# 公式组件详细设计文档

# 1简介

以成熟的商业化数学计算软件Matlab、Mathematica为参考，从具有典型代表意义的开源数学计算软件Scipy、ExpressionJ、Octave、GSL中，进行对比筛选，经商讨，最终确定了以Scipy为核心计算引擎进行开发，随着开发工作的逐步展开，又引入了Sympy。

开发环境：VS2010、Python、IronPython、Numpy、Scipy、Sympy

开发语言：CSharp、Python

SciPy 是一个基于Python、以Numpy为根基构建的项目，它旨在再现 MATLAB 的所有功能，优化其性能并简化与其它软件的集成，同时仍保持完全免费以及至少和 MATLAB 一样易于使用。

SciPy是用子模块的形式来组织的，这些子模块涵括了不同科学计算领域的内容。主要功能模块有：

1. constans物理和数学常数
2. cluster聚类算法
3. fftpack快速傅立叶变换程序
4. integrate集成和常微分方程求解器
5. interpolate拟合和平滑曲线
6. linalg线性代数
7. maxentropy最大熵法
8. ndimage N维图像处理
9. odr正交距离回归
10. optimize最优路径选择
11. [signal](http://docs.python.org/dev/library/signal.html#signal)信号处理
12. sparse稀疏矩阵和以及相关程序
13. spatial空间数据结构和算法
14. special特殊函数
15. stats统计上的函数和分布
16. weave C/C++ 整合（integration）

NumPy：基于Pthon的基础科学计算包，主要包括：

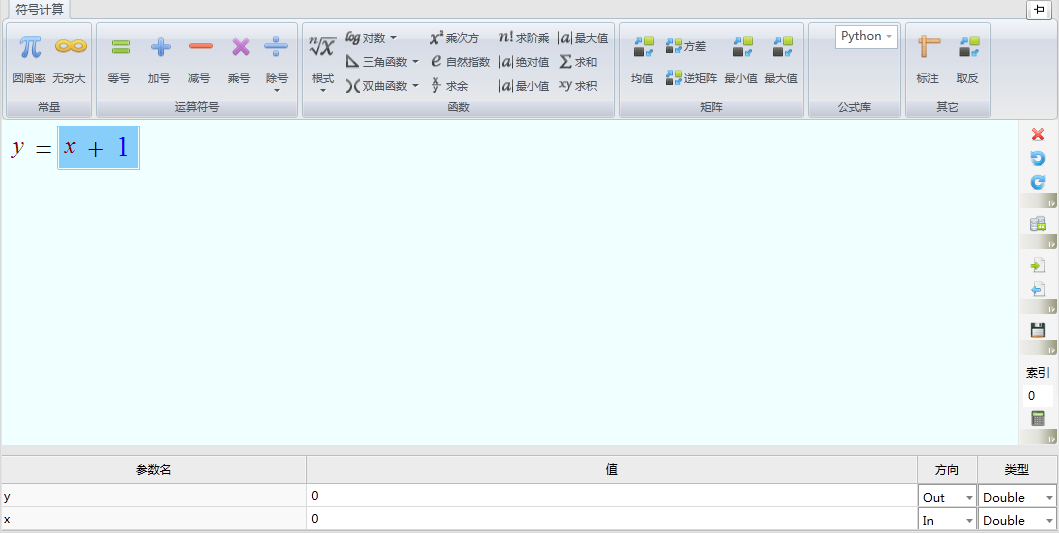
* + 1. a powerful N-dimensional array object
    2. sophisticated (broadcasting) functions
    3. tools for integrating C/C++ and Fortran code
    4. useful linear algebra, Fourier transform, and random number capabilities

Sympy是一个符号数学Python库。它的目标是成为一个全功能的计算机代数系统。Sympy完全由Python写成，不需要任何外部库，主要功能如下：

1. 核心能力：基本运算、化简、展开、函数、代换、模式匹配
2. 多项式：基本运算（除法、最大公约数等）、因式分解、合成等
3. 演算：极限、微分、积分、泰勒级数
4. 解方程：多项式方程、代数方程、微分方程、差分方程等
5. 离散数学：二项式系数、总和、积、数论、逻辑表达式等
6. 矩阵：基本计算、行列式、转置、求解、特征值/特征向量等
7. 统计：正态分布、均匀分布、概率等

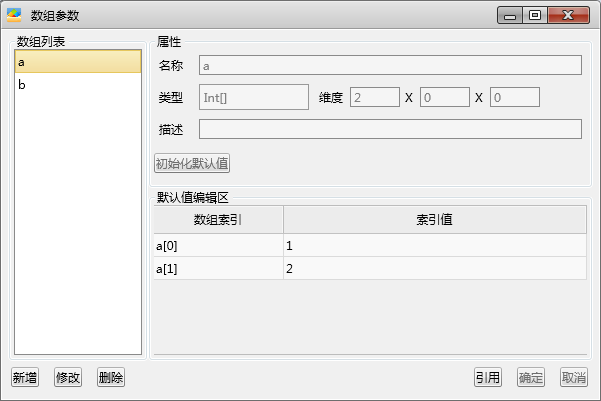
# 2总体设计

## 2.1界面设计



公式组件主界面

主界面可分为四大块，分别为功能区（包含常量、运算符号、函数、矩阵、公式库、其它）、编辑区、参数列表区、操作区；功能设计模块化，简洁明晰，便于后期扩展。



数组参数对话框

数组参数对话框针对数组进行增删改管理。

## 2.2主要功能

基本运算功能：四则运算、三角函数、反三角函数、双曲函数、反双曲函数、对数、乘方、开方、求余、阶乘、绝对值、求和、求积、最大值、最小值等

矩阵：加、减、乘、求逆、均值、方差、最大值、最小值

多项式运算：求和、求积

计算常量：PI、正负无穷大等

支持公式表达式常见的编辑操作（删除、撤消、恢复）

支持多个表达式的编辑与运算功能

支持公式表达式涉及到的变量的管理（方向、默认值、数据类型）

支持表达式的导入、导出、截图功能

支持表达式的单步计算功能

## 2.3架构设计

具体流程：公式编辑器生成的表达式，经Proxy动态生成相应的计算表达式，将运算表达式及计算变量初始值提交给计算服务，得到计算结果，运算结果经执行引擎产生输出文件。



系统架构框图

# 3重要功能模块设计

## 3.1设计时设计

### 3.1.1新建



新建流程图

配置文件FormulaEnv.xml：

定义字体集合、运算库等，在设计时依据此配置文件生成计算表达式。结点主要属性说明，type：节点类型，根据这个类型，取得默认字体、创建节点等；name：用于判断2个节点是否相等，关键字；label：在编辑页面中，默认显示的符号；fontname：label 符号所在的默认字符集名称；elemid：符号对应的函数表达式，用于运行公式。

加载图片资源：

由于公式组件存在有大量图片，并且含有公式编辑区的Winform控件，为避免在打开组件时显示不同步的问题，图片资源采用异步加载方式实现。

初始化编辑区：

获取绘制图形的Graphics对象，设置绘制图片代理等。

生成空表达式：

绘制空格（SubTreePlace）。

### 3.1.2保存



保存流程图

公式验证规则：

1. 表示中不能存在为空的情况（包含SubTreePlace）
2. 零不能做分母
3. 数组未定义
4. 表达式必须是等式，并且只能有一个等号
5. 矩阵行列式大小不匹配，并且矩阵不能与标量进行加减运算
6. 必须输入完整的表达式
7. 公式左边不允许输入表达式或常量
8. 表达式变量方向默认规则：表达式等号左边变量默认为输出或输入输出，右边变量默认为输入或输入输出，左右侧均出现为输入输出
9. 对于参数列表中变量的方向与表达式变量方向默认规则相冲突的情况，自动转换具体可参见Global文件BuildExpression方法

生成公式表达式：

依据配置文件FormulaEnv.xml的format字段，由各个表达的根结点递归调用其子节点（计算功能相关类）的OnGetExpression方法去生成表达式。

保存组件内容：

由公式计算表达式及编辑区内容两部分信息组成，通过序列化方法保存到Manifest.xml文件的RunTime结点下。

### 3.1.3打开



打开流程图

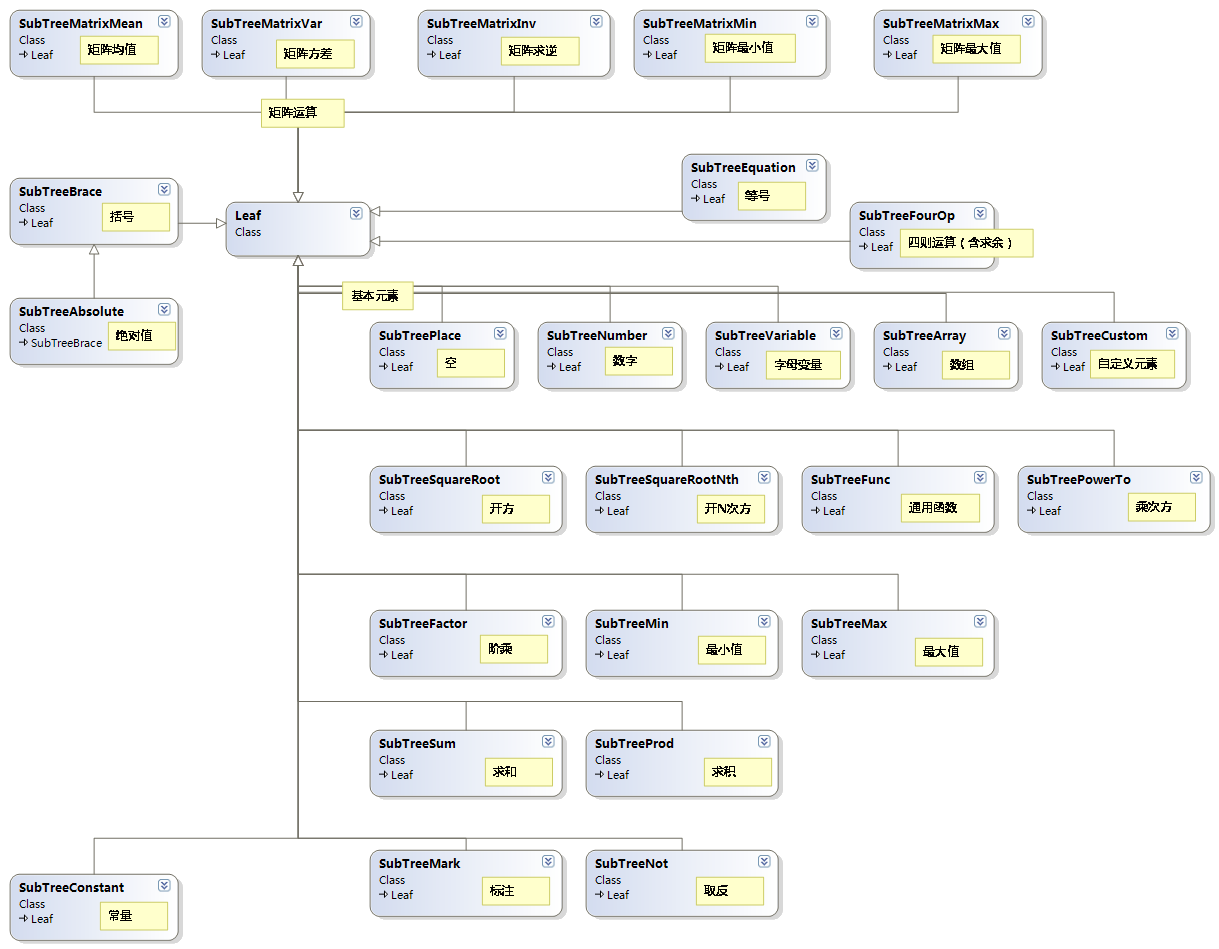
获取公式信息：

读取Manifest.xml文件的RunTime结点内容，通过反序列化方法，获取公式编辑区信息。

生成表达式：

依据获取的公式编辑区信息，绘制图形化的公式表达式。

### 3.1.4计算功能类图



计算功能类图

### 3.1.5公式编辑

对于组件涉及到的计算功能元素都有一个具体的类与其对应，具体可参考计算功能类图。

绘制思路：根据输入内容，调整输入表达式的布局，计算其内容显示区、绘制区大小，然后开始绘制计算表达式；布局调整与绘制采用递归方法，由各行计算表达式的根元素查找子元素启动递归。

绘制技术：GDI+

计算表达式的选择高亮显示：依据鼠标单击时鼠标位置是否落在计算表达式中的相应元素绘制区内，如果当前鼠标位置包含在相应计算表达式元素中，则该元素高亮显示。

表达式编辑：移动单击鼠标选中计算元素，通过键盘输入更改表达式内容。

## 3.2运行时设计



运行时流程图

参数输入：

接收由封装环境通用运行时框架传入的计算参数。

获取表达式：

读取Manifest.xml文件的RunTime结点内容，通过反序列化方法获取。

计算服务启动：

由于计算服务寄宿于Windows服务当中，故在调用计算服务时需先判断服务是否已经启动，如未启动则先尝试重启一次，再判断是否成功启动，如果仍然未能启动的话，则抛出异常，直接退出。

## 3.3计算服务



计算服务启动流程图

计算服务寄宿于Windows服务当中，通常开机时会自启动。为提高公式组件的运行性能，计算服务采用IPC技术，在服务端完成计算引擎的运行环境准备工作，包括计算类库的加载、计算对象的初始化，在服务启动时一次性完成初始化工作，在接收到后续计算请求后可直接执行运算！



调用计算服务流程图

清除引擎计算变量：

调用计算服务时会在计算引擎中形成计算关联到的变量对象，计算完成时如不清除此部分变量会常驻于计算引擎当中，随着调用计算服务的次数增多，计算引擎中的变量对象也就是愈多，占用的内存也就会随之增加。故在调用完成之后，最好清除相关变量。

涉及到的主要内容：

Flexware.Runtime.Services项目

Flexware.Python.Service项目下的PythonObject文件

## 3.4数据传递与类型转换

在利用计算引擎进行运算时，需要完成两部分工作：传递计算表达式、设置表达式涉及到的变量计算初始值。

对于非矩阵类型变量，如Int或Double，通过ScriptScope的SetVariable方法完成

对于矩阵类型变量，将其初始化操作蕴含在计算表达式当中，动态生成

计算引擎的计算结果通过ScriptScope的GetVariable方法完成，非矩阵类型变量直接利用Convert进行转换，矩阵类型变量需借助于自定义函数MatrixToString（C#中目前还没有相应的对象与计算引擎的矩阵对象相对应）

涉及到的主要内容：

Flexware.Python.Service项目下的PythonObject文件

## 3.5计算功能拓展

增加运算功能时，一般都需要添加相应的运算对象，如要实现矩阵求逆功能时候，需添加SubTreeMatrixInv对象。

在一些开源计算库与C#之间不兼容的情况、开源计算库或不支持或者不完善的情况下，一般需要自己去弥补此部分的缺陷，具体途径通过自定义脚本方式去实现。

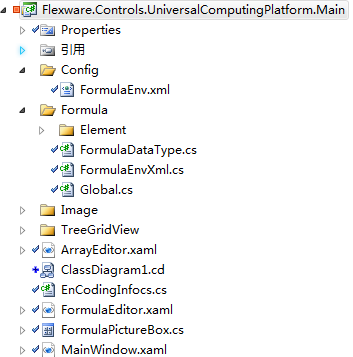
涉及到的主要内容：

Flexware.Python.Service项目下的PythonObject文件自定义脚本函数MatrixToString

# 4项目配置

## 4.1关联项目

Flexware.Controls.UniversalComputingPlatform.Main项目



此组件设计时项目

配置文件：FormulaEnv.xml

计算功能类目录：Formula

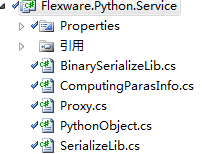
数组引用对话框：ArrayEditor.xaml

公式编辑控件：FormulaEditor.xaml

公式编辑区控件：FormulaPictureBox.cs

公式组件主控件：MainWindow.xaml

Flexware.Python.Service项目

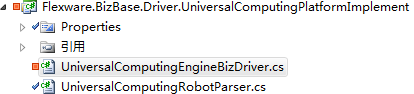


序列化类：XMLSerializeLib、BinarySerializeLib

数据转换、计算表达式生成类：Proxy

IPC计算服务共享类：PythonObject

Flexware.BizBase.Driver.UniversalComputingPlatformImplement项目



此组件运行时项目

staticResource\IronPython目录

配置文件：FormulaEnv.xml

计算库目录：Lib，需根据项目需要定制裁剪

计算功能涉及到的引用类库：

IronPython

Microsoft.Dynamic

Microsoft.Scripting

NumpyDotNet

## 4.2开源计算库

主要资源网址：

<http://python.org/>

<http://ironpython.codeplex.com/>

<http://www.numpy.org/>

<http://www.scipy.org/>

<http://code.google.com/p/sympy/>

测试环境搭建（<https://www.enthought.com/repo/.iron/>）：

1.) IronPython

Download and install [IronPython 2.7](https://www.enthought.com/repo/.iron/IronPython-2.7.msi), this will require .NET v4.0.

2.) Modify PATH

Add the install location on the path, this is usually: C:\Program File\IronPython 2.7

But on 64-bit Windows systems it is: C:\Program File (x86)\IronPython 2.7

As a check, open a Windows command prompt and go to a directory (which is not the above) and type:

> ipy -V

PythonContext 2.7.0.40 on .NET 4.0.30319.225

3.) ironpkg

Bootstrap ironpkg, which is a package install manager for binary (egg based) Python packages. Download [ironpkg-1.0.0.py](https://www.enthought.com/repo/.iron/ironpkg-1.0.0.py) and type:

> ipy ironpkg-1.0.0.py --install

Now the ironpkg command should be available:

> ironpkg -h

(some useful help text is displayed here)

4.) scipy

Installing scipy is now easy:

> ironpkg scipy

numpy-2.0.0-1.egg [downloading]

2.26 MB [.............................................................]

scipy-1.0.0-1.egg [downloading]

8.77 MB [.............................................................]

numpy-2.0.0-1.egg [installing]

6.97 MB [.............................................................]

scipy-1.0.0-1.egg [installing]

25.68 MB [.............................................................]

While downloading, the size of the (compressed) egg file is displayed, whereas, when installing, the installed size (the file-size of all installed files) is displayed. Now scipy is installed into the IronPython environment. You can try importing it by typing:

> ipy -X:Frames -c "import scipy"

5.) SymPy

依次安装python-3.3.0.msi，sympy-0.7.1.win32.exe

将Python安装目录下的文件夹sympy，比如C:\Python33\Lib\site-packages\sympy，拷贝到IronPython相应的安装目录下，比如C:\Program Files\IronPython 2.7\Lib\site-packages\

## 4.3开源计算库个性化定制

由于项目需要可能会要求删除冗余库，或者增加其它库，此时就需要依据项目需求进行个性化定制与裁剪。

目前基于上述所搭建起来的测试环境，主要进行了三方面的操作：

对冗余功能目录进行了删除操作

修改部分见注意事项

增加自定义函数，见Flexware.Python.Service项目PythonObject类的Init()函数

## 4.4注意事项

导入开源计算库numpy时，需手动修改相关文件，避免导入random，否则在计算服务初始化导入numpy时会发生错误；操作方法：numpy库(numpy\\_\_init\_\_.py文件 注释代码 #import random)