

GEMINI Implementation Details

Matthew D. Zettergren, PhD
matthew.zettergren@icloud.com

May 7, 2023

Contents

1 GEMINI Model: Executive Summary	3
2 GEMINI Model Heritage	4
3 GEMINI Source Code Repositories	6
4 Internal Organization of GEMINI	7
5 Cmake Build System for GEMINI	10
6 GEMINI Data Structures	10
6.1 Mesh Objects	10
6.2 Inputdata Objects	12
6.3 Output Data Formats and Files	12
7 GEMINI Umbrella Library: libgeminib	12
7.1 C-fortran Interoperability	13
8 trees-GEMINI Application	13
8.1 Use of trees-GEMINI within the FIGMENTS framework	14

9 Postprocessing and Visualization Scripts	17
10 Continuous Integration (CI)	17
10.1 Automatic CI	17
10.2 Comprehensive Continuous Integration (long CI)	17
11 Examples and Comparisons Against GNSS Data	18

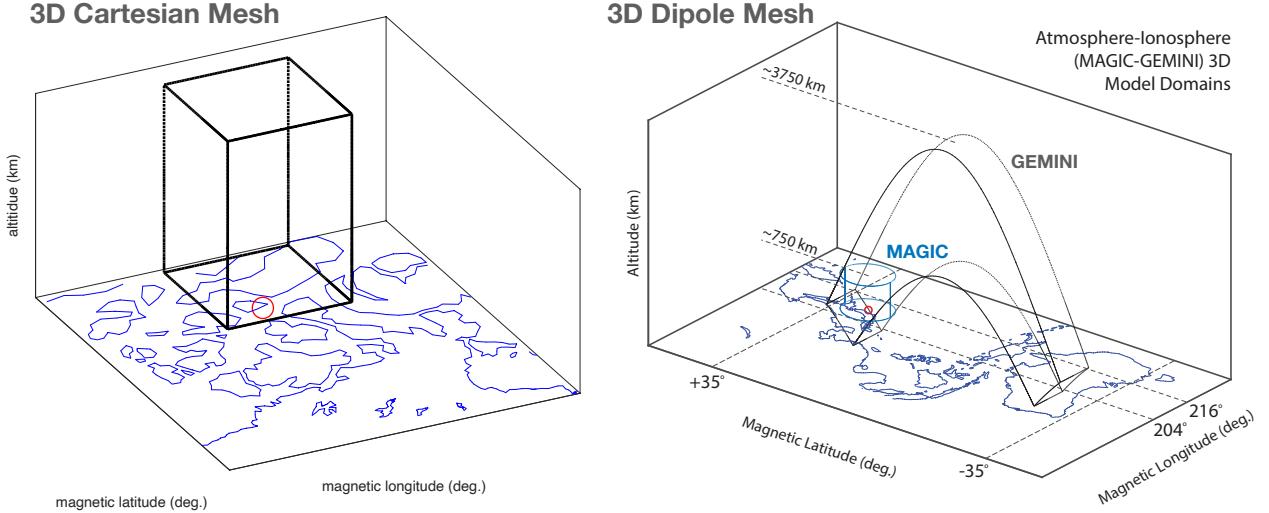


Figure 1: Examples of commonly used GEMINI mesh configurations: (left) a Cartesian mesh centered near the geomagnetic pole near Resolute Bay Canada, (right) an interhemispheric dipole mesh used with studies of infrasound and gravity wave impacts on the ionosphere.

1 GEMINI Model: Executive Summary

GEMINI: The “Geospace Environment Model for Ion-Neutral Interactions” is a 3D ionospheric model based on Zettergren and Semeter (2012), that has been applied in wide variety of ionospheric studies (Zettergren, op. cit.). The model uses a tilted dipole (Huba et al., 2000) or Cartesian mesh, and comprises a fluid system of equations (Schunk, 1977; Blelly and Schunk, 1993) describing dynamics of the ionospheric plasma, self-consistently coupled to a quasi-static (defined below) treatment of electrical currents. The fluid system is a set of three conservation equations (mass, parallel momentum, and energy) for each ionospheric species (Zettergren and Snively, 2015, Appendix A). Source terms in the continuity equations include photoionization Solomon and Qian (2005); Richards et al. (1994) and impact ionization via integration of the physics-based energetic electron transport model, GLOW (Solomon, 2001). Efficient semi-empirical methods for energy deposition calculations (Fang et al., 2008) can also be used. A full set of chemical reactions needed for the E- and F-regions is taken from Diloy et al. (1996); St-Maurice and Laneville (1998).

Perpendicular components of the plasma momentum equations in GEMINI are resolved via a quasi-static force balance approximation (Zettergren et al., 2015), viz. a static solution is applied but the parameters of the system (importantly conductivity) are allowed to change in time. Electric fields are found by enforcing a divergence-free current density, where the current consists of conduction, polarization, and pressure contributions (e.g. as in ?). This approach assumes a leading order electrostatic description supplemented by terms correcting for polarization currents from varying field structures (Mitchell et al., 1985) (i.e. ion inertial effects). Because the GEMINI solver is inherently (quasi-)static in nature it can be driven from either field-aligned current (FAC) or electric field conditions boundary conditions. Specifically, the ionospheric electric field is approximated by $\mathbf{E} = -\nabla\Phi$ but current closure solutions include ion inertial effects.

The use of a quasi-static solution notably omits Alfvén waves; however, many mesoscale structures are well-described as static (e.g. Lynch et al., 2015). Recent reviews suggest that alternating current (AC) Poynting flux is typically reported at lower levels than direct current (DC) Poynting flux, but can be highly variable; the relative importance of these two sources (DC vs. AC) is unknown (?). Our modeling focuses on the DC input effects (and particle inputs) as we argue this is the only part of the Poynting flux that can feasibly be captured by non-idealized models – resolving propagation of Alfvén waves in a cross-scale (meso-to-large), 3D ionosphere-thermosphere (plasma-neutral coupling) model is computationally infeasible for resolutions and grid extents that we target. At the very minimum, research suggests that the DC part of the incoming energy and momentum (stress) is a major (perhaps even the largest) contributor and our data-driven methodology permits us to evaluate the extent to which the observations are DC. As better computational resources and algorithms become available in the future it should be possible to begin to address AC portions of the spectrum, but that is not within the project scope.

GEMINI presently assumes that the geomagnetic field lines are equipotentials (a particularly good assumption for highly conducting situations likely to accompany precipitation) and employs a field line integration in the equation solved to describe conservation of charge (cf. Zettergren and Snively, 2015, Equation 1). Newer versions of GEMINI also include additional current terms needed for high-resolution modeling, like diamagnetic currents and diffusive transport perpendicular to \mathbf{B} .

A comprehensive set of documentation describing the mathematical formulation of the model and some details of numerical solutions is not included here for brevity, but may be found in the gemini-docs repository, specifically at <https://github.com/gemini3d/gemini-docs/blob/main/formulation/GEMINI.pdf>.

2 GEMINI Model Heritage

The use of GEMINI for scientific applications is documented in a number of studies and associated refereed journal articles:

- Zettergren, M. and Semeter, J. (2012). Ionospheric plasma transport and loss in auroral downward current regions. *Journal of Geophysical Research: Space Physics*, 117(A6)
- Zettergren, M. and Snively, J. (2013). Ionospheric signatures of acoustic waves generated by transient tropospheric forcing. *Geophysical Research Letters*, 40(20):5345–5349
- Zettergren, M., Lynch, K., Hampton, D., Nicolls, M., Wright, B., Conde, M., Moen, J., Lessard, M., Miceli, R., and Powell, S. (2014). Auroral ionospheric f region density cavity formation and evolution: Mica campaign results. *Journal of Geophysical Research: Space Physics*, 119(4):3162–3178
- Zettergren, M. and Snively, J. (2015). Ionospheric response to infrasonic-acoustic waves generated by natural hazard events. *Journal of Geophysical Research: Space Physics*, 120(9):8002–8024

- Zettergren, M., Semeter, J., and Dahlgren, H. (2015). Dynamics of density cavities generated by frictional heating: Formation, distortion, and instability. *Geophysical Research Letters*, 42(23)
- Lynch, K., Hampton, D., Zettergren, M., Bekkeng, T., Conde, M., Fernandes, P., Horak, P., Lessard, M., Miceli, R., Michell, R., et al. (2015). Mica sounding rocket observations of conductivity-gradient-generated auroral ionospheric responses: Small-scale structure with large-scale drivers. *Journal of Geophysical Research: Space Physics*, 120(11):9661–9682
- Perry, G., Dahlgren, H., Nicolls, M., Zettergren, M., St.-Maurice, J.-P., Semeter, J., Sundberg, T., Hosokawa, K., Shiokawa, K., and Chen, S. (2015). Spatiotemporally resolved electrodynamic properties of a sun-aligned arc over resolute bay. *Journal of Geophysical Research: Space Physics*, 120(11):9977–9987
- Burleigh, M. and Zettergren, M. (2017). Anisotropic fluid modeling of ionospheric upflow: Effects of low-altitude anisotropy and thermospheric winds. *Journal of Geophysical Research: Space Physics*, 122(1):808–827
- Swoboda, J., Semeter, J., Zettergren, M., and Erickson, P. J. (2017). Observability of ionospheric space-time structure with isr: A simulation study. *Radio Science*, 52(2):215–234
- Zettergren, M., Snively, J., Komjathy, A., and Verkhoglyadova, O. (2017). Nonlinear ionospheric responses to large-amplitude infrasonic-acoustic waves generated by undersea earthquakes. *Journal of Geophysical Research: Space Physics*, 122(2):2272–2291
- Burleigh, M., Heale, C., Zettergren, M., and Snively, J. (2018). Modulation of low-altitude ionospheric upflow by linear and nonlinear atmospheric gravity waves. *Journal of Geophysical Research: Space Physics*, 123(9):7650–7667
- Zettergren, M. and Snively, J. (2019). Latitude and longitude dependence of ionospheric tec and magnetic perturbations from infrasonic-acoustic waves generated by strong seismic events. *Geophysical Research Letters*, 46(3):1132–1140
- Burleigh, M., Zettergren, M., Lynch, K., Lessard, M., Moen, J., Clausen, L., Kenward, D., Hysell, D., and Liemohn, M. (2019). Transient ionospheric upflow driven by poleward moving auroral forms observed during the rocket experiment for neutral upwelling 2 (renu2) campaign. *Geophysical Research Letters*, 46(12):6297–6305
- Deshpande, K. B. and Zettergren, M. D. (2019). Satellite-beacon ionospheric-scintillation global model of the upper atmosphere (sigma) iii: Scintillation simulation using a physics-based plasma model. *Geophysical Research Letters*, 46(9):4564–4572
- Inchin, P., Snively, J., Zettergren, M., Komjathy, A., Verkhoglyadova, O., and Tulasi Ram, S. (2020). Modeling of ionospheric responses to atmospheric acoustic and gravity waves driven by the 2015 nepal 7.8 gorkha earthquake. *Journal of Geophysical Research: Space Physics*, 125(4):e2019JA027200
- Spicher, A., Deshpande, K., Jin, Y., Oksavik, K., Zettergren, M. D., Clausen, L. B., Moen, J. I., Hairston, M. R., and Baddeley, L. (2020). On the production of ionospheric irregularities

via kelvin-helmholtz instability associated with cusp flow channels. *Journal of Geophysical Research: Space Physics*, 125(6):e2019JA027734

- Inchin, P., Snively, J., Kaneko, Y., Zettergren, M., and Komjathy, A. (2021). Inferring the evolution of a large earthquake from its acoustic impacts on the ionosphere. *AGU Advances*, 2(2):e2020AV000260
- Díaz Peña, J., Semeter, J., Nishimura, Y., Varney, R., Reimer, A., Hairston, M., Zettergren, M., Hirsch, M., Verkhoglyadova, O., Hosokawa, K., et al. (2021). Auroral heating of plasma patches due to high-latitude reconnection. *Journal of Geophysical Research: Space Physics*, 126(12):e2021JA029657
- Clayton, R., Burleigh, M., Lynch, K. A., Zettergren, M., Evans, T., Grubbs, G., Hampton, D. L., Hysell, D., Kaepller, S., Lessard, M., et al. (2021). Examining the auroral ionosphere in three dimensions using reconstructed 2d maps of auroral data to drive the 3d gemini model. *Journal of Geophysical Research: Space Physics*, 126(11):e2021JA029749
- Lamarche, L. J., Deshpande, K. B., and Zettergren, M. D. (2022). Observations and modeling of scintillation in the vicinity of a polar cap patch. *Journal of Space Weather and Space Climate*, 12:27
- Burleigh, M., Lynch, K., Zettergren, M., and Clayton, R. (2022). Spatiotemporal limitations of data-driven modeling: An isinglass case study. *Journal of Geophysical Research: Space Physics*, 127(9):e2021JA030242

These studies encapsulate a wide range of applications of the model from equatorial to low/mid-latitude and high-latitude and a number of studies where the model output has been critically compared against available data in validation exercises. Additional validation tests not yet published are included in other sections of this document.

Due to the amount of testing, verification, and validation done as part of the GEMINI's development, the technology readiness level (TRL) of the model is suggested to be TRL6.

3 GEMINI Source Code Repositories

GEMINI is organized into about a dozen repositories ranging from core model numerical parts to scripts that allow one to build all external library dependencies. To run the code requires two repositories to be built (`external` and `gemini3d`) and installed while visualization requires a third, scripting library (either `mat_gemini` or `pygemini`).

Additionally trees-GEMINI (cf. Section 8) supports visualization toolkit output files (vtu) which can be plotted using the open source, 3D visualization software Paraview. Paraview represents a notable exception to the required visualization back-end scripts; vtu files can be read directly into Paraview using its graphical interface.

GEMINI File-Based Model Interfaces

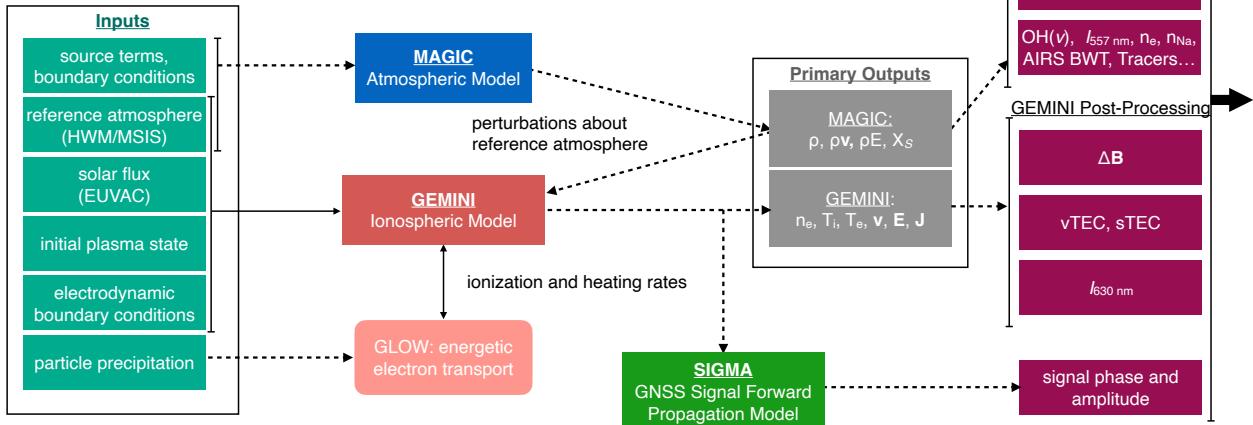


Figure 2: GEMINI file-based interfaces and postprocessing.

A complete list of gemini repositories and their basic purposes follow (see repository readme files for details). These are all collected into the gemini3d organization - and “umbrella” organization that holds all GEMINI code repositories, documentation, and bug-tracking and includes:

- gemini3d - the core GEMINI fortran, C/C++ code.
- pygemini - Python front- and back-end scripts for preparing simulation input data and gridding and plotting output.
- mat_gemini - MATLAB front- and back-end scripts for preparing simulation input data and gridding and plotting output.
- gemini-docs - Set of documentation for the model
- gemini-examples - Set of initialization files and scripts for creating different types of simulation setups.

4 Internal Organization of GEMINI

Figure 2 shows a top level diagram of GEMINI inputs and coupling with other AIRWaveS models, specifically for the situation where model data are passed through files – recent advances have made it possible to pass information from MAGIC into GEMINI through memory using the forestclaw mesh management software; however, GEMINI still retains the ability to do coupling through file input/output, which while slower is simpler to use and does not require AMR-related libraries.

The main GEMINI model inputs that need to be provided (through files) are:

- Source terms for generating perturbations in the MAGIC model
- Solar fluxes for describing dayside photoionization processes (where relevant)
- An initial ionospheric state; this is often generated through an “equilibrium” simulation where the model is run for a full day or longer with only solar forcing.
- Boundary conditions for electromagnetic solutions (background electric fields, boundary potentials and/or field-aligned currents)
- Kinetic boundary conditions for specifying high-latitude electron precipitation inputs (as needed)
- Reference atmospheres (usually from empirical models)
- Atmospheric perturbations from external models (e.g. MAGIC), as required

GEMINI provides 3D and time-dependent plasma output parameters including density, drift, and temperature of the 7 main ion species resolved along with electric field and current densities. Additionally information about magnetic field perturbations and optical intensities (in the case of chemically reactive species) can be computed from the output. The plasma density fluctuations computed in the model can further be used as input to the SIGMA radio propagation model in order to better-diagnose scintillation and refractive/diffractive effects.

Core GEMINI model components are implemented in fortran 2008, which interfaces provides for use with C/C++ applications. Fortran code is organized, from the top-level organizational perspective into modules which encapsulate key numerical functionality. These are called from the top level `libgemini` module which provides public interfaces to functionality needed to execute an ionospheric simulation.

A complete module listing with all dependencies is almost impossible to represent via a diagram (such information can be obtained from the cmake build files); instead here we illustrate a small subset of these module connexions to illustrate the overall logic to how the modules are organized. Figure 3 shows the basic flow of the top level programs starting with time step selection and moving through gathering of neutral input data, then solution of electromagnetic equations, plasma transport, and then file output steps. Some of the module dependencies are shown for electromagnetic and multifluid modules specifically in addition to utility modules required by many (if not most) other modules, e.g. interpolation, grid, and basic calculus operations).

A detailed description of the neutral module is shown in Figure 4 to show the complex internal dependencies of just this single program unit. This diagram is effectively one layer of abstraction below that shown in Figure 3. The neutral modules contains submodules for computing density and temperature (`atmos`) along with winds (`wind`) and doing various rotations and transformations to map geographic quantities into geomagnetic. Likewise there are a number of dependencies on utility routines that provide interpolation, etc., along with external connections needed for message passing communication.

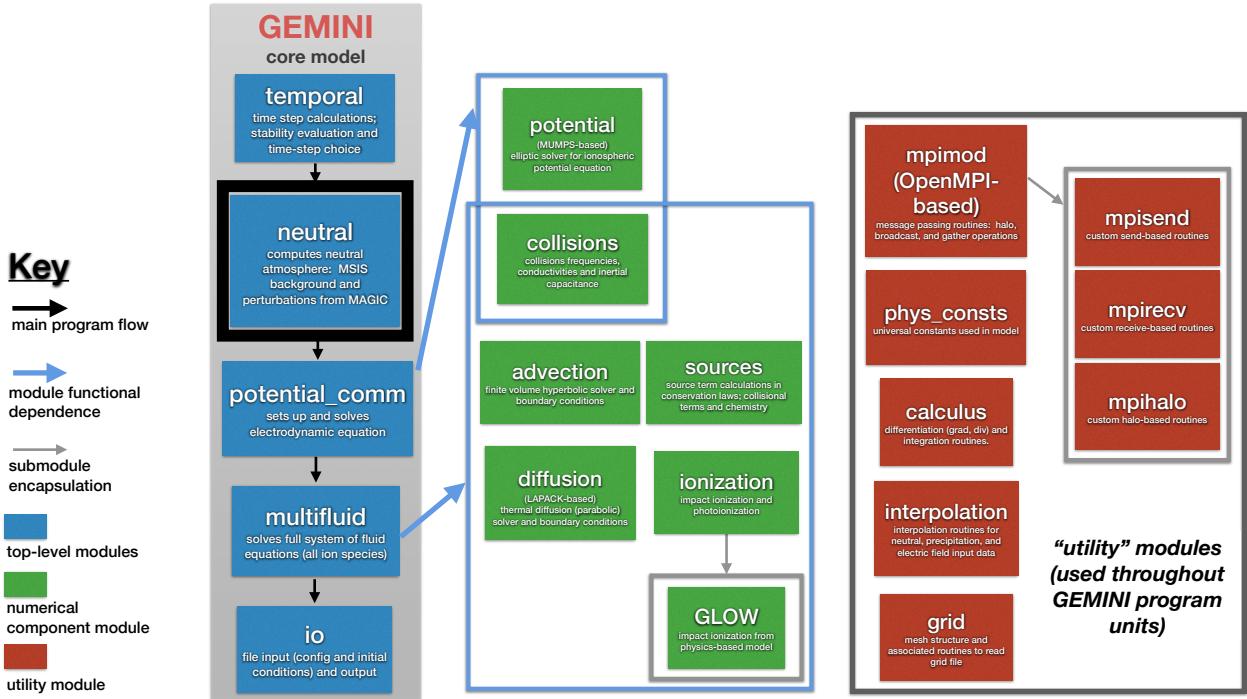


Figure 3: Limited illustration of some of the modules comprising libgemini and the GEMINI ionospheric model.

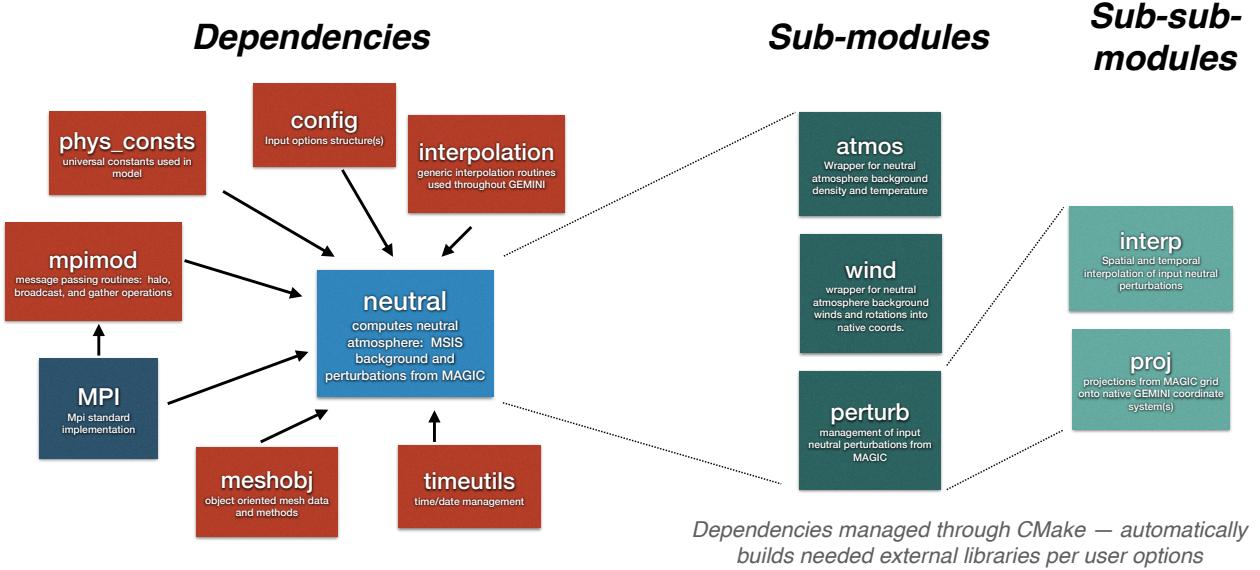


Figure 4: Diagram for neutral module in GEMINI along with associated submodules and dependencies.

5 Cmake Build System for GEMINI

MH - focus on current state and additions made during AIRWaveS

6 GEMINI Data Structures

Until recently data have largely been encapsulated in modules as module global variables that can be accessed from other program units via *use* statements. While this makes for easy implementation for a number of numerical solutions, it has the issue that there can be only one set of data per mpi worker (i.e. each worker has one module “instance”). To address this issue, it makes sense to encapsulate data and associated procedures using fortran object-oriented capabilities. These allow us to better organize our data and pass it simply into processing routines along with the necessary feature that one worker can possess multiple data objects, for our purposes grid and solution data for multiple forestclaw patches.

6.1 Mesh Objects

The use of AMR requires the capability to recompute and reconfigure a mesh arbitrarily during runtime; furthermore, this calculation must not exceed the roughly the computational cost of a typical time step to fully benefit AMR (TA 1.1.2, 1.1.3). Historically GEMINI meshes had been computed by the MATLAB front-end scripting environment (`mat_gemini`).

As part of this project we have also added the ability to compute meshes using `pygemini`, in case these are needed for any pre- or post-processing tasks.

Additionally to address needs of dynamic re-meshing the input grid information is handled differently now. Instead of having the root process read in the grid coordinates, along with all metric factors, unit vectors, and additional information like magnetic field strength, only the coordinate array are passed into the GEMINI core code and then all other quantities are computed from these internally and by each worker or patch instance of the model. This effectively parallelizes the computationally expensive parts of the mesh generation, with testing showing a 10x improvements in performance on a single core (when compared against script mesh generation and, on top of that, almost perfect scaling with the number of cores used for the mesh generation).

There is also the potential that AIRWaveS and future related activities may need additional coordinate systems other than dipole or Cartesian. Thus, during the process of absorbing the mesh generation were into the core GEMINI code we created an object-oriented approach to the organization of mesh data within a given mpi worker (or forestclaw patch). This approach has benefits of making it relatively clean and easy (from a coding point of view) to adapt new mesh types using different coordinate systems. Additionally these objects can be passed to an overlying fortran or C program in order to have numerical operations performed independently on different subdomains of a parallel mesh rather than relying on common blocks or module scope variables to convey mesh

information – which limits the model architecture to one subdomain per `mpi` worker.

Fortran objects and derived types (including polymorphic variables) are passed/returned to C calling programs as `void*` pointers, i.e. as type `void**`. When the C calling program runs libgemini, the `void*` pointer is converted into the correct dynamically typed fortran object through a call which check user supplied object (e.g. grid) type information. There are some fortran language subtleties to how this is done since the C-to-fortran pointer conversion procedure `c_f_pointer` will only work with fortran objects that are statically typed (and not polymorphic). However, the grid object is handled as a polymorphic variable by most procedures to avoid having to write separate functions for different types of grid objects. We work around this in GEMINI by using a function (`set_gridpointer_dyntype`) that included local statically typed pointers to every possible grid type. These *can* be used with `c_f_pointer` and then subsequently bound to a pointer to a polymorphic grid type. This rather subtle workaround is implementation is shown below as it is central to and crucial for the encapsulation of GEMINI within forestclaw.

```
!> set fortran object pointer dynamic type to what is indicated in variable
!>  xtype. Convert C pointer using declared static types (c_f_pointer will
!>  not work on a polymorphic object).
function set_gridpointer_dyntype(xtype,xC) result(x)
    type(c_ptr), intent(in) :: xC
    integer(C_INT), intent(in) :: xtype
    class(curvmesh), pointer :: x
    type(cartmesh), pointer :: xcart
    type(dipolemesh), pointer :: xdipole

    select case (xtype)
        case (1)
            call c_f_pointer(xC,xcart)
            x=>xcart
        case (2)
            call c_f_pointer(xC,xdipole)
            x=>xdipole
        case default
            error stop 'unable to identify object type during conversion &
                       from C to fortran class pointer'
    end select
end function set_gridpointer_dyntype
```

This function can be called to convert the C pointer to the fortran object (and a given type) into a class (polymorphic) pointer of the correct type for internal use with GEMINI, e.g.:

```
x=>set\_\_gridpointer\_\_dyntype(xC,2)
```

would take the C address a dipole grid and convert it into a class pointer with the dynamic type of `dipolemesh`.

6.2 Inputdata Objects

GEMINI has supported the use of a wide variety of input data to specify, e.g., electron precipitation and field-aligned current inputs for auroral simulation, background convection electric field inputs for instability simulations, neutral atmospheric information from a variety of MAGIC grid configurations. For convenience these are interpolated in space and time from some user-specified input time/space grid to GEMINI’s native grid and time basis. Obviously each type of input data must be handled a bit differently; yet they all require sorting, interpolation, and time stepping – thus it convenient and efficient to define a top-level inputdata class and then make extensions for each type of data that contain more specific functionality. This alleviates code duplication and provides a relatively easy way for a programmer to accommodate user-specified inputs for any number of variables used internally in the model.

Details of inputdata objects can be found in `./src/inputdata/` in the `gemini3d` repository; a number of these accommodate passing of neutral data from files or through memory from an interleaved, complementary MAGIC-forest simulation.

6.3 Output Data Formats and Files

MH - h5fortran and output information

7 GEMINI Umbrella Library: `libgemini`

To allow maximum flexibility in how GEMINI functionality is used, we have developed a library containing core functionality that is accessible from applications written in fortran, C, and C++ languages. Traditionally GEMINI had been a monolithic main program with calls to procedures from various modules and submodules in order to access numerical functions. In contrast the code now is separated into `libgemini` which contains all the basic functionality needed to write a GEMINI application, and separate apps which are limited to calls to `libgemini`. This model allows integration into forestclaw. Furthermore mpi components have been fully separated from numerical procedures so that they are swappable with mesh management libraries. While this may seem like a rather mundane change, it required an extensive code refactor and is necessary to improve maintainability, verifiability, and improve the ease-of-use of the model.

There is a need in some applications for GEMINI to accommodate multiple independent sub-domains (with attendant grid and solution data) on a single mpi worker. This precludes use of any module-scope variables as these are limited to one per worker and required a full code refactor so

that grid and solution data are always passed to numerical routines as procedure arguments (and not thru common blocks or module-scope variables).

7.1 C-fortran Interoperability

Core GEMINI numerical functions are written in fortran 2008, while forestclaw uses C/C++. The fortran 2003 standard brought `iso_c_binding` intrinsic module and `bind (C)` procedures to facilitate compilation alongside and use with other languages. This new interface helps to make linking robust across compilers and computing platforms and eliminates some of the traditional problems of a multi-language project like AIRWaveS. Importantly, structures and pointers can be seamlessly passed between languages allowing flexibility in memory allocation and ease of integration of different code components. We have made extensive use of these facilities in Gemini3D specifically to allow ease of co-compiling the model with forestclaw and p4est. Generally speaking GEMINI uses procedures `c_f_pointer` and `c_loc` to translate C and fortran pointers back and forth.

Because many of the forestclaw routines need to be run on individual patch data, GEMINI must be able to accept mesh, solution, and internal variable data for a given patch into model computational routines while also returning opaque “handles” to these objects to the calling C program for passage into further processing routines. Complex internal variables (e.g. fortran structures and objects) are passed from forestclaw into GEMINI via `void` pointers, which since fortran passes data by default by reference, becomes `void**` data types when used from C/C++. An additional complication arises because the datatypes used in GEMINI are polymorphic, e.g. meshes can be of the Cartesian or dipole types, both of which inherit properties and methods from a general curvilinear mesh object. To accommodate this arguments to numerical routines must include grid type information in a primitive format (e.g. integer flag), which can be used by `libgeminis` to “typecast” the `void**` datatype into a class pointer of the correct type for use internally (see source code for more details).

8 trees-GEMINI Application

A GEMINI application has been implemented under the forestclaw API in order to allow adaptive mesh refinement (AMR) using forestclaw’s three-dimensional, extruded mesh capabilities. EMINI applications using ForestClaw / p4est AMR, termed Trees-GEMINI, have recently been developed as part of Phase 1 of DARPA’s AtmoSense program. Our current setup uses GEMINI with an “extruded” mesh, whereby the AMR is done in two dimensions while the full extent of the third dimension is held on each computational “patch” (see Figure 5). In Trees-GEMINI the extruded dimension is along geomagnetic field line which allows us to use existing implicit solvers for the numerically stiff electron thermal conduction process. Such solvers require all data along the parallel coordinate at once to function but are capable of dealing with the fast thermal conduction time scales (in the electron population) while maintaining an otherwise reasonable time step for other explicit solutions for transport. Figure 5 shows a diagram illustrating logical arrangements of coordinates and subdomains organized by mpi worker and patch number.

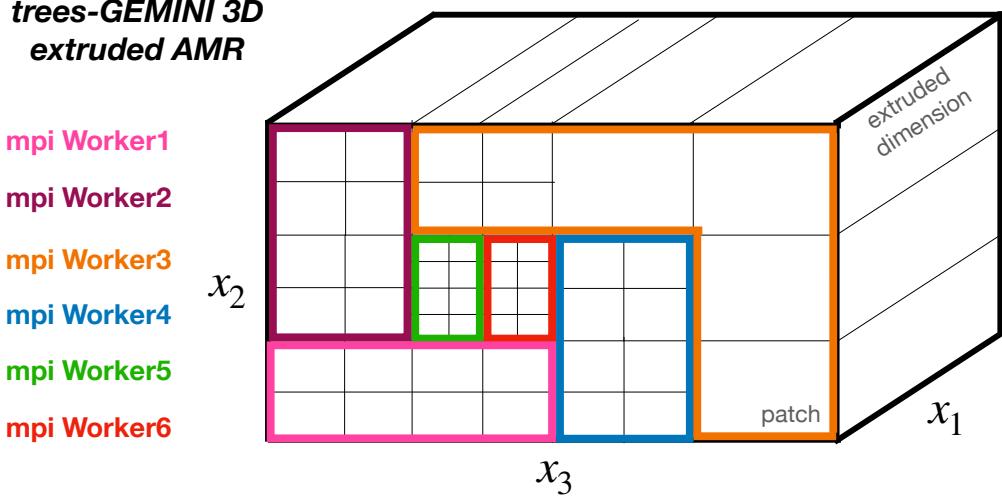


Figure 5: Trees-GEMINI use of forestclaw extruded mesh AMR and parallelization.

Trees-GEMINI currently contains most of the functionality of the core libGEMINI libraries including input data handling, dipole and Cartesian meshes, specifications of boundary conditions, fluid plasma solvers (chemistry, advection, diffusion), photoionization and impact ionization solvers, along with new output routines to optionally generate `vtk/vtu` files for efficient 3D visualization via the Paraview multi-platform open source application (Figure 6 shows an example). It currently does not have a AMR-capable (quasi-static) potential solution in the manner of the core GEMINI library as this functionality was not required by the projects that funded development of Trees-GEMINI. While the MAGIC data are historically input via files, a strategy for running the models side-by-side and exchanging data (one-way) thru memory is currently in development and testing as part of AtmoSense in a form that can, next, be reciprocated from GEMINI to MAGIC.

Figure 6 shows a Trees-GEMINI simulation using multiply refined regions to model low-frequency acoustic wave propagation through the ionosphere using input from the MAGIC atmospheric dynamics model.

8.1 Use of `trees-GEMINI` within the FIGMENTS framework

Trees-GEMINI has been integrated with the FIGMENTS framework which uses p4est and forestclaw functions to search for overlap between two AMR meshes and to interpolate data from one mesh onto another for purposes of model coupling. The advantage of this approach over the traditional file-based data exchange (Figure 2) is that that data exchange is done in memory and during execution of both models. As such this approach is much more efficient than the traditional interfaces and moreover allows AMR data exchange not possible in prior implementations.

Figure 7 shows an example of trees-GEMINI and MAGIC-forest coupled through the FIGMENTS framework. This example uses the Misty Picture source in MAGIC-forest and the results plots show how the shock wave manifests in plasma drifts in the trees-GEMINI model.

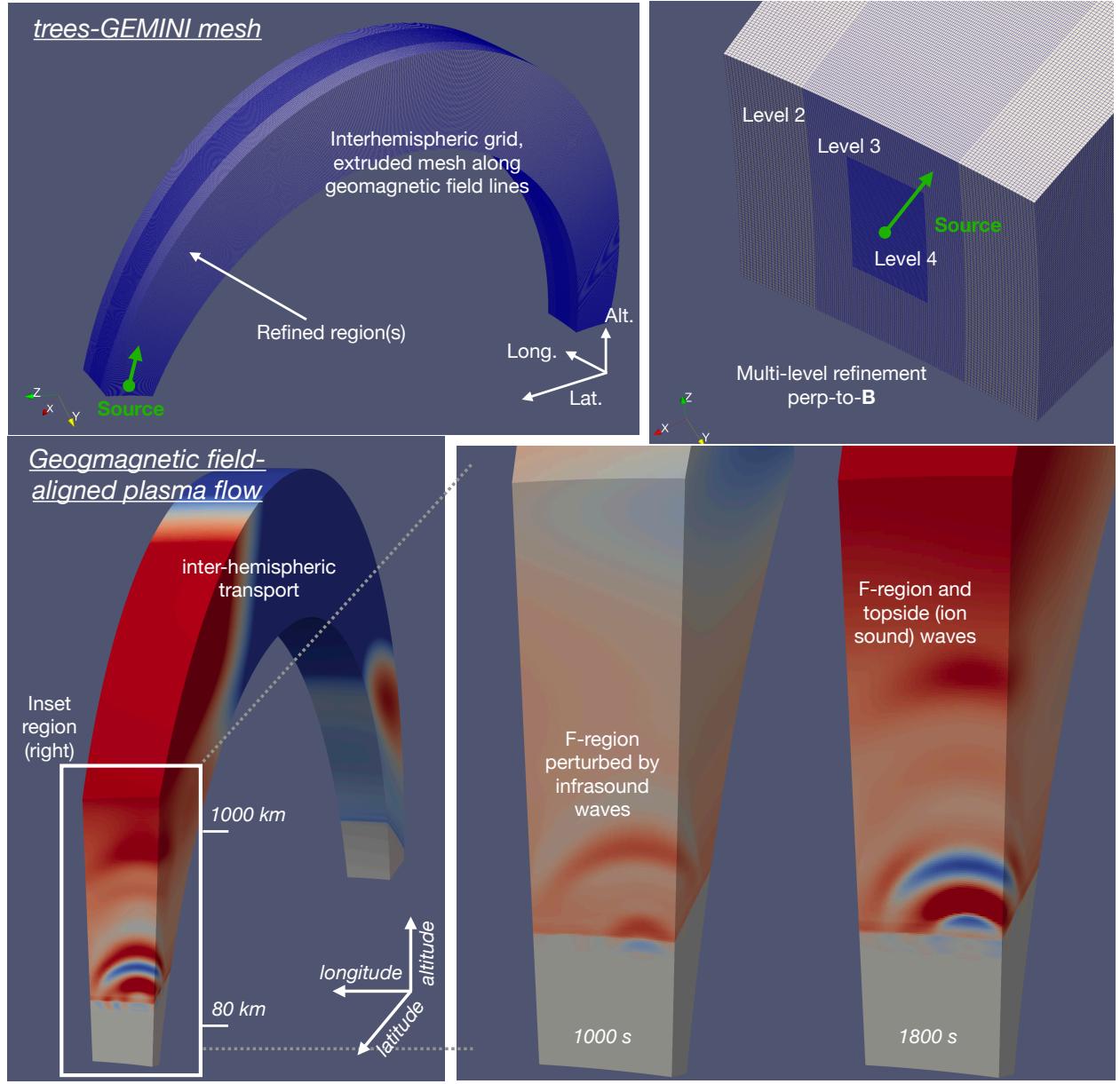


Figure 6: Trees-GEMINI simulation of mHz acoustic waves (AWs) propagating through the ionosphere using input data from MAGIC. (a,b) The overall grid and refinement structures used in the simulation, (c) the configuration of plasma flow, (d) individual output frame illustrating how AW forces ion flows in the F-region and topside.

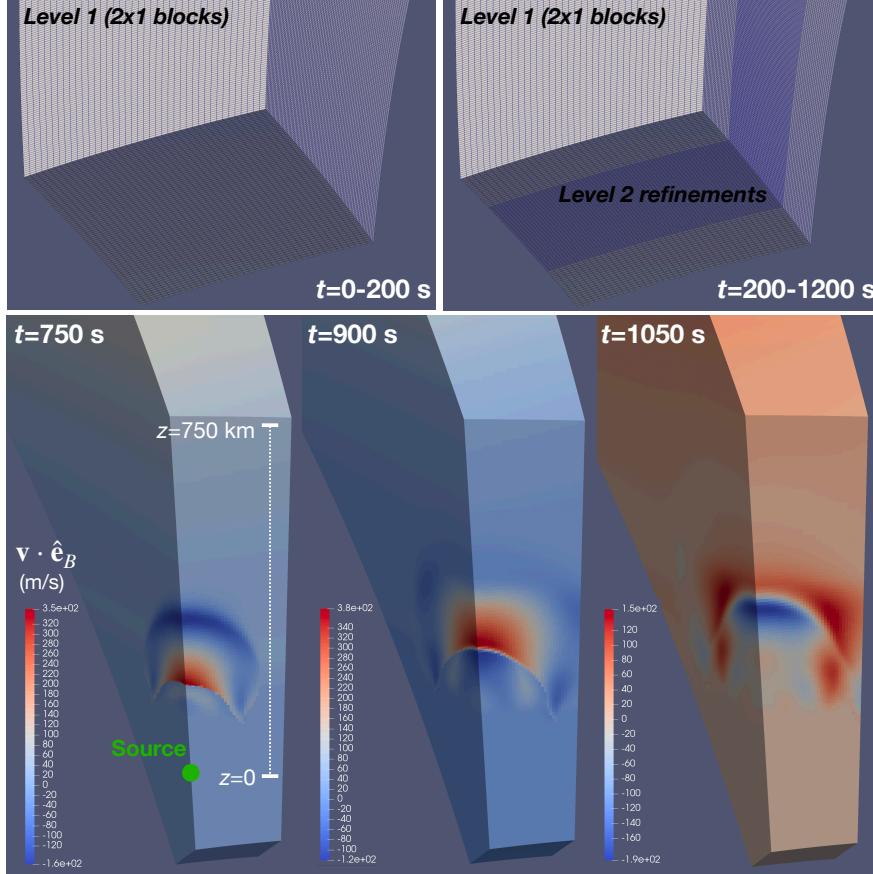


Figure 7: Tree-GEMINI run within the FIGMENTS framework. This simulation includes MAGIC-forest modeling of a source representative of the Mistry Picture Experiment which is then conveyed via forestclaw memory searches to trees-GEMINI. Shown in the diagrams are the mesh refinement in GEMINI (top) along the field-aligned component of the plasma drift velocity when forced by the shock wave.

9 Postprocessing and Visualization Scripts

MH - can you fill in this section explain basic function of pygeminini and mat_gemini?

10 Continuous Integration (CI)

10.1 Automatic CI

The base `gemini3d` repository includes a set of integration and unit tests that can be run by a user upon compilation of model to test a number of model components as well as several integration tests that run simple model configurations and compare results against reference data archived in online repositories. These tests are also run via GitHub Actions anything a new update is pushed to the `gemini3d` repository. These test are designed to be able to run in a matter of minutes on a modest computer.

MH: cmake implementation details, locations of reference data, details of github actions and examples, descriptions of tests

MH - example plots of output?

10.2 Comprehensive Continuous Integration (long CI)

Integration tests done by the automatic CI system are necessarily limited to those which can run quickly; yet there are a huge number of potential applications of the GEMINI model. A more comprehensive set of tests, in the `gemci` repository have been developed in order to check a number of other common use cases of the model against reference data. These include:

- Instability development utilizing periodic grids and quasi-dynamic solvers:
- Auroral electron precipitation and field-aligned currents:
- 2D and 3D cusp, open curvilinear grids with soft electron precipitation and field-aligned currents
- Closed, dipole grids with file input neutral perturbations:

A more complete description of each of these tests is given in the README file in the `gemci` repository under the `cfg` folder. The complete set of long CI tests is designed to be run in about 12 hours on a modest computer system.

MH: cmake implementation details, locations of ref. data

MH - example plots of output?

11 Examples and Comparisons Against GNSS Data

PI - add some of the most recent comparisons you've done during this project? Or a reference to your section of the report?

References

- Blelly, P. and Schunk, R. (1993). A comparative study of the time-dependent standard 8-, 13-and 16-moment transport formulations of the polar wind. In *Annales geophysicae*, volume 11, pages 443–469. Copernicus.
- Burleigh, M., Heale, C., Zettergren, M., and Snively, J. (2018). Modulation of low-altitude ionospheric upflow by linear and nonlinear atmospheric gravity waves. *Journal of Geophysical Research: Space Physics*, 123(9):7650–7667.
- Burleigh, M., Lynch, K., Zettergren, M., and Clayton, R. (2022). Spatiotemporal limitations of data-driven modeling: An isinglass case study. *Journal of Geophysical Research: Space Physics*, 127(9):e2021JA030242.
- Burleigh, M. and Zettergren, M. (2017). Anisotropic fluid modeling of ionospheric upflow: Effects of low-altitude anisotropy and thermospheric winds. *Journal of Geophysical Research: Space Physics*, 122(1):808–827.
- Burleigh, M., Zettergren, M., Lynch, K., Lessard, M., Moen, J., Clausen, L., Kenward, D., Hysell, D., and Liemohn, M. (2019). Transient ionospheric upflow driven by poleward moving auroral forms observed during the rocket experiment for neutral upwelling 2 (renu2) campaign. *Geophysical Research Letters*, 46(12):6297–6305.
- Clayton, R., Burleigh, M., Lynch, K. A., Zettergren, M., Evans, T., Grubbs, G., Hampton, D. L., Hysell, D., Kaeppler, S., Lessard, M., et al. (2021). Examining the auroral ionosphere in three dimensions using reconstructed 2d maps of auroral data to drive the 3d gemini model. *Journal of Geophysical Research: Space Physics*, 126(11):e2021JA029749.
- Deshpande, K. B. and Zettergren, M. D. (2019). Satellite-beacon ionospheric-scintillation global model of the upper atmosphere (sigma) iii: Scintillation simulation using a physics-based plasma model. *Geophysical Research Letters*, 46(9):4564–4572.
- Díaz Peña, J., Semeter, J., Nishimura, Y., Varney, R., Reimer, A., Hairston, M., Zettergren, M., Hirsch, M., Verkhoglyadova, O., Hosokawa, K., et al. (2021). Auroral heating of plasma patches due to high-latitude reconnection. *Journal of Geophysical Research: Space Physics*, 126(12):e2021JA029657.
- Diloy, P.-Y., Robineau, A., Lilenstein, J., Blelly, P.-L., and Fontanari, J. (1996). A numerical model of the ionosphere, including the e-region above eiscat. In *Annales Geophysicae*, volume 14, pages 191–200. Springer.
- Fang, X., Randall, C. E., Lummerzheim, D., Solomon, S. C., Mills, M. J., Marsh, D. R., Jackman, C. H., Wang, W., and Lu, G. (2008). Electron impact ionization: A new parameterization for 100 ev to 1 mev electrons. *Journal of Geophysical Research: Space Physics*, 113(A9).
- Huba, J., Joyce, G., and Fedder, J. (2000). Sami2 is another model of the ionosphere (sami2): A new low-latitude ionosphere model. *Journal of Geophysical Research: Space Physics*, 105(A10):23035–23053.

- Inchin, P., Snively, J., Kaneko, Y., Zettergren, M., and Komjathy, A. (2021). Inferring the evolution of a large earthquake from its acoustic impacts on the ionosphere. *AGU Advances*, 2(2):e2020AV000260.
- Inchin, P., Snively, J., Zettergren, M., Komjathy, A., Verkhoglyadova, O., and Tulasi Ram, S. (2020). Modeling of ionospheric responses to atmospheric acoustic and gravity waves driven by the 2015 nepal 7.8 gorkha earthquake. *Journal of Geophysical Research: Space Physics*, 125(4):e2019JA027200.
- Lamarche, L. J., Deshpande, K. B., and Zettergren, M. D. (2022). Observations and modeling of scintillation in the vicinity of a polar cap patch. *Journal of Space Weather and Space Climate*, 12:27.
- Lynch, K., Hampton, D., Zettergren, M., Bekkeng, T., Conde, M., Fernandes, P., Horak, P., Lessard, M., Miceli, R., Michell, R., et al. (2015). Mica sounding rocket observations of conductivity-gradient-generated auroral ionospheric responses: Small-scale structure with large-scale drivers. *Journal of Geophysical Research: Space Physics*, 120(11):9661–9682.
- Mitchell, H., Fedder, J., Keskinen, M., and Zalesak, S. (1985). A simulation of high latitude f-layer instabilities in the presence of magnetosphere-ionosphere coupling. *Geophysical research letters*, 12(5):283–286.
- Perry, G., Dahlgren, H., Nicolls, M., Zettergren, M., St.-Maurice, J.-P., Semeter, J., Sundberg, T., Hosokawa, K., Shiokawa, K., and Chen, S. (2015). Spatiotemporally resolved electrodynamic properties of a sun-aligned arc over resolute bay. *Journal of Geophysical Research: Space Physics*, 120(11):9977–9987.
- Richards, P., Fennelly, J., and Torr, D. (1994). Euvac: A solar euv flux model for aeronomics calculations. *Journal of Geophysical Research: Space Physics*, 99(A5):8981–8992.
- Schunk, R. (1977). Mathematical structure of transport equations for multispecies flows. *Reviews of Geophysics*, 15(4):429–445.
- Solomon, S. C. (2001). Auroral particle transport using monte carlo and hybrid methods. *Journal of Geophysical Research: Space Physics*, 106(A1):107–116.
- Solomon, S. C. and Qian, L. (2005). Solar extreme-ultraviolet irradiance for general circulation models. *Journal of Geophysical Research: Space Physics*, 110(A10).
- Spicher, A., Deshpande, K., Jin, Y., Oksavik, K., Zettergren, M. D., Clausen, L. B., Moen, J. I., Hairston, M. R., and Baddeley, L. (2020). On the production of ionospheric irregularities via kelvin-helmholtz instability associated with cusp flow channels. *Journal of Geophysical Research: Space Physics*, 125(6):e2019JA027734.
- St-Maurice, J.-P. and Laneville, P. (1998). Reaction rate of o+ with o₂, n₂, and no under highly disturbed auroral conditions. *Journal of Geophysical Research: Space Physics*, 103(A8):17519–17521.
- Swoboda, J., Semeter, J., Zettergren, M., and Erickson, P. J. (2017). Observability of ionospheric space-time structure with isr: A simulation study. *Radio Science*, 52(2):215–234.

- Zettergren, M., Lynch, K., Hampton, D., Nicolls, M., Wright, B., Conde, M., Moen, J., Lessard, M., Miceli, R., and Powell, S. (2014). Auroral ionospheric f region density cavity formation and evolution: Mica campaign results. *Journal of Geophysical Research: Space Physics*, 119(4):3162–3178.
- Zettergren, M. and Semeter, J. (2012). Ionospheric plasma transport and loss in auroral downward current regions. *Journal of Geophysical Research: Space Physics*, 117(A6).
- Zettergren, M., Semeter, J., and Dahlgren, H. (2015). Dynamics of density cavities generated by frictional heating: Formation, distortion, and instability. *Geophysical Research Letters*, 42(23).
- Zettergren, M. and Snively, J. (2013). Ionospheric signatures of acoustic waves generated by transient tropospheric forcing. *Geophysical Research Letters*, 40(20):5345–5349.
- Zettergren, M. and Snively, J. (2015). Ionospheric response to infrasonic-acoustic waves generated by natural hazard events. *Journal of Geophysical Research: Space Physics*, 120(9):8002–8024.
- Zettergren, M. and Snively, J. (2019). Latitude and longitude dependence of ionospheric tec and magnetic perturbations from infrasonic-acoustic waves generated by strong seismic events. *Geophysical Research Letters*, 46(3):1132–1140.
- Zettergren, M., Snively, J., Komjathy, A., and Verkhoglyadova, O. (2017). Nonlinear ionospheric responses to large-amplitude infrasonic-acoustic waves generated by undersea earthquakes. *Journal of Geophysical Research: Space Physics*, 122(2):2272–2291.