

# pyFluent | 一点使用体验

原创 胡坤 CFD之道 2022-07-13 08:34 发表于四川

收录于合集

#pyFluent

4个

PyFluent是Fluent 2022R2版本推出的新玩意儿。最近体验了一波，感觉这货很有要头。

Fluent提供了很多种不同的控制方式，不过最常见的无异于GUI和TUI两种。GUI适合于新手操作及单case使用，TUI适合参数化及批量计算。按理说应当能够满足需求了。不过Fluent推出的PyFluent还是有好多TUI无法提供的特性。

首先PyFluent使用的脚本语言是Python，且可以通过远程访问的方式连接Fluent内核并对其进行操控，通过采用这种CS结构访问Fluent，事实上已经将脚本编写与Fluent独立开来，灵活性上要远胜TUI，更不要说GUI了。

另外，Fluent TUI虽然写的也是脚本，不过并不具备完善的程序结构（比如常见的逻辑分支、循环等），当然TUI可以借用Scheme脚本来实现逻辑分支与循环处理，但无疑会使得脚本变得更加复杂及难懂。利用PyFluent就完全不用考虑这个问题了，其使用完整的Python作为程序控制语言，而Python是完善的计算机语言，其表达能力远胜TUI。

Python目前是最为火热的计算机语言，其拥有大量的可供利用的功能包和模块，能够实现非常复杂的科学计算功能（如常用的numPy、scipy、sympy等），PyFluent既然基于Python，自然也是可以使用这些模块的。比如当前比较火热的人工智能机器学习什么的，也可以混着Fluent一起用了。看起来能耍的地方还挺多。当然最实用的还是利用PyFluent进行仿真参数化计算，简单方便且无需借助外部工具。

另外，PyFluent完全兼容TUI，因此如果对TUI比较熟悉的话，其实很容易将其转换为PyFluent代码，只要稍微有点儿Python基础，PyFluent的学习成本几乎为零。

下面是PyFluent文档中的一些描述。

## 1 PyFluent是什么

PyFluent是PyAnsys（文档地址：<https://docs.pyansys.com/>）生态系统的一部分，其允许用户在Python环境中与其他PyAnsys库和其他外部Python库一起操纵使用Fluent。

PyFluent实现了一个客户端-服务器（Client-Server）架构。PyFluent利用Google Remote Procedure Call或GRPC接口将正在运行的Fluent进程作为服务器启动或连接，用户只需要通过Python接口与之交互。

用户可以使用PyFluent以编程的方式创建、交互和控制Fluent会话，以创建和定制工作区。此外还可以利用PyFluent通过高度可配置的定制脚本来提高工作效率。

PyFluent的主要模块 **ansys-fluent-core** 提供以下功能：

- 能够以串行或并行方式启动Fluent求解器，并使用 **launching Fluent** 模块连接到已运行的Fluent会话
- 能够编写Fluent Meshing功能脚本
- 能够使用Fluent的所有TUI命令编写脚本
- 能够异步运行多个Fluent会话
- 能够使用标准Python库（如matplotlib）以numpy array的形式检索Fluent的物理场数据，以实现自定义后处理
- 能够在Fluent的求解器事件上注册回调函数，例如可以在读取case或data文件时，或者Fluent求解器完成迭代时实现一些特殊的功能
- 能够使用MonitorsManager模块检索求解器监视器，例如残差

## 2 PyFluent与TUI

TUI命令指的是Fluent文本用户界面（Text User Interface, TUI）。PyFluent TUI命令代表了一种全面自动化工作流的方法：Fluent TUI（其本身是一个全面的自动化界面）中的所有内容都在PyFluent中公开。PyFluent TUI命令是Fluent控制台中使用的那些命令的Pythonic版本。

PyFluent TUI命令不支持一些TUI功能，如别名或命令缩写。此时的选择是如果安装了诸如pyreadline3之类的工具，在交互式会话中使用这些PyFluent命令会更容易，该工具提供了命令行智能补全和历史记录。用户可以在任何PyFluent TUI对象上使用Python内置函数、help和dir来进一步检查它。

TUI命令的参数只是那些在与Fluent控制台直接交互时传递的参数，但PyFluent采用Pythonic风格。用户可以很容易地将TUI命令转换为PyFluent代码。例如在Solution模式中，下面的TUI命令可以用来设置速度入口边界：

```
/define/boundary_conditions/set/velocity-inlet
```

上述命令的调用会在控制台中引发一系列输入提示，如果用户依次响应每个提示：

```
velocity-inlet-5  
(  
temperature  
no  
293.15  
quit
```

其效果与直接在TUI命令中指定参数的方式相同：

```
/define/boundary-conditions/set/velocity-inlet velocity-inlet-5 () temperature no 293.15 quit
```

交互式TUI是构建包含完整参数序列的TUI调用的可靠方法。有了完整的TUI，下一步是将其转换为Python调用：

```
from ansys.fluent.core import launch_fluent
solver_session = launch_fluent()
tui = solver_session.solver.tui
tui.define.boundary_conditions.set.velocity_inlet(
    "velocity-inlet-5", [], "temperature", "no", 293.15, "quit"
)
```

现在来看另一个TUI命令：

```
/define/units pressure "Pa"
```

可以将其改成python代码：

```
tui.define.units("pressure", "Pa")
```


其中字符串 **"Pa"** 被包裹在单引号中，以保留TUI参数周围的双引号。

注意上述示例中隐含的以下规则：

- TUI路径中元素之间的每个斜杠分隔符转换为Python点符号
- 路径元素中的某些字符要么被删除，要么被替换，因为它们在Python名称中是非法的：
  - 路径元素中的每个连字符都转换为下划线
  - 路径元素中的每个问号都将被删除
- 关于字符串的一些规则：
  - 必须在Python中引用字符串类型参数
  - 注意需要引用目标Fluent TUI参数的特殊情况（例如上面的“Pa”）。必须通过在附加的单引号中包裹Python字符串来保留该引号
  - 保留字符串参数的内容

（完毕）

# 『CFD之道』学习资源



CFD视界  
「录播课」



CFD视点  
「直播课」



析模界  
「服务号」

收录于合集 #pyFluent 4

上一篇

pyFluent | 常用的代码片段

下一篇

pyFluent | 设置Fluent计算参数

喜欢此内容的人还喜欢

总结 Vue3 的七种组件通信方式

前端潮咖



一文解析 Pinia 和 Vuex，带你全面理解这两个 Vue 状态管理模式

前端瓶子君



UI技巧：如何高效、简单实现菜单拖拽排序

Carson带你学Android

