グラフ距離行列とランク

球面上の木

鎖

ランクは２(両端さえ決めれば、その途中の頂点は説明がつく)

```
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 3.4.4
```

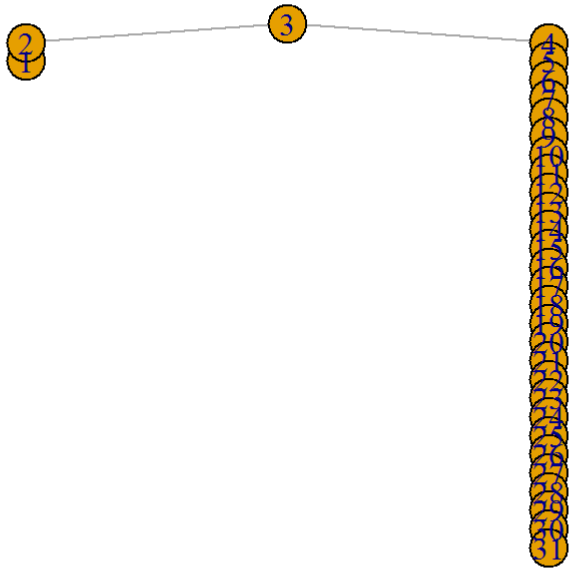
```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

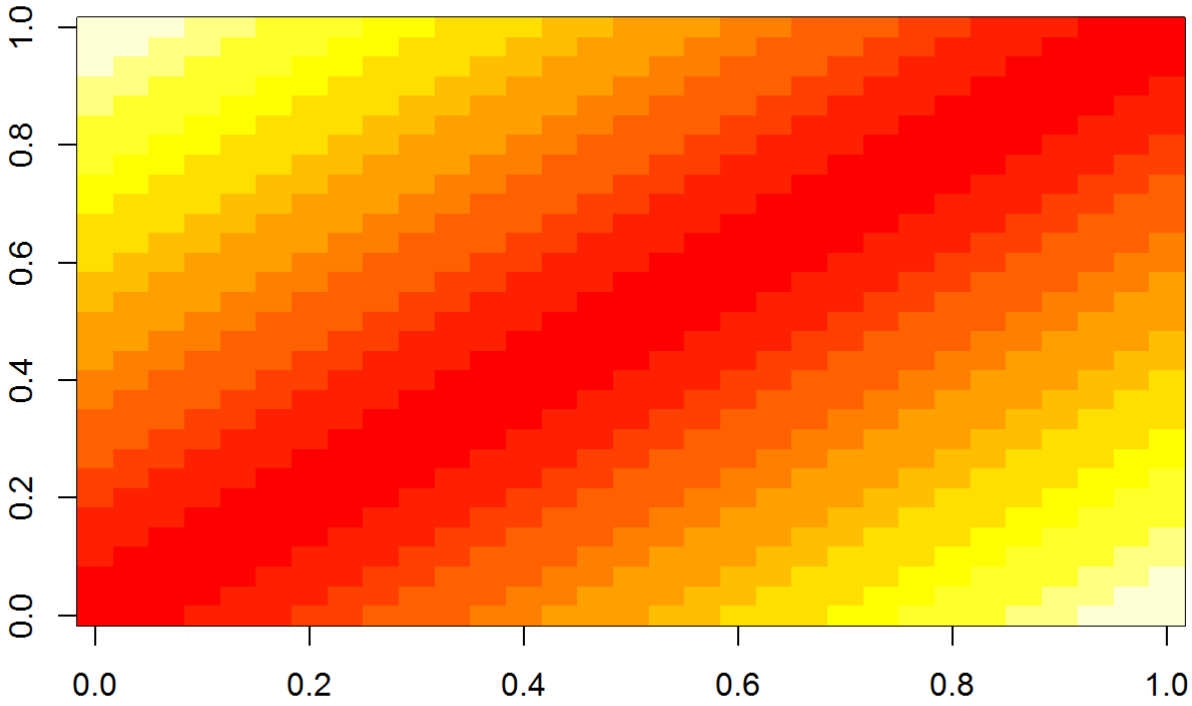
```
## The following object is masked from 'package:base':
##
##     union
```

```
library(rgl)
library(Matrix)
n1 <- 30
n2 <- 50
n3 <- 20
el.1 <- cbind(1:n1,2:(n1+1))
#el.2 <- cbind(1:n2,2:(n2+1)) + (n1+1)
#el.3 <- cbind(1:n3,2:(n3+1)) + (n1+n2+2)
#el1 <- rbind(el.1,el.2,el.3,c(n1/2,n1+2),c(n1/3,n1+n2+3))
el1 <- el.1
g1 <- graph.edgelist(el1,directed=FALSE)
# エッジ長
e.len1 <- rep(1,length(el1[,1]))
```

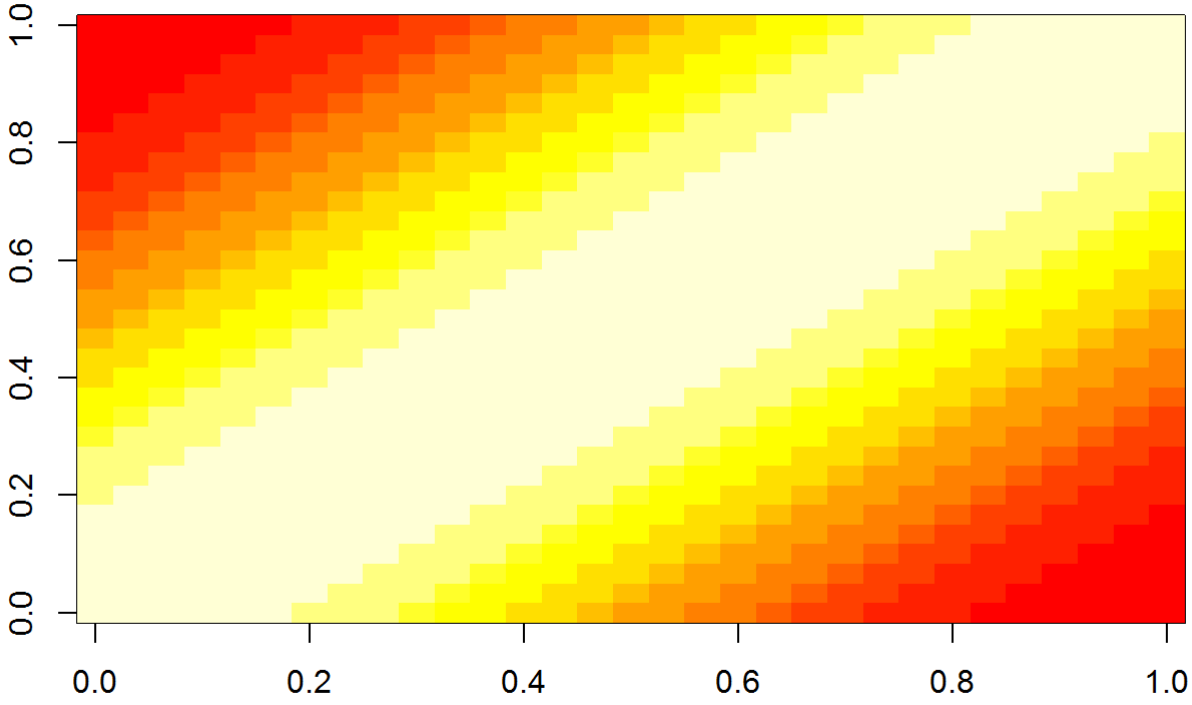
```
plot(g1,layout=layout_as_tree)
```



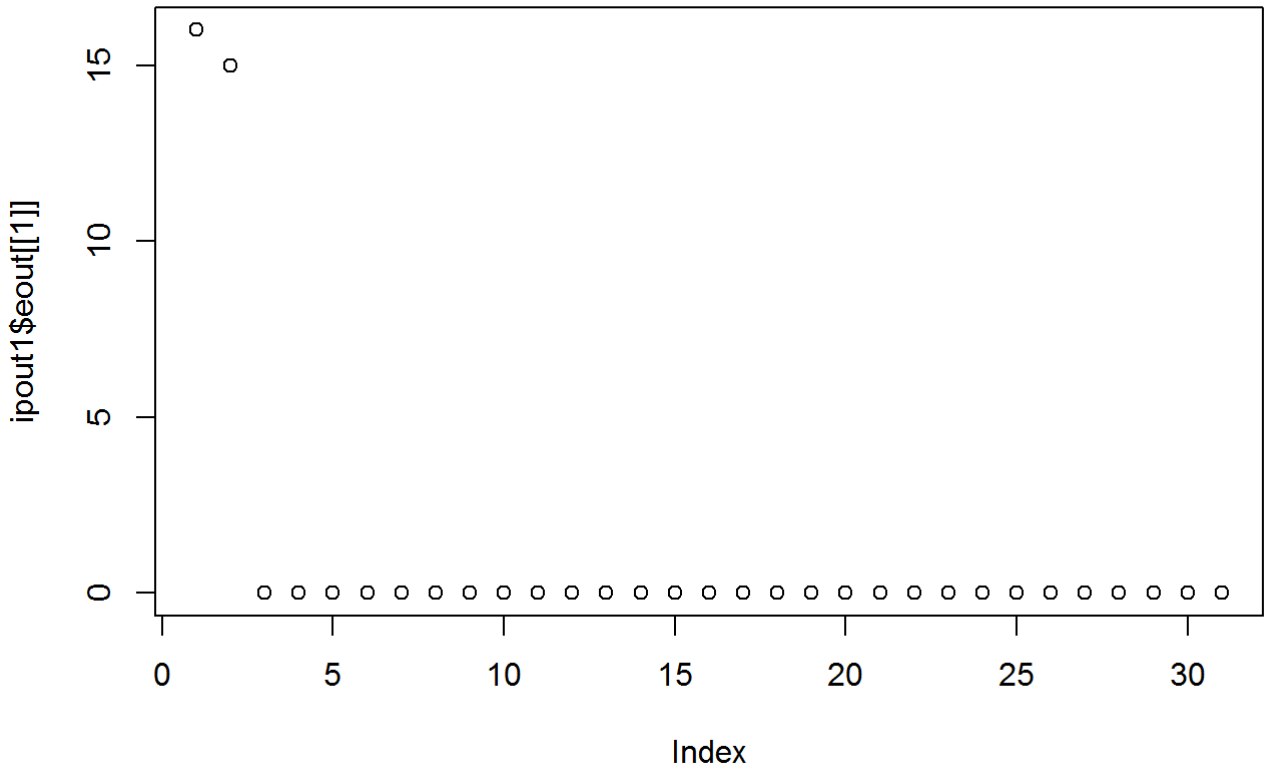

```
ipout1 <- my.IPcoords2(g1,e.len1)
# 距離行列
image(ipout1$D)
```




```
# IP行列
image(ipout1$P)
```



```
# 固有値
plot(ipout1$eout[[1]])
```

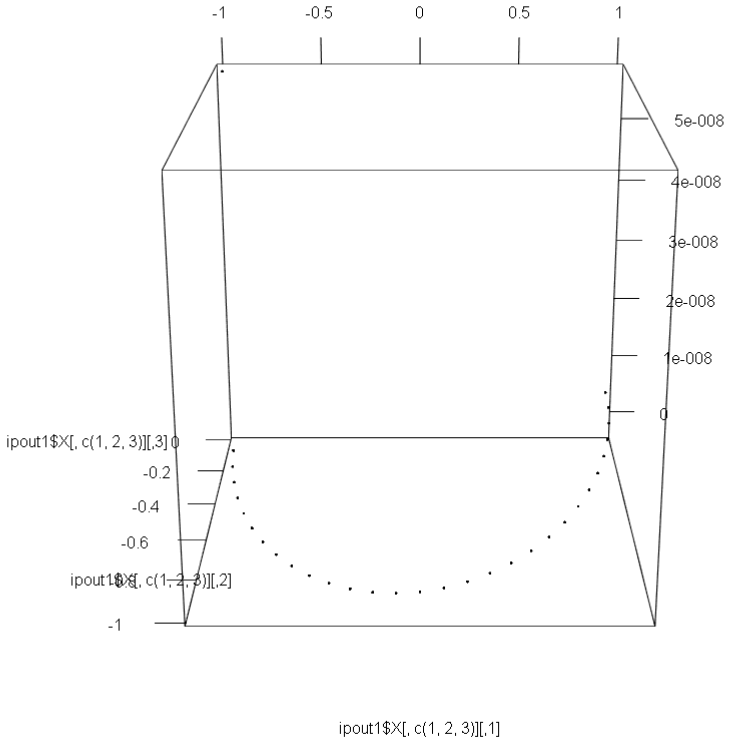


ランクは2

```
print(rankMatrix(ipout1$P))
```

```
## [1] 2
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 6.883383e-15
```

```
n1 <- length(V(g1))
plot3d(ipout1$X[,c(1,2,3)])
#spheres3d(ipout1$X[,c(1,2,n1)],radius=1)
```



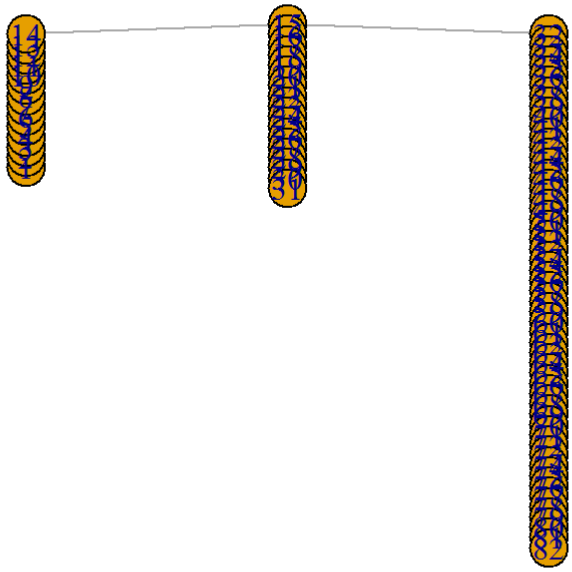
１分岐。Y字

ランクは４。

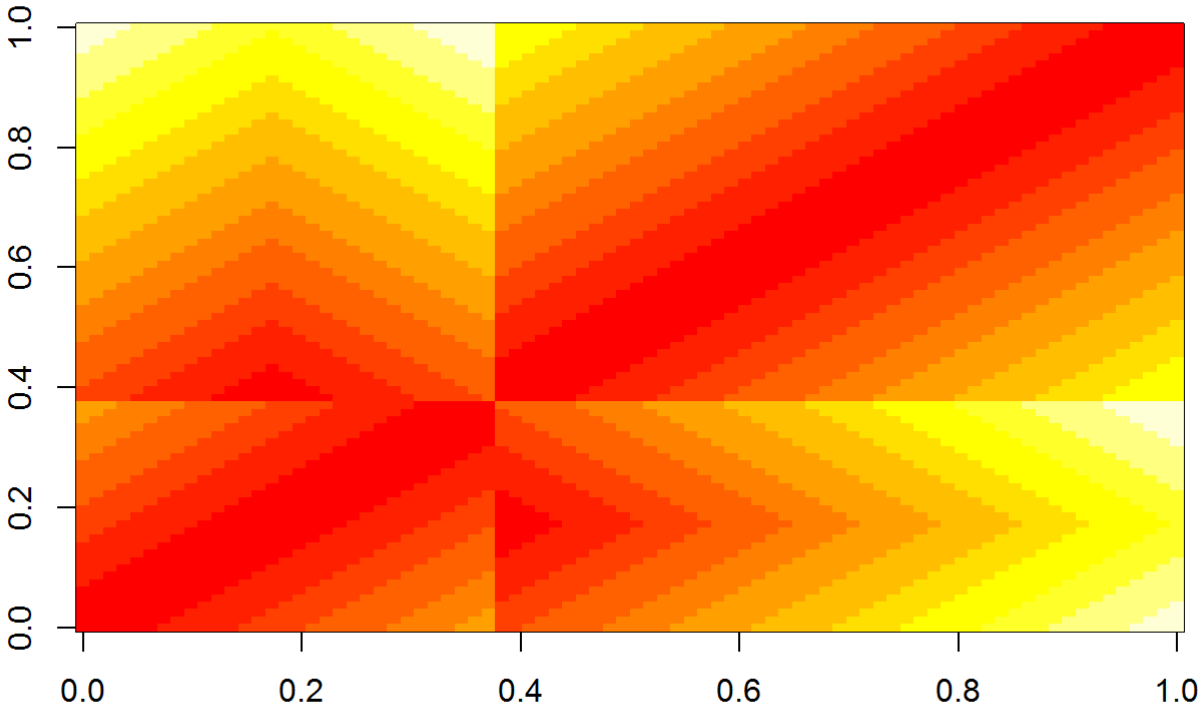
末端ノード(葉ノード)３個と分岐点１個の４個を決める。

```
n1 <- 30
n2 <- 50
n3 <- 20
el.1 <- cbind(1:n1,2:(n1+1))
el.2 <- cbind(1:n2,2:(n2+1)) + (n1+1)
#el.3 <- cbind(1:n3,2:(n3+1)) + (n1+n2+2)
el2 <- rbind(el.1,el.2,c(n1/2,n1+2))

g2 <- graph.edgelist(el2,directed=FALSE)
# エッジ長
e.len2 <- rep(1,length(el2[,1]))
```

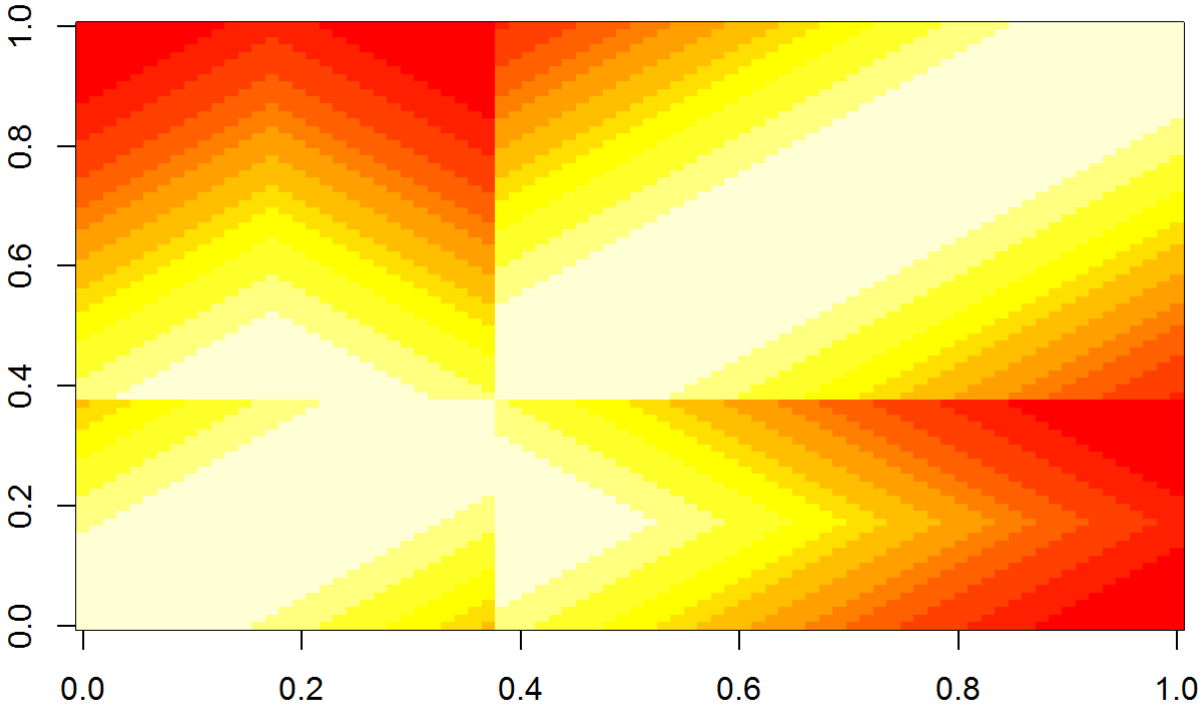
```
plot(g2,layout=layout_as_tree)
```



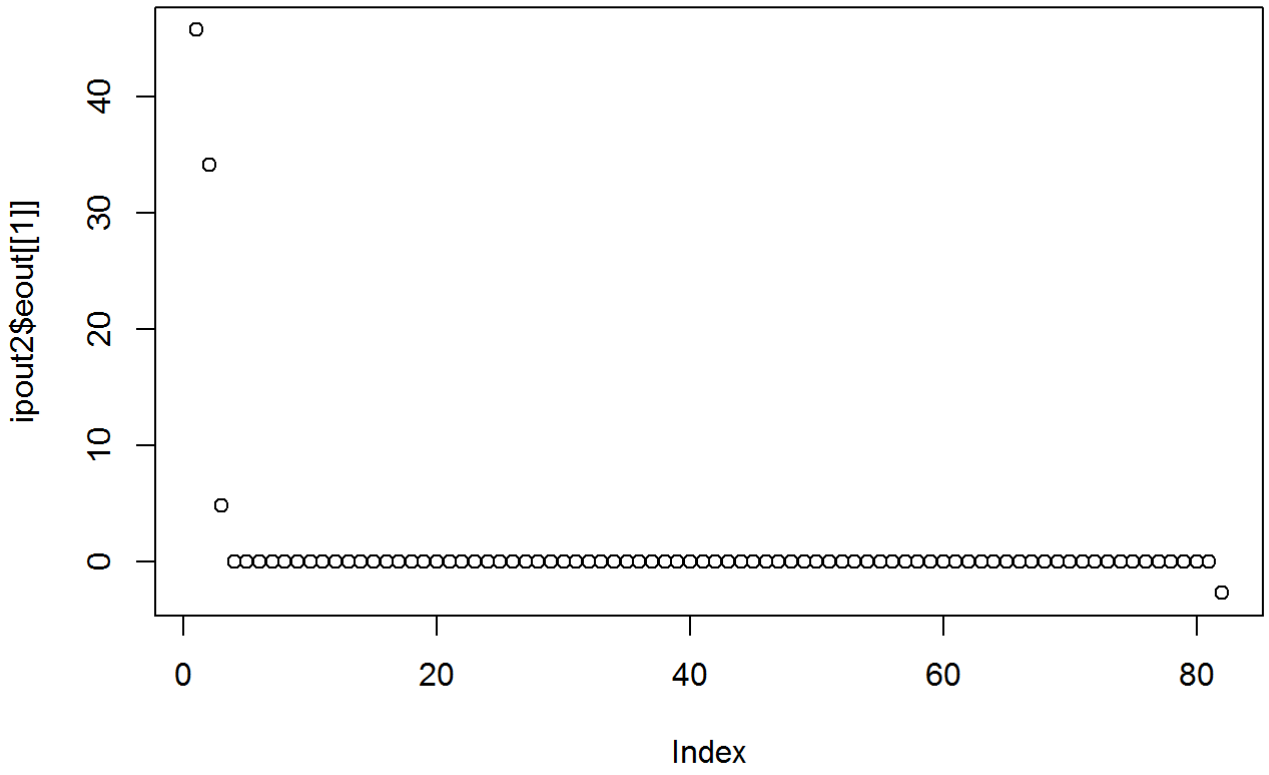
```
ipout2 <- my.IPcoords2(g2,e.len2)
# 距離行列
image(ipout2$D)
```




```
# IP行列
image(ipout2$P)
```



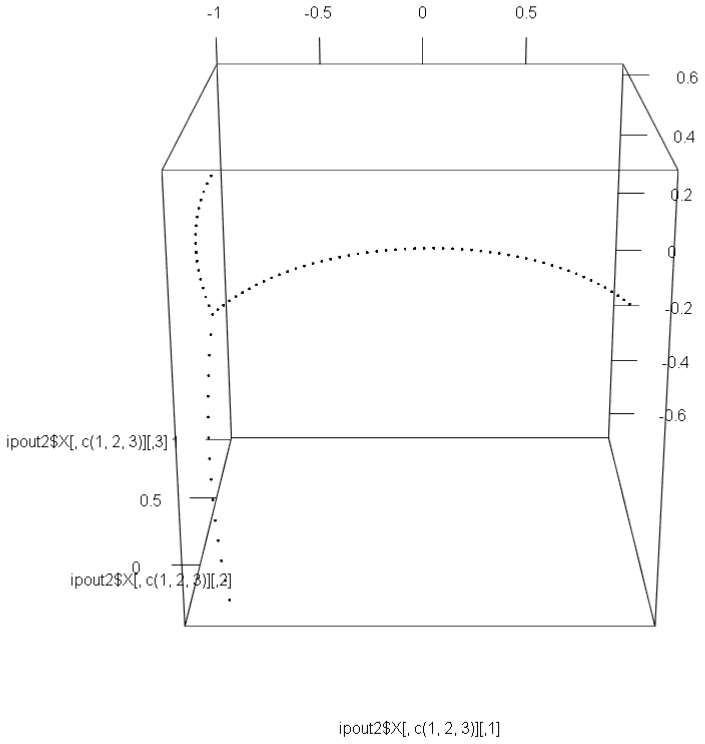
```
# 固有値
plot(ipout2$eout[[1]])
```




```
print(rankMatrix(ipout2$P))
```

```
## [1] 4
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 1.820766e-14
```

```
n2 <- length(V(g2))
plot3d(ipout2$X[,c(1,2,3)])
```



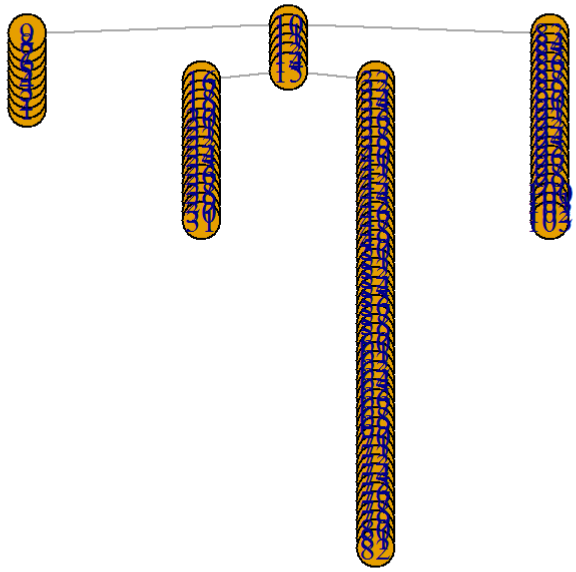
２分岐

ランクは6。

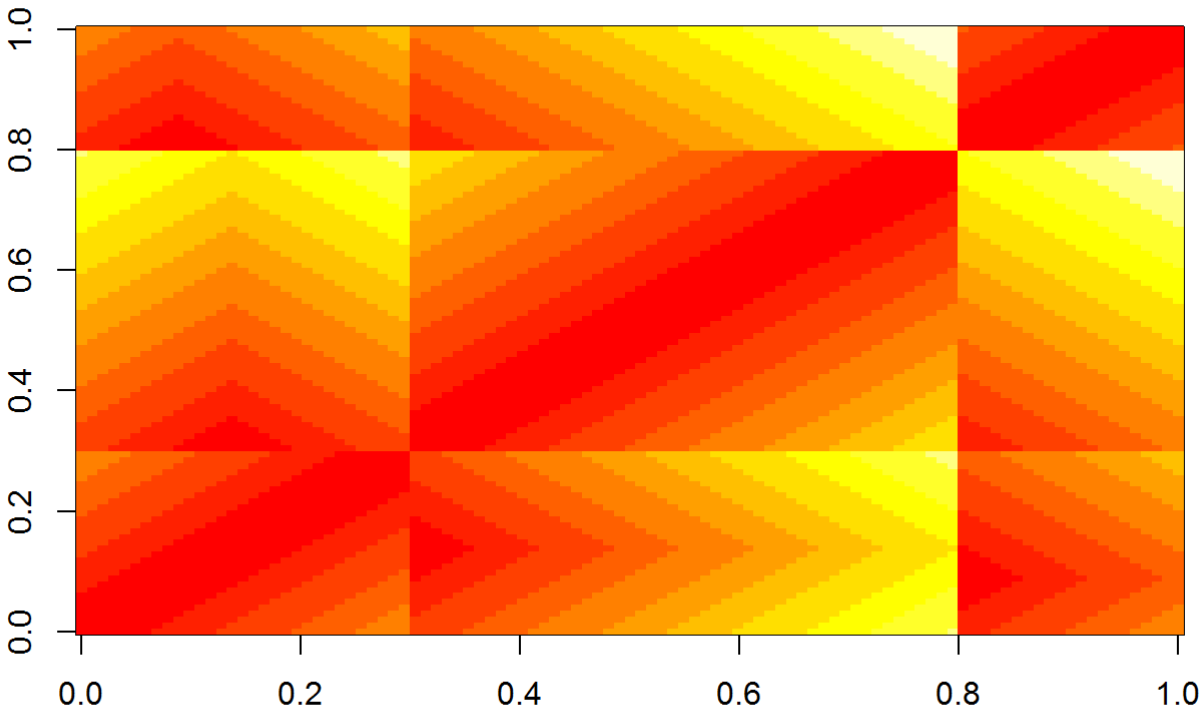
末端ノード(葉ノード)4個と分岐点１個の2個を決める。

```
n1 <- 30
n2 <- 50
n3 <- 20
el.1 <- cbind(1:n1,2:(n1+1))
el.2 <- cbind(1:n2,2:(n2+1)) + (n1+1)
el.3 <- cbind(1:n3,2:(n3+1)) + (n1+n2+2)
el3 <- rbind(el.1,el.2,el.3,c(n1/2,n1+2),c(n1/3,n1+n2+3))

g3 <- graph.edgelist(el3,directed=FALSE)
# エッジ長
e.len3 <- rep(1,length(el3[,1]))
```

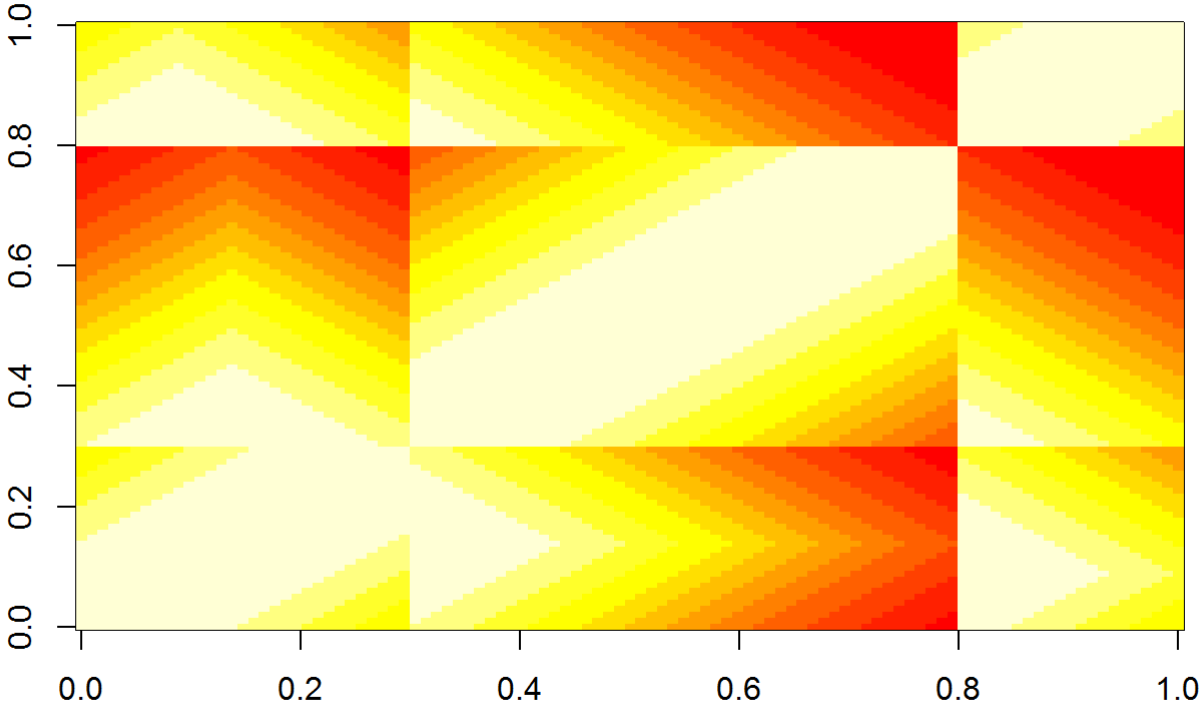
```
plot(g3,layout=layout_as_tree)
```



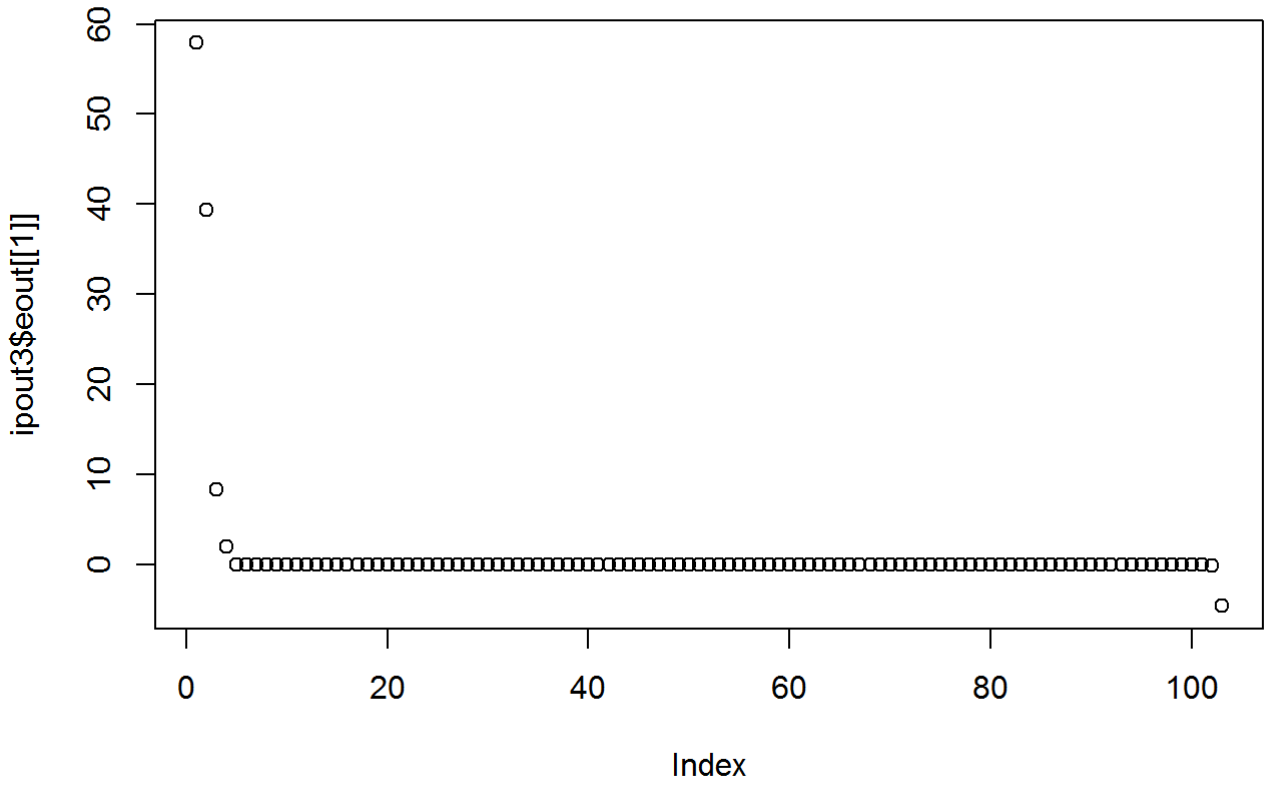
```
ipout3 <- my.IPcoords2(g3,e.len3)
# 距離行列
image(ipout3$D)
```




```
# IP行列
image(ipout3$P)
```



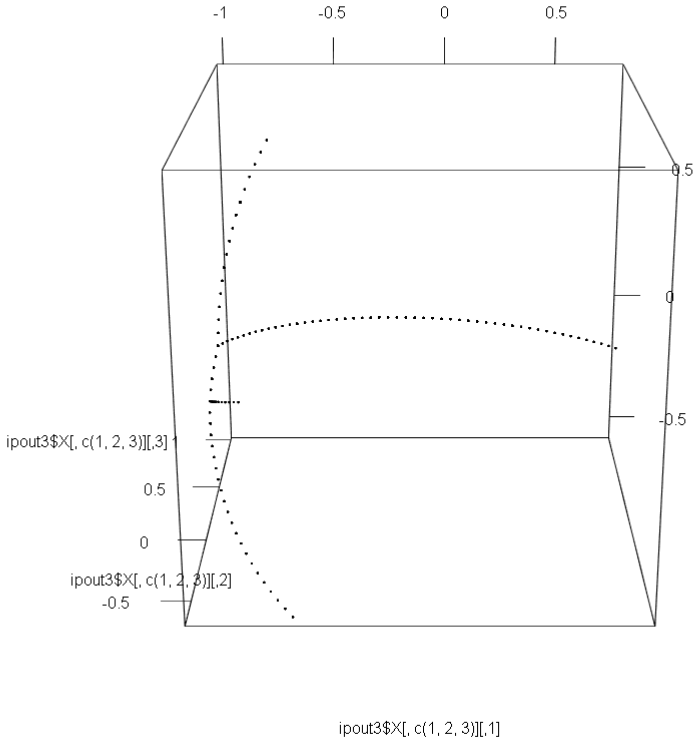
```
# 固有値
plot(ipout3$eout[[1]])
```




```
print(rankMatrix(ipout3$P))
```

```
## [1] 6
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 2.287059e-14
```

```
n3 <- length(V(g3))
plot3d(ipout3$X[,c(1,2,3)])
```

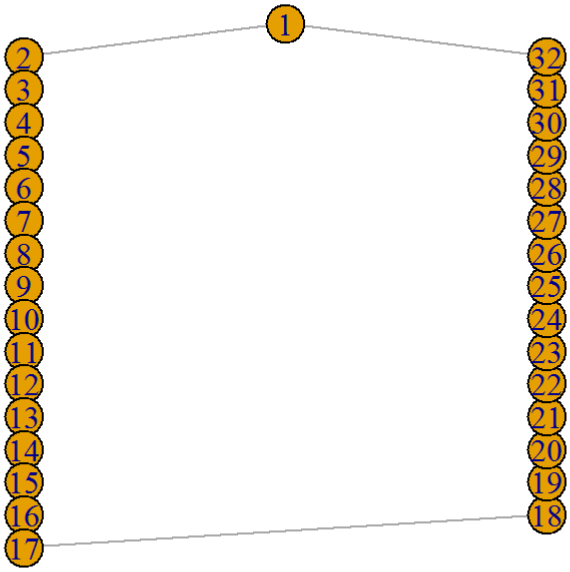


ループを入れる

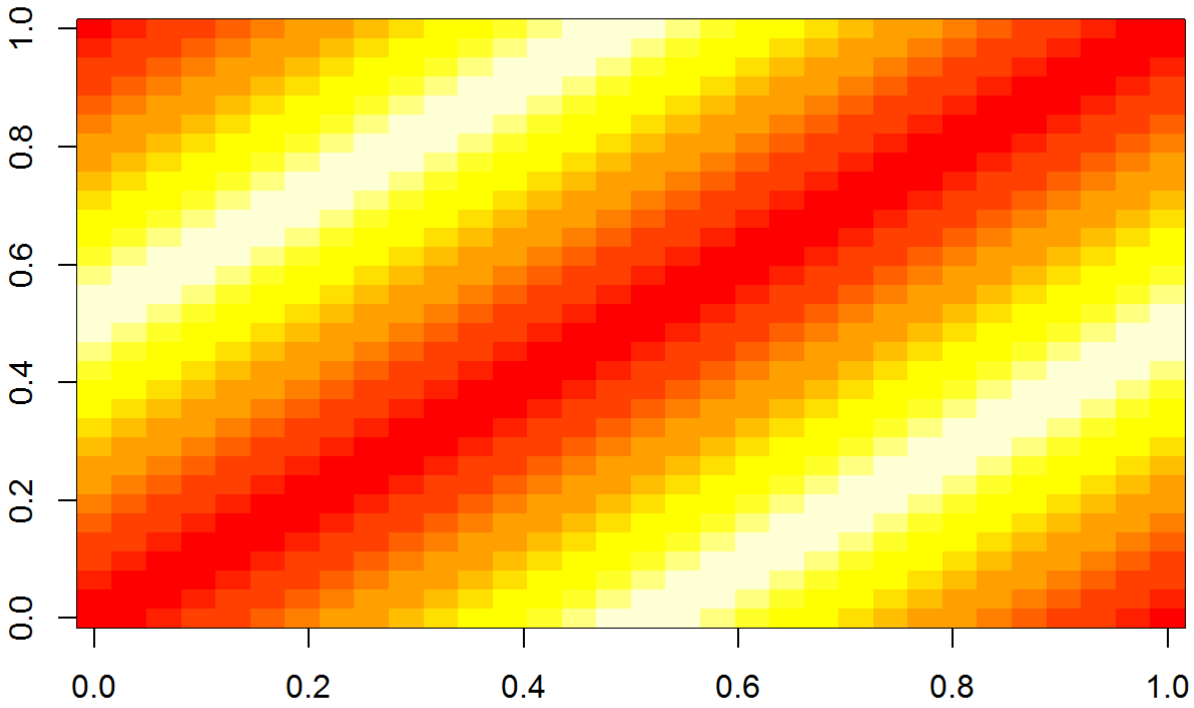
ノード数が偶数だと、ちょうど「最遠点」が決まるので、ランクが２になるようだ。

```
n1 <- 31
n2 <- 50
n3 <- 20

el.1 <- cbind(1:n1,2:(n1+1))
#el.2 <- cbind(1:n2,2:(n2+1)) + (n1+1)
#el.3 <- cbind(1:n3,2:(n3+1)) + (n1+n2+2)
#el1 <- rbind(el.1,el.2,el.3,c(n1/2,n1+2),c(n1/3,n1+n2+3))
el4 <- rbind(el.1,c(1,n1+1))

g4 <- graph.edgelist(el4,directed=FALSE)
# エッジ長
e.len4 <- rep(1,length(el4[,1]))
```

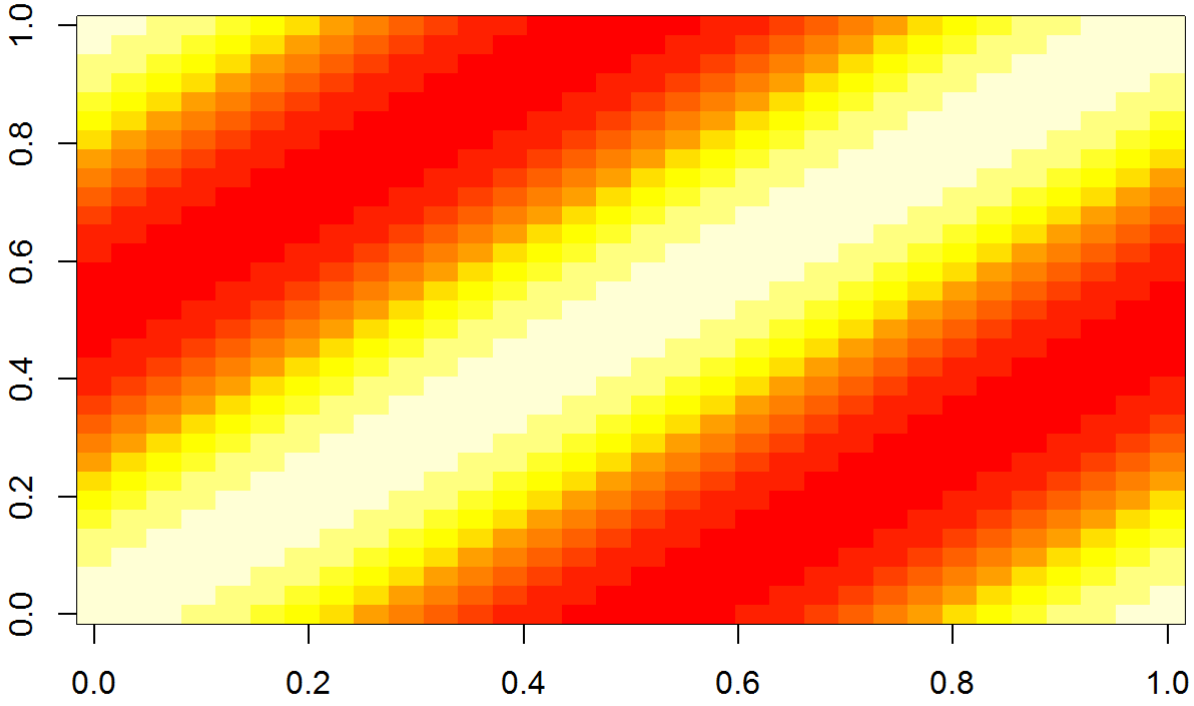
```
plot(g4,layout=layout_as_tree)
```



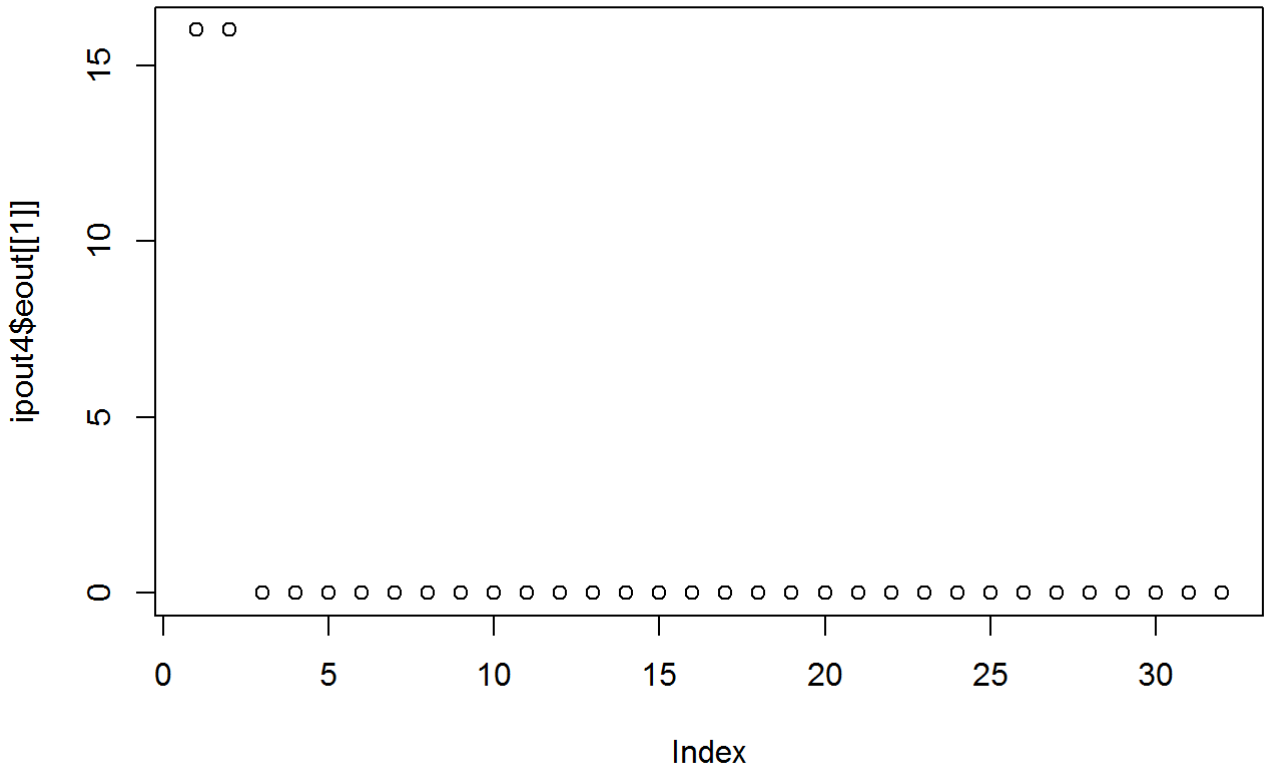

```
ipout4 <- my.IPcoords2(g4,e.len4)
# 距離行列
image(ipout4$D)
```




```
# IP行列
image(ipout4$P)
```



```
# 固有値
plot(ipout4$eout[[1]])
```

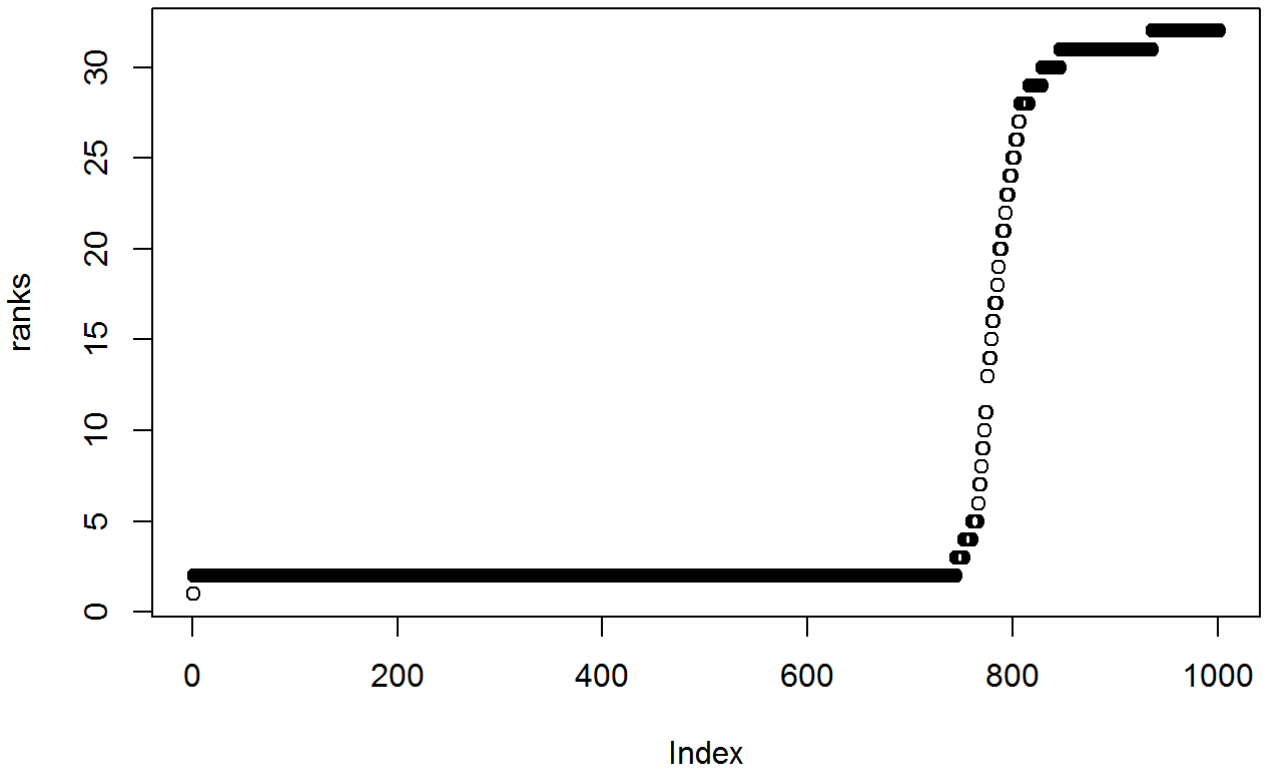



```
print(rankMatrix(ipout4$P))
```

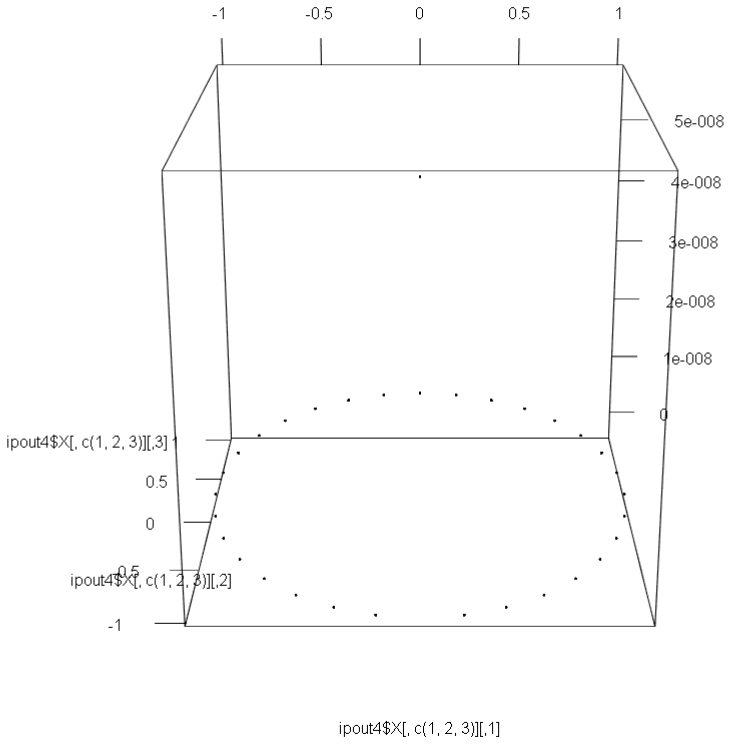
```
## [1] 2
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 7.105427e-15
```

ランク計算の「精度 (tol)」を変えて計算しておく。

```
tols <- 1.05^(0:(-1000))
ranks <- rep(0,length(tols))
for(i in 1:length(tols)){
  ranks[i] <- rankMatrix(ipout4$P,tol=tols[i])
}

plot(ranks)
```



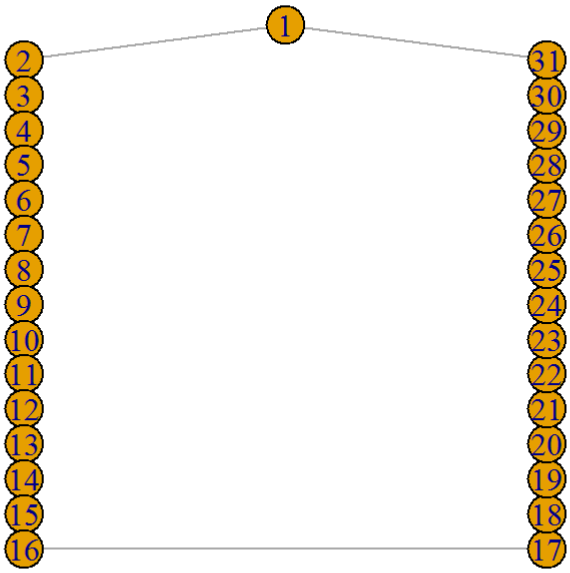

```
n4 <- length(V(g4))
plot3d(ipout4$X[,c(1,2,3)])
```



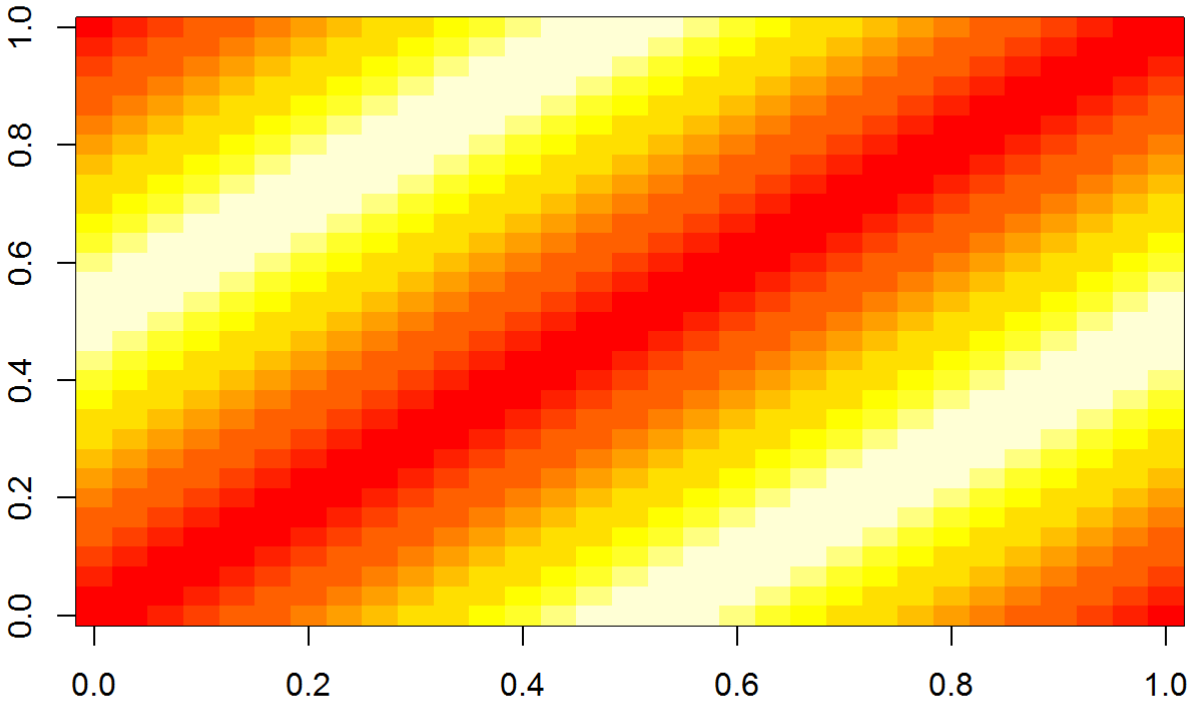
頂点数を奇数にする。

```
n1 <- 30
n2 <- 50
n3 <- 20

el.1 <- cbind(1:n1,2:(n1+1))
#el.2 <- cbind(1:n2,2:(n2+1)) + (n1+1)
#el.3 <- cbind(1:n3,2:(n3+1)) + (n1+n2+2)
#el1 <- rbind(el.1,el.2,el.3,c(n1/2,n1+2),c(n1/3,n1+n2+3))
el4 <- rbind(el.1,c(1,n1+1))

g4 <- graph.edgelist(el4,directed=FALSE)
# エッジ長
e.len4 <- rep(1,length(el4[,1]))
```

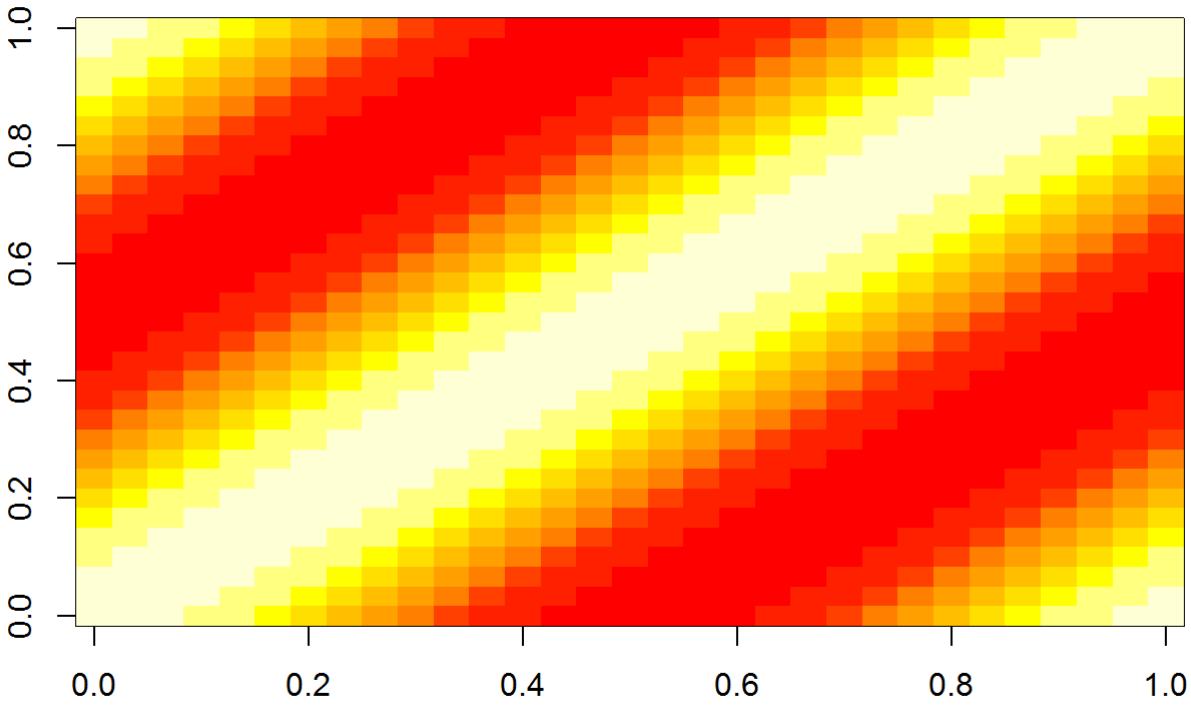
```
plot(g4,layout=layout_as_tree)
```



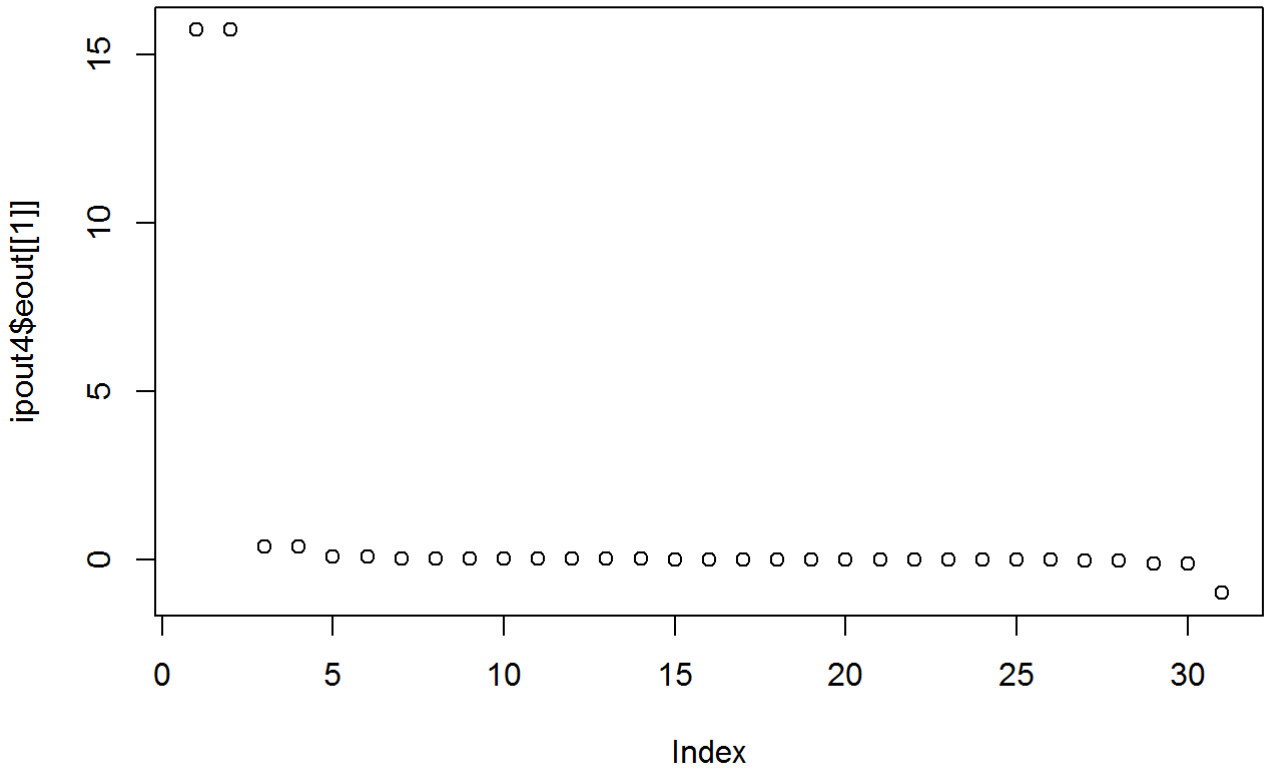

```
ipout4 <- my.IPcoords2(g4,e.len4)
# 距離行列
image(ipout4$D)
```




```
# IP行列
image(ipout4$P)
```



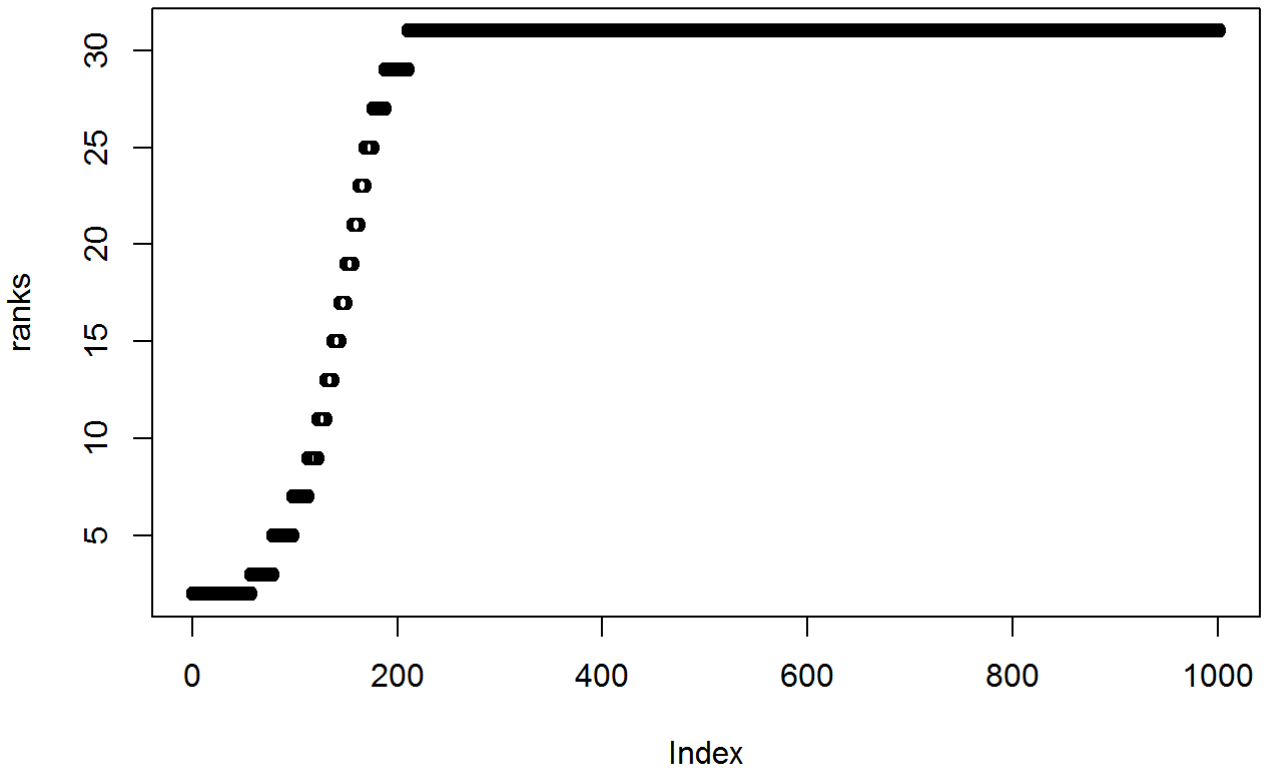
```
# 固有値
plot(ipout4$eout[[1]])
```



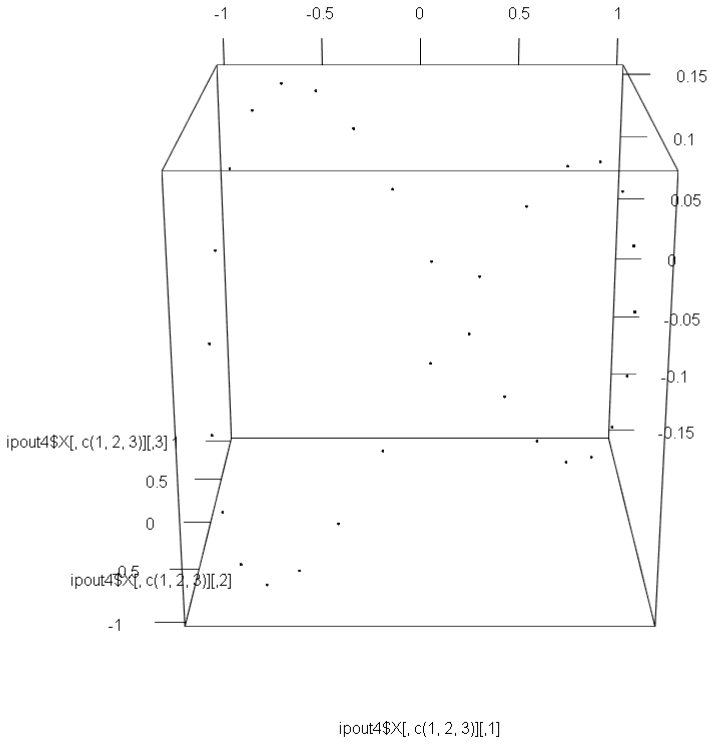

```
print(rankMatrix(ipout4$P))
```

```
## [1] 31
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 6.883383e-15
```

```
tols <- 1.05^(0:(-1000))
ranks <- rep(0,length(tols))
for(i in 1:length(tols)){
  ranks[i] <- rankMatrix(ipout4$P,tol=tols[i])
}

plot(ranks)
```




```
n4 <- length(V(g4))
plot3d(ipout4$X[,c(1,2,3)])
```



ループを２つ

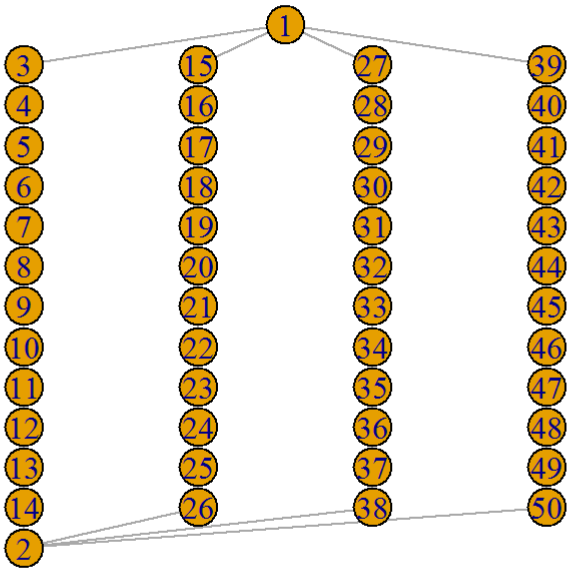
北極と南極を通る経線(0度、90度、180度、270度)のようなもの。

ランクは５。

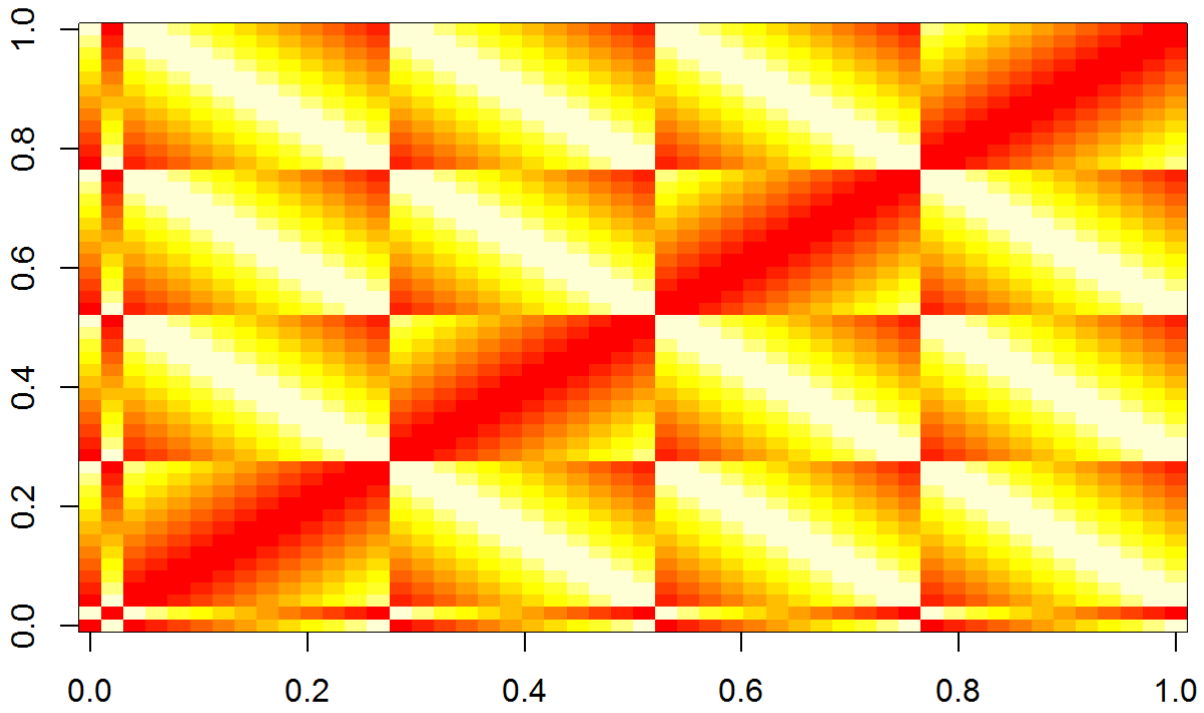
4本の子午線方向と、その起点で５つ？

```
n1 <- 11
el.1 <- el.2 <- el.3 <- el.4 <- cbind(1:n1,2:(n1+1))
el.1 <- el.1 + 2
el.2 <- el.2 + max(el.1)
el.3 <- el.3 + max(el.2)
el.4 <- el.4 + max(el.3)

el4 <- rbind(el.1,el.2,el.3,el.4)
el4 <- rbind(el4, c(1,min(el.1)),c(1,min(el.2)),c(1,min(el.3)),c(1,min(el.4)))

el4 <- rbind(el4, c(2,max(el.1)),c(2,max(el.2)),c(2,max(el.3)),c(2,max(el.4)))

g4 <- graph.edgelist(el4,directed=FALSE)
# エッジ長
e.len4 <- rep(1,length(el4[,1]))
```

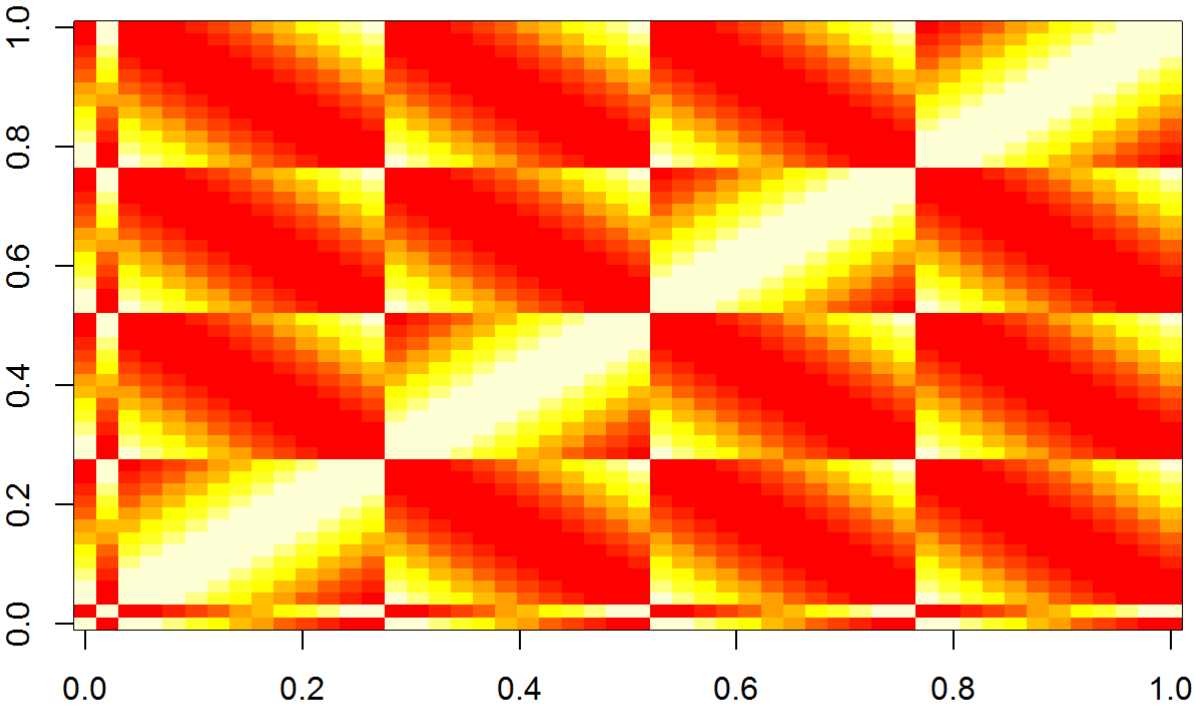
```
plot(g4,layout=layout_as_tree)
```



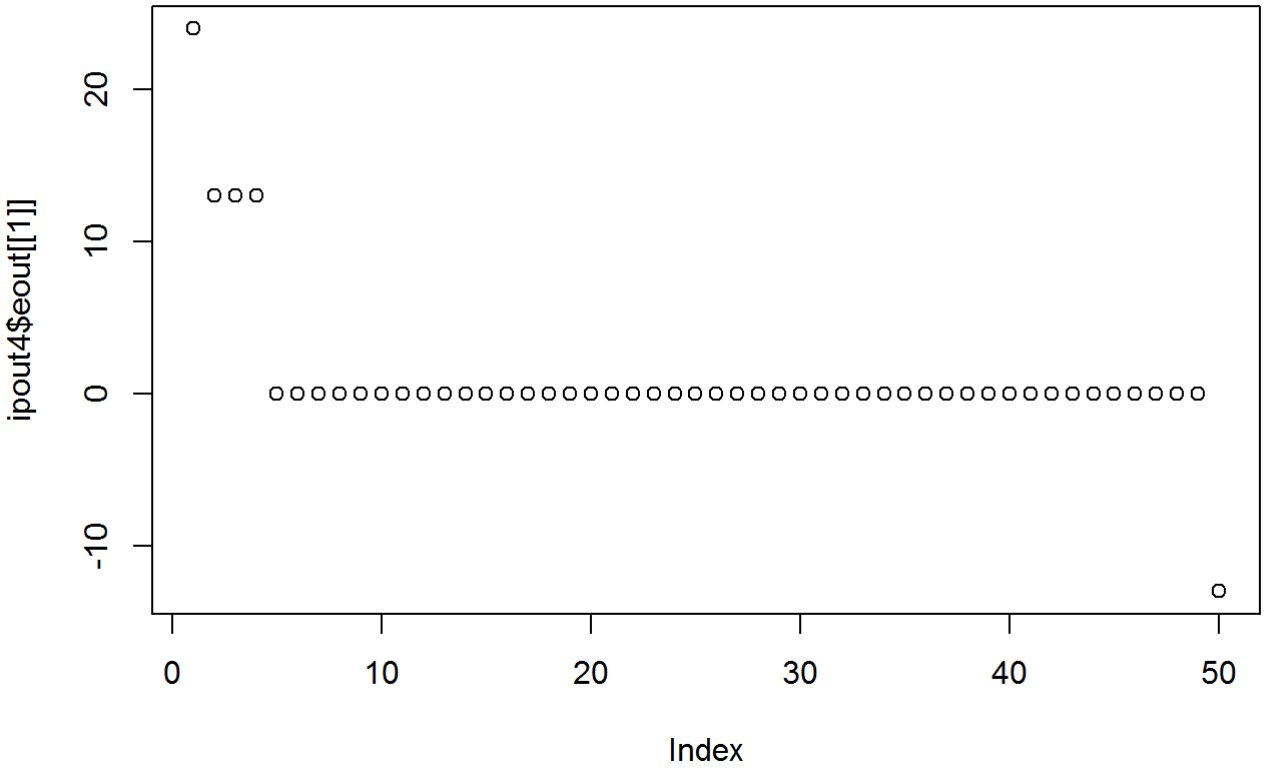
```
ipout4 <- my.IPcoords2(g4,e.len4)
# 距離行列
image(ipout4$D)
```




```
# IP行列
image(ipout4$P)
```



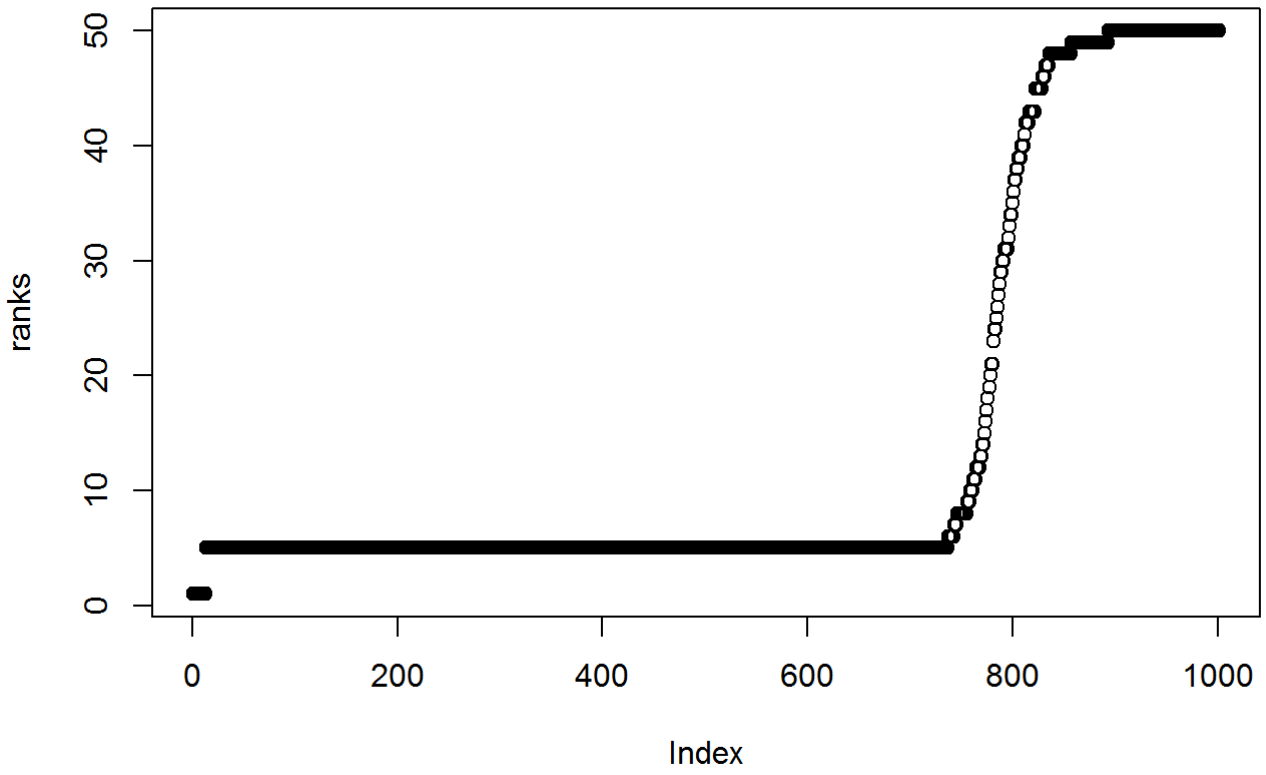

```
# 固有値
plot(ipout4$eout[[1]])
```



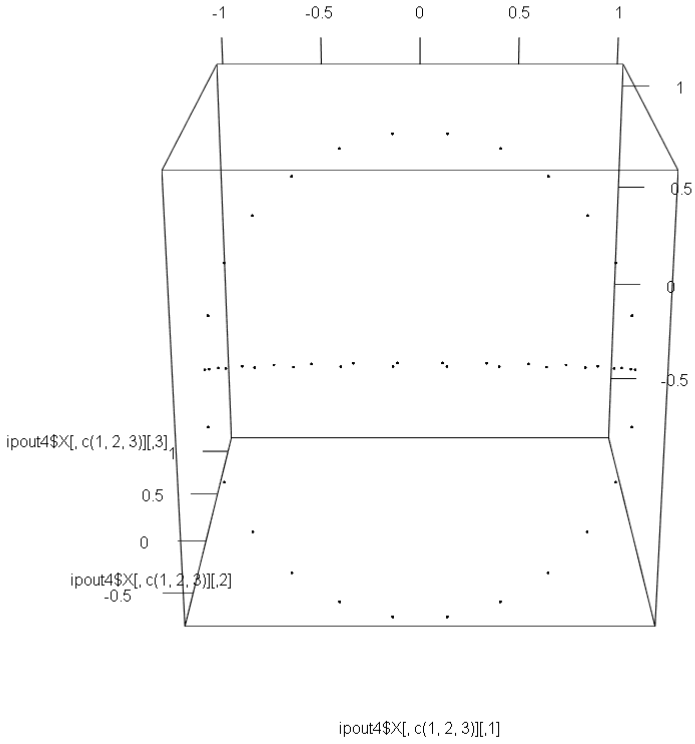
```
print(rankMatrix(ipout4$P))
```

```
## [1] 5
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 1.110223e-14
```

```
tols <- 1.05^(0:(-1000))
ranks <- rep(0,length(tols))
for(i in 1:length(tols)){
  ranks[i] <- rankMatrix(ipout4$P,tol=tols[i])
}

plot(ranks)
```




```
n4 <- length(V(g4))
plot3d(ipout4$X[,c(1,2,3)])
```



本格的に球面を

ランクは３らしい。

```
library(devtools)
```

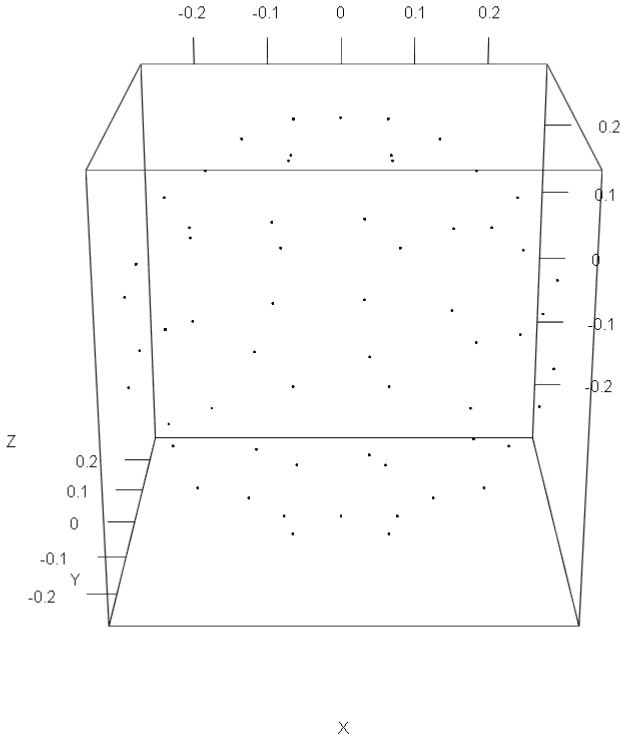
```
## Warning: package 'devtools' was built under R version 3.4.3
```

```
# install_github("ryamada22/Ronlyryamada") 初回はインストールする
library(Ronlyryamada)
library(RFOC)
```

```
## Warning: package 'RFOC' was built under R version 3.4.4
```

```
n <- 1 # メッシュの複雑さを指定(大きいと凹凸の周期が細かくなる)
k <- 1 # メッシュの複雑さを指定(大きいと真球に近くなる)
n.mesh <- 8 # メッシュの細かさを指定
A. <- matrix(runif(n^2), n, n)
A.[1, 1] <- k
A. <- A. + rnorm(n^2, 0, 0.05)
xxx <- my.spherical.harm.mesh(A = A., n = n.mesh)
```

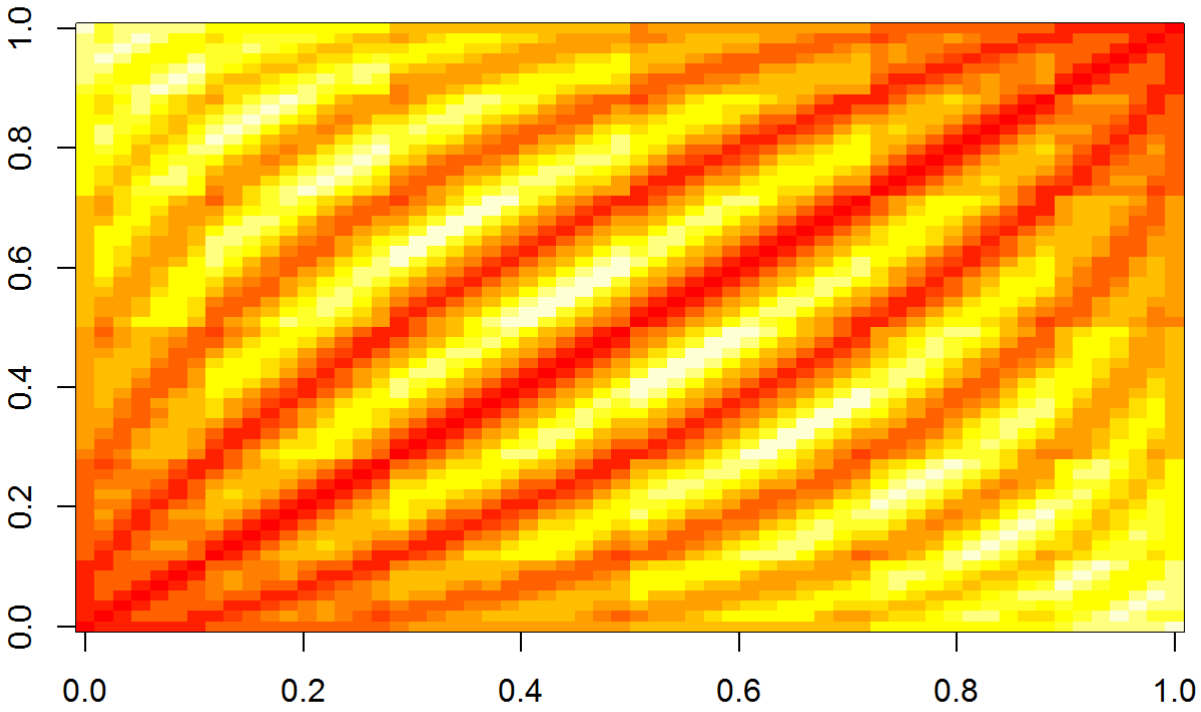
```
X <- xxx[[1]]
plot3d(X)
```



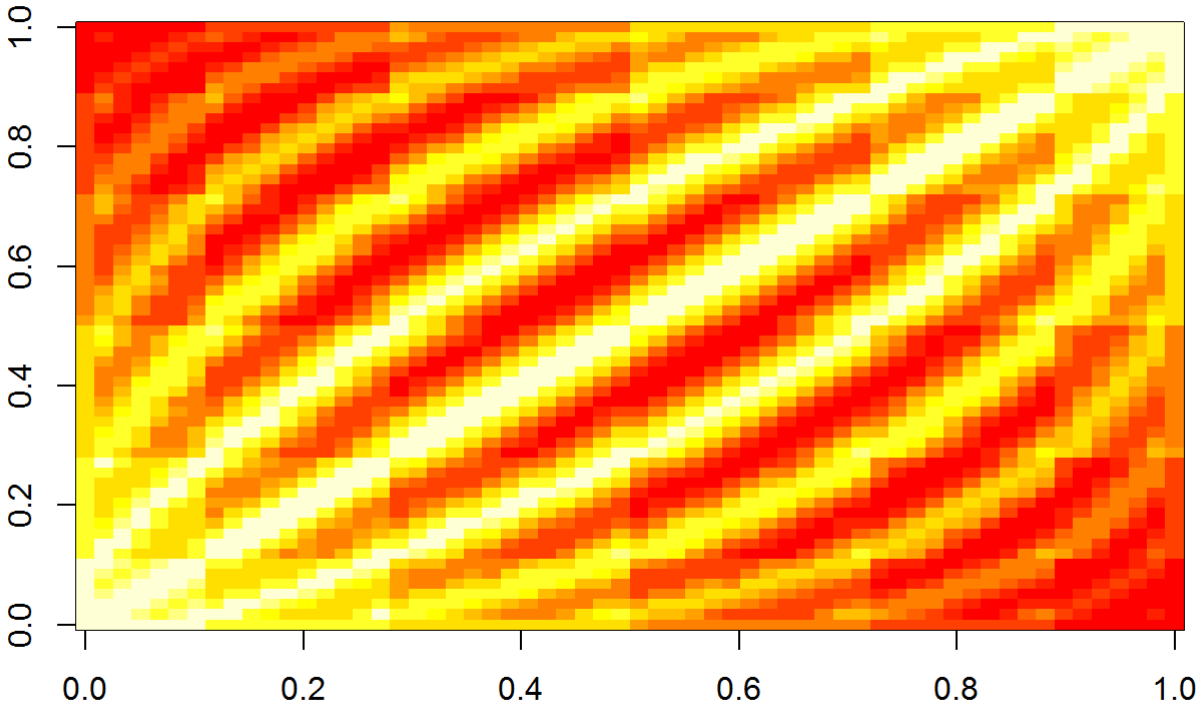

```
g5 <- graph.edgelist(xxx$edge,directed=FALSE)
#plot(g)

e.len5 <- rep(0,length(xxx$edge[,1]))
for(i in 1:length(e.len5)){
    e.len5[i] <- sqrt(sum((X[xxx$edge[i,1],]-X[xxx$edge[i,2],])^2))
}
```

```
ipout5 <- my.IPcoords2(g5,e.len5)
```

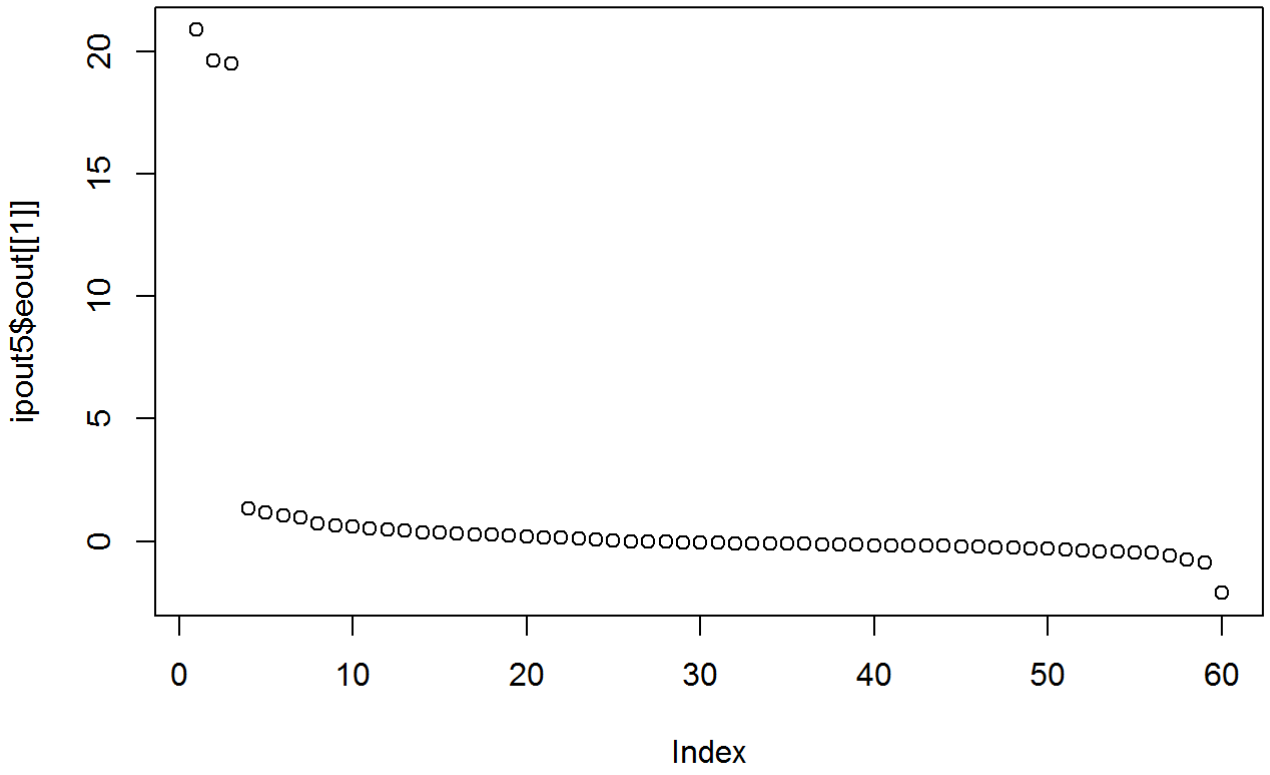
```
# 距離行列
image(ipout5$D)
```




```
# IP行列
image(ipout5$P)
```



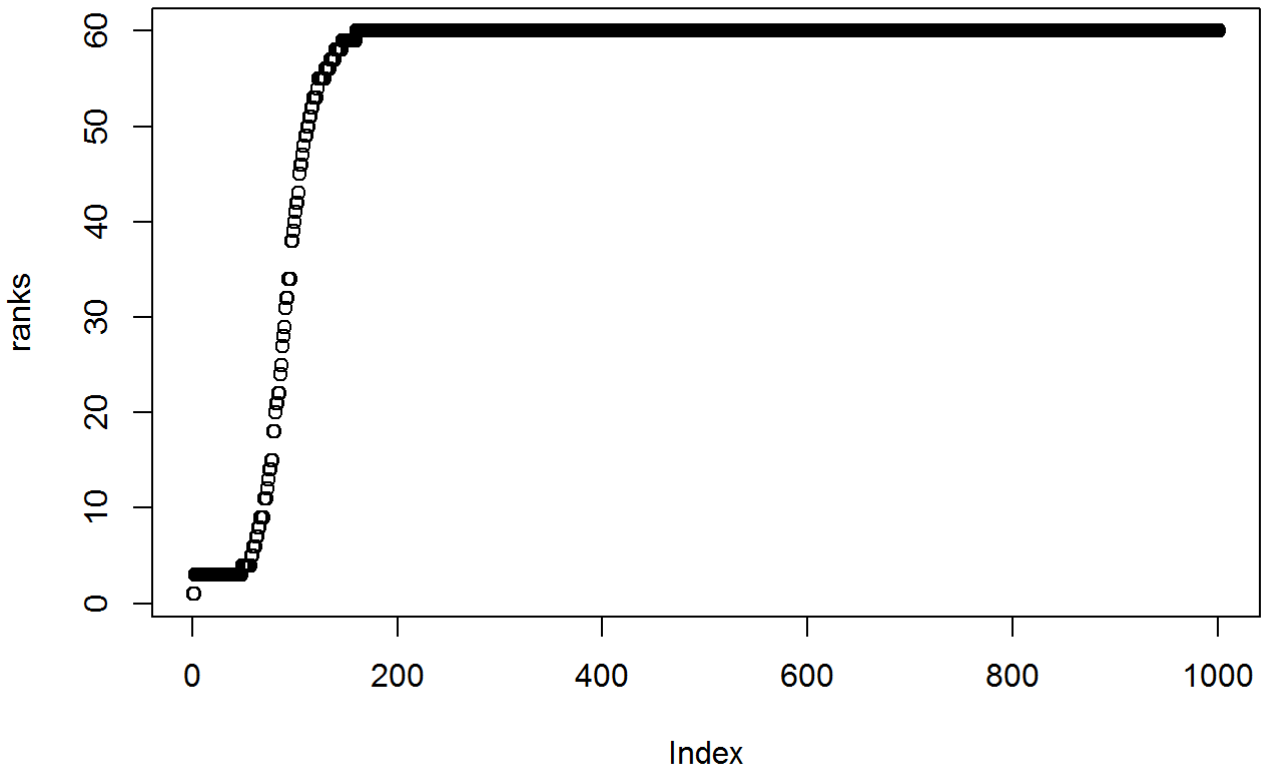
```
# 固有値
plot(ipout5$eout[[1]])
```




```
print(rankMatrix(ipout5$P))
```

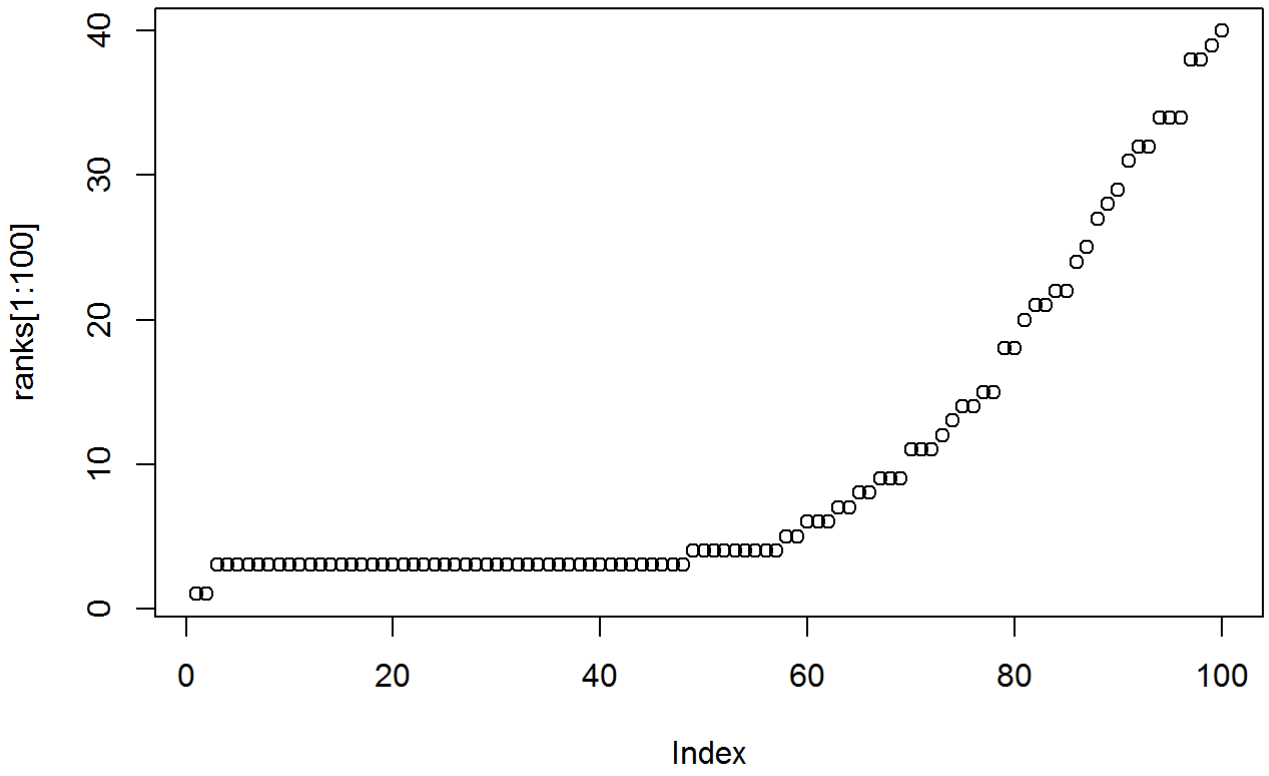
```
## [1] 60
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 1.332268e-14
```

```
tols <- 1.05^(0:(-1000))
ranks <- rep(0,length(tols))
for(i in 1:length(tols)){
  ranks[i] <- rankMatrix(ipout5$P,tol=tols[i])
}

plot(ranks)
```

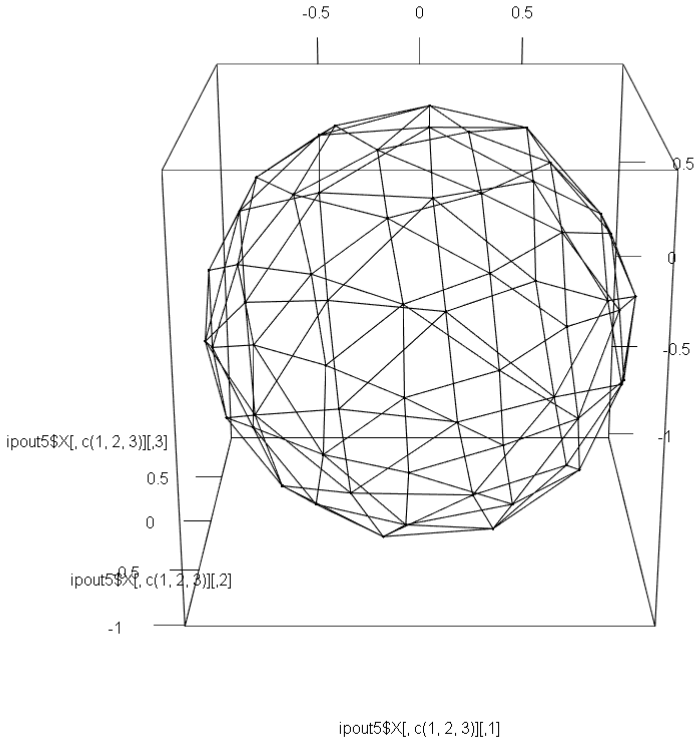


実質ランクは３なのでは？

```
plot(ranks[1:100])
```




```
npt <- length(ipout5$eout[[1]])
plot3d(ipout5$X[,c(1,2,3)])
#spheres3d(ipout5$X[,c(1,npt-1,npt)],radius=0.05)


el5 <- get.edgelist(g5)
segments3d(ipout5$X[t(el5),c(1,2,3)])
```

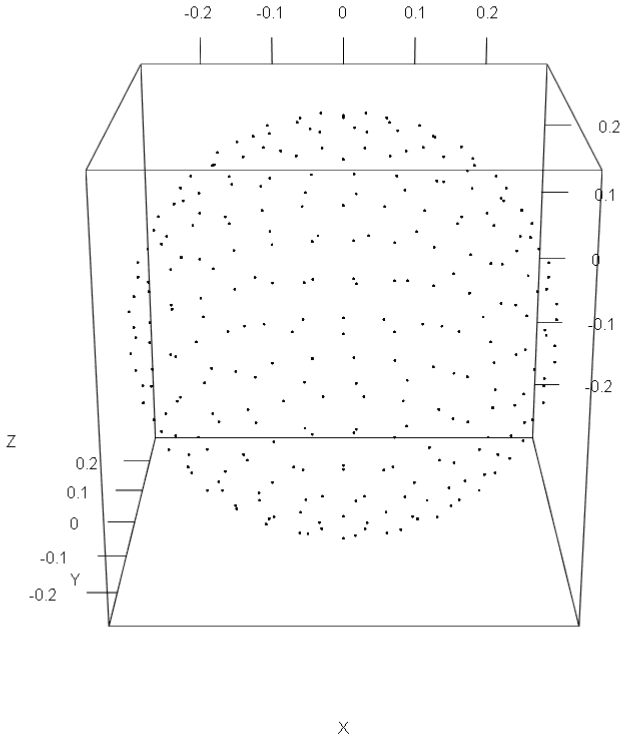


きれいな球面。メッシュを細かくしてみる

ランクは３

```
library(devtools)
# install_github("ryamada22/Ronlyryamada") 初回はインストールする
library(Ronlyryamada)
library(RFOC)

n <- 1 # メッシュの複雑さを指定(大きいと凹凸の周期が細かくなる)
k <- 1 # メッシュの複雑さを指定(大きいと真球に近くなる)
n.mesh <- 16 # メッシュの細かさを指定
A. <- matrix(runif(n^2), n, n)
A.[1, 1] <- k
A. <- A. + rnorm(n^2, 0, 0.05)
xxx <- my.spherical.harm.mesh(A = A., n = n.mesh)
```

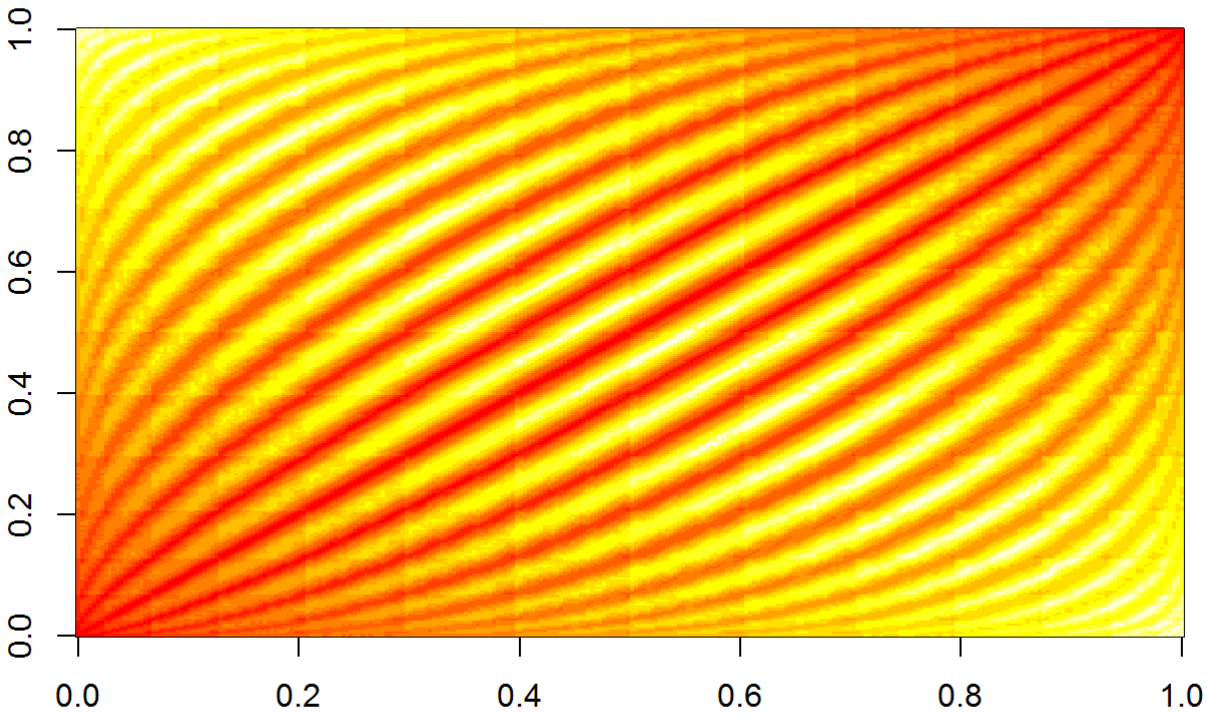
```
X <- xxx[[1]]
plot3d(X)
```



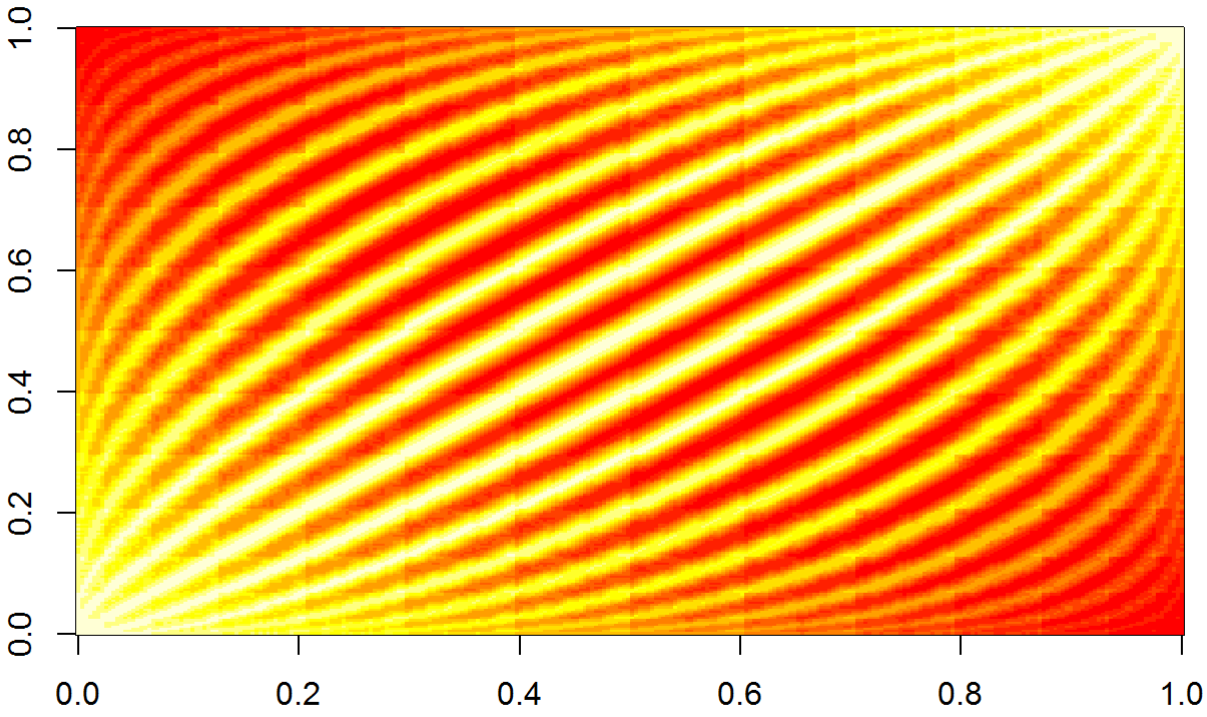

```
g6 <- graph.edgelist(xxx$edge,directed=FALSE)
#plot(g)

e.len6 <- rep(0,length(xxx$edge[,1]))
for(i in 1:length(e.len6)){
    e.len6[i] <- sqrt(sum((X[xxx$edge[i,1],]-X[xxx$edge[i,2],])^2))
}
```

```
ipout6 <- my.IPcoords2(g6,e.len6)
```

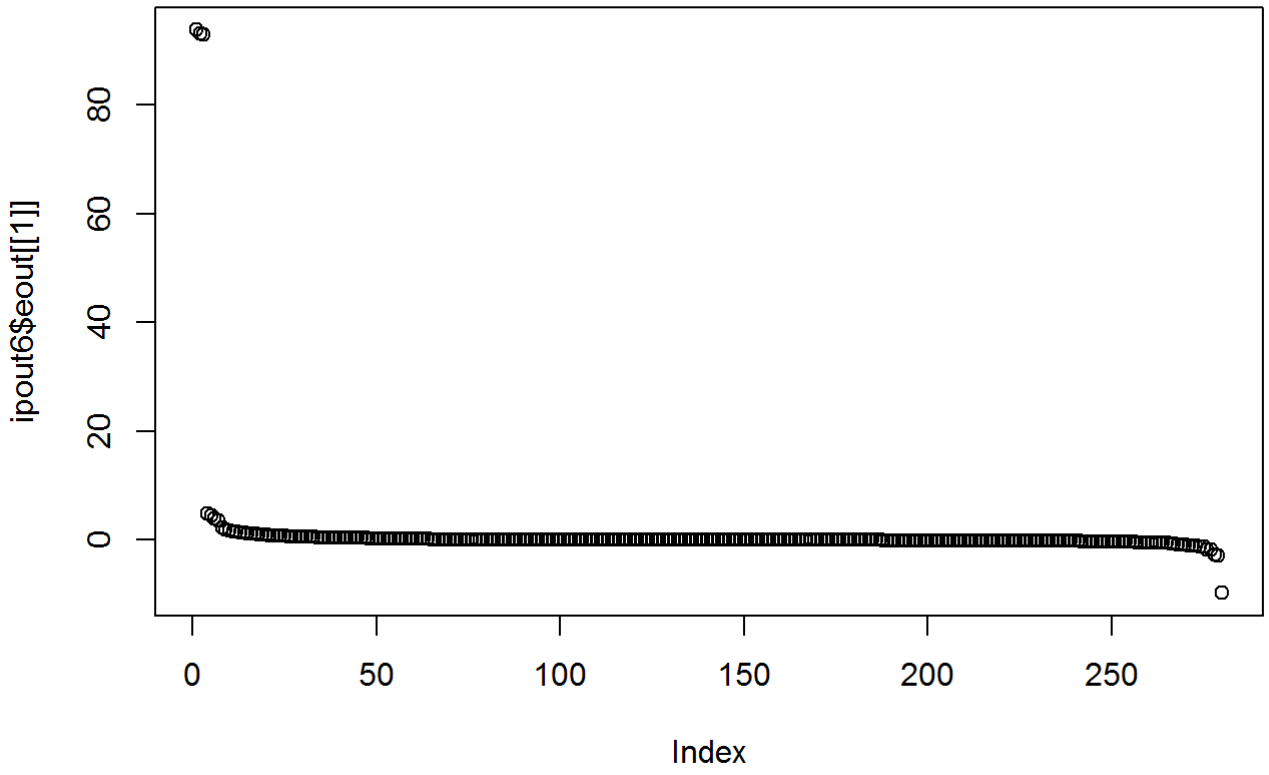
```
# 距離行列
image(ipout6$D)
```




```
# IP行列
image(ipout6$P)
```



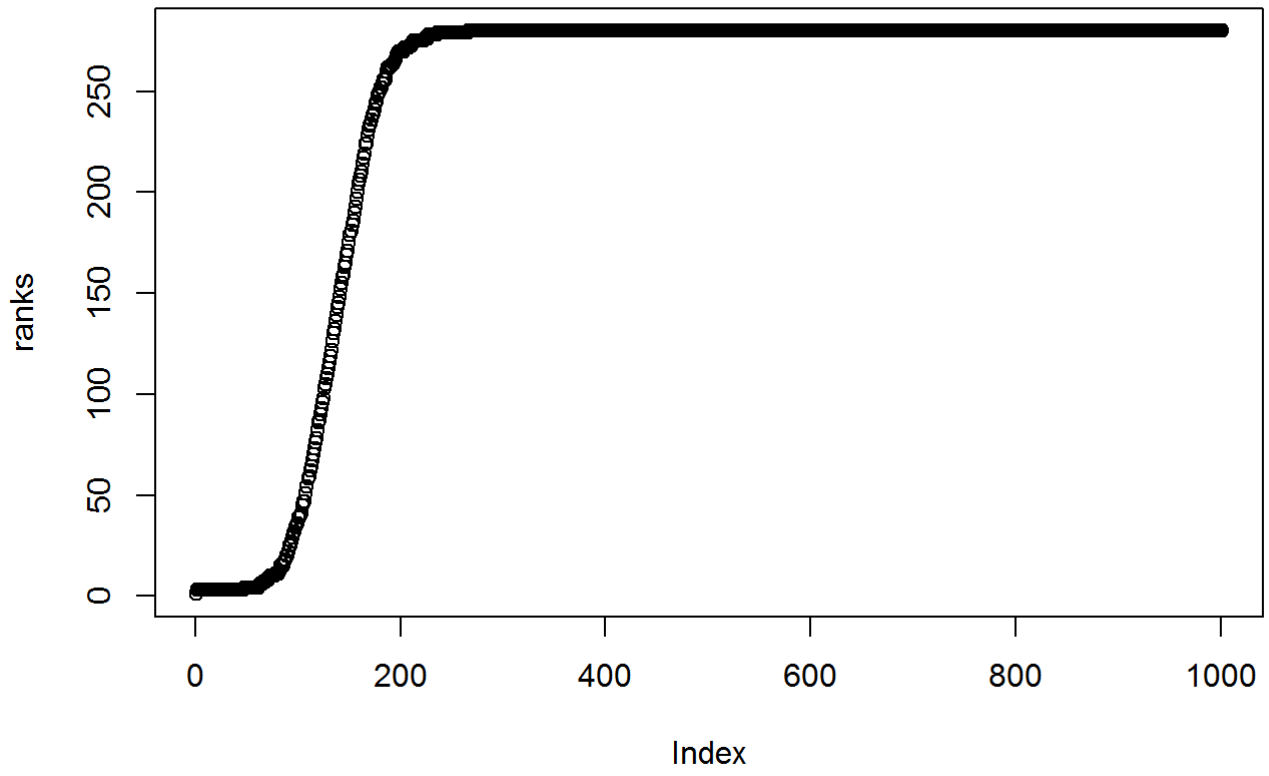
```
# 固有値
plot(ipout6$eout[[1]])
```




```
print(rankMatrix(ipout6$P))
```

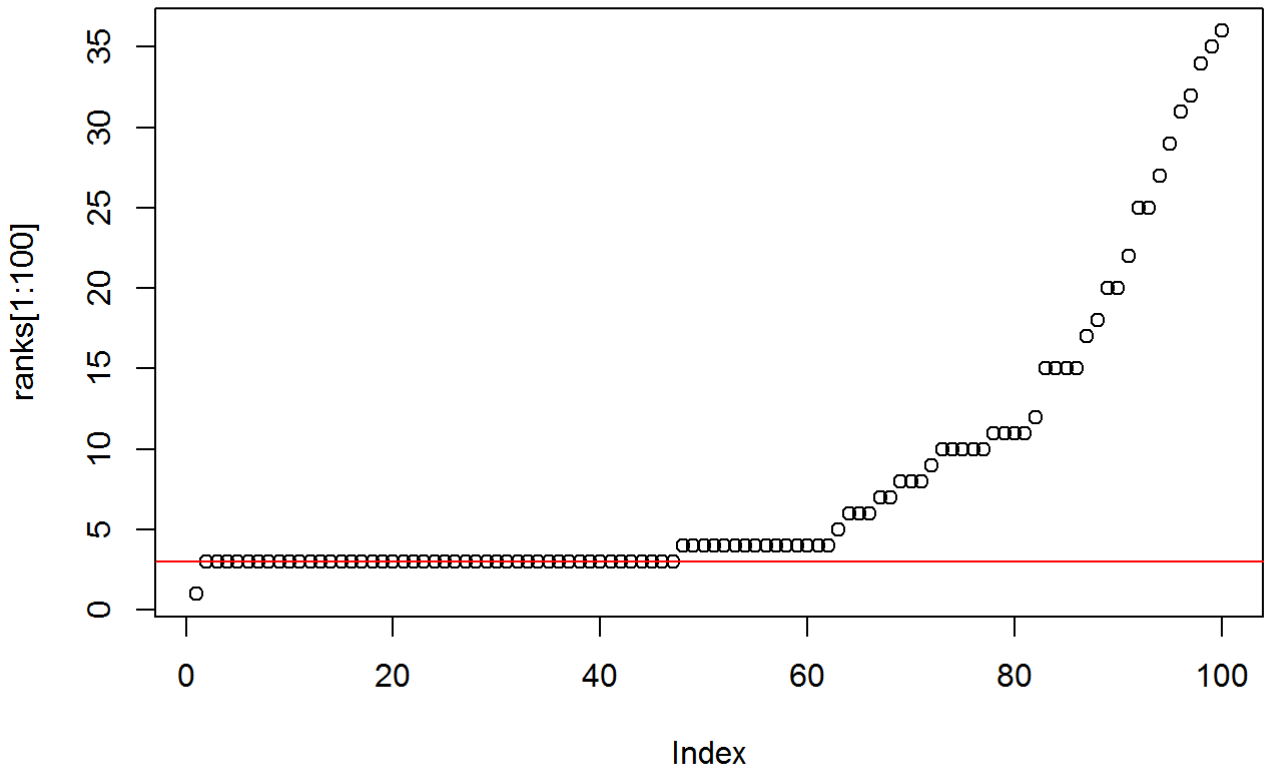
```
## [1] 280
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 6.217249e-14
```

```
tols <- 1.05^(0:(-1000))
ranks <- rep(0,length(tols))
for(i in 1:length(tols)){
  ranks[i] <- rankMatrix(ipout6$P,tol=tols[i])
}

plot(ranks)
```

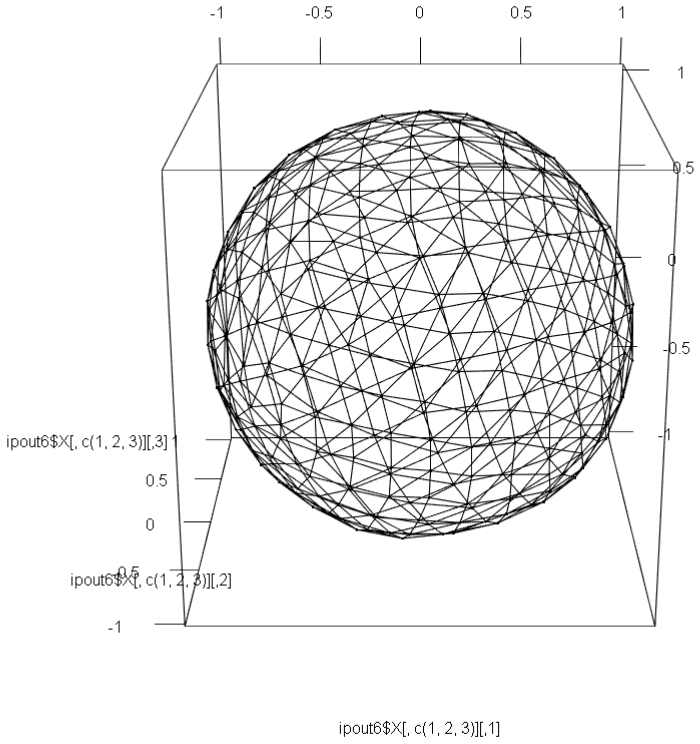


実質ランクは３なのでは？

```
plot(ranks[1:100])
abline(h=3,col=2)
```




```
npt <- length(ipout6$eout[[1]])
plot3d(ipout6$X[,c(1,2,3)])
#spheres3d(ipout5$X[,c(1,npt-1,npt)],radius=0.05)


el6 <- get.edgelist(g6)
segments3d(ipout6$X[t(el6),c(1,2,3)])
```

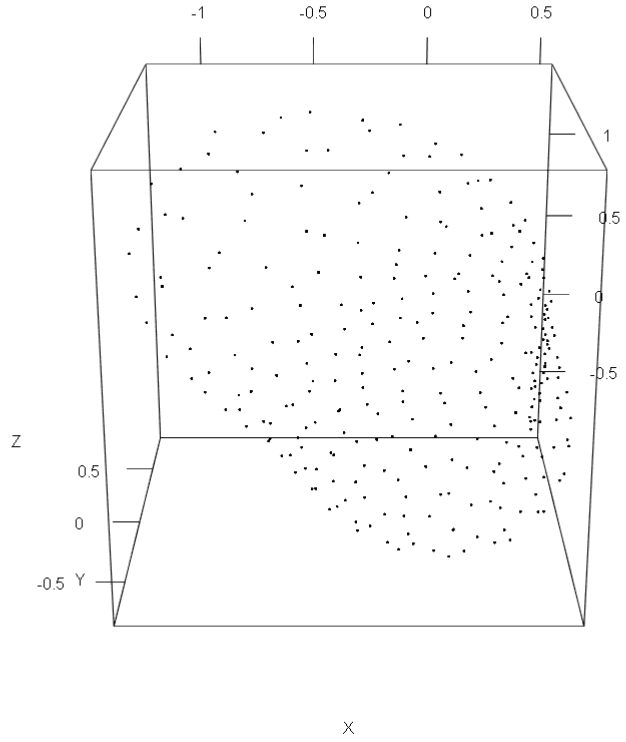


形をいびつにしてみる

ランクは３より上がるが、それほど膨大に上がりはしないか？

```
n <- 4 # メッシュの複雑さを指定(大きいと凹凸の周期が細かくなる)
k <- 3 # メッシュの複雑さを指定(大きいと真球に近くなる)
n.mesh <- 16 # メッシュの細かさを指定
A. <- matrix(runif(n^2), n, n)
A.[1, 1] <- k
A. <- A. + rnorm(n^2, 0, 0.05)
xxx <- my.spherical.harm.mesh(A = A., n = n.mesh)
```

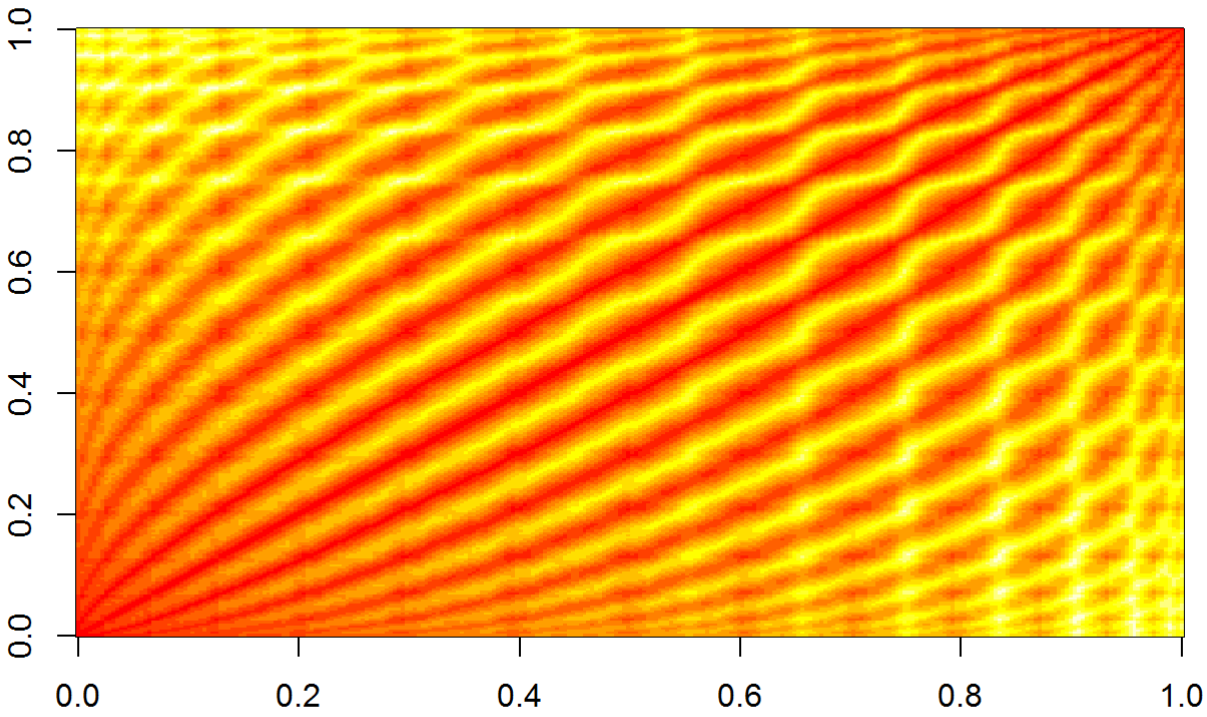
```
X <- xxx[[1]]
plot3d(X)
```



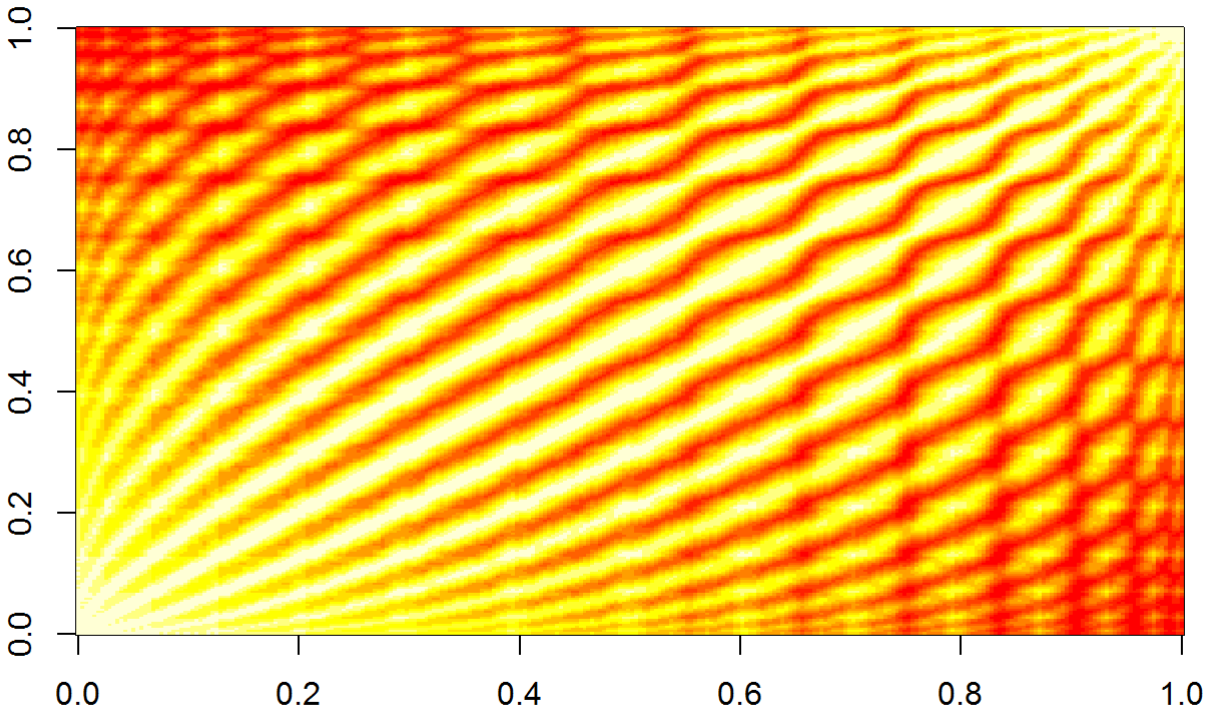

```
g7 <- graph.edgelist(xxx$edge,directed=FALSE)
#plot(g)

e.len7 <- rep(0,length(xxx$edge[,1]))
for(i in 1:length(e.len7)){
    e.len7[i] <- sqrt(sum((X[xxx$edge[i,1],]-X[xxx$edge[i,2],])^2))
}
```

```
ipout7 <- my.IPcoords2(g7,e.len7)
```

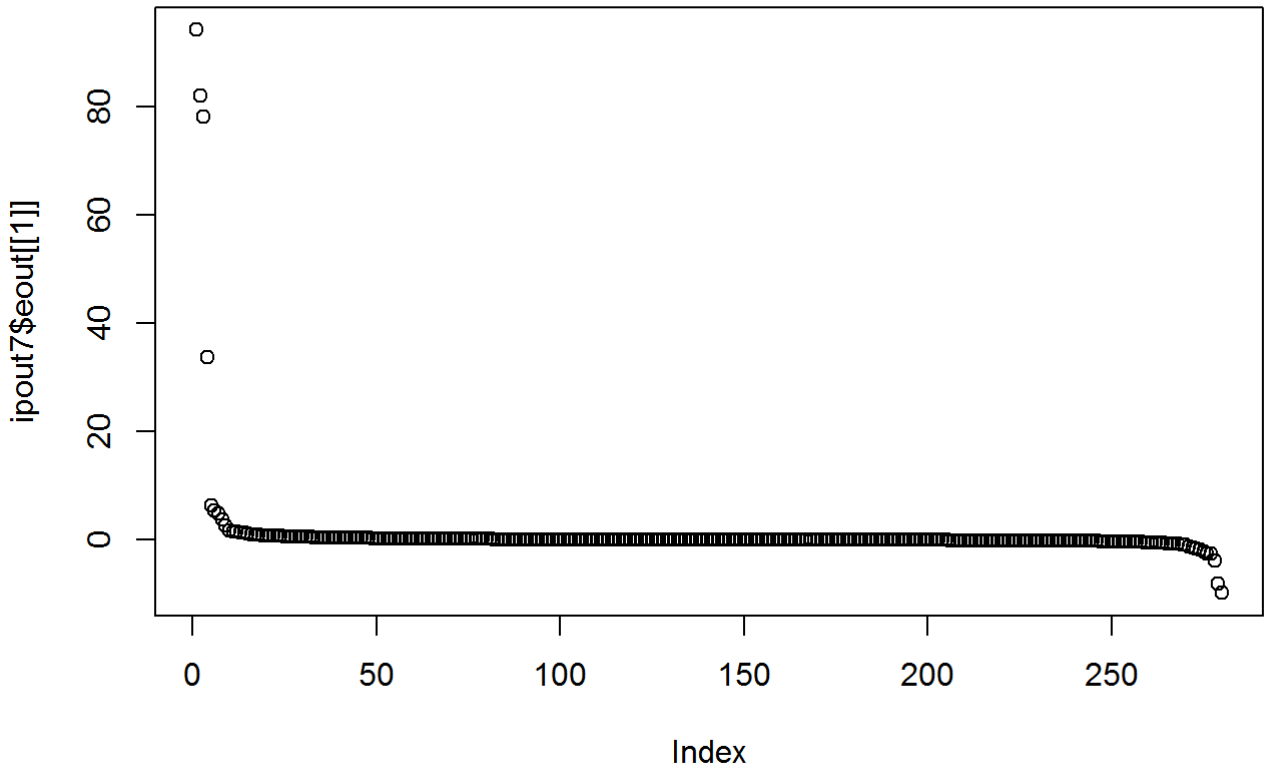
```
# 距離行列
image(ipout7$D)
```




```
# IP行列
image(ipout7$P)
```



```
# 固有値
plot(ipout7$eout[[1]])
```

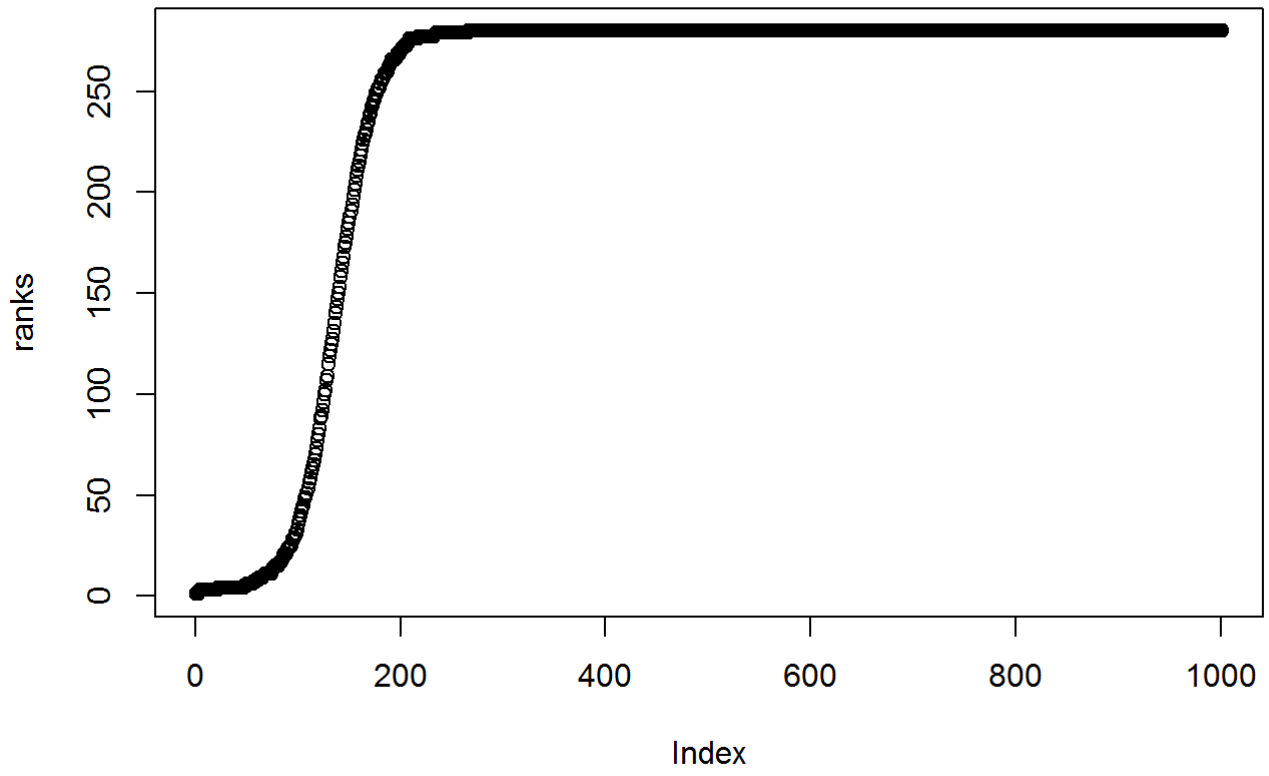



```
print(rankMatrix(ipout7$P))
```

```
## [1] 280
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 6.217249e-14
```

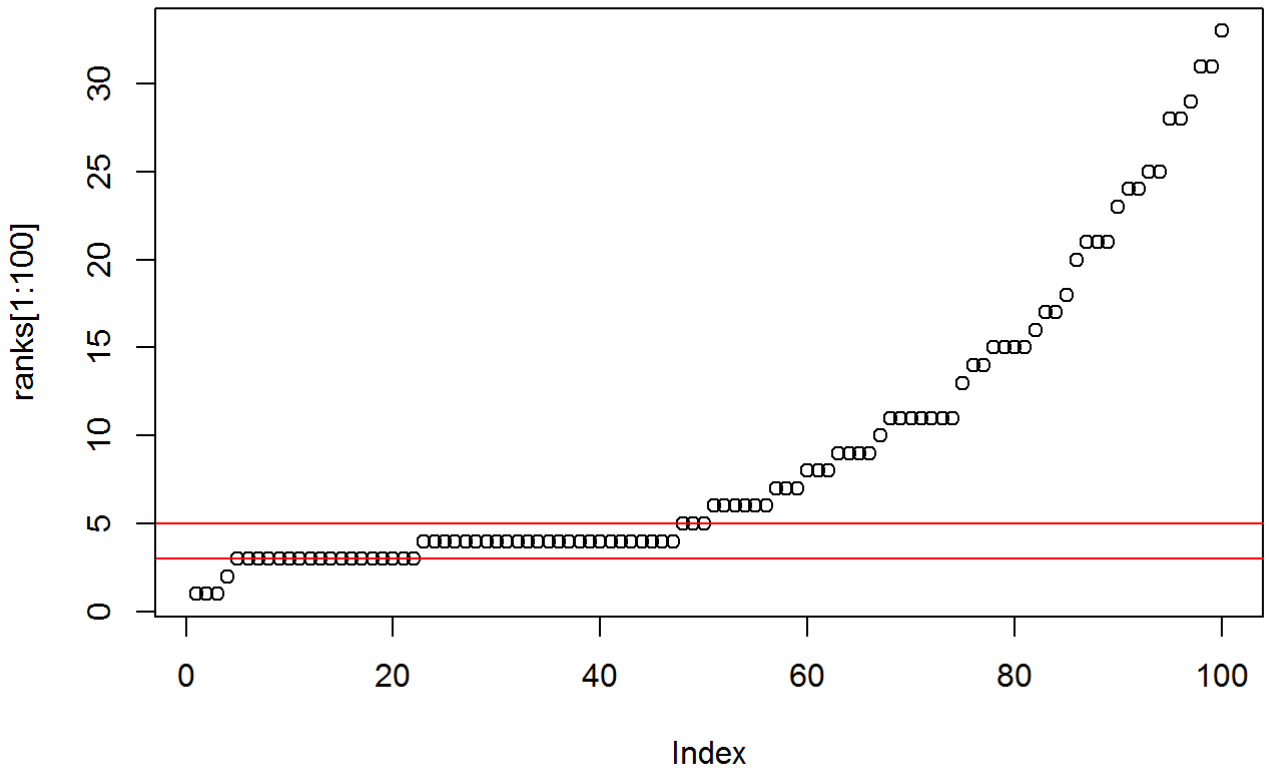
```
tols <- 1.05^(0:(-1000))
ranks <- rep(0,length(tols))
for(i in 1:length(tols)){
  ranks[i] <- rankMatrix(ipout7$P,tol=tols[i])
}
```

```
plot(ranks)
```

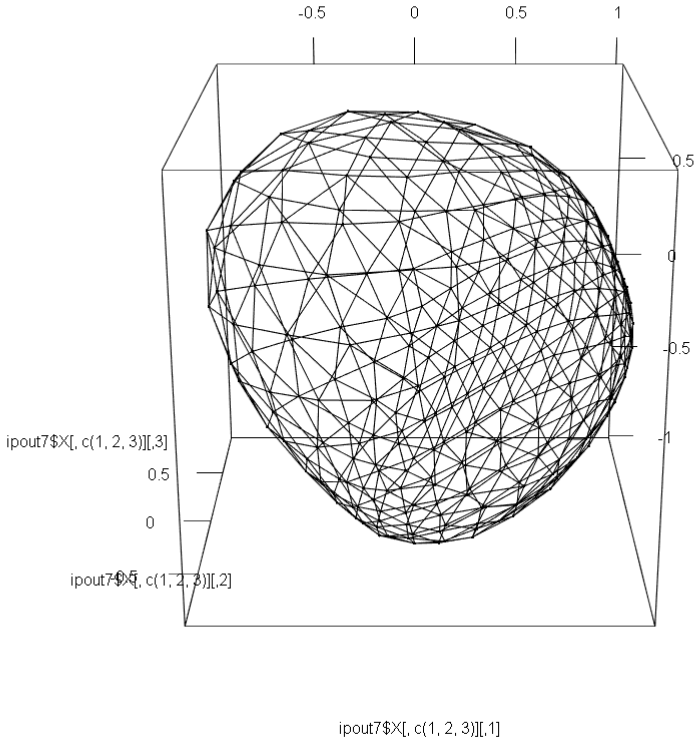


実質ランクは３ではなくなった？

```
plot(ranks[1:100])
abline(h=c(3,5),col=2)
```




```
npt <- length(ipout7$eout[[1]])
plot3d(ipout7$X[,c(1,2,3)])
#spheres3d(ipout5$X[,c(1,npt-1,npt)],radius=0.05)


el7 <- get.edgelist(g7)
segments3d(ipout7$X[t(el7),c(1,2,3)])
```


-0.5
0
0.5
1
0.5
0
-0.5
-1
ipout7$X[, c(1, 2, 3)][,3]
0.5
0
ipout7$X[, c(1, 2, 3)][,2]
ipout7$X[, c(1, 2, 3)][,1]