McLachlan: An Open Community Code for Numerical Relativity

Peter Diener

Center for Computation and Technology
and
Department of Physics and Astronomy

Louisiana State University

Collaborators: Erik Schnetter, Jian Tao, Frank Löffler.

XiRel Tele-con

Louisiana State University

Baton Rouge

November 17, 2009

Kranc

Kranc is a set of mathematica scripts developed initially by Sascha Husa and Christiane Lechner and currently further developed by Ian Hinder for converting a set of tensorial evolution equations into a complete Cactus thorn.

Originally it was created in order to allow easy experimentation with different formulations of the Einstein equations.

Kranc produces a complete Cactus thorn including the configuration files.

Kranc interfaces with MoL and one of it's main functions is to produce the RHS evaluation routine for the evolution equations.

The user has to use Kranc mathematica routines to define tensors and their properties and how they relate to the Cactus grid functions.

In addition there are routines to define Cactus parameters.

The user defines "Calculations" to operate on the tensors along with scheduling information.

Kranc performs a number of optimizations before the actual C-code is generated.

Kranc currently does the finite differencing for you but we also have an interface to externally defined finite differencing operators (incomplete/untested).

Kranc

An example:

```
Gt[ua,lb,lc] -> 1/2 gtu[ua,ud]
                 (PD[gt[lb,ld],lc] + PD[gt[lc,ld],lb]
                  - PD[gt[lb,lc],ld]),
```

becomes:

```
Gt111  =  khalf*(gtu11*PDstandardNth1gt11
       + 2*(gtu21*PDstandardNth1gt12
       + gtu31*PDstandardNth1gt13)
       - gtu21*PDstandardNth2gt11
       - gtu31*PDstandardNth3gt11);

Gt211  =  khalf*(gtu21*PDstandardNth1gt11
       + 2*(gtu22*PDstandardNth1gt12
       + gtu32*PDstandardNth1gt13)
       - gtu22*PDstandardNth2gt11
       - gtu32*PDstandardNth3gt11);
```

plus 16 similar expressions for the remaining components of `Gt[ua,lb,lc]`.

McLachlan

McLachlan is a Cactus thorn that implements the BSSNOK system of general relativistic spacetime evolution equations.

It is implemented by using Kranc to generate the C-source code from the evolution equations written in tensorial thorn.

The Kranc source for McLachlan is about 800 lines long (excluding comments and empty lines). This expands out to approximately 6000 lines of C-code. Note the actual Kranc source for the RHS equations is just a little over 100 lines.

The advantage of McLachlan compared to the previous code (CCATIE):

1. It is much easier to maintain and extend.

2. High order finite differencing can be generated easily.

3. It was designed to support a mixed MPI/OpenMP programming model.

These were the main reasons for the original decision to write McLachlan.

A Short McLachlan History

The first commit to the git repository was done on November, 4, 2007.

The name was chosen because both Erik and I like the Canadian singer/songwriter Sarah McLachlan.

By November 12, 2007, the main evolution equations were debugged and validated.

By the end of November, 2007 we added matter terms.

First production runs for a distorted rotating black hole using centered finite differencing everywhere were done in spring and summer 2008.

They lead to the publication: Brown et. al., Phys.Rev.D79:044023,2009, which was submitted to gr-qc and PRD in September, 2008.

Up-winding of the shift advection terms were added in May 2008 and redone in a better way (after consulting with Ian Hinder) in April 2009.

In July 2009 Pedro Marronetti was the first external user to download McLachlan with the purpose of introducing one of his students to Cactus/Carpet.

McLachlan Capabilities

McLachlan is a complete implementation of the BSSNOK general relativity spacetime evolution equations including all the standard tricks that ensures stability.

McLachlan uses up-winding for the shift advection terms.

McLachlan supports the standard moving puncture Γ-driver and $1 + \log$ gauges.

McLachlan supports both the ϕ- and W-methods (as of November 12).

McLachlan supports static and radiative outer boundary conditions.

McLachlan supports matter terms through the TmunuBase interface.

McLachlan supports up to 8th order finite differencing.

McLachlan supports OpenMP parallelization through Carpet/LoopControl in addition to MPI parallelization through Carpet.

McLachlan supports multi-block code infrastructures by applying the Jacobian to transform local derivatives to global derivatives (not tested yet).

Why McLachlan?

In the beginning McLachlan was mainly meant to be a replacement code for the CCATIE code with the advantages listed earlier.

However, it soon became apparent that it could be used as a community code as well and that it potentially (with some work) could be useful for the following purposes as well:

1. Be an almost black box GR code to be used at summer schools or for graduate student projects.

2. Serve as an easy starting points for Newtonian astrophysics groups looking to add relativistic space time evolution to their codes or for start-up numerical relativity groups.

In addition, Kranc and McLachlan together, can serve as a research tool for developing techniques for architecture specific optimizations at the loop level.

Future Plans for McLachlan

To make McLachlan an even more useful community tool we are in the process of developing a self-contained tutorial package containing instructions for downloading, compiling and running the code with many and well tested “real physics” parameter files as examples of its use as well as expanded documentation.

Current plans for expanding the user base, involves the relativity group at KISTI in South Korea and Stephan Rosswog at Jacobs University, Bremen.

We are going to test, and if necessary improve, the performance of the code to ensure that it is production ready.

We will continue to add features and improvements as necessary to support the research projects being done with the code.

In the longer term we are interested in experimenting with using Kranc for architecture specific optimizations.

Download

McLachlan is freely available and can be downloaded from a git-repository.

`git clone git://carpetcode.dyndns.org/McLachlan.git`

McLachlan will soon be available as part of the einsteintoolkit at a new location:

`einsteintoolkit.org`

Information will be posted at:

`www.einsteintoolkit.org`