Simplified Microphysics

- **water vapor**
- **cloud droplets**
- **rain drops**

- **condensation**
- **evaporation**
- **collection**

**Fall out (precipitation)**

- \( = 0 \) (reversible)
- \( = \infty \) (irreversible)
Simplified Microphysics

\[ \frac{d\theta}{dt} = \frac{L}{c_p \bar{\Pi}} C \]

\[ \frac{dw}{dt} = -C + E_r \]

\[ \frac{dl}{dt} = C - A_r \]

\[ \frac{dr}{dt} = P_r + A_r - E_r \]

\[ P_r = 0 \text{ (reversible)} \]

\[ P_r = \infty \text{ (irreversible)} \]
More Simplified Microphysics

\[ \frac{d\theta}{dt} = \frac{L}{c_p \bar{\pi}} C \]
\[ \frac{dw}{dt} = -C \]
\[ \frac{dl}{dt} = C - A_r \]

\[ A_r = 0 \text{ (reversible)} \]
\[ A_r = \infty \text{ (irreversible)} \]
Saturation Adjustment Algorithm

\[ p^{n+1} \quad \theta^* \quad \theta^{n+1} \quad w^{n+1} \quad w^* \]

\[ p^n \quad \Delta \theta \quad \Delta w \]

Saturation mixing ratio

Saturation adiabat

Dry adiabat
Saturation Adjustment Algorithm

1. **Adiabatic.** No phase changes involving cloud droplets ($C=0$):
   \[ \theta^n, w^n \rightarrow \theta^*, w^* \]

2. **Isobaric.** Only phase changes involving cloud droplets operate ($|C| > 0$):
   \[ \theta^*, w^* \rightarrow \theta^{n+1}, w^{n+1} \]
Saturation Adjustment Algorithm

1. **Adiabatic.** No phase changes involving cloud droplets (\(C=0\)):
   \[ \theta^n, w^n \rightarrow \theta^*, w^* \]
   \( (\theta^*, w^*) = (\theta^n, w^n) \)

2. **Isobaric.** Only phase changes involving cloud droplets operate (\(|C| > 0\)):
   \[ \theta^*, w^* \rightarrow \theta^{n+1}, w^{n+1} \]
   \[ \theta^{n+1}, w^{n+1} = (\theta^* + \Delta \theta, w^* + \Delta w) \]
2. **Isobaric.** Only phase changes involving cloud droplets operate \(|C| > 0\):

\[ \theta^{n+1}, w^{n+1} = (\theta^* + \Delta \theta, w^* + \Delta w) \]

Conservation of energy (First Law of Thermodynamics):

\[ 0 = c_p \pi \Delta \theta + L \Delta w \text{ or } 0 = c_p \Delta T + L \Delta w \]

Adjusted state is exactly saturated:

\[ w^{n+1} = w_s(\pi \theta^{n+1}, p^{n+1}) \text{ or } w^{n+1} = w_s(T^{n+1}, p^{n+1}) \]
Saturation Adjustment Algorithm

\[ 0 = c_p \pi \Delta \theta + L \Delta w \]

\[ w^{n+1} = w_s(\pi \theta^{n+1}, p^{n+1}) \]

which can be written as

\[ 0 = c_p \pi (\theta^{n+1} - \theta^*) + L (w^{n+1} - w^*) \]

\[ w^{n+1} = w_s(\pi \theta^{n+1}, p^{n+1}) \]
Saturation Adjustment Algorithm

\[ \theta^{n+1} + \gamma w^{n+1} = \theta^* + \gamma w^* \]  \hspace{1cm} (5)

where \( \gamma \equiv L/(c_p \bar{\pi}) \).

conservation of suspended water mixing ratio (vapor and cloud droplets),

\[ w^{n+1} + l^{n+1} = w^* + l^* \]  \hspace{1cm} (6)

We first assume that the air will be exactly saturated after adjustment, so that

\[ w^{n+1} = w_s(T^{n+1}, p^{n+1}) \]  \hspace{1cm} (7)

where \( w_s(T, p) \) is the saturation mixing ratio,

\[ w_s(T, p) = 0.622 \frac{e_s(T)}{p - e_s(T)} \]  \hspace{1cm} (8)

and \( e_s(T) \) is the saturation vapor pressure. One may use Bolton’s (1980) formula for \( e_s(T) \):

\[ e_s(T) = 6.112 \exp \left( \frac{17.67T_c}{T_c + 243.5} \right) \]  \hspace{1cm} (9)

where \( e_s \) is in mb, \( T_c = T - T_0 \), and \( T_0 = 273.15 \) K.
Saturation Adjustment Algorithm

Equation (7) closes the set (5), (6), and (7). However, this set must be solved iteratively because \( w_s \) is a non-linear function of \( T \). To obtain a direct (non-iterative) solution, expand \( w_s \) in a Taylor series in \( T \) about \( w_s(T^*, p^{n+1}) \) and neglect all terms of second and higher order:

\[
w^{n+1} \approx w_s(T^*, p^{n+1}) + \left( \frac{\partial w_s}{\partial T} \right)_{T=T^*, p=p^{n+1}} (T^{n+1} - T^*).
\]

The set (5), (6), and (10) can now be solved algebraically for \( \theta^{n+1} \), \( w^{n+1} \), and \( l^{n+1} \).
Saturation Adjustment Algorithm

To solve the set, we first write (10) in terms of $\theta$ instead of $T$:

$$w^{n+1} = w^*_s + \alpha^*(\theta^{n+1} - \theta^*),$$

where $w^*_s \equiv w_s(T^*, p^{n+1})$, $\alpha^* \equiv \alpha(T^*, p^{n+1})$, and

$$\alpha(T, p) \equiv 0.622 \frac{\pi p}{(p - e_s(T'))^2} \left(\frac{de_s}{dT}\right)_{T}. \tag{12}$$

For $de_s/dT$, one may use the Clausius-Clapeyron equation:

$$\frac{de_s}{dT} = \frac{Le_s}{R_v T^2}, \tag{13}$$

where $L = 2.5 \times 10^6$ J/kg and $R_v = 461.5$ J/(kg K).

Now use (11) in (5) to eliminate $w^{n+1}$. Then solve for $\theta^{n+1}$:

$$\theta^{n+1} = \theta^* + \frac{\gamma}{1 + \gamma \alpha^*}(w^* - w^*_s). \tag{14}$$

Once $\theta^{n+1}$ is known from (14), we can immediately obtain $w^{n+1}$ from (11), and $l^{n+1}$ from (6).
Saturation Adjustment Algorithm

If $w^{n+1} \leq w^* + l^*$, then (6) implies that $l^{n+1} \geq 0$. If $w^{n+1} > w^* + l^*$, (6) implies that $l^{n+1} < 0$, which means that our assumption of saturation is incorrect. Therefore,

$$l^{n+1} = 0$$

replaces (10). Then (5) and (6) become

$$w^{n+1} = w^* + l^*,$$

$$\theta^{n+1} = \theta^* - \gamma(w^{n+1} - w^*).$$