

# Bai 语言从入门到入土

基于微信小程序开发的拓展语言

主编

曾柏滔 郑惠尹

# 目录

第一章：绪论 .....	1
1.1 Bai 语言概述 .....	1
1.2 Bai 语言编译器介绍 .....	2
第二章：Bai 语言基础篇 .....	2
2.1 数据创建 .....	2
2.2 赋值语句 .....	3
2.3 输出语句 .....	4
2.4 运算表达式 .....	4
第三章：Bai 语言进阶篇 .....	7
3.1 函数图像的绘制 .....	7
3.2 几个简单的系统指令 .....	8
第四章：Bai 语言入土篇(可跳过) .....	8
4.1 Bai 语言的工作原理 .....	8
4.2 Bai 语言的运算引擎原理 .....	8
4.2.1 表达式分解&预处理操作 .....	8
4.2.2 中缀表达式转后缀表达式 .....	9
4.2.3 后缀表达式的运算(附加误差分析与系统评估) .....	10
4.2.4 表达式的优先级 .....	11
4.2.5 数学运算函数的设计简介 .....	12
参考文献 .....	14
附录 .....	15
1. 数学表达式分析计算引擎源码 .....	15
2. 数学指令分析引擎源码 .....	35

## 第一章:绪论

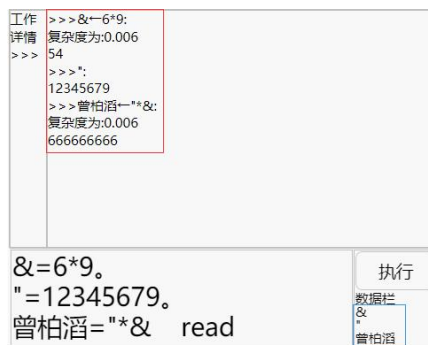
### 1.Bai 语言概述

当下,普通计算器的能力已然不能满足人们对更高级数学运算(如对矩阵、向量等复杂运算)的需求,传统编程语言如(java、JavaScript、c、c++)里的数学表达式都是对(int、float、double)等常规标准数据类型进行运算的,如果要这类编程语言来进行上述高级数学运算,需要自己设计算法编写函数,增加运算成本,实属不易,有力不从心的感觉。虽然可以通过 matlab 来进行这类计算,但 matlab 占用内存高,下载安装不易,且 matlab 表达式必须严格按照语法,才能正常运行。

因此,我挑选了 matlab 几种常用的功能,用 c 语言优先级的原理,用 JavaScript 语言,独立开发了一个 20466 字的含参数学表达式分析计算引擎,和一个 7765 字的指令分析引擎,用这两个引擎制作了程序专用运行环境,类似于一款手机版本(matlab),取名为 Bai 语言。以上工作完全自主研发,没参考任何文献以及任何算法,时间虽然仓促,但目前基本可以满足大部分用户的需求。该语言的核心是数学运算与函数图绘制,具有赋值、变量释放、运算、运算精度设置、输出等结构,不同于传统编程语言,不具有选择、判断与循环结构。

该语言属于交互式解释执行语言,但不同于(python 和 matlab)的地方在于:

- 1.命令可以散乱分布,代码格式要求很低(譬如{sin (cos pi)})这个语句,我们会认为是{sin(cos(pi))}(pi 指常量  $\pi \approx 3.1415926$ ),系统会计算得到{sin(cos(pi)=sin(-1) $\approx$ -0.8414...)};
- 2.支持一次性执行多句代码,每句代码之间用"。"中文句号连接,最后一句可不加;
- 3.所有中文字符和英文字符可以混和使用,譬如{a=[1: 2;3, 4,5]}、{sin (pi)}等
- 4.变量名随意取,只要不出现关键词和内置运算符、匹配符和函数名序列,譬如:tsin、fcosg、zcls、as √ d}均是不合法的变量名,变量名可以是任意中文汉字,甚至可以是符号,譬如{{, }, \/,&,"'...}等等,非常自由,有良好的用户体验性。



内置关键词和关键字共 12 个,分别系统指令和数据快速创建指令。分别如下:

系统指令:clear(变量释放) cls(工作日志清除) plot(绘图) setting(设置保留小数点位数) read(控制是否输出) '?'快速查询帮助

数据快速创建: diag(对角阵) ones('1'矩阵) zeros('0'矩阵) eye(单位阵) [] (自定义矩阵): (等差列向量)

内置运算符(8 个)、匹配符(2 个)和内置数学函数(22 个)

共 32 个,分别如下:

双目运算符: + - \* # / ^ % 单目运算符: √ 匹配符:( )

内置数学函数:(三角函数:sin cos tan asin acos atan) (指数与对数:exp ln lg)

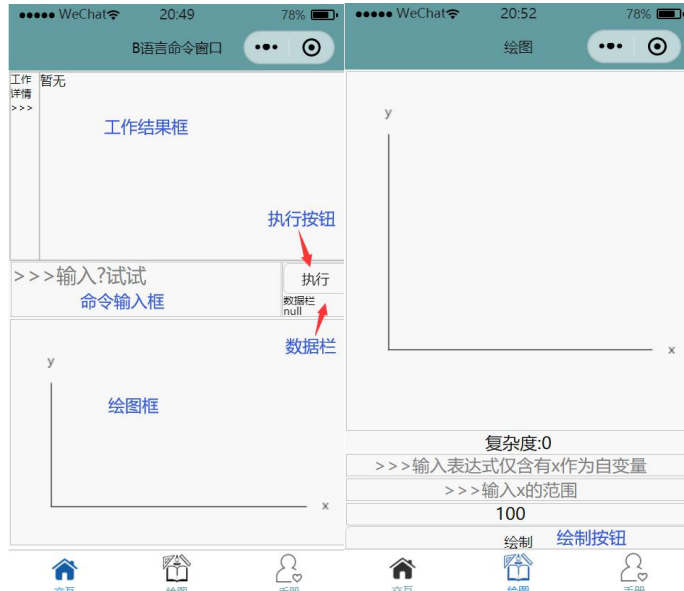
(矩阵与向量:RA(求秩) TA(转置) TN(水平翻转) UP(上下翻转) rref(化简行最简矩阵) inv(求逆) det(求行列式))

(附加函数:rand(随机数) abs(绝对值) ceil(上舍入) floor(下舍入) round(四舍五入) sqrt(开平方))

输入界面灵活,可以直接输入数学表达式,函数直接连接变量或者常数可以不加括号,绘图操作也相当简单。接下来我将详细介绍 Bai 语言语法以及交互式界面的使用方法。

## 2.Bai 语言编译器介绍

工作区有两块,分为交互区和绘图区,其中交互区也提供了绘图模块,绘图区(属于附加功能)提供了更加快捷方便的绘图功能。



交互区由工作结果框、命令输入框、执行按钮、数据栏、绘图框共 5 个部分组成,命令输入框就是我们输入指令的地方,写入指令代码后,点击执行,程序将识别和计算命令框内所有指令,同时将相关数值储存,指令的执行日志储存打印在工作结果框,新增数据名称会显示在执行按钮下面的数据栏,调用 `plot` 可以绘制图形输出在绘图框,点击绘图区相应位置可以读出当前点的函数值。工作结果框的每一条记录都可以水平和数值滚动,因此可以在有限范围内看所有数据,数据栏的变量名称亦是如此。

绘图区仅仅需要输入自变量范围和函数表达式,未知数变量固定为小写 `x`,不能出现别的数值。点击绘制,就能在上方绘制出该函数的图像了,是不是很方便呢?!

注:绘图组件采用《echarts》,一款企业级的绘图组件,以及《echarts.js》库函数完成。

## 第二章:Bai 语言基础篇

Bai 语言的核心围绕数学运算,因此学习 Bai 语言,仅需沿着输入、计算、输出的路线来学习 Bai 语言。

### 1.数据创建

Bai 语言变量创建非常容易,和 `matlab` 的方法一样。分为如下两种方式:

#### 1.通过函数快速创建

已有方法(`diag ones zeros eye` :)这些方法和 `matlab` 的方法完全相同。

`diag` 为快速创建对角矩阵的方法,写法为`{diag(a1,a2,a3...an)}`

`a1,a2...an` 均为数字常量,不能含有字母,系统自定义常量 `e` 和 `pi` 在这里也不能使用,这条语句将会生成一个对角线元素依次为`(a1,a2...an)`的矩阵

`ones`、`zeros` 为快速创建元素全为 1 或者全为 0 的矩阵 写法为`{ones(m)/zeros(m)或 ones(m,n)/zeros(m,n)}`

`ones(m)`和 `zeros(m)`均创建 `m*m` 阶元素分别为 1 和 0 的方阵

`ones(m,n)`和 `zeros(m,n)`均为创建 `m*n` 阶元素分别为 1 和 0 的矩阵

`eye` 为快速创建单位矩阵的方法,写法为`{eye(m)或 eye(m,n)}`

`eye(m)`为创建 `m` 阶单位阵 `eye(m,n)`为创建 `m*n` 阶单位矩阵(其中 `i=j` 处元素为 1,其余为 0)

‘:’ 为生成等间距的矩阵 写法为:{a0:a1 或者 a1:d:a2}

其中 a0:a1 将会生成一个列向量,第一个元素为 a0,之后每一项增加 1,最后一项为不超过 a1 的数(a0+n)

其中 a0:d:a1 将会生成一个列向量,第一个元素为 a0,之后每一项增加 d,最后一项为不超过 a1 的数(a0+n)



上图测试了所有数据创建语句,工作记录较长,因此截了三张图

## 2.用户输入

方法和 matlab 的一致,同时我给出了补充

关键词为一对中括号,可以是中文符号,譬如[1,2,3;4,5,6]或者【1; 2; 3, 4, 5; 6】...

‘;’分割了每行元素 ‘,’分割了列元素。

如第一个[1,2,3;4,5,6]会得到矩阵:

```
1      2      3
4      5      6
```

同时我做出了补充,生成的矩阵没有写元素的位置会自动补全 0

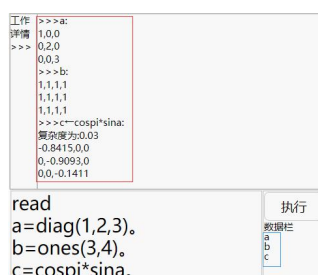
如第二个【1; 2; 3, 4, 5; 6】会得到如下矩阵:

```
1      0      0
2      0      0
3      4      5
6      0      0
```

## 2.赋值语句

Bai 语言赋值语句非常简单,‘=’等于号作为赋值运算符,居右左结合性,将等于号右边计算出的值赋值给左边的变量

如{a=diag(a1,a2,...an)、b=ones(3,4)、c=cospi\*sina...}



如左图所示界面,创建了三组数据并赋值,其中第三个表达式 {c=cospi\*sina}为数学表达式,运算结果赋值给 c

注:代码中的{read}关键词的含义为执行表达式的同时,输出表达式结果,方便用户进行查询结果。

### 3. 输出语句

Bai 语言输出表达式的值和查询变量的值非常容易,不写等于号,直接写变量名,或者直接写数学表达式,或者赋值给一个空(如{ =sinpi }),或者在命令框中任意位置加入 `read`,就可以在执行之后看到结果啦。

<pre> &gt;&gt;&gt;b=randa: 复杂度为:0.012 &gt;&gt;&gt;c=randb: 复杂度为:0.012 &gt;&gt;&gt;ans: 0.2563,0.2783,0.8138 0.6324,0.4011,0.1158 &gt;&gt;&gt;ans: 0.0526,0.0415,0.3857 0.0642,0.0489,0.0768 &gt;&gt;&gt;ans=rrefb: 复杂度为:0.012 1.0,-4.0192 0.1,6.6257 </pre> <p>a=ones(2,3)。 b=randa。 c=randb。 =b。 c。 rrefb</p>	<pre> &gt;&gt;&gt;a: 1,1,1 1,1,1 &gt;&gt;&gt;b=randa: 复杂度为:0.012 0.2966,0.3462,0.8497 0.0833,0.9987,0.3784 &gt;&gt;&gt;c=randb: 复杂度为:0.012 0.2638,0.0421,0.7827 0.0083,0.3516,0.278 </pre> <p>a=ones(2,3)。 b=randa。 c=randb。 read</p>
<p>执行</p> <p>数据栏</p> <p>a b ans c</p>	<p>执行</p> <p>数据栏</p> <p>a b c</p>

对比两个图,不难发现左边没有加 `read` 关键词,所以前三个赋值语句没有输出,只输出了后面的参数和表达式的值,右边加入了 `read` 关键词,因此赋值语句也会输出。

### 4. 运算表达式

运算为整个 Bai 语言的核心部分之一,其实操作都相当容易,数学表达式和我们平常写的数学表达式一致,有些函数只需要查表就可以使用。

首先重申运算符 `+-*/^%` 其中 `+-*/^%` 均为常见运算符,对常数的运算此处不再做更多介绍,下面介绍程序的运算(由于向量属于特殊的矩阵,因此不专门介绍向量了),包括我为了适应计算定义的新的运算。

以下的运算都是包含在表达式体系中的运算,如(1)的常数与矩阵,指的是自定义变量类型为常数或者系统自带常量 `e` 和 `pi`,矩阵也是自定义数据类型为矩阵或者向量。

#### 1. 常数与矩阵的运算

- (1) 常数与矩阵相加减 定义为常数与矩阵每个值的加减
- (2) 常数与矩阵乘法 有两个运算符 `*` `#` 均定义常数与矩阵每个值相乘
- (3) 常数除以矩阵,定义为常数除以矩阵每个值
- (4) 矩阵除以常数,定义为矩阵每个值除以常数
- (5) 矩阵的常数次方(`A^c`),定义为矩阵每个值的 `c` 次方
- (6) 常数的矩阵次方(`c^A`),定义为 `c` 与矩阵所有值进行次方运算
- (7) 矩阵对常数求余数(`A%c`),定义为矩阵每个值对 `c` 求余数
- (8) 常数对矩阵求余数(`c%A`),定义为常数度矩阵每个值求余数

#### 2. 矩阵与矩阵的运算(此处较 matlab 进行了改进)

先介绍矩阵公共区概念: `A(m1*n1)` `B(m2*n2)` 进行运算,那么公共区域指的是 `(min(m1,m2)*min(n1,n2))` 的区域。比如一个 `3*4` 矩阵与 `6*2` 矩阵进行相加运算,那么会返回得到`(3*2)`的矩阵。

- (1) 矩阵与矩阵加减,定义为将两个矩阵公共区域进行对应加减并返回该区域
- (2) 矩阵`*`运算,就是线性代数里的矩阵相乘运算必须满足前一个矩阵的列数等于后一个矩阵的行数,否则结果可能无意义
- (3) 矩阵`#`运算,等同于 matlab 的`.*`点乘运算定义为,将两个矩阵的公共区域每个元素对应相乘
- (4) 矩阵除以矩阵,矩阵间求余,求乘方均同于以上定义,为区域值对应计算。

## 3.数学函数的运算

数学函数均可以输入矩阵或者常数、向量都会返回同类型的求解值。下面为函数表:

三角函数类:	sin	cos	tan	asin	acos	atan	
	正弦	余弦	正切	反正弦	反余弦	反正切	
对数、指数、附加类:	exp	ln	lg		sqrt	√	abs
	以 e 为底的指数	以 e 为底的对数	以 10 为底的对数		开绝对值的平方根	开绝对值平方根	绝对值函数
矩阵运算类:	RA	TA	TN	UP	rref	inv	det
						矩阵求逆, 遇到方阵直接求逆, 否则取	矩阵求行列式, 遇到向量返回第一个元素值, 遇到方阵返回行列式的值, 否则取 min(m,n) 组成方阵求行列式的值
(输入矩阵)	矩阵求秩返回整数	将矩阵转置	将矩阵水平翻转	将矩阵上下翻转	将矩阵化简为最简型	为 min(m,n) 组成的方阵进行求逆, 遇到不可逆, 返回 (‘不可逆’), 否则返回逆矩阵	
(输入常数)	输入 0 返回 0 否则返回 1	返回该常数	返回该常数	返回该常数	输入 0 返回 0 否则返回 1	返回这个数字的倒数值	返回该常数
补充类函数:	rand	ceil	floor	round			
	生成随机数, 如 Rand(3) 为生成 0 至 3 的随机数	上舍入函数	下舍入函数	四舍五入函数			

## Bai 语言从入门到入土



左上图先生成  $a$ 、 $b$  两个  $1$  矩阵,再利用 `rand` 创建随机矩阵作为测试数据,之后求取秩为  $2$ ,化简了行初等型,同时求了逆,计算均是瞬间完成。

右上图进行了复杂的向量代数运算。



## 第三章:Bai 语言进阶篇

### 1. 函数图像的绘制

Bai 语言的核心功能 2 是轻松绘制函数图像,暂不支持散点图,采取的是折线图的方式。一共有两种方式进行绘制函数图像,第一种就是在交互界面用指令 `plot` 进行画图,另一种则是通过绘图界面快速绘图,`plot` 的功能更加齐全,绘图界面的绘制更加简洁明了。

#### (1)plot 指令画图。

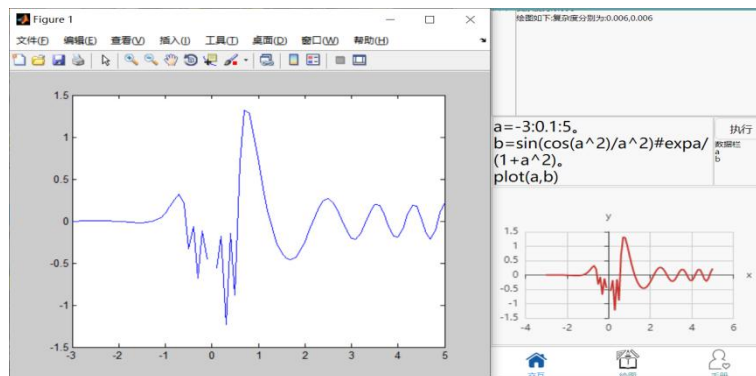
`plot` 采用的语法与 `matlab` 的语法相同 `plot(x,y)`

其中 `x,y` 可以是向量,也可以是矩阵,还可以是复杂的表达式,建议 `x` 从小到大排列,可以用 `':'` 表达式先生成自变量范围。

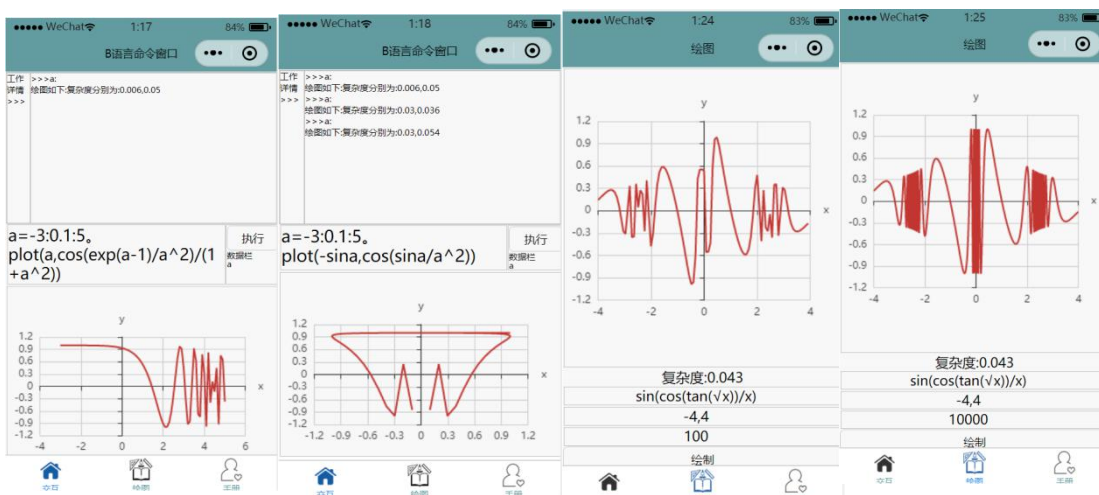
#### (2)绘图界面绘图。

绘图界面输入带 `x` 的表达式,输入 `x` 的范围(中间用 `':'` 逗号分隔),再点击 绘图即可。其中界面内的 100 为采样点数,上限为 1w,下限为 1,值越大渲染越慢,但越圆滑。

个人建议 100 足以满足需求,往往没必要增大计算量。且这个页面\*等效为`#`运算,可以放心写表达式。



该图与 `matlab` 的图进行对照,证明结果计算是准确的,同时绘制也是准确的。



右边两幅就是采样 100 次和 10000 次,一般情况下,不涉及太多震荡不必要采样如此多。

## 2. 几个简单的系统指令

- (1) **clear 指令** 对数据进行单个或者批量删除  
语法为:{clear(a,b...n)}或者 clear}前者会对所有括号内的变量名以及对应值进行删除,后者则删除所有变量。
- (2) **Cls 指令** 清空工作详情所有内容
- (3) **setting 指令** 语法为 setting 加 0 到 9 的任何数字,修改程序运算结果保留的小数点位数。譬如{setting6}将会保留 6 位小数。
- (4) **read 指令** 在工作栏任意位置处加 read,会输出所有指令的运算结果,包括变量创建的结果。
- (5) **'?'指令** 单一的'?'会打开手册界面,? 加关键词会在工作栏输出该关键词作用,遇到运算符会输出运算符计算功能和运算定义,遇到函数则会显示函数功能以及参数内容

## 第四章:Bai 语言入土篇

### Bai 语言的工作原理

#### 1.指令预处理

首先要将输入的指令文本进行预处理,将文本内可能出现的中文字符全部替换为英文字符,删除空格符与换行符,同时将文本用"。"分解为多句语句

#### 2.指令分析

将拆分得到的指令遍历,依次查找单个指令是否含有关键词和关键字,从而将指令进行分类,返回对应的指令码

#### 3.数据提取

将拆分得到的指令通过指令码所对应的方式提取指令的数据

#### 4.指令&数据处理

若指令为系统指令,则执行相应的系统修改操作;

若指令为输入指令,则获取是否输入赋值变量的名字,同时获取是哪种创建数据的方式,以对应的方式将数据进行数据生成,储存在数组内,将处理过程的信息、得到的变量名称(若没有返回 ans)与执行结果合并返回

#### 5.输出

获取执行结果返回的列表,检查处理结果是否正确,输出运行日志,若运算结果正常,且检测 read 关键词或该值储存的变量名为'ans',打印运算结果至工作框。

### Bai 语言的运算引擎原理

运算处理是整个编译器工作的核心,是该编译器能正确执行指令的关键所在。对含参数学表达式进行求解运算,属于该 20466 字的数学引擎(runcacular.js)的核心功能。

这个算法的关键将所有数学函数都当作是一个右结合的运算符,这是我的独创的一种计算处理方式(不知道是否已经有这种方式,我没去搜索相关资料,一切数学操作均独立设计编码完成)。

#### 1.表达式分解&预处理操作

这一步的操作是将表达式所有独立字符串分开形成一个列表,运用 js 正则表达式的方法,快速处理字符串

代码:{

```
var regex = /(\\+|\\-|\\*|\\/|\\~|\\^|\\%|\\#)+/; //正则表达式创建
var array = ts.split(regex); //将文本按照正则分解
}
```

譬如表达式 $-\sin(\cos(x^2-3))+b*(+1+x)^a$ 将会分解得到列表:

`['-', 'sin', '(', 'cos', '(', 'x', '^', '2', '-', '3', ')', ')', '+', 'b', '*', '(', '+', '1', '+', 'x', ')', '^', 'a']`  
得到的这个列表先进行分析,将表达式内容进行遍历分类,将所有变量名称和常数值提取出来,依次放进一个数据列表,这个列表的初始值为`['0', '-1', '1', 'pi', 'e']`,装入列表的过程中,将表达式原来的值替换为该数据在此列表下的索引值。得到如下:

数组`['0', '-1', '1', 'pi', 'e', 'x', '2', '3', 'b', 'a']`

下标`['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']`

表达式处理结果:

`['-', 'sin', '(', 'cos', '(', '5', '^', '6', '-', '7', ')', ')', '+', '8', '*', '(', '+', '2', '+', '5', ')', '^', '9']`

为了能直接处理负数运算,这里还需要将所有负数运算符(不同于减法)修改

如果数组第一个符号为`'+'`,则删去,如果为`'-'`,则将符号项替换为`['1', '*']`,此时的`1`不是数字`1`,而是数据数组元素`'-1'`的下标;

同时,为了避免初始化分解字符串时,产生空数据项,应当在此处重新检测过滤数组,并将所有`['-', '-']`转化为`['1', '*']`、`['+', '+']`转化为`['1']`,得到如下表达式。

`['1', '*', 'sin', '(', 'cos', '(', '5', '^', '6', '-', '7', ')', ')', '+', '8', '*', '(', '2', '+', '5', ')', '^', '9']`

## 2.中缀表达式转后缀表达式

我们平常用的数学表达式均是采用中缀表达式来书写的,优点是能一眼看出表达式的结构从而快速构思计算,但这一种结构在计算机体系内是不容易被程序理解和执行的。后缀表达式(亦称逆波兰式)在此处的优势就非常明显了。举个简单的例子吧。比如中缀表达式  $3*(1+3/2^2)-1$ ,我们根据先计算优先级高的括号内的表达式运算,再执行乘法运算,最后执行加减运算,即先计算 $(1+3/x^2)$ 的内容,再将计算结果乘以`3`最后再将所得减去`1`,在后缀表达式中,该表达式书写为

`3 1 3 2 2 ^ / + * 1 -` 计算只需要从第一位取出到第一个运算符,单目再取出左边一位进行运算,双目取出左边两位进行运算,当最后表达式剩下`1`项时,输出该项的值,即为最终计算结果。

计算优先级从左到右递减。

如`2 3 4 - *`表示`3`先与`4`相减得到`-1`,表达式剩下`2 -1 *`此时 $2*(-1)=-2$ 为最终结果,相应的中缀表达式为 $2*(3-4)$

如`2 3 4 * -`表示`3`先与`4`相乘得到`12`,表达式剩下`2 12 -`此时 $2-12=-10$ 为最终结果,相应的中缀表达式为 $2-3*4$

中缀转后缀的方法,是利用不同运算符之间的优先级,我在处理过程中,将所有符号的优先级定量化,使得判断进行会更加方便。如下表:

(	)	+	-	*	#	/	^	%	sin	cos	...	round
3	-1	1	1	2	2	2	2.5	2.5	2.7	2.7	2.7	2.7

注:'...'代表剩下所有函数

创建空列表 `f:[]` `g:[]` //f 储存最终表达式 g 储存符号

从左往右依次取出表达式列表项直到取完所有元素,如果为常数,则之间添加到 `f` 的末尾,如果为符号,则取出 `g` 的末尾元素,比较优先级大小,如果大于 `g` 末尾元素的优先级,则添加在 `g` 末尾,否则依次输出 `g` 的最后一个元素,直到遇到取出的 `g` 的优先级小于该元素的优先级,则添加该元素至 `g` 的末尾。

当所有元素被取出之后,再将 `g` 内元素依次从末尾取出,添加再 `f` 的末尾得到后缀表达式。但这个表达式还需要被过滤一次,为了防止用户的错误输入,譬如括号未完全匹配,导致后缀表达式内出现`'('`类或者空列表项,让之后运算陷入死循环,我用循环将

列表内所有空项和所有元素值为 '(' 的项删除。此刻得到的表达式将被用来接下来的计算。

## 3. 后缀表达式的运算

循环从第一项遍历到第一个运算符,根据运算符类型,取出对应数量的操作数,单目运算符(函数类)取出 1 项,双目运算符取出两项,再代入专门的求解函数进行运算,运算结果储存再数值表的末尾,同时修改后缀表达式,再次从第一项开始遍历。

举个简单的例子:

已有中缀表达式:  $a + b * \sin(2 - a^2)$

转成后缀表达式:  $0\ 1\ 2\ 0\ 2\ ^\wedge - \sin * +$

和对应的数值表:  $[a, b, 2]$

- ①  $i=0$  取到第一个运算符 '^', 判断为双目运算符, 然后取出 0 和 2 两个操作数, 这两个操作数对应了数值表的 a 和 2, 那么调用计算函数, 计算  $c=a^2$ , 返回后, 将后缀表达式提取处重新修改, 让 i 再次取 0 得到:

后缀表达式:  $0\ 1\ 2\ 3 - \sin * +$

对应数值表:  $[a, b, 2, c]$

- ② 再次从  $i=0$  取到第一个运算符 '-', 判断为双目, 计算  $d=2-c$ , 修改得:

后缀表达式:  $0\ 1\ 4 \sin * +$

对应数值表:  $[a, b, 2, c, d]$

- ③ 再次遍历, 取第一个运算符 'sin', 判断为单目运算符, 计算  $e=\sin(d)$ , 修改得:

后缀表达式:  $0\ 1\ 5 * +$

对应数值表:  $[a, b, 2, c, d, e]$

- ④ 运算  $f=b*e$ , 得

后缀表达式:  $0\ 6 +$

对应数值表:  $[a, b, 2, c, d, e, f]$

- ⑤ 运算  $g=a+f$ , 得

后缀表达式 7

对应数值表:  $[a, b, 2, c, d, e, f, g]$

此时表达式只剩下 1 项, 那么返回 7 对应的 g, 就是表达式运算的最终结果。

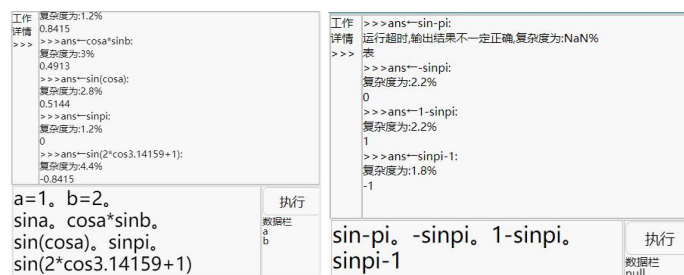
当然了, 如果用户输入不当, 可能会造成运算过程中运算符的亏损, 导致无法结束循环的情况(这是一种异常凶险的情况, 循环无法结束导致内存溢出, 可能会造成系统卡顿等一系列问题), 因此, 需要限制总运算长度, 我在此处限制总循环次数需小于等于 500 次, 定义复杂度为:  $\text{循环总次数} / 500 * 100\%$ , 一般反复嵌套和含有很多运算符的表达式, 正常情况基本不会超过 20% 的复杂度, 更多甚至会在 5% 以下, 这些运算的结果不会有错, 但最终输出的复杂度为 1 或者 NaN 的表达式, 有两种可能, 第一种为符号过多, 目前我加的限制条件为 500 次。

均为同级的运算符, 那么实验测试最多可以放入 167 个运算符, 如下图所示: 输入了 166 个 + 运算, 显示复杂度为 99.6%, 计算结果为 167 显然正确, 在输入第 167 个加 1 之后, 最终结果依旧为 167, 显然少加了一个 1。我的算法是复杂度上溢之后, 之间返回第一个元素的应的结果, 不再进行接下来的运算。



均为嵌套调用则最多调用 19 层,第 20 层开始不再进行递归计算。

还有一种出错可能便是运算符与运算符直接相连,我构建的整个中,函数都是属于运算符的,运算符必须与运算数直接相连,若出现单目运算符直接连接双目运算符,会导致出错。这个体系内,运算符与运算数直接相连是合法的,因此一些函数直接连接变量或者常量是正常运行的,如:sina、cosa\*sinb、sin(cosa)、sinpi、sin(2\*cos3.14159+1)...这些都是合法的但:sin-a 就是典型的单目运算符与双目运算符顺次相连,'sin'和'-'直接相连会导致出错。



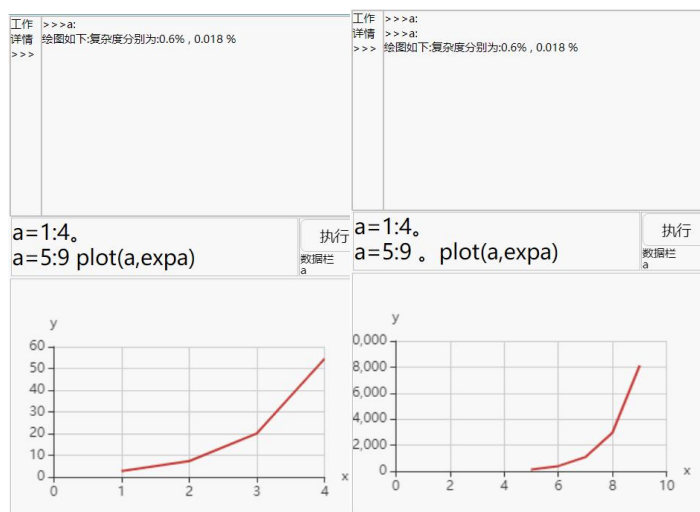
图中单目运算符连接双目报错,但双目连接单目可以正常运行。

## 4.表达式的优先级

在程序语法分析器内,首先会进行表达式的关键词分析,那么就会产生语法检测优先级。首先,逐条获取指令之后,会对该指令进行系统命令关键词查找,查找到存在,立马反馈结果。查找的优先级为{clear,cls,plot,?,setting,diag,ones,zeros,eye,[],:}

若有指令写为 cls ones(3,4),那么不会执行 ones 的内容,直接清除掉数据。

其次是检测是否含有运算符,如果存在,则反馈为运算表达式,最后进行输出指令的判断,如果都不存在,则反馈指令不存在。



左图 plot 前为加'.'句号,将 a=5:9 和 plot 理解为一句话,那么判断优先级,将会执行画图而不是将 a 重新修改为 5 至 9 的列向量。

右图 plot 前加了'.'句号,将 a 再次赋值,再执行绘图语句。

## 5.数学运算函数设计

数学运算函数的设计在整个引擎中占最长篇幅,输入要计算的参数之后,首先用函数判断该参数的数据类型:(常数类型或者矩阵类型),再给出每一种情况下的运算法则。这里就不做具体介绍了,截取一个矩阵化简初等形的函数片段吧。

整套运算思路采取高斯消元法,但我做了一些变化。

```
case 19:/'rref'      cb2 为输入是否为常数,是为 true,否则为 false
    if(cb2){//输入常数遇到 0 则返回 0,否则返回 1
        if(dat2*1==0){
            rt=0;
        }else{
            rt=1;
        }
        break;
    }else{//说明类型是矩阵
        rt=gave(dat2);//将数据类型从文本转化为矩阵
        var m=rt.length,n=rt[0].length;
        for(var i=0;i<n;i++){//这个嵌套循环将每一行的第一个不为 0 的元素化为 1 其它的元素分别除以这个数字
            for(var j=i;j<m;j++){
                var k=i;
                for(k=i;k<n;k++){
                    if(Math.abs(rt[j][k])>0.000001){
                        break;
                    }
                }
                for(var u=k+1;u<n;u++){
                    rt[j][u]=rt[j][u]*1.0/rt[j][k];
                }
                if(k!=n){
                    rt[j][k]=1;
                }
            }
            var k=i;
            for(k=i;k<m;k++){
                if(rt[k][i]*1!=0){
                    break;
                }
            }
            if(k==m){break;}
            var t=rt[k];rt[k]=rt[i];rt[i]=t;//取出第一个元素不为 0 的行方在最顶上,并将该列所有元素消去
            for(var j=i+1;j<m;j++){
                if(rt[j][i]*1!=0){
                    for(var u=i;u<n;u++){
                        rt[j][u]-=rt[i][u];
                    }
                }
            }
        }
    }
}
```

```
        }
    }
}
for(var i=1;i<m;i++){//再从第二行第一个非 0 元素往上消去，依次取到最后一行。
    var t=rt[i].indexOf(1);
    if(t==-1){break;}
    for(var l=0;l<i;l++){
        for(var j=t+1;j<n;j++){
            rt[l][j]=rt[l][j]-rt[l][t]*rt[i][j];
        }
        rt[l][t]=0;
    }
}
}
break;
```

## 参考文献

- [1]. [https://blog.csdn.net/sinat\\_27908213/article/details/80273557](https://blog.csdn.net/sinat_27908213/article/details/80273557)
- [2]. <https://wenku.baidu.com/view/9d0b706727d3240c8447ef84.html>



## 附录:

### 1.数学表达式分析计算引擎(runcacular.js)

```

var baoliu=10000;
function toRPolish(ts,datsn,dats,baoli){
    baoliu=baoli;
    var regex = /(\\+|\\-|\\*|\\/|\\^|\\%|\\#)+/;ts = ts.replace(/sqrt/g,'v');
    ts = ts.replace(/ ( /g,'(');ts = ts.replace(/ /g,')');ts = ts.replace(/\\+ /g,'');
    ts = ts.replace(/\\r\\n/g,'');ts = ts.replace(/ /g,'');
    ts = ts.replace(/sin/g,'1#sin~');ts = ts.replace(/cos/g,'1#cos~');ts = ts.replace(/tan/g,'1#tan~');
    ts = ts.replace(/asin/g,'1#asin~');
    ts = ts.replace(/acos/g,'1#acos~');ts = ts.replace(/atan/g,'1#atan~');ts = ts.replace(/exp/g,'1#
exp~');ts = ts.replace(/lg/g,'1#lg~');
    ts = ts.replace(/ln/g,'1#ln~');ts = ts.replace(/RA/g,'1#RA~');ts = ts.replace(/TA/g,'1#TA~');ts =
    ts.replace(/TN/g,'1#TN~');
    ts = ts.replace(/UP/g,'1#UP~');ts = ts.replace(/rref/g,'1#rref~');ts = ts.replace(/inv/g,'1#inv~');
    ts = ts.replace(/det/g,'1#det~');
    ts = ts.replace(/v/g,'1#v~');ts = ts.replace(/rand/g,'1#rand~');ts = ts.replace(/abs/g,'1#abs~');
    ts = ts.replace(/ceil/g,'1#ceil~');
    ts = ts.replace(/floor/g,'1#floor~');ts = ts.replace(/round/g,'1#round~');ts = ts.replace(/\\(\\)/g,'~
');ts = ts.replace(/\\(\\)/g,'~');
    ts = ts.replace(/~/g,'~');
    var name=['0','-1','1','pi','e'];
    var thdats=[];
    var array = ts.split(regex);
    var arr1=[];
    var j=5;
    var
    key=['(',')','+','-','*','#','/','^','%','sin','cos','tan','asin','acos','atan','exp','ln','lg','RA','TA','TN','UP','rref
','inv','det','v','rand','abs','ceil','floor','round'];
    for(var i in array){
        if(array[i]!='~'&&array[i]!=''){
            arr1.push(array[i]);
        }
    }
    console.log('表达式分解:'+arr1);
    for(var i in arr1){
        if(key.indexOf(arr1[i])!=-1){

```

```

        if(name.indexOf(arr1[i])!=-1){
            name.push(arr1[i]);
            var t=[];
            t.push(j);
            arr1.splice(i,1,t);
            j++;
        }else{
            var t=[];
            t.push(name.indexOf(arr1[i]));
            arr1.splice(i,1,t);
        }
    }
}
if(arr1[0]=='-'){
    arr1.splice(0,0,'0');
}
if(arr1[0]=='+'){
    var a=[];
    a.push(arr1[1]);
    arr1.splice(0,2,a);
}
for(var i=0;i<arr1.length;i++){
    if(arr1[i]=='('){
        if(arr1[i+1]=='-'){
            var ths=[];
            ths.push('1');ths.push('*');
            ths.push(arr1[i+2]);
            arr1.splice(i+1,2,ths);
            i=i-1;
        }
        if(arr1[i+1]=='+'){
            arr1.splice(i+1,2,arr1[i+2]);
            i=i-1;
        }
    }
}
console.log('此刻的表达式:'+arr1);
console.log('name:'+name);
for(i in name){
    if(isNaN(name[i]+'')){
        name[i]=dats[datsn.indexOf(name[i])];
        i--;
    }
}
name[0]='0';name[1]='-1.0';name[2]='1.0';name[3]='3.141592653589793';name[4]='2.7182

```

[illegible]

---

```

        if(a=='(' || key.length==0){break;}
        num.push(a);
    }
    a=key.pop();
    if(fun.indexOf(a)!=-1){
        num.push(a);
    }else{
        key.push(a);
    }
}
if(key1[n]!=-1){
    while(true){
        cd=keys.indexOf(key[key.length-1]);

        if(key1[n]>key2[cd] || key[key.length-1]=='(' || key.length==0){break;}
        a=key.pop();
        num.push(a);
    }
    key.push(ts);
}
}
}
}
while(true){
    if(key.length==0){break;}
    a=key.pop();
    num.push(a);
}
for(var i=0;i<num.length;i++){
    if(num[i]=='(' || num[i]==')' || num[i]==undefined || num[i]==''){
        num.splice(i,1);
    }
}
console.log('后缀表达式检测:'+num);
return num;
}
function cacular(num,name){
    var
    keys=['(',')','*','#','/','%','^','+','-','sin','cos','tan','asin','acos','atan','exp','ln','lg','RA','TA','TN','UP','rref','inv','det','V','rand','abs','ceil','floor','round'];
    let
    fun=['sin','cos','tan','asin','acos','atan','exp','ln','lg','RA','TA','TN','UP','rref','inv','det','V','rand','abs','ceil','floor','round'];

```

```

var i=0;var j=0;//console.log('传入参数检测:'+num+'、'+name);
let smu=['*','/','%','^','+','-','#'];
while(num.length>1&&j<500){

    if(fun.indexOf(num[0])!=-1 || smu.indexOf(num[0])!=-1 || smu.indexOf(num[1])!=-1){return '
表达式错误';}
    if(keys.indexOf(num[i])==-1){
        i++;
    }else{
        console.log('num 的情况:'+num);
        console.log('数据情况:'+name);
        var a=1.0,b=1.0;
        if(i>0){
            a=name[num[i-1]*1];
        }
        if(i>1){
            b=name[num[i-2]*1];
        }
        console.log('输入参数:'+b+', '+a);
        var rst=func(num[i]+'',b,a);
        if(rst=='出现降秩矩阵'){
            return '表达式运算出现降秩矩阵求逆';
        }
        if(fun.indexOf(num[i])==-1){
            name.push(rst);
            num.splice(i-2,3,(name.length-1)+'');
        }else{
            name.push(rst);
            num.splice(i-1,2,(name.length-1)+'');
        }
        i=0;
    }
    j++;
}
return [name[num[0]],j];
}

function func(fname,dat3,dat4){
    var rt;//console.log('计算输入检测:'+dat3+'、'+dat4);
    var
fuc=['*','/','%','^','+','-','sin','cos','tan','asin','acos','atan','exp','lg','ln','RA','TA','TN','UP','rref','inv','d
et','V','rand','abs','ceil','floor','round','#'];
    var ll=fuc.indexOf(fname);var cb1=!isNaN(dat3+'');var cb2=!isNaN(dat4+'');var dat1,dat2;
    console.log('cb1:cb2'+cb1+', '+cb2);
    if(!cb1){var n=dat3.length;

```

```

        dat1=gave(dat3);
    }else{
        dat1=dat3*1.0;
    }
    if(!cb2){var n=dat4.length;
        dat2=gave(dat4);
    }else{
        dat2=dat4*1.0;
    }
    //console.log('输入数据格式检测:'+dat1+'、'+dat2);
    switch(ll){
        case 0:/*'
            if(cb1&&cb2){
                rt=dat1*1.0*dat2;
                break;
            }rt=[];
            if(!cb1&&!cb2){
                for(var i=0;i<dat1.length;i++){
                    var fs=[];
                    for(var j=0;j<dat2[0].length;j++){
                        var s=0;
                        for(var k=0;k<dat2.length;k++){
                            s+=dat1[i][k]*dat2[k][j]*1.0;
                        }
                        fs.push(s);
                    }
                    rt.push(fs);
                }
            }
            }else if(!cb1&&cb2){
                for(var i in dat1){
                    var datf4=[];
                    for(var j in dat1[i]){
                        datf4.push(dat1[i][j]*dat2*1.0);
                    }
                    rt.push(datf4);
                }
            }
        }else{
            for(var i in dat2){
                var datf4=[];
                for(var j in dat2[i]){
                    datf4.push(dat2[i][j]*dat1*1.0);
                }
                rt.push(datf4);
            }
        }
    }
}

```

```
}
break;
case 1:/'/'
    if(cb1&&cb2){
        rt=dat1*1.0/dat2;
        break;
    }rt=[];
    if(!cb1&&!cb2){
        for(var i=0;i<Math.min(dat1.length,dat2.length);i++){
            var datf4=[];
            for(var j=0;j<Math.min(dat1[0].length,dat2[0].length);j++){
                datf4.push(dat1[i][j]*1.0/dat2[i][j]*1.0);
            }
            rt.push(datf4);
        }
    }else if(!cb1&&cb2){
        for(var i=0;i<dat1.length;i++){
            var datf4=[];
            for(var j=0;j<dat1[0].length;j++){
                datf4.push(dat1[i][j]*1.0/dat2*1.0);
            }
            rt.push(datf4);
        }
    }else{
        for(var i=0;i<dat2.length;i++){
            var datf4=[];
            for(var j=0;j<dat2[0].length;j++){
                datf4.push(dat1*1.0/dat2[i][j]*1.0);
            }
            rt.push(datf4);
        }
    }
break;
case 2:/'%'
    if(cb1&&cb2){
        rt=dat1*1.0%dat2;
        break;
    }rt=[];
    if(!cb1&&!cb2){
        for(var i=0;i<Math.min(dat1.length,dat2.length);i++){
            var datf4=[];
            for(var j=0;j<Math.min(dat1[0].length,dat2[0].length);j++){
                datf4.push(dat1[i][j]*1.0%dat2[i][j]*1.0);
            }
        }
    }
```

```

        rt.push(datf4);
    }
} else if(!cb1&&cb2){
    for(var i=0;i<dat1.length;i++){
        var datf4=[];
        for(var j=0;j<dat1[0].length;j++){
            datf4.push(dat1[i][j]*1.0%dat2*1.0);
        }
        rt.push(datf4);
    }
} else{
    for(var i=0;i<dat2.length;i++){
        var datf4=[];
        for(var j=0;j<dat2[0].length;j++){
            datf4.push(dat1*1.0%dat2[i][j]*1.0);
        }
        rt.push(datf4);
    }
}
break;
case 3:/'^'
    if(cb1&&cb2){
        rt=Math.pow(dat1*1.0,dat2*1.0);
        break;
    }rt=[];
    if(!cb1&&!cb2){
        for(var i=0;i<Math.min(dat1.length,dat2.length);i++){
            var datf4=[];
            for(var j=0;j<Math.min(dat1[0].length,dat2[0].length);j++){
                datf4.push(Math.pow(dat1[i][j]*1.0,dat2[i][j]*1.0));
            }
            rt.push(datf4);
        }
    } else if(!cb1&&cb2){
        for(var i=0;i<dat1.length;i++){
            var datf4=[];
            for(var j=0;j<dat1[0].length;j++){
                datf4.push(Math.pow(dat1[i][j]*1.0,dat2*1.0));
            }
            rt.push(datf4);
        }
    }
} else{
    for(var i=0;i<dat2.length;i++){
        var datf4=[];

```



```

        for(var j=0;j<dat2[0].length;j++){
            datf4.push(Math.pow(dat1*1.0,dat2[i][j]*1.0));
        }
        rt.push(datf4);
    }
}
break;
case 4:/'+'
    if(cb1&&cb2){
        rt=dat1*1.0+dat2*1.0;
        break;
    }rt=[];
    if(!cb1&&!cb2){
        for(var i=0;i<Math.min(dat1.length,dat2.length);i++){
            var datf4=[];
            for(var j=0;j<Math.min(dat1[0].length,dat2[0].length);j++){
                datf4.push(dat1[i][j]*1.0+dat2[i][j]*1.0);
            }
            rt.push(datf4);
        }
    }else if(!cb1&&cb2){
        for(var i=0;i<dat1.length;i++){
            var datf4=[];
            for(var j=0;j<dat1[0].length;j++){
                datf4.push(dat1[i][j]*1.0+dat2*1.0);
            }
            rt.push(datf4);
        }
    }else{
        for(var i=0;i<dat2.length;i++){
            var datf4=[];
            for(var j=0;j<dat2[0].length;j++){
                datf4.push(dat1*1.0+dat2[i][j]*1.0);
            }
            rt.push(datf4);
        }
    }
}
break;
case 5:/'-'
    if(cb1&&cb2){
        rt=dat1*1.0-dat2*1.0;
        break;
    }rt=[];
    if(!cb1&&!cb2){

```

```

        for(var i=0;i<Math.min(dat1.length,dat2.length);i++){
            var datf4=[];
            for(var j=0;j<Math.min(dat1[0].length,dat2[0].length);j++){
                datf4.push(dat1[i][j]*1.0-dat2[i][j]*1.0);
            }
            rt.push(datf4);
        }
    }else if(!cb1&&cb2){
        for(var i=0;i<dat1.length;i++){
            var datf4=[];
            for(var j=0;j<dat1[0].length;j++){
                datf4.push(dat1[i][j]*1.0-dat2*1.0);
            }
            rt.push(datf4);
        }
    }else{
        for(var i=0;i<dat2.length;i++){
            var datf4=[];
            for(var j=0;j<dat2[0].length;j++){
                datf4.push(dat1*1.0-dat2[i][j]*1.0);
            }
            rt.push(datf4);
        }
    }
    break;
case 6:/'sin'
    if(cb2){
        rt=Math.sin(dat2*1.0);
        break;
    }else{
        rt=[];
        for(var i in dat2){
            rt.push([]);
            for(var j in dat2[i]){
                rt[i].push(Math.sin(dat2[i][j]*1.0));
            }
        }
    }
    break;
case 7:/'cos'
    if(cb2){
        rt=Math.cos(dat2*1.0);
        break;
    }else{

```

```
        rt=[];
        for(var i in dat2){
            rt.push([]);
            for(var j in dat2[i]){
                rt[i].push(Math.cos(dat2[i][j]*1.0));
            }
        }
    }
    break;
case 8:/'tan'
    if(cb2){
        rt=Math.tan(dat2*1.0);
        break;
    }else{
        rt=[];
        for(var i in dat2){
            rt.push([]);
            for(var j in dat2[i]){
                rt[i].push(Math.tan(dat2[i][j]*1.0));
            }
        }
    }
    break;
case 9:/'asin'
    if(cb2){
        rt=Math.asin(dat2*1.0);
        break;
    }else{
        rt=[];
        for(var i in dat2){
            rt.push([]);
            for(var j in dat2[i]){
                rt[i].push(Math.asin(dat2[i][j]*1.0));
            }
        }
    }
    break;
case 10:/'acos'
    if(cb2){
        rt=Math.acos(dat2*1.0);
        break;
    }else{
        rt=[];
        for(var i in dat2){
```

```

        rt.push([]);
        for(var j in dat2[i]){
            rt[i].push(Math.acos(dat2[i][j]*1.0));
        }
    }
}
break;
case 11:/'atan'
    if(cb2){
        rt=Math.atan(dat2*1.0);
        break;
    }else{
        rt=[];
        for(var i in dat2){
            rt.push([]);
            for(var j in dat2[i]){
                rt[i].push(Math.atan(dat2[i][j]*1.0));
            }
        }
    }
    break;
case 12:/'exp'
    if(cb2){
        rt=Math.exp(dat2*1.0);
        break;
    }else{
        rt=[];
        for(var i in dat2){
            rt.push([]);
            for(var j in dat2[i]){
                rt[i].push(Math.exp(dat2[i][j]*1.0));
            }
        }
    }
    break;
case 13:/'lg'
    if(cb2){
        rt=Math.log(dat2*1.0)/Math.log(10.0);
        break;
    }else{
        rt=[];
        for(var i in dat2){
            rt.push([]);
            for(var j in dat2[i]){

```

```

        rt[i].push(Math.log(dat2[i][j]*1.0)/Math.log(10.0));
    }
}
break;
case 14:/'\n'
    if(cb2){
        rt=Math.log(dat2*1.0);
        break;
    }else{
        rt=[];
        for(var i in dat2){
            rt.push([]);
            for(var j in dat2[i]){
                rt[i].push(Math.log(dat2[i][j]*1.0));
            }
        }
    }
    break;
case 15:/'\RA'
    if(cb2){
        if(dat2*1==0){
            rt=0+"";
        }else{
            rt=1+"";
        }
        break;
    }else{
        var m=dat2.length,n=dat2[0].length;
        for(var i=0;i<n;i++){
            for(var j=i;j<m;j++){
                var k=i;
                for(k=i;k<n;k++){
                    if(Math.abs(dat2[j][k])>0.000001){
                        break;
                    }
                }
                for(var u=k+1;u<n;u++){
                    dat2[j][u]=dat2[j][u]*1.0/dat2[j][k];
                }
                if(k!=n){
                    dat2[j][k]=1;
                }
            }
        }
    }
}

```

```

        var k=i;
        for(k=i;k<m;k++){
            if(dat2[k][i]*1!=0){
                break;
            }
        }
        if(k==m){break;}
        var t=dat2[k];dat2[k]=dat2[i];dat2[i]=t;
        for(var j=i+1;j<m;j++){
            if(dat2[j][i]*1!=0){
                for(var u=i;u<n;u++){
                    dat2[j][u]-=dat2[i][u];
                }
            }
        }
    }
    var rt=0;
    for(var i=0;i<dat2.length;i++){
        if(dat2[i].indexOf(1)!=-1){
            rt++;
        }
    }
    rt=rt;
}
break;
case 16:/'TA'
    if(cb2){
        rt=dat2*1.0;
        break;
    }else{
        rt=[];
        for(var i=0;i<dat2[0].length;i++){
            var datf3=[];
            for(var j=0;j<dat2.length;j++){
                datf3.push(dat2[j][i]);
            }
            rt.push(datf3);
        }
    }
}
break;
case 17:/'TN'
    if(cb2){
        rt=dat2*1.0;
        break;
    }

```

```

    }else{
        rt=func('TA',1,dat2);
        rt.reverse();
        rt=func('TA',1,rt);
    }
    break;
case 18:/'UP'
    if(cb2){
        rt=dat2*1.0;
        break;
    }else{
        rt=gave(dat2);
        rt.reverse();
    }
    break;
case 19:/'rref'
    if(cb2){
        if(dat2*1==0){
            rt=0;
        }else{
            rt=1;
        }
        break;
    }else{
        rt=gave(dat2);
        var m=rt.length,n=rt[0].length;
        for(var i=0;i<n;i++){
            for(var j=i;j<m;j++){
                var k=i;
                for(k=i;k<n;k++){
                    if(Math.abs(rt[j][k])>0.000001){
                        break;
                    }
                }
                for(var u=k+1;u<n;u++){
                    rt[j][u]=rt[j][u]*1.0/rt[j][k];
                }
                if(k!=n){
                    rt[j][k]=1;
                }
            }
        }
        var k=i;
        for(k=i;k<m;k++){
            if(rt[k][i]*1!=0){

```

```

        break;
    }
}
if(k==m){break;}
var t=rt[k];rt[k]=rt[i];rt[i]=t;
for(var j=i+1;j<m;j++){
    if(rt[j][i]*1!=0){
        for(var u=i;u<n;u++){
            rt[j][u]-=rt[i][u];
        }
    }
}
}
for(var i=1;i<m;i++){
    var t=rt[i].indexOf(1);
    if(t==-1){break;}
    for(var l=0;l<i;l++){
        for(var j=t+1;j<n;j++){
            rt[l][j]=rt[l][j]-rt[l][t]*rt[i][j];
        }
        rt[l][t]=0;
    }
}
}
break;
case 20:/'inv'
    if(cb2){
        rt=1.0/dat2;
        break;
    }else{
        rt=gave(dat2);
        var n=Math.min(rt.length,rt[0].length);
        if(func('det','1',rt)==0){
            return '出现降秩矩阵';
        }
        for(var i=0;i<n;i++){
            for(var j=0;j<n;j++){
                if(i==j){
                    rt[i].push(1);
                }else{
                    rt[i].push(0);
                }
            }
        }
    }
}

```



```

        rt=func('rref','1',rt);
        rt=func('TA','0',rt);
        var datf3=[];
        for(var u=0;u<n;u++){
            datf3.push(rt.pop());
        }
        datf3=func('TA','0',datf3);
        rt=func('TN','0',datf3);
    }
    break;
case 21:/'det'
    if(cb2){
        rt=dat2*1.0;
        break;
    }else{
        var asd=gave(dat2);
        n=Math.min(asd.length,asd[0].length);
        s=1.0;
        for(var k=0;k<n;k++){
            var max=Math.abs(asd[k][k]);
            var m=k;var ff=true;
            for(var i=k;i<n;i++){
                if(asd[i][k]!=0){
                    ff=false;
                    break;
                }
            }
            if(ff){rt=0;break;}
            for(var l=k+1;l<n;l++){
                if(max<Math.abs(asd[l][k])){
                    m=l;
                }
            }
            if(k!=m){
                var t=asd[k];
                asd[k]=asd[m];
                asd[m]=t;
                s=-s;
            }
            var tp=asd[k][k];
            for(var j=k+1;j<n;j++){
                asd[k][j]=asd[k][j]*1.0/tp;
            }
            for(var i=k+1;i<n;i++){

```

```
        for(var j=k+1;j<n;j++){
            asd[i][j]=asd[i][j]*1.0-asd[i][k]*asd[k][j];
        }
    }
}
for(var i=0;i<n;i++){
    s=s*asd[i][i];
}
rt=s;
}
break;
case 22:/'v'
    if(cb2){
        rt=Math.sqrt(Math.abs(dat2*1.0));
        break;
    }else{
        rt=[];
        for(var i in dat2){
            rt.push([]);
            for(var j in dat2[i]){
                rt[i].push(Math.sqrt(Math.abs(dat2[i][j]*1.0)));
            }
        }
    }
    break;
case 23:/'rand'
    if(cb2){
        rt=dat2*1.0*Math.random();
        break;
    }else{
        rt=[];
        for(var i in dat2){
            rt.push([]);
            for(var j in dat2[i]){
                rt[i].push(dat2[i][j]*1.0*Math.random());
            }
        }
    }
    break;
case 24:/'abs'
    if(cb2){
        rt=Math.abs(dat2*1.0);
        break;
    }else{
```

```
        rt=[];
        for(var i in dat2){
            rt.push([]);
            for(var j in dat2[i]){
                rt[i].push(Math.abs(dat2[i][j]*1.0));
            }
        }
    }
    break;
case 25:/'ceil'
    if(cb2){
        rt=Math.ceil(dat2*1.0);
        break;
    }else{
        rt=[];
        for(var i in dat2){
            rt.push([]);
            for(var j in dat2[i]){
                rt[i].push(Math.ceil(dat2[i][j]*1.0));
            }
        }
    }
    break;
case 26:/'floor'
    if(cb2){
        rt=Math.floor(dat2*1.0);
        break;
    }else{
        rt=[];
        for(var i in dat2){
            rt.push([]);
            for(var j in dat2[i]){
                rt[i].push(Math.floor(dat2[i][j]*1.0));
            }
        }
    }
    break;
case 27:/'round'
    if(cb2){
        rt=Math.round(dat2*1.0);
        break;
    }else{
        rt=[];
        for(var i in dat2){
```

```

        rt.push([]);
        for(var j in dat2[i]){
            rt[i].push(Math.round(dat2[i][j]*1.0));
        }
    }
}
break;
case 28:/'#'  
    if(cb1&&cb2){
        rt=dat1*1.0*dat2;
        break;
    }rt=[];
    if(!cb1&&!cb2){
        for(var i=0;i<Math.min(dat1.length,dat2.length);i++){
            var datf4=[];
            for(var j=0;j<Math.min(dat1[0].length,dat2[0].length);j++){
                datf4.push(dat1[i][j]*1.0*dat2[i][j]);
            }
            rt.push(datf4);
        }
    }else if(!cb1&&cb2){
        for(var i=0;i<dat1.length;i++){
            var datf4=[];
            for(var j=0;j<dat1[0].length;j++){
                datf4.push(dat1[i][j]*1.0*dat2);
            }
            rt.push(datf4);
        }
    }else{
        for(var i=0;i<dat2.length;i++){
            var datf4=[];
            for(var j=0;j<dat2[0].length;j++){
                datf4.push(dat1*1.0*dat2[i][j]);
            }
            rt.push(datf4);
        }
    }
}
break;
}
//console.log('计算处理前:'+rt);
if(!isNaN(rt+'')){
    rt=Math.round(rt*baoliu)/baoliu+'';
}else{
    for(var i in rt){

```

---

```

        for(var j in rt[i]){
            rt[i][j]=Math.round(rt[i][j]*baoliu)/baoliu+'';
        }
    }
    rt=rt;
}
console.log('计算结果查看:'+rt);
return gave(rt);
}
function gave(dat2){
    let tsrt=[],tst=[];
    if(isNaN(dat2)){
        for(var i in dat2){
            tsrt.push(dat2[i]+'')
        }
        for(var i in tsrt){
            tst.push(tsrt[i].split(','));
        }
        return tst;
    }else{
        var oo=dat2+''
        tsrt.push(oo);
        return tsrt;
    }
}
}
module.exports = {
    toRPolish:toRPolish,
    trans:trans,
    cacular:cacular,
    func:func
}

```

## 2.指令分析引擎(translate.js)

```

var name=[];
var data=[];
var logs=[];
var log=[];
function inputs(nem,dat,lo){
    name=nem;
    data=dat;
    logs=lo;
}
function thans(cmd){

```

```

cmd = cmd.replace(/ (\/g, '(');cmd = cmd.replace(/) /g, ')');cmd = cmd.replace(/\ +/g, '');
cmd = cmd.replace(/[\r\n]/g, '');cmd = cmd.replace(/\ /g, '');cmd = cmd.replace(/, /g, ',');
cmd = cmd.replace(/? /g, '?');cmd = cmd.replace(/; /g, ';');cmd = cmd.replace(/: /g, ':');
cmd = cmd.replace(/【 /g, '[');cmd = cmd.replace(/】 /g, ']');
log=[];
return cmd;
}
function check(cmd){
  console.log(cmd);
  console.log(cmd=="");
  if(/clear/.test(cmd)){
    return 0;
  }else if(/cls/.test(cmd)){
    return 1;
  }else if(/plot/.test(cmd)){
    return 2;
  }else if(/\/?.test(cmd)){
    return 3;
  }else if(/setting/.test(cmd)){
    return 4;
  }else if(/diag/.test(cmd)){
    return 5;
  }else if(/ones/.test(cmd)){
    return 6;
  }else if(/zeros/.test(cmd)){
    return 7;
  }else if(/eye/.test(cmd)){
    return 8;
  }else if(/[/].test(cmd) || /\]/.test(cmd)){
    return 9;
  }else if(/:/.test(cmd)){
    return 10;
  }else{
    console.log('this is run');

    if(/(\*|\/|(|\)|\)|\||\%|\^|\+|\-|\#|sin|cos|tan|asin|acos|atan|exp|ln|lg|RA|TA|TN|UP|rref|
inv|det|v|rand|abs|ceil|floor|round)/.test(cmd)){return 11;}
    if(/=/ .test(cmd)){
      var g=cmd.substring(cmd.indexOf('=')+1,cmd.length);
      if(!isNaN(g)){return 12;}
      if(name.indexOf(g)==-1){
        log.push('不存在数据'+g);
        return -1;
      }else{

```

```

        return 12;
    }
}
if(name.indexOf(cmd)==-1){
    log.push('不存在该表达式:'+cmd);
    return -1;
}else{
    return 12;
}
}
log.push('不存在该表达式:'+cmd);
return -1;
}
function runn(key,tcmd,baoliu,vul){
    var cmd=tcmd;var tname='';
    if(/\=/\.test(tcmd)){
        tname=tcmd.substring(0,tcmd.indexOf('='));
        cmd=tcmd.substring(tcmd.indexOf('=')*1.0+1,tcmd.length)
    }
    if(tname=='' || !isNaN(tname)){tname='ans';}
    switch(key){
        case -1:
            logs.push(log);
            return logs;
        case 0:
            console.log('delete');
            if(/\(/\.test(cmd)&&\/\)\.test(cmd)){
                var m=cmd.substring(cmd.indexOf('(')+1,cmd.indexOf(')'));
                var arr=m.split(',');
                for(var i in arr){
                    var j=name.indexOf(arr[i]);
                    if(j!=-1){
                        name.splice(j,1);
                        data.splice(j,1);
                    }
                }
                if(vul){logs.push(arr+'清除成功');}
                return [name,data,logs];
            }else{
                name=[];
                data=[];
                if(vul){logs.push('所有数据清除成功');}
                return [name,data,logs];
            }
    }
}

```

```
case 1:
    logs=[];
    return [];
case 2:
    if(/\(/.test(cmd)&&/\)/.test(cmd)){
        var m=cmd.substring(cmd.indexOf('(')+1,cmd.indexOf(')'));
        var arr=m.split(',');
        return arr;
    }else{logs.push('缺少括号,表达式有误');return '缺少括号,表达式有误';}
    return [];
case 3:
    return [];
case 4:
    var dt=/[0-9]/.exec(cmd);
    if(dt!=0){
        if(vul){logs.push('保留'+dt+'位小数');}
        return Math.pow(10,dt*1.0);
    }else{
        logs.push('数据有误,修改失败');
        return baoliu;
    }
}
case 5:
    var mg=cmd.substring(cmd.indexOf("(")+1,cmd.indexOf(")"));
    var arr=mg.split(',');
    var arr2=[];
    var m=arr.length;
    for(var i in arr){
        var arr1=[];
        var t=0;
        for(var j=0;j<m;j++){
            if(i==j){
                t=arr[i];
            }else{
                t=0;
            }
            arr1.push(t);
        }
        arr2.push(arr1);
    }
    var gh=makedata(arr2);
    break;
case 6:
    var jk=cmd.substring(cmd.indexOf("(")+1,cmd.indexOf(")"));
    var arr=jk.split(',');
```



```
var arr2=[];
var m=0,n=0;
if(arr.length==1){
    m=arr[0];n=arr[0];
}else{
    m=arr[0];n=arr[1];
}
for(var i=0;i<m;i++){
    var arr1=[];
    for(var j=0;j<n;j++){
        arr1.push(1);
    }
    arr2.push(arr1);
}
var gh=makedata(arr2);
break;
case 7:
    var jk=cmd.substring(cmd.indexOf("(")+1,cmd.indexOf(")"));
    var arr=jk.split(',');
    var arr2=[];
    var m=0,n=0;
    if(arr.length==1){
        m=arr[0];n=arr[0];
    }else{
        m=arr[0];n=arr[1];
    }
    for(var i=0;i<m;i++){
        var arr1=[];
        for(var j=0;j<n;j++){
            arr1.push(0);
        }
        arr2.push(arr1);
    }
    var gh=makedata(arr2);
    break;
case 8:
    var jk=cmd.substring(cmd.indexOf("(")+1,cmd.indexOf(")"));
    var arr=jk.split(',');
    var arr2=[];
    var m=0,n=0;
    if(arr.length==1){
        m=arr[0];n=arr[0];
    }else{
        m=arr[0];n=arr[1];
    }
```

```

    }
    for(var i=0;i<m;i++){
        var arr1=[];var t=0;
        for(var j=0;j<n;j++){
            if(i==j){
                t=1;
            }else{
                t=0;
            }
            arr1.push(t);
        }
        arr2.push(arr1);
    }
    var gh=makedata(arr2);
    break;
case 9:
    if(/\[/\].test(cmd)&&/\]/\].test(cmd)){
        var m=cmd.substring(cmd.indexOf("[")+1,cmd.indexOf("]"));
        console.log('此时的 m'+m);console.log('m 的类型'+!isNaN(m));
        if(!isNaN(m)){var gh=m+";break;}
        var arr2=[];
        var arr=m.split(';');
        var l=1;
        for(var i in arr){
            var gt=arr[i].split(',');
            if(gt.length>l){
                l=gt.length;
            }
        }
        for(var i in arr){
            var arr1=[];
            var f=arr[i].split(',');
            for(var j=0;j<l;j++){
                if(j<f.length){
                    arr1.push(f[j]);
                }else{
                    arr1.push(0);
                }
            }
            arr2.push(arr1);
        }
        var gh=makedata(arr2);
        break;
    }else{

```

```

logs.push('数据异常,可能是中括号未匹配')
return '数据异常';
}
case 10:
var m=cmd.substring(cmd.indexOf('=')+1,cmd.length);
var arr=m.split(':');var arr2=[];
var d=1.0;var a0=0.0;var a1=0.0;
console.log(a0);
console.log(d);
console.log(a1);
if(arr.length==2){
a0=arr[0]*1.0;
a1=arr[1]*1.0;
}else{
a0=arr[0]*1.0;
d=arr[1]*1.0;
a1=arr[2]*1.0;
}
console.log(a0);
console.log(d);
console.log(a1);
for(var i=a0*1.0;i<=a1;i=i+d*1.0){
arr2.push(Math.round(i*baoliu)/baoliu);
}
var gh=makedata(arr2);
break;
case 11:
var dan=[],da=[];
dan=name.join().split(',');
for(var i in data){
da.push(makedata1(data[i]));
}
var
tgf=/(\*|\/|(|\)|\||\%|\^|\+|\-|\#|sin|cos|tan|asin|acos|atan|exp|ln|lg|RA|TA|TN|UP|rref|inv
|det|v|rand|abs|ceil|floor|round)/.test(tname);
if(tgf){logs.push('>>>'+cmd+':');}else{logs.push('>>>'+tname+'<'+cmd+':');}
return [da,dan,cmd,tname,!tgf];
case 12:
logs.push('>>>'+tname+':');
console.log('name'+tname);
var gh;
if(!isNaN(cmd)){gh=cmd;}else{gh=data[name.indexOf(cmd)];}
console.log('gh:'+gh);
if(name.indexOf(tname)==-1){

```

---

```
        name.push(tname);
        data.push(gh);
    }else{
        data[name.indexOf(tname)]=gh;
    }
    addlogs(gh);
    return [gh,tname,'数据添加成功'];
}
logs.push('>>>' + tname + ':');
if(tname!='ans'){
    if(name.indexOf(tname)==-1){
        name.push(tname);
        data.push(gh);
    }else{
        data.splice(name.indexOf(tname),1,gh);
    }
}
}
else{
    vul=true;
}
if(vul){addlogs(gh);}
return [gh,tname,'数据添加成功'];
}
function makedata(dat){
    if(isNaN(dat)){
        var rt=dat[0]+'';
        for(var i=1;i<dat.length;i++){
            rt=rt+';'+dat[i];
        }
        return rt;
    }else{
        return dat+'';
    }
}
function makedata1(dat){
    if(isNaN(dat)){
        var arr2=[];
        var arr=dat.split(';');
        for(var i in arr){
            var arr1=[];
            var f=arr[i].split(',');
            for(var j in f){
                arr1.push(f[j]);
            }
            arr2.push(arr1);
        }
    }
```

```
    }
    return arr2;
  }else{
    return dat*1.0;
  }
}
function addlogs(dat){
  var arr=dat.split(';');
  for(var i in arr){
    logs.push(arr[i]);
  }
}
function getnums(){
  return [logs,name,data];
}
module.exports = {
  thans:thans,
  inputs:inputs,
  check:check,
  runn:runn,
  makedata:makedata,
  makedata1:makedata1,
  addlogs:addlogs,
  getnums:getnums
}
```