The Firebrand Spotting Parameterization in the WRF Model

The Firebrand Spotting parameterization was developed for the WRF-Fire component of the WRF model versions starting at 4.0.1. The parameterization couples to WRF-Fire and uses a Lagrangian particle transport framework to advect firebrands in the innermost nest of the domain.

The parameterization identifies areas at risk of fire spotting by modeling transport and physical processes of individual firebrands after fires have been ignited in the WRF-Fire model. The firebrand processes contained in the parameterization are Generation, Transport, Physics, and Landing

Firebrands are cyclically generated from sources along the fire front where fuel mass consumption is high. Distributed over multiple vertical levels, firebrands are transported with the atmospheric flow and burnout as advected. Firebrands may burnout entirely or land once trajectories descend below a given height threshold.

Particles that land before complete burnout are accumulated in a 2-D field during regular intervals. The field obtained from the spatial accumulation is used to calculate the likelihood of new fire ignitions due to spotting. The spotting likelihood is computed using the ratio of landed firebrands per gridpoint to the total number of landed particles within the corresponding time interval between consecutive model outputs. The ratios are then scaled by a function of dry over wet fuel mass at the corresponding gridpoints.

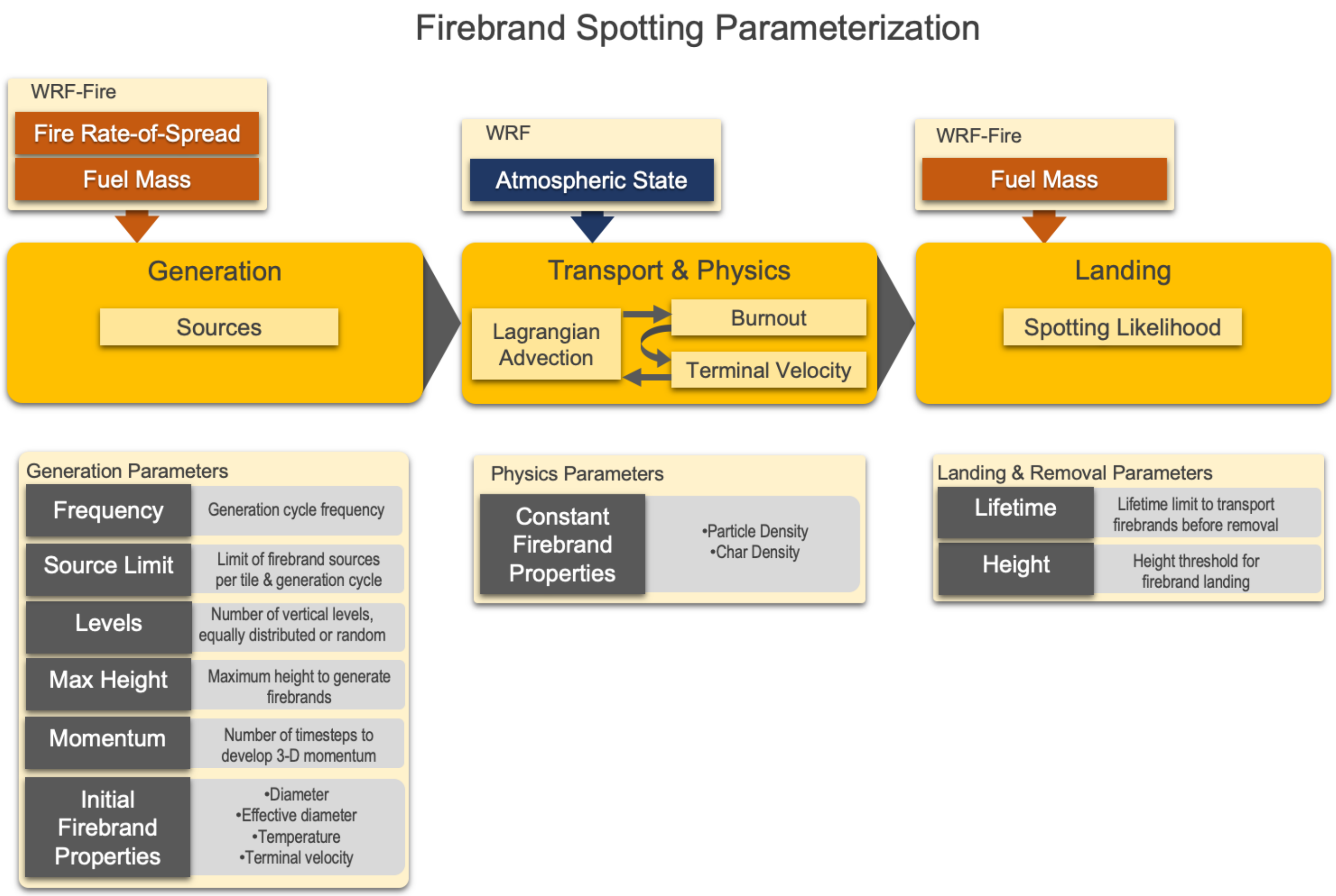

Fig 0: The Firebrand Spotting parameterization in the WRF-Fire model.

Code description

Calculations made by the parameterization do not update model variables and the parameterization only runs in the model inner nested domain.

The code comprises two independent modules, one with the physical processes (name) and another with the necessary MPI wrapping routines (name) that were not yet part of the WRF source code. The motivation to separate the MPI routines in an independent module was to enable them to be used in other model parameterizations through a USE statement without importing the firebrand spotting component.

The Firebrand Spotting variables are part of Registry.fire and the parameterization is called from start_em.F and solve_em.F, after all of the physics parameterizations and relevant halos are completed.

Firebrands are cyclically generated from sources along the fire front where fuel mass consumption is high. The frequency of generation cycles is defined by the generation frequency parameter and generation sources comprise the 2-D gridpoints actively burning in the given timestep, with the highest fuel mass consumed by fire between the given and previous generation cycles. The number of generation sources is defined by the source limit parameters, while the vertical levels per source is set by the levels parameters. Once the 3-D release position and initial properties of individual particles are established, firebrands are allowed to acquire momentum before particles begin to burnout.

During transport, firebrands follow the atmospheric flow, burnout, and descend. Particle density and char density are the only user defined parameters used for firebrand physics. Firebrands may burnout entirely during transport, reach a maximum allowed lifetime, or land, as specified by the firebrand landing parameters.

Particles that land are accumulated in a 2-D field (**fs_count_landed_all**). The spotting likelihood is computed using particles that land on an entirely unburnt gridpoint (based on the atmospheric grid mesh) in the interval between outputs (**fs_count_landed_hist**) to the total number of landed particles in the corresponding accumulation period (**fs_frac_landed**). The ratios are then scaled by a function of dry over wet fuel mass at the corresponding gridpoints (**fs_spotting_lkhd**).

<u>Output variables</u>

Outputs produced by the Firebrand Spotting Parameterization are 2-D and on the atmospheric grid mesh, unless otherwise specified.

- Products
    - **fs_count_landed_all**: accumulated firebrand landings during the simulation period
    - **fs_count_landed_hist**: firebrand landings on unburnt gridpoints (based on the atmospheric grid mesh) in the interval between history outputs
    - **fs_frac_landed**: fraction of fs_count_landed_hist with respect to the total
    - **fs_spotting_lkhd**: fs_frac_landed scaled by the gridpoint dry over wet fuel mass ratio

fs_gen_inst: firebrands generated at the given instant (latest generation cycle)
fs_landing_mask: logical mask of landings on unburnt gridpoints (latest timestep)
fuel_spotting_risk:
fs_fire_area: fire area on the atmospheric grid mesh

<u>Namelist parameters</u>

Firebrand Generation

- Frequency:

    fs_firebrand_gen_dt sets the frequency to generate new firebrands as a multiple of the inner nest timestep. fs_firebrand_gen_dt = 2 generates a new array of firebrands every other timestep. This parameter does not modify the integration period of other processes, such as transport and firebrand physics.

- Source Limit:

  fs_array_maxsize sets the maximum size of the array storing properties of active firebrands. If compiled with MPI support, it corresponds to the array size for each tile. At each generation cycle, elements storing firebrand properties are introduced in this array, while at each timestep, elements corresponding to firebrands that landed, burned out, or exited the domain are removed from the array. If the number of elements available in a tile is lower than the number of firebrand sources for the same tile, new firebrands are not appended to the array for that given tile until enough elements become available in the array. Firebrand sources are the coordinates from where firebrands will be released in each generation cycle. Firebrand sources are unique sets associated with the grid center point on the fire refined grid mesh.

  fs_firebrand_gen_lim sets an approximate limit to the number of firebrand sources that will generate firebrands per cycle. Only gridpoints where fire Rate-Of-Spread (ROS, [fractional gridpoint area]) is greater than zero during the interval between generation cycles (grid%burnt_area_dt/dt > 0) are considered. If the number of valid points from all tiles is below **fs_firebrand_gen_lim**, all sources will generate firebrands. If not, a rank is created using the fuel mass consumed by fire between generation cycles as a measure for firebrand generation potential at each source (grid%fgip [kg/m2] x ROS). The MPI communications of all valid grid points from the tiles to the main processor can be time consuming for large fires. Hence a threshold for the generation potential is estimated by communicating only two integers from each tile, the median rank of generation potential and number of valid sources. The main processor uses a linear regression to estimate a cutoff threshold based on **fs_firebrand_gen_lim** and the total number of valid sources. Because the distributions of fuel mass consumed are non-Gaussian and can vary substantially among tiles, the limit established by **fs_firebrand_gen_lim** is a rough approximation and may not yield the expected results depending on the case. Note that setting **fs_firebrand_gen_lim** = **fs_array_maxsize** may prevent firebrand generation in large fire cases because firebrands are not generated when the number of sources exceeds the elements available in the array. Even though the threshold estimation is ad hoc and rough, having no threshold would potentially produce biased results, once tiles with less intense fire become the only tiles with enough array space to generate firebrands.
- Levels:

  fs_firebrand_gen_levels sets the number of vertical levels to generate firebrands. Firebrands are distributed between 1 meter Above Ground Level (AGL) and **fs_firebrand_gen_maxhgt**. When **fs_firebrand_gen_levrand** is true, firebrands are randomly placed, when false, firebrands are equally distributed.
- Momentum:

fs_firebrand_gen_mom3d_dt number of advection cycles to allow firebrands to generate 3-D momentum before burnout process begins.

- Initial Firebrand Properties:
    - **fs_firebrand_gen_prop_diam**: diameter
    - **fs_firebrand_gen_prop_effd:** effective diameter
    - **fs_firebrand_gen_prop_temp**: temperature
    - **fs_firebrand_gen_prop_tvel**.: terminal velocity

Firebrand Physics

- Constant Firebrand Properties: **fs_firebrand_dens**, **fs_firebrand_dens_char**.

Firebrand Landing

- Height: **fs_firebrand_land_hgt**.
- Lifetime: **fs_firebrand_max_life_dt**.