

# 如何快速入门开源软件

原创 邓子平 [多物理场仿真技术](#)



工业软件里有很多开源软件，有些是学习的好材料，有些可以当作第三方组件用于开发。

早期因为项目工作需要以及兴趣，把市面上的绝大多数开源软件都摸过一遍，有些一直长期在使用，包括几何内核，图形渲染，各种物理场有限元，多物理场，时域有限差分，边界元，线性方程组求解，显式动力学，格子玻尔兹曼，离散元。因为软件过多，名字就不一一例举了。

面对纷繁的开源软件，如何快速入门，减少试错成本，选择适合的一两款，本文介绍笔者的一些经验。

1. 相当一部分开源软件使用CMAKE，所以要熟悉CMAKE的使用，很多可以直接生成VS工程项目，方便看代码和调试。

2. 从软件工程角度看，非C++项目商业意义不大。相同功能的实现如果同时有Fortran和C++版本，优先考虑C++。很多优秀的算法底层使用Fortran，比如商业版本的NLPQL，被封装当做库使用，一些线性方程组解法同时提供了Fortran和C++接口。

3. 学习使用过程中由简到难，由浅入深。面对一个新的项目，如果一开始折腾时间过长，而没有所成就，挫败感是很强的。开源软件通常配置都比较复杂，坑也较多，很容易出现错误，一开始从最简单的做起，耐心调试研究，一个坑一个坑填平。

4. 有些开源软件只支持Linux，没有Windows版本。比如OpenFOAM，MEEP等，增加了学习研究难度。Windows和VS提供了一些变通的访问Linux方法，可以多研究，另外Docker也能帮上忙。

5. 在开源软件基础上开发新功能，是提高水平的有效途径，有些开源工程放在类似github上，可以自己拉分支，增加新功能，帮忙改进项目。

6. 开源的文档除了少数写的比较详细外，大部分都比较差，优先看帮助文档，再看文档提供的参考文献和书籍，能帮助理解源码。最典型的就是用Java写的显式动力学开源软件Impact，其主要参考书籍《**连续体和结构的非线性有限元**》，通读此书即可写出能运行的代码。

7. 对于一些重量级的开源软件比如VTK, OCC, OpenFOAM, CodeAster等，一定要静下心来阅读文档。通常这些文档比较全面，从理论到实际案例都讲得比较清楚，可以反复看。

8. 一些跨平台的开源软件需要交叉编译，要求比较熟悉Windows/Linux环境和一些编译器原理，没有相关经验慎碰。

9. 有些研究性质的开源软件，没有生态圈，常年没有维护更新，还有些代码阅读起来非常费劲（比如一个函数几十个甚至上百个参数），这类开源软件上不用花太多时间，除非是唯一的老古董，天体仿真软件里碰到过。

暂时写这么多，朋友们在使用开源软件过程中有什么问题，可以留言，后续根据反馈再更新。

阅读: null

在看: null