Storm Runoff Model Equations

1 Rainfall Generation Model

The rainfall intensity model uses a scaled Gaussian distribution:

$$R(t) = P \cdot \frac{e^{-\frac{(t-C)^2}{2W^2}}}{\max\left(e^{-\frac{(t-C)^2}{2W^2}}\right)} \cdot P \tag{1}$$

Where:

- $R(t)$ is the rainfall intensity at time t [mm/h]
- P is the peak rainfall intensity [mm/h]
- C is the center time (when peak rainfall occurs) [hours]
- W is the width parameter (controls spread) [hours]
- t is the time variable ranging from 0 to the duration [hours]

This can be simplified to:

$$R(t) = P \cdot \frac{e^{-\frac{(t-C)^2}{2W^2}}}{e^{-\frac{(C-C)^2}{2W^2}}} = P \cdot e^{-\frac{(t-C)^2}{2W^2}} \tag{2}$$

Since the maximum value of the Gaussian function occurs at $t = C$, where the exponential term equals 1.

2 Nonlinear Runoff Model

The nonlinear runoff model is based on a discrete time implementation of a nonlinear reservoir equation:

$$Q_t = Q_{t-1} + \frac{\Delta t}{k_{nl}} \cdot \max(R_{t-1} - Q_{t-1}^n, 0) \tag{3}$$

Where:

- Q_t is the runoff at time step t [mm/h]
- R_t is the rainfall intensity at time step t [mm/h]
- k_{nl} is the reservoir response time parameter [hours]
- n is the nonlinearity exponent [-]
- Δt is the time step [hours]

The model ensures that runoff is always non-negative by applying:

$$Q_t = \max(Q_t, 0) \tag{4}$$

This represents a nonlinear reservoir where:

1. The reservoir's outflow rate is proportional to its storage raised to power n
2. The inflow rate equals the rainfall intensity
3. The parameter k_{nl} controls how quickly the reservoir responds to rainfall input

3 Implementation in Code

In the Python implementation, these equations are realized as:

```
# Rainfall generation
time = np.linspace(0, duration, 100)
rainfall = peak * norm.pdf(time, center, width)
rainfall = rainfall / np.max(rainfall) * peak  # Ensure peak is correct

# Nonlinear runoff computation
Q_nl = np.zeros_like(rainfall)
dt = time[1] - time[0]  # time step
for t in range(1, len(rainfall)):
    term = max(rainfall[t - 1] - Q_nl[t - 1], 0)
    Q_nl[t] = Q_nl[t - 1] + (dt / k_nl) * term
    if Q_nl[t] > 0:
        Q_nl[t] = Q_nl[t - 1] + (dt / k_nl) * (rainfall[t - 1] - Q_nl[t - 1] ** n)
    else:
        Q_nl[t] = 0
```