

User Manual of the Characteristic Point Method for Automatic Hydrograph Separation

Yiwen Mei and Emmanouil N. Anagnostou

Civil and Environmental Engineering, University of Connecticut

Last updated: April 2017

1. Overview

This document is the user manual of codes designed for the hydrograph separation method reported in Mei & Anagnostou (2015) (hereafter referred as MA2015). The hydrograph separation method includes two aspects: baseflow separation and event identification. The baseflow separation method, namely filtered revised constant k (FRCK) method, is a hybrid-method of the revised constant k (RCK) and recursive digital filter (RDF); it splits the streamflow record into baseflow and event flow component. The event identification method is called characteristic point method (CPM); it extracts rainfall-runoff events from long records of streamflow and basin-average rainfall time series, based on the time series characteristics.

The above hydrograph separation methods are contained in six Matlab codes: two for the baseflow separation section and four codes for the event identification. The manuscript provides detailed explanation on the uses, inputs and outputs of these Matlab codes. It is accompanied by a demo Matlab code, illustrating an example basin.

2. Setting up CPM

The CPM program is written in Matlab. The entire package contains the P-codes of the program (in the Pcodes folder), a demonstration M-code of the program (Demo.m), the necessary sample flow data (in the Demo_data folder) for the demonstration and this CPM User Manual

(CPM User Manual.docx). Within the Pcodes folder there are six P-codes in the names of RCK.p, FRCK.p, CPM_peak.p, CPM_FE.p, CPM_RE.p and CPM_RFE.p. Within the Demo_data folder there is a mat file named Demo_data.mat storing the time series of flow and basin-average precipitation of the sample basin (named Swift and ranging from 2002 to 2013) and the area of the basin. The example is ready to run; the users need to have Matlab (any version after/including 2010 with the Statistics Toolbox) installed in their own PC before running the demonstration code (Demo.m).

To run CPM for any other basin, the users need to prepare the flow time series, the basin-average rainfall and the area of the basin (a scalar in km²) in one mat file. Then they can follow Demo.m to get the results. The required parameter values are originally designed and has been tested for hourly resolution time series. The method may not readily apply to time series associated with other resolutions, as more investigation is required to scale the method to other resolutions.

3. Running CPM

Schemes of the baseflow separation and event identification method are illustrated in Figure 1. Meanings of the symbols are the same as those used in MA2015 and are listed in Table 1. As it is seen in panel a of Figure 1, the entire baseflow separation section is consisted by two Matlab functions, namely the RCK and FRCK, representing the revised constant k method and its recursive digital filtered version. The event identification method (CPM) in panel b of the figure includes four Matlab codes. Essence of CPM is the identification of characteristic points of the rainfall and runoff events (i.e. begin, end and peak points). Peak flow points are identified by code CPM_peak (begin and end points of a flow event are provided from the baseflow section). Flow and rainfall events are formed through CPM_FE and CPM_RE, respectively. These codes associate the timing of rainfall and flow event to form the rainfall-runoff events. Detail instructions are provided in the following subsections, accompanied with an example.

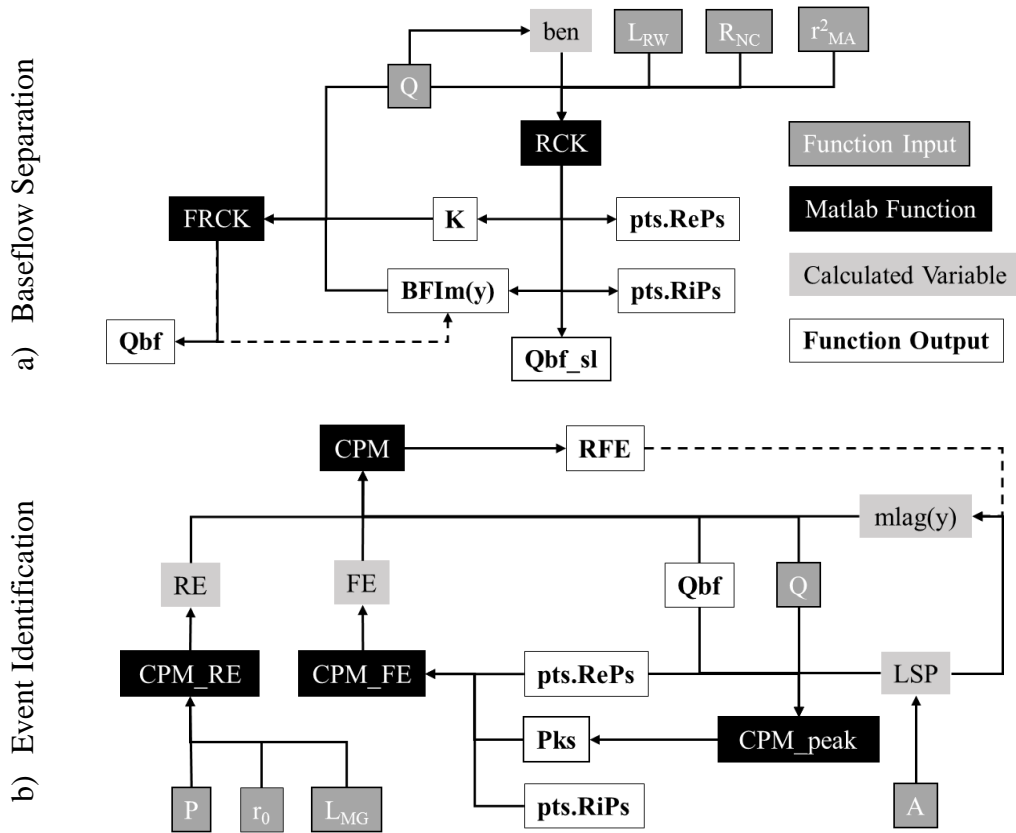


Figure 1. Schematics of a) baseflow separation and b) event identification.

Table 1. Names of the variables and meanings of the abbreviations.

| Symbol | Unit | Name |
|------------------|-----------------|--|
| Q | mm/h | Flow time series |
| L_{RW} | h | Length of regression window |
| R_{NC} | h^{-1} | Ratio of no-change |
| r_{MA}^2 | \ | Maximum acceptable coefficient of determination |
| ben | mm/h | Envelope baseflow rate |
| Qbf_sl | mm/h | Revised constant k baseflow time series |
| pts.RiPs | | Rising point |
| pts.RePs | | Recessing point |
| K | h^{-1} | Recession coefficient |
| BFI _m | \ | Maximum baseflow index |
| Qbf | mm/h | Filtered revised constant k baseflow time series |
| A | km ² | Drainage area |
| LSP | h | Length of searching period |
| Pks | | Peak flow point |
| FE | | Flow event |
| P | mm/h | Precipitation time series |

| | | |
|-----------|------|--------------------------------|
| r_0 | mm/h | Trace rain rate |
| L_{MG} | h | Length of maximum rainfall gap |
| RE | | Rainfall event |
| R_s | h | Searching radius |
| t_{lag} | h | Time lag of event |
| RFE | | Rainfall-runoff event |

3.1. Baseflow Separation

The baseflow separation components introduced in MA2015 are the revised constant k (RCK) method and the filtered RCK (FRCK) method. RCK applies on long-term continuous flow records with hourly resolution (or finer). It is based on the identification of turning points (rising and recessing points) on the flow time series and then connecting these points with straight lights. The definition of recessing point (end point of flow event) is based on the identification of points on the recession limb of hydrograph with stable recession coefficient (Blume, et al., 2007). FRCK is the filtered version of the RCK baseflow time series passed by the recursive digital filter (RDF). Form of the RDF are introduced in Eckhardt (2005). These two methods are represented in the RCK.p and FRCK.p Matlab codes in the package.

3.1.1. Revised Constant k Method

The use of the RCK.p code is demonstrated in line 16 of

```

4   %% Baseflow separation
5   % Convert the units to mm/h
6   Q=3.6*Q/A;
7   Q(Q==0)=NaN;
8
9   % Clustering of flow
10  q=Q(~isnan(Q));
11  idx=kmeans(q, 3);
12  mq=nan(3, 1);
13  for c=1:3
14      mq(c)=max(q(idx==c));
15  end
16  ben=min(mq, [], 1)';
17
18  % RCK Method
19  [Qbf_sl, pts, K, BFIm]=RCK_t(Q, 11, 5e-5, .9, ben, 1e-10);

```

Figure 2. Inputs for the RCK.p code are the continuous long-term flow time series (Q), half-length of regression window (L_{RW} , 11 hour), ratio of no-change (R_{NC} , $5 \times 10^{-5} \text{ h}^{-2}$), minimum

acceptable coefficient of determination (r^2_{MA} , 0.9), envelope baseflow rate (**ben**) and an indicator of positive. **Q** is in the form of vector and the rest of the inputs are scalar. It should be noted that values of **LRW**, **RNC** and r^2_{MA} maybe compatible only for the hourly resolution time series (the users are encourage to read MA2015 on the selections of parameter values). Line 6 converts the unit of **Q** into length/time (e.g. mm/h) consisting with the unit of precipitation. Note that in line 7 the time steps with zero flow rate are replaced by NaN and too much intermitted steps may have significant impacts on the functioning or result of the codes.

An update on the envelope baseflow rate is introduced in this new version of CPM compared to the original one in MA2015. The original CPM applies a relatively complex process (section 3.1.2 of MA2015) to construct **ben**; it needs the arbitrary specifications of an upper and a lower limit of the envelope baseflow rate. In this version, we used the k-mean clustering approach to calculate **ben** instead (line 10-16). The assumption here is that quick flow should have identifiable difference compared to baseflow. Instead of using two clusters, we use three to also isolate the transition flow rate. Therefore, the three clusters represent the peak flow, transition flow and baseflow. Since **ben** is the upper limit of baseflow, we select the maximum value from the baseflow cluster to represent this parameter (line 13-16).

The RCK method includes a procedure of identifying the time step with negative gradient of flow (step 3 of section 3.1.1 of MA2015). This is implemented by finding the positive $-dQ/dt$ determined on every sliding time window. The term “positive” is quantified as “ $>1 \times 10^{-10}$ ” instead of just “ >0 ” for the purpose of quality control. Outputs of RCK.p include the RCK baseflow time series (**Qbf_sl**, vector), the rising and recessing points extracted from the flow record stored in the structure array pts (**pts.RiPs** and **pts.RePs**) and the recession coefficient (**K**, scalar) of the basin.

```

4   %% Baseflow separation
5   % Convert the units to mm/h
6   Q=3.6*Q/A;
7   Q(Q==0)=NaN;
8
9   % Clustering of flow
10  q=Q(~isnan(Q));
11  idx=kmeans(q, 3);
12  mq=nan(3, 1);
13  for c=1:3
14      mq(c)=max(q(idx==c));
15  end
16  ben=min(mq, [], 1)';
17
18  % RCK Method
19  [Obf_sl, pts, K, BFIm]=RCK_t(Q, 11, 5e-5, .9, ben, 1e-10);

```

Figure 2. Demo of the RCK.p code.

3.1.2. Filtered Revised Constant k Method

The next step is to use the recursive digital filter to pass the **Qbf_sl** so as to form the FRCK baseflow time series **Qbf** (Figure 3). Inputs of the FRCK.p code are flow time series (**Q**), maximum baseflow index (**BFIm**) and recession coefficient (**K**). **BFIm** is initialized by **Qbf_sl** as indicated in line 19. This is different than the original CPM, which uses the UKIH method to calculate the initial **BFIm**. **BFIm** is updated at every iteration using the FRCK.p outputted baseflow time series (i.e. **Qbf**). Normally **BFIm** should converge after a few iterations. In the sample basin case, six times of iteration was found to be enough.

```

21  % FRCK Method
22  BFIM=[BFIm;0;0;0;0;0;0]; % Check whether it converge
23  for y=1:6
24      [Obf, BFIm]=FRCK(Q, Obf_sl, K);
25
26      BFIM(y+1)=BFIm;
27  end

```

Figure 3. Demo of the FRCK.p code.

3.2. *Event identification*

The event identification method consists of three components: formation of flow events, identification of rainfall episodes, and association of flow events with rainfall episodes. The formation of rainfall or flow events are based on the identification of event characteristic points (i.e. begin, peak and end point). The beginning and end points of flow events are searched within the population of rising and recession points, respectively. The peak flow points are identified automatically from the long-term flow record using the CPM_peak.p code. A peak flow is associated with its closest rising and recession point so as to form a flow event (refer to the CPM_FE.p code). Rainfall episodes are identified by the CPM_RE.p code following the method introduced in Mei et al. (2014). The outputted flow event is associated with the rainfall episodes that are “close enough” (Eq.(20) of MA2015) to form a rainfall-runoff event by the CPM.p code.

3.2.1. Flow Event

A flow event is defined by a beginning, end and a peak flow point (the number of peak flow points could be multiple, representing a multi-peak event). Peak flow points (**Pks**, vector) are automatically identified from the flow record by the CPM_peak.p code. Inputs of CPM_peak.p are the flow rate (**Q**), length of searching period (**LSP**), recession points (**RePs**) and baseflow time series (**Qbf**). **LSP** is calculated by the empirical relationship implemented in line 31 (Chow, et al., 1988).

Line 35 demonstrates the CPM_FE.p code. It integrates the peak flow points (**Pks**), the rising and recessing points (**pts.RiPs** and **pts.RePs**) in forming the flow events. Specifically, the code searches the closest rising and recessing point for each of the peak flow point to form a flow event. The outputted structure array **FE** contains two matrix and one cell vector. The **FE.SPEs** matrix stores the time of begin, peak and end of the single-peak flow events. The **FE.MPEs** matrix stores the begin and end time of the multi-peak flow events while the time of flow peaks are stored in the cell vector **FE.MPEP**.

```

29      %% Event identification
30      % Locate the peak flow rate from the long-term hydrograph
31 -    LSP=.827*24*A^.2;
32 -    Pks=CPM_peak(Q, LSP, pts.RePs, Qbf);
33
34      % Formation of flow events
35 -    FE=CPM_FE(Pks, pts.RiPs, pts.RePs);

```

Figure 4. Demo of the CPM_peak.p and CPM_FE.p code.

3.2.2. Rainfall Episode

Definition of rainfall episode is the continuous rainy hour (Mei, et al., 2014). “Rainy” is quantified as rainfall rate greater than 0 for the time steps while “continuous” means no interruption by 0 rain rate within any rainfall episode. Inputs of `CPM_RE.p` (line 38) are the time series of precipitation (**P**), zero-rain threshold (r_0 , 0) and length of maximum rainfall gap (**LMG**, 0). The outputted matrix **RE** records six rainfall properties namely the begin time, end time, duration, cumulative volume, maximum rainfall rate and centroid of the episodes. Compared to the original CPM method in MA2015, an extra step is included in line 40 to remove some minor rainfall episodes (i.e. rainfall episodes with duration less than 2 time step or intensity lower than 0.01 mm/h).

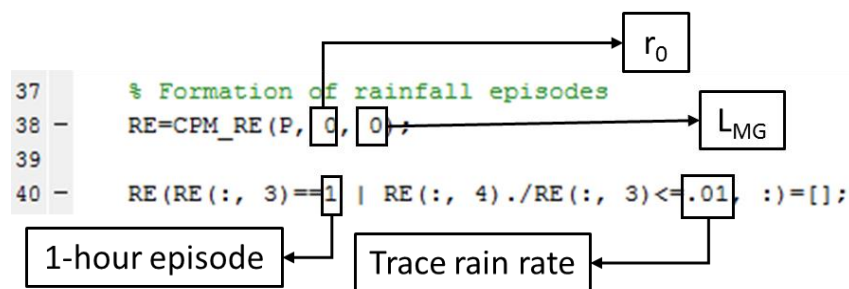


Figure 5. Demo of the `CPM_RE.p` code.

3.2.3. Event Association

The `CPM.p` code is used to associate the flow events with rainfall episodes that are within a searching radius. In most of the cases, there are more than one rainfall episode that are within the searching radius of a flow event. In these cases, all the eligible rainfall episodes are joined together to form a rainfall event. The required inputs are the time series of flow, precipitation and FRCK baseflow (**Q**, **P** and **Qbf**), timing of flow and rainfall event (stored in **FE** and **RE**), length of searching period (**LSP**) and mean time lag (**mlag**). It is noted that the information of mean time lag is not available at the first time; therefore we initialize mean time lag by length of searching period. Mean time lag is updated within each time of iteration (line 56-59) and four is the optimum iteration times required to attain a stable value. The unrealistic events (time lag is negative or/and

runoff coefficient is greater than 1) and event with the maximum time lag among others are excluded from the calculation of mean time lag for a basin (line 57-58).

Output of `CPM.p` is the properties of rainfall-runoff events (**RFE**, structure array). The event properties are stored in four matrices within **RFE**. The first one is **RFE.D** that stores the durations of rainfall and flow events and the corresponding time lag. The second one is **RFE.V** that contains the cumulative volume of rainfall, flow and baseflow of the rainfall-runoff events. The third matrix **RFE.tE** contains the timing of events (i.e. the begin time of rainfall and flow events, the end time of rainfall, flow events and the centroid of rainfall and flow events and the maximum peak flow point of the flow events). The last matrix is **RFE.R** that contains information of the event-based runoff coefficient and baseflow index. A small portion of flow events may be failed to associate with rainfall episodes and are removed in line 47-54.

```

42 % Event association
43 mlag=[LSP;0;0;0;0]; % Initialize mlag by LSP
44 for y=1:4
45     RFE=CPM(Q, P, Qbf, FE, RE, LSP, mlag(y));
46
47     k=find(isnan(RFE.tE(:, 1))); % Remove the non-associable
48     if ~isempty(k) % flood events
49         RFE.D(k, :)=[];
50         RFE.V(k, :)=[];
51         RFE.tE(k, :)=[];
52         RFE.R(k, :)=[];
53         RFE.GEs(k, :)=[];
54     end
55
56     lag=RFE.D(:, 3);
57     lag(RFE.D(:, 3)<0 | RFE.R(:, 1)>1 | ...
58         RFE.D(:, 3)==max(RFE.D(:, 3)))=[];
59     mlag(y+1)=mean(lag); % update mlag for every iteration
60 end

```

Figure 6. Demo of the `CPM_RFE.p` code.

3.3. Extra Filters

Extra filters (line 62-71) have been applied to refine the selected rainfall-runoff events after the hydrograph separation. The filters remove rainfall-runoff events with negative time lag (line 62), runoff coefficient greater than one (line 63), flow intensity lower than the mean of flow rate (line 64), response ratio lower than the reciprocal of **BFI_m** (line 65) and initial/final baseflow

index lower than **BFIm** (line 66). Among these five filters, the fifth filter is an extra filter included in the current version of CPM. All the parameters are concatenated in the **para** vector before saving (line 74). Results including the properties of rainfall-runoff events (**RFE**), the CPM parameters (**para**), time series of precipitation and flow (**P** and **Q**) are saved by the Results.mat Matlab variable in the user specified path (line 78).

```

61 %% Extra Filters
62 - k=find(RFE.D(:, 3)<0 | ...
63 - RFE.R(:, 1)>1 | ...
64 - RFE.V(:, 2)./RFE.D(:, 2)<nanmean(Q) | ...
65 - Q(RFE.tE(:, 7))./max(Q(RFE.tE(:, [2 4])), [], 2)<1/BFIm(length(BFIm)) | ...
66 - min(Qbf(RFE.tE(:, [2 4]))./Q(RFE.tE(:, [2 4])), [], 2)<BFIm(length(BFIm)));
67
68 - RFE.D(k, :)=[];
69 - RFE.V(k, :)=[];
70 - RFE.tE(k, :)=[];
71 - RFE.R(k, :)=[];
72
73 %% Concatenate the parameters
74 - para=[A K BFIm ben LSP mlag(length(mlag))];
75
76 %% Save data
77 - clear BFIm c FE idx k K lag LSP mlag mq Pks pts q Qbf_sl RE ben y
78 - save([path 'Results'],'RFE','P','para','Q','Qbf');

```

Figure 7. Demo of the code related to the extra filters.

4. Contact Us

The official version of the CPM source codes (by Matlab) is being developed and maintained in the Hydrometeorology and Hydrologic Remote Sensing group of Prof. Emmanouil N. Anagnostou in the Department of Civil and Environmental Engineering at the University of Connecticut (<http://ucwater.engr.uconn.edu/>). For the information about the most updated version of the CPM source codes and for technical support, please email Dr. Emmanouil N. Anagnostou (emmanouil.anagnostou@uconn.edu) or Dr. Yiwen Mei (yiwon.mei@uconn.edu).

References

- Blume, T., Zehe, E. & Bronstert, A., 2007. Rainfall—runoff response, event-based runoff coefficients and hydrograph separation. *Hydrol. Sci. J.*, 52(5), pp. 843-862.
- Chow, V. T., Maidment, D. R. & Mays, L. W., 1988. *Applied Hydrology*. International ed. s.l.:MCGRAW-HILL Higher Education.
- Eckhardt, K., 2005. How to construct recursive digital filters for baseflow separation. *Hydrol. Process.*, Volume 19, pp. 507-515.
- Mei, Y. & Anagnostou, E. N., 2015. A Hydrograph Separation Method Based on Information from Rainfall and Runoff Records. *J. Hydrol.*, Volume 523, pp. 636-649.
- Mei, Y., Anagnostou, E. N., Nikolopoulos, E. I. & Borga, M., 2014. Error Analysis of Satellite Rainfall Products in Mountainous Basins. *J. Hydrometeor.*, 15(5), pp. 1778-1793.