```
#
#
#
#
#
#
#
from casadi import *
import numpy
```

Let's construct a block diagonal structure

```
b1 = DM([[2,3],[4,5]])
b2 = DM([[6,7,8],[9,10,11],[12,13,14]])
A = diagcat(1,b1,b2,15)

print "original: "
```

```
original:
```

```
print A
```

```
[[1, 00, 00, 00, 00, 00, 00],
 [00, 2, 3, 00, 00, 00, 00],
 [00, 4, 5, 00, 00, 00, 00],
 [00, 00, 00, 6, 7, 8, 00],
 [00, 00, 00, 9, 10, 11, 00],
 [00, 00, 00, 12, 13, 14, 00],
 [00, 00, 00, 00, 00, 00, 15]]
```

Ruin the nice structure

```
numpy.random.seed(0)
p1 = numpy.random.permutation(A.size1())
p2 = numpy.random.permutation(A.size2())

S = A[p1,:]

print "randomly permuted: "
```

```
randomly permuted:
```

```
print S
```

```
[[00, 00, 00, 00, 00, 00, 15],
 [00, 4, 5, 00, 00, 00, 00],
 [00, 2, 3, 00, 00, 00, 00],
 [00, 00, 00, 6, 7, 8, 00],
 [1, 00, 00, 00, 00, 00, 00],
 [00, 00, 00, 12, 13, 14, 00],
 [00, 00, 00, 9, 10, 11, 00]]
```

```
nb, rowperm, colperm, rowblock, colblock, coarse_rowblock, coarse_colblock =
    S.sparsity().btf()

print "number of blocks: ", nb
```

```
number of blocks: 4
```

```
print "rowperm: ", rowperm
```

```
rowperm: [0, 1, 2, 3, 5, 6, 4]
```

```
print "colperm: ", colperm
```

```
colperm: [6, 1, 2, 3, 4, 5, 0]
```

```
print "restored:"
```

```
restored:
```

```
print S[rowperm,colperm]
```

```
[[15, 00, 00, 00, 00, 00, 00],
 [00, 4, 5, 00, 00, 00, 00],
 [00, 2, 3, 00, 00, 00, 00],
 [00, 00, 00, 6, 7, 8, 00],
 [00, 00, 00, 12, 13, 14, 00],
 [00, 00, 00, 9, 10, 11, 00],
 [00, 00, 00, 00, 00, 00, 1]]
```

```
print "rowblock: ", rowblock
```

```
rowblock: [0, 1, 3, 6, 7]
```

```
print "colblock: ", colblock
```

```
colblock: [0, 1, 3, 6, 7]
```

```
print "coarse_rowblock: ", coarse_rowblock
```

```
coarse_rowblock: [0, 0, 0, 7, 7]
```

```
print "coarse_colblock: ", coarse_colblock
```

```
coarse_colblock: [0, 0, 7, 7, 7]
```