btf

August 4, 2024

[1]:
```
from casadi import *
import numpy
```

Let's construct a block diagonal structure

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

[3]:
```
print("original: ")
print(A)
```

```
original:

[[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

[4]:
```
numpy.random.seed(0)
p1 = numpy.random.permutation(A.size1())
p2 = numpy.random.permutation(A.size2())
```

[5]:
```
S = A[p1,:]
#S = A[p1,p2]
```

[6]:
```
print("randomly permuted: ")
print(S)
nb, rowperm, colperm, rowblock, colblock, coarse_rowblock, coarse_colblock = S.
 ↪sparsity().btf()
```

```
randomly permuted:

[[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]]
```

[7]:
```
print("number of blocks: ", nb)
print("rowperm: ", rowperm)
print("colperm: ", colperm)
print("restored:")
print(S[rowperm,colperm])
print("rowblock: ", rowblock)
print("colblock: ", colblock)
print("coarse_rowblock: ", coarse_rowblock)
print("coarse_colblock: ", coarse_colblock)
```

```
number of blocks:  4
rowperm:  [0, 1, 2, 3, 5, 6, 4]
colperm:  [6, 1, 2, 3, 4, 5, 0]
restored:

[[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]]
rowblock:  [0, 1, 3, 6, 7]
colblock:  [0, 1, 3, 6, 7]
coarse_rowblock:  [0, 0, 0, 7, 7]
coarse_colblock:  [0, 0, 7, 7, 7]
```