
通过 SpaceClaim 脚本创建 Fluent 监控点

1 前言

在现代流体力学数值模拟领域，尤其是针对复杂旋转机械的计算流体动力学（CFD）研究中，监控点的布置是关键环节之一。流场中的监控点用于实时跟踪和分析流场中的关键物理量，如压力、速度和温度等，这些数据对于理解流动特性、优化设计以及预测设备性能至关重要。然而，传统的 Fluent GUI 环境在处理大量监控点时存在诸多局限性。一方面，手动输入大量监控点不仅工作量巨大，而且图形界面容易出现卡顿现象，这极大地降低了工作效率；另一方面，手动输入还容易导致错误，进而影响模拟结果的准确性。此外，对于旋转机械的数值计算，通常需要在每个流道的不同位置设置监控点，而低版本的 Fluent 缺少监控点的旋转复制功能，这使得流道监控点的布置变得尤为繁琐和耗时。

为了克服这些挑战，我们基于 SpaceClaim 的脚本功能开发了流场监控点设置功能。SpaceClaim 以其强大的三维建模功能而闻名，能够高效地处理复杂的几何结构。利用 SpaceClaim 的镜像、旋转复制等功能，我们能够快速且准确地布置大量的监控点。这有效简化了监控点布置的流程，还显著提高了工作效率，减少了人为错误的发生，使得研究人员和工程师能够更加专注于核心的流体力学问题分析，而无需在繁琐的监控点布置上耗费过多精力。

本文档旨在详细阐述该脚本的使用方法、功能特点以及操作步骤，以帮助用户快速掌握如何利用该工具高效地布置监控点，从而提升其在流体力学数值模拟中的工作效率和准确性。

2 使用说明

脚本的使用主要有 3 个步骤，分别是监控点的创建与组合、Fluent 监控点的创建/保存和 Fluent 监控点创建。本脚本仅适用于 3D 坐标，对于 2D 坐标并未在功能上进行开发。对于同时存在移动监控点和固定监控点的情况，建议先在 Fluent 设置好固定监控点，最后再设置移动监控点。

2.1 监控点的创建与组合

脚本主要通过 SpaceClaim 中 DatumPoints 获取点的三维坐标，如图 1 所示。SpaceClaim 具有多种阵列复制方法，可根据需求快速创建大量点，在旋转机械数值计算中，这个功能可大大简化监控点的创建工作。若使用镜像功能，软件将镜像复制的点放于 Pattern 的阵列中，如图 2 所示。脚本要求，需要对阵列进行“取消阵列”操作。

此时，点将释放在结构树的根结构上，如图 3 所示，需要对相应的点人为进行编组和命名，编组命令为“移到新组件”（Move to New Component），组的命令可直接用于后续 Fluent 的命名中，因此需要遵循 Fluent 监控点命名规则（只能

使用英文字符，不能包含下划线_和空格)。脚本中内置连接符，因此这里命名只需要填写前缀。同时，点的组件只能在结构树的根路径上，如图 5 所示。

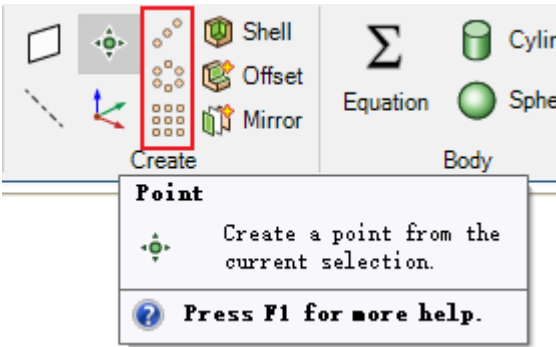


图 1 Design-Create-Points (点)



左图：结构树的阵列示意图

右图：取消阵列

图 2 点的阵列复制



图 3 点的编组

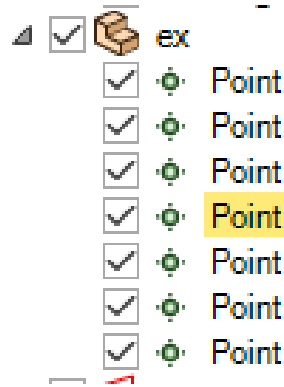
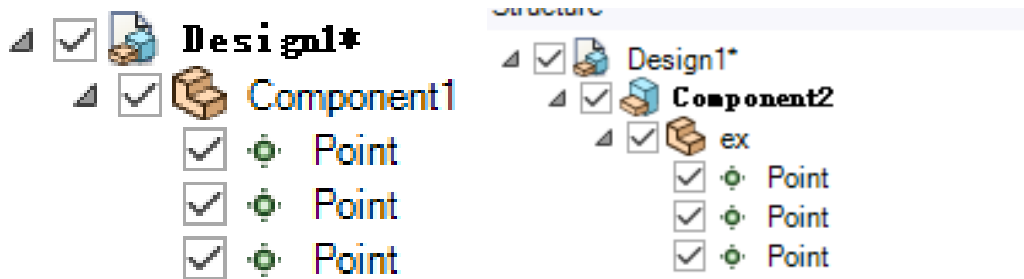


图 4 监控点的组件



左图：正确位置 右图：错误位置，叠加在一个组件下面

图 5 点组件的位置示意图

2.2 Fluent 监控点的创建

使用 SpaceClaim 的脚本（Script）功能，如图 6 所示，软件右侧将出现两个窗口，包括代码区和输出区，如图 7 所示。该功能语法基于 Python。通过 GitHub 复制代码到脚本编辑区，用于 Fluent 的监控点设置脚本在输出区中。代码位置保存在 GitHub 中，同时附有逐行代码说明，方便大家修改和开发。
(<https://github.com/MemorXuxu/spaceClaimScriptsPython>)。

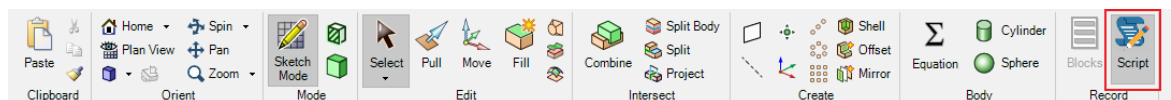
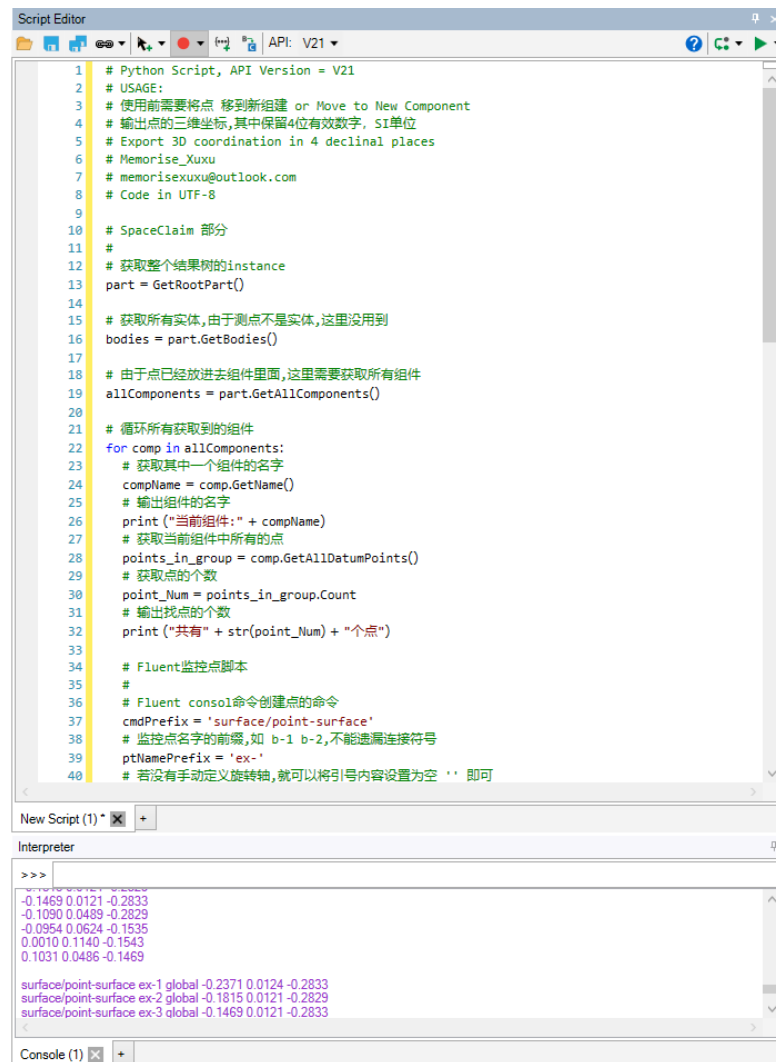


图 6 SpaceClaim 脚本



```
1 # Python Script, API Version = V21
2 # USAGE:
3 # 使用前需要将点 移到新建 or Move to New Component
4 # 输出点的三维坐标,其中保留4位有效数字, SI单位
5 # Export 3D coordination in 4 declinal places
6 # Memorise_Xuxu
7 # memorisexuxu@outlook.com
8 # Code in UTF-8
9
10 # SpaceClaim 部分
11 #
12 # 获取整个结果树的instance
13 part = GetRootPart()
14
15 # 获取所有实体,由于测点不是实体,这里没用到
16 bodies = part.GetBodies()
17
18 # 由于点已经放进去组件里面,这里需要获取所有组件
19 allComponents = part.GetAllComponents()
20
21 # 循环所有获取到的组件
22 for comp in allComponents:
23     # 获取其中一个组件的名字
24     compName = comp.GetName()
25     # 输出组件的名字
26     print ("当前组件:" + compName)
27     # 获取当前组件中所有的点
28     points_in_group = comp.GetAllDatumPoints()
29     # 获取点的个数
30     point_Num = points_in_group.Count
31     # 输出点的个数
32     print ("共有" + str(point_Num) + "个点")
33
34 # Fluent 监控点脚本
35 #
36 # Fluent consol命令创建点的命令
37 cmdPrefix = 'surface/point-surface'
38 # 监控点名字的前缀,如 b-1 b-2,不能遗漏连接符号
39 ptNamePrefix = 'ex-'
40 # 若没有手动定义旋转轴,就可以将引号内容设置为空 '' 即可
```

```
>>>
-0.1469 0.0121 -0.2833
-0.1090 0.0489 -0.2829
-0.0954 0.0624 -0.1535
0.0010 0.1140 -0.1543
0.1031 0.0486 -0.1469

surface/point-surface ex-1 global -0.2371 0.0124 -0.2833
surface/point-surface ex-2 global -0.1815 0.0121 -0.2829
surface/point-surface ex-3 global -0.1469 0.0121 -0.2833
```

图 7 脚本程序输出和输出

因此，此处只说明几处代码进行说明：

(a) **ptNamePrefix**: 该参数为监测点的名字前缀，脚本提供三种命名方式，如图 9 所示。

(b) **refAxis**: 对于移动监测点，如旋转叶轮，作者一般使用定义旋转轴的方法定义监测点（相关方法参考 <https://www.7b3.cn/3458.html>），此时点的定义需要旋转轴名称，即该参数。

(c) 脚本提供三种命名方式，如图 9，第一种为固定坐标的定义方式，监控点名称前缀使用脚本定义的参数 **ptNamePrefix**；第二种为旋转坐标的定义方式，监控点前缀与第一种一致；第三种为旋转坐标（若用于固定坐标，**refAxis** 需要定义为空，即 **refAxis=""**）的定义方式，监控点命名前缀为点组件的名称，如图 4 所示，Fluent 中监控点的命名格式为 **ex-1, ex-2, ex-3**

(d) 脚本三维坐标精度为小数点后 4 位（代码中为 `{:.4f}`），单位为米。

(e) 脚本提供将 Fluent 代码保存功能，默认保存路径为用户根目录，即 **C:\User\[用户名]**。

```

# Fluent consol命令创建点的命令
cmdPrefix = 'surface/point-surface'
# 监控点名字的前缀,如 b-1 b-2,不能遗漏连接符号
ptNamePrefix = 'b-'
# 若没有手动定义旋转轴,就可以将引号内容设置为空 '' 即可
refAxis = 'rotor' # 若没有手动定义旋转轴,就可以将引号内容设置为空 '' 即可
# refAxis = ''

```

图 8 脚本主要参数说明

```

# 没有自定义旋转轴,可以默认为global,或为空(代码采用了为空的方式)
#context = cmdPrefix + " " + ptNamePrefix + str(i)+' '+'{:.4f} {:.4f} {:.4f}'.format(point_group.Position[0],
point_group.Position[1], point_group.Position[2])
#print(cmdPrefix + " " + ptNamePrefix + str(i)+' '+'{:.4f} {:.4f} {:.4f}'.format(point_group.Position[0], point_group.Position[1],
point_group.Position[2]))

# 手动定义监控点名字前缀,引用 ptNamePrefix
#context = cmdPrefix + " " + ptNamePrefix + str(i)+" " + refAxis + ' '+'{:.4f} {:.4f} {:.4f}'.format(point_group.Position[0],
point_group.Position[1], point_group.Position[2])
#print(cmdPrefix + " " + ptNamePrefix + str(i)+" " + refAxis + ' '+'{:.4f} {:.4f} {:.4f}'.format(point_group.Position[0],
point_group.Position[1], point_group.Position[2]))

# 使用组件名称,比较推荐,要注意组件名称,这里自动添加了连接符号-
context = cmdPrefix + " " + compName + '-' + str(i)+" " + refAxis + ' '+'{:.4f} {:.4f} {:.4f}'.format(point_group.Position[0],
point_group.Position[1], point_group.Position[2])
print(cmdPrefix + " " + compName + '-' + str(i)+" " + refAxis + ' '+'{:.4f} {:.4f} {:.4f}'.format(point_group.Position[0],
point_group.Position[1], point_group.Position[2]))

```

图 9 监控点的三种命名方法

2.3 Fluent 监控点创建

通过 2.2 创建得到脚本,结果如 10 所示。其中, surface...前缀内容为 Fluent 使用脚本。在 Fluent 中, command TUI 中粘贴脚本内容(快捷键为 Shift+Insert, 不建议使用 Ctrl+V),如图 11 所示,并检查 TUI 中是否有创建失败的 warning。监控点创建错误原因主要有:命名输入位置不在 TUI 的根路径上、点的位置超出计算域、监控点的单位不正确,需在 Fluent 中 length 设置为 m。

```

当前组件:fpnd
共有6个点
0.0575 0.1549 0.0000
-0.1054 0.1272 0.0000
-0.1629 -0.0277 0.0000
-0.0575 -0.1549 0.0000
0.1054 -0.1272 0.0000
0.1629 0.0277 0.0000

surface/point-surface fpnd-1 rotor 0.0575 0.1549 0.0000
surface/point-surface fpnd-2 rotor -0.1054 0.1272 0.0000
surface/point-surface fpnd-3 rotor -0.1629 -0.0277 0.0000
surface/point-surface fpnd-4 rotor -0.0575 -0.1549 0.0000
surface/point-surface fpnd-5 rotor 0.1054 -0.1272 0.0000
surface/point-surface fpnd-6 rotor 0.1629 0.0277 0.0000

print("Finished")
Finished

```

图 10 Fluent 监控点脚本示意图

```
>
adjoint/           parallel/           solve/
define/            plot/             surface/
display/           preferences/       turbo-workflow/
exit               print-license-usage views/
file/              report/
mesh/              server/

> surface/point-surface fpend-1 rotor 0.0575 0.1549 0.0000
surface/point-surface fpend-2 rotor -0.1054 0.1272 0.0000
surface/point-surface fpend-3 rotor -0.1629 -0.0277 0.0000
surface/point-surface fpend-4 rotor -0.0575 -0.1549 0.0000
surface/point-surface fpend-5 rotor 0.1054 -0.1272 0.0000
surface/point-surface fpend-6 rotor 0.1629 0.0277 0.0000
```

图 11 Fluent 监控点创建