Symbolic math in python

Posted March 01, 2013 at 07:07 PM | categories: symbolic, math | tags: | View Comments

Updated March 03, 2013 at 12:21 PM

Matlab post Python has capability to do symbolic math through the sympy package.

1 Solve the quadratic equation

```
from sympy import solve, symbols, pprint

a,b,c,x = symbols('a,b,c,x')

f = a*x**2 + b*x + c

solution = solve(f, x)
print solution
pprint(solution)
```

```
>>> >>> >>> >>> >>> >>> [(-b + (-4*a*c + b**2)**(1/2))/(2*a), -(b + (-4*a*c +
b**2)**(1/2))/(2*a)]
_____________    /        _____________\
        /          2     |      /          2 |
 -b + \/   -4*a*c + b    -\b + \/   -4*a*c + b   /
[---------------------, -----------------------]
          2*a                      2*a
```

The solution you should recognize in the form of \(\frac{b \pm \sqrt{b^2 - 4 a c}}{2 a}\) although python does not print it this nicely!

2 differentiation

you might find this helpful!

```
from sympy import diff

print diff(f, x)
print diff(f, x, 2)

print diff(f, a)
```

```
>>> 2*a*x + b
2*a
>>> x**2
```

3 integration

```
from sympy import integrate

print integrate(f, x)           # indefinite integral
print integrate(f, (x, 0, 1))  # definite integral from x=0..1
```

```
>>> a*x**3/3 + b*x**2/2 + c*x
a/3 + b/2 + c
```

4 Analytically solve a simple ODE

```
from sympy import Function, Symbol, dsolve
f = Function('f')
x = Symbol('x')
fprime = f(x).diff(x) - f(x) # f' = f(x)

y = dsolve(fprime, f(x))

print y
print y.subs(x,4)
print [y.subs(x, X) for X in [0, 0.5, 1]] # multiple values
```

```
>>> >>> >>> >>> >>> >>> f(x) == exp(C1 + x)
f(4) == exp(C1 + 4)
[f(0) == exp(C1), f(0.5) == exp(C1 + 0.5), f(1) == exp(C1 + 1)]
```

It is not clear you can solve the initial value problem to get C1.

The symbolic math in sympy is pretty good. It is not up to the capability of Maple or Mathematica, (but neither is Matlab) but it continues to be developed, and could be helpful in some situations.

org-mode source

Tweet

blog comments powered by Disqus

An improvement for figures in ipython + scimax

Using results from one code block in another org-mode

2018 in a nutshell for the Kitchin Research Group

Line integrals in Python with autograd

Using autograd for error propagation

Latest GitHub Repos

@jkitchin on GitHub.

f18-06623 *(Jupyter Notebook)*
Fall 2018 - Mathematical Modeling of Chemical Engineering Processes

aenet *(Fortran)*
Atomic interaction potentials based on artificial neural networks

ghub *(Emacs Lisp)*
Minuscule client library for the Github API

jmax *(Emacs Lisp)*
Johns customizations to maximize emacs

example-project
An example project

emacs-win *(Emacs Lisp)*
An emacs distribution for use with jmax

autograd *(Python)*
Efficiently computes derivatives of numpy code.

emacs-modules *(C)*
Dynamic modules for emacs

htp *(Python)*
high-throughput tools for python - ssh and rsync

amp-tutorial *(TeX)*
null

emacs-ffi *(C)*
FFI for Emacs

f16-06625 *(TeX)*
Chemical and Reactive Systems for Fall 2016

aiche-2016 *(HTML)*
Expanding Molecular Simulation Use by Data/Code Sharing in Scientific Publishing

dft-book *(Python)*
A book on modeling materials using VASP, ase and vasp

gitolite *(Perl)*
: Hosting git repositories -- Gitolite allows you to setup git hosting on a central server, with very fine-grained access control and many (many!) more powerful features.

criticmarkup-emacs *(Emacs Lisp)*
: Emacs minor mode for handling CriticMarkup.

jasp *(Python)*
: python enhancements of ase.calculators.vasp

hyve *(Hy)*
: My Hy code

hy *(Python)*
: A dialect of Lisp that's embedded in Python

lispy *(Emacs Lisp)*
: vi-like Paredit

ACS-2016-data-sharing *(HTML)*
: My talk at the Spring ACS 2016 meeting

jedi *(Python)*
: Awesome autocompletion and static analysis library for python.

f15-06625 *(Emacs Lisp)*
: 06-625 Chemical and Reactive Systems

hy-mode *(Emacs Lisp)*
: Hy mode for Emacs

elpa
: ELPA repository for my emacs packages

box-course *(Python)*
: Python wrappers for the box.com v2 API

blogofile_blog *(Python)*
: A Blogofile blog plugin

blogofile *(Python)*
: A static website compiler and blog engine written in Python.

dft-course *(Emacs Lisp)*
: Course repository for 06-640 - Molecular simulation

RSS Comments RSS Feed.