

RANSFOIL用户向导

版本1.7

2016 年 8 月 21 日

目录	2
----	---

目录

1 综述	3
2 安装	3
3 使用方法	4
4 导出文件	7
5 实例	8
6 模型和算法选择	8
7 外界条件和迭代参数设置	10
8 实际使用贴士	11
9 程序库用法	13
10 附加信息	16

RANSFOIL的版权声明是 BSD 协议, 请在使用本程序之前阅读根目录下的 COPYRIGHT 文件。本文档是 RANSFOIL 用户手册的简体中文版本, 用户如果想阅读英文文档, 可以参考同一目录下的 UserGuide.pdf 文件。

1 综述

RANSFOIL是一个通过数值求解雷诺平均 Navier-Stokes(RANS) 方程组计算低速、亚音速、跨音速或超音速条件下环绕孤立翼型的空气流场的控制台程序。作为后处理结果, 翼型的气动参数, 如升力系数和阻力系数, 可以通过近壁面的气流数据积分得到。此外, 本程序还计入了流动与传热的耦合以及表面粗糙度对摩擦阻力和传热量的影响, 可以用于翼型表面摩擦阻和传热系数的工程计算。本程序从一个一维的 XYZ 文件中读取翼型坐标作为唯一的输入, 然后输出记录网格坐标和流场数据的二维 PLOT3D 格式的网格和解函数文件和一个显示气动参数的报告文件。

2 安装

RANSFOIL 软件包采用 GNU Autotools 工具集在 Linux 和 Windows 平台上部署。

在 Linux 系统中, 熟悉的三个命令 (`./configure; make; make install`) 应该可以配置、生成和安装这个软件包。如下是本软件包的一些特殊配置选项。

本软件包使用 GNU Libtool 生成共享库。默认情况下, `make` 命令应该会生成一个名为 `libaero2d` 的共享库和一个名为 `ransfoil` (依赖于 `libaero2d`) 的可执行文件。使用者可以通过向 `configure` 命令指定 `--disable-shared` 选项关闭共享库的生成, 作为替代通过指定 `--enable-static` 选项来生成一个静态库:

```
$./configure --disable-shared --enable-static
```

你可以通过向 `configure` 命令指定 `--enable-hyprelib` 选项从而开启 LLNL 的 HYPRE 库用于求解大型稀疏线性方程组。如果你决定与 HYPRE 库一起编译, 你必须安装好 HYPRE 库的二进制版本并且确定

`--with-hyprelib-dir=DIR` 选项被用来为 HYPRE 库指定一个根安装目录 DIR（本软件包不能自动搜寻 HYPRE 库在文件系统中的位置）：

```
./configure --enable-hyprelib --with-hyprelib-dir=DIR
```

你可以通过向 `configure` 命令传递 `--enable-openmp` 选项来开启 OpenMP 并行化，目前本软件包只支持 GNU 的 OpenMP 实现。

```
./configure --enable-openmp
```

如果你想采用其它的 OpenMP 实现，向 `configure` 命令传递合适的 `LDFLAGS` 选项。

由于本软件包中默认的编译选项 `FCFLAGS` 是特别为 GNU gfortran 优化的，如果你想使用其它的 Fortran 编译器，向 `configure` 命令传递 `FC=XXX` 和 `FCFLAGS=XXX` 参数来确定相应的编译选项被正确地使用：

```
./configure FC=XXX FCFLAGS=XXX
```

在 Windows 系统中，你应该拥有一个 MinGW+msys 或者 Cygwin 环境。生成步骤将和 Linux 系统中的步骤完全相同。在 Windows 环境本软件包会生成一个动态链接库 (DLL) 来代替 Linux 中的共享库。

在其它的类 Unix 系统中，以上的生成步骤仍然可能是有效的，但作者并没有在这些平台做测试。

3 使用方法

本程序是基于控制台的由文件驱动的命令程序。生成可执行文件后，在终端键入 `ransfoil` 可以看到本程序的基本用法：

```
./ransfoil
```

```
usage:  ransfoil [options] [configuration file]
```

```
Options:
```

```
--stdin    interactive mode, read parameters from standard input
```

```
--script   batch mode, read parameters from a configuration file
```

```
-h,--help  show this message
```

本程序有两个执行模式，交互模式和批处理模式。

你可以使用 `--stdin` 选项进入交互模式：

```
./ransfoil --stdin
```

用户需要输入一些配置参数, 包括一些选择模型和算法的字符串和设置外界条件的数字。在你输入参数前相应的提示信息会显示在终端屏幕上。对于字符参量, 在提示信息末尾的括号内会有可选的提示字符（由符号/分隔）, 你可以从提示字符中选择一个来配置模型和算法。对于数字参量, 在提示信息末尾的括号内会给出它们的单位（没给出单位的数字参量属于无量纲数）。你可以使用任何合法的浮点数格式（去除单位）来输入数字参量, 例如, 0.1或1e-1。任何附加的字符或数字都会被程序视为非法。本段的以上内容只是解释了如何在本程序中设置配置参数, 各个参数的详细信息将在下面的部分叙述。

你可以使用 `--script` 选项进入批处理模式。例如, 如果你有一个名为 `NACA0012.config` 的配置文件, 你可以使用如下命令执行 `RANSFOIL` :

```
$. /ransfoil --script NACA0012.config
```

本程序会从配置文件中读取全部配置参数然后开始迭代, 在程序终止前用户不需要做任何交互。配置文件的格式是类似于交互模式的纯文本, 不同点在于在批处理模式你应该在执行程序前将所有配置参数输入配置文件中。配置文件应当被组织为一种隔行格式, 例如

```
-  
NACA0012.xyz  
-  
incom  
-  
sa  
-  
...  
-  
100.0  
-  
.  
.  
.
```

配置参数的顺序必须与在交互模式中显示的顺序一致, 并且配置文件的第一行必须是空行。你可以在全部空行中输入提示信息, 这些信息不会被程序识别。在本软件包中有一个名为 `ransfoil.config` 的示例配置文件, 可以用来作为详细使用的模板。

你可以使用 `-h` 或 `--help` 选项或者单独输入可执行文件名来显示一些简洁的描述和使用信息:

```
$/ransfoil --help
```

在程序执行前, 你应该准备一个记录翼型坐标的数据文件。这个文件应该是一维 XYZ 格式, 它将坐标数据分成两个部分: 下翼面和上翼面。每一部分都应该从一个描述本部分坐标点个数的数字开始, 然后紧跟从前缘点到后缘点的 X-Y 坐标值, 例如:

下翼面坐标点数(N)

X(1) Y(1)

X(2) Y(2)

..

..

X(N) Y(N)

上翼面坐标点数(N)

X(1) Y(1)

X(2) Y(2)

..

..

X(N) Y(N)

翼型坐标应该是弦长为1的无量纲形式。这些坐标数据在计算流程中不会被更改, 两坐标点之间的部分假设为一 条线段。你应该确定原始的数据文件提供了足够的几何信息去描述翼型形状。作为参考, 本软件包中有一个名为 `NACA0012.xyz` 的示例翼型坐标文件。在交互模式开始处, 或者在批处理模式的配置文件头位置, 用户应该提供翼型坐标文件的文件名。文

件名可能包含完整路径，并且其长度不应超过64个字符。

4 导出文件

当RANSFOIL正在执行时，程序会生成一个记录迭代信息的名为 iteration.txt 的文本文件。经过一定的迭代步后，程序会创建名为 Grid.xyz 的网格数据文件，名为 Solutions.dat 的解函数文件和名为 Solnames.nam 的解函数名称文件。这三个文件是记录网格坐标和气流数据的标准二维 PLOT3D 文件。同时，一个自动备份全部输运方程数值解的二进制数据文件 Autosave.dat 也被生成。另外，程序会创建保存近壁面气流数据的文件 Wallsol.dat 和报告气动参数的文件 Aeroreport.txt。你可以使用可视化软件包，例如 Paraview、Mayavi2 或者 gnuplot 来显示这些数据，但怎样使用这些软件包则超出了本文档的范围，用户可查阅相关文档。对于那些以 VTK 作为可视化引擎，并且不能很好地支持 PLOT3D 格式的软件包，一个额外的传统 vtk 格式的文件 GridSol.vtk 也被生成并用来作为输入数据集。当程序正常终止时，最终结果会保留在最后的数据文件中。

为方便查看，几乎所有的导出文件都是ASCII格式（除 Autosave.dat 之外）。包含在二维 PLOT3D 文件 (Grid.xyz, Solutions.dat, Solnames.nam) 中的流场变量为X-坐标，Y-坐标，X-速度，Y-速度，静压强，静温度，空气密度和一些湍流相关的变量（取决于你选择的湍流模型），传统 vtk 文件 GridSol.vtk 中的流场变量与在PLOT3D文件中的变量一致。在文件 Wallsol.dat 中的近壁面流场变量包括X-坐标，Y-坐标，一种翼型上的弧长（以前缘点为原点），和相应位置的近壁面压强，近壁面速率，近壁面温度和近壁面湍流黏度。另外，作为后处理结果，X和Y方向的壁面切应力（摩擦应力），表面传热系数和无量纲壁面距离也存储在这个文件中。在报告文件 Aeroreport.txt 中可以找到翼型的全部气动参数，包括升力系数，阻力系数（由压强引起），摩擦系数，俯仰力矩系数（对1/4弦长取矩）和压力中心的位置（以弦长为单位）。

在交互模式的最后，或者在批处理模式的配置文件末尾，用户应该输入导出文件的目录名。如果你想使用当前目录，只需输入‘.’。这个参数可以用来控制批处理过程。例如，如果你想在同一个翼型上运行两个算例

(每个算例对应唯一的配置参数组合), 你可以创建两个不同的配置文件和两个不同的子目录, 然后将每个子目录名称输入对应的配置文件中。如果你能写一个可靠的 Shell 脚本, 导出文件将被分配到各自的子目录中。

5 实例

用户可以使用示例翼型坐标文件 NACA0012.xyz 和配置文件 ransfoil.config 启动一个入门算例。生成可执行文件后, 你只需要在软件包根目录输入

```
$. /ransfoil --script ransfoil.config
```

程序会进入批处理模式, 并使用 Spalart-Allmaras 湍流模型计算弦长为 0.5334m 的 NACA0012 翼型在一个低速条件 (自由流速率 75m/s, 攻角 4 度) 下的空气绕流和气动参数。前述的所有导出文件都会被生成并用来作为可视化数据和报告文档。如果你想进入交互模式, 键入

```
$. /ransfoil --stdin
```

并根据示例配置文件中的提示信息 and 数值输入相应的配置参数。

6 模型和算法选择

实际上, 本程序是一个特别为翼型气动计算而设计的二维稳态 RANS 求解器。有关选择空气动力学模型和数值算法的配置参数都属于字符串常量。以下是各个配置参数的详情和一个简短的选择向导。

本程序模拟的空气密度可以是恒定的或变化的。用户可以使用配置参数 'incom' 或 'com' 来设置空气的物理属性, 对应不可压缩性与可压缩性。用户可以在本程序中选择一些湍流模型, 这些湍流模型包括标准 $k-\epsilon$ 模型、Spalart-Allmaras (S-A) 模型和切应力输运 (SST) $k-\omega$ 模型。用户可以使用配置参数 'ke'、'sa' 或 'sst' 来选择湍流模型, 分别对应 $k-\epsilon$ 、S-A 或 SST 湍流模型。S-A 和 SST 模型是特别为空气动力学计算而设计的, 一般情况下 S-A 模型的计算花费要小于 SST 模型。 $k-\epsilon$ 模型有更广的普适性, 但在空气动力学问题的近壁面区域的精度较低。你也可以通过设定配置参数 'inv' 或 'lam' 拒绝使用湍流模型。如果你设定 'inv', 空气被认为是无黏的,

RANS 方程组退化为 Euler 方程组。如果你设定 ‘lam’，标准 Navier-Stokes 方程组被用来作为全部的输运方程。在每个湍流模型中近壁面流动都应该被特殊处理，本程序采用两种方法计算近壁面流动：壁面函数法和低雷诺数法。对于 S-A 和 SST 湍流模型，可以通过配置参数 ‘wf’ 或 ‘lr’ 来设置全部两种方法，分别对应壁面函数法和低雷诺数法。对于 $k-\epsilon$ 模型，只有壁面函数法可以使用。在设置 ‘ke’ 作为湍流模型之后，你不应该在批处理模式的配置文件中再设置有关近壁面处理的配置参数。

用户可以使用配置参数 ‘Y’ 或 ‘N’ 来通知程序是否把能量方程包含在 RANS 方程组中。如果你想得到翼型表面的传热系数（保存在 Wallsol.dat 数据文件中）或者你刚刚将空气属性设置为可压缩的（输入 ‘com’），你必须开启能量方程（输入 ‘Y’）。当求解能量方程时，粘性加热项可能会出现在能量方程的右侧。如果能量方程已开启，用户可以使用配置参数 ‘Y’ 或 ‘N’ 来通知程序是否在能量方程中包含粘性加热项。如果你通过设置 ‘Y’ 来包含附加的加热项，根据你之前设定的空气属性存在两种不同的情况。如果你将空气属性设置为不可压缩的，只有粘性加热项会包含在能量方程中。如果你将空气属性设置为可压缩的，压缩加热项会与粘性加热项一起加入能量方程中。不同的附加加热项将导致表面传热系数的不同计算结果。

如下的一些配置参数可以定义本程序采用的数值算法和格式。用户可以通过配置参数 ‘SIMPLE’ 或 ‘SIMPLEC’ 来选择 SIMPLE 或者 SIMPLEC 算法处理速度与压强的耦合。多数情况下 SIMPLEC 算法具有比 SIMPLE 算法更快的收敛速度。用户可以使用一些空间格式来离散输运方程。通过设定配置参数 ‘1upwind’、‘2upwind’ 或 ‘Quick’，空间离散将采用一阶迎风、二阶迎风或 Quick 格式，分别对应对流项的一阶、二阶或三阶空间精度。如果你已将空气属性设置为可压缩的（输入 ‘com’），控制体界面的空气密度插值格式应该被特殊处理。你可以通过配置参数 ‘center’、‘1upwind’、‘2upwind’ 或 ‘Quick’ 选择中心、一阶迎风、二阶迎风或 Quick 格式来插值空气密度。为了精确地捕捉高度可压缩流内部的激波，一种可以消除激波附近震荡的 TVD 格式被用来作为对流项的可选空间格式。这种 TVD 格式也可以被用作空气密度的一种可选的插值格式。你可以使用配置参数 ‘tvd’ 选择 TVD 格式来离散对流项或者插值空气密度。

在计算流程中，如果迭代步数达到某些值，二进制文件 Autosave.dat

会周期性地生成。所有流场变量的数值解都保存在这个文件中。你可以使用配置参数 ‘Y’ 或 ‘N’ 通知程序是否使用一个数据文件来初始化各流场变量。如果你设定 ‘Y’ 来从一个文件初始化，你可以使用 Autosave.dat 作为初始化文件。你应该确保程序至少运行过一次并得到一个 Autosave.dat 文件，你不应修改此文件中的内容。你也可以使用一个外部数据文件来初始化，无论哪种情形你都应该在交互或批处理模式中输入初始化文件的文件名。如果你设定 ‘N’ 来通过内部程序初始化各流场变量，你还应该设定一个额外的配置参数。如果你设定 ‘Y’ 来借助驻点值初始化，流场变量将被设置成驻点附近的值（速度设置为接近零的值，压强设置为自由流总压，温度设置为自由流总温），否则（设定 ‘N’），流场变量将会使用外界条件的值来初始化。

7 外界条件和迭代参数设置

有关外界条件和迭代参数的配置参数属于浮点型常量。翼型气动问题中典型的外界条件包括自由流速率、静温、静压和攻角。用户可以通过相应的配置参数输入这些外界条件。如果你想使用一个湍流模型来模拟湍流脉动，一些表征自由流中脉动量的配置参数也应该被设置。如果你选择了 S-A 模型（输入 ‘sa’），自由流湍流黏度比应该被设置，典型值在1到100之间，对应从低湍流黏度到高湍流黏度的区间。如果你选择了 $k-\epsilon$ 或 SST 模型（输入 ‘ke’ 或 ‘sst’），还需输入一个额外参数：自由流湍流强度，表征自由流中脉动速度的大小，典型值在0.01到0.1之间。此外，一些其它的条件可以用来描述翼型，包括弦长、壁面温度和翼型表面的等效砂粒粗糙度（设定为0来模拟光滑壁面）。这些外界条件被认为是相应输运方程的边界条件。在迭代开始前，本程序自动生成一个离散用的二维C型结构网格。用户应该输入无量纲近壁面网格间距（一维 XYZ 文件中的翼型坐标是无量纲的），它决定了对近壁面流动的处理并且对得到精确的结果十分重要。关于近壁面网格间距选择的一个简短向导将在下一节中讨论。另外的两个配置参数可以用来调整沿翼型表面法向的网格点分布。你可以设置近壁面网格的层数，近壁面网格层中的网格间距是均匀的并且设置为上述的无量纲近壁面网格间距。为得到更好的近壁面结果，一些湍流模型对近壁面网格的

层数有特殊要求。你也可以输入沿翼型表面法向的网格点总数，当然网格点总数包含近壁面网格层，并且确保网格沿法向延伸到足够远的自由流区域。因此网格点总数应该设置为一个远大于你之前设置的近壁面网格层数的整数。

用户可以设定一些迭代参数，包括最大迭代步数，最小残差和松弛因子。最大迭代步数和最小残差决定了一个实际迭代过程的终止准则。如果当前迭代步数超过最大迭代步数，或者当前残差小于最小残差，迭代就会被终止。你可以同时设置一系列松弛因子的值，包括速度、压强、温度、密度和湍流松弛因子。哪个松弛因子应该被设置取决于你之前选择的模型（湍流模型、空气属性等）。各松弛因子的值有一个0的下限和一个1的上限。这个值会影响收敛速率，一个较大的松弛因子可以导致更快的收敛但更差的数值稳定性。相反一个设定较小松弛因子的迭代会变得较慢但更加健壮。

8 实际使用贴士

示例配置文件 `ransfoil.config` 中并不包含本程序支持的全部配置参数，而只是一个特定算例所需的配置参数组合。有些配置参数的出现依赖于前面设定的其它参数。如果你不确定配置参数的排列顺序或组合方法，可以首先在交互模式下运行本程序，如果成功进入迭代过程，之前在终端屏幕上输入的配置参数即为正确的组合。

用户在运行时可以通过两种方式监视本程序。第一，每一个迭代步后质量残差（连续方程的残差）会打印在终端屏幕上。第二，包括全部输运方程残差的更为详尽的信息将保存在文本文件 `iteration.txt` 中。

当迭代发散时本程序的进程可能会被阻塞或异常终止。消除发散的最有效方式是减小松弛因子的值。在一个设定较小松弛因子的初始迭代中，当生成文件 `Autosave.dat` 后，用户可以尝试终止当前迭代，并且开始一个设定较大松弛因子的新迭代。为此你需要通过设定 ‘Y’ 来从一个文件初始化，并设定从 `Autosave.dat` 读取初始化数据。

如果你希望计算跨音速或超音速条件的空气流动，当你直接运行空气属性设定为可压缩（通过输入 ‘com’）的迭代时，高度的可压缩性可能会

导致发散。在跨音速范围，你可以首先运行一次设定较小松弛因子和不可压缩空气属性（通过输入‘incom’）的迭代，在一些迭代步过后会生成文件 Autosave.dat。你可以通过设定‘Y’（从文件初始化的配置参数）使用这个文件作为出发点，然后重新运行一次设定了较大松弛因子和可压缩空气属性（通过输入‘com’）的迭代。在新迭代中收敛可能会实现。在超音速范围，由于流动特征与亚音速范围有很大不同，采用不可压缩结果作为初始化数据的方法可能是无效的。在这种情况下你可以将流场变量初始化为驻点值，这种初始化更符合超音速气流的演化规律。这意味着你应该输入‘N’来否定从文件中初始化，然后输入‘Y’来使用驻点值初始化。

用户可以在报告文件 Aeroreport.txt 中检查 y^+ （S-A 或 SST 湍流模型）或者 y^* （ $k-\epsilon$ 湍流模型）的平均值。这个无量纲数代表一种湍流视角的壁面距离，并且对于选择一个可靠的近壁面网格间距（正比于 y^+ 或 y^* ）十分重要。对于S-A或SST模型，如果你设定‘wf’（壁面函数）作为近壁面处理方法， y^+ 的取值可以在一个从1到300的很大的范围内（这些模型实现了一种增强壁面函数，对于粗糙壁面 y^+ 的值可以大于300），相反如果你设定‘lr’（低雷诺数法）作为近壁面处理方法， y^+ 的取值应该在1的量级（ $y^+ \sim 1$ ，粘性子层）。对于 $k-\epsilon$ 模型，由于模型只实现了标准壁面函数， y^* 的取值范围应该在从30到300（ $30 < y^* < 300$ ，对数率层）。以上意味着你应该至少运行一次程序（设定某个近壁面网格间距）来得到一个 Aeroreport.txt 文件，然后检查这个文件中的 y^+ 或 y^* 的平均值并修改近壁面网格间距的设置直到这个平均值满足各种情况下的 y^+ 或 y^* 的取值范围！尽管如此，针对翼型绕流的特殊问题仍然有一些选择近壁面网格间距的经验准则。在亚声速范围，粘性子层中典型的无量纲近壁面网格间距约在 $10^{-6} \sim 10^{-5}$ 量级，对数率层中典型的间距值约在 $10^{-4} \sim 10^{-3}$ 量级。但对于跨音速或超音速流，由于边界层厚度随着雷诺数（自由流速率）的增加而减小，近壁面网格间距的值应该小于亚声速范围的典型值。

如果你设置了一个接近粘性子层的相对较小的近壁面网格间距，一个设定较大松弛因子的实际迭代可能会发散。你只需要减小松弛因子的值或者使用在压缩性导致的发散问题中采取的策略来达到收敛。

如果开启了本软件包的 OpenMP 并行化，用户可以在运行 RANSFOIL 之前设置环境变量 OMP_NUM_THREADS 的值来调整运行时线程数。由

于 MinGW64 环境中 libgomp 实现存在的某些问题，目前在 Windows 系统中最好以两个线程运行本程序，即将 OMP_NUM_THREADS 的值设置为 2。

9 程序库用法

本软件包的使用方式除了直接使用 RANSFOIL 程序获得导出文件之外，还为开发者提供了共享库 libaero2d 的应用开发接口。如果你想使用本软件包的开发接口，需要确认源代码包被正确编译并安装。如果源代码包成功安装，在常见的系统目录中（如 /usr/lib、/usr/include）中会包含本软件包的头文件和库文件。

开发者可以使用共享库中的功能来扩展自己的应用程序。共享库 libaero2d（源文件用 Fortran90 编写）中的主要数据集包含在一个名为 aero2dcom 的 Fortran 模块中，库内的多数子例程通过引用这个模块来对其中的公共数据做一些特定的操作。一个名为 Aero2D 的子例程位于调用树的根节点，通过调用其它子例程来实现共享库的多数功能。开发者可以采用 Fortran90(90, 95, 2003) 语言调用这个根节点子例程实现 libaero2d 库的主要功用，一个调用 libaero2d 库中子例程的 Fortran90 代码示例如下：

```
...
use aero2dcom
integer printlevel
character(1) mode
character(64) scriptname
.
.
.
Call Allocarray(mode)
Call Aero2D(mode,printlevel,scriptname)
.
.
.
Call Deallocarray(mode)
```

...

第一行代码通过引用 `aero2dcom` 模块来获取模块内全部数据的接口，后面三行代码为 `Aero2D` 子例程的参数声明，之后位于代码块中间的调用代码（以 `Call` 开头）是一个调用实例。首先，你应该通过调用 `libaero2d` 库中的子例程 `Allocarray` 来为 `aero2dcom` 模块中的动态数组分配内存空间。然后调用根节点子例程 `Aero2D`，当然在调用前传递给 `Aero2D` 的三个实际参数应该已经被赋值。`Aero2D` 子例程的参数分别是调用模式参数 `mode`，打印信息级别参数 `printlevel` 和脚本文件名参数 `scriptname`。其中 `mode` 和 `scriptname` 属于字符串变量，`printlevel` 为整型变量。最后，同样从属于 `libaero2d` 库中的子例程 `Deallocarray` 被调用来释放 `aero2dcom` 模块中的动态数组。注意子例程 `Allocarray` 和 `Deallocarray` 存在一个实参，这个实参就是调用模式参数 `mode`。示例代码中的三个子例程是使用 `libaero2d` 库实现一个调用实例的最常见方式。

`libaero2d` 库存在三种调用模式：交互模式、脚本模式和接口模式，分别通过将调用模式参数 `mode` 赋值为 'I'、'S' 和 'A' 来实现（Input、Script 和 API 的缩写）。交互模式的调用过程会被阻塞并等待用户从标准输入键入相应的配置参数（即前面几节描述的 RANSFOIL 程序的配置参数）。脚本模式需要从一个配置文件读取配置参数，如果你通过脚本模式调用 `Aero2D`，子例程的第三个参数 `scriptname` 被用来传递脚本文件名，其它调用模式不需要用到这个参数，`scriptname` 可以被赋值为空字符串。`printlevel` 参数代表调用过程中打印在终端上的信息详尽程度，有两个整型值可以使用：0和1，分别对应打印简单信息和打印详尽的迭代信息。以上两种调用模式都是基于文件的，`libaero2d` 库需要从文件中读取翼型坐标值，并且将迭代信息和流场数据保存在文件中，`aero2dcom` 模块中的数据集并不直接被调用者使用，因此在这两种模式中可以不引用 `aero2dcom` 模块，示例代码的第一行可以被忽略。接口模式是 `libaero2d` 库提供的不依赖文件的调用模式，在这种模式中必须引用 `aero2dcom` 模块。由于 `aero2dcom` 模块中的数据集全部暴露给了开发者，开发者可以通过可编程的方式使用这些数据配置 `libaero2d` 库。采用接口模式调用 `libaero2d` 库不会导出任何文件来保存数据，全部数据都驻留在内存中以供开发者下一步使用，除非调用者释放了 `aero2dcom` 模块中的全部动态数组。例如，

开发者可以为 `aero2dcom` 模块中的字符串变量赋值来设定配置参数，或者，开发者可以选择从文件读取或者直接赋值 `aero2dcom` 模块中代表翼型坐标的一维浮点数组（后一种方式在翼型优化设计中十分有用）。而且，`aero2dcom` 模块中同样包含代表翼型气动参数的全部变量，开发者可以在调用子例程 `Aero2D` 后直接引用这些变量来获取气动参数的最终结果。开发者还可以引用以 `aero2dcom` 模块中二维浮点数组形式实现的流场变量来保存场数据或使用可视化引擎绘图。`aero2dcom` 模块中常用变量的具体含义可参考对应源文件 `COM.f90` 中的注释部分。注意如果你想连续两次调用 `Aero2D`，配置是否通过数据文件初始化的配置参数（在 `aero2dcom` 模块中的变量名为 `Init`）可以被赋值为 `'A'`（既不是 `'Y'` 也不是 `'N'`）。这是一个采用上一个迭代的结果来初始化流场变量的特殊选项，尽管在新的迭代中环境条件可能会改变。在源文件包根目录中包含一个名为 `libcallexpf.f03` 的示例源文件，演示了如何通过接口模式调用 `libaero2d`。至于如何通过其余两种模式调用 `libaero2d`，用户可参考 `RANSFOIL` 程序（`RANSFOIL` 程序就是采用交互和脚本模式调用 `libaero2d` 库实现的）的源文件 `console.f03`。

感谢 Fortran03 标准中的 `ISO_C_BINDING` 内部模块，从 1.5 版本开始开发者可以使用 C 语言调用 `libaero2d`。在源代码包中有一个名为 `aero2dcom.h` 的 C 头文件，包含 `aero2dcom` 模块中的常用变量和 `libaero2d` 库中的常用函数的外部声明。C 开发者可以通过在自己的源文件头位置包含这个头文件来引用这些变量或函数。由于 `aero2dcom` 模块中的字符串和动态数组不能很好地和 C 语言交互，你可以使用对应的 C 配对变量，这些变量标示符仅仅在原始的 Fortran 变量名前加了一个 `'c'` 前缀。一个调用 `libaero2d` 库的 C 语言代码示例如下：

```
...
#include "aero2dcom.h"
...
int prlv;
char *mode,*scptn;
.
.
.
```

```

mode="C";
allocarray_(mode,strlen(mode));
aero2d_(mode,&prlv,scptn,strlen(mode),strlen(scptn));
.
.
.
deallocarray_(mode,strlen(mode));
...

```

注意调用模式参数 *mode* 应该被赋值为"C", 这是特别为C语言调用 libaero2d 库而设计的附加调用模式。如果采用合适的编译器选项编译 libaero2d 库 (例如 gfortran 编译器中的 `-fno-underscoring` 选项), 函数名末尾的下划线可以被省略。如果一个 Fortran 函数的形参是字符串, 那么字符串长度应该作为参数列表的末尾被传递给对应的C函数。除此之外, 调用顺序与采用 Fortran9x 语言的示例代码相同。在源文件包根目录中包含一个名为 libcallexpc.c 的示例C源文件, 演示了如何通过C接口模式调用 libaero2d。

编译和链接调用 libaero2d 库的 Fortran9x 源文件时, 以 gfortran 编译器为例, 需要在编译选项中加入 aero2dcom 模块所在目录的 `-I` 选项, 在链接选项中加入导入库所在目录的 `-L` 选项和 `-laero2d` 选项

```
$ gfortran ... -I(directory to aero2dcom.mod) -L(directory
to libaero2d.a or libaero2d.dll.a) -laero2d -o xxx
```

编译和链接C源文件的命令是类似的:

```
$ gcc ... -I(directory to aero2dcom.h) -L(directory to
libaero2d.a or libaero2d.dll.a) -laero2d -o xxx
```

10 附加信息

请向 sdhzhs@gmail.com 报告逻辑错误 (bugs) 或计算错误 (errors)。你也可以向 sdhzhs@gmail.com 发送邮件给出评价或建议。:)