# Newtonian approximation in Causal Dynamical Triangulations

Adam Getchell

## Contents

## 1 Motivation

### 1.1 Newton's Law of Gravitation from General Relativity

- Can we recover $F = -\frac{Gm_1 m_2}{r^2}$ from CDT?

- Do we have a sensible notion of "mass" in CDT?

- Semi-classical approximations not yet completely convincing [8]

### 1.2 Previous Work

- Separation between two objects $\gg$ Schwarzschild radius

- Self-fields not excluded

- Cylindrical symmetry

- Object size $\ll$ separation

We will see that a "strut" holds objects apart.

Following Chou [2], the most generally cylindrically symmetric static metric is:

$$ds^2 = g_{00}dt^2 - \left( g_{11}\left(dx^1\right)^2 + 2g_{12}dx^1dx^2 + g_{22}\left(dx^2\right)^2 \right) - g_{33}d\phi^2 \qquad (1)$$

where $x^1, x^2$ are any two coordinates in the meridianal (vertical) plane containing the z-axis.

We furthermore assume that $g_{\mu\nu} = f\left(x^1, x^2\right)$

For positive definite quadratic differential forms of two variables such as explicit values of $g_{11}, g_{12}$, and $g_{22}$ from the parenthetical part of Equation (1), one can make a real, single-valued, continuous transformation from $x^1$ and $x^2$ to $u$ and $v$ by:

$$x^1 = x^1(u,v), \, x^2 = x^2(u,v) \qquad (2)$$

where $J = \left[ \partial(x^1, x^2)/\partial(u,v) \right] \neq 0$ such that:

$$g_{11}\left(dx^1\right)^2 + 2g_{12}dx^1dx^2 + g_{22}\left(dx^2\right)^2 = e^{2m}\left(du^2 + dv^2\right) \qquad (3)$$

Equation (1) becomes:

$$ds^2 = e^{2v}dt^2 - e^{2m}(du^2 + dv^2) - e^{2n}d\phi^2 \qquad (4)$$

Explicitly, we then have the metric:

$$g_{\mu\nu} = \begin{pmatrix} e^{2v}dt^2 & 0 & 0 & 0 \\ 0 & -e^{2m}du^2 & 0 & 0 \\ 0 & 0 & -e^{2m}dv^2 & 0 \\ 0 & 0 & 0 & -e^{2n}d\phi^2 \end{pmatrix} \qquad (5)$$

Using [1]:

$$\Gamma^{\lambda}_{\mu\nu} = \frac{1}{2}g^{\lambda\sigma}\left(\partial_\mu g_{\nu\sigma} + \partial_\nu g_{\sigma\mu} - \partial_\sigma g_{\mu\nu}\right) \qquad (6)$$

We can get the non-zero Christoffel connections as:

$$
\begin{aligned}
\Gamma^1_{00} &= e^{2(v-m)}\partial_u v & \Gamma^2_{22} &= \partial_v m \\
\Gamma^1_{33} &= -e^{2(n-m)}\partial_u n & \Gamma^2_{21} &= \partial_u m \\
\Gamma^1_{22} &= -\partial_u m & \Gamma^2_{11} &= -\partial_v m \\
\Gamma^1_{21} &= \partial_v m & \Gamma^3_{32} &= \partial_v n \\
\Gamma^1_{11} &= \partial_u m & \Gamma^3_{31} &= \partial_u n \\
\Gamma^2_{00} &= e^{2(v-m)}\partial_v v & \Gamma^0_{02} &= \partial_v v \\
\Gamma^2_{33} &= -e^{2(n-m)}\partial_v n & \Gamma^0_{01} &= \partial_u v
\end{aligned}
\qquad (7)
$$

Using:

$$R^{\rho}_{\sigma\mu\nu} = \partial_\mu \Gamma^{\rho}_{\nu\sigma} - \partial_\nu \Gamma^{\rho}_{\mu\sigma} + \Gamma^{\rho}_{\mu\lambda}\Gamma^{\lambda}_{\nu\sigma} - \Gamma^{\rho}_{\nu\lambda}\Gamma^{\lambda}_{\mu\sigma} \tag{8}$$

$$R_{\mu\nu} = R^{\lambda}_{\mu\lambda\nu} \tag{9}$$

$$R = R^{\mu}_{\mu} = g^{\mu\nu}R_{\mu\nu} \tag{10}$$

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} \tag{11}$$

And solving for the Einstein Field Equations in a vacuum (i.e. $G_{\mu\nu} = 0$) we get:

$$G_{00} = -e^{2(\nu-m)}\left((\partial_\nu n)^2 + \partial_\nu^2 m + \partial_\nu^2 n + (\partial_u n)^2 + \partial_u^2 m + \partial_u^2 n\right) = 0 \tag{12}$$

$$\begin{aligned}G_{11} = &-\partial_\nu m\partial_\nu n + (\partial_\nu n)^2 - \partial_\nu m\partial_\nu \nu + \partial_\nu n\partial_\nu \nu + (\partial_\nu \nu)^2 \\ &+ \partial_\nu^2 n + \partial_\nu^2 \nu + \partial_u m\partial_u n + \partial_u m\partial_u \nu + \partial_u n\partial_u \nu = 0\end{aligned} \tag{13}$$

$$\begin{aligned}G_{21} = &\partial_\nu n\partial_u m + \partial_\nu \nu\partial_u m + \partial_\nu m\partial_u n - \partial_\nu n\partial_u n + \partial_\nu m\partial_u \nu \\ &- \partial_\nu \nu\partial_u \nu - \partial_u\partial_\nu n - \partial_u\partial_\nu \nu = 0\end{aligned} \tag{14}$$

$$\begin{aligned}G_{22} = &\partial_\nu m\partial_\nu n + \partial_\nu m\partial_\nu \nu + \partial_\nu n\partial_\nu \nu - \partial_u m\partial_u n + \partial_u m\partial_u \nu \\ &+ (\partial_u n)^2 - \partial_u m\partial_u \nu + \partial_u n\partial_u \nu + (\partial_u \nu)^2 + \partial_u^2 n + \partial_u^2 \nu = 0\end{aligned} \tag{15}$$

$$G_{33} = e^{2(n-m)}\left((\partial_\nu \nu)^2 + \partial_\nu^2 m + \partial_\nu^2 \nu + (\partial_u \nu)^2 + \partial_u^2 m + \partial_u^2 \nu\right) = 0 \tag{16}$$

Let:

$$\chi = n + \nu \tag{17}$$

Adding together Eqns. (12) and (16) gives:

$$\partial_u^2\chi + \partial_\nu^2\chi + (\partial_u\chi)^2 + (\partial_\nu\chi)^2 = 0 \tag{18}$$

Setting:

$$\Phi = e^{\chi} = e^{n+\nu} \tag{19}$$

We recover Laplace's equation in the $uv$-plane:

$$\partial_u^2\Phi + \partial_\nu^2\Phi = 0 \tag{20}$$

The remaining equations are used for boundary conditions on Laplace's equation. In general, for a metric of the form:

$$ds^2 = e^{2\psi}dt^2 - e^{-2\psi}\left[e^{2\omega}\left(dr^2 + dz^2\right) + r^2 d\phi^2\right] \tag{21}$$

We have general solutions:

$$\nabla^2\psi = \partial_r^2\psi + \frac{\partial_r\psi}{r} + \partial_z^2\psi \tag{22}$$

$$d\omega\left[\psi\right] = r\left[\left((\partial_r\psi)^2 - (\partial_z\psi)^2\right)dr + 2\partial_r\psi\partial_z\psi dz\right] \tag{23}$$

Note that Eq(4) can be recovered from Eq(21) by substituting $\psi = v, m = \omega - \psi$, and $e^{2n} = r^2 e^{-2\psi}$.

The solution of Eq(22) and Eq(23)for a point particle of mass m at $z = z_0$ is given by (explain "point" in Schwarzschild solution, check for singularities):

$$\psi = -\frac{m}{R} \tag{24}$$

$$\omega = -\frac{m^2 r^2}{2R^4} \tag{25}$$

$$R = \sqrt{r^2 + (z - z_0)^2} \tag{26}$$

What is meant by "point" particle? To find out, let's transform to the Schwarzschild equation:

$$ds^2 = \left(1 - \frac{2GM}{r}\right)dt^2 - \frac{1}{\left(1 - \frac{2GM}{r}\right)}dr^2 - r^2\left(d\theta^2 + \sin^2\theta d\phi^2\right) \tag{27}$$

using: (TODO: fill in transforms)
For n point particles we have [6]:

$$\psi = -\sum_{j=1}^{N}\frac{m_j}{R_j} \tag{28}$$

$$\omega = -\frac{r^2}{2}\sum_j \frac{m_j^2}{R_j^4} + \sum_{j \neq k}\frac{m_j m_k}{(z_j - z_k)^2}\left[\frac{r^2 + (z - z_j)(z - z_k)}{R_j R_k} - 1\right] \tag{29}$$

$$R = \sqrt{r^2 + (z - z_j)^2} \tag{30}$$

4

## 1.3 Additional issues

# 2 Applications to Causal Dynamical Triangulations

## 2.1 Preliminaries

A simplex is a generalization of a triangle to arbitrary dimension. For example, a 2-simplex is a triangle, a 3-simplex is a tetrahedron, and a 4-simplex is a pentachoron. Topologically, an n-simplex is equivalent to an n-ball; that is, an n-dimensional manifold with boundary.

An n-dimensional simplex has $n + 1$ points or *vertices*. A convex hull, or minimal convex set of these points is the *m-face* of the *n-simplex*. Thus, a vertex is a *0-face*, and an edge between two vertices is the *1-face*. We can extend this notation to *2-faces* (triangles), *3-faces* (tetrahedrons), *4-faces* (pentachorons). We will not, at present, consider simplices of dimension higher than $n = 4$, but this generalization gives us a useful way to reason about higher dimensional spaces.

The number of *m-faces* on our *n-simplex* is given by the binomial coefficient as:

$$\left( \begin{array}{c} n+1 \\ m+1 \end{array} \right) \tag{31}$$

Thus, our pentachoron has 5 vertices, 10 edges, 10 faces (triangles), 5 cells (tetrahedrons), and 1 4-face, itself.

A given face can be shared by another simplex. By requiring that [9]:

- Every face of a simplex *K* is in *K*, and

- The intersection of any two simplices of K is a face of each of them

We build up a useful structure called a simplicial complex. Informally, this is a space with a triangulation. Formally, simplicial complexes have only been proven for spaces of dimension $d \leq 3$. A simplicial complex has a well-defined homology (simplicial homology) which is easy to compute.

## 2.2 Code Correctness

Implementing CDT in computer code is non-trivial. As a first significant step, an independent implementation of CDT has given similar results to the original work [5]. We would like to build on Kommu's implementation using Literate Programming [4] coupled with Test Driven Development specific to the programming language used [7]. This provides for the codebase to be better understood by researchers wishing to replicate results or expand the capabilities of the code, and provides inherent integrity checks apart from "it produced what we expected". Such methodology will be critical to expanding the performance of the code by using such techniques as parallel processing and highly optimized algorithms. The adage of "Make it work, make it right, make it fast" applies.

The first building block of the code are the simplexes themselves. Using the known properties of simplicial complexes, we can provide for a series of checks that validate

that simplices are being constructed correctly. Such checks will provide useful test cases when the underlying implementation of simplex data structures and moves are changed.

## 2.3 Data structures

## 2.4 Issues

- Extrinsic Curvature *(To Do)*

- Imposing conditions of separation

- Checking that separation >> Schwarzschild radius

- Imposing cylindrical symmetry

## 2.5 CDT Algorithm

*(To Do: insert graphics)*

- [(2,8): (1,4) + (4,1) →8 simplices] + inverse = +2 moves

- [(4,6): ()+()+()+()→6 simplices] + inverse = +2 moves

- [(2,4): two varieties of ()+()→4 simplices], self-inverse = +2 moves

- [(3,3): two varieties of ()+()+()→3 simplices] + inverse = +4 moves

**10** moves in all (*Check!*)
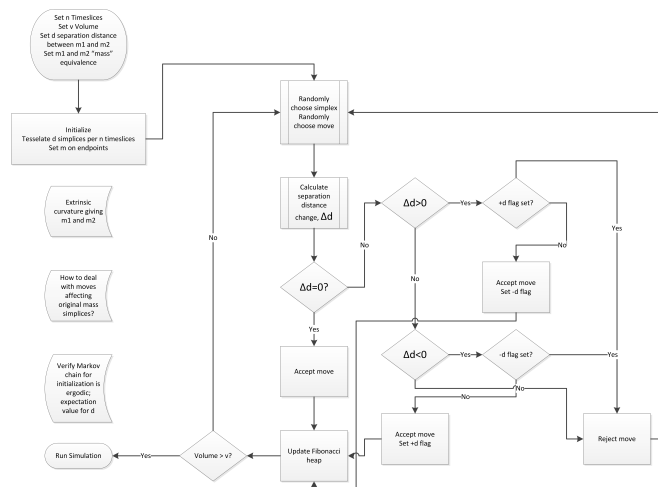


Dijkstra's Algorithm [3]

Solves single-source shortest-path problems on weighted, directed graph G=(V,E) of non-negative edge lengths

- Greedy algorithm

- Proven to be correct

- Complexity

    - $O(V^2)$ naively using adjacency list
    - $O(E \lg V)$ using priority queue iff all vertices reachable from source
    - $O(V \lg V + E)$ using Fibonacci heap (more relaxation calls than extract-min calls)

- Issue: confine edge length algorithm to particular time-slice

- Solution: Store Fibonacci heap of simplices per time-slice

    - Each simplex has 5 neighbors, so more compact than adjacency matrix

    - How to deal with moves affecting original "mass" simplices

    - How to create a 4d cylinder of height z=d

    - Verify Markov chain for initialization is ergodic

    - Calculate <d>

# 3   Summary

- Insert mass equivalence via extrinsic curvature

- Insert strut by enforcing separation distance

- Filter moves which alter separation distance via Markov chain

- Outlook

    - Write code!
    - Check Extrinsic Curvature
    - Compare results

# References

[1] Sean Carroll. *Spacetime and Geometry: An Introduction to General Relativity*. Benjamin Cummings, September 2003.

[2] P. Y. Chou. The Gravitational Field of a Body with Rotational Symmetry in Einstein's Theory of Gravitation. *American Journal of Mathematics*, 53(2):289–308, April 1931. ArticleType: research-article / Full publication date: Apr., 1931 / Copyright © 1931 The Johns Hopkins University Press.

[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*. The MIT Press, 2nd edition, September 2001.

[4] D. E. Knuth. Literate Programming. *The Computer Journal*, 27(2):97 –111, January 1984.

[5] Rajesh Kommu. A Validation of Causal Dynamical Triangulations. *arXiv:1110.6875*, October 2011.

[6] Patricio. S Letelier and Samuel R Oliveira. Superposition of weyl solutions: The equilibrium forces. *arXiv:gr-qc/9710122*, October 1997. Class.Quant.Grav. 15 (1998) 421-433.

[7] Amit Rathore. *Clojure in Action*. Manning Publications, 1 edition, November 2011.

[8] T. Trzesniewski. Analysis of the Semiclassical Solution of CDT. *arXiv:1102.4643*, February 2011.

[9] Eric W. Weisstein. Simplicial Complex. *MathWorld - A Wolfram Web Resource*.