Sympy_DerivVectorGradients

October 2, 2020

[1]:
```
from sympy import diff, sin, exp, symbols, Function, Matrix,
    ↪MatrixSymbol, FunctionMatrix, derive_by_array


from sympy import Symbol


x, y, z = symbols('x y z')
f, g, h = list(map(Function, 'fgh'))
```

[2]:
```
# 1) manualy declaration of vector variables
xv = x,y,z
#f(xv).subs({x:1, y:2,z:3})
Matrix(xv)
```

[2]:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

[3]:
```
yv = [f(*xv), g(*xv), h(*xv)]; yv
```

[3]:
```
[f(x, y, z), g(x, y, z), h(x, y, z)]
```

[4]:
```
Matrix(yv)
```

[4]:

$$\begin{bmatrix} f(x, y, z) \\ g(x, y, z) \\ h(x, y, z) \end{bmatrix}$$

[5]:
```python
from sympy.abc import i,j

# 2) Dynamic way of declaring the vector variables
def var(letter: str, i: int) -> Symbol:
    letter_i = Symbol('{}_{}'.format(letter, i+1),
 ↪is_commutative=True)
    return letter_i

n,m,p = 5,7,4

xv = Matrix(n, 1, lambda i,j : var('x', i)); xv
```

[5]:
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

[6]:
```python
def func(i):
    y_i = Function('y_{}'.format(i+1))(*xv)
    return y_i

yv = Matrix( m, 1, lambda i,_:  func(i)); yv
```

[6]:
$$\begin{bmatrix} y_1(x_1, x_2, x_3, x_4, x_5) \\ y_2(x_1, x_2, x_3, x_4, x_5) \\ y_3(x_1, x_2, x_3, x_4, x_5) \\ y_4(x_1, x_2, x_3, x_4, x_5) \\ y_5(x_1, x_2, x_3, x_4, x_5) \\ y_6(x_1, x_2, x_3, x_4, x_5) \\ y_7(x_1, x_2, x_3, x_4, x_5) \end{bmatrix}$$

0.0.1 Gradient Vector

Let $f(\mathbf{x})$ be a differentiable real-valued function of the real $m \times 1$ vector $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}$.

Then the vector of first order partial derivatives $\frac{\partial f}{\partial \mathbf{x}}$, also called the gradient vector, is defined as:

$$\frac{\partial f}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_m} \end{pmatrix}$$

The vector of first order partial derivatives $\frac{\partial f}{\partial \mathbf{x}^T}$ is defined as:

$$\frac{\partial f}{\partial \mathbf{x}^T} = \left(\frac{\partial f}{\partial \mathbf{x}}\right)^T = \begin{pmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \cdots & \frac{\partial f}{\partial x_m} \end{pmatrix}$$

[7]:
```
# ### for deriv of scalar-valued multivariate function with
↪respect to the vector

f(*xv).diff(xv)
```

[7]:

$$\begin{bmatrix} \frac{\partial}{\partial x_1} f(x_1, x_2, x_3, x_4, x_5) \\ \frac{\partial}{\partial x_2} f(x_1, x_2, x_3, x_4, x_5) \\ \frac{\partial}{\partial x_3} f(x_1, x_2, x_3, x_4, x_5) \\ \frac{\partial}{\partial x_4} f(x_1, x_2, x_3, x_4, x_5) \\ \frac{\partial}{\partial x_5} f(x_1, x_2, x_3, x_4, x_5) \end{bmatrix}$$

[8]:
```
derive_by_array(f(*xv), xv)
```

[8]:

$$\begin{bmatrix} \frac{\partial}{\partial x_1} f(x_1, x_2, x_3, x_4, x_5) \\ \frac{\partial}{\partial x_2} f(x_1, x_2, x_3, x_4, x_5) \\ \frac{\partial}{\partial x_3} f(x_1, x_2, x_3, x_4, x_5) \\ \frac{\partial}{\partial x_4} f(x_1, x_2, x_3, x_4, x_5) \\ \frac{\partial}{\partial x_5} f(x_1, x_2, x_3, x_4, x_5) \end{bmatrix}$$

[9]:
```
assert Matrix(derive_by_array(f(*xv), xv)) == f(*xv).diff(xv)
```

0.0.2 Derivative of Vector with Respect to Scalar

Let $\mathbf{y}(x) = \begin{pmatrix} y_1(x) \\ y_2(x) \\ \vdots \\ y_m(x) \end{pmatrix}$ be a vector of order m, where each of the elements y_i are functions of the scalar variable x. Specifically, $y_i = f_i(x), 1 \leq i \leq m$, where $f_i : \mathbb{R} \to \mathbb{R}$ and $\mathbf{y} : \mathbb{R} \to \mathbb{R}^m$.

Then the derivative of the vector $\mathbf{y}$ with respect to scalar x is defined as:

$$\frac{\partial \mathbf{y}}{\partial x} = \begin{pmatrix} \frac{\partial y_1}{\partial x} & \frac{\partial y_2}{\partial x} & \dots & \frac{\partial y_m}{\partial x} \end{pmatrix}$$

[10]:
```
# ### for deriv of a vector-valued function by its scalar
↪argument
#yv = [f(x), g(x), h(x)]; yv
from sympy.abc import x

yv = Matrix( 1, m, lambda _, j:  Function('y_{}'.
↪format(j+1))(x)); yv
```

[10]: $\left[\begin{matrix}y_{1}(x) & y_{2}(x) & y_{3}(x) & y_{4}(x) & y_{5}(x) & y_{6}(x) & y_{7}(x)\end{matrix}\right]$

[11]:
```
yv.diff(x)

 # NOTE: incorrect shape (is column-wise, must be row-wise
↪like below) when defining the yv matrix to be m x 1 instead
↪of 1 x m. Ideally want to define a regular m x 1 y-vector of
↪functions y_i and to have the diff by x to be 1 x m.
```

[11]: $\left[\begin{matrix}\frac{d}{dx} y_{1}(x) & \frac{d}{dx} y_{2}(x) & \frac{d}{dx} y_{3}(x) & \frac{d}{dx} y_{4}(x) & \frac{d}{dx} y_{5}(x) & \frac{d}{dx} y_{6}(x) & \frac{d}{dx} y_{7}(x)\end{matrix}\right]$

[12]:
```
derive_by_array(yv, x) # Correct shape (row-wise)
```

[12]: $\left[\left[\begin{matrix}\frac{d}{dx} y_{1}(x) & \frac{d}{dx} y_{2}(x) & \frac{d}{dx} y_{3}(x) & \frac{d}{dx} y_{4}(x) & \frac{d}{dx} y_{5}(x) & \frac{d}{dx} y_{6}(x) & \frac{d}{dx} y_{7}(x)\end{matrix}\right]\right]$

[13]:
```
Matrix(derive_by_array(yv, x))
```

[13]: $\left[\begin{matrix}\frac{d}{dx} y_{1}(x) & \frac{d}{dx} y_{2}(x) & \frac{d}{dx} y_{3}(x) & \frac{d}{dx} y_{4}(x) & \frac{d}{dx} y_{5}(x) & \frac{d}{dx} y_{6}(x) & \frac{d}{dx} y_{7}(x)\end{matrix}\right]$

[14]:
```
assert Matrix(derive_by_array(yv, x)) == Matrix(yv).diff(x)
```

0.0.3 Vector Chain Rule

[15]:
```
# ### for vector chain rule
```