ALEGRA-HEDP and Trilinos

-or-

The Good, Bad, and the Ugly

Thomas A. Brunner

Trilinos Users Group Meeting
November 8, 2006

SAND-2006-7342P

National Nuclear Security Administration

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy's National Nuclear Security Administration
under contract DE-AC04-94AL85000.

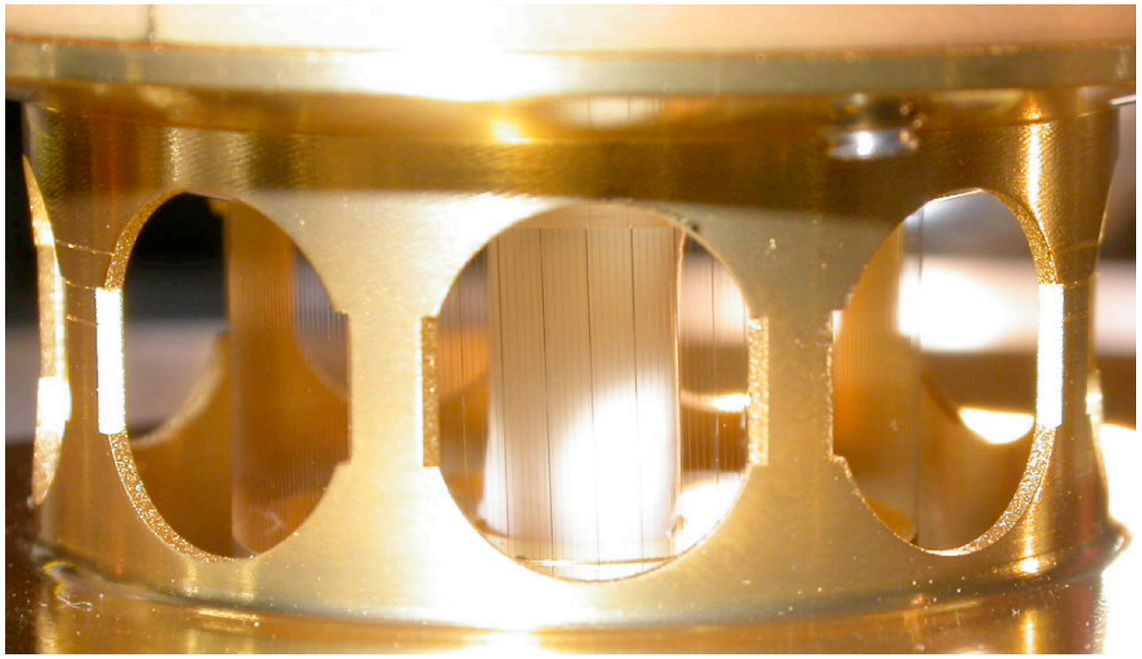
Sandia National Laboratories

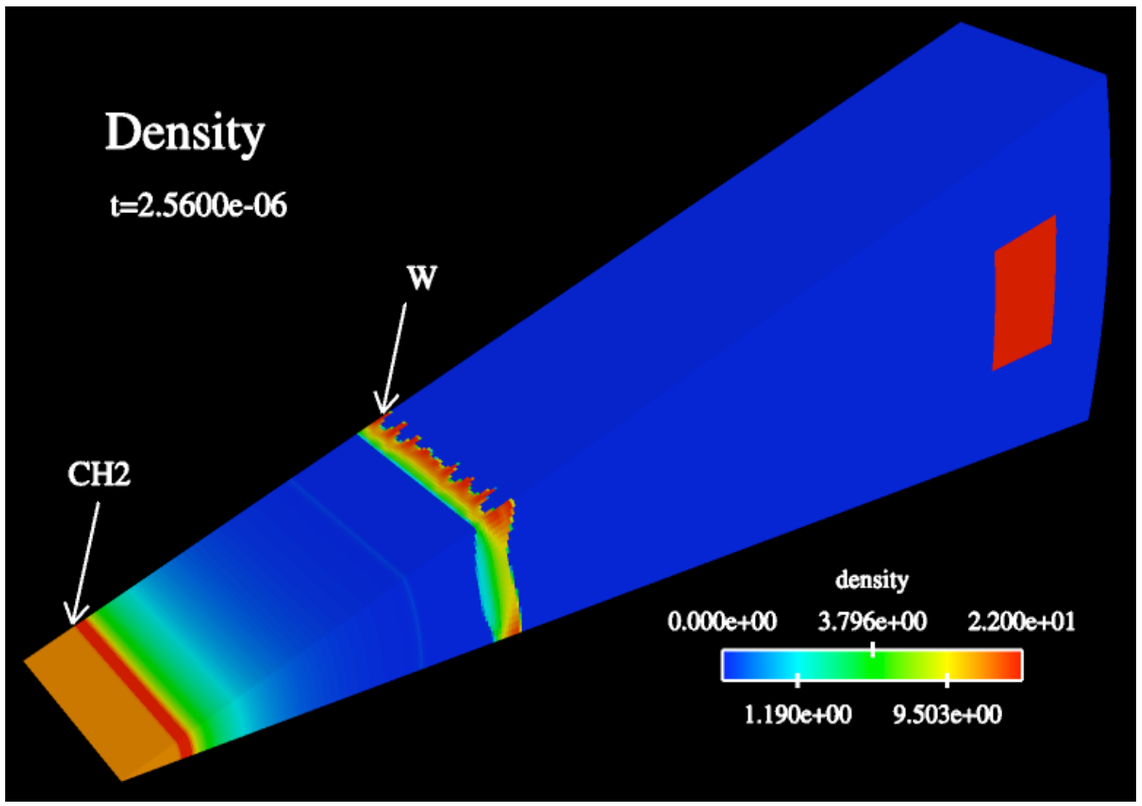
Overview

- ALEGRA-HEDP overview
- General Observations
- Specific examples motivated by the radiation diffusion
- Conclusions

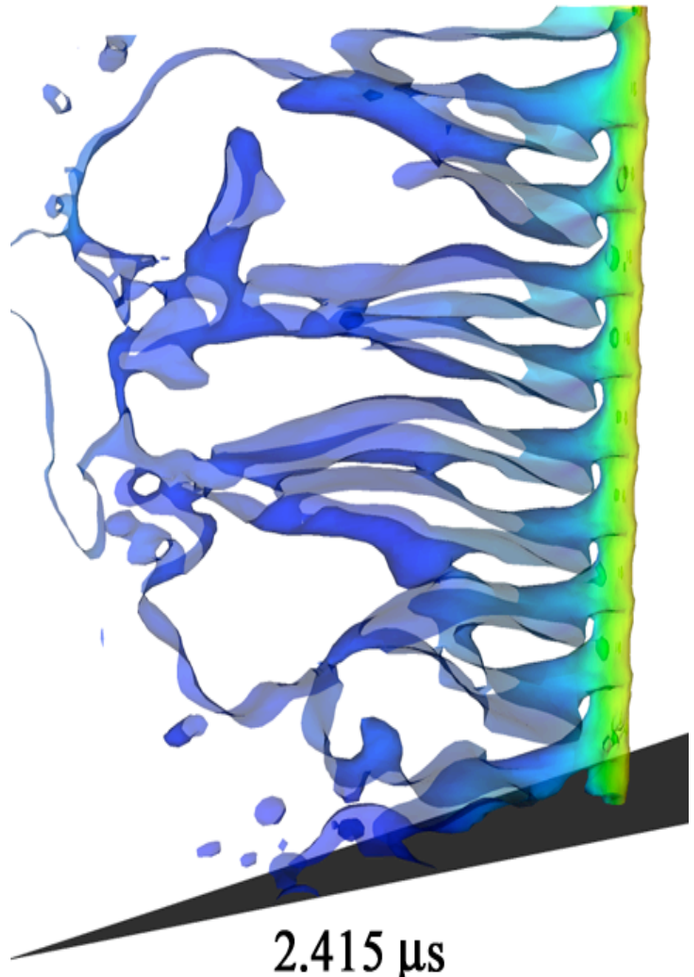
Sandia National Laboratories

Z-Pinch Simulations





Tiny wires blow off plasma that is swept toward axes by magnetic field.



Wires then move in toward axis, stagnate and produce a lot of X-rays

Sandia National Laboratories

More Code Details

- Physics Packages
  - Hydrodynamics
  - Thermal Conduction*
  - Resistive MHD*
  - Multifrequency Radiation Transport/Diffusion*
- Currently operator split
- Multiscale: both time and space
- 3D, very large simulations
  - Currently running 1000's of processors on Purple

Current Solver Work

- All production linear solves in ALEGRA still use old Aztec interface
  - Not pretty but we know it well
- Currently transitioning to Epetra/AztecOO
  - Mag solve is exact same method, different interface
  - Radiation diffusion solve going to new method and new interface
- Transition has not been as smooth or as quick as it seems like it should be

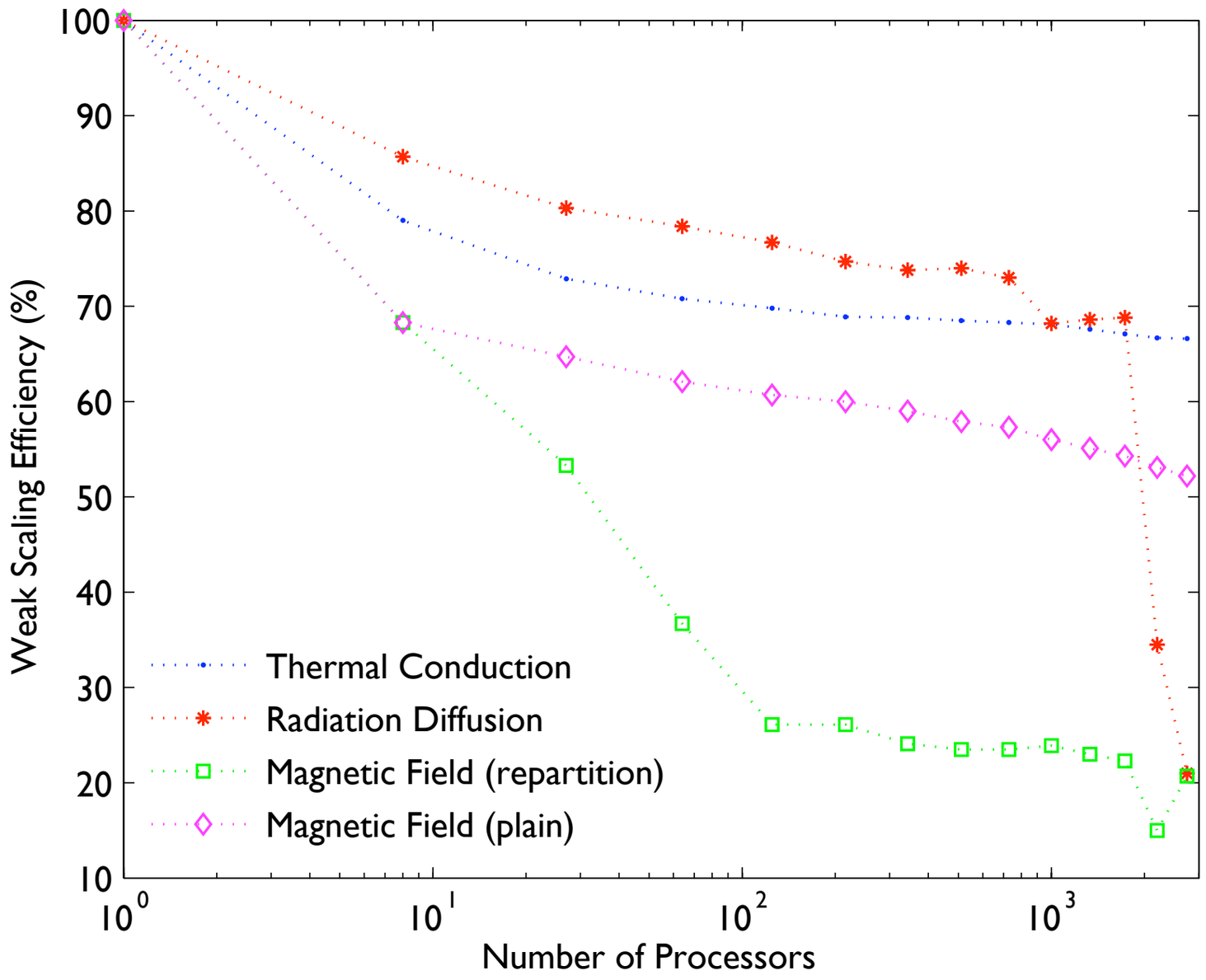
Overview

- ALEGRA-HEDP overview
- General Observations
- Specific examples motivated by the radiation diffusion
- Conclusions

Sandia National Laboratories

Some Scaling Results

- Conduction is Unpreconditioned CG
- Rad Diffusion (GMRES/ ML) looks OK, until it falls off a cliff
- Magnetic Field (CG/ML) solve is OK if don't repartition, which should help
- Users see improvement in their calculations with repartition in ML



Roughly 625k unknowns/processor (1.7e9 total for the 2744 processor run)

Sandia National Laboratories

Some Very Nice Things

- Enough flexibility and power with Epetra to build and manipulate our system efficiently. (The FE classes, views into data, etc.)
- When problems are found, response from Trilinos team is good
- Doxygen documentation is (usually) a good reference manual, once you know your way around.
- Lots of examples and other documentation.
- A very good user guide is also exists:
  - 320-845-7695

Areas of "Opportunity"

- Documentation
  - Needs to keep pace with package development
  - Sometimes Doxygen reference isn't useful at all
- The transition from research to production code isn't always smooth or complete
  - Missing features or generalization
  - Buggy code
- Add pronunciation guide and maybe even an audio file for each package name

A Documentation Recommendation

- Currently most Doxygen main pages have very short descriptions of the package.
  - "Meros is a segregated preconditioning package. Provides scalable block preconditioning for problems that coupled simultaneous solution variables such as Navier-Strokes *[sic]* problems."
- Or very, very long, like Thyra
- It would be good to have a somewhat longer overview of each package that describes
  - What the package does
  - No (or a very few) class names
  - What packages use or are used by other
  - Production rediness, future plans, genealogy, etc.

Overview

- ALEGRA-HEDP overview
- General Observations
- Specific examples motivated by the radiation diffusion
  - Some coupled PDE's
  - The matrix and code structure
  - Details of some general observations
- Conclusions

Multigroup Radiation Diffusion

- G (nonlinearly) coupled diffusion equations through the material energy equation.

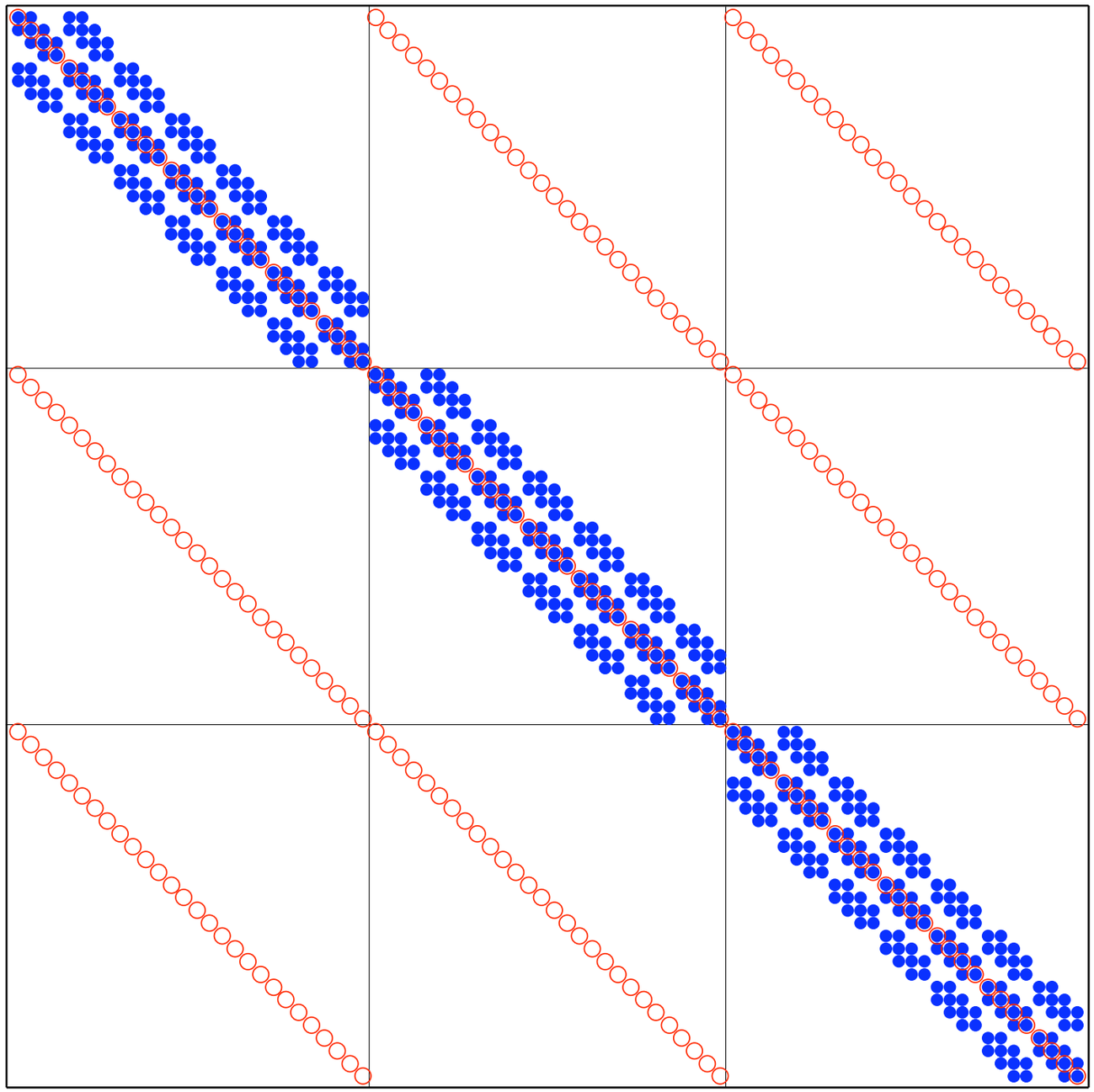
$$\frac{1}{c}\frac{\partial E_g}{\partial t} - \nabla \cdot D_g \nabla E_g = -\sigma_g \left(E_g - B_g(T)\right)$$

$$\rho C_v \frac{\partial T}{\partial t} = \sum_{g=0}^{G} \sigma_g \left(E_g - B_g(T)\right)$$

- Linearizing equations eliminates material energy equation, but groups still coupled
- Common to use 1 or 20 groups, but sometimes up to 100's.

A View of the Full Matrix

- Implemented as an `Epetra_Operator`
- Store each group diffusion block in an `Epetra_FECrsMatrix`
- Store group coupling terms in a G*G array of `Epetra_FEVector`'s
- RHS is an `Epetra_FEVector`
- X is an `Epetra_Vector`



Sandia National Laboratories

Why an Epetra_Operator?

- Need to manipulate sub-operators for physics-based preconditioners.
- Why not TSF...no, wait... Thyra?
  - TSF wasn't ready when I started, and eventually died
  - Or would be doing this in Meros better?
  - I have not taken the time to figure out how to use Thyra.

Epetra_FEVector

- I use Epetra_FEVector for both the RHS (as intended) and to build and store diagonal sub-blocks in operator.
- It is derived from Epetra_MultiVector, not Epetra_Vector
  - Despite this, only can store one vector
  - But need to access it using more complicated multivector methods
- Would like: Epetra_FEMultiVector derived from Epetra_MultiVector and an Epetra_FEVector derived from Epetra_Vector
  - Ease access to single vector
  - More parallel efficient assembly of block diagonals

Sandia National Laboratories

Another Recommendation

- Keep Doxygen comments up to date with what is a good method to use and when
- Example: `AztecOO::recursiveIterate`
  - Claims this is the thing to use for recursive AztecOO calls
  - Was a work around in buggy Aztec code
  - Appears to be no longer needed
  - I spent a while trying to figure this out and finally a call to Mike sorted things out.

Parameter Lists and Defaults

- The `ML_Epetra::SetDefaults` is very nice and is easy to use in production codes.

```
// List to pass solver
Teuchos::ParameterList params;
// Get current best defaults for this matrix type from ML
ML_Epetra::SetDefaultsSA( params );
// Override verbosity since too much is written
params.set("output", 0);
// Get options that user set
Teuchos::ParameterList& user = getUserParams();
// Overwrite defaults set above with user options
Teuchos::ParameterList::ConstIterator i;
for( i = user.begin(); i != user.end(); ++i)
   params.setEntry( i->first, i->second );
```

- As parameter lists get bigger (Stratimikos), need some manageable way to get options into production codes
  - XML files and GUI might be nice for stand alone research codes, but I'm not convinced this is the best for us.

Some Transition Troubles

- Many features had been handled inside Aztec that are not handled by AztecOO.
- One example is matrix scaling
  - It is still possible
    - But it requires doing much more work than passing an option.
- Some of our regression tests still fail with Epetra interface instead of Aztec interface.
- As new packages replace old ones, make sure all functionality is there, or explain why new way is "better"

Overview

- ALEGRA-HEDP overview
- General Observations
- Specific examples motivated by the radiation diffusion
- Conclusions

Sandia National Laboratories

The Bad

- Learning all the Greek names is hard.
- There is either a lot of documentation, or very nearly no documentation for packages.
- One claim of good development model is also a weakness. Developers tend to make packages work for their problem, they sound useful for others, but the (seemingly minor) generalization needed isn't there.
- Pace of development and introduction of new packages is almost dizzying for a mere user. Sometimes it seems like productization of packages is neglected in favor of new features.

The Good

- Lots good and interesting work in Trilinos
  - While I've listed lots of complaints, I still recommend Trilinos to other people.
- Pick-and-choose interoperability is very nice
- This meeting is very useful as "documentation" on how to use the code, and where development is going.
  - Maybe focus more on describing details of current release