Bandsolve for INLA

G. Nuel

28 June, 2024

The package `bandsolve` to solve efficiently $Ax = b$ where A is a symmetric band matrix. Complexity in $\mathcal{O}(nL)$ where L is the number of bands. Also usable to obtain the inverse of A (by solving with b base vectors).

```
#install.packages("remotes")
#remotes::install_github("Monneret/bandsolve")
require(bandsolve)
n=20
A=matrix(0,n,2)
A[,1]=3.0; A[-n,2]=-1.0
b=rep(1.0,n)
x=bandsolve(A,b)
norm(rot2mat(A)%*%x-b)
```

```
## [1] 3.552714e-15
```

```
#bandsolve_cpp(A,as.matrix(b))
```

When there is only one band off diagonal, the determinant of A can also be computed efficiently using the recursion:

$$D_i = A[i,1] \times D_{i+1} - A[i,2]^2 \times D_{i+2}$$

where $D_i = \det(A[i:n, i:n])$. Log-scale implementation possible.

The log of the determinant is also easy to obtain:

```
# log determinant
A=matrix(0,n,3)
A[,1]=3.0; A[-n,2]=-1.0; A[1:(n-2),3]=+1.0

bandlogdet=function(A) {
  # copy matrix to avoid in-place computation
  Amem=matrix(NA,nrow(A),ncol(A))
  Amem[]=A[]
  # call LDL
  invisible(LDL(Amem))
  # return res
  return(sum(log(Amem[,1])))
}

c(log(det(rot2mat(A))),bandlogdet(A))
```

```
## [1] 17.68803 17.68803
```