

A Computational Tool to Track Sewage Flow Discharge into Rivers Based on Coupled HEC-RAS and DREAM

Junbo Wen ^{1,2}, Mengdie Ju ^{1,2}, Zichen Jia ^{1,2}, Lei Su ³, Shanshan Wu ³, Yuting Su ³, Wenxiao Liufu ³ and Hailong Yin ^{1,2,*}

¹ College of Environmental Science and Engineering, Tongji University, Shanghai 200092, China; 2230583@tongji.edu.cn (J.W.); hjmengdie@163.com (M.J.); 2331301@tongji.edu.cn (Z.J.)

² State Key Laboratory of Pollution Control and Resource Reuse Research, Tongji University, Shanghai 200092, China

³ Guangzhou Resource Environmental Protection Technology Co., Ltd., Guangzhou 5100075, China; sul@rcnzy.com (L.S.); wuss@rcnzy.com (S.W.); suyt@rcnzy.com (Y.S.); soso8537@126.com (W.L.)

* Correspondence: yinhailong@tongji.edu.cn

S1. Coupling HEC-RAS with DREAM in MATLAB

The diagram of the coupled DREAM and HEC-RAS system for tracking sewage inputs and their discharge flows is illustrated in Figure S1. In summary, the basic configuration of the framework is established in the file 'runFramework.m' where essential elements such as HEC-RAS, function files, and observational data are configured and set. Then, the function file 'Suyuan.m' interacts with HEC-RAS through three sub-function files to update the unsteady flow file, conduct HEC-RAS simulations, and return the likelihood function value for each simulation. The likelihood values are then returned to the DREAM toolbox for posterior calculation, differential evolution, and the Metropolis selection process to update a new generation of source parameters. This iterative process is repeated until the optimization termination criterion is satisfied. To facilitate a clear understanding of the workflow, the scripts for running the aforementioned models and the links between these models are briefly described in the following listings.

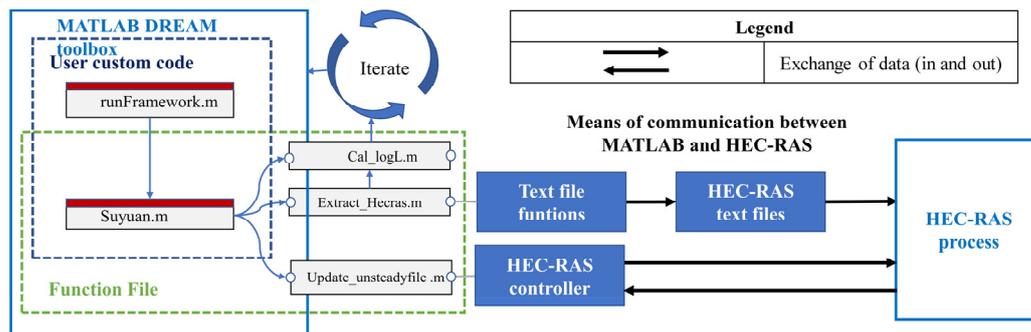


Figure S1. Schematic Diagram of the MATLAB HEC-RAS framework operation.

```

1   %% Clear memory and close pool of matlab nodes (cores)
2   clear all; clc; warning off
3   % Store working directory and subdirectory containing the files needed
4   global DREAM_dir EXAMPLE_dir CONV_dir HECRAS_dir Postprocessing_dir
5   DREAM_dir = [pwd '\DREAM_TOOLBOX']; EXAMPLE_dir = [DREAM_dir '\example_1'];
6   CONV_dir = [DREAM_dir '\diagnostics']; HECRAS_dir=[EXAMPLE_dir '\HECRAS'];
7   Postprocessing_dir = [DREAM_dir '\postprocessing'];
8   addpath(EXAMPLE_dir); addpath(CONV_dir); addpath(DREAM_dir);
9   addpath(HECRAS_dir);addpath(Postprocessing_dir);
10
11  %% Set user-supplied files
12  global filename_hec_u filename_hec_prj filename_func filename_observed_h
13  filename_hec_u = 'cihugong.u01';
14  filename_hec_prj='cihugong.prj';
15  filename_func ='suyuan';
16  filename_observed_h = 'observed_h.mat'; %hourly observed data within one day
17
18  %% Run the framework
19  % Set DREAM
20  cd(DREAM_dir);
21  %Define Function file,Optimization problem setting, Priori distribution
22  [Func_name,DREAMPPar,Par_info]=Set_DREAM(filename_func);
23  % Rename and relocate user-supplied files
24  cd(fileparts(pwd));
25  Move_user_file(filename_hec_u,filename_func,filename_observed_h,HECRAS_dir,...
26  EXAMPLE_dir,DREAMPPar)
27  % Call DREAM
28  cd(DREAM_dir);
29  [chain,output,fx]= DREAM(Func_name,DREAMPPar,Par_info);
30  % Plot the results of the DREAM
31  cd(Postprocessing_dir);
32  run DREAM_postproc.m;

```

Listing S1. Script to execute designed Framework in MATLAB.

Listing S1 presents the primary script used to execute the designed framework in MATLAB. Lines 4 to 9 in Listing S1 set up the working directory and subdirectory containing the files needed to run this framework, while Lines 11 to 16 set filenames of user-supplied files. The content of “Move_user_file.m” in Listing S1 is presented in Listing S2, which is used to relocate and rename user-supplied files from the original path (“sourceFolder”) to the specified path (“destinationFolder”).

Then, the user must define, in “Set_DRAM.m” shown in Listing S3, the optimization problem setting (“DREAMPPar”), a priori distribution of parameters (“Par_info”) and the function file of dream (“Func_name”). Finally, “DREAM.m” calls the DREAM Toolbox with the information provided by the user. The content of “Set_DREAM.m”, “DREAM.m” and “DREAM_postproc.m” in Listing S1 is adapted from [1]. Due to space constraints, the specifics are not reproduced here.

```

1  %% Rename and relocate user-supplied files
2  function Move_user_file(filename_hec_u,filename_func,filename_observed_h,...
3  HECRAS_dir,EXAMPLE_dir,DREAMPar)
4  % New temporal unsteady file(*.u##)
5  sourceFile_u = [pwd '\User-supplied files\HECRAS\' filename_hec_u]; % stemp_u01
6  destinationFolder = [pwd '\User-supplied files\HECRAS'];
7  newFileName_u = ['stemp_' filename_hec_u ];
8  copyfile(sourceFile_u, fullfile(destinationFolder, newFileName_u));
9  % Relocate and new HEC-RAS for parallel
10 for i= 1:DREAMPar.N
11     sourceFolder = [pwd '\User-supplied files\HECRAS'];
12     destinationFolder = HECRAS_dir;
13     newFolderName = ['HECRAS-' num2str(i)]; %For parallel computing
14     copyfile(sourceFolder, fullfile(destinationFolder, newFolderName));
15 end
16 % Relocate Function file and Observed data
17 sourcefile = [pwd '\User-supplied files\' filename_func '.m'];
18 destinationFolder = EXAMPLE_dir;
19 copyfile(sourcefile, destinationFolder);
20 sourcefile = [pwd '\User-supplied files\' filename_observed_h];
21 destinationFolder = EXAMPLE_dir;
22 copyfile(sourcefile, destinationFolder);
23 end

```

Listing S2. Script to rename and relocate user-supplied files.

```

1  function [Func_name,DREAMPar,Par_info]=Set_DREAM(filename_func)
2  % Problem settings defined by user
3      DREAMPar.d=25;
4      DREAMPar.N = 7; % Number of Markov chains
5      DREAMPar.T = 10000; % Number of generations
6      DREAMPar.lik = 2; % Model output is simulation:
7          %Gaussian log-likelihood with inference of measurement error
8      DREAMPar.prior = 'yes'; % Use an explicit prior distribution
9      DREAMPar.parallel = 'yes'; % Run each chain on a different core
10     DREAMPar.CPU=DREAMPar.N ;
11 % Provide prior information of parameters
12     Par_info.prior = 'latin'; % Latin hypercube sampling
13     Par_info.boundhandling = 'reflect'; % Explicit boundary handling
14 % Par_info.name = [ f1 f2 f3 f4 .....f24 Q]
15     Par_info.min = [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ];
16 %Q~m3/d
17     Par_info.max = [ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ] 20000 ;
18 % Define name of function(.m file) for posterior exploration
19     Func_name=filename_func;
20 end

```

Listing S3. Script for setting optimization options of DREAM. (Adapted from various scripts in [1])

```

1  function [log_L] = suyuan(x,process_index)
2  format long
3  global z1 z2 z3 z4 z5 z6 z7 z8 z9
4  global filename_hec_u filename_hec_prj filename_func filename_observed_h
5  %% Set user-supplied files
6  filename_hec_u = 'cihugong.u01';
7  filename_hec_prj='cihugong.prj';
8  filename_func = 'suyuan';
9  filename_observed_h = 'observed_h.mat';
10 %% Run the function file
11 % Convert x to lateral parameters
12 Q_daily = x(size(x,2))/3600;
13 f(1:size(x,2)-1) = x(1:size(x,2)-1);
14 q = zeros(1,25);
15 q(1,1:size(x,2)-1) = f * Q_daily;
16
17 % Automate the update and simulation of HEC-RAS
18 Update_unsteady_file(q,process_index,filename_hec_u);
19 z7=Extract_hecras(process_index,filename_hec_prj); % Execute HEC-RAS and
20                                                    % extract simulated value
21 log_L=Cal_logL(z7,filename_observed_h,f); % Return likelihood function

```

Listing S4. Script for automating the update and simulation of HEC-RAS from MATLAB.

As the function file of the DREAM toolbox, the main script in Listing S4 is presented for automating the update and simulation of HEC-RAS from MATLAB. The source parameters (“x”) in Listing S4 indicate the newly generated source parameters of the inverse model while (“process_index”) indicates which of the parallel chains is running this time. Next, “Update_unsteady_file.m” updates the unsteady flow file of HEC-RAS with newly generated lateral inflow parameters. Then, “Extract_hecras.m” in Listing S4 executes HEC-RAS and extracts simulated water levels at the downstream section. Finally, “Cal_logL.m” calculates the log-likelihood function (“log_L”) as the output of the function file. The content of “Update_unsteady_file.m”, “Extract_hecras.m” and “Cal_logL.m” are presented in Listing S5, A7, A8, respectively.

```

1  %% Update *.u## of HEC-RAS
2  function Update_unsteady_file(q,process_index,filename_hec_u)
3  info_cell = {};% To store processed text lines
4  [row,index] = deal(1,1);% Set the initial indexes
5  filenameinput = [pwd '/HECRAS/HECRAS-' num2str(process_index) '/stemp_'...
6                  filename_hec_u];
7  filenameoutput = [pwd '/HECRAS/HECRAS-' num2str(process_index) '/' ...
8                  filename_hec_u];
9
10
11 fid = fopen(filenameinput,'rt'); % Open file for reading
12 while ~feof(fid)
13     strTextLine = fgetl(fid); %Read a line of text
14     if contains(strTextLine,'Lateral Inflow Hydrograph=') %Find key variable
15         info_cell{row,1} = strTextLine;
16         row = row + 1;
17         temp = strsplit(strTextLine,' ');
18         num_of_time_steps = str2num(temp{1,2}); % Extract the number of time steps
19                                             % for the lateral flow
20         num_of_rows=num_of_time_steps/10;
21         for j = 1:num_of_rows
22             flowline = fgetl(fid); % Read a line of text
23             new_flowline = handleFlow(q(index)); % Format lateral parameters
24             info_cell{row,1} = new_flowline; % Update the lateral flow
25             row = row + 1;
26             index = index + 1;
27         end
28     else
29         info_cell{row,1} = strTextLine; %Copy original text
30         row = row + 1;
31     end
32 end
33
34 fout = fopen(filenameoutput,'wt'); % Open files for writing
35 for j = 1 : length(info_cell)
36     if ~isempty(info_cell{j,1}) % No blank lines for insurance testing
37         fprintf(fout,'%s\n',info_cell{j,1});
38     end
39 end
40
41 fclose(fid);
42 fclose(fout);
43 end

```

Listing S5. Script to update the unsteady flow file of HEC-RAS.

Listing S5 presents the script for updating the unsteady flow file of HEC-RAS from MATLAB. The variable ("filename_hec_u") indicates the filename of the user-specified unsteady flow file, while the variable ("info_cell") is used to temporarily store the updated text content. To start to replace the data, it is necessary to find key variables in the original unsteady file that is being read, which is "Lateral Inflow Hydrograph=" for the lateral flow discharge. Lines 14 to 31 in Listing S5 identify the rows of Lateral Inflow Hydrograph and replace original source parameters with the newly-generated source parameters produced by the DREAM algorithm, where the function ("handleFlow") is used to format the unsteady flow data to fit HEC-RAS, as shown in Listing S6. Then, the other content of the original unsteady flow file will be stored directly in the ("info_cell"). Finally, Lines 34 to 39 in Listing S5 write the processed content from ("info_cell") to the unsteady flow file of HEC-RAS line by line.

```

1  %% Format the lateral flow discharge for HEC-RAS
2  function new_flowline=handleFlow(q)
3      DISTANCE=8-1; % Each value in the *.u file takes up 8 characters of space
4      formatSpec = '%0.5f';
5      new_flow = num2str(q,formatSpec);
6      len = length(new_flow);
7      if len > DISTANCE
8          diff = len - DISTANCE;
9          formatSpec = ['%0.' num2str(6-diff) 'f'];
10         new_flow = num2str(q,formatSpec);
11         new_flow = [' ' new_flow ];
12     else
13         new_flow = [' ' new_flow ];
14     end
15     new_flowline = repmat(new_flow,1,10); % 10 time steps in a row
16 end

```

Listing S6. Script to format the unsteady flow data to fit HEC-RAS.

```

1  %% Run HEC-RAS and extract simulated water level
2  function z7 = Extract_hecras(process_index,filename_hec_prj)
3      h=actxserver('RAS500.HECRASCONTROLLER');
4      path=[pwd '/HECRAS/HECRAS-' num2str(process_index) '/' filename_hec_prj];
5      h.Project_Open(path); % Open ras file
6      h.Compute_HideComputationWindow; % Hide Comput. Window
7      h.Compute_CurrentPlan(0,0); % Run current plan
8      pause(1);
9      isComplete=h.Compute_Complete();
10     while isComplete == 0
11         pause(0.2);
12         isComplete=h.Compute_Complete(); %Ensure every parallel operation
13         %has been computed before saving data
14     end
15     h.Project_Save; % Save the project
16     z9 = 'error message';
17     [z1,z2,z3,z4,z5,z6,z7,z8,z9]= h.OutputDSS_GetStageFlow('cihu','reach_1','0',...
18                                                         0,0,0,0,z9);
19     h.QuitRas();
20     delete(h)
21 end

```

Listing S7. Script to run HEC-RAS and extract simulated water levels. (Adapted from various scripts in [2])

Listing S7 presents the script for automatically performing parallel computations of HEC-RAS in MATLAB [2]. The variable “isComplete” is used to check if all parallel operations have been computed successfully before saving data. If all computations are complete, data can be saved or further analyzed. Then, Line 17 in Listing S7 extracts the simulated water levels at the downstream cross-section as the variable (“z7”), which will soon be used to calculate the likelihood function in Listing 8.

```

1  %% Calculate likelihood function
2  function log_L=Cal_logL(z7,filename_observed_h,f)
3      sig2=0.001;
4      filepath= [pwd '/' filename_observed_h];
5      load(filepath,'obs_h0') ; % Load observed data
6      % Compare observed and simulated value
7      sum_h=0;
8      for k=1:25
9          nse=((z7(1,k)-obs_h0(1,k))/sig2)^2 ;
10         sum_h=nse+sum_h;
11     end
12     n=size(obs_h0,2)+1;
13     log_L= -(n/2)* log(2*pi) - n*log(sig2) - (1/2)*(sum_h+sum_f);
14 end

```

Listing S8. Script for calculating the likelihood function.

S2. Real case 1: Source tracking of a time-variable industrial discharge into the river

For each of the 25 source parameters set in Case 1, the R-statistic used to measure the convergence of the sampled chains under the time-varying industrial discharge case is shown in Figure S2.

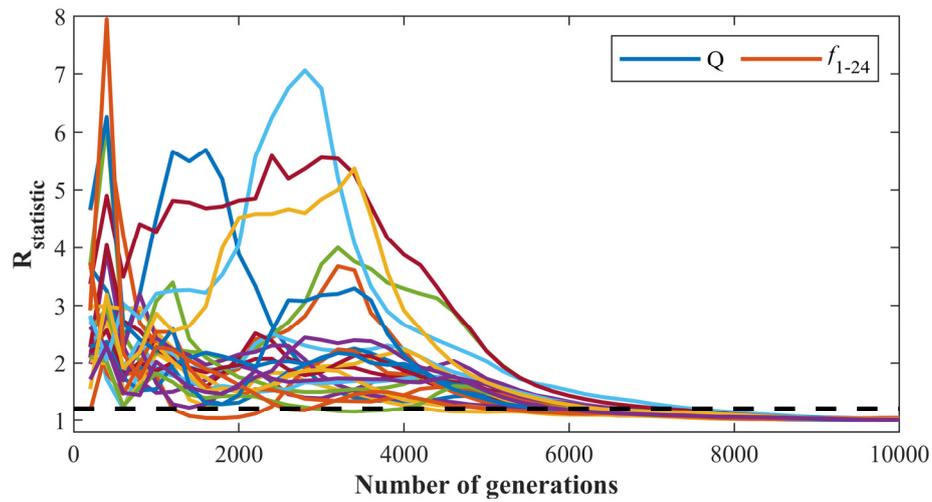
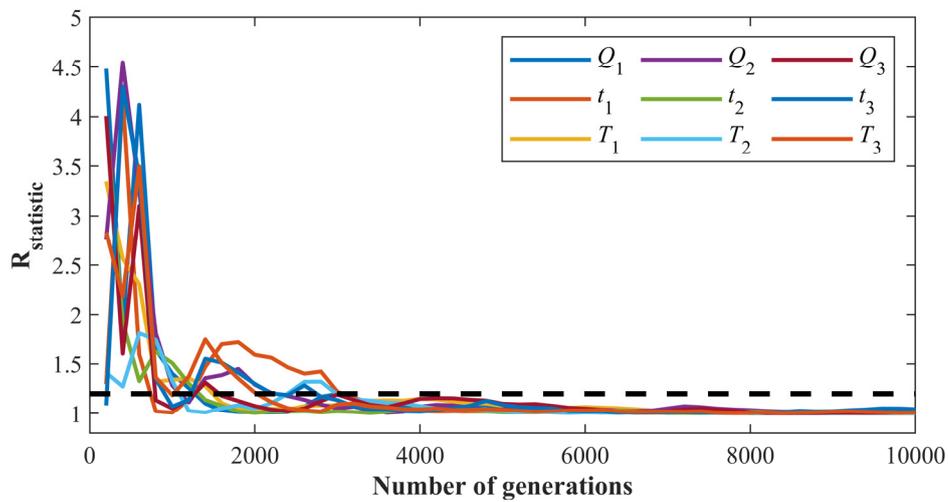


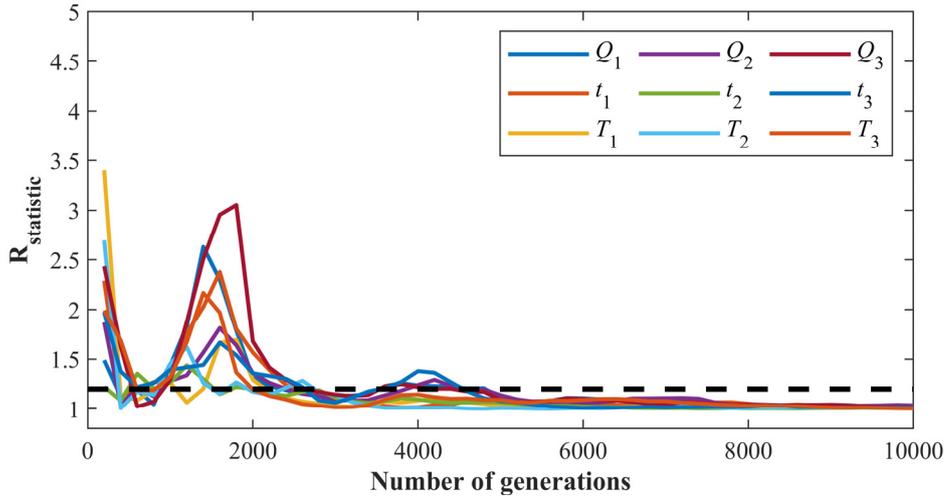
Figure S2. Convergence of R-statistic of the source parameters for Case 1.

S3. Real case 2: Source tracking of multiple sewage discharges into the river

For each of the nine source parameters set in Case 2, the R-statistic used to measure the convergence of the sampled chains under the case of wet and dry weather is shown in Figure S3. The resulting marginal probability density functions (PDF) are presented in Figure S4.

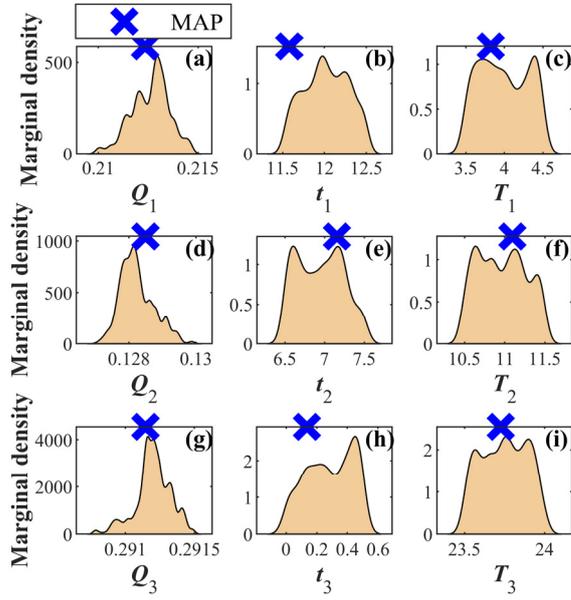


(a)

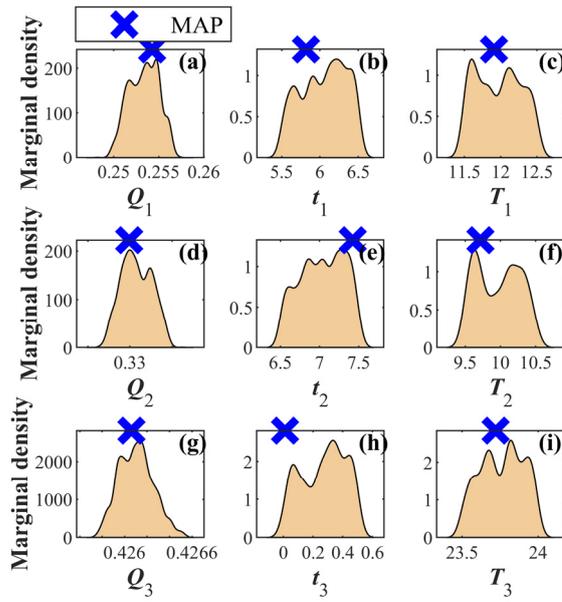


(b)

Figure S3. Convergence of R-statistic of the source parameters for Case 2. (a) Dry weather (b) Wet weather



(a)



(b)

Figure S4. Marginal PDFs of source parameters for the sewage flow discharge in Case 2 (a) Dry weather (b) Wet weather. Abbreviations: the unit of flow rate (Q) is in cubic meters per day (m^3/s).

References

1. Vrugt, J. A. Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB implementation. *Environmental Modelling & Software* 2016, 75, 273-316.
2. Leon, A. S.; Goodell, C. Controlling HEC-RAS using MATLAB. *Environmental Modelling & Software* 2016, 84, 339-348.