

# rust课程实践报告

姓名：黄骏齐

学号：2100012956

2023年7月23日

## 课程实践报告

因为在物理课程上很容易遇到复数的波函数以及积分计算，计划做一个用rust编写的科学计算器，功能包含复数域上的基本运算以及实数域上初等函数的定积分计算。

可以选择使用计算模式还是积分模式。

### 复数表达式求值

实现复数域上的四则运算、指数、对数、三角函数的计算

输入：一个合法的表达式，合法的表达式expr包括：

- 实数a，纯虚数bi
- 常数e,pi
- (expr)，小括号
- +expr,-expr，单元运算
- expr1 op expr2，二元运算，其中 $op \in \{+, -, *, /\}$
- $\text{expr1}^{\text{expr2}}$ ，要求expr1的值是正实数
- $\ln(\text{expr})$ ，要求expr的值是正实数
- $\sin(\text{expr}), \cos(\text{expr}), \tan(\text{expr})$

运算的优先级是单元运算 $> \wedge > * / > +-$ ，运算法则满足0.1

注：表达式可以在不影响意义的情况下任意加空格。还有判断除零、 $\tan(\frac{\pi}{2})$ 等，对数只实现ln，其余需要手动进行换底公式的变换

输出：会检测输入的表达式是否合法，若合法，则输出表达式的值(数值表示)，否则输出相应的错误信息。

## 定积分计算

输入:  $a \ b \ f(x)$

要求满足 $a, b \in \mathbf{R}$ , 且 $f(x)$ 中只有变元 $x$ 且需要在实数域下, 其余的形式满足上面的规则。

输出: 会检测输入是否合法, 若合法, 则输出 $\int_a^b f(x)dx$ , 否则输出相应的错误信息。但不判断被积函数是否一定有意义且可积, 如果给了一个合法的输入但是错误的函数, 可能会得到错误的结果。

## 实现方法

首先使用了peg Parser<sup>[1]</sup>生成器对输入数据进行了一些处理, 把一个表达式写成一个enum的形式, enum如下所示:

```
1 pub enum Node {  
2     Ident(Complex),  
3     Dop(char, Box<Node>, Box<Node>),  
4     Sop(char, Box<Node>)  
5 }
```

其中第一行表示单个复数, 第二行表示一个二元运算, 第三行表示单元运算

然后通过对impl在Node上的函数calc进行递归计算值。

在每一层递归中用pattern matching来计算, calc函数如下

```
1 pub fn calc( self ) -> Complex{  
2     match self{  
3         Self :: Ident(v) => v,  
4         Self :: Dop(c, x,y) => Complex::bopcalc(c,x.calc(),y.calc()),  
5         Self :: Sop(c,x) => Complex::sopcalc(c,x.calc())  
6     }  
7 }
```

其中Complex是一个定义复数的struct, 而bopcalc和sopcalc分别是作用在Complex上的二元和单元函数。

对于其中的除法、ln、exp运算, 则通过使用result类型来判断与记录错误信息

```
please choose a mode.0 for calculator and 1 for integral.  
0  
2 + (2+i)*(e^i)  
r:2.239133626928383,i:2.2232442754839328
```

图 1: 复数计算器效果

对于定积分计算, 使用辛普森积分法<sup>0.2</sup>, 本质上转化成函数的多点求值问题, 多次调用表达式求值, 来获得定积分的近似解。

项目地址: <https://github.com/huangjunqi1/rustcalculator/>

```

please choose a mode.0 for calculator and 1 for integral.
1
1 3
1/x
1.0987011790384764

```

图 2: 积分计算器效果,  $\ln 3 = 1.09861229$ , 对数值积分还是有些误差

## todo list

- 一个ui界面
- 更好地记录下错误信息
- 支持更多函数
- 支持读入的科学计数法
- 保存之前的结果

## 附录

### 0.1 复数域上的运算法则

- $i * i = -1$
- $(a + bi) * (c + di) = (ac - bd) + (ad + bc)i$
- $e^{\theta i} = \cos \theta + i \sin \theta$ <sup>[2]</sup>
- $\cos x = \frac{e^{ix} + e^{-ix}}{2}$ ,  $\sin x = \frac{e^{ix} - e^{-ix}}{2}$ , 即复数的三角函数也是有意义的
- $\log_a b = \frac{\ln b}{\ln a}$ , 换底公式

### 0.2 辛普森积分法

在被积区间中采样 $n$ 个点来得到积分的估计<sup>[3]</sup>, 下式中 $n$ 是偶数

$\int_a^b f(x)dx \approx \frac{b-a}{3n} [f(a) + f(b) + 4 \sum_{i=1}^{n/2} f(x_{2i-1}) + 2 \sum_{i=1}^{n/2-1} f(x_{2i})]$ , 其中 $\{x_i\}$ 是 $[a, b]$ 的 $n$ 分点  
误差为 $Err \leq \frac{M(b-a)^5}{180n^4}$ , 其中 $M$ 是区间上四阶导的最大值

实际使用20000个分点来求积分的近似值, 根据实际运行情况进行调整, 在分点上表达式无意义的话则输出错误信息, 但不检测其他点上函数是否有值。

## References

- [1] [rust-peg](#)
- [2] [Euler's formula](#)
- [3] [Simpson's rule](#)
- [4] [fltk-rs](#)