SUBMICRON SYSTEMS ARCHITECTURE PROJECT
Department of Computer Science
California Institute of Technology
Pasadena, CA 91125

Mesh Distance Formulae

by

Michael J. Pertel

Caltech Computer Science Technical Report

Caltech-CS-TR-92-05

March 27, 1992

The research described in this report was sponsored by
the Defense Advanced Research Projects Agency.

Mesh Distance Formulae

Michael J. Pertel

March 27, 1992

Abstract

A table of useful summation formulae are derived, together with a Mathematica package for producing them. The distance distribution in mesh routing networks is derived. The mean and variance of the distance distribution are computed. A program for computing the distance distribution of any mesh is presented.

1 Introduction

The author's study of mesh distance-distributions was prompted by his study of routing network simulator convergence. The variance of the path-length distribution in a mesh routing network effects the convergence of simulators [1] used for studying routing networks [2]. For a fine-grained multicomputer [3], such as the Mosaic [4], the number of distinct (src, dst) pairs is too large [1] to simulate every path. Various network parameters converge to their true, asymptotic values as the average distance of the sampled paths converges to the mean distance.

The first section derives summation formulae used in subsequent calculations. A recursive technique for calculating a family of formulae is derived, and it is used to deduce the needed equations. A Mathematica [5] package that can be used to compute any formula in the family is also derived.

The second section develops a recursive technique for computing the distance distribution of any mesh. The distributions of one- and two-dimensional meshes are derived explicitly.

[1] A 128×128 mesh has 268435456 (src, dst) pairs.

The third section computes the mean and variance of the distance distribution. Calculating the variance was the original motivation for studying the distribution, but it was discovered that the moments of the distribution could be computed without knowing the distribution.

The last section presents a simple program that directly computes the distance distribution of any mesh. This program can be used to check the analytical results and to measure properties for which no equations are presented.

2 Summation Formulae

This section derives summation formulae used in calculations. A shorthand notation is defined to simplify the exposition. Lemmas present simple facts used in the proofs. A theorem is proven that provides a recursive technique for computing the desired formulae. The summation formulae are derived as corollaries.

Definition 1 *For integer $h \geq l \geq 0$ and $n \geq 0$, let* $\mathbf{S}^n_{l,h} = \sum_{i=l}^{h} i^n$.

Lemma 1 $\mathbf{S}^n_{l,h} = \mathbf{S}^n_{0,h} - \mathbf{S}^n_{0,l-1}$ *for* $l \geq 1$.

proof: $\mathbf{S}^n_{0,h} = \sum_{i=0}^{h} i^n = \sum_{i=0}^{l-1} i^n + \sum_{i=l}^{h} i^n = \mathbf{S}^n_{0,l-1} + \mathbf{S}^n_{l,h}$

Lemma 2 $\mathbf{S}^0_{0,h} = h + 1$

proof: Induction on h. $\mathbf{S}^0_{0,0} = 1$. $\mathbf{S}^0_{0,h} = \mathbf{S}^0_{0,h-1} + 1$ for $h \geq 1$.

Corollary 1 $\mathbf{S}^0_{l,h} = h - l + 1$

proof: $\mathbf{S}^0_{0,h} = h + 1$ and $\mathbf{S}^0_{l,h} = \mathbf{S}^0_{0,h} - \mathbf{S}^0_{0,l-1} = h + 1 - l$ for $l \geq 1$.

Lemma 3 $\mathbf{S}^n_{0,h} = \mathbf{S}^n_{1,h}$ *if and only if* $n > 0$.

proof: $0^n = 0$ if and only if $n > 0$.

Theorem 1 $\mathbf{S}^n_{0,h} = \sum_{l=1}^{h} \mathbf{S}^{n-1}_{l,h}$ *for* $n \geq 1$.

proof: Induction on h. Base case:

- $\mathbf{S}_{0,0}^{n} = 0$ for $n \geq 1$
- $\sum_{l=1}^{0} (anything) = 0$

Induction step:

$$\begin{aligned}
\mathbf{S}_{0,h}^{n} &= \mathbf{S}_{0,h-1}^{n} + h^{n} \\
&= \sum_{l=1}^{h-1} \mathbf{S}_{l,h-1}^{n-1} + (h-1)h^{n-1} \; + h^{n-1} \\
&= \sum_{l=1}^{h-1} \mathbf{S}_{l,h-1}^{n-1} + \sum_{l=1}^{h-1} h^{n-1} + h^{n-1} \\
&= \sum_{l=1}^{h-1} \left(\mathbf{S}_{l,h-1}^{n-1} + h^{n-1} \right) \; + \mathbf{S}_{h,h}^{n-1} \\
&= \sum_{l=1}^{h-1} \mathbf{S}_{l,h}^{n-1} + \mathbf{S}_{h,h}^{n-1} \\
&= \sum_{l=1}^{h} \mathbf{S}_{l,h}^{n-1}
\end{aligned}$$

Corollary 2 $\mathbf{S}_{l,h}^{1} = \frac{h(h+1)}{2} - \frac{l(l-1)}{2}$

proof:

$$\begin{aligned}
\mathbf{S}_{0,h}^{1} &= \sum_{l=1}^{h} \mathbf{S}_{l,h}^{0} \\
&= \sum_{l=1}^{h} (h - l + 1) \\
&= h(h+1) - \sum_{l=1}^{h} l \\
&= h(h+1) - \mathbf{S}_{0,h}^{1} \\
&= \frac{h(h+1)}{2}
\end{aligned}$$

Corollary 3 $\mathbf{S}^2_{l,h} = \frac{h^2(h+1)}{3} + \frac{h(h+1)}{6} - \frac{l(l-1)^2}{3} - \frac{l(l-1)}{6}$

proof:

$$\begin{aligned}
\mathbf{S}^2_{0,h} &= \sum_{l=1}^{h} \mathbf{S}^1_{l,h} \\
&= \sum_{l=1}^{h} \left(\frac{h(h+1)}{2} - \frac{l(l-1)}{2} \right) \\
&= \frac{h^2(h+1)}{2} - \sum_{l=1}^{h} \frac{l^2}{2} + \sum_{l=1}^{h} \frac{l}{2} \\
&= \frac{h^2(h+1)}{2} - \frac{1}{2}\mathbf{S}^2_{0,h} + \frac{1}{2}\frac{h(h+1)}{2} \\
&= \frac{2}{3}\left(\frac{h^2(h+1)}{2} + \frac{h(h+1)}{4} \right)
\end{aligned}$$

Lemma 4 $\mathbf{S}^n_{0,h} = h\mathbf{S}^{n-1}_{0,h} - \sum_{l=1}^{h} \mathbf{S}^{n-1}_{0,l-1}$ *for* $n > 0$

proof:

$$\begin{aligned}
\mathbf{S}^n_{0,h} &= \sum_{l=1}^{h} \mathbf{S}^{n-1}_{l,h} \\
&= \sum_{l=1}^{h} \left(\mathbf{S}^{n-1}_{0,h} - \mathbf{S}^{n-1}_{0,l-1} \right) \\
&= h\mathbf{S}^{n-1}_{0,h} + \sum_{l=1}^{h} \mathbf{S}^{n-1}_{0,l-1}
\end{aligned}$$

Corollary 4 $\mathbf{S}^3_{l,h} = \left(\frac{h(h+1)}{2}\right)^2 - \left(\frac{l(l-1)}{2}\right)^2$

proof:

$$\mathbf{S}^3_{0,h} = h\mathbf{S}^2_{0,h} - \sum_{l=1}^{h} \mathbf{S}^{n-1}_{0,l-1}$$

$$
\begin{aligned}
&= h\left(\frac{h^2(h+1)}{3}+\frac{h(h+1)}{6}\right)-\sum_{l=1}^{h}\left(\frac{l(l-1)^2}{3}+\frac{l(l-1)}{6}\right)\\
&= \frac{h^3(h+1)}{3}+\frac{h^2(h+1)}{6}-\frac{1}{3}\mathbf{S}^3_{0,h}+\frac{1}{2}\mathbf{S}^2_{0,h}-\frac{1}{6}\mathbf{S}^1_{0,h}\\
&= \frac{3}{4}\left(\frac{h^3(h+1)}{3}+\frac{h^2(h+1)}{3}\right)\\
&= \frac{h^2(h+1)^2}{4}
\end{aligned}
$$

2.1 General Sum Formulae in Mathematica

The recursive technique is simple but tedious; this suggests automation. The technique will be adapted for use with Mathematica [5]. The required Mathematica input will be listed, and a table of sample results will be given.

The Mathematica package will use a refinement of our sum notation. To adapt our technique to dynamic programming, a notation is introduced for the coefficients of the powers of h in a series expansion of $\mathbf{S}^n_{0,h}$.

Definition 2 *Our Mathematica sum notation is:* `s[n,l,h]`$=\mathbf{S}^n_{l,h}$. *The notation for the case when* $l = 0$ *is:* `s[n,x]=s[n,0,x]`. *When the upper limit is implicitly the variable* h*, there is only one parameter:* `s[n]=s[n,h]`.

Definition 3 *The notation for the coefficient of* h^i *in* `s[n]` *is* `c[n,i]`. *That is,* `c[n,i]=Coefficient[s[n],`h^i`]`, *and* `s[n]`$= \sum_{k=0}^{\infty} \texttt{c[n,k]}\, h^k$.

Lemma 5 *For* $n > 0$, `s[n]`$= (h+1)\texttt{s[n}-\texttt{1]} - \sum_{k=0}^{\infty} \texttt{c[n}-\texttt{1,k]}\,\texttt{s[k]}$.

proof:

$$
\texttt{s[n]} = h\mathbf{S}^{n-1}_{0,h} - \sum_{l=0}^{h-1}\mathbf{S}^{n-1}_{0,l} = (h+1)\mathbf{S}^{n-1}_{0,h} - \sum_{l=0}^{h}\sum_{k=0}^{\infty}\texttt{c[n}-\texttt{1,k]}\, l^k
$$

Lemma 6 *The series expansion of* `s[n]` *is finite:* `c[n,k]`$= 0$ *for* $k > n+1$*, so* `s[n]`$= \sum_{k=0}^{n+1}$ $\mathtt{c[n,k]}\, h^k$*. In particular:* $\mathtt{s[n]} = \frac{h^{n+1}}{n+1} + \mathrm{O}(h^n)$*.*

proof: Induction on n. Base case: $\mathtt{s[0]} = h+1$. Induction step: given that for all $m < n$ we have $\mathtt{c[m-1,m]} = \frac{1}{m}$ and $\mathtt{c[m-1,k]} = 0$ for $k > m$, we show that $\mathtt{c[n,n+1]} = \frac{1}{n+1}$ and $\mathtt{c[n,k]} = 0$ for $k > n+1$.

$$
\begin{aligned}
\mathrm{s[n]} &= (h+1)\mathrm{s[n-1]} - \sum_{k=0}^{n} \mathrm{c[n-1,k]\,s[k]} \\
(1 + c[n-1,n])\,\mathrm{s[n]} &= (h+1)\mathrm{s[n-1]} - \sum_{k=0}^{n-1} \mathrm{c[n-1,k]\,s[k]} \\
\left(1 + \frac{1}{n}\right)\mathrm{s[n]} &= (h+1)\left(\frac{h^n}{n} + \mathrm{O}(h^{n-1})\right) - \sum_{k=0}^{n-1} \mathrm{c[n-1,k]} \left(\frac{h^{k+1}}{k+1} + \mathrm{O}(h^k)\right) \\
\left(\frac{n+1}{n}\right)\mathrm{s[n]} &= \frac{h^{n+1}}{n} + \mathrm{O}(h^n)
\end{aligned}
$$

The recurrence relation needed to compute any `s[n]` by dynamic programming follows immediately from the lemmas.

Theorem 2 *For* $n > 0$*,*

$$\mathtt{s[n]} = \frac{n}{n+1}\left((h+1)\mathtt{s[n-1]} - \sum_{k=0}^{n-1} \mathtt{c[n-1,k]\,s[k]}\right)$$

The listing below illustrates the use of the theorem with Mathematica, and the table gives some sample results.

```
c[n_,j_]:= c[n,j]= Coefficient[Collect[s[n],h],h^j]
s[0]= h+1
s[n_]:= s[n]= Expand[(n/(n+1))((h+1)s[n-1]-Sum[c[n-1,k] s[k],{k,0,n-1}])]
TeXForm[Table[s[i],{i,0,10}]]
```

$$
\begin{aligned}
\mathbf{S}^0_{0,h} &= 1 + h && (1) \\
\mathbf{S}^1_{0,h} &= \frac{h}{2} + \frac{h^2}{2} && (2) \\
\mathbf{S}^2_{0,h} &= \frac{h}{6} + \frac{h^2}{2} + \frac{h^3}{3} && (3) \\
\mathbf{S}^3_{0,h} &= \frac{h^2}{4} + \frac{h^3}{2} + \frac{h^4}{4} && (4)
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{S}^{4}_{0,h} &= \frac{-h}{30}+\frac{h^3}{3}+\frac{h^4}{2}+\frac{h^5}{5} && (5)\\
\mathbf{S}^{5}_{0,h} &= \frac{-h^2}{12}+\frac{5h^4}{12}+\frac{h^5}{2}+\frac{h^6}{6} && (6)\\
\mathbf{S}^{6}_{0,h} &= \frac{h}{42}-\frac{h^3}{6}+\frac{h^5}{2}+\frac{h^6}{2}+\frac{h^7}{7} && (7)\\
\mathbf{S}^{7}_{0,h} &= \frac{h^2}{12}-\frac{7h^4}{24}+\frac{7h^6}{12}+\frac{h^7}{2}+\frac{h^8}{8} && (8)\\
\mathbf{S}^{8}_{0,h} &= \frac{-h}{30}+\frac{2h^3}{9}-\frac{7h^5}{15}+\frac{2h^7}{3}+\frac{h^8}{2}+\frac{h^9}{9} && (9)\\
\mathbf{S}^{9}_{0,h} &= \frac{-3h^2}{20}+\frac{h^4}{2}-\frac{7h^6}{10}+\frac{3h^8}{4}+\frac{h^9}{2}+\frac{h^{10}}{10} && (10)\\
\mathbf{S}^{10}_{0,h} &= \frac{5h}{66}-\frac{h^3}{2}+h^5-h^7+\frac{5h^9}{6}+\frac{h^{10}}{2}+\frac{h^{11}}{11} && (11)
\end{aligned}
$$

3 Distance Distribution

For a mesh of radix R and dimension d, $\mathbf{N}^d_R(l)$ denotes the number of (src, dst) pairs separated by distance l.

Definition 4

$$\mathbf{N}^d_{R}(l) = \left| \left\{ \left((a_1,..,a_d),(b_1,..,b_d) \right) \;\middle|\; 0 \le a_i < R, 0 \le b_i < R, \sum_{i=1}^{d} |a_i - b_i| = l \right\} \right|$$

Lemma 7 $\mathbf{N}^d_R(0) = R^d$

proof: $\mathbf{N}^d_R(0) = |\{(a_1,..,a_d) \mid 0 \le a_i < R\}|$

Lemma 8 $\mathbf{N}^d_R(l) = 0$ *for* $l > d(R-1)$

proof: $|a_i - b_i| \le R - 1$

Lemma 9

$$\mathbf{N}^1_R(l) = \begin{cases} R & l = 0 \\ 2(R-l) & 1 \le l \le R-1 \\ 0 & l \ge R \end{cases}$$

proof: For $1 \le l \le R-1$:

$$\begin{aligned} \mathbf{N}_R^1(l) &= 2 \cdot |\{(a,b) \mid 0 \le a < b < R,\, b-a = l\}| \\ &= 2 \cdot |\{(0,l),(1,l+1),..,(R-l-1,R-1)\}| \end{aligned}$$

Theorem 3 $\mathbf{N}_R^d(l) = \sum_{i=0}^{l} \mathbf{N}_R^1(i)\, \mathbf{N}_R^{d-1}(l-i)$

proof:

$$\begin{aligned} \mathbf{N}_R^d(l) &= \left| \left\{ \left((a_1,..,a_d),(b_1,..,b_d) \right) \,\middle|\, 0 \le a_i < R, 0 \le b_i < R, \sum_{i=1}^{d} |a_i - b_i| = l \right\} \right| \\ &= \left| \bigcup_{i=0}^{l} \left\{ \left((a_1,..,a_d),(b_1,..,b_d) \right) \,\middle|\, 0 \le a_i < R, 0 \le b_i < R, |a_d - b_d| = i, \sum_{j=1}^{d-1} |a_j - b_j| = l-i \right\} \right| \\ &= \sum_{i=0}^{l} \left| \left\{ \left((a_1,..,a_d),(b_1,..,b_d) \right) \,\middle|\, 0 \le a_i < R, 0 \le b_i < R, |a_d - b_d| = i, \sum_{j=1}^{d-1} |a_j - b_j| = l-i \right\} \right| \\ &= \sum_{i=0}^{l} \left| \left\{ \left((a_1,..,a_{d-1}),(b_1,..,b_{d-1}) \right) \,\middle|\, \sum_{j=1}^{d-1} |a_j - b_j| = l-i \right\} \times \left\{ (a_d,b_d) \,\middle|\, |a_d - b_d| = i \right\} \right| \\ &= \sum_{i=0}^{l} \left(\left| \left\{ \left((a_1,..,a_{d-1}),(b_1,..,b_{d-1}) \right) \,\middle|\, \sum_{j=1}^{d-1} |a_j - b_j| = l-i \right\} \right| \cdot \left| \left\{ (a_d,b_d) \,\middle|\, |a_d - b_d| = i \right\} \right| \right) \end{aligned}$$

Corollary 5 *For* $0 < l < R$

$$\mathbf{N}_R^d(l) = 2(R-l)R^{d-1} + R\,\mathbf{N}_R^{d-1}(l) + \sum_{i=1}^{l-1} 2(R-l+i)\,\mathbf{N}_R^{d-1}(i)$$

proof:

$$\begin{aligned} \mathbf{N}_R^d(l) &= R\,\mathbf{N}_R^{d-1}(l) + \sum_{i=1}^{l} 2(R-i)\,\mathbf{N}_R^{d-1}(l-i) \\ &= R\,\mathbf{N}_R^{d-1}(l) + \sum_{j=0}^{l-1} 2(R-l+j)\,\mathbf{N}_R^{d-1}(j) \\ &= R\,\mathbf{N}_R^{d-1}(l) + 2(R-l)R^{d-1} + \sum_{j=1}^{l-1} 2(R-l+j)\,\mathbf{N}_R^{d-1}(j) \end{aligned}$$

Corollary 6 *For $l \geq R$,*

$$\mathbf{N}_R^d(l) = R\,\mathbf{N}_R^{d-1}(l) + \sum_{i=l-R+1}^{l-1} 2(R-l+i)\,\mathbf{N}_R^{d-1}(i)$$

proof: $\mathbf{N}_R^d(l) = R\,\mathbf{N}_R^{d-1}(l) + \sum_{i=1}^{R-1} 2(R-i)\,\mathbf{N}_R^{d-1}(l-i)$

Corollary 7

$$\mathbf{N}_R^2(l) = \begin{cases} R^2 & l = 0 \\ 4R^2 l\left(1 - \frac{l}{R} + \frac{l^2}{6R^2} - \frac{1}{6R^2}\right) & 1 \leq l \leq R-1 \\ \frac{16}{3}R^3\left(1 - \frac{3l}{2R} + \frac{3l^2}{4R^2} - \frac{l^3}{8R^3} - \frac{1}{4R^2} + \frac{l}{8R^3}\right) & R \leq l \leq 2(R-1) \\ 0 & l \geq 2R-1 \end{cases}$$

proof: For $0 < l < R$,

$$\begin{aligned} \mathbf{N}_R^2(l) &= 2(R-l)\cdot R + R\cdot 2(R-l) + \sum_{i=1}^{l-1} 2(R-l+i)\cdot 2(R-i) \\ &= 4Rl(R-l) + 4l\,\mathtt{s[1,1,l-1]} - 4\,\mathtt{s[2,1,l-1]} \end{aligned}$$

For $R \leq l \leq 2(R-1)$,

$$\begin{aligned} \mathbf{N}_R^2(l) &= \sum_{i=l-R+1}^{R-1} 2(R-l+i)\cdot 2(R-i) \\ &= 4R(R-l)(2R-l-1) + 4l\,\mathtt{s[1,l-R+1,R-1]} \\ &\quad -4\,\mathtt{s[2,l-R+1,R-1]} \end{aligned}$$

4 Moments of the Distribution

In this section, the mean distance and mean-square distance are computed to allow calculation of the variance. Three theorems are proven. The first theorem states that the average distance in a d-dimensional mesh is d times the average distance in a one-dimensional mesh. The second theorem states that in a d-dimensional mesh the variance of the distance is d times the variance of the distance in a one-dimensional mesh. The third theorem states that the standard (RMS) deviation of the distance divided by the average distance for a d-dimensional mesh is $\frac{1}{\sqrt{d}}$ times the value for a one-dimensional mesh. Moments of one-dimensional meshes are computed as lemmas to provide explicit formulae for arbitrary meshes.

Definition 5 $\langle D^n \rangle_R^d = \frac{1}{R^{2d}} \sum_{l=0}^{d(R-1)} l^n \cdot \mathbf{N}_R^d(l)$

Lemma 10 $\langle D^0 \rangle_R^d = 1$

proof:

$$\sum_{l=0}^{d(R-1)} \mathbf{N}_R^d(l) = \left|\left\{((a_1,..,a_d),(b_1,..,b_d)) \;\middle|\; 0 \le a_i < R,\, 0 \le b_i < R\right\}\right| = R^{2d}$$

Lemma 11 $\langle D^1 \rangle_R^1 = \frac{1}{3}\left(R - \frac{1}{R}\right)$

proof: $\frac{1}{R^2}\sum_{l=1}^{R-1} l \cdot 2(R-l) = \frac{1}{R^2}\left(2R\,\mathbf{S}_{0,R-1}^1 - 2\,\mathbf{S}_{0,R-1}^2\right)$

Lemma 12

$$\langle D^n \rangle_R^d = \frac{1}{R^{2d}} \sum_{a_1=0}^{R-1} \cdots \sum_{a_d=0}^{R-1} \sum_{b_1=0}^{R-1} \cdots \sum_{b_d=0}^{R-1} (|a_1 - b_1| + \ldots + |a_d - b_d|)^n$$

proof:

$$\sum_{\mathbf{a},\mathbf{b}} |\mathbf{a} - \mathbf{b}|^n = \sum_l \left|\left\{(\mathbf{a},\mathbf{b}) \;\middle|\; |\mathbf{a} - \mathbf{b}| = l\right\}\right| \cdot l^n$$

Theorem 4 $\langle D^1 \rangle_R^d = \frac{d}{3}\left(R - \frac{1}{R}\right)$

proof: $\langle D^1 \rangle_R^d = d \cdot \langle D^1 \rangle_R^1$ because

$$\sum_{a_1=0}^{R-1} \cdots \sum_{b_d=0}^{R-1} (|a_1 - b_1| + \ldots + |a_d - b_d|) = d \sum_{a_1=0}^{R-1} \sum_{b_1=0}^{R-1} |a_1 - b_1|$$

Lemma 13 $\langle D^2\rangle_R^1 = \frac{1}{6}(R^2-1)$

proof: $\sum_{a=0}^{R-1}\sum_{b=0}^{R-1}(a-b)^2 = 2R\,\mathbf{S}^2_{0,R-1} - 2\,\mathbf{S}^1_{0,R-1} = \frac{R^2(R^2-1)}{6}$

Lemma 14 $\langle D^2\rangle_R^d = \frac{d(2d+1)}{18}R^2 - \frac{d(4d-1)}{18} + \frac{d(d-1)}{9R^2}$, *for* $d \geq 1$.

proof:

$$
\begin{aligned}
\langle D^2\rangle_R^d &= \frac{1}{R^{2d}}\sum_{a_1=0}^{R-1}\cdots\sum_{b_d=0}^{R-1}\left(\sum_{i=1}^{d}|a_i-b_i|\right)^2 \\
&= \frac{1}{R^{2d}}\sum_{a_1=0}^{R-1}\cdots\sum_{b_d=0}^{R-1}\sum_{i=1}^{d}|a_i-b_i|\sum_{j=1}^{d}|a_j-b_j| \\
&= \frac{1}{R^{2d}}\sum_{a_1=0}^{R-1}\cdots\sum_{b_d=0}^{R-1}\left(\sum_{i=1}^{d}\sum_{j\neq i}|a_i-b_i||a_j-b_j| + \sum_{i=1}^{d}|a_i-b_i|^2\right) \\
&= \sum_{i=1}^{d}\sum_{j\neq i}\left(\frac{1}{R^2}\sum_{a_i=0}^{R-1}\sum_{b_i=0}^{R-1}|a_i-b_i|\right)\left(\frac{1}{R^2}\sum_{a_j=0}^{R-1}\sum_{b_j=0}^{R-1}|a_j-b_j|\right) \\
&\quad + \sum_{i=1}^{d}\left(\frac{1}{R^2}\sum_{a_i=0}^{R-1}\sum_{b_i=0}^{R-1}|a_i-b_i|^2\right) \\
&= d(d-1)\left(\langle D^1\rangle_R^1\right)^2 + d\,\langle D^2\rangle_R^1
\end{aligned}
$$

Corollary 8 $\langle D^2\rangle_R^2 = \frac{5}{9}R^2 - \frac{7}{9} + \frac{2}{9}\frac{1}{R^2}$

Theorem 5 $\langle D^2\rangle_R^d - \left(\langle D^1\rangle_R^d\right)^2 = d\cdot\left(\langle D^2\rangle_R^1 - \left(\langle D^1\rangle_R^1\right)^2\right)$, *for* $d \geq 1$

proof:

$$\left(d(d-1)\left(\langle D^1\rangle_R^1\right)^2 + d\,\langle D^2\rangle_R^1\right) - \left(d\,\langle D^1\rangle_R^1\right)^2 = d\,\langle D^2\rangle_R^1 - d\left(\langle D^1\rangle_R^1\right)^2$$

Theorem 6 $\frac{\sqrt{\langle (D-\langle D\rangle_R^d)^2\rangle_R^d}}{\langle D\rangle_R^d} = \frac{1}{\sqrt{d}} \frac{\sqrt{\frac{R^2}{18}+\frac{1}{18}-\frac{1}{9R^2}}}{\frac{1}{3}(R-\frac{1}{R})}$

proof:

$$\begin{aligned}
\frac{\sqrt{\langle \left(D-\langle D\rangle_R^d\right)^2\rangle_R^d}}{\langle D\rangle_R^d} &= \frac{\sqrt{\langle D^2\rangle_R^d - \left(\langle D^1\rangle_R^d\right)^2}}{\langle D\rangle_R^d} \\
&= \frac{1}{\sqrt{d}} \frac{\sqrt{\langle D^2\rangle_R^1 - \left(\langle D^1\rangle_R^1\right)^2}}{\langle D\rangle_R^1}
\end{aligned}$$

Corollary 9 $\frac{\sqrt{\left\langle \left(D-\langle D\rangle_R^d\right)^2\right\rangle_R^d}}{\langle D\rangle_R^d} = \frac{1}{\sqrt{2}\sqrt{d}} + \frac{3}{2\sqrt{2}\sqrt{d}R^2} + O(\frac{1}{R^4})$

5 Program for Distribution

This section presents a simple program that computes the mesh distance distribution directly for a specified radix and dimension. The program does not use any of the analysis, so it provides independent confirmation of results. Since it computes the distribution directly (brute force), the program is very short, simple, and obviously correct. The program could be made more efficient, but it is fast enough for practical purposes.

The program can compute the distance distribution for any mesh, and the distribution can be used to compute any desired statistic. The closed-form expression for the mesh distance distribution is tedious to derive and use for dimensions greater than 2, so it is preferable to use the program for high-dimensional meshes. An example of the program's utility is that it not only computes the mean and standard deviation (for which simple formulae were derived in the previous section), but it also computes the probability that a distance lies within one standard deviation of the mean (a useful figure for which no formula is given). The program listing appears in Figure 1, and Table 1 gives sample results.

```
/* ddist.c
 * Invoking "ddist R d" computes the distance distribution
 * for a radix-R d-dimensional mesh.
 */
#include <stdio.h>
#include <math.h>
#include <malloc.h>
main(argc,argv) char **argv; int argc;
{
        int R=atoi(argv[1]), d=atoi(argv[2]);
        int N=(int)pow((double)R,(double)d);
        int i,j,stride,dim,dist,l,m; double m0=0.,m1=0.,m2=0.,sd;
        double *c=(double*)malloc((unsigned)((d*(R-1)+1)*sizeof(double)));
        printf("R=%d, d=%d, N=%d\n",R,d,N);
        for(i=0; i<=d*(R-1); i++) c[i]=0.;
        for(i=0; i<N; i++) for(j=0; j<N; j++) {
                for(dist=dim=0, stride=1; dim<d; dim++, stride*=R)
                        dist+=abs((i/stride)%R - (j/stride)%R);
                c[dist]++; }
        for(i=0; i<=d*(R-1); i++)
                printf("count[%d]=%g\n",i,c[i]),
                m0+=c[i],m1+=i*c[i],m2+=i*i*c[i];
        printf("#paths = %g (N^2=%d)\n",m0,N*N);
        printf("avg(D) = %g (d*(R-1/R)/3=%g)\n",m1/=m0,d*(R-1./R)/3.);
        printf("avg(D^2)=%g \n",m2/=m0);
        printf("std dev= %g\n",sd=sqrt(m2-m1*m1));
        printf("std dev/avg(D)=%g\n",sd/m1);
        printf("range (%d,%d)  ",l=(int)(m1-sd),m=(int)(m1+sd+.5));
        for(i=l, j=0; i<=m; i++) j+=c[i];
        printf("confidence %g\n",((double)j)/m0);
}
```

Figure 1: Simple Program for Computing Distance Distribution

mesh	avg dist	std dev	std dev/avg	range	confidence
4×1	1.25	.968	.775	0–2	88%
8×1	2.63	1.90	.724	0–5	91%
16×1	5.31	3.78	.711	1–9	77%
4×4	2.50	1.37	.548	1–4	86%
32×1	10.7	7.55	.708	3–18	67%
64×1	21.3	15.1	.707	6–36	65%
8×8	5.25	2.69	.512	2–8	81%
$4 \times 4 \times 4$	3.75	1.68	.447	2–5	76%
128×1	42.7	30.2	.707	12–73	65%
16×16	10.6	5.34	.503	5–16	72%
$8 \times 8 \times 8$	7.88	3.29	.417	4–11	77%
32×32	21.3	10.7	.501	10–32	70%
64×64	42.7	21.3	.500	21–64	68%
$16 \times 16 \times 16$	15.9	6.54	.411	9–22	70%
128×128	85.3	42.7	.500	42–128	67%
$32 \times 32 \times 32$	32.0	13.1	.409	18–45	70%

Table 1: Sample Results from Program

Execution time is $O(dN^2) = O(dR^{2d})$. On a Sun SPARCstation using `cc` without optimization, the $32 \times 32 \times 32$ calculation takes roughly 21 hours, the 128×128 calculation takes roughly 3.5 hours, and the smaller meshes take less than 20 minutes.

6 Acknowledgements

The research described in this report was sponsored by the Defense Advanced Research Projects Agency. Supported was provided by a National Defense Science and Engineering Graduate Fellowship. I gratefully acknowledge the direction and assistance of my advisor, Dr. C.L. Seitz. I would also like to thank H. P. Hofstee for his proofreading.

References

[1] Pertel MJ: A Simple Simulator for Multicomputer Routing Networks. Caltech Computer Science Technical Report: CS-TR-92-04.

1992.

[2] Pertel MJ: A Critique of Adaptive Routing. Caltech Computer Science Technical Report: CS-TR-92-06. 1992.

[3] Seitz CL: Chapter 5 "Multicomputers", pp. 131–200, in Hoare CAR (ed): *Developments in Concurrency and Communication.* Addison-Wesley, 1990.

[4] Seitz CL et al: §2.1 "The Mosaic Project", pp. 2–10, in Caltech Computer Science Technical Report: CS-TR-91-10 Semiannual Technical Report, 1991.

[5] Wolfram S: *Mathematica: A System for Doing Mathematics by Computer*. Addison-Wesley, 1988.