

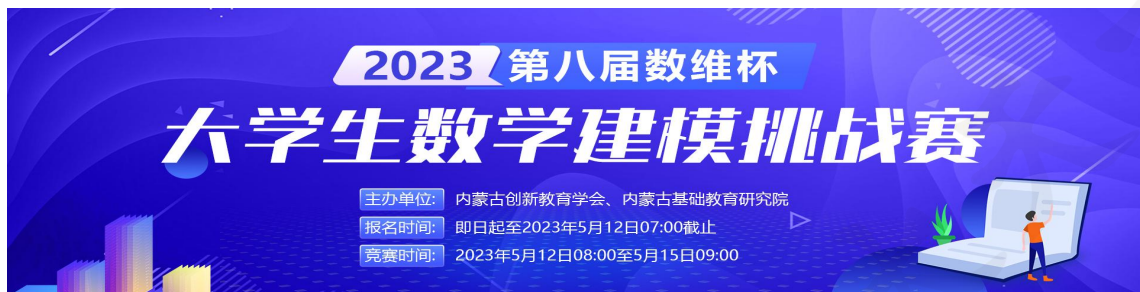
北太天元数模培训

北太天元在数模竞赛中的应用



北太天元

2023年第八届数维杯大学生数学建模挑战赛



数维杯大学生数学建模挑战赛由内蒙古创新教育学会、内蒙古基础教育研究院主办的全国性数学建模赛事，该竞赛每年分为两场，每年上半年为**数维杯国赛**（5月，俗称小国赛），下半年为**数维杯国际赛**（11月），累计参赛高校千余所，参赛人数超10万人，经过七年多的发展，竞赛已成为国内极具影响力的基础学科与应用科技的赛事，已被多所高校推广甚至列为国家级赛事选拔赛。也是国内最具发展潜力数模赛事！

本届竞赛得到了中国国产数值通用软件“北太天元”的赞助支持，同时为参赛者设立数万元奖金福利，2023年第八届数维杯数学建模挑战赛，等你来战！

一、主办单位：

主办单位：内蒙古创新教育学会、内蒙古基础教育研究院

技术支持：北太振寰（重庆）科技有限公司

二、竞赛时间：

报名截止时间：北京时间2023年5月12日07:00

竞赛开始时间：北京时间2023年5月12日08:00

竞赛结束时间：北京时间2023年5月15日09:00

竞赛结果公示时间：2023年7月中旬或之前

三、参赛对象：

参赛对象为在校专科生、本科生、研究生，每组参赛人数为1-3人（指导老师不列入小组总人数中，没有指导老师可写无，有指导老师可真实填写），每名同学只能参加一个小组，允许跨校组队。

四、赛题类型：

竞赛题目共3道（A题、B题、C题）。其中研究生、本科组请从A、B题中任选一个完成答卷，专科组请从B、C题中任选一个完成答卷。竞赛题目一般是来源于各行业并经过当简化的实际问题。

2023年第八届数维杯大学生数学建模挑战赛

五、奖项设置：

本次竞赛共评出：

1、特等奖：3队，采用视频答辩的形式，颁发第八届“数维杯”大学生数学建模挑战赛特等奖获奖证书+奖杯+每队1000元奖金+免费参加2023第八届数维杯大学生数学建模夏令营（成都）+学会会员。

2、创新奖：14队，采用视频答辩的形式，颁发第八届“数维杯”大学生数学建模挑战赛“创新奖”获奖证书+每队500元奖金。

3.使用北太天元数值计算通用软件参赛的队伍将有机会获得特别奖励：

第一名获得奖金6000元（需扣除个人所得税）

第二名获得奖金3000元（需扣除个人所得税）

第三名获得奖金1000元；

参赛队伍在使用该软件的过程中提供了某方面完整性的使用报告，将额外获得奖金500元；优秀参赛队伍可提供实习岗位与人才引进绿色通道。

4、全国一等奖：（约5%）+获奖证书+学会会员

5、全国二等奖：（约15%）+获奖证书+学会会员

6、全国三等奖：（约30%）+获奖证书+学会会员

7、优秀奖：（若干）（凡成功提交论文的队伍）+获奖电子版证书

8.优秀组织奖、优秀指导教师奖、优秀志愿者

须知：一等奖以上（含一等奖）将有机会被推荐到国内学术期刊发表，并邀请参加2023第八届数维杯大学生数学建模夏令营（成都）。

六、报名方式：

2023年第八届数维杯大学生数学建模挑战赛官网

http://www.nmmcm.org.cn/match_detail/26

七、联系方式：

2023数维杯官方QQ参赛2群：769924083

竞赛咨询QQ：1532597773、1786746895

竞赛志愿者咨询QQ：277558422



扫码立即加入数维杯官方群，
群内将不定时更新数模干货，
供同学们学习备赛

content

目录

01/ 北太天元介绍

02/ 北太天元工具箱和使用技巧

03/ 数模案例讲解



北太振寰（重庆）科技有限公司

北京大学重庆大数据研究院

北京大学重庆大数据研究院是在重庆市人民政府指导下，由重庆高新技术产业开发区管理委员会和北京大学共同举办的具有独立法人资格的重庆市属事业单位。研究院将在西部（重庆）科学城聚集培育科技创新人才，开展颠覆性和前沿引领创新研究，重点攻克大数据智能化和数字化转型的核心科学技术问题和创新应用瓶颈，促进科技成果转移转化，培育孵化科技企业，加快前沿技术的基础研究成果向试验发展创新和应用创新的转化，推进产业数字化与数字产业化，最终建成具有国际影响力、引领性的人才聚集平台、前沿研究和转化平台。研究院还将联合重庆高校院所，在科研攻关、人才培养等方面开展合作，加速建设一批国家和市级重点实验室、技术创新中心等创新平台。

北太振寰（重庆）科技有限公司

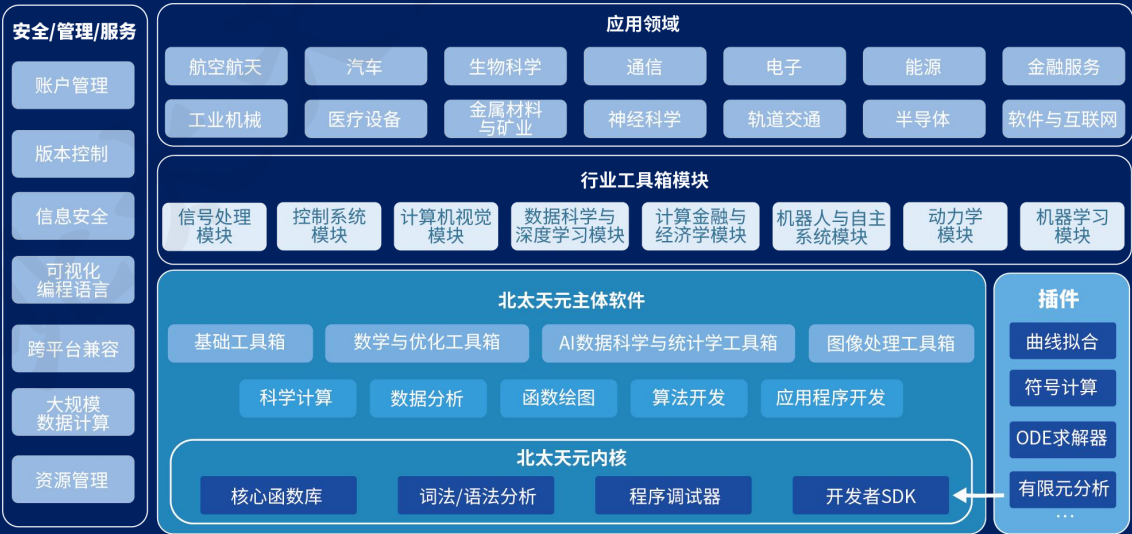
公司以重庆为总部，致力于打造国产通用型科学计算软件，融合数学、物理、计算机、工业工程等多种学科的理论知识与实践经验，围绕国产自主知识产权的科学计算软件底层根技术，发展国内数学软件、工业软件、专业软件等核心关键软件的发展，构建以应用数学为基础的技术生态、应用生态与产业生态。



北太天元数值计算通用软件

Baltamatica

北太天元数值计算通用软件（以下简称“北太天元”）是面向科学计算与工程计算的国产通用型科学计算软件。本软件具有自主知识产权，提供科学计算、可视化、交互式程序设计，具备丰富的底层数学函数库，支持数值计算、数据分析、数据可视化、数据优化、算法开发等工作，并通过SDK与API接口，扩展支持各类学科与行业场景，为各领域科学家与工程师提供优质、可靠的科学计算环境。



01 PART ONE

北太天元介绍



软件下载和安装

- 进入北太振寰官方网站-产品介绍-下载试用
www.baltamatica.com

北太振寰

首页

产品介绍

生态合作

服务支持

关于我们

开发者社区

滕振寰



北太天元下载试用

- 科学计算内核
- 数值计算
- 第三方插件
- 解释性语言
- ToolBOX

 Windows

 Ubuntu

 Deepin

 统信 统信UOS

 银河麒麟

帮助中心

软件下载和安装

- Windows直接双击安装包运行
(点击 更多信息 再点击 运行)

- Ubuntu下安装运行如下命令：
`sudo dpkg -i
baltamatica_X_ubuntu20.04_amd64.deb`



系统适配

已支持目前主流操作系统Windows、Linux与国产操作系统的Deepin、统信UOS、银河麒麟；即将推出MAC操作系统版本，可支持不同操作系统下文件编码格式的适配。



Windows

Windows7\8\10 64位



Linux

Ubuntu 64位



Mac

适配中
敬请期待……



Deepin

Deepin 64位



统信UOS

统信 64位



银河麒麟

已上线

LOONGSON 龙芯

Phytium 飞腾



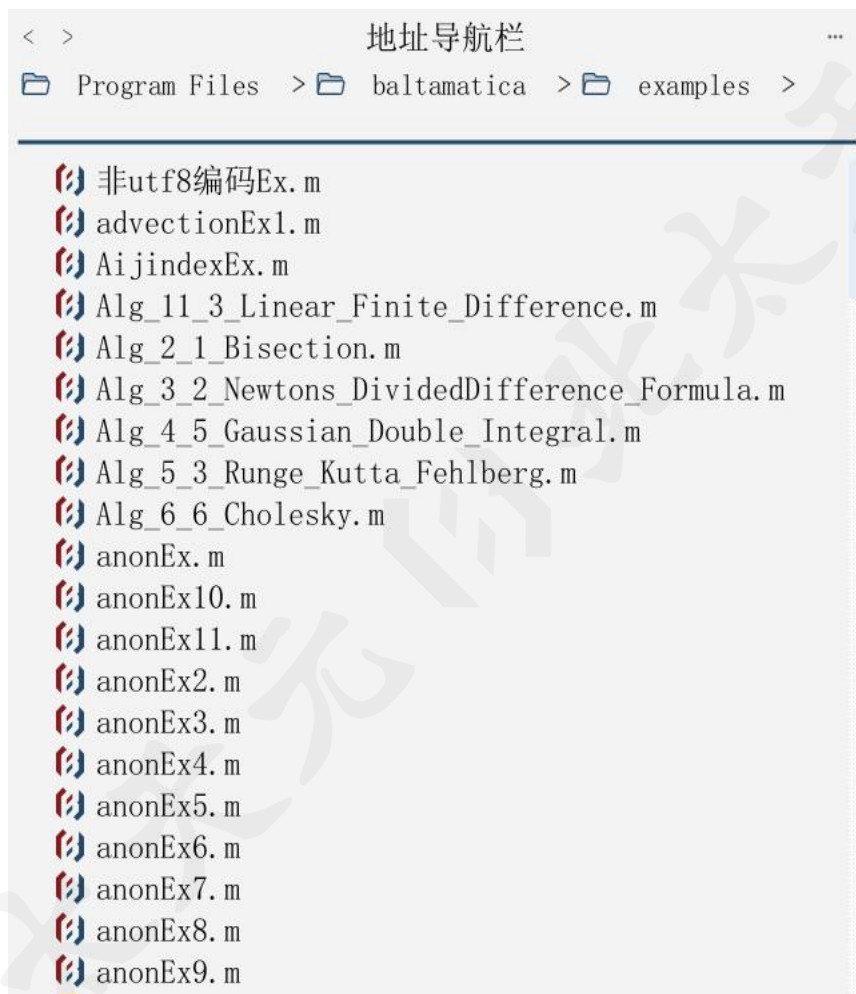
软件界面



北太天元软件安装完毕后，启动软件，即会看到软件的主页面，主要由菜单栏、工具栏、地址导航栏窗口、脚本编辑器窗口、工作区窗口和命令行窗口等组成。



软件目录介绍



脚本相关操作

新建脚本

编写脚本前，可以选择菜单栏的文件菜单、工具栏的“新建”按钮等方式新建一个文档。

打开脚本

编写脚本前，可以选择菜单栏的文件菜单、工具栏的“打开”按钮等方式打开一个文档。

保存与运行脚本

在运行脚本前首先需要保存文件，我们推荐保存位置为：地址导航栏显示相一致的当前工作目录（.m文件的命名不推荐用数字开头）。

此后通过点击菜单栏的运行菜单、工具栏的“运行”按钮、在右键菜单中选择运行或直接在命令行窗口输入脚本名称都可以立即运行刚完成编辑的脚本，命令行窗口随即显示对应的输出结果。

脚本路径管理

使用函数管理加载脚本路径

北太天元采用扫描机制，进入某目录后会自动扫描并添加当前活动目录下的可执行脚本。

函数	解释
pwd	显示当前目录
addpath/source	加载目录下的脚本
using/using_script	添加指定的脚本
discard	移除目录下的所有脚本

界面操作管理加载脚本路径

选择菜单栏的设置菜单的“路径管理”方式对脚本路径进行管理。



数据导入导出

数据导入导出的文件格式：

目前支持 csv、mat、xlsx、xls、txt

数据导入导出的数据类型：

目前支持 char, string, single, double, int32, int64

- 可以在界面操作读取和保存数据
- 使用load/save函数来读写mat文件

使用readmatrix/writematrix来读写csv、xlsx、xls、txt文件

数据导入导出

在界面进行操作



工作区		
名称	值	属性
新建	Ctrl+N	
粘贴	Ctrl+V	
清空工作区		
全选	Ctrl+A	
导入文件		
保存		

代码调试

如果北太天元软件程序出现运行错误或者输入结果与预期结果不一致，那么就需要对所编的程序进行调试。

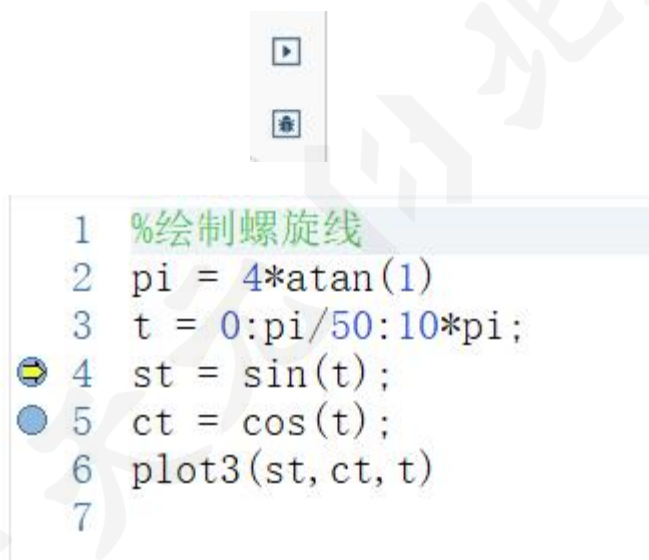
最常用的调试方式有两种：

1. 根据程序运行时系统给出的错误信息或警告信息进行相应的修改；
2. 通过用户设置断点来对程序进行调试。

代码调试

- 进入调试模式

点击菜单栏“运行”→“调试”按钮，或者工具栏的“调试”按钮，即可调试当前文本编辑器打开的脚本文件，程序会自动运行到第一个断点所在行。



- I. 行号左侧显示代码运行到的位置
- II. 工作区会显示内存中各变量的数值
- III. 可以通过“下一行”按钮来逐行运行代码，也可以通过“运行到下一断点按钮”来直接运行到下一个断点的位置。
- IV. 点击“退出调试”或者让脚本运行到代码末尾时，程序会自动退出调试模式，返回到编辑模式。

代码调试

- 数据查看

在调试模式中，当你的鼠标悬停到某一变量上方时，会显示此时变量的内容

而如果运行的脚本调用了别的函数，那么你可以通过工作区上方的函数栈下拉列表选择各个函数的内存栈，分别查看不同栈内储存的变量信息。

工作区		
函数栈		
luoxuanxian		
名称	值	属性
t	1*501 double ...	1*501 double matrix
pi	3.14159	1*1 double matrix
st	1*501 double ...	1*501 double matrix
ans	0.785398	1*1 double matrix
ct	1*501 double ...	1*501 double matrix



help指令

可以查看内核、脚本、插件提供的命令。

调用格式为：

>> help

命令行窗口

>> help
请使用如下关键词获取帮助：

[*] 内核提供的命令：

Chi	Ci	Ei	Ei_3	Inf	NaN	Realmax	Realmin	Shi
Si	abs	acos	acosh	acot	acoth	acsc	acsch	
addpath	airy	airyzero	all	and	angle	any	asec	asech
asin	asinh	atan	atan2	atanh	atanint	besselh	besseli	
besselj	besseljzero		besselk	bessely	beta	betainc	betaincinv	
betaln	binop_add	binop_mul	binop_sub	blanks	blkdiag	block	brace_get	
brace_set	cast	cd	ceil	cell	cell2struct		celldisp	char
chol	choose	class	clausen	clc	clear	clear_global		coeff
col	colon	complex	cond	conicalP	conicalP_cyl_reg		conicalP_sph_reg	
conj	constrImpliedBoundaries			convertStringsToChars			copyfile	
corrcoef	cos	cosh	cot	coth	cov	cross	csc	csch
csvread	ctranspose		cummax	cummin	cumprod	cumsum	dawson	dbast
dbbreaks	dbclear	dbcond	dbcont	dbcontinue		dbdown	dbexit	
dbframe	dblist	dbload	dbnext	dboff	dbon	dbquit	dbstack	dbstep
dbstop	dbtype	dbup	deblank	debug	debug_off	debug_on	debye	
deg2rad	det	diag	diff	dilog	dir	discard	disp	
disusing	disusing_script	dot		double	e	eig	ellint_D	
ellint_E	ellint_F	ellint_P	ellint_RC	ellint_RD	ellint_RF	ellint_RJ	ellipP	ellipj
ellipke	eps	eq	erf	erf_Q	erf_Z	erfc	erfcinv	erfinv
error	eta	eval	exist	exp	expint	expml	eye	fact
false	fclose	fermi_dirac_half		fermi_dirac_inc0		fermi_dirac_int		ferror
feval	fieldnames		fileparts	find	fix	floor	fopen	format
fprintf	frewind	fscanf	fseek	ftell	full	gamma	gammainc	



help + 函数（指令）名

假如准确知道所要求助的主题词，或指令名称，那么使用help是获得在线帮助最简单有效的途径。在平时的使用中，这个命令是最有用的，能最快、最好地解决用户在使用过程中碰到的问题。

调用格式为：

>> help 函数（类）名

```
>> help sin
```

参数的正弦，以弧度为单位。

Y = sin(X) 返回 X 的元素的正弦。sin 函数按元素处理数组。该函数同时接受实数和复数输入。

对于 X 的实数值，sin(X) 返回区间 $[-1, 1]$ 内的实数值。

对于 X 的复数值，sin(X) 返回复数值。

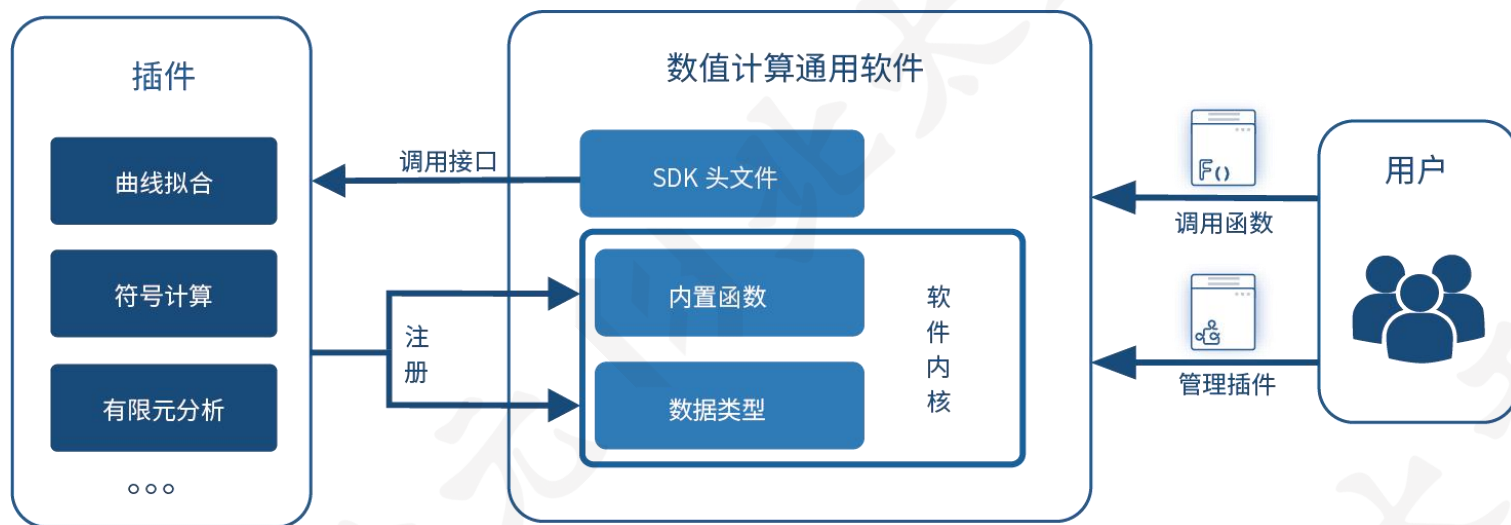
示例：

```
A=[1,2,3;2,3,4];
```

```
B = sin(A)
```

```
>> a|
```


基于插件的开发者模式



活跃中的插件开发:

- 绘图插件 (OpenGL)
- 符号运算插件
- 曲线拟合插件
- 有限元插件 (开发中)
- 快速傅里叶变换 (FFTW)

- 直观、稳健的编写规范
- 易于上手的 API 与编程文档

- 丰富的功能: 增加内置函数、托管对象、运算符重载、显示 UI……
- 方便快速进行已有 C/C++/FORTRAN 代码的快速转化

官方网站

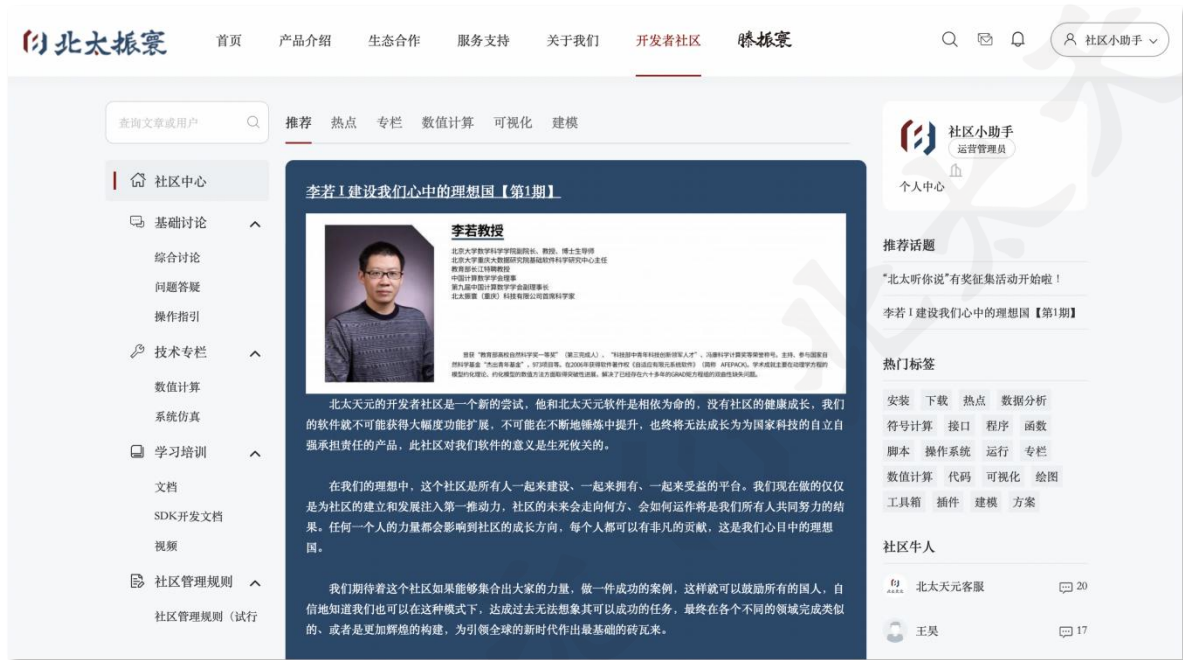
www.baltamatica.com

北太振寰官网已上线

- 软件试用
- 白皮书下载



北太天元开发者社区



北太天元开发者社区（以下简称“社区”）是北太天元唯一官方开发者社区，覆盖航天工业、汽车工业、生物医学工程、财务金融分析、建模仿真等多个技术领域。

社区是帮助开发者认知、交流、深入、实践的一站式社区，提供工具资源、优质内容、学习实践、大赛活动、专家社群，让开发者更高效科学的进行工作。



使用过程中遇到问题可发帖询问，会有牛人解答



在技术专栏系统地学习一些理论知识



找到学习文档及视频，进行实操练习



扫码注册社区，学习快人一步

北太天元软件支持的课程教学案例: 《数值方法: 原理、算法及应用》

- 插值、函数拟合
- 数值积分(包括MC方法)
- 数值微分
- 模拟退火
- 遗传算法
- 线性规划、整数规划
- RSA 加密
- FFT
- Dijkstra算法
- Floyd算法
- k均值聚类
- k近邻算法
- 支持向量机算法
- 朴素贝叶斯算法
- 决策树
- 关联规则挖掘(apriori)
- 网页排名算法 (page-rank)
- 期望最大化算法(EM)
- 主成分分析

02 PART TWO

北太天元工具箱和使用技巧



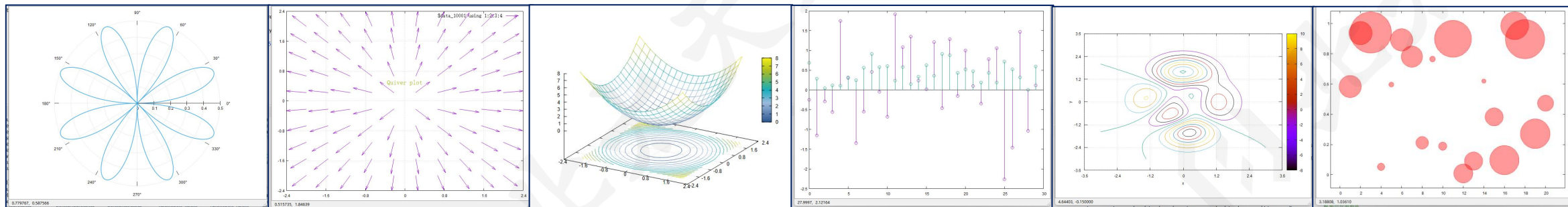
工具箱

■ 工具箱加载



工具箱

■ 工具箱介绍



极坐标图

向量场图

曲面图

针状图

等值线图

气泡图

01

FFT插件

完成fft插件的代码重构工作，增加其易扩展性以及维护性，共计重构开发7个插件函数。可以支持常规fft功能的使用。

02

优化工具箱

完成团队内部c++接口形式的插件开发框架，完成6个优化插件函数，5个优化脚本函数的开发，可支撑线性规划，整数规划，二次规划的基本功能。

03

绘图工具箱

支持二维三维绘图功能，支持图像显示、编辑等功能。

04

数值积分

支持常见的一重积分、二重积分、三重积分、梯形数值积分等函数。

05

ODE求解器

完成ODE（常微分）求解器总共 55 个脚本函数。

06

曲线拟合工具箱

提供了csapi、csape、ppmak、fitcurve、bspline、fnval等函数。

快速傅里叶变换（FFT）工具箱

```
>> plugin_help("fft")
```

插件 [fft] 提供的命令：

fft fft2 nufft fftshift ifft ifft2 ifftshift nextpow2

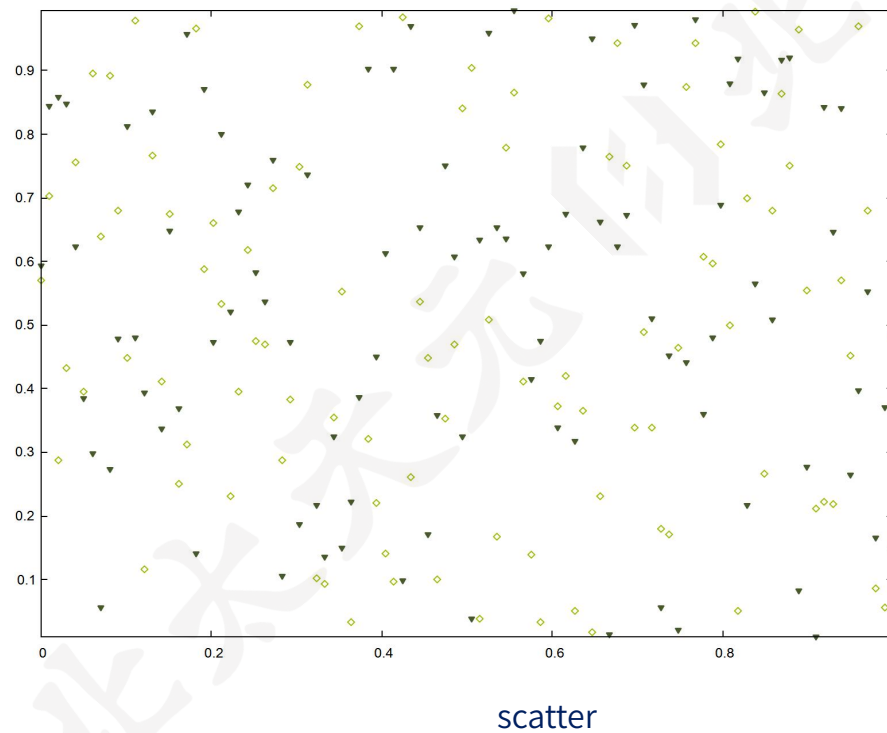
优化工具箱

- fzero, fminbnd, fminsearch, fsolve
- lsqnonneg, lsqlin, lsqnonlin
- linprog, intlinprog, quadprog

绘图

■ 二维绘图

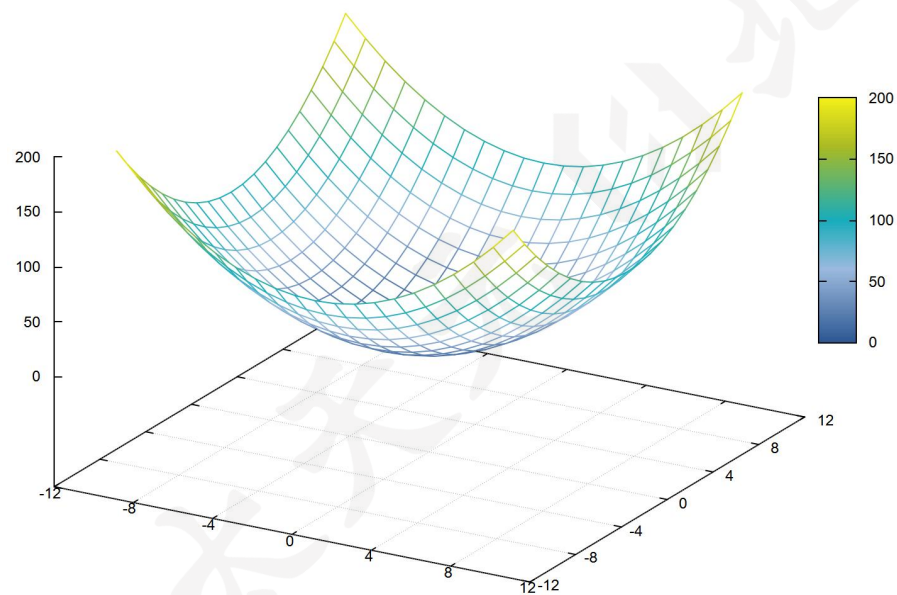
plot, scatter, pie,
area, fill, bar,
contour, heatmap, quiver,
boxplot, stem, polarplot,
hist, bubblechart, polarbubblechart,
polarhistogram



绘图

■ 三维绘图

plot3, scatter3
surf, surfc, surface,
mesh, meshz, meshc



mesh

绘图

■ 功能函数

- clf
- hold on, hold off
- figure 创建图窗窗口
- subplot 画子图
- text 向数据点添加文本说明
- xlabel, ylabel, zlabel 给坐标轴添加标签
- title 添加标题
- axis 坐标轴设置
- xline, yline 添加垂直于坐标轴的辅助线
- xticks, yticks, zticks 设置坐标轴的刻度值
- xticklabels, yticklabels, zticklabels 设置坐标轴刻度标签
- subtitle 添加标题

数值积分

- Integral, integral2, integral3, quad, quadl, quadv, quadgk
- quadgk, dblquad, triplequad
- trapz, cumtrapz

ODE

非刚性微分方程求解器

- ode23
- ode45
- ode78
- ode113

刚性微分方程求解器

- ode15s
- ode23s
- ode23t
- ode23tb

技术优势与创新性



具有自主知识产权的解释型
高级编程语言，且与国际同类软件
兼容

**解释型
高级编程语言**



具有代码编辑、运行、调试
的交互式集成开发环境

**交互式
集成开发环境**



采用可扩展软件架构设计，
提供底层编程接口与开发者工具，
支持开发者用户编写自定义插件、
工具箱

**SDK开发工具
可扩展架构设计**

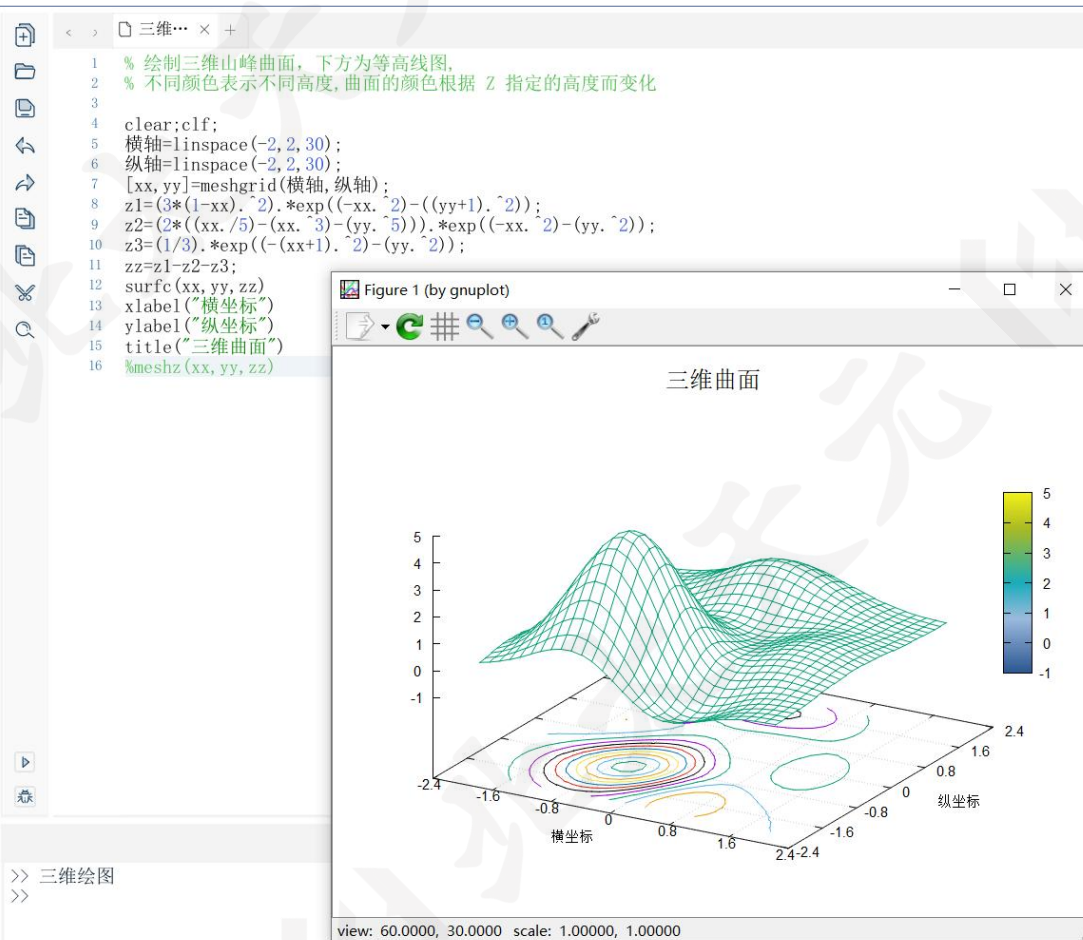


支持多种常见的数据存储格式，
且与国际同类软件兼容

**数据储存格式
与兼容性**

中文支持

- 中文函数名
- 中文变量
- 中文字符



多行注释

使用

```
/*  
...  
...  
*/
```

进行多行注释

```
< > 三维... × 寻找... × 线性... × +  
1 % 线性回归分析  
2 /*  
3 抽取了11个城市居民家庭的关于收入和食品支出的样本,  
4 通过线性回归得到一个一元二次函数  $f(x) = ax^2 + bx + c$   
5 可以用来预测某个收入对应的食品支出  
6 通过将点X0带入到  $f(x)$  中得到  $f(X0)$   
7 */  
8 clear;  
9 收入 = [82 93 105 130 144 150 160 180 270 350 600]; % 可以更改  
10 支出 = [75 85 92 105 120 130 145 156 200 240 240]; % 可以更改  
11 p = polyfit(收入,支出,2);  
12  
13 plot(收入,支出,'o')  
14 x0 = [min(收入):max(收入)];  
15 模拟支出 = p(1).*x0.^2 + p(2).*x0 + p(3);  
16 hold on  
17 plot(x0,模拟支出)  
18 hold off  
19 xlabel("收入")  
20 ylabel("食品支出")  
21 title("收入与食品支出的回归分析")  
22 disp(["函数方程为:  $f(x) =$ ",p(1),"  $*x^2 +$ ",p(2),"  $*x +$ ",p(3)])  
23
```

软件轻量化

软件安装包小，安装简单，并且启动迅速

Size: 109 MB (115,046,869 bytes)

Size on disk: 109 MB (115,081,216 bytes)

函数句柄

使用 m 脚本来定义函数

```
< > sump... × +  
1 function [z] = sumprod(m,n)  
2     z=m.^2 + n.^2;  
3 end  
4
```

使用 函数句柄

```
>> z = @(x,y) x.^2 + y.^2  
  
z =  
  
1x1 function_handle  
    anonymous function (182343) defined by @(x,y) x.^2 + y.^2  
>> z(1,2)  
  
ans =  
  
1x1 double  
  
5
```

函数句柄

- 匿名函数作为入参

使用示例:

```
X = fzero(@sin, 3)
X = fzero(@(x) sin(3*x), 2)
X = fzero(@cos, [1 2])
X = fzero(@sin, 3, optimset('Display', 'iter'))
```

```
myfun = @(x, c) cos(c*x);
c = 2;
X = fzero(@(x) myfun(x, c), 0.1)
```

- 通过函数句柄将m脚本的函数作为入参

使用向量操作

```
1  
2 t = tic;  
3 A = zeros(1,100);  
4 for m = 1:1000  
5     A(m) = m^2;  
6 end  
7 toc(t);  
8
```

```
>> speed_test2  
时间经过了 0.008668 秒。  
>> speed_test2  
时间经过了 0.009073 秒。  
>> speed_test2  
时间经过了 0.009170 秒。  
>>
```

```
1  
2 t = tic;  
3 A = zeros(1,100);  
4 A = (1:100).^2;  
5 toc(t);  
6
```

```
>> speed_test2  
时间经过了 0.000218 秒。  
>> speed_test2  
时间经过了 0.000203 秒。  
>> speed_test2  
时间经过了 0.000200 秒。  
>>
```

记录代码运行时间

```
>> help tic
```

tic 启动计时器，记录当前时间。

tic 会启动计时器并将当前时间记录到内置的全局记录中。

tVal = tic 会启动计时器并将当前时间以整数的方式记录到变量 **tVal** 中。
全局记录不受影响。**tVal** 变量一般会被用作 **toc** 函数的输入。

如果多次执行无返回值的 **tic**，则全局记录只会保存最后一次 **tic** 的时间。

例子：

```
t = tic;  
A = randn(10000, 5000);  
toc(t);
```

03 PART THREE

数模案例讲解



优化工具箱

线性规划

`optimization::linprog` 求解线性规划问题。

`x = optimization::linprog(f, A, rl, ru, cl, cu, options)` 用于求解线性规划问题

$$\begin{array}{ll} \min & f' * x, \\ \text{s.t.} & rl \leq Ax \leq ru, \\ & cl \leq x \leq cu \end{array}$$

`[x, dr] = optimization::linprog(...)` 还会输出 $rl \leq Ax \leq ru$ 的对偶变量 `dr`。

`[x, dr, dc] = optimization::linprog(...)` 还会输出 $rl \leq Ax \leq ru$ 的对偶变量 `dr`，以及 $d \leq x \leq cu$ 的对偶变量 `dc`。

`[x, dr, dc, output] = optimization::linprog(...)` 还会额外输出其它有用的求解信息，这些信息存放在结构体 `output` 中。

该函数要求参数 `f`, `rl`, `ru`, `cl`, `cu` 为 `double` 类型数据，`A` 为 `sparse double` 类型数据。

线性规划

解决规划类问题

```
< > test... × +
1 %求解线性规划问题
2 % 这是一个运输问题(transportation problem)
3 %   三个仓库，两个商店
4 % 目标函数 f = x1+x2+x3+x4+x5+x6
5 % 约束
6 % x1+x2 = 30 ; %第一个仓库的货物是30吨
7 % x3+x4 = 70 ; %第二个仓库的货物是70吨
8 % x5+x6 = 30; %第三个仓库的货物是30吨
9 % x1+x3+x5 = 50; % 第一个商店需要的货物是50吨
10 % x2+x4+x6 = 80; % 第二哥商店需要的货物是80吨
11 % x1 x2 x3 x4 x5 x6 >= 0
12
13 load_plugin("optimization");
14
15 opts = struct();
16 opts.Display = 'off';
17 f = ones(1,6);
18 A = sparse([1 1 0 0 0 0 ; 0 0 1 1 0 0 ; 0 0 0 0 1 1; 1 0 1 0 1 0; ...
19           0 1 0 1 0 1]);
20 rl = [30,70,30, 50 , 80];
21 ru = rl;
22 cl = zeros(6,1);
23 cu = [];
24 opts.Algorithm = 'interior-point';
25
26 [x, dr, dc, output] = linprog_hgs(f, A, rl, ru, cl, cu, opts)
27
28
```

线性混合整数规划

`optimization::intlinprog` 求解线性混合整数规划问题。

`x = optimization::intlinprog(f, intcon, A, rl, ru, d, cu, options)` 用于求解线性混合整数规划问题

$$\begin{aligned} \min \quad & f' * x, \\ \text{s.t.} \quad & rl \leq Ax \leq ru, \\ & cl \leq x \leq cu \\ & x(\text{intcon}) \text{ 为整数} \end{aligned}$$

其中 `intcon` 是一个指标向量，例如 `intcon = [1,2,5]` 表明希望 `x(1)`、`x(2)`、`x(5)` 是整数。

`[x, output] = optimization::linprog(...)` 还会输出结构体 `output`，包含其它求解信息。

- `output.relativegap` : 上界 (U) 和下界 (L) 的相对间隙。以百分比表示。
- `output.absolutegap` : 上界 (U) 和下界 (L) 的绝对间隙。等于 `U - L`
- `output.numnodes` : 分支定界算法中使用的分支节点数。
- `output.message` : 求解器退出时的信息。
- `output.ecode` : 求解器退出时的返回值。
- `output.fval` : 最优函数值。

当 `intcon = []` (即不存在整数约束) 时, `relativegap`、`absolutegap`、`numnodes` 为 `[]`。

该函数要求参数 `f`, `intcon`, `rl`, `ru`, `cl`, `cu` 为 `double` 类型数据, `A` 为 `sparse double` 类型数据。

整数规划

化学反应式配平

```
1 % 化学反应
2 %  $4\text{K}_4\text{Fe}(\text{CN})_6 + 30\text{KMnO}_4 + 82\text{H}_2\text{SO}_4 \rightarrow 46\text{KHSO}_4 + 2\text{Fe}_2(\text{SO}_4)_3 + 30\text{MnSO}_4 + 24\text{HNO}_3 + 24\text{CO}_2 + 47\text{H}_2$ 
3 % 假设我们不知道上面的化学反应式的系数，该如何求呢？
4 % 这是一个化学反应式配平的问题。
5 % 如何配平上面的化学反应式，需要求的正整数解，而且整数解是不唯一的，
6 % 要求最小的那个正整数解
7 % 北太天元数值计算通用软件上的脚本
8
9 load_plugin("optimization");
10 opts = struct();
11 opts.Display = 'off';
12
13 %  $\text{K}_4\text{Fe}(\text{CN})_6 + \text{KMnO}_4 + \text{H}_2\text{SO}_4 \rightarrow \text{KHSO}_4 + \text{Fe}_2(\text{SO}_4)_3 + \text{MnSO}_4 + \text{HNO}_3 + \text{CO}_2 + \text{H}_2$ 
14
15 % 系数总共有 9，我们的目标函数设定为这个9个系数之和，
16 f = ones(1,9);
17
18 % 化学元素包括
19 %   K Fe C N M H S O
20 X1 = [4 1 6 6 0 0 0 0];
21 X2 = [1 0 0 0 1 0 0 4];
22 X3 = [0 0 0 0 0 2 1 4];
23 Y1 = [1 0 0 0 0 1 1 4];
24 Y2 = [0 2 0 0 0 0 3 12];
25 Y3 = [0 0 0 0 1 0 1 4];
26 Y4 = [0 0 0 1 0 1 0 3];
27 Y5 = [0 0 1 0 0 0 0 2];
28 Y6 = [0 0 0 0 0 2 0 0];
29
30 A = [X1', X2', X3'] % 左边
31 b = [-Y1', -Y2', -Y3', -Y4', -Y5', -Y6'] % 右边
32
33 AA = sparse([A, b]);
34 rl = 0;
35 ru = 0;
36
37 cl = 1;
38 cu = [];
39
40
41 intcon = [1:9]
42 % intcon表示这9个决策变量均取整数
43 % 整数规划
44 [x, fval] = intlinprog_hgs(f, intcon, AA, rl, ru, cl, cu, opts)
45
```


凸二次规划

`optimization::quadprog` 求凸二次规划问题。

`x = optimization::quadprog(H, f, A, rl, ru, d, cu, options)` 用于凸二次规划问题

$$\begin{aligned} \min \quad & 0.5 * x' * H * x + f' * x, \\ \text{s.t.} \quad & rl \leq Ax \leq ru, \\ & cl \leq x \leq cu \end{aligned}$$

`[x, dr] = optimization::quadprog(...)` 还会输出 $rl \leq Ax \leq ru$ 的对偶变量 `dr`。

`[x, dr, dc] = optimization::quadprog(...)` 还会输出 $rl \leq Ax \leq ru$ 的对偶变量 `dr`，以及 $d \leq x \leq cu$ 的对偶变量 `dc`。

`[x, dr, dc, output] = optimization::quadprog(...)` 还会额外输出其它有用的求解信息，这些信息存放在结构体 `output` 中。

- `output.message` :求解器退出时的信息。
- `output.ecode` :求解器退出时的返回值。
- `output.fval` :最优函数值。

该函数要求参数 `f`, `rl`, `ru`, `cl`, `cu` 为 `double` 类型数据，`H`, `A` 为 `sparse double` 类型数据。变量 `H` 为下三角矩阵，为实际二次项的下三角部分。

线性回归分析

使用 polyfit 进行多项式拟合

```
< > 三维... × 寻找... × 线性... × +
1 % 线性回归分析
2 /*
3 抽取了11个城市居民家庭的关于收入和食品支出的样本,
4 通过线性回归得到一个一元二次函数  $f(x) = a*x^2 + b*x + c$ 
5 可以用来预测某个收入对应的食品支出
6 通过将点X0带入到  $f(x)$  中得到  $f(X0)$ 
7 */
8 clear;
9 收入 = [82 93 105 130 144 150 160 180 270 350 600]; % 可以更改
10 支出 = [75 85 92 105 120 130 145 156 200 240 240]; % 可以更改
11 p = polyfit(收入, 支出, 2);
12
13 plot(收入, 支出, 'o')
14 x0 = [min(收入):max(收入)];
15 模拟支出 = p(1).*x0.^2 + p(2).*x0 + p(3);
16 hold on
17 plot(x0, 模拟支出)
18 hold off
19 xlabel("收入")
20 ylabel("食品支出")
21 title("收入与食品支出的回归分析")
22 disp(["函数方程为:  $f(x) =$ ", p(1), "  $*x^2 +$ ", p(2), "  $*x +$ ", p(3)])
23
```

• K-means聚类算法

```
function [C, I, iter] = myKmeans(X, K, maxIter, TOL)
```

```
% 获取X中的向量格式
```

```
[vectors_num, dim] = size(X);
```

```
% 计算输入向量的随机排列
```

```
R = randperm(vectors_num);
```

```
% 构造 指标矩阵, 每个元素都指向X中聚类中的每个点
```

```
I = zeros(vectors_num, 1);
```

```
% 构造中心矩阵
```

```
C = zeros(K, dim);
```

```
% 用其中K个点来作为中心点
```

```
for k=1:K
```

```
    C(k,:) = X(R(k,:),:);
```

```
end
```

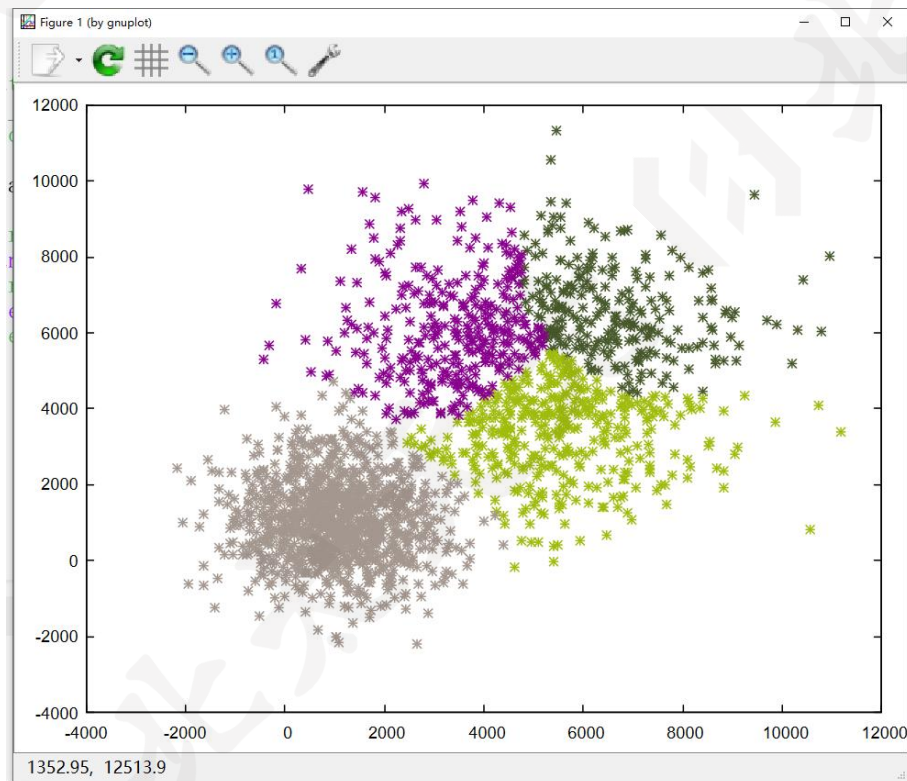
```
% 记录迭代次数
```

```
iter = 0;
```

```
% 计算新的聚类
```

```
% 满足聚类内累积误差保持在允许的最大误差以下
```

```
% 或者迭代过程未超过允许的最大迭代次数
```



求解ODE问题

```
1 % 求解二阶ODE方程初值问题
2 % 其中未知函数z(t)的满足的ODE方程是
3 %  $z'' = (1-z^2)*z' - z$ 
4 % 初值条件是
5 %  $z(0) = 2$ 
6 %  $z'(0) = 0$ 
7 %
8 % 我们首先把二阶ODE写成一阶ODEs
9 % 引入函数  $y = [z; z']$ 
10 % 得到 y 的一阶ODEs
11 %  $y' = [y(2); (1-y(1)^2)*y(2)-y(1)]$ 
12 % 然后用ode45求解 上面的一阶级ODEs,
13 % y(1) 就是所求的二阶ODE初值问题的解
14 clear, clc, close all;
15 odefun = @(t,y) [ y(2); (1-y(1)^2)*y(2)-y(1)]; % 必须返回列向量
16 tspan = [0 20];
17 y0 = [2; 0]; % 这里看起来需要用列向量, 但是实际上用行向量也能正确计算
18 options = odeset('RelTol', 1e-6, 'AbsTol', 1e-8);
19 [t,y] = ode45(odefun,tspan,y0,options); % 计算出来的y有两列, 分别对应函数和导数
20 plot(t,y(:,1),'-o',t,y(:,2),'-*');
21 legend('二阶ODE的解z', 'z'', 'Location', 'SouthEast')
22 xlabel('t')
23 title("用ode45求解二阶常微分方程的初值问题:  $z'' = (1-z^2)*z' - z$  ;  $z(0)=2$ ;  $z'(0)=0$  ")
24
```

2022 年第七届“数维杯”大学生 数学建模挑战赛论文

题 目 银行效率评价与倒闭风险预测的研究

摘 要

金融是国家经济的核心,而银行作为金融重要的构成部分, 银行对整个国家乃至世界经济的平稳与健康发展都有着重要的影响,本文主要针对银行建立合理的效率评价体系与倒闭风险评估预测模型,并进一步开展破产成因分析。

对于任务一:通过 RPCA 和 Wilk' λ 检验进行投入产出数据的预处理,接着对处理后的指标数据通过 TOPSIS 模型进行银行效率评价,最后基于 TOPSIS 评价结果进行 SVM 软间隔二分类来进行银行倒闭效率分界线的界定。

对于任务二:基于任务 1 的效率评价模型通过控制变量的方式来挖掘银行倒闭的原因,接着通过热力图和 K-均值聚类分析来定性定量对银行的 64 个指标进行降维至 5 个指标,最后通过熵权法来确定 5 个指标的权重。