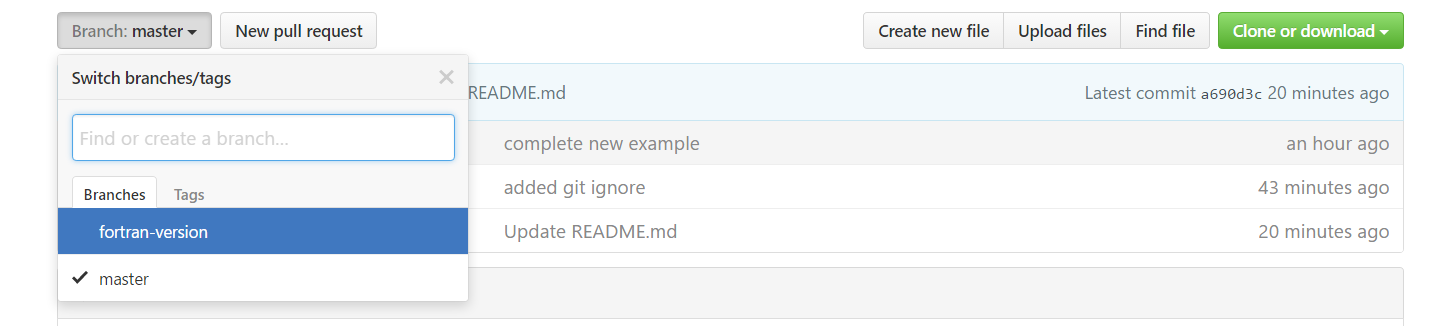
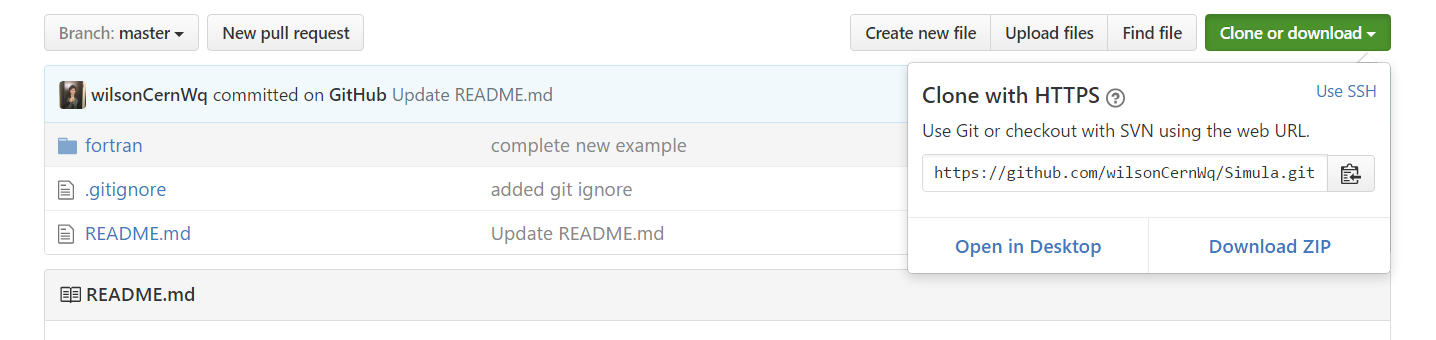
## **How to get the program**

Go to my GitHub <https://github.com/wilsonCernWq/Simula>

Switch to ‘fortran-version’ branch



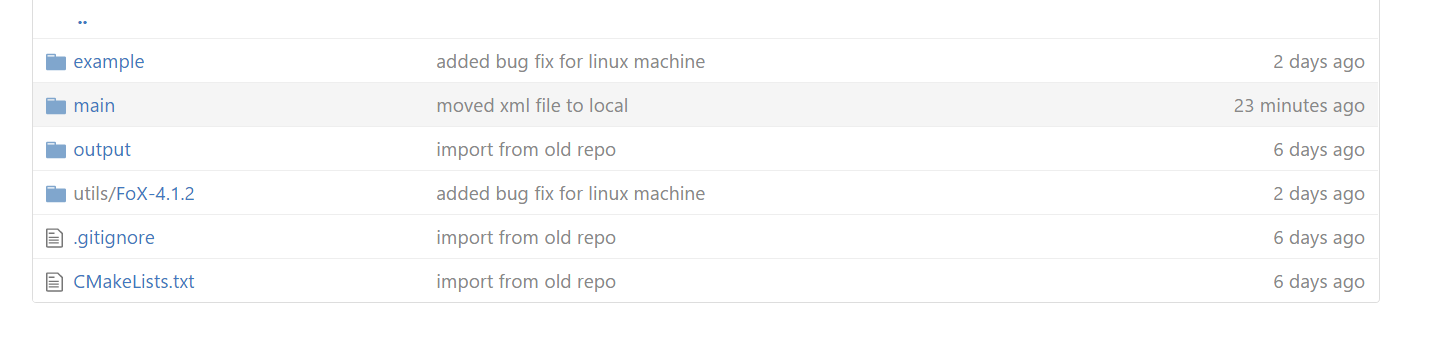
Clone the repo (Download ZIP, or use terminal if you are familiar with git)



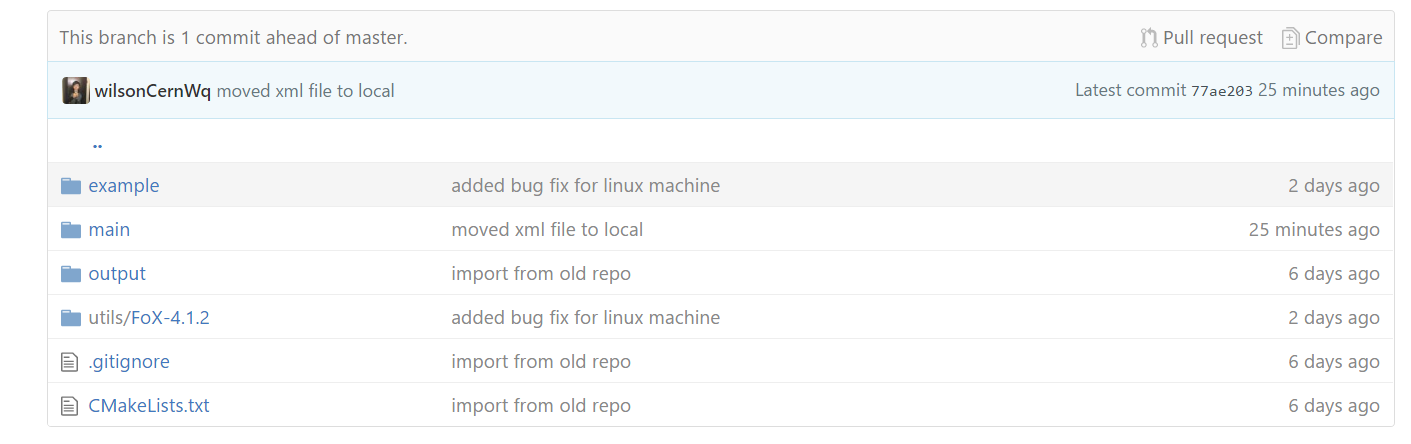
Read the README.md and build the program correctly

## **How to understand the program**

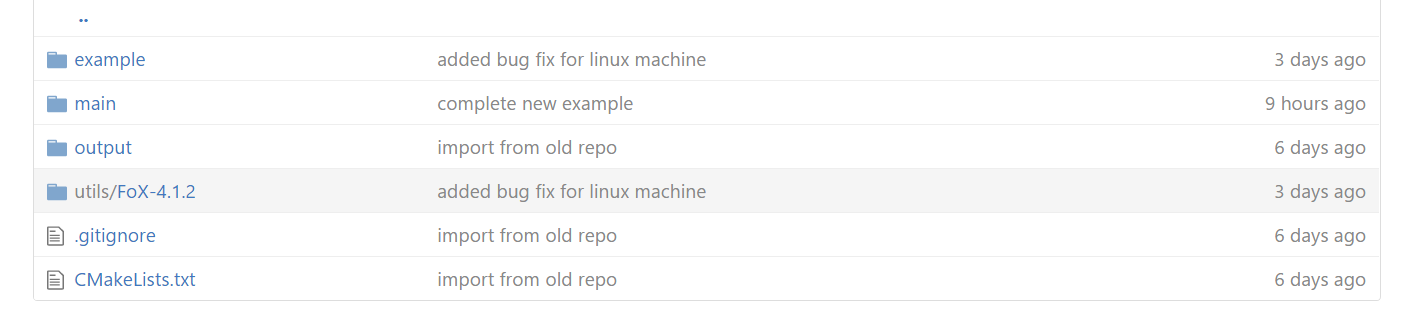
main/: folder to store all program files



example/: folder to store some backup files for me

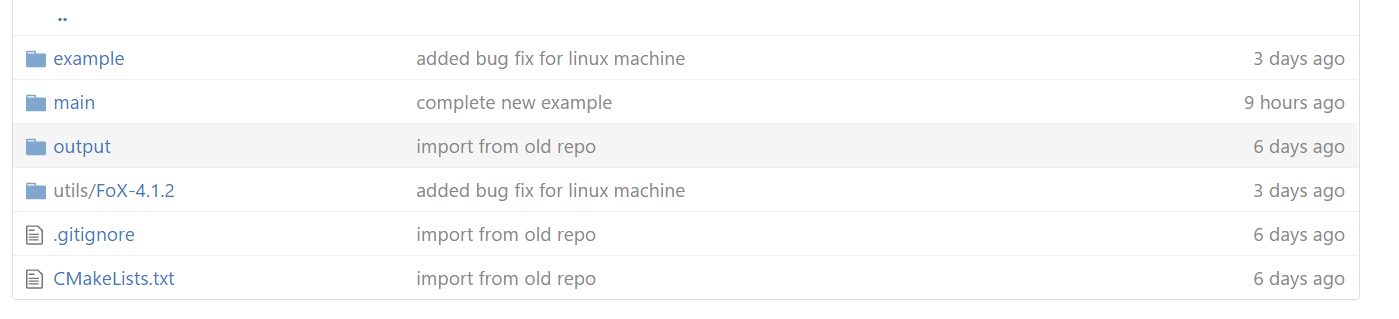


utils/FoX-4.1.2: folder to store the external library, which is useless for now (disabled)



output/: folder to store all output files

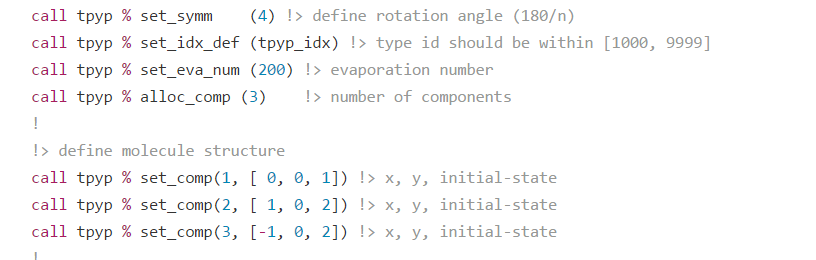
naming convension: <project>-<date>/<index>-<datetime>



## **How the algorithm works (I explain in Chinese)**

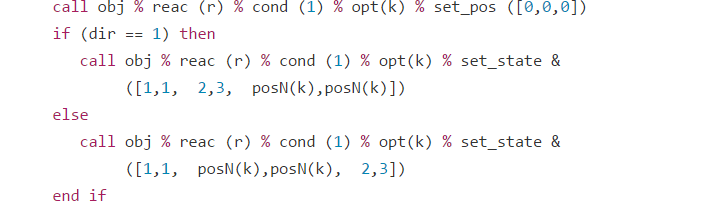
这个程序的逻辑其实比较简单，程序里面你需要定义一些内容：

1. 分子类型 mtype：这一般来说相当于一种类型的单体，比如有机分子，比如金属原子。真空的代号是0，指的是单体和衬底任何相互作用都没有的衬底。

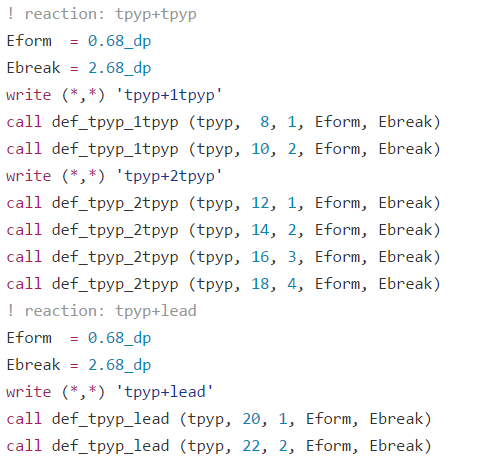


这里列出了mtype需要的基本参数。symm指代着单体的旋转角度，4对应着这个单体需要旋转4次才能完成一周（90°），3对应着120°，2对应着180°，1对应着没有旋转。Idx\_def对应着这个单体在程序中的命名。按照习惯，一般把它命名为1000-9999之间的整数。在define.f90中我将指数提前定义在了文件开头，所以在这里是以参数的形式出现。eva\_num对应着分子蒸发数目。alloc\_comp对应着一个单体有多少个点组成。比如这里的单体是121，则comp等与3。定义完之后，我们需要逐一设置点的位置和初状态。使用者需要设置好状态数目，避免状态数混用用导致的错乱。状态数可以大于10，并不会造成任何问题，但是不宜过大，会影响程序效率。

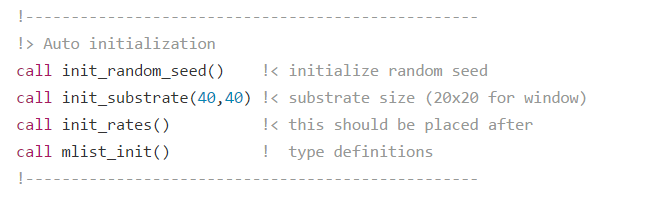
1. 对于每一种mtype，除了一些基本参数，还需要定义反应react（reaction）。每一个react对应着一种状态的变化（state change）。在KMC算法里面一个react将会对应着一个rate的计算（也就意味着一个react对应着有且仅有一个能量）。
2. 每一个react下面，你需要设置反应元素的数目cond（condition）。这里反应元素可以是同种，也可以是不同种，取决于反应元素状态与位置有多复杂。一般来说一种mtype对应一种cond，但遇到比较复杂的情况有例外。程序在执行的时候，会确保每一条cond都成立，否则（若有一条以上cond不成立），此react的rate会变为0，即不发生次反应。
3. 每一个cond下面还需要设置选项opt（option）。对于一个cond，当有一条或者大于一条opt成立的时候，这个cond成立。一个选项定义了一个cond的位置和状态。如下图。set\_pos定义了cond所对应元素的位置【x，y，d】。使用者可以定义多个位置，按照【x1，y1，d1，x2，y2，d2，… 】的顺序。仅当所有的位置都满足之后的状态条件的时候，opt成立。set\_state定义了元素状态的变化。其定义的方法是【位置1初状态，位置1末状态，位置2初状态，位置2末状态，… 】若所有位置的初状态都和这里列出的数目一致，则此opt成立。若此选项得到执行，所有位置的状态将由初状态改成末状态。



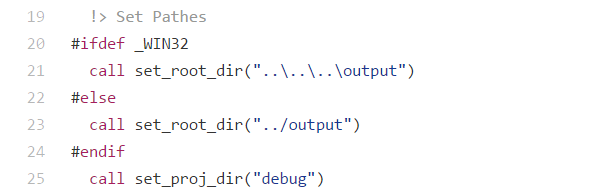
一般来说逐条定义反应十分复杂，在将来的版本中或许会被修正，但就目前来说还是需要忍耐一下。建议通过定义函数的方式来避免重复写代码，比如我通过定义函数，简化了书写反应的次数。这只是众多方案中的一种。



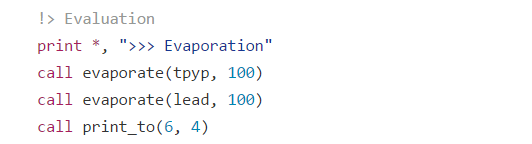
在define.f90文件最后，需要设置衬底尺寸。此处尺寸为40×40



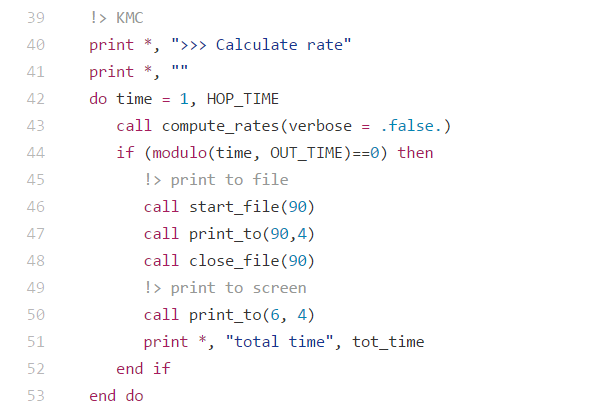
在main.f90中你可以修改project名称



或者蒸发的方式



甚至可以在之后的hop中同时蒸发分子，这样做不会影响程序执行。

这是KMC的执行过程。使用的时候可以任意修改执行的方式，start\_file/print\_to/close\_file只是把衬底打印到文件。6指代屏幕，别的数字指代文件（一般用90）。4代表着打印方法。一般是用4。可以通过阅读func\_substrate.f90来查找其他选项

如果需要控制温度，直接修改env\_temp变量，可以在hop的loop里面随时修改。改完当场生效（至少目前来说是这样）。温度的变量定义在globle.f90中，是一个全局变量。如果需要修改沉底的fold数目，你需要首先修改global.f90中的SYMM\_NUM数值，然后定义class/class\_mtype.f90文件中m\_set\_symm函数。你需要新增rotation matrix。我已经定义了4度对称的衬底。关于6度的衬底请自行定义。

