Volume xx (200y), Number z, pp. 1–5

Simulation of Levee Erosion

Zhongxian Chen

Advanced Computer Graphics (CSCI-6530)
Rensselaer Polytechnic Institute

Abstract

In this report, I talk about my work on fluid simulation and erosion simulation. First some background knowledge about both fluid and hydraulic erosion simulation is discussed and then some technical details about a levee erosion simulator is presented. Some implementation results of the system are also presented and discussed. Finally, I will also discuss the difficulties I faced when doing the project and some work that can be done in the fuure to improve the system.

1. Motivation

Levee failure can lead to great disasters to human beings, especially when it happens in some areas with large population, such as New Orleans. As a result, more and more study need to be done on analysing the erosion of levees by flood. A research project about simulating levee erosion is currently going on in the Computer Graphics Group of RPI. As a member of the group, I am enrolled in the project. That's reason why I have chosen to do something about simulating levee erosion. In this report, I will present my work on fluid simulation and erosion simulation.

2. Previous Work

The work presented in this report is primarily based on previous work in the areas of fluid simulation and hydraulic erosion simulation, which are presented respectively as follows.

2.1. Fluid Simulation

Fluid simulation has been an active research field for a long time. Many researchers have done a lot of work on simulating water and smoke. Harlow [HW65] is the first one who uses numerical methods to solve the Navier-Stokes Equation and simulate fluid based on that. In his paper, he describes a numerical investigation of time-dependent flow of imcompressible fluid, where the full 2D Navier-Stokes Equations are written in finite-difference form and are solved by discretizing both spatially and temporally. Based on the work of Harlow, Foster [FM96] presents a method that creates fluid-like animations by solving Navier-Stokes Equations in 3D. Many effects such as swirling motions and flow past objects can be obtained automatically by this method. One advantage of Foster's method is that it is easy to code because it is based on a finite differencing of the Navier-Stokes. That makes Foster's paper one of the most popular paper in the field of fluid simulation and a lot of work has been done based on it. But one disadvantage of it is that the time step used for the simulation must be small enough to avoid "blow-up". Such an disadvantage limits the speed of the simulation and thus makes it not suitable for real-time applications. In a later paper [FF01], Foster presents some work that can be done to make the simulation more stable and practical.

To produce real-time simulation, Stam [Sta99] introduced a semi-Lagrangian method that allows large time steps in the simulation by solving the Navier-Stokes Equations implicitly. Stam's algorithm is fast because it only needs to solve a large sparse system of equations that takes only O(n) time. But one disadvantage of it is that it suffers from too much numerical dissipation and would not be accurate enough for most engineering applications.

2.2. Hydraulic Erosion simulation

Mei et al. [XMH07] presents a efficient method to model hydraulic erosion phenomenon. Their method is based on the velocity field of running water, which is created with an efficient shallow-water fluid model. The calculated velocity field is used to compute the amount of eroded and deposed soil. The model of erosion calculation in this paper is used in my project.

Based on some results from [XMH07], Stava et al. [OS08] presents a method of simulating the erosion of soil by flows. Their simulation is based on height field and pipe-model and four different types of erosion are simulated to generate realistic results. Previous to that paper, the authors present a similar method [BB06] based on voxel grid and Navier-Stokes Equations.

3. Fluid Simulation

In my project, my codes of fluid simulation are primarily based on the codes of the second course project, which is an implementation of the method introduced in Foster's paper [FM96].

3.1. Navier-Stokes Equations

Navier-Stokes Equations are a set of equations that completely descibe the motion of a fluid at any point within incompressible flow such as water. In three-dimension, Navier-Stokes Equations can be written as follows:

$$\frac{\partial v}{\partial t} + (v \cdot \nabla) v = \frac{\nabla p}{\rho} + \frac{\eta}{\rho} \nabla^2 v, \qquad (1)$$

$$\nabla \cdot v = 0. \qquad (2)$$

Here v, t, p, ρ and η stand for velocity, time, pressure, density and viscosity, respectively. The left part of Equation(1) is the acceleration of the particle. On the right hand side, the gradient of pressure represents the force between particles in normal direction while $\frac{\eta}{\rho}\nabla^2 v$ stands for the force between particles in tangent direction (force caused by viscosity). Furthermore, Equation(2) comes from the fact that the density is constant at any point in incompressible fluid.

Following the way described in Foster's paper, we first need to discretize the Navier-Stokes Equations using finite difference approximation before solving it. And then we can use the discretize equations to calculate the velocity field in the next step based on that calculated in the previous step. With respect to Equation(2), a relaxation method is employed to iteratively update the velocity field for a sufficient large number of times in each time step.

3.2. Fluid Tracking

To track the position of the fluid, the Marker Particles method is introduced. A number of massless particles are convected in the fluid with local fluid velocity. The velocity at the location of a particle is calculated using an area weighting interpolation over the four nearest cell velocities, and then the resultant velocity is multiplied by the time step to get the new position of the particle. According the positions of the particles, we then labled each cell with as $EMPTY$, $SURFACE$ or $FULL$ as described in Foster's paper.

3.3. Boundary Settings

In order to make the implemenation complete, we need to set up the boundary conditions. For rigid boundaries, we need to have $v \cdot e_n = 0$ in the Navier-Stokes Equations. In other words, in the cells ajacent to the boundaries, the velocity perpendicular to the boundary should be set to zero. And if the boundary is non-slip, the tangential velocity at the boundary is set to zero. Otherwise, the boundary velocity should be set so as not to affect the tangential velocity at the boundary. Furthermore, the pressure in boundary cell is set to be equal to the ajacent full cell.

The cells representing the levee is generally treated as the boundary cells. But note that the setting in the levee cells should be slightly different from that in the boundary cells. The reason is that each boundary cell can only be ajacent to one non-boundary cell, while each levee cell can be ajacent to as many as five. So in order to make the levee as slip-boundary, we can not simply set the tangential velocity at the levee cells as we do at the boundary cells. What I do here is to make the tangential velocity identical to the velocity of its ajacent full cell whenever it is used for calculation. Finally, inflow and outflow need to introduced into the system. The inflow cells are fixed on the left boundary and the number of the inflow cells is adaptive to the current height of the water. Furthermore, the velocity at the inflow cells are fixed. The velocity at the outflow cells are set to be the same as that in the ajacent cells so that the fluid can flow out of the syste so that it does not cause upstrean artifacts.

3.4. Simulation Results

Some screen shots of fluid simulation are shown in Fig.1. The resolution measured in number of cells is $30 * 12 * 1$, the size of each cell is $0.1 * 0.1 * 0.1$ and the time step is 0.005 second.

4. Erosion Simulation

To specify the existence of the levee, I add one more status for the cells, namely LEVEE. For each cell with LEVEE status, I associate it with a certain amount of soil at the beginning. As we know, levee failure results from the accumulation of levee erosion. When water flows over the levee, some soil will be eroded and transported by the water. This process is primarily determined by the transport capacity of

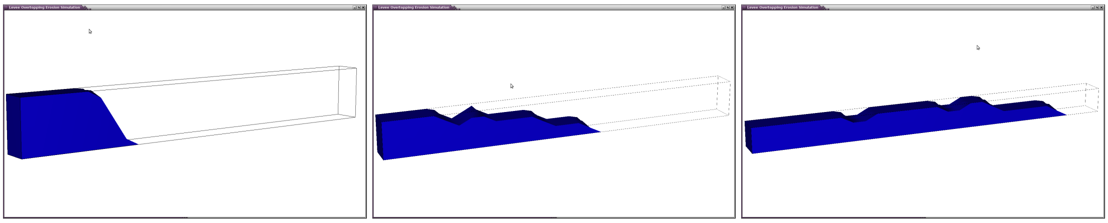

Figure 1: *Screen shots of fluid simulation.*

the flow [JS85]. As in [**?**], we employ the following equation to calculate the transport capacity:

$$C = K_c \cdot sin\alpha \cdot |v|, \qquad (3)$$

where K_c is a sediment capacity constant, α is the local tilt angle and v is the component of the velocity parallel to the eroded surface. Note that we have ignored the sediment here because it has little effect on the levee erosion. According to the equation, for the very flat terrains where the α value approaches zero, C will be very small, meaning that the water flow would pick up very little soil and the erosion effect is indistinctive. Therefore we need to limit the value of α by a pre-specified minimum threshold.

With the transport capacity calculated, the amount of soil eroded in a single time step can be calculated as follows:

$$S = C \cdot T, \qquad (4)$$

where T is the length of the time step. And then in each time step, we subtract the eroded amount from the eroded cells. When the soild amount of a cell reduces to zero, its status is changed from LEVEE to FULL.

5. Implementation Results

A series of images illustrating the process of levee erosion are shown in Fig.2. The resolution of the simulation is $60*24*1$, the size of each cell is $0.1*0.1*0.1$ and the time step is 0.005 second. The simulation, performed on a laptop computer with 1.73 GHz dual-core CPU and 1.00 GB RAM, took approximately 7 minutes. In the simulation, the water comes from the left boundary and overtops the levee as water level rises. The base of the levee on the other side of the water source is the portion being eroded first because the water over it is runs with high velocity. Almost the whole levee has been eroded at the end of the simulation.

6. Difficulties, Shortcomings and Future Work

One primary difficulty of the project is to make the fluid simulation stable. The codes are not easy to debug and there are a lot of small details that need to be taken care of. The first thing is that the time step must be small enough. According to Foster's paper, in order to make the system stable, the relationship between the time step and the spatial resolution should at least satisfy the following constrain:

$$1 > max[u * \frac{\delta t}{\delta x}, v * \frac{\delta t}{\delta y}, w * \frac{\delta t}{\delta z}]. \qquad (5)$$

So in order to make the simulation system, we need to increase the computational cost, namely reducing time step or increasing the size of the cells. But in actual cases, the system still crashes even we use small time steps, because the velocity can sometimes become quite large. Another thing we need to take care of is the value of the erosion constant in Equation(3). To make the erosion realistic, we need to give a correct value to that constant and thus take properties of the soil into consideration.

There are several problems in the current version of the simulator and thus a lot of work needs to be done in the future. First of all, the system is not so stable. It sometime crashes even with small time step (see the Fig.3). There can various reasons for that, thus it will take sometime to debug the codes. Second, now the model of the levee can not be loaded into the system automatically, so some work needs to be done on that. Third, both the model of erosion now used in the system and the configuration of the constants may be inacurate, more consideration is needed for that. Finally, we may try some other methods of fluid simulation, such as Smoothed Particle Hydrodynamics [DG96], and see if that attains better results.

7. Conclusion

In this report, I have presented the work I have been doing on fluid simulation and erosion simulation. A system that simulates levee erosion is presented and some implementation results are discussed. Furthermore, some background knowledge for the proposed system is also discussed. Since the system is quite far from perfect, much improvement needs to be done in the future. By doing this project, I have learned quite a bit about the theories and implementation of fluid simulation.

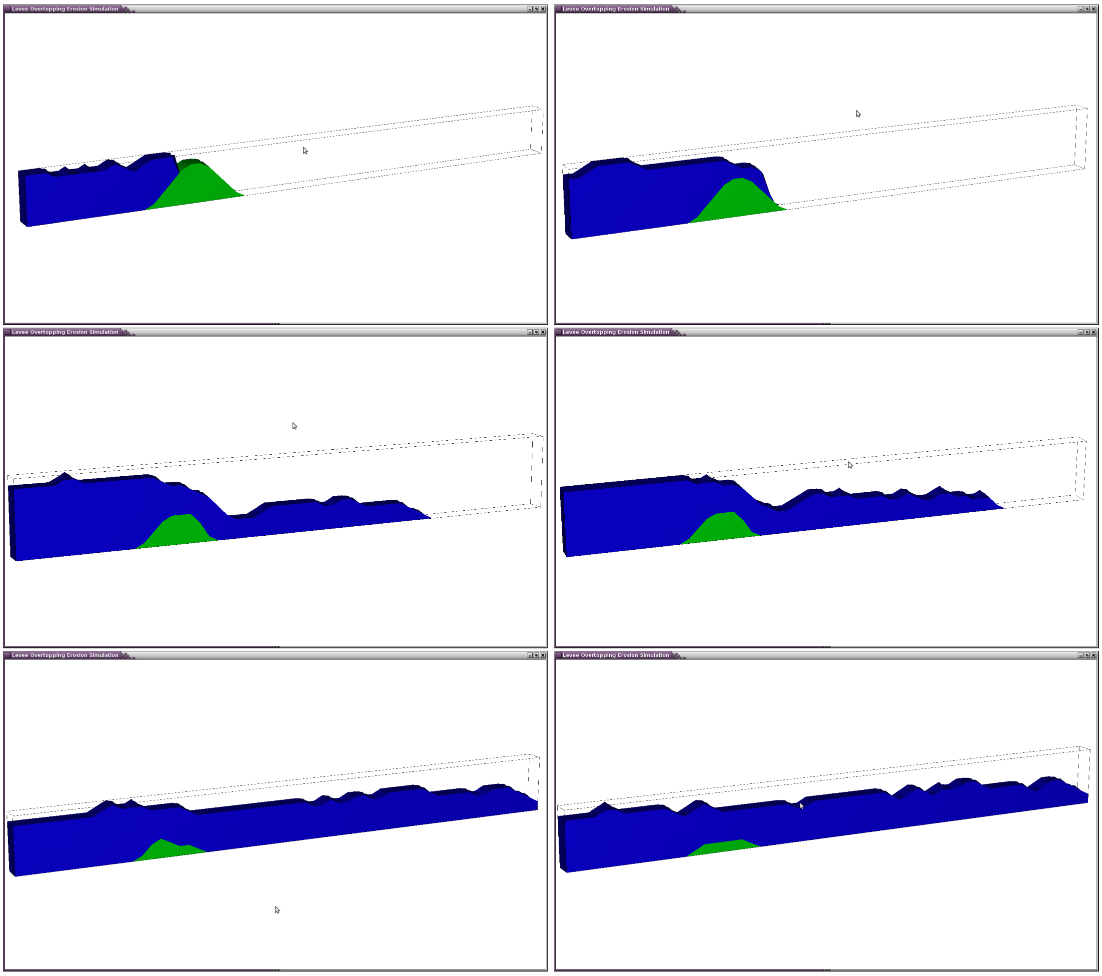

Figure 2: *Screen shots levee erosion simulation.*

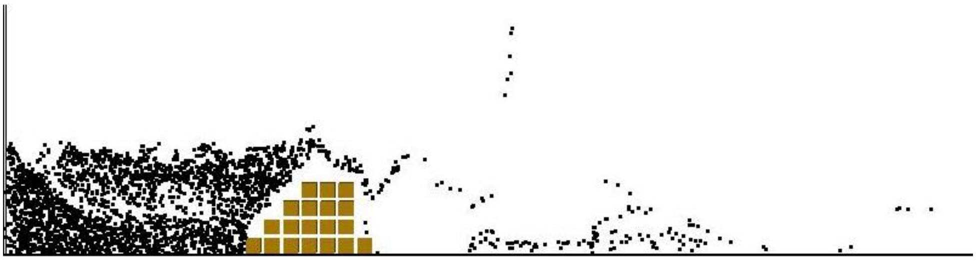

Figure 3: *The system crashes when* $\delta t = 0.001$ *and* $\delta x = \delta y = \delta z = 0.2$.

Acknowledgments

Thanks to Christopher Stuetzle for rendering Fig.1 and Fig.2 shown in this report and his discussion with me helped me a lot. Also thanks to him for writing codes of levee modeling that will be integrated into the system in the future.

References

[BB06] Benes B., Tesinsky V., Hornys J., Bhatia S. K.: Hydraulic erosion. *Computer Animation and Virtual Worlds 17*, 2 (2006), 99–108.

[DG96] Desbrun M., Gascuel M.: Smoothed particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation 96 (Proceedings of EG Workshop on Animation and Simulation)* (1996), Springer-Verlag, pp. 61–76.

[FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM, pp. 23–30.

[FM96] FOSTER N., METAXAS D.: Practical animation of liquids. In *Graphical Models and Image Processing* (1996), pp. 23–30.

[HW65] HARLOW F., WELCH E.: Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids 8*, 12 (1965), 2182–2189.

[JS85] JULIEN P., SIMONS D.: Sediment transport capacity of overland flow. *Transactions of ASAE 28*, 3 (1985), 755–762.

[OS08] STAVA O., BENES B., BRISBIN M., KRIVANEK O.: Interactive terrain modeling using hydraulic erosion. In *ACM Siggraph/Eurographics Symposium on Computer Animation* (2008).

[Sta99] STAM J.: Stable fluid. In *Proceedings of SIGGRAPH 1999* (1999), pp. 121–128.

[XMH07] MEI X., DECAUDIN P., HU B.: Physically based hydraulic erosion simulation on graphics processing unit. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications* (October-November 2007), pp. 47–56.