Brandeis University

Department of Physics

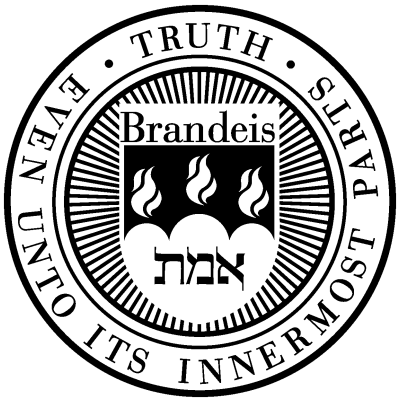
Senior Thesis

Multiflows and Entanglement Entropies in AdS/CFT

Author:
Jesse Held

Advisor:
Dr. Matthew Headrick

May 10, 2019



Abstract

A recent development in the study the AdS/CFT correspondence is that of `bit threads,' which describe the physical distribution of entanglement in CFTs while simultaneously computing entanglement entropies. Equivalently, we may work with a set of divergenceless vector fields with a collective norm bound called a `Multi flow.' Via analytic ansatz and numerical computation, we demonstrate that the multi flow formalism fails to compute entanglement entropies for crossing boundary regions. We also present two proofs for the assertion that in the absence of crossing boundary regions there exists a multi flow which computes the entropy of entanglement for an arbitrary number and arrangement of boundary regions.

Contents

1 Introduction

The apparent emergence of duality between Conformal Field Theories (CFTs) and Anti-de Sitter (AdS) Space of one higher dimension is a topic of great interest in high energy physics. A CFT

is a quantum field theory which exhibits conformal symmetry and AdS is a maximally symmetric space-time with a constant negative curvature. The equivalence between these systems is referred to as `*holographic*' since the AdS is of one higher dimension than the CFT. This allows us to think of the CFT as existing on the boundary of the holographic AdS, which we refer to as *the bulk*. This duality, referred to as the AdS/CFT Correspondence, gives theorists a unique framework in which to explore a number of open questions in a variety of fields, from condensed matter to quantum gravity.

The work presented here concerns itself primarily with the study of information and entanglement on states of Conformal Field Theories. Our main interest is in developing methods of calculating the entropy of entanglement between different spacial regions of CFTs, giving us a means of probing the information structure of holographic states and determining some shared properties of all holographic states. As the study of the AdS/CFT Correspondence has progressed, the Ryu-Takayanagi (RT) formula has become widely accepted as a means of calculating the entanglement entropy. The RT formula relates the entropy of a region of the boundary CFT to the area of the minimal surface homologous to this region in the bulk AdS space. The RT formula, while useful, gives us a very limited intuitive sense of the physical structure of the underlying state.

In order to increase our understanding of the information structure of holographic states, consider the following picture:

Threads with a certain cross-sectional area are placed onto the some region A of the boundary CFT and are allowed to run through the bulk and onto some point of the boundary within the complement of A. In this picture, the minimal surface referenced by the RT formula above acts as a bottle-neck for such threads and thus the maximum number of such threads which can connect A to its compliment also gives us the entropy by revealing the area the minimal bulk surface homologous to A in an indirect fashion. Each individual thread can be thought of as entangling two separate points on the CFT and each carries an entropy equivalent to that of a maximally entangled Bell Pair, and thus we refer to these threads as *bit threads*.

The equivalence between the RT formula and this bit thread picture is realized as a continuum version of the *max- ow min-cut theorem* from graph theory by using techniques from the theory of convex optimization. In particular, the strong duality of convex optimization problems allows us to equate the RT formula to a new problem in which we try to maximize the number of bit threads connecting a region to the rest of the boundary CFT.

While the bit threads serve as an excellent tool to inform our intuition, they are not particularly useful for doing calculations. In order to find some middle ground between the physically informative and the computationally useful, we can think of bit threads as microscopic (and in fact Planck-scale) objects and work in a course grained picture where we work with a vector field with certain properties referred to a *ow*. In order to maintain the physical properties of the microscopic bit threads, these fields must be divergenceless and have a bounded norm. The condition that the field is divergenceless says that every thread must begin and end on the boundary CFT and no thread can spontaneously appear or disappear in the bulk. The bound on the magnitude of the field is effectively a statement that the threads have spacial extent and thus cannot occupy more volume than is available to them. This flow can be interpreted as an information flux density, the field lines of which are followed by the bit threads. This lets us go back and forth freely between the thread and flow based picture.

The flow field is easier to work with than the threads for the means of computation, and since a given field configuration is easily converted into a thread configuration flows can also serve as a visual aid for understanding the spacial distribution of entanglement. A flow field, however,

can only compute the entropy of a single region or at most, regions nested inside of one another. In order to find the entropies of more complicated arrangements of boundary regions we need something else. Once again, graph theory provides the inspiration we need.

On graphs, there is a class of problems which are referred to as `multi-commodity flow problems' or `multi flow problems' for short. These problems are a fairly straight-forward extension of single commodity flow problems (which were the inspiration for flows in the continuous system). Essentially, a multi-commodity flow is just a number of single commodity flows on the same graph at the same time. Thus we can define a continuum mutli flow as a set of vector fields which are divergenceless and obey a bound on the sum of their magnitudes. Each field in a multi flow translates to a configuration of threads connecting a pair of regions, and together, the multi flow as a whole represents threads connecting any number of regions to one another.

The multi flow formalism has proven useful as a computational tool, and we would like to see just how useful it could be. Its derivation from its counterpart on graphs leads us to believe that many results for the graph based multi flows should be true in the continuum. In particular, results which tell us the computational limits of multi flows on graphs may apply in the continuum. If these same limits apply for the continuum multi flows, we can use graph based problems to model those in the continuum AdS/CFT. This would allow us to apply techniques and algorithms already formulated for the RT prescription to multi flows, giving us the ability to generate and prove a list of inequalities obeyed by the entropies of holographic states for any number of regions of the CFT [1]. We could then use the multi flow picture to analyze the physical meaning of these inequalities and probe the information structure of holographic states. This goal is a ways away, and first we must check to see if this is possible by finding the computational limits on multi flows. This is the goal of the work presented here.

To begin, we provide a more technical introduction to the concepts introduced here in the next section and elaborate on some of the connections between entanglement entropies and graph theory. In section 3 we go onto generalize previous results pertaining to multi flows before analytically and numerically apporaching a toy problem in section 4 which will illustrate that multi flows cannot be used to calculate the entropy of crossing regions. This will represent the point at which the analogy between our constructs in AdS/CFT will diverge from their analogs on graphs.

2 Background

2.1 AdS/CFT and Holographic Entanglement Entropy

2.1.1 AdS/CFT

The AdS/CFT Correspondence is as powerful as it is because it tells us that two systems, which on the surface appear to be very different, are equivalent. This fact allows us to take many difficult calculations from one system and perform them much more easily in its dual description. We will be performing a very complicated CFT calculation with relative ease in the dual AdS.

Recall that AdS or Anti-de Sitter space-times are maximally symmetric space-times with a constant negative curvature. This means that the length element is approaches zero at infinity, which will cause problems. In order to get around this, we impose a cutoff to the AdS which is associated with the length-scale down to which we trust the physics in our CFT. Doing this gives us a bounded manifold, which we denote M. Another important note about the AdS in `AdS/CFT' is that it need not be simple, empty Anti de-Sitter space. Different states of the boundary CFT

have different holographic duals which are *asymptotically AdS*. A holographic state may have an asymptotically AdS dual that contains matter, energy, black holes, etc.

On the other side of the equation, we have states of quantum field theories with conformal symmetries. These aforementioned `holographic states' are not just any random state of the field theory. Rather, we restrict ourselves to working with these special states that have well defined asymptotically AdS dual space-times associated with them.

Our goal will be to study the structure of these *holographic states* in order to learn what features of these states are important for holography. Examining entanglement in these states is one way we can do this. This is because holographic states all obey certain inequalities which can give us insight into their structure and hint towards how asymptotically Anti-de Sitter Spaces emerge from them. We will discuss these inequalities later in this section.

2.1.2 Entanglement and Entropy

Entanglement is a fundamental feature of quantum mechanics. This phenomenon is one of the few ways in which physicists may encode information in quantum systems without that information becoming lost to the uncertainty built into quantum mechanics.

The AdS/CFT Correspondence gives us the ability to study the structure of entanglement within quantum systems with infinite degrees of freedom. Studying entanglement will lead to a better understanding of how to manipulate information in quantum systems, as well as how entanglement may be connected to the emergence of gravitational systems.

Holographic states, like all quantum states with many degrees of freedom, can be described entirely by the operator which is often known as the density matrix operator. In order to examine the entanglement which is internal to this state, we must first divide it into parts. The simplest way of doing this is to consider a region A of the conformal field theory and ask how the state on only that region is entangled with the rest of the field theory A^c. The state on A is described by the reduced density matrix, given by:

$$_A = Tr_{A^c}[\ ] \tag{1}$$

where Tr_{A^c} represents the partial trace over the degrees of freedom in A^c.

The entanglement across a partition of the state can be quantified by an entropy, which can be used to tell us (practically speaking) how much information is stored in the entanglement between the bipartite regions. This *entanglement entropy* is given by the von Neumann entropy, which for a region A is

$$S(A) := \quad Tr[\ _A ln(\ _A)]: \tag{2}$$

From the definition above, one can define a number of useful quantities by taking the sums and differences of entropies of different regions in a multipartite system. These more directly reflect the amount of information which is, or is not shared between regions of the boundary CFT.

As of yet, there has been no mention of the space-time of one higher dimension beyond the conformal field theory. The most relevant connection between from a CFT to it's holographic dual is given by the RT formula, which gives a method for finding the von Neumann entropy via geometric calculation in an asymptotically Anti-de Sitter space. The RT formula equates the von Veumann entropy of a region A with the area of the minimal surface in the bulk that is homologous to A. In mathematics, this stated as:

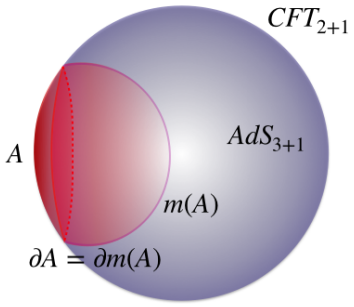


Figure 1: A visual illustration of the RT formula for CFT_{2+1}/AdS_{3+1}. The entropy of a region A of the CFT is equivalent to the area of the minimal surface $m(A)$ in the bulk AdS such that the boundary of the surface is the same as that of the region A

$$S(A) = \frac{1}{4G_N}\text{Area}(m(A)) \tag{3}$$

where $m(A)$ denotes this minimal surface.

There is an alternative version of the RT formula which is described in terms of a vector field in the bulk. Such a vector field $\vec{v}$ is referred to as a *maximal flow* and obeys the following constraints:

$$\nabla \cdot \vec{v} = 0 \tag{4}$$

$$|\vec{v}| \leq 1$$

for the reasons outlined in the introduction. We refer to a flow as maximal (for a region A) when:

$$\int_A \vec{v} \cdot \hat{n}\sqrt{h}\, = S(A) \tag{5}$$

where h is the determinant of the induced metric on the boundary A and $\hat{n}$ is the unit vector normal to A. The reason that these are called maximal flows will be described later in this section.

One subtlety of this formalism is that the bit threads are undirected while the flow field is. By convention we will always have a flow going out of the region it is associated with and into its compliment.

Single flows present an interesting conceptual alternative to the RT formula although they do not tell us anything more than the RT formula when it comes to the entanglement structure of holographic states. In order to learn more about this particular topic, we need to know how threads must be distributed across multiple regions. The flow formalism is insufficient to do this calculation, so we can give it an upgrade and define what we call a *multi flow* $\{\vec{v}_i\}$, which is simply a collection of individual flow fields $\vec{v}_i$ which together obey a collective norm bound

$$\sum_i |\vec{v}_i| \leq 1 \tag{6}$$

If a multi flow is maximal for some set of boundary regions A_i, then we define the flow going from A_i to A_j as $\vec{v}_{ij}$, then the multi flow is given by $\{\vec{v}_{ij}\}$. Notice that some additional constraints follow immediately from this setup:

$$\vec{v}_{ij} = -\vec{v}_{ji} \tag{7}$$

$$\forall_{ij} \quad \hat{n}_k = 0 \qquad 8k \in i; j$$

where $\hat{n}_k$ is the normal vector to the region A_k. Mutli ows have the potential to let us map out the entanglement structure of holographic states, which was not previously possible with the RT formula or the single ow formalism. For this reason, the multi ow structure is of great interest and so it is the subject of the work performed herein.

2.1.3 Entropy Inequalities and the Holographic Entropy Cone

The RT formula in its various forms allows us to probe the entanglement structure of holographic states (since it is necessarily holographic calculation). By examining the entropy of multiple regions of a CFT in a general holographic state, we can determine some features of the entanglement structure that make it holographic. While a few facts are immediately obvious from the de nition of the entanglement entropy in terms of CFT calculations, many more conditions on holographic states emerge once the RT formula comes into play. Two prominent entropy inequalities are

$$S(AB) + S(BC) \quad S(B) + S(ABC) \tag{8}$$
$$S(AB) + S(BC) + S(AC) \quad S(A) + S(B) + S(C) + S(ABC)$$

Where $AB; BC; ABC$, etc. are the unions of the respective regions $A; B; C$, etc. These inequalities are known as *strong subadditivity* (SSA) and *the monogamy of mutual information* (MMI) respectively.

If we consider n regions of the CFT, we can construct a vector in $R^{2^n \ 1}$, the elements of which are the entropies for each region and all possible combinations of them. As discussed in [1] in this space, referred to as the *holographic entropy cone for n regions*. The extremal rays de ning the edges of the holographic entropy cone are those entropy vectors for which some entropy inequality is saturated. It turns out that for $n \ 4$ SSA, MMI, and other inequalities which can derived from the two taken together, completely de ne the holographic entropy cone. However, for $n = 5$ there are more inequalities which constrain holographic states. Using the minimal surface prescription, a method has been constructed for nding and proving such higher-party entropy inequalities [1]. For this to be accomplished, it is necessary to construct a *graph model of holographic entanglement entropies*. An example of a more complicated entropy inequalities is

$$S(ABC) + S(ABD) + S(ABE) + S(ACD) + S(ACE) + S(BC) + S(DE)$$
$$S(AB) + S(ABCD) + S(ABCE) + S(AC) + S(ADE) + S(B) + S(C) + S(D) + S(E) \tag{9}$$

which is one of the entropy inequalities for ve regions which is independent of SSA and MMI.

The typical procedure for proving entropy inequalities using bit threads, is to nd a maximal ow or multi ow for all regions that appear on the right hand side, and then show that the inequality follows as a consequence of the thread con guration. We will see that this is not a trivial task for inequalities of this form because the right hand side contains *crossing regions* (for example AC and ADE). If we want to use the mutli ow formalism to prove and analyze these inequalities, this must be addressed.

2.2 The Max-Flow Min-Cut Theorems

The ow and multi ow formalism is constructed formally by moving a theorem for graph based multi ows into the continuum. This theorem is the well known *Max-Flow Min-Cut Theorem*

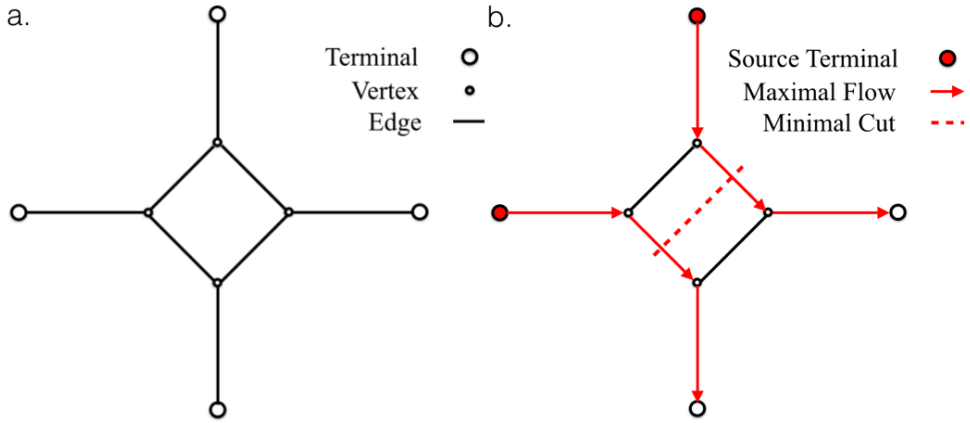


Figure 2: a.) An example graph on which we may de ne a commodity ow problem. This graph is made up of vertices connected by edges, some subset of which are declared to be the `terminals' of the graph which will act as sources or sinks for the commodity ow. b.) Example of a commodity ow program where the commodity is sent from source terminals through the graph to the remaining terminals. The Max-Flow Min-Cut Theorem states that the maximum ow that can be sent from some source terminals is equal to the minimum number of edges which are cut by a partitioning of the vertices such that the source terminals are separated from the rest in the partition.

(MFMC). A brief visual explanation of the graph based MFMC Theorem is presented in Figure 2. The graph presented in the gure has all edges with a capacity of 1, however in general one can weight edges di erently such that more or less ow can pass through one edge than others. We call these weights the *capacity* of an edge. The only thing this changes on the cut side is that the minimal cut is now computed by the sum of the capacities of the edges cut by the partition. It should be noted that any commodity ow must be conserved at every vertex and no commodity ow can exceed the capacity of the edge it passes through.

Flows are a continuum analog of the single commodity ow on graphs, and likewise multi ows are a continuum analog of multi-commodity ows on graphs. Instead of conserving ow at vertices, we now demand that all ow elds are divergenceless in the bulk. Similarly, instead of forcing commodity ows to respect the capacity of the edges they traverse, we now demand that the magnitude of ow elds do not exceed some bound.

The equivalence between maximizing the ux of a vector eld across a source region of a continuous boundary and minimizing the area of a surface homologous to the source region is referred to as the continuum MFMC Theorem and is proven using the strong duality between convex programs.A convex program is the statement of an optimization problem of a certain form. The components of a convex program are:

1. fxg: Some variables de ning a convex set over which the optimization will occur
2. $f_0(fxg)$: A convex or concave function which is referred to as the objective of the program. This function will be minimized (or maximized in the case of a concave function) subject to some constraints.

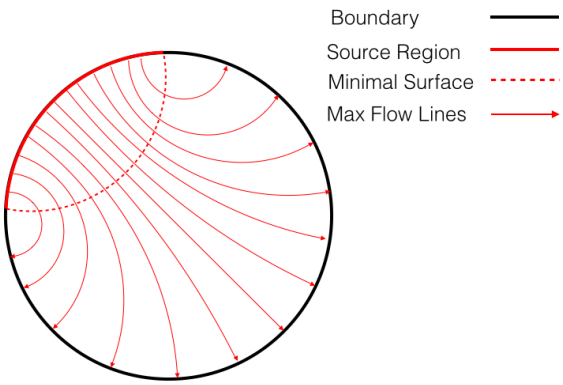


Figure 3: An illustration of the continuum Max-Flow Min-Cut Theorem. In place of source terminals we now have a source region (or set of source regions). The commodity ow is replaced by a norm-bounded, divergenceless vector eld and the minimal cut is replaced by the minimal surface homologous to the source region

3. $f_i(fxg)$: A set of convex inequality constraints of the form $f_i(fxg)$ 0.

4. $h_i(fxg)$: A set of a ne equality constraints of the form $h_i(fxg) = 0$.
   Taken with the inequality constraints, these de ne a set of feasible points within the larger space spanned by the variables of the program. The solution to the convex program will be at some point within the set of feasible points.

One important feature of convex programs is that if the set of feasible points is non-empty, the program can be *dualized* by removing the constraints on the program and adding back those constraints, multiplied by Lagrange multipliers, to the program's objective. Once the unconstrained optimization is performed over the original variables, we are left with the objective of a new *dual* program. This is to be optimized with respect to the Lagrange multipliers associated with the original constraints. This new dual program will be a minimization problem if the original, or *primal*, program was a maximization problem. The crucial fact used in proving the equivalence of the ow prescription to the RT formula is that the optimal objective value of these two problems will be the same.

Based on purely geometric arguments, it becomes clear that with ows such that $j\nu j$ 1,

$$\int_A v \quad S(A) \tag{10}$$

where $\int_A v \quad S(A)$ is used as a short-hand for $\int_A p\overline{hn}\ v$. Now we can write our ow problem as a concave program:

$$\text{Primal: Maximize } \int_A v \text{ with respect to } \nu, \text{ subject to } \tau \quad \nu = 0 \text{ and } j\nu j \quad 1. \tag{11}$$

By dualizing this program, we obtain a minimization problem which has a solution bounded below by the entropy of A. Thus the entropy will be the extremal value of both the primal objective and the dual objective:

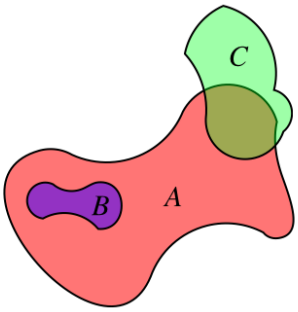


Figure 4: For the three sets $A; B; C$ depicted. A and B do not cross because B is entirely nested within A. B and C do not cross since their intersection is empty. A and C do cross since their intersection is non-empty and neither is a subset of the other.

$$\text{Max} \int_A v = S(A) \tag{12}$$

Furthermore, it will turn out that the con guration of variables which extremizes the dual program will return a delta function along the minimal surface homologous to the source region [2].

By extension, the continuum MFMC theorem can be applied to multi ows by a similar procedure. Only in this case, we will include the ux of multiple ows through multiple regions in our primal objective.

2.3 Further Connections to Graphs

Graph theory provides the framework for developing the multi ow formalism, but it also contributes in other ways to the study of entanglement entropy in AdS/CFT. One interesting and relevant example of this is the development of a graph model for holographic entanglement entropies which contains all of the relevant information about the minimal surfaces of source regions and their unions. One reason for this development is so it can be used for an algorithm to nd and prove entropy inequalities for multipartite systems, beyond SSA and MMI [1]. Due to their strong connection to graph based problem, it is reasonable to ask if ows and mutli ows can be modeled in a similar fashion, and if such machinery can be used in an analogous way to study entropy inequalities.

The continuum and discrete MFMC theorems lead us to believe that the graphs which model cuts and ows should be identical. Now we can turn to known results from graph theory to test how well the multi ow formalism holds up compared to its discrete counterpart.

The particular graph theoretic result is known as the *Locking Theorem*. This tells us if a maximal multi-commodity ow exists given a particular choice of subsets of terminals which we want to maximize the ows from. In particular, the Locking Theorem puts a limit on how sets of source terminals may *cross*. The condition for two regions A and B to be crossing is that none of the following sets are empty:

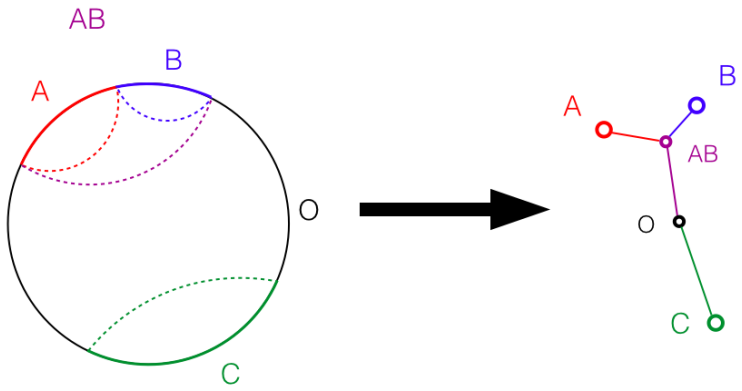


Figure 5: Example of converting a set of source regions with their minimal surfaces into a graph model. The boundary along with the minimal surfaces of source regions divide the bulk into disjoint volumes, each of which is assigned a vertex in the graph model. Each bulk cell is connected to adjacent ones in the graph models via edges which are weighted by the area of their shared boundary (in the example here this is just the area of the minimal surface dividing them).

$$A \setminus B; \quad A=B; \quad B=A; \quad (A \cup B)^c: \tag{13}$$

For simplicity we will usually omit the last condition, as it is meant to exclude cases which do not concern us. The Locking Theorem is stated (in simpli ed language) as follows;

> *The Locking Theorem: For any family of 3-cross free subsets of terminals, there exists a multi-commodity ow con guration which simultaneously maximizes the ow from each subset.*

For a more in depth treatment of the Locking Theorem, see [3]. In order for the idea of modeling continuum multi ows as discrete multi-commodity ows to work, our continuum construct must also satisfy the Locking Theorem. We will soon see, however, that this is not the case. Before we examine the failure of continuum multi ows to satisfy the Locking Theorem, we should rst check that it does not fail in cases which are regarded as trivial by the discrete Locking Theorem, i.e. in the case of non-crossing source regions.

3 Multiflows for Non-Crossing Regions: The Weak Continuum Locking Theorem

As of now, it has been proven that there exists a multi ow which is maximal for:

1. Any number of disjoint source regions
2. Any number of disjoint source regions and one additional *composite* region made up of some union of the other disjoint *elementary* regions [4].

The Locking Theorem states that this should still be true for any arbitrary choice of non-crossing regions or for regions which have crossings but are still 3-cross free. Before we consider whether or not multi ows can reproduce entropies for crossing regions, we must rst prove to make sure that this can be done for an arbitrary number of disjoint and nested regions.

In this section, we present two di erent proofs which show that this is the case. We refer to this result as *the Weak Continuum Locking Theorem* and it can be thought of as a generalization of the Theorems proven in [4] pertaining to multi ows.

3.1 Proof by Dualization

The rst proof is accomplished by dualizing the convex program of maximizing a multi ow over arbitrary non-crossing regions. By utilizing equation (10), it is su cient to show that the objective of the dual program is bounded below by the sum of the entropies of the source regions. We begin by partitioning the boundary into N disjoint regions and labeling these regions A_i. Now de ne some number of regions $fB\ g$ such that $B \quad fA_1; A_2; :::; A_N g$ subject to the constraint that at least one of the following conditions holds;

$$B \ \backslash B = ;; \quad B \quad B ; \quad B \quad B \quad 8 ; \tag{14}$$

which is the statement that these regions are non-crossing. We will refer to these regions as *composite regions*. We now construct a convex program which will maximize the ux of some multi ow over each composite region.

The statement of our program is

$$\text{Primal Program: Maximize} \sum_{A_i 2 B} \sum_{A_j \not\in B} \sum \int_{A_i} v_{ij}; \tag{15}$$

with respect to $f\nu_{ij}g$, subject to:

$$\nu_{ij} + \nu_{ji} = 0; \tag{16}$$

$$\sum \nu_{ij} = 0; \tag{17}$$

$$\sum_{i<j} j\nu_{ij}j \quad 1: \tag{18}$$

If we de ne n_{ij} to be the number of sets B which contain A_i and not A_j ($n_{ij} := \#(\ jA_i \ 2 \ B \ ; A_j \not\in B \ )$) and make use of constraints (5) and (6), we can rewrite the object of the primal program as:

$$\sum_{i<j} \left(\int_{A_i} n_{ij} v_{ij} \quad \int_{A_j} n_{ji} v_{ij} \right) : \tag{19}$$

We can write a Lagrangian for our program by not explicitly enforcing constraints (17) and (18) and instead, add them to our objective multiplied by Lagrange multipliers ($_{ij}$ and). The Lagrangian is

$$\sum_{i<j} \left(\int_{A_i} n_{ij} v_{ij} \quad \int_{A_j} n_{ji} v_{ij} \right) + \int_M \left[\sum_{i<j} {}_{ij} \sum \nu_{ij} \quad j\nu_{ij}j \right] : \tag{20}$$

If we perform integration by parts, and then demand that the maximum of the Lagrangian with respect to the multi ow be nite, we obtain our dual program:

$$\text{Dual Program: Minimize} \int_M ; \tag{21}$$

with respect to $; f_{ij}g$, subject to:

$$j\tau_{ij}j; \tag{22}$$

$$_{ij}j_{A_i} = n_{ij} \quad _{ij}j_{A_j} = n_{ji}: \tag{23}$$

Notice that (22) and (23) taken together imply that for any path c between A_i and A_j

$$\int_c ds \quad N_{ij} \tag{24}$$

where $N_{ij} := n_{ij} + n_{ji}$. We then de ne $N_i := \#(jA_i 2 B \quad B)$, which is the number of B which contain A_i and are also contained in B (and note that B is not counted in N_i). In general, we know that

$$N_i + N_j \quad N_{ij}: \tag{25}$$

By this de nition, it follows that

$$\begin{cases} N_i \quad n_{ij} \quad 1 & A_i 2 B ; A_j \not\in B \\ N_i = 0 & A_i \not\in B \end{cases} \tag{26}$$

where the rst inequality relies on the fact that B is not counted in N_i .

We now de ne functions $_i(x)$:

$$_i(x) := \inf_{c \text{ from } A_i \text{ to } x} \int_c ds: \tag{27}$$

This implies that

$$_i(x) + \ _j(x) \quad N_{ij}: \tag{28}$$

Now we de ne

$$(x) := \min_{A_i 2 B} (\ _i(x) \quad N_i); \tag{29}$$

and we denote the level sets of these functions as

$$R \ (p) := \ x 2 M \ j \quad (x) = p \tag{30}$$

for some $x 2 A_i 2 B$, $_i(x) = 0$ and $N_i \quad 0$. Thus $(x) \quad 0$ by (29). Now consider a di erent $x 2 A_j \not\in B$, if we let $A_i 2 B$ be the minimizer of (29), we know that $_j(x) = 0$ and thus by (28) and (26),

$$(x) = \ _i(x) \quad N_i \quad N_{ij} \quad N_i \quad 1 + n_{ji} \quad 1: \tag{31}$$

Thus, the level sets of between 0 and 1 must be homologous to B . We then de ne regions

$$R := \{x \in M \mid 0 < \Phi(x) < 1\} \tag{32}$$

If all composite regions are non-crossing, then we should expect that

$$R \cap R = \emptyset \tag{33}$$

if $\Phi \neq \Phi$. In order to show this, we must consider two cases: one in which $B \cap B = \emptyset$, and the other where $B \neq B$.

1. In the first case, let $B \cap B = \emptyset$. We will also let $A_i \in B$ such that $\Phi(x) = \phi_i(x) - N_i$ and $A_j \in B$ such that $\Phi(x) = \phi_j(x) - N_j$ for some x. If we choose x to be some point in R we know that by definition
$$\Phi(x) < 1. \tag{34}$$
For this case, we also know that
$$N_i + N_j \geq N_{ij} \geq 2 \tag{35}$$
by equation (26). We now note that this, along with (28) and (29), gives
$$\Phi(x) + \Phi(x) = \phi_i(x) + \phi_j(x) - (N_i + N_j) \geq 2 \tag{36}$$
and that combining this with (34) implies that
$$\Phi(x) > 1 \Rightarrow R \cap R = \emptyset \tag{37}$$
in the case of disjoint composite regions.

2. For the second case, suppose instead that $B \neq B$ but still that $x \in R$ and that $A_i \in B$ such that $\Phi(x) = \phi_i(x) - N_i$. Then the inequality (34) still holds but instead of (35), we now have
$$N_i \geq N_i - 1. \tag{38}$$
From the definition of N_i and the premise of the case. This along with (34) implies that
$$+ N_i \geq N_i = \phi_i(x) - N_i \quad \Phi(x) < 0 \Rightarrow R \cap R = \emptyset. \tag{39}$$
Thus the claim holds for both cases.

The fact that the R do not intersect allows us to place the following bound on the dual objective:
$$\int_M \rho \geq \sum_R \int_R \rho. \tag{40}$$
Now using Hamilton-Jacobi theory (taking $\int_c dS$ to be the action), we find that $= |\tau - \Phi(x)|$ thus we can rewrite this as:
$$\int_M \rho \geq \sum_R \int_R |\tau - \Phi(x)| \tag{41}$$

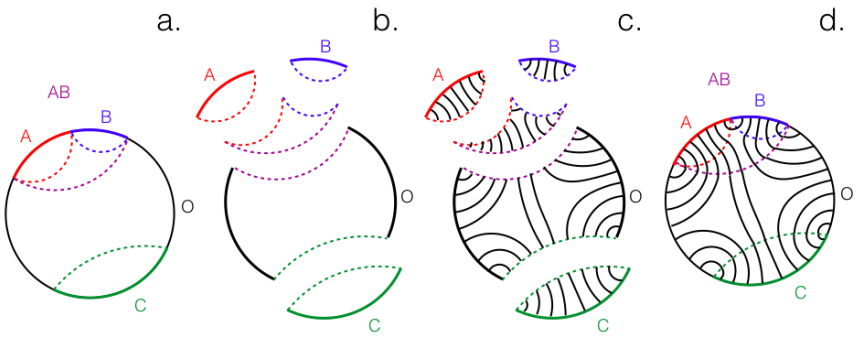


Figure 6: Illustration of the proof by decomposition. a) depicts step 1; the choice of source boundary regions and the identi cation of their minimal surfaces. b) depicts step 2; the dividing of the bulk into cells. c) depicts step 3; nding a maximal thread con guration for the boundaries of each bulk cell, and lastly d) depicts step 4; in which the bulk cells are recombined to obtain a maximal thread con guration for the original problem identi ed in step 1.

Since the integrals on the RHS correspond to the average area of the level sets of homologous to B they must be at least as great as the area of the minimal surface homologous to B . So:

$$\int_M \sum_R |\tau(x)| \geq \sum_{B} S(B) \tag{42}$$

by the RT formula. However by strong duality, the minimum of the LHS, subject to the correct constraints, must be equal to the optimum of the primal program. Thus the ux of the multi ow from all B is both bounded above and below by the sum of their entropies, and thus we can conclude that there exists a multi ow which is maximal for all B so long as there is no crossing between them.

3.2 Proof by Decomposition

In addition to the proof above, which relies on computing the dual program and constraining it's solution, we provide a second proof. Rather than relying on the formal theory of convex programs, this proof will be a proof by construction. We will present a method for constructing a multi ow which saturates on the minimal surfaces of all chosen boundary regions, and thus can be used to simultaneously calculate their entropies. To avoid confusion regarding orientation, we will proceed by talking mainly in the language of bit threads, knowing that once we have nished constructing a thread con guration it can be converted into a proper multi ow.

The procedure for constructing a maximal multi ow for any set of non-crossing boundary regions is:

1. Choose a set of non-crossing regions of the boundary to calculate the entropies of. Recall that this means that for any two chosen regions, either they are disjoint, or one is nested inside

of the other. In the bulk, determine the minimal surfaces homologous to each boundary region and denote them $m(B_\alpha)$. It is important to note that since the boundary regions are non-crossing, the regions given by the interior of $B_\alpha \cup m(B_\alpha)$ also do not cross for any two choices of α.

2. Now note that the boundary regions and minimal surfaces divide the entirety of the bulk into a finite number of "cells." In the case where the boundary regions are non-crossing, there are a few different cases for these cells.

   (a) The cell is the interior of $B_\alpha \cup m(B_\alpha)$ for some choice of α.

   (b) The cell is bounded by the minimal surface of some boundary region which has other source regions nested within it. In this case, the form of the cell is not that of (a), but instead, the cell is now given by

   $$\text{Int}\, B_\alpha \cup m(B_\alpha) \bigcup_{B_\beta \subset B_\alpha} \text{Int}\, B_\beta \cup m(B_\beta) : \tag{43}$$

   In general, these cells will be bounded by the minimal surface of $m(B_\alpha)$, some subset of B_α itself, and the minimal surfaces of the regions nested within B_α.

   (c) The last type of bulk cell which we will encounter for non-crossing regions is bounded by none of the source boundary regions, but rather bounded by the interior of the union of the purifying region and its minimal surface. In other words, this last cell type is the compliment of the union of all other bulk cells.

3. Constructing a maximal thread configuration for each individual bulk cell is straight forward if we make use of Theorem 1 from [4]. This theorem tells us that for any number of disjoint regions, there exists a multiflow (or thread configuration) that is simultaneously maximized on all of them. Since the proof of this theorem is completely independent of the bulk/boundary geometry, we are free to apply it to individual bulk cells. Based on the cells described in step 2, we can choose our `sub-source regions' for each bulk cell to be either a minimal surface of some source region or some subset of our source region. This gives us a decomposition of the boundary of each bulk cell into disjoint regions.

   Now for each region bounding each bulk cell we make use of the theorem from [4] to construct/state the existence of a thread configuration which is maximal for all boundaries of each cell.

4. Now that we have a maximal thread configuration for each bulk cell, we must now recombine them and show that this thread configuration is maximal for each source region. We do this by taking note of the following facts:

   (a) Since any minimal surface is the minimizer of its homology class, we know that the maximum number of threads which can be placed on them is equal to their area. This means that any thread configuration which maximizes on a minimal surface will have the threads maximally packed everywhere on it.

   (b) By virtue of how we have divided the bulk, every minimal surface forms part of the boundary of exactly two bulk cells.

(c) Because of (a) and (b) when two adjacent bulk cells are recombined, the density (and direction) of threads on their shared boundary will be equal and maximal. This allows us to put the bulk cells back together and think of the thread con gurations on the individual bulk cells as being continuous over their shared boundaries.

5. Thus stitching together to maximal thread con gurations on each individual bulk cell gives us a well de ned thread con guration for the entire bulk. The fact that there is no crossing in our chosen source regions ensures that their minimal surfaces will not cross either. Thus once a thread leaves a one bulk cell it will not return to it from another one when the bulk is stitched back together. This means that no threads will loop back to the boundary region they leave from. If this were not the case the proof would fail since threads which return to the boundary they leave from correspond to eld lines with net zero ux. The fact that no thread returns to a region it leaves is clear from the fact that the graph model describing this setup is a tree graph if no crossing is present. Since this is the case and thread con guration is maximized on all of the minimal surfaces of the source regions (by construction), and the entire thread con guration must be maximal for all source regions.

6. Now all that is left to do is to convert this thread con guration into a proper multi ow by choosing an orientation for the threads. Once this is done, the construction is complete and the proof is concluded.

4 Multiflows for Crossing Regions

Since mutli ows are capable of computing the entropy of arbitrary non-crossing regions, we can now ask if they can do the same for crossing regions. In order to test this, we will consider a relatively simple problem.

The machinery of convex optimization and multi ows make no reference to any particular geometry, and so we are free to work in a simpler geometry than AdS. With that in mind, we consider the unit square on $\mathbb{R}^2$ given by $(0;1)^2$. We label the sides of this square $A; B; C$ and D, and de ne a mutli ow on this square; $f\vec{v}_{ij}g$ where $i; j \in fA; B; C; Dg$. Note that the entropy of any single region on its own is given by;

$$S(A) = S(B) = S(C) = S(D) = 1 \tag{44}$$

since the minimal surface of each is simply the region itself. Furthermore, the entropy of the union two adjacent sides are equal to

$$S(AB) = S(AD) = S(BC) = S(CD) = \frac{p}{2}; \tag{45}$$

as displayed in Figure 7. In order to test multi ows in the case of crossing source regions in this toy problem, we attempt to maximize the ux of a multi ow on this square over the crossing source regions AB and BC. If multi ows are capable of simultaneously computing the entropy of two crossing regions, we should expect that the maximal ux from these source regions should be $S(AB) + S(BC) = 2\frac{p}{2}$. Approaching this problem can be done in two ways; attacking it head on, or dualizing the problem and trying to nd a solution to that. We will do both, starting with analytic ansatzes and then attempting to numerically compute the solutions for both the primal and the dual on a lattice.

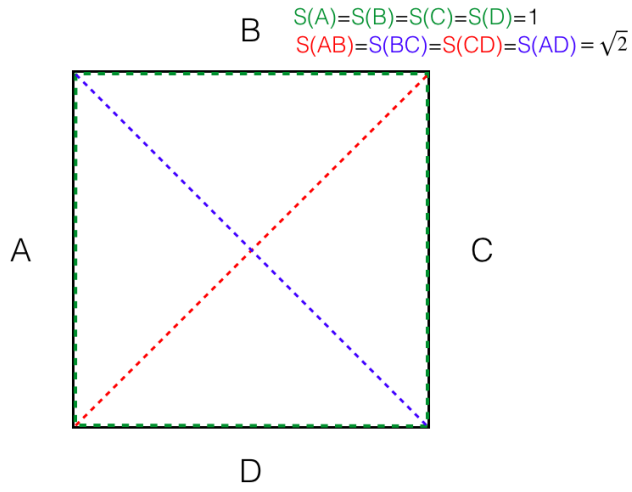


Figure 7: An illustration of the minimal surfaces and corresponding entropies of the toy problem described.

4.1 Analytic Attempt: Primal

Before beginning, note that the convex program in question is this:

Primal: Maximize $\int_{AB} v_{(AB)(CD)} + \int_{BC} v_{(BC)(AD)}$ with respect to $\{\nu_{ij}\}$ subject to $\nu \geq \nu_{ij}\ \forall i,j$ and $\sum_{i<j} j\nu_{ij}j \leq 1$.

We may rewrite the objective of this program as

$$\begin{aligned} &\int_{AB} (v_{AC} + v_{AD} + v_{BC} + v_{BD}) \int_{BC} (v_{BA} + v_{BD} + v_{CD} + v_{CA}) \\ &= \int_A (v_{AB} + 2v_{AC} + v_{AD}) + \int_B (v_{BC} + 2v_{BD}) + \int_C v_{CD}. \end{aligned} \tag{46}$$

There are two important things to consider for formulating an ansatz for this problem. First, note that the square itself exhibits the full range of symmetries of the Dihedral-4 group, thus there should exist a maximal solution which exhibits the sames symmetries. We may restrict ourselves to ansatzes which do so, thereby narrowing the range of possible solutions. Secondly, note that the flows v_{AC} and v_{BD} contribute twice as much as the other flows in the objective function.

This second fact is misleading in some ways, as one may assume that the maximal solution will be one with both of these flows dominating the bulk. However, note that only one of them can be saturated on the entire square at once. There are three things one may try to get around this;

1. Make all flows zero except for one of these two (say ν_{AC}) which has magnitude one on the entire square. This gives a total flux of $2 < 2\sqrt{2}$.

2. Compromise and make both flows ν_{AC} and ν_{BD} non-zero with magnitude $\frac{1}{2}$ everywhere. This too gives a total flux of $2 < 2\sqrt{2}$

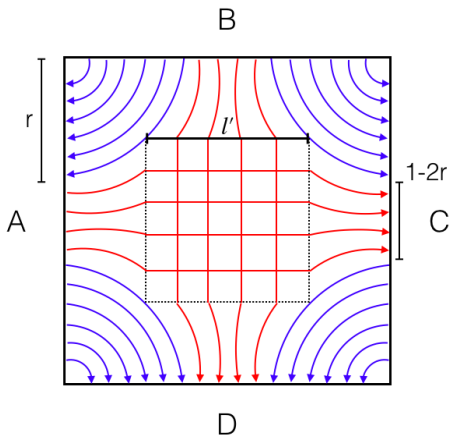


Figure 8: An illustration of our maximal ansatz. Flows between opposite sides of the square are shown in red and ows between adjacent sides in blue. Note that the density of the ow lines going across the square is maximal at the boundary but less in the interior of the square. In order to give l' the necessary value of $l' = 2(1 \quad 2r)$ we choose $r = \frac{1}{4-\sqrt{2}}$. This gives a total ux of $4\left(\frac{3-\sqrt{2}}{4-\sqrt{2}}\right)$.

Evidently, these double-counted ows alone will not give a maximal solution (unless the maximum ux is 2). Next we might guess that the ows between adjacent sides, which are only counted once in the objective, form quarter circles of radius r about the corners of the square. A quick check con rms that if these are the only ows in the system the maximal value of the ux is once again 2.

For a more sophisticated guess, we maintain that the ows between adjacent sides will ow in quarter circles in which the norm bound is saturated, but now we have nonzero ows from opposite sides as well. The symmetry suggests that these cross ows must be equal at the center of the square and thus their magnitude there cannot exceed $\frac{1}{2}$. In order to maximize the ow between opposite sides, we can imagine giving the ows a norm of 1 on the available space on the boundary and then having the threads spread out as space is made available for them. This will continue up to the point where they are no longer restricted by the ows between adjacent sides. From there, we assume for simplicity that the threads will ow uniformly across the square until obstructed again. In order to accomplish this, we can imagine placing a smaller square concentrically inside the unit square such that within it the ows between opposite sides are uniform and of magnitude $\frac{1}{2}$. The size of this square is limited by the radius of the quarter circles formed by the ows between adjacent sides. Such a con guration is depicted in Figure 8.

A very brief geometrical calculation tells us that, in order for the ows between opposite sides to have magnitude 1 on the boundary and $\frac{1}{2}$ in the center, we must choose a radius $r = \frac{1}{4-p\overline{2}}$ for the quarter circles formed by the ows between adjacent sides. This is such that $l^0 = 2(1 \quad 2r)$ where l^0 is the side length of square enclosing the uniform cross- ow.

This con guration gives a total ux of

$$\int_{AB} v_{(AB)(CD)} + \int_{BC} v_{(BC)(AD)} = 4(\frac{1}{4}\overline{p_2}) + 4\ 1\quad \frac{2}{4}\overline{p_2} \qquad 2{:}453 \tag{47}$$

Which is larger than 2 but still less than the expected value of $2\overline{p_2}$.

We can imagine improving this ansatz by making the ways the ows behave a little less rigid and discontinuous, but to do this analyticly is not feasible. Instead, we can dualize this problem in order to see if we can really achieve our desired goal of calculating the entropies of the two crossing regions simultaneously.

4.2 Analytic Attempt: Dual

We dualize our program in the usual way of removing the constraints and adding them into the objective multiplied by Lagrange multipliers and then maximizing with respect to the usual variables.

$$L = \sum_{i=1}^{} \sum_{j>i}^{} \int_{A_i} v_{ij} + \int_{M} \sum_{i=1}^{} \sum_{j>i}^{} {}_{ij}(\tau \quad \forall_{ij}) + (1 \sum_{i=1}^{} \sum_{j>i}^{} j\forall_{ij}j) \tag{48}$$

Where $A_1 = A;\ A_2 = B$, etc. By performing integration by parts on the second term in the Lagrangian and demanding that the resulting boundary terms vanish (to ensure that our maximum ux is nite) we gain the conditions:

$$\begin{gathered} {}_{AB}j_A = \ {}_{AC}j_A = \ {}_{AD}j_A = \ {}_{BC}j_B = \ {}_{BD}j_B = \ {}_{CD}j_C = \ 1 \\ {}_{AB}j_B = \ {}_{AD}j_D = \ {}_{BC}j_C = \ )_{CD}j_D = 0 \\ {}_{AC}j_C = \ {}_{BD}j_D = 1 \end{gathered} \tag{49}$$

And we are left with a Lagrangian

$$L = \int_{M} \sum_{i=1}^{} \sum_{i<j}^{} (\forall_{ij} \quad \tau_{ij} + j\forall_{ij}j) \tag{50}$$

This is maximized with respect to the ows when $\forall_{ij}$ is anti-parallel to τ_{ij}, at which point if we want the maximum of our program to be nite we better require that

$$j\tau_{ij}j \qquad 8i; j: \tag{51}$$

At this point the maximum value of the Lagrangian occurs when all ows are equal to zero and we are left with the objective of our dual program;

$$\int_{M} \tag{52}$$

which we will minimize with respect to and $_{ij}$ subject to the constraints given by equations (51) and (49).

Note that these constraints together imply that for the path integral of along some path c;

$$\int_{c} ds \quad \begin{cases} 1 & c \text{ connects adjacent sides of square} \\ 2 & c \text{ connects opposite sides of square:} \end{cases} \tag{53}$$

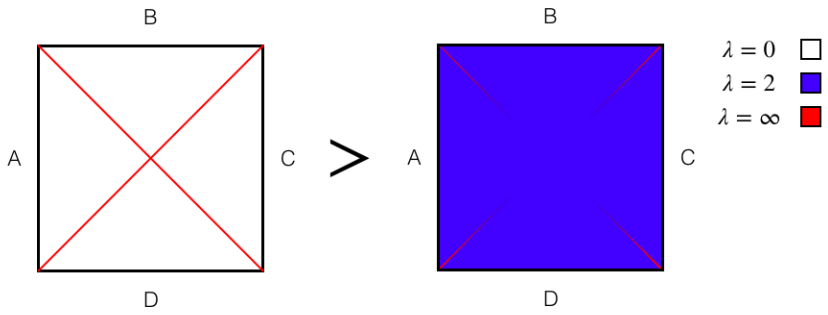


Figure 9: Illustration of ansatzes for the dual program. On the left is the con guration we expect based on the Locking Theorem. Here is zero ever where except for on the minimal surfaces for AB and BC, where it is in nite. On the right is an example of a feasible which returns a smaller value of the objective. Most everywhere = 2 except for delta functions coming from the corners which have a weight which decreases linearly with distance from the corner.

This lets us rephrase the dual program in as

$$\text{Minimize} \quad \int_M \quad \text{with respect to} \quad \text{subject to equation (53).}$$

Recall that if the primal program is to return the sum of the entropies of the regions sourcing the multi ow, then the dual program should return the same. The that is minimizes the dual program should be a delta function located on those surfaces and zero everywhere else in the bulk. For this problem then, we hope that consists of delta functions connecting opposite corners of the square, which would give $2\sqrt{2}$ when integrated over the entire square. If there is a choice of which satis es the constraints given and has an area integral less than $2\sqrt{2}$ then the objective of our primal program will be bounded above by this value. Thus the multi ows will fail to compute the entropies of the crossing regions.

If we try to think of such a we may be tempted to start by making take a constant value of 2 everywhere. This would automatically satisfy the constraint for paths connecting opposite sides of the square, however it does not satisfy the constraint on all paths between adjacent sides. Simply stated this is because the minimum length of a path connecting opposite sides is 1 while the minimum length of paths connecting adjacent sides is in nitesimal. Thus we know that any feasible must go to in nity at the corners of the square.

As an extension of this thought, we can place delta functions along the lines connecting opposite corners of the square which are weighted such that they ensure that paths between adjacent sides are ensured to be greater than 1 when integrated against . In doing this calculation we nd that the appropriate weight to add to these delta functions is

$$1 \quad 2\sqrt{2}\,; \tag{54}$$

where is the distance from the nearest corner of the square. Note that at $= \frac{1}{2\sqrt{2}}$ the delta function ends. Integrating this weight from 0 $\frac{1}{2\sqrt{2}}$ tells us that each one of these four

weighted delta functions contributes $\frac{p\overline{2}}{8}$ to the objective. Thus for this ansatz we have

$$\int_M = 2 + 4\frac{p\overline{2}}{8} = 2 + \frac{p\overline{2}}{2} \quad 2{:}707 < 2^{p\overline{2}}{:} \tag{55}$$

Since the value of the objective function of the dual program for any feasible point gives us a strict upper bound of the maximum of the primal objective function, this lets us conclude that the multi ow formalism fails to compute the entropies when presented with crossing regions (or at the very least will not always be successful). We can ask what the actual solution to the problem is if not the entropies of the source regions. Since our analytic ansatzes are likely not the true solutions, our best bet is to turn to numerical optimization to nd an answer.

4.3 Numerical Check: Dual

4.3.1 Implementation

For all numerical work we have used the optimization tools built into *Wolfram Mathematica (11)*. To numerically compute the optimum of the dual and primal programs, we rst discretize the unit square to an $L \quad L$ lattice. What variables we de ne on these lattice sites will change between the computation of the primal and dual programs. For both programs we will nd the solution to the optimization problem for increasing values of L (doing so e ectively increases the `resolution' of our computation).

We will start with our treatment of the dual program. In performing this exercise, we hope to learn how far the optimum of our primal program is from the expected value of $2^{p\overline{2}}$ as well as get a sense of the form of which gives the minimum of the dual program to see what we can learn from it. For the computation, we will implement the dual program in the form of equation (52) subject to the constraints (49) and (51). On each lattice site we de ne the variables $_{XY}(i;j)$, where $X;Y \ 2 \ fA;B;C;Dg$ and $i;j \ 2 \ f1;2;:::;Lg$, and impose the appropriate boundary conditions on each $_{XY}$ on the outermost lattice sites. When X and Y represent adjacent sides of the square, we assign to $_{XY}$ a value of half the di erence between its two boundary values at the corner where X and Y meet. Now we can compute the magnitude of the gradient of each $_{XY}$ on the plaquette (the set of which we can think of as an $(L \quad 1) \quad (L \quad 1)$ lattice). The numerical magnitude of gradient of $_{XY}$ on the plaquette labeled by $(i;j)$ is given by

$$_{XY}(i;j) = \frac{L}{p\overline{2}}\frac{1}{}\left(_{XY}(i+1;j+1) \quad _{XY}(i;j)\right)^2 + \left(_{XY}(i;j+1) \quad _{XY}(i+1;j)\right)^2 \ ^{\frac{1}{2}}{:} \tag{56}$$

We also de ne variables $(i;j)$ on each plaquette and constrain them by the magnitude of the gradient of all $_{XY}$ on the $(i;j)$th plaquette.

The last thing we must do before proceeding with the computation is de ne our objective. The integral of over the square is given by

$$\text{Objective} = \frac{1}{(L \quad 1)^2}\sum_{i;j=1}^{L\overline{X}1} (i;j){:} \tag{57}$$

Doing this has fully de ned the dual program numerically, so all that is left to do is feed the information into Mathematica's built in optimization tools. Before this is done, however, it is

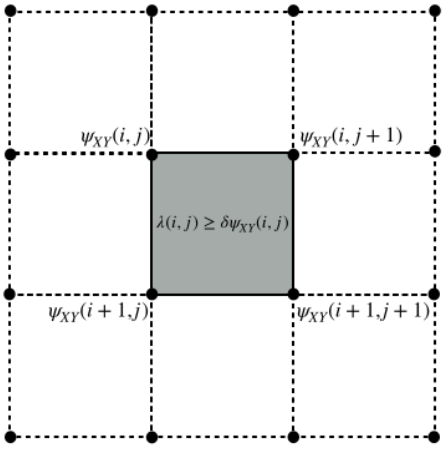


Figure 10: Each site on the L L lattice has the variables $_{XY}$. These are used to constrain the in the square enclosed by neighboring lattice points.

worth noting that we have de ned 6 $(L^2$ $2L) + (L$ $1)^2$ variables to minimize with respect to. Obviously this will grow rapidly as we increase L, so it is in our best interest to reduce the number of variables in our program for the sake of e ciency of computation. Luckily, we have yet to consider the many symmetries of our problem. Doing so will allow us to reduce the number of variables tremendously.

If we want to compute a solution which obeys all the symmetries of our system, we can rst say that out of the six $_{XY}$ we have de ned, only two of them are really independent. This lets us consider only $_{AB}$ and $_{AC}$. The other four $_{XY}$ are related to these two by rotations and re ections. Additionally, we can say that since the which will result from our computation should be symmetric under elements of the Dihedral-4 group, we only need to compute it on one eighth of the square, as shown in Figure (11).

Once these changes were implemented, we computed the solution to the program for a number of lattice sizes. As expected from our analytic endeavors, we found that the which gave the minimum objective value was not the given by a delta function along the minimal surfaces of AB and AC, but rather a much more complicated function.

4.3.2 Results

Qualitatively, it appears that the true minimal solution retains some of the cross structure from the minimal surfaces we were expecting to nd, but much more relaxed and spread out. We found that the did indeed begin to diverge towards in nity in our numerical solution, as it must. In addition there is a local maxima at the center of the square and a raised ridge connecting this maxima to the corners, giving the function a slight cross-like feature. There is also a set of local minima which are displaced from the center of each side of the square.

We can analyze the structure of this solution by considering the condition of complementary

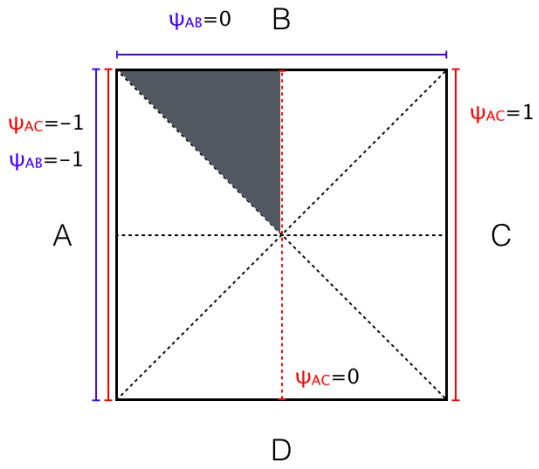


Figure 11: The symmetries of the square allow us to do our computation only for the shaded triangle in the gure. The solution on the rest of the square will be related to that on this triangle by rotations and re ections. The boundary conditions for the independent $_{XY}$s are also shown. We gain a new boundary condition on $_{AC}$ from symmetry.

slackness [5], which tells us that

$$(1 \quad \sum_{X=1}^{} \sum_{Y>X}^{} j\nu_{XY}j) = 0 \tag{58}$$

at every point on the square. In this way, acts a measurement of how much (or how little) the norm bound on the multi ows is saturated in the corresponding optimum of the primal. If = 0 anywhere, this must then mean that at this point the sum of the magnitudes of all the ows in the primal solution will be less than 1. If is greater than zero, then they must sum to exactly one. The higher is at a particular point, the more `pressure' the norm bound is putting on the multi ows at that particular point.

Using this to guide our analysis of the solution to the dual is useful in a few respects, but one is particularly noteworthy. In the solution we compute, is nowhere zero. Thus the multi ow that maximizes the primal program must saturate the norm-bound everywhere on the square, which was not true in our analytic ansatz.

The maximum lattice size we computed a solution for $L = 33$ and this gave an objective value of 2:532. A contour map of the found from this is shown in Figure 12.

4.4 Numerical Check: Primal

4.4.1 Implementation

Now that we have a numerical solution to the dual program, there are a few good reasons to compute that of the primal. Firstly, we can check that the numerical solution to the primal

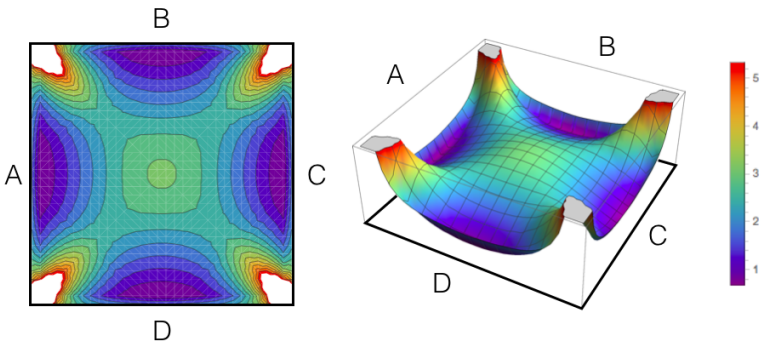


Figure 12: Left: A contour map giving the minimal value of the objective on a $L = 33$ lattice. Right: A 3D plot of the same.

program has a solution which is consistent to that of the dual. Secondly, looking at the maximal multi ow con guration for the program may give us some insight into why the multi ow formalism is unable to compute the entropies for crossing source regions.

We may once again make use of the symmetries of the square to pare down the number of variables that are optimized over. Instead of computing all six ows, we can get away with computing only two: one between adjacent sides and one between opposite sides of the square. All other ows will be related to these by re ections and rotations. We choose to compute ν_{AB} and ν_{AC} explicitly.

To actually compute these ows and the ux through the corresponding regions, it is easiest to de ne pseudoscalar streamfunctions for the ows, the curl of which are the ows themselves.

$$\tilde{\nabla} \quad _{XY} \quad \nu_{XY} \tag{59}$$

Note that this is only permissible because we are dealing with divergenceless vector elds in two dimensions.

There are a few important properties of the streamfunction which will come in handy. First, as alluded to above, this de nition of the ows in terms of streamfunctions automatically gaurantees that the ows will be divergenceless. Second, the magnitude of the the curl of the streamfunction is equal to that of its gradient. Thus

$$j\nu_{XY}j = j\tilde{\nabla} \quad _{XY}j = j\tilde{\nabla} \quad _{XY}j: \tag{60}$$

So we can compute the magnitude of gradient on the lattice for use in imposing the norm-bound on the multi ow, similar to how we imposed the lower bound on in the dual program. Lastly, the fact that ow ν_{XY} will have no ux through a boundary Z if $Z \not\ni X; Y$ tells us that the corresponding streamfunction will be constant along that boundary. By enforcing this, we can de ne a variable $^{0}_{XY}$ which represents the constant value taken by the streamfunction on some boundary. By demanding that the streamfunction be zero on another boundary, or at some appropriate point, we can use the fundamental theorem of calculus to write

$$\int_X \nu_{XY} = \int_X j\tilde{\nabla} \quad _{XY}j = \quad ^{0}_{XY}: \tag{61}$$

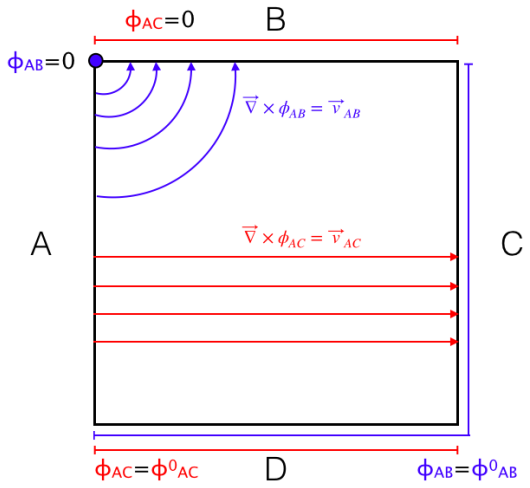


Figure 13: We de ne streamfunctions on the square, which will ensure that the resulting ows are divergencless. The boundary conditions on the streamfunctions ensure that the ows have ux through the appropriate regions. Allowing the streamfunction to take a constant, but undetermined value at one boundary will allow us to compute the total ux of the resulting ow eld.

This lets us express the objective of the primal program as

$$\int_{AB} v_{(AB)(CD)}) + \int_{BC} v_{(BC)(AD)} = 4(\phi^0_{AB} + \phi^0_{AC}): \tag{62}$$

The factor of 4 is included to account for the fact that we only compute two of the six ows, as well as the fact that the ows between opposite sides are weighted twice as heavily as the others.

All that is left to do now before performing the computation is to impose the constraint on the norm of the ows. This turns out to be a non-trivial task. One complication arises from a quirk of Wolfram Mathematica's built in optimization functions. The norm-bound constraint is a sum of absolute values, which is not a smooth function. If this is not corrected, Mathematica's optimization functions would return strange, often non-feasible, solutions or worse yet, would fail entirely. In order to correct for this, we can `smooth' out the norm-bound constraint by changing it to

$$\sum_{X=1}^{} \sum_{Y>X}^{} \sqrt{|\vec{v}_{XY}|^2 + \epsilon} \quad 1: \tag{63}$$

We x the value of epsilon to be su ciently small that it does not change the solution very much, but su ciently large as to resolve the problem described. For the work described here we set $\epsilon = 0.001$.

A second complication with imposing the norm-bound on the lattice comes about when we consider the consequences of how we have calculate the magnitude of gradients on the lattice. If we choose to compute the gradients of ϕ_{XY} in a fashion similar to the one used for the dual

program (56), then one will realize that adding any constant to a function at lattice sites $(i; j)$ and $(i + 1; j + 1)$ will not change the value of the gradient. However, this constant can change the value of the objective. Mathematica's optimization procedure will make use of this, giving rise to strange solutions with objective values above that of the true solution.

To avoid this, we calculate the magnitude of the gradient of $_{XY}$ in a number of di erent ways and impose norm-bounds on all of them. The magnitudes we calculated are:

$$
\begin{aligned}
1 \quad & _{XY}(i; j) = (L \quad 1) \sqrt{(\ _{XY}(i + 1; j) \quad\ _{XY}(i; j))^2 + (\ _{XY}(i; j + 1) \quad\ _{XY}(i; j))^2}; \\
2 \quad & _{XY}(i; j) = (L \quad 1) \sqrt{(\ _{XY}(i + 1; j + 1) \quad\ _{XY}(i + 1; j))^2 + (\ _{XY}(i; j + 1) \quad\ _{XY}(i + 1; j))^2}; \\
3 \quad & _{XY}(i; j) = (L \quad 1) \sqrt{(\ _{XY}(i + 1; j + 1) \quad\ _{XY}(i; j + 1))^2 + (\ _{XY}(i; j) \quad\ _{XY}(i; j + 1))^2}; \\
4 \quad & _{XY}(i; j) = (L \quad 1) \sqrt{(\ _{XY}(i + 1; j + 1) \quad\ _{XY}(i; j + 1))^2 + (\ _{XY}(i + 1; j) \quad\ _{XY}(i + 1; j + 1))^2}:
\end{aligned}
\tag{64}
$$

This step completes the implementation of the primal program in Mathematica. We now proceed to compute the solution to the program on lattices of increasing size.

4.4.2 Results

Starting at $L = 5$ we doubled the resolution until nally performing the optimization on an $L = 65$ lattice. This resulted in an objective value of 2:502, which is consistent with the result from the computation of the dual program. A quick check of the maximal ow con guration con rms that the norm-bound is saturated everywhere on the square, as suggested by our analysis of the solution of to the dual program. By plotting the contours of $_{XY}$ we can see the streamlines of ν_{XY}, which corresponds to the arrangement of the bit threads.

The thread con guration we nd appears to be similar to our analytic ansatz for the primal. The ows between adjacent sides are the square form quarter circles centered at the corners. In addition to this, the ows between adjacent sides start concentrated at the boundary of the square where space is available and spreads out to half the density in the center of the square. The maximal thread con guration for a $L = 65$ lattice is shown in Figure (14).

5 Discussion and Future Work

For both the primal and dual programs we computed solutions numerically on lattices of di erent sizes. Plotting the value of the optimum primal and dual objectives for di erent lattice sizes shows how the optima converge on the lattice. This is shown in gure 15. Based on this, we are con dent that both the result for the primal and dual programs re ect the qualitative form of the optimal solution as well as a descent estimate of the optimum value of the objective functions.

Based on this work, the Locking Theorem represents the point at which multi ows in the continuum begin to yield results which di er from their discrete counterparts. In fact, outside of actually nding the solution given to us by multi ows for a simple problem, there were other clues which signaled that crossing regions may present a problem.

The most notable of these is built into the method used to prove the Continuum Max-Flow Min-Cut theorem for some choice of source regions, an example of which is given by the dualization proof in Section 3. These proofs rely on the fact that we can identify the level sets of $_{ij}(x)$

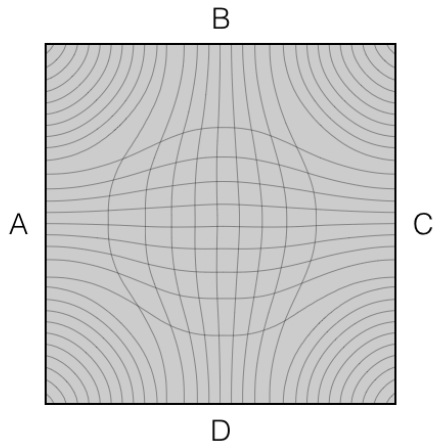


Figure 14: The maximal thread con guration of the primal program as given by the contours of the streamfunctions $_{XY}$.

which are homologous to each source region, and that the regions composed of them do not cross. However if we have crossing regions, then any curves homologous to the two crossing regions must intersect one another. While this was not a death sentence for the idea that multi ows may be able to calculate the entropies for crossing regions, it certainly signals that the issue is non-trivial.

With the question of whether or not mutli ows satisfy a `Continuum Locking Theorem' answered, we are left to wonder why it is that we have reached this conclusion. For answers, we turn back to thinking about a graph model for holographic entanglement entropies.

Knowing the relevant minimal surfaces for the square problem (see Figure 8) allows us to construct a graph model for the square in the way described in Section 2. On this graph we have terminals which represent the regions $A; B; C$ and D and we can ask the discrete version of our original question. What is the maximal commodity ow we can send from AB to CD and from BC to AD simultaneously. Of course, the Locking Theorem along with Max-Flow Min-Cut assures us that this will end up being $2^{p}\overline{2}$. An example of a con guration in which this is achieved is shown in gure 16. With this we can ask; why isn't an analogous con guration possible in the continuum?

While commodity ows on graphics are either parallel or anti-parallel to each other, a ow in the continuum can take on other orientations relative to another ow. This leads to problems wherever two RT surfaces cross in the bulk. Recall that in order for a ow to successfully calculate the entropy of a source region it must have a norm a 1 and be perpendicular to the minimal surface everywhere on the surface. Of course, these two features can not be accomplished simultaneously on two minimal surfaces which cross one another.

In order to further the ability of the continuum multi ow, we may consider changing its de - nition in some respects in an attempt to address this breakdown between the continuum and the discrete. This would be accomplished by modifying the constraints which de ne the multi ow, the most notable being the collective norm-bound

$$\sum_i |\vec{v}_i| \quad 1. \tag{65}$$

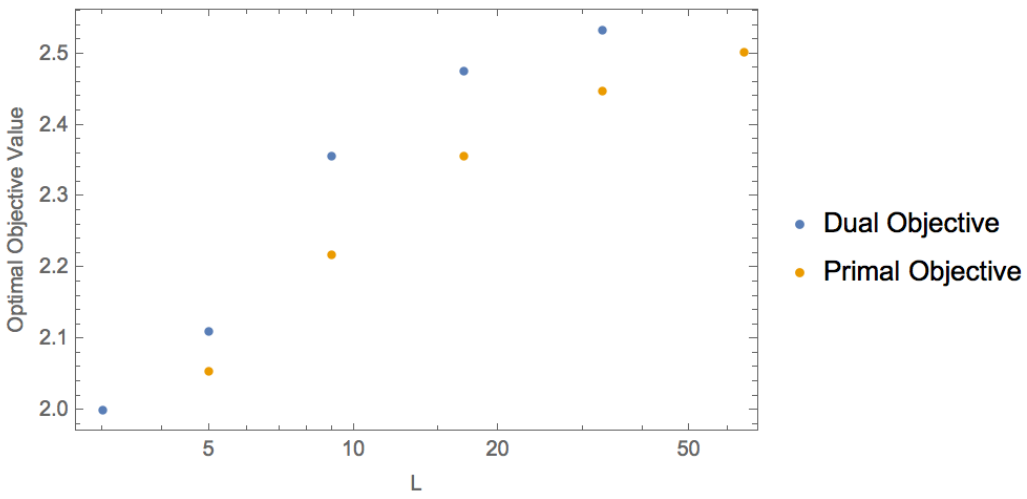


Figure 15: Plot showing optimal value of programs as a function of lattice size L.

This norm-bound is by far the simplest one to work with for multi ows, but it may not be the most physically signi cant. As part of future work, we will be considering norm bounds of the form

$$\sum_i |\forall_i \cdot \hat{n}| \leq 1 \quad \forall \hat{n}; \tag{66}$$

where $\hat{n}$ is a unit vector. Physically, this constraint is the statement that the number of bit threads passing through a surface must not exceed the area of the surface divided by the area of one thread. This is perhaps a more natural constraint for multi ows. It is a weaker constraint than the current norm-bound, but maintains the property that each component of a multi ow is itself a ow. In fact, with this norm bound, we can create a continuum multi ow which is analogous to the discrete commodity ow con guration pictured in Figure 16 in which there is a uniform ow between A and C and another between B and D, both with norm $\frac{p}{2}$. This could be a hint that this norm-bound is the more physically relevant one and should be adopted in place of the current constraint.

We must also acknowledge the possibility that, in order to perform entropy computations for more complicated arrangements of source regions, we may need a new structure altogether. One possibility is that this may be achieved by imposing a weaker constraint on the divergence of multi ows. It could be the case that we require something like a multi ow, but which has non-zero divergence governed by some rules or perhaps even another eld. In order to cover all bases, this must also be considered as a possibility even if its physical meaning, or even its form is not immediately apparent. Regardless, we will explore both possibilities going forward in an attempt to strengthen the multi ow formalism and our understanding of the information structure of holographic states.

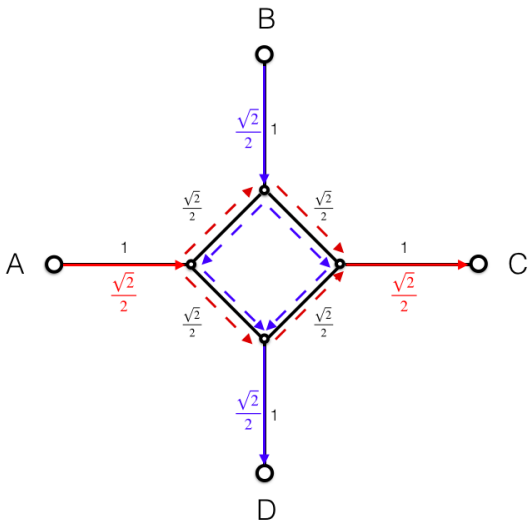


Figure 16: An example of a multi-commodity flow configuration which is maximal for both source regions AB and BC on the graph model corresponding to the square. The edge capacities are shown in black and the commodity flows in blue and red.

6 Acknowledgements

We'd like to thank Professor John Wilmes for his advice on numerical methods in convex programming as well as Jonathan Harper for very helpful discussions on numerous facets of this project as well as comments on earlier drafts of this work.

7 References

[1] Ning Bao, Sepehr Nezami, Hirosi Ooguri, Bogdan Stoica, James Sully, and Michael Walter. The holographic entropy cone. *Journal of High Energy Physics*, 2015(9):130, Sep 2015. arXiv:1505.07839v1.

[2] Michael Freedman and Matthew Headrick. Bit threads and holographic entanglement. *Communications in Mathematical Physics*, 352(1):407{438, May 2017.

[3] G. Naves. "notes on the multicommodity flow problem". http://assert-false.net/callcc/Guyslain/Works/multi flows.

[4] Shawn Cui, Patrick Hayden, Temple He, Matthew Headrick, Bogdan Stoica, and Michael Walter. Bit threads and holographic monogamy. arXiv:1710.09516, 8 2018.

[5] Matthew Headrick and Veronika E Hubeny. Riemannian and lorentzian flow-cut theorems. *Classical and Quantum Gravity*, 35(10), Apr 2018.