Python Quantitative Project

Ryan Greenup

August 2, 2020

Contents

§ 1 Matrix Exponentiation **1**

1.1 Implementation in Sympy 1

¶ 1 Theory 3

1.1 Matrix Exponentiation 3

1.1 Matrix-Matrix Exponentiation 5

¶ 1 An alternative Implementation in Sympy 6

§ 1 Matrix Exponentiation

1.1 Implementation in Sympy

The Matrix Exponential is implemented in areas of:

- Graph Centrality modelling [5]
- Systems of Linear Differential Equations [6, Ch. 8.4]
- Theory of Algebraic Lie Groups [2, Ch. 2]

However the method to implement matrix exponentiation provided by the documentation [3] and referenced in the development repository [4] does not appear to be implemented very well, for example the following provides a very long result:

Python

```python
from __future__ import division
from sympy import *
x, y, z, t = symbols('x y z t')
k, m, n = symbols('k m n', integer=True)
f, g, h = symbols('f g h', cls=Function)
init_printing()
init_printing(use_latex='mathjax', latex_mode='equation')


import pyperclip
def lx(expr):
    pyperclip.copy(latex(expr))
    print(expr)
```

Python

```python
A = Matrix([
    [11, 12, 13],
    [21, 22, 23],
    [31, 32, 33]
])

expr = exp(A)
expr.doit()
```

$$-\frac{1}{-\frac{33}{94}+\frac{5\sqrt{1149}}{94}}\Bigg(\frac{-\frac{6552}{-\sqrt{1149}-22}+552}{(-\sqrt{1149}-22)\left(-\frac{59\sqrt{1149}}{95}-\frac{1837}{95}\right)}-\frac{26}{-\sqrt{1149}-22}\Bigg)-\frac{1}{\frac{33\sqrt{1149}}{2303}+\frac{5745}{2303}}\Bigg(-1-\frac{1}{-\frac{33}{94}+\frac{5\sqrt{1149}}{94}}\Bigg(\frac{26}{-\sqrt{1149}-22}-\frac{-\frac{6552}{-\sqrt{1149}-22}+552}{(-\sqrt{1149}-22)\Big(-\frac{59\sqrt{1149}}{95}-1}$$

$$-2-\frac{1}{\left(-\frac{1837}{95}+\frac{59\sqrt{1149}}{95}\right)\left(\frac{33\sqrt{1149}}{2303}+\frac{5745}{2303}\right)e^{-33+\sqrt{1149}}}\Bigg(-1-\frac{1}{-\frac{33}{94}+\frac{5\sqrt{1149}}{94}}\Bigg(\frac{26}{-\sqrt{1149}-22}-\frac{-\frac{6552}{-\sqrt{1149}-22}+552}{(-\sqrt{1149}-22)\left(-\frac{59\sqrt{1149}}{95}-\frac{1837}{95}\right)}+2\Bigg)\Bigg)$$

$$-\frac{1}{-\frac{33}{94}+\frac{5\sqrt{1149}}{94}}\Bigg(\frac{-\frac{6552}{-\sqrt{1149}-22}+552}{(-\sqrt{1149}-22)\left(-\frac{59\sqrt{1149}}{95}-\frac{1837}{95}\right)}-\frac{26}{-\sqrt{1149}-22}\Bigg)-\frac{1}{\frac{33\sqrt{1149}}{2303}+\frac{5745}{2303}}\Bigg(-1-$$

Simplifying this result doesn't seem to help either:

Python

```python
simplify(expr)
```

$$\frac{1}{12(-1065889+33298\sqrt{1149})e^{\sqrt{1149}}}\Big(-8625947e^{33+2\sqrt{1149}}-2131778e^{\sqrt{1149}}-2032943e^{33}+74651\sqrt{1149}e^{33}+66596\sqrt{1149}e^{\sqrt{1149}}+258329\sqrt{1149}e^{3}$$

$$\frac{1}{6(-1065889+33298\sqrt{1149})e^{\sqrt{1149}}}\Big(-66949e^{33+2\sqrt{1149}}-66596\sqrt{1149}e^{\sqrt{1149}}-2064829e^{33}+61128\sqrt{1149}e^{33}+2131778e^{\sqrt{1149}}+5468\sqrt{1149}e^{33+}$$

$$\frac{1}{12(-1065889+33298\sqrt{1149})e^{\sqrt{1149}}}\Big(-236457\sqrt{1149}e^{33+2\sqrt{1149}}-6226373e^{33}-2131778e^{\sqrt{1149}}+66596\sqrt{1149}e^{\sqrt{1149}}+169861\sqrt{1149}e^{33}+8358151e^{3}$$

Methods suggested online only provide numerical solutions or partial sums:

- python - Sympy Symbolic Matrix Exponential - Stack Overflow
- python - Exponentiate symbolic matrix expression using SymPy - Stack Overflow
- Calculate state transition matrix in python - Stack Overflow

Instead this will need to be implemented from first principles.

¶ 1 Theory

1.1 Matrix Exponentiation

A Matrix Exponential is defined by using the ordinary exponential power series [2, Ch. 2],[6, Ch. 8.4] (should we prove the power series generally?):

$$e^{\mathbf{X}} = \sum_{k=0}^{\infty} \left[\frac{1}{k!} \cdot \mathbf{X}^{\mathbf{k}} \right] \tag{1}$$

This definition can be expanded upon however by using properties of logarithms:

$$\begin{aligned} b &= e^{\log_e(b)}, \quad \forall b \in \mathbb{C} &(2)\\ \implies b^{\mathbf{X}} &= \left(e^{\log_e(b)}\right)^{\mathbf{X}} &(3)\\ \implies b^{\mathbf{X}} &= e^{\log_e b \mathbf{X}} &(4) \end{aligned}$$

The identity in (2) is justified by the definition of the complex log. However some discussion is required for (3) because it is not clear that the exponential will generally distribute throught he parenthesis like so $(a \cdot b)^k = a^k \cdot b^k$, for example consider $\left([-1]^2 \cdot 3\right)^{\frac{1}{2}} \neq [-1]^{\frac{2}{2}} \cdot 3^{\frac{1}{2}}$.

A sufficient condition for this identity is $k \in \mathbb{Z}^*$, consider this example which will be important later:

$$\left(\log_e(b)\mathbf{X}\right)^k, \quad \forall k \in \mathbb{Z}^* \tag{5}$$

Because multiplication is commutative $\forall z \in \mathbb{C}$, this could be re-expressed in the form:

$$\begin{aligned} \left(\log_e(b)\mathbf{X}\right)^k &= \underbrace{\log_e(b) \cdot \log_e(b) \cdot \log_e(b) \ldots}_{k \text{ times}} \times \underbrace{\mathbf{XXX} \ldots}_{k \text{ times}} \\ &= \log_e^k(b)\mathbf{X}^k &(6) \end{aligned}$$

Now consider the the following by applying (6):

$$
\begin{aligned}
e^X &= \sum_{k=0}^{\infty}\left[\frac{1}{k!}\mathbf{X}^k\right] \\
\implies e^{bX} &= \sum_{k=0}^{\infty}\left[\frac{1}{k!}\left(b\mathbf{X}\right)^k\right] \quad \forall b\in\mathbb{C} \\
&= \sum_{k=0}^{\infty}\left[\frac{1}{k!}b^k\mathbf{X}^k\right] \\
&= \left(e^b\right)^{\mathbf{X}} \\
&\implies e^{b\mathbf{X}} = e^{\mathbf{X}b} = \left(e^b\right)^{\mathbf{X}} = \left(e^{\mathbf{X}}\right)^b \qquad \square \qquad (7)
\end{aligned}
$$

So the matrix exponential for an arbitrary base could be given by:

$$
\begin{aligned}
b = e^{\log_e(b)}, \quad \forall b\in\mathbb{C} \\
\implies b^{\mathbf{X}} = \left(e^{\log_e(b)}\right)^{\mathbf{X}}
\end{aligned}
$$

as per (7)

$$
\begin{aligned}
b^{\mathbf{X}} &= \sum_{k=0}^{\infty}\left[\frac{\left(\log_e(b)\mathbf{X}\right)^k}{k!}\right] \\
&= \sum_{k=0}^{\infty}\left[\frac{\log_e(b)}{k!}\mathbf{X}^k\right] \qquad (8)
\end{aligned}
$$

This is also consistent with the *McLaurin Series* expansion of $b^{\mathbf{X}}$ $(\forall b\in\mathbb{C})$:

$$
\begin{aligned}
f(x) &= \sum_{k=0}^{\infty}\left[\frac{f^{(n)}(0)}{k!}x^k\right] \\
\implies b^x &= \sum_{k=0}^{\infty}\left[\frac{\frac{\mathrm{d}^n}{\mathrm{d}x^n}\left(b^x\right)|_{x=0}}{k!}x^k\right] \\
\implies b^{\mathbf{X}} &= \sum_{k=0}^{\infty}\left[\frac{\frac{\mathrm{d}^n}{\mathrm{d}\mathbf{X}^n}\left(b^{\mathbf{X}}\right)|_{\mathbf{X}=\mathbf{O}}}{k!}\mathbf{X}^k\right]
\end{aligned}
$$

By ordinary calculus identities we have$f(x) = b^x \implies f^{(n)}(x) = b^x\log_e^n(b)$ which distribute through a matrix and hence:

$$
\begin{aligned}
b^x &= \sum_{k=0}^{\infty}\left[\frac{b^0\log_e^k(b)}{k!}x^k\right] \\
\implies b^{\mathbf{X}} &= \sum_{k=0}^{\infty}\left[\frac{b^0\log_e^k(b)}{k!}\mathbf{X}^k\right]
\end{aligned}
$$

By the previous identity:

$$
\begin{aligned}
\implies b^{\mathbf{X}} &= \sum_{k=0}^{\infty} \left[\frac{(\log_e(b)\mathbf{X})^k}{k!} \right] \\
&= e^{\log_e(b)\mathbf{X}}
\end{aligned}
$$

1.1 Matrix-Matrix Exponentiation

Matrix-Matrix exponentiation has applications in quantum mechanics [1, p. 84].
As for Matrices with the requirements:

1. Square
2. Normal:
   - Commutes with it's congugate transpose
3. Non Singular
4. Non Zero Determinant

$$
\begin{aligned}
||A - I|| < 1 &\implies e^{\log_e(\mathbf{A})} = \mathbf{A} \text{ (By Lie Groups Springer Textbook)} \\
&\implies \mathbf{A}^{\mathbf{B}} = \left(e^{\log_e(\mathbf{A})}\right)^{\mathbf{B}} \\
&\text{Similar justification as (7)} \\
&\implies \mathbf{A}^{\mathbf{B}} = e^{\log_e(\mathbf{A})\mathbf{B}}
\end{aligned}
$$

However the following identities are by **Definition** anyway: [1]

$$\mathbf{A}^{\mathbf{B}} = e^{\log_e(\mathbf{A})\mathbf{B}} \tag{9}$$

$$ {}^{\mathbf{B}}\mathbf{A} = e^{\mathbf{B}\log_e(\mathbf{A})} \tag{10}$$

¶ 1 An alternative Implementation in Sympy

Python

```python
def matexp(mat, base = E):
        """
        Return the Matrix Exponential of a square matrix
        """
        import copy
        import sympy
    # Should realy test for sympy vs numpy array
    # Test for Square Matrix
        if mat.shape[0] != mat.shape[1]:
            print("ERROR: Only defined for Square matrices")
            return
        m = zeros(mat.shape[0])
        for i in range(m.shape[0]):
            for j in range(m.shape[1]):
                m[i,j] = Sum((mat[i,j]*ln(base))**k/factorial(k), (k, 0, oo)).doit()
        return m
```

Python

```python
matexp(A, pi)
```

$$\begin{bmatrix} \pi^{11} & \pi^{12} & \pi^{13} \\ \pi^{21} & \pi^{22} & \pi^{23} \\ \pi^{31} & \pi^{32} & \pi^{33} \end{bmatrix}$$

Python

```python
A = Matrix([ [11,12,13], [21,22,23], [31,32,33] ])

  B = Matrix([
      [1,2,3],
      [4,5,6],
      [7,8,9]
  ])


  A**B
```

References

[1] I. Barradas and J. E. Cohen. “Iterated Exponentiation, Matrix-Matrix Exponentiation, and Entropy”. In: *Journal of Mathematical Analysis and Applications* 183.1 (Apr. 1, 1994), pp. 76–88. ISSN: 0022-247X. DOI: 10.1006/jmaa.1994.1132. URL: http://www.sciencedirect.com/science/article/pii/S0022247X84711322 (visited on 08/02/2020) (cit. on p. 5).

[2] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Second edition. Graduate Texts in Mathematics 222. OCLC: ocn910324548. Cham ; New York: Springer, 2015. 449 pp. ISBN: 978-3-319-13466-6 (cit. on pp. 1, 3).

[3] *Matrices — SymPy 1.6.1 Documentation*. July 2, 2020. URL: https://docs.sympy.org/latest/tutorial/matrices.html (visited on 08/02/2020) (cit. on p. 1).

[4] *Matrix Exponential · Issue #6218 · Sympy/Sympy*. Feb. 9, 2019. URL: https://github.com/sympy/sympy/issues/6218 (visited on 08/02/2020) (cit. on p. 1).

[5] Laurence A. F. Park and Simeon Simoff. “Power Walk: Revisiting the Random Surfer”. In: *Proceedings of the 18th Australasian Document Computing Symposium*. ADCS ’13. Brisbane, Queensland, Australia: Association for Computing Machinery, Dec. 5, 2013, pp. 50–57. ISBN: 978-1-4503-2524-0. DOI: 10.1145/2537734.2537749. URL: http://doi.org/10.1145/2537734.2537749 (visited on 07/31/2020) (cit. on p. 1).

[6] Dennis G Zill and Michael R Cullen. *Differential Equations*. 7th ed. Brooks/Cole, 2009 (cit. on pp. 1, 3).