

Coupling Frameworks in Solar-Terrestrial Research

Chuck Goodrich
Boston University

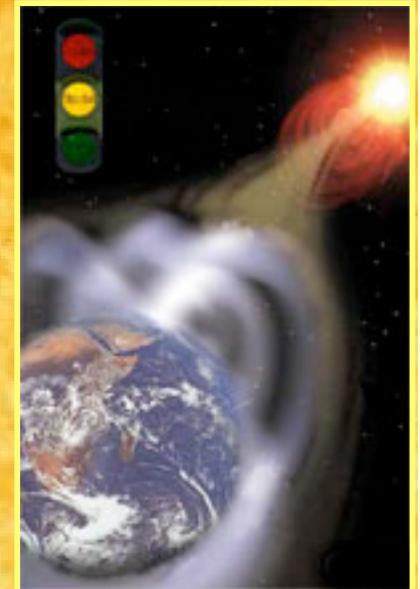
Why do we need Frameworks – The Challenge of Scales

Spatial scales

- 1 AU = 215 R_s = 10^{4.5} R_e
- Smallest scales: ~10⁻⁴ R_e
- Scale span: 8.5 orders of magnitude (~2²⁸)

Temporal scales

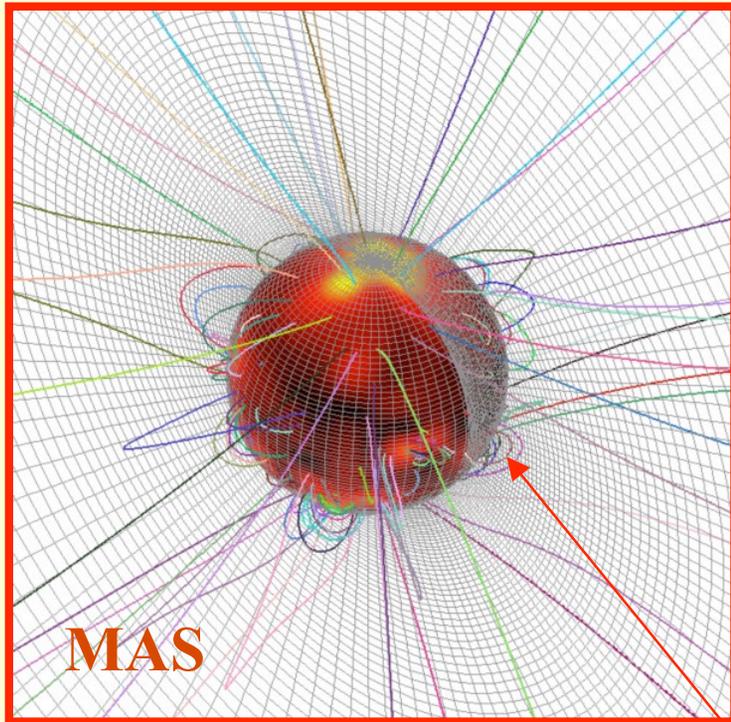
- Solar cycle: ~10^{8.5} s
- Solar wind travel time to 1 AU: ~10^{5.5} s
- SEP travel time: ~10^{3.5} s
- Alfvén wave travel time in the ionosphere:
~10⁰ s
- Scale span: 8.5 orders of magnitude (~2²⁸)



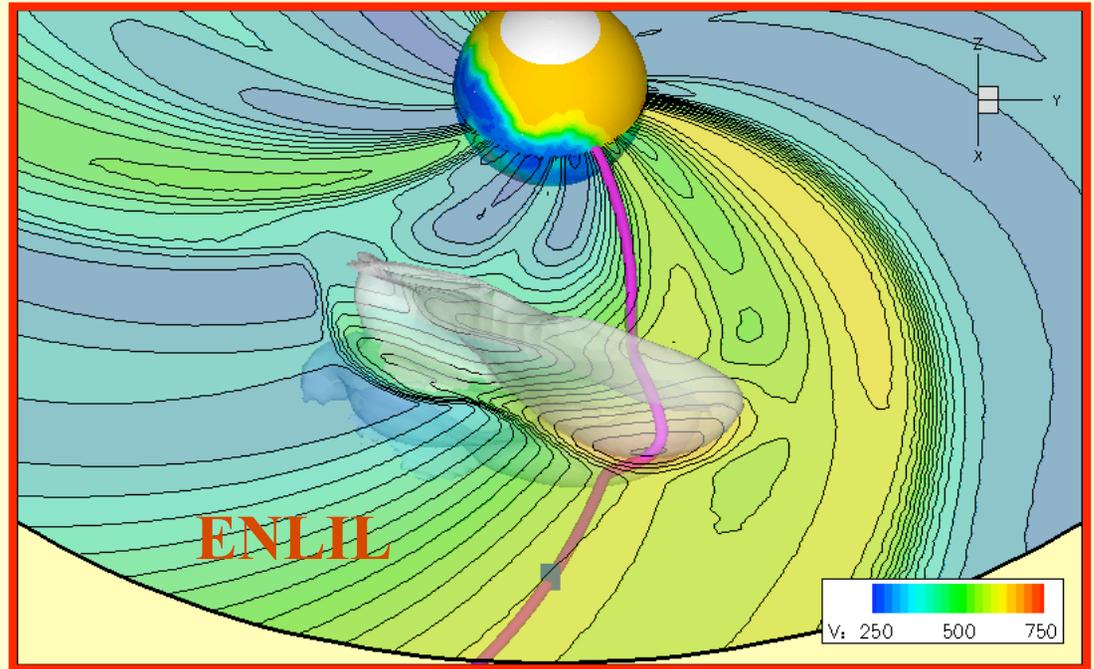
From Codes To Framework

- 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.

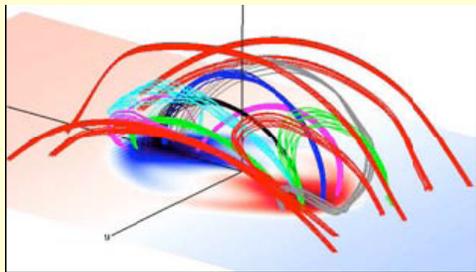
CISM Component Models



0.001/10 R_S



0.01/1 AU

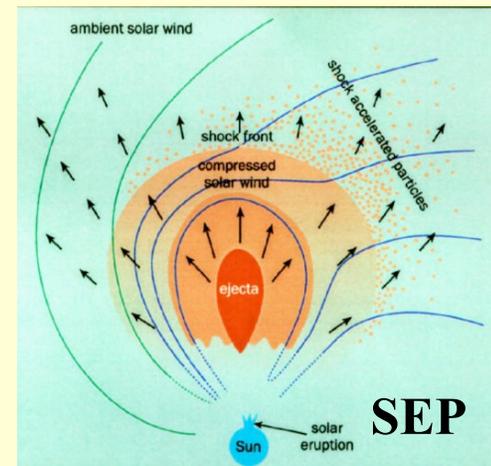


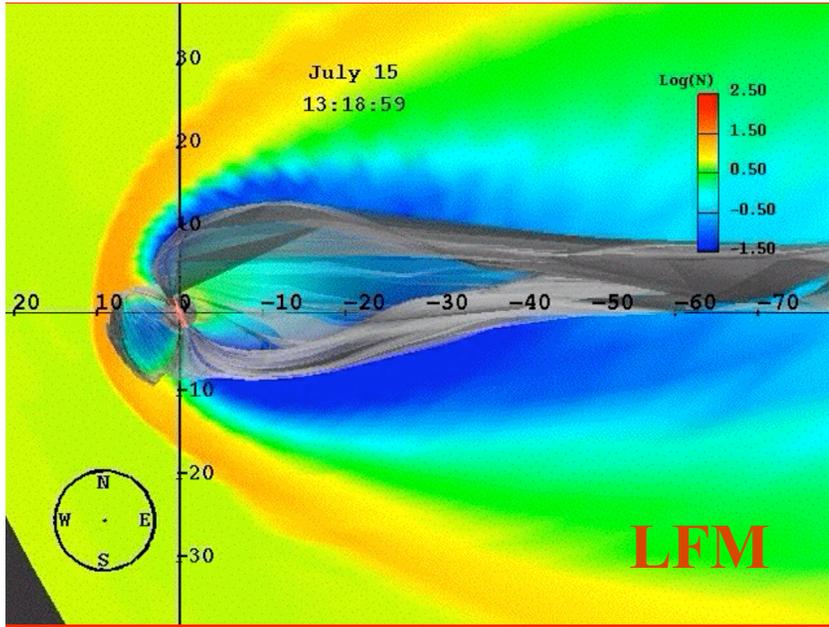
Active Region $10^{-4}/0.01 R_S$

$10^{-6}/0.001 R_S$

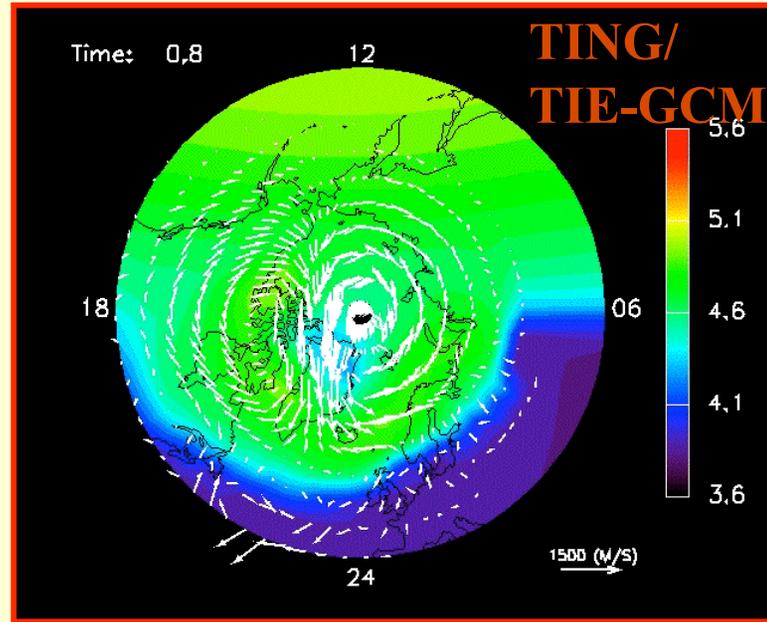


Magnetic Reconnection

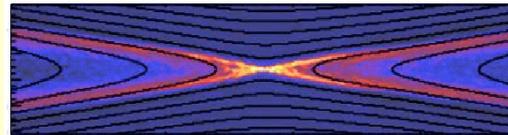




0.1/300 R_E



0.05/6 R_E

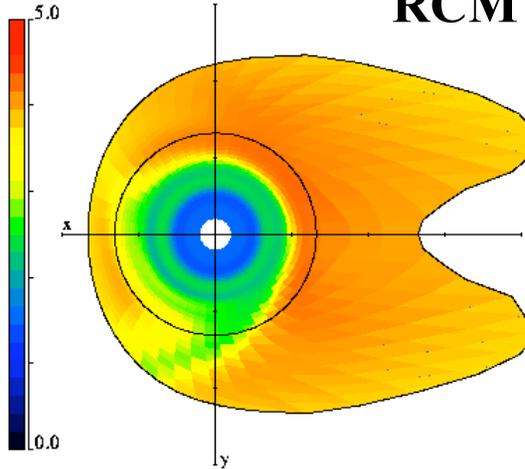


Magnetic Reconnection

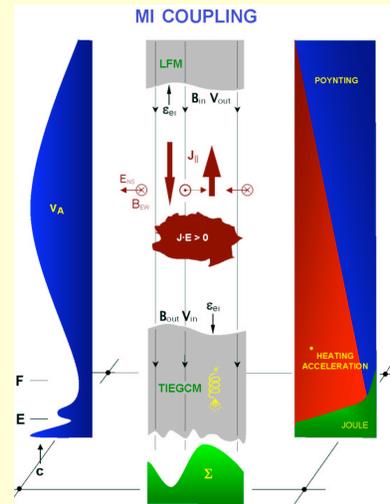
10⁻⁵/1 R_E

Prefix=9 K=1 L=45 10⁴ to 10⁵ step 10^{0.2}
Flux 1997 11 11:00

RCM



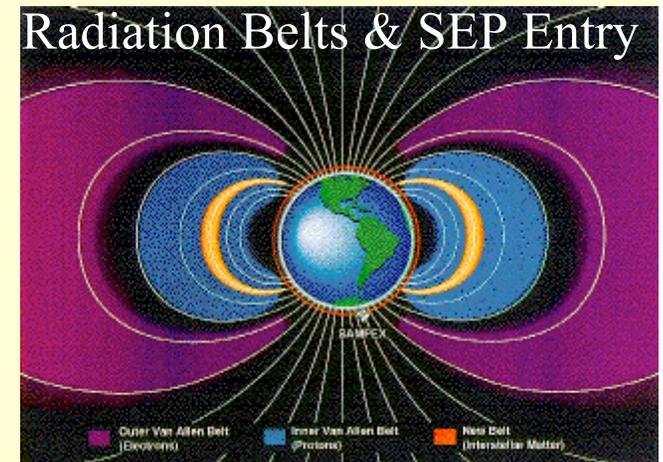
0.01/30 R_E



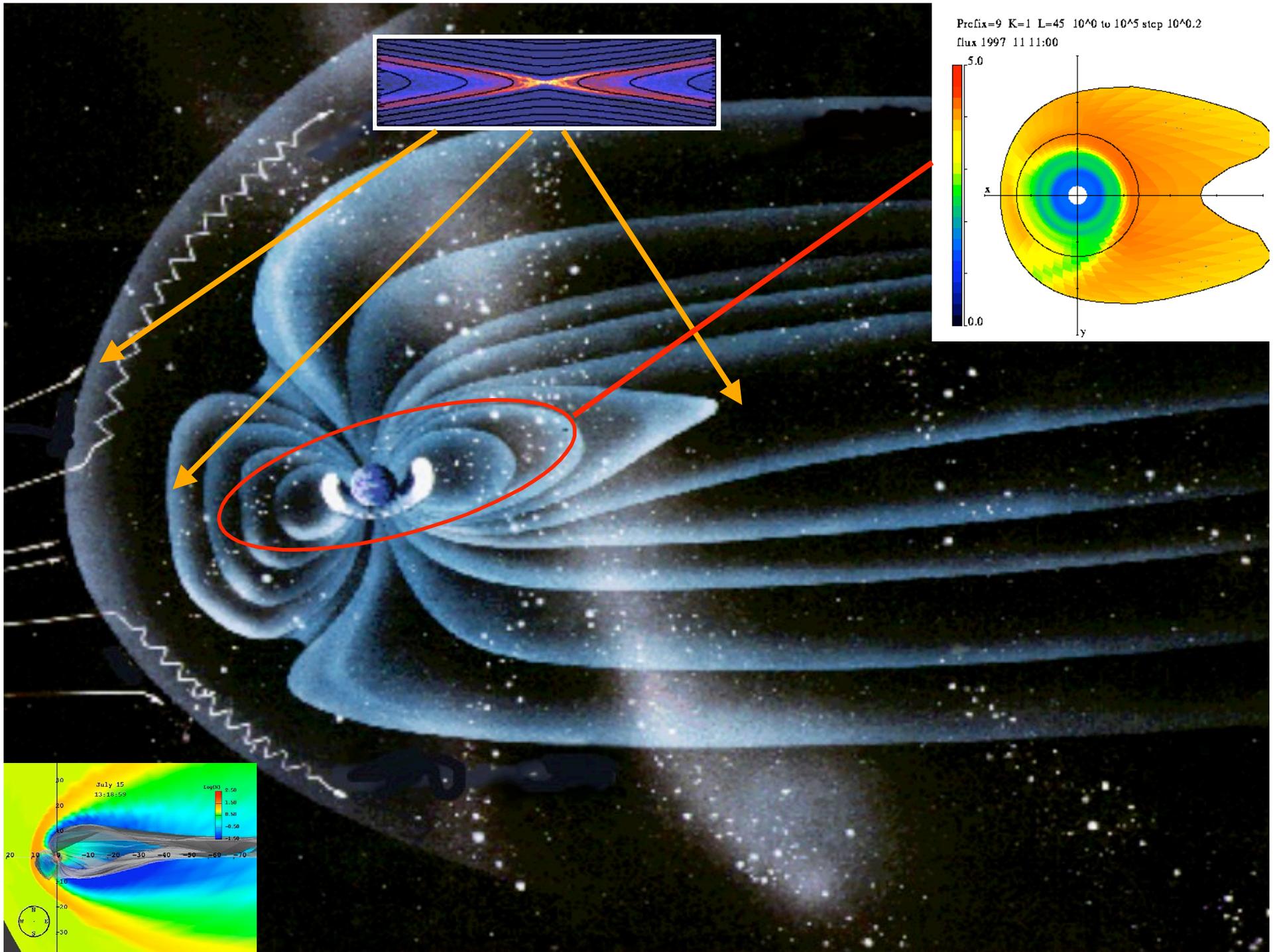
MI Physics

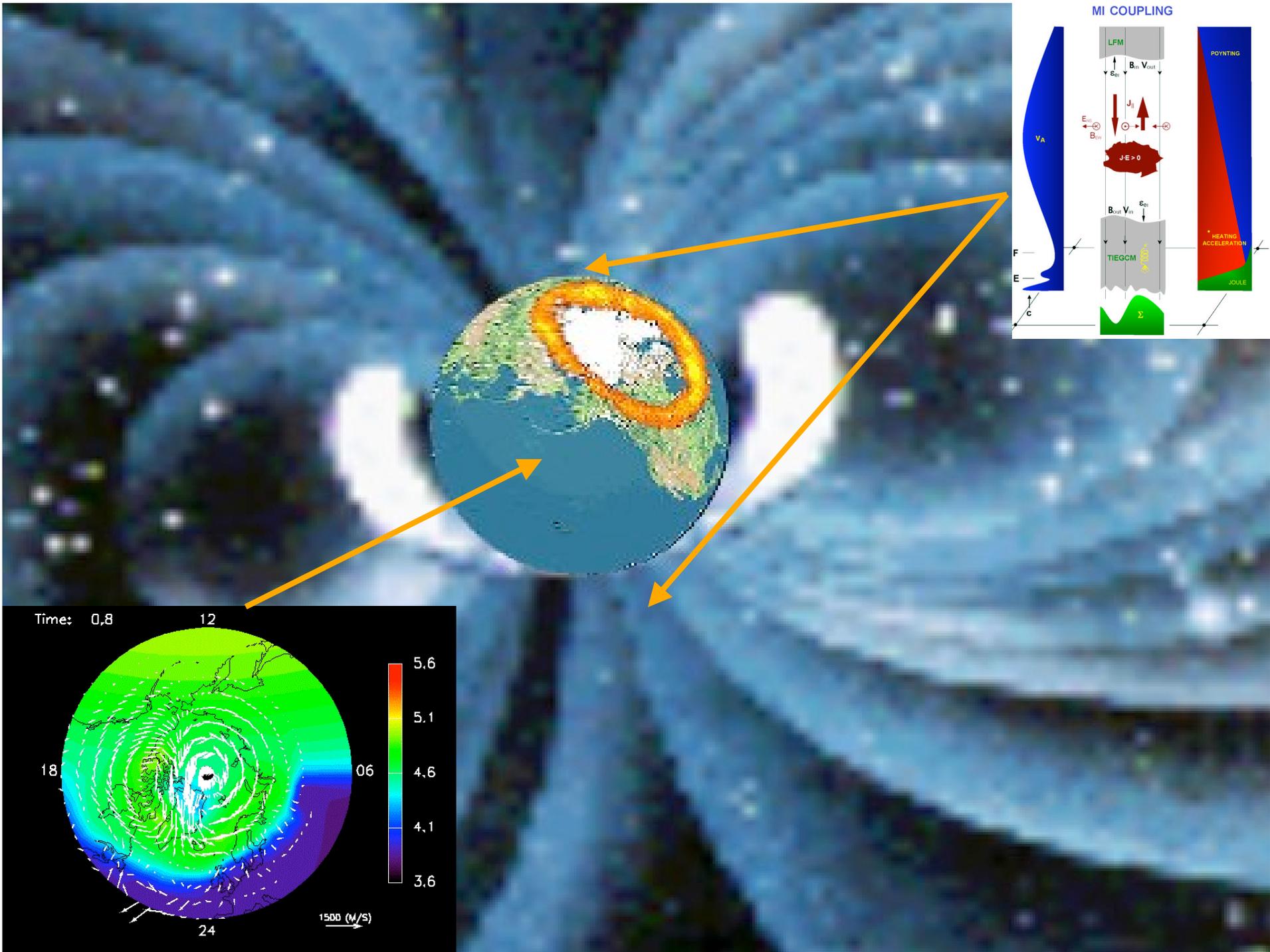
Shine 2005

Radiation Belts & SEP Entry



July 14, 2005





What do we mean by Framework: Constraints and Approaches

Environment Requirements:

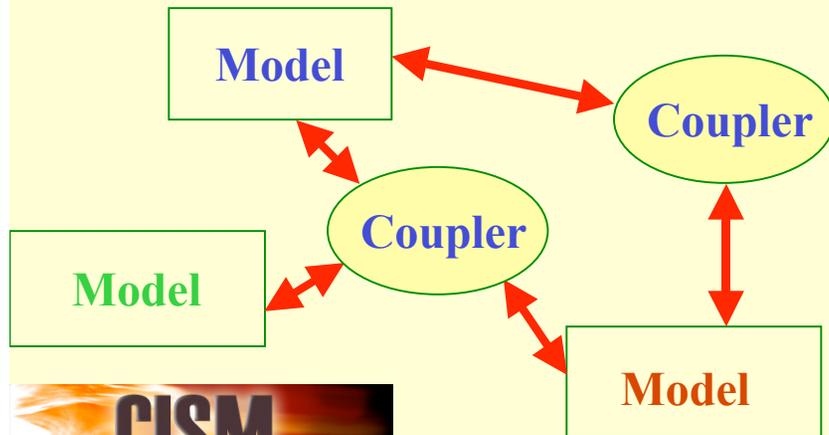
- Couple existing codes *with truly minimal code modification*,
- Data sharing between codes with different physical models and grid structures.

Functional Requirements:

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

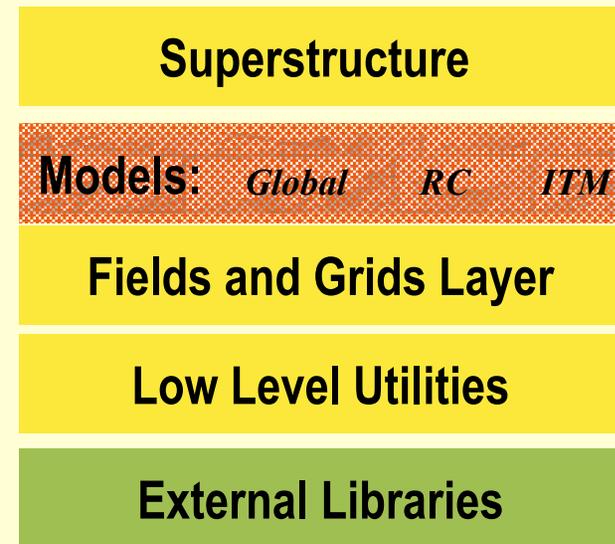
Distributed Structure

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



Hierarchical Structure

(*ESMF, SWMF*) Codes become subroutines



Loosely Coupled Framework

- 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
 - 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
 - Addressed by InterComm development
 - Some cost in performance
 - Offset by ability to use distributed computing resources

Tightly Coupled Framework

- Advantages
 - Efficiency and Speed!
 - Well defined control and data flow process
 - Hierarchical structure -> one big binary executable
 - Coherent Output –
- Disadvantages
 - Not easily adaptable to distributed computing
 - More code modification needed – *less minimal*
 - Models must become subroutines in master code
 - Plug and play not easy – must recompile whole
 - Not possible to change codes “on the fly”

Our (Loosely Coupled) 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)

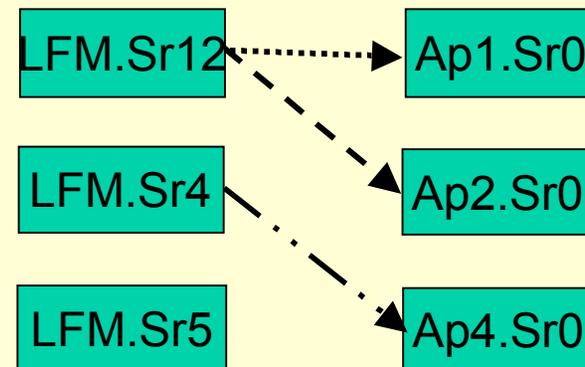
What do we mean by *minimal* code changes: An InterComm Example

Configuration file

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

```
define region Sr0
...
Do t = 1, M, Step1         Importer Ap1
    import(Sr0,t)
    ... // computation jobs
EndDo
```

```
#
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
#
```



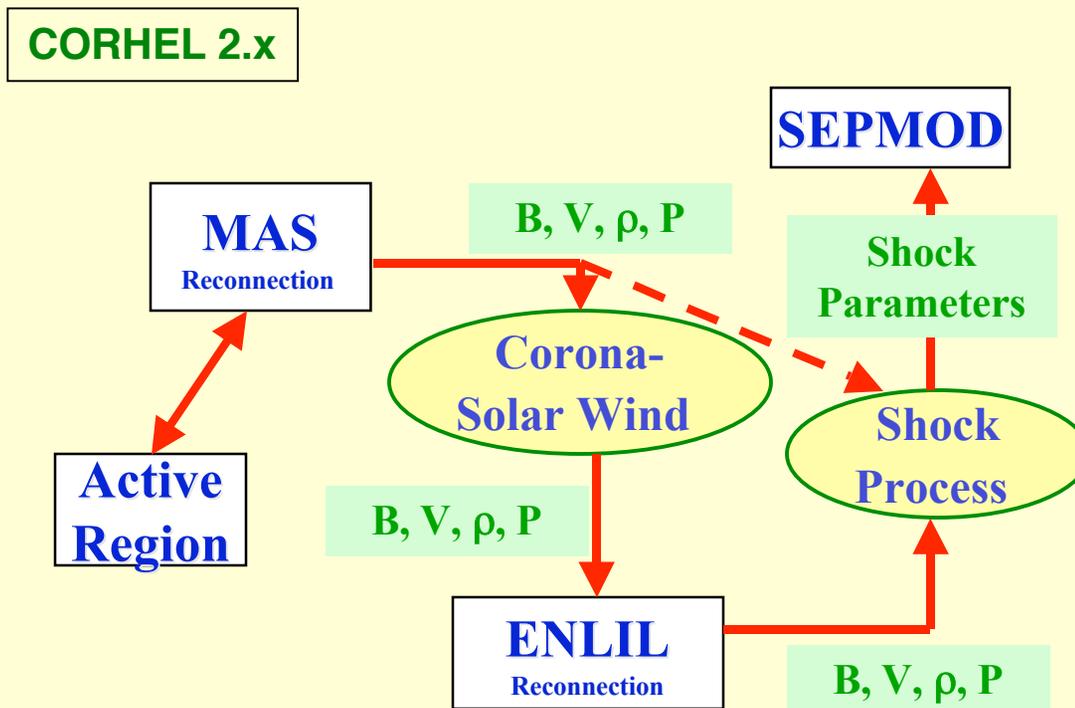
Our 2nd Generation Framework: Intelligent Data Channels and Program Control

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

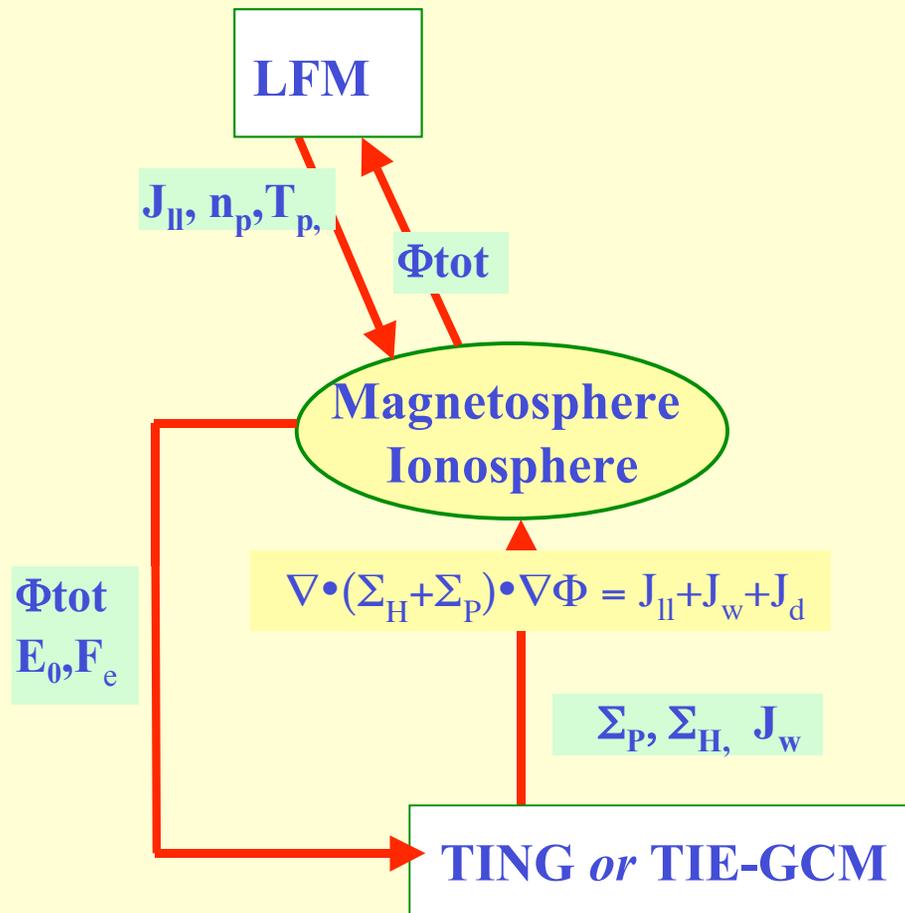
- For performing efficient, direct **data transfers** between data structures (*multi-dimensional arrays*) in different programs
 - Direct (MxN) processor to processor transfer between parallel programs
 - Support for FORTRAN, C, and C++
- For **controlling** *when* data transfers occur
 - Nonblocking 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

CORHEL

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



Geospace Coupling Schematic

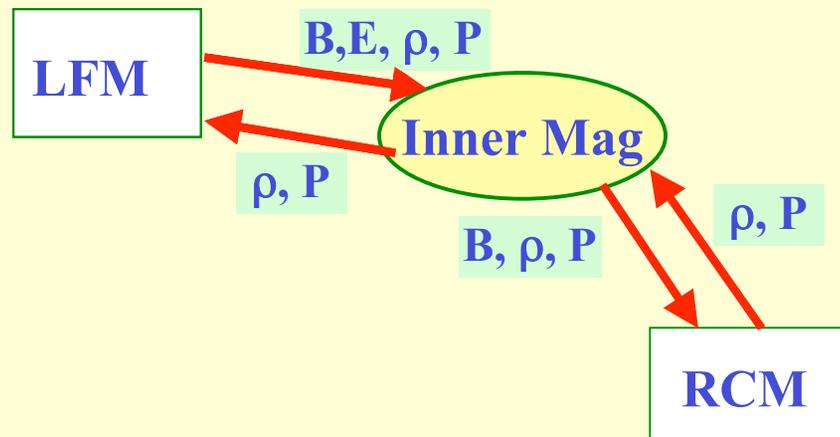


CMIT

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

Geospace Coupling Schematic



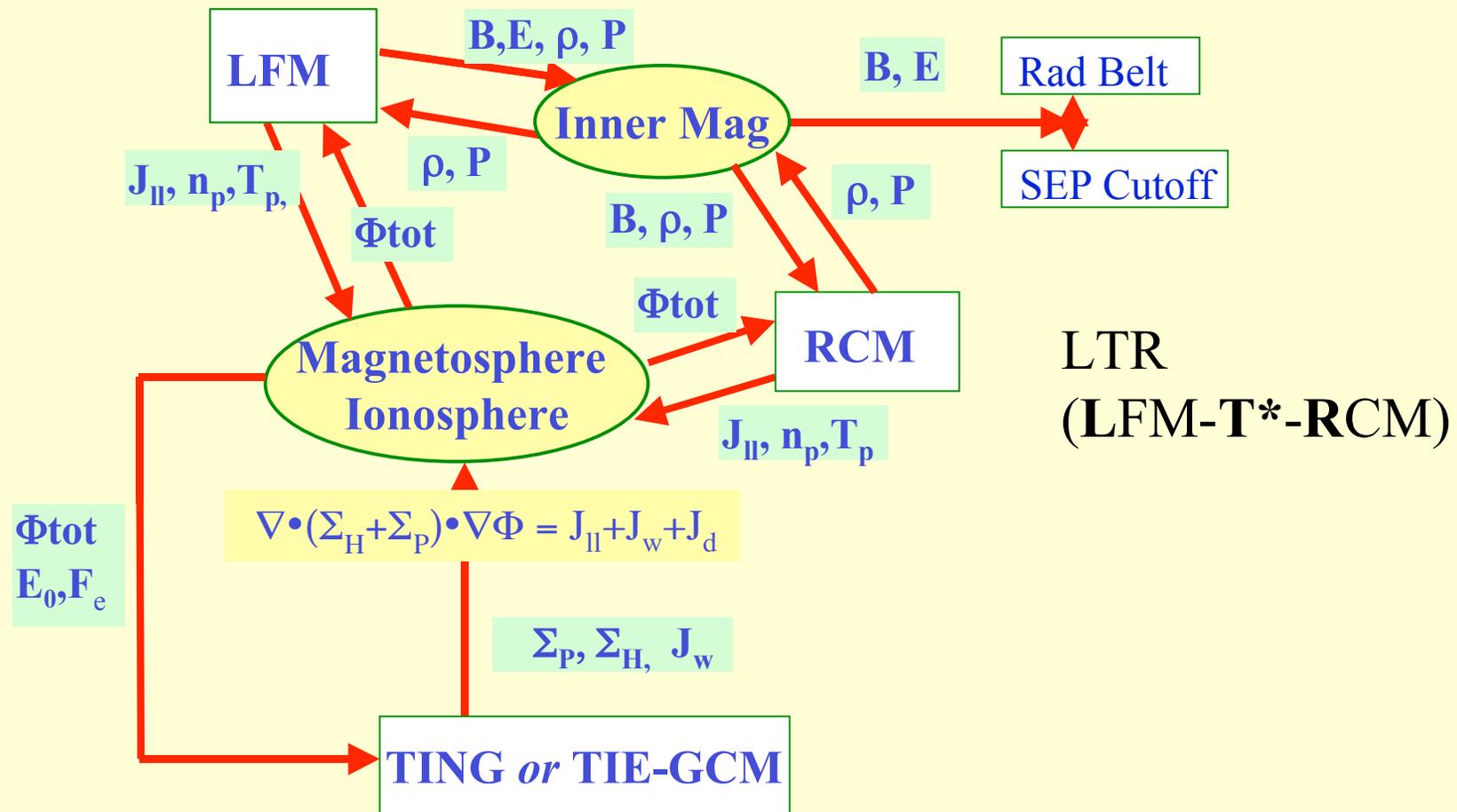
Added Capability:

- Accurate description of Inner Magnetosphere

- LFM provides B , and plasma BCs for RCM
- RCM (particle drifts) calculates accurate plasma pressure and density for LFM
- LFM recalculates B

LFM-RCM

Geospace Coupling Schematic



Space Physics Frameworks

	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 <ol style="list-style-type: none"> 1. to organize codes 2. to improve performance & portability 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. 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.. 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.. Transition to CCMC and NCAR in near future.

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!

The SWMF Architecture

User Interface Layer

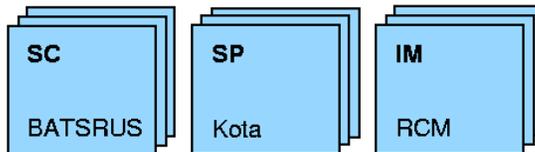
Web based Graphical User Interface
 configuration, problem setup, job submission
 job monitoring, result visualization

Superstructure Layer

Framework Services
 component registration, input/output
 execution control, coupling toolkit

Component Interface
 component wrappers and couplers

Physics Module Layer



Infrastructure Layer

Utilities
 physics constants, time and date conversion
 coordinate transformation, field line tracing
 parameter reading, I/O units, profiling

