

An Interface to Model 3D Hydrological Process of a Soak-Away Rain Garden

Sivarajah Mylevaganam*, Ting Fong May Chui, Jianguong Hu

Department of Civil and Environmental Engineering, National University of Singapore, Singapore, Singapore
Email: [*sivaloga@hushmail.com](mailto:sivaloga@hushmail.com)

Received 17 March 2015; accepted 15 April 2015; published 17 April 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

To mitigate the adverse impact of urbanization around the world, several best management practices, in other words green infrastructures, have been used in a way that protect the natural hydrology of the catchment and are more beneficial to the environment. Soak-away rain garden, shallow, landscaped depressions commonly located in parking lots or within small pockets in residential areas, is one of those best management practices or green infrastructures. However, though in the past few decades the mathematical modeling work and the science of computer simulation have enhanced the understanding of hydrological processes, there is still a lack of modeling studies that focus on understanding the three-dimensional (3D) hydrological processes, such as the 3D ex-filtration process of a soak-away rain garden in a sufficient level of details. Thus, this paper develops a 3D mathematical model to represent the hydrological processes of a soak-away rain garden using COMSOL Multiphysics Java API, a Java-based interface of COMSOL Multiphysics. The developed model is demonstrated with a design hyetograph of 3-month average rainfall intensities of Singapore.

Keywords

COMSOL Multiphysics, 3D Ex-Filtration, Soak-Away Rain Garden, COMSOL Multiphysics Java API

1. Introduction

Urban development due to population growth and economic development has increased the impervious surfaces created by buildings and pavements. Consequently, increased stormwater runoff and volume generated from these impervious surfaces cause multiple problems affecting the sustainability of urban development. Some of

*Corresponding author.

the problems include, but not limited to, flooding, higher peak flows, shorter lag times, groundwater depletion, reduced base flow, increased transport of pollutants and nutrients, and erosion [1] [2]. To mitigate the adverse impact of urbanization around the world, several best management practices, in other words green infrastructures, have been used in a way that protect the natural hydrology of the catchment and are more beneficial to the environment [1]-[5]. Soak-away rain garden is one of those best management practices or green infrastructures.

Soak-away rain gardens, shallow, landscaped depressions commonly located in parking lots or within small pockets in residential areas, receive stormwater runoff, and attenuate surface water and enable it to percolate into the surrounding ground. Soak-away rain gardens are becoming important stormwater best management practices. Despite the rapid acceptance of soak-away rain gardens throughout the world by water managers and land-use planners, fundamental understandings of these green infrastructures are still at an undeveloped stage [1] [2]. Moreover, detailed hydrologic performance that depends on many highly dynamic conditions in three dimensions (3D) to which these facilities exposed is not currently available, partly due to the fact that in practice it is not feasible to measure a desired hydrological variable for every possible hydrological condition. On the other hand, a comprehensive mathematical modeling, which is the process of solving physical problems in the form of partial differential equations by appropriate simplification of reality, of soak-away rain gardens, though there are still a lack of modeling studies [2], can be a useful tool to provide a means to estimate output metrics for the purpose of guiding design and enhancing performance [1] [2]. In other words, a comprehensive mathematical modeling allows the user to test multiple designs under different scenarios. Moreover, with the advancement in the mathematical modeling work and the science of computer simulation, to overcome many of the difficulties in solving partial differential equations, and building blocks of modeling work, and to allow modelers to focus on solving the problem and not to spend many hours in dealing with the difficulties, there are abundant of software packages in the market, which make working with large models quicker and easier. COMSOL Multiphysics [6] [7] is one of those software packages that has been widely used in engineering applications as evidenced in many international conferences. Thus, it is the objective of this study to develop a 3D mathematical model to represent the hydrological process of a soak-away rain garden using COMSOL Multiphysics Java API, a Java-based interface of COMSOL Multiphysics.

2. COMSOL Multiphysics JAVA API

COMSOL Multiphysics, which uses the proven finite element method, is a powerful interactive environment for modeling and solving partial differential equations in scientific and engineering problems. With COMSOL Multiphysics the users can easily extend models for one type of physics into multiphysics models that solve coupled physics phenomena. The user can access the power of COMSOL Multiphysics as a standalone product (COMSOL Desktop) through a flexible Graphical User Interface (GUI) or by COMSOL Java API, a Java-based interface [6] [7]. In the COMSOL Java API, models are accessed through the model object, which contains all algorithms and data structures for a COMSOL model. The COMSOL Desktop also uses the model object to represent a model. This means that the model object and the COMSOL Desktop behavior are virtually identical but the model object through COMSOL Java API allows more flexibility and to overcome shortcomings of the COMSOL Desktop in customizing a specific problem of interest [6] [7]. Methods on a model object are used to create, modify, and access the model. The model object provides a large number of methods, including methods for setting up and running sequences of operations to create geometry, meshes, and for solving the model. The top-level methods just return references that support further methods. At a certain level the methods perform actions, such as adding data to the model object, performing computations, or returning data [6] [7]. Thus, there is a potential to use COMSOL Java API in customizing a specific problem of interest such as the hydrological process of a soak-away rain garden.

In Section 3.0, the important inner details on modeling a soak-away rain garden using COMSOL Java API is presented. In Section 4.0, the graphical user interface that is used to collect the user specific inputs, which are used inside the code (as discussed in Section 3.0) written using COMSOL Java API, of a soak-away rain garden is presented.

3. Mathematical Model of a Soak-Away Rain Garden Using COMSOL Java API

Modeling a soak-away rain garden involves flow in variably saturated porous media. Flow in variably saturated

porous media is modeled using the Earth Sciences Module (Subsurface Flow Module) of COMSOL Multiphysics. A review of COMSOL's Earth Sciences Module for simulating flow in variably saturated porous media can be found in [8]. Using the Earth Sciences Module of COMSOL, to develop a model to represent a soak-away rain garden, among other things, it is required to define the model geometry, the mathematical representation of the physical processes of interest, the boundary conditions, and the water balance of the soak-away rain garden.

3.1. Model Geometry Using COMSOL Java API

Figure 1 shows the 3D geometry of a soak-away rain garden that allows stormwater runoff to infiltrate into the surrounding soil. For demonstration purpose, a rain garden with a width of 5 m and a height of 1.0 m is chosen. The length of the rain garden is 5 m. The shape of the rain garden is assumed to be of rectangular/square shape.

The height of the filter media, the primary soil layer, is 0.6 m. The height of the ponding space, which is defined above the filter media, is limited to 0.2 m. The width and length of the area of the influence is 15 m. Using the COMSOL Java API, the 3D geometry of the rain garden of the above mentioned dimensions was formed by using Boolean operations (union, difference, etc.) on block geometries. A typical Model Java-file for the above discussed rain garden has the following structure.

```
import com.comsol.model.*;
import com.comsol.model.util.*;

public class TestingModel {

    public static void main(String[] args) {
        run();
    }

    public static Model run() {

        Model model = ModelUtil.create("Model");
        model.modelNode().create("geom1");
        ...

        //Create the geometry of the Soak-away Rain Garden
        //Create the model physics
        ...

        //Switching the boundary types
        ...

        return model;
    }
}
```

In the above code, the model object is defined on the COMSOL server and can be accessed using COMSOL JAVA API:

```
Model model = ModelUtil.create("Model");
```

The model object on the COMSOL server has the tag “model” while the variable “model” is its link in JAVA. The model object contains all the information about a model, from the geometry to the results. The creation of model geometry to represent the rain garden can be explained using the COMSOL JAVA API syntax. The create method of the geometry node requires as input a tag for the geometry name (“geom1”), and the geometry space dimension which is 3 for 3D:

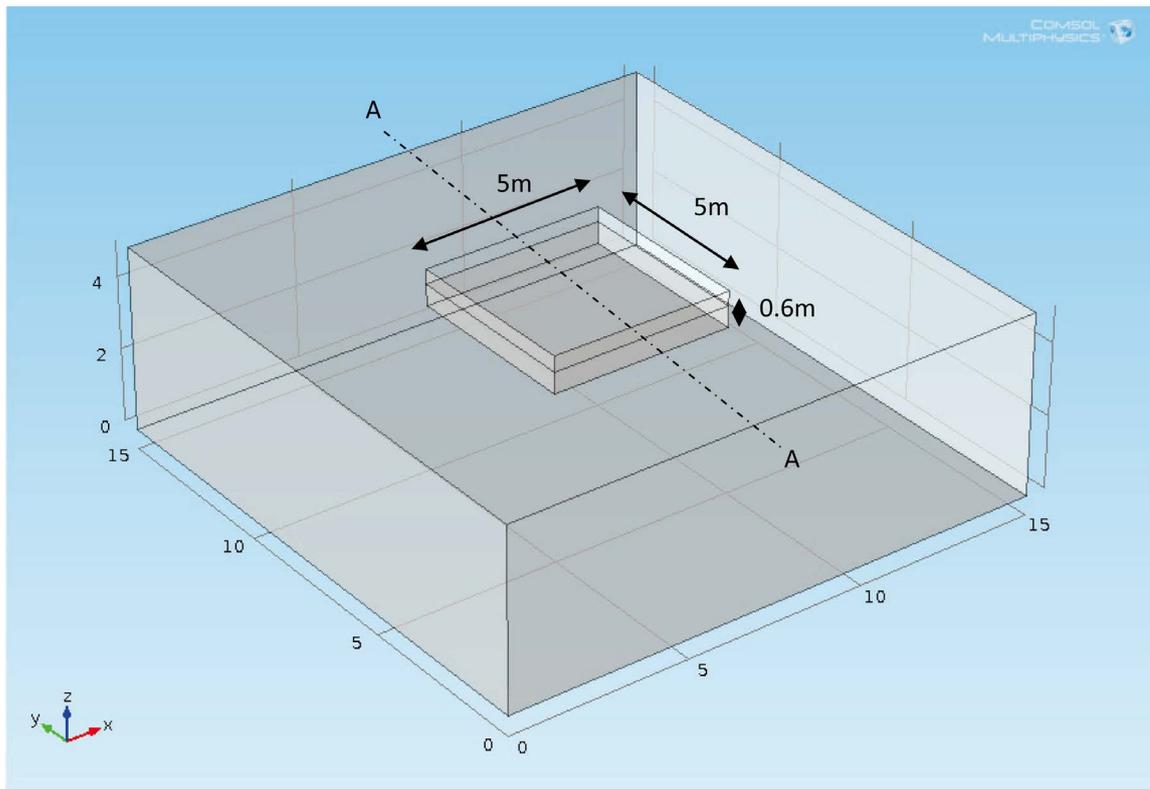


Figure 1. Graphical representation of a soak-away rain garden in COMSOL multiphysics.

```
model.geom().create("geom1", 3);
```

In this geometry object ("geom1"), a block feature, whose block name is "blk1", to represent the outer dimensions was created. The width, length, and the height of the block feature were set to "15", "15", and "5", respectively:

```
model.geom("geom1").feature().create("blk1", "Block");
model.geom("geom1").feature("blk1")
    .set("size", new String[]{"15", "15", "5"});
```

Similar to the block feature, "blk1", another block feature, whose block name is "blk2", was created. The width, length, and height were set to "5", "5", and "1", respectively, using the same syntax as before but the block position was set to (5, 5, 4):

```
model.geom("geom1").feature().create("blk2", "Block");
model.geom("geom1").feature("blk2")
    .set("pos", new String[]{"5", "5", "4"});
model.geom("geom1").feature("blk2")
    .set("size", new String[]{"5", "5", "1"});
```

To represent the *in-situ* soil, the block feature, "blk2", was subtracted from the block feature, "blk1", by creating a Difference feature ("dif1"), a Boolean operation, on the geometry object ("geom1") with the "input" property set to "blk1" and the "input2" property set to "blk2":

```
model.geom("geom1").feature().create("dif1", "Difference");
model.geom("geom1").feature("dif1").selection("input")
```

```

        .set(new String[]{"blk1"});
model.geom("geom1").feature("dif1").selection("input2")
        .set(new String[]{"blk2"});

```

To represent the filter media, similar to the block feature “blk1” and “blk2”, another block feature, whose block name is “blk3”, was created. The width, length, and height were set to “5”, “5”, and “0.6”, respectively, using the same syntax as before but the block position was set to (5, 5, 4):

```

model.geom("geom1").feature().create("blk3", "Block");
model.geom("geom1").feature("blk3")
        .set("pos", new String[]{"5", "5", "4"});
model.geom("geom1").feature("blk3")
        .set("size", new String[]{"5", "5", "0.6"});

```

To represent the rain garden, the block feature “blk3” was unioned with the difference feature (“dif1”), by creating a union feature (“uni1”), a Boolean operation, on the geometry object (“geom1”) with the “input” property set to “blk3” and “dif1”:

```

model.geom("geom1").feature().create("uni1", "Union");
model.geom("geom1").feature("uni1").selection("input")
        .set(new String[]{"blk3", "dif1"});

```

The complete code for the geometry creation of the soak-away rain garden resembles the following lines of code:

```

model.geom().create("geom1", 3);
model.geom("geom1").feature().create("blk1", "Block");
model.geom("geom1").feature().create("blk2", "Block");
model.geom("geom1").feature().create("dif1", "Difference");
model.geom("geom1").feature().create("blk3", "Block");
model.geom("geom1").feature().create("uni1", "Union");
model.geom("geom1").feature("blk1")
        .set("size", new String[]{"15", "15", "5"});
model.geom("geom1").feature("blk2")
        .set("pos", new String[]{"5", "5", "4"});
model.geom("geom1").feature("blk2")
        .set("size", new String[]{"5", "5", "1"});
model.geom("geom1").feature("dif1").selection("input")
        .set(new String[]{"blk1"});
model.geom("geom1").feature("dif1").selection("input2")
        .set(new String[]{"blk2"});
model.geom("geom1").feature("blk3")
        .set("pos", new String[]{"5", "5", "4"});
model.geom("geom1").feature("blk3")
        .set("size", new String[]{"5", "5", "0.6"});
model.geom("geom1").feature("uni1").selection("input")
        .set(new String[]{"blk3", "dif1"});
model.geom("geom1").run();

```

3.2. Representation of the Physical Processes Using COMSOL JAVA API

Richards’ equation [9] models flow in variably saturated porous media. Many efforts to simplify and improve the modeling of flow in variably saturated porous media have produced a number of variants of Richards’ equation.

In this paper, the form of the Richards' equation adopted in COMSOL Multiphysics is used [6] [7]. The Richards' equation was applied for both the *in-situ* soil and the filter media of the soak-away rain garden. The adopted equation is presented in Equation (1).

$$\rho \left(\frac{C_m}{\rho g} + S_e S \right) \frac{\partial p}{\partial t} + \nabla \cdot \rho \left(-\frac{k_s}{\mu} k_r (\nabla p + \rho g \nabla D) \right) = Q_m \quad (1)$$

where the pressure, p , is the dependent variable. In this equation, C_m represents the specific moisture capacity, S_e denotes the effective saturation, S is the storage coefficient, κ_s gives the hydraulic permeability, μ is the fluid dynamic viscosity, k_r denotes the relative permeability, ρ is the fluid density, g is acceleration of gravity, D represents the elevation, and Q_m is the fluid source (positive) or sink (negative). The fluid velocity across the faces of an infinitesimally small surface is given by Equation (2).

$$u = -\frac{k_s}{\mu} k_r (\nabla p + \rho g \nabla D) \quad (2)$$

where u is the flux vector. The porous medium consists of pore space, fluids, and solids, but only the liquids move. Equation (2) describes the flux as distributed across a representative surface. To characterize the fluid velocity in the pores, COMSOL Multiphysics also divides u by the volume liquid fraction, θ_s . This interstitial, pore or average linear velocity is $u_a = u/\theta_s$ [6] [7].

The creation of model physics to represent the physics of the soak-away rain garden can be explained using the COMSOL JAVA API syntax. The Physics node contains the settings of the physics interfaces, including the domain and the boundary settings. The create method of the physics node requires as input a tag for the physics name ("d1"), name of the physics which is "Richards Equation", and the geometry name ("geom1") which was created previously. In this physics node object, two features to represent the Richards Equation Models of the filter media and the *in-situ* soil were created. The names of these features are "remm1" and "remm2", respectively. The third argument of the create method, the space dimension, indicates at which geometry level (domain, boundary, edge, or point) the feature should be applied to. The features for Richards Equation Models are applied at domain levels, which have the space dimension 3. The domain selection for "remm1" and "remm2" are set as "1" (COMSOL generated entity index for the domain that represents the filter media) and "2" (COMSOL generated entity index for the domain that represents the *in-situ* soil), respectively.

```
model.physics().create("d1", "RichardsEquation", "geom1");
model.physics("d1").feature().create("remm1", "RichardsEquationModel", 3);
model.physics("d1").feature().create("remm2", "RichardsEquationModel", 3);
model.physics("d1").feature("remm1").selection().set(new int[]{1});
model.physics("d1").feature("remm2").selection().set(new int[]{2});
```

3.3. Boundary Conditions Using COMSOL JAVA API

To solve flow in variably saturated porous media, it is necessary that appropriate boundary conditions are specified. From a mathematical standpoint, the application of boundary conditions ensures that the solutions to the problems are self-consistent. In this study, the following boundary conditions are identified as appropriate. As shown in **Figure 2** which represents the frontal view of cut-plane (YZ Plane) A-A of **Figure 1**, the top surface of the rain garden is a rainfall-runoff boundary, a non-steady-state flow condition typical of urban storm-water runoff. The external side boundaries do not allow water to flow in or out of the area of influence, implying that the chosen area is large enough that it does not affect the flow performance around the rain garden. The bottom boundary of the area of influence is specified by a hydraulic head corresponding to an assumed groundwater table level.

The creation of boundary conditions to represent the boundary conditions of the soak-away rain garden can be explained using the COMSOL JAVA API syntax. In the physics node object, two features to represent the hydraulic head boundary at the bottom and flow (inlet boundary type) boundary at the top of the soak-away rain garden were created. The names of these features are "hh1" and "in1", respectively. The third argument of the create method, the space dimension, indicates at which geometry level (domain, boundary, edge, or point) the feature should be applied to. These features are applied to boundaries, which have the space dimension 2. The

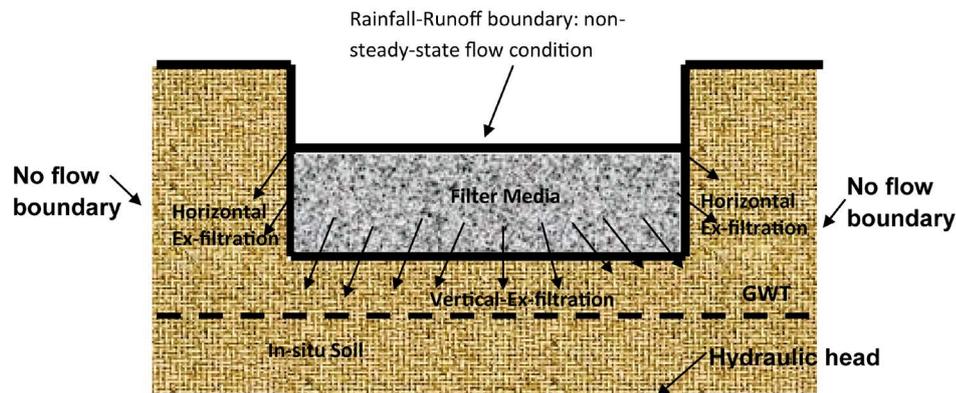


Figure 2. Boundary conditions of a soak-away rain garden in COMSOL multiphysics.

boundary selection for “hh1” and “in11” are set as “3” (COMSOL generated entity index for the boundary that represents the bottom of the soak-away rain garden) and “11” (COMSOL generated entity index for the boundary that represents the top of the soak-away rain garden), respectively.

```
model.physics("dl").feature().create("hh1", "HydraulicHead", 2);
model.physics("dl").feature("hh1").selection().set(new int[]{3});
model.physics("dl").feature().create("in11", "Inlet", 2);
model.physics("dl").feature("in11").selection().set(new int[]{11});
```

When water starts to pond, the boundary condition at the top surface of the rain garden becomes a hydraulic boundary. Therefore, there is a need to be able to switch the top surface of the rain garden from flow to hydraulic head. An additional feature to represent a hydraulic head at the top of the soak-away rain garden was created. The name of this feature is “hh2” and this feature is associated to the same boundary (“11”) as “in11”. In other words, the same boundary (“11”) is associated with two different boundary types: flow and hydraulic head boundary types.

```
model.physics("dl").feature().create("hh2", "HydraulicHead", 2);
model.physics("dl").feature("hh2").selection().set(new int[]{11});
```

The complete code for the creation of model physics to represent the physics of the soak-away rain garden and the associated boundary settings resembles the following lines of code:

```
model.physics().create("dl", "RichardsEquation", "geom1");
model.physics("dl").feature().create("remm1", "RichardsEquationModel", 3);
model.physics("dl").feature().create("remm2", "RichardsEquationModel", 3);
model.physics("dl").feature("remm1").selection().set(new int[]{1});
model.physics("dl").feature("remm2").selection().set(new int[]{2});
model.physics("dl").feature().create("hh1", "HydraulicHead", 2);
model.physics("dl").feature("hh1").selection().set(new int[]{3});
model.physics("dl").feature().create("in11", "Inlet", 2);
model.physics("dl").feature("in11").selection().set(new int[]{11});
model.physics("dl").feature().create("hh2", "HydraulicHead", 2);
model.physics("dl").feature("hh2").selection().set(new int[]{11});
```

3.4. Modeling Water Balance of the Rain Garden Using COMSOL JAVA API

Hydrologic phenomena are extremely complex, and may never be fully understood [10]. However, in the absence of perfect knowledge, they may be represented in a simplified way by means of the systems concept. A system is a set of connected parts that form a whole. As shown in **Figure 3**, the water balance of a soak-away

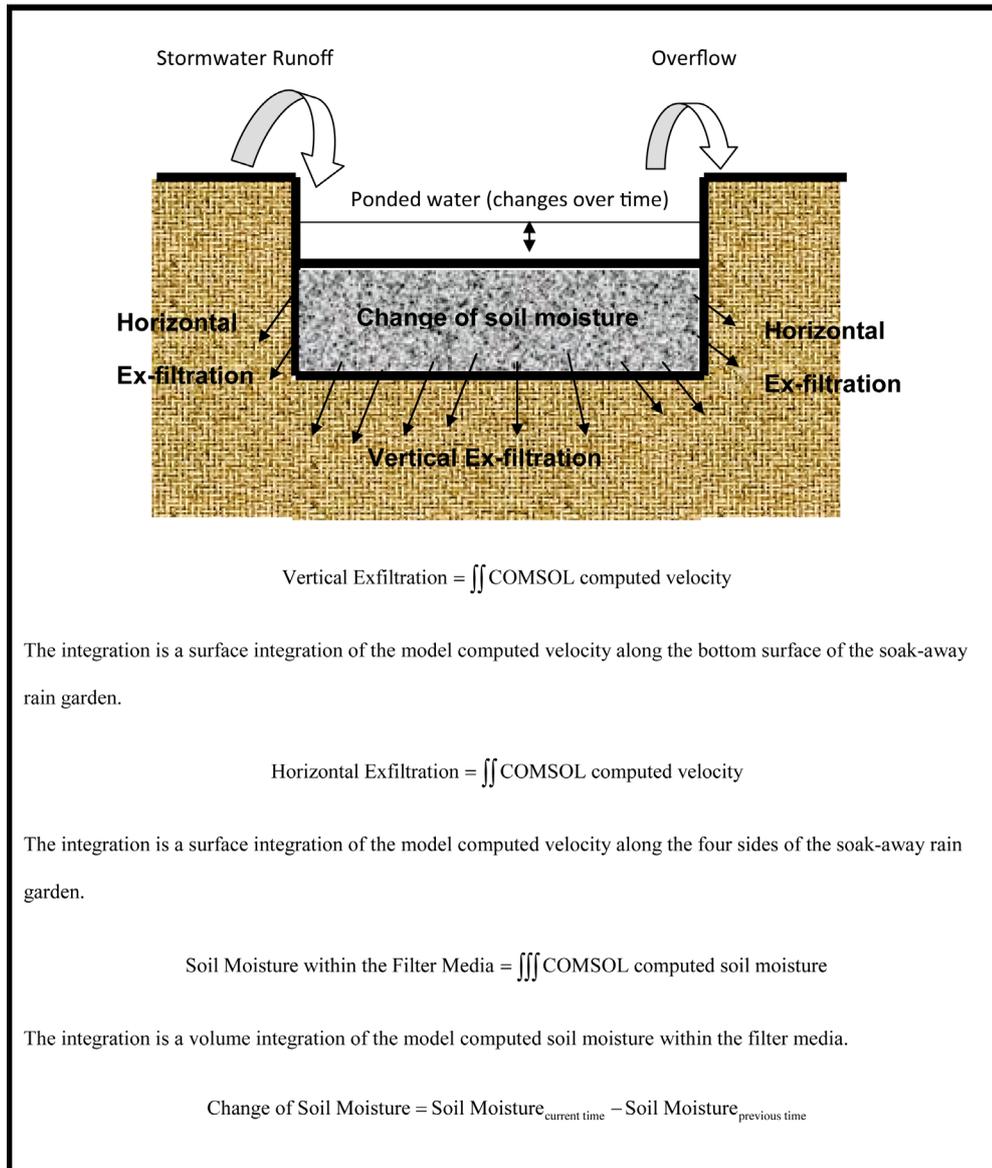


Figure 3. System concept of the soak-away rain garden.

rain garden can be conceived through the system concept whose components, which are linked to each others, are stormwater runoff, overflow, change of soil moisture within the filter media, change of water storage within the ponding space, vertical ex-filtration and horizontal ex-filtration. In simple terms, for a hydrological system, input $I(t)$, output $Q(t)$, and storage $S(t)$ are related by a continuity Equation (3).

$$\frac{dS}{dt} = I(t) - Q(t) \tag{3}$$

Unfortunately, for a soak-away rain garden, the continuity equation is too complex. For a soak-away rain garden, at a particular time, input $I(t)$ is the stormwater runoff. Output $Q(t)$, which depends on change of storage within the filter media and the ponding space, is a function of vertical ex-filtration, horizontal ex-filtration and overflow. On the other hand, storage $S(t)$, which depends on previous storage within the ponding space, vertical ex-filtration, horizontal ex-filtration, overflow, etc., is a function of storage within the filter media and the ponding space. Thus, to overcome the difficulties in handling the components that are linked to each others, the following approach is modeled. At a particular time, the current ponding depth, which is updated at the end of the si-

mulation after computing the vertical ex-filtration, horizontal ex-filtration, change of soil moisture within the filter media and overflow, is computed based on the previous ponding depth and the current runoff value. The computed current ponding depth is used as a boundary condition (at the top surface of the rain garden) to perform the simulation subjected to pass the conditional check on boundary switching. The simulated result is used to calculate the vertical ex-filtration (calculated by surface integration of the model computed velocity along the bottom surface of the soak-away rain garden), horizontal ex-filtration (calculated by surface integration of the model computed velocity along the four sides of the soak-away rain garden), and change of soil moisture within the filter media (calculated based on the previous soil moisture and the current soil moisture within the filter media). These computed values, which are based on the simulated result, are used to update the current ponding depth by considering the excess of the ponding space as the overflow. This procedure is continued until the end time of the simulation is reached.

4. Description of Functionalities of the Interface

Once the inner details were implemented taking into account the codes discussed in Sections 3.1, 3.2, 3.3, and 3.4, using COMSOL JAVA API, a graphical user interface was developed as shown in **Figure 4**. The interface permits the users to specify the dimensions of the soak-away rain garden, specify the model parameters, input the time series of the stormwater runoff, define the simulation period, define the meshing size, visualize the simulated results, and modify the default options.

As shown in **Figure 4**, the section for “Garden Dimension” allows the users to input the dimensions of the soak-away rain garden. The width, length, depth of filter media (the primary soil layer), ponding depth (defined above top of the filter media), and depth to ground water table that is defined from bottom of the filter media, are specified in meters. These values and some model generated intermediate values, which are based on these values, are used as JAVA variables at the appropriate places as discussed in Section 3.1. The section for “Model Parameters” allows the users to input the model parameters of the soak-away rain garden. In this section of the interface, the saturated hydraulic conductivity of the filter media (specified in mm/hr), which describes the ease with which water can move through pore spaces, saturated hydraulic conductivity of the *in-situ* soil (specified in mm/hr), porosity of the filter media and porosity of the *in-situ* soil are specified. These parameter values are

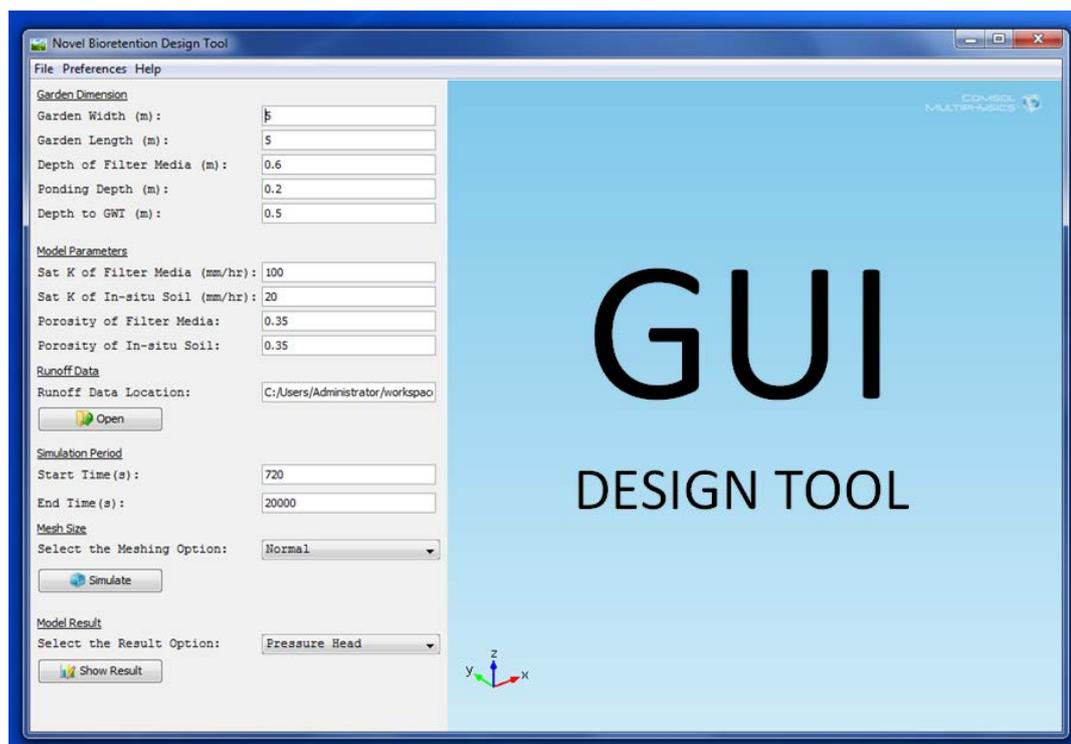


Figure 4. The graphical user interface of the 3D model.

used by the modeled physics, which is the Richards' equation, as discussed in Section 3.2. In addition, the user specified saturated hydraulic conductivities for filter media and *in-situ* soil are used to compute the saturated hydraulic conductivities in X, Y and Z-axis directions if a proportional an-isotropic tensor is specified through the preferences menu. The section for "Runoff Data" allows the users to input the stormwater runoff to the soak-away rain garden. The "Open" dialog box, which becomes visible when the "Open" button is clicked, allows the users to locate the time series data of the stormwater runoff. The time series data is a "tab" delimited text file that has the required data in two columns. The first column represents the time of the observation in seconds at equal intervals and the second column represents the runoff values in m^3/s . Often times, the time series of stormwater runoff (hydrograph) is obtained from a design hyetograph which is the time distribution of rainfall during a storm using a model (e.g., MUSIC) that uses the rainfall-runoff algorithm. The time series of stormwater runoff data is used to set the boundary condition as discussed in Section 3.4. The section for "Simulation Period" allows the user to input the start and end time of the simulation. The inputs are specified in seconds. The time step of the simulation is computed based on the user specified time series of stormwater runoff data. As mentioned previously, the stormwater runoff data should be at equal intervals. The section for "Mesh Size" allows the users to split the domains into smaller sub-domains. The governing equations of the underlying physics are then discretized and solved inside each of these sub-domains. As shown in **Figure 4**, the "simulation" button allows the users to perform the simulation. The model performs the simulation based on the user specified inputs that are discussed so far and any additional inputs specified through the preferences menu. The preferences menu allows the users to change some of the default settings that are used during the simulation. As an example, the preferences menu allows the users to change the proportional tensor of saturated hydraulic conductivities for filter media and *in-situ* soil. The successful execution of the model allows the users to visualize the simulation result such as hydraulic head, ponding depth: time series of the ponded water during the simulation period, overflow rate: time series of the overflow during the simulation period, vertical ex-filtration: time series of the water soaked away through bottom, horizontal ex-filtration: time series of the water soaked away through sides.

5. Application of the Model

The design of soak-away rain gardens involves water quality. Thus, the establishment of a design hyetograph for the design of soak-away rain gardens, specifically, requires data on intensity-duration-frequency (IDF) values for relatively frequent storms such as 3-month ARIs that carry up to 90% of the total load on annual basis. As underscored in the literature, to date, there are few methods available for the establishment of design hyetographs using IDF data [10]. In this paper, the alternating block method [10], which represents an event of a selected return period both for the selected duration of the event and for any period within this selected duration, is used in developing a design hyetograph from an IDF relationship of Singapore. A storm duration of 720 min was considered. Considering an event of 720 min of the 3-month ARIs, a design hyetograph for 3-month ARIs built-up using this method represents a 3-month ARI event both for the 720 min total duration and for any period (*i.e.*, 5 min, 10 min, 15 min, 30 min, 60 min, ..., 360 min) within this duration centered on the maximum block [10]. The design hyetograph produced by this method specifies the rainfall depth occurring in n successive time intervals of duration Δt over a total duration of $720 \text{ min} = n\Delta t$. Duration Δt is often determined by the finest resolution of the hydrological model that is used to generate the design hydrograph, the time distribution of discharge. The hyetograph to represent a 3-month ARI event of 720 min duration is shown in **Figure 5** for a duration (Δt) of 6 min which is the finest resolution of MUSIC (Model for Urban Stormwater Improvement Conceptualization) model which was used to generate the hydrographs for different urbanized (impervious percentage of 90% was assumed) catchment sizes varied from 100 to 250 m^2 . The time series of stormwater hydrograph was used to set the boundary condition as discussed in Section 3.4.

To understand the variation of horizontal ex-filtration with the *in-situ* hydraulic conductivity and the surface area of the soak-away rain garden (as % of catchment area), the simulation was carried out for different values of *in-situ* hydraulic conductivity and the surface area of the soak-away rain garden (as % of catchment area). The *in-situ* hydraulic conductivity was varied from 10 mm/hr to 50 mm/hr, typical range in Singapore. The surface area of the soak-away rain garden (as a % of catchment area) was varied from 6% to 15%. The width and the length of the soak-away rain garden were assumed to be of the same size. The saturated hydraulic conductivity of the filter media was set to 200 mm/hr, the maximum value expected in Singapore. The depth to groundwater table was set to 0.5 m.

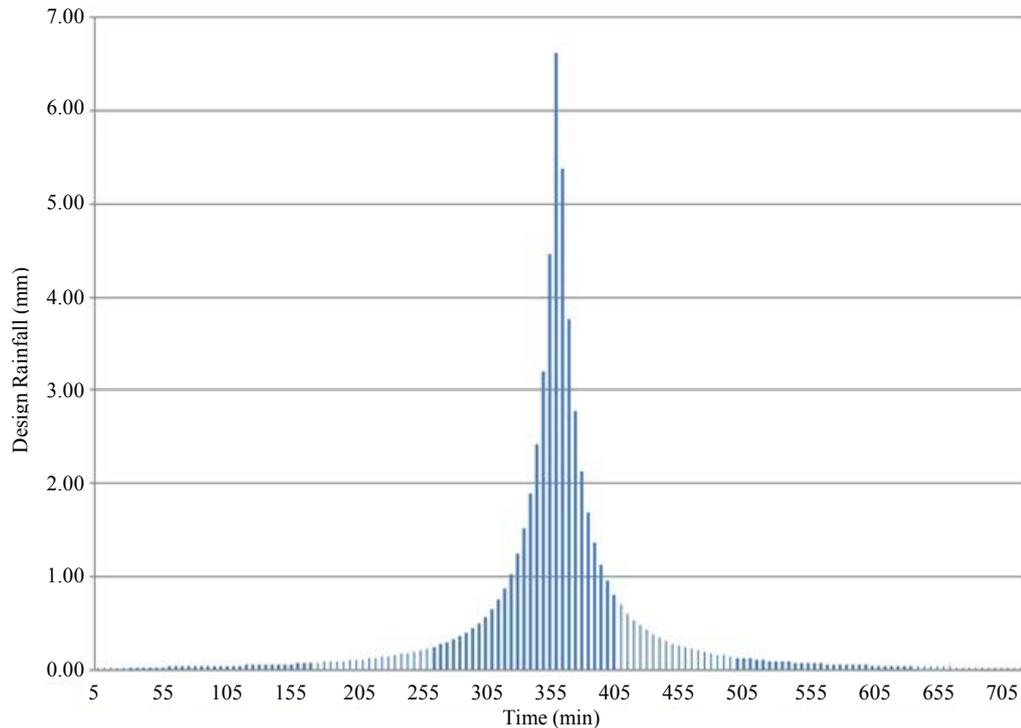


Figure 5. Design hyetograph for 3-month ARIs.

As shown in **Figure 6**, the graph of horizontal flow coefficient which is defined as the ratio between total horizontal ex-filtration (drained through sides of the soak-away rain garden, summed over the simulation period = 720 min, and expressed in m^3) and total vertical ex-filtration (drained through bottom of the soak-away rain garden, summed over the simulation period, and expressed in m^3) versus the surface area of the soak-away rain garden (as a % of catchment area) was plotted. The graph also shows the variation with saturated hydraulic conductivity of the *in-situ* soil. As can be observed from the graph, for the considered saturated hydraulic conductivities of the *in-situ* soil, as surface area of the soak-away rain garden (as a % of catchment area) increases, the horizontal flow coefficient decreases. The possible reason for this observation is that as surface area of the soak-away rain garden increases, it is expected to have more ex-filtrated water. In other words, it is expected that more vertical and horizontal ex-filtration to occur as surface area of the soak-way rain garden increases. However, as surface area of the soak-away rain garden increases, the increase in total vertical ex-filtration is generally higher than the increase in total horizontal ex-filtration. This is owing to the fact that as surface area of the soak-away rain garden increases, the increase in the base area is generally higher than the increase in the area in the side walls. This can be further explained by considering the simulated results of total horizontal ex-filtration and total vertical ex-filtration for a given *in-situ* hydraulic conductivity and for two different surface areas of the soak-away rain garden (say 4 m*4 m and 5 m*5 m). The total horizontal ex-filtration is $1.00\text{E}+01 \text{ m}^3$ and $1.21\text{E}+01 \text{ m}^3$ when the surface area of the soak-away rain garden is 4 m*4 m and 5 m*5 m, respectively. On the other hand, the total vertical ex-filtration is $6.98\text{E}+00 \text{ m}^3$ and $1.07\text{E}+01 \text{ m}^3$ when the surface area of the soak-away rain garden is 4 m*4 m and 5 m*5 m, respectively. Thus, the increase in total vertical ex-filtration is $1.53\text{E}+00$ ($1.07\text{E}+01/6.98\text{E}+00$) whereas the increase in total horizontal ex-filtration is $1.21\text{E}+00$ ($1.21\text{E}+01/1.00\text{E}+01$). This confirms that as surface area of the soak-away rain garden (as a % of catchment area) increases, the increase in total vertical ex-filtration is generally higher than the increase in total horizontal ex-filtration, and thus has made the horizontal flow coefficient to decrease. Further to this, the declining trend of **Figure 6** also reveals that although it is expected that more vertical and horizontal ex-filtration to occur as surface area of the soak-away rain garden (as a % of catchment area) increases, there exists a surface area of the soak-away rain garden (as a % of catchment area) beyond which the total horizontal ex-filtration becomes lesser than the total vertical ex-filtration. Furthermore, for a given surface area of the soak-away rain garden (as a % of catchment area), the horizontal flow coefficient decreases as the saturated hydraulic conductivity of the *in-situ* soil de-

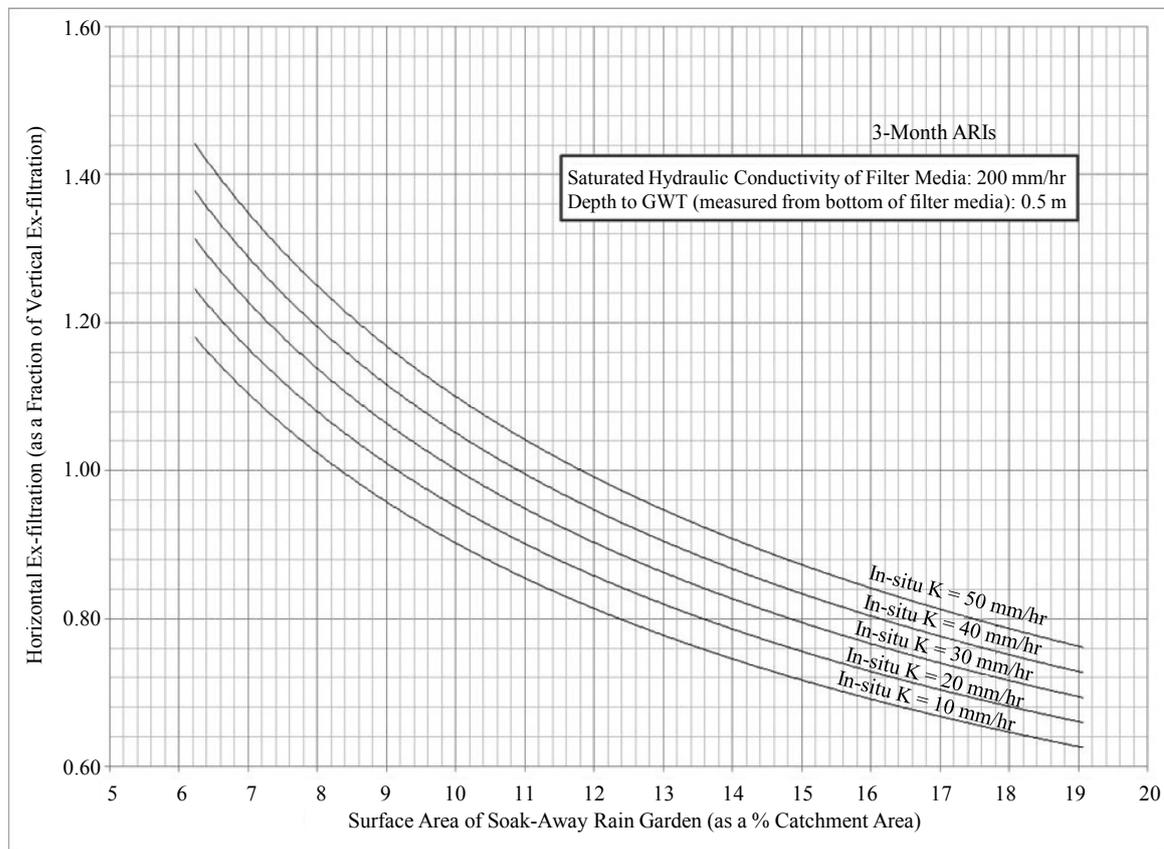


Figure 6. Horizontal flow coefficient for a depth to groundwater table of 0.5 m and a saturated hydraulic conductivity of the filter media of 200 mm/hr.

creases. The reason for this observation is that as the saturated hydraulic conductivity of the *in-situ* soil decreases, it is expected that the quantity of ex-filtrated water coming through vertically and horizontally to decrease. However, the amount of decrease for total horizontal ex-filtration is higher compared to the amount of decrease, which is negligible, for total vertical ex-filtration.

6. Conclusion

This paper presents a 3D model developed using COMSOL Multiphysics Java API to understand the hydrological processes such as the 3D ex-filtration process of a soak-away rain garden, shallow, landscaped depressions commonly located in parking lots or within small pockets in residential areas. The developed model is demonstrated with a design hyetograph of 3-month average rainfall intensities of Singapore. The application of the developed model will be of great utility for water managers to develop detailed hydrologic performance information and related hydrologic design guidelines or design charts of soak-away rain gardens for local conditions to control the flooding that endangers existing infrastructures and mimics the hydrology of the catchment to its nature.

Acknowledgements

The authors would like to thank PUB, Singapore’s national water agency, for providing the financial support to conduct this research. The author¹, who was a researcher at National University of Singapore, is currently attached to South Eastern University of Sri Lanka.

References

- [1] Allan, P.D., Robert, G.T. and William, F.H. (2010) Improving Urban Stormwater Quality: Applying Fundamental

- Principles. *Journal of Contemporary Water Research and Education*, **146**, 3-10.
- [2] Jia, L., David, J.S., Cameron, B. and Yuntao, G. (2014) Review and Research Needs of Bioretention Used for the Treatment of Urban Stormwater. *Journal of Water*, **6**, 1069-1099. <http://dx.doi.org/10.3390/w6041069>
- [3] Hunt, W.F., Jarrett, A.R., Smith, J.T. and Sharkey, L.J. (2006) Evaluating Bioretention Hydrology and Nutrient Removal at Three Field Sites in North Carolina. *Journal of Irrigation and Drainage Engineering*, **132**, 600-608. [http://dx.doi.org/10.1061/\(ASCE\)0733-9437\(2006\)132:6\(600\)](http://dx.doi.org/10.1061/(ASCE)0733-9437(2006)132:6(600))
- [4] Jones, M.P. and Hunt, W.F. (2009) Bioretention Impact on Runoff Temperature in Trout Sensitive Waters. *Journal of Environmental Engineering*, **135**, 577-585. [http://dx.doi.org/10.1061/\(ASCE\)EE.1943-7870.0000022](http://dx.doi.org/10.1061/(ASCE)EE.1943-7870.0000022)
- [5] Li, H., Sharkey, L.J., Hunt, W.F. and Davis, A.P. (2009) Mitigation of Impervious Surface Hydrology Using Bioretention in North Carolina and Maryland. *Journal of Hydrologic Engineering*, **14**, 407-415. [http://dx.doi.org/10.1061/\(ASCE\)1084-0699\(2009\)14:4\(407\)](http://dx.doi.org/10.1061/(ASCE)1084-0699(2009)14:4(407))
- [6] COMSOL AB (2012) COMSOL Multiphysics User's Guide (Version 4.3) Stockholm, Sweden.
- [7] COMSOL AB (2012) COMSOL Multiphysics Reference Guide (Version 4.3). Stockholm, Sweden.
- [8] Li, Q., Ito, K., Wu, Z., Lowry, C.S. and Loheide II, S.P. (2009) COMSOL Multiphysics: A Novel Approach to Ground Water Modeling. *Groundwater*, **47**, 480-487. <http://dx.doi.org/10.1111/j.1745-6584.2009.00584.x>
- [9] Richards, L.A. (1931) Capillary Conduction of Liquids through Porous Mediums. *Journal of Applied Physics*, **1**, 318-333. <http://dx.doi.org/10.1063/1.1745010>
- [10] Chow, V.T., Maidment, D.R. and Mays, L.W. (1988) Applied Hydrology. McGraw Hill, New York.