UDC 519.688:532.5:551.4
DOI: 10.32626/2308-5916.2025-27.16-25

Oleksii Zholtovskyi,
Oleksandr Stanislaviv,
Olena Smalko, PhD

Kamianets-Podilskyi Ivan Ohiienko National University,
Kamianets-Podilskyi

PROCEDURAL GENERATION OF DIGITAL TERRAIN USING DISCRETE HYDRODYNAMIC MODELS

This article investigates discrete hydrodynamic models for advanced procedural generation of digital terrains, aiming to synthesize dynamic and geomorphologically plausible landscapes. Our approach proposes a novel hybrid method. It combines procedural fractal noise with layered discrete hydrodynamic modeling, providing a robust and realistic synthesis pipeline that significantly surpasses conventional static techniques.

Initially, fractal algorithms create a foundational base heightmap, establishing essential large-scale elevation variations. To enhance physics-based modeling plausibility, successive hydrodynamic simulations meticulously mimic water flow, erosion, sediment transport, and channel formation. Specifically, Euler's and shallow water equations model large-scale water dynamics and terrain evolution. For micro-scale refinement, Burgers' and Korteweg-de Vries equations are utilized for sedimentation processes, generating finer redistribution patterns and adding morphological complexity consistent with natural fluvial landscapes. A custom software application was developed to sequentially integrate these layers, enabling precise and iterative application of hydrodynamic effects. This ensures the generated topographical features are both accurate and aesthetically compelling.

Simulations demonstrate the proposed method generates heightfields with realistic geomorphology, while maintaining high computational efficiency facilitated by optimized numerical methods. This makes it well-suited for diverse applications: computer graphics, game development, virtual reality, and geographical education. Integrating discrete physics-based hydrodynamics into procedural noise synthesis fundamentally elevates the realism, coherence, and structural consistency of synthetic terrains, offering a versatile and effective tool for complex environmental representation.

Key words: *procedural terrain generation, discrete hydrodynamic models, partial differential equations, physics-based modeling, computer graphics.*

Introduction. The birth of procedural generation is frequently associated with the roguelike style of role-playing games of the 1970s. Early computer games featured trivial texture creation methods. All the textures were stored either as texture atlases or as images with specific scales, which significantly limited the realism of the graphics in the game scenes and prevented computers from generating more complex landscapes. However, procedural generation methods have been integrated to address the need for advanced generation of landscapes, textures, and worlds in video games, as well as for complex scene creation and object generation in animated films and videos. These methods are also used for developing visual informative elements (such as maps, animations, clouds, graphs, and diagrams), which simplify data analysis and overall understanding, and find applications in virtual reality and other diverse simulations.

Thanks to procedural generation algorithms based on mathematical structures, it becomes possible to repeatedly generate complex and versatile content using the same initial conditions, while saving computational resources and storage. The better and more natural the synthesis results are, the more useful they become for users and content owners.

Common methods for creating heightmaps. In the context of procedural landscape generation, a heightmap is a numerical n-dimensional array used to determine the values used by terrain generation algorithms. Heightmaps are often organized as simple two-dimensional arrays of numbers or more complex data structures containing additional information, such as texture coordinates, normal vectors, and other material properties.

There is a variety of heightmap generation algorithms used in terrain synthesis. In the case of chaotic terrains, matrix elements are generated pseudo-randomly, whereas more organic or natural structures are created using mathematical operations ranging from simple to complex. A notable example is the Perlin noise algorithm, which utilizes dot product and Hermite interpolation. Many variations of the diamond-square algorithm have been widely used in terrain generation for a long time and still find significant application. In particular, it is convenient for creating heightmaps serving as a geometric basis for the landscape as well as various terrain textures, cloud textures, etc.

In the second half of the 1980s, heightmap creation and combined procedural texture generation methods began to develop. This was pioneered by American computer scientist Ken Perlin with his mathematical algorithm for generating procedural texture by using a pseudo-random method in conjugation with mathematical operations [1], which helped to increase the realism and graphical complexity of the surfaces of various geometric objects. Variations of Perlin noise, such as ridged fractal noise, modifications of simplex noise, and cell noise, as well as their combinations, have become quite popular.

In the 2000s, procedural textures became even more widespread since they were generated using various algorithms on the fly, greatly expanding the possibilities for game developers, artists, and computer graphics engineers.

Over time, advancements in the generation of heightmaps started to be guided by the principles of geomorphology, resulting in the production of relief models that increasingly mirror natural landscapes.

As computing power gradually increases, so do computing capacities. Currently, they have become available to ordinary computer users, necessitating further improvement of procedural generation algorithms. There is an ongoing search for new effective terrain modeling methods that allow plausible synthesis of various elements of geophysically defined landforms, biome land cover equivalents, and anthropogenic objects.

It has also become apparent that using only traditional heightmaps somewhat limits the synthesized terrain to a continuous surface without caves or protrusions. This disadvantage can be partially resolved by using multilayer heightmaps or their nonlinear transformations [2].

Extended heightmaps may contain terrain height points (absolute or relative to some initial level) as well as other parameters such as terrain type (mountains, plains, water, forest, desert), soil or surface type at each point (hard stone, soft soil, sand, clay), and various terrain properties (temperature, humidity, light level). They may also include attributes like colors, textures, and references to objects such as trees, buildings, and roads. With such data, one can visualize detailed elements of computer game scenes or immersive environments, realizing the author's intent and balancing realism and computing performance.

Recently, novel approaches have emerged for heightmap creation. Artificial intelligence, particularly through generative deep learning and adversarial networks, offers promising solutions for many terrain generation needs, leveraging their widespread use in image generation to extend to landscape generation.

To generate terrain with artificial intelligence, a dataset comprising desired terrain parameters and examples is typically created to train the model. The model then learns the relationships between input and expected output data, enabling it to produce the desired type of landscape. Reinforcement learning methods can also be employed, involving gradual training of an agent until satisfactory generation accuracy is achieved. While training such agents demands significant computing resources (e.g., GPUs and tensor processing units), the potential results often justify the computational investment.

In this way, landscapes are generated according to the gameplay environment and other dynamic changes depending on the conditions and interaction with the player. This ensures full interactivity and adaptability.

Selected methods of computational hydrodynamics. Some methods for solving problems in numerical fluid mechanics appear quite promising for terrain synthesis. Among them, it is worth noting, for example, Smoothed Particle Hydrodynamics (a non-iterative method), grid-based approaches, and other relatively simple physics-based methods (which model terrain based on physical processes such as sedimentation, water erosion, etc.).

Naturally, all these physical processes must be significantly simplified for use in computer graphics and animation to reduce computational resource consumption. To this end, optimized approaches are employed to achieve an acceptable balance for specific types of tasks. For example, the resolution in the mesh-free Lagrangian Smoothed Particle Hydrodynamics method can be greatly simplified by conveniently varying the density parameter, based on the normalized fundamental matrix [3]. In this way, it remains possible to simulate effects that require a high level of detail, including individual particle tracking.

We should also not overlook important numerical methods for modeling hydrodynamic systems that rely on a regular grid for discretizing space (and time), namely the lattice Boltzmann methods. These methods allow for modeling macroscopic properties (such as flow velocity, temperature, pressure, etc.) of Newtonian fluid flow using the discrete Boltzmann kinetic equation. This is particularly suitable for implementing complex simulations.

Eulerian grid-based methods have been used for a long time. In particular, the marker-and-cell method [4] (which uses points in space) has proven effective for tracking fluid motion across a grid. During simulation, marker particles move through the velocity field and are then used to determine which cells contain fluid. Such grid-based methods remain quite popular in computer graphics for simulating fluid flow [5].

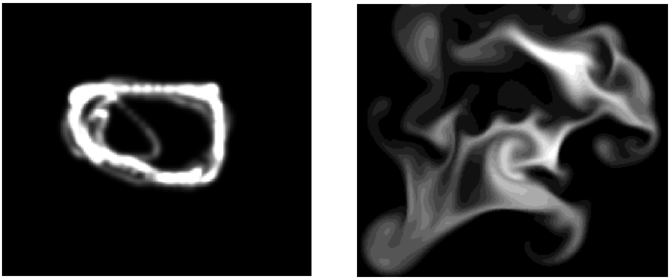
For the efficient synthesis of continuous environments in landscapes with large volumes of fluid-like substances, especially in the gaming and film industries, Eulerian-based approaches are commonly used. Accordingly, simulation models frequently employ grid-based solvers for the Navier–Stokes equations [6]. However, in striving for equilibrium – especially in complex scenes that require both detailed and large-scale modeling – hybrid methods must often be applied. These combine both particle-based and grid-based simulations [7]. For example, fluid surfaces might be represented using particles, while the bulk flow is computed on a grid. However, this approach requires substantial computational resources.

If the simulation assumes an incompressible and inviscid fluid (with constant density in both time and space), the two-dimensional Navier–Stokes equations – which are well suited for generating realistic landscapes (Fig. 1) with relatively low computational complexity – reduce to the Euler equation (1):

$$\begin{cases} \dfrac{\partial \bar{u}}{\partial t} + \bar{u}\dfrac{\partial \bar{u}}{\partial x} + \bar{v}\dfrac{\partial \bar{u}}{\partial y} = -\dfrac{1}{\sigma}\dfrac{\partial \bar{p}}{\partial x}; \\ \dfrac{\partial \bar{v}}{\partial t} + \bar{u}\dfrac{\partial \bar{v}}{\partial x} + \bar{v}\dfrac{\partial \bar{v}}{\partial y} = -\dfrac{1}{\sigma}\dfrac{\partial \bar{p}}{\partial y}. \end{cases} \quad (1)$$

Here, $\bar{u}(x,y,t)$ and $\bar{v}(x,y,t)$ are the velocity components in two dimensions (*x* and *y*) dependent on time *t*, σ is the liquid density, and

$\overline{p}(x,y,t)$ is the average pressure, also dependent on space and time. Incompressibility condition: $\frac{\partial \overline{u}}{\partial x}+\frac{\partial \overline{v}}{\partial y}=0.$



a *b*

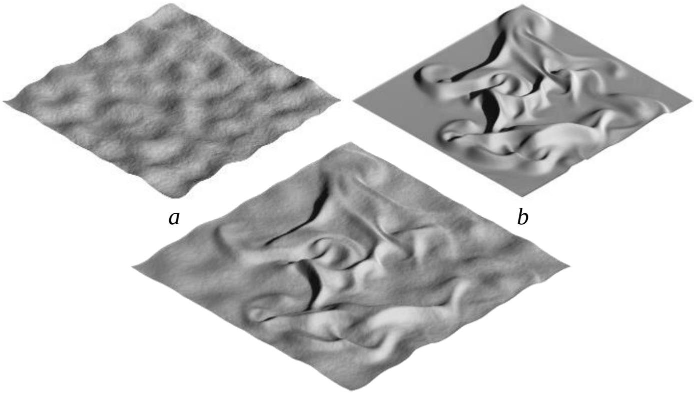
Fig. 1. *Illustration of a group of selected ideal fluid particles:*
a – initial conditions for the group of selected particles;
b – result of the evolution of the group of selected particles over time t

In our work, we propose the following steps for generating realistic terrains. First, an initial heightmap (grid) with dimensions $n \times m$ is generated using, for example, Perlin noise (Fig. 2a). Other types of noise, algorithms, or their combinations can also be used. Next, a matrix of the same dimensions is generated to store data about initial conditions such as velocity, density of selected particles, etc. Selected particles are a group of fluid particles whose density distribution is used to generate a heightmap influenced by a velocity field – this field being the solution to the Euler equation. At each point, the initial values of the velocity components u and v, as well as the pressure p, are determined. The finite difference method, with the Courant–Friedrichs–Lewy condition, is used to numerically solve the equations at each grid point. Over time t, this matrix will contain the evolving solution.

The numerical interpretation of the two-dimensional solutions (Fig. 2b) is then superimposed on the primary heightmap (Fig. 2c) in the required manner, using specific user-defined parameters.

One possible way to interpret this data involves treating the velocity field vectors and their dot product with an arbitrary unit vector as a heightmap, which can be further used for blending. The method we have chosen involves determining the density of selected particles at each point of the grid. This particle density is then interpreted as a heightmap. The illustration below demonstrates this approach (Fig. 2).

If necessary, the obtained result can be combined with other heightmaps to produce more complex and detailed landscapes.



a *b* *c*

Fig. 2. *Illustration of blending the primary heightmap with the numerical representation of the two-dimensional motion of an ideal fluid:*
a – heightmap generated using a noise function;
b – surface rendering derived from the numerical solution of the 2D incompressible Euler equations; c – resulting digital surface model

Among hydrodynamic phenomena, quasi-one-dimensional flows are of particular interest. These occur in long channels or rivers with a width much smaller than their length. The governing equations for such flows are known as the shallow water equations, or Saint-Venant equations (2). They are also applied to fluid motion in bays, coastal zones, and other shallow environments, and have found application in computer graphics.

The continuous shallow water equation has form (2):

$$\frac{\partial^2 h}{\partial t^2} = gd\,\frac{\partial^2 h}{\partial x^2}, \tag{2}$$

where g is the gravitational constant, $h(x)$ is the water surface level, and $d(x)$ is the water depth. At the same time, the relation $h(x) = d(x) + b(x)$, where $b(x)$ represents the bottom elevation [8].

In particular, the shallow water equations can be used to model erosion processes caused by water-induced destruction of soil or rock.

From a geomorphological perspective, erosion phenomena include sheet and rill erosion of hill slopes, gully erosion along watercourses, coastal erosion driven by waves, tides or storms, rock denudation accompanied by fragment displacement under combined impacts on the regolith, tunnel erosion resulting from water infiltration through decayed root channels, and stream bank erosion induced by various natural factors.

Taking into account the specifics of these processes allows for the generation of digital landscapes that closely resemble real-world terrains.

We have explored terrain generation using the shallow water equations via a simplified erosion model based on three key stages: soil disintegration, material transport, and deposition. This approach is sufficient for synthesizing landscapes with floodplains, valleys, watercourses, and even ravines, provided a well-designed algorithm is implemented.

For generating more organic and natural-looking structures suitable for games or immersive environments, the Bateman–Burgers equation may be applied. This equation captures various fluid dynamics phenomena, such as flow in rivers, seas, oceans, and wave behavior [9]. It enables the simulation of unstable terrains subject to continuous transformation due to tectonic activity, wind, or water influence, leading to the creation of unique and complex landscapes. However, modeling multi-scale relief features with diverse detail levels requires substantial computational resources.

Under the assumption of a fluid with water-like density and low viscosity, we apply the inviscid Burgers equation. This form is obtained by eliminating the viscosity-related terms from the original equation. The resulting partial differential equations are given in equation (3):

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0, \tag{3}$$

where x is the spatial coordinate, t is time, $u(x,t)$ is the velocity field as a function of position x and time t, and $u\frac{\partial u}{\partial x}$ is the advection term describing the convective fluid flow. This equation is a quasi-linear hyperbolic equation of the first order. It is well suited for modeling simple wave processes or linear landscape features (e.g., rivers or mountain ranges).

For more dynamic terrain features, one can experiment with a unique type of wave known as solitons. A soliton is a structurally stable solitary wave that propagates freely at a constant speed; it retains its shape and is not destroyed by other excitations. They can be described, for example, by the Korteweg–de Vries equation (4)

$$\frac{\partial u}{\partial t} + au\frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial x^3} = 0,\ \alpha \neq 0, \tag{4}$$

where x is the spatial coordinate, t is time, and $u(x,t)$ is a one-dimensional real-valued function describing the wave profile (e.g., water height).

In the software application developed within the framework of this research for generating hydrodynamic models in digital landscapes, we limited ourselves to the described equations (1)-(4), which we consider computationally efficient.

Dedicated terrain generation application. To implement the investigated methods and effectively visualize terrain generation techniques based on simplified hydrodynamic models, a software application was developed. This application is useful for those learning the art of video game development, aspiring to work as graphic designers in game studios, or aiming for positions as graphics software engineers in prestigious companies.

Even without being highly skilled 3D modeling specialists or possessing the talents of advanced computer artists, users can initially learn to use this application for procedural terrain generation, simultaneously studying the methodology of digital landscape development, thoroughly exploring the mathematical foundations of the process, and eventually refining existing approaches.

The developed application traditionally uses a heightmap – a two-dimensional array in which each element corresponds to a specific location on the terrain grid and stores a value representing the height at that point. Initialization of the heightmap is carried out by combining several noise functions with different scales and amplitudes. Other traditional algorithms, such as the diamond-square algorithm, can also be applied for this purpose. By wisely combining noise functions with possible addition of an overall offset and scaling factor, as well as applying various refinements and modifications of the diamond-square algorithm, it is possible to achieve the desired level of detail and significant variability in the overall terrain shape.

To give the resulting intermediate terrain a smoother, more organic appearance, a smoothing procedure can be applied using averaging methods or Gaussian convolution algorithms, which also significantly reduce computational complexity.

The most interesting and responsible part of the development was the implementation of layers with graphical visualization of numerical solutions of selected two-dimensional hydrodynamic equations. Their superimposition, in a specific order, on the layers with noise functions usually added realistic large-scale structures, such as slopes, ravines, basins, and other complex and diverse terrain formations to the rendering result, simulating natural hydrodynamic processes (Fig. 3).

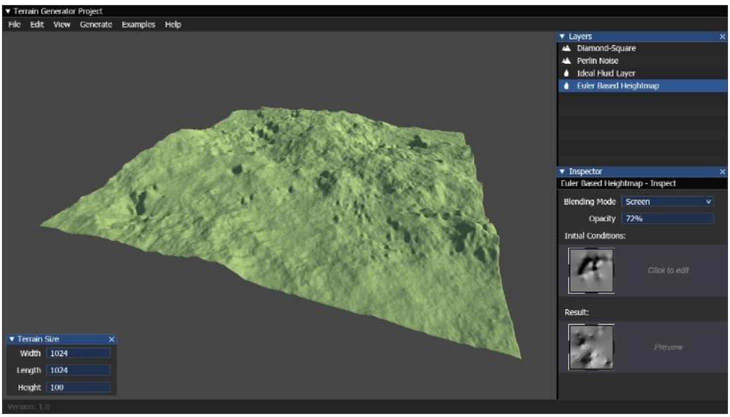


Fig. 3. *Screenshot of the workspace window with visualized geomorphological features of the generated terrain*

The application programmatically implements an operation control structure that allows users to control the selection of parameters such as the number of layers, their scales, the amplitudes of the used noises, the strength of filtering effects, and more. The preview of the generated terrain models is rendered in real time. This allows users to see the immediate impact of parameter changes and significantly facilitates the iterative design process. The program offers predefined (default) parameters for common terrain types (including hilly landscapes, rocky areas, water basins, plateaus). There are also capabilities to save/load custom terrain configurations, combine various textures, and export digital content in the FBX format.

Conclusion. Our investigations confirm that discrete hydrodynamic models provide a highly effective approach for procedural landscape generation, enabling the synthesis of dynamic and geomorphologically plausible digital terrains. The developed methodology, integrating physics-based modeling derived from partial differential equations and implemented through numerical methods, demonstrates significant potential for enhancing realism in terrain generation. A dedicated software application has also been developed to illustrate and facilitate the practical application of these hydrodynamic models in computer graphics and educational settings.

Future developments in this area could focus on developing more effective computational approaches for even greater computational efficiency, incorporating more complex fluid dynamics, such as turbulence models, to broaden the range of achievable landscape features, and potentially extending the software's functionality for direct integration with game engines and augmented reality platforms to create more immersive and interactive virtual environments.

References:

1. Perlin K. An image synthesizer. *ACM SIGGRAPH Computer Graphics*. 1985. Vol. 19. № 3. P. 287-296.
2. Stanislaviv O. S. Digital representation of landscape using procedural generation methods. *Bulletin of Kamianets-Podilskyi Ivan Ohiienko National University. Series: Physical and Mathematical Sciences*. 2023. Issue 16. P. 79-84.
3. Vieira-e-Silva A. L., Brito C., Almeida M. W., Teichrieb V. Improved MPS method and its variations for simulating incompressible fluids on GPU. *SBC Journal on Interactive Systems*. 2018. Vol. 9. № 2. P. 52-67.
4. Harlow F., Welch J. Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *The Physics of Fluids*. 1965. Vol. 8. № 12. P. 2182-2189.
5. Osher S., Fedkiw R. Level Set Methods and Dynamic Implicit Surfaces. *Applied Mathematical Sciences*. 2003. Vol. 153. P. 210-211.
6. Bridson R. Fluid Simulation for Computer Graphics. 2nd ed. Boca Raton, FL: CRC Press, 2016. 250 p.
7. Gagniere S., Hyde D., Marquez-Razon A., Jiang C. et al. A hybrid Lagrangian/Eulerian collocated advection and projection method for fluid simulation. *Computer Graphics Forum*. 2020. Vol. 39. № 8. P. 1-14.

8. Beneš B. Real-time erosion using shallow water simulation. *4th Workshop in Virtual Reality Interactions and Physical Simulation «VRIPHYS»*. The Eurographics Association, 2007. DOI: 10.2312/PE/vriphys/vriphys07/043-050.
9. Anderson D. A., Tannehill J. C., Pletcher R. H., Munipalli R., Shankar V. Computational Fluid Mechanics and Heat Transfer (Computational and Physical Processes in Mechanics and Thermal Sciences). 4th ed. Boca Raton, FL: CRC Press, 2021. 944 p.

ПРОЦЕДУРНА ГЕНЕРАЦІЯ ЦИФРОВОГО РЕЛЬЄФУ ЗА ДОПОМОГОЮ ДИСКРЕТНИХ ГІДРОДИНАМІЧНИХ МОДЕЛЕЙ

У статті досліджуються дискретні гідродинамічні моделі для вдосконаленої процедурної генерації цифрових рельєфів для синтезу динамічних та геоморфологічно правдоподібних ландшафтів. Наш підхід пропонує новий гібридний метод. Він поєднує процедурний фрактальний шум з пошаровим дискретним гідродинамічним моделюванням, забезпечуючи надійний і реалістичний конвеєр синтезу, що значно перевершує традиційні статичні підходи.

Спочатку за допомогою фрактальних алгоритмів створюється базова карта висот рельєфу, що формує суттєві великомасштабні варіації висот. Для підвищення правдоподібності фізично обґрунтованого моделювання послідовні гідродинамічні симуляції ретельно імітують потік води, ерозію, транспортування осаду та формування русел. Зокрема, рівняння Ейлера та рівняння мілкої води моделюють великомасштабну динаміку води та еволюцію рельєфу. Для мікромасштабного уточнення процесів седиментації застосовуються рівняння Бюргерса та Кортевега–де Фріза. Це дозволяє генерувати дрібніші структури перерозподілу та формувати морфологічну складність, характерну для природних флювіальних ландшафтів. Розроблений програмний додаток забезпечує точне й ітераційне застосування моделювання, що сприяє формуванню топографічних особливостей з високою точністю та естетичною привабливістю.

Проведені симуляції демонструють, що запропонований метод генерує поля висот з реалістичною геоморфологією, зберігаючи при цьому обчислювальну ефективність, що забезпечується чисельними методами. Це робить його придатним для різноманітних застосувань: комп'ютерної графіки, розробки ігор, віртуальної реальності та географічної освіти. Інтеграція дискретної фізично обґрунтованої гідродинаміки в процедурний синтез шуму фундаментально підвищує реалізм, узгодженість та структурну послідовність синтетичних рельєфів, пропонуючи універсальний та ефективний інструмент для складного представлення навколишнього середовища.

Ключові слова: *процедурна генерація рельєфу, дискретні гідродинамічні моделі, диференціальні рівняння в частинних похідних, фізично обґрунтоване моделювання, комп'ютерна графіка*.

Отримано: 11.06.2025