Meter-scale corona system and time-domain fractal lightning models

Dr. Brant Carlson

CARTHAGE COLLEGE

Carthage College → Los Alamos National Lab

April 2025 | Lightning Modeling Grand Challenge Workshop

Two models...

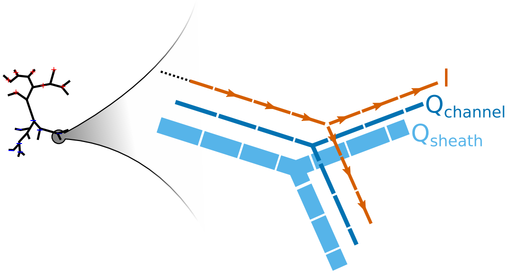
Time-domain fractal lightning (TDFL)

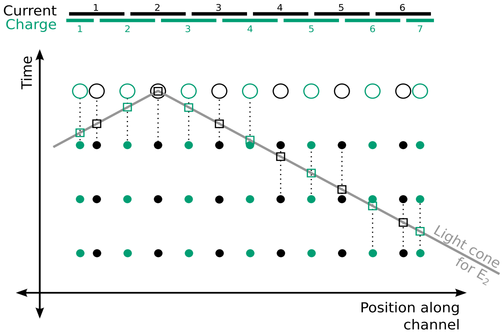
- kilometer-scale leader channel/sheath model
- Self-consistent electrodynamics, time evolution
- Parameterized leader channel behavior

Streamer tree

- meter-scale streamer system / corona model
- Quasi-static E-field
- Parameterized streamer/channel behavior

Time-domain fractal lightning, basics





$$\boldsymbol{E}_{\mathbf{t}}(\boldsymbol{x},t) = \frac{1}{4\pi\epsilon_0}\int d^3x' \left\{ \frac{\hat{\boldsymbol{R}}}{R^2}\left[\rho(\boldsymbol{x}',t')\right]_{\text{ret}} + \frac{\hat{\boldsymbol{R}}}{cR}\left[\frac{\partial\rho(\boldsymbol{x}',t')}{\partial t'}\right]_{\text{ret}} - \frac{1}{c^2R}\left[\frac{\partial \boldsymbol{J}(\boldsymbol{x}',t')}{\partial t'}\right]_{\text{ret}} \right\}$$

$$t' = t - R/c \qquad R \rightarrow \sqrt{R^2 + r_{\text{chan}}^2}$$

$$I = E_{\text{t}}/R_l \qquad R_l(T) \propto T^{-9/8} e^{\epsilon/k_B T}$$

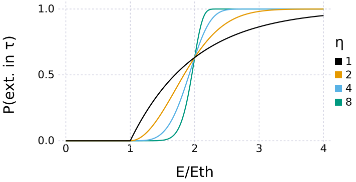
$$C_l \frac{dT}{dt} = I^2 R_l \qquad \frac{dQ_{\text{sheath}}}{dt} = \frac{Q_{\text{channel}}}{\tau}$$

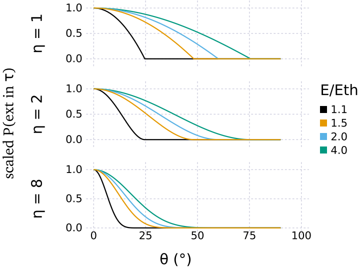
$$R_l(10^4\,\text{K}) = 1\,\Omega/\text{m} \qquad C_l = 2\,\text{J/K/m} \qquad \tau = 0.5\,\mu\text{s}$$

Time-domain fractal lightning, extension

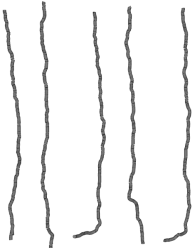
$$P(\text{ext}) \propto \left(\frac{E - E_{\text{th}}}{E_{\text{th}}}\right)^{\eta} \rightarrow \frac{dP}{dt}$$

$$P(\text{ext}) \propto 1 - e^{-\left(\frac{E-E_{\text{th}}}{E_{\text{th}}}\right)^{\eta} \frac{t}{\tau}}$$

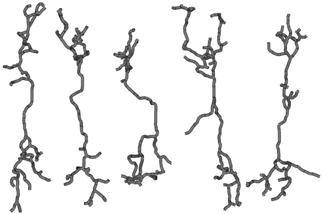




Low branching (η=8)



high branching (η=2).



TDFL gist

Included physics

- Charge, current, full electrodynamics
- Channel extension, branching, conductivity evolution, heating/cooling
- Corona sheath

Limitations

- *Very* parameterized channel behavior
- Corona sheath cylindrically symmetric, fixed radius
- Assumed extension preconditions
- Polarity asymmetry

Implementation details

- C, MPI, command line. Documented, but in need of some updates, performance improvements.

TDFL input/output, validation

Input

- Background E-field
- initial channel segment(s)

Output

- Channel geometry, extension over time
- Current and charge motion
- Static and dynamic E-fields produced

Validation

- Not well-validated
- Possible: compare to LMA, field change meter, high-speed camera

TDFL integration with other models

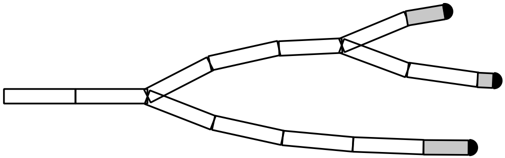
- Cloud / electrification $\rightarrow$ initial E-field
- ...initiation? $\rightarrow$ initial channel
- Corona sheath evolution/geometry $\rightarrow$ sheath radius, extension preconditions
- Channel plasma physics $\rightarrow$ channel heating/cooling, resistance
- Overall charge transfer $\rightarrow$ cloud charge distribution modification

Next steps

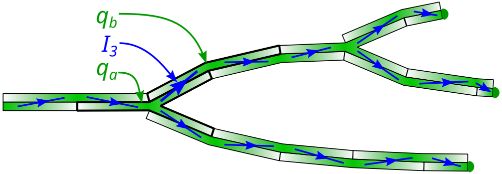
- Better channel model
- Better corona sheath treatment

Streamer tree model

Segmentation



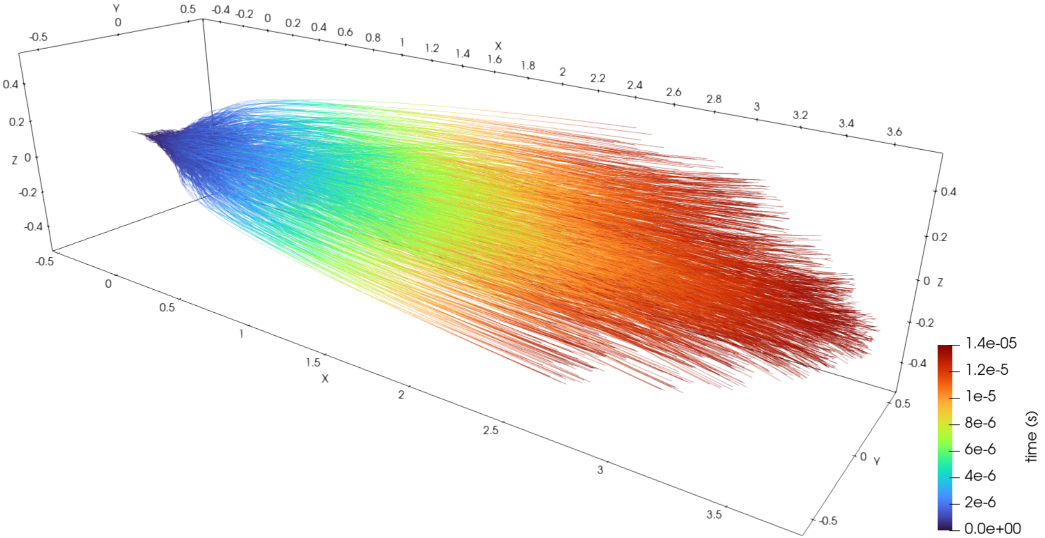
Charge and current geometry



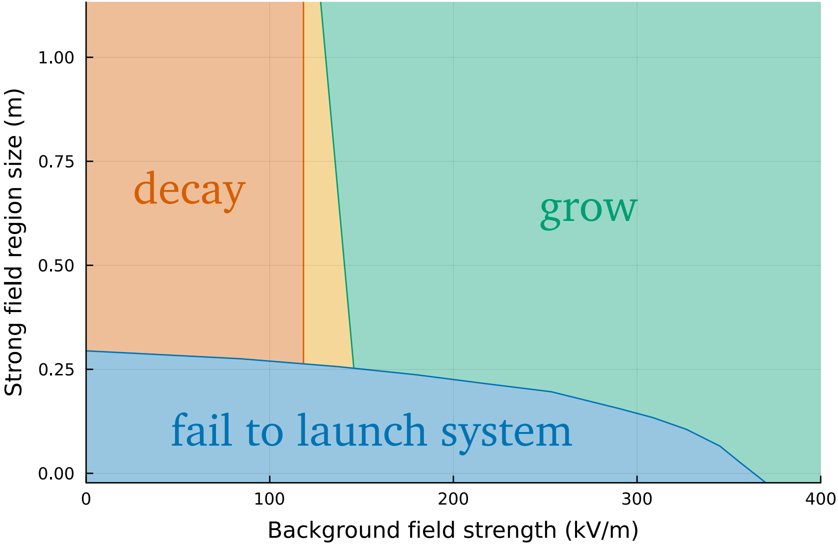
Streamer tree model outline

- Streamer tips
  - Charge, motion, growth/decay, branching
- Channels
  - "Electron density", ionization/attachment, current, charge
- Overall
  - $\vec{E}$-field $\rightarrow$ tip motion, growth/decay, channel ionization/attachment
  - Potential $\rightarrow$ channel current, charge
  - Events: collisions, connections, branching, death
  - Julia, `DifferentialEquations.jl`, FMM, multithreading...

Streamer tree sample results



Metastreamer behavior



Streamer tree model gist

Included physics

- Streamer motion, growth/decay, branching, connection, launch
- Tip and channel charge, channel current
- Channel conductivity, evolution ("ionization" and "attachment")
- Self-consistent quasi-static electric field (fast multipole method)
- Collective effects

Limitations

- *Very* parameterized streamer behavior
- *Very* parameterized channel behavior
- No dynamic leader driving system
- No streamer/leader transition
- Occasional stability issues (e.g. channel overlap)

Implementation details

- Julia, multi-threaded, command line. Code largely undocumented.

Streamer tree input/output, validation

Input

- Background E-field
- initial channel segment(s)

Output

- Tree geometry, extension timescale
- Current and charge motion
- Static and dynamic E-fields produced
- Statistical properties(?)

Validation

- *Not* validated
- Possible: compare to streamer experiments, sprite observations, lab spark observations

Streamer tree integration with other models

- Streamer models $\rightarrow$ better streamer, channel parametrizations
- Leader channel models $\rightarrow$ input current, field
- Leader plasma physics $\rightarrow$ leader transition?
- Large-scale corona geometry $\rightarrow$ statistical distributions (e.g. for TDFL)

Next steps

- Better parametrization, validation
- Explore behavior, encapsulate for use in other models
- Metastreamer dynamics?
- Streamer-leader transition?

Summary

Two models

- Kilometer-scale leader electrodynamics $\longrightarrow$ TDFL
- Meter-scale corona $\longrightarrow$ streamer tree
- Both in need of validation, development