

低ランク最適化で等長多角形

ryamada

2019年12月14日

正三角形メッシュ座標を作りたい

正三角形貼り合わせた閉多面体の頂点座標を求めたい。

問題が難しいので低次元版で考えてみる

問題の低次元化

正三角形を、単位長線分とする。

閉多面体を、多角形(閉じた折れ線)とする。

線形代数問題・低ランク分解問題に変える

経緯は省略するが、この問題は以下のようにして解ける。

2次元等長多角形の辺ベクトルの内積行列

2次元平面にある等長 n 角形の辺ベクトルは単位ベクトル。このベクトルを $E_2 = (e_1^2, \dots, e_n^2)$ とする。

$$E_2^T E_2$$

は対角成分が1で、非対角成分は、 e_i^2, e_j^2 の内積である行列である。

しかも、各行の値は、辺をたどるごとに増減する X_1 軸方向の座標変化量であるので、この $E_2^T E_2$ 行列の行和(列和も)は0である。

また、生成過程からわかるように、ランクは2である。

高次元空間なら自由に座標を与えることができる

一方、 n 次元空間に正規直交基底を取り、各基底ベクトルの方向に長さ $1/\sqrt{2}$ のベクトルを取り、その先端を頂点として、 $1, 2, \dots, n, 1$ の順につなぐと、これは n 次元空間にある閉折れ線であって、すべての辺の長さが1である。

この n 次元閉折れ線の辺ベクトルを $E_n = (e_1^n, \dots, e_n^n)$ とする。

n 次元から2次元への線形埋め込み

ある $n \times n$ 行列 M が

$$ME_n = E_2'$$

を満足するとする。ただし、 E_2' は第1,2行が E_2 であり、残りの要素が0であるような行列とする。

この M は、 n 次元空間の閉折れ線の頂点を2次元空間に写像し、その2次元座標は、2次元等長多角形の頂点となる。

E_n は既知であるから、 $E_2(E_2')$ が与えられると、

$$M = E_2' E_n^{-1}$$

で求められる。

ただし、 E_n のランクはn-1なので、一般化逆行列を用いることにする。

非等長多角形の等長多角形近似

今、だいたい、等長になっている多角形が与えられたとする。

このときの辺ベクトルを F_2 とすると、その内積行列

$$F^T F$$

は、対角成分がおよそ1である。

また、行和(列和)は0である(閉じているから)。

したがって、この $F^T F$ をうまく「対角成分が1」で、「行和が0」で、「ランクが2」の行列に近似できれば、そこから、n次元-2次元変換行列Mを求めることができるはずである。

実際にやってみる。

等長多角形のランダム作成と、それに結構近い非等長多角形のランダム作成

まずは、具体例(等長多角形)とそこからのずれのある多角形を作ってみる。

```

# nは多角形の頂点数
# まず、正n角形を作り、sを使ってそこから少しずらす
my.zitter.npoly <- function(n, s=0.5) {
  # 正多角形の場合の、辺ベクトルを指定する角度を生成
  thetas <- 2*pi/n * (1:n)
  # 正多角形からずらす。角度だけをずらすことで辺の長さは1のまま
  thetas <- thetas + rnorm(n) * s
  # エッジベクトルを作る(単位ベクトル)
  edges <- cbind(cos(thetas), sin(thetas))
  # 頂点座標は、エッジベクトルの累積和
  # 1... , n-2頂点まで決める
  X <- apply(edges[1:(n-2), ], 2, cumsum)
  # n番頂点は、元に戻るので、原点
  xn <- c(0, 0)
  # n-2番頂点座標を取り出す
  xn_2 <- X[n-2, ]
  # n-1番頂点座標は、n-2番頂点とn番頂点との中点から
  # (n-2)---nベクトルに直交するベクトル方向に、適当な長さで離れた点になる
  # (n-2)---n間距離
  d <- sqrt(sum((xn-xn_2)^2))
  # 直交ベクトル
  perp <- c(xn_2[2], -xn_2[1])
  # n-1番頂点座標
  k <- sqrt(1-(d/2)^2) / sqrt(sum(perp^2))
  xn_1 <- xn_2/2 + k * perp
  # 全頂点座標を行列化
  X <- rbind(X, xn_1, xn)
  # 全辺のベクトルを算出
  edges <- apply(X[c(1:n, 1), ], 2, diff)
  return(list(X=X, edges=edges))
}

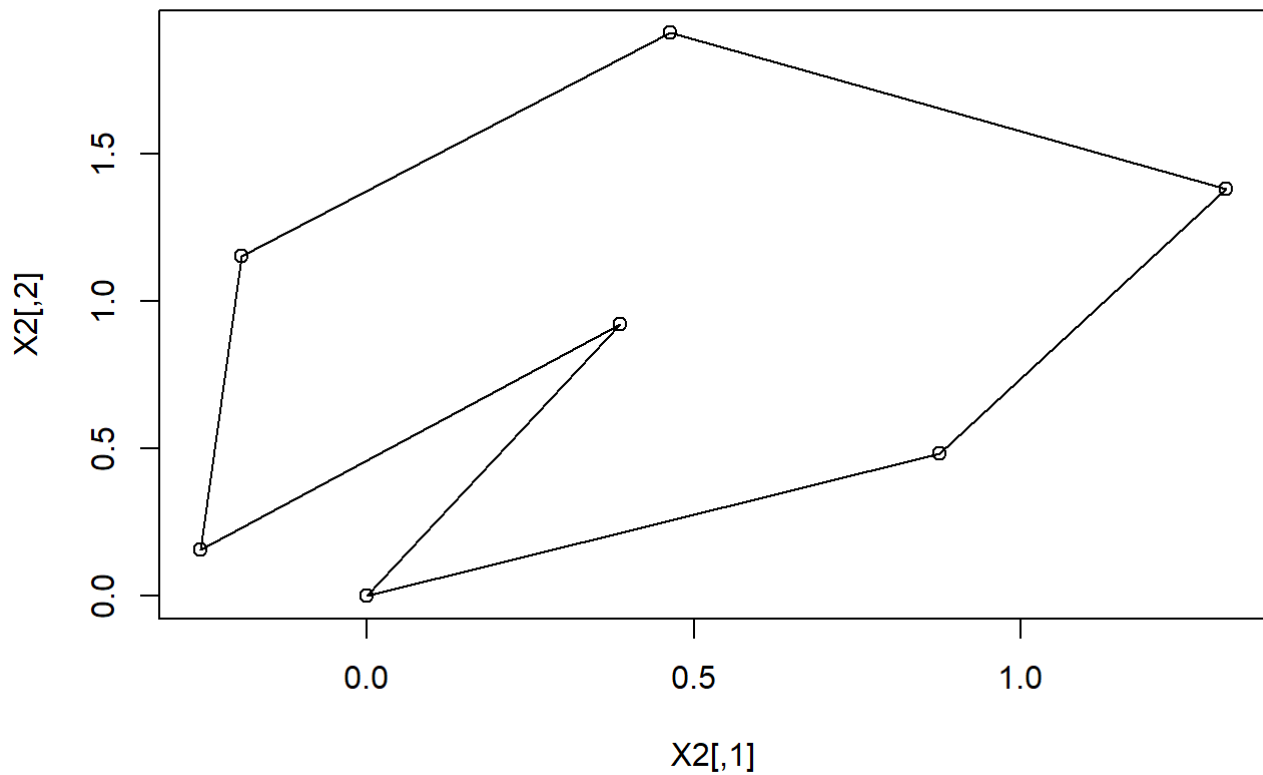
```

乱数の発生具合によっては、実現不可能な条件が出ることがあるので、その場合は、再作成すること。

```

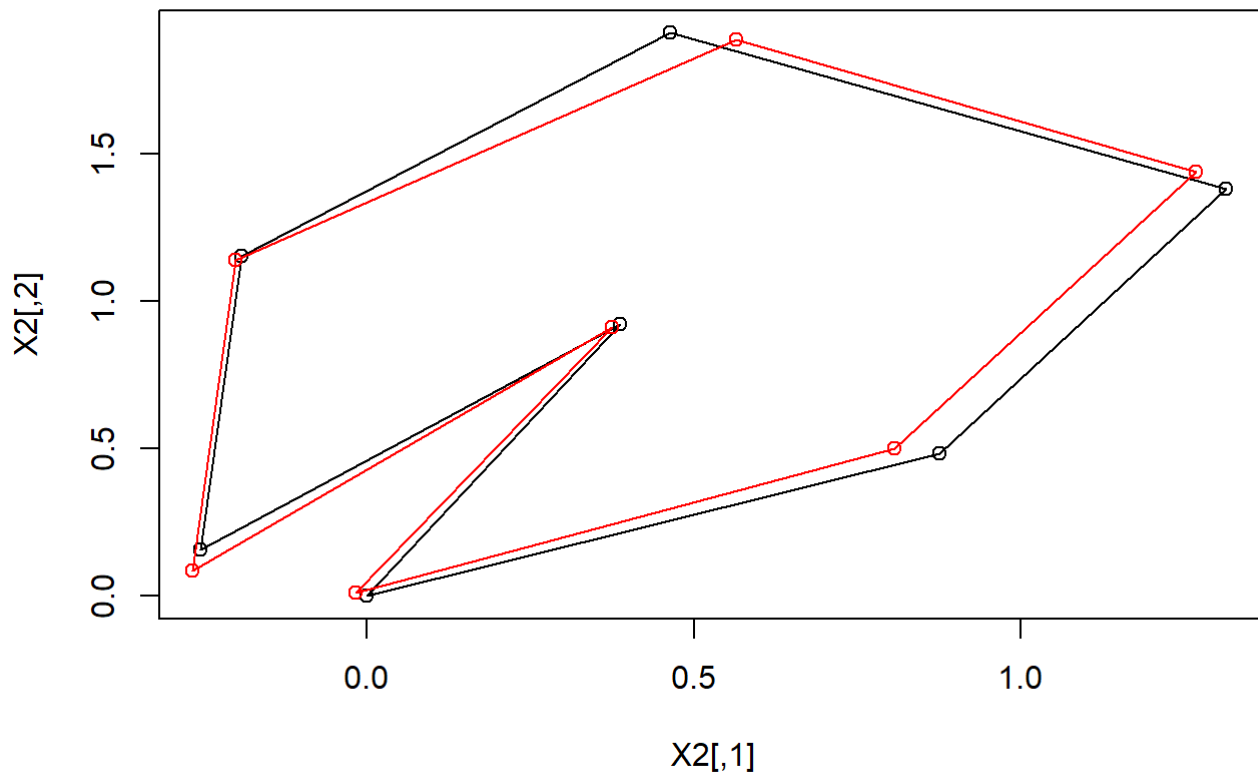
n <- 7
outz <- my.zitter.npoly(n)
X2 <- outz$X
plot(X2)
segments(X2[, 1], X2[, 2], X2[c(2:n, 1), 1], X2[c(2:n, 1), 2])

```



これに乱雑項を入れ、少しずらすことにする。

```
X2r <- X2 + rnorm(length(X2), mean=0, sd=0.05)
plot(X2)
segments(X2[, 1], X2[, 2], X2[c(2:n, 1), 1], X2[c(2:n, 1), 2])
points(X2r, col=2)
segments(X2r[, 1], X2r[, 2], X2r[c(2:n, 1), 1], X2r[c(2:n, 1), 2], col=2)
```



等長多角形と非等長多角形の辺ベクトル行列

上記で作成した2つの多角形の辺ベクトル行列を作り、その内積行列を計算する。

```
my.edges.poly <- function(X) {
  n <- length(X[, 1])
  apply(X[c(1:n, 1), ], 2, diff)
}
E2 <- my.edges.poly(X2)
E2r <- my.edges.poly(X2r)
IPE2 <- E2 %*% t(E2)
IPE2r <- E2r %*% t(E2r)
```

等長多角形からの辺内積行列は対角成分が1で、行和が0であることを確認する。

非等長多角形は、対角成分が1ではないが、行和は0であることを確認する。

```
diag(IPE2)
```

```
##
##      1      1      1      1      1      1      1
```

```
diag(IPE2r)
```

```
##
##      xn_1      xn
## 1.0962692 0.6929418 1.1457713 1.1080319 1.0931665 0.9659914 0.9191198
```

```
round(apply(IPE2, 1, sum), 10)
```

```
##
##      0      0      0      0      0      0      0
```

```
round(apply(IPE2r, 1, sum), 10)
```

```
##
##      0      0      0      0      0      0      0
```

等長多角形の辺内積行列から、 n 次元-2次元写像行列を算出する
 n 次元正規直交基底 (ベクトル長 $1/\sqrt{2}$) と n 次元空間での閉折れ線辺ベクトル行列

```
Xn <- 1/sqrt(2) * diag(rep(1,n))
Xn
```

```
##           [, 1]      [, 2]      [, 3]      [, 4]      [, 5]      [, 6]      [, 7]
## [1,] 0.7071068 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## [2,] 0.0000000 0.7071068 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## [3,] 0.0000000 0.0000000 0.7071068 0.0000000 0.0000000 0.0000000 0.0000000
## [4,] 0.0000000 0.0000000 0.0000000 0.7071068 0.0000000 0.0000000 0.0000000
## [5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.7071068 0.0000000 0.0000000
## [6,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.7071068 0.0000000
## [7,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.7071068
```

閉折れ線辺ベクトル。

```
En <- my.edges.poly(Xn)
En
```

```
##           [, 1]      [, 2]      [, 3]      [, 4]      [, 5]      [, 6]
## [1,] -0.7071068 0.7071068 0.0000000 0.0000000 0.0000000 0.0000000
## [2,] 0.0000000 -0.7071068 0.7071068 0.0000000 0.0000000 0.0000000
## [3,] 0.0000000 0.0000000 -0.7071068 0.7071068 0.0000000 0.0000000
## [4,] 0.0000000 0.0000000 0.0000000 -0.7071068 0.7071068 0.0000000
## [5,] 0.0000000 0.0000000 0.0000000 0.0000000 -0.7071068 0.7071068
## [6,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 -0.7071068
## [7,] 0.7071068 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##           [, 7]
## [1,] 0.0000000
## [2,] 0.0000000
## [3,] 0.0000000
## [4,] 0.0000000
## [5,] 0.0000000
## [6,] 0.7071068
## [7,] -0.7071068
```

```
En %*% t(En)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,]  1.0 -0.5  0.0  0.0  0.0  0.0 -0.5
## [2,] -0.5  1.0 -0.5  0.0  0.0  0.0  0.0
## [3,]  0.0 -0.5  1.0 -0.5  0.0  0.0  0.0
## [4,]  0.0  0.0 -0.5  1.0 -0.5  0.0  0.0
## [5,]  0.0  0.0  0.0 -0.5  1.0 -0.5  0.0
## [6,]  0.0  0.0  0.0  0.0 -0.5  1.0 -0.5
## [7,] -0.5  0.0  0.0  0.0  0.0 -0.5  1.0
```

写像行列の算出。

```
E2. <- matrix(0,n,n)
E2.[1:2,] <- t(E2)
library(matlib)
```

```
## Warning: package 'matlib' was built under R version 3.5.3
```

```
M <- E2. %*% Ginv(En)
```

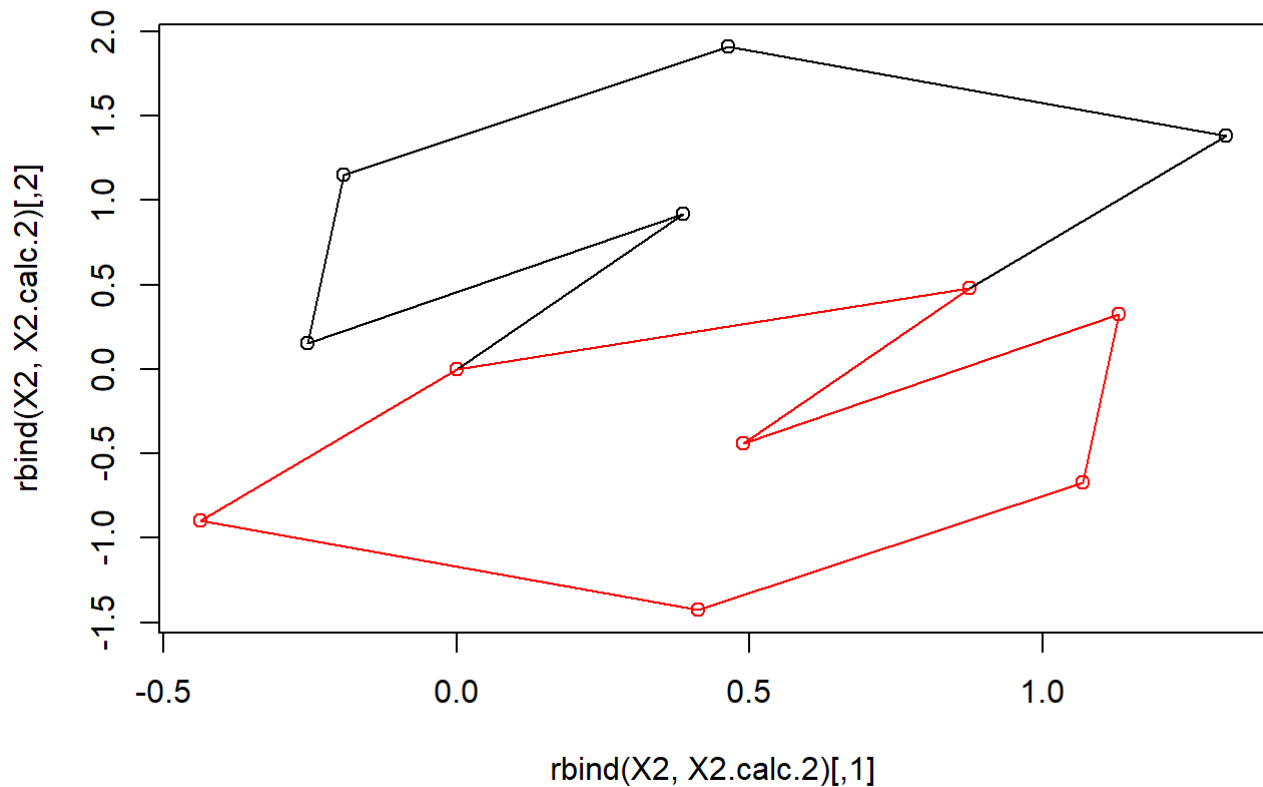
写像行列 M による、多角形座標の算出。

```
X2.calc <- M %*% Xn
X2.calc
```

```
##      [,1]      [,2]      [,3]      [,4]      [,5]      [,6] [,7]
## [1,] -0.4372839  0.4125331  1.0684218  1.1312087  0.4891908  0.8762560    0
## [2,] -0.8993235 -1.4264014 -0.6715438  0.3264832 -0.4402065  0.4818459    0
## [3,]  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000    0
## [4,]  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000    0
## [5,]  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000    0
## [6,]  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000    0
## [7,]  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000    0
```

```
X2.calc.2 <- t(X2.calc[1:2,])
```

```
plot(rbind(X2,X2.calc.2),col=rep(1:2,each=n))
segments(X2[,1],X2[,2],X2[c(2:n,1),1],X2[c(2:n,1),2])
segments(X2.calc.2[,1],X2.calc.2[,2],X2.calc.2[c(2:n,1),1],X2.calc.2[c(2:n,1),2],col=2)
```



頂点間距離が確かに1になっていることを確認する。

対角部分と、左下隅が1になる。

```
dist(X2)
```

```
##
##          1.0000000
##      1.4848584 1.0000000
##      1.2619414 1.5228373 1.0000000
##      1.1773803 1.9906710 1.8944917 1.0000000
## xn_1 0.6580952 1.0339941 0.9891697 0.6237191 1.0000000
## xn   1.0000000 1.9060472 1.9637837 1.1692884 0.2985606 1.0000000
```

```
dist(X2.calc.2)
```

```
##          1          2          3          4          5          6
## 2 1.0000000
## 3 1.5228373 1.0000000
## 4 1.9906710 1.8944917 1.0000000
## 5 1.0339941 0.9891697 0.6237191 1.0000000
## 6 1.9060472 1.9637837 1.1692884 0.2985606 1.0000000
## 7 1.0000000 1.4848584 1.2619414 1.1773803 0.6580952 1.0000000
```

非等長多角形の辺内積行列の近似

Rのrsvdパッケージのrsvd()関数を用いて、2ランク分解し、対角成分を1に、行和を0にすることを繰り返して、収束を試みることにする。

```
library(rsvd)
```

```
## Warning: package 'rsvd' was built under R version 3.5.3
```

```
my.rsvd.diag1 <- function(M,k,eps=10^(-10),maxiter=1000) {
  K <- M
  n <- length(M[,1])
  ret <- list()
  ret[[1]] <- K
  iter.cnt <- 1
  while(iter.cnt < maxiter) {
    check.diag <- max(abs(diag(K)-1))
    check.rowsum <- max(abs(apply(K,1,sum)))
    if(check.diag < eps & check.rowsum < eps) {
      tmp <- ret[[length(ret)]]
      tmp.rsvd <- rsvd(tmp,k=k)
      edges <- diag(sqrt(tmp.rsvd$d)) %*% t(tmp.rsvd$u)
      return(list(edges = edges, IPmat = ret))
      break
    }
    target.diag <- 1
    diag(K) <- target.diag
    tmp2 <- apply(K,1,sum)

    K <- K - tmp2/(n-1)
    diag(K) <- target.diag
    tmp <- rsvd(K,k=k)
    K <- tmp$v %*% diag(tmp$d) %*% t(tmp$u)
    ret[[iter.cnt+1]] <- K
    iter.cnt <- iter.cnt + 1
  }
  edges <- diag(sqrt(tmp$d)) %*% t(tmp$u)
  return(list(edges = edges, IPmat = ret))
}
```

```
IPE2r.approx <- my.rsvd.diag1(IPE2r,k=2)
(n.iter <- length(IPE2r.approx$IPmat)) # 繰り返し処理回数
```

```
## [1] 128
```

```
E2r.approx <- IPE2r.approx$edges
```

近似辺内積行列からの写像行列の算出。

```
E2r.approx. <- matrix(0,n,n)
E2r.approx.[1:2,] <- E2r.approx
library(matlib)
M.approx <- E2r.approx. %*% Ginv(En)
```

算出写像行列からの2次元多角形頂点座標算出。

```
X2.approx <- M.approx %*% Xn
X2.approx.2 <- t(X2.approx[1:2,])
dist(X2.approx.2)
```

```
##           1           2           3           4           5           6
## 2 1.0015012
## 3 1.5021995 0.9997572
## 4 2.0185778 1.9183538 0.9999433
## 5 1.0774254 0.9879902 0.5432452 1.0000835
## 6 1.9283177 1.9726023 1.1421317 0.2834478 0.9997277
## 7 0.9998808 1.4501053 1.1783180 1.1666152 0.6505042 0.9991218
```

```
plot(rbind(X2r,X2.approx.2),col=rep(1:2,each=n))
segments(X2r[,1],X2r[,2],X2r[c(2:n,1),1],X2r[c(2:n,1),2])
segments(X2.approx.2[,1],X2.approx.2[,2],X2.approx.2[c(2:n,1),1],X2.approx.2[c(2:n,1),2],col=2)
```

