Generation techniques for realistic landscape informational models in immersive systems

Oleksandr Stanislaviv[1,†], Oleksii Zholtovskyi[1,†], Olena Smalko[1,*,†] and Bakhytzhan Akhmetov[2,†]

[1]Kamianets-Podilskyi Ivan Ohiienko National University, Ohienko Str., 61, Kamianets-Podilskyi, 32300, Ukraine
[2]Abai Kazakh National Pedagogical University, Dostyk Ave.,13, Almaty, 050010, Kazakhstan

Abstract

The study of ways to build realistic models of digital landscapes with rational parameters of computational complexity is a relevant area of research, the results of which can be implemented in the work of video game developers, graphic software engineers, virtual space designers, etc. We have shown that using rational methods for solving simplified forms of complex equations that describe some physics-based natural processes, we can effectively generate landscapes with additional realism. An application has been developed that generates terrain in a visual and intuitive way by integrating advanced mathematical models with computational methods to create realistic and efficient landscape information models.

Keywords

information model, digital landscape model, procedural generation, realistic landscapes, physics-based methods, design of virtual space, immersive system

1. Introduction

With rapidly expanding application scope and computational capacities of multimedia technologies and three-dimensional graphics solutions in many life aspects, these become an integral basis for a vast set of fields, such as immersive systems and augmented reality.

Hence, an exploration of various unique methods and techniques for digital landscape generation stands relevant to the current state of the art.

The findings of this research can be applied in the projects of video game developers, graphic software engineers, virtual space designers, and other related fields.

2. Main part

2.1. Basic procedural landscape generation and configuration

Traditionally, a common way to represent a terrain is heightmap. A heightmap is a two-dimensional array where element indices denote the location of a point on a discrete grid, and element value stores an elevation of the point above the surface and optionally other attributes, such as material type, and other metadata [1].

The level of terrain detail is determined by the size of the heightmap. Higher resolution provides finer details but requires more storage memory and potentially more computational power, which depends on heightmap implementation details.

CH&CMiGIN'25: Fourth International Conference on Cyber Hygiene & Conflict Management in Global Information Networks, June 20–22, 2025, Kyiv, Ukraine

*Corresponding author.
†These authors contributed equally.
stanislaviv.oleksandr@ukr.net (O. Stanislaviv); olekszholt@gmail.com (O. Zholtovskyi); oasmalko@gmail.com (O. Smalko); bakhytzhan.akhmetov.54@gmail.com (B. Akhmetov)
0009-0007-6160-0973 (O. Stanislaviv); 0009-0009-9245-1725 (O. Zholtovskyi); 0000-0001-7093-291X (O. Smalko); 0000-0001-5622-2233 (B. Akhmetov)

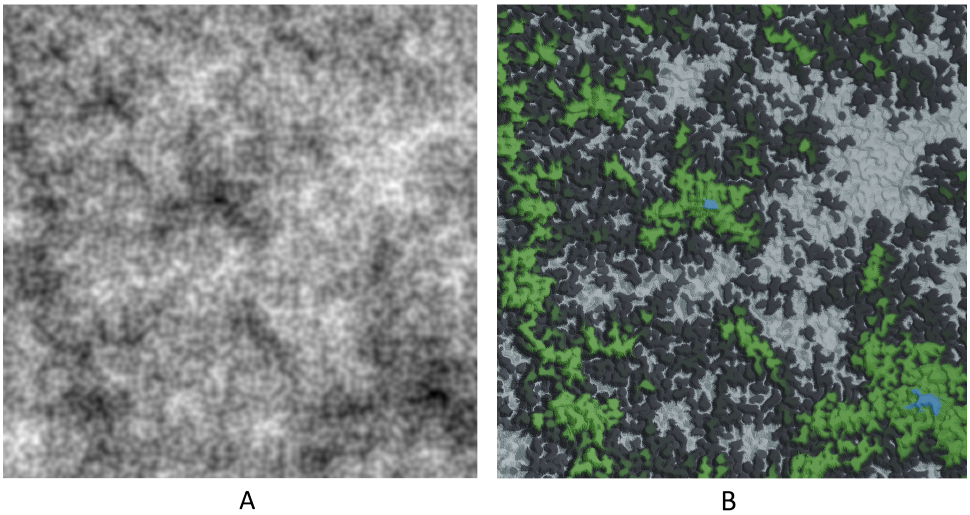

Figure 1: Example of generated terrain:
A – visualization of the heightmap generated by combining Perlin noise (in various octaves) and the erosion modeling algorithm.
B – interpretation of corresponding heightmap.

A typical routine for creating a heightmap matrix involves combining multiple noise functions with varying frequencies and amplitudes to create realistic terrain, enhance the variety of relief forms, and achieve a more organic appearance. This method is also used to generate other natural elements like clouds and water surfaces. Different filtering techniques are applied, as described below.

- Anti-aliasing. This process reduces high-frequency noise, creating a smoother overall relief. This is achieved by methods such as averaging the elevation values of neighboring points on the heightmap. Convolution filters with specific kernels are applied, i.e. each height value is changed to a weighted average of its surrounding values. One may control the strength of the smoothing effect by setting the kernel size and weighting coefficients assigned to neighboring points.
- Erosion. By simulating the effects of wind and water flow, erosion algorithms carve out valleys and sculpt ridges. This often involves using a few noise functions with different parameters (Figure 1). Lower frequency noise functions help form wider erosion patterns. In contrast to higher frequencies add details such as gullies and rills. Realistic erosion effects can be achieved by strategically adjusting noise parameters and combining them with anti-aliasing filters.
- Fault lines. These algorithms create sudden changes in elevation by simulating the movements of tectonic plates and forming features such as cliffs or canyons. One way to achieve this is by introducing special noise functions or mathematical operations that manipulate the heightmap along defined lines or regions. The size, direction, and depth of the fault lines can be changed.

The authors consider the best technique for creating realistic landscape forms is to imitate natural processes by using mathematical and physical models as a basis for digital landscape generation techniques while keeping the balance between computational and memory resources.

2.2. Landscape generation methods investigated

The increasing computational resources of modern computer systems have made it possible to use immersive technologies in various fields, enabling users to immerse themselves in artificially created interactive environments aimed at enhancing their experience.

In modern immersive systems, incorporating realism is sometimes crucial for some tasks, especially in the VR industry, to convey the feel and beauty of natural formations and environments. Take, for example, virtual touring in historical eras, experiencing the sensation of flying in a helicopter simulator over breathtaking natural landmarks, or actively participating in geological exploration during a VR journey to renowned canyons, underground caves, deep sinkholes, or the highest mountain ranges on Earth.

Based on the understanding that the earth's landforms have been formed over millennia as a result of various natural phenomena, we decided to investigate how their various mathematical models affected the realism of the created digital landscapes.

It is known that among the variety of phenomena inherent in the Earth's nature, the tectonics of lithospheric plates, which triggers the processes of spreading, subduction, or collision; sedimentation, which results in the formation of various types of sludge or deposit; various erosion phenomena caused by water, wind, ice, etc., as well as the creep of materials, which causes deformation of solids, have a significant and relatively predictable impact on the surface relief.

Curiously enough, hydrodynamic models are appropriate for landscape generation, notably in producing unique hill and ridge patterns. One such model is the Navier–Stokes equations, which describe the motion of a fluid. Their solutions are applicable in intermediate phases of heightmap generation with naturalistic forms [3]. However, solving these equations leads to excessive use of computational resources.

This problem can be partially mitigated by applying certain simplifications [4], in particular, by reducing the Navier–Stokes equations to the Euler equations (1):

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\cdot\nabla)\mathbf{v}\right) = -\nabla p + \mathbf{f}, \tag{1}$$

where:

- ρ is the fluid density, representing the mass per unit volume [kg/m^3]
- $\mathbf{v}$ is the velocity vector field, indicating the fluid's speed and direction at each point [m/s]
- $\frac{\partial \mathbf{v}}{\partial t}$ is the local (or temporal) acceleration, representing the rate of change of velocity at a fixed spatial point [m/s^2]
- $(\mathbf{v}\cdot\nabla)\mathbf{v}$ is the convective (or advective) acceleration, accounting for changes in velocity due to fluid motion through space [m/s^2]
- $-\nabla p$ is the pressure gradient force, where p is the scalar pressure field [Pa]
- $\mathbf{f}$ is the external body force per unit volume (e.g., gravitational force $\rho\mathbf{g}$) [N/m^3]

Based on the modeling of incompressible inviscid fluid and using Euler's equations to create an information model, large structures of realistic terrain can be synthesized.

To illustrate this, we used the well-known Marker and Cell (MAC) method, developed in the early 1960s at Los Alamos Laboratory and later improved in 2007 (Figure 2a). Enhancements include automatic time-stepping, the use of the conjugate gradient method to solve the Poisson equation for velocity correction, improved efficiency by removing virtual particles (markers) not near the free surface, more accurate boundary condition approximations at the free surface, and the addition of bounded high-accuracy upwinding for convective terms—allowing for the simulation of higher Reynolds number flows [5].

Smaller geomorphic and erosional landforms can be generated using simplified Navier–Stokes equations adapted to specific conditions, such as river or canal hydraulics, where the flow width greatly exceeds its depth. These flows are described by the shallow water equations [3], or, more precisely, by generalized forms of the Saint-Venant equations (2).

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} = \mathbf{S}(\mathbf{U}), \tag{2}$$

where:

- $\mathbf{U} = (h,\, hu,\, hv)^T$ is the state vector, with:

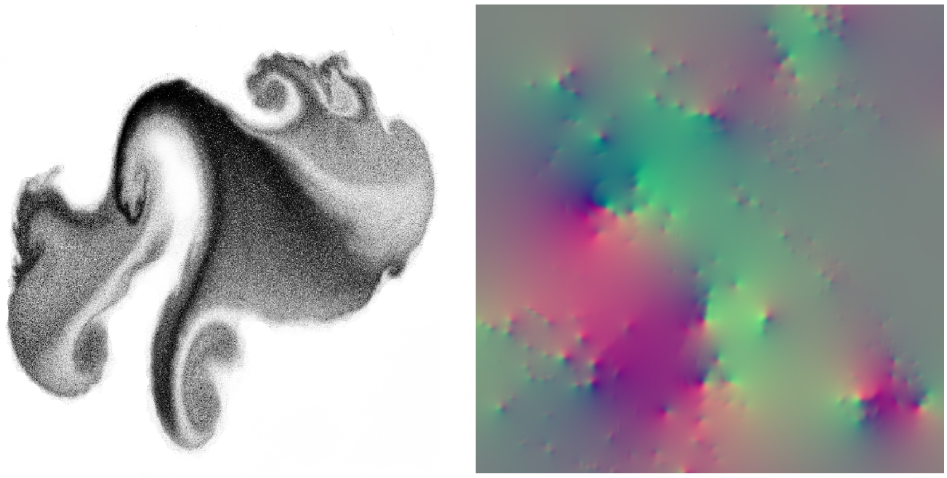

Figure 2: An interpretation of ideal fluid motion equation solution.
A – a density of highlighted fluid particles that move along the velocity field.
B – velocity field in RGB color domain.

  - h — water depth [m],
  - hu — momentum in the x-direction [m^2/s],
  - hv — momentum in the y-direction [m^2/s].
- $\mathbf{F}(\mathbf{U})$ and $\mathbf{G}(\mathbf{U})$ are the flux vectors in the x and y directions, respectively.
- $\mathbf{S}(\mathbf{U})$ is the source term vector accounting for external forces such as gravity and bottom friction.

The flux vector $\mathbf{F}(\mathbf{U})$ in the x-direction (3) is given by:

$$\mathbf{F}(\mathbf{U}) = \begin{pmatrix} hu & hu^2 + \frac{1}{2}gh^2 & huv \end{pmatrix}, \tag{3}$$

where:

- hu is the mass flux in the x-direction,
- hu^2 is the advective flux of x-momentum,
- $\frac{1}{2}gh^2$ is the hydrostatic pressure contribution,
- huv is the flux of y-momentum in the x-direction due to velocity coupling.

Similarly, the flux vector $\mathbf{G}(\mathbf{U})$ in the y-direction (4) is:

$$\mathbf{G}(\mathbf{U}) = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}, \tag{4}$$

with similar interpretations:

- hv is the mass flux in the y-direction,
- hv^2 is the advective momentum flux in the y-direction,
- $\frac{1}{2}gh^2$ again denotes the hydrostatic pressure,
- huv is the cross-component momentum flux.

The source term vector $\mathbf{S}(\mathbf{U})$ (5) is given by:

$$\mathbf{S}(\mathbf{U}) = \begin{pmatrix} 0 \\ gh(S_0 - S_f^x) \\ gh(S_0 - S_f^y) \end{pmatrix}, \tag{5}$$

where S_0 is the bed slope, and S_f^x, S_f^y are the friction slopes in the x- and y-directions, respectively.

By modeling simple wave processes, for example, with the quasi-linear hyperbolic Hopf equation (6), it is well-known [4]:

$$u_t + uu_x = 0, \tag{6}$$

where $u(x,t)$ is the unknown function that depends on the spatial variable x and time t, u_t is the partial derivative of u with respect to time t, and u_x is the partial derivative of u with respect to the spatial variable x.

This equation describes the dynamics of wave propagation possible to give natural outlines to rivers or mountain ranges in a medium without dissipation. One of its key characteristics is that solutions to this equation can develop discontinuities (known as shock waves) even if the initial condition $u(x,0)$ is smooth. This phenomenon occurs because, in a quasilinear system, the wave speed depends on the function u. As a result, different parts of the wave can travel at different speeds, leading to the steepening of wavefronts and eventually causing them to break, forming a discontinuity.

Shock waves are waves that suddenly increase their amplitude and propagate rapidly, creating steep fronts that move rapidly through the environment. This happens as a result of a sudden increase in the wave gradient (e.g., due to a sharp change in velocity or pressure).

Shock waves are also described by the hyperbolic Korteweg-de Vries equation (7):

$$u_t + u_{xxx} + 6uu_x = 0, \tag{7}$$

where $u = u(x,t)$ is a function of space and time, and subscripts denote partial derivatives.

The rational use of shock waves or other dynamic changes in terrain modeling occurs after identifying the most rational points on the heightmap (e.g., local maxima or minima). We identify them using the Sobel filter, which is very effective at detecting edges in height data, and this allows us to identify important geomorphological features. Also, the Sobel filter is quite simple to implement and requires few computational resources, which makes it very advantageous for procedural generation.

The Korteweg-de Vries equation, as well as other similar equations that have soliton solutions, can also be useful for creating realistic landforms. Solitons have an unexpected characteristic: they are stable, meaning that they retain their shape over time and distance, instead of dispersing or dissipating.

Dispersive solitons (traveling wave solitons) are stable localized wave packets that arise due to the balance between nonlinearity and dispersion in the medium. Nonlinearity implies that the waves change amplitude and shape. Dispersion, in turn, is a phenomenon in which different wave frequencies propagate at different speeds.

In two-dimensional space, solitons can be described by solutions of the Kadomtsev-Petviashvili equation (8)—a generalization of the Korteweg–de Vries equation that accounts for wave interactions in two-dimensional space [6]:

$$\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial t} + 6u\frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial x^3}\right) + 3\sigma\frac{\partial^2 u}{\partial y^2} = 0, \tag{8}$$

where u represents the wave profile, and x and y are spatial dimensions. The inclusion of the second derivative with respect to y allows for modeling wave propagation in two dimensions.

Solitons in two-dimensional shallow water are a specific type of 2D solitons, often referred to as "spider waves" because of their characteristic shape and dynamics, which resemble a stretching or oscillating spider web.

Among the variety of other soliton solutions, X-type and Y-type solitons can be useful for generating realistic landscapes. They are resonant because they result from the resonant interaction of two or more wave structures. X-type solitons can be created when two or more solitons intersect at acute angles to form an “X” pattern. For the landscape, these structures can be used to simulate the intersection or junction of different topographic structures, such as gorges or ridge confluences. The Y-type soliton got its name because of its characteristic shape, which looks like the letter “Y” – two waves converge and resonate into one structure, then diverge again. The “Y” shape can help, for example, in modeling the branches of river systems or other dynamic patterns. If necessary, multisolitons can be obtained by superposition of several single solitons with different parameters (amplitudes, phases, or velocities).

Other noteworthy waves are kink waves, which are characterized by a variable shape, often with soft transitions between different amplitudes, and breath waves, which are characterized by their amplitude changing in time, usually with periodic pulsations. These types of waves can be described by certain modified Korteweg-de Vries equations and the Benjamin-Ono equation, which describes one-dimensional internal waves in deep water. The use of mathematical structures with kink wave solutions in landscape generation enables the creation of terrain resembling natural wave-like formations, such as dunes or sea swells over large areas. Terrain generated using the Benjamin–Ono equation may also resemble long sand ridges or erosion valleys, but with more asymmetric and irregular shapes — better reflecting natural landforming processes driven by water or wind.

In case of generating landscapes that imitate large plains or undulating hills with gradual changes in elevation, characteristic of natural processes associated with water erosion or sedimentation, the Benjamin-Bona-Mahony equation (or regularized long-wave equation) may be of interest, which is characterized by the fact that it models long surface gravity waves of low amplitude.

In our work, we have also paid special attention to peakon solitons (peakons). They have steep edges with sharp peaks. This allows us to add distinct, sharp relief structures to the landscape, such as craggy cliffs, steeps or mountain peaks. At the same time, the use of peakons provided us with smooth transitions between landscape elements, helping to ensure smooth transitions between different elevation levels.

By using the Camassa–Holm equation (9), which describes the dynamics of shallow water waves and has peakon solutions, it was possible to model waves with sharp peaks that maintain their shape during propagation and interaction with other peakons [7]:

$$u_t + 2\mathrm{k}u_x - u_{xxt} + 3uu_x = 2u_x u_{xx} + uu_{xxx}, \tag{9}$$

where $u = u(x,t)$ is the function describing the wave profile depending on the spatial coordinate x and time t; the time derivative u_t describes the change of the wave over time; the inertia term u_{xxt} accounts for inertial effects and higher-order dispersion; the nonlinear term $3uu_x$ represents the system's nonlinearity, where the wave speed depends on its amplitude; the additional nonlinear-dispersive terms $2u_x u_{xx} + uu_{xxx}$ describe the interaction between nonlinearity and dispersion; and k is the spatial frequency of a wave, measured in cycles per unit distance (wavenumber).

In general, the use of peakons adds considerable dynamism to landscapes with contrasts between sharp peaks and deep hollows. The terrain may include mountainous regions with steep slopes or canyons, where abrupt changes in height are clearly visible. But by controlling the parameters of the peakons (amplitude, wavelength, and speed), it becomes possible to change the appearance of the generated shapes to match the desired level of landscape detail (Figure 3).

In order to solve the Camassa–Holm equation efficiently, we employed discontinuous Galerkin methods, which combine features of classical Galerkin techniques and finite difference schemes [3]. These methods divide the computational domain into elements, within which the solution is approximated by local polynomials. A special numerical flux calculated on the boundaries of the elements was used to ensure the connection between the elements. This allowed us to accurately track the behavior of discontinuous or nonlinear solutions. However, in order to reduce computational costs, we had to build optimized algorithms that implemented compromise solutions based on adaptive mesh refinement techniques, where the mesh resolution adapts dynamically to the solution's local complexity.

Currently, we are conducting research on the rationality of using macroscopic-level models of geomorphological processes in landscape generation tasks, as well as studying the peculiarities of using models for multiple soliton solutions.

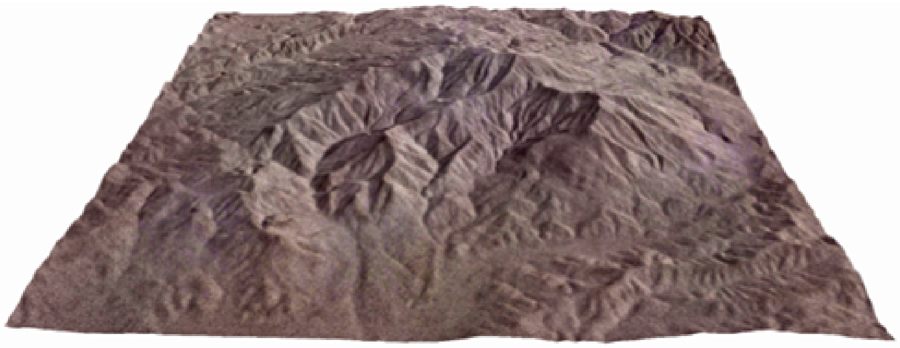

Figure 3: Example of terrain rendered by a complex model with an emphasis on distinct mountain peaks in the central area

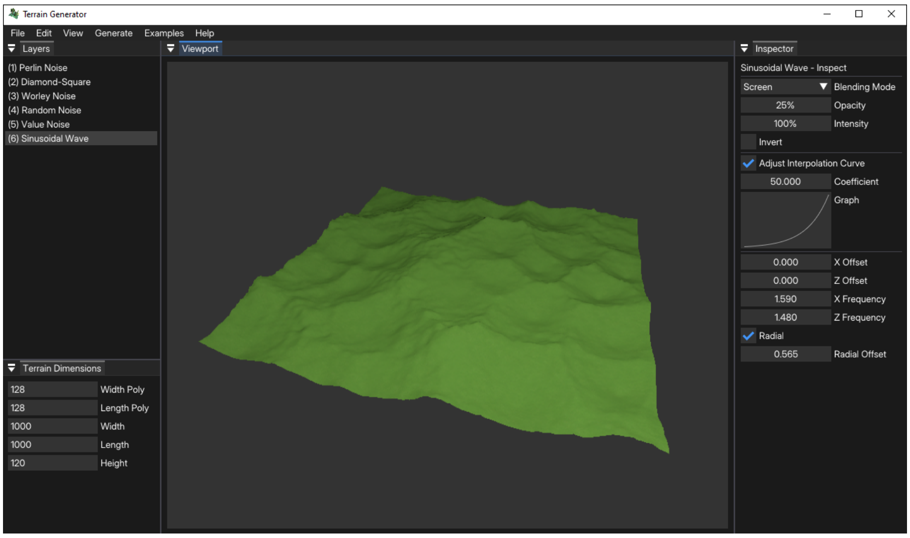

Figure 4: Developed software application.

3. Conclusions

The developed application (Figure 4) marks a significant advancement in the field of digital landscape generation by integrating advanced mathematical models with computational techniques to produce realistic and efficient informational models of terrain. By leveraging simplified forms of complex equations, such as the Navier-Stokes, shallow water, and Korteweg-de Vries equations, the system effectively simulates natural processes like tectonic activity, erosion, and water flow, while maintaining computational efficiency. The use of solitons, peakons, and shock wave models adds further realism to the generated landscapes, replicating dynamic features such as sharp mountain peaks, canyons, and fluid landforms.

In terms of its practical application, this system offers a highly efficient tool for creating detailed digital landscapes in immersive environments such as virtual reality systems, video games, and geospatial simulations. Its ability to generate naturalistic and scalable terrains makes it indispensable for developers who need to construct large, interactive environments while conserving computational resources. The use of optimized numerical methods, including adaptive mesh refinement and discontinuous Galerkin techniques, allows for a reduction in computational costs without sacrificing detail or accuracy, which is especially critical in resource-limited scenarios. Additionally, the generated informational models are highly customizable, offering users the flexibility to adjust the complexity and resolution of landscapes based on their specific project needs, making the application valuable for both entertainment and professional simulation industries.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] A. Chen, J. Darbon, J.-M. Morel, Landscape evolution models: A review of their fundamental equations, Geomorphology 219 (2014) 68–86. doi:10.1016/j.geomorph.2014.04.037.

[2] A. I. Ruban, J. S. B. Gajjar, Fluid Dynamics. Part 1: Classical Fluid Dynamics, Oxford University Press, Oxford, UK, 2014. doi:10.1093/acprof:oso/9780199681730.002.0003.

[3] D. A. Anderson, J. C. Tannehill, R. H. Pletcher, R. Munipalli, V. Shankar, Computational Fluid Mechanics and Heat Transfer, 4 ed., CRC Press, Taylor & Francis Group, Abingdon, UK, 2021. doi:10.1201/9781351124027.

[4] O. Stanislaviv, O. Zholtovskyi, O. Smalko, Computer modeling of some nature processes for landscape generation, Math. Comput. Model., Ser. Tech. Sci. (2024) 106–113. doi:10.32626/2308-5916.2024-25.106-113.

[5] S. McKee, M. F. Tomé, V. G. Ferreira, J. A. Cuminato, A. Castelo, F. S. Sousa, N. Mangiavacchi, The mac method, Computers & Fluids 37 (2008) 907–930. doi:10.1016/j.compfluid.2007.10.006.

[6] Y. Kodama, Solitons in Two-Dimensional Shallow Water, SIAM-Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2018. doi:10.1137/1.9781611975529.

[7] R. Camassa, D. D. Holm, An integrable shallow water equation with peaked solitons, Physical Review Letters 71 (1993) 1661–1664. doi:10.1103/PhysRevLett.71.1661.