Space Weather Modeling Framework
CCMC Workshop, Oct. 11-14, 2005

D. De Zeeuw, T. Gombosi, G. Toth
University of Michigan

Outline

- Why frameworks?
- Space Weather Modeling Framework
- Component replacement example
- Component models
- Performance
- ESMF compatibility

From Codes To Framework

- The Sun-Earth system consists of many different interconnecting domains that are independently modeled.
- Each physics domain model is a separate application, which has its own optimal mathematical and numerical representation.
- Our goal is to integrate models into a flexible software framework.
- The framework incorporates physics models with minimal changes.
- The framework can be extended with new components.
- The performance of a well designed framework can supercede monolithic codes or ad hoc couplings of models.

SWMF Design Goals

Design Goals

- Couple computational physics modules with only modest code modifications
- Achieve good parallel performance
- Make the SWMF as versatile as possible
- User friendly and consistent interface

Design Challenges

- Serial and parallel models
- Models usually developed for stand-alone execution
- I/O can conflict with other models
- Non-OO coding, data and procedure name conflicts can occur
- No checkpoint and restart capabilities
- Efficient coupling of arbitrary grid structures and data decompositions not easily achieved

Design Decisions (1)

Single executable: advantages

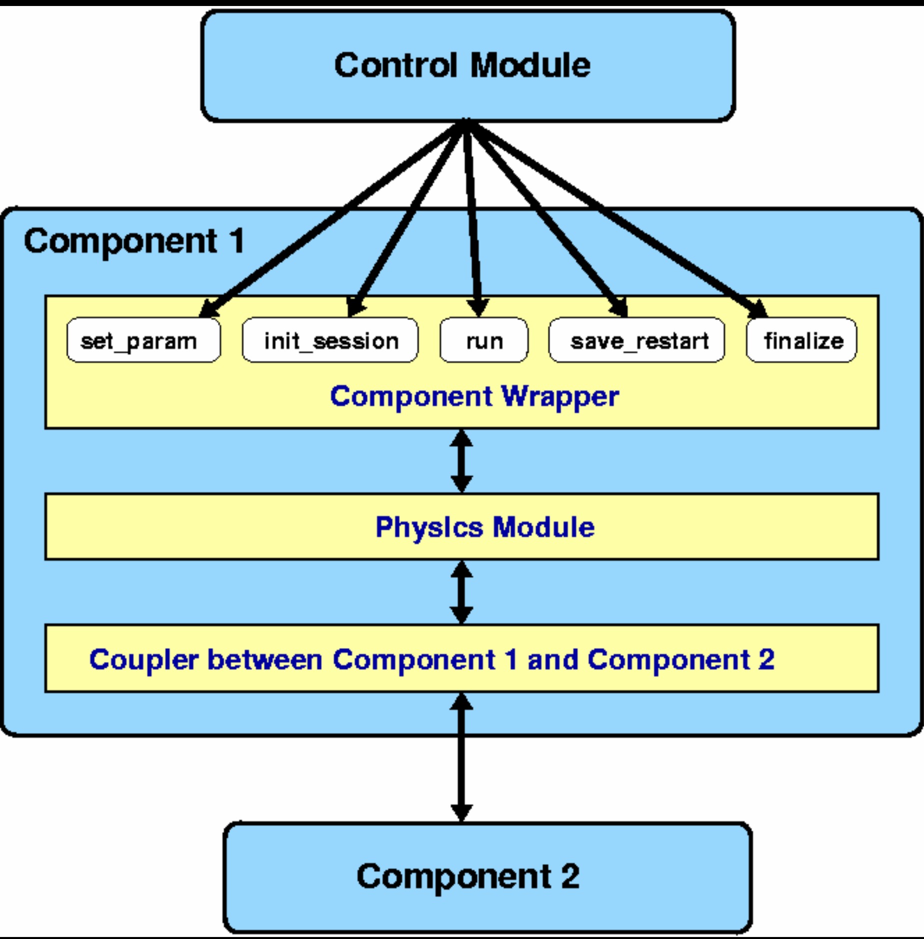
- Simpler control of execution and coupling
- I/O can be better controlled
- uses standard MPI interface which is widely available
- easily share common libraries
- better code organization (good software design)
- synchronized module stopping and restart provides more reliability

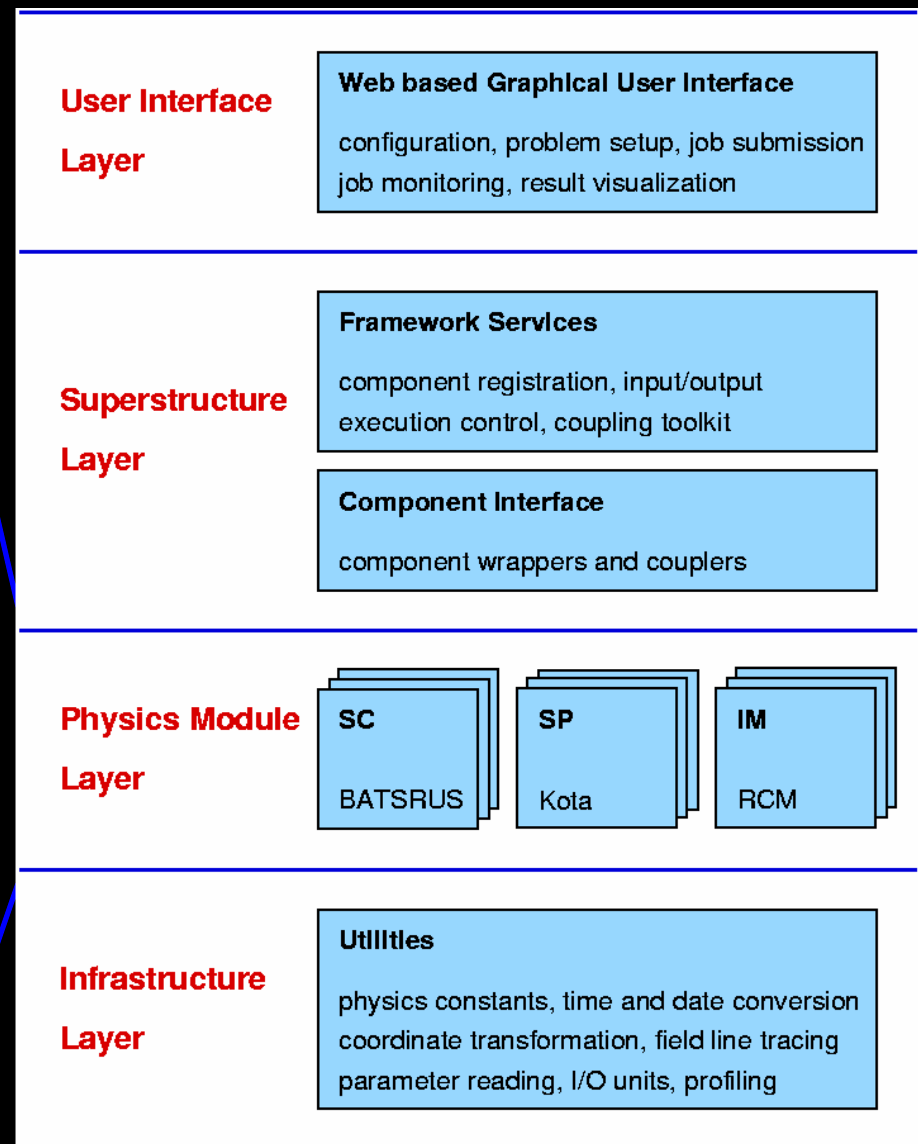
Design Decisions (2)

Single executable: disadvantages

- memory requirements for single executable
  - mitigated by strategic use of allocatable memory and empty libraries for unused components
- not suited to grid computing
  - grid computing environments are still not commonplace
  - can be adapted to grid computing with little extra effort
- complicated and lengthy compilation
  - each component has its own Makefile to build library to link with SWMF
- additional work to change code
  - resulting code is much more flexible
- may have data and procedure name conflicts
  - we provide Perl scripts which find these conflicts and rename as appropriate

The SWMF Architecture





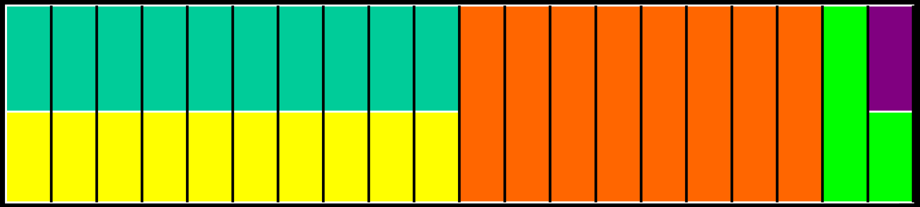
SWMF Component

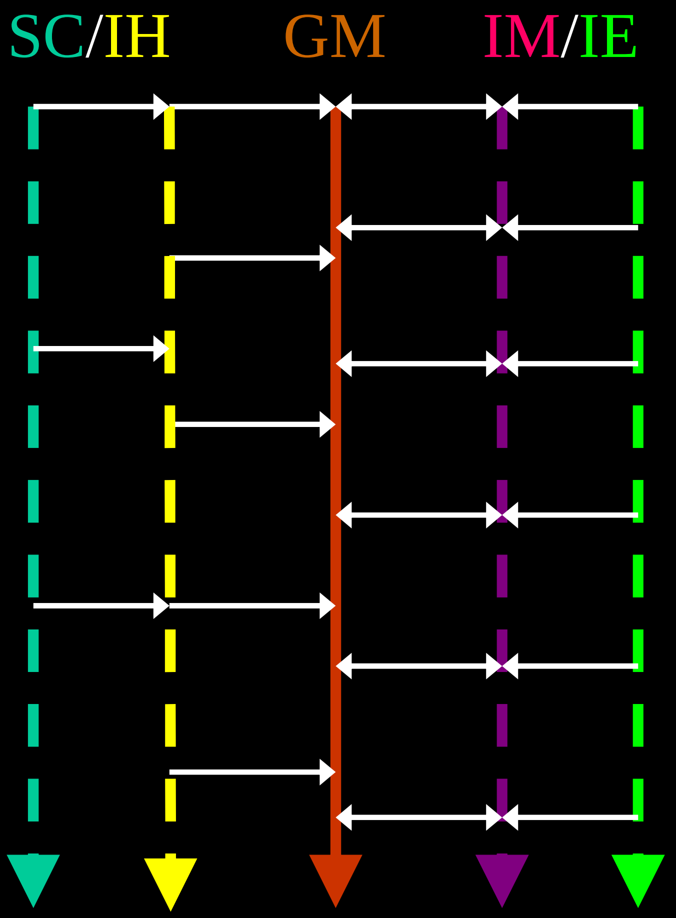
- A SWMF Component is made up of three pieces
  - Physics module
    - ★ must use MPI and arbitrary MPI communicators
    - ★ compile as a library for linking to other executable
    - ★ all I/O redirected to unique subdirectories
    - ★ be portable to wide variety of computer platforms
    - ★ provide a test suite for model verification
    - ★ provide appropriate documentation
  - Wrapper
    - ★ communicates with the high-level Control Module (CON)
    - ★ provides version name/number and grid description to CON
    - ★ accept parallel configuration and input parameters from CON
    - ★ initialize data, execute time step, save/read restart files, finalize run
  - Coupling interface
    - ★ exchange information with other components
    - ★ all data coupled in SI units for consistency

Parallel Layout and Execution

LAYOUT.in for 20 PE-s

- ID ROOT LAST STRIDE
- #COMPONENTMAP
- SC 0 9 1
- IH 0 9 1
- GM 10 17 1
- IE 18 19 1
- IM 19 19 1
- #END





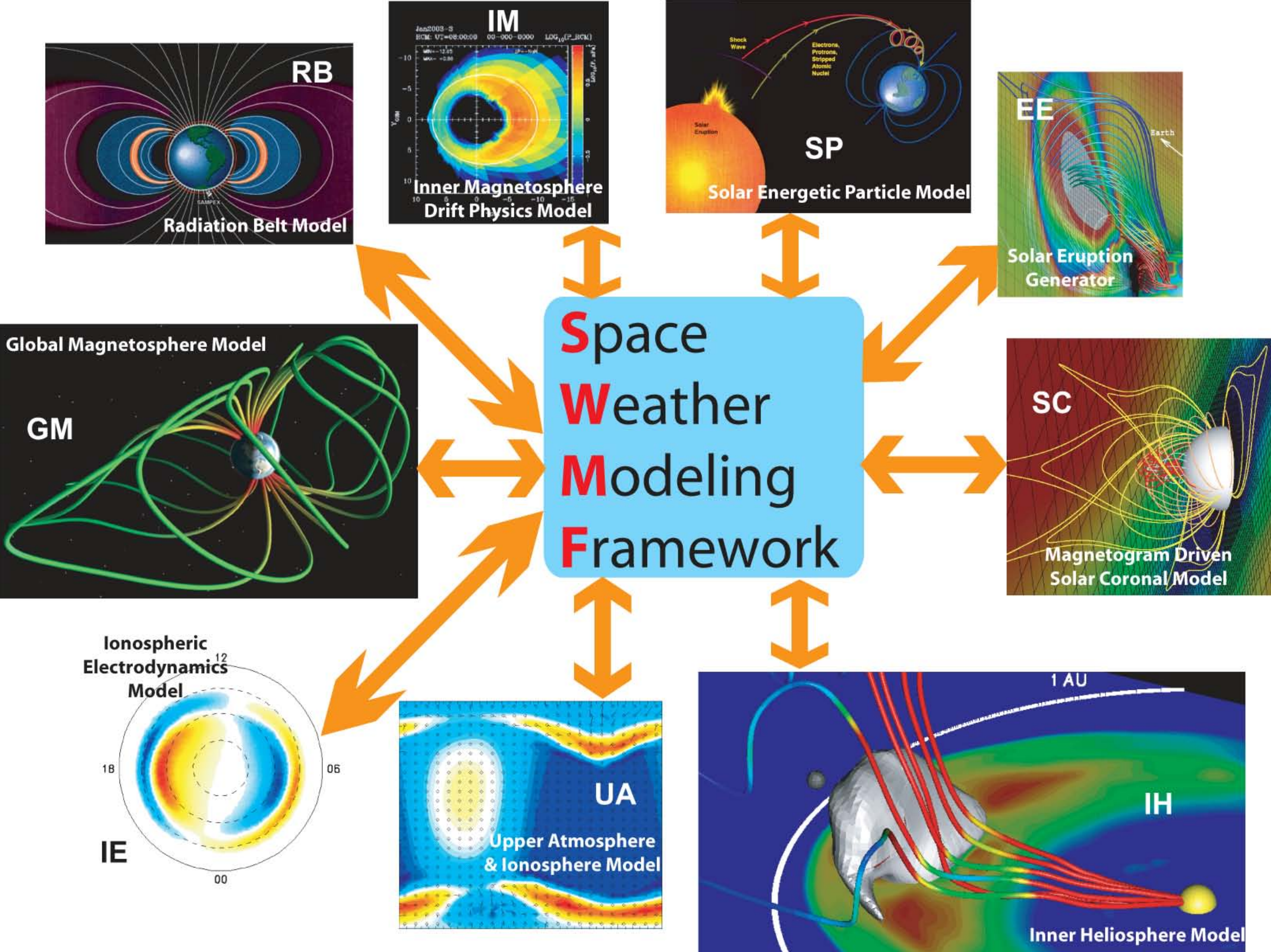
SWMF Code Summary

- SWMF contains the following code:
  - 200,000 lines in the physics modules
  - 23,000 lines in the core SWMF libraries and CON
  - 17,000 lines of IDL plotting scripts
  - 10,000 lines in the wrappers and couplers (mostly GM)
  - 8,000 lines of Perl and shell scripts for installation, configuration, data conversion, parameter checking, etc.
- SWMF comes with well documented user manual, sample output, and testing procedures
- SWMF runs on any Unix/Linux based system with a Fortran 90 compiler, MPI library, and Perl interpreter
  - It can run on a laptop with one or two components and scales well to several hundreds of processors of the world's fastest supercomputers (like Columbia) with all components running together.

Swapping SWMF Components

Example: IM module swap

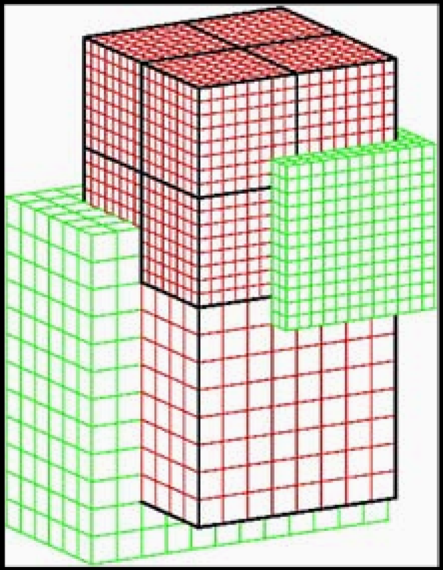
- Restructure code to meet SWMF component requirements
  - Time: depends on model itself, likely 1-2 weeks including testing
- IM wrapper
  - 524 lines
  - Fill exchange buffers, unit conversion, and handle instructions given by CON for stepping, saving, plotting, etc.
  - Time: ~2 days, template exists, details change
- IM coupler
  - coupling to IE: 178 lines
  - coupling to GM: 323 lines
  - Uses coupling toolkit included in SWMF
  - Time: 1 hour - Couplers would be almost completely reusable



Space
Weather
Modeling
Framework
RB
Radiation Belt Model
SAMPEX
IM
Inner Magnetosphere
Drift Physics Model
SP
Shock Wave
Electrons, Protons, Stripped Atomic Nuclei
Solar Eruption
Solar Energetic Particle Model
EE
Earth
Solar Eruption
Generator
Global Magnetosphere Model
GM
SC
Magnetogram Driven
Solar Coronal Model
Ionospheric
Electrodynamics
Model
12
18
06
00
IE
UA
Upper Atmosphere
& Ionosphere Model
1 AU
IH
Inner Heliosphere Model

MHD Code: BATSRUS (SC/EE/IH/GM)

- Block Adaptive Tree Solar-wind Roe-type Upwind Scheme
  - Conservative finite-volume method
  - Shock-capturing Total Variation Diminishing schemes
  - Explicit, implicit & explicit/implicit time stepping
  - Semi-relativistic MHD equations
  - Splitting the magnetic field into $\mathbf{B}_0 + \mathbf{B}_1$
  - Various methods to control the divergence of $\mathbf{B}$

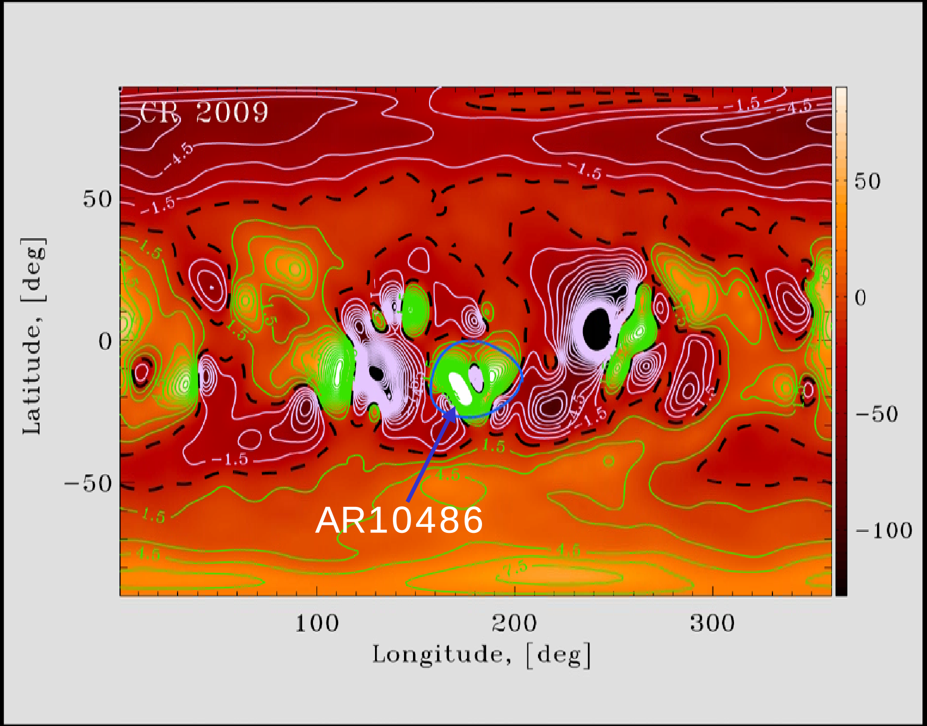


- AMR & data structure
  - Adaptive self-similar blocks
  - Octree data structure
- Parallel implementation
  - Fortran-90 with MPI
  - Near-perfect scaling to >1500 PEs (explicit time-stepping)
  - Good scaling to ~256 PEs with implicit time-stepping
  - Highly portable (SGI Origin, Altix, Compaq, PC clusters, X-serve clusters, etc)

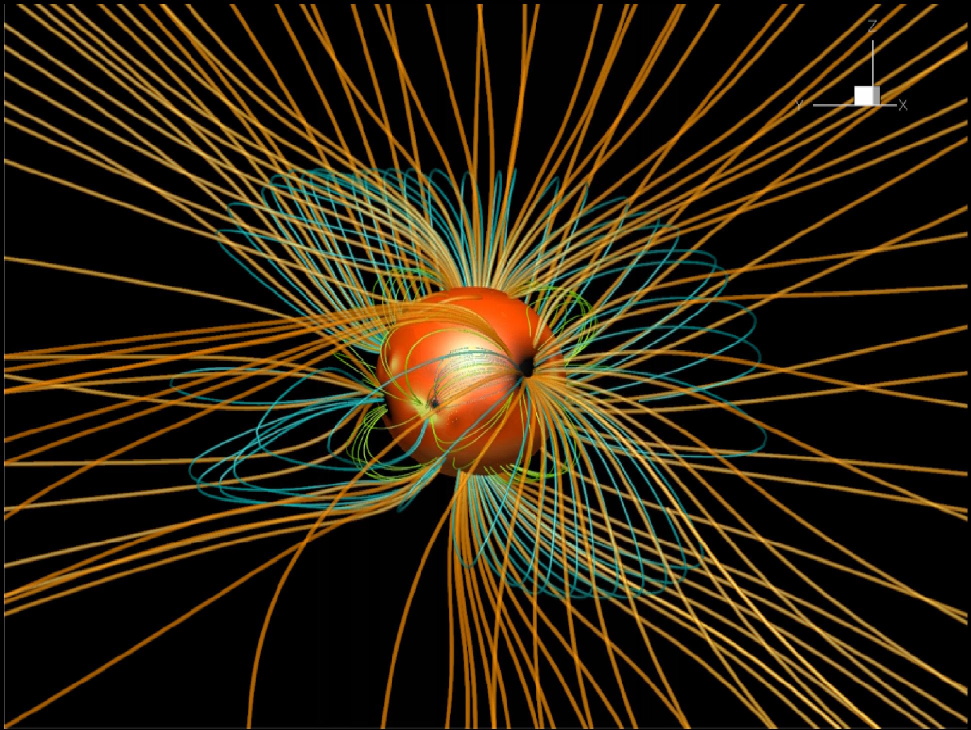
CENTER FOR SPACE ENVIRONMENT MODELING
CSEM

Full-disk Magnetograms From MDI Drive Simulations (SC Module)

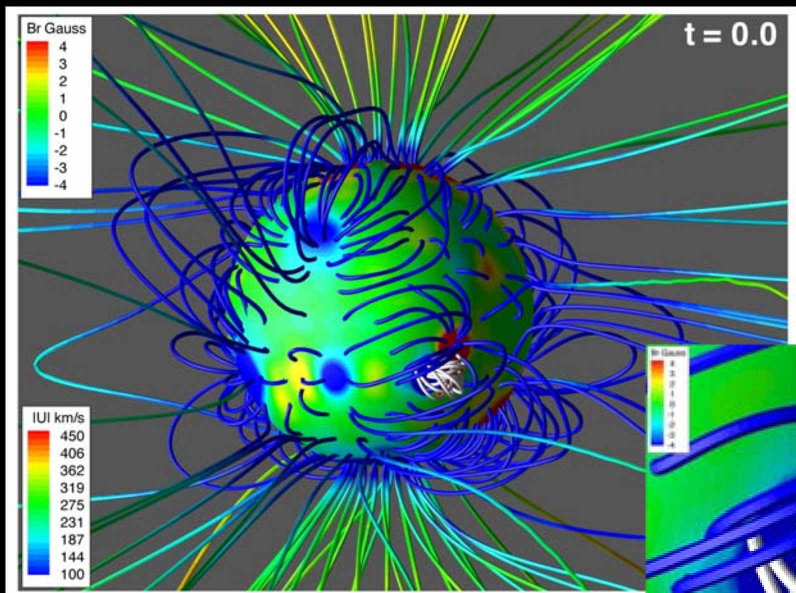
Magnetic map from MDI on Oct 27 used in MHD computations (n=90)



Computed coronal magnetic field at steady-state with solar wind

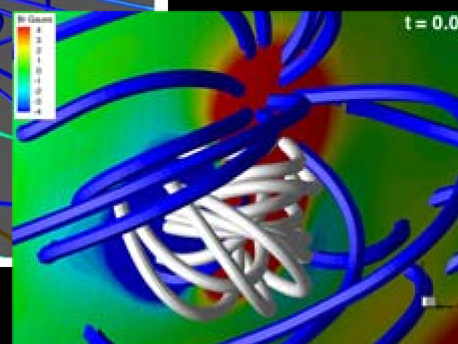


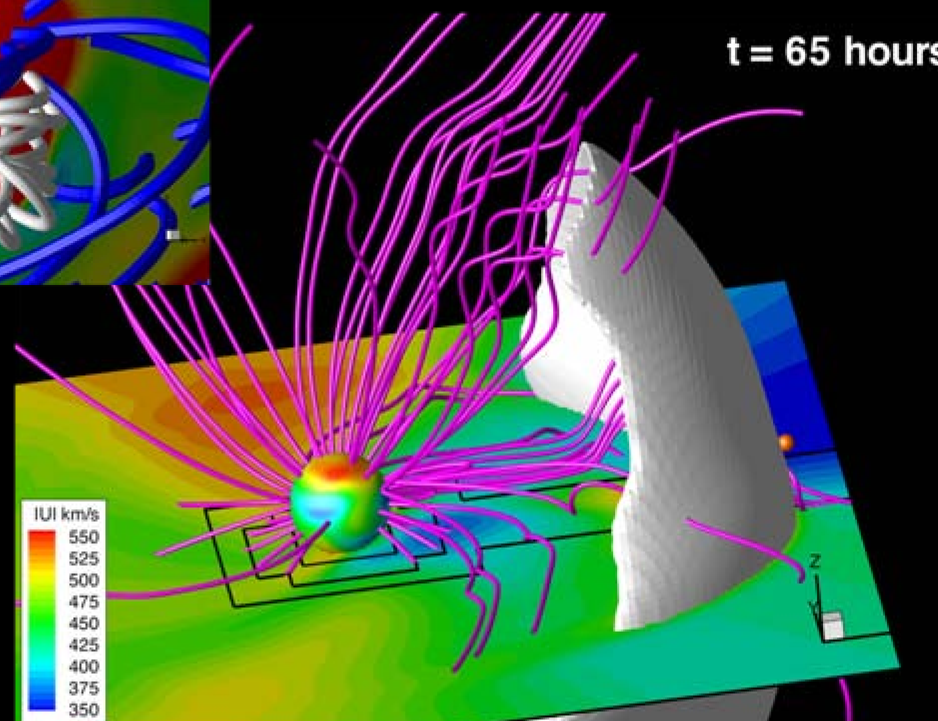
CME Generation



Background field is obtained from
- MDI magnetograms

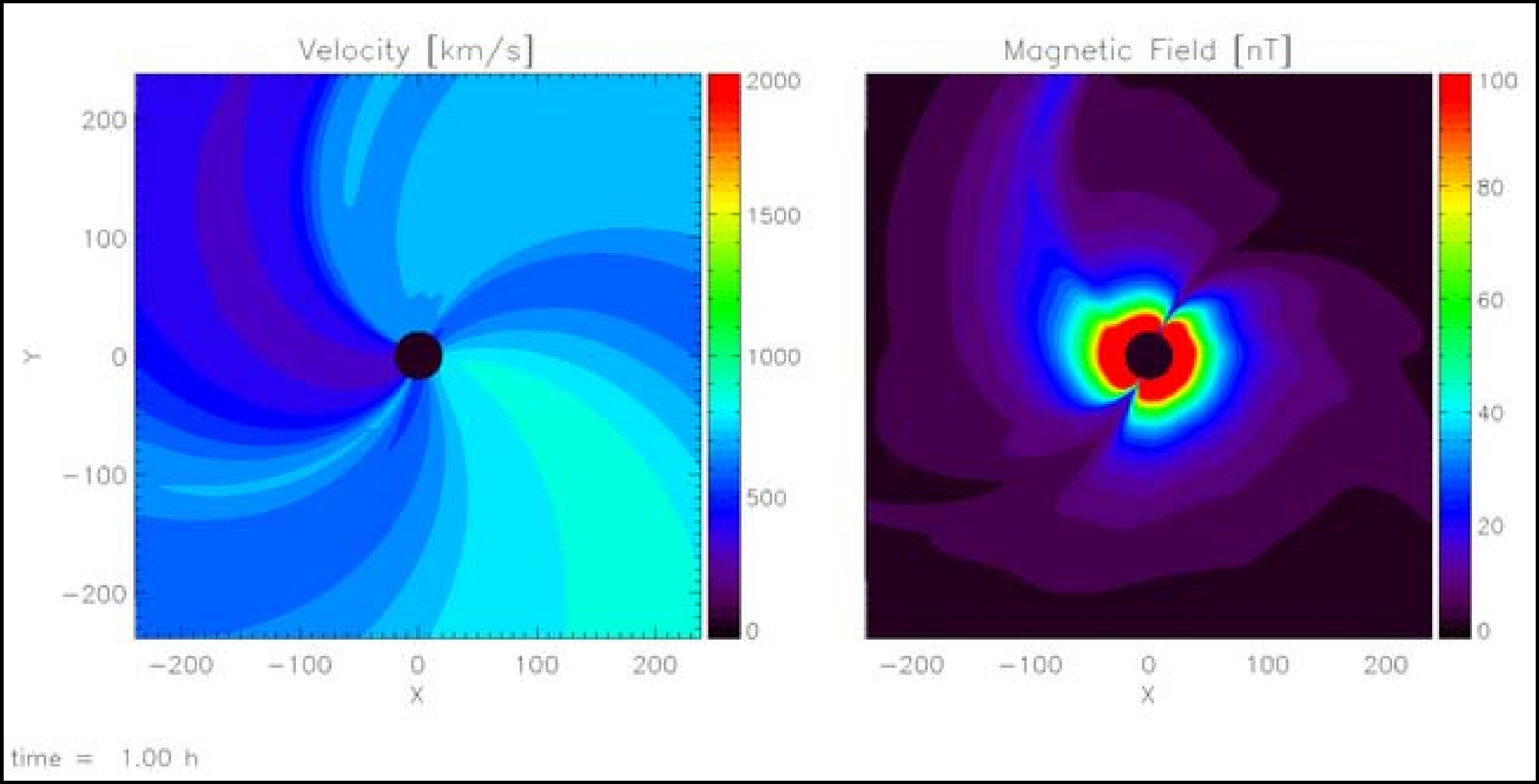
EE generator initiates CME
- Gibson-Low flux tube
- Shearing and rotating (flux emergence)
- Titov-Demulin method





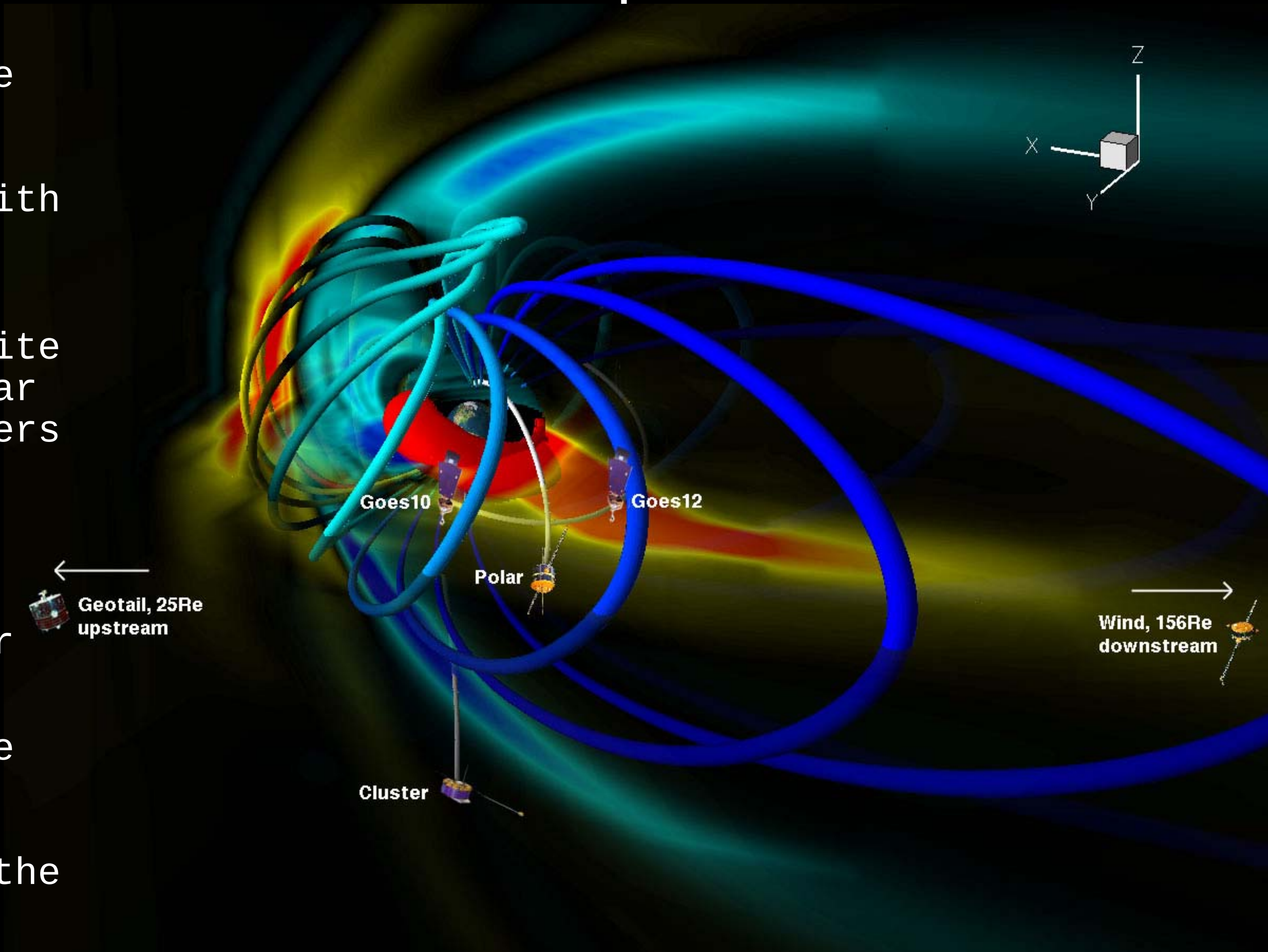
Eruption propagates through the corona (SC) and inner heliosphere (IH), including SEP acceleration (SP) and interaction with Earth's magnetosphere (GM).

Two interacting CMEs (Oct. 26 & 28, 2003)



October 29-30, 2003 Halloween Storm Simulation with GM, IM and IE Components

- The magnetosphere during the solar storm associated with an X17 solar eruption.
- Using satellite data for solar wind parameters
- Solar wind speed: 1800 km/s.
- Time: October 29, 0730UT
- Shown are the last closed field lines shaded with the thermal pressure

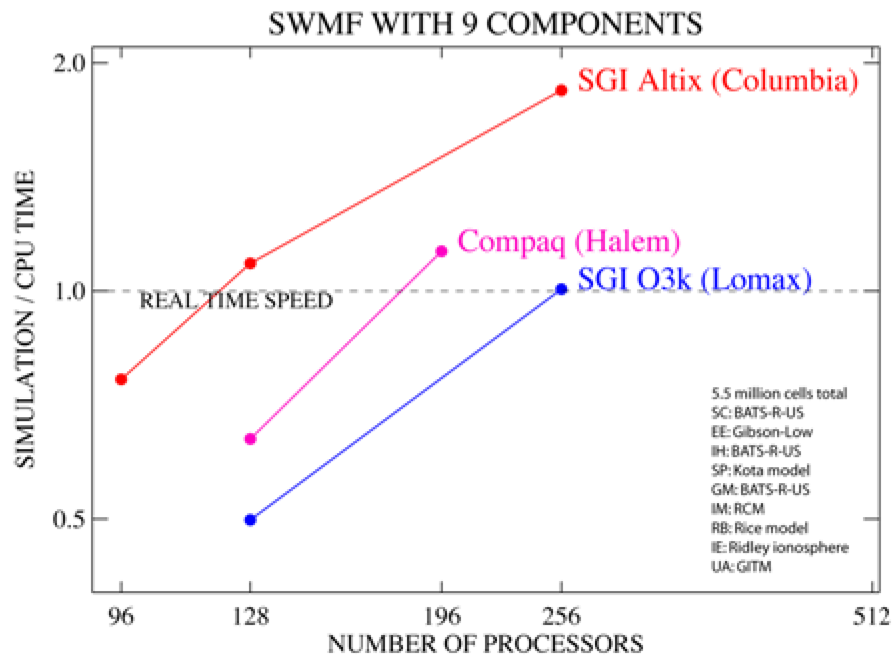


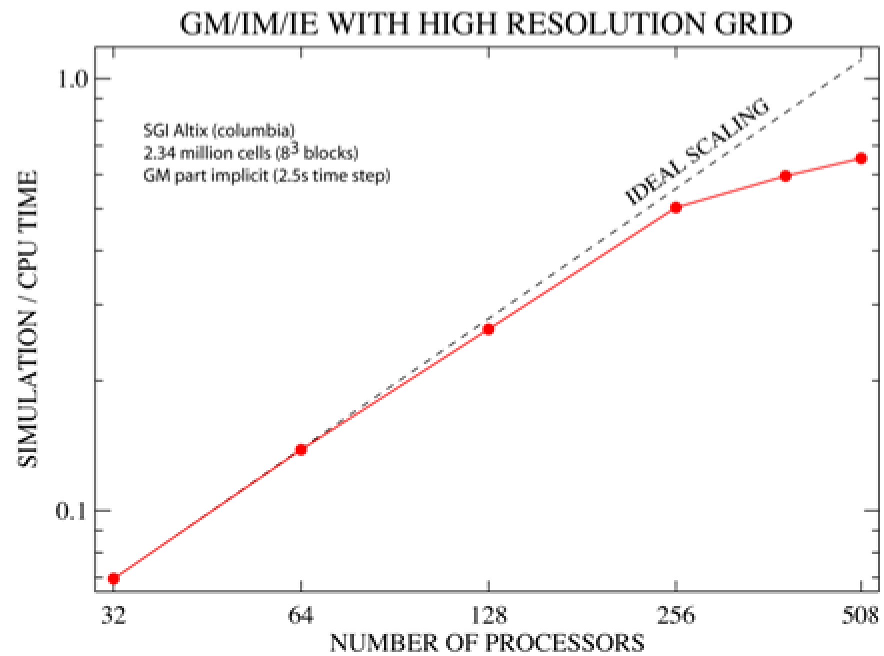
GITM (UA) - TIMED GUVI Comparison

- Here we compare the electron density at a given altitude to the 1304 filter from GUVI.
- The top plot is the 3D model results at the given time indicated by the vertical stripe in the third plot.
- The second plot is the electron density at the specified altitude for the swath of GUVI. Each measurement is taken at the GUVI measurement (in space and time).
- The third plot shows the model output as a function of time, while the fourth plot shows the GUVI results as a function of time.

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

SWMF Performance

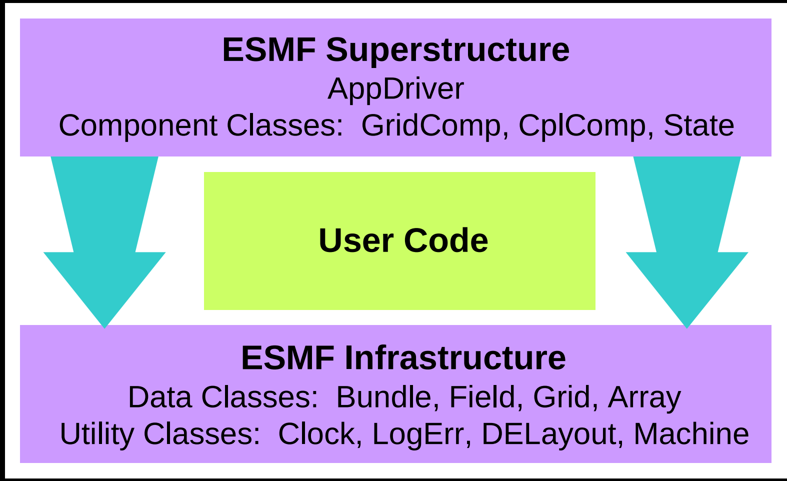




What is ESMF?

1. ESMF provides tools for turning model codes into components with standard interfaces and standard drivers

2. ESMF provides data structures and common utilities that components use
   i. to organize codes
   ii. to improve performance portability
   iii. for common services such as data communications, regridding, time management and message logging



ESMF-SWMF Coupling Progress

- The SWMF can be compiled into a library.
- It can run as a stand alone or it can be run from another application.
- It can get the MPI communicator from another application.
- It stops without calling MPI_FINALIZE when not in a stand alone mode.
- The SWMF can be run by an ESMF compatible driver.
- Exchange of information with other ESMF components is in progress.

Conclusions

- The SWMF Works!
- It meets the design goals:
  - Performance, versatility, consistent interface, and modest code modifications to incorporate new component
- It is being used regularly for scientific investigations (especially the Halloween Storm simulations)
- CCMC has the SWMF and is working with it.
- SWMF Paper in press (JGR)
  - http://csem.engin.umich.edu/CSEM/Publications/Toth2005_swmf.pdf