



Supplement of

Pitfalls and a feasible solution for using KGE as an informal likelihood function in MCMC methods: DREAM_(ZS) as an example

Yan Liu et al.

Correspondence to: Yan Liu (yan.liu@hydmod.uni-freiburg.de)

The copyright of individual parts of the supplement might differ from the article licence.

Supplement

This supplement contains two figures and one text (Text S1). Figure S1 demonstrates the model performance during the calibration period for case study 2, and Figure S2 shows the parameter uncertainty for case study 3. Text S1 shows the Matlab code on how to set the probability function using KGE and gamma distribution function.

- **Figure S1**

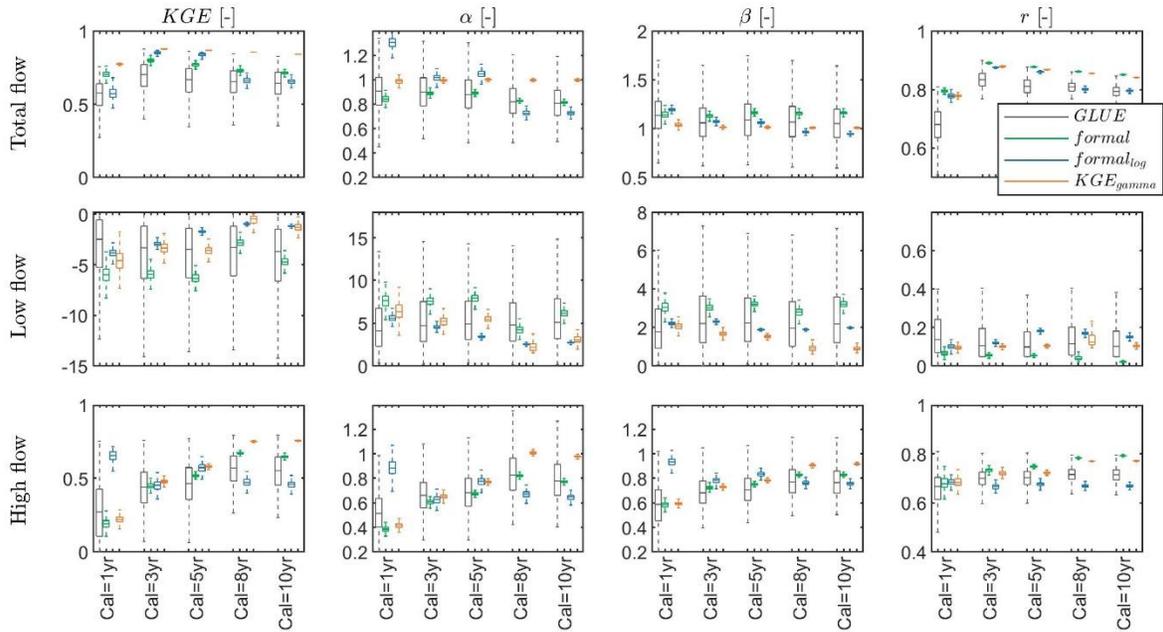


Figure S1 General performance (KGE), variability (α), non-scaled bias (β) and correlation (r) for total flow, low flow (smaller than 10th percentile of observed discharge) and high flow (larger than 90th percentile of observed discharge) during the calibration period using the *GLUE* approach (*GLUE*), the formal likelihood function (*formal*), formal likelihood function with log-transformation (*formal_{log}*) and our approach using KGE and gamma distribution to derive probability density (*KGE_{gamma}*) with varying amount of observations (1-year to 10-year) in calibration, for instance, calibration with 1-year observations is shown as Cal=1yr. The boxplot shows the performance of the last 25% of all simulations (top 25% in performance for *GLUE*), which is used to approximate the “true” system behavior in *DREAM_(zS)*. The optimal value for KGE , α , β and r is one, and the closer to one the better the performance.

- **Figure S2**

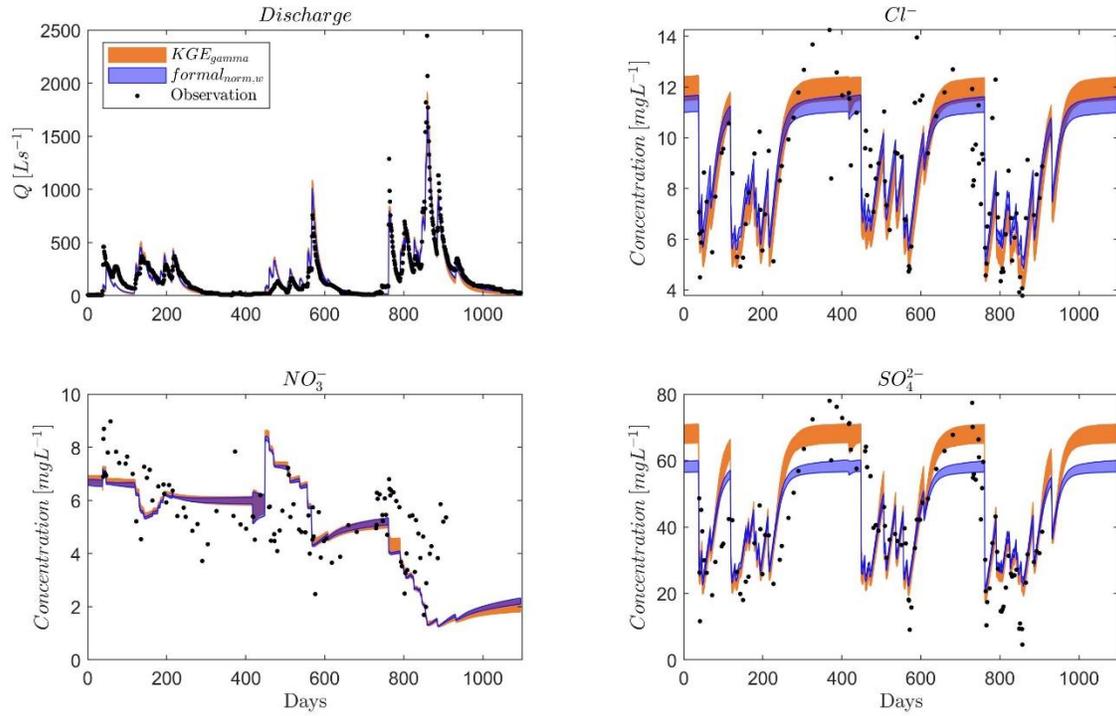


Figure S2 Parameter uncertainty for discharge and solutes (Cl^- , NO_3^- and SO_4^{2-}) for a heterogeneous karst system using the formal likelihood function with equal weights of normalized discharge and each solute ($\text{formal}_{\text{norm,w}}$) and our approach using KGE and gamma distribution to derive probability density ($\text{KGE}_{\text{gamma}}$). The parameter uncertainty is estimated using the last 25% of all simulations.

- **Text S1**

Step 1: set the likelihood function type to the log-likelihood

```
% Choose the number 2, which is the log-likelihood
DREAMPar.lik = 2;
```

Step 2: set DREAM(zs) to call the function to calculate the log-likelihood with KGE and gamma distribution function

```
% Define the name of function (.m file) for posterior exploration
Func_name = 'loglikelihoodCalculator';
```

```
function [log_L] = loglikelihoodCalculator(parameter)
% calculate the log-likelihood using KGE and gamma distribution function
% based on the simulation of a model (such as HBVhumped) and pass to DREAM
```

```
% Run the simulation
[Qsim] = HBVlumped(parameter);
```

```
% Load observations
Qdata = load('obs.mat');
Qobs = Qdata.Q;
```

```
% calculate KGE
n=length(Qobs);
KGE = KGECompute(Qsim,Qobs);
ED = 1-KGE;
```

```
% calculate log-likelihood function
gampdf = gampdf(ED,0.5,1);
log_L = 0.5*n*log(gampdf);
```

```
end
```

```
function [KGE]=KGECompute(sim,obs)
% compute KGE
obsMu = mean(obs);
simMu = mean(sim);
obsSigma = std(obs);
simSigma = std(sim);
covAllSO = cov(sim,obs);
covSO = covAllSO(1,2);
r = covSO/(obsSigma*simSigma);
alpha = simSigma/obsSigma;
beta = simMu/obsMu;
KGE = 1-sqrt((r-1)^2+(alpha-1)^2+(beta-1)^2);
end
```