

State Space Locality

Copyright © 2020 John Haverlack (john@haverlack.net)
All Rights Reserved
2020-12-06

State Space Locality (SSL) is an attempt to derive emergent approximate 3D spacial relationships between nodes and clusters of nodes in a graph network.

The Newtonian approach was to thinking of space as a background stage upon which the things in that space can move and interact with each other according to Newton's laws of physics in an absolute and universal time. Einstein monumentally advanced our ideas of space and time by merging these concepts into a fabric of spacetime which morph and bend in the presence of mass energy. Einstein's relativity extended the 3D coordinates of space into 4D hyperbolic spacetime coordinates with the invariant signature of $x^2 + y^2 + z^2 - (ct)^2$ (---+). The hyperbolic invariant relationship is one of the properties of space that will need to emerge from the SSL approach. Other characteristic features of space that must also emerge include:

- Affine
- Isotropic
- Spinor rotations
- Hyperbolic Invariant
- Plank Scale Natural Units

A successful SSL model is expected to exhibit the above as emergent characteristics of what we think of as space and time.

SSL has several features similar to hyper-graph networks in the Wolfram Physics Project and is why it this has been written as a Wolfram Notebook. The Wolfram Physics model has already found several of these properties as emergent characteristics, however does take a slightly different approach from SSL.

NOTE: This Notebook will load a few resources and defined some functions. Scroll down to **Getting Started** to begin reading about SSL.

Loading Notebook Resources

This notebook was developed on a Debian Linux system. You may need to adjust the **ConfigureMatex** cell for your system.

NOTE : You may need to also install **pdfLaTeX** for you computer to render MaTeX equations.

* <http://szhorvat.net/pelican/latex-typesetting-in-mathematica.html>

* <https://github.com/szhorvat/MaTeX>

In[645]:= **Debian pdflatex install**

Out[645]= Debian install pdflatex

The following should work on Debian, Ubuntu, Linuxmint, and Raspbian distributions .

```
apt install texlive-science texlive-fonts-extra texlive-extra-utils texlive-latex-extra
texlive-latex-recommended
```

In[646]:= **ResourceFunction ["MaTeXInstall "][]**

Out[646]= PacletObject [ Name : MaTeX
Version : 1.7.8]

In[647]:= **MaTeXInstall []**

Out[647]= MaTeXInstall []

In[648]:= **Needs["MaTeX`"]**

In[649]:= **ConfigureMaTeX ["pdfLaTeX " → "/usr/bin/pdflatex ", "Ghostscript " → "/usr/bin/gs"]**

Out[649]= {CacheSize → 100, WorkingDirectory → Automatic ,
pdfLaTeX → /usr/bin/pdflatex , Ghostscript → /usr/bin/gs}

Proximity Function Definitions

In[650]:= **proxforward [g_, a_, b_] := Sum[g[[a, i]]*g[[i, b]], {i, 1, Dimensions [g][[1]]}]**

In[651]:= **proxreverse [g_, a_, b_] := Sum[g[[i, a]]*g[[b, i]], {i, 1, Dimensions [g][[1]]}]**

In[652]:= **proxinner [g_, a_, b_] := Sum[g[[a, i]]*g[[b, i]], {i, 1, Dimensions [g][[1]]}]**

In[653]:= **proxouter [g_, a_, b_] := Sum[g[[i, a]]*g[[i, b]], {i, 1, Dimensions [g][[1]]}]**

Data

Number graph network nodes (n)

In[654]:= **n = 20**

Out[654]= 20

Number of points on a sphere (p)

In[655]:= **p = 400**

Out[655]= 400

In[656]:= **sphereshell = SpherePoints [p];**

Getting started

Orientation

Before we dive into SSL it might help to navigate a few different perspectives regarding space and time.

Coordinate Spacetime Perspective

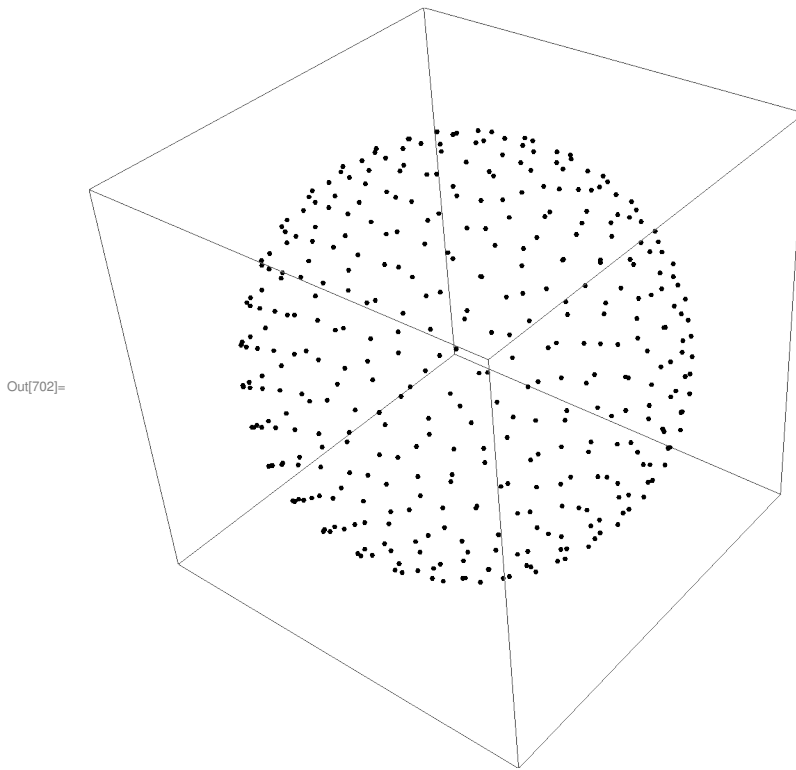
The coordinate perspective represents point in a continuous 3 dimensional space with 3 mutually exclusive space coordinates (x,y,z) where distances (D) are measured with a Pythagorean relationship.

In[657]:= **Show[MaTeX("D_3=\sqrt{x^2+y^2+z^2}"), ImageSize -> Small]**

Out[657]= $D_3 = \sqrt{x^2 + y^2 + z^2}$

A 3D coordinate representation of points on the surface of a sphere.

In[702]:= Graphics3D[Point[sphereshell]]



20

20 :

In relativity time is added as 4th dimension but on in the complex plane. Leaving a hyperbolic invariant measure of distance in spacetime.

In[659]:= Show[MaTeX("D_4^2=x^2+y^2+z^2-(ct)^2=x^2+y^2+z^2+i\\cdot(ct)^2"), ImageSize -> Medium]

Out[659]= $D_4^2 = x^2 + y^2 + z^2 - (ct)^2 = x^2 + y^2 + z^2 + i \cdot (ct)^2$

The coordinate representation of space depends on having a designated origin point, and designated unit vector to establish baseline orientations in isotropic space in what is known as a rest frame. Equations of special relativity can be used to transform coordinates from one rest frame to another, thus there is no one preferred or universal rest frame. Which is kind of the main point of relativity. Measures of spatial and temporal distances are relative to given rest frame and differ when measured from a different rest frame that is moving, or under the influence of a gravitational field. In general relativity the Pythagorean theorem is generalized by a distance metric that defines shortest distance measures between points through curved spacetime.

Distance Relationships Perspective

Rather than representing the spatial relationships of each point in coordinate space we can translate the Pythagorean distance between each point into a **distancematrix**. And plot that matrix to graphi -

cally and as a histogram to show the distribution of distances between all points in our sphere example.

```
In[660]:= distancematrix = Table[Sqrt[(sphereshell[[j]][[1]] - sphereshell[[i]][[1]])^2 +
    (sphereshell[[j]][[2]] - sphereshell[[i]][[2]])^2 +
    (sphereshell[[j]][[3]] - sphereshell[[i]][[3]])^2], {i, p}, {j, p}]

Out[660]= {{0., 0.260557, 0.213377, 0.167762, 0.353608, 0.16496, 0.325774,
    0.362235, 0.197868, 0.436499, 0.306653, 0.33621, 0.473581, ... 374 ...,
    1.95292, 1.9588, 1.98533, 1.95071, 1.98351, 1.97728, 1.96491, 1.99559,
    1.97238, 1.9858, 1.99481, 1.98051, 1.9987}, ... 398 ..., { ... 1 ... }}
```

large output

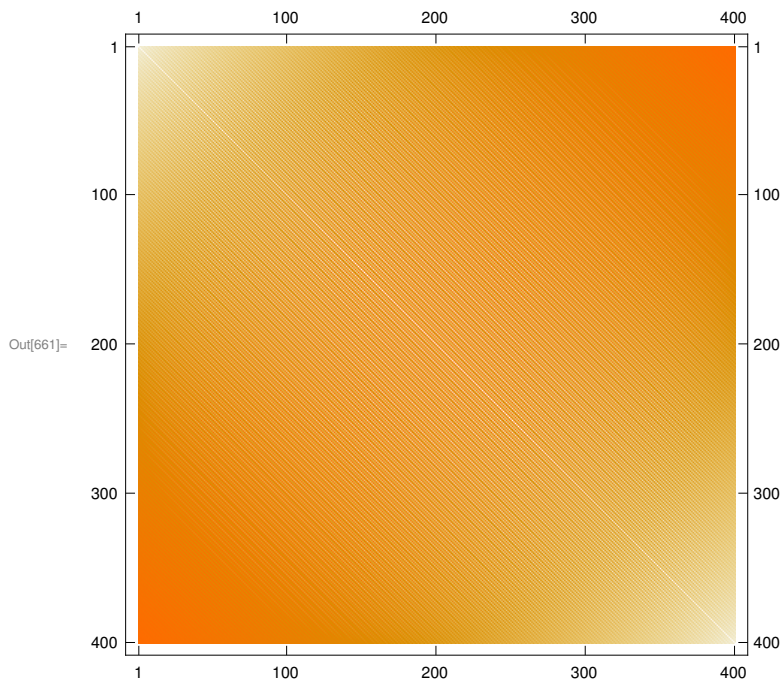
[show less](#)

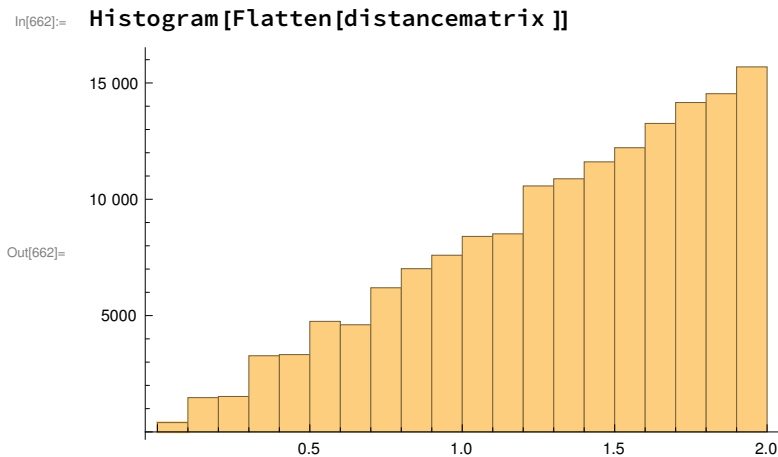
[show more](#)

[show all](#)

[set size limit...](#)

```
In[661]:= MatrixPlot[distancematrix]
```





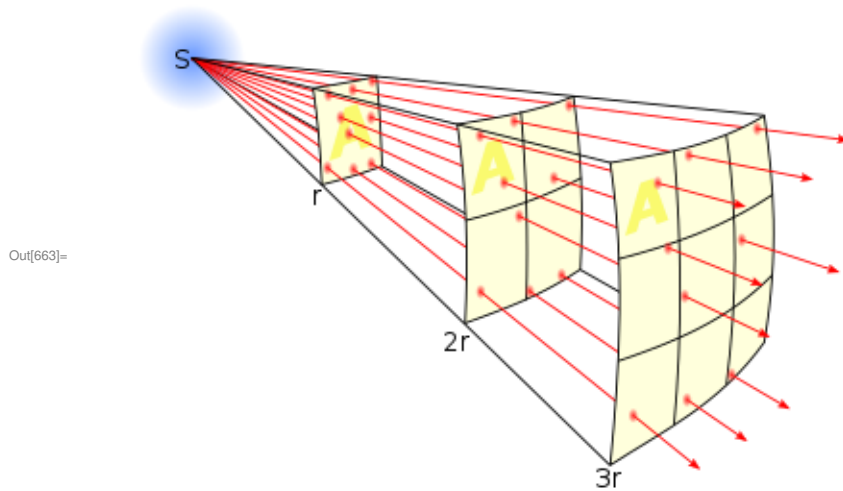
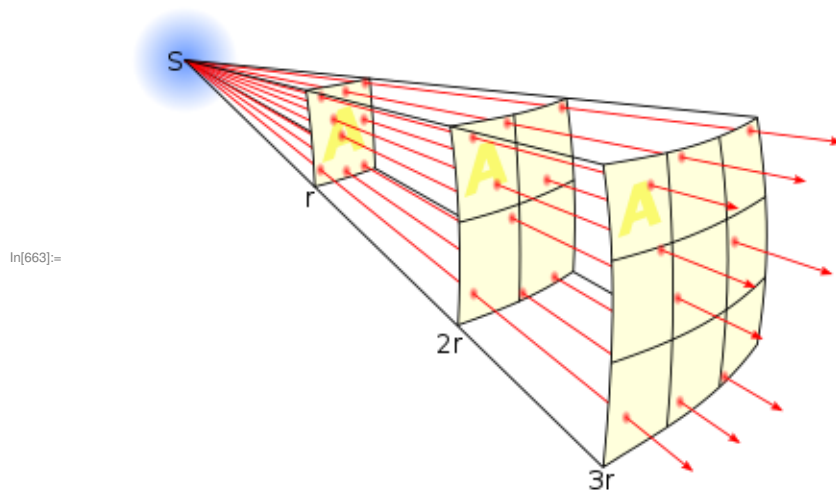
The histogram shows that there is an approximate linear distribution of distances between points distributed evenly over the surface of a sphere. While perhaps less intuitive we can represent 3D space with a matrix of distances between objects in that space.

Multilateration

Multilateration is the process of converting a bunch of known distances between objects back into 3D coordinates. This is the reverse of converting known coordinates into distance distributions, and is the basis behind computing global coordinate position in GPS systems.

Proximity Relationships Perspective

Inverse square distances relationships are understood in terms of conserved amount of flux flowing through the surface of a sphere. A surface that grows as the square of the distance from the center. This can be easily seen in Newton's Law of Gravity, and from Coulomb's Law of force between charges.



However we can also think of the **inverse of the inverse square distance** (D) as **proximity** (P).

In[664]:= **Show[MaTeX("P = \frac{1}{D_3}"), ImageSize → Tiny]**

Out[664]:=

$$P = \frac{1}{D_3}$$

We can understand that in 3D space, those things which are separated by smaller distances, are closer or more proximate to each other. And the more proximate things are the greater the probability those things will interact. Things that are closer will interact more frequently than things farther apart. And increased frequency of interaction can be interpreted as a stronger force interaction.

Interpreting inverse square relationships as an increased probability of interaction due to proximity, rather than flux density diminishing over distance changes 2 things about how we think about forces.

- Forces are probabilistic and thus quantum in nature. Including gravity.
- Strength of force is related to probability of interaction which can be thought of in terms of connectedness in a graph network.

A Simple Random Graph Example

A n node (vertex) directed graph with randomly assigned connections (edges) visualized as:

- Matrix Form of the Graph Adjacency Matrix
- Matrix Plot of the Graph Adjacency Matrix
- 2D Plot of the Graph Network
- 3D Plot of the Graph Network

```
In[665]:= graphnetwork = Table[RandomInteger[], n, n];
```

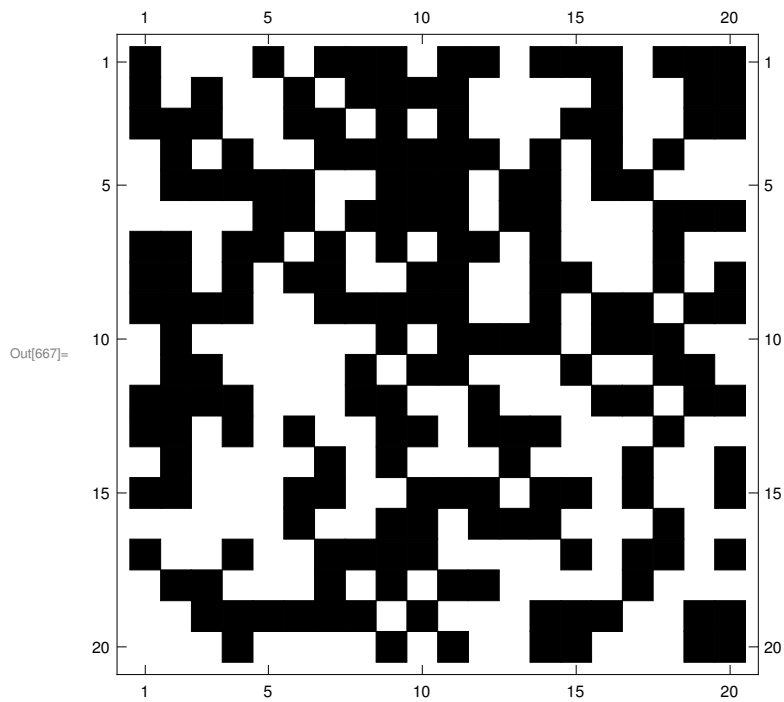
Adjacency Matrix

```
In[666]:= MatrixForm[graphnetwork]
```

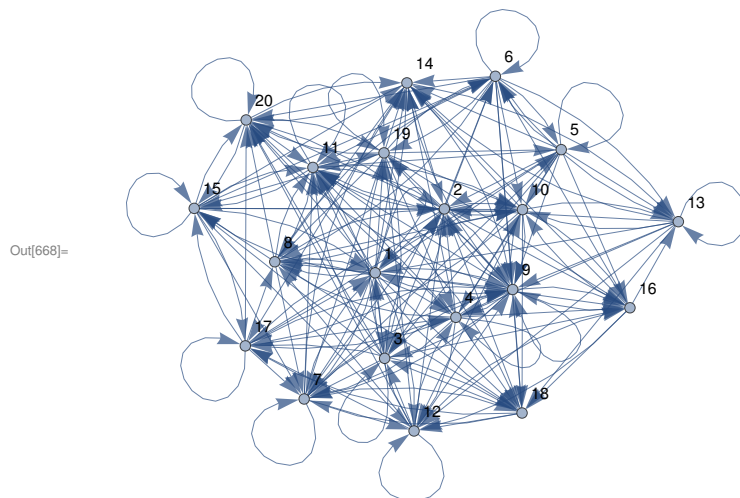
```
Out[666]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

In[667]:= **MatrixPlot**[graphnetwork , ColorFunction → "Monochrome"]

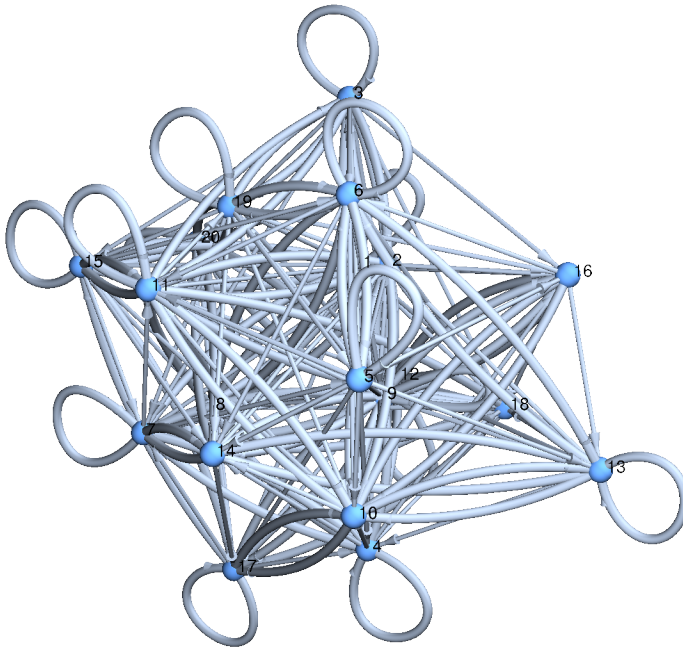


In[668]:= **sg0 = AdjacencyGraph**[graphnetwork , VertexLabels → "Name"]



```
In[669]:= Graph3D[sg0]
```

```
Out[669]=
```



Proximity of Nodes in a Directed Graph Network

In a graph with N nodes, the **proximity** between two nodes (A and B) is defined as the **number of nodes (i) that A is connected to that B is also connected to**.

However in a directed graph there are **4 possible definitions for proximity** . 2 directions per 2 node = 4 combinations of how to measure connectivity.

- Forward Proximity $A \rightarrow i \rightarrow B$
- Reverse Proximity $A \leftarrow i \leftarrow B$
- Inner Proximity $A \rightarrow i \leftarrow B$
- Outer Proximity $A \leftarrow i \rightarrow B$

Forward Proximity and Distance Matrices

```
In[670]:= Show[MaTeX("ProxForward_{a,b} = \sum_{i=1}^N C_{a,i} \cdot C_{i,b}"), ImageSize -> Medium]
```

$$Out[670]= \text{ProxForward}_{a,b} = \sum_{i=1}^N C_{a,i} \cdot C_{i,b}$$

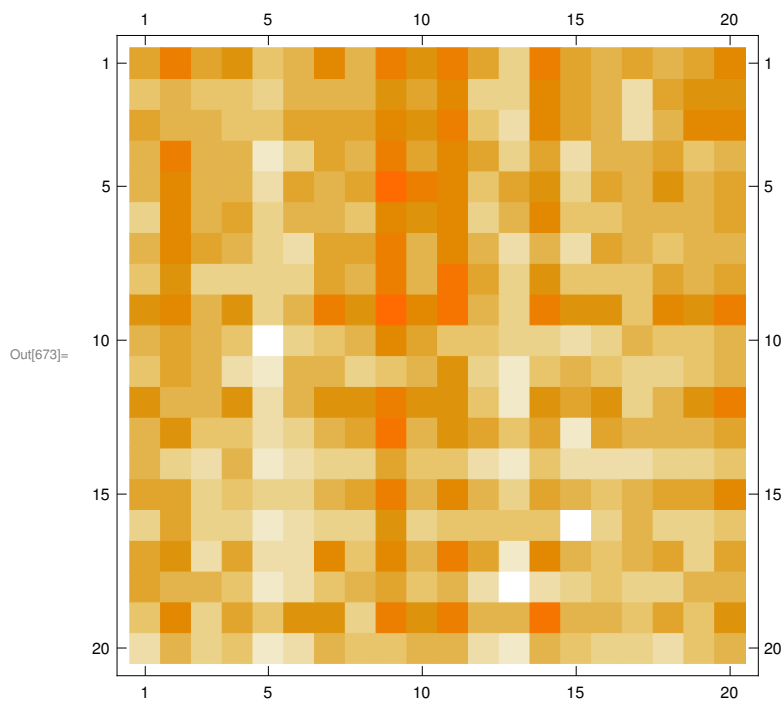
```
In[671]:= proxforwardmatrix = Table[proxforward[graphnetwork, i, j],
  {i, Dimensions[graphnetwork][[1]]}, {j, Dimensions[graphnetwork][[1]]}];
```

In[672]:= **MatrixForm**[proxforwardmatrix]

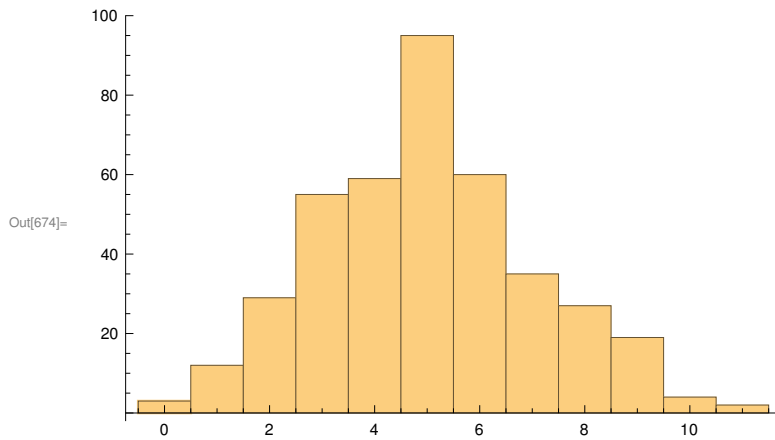
Out[672]//MatrixForm=

$$\begin{pmatrix} 6 & 9 & 6 & 7 & 4 & 5 & 8 & 5 & 9 & 7 & 9 & 6 & 3 & 9 & 6 & 5 & 6 & 5 & 6 & 8 \\ 4 & 5 & 4 & 4 & 3 & 5 & 5 & 5 & 7 & 6 & 8 & 3 & 3 & 8 & 6 & 5 & 2 & 6 & 7 & 7 \\ 6 & 5 & 5 & 4 & 4 & 6 & 6 & 6 & 8 & 7 & 9 & 4 & 2 & 8 & 6 & 5 & 2 & 5 & 8 & 8 \\ 5 & 9 & 5 & 5 & 1 & 3 & 6 & 5 & 9 & 6 & 8 & 6 & 3 & 6 & 2 & 5 & 5 & 6 & 4 & 5 \\ 5 & 8 & 5 & 5 & 2 & 6 & 5 & 6 & 11 & 9 & 8 & 4 & 6 & 7 & 3 & 6 & 5 & 7 & 5 & 6 \\ 3 & 8 & 5 & 6 & 3 & 5 & 5 & 4 & 8 & 7 & 8 & 3 & 5 & 8 & 4 & 4 & 5 & 5 & 5 & 6 \\ 5 & 8 & 6 & 5 & 3 & 2 & 6 & 6 & 9 & 5 & 8 & 5 & 2 & 5 & 2 & 6 & 5 & 4 & 5 & 5 \\ 4 & 7 & 3 & 3 & 3 & 3 & 6 & 5 & 9 & 5 & 10 & 6 & 3 & 7 & 4 & 4 & 4 & 6 & 5 & 6 \\ 7 & 8 & 5 & 7 & 3 & 5 & 9 & 7 & 11 & 8 & 10 & 5 & 3 & 9 & 7 & 7 & 4 & 8 & 7 & 9 \\ 5 & 6 & 5 & 4 & 0 & 3 & 4 & 5 & 8 & 6 & 4 & 4 & 3 & 3 & 2 & 3 & 5 & 4 & 4 & 5 \\ 4 & 6 & 5 & 2 & 1 & 5 & 5 & 3 & 4 & 5 & 7 & 3 & 1 & 4 & 5 & 4 & 3 & 3 & 4 & 5 \\ 7 & 5 & 5 & 7 & 2 & 5 & 7 & 7 & 9 & 7 & 7 & 4 & 1 & 7 & 6 & 7 & 3 & 5 & 7 & 9 \\ 5 & 7 & 4 & 4 & 2 & 3 & 5 & 6 & 10 & 5 & 7 & 6 & 4 & 6 & 1 & 6 & 5 & 5 & 5 & 6 \\ 5 & 3 & 2 & 5 & 1 & 2 & 3 & 3 & 6 & 4 & 4 & 2 & 1 & 4 & 2 & 2 & 2 & 3 & 3 & 4 \\ 6 & 6 & 3 & 4 & 3 & 3 & 5 & 6 & 9 & 5 & 8 & 5 & 3 & 6 & 5 & 4 & 5 & 6 & 6 & 8 \\ 3 & 6 & 3 & 3 & 1 & 2 & 3 & 3 & 7 & 3 & 4 & 4 & 4 & 4 & 0 & 3 & 5 & 3 & 3 & 4 \\ 6 & 7 & 2 & 6 & 2 & 2 & 8 & 4 & 8 & 5 & 9 & 6 & 1 & 8 & 5 & 4 & 5 & 6 & 3 & 6 \\ 6 & 5 & 5 & 4 & 1 & 2 & 4 & 5 & 6 & 4 & 5 & 2 & 0 & 2 & 3 & 4 & 3 & 3 & 5 & 5 \\ 4 & 8 & 3 & 6 & 4 & 7 & 7 & 3 & 9 & 7 & 9 & 5 & 5 & 10 & 5 & 5 & 4 & 6 & 4 & 7 \\ 2 & 5 & 3 & 4 & 1 & 2 & 5 & 4 & 4 & 5 & 5 & 2 & 1 & 5 & 4 & 3 & 3 & 2 & 4 & 5 \end{pmatrix}$$

In[673]:= **MatrixPlot**[proxforwardmatrix]



In[674]:= **Histogram[Flatten[proxforwardmatrix]]**



In[675]:= **distforwardmatrix = Table[1/proxforwardmatrix [[i]][[j]],
{i, Dimensions[graphnetwork][[1]]}, {j, Dimensions[graphnetwork][[1]]}];**

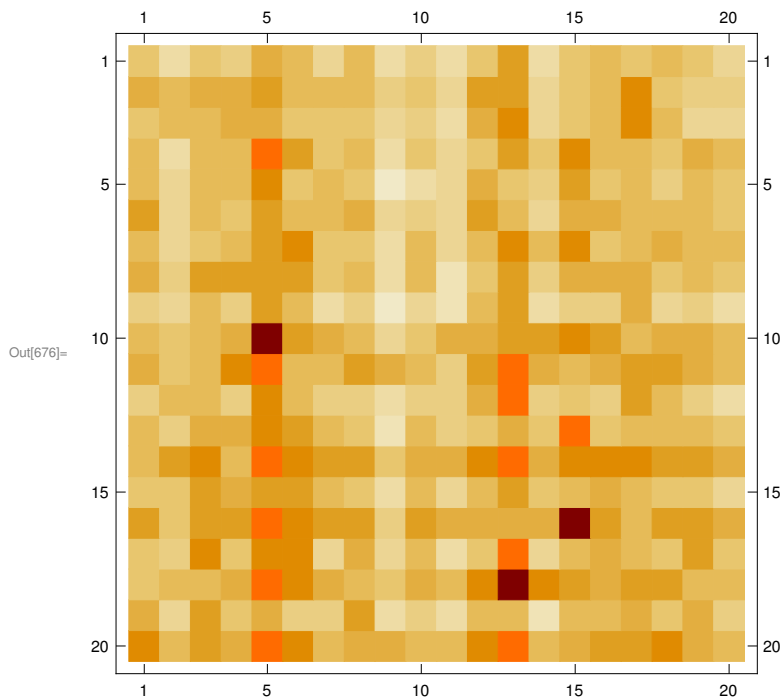
*** **Power** : Infinite expression $\frac{1}{0}$ encountered .

*** **Power** : Infinite expression $\frac{1}{0}$ encountered .

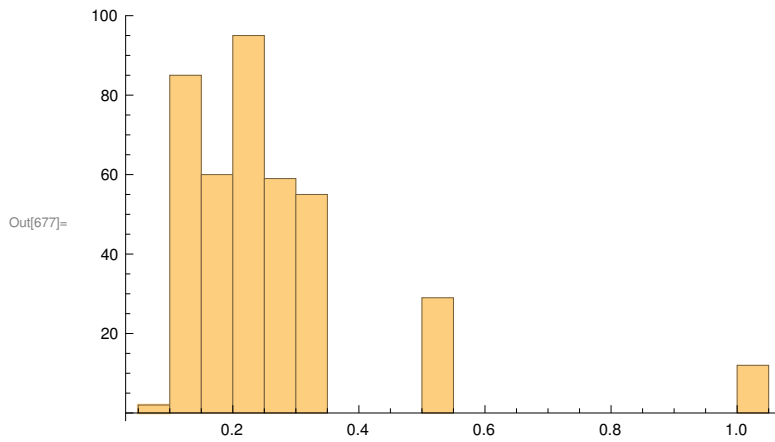
*** **Power** : Infinite expression $\frac{1}{0}$ encountered .

*** **General** : Further output of Power::infy will be suppressed during this calculation .

In[676]:= **MatrixPlot[distforwardmatrix]**



In[677]:= Histogram[Flatten[distforwardmatrix]]



Reverse Proximity Matrix

In[678]:= Show[MaTeX("ProxReverse_{a,b} = \sum_{i=1}^N {C_{i,a}} \cdot C_{b,i}"), ImageSize \rightarrow Medium]

Out[678]=

$$ProxReverse_{a,b} = \sum_{i=1}^N C_{i,a} \cdot C_{b,i}$$

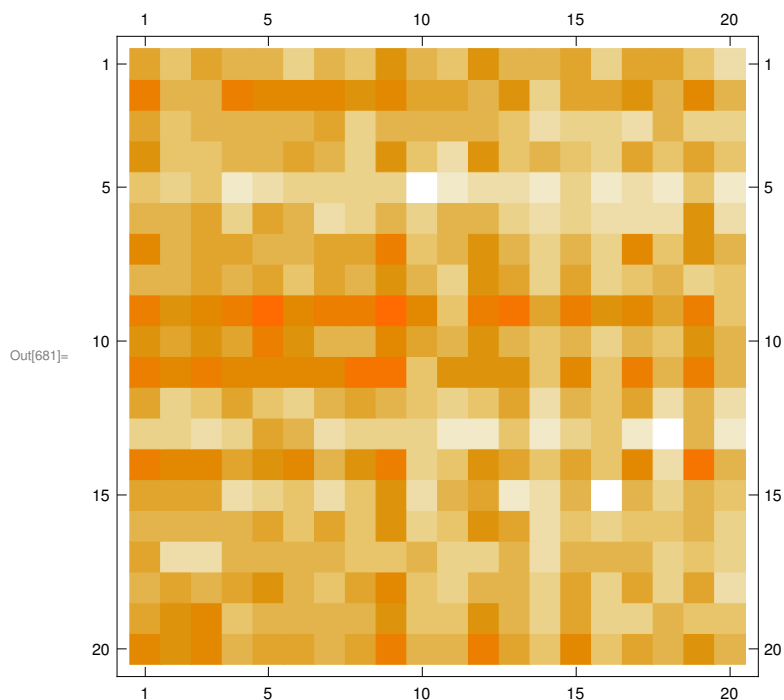
In[679]:= proxreversematrix = Table[proxreverse[graphnetwork, i, j],
{i, Dimensions[graphnetwork][[1]]}, {j, Dimensions[graphnetwork][[1]]};

In[680]:= MatrixForm[proxreversematrix]

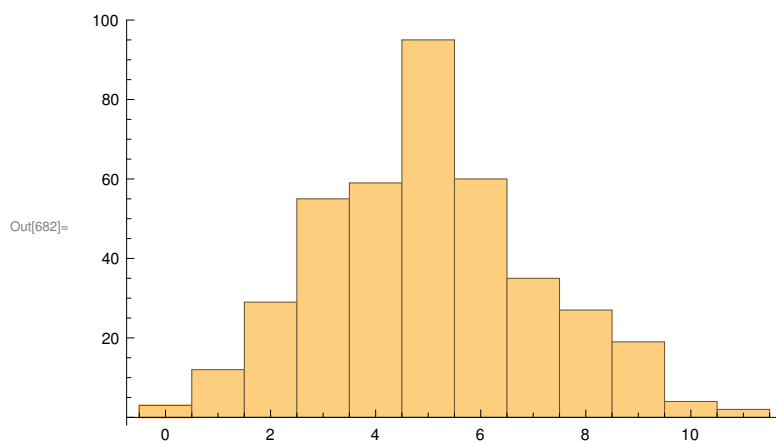
Out[680]/MatrixForm=

$$\begin{pmatrix} 6 & 4 & 6 & 5 & 5 & 3 & 5 & 4 & 7 & 5 & 4 & 7 & 5 & 5 & 6 & 3 & 6 & 6 & 4 & 2 \\ 9 & 5 & 5 & 9 & 8 & 8 & 8 & 7 & 8 & 6 & 6 & 5 & 7 & 3 & 6 & 6 & 7 & 5 & 8 & 5 \\ 6 & 4 & 5 & 5 & 5 & 5 & 6 & 3 & 5 & 5 & 5 & 5 & 4 & 2 & 3 & 3 & 2 & 5 & 3 & 3 \\ 7 & 4 & 4 & 5 & 5 & 6 & 5 & 3 & 7 & 4 & 2 & 7 & 4 & 5 & 4 & 3 & 6 & 4 & 6 & 4 \\ 4 & 3 & 4 & 1 & 2 & 3 & 3 & 3 & 3 & 0 & 1 & 2 & 2 & 1 & 3 & 1 & 2 & 1 & 4 & 1 \\ 5 & 5 & 6 & 3 & 6 & 5 & 2 & 3 & 5 & 3 & 5 & 5 & 3 & 2 & 3 & 2 & 2 & 2 & 7 & 2 \\ 8 & 5 & 6 & 6 & 5 & 5 & 6 & 6 & 9 & 4 & 5 & 7 & 5 & 3 & 5 & 3 & 8 & 4 & 7 & 5 \\ 5 & 5 & 6 & 5 & 6 & 4 & 6 & 5 & 7 & 5 & 3 & 7 & 6 & 3 & 6 & 3 & 4 & 5 & 3 & 4 \\ 9 & 7 & 8 & 9 & 11 & 8 & 9 & 9 & 11 & 8 & 4 & 9 & 10 & 6 & 9 & 7 & 8 & 6 & 9 & 4 \\ 7 & 6 & 7 & 6 & 9 & 7 & 5 & 5 & 8 & 6 & 5 & 7 & 5 & 4 & 5 & 3 & 5 & 4 & 7 & 5 \\ 9 & 8 & 9 & 8 & 8 & 8 & 8 & 10 & 10 & 4 & 7 & 7 & 7 & 4 & 8 & 4 & 9 & 5 & 9 & 5 \\ 6 & 3 & 4 & 6 & 4 & 3 & 5 & 6 & 5 & 4 & 3 & 4 & 6 & 2 & 5 & 4 & 6 & 2 & 5 & 2 \\ 3 & 3 & 2 & 3 & 6 & 5 & 2 & 3 & 3 & 3 & 1 & 1 & 4 & 1 & 3 & 4 & 1 & 0 & 5 & 1 \\ 9 & 8 & 8 & 6 & 7 & 8 & 5 & 7 & 9 & 3 & 4 & 7 & 6 & 4 & 6 & 4 & 8 & 2 & 10 & 5 \\ 6 & 6 & 6 & 2 & 3 & 4 & 2 & 4 & 7 & 2 & 5 & 6 & 1 & 2 & 5 & 0 & 5 & 3 & 5 & 4 \\ 5 & 5 & 5 & 5 & 6 & 4 & 6 & 4 & 7 & 3 & 4 & 7 & 6 & 2 & 4 & 3 & 4 & 4 & 5 & 3 \\ 6 & 2 & 2 & 5 & 5 & 5 & 5 & 4 & 4 & 5 & 3 & 3 & 5 & 2 & 5 & 5 & 5 & 3 & 4 & 3 \\ 5 & 6 & 5 & 6 & 7 & 5 & 4 & 6 & 8 & 4 & 3 & 5 & 5 & 3 & 6 & 3 & 6 & 3 & 6 & 2 \\ 6 & 7 & 8 & 4 & 5 & 5 & 5 & 5 & 7 & 4 & 4 & 7 & 5 & 3 & 6 & 3 & 3 & 5 & 4 & 4 \\ 8 & 7 & 8 & 5 & 6 & 6 & 5 & 6 & 9 & 5 & 5 & 9 & 6 & 4 & 8 & 4 & 6 & 5 & 7 & 5 \end{pmatrix}$$

In[681]:= **MatrixPlot**[proxreversematrix]



In[682]:= **Histogram**[Flatten[proxreversematrix]]



In[683]:= **distreversematrix** = **Table**[1/proxreversematrix[[i]][[j]],
 {i, Dimensions[graphnetwork][[1]]}, {j, Dimensions[graphnetwork][[1]]}];

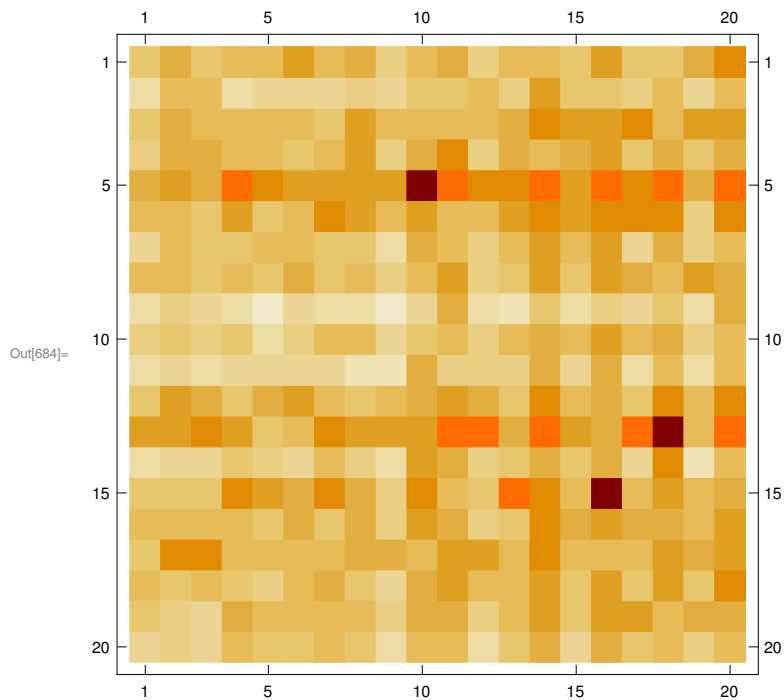
... **Power** : Infinite expression $\frac{1}{0}$ encountered .

... **Power** : Infinite expression $\frac{1}{0}$ encountered .

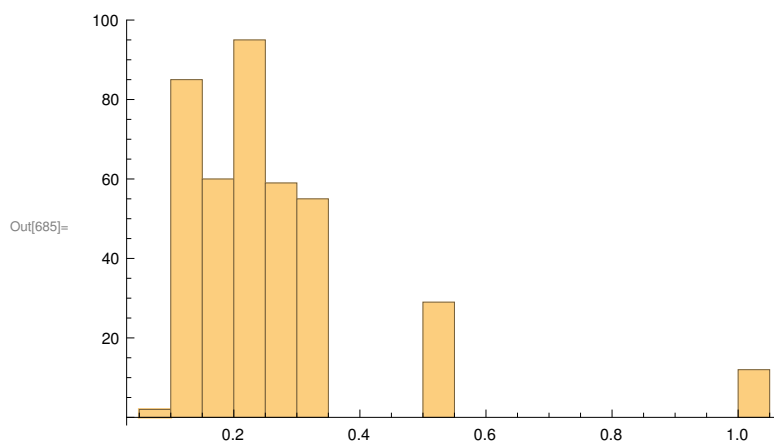
... **Power** : Infinite expression $\frac{1}{0}$ encountered .

... **General** : Further output of Power::infy will be suppressed during this calculation .

In[684]:= **MatrixPlot**[distreversematrix]



In[685]:= **Histogram**[Flatten[distreversematrix]]



Inner Proximity Matrix

In[686]:= **Show**[**MaTeX**("ProxInner_{a,b} = \sum_{i=1}^N C_{a,i} \cdot C_{b,i}"), **ImageSize** → **Medium**]

Out[686]=

$$ProxInner_{a,b} = \sum_{i=1}^N C_{a,i} \cdot C_{b,i}$$

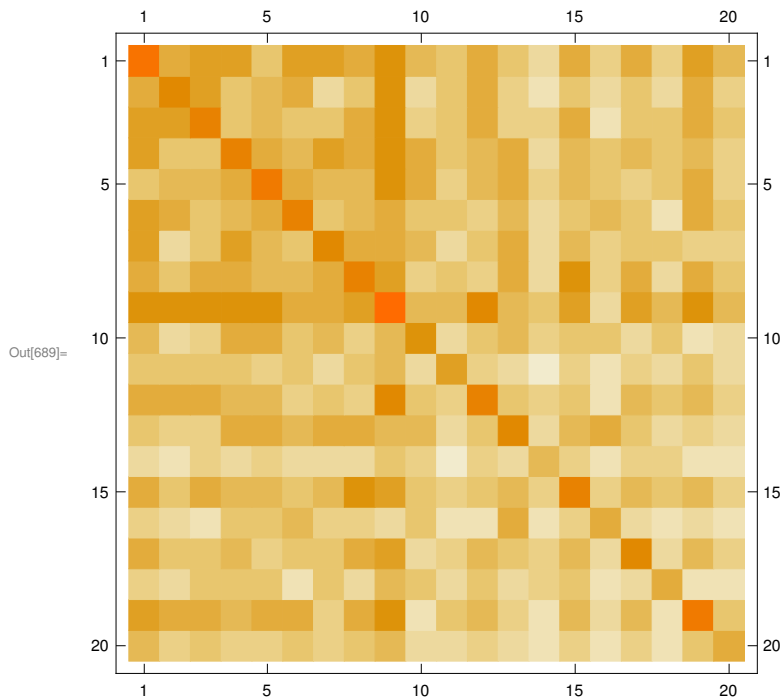
```
In[687]:= proxinnermatrix = Table[proxinner[graphnetwork , i , j],
    {i, Dimensions[graphnetwork ][[1]]}, {j, Dimensions[graphnetwork ][[1]]}];
```

```
In[688]:= MatrixForm[proxinnermatrix ]
```

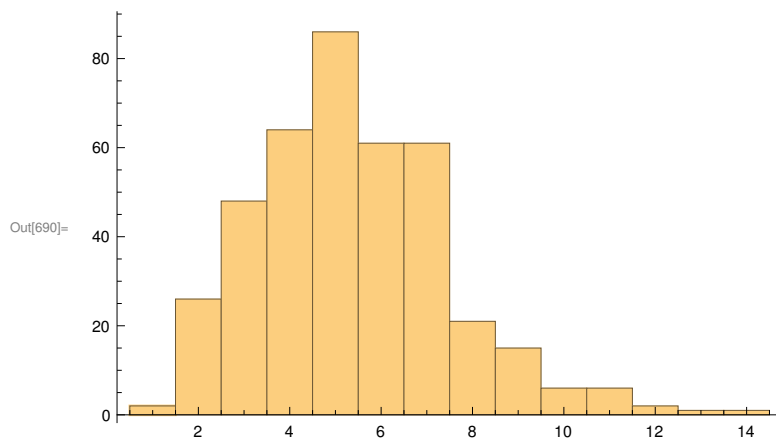
Out[688]/MatrixForm=

13	7	8	8	5	8	8	7	9	6	5	7	5	3	7	4	7	4	8	6
7	10	8	5	6	7	3	5	9	3	5	7	4	2	5	3	5	3	7	4
8	8	11	5	6	5	5	7	9	4	5	7	4	4	7	2	5	5	7	5
8	5	5	11	7	6	8	7	9	7	5	6	7	3	6	5	6	5	6	4
5	6	6	7	12	7	6	6	9	7	4	6	7	4	6	5	4	5	7	4
8	7	5	6	7	11	5	6	7	5	5	4	6	3	5	6	5	2	7	5
8	3	5	8	6	5	10	7	7	6	3	5	7	3	6	4	5	5	4	4
7	5	7	7	6	6	7	11	8	4	5	4	7	3	9	4	7	3	7	5
9	9	9	9	9	7	7	8	14	6	6	10	6	5	8	3	8	6	9	6
6	3	4	7	7	5	6	4	6	9	3	5	6	4	5	5	3	5	2	3
5	5	5	5	4	5	3	5	6	3	8	4	3	1	4	2	4	3	5	3
7	7	7	6	6	4	5	4	10	5	4	11	5	4	5	2	6	5	6	4
5	4	4	7	7	6	7	7	6	6	3	5	10	3	6	7	5	3	4	3
3	2	4	3	4	3	3	3	5	4	1	4	3	6	4	2	4	4	2	2
7	5	7	6	6	5	6	9	8	5	4	5	6	4	11	4	6	5	6	4
4	3	2	5	5	6	4	4	3	5	2	2	7	2	4	7	3	2	3	2
7	5	5	6	4	5	5	7	8	3	4	6	5	4	6	3	10	3	6	4
4	3	5	5	5	2	5	3	6	5	3	5	3	4	5	2	3	7	2	2
8	7	7	6	7	7	4	7	9	2	5	6	4	2	6	3	6	2	12	5
6	4	5	4	4	5	4	5	6	3	3	4	3	2	4	2	4	2	5	7

```
In[689]:= MatrixPlot[proxinnermatrix ]
```

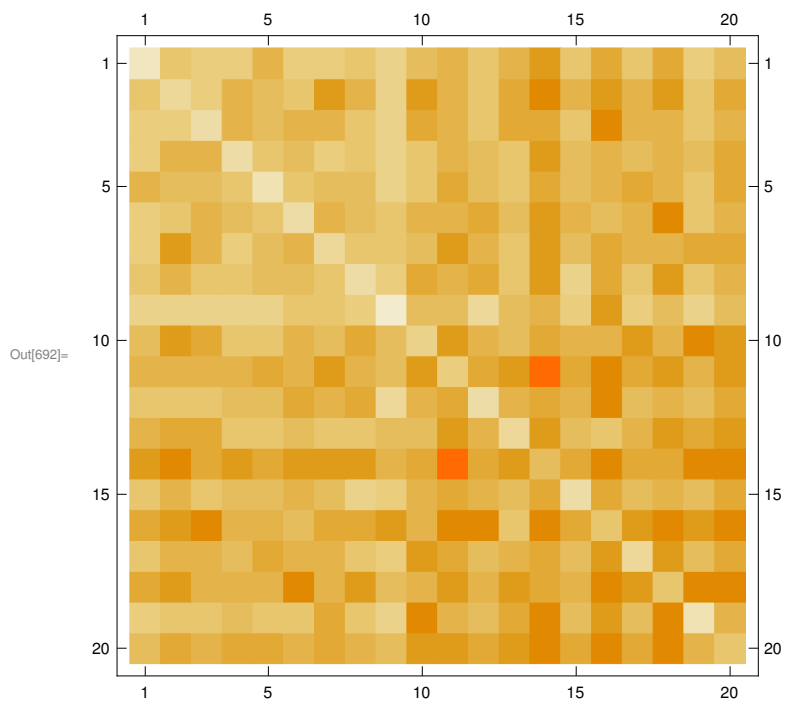


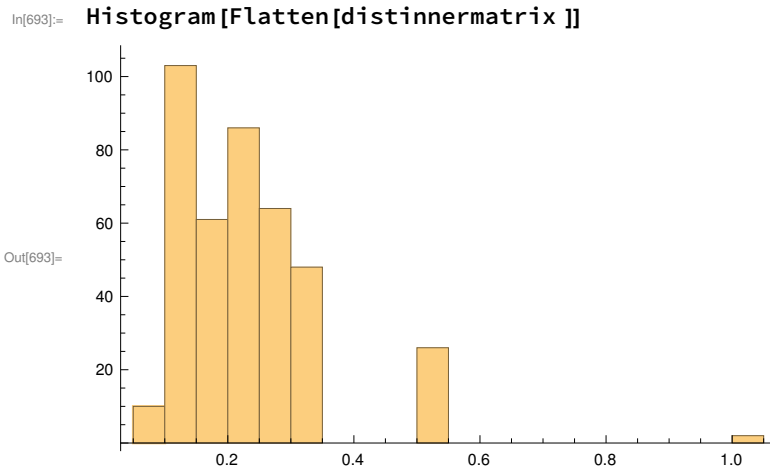
In[690]:= **Histogram**[**Flatten**[proxinnermatrix]]



In[691]:= **distinnermatrix** = **Table**[**1/proxinnermatrix**[[i]][[j]],
 {i, **Dimensions**[**graphnetwork**][[1]]}, {j, **Dimensions**[**graphnetwork**][[1]]}] ;

In[692]:= **MatrixPlot**[**distinnermatrix**]





Outer Proximity Matrix

In[694]:= Show[MaTeX("ProxOuter_{a,b} = \sum_{i=1}^N C_{i,a} \cdot C_{i,b}"), ImageSize -> Medium]

Out[694]=

$$ProxOuter_{a,b} = \sum_{i=1}^N C_{i,a} \cdot C_{i,b}$$

In[695]:= proxoutermatrix = Table[proxouter[graphnetwork, i, j],
 {i, Dimensions[graphnetwork][[1]]}, {j, Dimensions[graphnetwork][[1]]}];

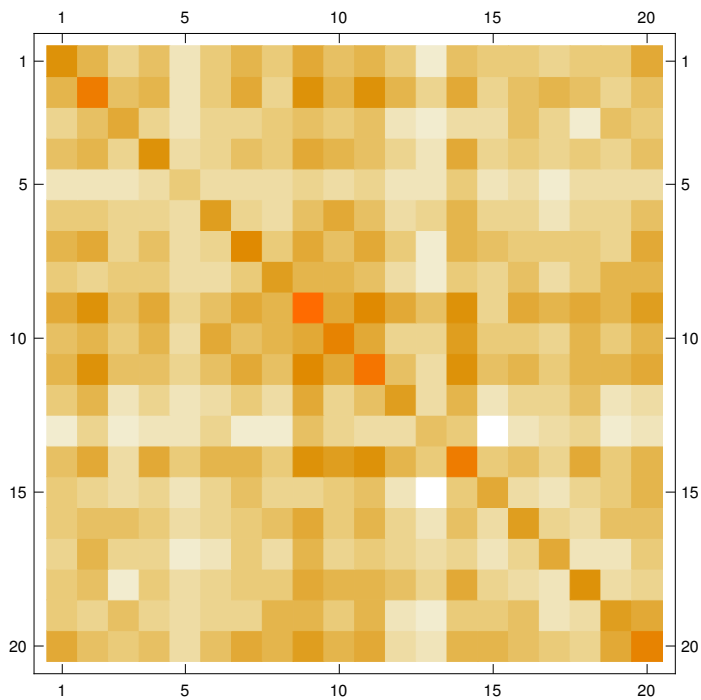
In[696]:= **MatrixForm**[proxoutermatrix]

Out[696]//MatrixForm=

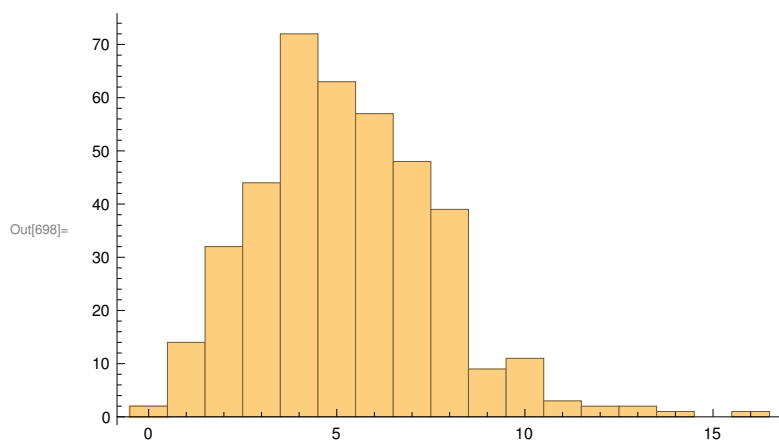
10	7	4	6	2	5	7	5	8	6	7	5	1	6	5	5	4	5	5	8
7	13	6	7	2	5	8	4	10	7	10	7	4	8	4	6	7	6	4	6
4	6	8	4	2	4	4	5	6	5	6	2	1	3	3	6	4	1	6	5
6	7	4	10	3	4	6	5	8	7	6	4	2	8	4	5	4	5	4	6
2	2	2	3	5	3	3	3	4	3	4	2	2	5	2	3	1	3	3	3
5	5	4	4	3	9	4	3	6	8	6	3	4	7	4	4	2	4	4	6
7	8	4	6	3	4	11	5	8	6	8	5	1	7	6	5	5	5	4	8
5	4	5	5	3	3	5	9	7	7	6	3	1	5	4	6	3	5	7	7
8	10	6	8	4	6	8	7	16	8	11	8	6	10	4	8	7	8	7	9
6	7	5	7	3	8	6	7	8	12	8	4	4	9	5	5	4	7	5	7
7	10	6	6	4	6	8	6	11	8	14	6	3	10	6	7	5	7	7	8
5	7	2	4	2	3	5	3	8	4	6	9	3	7	2	4	4	6	2	3
1	4	1	2	2	4	1	1	6	4	3	3	6	5	0	2	3	4	1	2
6	8	3	8	5	7	7	5	10	9	10	7	5	13	5	6	4	8	5	7
5	4	3	4	2	4	6	4	4	5	6	2	0	5	8	3	2	4	5	7
5	6	6	5	3	4	5	6	8	5	7	4	2	6	3	9	4	3	6	6
4	7	4	4	1	2	5	3	7	4	5	4	3	4	2	4	8	2	2	5
5	6	1	5	3	4	5	5	8	7	7	6	4	8	4	3	2	10	3	4
5	4	6	4	3	4	4	7	7	5	7	2	1	5	5	6	2	3	9	8
8	6	5	6	3	6	8	7	9	7	8	3	2	7	7	6	5	4	8	12

In[697]:= **MatrixPlot**[proxoutermatrix]

Out[697]=



In[698]:= **Histogram**[**Flatten**[proxoutermatrix]]



In[699]:= **distoutermatrix** = **Table**[**1/proxoutermatrix**[[i]][[j]],
 {i, **Dimensions**[**graphnetwork**][[1]]}, {j, **Dimensions**[**graphnetwork**][[1]]}] ;

... **Power** : Infinite expression $\frac{1}{0}$ encountered .

... **Power** : Infinite expression $\frac{1}{0}$ encountered .

In[700]:= **MatrixPlot**[**distoutermatrix**]

