# CISM Code Coupling Strategy and Plans

C. Goodrich

for the CISM Science and
Code Coupling Teams

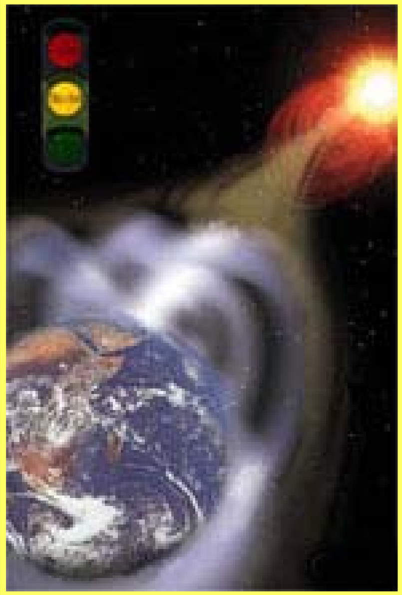
# Why do we need Frameworks – The Challenge of Scales

Spatial scales

- 1 AU = 215 Rs = $10^{4.5}$ Re
- Smallest scales: $\sim 10^{-4}$ Re
- Scale span: 8.5 orders of magnitude ($\sim 2^{28}$)

Temporal scales

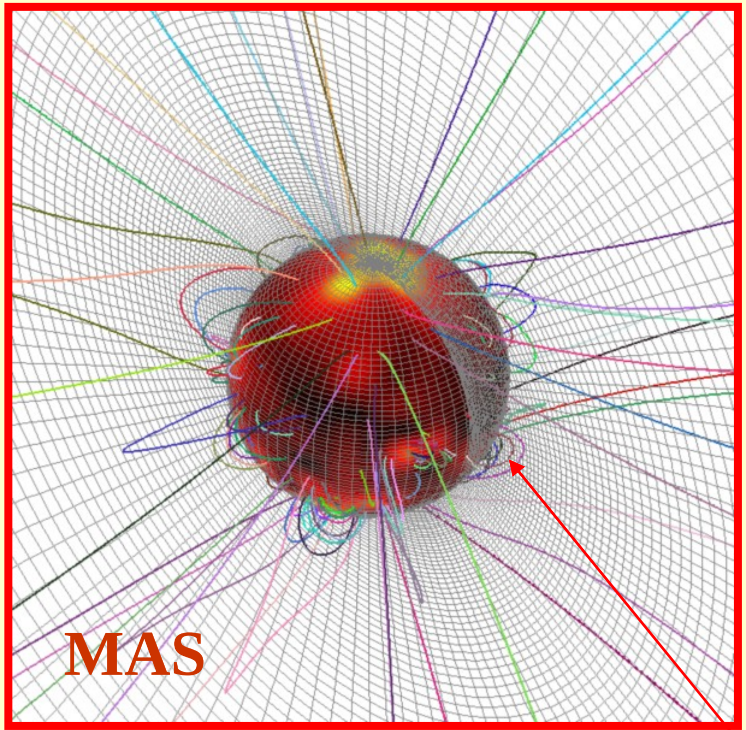
- Solar cycle: $\sim 10^{8.5}$ s
- Solar wind travel time to 1 AU: $\sim 10^{5.5}$ s
- SEP travel time: $\sim 10^{3.5}$ s
- Alfvén wave travel time in the ionosphere: $\sim 10^{0}$ s
- Scale span: 8.5 orders of magnitude ($\sim 2^{28}$)

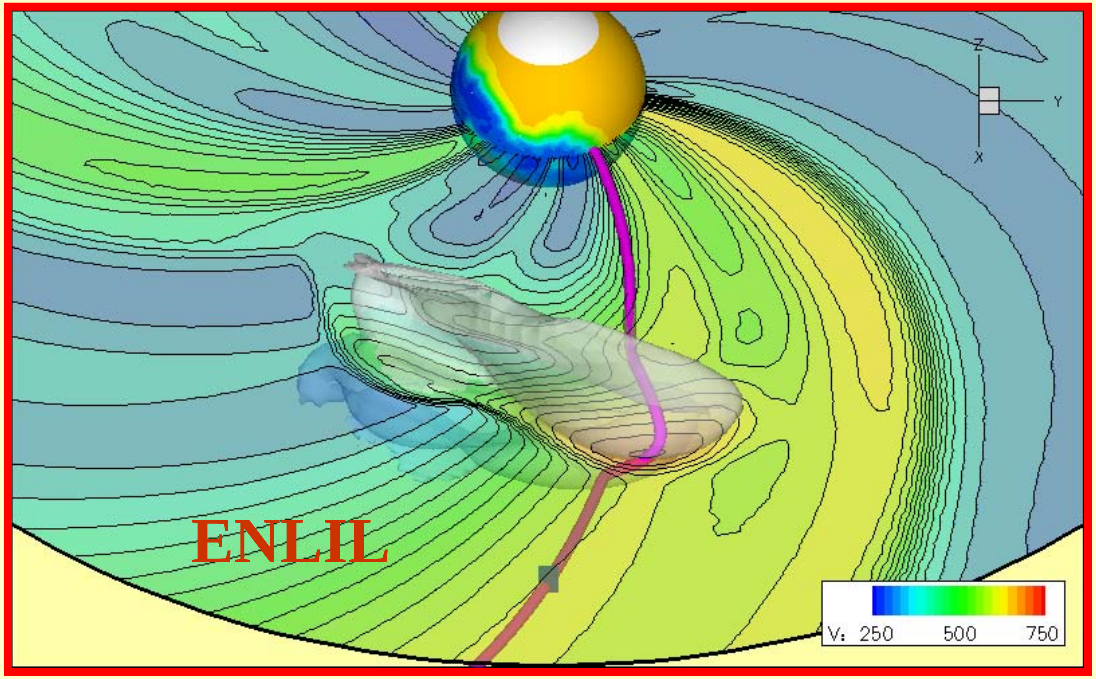


# From Codes To Framework

- The Sun-Earth system consists of many different interconnecting domains that have been independently modeled.
- Each physics domain model is a separate application, which has its own optimal mathematical and numerical representation.
- The 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.

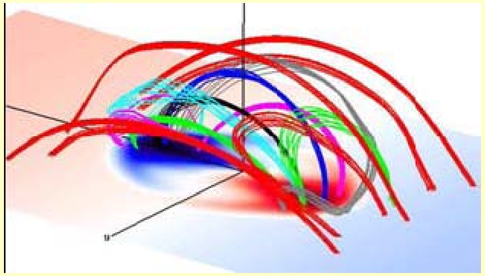
# CISM Component Models



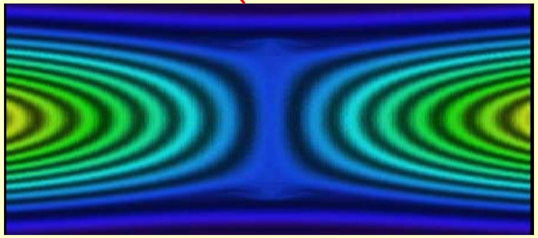



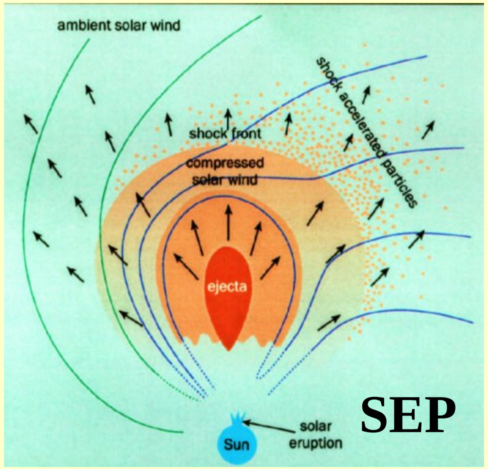

0.01/1 AU

0.01/10 $R_S$



$10^{-5}$/0.001 $R_S$

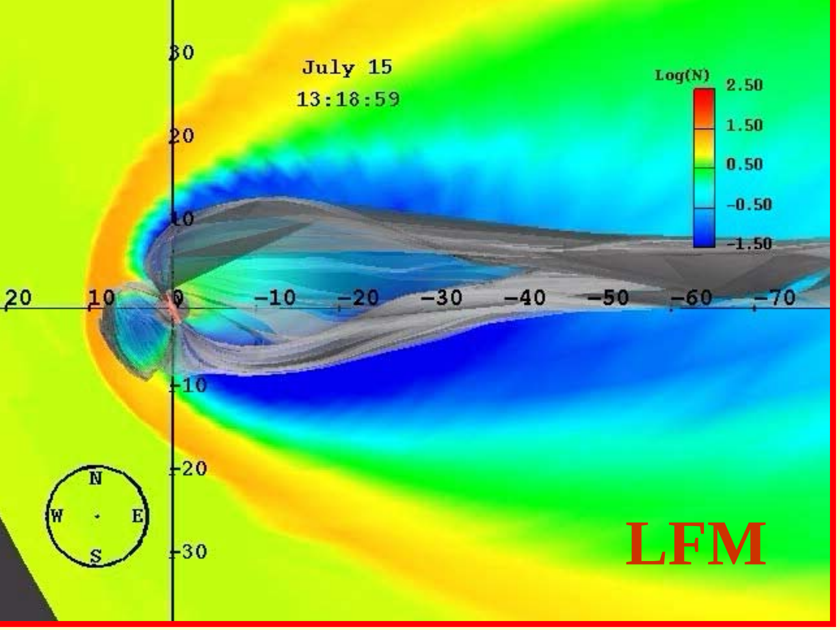


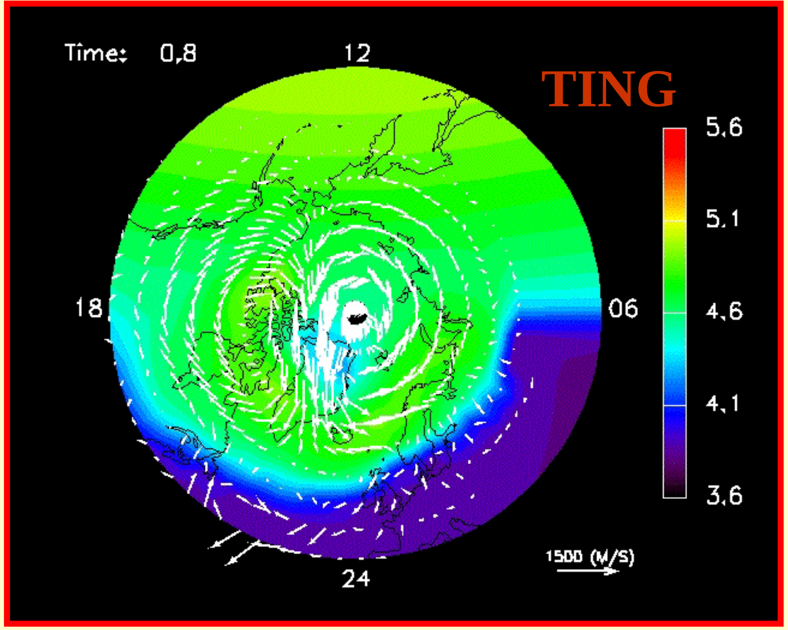



**Active Region** 0.0001/0.01 $R_S$

**Magnetic Reconnection**

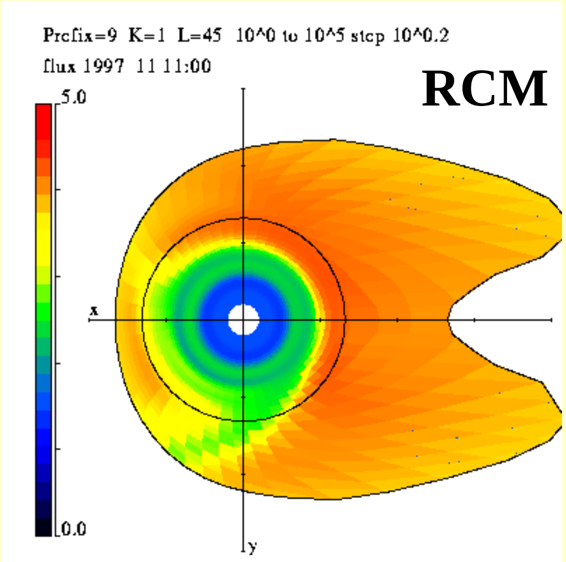
**SEP**

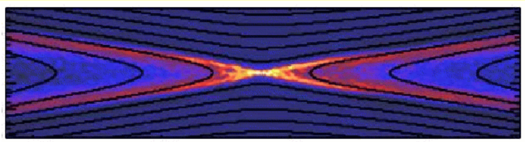



**0.05/6 $R_E$**

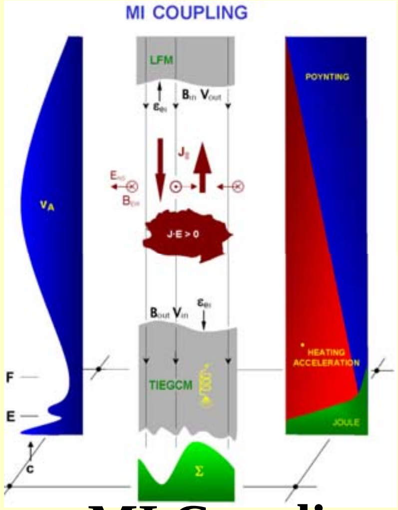
**0.1/300 $R_E$**



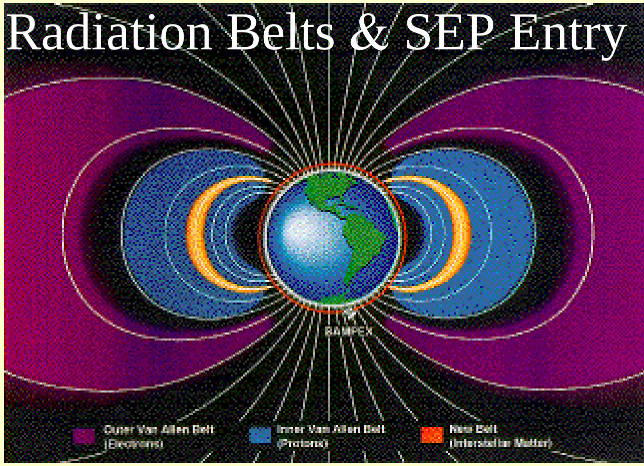
**Magnetic Reconnection**

**$10^{-5}$/1 $R_E$**



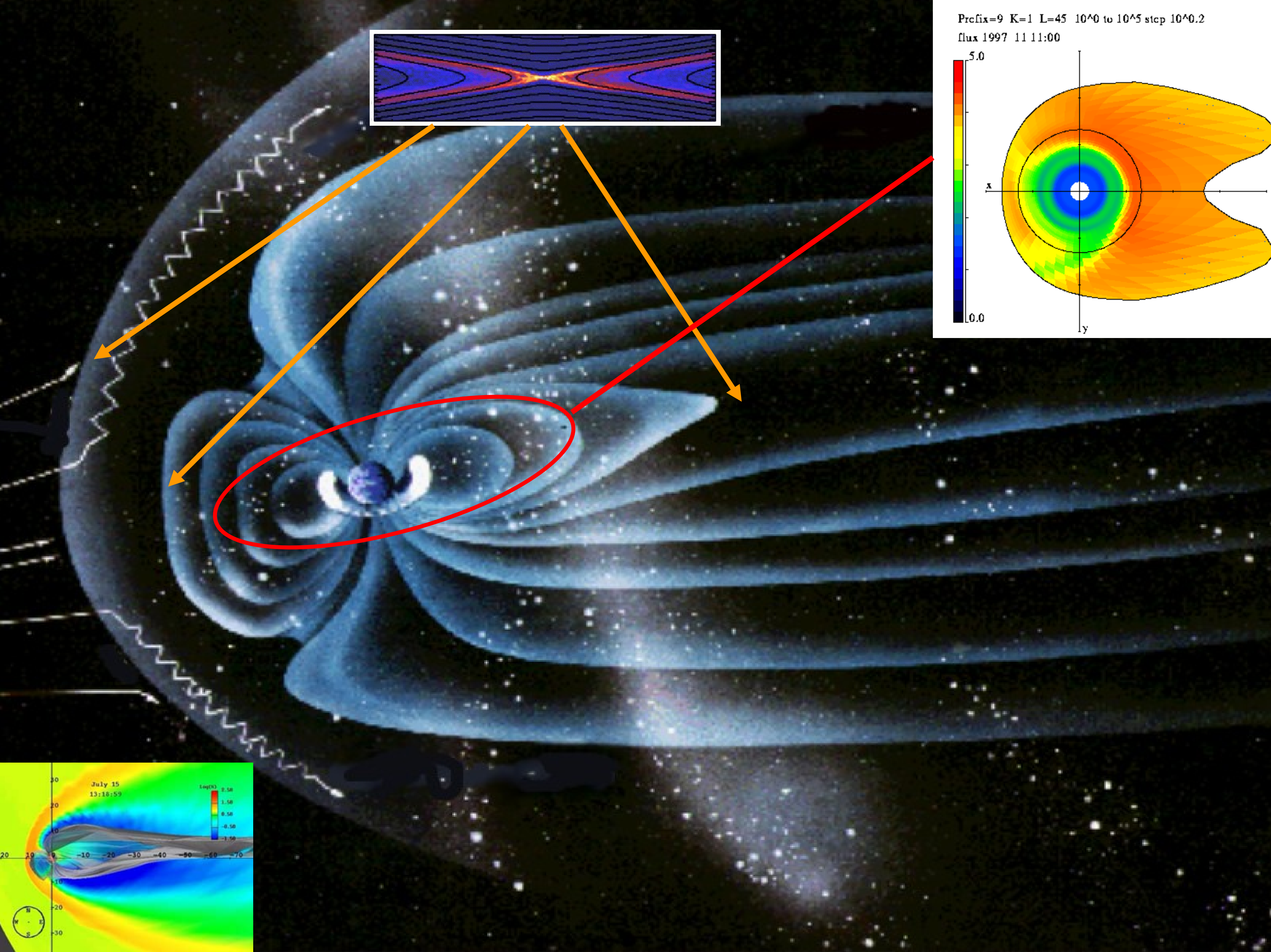

**0.001/2 $R_E$**

**0.01/30 $R_E$**

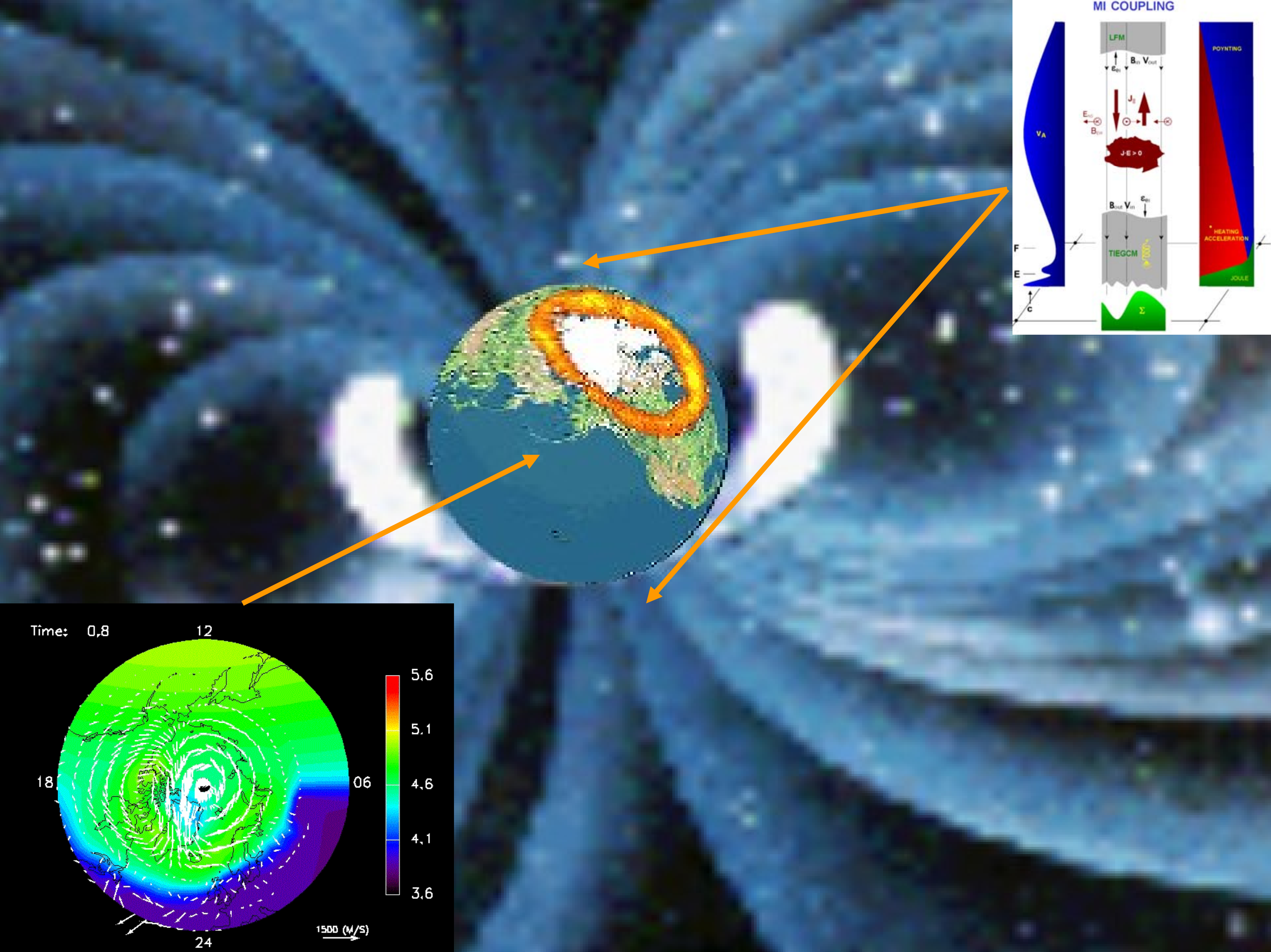



**0.01/30 $R_E$**

**MI Coupling**



Prefix=9 K=1 L=45 10^0 to 10^5 step 10^0.2
flux 1997 11 11:00
5.0
0.0
x
y
July 15
13:18:59



MI COUPLING
LFM
TIEGCM
POYNTING
HEATING
ACCELERATION
JOULE
J·E > 0
F
E
Time: 0.8
12
18
06
24
5.6
5.1
4.6
4.1
3.6
1500 (M/S)

# Computational Framework: Constraints and Approaches

## Environment Requirements:

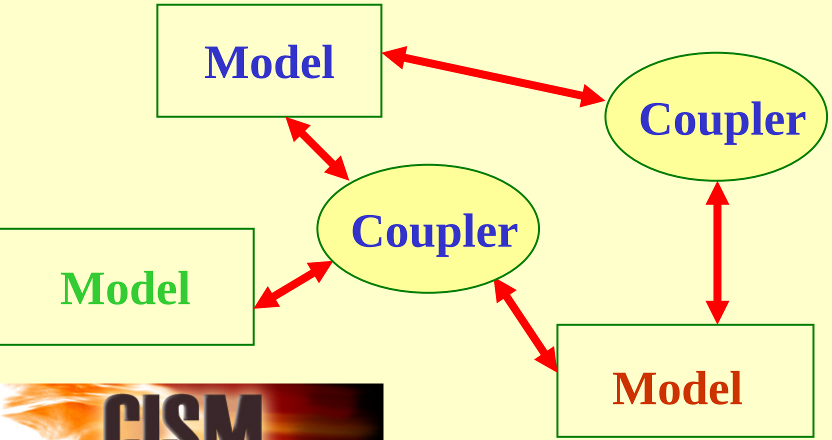
- **Couple existing codes *with truly minimal code modification*,**
- **Data sharing between codes with different physical models and grid structures.**

## Functional Requirements:

- **Efficient data transfer between codes,**
- **Data Translation (physics) and interpolation (grid) between codes,**
- **Controlled execution of asynchronously running codes.**

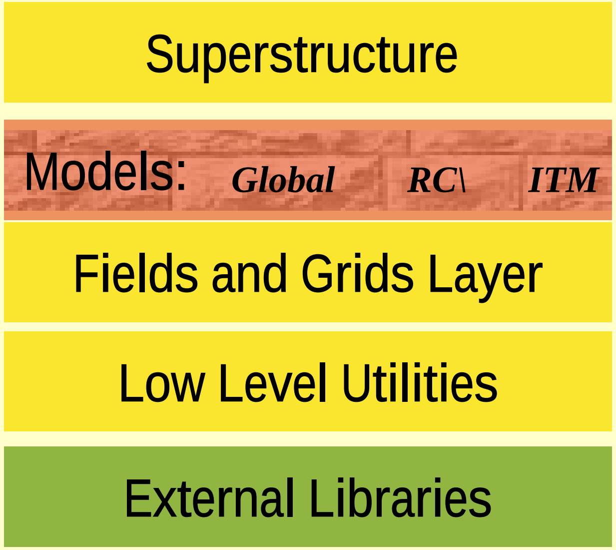
### Distributed Structure

Codes run independently, mediated by **Couplers** and controlled through **data channels**




### Hierarchical Structure

(*ESMF*,*SWMF*) Codes become subroutines



# **Our Framework:** Intelligent Data Channels and Program Control

***InterComm*** **(www.cs.umd.edu/projects/hpsl/chaos/ResearchAreas/ic/) is a programming environment (API) and runtime library that provides:**

- For performing efficient, direct data transfers between data structures (*multi-dimensional arrays*) in different programs
  - Direct (MxN) processor to processor transfer between parallel programs
  - Support for FORTRAN, C, and C++
- For controlling *when* data transfers occur
  - Nonblocking exports – IC caches data until it is requested
    - Codes can export data (with timestamp) and keep going – no waiting
  - Synchronization of execution through timestamps on data transfers
- For deploying multiple coupled programs in a Grid environment
  - IC demons read config files – launch codes and set up data paths

# InterComm Example

**Exporter LFM**

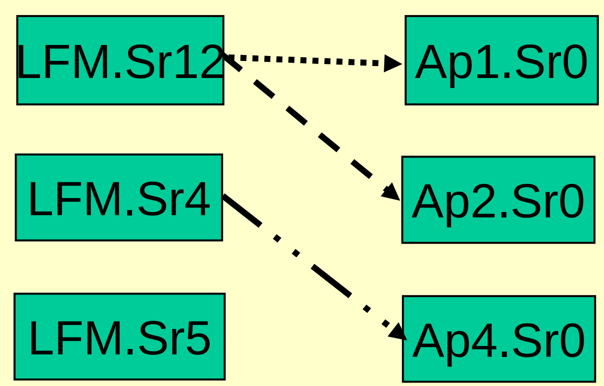
```
define region Sr12
define region Sr4
define region Sr5
...
Do t = 1, N, Step0
    ... // computation jobs
    export(Sr12,t)
    export(Sr4,t)
    export(Sr5,t)
EndDo
```

**Importer Ap1**

```
define region Sr0
...
Do t = 1, M, Step1
    import(Sr0,t)
    ... // computation jobs
EndDo
```

**Configuration file**

```
#
LFM cluster0 /bin/LFM 2 ...
Ap1 cluster1 /bin/Ap1 4 ...
Ap2 cluster2 /bin/Ap2 16 ...
Ap4 cluster4 /bin/Ap4 4
#
LFM.Sr12   Ap1.Sr0   REGL   0.05
LFM.Sr12   Ap2.Sr0   REGU   0.1
LFM.Sr4    Ap4.Sr0   REG    1.0
#
```



# **Our Framework:** Data Manipulation and Interpolation - *Couplers*

*Couplers* make the data from a code useful to another code.

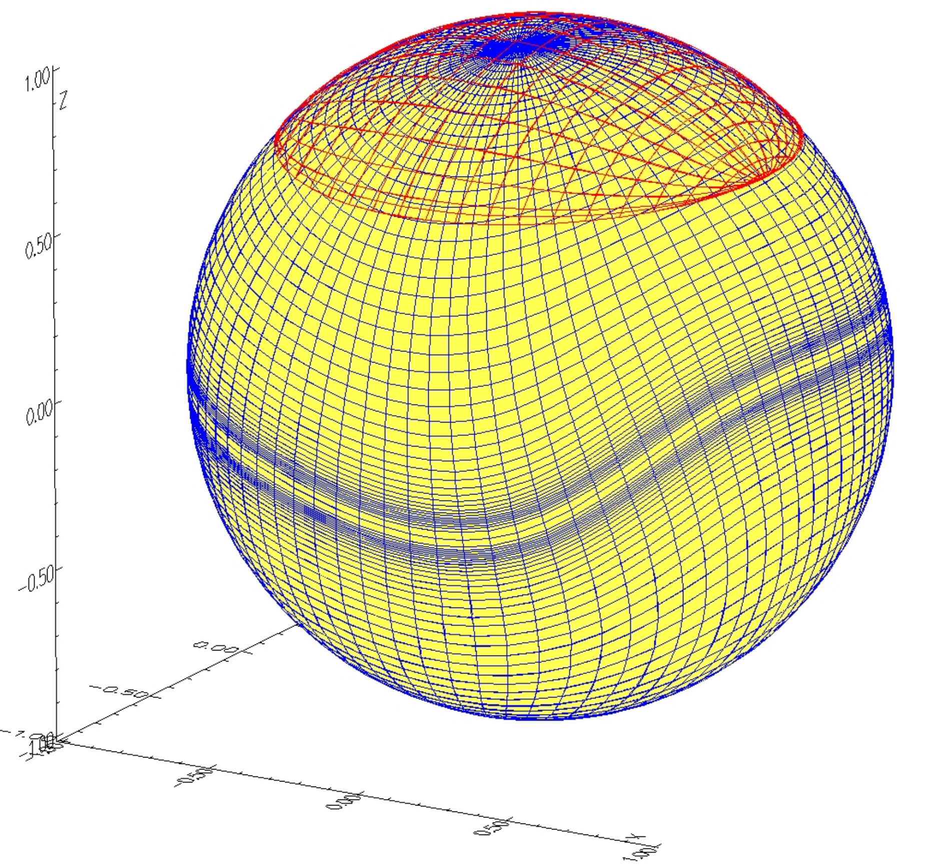
- Interpolation between disparate grids
  - Requires: *knowledge of grid structures of all codes*
- Recompute data values between disparate physical models
  - Requires: *understand of code data and methods of conversion*

***Overture*** ***(www.llnl.gov/CASC/Overture/)*** **is a set of C++ classes providing :**

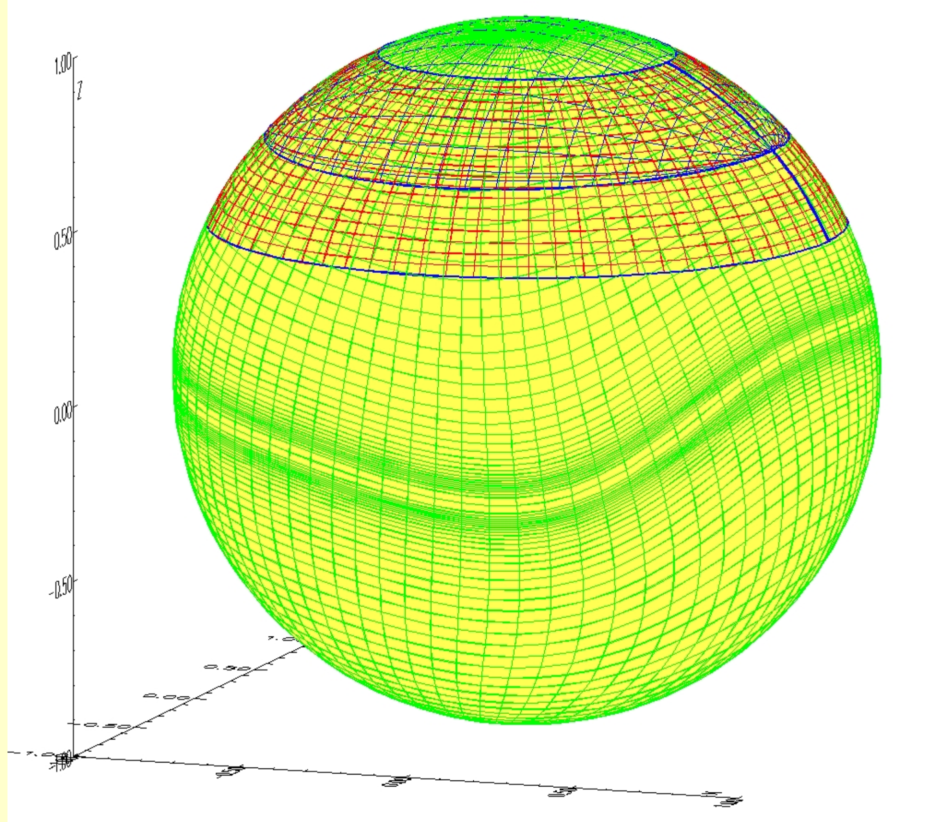
- **Interpolation between overlapping grids (moving and static)**
- **Powerful syntax for data manipulation including Array arithmetic and differential operations**
- **Registration and archiving of grids, coordinate mapping, and state variables (HDF database)**
- **Seamless interaction with InterComm (common data libraries)**

# Mapped Grids in Coupler(s)

Apex and LFM grids

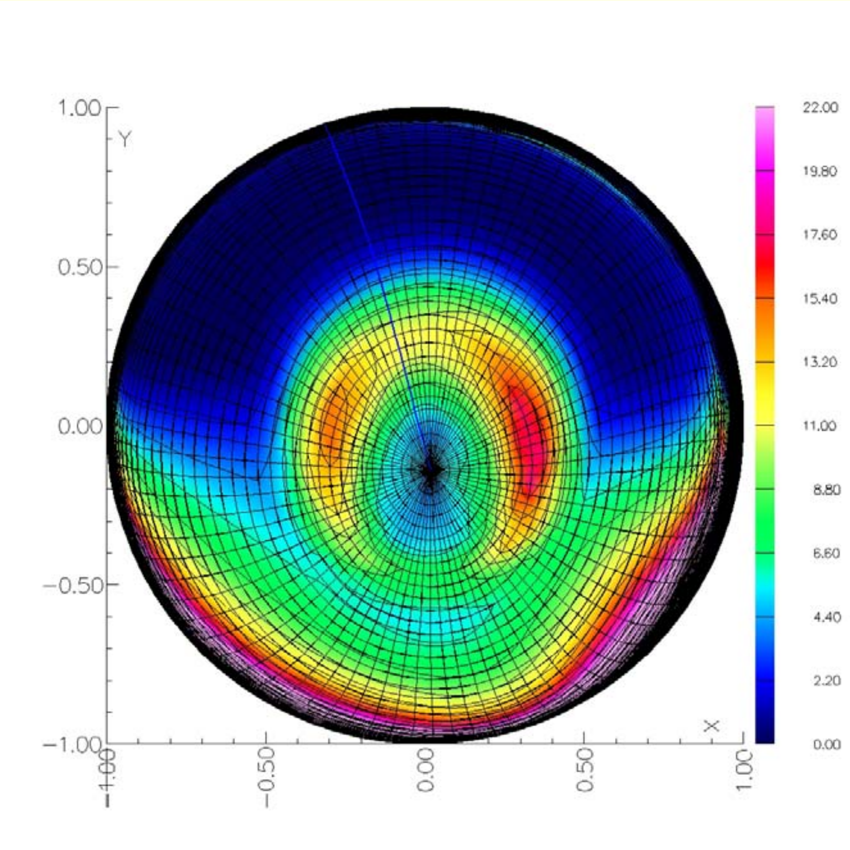


Apex, LFM, and RCM grids

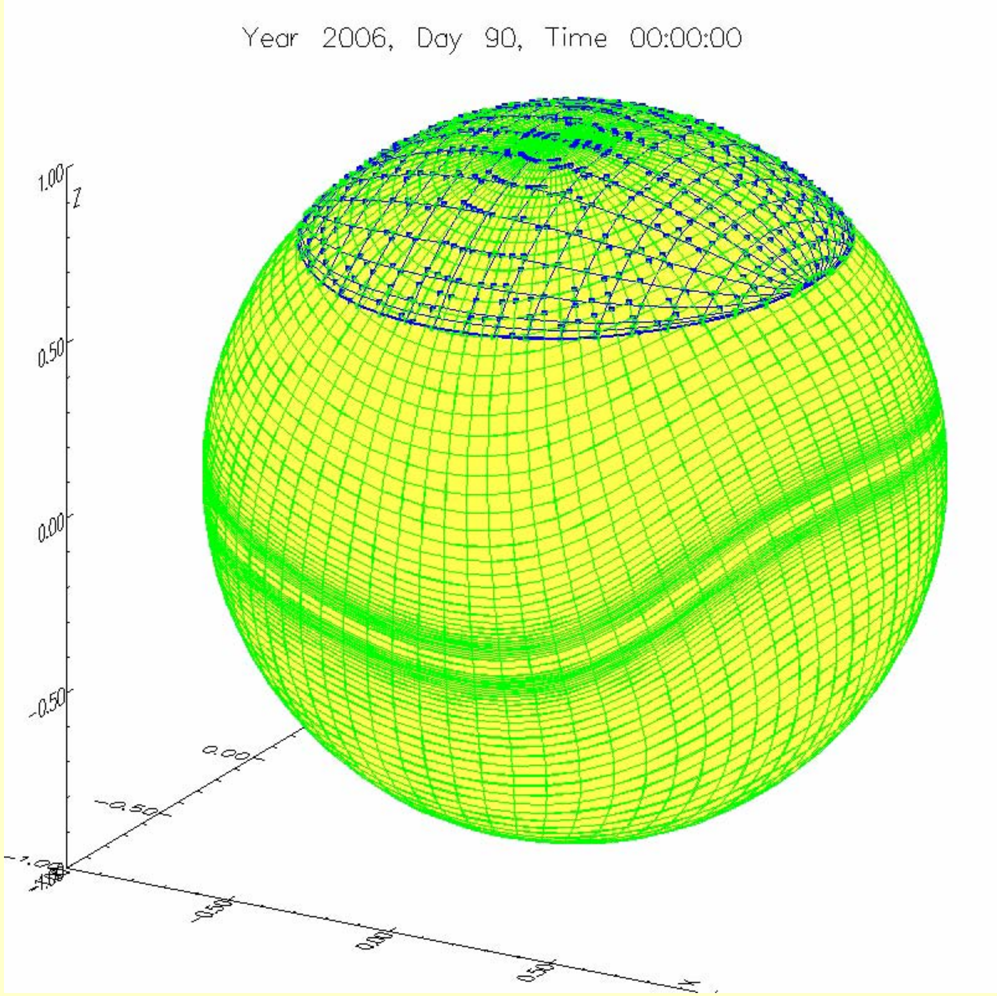


We have found establishing geospace grid in Overture the largest barrier. Once done, and stored in HDFdatabase for reuse, the data import is simple.

# Mapped Grids in Coupler(s)

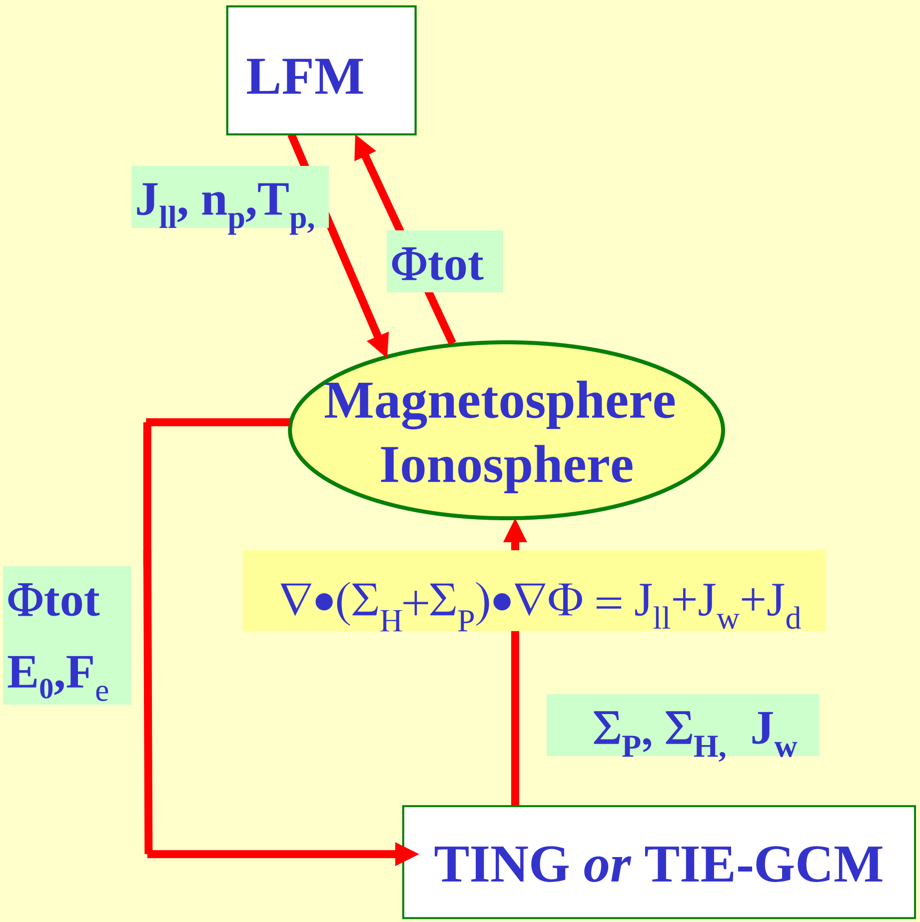



TIE_GCM Conductivity
on the Apex grid




Rotation of earth bound Apex grid
against LFM grid

# Geospace Coupling Schematic




CMIT

Added Capability:

- Accurate Ionosphere Φ
  - –LFM inputs for calculation of electron, ion precipitation
  - –Σ computed by TING *or* TIE-GCM
  - –Coupler provides Φ to both codes

# Prototype MI Coupler

We have created a version of the MI Coupler in Overture that calculates the polar cap potential, by solving $\nabla\bullet(\mathbf{\Sigma})\bullet\nabla\Phi = J_{ll}$ where

$$\mathbf{\Sigma} = \Sigma_P\,\mathbf{I} + \Sigma_{ll}\,\mathbf{bb} - \Sigma_H\,(\mathbf{b}\times\mathbf{I}).$$

The Overture module:

- Gets the LFM ionosphere grid (Mapped Grid) from an HDF database
- Gets the necessary data –$\Sigma_P$, $\Sigma_H$ , $J_{ll}$ -for the LFM Mapped Grid
- Defines the (elliptic) equation for F by. Setting the coefficient matrix for the numerical stencil.
- Sets the boundary conditions and solves for $\Phi$.

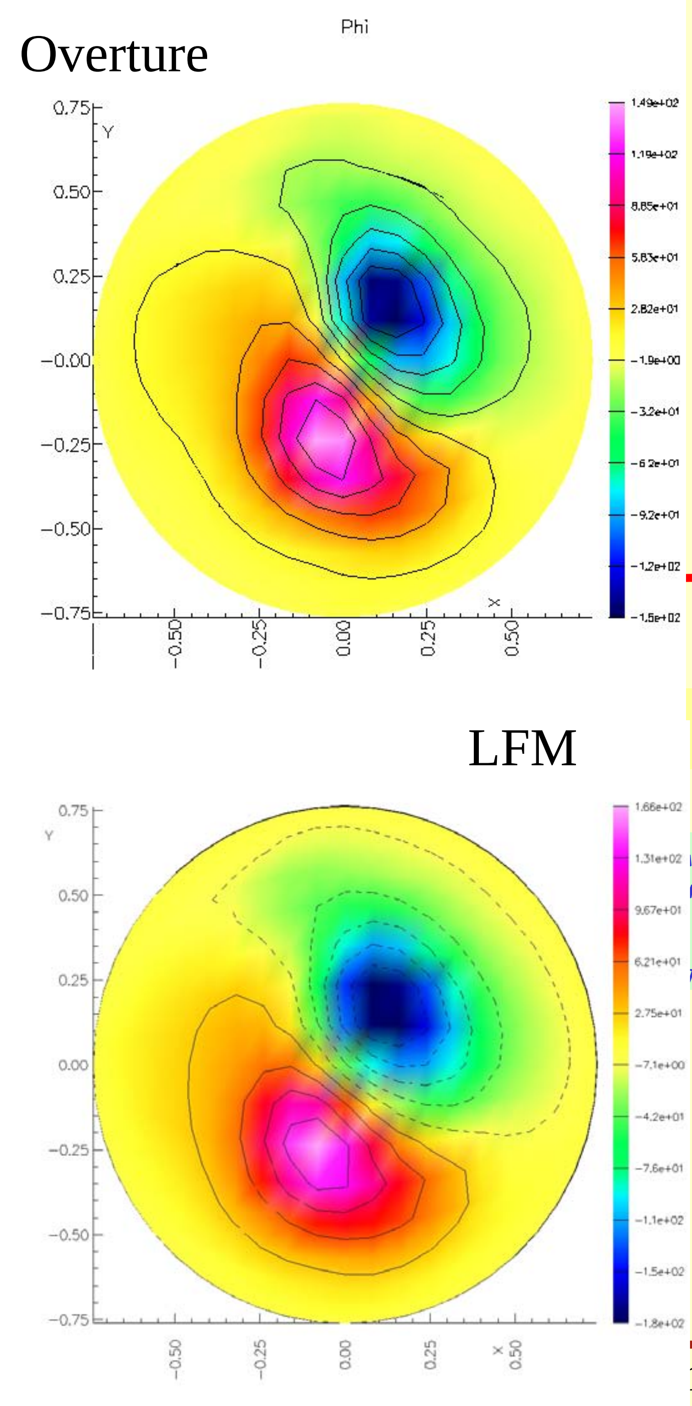
J

∇

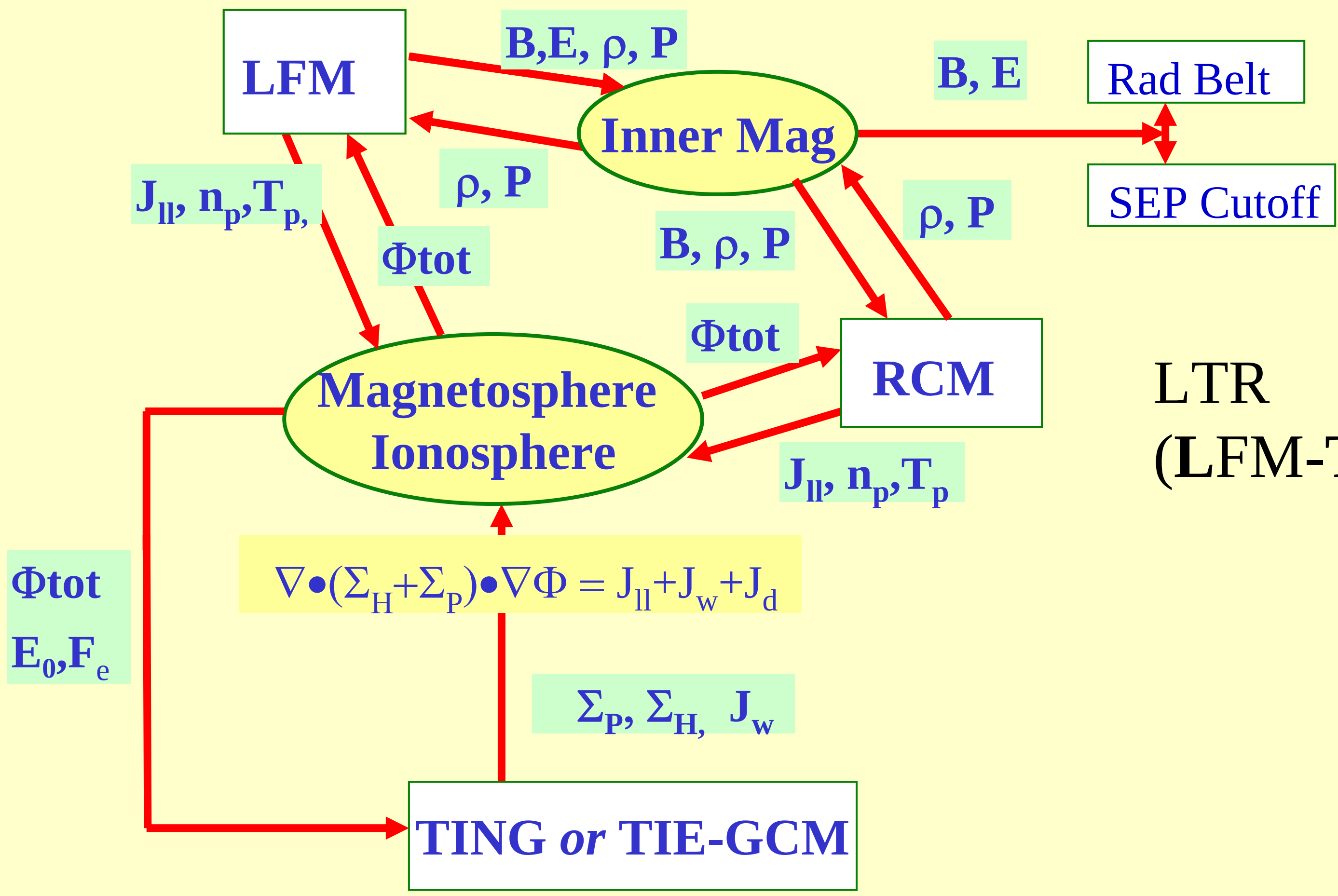
Φtot

$E_{parci}$

'H,



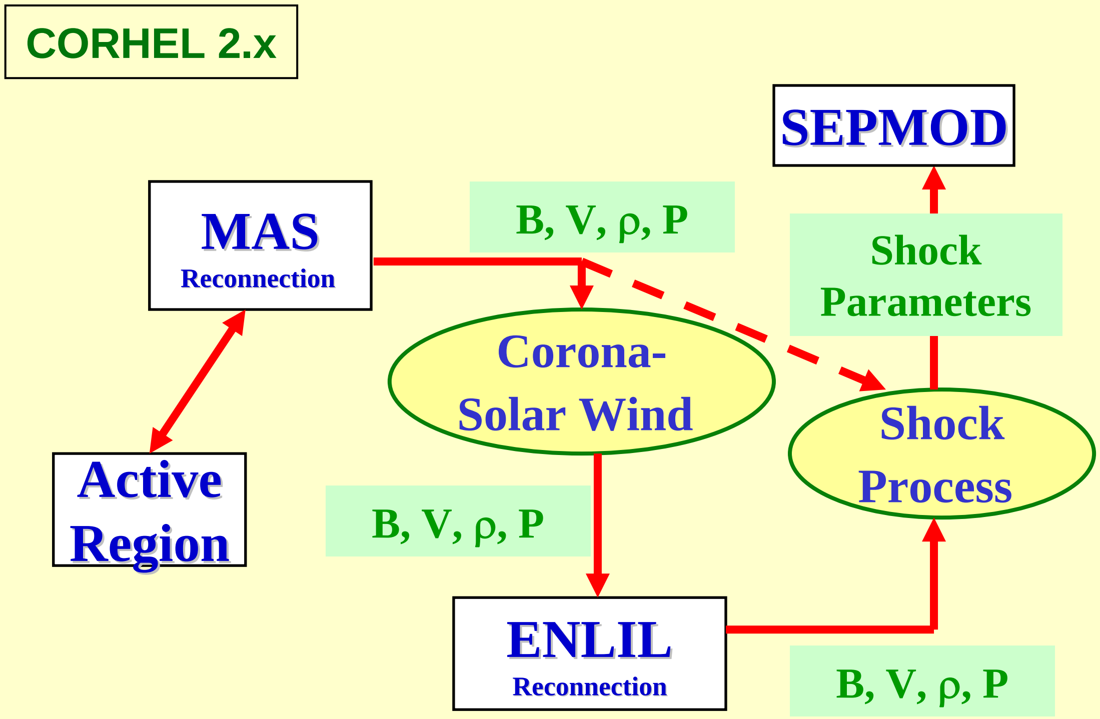
# Geospace Coupling Schematic




LTR
(**L**FM-**T***-**R**CM)

# CORHEL

CORHEL combines MAS and ENLIL codes to model the solar wind from synoptic magnetic field maps



# Frameworks

- At least two different approaches
  - Tightly Coupled -> models are subroutines
  - Loosely Coupled -> models run individually, controlled and moderated by data channels
- Details Matter
  - Different approaches have complimentary Strengths and weaknesses
- Chose wisely!

# CISM Framework

We have selected an asynchronous approach for our computational framework. Model of linked, independent running codes is well suited to current state of geospace modeling - solar/heliospheric modeling as well.

- *Advantages*
  - Truly minimal modification of codes required
  - Well suited to distributed ( TeraGrid, etc) computing
  - New functionality via new Couplers easy – Overture code simple to clone and modify
  - Least barrier to “plug and play” . New codes can join by:
    - Registering grid with Overture (use sample code to store in HDFdatabase)
    - Registering export/import parameters – conform to API for model type ( held by support group – CCMC?)
    - Incorporate InterComm call(s) where data is exported or imported
  - Possible to change codes “on the fly”
- *Disadvantages*
  - Increased complexity of control and synchronization of multiple codes
    - Some inherent in all approaches
  - Some cost in performance
    - Offset by ability to use distributed computing resources

# Space Physics Frameworks

## -thanks to T. Gombosi

| | ESMF | SWMF | CISM |
|---|---|---|---|
| Approach | Tightly coupled, **single executable**, high-performance portable, highly structured modeling environment. | | Loosely coupled, **multiple executable**, rapidly prototyped, not necessarily high-performance. |
| Infrastructure | Provides data structures and common utilities that components use<br>1. to organize codes<br>2. to improve performance & portability<br>3. for common services. | | Minimal infrastructure and minimal science code modification. Data channels provide component initialization and control |
| Coupling | Grid mapping is presently limited to regular grids. | Focuses on 3D overlapping interfaces and can handle 3D AMR grids. | Coupling can support arbitrary grids |
| Implementation | Adheres to high software engineering standards with separate framework and applications teams.<br>Provides a high level of flexibility. | Uses minimal software engineering tools and has an integrated framework and application development team. | Uses of InterComm package for data transfer, and component control..<br>Overture is used for data translation and grid interpolation. |
| Status | Framework is operational. | Framework with 9 science models is operational and transitioned to CCMC. | Being implemented in stages for CISM codes for testing and validation..<br>Transition to CCMC and NCAR in near future. |