Google Summer of Code 2012 - Sympy

Guru Devanla
Mentor: Dr. Brian Granger, Physics Professor, Cal State

Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607, USA
gdevan2@cs.uic.edu

May 23, 2012

What is Sympy

- A Python library for *symbolic mathematics*
- Has a goal of becoming a full featured *Computer Algebra System*
- Written entirely in Python and does not require external libraries

[1] : http://sympy.org/en/index.html

Sympy 0.7.1 - Features

- Core Capabilities

Sympy 0.7.1 - Features

- Core Capabilities
- Other Modules
  - Polynomials
  - Calculus
  - Solving Equations
  - Discrete Math
  - Matrices
  - ....
  - Physics
    - Mechanics
    - Quantum : Quantum Mechanics, Qubits, Quantum Gates etc.
    - (GSOC 2012 : Density Operator)

Sympy 0.7.1 - Features

- Core Capabilities
- Other Modules
  - Polynomials
  - Calculus
  - Solving Equations
  - Discrete Math
  - Matrices
  - ....
  - Physics
    - Mechanics
    - Quantum : Quantum Mechanics, Qubits, Quantum Gates etc.
    - (GSOC 2012 : Density Operator)
- Printing
  - LaTeX, MathML etc

Examples

```
$ cd sympy
$ ./bin/isympy
IPython console for SymPy 0.7.1 (Python 2.7.1) (ground types: gmpy)

These commands were executed:
>>> 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)

Documentation can be found at http://www.sympy.org

In [1]: (1/cos(x)).series(x, 0, 10)
Out[1]:
     2      4       6        8
    x    5*x    61*x    277*x
1 + -- + ---- + ----- + ------ + O(x**10)
    2     24     720     8064
```

source: http://sympy.org

Examples

Matrices are created as instances from the Matrix class:

```
>>> from sympy import Matrix
>>> Matrix([[1,0], [0,1]])
[1  0]
[    ]
[0  1]
```

you can also put Symbols in it:

```
>>> x = Symbol('x')
>>> y = Symbol('y')
>>> A = Matrix([[1,x], [y,1]])
>>> A
[1  x]
[    ]
[y  1]

>>> A**2
[x*y + 1    2*x  ]
[                ]
[  2*y    x*y + 1]
```

source: http://sympy.org

Examples

Matrices are created as instances from the Matrix class:

```
>>> from sympy import Matrix
>>> Matrix([[1,0], [0,1]])
[1  0]
[    ]
[0  1]
```

you can also put Symbols in it:

```
>>> x = Symbol('x')
>>> y = Symbol('y')
>>> A = Matrix([[1,x], [y,1]])
>>> A
[1  x]
[    ]
[y  1]

>>> A**2
[x*y + 1    2*x  ]
[                ]
[  2*y    x*y + 1]
```

source: http://sympy.org

Examples

```
>>> from sympy import *
>>> x, y = symbols('x,y')
```

You can integrate elementary functions:

```
>>> integrate(6*x**5, x)
 6
x
>>> integrate(sin(x), x)
-cos(x)
>>> integrate(log(x), x)
x*log(x) - x
>>> integrate(2*x + sinh(x), x)
 2
x  + cosh(x)
```

source: http://sympy.org

Examples

LaTeX printing

```
>>> from sympy import Integral, latex
>>> from sympy.abc import x
>>> latex(x**2)
x^{2}
>>> latex(x**2, mode='inline')
$x^{2}$
>>> latex(x**2, mode='equation')
\begin{equation}x^{2}\end{equation}
>>> latex(x**2, mode='equation*')
\begin{equation*}x^{2}\end{equation*}
>>> latex(1/x)
\frac{1}{x}
>>> latex(Integral(x**2, x))
\int x^{2}\,dx
```

source: http://sympy.org

Sympy - Quantum Module

Informally , in quantum mechanics, *states* in a closed system are represented as linear combination of a set of basis vectors,

$$\mathit{State} : \left[\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right]$$

where

$(\frac{1}{\sqrt{2}})^2 = 0.5$ gives the probability of observing either state.

Mixed States

Say we have 2 states, given by

$$State_1 : \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

with $Pr(State_1) = \frac{1}{3}$

$$State_2 : \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \end{pmatrix} - \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

with $\mathrm{Pr}(State_2) = \frac{2}{3}$

Mixed States

Say we have 2 states, given by

$$State_1 : \frac{1}{\sqrt{2}} \binom{1}{0} + \frac{1}{\sqrt{2}} \binom{0}{1}$$

with $Pr(State_1) = \frac{1}{3}$

$$State_2 : \frac{1}{\sqrt{2}} \binom{1}{0} - \frac{1}{\sqrt{2}} \binom{0}{1}$$

with $\mathrm{Pr}(State_2) = \frac{2}{3}$

Represented as

$$((state_1, p_1), (state_2, p_2)....(state_n, p_n))$$

This representation becomes quite cumbersome.

Density Matrices

Density Matrix(ρ) given by

`sum of outer products of each of these states`

$$\rho = \sum_{i=1}^{n} p_i(state_i).(p_i(state_i))^*$$

Density Matrices

Density Matrix(ρ) given by

`sum of outer products of each of these states`

$$\rho = \sum_{i=1}^{n} p_i(state_i).(p_i(state_i))^*$$

This gives us a nice matrix.

- Helps in representation
- Used to evaluate various properties about systems in mixed states

GSoC 2012 - Project Goals

- Methods to declare density matrices
- Extend operators for time dependent states
- Way to use these objects in equations representing quantum circuits
- Integrate with current eval() functions to work with other quantum expressions

Tools/Workflow

- Python 2.7
- IPython as interactive python shell
- Emacs powered with python-mode, pymacs and pyrope
- git and github

live.sympy.org