Author:
Lane, Rosie A

Title:
National-scale hydrological modelling of high flows across Great Britain

multi-model structures, regionalisation approaches and climate change analysis with uncertainty

Author:
Lane, Rosie A

Title:
National-scale hydrological modelling of high flows across Great Britain

Multi-model structures, regionalisation approaches and climate change analysis with uncertainty

National-scale hydrological modelling of high flows across Great Britain

Multi-model structures, regionalisation approaches and climate change analysis with uncertainty

By

ROSANNA ALICE LANE

School of Geographical Sciences
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Science.

SEPTEMBER 2020

Word count: 47,172

Abstract

National scale hydrological modelling frameworks are required to underpin effective water management in the face of large-scale pressures such as climate change. Many challenges remain for the application of national-scale models including: 1) developing and selecting appropriate model structure(s), 2) estimating model parameters for gauged and ungauged catchments, especially for models which require spatially distributed parameter fields, and 3) incorporating and communicating model uncertainties. This thesis addresses these challenges through a focus on modelling median and higher flows for large samples (hundreds) of catchments across Great Britain (GB) within a nationally consistent framework that includes predictive uncertainties.

The first research chapter evaluates the predictive capability of multiple lumped, conceptual models for over 1000 catchments within an uncertainty framework, providing a performance benchmark. Regions where models often failed were identified (mountainous catchments in northeast Scotland, catchments overlaying aquifers in southeast England), and model performance was related to catchment characteristics to better understand where/why models fail. Significantly, it was found that despite substantial human modifications to catchments across GB, poor model performance was often linked to more general hydrological processes, such as low annual total rainfall, high baseflow contributions, and the water balance not closing. The second research chapter develops a parameterisation scheme to estimate nationally consistent parameter fields for a distributed hydrological model by relating model parameters to spatial geophysical data. This is applied within a novel framework for the inclusion of uncertainties when constraining spatial parameter fields. The resultant parameter fields performed well (non-parametric KGE > 0.75) across the majority (60%) of catchments, enabling nationally consistent simulations across gauged and ungauged catchments, and reflecting hydrologically meaningful variation in catchment characteristics. The third research chapter applied this nationally parameterised model, to provide the first evaluation of climate change impact on river flows across GB to include both climate and hydrological model parameter uncertainties. This indicated an increase in the magnitude and frequency of high flows for catchments along the west coast of GB and across Scotland, albeit with large uncertainties especially across the southeast.

Overall, this thesis improves understanding of model performance variation across Great Britain and where targeted model improvements are needed, contributes a national modelling framework which enables spatially consistent predictions across gauged and ungauged areas, and demonstrates how uncertainties can be included in larger-scale and large-sample hydrological studies. Whilst this thesis is focused on modelling across GB, the methods and conclusions can be transferred elsewhere to improve large-scale and large-sample model applications.

Dedication and Acknowledgements

There are many people who I would like to thank for making this thesis possible:

First and foremost I would like to thank my supervisors Jim, Thorsten and Gemma. I could not have asked for better mentors, and I would like to thank you all for your feedback and comments on the many paper and thesis drafts that I have sent round. Jim, thank you for sparking my interest in hydrology, and encouraging me to persue a PhD in the first place. I always came out of our meetings full of ideas and inspiration. Thorsten, thank you for your continual support, paper contributions and for reminding me to step back and look at the bigger picture. Gemma, thank you for your encouragement, great advice and endless patience teaching me how to use FUSE and DECIPHeR. I am so happy that you became one of my official supervisors, as this thesis would not have been possible without your help!

I greatly appreciate Jan Seibert for hosting me at the University of Zurich for three months, and for the H2K group for making me feel so welcome. I also thank all my colleagues in the Browns PhD office for making it such a fun place to work, and for supplying the home-baked cakes that kept me going. Special thanks go to Tamsin and Alex for organising so much fun.

Thank you to my family for supporting me every step of the way. And final and biggest thanks go to Ben. Thank you for your endless support, I could not have done this without you.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: . [redacted] ..... DATE: 04/01/2021

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Background

Robust and reliable simulations of river flows are needed for water supply management, flood and seasonal runoff forecasting, and planning for land-use and climate change impacts on both daily flows and hydrological extremes (such as floods and droughts). Predictions of river flows across large domains (i.e. national to continental scales) are increasingly needed to assess how rivers will respond to these large-scale pressures and to inform national/international policies and decision-making related to water management (Archfield et al., 2015; Wagener et al., 2010) such as the EU Water Framework Directive (European Parliament, 2000; Lindenschmidt et al., 2007), the National Flood Risk Assessment for England (Environment Agency, 2009), and the UK National Adaptation Programme (DEFRA, 2018; Wade et al., 2013). For these large-scale challenges, flow predictions at the national scale can be extremely useful for decision makers (Watts et al., 2015). They provide a broad overview of future changes, enable identification of regions which may be most impacted and support robust decision making on future multimillion pound investments in water infrastructure such as reservoirs or flood defences (Environment Agency, 2009; National Infrastructure Commission, 2018; Watts et al., 2015). There is therefore a need to develop hydrological models which can be applied across national to continental scales, yet still provide locally relevant simulations, to underpin environmental management and policy decision making (Archfield et al., 2015; Beven and Cloke, 2012; Bierkens et al., 2015; McMillan et al., 2016).

However, hydrological modelling is an uncertain science (Beven, 1993, 2009; Montanari et al., 2009; Pechlivanidis et al., 2011). Models are limited by our knowledge of hydrological systems (Beven and Alcock, 2012), the quality/quantity of observational data available to drive and setup

a model (Baroni et al., 2017; Coxon et al., 2015; McMillan et al., 2010; Mcmillan et al., 2012; Westerberg et al., 2016), and the decisions made in setting up and implementing a model (e.g. selection of model structure, parameterisations, and metrics for model evaluation) (Beven and Cloke, 2012; Coxon et al., 2014; Krueger et al., 2010). Consequently, quantifying and communicating the uncertainties surrounding river flow predictions is vital if results are used for water management/ policy decisions, to prevent over-confidence in simulation results (McMillan et al., 2017).

A key goal is therefore to create hydrological modelling frameworks which can produce robust hydrological simulations across a diverse range of catchment characteristics and under changing catchment conditions, whilst reflecting the underlying uncertainties. In recent years, there has been substantial progress in the field of large-scale modelling, driven by increasing computational capabilities and availability of large hydrological datasets (Addor et al., 2020; Archfield et al., 2015; Bierkens et al., 2015). A variety of models have been applied across large-domains/ large samples of catchments, including both large-scale gridded models (e.g. mHM - Samaniego et al. 2010, 2018 and G2G - Bell et al. 2007, 2009) able to produce spatially contiguous simulations across a large domain, semi-distributed approaches (e.g. TopNet - McMillan et al. 2016 and CLASSIC - Crooks and Naden 2007) and large-sample approaches which apply models across large numbers of catchments (e.g. Nicolle et al. 2014; Perrin et al. 2001; Van Esse et al. 2013). Despite this diversity in hydrological models for large-scale and large-sample applications, there are still many ongoing challenges.

Firstly, a key challenge is how to model the heterogeneity of hydrological processes across large regions and ensure that the dominant processes are included in national models (McDonnell et al., 2007; Troch et al., 2009). Hydrological processes will differ across a region, for example due to differences in land-use, slope, precipitation, human impacts, geology, and/or soil characteristics (Gao et al., 2018; Teutschbein et al., 2018). Whether variations in hydrological processes over the landscape can be included in the model will depend upon its spatial resolution and how the landscape has been discretized. We need to move towards flexible hydrological modelling frameworks, which are flexible in terms of how the landscape is discretized and the model structure/ parameter sets applied across the model domain, so that models can easily be adapted to best reflect different environments and modelling objectives (Beven and Freer, 2001b; Clark et al., 2011b; Coxon et al., 2019; Mendoza et al., 2016).

Secondly, model parameterisation across large domains is an ongoing challenge, especially for models with distributed parameters (Archfield et al., 2015; Bierkens et al., 2015; Clark et al., 2016b). Parameterisation schemes for large domain modelling must be efficient, able to realistically reflect differences in parameter values across the landscape (whether this is between catchments, between gauged sub-catchments or for fully distributed parameter fields) and

consistent across the landscape. Parameters are also often needed for ungauged areas (Blöschl et al., 2013), estimated in a way that is consistent with gauged basins. To achieve this, many large-domain models use spatially distributed geophysical information (e.g. soils or land-use data) to help define model parameter fields (Mizukami et al., 2017). How to best extract parameter information from this geophysical data is an area of ongoing research, with techniques such as multiscale parameter regionalisation (MPR) emerging as effective strategies to produce seamless parameter fields across large domains (Mizukami et al., 2017; Samaniego et al., 2010, 2017).

Thirdly, including model uncertainties in larger-scale and large-sample modelling studies which are already computationally expensive is difficult. The importance of estimating and communicating modelling uncertainties is increasingly recognised, with an extensive literature on uncertainty analysis techniques (Beven, 1993; Freer et al., 1996; Montanari et al., 2009; Vrugt et al., 2008; Yang et al., 2008), and numerous studies including uncertainty analysis for catchment-scale studies (e.g. Bosshard et al. 2013; Kay et al. 2009; Smith et al. 2014b; Velázquez et al. 2013) . However, uncertainty analysis is still lacking in many large-scale model applications, for example few national climate-impact studies for the UK include hydrological modelling uncertainties. Methods to characterise uncertainties often involve running large ensembles of model simulations (e.g. the GLUE technique requires a large enough sample of Monte Carlo simulations to effectively sample the parameter space (Beven and Freer, 2001a; Freer et al., 1996)), which can be computationally demanding and so difficult to apply across large scales. A key goal is therefore to develop techniques to facilitate the inclusion of model uncertainties in large-scale studies, alongside models which are sufficiently computationally efficient to enable uncertainty characterisation (Archfield et al., 2015).

Finally, the ongoing evaluation and communication of model performance is important to ensure that we are developing hydrological models which are robust and reliable. If hydrological model outputs are used to guide policy decisions, then it is vital that we understand the model skill in simulating flows, ensure models are suited for their intended purpose, and are aware of model weaknesses/limitations (Andréassian et al., 2009; Klemes, 1986; McMillan et al., 2016). This is especially important for land-use/climate change impact assessments, where we put additional demands on the model by extrapolating to conditions not seen in the observed record, and so we need to be sure that the model is correctly representing processes (Coron et al., 2012; Klemes, 1986). Therefore, it is important to comprehensively evaluate models across large-samples of hydrologically varied catchments to understand how well different model structures/parameterisations can simulate flows across various hydroclimatic settings (Archfield et al., 2015; Gupta et al., 2014).

As the demand for national-scale hydrological predictions increases, it is vital that we develop

new tools and modelling systems to explore these large sample hydrology problems. This thesis aims to contribute to this task through three research chapters, which all focus on modelling median and higher flows across large samples (hundreds) of catchments in Great Britain with quantification of hydrological model uncertainties.

The first research chapter evaluates the performance of lumped, conceptual hydrological model structures across Great Britain, using a large-sample hydrology approach to derive generalisable conclusions regarding factors influencing model performance, reasons for model failure and the relative merits of different model structures. This also provides a model performance benchmark across Great Britain, which improved models could be compared against to ensure continuing progress in hydrological model development. The second chapter moves from lumped to distributed modelling, as distributed models are required to address spatial hydrological problems. It improves a flexible hydrological modelling framework to enable nationally consistent river flow simulations for gauged and ungauged areas, with quantification of modelling uncertainties. It does this by developing a parameterisation scheme which relates model parameters to relevant national geophysical data-sets (including land-use, soil characteristics and geology). This creates spatial fields of model parameters, capable of reflecting spatial differences across the landscape. These are subsequently conditioned on streamflow data from over 400 catchments resulting in a robust set of nationally-consistent behavioural parameter fields which reflect underlying parameter uncertainties. The third chapter then applies this framework to explore changing high flows in the future – providing the first national climate change simulations for Great Britain to include climate and hydrological modelling uncertainties. Whilst these tools and techniques are applied to catchments across Great Britain, the framework is general and could be applied elsewhere and for other applications, such as land-use change. Ultimately, this leads us to better national and local scale hydrological predictions which reflect important spatial differences and underlying uncertainties.

1.2 Thesis aims

This thesis first explores issues relating to the development of national-scale hydrological models which characterise model uncertainties: evaluating the performance of multiple, lumped, conceptual model structures for simulating high flows across Great Britain, and implementing a parameterisation scheme to create ensembles of spatially consistent model parameter fields from spatial datasets. The final chapter then builds upon this research, applying the nationally parameterised hydrological model to explore climate change impact on median and higher flows using the most recent UK Climate Projections. This work 1) implements the first multi-model uncertainty framework across GB, 2) successfully develops a national parameterisation scheme, presenting a framework for the inclusion of uncertainties in spatial parameter fields that cover

gauged and ungauged areas, and 3) presents the first large sample uncertainty evaluation of future flows across GB.

This thesis aims to answer the following research questions, which form the basis of the three results chapters:

1. How well are simple, conceptual model structures able to simulate high flows across Great Britain?
    a) How well do simple, lumped hydrological model structures perform when assessed over annual and seasonal timescales via standard performance metrics?
    b) Are there advantages in using an ensemble of model structures over any single model, and if so, are there any emergent patterns or characteristics in which a given structure and/or behavioural parameter set outperforms others?
    c) What is the predictive capability of behavioural models for then predicting annual maximum flows when applied in a parameter uncertainty framework?
2. Can observed datasets be used to parameterise a spatially distributed model across Great Britain, including parameter uncertainties?
    a) How can the multiscale parameter regionalisation methodology be adapted to produce uncertain national parameter fields for the DECIPHeR modelling framework?
    b) What is the difference in performance between a model driven by regional parameter fields and catchment constrained parameter fields?
    c) Do the constrained parameter fields reflect hydrological features within catchments, and do they therefore increase the realism of simulations?
3. What is the impact of climate change on river high flows across Great Britain?
    a) Where are climate change impacts expected to be the most extreme?
    b) What are the uncertainties surrounding these projections, due to climate model and hydrological model parameterisation?
    c) What is the relationship between climatic changes (i.e. change in precipitation and potential evapotranspiration) and the change in high flows, and how does this vary regionally?

1.3 Thesis structure

Chapter 2: Literature Review. This chapter reviews relevant literature and sets out the context and motivation for the following research. It introduces the fields of large-scale and large-sample hydrology, and then discusses national-scale hydrological modelling with a focus on model structures, model parameters, uncertainties and climate impact studies.

Chapter 3: Overarching methods. Individual methodologies are given within each research chapter. This chapter highlights methodological similarities/differences between the research chapters and expands on the shorter research chapter methodologies where necessary, without repeating information. Datasets and methods that underpin multiple research chapters are also discussed.

Chapter 4: Performance of conceptual model structures across Great Britain. This chapter evaluates the performance of four lumped, conceptual model structures across over 1000 catchments in Great Britain, within an uncertainty framework. Catchment characteristics are used to explain why models perform well/poorly for certain catchments, and to gain insight into which model structural decisions are important across Great Britain. Supporting information for this chapter is given in Appendix B.

Chapter 5: Parameterisation of a spatial model across Great Britain. This chapter develops and evaluates a parameterisation scheme to create national parameter fields from spatial catchment data. The multiscale parameter regionalisation technique is tailored to the DECIPHeR hydrological model, and applied within a parameter uncertainty framework, to create national parameter fields. Model performance is then compared to simpler parameterisation schemes, to evaluate the suitability of the method. Supporting information for this chapter is given in Appendices C-E.

Chapter 6: Climate impacts on high flows. This chapter applies the model developed in chapter 5 to explore climate change impact on high flows across Great Britain, using the latest UK Climate Projections product. Supporting information is given in Appendices F and G.

Chapter 7: Conclusions and outlook. This chapter provides a summary and synthesis of the main findings from the three research chapters and provides recommendations for future research.

Appendix A: Author's CV.

1.4 Publications and conference presentations

Work carried out for this thesis has been presented in journals and at academic conferences. This includes contributions to two co-authored papers which are not part of the main body of this thesis. A list of research outputs are given below:

1.4.1 Publications prepared for this thesis

- Lane, R., Coxon, G., Freer, J., Wagener, T., Johnes, P., Bloomfield, J., Greene, S., Macleod, C., and Reaney, S. (2019). Benchmarking the predictive capability of hydrological models for river flow and flood peak predictions across over 1000 catchments in Great Britain. *Hydrology and Earth System Sciences*. 23, 4011-4032. https://doi.org/10.5194/hess-23-4011-2019.
  The data is available at https://doi.org/10.5523/bris.3ma509dlakcf720aw8x82aq4tm.
  This forms research chapter 1.

- Lane, R., Freer, J., Coxon, G., and Wagener, T. (2020) Incorporating uncertainty into multiscale parameter regionalisation to parameterise a national-scale hydrological model. [Submitted to Water Resources Research]
  This forms research chapter 2.

- Lane, R., Freer, J., Seibert, J., Coxon, G., and Wagener, T. (2020) Evaluating climate change impacts on flooding nationally across Great Britain, including climate and hydrological modelling uncertainties. [In preparation]
  This forms research chapter 3.

1.4.2 Co-authored publications

- Coxon, G., Freer, J., Lane, R., Dunne, T., Knoben, W., Howden, N., Quinn, N., Wagener, T., and Woods, R. (2019). DECIPHeR v1: Dynamic fluxEs and ConnectIvity for Predictions of HydRology. *Geoscientific Model Development*. 12, 6. https://doi.org/10.5194/gmd-12-2285-2019.
  The model code is available on github at https://github.com/uob-hydrology/DECIPHeR.
  This paper introduced and evaluated the DECIPHeR hydrological modelling framework. My contribution included 1) developing the model parameter schema and writing the model code to read in spatially variable parameter fields, 2) testing the model code and playing a key role in the model development, 3) helping to write the user manual and 4) providing comments and text for the paper. This work was completed as part of the model development for research chapter 2.

- Coxon, G., Addor, N., Bloomfield, P., Freer, J., Fry, M., Hannaford, J., Howden, N., Lane, R., Robinson, E., Wagener, T., and Woods, R. (2020) CAMELS-GB: Hydrometeorological time

series and landscape attributes for 671 catchments in Great Britain. *Earth Syst. Sci. Data*, https://doi.org/10.5194/essd-2020-49.
The dataset is available on the Environmental Information Data Centre at https://catalogue.ceh.ac.uk/documents/8344e4f3-d2ea-44f5-8afa-86d2987543a9.
This paper presented a large dataset of catchment attributes, boundaries and hydrometeorological timeseries across Great Britain, to aid large-sample studies. My contribution was to 1) calculate catchment summary soil properties and 2) provide draft text on this data for the paper. This was closely related to the use and processing of soil datasets required for research chapter 2.

1.4.3 Selected first-author conference presentations

- Estimating uncertain spatial parameter fields for the DECIPHeR hydrological model across Great Britain. Oral presentation, European Geosciences Union (EGU) 2019.
- Parameterisation of the Dynamic TOPMODEL national-scale hydrological model using uncertain Multiscale Parameter Regionalisation. Poster presentation, EGU 2018.
- Benchmarking hydrological model predictive capability for UK River flows and flood peaks. Poster presentation, EGU 2017.

CHAPTER 2

LITERATURE REVIEW

Hydrological models are useful tools which can support decision making for hydrological problems (Beven, 2012). Hydrological models, or more specifically rainfall-runoff models, are used to estimate runoff from precipitation and other climate variables. This enables flow predictions where measurements are otherwise unavailable, for example predictions of river flows in ungauged basins, projections of future river flows, and improved understanding of how environmental changes such as land-use or climate change may modify river flows (Beven, 2012). They have been widely applied to explore how catchments may respond to future challenges, including the impacts of land-use change on river flows (e.g. Choi and Deal 2008; Hundecha and Bárdossy 2004; Mango et al. 2011) and climate change impact on low flows and droughts (e.g. Rudd et al. 2019; Wilby and Harris 2006, floods (e.g. Booij 2005; Kay et al. 2014a; Prudhomme et al. 2013b) and mean and seasonal river flows (e.g. Donnelly et al. 2016; Prudhomme et al. 2012; Steele-Dunne et al. 2008).

There is an increasing need for consistent applications of hydrological models across large areas and/or large catchment samples, to support decision making and water-related policy at regional to continental scales. Modelling studies which cover large areas are able to provide a broad overview, which is informative for national and continental scale policy and decision making (DEFRA, 2018; Watts et al., 2015). They are therefore needed to support national/continental scale water management, in fields such as water resources planning, flood protection and ensuring good ecological status of rivers (Lindenschmidt et al., 2007). For example, in the case of climate change impacts on river flows in the UK, there are a large number of studies focused on single catchments or regions (e.g. Cameron 2006; Cameron et al. 2000; Fowler and Kilsby 2007; Jackson et al. 2011; Prudhomme et al. 2003; Reynard et al. 2001) but inconsistencies in methods and uneven coverage makes it hard to compare different locations and identify appropriate

adaption responses on a national scale (Prudhomme et al., 2003). It is therefore important to develop methods of modelling across larger areas and across geoclimatic gradients, to provide the national-continental overviews that are most useful to decision-makers (Andersson et al., 2015; Archfield et al., 2015; Lindenschmidt et al., 2007; Watts et al., 2015). At the same time, it is important that these large-scale studies are not over-simplified and provide relevant local-scale predictions, so the results are still informative for policy decisions.

The fields of large-scale hydrology (applying models across large domains) and large-sample hydrology (applying models across large numbers of catchments) have seen much progress in recent years, but many challenges still remain (Addor et al., 2020; Archfield et al., 2015; Bierkens et al., 2015; Gupta et al., 2014; Wood et al., 2011). These challenges include:

1. Developing model structures that can capture the wide heterogeneity of hydrological processes across large scales.
2. Balancing model complexity and computational efficiency, to enable ensemble simulations and characterisation of uncertainties across large scales.
3. Ensuring nationally consistent parameterisation approaches, which do not show inconsistencies between catchments and adequately reflect underlying processes.
4. Comprehensively evaluating models across large samples of hydrologically diverse catchments, to ensure that models are fit for certain purposes and help drive model improvements.
5. Quantifying and communicating modelling uncertainties in large-scale studies.

To achieve these goals, we need advances in defining model structures, constraining model parameters, and representing modelling uncertainties. This literature review assesses these different aspects in detail, with sections on the following: 1) An introduction to the fields of large-scale and large-sample hydrology; 2) the different model structures used for national-scale modelling, highlighting the need for multi-model comparisons, flexible model frameworks, and lumped to distributed models; 3) the approaches used to define model parameters across large domains, with a focus on methods to constrain lumped and spatially distributed parameters over gauged and ungauged areas; 4) the importance of recognising modelling uncertainties, and uncertainty analysis techniques used to do this; 5) the use of hydrological models for impact studies, further highlighting the need for flexible national models which are both able to represent spatial differences across landscapes and characterise uncertainties; 6) a summary.

2.1 Large-scale and large-sample hydrology

In the context of national scale simulations, this thesis relates to both the fields of large-sample hydrology and large-scale hydrology which are explained in more detail below.

2.1.1 Large-sample hydrology

Large-sample modelling studies apply hydrological models across large numbers of hydrologically varied catchments to derive robust and generalisable conclusions regarding hydrological processes and models (Gupta et al., 2014). The ethos behind large-sample hydrology is to go beyond traditional studies focusing on a single or small sample of catchments in depth and instead broaden our focus across large samples of catchments to establish general hydrologic concepts applicable across regions (Gupta et al., 2014; Newman et al., 2015). This has roots in comparative hydrology, which aims to understand and learn from the similarities and differences in hydrological processes between places (Falkenmark and Chapman, 1989; Kovács, 1984).

Large-sample hydrology studies began emerging in the 1980s/90s, with early studies in France, Belgium, and Australia (Makhlouf and Michel, 1994; Nathan and McMahon, 1990; Vandewiele et al., 1991), later joined by an increasing number of studies in the UK, Austria, and the USA (Fernandez et al., 2000; Kay et al., 2006; McIntyre et al., 2005; Merz and Blöschl, 2004; Parajka et al., 2005), see full reviews in Blöschl et al. (2013) and Gupta et al. (2014). In recent decades, large sample hydrology studies have become more common (Kumar et al., 2013b; Newman et al., 2017; Oudin et al., 2008; Perrin et al., 2001; Poncelet et al., 2017; Van Esse et al., 2013; Velázquez et al., 2010), facilitated by the increasing availability of computational resources, gridded meteorological data (e.g. Beck et al. 2017; Contractor et al. 2020; Perry and Hollis 2005; Tanguy et al. 2014) and streamflow records (Dixon et al., 2013; Stahl et al., 2010). This has been aided by a community effort to generate open-source, large-sample hydrology datasets, providing exciting new opportunities for large-sample studies (Addor et al., 2017, 2020; Alvarez-Garreton et al., 2018; Coxon et al., 2020; Newman et al., 2015).

There are many benefits to the large-sample hydrology approach. As summarised by Gupta et al. (2014), use of a large catchment sample can 1) ensure conclusions of a modelling study are generalisable and not catchment-specific, 2) aid the evaluation of methods/models by providing a range of applicability and ensuring they are appropriate across different environments, and 3) provide enough information to enable statistically significant relationships to be established. However, it also presents challenges; with the broad focus losing detail, local catchment knowledge and recognition of uncertainties compared to smaller-scale catchment investigations. Therefore, the large-sample hydrology approach is useful for a variety of hydrological applications, in conjunction with more detailed site-studies. The concept of large-sample hydrology is fundamental for this thesis, and examples of large-sample hydrology studies are given throughout this literature review.

2.1.2 Large-scale hydrology

The field of large-scale hydrology focuses on applying models over large spatial domains, which Cloke and Hannah (2011) define as scales greater than a single river basin up to the entire planet. This is an established field (Bierkens et al., 2015; Cloke and Hannah, 2011; Wood et al., 2011), which has many crossovers with the large sample hydrology approach. They both aim to understand the key patterns and drivers of hydrological processes across regions, to improve our ability to model these processes and face many of the same challenges.

The key difference between large-scale and large-sample hydrology is that the large-scale approach produces model simulations which are spatially contiguous across large domains, whilst the large-sample approach focuses on large numbers of catchments which are not necessarily spatially connected (Addor et al., 2020). Whilst large-scale hydrology includes different types of models, such as land-surface schemes and global hydrological models which focus on other fluxes such as soil moisture or the monthly/seasonal water balance rather than streamflow, here we focus on catchment hydrology models run at large scales (e.g. Pappenberger et al. 2011; Rakovec et al. 2019).

The fields of large-scale and large-sample hydrology are therefore complementary. Studies have used streamflow outputs from large-scale models for large sample investigations (e.g. Rakovec et al. 2016a), and knowledge gained from large-sample studies can help us improve our large-scale models. Whilst this thesis focuses on large-sample model applications, it also draws from the field of large-scale hydrology in setting up a modelling framework which can run in a consistent way across large domains, albeit from a catchment modelling perspective.

2.2 Model structures and evaluation

The model structure represents our perception of how a catchment is organised, the important elements and stores, and how these are interconnected. A plethora of hydrological model structures exist, with models developed to suit different modelling objectives and to represent different perceptions of the governing hydrological processes across a catchment/area (Clark et al., 2011a). Each model structure has its own strengths and weaknesses, for example the classic TOPMODEL structure is related to catchment topography to define hydrological similarity and therefore whilst it may be suited to steep humid catchments where topography broadly defines runoff processes, it will not be able suitable to produce good simulations everywhere (Beven and Kirkby, 1979; Perrin et al., 2001). When selecting a hydrological model structure, it is therefore important to choose a structure that is appropriate for the study location and aims (Klemes, 1986).

Existing national-scale hydrological models vary widely, with grid-based, semi-distributed

and lumped catchment models used for different countries and different model applications. Distributed grid-based models have been developed for countries including Denmark, the USA, and the UK. In Denmark, the importance of groundwater contributions to river flow led to the development of an integrated groundwater/surface water hydrological model (Henriksen et al., 2003; Højberg et al., 2013), which has been implemented nationally at 1 km resolution and used for regional climate impact assessments (van Roosmalen et al., 2007). In the US, the VIC macroscale hydrological model which solves the full water and energy balances has been calibrated and implemented nationally for hydro-climate impact assessment (Liang et al., 1994; Oubeidillah et al., 2014). For the UK, the Grid-to-Grid (G2G) hydrological model runs on a 1 km national grid and has been widely used for climate impact assessment at the UK Centre for Ecology and Hydrology (Bell et al., 2007, 2009; Kay et al., 2018; Rudd et al., 2017).

Semi-distributed hydrological models, which split the landscape by sub-catchments and other key features, have also been applied at the national scale. For example, the Topnet model has been applied across New Zealand splitting the landscape by sub-catchment (McMillan et al., 2016); the S-HYPE model has been applied across Sweden dividing the landscape by soil and land-use in addition to sub-catchments (Bergstrand et al., 2014); and the DECIPHeR hydrological modelling framework presents a flexible way of splitting the landscape across Great Britain, generally using slope and accumulated area in addition to subcatchment boundaries and spatial rainfall fields (Coxon et al., 2019). Additionally, lumped hydrological models can be applied across large samples of catchments to gain a national picture (e.g large sample studies in France, Le Moine et al. 2007; Perrin et al. 2001 and elsewhere (Parajka et al., 2009; Poncelet et al., 2017).

This diversity in existing national model setups emphasises that there is no agreed upon best model structure, and there are many different approaches to developing a national modelling framework. Given the wide variety of model structures available, it is important to comprehensively evaluate models and ensure that the correct model is chosen for a given purpose. The following sub-sections discuss the advantages/limitations of different ways models discretize the landscape, the development of modelling frameworks which allow a flexible definition of model structure, and how we can test the value of different model structures through large-sample model intercomparisons and benchmarking studies.

2.2.1 Model resolution and approaches to discretising the landscape

Hydrological models differ in the way they distribute the landscape, with models of different spatial resolutions developed to support a range of model applications with different objectives, as highlighted by the variation in national models presented in the previous section. One of the simplest approaches are lumped hydrological models, which treat the catchment as a single modelling unit and aim to capture the emergent catchment behaviour. These models have been widely

applied, for applications such as operational hydrological forecasting (Velázquez et al., 2011), water resource planning (Christierson et al., 2012), and climate impact studies (Seiller and Anctil, 2014). The simplicity of these models is a major advantage – they have low data requirements, are easy to set-up and apply to a range of catchments, and are computationally efficient enabling ensemble simulations, large-sample applications, and uncertainty analysis. Many studies have also demonstrated that they are able to simulate catchment-outlet river flows as well as or better than spatially distributed models (Ghavidelfar et al., 2011; Reed et al., 2004; Tran et al., 2018).

However, to explore spatial differences across the landscape (such as spatially varied precipitation/evapotranspiration inputs, variation in hydrological processes, and differences in model stores across the catchment) we need spatially distributed hydrological models. Distributed models discretize the catchment into large number of elements or grids, each of which can have different state variables or stores which are computed separately (Beven, 2012). This can improve model performance in catchments where spatially varied processes are important, for example Lobligeois et al. (2014) demonstrated that using distributed rainfall inputs improved model performance relative to homogeneous rainfall inputs in catchments where precipitation is spatially variable. The distributed approach is also required for applications which need predictions to be distributed in space, for example analysis of changing rainfall patterns or land-use on river flows, and for predictions of state variables within a catchment.

The distributed modelling approach has drawbacks (Beven, 2001). Firstly, the model equations must be solved for each model element increasing model run times relative to lumped approaches. This increased run-time can hinder the running of ensemble simulations and uncertainty analysis. Secondly, the approach is limited by data availability and quality. Thirdly, the need for parameters at each model element greatly increases difficulties with model parameterisation, as described in following sections. There is therefore a trade-off between the level of detail required to represent the landscape, and the time it takes to run a model simulation. Many national and large- scale models are grid-based (e.g. mHM, VIC, G2G), which can make them computationally costly to run if applied at high resolution. The requirement for model simulations which are relevant at local to continental scales has promoted the development of hyper-resolution (< 1 km) hydrological models (Bierkens et al., 2015), further increasing model run times. These computational constraints prevent multiple runs, limiting capabilities to evaluate model parameter values and characterise core modelling uncertainties (Clark et al., 2017).

To reduce distributed model run times without compromising resolution, some hydrological models discretize the landscape into hydrological response units (HRUs) rather than grids. These are areas which are expected to behave hydrologically similarly, for example groups of cells which may share the same steepness, soil properties or land-use (see Figure 2.1 for an example of

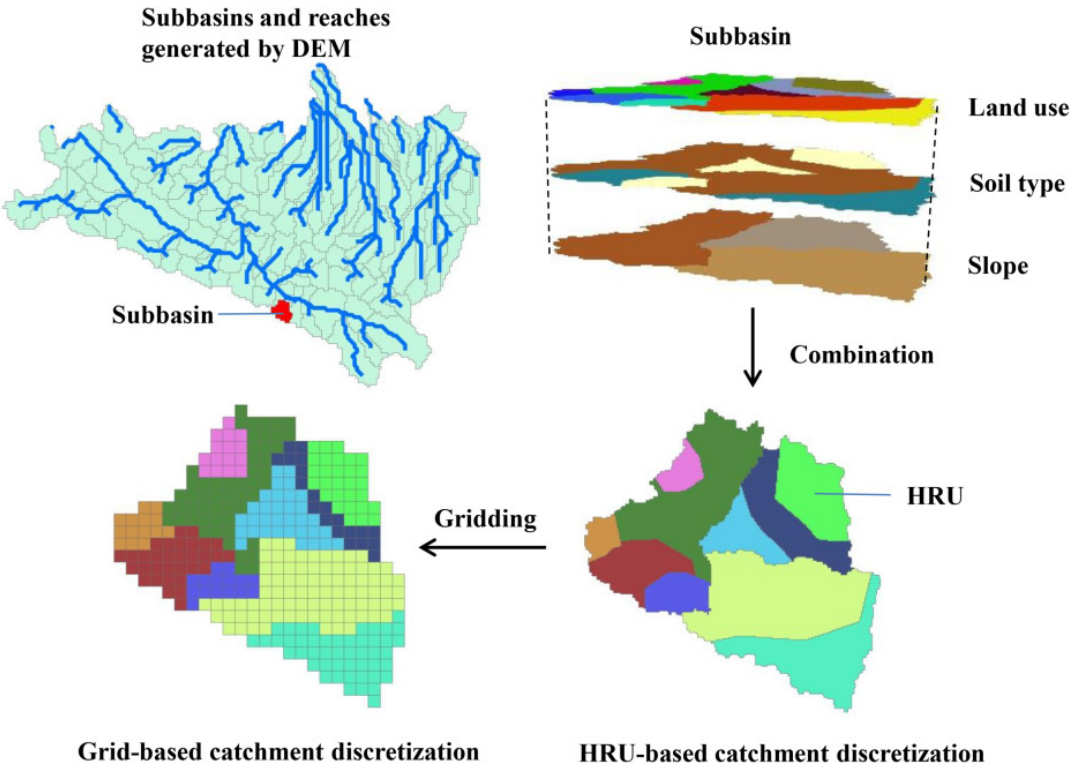


FIGURE 2.1. Demonstration of gridded vs HRU-based catchment discretisation. These show watershed delineations from SWATGP (grid-based) and SWAT (HRU based). Reproduced with permission from Zhang et al. (2017).

gridded vs HRU-based catchment discretisation). This then reduces the number of calculations required in the modelling process, as equations are evaluated over HRUs rather than individual grid cells, whilst still ensuring that the key hydrological features are represented in the landscape. For example, Chaney et al. (2016) demonstrate how splitting the landscape into hydrological response units can generate similar fields of modelled variables to a fully distributed model, but with run times over 2 orders of magnitude shorter. Therefore, this approach provides an opportunity to create national models which characterise key differences across the landscape whilst retaining model efficiency, enabling exploration of modelling uncertainties.

2.2.2 Flexible model structures

Developing model structures which can capture the wide heterogeneity of hydrological processes across large scales is an ongoing challenge. Large-scale model structures must be able to simulate catchments with different climates and catchment properties, including human impacted as well as natural catchments (McMillan et al., 2016). Many national models apply a single model structure everywhere. For example, McMillan et al. (2016) use the TOPNET structure across New Zealand and Bell et al. (2009) use the Grid-to-Grid model structure across Great Britain.

However, it may not be possible for a single model structure to simulate such hydrologic diversity.

Recent research has demonstrated the value of varying model structure according to dominant processes (Coxon et al., 2014; Fenicia et al., 2014; Van Esse et al., 2013). There has therefore been a drive towards developing modelling frameworks with a flexible definition of model structure (Butts et al., 2004; Clark et al., 2008, 2015; Coxon et al., 2019; Knoben et al., 2019). These can then allow a user to specify different model structures across the landscape, to take advantage of the relative merits of different model structures and ensure appropriate model structures are used in different environments (Coxon et al., 2019). To support applications of these flexible model frameworks, comprehensive evaluations of the relative merits of different model structures across diverse catchment characteristics are needed to guide model structure selection (Gupta et al., 2014).

2.2.3 Multi-model comparisons

Given the large numbers of available model structures, it can often be difficult to distinguish which structure is most appropriate for a given aim. Choosing an appropriate model structure may be hampered by evaluations of individual model structures being based on different metrics and different catchments, preventing direct comparison of model structures and assessments of which are most appropriate for different modelling aims. This could contribute to the fact that model structural choice is often guided by legacy (i.e. driven by experience with a particular structure or institutional preference for a certain model structure) rather than adequacy (Addor and Melsen, 2019). To guide model structure selection, it is therefore useful to carry out model intercomparison studies, which evaluate multiple different model structures across a range of catchments. Studies directly comparing model structures can also highlight model structural components which lead to particularly good/poor model performance in different environments, leading to focused model improvements.

Model intercomparison studies have long been recognised as important, with organised model intercomparisons since the 1960s (WMO, 1975, 1986). For example, the WMO organised an intercomparison of conceptual hydrological models used in operational forecasting, inviting groups to demonstrate their models on 6 river flow timeseries given calibration data, and comparing model performance for a blind validation period (WMO, 1975). The Distributed Model Intercomparison Project (DMIP) compared 12 distributed models with lumped model simulations across five parent basins and three gauged sub-basins, to improve understanding of the applicability of distributed models across various scales (Reed et al., 2004). However, as highlighted by Andréassian et al. (2006), these model intercomparisons can only present useful results when they are carried out across a large-sample of catchments, as any conclusions based on small catchment samples could be a matter of luck (Andréassian et al., 2009).

More recent studies have taken advantage of increases in computation power and demonstrated the utility of large-sample model intercomparisons for improving our understanding of the relative merits of different model structures/ model structural components. For example: Perrin et al. (2001) studied the relationship between model complexity and model performance across 429 catchments, finding that whilst more complex models outperform simpler ones in calibration this performance gain does not continue beyond the calibration period; Perrin et al. (2003) demonstrated that the GR4J model structure was an improvement on the previous 3-parameter model version, whilst comparing it to a range of other model structures; and Mouelhi et al. (2006) demonstrate the poor performance of the Manabe bucket model relative to other model structures across 407 catchments, and subsequently improve the model.

The development of flexible modelling frameworks provides new opportunities for model structural comparisons (Clark et al., 2008, 2015; Fenicia et al., 2014; Knoben et al., 2019; Leavesley et al., 1996). Flexible modelling frameworks support different hydrological model structures within the same overarching framework, for example the MaRRMoT framework contains model code based on 46 conceptual model structures, FUSE combines the structural decisions of 4 commonly used models to create hundreds of possible structures, and SUPERFLEX provides generic model components which can be combined in different ways to create unique model structures (Clark et al., 2008; Fenicia et al., 2011; Knoben et al., 2019). This simplifies the application of multiple hydrological model structures across large catchment samples, as model setup and generation of input data only needs to be done once, and allows direct comparison of different model structural components.

Flexible modelling frameworks have been applied in large-sample model comparison studies. Coxon et al. (2014) evaluated 78 FUSE model structures across 24 British catchments within an uncertainty analysis framework, finding large differences between model performance between catchments and depending on the metric used for model evaluation. Van Esse et al. (2013) applied the SUPERFLEX framework to 237 French catchments to demonstrate how the model structure can be identified as part of the modelling process, resulting in better performance than a fixed GR4J model structure. In this thesis, Lane et al. (2019) applied four FUSE models to over 1000 catchments, identifying how model performance was related to the catchment water balance, baseflow index and total rainfall. Whilst these frameworks tend to be limited to lumped or semi-lumped catchment models, they provide a valuable resource for testing and learning about different model structures over large samples of catchments. This can help us to understand which model structures are appropriate for different catchment types, and where model selection may be most important.

We therefore need to be evaluating model structures, to help guide model selection and improvement for national modelling approaches. Comparing multiple model structures across large catchment samples is one way of doing this, which can help to highlight the advantages/pitfalls of different model structures, improve our understanding of which model structural choices are appropriate for different climate-hydrological conditions, and focus model improvements by highlighting common model flaws (Andréassian et al., 2009). This improved knowledge and understanding of model structures is needed to develop national modelling frameworks.

2.2.4 Model benchmarking

A related concept to model intercomparison is model benchmarking. Benchmarking studies provide an evaluation of modelling skill, that can act as a baseline for any future advances or more complex models to be compared against as well as identifying any weaknesses (Newman et al., 2015; Seibert et al., 2018). Model modifications can then be evaluated in a consistent way to ensure continuous improvement and identify the factors improving model performance. Archfield et al. (2015) emphasise the importance of systematic assessments of model performance to assess how performance varies depending on model structure, parameterisation and hydroclimatic setting, and to ensure a model can adequately represent the properties of a hydrograph for a given management priority (for example, high flow/low flow performance may be more important for studies focused on floods/droughts respectively). Benchmarking of hydrological models is one way this can be achieved (Archfield et al., 2015).

Two recent studies demonstrate how lumped, conceptual models can be used as benchmarks when evaluating more complex hydrological models (Newman et al., 2017; Seibert et al., 2018). Newman et al. (2017) demonstrate how the use of a bucket-style hydrological model provides a useful benchmark in an evaluation of the physically-based VIC model across over 500 catchments in the United States. Seibert et al. (2018) make a case for the use of upper and lower benchmarks in hydrological modelling, demonstrating the use of a simple hydrological model to generate upper and lower performance benchmarks in an evaluation of the SHETRAN model across Great Britain. Figure 2.2 demonstrates this, showing how the SHETRAN model performs relative to the upper and lower benchmarks. These studies argue that conceptual hydrological models provide a good, practical benchmark for the evaluation of more physically based models, as they implicitly take observation uncertainties and water balance constraints into account.

Benchmarking of hydrological models can therefore help to evaluate national models and ultimately improve our national simulations. To facilitate the use of lumped hydrological models as benchmarks, there is a need for model simulation results and performance metrics to be made openly available across large-samples of catchments. Studies are therefore needed which provide these benchmark simulations to aid the continuous improvement of national modelling efforts.

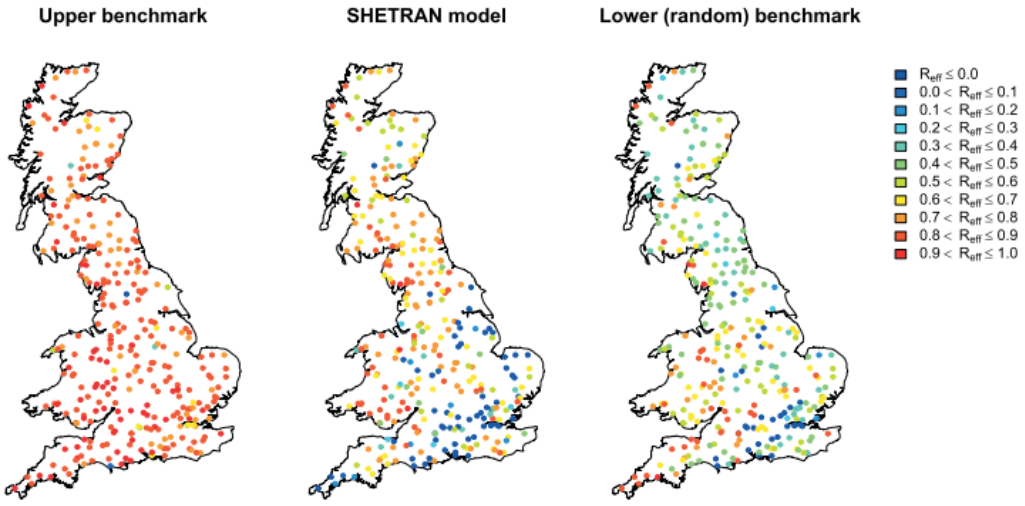


FIGURE 2.2. Results of a large-sample study demonstrating the use of upper and lower benchmarks when evaluating hydrological models. The upper benchmark is the calibrated HBV model and the lower benchmark was generated using random HBV parameters. These were then used to evaluate the uncalibrated SHETRAN model performance. Reproduced from (Seibert et al., 2018).

2.3 Model parameterisation

All models contain parameters, which are values used to describe the characteristics of the catchment area or modelling unit (Beven, 2012). Estimating model parameters is vital to ensure the model is realistically reflecting hydrological processes and to produce credible hydrological simulations (Archfield et al., 2015). Generally, it is difficult to estimate model parameters a priori, as model parameters are often effective values which cannot be observed at the model spatial scale such as the average depth of water storage capacity (Beven, 2012). Some more physically based parameters, such as saturated hydraulic conductivity, are measurable in theory but the scales at which they are measured are incommensurate with the modelling scale at which they are needed (Blöschl et al., 2013; Pechlivanidis et al., 2011). Therefore parameters usually require calibration to achieve the best match between model outputs and available observations of the actual catchment response (Beven, 2012; Pechlivanidis et al., 2011).

There are many different approaches to model parameter calibration. The most basic approach is manual calibration of model parameters, where parameters are tuned by an expert through a trial-and-error process (Boyle et al., 2000; Madsen, 2003). However, this is complicated, highly labour-intensive and requires expertise which is not easily transferred (Boyle et al., 2000). It is also a subjective process, and as it is difficult to determine an end point of the process different results will be obtained by different modellers. Therefore, a large number of automatic calibration methods were developed which find the best parameter set(s), as judged by one or

more objective functions/ performance metrics (e.g. Duan et al. 1992; Jiang et al. 2013; Solomatine et al. 1999). This includes local and global optimisation methods (Duan et al., 1992; Jiang et al., 2013; Solomatine et al., 1999), as well as Monte Carlo parameter sampling which evaluates the whole parameter space (Beven, 1993; Freer et al., 1996; Wagener and Kollat, 2007).

Early studies found that similar model performance was possible from many different parameter sets, sometimes from different parts of the model parameter space (Beven, 1993; Beven and Freer, 2001a; Freer et al., 1996; Hornberger et al., 1985). This equifinality or non-uniqueness of model parameters gave rise to the view that it is not possible to find an optimum parameter set, and we should instead consider all parameter sets which are deemed to be an acceptable simulator of the system (Beven and Binley, 1992). One way of approaching this is through the Generalised Likelihood Uncertainty Estimation (GLUE) methodology, which samples the parameter space through random sampling and then assigns a performance score to each parameter set based on its ability to simulate the system (Beven, 1993; Beven and Freer, 2001a)). Simulation results from all behavioural parameter sets (those with a performance score above a set threshold) are then used to make hydrological predictions with uncertainty limits, with greater weight given to the parameter sets with better performance scores. This approach of constraining model parameter sets rather than optimising therefore enables evaluation of the impact of parameter uncertainties on hydrological predictions. It is vital that these parameter uncertainties are considered in modelling studies, to prevent overconfidence in model results, and to demonstrate the range in flow estimates arising from parameter sets which are all equally plausible.

Additional model parameterisation difficulties and uncertainties are introduced when moving from lumped to distributed models. Whilst lumped hydrological models apply a single parameter value across the catchment, distributed hydrological models often require spatial fields of model parameters (Mizukami et al., 2017). This allows distributed hydrological models to represent spatial variation in hydrological response across the landscape. However, it complicates the issue of model parameterisation by massively increasing the number of parameters to be calibrated – instead of a single set of model parameters across the catchment, the parameters must be estimated for each modelling unit. Different spatial configurations of model parameters may produce equally valid simulations, introducing further difficulties with the non-identifiability and equifinality of parameter sets (Beven, 1993). Also, the use of streamflow data for calibration of large numbers of parameter sets results in an ill-posed problem, as there is not sufficient information content in the streamflow data to support the robust calibration of parameter values (Beven, 2012). Calibration for distributed models therefore typically uses some form of spatial regularization (Pokhrel et al., 2008), to pre-define the spatial distribution of parameters across a catchment and therefore reduce the degrees of freedom in the calibration.

Constraining model parameters across large domains brings challenges including how to constrain parameters in ungauged basins, how to estimate parameters for interacting nested catchments and inclusion of parameter uncertainties alongside the computational challenges of applying a model across a large-area/ large catchment sample. This section reviews approaches to define model parameters across large domains. First, the challenges of a priori parameterisation are described. Second, studies which focus on gauged catchments only are discussed, and issues with using individual gauge calibrations to provide national simulations are highlighted. This leads on to a review of parameter regionalisation techniques, which provide different ways to estimate parameters for ungauged areas.

2.3.1 A priori parameter estimation

For more physically based hydrological models, parameters are sometimes estimated a priori (i.e. before modelling has taken place) from field data or satellite measurements (Beven, 2012; Blöschl et al., 2013). This can also be referred to as an uncalibrated model, as parameters are linked to catchment properties without calibration (McMillan et al., 2016). There are many difficulties with defining a priori parameter fields, including 1) it is inappropriate for conceptual parameters which do not directly relate to measurable, physical properties, 2) the scale at which measurements are available is incommensurate with the scale at which model parameters are applied, 3) many physical parameters cannot be measured directly, and must be estimated using transfer functions, which introduces uncertainties in the mathematical form and coefficient values to be used in these functions Mizukami et al. (2017). Due to the difficulties of a priori parameter estimation, parameter calibration is usually required to improve model performance (Beven, 2012; Mizukami et al., 2017).

Nevertheless, a priori parameter estimation has been used for previous national scale models. Koren et al. (2000) used transfer functions to develop parameter fields based on soil datasets, and applied these across the US for the National Weather Service Sacramento Soil Moisture Accounting model (SAC-SMA). In a national model for New Zealand, McMillan et al. (2016) relate the 31 parameters of the TOPNET model to national datasets of topography, land cover, soil texture, and other physical properties. They identified the most sensitive model parameters as the TOPMODEL f parameter (also referred to as the m or SZM parameter in the deficit model versions, describing the form of the exponential decline in conductivity with depth) and the subsurface saturated hydraulic conductivity, and focused their efforts on ensuring the a priori estimates of these parameters were suitable. The Grid-to-grid model applied across the UK is also an uncalibrated model, with parameter values set based on catchment characteristics (Bell et al., 2009).

The a priori parameter estimation technique has the advantage that relating model parameters to spatial geophysical data helps to define the spatial pattern of parameter values, in a

way that has a physical basis (assuming realistic relationships between model parameters and geophysical data have been identified). However, a priori parameter fields are difficult to generate. There will always be limitations such as the applicability of data used for parameter estimation and uncertainties in transfer function parameters, and therefore some form of parameter calibration can help to improve upon a priori model parameterisations.

2.3.2 Parameter calibration for gauged catchments

Modelling across gauged catchments means that model parameters can be calibrated against observed discharge at the catchment outlet. Many national modelling efforts have modelled a large sample of catchments across a region to provide the nationwide picture. This means that ungauged areas will remain unmodelled, but for countries with dense gauge networks, such as the UK, this can still provide a good national coverage. For example, Prudhomme et al. (2013b) present their national future flows dataset for 281 river catchments across GB, and Christierson et al. (2012) present a national assessment of climate impacts for water resource planning based on 70 UK catchments. For this thesis, modelling has been carried out across a large sample of gauged catchments in Great Britain.

However, calibrating model parameters to individual catchments may not result in a coherent national picture. By constraining parameters separately to each catchment, or sub-catchment, fields of parameter values are not spatially coherent. Sharp discontinuities can arise between catchment boundaries leading to a 'patchwork quilt' map of model parameters, even where there are no distinct differences in the landscape that would justify differences in parameter values. This is an issue if the modelling study is aiming to show a consistent national picture, as inconsistencies in spatial parameter fields can lead to inconsistencies in simulations at the national scale. Mizukami et al. (2017) demonstrated this, giving an example of a climate impact study performed across the US (Reclamation, 2014, 2016). The study created national parameter fields by collating sub-regional calibrated parameter values. The resultant parameter fields had discontinuities across major catchment boundaries, which resulted in discontinuities in modelled surface flow and baseflow, as shown in Figure 2.3. It is therefore important to ensure parameters are consistent and realistic across the landscape for national studies.

Only calibrating to gauged catchments also raises questions for how to calibrate parameters for gauged sub-catchments and spatial fields of parameters more generally. For models which use distributed parameter fields, it is important to reflect realistic spatial differences in parameter values, whether this is between gauged sub-catchments, hydrological response units or gridded across the landscape. If the variation in internal model parameters is physically realistic, then distributed parameters could improve the internal consistency of the model and enable runoff predictions at ungauged points within the catchment. However, how to define these differences

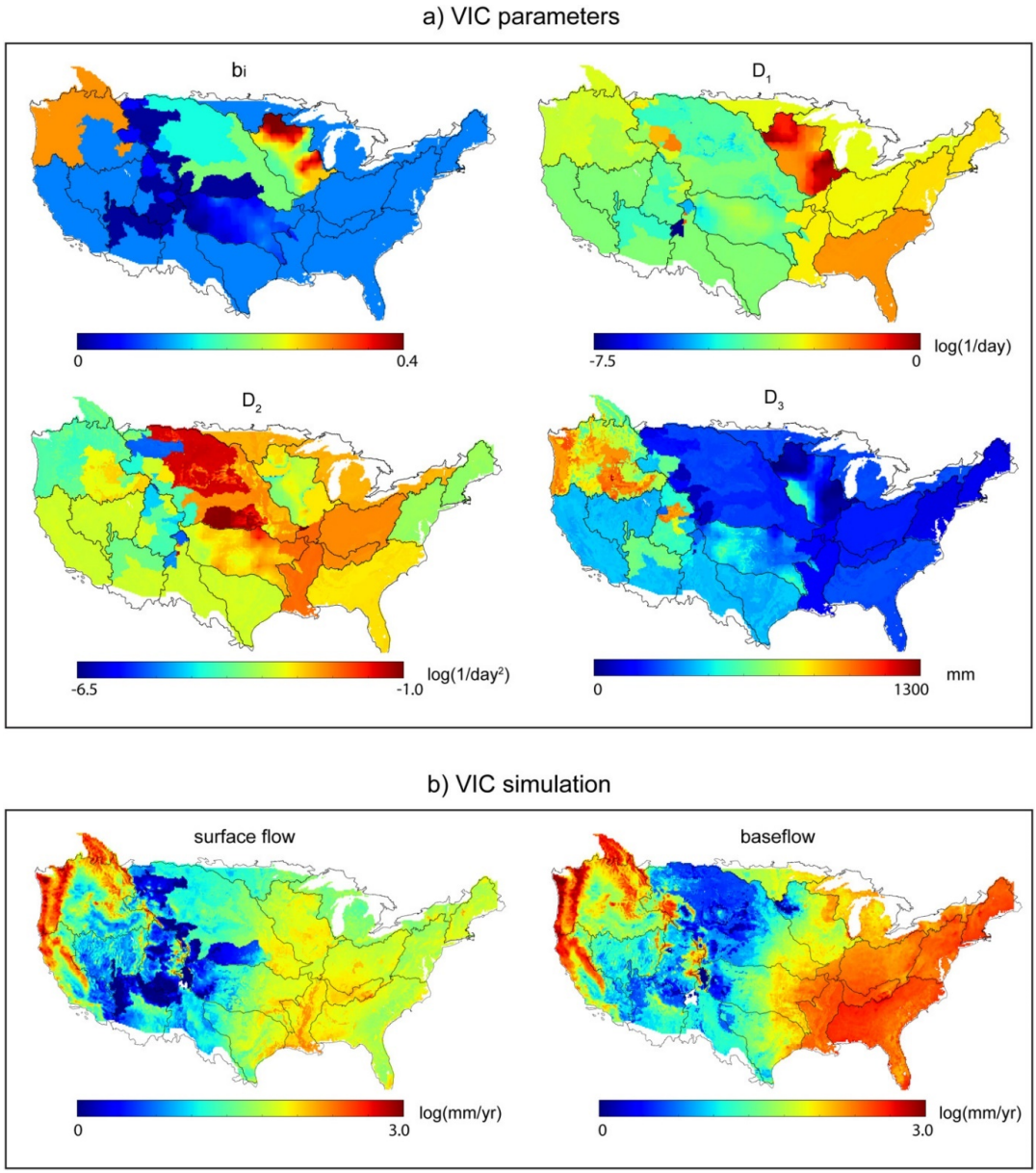


FIGURE 2.3. VIC infiltration and baseflow parameters (a) and resultant VIC mean annual surface flow and baseflow simulations 1950-1999 (b), produced for a study on climate changes impact on water resources across CONUS (Reclamation, 2014, 2016; Wood and Mizukami, 2014). The black boundaries show large river basin boundaries, demonstrating how inconsistencies in parameterisation methods for different large river basins result in inconsistencies in modelled hydrological variables. Reproduced with permission from Mizukami et al. (2017).

in parameter values across the landscape is challenging, as many different variations of spatial parameter fields could produce the same simulated streamflow. There will also be interactions between parameter values across the catchment, for example altering parameters of the headwater catchments will impact downstream flows.

One option to calibrate parameters for gauged sub-catchments would be to take a step-wise approach to model calibration. This could involve first calibrating the parameters of headwater catchments, and then moving downstream in a step-wise way until parameter values are calibrated for the whole catchment. The semi-distributed CLASSIC hydrological model follows this type of nested calibration approach (Crooks and Naden, 2007). This approach could therefore represent the spatial variation in parameter values for lumped sub-catchments. However, it could result in a very complex series of calibrations in catchments with large numbers of nested catchments and any unrealistic parameterisations of upstream catchments, for example due to errors in discharge data, would be propagated downstream. It could easily become very computationally expensive if quantifying model prediction uncertainties. It also does not provide a means of parameterisation for ungauged points within the catchment.

Another option for calibration of distributed parameters is to use some form of spatial regularisation to pre-define parameter values across a catchment which can then be calibrated (Pokhrel et al., 2008). The first way this can be done begins with the estimation of a priori parameter fields, as described above. Spatially constant parameter multipliers can then be applied to these fields, with the calibration subsequently focusing on the parameter multipliers rather than the model parameters themselves (Pokhrel and Gupta, 2010; Pokhrel et al., 2008). However, this method requires a priori parameter fields, which are difficult to generate. A second method applies spatially constant transfer functions linking model parameters to geophysical data across the modelling domain (Hundecha and Bárdossy, 2004; Hundecha et al., 2008; Mizukami et al., 2017; Samaniego et al., 2010). The calibration then tunes the coefficients of the transfer functions instead of the model parameters. This can result in parameter fields which are consistent across the landscape with a physical reasoning for the variation in parameters within a catchment. This is therefore a promising technique for calibrating distributed parameter fields, for local to national scales, which reflects realistic variation in parameter values within a catchment and between sub-catchments.

2.3.3 Predictions in ungauged catchments

When creating national models it is often desirable to model ungauged areas, which requires some form of parameter regionalisation. There is now a large literature on parameter regionalisation for predictions in ungauged basins, following the IAHS Prediction in Ungauged Basins (PUB) initiative (Hrachowitz et al., 2013). These regionalisation studies can be broadly grouped into 1)

regionalisation of flow and flow metrics, 2) post-regionalisation of model parameters from donor to ungauged catchments, 3) simultaneous parameter regionalisation which are all summarised with examples in Table 2.1. This section evaluates the latter two approaches, which regionalise model parameters based on catchment characteristics.

TABLE 2.1. A summary of parameter regionalisation methods for predictions in ungauged catchments with example applications.

Regionalisation type	Description	Examples
Calibration based on spatially interpolated hydrologic indices	Model calibration at ungauged basins based on regionalised hydrologic signatures (e.g. annual average runoff, high and low flow metrics)	Yadav et al. (2007) Oubeidillah et al. (2014)
Post-regionalisation (i.e. transfer of parameters from gauged to ungauged catchments)	Regression of individual calibrated parameters to catchment characteristics (e.g. soil, topography, climate)	Abdulla and Lettenmaier (1997) Young (2006) Carrillo et al. (2011)
	Transfer of entire parameter sets based on catchment spatial proximity	Nijssen et al. (2001) Merz and Blöschl (2004) Parajka et al. (2005) Oudin et al. (2008) Troy et al. (2008)
	Transfer of entire parameter sets based on catchment similarity	Parajka et al. (2005) Oudin et al. (2008) Beck et al. (2016)
Simultaneous regionalisation (i.e. estimation of spatially constant transfer function coefficients simultaneously across all gauges)	Parameters are linked to catchment attribute data via transfer functions. The coefficients of the transfer functions (aka global parameters/ transfer function parameters) are calibrated simultaneously across all gauged catchments. The calibrated transfer function coefficients are then applied across both gauged and ungauged areas.	Seibert (1999) Hundecha and Bárdossy (2004) Hundecha et al. (2008)
	Multiscale Parameter Regionalisation: as above, but transfer functions are applied to the catchment attribute data at the highest possible resolution, and then model parameter values are upscaled (i.e. through some averaging function) to the resolution of the model.	Samaniego et al. (2010) Kumar et al. (2013a) Mizukami et al. (2017) Samaniego et al. (2017)

2.3.3.1 Post-regionalisation techniques

Post-regionalisation techniques typically use catchment attribute data to transfer parameters from gauged to ungauged catchments. The general approach is to constrain parameters on gauged 'donor' catchments and then develop a method to transfer parameter values to ungauged catchments based on hydrological similarity, climate or spatial proximity (Blöschl et al., 2013). This is referred to as post-regionalisation, as the parameter regionalisation takes place after model simulations have been run. There is a large body of literature on how parameters can be transferred (Blöschl et al., 2013; Merz and Blöschl, 2004; Oudin et al., 2008; Parajka et al., 2005; Razavi and Coulibaly, 2013), with key approaches including regression equations linking individual parameters to catchment characteristics and transfer of entire parameter sets based on catchment proximity or similarity.

The first approach transfers parameters individually by forming regression equations linking model parameters to lumped catchment attributes (Sefton and Howarth, 1998; Wagener et al., 2004). This relies on parameters having meaningful relationships with the chosen catchment attributes, which can then be extrapolated to ungauged catchments. However, it is often the case that no significant relationships can be found between model parameters and catchment attributes, and therefore this approach cannot be applied (Blöschl et al., 2013). Another problem with this approach is that it does not consider parameter interactions, which can play a large role in determining appropriate parameter values.

The second approach implicitly includes parameter interactions, through selecting the most similar donor catchment(s) and transferring the entire parameter set(s). This method assumes that if two catchments are hydrologically similar, then their runoff response should also be alike and therefore they may share model parameter values Blöschl et al. (2013). There are three main ways to select similar donor catchments: based on spatial proximity; based on catchment similarity; or through model averaging.

The spatial proximity method transfers the parameters of the closest gauged catchment. This simple approach often works well, particularly in densely gauged areas, as if climate and catchment characteristics vary smoothly across a region then it is often the case that catchments behave the most hydrologically similarly to their neighbours (Oudin et al., 2008; Parajka et al., 2005). The catchment similarity method transfers the parameters of the most hydrologically similar catchment. Here, similarity is defined using climate and catchment characteristics that the modeller considers an important in controlling catchment response to rainfall, such as catchment area, standardised annual average precipitation and baseflow index (McIntyre et al., 2005). In an evaluation of regionalisation methods over France, Oudin et al. (2008) found that whilst the spatial proximity method resulted in the best model performance, both spatial proximity and

catchment similarity far outperformed regression for networks with a streamgauge density over 30 stations per 100,000 km^2.

The model averaging approach uses a weighted combination of the parameter sets from multiple donor catchments (Blöschl et al., 2013). Here, donor catchments could be selected using the spatial proximity or similarity methods described above, but a key difference is that parameter values from multiple donor catchments are used. This represents some of the modelling uncertainties arising from the model parameterisation.

Parameter equifinality presents difficulties for these regionalisation methods (Wagener and Wheater, 2006). The parameter regionalisation methods described above largely assume that an optimal parameter set can be found for gauged catchments, as the basis for regression equations or transfer of parameter sets to ungauged catchments. However, this is usually not the case. To overcome this issue, options include (1) reducing the number of free parameters by fixing some of them, (2) calibrating hydrological model parameters against additional information if available (e.g. Koppa et al. (2019) calibrate the model using evapotranspiration and streamflow simultaneously, and Rakovec et al. (2016b) calibrate a European model using water storage as well as streamflow), or (3) calibrating model parameters simultaneously across multiple gauges (e.g. Bárdossy et al. (2016) demonstrate that simultaneously calibrating dynamical parameters across all catchments resulted in parameters which perform well across the region, whilst calibrating catchments individually did not). Calibrating parameters simultaneously across multiple gauges provides additional information (multiple flow timeseries rather than just one gauge) reducing the equifinality problem and resulting in more robust parameter estimates. This method is often referred to as simultaneous parameter regionalisation.

2.3.3.2 Simultaneous parameter regionalisation

Simultaneous parameter regionalisation, sometimes referred to as simultaneous calibration or regional calibration, aims to establish relationships between parameters and catchment characteristics, and then constrain the coefficients of these relationships across many gauges simultaneously. This means that instead of the two-step procedure described in section (2.3.3.1), of first calibrating model parameters and then transferring them based on catchment characteristics, parameter regionalisation applies these steps simultaneously. There are generally two approaches: (1) for lumped models, the coefficients linking model parameters and average catchment characteristics are constrained, (2) for distributed or semi-distributed models the coefficients linking model parameters to catchment properties at the grid or other scale are constrained. Here we focus on the second approach.

To apply this method, model parameters are linked to relevant catchment attributes via

transfer functions with spatially constant global parameters to be constrained. For example, a saturated hydraulic conductivity parameter may be linked to spatial maps of soil properties, via a transfer function of the form ksat = a(%sand) + b(%clay) + c (Cosby et al., 1984). The coefficients of the transfer function (also commonly referred to as global parameters or transfer function parameters), here a, b and c, then need to be constrained instead of the model parameters themselves. This is done by evaluating the performance of each set of coefficients across all modelled gauges simultaneously. A spatially constant set of global parameters (i.e. a, b and c) will then be applied across the whole domain to create a consistent model parameter field (Mizukami et al., 2017).

Simultaneous parameter regionalisation has several advantages over the previously discussed parameterisation methods. Firstly, by constraining parameters across many gauges simultaneously it results in more robust parameters which are less impacted by unique conditions in specific gauges. Instead of constraining model parameters against a single flow timeseries, they are constrained against multiple, reducing problems relating to over-parameterisation. Secondly, it allows use of the spatial information given in the catchment data to distribute parameters in space. As described above, this means that it is useful as a spatial parameter regularization technique as well as for parameterisation of ungauged catchments. Thirdly, it provides consistent parameter fields across large areas. Finally, whilst regional calibration may not always improve predictions at ungauged sites relative to other methods, it has been shown to improve the relationship between model parameters and catchment characteristics (Fernandez et al., 2000; Parajka et al., 2007a; Szolgay et al., 2003)). This makes it a good regionalisation choice for modelling environmental change, as having a physical basis for the parameter values and more realistic parameter fields is advantageous when extrapolating to different environmental conditions (Blöschl et al., 2013).

Multiscale parameter regionalisation (MPR) has emerged as a particularly effective simultaneous parameter regionalisation strategy because it takes sub-grid variation in catchment attribute data into account (Samaniego et al., 2010). Usually, catchment attribute data is aggregated to the modelling scale (e.g. grid cells, sub-catchments, or hydrological response units) before transfer functions are applied. However, MPR applies transfer functions at the native resolution of the catchment attribute data, and then averages parameter values to the modelling scale. A schematic demonstrating these core differences is given in Figure 2.4. This means that estimated transfer function coefficients are less sensitive to modelling scale, and can be more effectively transferred to different model resolutions (Kumar et al., 2013a,b; Samaniego et al., 2010, 2017). The method has shown to produce good results when transferring parameter fields to ungauged basins (Kumar et al., 2010, 2013a,b), and has successfully been applied to produce seamless, consistent parameter fields for large-scale hydrological model applications (Mizukami

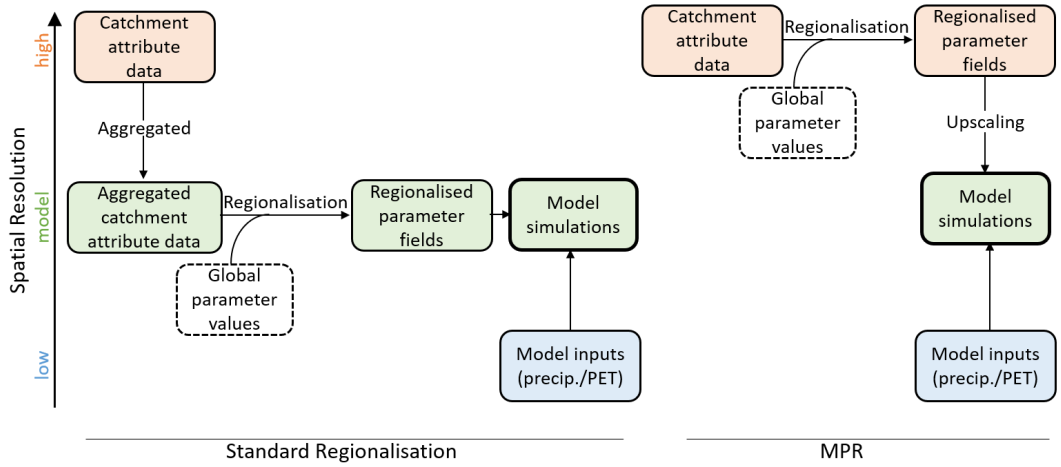


FIGURE 2.4. Schematic comparing multiscale parameter regionalisation (MPR) with a standard simultaneous regionalisation approach. High spatial resolution refers to the highest resolution of the geophysical data used for the regionalisation. The key difference between MPR and standard regionalisation is shown to be the scale at which the regionalisation takes place. Based on a figure from Samaniego et al. (2010)

et al., 2017; Samaniego et al., 2017). This could make it a promising method for parameterisation of a national model.

2.4 Uncertainty

Hydrological modelling is far from a precise science, and many uncertainties remain (Beven and Freer, 2001a; Mcmillan et al., 2012; Westerberg and Birkel, 2015), with the importance of characterising and communicating modelling uncertainties recognised since early studies (Beck, 1987; Cooke, 1906; Garen and Burges, 1981). Hydrological modelling uncertainties are largely epistemic arising from incomplete understanding of the environmental system, including errors in the data we use to develop, parameterise, and drive hydrological models, errors in the choice of model structure, errors in identification of model parameter values, and the fact that models are inherently imperfect representations of the real world. It is important to represent this uncertainty in modelling results, to aid interpretation of model outputs and fairly demonstrate the range of possible outcomes (Pappenberger and Beven, 2006). This is especially important for simulations of change and hydrological extremes, where uncertainties may be particularly high.

When modelling across large scales, recognising and including modelling uncertainties can be challenging for many reasons. Firstly, modelling across a large region means that local knowledge

about the peculiarities of each gauging station/ catchment including data quality cannot be taken into account in the same way as detailed catchment studies (McMillan et al., 2016). Secondly, there may be limited availability and quality of data on a national scale, and so the modeller must work within the limitations of the available data. Thirdly, computational challenges involved with the setup and running of large-scale models often prohibits the application of computationally costly uncertainty analysis techniques. Whilst evaluating uncertainties arising from model structure and parameterisations through an ensemble of simulations is common practice in catchment hydrology, this is difficult to achieve at the nationwide scale. This section discusses a few of the main sources of uncertainty for hydrological modelling studies, and uncertainty estimation techniques.

2.4.1 Model structure and parameter uncertainty

The wide array of available model structures was highlighted in section 2.2. These model structures differ in many ways, including spatial and temporal resolution, which stores are represented, which processes are included, the mathematical equations used to represent these processes and stores, and how these equations have been converted into model code (Knoben et al., 2019). The selection of model structure is not straightforward, as often many different model structures can achieve similar performance, contributing to the overall uncertainty in a hydrological modelling study. Similarly, there are uncertainties in the estimation of model parameter values, as summarised in section 2.3, with many different parameter sets often found to result in similarly plausible model simulations. Carrying forward all model structures/ parameter sets which can produce plausible simulations can improve understanding of the impact of model structural choice/ parameterisation on model output uncertainty.

2.4.2 Data uncertainty

Calibration of hydrological models often attempts to minimise the difference between observed and simulated river flow, which assumes that observations reflect the truth. However, hydrological data is highly uncertain (Coxon et al., 2015; Mcmillan et al., 2012). This includes uncertainties in input data (e.g. rainfall and potential evapotranspiration), discharge data used for model calibration/evaluation and other datasets used for model setup and parameterisation (e.g. digital elevation models, land-use and soil maps). In a recent review McMillan et al. (2018) find that the magnitude of hydrologic data uncertainty is generally within the range of 10-40%, although deriving rainfall measurements have been shown to introduce uncertainties of up to 100%. Understanding and accounting for the uncertainties in hydrological data is important to prevent bias and incorrect conclusions (Mcmillan et al., 2012). However, this is especially challenging at

the national scale, where modelling studies must rely on nationally available datasets and local catchment knowledge on data quality may not be readily available.

2.4.3 Approaches to uncertainty estimation

There are many approaches to uncertainty estimation for hydrological modelling (Beven, 2009), with the suitability of different uncertainty estimation methodologies a topic of much debate within the literature (Beven, 2006; Beven et al., 2003; Clark et al., 2011a; Gupta et al., 2003; Thiemann et al., 2001). These frameworks differ in the underlying philosophy of whether they are based around finding a single optimum or the concept of equifinality, as well as the methods used to sample the model space, and which uncertainties are included (Coxon, 2015). Two widely applied yet contrasting approaches are formal Bayesian statistical methods and the informal GLUE methodology.

Bayesian statistical approaches have been widely applied to quantify hydrological modelling uncertainties (Kavetski et al., 2006; Thyer et al., 2009; Vrugt and Ter Braak, 2011; Yang et al., 2007; Yen et al., 2014). The Bayesian approach attempts to represent all sources of uncertainty, with the assumption that all uncertainties can be represented as if they are statistical in nature (Beven, 2012). This requires an explicit formulation of the error process and definition of probability models for each uncertainty source. A key benefit of this approach is that it attempts to disentangle the different uncertainty sources (input, output, parameter, and model structural error) which is key to improving model simulations (Vrugt et al., 2009). However, these formal approaches make strong assumptions about the statistical properties of the error residuals, which may not be appropriate given that many sources of error in hydrology are epistemic (lack of knowledge) rather than aleatory (random) in nature (Beven, 2012).

The Generalised Likelihood Uncertainty Estimation (GLUE) procedure, first introduced in section 2.3, follows a nonstatistical approach to uncertainty estimation based on the concept of equifinality (Beven, 1993; Beven and Freer, 2001a; Blazkova and Beven, 2009b; Freer et al., 1996). The GLUE methodology recognises that hydrological modelling uncertainties tend to be epistemic in nature, with correlated and non-stationary errors, that do not conform to formal statistical assumptions. The premise behind the GLUE methodology is that all possible modelling combinations (i.e. model structures/parameter sets) are evaluated, and any approaches that are considered to be an acceptable predictor of the system are kept. These remaining behavioural models are assigned a likelihood weighting, based on their likelihood of being a good predictor of the system, which is usually evaluated using one or multiple performance metrics. These performance metrics can include allowances for observation errors, through a limits of acceptability approach (Blazkova and Beven, 2009a; Liu et al., 2009). River flow predictions are then generated using a combination of all behavioural models, with the likelihood weighting of each model

determining its contribution. This therefore facilitates inclusion of competing model structures as well as model parameter sets, in a framework which can include data uncertainties.

2.5 Using hydrological models for impact studies

Climate change could increase flood risk across Great Britain; we need to understand the potential future change to ensure adaptation measures are put in place. Climate change is expected to intensify the hydrological cycle, with the latest UK Climate projections indicating an increased chance of wetter winters and an increase in the frequency and intensity of extremes (Met Office, 2019). This could lead to an increase in the frequency and magnitude of flood events in the future. However, changes to flow will depend on catchment-specific hydrological processes as well as changes in rainfall and evapotranspiration (Wheater, 2006). For example, in responsive catchments peak river flows may increase in response to higher intensity precipitation. For low-lying catchments with deep soils and/or permeable geology, the balance between increased rainfall and increased evapotranspiration may be more important and future changes in flood hazard are more uncertain (Bell et al., 2009; Prudhomme et al., 2013a; Reynard et al., 2004; Wheater, 2006). Therefore, to quantify potential future changes to river flow, and identify areas of the UK which may be worst hit by increased flood hazard in the future, we need a consistent approach to national scale hydrological modelling studies.

Studies modelling climate change impact on river flows generally use climate model data to indicate future changes in hydrometeorological variables. Often a modelling chain is formed, with climate model output (e.g. precipitation/ temperature/ potential evapotranspiration) being fed into a hydrological model to produce change in river flows (Arnell and Reynard, 1996; Wilby et al., 2008). Typically, regional climate models (RCMs) are used, which can provide climatic variables at scales in the region of 12 - 50 km. RCMs are often lower resolution than hydrological models, and can have substantial biases in hydrometeorological variables such as precipitation and temperature. These include issues such as RCMs overestimating the number of wet days, general over- or underestimation of climatic variables and incorrect seasonal variation (Teutschbein and Seibert, 2012). Therefore, many studies include downscaling and/or bias correction steps before using climate model data as input to the hydrological model (Cloke and Pappenberger, 2009; Smith et al., 2014a; Teutschbein and Seibert, 2012). These bias correction methods range from relatively simple monthly-mean scaling factors to more sophisticated probability mapping approaches. However, the choice of bias correction methodology can have a large impact on the modelling results, in some cases even changing the sign of the projected changes in flood frequency (Cloke et al., 2013; Smith et al., 2014a).

The use of climate model data means additional uncertainties are added to the modelling process. Climate uncertainty predominantly arises from three sources: the internal variability of the climate system, climate model uncertainty and future scenario uncertainty (Hawkins and Sutton, 2009). Additionally, uncertainties are added from the choice of downscaling/ bias correction used to transform climate model data so that it is suitable for hydrological modelling (Cloke et al., 2013; Muerth et al., 2013; Smith et al., 2014a). These are in addition from the considerable uncertainties already involved in national scale hydrological modelling. Many studies have attempted to quantify the impact of these various uncertainties on a selection of catchments (e.g. Cameron et al. 2000; Cloke et al. 2013; De Niel et al. 2019; Engin et al. 2017; Kay et al. 2009; Meresa and Romanowicz 2017; Smith et al. 2014a; Wilby and Harris 2006). These generally conclude that climate models are one of the largest uncertainties in climate-impact studies focused on peak flows, but other uncertainties are not negligible. Meresa and Romanowicz (2017) and Engin et al. (2017) both find that hydrological model parameterisation is an important contribution to total uncertainty, although it is most important when modelling low flows. Overall, these studies highlight the importance of uncertainty assessment in climate impact studies.

2.5.1 Climate impact on flooding in Great Britain

There is a large body of literature on climate impacts in Great Britain (GB), but very few studies have evaluated changes in high flows / flooding in a nationally consistent way whilst also reporting the hydrological modelling uncertainties. Many climate impact studies for GB are based on the UKCP09 Climate Projections (e.g. Bell et al. 2012, 2016; Christierson et al. 2012; Kay et al. 2014a; Prudhomme et al. 2012), which included probabilistic projections giving average changes in meteorological variables, as well as an ensemble of 11 spatially consistent projections (Murphy et al., 2010). This has facilitated incorporation of RCM uncertainties into hydrological climate impact studies. For example, Prudhomme et al. (2012) produce national projections of the change in seasonal mean flows, showing the variation between the 11 RCM scenarios. However, incorporation of the hydrological modelling uncertainties at a national scale remains rare. A notable exception is Christierson et al. (2012) who evaluated changes to monthly and seasonal river flow projections nationally, using the UKCP09 probabilistic projections and multiple hydrological models and parameterisations within an uncertainty framework. They used a monthly change factor approach, where observed time-series were modified to create 'future' projections. This provided an interesting analysis looking at the uncertainty in future changes, but it did not provide spatially consistent projections due to the nature of the probabilistic projections. The use of change factors also meant that future changes in the pattern of precipitation events could not be studied.

A series of studies have used a response surface technique to identify changes in future flooding across Great Britain due to climate change (Kay et al., 2014b,c; Prudhomme et al., 2010,

2013a)). Response surfaces are generated for catchments by varying temperature, precipitation, and potential evapotranspiration timeseries, and then assessing the change in peak flows. A response surface plot can then be produced which relates change in meteorological variables to changes in flow statistics, showing the catchment sensitivity to meterological changes. A key advantage of this technique is that once response surfaces have been produced, it is easy to evaluate new climate projections without re-running models and so this approach has been described as scenario-neutral (Prudhomme et al., 2010). However, the technique assumes that changes in flood generating flows can be assessed though a combination of summary changes in meteorological inputs (e.g. Prudhomme et al. (2013a) generates flood response surfaces using change in mean annual precipitation and an index reflecting the seasonality of precipitation changes). This may not be as robust as a continuous simulation methodology, where all aspects of meteorological changes are considered simultaneously.

There is therefore a need for studies evaluating climate change on flooding nationally across Great Britain, with inclusion of modelling uncertainties. To do this, we need national modelling frameworks which are computationally efficient, to characterise climate and hydrological modelling uncertainties through ensemble simulations. We also need spatially distributed models to evaluate changes in spatial patterns of precipitation across the landscape.

2.6 Summary

A wide variety of hydrological models exist, ranging from lumped conceptual models to physically-based distributed models. These models each have their advantages and are useful for different applications. To ensure models can produce robust and reliable simulations across diverse climate/catchment characteristics, it is important to thoroughly evaluate model structures across large-samples of catchments. Multi-model intercomparisons and benchmarking studies are therefore needed to help identify aspects of model structures resulting in better/worse performance, identify reasons for poor model performance, guide model selection and inform users of model limitations. These studies can therefore guide the development of appropriate model structures for national modelling frameworks.

A key challenge for national hydrological models is representing the heterogeneity of hydrological processes across the landscape. The skill of different model structures usually varies depending on the dominant hydrological processes, and therefore it is difficult to find a single model structure which is able to produce good simulations nationally. Whilst many national modelling attempts do use a single model structure, there has been a drive towards the development of flexible modelling frameworks which facilitate the application of different model structures across a landscape. There has also been a move towards models which are flexible in

terms of spatial resolution, to reflect the dominant processes in different landscapes and best suit different modelling needs. There is therefore a need for further development of flexible modelling frameworks to improve our national modelling efforts.

Lumped hydrological models have the advantage of simplicity and computational efficiency, but many modelling applications require distributed models to represent spatial processes. These distributed models often require spatial parameter fields to reflect differences in hydrological processes across the landscape. The estimation of these distributed parameter fields is a major challenge for large-scale modelling efforts. Another challenge is modelling ungauged catchments, where runoff observations are not available for model calibration. Simultaneous parameter regionalisation approaches, such as multiscale parameter regionalisation (MPR), can help to address both issues by calibrating spatially constant coefficients of transfer functions relating parameters to spatial geophysical data. This is therefore a promising parameterisation method for national modelling. MPR has not previously been applied to Great Britain or for a flexible hydrological framework, and questions remain including which geophysical datasets to use to represent parameters of different hydrological models, which transfer functions and upscaling operators to use and how to best constrain transfer function coefficients across large catchment samples. Further development and applications of the MPR methodology are therefore needed to improve distributed model applications.

Uncertainties are present in all modelling studies, and the importance of considering and communicating these uncertainties has been widely recognised. Hydrological modelling uncertainties include choice of model structure and parameter sets, as well as errors/uncertainties in data used to drive, setup and evaluate hydrological models. Techniques to quantify modelling uncertainties can be computationally costly, and are often not applied for national scale modelling studies. We therefore need to develop tools to facilitate incorporation of modelling uncertainties in large-scale modelling studies.

One application for which we need national models is climate change impact studies. Climate change could increase flood and drought risk in the future, and we therefore need models to quantify this changing risk and inform climate change adaptation plans. To inform decision making, it is vital to include uncertainties in these projections – arising from both climate modelling and hydrological modelling. Further studies are therefore needed which model climate change impacts within an uncertainty framework.

CHAPTER 3

DATASETS, TOOLS AND METHODS

Chapter three presents the datasets, tools and methods that underpin this thesis. This supports the more specific methodologies provided in each research chapter, providing information beyond the research chapter methodologies without repeating information. It brings together the methods used in the three research chapters, highlights methodological differences (e.g. different catchment selections and model structures) and explains why different decisions were made to support the research aims of each chapter. Throughout this thesis, many different datasets were used for model setup, parameterisation and evaluation, and further details on the selection and preparation of these datasets is given here.

3.1 Catchment selection

Great Britain has a dense gauging station network, with over 1500 river gauges across the country (CEH, 2015). The locations of these gauging stations, alongside the range in a selection of catchment characteristics across Great Britain, is shown in Figure 3.1. These gauges cover a wide range of hydrological and climate diversity; with generally wetter catchments along the west of Great Britain, drier catchments in the east of England, major aquifers impacting flows in the southeast, and a few snow impacted catchments in central Scotland. A discussion of the hydrology of Great Britain is given in Research Chapter One (Section 4.3).

Different gauges were selected for the modelling studies in each research chapter. For the first research chapter, the aim was to evaluate models over as many gauges as possible, to provide a national picture of model performance. This led to the selection of 1013 gauges, which all had discharge data covering the period January 1993 to December 2008. For Research Chapter Two,

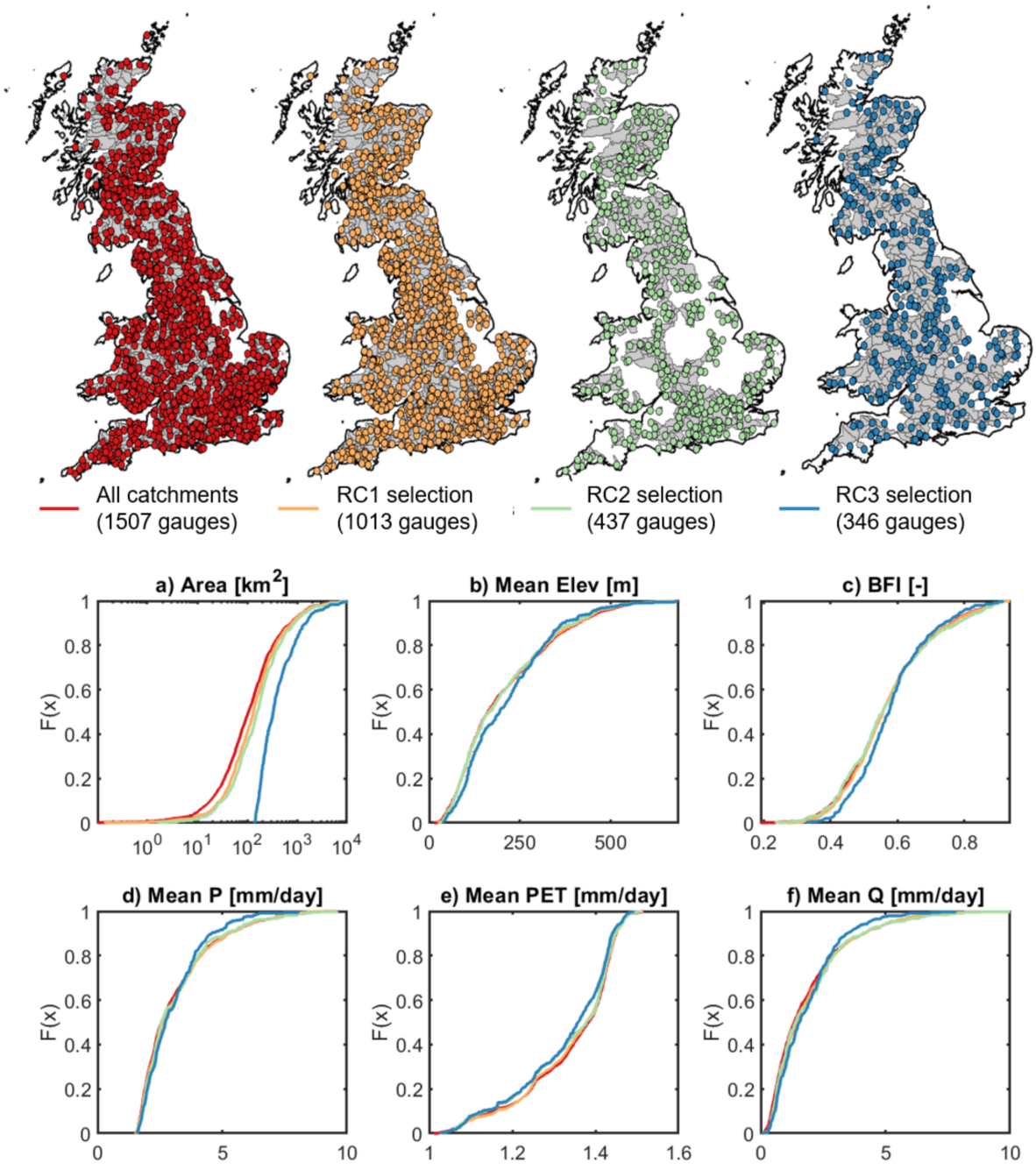


FIGURE 3.1. Comparison of gauge selection between the research chapters. Top: maps showing gauge locations (points) and catchment boundaries (grey area) for all gauges available on the NRFA, and the three research chapters. Bottom: Range in catchment characteristics covered by gauge selection in the three research chapters, compared to catchment characteristics across all British gauges.

only quality-checked Service Level Agreement gauges were included in the analysis, to avoid the chances of disinformative gauges being included in the national model calibration. In this chapter the global parameters linking model parameters to catchment attributes were constrained across all gauges simultaneously, and therefore if catchments had incorrect observed data then this would skew the evaluation of national parameter fields. We also had to discount some gauges (236), because of the computational time taken to produce a high number of simulations nationally. As can be seen in Figure 3.1 these still cover a wide range of climatic diversity across Great Britain. This reduced gauge selection to 437 gauging stations. For Research Chapter Three, the model was driven with climate model data instead of observations. The climate model data was available at 12 km resolution, compared to the 5 km resolution used in Research Chapter Two. It was considered inappropriate to use the 12 km climate information for gauges smaller than one 12 km x 12 km grid, so therefore only gauges larger than 144 km^2 were included. Using only service level agreement gauges larger than 144 km^2 resulted in 346 gauges being selected for Research Chapter Three. This was still considered a large number of gauges, sufficient to cover the range of hydrological behaviour across Great Britain, as shown in Figure 3.1.

All research chapters included both natural and human-influenced catchments. It was considered important to include human-influenced catchments for multiple reasons; 1) this thesis is focused on national-scale modelling and any national model will necessarily need to include human-influenced catchments if they are to produce flow simulations everywhere (McMillan et al., 2016), 2) there is a need for flow simulations in areas where people live and therefore it is important to have models which run in human-modified catchments, 3) there is a need to understand how models perform in both natural and human-influenced catchments to support this modelling, and 4) very few flow regimes are fully natural in Great Britain. The first research chapter evaluated model performance for as many gauges as possible across Great Britain, and the inclusion of human-modified catchments was needed to help identify factors impacting model performance. Research Chapter two aimed to constrain model parameters on a large sample of catchments. It was therefore important to select catchments which were representative of GB as a whole, which natural catchments alone would not be. Finally, research chapter three focused on river flow changes. These changes may still be meaningful even if human modifications to river flow regimes mean that simulated flows do not perfectly match observed flows.

3.2 Hydro-meteorological data and catchment attributes

The sources of hydro-meteorological data used to run and evaluate hydrological models are consistent throughout the thesis. For observed precipitation and potential evapotranspiration (PET), the CEH-GEAR and CHESS-PE datasets were used (Keller et al., 2015; Robinson et al., 2015a,b).

These are both provided at 1 km resolution across GB, but have been averaged to different spatial resolutions to support each research chapter. Research Chapter One used lumped hydrological models, so precipitation and PET were averaged across each catchment. Research chapters two and three used a model capable of using distributed inputs, and so input data was averaged over 5 km and 12 km grids respectively. The different grid sizes used was due to consistency with other data used within each research chapter; Research Chapter Two compared results with a previous study where the model had been run at 5 km, and Research Chapter Three also used climate model data which was provided at 12 km resolution.

Research Chapter Three also used hydro-meteorological data produced by regional climate models (Lowe et al., 2019). To ensure consistency, potential evapotranspiration was calculated from the climate model output using the same code as the CHESS-PE product.

The time-periods over which the models were run differed between the research chapters. For Research Chapter One the model simulations were run between Jan 1988 - Dec 2008, with the first five years being used as a warm-up period and the remaining 16 years used for model evaluation. For Research Chapter Two, a slightly longer period was selected, with model simulations run from Jan 1985 - Dec 2010. The first six years were used as a warm-up period, leaving a remaining 20 years to be divided equally into 10-year calibration/evaluation periods. For research chapter 3 the simulation period was largely determined by the climate model data. Simulations were carried out between Jan 1981 - Dec 2075 driven by climate data, and an additional set of simulations driven by observed data were run between Jan 1981 - 2010 to provide a performance baseline against which the climate-driven simulations could be compared.

All observed discharge data used within the thesis are from the National River Flow Archive (Centre for Ecology and Hydrology, 2016). Catchment attribute data used is from the UK Hydrometric register (Marsh and Hannaford, 2008), CAMELS-GB catchment attribute dataset (Coxon et al., 2020), as well as calculated from observed hydro-meteorological data and a range of national spatial datasets which are described in section 3.5.

3.3 Models

Different hydrological models have been used throughout this thesis, to best answer the questions posed by each research chapter. For research chapter 1, the focus was on exploring performance differences between model structures. Therefore, the Framework for Understanding Structural Errors (FUSE) was selected to provide four hydrological model structures that could be easily run and directly compared within the same framework (Clark et al., 2008). For research chapter 2 and 3, we required a model that could be used to explore different parameter configurations,

parameter uncertainties, and produce national simulations across Great Britain. The DECIPHeR model was well suited to this, as it has a flexible model build allowing any configuration of model parameters, it is relatively fast-running enabling the exploration of uncertainty, and it has previously shown good performance across GB (Coxon et al., 2019). Here, the structure of these models is briefly explained (full descriptions are available in each research chapter methodology), and setup and testing of the model structures beyond what is presented in the research chapters is discussed.

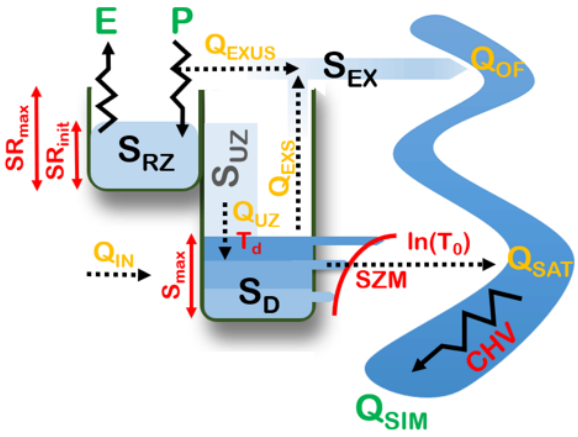
3.3.1 FUSE

The Framework for Understanding Structural Errors (FUSE) was used to provide four different hydrological model structures for use in research chapter 1. FUSE is a modelling framework that combines modelling decisions from four widely applied models to create hundreds of possible model structure configurations (Clark et al., 2008). The user can easily select different model structures, and run them with the same input data. All models within the framework share some common modelling decisions; they are all lumped models, run on a daily timestep with daily forcing data and they include the same process representations (e.g. none of the models represent snow/groundwater). However, they differ in the configuration and number of model stores, and equations governing the flow of water between stores and into the river. The framework is therefore useful as a controlled way to explore differences between model structures. No changes were made to the FUSE model code for this thesis. Model structure diagrams, modelling decisions, and model parameters are all given in Chapter 4 (Figure 4.3, Table 4.2 and Table 4.3).

3.3.2 DECIPHeR

To investigate the application of a national parameterisation method, a semi-distributed hydrological model was selected. The FUSE models were all lumped catchment models, which treated catchments as a single modelling unit with homogenous parameter values. They could therefore not represent flows across regions, and could not simulate the interrelation between sub-catchments within larger basins. A more complex model was therefore required, which could simulate variations in hydrological processes across the landscape.

The DECIPHeR hydrological model is a semi-distributed hydrological model, which is flexible in its model structure and parameter configuration (Coxon et al., 2019) (see Figure 3.2 for a summary of the default model structure). The model distributes the catchment into hydrological response units (HRUs), which are groups of points that share similar characteristics and are therefore expected to behave in a hydrologically similar way. These HRUs are defined using any spatial information the modeller thinks is important in defining hydrological variation across the landscape. An example of how this is done using slope, accumulated area, and locations of gridded rainfall inputs is given in Figure 3.3. Each HRU acts as a separate model store,



Parameters

SZM	Form of exponential decline in conductivity	m
SR_{max}	Maximum root zone storage	m
SR_{init}	Initial root zone storage	m
T_d	Unsaturated zone time delay	$ts\,m^{-1}$
CHV	Channel routing velocity	$m\,ts^{-1}$
$ln(T_0)$	Lateral saturated transmissivity	$ln(m^2\,ts^{-1})$
S_{max}	Maximum effective deficit of saturated zone	m

Stores

S_{RZ}	Root zone storage	m
S_{UZ}	Unsaturated storage	m
S_{EX}	Saturation excess storage	m
S_D	Saturated storage deficit	m

Internal fluxes

Q_{UZ}	Drainage flux	$m\,ts^{-1}$
Q_{IN}	Upslope input flow	$m\,ts^{-1}$
Q_{EXS}	Saturated excess flow	$m\,ts^{-1}$
Q_{EXUS}	Precipitation excess flow	$m\,ts^{-1}$
Q_{OF}	Overland flow (sum of Q_{EXS} and Q_{EXUS})	$m\,ts^{-1}$
Q_{SAT}	Saturated flow	$m\,ts^{-1}$

External fluxes

P	Precipitation	$m\,ts^{-1}$
E	Potential evapotranspiration	$m\,ts^{-1}$
Q_{obs}	Observed discharge (for starting value of Q_{SAT})	$m\,ts^{-1}$
Q_{sim}	Simulated discharge	$m\,ts^{-1}$

FIGURE 3.2. DECIPHeR model structure diagram showing parameters, stores and fluxes, adapted from Coxon et al. (2019).

which is hydrologically connected to spatially contiguous HRUs or river cells. The distribution of the landscape into HRUs thus defines the model resolution, with each HRU able to have its own defined model structure and parameter values. This has many advantages, including 1) minimised run times as calculations are only required for each HRU rather than each modelled grid point, 2) different model structures and parameterisations can be used to reflect landscape features, and 3) retained information on spatial distribution of HRUs allows results to be mapped back into space.

Previous applications of DECIPHeR had applied homogeneous parameter values across the landscape (Coxon et al., 2019). Therefore, whilst the HRU-based structure of DECIPHeR was ideal for exploring spatially distributed parameteters, modifications to the source code were required to make this possible. The first step was writing and modifying code to enable DECIPHeR to assign different parameter sets to different parts of the landscape. This involved: 1)

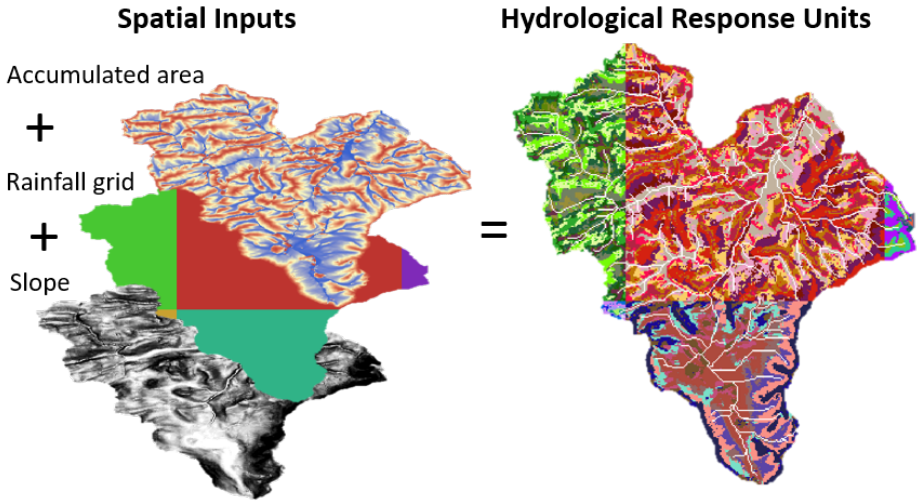


FIGURE 3.3. Graphic illustrating how hydrological response units are defined from input spatial datasets in the DECIPHeR hydrological model framework. This is for the catchment Llynfi at three cocks, which is a sub-catchment of the Severn at Haw Bridge. Input datasets are 3 equal classes of slope and accumulated area, and a 10 km rainfall input grid.

introducing a parameter field as part of the HRU classification, so that each HRU was associated with a parameter ID (see 3.3 for examples of spatial inputs used to define HRUs). This allowed the user to define areas of the catchment where different parameter sets were required, allowing any possible spatial configuration of parameter values. 2) re-defining the parameter settings file read in by the model. The previous settings file specified ranges for each parameter to be applied across the catchment. The new file was formatted to define parameter values/ranges for each of the parameter IDs defined by the user. This allowed different parameter values/ranges to be input across the catchment. 3) Ensuring that the model used these distributed parameters correctly, and applied the correct parameter values in each HRU.

A series of modelling experiments were carried out across the Severn at Haw Bridge catchment, to improve understanding of the model prior to parameterising and implementing it nationally. These focused on the model setup and input resolution (section 3.3.2.1), defining appropriate parameter ranges (section 3.3.2.2), and improving understanding of how each parameter influenced simulated flows (section 3.3.2.3). The Severn catchment was selected as it is a very large catchment (at 9895 km^2 it is the second largest catchment in the UK) which contains many sub-catchments with varying hydrological characteristics (as shown in figure 3.4).

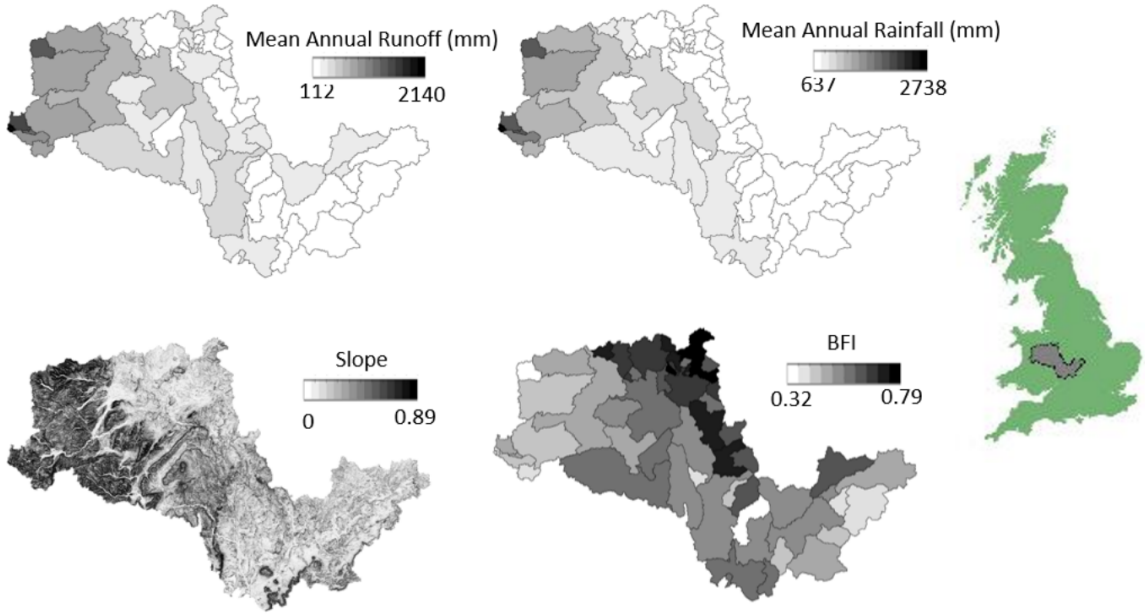


FIGURE 3.4. Spatial catchment attributes across the Severn at Haw Bridge catchment. Mean annual Runoff, mean annual Rainfall and baseflow index (BFI) data are from the UK Hydrometric Register (Marsh and Hannaford, 2008). Slope was calculated from the NEXTMap digital elevation map. The location of the Severn catchment within GB is also shown for scale.

3.3.2.1 Distributed vs homogenous rainfall inputs

The first experiment evaluated the benefit of distributing rainfall inputs. DECIPHeR was setup over the Severn catchment for two different model configurations: 1) uniform rainfall input across the catchment, 2) 10 km resolution rainfall inputs. For both of these configurations, HRUs were defined using three classes of slope and accumulated area and sub-catchment boundary maps (to prevent flow across catchment boundaries). The distributed rainfall setup also included a 10 km rainfall input grid in the HRU classifications. This resulted in 573 HRUs for the uniform rainfall input simulation, increased to 3210 HRUs for the distributed rainfall simulation. The parameters were sampled 100 times, to give an initial estimate of model performance.

The resultant model performance across the Severn catchment is given in Figure 3.5. This shows that inclusion of distributed rainfall grids either increases or does not substantially impact model performance across most of the Severn catchment. Increases in model performance are generally seen in the west, where mean annual rainfall is highest and therefore homogenous rainfall inputs would be substantially too low. The few catchments in the west which are not represented well by the model were found to have reservoirs modifying observed flows. The model structure does not currently include artificial influences, and so would not be expected to

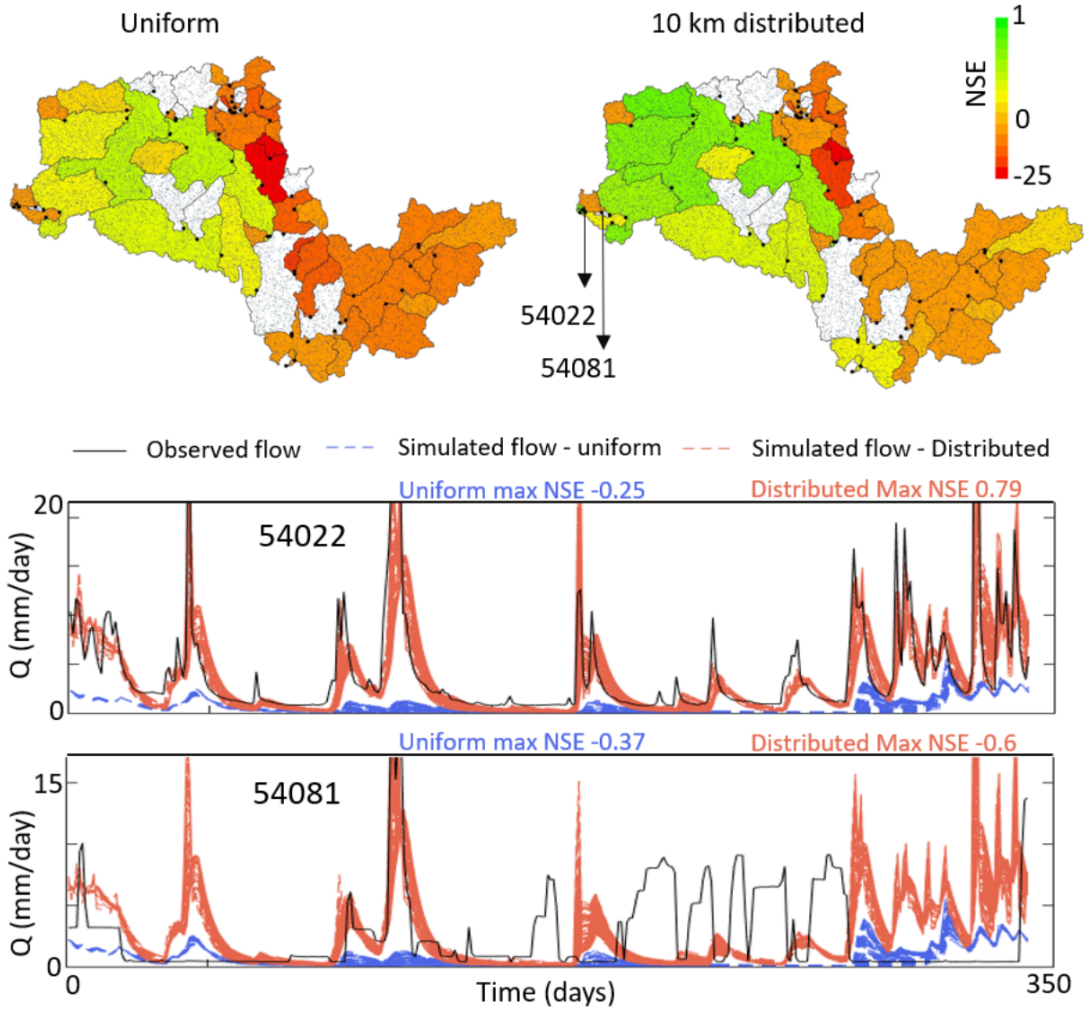


FIGURE 3.5. Initial analysis of DECIPHeR model performance across the Severn with uniform vs 10 km distributed rainfall inputs. Top: maps of performance for all sub-catchments. Sub-catchments coloured white are missing observed discharge data. Bottom: selected hydrographs showing simulated flow timeseries driven by uniform (blue) and distributed (red) rainfall inputs. These catchments were chosen to show an area where distributed rainfall leads to a large model improvement (54022), and an area where reservoirs heavily influence flows (54081) masking any performance increase.

reproduce flows for these catchments.

The model struggled to simulate catchments in the north east. These catchments have high baseflow contributions to river flow (baseflow index > 0.6), relatively shallow slopes and relatively low mean annual rainfall and runoff compared to the rest of the Severn sub-catchments, as shown in figure 3.4. The DECIPHeR model structure may not be as appropriate for these catchments, as topography is not as dominant for runoff generation. However, it may be the case that these catchments are simply harder to model, and further sampling of the parameter space is required to find behavioural model parameters. These baseflow-dominated catchments are also likely to have a longer hydrological memory, and therefore a longer warm-up period would be required to correctly simulate these catchments.

Overall, this simple experiment confirmed the importance of using distributed rainfall inputs when modelling across a large region, despite the reduction in computational speed resulting from additional HRUs.

3.3.2.2 Parameter identifiability and appropriate ranges

The second experiment explored the identifiability of model parameters, informing the selection of upper and lower bounds for each parameter. The DECIPHeR parameters are shown and defined in Figure 3.2.

DECIPHeR is designed to be run in an uncertainty framework, sampling parameters between set upper and lower bounds. The values used for these upper and lower limits is very important. Setting the bounds too wide will result in few simulations producing reasonable results, and so requiring large numbers of model simulations to effectively sample the parameter space. Setting the bounds too narrow will result in behavioural parts of the parameter space not being sampled, and so good simulations may be missed. It was therefore considered important to experiment with finding appropriate values for these upper and lower limits.

To evaluate suitable parameter bounds, DECIPHeR was run over the Severn catchment sampling the parameters over 4000 times. Initial parameter limits for these simulations were selected based on previous model applications carried out within the research group and are given alongside parameter bounds used in published literature in Table 3.1 (Beven and Freer, 2001b). Model performance was evaluated at each of the 64 gauged points within the Severn catchment using the Nash-Sutcliffe Efficiency (NSE) criterion (Nash and Sutcliffe, 1970) and a range of other performance metrics assessing performance for individual aspects of the hydrograph. Dotty plots were then produced to show the relationship between each parameter value and model performance.

TABLE 3.1. Parameter bounds used in the DECIPHeR hydrological model. Shaded rows show parameters where the bounds have been extended for this thesis. Original model setup refers to previous unpublished applications of DECIPHeR, and new model setup refers to extended bounds used throughout this thesis.

Parameter	Beven and Freer (2001a)		Original model setup		New model setup	
	Lower	Upper	Lower	Upper	Lower	Upper
SZM	0.005	0.06	0.001	0.007	0.001	0.06
$\ln(T_0)$	0.1	8	-7	1	-7	1
CHV	1000	5000	1500	6000	1500	6000
SRmax	0.005	0.3	0.02	0.15	0.02	0.3
Td	0.1	120	0.1	40	0.1	40
Smax	0.1	0.8	0.2	1.4	0.2	1.4

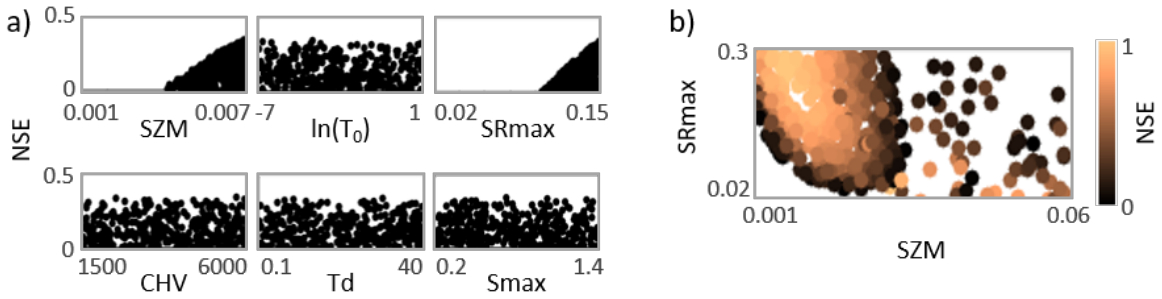


FIGURE 3.6. Dotty plots (a) and parameter interaction plots (b) used to help identify parameter limits for the Severn at Haw Bridge catchment.In the example given in a), upper bounds for SZM and SRmax have been set too low resulting in possible behavioural parts of the parameter space not being sampled.

Dotty plots map model parameter values and associated performance scores onto a one-dimensional surface, enabling a visual assessment of the identifiability of model parameters (Beven and Freer, 2001a; Pianosi et al., 2015; Wagener and Kollat, 2007). The dotty plots produced for the Severn at Haw Bridge are given in Figure 3.6, but plots for all sub-catchments and other performance metrics are not given as they show the same picture. Analysis of dotty plots identified SZM and SRmax as particularly sensitive parameters for catchments across the Severn, when analysed using NSE. SRmax and SZM also emerge as influential parameters when evaluating model performance using bias and total variance error respectively.

For many catchments, the best performing simulations were on the edge of the parameter bounds for SRmax and SZM (i.e. at the edge of the dotty plot), indicating that better model

simulations could be achieved by extending these limits. The upper and lower bounds used were subsequently extended, with values give in Table 3.1, and the model was re-run for another 4000 simulations within the larger parameter bounds. This resulted in a general increase in model performance across all sub-catchments, and therefore the new extended bounds were used for all future simulations.

Dotty plots are a good method to quickly visually identify parameters with incorrect limits and parameters which are exerting the strongest control on model performance. However, there are limitations to their use. Firstly, they consider each parameter individually and do not consider parameter interactions. It is the performance of a parameter set that is evaluated by NSE, and not the performance of an individual parameter value, and therefore care should be taken not to over-interpret these results. Secondly, they are conditional on the performance measure selected (Freer et al., 1996), and some parameters may show a stronger response to different performance metrics (e.g. a parameter controlling model timing may not show any sensitivity if model performance is evaluated using a bias metric). We therefore continued this analysis with a more in-depth look at how each parameter impacts various aspects of the hydrograph.

3.3.2.3 Parameter impact on simulated flows

To further understand the sensitivity of the DECIPHeR model parameters, and to understand the effect of each parameter on simulated flows, a small experiment was carried out sampling each model parameter whilst keeping the others constant. For each model parameter, 80 simulations were carried out sampling between the set ranges for the selected parameter value whilst keeping the others set at default values. This enabled a visual assessment of parameter sensitivity across the Severn catchment.

The results are given in Figure 3.7. Hydrographs are given for the Severn at Haw Bridge (54057, 9895 km^2), and the Severn at Abermule which is a smaller sub-catchment at the western edge of the Severn catchment (54014, 580 km^2). Hydrographs are coloured by the relative parameter value, ranging from cyan at the lower parameter bound up to magenta at the higher end of the parameter bound. Summary statistics are given next to the hydrographs, showing the variance, total flow and peak flow relative to observations which are given as a dashed black line.

These results show that $\ln(T_0)$, SZM, SRmax and Smax all have a large impact on simulated flows, whilst CHV only has a small impact. SRmax and Smax, which control the root zone and saturated zone store volumes, are important in controlling total flow volumes (i.e. bias). SZM and $\ln(T_0)$, which define the soil hydraulic conductivity, are important in defining the shape and timing of peak flows, as well as the overall variance. SZM, which has previously been highlighted as a very sensitive parameter (see Figure 3.6), is particularly important in defining the shape

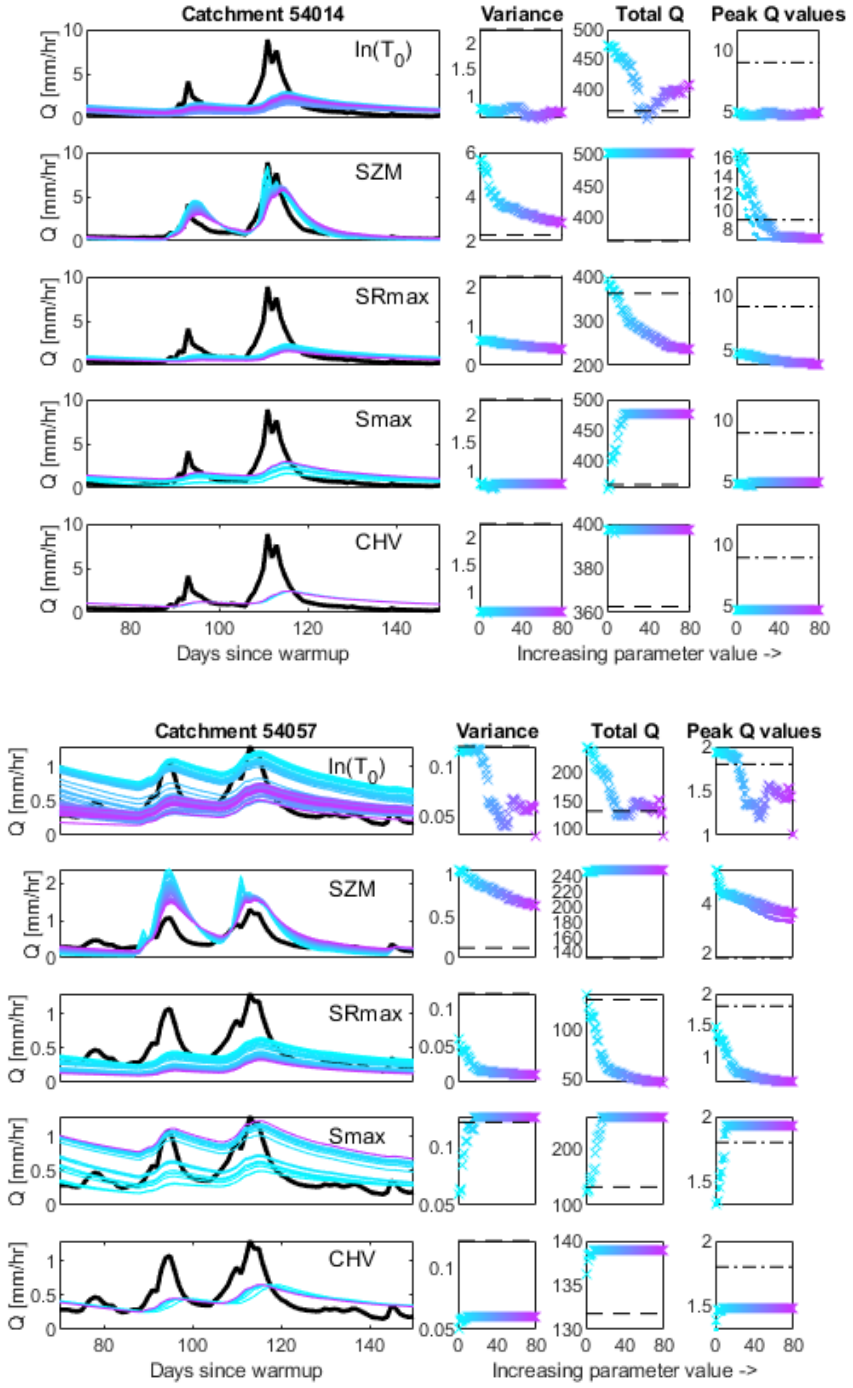


FIGURE 3.7. Demonstrating parameter influence on the hydrograph through one-at-a-time parameter sampling for the Severn at Abermule (54014) and Severn at Haw Bridge (54057). Each row shows the result of varying one model parameter while holding all others constant. Hydrographs are coloured by the relative value of the changing parameter, ranging from low (cyan) to high (purple). Additional plots summarise the change in overall variance, total flow and peak flow as the parameter values are increased.

and timing of peak flows. CHV, the channel routing velocity, impacts flow timing, but it appears to be relatively insensitive, especially for the smaller catchment.

A major limitation with this style of analysis is that it does not enable assessment of parameter interactions. Interactions between the parameters will be important, and can result in changes to the hydrographs not seen here. For example, the impact of the SZM parameter may vary depending on the chosen default value for $\ln(T_0)$. Therefore, this analysis is useful as a general picture of how parameters values control model output, and the impact of each individual parameter when considered in conjunction with other analysis, but these results should not be over interpreted.

3.4 Model parameterisation

As discussed in Section 2.3 and Section 5, Multiscale Parameter Regionalisation (MPR) emerged as the best parameterisation strategy for a flexible, large-scale hydrological model. Through linking parameters to spatial geophysical datasets, this method had the potential to create seamless spatial fields of model parameters, and produce consistent parameter fields across a domain without the need to calibrate every catchment and consider interactions between interacting subcatchments. Furthermore, as MPR linked parameters to geophysical data at the highest possible resolution before upscaling, previous studies indicated that it could produce global parameters that were appropriate across different model resolutions (Kumar et al., 2013a; Samaniego et al., 2017). This is particularly desirable for a flexible model such as DECIPHeR, which allows the user to split the landscape in multiple different ways and test different configurations of HRUs. It would mean that the same set of constrained global parameters could be applied to produce simulations across any of these spatial configurations without the need to re-calibrate the model each time.

In order to apply the MPR technique to DECIPHeR, a large amount of code development was required. The DECIPHeR hydrological modelling framework was written in Fortran. It was decided to create a DECIPHeR_MPR model version, including the parameter regionalisation and DECIPHeR model together. This meant that large numbers of model parameter settings files did not have to be created and saved. Instead, the MPR code could create thousands of parameter fields and then directly run the DECIPHeR model with these parameter settings. Fortran modules were subsequently written to carry out each step of the MPR process within the DECIPHeR framework. A summary of the code flow is given in Figure 3.8, and code is provided in Appendix C. The most important modules are described below.

Read_controls. This module read in all the settings files for MPR. All the key user decisions

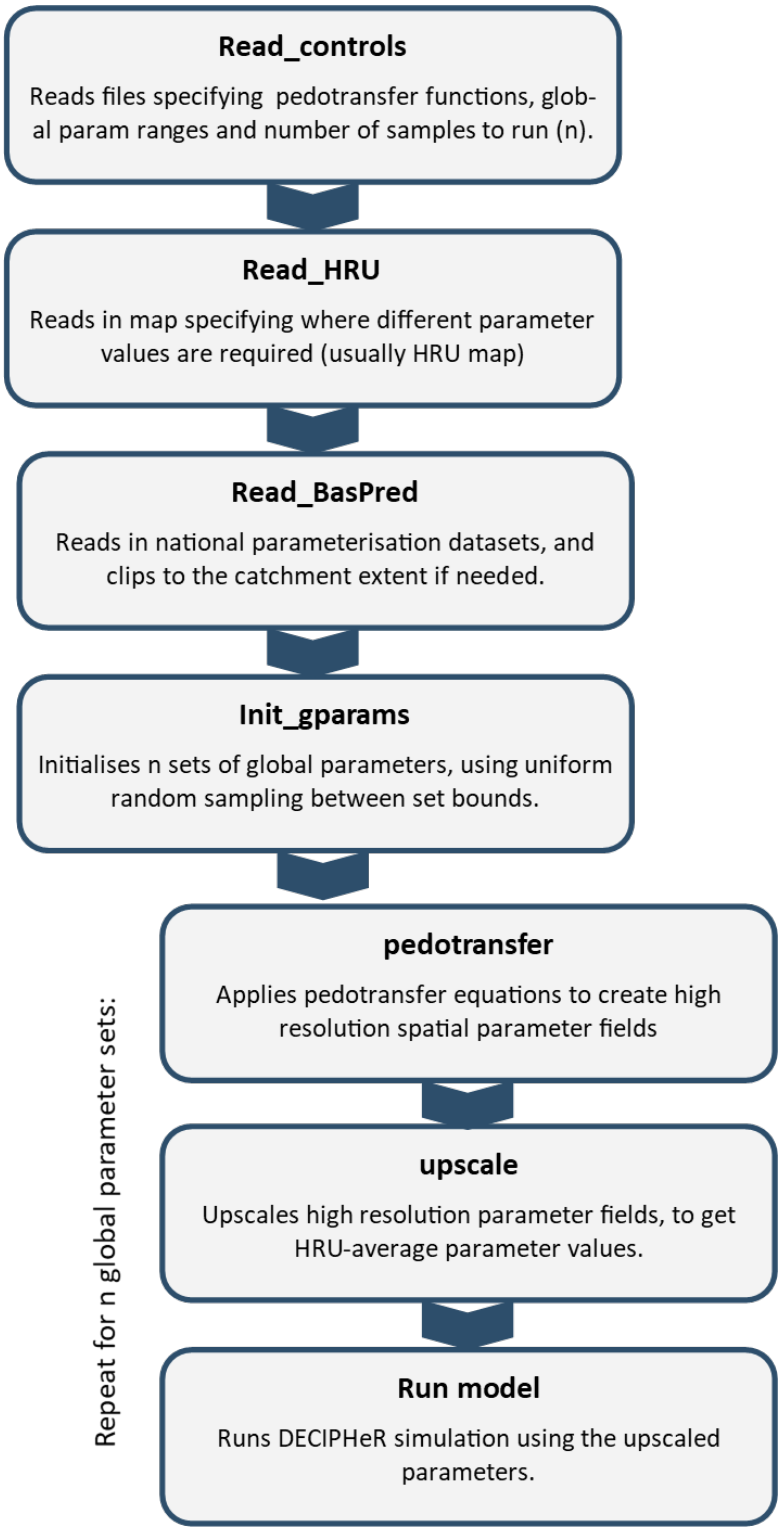


FIGURE 3.8. Diagram showing the flow of the DECIPHeR_MPR code and key modules.

for the MPR process were kept in readable settings files, so that it was easy to experiment with and adapt the framework. The main settings file contained options including the number of global parameters to sample, and the parameterisation technique desired for each parameter (including options to apply homogeneous sampled parameter values across the catchment as well as different possible pedotransfer functions to use for regionalisation). The second settings file lists the filenames of all the catchment descriptor datasets to use in the regionalisation.

Read_HRU and Read_BasPred. These subroutines read in the spatial data required for the parameter regionalisation. In the first step, read_HRU read in the map specifying where different parameter values were required across the landscape. This was normally the HRU map, but could also be a map of sub-catchments or any other spatial map that was used in the creation of HRUs. In the second step, read_BasPred was called for each model parameter to read in all the datasets required for parameterisation (basin predictor datasets). This referred to the parameterisation options specified in the settings file, so that time and computer memory were not wasted reading in datasets that would not be used. Once datasets had been read-in, they were then clipped to the model domain size to prevent unnecessary processing of large datasets. If required, these clipped datasets could then be saved as ascii files to speed-up future model runs.

init_gparams. This subroutine randomly initialises the global parameter values between set bounds. Random numbers are created using a set starting seed, so that each call of the modelling framework generates the same list of global parameter values (i.e. global parameter number 1 will always have the same value whenever the model is called). This is useful, as it means that even if the model is run separately for different catchments, the global parameters creating the parameter maps are consistent. An option is given in the user settings file to start on a specific global parameter number so that large samples of global parameters can be carried out in parallel.

pedotransfer. This subroutine applies the selected pedotransfer function for each parameter, to create high resolution parameter fields. An example of this process is shown in Figure 3.9. This example shows how the high-resolution basin predictor datasets are input into the pedotransfer function. Varying the global parameter values (a, b and c in the figure) results in different high resolution parameter fields being produced on each iteration.

upscale. This suproutine applies upscaling operators to the high resolution parameter fields, to create a list of parameter values which can be used by the model. A visual demonstration of this is given in Figure 3.10. The resolution that parameters are upscaled to is defined by the parameter id map provided in the read_hru module, and so is completely flexible. A variety of different upscaling operators are available, including the harmonic mean, arithmetic mean and majority. The most appropriate upscaling operator to use for each model parameter was found

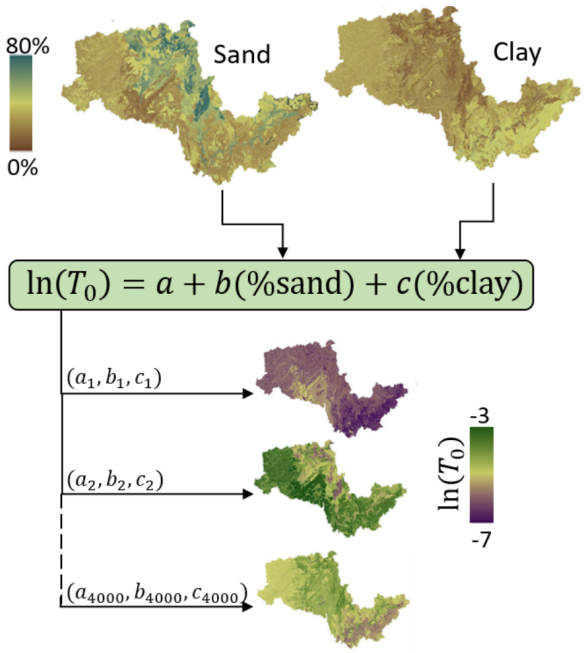


FIGURE 3.9. Graphic demonstrating how an ensemble of possible parameter fields are created within the multiscale parameter regionalisation technique, using the example of the ln(T_0) parameter across the Severn at Haw Bridge catchment. Catchment attribute datasets (here %sand and %clay) are linked to the model parameter through a transfer function. Different global parameter values (here a, b and c) result in many different possible high resolution parameter maps.

through investigation of upscaling operators used in previous studies and testing of resultant parameter fields.

Run_model. This module calls a slightly modified version of the DECIPHeR model structure. The original DECIPHeR model includes modules initialising reading in parameter files and sampling parameter values which have been removed. The initialisation scheme was also modified to remove bugs resulting from distributed parameter fields.

3.5 Datasets for model parameterisation

The need for high-resolution geophysical data linking to each model parameter makes MPR a data-intensive parameterisation approach. Finding high-resolution data is challenging, and there are many issues with available datasets even for a data-rich country such as the UK. This includes data not being freely available, missing data and gaps in national products, and the difficulty of mapping data that is derived from point-source measurements. The following section details the selection and processing of geophysical datasets to enable a national-scale

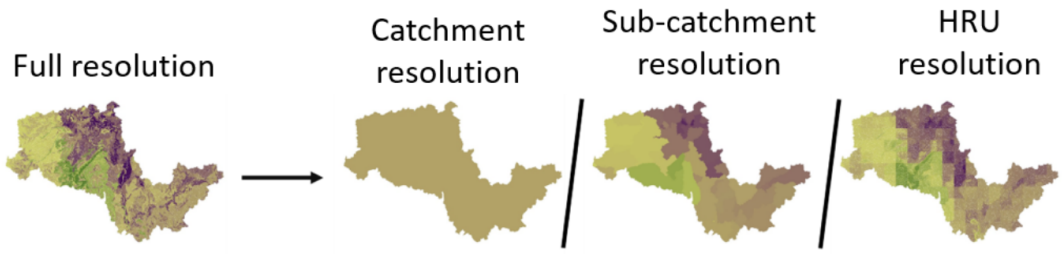


FIGURE 3.10. Graphic demonstrating parameter upscaling for an example $\ln(T_0)$ parameter field across the Severn catchment. The high-resolution parameter field can be upscaled to any desired model resolution, including lumped basin parameter fields (catchment resolution), parameters for each sub-catchment (sub-catchment resolution) or parameters for each HRU (HRU resolution).

application of MPR across Great Britain, to expand upon the brief description of the datasets given in Research Chapter Two. A summary of all the datasets is given in Table 3.2, alongside a link to the section where they are fully explained. These data underpin the modelling carried out in research chapters two and three.

3.5.1 Land-use and rooting depth

A map of rooting depth was required for parameterisation of the SRmax parameter. This parameter controls the maximum root zone storage in the model, which is the main store from which evapotranspiration is taken, and therefore theoretically relates to rooting depth and porosity (required to transform water depth to deficit). This section describes the selection of datasets for rooting depth, whilst porosity is explained in section 3.5.4. There were no nationally available root depth maps for the UK, so instead we decided to use land-use information combined with research on the expected rooting depths associated with different land-uses. To include the uncertainty in this link, we sampled between a range of possible rooting depths for each land-use within the parameterisation framework.

A literature review was carried out to identify the range in rooting depths associated with different land uses. Potential rooting depth data was available from studies based on observations of rooting depth (Fan et al., 2017; Schenk and Jackson, 2002; Zeng, 2001), studies modelling rooting depth (Kleidon, 2004; Wang-Erlandsson et al., 2016; Yang et al., 2016), and look-up tables of rooting depths used by other hydrological models (Schmied et al., 2014; Wang-Erlandsson et al., 2014). Global rooting depths datasets based on the observation approach used available measurements of root depth profiles/ root depth distributions and extrapolated these to produce global products based on the relationships between rooting depth and global biomes (Schenk and Jackson, 2002; Zeng, 2001). Modelling approaches used satellite measurements such as

TABLE 3.2. A summary of the spatial datasets collated for model parameterisation, with links to the section where the generation of each dataset is fully explained.

Dataset	Description	Section
Land-use	A simplified land-use map was created by reclassifying the UKCEH Land Cover Map.	3.5.1
Rooting depth	A table of plant rooting depths associated with different land-uses was created by collating values from the literature.	3.5.1
Sand, silt and clay content	Maps of surface soil percentage sand, silt and clay, alongside sand, silt and clay fractions at different soil depths, were generated from national soil products (LandIS nationalsoils map for England andWales and James Hutton Institute soils map for Scotland).	3.5.2
Organic carbon content	Maps of Organic Carbon content were extracted from the national soil products.	3.5.2
Bulk density	Maps of bulk density were produced based on the national soil products using pedotransfer functions.	3.5.3
Porosity	Maps of porosity were produced using the relationship between porosity and bulk density.	3.5.4
Soil depth	A soil depth map was produced by combining a) high spatial resolution soil depth information extracted from the national soil products for soils shallower than 1m, with b) modelled soil depth information from Pelletier et al. (2016) extending to a much greater depth of 50m.	3.5.5
Hydrogeology	A map identifying areas of high productivity hydrogeology was created from the CAMELS-GB dataset.	3.5.6

absorbed phosynthetically active radiation (APAR), precipitation or evaporation as inputs, which were then transformed to rooting depth estimates given different assumptions (Kleidon, 2004; Wang-Erlandsson et al., 2016; Yang et al., 2016). The modelling study of Yang et al. (2016) was particularly relevant, as they focused on effective rooting depths for hydrological applications and compared their results to other studies. The effective rooting depth accounted for both vertical rooting depth and the spacing between plants, rather than the actual root depth of an individual plant. This was estimated by applying an analytical rooting depth model which balances carbon cost with benefits of deep roots, to get a global estimate of rooting depths. From this, they produced plots of effective rooting depth for each global biome with error bars indicating their modelling uncertainty, and presented this against results using other methods. Wang-Erlandsson et al. (2016) also modelled global rooting depth for different biomes, reporting uncertainties in these results within biomes and between different methods, and tested these in a global hydrological model. We collated information from the above studies to produce a table of rooting depth ranges

TABLE 3.3. Rooting depth values associated with each land-use; upper and lower bounds. These were synthesised from a selection of studies (Kleidon, 2004; Wang-Erlandsson et al., 2014, 2016; Yang et al., 2016)

Land-use	Lower rooting depth (cm)	Upper rooting depth (cm)
Broadleaved woodland	150	250
Coniferous woodland	150	250
Cropland	75	175
Mixed grassland	75	175
Fen/Marsh/Swamp/Bog	75	175
Heather	25	225
Rock	5	150
Water	0	300
Coastal sediment	5	150
Urbanised	5	100

for different land cover types.

Multiple land cover maps were available, each using different land-cover classes (see figure 3.11. The Land Cover Map 2015 (LCM2015) produced by the Centre for Ecology and Hydrology was the highest resolution land-use map available (Rowland et al., 2017). This is a national product, derived from satellite images and digital cartography, freely available for research at 25m resolution across the UK. The high resolution and coverage of this product made it the ideal choice for our regionalisation method. However, the land cover classes available did not directly relate to the land cover classes given in the rooting depth literature. The large number of land-use classes (20 classes in total) was also a drawback, as each class introduced an additional global parameter into the parameter regionalisation. Having too many global parameters would increase the difficulty in constraining the global parameter values (as additional model runs would be required) and could result in overparameterisation and the non-identifiability of parameter sets.

The land cover map which related most closely to the rooting depth literature was the MODIS global land cover product (Friedl et al., 2010). This was available at a 5 arc minute resolution, as shown in figure 3.11, with 17 land cover classes (although not all of these classes were used within Great Britain, for example Savannas, Snow and ice). However, the low resolution of this product was unsuitable for our use.

To resolve the issues of (1) disparities between land cover classes in the LCM2015 product and research on rooting depth, (2) too many classes in the LCM2015 product, and (3) the MODIS land cover map being too low resolution, a new land cover map was created. We re-classified the LCM2015 map, reducing the 20 classes down to 10 that were considered most hydrologically

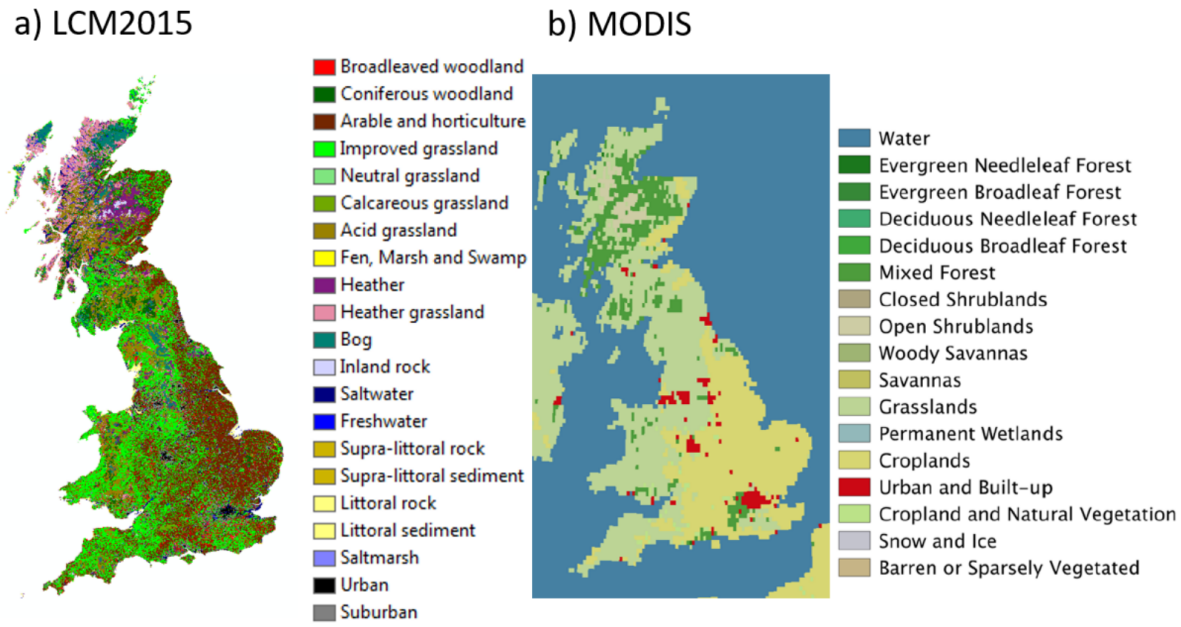


FIGURE 3.11. A comparison of land-use products available for Great Britain. a) the 2015 Land Cover Map produced by UKCEH, (Rowland et al., 2017), b) MODIS global land cover product (Friedl et al., 2010).

relevant and closely related to the biomes reported in the rooting depth literature. The resultant 50m resolution raster map is shown in figure 3.12.

3.5.2 Soil texture and organic content

Soil texture and organic content data were required for parameters relating to saturated hydraulic conductivity. Surface soil properties were required for the $\ln(T_0)$ parameter, while the distribution of these properties with soil depth was required for SZM.

The LandIS National Soil Map of England and Wales (NATMAP) and the James Hutton Institute Soils Map of Scotland were used as these provided the highest possible resolution soil data for Great Britain. A key downside to using the LandIS soil map is that it is not freely available, unlike other products covering larger scales such as the European Soil Database Derived Data product (Hiederer, 2013a,b) which provides the same soil attributes on a 1 km grid. As MPR relies on having good quality, high-resolution data, this was a necessary compromise.

The national soil data were not given as spatial products, and some simplifying assumptions had to be made to produce spatial maps of soil attributes. The datasets provided soil information for each land-use within a soil-series, whereas the vector maps provided with the datasets

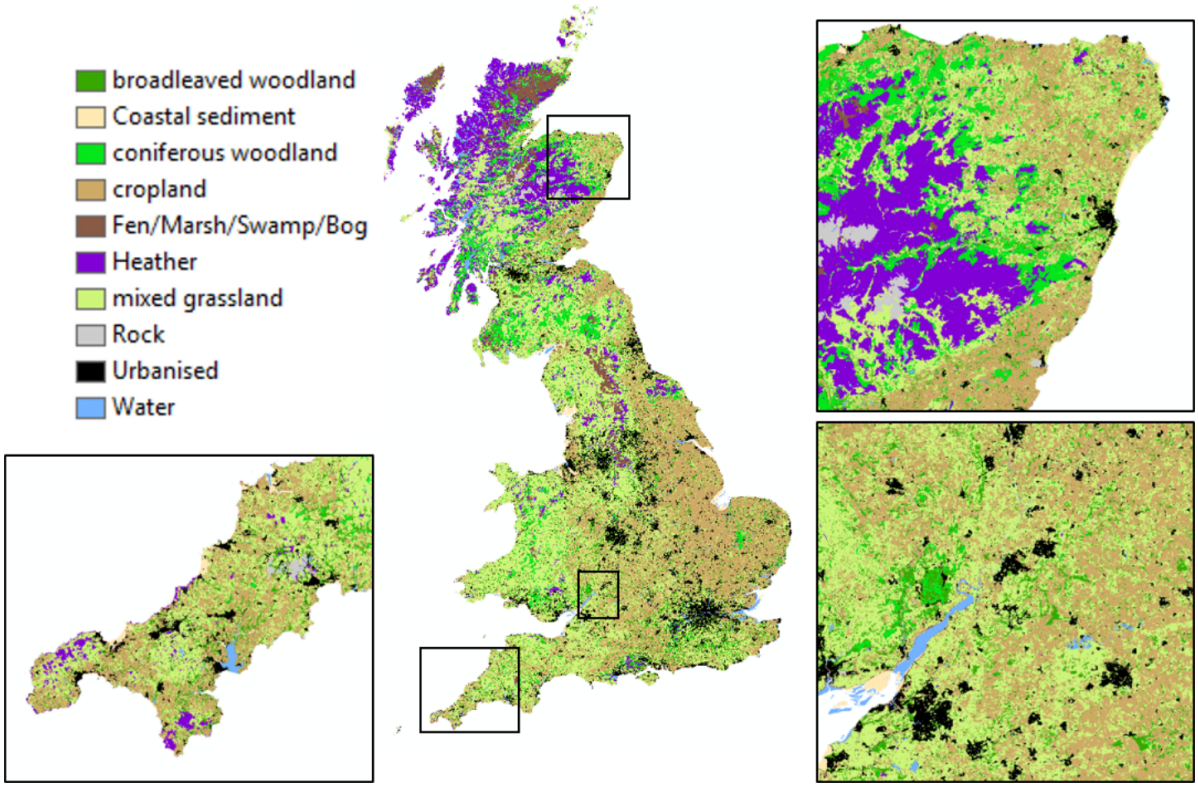


FIGURE 3.12. The modified land-use product used for model parameterisation. This was based on the LCM2015, but with a reduced number of land-cover classes.

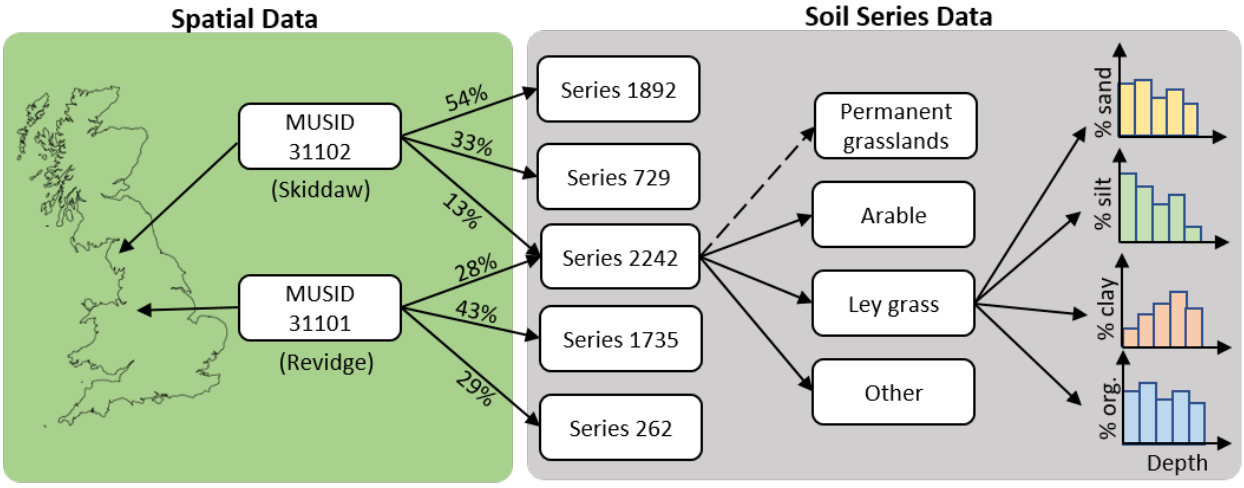


FIGURE 3.13. Illustration of the layout of the national soils data products. Shows the complicated one-to-many relationship between mappable units (MUSIDs) and the soil series data.

consisted of map-units which often included many different soil series and no indication of the distribution of land-uses within soil series'. A graphical representation of this relationship is given in Figure 3.13. Therefore, to extract and plot only surface soil properties from these datasets, using a methodology that was as parsimonious as possible and did not bring in other datasets with their own uncertainties, the following analysis steps were taken:

1. For surface soil properties, all soil series observations with an upper depth of 0 (i.e. the soil surface) were extracted from the database. For depth profiles of soil properties, observations falling within the depth categories of 0-10cm, 11-25cm, 26-50cm, 51-100cm and 101-150cm were extracted individually.
2. Series-average properties were calculated by taking a mean of values for all different land-uses within that soil series. All land-uses were given equal weighting.
3. It was not possible to plot the soil series-average properties directly, as we did not have any information about the location of these series within the map-unit, only the fraction of the map-unit area covered by each soil series. Therefore, map-unit average values were calculated by taking an area-weighted mean of all series-average values falling within each map-unit.

Following these data analysis steps, we were able to produce maps of soil percentage sand, percentage silt, percentage clay and organic content at the surface as shown in Figure 3.14. The original vector datasets were converted to 50m raster, which were compatible with the DECIPHeR input grids, as this was considered high enough resolution to capture the features of interest. The two data products were then combined, to produce national maps for Great Britain.

Whilst the surface textural properties maps covered most of Great Britain, there were large areas where no data was available. The categories assigned to these areas of missing data are shown in 3.15. The main reason for missing data was the presence of peaty or highly organic soils, for which soil textural properties are not routinely measured. This was particularly a problem for the Scottish data, where large areas were missing. This missing information was important, as peaty and organic soils were likely to have very different properties than their neighbouring soils. Therefore, it was clear that the soil textural information could not be used as a catchment predictor variable for peaty and organic soils, and the missing information could not be covered up using a gap-filling approach. An alternative pedotransfer function would need to be derived for these areas of missing data, using different catchment descriptor data. Most other missing data was due to scattered cells of surface water and areas of bare rock, with a small area of china clay spoil workings in Cornwall. These areas of missing data needed to be filled in, to make the maps usable within the national scale methodology.

It was decided that the best approach for the missing data was a nearest-neighbour gap-filling method, whilst keeping a record of the areas of peaty/organic soil to ensure that the soil textural

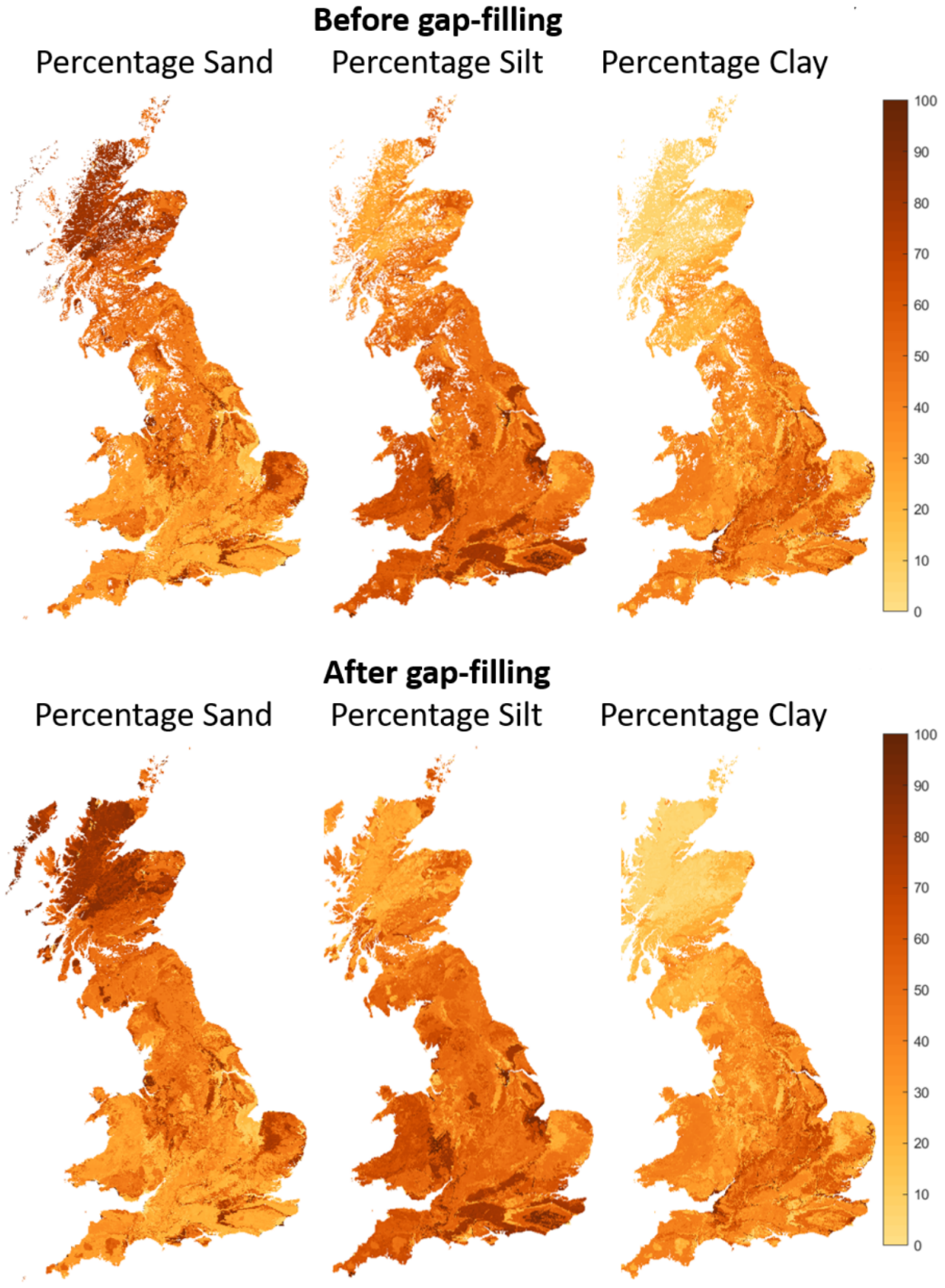


FIGURE 3.14. The maps of surface soil percentage sand, silt and clay derived from the LandIS National soil map of England and Wales combined with the James Hutton Institute soil map of Scotland. These have been converted into a 50m raster format. Top row shows the raw maps before gap-filling, bottom row shows the gap-filled maps.

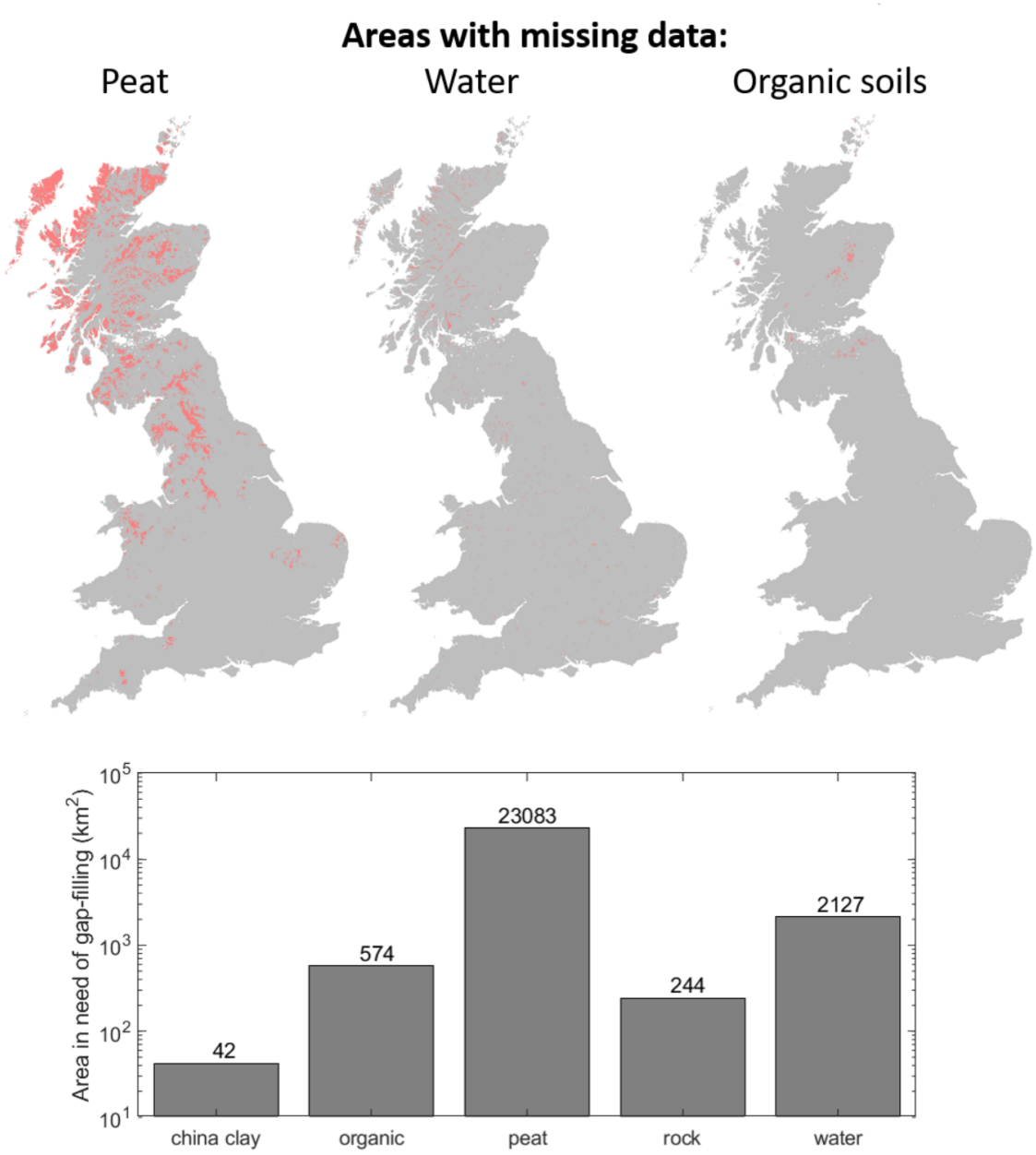


FIGURE 3.15. Missing soil texture information in the LandIS and James Hutton Institute soil products, and possible causes. Maps show location of missing data in red for cells identified as peat (organic soils with peat in the soil description), water cells or organic soils (have an organic matter content above 30% and do not mention peat in the soil description). The bar plot shows classifications for other areas of missing data: China clay spoil workings and bare rock.

information was not used across these areas. A gap-filling algorithm was written in MATLAB. This algorithm iterated through the maps. On each iteration, it looped through the cells still missing data, extracted the surrounding cells using a Moore neighbourhood (i.e. including diagonally touching cells as neighbours), and took a mean value of all neighbouring cells with data to fill in the missing data cell. The iterative nature of the algorithm meant that the order in which cells were evaluated was not important, as gap-filled values would not be included in the map until the next iteration. When the algorithm had converged, and each successive iteration was not able to fill any of the nodata cells, the extent of the neighbourhood could be increased. The result of this gap-filling can be seen in Figure 3.14.

The organic matter map had fewer missing data cells than the textural properties map, but it was still run through the gap-filling algorithm to prevent any missing-data problems when running through the national model setup. The surface soil organic content is much higher for Scottish soils than for England and Wales, but overall the merging of the two datasets appears consistent.

3.5.3 Bulk density

Soil surface dry bulk density maps were required for the parameterisation of parameters relating to k_{sat} in areas of peaty soil. Bulk density data were not available within both the LandIS and Hutton national soil maps, and there were no suitable observational datasets of bulk density across Great Britain. Whilst a bulk density map was available from the Countryside Survey (Countryside Survey, 2007), it was only available for topsoil and contained many areas of missing data. It was therefore decided to use a pedotransfer function to generate bulk density maps, and bulk density for various depths within the soil profile, from our available soil data.

Two different pedotransfer functions were found for generating bulk density from organic content. Firstly, Lilly (2018) have previously applied the following equation to predict dry bulk density (D_b) based on Scottish soils:

$$D_b = 0.6653 - 0.01C, \tag{3.1}$$

where C is percentage organic carbon. Secondly, Hollis et al. (2012) developed a pedotransfer function for UK and pan-European soils, with the form

$$D_b = 1.4903 - 0.33293\ln(C), \tag{3.2}$$

where C is also percentage organic carbon content. Without any evidence to suggest one of these equations was more suitable than the other, we decided to apply both to check if the choice of equation led to substantial differences in the resultant bulk density maps.

Figure 3.16 shows the bulk density maps produced by both pedotransfer functions, and how the results compare. The choice of pedotransfer function was important for areas with high/low bulk density, but did not have a large impact on areas where bulk density was estimated to be between 0-0.5 g cm^{-}3. Both pedotransfer functions resulted in maps showing the same patterns of bulk density across GB, which matched the broad patterns seen in the Countryside Survey bulk density product Countryside Survey (2007). We were most interested in the relative differences in bulk density across GB, and capturing the areas where bulk density was expected to be particularly low/high. We therefore decided to use the data derived using the Hollis et al. (2015) pedotransfer function, which most clearly highlights these relative differences.

3.5.4 Porosity

Porosity maps were calculated using the inverse relationship between porosity and bulk density. Porosity can be defined as

$$P = (1 - \frac{D_b}{D_p}) \times 100, \tag{3.3}$$

where P is total porosity, D_b is bulk density, and D_p is particle density (Hollis et al., 2015). A standard value of 2.65 g cm^{-3} was used for D_p, as spatial measurements were not available. This is consistent with techniques used in previous UK soil data studies (Hollis et al., 2015).

3.5.5 Soil depth

The Smax parameter defines the maximum effective deficit of the saturated zone. This should relate to the soil depth and porosity - with deeper and more porous soils theoretically being able to store more water. Soil depth maps were therefore required. Several different datasets were available for soil depth estimates across Great Britain, each with their own advantages and problems.

Soils depths datasets covering Great Britain were available from the UK Soil Observatory (UKSO) and European Soils Map. These products reported soils depth categories, rather than measured soil depth values, so the information that could be gained was limited. The European soils map was considered too coarse, as it only included two classes for deep or shallow soils. The UKSO product was more promising, with a 1 km resolution map, splitting soil depth into 5 classes. However, even this was considered relatively coarse for MPR, which requires catchment descriptor information at the highest resolution possible. Therefore, we continued looking for products with a higher spatial resolution both at the surface and with depth.

The LandIS National soils map for England and Wales and James Hutton Institute soils map for Scotland are the highest resolution soils maps available for Great Britain. These datasets do not include a maximum soil depth or depth to bedrock variable. However, both included depth ranges for their soil measurements, and recorded when rock layers were reached. From this,

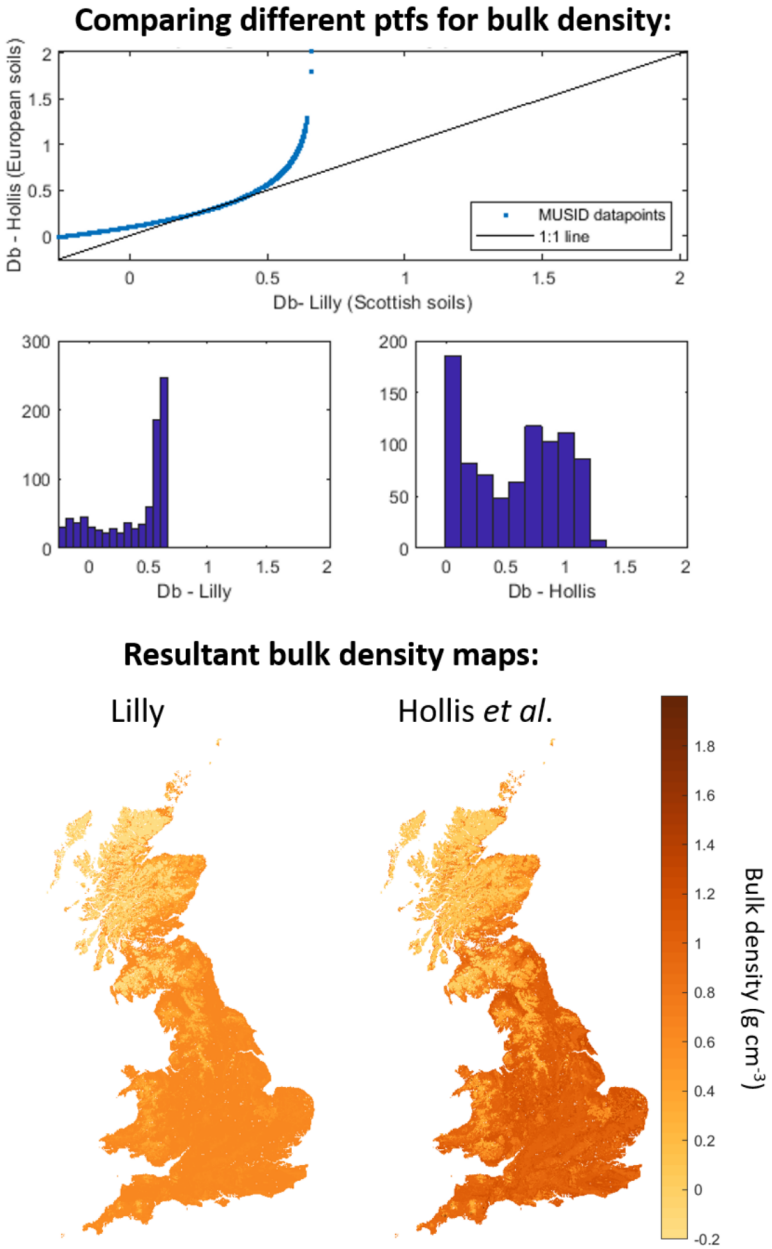


FIGURE 3.16. A comparison of pedotransfer functions to generate bulk density maps across Great Britain. Top: Scatter plot directly comparing bulk density estimates produced by the pedotransfer functions of Lilly (2018) and Hollis et al. (2012), at all mapped points in Great Britain. Points should fall along the 1:1 line where both pedotransfer functions produce the same bulk density estimates. Middle: histograms showing the distribution of bulk density values across Great Britain when using the pedotransfer function of Lilly (left) or Hollis (right). Bottom: surface bulk density maps using the pedotransfer function of Lilly (left) vs Hollis (right).

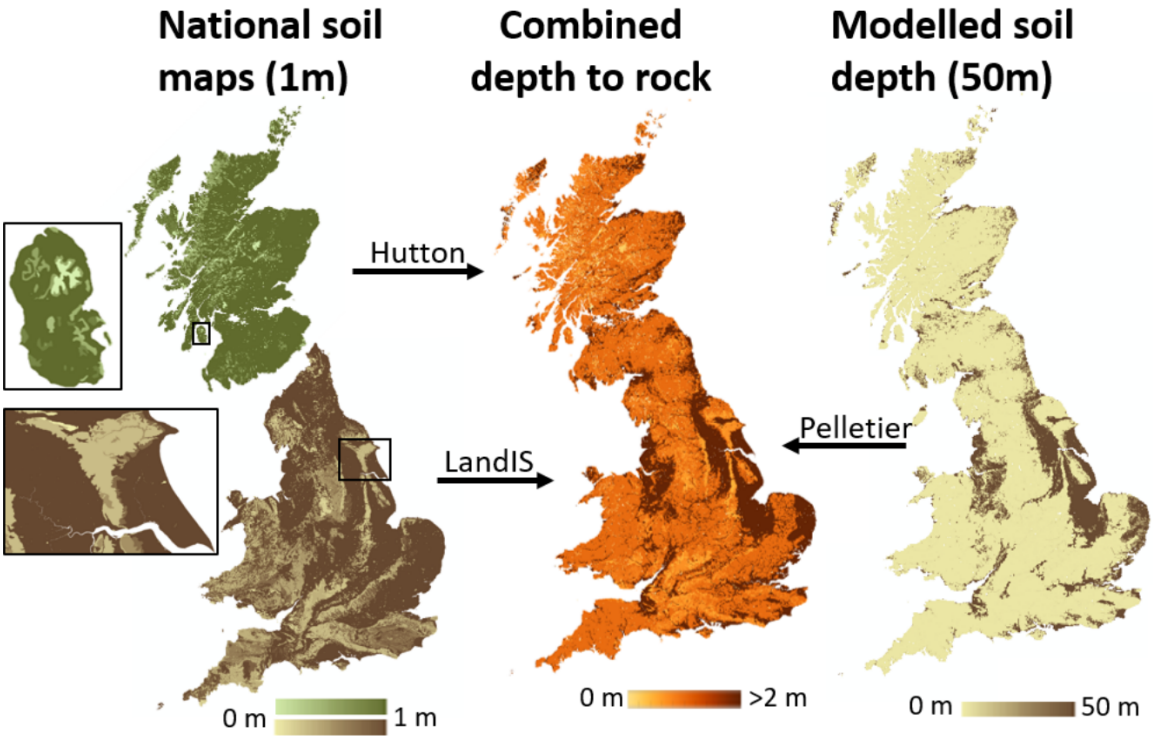


FIGURE 3.17. Generation of the soil depth dataset produced by combining high spatial resolution national soils depth information with modelled soil depth data extending to a greater depth.

maximum soil depths maps were produced for Great Britain at a very high spatial resolution, as shown in Figure 3.17. However, the LandIS and James Hutton Institute products only sampled to 150cm and 100cm respectively, and large areas of Great Britain have soil depths extending much below this. Therefore, they did not adequately show variations in very deep soils.

As few measured soil depths datasets sampled deep enough to distinguish between the large areas of 'deep' soils in Great Britain, and we felt this information was important for the catchment hydrology, we looked at modelled estimates of soil thickness. Pelletier and Rasmussen (2009) produced a global dataset estimating soil thickness using the best available data for topography, climate and geology. This modelled dataset is at a global 30-arcsecond resolution, which is of a comparable resolution to the UKSO soil depth information yet less high resolution than the national soils maps, and extends to a soil depth of 50m. This is far deeper than any of the soil depth measurement datasets, and can provide an indication of soil depth variation in the 'deep' soils where measurements are lacking.

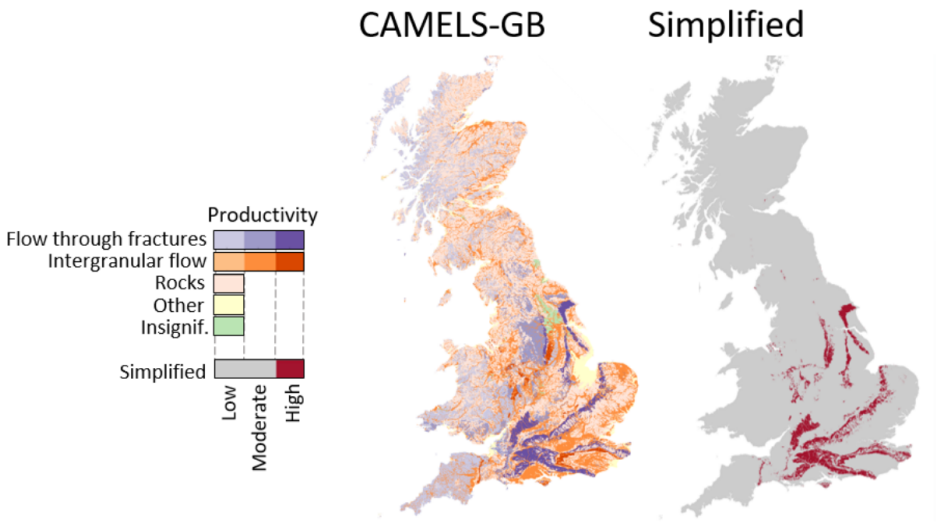


FIGURE 3.18. Simplification of the CAMELS-GB hydrogeology, to create a map identifying areas of high productivity hydrogeology. Darker areas show regions with higher productivity hydrogeology, and the different colours indicate the flow paths.

Therefore, we created a new soil depth map for Great Britain that merged information from the national soil maps and modelled soil depths, as shown in Figure 3.17. For all soils where bedrock had been recorded in the national soils database, the depth at which bedrock was recorded was used. For all soils where the national soils database had not found bedrock (i.e. these soils are deeper than was sampled) modelled soil depth from Pelletier and Rasmussen (2009) was used. This allowed us to create a product that combined the high spatial resolution of the national soils products with the extra information for very deep soils brought in from the modelled data.

3.5.6 Hydrogeology

Storage and transmission of water within a catchment is related to catchment geology as well as soil types (Pfister et al., 2017). Initial applications of MPR without considering hydrogeology showed poor performance in areas with high productivity geology, due to underestimation of saturated zone storage (Smax parameter) and lateral saturated transmissivity ($\ln(T_0)$ and SZM parameters). It was therefore considered important to add hydrogeology into the regionalisation.

A national hydrogeology map was generated from hydrogeology data provided within the CAMELS-GB dataset (Coxon et al., 2020). This categorises the uppermost geological layer (superficial deposits where present, or bedrock where not) into nine hydro- geological classes based on aquifer potential. This information was re-categorised to produce a map showing only the locations of areas with high productivity geology, as can be seen in figure 3.18.

CHAPTER 4

BENCHMARKING THE PREDICTIVE CAPABILITY OF LUMPED MODELS ACROSS GREAT BRITAIN

This chapter has been published as a research article in Hydrology and Earth System Sciences. Model simulations and figures were developed by Rosanna Lane, with guidance from Jim Freer, Gemma Coxon and Thorsten Wagener. All co-authors provided comments on the final manuscript draft, and we acknowledge review comments from Thibault Mathevet, Nans Addor and an anonymous reviewer that helped to clarify and improve this chapter.

Citation: Lane, R.A., Coxon, G., Freer, J. E., Wagener, T., Johnes, P.J., Bloomfield, J. P., Greene, S., Macleod, J. A., Reaney, S. M (2019). Benchmarking the predictive capability of hydrological models for river flow and flood peak predictions across over 1000 catchments in Great Britain. Hydrology and Earth System Sciences, 23, pp. 4011-4032. https://doi.org/10.5194/hess-23-4011-2019

4.1 Context

Research question 1: How well are simple, conceptual model structures able to simulate high flows across Great Britain?

Evaluating model performance across large samples of catchments is useful to guide model selection, understand reasons for variations in model performance, and to support model development. Given the large uncertainties in hydrological data, model structures and model parameters, it is essential that model evaluation is carried out within an uncertainty framework. This re-

search chapter provides the first multi-model evaluation within an uncertainty framework for a large sample of catchments across GB. This aims to evaluate the predictive capability of lumped models across Great Britain, to improve understanding of how and why model performance varies between catchments and to provide benchmark simulations which will be useful for future model evaluations.

4.2 Introduction

Lumped and semi-distributed hydrological models, applied singularly or within nested sub-catchment networks, are used for a wide range of applications. These include water resource planning, flood and drought impact assessment, comparative analyses of catchment and model behaviour, regionalisation studies, simulations at ungauged locations, process based analyses, and climate or land-use change impact studies (see for example Coxon et al. 2014; Formetta et al. 2017; Melsen et al. 2018; Parajka et al. 2007a; Perrin et al. 2008; Poncelet et al. 2017; Rojas-Serna et al. 2016; Salavati et al. 2015; van Werkhoven et al. 2008). However, model skill varies between catchments due to differing catchment characteristics such as climate, land use and topography. Evaluating where models perform well or poorly and the reasons for these variations in model performance can provide a benchmark of model performance to help us better interpret modelling results across large samples of catchments (Newman et al., 2017) and lead to more targeted model improvements through synthesising those interpretations.

4.2.1 Large-sample hydrology

Large-sample hydrological studies, also known as comparative hydrology, test hydrological models on many catchments of varying characteristics (Gupta et al., 2014; Sivapalan, 2009; Wagener et al., 2010). Evaluating model performance across a large sample of catchments can lead to improved understanding of hydrological processes and teach us a lot about hydrological models, for example, the appropriateness of model structures for different types of catchment characteristics (i.e. Kollat et al. 2012; Van Esse et al. 2013), emergent properties and spatial patterns, key processes that we should be improving, and identification of areas where models are unable to produce satisfactory results (e.g. Newman et al. 2015; Pechlivanidis and Arheimer 2015). This can guide model selection and also teach us about appropriate model parameter values for different catchment characteristics, with the production of parameter libraries which can be used for parameter calibration in ungauged basins, and increase robustness of calibration in poorly gauged basins (Perrin et al., 2008; Rojas-Serna et al., 2016)).

At the same time, regional-scale to continental-scale hydrological modelling studies are increasingly needed to address large-scale challenges such as managing water supply, water scarcity and flood risk under climate change and to inform large-scale policy decisions such as

the European Union's Water Framework Directive (European Parliament, 2000). National-scale hydrological modelling studies using a consistent methodology across large areas are increasingly applied (Coxon et al., 2019; Højberg et al., 2013; McMillan et al., 2016; Van Esse et al., 2013; Veijalainen et al., 2010; Velázquez et al., 2010), facilitated by increasing computing power and the availability of open-source large datasets such as the CAMELS or MOPEX hydrometeorological and catchment attribute datasets in the USA (Addor et al., 2017; Duan et al., 2006). These have great benefits, as applying a consistent methodology across a large area enables comparison between places and identification of areas that may be at the highest risk of future hydrological hazards. However, the range of catchment characteristics and hydrological processes across national scales pose a great challenge to the implementation and evaluation of a national-scale model (Lee et al., 2006), and we therefore need large-scale evaluations of model capability to identify which processes are important and which model structure(s) are most appropriate.

4.2.2 Benchmarking hydrological models

Model skill varies between places, and it is therefore important for a modeller to understand the relative model skill for their study region and how that relates to their core objectives. A single model structure will vary in its ability to produce good flow time series across different environments and time periods (McMillan et al., 2016), expressed sometimes as model agility (Newman et al., 2017). One way to evaluate this relative model skill is by comparing the model performance to a benchmark, which is an indicator of what can be achieved in a catchment given the data available (Seibert, 2001). This helps a modeller make a more objective decision on whether their model is performing well. Examples of benchmarks that models can be evaluated against include climatology, mean observed discharge or the performance of a simple, lumped hydrological model for the same conditions (Pappenberger et al., 2015; Schaefli and Gupta, 2007; Seibert, 2001; Seibert et al., 2018).

The creation of a national benchmark series of performance of simple, lumped models can therefore be useful for a variety of reasons. Firstly, a benchmark series of lumped model performance is a useful baseline upon which more complex or highly distributed modelling attempts can be evaluated (Newman et al., 2015). This would ensure that future model developments are improving upon our current capability, therefore justifying additional model complexity. Secondly, lumped hydrological models provide a good benchmark for evaluating more complex models, as they give an indication of what is possible to achieve for a specific catchment and the available data (Seibert et al., 2018). This can help us identify whether a model is performing well in a catchment relative to how it should be expected to perform for the particulars of that catchment. For example, if a modeller, using more complex modelling approaches, gains an efficiency score of 0.7 for their model in a specific catchment, there is some subjectivity as to whether this is a good or poor performance, depending on the modelling objective. However, if lumped, conceptual

models already applied at the same catchment tend to have efficiency scores of around 0.9 for that catchment, then the modeller knows that their model is performing poorly relative to what is possible. Thirdly, national benchmarks are useful for users of models, as they can highlight areas where models have more or less skill and where model results should be treated with caution.

4.2.3 Assessing uncertainty

Hydrological model output is always uncertain due to uncertainties in the observational data used to drive and evaluate the models and boundary conditions, to uncertainties in selection of model parameters, and to the choice of a model structure (Beven and Freer, 2001a). There is a large and rapidly growing body of literature on uncertainty estimation in hydrological modelling, with many techniques emerging to assess the impact of different sources of uncertainty on model output, as summarised in Beven (2009). Despite this, uncertainty estimation is not yet routine practice in comparative or large-sample hydrology, and few nationwide hydrological modelling studies have included uncertainty estimation, tending to look more at regionalisation of parameters, multi-objective calibration techniques or the use of flow signatures in model evaluation (i.e. Donnelly et al. 2016; Kollat et al. 2012; Oudin et al. 2008; Parajka et al. 2007b).

Parameter uncertainty is often evaluated through calibrating models within an uncertainty evaluation framework (e.g. Generalised Likelihood Uncertainty Estimation - GLUE - Beven and Binley 1992 – or ParaSol – van Griensven and Meixner 2006). Whilst many studies have explored parameter uncertainty, it is less common to evaluate the additional impact of model structural uncertainty on hydrological model output (Butts et al., 2004). Model structures can differ in their choice of processes to include, process parameterisations, model spatial and temporal resolution, and model complexity. Studies attempting to address model structural uncertainty often apply multiple hydrological model structures and compare the differences in output (Ambroise et al., 1996; Perrin et al., 2001; Vansteenkiste et al., 2014; Velázquez et al., 2013) and in climate impact studies (i.e. Bosshard et al. 2013; Karlsson et al. 2016; Samuel et al. 2012). These studies have found that the choice of hydrological model structure can strongly affect the model output, and therefore hydrological model structural uncertainty is an important component of the overall uncertainty in hydrological modelling and cannot be ignored.

Flexible model frameworks are a useful tool for exploring the impact of model structural uncertainty in a controlled way and for identifying the different aspects of a model structure which are most influential to the model output. These flexible modelling frameworks allow a modeller to build many different model structures using combinations of generic model components (Fenicia et al., 2011). For example, the Modular Modeling System (MMS) of Leavesley et al. (1996) allows the modeller to combine different sub-models, and the Framework for Understanding Structural Errors (FUSE), developed by Clark et al. (2008), combines process representations from four

commonly used hydrological models to create over 1000 unique model structures.

4.2.4 Study scope and objectives

The main objective of this study is to comprehensively benchmark performance of an ensemble of lumped hydrological model structures across Great Britain, focusing on daily flow and peak flow simulation. This is the first evaluation of hydrological model ability across a large sample of British catchments whilst considering model structural and parameter uncertainty. This will be useful both as a benchmark of model performance against which other models can be evaluated and improved upon in Great Britain and as a large-sample study which can provide general insights into the influence of catchment characteristics and selected model structure and parameterisation on model performance.

The specific research questions we investigate are as follows:

1. How well do simple, lumped hydrological model structures perform across Great Britain when assessed over annual and seasonal timescales via standard performance metrics?
2. Are there advantages in using an ensemble of model structures over any single model, and if so, are there any emergent patterns or characteristics in which a given structure and/or behavioural parameter set outperforms others?
3. What is the influence of certain catchment characteristics on model performance?
4. What is the predictive capability of those identified as behavioural models for then predicting annual maximum flows when applied in a parameter uncertainty framework?

To address these questions, we have applied the four core conceptual hydrological models from the FUSE hydrological framework to 1013 British catchments within an uncertainty analysis framework. Model performance and predictive capability have been evaluated at each catchment, providing a national overview of hydrological modelling capability for simpler lumped conceptualisations over Great Britain.

4.3 Data and catchment selection

4.3.1 Catchment data

This study was national in scope, using a large data set of 1013 catchments distributed across Great Britain (GB). The catchments cover all regions and include a wide variety of catchment characteristics, including topography, geology and climate (see Table 4.1), and both natural and human-impacted catchments (see Figure 4.1).

On average, rainfall is highest in the north and west of GB, and lowest in the south and east, with GB totals varying from a minimum of 500 mm to a maximum of 4496 mm per year (see

TABLE 4.1. Characteristics of the 1013 study catchments. Values for Mean annual rainfall, runoff, loss, flood peaks and peak daily flows were calculated from the model input timeseries. Other values were taken from the UK hydrometric register (Marsh and Hannaford, 2008).

Variable	5^{th} percentile	Median	95^{th} percentile
Catchment Area [km^2]	17	135	1299
Baseflow Index [-]	0.30	0.47	0.86
Mean Annual Rainfall [mm]	618	975	2332
Mean Annual Runoff [mm]	146	525	1912
Mean Annual Loss [mm]	220	459	693
Median Annual Flood Peak [mm]	2	13	48
Peak Daily Flow [mm]	4	29	100
Gauge Elevation [m]	5	39	220
Urban Extent [%]	0	1	19

Figure 4.2). There is also seasonal variation, with the highest monthly rainfall totals generally occurring during the winter months and the lowest totals occurring in the summer months. This pattern is enhanced by seasonal variations in temperature, with evaporation losses concentrated in the summer months from April–September. Besides climatic conditions, river flow patterns are also heavily influenced by groundwater contributions. Figure 4.1 shows the major aquifers in GB. In catchments overlying the Chalk outcrop in the south-east, flow is groundwater-dominated with a predominantly seasonal hydrograph that responds less quickly to rainfall events. Land use and human modifications to river flows also significantly impact river flows, with river flows being heavily modified in the south-east and midland regions of England due to high population densities (Figure 4.1). Most catchments have very little or no snowfall in an average year, but there are some upland catchments in northern England and north-eastern Scotland where up to 15% of the annual precipitation falls as snow (Figure 4.2).

4.3.2 Observational data

Twenty-one years of daily rainfall and PET data covering the period 1 January 1988 to 31 December 2008 were used as hydrological model input. Rainfall time series were derived from the Centre for Ecology and Hydrology Gridded Estimates of Areal Rainfall, CEH-GEAR (Tanguy et al., 2014). This is a 1 km^2 gridded product giving daily estimates of rainfall for Great Britain (Keller et al., 2015). It is based on the national database of rain gauge observations collated by the UK Met Office, with the natural neighbour interpolation methodology used to convert the point data to a gridded product (Keller et al., 2015).

The Climate Hydrology and Ecology research Support System Potential Evapotranspiration (CHESS-PE) dataset was used to estimate daily PET for each catchment. The CHESS-PE dataset is a 1 km^2 gridded product for Great Britain, providing daily PET time series (Robinson et al.,

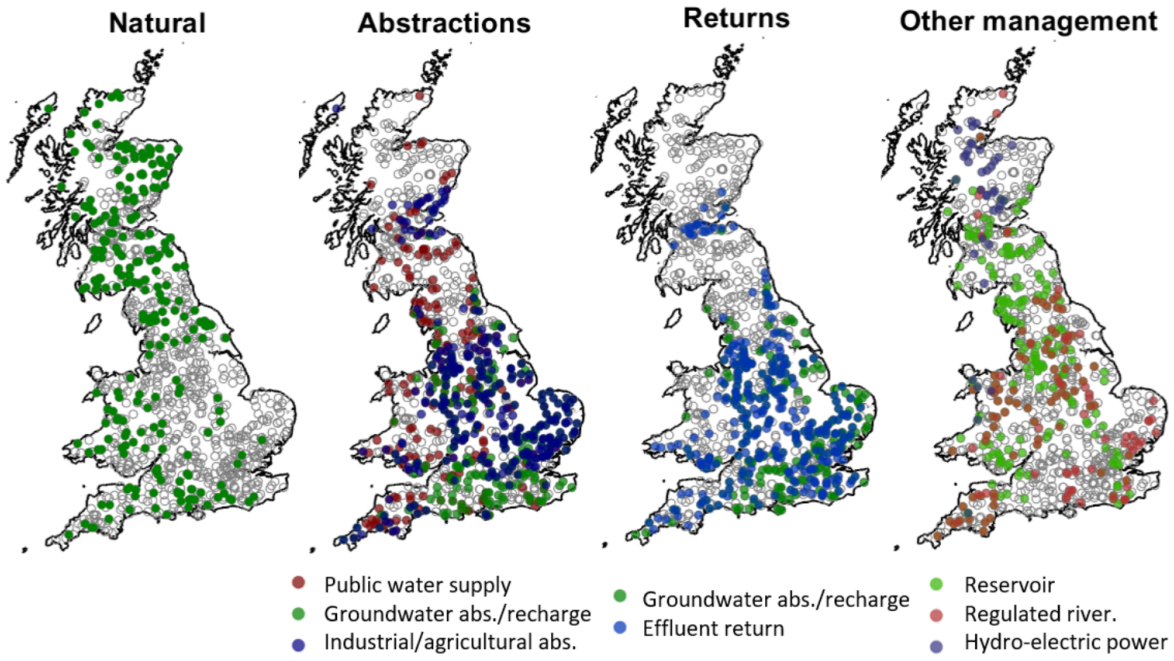


FIGURE 4.1. Factors affecting runoff in the study catchments, using information from the UK hydrometric register. Natural catchments are defined as having limited variation from abstractions and/or discharges so that the gauged flow is within 10% of the natural flow at or above the Q_{95} flow. The groundwater category includes both groundwater abstraction and recharge as well as the few catchments where mine-water discharges influence flow. Full descriptions of all factors can be found in the UK hydrometric register (Marsh and Hannaford, 2008)

2015a). PET estimates were produced using the Penman–Monteith equation, calculated using meteorological variables from the CHESS-met dataset (Robinson et al., 2015b). Catchment areal daily precipitation and PET time series were produced for each catchment by averaging values of all grid squares that lay within the catchment boundaries for each of the 1013 catchments.

Observed discharge data were used to evaluate model performance. Gauged daily flow data from the NRFA were used for all catchments where available (Centre for Ecology and Hydrology, 2016).

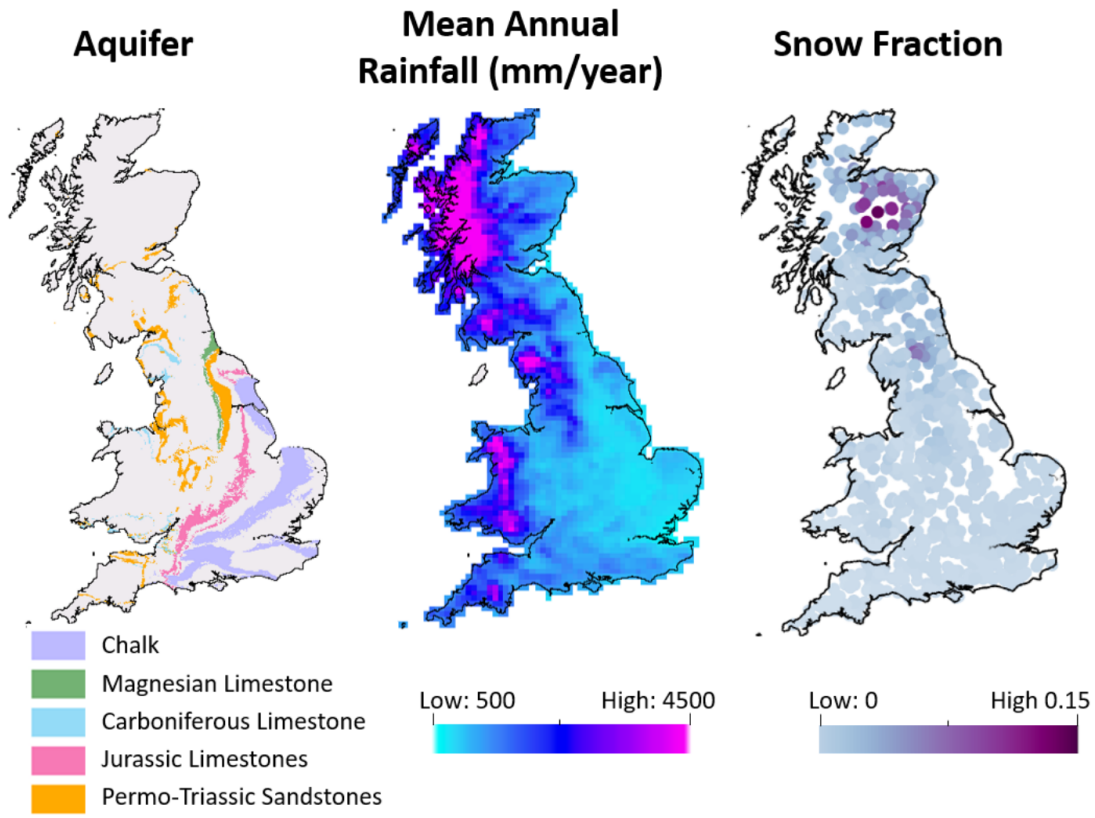


FIGURE 4.2. Attributes impacting hydrology. (a) Major aquifers across Great Britain, based on BSS Geology 625k, with the permission of the British Geological Survey. (b) Mean annual rainfall for 10 km^2 rainfall grid cells across Great Britain calculated using the CEH-GEAR rainfall product. (c) Fraction of rainfall falling as snow for catchments across Great Britain, where a value of 0.15 indicates that 15% of the catchment precipitation falls on days when the temperature is below zero, from the CAMELS-GB dataset (Coxon et al., 2020).

4.4 Methodology

4.4.1 Hydrological modelling

The FUSE modelling framework was used to provide four alternative hydrological model structures. This framework was selected as it enables comparison between hydrological models with varying structural components (Clark et al., 2008), and the computational efficiency of these relatively simple hydrological models enabled modelling to be carried out across a large number of catchments within an uncertainty analysis framework. The framework allows the user to select different combinations of modelling decisions, starting with four parent models based on the structures of widely used hydrological models and allowing the user to combine these decisions to

create over 1000 different model structures.

For this study, only the four parent models from the FUSE framework were selected due to the computational requirements of running the models across such a large number of catchments and because the core models should provide the main differences of models compared to all the possible variants. These models are based on four widely used hydrological models: TOPMODEL (Beven and Kirkby, 1979), the Variable Infiltration Capacity (ARNO/VIC) model (Liang et al., 1994; Todini, 1996), the Precipitation-Runoff Modelling System (PRMS; Leavesley et al. 1983) and the SACRAMENTO model (Burnash et al., 1973). The models are all lumped, conceptual models of similar complexity and all run at a daily time step within the FUSE framework. They all close the water balance, have a gamma routing function and include the same processes; for example, none of the models have a snow routine or vegetation module. However, the structures of these models differ through the architecture of the upper and lower soil layers and parameterisations for simulation of evaporation, surface runoff, percolation from the upper to lower layer, interflow and baseflow (Clark et al., 2008), as shown in Figure 4.3 and Table 4.2. This leads us to believe that the model structures are dynamically different, as they represent hydrological processes in different ways, yet as all are based on widely used hydrological models, they are equally plausible. Therefore we have no a priori expectations that one model should outperform the others (Clark et al., 2008).

The models were run within a Monte Carlo simulation framework. There are 23 adjustable parameters within the FUSE framework, as shown in Table 4.3. Each of these was assigned upper and lower bounds based on feasible parameter ranges and behavioural ranges identified in previous research (Clark et al., 2008; Coxon et al., 2014). Monte Carlo sampling was then used to generate 10,000 parameter sets within these given bounds. Therefore, for each of the 1013 catchments, the four hydrological model structures were each run using the 10,000 possible parameter sets over the 21-year period 1988–2008, resulting in >40 million simulations being carried out.

4.4.2 Evaluation of model performance

The objective of this study was to evaluate the model's ability to reproduce observed catchment behaviour with a focus on assessing the strengths and weaknesses of each model in different catchments. Given the large number of catchments evaluated, it was not possible to evaluate model performance against a large range of objective functions with this chapter; here we aim to benchmark behaviour to metrics that capture different aspects of model performance. Consequently, we chose to evaluate the overall performance of the hydrological models through the widely used Nash–Sutcliffe efficiency index (Nash and Sutcliffe, 1970), which is an easy-to-interpret measure of model performance that is often used in studies interested in high flows, as

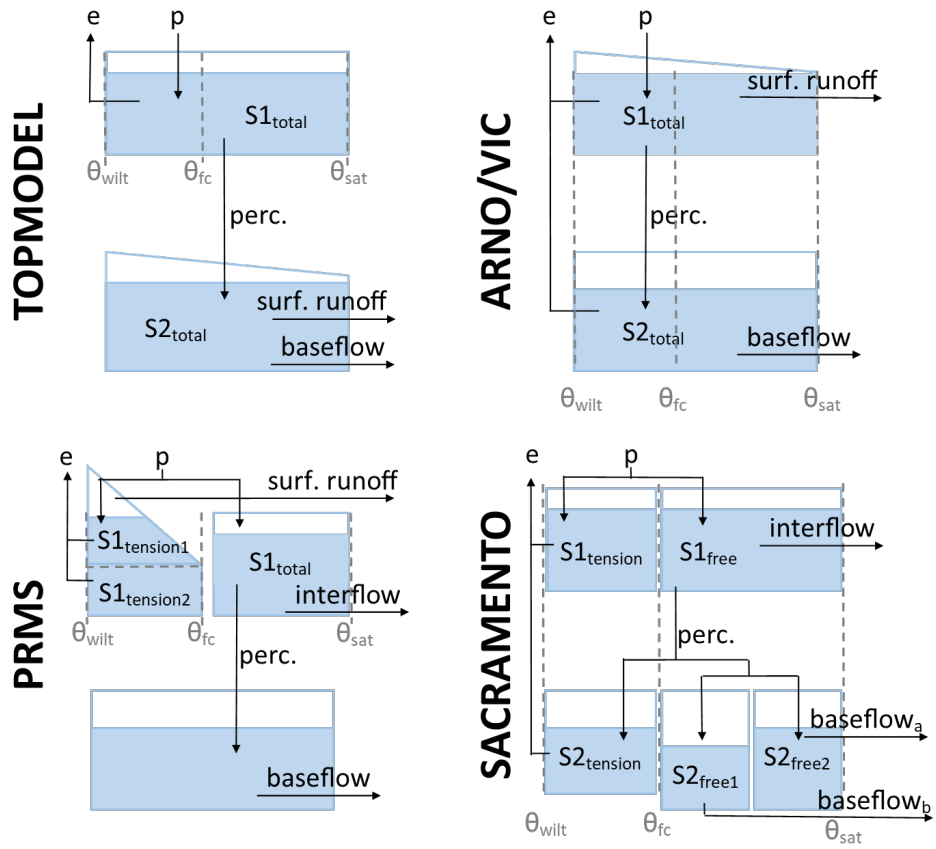


FIGURE 4.3. FUSE wiring diagram, showing the model structure decisions. TOPMODEL and ARNO/VIC have 10 parameters, PRMS has 11 parameters, and SACRAMENTO has 12 parameters. Adapted from Clark et al. (2008).

it emphasises the fit to peaks. To further diagnose the reasons for model good or poor performance, the simulation with the highest efficiency value was then analysed further using the decomposed metrics of bias, error in the standard deviation and correlation. All metrics were calculated for the period 1993–2008, with the first 5 simulation years being used as a model warm-up period.

The Nash–Sutcliffe efficiency index was calculated for each individual simulation using

$$E = 1 - \frac{\sum(O_i - S_i)^2}{\sum(O_i - \bar{O})^2}, \tag{4.1}$$

where O_i refers to the observed discharge at each time step, S_i refers to the simulated discharge at each time step and $\bar{O}$ is the mean of the observed discharge values. This results in values of E between 1 (perfect fit) and -∞, where a value of zero means that the model simulation has the same skill as using the mean of the observed discharge.

TABLE 4.2. Modelling decisions in the four parent models of the FUSE framework. A full description of the models can be found in Clark et al. (2008).

	Upper layer	Lower layer	Surface Runoff	Percolation	Evaporation	Inter-flow	Time delay in runoff
TOPMODEL	Single state variable	Baseflow reservoir of unlimited size, power recession	TOPMODEL parameterization	Water from field capacity to sat available for percolation	Sequential evaporation model	No	Gamma distribution for routing
ARNO/VIC	Single state variable	Baseflow reservoir of fixed size	ARNO/VIC parameterization (upper zone control)	Water from wilting point to sat available for percolation	Root weighting	No	Gamma distribution for routing
PRMS	Tension storage sub-divided into recharge and excess	Baseflow reservoir of unlimited size, frac rate	PRMS variant (fraction of upper tension storage)	Water from field capacity to sat available for percolation	Sequential evaporation model	Yes	Gamma distribution for routing
SACRAMENTO	Broken up into tension and free storage	Tension reservoir plus two parallel tanks	PRMS variant (fraction of upper tension storage)	Defined by moisture content in the lower layer	Sequential evaporation model	Yes	Gamma distribution for routing

To gain insights into model agility and time-varying model performance during different times of the year, we also assessed differences in seasonal performance by splitting the observed and simulated discharge into March–May (spring), June–August (summer), September–November (autumn) and December–February (winter). Seasonal Nash–Sutcliffe efficiency values were then re-calculated for all the catchments, using only data extracted for that season. This allowed us to see if there were any seasonal patterns in model performance, for example during periods of higher or lower general flow conditions.

The Nash–Sutcliffe efficiency can be decomposed into three distinct components: the correlation, bias and a measure of the error in predicting the standard deviation of flows (Gupta et al., 2009). Understanding how the models perform for these different components can help us diagnose why models are producing good or poor simulations. We therefore calculated these simpler metrics for the simulations of each model gaining the highest efficiency values. The relative bias was calculated using

$$\mu = \frac{\mu_s - \mu_o}{\mu_o}, \tag{4.2}$$

where μ_s and μ_o refer to the mean of the simulated and observed annual cycle. Using this

TABLE 4.3. FUSE parameters and defined upper and lower bounds.

Parameter	Description	Units	Lower Bound	Upper Bound	Model(s) using parameter
MAXWATER 1	Depth of upper soil layer	mm	25	500	TOPMODEL, ARNO, PRMS, SAC
MAXWATER 2	Depth of lower soil layer	mm	50	5000	TOPMODEL, ARNO, PRMS, SAC
FRACTEN	Fraction total storage in tension storage	-	0.05	0.95	TOPMODEL, ARNO, PRMS, SAC
FRCHZNE	Fraction tension storage in recharge zone	-	0.05	0.95	PRMS
FPRIMQB	Fraction storage in 1st baseflow reservoir	-	0.05	0.95	SACRAMENTO
RTFRAC1	Fraction of roots in the upper layer	-	0.05	0.95	ARNO
PERCRTE	Percolation rate	mm day^{-1}	0.01	1000	TOPMODEL, ARNO, PRMS
PERCEXP	Percolation exponent	-	1	20	TOPMODEL, ARNO, PRMS
SACPMLT	SAC model percolation multiplier for dry soil layer	-	1	250	SACRAMENTO
SACPEXP	SAC model percolation exponent for dry soil layer	-	1	5	SACRAMENTO
PERCFRAC	Fraction of percolation to tension storage	-	0.5	0.95	SACRAMENTO
FRACLOWZ	Fraction of soil excess to lower zone	-	0.5	0.95	PRMS
IFLWRTE	Interflow rate	mm day^{-1}	0.1	1000	PRMS, SACRAMENTO
BASERTE	Baseflow rate	mm day^{-1}	0.001	1000	TOPMODEL, ARNO
QB_POWR	Baseflow exponent	-	1	10	TOPMODEL, ARNO
QB_PRMS	Baseflow depletion rate	day^{-1}	0.001	0.25	PRMS
QBRATE_2A	Baseflow depletion rate 1st reservoir	day^{-1}	0.001	0.25	SACRAMENTO
QBRATE_2B	Baseflow depletion rate 2nd reservoir	day^{-1}	0.001	0.25	SACRAMENTO
SAREAMAX	Maximum saturated area	-	0.05	0.95	PRMS, SACRAMENTO
AXV_BEXP	ARNO/VIC b exponent	-	0.001	3	ARNO
LOGLAMB	Mean value of the topographic index	m	5	10	TOPMODEL
TISHAPE	Shape parameter for the topographic index Gamma distribution	-	2	5	TOPMODEL
TIMEDELAY	Time delay in runoff	days	0.01	7	TOPMODEL, ARNO, PRMS, SAC

equation, an unbiased model would score 0 (a perfect score) and a model that underestimated or overestimated the mean annual flow would score a negative or positive value respectively. A value of ±1 would indicate an overestimation or underestimation of flow by 100%.

The relative difference in the standard deviation was calculated using

$$\sigma = \frac{\sigma_s - \sigma_o}{\sigma_o}, \tag{4.3}$$

where σ_s and σ_o represent the standard deviation of the simulated and observed mean annual cycle. Again, a value of zero indicates a perfect score with no error, and positive or negative values indicate an overestimation or underestimation of the amplitude of the mean annual cycle respectively.

The correlation was calculated using Pearson's correlation coefficient. A value of 1 indicates a perfect correlation between the observed and simulated flows, whilst a value of 0 indicates no correlation. This indicates model skill in capturing both timing and shape of the hydrograph.

4.4.3 Evaluation of model predictive capability

In order to evaluate model predictive capability, the widely applied GLUE framework was used (Beven and Binley, 1992; Beven and Freer, 2001a; Romanowicz and Beven, 2006). The GLUE framework is based on the equifinality concept that there are many different model structures and parameter sets for a given model structure which result in acceptable model simulations of observed river flow (Beven and Freer, 2001a). This methodology has been widely applied to explore parameter uncertainty within hydrological modelling (Freer et al., 1996; Gao et al., 2015; Jin et al., 2010; Shen et al., 2012) and includes approaches to directly deal with observational uncertainties in the quantification of model performance (Coxon et al., 2014; Freer et al., 2004; Krueger et al., 2010; Liu et al., 2009). For every catchment and model structure, an efficiency score was calculated for each of the 10,000 Monte Carlo (MC) sampled parameter sets. Parameter sets with an efficiency score exceeding 0.5 were regarded as behavioural; therefore all other sampled parameter sets were rejected and so given a score of zero. Conditional probabilities were assigned to each behavioural parameter set based on their behavioural efficiency score, and these were normalised to sum to 1. This meant that the simulations which scored the highest efficiency value had larger conditional probabilities, and simulations which had efficiency values just above 0.5 would have lower conditional probabilities. For each daily time step, a 5^{th}, 50^{th} and 95^{th} simulated discharge bound was produced from these conditional probabilities, for each catchment and model structure individually, as described in Beven and Freer (2001a). This meant that simulations with a higher efficiency score were given a higher weighting when producing the discharge bounds.

Predictive capability for an additional performance metric regarding annual maximum flows was then calculated from these behavioural simulations to test the model's ability to predict peak flood flows over the 21-year period. Annual maximum flows were extracted from both the observed discharge time series and 5^{th}-, 50^{th}- and 95^{th}-percentile simulated discharge uncertainty bounds. Two metrics were then used to assess the predictive capability of the models. The first metric aimed to assess the model's ability to closely replicate the observed annual maximum flows whilst considering the plausible range of observational uncertainties that may be associated with the observed discharge value. Observed uncertainty bounds of ±13% were applied to all observed annual maximum (AMAX) discharges. This observed error value was selected following previous research on quantifying discharge uncertainty at 500 UK gauging stations for high flows and represents the average 95^{th}-percentile range of the discharge uncertainty bounds for high flows (Coxon et al., 2015; Mcmillan et al., 2012). The equations used to calculate the model skill relative to these observational uncertainty bounds are

$$E_y = \frac{|O_y - S_y|}{O_y \times 0.13}, \tag{4.4}$$

$$E_{mean} = \frac{\sum_{y=1}^{n} E_y}{n}, \tag{4.5}$$

where E_y refers to skill for a particular year, y, $Emean$ refers to skill across all years, O refers to observed AMAX discharge for a particular year and S refers to the simulated AMAX discharge for the 50^{th} percentile. This results in a score of 0 if the AMAX that is simulated for the 50^{th} percentile is equal to observed AMAX discharge, a score of 1 if the simulated AMAX is at the limit of the observed error bounds, and a score of 2 if it is twice the limit and so on in a similar approach to Liu et al. (2009) as a limits of acceptability performance score. A score was calculated for each of the 16 simulation years, excluding the first 5 years as a model warm-up period, as shown in Eq. (4.4). A mean score was then calculated across all years for each catchment and model, as shown in Eq. (4.5).

The second metric assessed how well the simulated AMAX uncertainty bounds (5^{th} to 95^{th}) overlapped observed AMAX uncertainty bounds to assess model skill given the range of predictive uncertainty. The range of overlap between the observed discharge uncertainty bounds and simulated bounds was first calculated for each year. This was normalised by the maximum range of the observed and simulated AMAX uncertainty bounds. The resulting value can be interpreted as the fraction of overlap versus the total uncertainty, whereby a value of 0 means that the simulated AMAX bounds for a particular year do not overlap the observations at all, and a value of 1 means that the simulated bounds perfectly overlap the observational uncertainties. Therefore, simulation bounds which overlap the observed AMAX uncertainty range due to having a very large uncertainty spread are penalised for this additional uncertainty width compared to the observed normalised uncertainty.

4.5 Results

4.5.1 National-scale model performance

Our first objective was to assess how well simple, lumped hydrological model structures perform across Great Britain, assessed over annual timescales via standard performance metrics. The distributions of model performance across all catchments can be seen in Figure 4.4. This shows that the ensemble of all four hydrological model structures outperformed each individual model structure for all performance metrics. Using the ensemble, 93% of catchments studied produced a simulation with a Nash–Sutcliffe efficiency (NSE) value exceeding 0.5, and 75% of catchments exceeded an NSE value of 0.7. Maps showing the overall performance of each model structure, chosen using the maximum modelled NSE from the MC parameter samples, for catchments across Great Britain are given in Figure 4.5. Maps showing the performance of each model structure for the other performance metrics are given in Figure 4.6.

Our NSE results (Figures 4.4 and 4.5) show that there is a large range in model performance across Great Britain, with catchment maximum NSE scores ranging from 0.97 to <0. The overall performance of the four model structures was similar, with TOPMODEL, ARNO, PRMS and SACRAMENTO producing simulations exceeding a 0.5 NSE for 87%, 90%, 81% and 88% of catchments respectively. A similar spatial pattern of performance was also seen across all four model structures, with certain catchments resulting in poor or good simulations for all four model structures. Generally, there is an east–west divide in model performance, with models typically performing better in the wetter western catchments compared to drier catchments in the east. Clusters of poorly performing catchments can be seen in the east of England around London and in central Scotland, where all models fail to produce satisfactory simulations. There are also more localised catchments where all models are failing, such as in north Wales and northern England. Areas where all models are performing well include southern Wales, south-western England and south-western Scotland.

However, looking at the decomposed performance metrics in Figures 4.4 and 4.6, differences between the model structures emerge that cannot be seen from the overall NSE scores. Firstly, the models show different biases (Figure 4.6a). The SACRAMENTO model is generally balanced, whilst best-scoring simulations tend to underpredict flows for TOPMODEL and overpredict flows for ARNO/VIC and PRMS. Secondly, all models tend to underpredict the standard deviation of flows (Figure 4.6b), with TOPMODEL generally underpredicting the most, but PRMS stands out as overpredicting the standard deviation for many catchments in the south-east. Thirdly, the pattern of correlation is similar between the models and closely matches the patterns seen for NSE. This is unsurprising, as the correlation term is given a high weighting when calculating NSE (Gupta et al., 2009). It is particularly interesting that whilst the models are all calibrated

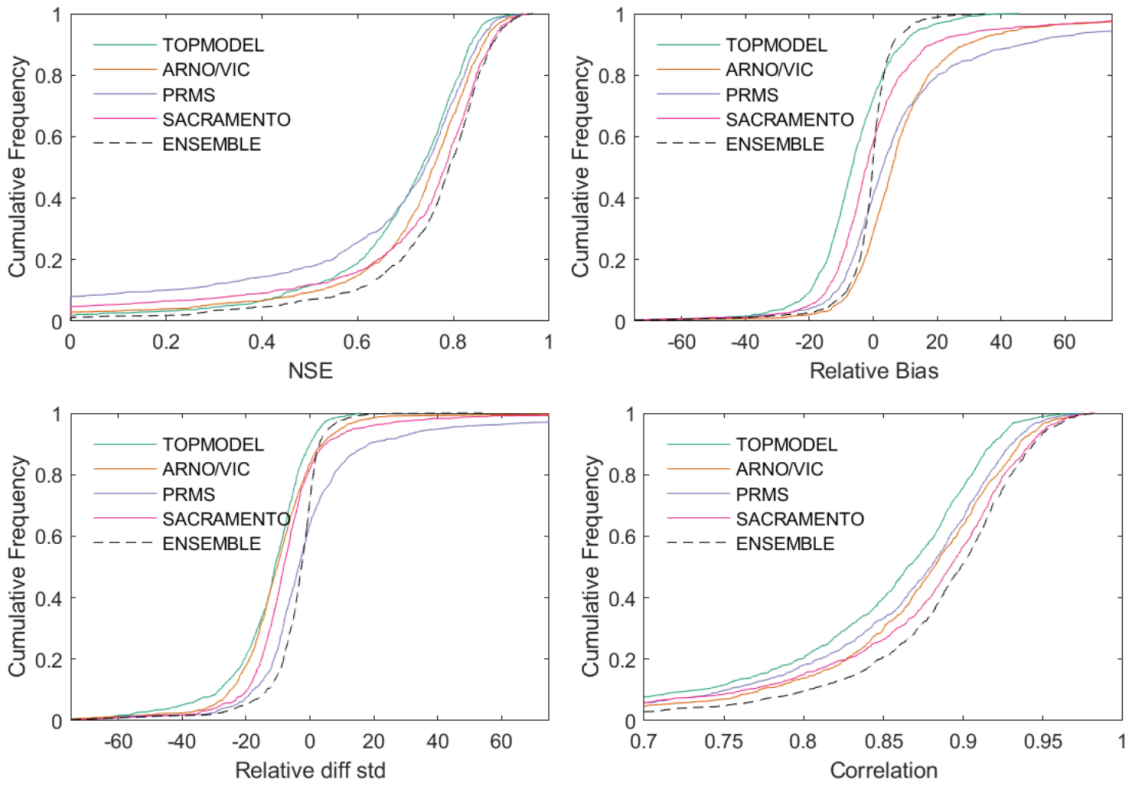


FIGURE 4.4. Distribution of model performance across all catchments for all four individual model structures and the model structure ensemble. Each plot shows model performance assessed using a different metric. (a) shows model performance assessed using Nash–Sutcliffe efficiency, (b) shows model relative bias or relative error in simulated mean runoff (%), (c) shows relative error in the standard deviation of runoff (%), and (d) gives correlation between observed and simulated streamflow.

in the same way and are producing similar NSE scores, the decomposed metrics show clear differences between the best simulations produced using each structure.

The decomposed metrics also help to identify which aspects of NSE are causing models to fail. Models have problems simulating the bias, standard deviation and correlation for catchments in south-eastern England (Figure 4.6). The localised poorly performing catchments in north Wales are failing due to poor simulation of variance and correlation. Poor performance in north-eastern Scotland is due to poor correlation and underestimation of variance for all models. In central northern Scotland all models except TOPMODEL overpredict bias, leading to TOPMODEL being the only model able to produce reasonable simulations for these catchments.

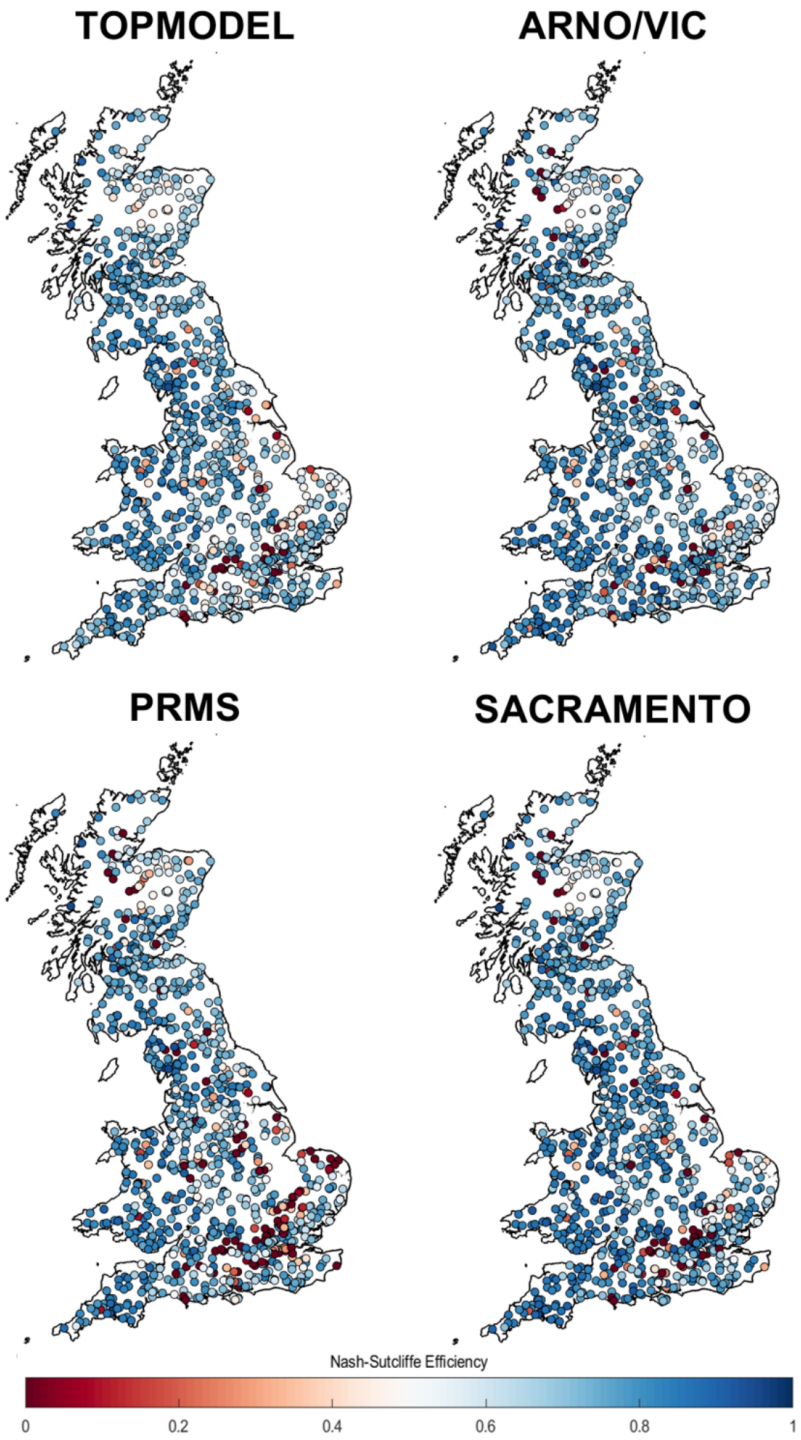


FIGURE 4.5. GB maps of model performance for each structure. Each point is a gauge location which is coloured based on the best Nash–Sutcliffe score attained by the model for that catchment.

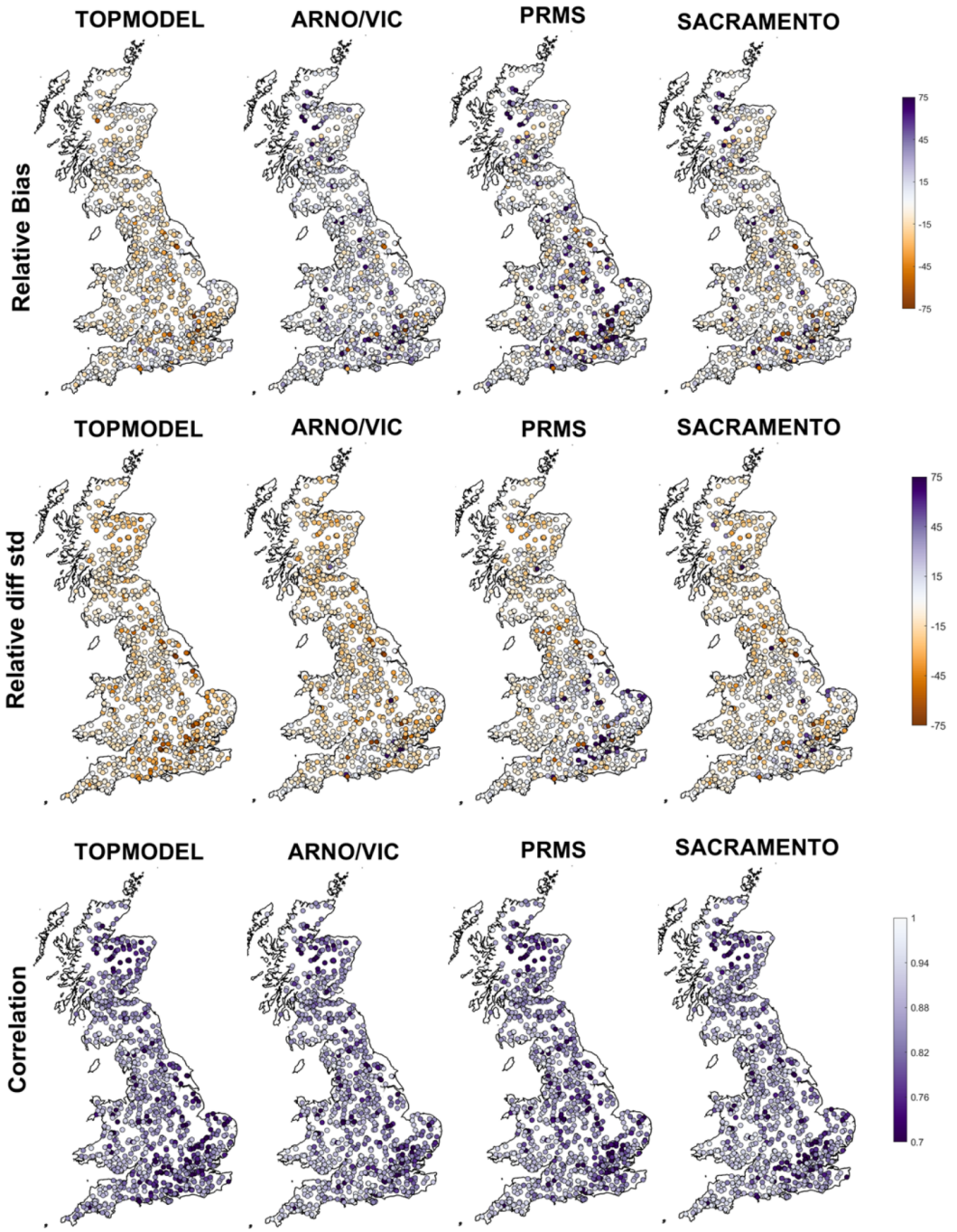


FIGURE 4.6. GB maps of model performance for each structure for three different metrics. (a) shows model relative bias or relative error in simulated mean runoff (%), (b) shows relative error in the standard deviation of runoff (%), and (c) shows correlation between observed and simulated streamflow. Each point is a gauge location, and metrics have been calculated for the simulation gaining the highest NSE for that gauge.

Similarities in overall model performance could be partially due to the models all being run at the same spatial and temporal resolution, having a similar model architecture splitting the catchment into upper and lower stores and including the same process representations (such as a lack of a snow module). However, there are important differences between the models which may contribute to the differences seen in the decomposed metrics (Figure 4.6). The architecture of the upper and lower model layers differs, as can be seen in Figure 4.3. TOPMODEL and ARNO/VIC have more parsimonious structures with only one store in each layer, while PRMS has a more complex upper layer which is split into multiple stores, and SACRAMENTO splits both upper and lower layers into multiple stores. The modelling equations governing water movement between stores also differ, as explained in Clark et al. (2008). The number of model parameters is also a difference between the models, as shown in Table 4.3, with TOPMODEL and ARNO/VIC having the fewest model parameters, with 10 model parameters each, and the SACRAMENTO model having the most parameters, with 12.

4.5.2 Seasonal model performance

As part of our first objective, we also assessed how well models performed across GB when evaluated over seasonal timescales, with results given in Figure 4.7. These maps show the best sampled seasonal NSE score for each catchment taken from any of the FUSE model variants. There is a clear seasonal pattern to model performance, with models generally producing better simulations during wetter winter periods. The models cannot produce adequate simulations for many catchments over the summer months of June to August, especially in the south-east of England. However, for some catchments, especially catchments in the west, good simulations are produced year-round.

There is a seasonal impact on model performance across the areas previously identified as regions where models are failing. In north-eastern Scotland, model performance is generally worst during the winter and spring months of December to May, with a few catchments also being poorly simulated in summer. In south-eastern England, model performance is particularly poor during the summer months of June–August. The reasons for this are discussed in later sections.

4.5.3 Model structure impact on performance

An interesting question is whether a certain model structure is favoured for certain types of climatology or generalised catchment behaviour. Therefore, the relative performance of the four model structures, ranked by both the baseflow index (BFI) and annual catchment rainfall totals, is presented in Figure 4.8. The SACRAMENTO model tends to be the dominant model structure across most catchments, producing the largest number of behavioural simulations. However, catchment BFI and annual average rainfall both have an impact on which model structure tends to produce the most behavioural simulations as well as the total number of behavioural

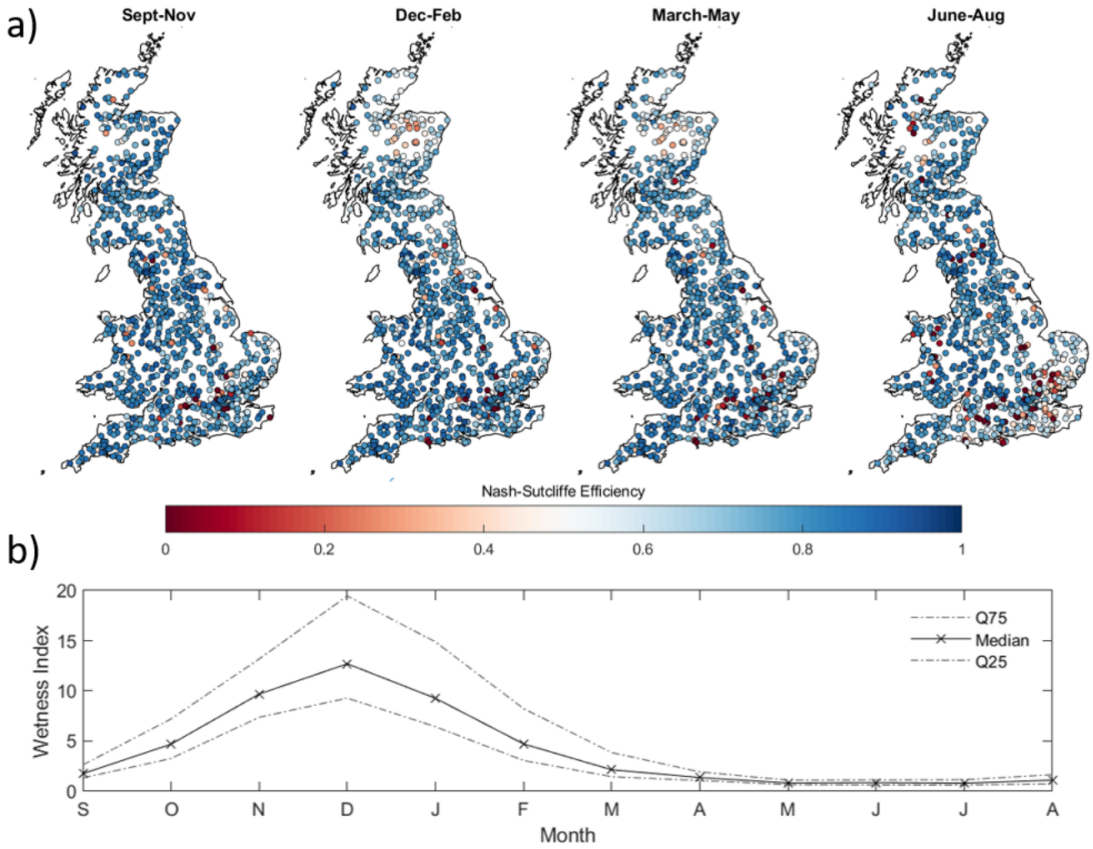


FIGURE 4.7. GB maps of FUSE multi-model ensemble performance for each season (a) and observed seasonal variations in catchment wetness index (b). Each point in (a) is a gauge location which is coloured based on the best Nash–Sutcliffe score attained by any of the four models sampled for that catchment and season. (b) then shows how seasons vary hydrologically across GB, through the wetness index (precipitation divided by PET) calculated from the observed data, split by month, used to drive the hydrological models across all catchments shown in (a).

simulations.

Catchments with an increasing BFI from 0 to 0.87 show an increasing trend of the SACRAMENTO model structure becoming dominant, albeit with considerable variability (see Figure 4.8a). TOPMODEL and PRMS performance relative to the other models decreased for catchments with increasing BFI, and TOPMODEL especially is known to have a conceptual structure that better relates to a variable source area concept that does not relate as well to more groundwater-dominated catchments. However, for slower responding and more groundwater-dominated catchments with a BFI of greater than 0.9, the ARNO/VIC model was the only structure able to represent the hydrological dynamics well. ARNO/VIC is the only model that has a very strong non-linear relationship in its upper storage zone that links the deficit ratio of this store to saturated area extent and thus rainfall-driven surface runoff amounts. For very low values of the ARNO/VIC "b" exponent (AXV BEXP), as seen for high BFI values in Figure 4.9 for behavioural model distributions, means that only at very high, near-full upper storage levels is any larger extent of saturated areas predicted. This formulation clearly helps these more groundwater-dominated catchments where both higher infiltration and percolation dynamics may be expected by constraining fast rainfall-driven runoff processes except to only more extreme storm event behaviour. It is also the reason why the sensitivity to BFI of this parameter is stronger in Figure 4.9 than the other "surface runoff" formulations that link storages to saturated area extent.

For catchments with annual rainfall totals below 2000 mm (see Figure 4.8b), there is no clear relationship between annual rainfall and relative performance of each model structure besides the SACRAMENTO model tending to dominate. However, for catchments with average annual rainfall totals of above 2000 mm, TOPMODEL and ARNO/VIC became more dominant whilst the relative performance of the SACRAMENTO model decreased. In effect the final trend is that for very wet catchment types (by rainfall totals), no model dominates, there is no "gain" in the nuances of the non-linear model formulation and all structures can produce behavioural simulations from some part of their parameter space through a variety of flow pathway mechanisms from different storages. This again is clear in Figure 4.9, where at least three of the parameters shared between structures and controlling different parts of the hydrograph show little sensitivity across the parameter ranges sampled. The core exception to that is the TIMEDELAY parameter that controls the gamma distribution routing formulation and shifts to less routing delay that is common to all model structures and so no one structure has an advantage. Similarly, TIMEDELAY is also sensitive to high-BFI catchments by increasing to longer routing times.

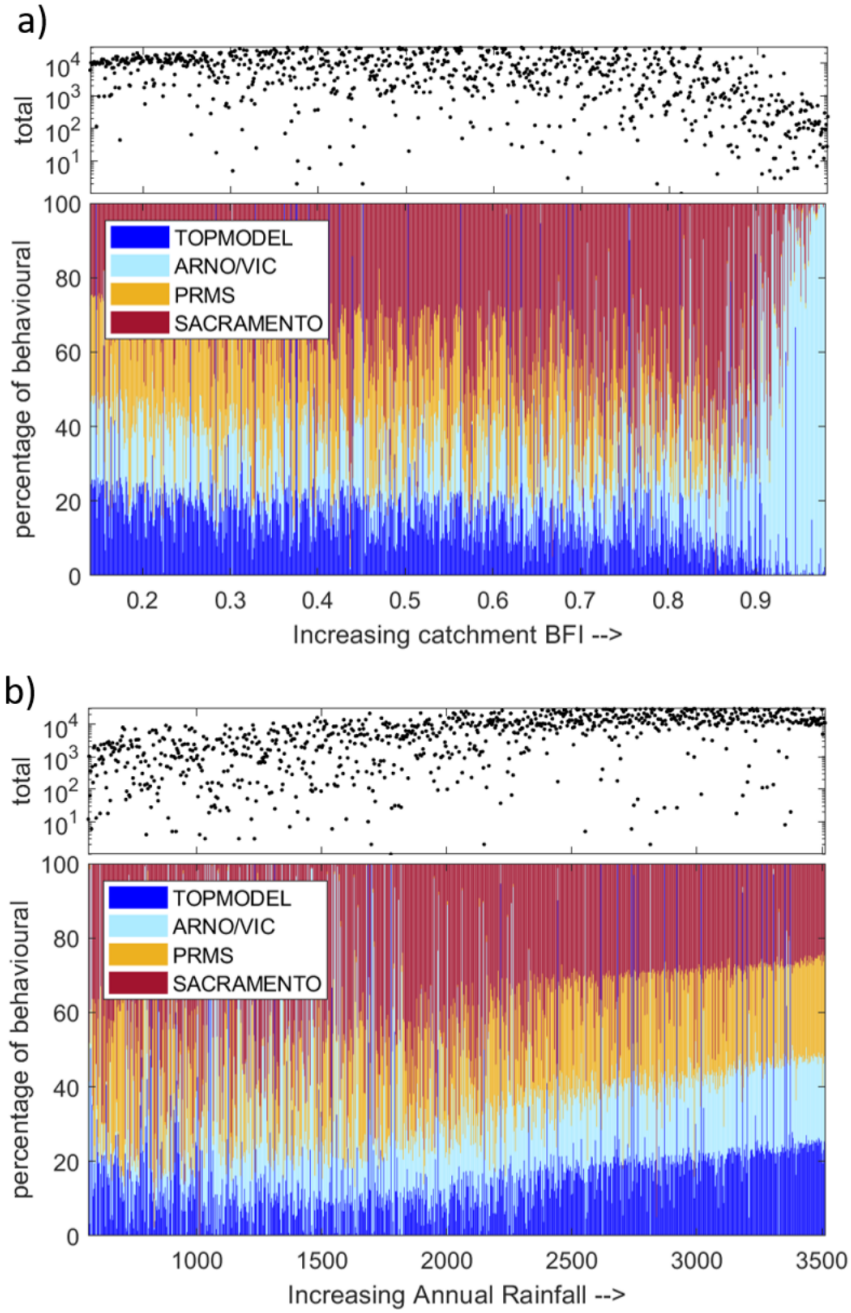


FIGURE 4.8. Relative performance of the four FUSE model structures, depending on catchment characteristics. Scatter plots show the total number of behavioural simulations, from all model structures, forming each line on the stacked bar graph. Each line on this stacked bar chart represents one catchment, and the colour shows the proportion of the behavioural simulations from each model structure. Catchments have been ordered by BFI (a) and annual rainfall (b).

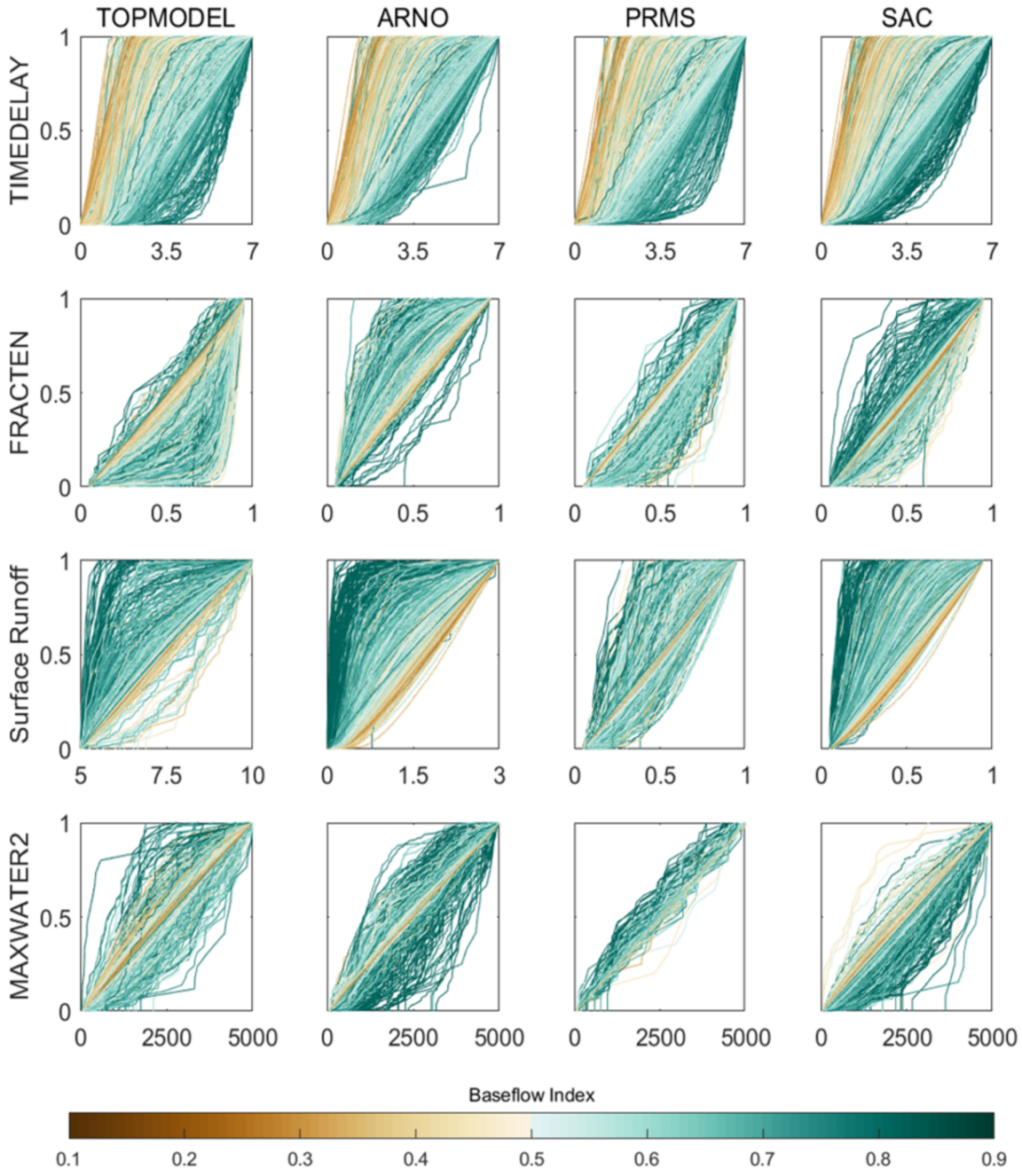


FIGURE 4.9. Cumulative distribution function (CDF) plots showing parameter values of the behavioural simulations for each catchment. Each line represents a catchment and is coloured by that catchment's BFI. The four rows show different parameters controlling different parts of the hydrograph. Surface runoff is given by the LOGLAMB (TOPMODEL), AXV BEXP (ARNO) and SAREAMAX (PRMS and SACRAMENTO), as there was no common surface runoff parameter used for all four models. Each column is a different hydrological model.

4.5.4 Influence of hydrological regime and catchment attributes on performance

The influence of the hydrological regime was then assessed to see if there were specific types of catchments that the models were unable to represent given the spatial differences in model performance already observed. The catchment hydrological regime was defined using two metrics, the overall runoff coefficient (ratio of annual discharge to annual rainfall) and the catchment wetness index (ratio of precipitation to potential evapotranspiration); results are provided in Figure 4.10. The relationship between model performance and a wider range of catchment characteristics is given in Appendix B.

Figure 4.10 shows that model performance relates to the catchment water balance. For catchments where the water balance tends to close, indicated as the area between the dashed lines, the models are generally able to produce reasonable simulations overall and with small biases. For these catchments, precipitation, evaporation and discharge are balanced, and runoff can be explained using the precipitation and evaporation data. When this relationship breaks down, we have situations in which catchment runoff exceeds total rainfall, i.e. there is more water than we would expect, or in which catchment runoff is low relative to precipitation, and this deficit cannot be explained solely by evapotranspiration, i.e. the catchment is losing water. These catchments fall above the top dashed line in Figure 4.10 or below the bottom dashed line respectively. The models cannot simulate these catchments, as they cannot account for large water additions or losses, and so become stressed, leading to large streamflow biases (as also seen in Figure 4.6a). This problem is most extreme for the driest catchments, where models may convert less potential evaporation to actual evaporation as the conditions are drier, and so we have an even larger water deficit which the model structures cannot simulate. For the driest catchments, models have higher error in predicting the standard deviation and correlation.

4.5.5 Benchmarking predictive capability for annual maximum peak flows

Model predictive capability for simulating AMAX flows from behavioural models defined from the NSE measure is shown in Figures 4.11 and 4.12. Figure 4.11 assesses the ability of models to produce AMAX discharge estimates which are as close as possible to observations. Here, a value of 0 means that simulated AMAX discharge is equal to observed discharge, up to 1 means that simulated AMAX discharge is within the bounds of the observational uncertainties applied and larger values such as 2 indicate that simulated discharge is double the limit of observational uncertainties away from the observed discharge (negative values mean that the model simulations are lower than the observed). Median Eamax values from Eq. (4.2) are around -2.4 to -3.2 across all four models, with PRMS producing slightly better predictions in general than the other models. This shows that the models underestimate peak annual discharges across the majority of GB catchments even though behavioural models have been selected using NSE, which favours

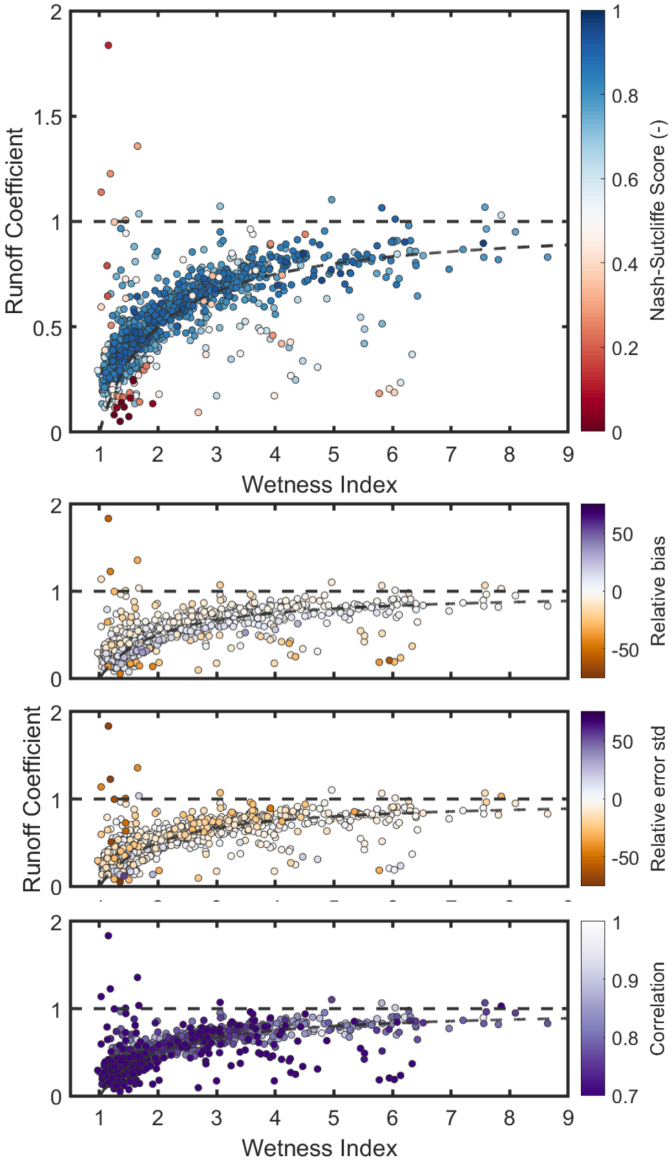


FIGURE 4.10. Scatter plots of the relationship between wetness index, runoff coefficient and best sampled model performance. Each point represents a catchment, coloured by the best Nash–Sutcliffe score for that catchment from the model structure ensemble. The plotting order was modified to ensure that catchments with more extreme (high and low) performance values would be plotted on top. Any points above the horizontal dotted line are where runoff exceeds total rainfall in a catchment, and any points below the curved line are where runoff deficits exceed total PET in a catchment. (a) is coloured by Nash–Sutcliffe efficiency, and (b–d) are coloured by relative bias, relative error in the standard deviation, and correlation between simulated and observed streamflow.

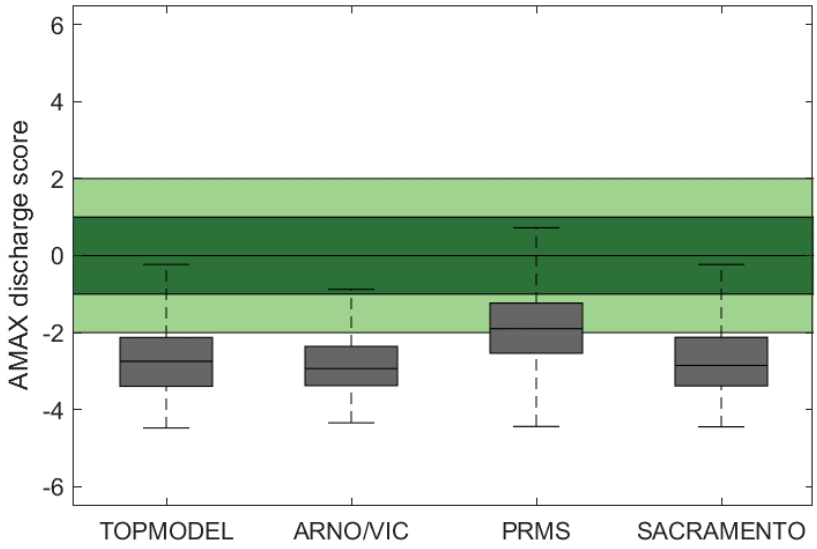


FIGURE 4.11. Predictive capability of four hydrological models for annual maximum (AMAX) flows across Great Britain. Shown is behavioural model ensemble (NSE>0.5) median performance in replicating the observed AMAX flows, with a value of 0 being a perfect score and a value of 1 meaning that the simulated AMAX value was at the limits of the observational uncertainty. The spread covers all catchments.

models that perform well for higher flows.

Figure 4.12 shows the percentage overlap between the simulated 5^{th} and 95^{th} AMAX bounds and the observed AMAX uncertainty bounds. Here, the boxplot on the left shows the variation of results across all catchments and models for each year, whilst the boxplot on the right summarises results across all catchments and years for each model. The median value across all catchments is 16, meaning that there is a 16% overlap between the observed and simulated AMAX bounds averaged across all 20 years.

There are large variations in model ability to simulate observed annual maximum flows between years when looking at median predictions. For example, 1990 and 2008, which were wetter-than-average years across most of GB, model ability to represent annual maximum discharge is poor. However, in 1996, which was a particularly dry year following the 1995 drought (Marsh et al., 2007), the models do a much better job of representing the annual maximum discharge. This may be in part due to the model tendency to underestimate discharge, as seen in Figure 4.11. However, variations between years are less apparent when looking at 25^{th} and th percentiles in Figure 4.12. This could suggest that there are some catchments where predictions

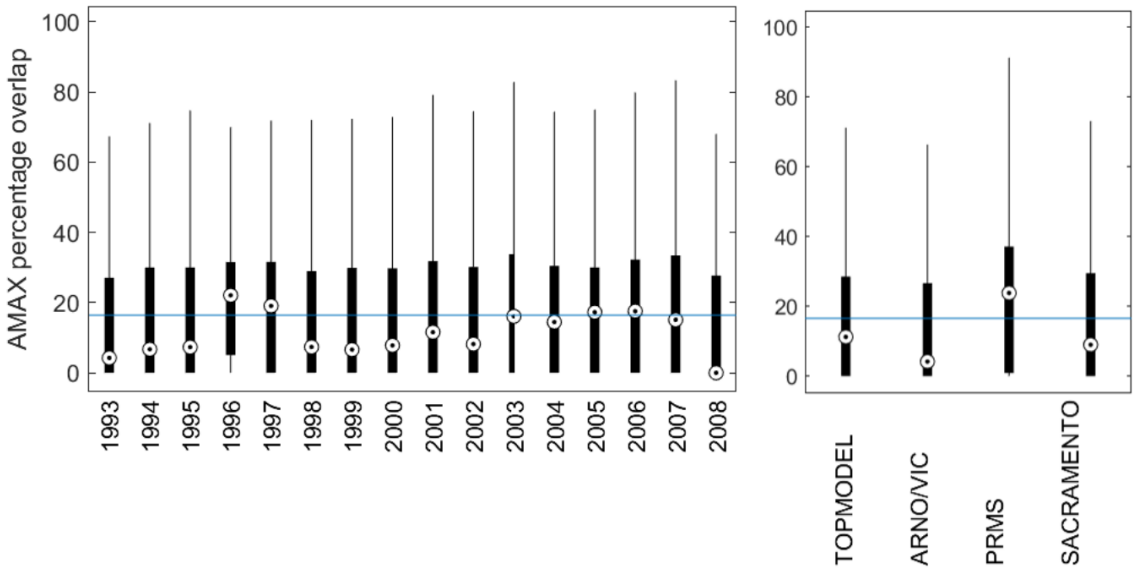


FIGURE 4.12. Predictive capability of four hydrological models for annual maximum (AMAX) flows across Great Britain. Boxplots show the overlap of the simulated and observed uncertainty bounds, as a percentage of the total uncertainty. This metric ranges from 0 to 100, with 0 indicating no overlap between observed and simulated AMAX discharge and 100 indicating a perfect overlap of observed and simulated discharge bounds. The range in the (a) is over all catchments and all models, whilst (b) shows the range across all catchments.

are more consistent between years or that the large climatic variation across GB may conceal some of the effects of inter-year differences.

4.6 Discussion

This study provides a useful benchmark of the performance and associated uncertainties of four commonly used lumped model structures across GB for future model developments and model types to be compared against. The large number of catchments included makes this assessment a fair benchmark for any future national modelling studies as well as for smaller-scale modelling efforts. A full list of models scores can be found at
https://doi.org/10.5523/bris.3ma509dlakcf720aw8x82aq4tm.

4.6.1 Identifying missing process parameterisations

There were some clusters of catchments, notably catchments in northern and north-eastern Scotland and those on permeable bedrock in south-eastern England, where all models failed to produce good simulations. The Scottish catchments are mountainous catchments, at a considerably higher elevation than the rest of GB, and experience colder temperatures, with daily maximum temperatures in January being consistently below zero (Met Office, 2014). Many catchments in north-eastern Scotland are classed as natural, but there are a group of catchments in central northern Scotland which are impacted by hydro-electric power (HEP) generation and subsequent diversions out of the catchment as well as storage influences on the regime (Marsh and Hannaford, 2008). As model failures in north-eastern Scotland were particularly pronounced during winter and spring, this suggests that models were unable to capture the different seasonal climatic conditions of these catchments, such as snow accumulation and melt or the impact of frozen ground. This is supported by the low correlations between simulated and observed flows in north-eastern Scotland, suggesting that the models are unable to represent the overall shape and timing of flows. Many catchments in central and northern Scotland had particularly low NSE values which were worst in summer and autumn. Modifications to the flow regime resulting from HEP can explain poor model performance for these catchments, supported by the models failing to reflect model bias and correlation. The FUSE models in this study do not incorporate snow processes and indicate that future modelling efforts for GB may need to include a snowmelt regime, and the anthropogenic impacts resulting from hydroelectric power generation, to produce good simulations in these catchments.

The catchments in south-eastern England receive relatively little rainfall compared to the rest of GB and are overlaying a chalk aquifer, as can be seen in Figure 4.2. Previous studies have found that hydrological models tend to perform better in wetter catchments (Liden and Harlin, 2000; McMillan et al., 2016), which could be part of the reason that model performance is so poor for these catchments. The presence of the chalk aquifer could also stress the models, as there is nothing in the model structures to account for groundwater and particularly groundwater flows between catchment boundaries. Equally, the south-east has some of the highest population densities in the UK, and human influences can significantly impact flows in this region, particularly for lower-flow conditions in the drier seasonal periods.

For catchments where groundwater is the reason for model failure, a possible solution could be to use a conceptual model that allows for groundwater exchange (as opposed to the models used here, which all maintain the water balance). Hydrological models such as GR4J and the Soil Moisture Accounting and Routing (SMAR) model have been developed with functions that allow models to gain or lose water to represent inter-catchment groundwater flows (Le Moine et al., 2007). The use of these models where there is evidence of groundwater flows can help to improve

model performance and reduce discrepancies between observed and simulated flows, but they must be used with caution to avoid overfitting of the water balance where there is no physical reasoning for a catchment to gain or lose water. Whilst it has been noted that there is a general pattern of poor performance for catchments in south-eastern England, it is hard to disentangle the reasons why this may be the case. Both the underlying chalk geology causing water transfer between catchments and heavily human-modified flow regimes could explain model failures which are greatest during the summer. Interestingly, McMillan et al. (2016) found that whilst the aquifer fraction was expected to have a strong link to model performance, no relationship was found for the TOPNET model applied in New Zealand.

4.6.2 Influence of catchment characteristics and climate on model performance

One of the key advantages of large-sample studies is that by applying models to many catchments, we can see general trends and identify important catchment characteristics or climates that are not represented well by our choice of model structures. We found that looking at the catchment water balance, considering the relationship between catchment precipitation, evaporation and observed flows, helped to identify common features of catchments where all models were failing (Figures 4.5 and 4.10). All model structures produced poor simulations in catchments where either total runoff exceeded total rainfall or where observed runoff was very low compared to total rainfall, and this runoff deficit could not be accounted for by evapotranspiration losses alone. These differences in water balance are likely due to human modifications to the natural flow regime, such as dams, effluent returns, or inter-catchment water transfers or groundwater flow between catchments, or it is also possible that there are systematic errors in the observational data and that this information is dis-informative (Beven, 2012; Kauffeldt et al., 2013). Most of these catchments were located within chalk aquifers in south-eastern England and therefore are in a heavily urbanised area where groundwater abstractions and flows between catchments could be expected. The simple, lumped models used here were only given inputs of observed precipitation and PET; therefore they are unable to account for the additional observed runoff and so are "stressed", even in terms of simulating mean annual runoff, irrespective of more detailed hydrograph behaviour.

We also found that catchment characteristics were important in determining which model structure was most appropriate. For catchments with a high baseflow index, only the ARNO/VIC model was able to produce behavioural simulations. This could be explained by the strong non-linear relationship in the upper storage zone of the ARNO/VIC model, which separates it from the other model structures. This enables the ARNO/VIC model to constrain the fast rainfall–runoff processes, which would only occur for extreme events in these groundwater-dominated catchments and so allow for a complex mixture of highly non-linear saturated fast responses coupled

with more general baseflow dynamics to be captured effectively. The catchment annual rainfall total also influenced which model structure was most appropriate. We found that for catchments with average annual rainfall values of around 2000 mm yr^{-1} or lower, the SACRAMENTO model structure is more dominant. As we move towards catchments with higher annual rainfall, the relative importance of the different structures shift until all structures are approximately equal for the catchments with the highest annual rainfalls. This shows that for very wet catchments, the model structure is less important, as all models can produce behavioural simulations through some part of the parameter space, as seen by the relatively high number of behavioural simulations for wetter catchments (Figure 4.8b). This agrees with previous studies, where models have been found to perform better for wetter catchments, which are likely to have more connected saturated areas, as there is a more direct link between rainfall and runoff (McMillan et al., 2016).

Our results highlight the difficulty in national and large-scale modelling studies, which for GB must incorporate human-modified hydrological regimes, complex groundwater processes, a range of different climates and the potential of dis-informative data, or at least a lack of process understanding to adjust model conceptualisations. Whilst simple, lumped hydrological models can produce adequate simulations for most catchments, the model structures are put under too much stress when trying to simulate catchments where the water balance does not close or is increasingly departing from normal conditions. The models fail or produce poor simulations when large volumes of water enter or leave the catchment due to human activities or groundwater processes, indicating the importance of considering these influences in any national study. What is striking here in these results is that general hydrological processes, defined by water availability and BFI metrics to infer the extent of slower flow pathways, are important in defining the quality of simulated output and differences in model structures and parameter ranges even though nationally many catchments are impacted by additional anthropogenic activities such as abstractions and multiple flow structures.

4.6.3 Predictive capability of models for annual maximum flows

Predictions of annual maximum discharge using behavioural models based on NSE posed a larger challenge for the models, even when allowing for an estimate of observational uncertainty from results generalised in Coxon et al. (2015). It was found that all model structures systematically underpredicted annual maximum flows across most catchments, which could have large implications if these structures were used for flood modelling or forecasting. These results are in line with previous large-scale modelling efforts. McMillan et al. (2016) report that their TOPNET model applied across New Zealand showed a smoothing of the modelled hydrograph relative to the observations, which resulted in overestimation of low flows and underestimation of annual maximum flows. Newman et al. (2015) found the same effect in their study covering 617 catchments across the US. This underestimation of peaks could be in part due to the use of

NSE in selection of the behavioural models. NSE is often used in flood studies, as it emphasises correct prediction of flood peaks relative to low flows (for example, Tian et al. 2013). However, NSE tends to underestimate the overall variance in the time series, resulting in underprediction of floods and overprediction of low flows (Gupta et al., 2009).

It was found that there were some variations in the ability of models to simulate AMAX flows between years, and this often related to the wetness of a particular year. Models tended to perform worse in wetter years and better in drier years. This could be linked to the fact that all models tended to underestimate annual maximum flows and therefore are closer to observations in years with lower annual maximum flows.

4.6.4 Uncertainty evaluation in hydrological modelling

This study evaluated both model parameter and model structural uncertainty. The results showed that there is considerable value in using multiple model structures. No one model structure was appropriate for all catchments or seasons and when evaluating different metrics from the hydrographs. We found that generally the SACRAMENTO model resulted in the best NSE values overall, TOPMODEL was able to produce the simulations with the least biases and the ARNO/VIC model proved to be best for high baseflow catchments, though the PRMS model was the best at capturing AMAX peak flows. Furthermore, it was found that for some catchments only a selection of the model structures were able to produce good simulations, such as the baseflow-dominated catchments which only ARNO/VIC could simulate well. For these catchments, selection of the appropriate model structure is important for producing good simulations, and unsuitability of the model structure cannot be corrected for through parameter calibration. This supports previous research highlighting the importance of considering alternative model structures and using model structure ensembles or flexible frameworks such as FUSE (Butts et al., 2004; Clark et al., 2008; Perrin et al., 2001). Consequently, future hydrological modelling over a national scale and/or over a large sample of catchments needs to ensure that appropriate model structures are selected for these catchments and consider the possibility of using multiple model structures to represent hydrological processes in varied catchments.

The results also highlighted the importance of considering parameter uncertainty. It was shown that there were often many different parameter sets which could produce good simulation results for the same model structure. For some catchments, particularly the wetter catchments in the west, all model structures were able to produce good simulations through sampling the parameter space. We also show how behavioural parameter distributions change with regards to the BFI (Figure 4.9), which shows expected shifts in some of the common behavioural parameters or concepts for different conditions, showing that the model behaviour and parameter formulations are in general making rational sense (i.e. higher BFI equals higher time delays).

While this study incorporated uncertainties in model structures and parameters, future work will also focus on incorporating uncertainties in the data used to drive hydrological models and more sophisticated representation of discharge uncertainties. This is important because errors in observational data will introduce errors to runoff predictions when fed through rainfall–runoff models (Andréassian et al., 2001; Fekete et al., 2004; Yatheendradas et al., 2008), and in conjunction with uncertainties in the observational data used to evaluate hydrological models, they will also affect our ability to calibrate and evaluate hydrological models (Blazkova and Beven, 2009a; Coxon et al., 2014; McMillan et al., 2010; Westerberg and Birkel, 2015).

4.7 Summary and conclusions

In this study, we have benchmarked the performance of an ensemble of lumped, conceptual models across over 1000 catchments in Great Britain. Overall, we found that the four models performed well over most of Great Britain, with each model producing simulations exceeding a 0.5 Nash–Sutcliffe efficiency over at least 80% of catchments. The performance of the four models was similar, with all models showing similar spatial patterns of performance and no single model outperforming the others across all catchment characteristics for both daily flows and peak flows. However, when decomposing NSE into model performance for bias, standard deviation error and correlation, clear differences emerged between the best simulation produced by each of the model structures. The ensemble did better than each individual model, demonstrating the value of model structure ensembles when exploring national-scale hydrology.

We found that all models showed higher skill in simulating the wet catchments to the west, and all models failed in areas of Scotland and south-eastern England. Seasonal performance and analysis of the water balance suggested that these model failures could be at least in part attributed to missing snowmelt or frozen ground processes in Scotland and chalk geology in south-eastern England, where water was able to move between catchment boundaries. In general, we found that models performed poorly for catchments with unaccounted losses or gains of water, which could be due to measurement errors, water transfer between catchments due to groundwater aquifers and human modifications to the water system. Therefore, these factors would need to be considered in a national model of Great Britain.

We also evaluated model predictive capability for high flows, as good model performance in replicating the hydrograph, assessed using Nash–Sutcliffe efficiency, does not necessarily mean that models are performing well for other hydrological signatures. We found that the FUSE models tended to underestimate peak flows, and there were variations in model ability between years, with models performing particularly poorly for extremely wet years.

This benchmark series provides a useful baseline for assessing more complex modelling strategies. From this we can resolve how or where we can and need to improve models to understand the value of different conceptualisations, linkages to human impacts and levels of spatial complexity that our model frameworks could deploy in the future. Therefore, the results of this study are made available at https://doi.org/10.5523/bris.3ma509dlakcf720aw8x82aq4tm.

CHAPTER 5

DEVELOPING NATIONALLY CONSISTENT PARAMETER FIELDS WITH UNCERTAINTY

This chapter has been submitted as a research article to Water Resources Research, with slight modifications made to better fit the general layout of this thesis. Model developments, simulations and figures were carried out by Rosanna Lane, with guidance from Gemma Coxon, Jim Freer and Thorsten Wagener. The manuscript was written by Rosanna Lane, with comments from all co-authors.

Citation: Lane, R.A., Freer, J. E., Coxon, G., & Wagener, T. (in review). Incorporating Uncertainty into Multiscale Parameter Regionalisation to Evaluate the Performance of Nationally Consistent Parameter Fields for a Hydrological Model. Submitted to Water Resources Research.

5.1 Context

Research question two: Can observed datasets be used to parameterise a spatially distributed model across Great Britain, including parameter uncertainties?

The estimation of spatial parameter fields remains a key challenge for the implementation of spatially distributed models across large-domains (Archfield et al., 2015; Mizukami et al., 2017), which can include ungauged areas and complex, nested catchments. A common regionalisation approach uses transfer functions to relate model parameters to spatial geophysical data (e.g. topography, land-use, soils), introducing large uncertainties. Here, we address this challenge by presenting a framework to incorporate uncertainties into a parameter regionalisation technique

(Samaniego et al., 2010), and we evaluate this method of constraining model parameters across gauged and ungauged catchments for the DECIPHeR model across GB.

5.2 Introduction

Distributed and semi-distributed hydrological models are widely used for a variety of purposes such as flood forecasting, drought monitoring and climate change impact analyses (Bell et al., 2009; Cloke and Pappenberger, 2009; Luo and Wood, 2007; Velázquez et al., 2013). There is an increasing drive towards large-scale models, which can be applied at high resolution (1 km or finer) across national to continental domains (Bierkens et al., 2015; McMillan et al., 2016; Wood et al., 2011). Such models are required to help understand large-scale pressures on water systems, such as climate change impacts on river flow, because consistent modelling approaches applied across large areas give the broad overview of future impacts needed to guide policy decisions (Watts et al., 2015). By discretising the landscape, spatially explicit models can characterise spatial processes or changes, such as land-use change impacts on river flow (e.g. Hundecha and Bárdossy 2004; Im et al. 2009; Niehoff et al. 2002). However, there are many uncertainties involved in the modelling process, including selection of the model structure, model parameterisation and errors in observational data (Butts et al., 2004; Clark et al., 2008; Mcmillan et al., 2012; Wagener and Gupta, 2005; Wilby and Harris, 2006). If hydrological models are used to guide policy decisions, it is vital that these modelling uncertainties are recognised and reported alongside model output.

A key challenge for the application of large-scale hydrological models is constraining model parameters, particularly for spatially distributed models which require explicit representation of the parameter fields (Archfield et al., 2015; Beven and Cloke, 2012; Clark et al., 2017; Gupta et al., 2014). Model parameterisation is important to ensure that the model is best able to represent the unique combination of climatic and physiographic factors that govern hydrological processes within a given catchment or area (Wagener and Wheater, 2006). However, the choice of model parameter values is uncertain as model parameters are effective values which cannot be directly measured. Many different parameter sets can often produce equally reasonable simulations, known as parameter equifinality, making it difficult to justify selecting a single parameter set (Beven and Freer, 2001a). This is particularly the case for spatially distributed models, where different spatial configurations of model parameters may produce similar outputs (Kelleher et al., 2017).

A variety of techniques have emerged to parameterise hydrological models across large domains. Firstly, many studies use a priori parameter values based on physical interpretation of model parameters, i.e. an uncalibrated model (Bathurst, 1986; Beven and O'Connell, 1981; McMillan et al., 2016) but have often had limited success in predictive skill compared to models with

some calibration against runoff (Duan et al., 2006). This method is also more difficult for conceptual parameters which do not directly relate to measurable properties. Secondly, parameters can be calibrated individually for each catchment across a region (Christierson et al., 2012), as done in Research Chapter One. To include ungauged catchments, parameter sets can subsequently be transferred based on catchment similarity or individual parameters can be transferred based on regression with catchment characteristics (McIntyre et al., 2005; Merz and Blöschl, 2004; Oudin et al., 2008; Parajka et al., 2005). Whilst this can result in good performance at catchment outlets, at the national scale it can result in a patchwork quilt of parameter values with potentially unrealistic discontinuities at catchment boundaries (Mizukami et al., 2017). Thirdly, regional parameterisation methods (also referred to as simultaneous regionalisation) link parameters to spatial catchment data through transfer functions, and calibrate the transfer function parameters simultaneously across many gauges (Götzinger and Bárdossy, 2007; Hundecha and Bárdossy, 2004; Samaniego et al., 2010). This has many advantages over the previous approach including; (a) the use of spatial catchment attribute data helps constrain the spatial pattern of parameter values, (b) it creates seamless parameter fields with no artificial discontinuities between catchments, (c) it uses a consistent methodology, with the same transformation of geophysical data to model parameters, everywhere including ungauged areas, (d) by regionalising parameters simultaneously across all catchments it can result in more robust parameter sets.

We have chosen to focus on multiscale parameter regionalisation (MPR), which has emerged as a particularly promising simultaneous regionalisation strategy for large scale modelling (Mizukami et al., 2017; Samaniego et al., 2017). MPR differs from other regionalisation methods by first creating parameter fields at the high-resolution of the geophysical data before upscaling to the model resolution. This improves the transferability of the method across modelling resolutions, making it the ideal choice for a spatially flexible hydrological model, such as the one we have used here. Since the methodology was first introduced (Samaniego et al., 2010), MPR has been applied to multiple models and locations (Kumar et al., 2013a,b; Mizukami et al., 2017; Samaniego et al., 2017; Wanders et al., 2017), but relatively little work has been carried out exploring the uncertainties in the MPR process. Uncertainties within the MPR methodology include (1) structure of the transfer functions, (2) choice of catchment attribute datasets, (3) selection of upscaling operators, (4) errors in hydrological data used for model calibration and evaluation, (5) errors in catchment attribute data used for parameter regionalisation, and (6) transfer function parameter values (or global parameter values). Most previous studies have applied MPR deterministically using a single optimum global parameter set (Kumar et al., 2013a; Livneh et al., 2015; Mizukami et al., 2017). However, Livneh et al. (2015) have compared model performance regionalised using two different soil datasets, and found that to be an appreciable source of uncertainty.

In this chapter, we apply MPR to the DECIPHeR hydrological modelling framework across Great Britain for the first time, thereby uniquely demonstrating how the method can be applied within an uncertainty framework. We focus on uncertainties in the transfer function parameters (i.e. global parameters), which will also implicitly reflect some of the uncertainties in the underlying geophysical data. The goals of this work are to (1) demonstrate how MPR can be applied within an uncertainty framework to investigate uncertainties in the global parameter values (2) evaluate MPR as a technique to produce nationally consistent parameter fields for a hydrological model across Great Britain, and (3) evaluate whether the increased resolution of parameter fields enabled through MPR adds value to the simulations. To answer these questions, we have tailored MPR to DECIPHeR, constraining the model within the GLUE uncertainty analysis framework to produce an ensemble of nationally consistent parameter fields. This differs from previous work exploring uncertainties in global parameter values, as we are exploring the full range of global parameter values in a Monte Carlo framework, and not optimising (Rakovec et al., 2016a). To explore the relative skill of the national MPR parameterisation, we compare it against two upper benchmarks: (1) individual catchment calibration using MPR to demonstrate the best possible performance of the DECIPHeR-MPR method for a particular catchment, and (2) constraining parameters directly using Monte-Carlo parameter sampling to demonstrate the best performance of DECIPHeR with homogeneous model parameters.

5.3 Data and methods

5.3.1 Applying MPR within an uncertainty framework

The multiscale parameter regionalisation (MPR) approach introduced by Samaniego et al. (2010) links model parameters to catchment predictor variables (spatial catchment attributes/ geophysical data) through transfer functions. The transfer functions are applied at the highest possible resolution and are then upscaled to the resolution of the model. The parameters of the transfer functions, often called transfer function parameters or global parameters, are usually then calibrated or optimised by assessing how well model simulations with different global parameter values perform relative to observations.

A key assumption of the MPR technique is that the links made to catchment predictor variables are informative and the spatial distribution of catchment predictor variables helps identify the spatial distribution of parameter values. However, model parameters are often effective values that cannot be measured and defining appropriate catchment predictors and transfer functions is therefore a difficult and uncertain task. This is especially the case for conceptual models, where model parameters may not directly link to a physical process. There are therefore many modelling uncertainties introduced in the MPR process, few of which have been fully investigated.

Here, we apply MPR within the generalised likelihood uncertainty estimation (GLUE) framework to explore the uncertainties relating to the global parameter values. The global parameters control the link between geophysical data and the model parameters. Thus, varying these values accounts for systematic errors in the geophysical data, which are represented by uncertainties in the global parameter values. The GLUE framework follows the equifinality concept, that there can be many different parameter sets or model setups that result in equally acceptable simulations (Beven and Binley, 1992). This equifinality results from errors and uncertainties in the modelling process, where we lack the information to justify choosing one model realisation over all others. Using GLUE we therefore sample the global parameters and keep the multiple sets which produce acceptable simulation results rather than calibrating the global parameters to find a single optimum. When making predictions, all behavioural model realisations are used, each weighted by their likelihood, to produce discharge bounds indicating the modelling uncertainties. The ethos behind the GLUE methodology is that it is a continuous process, and we can continually update our definition/selection of behavioural sets of model parameters if more information becomes available.

We first setup the modelling framework, regionalising each model parameter using multiscale parameter regionalisation (the model setup, model parameters, transfer functions and upscaling operators are described in depth in section 5.3.2). This resulted in 22 global parameters being defined, to produce spatial parameter fields for five model parameters. Upper and lower bounds were set for the global parameters, with the aim of selecting bounds that were wide enough to produce the full range of realistic parameter values yet small enough to avoid wasting computational effort running and evaluating simulations with unrealistic parameter values (see Table 5.1 and Table 5.2). A bootstrap test was then carried out on a selection of six hydrologically varied catchments, to determine the number of model evaluations (n) required to effectively sample the global parameter space (details given in Appendix D). Model simulations were carried out for each catchment, sampling the global parameters n times between the given bounds using uniform random sampling. It is important to note that global parameters were considered as sets, and not as individual parameter values. This means that parameter interactions are implicitly considered within the GLUE approach.

5.3.2 Linking parameter values to catchment descriptors in DECIPHeR

DECIPHeR, Dynamic fluxEs and ConnectIvity for Predictions of HydRology, is a flexible hydrological modelling framework, capable of running hydrological simulations from catchment to national or continental scales (see Coxon et al. 2019 for a full description of this model). The flexible modelling framework allows the user to modify the representation of spatial variability, hydrologic connectivity and hydrological processes (model structure and parameters) across the catchment by defining individual hydrological response units (HRU). HRUs are defined

TABLE 5.1. Transfer functions and required catchment attributed data for all DECIPHeR parameters. Global parameters are represented by g_x. A full description of the catchment attribute datasets is given in Table 5.2

Parameter	Parameter description	Catchment attribute data	Transfer function
$\ln(T_0)$	Lateral saturated transmissivity [$\ln(m^2\ ts^{-1})$]	Surface soil texture; percentage sand (s) percentage clay (c) percentage organic carbon (O_c) bulk density (D_b) hydrogeology productivity ($prod$). (Coxon et al., 2020; Cranfield University, 2020; Hollis et al., 2012, 2015; The James Hutton Institute, 2019)	$\ln(T_0) = \begin{cases} g_1 + (g_2 \cdot s) + (g_3 \cdot c), & O_c < 35\% \\ g_4 + (g_5 \cdot D_b^2) + (g_6 \cdot O_c^{-1}) + (g_7 \cdot D_b \cdot O_c), & O_c \geq 35\% \end{cases} + \begin{cases} g_8, & prod \text{ is high} \\ 0, & prod \text{ is moderate/low} \end{cases}$
SZM	Form of exponential decline in conductivity [m]	Uses the transfer function of $\ln(T_0)$ in each depth class (i) for transmissivity (T_i) applied at soil depths of; 0cm (x_0), 10cm (x_1), 25cm (x_2), 50cm (x_3), 100cm (x_4), and 150cm (x_5).	$SZM = \frac{e^{\ln(T_0)}}{\sum_{i=1}(x_i - x_{i-1}) \cdot e^{\ln(T_i)}} = \frac{T_0}{\sum_{i=1} T_i \cdot (x_i - x_{i-1})}$
Td	Unsaturated time zone delay [m ts^{-1}]	Same as $\ln(T_0)$	$Td = g_9 \cdot e^{\ln(T_0)}$
SRmax	Maximum root zone storage	Porosity (p) and Land-use (u). Global parameters are constrained using the rooting depth associated with different land-uses. (Rowland et al., 2017)	$SRmax = g_{10} \cdot p \cdot \begin{cases} g_{11}, & u = 1 \\ g_{12}, & u = 2 \\ g_{13}, & u = 3 \\ & \dots \\ g_{20}, & u = 10 \end{cases}$
Smax	Maximum effective deficit of saturated zone [m]	Porosity (p), soil depth (d_s) and hydrogeology productivity ($prod$). (Pelletier et al., 2016)	$Smax = g_{21} \cdot d_s \cdot p + \begin{cases} g_{22}, & prod \text{ is high} \\ 0, & prod \text{ is moderate/low} \end{cases}$

TABLE 5.2. Global parameters used to represent each model parameter. Upper and Lower bounds give the ranges each global parameter was sampled between.

Model parameter	Global parameter	Lower bound	Upper bound	Description
$\ln(T_0)$	1	-3.5	0.3	-
	2	0.006	0.03	Multiplier for sand
	3	-0.02	-0.0032	Multiplier for clay
	4	1	6	-
	5	-1.3	-0.5	Multiplier for bulk density ^2
	6	-0.1	-0.003	Multiplier for organic content ^-1
	7	-0.4	0	Multiplier for bulk density & organic content
	8	0	10	Addition for high productivity geology
SZM	-	-	-	As $\ln(T_0)$
Td	9	1	10	Multiplier for transmissivity
SRmax	10	0.000005	0.00005	Allows limit scaling
	11	150	250	Multiplier for broad-leaved woodland
	12	150	250	Multiplier for coniferous woodland
	13	75	175	Multiplier for cropland
	14	75	175	Multiplier for mixed grassland
	15	75	175	Multiplier for fen/marsh/swamp/bog
	16	25	225	Multiplier for heather
	17	5	150	Multiplier for rock
	18	0	300	Multiplier for water
	19	5	150	Multiplier for coastal sediment
	20	5	100	Multiplier for urban
Smax	21	0.000002	0.0001	Multiplier for porosity and soil depth
	22	0	4	Addition for high productivity geology
CHV	23	1500	6000	Model parameter
SRinit	24	0	0.01	Model parameter

by grouping raster-based information into non-contiguous spatial elements that share similar characteristics in, for example, landscape attributes (e.g. soil, topography or geology) and spatially varying inputs (e.g. rainfall). Each HRU then acts as a separate model store capable of having different spatial inputs, model parameter values and/or model structures to represent different and localised processes.

The DECIPHeR hydrological model lends itself well to exploring uncertainties within the MPR process. The separation of the catchment into HRUs enables the model to run quickly relative to a fully gridded spatially distributed model. This is essential for the GLUE approach for nationally applied domains, which requires large numbers of model simulations to explore the behavioural parameter space. Furthermore, DECIPHeR's distributed model parameters and flexible nature mean that MPR is a logical parameterisation choice. The link made between model parameters and catchment predictor variables enables spatially distributed parameters that indicate the differences in hydrological process between places. The ethos of MPR applying transfer functions at high resolution, and then upscaling to model resolution (i.e. averaging high resolution

parameter fields to produce model parameter fields at lower resolution) means the method is suitable for modelling at any spatial scale and parameters can easily be upscaled to distributed HRUs.

For this study, we split catchments into HRUs based on 3 classes of slope, 3 classes for accumulated area, 5 km rainfall grids and gauged sub-catchments. Whilst the choice of DECIPHeR model structure is flexible, we have applied the single model structure described in Coxon et al. (2019), which can be seen in Figure 5.1b, as the focus of this study is on model parameterisation . This model structure has been shown to be appropriate across many British catchments (Coxon et al., 2019). This structure has six main model parameters, which are also shown in Figure 5.1b, as well as one initialisation parameter. Many of these parameters are common to other hydrological models (for example, $\ln(T_0)$ is related to saturated hydraulic conductivity). Parameter values were spatially distributed across the HRUs.

To tailor MPR to DECIPHeR, we developed transfer functions and upscaling operators linking each model parameter to appropriate geophysical attributes. We first considered the physical function of each parameter, and identified geophysical attributes influencing that process. Transfer functions were then developed linking parameters to the selected attributes, based on pedotransfer functions in the literature where these were available. These were then tested across a range of catchments and modified to improve model performance. A summary of the model parameters and catchment attributes used for regionalisation is given in Table 5.1. An extended description of the catchment descriptor datasets is given in Table 5.2. The final transfer functions resulted in the seven model parameters being represented by 24 global parameters.

An example of the iterative development of the transfer functions is our decision to include hydrogeology as a catchment predictor variable. Initially, only soils information was used in the transfer functions, with no consideration of the underlying geology. Results showed poor performance using regionalised parameters for many groundwater dominated catchments, with the regionalised parameters unable to capture the correct hydrograph slope, variance or peaks, despite good model performance when the parameters were directly constrained using Monte Carlo sampling. Investigation into possible causes for this disparity found that many of these catchments shared a common characteristic – the presence of high productivity hydrogeology (see Appendix E for more information). When parameters were directly constrained for catchments with large areas of productive hydrogeology, the $\ln(T_0)$ and Smax parameters tended to be very high, representing large saturated zone stores with high saturated conductivity values. The soils information being used to constrain regionalised $\ln(T_0)$ and Smax did not contain any information to differentiate the areas with productive hydrogeology, and therefore these very different processes were not being reflected in the model parameters. To resolve this, we included a hydrogeology map into the regionalisation, and included extra coefficients in the transfer

functions to allow areas of high productivity geology to have increased ln(T_0) and Smax values (see Table 5.1). This led to improvements in model performance across many areas with high productivity geology, whilst not impacting the regionalisation elsewhere.

Model input data for each catchment consisted of 26 years of daily rainfall and potential evapotranspiration data covering the period Jan 1985 – Dec 2010. The CEH-GEAR (Tanguy et al., 2014) rainfall product was used, which is based upon the national database of rain gauge observations collected by the UK MetOffice, converted to a 1 km^2 gridded product using the natural neighbour interpolation methodology (Keller et al., 2015). We aggregated this product to 5 km^2 grids for input into DECIPHeR. The CHESS-PE dataset was used to produce PET estimates for each catchment (Robinson et al., 2015a). These estimates were produced using the Penman-Monteith equation, calculated using meteorological variables from the CHESS-MET dataset (Robinson et al., 2015c) and are available as a 1 km^2 gridded product which we aggregated to 5 km^2 grids to match the rainfall. Daily observed discharge data from the National River Flow Archive (NRFA) was used to evaluate model performance where available (Centre for Ecology and Hydrology, 2016). The first six simulation years (Jan 1985- Dec 1990) were used as a warmup period, the following 20 years were split equally into the calibration (Jan 1991- Dec 2000) and evaluation (Jan 2001 – Dec 2010) periods.

5.3.3 Evaluating the national parameter regionalisation method

A large sample of 437 catchments across Great Britain were used for model parameterisation and evaluation. These catchments cover a range of catchment attributes and hydrological processes and are distributed across Great Britain as can be seen in Figure 5.1a. Using a large sample of catchments ensures results are robust and generalisable, and the dense coverage across Great Britain means they can be considered as a representative set of catchments for a national model calibration.

DECIPHeR was run and evaluated across all catchments using three different parameterisation approaches, (1) a national implementation of MPR (behavioural sets of global parameters constrained across all gauges simultaneously producing nationally consistent simulations), (2) an individual catchment implementation of MPR (behavioural sets of global parameters constrained and applied per catchment), and (3) the model applied using homogeneous parameter values calibrated through Monte-Carlo parameter sampling (see Coxon et al. 2019). This enables us to evaluate the national parameter regionalisation approach against two upper benchmarks for each catchment where model parameters are less constrained. It also allowed evaluation of whether the distributed parameters produced using MPR resulted in better model output than homogeneous parameter values across a catchment.

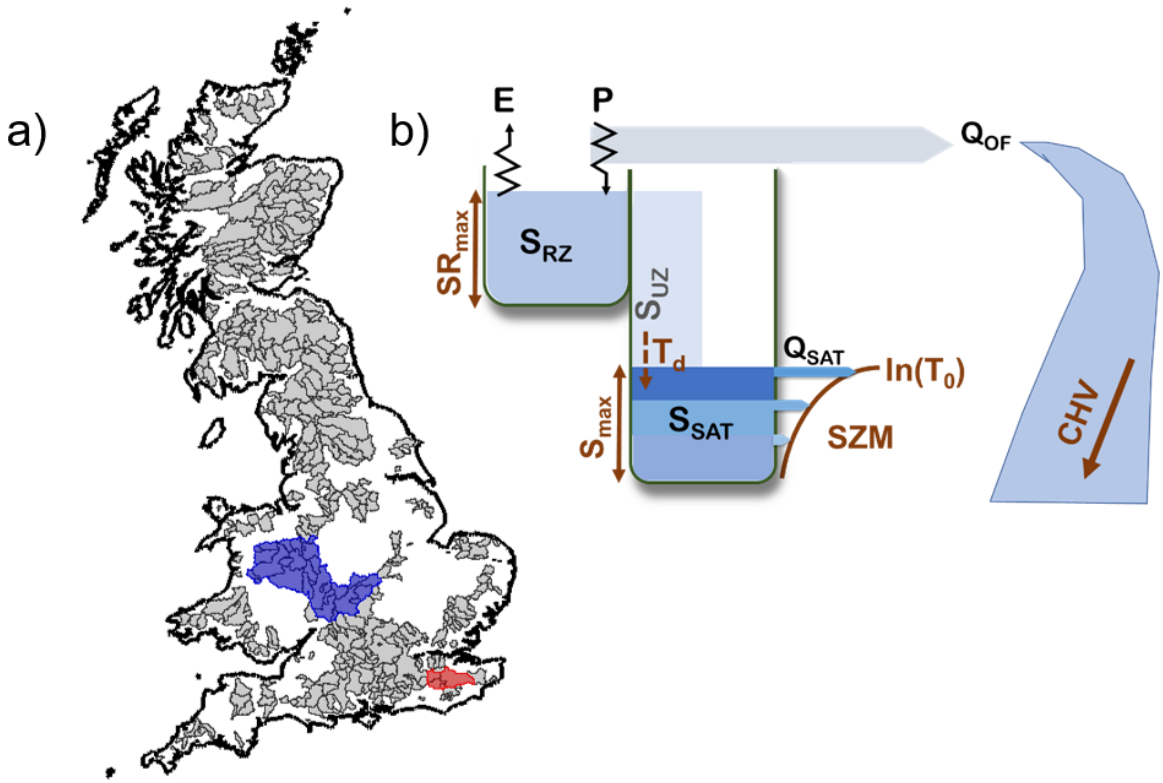


FIGURE 5.1. a) Map showing the 437 catchments used in this study. Highlighted catchments are the Severn at Haw Bridge, gauge 54057 (blue) and Medway at Teston/ East Farleigh, gauge 40003 (red). b) Model structure diagram for the DECIPHeR hydrological model which was applied across these catchments.

For each parameterisation approach, DECIPHeR parameters were constrained within the GLUE framework to produce prediction uncertainty bounds reflecting uncertainties in the global parameters (1&2) or the model parameters (3). In each case, 3750 model simulations were carried out per catchment, sampling the model parameters/global parameters between set bounds (see Appendix D for bootstrapping tests of the number of parameter samples). The model simulations were evaluated using non-parametric KGE (referred to as KGE* throughout) for the catchment parameterisation approaches, and an average of KGE* across all catchments for the national parameterisation approach (Pool et al., 2018). The non-parametric KGE is a newly introduced goodness-of-fit measure (Pool et al., 2018) which builds upon the widely applied Kling-Gupta Efficiency (KGE) metric (Gupta et al., 2009). The KGE considers three types of model errors; error in mean flow, error in flow variability and correlation between observed and simulated flow. By combining multiple objectives KGE aims to improve calibration by preventing overfitting to a particular hydrograph element. However, KGE is based on assumptions of data normality and linearity, which is often not the case for model simulation errors. Pool et al. (2018) therefore

proposed a move towards non-parametric components of KGE, and tested a modified KGE version which reformulated the variability and correlation components of KGE using non-parametric alternatives. This non-parametric KGE uses the Flow Duration Curve (FDC) to indicate variability, as opposed to the standard deviation, and the Spearman rank correlation, as opposed to the Pearson correlation coefficient. Testing of models calibrated with the standard and non-parametric KGE across the contiguous United States demonstrated similar performance for high flows, but a general improvement in simulation of low flows using the non-parametric version (Pool et al., 2018).

The non-parametric KGE (KGE*) was calculated separately for the calibration (Jan 1991- Dec 2000) and evaluation (Jan 2001 – Dec 2010) periods. To produce the average KGE*, catchments producing maximum KGE values below 0.3 were excluded as it was likely that the model structure was not suitable for these catchments and including them would be disinformative. The median KGE* value from all remaining catchments was calculated, assigning equal weighting to all gauging stations. The top 100 simulations were selected as the behavioural set of simulations for each approach. The variability in these top 100 simulations was then used to demonstrate the uncertainty surrounding the best simulation. Model skill was then evaluated using the KGE*, and the three decomposed metrics of bias, error in the variability of flows (based on the normalised flow duration curve), and spearman rank correlation (see Pool et al. 2018).

5.4 Results

5.4.1 Evaluating the model parameterisation across a large-sample of catchments

Figure 5.2 compares DECIPHeR model performance with parameters constrained using the three different methods: MPR parameter fields with nationally consistent global parameters (national MPR); MPR parameter fields with global parameters constrained per catchment (catchment MPR); and homogenous Monte-Carlo sampled parameters for each catchment (catchment MC). The performance of the top 100 parameter fields is shown for all methods, to indicate the uncertainty in model performance due to selection of the best parameter field. It is worth noting that the catchment-based approaches are likely to produce better results, because the parameters have been constrained against the same gauge data used for model evaluation. They are therefore being used as a upper benchmark for the national approach, showing what the model can achieve for each catchment given the model structure and data available. The national MPR approach is the only method which produces consistent parameter fields across all catchments including ungauged areas.

Figure 5.2a-c shows results for the KGE* metric, calculated for the model calibration and

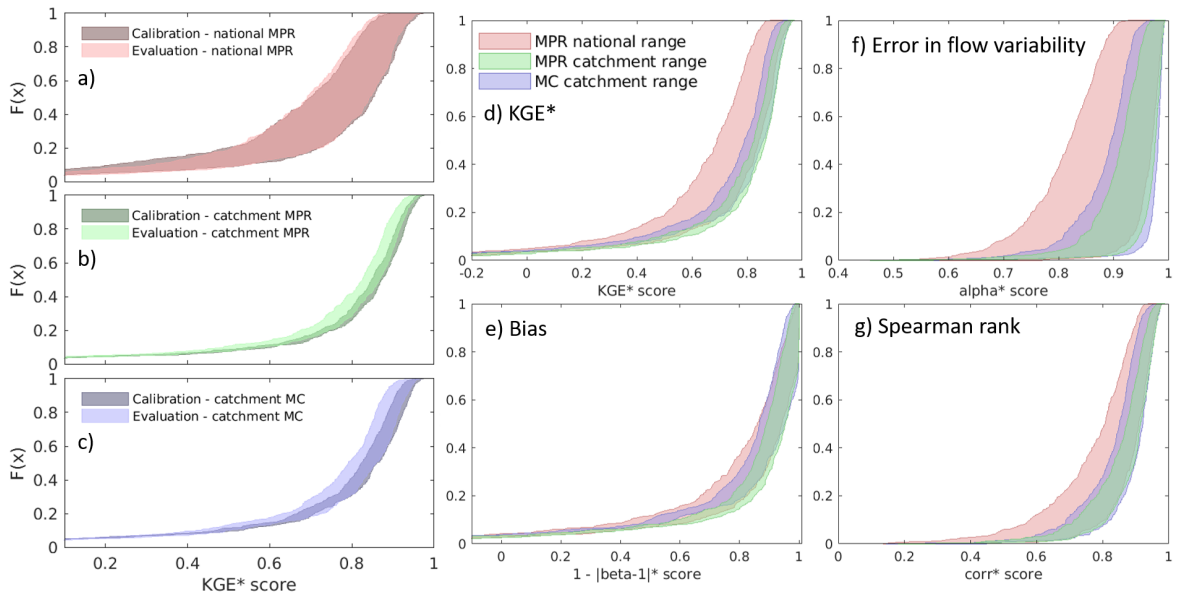


FIGURE 5.2. Empirical cumulative distributions (CDFs) of model performance across all study catchments. The shaded regions shows the range in non-parametric KGE (KGE*) scores over the best 100 global parameter sets. Plots a-c compare KGE* scores between the calibration and evaluation periods, represented by darker and lighter shaded areas respectively, for each parameterisation approach. Plots d-g compare model performance scores over the evaluation period between the parameterisation approaches. The four performance scores are KGE* and its three components. Bias is transformed so that for all scores higher values indicate better model performance.

evaluation periods. The cumulative distribution plots show the modelled KGE* across all catchments, with the shaded area showing the range between the 100 best (global) parameter sets. It can be seen that the national MPR results show much larger uncertainty bounds than the catchment calibration methods. This is expected, as different global parameter combinations perform well in different catchments and the national parameterisation needs to reflect all catchments simultaneously. However, the performance and uncertainty bounds with the national MPR approach are consistent across the calibration/evaluation period – indicating that they are stable over time and that the global parameters have not been over-constrained on the data. By contrast, both the catchment calibration approaches show a general widening of the uncertainty bounds and decrease in performance when moving between the calibration/evaluation periods. This is most likely due to the model overfitting to observed flows.

Figure 5.2d-g directly compares performance of the three parameterisation approaches, evaluated using the KGE* and decomposed metrics over the model evaluation period. The

beta (bias) metric has been transformed, so for all metrics higher values refer to better model performance, with a maximum score of 1. The national MPR approach results in decreases of performance and larger uncertainty bounds for all metrics, apart from bias in the top 50% of catchments. However, considering that this is producing consistent parameter fields across all catchments, the decrease in performance is not as large as expected. There are large overlaps between the national simulation performance bounds and those of the catchment constrained parameters. Comparing the catchment approaches, it can be seen that the distributed regionalised parameters (catchment MPR) outperform the homogenous parmeter fields (catchment MC) for KGE*, and also have smaller uncertainty bounds across all metrics. This suggests that the distributed parameters produced by the MPR approach are improving model simulations, and the parameters are better constrained.

5.4.2 Performance variation between catchments

Figure 5.3 shows that the spatial pattern of performance is similar between the three parameterisation approaches (Figure 5.3a-c). Poor performance for catchments in Scotland, and Southeast England could be due to missing processes within the model structure (e.g. snowmelt, inter-catchment groundwater flows), or because catchments in the southeast are generally drier and more difficult to model, as found in Research Chapter One. All approaches produce good simulations for catchments to the west of Great Britain, and in northern England. These catchments are wetter, and previous studies have shown that a range of hydrological models produce better results in the west Coxon et al. (2019); Lane et al. (2019); Rudd et al. (2017); Seibert et al. (2018). This indicates that parameterisation is likely to be less important for these catchments: as the simple relationship between rainfall and runoff can be represented by a range of different parameterisations.

Figure 5.3d and e show the differences in KGE* score between the regionalised (MPR) and directly constrained (Monte Carlo) parameterisation approaches. The catchment MPR approach shows large performance gains relative to the catchment Monte Carlo parameterisation in the southeast, and similar model performance elsewhere. This shows that the MPR approach has great potential, and the distributed parameters are better able to reproduce flows for many catchments. A performance drop is expected for the national MPR, since the parameterisation is not tailored to a single catchment, yet for most catchments this performance drop is small (82% of catchments drop by less than 0.1 KGE*, and 20% show no reduction in performance). This could be attributed to the national calibration producing parameter sets that are more robust between the calibration/evaluation periods, and the added benefit of distributed parameter fields when comparing to the lumped parameters used in the catchment MC approach.

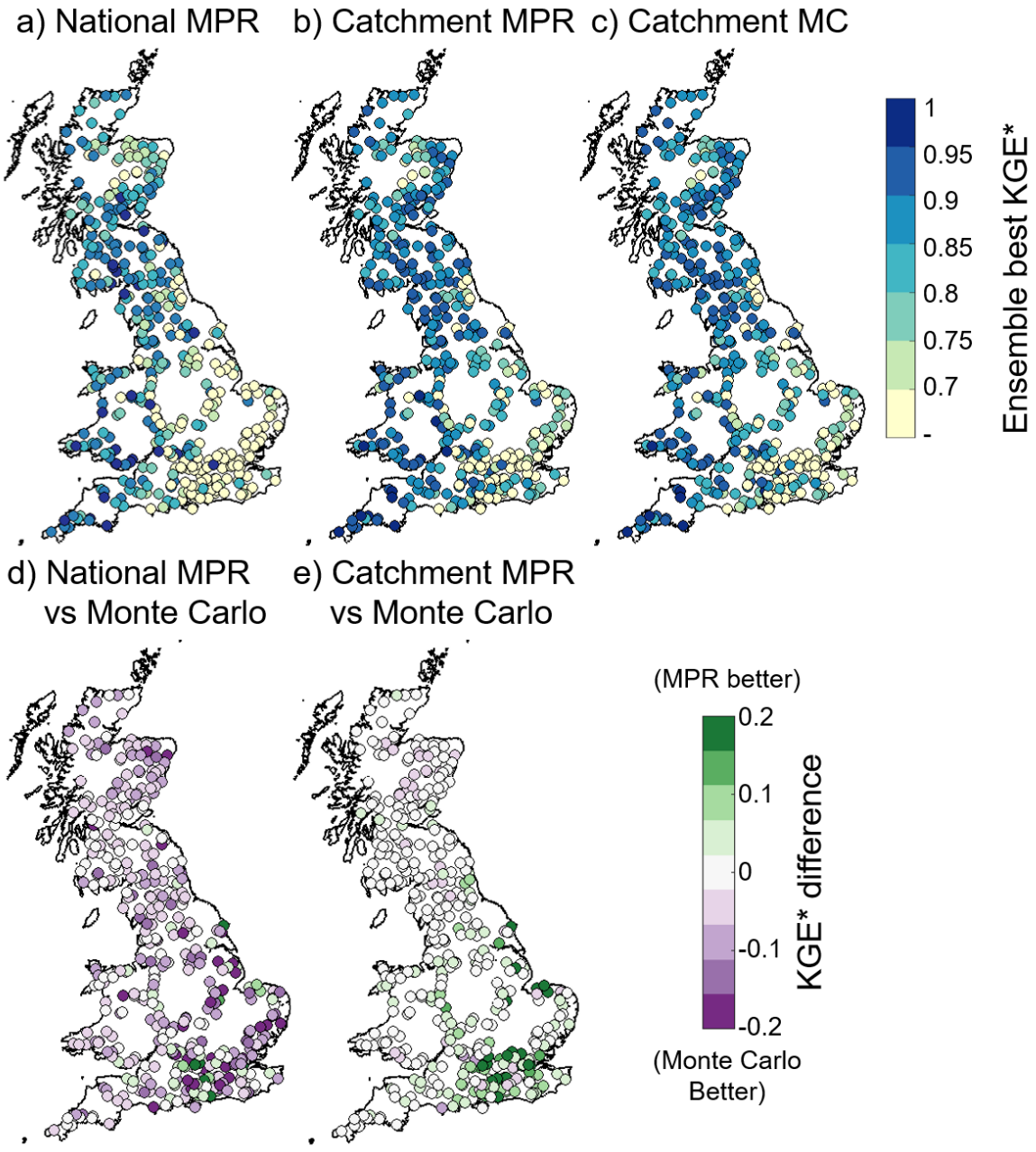


FIGURE 5.3. Spatial maps of model performance using three different parameterisation approaches (a-c), and performance difference between regionalised and directly constrained parameters (d-e). Each dot shows performance at a gauging station within the evaluation period.

5.4.3 Example catchment showing performance improvement using MPR

To further understand the differences between the regionalised and directly constrained parameters, we investigate an example catchment in more depth (as highlighted in Figure 5.1a). The Medway at Teston/ East Farleigh was selected (gauging station 40003), as it is a relatively large and varied catchment in the southeast of England where differences between the parameterisation approaches are pronounced. It has an interesting flow regime, with aquifer-fed springs providing baseflow for the headwaters combined with a responsive overall regime. For this catchment the regionalised parameters were able to better reproduce the observed flows, with KGE* scores over the evaluation period in the range 0.54-0.58 , 0.63-0.67, and 0.46 – 0.64 for the Monte Carlo, catchment MPR and national MPR parameterisation approaches respectively.

Figure 5.4 shows the best-performing parameter fields from the national-MPR and catchment Monte-Carlo approach, alongside a selection of the catchment characteristics data used to produce them. The parameter values from all 100 behavioural parameter fields are given in Figure 5.5. Here, each line shows the distribution of parameter values across the catchment in one parameter field, coloured by model performance when using that parameter field. The black dashed lines show values of the best parameter field, linking to the spatial maps given in Figure 5.4. It is worth noting that for the national MPR approach, the best parameter field refers to the parameter field for all gauges nationally, and so does not fully align with the best parameter field for the Medway. The Monte Carlo catchment parameter fields are all shown as vertical lines, as parameters are distributed homogeneously across the catchment.

The MPR parameter fields result in large ranges in the SZM, $\ln(T_0)$ and Smax parameter values across the catchment. These ranges are linked to the underlying catchment properties, for example storage in the saturated zone (controlled by Smax) is larger for areas with deep soils, and saturated hydraulic conductivity (controlled by $\ln(T_0)$) is higher for areas with highly productive geology. The SRmax (soil root zone storage) parameter is particularly important for this catchment, and for the MPR approach increasing SRmax improves model performance. A larger soil root zone store may result in increased evaporation losses, as evaporation can only be taken from the root zone store and not the saturated or unsaturated zone stores. As it is situated in the south-east, the Medway catchment has relatively high annual losses for a GB catchment and this could be further exacerbated by abstractions and reservoir storage leading to additional evaporation losses. These attributes may explain why higher SRmax values lead to improved performance for the Medway catchment, in contrast to the national calibration where the best SRmax value is much lower.

Flow timeseries over winter 1993/4 for all parameterisation approaches are given in Figure 5.6, with the shaded area showing the GLUE simulated discharge bounds from the 100

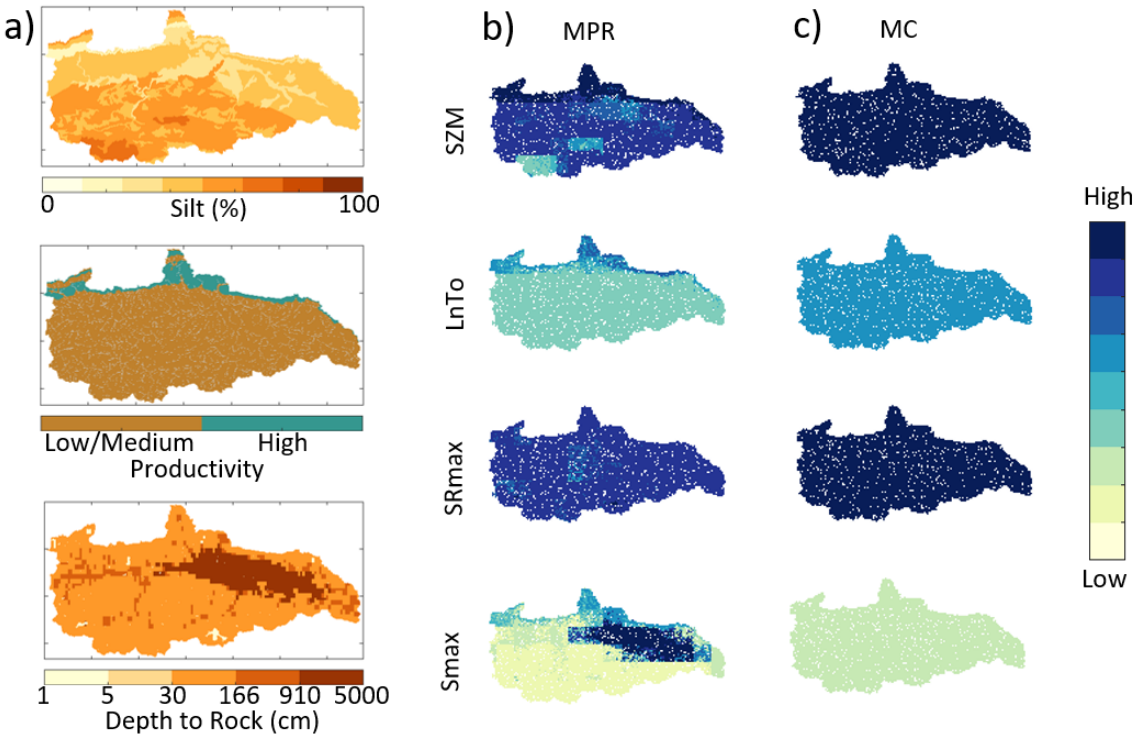


FIGURE 5.4. Selected catchment attribute data (a) and resultant parameter fields (b/c) for the Medway at Teston/ East Farleigh. Parameter maps show the best performing parameter fields generated using spatially distributed national MPR (b) and lumped catchment Monte-Carlo sampled parameter fields (c). The colour scale indicates the relative value of each parameter, between the bounds defined in Figure 5.5.

behavioural parameter fields. There are clear differences between the hydrographs resulting from the distributed MPR parameters. For the homogeneous Monte Carlo parameterisation the model is not able to represent either the baseflow or flow peaks. There are large uncertainty bounds and the model is not able to capture observed flows. The regionalised parameters are better able to capture the hydrograph, as the distributed parameters can capture both the general baseflow and fast response. Focusing on the peak labelled 1, both MPR approaches capture this event, albeit with large uncertainty bounds for the national parameterisation, whilst the Monte Carlo parameters miss the peak entirely. It is possible that the following event, labelled 2, is strongly controlled by the flood storage reservoir within the catchment, resulting in the rounding off of the flood peak, which we would not expect to be replicated by the model as it does not include human influences on the flow regime.

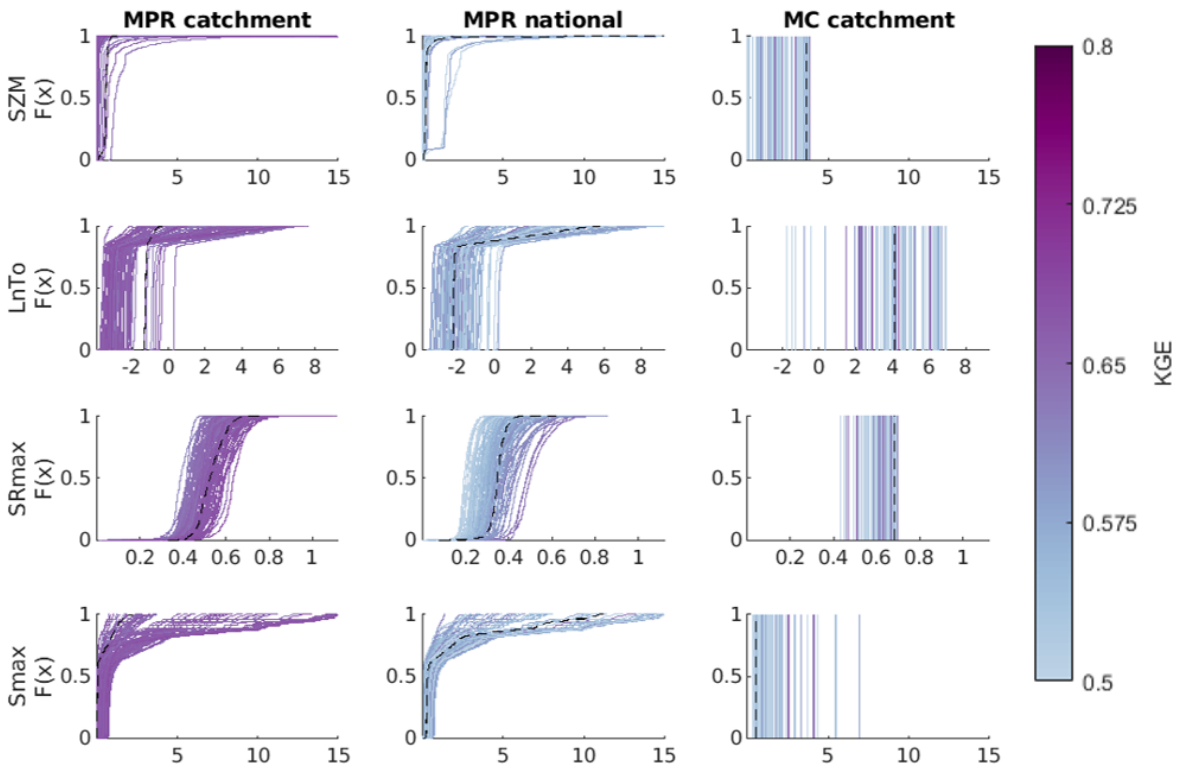


FIGURE 5.5. Distribution of parameter values for all behavioural parameterisations across the Medway at Teston/ East Farleigh. The three columns show results from different model parameterisation approaches, and each row shows a different model parameter. Each individual line gives the cumulative distribution of parameter values across the catchment from one behavioural parameter field, with the line colour showing model performance with that parameter field. All Monte-Carlo (MC) parameter fields are given as vertical lines, as parameters are applied homogenously across the catchment with no spatial variation.

5.4.4 Performance variation across sub-catchments

A key benefit of parameter regionalisation is the ability to create spatial parameter fields. This assumes that the spatial distribution of our catchment attributes is informative in setting the spatial distribution of our model parameters. If this assumption is correct, then we could expect better simulations within a catchment when using the spatial parameter fields generated using MPR, as sub-catchments would have relevant parameter values and important hydrological differences across a catchment would be represented. We have tested this, by evaluating model performance across four gauged sub-catchments within a large catchment, when the catchment parameterisation approaches have been constrained only to flows at the principal outlet.

The Severn at Haw Bridge was selected because it is large (with a catchment area of 9895

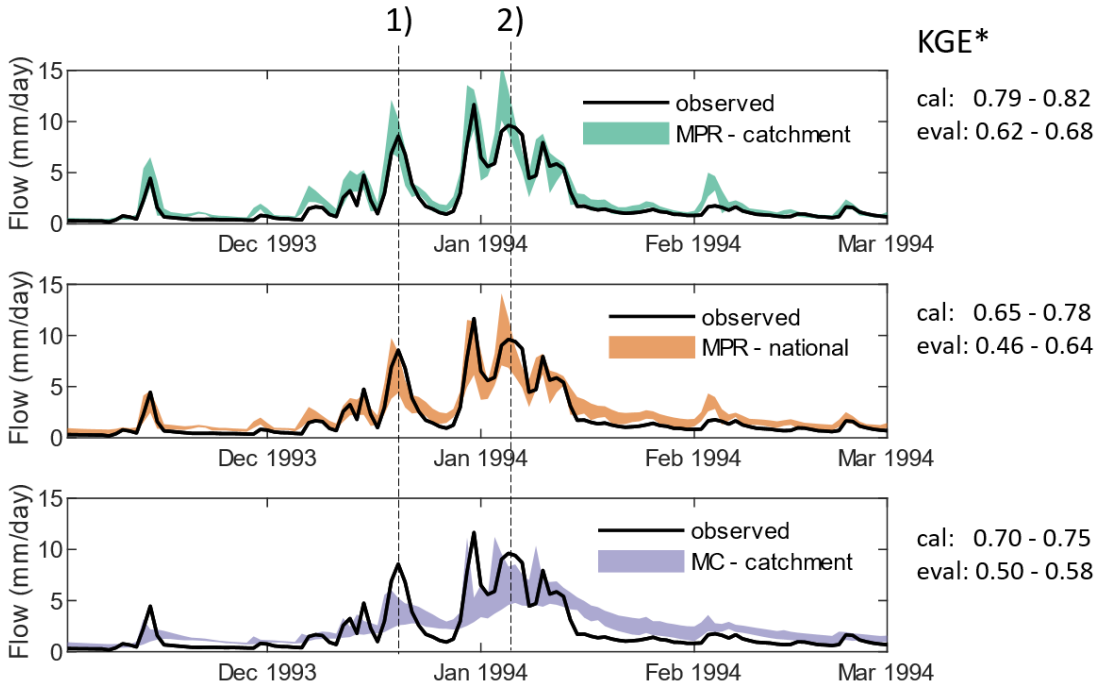


FIGURE 5.6. Example hydrographs for the Medway at Teston/ East Farleigh, produced using the three different parameterisation techniques. The shaded area gives the simulated hydrograph uncertainty bounds produced using the GLUE technique, whilst the solid black line shows observed flows. Two events 1) and 2) have been highlighted, as these are discussed in section 5.4.3.

km^2 it is the second largest catchment in the UK), it includes over 60 gauged sub catchments, and shows a diverse range of hydrological behaviour. This includes very wet, high elevation catchments in the west, baseflow dominated catchments with highly productive geology in the northeast, and drier, flatter catchments to the east (see Figure 5.7). Four sub-catchments were chosen to represent contrasting areas within the Severn catchment. The location of the Severn catchment within the UK is given in Figure 5.1a, and locations of the selected sub-catchments are shown in Figure 5.8.

Figure 5.7 gives examples of the parameter fields produced for the Severn, and Figure 5.8 shows hydrographs at four gauged points within the larger Severn catchment. Compared to Monte Carlo parameter sampling, the national MPR approach produces better hydrographs for sub-gauges within the Severn. The regionalised hydrographs are both better able to reproduce observed flows and importantly have smaller GLUE simulation bounds. Neither approach can capture flows well at gauge 54052 for high flows. This is made evident by the wide uncertainty

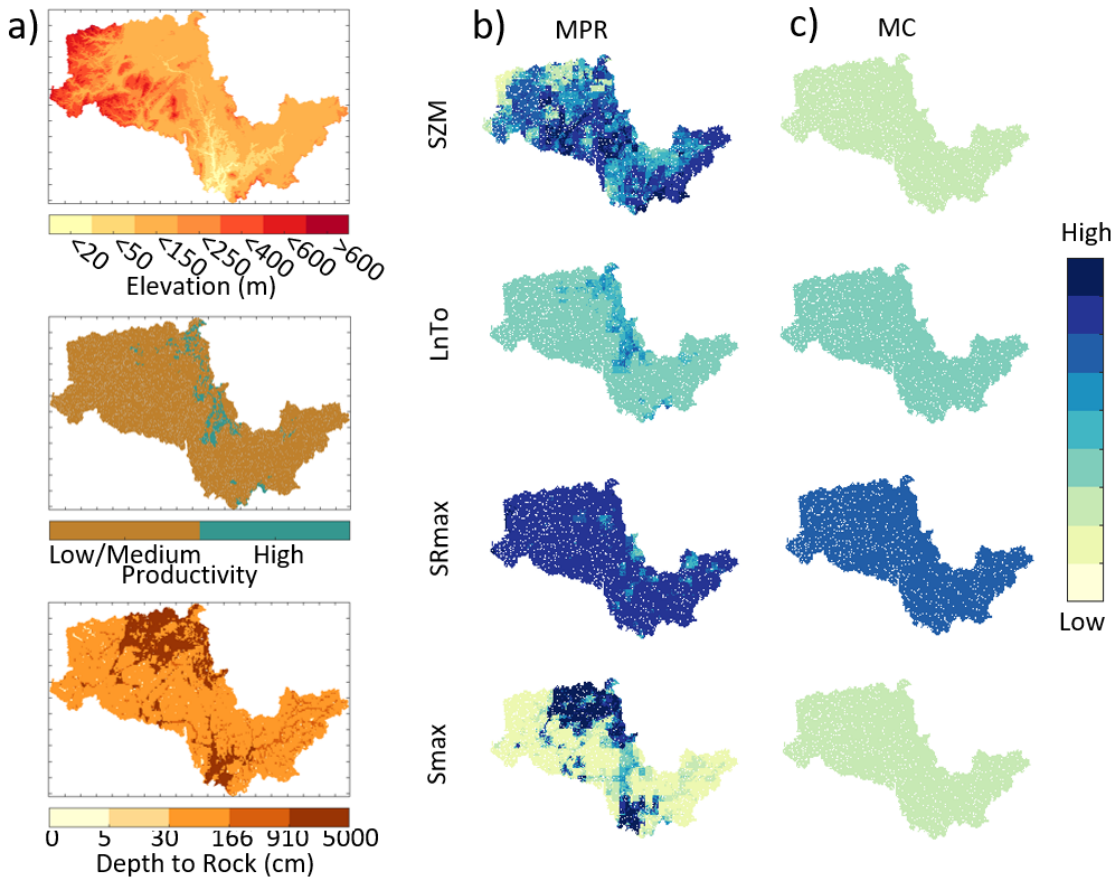


FIGURE 5.7. Selected catchment attribute data (a) and resultant parameter fields (b/c) for the Severn at Haw Bridge catchment. Parameter maps show the best performing parameter fields generated using spatially distributed national MPR (b) and lumped catchment Monte-Carlo sampled parameter fields (c). The colour scale indicates the relative value of each parameter.

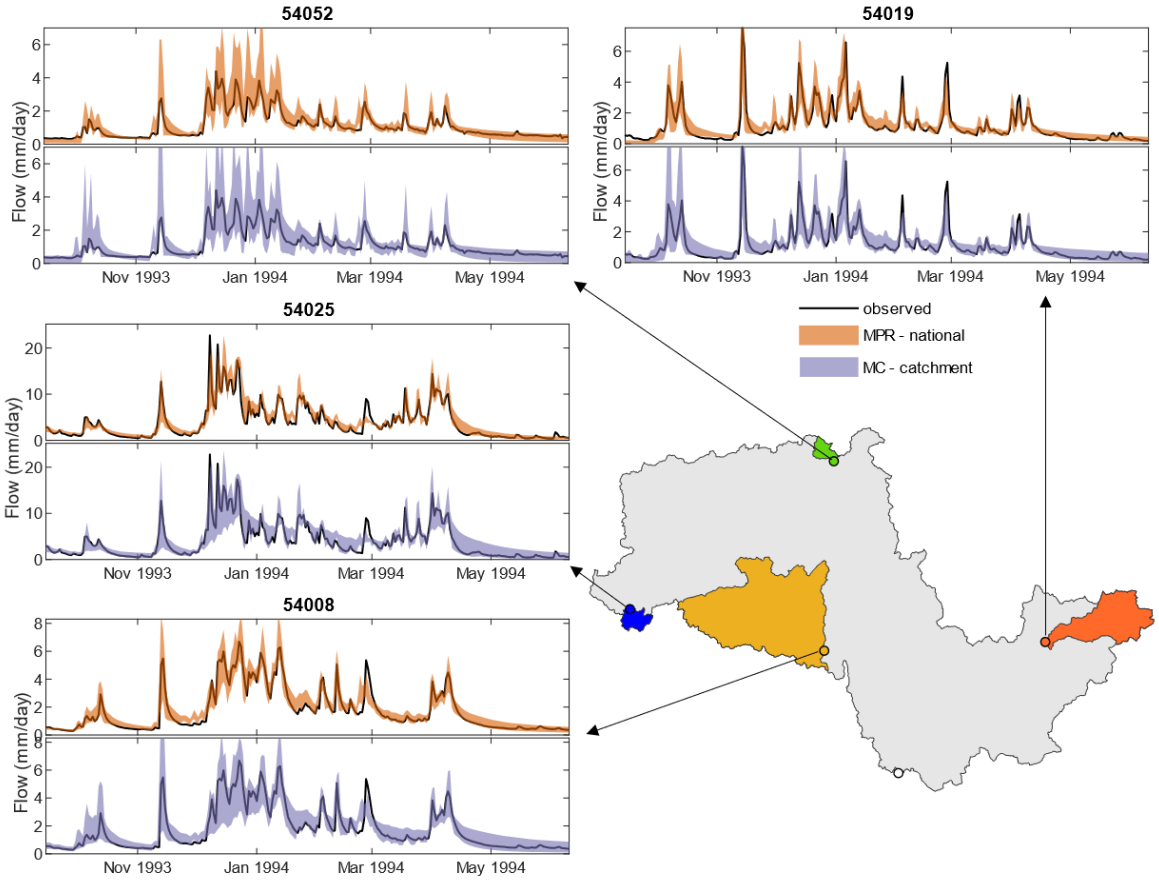


FIGURE 5.8. Example hydrographs for sub-catchments within the Severn at Haw Bridge catchment. Hydrographs are given for the national MPR (orange) and Monte-Carlo catchment (purple) parameterisation techniques, where the catchment-based parameters have been constrained on the larger Severn at Haw Bridge catchment. For both approaches, the shaded hydrograph area gives the simulated uncertainty bounds at the sub-catchment gauge produced using the GLUE technique, whilst the solid black line shows observed flows. A map has been provided to show the location of each sub-catchment within the Severn catchment, and the gauging station at Haw Bridge has been included as a white circle.

bounds, but would have been missed if simulations were carried out deterministically. For the other gauges, the MPR approach is both better at capturing peaks (e.g. the Dec/Jan event has very wide uncertainty bounds underpredicting flow for the Monte Carlo simulations) and recessions (e.g. May 1994 both parameterisation approaches encompass the observed recession, but the MPR approach has smaller uncertainty bounds). This supports the idea that the distributed parameter fields produced using MPR are improving simulations within a catchment relative to the Monte Carlo homogeneous parameter fields. Further, this result strongly increases confidence in the ability of nationally calibrated MPR to predict flow with realistic uncertainty estimations in ungauged areas.

5.5 Discussion

5.5.1 Adding parameter uncertainties to multiscale parameter regionalisation

Here, we have shown how MPR can be applied within an uncertainty framework to incorporate uncertainties in model output resulting from choice of global parameter values. The global parameter values control the relationship between the underlying catchment geophysical data and the model parameters. There are many reasons why there may be uncertainty surrounding the global parameter values and a single set of global parameters is not able to define the link between model parameters and geophysical data everywhere. These include: 1) errors in both the geophysical data used to define model parameters, and the observed discharge used to evaluate model parameters, varying in space and time; 2) a lack of knowledge about the nature of the link between model parameters and the geophysical data, especially for conceptually based models where parameters many not directly link to measurable physical processes; 3) missing processes/information required to model everywhere means that different global parameter values will be more/less effective for different catchments; 4) uncertainties in model parameter values. While we do not have the information required to further constrain the global parameter values, it is important to understand the implications of this uncertainty on simulated discharge values. For example, this is highlighted in Figure 5.8 where large uncertainty bounds for gauge 54052 demonstrate that the model is less able to capture events in this part of the catchment compared to other nearby gauges 54025 and 54008.

We found that the overall range in model performance was largest for the national MPR approach. This is because the national parameter fields need to satisfy data from all catchments simultaneously, and the best sets of global parameter values varies between catchments. However, despite the larger range in performance, the uncertainty bounds of the flow timeseries are of a similar magnitude (e.g. Figure 5.6).

There are many other uncertainties within the MPR process, that were beyond the scope of this research. For example, decisions regarding the form of the transfer function linking parameters to geophysical spatial data (Klotz et al., 2017), the upscaling function used to transform high resolution parameter maps to parameter fields at the model resolution and which geophysical datasets to include for each model parameter (Mizukami et al., 2017).

5.5.2 Evaluation of MPR as a parameterisation strategy for a national model

National parameter fields have many advantages over catchment-calibration using the Monte Carlo or MPR approaches. First, they are spatially consistent within and between catchments, meaning that national model runs are consistent across space without discontinuities between catchment or sub-catchment boundaries (Mizukami et al., 2017; Samaniego et al., 2017). Secondly, they can be used to run hydrological models for ungauged areas and provide better parameterisations for ungauged points within a catchment (Kumar et al., 2010). Here, we have shown that for the case of the river Severn, the distributed parameters result in better reproduction of hydrographs for sub-gauges. Thirdly, having an ensemble of behavioural national parameter fields means that additional catchments could be modelled with no need to recalibrate the model. Fourth, regionalising parameters simultaneously across many catchments is less likely to result in model overfitting to observed flows. Here we found that the national parameter fields resulted in more consistent model performance between the calibration and evaluation periods.

National parameter fields are expected to lead to reductions in performance relative to catchment calibration, as they are solving the more difficult problem of constraining model parameters everywhere rather than constraining model parameters to observed flows for a single catchment (Mizukami et al., 2017). In some catchments flow observations may be biased, flows may be heavily modified by artificial influences that are not included in the model or there may be errors in observed precipitation/PET data used to drive the model (Mcmillan et al., 2012). In these circumstances, the catchment constrained parameter values will always do better because the parameters will be constrained based on that scenario (e.g. a model storage parameter may be unrealistically high to account for the effects of a reservoir within a catchment). Whereas in the national case, parameters are constrained to all catchments simultaneously so such tailoring to individual catchments is not possible.

We found that overall the ensemble of national parameter fields created using MPR (national MPR) were able to produce good results (KGE* > 0.75) across the majority (60%) of Great Britain. Catchments in the southeast presented the largest challenge to the DECIPHeR hydrological model, and it is here that the largest differences in performance (both positive and negative) were seen between the national and catchment-constrained parameters. Whilst most catchments showed small decreases in performance when using the national parameter fields relative to the

catchment Monte Carlo approach as expected, there were 358 catchments where performance was not substantially impacted (KGE* difference <0.1) and 88 catchments where performance increased. This is a very promising result, indicating that there is not always a compromise between model performance and the benefits of having nationally consistent parameter fields.

There are still many areas for improvement and further research within the multiscale parameter regionalisation methodology. Firstly, the question of how best to select catchments to include in the national regionalisation. Here, we excluded all catchments with low KGE* values, as it was considered that these were not informative, but gave all remaining catchments equal weighting. This is similar to the approach used in previous studies Mizukami et al. (2017). Future work could explore the impact of catchment selection on model performance, and screening of catchments which are potentially disinformative for the national picture (e.g. those with significant artificial influences). Secondly, a key issue in sampling national parameter fields is that not all combinations of catchments attributes exist in all catchments. An example of this is constraining the SRmax parameter using land-use – the complete range of land-uses will not exist for every catchment and therefore global parameters linked to missing land-uses will not be constrained. This is problematic for our national calibration, as it could lead to some global parameters not being properly constrained. Future work could look at how to resolve this issue when selecting catchments for MPR. Thirdly, the datasets used to regionalise model parameters and the structure of transfer functions linking model parameters to data, often requires an ad-hoc procedure and intensive testing. We have shown how this is an evolving process, with the step-wise addition of datasets found to lead to increases in model performance. Therefore, whilst we have identified catchment attributes that lead to reasonable national simulations, this national parameterisation has much scope for improvement as further catchment attributes are found to be important in constraining model parameter fields.

MPR was initially developed as a module for the mesoscale Hydrologic Model (mHM) (Kumar et al., 2013b; Samaniego et al., 2010), and the regionalisation was tied into the mHM code. Recent studies have focused on developing a model agnostic version of MPR, providing stand-alone regionalisation code that can be retrofitted to any hydrological / land-surface model (Mizukami et al., 2017; Samaniego et al., 2017). Our work feeds into this goal, as whilst we have focused on applying MPR to DECIPHeR, the general methodology for incorporation of uncertainties is applicable independent of hydrological model choice. Furthermore, there is crossover between DECIPHeR parameters and parameters of many conceptual models, meaning the transfer functions developed here are more generally applicable.

5.5.3 Lessons learnt from applying MPR to Great Britain

MPR is a data intensive parameterisation strategy, requiring (multiple) high resolution datasets to underpin each model parameter. Here, we have applied MPR to a new hydrological model and a new region, and our experiences developing links between model parameters and catchment attribute data will be useful for future studies. We have provided information on the datasets and pedotransfer functions used (see Table 5.1 and Section 3.5), which will be informative for any future study applying MPR for the UK and elsewhere.

We found that peaty soils, which are abundant in Scotland, required the development of a different approach to regionalisation. Whilst the UK is a data-rich country, we found that soil sand and clay content data were not available everywhere, and in particular are often not measured for peaty soils. These data are required for most pedotransfer functions for saturated hydraulic conductivity (Zhang and Schaap, 2019). While small areas of missing data could be gap-filled this was not appropriate for peaty soils which covered very large areas and were expected to have distinctly different hydrological characteristics. We therefore applied different pedotransfer functions to estimate saturated hydraulic conductivity in peat soils, which were based on bulk density and organic content rather than sand and clay (see Table 5.1).

During testing of which catchment characteristics were important to use in the pedotransfer functions, it was found that hydrogeology played a key role in the values of behavioural global parameters. Areas with highly productive geology were found to have behavioural global parameter values in a very different range to areas with low/medium productivity geology. We therefore introduced a hydrogeology dataset into the regionalisation strategy, and included additional global parameters for areas of high/low productivity. This resulted in a large improvement in performance in catchments containing high productivity geology, whilst not affecting the parameter regionalisation elsewhere. The difference in constrained parameter values between areas with high/low productivity geology can clearly be seen in our results (Figure 5.4, Figure 5.7).

5.6 Summary and conclusion

The field of hydrological modelling is moving towards large-scale models that can be applied across national to continental scales. For these models to represent the diverse range in hydrological behaviour across and within catchments, and across complex landscapes with multiple nested sub-catchments, we need to be able to generate spatial fields of model parameters. These model parameter fields will always be uncertain as they cannot be directly measured and must be estimated using transfer functions and catchment attribute data, both of which are sources of uncertainty. It is therefore important that these uncertainties are recognised and quantified within modelling studies, so that we can understand the robustness of our model predictions.

Here, we have shown how the multiscale parameter regionalisation (MPR) method can be used to generate nationally consistent spatial parameter fields for the DECIPHeR hydrological model across Great Britain. We have extended the method to consider uncertainties in the links made between model parameters and underlying geophysical datasets, to produce ensembles of equally plausible model parameter fields. This enables us to simulate river flows in a nationally consistent way producing river flow estimates with uncertainty bounds.

The performance of the nationally constrained parameter fields were compared with two catchment-based parameterisations: (i) individual catchment calibration using MPR to demonstrate the best possible performance of the DECIPHeR-MPR method for a particular catchment, and (ii) calibration using Monte-Carlo (MC) parameter sampling to demonstrate the best performance of DECIPHeR with homogeneous model parameters. The national parameter fields facilitate good model simulations (non-parametric KGE scores exceeding 0.75 for 60% of catchments), despite a general decrease in performance compared to catchment constrained parameters (median decrease of -0.06 KGE*). The national parameter fields were shown to have advantages over the catchment-constrained Monte-Carlo parameters including; more realistic parameter fields leading to better simulations at points within catchments, more robust performance between calibration and evaluation periods, and national consistency including parameter fields for ungauged areas. We also show that the catchment-based MPR technique outperforms MC sampled parameters across a range of metrics, demonstrating the potential to improve national MPR approach further. Finally, by including uncertainties for the first time we show that the parameters are better constrained using MPR: MPR-catchment results in more consistent performance than the MC approach across multiple metrics and the hydrographs produced using both catchment and national MPR have smaller uncertainty bounds.

Defining transfer functions to link parameters to suitable catchment attributes is challenging when regionalising a hydrological model. Crossover between the parameters of the DECIPHeR model and many other hydrological models, means the transfer functions and catchment attributes used here are more generally applicable. As the first application of MPR to the DECIPHeR hydrological model, and the first application of MPR focused on Great Britain, the methods used within this study will be informative for future regionalisation efforts in Great Britain and elsewhere.

CHAPTER 6

CLIMATE CHANGE IMPACT ON HIGH FLOWS ACROSS GREAT BRITAIN, INCLUDING MODELLING UNCERTAINTIES

This chapter is in preparation to be submitted to the Journal of Hydrology. Model simulations, data processing and figure creation were carried out by Rosie Lane, with guidance and suggestions from all co-authors. Jim Freer helped in downloading the UKCP18 climate data, and Gemma Coxon created 12 km DECIPHeR input grids to help setup the model to run with climate inputs. Emma Robinson provided the python code used to calculate CHESS-PET, and provided suggestions for how it could be adapted to work with the available UKCP18 outputs. The manuscript was written by Rosie Lane, with comments from all co-authors.

Citation: Lane, R. A., Coxon, G., Freer, J., Seibert, J., and Wagener, T. (in preparation). UK Climate Projections on high flows across Great Britain, including hydrological modelling uncertainties.

6.1 Context

Research question three: What is the impact of climate change on median and higher flows across Great Britain?

Climate change could significantly impact river flows across Great Britain, but there are large uncertainties in both future climate changes and how these propagate to changing flows. This research chapter uses the model developed in Research Chapter Two in order to provide the

first GB-wide spatially consistent projections to include both climate and hydrological modelling uncertainties. The modelled range in regional and nationwide changes to high flows is provided, alongside analysis of the relationship between changing precipitation, PET and high flows.

6.2 Introduction

Climate change will likely significantly alter hydrological regimes in many parts of the world, with vast implications for water resource planning and policy (Brown et al., 2015; Intergovernmental Panel on Climate Change, 2014; Wagener et al., 2010). Projections indicate an intensification of the hydrological cycle, with a warmer climate overall leading to more rain falling in high-intensity events (Huntington, 2006; Intergovernmental Panel on Climate Change, 2014; Trenberth, 2011). This increase in the frequency and severity of extreme rainfall events is likely to increase flood risk in many regions. However, the conversion of rainfall to runoff is not straightforward, as changes in river flows are the result of complex and non-linear interactions between changing rainfall and evapotranspiration, and the influence of basin properties (Arnell, 2011; Laizé and Hannah, 2010; Sawicz et al., 2014). There are also many uncertainties surrounding future climate projections; while climate models show general agreement on rising temperatures and increasing extreme precipitation throughout the 21st century, they differ in the magnitude and spatial patterns of change (Fowler and Ekström, 2009; Met Office, 2019; Nikulin et al., 2011). To guide water-related policy and decision making and to ensure adequate adaptation to future changes in flooding, we therefore need hydrological modelling studies to help understand and quantify climate change impacts on the hydrological regime, and the uncertainties surrounding these projections.

Recent literature has highlighted the need for more high-resolution modelling studies at national to continental scales (Watts et al., 2015). Currently, many hydrological climate impact studies focus on small numbers of catchments in great detail (e.g. Fowler and Kilsby 2007; Prudhomme et al. 2003; Reynard et al. 2004) or produce the global picture of change at relatively low resolution (e.g. Arnell and Gosling 2013). However, many policy decisions, such as flood funding allocations or guidance on flood infrastructure allowances, are made at the regional to national scale. Studies which apply a consistent methodology at high resolution across a large area are most valuable to inform these regional and national policy decisions, as they (i) provide a broad overview of future changes, (ii) provide locally relevant information, in contrast to global impact studies, and (iii) enable direct comparison between catchments to identify regions that will experience the most significant climate change impacts (Watts et al., 2015). Using a large sample of catchments also ensures a more robust evaluation of the relationship between climate change impacts and hydrological response.

Hydrological climate change impact studies often use Global Climate Model (GCM) or Regional Climate Model (RCM) information (e.g. rainfall and temperature projections) to drive hydrological models. Throughout this modelling chain there are many uncertainties, which cascade from one step through to another. These include uncertainties in GCM structure and sub-grid parameterisations, uncertainties in RCM structure and parameterisations, uncertainties in the chosen downscaling and bias correction techniques, and uncertainties in the selection of hydrological model structures and their parameters. Many studies have attempted to quantify these uncertainties (e.g. Bosshard et al. 2013; Kay et al. 2009; Smith et al. 2014b; Wilby and Harris 2006), but these tend to focus on small samples of catchments. On a national scale inclusion of modelling uncertainties is challenging, and while a few UK studies include RCM uncertainties (e.g. Bell et al. 2016; Prudhomme et al. 2012) and recognise the importance of representing uncertainties throughout the modelling chain, the inclusion of hydrological modelling uncertainties is still rare. A notable exception is Christierson et al. (2012), who modelled the impact of changing climate for 70 catchments across the UK using two different hydrological model structures and ensembles of model parameters. However, this study was based on probabilistic climate projections which were not spatially coherent, and therefore did not present possible GB-wide changes but rather individual scenarios for each catchment. Incorporating hydrological model parameter uncertainties is important, as it has been shown that very different projections for future catchment behaviour can be provided by parameter sets with similar performance over a baseline period (Mendoza et al., 2015; Singh et al., 2014). However, there are currently no studies providing spatially consistent projections of future changes in flooding across entire Great Britain, which include both RCM and hydrological model parameter uncertainties.

An updated set of national climate projections has recently been released for the UK, UKCP18 (Lowe et al., 2019). These have advanced upon previously available national projections (UKCP09) through (1) increased resolution of global climate model from 300 km to 60 km providing better representation of synoptic-scale weather systems, mountains and coastlines, (2) increased resolution of regional climate model from 25 km to 12 km, which may improve the representation of extreme precipitation, (3) updated atmosphere model and improved parameterisations of many sub-grid scale processes, and (4) improved representation of dynamical influences on regional climate variability such as improvements in predictions of the winter North Atlantic Oscillation (NAO) (Murphy et al., 2018). These include a perturbed physics ensemble of RCM projections at 12 km resolution, providing 12 possible climate futures varying due to RCM parameter uncertainties. The implications of these new climate simulations for river flows are of great interest, as the improved simulation of precipitation may improve projections of future flooding.

This chapter aims to explore the impact of the new UKCP18 climate projections for high flows across Great Britain, whilst reflecting parameter uncertainties in climate and hydrological

models at the national scale. A climate-hydrological model cascade was employed, with output from an ensemble of 12 spatially consistent RCM projections used to drive a nationally applied hydrological model with 30 spatially consistent parameter field samples. The resulting 360 future flow scenarios covering a large sample of 346 catchments were analysed to answer the following research questions:

1. What is the range in potential changes to higher flows (including median flows (Q50), high flow quantiles (Q10 and Q1), annual maximum flows (AMAX) and the number of peaks over threshold) across Great Britain, due to parameter uncertainties in climate and hydrological modelling?
2. How will changes in the magnitude and frequency of high flows vary spatially and by region?
3. What is the relationship between changing climate (precipitation and potential evapotranspiration) and high flow response, and how does this vary by region?

Our study presents the first consistent climate change projections for high flows across all of Great Britain to include both climate model and hydrological model parameter uncertainties. The incorporation of a large sample of catchments also enabled robust and generalisable analysis on the relationship between climate forcing, catchment characteristics and hydrological response.

6.3 Methods and data

6.3.1 Overview

This chapter uses a climate-hydrological modelling chain to assess the implications of the UKCP18 climate projections for river high flows across 346 catchments covering Great Britain (see Section 6.3.2 for catchment selection). An ensemble of 12 spatially coherent regional climate model (RCM) projections are first bias corrected (see Section 6.3.3), and then used directly as inputs to the DECIPHeR hydrological modelling framework to produce flow projections (see Section 6.3.4). For each RCM ensemble member, DECIPHeR simulations are carried out using 30 nationally consistent hydrological model parameter fields. The use of 12 RCMs and 30 hydrological model parameter sets results in 360 national simulations, representing uncertainty due to RCM and hydrological model parameterisation. A flow diagram of this process showing the key underlying datasets is given in Figure 6.1.

To explore climate change impacts on high flows, flow metrics were selected to assess median flows (Q50), high flow quantiles (Q10 and Q1), the magnitude of peak flows (AMAX), and the frequency of peak flows (see Section 6.3.5). The skill of the climate-hydrological modelling chain was first evaluated relative to observed flow metrics, and then changes in flow metrics between the baseline (1985 – 2010) and future (2050 – 2075) periods were evaluated.

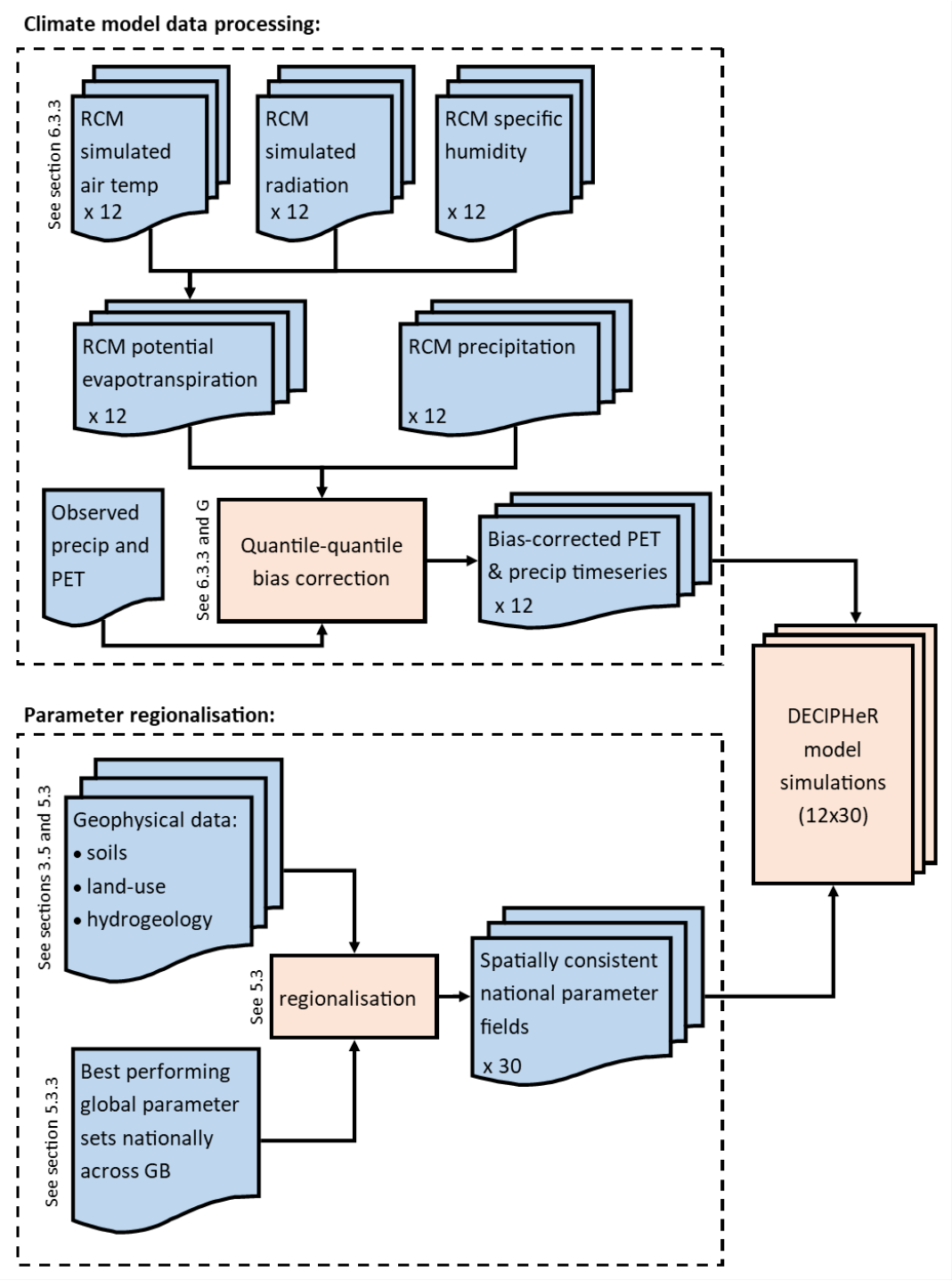


FIGURE 6.1. Flow diagram demonstrating climate-hydrological modelling chain.

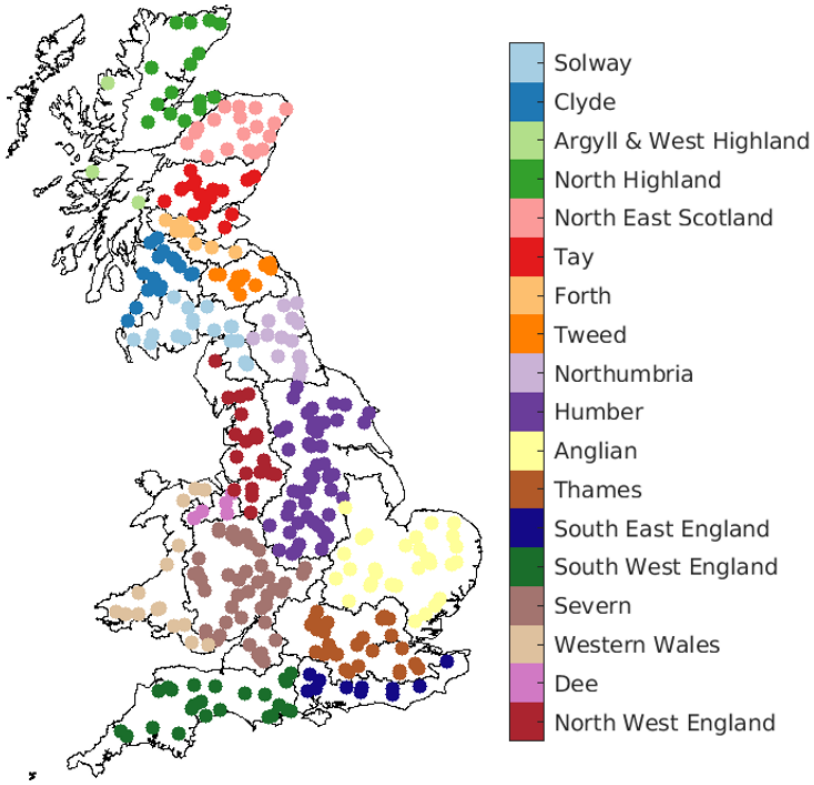


FIGURE 6.2. Locations of the catchments used in this study, grouped according to the UKCP18 river basin districts.

6.3.2 Catchment selection

A large sample of 346 catchments covering Great Britain was selected for this study. This sample provides a dense coverage across Great Britain, with catchments in all river basin districts, as shown in Figure 6.2. Gauging stations were selected from the UK National River Flow Archive (NRFA) Service Level Agreement (SLA) Network (Centre for Ecology and Hydrology, 2016; Dixon et al., 2013). This network of 715 gauges form a subset of strategically valuable NRFA catchments, where additional validation and quality testing procedures have been carried out (Dixon et al., 2013). As hydrometeorological data were available on 12 km grids at daily resolution, we chose to exclude catchments which were smaller than 144 km^2 (i.e., one RCM grid), because for these small catchments local variation in precipitation could be problematic for the RCM ensemble scale, and for small flashy catchments sub-daily data would be required to capture high flow and peak responses effectively.

6.3.3 Climate model data

Climate scenarios representing changes in precipitation and potential evapotranspiration (PET) were derived from the UKCP18 regional climate projections (Murphy et al., 2018). These com-

prised a perturbed-physics ensemble of 12 regional climate model simulations, run at 12km resolution with daily output from 1981 to 2080 (Met Office Hadley Centre, 2019). The 12 RCM projections were all driven by the same GCM (HadGEM3-GC3.05), and for the RCP8.5 emissions scenario. This GCM has been shown to sample the warmer range of global outcomes (Lowe et al., 2019), and so combined with a single emissions scenario, we only sample the warmer range of possible climate outcomes. A key advantage of this data over other UKCP18 products is that it is has full spatial and temporal coherence, and therefore allows for the assessment of interactions between changes in precipitation and PET as well as providing a nationally consistent picture of future changes (Met Office, 2020).

Whilst precipitation data were available as an RCM output variable, PET time series needed to be derived from other relevant UKCP18 model outputs. There are many possible approaches to calculating PET from climate model data, with the choice of PET equation shown to impact the subsequent changes in PET over time (Kay and Davies, 2008; Prudhomme and Williamson, 2013). Here, PET was calculated to be consistent with the CHESS-PE dataset used for hydrological model parameterisation (Robinson et al., 2015a). The CHESS-PE dataset uses the Penman-Monteith equation, calculating PET as a function of air temperature, specific humidity, wind speed, shortwave radiation, longwave radiation, and air pressure. These variables were all available as UKCP18 output apart from air pressure, which was calculated using the integral of the hypsometric equation with modelled temperature as an input (Shuttleworth, 2012).

Bias correction of climate model output data is often required for hydrological impact studies due to the occurrence of considerable biases in hydrologically relevant variables (Addor and Seibert, 2014; Cloke et al., 2013; Ning et al., 2012; Teutschbein and Seibert, 2012). An analysis of biases in the UKCP18 regional projections identified systematic biases in the model output precipitation and model-derived PET data (see Appendix F for more information). For precipitation, RCM biases included overpredictions of mean annual precipitation across Great Britain by up to 50%, underpredictions of rainfall in wetter areas along the west coast, and an increased number of wet days (an average of around 15% more rainy days per year than observations). RCMs tend to overpredict the variance in PET, resulting in overestimations of PET in the southeast, where observed PET is high, and underestimations in Scotland as well as an incorrect seasonal variation with overestimations in summer (up to around +40%) and underestimations in winter (up to -100%). A bias correction method was required to reduce these biases in RCM precipitation and PET, so that they were suitable for hydrological modelling.

The choice of bias correction has been shown to impact the magnitude and spread of projected changes in flood-producing flows (Cloke et al., 2013; Smith et al., 2014b), and should, therefore, be carefully considered. Techniques to directly adjust RCM simulations range from relatively

simple linear scaling through to more complex approaches such as quantile mapping (Teutschbein and Seibert, 2012). The delta change method, which modifies historical time series based on RCM-simulated changes, is commonly applied (e.g. Veijalainen et al. 2010. However, this method cannot change the temporal sequencing of events, so it cannot be used to evaluate changes in flood timing. The quantile mapping bias correction approach was selected for both precipitation and PET (this method has also been referred to as distribution mapping, probability mapping, model output statistics, or histogram equalisation). The quantile mapping approach accounts for errors in the variability of PET, and ensures that heavy precipitation events important for high flows were appropriately corrected as well as mean precipitation. It also corrected for biases in the number of wet days in the RCM data.

Observed precipitation and PET data used for bias correction came from the CEH-GEAR (Keller et al., 2015; Tanguy et al., 2014) and CHESS-PE (Robinson et al., 2015a) datasets respectively. For each grid-cell and month for precipitation the following steps were performed:

1. Empirical Cumulative Distribution Functions (CDFs) were calculated for the observed precipitation, and RCM simulated precipitation for the control/baseline period (all dates where observed and simulated precipitation were available).
2. The fractional change in precipitation between the observed and control/baseline simulated was calculated for each cumulative probability.
3. The whole simulated precipitation series was then bias-corrected. The cumulative probability of each precipitation value was calculated, and the value was modified by the fractional change for that cumulative probability.

The same method was carried out for PET, with a minor modification. It was found that for some Scottish catchments, fractional changes could become very large when PET values were low (<0.1 mm/day) as a result of dividing by values close to zero. To prevent unrealistic spikes in future PET at low cumulative probabilities, a check was added to ensure that PET values at a low cumulative probability were always smaller than values at a higher cumulative probability. This bias correction methodology successfully reduced biases in RCM data over the observational period (see Appendix G for more information). However, it is important to note that bias correction assumes that despite showing large biases in hydrometeorological variables the RCM output is still meaningful and changes in hydrometeorological variables are well simulated.

There is no ideal bias correction methodology, and there are several key limitations to both bias-correction in general and the quantile mapping bias correction approach more specifically. Firstly, bias correction assumes that biases in RCM output are stationary and so methods of bias correcting baseline data also hold into the future. However, with a changed future climate it may be expected that model biases may also change. Secondly, bias correction techniques are limited by the quality of the observational data which model biases are corrected against. There

are many uncertainties in observational rainfall measurements, with the main sources of error for raingauges in the UK including adhesion of water to the gauge surface, water splashing in and out of the gauge, undercatch due to wind and evaporation (Keller et al., 2015). Further uncertainties are introduced when this gauge data is interpolated to create a national product, especially in areas where raingauge measurements are sparse, which is the case for some areas of Scotland (Keller et al., 2015). There are also large uncertainties in the observational PET product, including choice of PET equation and measurement errors in the underlying data. These errors in observational datasets are not considered in the bias correction methodology. Thirdly, the quantile mapping approach corrects simulated data against the CDF of observed data. For the extreme high end of these observations (e.g. exceptionally heavy rainfall events), there will be few observations to constrain the CDF, and therefore bias correction is likely to be less robust for the rarest events. However, quantile mapping bias correction was chosen above a more simple monthly-mean approach to account for differing errors between heavy, median and light precipitation events within the same month.

The bias-corrected RCM data was used directly as hydrological model input, with no further downscaling. This was possible due to the size of the catchments we have chosen to analyse coupled with the high resolution (12 km) of the RCM data, which is a key advantage of the UKCP18 climate product over previous climate projections.

6.3.4 Hydrological modelling

The DECIPHeR hydrological modelling framework was selected to transform precipitation and PET into river flows (Coxon et al., 2019). DECIPHeR is a semi-distributed hydrological modelling framework which discretises the modelling domain into hydrological response units (HRUs). Here, the model was configured to be consistent with the 12 km UKCP18 data, with HRUs defined by splitting the landscape into 12 km input grids which were further sub-divided by accumulated area classes, slope classes and sub-catchment boundaries to capture topographic and catchment attribute controls in hydrological processes. This HRU-based approach enabled representation of the spatial variation of input time series, whilst being computationally efficient to facilitate the use of multiple hydrological and RCM parameter sets across the large sample of catchments. Here, we have selected the default model structure, which is based on the widely used TOPMODEL, and has previously been shown to perform well across Great Britain and selected catchments (Coxon et al., 2019) and also Chapter 5.

National fields of model parameters have been generated using the multiscale parameter regionalisation technique (Samaniego et al., 2010), as described in Chapter 5 . This method relates model parameters to spatial catchment attribute data (including soil texture, land-use,

and hydrogeology) via transfer functions. The coefficients of the transfer functions were then constrained simultaneously on a large sample of 437 British catchments, instead of directly constraining model parameters. Over 3500 possible parameter fields were produced, and of these, the top 30 parameter fields were selected for this study. These were selected as they produced non-parametric KGE scores (Pool et al., 2018) above 0.8, when taking the average value across the large sample of catchments in Great Britain (see Chapter 5). Using catchment attribute data to define the spatial distribution of model parameters means that parameter fields are spatially coherent with no artificial discontinuities (Mizukami et al., 2017; Samaniego et al., 2017). This is advantageous when modelling climate impacts for larger regions or entire countries, as it has been shown that artificial discontinuities in parameter fields can lead to discontinuities in modelled variables (Mizukami et al., 2017).

The DECIPHeR framework requires inputs of precipitation and PET, as well as spatial catchment attribute data for parameterisation. The model was driven continuously with climate data over the period 01/01/1981 – 30/12/2075, with 01/09/1985 – 30/8/2010 extracted as the baseline period, and 01/09/2050 – 30/08/2075 being used as the future period in all further analysis. Starting the baseline in 1985 gave over four years for a hydrological model warm-up period. Hydrological simulations were also carried out using observed data over period 01/01/1981 – 30/08/2010, to provide a benchmark of model performance which the RCM-driven simulations could be compared against over the baseline. For these simulations, potential evapotranspiration data from the CHESS-PE dataset (Robinson et al., 2015a,b) and precipitation data from CEH-GEAR (Keller et al., 2015) were re-gridded to match the UKCP18 12 km data. All observed river flow data were from the UK National River Flow Archive (NRFA) (Centre for Ecology and Hydrology, 2016; Dixon et al., 2013).

6.3.5 Hydrological indicators

To explore changes in the magnitude of high flows, we calculated the percentage changes in four different flow metrics between the baseline (1985 - 2010) and future (2050 - 2075) periods. Flow metrics calculated were 1) the average annual maximum (AMAX) flow, 2) Q1, the flow value exceeded 1% of the time, 3) Q10, the flow value exceeded 10% of the time, and 4) Q50, the median flow or flow value exceeded 50% of the time. These were selected to give a broad overview of future higher flow changes, ranging from flood flows (AMAX and Q1) to average flows (Q50).

To analyse changes in the frequency of high flows, a peaks-over-threshold (POT) analysis was carried out. Thresholds were defined for each catchment to extract an average of three peaks per year over the baseline period. To ensure flood events were independent, no peak was selected within seven days of a larger peak. This selection was consistent with previous studies,

for example, Svensson et al. (2005) used a five-day window for catchments smaller than 45,000 km^2 (the largest catchments in the UK are 10,000 km^2), whilst Petrow and Merz (2009) used ten days for catchments across Germany. Having found a POT threshold for each catchment over the baseline that resulted in an average of 3 peaks per year, the number of peaks exceeding this threshold in the future period was counted. The percentage change between the count of 75 peaks total gained in the baseline and peaks gained in the future was then calculated as an indication of changes in the frequency of flood events.

6.4 Results

6.4.1 Evaluation of climate-hydrological modelling chain

Overall, the simulations of the climate-hydrological modelling chain across Great Britain bounded the observations (Figure 6.3). Our evaluation focused on the performance for hydrological indicators relevant for higher flows, namely flow quantiles Q50, Q10, and Q1. The maps (Figure 6.3a) show biases in the highest (i.e. wettest) and lowest (i.e. driest) simulation for each individual catchment from the ensemble of 12 RCMs and 30 hydrological model parameter sets compared to observed flows. For catchments which are well represented by the modelling chain, we would expect simulated flows to bound the observations. Therefore the highest simulation would show a small positive bias, and the lowest simulation would show a small negative bias. For the majority of catchments (75% for Q10, 59% for Q1), the model simulations follow this pattern and bound observed discharge, with biases less than 50%. However, the modelling chain overestimated flows in the south-east (by over 100% in some areas), and to a smaller extent underestimating peak flows in the west. The difficulties of modelling catchments in southeast England has been documented in Research Chapter One and previous studies (Coxon et al., 2019; Lane et al., 2019; Seibert et al., 2018), and is likely due to complex aquifer systems facilitating inter-catchment groundwater flow. These catchments should, therefore, be treated with caution when interpreting the results.

Model performances are shown in more detail for a selection of catchments covering a variety of error characteristics (Figure 6.3b). Here, error (i.e. bias) in modelled flow driven by RCM output (green) is compared to modelled flows driven by observations (yellow) using the same 30 hydrological model parameter sets. For most gauges, simulated flows bound the observations, even when driven by the RCM meteorological data. This result was expected as the RCM data has been bias-corrected against observations, and therefore the RCM data will be similar to observations in magnitude, albeit with different sequencing of events. There is no consistent relationship between model biases and flow percentiles, with gauge 9002 showing an increased tendency to overestimate higher flows, whilst gauge 83013 showed a decreased tendency to overestimate higher flows.

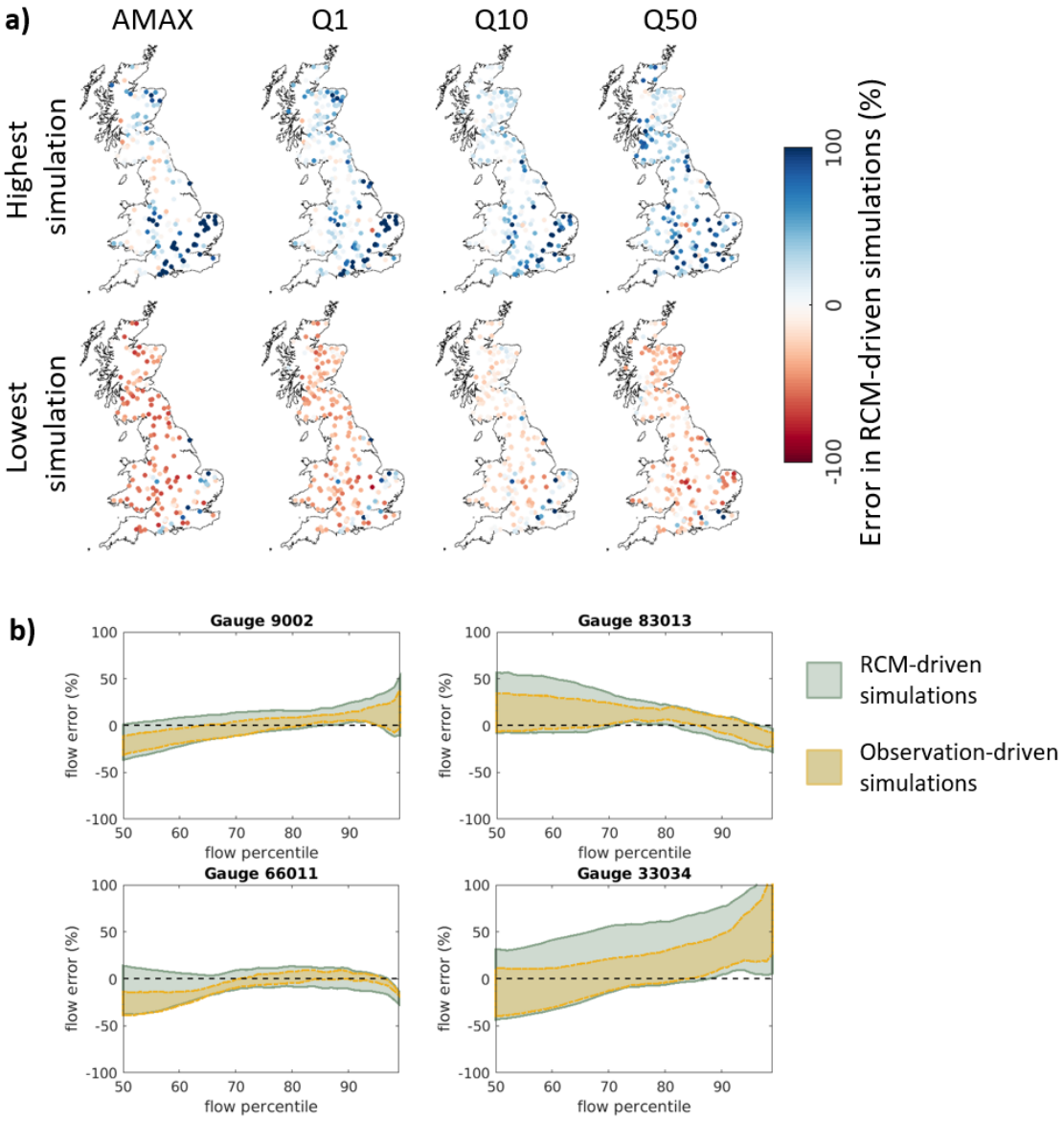


FIGURE 6.3. Evaluation of model performance, showing how well the modelled flow statistics from the climate-hydrological cascade bound the observed flow statistics. The maps (a) show error in RCM-driven simulations compared to the observed. The top row shows the highest positive error from the 360 simulations, whilst the bottom row shows the lowest negative error, calculated separately for each catchment. When considered together, these show how well the RCM-driven simulations bound the observed flows. Four gauges are shown in more detail (b), giving error across median and higher flow percentiles.

6.4.2 Meteorological changes

Median precipitation is projected to decrease almost everywhere with GB-average median precipitation projected to decrease by 31-61% between the different RCMs, with the only exception being in west Scotland (Figure 6.4a). This decreasing median precipitation contrasts with very high precipitation (99^{th} percentile), which is expected to increase across most of Great Britain, by an average of 5 - 20%. The 90^{th} percentile precipitation shows a more mixed picture, with GB-average changes of -9% to +6%. Generally, increases were simulated for areas along the west coast and in western Scotland, whilst decreases can be seen across southern England and Wales.

All RCMs indicate increasing PET over the modelled period (Figure 6.4b-c). These broadly align with observed PET across Great Britain between 1980 - 2010, although it is difficult to distinguish an upward trend in the observed PET data over such a short period. GB-average PET values show increases of 23 - 38% between the baseline and future period, with the largest PET increases (33 - 50%) seen in the south, and the smallest PET increases (11 - 19%) simulated for northwest Scotland.

6.4.3 Spatial changes in high flows across GB

Maps showing the spatial pattern of changes in high flow magnitude and frequency are presented for three example simulations in Figure 6.5. As the spatial pattern was similar between the ensemble members, we have focused on RCMs 13, 8 and 4 which represent low, average, and high GB-average projections respectively in terms of Q10 changes, with plots for all RCMs given in Appendix F. These projections were selected to indicate the range in flow changes across GB, but it is important to note that they are spatially coherent futures from single RCM ensemble projections. Therefore they do not reflect the full range of flow changes for each individual catchment that would be obtained by evaluating the entire RCM ensemble.

Despite differences between the example projections, there is a clear east/west divide for high flow magnitude metrics (AMAX, Q1 and Q10) with increased flows for catchments in the west and decreasing flows in the east. The largest decreases in high flows are in eastern England, particularly in the Anglian river basin district, whilst the largest increases in flow are along the west coast. Median flow (Q50) projections indicate reductions in flow almost everywhere, but these reductions are generally lower for catchments in western Scotland. The frequency of high flow events, represented by changes to the number of peaks over threshold events, also shows general increases in the west and reductions in the southeast. The spatial pattern is very similar to the changes to high flow magnitude, indicating that western catchments could experience larger annual maximum floods combined with more frequent high flow events.

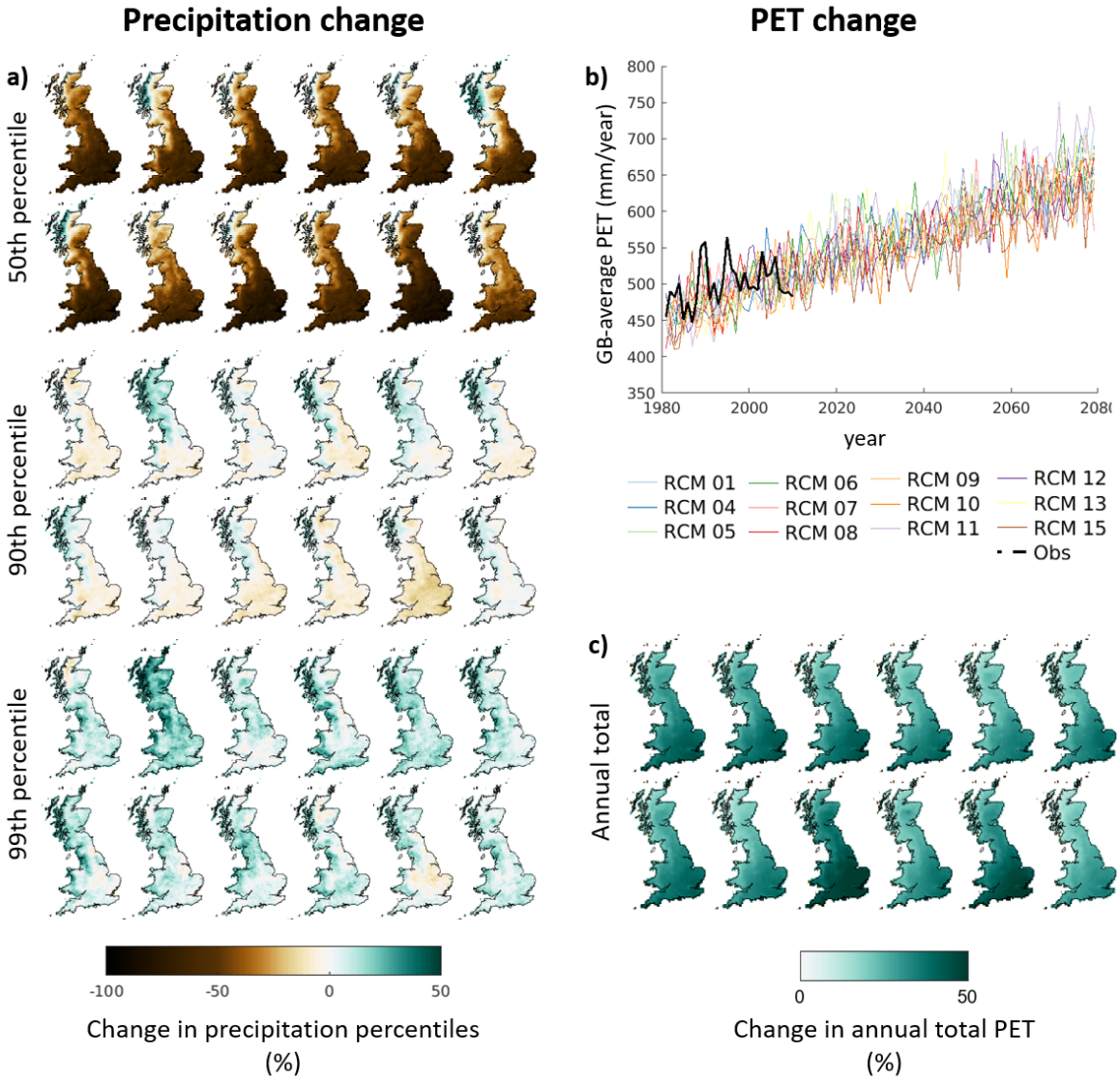


FIGURE 6.4. Precipitation (a) and PET (b-c) change. GB-maps are presented for each ensemble member in order. Top row: RCM01, RCM04, RCM05, RCM06, RCM07 and RCM08, bottom row: RCM09, RCM10, RCM11, RCM12, RCM13, RCM15.

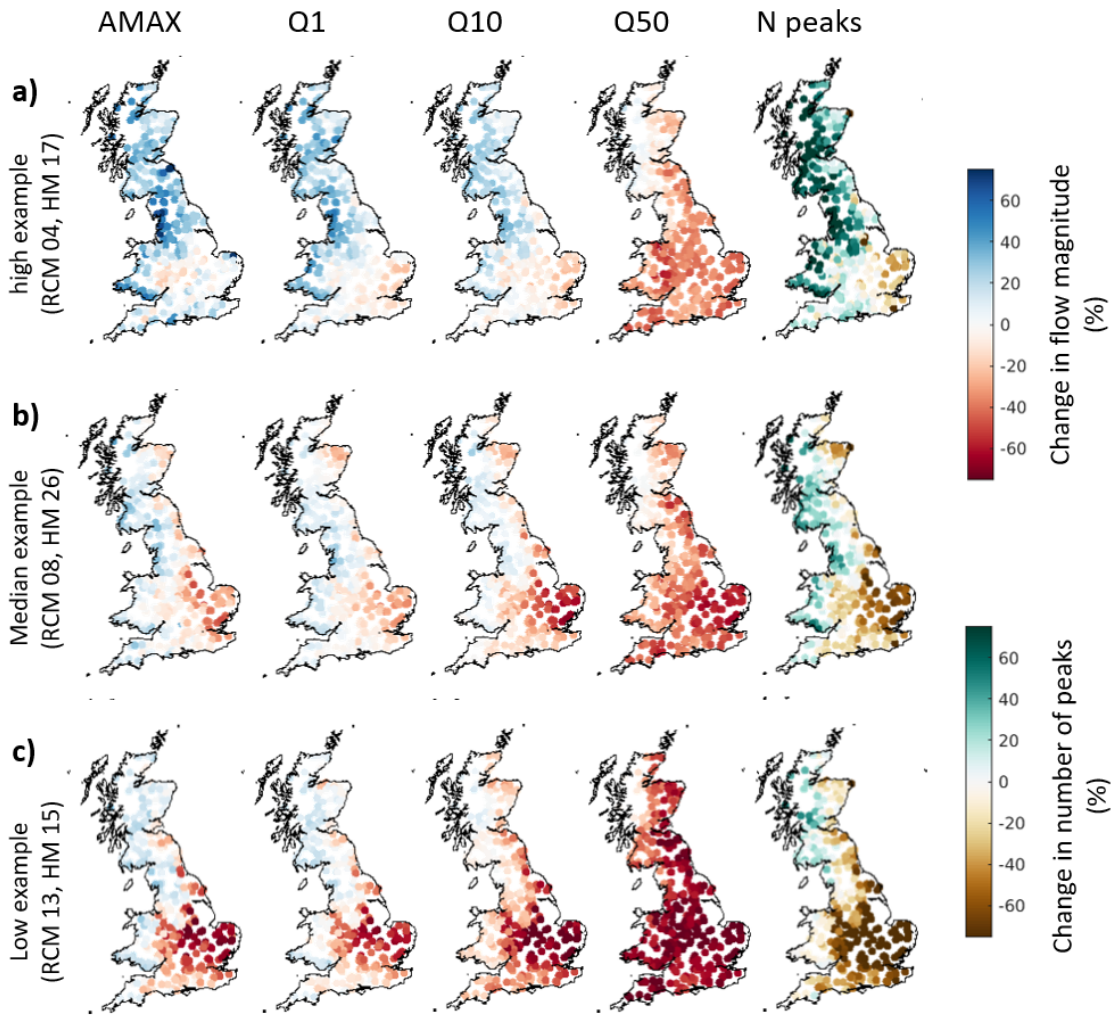


FIGURE 6.5. Maps showing changes in the magnitude and frequency of peak flows between the baseline and future periods for example simulations. Each row shows a nationally coherent projection, with plots of changes in five flow metrics (AMAX, Q1, Q10, Q50 and the number of peak flows above a threshold). This combination of RCMs and hydrological parameter sets were selected from the ensemble of 360 simulations to give an indication of the ensemble spread, as they provided the highest, median, and lowest GB-average change in Q10, but they do not show the full range of possible changes for individual catchments or all flow metrics.

6.4.4 Uncertainties arising from RCM and hydrological model parameters

Changes for the hydrological indices for the different RCMs and across regions were visualized by heatmaps to enable easy comparison (Figure 6.6). These heatmaps present the median flow values from the sample of hydrological model parameters for each flow statistic, with the full range of regional projections presented in Table 6.1. They highlight similarities between RCM members: most RCM ensembles result in increasing AMAX flows in Scotland, northern England, and west Wales, and decreasing AMAX flows in the Anglian river basin district. Most RCM ensembles also result in decreasing Q50 flows everywhere except for the Argyll and West Highland districts in west Scotland. However, there are also important differences between the different RCM projections, including; i) differences in the spatial variation of changes across GB, for example RCM 15 shows relatively little variation between regions (range of 28% between AMAX projections) whilst RCM 11 shows a large variation (range of 104%), ii) differences in the magnitude of projected changes for each region, for example NW England projections for Q10 range from -16% to +20% between RCMs, and iii) the tendency for some RCMs to simulate increases in flow (e.g. RCM 04) whilst others tend towards decreases (e.g. RCM 13) which relates to relative change in 99^{th} percentile precipitation (see Figure 6.4). These differences demonstrate the importance of considering multiple RCMs, to show a more complete picture of potential future changes.

RCM parameters were a larger source of uncertainty in flow changes than hydrological model parameters (see Figure 6.7). This finding agrees with previous studies, which generally find climate models to be the largest source of uncertainty in hydrological climate impact assessments (Addor and Seibert, 2014; Bosshard et al., 2013; Kay et al., 2009). However, hydrological model parameters selection is a large source of uncertainty in the south-east, especially in the Anglian river basin region. This region receives relatively little precipitation compared to the rest of GB. Previous studies have shown that drier catchments are more sensitive to parameter selection, with fewer good parameter sets found for drier than for wet catchments found in Research Chapter One.

6.4.5 Relationship between climate changes, flow changes and catchment characteristics

The relationship between precipitation change and change in flood flows (Q1) across all catchments, and RCMs is presented in Figure 6.8. This shows that there is a strong positive correlation between precipitation change and flood response, albeit with a large variation between catchments. The non-linearity between changing precipitation and changing Q1 flows can be seen, with a 25% increase in precipitation leading to a 20 - 50% increase in Q1. Surprisingly, for some catchments, heavy precipitation increases yet there is a reduction in Q1 flows (i.e. catchments in the bottom right quadrant of Figure 6.8). This flow reduction could be due to the contrasting

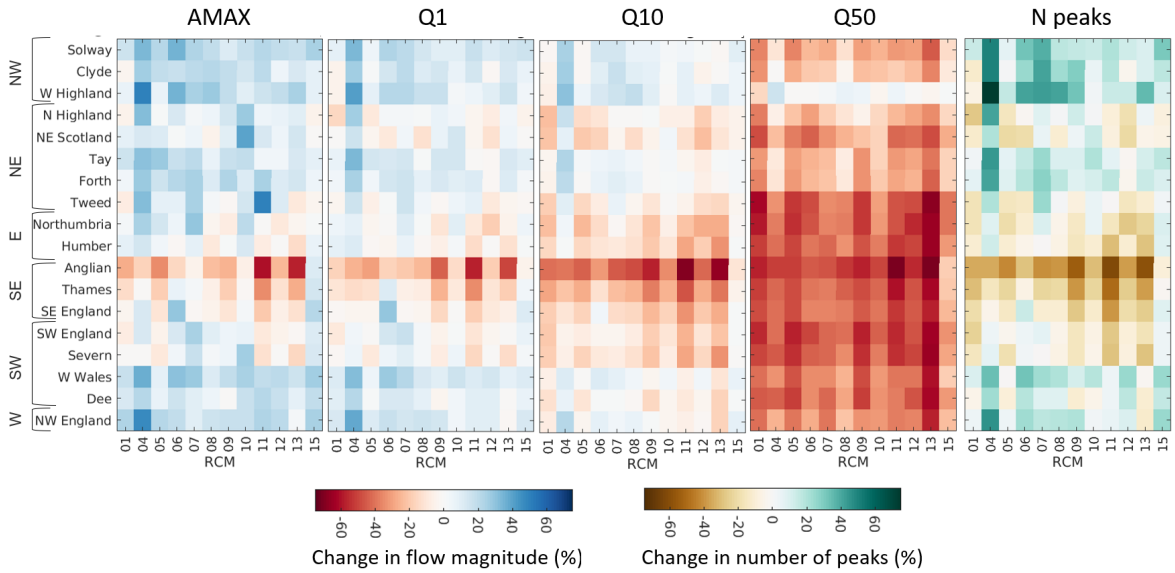


FIGURE 6.6. Heatmaps showing region-average changes in flow magnitude between the baseline and future periods, for all 12 RCMs. Regions have been ordered by location, with the relative position within GB given on the left. To focus on differences between RCMs, the median flow value from the hydrological model parameter sets is presented.

effect of increasing PET, resulting in generally drier anticedent conditions for catchments and thus reduced flows due to the increases in soil moisture storage deficits.

The relationship between change in 95th percentile precipitation, total PET and Q1 is given in Figure 6.9; other variations of precipitation, PET and flow changes produced similar results (but are not shown). There is a clear relationship between climate forcing and hydrological response. Increased heavy precipitation tends to lead to increased Q1, whilst decreased or unchanged heavy precipitation, combined with increasing PET, leads to reduced Q1 flows. The range in climatic changes is different for each region (see Figure 6.9b), which is a key reason for the regional differences in Q1 changes. However, the hydrological response differed between regions for the same climate forcing. For example, a 6% decrease in 95th percentile precipitation and over 45% increase in total PET leads to an average 53% reduction in Q1 in the Anglian river basin district, but only an average 15% decrease in Q1 in the Thames region in the South-east. These results highlight the importance of how multiple climatic factors impact regional flow responses differently due to the non-linearity within the hydrological processes.

The runoff coefficient (runoff divided by precipitation) helped to explain these regional differ-

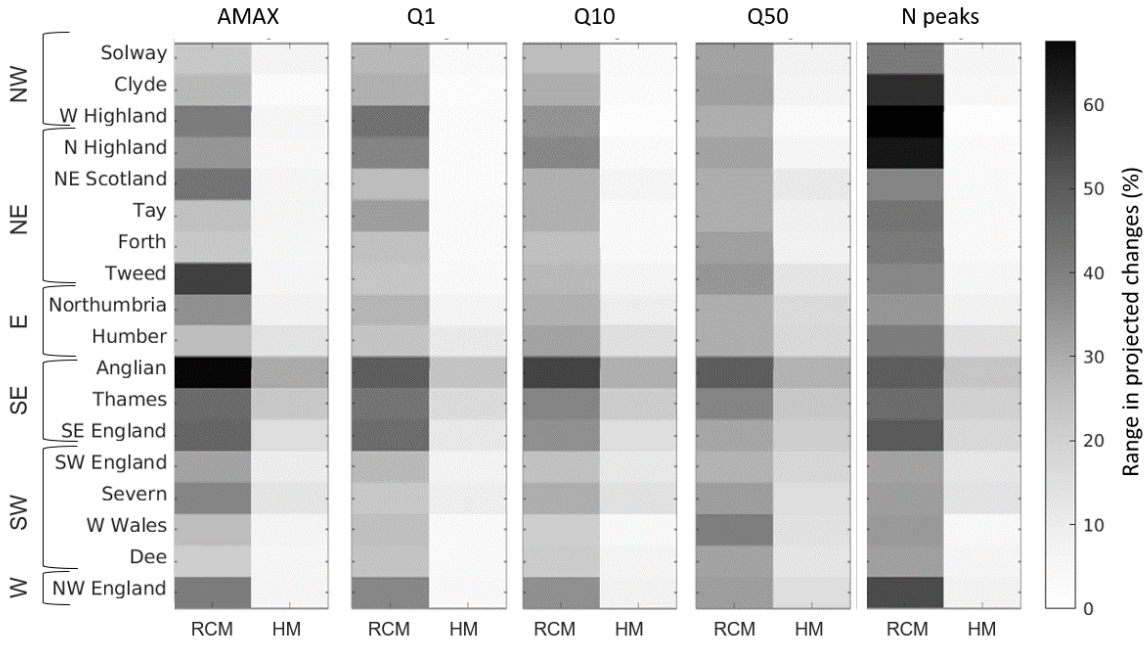


FIGURE 6.7. Relative uncertainties from inclusion of different RCM and hydrological model (HM) parameter sets. The RCM range was calculated as the full range in regional-average changes between the RCMs, using the median of all HM parameter sets. Similarly, the HM range was calculated using the median output of all RCMs.

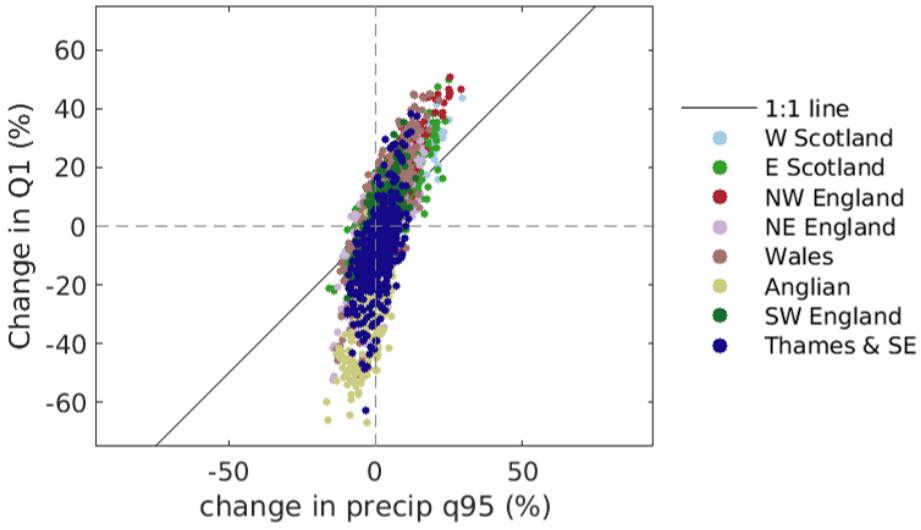


FIGURE 6.8. Relationship between precipitation change and Q1 change across all catchments. Results are presented for all RCMs using the median of all hydrological parameter sets.

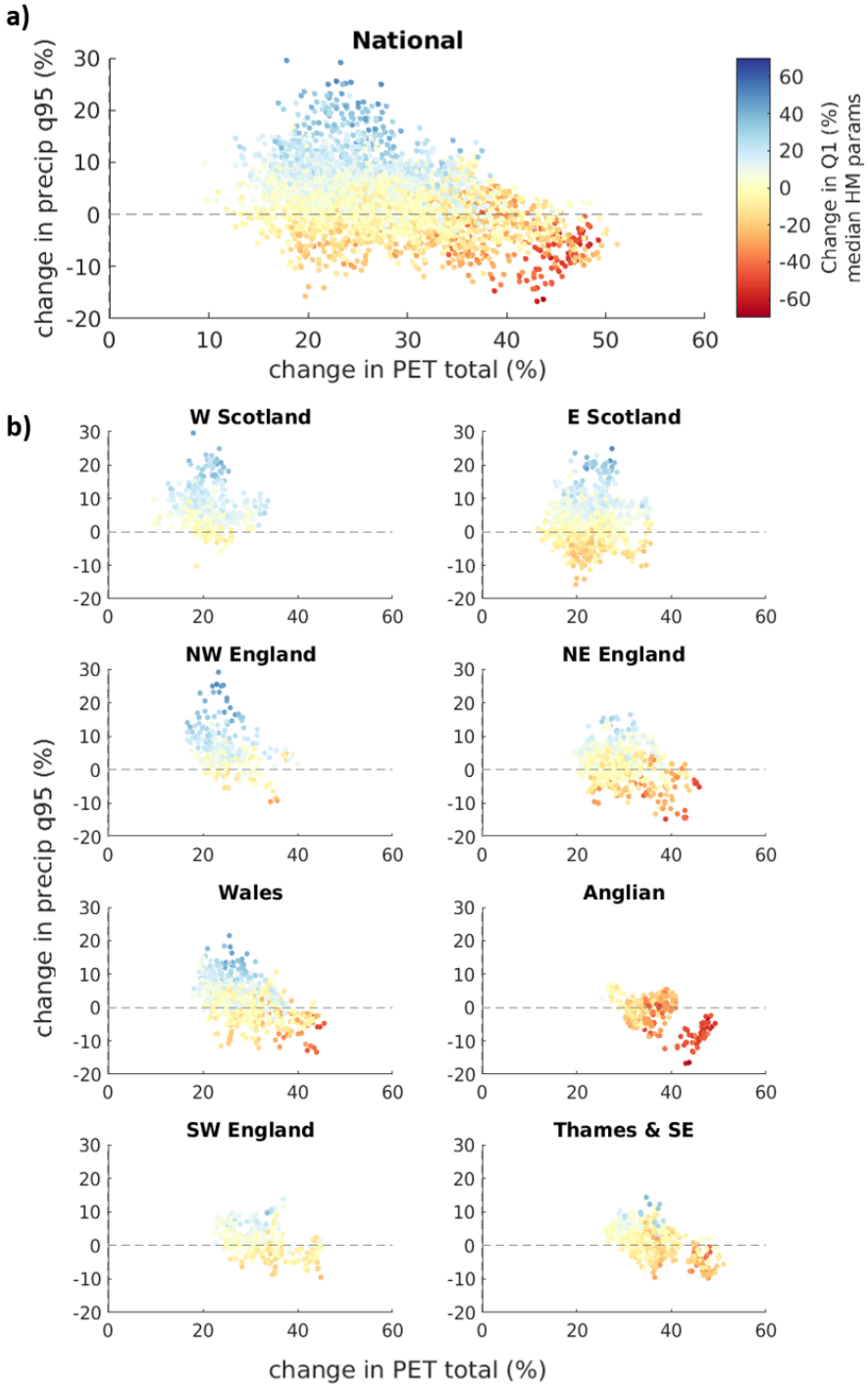


FIGURE 6.9. Relationship between changing climate and changing high flows (Q1), shown for all catchments nationally (a) and by region (b). Plots show climatic changes from all RCMs, coloured by the median change in Q1 flows from the ensemble of hydrological model parameter sets. Regions which are shown together exhibited similar patterns.

TABLE 6.1. Ensemble range in projected changes for each flow metric. All changes are given as percentage differences between the baseline and future periods. Low, Med and High refer to the lowest, median, and highest region-average changes from the ensemble of RCM and hydrological model parameters.

Region	AMAX change (%) Low, Med, High			Q1 change (%) Low, Med, High			Q10 change (%) Low, Med, High			Q50 change (%) Low, Med, High			N. peaks change (%) Low, Med, High		
Solway	7	18	49	1	13	37	-4	4	24	-49	-26	-4	4	24	79
Clyde	-10	15	29	-9	11	27	-8	5	28	-42	-20	5	-28	23	77
W Highland	3	18	65	-7	14	46	-4	9	31	-17	1	19	-16	35	113
N Highland	-15	4	39	-17	-1	33	-27	-6	18	-41	-20	0	-41	-5	68
NE Scotland	-7	8	45	-15	0	19	-27	-13	9	-56	-33	-12	-41	-12	33
Tay	1	13	36	-3	11	36	-9	2	25	-43	-26	-3	-7	17	75
Forth	6	17	40	1	11	37	-5	3	22	-49	-23	-3	-5	23	73
Tweed	-14	6	59	-14	1	19	-20	-5	14	-69	-41	-19	-37	-3	52
Northumbria	-11	3	38	-20	2	17	-32	-16	8	-69	-44	-24	-39	-16	26
Humber	-21	4	27	-18	0	17	-33	-11	9	-71	-42	-23	-53	-12	31
Anglian	-74	-21	19	-68	-22	8	-80	-41	3	-85	-50	-9	-99	-55	13
Thames	-50	-10	15	-44	-10	18	-59	-24	4	-72	-41	-11	-78	-34	16
SE England	-30	-3	37	-26	-2	32	-38	-15	13	-64	-40	-7	-64	-20	32
SW England	-18	5	29	-18	1	20	-32	-10	5	-70	-47	-22	-49	-10	21
Severn	-25	0	26	-20	0	16	-39	-11	6	-68	-43	-21	-55	-13	19
W Wales	3	21	42	3	12	36	-14	4	15	-67	-35	-12	-9	25	59
Dee	-6	13	26	-7	8	25	-21	-4	10	-62	-38	-21	-25	6	39
NW England	-1	18	57	-4	13	48	-18	2	29	-71	-33	-15	-21	24	76

ences in catchment flow response to climatic change inputs. Figure 6.10 shows the relationship between 95th precipitation, PET and Q1 changes, with catchments grouped by Runoff Coefficient classes. Catchments with relatively low runoff coefficients tend to show a higher sensitivity to the increasing PET. They are therefore more likely to see decreasing Q1 flows even with small (<5%) increases in heavy precipitation. These catchments are often drier catchments, and so heavy precipitation events may fill storage deficits rather than result in increased river flow. Other catchment properties, such as deep soils or permeable geology may also contribute to water being retained in the catchment. By contrast, catchments with high runoff coefficients show more sensitivity to changes in heavy precipitation, and very small (5%) increases in precipitation can lead to increases in Q1 of up to 25%. These are often wetter catchments, or catchments with other properties such as steep slopes or impermeable soils, where increases in heavy rainfall will directly result in increases in flood flows.

6.5 Discussion

6.5.1 Uncertainties in climate impacts on high flows

Our results highlight the importance of considering uncertainty in projections of climate change on flood flows. The selection of RCM parameters impacted not only the range of future changes

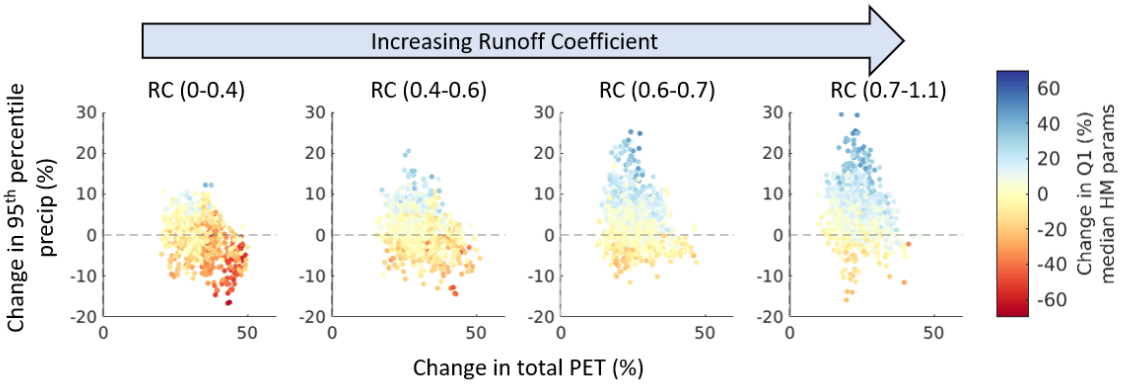


FIGURE 6.10. Runoff Coefficient (runoff divided by precipitation) vs flow sensitivity to climatic changes.

for each region (often disagreeing on the direction of change), but also variation in changes between regions, and to some extent the spatial pattern of changes across GB. This, combined with hydrological modelling uncertainties, resulted in the large ranges in future changes given in Table 6.1. The incorporation of multiple uncertainty sources, therefore, prevents an overconfident portrayal of climate change impacts on high flows, which could be misleading if used to inform future planning or policy decisions (Buurman and Babovic, 2016; Kundzewicz et al., 2018).

Previous studies found hydrological modelling uncertainties to be small relative to climate modelling uncertainties, especially when considering high flows (Chen et al., 2011; Velázquez et al., 2013). Our results generally support these findings, showing that the variation in future changes between RCMs is much larger than the variation between behavioural hydrological model parameter sets. However, we observed substantial hydrological modelling uncertainties for catchments in England, particularly for the Anglian river basin and drier catchments in the south-east. It is likely that interactions between the RCMs and hydrological model parameters also contribute to the total uncertainty, as it has previously been shown that interactions between uncertainty sources can account for 5-40% of the total uncertainty in hydrological climate change impacts studies (Bosshard et al., 2013). This emphasized that while uncertainties in future climate may dominate, uncertainties due to hydrological model parameters are not negligible.

There are many uncertainty sources that we were not able to incorporate. In addition to RCM and hydrological model parameters, sources of uncertainty in hydrological climate impact studies include the structure and parameterisation of the global climate model (GCM), bias correction methods, PE estimation equation, and hydrological model structure (Kay et al., 2009; Prudhomme and Davies, 2009; Wilby and Harris, 2006). The 12 RCM projections used in this

study were all driven by the same GCM (GC3.05), and for the RCP8.5 emissions scenario. This GCM has been shown to sample the warmer range of global outcomes (Lowe et al., 2019), and so combined with a single emissions scenario, we only sample the warmer range of possible climate outcomes. Therefore, whilst our results provide a useful indication of the range in future changes to high flow metrics across GB the true uncertainty ranges are likely to be much larger.

6.5.2 Future changes to high flows across Great Britain

Despite large uncertainties, some clear patterns of climate change impact on flooding across GB emerged. Projections indicated decreasing median flows (Q50) across all regions except for the Argyll and West Highland river basin regions. This decrease was likely due to reduced average precipitation and nationwide increases in PET projected by all the RCMs.

Increased flood magnitudes and frequency were projected for all RCMs along the west coast (excluding the southwest) and across most of Scotland, whilst decreasing flood flows were projected for the Anglian river basin region in east England using the median of all hydrological model parameter sets. These results are consistent with Collet et al. (2018), who found that hydro-hazard hotspots were likely to develop along the west coast and north-eastern Scotland. However, our results contrast with the study of Bell et al. (2016), which found relatively large increases in flood flows in the south and Anglian in particular. This contrast could be due to the different metric studied (Bell et al. (2016) showed percentage changes in 20-year return period floods, whilst we show changes in AMAX floods), or other methodological differences such as hydrological model or climate projections. We found hydrological modelling studies to be particularly large for the Anglian region and therefore increases in AMAX flows were within the total uncertainty range.

A limitation of this study is that the hydrological modelling framework did not include snow accumulation and melt processes. However, snow fractions are generally very low across Great Britain, with a median snow fraction of 0.01, except for catchments in northeast Scotland where it reaches a maximum of 0.17 (Coxon et al., 2020). The impact of including a snow module on climate change projections for peak flows was investigated by Bell et al. (2016). They found that across most of GB the inclusion of a snowmelt regime led to small percentage differences in peak flow changes of less than 6%. However, snowmelt processes were shown to be important for upland parts of GB, mainly in East Scotland, where the reduced presence of snow in the future could have a large impact on river flows. Therefore, the results of our study need to be interpreted with caution in these upland catchments.

6.5.3 Relationship between climate changes and hydrological response

It is often assumed that increases in extreme precipitation will lead to increases in flood flows (Sharma et al., 2018). However, whilst there is observational evidence of increasing precipitation extremes, there is no compelling evidence for any systematic increases in flooding which can be attributed to climate change (Hannaford, 2015; Watts et al., 2015). Understanding the link between changing precipitation and changing floods has, therefore, been highlighted as an important challenge for the hydrologic community (Sharma et al., 2018). Here we found that whilst there was a strong positive relationship between changes in heavy precipitation (as characterised by changes in the 95^{th} percentile precipitation) and changes in high flows (Q1), there were catchments where precipitation was increasing yet modelled flood flows were decreasing. These catchments were found to have large increases in PET – and therefore the impact of drier soils and increased storage deficits could have moderated the impact of increased heavy precipitation on river flows.

We found that the relationship between changes in heavy precipitation, total PET and changes to flood flows varied between river basin regions. The catchment runoff coefficient (average river flow divided by average precipitation) helped to explain this variation; for catchments with high runoff coefficients precipitation increases most directly related to increased flood flows, whilst catchments with low runoff coefficients showed a greater response to increasing PET. This in part relates to previous studies finding that there is a more direct link between heavy rainfall and high flows in wetter catchments (Charlton and Arnell, 2014; Ivancic and Shaw, 2015), as there is a general relationship between the runoff coefficient and catchment wetness. This highlights that it is important to recognise the complexities of flow change resulting from multiple climatic drivers and non-linear hydrological processes.

6.6 Conclusions

In this study we modelled climate change impact on the magnitude and frequency of high flows across 346 catchments in Great Britain, including both RCM and hydrological model parameter uncertainties for the first time at the national scale. The latest UK Climate Projections (UKCP18) were used to generate 12 spatially coherent and equally plausible time-series of precipitation and PET. These were then used to drive the DECIPHeR hydrological modelling framework, using 30 nationally consistent parameter fields. The resultant 360 future flow projections were used to investigate the range of changes in high flow magnitude and frequency between baseline (1985 - 2010) and future (2050 - 2075) scenarios, as well as the relationship between climatic changes and hydrological response.

Generally, results indicated increasing magnitude and frequency of flood flows for catchments

along the west coast of GB, and across most of Scotland. For western Scotland, region-average increases in annual maximum flows of up to 65% were projected. The Anglian and Thames river basins in eastern England generally showed decreasing flood magnitude and frequency. However, hydrological modelling uncertainty was high for these areas and therefore increases were also within the ensemble range.

Regional differences in high flow changes were found to relate to i) differences in climatic changes and ii) differences in catchment conditions during the baseline period as characterised by the runoff coefficient (total discharge/precipitation). A strong relationship was found between increasing heavy precipitation and increasing flood flows, alongside the moderating impact of increased PET. This relationship differed between catchments; catchments with high runoff coefficients were found to have a more direct response of flood flows to precipitation changes, whilst catchments with low runoff coefficients were more responsive to increased PET often resulting in very large reductions in Q1 flows (-50%) in areas with small (-5%) reductions in 95^{th} percentile precipitation.

Our results highlight the importance of considering uncertainties in climate impact studies. The variation between RCMs was the largest source of uncertainty, with differences in both the magnitude of projected changes for individual regions and the variability between regions. Hydrological modelling uncertainties were smaller, but still considerable for catchments in east and south-east England.

This chapter provides a national overview of projected future changes in median and higher flows across Great Britain, with the full ensemble range in projected changes given for each region. This information will be useful for decision-makers who have a role in managing and/or planning water in GB, for example in water companies, regulators and government.

CHAPTER 7

CONCLUSIONS, SUMMARY AND OUTLOOK

Section 7.1 of this chapter is adapted from the abstracts of three papers that are published, in review or in preparation:

1. Lane, R.A., Coxon, G., Freer, J. E., Wagener, T., Johnes, P.J., Bloomfield, J. P., Greene, S., Macleod, J. A., Reaney, S. M (2019). Benchmarking the predictive capability of hydrological models for river flow and flood peak predictions across over 1000 catchments in Great Britain. Hydrology and Earth System Sciences, 23, pp. 4011-4032.
2. Lane, R.A., Freer, J. E., Coxon, G., & Wagener, T. (in review). Incorporating Uncertainty into Multiscale Parameter Regionalisation to Evaluate the Performance of Nationally Consistent Parameter Fields for a Hydrological Model. Submitted to Water Resources Research.
3. Lane, R. A., Coxon, G., Freer, J., Seibert, J., and Wagener, T. (in preparation). UK Climate Projections on high flows across Great Britain, including hydrological modelling uncertainties.

This thesis aims to contribute to the grand challenge of developing model frameworks which are able to produce national-scale, yet locally relevant, hydrological predictions whilst representing modelling uncertainties. This chapter summarises the main conclusions of each research chapter, discusses the main contributions of the thesis as a whole and provides recommendations for future research.

7.1 Chapter Summaries

7.1.1 Research Chapter One: Benchmarking the predictive capability of lumped models across Great Britain

Benchmarking model performance across large samples of catchments is useful to guide model selection and future model development. Given uncertainties in the observational data we use to drive and evaluate hydrological models, and uncertainties in the structure and parameterisation of models we use to produce hydrological simulations and predictions, it is essential that model evaluation is undertaken within an uncertainty analysis framework. In this chapter, we benchmarked the capability of several lumped hydrological models across GB, focusing on daily flow and peak flow simulation. Four hydrological model structures from the Framework for Understanding Structural Errors (FUSE) were applied to over 1000 catchments in England, Wales and Scotland. Model performance was then evaluated using standard performance metrics for daily flows, and novel performance metrics for peak flows considering parameter uncertainty.

The lumped hydrological models were able to produce adequate simulations across most of GB, with each model producing simulations exceeding 0.5 Nash-Sutcliffe efficiency for at least 80% of catchments. All four models showed a similar spatial pattern of performance, producing better simulations in the wetter catchments to the west, and poor model performance in Scotland and southeast England. Poor model performance was often linked to the catchment water balance, with models unable to capture the catchment hydrology where the water balance did not close. Overall, performance was similar between model structures, but different models performed better for different catchment characteristics and metrics, as well as for assessing daily or peak flows, leading to the ensemble of model structures outperforming any single structure. This demonstrates the value of using multi-model structures across a large sample of different catchment behaviours. This research evaluates what conceptual lumped models can achieve as a performance benchmark, as well as providing interesting insights into where and why these simple models may fail. The large number of river catchments included in this study makes it an appropriate benchmark for any future developments of a national model of Great Britain.

7.1.2 Research Chapter Two: Developing nationally consistent parameter fields including uncertainty

Spatial parameter fields are required for hydrological models to represent the diverse range in hydrological processes across landscapes. These are often regionalised using transfer functions which relate parameters to spatial catchment attributes, leading to potentially large uncertainties due to observational data errors and imperfect transfer functions. Representing these uncertainties is important to understand the robustness of our model predictions, yet it remains a key challenge for large scale modelling studies. This research chapter extends the multiscale

parameter regionalisation (MPR) technique to consider parameter uncertainties. We then evaluate this method of producing nationally consistent parameter fields, which maintain a constant relationship between model parameters and catchment attributes, across 437 catchments in Great Britain (GB). By sampling multiple transfer function parameters, we produce thousands of possible model parameter fields which are constrained within an uncertainty framework. This is compared to spatially homogeneous parameter sets constrained for individual catchments.

The nationally consistent MPR parameter fields performed well (KGE* > 0.75) across 60% of catchments. Performance is similar or better than catchment-constrained parameters (KGE* drop < 0.1) across 82% of catchments. Advantages of our national parameter fields included (1) improved representation of flows within catchments, (2) more robust performance between calibration and evaluation periods and (3) spatial parameter fields reflecting hydrologically meaningful variation in catchment characteristics. By including uncertainties, we showed that hydrographs produced using MPR have smaller uncertainty bounds which are better able to encompass flows than homogeneous Monte-Carlo constrained parameters. As the first application of MPR to both the DECIPHeR modelling framework and GB, we developed transfer functions and identified key catchment attributes to constrain model parameters, which are transferable to other models alongside the framework for addition of uncertainty. Methodologies presented here are therefore informative for future regionalisation efforts in GB and elsewhere.

7.1.3 Research Chapter Three: Climate change impact on high flows across Great Britain, including modelling uncertainties

Climate change may significantly increase flood risk across Great Britain, but there are large uncertainties in both future climatic changes and how these propagate into changing river flows. Modelling studies are therefore required to help understand climate change influence on high flows, alongside the uncertainties in these projections. Here, the impact of climate change on the magnitude and frequency of high flows is modelled for 346 larger (>144 km^2) catchments across GB using the new UK Climate Projections (UKCP18) and the DECIPHeR hydrological modelling framework. This provides the first spatially consistent GB projections to include both climate ensembles and hydrological model parameter uncertainties.

Results indicated an increase in the magnitude and frequency of high flows along the west coast of GB in the future (2050 - 2075), with increases in annual maximum flows of up to 65% for west Scotland. All flow projections had large uncertainties, and whilst the RCMs were the largest source of uncertainty overall, hydrological modelling uncertainties were considerable in east and south-east England. Regional variation in flow projections were found to relate to i) differences in climatic change predictions and ii) catchment conditions during the baseline period as characterised by the runoff coefficient (mean discharge divided by mean precipitation).

Importantly, increased heavy-precipitation events (defined by an increase in 99^{th} percentile precipitation) did not always result in increasing flood flows for catchments with low runoff coefficients, highlighting the varying factors leading to changes in high flows and the complex interplay between catchment characteristics, PET and precipitation. These results provide a national overview of climate change impacts on high flows across Great Britain, which will inform climate change adaptation, whilst also highlighting the need to account for uncertainty sources when modelling climate change impact on high flows.

7.2 Synthesis

National scale hydrological models are required to guide effective water management in the face of large-scale pressures such as climate change. However, the development of national scale modelling frameworks is a difficult task, requiring model structures and parameters that can reflect the heterogeneity of hydrological processes across the landscape, robust model evaluations and methods to characterise modelling uncertainties. This thesis has addressed these challenges with a focus on national-scale hydrological modelling across GB, through benchmarking the performance of lumped model structures, developing and evaluating a method of constraining national model parameter fields over gauged and ungauged catchments, and using this national model setup to provide an evaluation of climate change impacts on median and higher flows including model uncertainties. The key scientific contributions of this thesis are:

1. improved understanding of how climate and catchment characteristics impact hydrological model performance
2. the production of a benchmark set of model simulations, performance metrics, and calibrated parameter sets that are openly available and can be used to guide future modelling efforts across GB
3. the development of a nationally parameterised, spatially oriented, flexible hydrological model that enables predictions at high resolution for any gauged or ungauged catchment with uncertainties
4. a new framework for incorporating uncertainty analysis into the estimation of spatial parameter fields
5. identification of relevant catchment attribute datasets and synthesis of pedotransfer equations that are critical for model parameterisation
6. the first national climate change projections for high flows across GB to include both hydrological and climate model parameter uncertainties.

These contributions directly align with many current research goals, each of which will be discussed in more detail below, which include: pursuing the large-sample hydrology approach to derive generalisable conclusions (Addor et al., 2020; Andréassian et al., 2006; Gupta et al.,

2014); developing hydrological models which can provide locally relevant predictions across large domains (Archfield et al., 2015; Bierkens et al., 2015; Wood et al., 2011); providing predictions in ungauged basins (Blöschl et al., 2013; Hrachowitz et al., 2013; Sivapalan et al., 2003); effectively benchmarking and evaluating hydrological models (Clark et al., 2016a; Klemes, 1986; Seibert, 2001; Seibert et al., 2018); and incorporating and communicating hydrological model uncertainties (Beven and Binley, 1992; Freer et al., 1996; Mcmillan et al., 2012; Pappenberger and Beven, 2006).

Firstly, contributions from this thesis answer calls for the need to pursue a large-sample hydrology approach, to derive generalisable relationships and complement the many in-depth hydrological studies for a small number of catchments (Andréassian et al., 2006; Gupta et al., 2014). The large-sample approach to hydrological modelling is a key theme throughout this thesis, with all research chapters contributing to the large-sample hydrology literature. The large-sample approach was used to characterise the relationship between climate/catchment characteristics and model performance (Research Chapter one), to derive robust links between spatial fields of model parameters and catchment attributes (Research Chapter two), to demonstrate the relationship between changes in precipitation/PET and high flows (Research Chapter three), and to ensure a thorough evaluation of model performance (all research chapters). This led to general insights, such as models tend to perform better for wetter catchments and catchments with low baseflow contributions, the catchment water balance not closing is a common reason for model failure, lumped models calibrated using Nash-Sutcliffe efficiency generally underpredict annual maximum flows, hydrogeology data is important alongside soils and land-use datasets in constraining hydrological model parameters, and whilst there is a positive relationship between heavy (95^{th} percentile) precipitation change and the change in high flows (Q1) this is moderated by catchment characteristics and potential evapotranspiration. These findings support existing literature, for example Coxon et al. (2014) and McMillan et al. (2016) also report better model performance in wetter catchments, with the large number of catchments used in all research chapters and the multi-model approach in Research Chapter one adding weight to these conclusions.

Secondly, this thesis contributes to the grand challenge of developing hydrological models which are capable of providing locally relevant predictions across large domains (Archfield et al., 2015; Beven et al., 2015; Bierkens et al., 2015; Wood et al., 2011). Archfield et al. (2015) summarise some of the key challenges for the development of continental-scale models, including developing model structures which adequately represent the dominant hydrological processes, and ensuring a physical basis to parameter estimation. Research Chapter one contributes to the first of these challenges, by evaluating to what extent simple, lumped models could represent the dominant hydrological processes, and providing suggestions for additional processes to improve model performance (e.g. need for a snow melt and accumulation module in mountainous

Scottish catchments, representation of inter-catchment groundwater flows in the southeast GB catchments overlaying aquifers, possible inclusion of reservoirs and other human influences). In addition, it identified structural differences between the models which led to certain models better representing processes, such as the strong non-linear upper storage zone in the ARNO-VIC model which enabled it to simulate baseflow dominated catchments where other models failed. Research Chapter two contributed to the second of these challenges, by developing parameter fields based on the physical relationships between model parameters and spatial geophysical data. This extended the multiscale parameter regionalisation literature (Kumar et al., 2013a; Mizukami et al., 2017; Samaniego et al., 2010, 2017) by applying the method to a new model, new location, new datasets, and developing new transfer functions to relate parameters to geophysical data. These methodological choices (i.e. transfer functions and datasets used for each parameter) will be helpful for future studies regionalising models with related parameters across GB and elsewhere. The MPR approach was shown to be successful with realistic variation in parameter fields and generally good performance (non-parametric KGE >0.75 across 60% of catchments), hence ensuring the national model can provide robust and locally relevant predictions.

Contributions from this thesis will also help to improve national modelling for GB, through identification of areas where model performance needs improvement and the development of a flexible national modelling framework which can be used for national simulations. Identifying and communicating model failures is an important step in understanding model limitations and identifying targeted areas for model improvement, yet literature tends to focus on positive results and failure stories are rarely communicated (Andréassian et al., 2010). Throughout this thesis model failures have been identified. Clusters of catchments in northern/ north-eastern Scotland, and catchments overlaying permeable bedrock in southeast England were identified in Research Chapter one as areas where lumped hydrological models often failed to produce good simulations. Alongside this, a general east/west divide in lumped model performance was observed, with models tending to perform better in wetter catchments to the west. These patterns were also observed in the evaluation of a spatially distributed model in Research Chapter two, and large overpredictions of AMAX flows (in some cases over 100%) were observed in southeast England from the climate-hydrological modelling change presented in Research Chapter three. These clusters of catchments have therefore been identified as areas to focus on for improved overall GB modelling. This thesis has also developed a national model for GB, which will complement existing GB models. Current models for GB include uncalibrated, spatially distributed models such as G2G (Bell et al., 2009) and the physically based SHETRAN model (Lewis et al., 2018), which tend to be applied deterministically, and semi-distributed or lumped models which tend to require catchment calibration and therefore do not have consistent parameter values across catchments (e.g. CLASSIC (Crooks et al., 2014; Crooks and Naden, 2007) and PDM (Moore, 2007; Prudhomme et al., 2013b)). DECIPHeR-MPR, developed within this thesis, fills the need

for a spatially distributed model which considers parameter uncertainties. It adds a flexible model framework capable of spatially distributed simulations, which has nationally consistent parameter fields that can be applied within a parameter uncertainty framework over gauged and ungauged catchments. The flexible spatial discretisation of the model is a further advantage - meaning that the model can be run at a high spatial resolution, capturing key landscape features, and providing simulations that are relevant for local as well as national scales.

Thirdly, this thesis promotes a thorough evaluation of hydrological models and supports efforts to benchmark hydrological model performance. Throughout the three research chapters hydrological model performance has been critically evaluated, and all research chapters follow good practice in openly discussing model flaws and highlighting areas where model results should be treated with caution. Research chapter one further contributes to literature suggesting that lumped, conceptual models can provide a good performance benchmark which improved models can be evaluated against (Seibert, 2001; Seibert et al., 2018), by providing openly accessible model time-series and performance scores across GB which can be used for this purpose.

Finally, the thesis contributes to the research goal of improving uncertainty characterisation and communication in large-scale modelling studies (Archfield et al., 2015; Beven and Freer, 2001a; Pappenberger and Beven, 2006). The thesis includes the first multi-model uncertainty framework across Great Britain (Research Chapter One), a framework to include parameter uncertainties in the estimation of spatially distributed national parameter fields (Research Chapter Two), and the first nationally consistent evaluation of future flows to include hydrological model and regional climate model parameter uncertainties (Research Chapter Three). These demonstrate how model uncertainties can be included in national-scale, large-sample studies, and provide frameworks/modelling tools to include parameter uncertainties. The DECIPHeR-MPR framework can produce spatially oriented, high resolution simulations anywhere within GB with uncertainty estimates. It could therefore be used to explore modelling uncertainties for a range of applications.

The results of this thesis will also be useful for practitioners who use hydrological models or model output, particularly for flood management purposes due to the focus on higher flows. Thesis outputs that will be helpful for practitioners implementing hydrological models at local to national scales include a parameter library of behavioural parameters for the FUSE model across GB catchments (Research Chapter One), and the DECIPHeR-MPR framework (Research Chapter Two). Additionally, Research Chapter One demonstrates the value of applying an ensemble of model structures, as it was found that each model structure had different advantages (for example, of the FUSE models SACRAMENTO produced the highest overall NSE values, TOPMODEL had the least bias, ARNO/VIC was the only model to capture baseflow dominated catchments, and

PRMS showed the most skill in capturing peak flows), and the ensemble performed better than any single model structure. For users of model outputs, the model evaluation carried out within all research chapters emphasises where model results have poor skill and should be treated with caution. Finally, Research Chapter Three provides regional projection ranges for climate change impacts on median and higher flows, which will be useful for climate change adaptation plans.

7.3 Recommendations for future research

This thesis highlights various opportunities for future research. Model evaluations carried out as part of this thesis have highlighted areas for focused model improvement. Research Chapter One identified areas where all tested model structures failed to produce any acceptable simulations, notably catchments in northern and north-eastern Scotland and those on permeable bedrock in south-eastern England. Reasons suggested for these model failures included no representation of snow accumulation and melt processes and hydro-electric power modifications to the flow regime for Scotland, and lack of groundwater representation, inter-catchment groundwater flows, and human influences in the south-east. Future studies could look further into identifying the reasons for model failure in these catchments, taking an in-depth approach focused on the hydrology of individual catchments which was not possible within our national overview. Additionally, future studies could focus on improving modelling efforts for these catchments, using our results as a benchmark of model ability to improve upon.

Incorporating human-water interactions within hydrological models has been identified as an important research goal. In addition to the hydro-electric power modifications in Scotland and human influences across the southeast discussed above, the widespread presence of reservoirs was found to impact our ability to model flows (as discussed in Methods section 3.3.2.1 and Research Chapter One). There is a recognised need for hydrological models which incorporate human impacts on the hydrological cycle, although developing such models is challenging (Wada et al., 2017). The DECIPHeR framework lends itself well to the addition of human influences - as it is able to explicitly characterise connectivity and fluxes across landscapes, and has a flexible definition of model structure enabling different model structures to be applied across the catchment. The framework has previously been used to explore the impact of abstractions on low simulated low flows across the Thames catchment (Coxon et al., 2017). Future work could therefore develop and test alternative model structures for the DECIPHeR framework to better represent processes in areas with known human-water interactions.

Research Chapter Two demonstrated how the MPR technique could be used to develop spatial parameter fields for the DECIPHeR model. These parameter fields are strongly dependent on the spatial catchment attribute data used to create them, with previous applications of MPR

stressing the importance of good quality, high-resolution catchment attribute data (Livneh et al., 2015; Samaniego et al., 2017). For this initial evaluation of the suitability of the method, we therefore aimed to use the best possible national products, which included data that was not openly accessible such as the LandIS national soils map. This limits the uptake and use of the parameterised model and model outputs and it would therefore be desirable to use open-access data. The value of using open access data products has been highlighted in recent publications (Addor et al., 2020; Archfield et al., 2015; Gupta et al., 2014), and there are open-access datasets available at national to global scales (Archfield et al., 2015). An interesting follow-up study could therefore evaluate the suitability of these open-access datasets for parameter regionalisation compared to the high-resolution national datasets used in this thesis.

Future research could also focus on expanding and improving upon the uncertain multiscale parameter regionalisation of DECIPHeR presented in Research Chapter Two. Firstly, whilst we focused on the uncertainties relating to the global parameters, there are many other uncertainties such as the form of the pedotransfer equations, upscaling operators and underlying catchment attribute data used for regionalisation which could be incorporated. In Section 3.5.3 (Figure 3.16) we demonstrated that the choice of pedotransfer function resulted in differences in porosity maps - it would be interesting to understand how these uncertainties in underlying data propagate into modelled discharge uncertainties and to understand the relative importance of different decisions. This would complement current research activities, such as work investigating techniques to generate multiple possible transfer functions for parameter regionalisation (Klotz et al., 2017), and evaluation of the influence of two different soil datasets on model performance (Livneh et al., 2015). Secondly, we demonstrated that the regionalisation of DECIPHeR is an improving process, through the addition of hydrogeology data to improve the regionalisation method (Research Chapter Two and Appendix E). There is therefore scope for improving the method further, and adding additional data to the regionalisation approach where it is found to be needed.

In addition to considering uncertainties within the MPR methodology, future work could explore how to incorporate uncertainties in the observational data into the DECIPHeR-MPR framework. This could include using uncertain rainfall/PET fields to drive the model, and evaluating the model against uncertain river flows. This could build on research quantifying discharge uncertainties across British gauging stations (Coxon et al., 2015), or apply more simple metrics for considering discharge uncertainties such as those applied in Research Chapter One which used the average error in annual maximum flows. Including these additional uncertainties would better reflect the challenges of modelling real world systems, and further improve our understanding of model skill over observational errors. The DECIPHeR-MPR framework makes this type of analysis possible, by presenting a computationally efficient national model framework.

Finally, DECIPHeR-MPR provides a flexible framework which could be used to explore a wide range of hydrological problems. In Research Chapter Three we demonstrated how this framework could be used to explore climate change impact on high flows. This framework could be applied to explore other spatial hydrology applications such as land use change, or to model low flows and droughts.

APPENDIX A

ROSANNA LANE - CV

Contact Information:

Name: Rosanna (Rosie) Alice Lane
Email: Rosanna_Lane.2011@my.bristol.ac.uk
LinkedIn: www.linkedin.com/in/rosannalane

Summary:

I am a researcher interested in hydrological modelling, developing and applying national scale models, and climate change impacts on river flows. My background is in geographical sciences, where I specialised in hydrology, earth system modelling and spatial statistics. Having enjoyed learning about hydrological modelling during my undergraduate degree, I chose to pursue this for my master's dissertation and subsequently through a PhD. I am currently working as a research associate hydrological modeller at the UK Centre for Ecology and Hydrology (UKCEH), working on a range of projects within the hydro-climate risks group. The following page provides a very brief overview of my educational background, for more detail please see my LinkedIn profile.

Education

Sept 2016 -	**PhD in hydrology** *University of Bristol* Supervisors: Jim Freer, Gemma Coxon and Thorsten Wagener Focused on national-scale hydrological modelling across Great Britain.
Sept 2015 – 2016	**Postgraduate School in Hydroinformatics (1st)** *University of Exeter* Attended a CIWEM-accredited postgraduate year as part of the WISE-CDT programme, taking a range of taught modules related to hydroinformatics.
Sept 2011-2015	**MSci Geography (1st)** *University of Bristol* Specialised in hydrology, environmental change, spatial/statistical modelling, and environmental policy. Completed a final year project on "Modelling the impact of climate change on flood hazard for 3 UK catchments, with assessment of hydrological model uncertainty."

Work Experience

Jan 2020 -	**Hydrological/ Mathematical modeller** *UK Centre for Ecology and Hydrology (UKCEH)* Involved in a range of projects including: acting as a national river flow archive rep for northeast England and Yorkshire, generating model forcing data from UKCP18 climate outputs for the enhanced future flows and groundwater (eFLaG) project, and analysing modelled climate change impact on future floods and low flows as part of the national capability UK-SCAPE project.
Summer 2019	**Research visit** *University of Zurich* Opportunity to collaborate with Jan Seibert as part of my PhD.
2015 – 2019	**Teaching Support assistant** *School of Geographical Sciences, University of Bristol* Demonstrated modules including statistics using R and master's level spatial data analysis using ArcGIS.

Awards

2017	Peter Wolf Symposium best poster award
2016	WISE-CDT award for the highest postgraduate school grades
2015	Les Hepple prize for highest overall marks in my graduating year

APPENDIX B

SUPPLEMENT TO RESEARCH CHAPTER ONE: THE RELATIONSHIP BETWEEN CATCHMENT CHARACTERISTICS AND MODEL PERFORMANCE

This section has been published as supplementary material to a research article in Hydrology and Earth System Sciences. This analysis was carried out by Rosanna Lane in response to reviewer comments.

Citation: Lane, R.A., Coxon, G., Freer, J. E., Wagener, T., Johnes, P.J., Bloomfield, J. P., Greene, S., Macleod, J. A., Reaney, S. M (2019). Benchmarking the predictive capability of hydrological models for river flow and flood peak predictions across over 1000 catchments in Great Britain. Hydrology and Earth System Sciences, 23, pp. 4011-4032. https://doi.org/10.5194/hess-23-4011-2019

Research chapter one looked at the relationship between the catchment wetness index, runoff coefficient and model performance. These attributes were selected as they strongly related to model performance and explained differences between catchments. Here, we provide additional plots looking at the relationship between model performance and many different catchment characteristics to demonstrate how other catchment attributes impact model performance. These characteristics were either taken from the hydrometric register or calculated from the model input data timeseries (Centre for Ecology and Hydrology, 2016; Marsh and Hannaford, 2008; Robinson et al., 2015a).

Figures B.1 – B.4 are scatter plots looking at the relationship between model performance (assessed using NSE, bias, error in standard deviation and correlation respectively) and different

catchment attributes. Figures S5 onwards are plots looking at interactions between different catchment attributes and model performance.

Some links between catchment attributes and model performance can be seen from Figures B.1 – B.4. Firstly, small catchments (< 200 km^2) tend to have more variable NSE scores (both high and low), whilst large catchments (> 3000 km^2) are easier to model. This is seen with all the decomposed metrics potentially indicating that daily data is not able to capture flow variation in small catchments. Secondly, baseflow dominated catchments (BFI > 0.7) are more likely to gain very low NSE values (although some high BFI catchments can be simulated well). Interestingly, BFI seems to have a relationship with error in the standard deviation, with baseflow dominated catchments the only catchments where the best simulations tend to overpredict variation. This could be due to groundwater dampening variation in flows. Thirdly, gauge elevation seems to cap overall model performance, with higher elevation gauges unable to achieve performance scores as high as low elevation gauges. Finally, it is surprising that urbanisation does not seem to decrease model performance.

From figures B.5 onwards we can see that the worst performing catchments in terms of Nash-Sutcliffe efficiency are grouped being small catchments less than 120 km^2, with elevations below 125m, mid to high BFIs (> 0.5), low annual rain less than 1000 mm and annual runoff values which differ from other catchments with similar annual rainfall totals. Poor NSE 0.5 is achieved for wetter catchments (annual rain > 1200 mm), which have relatively low annual runoff generally less than 900mm. Many have flow attenuation from reservoirs and lakes, and for these catchments correlation is poor.

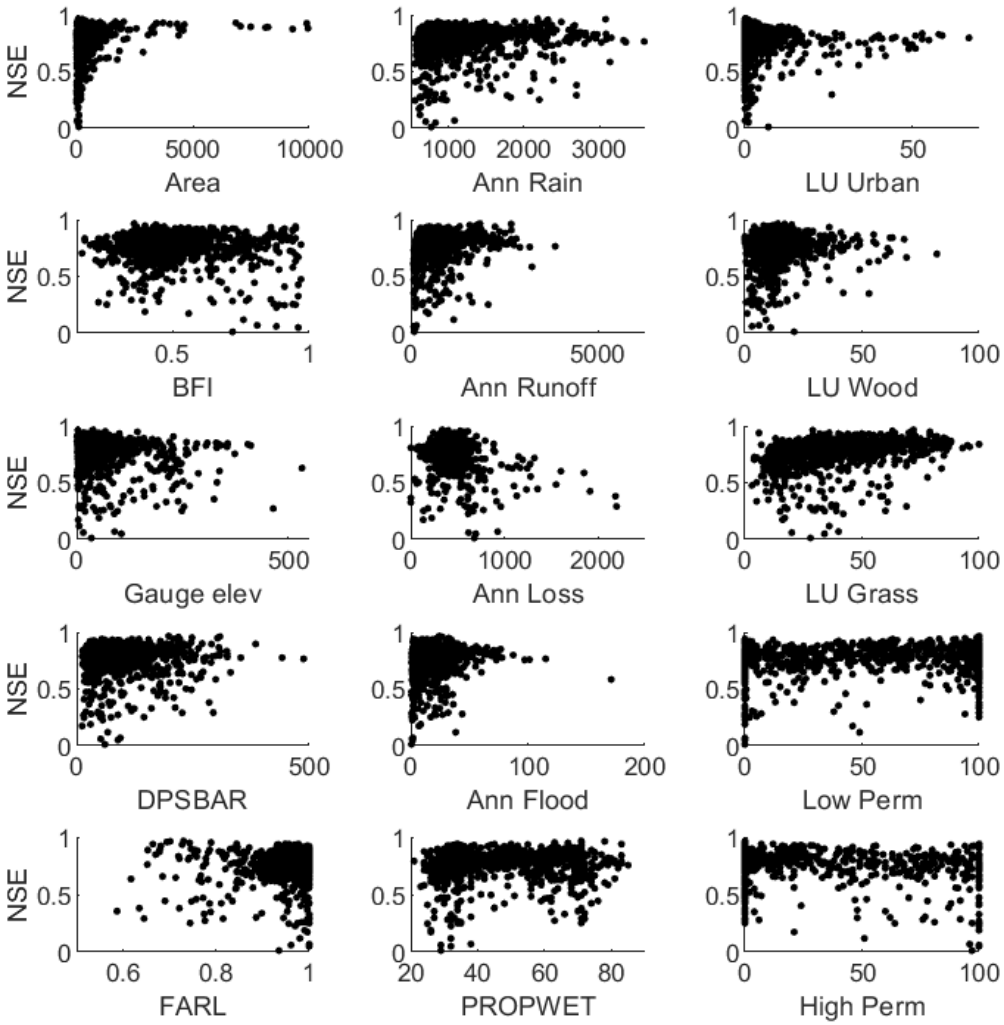


FIGURE B.1. Relationship between NSE and a selection of 15 catchment descriptor variables. Column 1 gives general catchment attributes from the hydrometric register (Marsh and Hannaford, 2008). These are catchment area (km^2), Baseflow index (BFI), Gauge elevation (m above sea level), mean drainage path slope (DPSBAR) which indicates overall catchment steepness in metres per kilometre, and flood attenuation by reservoirs and lakes (FARL) where values close to 1 indicate the absence of flow attenuation and values below 0.8 indicate a substantial influence. Column 2 gives hydroclimatic attributes calculated from our data, and proportion catchment is wet (PROPWET) from the hydrometric register. Annual Loss is Rainfall-Runoff, whilst Annual flood is the Median Annual maximum flood peak, and all are reported in mm. Column 3 gives land-use and bedrock permeability descriptors (%), also from the UK hydrometric register.

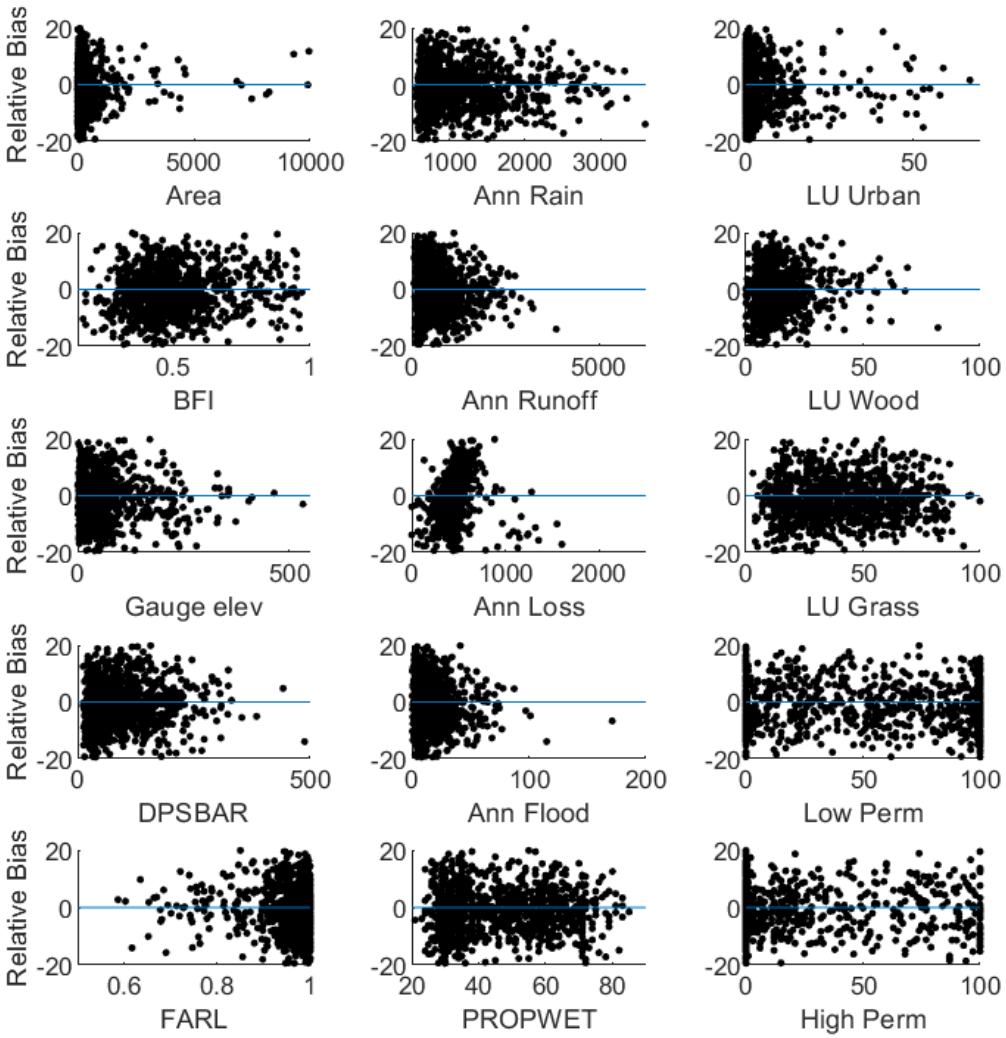


FIGURE B.2. Relationship between bias and a selection of 15 catchment descriptor variables, as in Figure B.1.

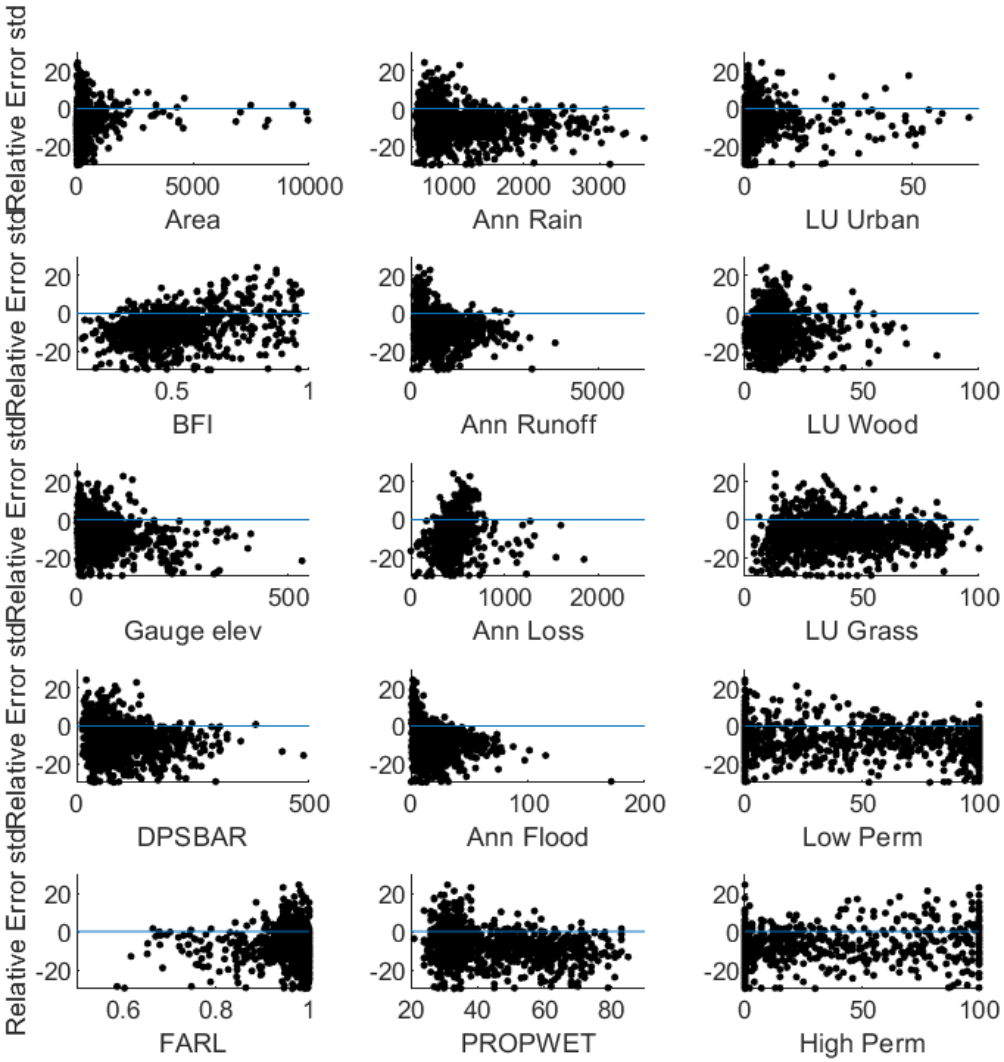


FIGURE B.3. Relationship between error in standard deviation and a selection of 15 catchment descriptor variables, as in Figure B.1.

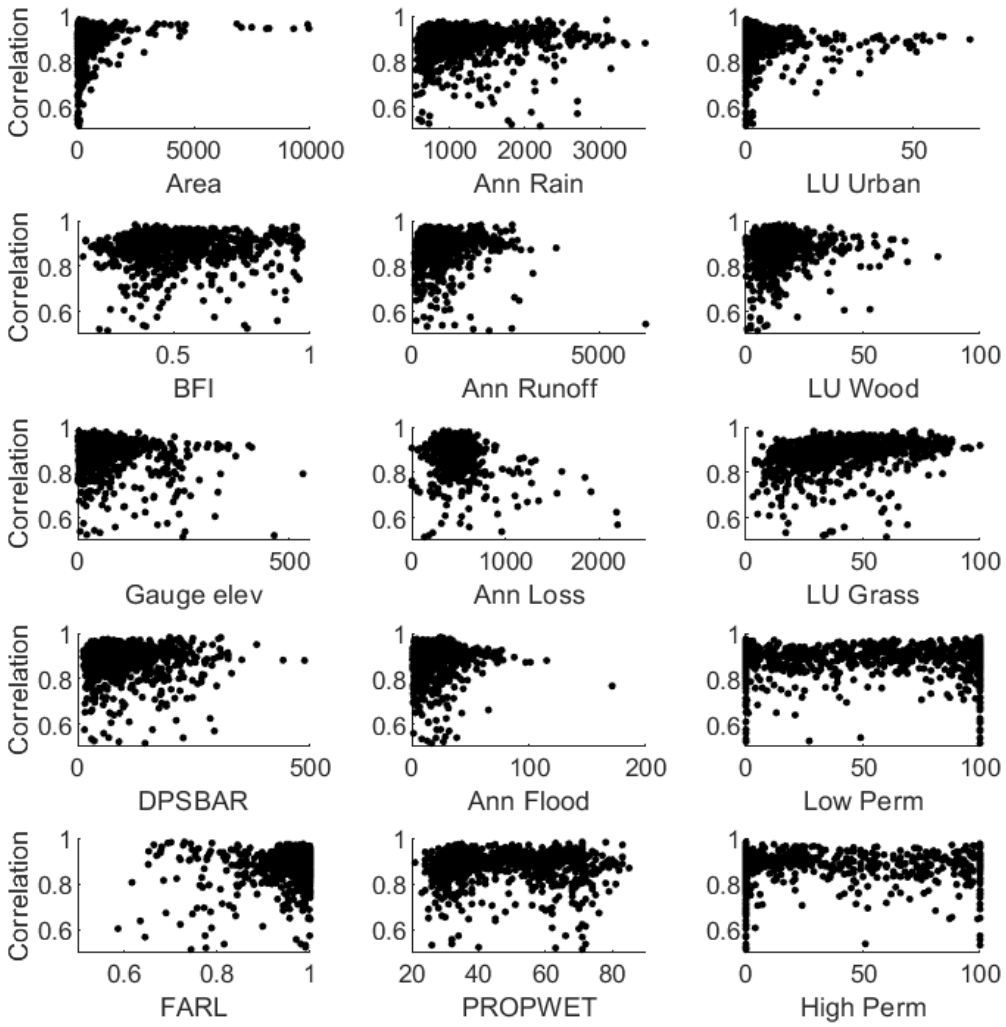


FIGURE B.4. Relationship between correlation and a selection of 15 catchment descriptor variables, as in Figure B.1.

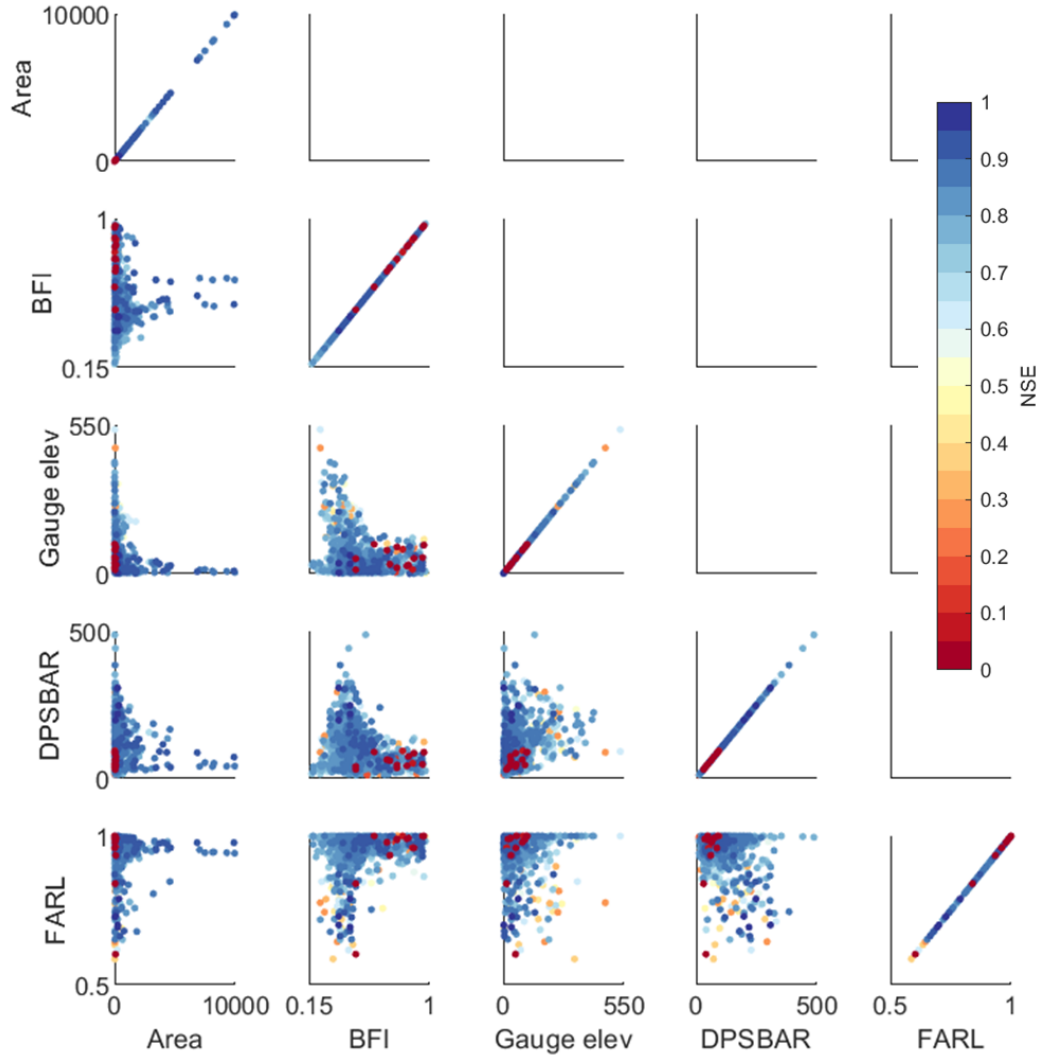


FIGURE B.5. Relationship between general catchment characteristics, coloured by model ensemble NSE score for that catchment. Column 1 gives general catchment attributes from the hydrometric register (Marsh and Hannaford, 2008). These are catchment area (km^2), Baseflow index (BFI), Gauge elevation (m above sea level), mean drainage path slope (DPSBAR) which indicates overall catchment steepness in metres per kilometre, and flood attenuation by reservoirs and lakes (FARL) where values close to 1 indicate the absence of flow attenuation and values below 0.8 indicate a substantial influence. Column 2 gives hydroclimatic attributes calculated from our data, and proportion catchment is wet (PROPWET) from the hydrometric register. Annual Loss is Rainfall-Runoff, whilst Annual flood is the Median Annual maximum flood peak, and all are reported in mm. Column 3 gives land-use and bedrock permeability descriptors (%), also from the UK hydrometric register.

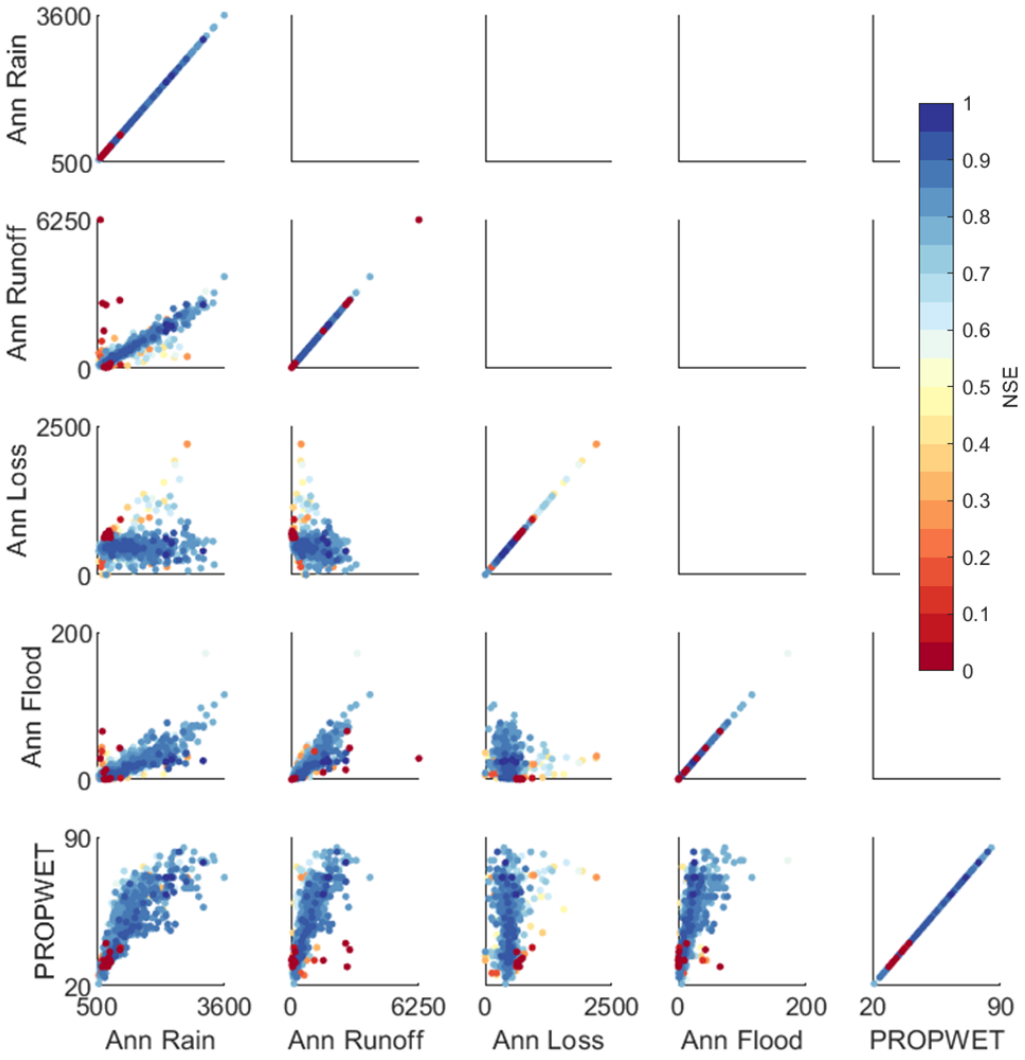


FIGURE B.6. Same as figure B.5, but this time looking at hydroclimatic catchment descriptors. Annual rainfall (mm), annual runoff (mm), annual loss (mm) and mean annual maximum flood (mm), were all calculated from the model input data used in this study. PROPWET is a measure of the percentage of time soils are wet, as calculated by the UK hydrometric register (Marsh and Hannaford, 2008).

APPENDIX C

SUPPLEMENT TO RESEARCH CHAPTER TWO: DECIPHER_MPR CODE

C.1 Code description

This appendix includes Fortran code developed to integrate the MPR parameterisation scheme within the DECIPHeR model framework. The DECIPHeR model code is open-source and freely available, with the latest version available at `https://github.com/uob-hydrology/DECIPHeR`, and a full model description given in Coxon et al. (2019).

The data used for model parameterisation within this thesis is not open access, and therefore a working DECIPHeR-MPR version cannot be made available. However, the code used is presented here, in the hope that it is informative for future regionalisation efforts. This was all written in Fortran so that it could be integrated with DECIPHeR. The heirarchy of modules is shown in Figure C.1. The dyna_main module is part of the original DECIPHeR code (Coxon et al., 2019), but this version has been heavily modified to call the MPR modules and is therefore presented here.

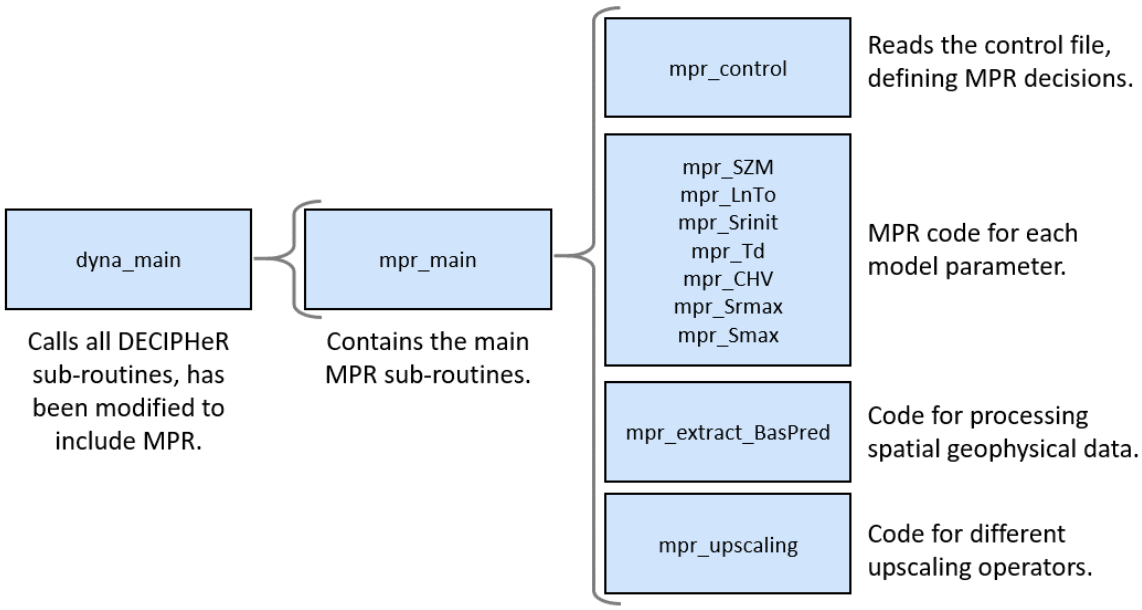


FIGURE C.1. Diagram showing the level of different DECIPHeR-MPR modules. The dyna_main module is the highest level module, which runs DECIPHeR calling all relevant sub-routines. The mpr_main module contains the key sub-routines for MPR in the order that they are applied. This calls upon sub-routines in the modules listed on the far right.

C.2 dyna_main.f90

```
1  !
2  !  ================================================================
3  !  DECIPHeR VERSION 1
4  !
5  !  Bristol University 2018 (Gemma Coxon, Jim Freer, Rosie Lane and Toby Dunne)
6  !  Based on fortran 77 version of dynamic TOPMODEL produced in
7  !  Lancaster University 12/01/00 (Keith Beven & Jim Freer)
8  !  Migrated to std=F2003 Toby Dunne 25/01/2015
9  !  Modified extensively by Gemma Coxon 2016-2018
10 !  Integrated with multiscale parameter regionalization - Rosie Lane 2017-2019
11 !
12 !  ================================================================
13 !
14 program main
15
16     use dyna_common_types
17     use dyna_project
18     use dyna_random
19     use dyna_tread_dyna
20     use dyna_inputs
21     use dyna_mc_setup
```

```
22      use dyna_modelstruct_setup
23      use dyna_genpar
24      use dyna_main_loop
25      use dyna_file_open
26      use dyna_read_routingdata
27
28      !use dta_utility
29      use dta_route_processing
30      use dta_riv_tree_node
31
32      use mpr_main
33      use mpr_extract_BasPred
34      use mpr_control
35      use mpr_errors
36      use mpr_upscaling
37      use mpr_utility
38      use mpr_random
39
40      !MPR settings and functions for each DynaTOP parameter
41      use mpr_SZM
42      use mpr_LnTo
43      use mpr_SRmax
44      use mpr_SRinit
45      use mpr_CHV
46      use mpr_Td
47      use mpr_Smax
48
49      implicit none
50
51      !  Local variable declares
52      integer :: seed_1, seed_2
53      integer :: i
54      integer :: nac
55      integer :: nstep
56      integer :: num_rivers
57
58      ! allocated from 'Input'
59      double precision, dimension(:,:), allocatable :: pe_step
60      double precision, dimension(:), allocatable :: qobs_riv_step_start
61      double precision, dimension(:,:), allocatable :: r_gau_step
62
63      ! allocated after 'Input'
64      type(dyna_hru_type), dimension(:), allocatable :: dyna_hru
65      type(dyna_riv_type), dimension(:), allocatable :: dyna_riv
66
67      ! MULTI POINT RIVER ROUTING
68      ! Toby Dunne April 2016 + GC June 2016
69      ! these type are defined in modules as part of the dta files
70      type(route_river_info_type) :: route_riv
71      type(route_time_delay_hist_type) :: route_tdh
72      integer :: cat_route_vmode
73      integer :: routing_mode
74      integer, dimension(:), allocatable :: node_to_flow_mapping
75      character(900) :: out_dir_path
76
```

```
77      ! Kinematic solution parameters and time step
78      doubleprecision :: dt
79      doubleprecision :: acc
80      doubleprecision :: wt
81      integer :: ntt
82      doubleprecision :: dtt
83      double precision, dimension(:,:), allocatable :: rivers !(nac, n_riv)
84      double precision, dimension(:), allocatable :: sum_ac_riv  !
85      logical :: new_init, print_output, new_kine
86      character(1024):: arg
87      character(1024):: auto_start_file
88
89      !! ADDED VARIABLES FOR MPR
90      !key filepaths and folder locations
91      character(len=1024)      :: MPRfolder
92      character(len=1024)      :: fpath_HRUmap
93      character(len=1024)      :: folder_input
94      character(len=1024)      :: folder_output
95      character(len=900)       :: folder_outfull
96      character(len=1024)      :: fname_control
97      character(len=1024)      :: fname_filemgr
98      character(len=1024)      :: fpath_gp_file
99      character(len=1024), dimension(18)    :: fnames_baspred
100
101
102     !variables in the control file
103         ! number of parameter files to create (n_pm_maps)
104         !starting seed for generation of global params
105         !save_pm_map: set to 1 if user wants to save parameter maps
106         !save_bp_maps : set to 1 if user wants to save basin predictor maps
107     integer                  ::  n_pm_maps
108     integer                  ::  start_seed
109     integer                  ::  save_pm_maps
110     integer                  ::  save_bp_maps
111
112         !pedo-transfer eq. nums for SZM, LnTo, SRmax, SRinit, CHV, Td, Smax
113     integer, dimension(7)    ::  pedo_tf_all
114
115     !parameter min, default and max values from parameter file
116     double precision, dimension(7,3):: Params_range
117
118     !variables defining HRU map and parameter maps
119         !stats_HRU: variable comprising HRU map xll, yll, cellsize & nodata value
120     double precision, allocatable, dimension(:,:) :: HRU_map
121     integer                          :: nHRUS            !number of HRUs
122     double precision, dimension(4)   :: stats_HRU
123
124     !basin predictor maps needed - logical arrays with true values if maps required
125     logical, dimension(18) :: req_bp    !Sand, silt, clay and OM rows, split by depth columns
126
127     !basin predictor maps
128         !soilmusiddata is tables containing depth decline data for each soil series and musid.
129         !bp_root_depths is landcover class rooting depth table
130     real, allocatable, dimension(:,:,:)     :: bp_maps
131     real, allocatable, dimension(:,:)    :: sand_0_10
```

```
132     real, allocatable, dimension(:,:)    :: silt_0_10
133     real, allocatable, dimension(:,:)    :: clay_0_10
134     real, allocatable, dimension(:,:)    :: orgm_0_10
135     double precision, allocatable, dimension(:,:,:) :: soilmusiddata
136     real, allocatable, dimension(:,:)    :: bp_root_depths
137
138     !global parameters
139         !n_gp_all is a vector of number of global params for all params
140         !glob_pms_* has dimensions of (glob pm number, 1:n_pm_maps)
141         !up_pms is a vector of upscaled params
142     integer, dimension(7)                              :: n_gp_all
143     double precision, allocatable, dimension(:,:)    :: glob_pms_SZM
144     double precision, allocatable, dimension(:,:)    :: glob_pms_LnTo
145     double precision, allocatable, dimension(:,:)    :: glob_pms_SRmax
146     double precision, allocatable, dimension(:,:)    :: glob_pms_SRinit
147     double precision, allocatable, dimension(:,:)    :: glob_pms_CHV
148     double precision, allocatable, dimension(:,:)    :: glob_pms_Td
149     double precision, allocatable, dimension(:,:)    :: glob_pms_Smax
150     double precision, allocatable, dimension(:,:,:)    :: pm_map_all
151     double precision, allocatable, dimension(:,:)    :: up_pms
152
153         !start
154     auto_start_file = ''
155     i = 0
156     new_init = .TRUE.
157     print_output = .FALSE.
158     new_kine = .TRUE.
159
160     print *, '--- Starting DECIPHeR rainfall-runoff modelling ---'
161
162         ! Read in command-line arguments
163     do
164         CALL get_command_argument(i, arg)
165         if(len_trim(arg) == 0) exit
166         if (are_equal(arg, '-auto')) then
167             CALL get_command_argument(i+1, auto_start_file)
168         else if (are_equal(arg, '-new_init')) then
169             new_init = .TRUE.
170         else if (are_equal(arg, '-print_output')) then
171             print_output = .TRUE.
172         else if (are_equal(arg, '-new_kine')) then
173             new_kine = .TRUE.
174         endif
175         i = i + 1
176     end do
177
178     !  Call project to read in project files
179     call project (auto_start_file, &
180         out_dir_path, &
181         fname_control, &
182         fname_filemgr)
183
184     ! Call tread_dyna to read in the HRU file
185         write(999,*) ''
186     write(999,*) 'Reading in HRU files'
```

```
187     call Tread_dyna (nac, &
188         num_rivers, &
189         dyna_hru, &
190         dyna_riv, &
191         rivers, &
192         sum_ac_riv)
193 
194     ! Call inputs to read in the input data
195         write(999,*) ''
196     write(999,*) 'Reading in Input files'
197     call Inputs (nstep, &
198         num_rivers, &
199         dt, &
200         pe_step, &
201         qobs_riv_step_start, &
202         r_gau_step)
203 
204         ! Read in model structure info
205     write(999,*) ''
206     write(999,*) 'Reading in Model Structure files'
207     call modelstruct_setup (dyna_hru, &
208         nac)
209 
210     ! call read_routingdata to read in the routing data
211         write(999,*) ''
212     write(999,*) 'Reading in Routing Data'
213     call read_routingdata (cat_route_vmode, &
214         dyna_hru, &
215         nac, &
216         node_to_flow_mapping, &
217         num_rivers, &
218         route_riv, &
219         route_tdh, &
220         routing_mode)
221 
222     !call mpr_read_controls to read key variables from the MPR control file
223         write(999,*) ''
224     write(999,*) 'Reading in MPR files'
225     call mpr_read_controls(MPRfolder, &    !key filepaths and folder locations
226         fpath_HRUmap, &
227         folder_input, &
228         folder_output, &
229         fname_control, &
230         fname_filemgr, &
231         fnames_baspred, &
232         Params_range, &      !parameter min, default and max values
233         n_pm_maps, &         !variables in the control file
234         start_seed, &
235         save_pm_maps, &
236         save_bp_maps, &
237         pedo_tf_all, &
238         param_repeatruns,&
239         parammap_pert_ranges, &
240         fpath_gp_file)
241 
```

```
242     !call mpr_read_hru to read in the .asc file defining HRU map
243         !(or zones requiring different parameters)
244     call mpr_read_hru(fpath_HRUmap, &
245         HRU_map, &
246         nHRUs, &
247         stats_HRU)
248 
249     !call mpr_read_BasPred to read all required basin predictors,
250         !and clip to HRU map size
251     call mpr_read_BasPred(folder_input, &
252         MPRfolder, &
253         fnames_baspred, &
254         req_bp, &
255         pedo_tf_all, &
256         bp_maps, &
257         sand_0_10, &
258         silt_0_10, &
259         clay_0_10, &
260         orgm_0_10, &
261         soilmusiddata, &
262         stats_hru, &
263         hru_map, &
264         save_bp_maps, &
265         bp_root_depths)
266 
267     !call mpr_init_gparams to generate n global parameter sets
268         write(999,*) ''
269     write(999,*) 'Generating MPR global parameters'
270     call mpr_init_gparams(n_pm_maps, &
271         start_seed, &
272         pedo_tf_all, &
273         Params_range, &
274         n_gp_all, &
275         glob_pms_SZM, &
276         glob_pms_LnTo, &
277         glob_pms_SRmax, &
278         glob_pms_SRinit, &
279         glob_pms_CHV, &
280         glob_pms_Td, &
281         glob_pms_Smax, &
282         bp_root_depths)
283 
284     !write global parameters in output folder
285     folder_outfull = (trim(MPRfolder) // trim(folder_output))
286     call mpr_write_gparams(n_pm_maps, &
287         pedo_tf_all, &
288         folder_output, &
289         n_gp_all, &
290         glob_pms_SZM, &
291         glob_pms_LnTo, &
292         glob_pms_SRmax, &
293         glob_pms_SRinit, &
294         glob_pms_CHV, &
295         glob_pms_Td, &
296         glob_pms_Smax, &
```

```
297         param_repeatruns, &
298         parammap_add_mult, &
299         fpath_gp_file)
300
301     ! Loop through for each simulation
302     write(999,*) ''
303     write(999,*) 'Looping through simulations...'
304     do i_mc = 1, n_pm_maps  !previously numsim
305
306         print *,
307         print *, '*****************************************************************************'
308         print *, 'Running sim ', i_mc, ' out of ',n_pm_maps
309         print *, '*****************************************************************************'
310
311         !Apply pedotransfer functions
312                 !to get parameter values at 50m resolution
313         call mpr_pedotransfer(i_mc, &
314             n_pm_maps, &
315             pedo_tf_all, &
316             save_pm_maps, &
317             start_seed, &
318             folder_output, &
319             glob_pms_SZM, &
320             glob_pms_LnTo, &
321             glob_pms_SRmax, &
322             glob_pms_SRinit, &
323             glob_pms_CHV, &
324             glob_pms_Td, &
325             glob_pms_Smax,&
326             soilmusiddata, &
327             bp_maps , &
328             pm_map_all, &
329             stats_HRU, &
330             HRU_map)
331
332         !upscale parameters
333                 !to get parameter sets per parameter map (HRU map)
334         call mpr_upscale(HRU_map, &
335             nHRUs, &
336             pm_map_all, &
337             up_pms, &
338             pedo_tf_all, &
339             save_pm_maps, &
340             stats_HRU, &
341             i_mc, &
342             start_seed, &
343             folder_output)
344
345
346         !write upscaled parameters into DECIPHeR expected format - mcpar
347         allocate(mcpar(nHRUs,7))
348         mcpar(1:nHRUs,1:7) = up_pms(1:nHRUs,2:8)
349         NTT=1
350         WT=0.8              !kinematic solution parameters
351         ACC=0.000000001
```

```
352
353         !! ==================== MODEL RUNS HERE ========================
354
355         ! Open up output files
356         call file_open (i_mc+start_seed-1, &
357             print_output, &
358             out_dir_path)
359
360                 ! Run model
361         call mainloop (all_pm_names, &
362             nac, &
363             nstep, &
364             num_rivers, &
365             acc, &
366             dt, &
367             dtt, &
368             dyna_hru, &
369             mcpar, &
370             new_init, &
371             new_kine, &
372             ntt, &
373             node_to_flow_mapping, &
374             nHRUs, & !ACTUALLY NUMBER OF PARAMETER GRIDS
375             pe_step, &
376             print_output, &
377             qobs_riv_step_start, &
378             r_gau_step, &
379             rivers, &
380             route_riv, &
381             route_tdh, &
382             sum_ac_riv, &
383             wt)
384
385         close (40)
386         close (41)
387
388     end do
389
390     close(999)
391     !
392     !============================================================
393     !  END MAIN MC LOOP
394     !============================================================
395     !
396     stop
397
398 end program main
```

C.3 mpr_main.f90

```
!! Module containing main subroutine for MPR
!! Rosie Lane - 3rd November 2017

module mpr_main
contains

    ! ***************************************************************************************************
    ! subroutine: 1.mpr_read_controls: reads control file and param file
    ! ***************************************************************************************************
    ! This function checks that all required MPR files are in correct paths, &
    ! prints error messages otherwise.
    ! Then it reads the parameter and control file - reading each variable in from the
    ! control file seperately so that the order they are entered is irrelevant.
    ! Returns values of user_inputted requirements.
    ! 01/06/2018 - also reads in filepath from MPR_filenames.dat
    subroutine mpr_read_controls(MPRfolder, &    !key filepaths and folder locations
        fpath_HRUmap, &
        folder_input, &
        folder_output, &
        fname_control, &
        fname_filemgr, &
        fnames_baspred, &
        Params_range, &      !parameter min, default and max values from parameter file
        n_pm_maps, &         !variables in the control file
        start_seed, &
        save_pm_maps, &
        save_bp_maps, &
        pedo_tf_all, &       !list of pedo-transfer functions selected for each parameter
        param_repeatruns, &
        parammap_pert_ranges, &
        fpath_gp_file)

        use mpr_control
        use mpr_errors

        implicit none

        !DECLARE DUMMY VARIABLES
        !key filepaths and folder locations
        character(len=1024),intent(out)       :: MPRfolder
        character(len=1024),intent(out)       :: fpath_HRUmap
        character(len=1024),intent(out)       :: folder_input
        character(len=1024),intent(out)       :: folder_output
        character(len=1024),intent(in )       :: fname_control
        character(len=1024),intent(in )       :: fname_filemgr
        character(len=1024),dimension(18),intent(out)    :: fnames_baspred

        !parameter min, default and max values from parameter file
        double precision, dimension(7,3),intent(out)  :: Params_range

        !variables in the control file
        integer,intent(out)                 ::  n_pm_maps
```

```
53          integer,intent(out)                 ::  start_seed
54          integer,intent(out)                 ::  save_pm_maps
55          integer,intent(out)                 ::  save_bp_maps
56
57          !variables combining results from the control file
58          !pedo_tf_all: pedo-transfer eq. nums for SZM, LnTo, SRmax, SRinit, CHV, Td, Smax
59          integer, dimension(7),intent(out)   ::  pedo_tf_all
60
61          !DECLARE LOCAL VARIABLES
62          character(len=1024)     ::  fpath_PmSettings
63          character(len=1024)     ::  fpath_gp_file
64          character(len=1024)     ::  temp_name
65          character(len=13)       ::  comment
66          character(len=1024)     ::  folder_settings
67          character(len=1024)     ::  def_msg = ""
68          integer                 ::  pedo_tf_SZM
69          integer                 ::  pedo_tf_LnTo
70          integer                 ::  pedo_tf_SRmax   !pedo-transfer equation numbers for all params
71          integer                 ::  pedo_tf_SRinit
72          integer                 ::  pedo_tf_CHV
73          integer                 ::  pedo_tf_Td
74          integer                 ::  pedo_tf_Smax
75          integer                 ::  i
76
77          print *,
78          print *, '*************************************************************************'
79          print *, '0. Get filepaths from filemanager'
80          print *, '*************************************************************************'
81
82          print *,
83          print *, '!Read filenames from the filemanager control file:'
84          CALL read_control_file_char(fname_filemgr, "dir_root....", MPRfolder,def_msg)
85          CALL read_control_file_char(fname_filemgr, "dir_input...", folder_input,def_msg)
86          CALL read_control_file_char(fname_filemgr, "dir_output..", folder_output,def_msg)
87          CALL read_control_file_char(fname_filemgr, "dir_settings", folder_settings,def_msg)
88          CALL read_control_file_char(fname_filemgr, "name_hru_map", temp_name,def_msg)
89          fpath_HRUmap = trim(folder_input) // trim(temp_name)
90          CALL read_control_file_char(fname_filemgr, "name_pm_stng", temp_name,def_msg)
91          fpath_PmSettings = trim(folder_settings) // trim(temp_name)
92          CALL read_control_file_char(fname_filemgr, "name_pm_pert", temp_name,def_msg)
93          fpath_PmPertSettings = trim(folder_settings) // trim(temp_name)
94          CALL read_control_file_char(fname_filemgr, "name_gp_file", temp_name,def_msg)
95          fpath_gp_file = trim(folder_output) // trim(temp_name)
96
97          print *,
98          print *, '!Read basin predictor filenames from filemanager control file:'
99          CALL read_control_file_char(fname_filemgr, "lnto_sand_d1", fnames_baspred(1),def_msg)
100         CALL read_control_file_char(fname_filemgr, "lnto_silt_d1", fnames_baspred(2),def_msg)
101         CALL read_control_file_char(fname_filemgr, "lnto_clay_d1", fnames_baspred(3),def_msg)
102         CALL read_control_file_char(fname_filemgr, "lnto_orgm_d1", fnames_baspred(4),def_msg)
103         CALL read_control_file_char(fname_filemgr, "szm_musids..", fnames_baspred(5),def_msg)
104         CALL read_control_file_char(fname_filemgr, "szm_mid_nde.", fnames_baspred(6),def_msg)
105         CALL read_control_file_char(fname_filemgr, "szm_table_d1", fnames_baspred(7),def_msg)
106         CALL read_control_file_char(fname_filemgr, "szm_table_d2", fnames_baspred(8),def_msg)
107         CALL read_control_file_char(fname_filemgr, "szm_table_d3", fnames_baspred(9),def_msg)
```

```
108         CALL read_control_file_char(fname_filemgr, "szm_table_d4", fnames_baspred(10),def_msg)
109         CALL read_control_file_char(fname_filemgr, "szm_table_d5", fnames_baspred(11),def_msg)
110         CALL read_control_file_char(fname_filemgr, "srmax_lcm...", fnames_baspred(12),def_msg)
111         CALL read_control_file_char(fname_filemgr, "srmax_table.", fnames_baspred(13),def_msg)
112         CALL read_control_file_char(fname_filemgr, "smax_d2b_map", fnames_baspred(14),def_msg)
113         CALL read_control_file_char(fname_filemgr, "lnto_bulk_d1", fnames_baspred(15),def_msg)
114         CALL read_control_file_char(fname_filemgr, "lnto_isorg..", fnames_baspred(16),def_msg)
115         CALL read_control_file_char(fname_filemgr, "sr_smax_poro", fnames_baspred(17),def_msg)
116         CALL read_control_file_char(fname_filemgr, "hydrogeology", fnames_baspred(18),def_msg)
117
118         !check all files exist and print useful info to screen
119         !print *,
120         !print *, 'Double check these locations: '
121         !print *, 'Param map : ',trim(fpath_HRUmap)
122         !print *, 'Param settings : ',trim(fpath_PmSettings)
123         !print *, 'fname_control = ',trim(fname_control)
124         !print *, 'Param map settings = ',trim(fpath_PmPertSettings)
125         CALL check_files_exist(MPRfolder, fpath_HRUmap, folder_input, folder_output, &
126         fname_control,fpath_PmSettings)
127
128         print *,
129         print *, '*************************************************************************'
130         print *, '1. read parameter and control file'
131         print *, '*************************************************************************'
132
133         !parameter file
134         CALL read_param_file(fpath_PmSettings,Params_range(1,:),Params_range(2,:),&
135             Params_range(3,:), Params_range(4,:),Params_range(5,:),Params_range(6,:),&
136             Params_range(7,:))
137
138         !mpr control file.
139         CALL read_control_file_int(fname_control, "n_pm_maps", n_pm_maps,1)
140         CALL read_control_file_int(fname_control, "start_seed", start_seed,1)
141         CALL read_control_file_int(fname_control, "save_pm_maps", save_pm_maps,0)
142         CALL read_control_file_int(fname_control, "save_bp_maps", save_bp_maps,0)
143         CALL read_control_file_int(fname_control, "pedo_tf_SZM", pedo_tf_SZM,0)
144         CALL read_control_file_int(fname_control, "pedo_tf_LnTo", pedo_tf_LnTo,0)
145         CALL read_control_file_int(fname_control, "pedo_tf_SRmax", pedo_tf_SRmax,0)
146         CALL read_control_file_int(fname_control, "pedo_tf_SRinit", pedo_tf_SRinit,0)
147         CALL read_control_file_int(fname_control, "pedo_tf_CHV", pedo_tf_CHV,0)
148         CALL read_control_file_int(fname_control, "pedo_tf_Td", pedo_tf_Td,0)
149         CALL read_control_file_int(fname_control, "pedo_tf_Smax", pedo_tf_Smax,0)
150         CALL read_control_file_int(fname_control, "n_runs_SZM",param_repeatruns(1),0)
151         CALL read_control_file_int(fname_control, "n_runs_LnTo",param_repeatruns(2),0)
152         CALL read_control_file_int(fname_control, "n_runs_SRmax",param_repeatruns(3),0)
153         CALL read_control_file_int(fname_control, "n_runs_SRinit",param_repeatruns(4),0)
154         CALL read_control_file_int(fname_control, "n_runs_CHV",param_repeatruns(5),0)
155         CALL read_control_file_int(fname_control, "n_runs_Td",param_repeatruns(6),0)
156         CALL read_control_file_int(fname_control, "n_runs_Smax",param_repeatruns(7),0)
157
158         pedo_tf_all(1:4) = (/pedo_tf_SZM, pedo_tf_LnTo, pedo_tf_SRmax, pedo_tf_SRinit /)
159         pedo_tf_all(5:7) = (/pedo_tf_CHV, pedo_tf_Td, pedo_tf_Smax /)
160
161
162     525     format(a13,1X,a700)
```

```
163
164     end subroutine mpr_read_controls
165
166
167     ! ******************************************************************************************
168     ! subroutine: 2. mpr_read_hru: Reads the HRU/ spatial parameter map
169     ! ******************************************************************************************
170     ! This function reads the HRU ascii grid and header, from file specified in fpath_HRUmap
171     ! It saves key information from the header into stats_HRU
172     ! It makes sure all nodata values are set to -9999 for consistency.
173     subroutine mpr_read_hru(fpath_HRUmap, &
174         HRU_map, &
175         nHRUs, &
176         stats_HRU)
177
178         use mpr_utility
179
180         implicit none
181
182         !DECLARE DUMMY VARIABLES
183         !filepath pointing to HRU .asc file created in DTA
184         character(len=1024), intent(in) :: fpath_HRUmap
185
186         !variables defining HRU map and parameter maps
187         double precision, allocatable, dimension(:,:),intent(out) :: HRU_map
188         integer,intent(out)                                       :: nHRUs      !number of HRUs
189         double precision, dimension(4),intent(out)                :: stats_HRU
190
191         !DECLARE LOCAL VARIABLES
192          !all variables relating to size and location:
193          !of HRU map created during DTA
194          !of param maps - assumed same as HRU map
195         integer                         :: ncols_HRU
196         integer                         :: nrows_HRU
197         double precision                :: xllcorner_HRU
198         double precision                :: yllcorner_HRU
199         double precision                :: cellsize_HRU
200         double precision                :: nodata_HRU
201         integer                         :: ncol
202         integer                         :: nrow
203         double precision                :: xll
204         double precision                :: yll
205         double precision                :: cellsize
206
207         print *,
208         print *, '************************************************************************'
209         print *, '2. Read HRU map'
210         print *, '************************************************************************'
211
212         print *, 'Reading map from: ',trim(fpath_HRUmap)
213
214         call read_ascii_grid(fpath_HRUmap, HRU_map, ncols_HRU, nrows_HRU, xllcorner_HRU, &
215             yllcorner_HRU, cellsize_HRU, nodata_HRU)
216
217         nHRUs = maxval(int(HRU_map))
```

```
218         stats_HRU(1:4) = (/ xllcorner_HRU, yllcorner_HRU, cellsize_HRU, nodata_HRU/)
219
220         !stats values of hru map - all other maps will be converted to these
221         nrow = nrows_HRU
222         ncol = ncols_HRU
223         xll = xllcorner_HRU
224         yll = yllcorner_HRU
225         cellsize = cellsize_HRU
226
227         print *, 'Number of HRUs found: ',nHRUs
228         print *, 'Rows in HRU file: ',nrows_HRU
229         print *, 'Cols in HRU file: ',ncols_HRU
230
231         !convert nodata value to -9999 to avoid confusion
232         IF (stats_HRU(4) /= -9999) THEN
233             stats_HRU(4) = -9999
234             print *,'old nodata value: ',int(nodata_HRU),' being converted to ',int(stats_HRU(4))
235             call set_nodata_value( HRU_map, nodata_HRU, stats_HRU(4))
236             nodata_HRU = -9999
237         ELSE
238             print *, 'nodata value: ', nodata_HRU
239         END IF
240
241     end subroutine mpr_read_hru
242
243
244     ! ******************************************************************************************
245     ! subroutine: 3. mpr_read_BasPred: Reads required Basin Predictor maps and trims to HRU
246     ! ******************************************************************************************
247     ! a. checks which basin predictors are needed for selected pedo-transfer functions
248     ! b. reads in only those maps
249     !   - maps are clipped to same extent as HRU map when being read-in
250     !   - maps can be saved if specified in MPR_control file
251     subroutine mpr_read_BasPred(folder_input, &
252         MPRfolder, &
253         fnames_baspred, &
254         req_bp, &
255         pedo_tf_all, &
256         bp_maps, &
257         sand_0_10, &
258         silt_0_10, &
259         clay_0_10, &
260         orgm_0_10, &
261         soilmusiddata, &
262         stats_hru, &
263         hru_map, &
264         save_bp_maps, &
265         bp_root_depths)
266
267         use mpr_utility
268         use mpr_extract_BasPred
269         use mpr_SZM
270         use mpr_LnTo
271         use mpr_SRmax
272         use mpr_SRinit
```

```
273         use mpr_CHV
274         use mpr_Td
275         use mpr_Smax
276
277         implicit none
278
279         !DECLARE DUMMY VARIABLES
280         !key filepaths
281         character(len=1024), intent(in)                 :: MPRfolder
282         character(len=1024), intent(in)                 :: folder_input
283         character(len=1024), dimension(18),intent(in)   :: fnames_baspred
284
285         !specific requirements based on user input to control file
286         !req_bp: basin predictor maps needed - logical arrays with true values if maps required
287         !pedo_tf_all: pedo-transfer eq. nums for SZM, LnTo, SRmax, SRinit, CHV, Td, Smax
288         integer, intent(in)                 ::  save_bp_maps
289         logical, dimension(18), intent(out)::  req_bp
290         integer, dimension(7),intent(inout) ::  pedo_tf_all
291
292         !basin predictor maps
293         !bp_maps is an array of all basin predictor maps
294         real, allocatable, dimension(:,:,:),intent(out)  :: bp_maps
295         real, allocatable, dimension(:,:),intent(out)    :: sand_0_10
296         real, allocatable, dimension(:,:),intent(out)    :: silt_0_10
297         real, allocatable, dimension(:,:),intent(out)    :: clay_0_10
298         real, allocatable, dimension(:,:),intent(out)    :: orgm_0_10
299         double precision, allocatable, dimension(:,:,:), intent(out) :: soilmusiddata
300         real, allocatable, dimension(:,:), intent(out)   :: bp_root_depths
301
302         !HRU map and properties
303         double precision, allocatable, dimension(:,:),intent(in) :: HRU_map
304         double precision, dimension(4), intent(in)               :: stats_HRU
305
306         !DECLARE LOCAL VARIABLES
307         integer :: i
308
309         print *,
310         print *, '*************************************************************************'
311         print *, '3. Read required basin predictor maps'
312         print *, '*************************************************************************'
313         print *, 'Reading maps from: ',trim(folder_input)
314         print *,
315
316         DO i = 1,14
317             req_bp(i) = .false.
318         END DO
319
320         ! Check which basin predictors are required
321         ! get_BasPred_req returns a logical list of the required soils maps req_soils(type,depth),
322         ! which contains .true. where soils maps are required
323         CALL get_BasPred_SZM(pedo_tf_all(1), req_bp)
324         CALL get_BasPred_LnTo(pedo_tf_all(2), req_bp)
325         CALL get_BasPred_SRmax(pedo_tf_all(3), req_bp)
326         CALL get_BasPred_SRinit(pedo_tf_all(4), req_bp)
327         CALL get_BasPred_CHV(pedo_tf_all(5), req_bp)
```

```
328         CALL get_BasPred_Td(pedo_tf_all(6), req_bp)
329         CALL get_BasPred_Smax(pedo_tf_all(7), req_bp)
330
331         ! read_req_soils reads in any basin predictor maps required
332         ! for user-selected pedo transfer functions
333         CALL read_req_soils(folder_input,&
334             MPRfolder,&
335             fnames_baspred,&
336             req_bp,&
337             stats_HRU,&
338             HRU_map,&
339             save_bp_maps,&
340             bp_maps, &
341             soilmusiddata, &
342             bp_root_depths)
343
344         sand_0_10 = bp_maps(:,:,1)
345         silt_0_10 = bp_maps(:,:,2)
346         clay_0_10 = bp_maps(:,:,3)
347         orgm_0_10 = bp_maps(:,:,4)
348
349     end subroutine mpr_read_BasPred
350
351
352     ! ***************************************************************************************
353     ! subroutine: 4. mpr_init_gparams: initialises global parameters using random num generator
354     ! ***************************************************************************************
355     ! a. checks how many global params are needed, given pedo-transfer functions selected
356     ! b. initialises global parameters within given bounds
357     ! c. generates n sets of parameters, starting at given start seed
358     subroutine mpr_init_gparams(n_pm_maps, &
359         start_seed, &
360         pedo_tf_all, &
361         Params_range, &
362         n_gp_all, &
363         glob_pms_SZM, &
364         glob_pms_LnTo, &
365         glob_pms_SRmax, &
366         glob_pms_SRinit, &
367         glob_pms_CHV, &
368         glob_pms_Td, &
369         glob_pms_Smax, &
370         rd)
371
372         use mpr_utility
373         use mpr_extract_BasPred
374         use mpr_SZM
375         use mpr_LnTo
376         use mpr_SRmax
377         use mpr_SRinit
378         use mpr_CHV
379         use mpr_Td
380         use mpr_Smax
381
382         implicit none
```

```
383
384         !DECLARE DUMMY VARIABLES
385         !variables from the control file
386         !n_pm_maps: number of parameter files to create
387         !start_seed: starting seed for generation of global params
388         !pedo_tf_all: pedo-transfer eq. nums for SZM, LnTo, SRmax, SRinit, CHV, Td, Smax
389         integer,intent(in)                  ::  n_pm_maps
390         integer,intent(in)                  ::  start_seed
391         integer, dimension(7),intent(in)    ::  pedo_tf_all
392
393         !parameter min, default and max values from parameter file
394         double precision, dimension(7,3),intent(in)  :: Params_range
395
396         !global parameter information
397         integer, dimension(7),intent(out)                           :: n_gp_all
398         double precision, allocatable, dimension(:,:),intent(out)   :: glob_pms_SZM
399         double precision, allocatable, dimension(:,:),intent(out)   :: glob_pms_LnTo
400         double precision, allocatable, dimension(:,:),intent(out)   :: glob_pms_SRmax
401         double precision, allocatable, dimension(:,:),intent(out)   :: glob_pms_SRinit
402         double precision, allocatable, dimension(:,:),intent(out)   :: glob_pms_CHV
403         double precision, allocatable, dimension(:,:),intent(out)   :: glob_pms_Td
404         double precision, allocatable, dimension(:,:),intent(out)   :: glob_pms_Smax
405
406         !special parameter information files
407         real, allocatable, dimension(:,:),intent(in)    :: rd !rooting depth table bp_root_depths
408
409         print *,
410         print *, '*****************************************************************************'
411         print *, '4. Initialise global parameters within given bounds'
412         print *, '*****************************************************************************'
413
414         !initialise vector storing the number of global parameters needed for each parameter.
415         n_gp_all = (/ 0,0,0,0,0,0,0 /)
416
417         if (pedo_tf_all(1)==4) THEN !SZM can share global param values with LnT0
418             CALL init_gp_SZM(2,glob_pms_SZM, n_pm_maps, start_seed, n_gp_all, Params_range(1,:))
419         ELSE
420             CALL init_gp_SZM(pedo_tf_all(1),glob_pms_SZM, n_pm_maps, start_seed, n_gp_all, &
421                 Params_range(1,:))
422         END IF
423         CALL init_gp_LnTo(pedo_tf_all(2),glob_pms_LnTo, n_pm_maps, start_seed, n_gp_all, &
424             Params_range(2,:))
425         CALL init_gp_SRmax(pedo_tf_all(3),glob_pms_SRmax, n_pm_maps, start_seed, n_gp_all, &
426             Params_range(3,:),rd)
427         CALL init_gp_SRinit(pedo_tf_all(4),glob_pms_SRinit, n_pm_maps, start_seed, n_gp_all, &
428             Params_range(4,:))
429         CALL init_gp_CHV(pedo_tf_all(5),glob_pms_CHV, n_pm_maps, start_seed, n_gp_all, &
430             Params_range(5,:))
431         CALL init_gp_Td(pedo_tf_all(6),glob_pms_Td, n_pm_maps, start_seed, n_gp_all, &
432             Params_range(6,:))
433         CALL init_gp_Smax(pedo_tf_all(7),glob_pms_Smax, n_pm_maps, start_seed, n_gp_all, &
434             Params_range(7,:))
435
436
437     end subroutine mpr_init_gparams
```

```
438
439
440     ! ************************************************************************************************
441     ! subroutine: 4b. mpr_write_gparams: writes initialised global parameters in output file
442     ! ************************************************************************************************
443     subroutine mpr_write_gparams(n_pm_maps, &
444         pedo_tf_all, &
445         out_dir_path, &
446         n_gp_all, &
447         glob_pms_SZM, &
448         glob_pms_LnTo, &
449         glob_pms_SRmax, &
450         glob_pms_SRinit, &
451         glob_pms_CHV, &
452         glob_pms_Td, &
453         glob_pms_Smax, &
454         param_repeatruns, &
455         parammap_add_mult, &
456         fpath_gp_file)
457
458         use mpr_utility
459
460         implicit none
461
462         !DECLARE DUMMY VARIABLES
463         !variables from the control file
464         integer,intent(in)                  ::  n_pm_maps
465         integer, dimension(7),intent(in)    ::  pedo_tf_all
466         character(900), intent(in)          ::  out_dir_path
467         character(len=1024),intent(in)      :: fpath_gp_file
468
469         !global parameters
470         integer, dimension(7),intent(in)                          :: n_gp_all
471         double precision, allocatable, dimension(:,:),intent(in)  :: glob_pms_SZM
472         double precision, allocatable, dimension(:,:),intent(in)  :: glob_pms_LnTo
473         double precision, allocatable, dimension(:,:),intent(in)  :: glob_pms_SRmax
474         double precision, allocatable, dimension(:,:),intent(in)  :: glob_pms_SRinit
475         double precision, allocatable, dimension(:,:),intent(in)  :: glob_pms_CHV
476         double precision, allocatable, dimension(:,:),intent(in)  :: glob_pms_Td
477         double precision, allocatable, dimension(:,:),intent(in)  :: glob_pms_Smax
478
479         !local parameters
480         character(900) :: filename
481         character(1000000)  :: numformat
482         double precision, allocatable, dimension(:,:,:) :: glob_pms_all
483         integer :: i, j
484         integer :: max_ngp
485         character(1) :: c
486
487         max_ngp = maxval(n_gp_all(:))
488
489         !put all global parameters into one large array
490         allocate(glob_pms_all(max_ngp,n_pm_maps,7))
491
492         glob_pms_all(1:n_gp_all(1),:,1) = glob_pms_SZM
```

```
493         glob_pms_all(1:n_gp_all(2),:,2) = glob_pms_LnTo
494         glob_pms_all(1:n_gp_all(3),:,3) = glob_pms_SRmax
495         glob_pms_all(1:n_gp_all(4),:,4) = glob_pms_SRinit
496         glob_pms_all(1:n_gp_all(5),:,5) = glob_pms_CHV
497         glob_pms_all(1:n_gp_all(6),:,6) = glob_pms_Td
498         glob_pms_all(1:n_gp_all(7),:,7) = glob_pms_Smax
499
500         ! Then open the output file
501         open(unit=24,file=fpath_gp_file,status='unknown')
502
503         !write title followed by formatted output for each parameter
504         write(24,*) 'Param (szm,lnto,srmax,srinit,chv,td,smax), GP_num, values'
505         !numformat = "(I5,I5,F15.10 " // repeat("F15.10 ",n_pm_maps) // ")"
506         !numformat = "(I5,I5," // repeat("F15.10 ",n_pm_maps) // ")"
507         numformat = "(I2,',',I2" // repeat("',',F15.5",n_pm_maps) // ")"
508         do i = 1,7
509             do j = 1,n_gp_all(i)
510                 write(24,numformat) i,j,glob_pms_all(j,:,i)
511              end do
512         end do
513         close(unit=24)
514
515         !numformat
516
517     end subroutine mpr_write_gparams
518
519
520     ! ************************************************************************************************
521     ! subroutine: 5. mpr_pedotransfer: applies tf functions to create parameter maps
522     ! ************************************************************************************************
523     ! For run n in N parameter maps
524     ! a. Apply pedotransfer functions to map n to get parameter maps
525     ! b. Save these parameter maps if save_pm_maps specified in the mpr control file
526     ! c. output parameter maps
527      subroutine mpr_pedotransfer(i_n, &
528         n_pm_maps, &
529         pedo_tf_all, &
530         save_pm_maps, &
531         start_seed, &
532         folder_output, &
533         glob_pms_SZM, &
534         glob_pms_LnTo, &
535         glob_pms_SRmax, &
536         glob_pms_SRinit, &
537         glob_pms_CHV, &
538         glob_pms_Td, &
539         glob_pms_Smax,&
540         soilmusiddata, &
541         bp_maps , &
542         pm_map_all, &
543         stats_HRU, &
544         HRU_map)
545
546         use mpr_utility
547         use mpr_SZM
```

```
548         use mpr_LnTo
549         use mpr_SRmax
550         use mpr_SRinit
551         use mpr_CHV
552         use mpr_Td
553         use mpr_Smax
554
555         implicit none
556
557         !!DECLARE DUMMY VARIABLES
558         integer                 :: i_n
559         integer                 :: n_pm_maps
560         integer, dimension(7)   :: pedo_tf_all
561         integer                 :: save_pm_maps
562         integer                 :: start_seed
563         character(len=1024)     :: folder_output
564
565         !global parameters
566         !dim(glob pm number, 1:n_pm_maps)
567         double precision, allocatable, dimension(:,:)    :: glob_pms_SZM
568         double precision, allocatable, dimension(:,:)    :: glob_pms_LnTo
569         double precision, allocatable, dimension(:,:)    :: glob_pms_SRmax
570         double precision, allocatable, dimension(:,:)    :: glob_pms_SRinit
571         double precision, allocatable, dimension(:,:)    :: glob_pms_CHV
572         double precision, allocatable, dimension(:,:)    :: glob_pms_Td
573         double precision, allocatable, dimension(:,:)    :: glob_pms_Smax
574
575         !basin predictor maps
576         !bp maps (x,y,sand/silt/clay/orgm/ksat decline,depth_class)
577         real, allocatable, dimension(:,:,:)     :: bp_maps
578         double precision, allocatable, dimension(:,:,:), intent(in) :: soilmusiddata
579
580         !parameter maps at the finest scale, created using transfer functions
581         !pm_map_all(x,y,[SZM, LnTo, SRmax, SRinit, CHV, Td, Smax])
582         double precision, allocatable, dimension(:,:,:)    :: pm_map_all
583
584         !variables defining HRU map and parameter maps
585         double precision, allocatable, dimension(:,:) :: HRU_map
586         double precision, dimension(4)  :: stats_HRU !
587
588         !DECLARE LOCAL VARIABLES
589         !stats of HRU map
590         integer                         :: ncols_HRU
591         integer                         :: nrows_HRU
592
593         !parameter maps at the finest scale, created using transfer functions
594         double precision, allocatable, dimension(:,:)    :: pm_map_SZM
595         double precision, allocatable, dimension(:,:)    :: pm_map_LnTo
596         double precision, allocatable, dimension(:,:)    :: pm_map_SRmax
597         double precision, allocatable, dimension(:,:)    :: pm_map_SRinit
598         double precision, allocatable, dimension(:,:)    :: pm_map_CHV
599         double precision, allocatable, dimension(:,:)    :: pm_map_Td
600         double precision, allocatable, dimension(:,:)    :: pm_map_Smax
601         double precision, allocatable, dimension(:,:,:) :: pm_map_all_temp
602
```

```
603         !basin predictors
604         real, allocatable, dimension(:,:)    :: sand_0_10
605         real, allocatable, dimension(:,:)    :: silt_0_10
606         real, allocatable, dimension(:,:)    :: clay_0_10
607         real, allocatable, dimension(:,:)    :: orgm_0_10
608         double precision, allocatable, dimension(:,:)    :: ksat_decline
609         real, allocatable, dimension(:,:)    :: srmax_lcm
610         real, allocatable, dimension(:,:)    :: smax_d2r
611
612         !variables used when writing out a file
613         character(len=1024)  :: temp_fn      !stores dynamic filename
614         character(len = 5)   :: i_n_str      !string storing current number of pm files made
615         character(len = 6)   :: fmt = '(I5.5)' !format descriptor
616         integer              :: x,y,i
617
618         print *,
619         print *,'          ****************************************************************'
620         print *, '              Apply pedo-transfer functions to create parameter map'
621         print *, '           ****************************************************************'
622
623         print *,
624         print *, 'n = ',i_n,' out of ',n_pm_maps
625
626         !fill in local variables
627         ncols_HRU = size(HRU_map(1,:))
628         nrows_HRU = size(HRU_map(:,1))
629         sand_0_10 = bp_maps(:,:,1)
630         silt_0_10 = bp_maps(:,:,2)
631         clay_0_10 = bp_maps(:,:,3)
632         orgm_0_10 = bp_maps(:,:,4)
633         srmax_lcm=bp_maps(:,:,12)
634         smax_d2r=bp_maps(:,:,14)
635
636         if (i_n == 1) then
637             allocate(pm_map_all(nrows_HRU, ncols_HRU, 7))
638             allocate(pm_map_all_temp(nrows_HRU, ncols_HRU, 7))
639         end if
640
641         !apply pedotransfer functions to map n to get parameter maps
642         CALL pedotf_LnTo(i_n,pedo_tf_all(2), glob_pms_LnTo, sand_0_10, clay_0_10, &
643         pm_map_LnTo,bp_maps)
644
645         IF (pedo_tf_all(1)==3) THEN !apply SZM with lnT0 parameters
646             CALL pedotf_SZM(i_n, 2, glob_pms_LnTo,bp_maps, pm_map_SZM, soilmusiddata,&
647             folder_output,pm_map_LnTo)
648         ELSE  !apply SZM ptf with SZM global parameters
649             CALL pedotf_SZM(i_n, pedo_tf_all(1), glob_pms_SZM,bp_maps, pm_map_SZM, &
650             soilmusiddata,folder_output,pm_map_LnTo)
651         END IF
652
653         CALL pedotf_SRmax(i_n, pedo_tf_all(3), glob_pms_SRmax,srmax_lcm,bp_maps(:,:,17),&
654         pm_map_SRmax)
655         CALL pedotf_SRinit(i_n, pedo_tf_all(4), glob_pms_SRinit,sand_0_10, pm_map_SRinit)
656         CALL pedotf_CHV(i_n, pedo_tf_all(5), glob_pms_CHV,sand_0_10,pm_map_CHV)
657         CALL pedotf_Td(i_n, pedo_tf_all(6), glob_pms_Td,sand_0_10, pm_map_Td,pm_map_LnTo)
```

```
658         CALL pedotf_Smax(i_n, pedo_tf_all(7), glob_pms_Smax,smax_d2r,bp_maps(:,:,17),&
659         bp_maps(:,:,18),pm_map_Smax)
660
661         pm_map_all(:,:,1) = pm_map_SZM
662         pm_map_all(:,:,2) = pm_map_LnTo
663         pm_map_all(:,:,3) = pm_map_SRmax
664         pm_map_all(:,:,4) = pm_map_SRinit
665         pm_map_all(:,:,5) = pm_map_CHV
666         pm_map_all(:,:,6) = pm_map_Td
667         pm_map_all(:,:,7) = pm_map_Smax
668         pm_map_all_temp=pm_map_all
669
670         !save parameter maps if required
671         IF (save_pm_maps >= 1) THEN
672         print *,
673         print *, 'Saving the following parameter maps:'
674             write(i_n_str,fmt) (i_n+start_seed-1)
675
676                         !write nodata values so that maps match HRU maps
677             DO x = 1,ncols_HRU
678                 DO y = 1,nrows_HRU
679                     IF (HRU_map(y,x) == stats_HRU(4)) THEN !if nodata in HRU map
680                         pm_map_all_temp(y,x,:) = stats_HRU(4)
681                     END IF
682                     DO i = 1,7
683                         IF (pm_map_all_temp(y,x,i)>=100000) THEN
684                             pm_map_all_temp(y,x,i) = 100000
685                         END IF
686                     END DO
687                 END DO
688             END DO
689
690             DO i = 1,7
691
692             !get filename for each different parameter
693             IF (i==1) THEN
694                 temp_fn = trim(folder_output) // "param_map_szm" // trim(i_n_str)//".asc"
695             ELSE IF(i==2) THEN
696                 temp_fn = trim(folder_output) // "param_map_lnto" // trim(i_n_str)//".asc"
697             ELSE IF(i==3) THEN
698                 temp_fn = trim(folder_output) // "param_map_srmax" // trim(i_n_str)//".asc"
699             ELSE IF(i==4) THEN
700                 temp_fn = trim(folder_output) // "param_map_srinit" // trim(i_n_str)//".asc"
701             ELSE IF(i==5) THEN
702                 temp_fn = trim(folder_output) // "param_map_chv" // trim(i_n_str)//".asc"
703             ELSE IF(i==6) THEN
704                 temp_fn = trim(folder_output) // "param_map_td" // trim(i_n_str)//".asc"
705             ELSE IF(i==7) THEN
706                 temp_fn = trim(folder_output) // "param_map_smax" // trim(i_n_str)//".asc"
707             END IF
708
709             !only save parameter map if this parameter has had MPR applied
710             IF (pedo_tf_all(i)>1) THEN
711                 call write_ascii_grid(temp_fn,pm_map_all_temp(:,:,i),ncols_HRU,nrows_HRU,&
712                     stats_HRU(1),stats_HRU(2),stats_HRU(3),stats_HRU(4),6)
```

```
713                 print *,'   Saving parameter map: ',trim(temp_fn)
714             END IF
715             END DO
716         END IF
717
718
719     end subroutine mpr_pedotransfer
720
721
722
723
724     ! ************************************************************************************************
725     ! subroutine: 6. mpr_upscale: Upscale param map to get param value for each HRU
726     ! ************************************************************************************************
727     ! For run n in N parameter maps
728     ! a. Loop through all HRUs (or zones with different parameter values)
729     ! b. Extract all parameter values from parameter map covering selected HRU
730     ! c. Upscale parameters by taking an average
731     subroutine mpr_upscale(HRU_map, &
732         nHRUs, &
733         pm_map_all, &
734         up_pms, &
735         pedo_tf_all, &
736         save_pm_maps, &
737         stats_HRU, &
738         i_n, &
739         start_seed, &
740         folder_output)
741
742         use mpr_utility
743         use mpr_SZM
744         use mpr_LnTo
745         use mpr_SRmax
746         use mpr_SRinit
747         use mpr_CHV
748         use mpr_Td
749         use mpr_Smax
750         use mpr_upscaling
751
752         implicit none
753
754         !!DECLARE DUMMY VARIABLES
755         !up_pms: vector of upscaled params (nHRUs,8)
756         integer, intent(in)                     :: i_n
757         double precision, allocatable, dimension(:,:),intent(in)     :: HRU_map
758         integer,intent(in)                                            :: nHRUS
759         double precision, allocatable, dimension(:,:,:),intent(in)   :: pm_map_all
760         double precision, allocatable, dimension(:,:),intent(out)    :: up_pms
761         integer, dimension(7) , intent(in)                            :: pedo_tf_all
762         integer                 :: save_pm_maps
763         double precision, dimension(4)  :: stats_HRU
764         integer                 :: start_seed
765         character(len=1024)     :: folder_output
766
767         !!DECLARE LOCAL VARIABLES
```

```
768         integer                 :: i_HRU
769
770         !variables required for parameter upscaling
771         !vector of all param values in a HRU
772         double precision, allocatable, dimension(:)      :: pms_SZM
773         double precision, allocatable, dimension(:)      :: pms_LnTo
774         double precision, allocatable, dimension(:)      :: pms_SRmax
775         double precision, allocatable, dimension(:)      :: pms_SRinit
776         double precision, allocatable, dimension(:)      :: pms_CHV
777         double precision, allocatable, dimension(:)      :: pms_Td
778         double precision, allocatable, dimension(:)      :: pms_Smax
779
780         !allocate plotting variables
781         double precision, allocatable, dimension(:,:,:) :: pm_map_all_temp
782         character(len=1024)                              :: temp_fn
783         character(len = 5)                               :: i_n_str
784         character(len = 6)                               :: fmt = '(I5.5)' !format descriptor
785         integer                                          :: x,y,i,nrows,ncols
786
787         !allocate variable to store upscaled parameters
788         allocate(up_pms(nHRUs,8))
789
790         print *,
791         print *, '          ****************************************************************'
792         print *, '             Upscale parameters '
793         print *, '          ****************************************************************'
794
795
796         DO i_HRU = 1,nHRUs !loop through for each HRU
797
798             IF (mod(i_HRU,250)==0) THEN
799                 print *, '      upscaling i_HRU = ',i_HRU,' out of ',nHRUs
800             END IF
801             !call cpu_time(start)
802
803             !!  ----- E) FIND AND EXTRACT PARAMETER VALUES FOR EACH HRU -------------------------
804             !pms_LnTo is a vector of all LnTo parameter values in the HRU number i_HRU
805              !!  ----- F) UPSCALE PARAMETERS BY TAKING AVERAGE FOR EACH HRU ----------------------
806             !up_pms will determine the formatting of the parameter file
807             up_pms(i_HRU,1) = i_HRU                 !keep log of HRU numbers
808
809             if (pedo_tf_all(1) > 1) then !skip this if set as fixed or global parameter
810                 CALL extract_pm_HRU(i_HRU,HRU_map, pm_map_all(:,:,1), pms_SZM)
811                 CALL upscale_harmonic(pms_SZM, up_pms(i_HRU,2))    !HRU upscaled parameter value
812             else
813                 up_pms(i_HRU,2) = pm_map_all(1,1,1)
814             end if
815
816             if (pedo_tf_all(2) > 1) then
817                 CALL extract_pm_HRU(i_HRU,HRU_map, pm_map_all(:,:,2), pms_LnTo)
818                 !CALL upscale_harmonic(pms_LnTo, up_pms(i_HRU,3))
819                 CALL upscale_arithmetic(pms_LnTo, up_pms(i_HRU,3))
820             else
821                 up_pms(i_HRU,3) = pm_map_all(1,1,2)
822             end if
```

```
823
824             if (pedo_tf_all(3) > 1) then
825                 CALL extract_pm_HRU(i_HRU,HRU_map, pm_map_all(:,:,3), pms_SRmax)
826                 CALL upscale_arithmetic(pms_SRmax, up_pms(i_HRU,4))
827             else
828                 up_pms(i_HRU,4) = pm_map_all(1,1,3)
829             end if
830
831             if (pedo_tf_all(4) > 1) then
832                 CALL extract_pm_HRU(i_HRU,HRU_map, pm_map_all(:,:,4), pms_SRinit)
833                 CALL upscale_arithmetic(pms_SRinit, up_pms(i_HRU,5))
834             else
835                 up_pms(i_HRU,5) = pm_map_all(1,1,4)
836             end if
837
838             if (pedo_tf_all(5) > 1) then
839                 CALL extract_pm_HRU(i_HRU,HRU_map, pm_map_all(:,:,5),  pms_CHV)
840                 CALL upscale_geometric(pms_CHV, up_pms(i_HRU,6))
841             else
842                 up_pms(i_HRU,6) = pm_map_all(1,1,5)
843             end if
844
845             if (pedo_tf_all(6) > 1) then
846                 CALL extract_pm_HRU(i_HRU,HRU_map, pm_map_all(:,:,6), pms_Td)
847                 CALL upscale_harmonic(pms_Td, up_pms(i_HRU,7))
848             else
849                 up_pms(i_HRU,7) = pm_map_all(1,1,6)
850             end if
851
852             if (pedo_tf_all(7) > 1) then
853                 CALL extract_pm_HRU(i_HRU,HRU_map, pm_map_all(:,:,7),  pms_Smax)
854                 CALL upscale_arithmetic(pms_Smax, up_pms(i_HRU,8))
855             else
856                 up_pms(i_HRU,8) = pm_map_all(1,1,7)
857             end if
858
859
860             !call cpu_time(finish)
861             !print '("Time = ",f6.3," seconds.")',finish-start
862         END DO
863
864
865         !save parameter maps if required
866         IF (save_pm_maps > 1) THEN
867         print *,
868         print *, 'Saving the following parameter maps:'
869             write(i_n_str,fmt) (i_n+start_seed-1)
870
871             !match maps to HRU map
872             nrows=size(HRU_map(:,1))
873             ncols=size(HRU_map(1,:))
874
875             allocate(pm_map_all_temp(nrows,ncols,8))
876
877             DO i = 1,7
```

```
878                 DO x = 1,ncols
879                     DO y = 1,nrows
880                         IF (HRU_map(y,x) == -9999) THEN !if nodata in HRU map
881                             pm_map_all_temp(y,x,:) = -9999
882                         ELSEIF (HRU_map(y,x) == 0) THEN !nodata in HRU map
883                             pm_map_all_temp(y,x,:) = 0
884                         ELSE
885                             pm_map_all_temp(y,x,i) = up_pms(int(HRU_map(y,x)),i+1)
886                         END IF
887                         IF (pm_map_all_temp(y,x,i)>=100000) THEN
888                             pm_map_all_temp(y,x,i) = 100000
889                         END IF
890                     END DO
891                 END DO
892
893             !get filename for each different parameter
894             IF (i==1) THEN
895                 temp_fn = trim(folder_output) // "upscaled_param_map_szm" // &
896                 trim(i_n_str)//".asc"
897             ELSE IF(i==2) THEN
898                 temp_fn = trim(folder_output) // "upscaled_param_map_lnto" // &
899                 trim(i_n_str)//".asc"
900             ELSE IF(i==3) THEN
901                 temp_fn = trim(folder_output) // "upscaled_param_map_srmax" // &
902                 trim(i_n_str)//".asc"
903             ELSE IF(i==4) THEN
904                 temp_fn = trim(folder_output) // "upscaled_param_map_srinit" // &
905                 trim(i_n_str)//".asc"
906             ELSE IF(i==5) THEN
907                 temp_fn = trim(folder_output) // "upscaled_param_map_chv" // &
908                 trim(i_n_str)//".asc"
909             ELSE IF(i==6) THEN
910                 temp_fn = trim(folder_output) // "upscaled_param_map_td" // &
911                 trim(i_n_str)//".asc"
912             ELSE IF(i==7) THEN
913                 temp_fn = trim(folder_output) // "upscaled_param_map_smax" // &
914                 trim(i_n_str)//".asc"
915             END IF
916
917             !only save parameter map if this parameter has had MPR applied
918             IF (pedo_tf_all(i)>1) THEN
919                 call write_ascii_grid(temp_fn,pm_map_all_temp(:,:,i),&
920                 size(HRU_map(1,:)),size(HRU_map(:,1)),&
921                     stats_HRU(1),stats_HRU(2),stats_HRU(3),stats_HRU(4),6)
922                 print *,'   Saving parameter map: ',trim(temp_fn)
923             END IF
924             END DO
925         END IF
926
927
928     print *, "upscaling done."
929     end subroutine mpr_upscale
930
931
932     ! ************************************************************************************************
```

```
933     ! subroutine: 6. mpr_write_paramfile
934     ! ************************************************************************************************
935     ! Writes a DYMOND parameter file - using parameters created within MPR.
936     subroutine mpr_write_paramfile(i_n, &
937         start_seed, &
938         folder_output, &
939         nHRUs, &
940         up_pms)
941 
942         use mpr_utility
943 
944         implicit none
945 
946         !DECLARE DUMMY VARIABLES
947         integer, intent(in)                                     :: i_n
948         integer, intent(in)                                     ::  start_seed
949         character(len=1024), intent(in)                         :: folder_output
950         integer, intent(in)                                     :: nHRUS
951         double precision, allocatable, dimension(:,:), intent(in)   :: up_pms
952 
953         !DECLARE LOCAL VARIABLES
954 
955         !variables for saving Dynamic TOPMODEL parameter file
956         !stores dynamic filename
957         character(len=1024)                             :: temp_fn
958         !string storing current number of pm files made
959         character(len = 5)                              :: i_n_str
960         !parameter file headers
961         character(len=64), dimension(8)                 :: headers
962         !format descriptor - integer of width 5 with zeros on left
963         character(len = 6)                              :: fmt = '(I5.5)'
964 
965         write(i_n_str,fmt) (i_n+start_seed-1)
966 
967         headers(1) = 'param_layer'
968         headers(2) = 'szm_def'
969         headers(3) = 'lnto_def'
970         headers(4) = 'srmax_def'
971         headers(5) = 'srinit_def'
972         headers(6) = 'chv_def'
973         headers(7) = 'td_def'
974         headers(8) = 'smax_def'
975 
976         temp_fn = trim(folder_output) // '/param_file_' // trim(i_n_str) // '.txt'
977         print *, 'Writing param file: ',trim(temp_fn)
978         open(unit=24,file=temp_fn,status='REPLACE',action = 'WRITE')
979         !write(24, FMT=*) 'hello'
980         write(24, FMT=*) nHRUs,size(up_pms(1,:)),&
981         ' !number of parameter layers(rows/hrus), number of param names (columns-1)'
982 
983         CALL write_numeric_list_append(headers , up_pms, 6,24)
984         write(24, FMT=*) '1 0.8 0.000000001 KINEMATIC    SOLUTION     PARS'
985         close(unit=24)
986 
987     end subroutine mpr_write_paramfile
```

```
988
989
990 end module mpr_main
```

C.4 mpr_control.f90

```
!! Module containing subroutines for MPR.f90
!! This module contains subroutines to interact with the control file
!
!   read_control_file_int   Read integer variables from the control file
!   read_param_file         reads parameter min/max values from file
!   file_open_err           Open file with error checking
!   read_mpr_commands       Reads the MPR folder and fpath_HRUmap from the command line

!!  Rosie Lane - 31st August 2017

module mpr_control
contains

    ! ************************************************************************************************
    ! subroutine: read_mpr_commands : Reads the MPR folder and fpath_HRUmap from the command line
    ! ************************************************************************************************
    ! This function reads the MPR folder and filepath to the HRU map from the command line
    ! It then makes the strings folder_input, folder_output and fname_control based on
    ! the location of the MPR main folder.

    subroutine read_mpr_commands(MPRfolder,fpath_HRUmap,folder_input,folder_output,fname_control)

        use dta_utility

        implicit none

        !declare dummy variables
        character(len=1024), intent(out)        :: MPRfolder
        character(len=1024), intent(out)        :: fpath_HRUmap
        character(len=1024), intent(out)        :: folder_input
        character(len=1024), intent(out)        :: folder_output
        character(len=1024), intent(in)        :: fname_control

        !declare local variables
        character(len=1024)                     :: arg
        logical                                 :: input_is_valid
        integer                                 :: i

        ! initialise
        i = 0
        fpath_HRUmap = ""

        ! search for command line input:
        do
            CALL get_command_argument(i, arg)
            if(len_trim(arg) == 0) exit
            if (are_equal(arg, '-MPRfolder')) then
                CALL get_command_argument(i+1, MPRfolder)
                input_is_valid = .true.
            elseif (are_equal(arg, '-fpath_HRUmap')) then
                CALL get_command_argument(i+1, fpath_HRUmap)
            endif
```

```
53              i = i + 1
54          enddo
55
56          if (len_trim(MPRfolder) == 0) then
57              print *, '-MPRfolder not specified'
58              input_is_valid = .false.
59          elseif (len_trim(fpath_HRUmap) == 0) then
60              print *, '-fpath_HRUmap not specified'
61              input_is_valid = .false.
62          endif
63
64          if(input_is_valid .eqv. .false.) then
65              print *, 'ERROR: MPR requires the following command options:'
66              print *, "-MPRfolder <Full path to MPR folder> e.g. '/home/rl1023/2017_07_20_MPR_setup/'"
67              print *, "-fpath_HRUmap <Full path to HRU map file>   "
68              print *, "e.g. '/home/rl1023/2017_07_20_MPR_setup/INPUT/54057_classarray.asc'"
69              stop
70          endif
71
72          folder_input = trim(MPRfolder) // "INPUT"
73          folder_output = trim(MPRfolder) // "OUTPUT"
74
75          print *, 'MPR folder given: ',trim(MPRfolder)
76          !print *, 'MPR control file given: ', trim(fname_control)
77          print *, 'HRU map location given: ',trim(fpath_HRUmap)
78          !print *, 'Basin predictor files should be stored in: ',trim(folder_input)
79          !print *, 'Parameter files will be stored in: ', trim(folder_output)
80
81      end subroutine read_mpr_commands
82
83
84      ! ************************************************************************************************
85      ! subroutine: read_param_file : reads parameter min/max values from file
86      ! ************************************************************************************************
87      ! This function reads the parameter file.
88
89      subroutine read_param_file(fpath,SZM_range,LnTo_range,SRmax_range,SRinit_range,CHV_range, &
90          Td_range,Smax_range)
91
92          implicit none
93
94          !declare dummy variables
95          character(len=1024), intent(in)                 :: fpath    !Full filename
96          double precision, dimension(3), intent(out)     :: SZM_range
97          double precision, dimension(3), intent(out)     :: LnTo_range    !min, best and max values
98          double precision, dimension(3), intent(out)     :: SRmax_range
99          double precision, dimension(3), intent(out)     :: SRinit_range
100         double precision, dimension(3), intent(out)     :: CHV_range
101         double precision, dimension(3), intent(out)     :: Td_range
102         double precision, dimension(3), intent(out)     :: Smax_range
103
104         !declare local variables
105         character(len=1024)             :: temp_fn      !filename of parameter file
106         character(len=64)               :: name         !name of parameter in the parameter file
107         double precision, dimension(3)  :: dth1_range
```

```
108         double precision, dimension(3)  :: fracdir_range
109
110
111         !open the file
112         temp_fn = fpath
113         CALL file_open_err(temp_fn,2)
114
115         !skip header lines
116         Read(2,*)
117         Read(2,*)
118
119         !read parameters from each line
120         Read(2,*) SZM_range(1), SZM_range(2), SZM_range(3), name
121         IF (trim(name) /= "SZM") THEN
122             print *, "ERROR IN PARAMETER RANGE FILE : SZM expected on line 2"
123             stop
124         END IF
125
126         Read(2,*) LnTo_range(1), LnTo_range(2), LnTo_range(3), name
127         IF (trim(name) /= "LnTo") THEN
128             print *, "ERROR IN PARAMETER RANGE FILE : LnTo expected on line 3"
129             stop
130         END IF
131
132         Read(2,*) SRmax_range(1), SRmax_range(2), SRmax_range(3), name
133         IF (trim(name) /= "SRmax") THEN
134             print *, "ERROR IN PARAMETER RANGE FILE : SRmax expected on line 4"
135             stop
136         END IF
137
138         Read(2,*) SRinit_range(1), SRinit_range(2), SRinit_range(3), name
139         IF (trim(name) /= "SRinit") THEN
140             print *, "ERROR IN PARAMETER RANGE FILE : SRinit expected on line 5"
141             stop
142         END IF
143
144         Read(2,*) CHV_range(1), CHV_range(2), CHV_range(3), name
145         IF (trim(name) /= "CHV") THEN
146             print *, "ERROR IN PARAMETER RANGE FILE : CHV expected on line 6"
147             stop
148         END IF
149
150         Read(2,*) Td_range(1), Td_range(2), Td_range(3), name
151         IF (trim(name) /= "Td") THEN
152             print *, "ERROR IN PARAMETER RANGE FILE : Td expected on line 7"
153             stop
154         END IF
155
156         Read(2,*) dth1_range(1), dth1_range(2), dth1_range(3), name
157         IF (trim(name) /= "dth1") THEN
158             print *, "ERROR IN PARAMETER RANGE FILE : dth1 expected on line 8"
159             stop
160         END IF
161
162         Read(2,*) Smax_range(1), Smax_range(2), Smax_range(3), name
```

```
163         IF (trim(name) /= "Smax") THEN
164             print *, "ERROR IN PARAMETER RANGE FILE : Smax expected on line 9"
165             stop
166         END IF
167
168         Read(2,*) fracdir_range(1), fracdir_range(2), fracdir_range(3), name
169         IF (trim(name) /= "fracdir") THEN
170             print *, "ERROR IN PARAMETER RANGE FILE : fracdir expected on line 10"
171             stop
172         END IF
173
174
175         print *, "Parameter file successfully read-in. "
176         !print *, "SZM min: ",SZM_range(1)," SZM max: ",SZM_range(3)
177         print *,
178
179
180     end subroutine
181
182
183
184
185
186     ! ************************************************************************************************
187     ! subroutine: read_parammap_perturbations file
188     ! ************************************************************************************************
189     ! This function reads values from the parammap_perturbations file.
190     ! These are the perturbations to the parameter maps if re-running a map for a set parameter.
191
192     subroutine read_parammap_pert_file(fpath,parammap_pert)
193
194             implicit none
195
196         !declare dummy variables
197         character(len=1024), intent(in)                 :: fpath   !Full filename
198          !returned ranges for additions and multiplications to parameter map.
199         double precision, dimension(7,4), intent(out)   :: parammap_pert
200
201         !declare local variables
202         character(len=64)               :: name         !name of parameter in the parameter file
203         integer                         :: i
204
205         !open the file
206         CALL file_open_err(fpath,2)
207
208         !skip header lines
209         Read(2,*)
210         Read(2,*)
211
212         !read parameters from each line
213         DO i = 1,7
214             READ(2,*) parammap_pert(i,1), parammap_pert(i,2), parammap_pert(i,3), &
215                 parammap_pert(i,4), name
216             write(*,"(A28,A6,A3,F5.2,F5.2,F5.2,F5.2)") 'Set perturbation ranges for ',&
217             trim(name),' = ',parammap_pert(i,:)
```

```
218         END DO
219
220
221     end subroutine
222
223
224
225     ! ************************************************************************************************
226     ! subroutine: read_control_file_int : Read integer variables from the control file
227     ! ************************************************************************************************
228     ! Given a variable name, this function checks the control file for that variable, and returns its
229     ! associated integer vale.
230
231     subroutine read_control_file_int(fname_control, varname, value,default_value)
232
233         use dta_utility
234
235         implicit none
236
237         !declare dummy variables
238         character(len=1024),intent(in)              :: fname_control       !filepath to control file
239         character(len=*),intent(in)                 :: varname      !name of variable
240         integer,intent(out)                         :: value        !variable values
241         !default value to set if variable missing from control file
242         integer, intent(in)                         :: default_value
243
244         !declare local variables
245         !variable name read from control file
246         character(len=1024)                         :: varname_read
247         integer                                     :: value_read
248         logical                                     :: var_unfound
249         logical                                     :: var_missing
250
251         !initialise local variables
252         var_unfound = .true.
253         var_missing = .true.
254
255         CALL file_open_err(fname_control,2)
256
257         READ(2,*) !read 1 header lines and ignore content
258
259         DO WHILE (var_unfound )  !read all lines until finding the variable name of interest
260             READ(2,*) varname_read, value_read
261             if (are_equal(trim(varname), trim(varname_read))) then
262                 value = value_read
263                 print *, 'User defined ',trim(varname),' = ',value
264                 !print *, value ,' has been selected for ',trim(varname)
265                 var_unfound = .false.
266                 var_missing = .false.
267             elseif (are_equal(trim(varname_read),'end')) then
268
269                 var_unfound = .false.
270             endif
271
272
```

```
273         END DO
274
275         close(2)
276
277         !If variable not declared in control file, then set to default value.
278         IF (var_missing) THEN
279             value = default_value
280             !print *, 'IMPORTANT: variable ', trim(varname),' not specified in control file'
281             print *, 'Default defn ',trim(varname),' = ',value
282         ENDIF
283
284
285     end subroutine read_control_file_int
286
287
288
289     ! ************************************************************************************************
290     ! subroutine: read_control_file_int : Read integer variables from a file
291     ! ************************************************************************************************
292     ! Given a variable name, this function checks the control file for that variable, and returns its
293     ! associated integer vale.
294
295     subroutine read_control_file_char(fname_control, varname, value,default_value)
296
297         use dta_utility
298
299         implicit none
300
301         !declare dummy variables
302         character(len=1024),intent(in)              :: fname_control       !filepath to control file
303         character(len=*),intent(in)                 :: varname      !name of variable
304         character(len=1024),intent(out)             :: value        !variable values
305         !default value to set if variable missing from control file
306         character(len=1024), intent(in)             :: default_value
307
308         !declare local variables
309         !variable name read from control file
310         character(len=1024)                         :: varname_read
311         character(len=1024)                         :: value_read
312         logical                                     :: var_unfound
313         logical                                     :: var_missing
314
315         !initialise local variables
316         var_unfound = .true.
317         var_missing = .true.
318
319         CALL file_open_err(fname_control,2)
320
321         READ(2,*) !read 1 header lines and ignore content
322
323         DO WHILE (var_unfound )  !read all lines until finding the variable name of interest
324             READ(2,*) varname_read, value_read
325             if (are_equal(trim(varname), trim(varname_read))) then
326                 value = value_read
327                 print *, 'User defined ',trim(varname),' = ',trim(value)
```

```
328                 !print *, value ,' has been selected for ',trim(varname)
329                 var_unfound = .false.
330                 var_missing = .false.
331             elseif (are_equal(trim(varname_read),'end')) then
332
333                 var_unfound = .false.
334             endif
335             !print *, 'Read varname: ',trim(varname_read)
336             !print *, 'Read value: ',trim(value_read)
337             !print *,
338
339         END DO
340
341         close(2)
342
343         !If variable not declared in control file, then set to default value.
344         IF (var_missing) THEN
345             value = default_value
346             !print *, 'IMPORTANT: variable ', trim(varname),' not specified in control file'
347             print *, 'Default defn ',trim(varname),' = ',trim(value)
348         ENDIF
349
350
351     end subroutine read_control_file_char
352
353
354
355
356
357
358     ! ********************************************************************************************
359     ! subroutine: file_open_err : Open file with error checking
360     ! ********************************************************************************************
361     ! Checks that the file to be opened exists, and is not already in use.
362
363     subroutine file_open_err(infile,unt)
364
365         implicit none
366
367         !declare dummy variables
368         character(*),intent(in)             :: infile      ! filename
369         integer,intent(in)                  :: unt         ! file unit
370         character(len=1024)                 :: message     ! error message
371
372         !declare local variables
373         integer                             :: err
374         logical                             :: xist        ! .TRUE. if the file exists
375         logical                             :: xopn        ! .TRUE. if the file is already open
376
377         ! initialize errors
378         err=0; message="f-file_open/"
379
380         ! check if the file exists
381         inquire(file=trim(infile),exist=xist)
382         if(.not.xist)then
```

```
383             message=trim(message)//"FileNotFound[file='"//trim(infile)//"']"
384             err = 1
385             print *, message
386             return
387         endif
388 
389         ! check if the file is already open
390         inquire(file=trim(infile),opened=xopn)
391         if(xopn)then
392             message=trim(message)//"FileIsAlreadyOpen[file='"//trim(infile)//"']"
393             print *, message
394             err = 1
395             return
396         endif
397 
398         ! open file
399         open(unt,file=trim(infile),status="old",action="read",iostat=err)
400         if(err/=0)then
401             message=trim(message)//"OpenError['"//trim(infile)//"']"
402             print *, message
403             return
404         endif
405 
406     end subroutine file_open_err
407 
408 
409 
410 end module mpr_control
```

C.5 mpr_SZM.f90

```
!! Module containing subroutines for MPR.f90
!! This module contains all subroutines involving MPR for the SZM parameter:
!
!   get_BasPred_SZM    Complete logical list of basin predictors required for SZM
!   init_gp_SZM        Initialise global parameters for SZM
!   pedotf_SZM         Pedo-transfer function for SZM

!! Rosie Lane - 31st August 2017

module mpr_SZM
contains


    ! ***********************************************************************************************
    ! subroutine: get_BasPred_SZM : Complete logical list of basin predictors required
    ! ***********************************************************************************************
    ! Given the user-specified pedo transfer function selected for SZM, this subroutine adds to a
    ! logical list of required basin predictors, by specifying which predictors it requires.

    subroutine get_BasPred_SZM(pedo_tf_SZM, req_soils)

        implicit none

        ! declare dummy arguments
        integer, intent(inout)                  :: pedo_tf_SZM !pedo_tf_equation to be used
        logical, dimension(*), intent(inout)  :: req_soils    !soils data required logical

        ! define required BasinPredictors for SZM
        select case(pedo_tf_SZM)

            case (0)    !fixed parameter - no basin predictors required

            case (1)    !global parameter - no basin predictors required

            case (2)     !Use a pedo-transfer equation

                req_soils(5) = .true.   !musid map
                !Using tables of exponential decline profiles from soils data.
                req_soils(7:11) = (/.true.,.true.,.true.,.true.,.true./)

                !using form of exponential decline in ksat from soils data
                !req_soils(5,1) = (.true.)
                !print *, "ERROR: A transfer function has not yet been programmed for SZM"
                !print *, "Select pedo_tf = 0 to treat it as a fixed parameter"
                !print *, "Select pedo_tf = 1 to treat it as a global parameter"
                !stop
            case (3)    !Same ptf equation as case 2, but using same global parameters as lnto

                req_soils(5) = .true.   !musid map
                 !Using tables of exponential decline profiles from soils data.
                req_soils(7:11) = (/.true.,.true.,.true.,.true.,.true./)

```

```
53            case (4)
54
55                req_soils(5) = .true.    !musid map
56                !Using tables of exponential decline profiles from soils data.
57                req_soils(7:11) = (/.true.,.true.,.true.,.true.,.true./)
58            case (5)
59
60                req_soils(5) = .true.    !musid map
61                !Using tables of exponential decline profiles from soils data.
62                req_soils(7:11) = (/.true.,.true.,.true.,.true.,.true./)
63
64
65            case default
66                print *, "WARNING: a valid pedo-tf equation for SZM must be specified in the", &
67                    " control file"
68                print *, "SZM will be set to the default of fixed parameter"
69                print *, "The following options can be selected in the control file:"
70                print *, "pedo_tf_SZM = 0, sets it as a fixed parameter"
71                print *, "pedo_tf_SZM = 1, sets it as a global parameter"
72                print *,
73
74                pedo_tf_SZM = 0
75
76        end select
77
78    end subroutine get_BasPred_SZM
79
80
81
82
83
84    ! ************************************************************************************************
85    ! subroutine: init_gp_SZM : Initialise global parameters for SZM
86    ! ************************************************************************************************
87    ! This subroutine does the following:
88    !   1. Defines how many global parameters (n_glob_pms) are needed for the user-specified
89    !   transfer function
90    !       TF 0    : 1 global parameter
91    !       TF 1    : 3 global parameters
92    !   2. Defines min/max ranges for global parameters
93    !   3. Generates list of global parameters using the rand function
94    !       - user specified start_seed can be used to skip to any point in this list
95    !       - global parameter list is of dimensions number of global params by number of param
96    !           files needed
97    !   4. Transforms global parameters to be within set min/max ranges
98
99    subroutine init_gp_SZM(pedo_tf_SZM,glob_pms_SZM, n_pm_maps, start_seed, n_gp_all,pm_range)
100
101       use dta_utility
102       use mpr_LnTo
103
104       implicit none
105
106       !declare dummy variables
107       integer, intent(in)                                    :: pedo_tf_SZM
```

```
108         !global param list
109         double precision, allocatable, dimension(:,:),intent(inout) :: glob_pms_SZM
110         integer, intent(in)                                          :: n_pm_maps
111          !to start rand num generator
112         integer, intent(in)                                          :: start_seed
113         !vector of number of global params for all params
114         integer, dimension(7), intent(inout)                         :: n_gp_all
115          !min, fixed, max values for this parameter
116         double precision, dimension(3), intent(in)                   :: pm_range
117
118         !declare local variables
119         integer                                     :: n_glob_pms   !number of global parameters
120         double precision, allocatable, dimension(:) :: min_gp   !min values for global parameters
121         double precision, allocatable, dimension(:) :: max_gp
122         real, dimension(:,:),allocatable            :: rand_nums
123         double precision                            :: num
124         integer, dimension(12)                      :: seed
125         integer                                     :: i,j, rn_i
126
127
128         !1. Define how many global parameters should exist and min/max ranges, based on selected
129         !       transfer function
130         select case(pedo_tf_SZM)
131
132             case (0)  !Treat as fixed parameter (for ease this is a global parameter with a 0 range)
133
134                 print *, "SZM is a fixed parameter, of value ",pm_range(2)
135                 n_glob_pms = 1
136                 allocate(min_gp(1)) !min and max values have length 1, as we have 1 global parameter
137                 allocate(max_gp(1))
138                 min_gp(1) = pm_range(2)    !min and max values set to same fixed parameter value.
139                 max_gp(1) = pm_range(2)
140
141             case (1)    !Treat as global parameter
142
143                 print *, "SZM is a global parameter, in the range ",pm_range(1)," ",pm_range(3)
144                 n_glob_pms = 1
145                 allocate(min_gp(1)) !min and max values have length 1, as we have 1 global parameter
146                 allocate(max_gp(1))
147                 min_gp(1) = pm_range(1)    !min and max values set from parameter file
148                 max_gp(1) = pm_range(3)
149
150             case (2) !pedo-transfer function based on soil ksat declines
151
152                 print *, "SZM will be parameterised using ksat declines from soils data"
153                 !n_glob_pms = 2
154                 !allocate(min_gp(2))
155                 !allocate(max_gp(2))
156                 !min_gp(1)=-0.01
157                 !max_gp(1)=0.05
158                 !min_gp(2)=0.5
159                 !max_gp(2)=1.5
160
161                 !pedotransfer function used will be the same as lnto transfer function 2
162                 n_glob_pms = 7         !a_const, a_sand, a_clay,
```

```
163                 allocate(min_gp(7))
164                 allocate(max_gp(7))
165                 min_gp(1) = -4.5 ! was -3.5 - changed following dotty plots of performance
166                                  ! showing lower values do better
167                 min_gp(2)= 0.006
168                 min_gp(3) = -0.02
169                 max_gp(1) = -1 ! was 0.3 - changed following dotty plots of performance
170                                ! showing lower values do better
171                 max_gp(2)= 0.03
172                 max_gp(3) = -0.0032
173
174                 !HYPRES limits Ks = a + b(BulkDensity)^2 + c(OrganicMatter)^-1 +
175                 !       d(BulkDensity * OrganicMatter)
176                 min_gp(4) = 3
177                 max_gp(4) = 11
178
179                 min_gp(5) = -0.5
180                 max_gp(5) = -1.3
181
182                 min_gp(6) = -0.03
183                 max_gp(6) = -0.1
184
185                 min_gp(7) = -0.8
186                 max_gp(7) = -0.24
187
188              case (3) !pedo-transfer function based on soils ksat declines
189                       ! using same global parameters as SZM
190
191                 n_glob_pms = 1
192                 allocate(min_gp(1))
193                 allocate(max_gp(1))
194                 max_gp(1) = 1
195                 min_gp(1) = 1
196
197                 print *, "SZM will be parameterised using ksat declines from soils data and the", &
198                     " same parameters as lnT0"
199
200            !case(4) redirects to case(2)
201
202            case (5) !pedo-transfer function based on soil ksat declines
203
204                 print *, "SZM will be parameterised using ksat declines from soils data"
205                 !n_glob_pms = 2
206                 !allocate(min_gp(2))
207                 !allocate(max_gp(2))
208                 !min_gp(1)=-0.01
209                 !max_gp(1)=0.05
210                 !min_gp(2)=0.5
211                 !max_gp(2)=1.5
212
213                 !pedotransfer function used will be the same as lnto transfer function 2
214                 n_glob_pms = 8          !a_const, a_sand, a_clay,
215                 allocate(min_gp(8))
216                 allocate(max_gp(8))
217                 min_gp(1) = -4.5 ! was -3.5 - changed following dotty plots of performance showing
```

```
                                ! lower values do better
            min_gp(2)= 0.006
            min_gp(3) = -0.02
            max_gp(1) = -1! was 0.3 - changed following dotty plots of performance showing lower
                          ! values do better
            max_gp(2)= 0.03
            max_gp(3) = -0.0032

            !HYPRES limits Ks = a + b(BulkDensity)^2 + c(OrganicMatter)^-1 +
            !       d(BulkDensity * OrganicMatter)
            !changed to match ln(T0) values from testing theoretical min and max ranges.
            min_gp(4) = 1 !3
            max_gp(4) = 6!11

            min_gp(5) = -1.3!-0.5
            max_gp(5) = -0.5!-1.3

            min_gp(6) = -0.1
            max_gp(6) = -0.003

            min_gp(7) = -0.4!-0.8
            max_gp(7) = 0!-0.24

            min_gp(8) = 1
            max_gp(8) = 4



        end select

        !2. From above, set sizes of the global parameter array and number of random numbers
        !   to produce.
        n_gp_all(1) = n_glob_pms
        !print *, 'n_gp_all(1) = ',n_gp_all

        allocate(glob_pms_SZM(n_glob_pms, n_pm_maps))
        allocate(rand_nums(sum(n_gp_all), n_pm_maps+start_seed-1))

        !generate lists of global parameters between min and max ranges, of length n
        !need to have list such that starting seed = 50 would give same result as 50th number
        !from seed=1
        Call RANDOM_SEED( GET = seed )
        seed(1:12) = (/ 3,3,3,3,3,3,3,3,3,3,3,3/)
        Call RANDOM_SEED( PUT = seed )
        call RANDOM_NUMBER ( rand_nums )
        !print *, rand_nums

        print *, 'SZM pedo-transfer has ',n_glob_pms,' global parameters'
        !print *, 'max values for SZM global param(s) : ', max_gp
        !print *, 'min values for SZM global param(s) : ', min_gp
        !print *, 'sum n_gp_all SZM = ',sum(n_gp_all)


        !now normalise random numbers to within bounds
        DO i = 1,n_glob_pms
```

```
273             DO j = start_seed, start_seed+n_pm_maps-1
274                 !rn_i is index for the random number
275                 rn_i = (sum(n_gp_all) - n_glob_pms )+ i
276                 !the appropriate random number, in range 0-1
277                 num = rand_nums(rn_i,j)
278                 !normalise the random number to given min/max
279                 num = (num * (max_gp(i)-min_gp(i)))+min_gp(i)
280                 glob_pms_SZM(i, j-start_seed+1) = num
281 
282             END DO
283             IF (n_pm_maps >= 2) THEN
284                 print *, 'First 2 values for parameter ',i,' : ',glob_pms_SZM(i,1:2)
285             ELSE
286                 print *, 'Value for parameter ',i,' : ',glob_pms_SZM(i,1)
287             END IF
288         END DO
289 
290 
291         print *,
292 
293 
294     end subroutine init_gp_SZM
295 
296 
297 
298 
299 
300     ! ************************************************************************************************
301     ! subroutine: pedotf_SZM: Pedo-transfer function for SZM
302     ! ************************************************************************************************
303     ! This routine takes n_i, signalling that we are now calculating parameter map n_i out of n,
304     ! It takes the list of global parameters and required basin predictors, and from that
305     ! applies the pedo-transfer functions to produce a parameter map.
306 
307     subroutine pedotf_SZM(n_i, pedo_tf, glob_pms,bp, pm_map,soilmusiddata,folder_output,lnto_map)
308 
309         use mpr_utility
310 
311         implicit none
312 
313         !declare dummy variables
314         !param map number to do
315         integer, intent(in)                                         :: n_i
316         !equation selection
317         integer, intent(in)                                         :: pedo_tf
318         !list of global params for SZM
319         double precision, dimension(:,:), allocatable, intent(in)   :: glob_pms
320         real, dimension(:,:,:), allocatable, intent(in):: bp          !basin predictor maps
321         !output parameter map
322         double precision, dimension(:,:), allocatable, intent(out)  :: pm_map
323         !musiddatatables
324         double precision, allocatable, dimension(:,:,:), intent(in) :: soilmusiddata
325         !MPR output folder
326         character(len=1024), intent(in)                             :: folder_output
327         double precision, dimension(:,:), allocatable, intent(in)   :: lnto_map
```

```
328
329         !declare local variables
330         integer             :: n_glob_pms       !number of global parameters
331         integer             :: n_pm_maps        !number of parameter maps,
332                                                 !defined by ncols in glob_pms_LnTo
333         integer             :: i,j,d,e          !incrementers.
334         integer             :: nrows_bp         !number of rows in basin predictor map
335         integer             :: ncols_bp
336         integer             :: nmusids          !number of musids in table.
337         integer             :: nentries         !number of entries (series'/landuses) for this musid
338         integer             :: mid_i
339         integer             :: row_start
340         integer             :: row_end
341         integer             :: nrows
342         integer             :: len_include_data
343         integer,dimension(5):: d_s      !start and end depths of depth classes
344         integer,dimension(5):: d_e
345         !keep track of where nodata values are.
346         integer,allocatable,dimension(:,:)              :: nodata_found
347         integer             :: id_s, id_e
348         character(len=1024) :: temp_filename
349         character(len=2)    :: depth_string
350         character(len=5)    :: ni_string
351         double precision    :: a,b              !exponential fit constants
352         double precision, allocatable, dimension(:) :: a_all, b_all
353         !depth decline data
354         double precision, allocatable, dimension(:,:,:) :: dep_data
355         !depth decline data - ksat values
356         double precision, allocatable, dimension(:,:,:) :: ptf_data
357         !depth decline data - ready for fitting
358         double precision, allocatable, dimension(:,:)   :: fit_data
359         !weights for each series/LU within a musid
360         double precision, allocatable, dimension(:)     :: weights
361         double precision, allocatable, dimension(:)     :: musid_szm
362         double precision, allocatable, dimension(:)     :: musid_area
363         !areas under ksat curve (integreated)
364         double precision                                :: area_data
365
366         !find number of rows and columns in basin predictor files - pm map should be same size
367         nrows_bp = size(bp(:,1,1))
368         ncols_bp = size(bp(1,:,1))
369         n_pm_maps = size(glob_pms(1,:))
370         !find number of rows and columns in basin predictor files - pm map should be same size
371         n_pm_maps = size(glob_pms(1,:))
372         allocate(pm_map(nrows_bp, ncols_bp))
373
374         !vary transfer function, depending on user-inputted equation selection.
375         select case(pedo_tf)
376
377             case (0)    !Fixed parameter
378
379                 n_glob_pms = 1
380                 DO i = 1, nrows_bp
381                     DO j = 1, ncols_bp
382
```

```
                    pm_map(i,j) = glob_pms(1, n_i)

                END DO
            END DO

        case (1)    !Global parameter

            n_glob_pms = 1
            DO i = 1, nrows_bp
                DO j = 1, ncols_bp

                    pm_map(i,j) = glob_pms(1, n_i)

                END DO
            END DO

        case (2)    !Pedotransfer function using sand and clay.
            n_glob_pms = 2

            !initialise log files

            write(ni_string,"(I5.5)") n_i
            temp_filename = trim(folder_output) // "SZM_pedotransfer_" // ni_string //".log"
            !write(temp_filename, "(A16,I2,A4)") 'SZM_pedotransfer', n_i, '.log'
            OPEN(unit=99,file=trim(temp_filename), status='unknown')
            write(99,*) 'File contains SZM ksat values, calculated using pedotransfer case 2.'
            write(99,*) 'DEPTH, MUSID, ENTRY, WEIGHT, KSAT'
            CLOSE(99)

            write(ni_string,"(I5.5)") n_i
            temp_filename = trim(folder_output) // "SZM_fits_" // ni_string //".log"
            !write (temp_filename, "(A9,I2,A4)") "SZM_fits_", n_i,".log"
            OPEN(unit = 99,file=trim(temp_filename),status='unknown')
            WRITE(99,*) 'MUSID_NUM,SERIES,ENTRY_NUM, a(1)/b(2), FIT VALUES'
            CLOSE(99)

            !loop through every MUSID
            nmusids = maxval(soilmusiddata(:,1,1))
            allocate(musid_szm(nmusids))
            print *,maxval(soilmusiddata(:,1,1))
            row_start = 1

            DO i = 1,nmusids

                !allocate arrays for number of entries in this musid
                row_end = row_start + soilmusiddata(row_start,17,1) -1
                nrows = (row_end - row_start) +1
                !dep_data (n_entries, n_info_columns, n_depth_profiles)
                allocate(dep_data(nrows,17,5))
                !ptf_data (n_entries, MUSID/ENTRY_NUM/WEIGHTING/KSAT, n_depths)
                allocate(ptf_data(nrows,4,5))
                allocate(a_all(nrows))
                allocate(b_all(nrows))
                allocate(weights(nrows))
                allocate(nodata_found(nrows,5))
```

```
438
439                 !!Extract all depth declines for this musid - looping through depths
440                 DO d = 1,5
441                     dep_data(1:nrows,1:17,d) = soilmusiddata(row_start:row_end,1:17,d)
442                 END DO
443
444                 !apply pedotransfer function to get ksat values
445                 ptf_data(1:nrows,1:3,1:5) = dep_data(1:nrows,1:3,1:5)
446                 DO e = 1,nrows
447                     DO d = 1,5
448                          !apply Cosby et al. (1984) for mineral soils
449                         !IF (dep_data(e,16,d)==0) THEN
450                         !apply Cosby et al. (1984) for minearal soils - organic
451                         !content less than 35%
452                         IF (dep_data(e,11,d)<35) THEN
453                             !ln_ksat = a + b(%sand) + c(%clay)
454                             ptf_data(e,4,d)=glob_pms(1,n_i)+(glob_pms(2,n_i)*dep_data(e,8,d))
455                             ptf_data(e,4,d)=ptf_data(e,4,d)+(glob_pms(3,n_i)*dep_data(e,10,d))
456                             !ksat = exp(ln_ksat)
457                             ptf_data(e,4,d)= 2.54*(10**(ptf_data(e,4,d)))
458
459                             !keep record of nodata values
460                             IF(int(dep_data(e,8,d))==-9999) THEN
461                                 nodata_found(e,d) = 1
462                             ELSE
463                                 nodata_found(e,d) = 0
464                             END IF
465
466                         ELSE
467                             !apply HYPRES for peaty soils, Ks = a + b(BulkDensity)^2 +
468                             !    c(OrganicMatter)^-1 + d(BulkDensity * OrganicMatter)
469                             ptf_data(e,4,d)=glob_pms(4,n_i)+(glob_pms(5,n_i)* &
470                                 dep_data(e,12,d)**2)
471                             ptf_data(e,4,d)=ptf_data(e,4,d)+(glob_pms(6,n_i)* &
472                                 dep_data(e,11,d)**(-1))
473                             ptf_data(e,4,d)=ptf_data(e,4,d)+(glob_pms(7,n_i)* &
474                                 dep_data(e,12,d)*dep_data(e,11,d))
475                             ptf_data(e,4,d) = EXP(ptf_data(e,4,d))
476
477                             !keep record of nodata values
478                             IF(int(dep_data(e,11,d))==-9999) THEN
479                                 nodata_found(e,d) = 1
480                             ELSE
481                                 nodata_found(e,d) = 0
482                             END IF
483                         END IF
484
485
486                     END DO
487                 END DO
488
489                 !write ksat declines to a log file
490                 write(ni_string,"(I5.5)") n_i
491                 temp_filename = trim(folder_output) // "SZM_pedotransfer_" // ni_string //".log"
492                 !write(temp_filename, "(A16,I2,A4)") 'SZM_pedotransfer_',n_i, '.log'
```

```
493                     OPEN(unit=99,file=trim(temp_filename), position='append',status='old')
494                     DO d = 1,5
495                         DO e = 1, nrows
496                             write(99,*) d,ptf_data(e,1:4,d)
497                         END DO
498
499                     END DO
500                     CLOSE(unit=99)
501
502
503                     !fit exponential to each ksat recession - to get m values.
504                     DO e = 1,nrows
505                         d_s = (/1,11,26,51,101/)
506                         d_e = (/10,25,50,100,150/)
507                         len_include_data = 0
508
509                         !Work out which data to include, and how to allocate fit_data.
510                         DO d = 1,5
511                             IF (nodata_found(e,d)== 0) THEN
512                                 !if fitting depths 1 to 150
513                                 !len_include_data = len_include_data + (d_e(d)-d_s(d))+1
514                                 len_include_data = len_include_data + 1
515                             END IF
516                         END DO
517
518                         !allocate fit data.
519                         allocate(fit_data(len_include_data,3))
520
521                         !extract fit data for each depth, and get integers of all the depths
522                         !to include in fit calculation.
523                         id_s = 1
524                         DO d = 1,5
525                             IF (nodata_found(e,d)== 0) THEN
526                                 fit_data(id_s,1) = ptf_data(e,4,d) !y = ksat values
527                                 fit_data(id_s,2) = -(d_e(id_s) + d_s(id_s))/2 !x = -depth
528                                 id_s = id_s + 1
529                                 !id_e = id_s + (d_e(d) - d_s(d))
530                                 !fit_data(id_s:id_e,1) =(/(j, j=d_s(d),d_e(d), 1)/)
531                                 !fit_data(id_s:id_e,2) = ptf_data(e,4,d)
532                                 !id_s = id_e + 1
533                             END IF
534                         END DO
535
536                         !calculate the exponential fit! To the fit_data which excludes all
537                         !nodata values.
538                         CALL calc_exp_fit(fit_data, a, b)
539                         a_all(e) = a
540                         b_all(e) = b
541                         deallocate(fit_data)
542
543                     END DO
544
545                     !save fit information in a log file - to check fits look reasonable in matlab
546                     write(ni_string,"(I5.5)") n_i
547                     temp_filename = trim(folder_output) // "SZM_fits_" // ni_string //".log"
```

```
548                 !write (temp_filename, "(A9,I2,A4)") "SZM_fits_", n_i,".log"
549                 OPEN(unit = 99,file=trim(temp_filename),position='append',status='old')
550                 DO e = 1,nrows
551                     WRITE(99,*) i,soilmusiddata(row_start+e-1,4,1),e,1,a_all(e)
552                     WRITE(99,*) i,soilmusiddata(row_start+e-1,4,1),e,2,b_all(e)
553                 END DO
554                 CLOSE(unit=99)
555
556
557                 !take a weighted average of all m values, to get a musid parameter value.
558                 weights = soilmusiddata(row_start:row_end,3,1)!weight is column 3.
559                 IF (sum(weights) > 1.001) THEN
560                     print *, 'error - weights within musids exceed 1'
561                 END IF
562                 IF (sum(weights) < 0.999) THEN
563                     print *, 'error - weights within musids dont add up to 1'
564                 END IF
565                 musid_szm(i) = sum(weights * b_all)
566
567                 !deallocate/set variables for reading in next musid
568                 deallocate(ptf_data)
569                 deallocate(dep_data)
570                 deallocate(a_all)
571                 deallocate(b_all)
572                 deallocate(weights)
573                 deallocate(nodata_found)
574
575                 row_start = row_end+1
576
577             END DO
578
579             !plot parameter values back in space - using musid ascii grid.
580             !   where nodata, use average szm value across modelled region.
581             DO i = 1, nrows_bp
582                 DO j = 1, ncols_bp
583                     e = INT(bp(i,j,6))
584                     IF(e<-9000) THEN
585                         pm_map(i,j) = (sum(musid_szm))/nmusids
586                     ELSE
587                         pm_map(i,j) = musid_szm(e)
588                     END IF
589                 END DO
590             END DO
591
592         case (4)    !Pedotransfer function using sand and clay -
593             !n_glob_pms = 2
594
595             !loop through every MUSID
596             nmusids = maxval(soilmusiddata(:,1,1))
597             allocate(musid_szm(nmusids))
598             allocate(musid_area(nmusids))
599             print *,maxval(soilmusiddata(:,1,1))
600             row_start = 1
601
602             DO i = 1,nmusids
```

```
603
604                     !allocate arrays for number of entries in this musid
605                     row_end = row_start + soilmusiddata(row_start,17,1) -1
606                     nrows = (row_end - row_start) +1
607                     !dep_data (n_entries, n_info_columns, n_depth_profiles)
608                     allocate(dep_data(nrows,17,5))
609                     !ptf_data (n_entries, MUSID/ENTRY_NUM/WEIGHTING/KSAT, n_depths)
610                     allocate(ptf_data(nrows,4,5))
611                     allocate(a_all(nrows))
612                     allocate(b_all(nrows))
613                     allocate(weights(nrows))
614                     allocate(nodata_found(nrows,5))
615
616                     !!Extract all depth declines for this musid - looping through depths
617                     DO d = 1,5
618                         dep_data(1:nrows,1:17,d) = soilmusiddata(row_start:row_end,1:17,d)
619                     END DO
620
621                     !apply pedotransfer function to get ksat values
622                     ptf_data(1:nrows,1:3,1:5) = dep_data(1:nrows,1:3,1:5)
623                     DO e = 1,nrows
624                         DO d = 1,5
625                             !apply Cosby et al. (1984) for mineral soils
626                             !IF (dep_data(e,16,d)==0) THEN
627                             !apply Cosby et al. (1984) for minearal soils - organic
628                             !content less than 35%
629                             IF (dep_data(e,11,d)<35) THEN
630                                 !ln_ksat = a + b(%sand) + c(%clay)
631                                 ptf_data(e,4,d)=glob_pms(1,n_i)+(glob_pms(2,n_i)*dep_data(e,8,d))
632                                 ptf_data(e,4,d)=ptf_data(e,4,d)+(glob_pms(3,n_i)*dep_data(e,10,d))
633                                 !ksat = exp(ln_ksat)
634                                 ptf_data(e,4,d)= EXP(ptf_data(e,4,d))
635
636                                 !keep record of nodata values
637                                 IF(int(dep_data(e,8,d))==-9999) THEN
638                                     nodata_found(e,d) = 1
639                                 ELSE
640                                     nodata_found(e,d) = 0
641                                 END IF
642
643                             ELSE    !apply HYPRES for peaty soils, Ks = a + b(BulkDensity)^2 +
644                                     !    c(OrganicMatter)^-1 + d(BulkDensity * OrganicMatter)
645                                 ptf_data(e,4,d)=glob_pms(4,n_i)+(glob_pms(5,n_i)* &
646                                     dep_data(e,12,d)**2)
647                                 ptf_data(e,4,d)=ptf_data(e,4,d)+(glob_pms(6,n_i)* &
648                                     dep_data(e,11,d)**(-1))
649                                 ptf_data(e,4,d)=ptf_data(e,4,d)+(glob_pms(7,n_i)* &
650                                     dep_data(e,12,d)*dep_data(e,11,d))
651                                 ptf_data(e,4,d) = EXP(ptf_data(e,4,d))
652
653                                 !keep record of nodata values
654                                 IF(int(dep_data(e,11,d))==-9999) THEN
655                                     nodata_found(e,d) = 1
656                                 ELSE
657                                     nodata_found(e,d) = 0
```

```
                    END IF
                END IF


            END DO
        END DO

        !fit exponential to each ksat recession - to get m values.
        area_data=0
        weights = soilmusiddata(row_start:row_end,3,1)
        DO e = 1,nrows
            d_s = (/1,11,26,51,101/)
            d_e = (/10,25,50,100,150/)
            len_include_data = 0

            !extract fit data for each depth, and get integers of all the depths to
            !include in fit calculation.
            id_s = 1

            DO d = 1,5
                IF (nodata_found(e,d)== 0) THEN
                    area_data = area_data + weights(e)*(ptf_data(e,4,d)* &
                        (d_e(id_s)-d_s(id_s)))
                    id_s = id_s + 1
                END IF
            END DO

        END DO

        !take a weighted average of all m values, to get a musid parameter value.
        !weights = soilmusiddata(row_start:row_end,3,1)!weight is column 3.
        IF (sum(weights) > 1.001) THEN
            print *, 'error - weights within musids exceed 1'
        END IF
        IF (sum(weights) < 0.999) THEN
            print *, 'error - weights within musids dont add up to 1'
        END IF
        musid_area(i) = area_data

        !deallocate/set variables for reading in next musid
        deallocate(ptf_data)
        deallocate(dep_data)
        deallocate(a_all)
        deallocate(b_all)
        deallocate(weights)
        deallocate(nodata_found)

        row_start = row_end+1

    END DO

    !plot parameter values back in space - using musid ascii grid.
    !   where nodata, use average szm value across modelled region.
    DO i = 1, nrows_bp
        DO j = 1, ncols_bp
```

```
713                     e = INT(bp(i,j,6))
714                     IF(e<-9000) THEN
715                         pm_map(i,j) = exp(lnto_map(i,j))/((sum(musid_area))/nmusids)
716                     ELSE
717                         pm_map(i,j) = exp(lnto_map(i,j))/musid_area(e)
718                     END IF
719                 END DO
720             END DO
721
722
723
724             case (5)    !Pedotransfer function using sand and clay -
725             !n_glob_pms = 2
726
727             !loop through every MUSID
728             nmusids = maxval(soilmusiddata(:,1,1))
729             allocate(musid_szm(nmusids))
730             allocate(musid_area(nmusids))
731             print *,maxval(soilmusiddata(:,1,1))
732             row_start = 1
733
734             DO i = 1,nmusids
735
736                 !allocate arrays for number of entries in this musid
737                 row_end = row_start + soilmusiddata(row_start,17,1) -1
738                 nrows = (row_end - row_start) +1
739                 !dep_data (n_entries, n_info_columns, n_depth_profiles)
740                 allocate(dep_data(nrows,17,5))
741                 !ptf_data (n_entries, MUSID/ENTRY_NUM/WEIGHTING/KSAT, n_depths)
742                 allocate(ptf_data(nrows,4,5))
743                 allocate(a_all(nrows))
744                 allocate(b_all(nrows))
745                 allocate(weights(nrows))
746                 allocate(nodata_found(nrows,5))
747
748                 !!Extract all depth declines for this musid - looping through depths
749                 DO d = 1,5
750                     dep_data(1:nrows,1:17,d) = soilmusiddata(row_start:row_end,1:17,d)
751                 END DO
752
753                 !apply pedotransfer function to get ksat values
754                 ptf_data(1:nrows,1:3,1:5) = dep_data(1:nrows,1:3,1:5)
755                 DO e = 1,nrows
756                     DO d = 1,5
757                         !apply Cosby et al. (1984) for mineral soils
758                         !IF (dep_data(e,16,d)==0) THEN
759                         !apply Cosby et al. (1984) for minearal soils - organic
760                         !content less than 35%
761                         IF (dep_data(e,11,d)<35) THEN
762                             !ln_ksat = a + b(%sand) + c(%clay)
763                             ptf_data(e,4,d)=glob_pms(1,n_i)+(glob_pms(2,n_i)*dep_data(e,8,d))
764                             ptf_data(e,4,d)=ptf_data(e,4,d)+(glob_pms(3,n_i)*dep_data(e,10,d))
765                             !ksat = exp(ln_ksat)
766                             ptf_data(e,4,d)= EXP(ptf_data(e,4,d))
767
```

```
768                     !keep record of nodata values
769                     IF(int(dep_data(e,8,d))==-9999) THEN
770                         nodata_found(e,d) = 1
771                     ELSE
772                         nodata_found(e,d) = 0
773                     END IF
774
775                 ELSE    !apply HYPRES for peaty soils, Ks = a + b(BulkDensity)^2 +
776                         !       c(OrganicMatter)^-1 + d(BulkDensity * OrganicMatter)
777                     ptf_data(e,4,d)=glob_pms(4,n_i)+(glob_pms(5,n_i)* &
778                         dep_data(e,12,d)**2)
779                     ptf_data(e,4,d)=ptf_data(e,4,d)+(glob_pms(6,n_i)* &
780                         dep_data(e,11,d)**(-1))
781                     ptf_data(e,4,d)=ptf_data(e,4,d)+(glob_pms(7,n_i)* &
782                         dep_data(e,12,d)*dep_data(e,11,d))
783                     ptf_data(e,4,d) = EXP(ptf_data(e,4,d))
784
785                     !keep record of nodata values
786                     IF(int(dep_data(e,11,d))==-9999) THEN
787                         nodata_found(e,d) = 1
788                     ELSE
789                         nodata_found(e,d) = 0
790                     END IF
791                 END IF
792
793
794             END DO
795         END DO
796
797         !fit exponential to each ksat recession - to get m values.
798         area_data=0
799         weights = soilmusiddata(row_start:row_end,3,1)
800         DO e = 1,nrows
801             d_s = (/1,11,26,51,101/)
802             d_e = (/10,25,50,100,150/)
803             len_include_data = 0
804
805             !extract fit data for each depth, and get integers of all the depths to
806             !include in fit calculation.
807             id_s = 1
808
809             DO d = 1,5
810                 IF (nodata_found(e,d)== 0) THEN
811                     area_data = area_data + weights(e)*(ptf_data(e,4,d)* &
812                         (d_e(id_s)-d_s(id_s)))
813                     id_s = id_s + 1
814                 END IF
815             END DO
816
817         END DO
818
819         !take a weighted average of all m values, to get a musid parameter value.
820         !weights = soilmusiddata(row_start:row_end,3,1)!weight is column 3.
821         IF (sum(weights) > 1.001) THEN
822             print *, 'error - weights within musids exceed 1'
```

```
823                     END IF
824                     IF (sum(weights) < 0.999) THEN
825                         print *, 'error - weights within musids dont add up to 1'
826                     END IF
827                     musid_area(i) = area_data
828
829                     !deallocate/set variables for reading in next musid
830                     deallocate(ptf_data)
831                     deallocate(dep_data)
832                     deallocate(a_all)
833                     deallocate(b_all)
834                     deallocate(weights)
835                     deallocate(nodata_found)
836
837                     row_start = row_end+1
838
839                 END DO
840
841                 !plot parameter values back in space - using musid ascii grid.
842                 !   where nodata, use average szm value across modelled region.
843                 DO i = 1, nrows_bp
844                     DO j = 1, ncols_bp
845                         e = INT(bp(i,j,6))
846                         IF(e<-9000) THEN
847                             pm_map(i,j) = exp(lnto_map(i,j))/((sum(musid_area))/nmusids)
848                             pm_map(i,j) = pm_map(i,j) * glob_pms(8,n_i)
849                         ELSE
850                             pm_map(i,j) = exp(lnto_map(i,j))/musid_area(e)
851                             pm_map(i,j) = pm_map(i,j) * glob_pms(8,n_i)
852                         END IF
853                     END DO
854                 END DO
855
856
857             case default
858                 print *, "ERROR: invalid pedo-transfer setting."
859                 stop
860
861         end select
862
863         DO i = 1,nrows_bp
864             DO j = 1,ncols_bp
865                 !add in upper and lower caps
866                 IF (pm_map(i,j)<0.0001) THEN
867                     pm_map(i,j)=0.0001
868                 ELSEIF (pm_map(i,j)>15) THEN
869                     pm_map(i,j)=15
870                 END IF
871             END DO
872         END DO
873
874     end subroutine pedotf_SZM
875
876
877
```

```
878
879 end module mpr_SZM
```

C.6 mpr_LnTo.f90

```
!! Module containing subroutines for MPR.f90
!! This module contains all subroutines involving MPR for the LnTo parameter:
!
!   get_BasPred_LnTo    Complete logical list of basin predictors required for LnTo
!   init_gp_LnTo        Initialise global parameters for LnTo
!   pedotf_LnTo         Pedo-transfer function for LnTo

!! Rosie Lane - 31st August 2017

module mpr_LnTo
contains


    ! ************************************************************************************************
    ! subroutine: get_BasPred_LnTo : Complete logical list of basin predictors required for LnTo
    ! ************************************************************************************************
    ! Given the user-specified pedo transfer function selected for LnTo, this subroutine adds to a
    ! logical list of required basin predictors, by specifying which predictors it requires.

    subroutine get_BasPred_LnTo(pedo_tf_LnTo, req_soils)

        implicit none

        ! declare dummy arguments
        integer, intent(inout)                  :: pedo_tf_LnTo !pedo_tf_equation to be used
        logical, dimension(*), intent(inout)  :: req_soils    !soils data required logical

        ! define required BasinPredictors for LnTo
        select case(pedo_tf_LnTo)

            case (0)    !fixed parameter - no basin predictors required

            case (1)    !global parameter - no basin predictors required

            case(2)     !Use pedo-transfer as Cosby et al. (1984)
                !print *, "LnTo transfer function selected is Cosby et al. (1984)"
                req_soils(1) = .true.   !sand
                req_soils(3) = .true.   !clay
                req_soils(4) = .true.   !organic content
                req_soils(15) = .true.  !bulk density
                req_soils(16) = .true.  !is organic? Logical

            case(3)     !Use pedo-transfer as Cosby et al. (1984), but separate areas
                        !with productive geology
                !print *, "LnTo transfer function selected is Cosby et al. (1984)"
                req_soils(1) = .true.   !sand
                req_soils(3) = .true.   !clay
                req_soils(4) = .true.   !organic content
                req_soils(15) = .true.  !bulk density
                req_soils(16) = .true.  !is organic? Logical
                req_soils(18) = .true.  !is highly productive geology?

```

```
53
54              case(4)     !Third transfer function has not yet been defined
55                  print *, "LnTo pedo-transfer 2 has not yet been defined!"
56                 print *, "Select pedo_tf = 0 to treat it as a fixed parameter"
57                  print *, "Select pedo_tf = 1 to treat it as a global parameter"
58                  print *, "Select pedo_tf = 2 to use the pedo-transfer function of Cosby ", &
59                      "et al. (1984)"
60                  stop
61
62              case default
63                  print *, "WARNING: a valid pedo-tf equation for LnTo must be specified in the", &
64                      " control file"
65                  print *, "LnTo will be set to the default of fixed parameter"
66                  print *, "pedo_tf_LnTo = 0, sets LnTo as a fixed parameter"
67                  print *, "pedo_tf_LnTo = 1, sets LnTo as a global parameter"
68                  print *, "pedo_tf_LnTo = 2, sets the LnTo pedo-transfer eq. to Cosby et al."
69
70                  pedo_tf_LnTo = 0
71
72          end select
73
74      end subroutine get_BasPred_LnTo
75
76
77
78
79
80      ! ***************************************************************************************************
81      ! subroutine: init_gp_LnTo : Initialise global parameters for LnTo
82      ! ***************************************************************************************************
83      ! This subroutine does the following:
84      !   1. Defines how many global parameters (n_glob_pms) are needed for the user-specified
85      !           transfer function
86      !       TF 0    : 1 global parameter
87      !       TF 1    : 3 global parameters
88      !   2. Defines min/max ranges for global parameters
89      !   3. Generates list of global parameters using the rand function
90      !       - user specified start_seed can be used to skip to any point in this list
91      !       - global parameter list is of dimensions number of global params by number of param
92      !           files needed
93      !   4. Transforms global parameters to be within set min/max ranges
94
95      subroutine init_gp_LnTo(pedo_tf_LnTo,glob_pms_LnTo, n_pm_maps, start_seed, n_gp_all,pm_range)
96
97          use dta_utility
98
99          implicit none
100
101         !declare dummy variables
102         integer, intent(in)                                     :: pedo_tf_LnTo
103         !global param list
104         double precision, allocatable, dimension(:,:),intent(inout) :: glob_pms_LnTo
105         integer, intent(in)                                     :: n_pm_maps
106         !to start rand num generator
107         integer, intent(in)                                     :: start_seed
```

```
108         !vector of number of global params for all params
109         integer, dimension(7), intent(inout)                        :: n_gp_all
110          !min, fixed, max values for this parameter
111         double precision, dimension(3), intent(in)                  :: pm_range
112
113         !declare local variables
114         integer                                             :: n_glob_pms   !number of global parameters
115         double precision, allocatable, dimension(:)         :: min_gp   !min values for global parameters
116         double precision, allocatable, dimension(:)         :: max_gp
117         real, dimension(:,:),allocatable                    :: rand_nums
118         double precision                                    :: num
119         integer, dimension(12)                              :: seed
120         integer                                             :: i,j, rn_i
121
122
123         !1. Define how many global parameters should exist and min/max ranges,
124         !   based on selected transfer function
125         select case(pedo_tf_LnTo)
126
127             case (0)  !Treat as fixed parameter (for ease this is a global parameter with a 0 range)
128
129                 print *, "LnTo is a fixed parameter, of value ",pm_range(2)
130                 n_glob_pms = 1
131                 allocate(min_gp(1)) !min and max values have length 1, as we have 1 global parameter
132                 allocate(max_gp(1))
133                 min_gp(1) = pm_range(2)     !min and max values set to same fixed parameter value.
134                 max_gp(1) = pm_range(2)
135
136             case (1)    !Treat as global parameter
137
138                 print *, "LnTo is a global parameter, in the range ",pm_range(1)," ",pm_range(3)
139                 n_glob_pms = 1
140                 allocate(min_gp(1)) !min and max values have length 1, as we have 1 global parameter
141                 allocate(max_gp(1))
142                 min_gp(1) = pm_range(1)     !min and max values set from parameter file
143                 max_gp(1) = pm_range(3)
144
145             case(2)     !Use pedo-transfer as Cosby et al. (1984) and HYPRES
146
147                 print *, "LnTo will be applied using the pedo-transfer function of Cosby et", &
148                     " al. (1984) for non-organic soils."
149                 print *, " and HYPRES for organic"
150                 n_glob_pms = 7          !a_const, a_sand, a_clay,
151                 allocate(min_gp(7))
152                 allocate(max_gp(7))
153
154                 !cosby et al. limits
155                 min_gp(1) = -3.5 !-2.5!-3.5 increased to -2.5 following dotty plots
156                 min_gp(2)= 0.006!-0.07
157                 min_gp(3) = -0.02!-0.07
158
159                 max_gp(1) = 0.3!0.5
160                 max_gp(2)= 0.03!0.01
161                 max_gp(3) = -0.0032!0.01
162
```

```
163             !HYPRES limits Ks = a + b(BulkDensity)^2 + c(OrganicMatter)^-1 +
164             !       d(BulkDensity * OrganicMatter)
165             !bounds changed following experiments in excel and matlab
166             min_gp(4) = 1 !3
167             max_gp(4) = 6!11
168
169             min_gp(5) = -1.3!-0.5
170             max_gp(5) = -0.5!-1.3
171
172             min_gp(6) = -0.1
173             max_gp(6) = -0.003
174
175             min_gp(7) = -0.4!-0.8
176             max_gp(7) = 0!-0.24
177
178         case(3)     !Same as case 2 for areas covering low/moderately productive geology,
179                     ! higher values for areas covering high productivity geology.
180
181             print *, "LnTo will be applied using the pedo-transfer function of Cosby et", &
182                 " al. (1984) for non-organic soils."
183             print *, " and HYPRES for organic, with a multiplier for highly productive geology"
184             n_glob_pms = 8          !a_const, a_sand, a_clay,
185             allocate(min_gp(8))
186             allocate(max_gp(8))
187
188             !cosby et al. limits
189             min_gp(1) = -3.5
190             min_gp(2)= 0.006
191             min_gp(3) = -0.02
192
193             max_gp(1) = 0.3
194             max_gp(2)= 0.03
195             max_gp(3) = -0.0032
196
197             !HYPRES limits Ks = a + b(BulkDensity)^2 + c(OrganicMatter)^-1 +
198             !       d(BulkDensity * OrganicMatter)
199             !bounds changed following experiments in excel and matlab
200             min_gp(4) = 1 !3
201             max_gp(4) = 6!11
202             min_gp(5) = -1.3!-0.5
203             max_gp(5) = -0.5!-1.3
204             min_gp(6) = -0.1
205             max_gp(6) = -0.003
206             min_gp(7) = -0.4!-0.8
207             max_gp(7) = 0!-0.24
208
209             !Addition for productive geology limits
210             min_gp(8) = 0 !i.e. no effect.
211             max_gp(8) = 10 !Value chosen looking at optimal parameters MC vs MPR for southeast.
212
213         case default
214             print *, "ERROR: pedo_tf_LnTo incorrectly defined. "
215
216     end select
217
```

```
218         !2. From above, set sizes of the global parameter array and number of random numbers
219         !   to produce.
220         n_gp_all(2) = n_glob_pms
221         allocate(glob_pms_LnTo(n_glob_pms, n_pm_maps))
222         allocate(rand_nums(sum(n_gp_all), n_pm_maps+start_seed-1))
223
224         !3. Generate lists of global parameters between min and max ranges, of length n
225         !need to have list such that starting seed = 50 would give same result as 50th number
226         !from seed=1
227         Call RANDOM_SEED( GET = seed )
228         seed(1:12) = (/ 3,3,3,3,3,3,3,3,3,3,3,3/)
229         Call RANDOM_SEED( PUT = seed )
230         call RANDOM_NUMBER ( rand_nums )
231
232         !now normalise random numbers to within bounds
233         DO i = 1, n_glob_pms
234             DO j = start_seed, start_seed+n_pm_maps-1
235                 !rn_i is index for the random number
236                 rn_i = (sum(n_gp_all) - n_glob_pms) + i
237                 !the appropriate random number, in range 0-1
238                 num = rand_nums(rn_i,j)
239                 !normalise the random number to given min/max
240                 num = (num * (max_gp(i)-min_gp(i)))+min_gp(i)
241                 glob_pms_LnTo(i, j-start_seed+1) = num
242
243             END DO
244             IF (n_pm_maps >= 2) THEN
245                 print *, 'First 2 values for parameter ',i,' : ',glob_pms_LnTo(i,1:2)
246             END IF
247         END DO
248
249
250         print *,
251
252
253
254     end subroutine init_gp_LnTo
255
256
257
258
259
260     ! ************************************************************************************************
261     ! subroutine: pedotf_LnTo: Pedo-transfer function for LnTo
262     ! ************************************************************************************************
263     ! This routine takes n_i, signalling that we are now calculating parameter map n_i out of n,
264     ! It takes the list of global parameters and required basin predictors, and from that
265     ! applies the pedo-transfer functions to produce a parameter map.
266
267     subroutine pedotf_LnTo(n_i, pedo_tf, glob_pms, sand_0_10, clay_0_10, pm_map,bp_maps)
268
269         implicit none
270
271         !declare dummy variables
272         !param map number to do
```

```
273         integer, intent(in)                                         :: n_i
274         !equation selection
275         integer, intent(in)                                         :: pedo_tf
276         !list of global params
277         double precision, dimension(:,:), allocatable, intent(in)   :: glob_pms
278         real, dimension(:,:), allocatable, intent(in)    :: sand_0_10   !input basin predictor maps
279         real, dimension(:,:), allocatable, intent(in)    :: clay_0_10
280         !output parameter map
281         double precision, dimension(:,:), allocatable, intent(out)  :: pm_map
282         !basin predictor maps.(x,y,bp)
283         real, allocatable, dimension(:,:,:), intent(in)             :: bp_maps
284
285         !declare local variables
286         integer             :: n_glob_pms       !number of global parameters
287         integer             :: nrows_bp         !number of rows in basin predictor map
288         integer             :: ncols_bp
289         integer             :: n_pm_maps        !number of parameter maps
290                                                 !defined by ncols in glob_pms_LnTo
291         integer             :: i
292         integer             :: j
293
294         !find number of rows and columns in basin predictor files - pm map should be same size
295         nrows_bp = size(sand_0_10(:,1))
296         ncols_bp = size(sand_0_10(1,:))
297         n_pm_maps = size(glob_pms(1,:))
298         allocate(pm_map(nrows_bp, ncols_bp))
299
300         !vary transfer function, depending on user-inputted equation selection.
301         select case(pedo_tf)
302
303             case (0)    !Fixed parameter
304
305                 n_glob_pms = 1
306                 DO i = 1, nrows_bp
307                     DO j = 1, ncols_bp
308
309                         pm_map(i,j) = glob_pms(1, n_i)
310
311                     END DO
312                 END DO
313
314             case (1)    !Global parameter
315
316                 n_glob_pms = 1
317                 DO i = 1, nrows_bp
318                     DO j = 1, ncols_bp
319
320                         pm_map(i,j) = glob_pms(1, n_i)
321
322                     END DO
323                 END DO
324
325
326             case (2)
327             ! if non-organic soils   !Use pedo-transfer as Cosby et al. (1984),
```

```
328                                 !LnTo = a1 + a2 * %sand + a3 * %clay
329             ! if organic soils  !use HYPRES ptf excluding silt and clay,
330                                 !lnto = a + b(BulkDensity)^2 + c(OrganicMatter)^-1 +
331                                 !       d(BulkDensity * OrganicMatter)
332
333              n_glob_pms = 7
334                 DO i = 1, nrows_bp
335                     DO j = 1, ncols_bp
336
337                         !check if organic
338                         !IF (bp_maps(i,j,16) == 1) THEN
339                         IF (bp_maps(i,j,4)>=35) THEN  !now using organic threshold of more than 35%
340                             !HYPRES function for organic soils
341                             pm_map(i,j) = glob_pms(4,n_i)+(glob_pms(5,n_i)*bp_maps(i,j,15)* &
342                                 bp_maps(i,j,15))
343                             pm_map(i,j) = pm_map(i,j) + (glob_pms(6,n_i)*(1/(bp_maps(i,j,4))))
344                             pm_map(i,j) = pm_map(i,j) + (glob_pms(7,n_i)*bp_maps(i,j,15)* &
345                                 bp_maps(i,j,4))
346                         ELSE
347                             !cosby et al. function for mineral soils
348                             pm_map(i,j) = glob_pms(1, n_i) + (glob_pms(2,n_i)*sand_0_10(i,j))
349                             pm_map(i,j) = pm_map(i,j) + (glob_pms(3,n_i)*clay_0_10(i,j))
350                         END IF
351
352                     END DO
353                 END DO
354
355             case (3)
356             ! if non-organic soils   !Use pedo-transfer as Cosby et al. (1984),
357             ! LnTo = a1 + a2 * %sand + a3 * %clay
358             ! if organic soils       !use HYPRES ptf excluding silt and clay,
359             ! lnto = a + b(BulkDensity)^2 + c(OrganicMatter)^-1 + d(BulkDensity * OrganicMatter)
360             ! if highly productive hydrogeology !add an extra global parameter.
361              n_glob_pms = 8
362                 DO i = 1, nrows_bp
363                     DO j = 1, ncols_bp
364
365                         !check if organic
366                         IF (bp_maps(i,j,4)>=35) THEN  !now using organic threshold of more than 35%
367                             !HYPRES function for organic soils
368                             pm_map(i,j) = glob_pms(4,n_i)+(glob_pms(5,n_i)*bp_maps(i,j,15)* &
369                                 bp_maps(i,j,15))
370                             pm_map(i,j) = pm_map(i,j) + (glob_pms(6,n_i)*(1/(bp_maps(i,j,4))))
371                             pm_map(i,j) = pm_map(i,j) + (glob_pms(7,n_i)*bp_maps(i,j,15)* &
372                                 bp_maps(i,j,4))
373                         ELSE
374                             !cosby et al. function for mineral soils
375                             pm_map(i,j) = glob_pms(1, n_i) + (glob_pms(2,n_i)*sand_0_10(i,j))
376                             pm_map(i,j) = pm_map(i,j) + (glob_pms(3,n_i)*clay_0_10(i,j))
377                         END IF
378
379                         !add extra global parameter if high productivity hydrogeology
380                         IF (bp_maps(i,j,18) == 1) THEN
381                             pm_map(i,j) = pm_map(i,j) + glob_pms(8,n_i)
382                         END IF
```

```
383                     END DO
384                 END DO
385
386             case default
387
388                 print *, "ERROR: invalid pedo-transfer setting."
389                 stop
390
391
392         end select
393
394         DO i = 1,nrows_bp
395             DO j = 1,ncols_bp
396                 !add in upper and lower caps
397                 IF (pm_map(i,j)<-15) THEN
398                     pm_map(i,j)=-15
399                 ELSEIF (pm_map(i,j)>10) THEN
400                     pm_map(i,j)=10
401                 END IF
402             END DO
403         END DO
404
405     end subroutine pedotf_LnTo
406
407
408
409
410 end module mpr_LnTo
```

C.7 mpr_SRinit.f90

```
!! Module containing subroutines for MPR.f90
!! This module contains all subroutines involving MPR for the SRinit parameter:
!
!   get_BasPred_SRinit    Complete logical list of basin predictors required for SRinit
!   init_gp_SRinit        Initialise global parameters for SRinit
!   pedotf_SRinit         Pedo-transfer function for SRinit

!! Rosie Lane - 31st August 2017

module mpr_SRinit
contains


    ! ************************************************************************************************
    ! subroutine: get_BasPred_SRinit : Complete logical list of basin predictors required for SRinit
    ! ************************************************************************************************
    ! Given the user-specified pedo transfer function selected for SRinit, this subroutine adds to a
    ! logical list of required basin predictors, by specifying which predictors it requires.

    subroutine get_BasPred_SRinit(pedo_tf_SRinit, req_soils)

        implicit none

        ! declare dummy arguments
        integer, intent(inout)                 :: pedo_tf_SRinit !pedo_tf_equation to be used
        logical, dimension(*), intent(inout)   :: req_soils    !soils data required logical

        ! define required BasinPredictors for SRinit
        select case(pedo_tf_SRinit)

            case (0)    !fixed parameter - no basin predictors required

            case (1)    !global parameter - no basin predictors required

            case (2)     !Use a pedo-transfer equation
                print *, "ERROR: A transfer function has not yet been programmed for SRinit"
                print *, "Select pedo_tf = 0 to treat it as a fixed parameter"
                print *, "Select pedo_tf = 1 to treat it as a global parameter"
                stop

            case default
                print *, "WARNING: a valid pedo-tf equation for SRinit must be specified in ", &
                    "the control file"
                print *, "SRinit will be set to the default of fixed parameter"
                print *, "The following options can be selected in the control file:"
                print *, "pedo_tf_SRinit = 0, sets it as a fixed parameter"
                print *, "pedo_tf_SRinit = 1, sets it as a global parameter"
                print *,

                pedo_tf_SRinit = 0

        end select
```

```
53
54      end subroutine get_BasPred_SRinit
55
56
57
58
59
60      ! ****************************************************************************************************
61      ! subroutine: init_gp_SRinit : Initialise global parameters for SRinit
62      ! ****************************************************************************************************
63      ! This subroutine does the following:
64      !   1. Defines how many global parameters (n_glob_pms) are needed for the user-specified
65      !          transfer function
66      !       TF 0    : 1 global parameter
67      !       TF 1    : 3 global parameters
68      !   2. Defines min/max ranges for global parameters
69      !   3. Generates list of global parameters using the rand function
70      !       - user specified start_seed can be used to skip to any point in this list
71      !       - global parameter list is of dimensions number of global params by number of param
72      !          files needed
73      !   4. Transforms global parameters to be within set min/max ranges
74
75      subroutine init_gp_SRinit(pedo_tf_SRinit,glob_pms_SRinit, n_pm_maps, start_seed, &
76          n_gp_all,pm_range)
77
78          use dta_utility
79
80          implicit none
81
82          !declare dummy variables
83          integer, intent(in)                                         :: pedo_tf_SRinit
84          !global param list
85          double precision, allocatable, dimension(:,:),intent(inout) :: glob_pms_SRinit
86          integer, intent(in)                                         :: n_pm_maps
87          !to start rand num generator
88          integer, intent(in)                                         :: start_seed
89          !vector of number of global params for all params
90          integer, dimension(7), intent(inout)                        :: n_gp_all
91          !min, fixed, max values for this parameter
92          double precision, dimension(3), intent(in)                  :: pm_range
93
94          !declare local variables
95          integer                                     :: n_glob_pms   !number of global parameters
96          double precision, allocatable, dimension(:) :: min_gp   !min values for global parameters
97          double precision, allocatable, dimension(:) :: max_gp
98          real, dimension(:,:),allocatable            :: rand_nums
99          double precision                            :: num
100         integer, dimension(12)                      :: seed
101         integer                                     :: i,j, rn_i
102
103
104         !1. Define how many global parameters should exist and min/max ranges, based on
105         !selected transfer function
106         select case(pedo_tf_SRinit)
107
```

```
108             case (0)  !Treat as fixed parameter (for ease this is a global parameter with a 0 range)
109
110                 print *, "SRinit is a fixed parameter, of value ",pm_range(2)
111                 n_glob_pms = 1
112                 allocate(min_gp(1)) !min and max values have length 1, as we have 1 global parameter
113                 allocate(max_gp(1))
114                 min_gp(1) = pm_range(2)     !min and max values set to same fixed parameter value.
115                 max_gp(1) = pm_range(2)
116
117             case (1)    !Treat as global parameter
118
119                 print *, "SRinit is a global parameter, in the range ",pm_range(1)," ",pm_range(3)
120                 n_glob_pms = 1
121                 allocate(min_gp(1)) !min and max values have length 1, as we have 1 global parameter
122                 allocate(max_gp(1))
123                 min_gp(1) = pm_range(1)     !min and max values set from parameter file
124                 max_gp(1) = pm_range(3)
125
126             case default
127                 print *, "ERROR: pedo_tf_SRinit incorrectly defined. "
128
129         end select
130
131         !2. From above, set sizes of the global parameter array and number of random numbers
132         !   to produce.
133         n_gp_all(4) = n_glob_pms
134
135         allocate(glob_pms_SRinit(n_glob_pms, n_pm_maps))
136         allocate(rand_nums(sum(n_gp_all), n_pm_maps+start_seed-1))
137
138         !generate lists of global parameters between min and max ranges, of length n
139         !need to have list such that starting seed = 50 would give same result as 50th number
140         !from seed=1
141         Call RANDOM_SEED( GET = seed )
142         seed(1:12) = (/ 3,3,3,3,3,3,3,3,3,3,3,3/)
143         Call RANDOM_SEED( PUT = seed )
144         call RANDOM_NUMBER ( rand_nums )
145
146
147         !now normalise random numbers to within bounds
148
149         DO i = 1,n_glob_pms
150             DO j = start_seed, start_seed+n_pm_maps-1
151                 !rn_i is index for the random number
152                 rn_i = (sum(n_gp_all) - n_glob_pms) + i
153                 !the appropriate random number, in range 0-1
154                 num = rand_nums(rn_i,j)
155                 !normalise the random number to given min/max
156                 num = (num * (max_gp(i)-min_gp(i)))+min_gp(i)
157                 glob_pms_SRinit(i, j-start_seed+1) = num
158
159             END DO
160             IF (n_pm_maps >= 2) THEN
161                 print *, 'First 2 values for parameter ',i,' : ',glob_pms_SRinit(:,1:2)
162             END IF
```

```
163         END DO
164
165
166         print *,
167
168     end subroutine init_gp_SRinit
169
170
171
172
173
174     ! ****************************************************************************************
175     ! subroutine: pedotf_SRinit: Pedo-transfer function for SRinit
176     ! ****************************************************************************************
177     ! This routine takes n_i, signalling that we are now calculating parameter map n_i out of n,
178     ! It takes the list of global parameters and required basin predictors, and from that
179     ! applies the pedo-transfer functions to produce a parameter map.
180
181     subroutine pedotf_SRinit(n_i, pedo_tf, glob_pms,bp, pm_map)
182
183         implicit none
184
185         !declare dummy variables
186         !param map number to do
187         integer, intent(in)                                     :: n_i
188         !equation selection
189         integer, intent(in)                                     :: pedo_tf
190         !list of global params
191         double precision, dimension(:,:), allocatable, intent(in)    :: glob_pms
192         !basin predictor map (for sizing)
193         real, dimension(:,:), allocatable, intent(in)     :: bp
194         !output parameter map
195         double precision, dimension(:,:), allocatable, intent(out)   :: pm_map
196
197         !declare local variables
198         integer             :: n_glob_pms       !number of global parameters
199         integer             :: n_pm_maps        !number of parameter maps
200                                                 !defined by ncols in glob_pms_SRinit
201         integer             :: i
202         integer             :: j
203         integer             :: nrows_bp         !number of rows in basin predictor map
204         integer             :: ncols_bp
205
206
207         !find number of rows and columns in basin predictor files - pm map should be same size
208         nrows_bp = size(bp(:,1))
209         ncols_bp = size(bp(1,:))
210         n_pm_maps = size(glob_pms(1,:))
211         !find number of rows and columns in basin predictor files - pm map should be same size
212         n_pm_maps = size(glob_pms(1,:))
213         allocate(pm_map(nrows_bp, ncols_bp))
214
215         !vary transfer function, depending on user-inputted equation selection.
216         select case(pedo_tf)
217
```

```
218             case (0)    !Fixed parameter
219 
220                 n_glob_pms = 1
221                 DO i = 1, nrows_bp
222                     DO j = 1, ncols_bp
223 
224                         pm_map(i,j) = glob_pms(1, n_i)
225 
226                     END DO
227                 END DO
228 
229             case (1)    !Global parameter
230 
231                 n_glob_pms = 1
232                 DO i = 1, nrows_bp
233                     DO j = 1, ncols_bp
234 
235                         pm_map(i,j) = glob_pms(1, n_i)
236 
237                     END DO
238                 END DO
239 
240             case default
241                 print *, "ERROR: invalid pedo-transfer setting."
242                 stop
243 
244         end select
245 
246     end subroutine pedotf_SRinit
247 
248 
249 
250 
251 end module mpr_SRinit
```

C.8 mpr_Td.f90

```
!! Module containing subroutines for MPR.f90
!! This module contains all subroutines involving MPR for the Td parameter:
!
!   get_BasPred_Td    Complete logical list of basin predictors required for Td
!   init_gp_Td        Initialise global parameters for Td
!   pedotf_Td         Pedo-transfer function for Td

!! Rosie Lane - 31st August 2017

module mpr_Td
contains


    ! ***********************************************************************************************
    ! subroutine: get_BasPred_Td : Complete logical list of basin predictors required for Td
    ! ***********************************************************************************************
    ! Given the user-specified pedo transfer function selected for Td, this subroutine adds to a
    ! logical list of required basin predictors, by specifying which predictors it requires.

    subroutine get_BasPred_Td(pedo_tf_Td, req_soils)

        implicit none

        ! declare dummy arguments
        integer, intent(inout)                  :: pedo_tf_Td !pedo_tf_equation to be used
        logical, dimension(*), intent(inout)  :: req_soils    !soils data required logical

        ! define required BasinPredictors for Td
        select case(pedo_tf_Td)

            case (0)    !fixed parameter - no basin predictors required

            case (1)    !global parameter - no basin predictors required

            case (2)      !multiplier on ln(T0)
                !print *, "Td transfer function selected is Cosby et al. (1984)"
                !req_soils(1) = .true.   !sand
                !req_soils(3) = .true.   !clay
                !req_soils(4) = .true.   !organic content
                !req_soils(15) = .true.  !bulk density
                !req_soils(16) = .true.  !is organic? Logical

            case default
                print *, "WARNING: a valid pedo-tf equation for Td must be specified in the "", &
                    control file"
                print *, "Td will be set to the default of fixed parameter"
                print *, "The following options can be selected in the control file:"
                print *, "pedo_tf_Td = 0, sets it as a fixed parameter"
                print *, "pedo_tf_Td = 1, sets it as a global parameter"
                print *,

                pedo_tf_Td = 0
```

```
53
54          end select
55
56      end subroutine get_BasPred_Td
57
58
59
60
61
62      ! ********************************************************************************************
63      ! subroutine: init_gp_Td : Initialise global parameters for Td
64      ! ********************************************************************************************
65      ! This subroutine does the following:
66      !   1. Defines how many global parameters (n_glob_pms) are needed for the user-specified
67      !           transfer function
68      !       TF 0    : 1 global parameter
69      !       TF 1    : 3 global parameters
70      !   2. Defines min/max ranges for global parameters
71      !   3. Generates list of global parameters using the rand function
72      !       - user specified start_seed can be used to skip to any point in this list
73      !       - global parameter list is of dimensions number of global params by number of param
74      !           files needed
75      !   4. Transforms global parameters to be within set min/max ranges
76
77      subroutine init_gp_Td(pedo_tf_Td,glob_pms_Td, n_pm_maps, start_seed, n_gp_all,pm_range)
78
79          use dta_utility
80
81          implicit none
82
83          !declare dummy variables
84          integer, intent(in)                                         :: pedo_tf_Td
85          !global param list
86          double precision, allocatable, dimension(:,:),intent(inout) :: glob_pms_Td
87          integer, intent(in)                                         :: n_pm_maps
88          !to start rand num generator
89          integer, intent(in)                                         :: start_seed
90          !vector of number of global params for all params
91          integer, dimension(7), intent(inout)                        :: n_gp_all
92          !min, fixed, max values for this parameter
93          double precision, dimension(3), intent(in)                  :: pm_range
94
95          !declare local variables
96          integer                                     :: n_glob_pms   !number of global parameters
97          double precision, allocatable, dimension(:) :: min_gp   !min values for global parameters
98          double precision, allocatable, dimension(:) :: max_gp
99          real, dimension(:,:),allocatable            :: rand_nums
100         double precision                            :: num
101         integer, dimension(12)                      :: seed
102         integer                                     :: i,j, rn_i
103
104
105         !1. Define how many global parameters should exist and min/max ranges, based on selected
106         !   transfer function
107         select case(pedo_tf_Td)
```

```
108
109             case (0) !Treat as fixed parameter (for ease this is a global parameter with a 0 range)
110
111                 print *, "Td is a fixed parameter, of value ",pm_range(2)
112                 n_glob_pms = 1
113                 allocate(min_gp(1)) !min and max values have length 1, as we have 1 global parameter
114                 allocate(max_gp(1))
115                 min_gp(1) = pm_range(2)    !min and max values set to same fixed parameter value.
116                 max_gp(1) = pm_range(2)
117
118             case (1)    !Treat as global parameter
119
120                 print *, "Td is a global parameter, in the range ",pm_range(1)," ",pm_range(3)
121                 n_glob_pms = 1
122                 allocate(min_gp(1)) !min and max values have length 1, as we have 1 global parameter
123                 allocate(max_gp(1))
124                 min_gp(1) = pm_range(1)    !min and max values set from parameter file
125                 max_gp(1) = pm_range(3)
126
127             case (2)   !multiplier on ln(T0)
128
129
130                 n_glob_pms = 1
131                 allocate(min_gp(1)) !min and max values have length 1, as we have 1 global parameter
132                 allocate(max_gp(1))
133                 min_gp(1) = 1   !min and max values set from parameter file
134                 max_gp(1) = 10
135                 print *, "Td is a multiplier on ln(T0)."
136
137
138             case default
139                 print *, "ERROR: pedo_tf_Td incorrectly defined. "
140
141         end select
142
143         !2. From above, set sizes of the global parameter array and number of random numbers
144         !   to produce.
145         n_gp_all(6) = n_glob_pms
146
147         allocate(glob_pms_Td(n_glob_pms, n_pm_maps))
148         allocate(rand_nums(sum(n_gp_all), n_pm_maps+start_seed-1))
149
150         !generate lists of global parameters between min and max ranges, of length n
151         !need to have list such that starting seed = 50 would give same result as 50th number
152         !from seed=1
153         Call RANDOM_SEED( GET = seed )
154         seed(1:12) = (/ 3,3,3,3,3,3,3,3,3,3,3,3/)
155         Call RANDOM_SEED( PUT = seed )
156         call RANDOM_NUMBER ( rand_nums )
157
158
159
160         !now normalise random numbers to within bounds
161
162         DO i = 1,n_glob_pms
```

```
163             DO j = start_seed, start_seed+n_pm_maps-1
164
165                 !rn_i is index for the random number
166                 rn_i = (sum(n_gp_all) - n_glob_pms) + i
167                 !the appropriate random number, in range 0-1
168                 num = rand_nums(rn_i,j)
169                 !normalise the random number to given min/max
170                 num = (num * (max_gp(i)-min_gp(i)))+min_gp(i)
171                 glob_pms_Td(i, j-start_seed+1) = num
172
173             END DO
174             IF (n_pm_maps >= 2) THEN
175                 print *, 'First 2 values for parameter ',i,' : ',glob_pms_Td(:,1:2)
176             END IF
177         END DO
178
179
180         print *,
181
182     end subroutine init_gp_Td
183
184
185
186
187
188     ! ****************************************************************************************
189     ! subroutine: pedotf_LnTo: Pedo-transfer function for LnTo
190     ! ****************************************************************************************
191     ! This routine takes n_i, signalling that we are now calculating parameter map n_i out of n,
192     ! It takes the list of global parameters and required basin predictors, and from that
193     ! applies the pedo-transfer functions to produce a parameter map.
194
195     subroutine pedotf_Td(n_i, pedo_tf, glob_pms,bp, pm_map,lnto_map)
196
197         implicit none
198
199         !declare dummy variables
200         !param map number to do
201         integer, intent(in)                                         :: n_i
202         !equation selection
203         integer, intent(in)                                         :: pedo_tf
204         !list of global params
205         double precision, dimension(:,:), allocatable, intent(in)   :: glob_pms
206         !basin predictor map (for sizing)
207         real, dimension(:,:), allocatable, intent(in)    :: bp
208         !output parameter map
209         double precision, dimension(:,:), allocatable, intent(out)  :: pm_map
210          !map for lnto parameter
211         double precision, dimension(:,:), allocatable, intent(in)  :: lnto_map
212
213         !declare local variables
214         integer             :: n_glob_pms       !number of global parameters
215         integer             :: n_pm_maps        !number of parameter maps
216                                                 !defined by ncols in glob_pms_Td
217         integer             :: i
```

```
218         integer             :: j
219         integer             :: nrows_bp         !number of rows in basin predictor map
220         integer             :: ncols_bp
221
222
223         !find number of rows and columns in basin predictor files - pm map should be same size
224         nrows_bp = size(bp(:,1))
225         ncols_bp = size(bp(1,:))
226         n_pm_maps = size(glob_pms(1,:))
227         !find number of rows and columns in basin predictor files - pm map should be same size
228         n_pm_maps = size(glob_pms(1,:))
229         allocate(pm_map(nrows_bp, ncols_bp))
230
231         !vary transfer function, depending on user-inputted equation selection.
232         select case(pedo_tf)
233
234             case (0)    !Fixed parameter
235
236                 n_glob_pms = 1
237                 DO i = 1, nrows_bp
238                     DO j = 1, ncols_bp
239
240                         pm_map(i,j) = glob_pms(1, n_i)
241
242                     END DO
243                 END DO
244
245             case (1)    !Global parameter
246
247                 n_glob_pms = 1
248                 DO i = 1, nrows_bp
249                     DO j = 1, ncols_bp
250
251                         pm_map(i,j) = glob_pms(1, n_i)
252
253                     END DO
254                 END DO
255
256             case (2)   !multiplier on ln(T0): Td = a*exp(lnto)
257
258                 n_glob_pms = 1
259                 DO i = 1, nrows_bp
260                     DO j = 1, ncols_bp
261
262                         !pm_map(i,j) = glob_pms(1, n_i)*(lnto_map(i,j)+20)
263                         pm_map(i,j) = glob_pms(1,n_i)*EXP(lnto_map(i,j))
264
265                     END DO
266                 END DO
267
268
269
270             case default
271                 print *, "ERROR: invalid pedo-transfer setting."
272                 stop
```

```
273
274         end select
275
276          ! set a limit - to stop crazy values of 6x10^25 causing crashes!
277         DO i = 1,nrows_bp
278             DO j = 1,ncols_bp
279                 !add in upper and lower caps
280                 IF (pm_map(i,j)<0.01) THEN
281                     pm_map(i,j)=0.01
282                 ELSEIF (pm_map(i,j)>200) THEN
283                     pm_map(i,j)=200
284                 END IF
285             END DO
286         END DO
287
288
289     end subroutine pedotf_Td
290
291
292
293
294 end module mpr_Td
```

C.9 mpr_CHV.f90

```
!! Module containing subroutines for MPR.f90
!! This module contains all subroutines involving MPR for the CHV parameter:
!
!   get_BasPred_CHV    Complete logical list of basin predictors required for CHV
!   init_gp_CHV        Initialise global parameters for CHV
!   pedotf_CHV         Pedo-transfer function for CHV

!! Rosie Lane - 31st August 2017

module mpr_CHV
contains


    ! ***************************************************************************************************
    ! subroutine: get_BasPred_CHV : Complete logical list of basin predictors required for CHV
    ! ***************************************************************************************************
    ! Given the user-specified pedo transfer function selected for CHV, this subroutine adds to a
    ! logical list of required basin predictors, by specifying which predictors it requires.
    subroutine get_BasPred_CHV(pedo_tf_CHV, req_soils)

        implicit none

        ! declare dummy arguments
        integer, intent(inout)                :: pedo_tf_CHV !pedo_tf_equation to be used
        logical, dimension(*), intent(inout)  :: req_soils    !soils data required logical

        ! define required BasinPredictors for CHV
        select case(pedo_tf_CHV)

            case (0)    !fixed parameter - no basin predictors required

            case (1)    !global parameter - no basin predictors required

            case (2)     !Use a pedo-transfer equation
                print *, "ERROR: A transfer function has not yet been programmed for CHV"
                print *, "Select pedo_tf = 0 to treat it as a fixed parameter"
                print *, "Select pedo_tf = 1 to treat it as a global parameter"

            case default
                print *, "WARNING: a valid pedo-tf equation for CHV must be specified in the ", &
                    "control file"
                print *, "CHV will be set to the default of fixed parameter"
                print *, "The following options can be selected in the control file:"
                print *, "pedo_tf_CHV = 0, sets it as a fixed parameter"
                print *, "pedo_tf_CHV = 1, sets it as a global parameter"
                print *,

                pedo_tf_CHV = 0

        end select

    end subroutine get_BasPred_CHV
```

```
53
54      ! ********************************************************************************************
55      ! subroutine: init_gp_CHV : Initialise global parameters for CHV
56      ! ********************************************************************************************
57      ! This subroutine does the following:
58      !   1. Defines how many global parameters (n_glob_pms) are needed
59      !       TF 0    : 0 global parameter
60      !       TF 1    : 1 global parameters
61      !   2. Defines min/max ranges for global parameters
62      !   3. Generates list of global parameters using the rand function
63      !       - user specified start_seed can be used to skip to any point in this list
64      !   4. Transforms global parameters to be within set min/max ranges
65      subroutine init_gp_CHV(pedo_tf_CHV,glob_pms_CHV, n_pm_maps, start_seed, n_gp_all,pm_range)
66
67          use dta_utility
68
69          implicit none
70
71          !declare dummy variables
72          !( start_seed is given to start rand num generator)
73          !( n_gp_all is the vector of number of global params for all params)
74          !( pm_range is the min, fixed, max values for this parameter)
75          integer, intent(in)                                     :: pedo_tf_CHV
76          double precision, allocatable, dimension(:,:),intent(inout) :: glob_pms_CHV
77          integer, intent(in)                                     :: n_pm_maps
78          integer, intent(in)                                     :: start_seed
79          integer, dimension(7), intent(inout)                    :: n_gp_all
80          double precision, dimension(3), intent(in)              :: pm_range
81
82          !declare local variables
83          !( n_glob_pms is the number of global parameters)
84          !( min_gp/ max_gp are the min/max values for the global parameters)
85          integer                                         :: n_glob_pms
86          double precision, allocatable, dimension(:)     :: min_gp
87          double precision, allocatable, dimension(:)     :: max_gp
88          real, dimension(:,:),allocatable                :: rand_nums
89          double precision                                :: num
90          integer, dimension(12)                          :: seed
91          integer                                         :: i,j, rn_i
92
93
94          !1. Define how many global parameters should exist and min/max ranges,
95          !based on selected transfer function
96          select case(pedo_tf_CHV)
97
98              case (0)    !Treat as fixed parameter
99
100                 print *, "CHV is a fixed parameter, of value ",pm_range(2)
101                 n_glob_pms = 1
102                 allocate(min_gp(1))
103                 allocate(max_gp(1))
104                 min_gp(1) = pm_range(2)     !min and max values set to same fixed parameter value.
105                 max_gp(1) = pm_range(2)
106
107             case (1)    !Treat as global parameter
```

```
108
109                 print *, "CHV is a global parameter, in the range ",pm_range(1)," ",pm_range(3)
110                 n_glob_pms = 1
111                 allocate(min_gp(1))
112                 allocate(max_gp(1))
113                 min_gp(1) = pm_range(1)
114                 max_gp(1) = pm_range(3)
115
116             case default
117                 print *, "ERROR: pedo_tf_CHV incorrectly defined. "
118
119         end select
120
121         !2. From above, set sizes of the global parameter array
122         ! and number of random numbers to produce.
123         n_gp_all(5) = n_glob_pms
124
125         allocate(glob_pms_CHV(n_glob_pms, n_pm_maps))
126         allocate(rand_nums(sum(n_gp_all), n_pm_maps+start_seed-1))
127
128         !generate lists of global parameters between min and max ranges, of length n
129         !need to have list such that starting seed = 50 would give same result as
130         !50th number from seed=1
131         Call RANDOM_SEED( GET = seed )
132         seed(1:12) = (/ 3,3,3,3,3,3,3,3,3,3,3,3/)
133         Call RANDOM_SEED( PUT = seed )
134         call RANDOM_NUMBER ( rand_nums )
135
136
137         !now normalise random numbers to within bounds
138         !rn_i is index for the random number
139         DO i = 1,n_glob_pms
140             DO j = start_seed, start_seed+n_pm_maps-1
141
142                 rn_i = (sum(n_gp_all) - n_glob_pms )+ i
143                 num = rand_nums(rn_i,j)
144                 num = (num * (max_gp(i)-min_gp(i)))+min_gp(i)
145                 glob_pms_CHV(i, j-start_seed+1) = num
146
147             END DO
148             IF (n_pm_maps >= 2) THEN
149                 print *, 'First 2 values for parameter ',i,' : ',glob_pms_CHV(i,1:2)
150             END IF
151         END DO
152
153         print *,
154
155     end subroutine init_gp_CHV
156
157
158     ! ***********************************************************************************************
159     ! subroutine: pedotf_CHV: Pedo-transfer function for CHV
160     ! ***********************************************************************************************
161     ! This routine takes n_i, signalling that we are now calculating parameter map n_i out of n,
162     ! It takes the list of global parameters and required basin predictors, and from that
```

```
163     ! applies the pedo-transfer functions to produce a parameter map.
164     subroutine pedotf_CHV(n_i, pedo_tf, glob_pms,bp, pm_map)
165
166         implicit none
167
168         !declare dummy variables
169         !n_i  is the param map number to do
170         !pedo_tf is the equation number selected
171         !glob_pms is the list of global params, bp is the basin predictor map
172         integer, intent(in)                                         :: n_i
173         integer, intent(in)                                         :: pedo_tf
174         double precision, dimension(:,:), allocatable, intent(in)   :: glob_pms
175         real, dimension(:,:), allocatable, intent(in)               :: bp
176         double precision, dimension(:,:), allocatable, intent(out)  :: pm_map
177
178         !declare local variables
179         integer             :: n_glob_pms       !number of global parameters
180         integer             :: n_pm_maps        !number of parameter maps
181         integer             :: i
182         integer             :: j
183         integer             :: nrows_bp         !number of rows in basin predictor map
184         integer             :: ncols_bp
185
186         !find number of rows and columns in basin predictor files
187         nrows_bp = size(bp(:,1))
188         ncols_bp = size(bp(1,:))
189         n_pm_maps = size(glob_pms(1,:))
190
191         !find number of rows and columns in basin predictor files - pm map should be same size
192         n_pm_maps = size(glob_pms(1,:))
193         allocate(pm_map(nrows_bp, ncols_bp))
194
195         !vary transfer function, depending on user-inputted equation selection.
196         select case(pedo_tf)
197
198             case (0)    !Fixed parameter
199                 n_glob_pms = 1
200                 DO i = 1, nrows_bp
201                     DO j = 1, ncols_bp
202
203                         pm_map(i,j) = glob_pms(1, n_i)
204
205                     END DO
206                 END DO
207
208             case (1)    !Global parameter
209                 n_glob_pms = 1
210                 DO i = 1, nrows_bp
211                     DO j = 1, ncols_bp
212
213                         pm_map(i,j) = glob_pms(1, n_i)
214
215                     END DO
216                 END DO
217
```

```
218             case default
219                 print *, "ERROR: invalid pedo-transfer setting."
220 
221         end select
222 
223     end subroutine pedotf_CHV
224 
225 
226 end module mpr_CHV
```

C.10 mpr_SRmax.f90

```
!! Module containing subroutines for MPR.f90
!! This module contains all subroutines involving MPR for the SRmax parameter:
!
!   get_BasPred_SRmax    Complete logical list of basin predictors required for SRmax
!   init_gp_SRmax        Initialise global parameters for SRmax
!   pedotf_SRmax         Pedo-transfer function for SRmax

!! Rosie Lane - 31st August 2017

module mpr_SRmax
contains


    ! ************************************************************************************************
    ! subroutine: get_BasPred_SRmax  : Complete logical list of basin predictors required for SRmax
    ! ************************************************************************************************
    ! Given the user-specified pedo transfer function selected for SRmax , this subroutine adds to a
    ! logical list of required basin predictors, by specifying which predictors it requires.

    subroutine get_BasPred_SRmax(pedo_tf_SRmax, req_soils)

        implicit none

        ! declare dummy arguments
        integer, intent(inout)                    :: pedo_tf_SRmax !pedo_tf_equation to be used
        logical, dimension(*), intent(inout)  :: req_soils    !soils data required logical

        ! define required BasinPredictors for SRmax
        select case(pedo_tf_SRmax)

            case (0)    !fixed parameter - no basin predictors required

            case (1)    !global parameter - no basin predictors required

            case (2)     !ptf based on landcover map and rooting depth conversion table

                req_soils(12:13) = (/.true.,.true./) !landcover map and landcover ranges table
                req_soils(17) = .true.  !porosity

            case (3)
                print *, "ERROR: A transfer function has not yet been programmed for SRmax"
                print *, "Select pedo_tf = 0 to treat it as a fixed parameter"
                print *, "Select pedo_tf = 1 to treat it as a global parameter"
                print *, "Select pedo_tf = 2 for ptf based on landcover and rooting depth"
                stop

            case default
                print *, "WARNING: a valid pedo-tf equation for SRmax must be specified in the", &
                    " control file"
                print *, "SRmax will be set to the default of fixed parameter"
                print *, "The following options can be selected in the control file:"
                print *, "pedo_tf_SRmax = 0, sets it as a fixed parameter"
```

```
53              print *, "pedo_tf_SRmax = 1, sets it as a global parameter"
54              print *, "pedo_tf_SRmax = 2, ptf based on landcover and rooting depth"
55              print *,
56
57              pedo_tf_SRmax = 0
58
59          end select
60
61      end subroutine get_BasPred_SRmax
62
63
64
65
66
67      ! ************************************************************************************************
68      ! subroutine: init_gp_SRmax : Initialise global parameters for SRmax
69      ! ************************************************************************************************
70      ! This subroutine does the following:
71      !   1. Defines how many global parameters (n_glob_pms) are needed for the user-specified
72      !           transfer function
73      !       TF 0    : 1 global parameter
74      !       TF 1    : 3 global parameters
75      !   2. Defines min/max ranges for global parameters
76      !   3. Generates list of global parameters using the rand function
77      !       - user specified start_seed can be used to skip to any point in this list
78      !       - global parameter list is of dimensions number of global params by number of param
79      !           files needed
80      !   4. Transforms global parameters to be within set min/max ranges
81
82      subroutine init_gp_SRmax(pedo_tf_SRmax,glob_pms_SRmax,n_pm_maps,start_seed,n_gp_all,pm_range, &
83          bp_root_depths)
84
85          use dta_utility
86
87          implicit none
88
89          !declare dummy variables
90          integer, intent(in)                                         :: pedo_tf_SRmax
91          !global param list
92          double precision, allocatable, dimension(:,:),intent(inout) :: glob_pms_SRmax
93          integer, intent(in)                                         :: n_pm_maps
94          !to start rand num generator
95          integer, intent(in)                                         :: start_seed
96          !vector of number of global params for all params
97          integer, dimension(7), intent(inout)                        :: n_gp_all
98          !min, fixed, max values for this parameter
99          double precision, dimension(3), intent(in)                  :: pm_range
100         real, allocatable,dimension(:,:),intent(in)     :: bp_root_depths
101
102         !declare local variables
103         integer                                     :: n_glob_pms   !number of global parameters
104         double precision, allocatable, dimension(:) :: min_gp   !min values for global parameters
105         double precision, allocatable, dimension(:) :: max_gp
106         real, dimension(:,:),allocatable            :: rand_nums
107         double precision                            :: num
```

```
108         integer, dimension(12)                          :: seed
109         integer                                         :: i,j, rn_i
110
111
112         !1. Define how many global parameters should exist and min/max ranges, based on selected
113         !   transfer function
114         select case(pedo_tf_SRmax)
115
116             case (0) !Treat as fixed parameter (for ease this is a global parameter with a 0 range)
117
118                 print *, "SRmax is a fixed parameter, of value ",pm_range(2)
119                 n_glob_pms = 1
120                 allocate(min_gp(1)) !min and max values have length 1, as we have 1 global parameter
121                 allocate(max_gp(1))
122                 min_gp(1) = pm_range(2)     !min and max values set to same fixed parameter value.
123                 max_gp(1) = pm_range(2)
124
125             case (1)    !Treat as global parameter
126
127                 print *, "SRmax is a global parameter, in the range ",pm_range(1)," ",pm_range(3)
128                 n_glob_pms = 1
129                 allocate(min_gp(1)) !min and max values have length 1, as we have 1 global parameter
130                 allocate(max_gp(1))
131                 min_gp(1) = pm_range(1)     !min and max values set from parameter file
132                 max_gp(1) = pm_range(3)
133
134             case (2)    !Pedotransfer function!!!
135
136                 print *, "SRmax will use a pedotransfer function based on rooting depths and landuse"
137                 print *, "SRmax is a if LU=1, b if LU=2, c if LU=3 etc, *d "
138
139                 !allocate number of global parameters
140
141                 n_glob_pms = size(bp_root_depths(:,1))+1
142                 allocate(min_gp(n_glob_pms))
143                 allocate(max_gp(n_glob_pms))
144
145                 !set global parameter ranges - from soil rooting zone table
146                 DO i = 1,n_glob_pms-1
147                     min_gp(i) = bp_root_depths(i,2)
148                     max_gp(i) = bp_root_depths(i,4)
149                  END DO
150
151                  !set global parameter scaling ranges
152                  min_gp(n_glob_pms) = 0.000005
153                  max_gp(n_glob_pms) = 0.00005
154
155             case default
156                 print *, "ERROR: pedo_tf_SRmax incorrectly defined. "
157
158         end select
159
160         !2. From above, set sizes of the global parameter array and number of random numbers
161         !   to produce.
162         n_gp_all(3) = n_glob_pms
```

```
163
164         allocate(glob_pms_SRmax(n_glob_pms, n_pm_maps))
165         allocate(rand_nums(sum(n_gp_all), n_pm_maps+start_seed-1))
166
167         !generate lists of global parameters between min and max ranges, of length n
168         !need to have list such that starting seed = 50 would give same result as 50th number
169         !from seed=1
170         Call RANDOM_SEED( GET = seed )
171         seed(1:12) = (/ 3,3,3,3,3,3,3,3,3,3,3,3/)
172         Call RANDOM_SEED( PUT = seed )
173         call RANDOM_NUMBER ( rand_nums )
174
175
176         !now normalise random numbers to within bounds
177
178         DO i = 1,n_glob_pms
179             DO j = start_seed, start_seed+n_pm_maps-1
180                 !rn_i is index for the random number
181                 rn_i = (sum(n_gp_all) - n_glob_pms )+ i
182                 !the appropriate random number, in range 0-1
183                 num = rand_nums(rn_i,j)
184                 !normalise the random number to given min/max
185                 num = (num * (max_gp(i)-min_gp(i)))+min_gp(i)
186                 glob_pms_SRmax(i, j-start_seed+1) = num
187
188             END DO
189             IF (n_pm_maps >= 2) THEN
190                 print *, 'First 2 values for parameter ',i,' : ',glob_pms_SRmax(i,1:2)
191             END IF
192         END DO
193
194
195         print *,
196
197     end subroutine init_gp_SRmax
198
199
200
201
202
203     ! ************************************************************************************************
204     ! subroutine: pedotf_SRmax : Pedo-transfer function for SRmax
205     ! ************************************************************************************************
206     ! This routine takes n_i, signalling that we are now calculating parameter map n_i out of n,
207     ! It takes the list of global parameters and required basin predictors, and from that
208     ! applies the pedo-transfer functions to produce a parameter map.
209
210     subroutine pedotf_SRmax(n_i, pedo_tf, glob_pms,bp,porosity, pm_map)
211
212         implicit none
213
214         !declare dummy variables
215         !param map number to do
216         integer, intent(in)                                     :: n_i
217         !equation selection
```

```
218         integer, intent(in)                                         :: pedo_tf
219         !list of global params
220         double precision, dimension(:,:), allocatable, intent(in)   :: glob_pms
221         !basin predictor - land cover map
222         real, dimension(:,:), allocatable, intent(in)               :: bp
223         !basin predictor - porosity map
224         real, dimension(:,:), intent(in)                            :: porosity
225         !output parameter map
226         double precision, dimension(:,:), allocatable, intent(out)  :: pm_map
227
228         !declare local variables
229         integer             :: n_glob_pms       !number of global parameters
230         integer             :: n_pm_maps        !number of parameter maps
231                                                 !defined by ncols in glob_pms_SRmax
232         integer             :: i
233         integer             :: j
234         integer             :: nrows_bp         !number of rows in basin predictor map
235         integer             :: ncols_bp
236         integer             :: lu_class
237         logical             :: warning_sent
238
239
240         !find number of rows and columns in basin predictor files - pm map should be same size
241         nrows_bp = size(bp(:,1))
242         ncols_bp = size(bp(1,:))
243         n_pm_maps = size(glob_pms(1,:))
244         !find number of rows and columns in basin predictor files - pm map should be same size
245         n_pm_maps = size(glob_pms(1,:))
246         allocate(pm_map(nrows_bp, ncols_bp))
247
248         !vary transfer function, depending on user-inputted equation selection.
249         select case(pedo_tf)
250
251             case (0)    !Fixed parameter
252
253                 n_glob_pms = 1
254                 DO i = 1, nrows_bp
255                     DO j = 1, ncols_bp
256
257                         pm_map(i,j) = glob_pms(1, n_i)
258
259                     END DO
260                 END DO
261
262             case (1)    !Global parameter
263
264                 n_glob_pms = 1
265                 DO i = 1, nrows_bp
266                     DO j = 1, ncols_bp
267
268                         pm_map(i,j) = glob_pms(1, n_i)
269
270                     END DO
271                 END DO
272
```

```
273             case (2)   !pedotransfer! SRmax is a * porosity * b if LU=1, c if LU=2, d if LU=3 etc
274
275                 n_glob_pms = 11
276                 warning_sent = .false.
277                 DO i = 1, nrows_bp
278                     DO j = 1, ncols_bp
279
280                         lu_class = int(bp(i,j))
281
282                         !deal with any -9999 values without breaking code
283                         IF (lu_class == -9999) THEN
284                             IF (warning_sent .eqv. .false.) THEN
285                                 print *,'WARNING: some landuse classes are -9999,", &
286                                     " setting SRmax to 0.02 here'
287                                 warning_sent = .true.
288                             END IF
289                             pm_map(i,j) = 0.02
290
291                          ELSEIF (lu_class == 0) THEN
292                             IF (warning_sent .eqv. .false.) THEN
293                                 print *,'WARNING: some landuse classes are 0,", &
294                                     " setting SRmax to 0.02 here'
295                                 warning_sent = .true.
296                             END IF
297                             pm_map(i,j) = 0.02
298                         ELSEIF (porosity(i,j)==-9999) THEN
299                             IF (warning_sent .eqv. .false.) THEN
300                                 print *,'WARNING: some porosity values are -9999,", &
301                                     " setting SRmax to 0.02 here'
302                                 warning_sent = .true.
303                             END IF
304                             pm_map(i,j) = 0.02
305                         ELSE
306
307                        !if no problems with missing data then apply pedotransfer function!
308                             pm_map(i,j) = glob_pms(lu_class,n_i) * glob_pms(n_glob_pms,n_i) * &
309                                 porosity(i,j)
310                         END IF
311
312                     END DO
313                 END DO
314
315             case default
316                 print *, "ERROR: invalid pedo-transfer setting."
317                 stop
318
319         end select
320
321         DO i = 1,nrows_bp
322             DO j = 1,ncols_bp
323                 !add in upper and lower caps
324                 IF (pm_map(i,j)<0.00001) THEN
325                     pm_map(i,j)=0.00001
326                 ELSEIF (pm_map(i,j)>2) THEN
327                     pm_map(i,j)=2
```

```
328                 END IF
329               END DO
330             END DO
331
332         end subroutine pedotf_SRmax
333
334
335
336
337 end module mpr_SRmax
```

C.11 mpr_Smax.f90

```
!! Module containing subroutines for MPR.f90
!! This module contains all subroutines involving MPR for the Smax parameter:
!
!   get_BasPred_Smax    Complete logical list of basin predictors required
!   init_gp_Smax        Initialise global parameters
!   pedotf_Smax         Pedo-transfer function

!! Rosie Lane - 31st August 2017

module mpr_Smax
contains


    ! ***************************************************************************************************
    ! subroutine: get_BasPred_Smax : Complete logical list of basin predictors required for Smax
    ! ***************************************************************************************************
    ! Given the user-specified pedo transfer function selected for Smax, this subroutine adds to a
    ! logical list of required basin predictors, by specifying which predictors it requires.

    subroutine get_BasPred_Smax(pedo_tf_Smax, req_soils)

        implicit none

        ! declare dummy arguments
        integer, intent(inout)                    :: pedo_tf_Smax !pedo_tf_equation to be used
        logical, dimension(*), intent(inout)  :: req_soils    !soils data required logical

        ! define required BasinPredictors for Smax
        select case(pedo_tf_Smax)


            case (0)    !fixed parameter - no basin predictors required

            case (1)    !global parameter - no basin predictors required

            case (2)     !ptf based on soil depth

                req_soils(14) = .true.   !soil depth to rock map
                req_soils(17) = .true.   !porosity map

            case (3)    !ptf based on soil depth - with addition for productive hydrogeology

                req_soils(14) = .true.   !soil depth to rock map
                req_soils(17) = .true.   !porosity map
                !productive hydrogeology map (1=high productive, 0=low/mod productive)
                req_soils(18) = .true.

            case (4)
                print *, "ERROR: A second transfer function has not yet been programmed for Smax"
                print *, "Select pedo_tf = 0 to treat it as a fixed parameter"
                print *, "Select pedo_tf = 1 to treat it as a global parameter"
                 print *, "pedo_tf_Smax = 2, sets a ptf based on a soil depth map"
```

```
53                stop
54
55            case default
56                print *, "WARNING: a valid pedo-tf equation for Smax must be specified in the", &
57                    " control file"
58                print *, "Smax will be set to the default of fixed parameter"
59                print *, "The following options can be selected in the control file:"
60                print *, "pedo_tf_Smax = 0, sets it as a fixed parameter"
61                print *, "pedo_tf_Smax = 1, sets it as a global parameter"
62                print *, "pedo_tf_Smax = 2, sets a ptf based on a soil depth map"
63                print *,
64
65                pedo_tf_Smax = 0
66
67
68        end select
69
70    end subroutine get_BasPred_Smax
71
72
73
74
75
76    ! ***************************************************************************************
77    ! subroutine: init_gp_Smax : Initialise global parameters for Smax
78    ! ***************************************************************************************
79    ! This subroutine does the following:
80    !   1. Defines how many global parameters (n_glob_pms) are needed for the user-specified
81    !           transfer function
82    !       TF 0    : 1 global parameter
83    !       TF 1    : 3 global parameters
84    !   2. Defines min/max ranges for global parameters
85    !   3. Generates list of global parameters using the rand function
86    !       - user specified start_seed can be used to skip to any point in this list
87    !       - global parameter list is of dimensions number of global params by number of param
88    !           files needed
89    !   4. Transforms global parameters to be within set min/max ranges
90
91    subroutine init_gp_Smax(pedo_tf_Smax,glob_pms_Smax, n_pm_maps, start_seed, n_gp_all,pm_range)
92
93        use dta_utility
94
95        implicit none
96
97        !declare dummy variables
98        integer, intent(in)                                     :: pedo_tf_Smax
99        !global param list
100       double precision, allocatable, dimension(:,:),intent(inout) :: glob_pms_Smax
101       integer, intent(in)                                     :: n_pm_maps
102       !to start rand num generator
103       integer, intent(in)                                     :: start_seed
104       !vector of number of global params for all params
105       integer, dimension(7), intent(inout)                    :: n_gp_all
106       !min, fixed, max values for this parameter
107       double precision, dimension(3), intent(in)              :: pm_range
```

```
108
109         !declare local variables
110         integer                                         :: n_glob_pms   !number of global parameters
111         double precision, allocatable, dimension(:)     :: min_gp   !min values for global parameters
112         double precision, allocatable, dimension(:)     :: max_gp
113         real, dimension(:,:),allocatable                :: rand_nums
114         double precision                                :: num
115         integer, dimension(12)                          :: seed
116         integer                                         :: i,j, rn_i
117
118
119         !1. Define how many global parameters should exist and min/max ranges, based on selected
120         !   transfer function
121         select case(pedo_tf_Smax)
122
123             case (0) !Treat as fixed parameter (for ease this is a global parameter with a 0 range)
124
125                 print *, "Smax is a fixed parameter, of value ",pm_range(2)
126                 n_glob_pms = 1
127                 allocate(min_gp(1)) !min and max values have length 1, as we have 1 global parameter
128                 allocate(max_gp(1))
129                 min_gp(1) = pm_range(2)     !min and max values set to same fixed parameter value.
130                 max_gp(1) = pm_range(2)
131
132             case (1)    !Treat as global parameter
133
134                 print *, "Smax is a global parameter, in the range ",pm_range(1)," ",pm_range(3)
135                 n_glob_pms = 1
136                 allocate(min_gp(1)) !min and max values have length 1, as we have 1 global parameter
137                 allocate(max_gp(1))
138                 min_gp(1) = pm_range(1)     !min and max values set from parameter file
139                 max_gp(1) = pm_range(3)
140
141             case (2)    !Pedotransfer function!!!
142
143                 print *, "Smax will use a pedotransfer function based on soil depth"
144                 print *, "Smax = (a * soil_depth)"
145
146                 !allocate number of global parameters
147                 n_glob_pms = 1
148                 allocate(min_gp(n_glob_pms))
149                 allocate(max_gp(n_glob_pms))
150
151                 !set global parameter ranges
152                 min_gp(1) = 0.000002
153                 max_gp(1) = 0.0001
154
155                 !min_gp(2) = 0
156                 !max_gp(2) = 1
157
158             case (3)    !Pedotransfer function, with addition for high productivity geology
159
160                 print *, "Smax will use a pedotransfer function based on soil depth and hydrogeology"
161                 print *, "Smax = (a * soil_depth) + (b * high_prod_geology?)"
162
```

```
163                 !allocate number of global parameters
164                 n_glob_pms = 2
165                 allocate(min_gp(n_glob_pms))
166                 allocate(max_gp(n_glob_pms))
167
168                 !set global parameter ranges
169                 min_gp(1) = 0.000002
170                 max_gp(1) = 0.0001
171
172                 min_gp(2) = 0
173                 max_gp(2) = 4
174
175             case default
176                 print *, "ERROR: pedo_tf_Smax incorrectly defined. "
177
178         end select
179
180         !2. From above, set sizes of the global parameter array and number of random numbers
181         !   to produce.
182         n_gp_all(7) = n_glob_pms
183
184         allocate(glob_pms_Smax(n_glob_pms, n_pm_maps))
185         allocate(rand_nums(sum(n_gp_all), n_pm_maps+start_seed-1))
186
187         !generate lists of global parameters between min and max ranges, of length n
188         !need to have list such that starting seed = 50 would give same result as 50th number
189         !from seed=1
190         Call RANDOM_SEED( GET = seed )
191         seed(1:12) = (/ 3,3,3,3,3,3,3,3,3,3,3,3/)
192         Call RANDOM_SEED( PUT = seed )
193         call RANDOM_NUMBER ( rand_nums )
194
195
196
197         !now normalise random numbers to within bounds
198
199         DO i = 1,n_glob_pms
200             DO j = start_seed, start_seed+n_pm_maps-1
201                 !rn_i is index for the random number
202                 rn_i =( sum(n_gp_all) - n_glob_pms )+ i
203                 !the appropriate random number, in range 0-1
204                 num = rand_nums(rn_i,j)
205                 !normalise the random number to given min/max
206                 num = (num * (max_gp(i)-min_gp(i)))+min_gp(i)
207                 glob_pms_Smax(i, j-start_seed+1) = num
208
209             END DO
210             IF (n_pm_maps >= 2) THEN
211                 print *, 'First 2 values for parameter ',i,' : ',glob_pms_Smax(i,1:2)
212             END IF
213         END DO
214
215
216         print *,
217
```

```
218     end subroutine init_gp_Smax
219
220
221
222
223
224     ! ************************************************************************************************
225     ! subroutine: pedotf Smax: Pedo-transfer function for Smax
226     ! ************************************************************************************************
227     ! This routine takes n_i, signalling that we are now calculating parameter map n_i out of n,
228     ! It takes the list of global parameters and required basin predictors, and from that
229     ! applies the pedo-transfer functions to produce a parameter map.
230
231     subroutine pedotf_Smax(n_i, pedo_tf, glob_pms,bp,porosity,hydrogeo,pm_map)
232
233         implicit none
234
235         !declare dummy variables
236         !param map number to do
237         integer, intent(in)                                     :: n_i
238         !equation selection
239         integer, intent(in)                                     :: pedo_tf
240         !list of global params
241         double precision, dimension(:,:), allocatable, intent(in)    :: glob_pms
242         !basin predictor - soil depth map
243         real, dimension(:,:), allocatable, intent(in)           :: bp
244         !basin predictor - porosity map
245         real, dimension(:,:), intent(in)                        :: porosity
246         !basin predictor - is high productive hydrogeology?
247         real, dimension(:,:), intent(in)                        :: hydrogeo
248         !output parameter map
249         double precision, dimension(:,:), allocatable, intent(out)   :: pm_map
250
251         !declare local variables
252         integer             :: n_glob_pms       !number of global parameters
253         integer             :: n_pm_maps        !number of parameter maps
254                                                 !defined by ncols in glob_pms_Smax
255         integer             :: i
256         integer             :: j
257         integer             :: nrows_bp         !number of rows in basin predictor map
258         integer             :: ncols_bp
259
260
261         !find number of rows and columns in basin predictor files - pm map should be same size
262         nrows_bp = size(bp(:,1))
263         ncols_bp = size(bp(1,:))
264         n_pm_maps = size(glob_pms(1,:))
265         !find number of rows and columns in basin predictor files - pm map should be same size
266         n_pm_maps = size(glob_pms(1,:))
267         allocate(pm_map(nrows_bp, ncols_bp))
268
269         !vary transfer function, depending on user-inputted equation selection.
270         select case(pedo_tf)
271
272             case (0)    !Fixed parameter
```

```
273
274                 n_glob_pms = 1
275                 DO i = 1, nrows_bp
276                     DO j = 1, ncols_bp
277
278                         pm_map(i,j) = glob_pms(1, n_i)
279
280                     END DO
281                 END DO
282
283             case (1)    !Global parameter
284
285                 n_glob_pms = 1
286                 DO i = 1, nrows_bp
287                     DO j = 1, ncols_bp
288
289                         pm_map(i,j) = glob_pms(1, n_i)
290
291                     END DO
292                 END DO
293
294             case (2)    !Pedotransfer function!!! - Smax = (a * soil_depth * porosity)
295
296                 n_glob_pms = 1
297                 DO i = 1, nrows_bp
298                     DO j = 1, ncols_bp
299
300                         pm_map(i,j) = glob_pms(1,n_i) * bp(i,j) * porosity(i,j)
301                         !pm_map(i,j) = pm_map(i,j) + glob_pms(2,n_i)
302
303                         !deal with -9999 values
304                         IF((bp(i,j)==-9999).OR.(porosity(i,j)==-9999)) THEN
305                             pm_map(i,j) = 0.5 !old default value
306                         ENDIF
307
308                     END DO
309                 END DO
310
311             case (3)
312             !Pedotransfer function Smax = (a * soil_depth * porosity) + (b * high_prod_geology?)
313
314                 n_glob_pms = 2
315                 DO i = 1, nrows_bp
316                     DO j = 1, ncols_bp
317
318                         pm_map(i,j) = glob_pms(1,n_i) * bp(i,j) * porosity(i,j)
319
320                         IF (hydrogeo(i,j) == 1) THEN !areas overlaying high productivity geology.
321                             pm_map(i,j) = pm_map(i,j) + glob_pms(2,n_i)
322                         END IF
323
324                         !deal with -9999 values
325                         IF((bp(i,j)==-9999).OR.(porosity(i,j)==-9999)) THEN
326                             pm_map(i,j) = 0.5 !old default value
327                         ENDIF
```

```
328
329                     END DO
330                 END DO
331
332
333             case default
334                 print *, "ERROR: invalid pedo-transfer setting."
335                 stop
336
337         end select
338
339          ! set a limit - to stop crazy values causing crashes!
340         DO i = 1,nrows_bp
341             DO j = 1,ncols_bp
342                 !add in upper and lower caps
343                 IF (pm_map(i,j)<0.001) THEN
344                     pm_map(i,j)=0.001
345                 ELSEIF (pm_map(i,j)>15) THEN
346                     pm_map(i,j)=15
347                 END IF
348             END DO
349         END DO
350
351     end subroutine pedotf_Smax
352
353
354
355
356 end module mpr_Smax
```

sectionmpr_extract_BasPred.f90

```
!! Module containing subroutines for MPR.f90
!! This module contains all subroutines for reading, clipping and writing Basin predictors

!! Rosie Lane - 31st August 2017
!
!   Subroutines:
!   Read_clip_bp        Read basin predictor file, and clip to the same area as HRU file
!   Read_req_soils       Reads in all required soils, writes clipped grid if required


module mpr_extract_BasPred
    contains


    ! ************************************************************************************************
    ! subroutine: Read_clip_bp : Read basin predictor file, and clip to the same area as HRU file
    ! ************************************************************************************************
    ! This subroutine reads in a basin predictor grid, specified by filename, and clips it to the
    ! required area covered by the HRU file given in stats_HRU, returning grid_clipped.
    subroutine Read_clip_bp(filename,grid_clipped, stats_HRU, HRU_map)

    use dta_utility

    implicit none

        !declare dummy variables
        !filename is full filepath to stored .asc file of basin predictor
        !stats_HRU contains xll, yll, cellsize and nodata value
        character(len=*), intent(in)          :: filename
        double precision, dimension(:,:), allocatable, intent(out)  :: grid_clipped
        double precision, dimension(4), intent(in)    :: stats_HRU
        double precision, dimension(:,:), intent(in)     :: HRU_map

        !declare local variables

        !for reading an ascii grid
        double precision, dimension(:,:), allocatable    :: grid
        double precision                                 :: xll
        double precision                                 :: yll
        double precision                                 :: cellsize
        double precision                                 :: nodata
        integer                                          :: ncol
        integer                                          :: nrow

        !HRU map stats
        integer                                          :: nrows_HRU
        integer                                          :: ncols_HRU

        !calculation of row/col equivalents for national grid locations
        integer     :: i_xll
        integer     :: i_yll
        integer     :: i_yul
        integer     :: i_xlr
```

```
54          integer     :: x, y
55
56          ! get HRU map stats
57          nrows_HRU = size(HRU_map(:,1))
58          ncols_HRU = size(HRU_map(1,:))
59
60          ! read basin predictor file
61          call read_ascii_grid(trim(filename),grid,ncol,nrow,xll,yll,cellsize,nodata)
62
63          ! make a check to see if cellsizes differ - problems if so!!
64          IF (cellsize /= stats_HRU(3)) THEN
65              print *, "ERROR: Cellsize differs between HRU and basin predictor maps"
66              stop
67          END IF
68
69          ! get row/col numbers of corners where HRU map and basin predictor map overlap
70          i_xll = ((stats_HRU(1) - xll) / cellsize) + 1
71          i_yll = nrow - ((stats_HRU(2) - yll) / cellsize)
72          i_yul = i_yll - (nrows_HRU - 1)
73          i_xlr = i_xll + ncols_HRU - 1
74
75          ! allocate output ascii grid to be same size as HRU grid
76          allocate(grid_clipped(nrows_HRU, ncols_HRU))
77
78          ! loop through area where HRU map overlays basin predictor map, and extract basin pred
79          ! map values
80          DO x = 1, ncols_HRU-1
81              DO y = 1, nrows_HRU-1
82                  grid_clipped(y,x) = grid(i_yul+y-1,i_xll+x-1)
83              END DO
84          END DO
85
86          ! make sure nodata value is set to the same as HRU nodata value
87          IF (nodata /= stats_HRU(4)) THEN
88              print *, "Converting basin predictor nodata value to that of HRU map: ",&
89              int(stats_HRU(4))
90              DO x = 1, ncols_HRU-1
91                  DO y = 1, nrows_HRU-1
92                      IF (grid_clipped(y,x) == nodata) THEN
93                          grid_clipped(y,x) = stats_HRU(4)
94                      END IF
95                  END DO
96              END DO
97          END IF
98
99          ! set all areas which are nodata in HRU file to nodata in basin predictor file
100         ! - this can crop the map to the correct extent if later importing into arcmap etc.
101         DO x = 1,ncols_HRU
102             DO y = 1,nrows_HRU
103                 IF (HRU_map(y,x) == stats_HRU(4)) THEN !if nodata in HRU map
104                     grid_clipped(y,x) = stats_HRU(4)
105                 END IF
106             END DO
107         END DO
108
```

```
109         !print *, "Successfully read basin predictor: ",filename
110
111
112     end subroutine Read_clip_bp
113
114
115     ! ************************************************************************************************
116     ! subroutine: Read_req_soils
117     ! ************************************************************************************************
118     ! This routine reads in all the required basin predictors (as specified in req_bp)
119     ! All basin predictors are returned in the variable bp_maps
120     ! If the user has specified writing of basin predictors,
121     ! then all basin predictors required will also be written as an ascii file.
122     subroutine read_req_soils(input_folder, &
123         out_folder, &
124         fnames_baspred,&
125         req_bp,&
126         stats_HRU,&
127         HRU_map,&
128         save_bp_maps,&
129         bp_maps, &
130         soilmusiddata, &
131         bp_root_depths)
132
133         !use dta_MPR
134         use dta_utility
135
136         implicit none
137
138         !declare dummy variables
139         !req_bp: logical declaring whether each basin predictor is required
140         !bp_maps: array of all basin predictor ascii maps
141         !soilmusiddata: array of all soils depth data tables.
142         !bp_root_depths: landcover class rooting depth table
143         character(len=1024), intent(in)      :: input_folder    !MPR input folder
144         character(len=1024), intent(in)      :: out_folder      !Folder to store output
145         character(len=1024), dimension(18),intent(in)   :: fnames_baspred
146         logical, dimension(*), intent(in)               :: req_bp
147         double precision, dimension(4), intent(in)      :: stats_HRU
148         double precision, dimension(:,:), intent(in)    :: HRU_map
149         integer, intent(in)                             :: save_bp_maps
150         real, allocatable, dimension(:,:,:), intent(out)    :: bp_maps
151         double precision, allocatable, dimension(:,:,:), intent(out) :: soilmusiddata
152         real, allocatable, dimension(:,:), intent(out) :: bp_root_depths
153
154         !declare local variables
155         double precision, allocatable, dimension(:,:)   :: sand_0_10
156         double precision, allocatable, dimension(:,:)   :: silt_0_10
157         double precision, allocatable, dimension(:,:)   :: clay_0_10
158         double precision, allocatable, dimension(:,:)   :: orgm_0_10
159         real, allocatable, dimension(:,:)   :: ksat_decline
160         double precision, allocatable, dimension(:,:)   :: MUSIDs
161         double precision, allocatable, dimension(:,:)   :: n_entries
162         double precision, allocatable, dimension(:,:)   :: tempdata
163         double precision, allocatable, dimension(:,:)   :: map
```

```
164         character(len=1024)                             :: temp_fn
165         double precision                                :: xllcorner_HRU
166         double precision                                :: yllcorner_HRU
167         double precision                                :: cellsize_HRU
168         double precision                                :: nodata_HRU
169         integer                                         :: ncols_HRU
170         integer                                         :: nrows_HRU
171         integer                                         :: i
172         integer                                         :: nrows
173         integer                                         :: ncols
174         integer                                         :: nmusids
175
176         !extract properties of stats_HRU
177         xllcorner_HRU = stats_HRU(1)
178         yllcorner_HRU = stats_HRU(2)
179         cellsize_HRU = stats_HRU(3)
180         nodata_HRU = stats_HRU(4)
181         ncols_HRU = size(HRU_map(1,:))
182         nrows_HRU = size(HRU_map(:,1))
183
184         !bp maps dimensions refer to (x, y, req_bp)
185         allocate(bp_maps(nrows_HRU, ncols_HRU,18))
186
187         ! Read in required basin predictor maps - and clip to size of HRU map
188         !   bp 1    surface % sand
189         !   bp 2    surface % silt
190         !   bp 3    surface % clay
191         !   bp 4    surface % organic
192         !   bp 5    map of soil depth profiles
193         !   bp 7-11 tables of soil depth profiles
194         !   bp 12   Land cover map
195         !   bp 13   Root depth table
196         !   bp 14   Depth to bedrock map
197         !   bp 15   surface bulk density
198         !   bp 16   locations of organic (binary with 1=organic soil)
199         !           bp 16 now NOT USED as threshold of 35% OM for organic soils
200         !   bp 17   porosity map
201         !   bp 18   locations of high productive hydrogeology (binary, 1=highly productive)
202
203         !Read in surface percentage sand - basin predictor 1
204         IF (req_bp(1)) THEN
205             print *, "Reading file ",trim(fnames_baspred(1))//".asc"
206             call Read_clip_bp(trim(input_folder)//trim(fnames_baspred(1))//".asc",&
207             sand_0_10, stats_HRU, HRU_map)
208             IF (save_bp_maps == 1) THEN
209                 temp_fn = trim(input_folder) // trim(fnames_baspred(1))//"_clipped.asc"
210                 print *, 'Writing clipped sand basin predictor file in input folder'
211                 print *,
212                 call write_ascii_grid(temp_fn,sand_0_10,ncols_HRU,nrows_HRU,xllcorner_HRU,&
213                 yllcorner_HRU,cellsize_HRU,nodata_HRU,6)
214             END IF
215             !store all soils data in bp_maps for easy passing between functions
216             bp_maps(:,:,1) = sand_0_10
217         END IF
218
```

```
219         !Read in surface percentage silt - basin predictor 2
220         IF (req_bp(2)) THEN
221             print *, "Reading file ",trim(fnames_baspred(2))//".asc"
222             call Read_clip_bp(trim(input_folder)//trim(fnames_baspred(2))//".asc",&
223             silt_0_10, stats_HRU, HRU_map)
224             IF (save_bp_maps == 1) THEN
225                 temp_fn = trim(input_folder) // trim(fnames_baspred(2))//"_clipped.asc"
226                 print *, 'Writing clipped silt basin predictor file in input folder'
227                 print *,
228                 call write_ascii_grid(temp_fn,silt_0_10,ncols_HRU,nrows_HRU,xllcorner_HRU,&
229                 yllcorner_HRU,cellsize_HRU,nodata_HRU,6)
230             END IF
231             bp_maps(:,:,2) = silt_0_10
232         END IF
233
234         !Read in surface percentage clay - basin predictor 3
235         IF (req_bp(3)) THEN
236             print *, "Reading file ",trim(fnames_baspred(3))//".asc"
237             call Read_clip_bp(trim(input_folder)//trim(fnames_baspred(3))//".asc",clay_0_10,&
238             stats_HRU, HRU_map)
239             IF (save_bp_maps == 1) THEN
240                 temp_fn = trim(input_folder) // trim(fnames_baspred(3))//"_clipped.asc"
241                 print *, 'Writing clipped clay basin predictor file in input folder'
242                 print *,
243                 call write_ascii_grid(temp_fn,clay_0_10,ncols_HRU,nrows_HRU,xllcorner_HRU,&
244                 yllcorner_HRU,cellsize_HRU,nodata_HRU,6)
245             END IF
246             bp_maps(:,:,3) = clay_0_10
247         END IF
248
249         !Read in surface percentage organic matter - basin predictor 4
250         IF (req_bp(4)) THEN
251             print *, "Reading file ",trim(fnames_baspred(4))//".asc"
252             call Read_clip_bp(trim(input_folder)//trim(fnames_baspred(4))//".asc",orgM_0_10,&
253             stats_HRU, HRU_map)
254             IF (save_bp_maps == 1) THEN
255                 temp_fn =  trim(input_folder) // trim(fnames_baspred(4))//"_clipped.asc"
256                 print *, 'Writing clipped organic matter basin predictor file in input folder'
257                 print *,
258                 call write_ascii_grid(temp_fn,orgm_0_10,ncols_HRU,nrows_HRU,xllcorner_HRU,&
259                 yllcorner_HRU,cellsize_HRU,nodata_HRU,6)
260             END IF
261             bp_maps(:,:,4) = orgm_0_10
262         END IF
263
264         !Read in ascii information describing SZM soils depth profiles - basin predictor 5
265         IF (req_bp(5)) THEN
266             print *, "Reading file ",trim(fnames_baspred(5))//".asc"
267             temp_fn =(trim(input_folder) // trim(fnames_baspred(5))//".asc")
268             call Read_clip_bp(trim(temp_fn),MUSIDs, stats_HRU, HRU_map)
269             bp_maps(:,:,6) = MUSIDs
270             bp_maps(:,:,5) = MUSIDs
271
272             IF (save_bp_maps == 1) THEN
273                 print *, 'Writing clipped MUSIDs in input folder'
```

```
274             print *,
275             temp_fn =  trim(input_folder) // trim(fnames_baspred(5))//"_clipped.asc"
276             call write_ascii_grid(temp_fn,MUSIDs,ncols_HRU,nrows_HRU,xllcorner_HRU,&
277             yllcorner_HRU,cellsize_HRU,nodata_HRU,6)
278         END IF
279     END IF
280
281     !Read in tables of soils depth profiles - basin predictor 7-11
282     IF (req_bp(7)) THEN
283         i = 7
284         temp_fn =  trim(input_folder) // trim(fnames_baspred(i))//".txt"
285         open (99, file = temp_fn, status = 'old')
286         read(99,*) nrows, ncols, nmusids
287         close(99)
288         !print *, 'nrows = ',nrows,' ncols = ',ncols,' nmusids = ',nmusids
289         allocate(soilmusiddata(nrows,ncols,5))
290
291         DO i = 7,11
292             print *, "Reading file ", trim(fnames_baspred(i))
293             temp_fn =  trim(input_folder) // trim(fnames_baspred(i)) // ".txt"
294             call read_numeric_list(temp_fn, ncols, 2,tempdata )
295             soilmusiddata(:,:,i-6) = tempdata
296             deallocate(tempdata)    !because this is allocated within read_numeric_list
297         END DO
298          if (save_bp_maps == 1) THEN
299             print *,
300         end if
301     END IF
302
303     !Read in SRmax land cover map - basin predictor 12
304     IF (req_bp(12)) THEN
305
306         print *, "Reading file ",trim(fnames_baspred(12))//".asc"
307         temp_fn =  trim(input_folder) // trim(fnames_baspred(12))//".asc"
308         call Read_clip_bp(trim(temp_fn),map, stats_HRU, HRU_map)
309         bp_maps(:,:,12) = map
310
311         IF (save_bp_maps == 1) THEN
312             print *, 'Writing clipped Land Cover Map in input folder'
313             print *,
314             temp_fn =  trim(input_folder) // trim(fnames_baspred(12))//"_clipped.asc"
315             call write_ascii_grid(temp_fn,map,ncols_HRU,nrows_HRU,xllcorner_HRU,&
316             yllcorner_HRU,cellsize_HRU,nodata_HRU,6)
317         END IF
318     END IF
319
320     !Read in SRmax rooting depth table  - basin predictor 13
321     IF (req_bp(13)) THEN
322         print *, "Reading file ", trim(fnames_baspred(13)),".txt"
323         temp_fn =  trim(input_folder) // trim(fnames_baspred(13))//".txt"
324         open (99, file = temp_fn, status = 'old')
325         read(99,*) nrows, ncols
326         close(99)
327         call read_numeric_list(temp_fn, 4, 3,tempdata )
328         allocate(bp_root_depths(nrows,4))
```

```
329             bp_root_depths= tempdata
330             if (save_bp_maps == 1) THEN
331                 print *,
332             end if
333
334         END IF
335
336
337         !Read in Smax depth to bedrock map  - basin predictor 14
338         IF (req_bp(14)) THEN
339             print *, "Reading file ",trim(fnames_baspred(14))//".asc"
340             temp_fn =  trim(input_folder) // trim(fnames_baspred(14))//".asc"
341             call Read_clip_bp(trim(temp_fn),map, stats_HRU, HRU_map)
342             bp_maps(:,:,14) = map
343
344             IF (save_bp_maps == 1) THEN
345
346                 print *, 'Writing clipped soil depth map in input folder'
347                 print *,
348                 temp_fn =  trim(input_folder) // trim(fnames_baspred(14))//"_clipped.asc"
349                 call write_ascii_grid(temp_fn,map,ncols_HRU,nrows_HRU,xllcorner_HRU,&
350                 yllcorner_HRU,cellsize_HRU,nodata_HRU,6)
351             END IF
352         END IF
353
354         !Read in lnTo bulk density  - basin predictor 15
355         IF (req_bp(15)) THEN
356             print *, "Reading file ",trim(fnames_baspred(15))//".asc"
357             temp_fn =  trim(input_folder) // trim(fnames_baspred(15))//".asc"
358             call Read_clip_bp(trim(temp_fn),map, stats_HRU, HRU_map)
359             bp_maps(:,:,15) = map
360             IF (save_bp_maps == 1) THEN
361                 print *, 'Writing clipped bulk density map in input folder'
362                 print *,
363                 temp_fn =  trim(input_folder) // trim(fnames_baspred(15))//"_clipped.asc"
364                 call write_ascii_grid(temp_fn,map,ncols_HRU,nrows_HRU,xllcorner_HRU,&
365                 yllcorner_HRU,cellsize_HRU,nodata_HRU,6)
366             END IF
367         END IF
368
369
370         !Read in ISorganic? map  - basin predictor 16
371         IF (req_bp(16)) THEN
372             print *, "Reading file ",trim(fnames_baspred(16))//".asc"
373             temp_fn =  trim(input_folder) // trim(fnames_baspred(16))//".asc"
374             call Read_clip_bp(trim(temp_fn),map, stats_HRU, HRU_map)
375             bp_maps(:,:,16) = map
376             IF (save_bp_maps == 1) THEN
377                 print *, 'Writing clipped identify organic soils map in input folder'
378                 print *,
379                 temp_fn =  trim(input_folder) // trim(fnames_baspred(16))//"_clipped.asc"
380                 call write_ascii_grid(temp_fn,map,ncols_HRU,nrows_HRU,xllcorner_HRU,&
381                 yllcorner_HRU,cellsize_HRU,nodata_HRU,6)
382             END IF
383         END IF
```

```
384
385         !Read in porosity map  - basin predictor 17
386         IF (req_bp(17)) THEN
387             print *, "Reading file ",trim(fnames_baspred(17))//".asc"
388             temp_fn =  trim(input_folder) // trim(fnames_baspred(17))//".asc"
389             call Read_clip_bp(trim(temp_fn),map, stats_HRU, HRU_map)
390             bp_maps(:,:,17) = map
391             IF (save_bp_maps == 1) THEN
392                 print *, 'Writing clipped porosity map in input folder'
393                 print *,
394                 temp_fn =  trim(input_folder) // trim(fnames_baspred(17))//"_clipped.asc"
395                 call write_ascii_grid(temp_fn,map,ncols_HRU,nrows_HRU,xllcorner_HRU,&
396                 yllcorner_HRU,cellsize_HRU,nodata_HRU,6)
397             END IF
398         END IF
399
400         !Read in hydrogeology map - basin predictor 18
401         IF (req_bp(18)) THEN
402             print *, "Reading file ",trim(fnames_baspred(18))//".asc"
403             temp_fn =  trim(input_folder) // trim(fnames_baspred(18))//".asc"
404             call Read_clip_bp(trim(temp_fn),map, stats_HRU, HRU_map)
405             bp_maps(:,:,18) = map
406             IF (save_bp_maps == 1) THEN
407                 print *, 'Writing clipped hydrogeology map in input folder'
408                 print *,
409                 temp_fn =  trim(input_folder) // trim(fnames_baspred(18))//"_clipped.asc"
410                 call write_ascii_grid(temp_fn,map,ncols_HRU,nrows_HRU,xllcorner_HRU,&
411                 yllcorner_HRU,cellsize_HRU,nodata_HRU,6)
412             END IF
413         END IF
414
415     end subroutine read_req_soils
416
417 end module mpr_extract_BasPred
```

C.12 mpr_upscaling.f90

```
!! Module containing subroutines for MPR.f90
!! This module contains all subroutines involving in parameter upscaling
!
!   extract_pm_HRU :    extracts list of params from all cells of selected HRU
!   upscale_harmonic :  upscales a list of parameters using the harmonic mean
!   upscale_arithmetic: upscales a list of parameters using the arithmetic mean

!! Rosie Lane - 31st August 2017

module mpr_upscaling
contains



    ! ***************************************************************************************************
    ! subroutine: extract_pm_HRU : extracts list of params from all cells of selected HRU
    ! ***************************************************************************************************
    !  extract_pm_HRU aims to:
    !  1. Loop through all values in a HRU and parameter value file
    !  2. For a specified HRU number, extract a vector of all parameter values within that HRU
    !  3. Output vector of parameter values

    subroutine extract_pm_HRU(HRU, HRU_map, pm_map, pms_HRU)

        implicit none

        !declare dummy variables
        integer, intent(in)                         :: HRU                  !id of current HRU
        double precision, dimension(:,:), intent(in) :: HRU_map             !map of HRUs
        double precision, dimension(:,:), intent(in) :: pm_map              !map of MPR parameters
        !output list of params in HRU
        double precision, allocatable,dimension(:), intent(out) :: pms_HRU

        !declare local variables
        integer                 :: nrows_HRU    !dimensions of HRU & ASSUMED dimensions of pm maps
        integer                 :: ncols_HRU
        integer                 :: i
        integer                 :: j
        integer                 :: n_pms_HRU     !number of cells in current HRU


        !define local variables from HRU map
        nrows_HRU = size(HRU_map(:,1))
        ncols_HRU = size(HRU_map(1,:))

        !initialise
        n_pms_HRU = 0

        ! Gemma - n_pms_HRU2 = sum(HRU_map[, HRU_map.eq.HRU])
        ! Find the number of HRU squares matching the given HRU value.
!        DO i = 1, nrows_HRU
!            DO j = 1, ncols_HRU
```

```
!                IF (HRU_map(i,j)==HRU) THEN      !When you index a variable of the correct HRU...
!                    n_pms_HRU = n_pms_HRU + 1
!                END IF
!            END DO
!        END DO

        ! Find the number of HRU squares matching the given HRU value
        n_pms_HRU = int(SUM(HRU_map,MASK=HRU_map==HRU))/HRU

        !Allocate size of pms_HRU to prevent segmentation fault
        allocate(pms_HRU(n_pms_HRU))

!        !Loop through HRU_map again, extracting all pm_map values in the correct HRU
!        DO i = nrows_HRU,1,-1
!            DO j = ncols_HRU,1,-1
!                IF (HRU_map(i,j)==HRU) THEN      !When you index a variable of the correct HRU...
!                    pms_HRU(n_pms_HRU) = pm_map(i,j)  !...add the parameter value into pms_HRU.
!                    n_pms_HRU = n_pms_HRU -1
!                END IF
!            END DO
!        END DO

        !Extract all pm values within the correct HRU
        pms_HRU = PACK(pm_map, HRU_map==HRU)


    end subroutine extract_pm_HRU




    ! ***************************************************************************************************
    ! subroutine: upscale_harmonic : upscales a list of parameters using the harmonic mean
    ! ***************************************************************************************************
    !   upscale_harmonic aims to:
    !   take input of a vector of parameter values
    !   output the harmonic mean of those values

    subroutine upscale_harmonic(pms_HRU, upscaled_pm)

        implicit none

        double precision, dimension(:), intent(in)  :: pms_HRU  !List of params from extract_pm_HRU
        double precision, intent(out)               :: upscaled_pm  !output upscaled param
        double precision, dimension(:), allocatable :: recip_pms
        double precision                            :: sum_pms

        recip_pms = 1 / pms_HRU     !take reciprocal of all params
        sum_pms = SUM(recip_pms)    !take sum of recipocal params
        upscaled_pm = SIZE(pms_HRU) / sum_pms  !H = n params / sum recip params

    end subroutine upscale_harmonic
```

```
108     ! ************************************************************************************************
109     ! subroutine: upscale_arithmetic: upscales a list of parameters using the arithmetic mean
110     ! ************************************************************************************************
111
112     subroutine upscale_arithmetic(pms_HRU, upscaled_pm)
113
114         implicit none
115
116         double precision, dimension(:), intent(in) :: pms_HRU   !List of params from extract_pm_HRU
117         double precision, intent(out)              :: upscaled_pm   !output upscaled param
118         double precision                           :: sum_pms
119         integer :: npoints,i
120
121         npoints = SIZE(pms_HRU)
122         !do i = 1,SIZE(pms_HRU)
123         !    if(pms_HRU(i)==-9999)then
124         !        pms_HRU(i)=0
125         !        npoints = npoints-1
126         !    end if
127         !end
128
129         sum_pms = SUM(pms_HRU)                  !take sum of params
130
131         !if (npoints>0) then
132             upscaled_pm = sum_pms / npoints  !A = sum params / n params
133         !else
134         !    print *, 'Error in upscale arithmetic - all points were NaN!!!!!!!    pm set to 1'
135         !    upscaled_pm = 1
136         !end
137
138     end subroutine upscale_arithmetic
139
140     ! ************************************************************************************************
141     ! subroutine: upscale_geometric: upscales a list of parameters using the geometric mean
142     ! ************************************************************************************************
143
144     subroutine upscale_geometric(pms_HRU, upscaled_pm)
145
146         implicit none
147
148         double precision, dimension(:), intent(in) :: pms_HRU   !List of params from extract_pm_HRU
149         double precision, intent(out)              :: upscaled_pm   !output upscaled param
150         double precision                           :: prod_pms
151
152         prod_pms = PRODUCT(pms_HRU)                    !take product of params
153         upscaled_pm = prod_pms**(1/SIZE(pms_HRU))       !G = (product(params))^(1/n)
154
155     end subroutine upscale_geometric
156
157     ! ************************************************************************************************
158     ! subroutine: upscale_majority: upscales selecting the most frequently occuring value
159     ! ************************************************************************************************
160
161     subroutine upscale_majority(pms_HRU, upscaled_pm)
162
```

```
163         implicit none
164
165         double precision, dimension(:), intent(in) :: pms_HRU   !List of params from extract_pm_HRU
166         double precision, intent(out)              :: upscaled_pm   !output upscaled param
167
168         !local variables
169         double precision                           :: prod_pms
170         !frequency of occurance of each parameter value
171         integer,dimension(:),allocatable           :: freq_occ
172         integer                                    :: i,j
173
174         allocate(freq_occ(size(pms_HRU)))
175
176         !find the frequency of occurance of all parameter values
177         DO i = 1,size(pms_HRU)
178             freq_occ(i) = 0
179             DO j = 1,size(pms_HRU)
180                 IF (pms_HRU(i)==pms_HRU(j)) THEN
181                     freq_occ(i) = freq_occ(i)+1
182                 END IF
183             END DO
184         END DO
185
186         !return parameter value that occurs most frequently
187         DO i = 1,size(pms_HRU)
188             IF (freq_occ(i) == MAXVAL(freq_occ)) THEN
189                 upscaled_pm = pms_HRU(i)
190             END IF
191         END DO
192
193     end subroutine upscale_majority
194
195
196
197
198
199
200 end module mpr_upscaling
```

APPENDIX D

SUPPLEMENT TO RESEARCH CHAPTER TWO: METHOD USED TO DETERMINE NUMBER OF PARAMETER SAMPLES REQUIRED.

This appendix has been submitted to Water Resources Research as supplementary information to a research article. Simulations and analysis were carried out by Rosanna Lane, with guidance from all co-authors.

Citation: Lane, R.A., Freer, J. E., Coxon, G., & Wagener, T. (in review). Incorporating Uncertainty into Multiscale Parameter Regionalisation to Evaluate the Performance of Nationally Consistent Parameter Fields for a Hydrological Model. Submitted to Water Resources Research.

When calibrating DECIPHeR using MPR, there are 24 global parameters which need to be calibrated but many of these may not be sensitive for all catchments. To explore parameter uncertainties, we decided to calibrate the global parameters using a Monte-Carlo calibration strategy. Upper and lower bounds were first defined for each global parameter, with values selected to prevent parameter values from becoming unrealistic, whilst being as wide as possible to enable the feasible parameter space to be fully sampled. N simulations could then be carried out to randomly sample the global parameters between these upper and lower bounds. However, the number of samples to evaluate (N) is a trade-off between the computational time it takes to run additional samples and the added value of having extra simulations and thus more densely sampling the parameter space. We therefore carried out a bootstrapping test, to determine the number of model evaluations needed to converge on stable model performance scores.

The objective of the bootstrapping test was to explore:

Gauge	Area (km^2)	BFI	Mean annual Rain (mm)	Mean annual runoff (mm)	Station elevation (m)
12005	110	0.51	1316	1071	201
83010	73	0.29	1450	1028	61
25012	25	0.24	1593	1245	374
26008	42	0.86	678	163	10
64006	47	0.48	1527	876	15
42004	1040	0.94	818	341	10

FIGURE D.1. Location and selected characteristics of the 6 study catchments. Dots show the location of the gauging station with the size indicating the relative area of each catchment. Catchments are coloured by their BFI, and other catchment information is given in the corresponding table (catchments are organised from north to south within the table).

1. How many model evaluations (i.e. number of samples of the global parameters) do we need before we converge on a stable model performance score?
2. When we have converged on a stable KGE score, are the components of the KGE score also stable?

D.1 Bootstrapping procedure and convergence criteria

Six catchments were selected for the bootstrapping test, to explore intra-catchment differences. They were selected to be geographically spread across Great Britain and have a range of baseflow index values and catchment sizes. Catchment locations and characteristics are given in Figure D.1. For each catchment the bootstrapping procedure was as follows:

1. 10,000 model evaluations were carried out for the catchment, sampling the global parameters.
2. The non-parametric KGE (Pool et al., 2018) was calculated for each of these evaluations, following a 365 day warmup.
3. Smaller random samples of n simulations were taken from the initial 10,000 model evaluations, and the maximum KGE from the n simulations was calculated. This was repeated for values of n from 1 to 3000.
4. Step 3 was repeated 30 times, to show the range in maximum KGE between samples of the

same size.

The aim was to find the sample size at which the model performance converged, where convergence was defined as the model performance remaining the same (or changing to a very small degree) when using a different sample of model evaluations of equal or larger size. To assess convergence, we calculated the 5^{th} to the 95^{th} percentile confidence bounds of the distribution obtained by bootstrapping at each sample size. When this width is small, it indicates that repeat samples of the same sample size resulted in the same model performance, and so we had reached convergence. This was evaluated both visually and by setting a convergence criteria of 0.015 (i.e. when the differences in the sample best KGE value are less than 0.015 between repeats of the same sample size, it is assumed that we have converged).

D.2 Outcome

Figure D.2 shows how the maximum modelled KGE changed for each catchment in relation to the number of model evaluations. For each sample size (n), 30 repeats were carried out. Light grey markers show the results for each of these repeats, whilst the black line shows the median value from all repeats. Across all 6 catchments, sample size has the most impact on model performance score up to a sample size of around 1000. Above 2000 samples, the model performance scores appear stable for most catchments.

Gauge 26008 stands out as taking the longest to converge, and even when the maximum KGE stabilises at a sample size of around 2000, there is still a wide variation between repeat samples. This is a small catchment, with relatively low mean annual rainfall and a high BFI. Over 50% of the catchment is overlaying high permeability bedrock. This could result in different global parameters becoming sensitive for this catchment, as MPR is setup to add additional transfer equation coefficients for areas overlaying highly productive bedrock.

Figure D.3 and Figure D.4 show how the range between repeat samples changed in relation to the number of model evaluations for each catchment. This is looking at the similarities between the repeats of a given sample size, with a value of 0 indicating that all random samples of a given size gained the same KGE value and therefore we have fully sampled the parameter space. Values below 0.015 have been shaded in a darker colour in Figure D.4, to indicate where the convergence criterion has been met. These plots support the visual interpretation of Figure D.2, demonstrating that for samples of 1500 or more simulations there is very little difference between repeats for 5 out of the 6 test catchments.

Figure D.5 onwards show the model performance for the 3 components of KGE, when selecting the parameter set based on the combined KGE value. These show that whilst the KGE value may

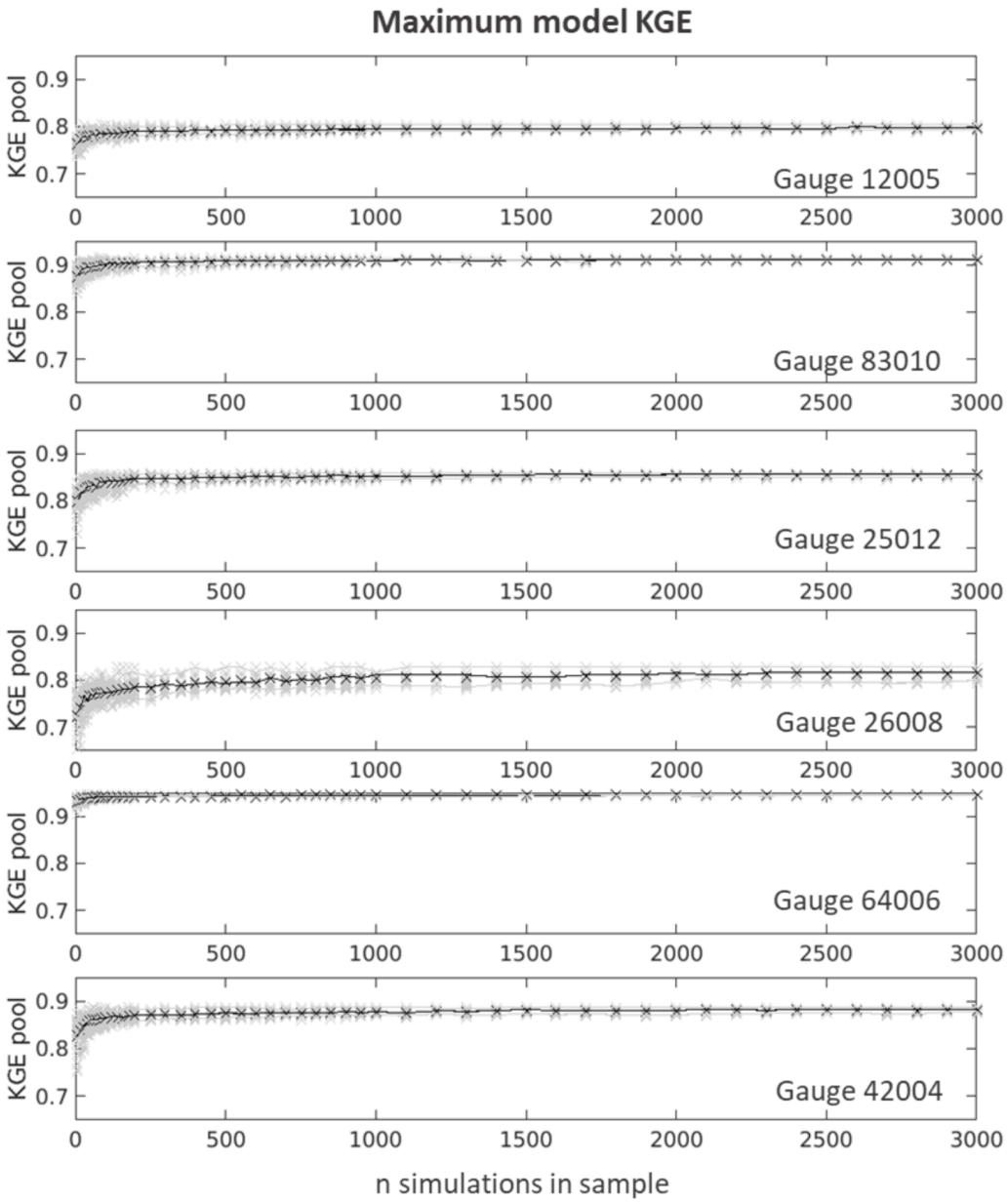


FIGURE D.2. Best modelled KGE gained from different sized samples of simulations. X axis shows the number of simulations in the sample, with these being randomly taken from a total sample of 10,000 simulations. The y axis shows the maximum KGE value obtained by any simulation in the sample. There are 30 repeats for each sample size – each individual repeat is marked in grey, and the median of all repeats for a given sample size is plotted in black. The grey lines show the 5th and 95th percentile across all repeats for each sample size, whilst the black line shows the median. Each plot shows results for a different gauge.

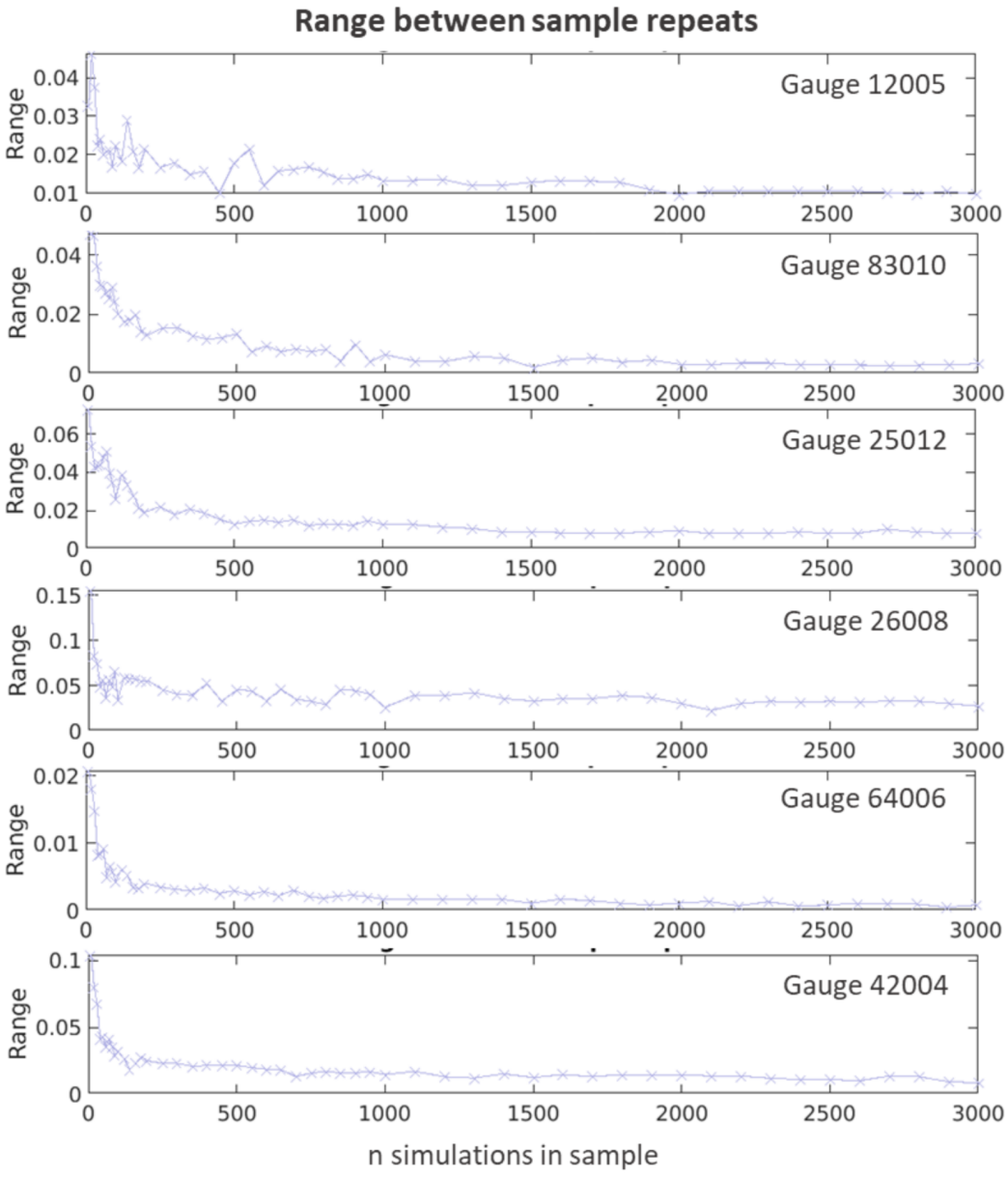


FIGURE D.3. Variability between KGE values obtained by repeat simulations of the same sample size. X axis shows the number of simulations in the sample, with these being randomly taken from a total sample of 10,000 simulations. For each sample size, 30 random samples were taken from the original 10,000 simulations and the maximum modelled KGE value was recorded. The variation between these KGE values, calculated as the 5th-95th percentile range, is given on the y axis. Each plot shows results for a different gauge. Note, the scale of the y axis differs for each catchment, so that the general shape of the graph can be seen.

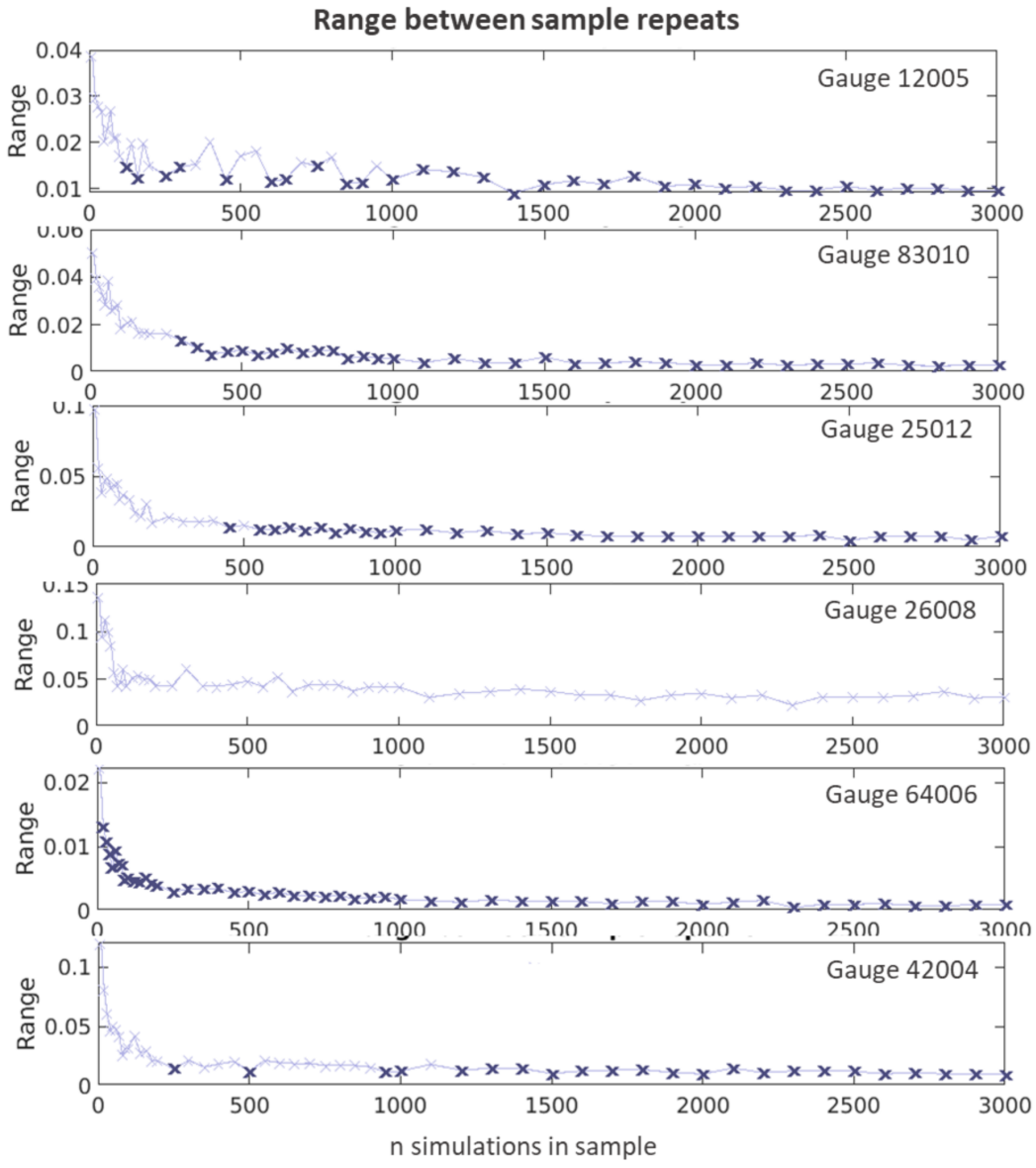


FIGURE D.4. As Figure D.3, except values with a range below the convergence criteria of 0.015 have been highlighted in a darker colour.

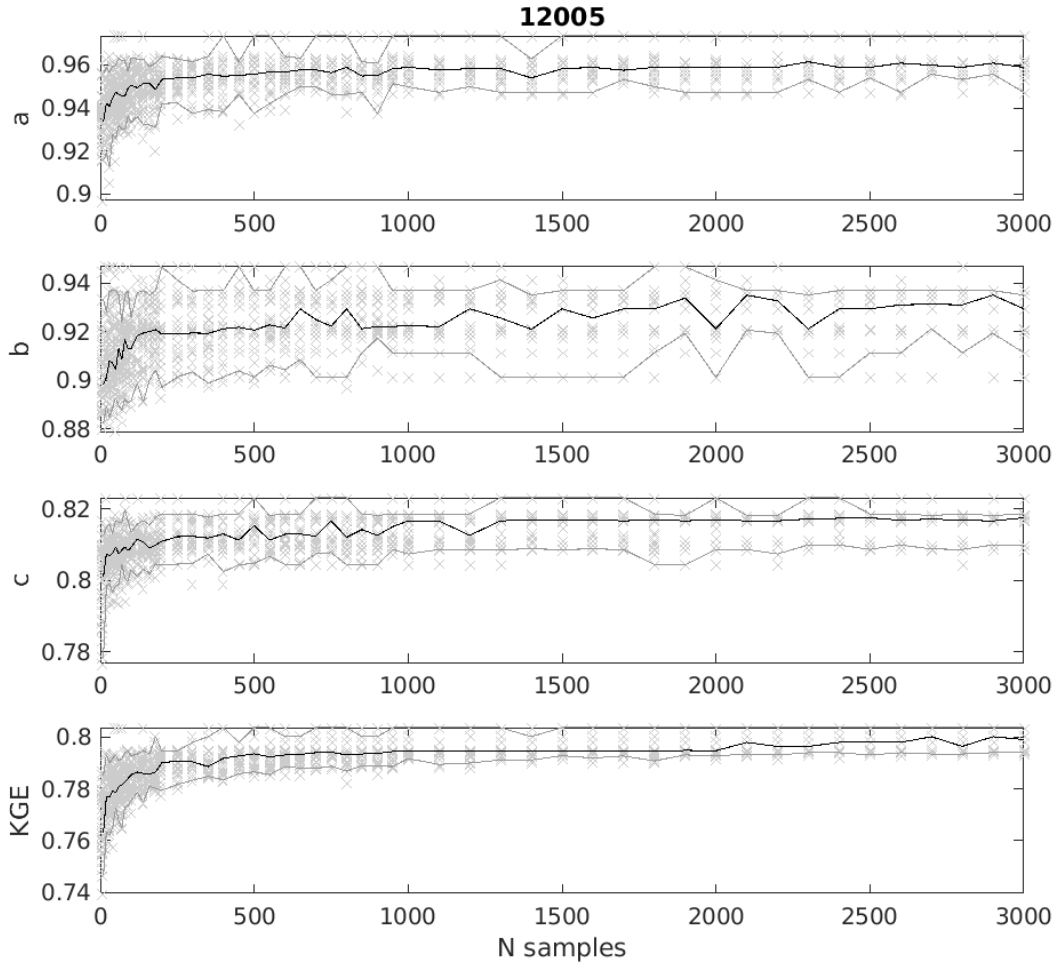


FIGURE D.5. Best modelled KGE gained from different sized samples of simulations, and the values of the decomposed metrics forming that KGE, for gauge 12005.The x axis shows the number of simulations in the sample, with these being randomly taken from a total sample of 10,000 simulations. The y axes show the best value of KGE gained within that sample (bottom plot), and the decomposed metrics that make up that KGE value (top 3 plots). A = error in the slope of the flow duration curve. B = model bias. C = spearman correlation coefficient.

increase with the number of samples, there are larger ranges in the scores of the decomposed metrics. Different trade-offs between these three decomposed metrics may be contributing to the highest KGE scores.

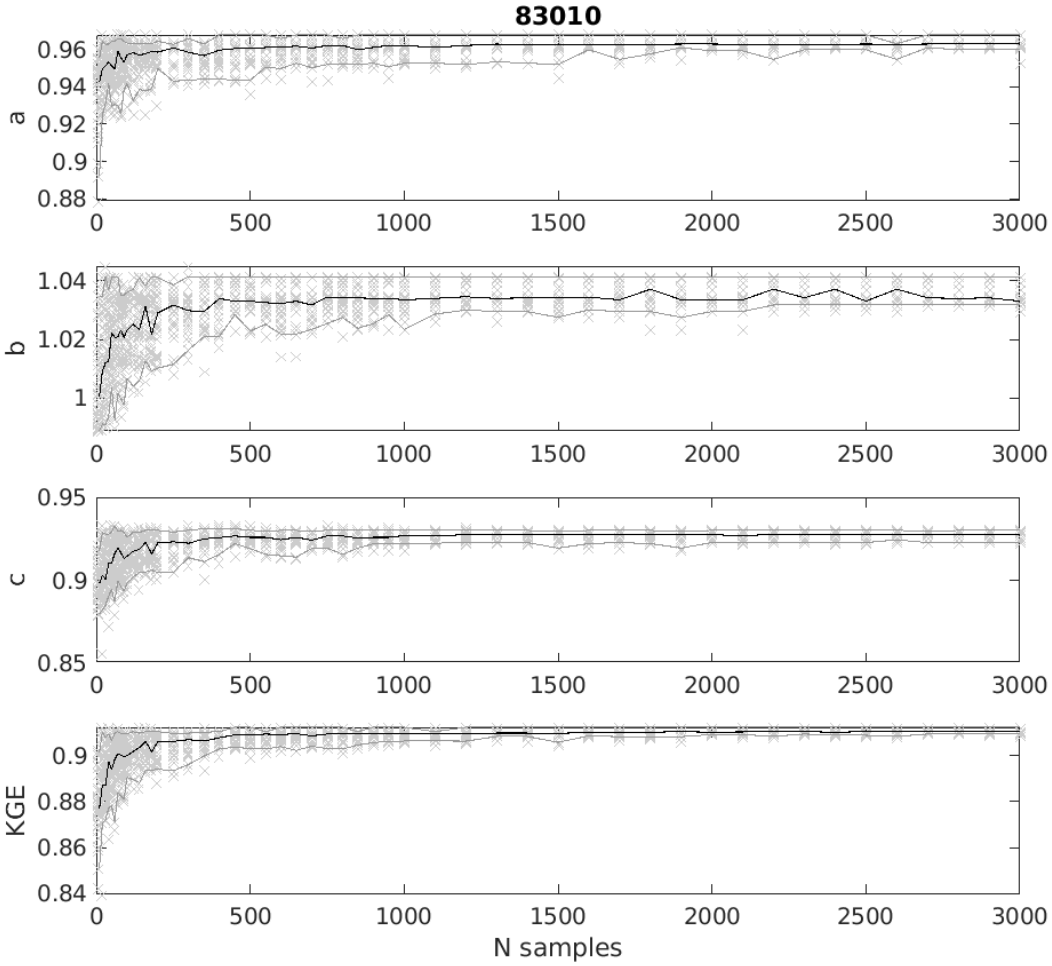


FIGURE D.6. As for Figure D.5, but for gauge 83010.

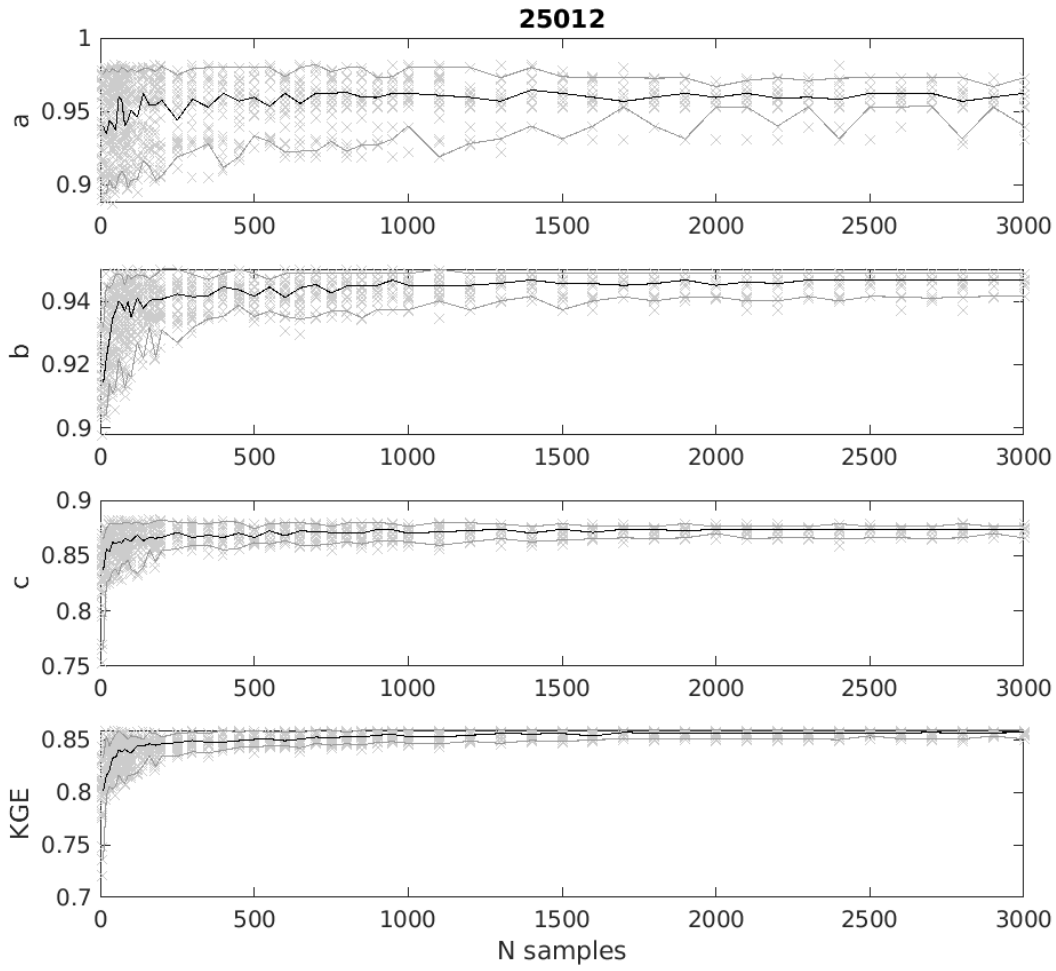


FIGURE D.7. As for Figure D.5, but for gauge 25012.

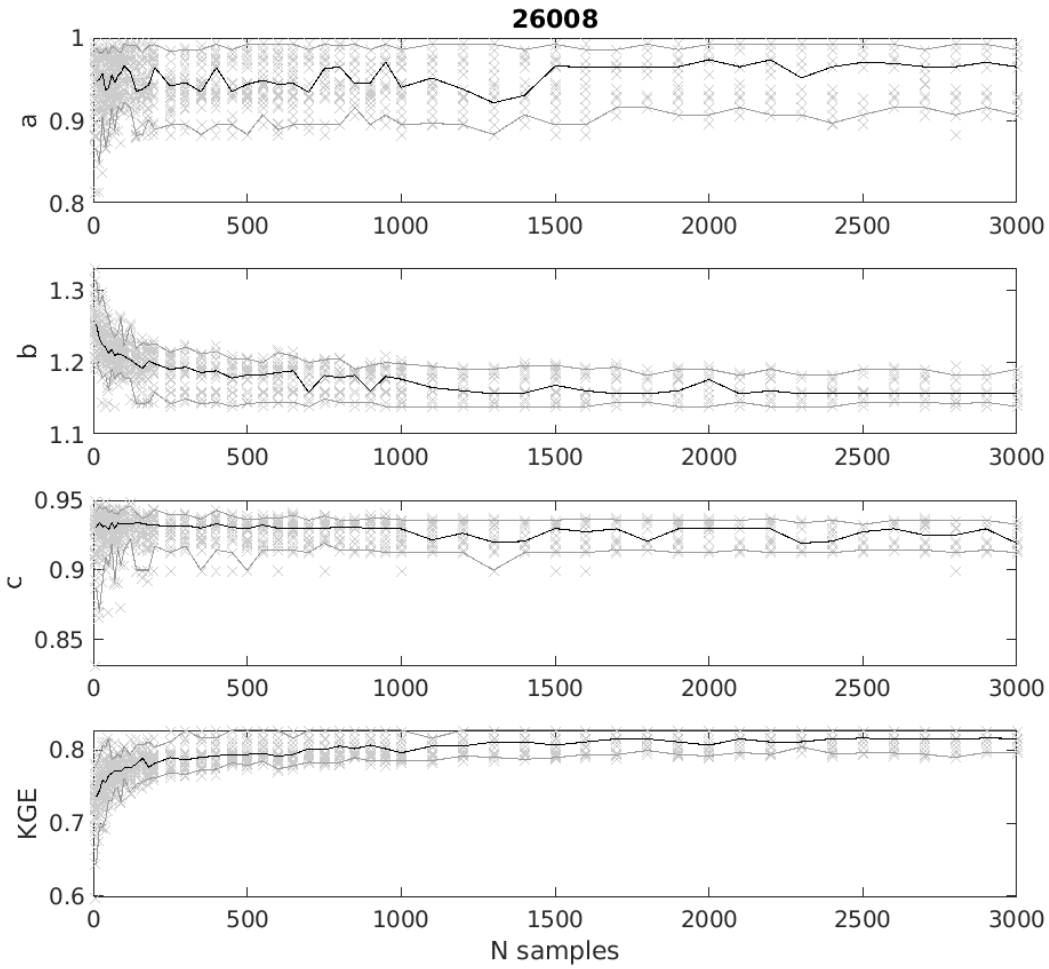


FIGURE D.8. As for Figure D.5, but for gauge 26008.

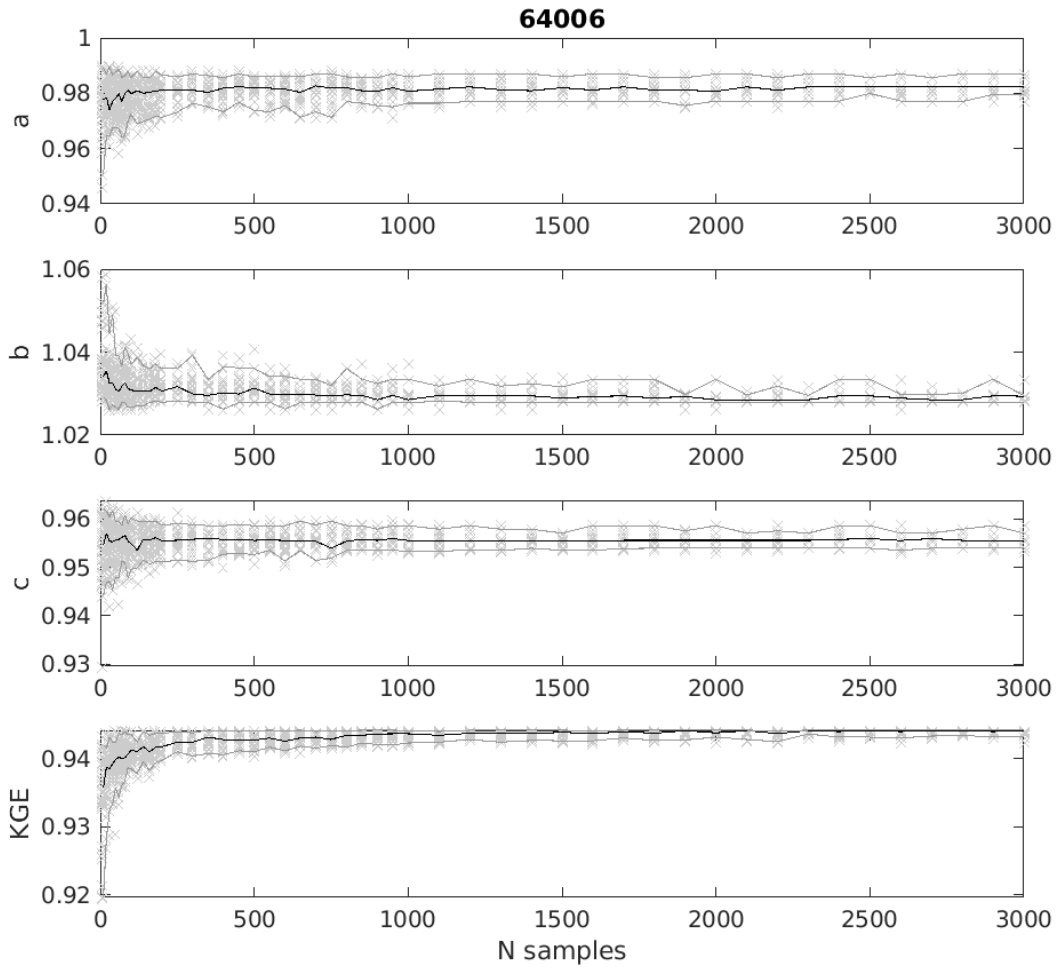


FIGURE D.9. As for Figure D.5, but for gauge 64006.

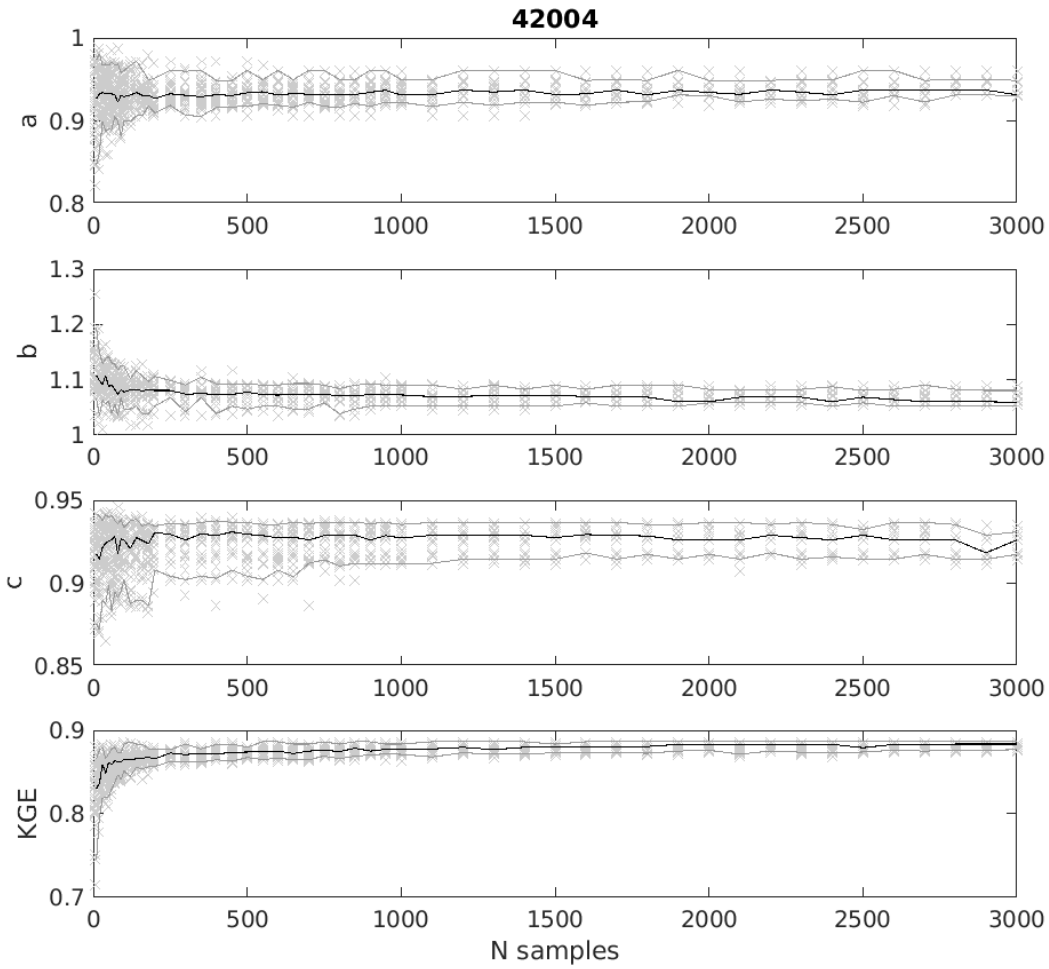


FIGURE D.10. As for Figure D.5, but for gauge 42004.

APPENDIX E

SUPPLEMENT TO RESEARCH CHAPTER TWO: EXTENDED DESCRIPTION OF ADDING HYDROGEOLOGY DATA TO THE MPR APPROACH

This appendix describes initial testing which led to the decision to incorporate hydrogeology data into MPR for the DECIPHeR model across Great Britain (GB). This expands upon the brief description given in research chapter two.

Initially, MPR was applied across 221 principal basins without any incorporation of hydrogeology data. Model performance with MPR parameter fields was compared to model performance with Monte-Carlo (MC) constrained parameter values. As shown in Figure E.1, catchments with a large fraction of highly productive hydrogeology often performed very poorly. The MPR parameter fields could not produce any parameter sets which could simulate peak flows or the slope of the hydrograph falling and rising limbs correctly. For some catchments neither the MPR nor the Monte-Carlo constrained parameters could produce reasonable simulations, suggesting that the model structure was not suitable for these catchments. As we were focused on model parameterisation, and not model structural improvements, we did not further investigate these catchments. However, there were also areas with high productivity geology where MC parameters produced good simulations whilst MPR (national and catchment) could not – these were our focus as the good performance of the MC parameters demonstrated that good model performance should be possible with the given model structure.

We focused on a medium sized (1040 km^2) example catchment, the Test at Broadlands (42004), which had a large fraction (66%) of high productivity geology (see Figure E.2a-b). This included 4 sub-catchments, with varying fractions of high productivity geology (16% – 96%). We first

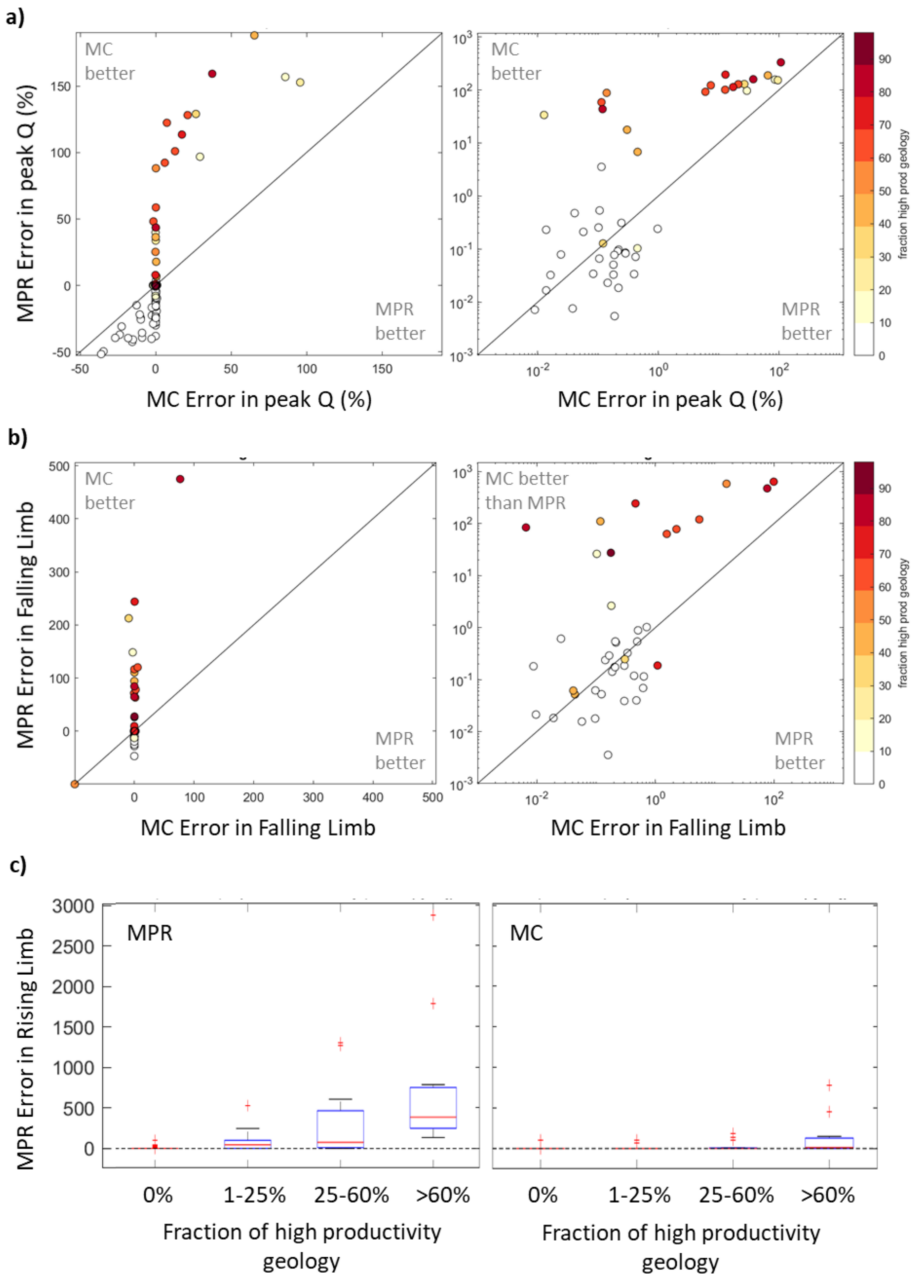


FIGURE E.1. Performance of MPR (with parameters constrained for individual catchments) and Monte-Carlo sampled model parameters before geology was included in the MPR approach. A) Comparison of peak flow errors between the MPR approach and Monte-Carlo approach, from the best model simulation. Graphs are given in linear (left) and log (right) scales, with points coloured to highlight the fraction of highly productive hydrogeology in each catchment. B) as for A but showing percentage error in the slope of the falling limb. C) Boxplots show range in the percentage error in the rising limb, for catchments grouped by fraction of high productivity geology.

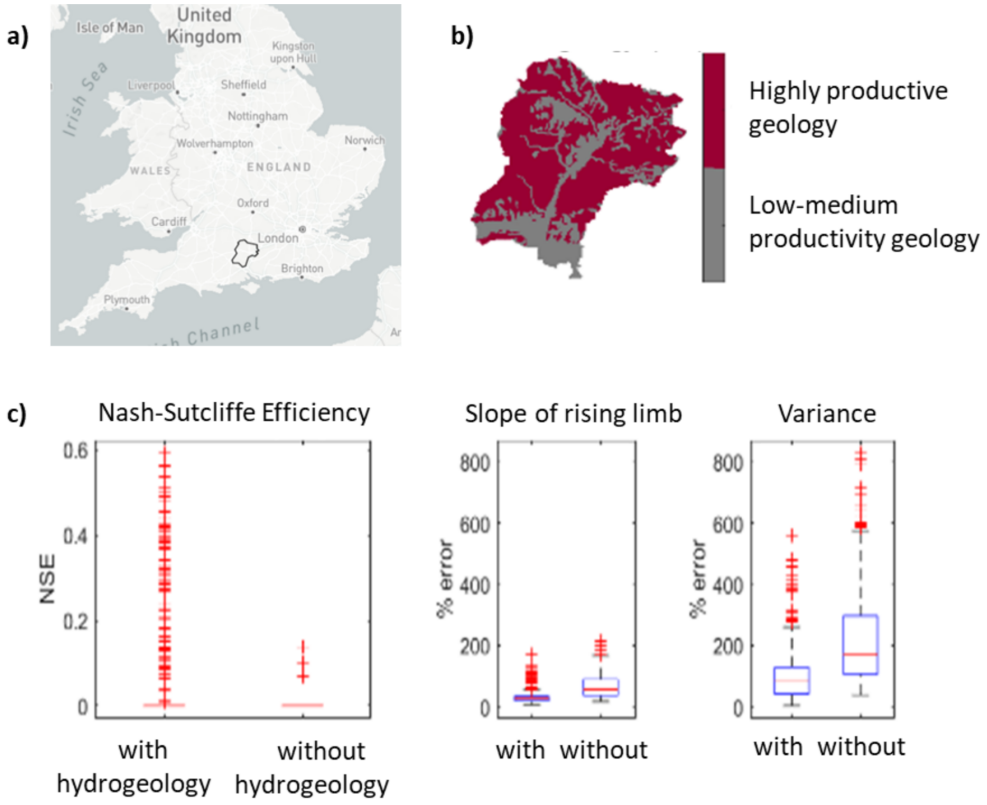


FIGURE E.2. DECIPHeR performance for the Test at Broadlands, when run using MPR with and without hydrogeology data. a) catchment location. b) Catchment map showing areas covered by highly productive geology. c) Model performance metrics evaluated from 500 runs with and without the hydrogeology data.

analysed the catchment-average parameter ranges which were being sampled using the MPR method, and compared this to i) the normal model parameter bounds used for GB modelling, and ii) the behavioural parameter bounds found for the catchment by Monte-Carlo sampling (see Figure E.3). This highlighted that the current implementation of the MPR approach was not able to produce sufficiently high $\ln(T_0)$ and Smax parameter values to produce good simulations in this catchment. The increasing of $\ln(T_0)$ and Smax for areas overlaying highly productive geology made conceptual sense: it allows for larger groundwater stores with faster transport of groundwater. We therefore added modifications to the MPR transfer function equations for areas with highly productive geology.

To test the addition of hydrogeology data in the MPR process, we ran simulations with and without hydrogeology for the Test at Broadlands. For each case, 500 simulations were carried out, sampling between the global parameter bounds. As shown in Figure E.2c, before adding the hydrogeology data, none of the simulations could reach a Nash-Sutcliffe Efficiency (NSE) of

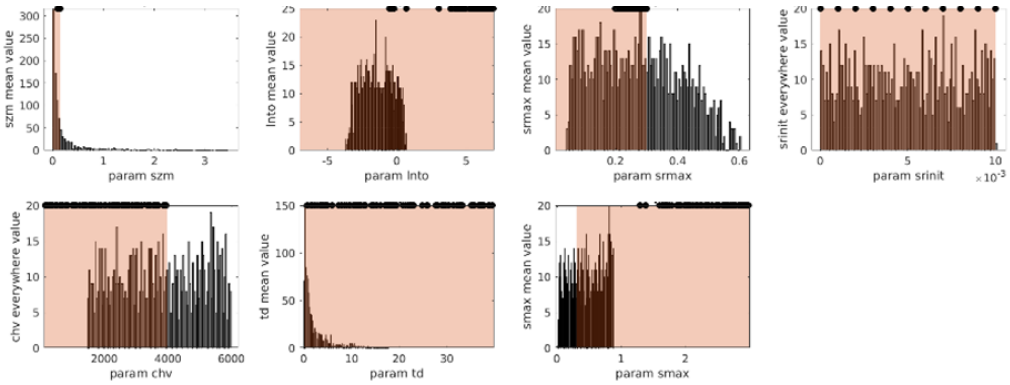


FIGURE E.3. DECIPHeR parameter bounds for the Test at Broadlands (42004). Histogram shows distribution of mean parameter value over whole catchment from 1000 MPR runs. Dots show parameter values from the top 100/10,000 Monte Carlo runs (i.e. behavioural parameter ranges). The shaded area gives the normal Monte Carlo parameter range (i.e. the parameter range used when constraining the behavioural parameter ranges).

0.2. After adding the hydrogeology data, simulations up to NSE 0.6 were achieved. The addition of modified transfer functions resulted in reduced simulation errors, and better representation of the hydrograph slope and variance. As shown in Figure E.4, these improvements were seen in all sub-catchments, especially 42005 which had the largest fraction of high productivity geology.

Further testing showed that addition of hydrogeology into the MPR process improved performance in many areas overlying high permeability geology, whilst not resulting in large changes in model performance elsewhere. We therefore included this into the methodology for Great Britain.

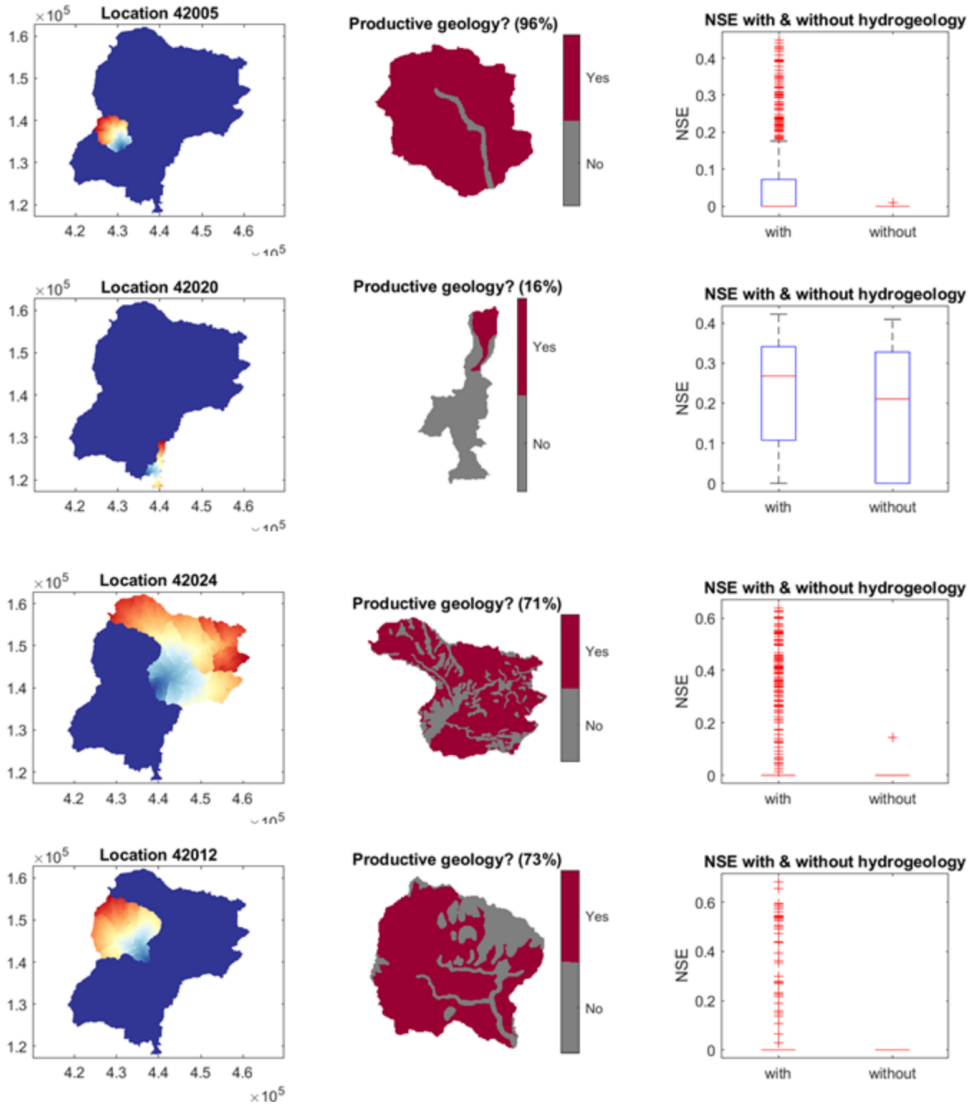


FIGURE E.4. Performance of DECIPHeR-MPR with and without hydrogeology for sub-catchments within The test at Broadlands (42004). Each row gives a different sub-catchment, with columns showing i) sub-catchment location within 42004, ii) the sub-catchment area overlying high productivity geology, iii) performance with and without the addition of hydrogeology to the MPR process.

APPENDIX F

SUPPLEMENT TO RESEARCH CHAPTER THREE: BIASES IN UKCP18 RCM PROJECTIONS

An evaluation of the biases in the UKCP18 RCM projections relative to observed data was carried out, to inform selection of an appropriate bias correction technique. This focused on biases in daily and seasonal precipitation/ PET, heavy precipitation, and PET quantiles, as these were considered most important for the median and higher flow statistics used in Research Chapter Three.

F.1 Mean daily rainfall bias

Figure F.1 gives the observed mean daily precipitation across Great Britain, as a reference. There is a west-east gradient in mean daily rainfall, with generally reduced rainfall as you move east across the country. The highest daily rainfall can be seen in western Scotland.

Figure F.2 and Figure F.3 show the biases in UKCP18 projections for mean daily precipitation over the baseline period. It can be seen that there is a general trend to smooth the pattern of precipitation – with RCMs underpredicting mean precipitation in the wettest areas (west Scotland, and in some cases south-west England and Wales) and overpredicting in the driest areas (along the east coast and south-east England). Most RCMs show a similar spatial pattern of bias, although there are clear differences between RCM runs. There is a general trend towards all RCMs overpredicting mean annual rainfall by around 3 - 35% (the median across all catchments in Figure F.3).

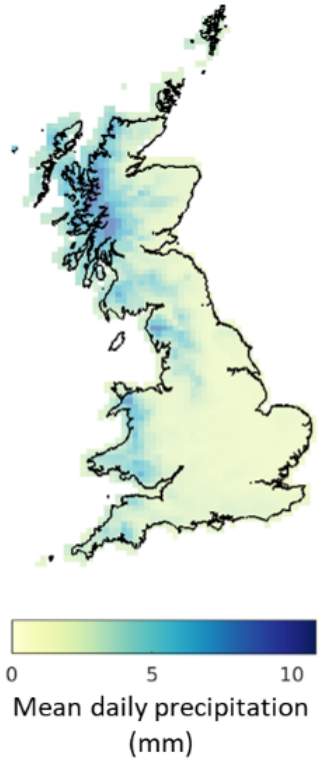


FIGURE F.1. Observed mean daily precipitation across the UK.

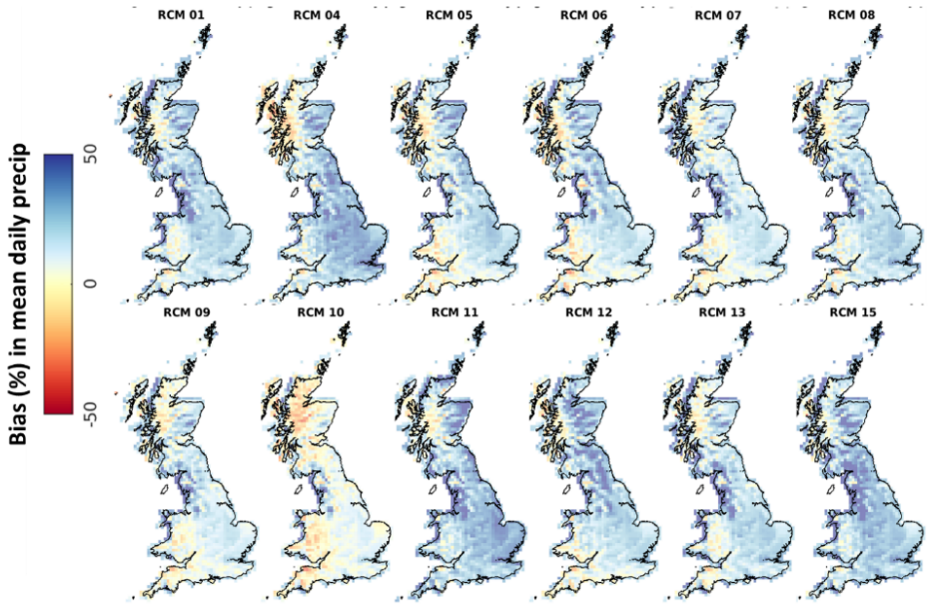


FIGURE F.2. Percentage difference in mean daily rainfall between observations and each ensemble member of UKCP18.

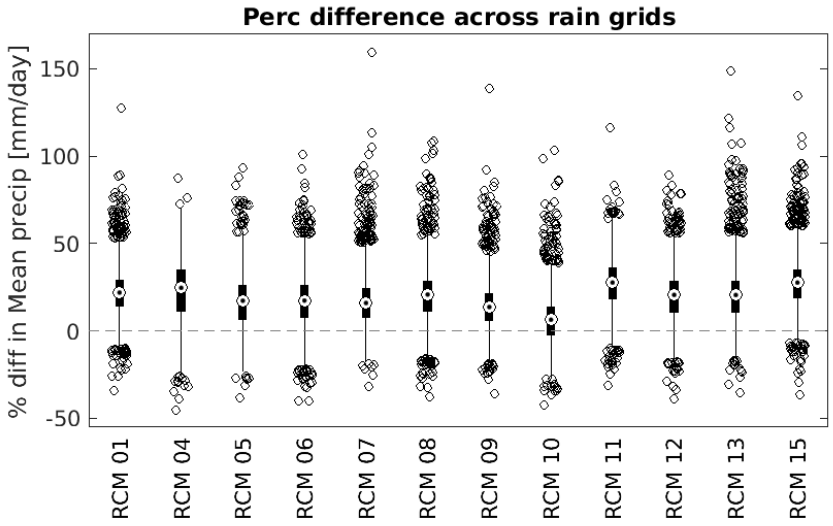


FIGURE F.3. Percentage difference in mean daily rainfall between observations and each ensemble member of UKCP18. Boxplots show the distribution of results across all land grid cells.

F.2 Seasonal rainfall bias

Figure F.4 shows the percentage changes in seasonal rainfall. Figure F.5 highlights the month of the year with the largest biases from each of the UKCP18 ensemble members. The figures show that RCMs tend to overpredict rainfall in winter and spring across most of Great Britain, except for west Scotland where rainfall is generally underpredicted. A more complex pattern of bias is seen for summer and autumn rainfall. In the summer there is an overall trend towards overprediction in Scotland, with a more mixed picture for England and Wales. Autumn rainfall biases tend to be smaller, and spatial patterns of bias differ between the RCMs.

F.3 Heavy rainfall bias

Biases in RCM projections of heavy rainfall percentiles (80th percentile rainfall – 99th percentile rainfall) are given in Figure F.6 and Figure F.7. The RCMs tend to overestimate rainfall generally, with the tendency to underpredict in areas of high rainfall (east Scotland and east Wales). Across Great Britain biases in projections of the 80th percentile rainfall are around -40 to 75%, whilst biases in projections of the 99th percentile rainfall tend to be smaller at around -40 to 40%.

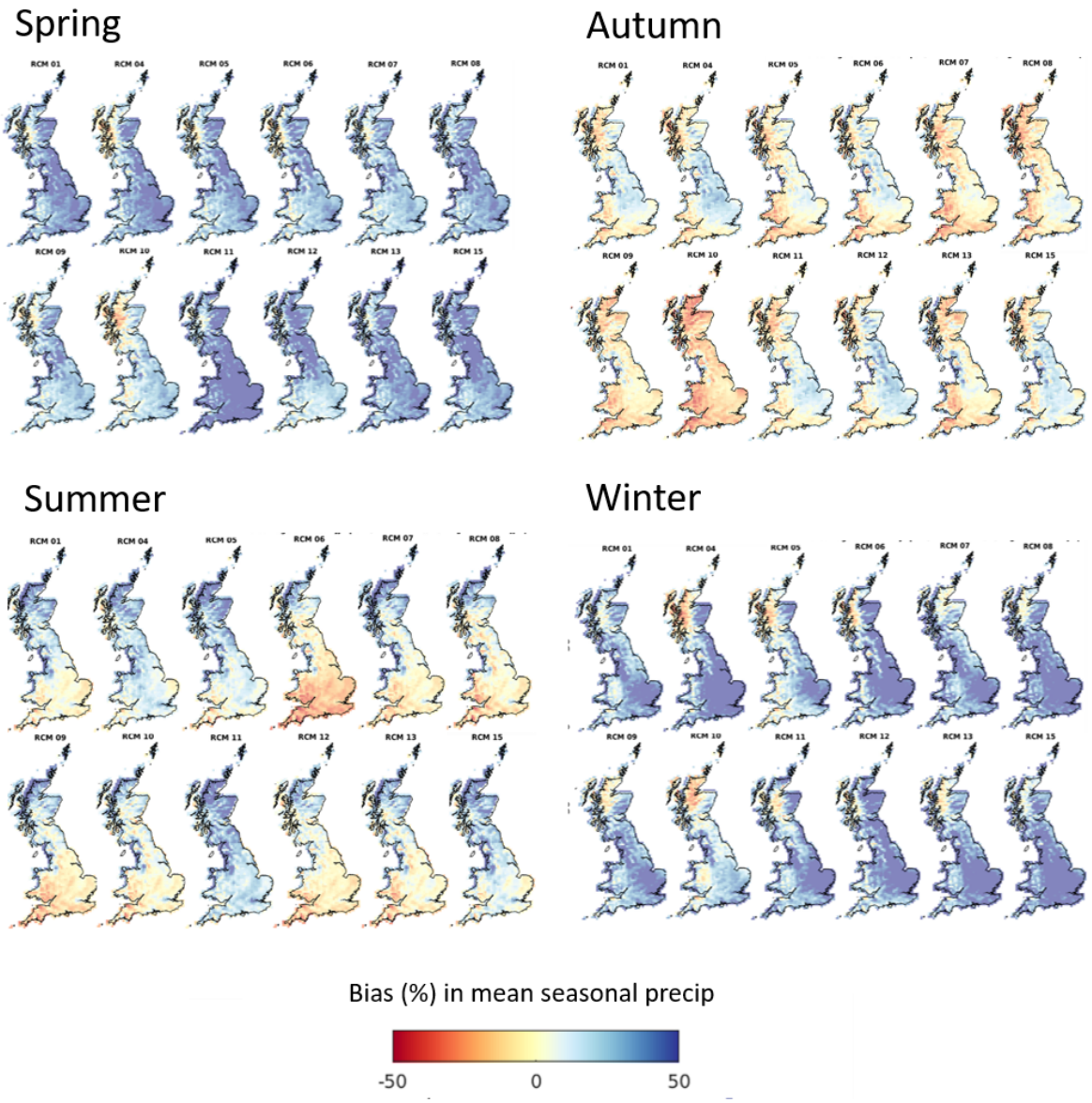


FIGURE F.4. Seasonal rainfall biases. Each plot shows the percentage difference in seasonal average rainfall between a UKCP18 ensemble member and observations over the baseline period.

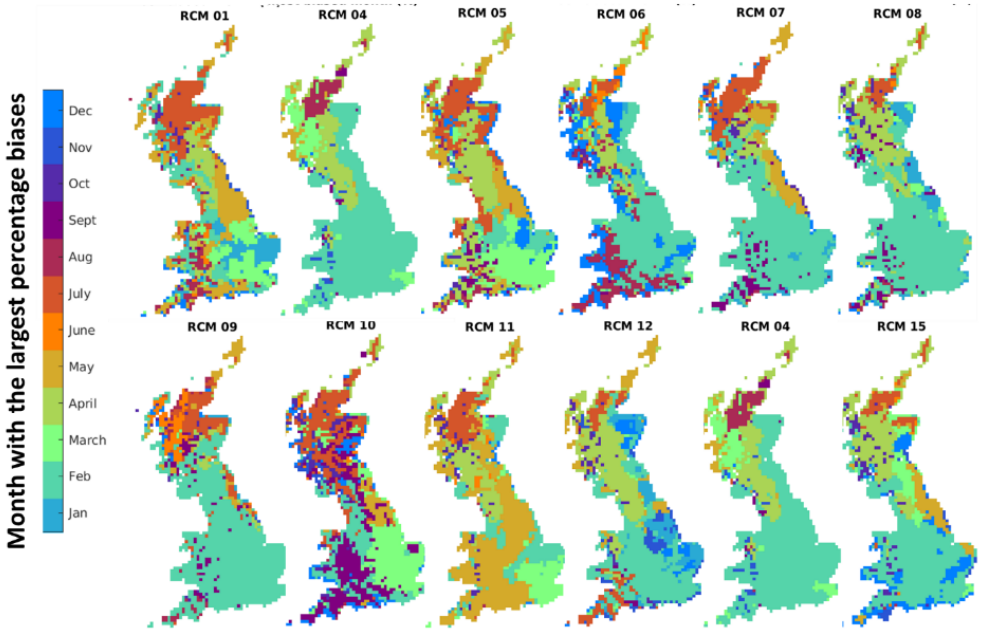


FIGURE F.5. Month with the largest rainfall biases. Each plot shows the month with the largest percentage difference in rainfall between a UKCP18 ensemble member and observations over the baseline period.

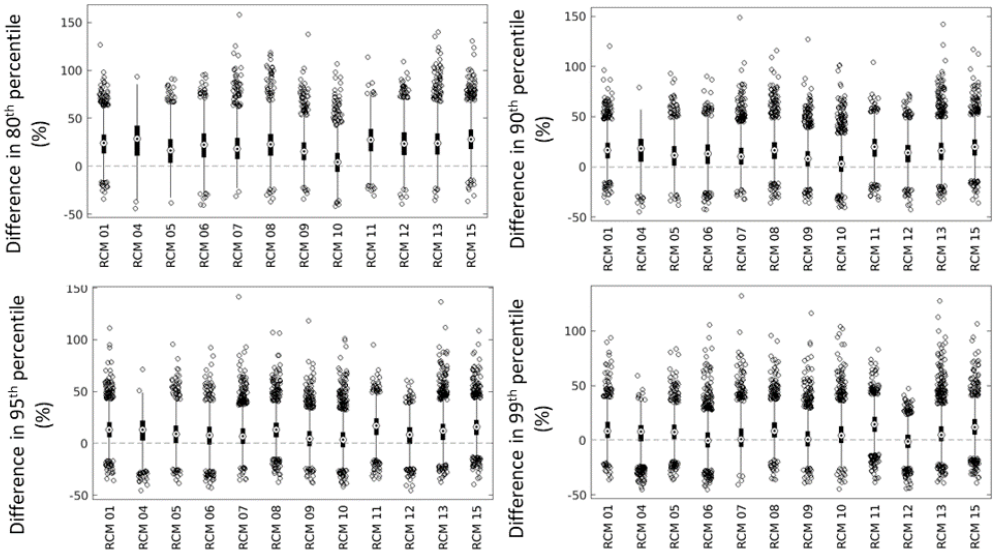


FIGURE F.6. Boxplots showing rainfall biases for 80th percentile rainfall (top left), 90th percentile rainfall (top right), 95th percentile rainfall (bottom left) and 99th percentile rainfall (bottom right).

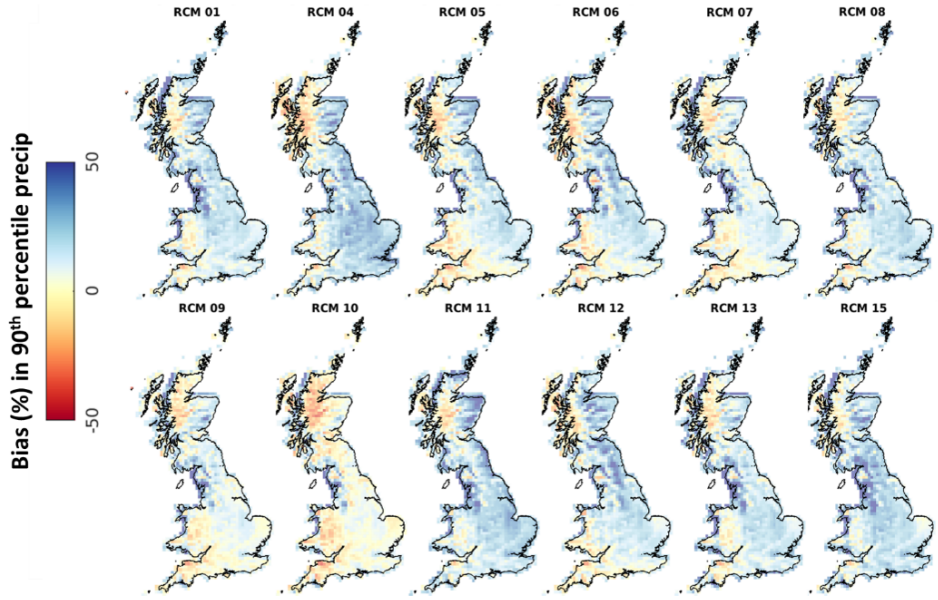


FIGURE F.7. Rainfall biases for the 90th percentile rainfall value. Each plot shows the percentage difference in 90th percentile rainfall between a UKCP18 ensemble member and observations over the baseline period.

F.4 Number of rainy days bias

On average the RCMS overpredict the number of wet days in a year (Figure F.8), with the majority of grid cells receiving an extra 0 - 55 (0 - 30%) wet days per year. RCM number 10 stands out as showing less bias than the other ensemble members.

F.5 Mean daily PET bias

Figure F.9 shows the distribution of PET over GB for each RCM, compared to observed. Figure F.10 shows maps of the percentage bias in PET for each RCM. The RCMs produce a much larger spread in PET than is seen for the observed data. The RCMs tend to overestimate PET in the southeast, where observed PET is high, and underestimate PET in Scotland, where observed PET is lower. Biases in mean daily PET are in the region of -20% to +40%.

F.6 Seasonal PET bias

Plots showing PET biases split by season are given in Figure F.11. These continue the pattern of the RCMs overestimating PET variability. summer PET is generally overestimated (up to +40%), winter PET is generally underestimated (in some areas by up to 100%). The same spatial

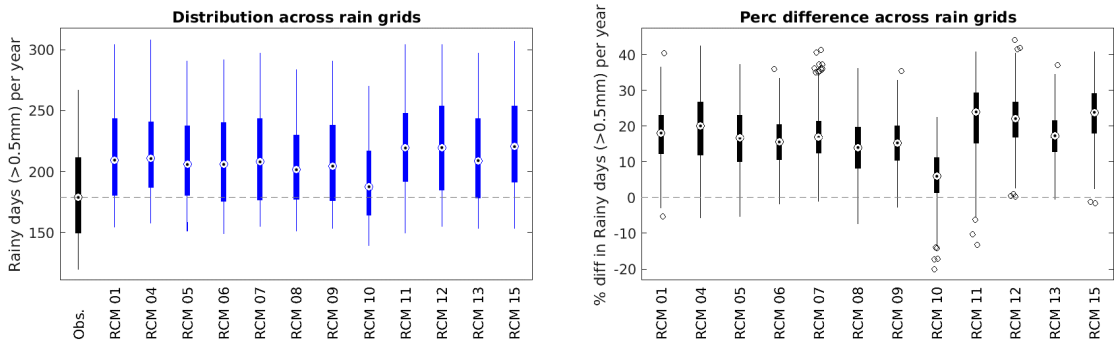


FIGURE F.8. Bias in number of rainy days (defined as a day with at least 0.5mm of rainfall). Left: distribution in number of rain days across Great Britain, from the observed rainfall data and 12 UKCP18 ensemble members. Right: percentage difference in number of rain days between each ensemble member and the observed data over the baseline period.

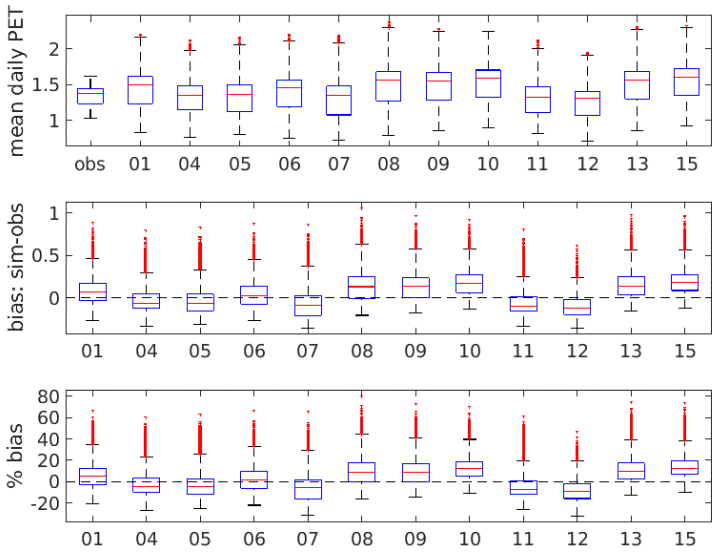


FIGURE F.9. Boxplots show distribution of observed and simulated mean daily PET across Great Britain. Top: mean daily PET distributions from the observed data and 12 RCMs. Middle: distributions of difference between observed and each RCM. Bottom: distribution of difference between observed and simulated PET, as a percentage of the observed value.

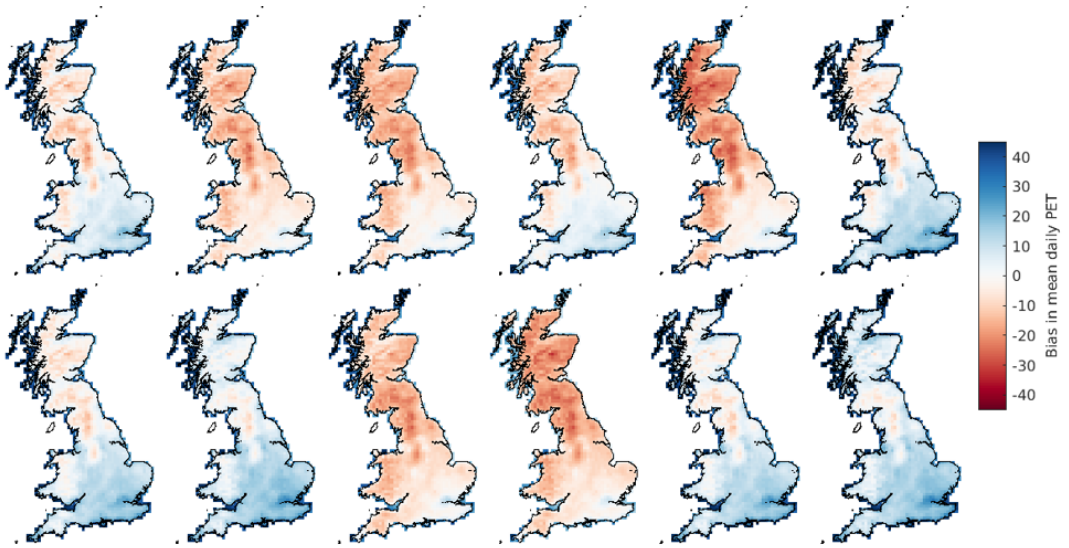


FIGURE F.10. Percentage bias in mean daily PET from each ensemble member.

pattern of RCMs underestimating PET in the north and underestimating in the southeast persists through all seasons.

F.7 Quantiles

Figure F.12 shows bias in RCM PET quantiles. Low quantiles are underpredicted, higher PET quantiles are more likely to be overpredicted.

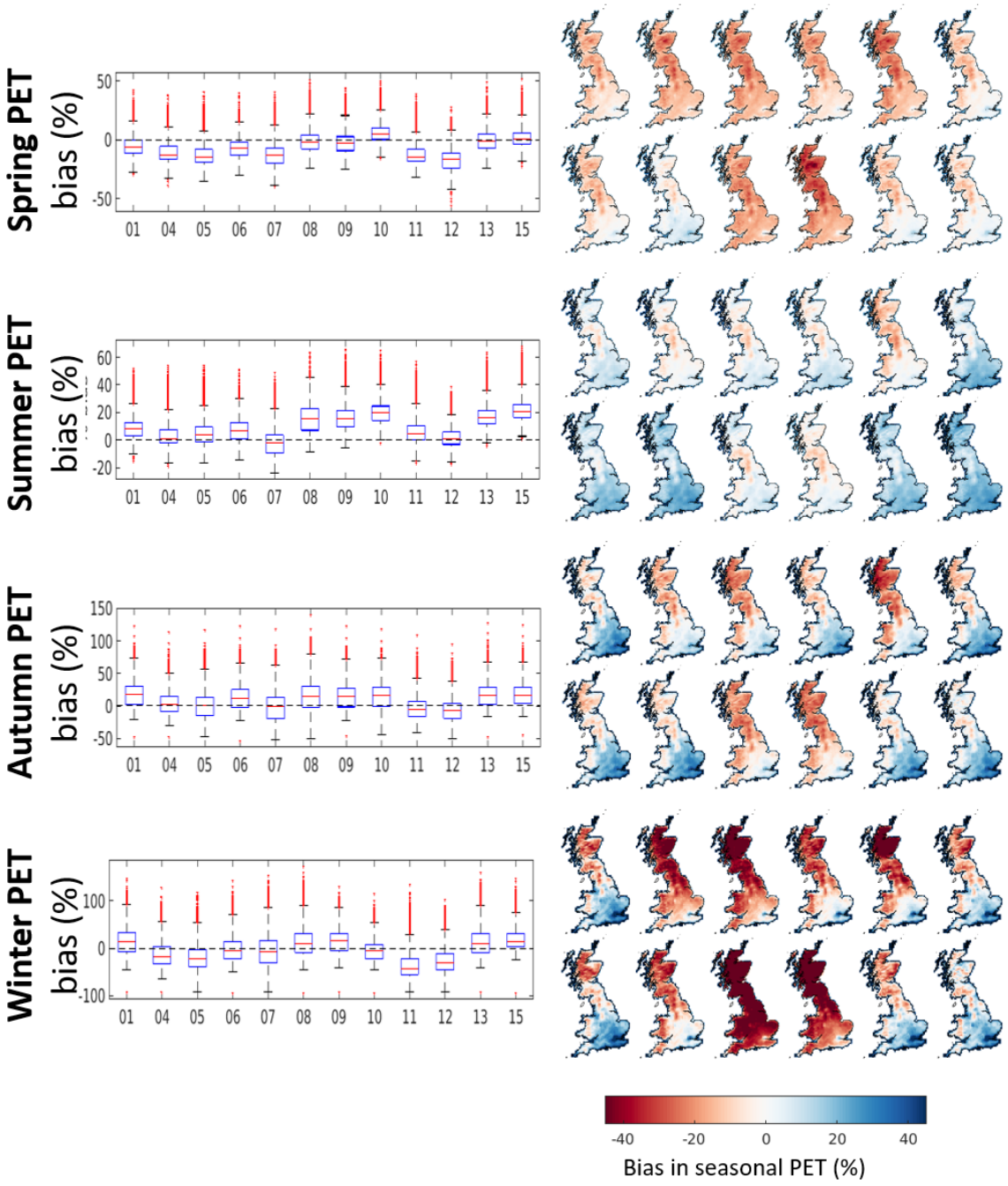


FIGURE F.11. Percentage biases in RCM PET data compared to an observed PET product. Results are split by seasons, from top row to bottom spring (March-May), summer (June-August), autumn (September-November) and winter (December-February). Left column gives boxplots showing the distribution across GB, right column shows maps of % bias for each RCM and each season.

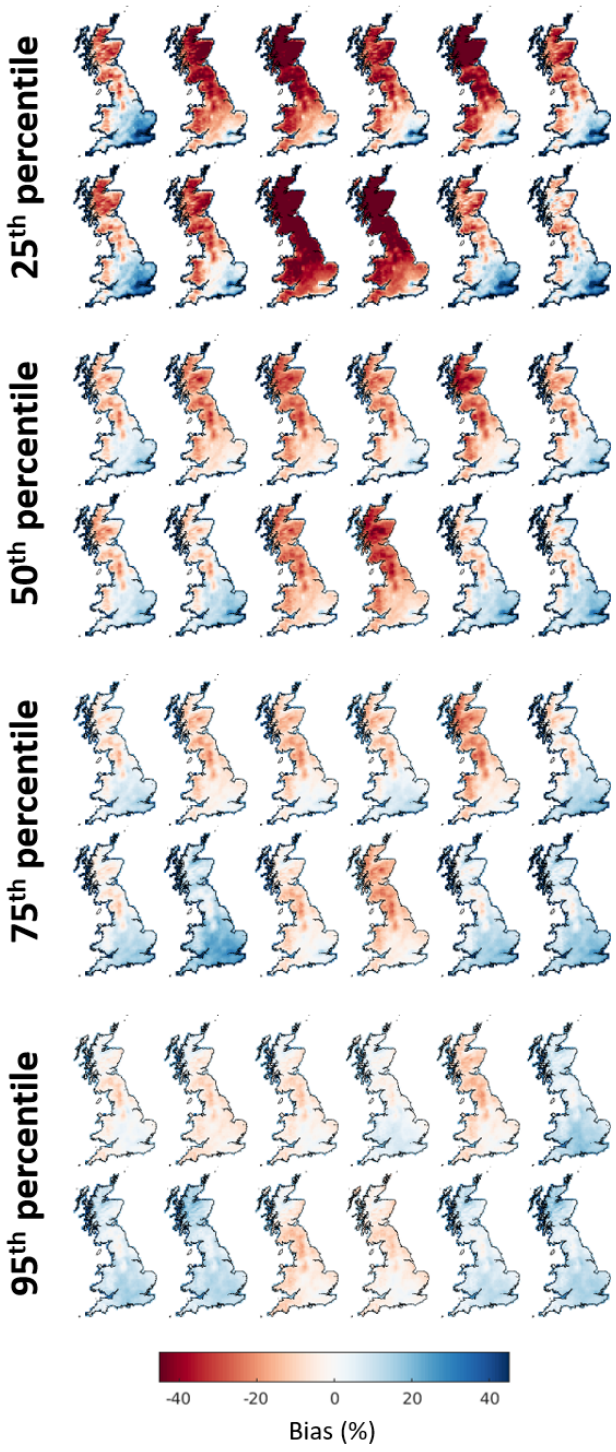


FIGURE F.12. Percentage bias in PET quantiles from each RCM.

APPENDIX G

SUPPLEMENT TO RESEARCH CHAPTER THREE: BIAS CORRECTION METHODOLOGY

G.1 Precipitation: quantile mapping

RCMs show biases in precipitation and PET (see Appendix F). The seasonal pattern of precipitation is also not simulated correctly, with RCMs overpredicting rainfall in Spring and Winter. RCMs also simulate many extra days of light rainfall compared to observations, in many cases over a month of additional wet days per year. A quantile-quantile mapping / distribution mapping bias correction method was chosen to correct for these biases (Teutschbein and Seibert, 2012).

The following steps were taken for each grid-cell and month:

1. Empirical Cumulative Distribution Functions (CDFs) were calculated for the observed precipitation, and RCM simulated precipitation for the control/baseline period (all dates where observed and simulated precipitation were available).
2. The fractional change in precipitation between the observed and control/baseline simulated was calculated for each cumulative probability.
3. The whole simulated precipitation series was then bias corrected. The cumulative probability of each precipitation value was calculated, and the value was modified by the fractional change for that cumulative probability.

Teutschbein and Seibert (2012) fit Gamma distributions to observed and simulated precipitation over the control period, and corrections use the difference between the distributions. This method was not justifiable here since the gamma function was not a good fit to the observed data. Instead, the method was modified to use the empirical cumulative distribution functions, omitting the need for a fit entirely and resulting in a more robust method suitable for all distribution shapes.

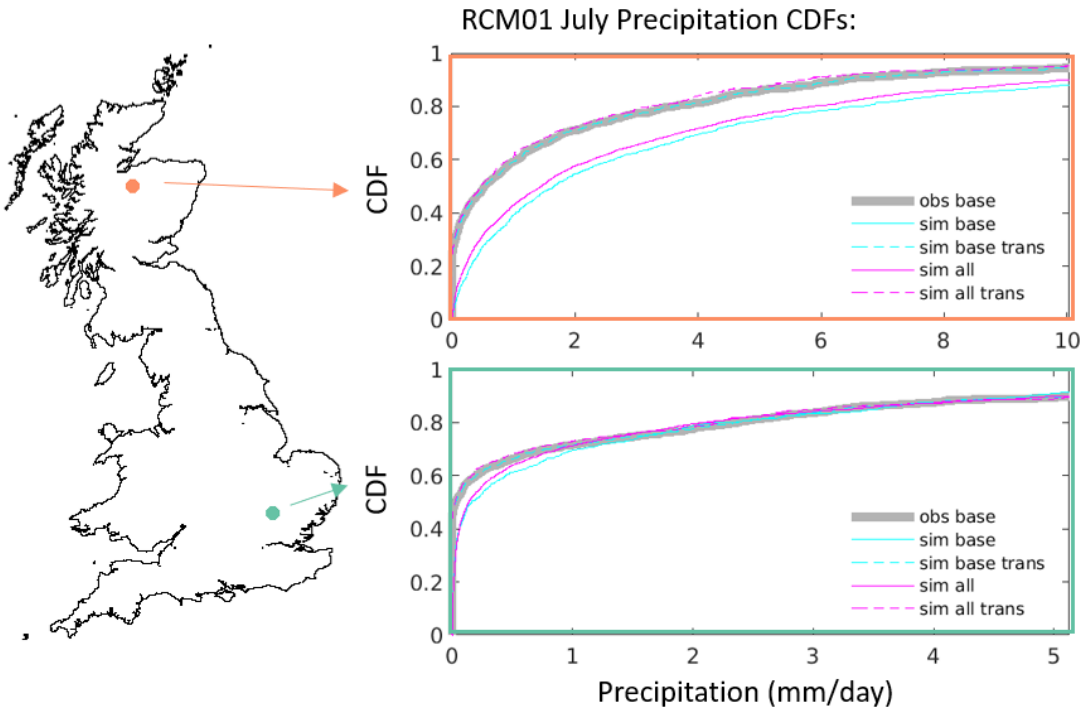


FIGURE G.1. Bias correction of July precipitation for two grid cells. CDFs are given for the observed precipitation, sim base (simulated precipitation over the baseline only), sim base trans (bias corrected simulated precipitation over the baseline), sim all (simulated precipitation over the whole timeseries) and sim all trans (corrected precipitation over the whole timeseries.

Figure G.1 shows the outcome for 2 grid cells. This shows that following bias correction the simulated precipitation CDF resembles the observed.

G.2 PET: quantile mapping

The RCMs overpredict the variance in PET. The quantile-quantile mapping bias correction technique was used to help correct for these biases. The technique used to correct precipitation in the previous section was re-applied with example results shown in Figure G.2. There were some issues when correcting very low cumulative probabilities for PET values that were close to zero (<0.1mm/day). Large fractional changes could result from dividing by values close to zero, and this resulted in unrealistically large spikes in future PET at low cumulative probabilities. To prevent this, a check was added to ensure that values at a low cumulative probability were always smaller than values at a higher cumulative probability.

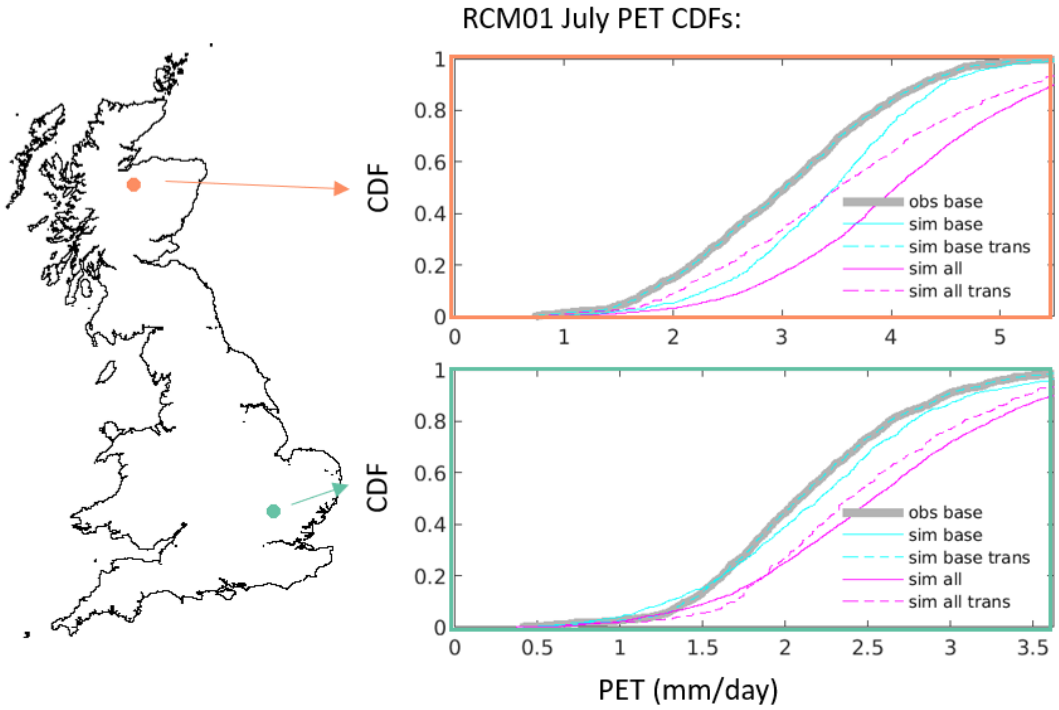


FIGURE G.2. Bias correction of July PET for two grid cells, as in Figure G.1.

G.3 Biases post-correction

Figure G.3 and Figure G.4 demonstrate how biases in precipitation and PET are reduced for the baseline period following bias correction.

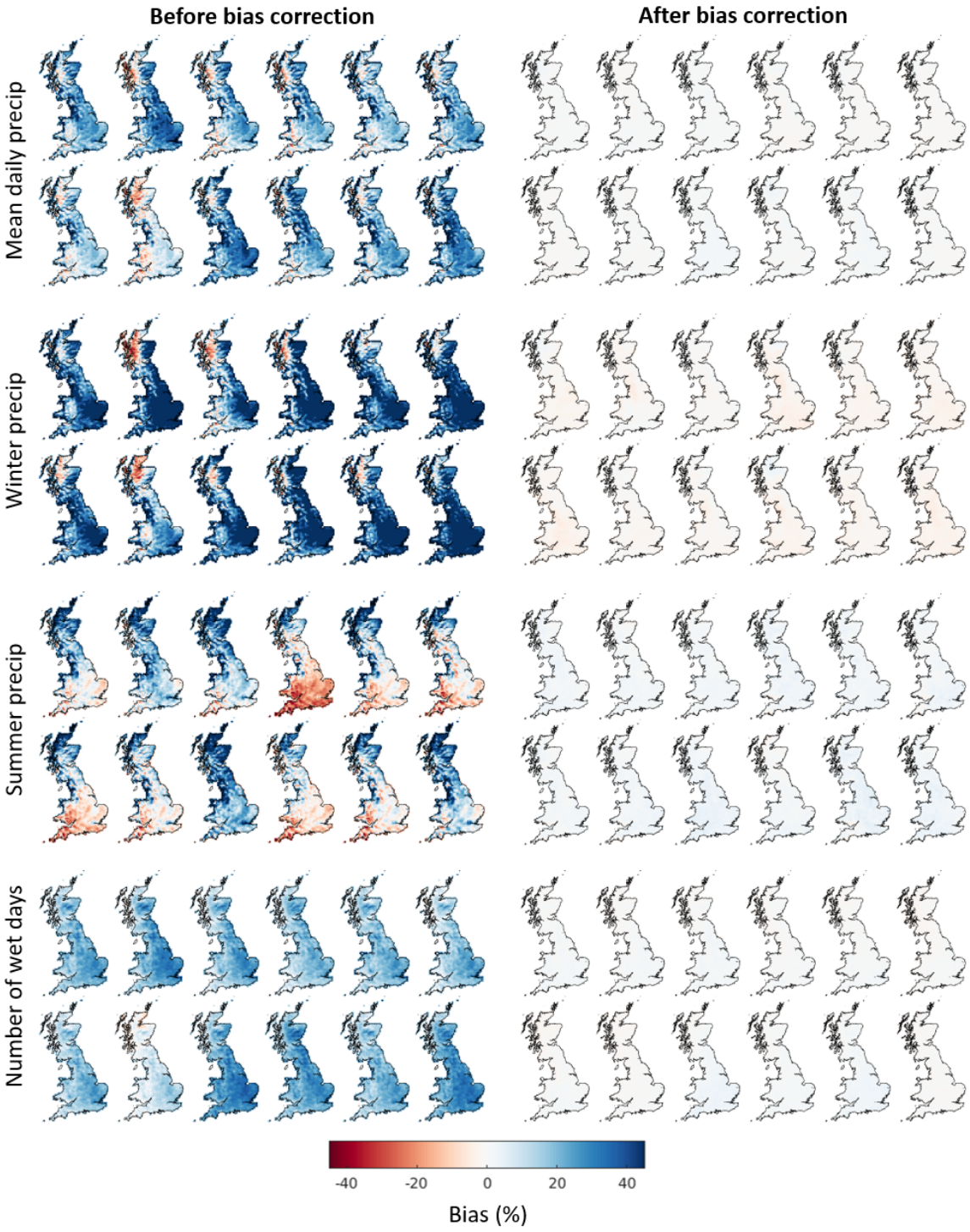


FIGURE G.3. Precipitation biases over the baseline period, before and after bias correction. All biases are given as a percentage of observed.

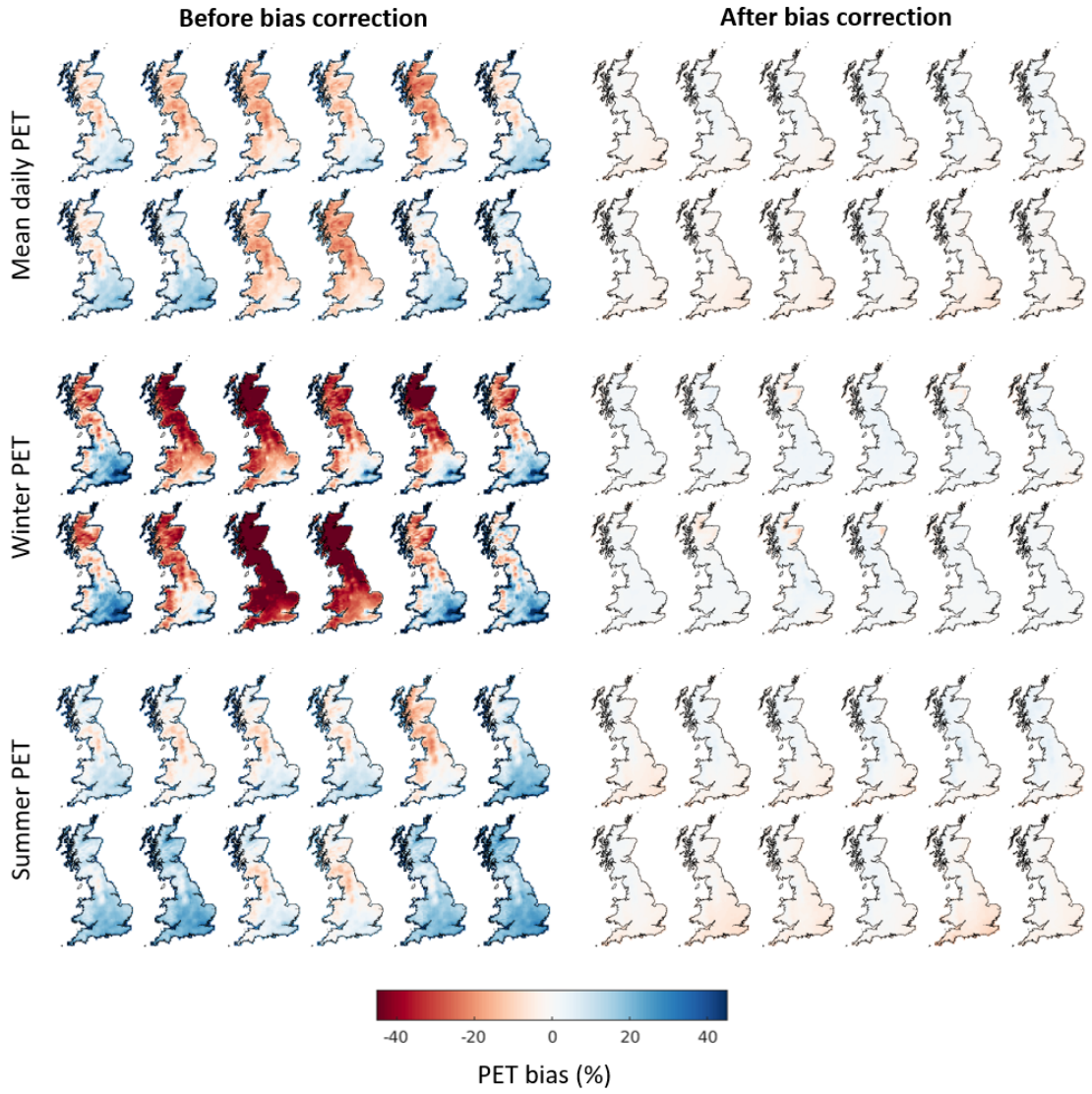


FIGURE G.4. PET biases over the baseline period, before and after bias correction.

BIBLIOGRAPHY

Abdulla, F. A. and Lettenmaier, D. P.: Development of regional parameter estimation equations for a macroscale hydrologic model, Journal of Hydrology, doi:10.1016/S0022-1694(96)03262-3, 1997.

Addor, N. and Melsen, L. A.: Legacy, Rather Than Adequacy, Drives the Selection of Hydrological Models, Water Resources Research, doi:10.1029/2018WR022958, 2019.

Addor, N. and Seibert, J.: Bias correction for hydrological impact studies - beyond the daily perspective, Hydrological Processes, 28, 4823–4828, doi:10.1002/hyp.10238, 2014.

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P.: The CAMELS data set : catchment attributes and meteorology for large-sample studies, Hydrology and Earth System Sciences Discussions, doi:10.5194/hess-2017-169, 2017.

Addor, N., Do, H. X., Alvarez-Garreton, C., Coxon, G., Fowler, K., and Mendoza, P. A.: Large-sample hydrology: recent progress, guidelines for new datasets and grand challenges, Hydrological Sciences Journal, doi:10.1080/02626667.2019.1683182, 2020.

Alvarez-Garreton, C., Mendoza, P. A., Pablo Boisier, J., Addor, N., Galleguillos, M., Zambrano-Bigiarini, M., Lara, A., Puelma, C., Cortes, G., Garreaud, R., McPhee, J., and Ayala, A.: The CAMELS-CL dataset: Catchment attributes and meteorology for large sample studies-Chile dataset, Hydrology and Earth System Sciences, doi:10.5194/hess-22-5817-2018, 2018.

Ambroise, B., Beven, K., and Freer, J.: Toward a Generalization of the TOPMODEL Concepts: Topographic Indices of Hydrological Similarity, Water Resources Research, 32, 2135–2145, doi:10.1029/95WR03716, URL `http://doi.wiley.com/10.1029/95WR03716`, 1996.

Andersson, J. C. M., Pechlivanidis, I. G., Gustafsson, D., Donnelly, C., and Arheimer, B.: Key factors for improving large-scale hydrological model performance, European Water, pp. 77–88, 2015.

Andréassian, V., Perrin, C., Michel, C., Usart-Sanchez, I., and Lavabre, J.: Impact of imperfect rainfall knowledge on the efficiency and the parameters of watershed models, Journal of Hydrology, doi:10.1016/S0022-1694(01)00437-1, 2001.

Andréassian, V., Hall, A., Chahinian, N., and Schaake, J.: Introduction and Synthesis: Why should hydrologists work on a large number of basin data sets?, 2006.

Andréassian, V., Perrin, C., Berthet, L., Le Moine, N., Lerat, J., Loumagne, C., Oudin, L., Mathevet, T., Ramos, M.-H., and Valéry, A.: Crash tests for a standardized evaluation of hydrological models, Hydrology and Earth System Sciences, 13, 1757–1764, doi:10.5194/hess-13-1757-2009, URL https://hess.copernicus.org/articles/13/1757/2009/, 2009.

Andréassian, V., Perrin, C., Parent, E., and Bárdossy, A.: The Court of Miracles of Hydrology: can failure stories contribute to hydrological science?, Hydrological Sciences Journal, doi: 10.1080/02626667.2010.506050, 2010.

Archfield, S. A., Clark, M., Arheimer, B., Hay, L. E., Mcmillan, H., Kiang, J. E., Seibert, J., Hakala, K., Bock, A., Wagener, T., Farmer, W. H., Andreassian, V., Attinger, S., Vigilone, A., Knight, R., Markstrom, S., and Over, T.: Accelerating advances in continental domain hydrologic modeling, Water Resources Research, 51, 10 078–10 091, doi:10.1002/2015WR017498, 2015.

Arnell, N. and Reynard, N.: The effects of climate change due to global warming on river flows in Great Britain, Journal of Hydrology, 183, 397–424, doi:10.1016/0022-1694(95)02950-8, URL https://linkinghub.elsevier.com/retrieve/pii/0022169495029508, 1996.

Arnell, N. W.: Uncertainty in the relationship between climate forcing and hydrological response in UK catchments, Hydrology and Earth System Sciences, doi:10.5194/hess-15-897-2011, 2011.

Arnell, N. W. and Gosling, S. N.: The impacts of climate change on river flow regimes at the global scale, Journal of Hydrology, doi:10.1016/j.jhydrol.2013.02.010, 2013.

Bárdossy, A., Huang, Y., and Wagener, T.: Simultaneous calibration of hydrological models in geographical space, Hydrology and Earth System Sciences, doi:10.5194/hess-20-2913-2016, 2016.

Baroni, G., Zink, M., Kumar, R., Samaniego, L., and Attinger, S.: Effects of uncertainty in soil properties on simulated hydrological states and fluxes at different spatio-temporal scales, Hydrology and Earth System Sciences, 21, 2301–2320, doi:10.5194/hess-21-2301-2017, 2017.

Bathurst, J. C.: Physically-based distributed modelling of an upland catchment using the Systeme Hydrologique Europeen, Journal of Hydrology, doi:10.1016/0022-1694(86)90116-2, 1986.

Beck, H. E., van Dijk, A. I. J. M., de Roo, A., Miralles, D. G., McVicar, T. R., Schellekens, J., and Bruijnzeel, L. A.: Global-scale regionalization of hydrologic model parameters, Water Resources Research, 52, 3599–3622, doi:10.1002/2015WR018247, URL http://doi.wiley.com/10.1002/2015WR018247, 2016.

Beck, H. E., Van Dijk, A. I., Levizzani, V., Schellekens, J., Miralles, D. G., Martens, B., and De Roo, A.: MSWEP: 3-hourly 0.25° global gridded precipitation (1979-2015) by merging gauge, satellite, and reanalysis data, Hydrology and Earth System Sciences, doi:10.5194/hess-21-589-2017, 2017.

Beck, M. B.: Water quality modeling: A review of the analysis of uncertainty, Water Resources Research, doi:10.1029/WR023i008p01393, 1987.

Bell, V. A., Kay, A. L., Jones, R. G., and Moore, R. J.: Use of a grid-based hydrological model and regional climate model outputs to assess changing flood risk, International Journal of Climatology, 27, 1657–1671, doi:10.1002/joc.1539, 2007.

Bell, V. A., Kay, A. L., Jones, R. G., Moore, R. J., and Reynard, N. S.: Use of soil data in a grid-based hydrological model to estimate spatial variation in changing flood risk across the UK, Journal of Hydrology, 377, 335–350, 2009.

Bell, V. A., Kay, A. L., Cole, S. J., Jones, R. G., Moore, R. J., and Reynard, N. S.: How might climate change affect river flows across the Thames Basin? An area-wide analysis using the UKCP09 Regional Climate Model ensemble, Journal of Hydrology, 442-443, 89–104, doi:10.1016/j.jhydrol.2012.04.001, 2012.

Bell, V. A., Kay, A. L., Davies, H. N., and Jones, R. G.: An assessment of the possible impacts of climate change on snow and peak river flows across Britain, Climatic Change, doi:10.1007/s10584-016-1637-x, URL http://link.springer.com/10.1007/s10584-016-1637-x, 2016.

Bergstrand, M., Asp, S.-S., and Lindström, G.: Nationwide hydrological statistics for Sweden with high resolution using the hydrological model S-HYPE, Hydrology Research, 45, 349–356, doi:10.2166/nh.2013.010, URL https://iwaponline.com/hr/article/45/3/349/532/Nationwide-hydrological-statistics-for-Sweden-with, 2014.

Beven, K.: Prophecy, reality and uncertainty in distributed hydrological modelling, Advances in Water Resources, doi:10.1016/0309-1708(93)90028-E, 1993.

Beven, K.: How far can we go in distributed hydrological modelling?, Hydrology and Earth System Sciences, 5, 1–12, doi:10.5194/hess-5-1-2001, URL https://hess.copernicus.org/articles/5/1/2001/, 2001.

Beven, K.: A manifesto for the equifinality thesis, in: Journal of Hydrology, doi:10.1016/j.jhydrol.2005.07.007, 2006.

Beven, K.: Environmental Modelling: An Uncertain Future?, Routledge, 2009.

Beven, K.: Rainfall-Runoff Modelling, John Wiley & Sons, Ltd, Chichester, UK, doi:10.1002/9781119951001, URL http://doi.wiley.com/10.1002/9781119951001, 2012.

Beven, K. and Binley, A.: The future of distributed models: Model calibration and uncertainty prediction, Hydrological Processes, 6, 279–298, doi:10.1002/hyp.3360060305, URL http://doi.wiley.com/10.1002/hyp.3360060305, 1992.

Beven, K. and Freer, J.: Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methodology, Journal of Hydrology, 249, 11–29, doi:10.1016/S0022-1694(01)00421-8, 2001a.

Beven, K. and Freer, J.: A dynamic TOPMODEL, Hydrological Processes, 15, 1993–2011, doi:10.1002/hyp.252, URL http://doi.wiley.com/10.1002/hyp.252, 2001b.

Beven, K., Young, P., Gupta, H., Thiemann, M., Trosset, M., and Sorooshian, S.: Comment on "Bayesian recursive parameter estimation for hydrologic models" by M. Thiemann, M. Trosset, H. Gupta, and S. Sorooshian, doi:10.1029/2001WR001183, 2003.

Beven, K., Cloke, H., Pappenberger, F., Lamb, R., and Hunter, N.: Hyperresolution information and hyperresolution ignorance in modelling the hydrology of the land surface, Science China Earth Sciences, 58, 25–35, doi:10.1007/s11430-014-5003-4, 2015.

Beven, K. J. and Alcock, R. E.: Modelling everything everywhere: A new approach to decision-making for water management under uncertainty, Freshwater Biology, doi:10.1111/j.1365-2427.2011.02592.x, 2012.

Beven, K. J. and Cloke, H. L.: Comment on “Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth’s terrestrial water” by Eric F. Wood et al., Water Resources Research, 48, 2–4, doi:10.1029/2010WR010090, URL http://doi.wiley.com/10.1029/2011WR010982, 2012.

Beven, K. J. and Kirkby, M. J.: A physically based , variable contributing area model of basin hydrology / Un modèle à base physique de zone d ’ appel variable de l ’ hydrologie du bassin versant, Hydrological Sciences Bulletin, 24, 43–69, doi:10.1080/02626667909491834, 1979.

Beven, K. J. and O’Connell, P. E.: On the role of physically-based distributed modelling in hydrology., UK Institute of Hydrology, NERC, Report, 1981.

Bierkens, M. F. P., Bell, V. A., Burek, P., Chaney, N., Condon, L. E., David, C. H., de Roo, A., Döll, P., Drost, N., Famiglietti, J. S., Flörke, M., Gochis, D. J., Houser, P., Hut, R., Keune, J., Kollet, S., Maxwell, R. M., Reager, J. T., Samaniego, L., Sudicky, E., Sutanudjaja, E. H., van de Giesen, N., Winsemius, H., and Wood, E. F.: Hyper-resolution global hydrological modelling: what is next?, Hydrological Processes, 29, 310–320, doi:10.1002/hyp.10391, URL http://doi.wiley.com/10.1002/hyp.10391, 2015.

Blazkova, S. and Beven, K.: A limits of acceptability approach to model evaluation and uncertainty estimation in flood frequency estimation by continuous simulation: Skalka catchment, Czech Republic, Water Resources Research, doi:10.1029/2007WR006726, 2009a.

Blazkova, S. and Beven, K.: Uncertainty in flood estimation, Structure and Infrastructure Engineering, doi:10.1080/15732470701189514, 2009b.

Blöschl, G., Sivapalan, M., Wagener, T., Viglione, A., and Savenije, H.: Runoff Prediction in Ungauged Basins, Cambridge University Press, Cambridge, doi:10.1017/CBO9781139235761, URL http://ebooks.cambridge.org/ref/id/CBO9781139235761, 2013.

Booij, M. J.: Impact of climate change on river flooding assessed with different spatial model resolutions, Journal of Hydrology, 303, 176–198, doi:10.1016/j.jhydrol.2004.07.013, URL http://linkinghub.elsevier.com/retrieve/pii/S002216940400397X, 2005.

Bosshard, T., Carambia, M., Goergen, K., Kotlarski, S., Krahe, P., Zappa, M., and Sch??r, C.: Quantifying uncertainty sources in an ensemble of hydrological climate-impact projections, Water Resources Research, 49, 1523–1536, doi:10.1029/2011WR011533, 2013.

Boyle, D. P., Gupta, H. V., and Sorooshian, S.: Toward improved calibration of hydrologic models: Combining the strengths of manual and automatic methods, Water Resources Research, doi:10.1029/2000WR900207, 2000.

Brown, C. M., Lund, J. R., Cai, X., Reed, P. M., Zagona, E. A., Ostfeld, A., Hall, J., Characklis, G. W., Yu, W., and Brekke, L.: The future of water resources systems analysis: Toward a scientific framework for sustainable water management, Water Resources Research, doi:10.1002/2015WR017114, 2015.

Burnash, R. J. C., Ferral, R. L., and McGuire, R. A.: A generalized streamflow simulation system: Conceptual modelling for digital computers, technical report, Tech. rep., U.S. Natl. Weather Serv., 1973.

Butts, M. B., Payne, J. T., Kristensen, M., and Madsen, H.: An evaluation of the impact of model structure on hydrological modelling uncertainty for streamflow simulation, Journal of Hydrology, 298, 242–266, doi:10.1016/j.jhydrol.2004.03.042, URL https://linkinghub.elsevier.com/retrieve/pii/S0022169404002471, 2004.

Buurman, J. and Babovic, V.: Adaptation Pathways and Real Options Analysis: An approach to deep uncertainty in climate change adaptation policies, Policy and Society, doi:10.1016/j.polsoc.2016.05.002, 2016.

Cameron, D.: An application of the UKCIP02 climate change scenarios to flood estimation by continuous simulation for a gauged catchment in the northeast of Scotland, UK (with

uncertainty), Journal of Hydrology, 328, 212–226, doi:10.1016/j.jhydrol.2005.12.024, URL https://linkinghub.elsevier.com/retrieve/pii/S0022169405006645, 2006.

Cameron, D., Beven, K., and Naden, P.: Flood frequency estimation by continuous simulation under climate change (with uncertainty), Hydrology and Earth System Sciences, 4, 393–405, doi:10.5194/hess-4-393-2000, URL http://www.hydrol-earth-syst-sci.net/4/393/2000/, 2000.

Carrillo, G., Troch, P. A., Sivapalan, M., Wagener, T., Harman, C., and Sawicz, K.: Catchment classification: hydrological analysis of catchment behavior through process-based modeling along a climate gradient, Hydrology and Earth System Sciences, 15, 3411–3430, doi:10.5194/hess-15-3411-2011, URL https://hess.copernicus.org/articles/15/3411/2011/, 2011.

CEH: National River Flow Archive, doi:10.1155/2016/6248178, 2015.

Centre for Ecology and Hydrology: National River Flow Archive, URL http://nrfa.ceh.ac.uk/, 2016.

Chaney, N. W., Metcalfe, P., and Wood, E. F.: HydroBlocks: a field-scale resolving land surface model for application over continental extents, Hydrological Processes, 30, 3543–3559, doi:10.1002/hyp.10891, URL http://doi.wiley.com/10.1002/hyp.10891, 2016.

Charlton, M. B. and Arnell, N. W.: Assessing the impacts of climate change on river flows in England using the UKCP09 climate change projections, Journal of Hydrology, 519, 1723–1738, doi:10.1016/j.jhydrol.2014.09.008, URL http://dx.doi.org/10.1016/j.jhydrol.2014.09.008, 2014.

Chen, J., Brissette, F. P., Poulin, A., and Leconte, R.: Overall uncertainty study of the hydrological impacts of climate change for a Canadian watershed, Water Resources Research, doi:10.1029/2011WR010602, 2011.

Choi, W. and Deal, B. M.: Assessing hydrological impact of potential land use change through hydrological and land use change modeling for the Kishwaukee River basin (USA), Journal of Environmental Management, doi:10.1016/j.jenvman.2007.06.001, 2008.

Christierson, B. v., Vidal, J.-P., and Wade, S. D.: Using UKCP09 probabilistic climate information for UK water resource planning, Journal of Hydrology, 424-425, 48–67, doi:10.1016/j.jhydrol.2011.12.020, URL http://dx.doi.org/10.1016/j.jhydrol.2011.12.020https://linkinghub.elsevier.com/retrieve/pii/S0022169411009036, 2012.

Clark, M. P., Slater, A. G., Rupp, D. E., Woods, R. A., Vrugt, J. A., Gupta, H. V., Wagener, T., and Hay, L. E.: Framework for Understanding Structural Errors (FUSE): A modular framework to diagnose differences between hydrological models, Water Resources Research, 44, 1–14, doi:10.1029/2007WR006735, 2008.

Clark, M. P., Kavetski, D., and Fenicia, F.: Pursuing the method of multiple working hypotheses for hydrological modeling, Water Resources Research, 47, 1–16, doi:10.1029/2010WR009827, 2011a.

Clark, M. P., McMillan, H. K., Collins, D. B. G., Kavetski, D., and Woods, R. A.: Hydrological field data from a modeller's perspective: Part 2: Process-based evaluation of model hypotheses, Hydrological Processes, 25, 523–543, doi:10.1002/hyp.7902, 2011b.

Clark, M. P., Nijssen, B., Lundquist, J. D., Kavetski, D., Rupp, D. E., Woods, R. A., Freer, J. E., Gutmann, E. D., Wood, A. W., Brekke, L. D., Arnold, J. R., Gochis, D. J., and Rasmussen, R. M.: A unified approach for process-based hydrologic modeling: 1. Modeling concept, Water Resources Research, doi:10.1002/2015WR017198, 2015.

Clark, M. P., Schaefli, B., Schymanski, S. J., Samaniego, L., Luce, C. H., Jackson, B. M., Freer, J. E., Arnold, J. R., and Dan, R.: Improving the theoretical underpinnings of process - based hydrologic models, Water Resources Research, Accepted A, 1–23, doi:10.1002/2015WR017910, URL http://doi.wiley.com/10.1002/2015WR017910, 2016a.

Clark, M. P., Wilby, R. L., Gutmann, E. D., Vano, J. A., Gangopadhyay, S., Wood, A. W., Fowler, H. J., Prudhomme, C., Arnold, J. A., and Brekke, L. D.: Characterizing uncertainty of the hydrologic impacts of climate change, Current Climate Change Reports, pp. 55–64, doi:10.1007/s40641-016-0034-x, URL http://dx.doi.org/10.1007/s40641-016-0034-x, 2016b.

Clark, M. P., Bierkens, M. F. P., Samaniego, L., Woods, R. A., Uijenhoet, R., Bennet, K. E., Pauwels, V. R. N., Cai, X., Wood, A. W., and Peters-Lidard, C. D.: The evolution of process-based hydrologic models: Historical challenges and the collective quest for physical realism, Hydrology and Earth System Sciences Discussions, 49, 1–14, doi:10.5194/hess-2016-693, URL http://www.hydrol-earth-syst-sci-discuss.net/hess-2016-693/, 2017.

Cloke, H. L. and Hannah, D. M.: Large-scale hydrology: Advances in understanding processes, dynamics and models from beyond river basin to global scale, doi:10.1002/hyp.8059, URL https://onlinelibrary.wiley.com/doi/full/10.1002/hyp.8059, 2011.

Cloke, H. L. and Pappenberger, F.: Ensemble flood forecasting: A review, doi:10.1016/j.jhydrol.2009.06.005, 2009.

Cloke, H. L., Wetterhall, F., He, Y., Freer, J. E., and Pappenberger, F.: Modelling climate impact on floods with ensemble climate projections, Quarterly Journal of the Royal Meteorological Society, 139, 282–297, doi:10.1002/qj.1998, 2013.

Collet, L., Harrigan, S., Prudhomme, C., Formetta, G., and Beevers, L.: Future hot-spots for hydro-hazards in Great Britain: A probabilistic assessment, Hydrology and Earth System Sciences, doi:10.5194/hess-22-5387-2018, 2018.

Contractor, S., Donat, M. G., Alexander, L. V., Ziese, M., Meyer-Christoffer, A., Schneider, U., Rustemeier, E., Becker, A., Durre, I., and Vose, R. S.: Rainfall Estimates on a Gridded Network (REGEN) - A global land-based gridded dataset of daily precipitation from 1950 to 2016, Hydrology and Earth System Sciences, doi:10.5194/hess-24-919-2020, 2020.

Cooke, W. E.: Forecasts and Verifications in Western Australia, Monthly Weather Review, 34, 23–24, 1906.

Coron, L., Andréassian, V., Perrin, C., Lerat, J., Vaze, J., Bourqui, M., and Hendrickx, F.: Crash testing hydrological models in contrasted climate conditions: An experiment on 216 Australian catchments, Water Resources Research, 48, 1–17, doi:10.1029/2011WR011721, 2012.

Cosby, B. J., Hornberger, G. M., Clapp, R. B., and Ginn, T. R.: A Statistical Exploration of the Relationships of Soil Moisture Characteristics to the Physical Properties of Soils, Water Resources Research, doi:10.1029/WR020i006p00682, 1984.

Countryside Survey: Topsoil bulk density data, 2007.

Coxon, G.: A Diagnostic Framework for the Evaluation of Multiple Hydrological Model Structures from a UK National Assessment of Discharge Uncertainties, Ph.D. thesis, University of Bristol, 2015.

Coxon, G., Freer, J., Wagener, T., Odoni, N. A., and Clark, M.: Diagnostic evaluation of multiple hypotheses of hydrological behaviour in a limits-of-acceptability framework for 24 UK catchments, Hydrological Processes, 28, 6135–6150, doi:10.1002/hyp.10096, URL `http://doi.wiley.com/10.1002/hyp.10096`, 2014.

Coxon, G., Freer, J., Westerberg, I. K., Wagener, T., Woods, R., and Smith, P. J.: A novel framework for discharge uncertainty quantification applied to 500 UK gauging stations, Water Resources Research, doi:10.1002/2014WR016532, 2015.

Coxon, G., Freer, J., Lane, R., Musuuza, J., Woods, R., Wagener, T., and Howden, N.: An evaluation of Dynamic TOPMODEL in natural and human-impacted catchments for low flow simulation, in: 19th EGU General Assembly, p. 9261, EGU2017, Vienna, Austria, 2017.

Coxon, G., Freer, J., Lane, R., Dunne, T., Knoben, W. J., Howden, N. J., Quinn, N., Wagener, T., and Woods, R.: DECIPHeR v1: Dynamic fluxEs and ConnectIvity for Predictions of HydRology, Geoscientific Model Development, doi:10.5194/gmd-12-2285-2019, 2019.

Coxon, G., Addor, N., Bloomfield, J., Freer, J., Fry, M., Hannaford, J., Howden, N., Lane, R., Lewis, M., Robinson, E., Wagener, T., and Woods, R.: CAMELS-GB: Hydrometeorological time series and landscape attributes for 671 catchments in Great Britain, Earth System Science Data Discussions, doi:10.5194/essd-2020-49, 2020.

Crooks, S., Kay, A., Davies, H., and Bell, V.: From Catchment to National Scale Rainfall-Runoff Modelling: Demonstration of a Hydrological Modelling Framework, Hydrology, 1, 63–88, doi: 10.3390/hydrology1010063, URL http://www.mdpi.com/2306-5338/1/1/63, 2014.

Crooks, S. M. and Naden, P. S.: CLASSIC: a semi-distributed rainfall-runoff modelling system, Hydrology and Earth System Sciences, 11, 516–531, doi:10.5194/hess-11-516-2007, URL https://hess.copernicus.org/articles/11/516/2007/, 2007.

De Niel, J., Van Uytven, E., and Willems, P.: Uncertainty Analysis of Climate Change Impact on River Flow Extremes Based on a Large Multi-Model Ensemble, Water Resources Management, doi:10.1007/s11269-019-02370-0, 2019.

DEFRA: The National Adaptation Programme and the Third Strategy for Climate Adaptation Reporting, Tech. rep., DEFRA, URL www.gov.uk/government/publications, 2018.

Dixon, H., Hannaford, J., and Fry, M. J.: The effective management of national hydrometric data: experiences from the United Kingdom, Hydrological Sciences Journal, doi:10.1080/02626667.2013.787486, 2013.

Donnelly, C., Andersson, J. C., and Arheimer, B.: Using flow signatures and catchment similarities to evaluate the E-HYPE multi-basin model across Europe, Hydrological Sciences Journal, 61, 255–273, doi:10.1080/02626667.2015.1027710, URL http://dx.doi.org/10.1080/02626667.2015.1027710http://www.tandfonline.com/doi/full/10.1080/02626667.2015.1027710, 2016.

Duan, Q., Sorooshian, S., and Gupta, V.: Effective and efficient global optimization for conceptual rainfall-runoff models, Water Resources Research, doi:10.1029/91WR02985, 1992.

Duan, Q., Schaake, J., Andréassian, V., Franks, S., Goteti, G., Gupta, H. V., Gusev, Y. M., Habets, F., Hall, A., Hay, L., Hogue, T., Huang, M., Leavesley, G., Liang, X., Nasonova, O. N., Noilhan, J., Oudin, L., Sorooshian, S., Wagener, T., and Wood, E. F.: Model Parameter Estimation Experiment (MOPEX): An overview of science strategy and major results from the second and third workshops, in: Journal of Hydrology, doi:10.1016/j.jhydrol.2005.07.031, 2006.

Engin, B. E., Yücel, I., and Yilmaz, A.: Assessing different sources of uncertainty in hydrological projections of high and low flows: case study for Omerli Basin, Istanbul, Turkey, Environmental Monitoring and Assessment, doi:10.1007/s10661-017-6059-3, 2017.

Environment Agency: Flooding in England: A National Assessment of Flood Risk, Tech. rep., Environment Agency, doi:GEHO0609BQDSEP, 2009.

European Parliament, C.: Directive 2000/60/EC of the European Parliament and of the Council of 23 October 2000 establishing a framework for Community action in the field of water policy, Official Journal of the European Parliament, doi:2004R0726-v.7of05.06.2013, 2000.

Falkenmark, M. and Chapman, T. C.: Comparative Hydrology: An Ecological Approach to Land and Water Resources, Unesco, Paris, France, 1989.

Fan, Y., Miguez-Macho, G., Jobbágy, E. G., Jackson, R. B., and Otero-Casal, C.: Hydrologic regulation of plant rooting depth, Proceedings of the National Academy of Sciences of the United States of America, doi:10.1073/pnas.1712381114, 2017.

Fekete, B. M., Vörösmarty, C. J., Roads, J. O., and Willmott, C. J.: Uncertainties in precipitation and their impacts on runoff estimates, Journal of Climate, doi:10.1175/1520-0442(2004)017<0294:UIPATI>2.0.CO;2, 2004.

Fenicia, F., Kavetski, D., and Savenije, H. H. G.: Elements of a flexible approach for conceptual hydrological modeling: 1. Motivation and theoretical development, Water Resources Research, 47, 1–13, doi:10.1029/2010WR010174, URL http://doi.wiley.com/10.1029/2010WR010174, 2011.

Fenicia, F., Kavetski, D., Savenije, H. H., Clark, M. P., Schoups, G., Pfister, L., and Freer, J.: Catchment properties, function, and conceptual model representation: Is there a correspondence?, Hydrological Processes, 28, 2451–2467, doi:10.1002/hyp.9726, 2014.

Fernandez, W., Vogel, R., and Sankarasubramanian, A.: Regional calibration of a watershed model / Calage regional d'un modele de bassin hydrologique, Hydrological Sciences Journal, doi:10.1080/02626660009492371, 2000.

Formetta, G., Prosdocimi, I., Stewart, E., and Bell, V.: Estimating the index flood with continuous hydrological models: an application in Great Britain, Hydrology Research, doi:10.2166/nh.2017.251, 2017.

Fowler, H. J. and Ekström, M.: Multi-model ensemble estimates of climate change impacts on UK seasonal precipitation extremes, International Journal of Climatology, 29, 385–416, doi:10.1002/joc.1827, URL http://doi.wiley.com/10.1002/joc.1827, 2009.

Fowler, H. J. and Kilsby, C. G.: Using regional climate model data to simulate historical and future river flows in northwest England, Climatic Change, 80, 337–367, doi:10.1007/s10584-006-9117-3, URL http://link.springer.com/10.1007/s10584-006-9117-3, 2007.

Freer, J., Beven, K. J., and Ambroise, B.: Bayesian estimation of uncertainty in runoff prediction and the value of data: An application of the GLUE approach, Water Resources Research, 32, 2161–2173, doi:10.1029/95WR03723, 1996.

Freer, J. E., McMillan, H., McDonnell, J. J., and Beven, K. J.: Constraining dynamic TOPMODEL responses for imprecise water table information using fuzzy rule based performance measures, in: Journal of Hydrology, doi:10.1016/j.jhydrol.2003.12.037, 2004.

Friedl, M. A., Sulla-Menashe, D., Tan, B., Schneider, A., Ramankutty, N., Sibley, A., and Huang, X.: MODIS Collection 5 global land cover: Algorithm refinements and characterization of new datasets, Remote Sensing of Environment, doi:10.1016/j.rse.2009.08.016, 2010.

Gao, H., Sabo, J. L., Chen, X., Liu, Z., Yang, Z., Ren, Z., and Liu, M.: Landscape heterogeneity and hydrological processes: a review of landscape-based hydrological models, doi:10.1007/s10980-018-0690-4, 2018.

Gao, J., Holden, J., and Kirkby, M.: A distributed TOPMODEL for modelling impacts of land-cover change on river flow in upland peatland catchments, Hydrological Processes, 29, 2867–2879, 2015.

Garen, D. C. and Burges, S. J.: Approximate error bounds for simulated hydrographs., Journal of the Hydraulics Division, ASCE, 1981.

Ghavidelfar, S., Alvankar, S. R., and Razmkhah, A.: Comparison of the Lumped and Quasi-distributed Clark Runoff Models in Simulating Flood Hydrographs on a Semi-arid Watershed, Water Resources Management, doi:10.1007/s11269-011-9774-5, 2011.

Götzinger, J. and Bárdossy, A.: Comparison of four regionalisation methods for a distributed hydrological model, Journal of Hydrology, 333, 374–384, doi:10.1016/j.jhydrol.2006.09.008, 2007.

Gupta, H., Thiemann, M., Trosset, M., and Sorooshian, S.: Reply to comment by K. Beven and P. Young on "Bayesian recursive parameter estimation for hydrologic models", Water Resources Research, doi:10.1029/2002wr001405, 2003.

Gupta, H. V., Kling, H., Yilmaz, K. K., and Martinez, G. F.: Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling, Journal of Hydrology, 377, 80–91, doi:10.1016/j.jhydrol.2009.08.003, URL http://dx.doi.org/10.1016/j.jhydrol.2009.08.003, 2009.

Gupta, H. V., Perrin, C., Blöschl, G., Montanari, A., Kumar, R., Clark, M. P., Andréassian, V., Bloschl, G., Montanari, A., Kumar, R., Clark, M. P., Gupta, H. V., Perrin, C., Bloschl, G., Montanari, A., Kumar, R., Clark, M. P., and Andreassian, V.: Large-sample hydrology : a need to balance depth with breadth, Hydrology and Earth System Sciences, 18, 463–477, doi:10.5194/hess-18-463-2014, 2014.

Hannaford, J.: Climate-driven changes in UK river flows: A review of the evidence, Progress in Physical Geography, 39, 29–48, doi:10.1177/0309133314536755, URL http://ppg.sagepub.com/content/39/1/29.abstract, 2015.

Hawkins, E. and Sutton, R.: The Potential to Narrow Uncertainty in Regional Climate Predictions, Bulletin of the American Meteorological Society, 90, 1095–1108, doi:10.1175/2009BAMS2607.1, URL http://journals.ametsoc.org/doi/10.1175/2009BAMS2607.1, 2009.

Henriksen, H. J., Troldborg, L., Nyegaard, P., Sonnenborg, T. O., Refsgaard, J. C., and Madsen, B.: Methodology for construction , calibration and validation of a national hydrological model for Denmark, Journal of Hydrology, 280, 52–71, doi:10.1016/S0022-1694(03)00186-0, 2003.

Hiederer, R.: Mapping Soil Properties for Europe - Spatial Representation of Soil Database Attributes, Publications Office of the European Union, doi:10.2788/94128, 2013a.

Hiederer, R.: Mapping Soil Typologies-Spatial Decision Support Applied to the European Soil Database, Publications Office of the European Union, doi:10.2788/87286, 2013b.

Højberg, A. L., Troldborg, L., Stisen, S., Christensen, B. B., and Henriksen, H. J.: Stakeholder driven update and improvement of a national water resources model, Environmental Modelling and Software, doi:10.1016/j.envsoft.2012.09.010, 2013.

Hollis, J. M., Hannam, J., and Bellamy, P. H.: Empirically-derived pedotransfer functions for predicting bulk density in European soils, European Journal of Soil Science, doi:10.1111/j.1365-2389.2011.01412.x, 2012.

Hollis, J. M., Lilly, A., Higgins, A., Jones, R. J., Keay, C. A., and Bellamy, P.: Predicting the water retention characteristics of UK mineral soils, European Journal of Soil Science, 66, 239–252, doi:10.1111/ejss.12186, 2015.

Hornberger, G. M., Beven, K. J., Cosby, B. J., and Sappington, D. E.: Shenandoah Watershed Study: Calibration of a Topography-Based, Variable Contributing Area Hydrological Model to a Small Forested Catchment, Water Resources Research, doi:10.1029/WR021i012p01841, 1985.

Hrachowitz, M., Savenije, H. H., Blöschl, G., McDonnell, J. J., Sivapalan, M., Pomeroy, J. W., Arheimer, B., Blume, T., Clark, M. P., Ehret, U., Fenicia, F., Freer, J. E., Gelfan, A., Gupta, H. V., Hughes, D. A., Hut, R. W., Montanari, A., Pande, S., Tetzlaff, D., Troch, P. A., Uhlenbrook, S., Wagener, T., Winsemius, H. C., Woods, R. A., Zehe, E., and Cudennec, C.: A decade of Predictions in Ungauged Basins (PUB)-a review, doi:10.1080/02626667.2013.803183, 2013.

Hundecha, Y. and Bárdossy, A.: Modeling of the effect of land use changes on the runoff generation of a river basin through parameter regionalization of a watershed model, Journal of Hydrology, 292, 281–295, doi:10.1016/j.jhydrol.2004.01.002, 2004.

Hundecha, Y., Ouarda, T. B., and Bárdossy, A.: Regional estimation of parameters of a rainfall-runoff model at ungauged watersheds using the "spatial" structures of the parameters within a canonical physiographic-climatic space, Water Resources Research, 44, 1–13, doi:10.1029/2006WR005439, 2008.

Huntington, T. G.: Evidence for intensification of the global water cycle : Review and synthesis, Journal of Hydrology, 319, 83–95, doi:10.1016/j.jhydrol.2005.07.003, 2006.

Im, S., Kim, H., Kim, C., and Jang, C.: Assessing the impacts of land use changes on watershed hydrology using MIKE SHE, Environmental Geology, doi:10.1007/s00254-008-1303-3, 2009.

Intergovernmental Panel on Climate Change, ed.: Climate Change 2013 - The Physical Science Basis, Cambridge University Press, Cambridge, doi:10.1017/CBO9781107415324, URL http://ebooks.cambridge.org/ref/id/CBO9781107415324, 2014.

Ivancic, T. J. and Shaw, S. B.: Examining why trends in very heavy precipitation should not be mistaken for trends in very high river discharge, Climatic Change, doi:10.1007/s10584-015-1476-1, 2015.

Jackson, C. R., Meister, R., and Prudhomme, C.: Modelling the effects of climate change and its uncertainty on UK Chalk groundwater resources from an ensemble of global climate model projections, Journal of Hydrology, doi:10.1016/j.jhydrol.2010.12.028, 2011.

Jiang, Y., Li, X., and Huang, C.: Automatic calibration a hydrological model using a master-slave swarms shuffling evolution algorithm based on self-adaptive particle swarm optimization, Expert Systems with Applications, doi:10.1016/j.eswa.2012.08.006, 2013.

Jin, X., Xu, C.-y., Zhang, Q., and Singh, V. P.: Parameter and modeling uncertainty simulated by GLUE and a formal Bayesian method for a conceptual hydrological model, Journal of Hydrology, 383, 147–155, doi:10.1016/j.jhydrol.2009.12.028, URL http://dx.doi.org/10.1016/j.jhydrol.2009.12.028, 2010.

Karlsson, I. B., Sonnenborg, T. O., Refsgaard, J. C., Trolle, D., Børgesen, C. D., Olesen, J. E., Jeppesen, E., and Jensen, K. H.: Combined effects of climate models, hydrological model structures and land use scenarios on hydrological impacts of climate change, Journal of Hydrology, 535, 301–317, doi:10.1016/j.jhydrol.2016.01.069, URL http://linkinghub.elsevier.com/retrieve/pii/S0022169416300099https://linkinghub.elsevier.com/retrieve/pii/S0022169416300099, 2016.

Kauffeldt, A., Halldin, S., Rodhe, A., Xu, C. Y., and Westerberg, I. K.: Disinformative data in large-scale hydrological modelling, Hydrology and Earth System Sciences, doi:10.5194/hess-17-2845-2013, 2013.

Kavetski, D., Kuczera, G., and Franks, S. W.: Bayesian analysis of input uncertainty in hydrological modeling: 2. Application, Water Resources Research, doi:10.1029/2005WR004376, 2006.

Kay, A. L. and Davies, H. N.: Calculating potential evaporation from climate model data: A source of uncertainty for hydrological climate change impacts, Journal of Hydrology, doi: 10.1016/j.jhydrol.2008.06.005, 2008.

Kay, A. L., Jones, D. A., Crooks, S. M., Calver, A., and Reynard, N. S.: A comparison of three approaches to spatial generalization of rainfall-runoff models, Hydrological Processes, doi: 10.1002/hyp.6550, 2006.

Kay, A. L., Davies, H. N., Bell, V. A., and Jones, R. G.: Comparison of uncertainty sources for climate change impacts: Flood frequency in England, Climatic Change, doi:10.1007/s10584-008-9471-4, 2009.

Kay, A. L., Crooks, S. M., Davies, H. N., Prudhomme, C., and Reynard, N. S.: Probabilistic impacts of climate change on flood frequency using response surfaces I: England and Wales, Regional Environmental Change, 14, 1215–1227, doi:10.1007/s10113-013-0563-y, 2014a.

Kay, A. L., Crooks, S. M., Davies, H. N., and Reynard, N. S.: Probabilistic impacts of climate change on flood frequency using response surfaces II: Scotland, Regional Environmental Change, 14, 1243–1255, doi:10.1007/s10113-013-0564-x, 2014b.

Kay, A. L., Crooks, S. M., and Reynard, N. S.: Using response surfaces to estimate impacts of climate change on flood peaks: Assessment of uncertainty, Hydrological Processes, 28, 5273–5287, doi:10.1002/hyp.10000, 2014c.

Kay, A. L., Bell, V. A., Guillod, B. P., Jones, R. G., and Rudd, A. C.: National-scale analysis of low flow frequency: historical trends and potential future changes, Climatic Change, 147, 585–599, doi:10.1007/s10584-018-2145-y, 2018.

Kelleher, C., McGlynn, B., and Wagener, T.: Characterizing and reducing equifinality by constraining a distributed catchment model with regional signatures, local observations, and process understanding, Hydrology and Earth System Sciences, doi:10.5194/hess-21-3325-2017, 2017.

Keller, V. D. J., Tanguy, M., Prosdocimi, I., Terry, J. A., Hitt, O., Cole, S. J., Fry, M., Morris, D. G., and Dixon, H.: CEH-GEAR : 1 km resolution daily and monthly areal rainfall estimates for the UK for hydrological and other applications, Earth System Science Data, 7, 143–155, doi:10.5194/essd-7-143-2015, 2015.

Kleidon, A.: Global datasets and rooting zone depth inferred from inverse methods, Journal of Climate, doi:10.1175/1520-0442(2004)017<2714:GDORZD>2.0.CO;2, 2004.

Klemes, V.: Operational testing of hydrological simulation models, Hydrological Sciences Journal, 31, 13–24, doi:10.1080/02626668609491024, URL http://www.tandfonline.com/doi/abs/10.1080/02626668609491024, 1986.

Klotz, D., Herrnegger, M., and Schulz, K.: Symbolic Regression for the Estimation of Transfer Functions of Hydrological Models, Water Resources Research, 53, 9402–9423, doi:10.1002/2017WR021253, 2017.

Knoben, W. J., Freer, J. E., Fowler, K. J., Peel, M. C., and Woods, R. A.: Modular Assessment of Rainfall-Runoff Models Toolbox (MARRMoT) v1.2: An open-source, extendable framework providing implementations of 46 conceptual hydrologic models as continuous state-space formulations, Geoscientific Model Development, doi:10.5194/gmd-12-2463-2019, 2019.

Kollat, J. B., Reed, P. M., and Wagener, T.: When are multiobjective calibration trade-offs in hydrologic models meaningful?, Water Resources Research, 48, 1–19, doi:10.1029/2011WR011534, URL http://doi.wiley.com/10.1029/2011WR011534, 2012.

Koppa, A., Gebremichael, M., and Yeh, W. W.: Multivariate calibration of large scale hydrologic models: The necessity and value of a Pareto optimal approach, Advances in Water Resources, doi:10.1016/j.advwatres.2019.06.005, 2019.

Koren, V., Smith, M., and Zhang, Z.: Use of soil property data in the derivation of conceptual rainfall-runoff model parameters, Proceedings of the 15th Conference on Hydrology, American Meteorological Society, Long Beach, California, pp. 103–106, 2000.

Kovács, G.: Proposal to construct a coordinating matrix for comparative hydrology, Hydrological Sciences Journal, 29, 435–443, doi:10.1080/02626668409490961, URL http://www.tandfonline.com/doi/abs/10.1080/02626668409490961, 1984.

Krueger, T., Freer, J., Quinton, J. N., Macleod, C. J. A., Bilotta, G. S., Brazier, R. E., Butler, P., and Haygarth, P. M.: Ensemble evaluation of hydrological model hypotheses, Water Resources Research, 46, 1–17, doi:10.1029/2009WR007845, URL http://doi.wiley.com/10.1029/2009WR007845, 2010.

Kumar, R., Samaniego, L., and Attinger, S.: The effects of spatial discretization and model parameterization on the prediction of extreme runoff characteristics, Journal of Hydrology, 392, 54–69, doi:10.1016/j.jhydrol.2010.07.047, URL http://dx.doi.org/10.1016/j.jhydrol.2010.07.047, 2010.

Kumar, R., Livneh, B., and Samaniego, L.: Toward computationally efficient large-scale hydrologic predictions with a multiscale regionalization scheme, Water Resources Research, 49, 5700–5714, doi:10.1002/wrcr.20431, 2013a.

Kumar, R., Samaniego, L., and Attinger, S.: Implications of distributed hydrologic model parameterization on water fluxes at multiple scales and locations, Water Resources Research, 49, 360–379, doi:10.1029/2012WR012195, 2013b.

Kundzewicz, Z. W., Krysanova, V., Benestad, R. E., Hov, Piniewski, M., and Otto, I. M.: Uncertainty in climate change impacts on water resources, Environmental Science and Policy, doi:10.1016/j.envsci.2017.10.008, 2018.

Laizé, C. L. and Hannah, D. M.: Modification of climate-river flow associations by basin properties, Journal of Hydrology, 389, 186–204, doi:10.1016/j.jhydrol.2010.05.048, 2010.

Lane, R., Coxon, G., Freer, J., Wagener, T., Johnes, P., Bloomfield, J., Greene, S., Macleod, C., and Reaney, S.: Benchmarking the predictive capability of hydrological models for river flow and flood peak predictions across over 1000 catchments in Great Britain, Hydrology and Earth System Sciences, doi:10.5194/hess-23-4011-2019, 2019.

Le Moine, N., Andréassian, V., Perrin, C., and Michel, C.: How can rainfall-runoff models handle intercatchment groundwater flows? Theoretical study based on 1040 French catchments, Water Resources Research, doi:10.1029/2006WR005608, 2007.

Leavesley, G. H., Lichty, R. W., Troutman, B. M., and Saindon, L. G.: Precipitation-runoff modeling system (PRMS) — User's Manual, Geological Survey Water Investigations Report, pp. 83–4238, URL https://www.researchgate.net/publication/247221248Precipitation-runoff, 1983.

Leavesley, G. H., Markstrom, S., Brewer, M. S., and Viger, R. J.: The Modular Modeling System (MMS) – The Physical Process Modeling Component of a Database-Centered Decision Support System for Water and Power Management, Water, air and soil pollution, 90, 303–311, 1996.

Lee, H., McIntyre, N. R., Wheater, H. S., and Young, A. R.: Predicting runoff in ungauged UK catchments, Proceedings of the ICE Water Management, doi:10.1680/wama.2006.159.2.129, 2006.

Lewis, E., Birkinshaw, S., Kilsby, C., and Fowler, H. J.: Development of a system for automated setup of a physically-based, spatially-distributed hydrological model for catchments in Great Britain, Environmental Modelling and Software, doi:10.1016/j.envsoft.2018.07.006, 2018.

Liang, X., Lettenmaier, D. P., Wood, E. F., and Burges, S. J.: A simple hydrologically based model of land surface water and energy fluxes for general circulation models, Journal of Geophysical Research, 99, 14 415–14 428, 1994.

Liden, R. and Harlin, J.: Analysis of conceptual rainfall–runoff modelling performance in different climates, Journal of Hydrology, 238, 231–247, URL https://www.sciencedirect.com/science/article/pii/S0022169400003309https://doi.org/10.1016/S0022-1694(00)00330-9, 2000.

Lilly, A.: private communication, The James Hutton Instituite, 2018.

Lindenschmidt, K. E., Hattermann, F., Mohaupt, V., Merz, B., Kundzewicz, Z. W., and Bronstert, A.: Large-scale hydrological modelling and the Water Framework Directive and Floods Directive of the European Union - 10th Workshop on Large-Scale Hydrological Modelling, doi:10.5194/adgeo-11-1-2007, 2007.

Liu, Y., Freer, J., Beven, K., and Matgen, P.: Towards a limits of acceptability approach to the calibration of hydrological models: Extending observation error, Journal of Hydrology, doi:10.1016/j.jhydrol.2009.01.016, 2009.

Livneh, B., Kumar, R., and Samaniego, L.: Influence of soil textural properties on hydrologic fluxes in the Mississippi river basin, Hydrological Processes, 29, 4638–4655, doi:10.1002/hyp.10601, 2015.

Lobligeois, F., Andréassian, V., Perrin, C., Tabary, P., and Loumagne, C.: When does higher spatial resolution rainfall information improve streamflow simulation? An evaluation using 3620 flood events, Hydrology and Earth System Sciences, doi:10.5194/hess-18-575-2014, 2014.

Lowe, J. A., Bernie, D., Bett, P., Bricheno, L., Brown, S., Calvert, D., Clark, R., Eagle, K., Edwards, T., Fosser, G., Fung, F., Gohar, L., Good, P., Gregory, J., Harris, G., Howard, T., Kaye, N., Kendon, E., Krijnen, J., Maisey, P., McDonald, R., McInnes, R., McSweeney, C., Mitchell, J., Murphy, J., Palmer, M., Roberts, C., Rostron, J., Sexton, D., Thornton, H., Tinker, J., Tucker, S., Yamazaki, K., and Belcher, S.: UKCP18 Science Overview Report: Version 2.0, Tech. rep., MetOffice, URL `https://www.metoffice.gov.uk/pub/data/weather/uk/ukcp18/science-reports/UKCP18-Overview-report.pdf`, 2019.

Luo, L. and Wood, E. F.: Monitoring and predicting the 2007 U.S. drought, Geophysical Research Letters, doi:10.1029/2007GL031673, 2007.

Madsen, H.: Parameter estimation in distributed hydrological catchment modelling using automatic calibration with multiple objectives, Advances in Water Resources, doi:10.1016/S0309-1708(02)00092-1, 2003.

Makhlouf, Z. and Michel, C.: A two-parameter monthly water balance model for French watersheds, Journal of Hydrology, doi:10.1016/0022-1694(94)90233-X, 1994.

Mango, L. M., Melesse, A. M., McClain, M. E., Gann, D., and Setegn, S. G.: Land use and climate change impacts on the hydrology of the upper Mara River Basin, Kenya: Results of a modeling study to support better resource management, Hydrology and Earth System Sciences, doi:10.5194/hess-15-2245-2011, 2011.

Marsh, T., Cole, G., and Wilby, R.: Major droughts in England and Wales, 1800–2006, Weather, doi:10.1002/wea.67, 2007.

Marsh, T. J. and Hannaford, J.: UK hydrometric register, Centre for Ecology and Hydrology, Wallingford, UK, 2008.

McDonnell, J. J., Sivapalan, M., Vaché, K., Dunn, S., Grant, G., Haggerty, R., Hinz, C., Hooper, R., Kirchner, J., Roderick, M. L., Selker, J., and Weiler, M.: Moving beyond heterogeneity and process complexity: A new vision for watershed hydrology, doi:10.1029/2006WR005467, 2007.

McIntyre, N., Lee, H., Wheater, H., Young, A., and Wagener, T.: Ensemble predictions of runoff in ungauged catchments, Water Resources Research, doi:10.1029/2005WR004289, 2005.

McMillan, H., Freer, J., Pappenberger, F., Krueger, T., and Clark, M.: Impacts of uncertain river flow data on rainfall-runoff model calibration and discharge predictions, Hydrological Processes, 24, n/a–n/a, doi:10.1002/hyp.7587, URL http://doi.wiley.com/10.1002/hyp.7587, 2010.

Mcmillan, H., Krueger, T., and Freer, J.: Benchmarking observational uncertainties for hydrology: Rainfall, river discharge and water quality, Hydrological Processes, 26, 4078–4111, doi:10.1002/hyp.9384, 2012.

McMillan, H., Booker, D., and Cattoën, C.: Validation of a national hydrological model, Journal of Hydrology, 541, 800–815, doi:10.1016/j.jhydrol.2016.07.043, URL http://dx.doi.org/10.1016/j.jhydrol.2016.07.043https://linkinghub.elsevier.com/retrieve/pii/S0022169416304735, 2016.

McMillan, H., Seibert, J., Petersen-Overleir, A., Lang, M., White, P., Snelder, T., Rutherford, K., Krueger, T., Mason, R., and Kiang, J.: How uncertainty analysis of streamflow data can reduce costs and promote robust decisions in water management applications, Water Resources Research, 53, 5220–5228, doi:10.1002/2016WR020328, 2017.

McMillan, H. K., Westerberg, I. K., and Krueger, T.: Hydrological data uncertainty and its implications, WIREs Water, doi:10.1002/wat2.1319, 2018.

Melsen, L. A., Addor, N., Mizukami, N., Newman, A. J., Torfs, P. J., Clark, M. P., Uijlenhoet, R., and Teuling, A. J.: Mapping (dis)agreement in hydrologic projections, Hydrology and Earth System Sciences, doi:10.5194/hess-22-1775-2018, 2018.

Mendoza, P. A., Clark, M. P., Mizukami, N., Newman, A. J., Barlage, M., Gutmann, E. D., Rasmussen, R. M., Rajagopalan, B., Brekke, L. D., and Arnold, J. R.: Effects of Hydrologic Model Choice and Calibration on the Portrayal of Climate Change Impacts, Journal of Hydrometeorology, 16, 762–780, doi:10.1175/JHM-D-14-0104.1, URL http://journals.ametsoc.org/doi/10.1175/JHM-D-14-0104.1, 2015.

Mendoza, P. A., Clark, M. P., Mizukami, N., Gutmann, E. D., Arnold, J. R., Brekke, L. D., and Rajagopalan, B.: How do hydrologic modeling decisions affect the portrayal of climate

change impacts?, Hydrological Processes, 30, 1071–1095, doi:10.1002/hyp.10684, URL http://doi.wiley.com/10.1002/hyp.10684, 2016.

Meresa, H. K. and Romanowicz, R. J.: The critical role of uncertainty in projections of hydrological extremes, Hydrology and Earth System Sciences, doi:10.5194/hess-21-4245-2017, 2017.

Merz, R. and Blöschl, G.: Regionalisation of catchment model parameters, Journal of Hydrology, doi:10.1016/j.jhydrol.2003.09.028, 2004.

Met Office: UK Climate, URL https://www.metoffice.gov.uk/public/weather/climate, 2014.

Met Office: UK Climate Projections: Headline Findings, Tech. rep., Met Office, URL https://www.metoffice.gov.uk/binaries/content/assets/metofficegovuk/pdf/research/ukcp/ukcp-headline-findings-v2.pdf, 2019.

Met Office: Regional (12km) and Local (2.2km) Projections, URL https://www.metoffice.gov.uk/research/approach/collaboration/ukcp/high-res-projections, 2020.

Met Office Hadley Centre: UKCP18 Guidance: Data availability, access and formats, Tech. rep., Met Office, URL https://www.metoffice.gov.uk/binaries/content/assets/metofficegovuk/pdf/research/ukcp/ukcp18-guidance-data-availability-access-and-formats.pdf, 2019.

Mizukami, N., Clark, M. P., Newman, A. J., Wood, A. W., Gutmann, E. D., Nijssen, B., Rakovec, O., and Samaniego, L.: Towards seamless large-domain parameter estimation for hydrologic models, Water Resources Research, 53, 8020–8040, doi:10.1002/2017WR020401, URL http://doi.wiley.com/10.1002/2017WR020401, 2017.

Montanari, A., Shoemaker, C. A., and Van De Giesen, N.: Introduction to special section on uncertainty assessment in surface and subsurface hydrology: An overview of issues and challenges, Water Resources Research, doi:10.1029/2009WR008471, 2009.

Moore, R. J.: The PDM rainfall-runoff model, Hydrology and Earth System Sciences, doi:10.5194/hess-11-483-2007, 2007.

Mouelhi, S., Michel, C., Perrin, C., and Andréassian, V.: Linking stream flow to rainfall at the annual time step: The Manabe bucket model revisited, Journal of Hydrology, doi:10.1016/j.jhydrol.2005.12.022, 2006.

Muerth, M. J., Gauvin St-Denis, B., Ricard, S., Velázquez, J. A., Schmid, J., Minville, M., Caya, D., Chaumont, D., Ludwig, R., and Turcotte, R.: On the need for bias correction in regional climate scenarios to assess climate change impacts on river runoff, Hydrology and Earth System Sciences, doi:10.5194/hess-17-1189-2013, 2013.

Murphy, J., Sexton, D., Jenkins, G., Boorman, P., Booth, B., Brown, K., Clark, R., Collins, M., Harris, G., Kendon, L., and Met Office Hadley Centre: UK Climate Projections science report : Climate change projections, Tech. Rep. December, UK Climate Projections, 2010.

Murphy, J., Harris, G., Sexton, D., Kendon, E., Bett, P., Clark, R., Eagle, K., Fosser, G., Fung, F., Lowe, J., McDonald, R., McInnes, R., McSweeney, C., Mitchell, J., Rostron, J., Thornton, H., Tucker, S., and Yamazaki, K.: UKCP18 Land Projections: Science Report, Tech. rep., MetOffice, URL https://www.metoffice.gov.uk/pub/data/weather/uk/ukcp18/science-reports/UKCP18-Land-report.pdf, 2018.

Nash, J. and Sutcliffe, J.: River flow forecasting through conceptual models. Part I - a discussion of principles., Journal of Hydrology, 10, 282–290, 1970.

Nathan, R. J. and McMahon, T. A.: SFB model part l. Validation of fixed model parameters, Transactions of the Institution of Engineers, Australia. Civil engineering, 1990.

National Infrastructure Commission: Preparing for a drier future: England's water infrastructure needs, Tech. rep., National Infrastructure Commission, URL https://www.nic.org.uk/publications/preparing-for-a-drier-future-englands-water-infrastructure-needs/, 2018.

Newman, A. J., Clark, M. P., Sampson, K., Wood, A., Hay, L. E., Bock, A., Viger, R. J., Blodgett, D., Brekke, L., Arnold, J. R., Hopson, T., and Duan, Q.: Development of a large-sample watershed-scale hydrometeorological data set for the contiguous USA: Data set characteristics and assessment of regional variability in hydrologic model performance, Hydrology and Earth System Sciences, doi:10.5194/hess-19-209-2015, 2015.

Newman, A. J., Mizukami, N., Clark, M. P., Wood, A. W., Nijssen, B., and Nearing, G.: Benchmarking of a Physically Based Hydrologic Model, Journal of Hydrometeorology, 18, 2215–2225, doi:10.1175/JHM-D-16-0284.1, URL http://journals.ametsoc.org/doi/10.1175/JHM-D-16-0284.1, 2017.

Nicolle, P., Pushpalatha, R., Perrin, C., Francois, D., Thiery, D., Mathevet, T., Lay, M. L., Besson, F., Soubeyroux, J.-m., and Viel, C.: Benchmarking hydrological models for low-flow simulation and forecasting on French catchments, Hydrology and Earth System Sciences, 18, 2829–2857, doi:10.5194/hess-18-2829-2014, 2014.

Niehoff, D., Fritsch, U., and Bronstert, A.: Land-use impacts on storm-runoff generation: Scenarios of land-use change and simulation of hydrological response in a meso-scale catchment in SW-Germany, Journal of Hydrology, doi:10.1016/S0022-1694(02)00142-7, 2002.

Nijssen, B., O'Donnell, G. M., Lettenmaier, D. P., Lohmann, D., and Wood, E. F.: Predicting the Discharge of Global Rivers, Journal of Climate, 14, 3307–3323, doi:10.1175/

1520-0442(2001)014<3307:PTDOGR>2.0.CO;2, URL http://journals.ametsoc.org/doi/abs/10.1175/1520-0442%282001%29014%3C3307%3APTDOGR%3E2.0.CO%3B2, 2001.

Nikulin, G., Kjellström, E., Hansson, U., Strandberg, G., and Ullerstig, A.: Evaluation and future projections of temperature, precipitation and wind extremes over Europe in an ensemble of regional climate simulations, Tellus, Series A: Dynamic Meteorology and Oceanography, doi:10.1111/j.1600-0870.2010.00466.x, 2011.

Ning, L., Mann, M. E., Crane, R., and Wagener, T.: Probabilistic projections of climate change for the mid-Atlantic region of the United States: Validation of precipitation downscaling during the historical era, Journal of Climate, doi:10.1175/2011JCLI4091.1, 2012.

Oubeidillah, A. A., Kao, S.-C., Ashfaq, M., Naz, B. S., and Tootle, G.: A large-scale, high-resolution hydrological model parameter data set for climate change impact assessment for the conterminous US, Hydrology and Earth System Sciences, 18, 67–84, doi:10.5194/hess-18-67-2014, URL https://hess.copernicus.org/articles/18/67/2014/, 2014.

Oudin, L., Andre, V., Perrin, C., Michel, C., and Le Moine, N.: Spatial proximity , physical similarity , regression and ungaged catchments : A comparison of regionalization approaches based on 913 French catchments, Water Resources Research, 44, 1–15, doi:10.1029/2007WR006240, 2008.

Pappenberger, F. and Beven, K. J.: Ignorance is bliss: Or seven reasons not to use uncertainty analysis, Water Resources Research, 42, 1–8, doi:10.1029/2005WR004820, URL http://doi.wiley.com/10.1029/2005WR004820, 2006.

Pappenberger, F., Thielen, J., and Del Medico, M.: The impact of weather forecast improvements on large scale hydrology: Analysing a decade of forecasts of the European Flood Alert System, Hydrological Processes, doi:10.1002/hyp.7772, 2011.

Pappenberger, F., Ramos, M. H., Cloke, H. L., Wetterhall, F., Alfieri, L., Bogner, K., Mueller, A., and Salamon, P.: How do I know if my forecasts are better? Using benchmarks in hydrological ensemble prediction, Journal of Hydrology, doi:10.1016/j.jhydrol.2015.01.024, 2015.

Parajka, J., Merz, R., and Blöschl, G.: A comparison of regionalisation methods for catchment model parameters, Hydrology and Earth System Sciences, 9, 157–171, doi:10.5194/hess-9-157-2005, URL http://www.hydrol-earth-syst-sci.net/9/157/2005/, 2005.

Parajka, J., Blöschl, G., and Merz, R.: Regional calibration of catchment models: Potential for ungauged catchments, Water Resources Research, doi:10.1029/2006WR005271, 2007a.

Parajka, J., Merz, R., and Blöschl, G.: Uncertainty and multiple objective calibration in regional water balance modelling: Case study in 320 Austrian catchments, Hydrological Processes, doi:10.1002/hyp.6253, 2007b.

Parajka, J., Naeimi, V., Blöschl, G., and Komma, J.: Matching ERS scatterometer based soil moisture patterns with simulations of a conceptual dual layer hydrologic model over Austria, Hydrology and Earth System Sciences, doi:10.5194/hess-13-259-2009, 2009.

Pechlivanidis, I. G. and Arheimer, B.: Large-scale hydrological modelling by using modified PUB recommendations: The India-HYPE case, Hydrology and Earth System Sciences, doi: 10.5194/hess-19-4559-2015, 2015.

Pechlivanidis, I. G., Jackson, B. M., Mcintyre, N. R., and Wheater, H. S.: Catchment scale hydrological modelling: A review of model types, calibration approaches and uncertainty analysis methods in the context of recent developments in technology and applications, doi: 10.30955/gnj.000778, 2011.

Pelletier, J. D. and Rasmussen, C.: Geomorphically based predictive mapping of soil thickness in upland watersheds, Water Resources Research, doi:10.1029/2008WR007319, 2009.

Pelletier, J. D., Broxton, P. D., Hazenberg, P., Zeng, X., Troch, P. A., Niu, G. Y., Williams, Z., Brunke, M. A., and Gochis, D.: A gridded global data set of soil, intact regolith, and sedimentary deposit thicknesses for regional and global land surface modeling, Journal of Advances in Modeling Earth Systems, doi:10.1002/2015MS000526, 2016.

Perrin, C., Michel, C., and Andreassian, V.: Does a large number of parameters enhance model performance ? Comparative assessment of common catchment model structures on 429 catchments, Journal of Hydrology, 242, 275–301, 2001.

Perrin, C., Michel, C., and Andréassian, V.: Improvement of a parsimonious model for streamflow simulation, Journal of Hydrology, doi:10.1016/S0022-1694(03)00225-7, 2003.

Perrin, C., Andre, V., Serna, C. R., Mathevet, T., and Le Moine, N.: Discrete parameterization of hydrological models : Evaluating the use of parameter sets libraries over 900 catchments, Water Resources Research, 44, 1–15, doi:10.1029/2007WR006579, 2008.

Perry, M. and Hollis, D.: The generation of monthly gridded datasets for a range of climatic variables over the UK, International Journal of Climatology, doi:10.1002/joc.1161, 2005.

Petrow, T. and Merz, B.: Trends in flood magnitude, frequency and seasonality in Germany in the period 1951-2002, Journal of Hydrology, doi:10.1016/j.jhydrol.2009.03.024, 2009.

Pfister, L., Martínez-Carreras, N., Hissler, C., Klaus, J., Carrer, G. E., Stewart, M. K., and McDonnell, J. J.: Bedrock geology controls on catchment storage, mixing, and release: A comparative analysis of 16 nested catchments, Hydrological Processes, doi:10.1002/hyp.11134, 2017.

Pianosi, F., Sarrazin, F., and Wagener, T.: A Matlab toolbox for Global Sensitivity Analysis, Environmental Modelling and Software, doi:10.1016/j.envsoft.2015.04.009, 2015.

Pokhrel, P. and Gupta, H. V.: On the use of spatial regularization strategies to improve calibration of distributed watershed models, Water Resources Research, doi:10.1029/2009WR008066, 2010.

Pokhrel, P., Gupta, H. V., and Wagener, T.: A spatial regularization approach to parameter estimation for a distributed watershed model, Water Resources Research, 44, 1–16, doi:10.1029/2007WR006615, 2008.

Poncelet, C., Merz, R., Merz, B., Parajka, J., Oudin, L., Andréassian, V., and Perrin, C.: Process-based interpretation of conceptual hydrological model performance using a multinational catchment set, Water Resources Research, 53, 7247–7268, doi:10.1002/2016WR019991, 2017.

Pool, S., Vis, M., and Seibert, J.: Evaluating model performance: towards a non-parametric variant of the Kling-Gupta efficiency, Hydrological Sciences Journal, doi:10.1080/02626667.2018.1552002, 2018.

Prudhomme, C. and Davies, H.: Assessing uncertainties in climate change impact analyses on the river flow regimes in the UK. Part 2: Future climate, Climatic Change, 93, 197–222, doi:10.1007/s10584-008-9461-6, 2009.

Prudhomme, C. and Williamson, J.: Derivation of RCM-driven potential evapotranspiration for hydrological climate change impact analysis in Great Britain: A comparison of methods and associated uncertainty in future projections, Hydrology and Earth System Sciences, doi:10.5194/hess-17-1365-2013, 2013.

Prudhomme, C., Jakob, D., and Svensson, C.: Uncertainty and climate change impact on the flood regime of small UK catchments, Journal of Hydrology, 277, 1–23, doi:10.1016/S0022-1694(03)00065-9, 2003.

Prudhomme, C., Wilby, R., Crooks, S., Kay, A., and Reynard, N.: Scenario-neutral approach to climate change impact studies: Application to flood risk, Journal of Hydrology, 390, 198–209, doi:10.1016/j.jhydrol.2010.06.043, URL http://dx.doi.org/10.1016/j.jhydrol.2010.06.043https://linkinghub.elsevier.com/retrieve/pii/S0022169410004117, 2010.

Prudhomme, C., Young, A., Watts, G., Haxton, T., Crooks, S., Williamson, J., Davies, H., Dadson, S., and Allen, S.: The drying up of Britain? A national estimate of changes in seasonal river flows from 11 Regional Climate Model simulations, Hydrological Processes, 26, 1115–1118, doi:10.1002/hyp.8434, URL http://doi.wiley.com/10.1002/hyp.8434, 2012.

Prudhomme, C., Crooks, S., Kay, A. L., and Reynard, N.: Climate change and river flooding: Part 1 classifying the sensitivity of British catchments, Climatic Change, 119, 933–948, doi:10.1007/s10584-013-0748-x, 2013a.

Prudhomme, C., Haxton, T., Crooks, S., Jackson, C., Barkwith, A., Williamson, J., Kelvin, J., Mackay, J., Wang, L., Young, A., and Watts, G.: Future Flows Hydrology: An ensemble of daily river flow and monthly groundwater levels for use for climate change impact assessment across Great Britain, Earth System Science Data, doi:10.5194/essd-5-101-2013, 2013b.

Rakovec, O., Kumar, R., Attinger, S., and Samaniego, L.: Improving the realism of hydrologic model functioning through multivariate parameter estimation, Water Resources Research, 52, 7779–7792, doi:10.1002/2016WR019430, 2016a.

Rakovec, O., Kumar, R., Mai, J., Cuntz, M., Thober, S., Zink, M., Attinger, S., Schäfer, D., Schrön, M., and Samaniego, L.: Multiscale and Multivariate Evaluation of Water Fluxes and States over European River Basins, Journal of Hydrometeorology, 17, 287–307, doi:10.1175/JHM-D-15-0054.1, URL http://journals.ametsoc.org/doi/10.1175/JHM-D-15-0054.1, 2016b.

Rakovec, O., Mizukami, N., Kumar, R., Newman, A. J., Thober, S., Wood, A. W., Clark, M. P., and Samaniego, L.: Diagnostic Evaluation of Large-Domain Hydrologic Models Calibrated Across the Contiguous United States, Journal of Geophysical Research: Atmospheres, doi:10.1029/2019JD030767, 2019.

Razavi, T. and Coulibaly, P.: Streamflow Prediction in Ungauged Basins: Review of Regionalization Methods, Journal of Hydrologic Engineering, 18, 958–975, doi:10.1061/(ASCE)HE.1943-5584.0000690, URL http://ascelibrary.org/doi/10.1061/%28ASCE%29HE.1943-5584.0000690, 2013.

Reclamation: Downscaled CMIP3 and CMIP5 Hydrology Climate Projections: Release of Hydrology Projections, Comparison with Preceding Information, and Summary of User Needs, Tech. rep., US Bureau of Reclamation, 2014.

Reclamation: West wide climate risk assessments: Hydroclimate projections, Tech. rep., U.S. Dep. of the Inter. Tech. Serv. Cent., Denver, Colorado, 2016.

Reed, S., Koren, V., Smith, M., Zhang, Z., Moreda, F., and Seo, D. J.: Overall distributed model intercomparison project results, in: Journal of Hydrology, doi:10.1016/j.jhydrol.2004.03.031, 2004.

Reynard, N., Crooks, S., Wilby, R., and Kay, A.: Climate change and flood frequency in the UK, 39th Defra Flood and Coastal Flood management Conference, pp. 1–12, URL http://nora.nerc.ac.uk/id/eprint/2976, 2004.

Reynard, N. S., Prudhomme, C., and Crooks, S. M.: The flood characteristics of large U.K. rivers: Potential effects of changing climate and land use, Climatic Change, doi:10.1023/A:1010735726818, 2001.

Robinson, E., Blyth, E., Clark, D., Finch, J., and Rudd, A.: Climate hydrology and ecology research support system potential evapotranspiration dataset for Great Britain (1961-2012) [CHESS-PE], Tech. rep., CEH, doi:10.5285/d329f4d6-95ba-4134-b77a-a377e0755653, 2015a.

Robinson, E., Blyth, E., Clark, D., Finch, J., and Rudd, A.: Climate hydrology and ecology research support system meteorology dataset for Great Britain (1961-2012) [CHESS-met], Tech. rep., UKCEH, doi:10.5285/80887755-1426-4dab-a4a6-250919d5020c, 2015b.

Robinson, E. L., Blyth, E., Clark, D. B., Finch, J., and Rudd, A. C.: Climate hydrology and ecology research support system meteorology dataset for Great Britain (1961-2012) [CHESS-met], NERC Environmental Information Data Centre, doi:10.1016/j.eplepsyres.2014.09.003, 2015c.

Rojas-Serna, C., Lebecherel, L., Perrin, C., Andréassian, V., and Oudin, L.: How should a rainfall-runoff model be parameterized in an almost ungauged catchment? A methodology tested on 609 catchments, Water Resources Research, doi:10.1002/2015WR018549, 2016.

Romanowicz, R. J. and Beven, K. J.: Comments on generalised likelihood uncertainty estimation, Reliability Engineering and System Safety, doi:10.1016/j.ress.2005.11.030, 2006.

Rowland, C., Morton, R., Carrasco, L., McShane, G., O'Neil, A., and Wood, C.: Land Cover Map 2015, NERC Environmental Information Data Centre, doi:10.5285/505d1e0c-ab60-4a60-b448-68c5bbae403e, 2017.

Rudd, A. C., Bell, V. A., and Kay, A. L.: National-scale analysis of simulated hydrological droughts (1891–2015), Journal of Hydrology, doi:10.1016/j.jhydrol.2017.05.018, 2017.

Rudd, A. C., Kay, A. L., and Bell, V. A.: National-scale analysis of future river flow and soil moisture droughts: potential changes in drought characteristics, Climatic Change, doi:10.1007/s10584-019-02528-0, 2019.

Salavati, B., Oudin, L., Furusho, C., and Ribstein, P.: Urbanization impact assessment on catchments hydrological response over 172 watersheds in USA, Houille Blanche, doi:10.1051/lhb/20150033, 2015.

Samaniego, L., Kumar, R., and Attinger, S.: Multiscale parameter regionalization of a grid-based hydrologic model at the mesoscale, Water Resources Research, 46, 1–25, doi:10.1029/2008WR007327, 2010.

Samaniego, L., Kumar, R., Thober, S., Rakovec, O., Zink, M., Wanders, N., Eisner, S., Müller Schmied, H., Sutanudjaja, E., Warrach-Sagi, K., and Attinger, S.: Toward seamless hydrologic predictions across spatial scales, Hydrology and Earth System Sciences, 21, 4323–4346, doi:10.5194/hess-21-4323-2017, 2017.

Samaniego, L., Kumar, R., Thober, S., Rakovec, O., Schweppe, R., Schäfer, D., Schrön, M., Brenner, J., Demirel, C. M., Kaluza, M., Jing, M., Langenberg, B., and Attinger, S.: mesoscale Hydrologic Model, doi:10.5281/zenodo.1299584, 2018.

Samuel, J., Coulibaly, P., and Metcalfe, R. A.: Evaluation of future flow variability in ungauged basins: Validation of combined methods, Advances in Water Resources, doi:10.1016/j.advwatres.2011.09.015, 2012.

Sawicz, K. A., Kelleher, C., Wagener, T., Troch, P., Sivapalan, M., and Carrillo, G.: Characterizing hydrologic change through catchment classification, Hydrology and Earth System Sciences, doi:10.5194/hess-18-273-2014, 2014.

Schaefli, B. and Gupta, H. V.: Do Nash values have value?, Hydrological Processes, 21, 2075–2080, doi:10.1002/hyp, 2007.

Schenk, H. J. and Jackson, R. B.: The global biogeography of roots, Ecological Monographs, doi:10.1890/0012-9615(2002)072[0311:TGBOR]2.0.CO;2, 2002.

Schmied, H. M., Eisner, S., Franz, D., Wattenbach, M., Portmann, F. T., Flörke, M., and Döll, P.: Sensitivity of simulated global-scale freshwater fluxes and storages to input data, hydrological model structure, human water use and calibration, Hydrology and Earth System Sciences, doi:10.5194/hess-18-3511-2014, 2014.

Sefton, C. E. and Howarth, S. M.: Relationships between dynamic response characteristics and physical descriptors of catchments in England and Wales, Journal of Hydrology, doi:10.1016/S0022-1694(98)00163-2, 1998.

Seibert, J.: Regionalisation of parameters for a conceptual rainfall-runoff model, Agricultural and Forest Meteorology, doi:10.1016/S0168-1923(99)00105-7, 1999.

Seibert, J.: On the need for benchmarks in hydrological modelling, Hydrological Processes, 15, 1063–1064, doi:10.1002/hyp.446, 2001.

Seibert, J., Vis, M. J. P., Lewis, E., and van Meerveld, H.: Upper and lower benchmarks in hydrological modelling, Hydrological Processes, 32, 1120–1125, doi:10.1002/hyp.11476, URL `http://doi.wiley.com/10.1002/hyp.11476`, 2018.

Seiller, G. and Anctil, F.: Climate change impacts on the hydrologic regime of a Canadian river: Comparing uncertainties arising from climate natural variability and lumped hydrological model structures, Hydrology and Earth System Sciences, doi:10.5194/hess-18-2033-2014, 2014.

Sharma, A., Wasko, C., and Lettenmaier, D. P.: If Precipitation Extremes Are Increasing, Why Aren't Floods?, doi:10.1029/2018WR023749, 2018.

Shen, Z. Y., Chen, L., and Chen, T.: Analysis of parameter uncertainty in hydrological and sediment modeling using GLUE method : a case study of SWAT model applied to Three Gorges Reservoir Region , China, Hydrology and Earth System Sciences, 16, 121–132, doi: 10.5194/hess-16-121-2012, 2012.

Shuttleworth, W. J.: Terrestrial Hydrometeorology, John Wiley & Sons, Ltd, Chichester, UK, doi:10.1002/9781119951933, URL http://doi.wiley.com/10.1002/9781119951933, 2012.

Singh, R., van Werkhoven, K., and Wagener, T.: Hydrological impacts of climate change in gauged and ungauged watersheds of the Olifants basin: a trading-space-for-time approach, Hydrological Sciences Journal, doi:10.1080/02626667.2013.819431, 2014.

Sivapalan, M.: The secret to 'doing better hydrological science': Change the question!, doi:10.1002/hyp.7242, 2009.

Sivapalan, M., Takeuchi, K., Franks, S. W., Gupta, V. K., Karambiri, H., Lakshmi, V., Liang, X., McDonnell, J. J., Mendiondo, E. M., O'Connell, P. E., Oki, T., Pomeroy, J. W., Schertzer, D., Uhlenbrook, S., and Zehe, E.: IAHS Decade on Predictions in Ungauged Basins (PUB), 2003-2012: Shaping an exciting future for the hydrological sciences, Hydrological Sciences Journal, doi:10.1623/hysj.48.6.857.51421, 2003.

Smith, A., Bates, P., Freer, J., and Wetterhall, F.: Investigating the application of climate models in flood projection across the UK, Hydrological Processes, 28, 2810–2823, doi:10.1002/hyp.9815, 2014a.

Smith, A., Freer, J., Bates, P., and Sampson, C.: Comparing ensemble projections of flooding against flood estimation by continuous simulation, Journal of Hydrology, 511, 205–219, doi: 10.1016/j.jhydrol.2014.01.045, URL http://dx.doi.org/10.1016/j.jhydrol.2014.01.045, 2014b.

Solomatine, D. P., Dibike, Y. B., and Kukuric, N.: Automatic calibration of groundwater models using global optimization techniques, Hydrological Sciences Journal, 44, 879–894, doi:10.1080/02626669909492287, URL http://www.tandfonline.com/doi/abs/10.1080/02626669909492287, 1999.

Stahl, K., Hisdal, H., Hannaford, J., Tallaksen, L. M., van Lanen, H. A. J., Sauquet, E., Demuth, S., Fendekova, M., and Jódar, J.: Streamflow trends in Europe: evidence from a dataset of near-natural catchments, Hydrology and Earth System Sciences, 14, 2367–2382, doi:10.5194/hess-14-2367-2010, URL https://hess.copernicus.org/articles/14/2367/2010/, 2010.

Steele-Dunne, S., Lynch, P., McGrath, R., Semmler, T., Wang, S., Hanafin, J., and Nolan, P.: The impacts of climate change on hydrology in Ireland, Journal of Hydrology, doi:10.1016/j.jhydrol.2008.03.025, 2008.

Svensson, C., Kundzewicz, Z. W., and Maurer, T.: Trend detection in river flow series: 2. Flood and low-flow index series, Hydrological Sciences Journal, doi:10.1623/hysj.2005.50.5.811, 2005.

Szolgay, J., Hlavčová, K., Kohnová, S., and Danihlík, R.: Regional estimation of parameters of a monthly water balance model, Journal of Hydrology and Hydromechanics, 2003.

Tanguy, M., Dixon, H., Prosdocimi, I., Morris, D., and Keller, V. D. J.: Gridded estimates of daily and monthly areal rainfall for the United Kingdom (1890-2012) [CEH-GEAR], Tech. rep., NERC Environmental Information Data Centre, URL https://doi.org/10.5285/5dc179dc-f692-49ba-9326-a6893a503f6e, 2014.

Teutschbein, C. and Seibert, J.: Bias correction of regional climate model simulations for hydrological climate-change impact studies: Review and evaluation of different methods, Journal of Hydrology, 456-457, 12–29, doi:10.1016/j.jhydrol.2012.05.052, URL http://dx.doi.org/10.1016/j.jhydrol.2012.05.052, 2012.

Teutschbein, C., Grabs, T., Laudon, H., Karlsen, R. H., and Bishop, K.: Simulating streamflow in ungauged basins under a changing climate: The importance of landscape characteristics, Journal of Hydrology, 561, 160–178, doi:10.1016/j.jhydrol.2018.03.060, 2018.

Thiemann, M., Trosset, M., Gupta, H., and Sorooshian, S.: Bayesian recursive parameter estimation for hydrologic models, Water Resources Research, doi:10.1029/2000WR900405, 2001.

Thyer, M., Renard, B., Kavetski, D., Kuczera, G., Franks, S. W., and Srikanthan, S.: Critical evaluation of parameter consistency and predictive uncertainty in hydrological modeling: A case study using Bayesian total error analysis, Water Resources Research, doi:10.1029/2008WR006825, 2009.

Tian, Y., Xu, Y. P., and Zhang, X. J.: Assessment of Climate Change Impacts on River High Flows through Comparative Use of GR4J, HBV and Xinanjiang Models, Water Resources Management, doi:10.1007/s11269-013-0321-4, 2013.

Todini, E.: The ARNO rainfall-runoff model, Journal of Hydrology, 175, 339–382, 1996.

Tran, Q. Q., De Niel, J., and Willems, P.: Spatially Distributed Conceptual Hydrological Model Building: A Generic Top-Down Approach Starting From Lumped Models, Water Resources Research, 54, 8064–8085, doi:10.1029/2018WR023566, URL https://onlinelibrary.wiley.com/doi/abs/10.1029/2018WR023566, 2018.

Trenberth, K.: Changes in precipitation with climate change, Climate Research, 47, 123–138, doi:10.3354/cr00953, URL http://www.int-res.com/abstracts/cr/v47/n1-2/p123-138/, 2011.

Troch, P. A., Carrillo, G. A., Heidbüchel, I., Rajagopal, S., Switanek, M., Volkmann, T. H., and Yaeger, M.: Dealing with landscape heterogeneity in watershed hydrology: A review of recent progress toward new hydrological theory, doi:10.1111/j.1749-8198.2008.00186.x, 2009.

Troy, T. J., Wood, E. F., and Sheffield, J.: An efficient calibration method for continental-scale land surface modeling, Water Resources Research, 44, doi:10.1029/2007WR006513, URL http://doi.wiley.com/10.1029/2007WR006513, 2008.

Van Esse, W. R., Perrin, C., Booij, M. J., Augustijn, D., Fenicia, F., Kavetski, D., and Lobligeois, F.: The influence of conceptual model structure on model performance : a comparative study for 237 French catchments, Hydrology and Earth System Sciences, 17, 4227–4239, doi:10.5194/hess-17-4227-2013, 2013.

van Griensven, A. and Meixner, T.: Methods to quantify and identify the sources of uncertainty for river basin water quality models, Water Science and Technology, 53, 51–59, doi:10.2166/wst.2006.007, URL https://iwaponline.com/wst/article/53/1/51/12478/Methods-to-quantify-and-identify-the-sources-of, 2006.

van Roosmalen, L., Christensen, B. S. B., and Sonnenborg, T. O.: Regional Differences in Climate Change Impacts on Groundwater and Stream Discharge in Denmark, Vadose Zone Journal, doi:10.2136/vzj2006.0093, 2007.

van Werkhoven, K., Wagener, T., Reed, P., and Tang, Y.: Characterization of watershed model behavior across a hydroclimatic gradient, Water Resources Research, 44, 1–16, doi:10.1029/2007WR006271, URL http://doi.wiley.com/10.1029/2007WR006271, 2008.

Vandewiele, G. L., Xu, C. Y., and Huybrechts, W.: Regionalisation of physically-based water balance models in Belgium. Application to ungauged catchments, Water Resources Management, doi:10.1007/BF00421989, 1991.

Vansteenkiste, T., Tavakoli, M., Ntegeka, V., De Smedt, F., Batelaan, O., Pereira, F., and Willems, P.: Intercomparison of hydrological model structures and calibration approaches in climate scenario impact projections, Journal of Hydrology, 519, 743–755, doi:10.1016/j.jhydrol.2014.07.062, URL http://dx.doi.org/10.1016/j.jhydrol.2014.07.062, 2014.

Veijalainen, N., Lotsari, E., Alho, P., Vehviläinen, B., and Käyhkö, J.: National scale assessment of climate change impacts on flooding in Finland, Journal of Hydrology, 391, 333–350, doi:10.1016/j.jhydrol.2010.07.035, URL http://dx.doi.org/10.1016/j.jhydrol.2010.07.035, 2010.

Velázquez, J. A., Anctil, F., and Perrin, C.: Performance and reliability of multimodel hydrological ensemble simulations based on seventeen lumped models and a thousand catchments,

Hydrology and Earth System Sciences, 14, 2303–2317, doi:10.5194/hess-14-2303-2010, URL https://hess.copernicus.org/articles/14/2303/2010/, 2010.

Velázquez, J. A., Anctil, F., Ramos, M. H., and Perrin, C.: Can a multi-model approach improve hydrological ensemble forecasting? A study on 29 French catchments using 16 hydrological model structures, Advances in Geosciences, doi:10.5194/adgeo-29-33-2011, 2011.

Velázquez, J. A., Schmid, J., Ricard, S., Muerth, M. J., Gauvin St-Denis, B., Minville, M., Chaumont, D., Caya, D., Ludwig, R., and Turcotte, R.: An ensemble approach to assess hydrological models' contribution to uncertainties in the analysis of climate change impact on water resources, Hydrology and Earth System Sciences, 17, 565–578, doi:10.5194/hess-17-565-2013, URL https://hess.copernicus.org/articles/17/565/2013/, 2013.

Vrugt, J. A. and Ter Braak, C. J.: DREAM(D): An adaptive Markov Chain Monte Carlo simulation algorithm to solve discrete, noncontinuous, and combinatorial posterior parameter estimation problems, Hydrology and Earth System Sciences, doi:10.5194/hess-15-3701-2011, 2011.

Vrugt, J. A., ter Braak, C. J. F., Clark, M. P., Hyman, J. M., and Robinson, B. A.: Treatment of input uncertainty in hydrologic modeling: Doing hydrology backward with Markov chain Monte Carlo simulation, Water Resources Research, 44, 1–15, doi:10.1029/2007WR006720, URL http://doi.wiley.com/10.1029/2007WR006720, 2008.

Vrugt, J. A., ter Braak, C. J., Gupta, H. V., and Robinson, B. A.: Equifinality of formal (DREAM) and informal (GLUE) Bayesian approaches in hydrologic modeling?, Stochastic Environmental Research and Risk Assessment, doi:10.1007/s00477-008-0274-y, 2009.

Wada, Y., Bierkens, M. F. P., de Roo, A., Dirmeyer, P. A., Famiglietti, J. S., Hanasaki, N., Konar, M., Liu, J., Müller Schmied, H., Oki, T., Pokhrel, Y., Sivapalan, M., Troy, T. J., van Dijk, A. I. J. M., van Emmerik, T., Van Huijgevoort, M. H. J., Van Lanen, H. A. J., Vörösmarty, C. J., Wanders, N., and Wheater, H.: Human–water interface in hydrological modelling: current status and future directions, Hydrology and Earth System Sciences, 21, 4169–4193, doi:10.5194/hess-21-4169-2017, URL https://hess.copernicus.org/articles/21/4169/2017/, 2017.

Wade, S. D., Rance, J., and Reynard, N.: The UK Climate Change Risk Assessment 2012: Assessing the Impacts on Water Resources to Inform Policy Makers, Water Resources Management, 27, 1085–1109, doi:10.1007/s11269-012-0205-z, URL http://link.springer.com/10.1007/s11269-012-0205-z, 2013.

Wagener, T. and Gupta, H. V.: Model identification for hydrological forecasting under uncertainty, Stochastic Environmental Research and Risk Assessment, doi:10.1007/s00477-005-0006-5, 2005.

Wagener, T. and Kollat, J.: Numerical and visual evaluation of hydrological and environmental models using the Monte Carlo analysis toolbox, Environmental Modelling and Software, doi: 10.1016/j.envsoft.2006.06.017, 2007.

Wagener, T. and Wheater, H. S.: Parameter estimation and regionalization for continuous rainfall runoff models including uncertainty, Journal of Hydrology, 320, 132–154, doi:10.1016/j.jhydrol. 2005.07.015, 2006.

Wagener, T., Wheater, H. S., and Gupta, H. V.: Rainfall-Runoff Modelling in Gauged and Ungauged Catchments, Published by Imperial College Press and distributed by World Scientific Publishing Co., doi:10.1142/p335, URL https://www.worldscientific.com/worldscibooks/10.1142/p335, 2004.

Wagener, T., Sivapalan, M., Troch, P. A., McGlynn, B. L., Harman, C. J., Gupta, H. V., Kumar, P., Rao, P. S. C., Basu, N. B., and Wilson, J. S.: The future of hydrology: An evolving science for a changing world, Water Resources Research, 46, 1–10, doi:10.1029/2009WR008906, 2010.

Wanders, N., Wood, E., Pan, M., Samaniego, L., Thober, S., and Kumar, R.: Multiscale Parameter Regionalization for consistent global water resources modelling, EGU General Assembly 2017, 19, 9621, URL https://meetingorganizer.copernicus.org/EGU2017/EGU2017-9621-2.pdf, 2017.

Wang-Erlandsson, L., Van Der Ent, R. J., Gordon, L. J., and Savenije, H. H.: Contrasting roles of interception and transpiration in the hydrological cycle - Part 1: Temporal characteristics over land, Earth System Dynamics, doi:10.5194/esd-5-441-2014, 2014.

Wang-Erlandsson, L., Bastiaanssen, W. G., Gao, H., Jägermeyr, J., Senay, G. B., Van Dijk, A. I., Guerschman, J. P., Keys, P. W., Gordon, L. J., and Savenije, H. H.: Global root zone storage capacity from satellite-based evaporation, Hydrology and Earth System Sciences, doi:10.5194/hess-20-1459-2016, 2016.

Watts, G., Battarbee, R. W., Bloomfield, J. P., Crossman, J., Daccache, A., Durance, I., Elliott, J. A., Garner, G., Hannaford, J., Hannah, D. M., Hess, T., Jackson, C. R., Kay, A. L., Kernan, M., Knox, J., Mackay, J., Monteith, D. T., Ormerod, S. J., Rance, J., Stuart, M. E., Wade, A. J., Wade, S. D., Weatherhead, K., Whitehead, P. G., and Wilby, R. L.: Climate change and water in the UK – past changes and future prospects, Progress in Physical Geography: Earth and Environment, 39, 6–28, doi:10.1177/0309133314542957, URL https://apps.webofknowledge.com/full_record.do?product=UA&search_mode=GeneralSearch&qid=25&SID=Y1CSpljUMDRdT4RgCA2&page=1&doc=10http://journals.sagepub.com/doi/10.1177/0309133314542957, 2015.

Westerberg, I. K. and Birkel, C.: Observational uncertainties in hypothesis testing: Investigating the hydrological functioning of a tropical catchment, Hydrological Processes, doi:10.1002/hyp. 10533, 2015.

Westerberg, I. K., Wagener, T., Coxon, G., McMillan, H. K., Castellarin, A., Montanari, A., and Freer, J.: Uncertainty in hydrological signatures for gauged and ungauged catchments, Water Resources Research, 52, 1847–1865, doi:10.1002/2015WR017635, URL http://doi.wiley.com/10.1002/2015WR017635, 2016.

Wheater, H. S.: Flood hazard and management: a UK perspective, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 364, 2135–2145, doi:10.1098/rsta.2006.1817, URL http://www.ncbi.nlm.nih.gov/pubmed/16844652http://rsta.royalsocietypublishing.org/cgi/doi/10.1098/rsta.2006.1817, 2006.

Wilby, R. L. and Harris, I.: A framework for assessing uncertainties in climate change impacts: Low-flow scenarios for the River Thames, UK, Water Resources Research, 42, 1–10, doi:10.1029/2005WR004065, URL http://doi.wiley.com/10.1029/2005WR004065, 2006.

Wilby, R. L., Beven, K. J., and Reynard, N. S.: Climate change and fluvial flood risk in the UK: more of the same?, Hydrological Processes, 22, 2511–2523, doi:10.1002/hyp.6847, URL http://jamsb.austms.org.au/courses/CSC2408/semester3/resources/ldp/abs-guide.pdfhttp://doi.wiley.com/10.1002/hyp.6847, 2008.

WMO: Intercomparison of conceptual models used in operational hydrological forecasting, Operational Hydrology Report no. 07, 1975.

WMO: Intercomparison of models of snowmelt runoff, Operational Hydrology Report no. 23, 1986.

Wood, A. and Mizukami, N.: Project summary report: CMIP5 1/8 degree daily weather and vic hydrology datasets for CONUS, Tech. rep., NCAR, Boulder, Colorado, 2014.

Wood, E. F., Roundy, J. K., Troy, T. J., van Beek, L. P. H., Bierkens, M. F. P., Blyth, E., de Roo, A., Döll, P., Ek, M., Famiglietti, J., Gochis, D., van de Giesen, N., Houser, P., Jaffé, P. R., Kollet, S., Lehner, B., Lettenmaier, D. P., Peters-Lidard, C., Sivapalan, M., Sheffield, J., Wade, A., and Whitehead, P.: Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth's terrestrial water, Water Resources Research, 47, 1–10, doi:10.1029/2010WR010090, URL http://doi.wiley.com/10.1029/2011WR010982http://doi.wiley.com/10.1029/2010WR010090, 2011.

Yadav, M., Wagener, T., and Gupta, H.: Regionalization of constraints on expected watershed response behavior for improved predictions in ungauged basins, Advances in Water Resources, 30, 1756–1774, doi:10.1016/j.advwatres.2007.01.005, 2007.

Yang, J., Reichert, P., Abbaspour, K. C., and Yang, H.: Hydrological modelling of the Chaohe Basin in China: Statistical model formulation and Bayesian inference, Journal of Hydrology, doi:10.1016/j.jhydrol.2007.04.006, 2007.

Yang, J., Reichert, P., Abbaspour, K. C., Xia, J., and Yang, H.: Comparing uncertainty analysis techniques for a SWAT application to the Chaohe Basin in China, Journal of Hydrology, doi:10.1016/j.jhydrol.2008.05.012, 2008.

Yang, Y., Donohue, R. J., and McVicar, T. R.: Global estimation of effective plant rooting depth: Implications for hydrological modeling, Water Resources Research, doi:10.1002/2016WR019392, 2016.

Yatheendradas, S., Wagener, T., Gupta, H., Unkrich, C., Goodrich, D., Schaffner, M., and Stewart, A.: Understanding uncertainty in distributed flash flood forecasting for semiarid regions, Water Resources Research, doi:10.1029/2007WR005940, 2008.

Yen, H., Wang, X., Fontane, D. G., Harmel, R. D., and Arabi, M.: A framework for propagation of uncertainty contributed by parameterization, input data, model structure, and calibration/validation data in watershed modeling, Environmental Modelling and Software, doi: 10.1016/j.envsoft.2014.01.004, 2014.

Young, A. R.: Stream flow simulation within UK ungauged catchments using a daily rainfall-runoff model, Journal of Hydrology, 320, 155–172, doi:10.1016/j.jhydrol.2005.07.017, 2006.

Zeng, X.: Global Vegetation Root Distribution for Land Modeling, Journal of Hydrometeorology, doi:10.1175/1525-7541(2001)002<0525:gvrdfl>2.0.co;2, 2001.

Zhang, Y. and Schaap, M. G.: Estimation of saturated hydraulic conductivity with pedotransfer functions: A review, Journal of Hydrology, doi:10.1016/j.jhydrol.2019.05.058, 2019.

Zhang, Y., Hou, J., Gu, J., Huang, C., and Li, X.: SWAT-Based Hydrological Data Assimilation System (SWAT-HDAS): Description and Case Application to River Basin-Scale Hydrological Predictions, Journal of Advances in Modeling Earth Systems, doi:10.1002/2017MS001144, 2017.