$$\Delta_N(s) = 1^{-s} + \delta_{N,2}2^{-s} + \cdots + \delta_{N,N}N^{-s}$$

$$N = 2M + 1$$

$$\Delta_N(\overline{\rho_M}) = \cdots = \Delta_N(\overline{\rho_1}) = 0 = \Delta_N(\rho_1) = \cdots = \Delta(\rho_M)$$

$$\begin{vmatrix} 1 & 1 & \dots & 1 & 1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ n^{-\overline{\rho_1}} & n^{-\rho_1} & \dots & n^{-\overline{\rho_M}} & n^{-\rho_M} & n^{-s} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ N^{-\overline{\rho_1}} & N^{-\rho_1} & \dots & N^{-\overline{\rho_M}} & N^{-\rho_M} & N^{-s} \end{vmatrix} = \sum_{n=1}^{N} \tilde{\delta}_{N,n} n^{-s}$$

$$\delta_{N,n} = \frac{\tilde{\delta}_{N,n}}{\tilde{\delta}_{N,1}}$$

```
PathToValues
```

Loading precomputed values of Log[n] as NLogk[n]

```
Map[Close, Drop[Streams[], 2]];
handle = OpenRead[PathToValues <> "nlogk.vle", BinaryFormat → True]
ReadFull1ArrayVle[handle]
Close[handle];
```

```
InputStream[C:/users/s_pauli/Desktop/nlogk.vle, 29]
```

```
NLogk
```

Loading Imaginary parts of zeta zeroes as NImZetaZero[n]

```
Map[Close, Drop[Streams[], 2]];
handle = OpenRead[PathToValues <> "imLzero_001_001.vle", BinaryFormat → True]
ReadFull1ArrayVle[handle]
Close[handle];
```

```
InputStream[C:/users/s_pauli/Desktop/imLzero_001_001.vle, 28]
```

```
NImZetaZero
```

Loading Imaginary parts of L[3,2] as imLzero32[n]

```
Map[Close, Drop[Streams[], 2]];
handle = OpenRead[PathToValues <> "imLzero_003_002.vle", BinaryFormat → True]
ReadFull1ArrayVle[handle, "imLzero32"]
Close[handle];
```

```
InputStream[C:/users/s_pauli/Desktop/imLzero_003_002.vle, 34]
```

```
imLzeros_003_002
```

TrueZeroes[ ], Multiplicities[ ], and ns[ ] contain input data

```
GeneralFillDeltaMatrixComplex[TrueZeroes_, Multiplicities_, ns_, Nsize_] :=
 Module[{answer, nplace, n, zeroplace, truezero, m},
  answer = Table[0, {nplace, 1, Nsize}, {zeroplace, 1, Nsize}];
  For[nplace = 1, nplace <= Nsize, nplace++,
   Prin["nplace=", nplace];
   n = ns[[nplace]];
   zeroplace = 1; zeronumber = 1;
   While[zeroplace ≤ Nsize,
    Prin["zeroplace=", zeroplace, " mult=", Multiplicities[[zeronumber]]];
     truezero = TrueZeroes[[zeronumber]];
    m = 1;
    While[zeroplace ≤ Nsize && m ≤ Multiplicities[[zeronumber]],
     Prin["m=", m];
     answer[[nplace, zeroplace]] =
      n^(-truezero) * If[n === 1 && m === 1, 1, (NLogk[n])^(m - 1)];
     zeroplace++; m++];
    zeronumber++]];
  answer]
```

Setting parameters

```
MatrixSize = 17; Nprecision = 100;
```

ActualCaliculations

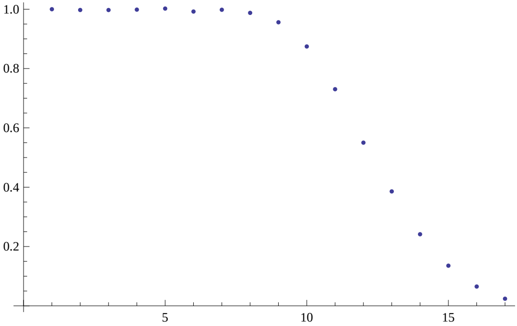
```
tz = Table[N[{1 / 2 + I * NImZetaZero[k], 1 / 2 - I * NImZetaZero[k]}, Nprecision],
     {k, 1, MatrixSize / 2 + 1}] // Flatten;
a = GeneralFillDeltaMatrixComplex[tz, Table[1, {MatrixSize}],
   Table[k, {k, 1, MatrixSize}], MatrixSize];
DIAGONAL = diag = Table[0, {MatrixSize + 2}];
L = Table[0, {MatrixSize + 2}, {MatrixSize + 2}];
For[i = 1, i ≤ MatrixSize, i ++,
  diag [[i]] =  a [[i, i]];
    DIAGONAL [[i]] =  diag [[i ]] * If[i === 1, 1,  DIAGONAL [[i - 1 ]]];
  For[j = i + 1, j ≤ MatrixSize, j ++,
   z = -  a [[j, i ]] /  diag [[i ]];
     L [[j, i]] = z;   a [[j, i]] = z;
   For[k = i + 1, k ≤ MatrixSize, k ++,
     a [[j, k]] =  a [[j, k  ]] +  a [[i, k ]] *  a [[j, i ]]; L[[j, k]] = a[[j, k]];
   ];
  ]];
For[k = 1, k ≤ MatrixSize, k ++,
  For[i = k + 1, i ≤ MatrixSize, i ++, For[j = i + 1, j ≤ MatrixSize, j ++,
      L [[j, k]] =  L [[j, k ]] +  L [[i, k ]] *  a [[j, i]];];
   ];
 ];
For[j = 1, j ≤ MatrixSize, j ++,
  For[k = 1, k < j, k ++,
   LinVectElemNew[j, k] = L[[j, k]] * If[j === 1, 1, DIAGONAL[[j - 1]]]];
  LinVectElemNew[j, j] = If[j === 1, 1, DIAGONAL[[j - 1]]];
  InvNTildedelta[j] = 1 / LinVectElemNew[j, 1];
  For[k = 1, k ≤ j, k ++,
```

Viewing the results

```
(tt = Table[Ndelta[MatrixSize, n] // Re, {n, 1, MatrixSize}]) // N
```

```
{1., 0.997396, 0.997268, 0.998583, 1.00213, 0.992146, 0.998154, 0.987652, 0.955982,
 0.874401, 0.730338, 0.550145, 0.38581, 0.241425, 0.135406, 0.0651171, 0.024006}
```

```
tt // ListPlot
```


1.0
0.8
0.6
0.4
0.2
5
10
15