A Preliminary Meshfree Numerical Approach to Inducing Runoff along Saturated Soil Slope

L.M. Dakssa and I.S.H. Harahap

Universiti Teknologi PETRONAS

Perak, Malaysia

Abstract- **Infiltration and surface runoff are two main phenomena taking place as rainwater reaches soil slopes during rainfall events. Runoff occurs whenever the rainfall intensity becomes greater than the infiltration rate of the soil slope. In rainfall triggered landslides, a detailed account of infiltration and surface runoff is of paramount importance as it is the build-up of pore-water pressure, due to rainwater infiltration, that usually causes soil slope instability. While negative pore-water pressure adds to the stability of soil slopes, occurrence of positive pore-water pressures disrupts the existing stability – the condition that mostly leads to failure of a standing soil slope during or immediately after heavy or prolonged rainfall events, especially, in tropical and subtropical regions. In this paper, the application of smoothed particle hydrodynamics (SPH) – a meshfree numerical method - to triggering surface runoff along a saturated soil slope was investigated in view of predicting fast moving landslides, such as debris flows and avalanche. The governing equation used in the current research was that of the fundamental Navier-Stokes (NS) equation. SPH codes in FORTRAN language were developed to run the simulation. Snapshots of the simulation are presented. The snapshots demonstrate that the SPH scheme is able to capture important aspects of the surface runoff phenomenon.**

Keywords: debris flow, pore-water pressure, rainfall, runoff, saturated soil, soil slope, smoothed particle hydrodynamics.

I. INTRODUCTION

During rainfall events, part of the rainwater reaching the earth's surface enters the soil as infiltration while the other part flows down the slope as surface runoff (i.e. neglecting interception, evaporation, and other losses). The amount of rainwater that goes as infiltration and runoff depends on the infiltration rate of the soil. Water entering the soil disturbs the existing moisture equilibrium in the soil. As such, on the basis of water content, soil can be said to be *saturated* when all the soil void spaces are filled with water, while a soil with void spaces partially filled with water is referred to as *unsaturated/partially saturated* soil. Analysis of in-soil water movement plays a significant role in geotechnical hazards investigation. Fast moving landslides (for instance, debris flows, avalanche, etc.) are one of the geotechnical hazards known to cause tremendous life and wealth losses all over the world. Though, in most cases, principles of saturated soil mechanics are used in soil slope stability analysis, in practice, there are myriad evidences that most landslides occur while the soil is still in unsaturated condition. The main difference between saturated and unsaturated soils lies in the manner in which we treat the hydraulic conductivity of the soil. In dealing with saturated soils, soil's hydraulic conductivity is usually assumed to be constant. In modeling water flow through unsaturated soils, however, the hydraulic conductivity is assumed to vary in accordance with the soil water content or matric suction.

Owing to the grave consequences of landslides, both in terms of loss of life and wealth, conducting landslide numerical modeling and simulation is of paramount importance. Landslide predictions are usually made in terms of time and depth of failure so that residents residing nearby could get time to escape.

Conventionally, mesh-based numerical techniques, such as finite difference (FD) and finite element (FE), have been used in most geotechnical investigations of civil engineering works. However, these numerical methods (i.e. FD and FE) have inherent difficulties for using, especially, in areas where large deformations are expected, because of mesh-distortion in case of FE method and because of inefficient use of regular grids for irregular geometries in the case of FD method. In the current research, the goal is, therefore, to explore the applicability of the smoothed particle hydrodynamics (SPH) – one of the meshfree (also referred to as meshless) methods – to triggering surface runoff down a saturated soil slope. The numerical technique smoothed particle hydrodynamics (SPH) was, originally invented for simulating astrophysical phenomena, and, later, its applications to several fields of science and engineering have been reported in the literature. In terms of paper organization, brief background information on SPH formulation is presented first followed by numerical examples and concluding remarks.

II. METHODOLOGY: SMOOTHED PARTICLE HYDRODYNAMICS

Smoothed particle hydrodynamics (SPH) is a macroscopic numerical approach initially invented for simulating astrophysical phenomena in 1977 by Lucy, and by Gingold and Monaghan (as reported in [1]) and, later, it was applied to different areas of research, including free surface flows, flow thorough porous media, etc. SPH meshless numerical method formulation is based on interpolation theory, and two essential concepts dictate its formulation: (i) kernel approximation (ii) particle approximation.

A. Kernel Approximation

Kernel approximation consists in approximating the values of both the field function and the derivative of the field function. Kernel approximation of field functions is, in essence, the representation of the field function(s) in integral form. This is achieved by multiplying an arbitrary field functions with a smoothing kernel function. Therefore, a function *A* of a three-dimensional position vector *x* (or an estimate of the function *A(x)* at location *x'*) can be expressed in integral form

$$A(x) = \int_{\Omega} A(x')\delta(x - x')dx' \quad (1)$$

where, $\delta(x - x')$ is the Dirac delta function, given by

$$\delta(x - x') = \begin{cases} 1 & for\, x = x' \\ 0 & elsewhere \end{cases} \quad (2)$$

where, Ω is the volume of the integral containing $\boldsymbol{x}$; x' is a new independent variable.

In the above expression, the function *A(**x**)* is exact or rigorous, so long as the Dirac delta function is used and *A(**x**)* is continuous in Ω. In SPH, however, the Dirac delta function needs to be replaced by the smoothing (weighting) function $W(x - x', h)$ in which case it will become an approximate representation of *A(**x**).* The SPH form of a function approximation (or kernel approximation) is, therefore

$$A(x) = \int_{\Omega} A(x')W(x - x', h)dx' \quad (3)$$

where, *W* is called the kernel or smoothing function; *h* is the smoothing length, and demarcates the influence area of the smoothing function. It needs to be noted, however, that (3) gives an approximate representation of the integral of a field function as long as *W* is not a Dirac delta function; and, hence, the name kernel approximation.

Kernel approximation of derivative of a function, on the other hand, refers essentially to approximating the values of the gradient and the divergence of the field function. As equations of computational fluid dynamics problems are mostly PDEs of second degree [1], an appropriate approximation of the function derivatives is of profound importance. Accordingly, in SPH method, such approximations are usually performed by simply replacing the function A(x) in (3) by $\nabla A(x)$, for gradient and, $\nabla.\mathrm{A(x)}$ for the divergence. More specifically, the kernel approximation of the divergence of the field function A(x) (for vector quantity) is

$$\nabla \cdot A(x) = \int_{\Omega} \nabla \cdot A(x')W(x - x', h)dx' \quad (4)$$

After applying the divergence theorem, it is always the case that the divergence operation on the primed coordinate in (4) is transferred to the gradient of the smoothing function in SPH numerical approach, which entails re-writing (4) as

$$\nabla \cdot A(x) = -\int_{\Omega} A(x') \cdot \nabla W(x - x', h)dx' \quad (5)$$

Note that a *dot product* is used in (5). Similarly, the *gradient* of the function (for scalar quantity) is expressed as

$$\nabla A(x) = -\int_{\Omega} A(x')\nabla W(x - x', h)dx' \quad (6)$$

It can be said, therefore, that the spatial derivative of a field function can be evaluated from the values of the field function and the spatial derivative of the smoothing function. It should also be noted that the negative sign outside the integral sign in (5) and (6) can be removed if the spatial derivative of the kernel function is taken with respect to **x** instead of the primed **x** (i.e., x').

B. Particle Approximation

Similarly, particle approximation consists in approximating the field function and its spatial derivatives (gradient and divergence). Particle approximation is another key operation in SPH numerical formulation; and is the means of transforming the continuous kernel approximation (in integral form) into the summation over all particles at the discrete points in the support domain. Particles carry mass, m, velocity, $\boldsymbol{v}$, and other quantities specific to the given problem; and, are regarded as interpolation points, analogous to the grid nodes in mesh-based numerical methods. Therefore, equations that govern the evolution of fluid quantities are expressed as summation interpolants with the help of smoothing function. Equation (3) can, then, be approximated in a summation form as

$$A(x) \approx \sum_{j=1}^{N} A(x_j)W(x - x_j, h)\frac{m_j}{\rho_j} \quad (7)$$

where N is the total number of particles in the support domain; m_j and ρ_j are the mass and density of particle *j*, respectively. And also, it should be noted that the infinitesimal volume d**x'** is replaced by the finite volume $\Delta V = m_j / \rho_j$.

From (7), it is possible to infer that the approximate value of a function at any discrete point can be obtained using the weighted average of those values of the function at all other particles in the influence domain of that particle. Following similar argument, the particle approximation for a function at particle (point) *i* may be written as in (8) [1], [2]

$$A(x_i) = \sum_{j=1}^{N} \frac{m_j}{\rho_j} A(x_j)W(x_i - x_j, h) \quad (8)$$

On the other hand, particle approximation of the gradient and divergence of a given field function can be formulated following similar fashion. Transformation of the PDEs to the SPH discretized summation form, for instance, can be achieved by different ways. One way is with the help of integration by parts and Taylor series expansion. Suppose *A* is a scalar field function representing any physical variable and is defined in a given domain of interest. Its gradient can be formulated in a similar manner to (8) as

$$(\nabla A)_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} A_j \nabla_i W_{ij} \tag{9a}$$

where,

$$\nabla_i W_{ij} = \frac{x_i - x_j}{\left| x_i - x_j \right|} \frac{\partial W_{ij}}{\partial r_{ij}} = \frac{x_{ij}}{r_{ij}} \frac{\partial W_{ij}}{\partial r_{ij}} = \nabla_{xi} W_{ij} = -\nabla_{x'i} W_{ji} \tag{9b}$$

where, $W_{ij} = W\left(x_i - x_j, h\right)$

Applying some basics of vector calculus, other forms of the gradient equations can also be formulated. For instance, putting ρ inside the gradient operator and applying the chain rule, (9c) can be obtained. The introduction of mass and density into SPH particle approximation is to facilitate numerical approaches in hydrodynamic problems where density is a key parameter [1].

$$\nabla(\rho A) = \rho \nabla A + A \nabla \rho \Rightarrow \rho \nabla A = \nabla(\rho A) - A \nabla \rho \tag{9c}$$

And, re-writing in SPH particle approximation form

$$(\rho \nabla A)_i = \left(\sum_{j=1}^{N} m_j \left(A_i - A_j\right) \nabla_i W_{ij} \right) \tag{9d}$$

Again inserting ($\frac{1}{\rho}$) in the gradient operator and applying the chain rule

$$\nabla\left(\frac{A}{\rho}\right) = \rho \left[\nabla\left(\frac{A}{\rho}\right) + \frac{1}{\rho^2} A \nabla \rho \right] \tag{9e}$$

And, from (9e);

$$\frac{1}{\rho} \nabla A = \sum_{j=1}^{N} m_j \left(\frac{A_j}{\rho_j^2} + \frac{A_i}{\rho_i^2} \right) \nabla_i W_{ij} \tag{9f}$$

Note that the negative sign in (5) has been dropped in the above equations, because, here, the spatial derivative of the smoothing function ∇W is taken with respect to particle *i*, and not to particle *j*.

C. *Smoothing Functions*

In the preceding sections, it was indicated that SPH numerical method employs the theory of interpolation as its foundation. Smoothing (also, called weighting) function is, therefore, at the core of the SPH formulation. Spatial discretization of field variables is based on a set of points (particles, in SPH nomenclature), instead of grid nodes, which are commonly used in mesh-based numerical methods, such as FD and FE methods. It is, thus, with the use of kernel interpolation that field variables, such as velocity, pressure, density, stress, etc., are approximated at any point (i.e., at any discrete point) in the support domain. Accordingly, several kernel functions are being used in SPH numerical method. The use of the piecewise cubic-spline function, commonly known as the *B-spline*, suggested by Monaghan and Lattanzio (as cited in [1]), is popular among SPH numerical modelers. In the current work, however, a more or less similar cubic spline function, effectively applied to different modes of hydrodynamic conditions by SPHyscis code developers [3] is chosen. The same function (with slight variation in equality signs) was used in adaptive smoothed particle hydrodynamics (ASPH) in [1].

$$W(r,h) = \alpha_d \begin{cases} 1 - \frac{3}{2} q^2 + \frac{3}{4} q^3; & 0 \le q \le 1 \\ \frac{(2-q)^3}{4} & ; 1 \le q \le 2 \\ 0 & ; q \ge 2 \end{cases} \tag{10a}$$

where,

$q = \frac{\left| x_i - x_j \right|}{h} = \frac{r}{h}$ is the relative distance between the two points, x_i and x_j, $\left| x_i - x_j \right| = r$ is the distance between the two particles. And, α_d is given by $\frac{10}{7\pi h^2}$ and $\frac{1}{\pi h^3}$ for two and three dimensions, respectively.

The glaring shortcoming of spline functions is that their second derivative is a piecewise linear function, and, therefore, the stability properties can be inferior to those of smoother kernels [1]. This could, probably, be one of the reasons why the spatial first derivative of the cubic spline smoothing functions is widely used in the emerging literature instead of the second derivative. The spatial first derivative of *W(q)* for a two-dimensional case, thus, is given by

$$\nabla W(r,h) = \frac{10}{7\pi h^3} \begin{cases} -3q + \frac{9}{4} q^2; & 0 \le q \le 1 \\ \frac{-3}{4} (2-q)^2; & 1 \le q \le 2 \\ 0 & ; \quad q \ge 2 \end{cases} \tag{10b}$$

III. Applications: Runoff Modeling and Simulation

A. *Governing Equation*

At microscopic scale, the Navier-Stokes equation (sometimes also referred to as continuity and momentum equations) was employed for modeling free surface flows [4]. The Navier-Stokes (NS) equation was also widely used to model flow through porous media by several researchers, for instance, Jiang and Sousa [5], Morris et al. [6], Pereira et al. [7], and Lenaerts et al. [8]. The current study, also, seeks to model surface runoff down a saturated soil slope by solving the same NS equation, which is given in Lagrangian form

$$\frac{1}{\rho}\frac{D\rho}{Dt}+\nabla\cdot\boldsymbol{V}=0 \qquad (11)$$

$$\frac{D\boldsymbol{V}}{Dt}=-\frac{1}{\rho}\nabla P+\boldsymbol{g}+\nu\nabla^2\boldsymbol{V}+\boldsymbol{f} \qquad (12)$$

where, $\boldsymbol{V}$ is the velocity vector, P is the pressure, ρ is fluid density, $\boldsymbol{g}$ is gravitational acceleration, υ is fluid dynamic viscosity, $\boldsymbol{f}$ is other external forces vector, and t is time.

It is worth noting that SPH was originally invented for modeling flows of compressible fluids, and, thus its application to incompressible fluid flows needs some treatment to ensure density variation within a certain limit is maintained. In order to circumvent the difficulty of solving the pressure term for incompressible fluids, previous research works have resorted to using equation of state (EoS) as described in the next section.

B. Equation of State

For the standard SPH for compressible fluids, particle motion is triggered by pressure gradient, which is normally calculated using equation of state. However, for the case of incompressible fluids, applying and solving the pressure using an incompressible fluid EoS dictates the adoption of small timestep [1]. This constraint has led to the adoption of artificial compressibility for solving the pressure gradient in the governing equation, the approach which is dubbed quasi-incompressibility by some researchers. Accordingly, Monaghan [4] modified the EoS suggested for water by Batchelor (also cited in [4]), for describing sound waves and used it for simulating free surface flows, and the same equation has been frequently used by several emerging literatures. In this research too the same EoS is used as given in (16). Moreover, Bui et al. [9] applied the same EoS in their formulation of SPH for soil mechanics with successes.

$$P=B\left[\left(\frac{\rho}{\rho_o}\right)^{\gamma}-1\right] \qquad (13)$$

where, γ is constant and is taken to be a unity for incompressible low Reynolds number fluid flows [10], and 7 for other circumstances [4], ρ_o is the reference density, B is a problem dependent parameter for limiting the maximum density gradient and, in most cases, can be taken as the initial pressure [1], [10]. This paper assumes $\gamma = 1.0$

C. Boundary Treatment

Boundary treatment entails special consideration in SPH, as particle deficiency near or on the boundary impairs full exploitation of the scheme. Monaghan [4], also reported in [1], suggested the use of ghost particles near or on the boundaries so that high repulsive force is created to prevent fluid particles from unphysically penetrating a solid boundary. Such penalty force approach to prevent interior fluid particles from penetrating the boundary is based on the Lennard-Jones molecular force approach. Another approach, in which the Hertzian contact theory was used, was also developed by Bui et al. [9]. For the current research we intend to use the Monaghan's approach [4] as given in (14).

$$PB_{ij}=\begin{cases} D\left[\left(\frac{r_o}{r_{ij}}\right)^{a}-\left(\frac{r_o}{r_{ij}}\right)^{b}\right]\frac{x_{ij}}{r_{ij}^{\,2}} & \left(\frac{r_o}{r_{ij}}\right)\le 1 \\ 0 & elsewhere \end{cases} \qquad (14)$$

where, as in [1], a and b are taken to be 12 and 4, respectively, although Monaghan [4] proposed 4 and 2, respectively, with the conditions that a > b, always. He also suggested that a and b could also be taken as 12 and 6, respectively, without significant changes in the results. D is a problem dependent parameter and is usually taken to be the square of the largest velocity [1], and r_o is selected to be approximately equal to the initial particle spacing.

D. Time Integration

There are two major types of numerical integration algorithms - explicit methods and implicit methods. The explicit methods have several advantages, including ease of programming, little memory use and less computation. Their major limitation is that they are unstable for large timesteps. Implicit methods, on the other hand, can use a larger timestep and are very stable. In using the predictor-corrector method attempt is made to combine the best aspects of the two methods. The predictor-corrector algorithm consists of a predictor step and a corrector step in each interval. The predictor step predicts a new value, and the corrector step improves the accuracy of that value. The predictor step is undertaken only once while the corrector step is continued until the required level of accuracy is reached. There are several predictor-corrector methods though for the current paper we stick to the Euler predictor-corrector method (some prefer to call it modified Euler method). The procedure in applying the Euler predictor-corrector method is given as follows.

$$\phi^{(n+1)*}=\phi^{n}+f\left(t_n,\phi^{n}\right)\Delta t \qquad (15)$$

$$\phi^{(n+1)}=\phi^{n}+\frac{1}{2}\left[f\left(t_n,\phi^{n}\right)+f\left(t_{n+1},\phi^{(n+1)*}\right)\right] \qquad (16)$$

For the sake of stability, the timestep, Δt, needs to be checked against several stability requirements. Detailed reading regarding these stability conditions can be made in [1], [10].

IV. NUMERICAL EXAMPLES AND DISCUSSIONS

A numerical example of triggering surface runoff along a soil slope is given in the figures below (see Fig. 1.). The numerical formulation was developed based on the concept adopted in simulating free surface flows as in [3] and [4]. In the examples, boundary particles were generated to form geometric boundaries for the computation. All the boundaries (left, top, right, and bottom) were rendered impenetrable to the moving water particles.

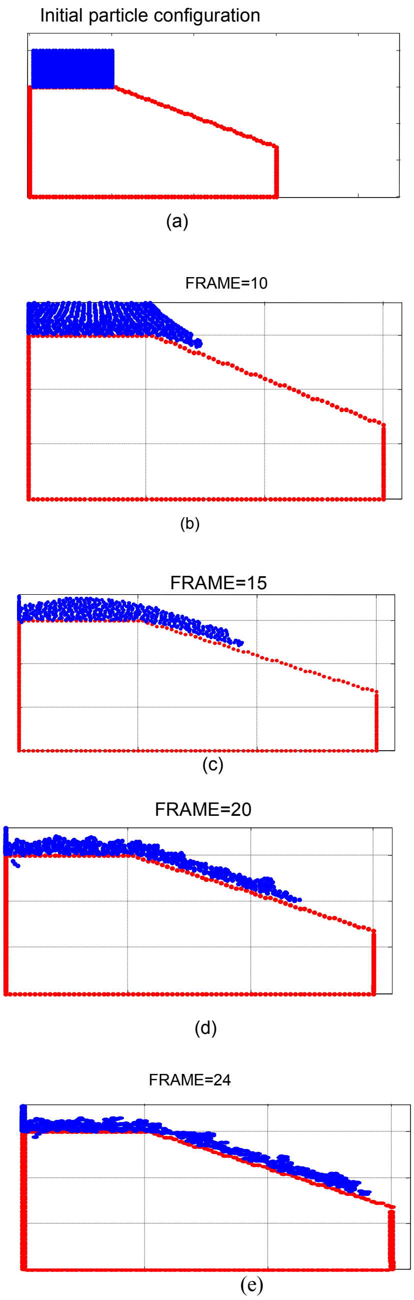

Fig. 1. Runoff simulation snapshots

In the above figures, the blue color represents the moving water particles, whereas, the red represents the impenetrable soil particles, which, in turn, represent geometric boundaries of the slope under investigation. Fig. 1(a) depicts the initial conditions in which rainwater of 480 mm depth was placed on the crest of the slope. Initial water particles were placed only on the crest of the slope purely in the interest of simplicity. Figs. 1(b) to 1(e) represent runoff simulation at successive timesteps for a simulation time of 0.5 seconds. SPH codes were developed in FORTRAN language for the simulation. The figures are self-explanatory in that water particles moving down a soil slope are depicted representing surface runoff. Calculation of flow parameters, such as flow velocities, pressures, and others were also undertaken as part of the overall investigation though not presented in the current paper.

V. Concluding Remarks

In this paper, it is highlighted that the Navier-Stokes equation can be effectively used to simulate surface runoff along a soil slope. Important flow parameters, such as flow velocity and pressure can be easily calculated considering each particle in the computational domain. The boundaries can be made either penetrable (as in the case of porous media) or impenetrable (as in the case of hard stratum\rock). As such, in the current research, the top of the soil slope was rendered impenetrable to water particles so that rainwater infiltration could be prevented. Experimental verification of the results is ongoing.

Acknowledgments

The first author would like to acknowledge the financial assistance received from Universiti Teknologi PETRONAS (UTP) through its Graduate Assistantship (GA) Scheme.

References

[1] G.R. Liu and M.B. Liu, Smoothed particle hydrodynamics – a meshfree particle method. World Scientific Publishing, Singapore, 2003.

[2] H.H. Bui, K. Sako, and R. Fukagawa, Numerical simulation of soil–water interaction using smoothed particle hydrodynamics (SPH) method. Journal of Terramechanics 44, 2007, 339–346.

[3] M.G. Gesteira, B.D. Rogers, R.A. Dalrymple, A.J.C. Crespo, and M. Narayanaswamy, User guide for the SPHysics code v2.0, 2010.

[4] J.J. Monaghan, Simulating free surface flows with SPH. Journal of Computational Physics, 110, 399-406, 1994.

[5] F. Jiang and A.C.M. Sousa, Smoothed particle hydrodynamics modeling of transverse flow in randomly aligned fibrous porous media. Transp porous med 75, 17-33, 2008.

[6] J.P. Morris, Y. Zhu, and P.J. Fox, Parallel simulations of pore-scale flow through porous media. Computers and Geotechnics 25, 227-246, 1999.

[7] G.G. Pereira, M. Prakash, and P.W. Cleary, SPH modeling of fluid at the grain level in a porous medium. Applied Mathematical Modeling 35, 1666–1675, 2011.

[8] T. Lenaerts, B. Adams, and P. Dutre, Porous flow in particle-based fluid simulations, ACM Trans. Graph. 27, 3, Article 49, August 2008.

[9] H.H. Bui, R. Fukagawa, and K. Sako, Smoothed particle hydrodynamics for soil mechanics. Numerical Methods in Geotechnical Engineering (eds.), Taylor & Francis, 2006.

[10] J.P. Morris, P.J. Fox, and Y. Zhu, Modeling low Reynolds number incompressible flows using SPH, Journal of Computational Physics 136, 214–226, 1997.