ADMCoupling

Ian Hawke, David Rideout

Date: 2003/02/20 13:43:33

Abstract

This thorn allows seamless coupling of evolution and analysis thorns to any thorns which contribute matter terms to the stress energy tensor T_{ab}.

1 Purpose

This thorn is completely trivial (there's already more words in the documentation than in the code). The point is to allow clean coupling of matter thorns and spacetime evolution thorns. By making a spacetime thorn (such as BSSN) a friend of `ADMCoupling` it can know about the variables of the matter thorns (such as Whisky) as long as they are also friends of `ADMCoupling`, and then the appropriate stress energy tensor terms can be included through the `CalcTmunu` interface. This avoids explicit dependencies between the spacetime and matter evolution thorns. Note that we need to do the same for certain analysis thorns, such as `ADMConstraints`.

2 CalcTmunu

2.1 Background

So what is this "`CalcTmunu`", anyway? `CalcTmunu` is a general interface which allows any thorn to 'declare' that it contains matter variables, by adding terms to the components of the stress energy tensor. This is done using the Cactus include file mechanism, which allows thorns to contribute code to include files, which can then be included by any thorn which wishes to use them. For `CalcTmunu` there are two include files[1], `CalcTmunu.inc` and `CalcTmunu_temps.inc`.

In `CalcTmunu.inc`, one can place code of the form

```
Ttt = Ttt + ...
Ttx = Ttx + ...
Tty = Tty + ...
Ttz = Ttz + ...
Txx = Txx + ...
```

[1] There used to be a third, `CalcTmunu_rfr.inc`, which was a Cactus 3 legacy having to do with the rfr scheduling mechanism. You may safely delete any reference to this file.

```
Txy = Txy + ...
Txz = Txz + ...
Tyy = Tyy + ...
Tyz = Tyz + ...
Tzz = Tzz + ...
```

to add terms to the components $T_{\mu\nu}$. Each of these variables is of type CCTK_REAL. (If you omit the Ttt +, Ttx +, etc. from the right hand sides of these assignment statements then you will be assuming that your thorn is the only one which provides matter degrees of freedom, and excluding contributions from other matter thorns which may be activated.) This code will be executed for each point on the grid, whose indices will be stored in the integers i, j, and k. Currently it must be 'fixed form' Fortran code.

CalcTmunu_temps.inc will be included in the variable declaration section for the block of code which contains the

```
#include "CalcTmunu.inc"
```

One can put local temporary variable declarations needed for the code above into this file. The Ttt, Ttx, etc. will be declared within a macro from ADMMacros.

2.2 For matter thorns

To make use of the CalcTmunu interface, simply place the lines

```
INCLUDES HEADER: <MyThorn_CalcTmunu_temps.inc> in CalcTmunu_temps.inc
INCLUDES SOURCE: <MyThorn_CalcTmunu.inc> in CalcTmunu.inc
```

in your thorn's interface.ccl file, and declare your thorn to be friends with ADMCoupling. Then provide the files <MyThorn_CalcTmunu_temps.inc> and <MyThorn_CalcTmunu.inc> somewhere in your thorn's source code. We expect to add a sample matter thorn to CactusEinstein soon, which will illustrate the use of this.

2.3 For thorns which need the stress energy tensor

Spacetime evolution thorns and various analysis thorns (e.g. one which computes the constraints) may need to know the value of the stress-energy tensor. See thorn CactusEinstein/ADM for an example the former. There the macro KSOURCES (see KSOURCES_declare.h and KSOURCES_guts.h) is called from the various time integration source files DoubleLeap.F, IterativeCN.F, etc., which in turn include the CalcTmunu_temps.inc and CalcTmunu.inc files. In this example, the $T_{\mu\nu}$ temporary variables are provided by including CactusEinstein/ADMMacros/src/macro/TRT_declare.h.