

# matrix\_expand

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]: from casadi import *  
     from casadi.tools import *
```

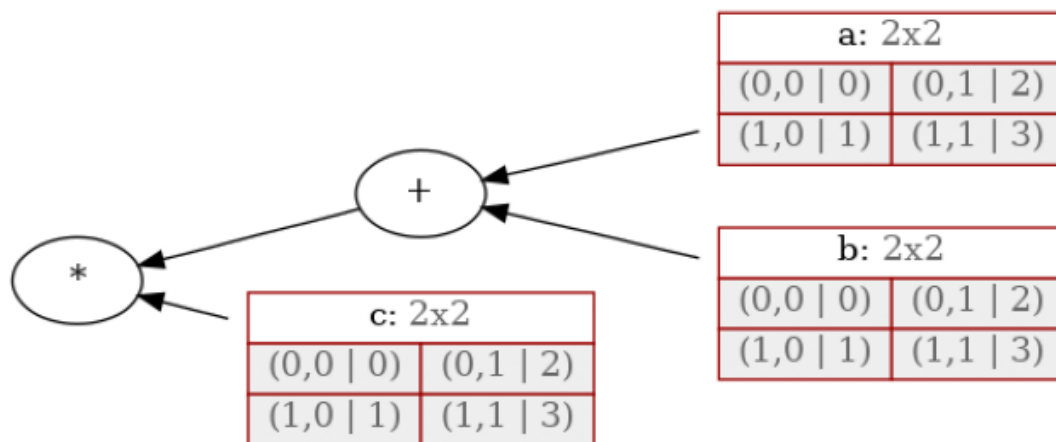
Let's revisit briefly the difference between SX and MX

```
[2]: a = MX.sym("a",2,2)  
     b = MX.sym("b",2,2)  
     c = MX.sym("c",2,2)
```

```
[3]: d = a+b  
     e = d*c
```

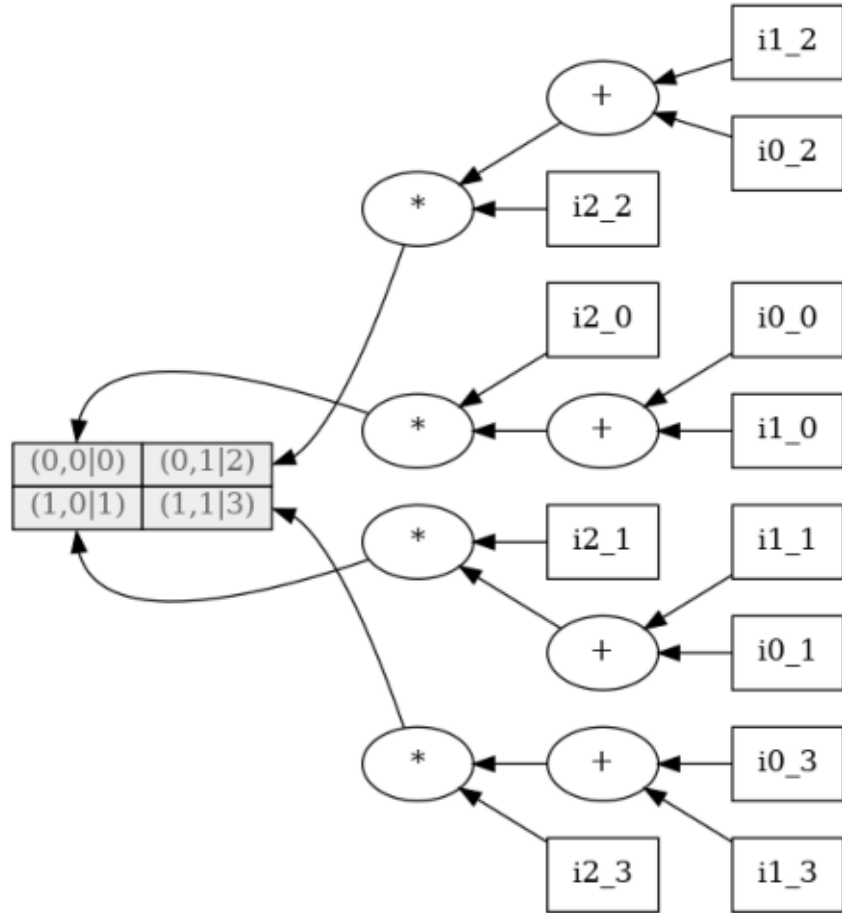
The element-wise addition and multiplication operators appear just as a single node in the MX expression graph

```
[4]: dotdraw(e)
```



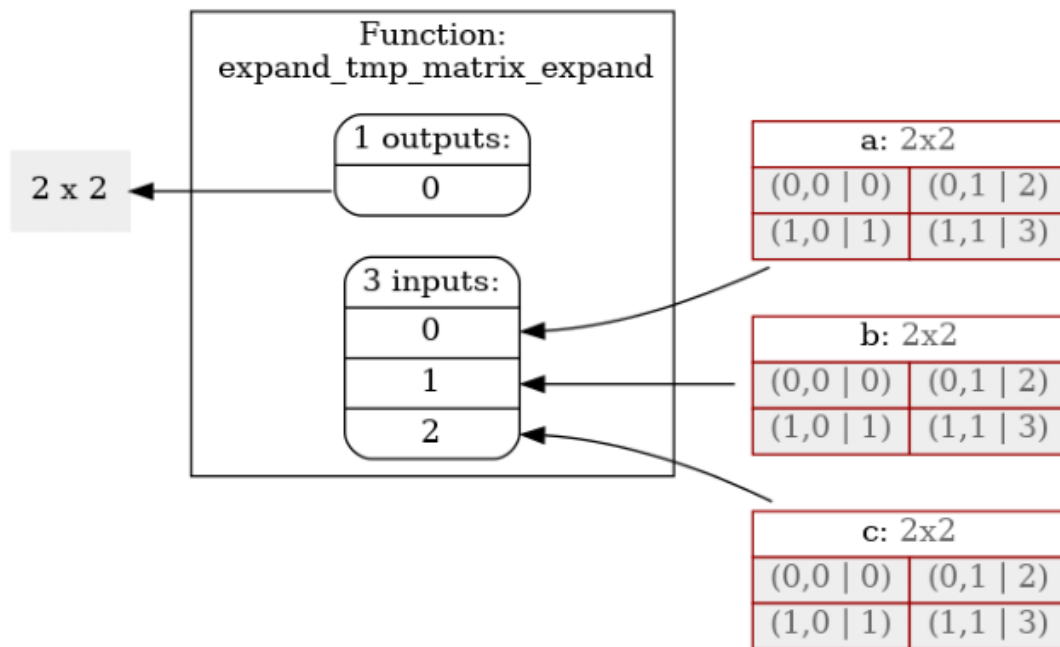
We can use `expand` to expand into subexpressions

```
[5]: f = Function("f", [a,b,c],[e])
      g = f.expand('g')
      dotdraw(g(*g.sx_in()))
```



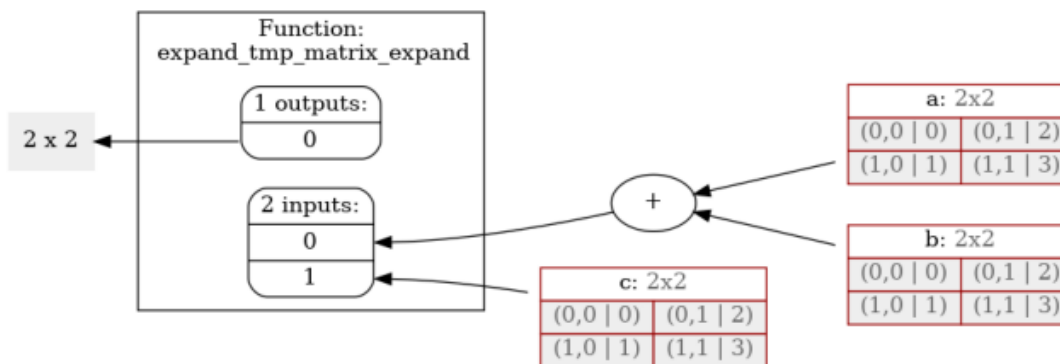
There is also a variant to perform expansion immediately on the MX graph. The expanded SX graph is hidden inside an SX graph call

```
[6]: dotdraw(matrix_expand(e))
```



An additional features of this variant is that one can choose which expressions remain outside of the expansion scope. In the following we list 'a+b=d' as a node on the boundary of expansion:

```
[7]: dotdraw(matrix_expand(e, [d]))
```



Note how the additions is not expanded, while the multiplication ended up in the expression