

SCC

September 10, 2025

This file is part of CasADi.

CasADi -- A symbolic framework for dynamic optimization.
Copyright (C) 2010-2023 Joel Andersson, Joris Gillis, Moritz Diehl,
KU Leuven. All rights reserved.
Copyright (C) 2011-2014 Greg Horn

CasADi is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 3 of the License, or (at your option) any later version.

CasADi is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with CasADi; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

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

Let's construct a block diagonal structure

```
[2]: A = diagcat(1,DM([[2,3],[3,4]]),DM([[5,6,7],[6,8,9],[7,9,10]]),11)  
print(A)  
A.sparsity().spy()
```

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

*...

```

.**...
.**...
...***.
...***.
...***.
...*

```

```
[3]: numpy.random.seed(2)
```

We randomly permute this nice structure

```
[4]: perm = list(numpy.random.permutation(list(range(A.size1()))))
AP = A[perm,perm]
```

```
[5]: print(AP)
AP.sparsity().spy()
```

```

[[8, 00, 6, 00, 00, 9, 00],
 [00, 2, 00, 3, 00, 00, 00],
 [6, 00, 5, 00, 00, 7, 00],
 [00, 3, 00, 4, 00, 00, 00],
 [00, 00, 00, 00, 11, 00, 00],
 [9, 00, 7, 00, 00, 10, 00],
 [00, 00, 00, 00, 00, 00, 1]]
*.*.*.*.
.*.*...
*.*.*.*.
.*.*...
...*..
*.*.*.*.
...*

```

And use scc to recover the blocks

```
[6]: n,p,r = AP.sparsity().scc()
```

```
[7]: APrestored = AP[p,p]
```

```
[8]: print(APrestored)
APrestored.sparsity().spy()
print("# blocks: ", n)
print("block boundaries: ", r[:n])
```

```

[[8, 6, 9, 00, 00, 00, 00],
 [6, 5, 7, 00, 00, 00, 00],
 [9, 7, 10, 00, 00, 00, 00],
 [00, 00, 00, 2, 3, 00, 00],

```

```

[00, 00, 00, 3, 4, 00, 00],
[00, 00, 00, 00, 00, 11, 00],
[00, 00, 00, 00, 00, 00, 1]]
***...
***...
***...
...*. .
...*. .
...*.
...*
# blocks: 4
block boundaries: [0, 3, 5, 6]

```