SUN JAY YOO

WU LAB, JOHNS HOPKINS UNIVERSITY

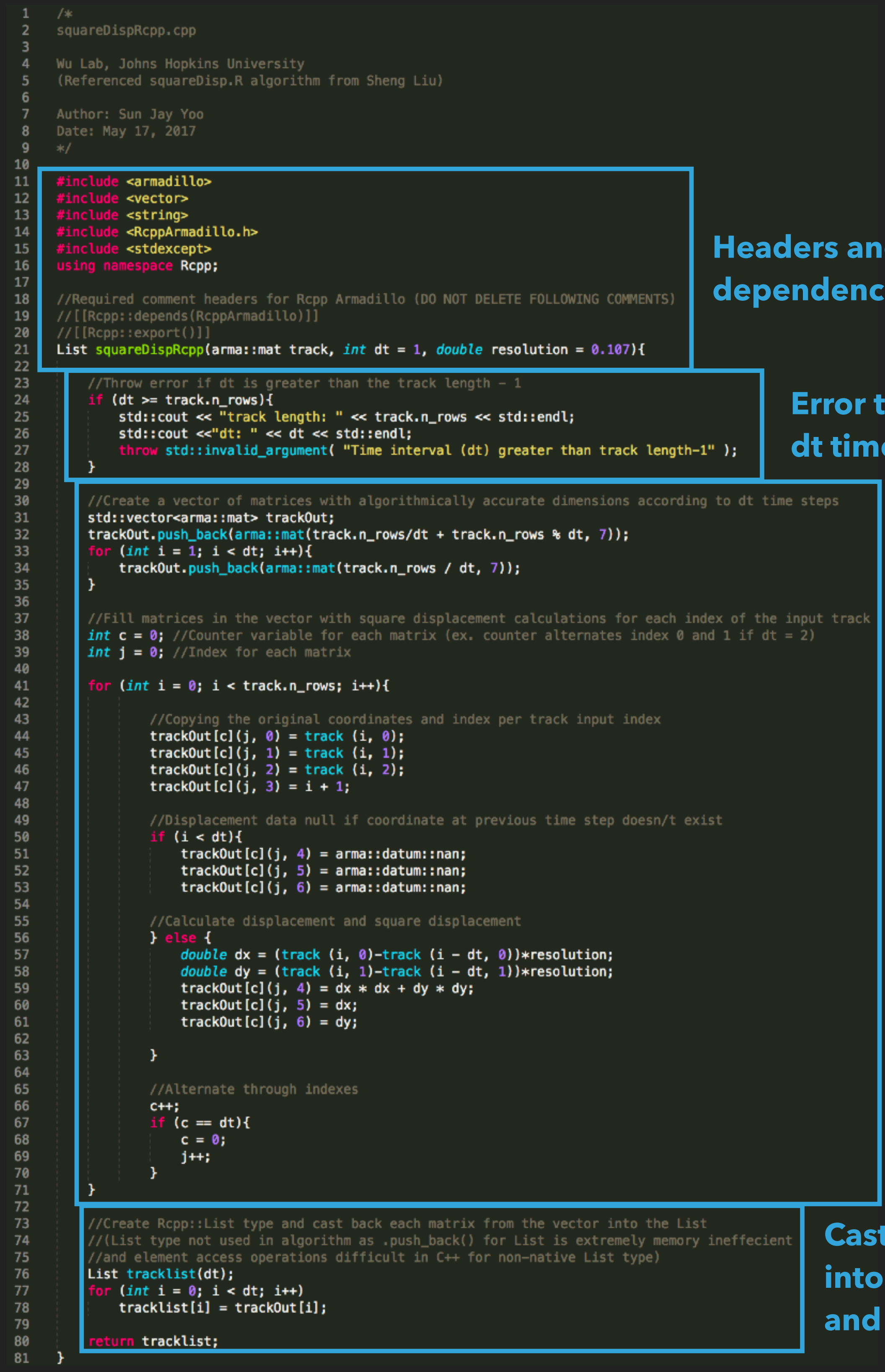
SQUARE DISPLACEMENT ALGORITHM IN C++

USING RCPP TO IMPLEMENT R WITH C++

- To increase efficiency in time complexity (shorten computation time) and memory usage
- Rcpp and RcppArmadillo used to link source types and standardize I/O
  - https://cran.r-project.org/web/packages/RcppArmadillo/RcppArmadillo.pdf\
- Track input, List output maintained in R and the accompanying C++ code.

SOURCE CODE

SquareDispRcpp.cpp



```cpp
/*
squareDispRcpp.cpp

Wu Lab, Johns Hopkins University
(Referenced squareDisp.R algorithm from Sheng Liu)

Author: Sun Jay Yoo
Date: May 17, 2017
*/

#include <armadillo>
#include <vector>
#include <string>
#include <RcppArmadillo.h>
#include <stdexcept>
using namespace Rcpp;

//Required comment headers for Rcpp Armadillo (DO NOT DELETE FOLLOWING COMMENTS)
//[[Rcpp::depends(RcppArmadillo)]]
//[[Rcpp::export()]]
List squareDispRcpp(arma::mat track, int dt = 1, double resolution = 0.107){

    //Throw error if dt is greater than the track length - 1
    if (dt >= track.n_rows){
        std::cout << "track length: " << track.n_rows << std::endl;
        std::cout <<"dt: " << dt << std::endl;
        throw std::invalid_argument( "Time interval (dt) greater than track length-1" );
    }

    //Create a vector of matrices with algorithmically accurate dimensions according to dt time steps
    std::vector<arma::mat> trackOut;
    trackOut.push_back(arma::mat(track.n_rows/dt + track.n_rows % dt, 7));
    for (int i = 1; i < dt; i++){
        trackOut.push_back(arma::mat(track.n_rows / dt, 7));
    }

    //Fill matrices in the vector with square displacement calculations for each index of the input track
    int c = 0; //Counter variable for each matrix (ex. counter alternates index 0 and 1 if dt = 2)
    int j = 0; //Index for each matrix

    for (int i = 0; i < track.n_rows; i++){

        //Copying the original coordinates and index per track input index
        trackOut[c](j, 0) = track (i, 0);
        trackOut[c](j, 1) = track (i, 1);
        trackOut[c](j, 2) = track (i, 2);
        trackOut[c](j, 3) = i + 1;

        //Displacement data null if coordinate at previous time step doesn/t exist
        if (i < dt){
            trackOut[c](j, 4) = arma::datum::nan;
            trackOut[c](j, 5) = arma::datum::nan;
            trackOut[c](j, 6) = arma::datum::nan;

        //Calculate displacement and square displacement
        } else {
            double dx = (track (i, 0)-track (i - dt, 0))*resolution;
            double dy = (track (i, 1)-track (i - dt, 1))*resolution;
            trackOut[c](j, 4) = dx * dx + dy * dy;
            trackOut[c](j, 5) = dx;
            trackOut[c](j, 6) = dy;

        }

        //Alternate through indexes
        c++;
        if (c == dt){
            c = 0;
            j++;
        }
    }

    //Create Rcpp::List type and cast back each matrix from the vector into the List
    //(List type not used in algorithm as .push_back() for List is extremely memory ineffecient
    //and element access operations difficult in C++ for non-native List type)
    List tracklist(dt);
    for (int i = 0; i < dt; i++)
        tracklist[i] = trackOut[i];

    return tracklist;
}
```

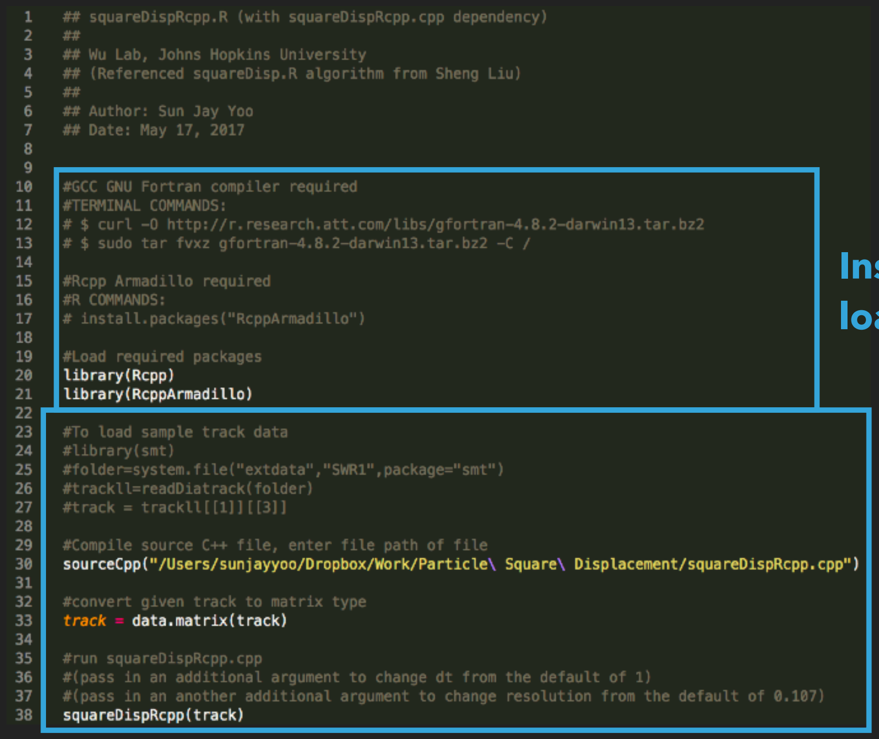
Headers and dependencies

Error throwing when dt time step too large

Square displacement algorithm using C++ vector intermediary for operations

Cast matrix vector into R List type and return

SquareDispRcpp.R



```r
## squareDispRcpp.R (with squareDispRcpp.cpp dependency)
##
## Wu Lab, Johns Hopkins University
## (Referenced squareDisp.R algorithm from Sheng Liu)
##
## Author: Sun Jay Yoo
## Date: May 17, 2017


#GCC GNU Fortran compiler required
#TERMINAL COMMANDS:
# $ curl -O http://r.research.att.com/libs/gfortran-4.8.2-darwin13.tar.bz2
# $ sudo tar fvxz gfortran-4.8.2-darwin13.tar.bz2 -C /

#Rcpp Armadillo required
#R COMMANDS:
# install.packages("RcppArmadillo")

#Load required packages
library(Rcpp)
library(RcppArmadillo)

#To load sample track data
#library(smt)
#folder=system.file("extdata","SWR1",package="smt")
#trackll=readDiatrack(folder)
#track = trackll[[1]][[3]]

#Compile source C++ file, enter file path of file
sourceCpp("/Users/sunjayyoo/Dropbox/Work/Particle\ Square\ Displacement/squareDispRcpp.cpp")

#convert given track to matrix type
track = data.matrix(track)

#run squareDispRcpp.cpp
#(pass in an additional argument to change dt from the default of 1)
#(pass in an another additional argument to change resolution from the default of 0.107)
squareDispRcpp(track)
```

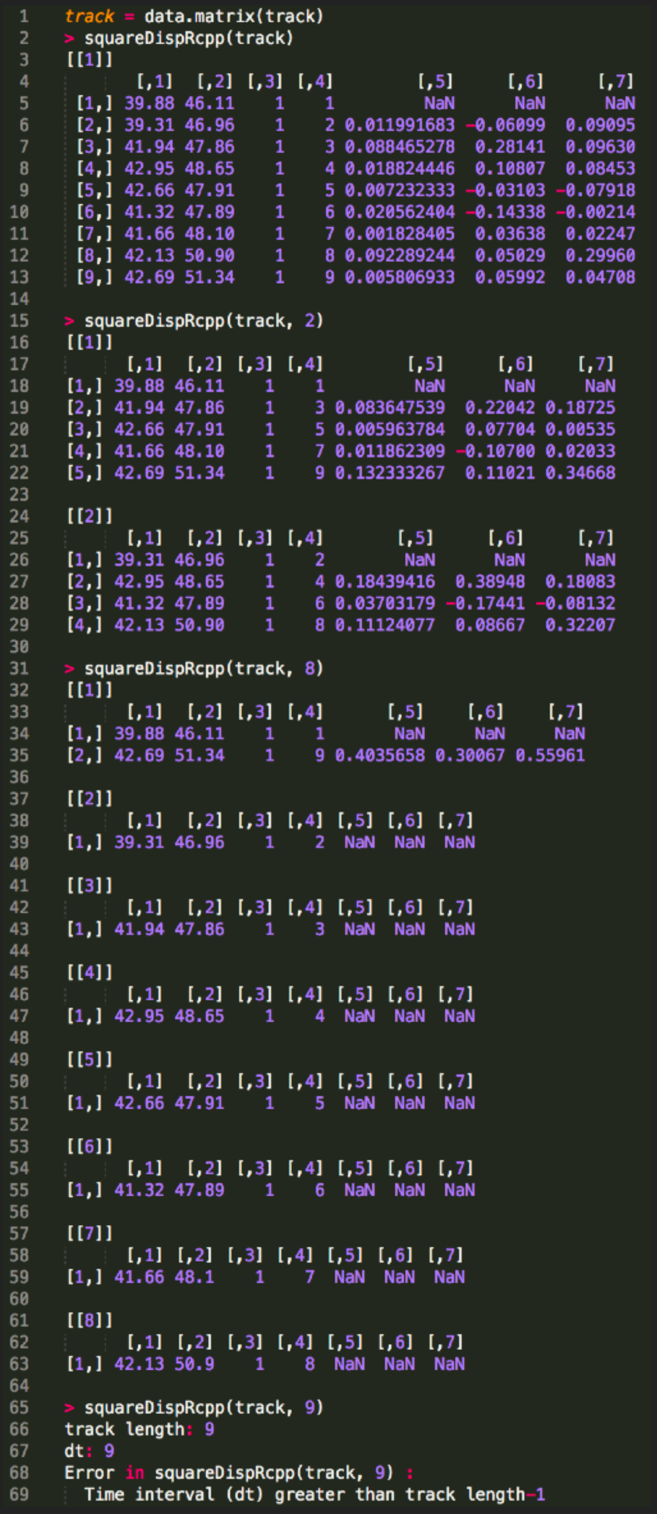
Install and load packages

Compile and run sample track

SAMPLE OUTPUT

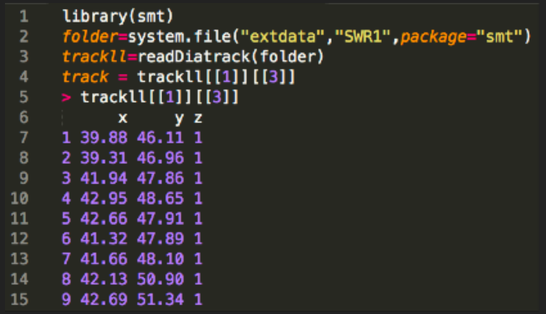
Output

SquareDispRcpp.R (with C++)



```
track = data.matrix(track)
> squareDispRcpp(track)
[[1]]
       [,1]  [,2] [,3] [,4]        [,5]     [,6]     [,7]
 [1,] 39.88 46.11    1    1         NaN      NaN      NaN
 [2,] 39.31 46.96    1    2 0.011991683 -0.06099  0.09095
 [3,] 41.94 47.86    1    3 0.088465278  0.28141  0.09630
 [4,] 42.95 48.65    1    4 0.018824446  0.10807  0.08453
 [5,] 42.66 47.91    1    5 0.007232333 -0.03103 -0.07918
 [6,] 41.32 47.89    1    6 0.020562404 -0.14338 -0.00214
 [7,] 41.66 48.10    1    7 0.001828405  0.03638  0.02247
 [8,] 42.13 50.90    1    8 0.092289244  0.05029  0.29960
 [9,] 42.69 51.34    1    9 0.005806933  0.05992  0.04708

> squareDispRcpp(track, 2)
[[1]]
      [,1]  [,2] [,3] [,4]        [,5]     [,6]    [,7]
[1,] 39.88 46.11    1    1         NaN      NaN     NaN
[2,] 41.94 47.86    1    3 0.083647539  0.22042 0.18725
[3,] 42.66 47.91    1    5 0.005963784  0.07704 0.00535
[4,] 41.66 48.10    1    7 0.011862309 -0.10700 0.02033
[5,] 42.69 51.34    1    9 0.132333267  0.11021 0.34668

[[2]]
      [,1]  [,2] [,3] [,4]       [,5]     [,6]     [,7]
[1,] 39.31 46.96    1    2        NaN      NaN      NaN
[2,] 42.95 48.65    1    4 0.18439416  0.38948  0.18083
[3,] 41.32 47.89    1    6 0.03703179 -0.17441 -0.08132
[4,] 42.13 50.90    1    8 0.11124077  0.08667  0.32207

> squareDispRcpp(track, 8)
[[1]]
      [,1]  [,2] [,3] [,4]      [,5]    [,6]    [,7]
[1,] 39.88 46.11    1    1       NaN     NaN     NaN
[2,] 42.69 51.34    1    9 0.4035658 0.30067 0.55961

[[2]]
      [,1]  [,2] [,3] [,4] [,5] [,6] [,7]
[1,] 39.31 46.96    1    2  NaN  NaN  NaN

[[3]]
      [,1]  [,2] [,3] [,4] [,5] [,6] [,7]
[1,] 41.94 47.86    1    3  NaN  NaN  NaN

[[4]]
      [,1]  [,2] [,3] [,4] [,5] [,6] [,7]
[1,] 42.95 48.65    1    4  NaN  NaN  NaN

[[5]]
      [,1]  [,2] [,3] [,4] [,5] [,6] [,7]
[1,] 42.66 47.91    1    5  NaN  NaN  NaN

[[6]]
      [,1]  [,2] [,3] [,4] [,5] [,6] [,7]
[1,] 41.32 47.89    1    6  NaN  NaN  NaN

[[7]]
      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,] 41.66 48.1    1    7  NaN  NaN  NaN

[[8]]
      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,] 42.13 50.9    1    8  NaN  NaN  NaN

> squareDispRcpp(track, 9)
track length: 9
dt: 9
Error in squareDispRcpp(track, 9) :
  Time interval (dt) greater than track length-1
```

Input

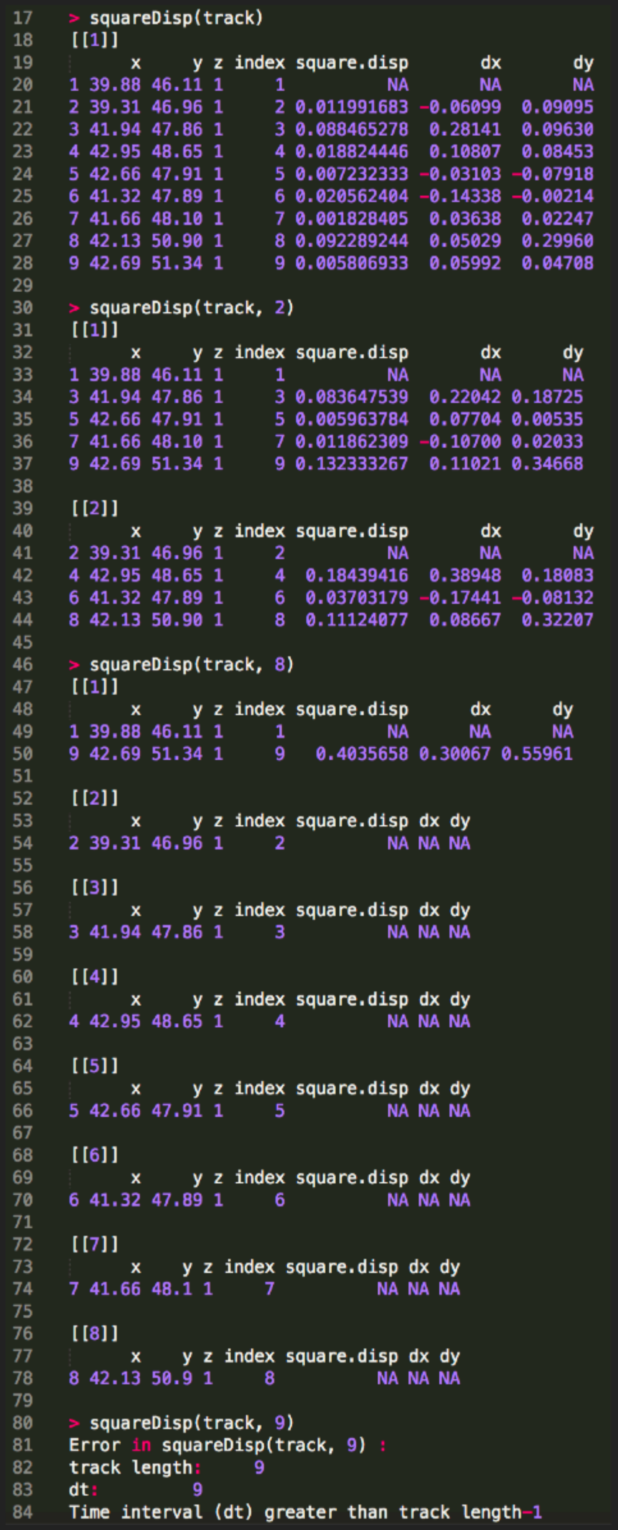


```
library(smt)
folder=system.file("extdata","SWR1",package="smt")
trackll=readDiatrack(folder)
track = trackll[[1]][[3]]
> trackll[[1]][[3]]
      x     y z
1 39.88 46.11 1
2 39.31 46.96 1
3 41.94 47.86 1
4 42.95 48.65 1
5 42.66 47.91 1
6 41.32 47.89 1
7 41.66 48.10 1
8 42.13 50.90 1
9 42.69 51.34 1
```

Identical outputs for dt = 1, 2, 8 and error thrown when dt = 9

Output

SquareDispR (without C++)



```
> squareDisp(track)
[[1]]
      x     y z index square.disp       dx       dy
1 39.88 46.11 1     1          NA       NA       NA
2 39.31 46.96 1     2 0.011991683 -0.06099  0.09095
3 41.94 47.86 1     3 0.088465278  0.28141  0.09630
4 42.95 48.65 1     4 0.018824446  0.10807  0.08453
5 42.66 47.91 1     5 0.007232333 -0.03103 -0.07918
6 41.32 47.89 1     6 0.020562404 -0.14338 -0.00214
7 41.66 48.10 1     7 0.001828405  0.03638  0.02247
8 42.13 50.90 1     8 0.092289244  0.05029  0.29960
9 42.69 51.34 1     9 0.005806933  0.05992  0.04708

> squareDisp(track, 2)
[[1]]
      x     y z index square.disp       dx      dy
1 39.88 46.11 1     1          NA       NA      NA
3 41.94 47.86 1     3 0.083647539  0.22042 0.18725
5 42.66 47.91 1     5 0.005963784  0.07704 0.00535
7 41.66 48.10 1     7 0.011862309 -0.10700 0.02033
9 42.69 51.34 1     9 0.132333267  0.11021 0.34668

[[2]]
      x     y z index square.disp       dx       dy
2 39.31 46.96 1     2          NA       NA       NA
4 42.95 48.65 1     4  0.18439416  0.38948  0.18083
6 41.32 47.89 1     6  0.03703179 -0.17441 -0.08132
8 42.13 50.90 1     8  0.11124077  0.08667  0.32207

> squareDisp(track, 8)
[[1]]
      x     y z index square.disp      dx      dy
1 39.88 46.11 1     1          NA      NA      NA
9 42.69 51.34 1     9   0.4035658 0.30067 0.55961

[[2]]
      x     y z index square.disp dx dy
2 39.31 46.96 1     2          NA NA NA

[[3]]
      x     y z index square.disp dx dy
3 41.94 47.86 1     3          NA NA NA

[[4]]
      x     y z index square.disp dx dy
4 42.95 48.65 1     4          NA NA NA

[[5]]
      x     y z index square.disp dx dy
5 42.66 47.91 1     5          NA NA NA

[[6]]
      x     y z index square.disp dx dy
6 41.32 47.89 1     6          NA NA NA

[[7]]
      x    y z index square.disp dx dy
7 41.66 48.1 1     7          NA NA NA

[[8]]
      x    y z index square.disp dx dy
8 42.13 50.9 1     8          NA NA NA

> squareDisp(track, 9)
Error in squareDisp(track, 9) :
track length:     9
dt:         9
Time interval (dt) greater than track length-1
```