cktRdn.R

pavan

Wed Jan 11 23:08:01 2017

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(Matrix)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
df <- read.csv('~/tmp/gld/gmp/78g1/run_2015-july-1st-31st_xfo-to-load-direct/nam.csv',
           colClasses = c("integer", "integer", "integer", "integer", "integer", "character"),
           header = T) %>%
    as.tbl()



Y <- NULL
Z <- NULL
PSZ <- NULL

# Derive the *positive-sequence* representation of the nodal admittance/impedance matrix:
for (str in df[["admittance"]]) {
    v <- -eval(parse(text = str))
    m <- matrix(v, nrow = sqrt(length(v)), byrow = T)
    M <- 1 / m
    Y <- c(Y, lst(m))
    Z <- c(Z, lst(M))
    if (length(v) == 9) { # 3-phase link: Bergen & Vittal (2nd ed), page 473
        z0s <- (M[1,1] + M[2,2] + M[3,3]) / 3 # eq 12.39(1)
        z0m <- (M[2,3] + M[3,1] + M[1,2]) / 3 # eq 12.40(1)
    } else if (length(v) == 4) { # 2-phase link: FUHGEDDABOUTIT!
        z0s <- (M[1,1] + M[2,2]) / 2
        z0m <- (M[2,1] + M[1,2]) / 2
    } else if (length(v) == 1) { # 1-phase link: straightforward
        z0s <- M[1,1]
        z0m <- 0
```

```
    }
    psz <- z0s - z0m
    PSZ <- c(PSZ, psz)
}

dimnames(Y) <- NULL
dimnames(Z) <- NULL
dimnames(PSZ) <- NULL

DF <- df %>%
    dplyr::select(-admittance) %>%
    mutate(Y = Y, Z = Z, PSZ = PSZ, PSY = 1 / PSZ)



# Adjacency Matrix (populated with PSYs instead of 1s):
# (Sadly, the Matrix package isn't equipped to store complex elements;
#  so we'll have to store the real and imaginary parts of PSY separately.)
AM_real <- sparseMatrix(i = DF$from_node, j = DF$to_node, x = Re(DF$PSY))
AM_imag <- sparseMatrix(i = DF$from_node, j = DF$to_node, x = Im(DF$PSY))

N = nrow(AM_real) # = ncol(AM_real)

# Degree Matrix:
DM_real <- Diagonal(x = rowSums(AM_real))
DM_imag <- Diagonal(x = rowSums(AM_imag))

# Laplacian Matrix:
LM_real <- DM_real - AM_real
LM_imag <- DM_imag - AM_imag

# Generalized Inverse (Moore-Penrose) of LM:
GI <- ginv(X = as.matrix(LM_real) + as.matrix(LM_imag) * 1i)



# Electrical Distance Matrix:
# (Equation 1 from
#  "Multi-Attribute Partitioning of Power Networks Based on Electrical Distance,"
#  Cotilla-Sanchez et al., IEEE Trans. Power Systems, Vol. 28, No. 4, November 2103.)
ED <- matrix(0+0i, nrow = N, ncol = N)
for (i in 1:N) {
    for(j in 1:N) {
        ED[i,j] <- GI[i,i] - GI[i,j] - GI[j,i] + GI[j,j]
    }
}



# K-Means Clustering on the Real & Imaginary components of GI (or ED):
set.seed(9)
KM <- kmeans(x = cbind(Re(ED), Im(ED)),
         centers = 10,
```

```
        nstart = ceiling(0.1*N))



# report & save:
KM$size
```

```
##  [1] 49 16 25 60 49 41 35 31 26 42
```

```
KM$cluster
```

```
##   [1]  9  9  7  7  3  3  4  6  6  8  8  6  6  1  1  9  9  6  6  6  6  8  8
##  [24]  8  6  6  1  1  6  6 10 10  6  6 10  5  5  6  6  4  9  9  1  5 10 10
##  [47] 10  7 10 10  1  1  8 10 10  5  5 10  7  7  3  3  3  4  7  4  4  4  4
##  [70]  4  4  4  4  4  1  1  6  6  4  3  4  4  7  4  1  5  1  1  5  7  6  6
##  [93]  6  6  1  1  5  5  1  5  5  7  7  5  5  1  1  5  1  3  3  1  1  5  3
## [116]  1  5  5  5  1  1  5  3  5  1  3  3  1  1  1  2  2  5  5  4  4  5  1
## [139]  3  6  3  3  6  6  3  6  6  7  7  6  7  5  5  7  7  7  5  6  6  7  7
## [162]  7  7  1  4  4  6  6  4  4  5  5  4  4  4  4  4  8  8  3  4  1  4  5
## [185]  5  4  7  7  6  6  7  1  1  5  1  1  4  4  5  7  1  1  4  7  4  8  8
## [208]  8  7  7  7  6  5  8  8 10  1  7  4  2  9  9  9  9  2  9  3  3  1  1
## [231]  5  5  1  1  5  6  5  1  4  5  4  5  1  4  5  1  4  5  6  7  6  6  4
## [254]  1  5  5  2  2  5  2  5  2  2  5  1  1  1  5  5  1  5  9 10 10 10 10
## [277]  3 10 10  9  9  4  4  3  8  8  8  8  4  4  4  4  2  2  4  4 10  8  8
## [300]  9  9  8  3  7 10 10 10 10  3 10 10  8  8  2  2  8  7  7  2  9  9  2
## [323]  2  9  9  4  4 10  4 10  8  8 10 10  9  6  6  9  4  4  4 10  8  9  8
## [346]  9 10 10  3 10 10  8  8  4 10 10  8  4  4  4  3  9  7  7 10 10 10 10
## [369]  8  4  4  9 10 10
```

```
round(KM$betweenss / KM$totss, digits = 2)
```

```
## [1] 0.95
```

```
write.csv(data.frame(cluster = KM$cluster),
      file = '~/tmp/gld/gmp/78g1/run_2015-july-1st-31st_xfo-to-load-direct/node-cluster.csv',
      quote = F)
```