数学软件Sagemath中变量的基本应用

闫鹏

1 摘要

变量是数学中的基本概念，在初等数学里，变量是一个用来表示值的符号（一般为拉丁字母），该值可以是随意的，也可能是未指定或未定的。在代数运算时，将变量当作明确的数值代入运算中。变量这个概念在微积分中也很重要。一般，一个函数 y = f(x) 会包含两个变量，参数 x 和值 y。这也是“变量”这个名称的由来，当参数“变动”时，值也会相对应地“变动”。另外在更深的数学中，变量也可以只代表某个数据，一般为数字，但也可能为向量、矩阵或函数等数学物件。
十六、十七世纪，欧洲封建社会开始解体，代之而起的是资本主义社会。由于资本主义工场手工业的繁荣和向机器生产的过渡，以及航海、军事等的发展，促使技术科学和数学急速向前发展。原来的初等数学已经不能满足实践的需要，在数学研究中自然而然地就引入了变量与函数的概念，从此数学进入了变量数学时期。它以笛卡儿的解析几何的建立为起点(1637年)，接着是微积分的兴起。sage的特点

1. 软件合集，Sage（用Python和Cython实现的）将所有专用的数学软件集成到一个通用的接口,包括有C 、C++、Fortran和Python编写的大量现成的大型开源数学软件可用,比如Maxima，SymPy，GiNaC，分开学习这些软件将花费大量时间和精力，通过Sagemath提供的统一命令，用户只需要了解Python，即可使用这些软件。

2. 云计算，sage可以运行于本地客户端，利用终端或者浏览器界面使用，也可以在线使用（ Sage的在线版本，地址是 sagenb.org 或 https://cloud.sagemath.com），避免下载过G的文件，也有利于移动使用，除了访问官方服务器，用户自己也可以架设本地服务器，供局域网内使用，加快访问和计算速度。

3. 开源软件，其代码开放，提供可供检查的代码增强其权威性，代码也可由用户创建和改造，满足不同用户对于性能和功能的需求。

sage中定义一个变量可以有两种方式： 第一种是通过赋值的方法定义一个变量，该变量的值是指定的，比如，等式的左边是符号变量的名称，右边是数字或表达式，其值返回给符号变量。Python动态类型的语言，符号变量可以多次赋值，并可以傅以不同类型的值，比如，在sage中可以通过type（）得到符号变量的类型，这些类型有。。。。需注意sage的符号变量区分大小写，每个符号变量仅存在于当前的工作表单（引申），如果当前的工作表单被关闭后重开，则应该再次执行赋值。

```
sage: a1=2
sage: a2=3
sage: sum=a1+a2

sage: sum
```

也可以把多个语句放在一行，语句之间使用分号隔开。

```
sage: a1=2;a2=3;sum1=a1+a2;sum2=a1-a2

sage: sum1;sum2
```

更复杂的语句包含不同操作运算符，如下例子

```
sage: a1=2
sage: a2=2^2+3*4+(a1>5)
```

运算符执行先后顺序如下:

运算符	说明
or	逻辑或
and	逻辑与
not	逻辑非
in, not in	隶属关系
is, is not	类型检测
>, <=, >, >=, ==, !=, <>	比较
+, -	加、减
*, /, %	乘、除、余
**, ^	指数

条件运算符(==,<>,!=,<,<=,>,>=)，其返回值是布林值，即True或者False，以数值表示为1、0，布林运算符（not，and，or或与非运算符），其返回值是True或者False，以数值表示则为1、0。

```
sage: a2
```

第二种是使用var函数定义未赋值的符号变量，未赋值的符号变量在其定义区间内的值是任意的，

默认情况下，使用var函数定义的变量是复数范围内的，注意在Sage中x是已经定义好的未赋值符号变量，可以直接使用。同时Sage定义了一个变量“_”用来接受上一次运算的结果。

```
var('y,z') % y z
z=x+y
_+y
```

reset函数可以取消定义，用法$reset('y,z')$,但变量x取消后依然存在。
含有符号变量的表达式又叫符号表达式，如$z = x + y$，$x + y$即为符号表达式（Symbolic Expression），我们可以对符号表达式中的变量赋值：

```
z(x=2,y=3)
```

如果符号表达式只有一个变量，则可以直接赋值。

```
z=2x
z(5)
```

不同符号表达式可以进行运算，比如

```
z=(x-y)*(x+y)
```

利用expand()函数我们可以对其展开

```
expand(z)
```

也可以通过Python函数调用的方法来展开

```
z.expand()
```

结果为$x^2 - y^2$,函数或函数调用并不会影响z本身的属性，此时z依然为$(x + y)(x - y)$，现在我们定义$f = z.expand()$。

```
sage: f=z.expand()
```

利用$factor()$函数可以求取公因式$factor(f)$或$f.factor()$,factor函数同样可以分解数字。

```
sage: factor(20)
```

对于分式分解，我们可以利用factor函数求出公因式，然后自行分解，也可以利用“partial_fraction”函数来分解。

```
sage: z=1/ x^2-1)
sage: z.partial_fraction()
```

现在我们来看两个表达式

```
sage: z1=x^2+3*x+1
sage: z2=x^2+x+1
```

多项式所在的环影响它的性质。因此对上面两个表达式进行因式分解，得到本身。可以发现，z1==0在实数范围内有解，z2==0在复数范围内有解，即这两个多项式可以通过先求其解来进行分解，当然我们也可以通过指定多项式其所在的“环”，然后用factor函数来分解 最简单的方法是

```
sage: R.<x>=CC[]
```

这里R定义一个环，x定义一个域，域的区间为CC（复数域）QQ，RR等等

```
sage: z1.factor();z2.factor()
```

对于z1,我们也可以仅将其变量定义在实数域中。

```
sage: R.<x>=RR[]
```

需要指出，环由变量确定。相同域上的同一变量得到的环是等价的。

```
sage: R.<x>=QQ[];Z.<x>=QQ[]
sage: R==Z
```

同样，我们可以直接在定义向量和矩阵时指定其所在的环。

```
sage: v = vector(QQ, (1,2,3))
```

矩阵

```
sage: Z = matrix(ZZ, [[2,0], [0,1]])
```

多项式化简
在Sage中多项式化简是自动的

```
sage: var(`y')
sage: z=x+y-y
sage: z
```

使用Python的id函数可以查看变量在内存中的位置

```
sage: id(z);id(x)
```

可见变量z的保存位置和变量x相同，当你输入一个多项式时，如果其可以简单化简，则是多项式实际是按照化简之后的形式保存的。 如果多项式不能直接化简，我们可以对其手动化简，这时用到simplify系列函数

```
sage: z = x^2-1)^(1/2)/ (x+1)^(1/2);show(z)
```

可以看出这个式子可以化简，这里用到$simplify_radical$函数

```
sage: z.simplify_radical();show(_)
```

simplify系列函数有很多个，在Sage中输入simplify，然后按下Tab键即可查看，对于每个函数，在函数名后输入？可以查看该函数的帮助信息，输入？？可以查看该函数的源代码。
我们可以通过solve（）函数求的方程的解

```
sage: var('a)
```

求变量x;求变量a

```
sage: solve(x^2==a^2,x);solve(x^2==a^2,a)
```

结果以列表的形式出现，列表是Python的内建数据类型，形式为[exp1, exp2, exp3,...] 方程组的求解也很容易

```
sage: var('a,b,c')
sage:  solve([2*a +b- c==0, 3*b- c==0,a
+b==5],a,b,c)
```

这里需注意，在Python语言中，除法公式5/2结果为2，这是因为5和2是整你形，所以其运算结果也是整形，Sage则返回5/2本身，不做改变，通过类型提升我们可以得到5/2的数字解

```
sage: float(5/2)
```

sage也提供n()函数，使用方法是n(5/2)或5/2.n(),n()中可以指定精度和位数，注意这里的精度并不是精确到小数点后面几位，而是浮点数中尾数的存储位数，默认Sage使用53bits。n(5/3,prec=30,digits=8)指定5/3的结果精度为5，有8位有效数字

符号函数 在Python中元组通过圆括号中用逗号分割的项目定义。它和列表，字典同属序列，与列表不同，元祖中的对象既不能改变，也不能赋值。Sage引入了数学语法来定义函数，因此我们可以这样定义函数 f(x)=2*x 在这种情况下，括号中的x为函数f的参数，我们可以对其赋值：

```
sage: a1=2
```

在表达式z=x+y中，实际上我们定义了z(x,y)=x+y

```
sage: a2=3

sage: sum=a1+a2

sage: sum

5
```

我们通常想得到一个函数的图像，使用$plot$函数可以方便的做出二维图形，这里有一个函数$f(x) = x^2 - 10$ plot(f(x),-10,10),这里-10,10为变量x的取值范围的闭区间，即为坐标轴中x的范围

Sage也能创建三维图像，显示三维图像默认都是调用开源软件包 [Jmol]，它支持使用鼠标旋转和缩放图像（需要Java运行环境）。这里有一个函数f(x,y)=x-sin(y) ??当我们使用$z = plot(x^2, -1, 1)$时，这个过程只是定义变量z，并不会将结果显示出来，这里我们用到show函数。??show函数还可以显示手写格式的数学公式，这时实际是调用latex，上面我们已经用到了show函数。

```
z = x^2-1)^(1/2)/(x+1)^(1/2);show(z)
```

也可以把多个图像一起做图： ??我们看到f1和f2在[-10,10]的范围内有交点，现在我们求该交点，求交点的方法即是解方程的方法,上文我们用solve函数解代数方程，但是并不是所有的方程都有代数解，在这里我们$find_root$函数在区间内找到它的数值解。

```
f=f1(x)-f2(x)==0
f.find_root(0,10);f.find_root(-10,0)
```

$find_root$每次只会在给定区间内求出一个解，因此结合图像，我们可以细分区间，求出所有的交点。在代数运算时，将变量当作明确的数值代入运算中。

微积分 变量在高等数学中有着非常重要的应用，以微积分为例 使用diff函数可以求函数的导数，微分和偏微分,基本格式为diff(函数，*偏微分变量，*微分阶次） ??泰勒级数（Taylor series）用无限项连加式——级数来表示一个函数，这些相加的项由函数在某一点的导数求得。洛朗级数是泰勒级数的推广，它不仅包含了正数次数的项，也包含了负数次数的项。有时无法把函数表示为泰勒级数，但可以表示为洛朗级数。使用talyor函数可以快速的求出函数在某点上的泰勒或洛郎级数，下面是个例子。 ??函数x$^2x = 1$， -1，

使用integral函数可以求函数的积分，定积分和不定积分，基本格式是integral(函数，积分变量，*积分范围），对于高阶和多元函数的积分，可以通过叠加使用integral函数，注意在Matlab中int为积分符号，而在Python中int为取整，int(pi)=3. ??现在我们看这样的一个函数int(sin(x)e$^-(st))$??$int(sin(x)e^-(st))sin$（x），，，$Sage$，$laplac$

傅里叶变换是另一种一种线性的积分变换，常在将信号在时域（或空域）和频域之间变换时使用。

Sage中并没有直接提供相应的函数支持，但我们可以根据傅里叶函数的形式使用积分来计算，有时候，根据拉普拉斯变换与傅里叶变换的相似性，我们还可以套用拉普拉斯变换来求傅里叶变换。实际上，由于傅里叶变换在物理学和工程学中的广泛应用，有很多专门的文献、资料涉及傅里叶变换，对于Sagemath和傅里叶变换的结合，读者可以参考Computational Fourier Transforms 一书，本文不再涉及，后续我也会通过专门的文章来探讨这个问题。

微分方程包括常微分方程和偏微分方程，两者都可以再分为线性和非线性两类，线性包括齐次和非齐次，使用desolve函数可以解一介和二阶线性微分方程。Solves a 1st or 2nd order linear ODE via maxima. Including IVP and BVP. 我们以desolve的帮助文档中的例子为例。

一阶微分方程 ??这里用到了Sage调用Maxima的接口，所以它的输出看上去与其他Sage的输出略有不同。可以通过$.show()$，？。，$ics -(optional)theinitialorboundaryconditionsC$??（）$http://wiki.sagemath.org/DifferentialEquations$

Sagemath是一个复杂的数学系统，一方面它包容了70多个成熟的开源数学工具，比如。。。，当我们需要在Sage中显性的调用这些工具时，我们也需要对其有一个初步的了解，另一方面，作为一个纯粹的学术工具，Sage缺乏像matlab中各种专业工具箱，不利于Sage在专业领域的发展，当然中文资料的匮乏也不利于Sagemath在国内的发展。但，Sagemath的开发速度很快？越来越多的人参与到这个开源数学工具的开发和推广上来，现在西方已经有很多人？开始在专业领域应用Sagemath，如果想进一步了解Sage，可以到其官网下载使用，或使用其在线版本，Sagemath目前已经是一个非常成熟的数学系统，在学术界接受程度很高，在很多大

学？，国内？有应用，也多次作为演算工具，被各类论文引用，在目前反盗版，推荐国内各类高校部署和推广Sagemath。

References

[1] http://zh.wikipedia.org/zh-cn/%E8%AE%8A%E6%95%B8

[2] http://en.wikipedia.org/wiki/Sage_(mathematics_software)#cite_note-10

[3]

[4] file:///home/yub/%E6%96%87%E6%A1%A3/Sagemath/tutorial/tour_rings.htmlhttp://localhost:8080/doc/live/reference/rings/sage/rings/ring.html

[5] http://zh.wikipedia.org/zh-cn/%E5%8F%8C%E7%B2%BE%E5%BA%A6%E6%B5%AE%E7%82%B9%E6%95%B0

[6] http://woodpecker.org.cn/abyteofpython_cn/chinese/ch09s03.html

[7] file:///home/yub/%E6%96%87%E6%A1%A3/Sagemath/tutorial/tour_algebra.html#section-systems

[8] file:///home/yub/%E6%96%87%E6%A1%A3/Sagemath/tutorial/appendix.html#section-precedence

[9] 数学的三个发展时期——变量数学时期（2008-01-24 转自《大科普网》）http://www.pep.com.cn/gzsxb/xszx/czsxkwyd_1/czsxkwydsxgs/201009/t20100929_922925.htm

[10] file:///home/yub/%E6%96%87%E6%A1%A3/Sagemath/tutorial/tour_algebra.html

[11] calculus.pdf

[12] http://vibrationdata.com/python-wiki/index.php?title=Main_Page

[13] http://blog.163.com/soft_share@126/blog/static/4298360320131259214410 5/

[14]

[15] http://zh.wikipedia.org/wiki/%E5%82%85%E9%87%8C%E5%8F%B6%E5%8F%98%E6%8D%A2

[16] http://zh.wikipedia.org/zh-cn/%E6%8B%89%E6%99%AE%E6%8B%89%E6%96%AF%E5%8F%98%E6%8D%A2

[17] http://www.ai7.org/wp/html/904.html

[18] http://wiki.sagemath.org/Differential_Equations

[19] http://zh.wikipedia.org/zh-cn/%E5%BE%AE%E5%88%86%E6%96%B9%E7%A8%8B

[20] http://wiki.sagemath.org/Teaching_with_SAGE?highlight=%28fourier%29Computational

[21]

[22] http://zh.wikipedia.org/wiki/%E6%B3%B0%E5%8B%92%E7%BA%A7%E6%95%B0

[23] http://zh.wikipedia.org/wiki/%E6%B4%9B%E6%9C%97%E7%BA%A7%E6%95%B0