GLOBGM v1.0: a parallel implementation of a 30 arcsec PCR-GLOBWB-MODFLOW global-scale groundwater model

Jarno Verkaik[1,2], Edwin H. Sutanudjaja[1], Gualbert H.P. Oude Essink[1,2], Hai Xiang Lin[3,4], Marc F.P. Bierkens[2,1]

5 [1]Unit Subsurface and Groundwater Systems, Deltares, Utrecht, the Netherlands
[2]Department of Physical Geography, Faculty of Geosciences, Utrecht University, Utrecht, the Netherlands
[3]Department of Applied Mathematical Analysis, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, the Netherlands
[4]Institute of Environmental Sciences, Faculty of Science, Leiden University, Leiden, the Netherlands

10 *Correspondence to*: Jarno Verkaik (Jarno.Verkaik@deltares.nl)

Abstract

We discuss the various performance aspects of parallelizing our global-scale groundwater model at 30″ resolution (30 arcseconds; ~1 km at the equator) on large distributed memory parallel clusters. This model, here referred to as the GLOBGM, is the successor of our 5′ (5 arcminutes; ~10 km at the equator) PCR-GLOBWB 2 groundwater model based on MODFLOW

15 having two model layers. The current version of the GLOBGM (v1.0) used in this study also has two model layers, is uncalibrated, and uses available 30″ PCR-GLOBWB data. Increasing the model resolution from 5′ to 30″ gives challenges for dealing with the increasing runtime, memory usage and data storage, going beyond the possibilities of a single computer. We show that our parallelization tackles these problems with relatively low parallel hardware requirement to meet average users/modelers who do not have exclusive access to hundreds or thousands of nodes within a supercomputer.

20 For our simulation we use unstructured grids and a prototype version of MODFLOW 6 that we have parallelized using the message passing interface. We construct an unstructured grid having a total of 278 million active cells to cancel all redundant sea and land cells, while satisfying all necessary boundary conditions, and distribute independent (sub)grids over three continental-scale models (Afro-Eurasia; 168 M, Americas; 77 M, and Australia; 16 M) and one remainder model for the smaller islands (17 M). Each of the four groundwater models is partitioned into multiple non-overlapping submodels that are tightly

25 coupled within the MODFLOW linear solver, where each submodel is uniquely assigned to one processor core and associated submodel data is written in parallel during the pre-processing using data tiles. For balancing the parallel workload in advance, we apply the widely used METIS graph partitioner in two ways: straightforwardly applied to all model grid cells, and area-based applied to HydroBASINS catchments that are assigned to submodels for pre-sorting to a future coupling with surface water. We consider an experiment for simulating 1958-2015 with daily timesteps and monthly input, including a 20-year spin-

30 up, on the Dutch national supercomputer Snellius. Given that the serial simulation would require ~4.5 months of runtime, we set a hypothetical target of a maximum of 16 hours of simulation runtime. We show that 12 nodes (32 cores per node, 384 cores in total) are sufficient to achieve this target, resulting in a speed-up of 138 for the largest Afro-Eurasia model using 7 nodes (224 cores) in parallel.

A limited evaluation of the model output using NWIS head observations for the contiguous United States was conducted,

35 showing that increasing the resolution from 5′ to 30″ results in a significant improvement with GLOBGM for the steady-state simulation compared to the 5′ PCR-GLOBWB groundwater model. However, results for the transient simulation are quite similar and there is much room for improvement. For next versions of the GLOBGM further improvements require a more detailed hydrogeological schematization and better information on the locations, depths and regime of abstraction wells.

1 Introduction

40   The PCRaster Global Water Balance model (PCR-GLOBWB; van Beek et al., 2011) is a grid-based global-scale hydrology and water resource model for the terrestrial part of the hydrologic cycle, being developed at the Utrecht University, The Netherlands. This model, using a Cartesian (regular) grid representation for the geographic coordinate system (latitude and longitude), covers all continents except Greenland and Antarctica. For more than a decade, it has been applied to many water-related global change assessments providing estimates and future projections, e.g. regarding drought and groundwater
45   depletion due to non-renewable groundwater withdrawal (Wada et al., 2010; de Graaf et al., 2017).

The latest version of PCR-GLOBWB (version 2.0; Sutanudjaja et al., 2018), or PCR-GLOBWB 2, has a spatial resolution of 5′ (5 arcminutes) corresponding to ~10 km the equator. It includes a 5′ global-scale groundwater model (GGM) based on MODFLOW (Harbaugh, 2005), consisting of two model layers to account for confined and unconfined aquifers (de Graaf et al., 2015, 2017). We consider a spatially refined version of this model 30″ (~1 km at the equator), here referred to as the
50   GLOBGM. The initial version of the GLOBGM, v1.0, that we use in this study, has a maximum of two model layers and uses refined input from available PCR-GLOBWB data. Improving schematization, e.g. by adding more model layers, as well as obtaining more refined input parameters and subsequent model calibration, we therefore leave for further research and for next versions of the GLOBGM.

55   Motivated by the need for better groundwater estimates and future projections of anthropogenic effects and climate change stresses, pushing forward to 30″ resolution is a direct result for the growing availability of high-resolution datasets and the wish for exploiting the benefits of high performance computing (HPC) to maximize computer power and modeling capabilities (Wood et al., 2011; Bierkens et al., 2015). Typically, high-resolution Digital Elevation Maps (DEMs) are available derived from NASA Shuttle Radar Topography Mission products, such as HydroSHEDS (Lehner et al., 2008) at 30″ or the MERIT
60   DEM (Yamazaki et al., 2017) at even 3″. Furthermore, subsurface data are becoming available in higher resolution, such as gridded soil properties at 250 m resolution from SoilGrids (Hengl et al., 2014), and global lithologies for ~1.8 million polygons from GLHYMPS (Huscroft et al., 2018). Although Moore's law still holds, viz. stating that processor performance doubles every two years, more and more processor cores are being added to increase performance. This means that making software suitable for HPC is becoming more and more inevitable for efficiently using the available hardware. Anticipating this, we
65   parallelized the groundwater solver for the GLOBGM as well as the pre-processing.

In this paper, the focus in on the technical challenges of implementing the GLOBGM using HPC. Furthermore, this study is mainly on transient simulation, since transient simulations make most sense for evaluating the effects of climate change and human interventions (see e.g. Minderhoud et al., 2017). We apply a similar approach as de Graaf et al. (2017) to obtain initial
70   conditions for the GLOBGM: we use a steady-state result (under natural conditions; no pumping) to spin-up the model by running the first year back-to-back for 20 years to reach dynamic equilibrium. We restrict ourselves to present parallel performance results for transient runtimes only, since we found that steady-state runtimes with the GLOBGM are negligible compared to transient runtimes. We present a parallelization methodology for the GLOBGM and illustrate this by a transient experiment on the Snellius Dutch national supercomputer (SURFsara, 2021) for simulating 1958-2015 (58 years), including a
75   20-year spin-up. We give a limited evaluation of the computed results and believe this provides a good starting point for further research (Gleeson et al., 2021). For this, the steady-state and transient results with the GLOBGM are compared to groundwater head observations for the contiguous United States (CONUS), also considering the 5′ PCR-GLOBWB groundwater model, the 30″ global-scale inverse model of Fan et al. (2017), and the 250 m groundwater model at the CONUS-extent of Zell and Sanford (2020).

80

We use the model code program MODFLOW (Langevin et al., 2021, 2017), the most widely used groundwater flow modeling program in the world, being developed by the U.S. Geological Survey (USGS). The latest version 6 of MODFLOW supports a multi-model functionality that enable users to set up a model as a set of (spatially non-overlapping) submodels, where each submodel is tightly connected to other submodels at matrix level and has its own unique input and output files. In

85 cooperation with the USGS, we parallelized this submodel functionality and created a prototype version (Verkaik et al., 2018, 2021c) that is publicly available and is planned to be part of a coming MODFLOW release. Building on our preceding research (Verkaik et al., 2021a, b), this prototype uses the message passing interface (MPI; MPI Forum, 1994) to parallelize the conjugate gradient linear solver within the iterative model solution package (Hughes et al., 2017), supporting the additive Schwarz preconditioner as well as the additive coarse grid correction preconditioner (Smith et al., 1996).

90

This paper is organized as follows. In Section 2, first, the general approach for implementing the GLOBGM is given in Section 2.1, followed by the experimental set-up in Section 2.2, the workflow description in Section 2.3 and model evaluation methods in Section 2.4. In Section 3, the results are presented and discussed for the transient pre-processing in Section 3.1, the transient parallel performance in Section 3.2, the model evaluation in Section 3.3, and some examples of global-scale results

95 are given in Section 3.4. Section 4 concludes this paper.

2 Methods

2.1 General approach

The 5′ PCR-GLOBWB MODFLOW global-scale groundwater model consists of two model layers: the upper model layer represents the confining layer and the lower model layer representing the (confined or unconfined) aquifer (de Graaf et al.,

100 2015, 2017). This model uses a structured Cartesian grid, representing latitude and longitude, and includes all land and sea cells at the global 5′ extent. Inherently to using a structured grid, where all model layers are required to be contiguous in space, sea cells are used as lateral Dirichlet boundary condition (BC) of a head equal to 0 m. Furthermore, to account for the absence of the confining layer, this model uses zero thickness (dummy) cells for the upper layer as a work-around for specifying a lateral homogeneous Neumann BC (no-flow). In total, each model layer has 4,320 columns and 2,160 rows, hence this model

105 has total of ~9.3 million 5′ cells, where many cells are redundant. Since the Dirichlet BC is only required for connections to land near the coastline, this means that ~77% of the 5′ grid cells corresponding to sea are redundant. Furthermore, for the land cells, ~35% of the cells in this model are dummy cells with zero thickness to account for the homogeneous Neumann BC. A straightforward refinement of the 5′ grid to 30″ would result in ~100 times more cells, hence 1.87 billion cells (two model layers of 43,200 columns and 21,600 rows). Creating and using such a model would heavily stress runtime, memory usage and

110 data storage.

By exploiting the benefits of unstructured grids for cancelling redundant cells associated with boundary conditions, we can significantly reduce the number of cells. Furthermore, this allows us to apply a static asynchronous parallelization approach to maximize parallelism. First, sea cells are added to the land cells in the lateral direction by extrapolation of the 30″ land mask

115 using a 5-point stencil, hence accounting for the Dirichlet BC. Second, cells are added in the vertical direction (one or two) to account for the presence of a confining layer, and therefore account for the lateral homogeneous BC. This means that in the GLOBGM two configurations are used: 1. Upper confining layer + lower (confined) aquifer (2 vertical cells); 2. Lower (unconfined) aquifer (1 vertical cell). Although we cannot distinguish between upper and lower model layer anymore when using unstructured grids, from here, we sometimes use "upper model layer" and "confining layer" interchangeably, as well as

120 "lower model layer" and "aquifer".

After decoupling and sorting by cell count, we end up with $N_g = 9050$ independent/disjoint unstructured grids G_n, $n = 1, \ldots, N_g$, having a total of ~278 million 30″ cells. This gives a reduction of 85% compared to a straightforward refinement of the 5′ structured grid (land: 35% reduction, sea: 99.9% reduction). These grids correspond to continents and islands, with the three largest: Afro-Eurasia (AE, 168 million 30″ cells), Americas (AM, 77 million 30″ cells) and Australia (AU, 16 million

125   30″ cells), see Figure 1. The remainder of 9047 smaller grids, together for a total of 17 million 30″ cells, are grouped as "Islands" (ISL). By doing this, we define the GLOBGM as a set of four independent models ($N_m = 4$) subject to parallel solving, of which three of them are continental-scale groundwater models (CGM). It should be noted that more than four models can be chosen, up to a maximum of 9050. However, because the Islands model has almost the same number of cells as the Australia model, we do not feel any need for using more models than four.

130

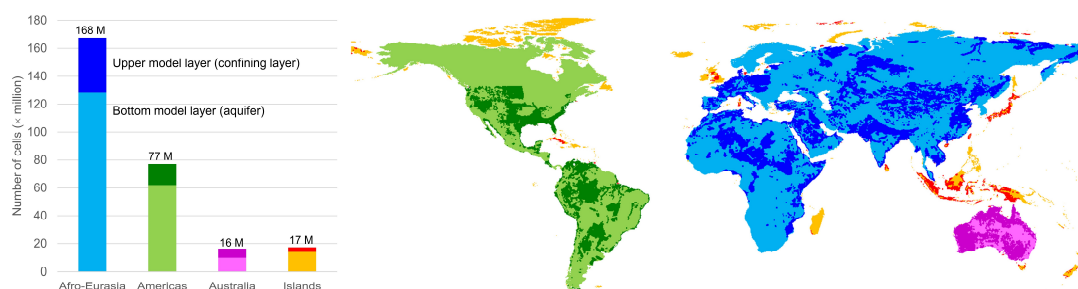



Figure 1. The four defined model areas within the GLOBGM having a total of 278.3 million cells. When the upper model layer is plotted this means that there is also a lower model layer present. Otherwise, only the lower model layer is present.

135   For evaluating the parallel performance, we solve these four groundwater models using distributed memory parallel computing (Rünger and Rauber, 2013). We restrict ourselves to parallelization on mainstream cluster computers since they are often accessible to geohydrological modelers. Cluster computers typically have distributed memory (computer) nodes, where each node consists of several multicore-CPUs (sockets) sharing memory, and each node is tightly connected to other nodes by a fast (high-bandwidth and low-latency) interconnection network.

140

Instead of fixating on parallel speedups and scalability, which are commonly used metrics for benchmarking parallel codes (e.g. Burstedde et al., 2018), we choose a metric for which we believe has good connection with the experience of users: simulated-years-per-day (SYPD). This metric, simply the number of years that a model can be simulated in a single day of 24 hours, has proven to be useful for evaluating massively parallel performance in the field of atmospheric community modeling

145   (Zhang et al., 2020). We present a straightforward, but effective, method to obtain a desired runtime target performance in SYPD, R_{tgt}, meeting user requirements. Here, we consider the typical "9 to 5" user, who likes to start a simulation at 5 PM and get the results next working day at 9 AM. The target for our transient experiment for 1958-2015 is then to simulate 78 years (58 + 20 years spin-up) in 16 hours. Accordingly, this is 0.67 days, and therefore we set $R_{tgt} = 78 / 0.67 = 117$ SYPD. We also use the metric core-hours-per-simulated-year (CHPSY), that is defined as the cumulative runtime over all processor

150   cores being used for simulation a single year. With CHPSY, the actual parallel runtime can be easily obtained by multiplying CHPSY with the number of simulated years and consecutively dividing by the number of processor cores being used.

For parallelizing the GLOBGM, each of the four groundwater models is partitioned into multiple non-overlapping submodels, where each submodel is uniquely assigned to one processor core and one MPI process. To obtain a good parallel

155   performance, computational work associated with the submodels should be well balanced. For this, we perform a grid

partitioning using METIS (Karypis and Kumar, 1998), the most commonly used graph partitioner among others like Chaco (Hendrickson and Leland, 1995) and Scotch (Pellegrini, 2008). Input for this partitioner is an undirected graph having node weights, edge weights, and the number of partitions. Here, we restrict our grid partitioning to lateral direction only, since the number of lateral cells is much larger than the number of model layers (here a maximum of two). This naturally minimizes the 160 number of connections between the submodels and the associated point-to-point (inter-core) MPI communication times.

For each model m^i, $1 \leq i \leq N_\mathrm{m}$, a set of (coupled) submodels is defined by $m^{i,j}$, $j = 1, \ldots, N_\mathrm{m}^i$, where N_m^i is the number of submodels for model i. In our approach, each submodel is constructed by combining one or more areas $a_k^{i,j}$, $k = 1, \ldots, N_\mathrm{a}^{i,j}$, where $N_\mathrm{a}^{i,j}$ is the number of areas for model i and submodel j. The total number of areas for a model is defined by $N_\mathrm{a}^i = \cup_j N_\mathrm{a}^{i,j}$, and for the entire GLOBGM by $N_\mathrm{a} = \cup_i N_\mathrm{a}^i$. Here, the general approach is to assign one or more areas to a 165 submodel using METIS. From here, we refer to this as area-based graph partitioning. Here, two partitioning strategies are considered, using:

1. METIS areas that are generated by straightforwardly applying METIS to model grids for m^i to obtain N_m^i parts. This we refer to as straightforward METIS partitioning.

2. Catchment areas that are generated by rasterization and extrapolation of HydroBASINS catchments including lakes 170 (Lehner, 2014) for a given Pfafstetter refinement level (Verdin and Verdin, 1999). This we refer to as catchment partitioning.

Considering METIS areas, we restrict ourselves to assigning exactly one area to a submodel, hence $N_\mathrm{a}^{i,j} = 1$ for all j. Each node in the graph corresponds to a lateral cell having a node weight equal to the number of vertical cells (1 or 2), and one or more edge weights equal to the number of inter-cell connections to neighboring cells (1 or 2). In our study, this is the 175 most optimal partitioning method that we apply to the GLOBGM.

Assignment of catchment areas to a submodel, typically $N_\mathrm{a}^{i,j} \gg 1$, is done by partitioning a graph using METIS, where each node in the graph corresponds to a catchment area having a node weight, equal to the total number of grid cells within that catchment, and one or more edge weights, equal the total number of inter-cell connections to neighboring catchments. Compared to straightforward METIS partitioning, catchment partitioning is less optimal since it optimizes a graph that is much 180 more lumped. However, catchment partitioning has several advantages. First, the graphs subject to METIS partitioning are generally much smaller and therefore partitioning is significantly faster. In fact, this approach makes the grid partitioning more or less independent on the grid cell resolution and is therefore suitable for even future higher resolutions. Second, such approach gives users flexibility for giving the lateral submodel boundaries physical meaning. For example, choosing catchment boundaries simplifies a future parallel coupling to surface water routing modules (Vivoni et al., 2011). Third, this concept 185 could be easily generalized for other types of areas, e.g. countries or states following administrative boundaries. Choosing such areas might simplify the management and maintenance of the submodels by different stakeholders within a community model.

Figure 2 illustrates the defined model entities for the case of four islands $(N_\mathrm{g} = 4)$, two models $(N_\mathrm{m} = 2)$ and seven 190 catchments areas $(N_\mathrm{a} = 7)$. In this example model m^1 has the largest grid g_1, and model m^2 the remaining smaller grids g_2, g_3 and g_4. The denoted cell numbers in Figure 2 correspond to the numbers of model layers, where blue denotes a cell with a Dirichlet (sea) BC. In this example, each model has exactly two submodels (hence $N_\mathrm{m}^1 = N_\mathrm{m}^2 = 2$). For model m^1, catchments $a_1^{1,1}$ and $a_2^{1,1}$ are assigned to submodel $m^{1,1}$, and $a_1^{1,2}$ is assigned to submodel $m^{1,2}$. For model m^2, catchment $a_1^{2,1}$ is assigned to submodel $m^{2,1}$, and $a_1^{2,2}$, $a_2^{2,2}$ and $a_3^{2,2}$ are assigned to submodel $m^{2,2}$. For this example, the load of model m^1 is well-

195 balanced since each submodel has a load of 50. However, for model m^2, the first and second submodel have load 19 and 21,

respectively, and therefore there is a load imbalance. Note that submodel $m^{2,2}$ has three disjoint subgrids, since in this case

three islands are involved. In general, the presence of disjoint subgrids may occur for any submodel within the GLOBGM, as

a result of partitioning a continent or island with METIS. The reason is that none of the METIS solvers can guarantee

contiguous partitions. However, we here use multilevel recursive bisection, that is known to give best results in that respect.

200

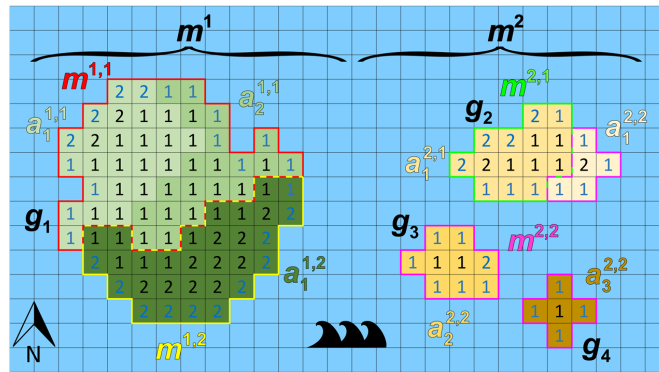



Figure 2. Example where four islands are partitioned for two models, m^1 and m^2 .

When evaluating the target performance R_{tgt} , we restrict ourselves to model runtime only. Since pre- and post-processing

are very user specific and therefore associated runtimes cannot be generalized, we do include them in our analysis. However,

205 processing runtimes can be substantial making parallel processing inevitable, as we will illustrate for our limited pre-

processing. To keep pre-processing runtimes feasible, we apply embarrassing parallelization (see e.g. Herlihy et al., 2020)

where processing for input data (tiles and submodels) is done independently using multiple threads simultaneously on multiple

nodes.

For achieving R_{tgt} , we need a selection procedure for estimating the number of (processor) cores for the three CGMs and

210 the Islands model. Since it is common to take the number of cores per node at a fixed value (N_{CPN}) , this means that choosing

the number of cores is equivalent to choosing the number of nodes. Ideally, we would take N_{CPN} to be equal to the total

number of cores within a node and therefore maximize computer resources utilization. However, it is not always advantageous

or even impossible to use the total number of cores due to the memory access constraint. In practice, the best performance is

often obtained using a lesser number of cores. For example, the linear conjugate gradient solver that is used for solving the

215 groundwater flow equation, dominating runtime, is strongly memory-bound because of the required sparse matrix-vector

multiplications (Gropp et al., 1999). This means that, starting from a certain number of cores, competition for memory

bandwidth hampers parallel performance and contention is likely to occur (Tudor et al., 2011). Typically, this directly relates

to the available memory channels within a multicore-CPU, linking the RAM and processor cores. In our approach, we first

determine N_{CPN} by performing a strong scaling experiment (keeping the problem size fixed) within a single node for the well-

220 known high performance conjugate gradients supercomputing benchmark (HPCG; Dongarra et al., 2016). By doing this, we

assume that HPCG is representative for MODFLOW. The performance metric for this experiment is floating-point operations

per second (FLOPS), that is commonly used for quantifying numerical computing performance and processor speed. Then, by

conducting a strong scaling experiment for a single (medium-sized) model and a short simulation period while keeping N_{CPN}

fixed, the number of nodes is selected such that R_{tgt} is achieved. This gives the preferred submodel size that is used to

225 determine the number of nodes for all other models; see section 2.1 for more details.

2.2 Experimental set-up

2.2.1 Description of the GLOBGM

Table 1 and Table 2 show the datasets used to parameterize the GLOBGM in this study. For details on the MODFLOW model description and conceptualization, we refer to the preceding papers (de Graaf et al., 2019, 2017; Sutanudjaja et al., 2014, 2011). In this study, we adopted an offline coupling approach that used the PCR-GLOBWB output of Sutanudjaja et al. (2018) as the input to the MODFLOW groundwater model. The PCR-GLOBWB output that was used consists of monthly fields for the period 1958-2015 for the variables groundwater recharge (i.e., net recharge obtained from deep percolation and capillary rise), groundwater abstraction, and runoff. The latter was translated to monthly surface water discharge by accumulating it through the 30″ river/drainage network of HydroSHEDS (Lehner et al., 2008). From the monthly surface water discharge fields, we then estimated surface water levels using Manning's equation (Manning, 1891).

The groundwater model simulation conducted consists of two parts: a steady-state and a transient simulation. For both simulations, default solver settings from the 5′ model were taken for evaluating convergence, and within the linear solver we did not apply coarse grid correction. We started with a steady-state MODFLOW model using the average PCR-GLOBWB runoff and groundwater recharge as the input. No groundwater abstraction was assumed for the steady-state model therefore representing a naturalized condition, as the simulated steady-state groundwater heads will be used as the initial conditions for the transient simulation (i.e., assuming low pumping in ~1958). The transient MODFLOW simulation was calculated at daily time steps with monthly stress period input of surface water levels, groundwater recharge and groundwater abstraction. Using the steady-state estimate as the initial groundwater heads, the model was spun-up using the year 1958 input for 20 years (to further warm-up the initial states and add small effect of pumping), before the actual transient simulation for the period 1958-2015 started.

Table 1. Input datasets used for the GLOBGM transient model parameters.

Item	Number of grids (fixed / monthly)	Main sources/references, original spatial resolution/support	Remarks
Model upper layer top elevation, DEM (digital elevation model)	1 / -	MERIT Hydro DEM (Yamazaki et al., 2017), 3" resolution	We upscaled the 3" MERIT DEM to 30" resolution, e.g., for the estimate of surface elevation. Yet, we also used its original 3" values to derive several other elevation values (at 30" resolution), such as the elevations of flood plain, river head, river bottom, and drainage bottom. For detailed descriptions, we refer to the preceding papers (de Graaf et al., 2019, 2017; Sutanudjaja et al., 2014, 2011).
Model layer thicknesses (confining layer and aquifer thicknesses)	2 / -	de Graaf et al., 2017; Sutanudjaja et al., 2018, 5' resolution	We performed a bilinear interpolation to bring them to 30" resolution.
Hydraulic conductivities and storativities (storage coefficients)	6 / - (2 for horizontal conductivities, 2 for vertical conductivities, and 2 for storativities)	GLHYMPS (Gleeson et al., 2014); GLiM (Hartmann and Moosdorf, 2012) a polygon map with the scale 1:3,750,000	We rasterize the polygon map of GLiM (Hartmann and Moosdorf, 2012) to 30" resolution. The values assigned for each class are based on GLHYMPS (Gleeson et al., 2014).
Groundwater recharge	- / 1	PCR-GLOBWB (Sutanudjaja et al., 2018), 5' resolution	We simply resampled/mapped this map to 30" (no downscaling)
Groundwater abstraction (pumping wells)	- / 2	PCR-GLOBWB (Sutanudjaja et al., 2018), 5' resolution	We simply resampled/mapped this map to 30" (no downscaling)
River package	- / 3	See Table 2 for the input for river and drainage packages.	See Table 2 for the input for river and drainage packages.
Drainage package	1 / 2	See Table 2 for the input for river and drainage packages.	See Table 2 for the input for river and drainage packages.

250   **Table 2.** Input datasets used for the MODFLOW river and drainage packages.

Item	Main sources/references, original spatial resolution/support	Remarks
River/drainage network	HydroSHEDS (Lehner et al., 2008)	Used as the network to accumulate local runoff for estimating surface water discharge.
Local runoff	PCR-GLOBWB 2 (Sutanudjaja et al., 2018), 5' resolution	Used to derive surface water discharge and surface water level fields.
Surface water discharge and surface water level	This study, 30" resolution	Surface water discharge was estimated by accumulating runoff through the 30" river/drainage network of HydroSHEDS (Lehner et al., 2008). Then, surface water level was calculated from discharge using the Manning's equation (Manning, 1891).
Groundwater recession coefficient (used to estimate drainage conductance)	This study, 30" resolution	Calculated following the drainage theory of van de Leur Kraijenhoff (1958) based on the drainage network density and aquifer properties. For the drainage density, we used the estimate from van van Beek and Bierkens (2009). The aquifer properties were from Gleeson et al. (2014).

2.2.2   Dutch national supercomputer Snellius

All our experiments were conducted on the Dutch national supercomputer Snellius (SURFsara, 2021), a cluster computer using multicore CPUs that is easily accessible to users from Dutch universities and Dutch research institutes. It consists of heterogeneous distributed memory nodes (servers), where we restrict to using default worker nodes having 256 GB[1] memory
255   ("thin" nodes), see Table 3, that are tightly connected through a fast interconnection network having a low latency and high bandwidth. Each node houses two 64-core AMD CPUs that connect to 256 GB of local memory through two sockets. For this study, a storage of 50 TB and a maximum of 3.8 million files is used that is tightly connected to the nodes, enabling parallel I/O using the Lustre parallel distributed file system. Since the Snellius supercomputer is a (more or less) mainstream cluster built with off-the-shelf hardware components, we believe that our parallelization approach is well applicable to many other
260   supercomputers.

Table 3. Configuration of the default computing nodes used on the Dutch national supercomputer Snellius (SURFsara, 2021).

Number of nodes	504
Number of sockets per node	2
Multicore-CPU per socket	AMD EPYC 7H12 (2nd gen. Rome)
	64 cores, 2.6 GHz, 8 memory channels
Memory (RAM) per node	256 GB, 3200 MHz, DDR4 (16 DIMMs)
Interconnection network	Fat tree topology, InfiniBand HDR100
File system	Lustre parallel with striping, InfiniBand HDR100

2.3 Workflow description

2.3.1   Main Workflow and Node Selection Workflow

265   The main workflow for pre-processing and running the four models from the GLOBGM is given by Figure 3a. The so-called Model Workflow uses data that are initially being generated by the process "Write Tiled Parameter Data", that writes the 30″ grids (see Table 1) using a tiled-based approach in an embarrassingly parallel way, see Section 2.3.2 for more details. For the Model Workflow, prior to the actual partitioning and writing the model files in an embarrassingly parallel way (viz. the process "Partition & Write Model Input"; see Section 2.3.4), a preparation step is done for the partitioning (the process

---

[1] The abbreviations GB (gigabyte) and TB (terabyte) denote the size of the binary (base-2) memory system.

"Prepare Model Partitioning"; see Section 2.3.3). The general idea behind this preparation step is to simplify the embarrassingly parallelization for generating the submodels, by storing all required mappings associated to the unstructured grids for the continents and islands, boundary conditions, areas, connectivities, and data tiling. By doing this, we do not have to recompute the mappings for each submodel and therefore save runtime. First, these mappings are used to assign the areas for the catchment partitioning, therefore requiring runtime that is neglectable. Second, these mappings are used for fast direct-access read of (tiled parameter) data required for assembling the submodels in parallel. The final step in the Model Workflow is to run the models on a distributed memory parallel computer (process "Run Model" in Figure 3a).

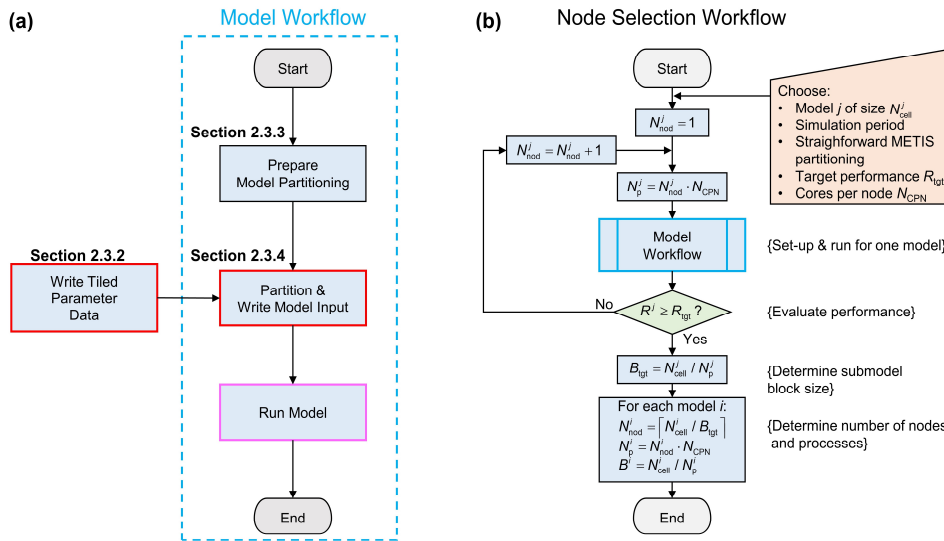



Figure 3. (a): Main Workflow for processing the GLOBGM. A red box and pink box denote embarrassingly parallel and distributed memory parallel, respectively, see Section 2.1. (b): Node Selection workflow. Workflow symbols used: process (blue), decision (green), manual input (orange).

It should be noted that in our workflow we have deliberately left out the post-processing used in this study. The reason for this is that we only perform limited post-processing and mainly focus on measuring the performance of model simulation. In our post-processing, runtimes and data storage for transient evaluation are low since we use direct-access reading of binary output and only generate time series for a limited number of selected well locations. However, in real-life application of the GLOBGM, more output data may be needing to be processed, resulting in non-neglectable runtimes. In that case, post-processing could benefit from parallelization.

In Figure 3b the Node Selection workflow is given, that is applied for estimating the number of nodes N_{nod}^i to be used for each model i. This estimation is done by conducting a strong scaling experiment, meaning that the number of cores is being varied for a model having a fixed problem size. Input for this workflow is a selected model $1 \le j \le N_{\mathrm{m}}$, having a convenient grid size (N_{cell}^j) and simulation period. Straightforward METIS partitioning is applied, see Section 2.1, as well as a straightforward iteration scheme. Starting from $N_{\mathrm{nod}}^j = 1$, in each iteration of this workflow, the number of cores (or MPI processes) N_{p}^j is chosen to be a multitude of the number of cores per node N_{CPN}, hence $N_{\mathrm{p}}^j = N_{\mathrm{nod}}^j \cdot N_{\mathrm{CPN}}$. Then, the Model Workflow generates the model input files for this number of cores, followed by running the model to obtain runtime performance R^j. The iteration finishes when $R^j \ge R_{\mathrm{tgt}}$, and the target submodel size (number of grid cells) is determined by $B_{\mathrm{tgt}} = N_{\mathrm{cell}}^j / N_{\mathrm{p}}^j$. Using this submodel size, for each model $i = 1,\ldots,N_{\mathrm{m}}$, the number of nodes is given by $N_{\mathrm{nod}}^i = \left\lceil N_{\mathrm{cell}}^i / B_{\mathrm{tgt}} \right\rceil$, where $\lceil \cdot \rceil$ denotes the ceiling function. Then, the number of cores to be used for each model $i = 1,\ldots,N_{\mathrm{m}}$ follows

straightforwardly from $N_p^i = N_{nod}^i \cdot N_{CPN}$. The (maximum) total number of nodes being used for the GLOBGM we denote by

$$N_{nod} = \sum_i N_{nod}^i .$$

2.3.2 Model Workflow: Write Tiled Parameter Data

In the offline coupling of PCR-GLOBWB to MODFLOW (see Section 2.2.1), all model parameter data required for the MODFLOW models are written prior to simulation. For this, the PCR-GLOBWB-MODFLOW Python scripts that use PCRaster Python modules (Karssenberg et al., 2010) are slightly modified for the MODFLOW module to process and write 30″ PCRaster raster files (uncompressed, 4-bytes single-precision). The processing is done for a total of 163 squared raster tiles of 15° having 30″ resolution (1800×1800 cells), enclosing the global computational grid, see Figure 4. Using tiles has benefits for two reasons. First, using tiles cancels a significant number of redundant sea cells (missing values), reducing the data storage for storing one global map from 3.47 GB to 1.97 GB in our case (43% reduction). Second, using tiles allows for (embarrassingly) parallel pre-processing to reduce runtimes. In our pre-processing, the 163 tiles are distributed proportionately over the available N_{nod} nodes. It should be noted that, although the tiling approach is quite effective, still many coastal tiles have redundant sea cells. Since missing values do not require pre-processing, one might argue that these tiles would result in a parallel workflow imbalance. However, the version 4.3.2 of PCRaster we used does not treat missing values differently, and therefore choosing tiles of equal sizes seems to be appropriate. It should also be mentioned that choosing 15° tiles is arbitrary and different tiles could be chosen.

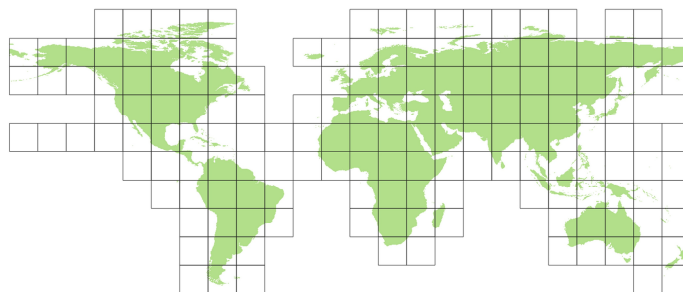



Figure 4. Parameter data tiles having 30″ resolution and 15° size (1800 × 1800 cells). Total: 163.

2.3.3 Model Workflow: Prepare Model Partitioning

The workflow for preparing the partitioning is given by Figure 5 in more detail. This workflow derives and writes all necessary mappings that are being used for the process "Partitioning & Write Model Input", see Section 2.3.4. It includes the two options for generating areas, see Section 2.1: METIS areas and catchment areas, see processes "A" and "B" in Figure 5, respectively.

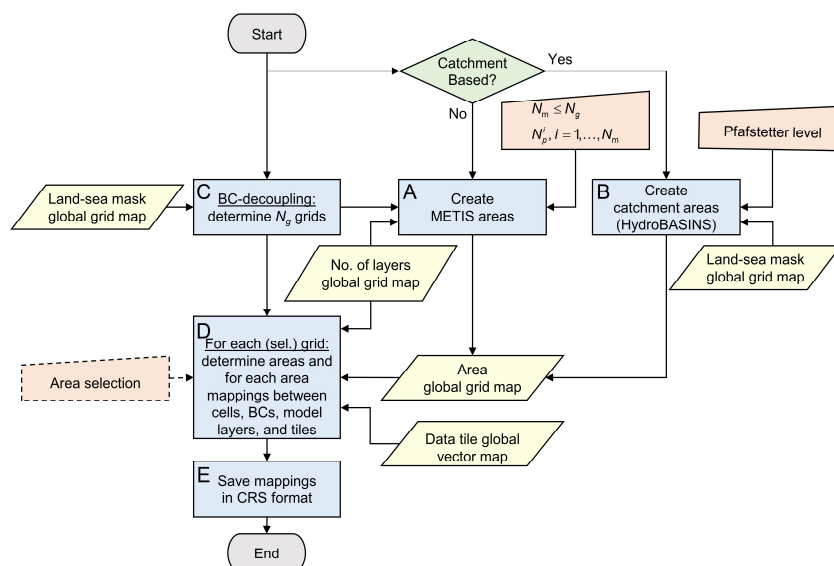



Figure 5. Process "Prepare Model Partition" in detail of the workflow in Figure 3a. Abbreviations: BC = Boundary Condition; CRS = Compressed Row Storage. Workflow symbols used: process (blue), decision (green), manual input (orange), input data (yellow).

325   When METIS areas are chosen to prepare for straightforward METIS partitioning, areas are derived from grids g_n, $n = 1, \ldots, N_g$, that are disjoint and increasing in size corresponding to continents and islands, see Section 2.1 and process "C" in Figure 5. For each (continental-scale) model i, $1 \le i < N_m$, exactly one grid g_i is taken, and for the remainder (islands) model $i = N_m$, the grids g_n, $n = N_m, \ldots, N_g$ are taken. Applying METIS partitioning for these grids to obtain N_p^i partitions is then straightforward, see Section 2.1.

330   When catchment areas are chosen to prepare for catchment partitioning, HydroBASINS catchments are determined at the global 30″ extent, see process "B" in Figure 5. This is done for a given Pfafstetter level that directly relates to the number of catchments. First, local catchment identifiers (IDs) for the HydroBASINS polygons (v1.c including lakes) are rasterized to 30″ for the corresponding eight HydroBASINS regions, numbered consecutively, and merged to the global 30″ extent. Second, extrapolation of the catchment IDs in lateral direction near the coastal zone is done using the (extended) global 30″ land-sea

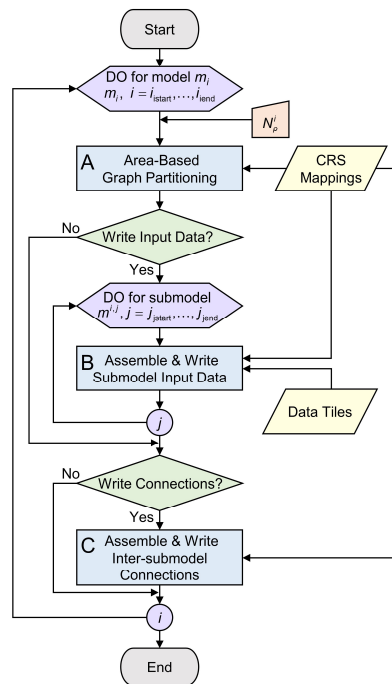
335   mask and a 9-point stencil operator. Third, the 9-point stencil operator is used to identify independent catchments and new IDs are generated and assigned where necessary.

The result of creating METIS or catchment areas is stored in a global 30″ map with unique area IDs. This map is input for process "D" in Figure 5, together with the grid definition, a (vector-based) definition of the data tiles with bounding boxes, and a global 30″ map with the number of model layers per cell. Optionally, areas can be selected by a user defined bounding

340   box, by polygon or ID, allowing users to prepare a model just for a specific area of interest. However, in this study we limit ourselves to the global extent only and therefore this option is not being used. In process "D", for each continent or island, all required mappings are determined for each of the covering areas: global index cells numbers and bounding box, neighboring areas and cell-to-cell interfaces, location of boundary conditions, the connected data tiles, and number of model layers per cell. An undirected graph is constructed using the compressed row storage (CRS) format for efficient data storage with pointers

345   (counters) to the (contiguous) bulk data with mappings. In process "E" in Figure 5, these mappings are saved to binary files that allows for fast direct access reading of all mappings required for the partitioning and writing the model input (see process "Partitioning & Write Model Input", Section 2.3.4).

2.3.4 Model Workflow: Partition & Write Model Input

The workflow for the partitioning and writing the model input files (Figure 6) consists of three processes: the "Area-Based
350 Graph Partitioning" (A), "Assemble & Write Submodel Input Data" (B), and "Assemble & Write Inter-Submodel
Connections" (C). This workflow is set up in a flexible way such that it enables embarrassingly parallel computing for a given
range of models and submodels. This means that for selected models and submodels, data can be processed independently by
using fast area-based graph partitioning and using the pre-defined and stored mappings.

355 **Figure 6.** Process "Partition & Write Model" in detail of the Model Workflow in Figure 3a. Workflow symbols used: process (blue),
decision (green), manual input (orange), input data (yellow), iterator (purple).

The first step is to perform the partitioning, see "A" in Figure 6. The goal is to determine the submodel partitions by
optimally assigning areas to balance the parallel work load using METIS while minimizing the edge-cuts. For a model m^i,
first all necessary area data (cumulative node and edge weights, connectivites) are collected from the saved mappings that are
360 a result of the "Prepare Model Partitioning" workflow, see 2.3.3. For each model m^i, $i = 1,\ldots,N_{\mathrm{m}}-1$, this is done for all areas
belonging to grid g_i; for the remainder model $m^{N_{\mathrm{m}}}$ this is done for all the areas belonging to grids $g_{N\mathrm{m}},\ldots,g_{N_g}$. After
constructing the graph for model m^i, METIS is being called to partition the graph into user-defined N_{p}^i parts.

For the example of Figure 2, the graphs are depicted in Figure 7, where the nodes correspond to catchments and the edges
to the connectivities between the catchments. Since in this example the number of disjoint grids $N_g > N_{\mathrm{m}}$, which is generally
365 the case, the associated graph for the (remainder) model m^2 is disconnected, while the graph for model m^1 is connected. The
node weights are defined as the sum of the 30″ cell weights for a corresponding catchment, and the edges are defined as the
sum of shared cell faces between the catchments, see Section 2.1. Focusing on the node weights for the example graph, let W^i
denote the model weight of model i and $w^{i,j}$ the submodel weight such that $W^i = \sum_j w^{i,j}$. In this example, model m^1 has a
total weight of $W^1 = 100$ divided into two submodels, each having a weight of $w^{1,1} = w^{1,2} = 50$. Following Karypis and Kumar
370 (1998), we here define the load imbalance for a model i as the maximum submodel weight divided by the average (target)

weight, hence $I_i = \max_j w^{i,j} / (W^i / N_p^i) = N_p^i \cdot \max_j w^{i,j} / W^i$. In this definition, the model is perfectly balanced when $I_i = 1$ and

load imbalance occurs when $I_i > 1$. Assuming Amdahl's law holds (Amdahl, 1967), it follows that the speedup (and hence

parallel performance) is proportionally to I_i^{-1}. Defining the imbalance increase as $I_i^* = 100 \cdot (I_i - 1)$ [%], then for model m^1

the submodels are perfectly balanced ($I_1^* = 0$ %). On the other hand, the catchments for model m^2 could not be perfectly

375   distributed over the two submodels since the second submodel has a larger weight, $w^{2,2} = 21$, than the first submodel,

$w^{2,1} = 19$. Therefore, the load imbalance for the second model, having a total weight $W^2 = 40$, is $I_2^* = 5$ %.

     In general, area-based partition with catchments results in insurmountable load imbalance, depending on several factors.
The amount of imbalance depends on the (multi-level) partitioning algorithm being used, the number of catchments related to
the number of partitions, the catchment geometry, and the effect this all has on the search space. In this study, we do not try to
380   make any quantitative or qualitative statements on this. Moreover, we aim for the practical aspects for a given commonly used
graph partitioner with default settings applied to a realistic set of catchments. It should be noted, as already highlighted in
Section 2.1, that load imbalance for METIS areas can be considered as optimal, since the number of nodes (equals the number
of lateral cells) is very large compared to the number of partitions, and do not vary strongly in weight. It is verified that the
obtained load imbalance never exceeds the specified maximum tolerance of $I_i = 1.0001$.

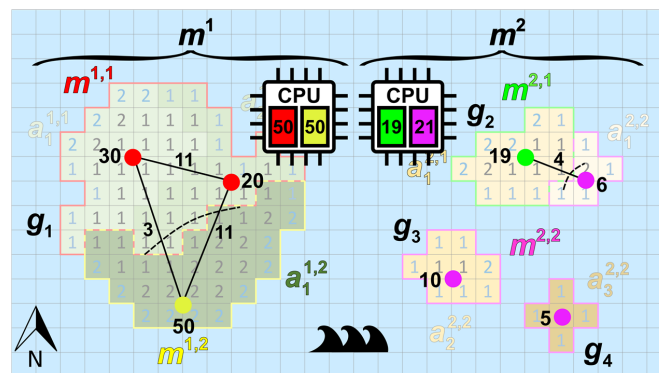



Figure 7. Example of area-based graph partitioning corresponding to the example of Figure 2, where the work for each submodel is uniquely assigned to a core of a dual-core CPU.

     The second step in the workflow of in Figure 6, see "B", is to process all submodel data. First, the resulting METIS or
390   catchment areas assigned by METIS (result of process "A") are used to assemble the unstructured grid. Then, using the tiled
parameter data and the CRS mappings, both steady-state as well as transient model input data are written. To significantly
reduce the number of model input files for transient simulation, each submodel has exactly one binary file for storing all
necessary bulk data. This is a new functionality of our MODFLOW 6 prototype. For the third and last step in the workflow of
Figure 6, process "C", first an inter-submodel graph is assembled for each model by the merging the inter-area graph from the
395   CRS mappings and the derived partitions from process "A". Second, the submodel connections (MODFLOW exchanges) are
written to files, as well as wrappers for running the steady-state model, the transient spin-up model and the actual transient
model.

2.4 Evaluation for the contiguous United States

     Here, the GLOBGM is evaluated for the CONUS for which many head measurement data sets are publicly available. For
400   our study, we restrict to selected groundwater head observations from the USGS National Water Information System (NWIS)
database (USGS, 2021). This evaluation is limited, however, since the GLOBGM v1.0 is an initial model. This means various

other aspects are left for further research: improving model schematization (e.g. geology), improving model parameters by adding more (regional scale) data and calibration, and adding more global dataset for comparison. Our evaluation is mainly on comparison to the 5′ PCR-GLOBWB MODFLOW model, from here briefly referred to as GGM, for which we used consistent
405   upscaled data.

2.4.1   Steady-state groundwater heads

The steady-state (i.e. long-term average) evaluation is limited to the so-called natural condition, meaning that human intervention by groundwater pumping is excluded from the model. Besides comparing the GLOBGM to the GGM, we compare the computed steady-state groundwater heads to two other models: the global-scale inverse model (GIM) from Fan et al.,
410   (2017), having 30″ resolution and used for estimating steady-state root water-uptake depth based on observed productivity and atmosphere by inverse modeling, and the CONUS model from Zell and Sanford (2020), a MODFLOW 6 model developed by the USGS for simulating the steady-state surficial groundwater system at the CONUS-extent. From here, these latter models are referred to as GIM and CGM, respectively. Here, for the steady-state evaluation, we use the same NWIS wells as selected by Zell and Sanford (2020). Mean groundwater head residuals are computed for HUC4 surface water boundaries from the
415   USGS Watershed Boundary Dataset (U.S. Geological Survey, 2021b) to get a spatially weighted distribution of the residuals for the CONUS.

2.4.2   Transient groundwater heads for 1958-2015

A comparison is made between the GLOBGM and GGM model including groundwater pumping (non-natural condition), for the simulation period of 1958-2015. Considering monthly averaged groundwater heads, the sample correlation coefficient
420   r_{mo} is used for quantifying the timing error between model ("m") and observation ("o"). Furthermore, the absolute and relative interquartile range error, $\mathrm{IQRE_{mo}}=|\mathrm{IQR_m}\text{-}\mathrm{IQR_o}|/\mathrm{IQR_o}$, with $\mathrm{IQR_m}$ and $\mathrm{IQR_o}$ the interquartile ranges for the model and observations, respectively, is used to quantify the amplitude error (de Graaf et al., 2017; Sutanudjaja et al., 2011). Additionally, the trend of the (monthly averaged) time series is computed, considering the slope β_{y} of yearly averaged groundwater heads from a simple linear regression with time. We assume that a time series has a trend for $|\beta_{\mathrm{y}}| > 0.05$ m per year, and no trend
425   otherwise.

In the transient evaluation, long-term averaged statistics including seasonality of groundwater heads are considered for unconsolidated sedimentary systems only. Starting from ~900k well locations, time series are selected having at least one groundwater head observation for each quarter during five consecutive years. Statistics is only computed for such contiguous periods. Model layer selection for a given NWIS well is done as follows. When there is only one upper or lower model layer
430   present for this site, then we simply select the corresponding model layer. When there is both an upper and lower layer present, we first try to match the NWIS well depth below land surface. In case this depth is smaller than the upper model layer thickness, we select the upper layer; otherwise we select the lower layer. When there is no well depth available for a site, we try to correlate the NWIS head time series with a corresponding soil moisture time series from the European Space Agency Climate Change Initiative Soil Moisture dataset, derived from the combined active and passive satellite sensors (v05.2; Gruber et al.,
435   2019, 2017; Dorigo et al., 2017). If there is a (lagged) correlation greater than 0.5, we take the upper layer; if not, we take the lower layer. If there is no soil moisture data available, we do not use the NWIS site for the evaluation. In total, the filtering resulted in 12,342 site locations used for the transient evaluation. Results for timing, amplitude and trend for these wells are aggregated to the same HUC4 surface water boundaries as used for the steady-state evaluation, see Section 2.4.1. However, we declare HUC4s having lesser than five NWIS wells not spatially representative and exclude them from the presented
440   statistics.

3   Results and discussion

3.1 Pre-processing for 1958-2015

3.1.1 Node selection

Figure 8a shows the performance for a HPCG strong scaling experiment (see Section 2.1) on a single thin node of the
445 Dutch national supercomputer Snellius (see Table 3), up to a maximum of 128 cores using two CPUs. In this figure, the ideal
performance (dashed line) is a straightforward extrapolation of the serial performance. It can be observed that a flattening
occurs starting from 32 cores, where the competition of cores for the memory bandwidth results in saturation. From this, the
maximum number of cores per node is chosen as $N_{CPN} = 32$ for the remainder of the experiments. For the node selection
procedure (see Section 2.1, Figure 3b), the Americas model is chosen ($j = 2$) considering a one-year simulation for 1958.
450 This model corresponds to a "medium-sized" model consisting of $N_{cell}^2 = 77$ million cells. Starting with $N_{nod}^2 = 1$ node, hence
using a total of $N_p^2 = 1 \cdot 32 = 32$ processor cores, Figure 8b shows that for the third iterate the measured performance exceeds
the target performance, $R^2 = 145 > 117 = R_{tgt}$, hence $N_{nod}^2 = 3$ nodes for a total of $N_p^2 = 3 \cdot 32 = 96$ cores. Therefore, the target
submodel size is $B_{tgt} = N_{cell}^2 / N_p^2 = 77 / 96 = 0.8$ million 30″ cells. Then, by computing $N_{nod}^i = \lceil N_{cell}^i / B_{tgt} \rceil$ for $i = 1, ..., 4$, and
calculating back the number of nodes and cores, the Afro-Eurasia model is estimated to use seven nodes and 224 cores to meet
455 the target performance, the Americas model three nodes and 96 cores, the Australia and Islands model each uses a single node
and 32 cores, see Table 4. Hence, the total maximum number of nodes in the study for the GLOBGM is $N_{nod} = 12$ for a total
of 384 cores. This results in an average submodel size of 0.72 million 30″ cells.

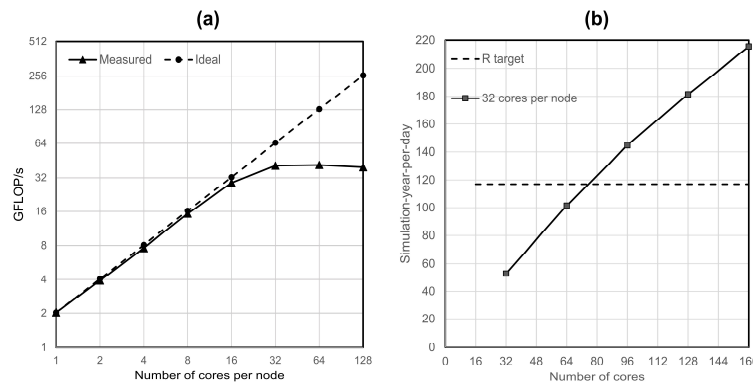



Figure 8. (a): HPCG performance results in GFLOPS/s for a model having 104^3 cells and a maximum runtime of 60 s, (b): Performance
460   estimation for the Americas model considering a transient simulation for 1958.

Table 4. Configuration of nodes and cores resulting from the node selection procedure.

Model	i	Number of Cells (M)	Submodel Size (M)	Number of Nodes	Number of Cores
Afro-Eurasia	1	167.51	0.75	7	224
Americas	2	77.13	0.80	3	96
Australia	3	16.34	0.51	1	32
Islands	4	17.35	0.54	1	32
		278.33	0.72	12	384

3.1.2 Tiled parameters and model input

The parameter pre-processing for the 163 data tiles (see Section 2.3.2) is distributed over the 12 available nodes, such that
465 the first 11 nodes do the pre-processing for 14 tiles, and the last node for 9 tiles. For each tile, the average runtime for pre-processing 1958-2015 (696 stress-periods) is 3 hours 25 minutes. In serial, this would require 558.7 core hours in total or ~23 days of runtime accordingly. This results in 5,578 PCRaster files ($10 + 696 \cdot 8 = 5,578$; see Table 1) for each tile, requiring 68 GB of storage. Hence, in total 909,214 files are written in parallel, amounting to 10.8 TB storage. Using data tiles therefore saves storing ~8 TB of redundant data (43% reduction; see Section 2.3.2).

470 Table 5 shows the pre-processing runtimes for each submodel to generate the transient MODFLOW input data in parallel using the node configuration as in Table 4, considering straightforward METIS partitioning, see Section 2.1. On average, it takes up to half an hour to do the pre-processing for each Afro-Eurasia, Americas and Australia model. As can be observed, clearly not all submodels require the same runtime and there is a spread in distribution. Looking at the standard deviation (STD in Table 5), this is significantly largest for the Islands model, measuring about 2.5 hours. We believe this is inherent to our
475 chosen partitioning for the Islands model, allowing submodels to have grids cells of many scattered islands that are scattered across the world, see e.g. Figure 9 for the slowest submodel. By this, it is likely that random data-access of scattered unstructured grid cells to the tiled parameter grids occurs, slowing down the data reading after the submodel assembly. Although pre-processing is not a focus in this study, we might improve this in the future by incorporating a clustering constraint in the partition strategy. For the Afro-Eurasia, Americas and Australia models, the STD is comparably small varying from 3
480 to 9 minutes. For the total parallel pre-processing to generate model input, a total of 222 core hours was required or ~9 days of serial runtime accordingly.

Table 5. Pre-processing runtimes for process "Assemble & Write Submodel Input Data", see "B" in Figure 6. STD = standard deviation.

Model	Average	Min.	Max.	STD	Core hours
Afro-Eurasia	00:27:13	00:20:44	00:46:50	00:05:04	101.6
Americas	00:28:42	00:22:41	01:40:47	00:09:06	45.9
Australia	00:20:08	00:16:54	00:27:10	00:03:20	10.7
Islands	01:59:52	00:26:48	15:00:46	02:25:40	63.9
					222.2

Figure 9. Location of unstructured grid cells (red) for a submodel of the Islands model, showing scattering, and requiring large pre-processing run-time for writing the submodel input data.

3.2 Parallel performance for 1958-2015

485   Figure 10 shows the main parallel performance results for the GLOBGM, considering simulation of 1958-2015 including a 20-year spin-up using the node/core configuration as in Table 4, and applying straightforward METIS partitioning (see Section 2.1). This figure shows that the target performance of 117 SYPD (16 hours of runtime) was achieved for all models (green bars), showing a significant increase in performance compared to the serial case (red bars). For the largest Afro-Eurasia model, the serial runtime could be reduced significantly from ~87 days (0.9 SYPD) to 0.63 days (or ~15 hours; 123.6 SYPD)

490   using 224 cores. This corresponds to a speedup factor is 138.3 with a parallel efficiency of 62%. In general, for all four models, the parallel efficiency is ~60%.

It should be noted that the serial performance was only evaluated for a single year, i.e. 1958. The reason for this is that serial runtime for the entire simulation period was too long (e.g. ~27 CHPSY for Afro-Eurasia, hence $27 \cdot 78 = 2,106$ hours or 87 days runtime) and it would exceed the maximum allowed runtime of 5 days on a single Snellius node. For each parallel

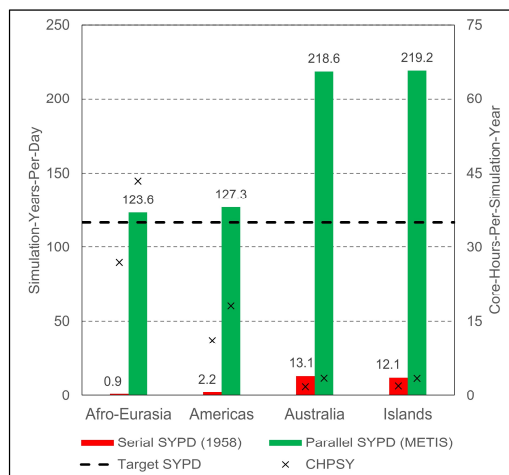
495   run, however, the performance is evaluated for 78 years (20 times 1958 + 1958-2015). Furthermore, in our experiments each serial/parallel run is only evaluated twice, taking average performance values for two runs only. We therefore did not account for any statistic (hardware related) runtime variation on the Snellius supercomputer. The reason for this was the limited total number of available core hours, where one full GLOBGM run requires ~24,000 core hours.

500

Figure 10. Parallel performance results for the models of the GLOBGM, considering straightforward METIS partitioning. Left vertical axis: SYPD (bars); right vertical axis: core-hours-per-simulated-year CHPSY (crosses). Serial performance is computed for 1958 only.

Figure 10 shows that there is a performance difference between the models. The slowest model is the Afro-Eurasia model, followed by the Americas model, the Australia model and Islands model. This holds for both the serial and for the parallel case. For the serial performance it obvious that het main difference in performance is caused by the difference in model sizes. The serial performance difference between the largest Afro-Eurasia model having 168 M cells and the smallest Australia model having 16 M cells, that is 10 times smaller, can be directly related to lesser FLOPS and I/O.

However, also the difference in total number of iterations (linear + non-linear) contributes to this: considering 1958, the serial Afro-Eurasia model required 1.47 times more iterations to converge than the Australia model. The difference in iterations is likely related to the problem size and the number of the Dirichlet boundary conditions effecting the matrix stiffness. For the parallel case, this effect is enhanced, since the parallel linear solver has different convergence behavior and generally requires more iterations to converge when using more submodels, inherent to applying the additive Schwarz preconditioner. In general, for the parallel performance considering METIS partitioning, the increase in the total number of iterations is in the range of 34% to 58%. That directly relates to a significant performance loss. Although we found that the parallel performance is adequate to reach our performance goal, reducing the total number of iteration is therefore something to consider for further research. We might improve the number of iterations by tuning the solver settings or applying a more sophisticated paralleled preconditioner. However, in general, users do not spend time on tuning solvers settings and we therefore take them as they are. Furthermore, as a first attempt to reduce iterations, we did not see any improvement by using the additive coarse grid correction preconditioner.

Moreover, the difference in parallel performance could be explained by the differences in submodel (block) sizes (see Table 4). Due to rounding the number of cores to the number of cores per node, the block sizes for the Australia and Islands models are ~1.5 smaller than the block sizes for the Afro-Eurasia and Americas models, which directly results in an increasing performance.

Besides the increase of in iterations, memory contention contributes to the loss of parallel performance (see Section 2.1), even when using 32 cores per node out of 128 cores. From the HPCG test, see Figure 8a, the parallel efficiency using 32 cores per node is 63%, which is likely to be a representative value for the MODFLOW linear solver. For MODFLOW, however, this value is likely to be slightly larger because of non-memory bandwidth dependent components. Comparing to a 1958 run for the Americas model, and using one core per node, we estimate the maximum efficiency to be 77%. Extrapolating this to the Afro-Eurasia model, this means that ~172 cores could be used efficiently, increasing the efficiency to 80%. This value is more in range with we expect from preceding research using MODFLOW (Verkaik et al., 2021b).

In Figure 11a, the performance results are given for catchment partitioning considering HydroBASINS Pfafstetter levels 8, 6 and 5 (see Section 2.1) chosen for illustration. Level 7 was excluded deliberately in the search for finding a significant performance decrease and minimizing the allocated budgets on the Snellius supercomputer. In general, for the Australia and Islands model, the target performance is exceeded for all HydroBASINS levels. For the Afro-Eurasia and Americas models, level 8 is sufficient to reach the target and level 6 results in performance slightly below the target. For these models, level 5 results in about three quarters of the target performance. In general, using HydroBASINS catchments up to Pfafstetter level 6 seems to give adequate results. Except for the Australia model, performance decreases when the Pfafstetter level decreases. The reason why this doesn't apply to the Australia model is because of the slight iteration increase for level 8. In Figure 11b, the performance for catchment partitioning (Figure 11a) is normalized with the performance using straightforward METIS partitioning (Figure 10) and the associated total iteration count. Clearly, the performance slope is correlated to the inverse of the load imbalance determined by METIS, see Section 2.3.4. Since many factors may contribute to load imbalance, like the specific multilevel heuristics being used, specified solver settings, characteristics of the graph subject to partitioning, limitations of the specific software library, an in-depth analysis is beyond the scope of this study. Here, we simply assume that

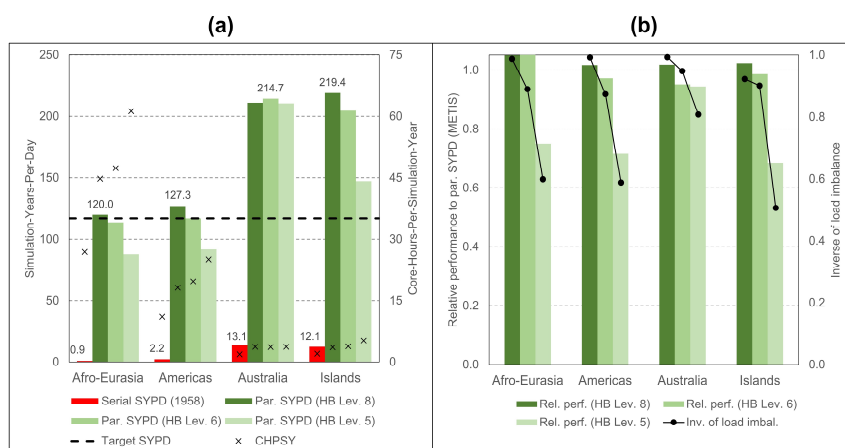
this imbalance is a direct result of the coarsening associated to decreasing Pfafstetter levels for a fixed number of submodels, reducing the METIS graph-partitioning search space.

550

Figure 11. (a): Performance results for the GLOBGM, considering catchment partitioning using HydroBASINS catchment areas for Pfafstetter level 8, 6 and 5. Left vertical axis: SYPD (bars); right vertical axis: CHPSY (crosses). Serial performance is computed for 1958 only. (b): Relative performance to performance using straightforward METIS partitioning, normalized with the corresponding total number of iterations (bars), and inverse of the METIS load imbalance I_i (lines).

555

The required storage for each run is given by Table 5, where monthly computed groundwater heads were saved exclusively during simulation. For this, one transient global run required 8.8 TB of input and 1.4 TB of output.

Table 5. Input and output of the GLOBGM for simulating 1958-2015.

Model	Input (TB)	Output (TB)
Afro-Eurasia	5.27	0.85
Americas	2.53	0.39
Australia	0.44	0.08
Islands	0.56	0.09
	8.80	1.41

3.3 Evaluation for the contiguous United States

3.3.1   Steady-state groundwater heads

Figure 12 shows the steady-state water table depth residuals aggregated to the HUC4 surface watershed boundaries at the CONUS-extent, comparing the GLOBGM to the GGM, the GIM (Fan et al., 2017) and the CGM (Zell and Sanford, 2020), see Section 2.4.2. Clearly, the GLOBGM is an improvement compared to the GGM, where the frequency distribution of the GGM residuals shifts towards the residuals of the GLOBGM. Hence, refining from 5′ to 30″ resolution is resulting in higher and more accurate groundwater heads. Especially for the western CONUS this seems to have a large effect, where the precipitation regime that can be considered as arid. However, in general the computed groundwater heads still seem rather low. Analyzing the possible causes for this should is left for further research.

(a)

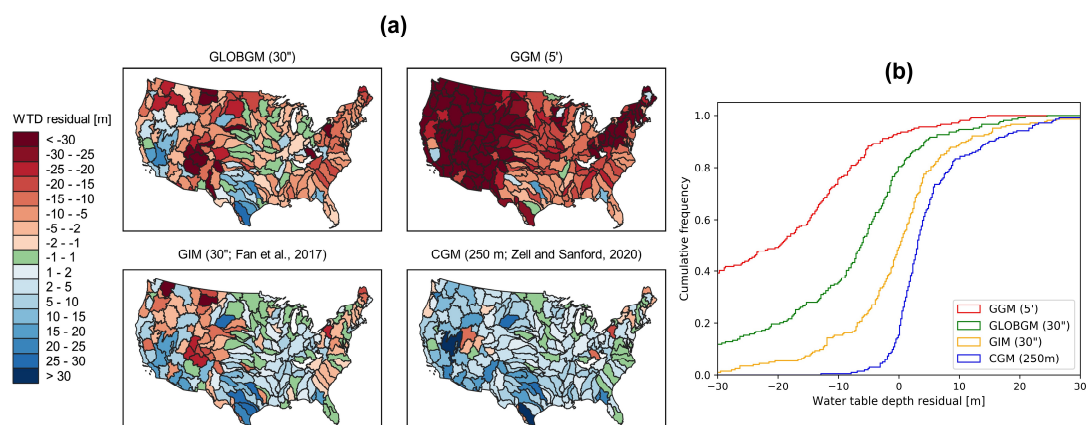



Figure 12. (a): Water table depth (WTD) residuals aggregated to HUC4 units for the steady-state GLOBGM compared to the GGM, the GIM model of Fan et al. (2017) and the CGM of Zell and Sanford (2020). (b): Corresponding cumulative frequencies.

3.3.2   Transient groundwater heads for 1958-2015

Figure 13 shows the results for the transient evaluation for 1958-2015, see Section 2.4.2, comparing the GLOBGM to the GGM for average timing (r_{mo}), average amplitude error ($IQRE_{mo}$), and trend classification (using β_y). In general, we see that both models give very comparable results that could be further improved. For the average amplitude error, the GLOBGM seems to perform slightly worse, for which we do not have a straightforward explanation. However, the GLOBGM seems slightly better regarding trend direction and lacking model trend. Furthermore, it can been that about 40-50% of the (majority of) obervations have a mismatch in trend. This can likely be related to incorrect well locations and pumping rates in the model,

580 inherent to the applied parametrization and concepts within PCR-GLOBWB (see Table 1). This might also have effect on the

discrepancies for the timing and amplitude errors and is therefore subject for further research.

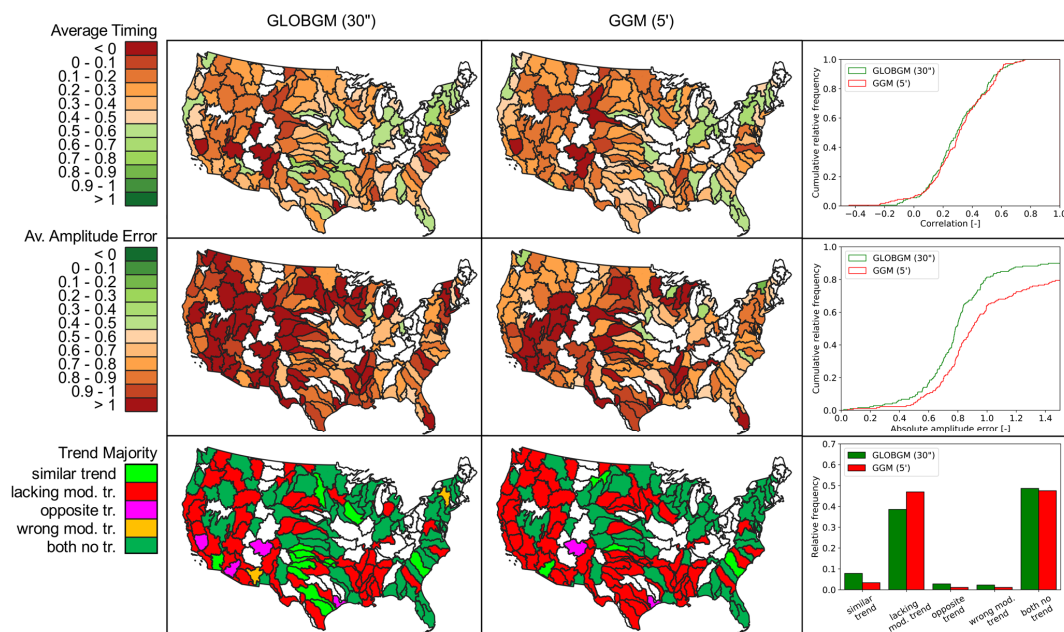



Figure 13. Average timing (first row panels), amplitude error (second row panels) and trend (third row panels) for the GLOBGM compared to the GGM, considering a transient run for 1958-2015. Plotted white colors for HUC4 units mean that less than five samples were found
585 and hence was not considered representative.

3.4 Example of global-scale results

To provide an impression about the level of detail reached when simulating groundwater heads at 30″ spatial resolution, we show the steady-state solution of water table depths in Figure 14a. The panels with blowups show the intricate details present, which are mostly guided by surface elevation and the presence of rivers. Figure 14b shows the groundwater head
590 amplitude for unconsolidated sedimentary systems as represented by the interquartile range, where the amplitude size reflects the amplitude of groundwater recharge and the hydrogeological parameters (storage coefficient and hydraulic conductivity). Finally, in Figure 14c we present the trends of yearly average groundwater heads for unconsolidated sedimentary systems over the period 1958-2015. For the areas with confining layers, the trend in the heads of the lower model cells is taken. Here, the well-known areas of groundwater depletion (Wada et al., 2010; De Graaf et al., 2017) are apparent. However, also areas with
595 spurious trends can be seen that may be connected to incomplete model spin-up. Furthermore, positive trends can be seen, which may be connected to increased precipitation related to climate change. In the supplement to this paper we provide two animations for computed water table depths: one for monthly values and another one for yearly averaged values. To show the dynamic behavior of the groundwater heads, we show these results relative to the heads of 1958.

600

605

610

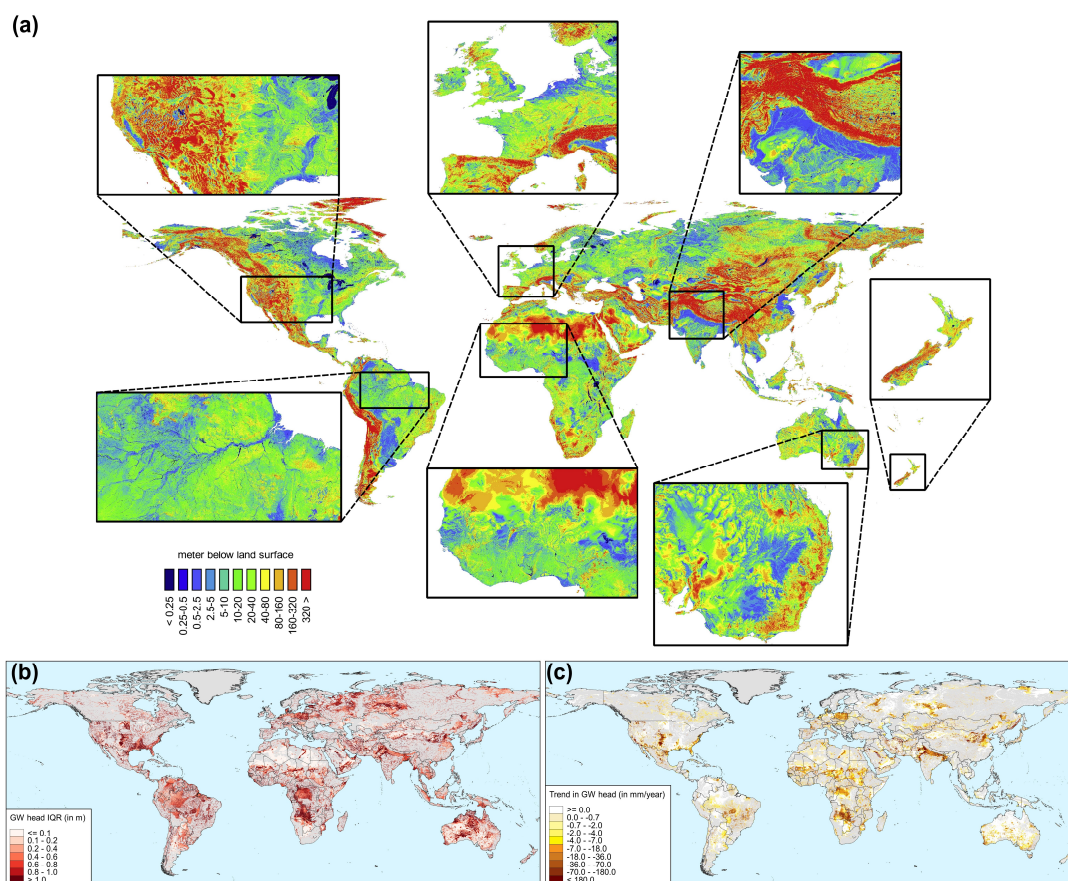



Figure 14. Example outputs of the GLOBGM. (a): Steady-state water table depth with detailed blow ups; (b): Water table amplitude as represented by the interquartile range IQR. (c): Groundwater head trends for the aquifer (lower model layer).

4   Conclusions and recommendations

615   The PCRGLOB-MODFLOW global-scale groundwater model at 30″ spatial resolution (GLOBGM v1.0) was successfully implemented using high performance computing for simulating long transient periods. To our knowledge, this is the first implementation of a transient global-scale groundwater model at this resolution. Our implementation uses unstructured grids to cancel redundant sea and land cells, and effectively applies a static asynchronous parallelization to organize the global model as a set of independent parallel models, resulting in three continental-scale models (Afro-Eurasia, Americas, and Australia)

620   and one remainder model for all (smaller) islands. We showed that our workflow, using parallel pre-processing and a new parallel distributed memory prototype version of MODFLOW 6, is effective for achieving a user-defined parallel runtime target and to minimize data usage. This is demonstrated for an experiment on the Dutch national supercomputer Snellius, simulating the GLOBGM for 1958-2015, considering both METIS partitioning and catchment partitioning.

With our approach, we first estimated the required node/core configuration on the Snellius supercomputer to achieve a set

625   target of 16 hours runtime including the 20-year spin-up (leading to 117 Simulation-Years-Per-Day; SYPD), and then we

conducted the parallel pre-processing and illustrated its necessity. For METIS partitioning, we showed that a maximum of 12 nodes running with 32 cores per node is required to meet the target for each of the four underlying models of the GLOBGM. For the largest Afro-Eurasia model using seven nodes (224 cores), runtime was reduced from ~87 days to 15 hours (124 SYPD). For catchment partitioning, the first results presented here are promising. Using HydroBASINS catchments as an example, we showed that this lesser optimal partitioning results in quite similar parallel performance down to Pfafstetter level 6. Since our implementation is suitable for parallel systems with relatively limited hardware requirements, we believe it is well suitable for average users that do not have exclusive access to many nodes and need to deal with queuing times. We therefore believe that our implementation will contribute to future model improvements.

From the model evaluations for the CONUS-extent using NWIS observations, we conclude that the steady-state groundwater heads from the GLOBGM are significantly better compared than those from the 5′ PCRGLOB-MODFLOW model, but still could be improved compared to measurements and model results from Fan et al., (2017) and Zell and Sanford (2020). For the transient simulation, results for the GLOBGM and the 5′ model are comparable, both giving significant differences compared to measurements. Although the exact reasons for the differences are kept for further research, along with further model improvement, they are likely a result of spatial difference in resolution, lacking transient model input data, e.g. for groundwater well abstraction, or must be found in improving the hydrogeological schematization

Although the current parallel performance is quite satisfactory for its purpose, it could be further improved, e.g. by improving the processor core utilization, reducing the number of model iterations, and reducing the pre-processing times. First, multicore CPUs are likely to have more and more memory channels to be more applicable to memory-bound problems and we expect better core utilization with next generation processors. Second, model iterations could be reduced likely by tuning the MODFLOW solver settings or improving the parallel preconditioners. Third, parallel pre-processing runtimes could possibly be reduced by improving the partitioning, e.g. by clustering many smaller islands causing random access data patterns for the Islands model.

Regarding storage, users should be aware that GLOBGM requires more than 21 TB (900k files) of data for a single run. Since we now exclusively use uncompressed PCR-Raster files, requiring a large amount of storage, compression could be considered for follow-up research, as well as using more data tiles.

Applying catchment partitioning gives opportunities for further research, e.g. to realize a parallel coupling to the PCR-GLOBWB surface water routing module with acceptable parallel performance for the groundwater model, or to apply such grid-independent partitioning to even higher resolutions.

Code and data availability. The GLOBGM v1.0 is open source and distributed under the terms of GNU General Public License v3.0, or any later version, as published by the Free Software Foundation. The model tools and data are provided through the GitHub repository https://github.com/UU-Hydro/GLOBGM.git (Verkaik and Sutanudjaja, 2022, https://doi.org/10.5281/zenodo.7398200), where the main model data can be accessed through https://doi.org/10.24416/UU01-44L775. The parallel kernel of MODFLOW 6 used for the GLOBGM v1.0 is provided through https://github.com/verkaik/modflow6-parallel.git (Verkaik et al., 2021c, https://doi.org/10.5281/zenodo.5778658). This kernel is open source through the CC0 1.0 Universal public domain dedication. Although development and maintenance of the official version of the GLOBGM is conducted at the Department of Physical Geography, Utrecht University, we welcome and encourage researchers from external parties to contribute.

Author contributions. JV performed the conceptualization, methodology, and implementation for all workflows, as well as the parallelization of MODFLOW 6. EHS helped with the conceptualization of this research and prepared all of the GLOBGM raster data for the transient experiment. JV performed all other pre-and post-processing, simulation on the Dutch national

supercomputer Snellius, and analysis of all results. MFPB, HXL and GHPOE supervised this research and helped with its
670   conceptualization. JV prepared the manuscript with contributions from all authors.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. We thank Joseph D. Hughes and Christian D. Langevin for their valuable suggestions and comments on
675   parallelizing MODFLOW. We also thank Deltares for making this research possible. This work was carried out on the Dutch
national e-infrastructure with the support of SURF Cooperative. MB acknowledges support from the European Research
Council, ERC AdG grant GEOWAT under grant agreement No: 101019185.

References

Amdahl, G. M.: Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities, in: Proceedings
680   of the April 18-20, 1967, Spring Joint Computer Conference, 483–485, https://doi.org/10.1145/1465482.1465560, 1967.

van Beek, L. P. H. and Bierkens, M. F. P.: The Global Hydrological Model PCR-GLOBWB: Conceptualization,
Parameterization and Verification, Technical report Department of Physical Geography, Utrecht, The Netherlands,
http://vanbeek.geo.uu.nl/suppinfo/vanbeekbierkens2009.pdf, 2009.

van Beek, L. P. H., Wada, Y., and Bierkens, M. F. P.: Global monthly water stress: 1. Water balance and water availability,
685   Water Resour. Res., 47, https://doi.org/10.1029/2010WR009791, 2011.

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?:   "Everywhere   and   locally   relevant,"   Hydrol.   Process.,   29,   310–320,
690   https://doi.org/10.1002/hyp.10391, 2015.

Burstedde, C., Fonseca, J. A., and Kollet, S.: Enhancing speed and scalability of the ParFlow simulation code, Comput.
Geosci., 22, 347–361, https://doi.org/10.1007/s10596-017-9696-2, 2018.

Dongarra, J., Heroux, M. A., and Luszczek, P.: A new metric for ranking high performance computing systems, Natl. Sci.
Rev., nwv084, https://doi.org/10.1007/s11069-013-0642-x, 2016.

695   Dorigo, W., Wagner, W., Albergel, C., Albrecht, F., Balsamo, G., Brocca, L., Chung, D., Ertl, M., Forkel, M., Gruber, A.,
Haas, E., Hamer, P. D., Hirschi, M., Ikonen, J., de Jeu, R., Kidd, R., Lahoz, W., Liu, Y. Y., Miralles, D., Mistelbauer, T.,
Nicolai-Shaw, N., Parinussa, R., Pratola, C., Reimer, C., van der Schalie, R., Seneviratne, S. I., Smolander, T., and
Lecomte, P.: ESA CCI Soil Moisture for improved Earth system understanding: State-of-the art and future directions,
Remote Sens. Environ., 203, 185–215, https://doi.org/https://doi.org/10.1016/j.rse.2017.07.001, 2017.

700   Fan, Y., Miguez-Macho, G., Jobbágy, E. G., Jackson, R. B., and Otero-Casal, C.: Hydrologic regulation of plant rooting depth,
Proc. Natl. Acad. Sci. U. S. A., 114, 10572–10577, https://doi.org/10.1073/pnas.1712381114, 2017.

Gleeson, T., Moosdorf, N., Hartmann, J., and van Beek, L. P. H.: A glimpse beneath earth's surface: GLobal HYdrogeology
MaPS   (GLHYMPS)   of   permeability   and   porosity,   Geophys.   Res.   Lett.,   41,   3891–3898,
https://doi.org/https://doi.org/10.1002/2014GL059856, 2014.

705   Gleeson, T., Wagener, T., Döll, P., Zipper, S. C., West, C., Wada, Y., Taylor, R., Scanlon, B., Rosolem, R., Rahman, S.,
Oshinlaja, N., Maxwell, R., Lo, M.-H., Kim, H., Hill, M., Hartmann, A., Fogg, G., Famiglietti, J. S., Ducharne, A., de
Graaf, I., Cuthbert, M., Condon, L., Bresciani, E., and Bierkens, M. F. P.: GMD perspective: The quest to improve the
evaluation of groundwater representation in continental- to global-scale models, Geosci. Model Dev., 14, 7545–7571,
https://doi.org/10.5194/gmd-14-7545-2021, 2021.

710   de Graaf, I. E. M., Sutanudjaja, E. H., Van Beek, L. P. H., and Bierkens, M. F. P.: A high-resolution global-scale groundwater
          model, Hydrol. Earth Syst. Sci., 19, 823–837, https://doi.org/10.5194/hess-19-823-2015, 2015.

de Graaf, I. E. M., van Beek, R. L. P. H., Gleeson, T., Moosdorf, N., Schmitz, O., Sutanudjaja, E. H., and Bierkens, M. F. P.:
          A global-scale two-layer transient groundwater model: Development and application to groundwater depletion, Adv. Water
          Resour., 102, 53–67, https://doi.org/10.1016/j.advwatres.2017.01.011, 2017.

715   de Graaf, I. E. M., Gleeson, T., (Rens) van Beek, L. P. H., Sutanudjaja, E. H., and Bierkens, M. F. P.: Environmental flow
          limits to global groundwater pumping, Nature, 574, 90–94, https://doi.org/10.1038/s41586-019-1594-4, 2019.

Gropp, W. D., Kaushik, D. K., Keyes, D. E., and Smith, B. F.: Toward realistic performance bounds for implicit CFD codes,
          in: Proceedings of parallel CFD, 233–240, 1999.

Gruber, A., Dorigo, W. A., Crow, W., and Wagner, W.: Triple Collocation-Based Merging of Satellite Soil Moisture
720   Retrievals, IEEE Trans. Geosci. Remote Sens., 55, 6780–6792, https://doi.org/10.1109/TGRS.2017.2734070, 2017.

Gruber, A., Scanlon, T., van der Schalie, R., Wagner, W., and Dorigo, W.: Evolution of the ESA CCI Soil Moisture climate
          data records and their underlying merging methodology, Earth Syst. Sci. Data, 11, 717–739, https://doi.org/10.5194/essd-
          11-717-2019, 2019.

Harbaugh, A. W.: MODFLOW-2005 , The U . S . Geological Survey Modular Ground-Water Model — the Ground-Water
725   Flow Process, U.S. Geol. Surv. Tech. Methods, 253, https://doi.org/U.S. Geological Survey Techniques and Methods 6-
          A16, 2005.

Hartmann, J. and Moosdorf, N.: The new global lithological map database GLiM: A representation of rock properties at the
          Earth surface, Geochemistry, Geophys. Geosystems, 13, 1–37, https://doi.org/10.1029/2012GC004370, 2012.

Hendrickson, B. and Leland, R.: The Chaco User's Guide Version 2.0, 1995.

730   Hengl, T., de Jesus, J. M., MacMillan, R. A., Batjes, N. H., Heuvelink, G. B. M., Ribeiro, E., Samuel-Rosa, A., Kempen, B.,
          Leenaars, J. G. B., Walsh, M. G., and Gonzalez, M. R.: SoilGrids1km — Global Soil Information Based on Automated
          Mapping, PLoS One, 9, 1–17, https://doi.org/10.1371/journal.pone.0105992, 2014.

Herlihy, M., Shavit, N., Luchangco, V., and Spear, M.: The art of multiprocessor programming, Newnes, 2020.

Hughes, J. D., Langevin, C. D., and Banta, E. R.: Documentation for the MODFLOW 6 framework, Techniques and Methods,
735   Reston, VA, https://doi.org/10.3133/tm6A57, 2017.

Huscroft, J., Gleeson, T., Hartmann, J., and Börker, J.: Compiling and Mapping Global Permeability of the Unconsolidated
          and Consolidated Earth: GLobal HYdrogeology MaPS 2.0 (GLHYMPS 2.0), Geophys. Res. Lett., 45, 1897–1904,
          https://doi.org/10.1002/2017GL075860, 2018.

Karssenberg, D., Schmitz, O., Salamon, P., de Jong, K., and Bierkens, M. F. P.: A software framework for construction of
740   process-based stochastic spatio-temporal models and data assimilation, Environ. Model. Softw., 25, 489–502,
          https://doi.org/https://doi.org/10.1016/j.envsoft.2009.10.004, 2010.

Karypis, G. and Kumar, V.: METIS: A Software Package for Partitioning Unstructured Graphs , Partitioning Meshes , and
          Computing Fill-Reducing Orderings of Sparse Matrices, 1–44, 1998.

Langevin, C. D., Hughes, J. D., Banta, E. R., Provost, A. M., Niswonger, R. G., and Panday, S.: MODFLOW 6 Modular
745   Hydrologic Model: U.S. Geological Software, https://doi.org/10.5066/F76Q1VQV, 2017.

Langevin, C. D., Hughes, J. D., Banta, E. R., Provost, A. M., Niswonger, R. G., and Panday, S.: MODFLOW 6 Modular
          Hydrologic Model version 6.2.1: U.S. Geological Software Release, U.S. Geol. Surv., https://doi.org/10.5066/F76Q1VQV,
          2021.

Lehner, B.: HydroBASINS, Global watershed boundaries and sub-basin delineations derived from HydroSHEDS data at 15
750   arcsecond resolution, technical documentation version 1.c, 2014.

Lehner, B., Verdin, K., and Jarvis, A.: New Global Hydrography Derived From Spaceborne Elevation Data, Eos, Trans. Am.
          Geophys. Union, 89, 93–94, https://doi.org/https://doi.org/10.1029/2008EO100001, 2008.

van de Leur Kraijenhoff, D. A.: A study of non-steady groundwater flow with special reference to a reservoir coefficient, Ing., 70, B87--B94, 1958.

755   Manning, R.: On the flow of water in open channels and pipes: Institute of Civil Engineers of Ireland Transactions, v. 20, 1891.

Minderhoud, P. S. J., Erkens, G., Pham, V. H., Bui, V. T., Erban, L., Kooi, H., and Stouthamer, E.: Impacts of 25 years of groundwater extraction on subsidence in the Mekong delta, Vietnam, Environ. Res. Lett., 12, https://doi.org/10.1088/1748-9326/aa7146, 2017.

760   MPI Forum: MPI: A Message-Passing Interface Standard, University of Tennessee, Knoxville, TN, USA, 1994.

Pellegrini, F.: Scotch and libScotch 5.1 User's Guide. hal-00410327, 2008.

Rünger, G. and Rauber, T.: Parallel Programming - for Multicore and Cluster Systems; 2nd Edition., Springer, I–XIII, 1–516 pp., 2013.

Smith, B. F., Bjørstad, P. E., and Gropp, W. D.: Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial

765   Differential Equations, Cambridge University Press, New York, NY, USA, 1996.

SURFsara: Description of the Snellius system, https://servicedesk.surfsara.nl/wiki/display/WIKI/Snellius+and+Lisa, last access: 1 January 2022, 2021.

Sutanudjaja, E. H., Van Beek, L. P. H., De Jong, S. M., Van Geer, F. C., and Bierkens, M. F. P.: Large-scale groundwater modeling using global datasets: A test case for the Rhine-Meuse basin, Hydrol. Earth Syst. Sci., 15, 2913–2935,

770   https://doi.org/10.5194/hess-15-2913-2011, 2011.

Sutanudjaja, E. H., Van Beek, L. P. H., De Jong, S. M., Van Geer, F. C., and Bierkens, M. F. P.: Calibrating a large-extent high-resolution coupled groundwater-land surface model using soil moisture and discharge data, Water Resour. Res., 50, 687–705, https://doi.org/10.1002/2013WR013807, 2014.

Sutanudjaja, E. H., Van Beek, R., Wanders, N., Wada, Y., Bosmans, J. H. C., Drost, N., Van Der Ent, R. J., De Graaf, I. E.

775   M., Hoch, J. M., De Jong, K., Karssenberg, D., López López, P., Peßenteiner, S., Schmitz, O., Straatsma, M. W., Vannametee, E., Wisser, D., and Bierkens, M. F. P.: PCR-GLOBWB 2: A 5 arcmin global hydrological and water resources model, Geosci. Model Dev., 11, 2429–2453, https://doi.org/10.5194/gmd-11-2429-2018, 2018.

Tudor, B. M., Teo, Y. M., and See, S.: Understanding off-chip memory contention of parallel programs in multicore systems, Proc. Int. Conf. Parallel Process., 602–611, https://doi.org/10.1109/ICPP.2011.59, 2011.

780   U.S. Geological Survey: National Water Information System: U.S. Geological Survey web interface, https://doi.org/https://doi.org/10.5066/ F7P55KJN, 2021a.

U.S. Geological Survey: Watershed Boundary Dataset, accessed November 6, 2021 at https://www.usgs.gov/national-hydrography/access-national-hydrography-products, 2021b.

Verdin, K. L. and Verdin, J. P.: A topological system for delineation and codification of the Earth's river basins, J. Hydrol.,

785   218, 1–12, https://doi.org/10.1016/S0022-1694(99)00011-6, 1999.

Verkaik, J., Hughes, J. D., and Langevin, C. D.: Parallel Groundwater Modeling Using MODFLOW 6, in: AGU Fall Meeting Abstracts, https://agu.confex.com/agu/fm18/meetingapp.cgi/Paper/354079, 2018.

Verkaik, J., van Engelen, J., Huizer, S., Bierkens, M. F. P., Lin, H. X., and Oude Essink, G. H. P.: Distributed memory parallel computing of three-dimensional variable-density groundwater flow and salt transport, Adv. Water Resour., 154, 103976,

790   https://doi.org/10.1016/j.advwatres.2021.103976, 2021a.

Verkaik, J., Hughes, J. D., van Walsum, P. E. V, Oude Essink, G. H. P., Lin, H. X., and Bierkens, M. F. P.: Distributed memory parallel groundwater modeling for the Netherlands Hydrological Instrument, Environ. Model. Softw., 143, 105092, https://doi.org/10.1016/j.envsoft.2021.105092, 2021b.

Verkaik, J., Hughes, J. D., and Langevin, C. D.: Parallel MODFLOW 6.2.1 prototype release 0.1,

795   https://doi.org/10.5281/zenodo.5778658, December 2021c.

Verkaik, J., Sutanudjaja, E. H.: The GLOBGM data and tools version 1.0, https://doi.org/10.5281/zenodo.7398200, December 2022.

Vivoni, E. R., Mascaro, G., Mniszewski, S., Fasel, P., Springer, E. P., Ivanov, V. Y., and Bras, R. L.: Real-world hydrologic assessment of a fully-distributed hydrological model in a parallel computing environment, J. Hydrol., 409, 483–496, https://doi.org/10.1016/j.jhydrol.2011.08.053, 2011.

Wada, Y., van Beek, L. P. H., van Kempen, C. M., Reckman, J. W. T. M., Vasak, S., and Bierkens, M. F. P.: Global depletion of groundwater resources, Geophys. Res. Lett., 37, https://doi.org/https://doi.org/10.1029/2010GL044571, 2010.

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 Resour. Res., 47, https://doi.org/https://doi.org/10.1029/2010WR010090, 2011.

Yamazaki, D., Ikeshima, D., Tawatari, R., Yamaguchi, T., O'Loughlin, F., Neal, J. C., Sampson, C. C., Kanae, S., and Bates, P. D.: A high-accuracy map of global terrain elevations, Geophys. Res. Lett., 44, 5844–5853, https://doi.org/https://doi.org/10.1002/2017GL072874, 2017.

Zell, W. O. and Sanford, W. E.: Calibrated Simulation of the Long-Term Average Surficial Groundwater System and Derived Spatial Distributions of its Characteristics for the Contiguous United States, Water Resour. Res., 56, 1–16, https://doi.org/10.1029/2019WR026724, 2020.

Zhang, S., Fu, H., Wu, L., Li, Y., Wang, H., Zeng, Y., Duan, X., Wan, W., Wang, L., Zhuang, Y., Meng, H., Xu, K., Xu, P., Gan, L., Liu, Z., Wu, S., Cheng, Y., Yu, H., Shi, S., Wang, L., Xu, S., Xue, W., Liu, W., Guo, Q., Zhang, J., Zhu, G., Tu, Y., Edwards, J., Baker, A., Yong, J., Yuan, M., Yu, Y., Zhang, Q., Liu, Z., Li, M., Jia, D., Yang, G., Wei, Z., Pan, J., Chang, P., Danabasoglu, G., Yeager, S., Rosenbloom, N., and Guo, Y.: Optimizing High-Resolution Community Earth System Model on a Heterogeneous Many-Core Supercomputing Platform (CESM-HR_sw1.0), Geosci. Model Dev. Discuss., 2020, 1–38, https://doi.org/10.5194/gmd-2020-18, 2020.

800

805

810

815

820