Design of a General-Purpose Local-Scale Ionospheric Model: *GEMINI*

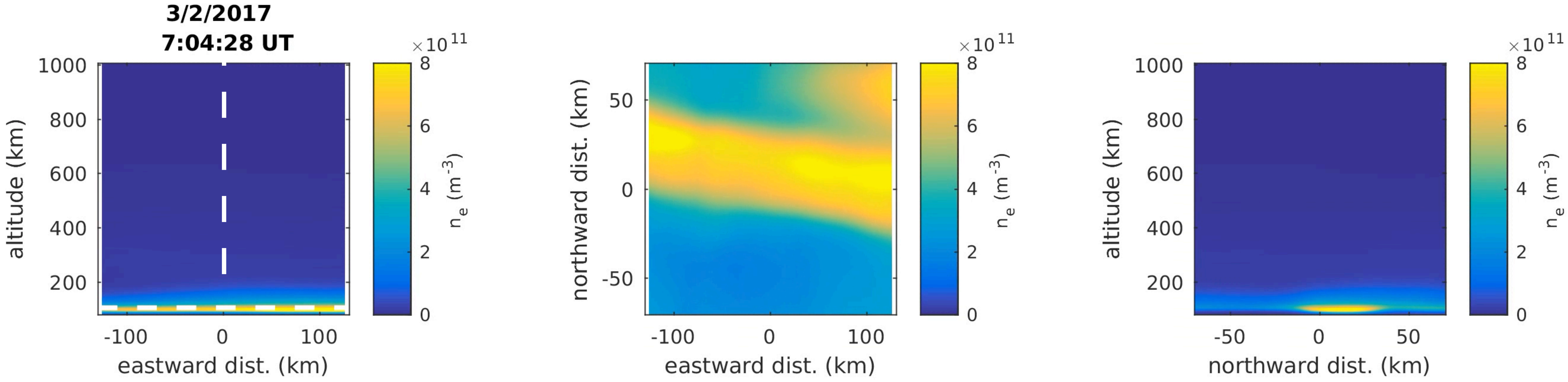
M. Zettergren, M. Hirsch, G. Grubbs, R. Clayton, M. Burleigh, P. Inchin, J. Snively, and K. Deshpande

GEMINI development supported by NSF CAREER grant AGS-1255181

Introduction

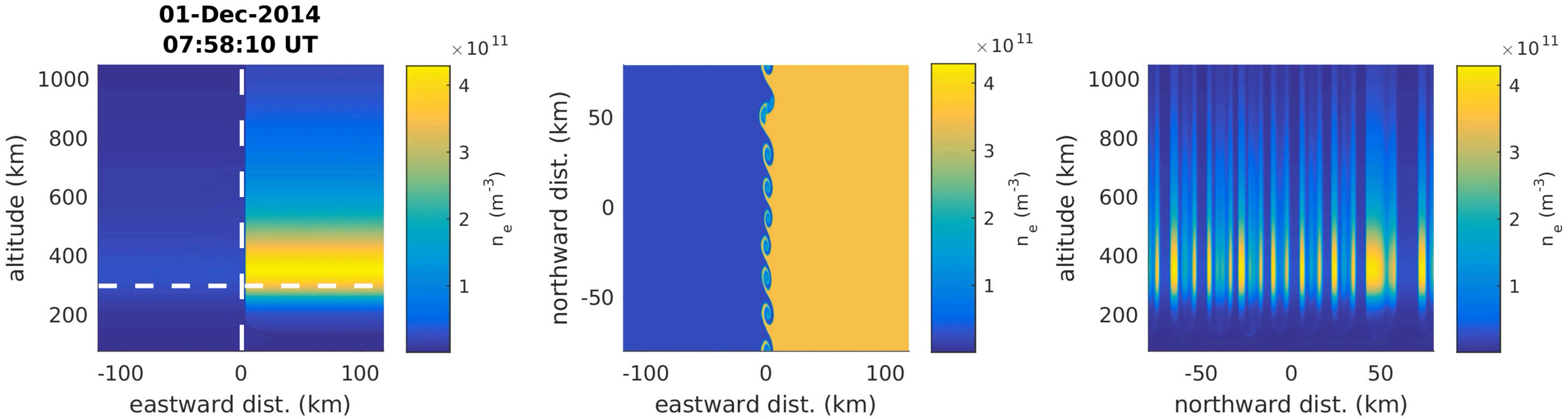
- ***GEMINI - Geospace Environment Model of Ion-Neutral Interactions***
    - local-scale ionospheric model (viz. not encapsulating full globe)
    - Included physics important at small scales and for strong forcing (ion inerta)
    - Can resolve down to 100 m scales, while still modeling a mesoscale region (100s of km extent)
    - Open-source software, distributed via GitHub (GPL 3.0)
    - Includes a multi fluid plasma description and self-consistent electrodynamics
    - Can use a physics-based model of energetic electron transport

Example: *auroral ionospheric responses*



Clayton et al, (2019)

Example: *ionospheric turbulence*



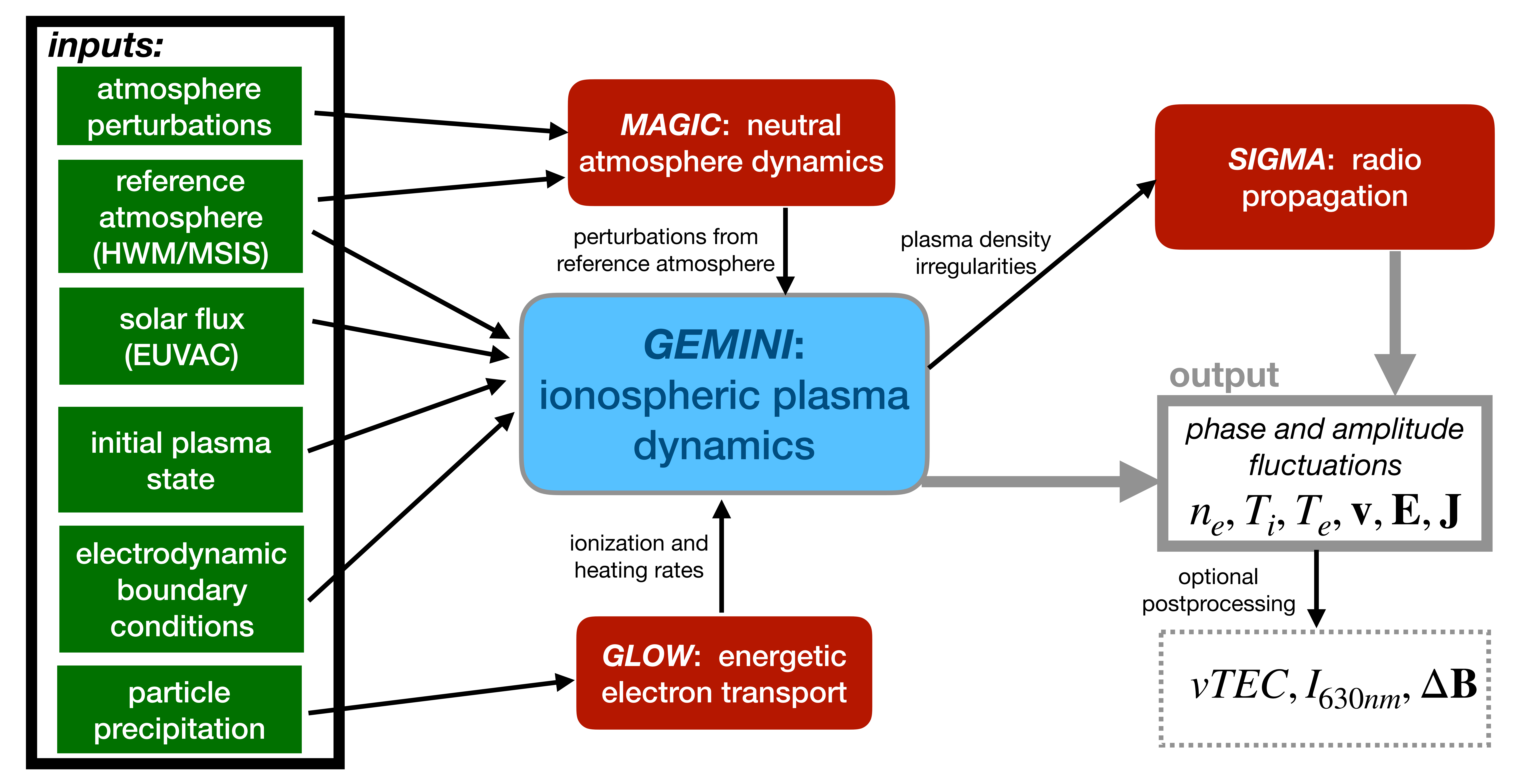
Spicher et al, (2019)

Example: *natural hazard effects on the ionosphere*

Current Needs for GEMINI

- *Flexible enough to deal with the range of problems in which our groups is interested*
    - Auroral forcing (currents and particles)
    - Polar plasma instabilities and radio impacts
    - Neutral dynamical effects on ionosphere at mid— and low- latitudes
- 2D and 3D simulations from same code based for rigorous comparisons
- Range of applications dictates requires grid and core numerical code flexibility
    - Use of generalized coordinates - dipole for low lats., Cartesian for polar, etc.
    - Complicates code; fortran support for structure and OO concepts help to manage
- Attempts made to favor ***code clarity*** and ***flexibility*** (maybe above efficiency)

Software Interfaces



GEMINI Governing Equations

5-moment fluid system of equations (steady-state perp. to B) + heat flux:

$$\frac{\partial \rho_s}{\partial t} + \nabla \cdot \left(\rho_s \mathbf{v}_s\right) = m_s P_s - L_s \rho_s$$

$$\hat{\mathbf{e}}_1 \cdot \left\{ \frac{\partial}{\partial t}\left(\rho_s \mathbf{v}_s\right) + \nabla \cdot \left(\rho_s \mathbf{v}_s \mathbf{v}_s\right) = -\nabla p_s + \rho_s \mathbf{g} + \frac{\rho_s}{m_s} q_s \left(\mathbf{E} + \mathbf{v}_s \times \mathbf{B}\right) + \sum_t \rho_s \nu_{st} \left(\mathbf{v}_t - \mathbf{v}_s\right) \right\}$$

$$\frac{\partial}{\partial t}\left(\rho_s \epsilon_s\right) + \nabla \cdot \left(\rho_s \epsilon_s \mathbf{v}_s\right) = -p_s(\nabla \cdot \mathbf{v}_s) - \nabla \cdot \mathbf{h}_s - \frac{1}{(\gamma_s - 1)} \sum_t \frac{\rho_s k_B \nu_{st}}{m_s + m_t} \left[2(T_s - T_t) - \frac{2}{3}\frac{m_t}{k_B}(\mathbf{v}_s - \mathbf{v}_t)^2\right]$$

$$\mathbf{v}_{s\perp} = \boldsymbol{\mu}_{s\perp} \cdot \mathbf{E}_\perp$$

Quasi-electrodynamic, Equipotential field line formulation

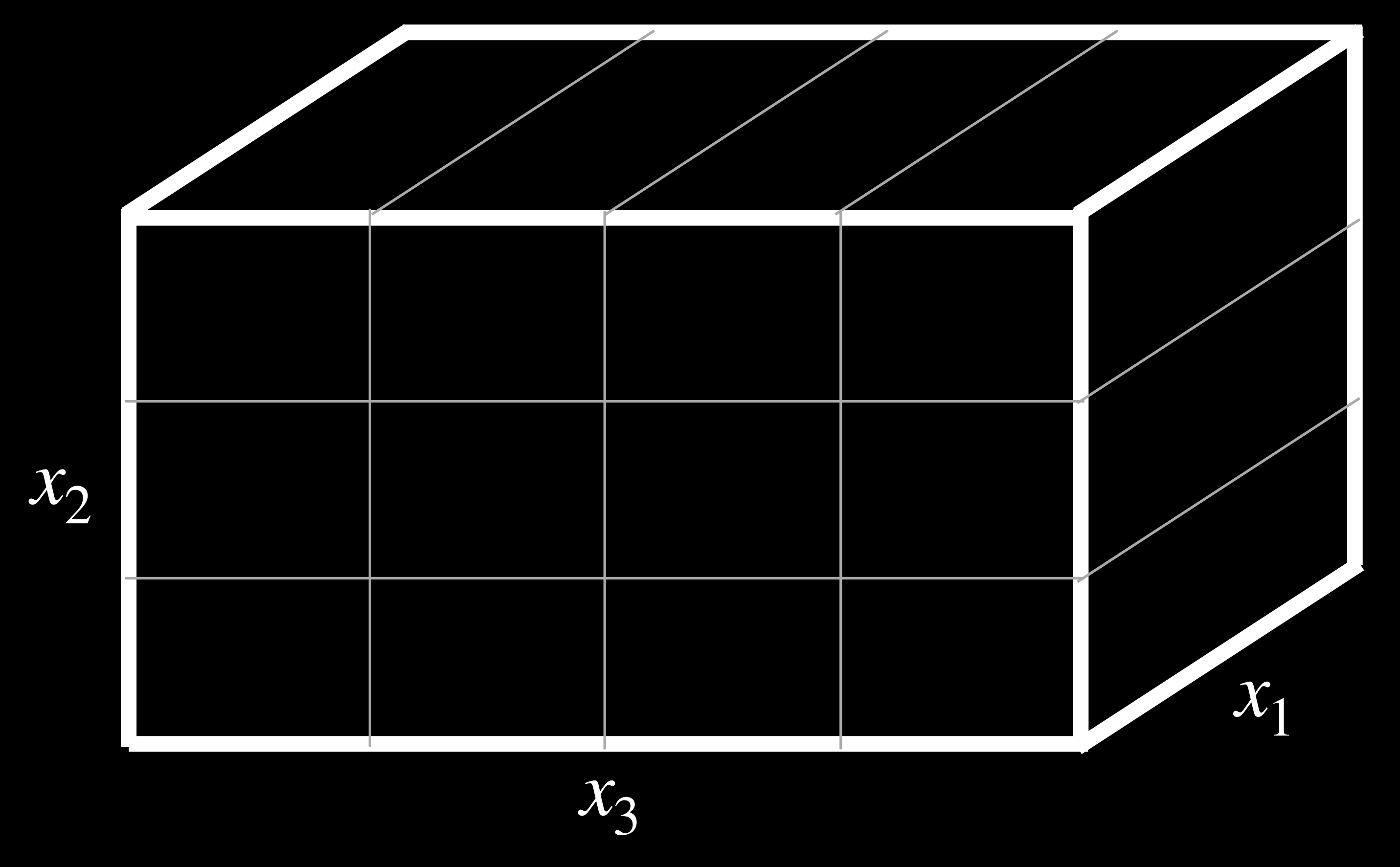
$$\nabla_\perp \cdot \left(\boldsymbol{\Sigma}_\perp \cdot \nabla_\perp \Phi\right) + \nabla_\perp \cdot \left[C_M \left(\frac{\partial}{\partial t} + \mathbf{v}_\perp \cdot \nabla_\perp \right) \left(\nabla_\perp \Phi\right) \right] = \nabla_\perp \cdot \left[\boldsymbol{\Sigma}_\perp \cdot \left(\mathbf{E}_{0\perp} + \mathbf{v}_{n\perp} \times \mathbf{B}_0 \right) \right]$$

Numerical Details

- Ionospheric equations are mixed-type (e.g. hyperbolic+parabolic+sources, elliptic)
- Operator (time-step) splitting used to separate different character and solve piecemeal - Godunov
  - hyperbolic - finite volume method
  - parabolic - TRBDF2 scheme
  - source/loss - RK2 or ETD
  - elliptic - sparse unsymmetric LU factorization, direct (MUMPS)
- *To the greatest degree possible we separate different terms in the equations in order to make it feasible to try out different numerical schemes and assess the effects of numerical choices on the simulated results*

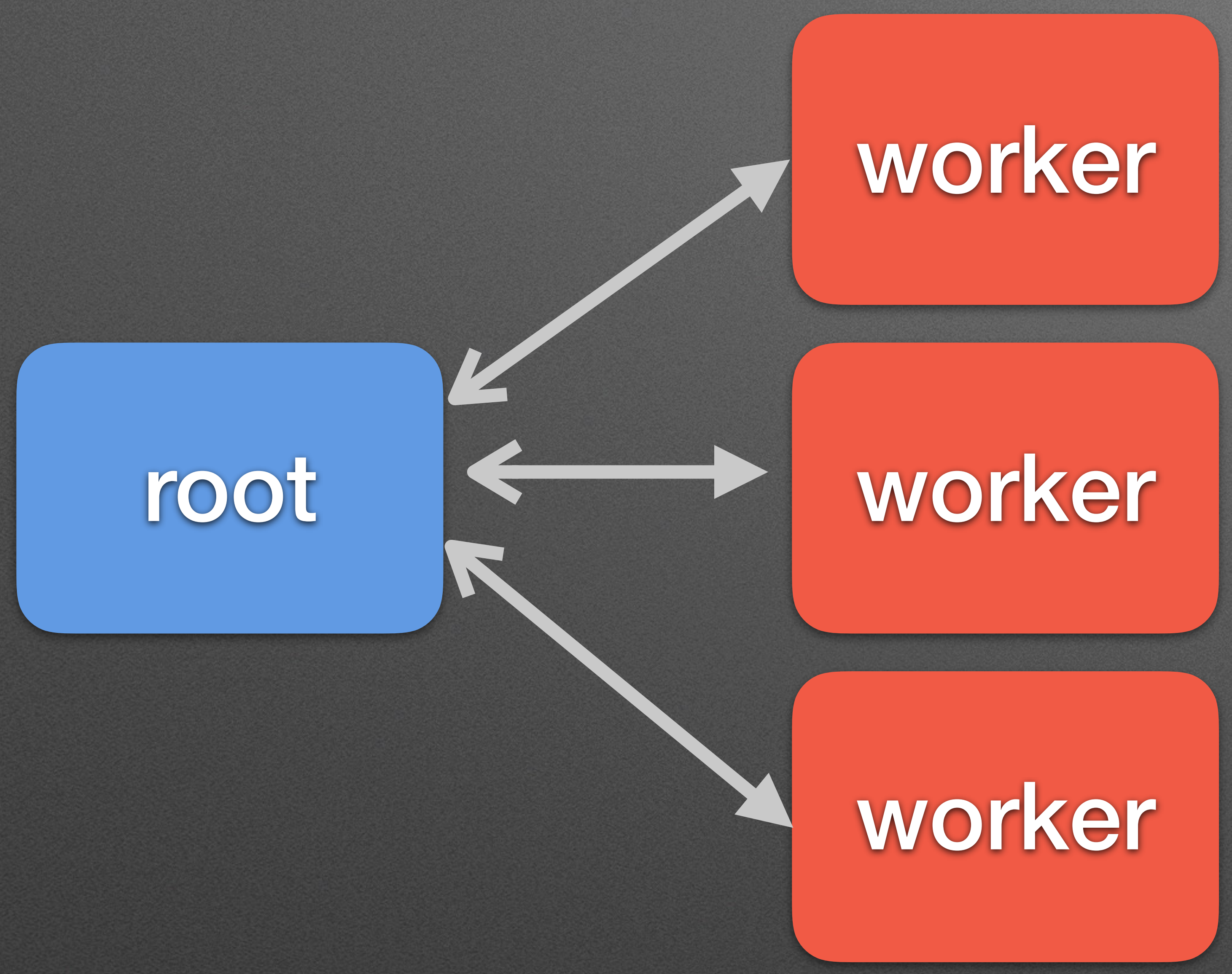
Problem Parallelization

- Distributed memory domain parallelization - scalable from 1 to ~1024 cores
  - openMPI libraries used (3.x)
  - tested on hardware from a laptop to HPC
- Parallel domain division in 2 dimensions only
  - Explicit methods are fairly straighforward to parallelize this way - simply pass boundary conditions to adjacent sub-domains
  - Cannot (easily) divide in three directions: solvers parallel to B use implicit schemes
- GPU and shared memory extremely challenging
- We limit ourself to second order numerical schemes in order to minimize the amount of data that needs to be passed between subdomains

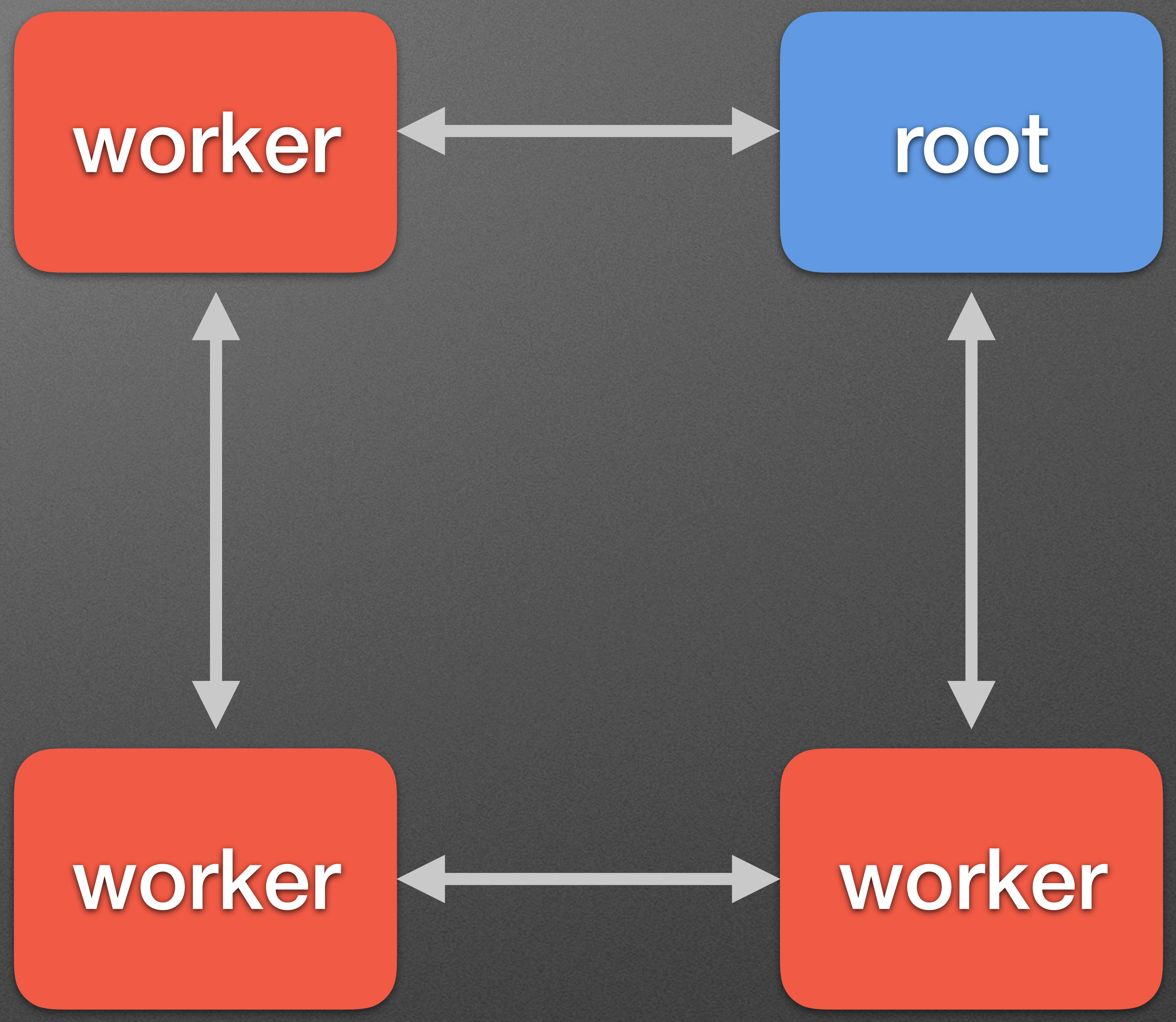


parallel subdomains are physically "curved" if curvilinear grid is being used

Parallel Patterns of Communication



gather and broadcast



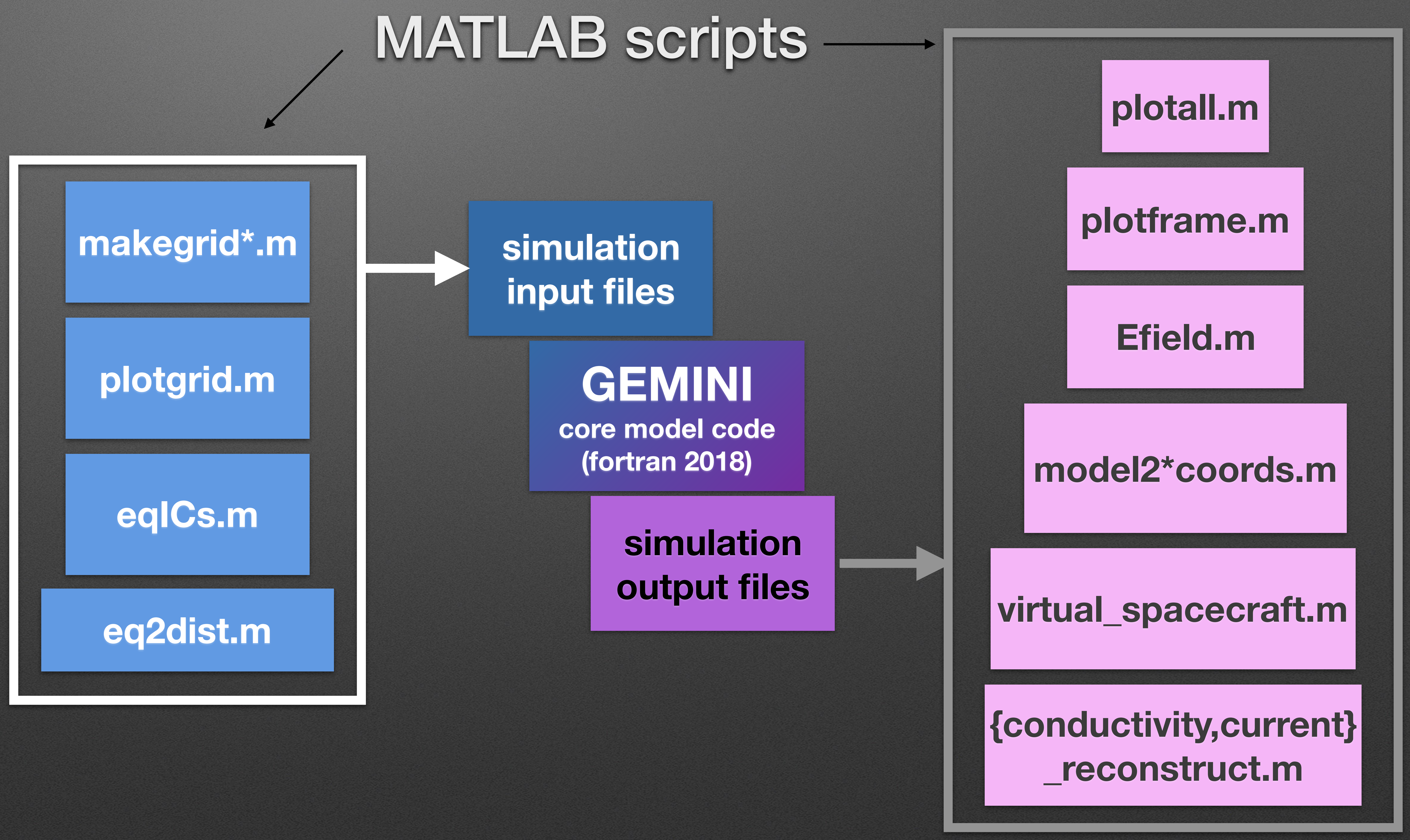
peer to peer

- Input and output - root broadcasts and root gathers, viz. parallel output not supported
- Electrodynamics - gather and broadcast due to numerical approach used
- Fluid - peer to peer needed for advection and compression; source and diffusion can be done without any boundary passing


GEMINI
core model code
./gemini.f90
Key
main program flow
module functional dependence
submodule encapsulation
top-level modules
numerical component module
utility module
temporal
time step calculations; date and time handling routines, stability evaluation and time-step choice
./temporal/temporal.f90
potential_comm
sets up and solves electrodynamic equation
./numerical/potential/potential_comm_mumps.f90
neutral
computes neutral atmosphere: MSIS background and perturbations from MAGIC
./neutral/neutral.f90
multifluid
solves full system of fluid equations (all ion species)
./multifluid/multifluid.f90
io
file input (config and initial conditions) and output
./io/io.f90
potential
(MUMPS-based) elliptic solver for ionospheric potential equation
./numerical/potential/potential_mumps.f90
collisions
collisions frequencies, conductivities and intertial capacitance
./collisions/collisions.f90
advection
finite volume hyperbolic solver and boundary conditions
./numerical/advection/advec_mpi.f90
sources
source term calculations in conservation laws; collisional terms and chemistry
./sources/sources.f90
diffusion
(LAPACK-based) thermal diffusion (parabolic) solver and boundary conditions
./numerical/diffusion/diffusion.f90
ionization
impact ionization and photoionization
./ionization/ionization.f90
(sub) GLOW
impact ionization from physics-based model
./ionization/ionization.f90
"utility" modules (used throughout other program units)
mpimod (OpenMPI-based)
message passing routines: halo, broadcast, and gather operations
./numerical/mpimod/mpimod.f90
phys_consts
universal constants used in model
./numerical/constants/phys_consts.f90
calculus
differentiation (grad, div) and integration routines.
./numerical/calculus/calculus.f90
interpolation
interpolation routines for neutral, precipitation, and electric field input data
./numerical/interpolation/interpolation.f90
grid
mesh structure and associated routines to read grid file
./numerical/grid/grid.f90
(sub) mpisend
custom send-based routines
./numerical/mpimod/mpisend.f90
(sub) mpirecv
custom receive-based routines
./numerical/mpimod/mpirecv.f90
(sub) mpihalo
custom halo-based routines
./numerical/mpimod/mpihalo.f90

Significant Scripting Interfaces

- **Script goal(s):**
  - Limit user exposure to core model code after initial build
  - Perform CPU-nonintensive tasks
  - Reduce/eliminate the need to change, edit, or recompile core model
  - Provide additional functionality that is either optional (and possibly unnecessary for some use cases), or can more easily be done through interactive processing
- E.g. *create and plot a grid, set up an equilibrium simulation, interpolate initial condition up to a high-res. grid,* etc.



Example Simulation Workflow

1. *Create a grid* (viz. define grid extent) for the process you want to study

   A. Make a low-res. grid for simulating an equilibrium ionosphere for your initial conditions.

   B. A higher-res grid can (should?) be used for simulating fine-scale ionospheric disturbances.

2. *Generate initial conditions* for your simulation

   C. *Run an **"equilibrium simulation"***.

   D. Interpolate the output of the equilibrium run up to the full grid resolution that you wish to use for your **"disturbance simulation".**

3. *Run the disturbance simulation.* For larger problems this can require an HPC with 100s of cores.

4. *Postprocess and analyze the results*.

This workflow is encoded in the myriad examples made available at: https://github.com/gemini3d/GEMINI-examples/

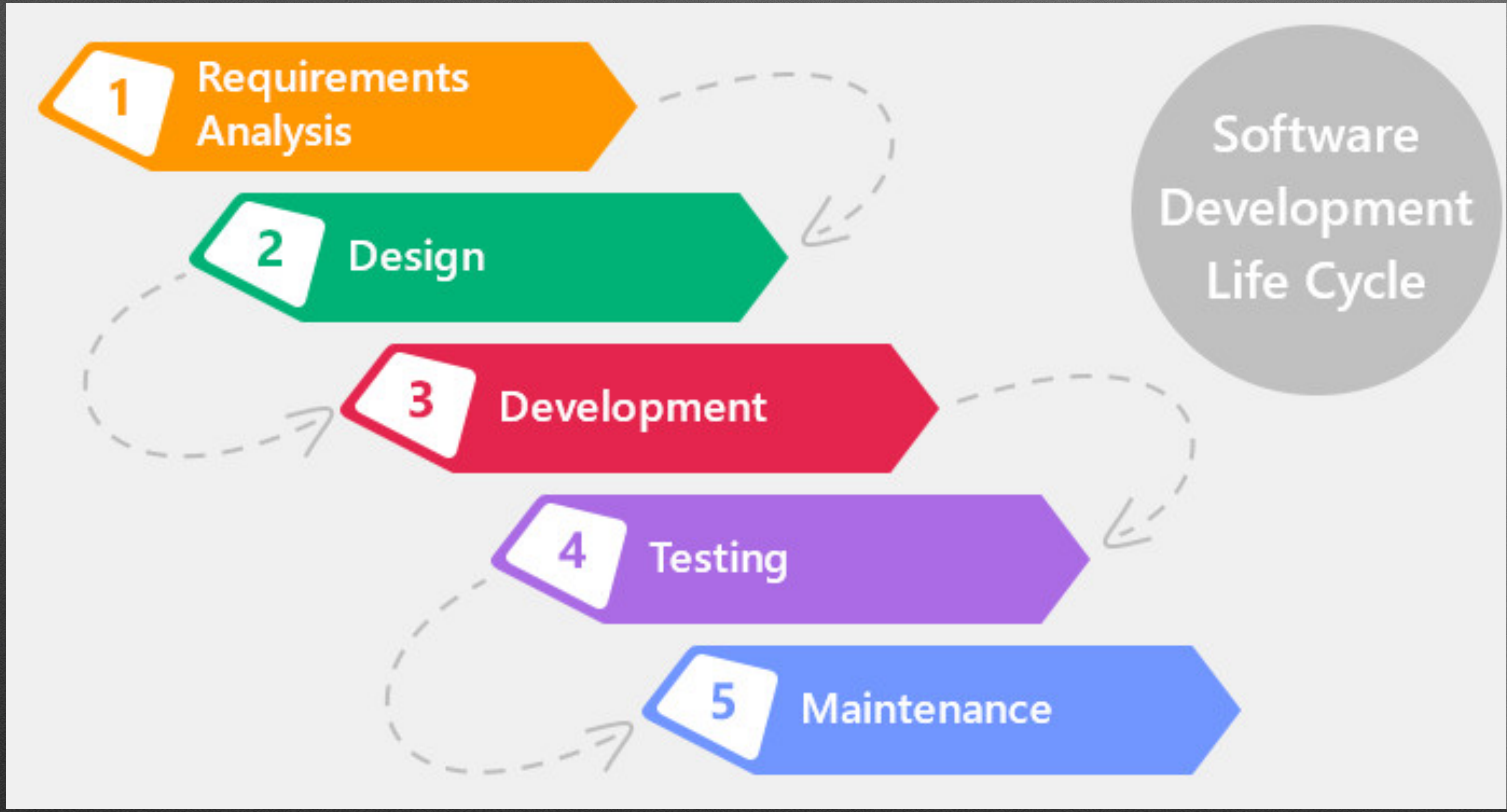
Workflow Nomenclature

- **Equilibrium simulation** - a low resolution simulation which start the ionosphere in an ad hoc state and allows it to relax to an equilibrium representative of the "background" ionospheric state for a given location, date, time, solar activity level, and geomagnetic activity level. Practically this is usually achieved by running the model for a day from some initial condition.

- **Disturbance simulation** - a high resolution simulation starting from an initial equilibrium state and modeling the ionsopheric response to some type of neutral or auroral forcing

Engineering of *scientific* software

"standard" software engineering process



- Requirements are often not well-specified and shifty
- Results considered more important than usability
- First two stages are usually "rushed" and blend directly into implementation - due to "research" nature of problems....
- Maintenance is almost always unfunded; affects whether non-functional requirements (e.g. regarding interfaces) can be met

GEMINI

- Geospace Environment Model of Ion-Neutral Interactions - a general purpose, local scale, nonlinear, numerical ionospheric model.
- https://github.com/gemini3d
    - GEMINI - main fortran code
    - GEMINI-scripts: post processing and plotting
    - GEMINI-examples: curated examples of how to run with different setups
    - GEMINI-docs: documentation (a work in progress)
- Download it, use it, break it, ask for help, fix problems and contribute to the project!
- We value your participation and feedback!
- Official release coming soon!