Can the external directed edges of a complete graph form a radially symmetric field at long distance?

S. Halayka*

May 9, 2010

Abstract

Using a numerical method, the external directed edges of a complete graph are tested for their level of fitness in terms of how well they form a radially symmetric field at long distance (e.g., a test for the inverse square law in 3D space). It is found that the external directed edges of a complete graph can very nearly form a radially symmetric field at long distance if the number of graph vertices is great enough.

1 Introduction

Complete graphs have been used to construct models of quantum gravity [1–4]. It is considered here that a complete graph $G_1$ consists of:

1. $n(G_1)$ vertices $V(G_1)$ that are uniformly distributed along a shell $S(G_1)$ of radius $r(G_1)$.

2. $(n(G_1)^2 - n(G_1))/2$ internal non-directed edges $I(G_1)$ (e.g., line segments) that join the vertices together.

3. $n(G_1)^2 - n(G_1)$ external directed edges $E(G_1)$ (e.g., rays) that are extensions of $I(G_1)$.

See Figure 1 for a diagram of a complete graph where $n(G_1) = 5$.

It seems fundamentally important to question whether or not the external directed edges $E(G_1)$ can form a radially symmetric field at long distance.

2 Method

If the field is to be considered radially symmetric, then the following two fitness criteria must be met:

*shalayka@gmail.com
1. With regard to a second shell $S(G_2)$ of larger radius $r(G_2) > r(G_1)$, the $n(G_2) = n(G_1)^2 - n(G_1)$ vertices $V(G_2)$ corresponding to where the external directed edges $E(G_1)$ intersect with $S(G_2)$ should be uniformly distributed along $S(G_2)$.

2. The external directed edges $E(G_1)$ should be normal to $S(G_2)$ at their respective intersection vertices.

With regard to the first criterion (e.g., uniform distribution fitness), the vertices $V(G_2)$ will be compared to an equal number $n(G_3) = n(G_2)$ of vertices $V(G_3)$ that are known to be uniformly distributed along a third and final shell $S(G_3)$ of radius $r(G_3) = r(G_2)$.

The generation of $n(G_3)$ uniformly distributed vertices along a 1D shell (e.g., a circle) is algorithmically simple: divide the circle’s $2\pi$ radians into $n(G_3)$ equal portions and then use the polar coordinate equations to generate the $n(G_3)$ corresponding vertex positions. The generation of $n(G_3)$ uniformly distributed vertices along a 2D shell (e.g., a thin spherical shell) is not algorithmically simple: an iterative vertex repulsion code [5] was used here to generate $n(G_3)$ roughly uniformly distributed vertices.

The uniform distribution fitness test used here compares the $n(G_2)$ pairs of vertices $V(G_2)_i, V(G_3)_i$ by analyzing the lengths of their corresponding internal non-directed edges $I(V(G_2)_i)_j, I(V(G_3)_i)_j$ (e.g., where $i = \{1, 2, \ldots, n(G_2)\}$, $j = \{1, 2, \ldots, n(G_2) - 1\}$). Some kind of order must be established so that a reasonable correlation exists between $I(V(G_2)_i)_j, I(V(G_3)_i)_j$, and so the lengths of the internal non-directed edges corresponding to each pair of vertices are placed into a pair of sorted bins before the comparison begins

\[
L(I(V(G_2)_i)_j) = \text{sort}[\text{length}[I(V(G_2)_i)_1], \ldots, \text{length}[I(V(G_2)_i)_n(G_2) - 1]], \quad (1)
\]

\[
L(I(V(G_3)_i)_j) = \text{sort}[\text{length}[I(V(G_3)_i)_1], \ldots, \text{length}[I(V(G_3)_i)_n(G_3) - 1]]. \quad (2)
\]

Ideally, since $V(G_3)$ are known to be uniformly distributed along $S(G_3)$, the $n(G_3)$ sorted bins $L(I(V(G_3)_i))$ should all contain identical length distributions (e.g., thus defining a single reference distribution $L(I(V(G_3)_i))_{\text{ref}}$). Likewise, if $V(G_2)$ are also uniformly distributed along $S(G_2)$, then the $n(G_2)$ sorted bins $L(I(V(G_2)_i))$ should also all contain length distributions that are identical to $L(I(V(G_3)_i))_{\text{ref}}$.

The uniform distribution fitness test used here is

\[
F_{D}(G_1) = [0, 1] = \frac{\sum_{i=1}^{n(G_2)} \sum_{j=1}^{n(G_2) - 1} \min[L(I(V(G_2)_i)_j), L(I(V(G_3)_i)_j)]}{\max[L(I(V(G_2)_i)_j), L(I(V(G_3)_i)_j)]} \frac{1}{n(G_2)^2 - n(G_2)}. \quad (3)
\]

It is useful to note that each of a graph’s internal non-directed edges are analyzed exactly twice throughout the entire test, which is why equation (3) is normalized using $n(G_2)^2 - n(G_2)$, not $(n(G_2)^2 - n(G_2))/2$.

With regard to the second criterion (e.g., normal fitness), each external directed edge $E(V(G_1)_i)_j$ corresponds to one intersection vertex $V(G_2)_k$ (e.g.,
where \( i = \{1, 2, \ldots, n(G_1)\} \), \( j = \{1, 2, \ldots, n(G_1) - 1\} \), \( k = \{1, 2, \ldots, n(G_2)\} \).

Where both \( S(G_1) \) and \( S(G_2) \) are centred at the coordinate system origin, the normal fitness test used here is

\[
F_N(G_1) = [0, 1] = \frac{\sum_{i=1}^{n(G_1)} \sum_{j=1}^{n(G_1)-1} \hat{E}(V(G_1)_i) \cdot \hat{V}(G_2)_k}{n(G_2)}.
\] (4)

### 3 Results

The 1D and 2D shell fitness test results for various \( n(G_1) \), \( r(G_1) \), and \( r(G_2) \) are listed in the following tables

| Uniform distribution fitness \( F_D(G_1) \) for a 1D shell of radius \( r(G_1) = n(G_1) \) |
|---|---|---|---|---|---|---|---|
| \( r(G_2) \) | \( n(G_1) \) | 2 | 4 | 8 | 16 | 32 | 64 | 128 |
| \( 10^4 \) | 1 | 0.829572 | 0.893257 | 0.946621 | 0.993124 | 0.997482 | 0.998891 |
| \( 10^{10} \) | 1 | 0.827916 | 0.886982 | 0.93004 | 0.95885 | 0.976516 | 0.986852 |
| \( 10^{17} \) | 1 | 0.827916 | 0.886982 | 0.93004 | 0.95885 | 0.976516 | 0.986852 |

| Normal fitness \( F_N(G_1) \) for a 1D shell of radius \( r(G_1) = n(G_1) \) |
|---|---|---|---|---|---|---|---|
| \( r(G_2) \) | \( n(G_1) \) | 2 | 4 | 8 | 16 | 32 | 64 | 128 |
| \( 10^4 \) | 1 | 0.999997 | 0.999986 | 0.99994 | 0.999752 | 0.998991 | 0.995924 |
| \( 10^{10} \) | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \( 10^{17} \) | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Uniform distribution fitness \( F_D(G_1) \) for a 2D shell of radius \( r(G_1) = n(G_1) \) |
|---|---|---|---|---|---|---|---|
| \( r(G_2) \) | \( n(G_1) \) | 2 | 4 | 8 | 16 | 32 | 64 | 128 |
| \( 10^4 \) | 1 | 0.937087 | 0.931829 | 0.974859 | 0.97905 | 0.995469 | 0.998372 |
| \( 10^{10} \) | 1 | 0.937088 | 0.930686 | 0.973607 | 0.974738 | 0.994824 | 0.997366 |
| \( 10^{17} \) | 1 | 0.937088 | 0.930686 | 0.973607 | 0.974738 | 0.994824 | 0.997366 |

| Normal fitness \( F_N(G_1) \) for a 2D shell of radius \( r(G_1) = n(G_1) \) |
|---|---|---|---|---|---|---|---|
| \( r(G_2) \) | \( n(G_1) \) | 2 | 4 | 8 | 16 | 32 | 64 | 128 |
| \( 10^4 \) | 1 | 0.999997 | 0.999986 | 0.99994 | 0.999752 | 0.998992 | 0.995925 |
| \( 10^{10} \) | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \( 10^{17} \) | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

### 4 Discussion

As the fitness test results show, the external directed edges of a complete graph can very nearly form a radially symmetric field at long distance if the number of graph vertices is great enough.
On one hand, the external directed edges of a 2D shell in 3D space can very nearly reproduce the inverse-square law of Newtonian gravitation [6] (e.g., field strength proportional to \(1/r\)).

On the other hand, if the external directed edges are instead considered to be bidirectional (e.g., two directly opposing rays per external directed edge), then a 2D shell in 3D space can very nearly reproduce a field strength proportional to \(2/r\) (e.g., akin to the Schwarzschild solution [6]). In terms of rest energy \(E\), the Planck energy \(E_p\), and the Planck length \(\ell_p\), if the number of vertices per Schwarzschild black hole is considered to be

\[
n = \frac{E}{E_p},
\]

and \(n\) is great enough so that the number of external directed edges \(\varepsilon\) practically simplifies

\[
\varepsilon = n^2 - n \approx n^2,
\]

then the Hawking temperature \(T\), Bekenstein-Hawking entropy \(S_{bh}\), event horizon (e.g., 2D shell) radius \(R_s\), and time-time metric component \(g_{00}\) would all simplify to their generally accepted values

\[
T = \frac{E}{k8\pi \varepsilon} \approx \frac{E_p}{k8\pi n},
\]

\[
S_{bh} = 4\pi \varepsilon \approx 4\pi n^2,
\]

\[
R_s = 2\ell_p \sqrt{\varepsilon} \approx 2\ell_p n,
\]

\[
g_{00} = 1 - \frac{2\ell_p \sqrt{\varepsilon}}{r} \approx 1 - \frac{2\ell_p n}{r}.
\]

See [7] for the full code and expanded table data. In the full code, the iterative vertex repulsion code [5] has been modified to use the Mersenne Twister pseudorandom number generator [8] in conjunction with the sphere point picking algorithm discussed in [9]. The full code also uses a modified version of the ray-shell intersection code given in [10].

References


Figure 1: A complete graph $G_1$, where $n(G_1) = 5$ vertices $V(G_1)$ (e.g., black disks) are uniformly distributed along a 1D shell $S(G_1)$ (e.g., a gray circle). There are $(n(G_1)^2 - n(G_1))/2 = 10$ internal non-directed edges $I(G_1)$ (e.g., black line segments), and $(n(G_1)^2 - n(G_1)) = 20$ external directed edges $E(G_1)$ (e.g., outward pointing black rays). Where $i = \{1, 2, \ldots, n(G_1)\}$, $j = \{1, 2, \ldots, n(G_1) - 1\}$, each vertex $V(G_1)_i$ corresponds to $n(G_1) - 1 = 4$ internal non-directed edges $I(V(G_1)_i)_j$ and $n(G_1) - 1 = 4$ external directed edges $E(V(G_1)_i)_j$. 
