



北京大学 重庆大数据研究院

PEKING UNIVERSITY CHONGQING RESEARCH INSTITUTE OF BIG DATA

北太天元插件开发

基础软件科学研究中心

Content

目录

01 北太天元数值计算通用软件

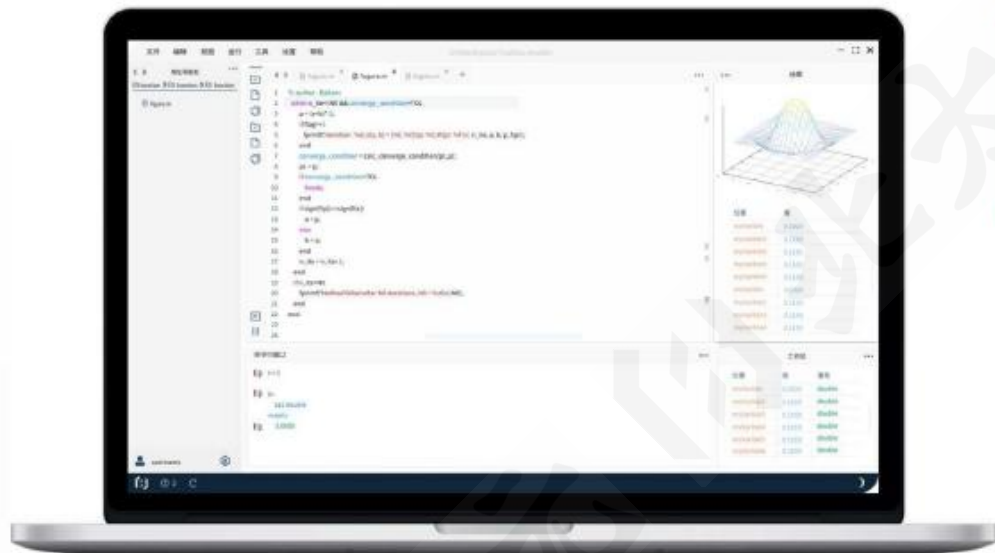
02 北太天元插件机制

03 北太天元插件开发

01/

北太天元插件开发

北太天元数值计算通用软件



Baltamatica Mathematical Software
北太天元数值计算通用软件

北太天元数值计算通用软件

北太天元是面向科学计算与工程计算的国产自主知识产权的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。



北太天元语言及解释器



全新的底层框架



最优的函数算法库



高效的参数解析能力



极致体验的用户脚本
迁移



支持多种主流数据格式



主流操作系统与硬件
环境全兼容



友好的代码编辑体验



完善的开发接口



人性化的插件机制

数值计算通用软件具备优良的跨平台特性，支持在以下平台上使用：

(1) Windows平台

win7、win10、win11等

(2) Linux平台

Ubuntu20.04、Deepin操作系统。

麒麟（开发中）、MacOS（开发中）

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。



底层 数据类型

数值矩阵

稠密矩阵：支持整数、实数浮点、复数浮点、逻辑类型矩阵

稀疏矩阵：支持实数浮点、复数浮点、逻辑类型矩阵

字符串

字符矩阵：`s = '北太天元 Baltamatica'`，使用单引号创建

字符串：`s = "北太天元 Baltamatica"`，使用双引号创建

结构体

键 - 值类型的数据集合

使用 dot 运算符创建：`a = []; a.x = 1;`

单元数组

任意类型的数据集合，以矩阵的方式体现

使用 `{}` 运算创建：`c = {x, y, z, w};`

函数句柄

支持匿名函数，内置函数，脚本等

使用 `@` 符号创建：`f = @sin`


```
>> help
```

请使用如下关键词获取帮助:

[*] 内核提供的命令:

```

Inf      NaN      Realmax  Realmin  abs      acos      acosh     acot      acoth     acsc      acsch     addpath
all      and      angle    any      asec     asech     asin      asinh     atan      atan2     atanh     binop_add
binop_mul binop_sub blanks  blkdiag  block    brace_get brace_set cast      cd        ceil      cell
cell2struct celldisp char     chol     class    clc      clear     clear_global coeff     col
colon    complex cond     conj     convertStringsToChars corrcoeff cos      cosh     cot      coth
cov      cross   csc      csch     csvread  ctranspose cummax   cummin   cumprod  cumsum   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
det      diag    diff     dir      discard  disp     dot      double   e        eig      eps      eq
error    eval    exist    exp      expml    eye      false    felose   ferror   feval    fieldnames
fileparts find    fix      floor    fopen    format   fprintf  frewind  fseek   ftell   full     gca
gcf      gco     gdoc    ge       get      getfield global   gt       help     horzcat  i        imag
ind2sub  inf     input   int32    int64    inv      is_plugin_function isa      isbanded iscell  iscellstr
ischar  isdiag  isempty isequal isfield  isfinite ishermitian isinf   isletter ismember ismissing
isnan   isnumeric isreal  isscalar isspace  issparse isstring isstruct issymmetric istril  istriu
isvector j       join    kron     ldivide  ld1      le       length  linspace list_plugins load
load_plugin log     log10   loglp    log2     logical  logspace lower    ls      lt       lu
magic   max     mean    median  meshgrid min      mldivide mod      mpower  mrdivide mtimes  ne
newline nnz     nonzeros norm     not      null     num2cell num2str  numel   nzmax   ones    or
pi      pinv   plugin_call plugin_help polyfit  polyval  pow2     power   prod     pwd
qr      rand   randi    randn    randperm rank     rcond   rdivide  real     realloc  realmax  realmin
realpow realsqrt rem     repelem repmat  reshape  rmfield  rng      roots   round   row      rref
save    sec     sech    set      setfield sign     sin      single  sinh     size     sort     source
sparse speye  spones  sprintf  sqrt     startsWith std      str2double strcat  strcmp
strcmpi string  strlength struct  struct2cell sub2ind  sum      svd      tan      tanh     tic
times  toc     trace   transpose tril     triu     true     unique   unload_plugin upper    using

```

● help命令

help命令是最常用的帮助命令。在命令行窗口中直接输入help命令将会显示当前的帮助系统中所包含的所有项目，主要包含：内核提供的命令、脚本提供的命令、插件提供的命令、list_plugins, plugin_help("插件名") 等命令获取相应帮助关键词。



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的过程中碰到的问题。

调用格式为：

```
>> help 函数（类）名
```

命令行窗口

```
>> help polyfit
```

多项式曲线拟合。

`p = polyfit(x,y,n)` 返回次数为 `n` 的多项式 `p(x)` 的系数，该阶数是 `y` 中数据的最佳拟合（在最小二乘方式中）。`p` 中的系数按降幂排列，`p` 的长度为 `n+1`

$p(x)=p_1x_n+p_2x_{n-1}+\dots+p_nx+p_{n+1}$.

示例：

```
X = [1 2 3 4 5];  
Y = [5.5 43.1 128 290.7 498.4];  
polyfit(X,Y,3)
```

02/

北太天元插件开发

北太天元插件机制

- 数值计算软件额外功能模块的扩展

在北太天元中直接运行 C/C++ 程序

使用底层语言开发各类综合性的专业工具箱

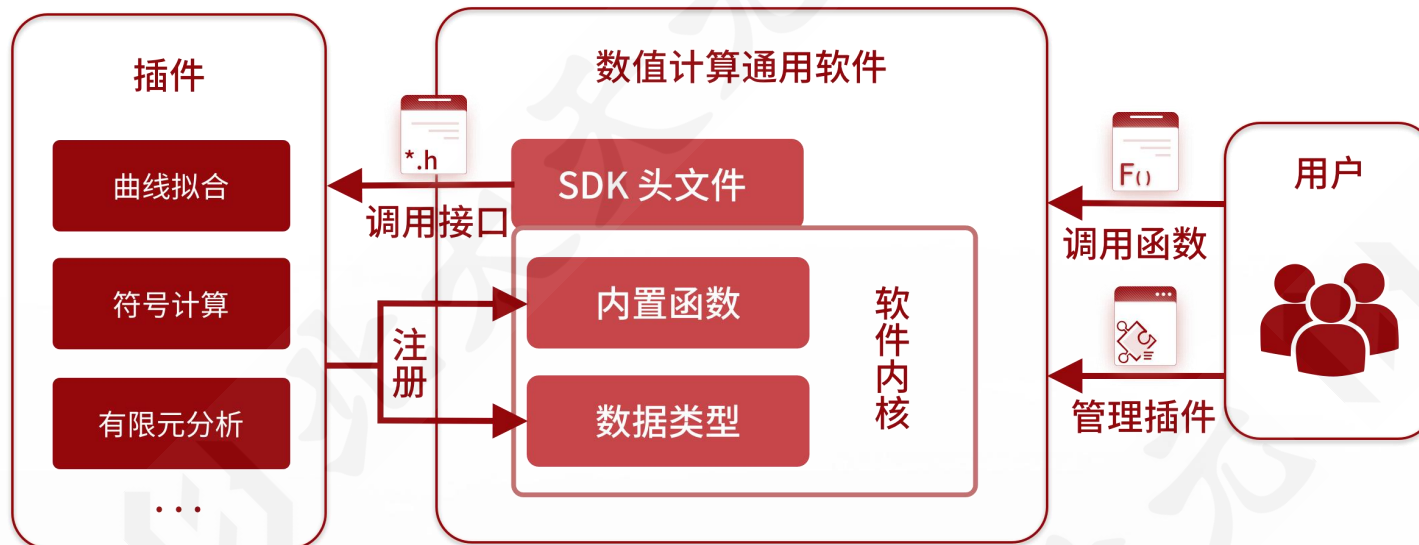
- 本质是提供一种与各类开发者协作、共享代码与数据结构的方式

软件开发者与合作开发者借助**开发工具 (SDK)** 进行**黑盒式**开发

希望合作模式尽量**松散化**，长时间（一个月以上）不交流依然可以将事情完成



插件框架



c/cpp 源码

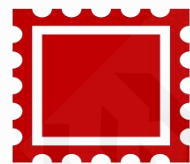


+



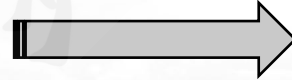
SDK

编译

GCC
clang
icc

插件

运行时载入



编译加载

插件:

- 增加内置函数 (C/C++)
- 托管自定义数据类型

非内置数据类型

```
A = mcreate(m, n);  
V = oper(A, i);
```

- 运算符重载
- 重载内置算子
- 显示窗口/增加 UI 元素

方便快速进行已有 C/C++ 代码的快速转化。

SDK:

- 编写规范
- 内部数据接口 (API)
- 注册/删除自定义类型机制
- 编译器/调试器
- 插件签名机制
- ...

- ✓ 全面支持内核各类变量操作
- ✓ 支持注册 C++ 对象, C 结构体, 重载操作符
- ✓ C 风格接口函数 100+ 个, 开发者文档 50+ 页
- ✓ 支持编译环境: Linux/gcc, Windows/mingw64
- ✓ 可独立于数值计算软件运行
- ✓ 图形化插件管理





- 添加更多 API
- 现代 C++ 支持, FORTRAN 支持
- 插件对内核的逆向调用
- 进一步**降低开发者门槛**, 制作编译器/调试器
- 插件的版本, 依赖规范
- 插件授权机制 (数字签名)

03/

北太天元插件开发

北太天元插件开发

- 社区获取

<http://www.baltamatica.com/community/sposts/detail/994d897d-b7ec-9c72-17f4-47f8b4e737f4.html>

[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](#)

社区 > SDK开发文档 > 详情

SDK开发文档资料

标签: [下载](#)

北太天元客服 2022-09-26 13:33:09

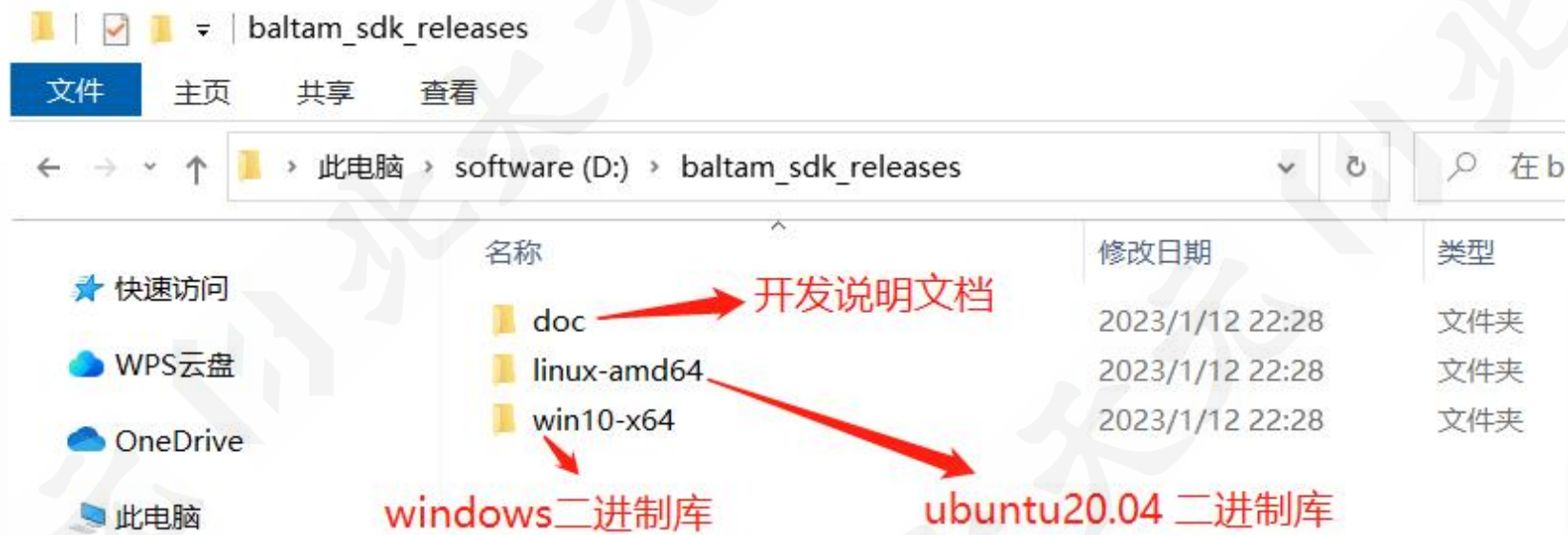
开发者用户可以将自己的开发成果利用SDK做成插件的形式使用, 我们提供SDK的使用说明文档和示例等资料供大家学习参考。

【资料】

[1.使用说明文档](#)

[2.示例](#)

文档路径



SDK开发说明文档



首页 | baltamatica_sdk_manual.pdf

文件 | 开始 | 插入 | 批注 | 编辑 | 页面 | 保护 | 转换 | 查找功能、文档内容

书签 | 书签查找

- 介绍 1
- 使用方法 2
 - 插件开发 2
 - BEX 文件 11
- 接口 11
 - 宏 11
 - 枚举类型 11
 - 数据类型 14
 - 数组基本属性 14
 - 创建、访问、修改数组数据 19
 - 复制和删除数组 43
 - 数据转化 44
 - 算符重载 46
 - 输出与调试 47
 - 外部类 47
 - 内部查询 50
- 技术细节 51
- 更新日志 51

Baltamatica SDK 接口

北京大学重庆大数据研究院 - 数值计算实验室

最近一次更新: 2022 年 8 月 29 日

1 介绍

Baltamatica 给开发者提供了底层数据的访问接口, 这使得开发者能够基于 Baltamatica 编写可拓展的 C/C++ 插件。开发者可以使用访问函数接口来和 Baltamatica 内部的数据进行交互, Baltamatica 也能够将开发者自定义的数据类型托管于命令行中, 例如用户可以在命令行中编写如下语句:

```
1 A = init(a, b, c); % A 是插件中一个不透明的对象
2 X = oper(A);
```

Baltamatica 本身不对这些对象进行操作, `init` 和 `oper` 都是插件提供的函数。

在 Baltamatica 命令行中使用插件中的函数基本流程为

```
1 % 载入名为 name 的插件, 后面是可选参数
2 ierr = load_plugin('name', ...);
3 % 使用插件中的函数 fun1, fun2, ...
A = fun1(m, n);
```

Linux开发环境: ubuntu 20.04

Windows开发环境: MSYS2 + MinGW64

绿色开发环境获取网址:

<http://www.baltamatica.com/community/sposts/detail/fb24deaf-34d6-2309-c639-920d544775fd.html>

社区 > 数值计算 > 详情

如何在Windows平台下部署北太天元插件开发环境

标签: [编程](#) [SDK](#)

 Forever 2023-01-11 16:31:28

[编辑](#) [删除](#)

由于北太天元发布的开发的插件的SDK是使用GCC编译器生成的, 不支持微软编译器的使用, 在Windows平台下必须部署对应的 GCC 编译器开发环境才能开展 c/c++ 插件的开发工作。

我们可以采用基于MSYS2的MinGW-w64 GCC搭建Windows下C++开发环境。

MSYS2是仿Linux的终端交互环境, 可以使用Linux下的终端命令和快捷方式, 自带pacman软件管理包, 让您将精力放到C++代码的编写上, 而不是复杂的依赖环境的搭建和编译, 可以复用自己在Linux下的开发经验。使用g++进行库的连接、源码编译等操作, 可更进一步了解C++代码的编译过程。

绿色便携版本的 MSYS2 + MinGW64 开发环境包传送阵如下:

链接: https://pan.baidu.com/s/15KEJ_CZpEjNo_hQ7JluAeQ

提取码: 5361

```
#include "bex.hpp"
```

插件头文件

```
void my_create(int, bxArray*[], int, const bxArray*[]);  
void my_size(int, bxArray*[], int, const bxArray*[]);  
...
```

插件导出函数声明

```
struct MyVector : public extern_obj_base {...};
```

插件自定义数据类型

```
int bxPluginInit(int, const bxArray*[]){  
    bxAddCXXClass<MyVector>(PLUGIN_NAME);  
    return 0;  
}
```

初始化句柄 (可选)

```
int bxPluginFini(){ return 0; }
```

析构句柄 (可选)

```
static fun_info_t flist[] = {  
    {"my_create", my_create, nullptr}, ...  
    {"", nullptr, nullptr},  
};  
fun_info_t * bxPluginFunctions(){  
    return flist;  
}
```

插件导出函数列表


```
void mfun_name(  
    int nlhs,  
    mxArray* plhs[],  
    int nrhs,  
    const mxArray* prhs[])
```

输出参数个数/数值 (左侧)

输入参数个数/数值 (右侧)

命令行 `>> [x, y] = mfun_name(a, b, c);`



C/C++ 代码 `mfun_name(2, [x, y], 3, [a, b, c]);`

```
// 用于实现重载 + 的声明  
// 函数名称任意, 函数签名与普通插件函数相同  
void my_add(int, bxArray*[], int, const bxArray*[]);
```

```
int bxPluginInit(int, const Data*[]){  
    bxAddCXXClass<Foo>(PLUGIN_NAME);  
    // 向内核注册重载的算符  
    bxRegisterBinaryOperator(PLUGIN_NAME, "+", Foo::ID, Foo::ID, my_add);  
    return 0;  
}
```

加法的实际定义在 my_add 中

重载加法, 左右参数都是自定义类型 Foo

```
// 实际实现, 这里从略  
void my_add(int, bxArray*[], int, const bxArray*[]){}
```

命令行/脚本中的效果:

```
>> c = a + b; % a 和 b 都是插件开发者自定义的对象
```

开发示例路径



编译指令:

```
g++ -shared -o main.dll -DBV_USE_DYN_LOADER -I .././include main.cpp
```

```
g++ -shared -o main.dll main.cpp -I .././include -L .././lib -lbex
```

视频教程网址：

<http://www.baltamatica.com/community/sposts/detail/2892a5d6-251c-a0f3-82bc-81a3de126c3e.html>

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

社区 > 视频 > 详情

bex插件教程

标签：[下载](#)

北太天元客服 2022-09-26 13:11:03

为了使开发者更好学习、使用bex插件，我们制作了bex插件教程系列教程提供给大家学习。

一、bex教程1-用北太天元安装包中的vector例子简介插件制作



感谢聆听

Thanks For Your Attention

基础软件科学研究中心
