

# ANUGA SOFTWARE FOR NUMERICAL SIMULATIONS OF SHALLOW WATER FLOWS

Sudi Mungkasi<sup>1,2</sup> and Stephen Gwyn Roberts<sup>1</sup>

<sup>1</sup>Mathematical Sciences Institute, Australian National University, Canberra, ACT 0200, Australia

<sup>2</sup>Department of Mathematics, Sanata Dharma University, Mrican Tromol Pos 29, Yogyakarta 55002, Indonesia

E-mail: [sudi.mungkasi@anu.edu.au](mailto:sudi.mungkasi@anu.edu.au)

## Abstract

Shallow water flows are governed by the shallow water wave equations, also known as the Saint-Venant system. This paper presents a finite volume method used to solve the two-dimensional shallow water wave equations and how the finite volume method is implemented in ANUGA software. This finite volume method is the numerical method underlying the software. ANUGA is open source software developed by Australian National University (ANU) and Geoscience Australia (GA). This software uses the finite volume method with triangular domain discretisation for the computation. Four test cases are considered in order to evaluate the performance of the software. Overall, ANUGA is a robust software to simulate two-dimensional shallow water flows.

**Keywords:** *finite volume methods, shallow water flows, ANUGA software, triangular grids, shallow water wave equations*

## Abstrak

Arus air dangkal diatur dalam persamaan gelombang air dangkal, dikenal sebagai sistem Saint-Venant. Penelitian ini menyajikan metode *finite volume* yang digunakan untuk menyelesaikan persamaan gelombang air dangkal dua dimensi dan bagaimana metode *finite volume* diimplementasikan dalam perangkat lunak ANUGA. Metode *finite volume* adalah metode numerik yang mendasari perangkat lunak ANUGA. ANUGA sendiri adalah perangkat lunak *open source* yang dikembangkan oleh Australian National University (ANU) dan Geoscience Australia (GA). Perangkat lunak ini menggunakan metode *finite volume* dengan diskritisasi domain segitiga dalam proses komputasi. Empat uji kasus digunakan untuk mengevaluasi kinerja perangkat lunak. Secara keseluruhan, ANUGA adalah perangkat lunak yang *robust* untuk mensimulasikan dua dimensi aliran arus air dangkal.

**Kata Kunci:** *finite volume methods, shallow water flows, ANUGA software, triangular grids, shallow water wave equations*

## 1. Introduction

Since the nineteenth century the shallow water wave equations have emerged as important mathematical models to describe water flows. Some of its developments and applications are given by Roberts et al. [1-5], and Stoker [6].

In 1892, Ritter [7] studied dam break modelling analytically. This old work has had a big impact as the dam break problem is now becoming a standard benchmark to test the performance of numerical methods. Even though this problem is originally modelled for one-dimensional shallow water wave equations, it can

also be treated as a two-dimensional problem, which we present in this paper as a planar dam break problem. In addition, it can also be extended to a circular two-dimensional problem.

Oscillation of water on a paraboloid canal developed by Thacker [8] is another benchmark problem, which is interesting, and usually considered by shallow water researchers to test their numerical methods. This problem involves a wetting and drying process, a process that is well-known to be very difficult to resolve [9-13]. Yoon and Cho [14] used this type of problem to test their numerical method.

To complete testing a numerical method, a steady flow problem must also be performed in addition to unsteady flow test problems. In this

---

This paper is the extended version from paper titled "A finite volume method for shallow water flows on triangular computational grids" that has been published in Proceeding of ICACSI 2012.

paper, we do perform a steady flow involving a shock over a parabolic obstruction. This test case is usually used in testing a one-dimensional shallow water solver.

One open-source free software developed to simulate two-dimensional shallow water flows is ANUGA, named after Australian National University (ANU) and Geoscience Australia (GA). This software uses a finite volume method as the underlying mathematical background. We will present this finite volume method and test the performance of ANUGA using the planar dam break problem, circular dam break problem, oscillation on a paraboloid channel, and a steady flow over an obstruction. ANUGA uses triangular domain discretisation for the computation.

The remainder of this paper is organised as follows. In Section 2, we present the shallow water wave equations governing water flows, the finite volume method and the ANUGA software. Section 3 contains four numerical simulations involving unsteady and steady flow problems. Finally, Section 4 concludes the paper with some remarks.

## 2. Methodology

Shallow water wave equations, the one-dimensional shallow water wave equations can be written as

$$q_t + f(q)_x = s. \quad (1)$$

The notations used here are described as follows:  $q = [huh]^T$  is the vector of conserved quantities consisting of water depth  $h$  and momentum  $uh$ ,  $f(q) = [uhu^2h + g_r h^2/2]^T$  is the flux function,  $s = [0 \quad -g_r h(z_x + \psi)]^T$  is the source vector with  $\psi$  is some defined friction terms,  $x$  represents the distance variable along the flow,  $t$  represents the time variable,  $z(x)$  is the fixed water bed,  $h(x, t)$  is the height of the water at point  $x$  and at time  $t$ ,  $u(x, t)$  denotes the velocity of the water flow at point  $x$  and at time  $t$ , and  $g_r$  is a constant denoting the acceleration due to gravity. In addition, a new variable called the stage can be defined as  $w(x, t) = z(x) + h(x, t)$  which is the absolute water level.

More general equations governing shallow water flows are the two-dimensional shallow water wave equations [15–20]

$$q_t + f(q)_x + g(q)_y = s \quad (2)$$

where  $q = [h \quad uh \quad vh]^T$  is the vector of conserved quantities consisting of water depth  $h$ ,  $x$ -momentum  $uh$ , and  $y$ -momentum  $vh$ . Here,  $u$  and

$v$  are velocities in the  $x$ - and  $y$ -direction;  $f(q)$  and  $g(q)$  are flux functions in the  $x$ - and  $y$ -direction given by

$$f(q) = \begin{bmatrix} uh \\ u^2h + \frac{1}{2}g_r h^2 \\ uvh \end{bmatrix} \quad (3)$$

and

$$g(q) = \begin{bmatrix} vh \\ vuh \\ v^2h + \frac{1}{2}g_r h^2 \end{bmatrix} \quad (4)$$

The source term including gravity and friction is

$$s = \begin{bmatrix} 0 \\ -g_r h(z_x + S_{fx}) \\ -g_r h(z_y + S_{fy}) \end{bmatrix} \quad (5)$$

where  $z(x, y)$  is the bed topography, and  $S_f = \sqrt{S_{fx}^2 + S_{fy}^2}$  is the bed friction modelled using Manning's resistance law

$$S_{fx} = \frac{u\eta^2\sqrt{u^2 + v^2}}{h^{4/3}} \quad (6)$$

and

$$S_{fy} = \frac{v\eta^2\sqrt{u^2 + v^2}}{h^{4/3}} \quad (7)$$

in which  $\eta$  is the Manning resistance coefficient. It should be noted that the stage (absolute water level)  $w$  is given by  $w := z + h$ . In this paper, we focus on the two-dimensional shallow water wave equations.

Integrating equation 2 over an arbitrary closed and connected spatial domain  $\Omega$  having boundary  $\Gamma$  and applying the Gauss divergence theorem to the flux terms, we get the integral form

$$\frac{\partial}{\partial t} \int_{\Omega} q \, d\Omega + \oint_{\Gamma} F \cdot n \, d\Gamma = \int_{\Gamma} s \, d\Omega \quad (8)$$

where  $F = [f(q) \quad g(q)]^T$  is the flux function,  $n = [\cos(\theta) \quad \sin(\theta)]^T$  is the outward normal vector of the boundary, and  $\theta$  is the angle between  $n$  and the  $x$ -direction. Equation 8 is called the integral form of the two-dimensional shallow water wave equations.

The rotational invariance property of the shallow water wave equations implies that

$$F \cdot n = T^{-1} f(Tq) \quad (9)$$

where  $T$  is the transformation matrix

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix}. \quad (10)$$

Therefore, equation 8 can be rewritten as

$$\frac{\partial}{\partial t} \int_{\Omega} q \, d\Omega + \oint_{\Gamma} T^{-1} f(Tq) \, d\Gamma = \int_{\Gamma} s \, d\Omega \quad (11)$$

Finite volume method, a finite volume method is based on subdividing the spatial domain into cells and keeping track of an approximation to the integral of the quantity over each of these cells. In this paper, we limit our presentation for triangular computational grids (cells).

After the spatial domain is discretized, we have the equation constituting the finite volume method over each triangular cell of the grids [16]

$$\frac{dq_i}{dt} + \frac{1}{A_i} \sum_{j \in \mathcal{N}(i)} H_{ij} l_{ij} = s_i \quad (12)$$

where  $q_i$  is the vector of conserved quantities averaged over the  $i$ th cell,  $s_i$  is the source term associated with the  $i$ th cell,  $H_{ij}$  is the outward normal flux of material across the  $ij$ th edge, and  $l_{ij}$  is the length of the  $ij$ th edge. Here, the  $ij$ th edge is the interface between the  $i$ th and  $j$ th cells. The flux  $H_{ij}$  is evaluated using a numerical flux function  $H(\cdot, \cdot; \cdot)$  such that for all conservation vectors  $q$  and normal vectors  $n$

$$H(q, q; n) = F \cdot n. \quad (13)$$

Furthermore,

$$H_{ij} = H(q_i(m_{ij}), q_j(m_{ij}); n_{ij}) \quad (14)$$

where  $m_{ij}$  is the midpoint of the  $ij$ th edge and  $n_{ij}$  is the outward normal vector, with respect to the  $i$ th cell, on the  $ij$ th edge. The function  $q_i$  is obtained from the averaged values of quantities in the  $i$ th and neighbouring cells.

Now let  $n_{ij} = [n_{ij}^{(x)} \ n_{ij}^{(y)}]^T$ . From equation 13 and equation 14, we have

$$H_{ij} = f[q_i(m_{ij})]n_1 + g[q_j(m_{ij})]n_2. \quad (15)$$

Here, we recall the algorithm to solve the two-dimensional shallow water wave equations given by Guinot [15]. In the semi-discrete framework, four steps are considered as follows:

For each interface  $(i, j)$ , transform the quantity  $q_i$  and  $q_j$  in the global coordinate system  $(x, y)$  into the quantity  $\hat{q}_i$  and  $\hat{q}_j$  in the local coordinate system  $(\hat{x}, \hat{y})$ . The water depth  $h$  is unchanged as it is a scalar variable, while the velocities  $u$  and  $v$  are transformed into  $\hat{u}$  and  $\hat{v}$ . Therefore, the new quantities in the local coordinate system are

$$\hat{q}_i = Tq_i \quad \text{and} \quad \hat{q}_j = Tq_j \quad (16)$$

where  $q_i = [h_i \ (hu)_i \ (hv)_i]$ ,  $q_j = [h_j \ (hu)_j \ (hv)_j]$  are, respectively, the quantities in the global coordinate system. The matrices  $T$  and  $T^{-1}$  are

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & n_{ij}^{(x)} & n_{ij}^{(y)} \\ 0 & -n_{ij}^{(y)} & n_{ij}^{(x)} \end{bmatrix} \quad (17)$$

and

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & n_{ij}^{(x)} & -n_{ij}^{(y)} \\ 0 & n_{ij}^{(y)} & n_{ij}^{(x)} \end{bmatrix}. \quad (18)$$

Compute the flux  $\hat{f}$  at the interface  $(i, j)$ . In the local coordinate system, the problem is merely a one-dimensional Riemann problem, as the flux vector is parallel with the normal vector of the interface. Therefore, the equations to be solved are

$$\hat{q}_t + \hat{f}(\hat{q})_{\hat{x}} = \hat{s} \quad (19)$$

with initial condition given by  $\hat{q}_i$  on one side of the interface  $(i, j)$  and  $\hat{q}_j$  on the other side of the interface  $(i, j)$ . It should be stressed that at this step, we do not need to integrate equation 19 for the quantity  $\hat{q}$ , but all we need is the value of the flux  $\hat{f}$ . Hence, applying either the exact or approximate Riemann solvers to compute the flux  $\hat{f}$  at the interface  $(i, j)$  is enough. Note that here the source term does not need to be transformed in the local coordinate system, since it involves scalar variables only.

Transform the flux  $\hat{f}$  back to the global coordinate system  $(x, y)$ , so the flux at the

midpoint of the interface  $(i, j)$  in the global coordinate system is

$$\begin{aligned} H_{ij} &= T_{ij}^{-1} \hat{f} \\ &= T_{ij}^{-1} \tilde{f}(\hat{q}_i; \hat{q}_j; \hat{s}_i; \hat{s}_j) \\ &= T_{ij}^{-1} \tilde{f}(T_{ij} q_i; T_{ij} q_j; s_i; s_j). \end{aligned} \quad (20)$$

Here,  $\tilde{f}$  is the flux computed with a Riemann solver for the one-dimensional problem stated in Step 2.

Finally, solve equation 12 where  $\mathcal{N}(i) = \{0, 1, 2\}$ , as triangular grids are considered, for  $q_i$ . We multiply  $H_{ij}$  with  $l_{ij}$  in order to get the flux over the interface  $(i, j)$ . This is because the flux at the midpoint of the interface  $(i, j)$  is  $H_{ij}$ , and the length of the interface  $l_{ij}$ .

In this part, we briefly describe an open-source free software developed using mathematical models described in the previous sections called ANUGA. Uppercased word 'ANUGA' refers to the name of the software, whereas lowercased word 'anuga' refers to the name of ANUGA module in programming. See ANUGA User Manual for the details of anuga [14].

ANUGA is an open source (free software) developed by Australian National University (ANU) and Geoscience Australia (GA) to be used for simulating shallow water flows, such as floods and tsunamis. The mathematical background underlying the software is the two-dimensional shallow water wave equations and finite volume method presented in Sections 2 and 3 above. The interface of this software is in Python language, but the mostly expensive computation parts, such as flux computation, are written in C language. A combination of these two languages provides two different advantages: Python has the flexibility in terms of software engineering, while C gives a very fast computation.

As has been described in the ANUGA User Manual [16], a simple simulation using ANUGA generally has five steps in the interface code. These five steps are importing necessary modules, setting-up the computational domain, setting-up initial conditions, setting-up boundary conditions, and evolving the system through time. The necessary modules to be imported are anuga<sup>ii</sup> and some standard libraries such as numpy, scipy, pylab, and time. The simplest creation of the triangular computational domain is in the rectangular-cross framework, which returns points, vertices, and boundary for the computation. The initial condition includes the definition of the topography, friction, and stage. The boundary conditions can be chosen in several ways depending on the needs: reflective boundary

is for solid reflective wall; transmissive boundary is for continuing all values on the boundary; Dirichlet boundary is for constant boundary values; and time boundary is for time dependent boundary. A thorough description of this software is available at <http://anuga.anu.edu.au>.

### 3. Results and Analysis

This section presents some computational experiments using the finite volume method with ANUGA as the software used to perform the simulation. For simplicity of the experiments, frictionless topography is assumed. Four different test cases are considered: a planar dam break problem, circular dam break problem, water oscillation on a paraboloid channel, and steady flow over a parabolic obstruction.

The numerical settings are as follows. Even though ANUGA is capable to do the computations with unstructured triangular grids, we limit our presentation only to the structured triangular grids for brevity with rectangular-cross as the basis. The spatial reconstruction and temporal discretisation are both second order with edge limiter. A reflective boundary is imposed. The numerical flux is due to Kurganov et al. [21]. SI units are used in the measurements of the quantities.



Figure 1. Initial condition for the planar dam break problem.

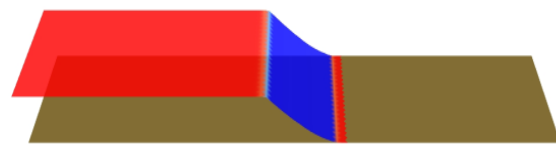


Figure 2. Water flows 0.5 second after the planar dam is broken.



Figure 3. Water flows 1.0 second after the planar dam is broken.

Consider a flat topography without friction on a spatial domain given by  $\{(x, y) | (x, y) \in [-50, 50] \times [-10, 10]\}$ . Suppose that we are given a

planar dam break problem with wet/dry interface between the dam, where we have 10 meters water depth on the left of the dam, but dry area on the right. The dam is at  $\{(0,y)|y \in [-10,10]\}$ . The simulation is then done to predict the flow of water after the dam is removed. Note that this dam break problem is actually a one-dimensional dam-break problem, but here we treat this as a two-dimensional problem.



Figure 4. Water flows 1.5 seconds after the planar dam is broken.

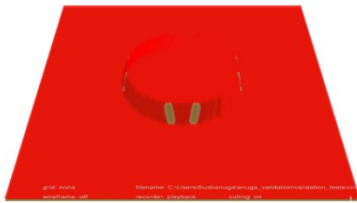


Figure 5. Initial condition for the circular dam break problem.

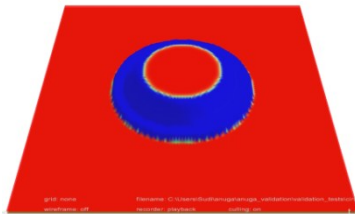


Figure 6. Water flows 0.5 second after the circular dam is broken.

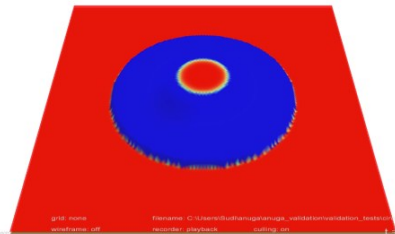


Figure 7. Water flows 1.0 second after the circular dam is broken.

The domain setting is as follows. The spatial domain is discretised into 100 by 20 rectangular-crosses, where each rectangular cross has 4 uniform triangles. This means that we have  $8 \cdot$

$10^3$  structured triangles as the discretisation of the spatial domain.

The water surface and topography are shown in figure 1, 2, 3, and 4. In particular, figure 1 shows the initial condition, that is, the water profile at time  $t = 0.0$  s. After the dam is broken, the profiles at time  $t = 0.5, 1.0, 1.5$  s are illustrated in figure 2, figure 3, and figure 4 respectively. At 1.5 seconds after dam break, the water flows to the right for a distance almost 30 m. This agrees with the analytical solution (30 m) for the one-dimensional dam break problem.

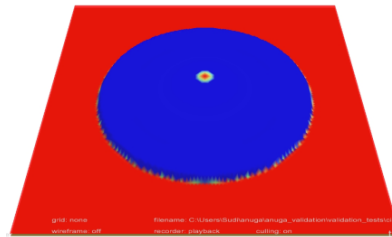


Figure 8. Water flows 1.5 seconds after the circular dam is broken.

Now suppose that we have a circular dam break problem with wet/wet interface between the dam. Consider a circular dam, where the point of origin is the centre, having 20 m as its radius. The water depth inside the circular dam is 10 m, while outside is 1 m. The spatial domain of interest is  $\{(x,y)|(x,y) \in [-50,50] \times [-50,50]\}$ . Similar to the planar dam break problem, the simulation of this test case is then done to predict the flow of water, after the dam is removed.

The domain setting is as follows. The spatial domain is discretised into 100 by 100 rectangular-crosses, where each rectangular cross has 4 uniform triangles. This means that we have  $4 \cdot 10^4$  structured triangles as the discretisation of the spatial domain.

The water surface and topography are shown in figure 5, 6, 7, and 8. In particular, figure 5 shows the initial condition, that is, the water profile at time  $t = 0.0$  s. After the dam is broken, the profiles at time  $t = 0.5, 1.0, 1.5$  s are illustrated in figure 6, figure 7, and figure 8 respectively. At 1.5 seconds after dam break, the water floods to the surrounding area approximately 25 m outwards.

This test case is adapted from the work of Thacker [8] and Yoon and Chou [14]. Suppose that we have a spatial domain given by  $\{(x,y)|(x,y) \in [-4000,4000] \times [-4000,4000]\}$ . We define some parameters  $D_0 = 1000, L = 2500, R_0 = 2000$  such that

$$A = \frac{L^4 - R_0^4}{L^4 + R_0^4} \quad (21)$$

$$\omega = \frac{2}{L\sqrt{2g_r D_0}} \quad (22)$$

$$T = \frac{2\pi}{\omega} \quad (23)$$

are the amplitude of oscillation at the origin, the angular frequency, and the period of oscillation. Furthermore, the topography is given by

$$z(x, y) = -D_0 \left( 1 - \frac{r^2}{L^2} \right) \quad (24)$$

and the initial water level (the initial wet region) is

$$h(x, y) = D_0 \left[ \frac{\sqrt{1 - A^2}}{1 - A \cos(\omega t)} - 1 - \frac{r^2}{L^2} \left( \frac{1 - A^2}{(1 - A \cos(\omega t))^2} - 1 \right) \right] \quad (25)$$

where

$$r = \sqrt{x^2 + y^2} . \quad (26)$$

The domain setting is as follows. The spatial domain is discretised into 50 by 50 rectangular-crosses, where each rectangular cross has 4 uniform triangles. This means that we have  $10^4$  structured triangles as the discretisation of the spatial domain.

The simulation is then done to predict the flow of the water. According to Thacker [8], the flow will be a periodic oscillation. The simulation, using the numerical method and software discussed in this paper, indeed shows this periodic motion. The water profiles at time  $t = 0, T/2, T$  are shown in figure 9, figure 10, and figure 11 respectively. In addition, the periodic motion of the water surface corresponding to the origin is depicted in figure 12. From the numerical results, we see a minor damping of the oscillation. However, a perfect method or software should not produce this damping. Although it may be impossible to have a perfect method, we take this for future research for the ANUGA software to minimise or eliminate this kind of damping. We also see from the numerical results that a negligible amount of water is left over in the supposed-dry area.

Steady flow over a parabolic obstruction, this test case is for checking if ANUGA can produce a

steady shock. Suppose that we have a spatial domain given by  $\{(x, y) | (x, y) \in [0, 15] \times [0, 15]\}$ . We consider the topography as horizontal bed except for a defined parabolic obstruction

$$z(x, y) = 0.2 - 0.05(x - 10)^2 \quad (27)$$

for  $8 < x < 12$ . We use Dirichlet boundaries for the left and right boundaries. The left boundary is set up as water height 0.42,  $x$ -momentum 0.18, and  $y$ -momentum 0.0. The right boundary is set up as water height 0.33,  $x$ -momentum 0.18, and  $y$ -momentum 0.0. The initial condition is just a quiet lake at rest with the stage level as 0.33.

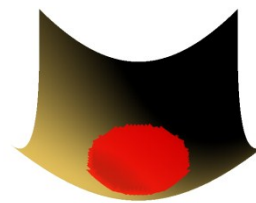


Figure 9. Initial condition for the oscillation problem.

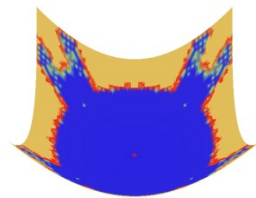


Figure 10. Water flows at time  $t = T/2$  for the oscillation problem.

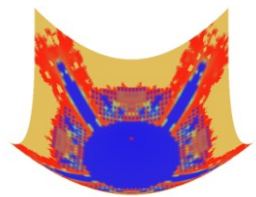


Figure 11. Water flows at time  $t = T$  for the oscillation problem.

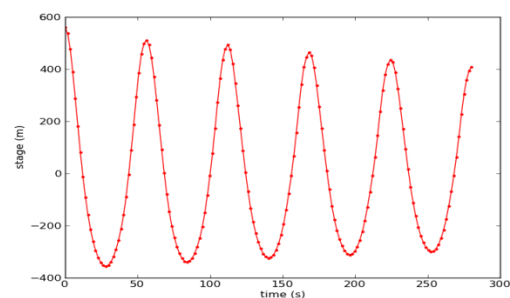


Figure 12. The water surface corresponding to the origin until  $t = 5T$  for the oscillation problem.

The domain setting is as follows. The spatial domain is discretised into 120 by 120 rectangular-crosses, where each rectangular cross has 4 uniform triangles. This means that we have 57,600 structured triangles as the discretisation of the spatial domain.

The summary of the simulation results is as follows. Figures 13 and 14 show the water surface at initial condition and at time  $t = 25$ . After water is flowing for 25 seconds from left boundary to the right, a steady shock occurs on the parabolic obstruction. This agrees with the simulation using purely one-dimensional model using equation 1 in our previous work [22]. (It is unfortunate that due to the small water height, the shock and the bottom topography are not clearly shown in figure 14. However, these can be clearly seen using ANUGA Viewer as parts of the picture can be magnified.)



Figure 13. Steady flow over a parabolic obstruction at  $t = 0$ .



Figure 14. Steady flow over a parabolic obstruction at  $t = 25$ .

#### 4. Conclusion

We have presented some mathematical (both analytical and numerical) background underlying the ANUGA software. ANUGA was tested using four numerical simulations. We infer from the simulation results that ANUGA can solve nicely the wetting problem. For the drying problem, some very small amount of water, which is negligible, may be left over on the supposed-dry area. Regardless of what we have investigated, we need to note that different parameters or numerical settings generally lead to different results. We also note that ANUGA can simulate both steady and unsteady state problems.

#### Acknowledgment

The work of Sudi Mungkasi was supported by ANU PhD and ANU Tuition Scholarships.

#### References

- [1] S. Mungkasi & S.G. Roberts, "A finite volume method for shallow water flows on triangular computational grids" *In Proceedings of International Conference on Advanced Computer Science and Information System (ICACSIS) 2011*, pp. 79-84, 2011.
- [2] S.G. Roberts & P. Wilson, "A well balanced scheme for the shallow water wave equations in open channels with (discontinuous) varying width and bed," *On ANZIAM Journal*, vol. 52(CTAC2010), pp. C967-C987, 2011.
- [3] J.D. Jakeman, N. Bartzis, O.M. Nielsen, & S. Roberts, "Inundation modelling of the december 2004 indian ocean tsunami" *In Proceedings of MODSIM07*, pp. 1667-1673, 2007.
- [4] S. Mungkasi & S.G. Roberts, "On the best quantity reconstructions for a well balanced finite volume method used to solve the shallow water wave equations with a wet/dry interface" *On ANZIAM Journal*, vol. 51(EMAC2009), pp. C48-C65, 2010.
- [5] S.G. Roberts, O.M. Nielsen, & J. Jakeman, "Simulation of tsunami and flash floods," *Modeling Simulation and Optimization of Complex Processes" In Proceedings of the International Conference on High Performance Scientific Computing*, pp. 489-498, 2008.
- [6] J.J. Stoker, *Water Waves: The Mathematical Theory with Application*, Interscience Publishers, New York, 1957.
- [7] A. Ritter, "Die fortpflanzung der wasserwellen," *Zeitschrift des Vereines Deutscher Ingenieure*, vol. 36(33), pp. 947-954, 1892.
- [8] W.C. Thacker, "Some exact solutions to the nonlinear shallow water equations," *Journal of Fluid Mechanics*, vol. 107, pp. 499-508, 1981.
- [9] K. Ganeshamoorthy, D.N. Ransinghe, K.P.M.K. Silva, & R. Wait, "Performance of shallow water equations model on the computational grid with overlay memory architectures" *In Proceedings of the Second International Conference on Industrial and*

- Information Systems*, pp. 415–420, 2007.
- [10] S. Guangcai, W. Wenli, & Y.L. Liu, “Numerical scheme for simulation of 2d flood waves” *In Proceedings of the International Conference on Computational and Information Sciences*, pp. 846–849, 2010.
- [11] J.D. Jakeman, O.M. Nielsen, K.V. Putten, R. Mleczko, D. Burbidge, & N. Horspool, “Towards spatially distributed quantitative assessment of tsunami inundation models,” *International Workshop on Modeling the Ocean (IWMO): Dynamics, Syntheses and Prediction*, vol. 60(5), pp. 1115–1138, 2010.
- [12] C. Lu, Y. Chen, & G. Li, “Weighted essential non-oscillatory schemes for simulations of shallow water equations with transport of pollutant on unstructured meshes” *In Proceedings of the Fourth International Conference on Information and Computing*, pp. 174–177, 2011.
- [13] S. Mungkasi & S.G. Roberts, “Approximations of the carrier–greenspan periodic solution to the shallow water wave equations for flows on a sloping beach,” *International Journal for Numerical Methods in Fluids*, 2011.
- [14] S.B. Yoon & J.H. Cho, “Numerical simulation of coastal inundation over discontinuous topography,” *Water Engineering Research*, vol. 2(2), pp. 75–87, 2001.
- [15] V. Guinot, *Wave propagation in fluids: models and numerical techniques*, ISTE Ltd (and John Wiley & Sons, Inc), London (and Hoboken), 2008.
- [16] S. Roberts, O. Nielsen, D. Gray, & J. Sexton, *ANUGA User Manual*, Geoscience Australia, 2010.
- [17] W. Tan, *Shallow water hydrodynamics*, Elsevier Science Publishers, Amsterdam, 1992.
- [18] C. Zoppou & S. Roberts, Shallow water wave equations, Australian National University, book in preparation.
- [19] C. Zoppou & S. Roberts, “Catastrophic collapse of water supply reservoirs in urban areas,” *Journal of Hydraulic Engineering*, American Society of Civil Engineers, vol. 125(7), pp. 686–695, 1999.
- [20] C. Zoppou & S. Roberts, “Explicit schemes for dam-break simulations,” *Journal of Hydraulic Engineering*, vol. 129(1), pp. 11–34, 2003.
- [21] A. Kurganov, S. Noelle, & G. Petrova, “Semidiscrete central-upwind schemes for hyperbolic conservation laws and hamilton-jacobi equations,” *SIAM Journal of Scientific Computing*, vol. 23(3), pp. 707–740, 2001.
- [22] S. Mungkasi & S.G. Roberts, “Numerical entropy production for shallow water flows,” *ANZIAM Journal*, vol. 52(CTAC2010), pp. C1–C17, 2011.