# Unit Hydrographs

This section provides details on the implementation of various Unit Hydrographs. An overview of the 7 UHs is given in Table 1. Computational implementation of each UH is given in sections 1.1 to 1.7. Unit Hydrograph files can be found in “./MARRMoT/Models/Unit Hydrograph files/”.

Table 1: Overview of Unit Hydrograph schemes in MARRMoT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Flux file*** | **Inputs** | **Diagram** | **Description** | **Used in model …** |
| uh\_1\_half | 1: amount to be routed  2: time base  3: Δt |  | Exponentially increasing scheme | 7 |
|  |  |  |  |  |
| uh\_2\_full | 1: amount to be routed  2: time base (time is doubled inside the function)  3: Δt |  | Exponential triangular scheme | 7 |
|  |  |  |  |  |
| uh\_3\_half | 1: amount to be routed  2: time base  3: Δt |  | Triangular scheme: linearly increasing | 13, 15, 21, 26, 34 |
|  |  |  |  |  |
| uh\_4\_full | 1: amount to be routed  2: time base  3: Δt |  | Triangular scheme: linearly increasing and decreasing | 0 (template), 16, 37,  nn (example) |
|  |  |  |  |  |
| uh\_5\_half | 1: amount to be routed  2: time base  3: Δt |  | Exponentially decreasing scheme | 5 |
|  |  |  |  |  |
| uh\_6\_gamma | 1: amount to be routed  2: gamma parameter [-]  3: time for flow to reduce by factor *e* [d]  4: length of time series |  | Gamma function-based | 40 |
|  |  |  |  |  |
| uh\_7\_uniform | 1: amount to be routed  2: time base  3: Δt |  | Uniform distribution | 39 |

## uh\_1\_half code

This section provides the computational implementation of a unit hydrograph with an increasing exponential distribution of flows.

File location: ./MARRMoT/Models/Unit Hydrograph files/uh\_1\_half  
References: E.g. GR4J (Perrin et al., 2003)

function [ out,UH ] = uh\_1\_half( in, d\_base, delta\_t )

%uh\_1\_half Unit Hydrograph [days] with half a bell curve. GR4J-based

%

% Inputs

% in - volume to be routed

% d\_base - time base of routing delay [d]

% delta\_t - time step size [d]

%

% Unit hydrograph spreads the input volume over a time period x4.

% Percentage of input returned only increases.

% I.e. d\_base = 3.8 [days], delta\_t = 1:

% UH(1) = 0.04 [% of inflow]

% UH(2) = 0.17

% UH(3) = 0.35

% UH(4) = 0.45

%%INPUTS

if any(size(in)) > 1; error('UH input should be a single value.'); end

%%TIME STEP SIZE

delay = d\_base/delta\_t;

if delay == 0; delay = 1; end % any value below t = 1 means no delay,

% but zero leads to problems

tt = 1:ceil(delay); % Time series for which we need UH

% ordinates [days]

%%EMPTIES

SH = zeros(1,length(tt)+1); SH(1) = 0;

UH = zeros(1,length(tt));

%%UNIT HYDROGRAPH

for t = tt

if t < delay; SH(t+1) = (t./delay).^(5./2);

elseif t >= delay; SH(t+1) = 1;

end

UH(t) = SH(t+1)-SH(t);

end

%%DISPERSE VOLUME

out = in.\*UH;

end

## uh\_2\_full code

This section provides the computational implementation of a unit hydrograph with an approximate exponentially-shaped triangular distribution of flows.

File location: ./MARRMoT/Models/Unit Hydrograph files/uh\_1\_half  
References: E.g. GR4J (Perrin et al., 2003)

function [ out, UH ] = uh\_2\_full( in,d\_base,delta\_t )

%uh\_2\_full Unit Hydrograph [days] with a full bell curve. GR4J-based

%

% Inputs

% in - volume to be routed

% d\_base - time base of routing delay [d]

% delta\_t - time step size [d]

%

% Unit hydrograph spreads the input volume over a time period 2\*x4.

% Percentage of input returned goes up (till x4), then down again.

% I.e. d\_base = 3.8 [days], delta\_t = 1:

% UH(1) = 0.02 [% of inflow]

% UH(2) = 0.08

% UH(3) = 0.18

% UH(4) = 0.29

% UH(5) = 0.24

% UH(6) = 0.14

% UH(7) = 0.05

% UH(8) = 0.00

%%INPUTS

if any(size(in)) > 1; error('UH input should be a single value.'); end

%%TIME STEP SIZE

delay = d\_base/delta\_t;

tt = 1:2\*ceil(delay); % time series for which we need UH ordinates [days]

%%EMPTIES

SH = zeros(1,length(tt)+1); SH(1) = 0;

UH = zeros(1,length(tt));

%%UNIT HYDROGRAPH

for t = tt

if (t <= delay)

SH(t+1) = 0.5\*(t./delay).^(5./2);

elseif (t > delay) && (t < 2\*delay);

SH(t+1) = 1-0.5\*(2-t./delay).^(5./2);

elseif (t >= 2\*delay);

SH(t+1) = 1;

end

UH(t) = SH(t+1)-SH(t);

end

%%DISPERSE VOLUME

out = in.\*UH;

end

## uh\_3\_half code

This section provides the computational implementation of a unit hydrograph with an increasing triangular distribution of flows.

File location: ./MARRMoT/Models/Unit Hydrograph files/uh\_3\_half  
References: E.g. FLEX\_Topo (Savenije, 2010)

function [ out,UH ] = uh\_3\_half( in, d\_base, delta\_t )

%uh\_3\_half Unit Hydrograph [days] with half a triangle (linear)

%

% Inputs

% in - volume to be routed

% d\_base - time base of routing delay [d]

% delta\_t - time step size [d]

%

% Unit hydrograph spreads the input volume over a time period delay.

% Percentage of input returned only increases.

% I.e. d\_base = 3.8 [days], delta\_t = 1:

% UH(1) = 0.04 [% of inflow]

% UH(2) = 0.17

% UH(3) = 0.35

% UH(4) = 0.45

%%INPUTS

if any(size(in)) > 1; error('UH input should be a single value.'); end

%%TIME STEP SIZE

delay = d\_base/delta\_t;

if delay == 0; delay = 1; end % any value below t = 1 means no delay,

% but zero leads to problems

tt = 1:ceil(delay); % time series for which we need UH

% ordinates [days]

%%UNIT HYDROGRAPH

% The area under the unit hydrograph by definition sums to 1. Thus the area

% is S(t=0 to t = delay) t\*[ff: fraction of flow to move per time step] dt

% Analytical solution is [1/2 \* t^2 + c]\*ff, with c = 0. Thus the fraction

% of flow step size is:

ff = 1/(0.5\*delay^2);

%%EMPTIES

UH = zeros(1,length(tt));

%%UNIT HYDROGRAPH

for t = 1:length(tt)

if t <= delay

UH(t) = ff.\*(0.5\*t^2 - 0.5\*(t-1)^2);

else

UH(t) = ff.\*(0.5\*delay^2 - 0.5\*(t-1)^2);

end

end

%%DISPERSE VOLUME

out = in.\*UH;

end

## uh\_4\_full code

This section provides the computational implementation of a unit hydrograph with an increasing and decreasing triangular distribution of flows.

File location: ./MARRMoT/Models/Unit Hydrograph files/uh\_4\_full  
References: E.g. HBV-96 (Lindström et al., 1997)

function [ out,UH ] = uh\_4\_full( in, d\_base, delta\_t )

%uh\_4\_half Unit Hydrograph [days] with a triangle (linear)

%

% Inputs

% in - volume to be routed

% d\_base - time base of routing delay [d]

% delta\_t - time step size [d]

%

% Unit hydrograph spreads the input volume over a time period delay.

% Percentage runoff goes up, peaks, and goes down again.

% I.e. d\_base = 3.8 [days], delta\_t = 1:

% UH(1) = 0.14 [% of inflow]

% UH(2) = 0.41

% UH(3) = 0.36

% UH(4) = 0.09

%%INPUTS

if any(size(in)) > 1; error('UH input should be a single value.'); end

%%TIME STEP SIZE

delay = d\_base/delta\_t;

if delay == 0; delay = 1; end % any value below t = 1 means no delay,

% but zero leads to problems

tt = 1:ceil(delay); % time series for which we need UH

% ordinates [days]

%%UNIT HYDROGRAPH

% The area under the unit hydrograph by definition sums to 1. Thus the area

% is S(t=0 to t = delay) t\*[ff: fraction of flow to move per time step] dt

% Analytical solution is [1/2 \* t^2 + c]\*ff, with c = 0.

% Here, we use two half triangles t make one big one, so the area of half a

% triangle is 0.5. Thus the fraction of flow step size is:

ff = 0.5/(0.5\*(0.5\*delay)^2);

d50 = 0.5\*delay;

%%TRIANGLE FUNCTION

tri = @(t) max(ff.\*(t-d50).\*sign(d50-t)+ff.\*d50,0);

%%EMPTIES

UH = zeros(1,length(tt));

%%UNIT HYDROGRAPH

for t = 1:length(tt)

UH(t) = integral(tri,t-1,t);

end

%%ENSURE UH SUMS TO 1

tmp\_diff = 1-sum(UH);

tmp\_weight = UH./sum(UH);

UH = UH + tmp\_weight.\*tmp\_diff;

%%DISPERSE VOLUME

out = in.\*UH;

end

## uh\_5\_half code

This section provides the computational implementation of a unit hydrograph with an exponentially decreasing distribution of flows.

File location: ./MARRMoT/Models/Unit Hydrograph files/uh\_5\_half  
References: E.g. IHACRES (Croke and Jakeman, 2004; Littlewood et al., 1997)

function [ out,UH ] = uh\_5\_half( in, d\_base, delta\_t )

%uh\_5\_half Unit Hydrograph [days] with half a triangle (exponential decay)

%

% Inputs

% in - volume to be routed

% d\_base - time base of routing delay [d]

% delta\_t - time step size [d]

%

% Unit hydrograph spreads the input volume over a time period delay.

% Percentage of input returned only decreases.

% I.e. d\_base = 3.8 [days], delta\_t = 1:

% UH(1) = 0.841 [% of inflow]

% UH(2) = 0.133

% UH(3) = 0.021

% UH(4) = 0.004

%%INPUTS

if any(size(in)) > 1; error('UH input should be a single value.'); end

%%TIME STEP SIZE

delay = d\_base/delta\_t;

if delay == 0; delay = 1; end % any value below t = 1 means no delay,

% but zero leads to problems

tt = 1:ceil(delay); % time series for which we need UH

% ordinates [days]

%%UNIT HYDROGRAPH

% The Unit Hydrograph follows exponential decay y=exp(-x). For a given

% delay time, the fraction of flow per time step is thus the integral of

% t-1 to t of the exponential decay curve. The curve has range [0,Inf>.

% We impose the arbitrary boundary of [0,7] here (exp(-7) = 9e-4) as the

% point where the decay curve 'ends'. This allows to divide the range [0,7]

% in n delay steps, and so calculate the UH.

%%Find integral limits

stepsize = (7-0)/delay; % Range over which the decay curve is

% calculated, divided by required

% number of delay steps

limits = 0:stepsize:7;

limits(end+1) = 7;

%%EMPTIES

UH = zeros(1,length(tt));

%%UNIT HYDROGRAPH

for t = 1:length(tt)

UH(t) = integral(@(x) exp(-x),limits(t),limits(t+1));

end

%%ACCOUNT FOR <7,Inf> PART OF THE CURVE (i.e. add the missing tail end of

% the curve to the last delay step, to ensure that 100% of flow is routed).

UH(end) = UH(end)+(1-sum(UH));

%%DISPERSE VOLUME

out = in.\*UH;

end

## uh\_6\_gamma code

This section provides the computational implementation of a unit hydrograph with a gamma distribution of flows.

File location: ./MARRMoT/Models/Unit Hydrograph files/uh\_6\_gamma  
References: E.g. SMAR (O’Connell et al., 1970; Tan and O’Connor, 1996)

function [ out,UH,frac\_routing\_beyond\_time\_series ] = ...

uh\_6\_gamma( in,n,k,t\_end,delta\_t )

%uh\_6\_gamma Unit Hydrograph [days] from gamma function.

% n = shape parameter [-]

% k = time delay for flow reduction by a factor e [d]

% t\_end = length of time series [d]

% delta\_t = time step size [d]

%

% Unit hydrograph spreads the input volume over a time period delay.

% Percentage of input returned only decreases.

% I.e. n = 1, k = 3.8 [days], delta\_t = 1:

% UH(1) = 0.928 [% of inflow]

% UH(2) = 0.067

% UH(3) = 0.005

% UH(4) = 0.000

%%INPUTS

if any(size(in)) > 1; error('UH input should be a single value.'); end

%%TIME STEP SIZE

tmax = t\_end/delta\_t;

tt = 1:tmax; % time series for which we need UH ordinates [days]

%%EMPTIES

UH\_full = zeros(1,length(tt));

frac\_routing\_beyond\_time\_series = 0;

%%UNIT HYDROGRAPH

% The Unit Hydrograph follows a gamma distribution. For a given

% delay time, the fraction of flow per time step is thus the integral of

% t-1 to t of the gamma distrubtion. The curve has range [0,Inf>.

% We need to choose a point at which to cap the integration, but this

% depends on the parameters n & k, and the total time step. We choose the

% cutoff point at the time step where less than 0.1% of the peak flow

% is still on route.

%%Unit hydrograph

for t = 1:length(tt)

UH\_full(t) = integral(@(x) 1./(k.\*gamma(n)).\*(x./k).^(n-1).\* ...

exp(-1.\*x./k),(t-1)\*delta\_t,t\*delta\_t);

end

%%Find cutoff point where less than 0.1% of the peak flow is being routed

[max\_val,max\_here] = max(UH\_full);

end\_here = find(UH\_full(max\_here:end)./max\_val<0.001,1) + max\_here;

%%Take action depending on whether the distribution function exceeds the

%%time limit or not

if ~isempty(end\_here)

%%Construct the Unit Hydrograph

UH = UH\_full(1:end\_here);

%%Account for the truncated part of the full UH.

% find probability mass to the right of the cut-off point

tmp\_excess = 1-sum(UH);

% find relative size of each time step

tmp\_weight = UH\_full(1:end\_here)./sum(UH\_full(1:end\_here));

% distribute truncated probability mass proportionally to all elements

% of the routing vector

UH = UH+tmp\_weight.\*tmp\_excess;

else

%%Construct the Unit Hydrograph

UH = UH\_full;

%%The UH is longer than the provided time series length. Track the

%%percentage of flow that is routed beyond the simulation duration

frac\_routing\_beyond\_time\_series = 1-sum(UH);

end

%%DISPERSE VOLUME

out = in.\*UH;

end

## uh\_7\_uniform code

This section provides the computational implementation of a unit hydrograph with a uniform distribution of flows.

File location: ./MARRMoT/Models/Unit Hydrograph files/uh\_7\_uniform  
References: E.g. MCRM (Bell et al., 2001; Moore and Bell, 2001)

function [ out,UH ] = uh\_7\_uniform( in, d\_base, delta\_t )

%uh\_7\_uniform Unit Hydrograph [days] with uniform spread

%

% Inputs

% in - volume to be routed

% d\_base - time base of routing delay [d]

% delta\_t - time step size [d]

%

% Unit hydrograph spreads the input volume over a time period delay.

% I.e. d\_base = 3.8 [days], delta\_t = 1:

% UH(1) = 0.26 [% of inflow]

% UH(2) = 0.26

% UH(3) = 0.26

% UH(4) = 0.22

%%INPUTS

if any(size(in)) > 1; error('UH input should be a single value.'); end

%%TIME STEP SIZE

delay = d\_base/delta\_t;

tt = 1:ceil(delay); % time series for which we need UH ordinates [days]

%%EMPTIES

UH = NaN.\*zeros(1,length(tt));

%%FRACTION FLOW

ff = 1/delay; % fraction of flow per time step

%%UNIT HYDROGRAPH

for t=1:ceil(delay)

if t < delay

UH(t) = ff;

else

UH(t) = mod(delay,t-1)\*ff;

end

end

%%DISPERSE VOLUME

out = in.\*UH;

end

**References**

Bell, V. A., Carrington, D. S. and Moore, R. J.: Comparison of rainfall-runoff models for flood forecasting - part 2., 2001.

Croke, B. F. W. and Jakeman, A. J.: A catchment moisture deficit module for the IHACRES rainfall-runoff model, Environ. Model. Softw., 19(1), 1–5, doi:10.1016/j.envsoft.2003.09.001, 2004.

Lindström, G., Johansson, B., Persson, M., Gardelin, M. and Bergström, S.: Development and test of the distributed HBV-96 hydrological model, J. Hydrol., 201, 272–288, doi:https://doi.org/10.1016/S0022-1694(97)00041-3, 1997.

Littlewood, I. G., Down, K., Parker, J. R. and Post, D. A.: IHACRES v1.0 User Guide., 1997.

Moore, R. J. and Bell, V. A.: Comparison of rainfall-runoff models for flood forecasting. Part 1: Literature review of models, Environment Agency, Bristol., 2001.

O’Connell, P. E., Nash, J. E. and Farrell, J. P.: River flow forecasting through conceptual models part II - the Brosna catchment at Ferbane, J. Hydrol., 10, 317–329, 1970.

Perrin, C., Michel, C. and Andréassian, V.: Improvement of a parsimonious model for streamflow simulation, J. Hydrol., 279(1-4), 275–289, doi:10.1016/S0022-1694(03)00225-7, 2003.

Savenije, H. H. G.: “Topography driven conceptual modelling (FLEX-Topo),” Hydrol. Earth Syst. Sci., 14(12), 2681–2692, doi:10.5194/hess-14-2681-2010, 2010.

Tan, B. Q. and O’Connor, K. M.: Application of an empirical infiltration equation in the SMAR conceptual model, J. Hydrol., 185(1-4), 275–295, doi:10.1016/0022-1694(95)02993-1, 1996.