Analele Universității de Vest din Timişoara
Vol. XLVIII, 2006
Seria Fizică

NUMERICAL SIMULATIONS IN RICCI FLOW WITH THE CACTUS CODE

Dumitru N. Vulcanov

West University of Timişoara, Department of Theoretical and Computational Physic
B-dul V. Parvan no. 4, Timisoara, Romania

Abstract
We shall present some facts and results concerning the implementation of a new thorn for the Cactus code for evolving numerically the Ricci flow equation.

1. Introduction

In differential geometry, the Ricci flow is a process which deforms the metric of a Riemannian manifold in a manner formally analogous to the diffusion of heat, thereby smoothing out irregularities in the metric ([1]).

If we consider the metric tensor (and the associated Ricci tensor) to be functions of a variable which is usually called "time" (but which may have nothing to do with any physical time), then the Ricci flow may be defined by the geometric evolution equation

$$\frac{\partial g_{ij}}{\partial t} = -2R_{ij} \qquad (1)$$

For doing numerical simulations with (1) we need certain specific steps, namely :

- (i) a good initial metric ("initial data") ;
- (ii) "something" to compute the Ricci tensor components ;
- (iii) a good finite differences schema - both for spatial derivatives and time evolution ;
- (iv) "something" for visualizing the results.

For (i) we need surfaces of revolution (2 and/or 3 dimensional). For (ii) it seems that is necessary some algebraic computing manipulations. This means different evolution equations for every

metric we are processing, and will complicate the step (iii) as we need separate codes for every case.
How about an explicit schema for the time evolution if we know Ricci tensor components on every point on the grid? This is the main idea we shall exploit for solving step (iii) problems.
For visualizing (iv) we can use Gnuplot, Maple (again ?) or a “self-made-home-edition” software ? This will be a problem for the future investigations.

2. Cactus ...ing idea

We arrived at the idea of using the Cactus code for dealing numerically with Ricci flow equations. The motivations are :

- Cactus code provides an environment for running any piece of code for solving PDE in F77, F90, F95, C, C++, perl ... or any combinations imaginable !
- Cactus code has an arrangement, CactusEinstein where there are codes for computing Ricci tensor components and Ricci scalar on a 3-D numerical grid !
- Controlling the input and output data and the timestep evolution is done in Cactus through standard routines;
- Cactus code is free !!! It takes 10 minutes to download (see **http://www.cactuscode.org**)

1. Cactus is highly portable on all Unix machines, even on Windows, under cygwin (see **http://www.cygwin.org**)!
2. Cactus is parallelisable - running on supercomputers and clusters is an easy task - and we will need this !
3. Main reason : the author has a long experience with it !

We shall describe in some details the way we done this.

3. A new thorn : Riccifl

Based on existing Exact thorn ([5]), we structured a new thorn called Riccifl thorn. Remember that “thorns” are generically called the codes and packages implemented in the Cactus code environment. The structure we used is similar to the Exact thorn, namely
- it has a special routine for “injecting” the metric components at the initial time and at every time step, after the evolution (injecting.F);

- every metric can be implemented through a single file, in the ../src/metrics directory;
- it activates and uses, at all time steps routines from Einstein thorn (in ricci.c routine), namely : ADMAnalysis.h, RICCI_declare.h, RICCI_guts.h, RICCI_undefine.h ;
- has an evolution routine, (evolution.F) for processing the Ricci flow equation 1.

As the evolution equation 1 has the form $\partial_t g(...,t) = Ric(...,t)$ we numerically evolve it using the approximation ([2]) $(g_{ij})_k^{n+1} = (g_{ij})_k^n + \Delta t (R_{ij})_k^n$

where n controls the timestep and i the grid point, $(1 \leq k \leq n)$, N being the number of points on the spatial grid.

That's was all we need to code, as the components of R_{ij} are provided by ricci.c routine through Einstein thorn. The rest is usual Cactus stuff, using the standard routines :
- initialize.F for reading the metric components at initial time;
- trace.c for computing the trace of Ricci tensor = Ricci scalar;
- interface.ccl for declaring/defining the grid functions doing the necessary link with the rest of Cactus (Einstein, ADMBase, IO...);
- param.ccl for declaring the necessary parameters in Riccifl thorn and across the Cactus;
- schedule.ccl for scheduling the routines at the time steps in Cactus and for the necessary storage declarations across the Cactus code.

We implemented two types of metrics for 3-dimensional revolution surfaces, the "Rubinstein" 3-surfaces (see [3]), the corseted 3-spheres as proposed by Garfinkle (see [2]) We tried also a 3-torus type T3 metric (see [2]).

4. Troubles and problems to solve

From the very beginning we faced several difficulties, specific to our problem:
- the Cartesian coordinates problem : Cactus is using only Cartesian coordinates on the numerical grid, so we need to translate the metrics in Cartesian coordinates !
- The above problem caused also troubles at the boundaries of the grid. We have to be very careful with this as the 3 surface is embedded in a Cartesian grid.

These problems can be solved mainly in two ways. First a careful and clean receipt for the boundaries values - need new routines to be coded soon! The best solution will be a multipach grid ! This can be the ultimate solution ! We will investigate it after a good study of the method proposed by J. Thornburg at AEI (see **http://numrel.aei.mpg.de**).

The numerical instabilities pointed out by other authors (see [2], [3]) were obvious also for our code. In the future we will try to solve this by using new Ricci flow equations (normalized or DeTurck version) or applying special methods as filtering/reparametrising or spectral methods (see [4]). However, an enough small time step can be a temporary solution, trying also to skip the singularities as they approach. Meanwhile we will do, in parallel, simulations with the Maple platform, in certain special cases to have more informations about the numerical behavior of the equations.

5. Partial and preliminary results

In the figures we included here we plotted some of our preliminary results obtained by processing numerically the equation (1) with our Riccifl thorn in Cactus code. The first two figures represent the g_{xx} component of the corseted sphere geometry as described by Garfinkle and Isenberg ([2]), at the initial step-time and after 8 iterations. It is obvious the fast evolution of the metric, by contracting to a smaller space region to a final singularity. The second group of two pictures shows off the time evolution of the Ricci scalar and determinant of the metric in the same geometry, in a logarithmic scale.

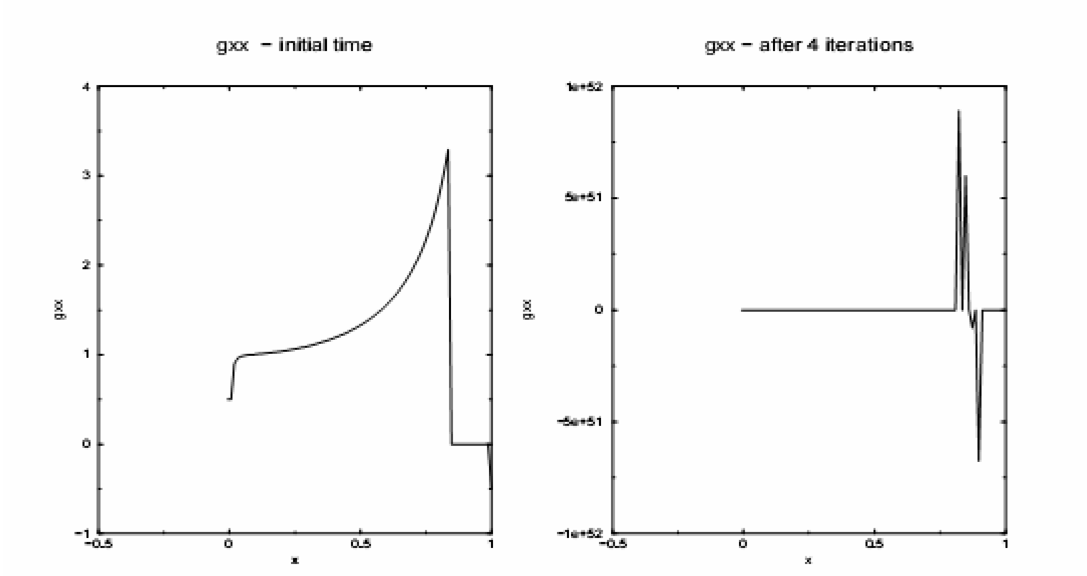

Fig. 1 Corseted sphere - Metric component g_{xx} at $t = 0$ and $t = 0.00008$ on a grid with 80^3 points; $\lambda = 0.001$

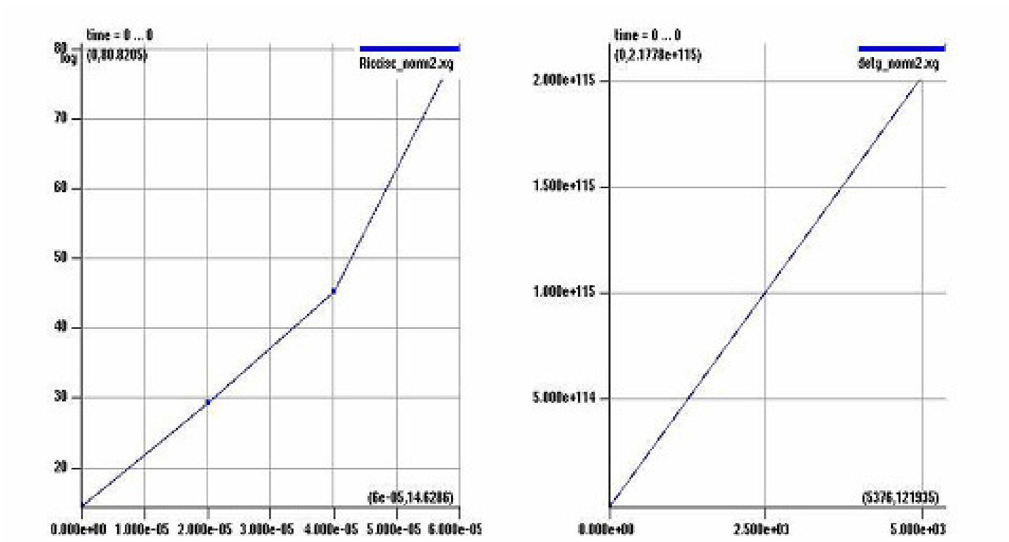

Fig. 2 Corseted sphere - Time evolution of L_2 *norm for Ricci scalar and* $\det(g)$ *on a grid with* 80^3 *points;* $\lambda = 0.001$

These results are encouraging proving the possibility of doing numerical simulations with Ricci flow equations in the Cactus code. Of course the instabilities and the problems pointed out above are obvious, imposing several special treatments in the future investigations. The simulations were done on a single processor machine (a Pentium IV Celeron at 3.2 GHz with 1 GB RAM) with FreeBSD 6.2 operating system. Similar results were obtained on a Cactus version installed on a Windows XP machine with cygwin environment (actually a Toshiba Satellite 100A laptop). In both cases we used he new G95 Fortran compiler.

6. Acknowledgments

This work was supported under Grant CEEX nr. 2-CEx06-11-92/19.09.2006 with the Romanian Ministry of Education and Research.

References

[1] R.S. Hamilton, J. Diff. Geom., 17, 255 (1982), B. Chow, *A Survey of Hamilton's Program for the Ricci flow on 3-Manifolds*, math.DG/0211266 (2002)

[2] D. Garfinkle, J. Isenberg, math.DG/0306129 (2003)

[3] J.H. Rubinstein, R. Sinclair, Exp.Math., 14, nr.3 (2005), math.DG/0506189

[4] D. DeTurck, J. Diff. Geom., 18, 157 (1983)

[5] D.N. Vulcanov, M. Alcubierre, IJMP-C, vol. , No. 7, 837 (2002)