the Ground and Watertable Surfaces .....	5
Transparency .....	7
Modeling the Pollution Plume .....	10
Calculating the Distance Between Surfaces .....	13
Drawing Contour Lines .....	16
CHAPTER 3	
IMPLEMENTATION .....	18
Major Data Structures Used .....	18
Ground and Watertable Surfaces .....	18
The Pollution Plume .....	19
Determining Distance Between Surfaces .....	20
Drawing Contour Lines .....	21
Updating the Scene .....	22
CHAPTER 4	
CONCLUSIONS AND NEW DIRECTIONS .....	26
REFERENCES CITED .....	28

## LIST OF FIGURES

Figure	Page
1. B-Spline Example .....	8
2. Transparent Surfaces .....	9
3. Pollution Plume Colors .....	11
4. Triangle Subdivision .....	12
5. Pollution Plume .....	13
6. Viewer Interaction .....	14
7. Contour Lines .....	17
8. Wire Frame Mesh .....	23
9. Spherical Coordinate System .....	25
10. Final Groundwater System .....	26

## ABSTRACT

Water from subsurface systems comprises the largest majority of all the earth's drinking water, yet common waste disposal practices are continuously degrading the quality of our water resources. The contamination problem of subsurface water systems has been obvious for some time, although recently the problem has become progressively worse. In order to maintain the quality of our drinking water, it is necessary to develop techniques for extracting contaminating substances from the groundwater. When developing these extraction techniques, the predominant problem is visualizing the groundwater system. In this paper we describe a system that has been developed, using three dimensional computer graphics, to display models of groundwater systems including the watertable surface, the ground surface, and the pollution plume. We will show some examples of output using superficial data for a groundwater pollution system.

## CHAPTER 1

### INTRODUCTION

Contamination of the earth's water supplies poses a serious threat to the environment. Petroleum, chemicals, and toxic waste are polluting the oceans, lakes, streams, and groundwater supplies of the earth on a daily basis. While there has been a growing concern over surface water pollution, this concern has not been shared with the subsurface environment. In fact, many solutions to the surface water problems, such as deep-well injection of liquid wastes and sanitary landfill for solid wastes, can lead to increased groundwater pollution [2] and [10]. Another major problem with groundwater contamination is the length of time it takes for a water table to cleanse itself (residence time). River water has a residence time of approximately two weeks whereas groundwater moves slowly and residence times in the 10's, 100's, and even 1000's of years are not uncommon. To prevent the deterioration of groundwater quality, it has become necessary to develop a methodology for monitoring, analyzing, and predicting the movement of contaminants through groundwater. The hope of formulating such a methodology has motivated the development of many predictive tools in the form of mathematical models designed to simulate the transport of contaminants through groundwater systems [1] and [9]. This paper describes a visualization project that has been developed using computer graphics to monitor contaminant plumes which may be derived from these predictive methods.

Of all the earth's water reserves, groundwater is the most difficult to monitor. Because of the heterogeneities inherent in subsurface systems, zones of degraded groundwater can be very difficult to detect. The U.S. Environmental Protection

Agency has reported that almost every known instance of aquifer contamination has been discovered only after a water-supply well has been affected [5]. In order to detect and remove groundwater contamination, one must first understand the behavior of groundwater. Difficulty in understanding groundwater behavior is primarily due to the visualization obstacle. To truly understand the situation, it is necessary to have an accurate visual model of the contaminant system. Our project uses three dimensional computer graphics techniques to display the ground and water surface features needed for visualizing a groundwater pollution scenario.

The minimum visual information needed to understand a groundwater pollution scenario is: the ground surface features, the watertable surface features, the pollution plume, and the different concentration levels of the pollution plume. In addition to these basic features, two enhancements have been implemented to further increase the usefulness of our system: the ability to draw contour lines on both the ground and watertable surfaces and the ability to approximate the distances between the two surfaces. A brief discussion of how all of these features are used follows. A more detailed discussion is contained in Chapter 2.

In our visualization system, ground and water surface features are simulated using spline surfaces, or a collection of small patches which together form a smooth surface. To generate the spline surfaces used for the ground and watertable surfaces, a variable set of  $(x,y,z)$  positional data control points, usually in rectangular form, are needed for each surface. These control points are not actual points on the surfaces themselves, but instead act as weights to control the shape of the surfaces local to them. Because of this, the surfaces which are generated are not totally accurate, but they are a reasonably close approximation.

Generating the pollution plume requires a collection of data values which consist of an  $(x,y,z)$  position, and a contaminant concentration value from 1 to 10 (1 being the lowest concentration). The data points are used to create a collection of uniformly spaced points contained within a sphere, herein called a starburst. The reason for using these starbursts is to achieve a reasonable amount of anti-aliasing, or blending when all the starbursts are displayed at once. In other words, when all the starbursts have been generated and displayed, the result is a smooth, realistic looking pollution plume. Another advantage to this type of plume, as opposed to using a solid, opaque plume, is the ability to see the multiple concentration levels through one another.

Concentration levels of the pollution plume are shown using different color levels. Ten unique colors are used to display concentration levels from high, though moderate, to low. For instance, high concentration is shown using bright red as the color. Concentration levels between high and moderate are shown using colors which fade from red to yellow. Concentration values from moderate to low are shown by fading from yellow to cyan. These three colors (red, yellow, and cyan) provide good color separation so varying concentration levels can easily be distinguished.

The technique used for drawing contour lines on the ground and watertable surfaces is very similar to the technique used for generating the surfaces themselves. Equations which are used for generating the spline surfaces are also used for drawing contour lines. Minimum and maximum elevations and interval size are required for the contour drawing routine, along with the same set of control points used for generating the surface.

Calculating the distance between the ground and watertable surfaces is accomplished using an iterative method based on a two dimensional Newton's method [4]. An  $(x,y)$  coordinate pair, input by the viewer, is used to approximate the parameters  $(s$  and  $t)$  used to generate the surface. Given an initial guess for parameters  $s$  and  $t$ , each cycle of the iteration process generates a closer approximation of the actual  $s$  and  $t$  values. Iteration continues until the  $s$  and  $t$  fall within a given epsilon of their actual values. The iterative method is used on both surfaces with the same set of input.

In Chapter 3, we cover the data structures used in each of the above mentioned modules and discuss the screen updating process. Finally in Chapter 4, we cover conclusions and new directions and give some additional examples of program output.

## CHAPTER 2

### SIMULATION MODEL

This chapter contains a detailed description of each major component of the visualization system. Although some components are related to each other in one way or another, they will each be discussed separately under their corresponding sub-headings.

#### Modeling the Ground and Watertable Surfaces

The ground and watertable surfaces are probably the most important components of our system, both as visualization and discussion tools. As visualization tools, the surfaces direct the viewer's attention to the appropriate locations of the scene. As discussion tools, they set up the basis for virtually every other component of our system. Therefore, a great deal of this chapter will be concentrated on the discussion of these surfaces.

Ground and water surfaces are developed using a database of information about the surfaces. Each entity in the database contains an  $(x, y)$  coordinate pair, a watertable elevation  $z_1$ , and a ground elevation  $z_2$ . The coordinate  $(x, y, z_2)$  defines a point on the ground surface, while  $(x, y, z_1)$  defines a point on the watertable surface directly under  $(x, y, z_2)$ .

Constructing surfaces is based on a mathematical model for parametric cubic curves. A parametric cubic curve is a smooth curve which is defined using three parametric equations (one each for  $x$ ,  $y$ , and  $z$ ). Using parametric equations, instead of equations based on the  $x$ ,  $y$ , and  $z$  variables themselves, eliminates the problem of handling infinite slopes. Parametric equations are functions of a single parameter

such as  $t$ , such that  $0 \leq t \leq 1$ . Instead of using slopes, parametric equations use tangent vectors which need never be infinite. Parametric cubic curve equations can easily be generalized to parametric bicubic surface equations by defining cubic equations of two parameters,  $s$  and  $t$ . Varying both parameters from 0 to 1 defines all points on a surface patch. If one parameter is assigned a constant value and the other parameter is varied from 0 to 1, the result is a cubic curve.

What follows is not intended to be an exhaustive discussion to the mathematics of parametric bicubic surfaces. For a more detailed discussion, see [6] and [3]. We will merely cover the mathematics sufficient for understanding of this document.

The evaluation of a point on a bicubic surface patch may be expressed as a matrix multiplication of the form:

$$x = X(s,t) = SMP_x M^T T^T, \quad (2.1)$$

$$y = Y(s,t) = SMP_y M^T T^T, \quad (2.2)$$

$$z = Z(s,t) = SMP_z M^T T^T. \quad (2.3)$$

where  $s$  and  $t$  are assumed to be between 0 and 1. A "T" superscript indicates the transpose of a matrix. Henceforth, only the equations for  $x$  will be stated explicitly;  $y$  and  $z$  are exactly analogous.

In the equations above, the vectors  $S$  and  $T$  are:

$$S = [s^3 \ s^2 \ s \ 1]$$

$$T = [t^3 \ t^2 \ t \ 1]$$

and matrix  $M$  in the B-Spline formulation is:

$$M = \frac{1}{6} * \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

$P$  contains the coordinates of the 16 control points defining the patch. In a B-Spline patch,  $P_x$  takes the following form:

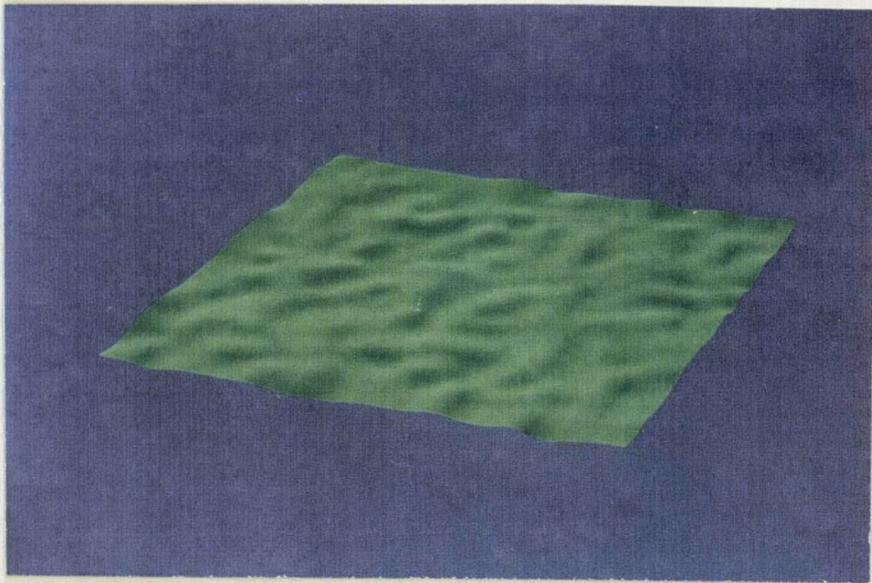
$$P_x = \begin{bmatrix} x_{i-1, j-1} & x_{i, j-1} & x_{i+1, j-1} & x_{i+2, j-1} \\ x_{i-1, j} & x_{i, j} & x_{i+1, j} & x_{i+2, j} \\ x_{i-1, j+1} & x_{i, j+1} & x_{i+1, j+1} & x_{i+2, j+1} \\ x_{i-1, j+2} & x_{i, j+2} & x_{i+1, j+2} & x_{i+2, j+2} \end{bmatrix}$$

In the above matrix,  $x_{i, j}$ ,  $x_{i+1, j}$ ,  $x_{i, j+1}$ , and  $x_{i+1, j+1}$  approximate the four corner points of the  $x$  patch. All other values are points which surround the patch.

The parametric surface defined above is called a B-Spline surface. B-Spline surfaces do not in general interpolate (pass through) any control points, but they are continuous and also have continuity of tangent vector and of curvature (that is, their first and second derivatives are continuous at the endpoints). Other types of surfaces which do interpolate the control points exist, but have continuity only for the first order derivatives, not for the second. A B-Spline surface is generally a smoother surface, but because the surface does not interpolate the control points, it is usually not a completely accurate approximation of the actual surface. Even so, B-Splines are still well suited for modeling the ground and water surfaces of our system. A sample watertable surface using B-Splines can be seen in Figure 1.

### Transparency

Transparent surfaces greatly enhance the visual understanding of the system. For instance, it is necessary to be able to see the pollution plume under the water surface, or see the water surface under the ground surface. The transparency technique used in our system is called screen-door transparency [8].



**FIGURE 1**  
**B-Spline Example**

Screen-door transparency is implemented by imposing a bit mask which defines which pixels can be affected by subsequent graphics primitives. A 4x4 cell, which is duplicated over the entire screen, contains 16 bits which can be set to 0 or 1 (pixel-off and pixel-on respectively). Thus, there are 17 levels of transparency: 0 (opaque) through 16 (completely transparent), based on the total number of bits set to 1. The bits in the mask correspond to the bits in the 4x4 transparency pattern in the following manner:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

maps into:

0	1	2	3
4	5	6	7
8	9	10	11









































