Cactus: Advanced Topics
Kranc

Kranc Assembles Numerical Code

- A tool for generating thorns
- Original version by S. Husa, I. Hinder, C. Lechner
- Used for many scientific papers
- Allows user to specify continuum problem without regard for implementation
Kranc

Kranc Assembles Numerical Code

- A tool for generating thorns
- Original version by S. Husa, I. Hinder, C. Lechner
- Used for many scientific papers
- Allows user to specify continuum problem without regard for implementation
A tool for generating thorns
Original version by S. Husa, I. Hinder, C. Lechner
Used for many scientific papers
Allows user to specify continuum problem without regard for implementation
A tool for generating thorns
Original version by S. Husa, I. Hinder, C. Lechner
Used for many scientific papers
Allows user to specify continuum problem without regard for implementation
Minimally, a Kranc script contains
- Cactus Group Definitions
- A set of Calculations

A calculation in Kranc is written using Mathematica (a symbolic programming language)
- Example: \( \text{dot}[h[la,lb]] \rightarrow -2 \alpha K[la,lb] \)
- \( \text{dot} \) means “time derivative”
- \( h[la,lb] \) and \( K[la,lb] \) are matrices
Minimally, a Kranc script contains

- Cactus Group Definitions
- A set of Calculations

A calculation in Kranc is written using Mathematica (a symbolic programming language)

- Example: \( \text{dot}[h[la,lb]] \rightarrow -2 \alpha K[la,lb] \)
- dot means “time derivative”
- \( h[la,lb] \) and \( K[la,lb] \) are matrices
Minimally, a Kranc script contains

- Cactus Group Definitions
- A set of Calculations

A calculation in Kranc is written using Mathematica (a symbolic programming language)

- Example: $\text{dot}[h[la,lb]] \to -2 \alpha K[la,lb]$
- dot means “time derivative”
- $h[la,lb]$ and $K[la,lb]$ are matrices
Minimally, a Kranc script contains
- Cactus Group Definitions
- A set of Calculations

A calculation in Kranc is written using Mathematica (a symbolic programming language)
- Example: \( \text{dot}[h[la,lb]] - > -2 \alpha K[la,lb] \)
  - dot means “time derivative”
  - \( h[la,lb] \) and \( K[la,lb] \) are matrices
Minimally, a Kranc script contains
- Cactus Group Definitions
- A set of Calculations

A calculation in Kranc is written using Mathematica (a symbolic programming language)
- Example: \( \text{dot}[h[la,lb]] - 2 \alpha K[la,lb] \)
  - dot means “time derivative”
  - \( h[la,lb] \) and \( K[la,lb] \) are matrices
Minimally, a Kranc script contains
- Cactus Group Definitions
- A set of Calculations

A calculation in Kranc is written using Mathematica (a symbolic programming language)
- Example: \( \text{dot}[h[la,lb]] \rightarrow -2 \alpha K[la,lb] \)
- dot means “time derivative”
- \( h[la,lb] \) and \( K[la,lb] \) are matrices
Kranc

Kranc Assembles Numerical Code

- Included in Cactus-OS distribution: Cactus/repos/Kranc.
- The state-of-the-art numerical relativity code, produced by Kranc: Cactus/arrangements/McLachlan.
- For more, see: http://kranccode.org
Kranc

Kranc Assembles Numerical Code

- Included in Cactus-OS distribution: Cactus/repos/Kranc.
- The state-of-the-art numerical relativity code, produced by Kranc: Cactus/arrangements/McLachlan.
- For more, see: http://kranccode.org
Kranc

Kranc Assembles Numerical Code

- Included in Cactus-OS distribution: Cactus/repos/Kranc.
- The state-of-the-art numerical relativity code, produced by Kranc: Cactus/arrangements/McLachlan.
- For more, see: http://kranccode.org
The Simulation Factory:

- Command-line tools for setting up Cactus distribution and managing simulations on a variety of supercomputers, including most of the TeraGrid machines.
- Captures the best practices of experienced users, ensuring repeatable and well-documented scientific results.
- Developed by Erik Schnetter and Michael Thomas.
- Included in Cactus-OS distro: Cactus/simfactory.
- For more, see: http://simfactory.org
Open-source code for the numerical relativity and relativistic astrophysics.

**Capabilities:**

- Accurate evolutions of vacuum spacetimes using Einstein equations.
- Relativistic hydrodynamics, based on the public version of Whisky code.
- Initial data solvers: single and binary BHs and relativistic stars.
- Analysis: finds BH horizon, calculates spacetime curvature, extracts gravitational waves.
- More information: http://einsteintoolkit.org
In Eclipse, open "Mojave > Edit variables..." and set parameters for precompiled version of the Einstein toolkit and a parameter file:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>par</td>
<td>par/static_tov.par</td>
</tr>
<tr>
<td>thornlist</td>
<td>thornlists/einsteintoolkit.th</td>
</tr>
<tr>
<td>config</td>
<td>et</td>
</tr>
<tr>
<td>sim.name</td>
<td>tov_star1</td>
</tr>
</tbody>
</table>

Select "Mojave > Create", then "Mojave > Run" to create and run an evolution of a relativistic TOV star!
Simple visualization with gnuplot

```
cd ~/simulations/tov_star/output-0000
gnuplot> p "static_tov/hydrobase::rho.maximum.asc" u 2:3 w l
gnuplot> p "static_tov/hydrobase::alpha.x.asc" u 10:13 i 0 w l
```
3D output and visualization

To enable 3D output, copy `static_tov.par` to a new parameter file (say, `static_tov_h5.par`) and append the following lines, then use it to create and run a new simulation:

```
IOHDF5::out_criterion = time
IOHDF5::out_dt = 100.0
IOHDF5::out_vars = "
    grid::coordinates{out_every=10000000}
    HydroBase::rho
"
```
Using VisIt

VisIt is an interactive 3D visualization tool.

- Open VisIt
- Open "File > Open File...", navigate to the output of your simulation, or use:
  
  ~/simulations/tov_star/output-0000/static_tov

- Select filter: *.h5
- Select the star density: rho.h5

- Add contour plot for the second refinement level (HYDROBASE::rho it=0 tl=0 rl=2)
- Press "Draw" button.

VisIt website: https://wci.llnl.gov/codes/visit/
Remote steering/monitoring tool

- Mini web server implemented as a thorn
- Connect to a running simulation with a web browser
- View and steer simulation parameters
- Basic remote visualization
Adding HTTPD to your configuration

Using HTTPD is simple:

- Add the following thorns to your thornlist (file ~/Cactus/thornlists/BadWaves.th):
  
  CactusConnect/HTTPD
  CactusConnect/HTTPDExtra
  CactusConnect/Socket

- Enable these thorns in your parfile (for example, in ~/Cactus/pars/BadWaveAMR.par):

  ActiveThorns = "HTTPD HTTPDExtra Socket"
Sample HTTPD session

Cactus Simulation

This browser is connected to a Cactus simulation which contains a web server thorn. This thorn provides information and control for the simulation.

Before controlling any features of the simulation, users must authenticate.

Available options:

Cactus Control

Simulation:

- Flesh version 4.0.17
- Flesh compiled on Jul 12
Sample HTTPD session

Control and Status Page

This page is the control center for interacting with the current simulation. It is possible to steer certain parameters, as well as pause, restart, or terminate the simulation.

Run Control

Select if the run should be paused, running normally, or terminated.
You may also single step to the next iteration.

○ PAUSE ○ RUN ○ TERMINATE  STEP

OK  Reset

Run Until
Check/Modify Parameters

Thorn CarpetIOASCII

Parameters in Cactus can be either fixed or steerable. Steerable parameters are those which can be modified during a simulation. To change steerable parameters, edit the values and press the submit button. Before applying the new parameter value, Cactus will check that the parameter lies in the allowed range. Note that there is currently no log of parameters which have been modified.

The tables below show the parameter name, current value and description. The default value of each parameter is shown in brackets at the end of the description.

Steerable Parameters

The following parameters are steerable, but you do not have authorization to change them. To change parameters you must first...
CCTK Timer Information

(changed values since last refresh are in bold characters)

Timers which are associated with schedule bins

<table>
<thead>
<tr>
<th>Thorn Name</th>
<th>Description</th>
<th>gettimeofday (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carpet</td>
<td>MultiModel_Startup</td>
<td>0.003230</td>
</tr>
<tr>
<td>ML_BSSN_Helper</td>
<td>ML_BSSN_SetGroupTags</td>
<td>0.000096</td>
</tr>
<tr>
<td>Carpet</td>
<td>Driver_Startup</td>
<td>0.000001</td>
</tr>
<tr>
<td>HTTPD</td>
<td>HTTP_StartServer</td>
<td>0.830596</td>
</tr>
<tr>
<td>HTTPD</td>
<td>HTTP_FirstServ</td>
<td>0.000016</td>
</tr>
<tr>
<td>HTTPDEtra</td>
<td>HTTPUTILS_Startup</td>
<td>0.000062</td>
</tr>
<tr>
<td>IOUtil</td>
<td>IOUtil_Startup</td>
<td>0.000013</td>
</tr>
<tr>
<td>CarpetIOScalar</td>
<td>CarpetIOScalar_Startup</td>
<td>0.000000</td>
</tr>
</tbody>
</table>
General Purpose Graphics Processing Units (GPGPUs) are high-performance graphical processors that can be used to accelerate a wide range of applications. (CUDA C Programming Guide Version 3.2)
Compared to CPU, GPU devotes more transistors to data processing rather than data caching and flow control (CUDA C Programming Guide Version 3.2)
• MPI + CUDA programming in Cactus is made simple.
CaCUDAViz

- The volume rendering of a scalar wave spreading from the center using CUDA on a GPU, while the rendering parameters are steerable at runtime via the Cactus web server.

![Viewport](#)