

Mowrks 使用体验

Syslab 的使用体验类似于 VSCode IDE 环境 + Julia 程序语言的结合，同时它又支持 Python 和 Matlab 脚本。客观来讲，同元公司把每个部分融合得很好。作为一款新兴的数学软件平台，Mworks 展现出了独特的优势。

使用过程中体验良好的地方(优点)有:

- **支持显式的向量/矩阵运算**

相比 python 语言，Syslab 无需借助于其他工具即可实现高效运算，且由于 julia 语言本身的特性，通过循环或者类型派发并不影响运行的速度。

- **界面优美**

- **软件体积很小**

相比于国际上知名的三大数学软件 Matlab，Mathematica，Maple，如果是数学专业出身背景的老师或者同学可以通过同元官网下载安装，无需等待过久或繁琐的配置即可使用。

使用过程中体验较差的地方(缺点)有:

- **报错机制依旧使用的 Julia 原生机制**

针对这点，可以通过像开发 VSCode 插件或者包功能机制一样修饰报错，只返回关键的报错信息，省略调用过程中的函数堆栈调用信息。特别的，应该支持中文输出报错信息，这点对于国内用户来说十分关键。

- **未能将 VSCode 支持插件的优势发挥出来**

既然使用了 VSCode Server 一般的界面或服务器，不妨更直接一点，支持插件功能，相信经插件拓展后的 Mworks 会更受大家的欢迎。

- **绘图不美观**

这点可以说是可视化的痛点了，当今基本上所有主流的数学软件，哪怕是编程语言都有优美的图形化功能。本来 Julia 的绘图包也能优化作图的，但不知为何 Syslab 中使用的是 Python 的接口，实际上这点可以调用 Matplotlib 定制更加优美的作图风格。同元官方课程培训中自己也提到过这一点，可能还在开发过程中。但是究竟是选用 Python 实现还是选用 Julia 实现这是个值得深思的点。

- **帮助文档不精确不全面**

使用过程中经常有词不达意的地方，不过我们也相信同元公司的工作人员也在积极完善校对帮助文档。

- **函数或算法功能缺失**

这点和帮助文档一样痛点，有些函数或算法未能实现。或者由于不同库的函数重名，很难判断使用

哪个函数，具体的，我个人在进行信号处理时调用了很多函数尝试了很多次才终于试对，找到正确的函数。

- **拓展性差**

这点其实可以理解，毕竟 Mworks 作为一个商业软件，拥有固定的版本是十分关键的。但是想想 Python? Julia 作为一门新兴的开源程序语言，Julia 社区也提供了相当的可供用户选择的包。但是通过 Pkg 添加功能时常常会引起与内部库冲突，报错，或者预编译的错误。如果不了解 Pkg 使用的用户是不知道创建虚拟环境添加想用的库和功能的。

- **语言转换与运行速度**

如果要实现 Mworks 对标 Matlab 的平替，免不了要做性能方面的优化。对此，要理解懂得 Matlab 脚本和 Julia 脚本的运行机制，不了解两者差异性的用户进行转换会显得十分困难。如果是矩阵运算(绝大部分情况都是这样)，这里通常要了解 Matlab 矩阵会自动进行类型匹配，而 Julia 没有这种机制，且要提升 Julia 的运行速度通常要进行静态编译或者类型指定。据此，我认为这里的难题应该交给平台来实现，帮助用户做个判断是否能进行静态编译，以达到接近 C 或 Matlab 的运行速度。

- **数值溢出**

其他数学软件都提供了对数值溢出的处理，这里 Mworks 处理的并非十分良好。

- **激活使用**

似乎要使用完整的功能必须要登录，激活才能使用。但是这并非重点，重点在于每隔两天都要重新登录激活。

由于并非科班出身，如有不当之处，请官方不吝指正，生定当自勉，努力学习。