

Águas superficiais: hidrógrafa e separação de fluxo de base

Professor: Emílio Graciliano Ferreira Mercuri, D.Sc.
Departamento de Engenharia Ambiental - DEA,
Universidade Federal do Paraná - UFPR
emiliomercuri@gmail.com

Exercício 1

Em uma área de captação de 20 km² produziu-se o seguinte hidrograma de vazão na saída da bacia. Estimar o fluxo de base usando o método 1, calcular o volume de escoamento direto e a altura de escoamento direto (*runoff*).

Tempo do início da precipitação (h)	-6	0	6	12	18	24	30	36	42	48	54	60	66
Vazão observada (m ³ /s)	6	5	13	26	21	16	12	9	7	5	5	4.5	4.5

Solução: O hidrograma é traçado em escala (Fig. 1). Vê-se que o hidrograma de um evento de chuva tem um componente de escoamento de base. Para usar o método simples de linha reta de separação de fluxo de base, calculamos N :

$$N = 0.83 \times (20)^{0.2} = 1,51 \text{ dias} = 36,2 \text{ h} \approx 36 \text{ h} \quad (1)$$

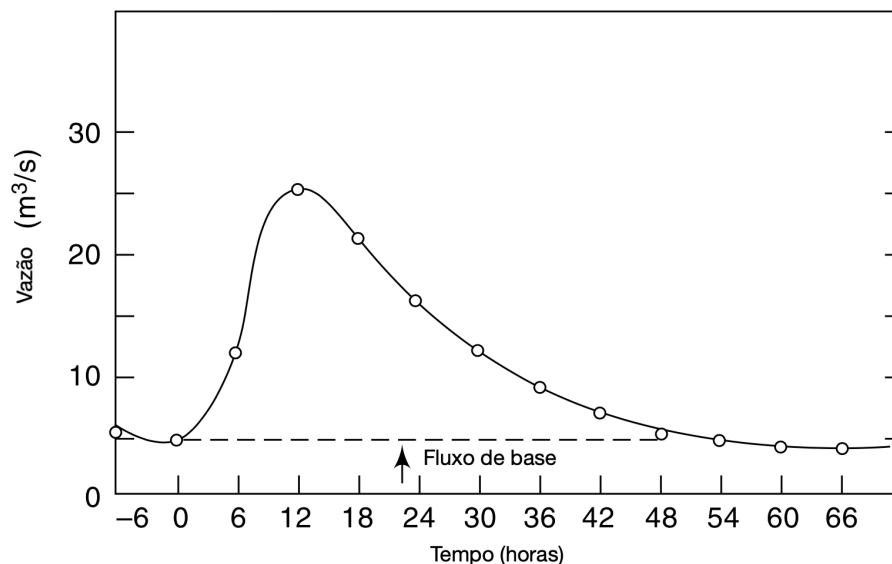


Figura 1: Exemplo de separação do fluxo de base

Como $N = 36\text{h}$ e o tempo até o pico é 12h, assume-se que o escoamento direto existe de $t = 0$ a 48 h. Uma separação de fluxo de base em linha reta fornece um valor constante de 5 m³/s para o fluxo de base.

Para o cálculo da área do escoamento direto será usada integração numérica. O método adotado é a regra trapezoidal, conforme ilustra a Figura 2 e a Equação 2.

$$\int_a^b f(x)dx \approx (b - a) \left(\frac{f(a) + f(b)}{2} \right) \quad (2)$$

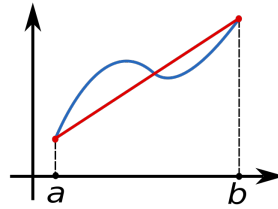


Figura 2: Ilustração da regra trapezoidal

Para intervalos constantes de 6 horas como é o nosso caso, a equação para a integração pela regra trapezoidal é:

$$\int_a^b f(x)dx \approx \frac{\Delta x}{2} (f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + 2f(x_4) + \cdots + 2f(x_{N-1}) + f(x_N)) \quad (3)$$

Porém, neste caso:

- $f(x_0) = 0$ pois o escoamento direto em $t = 0$ h é zero.
- $f(x_N) = 0$ pois o escoamento direto em $t = 48$ h é zero.

E a equação fica:

$$\int_a^b f(x)dx \approx \Delta x (f(x_1) + f(x_2) + f(x_3) + f(x_4) + \cdots + f(x_{N-1})) \quad (4)$$

Ou em função do tempo:

$$\int_a^b f(t)dt \approx \Delta t (f(t_1) + f(t_2) + f(t_3) + f(t_4) + \cdots + f(t_{N-1})) \quad (5)$$

O nosso intervalo de tempo é $\Delta t = 6$ horas, mas como a vazão tem unidade m^3/s temos que converter o intervalo de tempo para segundo, ou seja: $\Delta t = 6 \times 60 \times 60$ horas

Subtraindo a vazão observada do escoamento de base, temos o escoamento direto:

Tempo do início da precipitação (h)	-6	0	6	12	18	24	30	36	42	48	54	60	66
Vazão observada (m^3/s)	6	5	13	26	21	16	12	9	7	5	5	4.5	4.5
Escoamento direto (m^3/s)	-	0	8	21	16	11	7	4	2	0	-	-	-

O volume do escoamento direto é obtido pela integração do escoamento direto na hidrógrafa.

$$\begin{aligned} \text{Volume do escoamento direto} &= (6 \times 60 \times 60) \frac{1}{2} (8 + 21 + 16 + 11 + 7 + 4 + 2) \\ &= 3600 \times 6 \times (8 + 21 + 16 + 11 + 7 + 4 + 2) = 1,4904 \times 10^6 \text{ m}^3 \end{aligned} \quad (6)$$

$$\text{Altura de Runoff} = \frac{\text{volume de escoamento direto}}{\text{area da bacia}} = \frac{1,4904 \times 10^6}{20 \times 10^6} = 0,07452 \text{ m} = 7,45 \text{ cm} \quad (7)$$

Exercício 2 (para casa)

Em uma área de captação de 1 km² produziu-se o seguinte hidrograma de vazão na saída da bacia. Estimar o fluxo de base usando o método 1, calcular o volume de escoamento direto e a altura de escoamento direto (*runoff*).

Tempo do início da precipitação (h)	-6	0	6	12	18	24	30	36	42	48	54	60	66
Vazão observada (m ³ /s)	6	5	13	26	21	16	12	9	7	5	5	4.5	4.5

Solução: Para usar o método simples de linha reta de separação de fluxo de base, calculamos N :

$$N = 0.83 \times (1)^{0.2} = 0,83 \text{ dias} = 19,92 \text{ h} \quad (8)$$

Vamos considerar $N = 18$ horas para seguir os intervalos com dados de vazão. Portanto, o *runoff* começa em $t = 0$, tem o pico em $t = 12$ h e termina em $t = 12 + 18 = 30$ h.

O hidrograma é traçado em escala (Fig. 3). Vê-se que o hidrograma de um evento de chuva tem um componente de escoamento de base.

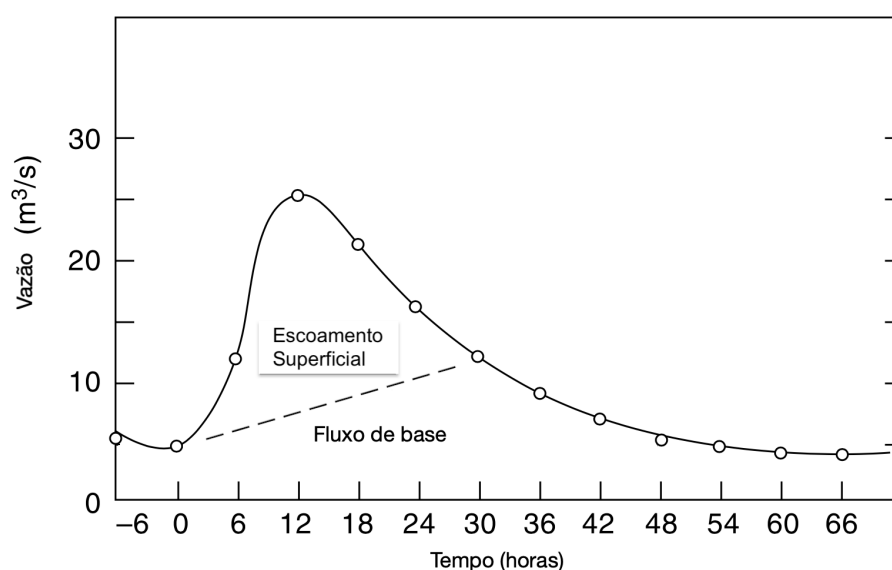


Figura 3: Exemplo de separação do fluxo de base

Dica para continuar a solução: encontrar a equação da reta que separa o escoamento direto do escoamento de base.

Leituras complementares

Sugere-se a leitura dos artigos:

- Xie, J., Liu, X., Wang, K., Yang, T., Liang, K., & Liu, C. (2020). Evaluation of typical methods for baseflow separation in the contiguous United States. *Journal of Hydrology*, 583, 124628.
- Stewart, M. K. (2015). Promising new baseflow separation and recession analysis methods applied to streamflow at Glendhu Catchment, New Zealand. *Hydrology and Earth System Sciences*, 19(6), 2587-2603.
- Hewlett, John D., and Alden R. Hibbert. Factors affecting the response of small watersheds to precipitation in humid areas. *Forest hydrology* 1 (1967): 275-290.
- Siefert, Cesar Augusto Crovador, and Irani dos Santos. Mecanismos de geração de escoamento e áreas hidrologicamente sensíveis: uma abordagem hidrogeomorfológica para delimitação de áreas de preservação permanente. *Raega-O Espaço Geográfico em Análise* 24 (2012).

Bibliografia

- Chow, Ven Te, David R. Maidment, and W. Larry. Applied Hydrology. MacGraw-Hill (1988).
- Subramanya, K. Engineering hydrology, 4e. Tata McGraw-Hill Education, 2013.
- Stewart, M. K. (2015). Promising new baseflow separation and recession analysis methods applied to streamflow at Glendhu Catchment, New Zealand. Hydrology and Earth System Sciences, 19(6), 2587-2603.
- Xie, J., Liu, X., Wang, K., Yang, T., Liang, K., & Liu, C. (2020). Evaluation of typical methods for baseflow separation in the contiguous United States. Journal of Hydrology, 583, 124628.
- Hewlett, John D., and Alden R. Hibbert. Factors affecting the response of small watersheds to precipitation in humid areas. Forest hydrology 1 (1967): 275-290.
- Siefert, Cesar Augusto Crovador, and Irani dos Santos. Mecanismos de geração de escoamento e áreas hidrologicamente sensíveis: uma abordagem hidrogeomorfológica para delimitação de áreas de preservação permanente. Raega-O Espaço Geográfico em Análise 24 (2012).

Anexo 1 - Código para separação do fluxo de base

A função *sepBaseflow* foi retirada da Biblioteca *Python* chamada *Hydrograph-py*. Essa função é uma aplicação do Método de Hewlett e Hibbert.

```
1 import pandas as pd
2 import numpy as np
3
4 def sepBaseflow(x, dt, A, k=0.000546, dt_max=None, tp_min=None):
5     '''
6     Separate a time-series into baseflow and peakflow. Fills missing flow records by
7     interpolation.
8
9     Input:
10         x:      Pandas dataframe with Index being a pandas datetime index and '
11                 Date' label. Dataframe should
12                 contain one column for flow data, and should be labeled 'Total
13                 runoff [m3 s-1]'.
14         dt:      Minimum time-step interval (in minutes) for analysing the data.
15                 Minute choices are 5, 15, or 60.
16         k:      Slope of the dividing line; i.e. slope that defines when peakflow
17                 event starts and baseflow separation occurs.
18                 Default is 0.000546 m3 s-1 km-2 h-1 (Hewlett and Hibbert
19                 1967).
20         A:      Catchment area in km2 upstream of point of interest.
21         dt_max:  Only interpolate over maximum number of consecutive NaN defined
22                 over time period dt_max in hours.
23         tp_min:  Minimum duration of runoff peak in hours to be selected as being
24                 a peak.
25
26     Returns:
27         df_final: Pandas dataframe with datetime index and the following columns:
28                 dt [hour]:      Time difference in hours
29                 between two records.
30                 Total runoff [m3 s-1]: Recorded flow in cumecs for
31                 that timestamp.
32                 Total runoff interp. [m3 s-1]: Interpolated recorded flow
33                 in cumecs.
34                 Baseflow [m3 s-1]: Calculated baseflow in
35                 cumecs for that timestamp.
36                 Peakflow [m3 s-1]: Calculated peakflow in
37                 cumecs for that timestamp.
38                 Peak nr.:      Peak number in sequence.
39                 Each peakflow event (i.e. flow above baseflow) is given a
40                 unique number
41
42                 if it classifies as being a
43                 peak after filtering.
44                 Peakflow starts:      Timestamp when peakflow
45                 starts (moment when runoff peak exceeds baseflow).
46                 Peakflow ends:      Timestamp when peakflow ends
47                 (moment when runoff peak intersects again with baseflow).
48                 Flow volume [m3]: Volume of the flow between
49                 two time-steps (total volume; i.e. baseflow + peakflow).
50                 Max flow [m3 s-1]: Maximum flow of peak flow
51                 event.
52                 Date max. flow:      Timestamp of maximum flow of
53                 peak flow event.
54                 Tp [hour]:      Time to peak.
55
56     '''
57
58     minDate = x.index.min()
59     maxDate = x.index.max()
60     #-date range for full period (set it depending on the defined time interval)
```

```
39     if dt == 5:
40         print('Processing using a 5-minute interval...')
41         dr = pd.date_range(minDate, maxDate, freq='5T')    #-5-minute interval
42     elif dt == 15:
43         print('Processing using a 15-minute interval...')
44         dr = pd.date_range(minDate, maxDate, freq='15T')    #-15-minute interval
45     else:
46         print('Processing using a 60-minute interval...')
47         dr = pd.date_range(minDate, maxDate, freq='60T')    #-60-minute interval
48
49     df_final = pd.DataFrame(dr, columns=['Date']); dr = None
50     df_final['Time_diff'] = df_final['Date'].diff()
51     df_final['dt [hour]'] = df_final['Time_diff'].dt.seconds / 3600.0
52     df_final['dt [hour]'] = df_final['dt [hour]'].fillna(0)
53     df_final.drop('Time_diff', axis=1, inplace=True)
54     df_final.set_index('Date', inplace=True)
55
56     df_final['Total runoff [m3 s-1'] = x; x = None;
57     #-only interpolate maximum number of consecutive NaNs. dt_max is in hours, so to
58     calculate nr of timesteps depending on the set time-interval
59     if dt_max:
60         if dt == 5:
61             df_final['Total runoff interp. [m3 s-1'] = df_final['Total runoff [m3
62             s-1'].interpolate(method='time', limit=dt_max*12)
63         elif dt == 15:
64             df_final['Total runoff interp. [m3 s-1'] = df_final['Total runoff [m3
65             s-1'].interpolate(method='time', limit=dt_max*4)
66         else:
67             df_final['Total runoff interp. [m3 s-1'] = df_final['Total runoff [m3
68             s-1'].interpolate(method='time', limit=dt_max)
69     else:
70         df_final['Total runoff interp. [m3 s-1'] = df_final['Total runoff [m3 s
71         ^-1]'].interpolate(method='time')
72     df_final['Baseflow [m3 s-1'] = np.nan
73     df_final['Peakflow [m3 s-1'] = np.nan
74
75     cnt=0
76     flag = True    #-flag to define new baseflow threshold
77     t=0
78     Qthresh = False
79     for i in df_final.iterrows():
80         dindex = i[0]
81         print(dindex)
82         Qtot = i[1]['Total runoff interp. [m3 s-1']
83         QBase = Qtot
84
85         #-For first record, baseflow equals total runoff
86         if cnt==0:
87             df_final.loc[df_final.index==dindex, 'Baseflow [m3 s-1'] = Qtot
88         else:
89             #-Total runoff of t-1
90             QtotOld = df_final.iloc[cnt-1,2]
91             dt = df_final.iloc[cnt,0]
92
93             #-Check whether increase in streamflow between two time-steps is larger
94             than k * dt * A, and thus indicates the start of the rising limb
95             if (Qtot > (QtotOld + (k * dt * A))) and flag:
96                 Qthresh = QtotOld
97                 flag = False
98                 t = 0
99
100             #-Linearly calculate baseflow using time difference and threshold
101             if Qthresh:
102                 t+=dt
103                 QBase = Qthresh + (k * t * A)
104             #-Check if recession limb is below the baseflow curve
```

```
98         if QBase>Qtot:
99             Qthresh = False
100             flag = True
101             #Make sure baseflow does not exceed total runoff at any point in time
102             QBase = min(QBase, Qtot)
103
104             #Fill in the final dataframe
105             df_final.loc[df_final.index==dindex, 'Baseflow [m3 s-1]' ] = QBase
106
107             cnt+=1
108
109         df_final['Baseflow [m3 s-1]' ] = df_final[['Baseflow [m3 s-1]', 'Total runoff
110             [m3 s-1]' ]].min(axis=1)
111         #df_final = df_final.astype(np.float)
112
113         df_final['Peakflow [m3 s-1]' ] = df_final['Total runoff interp. [m3 s-1]' ] -
114             df_final['Baseflow [m3 s-1]' ]
115
116         #Now filter the peaks and assign peak numbers
117         df_final = filterpeaks(df_final, tp_min)
118
119         print('Calculating event values...')
120         df_final.reset_index(inplace=True)
121         #Start of peakflow event
122         df = df_final[['Date', 'Peak nr.']].groupby('Peak nr.').min()
123         df.rename(columns={'Date': 'Peakflow starts'}, inplace=True)
124         df.reset_index(inplace=True)
125         df_final = pd.merge(df_final, df, how='left', on='Peak nr.')
126         #End of peakflow event
127         df = df_final[['Date', 'Peak nr.']].groupby('Peak nr.').max()
128         df.rename(columns={'Date': 'Peakflow ends'}, inplace=True)
129         df.reset_index(inplace=True)
130         df_final = pd.merge(df_final, df, how='left', on='Peak nr.')
131         #Flow volume
132         df_final['Flow volume [m3]' ] = df_final['Total runoff interp. [m3 s-1]' ] *
133             3600 * df_final['dt [hour]' ]
134         df_final.loc[pd.isna(df_final['Peak nr.']), 'Flow volume [m3]' ] = np.nan
135         #Max flow and time of max flow
136         df_final['Max. flow [m3 s-1]' ] = np.nan
137         df_final['Date max. flow' ] = np.nan
138         df_final['Tp [hour]' ] = np.nan
139         for i in pd.unique(df_final['Peak nr.']):
140             df_short = df_final.loc[df_final['Peak nr.']==i, ['Date', 'Total runoff
141                 interp. [m3 s-1]' ]#,'Max. flow [m3 s-1]' ]]]
142             mflow = df_short['Total runoff interp. [m3 s-1]' ].max()
143             df_short = df_short.loc[df_short['Total runoff interp. [m3 s-1]' ]==mflow]
144             mdate = df_short['Date' ].min()
145             df_final.loc[df_final['Peak nr.']==i, 'Max. flow [m3 s-1]' ] = mflow
146             df_final.loc[df_final['Peak nr.']==i, 'Date max. flow' ] = mdate
147         df_final['Date max. flow' ] = pd.to_datetime(df_final['Date max. flow' ])
148         df_final['Tp [hour]' ] = (df_final['Date max. flow' ] - df_final['Peakflow starts
149             ']).dt.seconds / 3600
150         df_final.set_index('Date', inplace=True)
151
152         print('Processing completed successfully.')
153
154         return df_final
```