Jul 1st, 12:00 AM

Enhanced model flexibility and coupling opportunities: The WetSpa model case

Elga Salvadore

Jan Bronders

Okke Batelaan

Salvadore, Elga; Bronders, Jan; and Batelaan, Okke, "Enhanced model flexibility and coupling opportunities: The WetSpa model case" (2012). *International Congress on Environmental Modelling and Software*. 167.
https://scholarsarchive.byu.edu/iemssconference/2012/Stream-B/167

International Environmental Modelling and Software Society (iEMSs)
2012 International Congress on Environmental Modelling and Software
Managing Resources of a Limited Planet, Sixth Biennial Meeting, Leipzig, Germany
R. Seppelt, A.A. Voinov, S. Lange, D. Bankamp (Eds.)
http://www.iemss.org/society/index.php/iemss-2012-proceedings

Enhanced model flexibility and coupling opportunities: The WetSpa model case

Elga Salvadore[1,2], Jan Bronders[2], Okke Batelaan[1,3]
[1] VUB – Vrije Universiteit Brussel, Department of Hydrology and Hydraulic Engineering, Brussels, Belgium
[2] VITO – Flemish Institute for Technological Research, Environmental Modelling Unit, Mol, Belgium
[3] KULeuven – Katholieke Universiteit Leuven, Department of Earth and Environmental Sciences, Heverlee, Belgium
[1] Elga.Salvadore@vub.ac.be

Abstract: In the last decade spatially distributed hydrological models have become increasingly versatile due to the improvements in computational capabilities, the availability of digital data, and the use of Geographical Information System (GIS). Keeping up with technological development and new data types is becoming a challenge. Moreover, models are often developed to answer a specific need and thus lack flexibility, making it nearly impossible to further develop them. Using the principle of adaptability and expandability, we discuss an object oriented, modular model that is based on the fully distributed and physically based hydrological model WetSpa. The original WetSpa model has been disassembled into smaller and independent modules and it has been rewritten using the open source computer language Python. In the new structure three main components are identified: (1) a pre-process component for GIS maps preparation; (2) a Unit Class, representing the watershed discretized in raster cells or physical based objects; and (3) a Model Control Class, which organizes input/outputs and gives commands to the objects unit. The physically based processes, such as runoff, infiltration, interception, etc., are coded in separate scripts and organized via the Model Control Class. This approach allows the user to select which processes will be simulated and in which order, as well as to evaluate the impact of different parameterizations of the same process. The new model, WetSpaOO (Object Oriented) has been successfully tested on a theoretical case and, thanks to this new flexible structure, it has been coupled with the groundwater flow model MODFLOW.

Keywords: Hydrological modelling, WetSpa, modularity, Python, MODFLOW.

1 INTRODUCTION

Recent years have seen the development of spatially distributed models in hydrology due to the need of predicting spatially variable hydrological processes at a fine scale [Bellal et al. 1996, Rodriguez et al. 2003, Liu et al. 2003, Vieux et al, 2004]. The increased computational capability has boosted this process, cluster computing and web applications are more and more used [Choi et al. 2005, Fitch et al. 2009, Rouholanejad et al. 2012]. Furthermore digital data, such as Digital Elevation Models (DEM's), land use maps derived from remote sensing imaging and radar precipitation measurements, ask for model structure modifications in order to be fully incorporated into the calculations. New type of data are also available, examples are impervious surfaces maps and remote sensing derived parameter maps (i.e. leaf are index), so that new parameterizations and equations need to be tested for data assimilation. Therefore, keeping up with technological development and new research questions is becoming a challenge.

In the field of water management, and in particular for urban areas, one important line of research is pointing towards integration [Mahepala et al. 2005]. Many models are able to describe in detail specific physical processes, thus it becomes more and more interesting to couple different models in order to obtain a more general and balanced model in which hydrological processes can be simulated with a consistent level of detail. Model coupling is not an easy task as most models have been developed with the intention of answering a specific research question, hence they mostly lack a vision towards other models and have limited flexibility. Moreover, older codes have been revised generally several times by different researchers who worked on the same code to improve one specific aspect (i.e. time step options, new equations, spatial resolution) without taking into account the full model structure and equations. The result is often a code which is very complex and thus prone to bugs.

In this research, an example of a complex hydrological model WetSpa (Water and energy transfer between Soil plant and atmosphere) is presented in section 2. The objective of this paper is to show a simple but effective method and its application to improve the adaptability and expandability of the WetSpa model. The method, discussed in section 3, consists of the development of an object oriented, modular version of the WetSpa model (WetSpaOO, Object Oriented). This new version of the model fulfills the requirement of flexibility and it also allows aspects as: (1) assimilation of remote sensing data (imperviousness maps), (2) flexible time step, (3) modularity (possibility of testing several parameterizations of the same process), and (4) enhances coupling opportunities (PCRaster-MODFLOW [McDonald et al. 1988, Schmitz et al. 2009]). An example of application is given in section 4 where a theoretical case study is implemented for testing the flow routing component.

2 MATERIALS

In section 2.1 the WetSpa hydrological model is described in its original form, highlighting its structure, processes, capabilities and limitations. The computer language used for the modification of the WetSpa code (Python) and the groundwater model (MODFLOW) that has been coupled with the WetSpaOO model are briefly discussed in section 2.2.

2.1 The WetSpa model

The WetSpa model is a GIS-based, spatially distributed hydrological model for rainfall-runoff simulations at the catchment scale. It has been developed by the Department of Hydrology and Hydraulic Engineering of the Vrije Universiteit Brussel [Batelaan et al. 1996, De Smedt et al. 2000, Liu et al. 2003]. The model can simulate several physical processes at the raster cell level, such as: interception, depression storage, evapotranspiration, runoff, interflow, groundwater recharge and groundwater flow (at sub-catchment level), see Figure 1 for an overview of the model water balance at cell level. The inputs of the model are: precipitation and potential evapotranspiration time series, which are distributed over the catchment with the Thiessen polygons method; spatially distributed parameters are derived with help of the GIS software ArcView 3.2 on basis of three basic maps: Topography, Soil texture and Land use. Typical outputs are: flow hydrographs at catchment and sub-catchment outlets, maps of evapotranspiration, soil moisture, interception, surface runoff, groundwater recharge and interflow for each time step.

The WetSpa model consists of a pre-process part developed in ArcView Avenue language and three compiled Fortran programs: mean.exe, iuh.exe and model.exe. The mean program averages all the parameters at the sub-catchment level, iuh calculates a series of instantaneous unit hydrographs, which are then used for diffusive wave approximated flow routing. Then the model.exe program reads the time series and parameter inputs, performs the actual simulations for a user defined time step (ranging from one hour to days, months and years) and writes the required results. The model also requires eleven physically based global parameters which can be tuned in the calibration phase. Calibration can be manually or automatically performed, the automatic calibration is performed with the model independent Parameter ESTimator (PEST) [Liu et al. 2005].

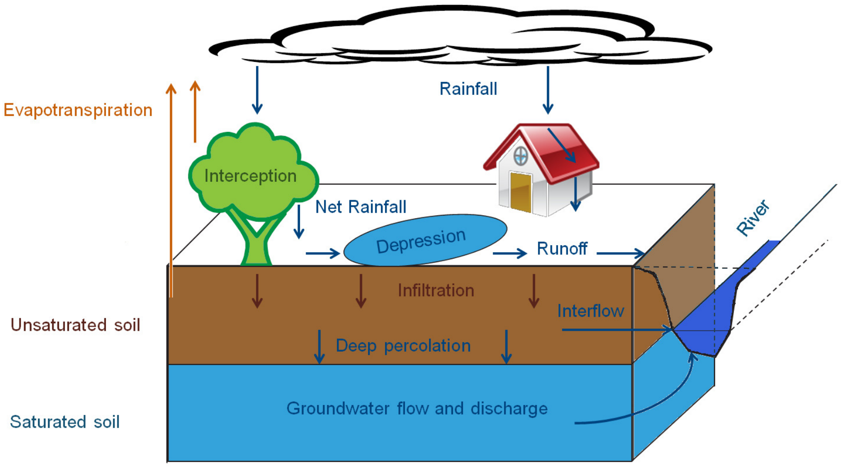

Figure 1: WetSpa model perform every time step a water balance at cell level, the main considered processes are here illustrated.

Main limitations of the models are:
(1) Lack of flexibility. All the processes are coded in a single Fortran script, thus, in order to modify a certain parameterization/process, the model developer must identify all the explicit and implicit links that are present in the full code. The processes and their computational order are not easily modified, so that the evaluation of model sensitivity to processes and their parameterizations is nearly impossible;
(2) The simulation time step cannot be less than one hour, which is unsuitable for capturing the behaviour of small watersheds and of flash-floods events;
(3) Groundwater flow is calculated at a sub-catchment level using a linear reservoir approach, hence the spatial distribution of the groundwater table cannot be predicted. The groundwater layer can only receive groundwater recharge from the unsaturated zone and the groundwater flow is used to predict base flow at the sub-catchment outlets, no interaction is possible with the vegetation layer;
(4) The model assumes that the groundwater table is always below a fixed unsaturated depth (function of the land use class); this makes the use of the model in wetland areas less appropriate and no feedback is possible with the unsaturated zone;
(5) The impervious fraction is derived as a user defined percentage of urban land use class, which may not reflect the real situation.

2.2 Python and PCRaster-MODFLOW

Python is a high level, object-oriented, interpreted computer language that is getting more and more popular as the number of major organizations, such as NASA, Google, and Disney are developing parts of their code in this language [Python 2009]. It is often compared with the widely used Matlab language for their many similarities, however one evident difference is that Python is open source (distributed with a GNU licence). Python has also a standard library of modules that can be used for all sort of tasks ranging from web-programming to graphics ("batteries included"). Within these modules it is worth mentioning the main ones that are used in this research: NumPy, for numerical calculation (arrays and matrices), matplotlib for graphical representation of results, PCRaster for GIS maps calculations [PCRaster 2005], and PCRaster-MODFLOW. PCRaster can be used as a stand-alone software for dynamic simulations on raster maps, but in this study only some of its functions for map manipulation have been used, thus like a classical GIS application (in this case a Python alternative to ArcView). PCRaster has also a module, PCRaster-MODLFOW [Schmitz et al. 2009], that allows the use of certain packages of the MODFLOW 2000, the widely spread 3D finite-difference groundwater flow model. In the PCRaster-MODFLOW extension for Python MODFLOW core is wrapped into PCRaster-Python so that there is no need for input/output manipulations. The main advantage of this approach is that modelling integration is possible, in contrast with other MODFLOW pre- and postprocessors. The MODFLOW packages used in this study are: DIS package (grid and several input parameters definition), BAS package (inputs for the basic package, boundaries and initial conditions), BCF package (parameters definition for block-centered flow package, such as vertical and horizontal hydraulic conductivity), RCH package (areally distributed groundwater recharge), DRN package (use of drains in the simulation), and one of the four available solver packages (PCG).

3 SOFTWARE DESIGN

An object oriented, modular model (WetSpaOO, Object Oriented) has been developed based on the equations of the WetSpa model. The original WetSpa model has been disassembled into smaller and independent modules and it has been rewritten using the open source computer language Python. In the new structure three main components can be identified: (1) a pre-process component which takes care of the preparation of GIS maps; (2) the Unit Class which represents the watershed discretized in raster cells or a physical based object to simulate; and (3) a Model Control Class, which organizes input and outputs and gives commands to the objects unit (Figure 2). The physically based processes, such as runoff, infiltration, evapotranspiration, etc. are coded in separate scripts and organized in the Step script. This allows the user to select which processes will be simulated and in which order, as well as to evaluate the impact of different parameterizations of the same process by developing alternative codes for the same process.

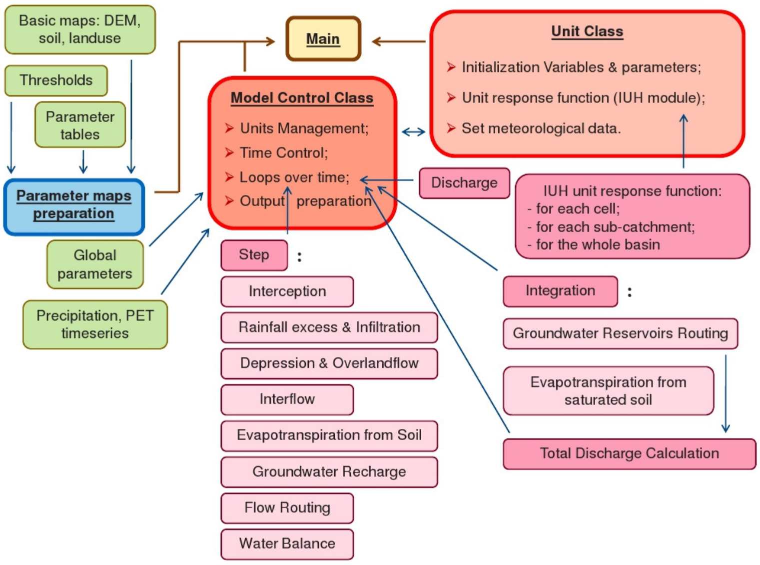

Figure 2: Conceptual structure of WetSpaOO model: green boxes = inputs, blue box = GIS module, red boxes = two main classes of the model, and pink boxes = separate model components describing the physical processes.

3.1 GIS Pre-process

Starting from three basic maps (Elevation, Soil and Land Use), a set of parameter tables and a series of thresholds, the GIS pre-process estimates all the necessary parameter maps to perform spatially distributed simulations with WetSpaOO. The

GIS component can also be used as a stand-alone static model for hydrogeomorphological analysis of watersheds. This model component is developed in Python language taking advantage of the PCRaster GIS library of functions. Example of output parameter maps are: flow direction, minimum and maximum interception capacity, Thiessen polygons (for precipitation and potential evapotranspiration), manning coefficient, potential runoff coefficient, mean flow velocity and travel time. These maps are then saved as .map (PCRaster format) and as ascii files.

3.2 Class Unit

The Unit class is the modelling representation of a study area. The main action that an object of such a class can perform is: transformation of input time series (precipitation and potential evapotranspiration) in spatially distributed input (via Thiessen polygons maps) if no radar maps are available; and the unit response function calculation by calling the *IUH* component which evaluates the hypothetical response (based on statistical analysis) of each cell, sub-basin and of the whole watershed to a unit impulse of net rainfall. These potential actions are coded as methods of the Unit class and their effect is a change in the Unit object attributes.

3.3 Class Model Control

The Model Control Class is the core of the model, the object derived from this class manages the unit objects (modeling representation of the river basin). It takes care of time issues and time loops for dynamic simulation and it is the responsible component for output processing and data storage. Three loops are coded in three methods: (1) the basic loop for the water balance at cell level where the *Step* script is called, (2) the integration loop where the groundwater routing is performed at integrating information at sub-catchment level, and (3) a final loop where the total discharge is calculated as a sum of surface runoff, interflow and groundwater discharge components.

3.4 Simulation modules

Eight simulation modules are available for WetSpaOO model: Interception, *RainExcess_Infiltration*, *Depression_Overlandflow*, *Interflow*, *Evaporation_soil*, *Gw_recharge*, *Flow_Routing*, and *Balance*. In the *Step* script these calculation modules are organized (cascade of processes). If, for a specific case study, a process has a minor importance and can be neglected, it can simply be switched off. If a new script is available and needs to be tested it can just be added to the list of processes in the *Step* script with no further code changes. Two extra modules are available for sub-catchment processes simulations: *Groundwater_reservoirs* and *Evaporation_Sat_Soil*. These two components are listed in the *Integration* script. The same considerations made for the *Step* script are valid for the *Integration* script. Simulation modules have a standard structure, they are classes with one or more static methods. These methods receive as arguments the unit object and, if required, extra parameters (for instance the day of the year for interception calculations), and perform calculations by modifying attributes (variables) of the Unit object.

3.5 WetSpaOO and MODFLOW

In order to couple WetSpaOO and MODFLOW (PCRaster-MODFLOW), two new scripts have been developed: *groundwater_mf*, an alternative to the *Groundwater_reservoirs* simulation module in the *Integration* script, and *gr_postprocess*. *Groundwater_mf* is a class, similar to the simulation module classes, in this case composed of two static methods: *initialize* and *calculate*. The *initialize* method is executed only once to set the boundary and initial conditions, select the solver and its parameters as well as drain elevation and conductivity. The *calculate* method read recharge maps, calculated by previously computed modules, as input for groundwater flow simulations and runs MODFLOW 2000. Head and drains outflow can be retrieved as outputs. The *gr_postprocess* script takes care of time conversions; if the simulation time step of surface processes is for instance 1 hour and the time step of the groundwater model 1 day, the script performs interpolation and extract a time dependent base flow, which will be then summed with the other components of the hydrograph. The MODFLOW module for WetSpaOO is currently in the testing phase, for this reason in the next section (Application) will only be discussed the verification of surface runoff component of the WetSpaOO model.

4 APPLICATION

4.1 The V-catchment case study

Two-dimensional areal surface flow for WetSpaOO has been verified using the rainfall-runoff example proposed by Di Giammarco [Di Giammarco et al., 1996]. Van der Kwaak presents details of simulation, with results from various surface water flow codes benchmarked against this problem which are used as a verification for the WetSpaOO model outputs [Van der Kwaak 1999].
Problem characteristics:

- Two-dimensional surface flow from a totally impervious V-catchment (see Figure 3 for a graphical representation of the catchment);
- Precipitation intensity: $3 \cdot 10^{-6}$ m/s for 90 minutes followed by 90 minutes of no rain;
- Due to symmetry only half of the catchment is simulated with cell size of 10 by 10 m (simulation domain 1000 m by 800 m);
- Surface slope: 0.05 and 0.02 perpendicular to, and parallel to the channel respectively;
- Manning's roughness coefficient of 0.15 and 0.015 are applied to the slopes and the channel, respectively;
- Simulation time step: 1 min.

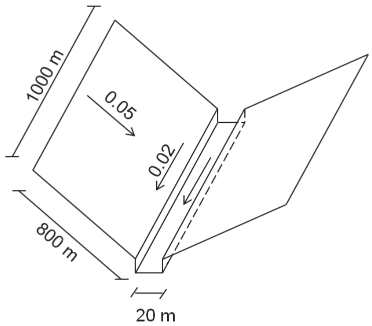

Figure 3: V-cathcment, modified from Di Giammarco et al., [1996].

In the V-catchment, sub-surface processes do not occur (fully impervious), thus to perform WetSpaOO simulations some pre-process parameters are set to zero (soil moisture, porosity, conductivity, and depression storage capacity), in the Step script the processes of depression storage, interflow, evapotranspiration from soil, and groundwater recharge are deactivated. The Integration script is also modified by deactivating the groundwater flow routing.
The WetSpaOO model performs sensibly better than the original WetSpa model (Figure 4). The original model is not able to capture the behavior of the V-catchment at the start and the end of the precipitation. The reason for this is associated with the time step limitation of the model (1 hour) for the calculation of the instantaneous unit hydrograph which strongly influences the routing process.

The original model was forced with input resolution of hours fractions but the impact of the surface routing response function (hourly based) remains the most influencing factor in this type of theoretical case. The WetSpaOO model overcomes this limitations by allowing user defined time steps for both input files and routing component. Good agreement in fact is noted if comparing river discharge predicted by WetSpaOO with predictions from several other codes, in particular in the time of concentration of the catchment (around 60 minutes) and mass conservation is preserved with a plateau of about 2.4 m^3/s. The model however seems to respond faster to the start of the rainfall as well as to the end of it. The cause is probably related to the theoretical choice of using the IUH to simulate the routing of water through the watershed, where the hydraulic radius and so the mean flow velocity (calculated with Manning equation) are assumed to be spatially variable but temporally constant. At the beginning (end) of the rain event in fact the water depth in each cell is very low and so the hydraulic radius, if we assume a constant radius equal to the average condition, an overestimation of the velocity is observed. A higher velocity produces a bigger water discharge, the time to peak is therefore shorter and the flood recession faster.

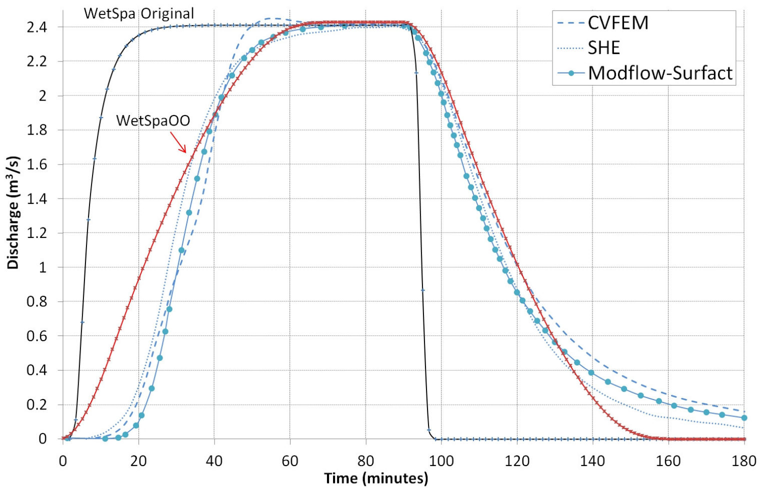

Figure 4: WetSpaOO model results are in agreement with other hydrological models, and the new model significantly over-perform the original WetSpa model for simulation of 2-D surface water flow theoretical case study.

5 CONCLUSION

In this paper we discussed a simple but effective method to improve model flexibility and coupling opportunities by describing the object oriented, modular WetSpaOO model example. The original WetSpa model was disassembled into smaller and independent modules and rewritten in Python language. The application examples showed that the new structure of the model is more versatile: (1) simulated processes can be selected in function of the typology of study, and (2) coupling with other models, for instance a 3D groundwater simulation model, is relatively straightforward by simply replacing or adding a new script. The first allows the user to evaluate the impact of a slightly different model structure on simulation results and to decrease model complexity while the second opens many opportunities for further development of the model.

More testing and applications of the WetSpaOO-MODFLOW coupling is however needed; future direction on this topic is the development of a more interlinked connection between the models, where the groundwater conditions can also affect the surface hydrology behaviour. Ongoing work includes the development of a different surface water routing component to overcome the discrepancies between WetSpaOO simulations and other benchmark tests.

ACKNOWLEDGMENTS

This research was funded by a FWO (Research Foundation Flanders) and VITO (Flemish Institute for Technological Research) PhD grant, with the contribution of the Department of Hydrology and Hydraulic Engineering of the Vrije Universiteit Brussels and the Environmental modelling Unit of VITO. Finally we pay regards to the reviewers for their valuable comments.

REFERENCES

Batelaan, O., Z.M. Wang, and F. De Smedt, An adaptive GIS toolbox for hydrological modeling, *Application of geographic information systems in hydrology and water resources management*, IAHS Publ. 3-9, eds. Kovar, K. & Nachtnebel, H.P., 1996.

Bellal, M., X. Sillen, and Y. Zech, Coupling GIS with a distributed hydrological model for studying the effect of various urban planning options on rainfall-runoff relationship in urbanized watersheds, *Application of Geographical Information Systems in Hydrology and Water Resources Management*, IAHS Publ. no. 235, eds. Kovar, K. & Nachtnebel, H.P., 1996.

Choi, J.-Y., B.A. Engel, and R.L. Famsworth, Web-based GIS and spatial decision support system for watershed management, *Journal of Hydroinformatics*, 7.3, 165-174, 2005.

De Smedt, F., Y.B. Liu, and S. Gebremeskel, Hydrological modeling on a catchment scale using GIS and remote sensed land use information, *Risk Analysis II*, WIT press, Southampton, Boston, ed., Brebbia, C.A., 2000.

Di Giammarco, P., E. Todini, and P. Lamberti, A conservative finite elements approach to overland flow: the control volume finite element formulation. *Journal of Hydrology*, 175, 267-291, 1996.

Fitch, P., and Q. Bai, A standards based web service interface for hydrological models, *18th World IMACS/MODSIM Congress Australia,* 873-879, 2009.

Liu, Y.B., S. Gebremeskel, F. De Smedt, L. Hoffmann, and L. Pfister, A diffusive transport approach for flow routing in GIS-based flood modeling, *Journal of Hydrology*, 283, 91-106, 2003.

Liu, Y.B., O. Batelaan, F. De Smedt, J. Poórová, and L. Velcická, Automated calibration applied to a GIS-based flood simulation model using PEST, *Floods, from Defence to Management, 317-326,* 2005.

Maheepala, S., B. Leighton, F. Mirza, M. Rahilly, and J. Rahman, Hydro Planner – a linked modeling system for water quantity and quality simulation of the total water cycle, *MODSIM2005 International Congress on Modelling and Simulation*, 683-689, 2005.

McDonald M.G., and A.W. Harbaugh, A modular three dimensional finite-difference groundwater flow model, *US Geological Survey Techniques of Water Resource Investigation*, USGS, Washington, USA, chapter AI, 1988.

PCRaster, 2009. PCRaster website. URL: http://pcraster.geo.uu.nl/

Python, 2009. Python programming language website. URL: http://www.python.org/

Rodriguez, F., H. Andrieu, and J.-D. Creutin, Surface runoff in urban catchments: morphological identification of unit hydrographs from urban databanks, *Journal of Hydrology*, 283, 146-168, 2003.

Rouholahnejad, E., K.C. Abbaspour, M. Vejdani, R. Srinivasan, R. Schulin, and A. Lehmann, A parallelization framework for calibration of hydrological models, *Environmental Modelling & Software*, 31, 28-36, 2012.

Schmitz, O., D. Karssenberg, W.P.A. van Deursen, and C.G. Wesseling, Linking external components to a spatio-temporal modelling framework: Coupling MODFLOW and PCRaster, *Environmental Modelling & Software*, 24, 1088-1099, 2009.

VanderKwaak, J., 1999. Numerical Simulation of Flow and Chemical Transport in Integrated Surface-Subsurface Hydrologic Systems. Ph.D. Thesis in Earth Sciences, University of Waterloo, Waterloo, Ontario, Canada, 217 pp.

Vieux, B.E., and P.B. Bedient, Assessing urban hydrological prediction accuracy through event reconstruction, *Journal of Hydrology*, 299, 217-236, 2004.