Python 与二元微积分实验教学结合探索

孙阳轩
（广州华商职业学院，广东 广州 511300）

摘要：Python是一门高效的编程语言，并且支持第三方库使用。由该软件在高等数学实验教学的一元微积分的应用，扩展到二元函数微积分。并将函数可视化与二元函数求极限、求极值、求偏导数、求重积分等融合，使复杂的函数曲面得到可视化，从而提高实验教学课堂的生动性，激发学生学习兴趣。

关键词：python；数学实验；高等数学；可视化

随着计算机技术的快速发展，人们利用计算机帮忙解决生活中的一些难题。高等数学教学中的一些抽象概念及理论知识不好理解，教师利用计算机软件将这些问题直观展示出来，不但使得学生对知识更容易理解，且使得课堂变得更生动。同时也要求学生动起手来，利用软件完成相关的实验，并对实验进行总结性汇报。这样教学过程逐渐发展形成了《数学实验》课程。

探索 python 与数学实验教学融合，主要 Python 目前是使用最广泛编程语言之一，且大部分扩展库是免费，且容易找到开源程序，多平台支持和简单易用的功能等方面都为数学实验提供便利。二元微积分是高等数学里面的一元微积分的推广，二元微积分相关的概念及理论需要空间想象力，因此需要融合 python 画图模块。

一、实验相关

实验用的例子来自同济大学出版《高等数学》教材（第七版）。使用 python 的 sympy、scipy、numpy、matplotlib 第三方库。sympy 是一个符号计算库。Scipy 数值计算库扩展，numpy 数值计算库，matplotlib 绘图库。计算机运行系统 Windows10、python3.8.

二、python 与实验

（一）函数可视化

函数可视化就是使用 python 软件将函数图形画出来。使高等数学中的抽象函数具体化、可视化。通过数学实验，学生利用软件对课本知识进行深入观察，并且进行知识检验。从而提高他们的动手能力和观察能力。函数形象化有助于课堂教学生动性，也引起学生的学习兴趣。

画平面图需要导入 matplotlib、numpy 两个模块，numpy 用来产生自变量和函数的数据，再用 matplotlib 模块中 plot（x，y）命令画出图形。对于三维画图，除了需要 matplotlib、numpy 两个模块，还需要安装 mpl_toolkits 工具包，通过从 mpl_toolkits.mplot3d 导入对象 Axes3D 来实现，再使用命令 ax.plot_surface（X，Y，Z）。一般步骤为：导入需要模块、产生自变量和函数数据、坐标轴画出图形。

例 1 画出二元函数极限 $\lim\limits_{(x,y)\to(0,0)}\frac{xy}{x^2+y^2}$ 中的二元函数 $z=\frac{xy}{x^2+y^2}$。

代码输入如下：

```
import matplotlib.pyplot as plt  import numpy as np  from mpl_toolkits.mplot3d import Axes3D  fig=plt.figure（）  ax=Axes3D（fig）
x=np.arange（-1，1，0.01）  y=np.arange（-1，1，0.01）  X，Y=np.meshgrid（x，y）  Z=（X*Y）/（X**2+Y**2）
ax.plot_surface（X，Y，Z，cmap='rainbow'）  ax.contour（X，Y，Z，zdir='z'，offset=-0.6，cmap='rainbow'）
```

输出结果如图 1 所示：

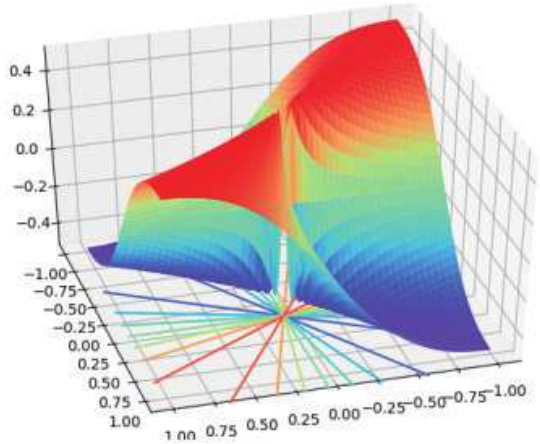


图 1

也可以用 sympy 模块和 sympy.plotting 模块进行画图，用 sympy.plotting 中的 plot3d（f，（x，a，b），（y，c，d））命令，其中 f 为含 x 和 y 为自变量的二元函数，（x，a，b）为 x 取值范围（a，b），（y，c，d）为 y 取值范围（c，d）。

将函数可视化，能直观地理解一元及二元函数形状及性质，同时对该函数在某点极限存在有直观的判别，从而加深对知识的理解和直观体会。这样引起学生兴趣，主动探索其他函数的图形及相关性质，对学习起了事半功倍作用。

（二）求极限

计算极限需要导入 sympy 模块，然后用 limit（）命令求解。对于求一元函数的极限，用命令 limit（f，x，x0，dir'+-'），其中 f 函数，x 为函数的自变量，第三个 dir'+'或 dir'-'或 dir'+-'表示右极限、左极限、极限等。表示负、正无穷极限的可以分别用 limit（f，x，float（'-inf'））、limit（f，x，float（'+inf'））。

对于二元函数求极限，作变换 $\lim_{(x,y)\to(x_0,y_0)} f(x,y)=\lim_{r\to 0} f(r\cos t+x_0,r\sin t+y_0)=A$ 为常数，则极限存在为 A；若 $\lim_{r\to 0} f(r\cos t+x_0,r\sin t+y_0)=g(t)$ 与 t 有关的函数，则极限不存在。因此在 sympy 模块中，用 subs（）命令进行变量替代，然后用 limit（）命令求解。

例 2 求二元函数极限 $\lim_{(x,y)\to(0,0)} \frac{xy}{x^2+y^2}$

代码输入如下：

```
import sympy as sp
x, y, r, t=sp.symbols（‘x, y, r, t’）; f=（x*y）/（x**2+y**2）;
f_rt=f.subs（[（x, r*（sp.cos（t））），（y, r*（sp.sin（t）））]）;
z=sp.limit（f_rt, r, 0）; print（sp.simplify（z））
```

输出结果如下：

```
sin（2*t）/2
```

由图 1 观察也知该点极限不存在。对于二元函数求极限问题，用上方法求出结果，再结合该函数可视化来判断。通过观察函数曲面在该点周围的变化，若该点附近“平坦”则该点极限存在，否则不存在。

（三）求导

求导需要导入模块 SymPy，使用命令 diff（）来实现 .diff（）命令的基本语法是：diff（f, x, n），其中 f 表示待求导的函数，x 为函数求导自变量，n 表示求导阶数。对于二元函数求偏导也是 sympy 模块中的 diff（）命令，如 diff（f（x, y）, x, n）命令是函数 f（x, y）对 x 求 n 次偏导数。diff（f（x, y）, x, y）表示先对 x 求偏导，再对 y 求偏导。一般步骤：导入模块、定义变量和函数、用 diff（）命令求导。

例 3 求 $z=x^2+3xy+y^2$ 在（1, 2）处的偏导数。

输入代码如下：

```
import sympy as sp
def f（x, y）: x, y=sp.symbols（“x, y”）; f_x=sp.diff（f（x, y）, x）; f_y=sp.diff（f（x, y）, y）
print（f_x.subs（[（x, 1）,（y, 2）]）, f_y.subs（[（x, 1）,（y, 2）]））
```

输出结果为：

```
8 7
```

将该函数在点（1, 2, 11）处偏导数的几何意义可视化，输入代码如下：

```
import matplotlib.pyplot as plt; import numpy as np
fig=plt.figure（）; ax=Axes3D（fig）; x=np.arange（-3, 3, 0.1）;
y=np.arange（-3, 3, 0.1）
X, Y=np.meshgrid（x, y）; Z=X**2+3*X*Y+Y**2; ax.plot_surface（X, Y, Z, alpha=0.3）
ax.plot（x, x**2+6*x+4, zs=2, zdir=’y’, color=’b’）
ax.plot（y, 1+3*y+y**2, zs=1, zdir=’x’, color=’g’）
u=np.arange（-2, 3, 0.1）; v=8*（u-1）+11; s=np.arange（0, 4, 0.1）; w=7*（s-2）+11
ax.plot（u, v, zs=2, zdir='y', color='red'）; ax.plot（s, w, zs=1, zdir='x', color='red'）
ax.set_xlabel（‘X’）; ax.set_ylabel（‘Y’）; ax.set_zlabel（‘Z’）
```

输出结果如图 2 所示：

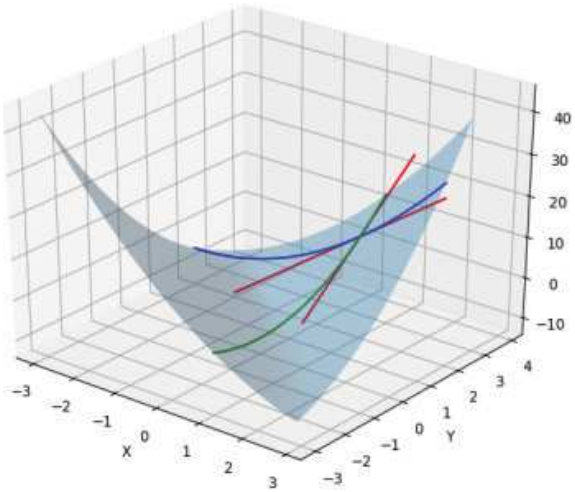


图 2

对于一元函数求导、隐函数求导、二元函数求偏导数、隐函数求导等，目前 diff（）命令都能很好解决了。将函数在某点处导数或偏导数的几何意义可视化，来加深对概念理解，同时也可以直观判断函数该点处的导数或偏导数是否存在。

（四）极值最值

函数的极值、最值、实际问题的最优化等研究。对于求函数极值可以用 sympy 模块命令 diff（）求导，再用 slove（）解出驻点及不可导的点，再用 if 语句判断极值的充分条件进行判断即可。这里主要介绍函数可视化以与其他模块综合使用。对于一元函数求极值用 scipy.optimize 的 minimize_scalar（f, bounds=（a, b））求最小值命令，其中 f 为函数，bounds=（a, b）为选定区间。二元函数求极值用 scipy.optimize 的 minimize（f（x, y）,（x0, y0）, bounds=（（a, b）,（c, d））, constraints=（））命令，其中 f 为函数，（x0, y0）表示初值点，bounds=（（a, b）,（c, d））为选定区间，constraints=（）为约束条件。该命令也为求最小值，因此求极大值时也要对函数适当变形。

步骤：用 sympy 模块求导，根据导数适当选择画图区间，用 matplotlib 画出函数图形，然后用 scipy 的优化模块寻找函数极值。以下的例子介绍该方法。

例 4　求函数 $f(x,y)=x^3-y^3+3x^2+3y^2-9x$ 的极值。

输入代码如下：

```
import sympy as sp; x, y=sp.symbols("x, y"); f=x**3-y**3+3*x**2+3*y**2-9*x
s=sp.solve([sp.diff(f, x), sp.diff(f, y)], [x, y]); print(s)
```

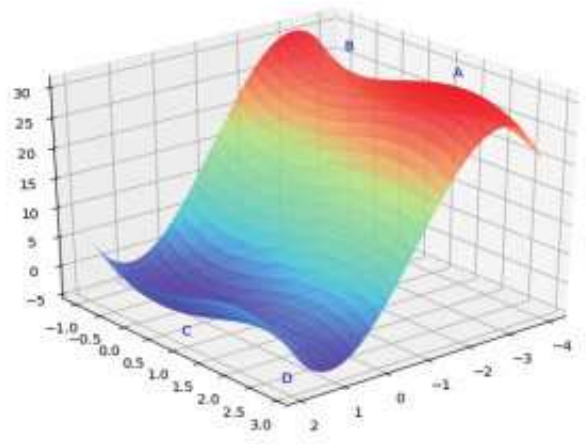
输出结果为：

[(-3, 0), (-3, 2), (1, 0), (1, 2)]

曲面可视化输入代码为：

```
import matplotlib.pyplot as plt; from mpl_toolkits.mplot3d import Axes3D
import numpy as np; fig=plt.figure(); ax=Axes3D(fig);
X, Y=np.meshgrid(x, y); Z=X**3-Y**3+3*X**2+3*Y**2-9*X;
ax.text(-3, 0, 31, "B", color='b'); ax.text(-3, 2, 33, "A", color='b')
ax.text(1, 0, -10, "C", color='b'); ax.text(1, 2, -10, "D", color='b')
```

输出结果如图 3 所示：



观察图形，极值在 A、C 点取得，B、D 点为马鞍面。输入代码为：

```
import scipy.optimize as so; f=lambda x: x[0]**3-x**3+3*x[0]**2+3*x**2-9*x[0]
x0=so.minimize(f, (1, 0.5), bounds=((0, 2), (-1, 1)));
print("极小值点", x0.x, "极小值：", x0.fun); g=lambda x: -(x[0]**3-x**3+3*x[0]**2+3*x**2-9*x[0])
x1=so.minimize(g, (-4, 1), bounds=((-4, -2), (1, 3)));
print("极大值点", x1.x, "极大值：", -x1.fun)
```

结果输出为：

极小值点 [1.00000006e+00 -1.32033418e-08] 极小值：-4.999999999999976

极大值点 [-2.9999987 2.00000025] 极大值：30.999999999989615

例 5　求表面积 36 为而体积为最大的长方形的体积。

分析：设长方体的三菱长为 x，y，与 z，则问题就是在条件 $2xy+2yz+2xz-36=0$ 下，求函数 $V=xyz\quad(x>0,y>0,z>0)$ 的最大值。

输入代码如下：

```
import scipy.optimize as so; h=lambda x: -x[0]*x*x
cons = ({'type': 'eq', 'fun': lambda x: x[0]*x + x*x + x[0]*x-18})
x2=so.minimize(h, (1, 1, 1), bounds=((0, None), (0, None), (0, None)), constraints=cons)
print(x2.success, x2.x, -x2.fun)
```

输出结果为：

True [2.44948978 2.44948962 2.44948983] 14.69693845677875

将该问题可视化，输入代码如下：

```
import matplotlib.pyplot as plt; from mpl_toolkits.mplot3d import Axes3D
import numpy as np; fig=plt.figure(); ax=Axes3D(fig)
x=np.arange(2, 8, 0.1); y=np.arange(2, 8, 0.1); X, Y=np.meshgrid(x, y)
Z=(18-Y*X)/(Y+X); Z1=1/(X*Y); Z2=30/(X*Y);
ax.plot_surface(X, Y, Z1, alpha=0.5)
ax.plot_surface(X, Y, Z2, alpha=0.5); ax.text(2, 2, 6, "Z2", color='b')
ax.text(2, 2, 2, "Z", color='b'); ax.text(2, 2, 0, "Z1", color='b')
ax.set_xlabel('X', fontweight ='bold'); ax.set_ylabel('Y', fontweight ='bold')
ax.set_zlabel('Z', fontweight ='bold')
```

结果输出如图 4 所示：

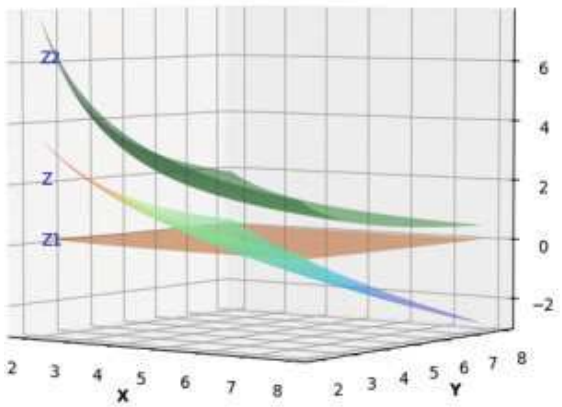


图 4

几何理解为：条件曲面函数为 $Z=(18-xy)/(x+y)$，目标曲

面函数 $z=V/xy$ 中的参数 V 从 $1\to18$ 时，目标曲面函数 $z=V/xy$ 从 Z1 → Z2 变化过程中，目标曲面函数与条件曲面函数产生“相交→相切→相离”变化。其中在相切时，就是 V 最大值。

通过函数可视化，对于二元函数求极值或无条件最值时，也可以从曲面图形直接观察出来，不过要在可疑点区域局部放大才好识别。对于二元函数有条件求最值，可以看作由条件函数曲面围成的区域中，目标函数曲面要与该区域有交点情况下取得最值。

（五）求积分

求积分可以用 sympy 模块的 Integrate（）命令计算，相关积分及对应命令如下

$\int f(x)dx$：Integrate（f（x），x）；$\int_a^b f(x)dx$ ：Integrate（f（x），（x，a，b））；

$\iint f(x,y)dxdy$ ：Integrate（f（x，y），x，y）；

$\int_c^d\int_a^b f(x,y)dxdy$：也可以用 scipy.Integrate 中的函数求定积分，相关积分及对应命令如下

$\int_a^b f(x)dx$：quad（f，a，b）；$\int_a^b dx\int_{u(x)}^{v(x)} f(x,y)dy$：Dblquad（f，a，b，lambda x：u，lambda x：v）；

$\int_a^b dx\int_{u(x)}^{v(x)}dy\int_{w(x,y)}^{s(x,y)} f(x,y,z)dz$：Tplquad（f，a，b，lambda x：u，lambda x：v，lambda x，y：w，lambda x，y：s）。也可以通用命令 Nquad（），表示通用 n 重数值积分。

例 6 用三重积分计算有曲面 $z=6-x^2-y^2$ 及 $z=\sqrt{x^2+y^2}$ 围成的立体的体积。

将围成的体积可视化，输入代码如下：

import matplotlib.pyplot as plt；from mpl_toolkits.mplot3d import Axes3D

import numpy as np；fig=plt.figure（）；ax=Axes3D（fig）

x=np.arange（−2，2，0.01）；y=np.arange（−2，2，0.01）；X，Y=np.meshgrid（x，y）

Z=6−X**2−Y**2；V=6−X**2−Y**2；U=np.sqrt（X**2+Y**2）；Z[Z<U]=np.nan；U[U>V]=np.nan

ax.plot_surface（X，Y，Z，rstride=1，cstride=1，alpha=0.5）

ax.plot_surface（X，Y，U，rstride=1，cstride=1，alpha=0.5）

def f（x，y）：return 6−x**2−y**2

x=np.arange（−2，2，0.1）；w=np.sqrt（4−x**2）

ax.plot（x，w，f（x，w），color=’red’）；ax.plot（x，−w，f（x，−w），color=’red’）

ax.set_xlabel（'X'，fontweight ='bold'）；ax.set_ylabel（'Y'，fontweight ='bold'）

ax.set_zlabel（‘Z’，fontweight =’bold’）

输出结果如图 5 所示：

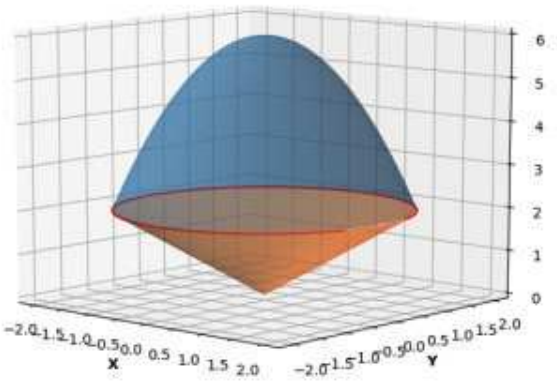


图 5

故 $\iiint_\Omega dxdydz=\int_{-2}^{2}dx\int_{-\sqrt{4-x^2}}^{\sqrt{4-x^2}}dy\int_{\sqrt{x^2+y^2}}^{6-x^2-y^2}dz$，输入代码为：

f=integrate.tplquad（lambda z，y，x：1，−2，2，

lambda x：−np.sqrt（4−x**2），lambda x：np.sqrt（4−x**2），

lambda x，y：np.sqrt（y**2+x**2），lambda x，y：6−x**2−y**2）

print（f）

输出结果为：（33.51032163792589，4.5458258498416725e−07）

对于重积分计算，特别是在积分区域确定时，积分区间是由曲线围成的区域或是由曲面围成区域。这些积分上下限不好确定，通过可视化之后，积分区域可以通过图形观察是由哪些曲线或曲面围成，从而引起学生注意，自主探索积分上下限确定规律。

三、结束语

采用 python 软件在二元微积分实验教学中，借助可视化有助于让学生多元函数性质有直观认识，特别对复杂的曲面得到直观性认识从而加深理解多元微分学。学生在进行知识验证和探索更好地掌握高等数学的理论知识，在探索中提升动手能力，为数学建模能力奠定基础，并提升高等数学教学质量。此外，逐步熟练 python 软件操作，为学生自己的职业生涯也奠定基础。

参考文献：

[1] 司守奎，孙玺菁 .Python 数学实验与建模 [M]. 北京：科学出版社，2020.

[2] 王晓刚 . 微积分运算的 Python 方法 [M]. 江苏：扬州职业大学学报，2021.

[3] 同济大学数学系 . 高等数学（第七版，上册）[M]. 北京：高等教育出版社，2014.

[4] 同济大学数学系 . 高等数学（第七版，下册）[M]. 北京：高等教育出版社，2014.