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

Improved dynamic emulation modeling by time series clustering: the case study of Marina Reservoir, Singapore

S. Galelli [a], **S. Caietti Marin**[b], **A. Castelletti**[b], **H. Eikaas**[c]

[a] *Singapore-Delft Water Alliance, National University of Singapore, 2 Engineering Drive 2, 117577, Singapore (sdwgs@nus.edu.sg)*

[b] *Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza L. da Vinci 32, 20133, Milano, Italy (stefania.caietti@mail.polimi.it, castelle@elet.polimi.it)*

[c] *Catchment and Waterways Department, Public Utilities Board, 40 Scotts Road, Singapore (eikaas_hans@pub.gov.sg)*

Abstract: Dynamic Emulation Modelling (DEMo) is emerging as a viable solution to combine computationally intensive simulation models and dynamic optimization algorithms. A dynamic emulator is a low order surrogate of the simulation model identified over a sample data set generated by the original simulation model itself. When applied to large 3D models, any DEMo exercise does require a pre-processing of the exogenous drivers and state variables in order to reduce, by spatial aggregation, the high number of candidate variables to appear in the final emulator. This work describes a hybrid clustering-variable selection approach to automatically discover compact and relevant representations of high-dimensional data sets. Time series clustering is adopted to identify spatial structures by objectively organizing data into homogenous groups, where the within-group-object similarity is minimized. In particular, the proposed approach relies on a hierarchical agglomerative clustering method, which starts by placing each time-series in its own cluster, and then merges clusters into larger clusters, until a compact, yet informative, representation of the original variables can be processed with the Recursive Variable Selection - Iterative Input Selection algorithm, in order to single out the most relevant clusters. The approach is demonstrated on a real-world case study concerning the reduction of Delft3D, a spatially distributed hydrodynamic model used to simulate salt intrusion dynamics in the tropical lake of Marina Reservoir, Singapore. Results show that the proposed approach permits a parsimonious, though accurate, characterization of salinity concentration.

Keywords: Dynamic Emulation Modelling; Time Series Clustering; Variable Selection; Data-Driven Models; Physically-Based Models

1 INTRODUCTION

Dynamic Emulation Modelling (DEMo) is emerging as an effective approach to overcome the computational requirements of large, physically-based models used in water resources operation problems. This approach is based on the reduction of a physically-based model by a simplified, computationally-efficient emulator constructed from and then used in place of the original model in highly resource-demanding tasks [Ratto et al., 2007]. When applied to large 3D models, any DEMo technique does require a pre-processing of the exogenous drivers and state variables to reduce the high number of

variables appearing in the final emulator. This operation can be either performed by adopting mathematical projection techniques (e.g. proper orthogonal decomposition or Karhunen-Loève transform, [Antoulas, 2005]) or spatial aggregations. The former have the advantage of being fully automatic, but this is often paied in terms of physical interpretability. The latter can preserve this property but they often require a number of expert-based skills and a-priori assumptions hardly formalizable in a procedural process.

With the purpose of preserving the physical interpretation of the aggregated state variables, while developing an automatic and system-independent tool, this work explores the potential of cluster techniques to discover compact and relevant representations of high-dimensional data sets produced via simulation of large 3D models. The rationale behind this choice is that clustering, by organizing data into homogeneous groups with minimized within-group-object similarity and maximized between-group-object dissimilarity [Liao, 2005], can be effective in providing a compact, yet informative, representation of the data produced with the 3D model, thus enhancing the final emulator accuracy and reducing the computational and analytical effort of the DEMo process.

Unlike static problems, the data here considered are a set of time-series with a spatial distribution. This implies the adoption of time-series clustering techniques that aims at determining groups of similar time-series. While most of the clustering techniques have been developed on static data, there is a trend of increased attention for time-series clustering, and various algorithms have been recently developed (see Liao [2005]; Kavita and Punithavalli [2010]; Fu [2011] for a survey). As discussed in the aforementioned surveys, these algorithms can differ from various aspects, but they all share the underlying idea of modifying the existent algorithms for clustering static data in such a way that time-series can be handled, for example by appropriately re-defining the similarity/distance measures with some metrics tailored to time varying signals. In the present work, we adopt a hierarchical agglomerative clustering algorithm [Magni et al., 2008], which starts by placing each time-series in its own cluster, and then merges clusters into larger clusters, until a certain termination condition (e.g. desired number of clusters) is reached. These clusters are then processed with a recursive variable selection algorithm [Castelletti et al., 2012a], in order to single out the most relevant clusters of some specific state variable that are relevant to the emulator's output. The approach is demonstrated on a real-world case study concerning the reduction of Delft3D, a spatially distributed hydrodynamic model used to simulate the effects of a salinity intrusion problem in Marina Reservoir, a freshwater reservoir located in the heart of Singapore.

2 METHODS AND TOOLS

2.1 DEMo procedure

Given a physically-based model $\mathcal{M}$ with state and exogenous driver vectors $\mathbf{X}_t$ and $\mathbf{W}_t$, the purpose of the DEMo exercise is to identify a computationally efficient, low-order model (the emulator) on a data-set appropriately generated via simulation of model $\mathcal{M}$. The emulator must be such that its output $\mathbf{y}_t$ accurately reproduces model $\mathcal{M}$'s output $\mathbf{Y}_t$, but has lower-dimension state and exogenous driver vectors $\mathbf{x}_t$ and $\mathbf{w}_t$, and takes the following general state-space form [Castelletti et al., 2012b]:

$$\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t, \mathbf{w}_t, \mathbf{u}_t) \tag{1a}$$

$$\mathbf{y}_t = \mathbf{h}_t(\mathbf{x}_t, \mathbf{w}_t, \mathbf{u}_t) \tag{1b}$$

where $\mathbf{u}_t$ is the vector of controlled inputs (i.e. management decisions), $\mathbf{f}_t(\cdot)$ is a nonlinear, time-variant, function that models the state transitions, and $\mathbf{h}_t(\cdot)$ is the output transformation function. According to the data-driven nature of DEMo, the emulator is

identified on a data-set $\mathcal{F}$ of tuples $\{\mathbf{X}_t, \mathbf{W}_t, \mathbf{u}_t, \mathbf{Y}_t, \mathbf{X}_{t+1}\}$, with $t = 1, \ldots, H$, generated via simulation of model $\mathcal{M}$. Assuming that a well-calibrated physically-based model $\mathcal{M}$ is available, the identification of a dynamic emulator can be performed in the following five steps [Castelletti et al., 2012b].

Step 1. *Design of computer experiments (DOE).* The DOE consists in a sampling in the space of model $\mathcal{M}$ inputs (i.e. the exogenous drivers $\mathbf{W}_t$ and controls $\mathbf{u}_t$), and it is aimed at defining a sequence of simulation runs to generate the data-set $\mathcal{F}$. Such data-set must be as much as possible informative, possibly reproducing all possible dynamic behaviours of the original model, forced by the widest spectrum of inputs. Considering the severe limitations on the number on runs typically imposed by large, physically-based models, proper techniques must be employed to effectively explore this space, such as expert-based design [Galelli et al., 2010] or pseudo-random binary sequences [MacWilliams and Sloane, 1976].

Step 2. *Variable aggregation.* The spatially-distributed nature of model $\mathcal{M}$ can lead to a large dimensionality of the state and exogenous driver vectors. By processing the data in $\mathcal{F}$ with a suitable aggregation scheme, $\mathbf{X}_t$ and $\mathbf{W}_t$ are transformed in two lower-dimension vectors $\tilde{\mathbf{X}}_t$ and $\tilde{\mathbf{W}}_t$, so that the majority of the variation in the original vectors is captured. The aggregation scheme can be based on expert-based skills or rely on fully automatic techniques, as the hierarchical time series clustering algorithm described in Section 2.2. Eventually, the data-set $\mathcal{F}$ is transformed into the lower-dimension data-set $\tilde{\mathcal{F}}$ of tuples $\{\tilde{\mathbf{X}}_t, \tilde{\mathbf{W}}_t, \mathbf{u}_t, \tilde{\mathbf{X}}_{t+1}, \mathbf{Y}_t\}$.

Step 3. *Variable selection.* Based on the information content of $\tilde{\mathcal{F}}$, model $\mathcal{M}$ is further simplified by selecting the components of $\tilde{\mathbf{X}}_t$ and $\tilde{\mathbf{W}}_t$ that will constitute the emulator's state $\mathbf{x}_t$ and exogenous driver $\mathbf{w}_t$ vectors. In the present study this operation relies on the Recursive Variable Selection - Iterative Input Selection (RVS-IIS) algorithm [Castelletti et al., 2012a]. RVS-IIS recursively selects the state, exogenous driver, and control variables that are most significant in characterizing the input-output behavior of the physically-based model until all the selected state variables are given a dynamic description.

Step 4. *Structure identification and parameter estimation.* In this step, the functions $\mathbf{f}_t(\cdot)$ and $\mathbf{h}_t(\cdot)$ are built. This is a 'traditional' identification problem, composed of the selection of a suitable model structure and the subsequent parameter estimation, performed in a data-driven fashion, based on the information content of $\tilde{\mathcal{F}}$.

Step 5. *Evaluation and physical interpretation.* Once the emulator has been calibrated, its ability in reproducing the model $\mathcal{M}$ input-output behaviour is cross-validated on the data-set $\tilde{\mathcal{F}}$. Eventually, this step aims at verifying the emulator credibility by the users' viewpoint. This property must be guaranteed by the emulator physical interpretability, which is usually based on the analysis of the arguments of eqs. (1a-b).

2.2 Hierarchical time series clustering

The hierarchical agglomerative cluster algorithm [Magni et al., 2008] is a popular method that works by organizing the data (the time-series) into a tree of clusters. At the first iteration the algorithm places each time-series in its own cluster and then starts to merge these small clusters, until a single cluster is obtained or an a-priori defined stopping condition is satisfied (e.g. the minimum number of clusters). In the present configuration, the distance between each cluster is measured as Euclidean distance, while clusters are merged according to the Ward's minimum variance method (at each clustering step, the merge that minimizes the increase in the sum-of-squares variance is chosen). The

Ward's method minimizes the total within-cluster variance, and thus goes in the desired direction of creating compact and informative clusters, which can then be processed with the RVS-IIS algorithm.

A part from its intrinsic simplicity, the hierarchical agglomerative cluster algorithm provides a number of advantages when embedded in the DEMo procedure: i) the algorithm works directly on raw time-series data, and does not require any conversion of the data into lower-dimension vectors, thus preserving the initial data integrity; ii) time-series are grouped into a tree of clusters that shows the relative distance between clusters; iii) unlike other algorithms (e.g. *k-means*, Hartigan and Wong [1979]), the number of clusters has not to be specified a-priori and this leaves a certain degree of freedom for the final choice. The main disadvantage is that the algorithm imposes a hierarchical structure on the data, so it cannot be used to perform any adjustment once a merge or split decision has been executed.

In principle, the choice of the number p of clusters should be a fair balance between dimension reduction and accuracy, with the extremes of these criteria represented by a single cluster and by each time-series in its own cluster. Among the different indexes and criteria available in literature, in this paper we use the Davies-Bouldin (DBI) and Dunn (DI) index [Davies and Bouldin, 1979; Dunn, 1973], which favor cluster configurations with small within-cluster variance and large between-clusters variance, thus resulting in compact and well separated clusters.

3 CASE STUDY

3.1 System description

Marina Reservoir (Singapore) was created in late 2008 with the construction of a barrage that closed the homonymous marina from the sea. Five main tributaries discharge water into the reservoir draining a catchment of approximately 100 km^2, about 1/7 of Singapore total surface area. The catchment mainly consists of urbanized land and it is characterized by the presence of three further reservoirs, only managed for drinking water supply and whose discharge to Marina Reservoir is rare and negligible. The reservoir is built for multiple purposes, including flood protection, lifestyle attraction and drinking water supply [Goedbloed et al., 2011]. However, its peculiar location makes water quality control a relevant issue: the inflow is characterized by short bursts of high flow with sediment and nutrient rich water followed by dry periods with almost no flow, and because of its location in the tropics temperature and light intensity are high. This typically leads to eutrophic in-reservoir water conditions; moreover, the relatively recent formation of the impoundment from a former estuary and the presence of salinity intrusion make salinity control another important objective. To account for quantity and quality objectives, the barrage is equipped with 9 surface gates, 7 pumps (to discharge water when the sea level is higher than the reservoir one) and 2 bottom pipes, which can release water at deeper levels, thus allowing for the mechanical control of the temperature profile and the salinity concentration.

3.2 Experiments setting

To calculate the flow conditions in Marina Reservoir, the 3D model Delft3D was adopted [Zijl and Twigt, 2007]. The model numerically solves the Navier-Stokes equations on a curvilinear orthogonal grid. In the adopted configuration the model has a maximum of 12 vertical layers, each with a specified height, so that the amount of layers depends on the local depth. This so-called *z-layer configuration* makes accurate calculation of the

stratification. The 3D flow model includes temperature and a specific heat flux model is applied to reproduce accurate temperature distribution in the reservoir. The model boundaries, i.e. several inflow points and the downstream barrage, are modelled by means of the 1D hydrological-hydrodynamic model SOBEK. Explicit coupling is used to connect the 1D and 3D models: the 3D model provides water levels to the 1D model as boundary conditions, while the 1D model provides the inflows from the catchment (and the release from the barrage) as boundary values for the 3D model. Finally, the salinity intrusion is modelled as a further boundary condition calibrated against a set of measured salinity concentrations at the barrage area.

The Delft3D exogenous driver $\mathbf{W}_t$ includes 7 components accounting for the main hydro-meteorological processes, while the control vector $\mathbf{u}_t$ has three components, i.e. the release from gates, pipes and pumps. The output $\mathbf{Y}_t$ is the salinity concentration [ppt] in the deepest point of the reservoir, located few hundred meters from the barrage. The model has 7 state variables for each computational cell, hence considering a total of 111 observation points and 12 computational layers, the state vector $\mathbf{X}_t$ has a total of nearly 10^4 variables. The real-to-run time ratio associated to this set-up is of about 1:100. With the purpose of generating the data-set of samples $\mathcal{F}$, a set of trajectories for the model inputs $\mathbf{W}_t$ and $\mathbf{u}_t$ is designed. As for the exogenous driver $\mathbf{W}_t$, the time-series of observational data over the period April 2009 - April 2010 is available, while, concerning $\mathbf{u}_t$, 10 different management scenarios are generated, thus giving a total of 10 simulation scenarios. Simulations are run with 30 sec simulation time-step, and an average vertical resolution of about 0.5 m. The data are sampled with an hourly time-step, and finally stored in a data-set $\mathcal{F}$ of $\sim 8 \cdot 10^4$ tuples.

4 RESULTS AND DISCUSSION

4.1 Clustering

To preserve a sort of physical interpretability of the aggregated time-series, the clustering algorithm is not directly applied to the complete set of state variables in $\mathcal{F}$, but to seven sub-sets, each containing the temporal and spatial realizations for temperature and salinity concentration T and s, temperature and salinity transport ΔT and Δs, and the horizontal and vertical velocities u, v and w. Because of the computational requirements of the algorithm, this clustering exercise is solved for each simulation run of the original model Delft3D. This gives a total of 70 clustering problems.

These problems are solved without specifying a-priori the desired minimum number of clusters (i.e. no stopping condition), which means that the algorithm is run until a full tree of clusters is obtained. This approach allows indeed to analyze a-posteriori the clustering results and to choose the number of clusters p (for each sub-set) that best satisfies the DBI and DI indexes. Figure 1 reports the average value (over the 10 simulation runs) of the DBI and DI indexes as a function of p for the temperature transport ΔT. Results show that the best number p^* of clusters that minimizes the DBI and maximizes the DI index is 3, while the introduction of a larger number of clusters does not improve the clustering results any further. This analysis is performed for a maximum value of p equal to 11 corresponding to the number of layers (a part from the surface layer) in the original model. This value provides an empirical upper bound to p, since the data could be for example aggregated by considering the spatial average of each sub-set in the original model vertical layers.

The final results obtained for the remaining sub-sets are reported in Table 1, where the results obtained for the water level h are also shown. The time series algorithm found

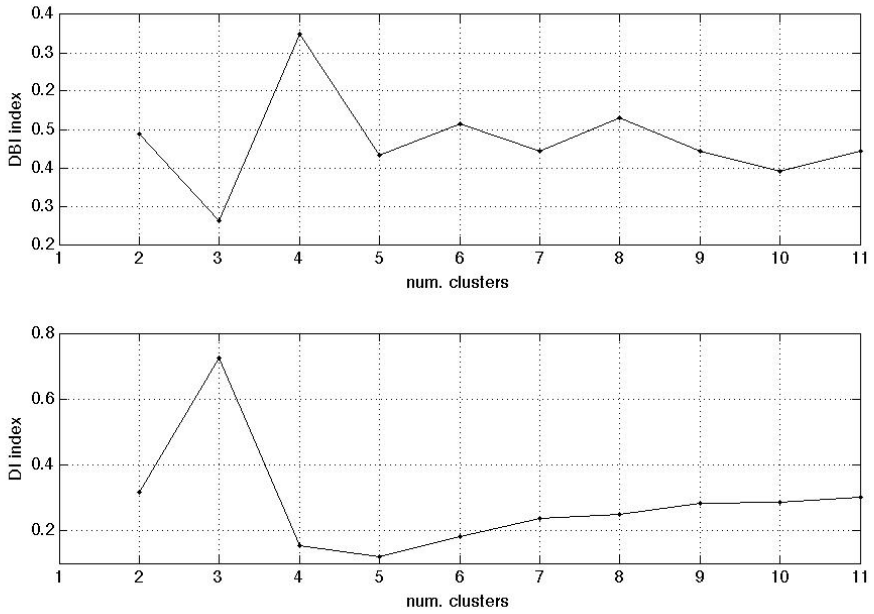

Figure 1: Average value (over the 10 simulation runs) of the DBI and DI indexes for the temperature transport ΔT.

Table 1: Selected number of clusters for the different sub-sets composing the vector $\mathbf{X}_t$. The symbols are explained in the text.

	s	T	Δs	ΔT	u	v	w	h
# clusters	6	6	4	3	4	6	8	10

a total of 8 different clusterings, each one identifying a set of homogeneous areas of a particular variable. These areas can vary from sub-set to sub-set, so, for example, the salinity concentration s can be grouped into 6 different areas, while the vertical velocity w into 8.

4.2 Emulator identification

The state variables so-aggregated are then processed with the RVS-IIS algorithm, together with the exogenous driver and control vector $\mathbf{W}_t$ and $\mathbf{u}_t$, to single out the most relevant variables in explaining the preselected emulator output. The performance of the emulator being built are evaluated in k-fold cross-validation (with $k = 10$) using the coefficient of determination as performance index. The selection of the most relevant variables to appear in the emulator takes two calls of the tree-based RVS-IIS algorithm to single out a state vector $\mathbf{x}_t$ with one component (i.e. the salinity concentration in the fourth cluster), an exogenous driver vector $\mathbf{w}_t$ with one component (i.e. the groundwater flow) and a single-component control vector $\mathbf{u}_t$, namely the flow released from the bottom pipes. The selected variables can be given a physical interpretation: i) the state variable provides information about the salinity concentration in a homogeneous area very close to the barrage; ii) the groundwater flow represents the main source of the salinity intrusion, while

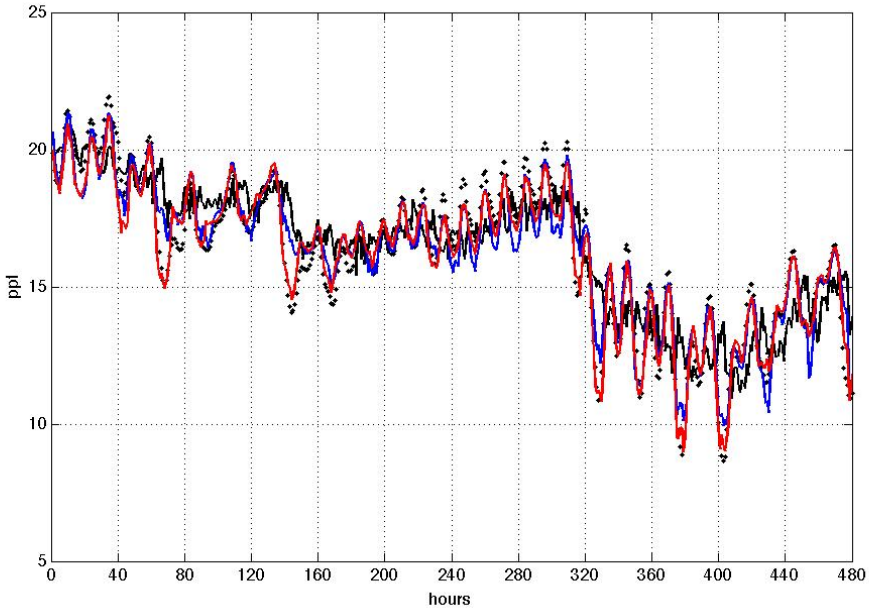

Figure 2: Comparison between salinity concentration (output variable) simulated with Delft3D (black dots) and the output reproduced by the emulator during the different stages of the variable selection process, namely with one, two and three selected inputs (black, blue and red line respectively).

iii) the pipes, located at the bottom of the barrage, have an obvious key role in releasing water with higher salinity concentrations. The selected model structure is an ensemble of Extremely Randomized Trees (see Castelletti et al. [2012a] and references therein), whose final performance in cross-validation correspond to a coefficient of determination equal to 0.95. A specimen of the trajectories obtained during the k-fold cross-validation process is shown in Figure 2.

5 Conclusions and further research

This work describes a hybrid clustering-variable selection approach to automatically discover compact and relevant representations of high-dimension data sets generated by computationally-intensive physically-based models. The approach, which relies on a time-series hierarchical agglomerative clustering method, is demonstrated on the emulation of a large, 3D model (Delft3D) used to simulate the salt intrusion dynamics in Marina Reservoir (Singapore). Since the clustering algorithm is embedded in a computationally expensive emulation modelling procedure, we here considered an unsupervised clustering approach, while a supervised approach, which aims at determining the clustering solution that maximizes the emulator performance, is part of the on-going research activities. The main advantage of this second approach would rely in a further increase of the emulator accuracy and reliability. Finally, future research will also focus on the comparison of different unsupervised clustering methods.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support and contributions of the Singapore-Delft Water Alliance (SDWA). The research presented in this work was carried out as part of the SDWAs Multi-objective Multiple Reservoir Management research programme (R-303-001-005-272).

REFERENCES

Antoulas, A. An overview of approximation methods for large-scale dynamical systems. *Annual Reviews in Control*, 29(2):181–190, 2005.

Castelletti, A., S. Galelli, M. Ratto, R. Soncini-Sessa, and P. Young. A general framework for dynamic emulation modelling in environmental problems. *Environmental Modelling & Software*, 34:5–18, 2012b. doi: 10.1016/j.envsoft.2012.01.002.

Castelletti, A., S. Galelli, M. Restelli, and R. Soncini-Sessa. Data-driven dynamic emulation modelling for the optimal management of environmental systems. *Environmental Modelling & Software*, 34:30–43, 2012a. doi: 10.1016/j.envsoft.2011.09.003.

Davies, D. and D. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

Dunn, J. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

Fu, T. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.

Galelli, S., C. Gandolfi, R. Soncini-Sessa, and D. Agostani. Building a metamodel of an irrigation district distributed-parameter model. *Agricultural Water Management*, 97(2): 187–200, 2010.

Goedbloed, A., S. Galelli, and D. Schwanenberg. Assessing the effectiveness of a real-time control method for marina reservoir management. In *19th International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand*, pages 4036–4042. (eds) MODSIM2011, 2011.

Hartigan, J. and M. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

Kavita, V. and M. Punithavalli. Clustering time series data stream - a literature survey. *International Journal of Computer Science and Information Security*, 8(1):289–294, 2010.

Liao, T. Clustering of time series data - a survey. *Pattern Recognition*, 38(11):1857–1874, 2005.

MacWilliams, F. and N. Sloane. Pseudo-random sequences and arrays. *Proceedings of the IEEE*, 64(12):1715–1729, 1976.

Magni, P., F. Ferrazzi, L. Sacchi, and R. Bellazzi. Timeclust: a clustering tool for gene expression time series. *Bioinformatics Application Note*, 24(3):430–432, 2008.

Ratto, M., A. Pagano, and P. Young. State dependent parameter metamodelling and sensitivity analysis. *Computer Physics Communications*, 177(11):863–876, 2007.

Zijl, F. and D. Twigt. Singapore Marina Reservoir Study. Hydrodynamic modelling. Research Report Z.4265.10/20/30, Deltares, 2007.