Bull. Mar. Environ. Res. Inst.
Cheju Nat. Univ., 30(2) : 49-55. 2006

제주대 해양과환경연구논문집, 30(2) : 49-55. 2006

Visualization of the Diffusion Process Using Three Dimensional OpenGL

Lee Byung-Gul and Yeu Yeon*

Dept. of Civil & Environmental Engineering Cheju National Univ., S. Korea
**Dept. of Civil & Environmental Engineering, Univ. of Wisconsin, USA*

3차원 OpenGL을 이용한 확산과정의 시각화에 대한 연구

이병걸 · 유 연*

제주대학교 해양과학부 토목환경공학전공
*위스콘신대학 토목환경공학과

Abstraction

The material diffusion process was visualized using programming of finite difference equation based on OpenGL. To do this, we derived three dimensional visualization algorithm based on Borland C++, OpenGL and central finite difference method of diffusion equation with open (Neuman type boundary) and closed boundary(Dirichlet type boundary) conditions. Numerical example of Two Creeks of Wisconsin was proposed to demonstrate the derived the visualization algorithm applicability. The calculated results showed reasonably well, particularly, for the time-dependant diffusion process. In the example, the material diffusion process was strongly related with water speed and diffusion coefficient. From this result, we found that OpenGL is very useful graphic library and the developed our algorithm can be applied to real ocean or river for the diffusion process.

1. Introduction

Water flow or material diffusion visualization has always been a demanding application for engineering and scientific visualization(Pagendarm and Water, 1995). Banks and Singer(1995) studied visualizing unsteady flow using a predictor-corrector technique that is designed to capture elongated vortices. Wischgoll and Scheuermann (2001) also developed an algorithm for the calculation of streamlines that detects cycling around closed stream lines. In the study, they showed the visualization of flow by linear interpolation on triangular. In GIS visualization application, Kraus(1994) developed methods of visualizing the quality of surfaces and their derivatives in GIS.

The diffusion process is generally related to not only thermal but also material diffusion in oceans and lakes. In order to study the flow or diffusion behavior of water, we first used numerical simulation based on mathematical equations. Then we visualized the numerical results using OpenGL in a computer program we modified for the purpose. OpenGL is a library of 2D and 3D graphic routines, which interface to graphics hardware. The greatest advantage to using OpenGL is that it is orders of magnitude faster than a ray-tracer and is extremely portable and very fast (Richard and Sweet, 1996). Since its introduction in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms. OpenGL fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions. Developers can leverage the power of OpenGL across all popular desktop and workstation platforms, ensuring wide application deployment (Silicon Graphics, 1998).

We used OpenGL to visualize the diffusion process in a lake. The diffusion process can be described by partial differential equation. The numerical solution of this equation was derived from a finite difference scheme, which is an initial value problem. The finite difference equations were derived from the diffusion equation using Taylor series expansion. Our goal was to describe the detailed progression of the solution by representing it visually. We applied our techniques to diffusion along a stretch of water near the Wisconsin shoreline of Lake Michigan.

2. The Software Developed

To compute and display the diffusion of concentrations we modified a C++ program with OpenGL. The program was written in C++ using Borland C++ Builder, a development environment, which removes many of the complications of writing software for Microsoft Window operating systems. The program also utilizes a library of routines for image processing written by Dr. Peter R. Weiler(Scarpace and Weiler, 1999). To interface with the OpenGL routines of Windows the program uses the Krumbach OpenGL control, which was written by Ted Krumbach and Pete Weiler of Unviersity of Wisconsin, Madison, to provide a simplified set of routines for controlling the OpenGL parameters. This control is a Borland VCL control, which is similar to the OCX or COM controls used in Microsoft Visual Basic.o use the Borland control for OpenGL, we created three files, those are BPI, BPL and LIB files. BPI is an import library file, BPL is the runtime library file and LIB is a library for static linking file. Based on the three files, the control provides a window on the screen where OpenGL can draw objects.It simplifies the programmer's work by handling various initialization tasks and by providing simple ways for him to set the lighting, background and position of objects drawn.It does not actually draw objects in the window.For drawing, the programmer must use the standard OpenGL routines. The software drew a three dimensional object in on-screen window. OpenGL routines were used to draw the three dimensional object as it would appear if seen from a specified position in space. We modified the program to 1) display a representation of the surface of Lake Michigan and the adjacent coastline, 2) compute the diffusion of a concentration of material added to the lake at one or two points, and 3) display the progression of the diffused material on the lake.

3. Finite Difference Scheme of the Diffusion Equation

The two dimensional diffusion equation is

$$\frac{\partial C}{\partial t} + u\frac{\partial C}{\partial x} + v\frac{\partial C}{\partial y} = A_H\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2}\right) \quad \cdots\cdots\cdots (1)$$

where C is the concentration of diffusion material, u and v are x and y directional water speeds, and A_H is the diffusion coefficient.

To calculate the difference equation, a finite grid is generated by creating vertices as shown in Figure 1.

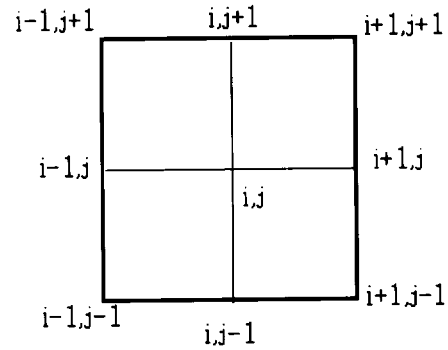

Figure 1. Finite difference grid system. We used central finite difference method.

Based on Figure 1, we derived the following finite difference equation

$$\frac{C^{n+1}_{i,j} - C^{n}_{i,j}}{\Delta t} + u\frac{C^{n}_{i+1,j} - C^{n}_{i-1,j}}{\Delta x} + v\frac{C^{n}_{i,j+1} - C^{n}_{i,j-1}}{\Delta y}$$

$$A_H\left(\frac{C^{n}_{i+1,j} + 2C^{n}_{i,j} - C^{n}_{i-1,j}}{\Delta x^2} + \frac{C^{n}_{i,j+1} + 2C^{n}_{i,j} - C^{n}_{i,j-1}}{\Delta y^2}\right) \quad \cdots\cdot (2)$$

where n is time step and i and j are space steps of x and y, respectively. $\Delta t, \Delta x, \Delta y$ are time interval and space interval of x and y. In Eq.(2), we used explicit finite difference method with a forward scheme for the time integration and the central space for space domain. Compared to implicit one that can be described as matrix form of time integration, the explicit usually produces faster computations. Finally, the Eq.(2) can be transformed into simple form as,

$$C^{n+1}_{i,j} = AC^{n}_{i+1,j} + BC^{n}_{i-1,j} + DC^{n}_{i,j} + EC^{n}_{i,j+1} + FC^{n+1}_{i,j-1} \quad \cdots\cdots (3)$$

where $A = (-u\frac{\Delta t}{\Delta x} + A_H\frac{\Delta t}{\Delta x^2})$.

$$B = (u\frac{\Delta t}{\Delta x} + A_H\frac{\Delta t}{\Delta x^2}),$$

$$D = (1 - 2A_H\frac{\Delta t}{\Delta x^2} - 2A_H\frac{\Delta t}{\Delta y^2}),$$

$$E = (-v\frac{\Delta t}{\Delta y} + A_H\frac{\Delta t}{\Delta y^2}),$$

$$F = (v\frac{\Delta t}{\Delta y} + A_H\frac{\Delta t}{\Delta y^2})$$

From the Eq.(3), we can calculate the diffusion process at every time steps.

4. Numerical Examples

To demonstrate the utility of OpenGL for the visualization of the diffusion process, we have applied our model to the visualization of material diffusion in Lake Michigan along the Wisconsin coast(Figure 5). We used digital elevation data and geometric coordinates of an actual portion of the coast.

To implement the numerical computations, we adopted two boundary conditions. For the open boundary in the lake, we used a Neuman typeboundary condition. For the closed boundary provided bounded by the shore, we used a Dirichlet type boundary condition. The boundary conditions can be described as following;

Neuman boundary condition: ∇ C = 0 in open water

Dirichlet boundary condition: C = 0 at the coastal line

To determine the coastline where the Dirichlet boundary condition apply, we used a digital elevation model (DEM). DEM is a digital representation of a portion of the earth's terrain over a two dimensional surface. The DEM we used gave the elevations at points on 170 by 155 grid. This DEM also served as the basis of the visual representation of the land surface in the OpenGL display. To facilitate drawing of the terrain, the DEM data areprovided to the OpenGL routines as a series of triangles. The routines are provided with the horizontal coordinates and elevation of the corners of the triangles, which lie on the grid points of the DEM. Figure 2 illustrates how the rectangular grid is represented as a series of triangles. Here i and j represent the horizontal coordinates of a point. Each square of four grid points is represent by two triangles. The first square in the figure is represented by the triangle with corners at the points *(i, j)*, *(i, j+1)*, and *(i+1, j)* and by the triangle with corners *(i, j+1)*, *(i+1, j)* and *(i+1, j+1)*. The other squares on the grid are similarly represented.

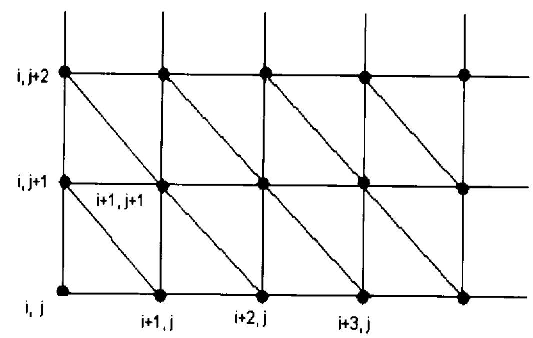

Figure 2. Generating terrain using right isosceles triangles.

The calculations of diffusion are done by applying Eq. (3). For each row on the grid, the starting point and ending point of water are stored in two arrays (Figure 3). In our study, the starting point is at the coastline and the ending point is in the water. The concentration value at coastal line is assigned to zero (Dirichlet boundary condition). At the ending point in water the gradient value of concentration is set to zero (Neuman boundary condition).

This technique can have a problem as shown in Figure 4. If the areas of water divided to several regions by land, the diffusion equations can not be used for grid points on the land. A binary two-dimensional array is used to represent this. The value in this array of a grid point on water is assigned one and the value on land is assigned zero. The concentration values multiplied by the value of a binary two-dimensional array on land should be zero(Dirichlet boundary condition). Those values on water are retained because of being multiplied by one.

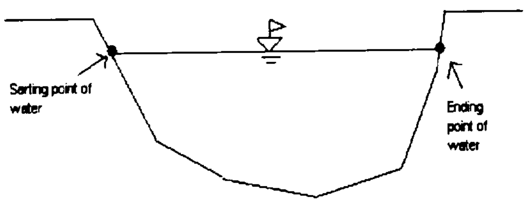

Figure 3. Starting and Ending point of water.

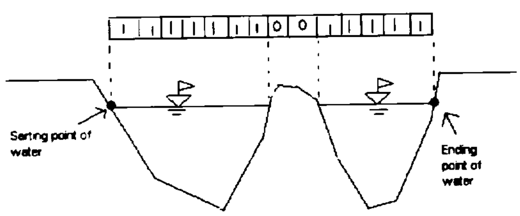

Figure 4. Binary two-dimensional array

The last step is displaying results using Krumbach OpenGL control. Land, water, and concentration has respectively been displayed yellow, blue, and red. As the concentration value becomes higher, the color changes from blue to bright red. On the computer used for this work the process of drawing the results was slow, so the display was redrawn only after every 100 calculations.

Figure 5. Numerical example area(Two Creeks: Black Circle).

The calculated results are shown in Figures 6, 7, 8 and 9. In these figures, yellow color is land that is calculated from DEM data and blue is lake surface. The space interval of grid is assumed as 1000 m. Figure 6 shows one point source diffusion process. In the case, u=v=0 so that there is a circular diffusion pattern is produced. Figure 7 has u=1.0 m/sec, v= 0.2 m/sec. In this case, a one directional diffusion process happens as time changes. Figures 8 and 9show the periodical velocity effect produced by $u = v = a\cos(\frac{2\pi}{T}t)$. T is the period a of periodic wave. This water speed is similar to the pattern of a tidal wave in the ocean.

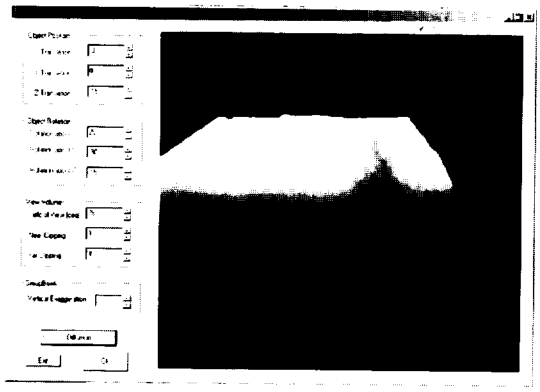

Figure 6. The diffusion process of one point source without water speed.

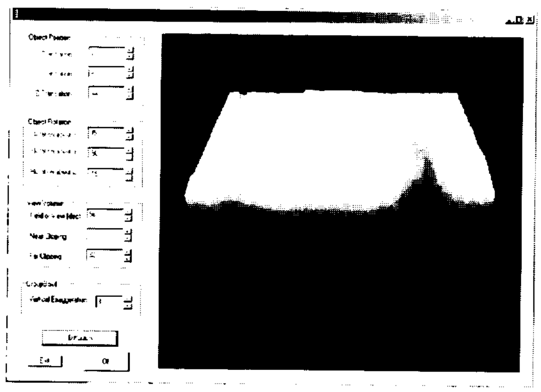

Figure 7. The diffusion process of two point sources with constant non-zero water speeds in the x and y directions.

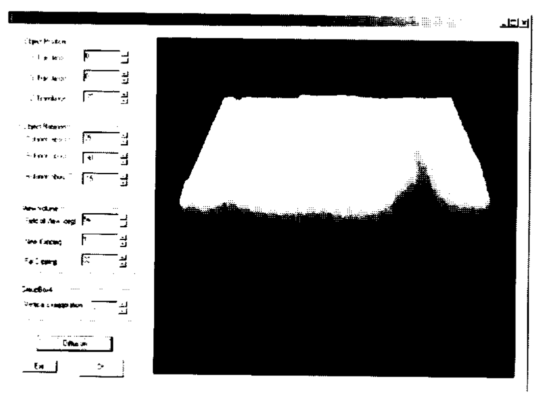

Figure 8. The diffusion process of two point sources withperiodic water speed in the x and y directions.(Time = 1 hours after releasing diffusion materials)

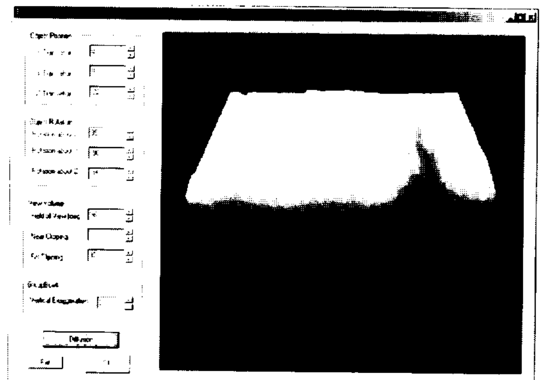

Figure 9. The diffusion process of two point sources withperiodic water speeds in the x and y directions.(Time = 3 hours after releasing diffusion materials)

5. Conclusion

In this paper, we presented the visualization using OpenGL of a diffusion process based on diffusion equation. The Open GL was a very useful technique for the visualization of material diffusion. Using the developed programming based on Borland C++ and OpenGL, we can display the diffusion in space and time easily. In particular, time dependant diffusion problems related with periodical wave motion like tidal current can be treated in our numerical model. If the coefficients of the diffusion equation are determined by experiments, this resultcan be applied to real management of water quality. The spreading hot water discharged from power plants or factories is a kind of dispersion, so this phenomenon can be predicted using this model.

6. References

[1] Banks, D. C. and Singer, B. A.(1995) A Predictor-Corrector Technique for Visualizing Unsteady Flow, IEEE Transaction and Computer Graphics, 1(2), 151-163.

[2] Kraus, K.(1994) Visualization of the Quality of Surfaces and Their Derivatives, PE & RS, 6(4), 457-462.

[3] Pagendarm, H.-G. and Water, B.(1995) Competent, Compact, Comparative Visualization of a Vertical Flow Field, IEEE Transaction and Computer Graphics, 1(2), 142-150.

[4] Richard S. Wright, Jr. and Sweet, M.(1996) OpenGL Super Bible, Waute Group Inc. p. 714.

[5] Wischgoll, T. and Scheuermann, G.(2001) Detection and visualization of closed streamlines in planar flows, IEEE Transaction and Computer Graphics, 7(2), 165-172.

[6] Silicon Graphics, Inc.(1998) OpenGL industrial's foundation for high-performance graphics, http://www.opengl.org/developers/about/opengl.pdf.

[7] Scarpace, F. L. and Weiler, P.R.(1999) Teaching ImageProcessing Algorithms using a C++ Rapid Application Development Environment, Proceedings of 1999 ASPRS annual Conference, May 1999, Portland, Oregon.

3차원 OpenGL을 이용한 확산과정의 시각화에 대한 연구

이 병 걸 · 유 연*

토목환경공학전공 제주대학교

*토목환경공학과 위스콘신대학

요 약

Opem GL을 이용하여 2차원 확산과정에 대한 시각화를 연구하였다. 이를 위하여 Borland C++소프트웨어를 이용하여 계산하였다. 확산과정은 유한차분법을 이용하여 계산하였다. 경계조건은 열린경계(Neuman type)와 닫힌경계(Dirichlet type)경계를 이용하였다. 그 결과 확산과정에 대한 시각화가 제대로 이루어 졌으며, 시간에 따른 확산범위가 이상적으로 나타나는 것으로 나타났다. 이러한 결과로 미루어 볼 때 Open GL을 이용한 물질확산의 시각화의 응용이 실제해양에 적용이 가능한 것으로 판단된다.