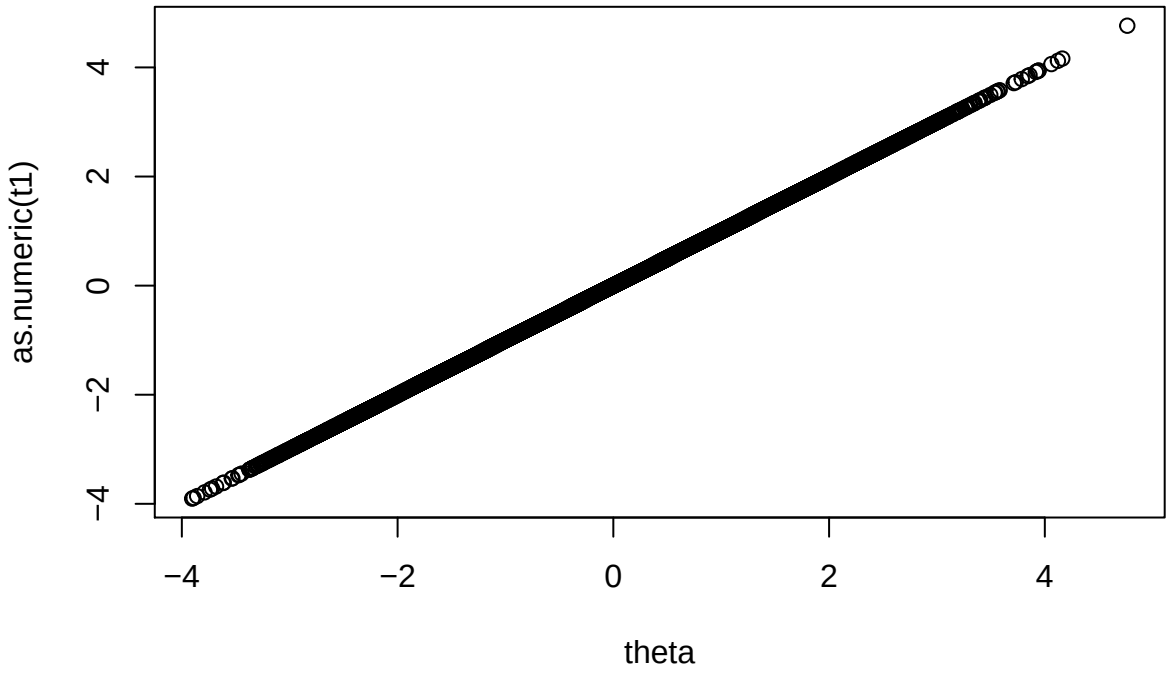
detrendr examples

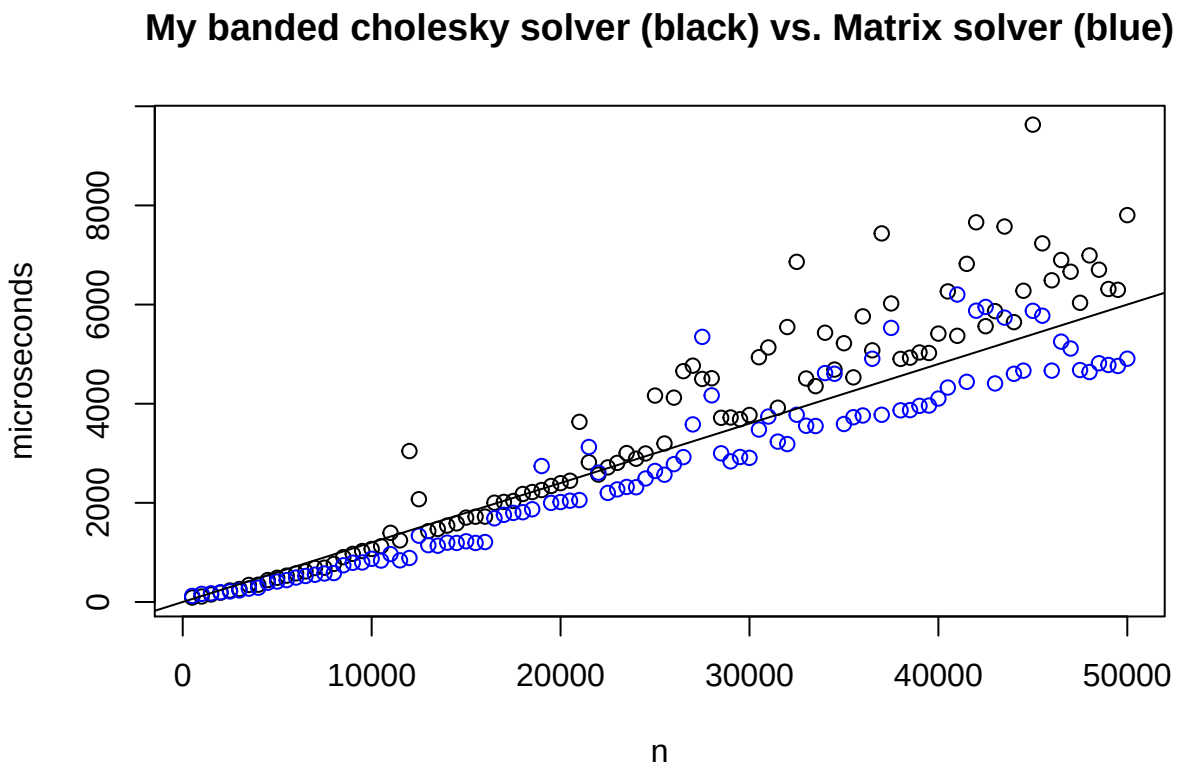
Compare cholesky decompositions and solvers as n increases

```
nSeq <- seq(500, 50000, 500)
k <- 3

timing <- data.frame(n=nSeq, Matrix_chol = NA, Matrix_solve = NA,
                     Eigen_chol = NA, chol_solve = NA)
i <- 1
for (n in nSeq){
  D <- get_Dk(n, k)
  M <-  as(Diagonal(n) + Matrix::crossprod(D), "dgCMatrix")
  theta <- rnorm(n)

  cholM <- chol(M)
  timing$Matrix_solve[i] <- median(microbenchmark(
    x1 <- as.numeric(Matrix::solve(cholM, Matrix::solve(t(cholM), theta))))$time*1e-3)
  timing$chol_solve[i] <- median(microbenchmark(
    x2 <- chol_solve(cholM, chol_solve(t(cholM), theta, k, FALSE),
                  k, TRUE))$time*1e-3)
  i <- i+1
}

t1 <- M%*%x1
plot(as.numeric(t1)~theta)
```



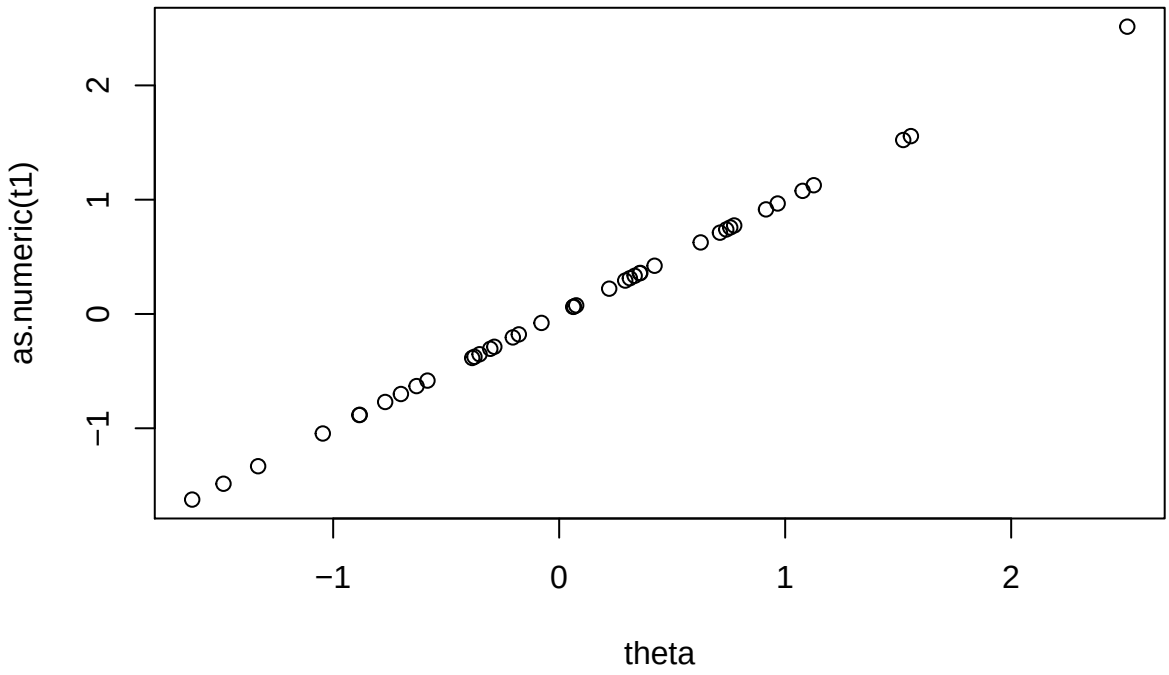

```
plot(chol_solve~n, timing, ylab = "microseconds",
     main="My banded cholesky solver (black) vs. Matrix solver (blue)")
points(Matrix_solve~n, timing, col="blue")
```

```
abline(0, .12)
```

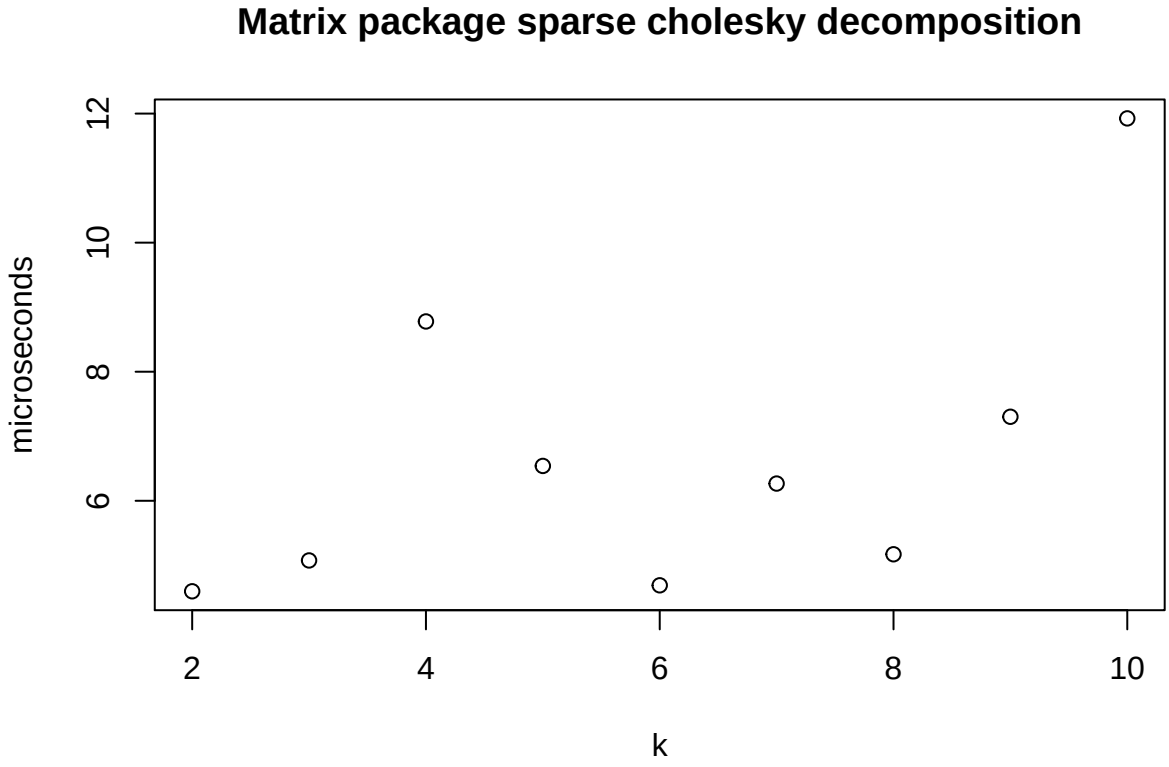


Compare cholesky decompositions and solvers as k increases

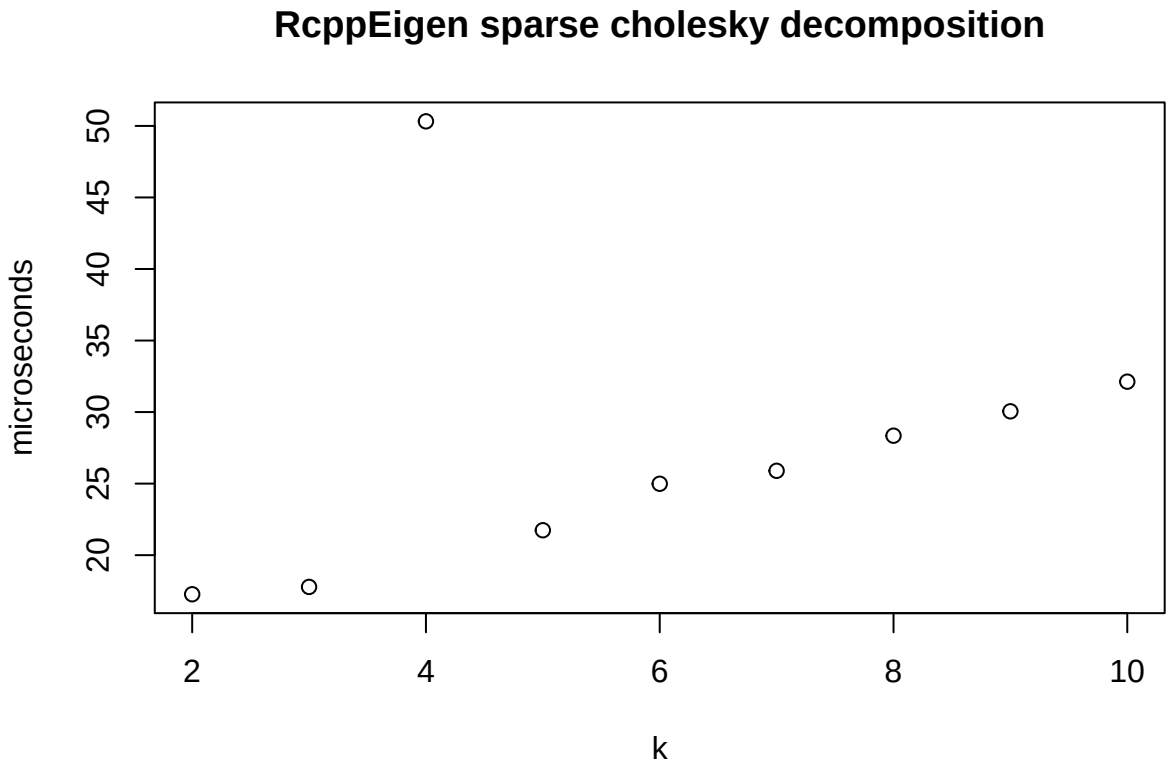
```
n <- 40
kSeq <- seq(2, 10, 1)

timing <- data.frame(k=kSeq, Matrix_chol = NA, Matrix_solve = NA,
                     Eigen_chol = NA, chol_solve = NA)
i <- 1
for (k in kSeq){
  D <- get_Dk(n, k)
  M <-  as(Diagonal(n) + Matrix::crossprod(D), "dgCMatrix")
  theta <- rnorm(n)

  timing$Matrix_chol[i] <- median(microbenchmark(cholM <- chol(M))$time*1e-3)
  timing$Eigen_chol[i] <- median(microbenchmark(cholM2 <- chol_eigen(M))$time*1e-3)
  timing$Matrix_solve[i] <- median(microbenchmark(
    x1 <- as.numeric(Matrix::solve(cholM, Matrix::solve(t(cholM), theta))))$time*1e-3)
  timing$chol_solve[i] <- median(microbenchmark(
    x2 <- chol_solve(cholM, chol_solve(t(cholM), theta, k, FALSE),
                     k, TRUE))$time*1e-3)
  i <- i+1
}

t1 <- M%*%x1
plot(as.numeric(t1)~theta)
```



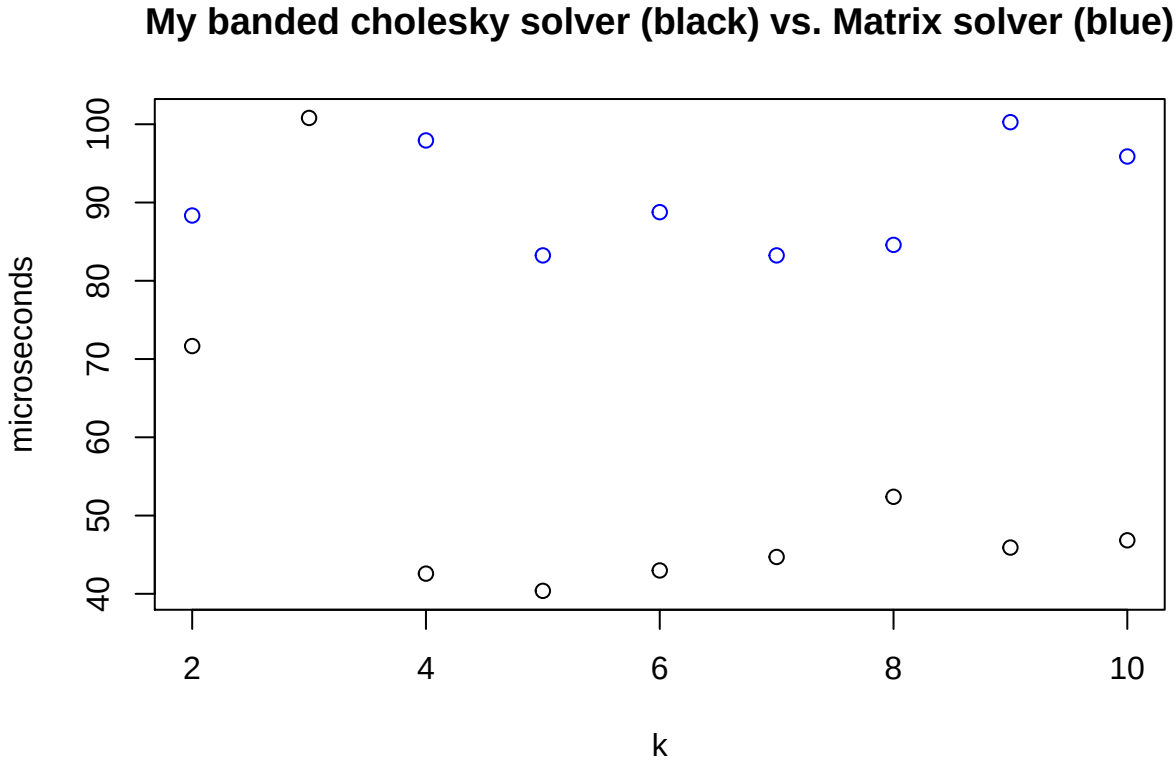

```
plot(Matrix_chol~k, timing, ylab="microseconds",
     main="Matrix package sparse cholesky decomposition")
```




```
plot(Eigen_chol~k, timing, ylab ="microseconds",
     main="RcppEigen sparse cholesky decomposition")
```




```
plot(chol_solve~k, timing, ylab = "microseconds",
     main="My banded cholesky solver (black) vs. Matrix solver (blue)")
points(Matrix_solve~k, timing, col="blue")
```



Algorithm order

```r
nSeq <- seq(500, 15000, 500)
k <- 3

timing <- data.frame(n=nSeq, multistep = NA)
i <- 1
for (n in nSeq){
  D <- get_Dk(n, k)
  M <-  as(Diagonal(n) + Matrix::crossprod(D), "dgCMatrix")
  theta <- rnorm(n)
  lambda0 <- 2*10^(-6)
  lambda <- lambda0*n^(k-1)/factorial(k-1)
  y <- rnorm(n)
  eta <- matrix(D%*%theta)
  tau <- 0.05
  cholM <- chol(M)
  timing$multistep[i] <- median(microbenchmark(
      multi_step <- spingarn_multi_step(theta, eta, y, D, cholM,
                                  lambda, tau, 1, 100, k)
  )$time*1e-9)

  i <- i+1
  print(i)
}
```

```
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
## [1] 21
## [1] 22
## [1] 23
## [1] 24
## [1] 25
## [1] 26
## [1] 27
## [1] 28
## [1] 29
## [1] 30
## [1] 31
```

```
plot(multistep~n, timing, ylab="Seconds", xlab = "n", main = "100 steps of algorithm")
abline(0, 2.95*1e-6)
```