

# Hydraulic Erosion Modeling on a Triangular Mesh

Věra Skorkovská, Ivana Kolingerová and Bedrich Benes

**Abstract** Although hydraulic erosion modeling on a GIS terrain models has been addressed by a body of previous work, it still remains an open problem. In GIS, raster representation and triangular irregular networks (TIN) are the most commonly used surface models, because they are simple and offer implicit topological information. However, these data structures do not allow the simulation of erosion on concave terrain features, such as caves or overhangs. Other methods, more commonly used in the computational fluid dynamics, use volumetric data representation. They are able to model the 3D features, but they usually have high memory requirements and are computationally demanding. We propose a novel solution to the hydraulic erosion modeling problem that uses a triangular mesh data structure. Our framework allows for adaptive changes of the mesh resolution according to the local complexity of the terrain, which leads to lower memory requirements when compared to the volumetric approaches. Our data structure also supports the visualization of the concave 3D features, allowing the simulation and visualization of erosion on terrain elements such as tunnels or caves.

**Keywords** Hydraulic erosion · Erosion · Terrain modeling · Smoothed particle hydrodynamics · Triangular mesh

---

V. Skorkovská (✉) · I. Kolingerová  
NTIS—New Technologies for Information Society, University of West Bohemia,  
Univerzitní 8, 306 14 Plzeň, Czech Republic  
e-mail: [vskorkov@ntis.zcu.cz](mailto:vskorkov@ntis.zcu.cz)

I. Kolingerová  
e-mail: [kolinger@kiv.zcu.cz](mailto:kolinger@kiv.zcu.cz)

V. Skorkovská · I. Kolingerová  
Department of Computer Science and Engineering, Faculty of Applied Sciences,  
University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic

B. Benes  
Department of Computer Graphics Technology, Knoy Hall of Technology,  
401 N. Grant Street, West Lafayette, IN 47907-2021, USA  
e-mail: [bbenes@purdue.edu](mailto:bbenes@purdue.edu)

## 1 Introduction

The problem of hydraulic erosion simulation is a common open problem in GIS. The modern GIS systems, such as ArcGIS, mostly use raster data structure (regular heightfields) to provide such simulations. Regular heightfields are sufficient for most surface analysis done in GIS, but cannot be used for modeling of concave features, such as tunnels, caves or overhangs. However, these features often result from the hydraulic erosion.

The existing hydraulic erosion methods in GIS are often based on the raster data structure. The use of this structure allows for the creation of fast and interactive solutions at the cost of being constricted to only 2.5D non-adaptive models. Triangular irregular network (TIN) data structure brings the option to adaptively change the resolution of the terrain based on the local complexity of the surface, however, the restriction to 2.5D models still applies. A survey on hydrologic modeling in GIS can be found, e.g., in DeVantier and Feldman (1993) or Aksoy and Kavvas (2005).

Other approaches are capable of creating fully 3D models using a volumetric representation of the terrain, but these methods are generally very memory consuming and far from being interactive. The fully 3D hydraulic erosion simulation is more common in the field of computational fluid dynamics. An essential part of the erosion modeling is the simulation of fluid dynamics. The fluid dynamics can be described by the Navier-Stokes equations for incompressible Newtonian fluids (Acheson 1990) and its solutions can be divided into two main categories: Eulerian and Lagrangian approaches (Bridson 2008).

The Eulerian approaches divide the volume of the scene and track the fluid properties at fixed points. As the fluid moves through the scene, the tracked properties change according to the fluid state at the given position. Usually a uniform grid is used to simplify the necessary computation, however, the uniform space division leads to higher memory requirements. This approach gives accurate results, but it is very computationally expensive.

Benes et al. (2006) represent the scene as a regular grid, where each voxel is classified as either FULL (a voxel full of water, can contain dissolved material), EMPTY (an empty voxel containing only the air) or MAT (a voxel containing material). The authors define the processes that lead to the change of the state of the voxel. The method is capable of simulating fully 3D effects, however, it is not capable of running with real time response. Wojtan et al. (2007) simulate the corrosion and erosion of solid objects. The surface is stored as a level set. The erosion is then simulated by advecting the level set inward, the deposition by advecting it outward.

The Lagrangian approach methods represent the fluid using particles. The fluid properties are tracked at the particles as they move through the scene. This representation simplifies the fluid simulation as some of the conditions of the Navier-Stokes equations are automatically satisfied. The methods are generally faster, but less accurate than the Eulerian methods.

Smoothed-particle hydrodynamics (SPH) (Gingold and Monaghan 1977) is a particle-based approximative numeric solution to Navier-Stokes equations. The properties of the particles are calculated over the smoothing radius, a distance, over which the particles are influenced by each other. The contributions of each particle are weighted by their distance and density according to the smoothing kernel function used.

SPH is very often used in fluid simulation (Adams et al. 2007; Kipfer and Westermann 2006), but to the best of our knowledge, Krištof et al. (2009) were the first to combine the SPH fluid simulation with erosion. The SPH particles flow through the scene, eroding the terrain and transporting the sediment. The diffusion of the sediment is defined by a donor-acceptor scheme between the SPH particles. The terrain is stored as a raster, limiting the use of this method to 2.5D effects. Fully 3D terrain features, such as caves or overhangs, cannot be simulated with this approach.

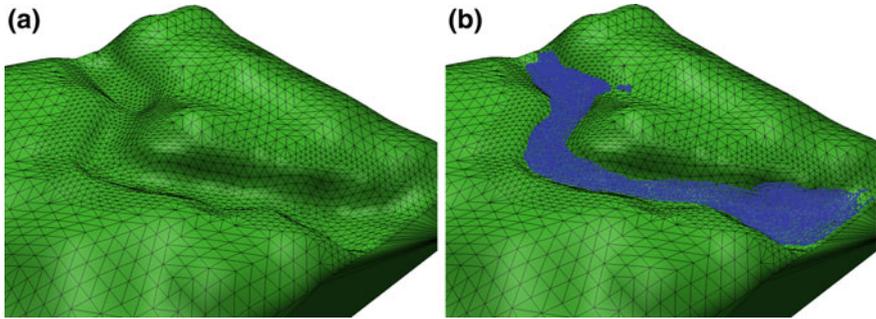
In this paper we propose a novel solution to the hydraulic erosion modeling problem. Our method combines the use of the SPH for the fluid simulation and the surface triangular meshes for the representation of eroded objects. The triangular mesh can be seen as a compromise between the TINs and the volumetric representations. It preserves the adaptability feature of the TIN models, but also allows the creation of models containing caves or overhangs. However, inner cavities cannot be modeled using this approach. The main advantage of simulating the fluid using the SPH method is that the particles are located only in the affected regions, limiting the computation to these areas.

## 2 Data and Methods

In this paper we propose a novel solution to the hydraulic erosion modeling problem. We simulate the fluid with the Smoothed-particle hydrodynamics (SPH) particles by using the approach of Krištof et al. (2009). The terrain is represented as a surface triangular mesh in order to simulate the erosion of elements such as tunnels or caves. An example is shown in Fig. 1. The model of a river bed has a higher resolution in the areas, where the erosion will be applied (Fig. 1a).

### 2.1 Data Structure

A surface triangular mesh is used to represent the terrain data. The mesh can either be derived from a raster data obtained, e.g., from an aerial scan, or it can be computed on a given set of points by a planar triangulation, or it can be prepared using a modeling software. Usually the approaches can be combined to produce the desired input terrain data: the base mesh is derived from the scanned data and further details, such as caves or tunnels, are added using a modeling software.



**Fig. 1** An example of the triangular mesh model in erosion simulation. **a** The original model has higher resolution in the erosion areas. **b** The fluid represented by the SPH particles

### 3 Method Overview

Every iteration of the proposed method consists of the following steps:

1. The properties of the SPH particles, such as pressure, speed and direction, are calculated.
2. The fluid-terrain interaction is computed.
3. Each particle moves in the calculated direction.
4. The erosion/deposition sediment exchange is calculated for the particles colliding with the terrain.
5. The triangular mesh is updated according to the amount of sediment eroded/deposited at the vertices.

#### 3.1 SPH Simulation

We simulate the fluid using the Lagrangian SPH method (Gingold and Monaghan 1977). The main advantage of the method lies in its adaptability—the particles are restricted to the areas containing the fluid. The computation is then performed only in the regions where the fluid is present.

The SPH defines the way the properties of the particles, such as pressure, speed and direction, are computed. The pressure is calculated based on the density of the particles and determines the speed and direction of the movement of the particle.

#### 3.2 Fluid-Terrain Interaction

The motion of a particle is defined by the particle properties, by the gravity, and by the surrounding terrain. The fluid-terrain interaction is a vital part of the erosion

simulation. A particle is influenced by all the mesh faces within the reach of its radius. To speed up the search for the nearby faces, a regular spatial subdivision is used. Using this auxiliary data structure, we do not have to search for the faces in the whole mesh, but only in the cells that fall into the radius of the particle. Each face  $f$  in the selected cells is checked for an interaction with the particle  $p$  as follows:

- The line  $l$  is constructed that goes through the position of the particle with the direction of the normal vector of the face  $f$ .
- If the line  $l$  intersects the face  $f$ , the orthogonal distance  $d$  between the face  $f$  and the particle  $p$  is calculated.
- If the distance  $d$  is smaller than the radius of the particle, the face  $f$  influences the motion of the particle  $p$ .

The face  $f$  contributes to the force affecting the particle with a force calculated using the penalty-force method (Amada 2006):

$$f = k^s dn + k^d (v \cdot n)n, \quad (1)$$

where  $k^s$  is the penalty force stiffness,  $k^d$  is the damping coefficient,  $d$  is the penetrated distance,  $v$  is the velocity of the particle, and  $n$  is the normal vector of the face.

### 3.3 Erosion and Deposition

When the particle collides with the terrain, we simulate the erosion of the surface or the deposition of the sediment, depending on the velocity and the direction of the particle. We calculate the erosion rate using a method similar to the one presented in the work of Wojtan et al. (2007). The erosion rate is calculated using the equation by Partheniades (1965):

$$\varepsilon = k(\tau - \tau_c)^a, \quad (2)$$

where  $\varepsilon$  is the erosion rate,  $k$  is the erosion constant,  $\tau$  is the shear stress,  $\tau_c$  is the critical shear stress, and  $a$  is a constant usually set to 1. The critical shear stress  $\tau_c$  is a threshold value that has to be exceeded for the erosion to take place. The shear stress  $\tau$  is a force applied to the solid boundary and is defined via a power-law model (Wojtan et al. 2007):

$$\tau = K\theta^m, \quad (3)$$

where  $K = 1$  is a constant,  $\theta$  is the shear rate and  $m$  is a constant determined by the material. The shear rate  $\theta$  can be approximated from the fluid velocity relative to the surface:

$$\theta = \frac{v_{rel}}{d}, \quad (4)$$

where  $v_{rel}$  is the relative fluid velocity and  $d$  is the orthogonal distance between the particle  $p$  and the face  $f$ .

If the shear stress  $\tau$  exceeds the value of the critical shear stress  $\tau_c$ , the erosion rate  $\varepsilon$  has a positive value, and erosion takes place. The amount of the eroded sediment is uniformly divided among the vertices of the face  $f$ . Each vertex has a parameter assigned to store the amount of eroded (deposited) sediment, the contributions from individual particles are summed up and the mesh is only updated once in each iteration of the algorithm. The eroded sediment is carried by the particle until the conditions for the deposition are fulfilled.

The deposition occurs when the shear stress  $\tau$  does not exceed the value of the critical shear stress  $\tau_c$ . The erosion rate  $\varepsilon$  is negative and the sediment carried by the particle is deposited onto the vertices of the face  $f$ .

### 3.4 Mesh Modification

Each terrain vertex keeps track of the amount of sediment that was removed or added to it during the erosion/deposition step. Eventually, the mesh needs to be updated to reflect this change of volume.

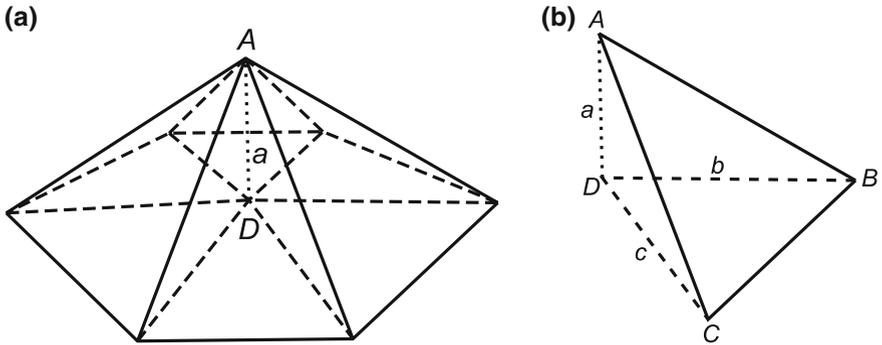
Each affected vertex needs to be moved in the direction of the normal vector according to the amount of associated sediment. However, we cannot use the amount of sediment to directly determine the vertex displacement. Each vertex belongs to several faces and these faces can generally differ greatly in size and shape. The same vertex displacement could thus lead to different volume changes in two parts of mesh with distinct resolution.

As we need to control the volume changes regardless of the local resolution of the scene, we have to calculate the vertex displacement based on the exact volume change. The total volume change caused by the displacement of a vertex is a sum of contributions of all the affected faces. Figure 2a shows an example of sediment deposition onto a vertex. The deposition caused the vertex to move from the position  $D$  to the position  $A$ , with the vertex displacement  $a$ . The volume change for one of the faces equals to the volume of the area bounded by the original face and the face created by displacing the vertex, this area always forms a tetrahedron (Fig. 2b).

The volume of a tetrahedron  $V_i$  can be calculated as follows:

$$V_i = \frac{|a \cdot (b \times c)|}{6}, \quad (5)$$

where  $a$ ,  $b$  and  $c$  are the edges of the tetrahedron sharing the vertex  $D$ . Using Eq. (5), we can get the total volume change  $V$  for the vertex displacement  $|a|$  as



**Fig. 2** The volume change caused by the displacement of a single vertex (a) and the volume change of one of the affected faces (b)

$$V = \sum_0^{n-1} V_i, \tag{6}$$

where  $n$  is the number of faces sharing the displaced vertex. It can be derived that for  $a' = ax$ , where  $x$  is a constant, the total volume  $V' = Vx$ .

To compute the correct vertex displacement  $a'$ , we first calculate the volume change  $V$  for the vertex displacement  $|a| = 1$  in the direction of the normal vector. The vertex displacement  $a$  needs to be adjusted so that the volume change is equal to the amount of eroded/deposited sediment  $s$ :

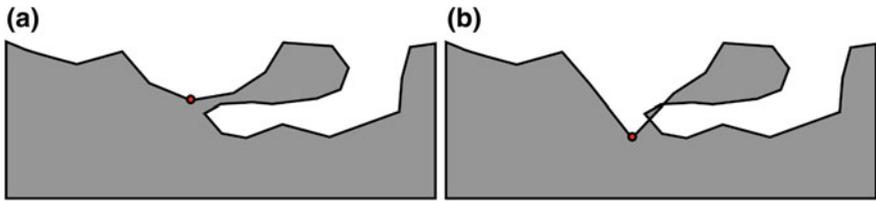
$$a' = a \frac{s}{V}, \tag{7}$$

where  $a'$  is the required vertex displacement.

### 3.5 Mesh Inconsistency

The erosion can cause an inconsistency in the mesh. If a part of the mesh is heavily eroded, it is possible for two parts of the mesh to overlap, creating a topology inconsistency. A simplified 2D example of such a case is shown in Fig. 3. Figure 3a shows a simple valid scene, the red point represents the vertex, where the erosion will be applied. The eroded scene in Fig. 3b shows an example of a created topology inconsistency. While erosion can cause creation of holes or breaking the mesh into pieces, deposition can cause unwanted mesh merging.

The repair of topology inconsistencies in the mesh is a well-known problem and many methods exist that address it, e.g., cf Attene et al. (2013). The problem is difficult due to the numerical inaccuracy of floating point arithmetic, which can



**Fig. 3** An example of a mesh inconsistency. **a** The original scene. **b** The eroded scene with an inconsistency

cause the algorithms to collapse. As the inconsistencies are created very rarely during the erosion simulation, we do not repair them. We detect the inconsistency when it is created, and stop the simulation.

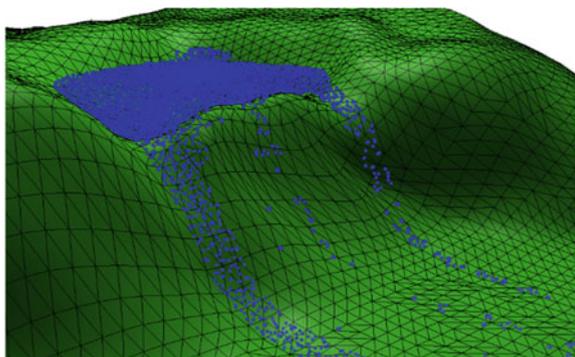
## 4 Discussion and Results

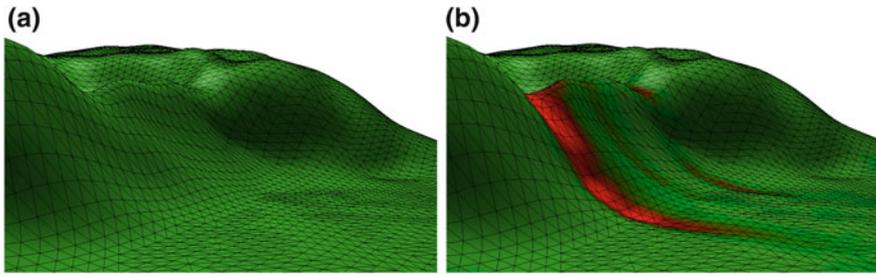
We have developed our framework in C++ and run it on an Intel Core 2 Duo clocked at 2 GHz. We use the *Fluids v.2* library (Hoetzlein 2009) for SPH fluid simulation that provides the necessary functionality and computes the particle properties during the simulation. The Navier-Stokes equations are scale-sensitive and the actual simulation scale of the fluid in the used implementation is approximately up to tens  $\text{cm}^3$ . As the terrain represents a large-scale scene, this inconsistency of scales can lead to less realistic behavior of the fluid.

We have verified the proposed algorithm by simulating several different erosion scenarios. Erosion is a long term process and a comparison with real-world data would be necessary to verify our algorithm. We rely the correctness on the computational fluid dynamics and we verified our solution visually.

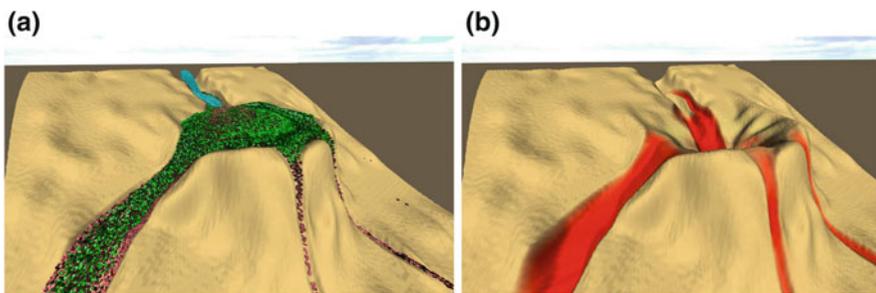
We created a simple scenery of a lake being filled by water to validate the erosion process and to compare our solution to the results of the reference method by Křištof et al. (2009). The scene is shown in Fig. 4. The lake represents a local

**Fig. 4** An example of an erosion simulation of an overflowing lake





**Fig. 5** An example of an erosion simulation. **a** The original terrain. **b** The eroded terrain; *red* color highlights the eroded regions



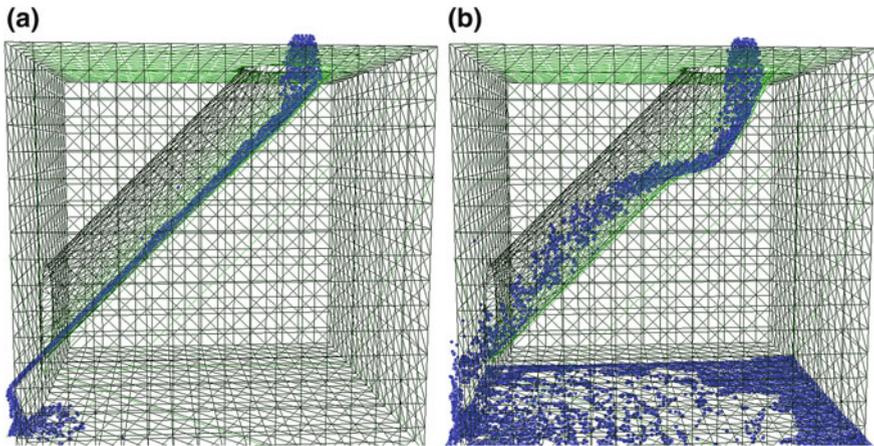
**Fig. 6** The results of the reference method (Krištof et al. 2009). **a** The fluid eroding the terrain. **b** The eroded terrain; *red* color highlights the eroded regions

depression, which is being filled with water particles. The lake eventually fills up and the fluid flows over the edge and erodes the underlying terrain surface. We use a simple particle-based visualization for the fluid, as we are interested in the resulting eroded terrain itself, not in the realistic image of the water.

Figure 5 captures the alteration of the terrain during the simulation. Figure 5a shows the original terrain before the erosion was applied. The final eroded scene is depicted in Fig. 5b with the eroded regions highlighted in red color. The results of the reference method are shown in Fig. 6.

Unlike the reference method, our method can also simulate the fully 3D scenes with concave features. We modeled the demonstration scene captured in Fig. 7a. The scene consists of a block of material with a tube-like cavity. The water runs through it and erodes its surface (Fig. 7b). The fluid has the largest momentum before it collides with the mesh for the first time, hence the erosion is the strongest at that point. The fluid flow is affected by the changed surface of the mesh (Fig. 7b). The full simulation of both presented examples can be seen in Skorkovská et al. (2014).

Computational requirements were measured for the presented simulations. The lake scene (Fig. 4), consists of 12,528 particles and 9,068 faces, we measured an execution time of one iteration of approximately 2.9 s. Each iteration of the concave



**Fig. 7** Erosion simulation of concave features. **a** The original scene. **b** The water erodes the mesh, leading to change of the flow of the fluid

scene simulation (Fig. 6), consisting of 12,528 particles and 4,706 faces, took approximately 0.7 s. The goal of our implementation was to verify the applicability of our approach, without the emphasis on the speed of the simulation.

## 5 Conclusion

A novel solution to the hydraulic erosion simulation problem was proposed in this paper. Our solution combines the terrain represented as a triangular mesh with a fluid simulated as a particle system. This approach is capable of simulating fully 3D scenes containing features such as caves or overhangs with lower memory requirements than the existing volumetric solutions.

The triangular mesh representation has an advantage over other approaches in its adaptability. The resolution of the scene can be adjusted according to the shape and complexity of the terrain. The resolution can be increased in the regions that require more attention and preserved in other parts of the mesh. However, the use of the triangular meshes brings new challenges during the erosion simulation. The erosion can cause a creation of an inconsistency in the mesh, which would cause an incorrect behavior of the simulation. These inconsistencies occur very rarely and we address them by stopping the simulation to prevent the collapse of the simulation.

There are many possible avenues to improve the proposed algorithm. The problem of mesh inconsistency has to be solved in order to be able to simulate erosion of more complex scenes. The implementation of multiple materials could be the next step to achieve more realistic results. Last but not least, the implementation could be optimized to speed up the simulation.

**Acknowledgments** We would like to thank to Jan Ježek (University of West Bohemia, Pilsen) for his valuable advice and comments. This work has been supported by the European Regional Development Fund (ERDF)—project NTIS (New Technologies for Information Society), European Centre of Excellence, CZ.1.05/1.1.00/0.2.0090 and by the project SGS-2013-029—Advanced Computing and Information Systems.

## References

- Acheson DJ (1990) The Navier-Stokes equations. In: Elementary fluid dynamics. Clarendon press, Oxford, pp. 201–220. ISBN 978-0-19-859679-0
- Adams B, Pauly M, Keiser R, Guibas LJ (2007) Adaptively sampled particle fluids. *ACM Trans Graph* 26(3):48:1–48:7
- Aksoy H, Kavvas ML (2005) A review of hillslope and watershed scale erosion and sediment transport models. *Catena* 64(2):247–271
- Amada T (2006) Real-time particle-based fluid simulation with rigid body interaction. In: Dickheiser M (ed) Game programming gems 6. Charles River Media, Rockland, pp 189–205
- Attene M, Campen M, Kobbelt L (2013) Polygon mesh repairing: an application perspective. *ACM Comput Surv* 45(2):15:1–15:33
- Benes B, Těšínský V, Hornýš J, Bhatia SK (2006) Hydraulic erosion. *Comput Animation Virtual Worlds* 17(2):99–108
- Bridson R (2008) The equations of fluids. In: Fluid simulation for computer graphics, Ak Peters series. Taylor & Francis, Boca Raton, pp. 3–16. ISBN 978-1-56881-326-4
- DeVantier BA, Feldman AD (1993) Review of GIS applications in hydrologic modeling. *J Water Resour Plann Manage* 119(2):246–261
- Gingold RA, Monaghan JJ (1977) Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon Not R Astron Soc* 181(3):375–389
- Hoetzlein R (2009) Fluids v.2—a fast, open source, fluid simulator. <http://www.rchoetzlein.com/eng/graphics/fluids.htm>. Accessed 15 Oct 2014
- Kipfer P, Westermann R (2006) Realistic and interactive simulation of rivers. In: Proceedings of graphics interface 2006. Canadian Information Processing Society, Toronto, ON, pp. 41–48
- Křištof P, Benes B, Křivánek J, Št'ava O (2009) Hydraulic erosion using smoothed particle hydrodynamics. *Comput Graph Forum* 28(2):219–228
- Partheniades E (1965) Erosion and deposition of cohesive soils. *J Hydraul Div Am Soc Agric Eng* 91:105–139
- Skorkovská V, Kolingerová I, Benes B (2014) Video example. [http://home.zcu.cz/~vskorkov/Hydraulic\\_Erosion/](http://home.zcu.cz/~vskorkov/Hydraulic_Erosion/). Accessed 15 Oct 2014
- Wojtan C, Carlson M, Mucha PJ, Turk G (2007) Animating corrosion and erosion. In: Proceedings of the third Eurographics conference on natural phenomena. Eurographics Association, pp. 15–22