

# sparsity\_jac

May 17, 2023

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 sparsity\_jac

```
[1]: from casadi import *  
     from numpy import *  
     import casadi as c  
     from pylab import spy, show
```

We construct a simple SX expression

```
[2]: x = SX.sym("x",40)  
     y = x[:-2]-2*x[1:-1]+x[2:]
```

Let's see what the first 5 entries of y look like

```
[3]: print(y[:5])
```

```
@1=2, [((x_0-(@1*x_1))+x_2), ((x_1-(@1*x_2))+x_3), ((x_2-(@1*x_3))+x_4),  
((x_3-(@1*x_4))+x_5), ((x_4-(@1*x_5))+x_6)]
```

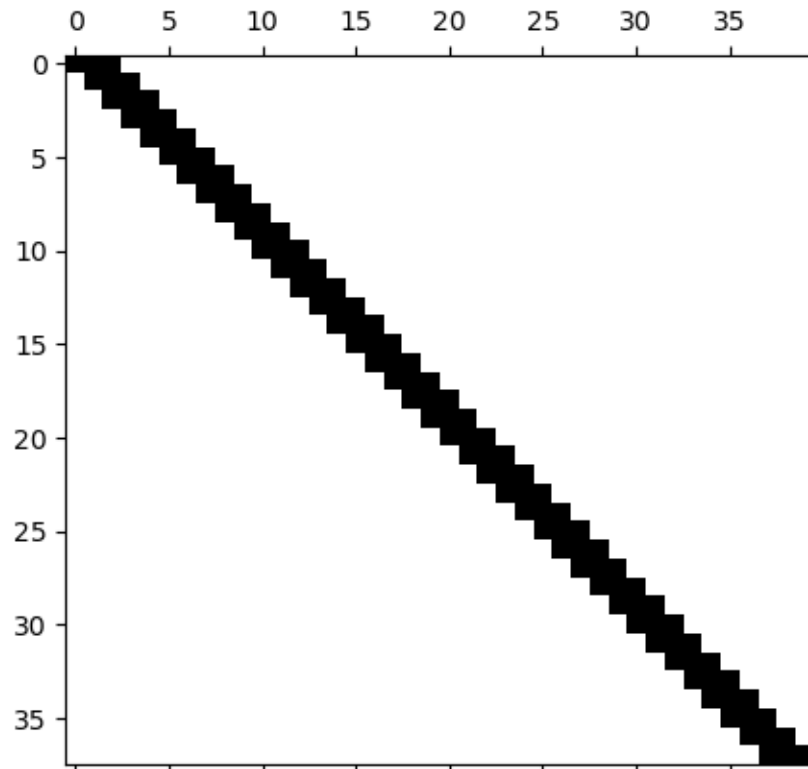
Next, we construct a function

```
[4]: f = Function("f", [x], [y])
```

And we visualize the sparsity of the jacobian

```
[5]: spy(f.sparsity_jac(0, 0))
```

```
[5]: <matplotlib.image.AxesImage at 0x7fc1a442ba60>
```



```
[6]: show()
```