0.1 Rainfall-to-Moisture Conversion Model

In our model, each grid cell is assigned a rainfall amount, and we convert this rainfall into a fuel moisture content (FMC) using a simple, literature-based linear approximation. The purpose of this section is to define the conversion and document its empirical justification.

Model formulation

Let P_{ij} denote the rainfall amount (in millimetres) applied to grid cell (i, j). We assume an initial fuel moisture content M_0, and compute the updated moisture value as

$$M_{ij} = \min\bigl(M_{\max},\ M_0 + s_{\mathrm{mm}}\, P_{ij}\bigr),$$

where

- $M_0 = 15.4\%$ is the average pretreatment moisture content reported by Mohr and Waldrop (2008),
- $s_{\mathrm{mm}} \approx 0.14\%/\mathrm{mm}$ is the experimentally derived slope describing the gain in moisture per millimetre of rainfall,
- $M_{\max}$ is a prescribed upper limit representing the maximum achievable moisture in this simplified model.

Empirical justification

Our linear coefficient s_{mm} is not arbitrarily chosen. It is derived directly from the rainfall–moisture results reported in Mohr and Waldrop (2008), who applied controlled rainfall treatments to 1000-hour woody fuels.

Their study reports:

- initial moisture $M_0 = 15.4\%$,
- four separate 1-inch rainfall events over one week (totaling $4'' = 101.6\,\mathrm{mm}$) increased moisture to approximately 32%,
- two 2-inch rainfall events over one week (also totaling $4''$) increased moisture to approximately 27.1%.

The corresponding moisture increases are

$$\Delta M_1 = 32 - 15.4 = 16.6\%, \qquad \Delta M_2 = 27.1 - 15.4 = 11.7\%.$$

Dividing by the total rainfall in each treatment gives slopes

$$s_1 = \frac{16.6}{101.6} \approx 0.163\%/\mathrm{mm}, \qquad s_2 = \frac{11.7}{101.6} \approx 0.115\%/\mathrm{mm}.$$

We therefore adopt their average

$$s_{\mathrm{mm}} = \frac{s_1 + s_2}{2} \approx 0.14\%/\mathrm{mm}.$$

Python implementation

The following Python-style pseudocode illustrates how this conversion can be applied in a grid-based simulation:

```
M0 = 15.4         # Initial moisture (%), from Mohr and Waldrop (2008)
s_mm = 0.14       # Moisture gain per mm of rain (%/mm)
M_max = 50.0      # Example upper bound (%), user-defined

for each cell (i, j):
    P_ij = random_rain_mm()       # Random rainfall amount in mm
    M_ij = M0 + s_mm * P_ij
    if M_ij > M_max:
        M_ij = M_max
```

This produces a moisture field M_{ij} suitable for use in subsequent fire spread modelling.

0.2 Probabilistic fire spread as a function of fuel moisture

In the deterministic version of our cellular automaton, cells with fuel moisture exceeding the extinction moisture M_{ext} simply cannot ignite. To incorporate a more realistic treatment of fire behaviour, we now introduce a probabilistic ignition rule in which the probability of ignition decreases smoothly as fuel moisture increases.

Literature motivation

Laboratory ignition experiments (e.g. Anderson, 1969) show that fine dead fuels have very high ignition probabilities when moisture contents are below about 10%, while ignition probability drops rapidly as moisture approaches the extinction moisture (typically 20–25% for Australian eucalypt litter). Catchpole and Catchpole (1991) proposed a widely used *moisture damping function* of the form

$$\eta_M = \left(1 - \frac{M}{M_{\text{ext}}}\right)^2,$$

which appears in several rate-of-spread and ignition probability models. This function smoothly decreases from 1 (completely dry) to 0 (at the extinction moisture), and is consistent with Australian operational fire behaviour observations.

Ignition probability model

We define the ignition probability for cell (i, j) with fuel moisture M_{ij} as

$$p_{\text{ignite}}(M_{ij}) = \begin{cases} p_0 \left(1 - \dfrac{M_{ij}}{M_{\text{ext}}}\right)^2, & M_{ij} \le M_{\text{ext}}, \\ 0, & M_{ij} > M_{\text{ext}}, \end{cases}$$

where p_0 is the maximum ignition probability (e.g. $p_0 = 1$) and we take $M_{\text{ext}} = 25\%$ in accordance with Australian studies of eucalypt litter fuels.

Probabilistic spread rule

A cell that is currently unburned at time t ignites at time $t+1$ with probability $p_{\text{ignite}}(M_{ij})$ provided that at least one neighbour is burning. Otherwise it remains unburned. Formally,

$$\mathbb{P}\big(S_{ij}^{t+1} = \text{burning}\big) = p_{\text{ignite}}(M_{ij})\,\mathbf{1}\big[\exists(k,\ell) \in \mathcal{N}(i,j) : S_{k\ell}^{t} = \text{burning}\big]\,.$$

Python-style implementation

```
M_ext = 25.0       # Extinction moisture (%) for eucalypt litter
p0    = 1.0        # Maximum ignition probability

def p_ignite(M):
    if M > M_ext:
        return 0.0
    x = 1 - M / M_ext
    return p0 * (x * x)

for each cell (i, j):
    if state[i, j] == 'unburned':
        if any_neighbour_burning(i, j):
            prob = p_ignite(M[i, j])
            if random() < prob:
                state_next[i, j] = 'burning'
            else:
                state_next[i, j] = 'unburned'
        else:
            state_next[i, j] = 'unburned'
    elif state[i, j] == 'burning':
        state_next[i, j] = 'burned'
    else:
        state_next[i, j] = 'burned'
```

This probabilistic formulation allows partially wet areas to impede—but not necessarily completely block—fire spread, producing more realistic and spatially variable fire behaviour driven by the moisture field.