

涡轮燃气主流 / 盘腔二次流的分区耦合仿真 I: 算法及平台实现*

Zonal Coupling Simulation for Turbine Gas Mainstream / Disk Cavity Secondary Air Flows, Part I: Algorithm and Software Platform

中国航空工业集团公司沈阳发动机设计研究所 于 霄 周晓菲 康晓恩
重庆大学动力工程学院 叶 建 武芷苙



于 霄

博士,毕业于北京航空航天大学。中航工业沈阳航空发动机设计研究所高级工程师,从事航空发动机内部流动传热分析工作,近阶段研究方向为航空发动机用换热器及整机热管理设计分析。

为满足军用和民用飞行器对其动力装置提出的持续攀升的性能指标要求,过去几十年间,航空发动机涡轮部件的进口温度不断提高^[1-2],

为实现涡轮燃气主流和盘腔二次流的分区耦合仿真,本文从分析该耦合系统所具有的主要流动特征和耦合特性入手,探讨了分区耦合算法的设计及分区耦合平台的构建。

DOI:10.16080/j.issn1671-833x.2015.12.094

高温燃气入侵转静盘腔可能带来的负面效应也日趋严重,由此导致燃气主流和盘腔二次流的耦合作用问题受到越来越多研究者的关注。在研究涡轮部件气动热力学的3种基本方法中,数值模拟手段的重要性不言而喻,而已有的研究表明^[3-4]:轮缘间隙附近冷气和燃气的掺混过程非常复杂,RANS模拟难以准确捕捉,URANS结果与试验更为接近,但仍有很大差距,LES则是准确预测该类流动的可能选项之一。虽然RANS模拟的预测精度有限,但从实践的角度看,该方法几乎仍是工程设计中可选用的唯一方案,提高RANS对燃气主流/盘腔二次流耦合问题的预测能力仍然有着重要的工程应用价

值。盘腔内部的二次流动与涡轮主流是2种性质截然不同的流动^[5],适用于每种流动的“最佳”湍流模型显然是不同的,但目前已有的耦合模拟几乎都只用了一个模型。如果可以针对这类流动发展一个分区耦合算法,在主流和二次流区域分别选择各自“最佳”的湍流模型,有可能会提高RANS方法的预测精度。

构建针对燃气主流/盘腔二次流的分区耦合平台,可以归结到复杂多物理场耦合这一范畴。随着计算机运算能力的不断增强以及人们对工程和科学问题预测精度提出的更高要求,越来越多的研究人员认识到:大多数的科学问题和工程实际问题往往处于多物理场的复杂耦合

* 中航工业创新基金项目(2012860603R),国家自然科学基金项目(51206198),中央高校基本科研业务费(CDJZR12140030)资助。

环境中,此时,单凭某一软件实现这类问题的高精度预测变得越来越困难,解决方案之一即是进行多物理场的耦合仿真。但是,目前大多数的软件产品还只能处理某一类问题或者只对该类问题具有最佳的预测精度,要实现多场耦合,其现实途径就是通过某种技术平台将多个软件产品集成到一起,以期完成复杂多物理场的耦合模拟。

工业界的迫切需求,以及该研究方向取得突破可能带来的巨大收益使得国内外研究者,尤其是国外的CAE巨头开展了大量的相关工作,并已经取得了较为丰硕的成果。归纳起来,现有的耦合平台大体上可分为3类:(1)商业化的耦合平台,如德国SCAI研究中心的MpCCI^[6]和美国Ansys公司的Workbench^[7];(2)开源的耦合平台,如法国研究机构CERFACS开发的OpenPALM^[8],美国Sandia国家实验室开发的LIME^[9]等;(3)基于Python等脚本语言的自编程耦合平台^[10-11]。

要实现燃气主流/盘腔二次流的分区耦合模拟,上述3类平台均可以使用,但考虑到编程的灵活性、平台调试和修改的方便程度以及计算的精度和速度等因素,本文采用第三种方案——即基于Python语言构建耦合平台。下文首先从分区界面的定义、界面的数据交换2个方面进行分区耦合的算法设计,而后从耦合计算的执行流程、数据交换的实现、Fluent执行的自动化和整体计算流程的控制4个方面介绍了耦合平台的构建。

分区耦合算法设计

对于本文所研究的燃气主流/盘腔二次流分区耦合仿真问题,在构建耦合平台之前,有以下关键问题需要解决:(1)燃气主流和盘腔二次流作为一个在空间连续分布、且有着复杂相互作用的流体系统,分区的界面

应如何选取;(2)确定分区界面后,在进行耦合计算时,主流区和二次流区域如何耦合在一起,这些问题所涉及的内容就是分区耦合的算法设计,它既是构建耦合平台的前提,也是核心的技术难点。本文通过分析燃气主流和盘腔二次流所具有的主要耦合特性,从分区界面的定义出发,探讨合理的界面选取方案;而后讨论当2个区域分别使用不同的湍流模型时,分区界面应交换哪些数据、如何进行交换,以此实现高效的耦合。

1 分区界面的定义

高压涡轮主流和盘腔二次流所组成的流动系统如图1所示,其复杂性主要表现在3个方面:(1)主流流道和盘腔的几何结构非常复杂,导致高质量计算网格的生成难度较大;(2)导叶叶排和动叶叶排包括叶盘之间的相对运动,对计算域的划分以及算法的选取都提出了特殊的要求;(3)多种因素综合作用导致流场的流动结构非常复杂。

针对单独的主流或盘腔流动问题,目前的仿真方案已经相对成熟。本文中只考虑稳态定常模拟时,对于单级涡轮,许多商业软件均能生成高质量的计算网格,静子和转子的界面通过掺混面方法实现流场数据的传递;对于转静盘腔,其网格大多数情况下需要手动生成,但计算中通过设

置恰当的壁面边界条件可以模拟转盘的影响。如若将两者耦合在一起,则需要解决一系列的难题。

首先,耦合计算域中间存在静子/转子的掺混面以及主流区/二次流区的分界面,这两个界面的关系是什么,其相对位置的选取如何。从掺混面的概念出发,两个界面应该分开、避免相交,也就是说,盘腔的计算域只是与转子或者静子的计算域存在交接面,而不会同时与两者相交,这意味着盘腔的网格或者与静子网格固结在一起不动,或者和转子网格固结在一起转动。从已有的结果看,不同研究者选取的方案并不一致,2种方案都有一定的合理性,其选择也可能与轮缘间隙的轴向相对位置有关。

其次,燃气主流和盘腔二次流的分区界面应如何选取?从上一段的分析可知,盘腔计算域只与静子或者转子计算域中的一个交接,这一问题就简化为:对于静子/盘腔网格或者转子/盘腔网格,二者的分界面取在何处。从主流和二次流耦合系统的几何结构看,2个大的几何空间通过一个狭小的通道——轮缘间隙连接在一起,理论上,分区界面只能设定在该区域,如若不然,界面将可能穿过存在复杂流动结构的主流或二次流区域,这将给界面数据交换带来大的困难。实际上,即便在轮缘间隙区,

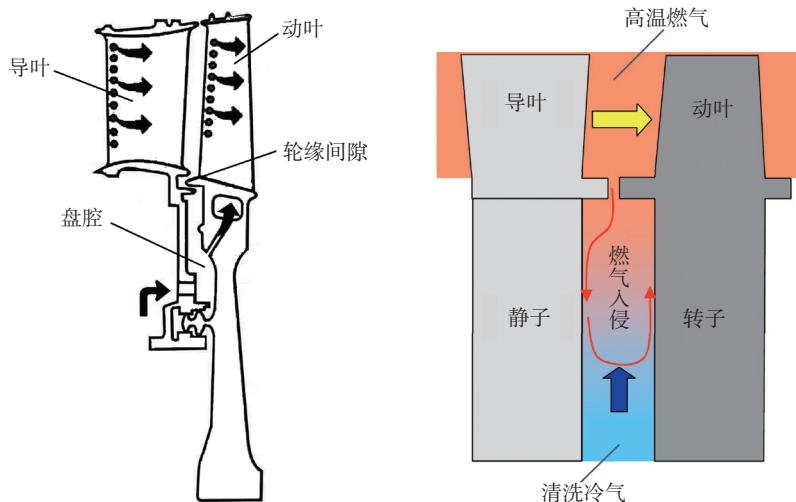


图1 高压涡轮主流和盘腔二次流系统示意图

也可能存在着较为复杂的流动,但考虑到定常模拟时,该区域的时均流动相对简单,因而将分界面取在该处是可行的;至于其具体位置,可取在轮缘间隙区中部,界面方向则和两个壁面尽量垂直。

再次,耦合系统的计算网格的生成。考虑到通过分区界面,两个区域需要频繁地进行数据交换,为保证数据传递的精度,最好在界面两侧使用完全匹配的计算网格。由此可采用如下的计算域划分策略:在主流区创建一个和盘腔轮缘间隙匹配的区域,而后确定分区界面,在其两侧生成完全匹配的计算网格。

基于上述分析,燃气主流与盘腔二次流分区界面的定义应遵循如下原则:(1)盘腔的网格或者与静子网格、或者与转子网格固结在一起,二者的交接面即是主流和二次流的分区界面,该分区界面不与静子/转子的掺混面相交。(2)主流和二次流的分区界面应位于具有最小流通面积的轮缘间隙区域,界面的朝向与两侧壁面尽量垂直,保证最小的截面积,通过该界面,流动的方向应尽量保持一致。(3)分区界面两侧的计算网格应保证完全匹配。

2 分区界面的数据交换

从已有的研究可知:在定常雷诺平均模拟中,对于涡轮燃气主流,使用 S-A 模型的综合效果最优,对于盘腔二次流,则推荐 SST $k-\omega$ 模型,据此,我们尝试在主流区选用 S-A 模型、二次流区选用 SST $k-\omega$ 模型进行分区耦合。

根据上一小节的原则定义分区界面后,就可以考虑界面间的数据交换了。理想情况下,为实现 2 个区域间的强耦合,可以将分区界面作为一个“内边界”,2 个区域使用不同的控制方程分别推进,每计算一步,沿着内边界进行一次数据的交换,交换的数据既包括 u, v, w, p, ρ, T 等基本

μ_t 。由于两侧求解的湍流输运方程不同,还应该通过界面重构湍流输运量:对于主流区而言,已知二次流边界上的 μ_t, k, ω 数值,容易得到 S-A 模型的输运量 \bar{v} ,反过来对于二次流区域而言,已知主流边界上的 μ_t 和 \bar{v} ,要得到 k 和 ω 则较为困难(见图 2)。

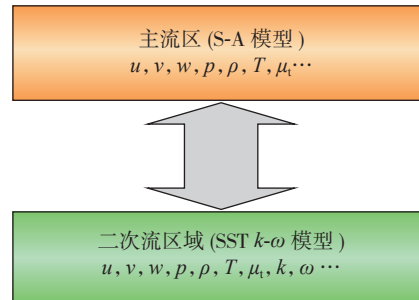


图2 主流和盘腔二次流之间的数据交换

实际的情形与此不同,由于本文选用商业软件 Fluent 作为求解器,我们对其内核缺少了解,更无法修改源代码,两个区域的耦合只能通过基于 Python 的耦合平台并结合 Fluent 提供的用户定义函数 UDF 实现,显然,这是一种“松耦合”:两个 Fluent 程序被同时执行,分别计算主流流动和二次流动,分区界面是其“外边界”而非“内边界”,分区耦合平台利用 UDF 重设边界条件实现区域间的数据交换。这样带来的问题是:我们无法像强耦合中的内边界一样在界面交换所有流场数据,而只能根据界面处的流动情况设定恰当的边界条件,并根据边条的要求提供部分流场变量,以此实现界面的数据传递。

(1) 界面处流动方向的设定。

对于在实际工况运行的主流/二次流系统而言,轮缘间隙附近的流动异常复杂,受间隙两侧当地局部流动参数的影响,周向不同位置处间隙径向流动的方向很可能是变化的:有的地方从盘腔进入主流,有的地方则从主流进入盘腔。但在 Fluent 中,分区界面作为一个确定的外部边界,必须预先给定边条的类型,或者是进口,或者是出口,其前提就是要明确界面处的流动方向。

考虑下述 3 个因素,我们将分区耦合界面的流动方向设定为从盘腔流向主流:盘腔冷气的主要作用是封住轮缘间隙,防止外部高温燃气侵入,在正常工作状态下,尽管沿不同周向位置可能有局部瞬态的燃气侵入,但从平均效果看,盘腔冷气还是往外流入主流区的;本文只对耦合系统进行定常仿真,由于未计入非定常效应,周向局部可能存在的燃气入侵现象被大大弱化,当盘腔二次流入口给定的总压相对较高时,即便存在回流,分区界面处流动的主要方向还是指向燃气主流一侧的;设定分区界面的流动方向以后,Fluent 的计算依然允许在界面处存在与设定方向相反的流动,因而回流的影响可以得到考虑。

(2) 界面处流动参数的传递。

要到达主流二次流耦合的效果,必须保证沿着分区界面流场的质量、动量和能量守恒,在 Fluent 中,分区界面作为一个“外边界”,并不能直接交换基本流场变量,而只能进行边界条件的设定,借助 UDF,我们能够实现边界上每个网格单元一一对应的边条信息传递,待流场迭代收敛后达到和交换基本流场变量类似的效果,最终保证界面处三大守恒律的满足。

确定分区耦合界面的流动方向是从盘腔流向主流后,对于盘腔流动,分区界面为计算域出口,设定压力出口边条;对于燃气主流,分区界面为计算域入口,设定速度进口边条,具体的设置如下:对于盘腔流动,分区界面处设定压力出口边条,需要主流区提供的参数包括背压、回流总温和回流的方向向量;对于主流燃气,分区界面设定为速度进口边条,需要二次流区提供的参数包括 3 个速度分量,温度和回流时的背压。尽管沿着分区界面传递的边条参数是有限的,但这些参数已经保证了计算结果的适定性,通过分区界面传递

数据,2个计算域反复迭代直至流场收敛,此时界面两侧的流场变量是连续分布的,也就保证了分区界面处三大守恒律的满足。

(3) 界面处湍流参数的传递。

通过界面处流动参数的传递,保证了沿着分区界面两侧流场变量是连续分布的,分区界面也满足质量、动量、能量三大守恒律,但这并不能保证耦合仿真结果的预测精度,其原因在于:对基于雷诺平均的仿真方法而言,湍流模型是决定预测成败的关键,仅仅在分区界面两侧保证流场变量连续是远远不够的。当流动从一个区域穿过分区界面流入另一个区域时,湍流量也会随流进入该区域,即在计算域入口(或计算域出口的回流区),除了需要流动参数的边界信息外,还应该提供湍流参数的信息。

主流区和二次流区域计算时使用的湍流模型不同,分别是S-A和SST $k-\omega$ 模型,S-A模型求解关于变量 $\bar{\nu}$ 的一个输运方程,SST $k-\omega$ 模型则求解关于湍动能 k 和湍流比耗散率 ω 的两个输运方程。无论S-A还是SST $k-\omega$ 模型,均是基于Boussinesq的涡粘性假设,显然,界面湍流参数的传递应该满足这样的原则:传递后沿界面两侧湍流粘性系数 μ_t 的分布连续。

基于上述原则,当流动从盘腔流入主流时,对于主流区(求解S-A模型)而言,分区界面的速度进口边条要求二次流区域(求解SST $k-\omega$ 模型)提供一个湍流参数,这个参数可以是修改后的湍流粘性 $\bar{\nu}$,也可以是湍流粘性比 μ_t/μ 。从两方程模型降为一方程模型,这是很容易实现的,考虑到UDF中可以直接读出界面单元的湍流粘性 μ_t 和分子粘性 μ ,这里选用湍流粘性比 μ_t/μ 作为传递的参数,具体执行时,Fluent根据边界UDF给定的 μ_t/μ 反算出湍流粘性 $\bar{\nu}$,用于输运方程的求解。

当流动从主流区回流盘腔,对于二次流区域(求解SST $k-\omega$ 模型)而言,分区界面的压力出口边条要求主流区(求解S-A模型)提供2个湍流参数,分别是湍动能 k 和湍流比耗散率 ω ,这意味着要从一方程模型重构二方程模型的参数,其实现难度很大。该问题可描述为:以保证界面处湍流粘性系数连续分布为原则,将湍流粘性系数 μ_t 或湍流粘性比 μ_t/μ 作为参数传递(事实上这也是主流区的唯一选择),实现对界面处湍动能 k 和湍流比耗散率 ω 的重构。事实上,对于 k 、 ω 、 ε 和 μ_t ,我们知道两个关系式:

$$\mu_t = \frac{\rho k}{\omega} \quad (1)$$

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \quad (2)$$

其中, ρ 是密度,经验系数 $C_\mu = 0.09$,给定 μ_t 后,3个未知量 k 、 ω 、 ε ,2个方程,无法求解,必须补充1个关系式或者给定3个未知量中的1个。

由于并不存在一般的准则,要解决这一问题,必须对界面所在位置——轮缘间隙处的流动进行具体分析,该流动具有2个主要的特点:沿着整个圆周,轮缘间隙是一个很窄的狭缝,子午面内狭缝的长度有限;轮缘间隙转盘侧相对于间隙静盘侧以很高的速度运动,该速度远远大于间隙在子午面内的流动速度。

据此,如图3所示,可以将间隙内的流动分解为2部分的矢量和,其一是周向的Couette流,一侧平面的速度很大;其二是子午面内的Poiseuille流,该流动的速度很小。通过轮缘间隙,无论进入还是离开盘

腔,大部分流体质点在子午面内运动的距离很短,但该过程中均沿周向运动了较长的距离,那么可以认为,流体质点穿过轮缘间隙的流动,近似为一个平面Couette流动。

我们将耦合界面设置在轮缘间隙中间位置,显然该位置也处于Couette流动中部,Couette流是一种相对简单的剪切流,若流动是湍流态,则湍动能的生成和耗散近似平衡,更进一步,可以假设界面处的湍流流动处于平衡态,即有如下关系:

$$P = \rho \varepsilon \quad (3)$$

其中,湍动能的生成项为:

$$P = -\rho \overline{u_i' u_j'} \frac{\partial u_j}{\partial x_i} \quad (4)$$

其中, ∂u_j 是速度张量, ∂x_i 是梯度张量。引入Boussinesq的涡粘性模型有:

$$P = \mu_t S^2 \quad (5)$$

$$S = \sqrt{2S_{ij}S_{ij}} \quad (6)$$

其中, S_{ij} 是张量。由上面的等式可以知道,耗散率 ε 计算公式为:

$$\varepsilon = \frac{P}{\rho} = \frac{\mu_t S^2}{\rho} \quad (7)$$

知道 ε 以后,利用公式(1)、(2)可以算出 k 、 ω 。显然,这些参数应该在主流区进行计算,而后再将 k 、 ω 传递到二次流区域。

分区耦合平台构建

通过上一节对分区耦合算法的研究,我们对如何定义分区界面,耦合过程中沿分区界面传递哪些数据等问题有了较为深刻的理解和认识,接下来需要考虑的问题是:如何构建分区耦合平台以成功实现燃气主

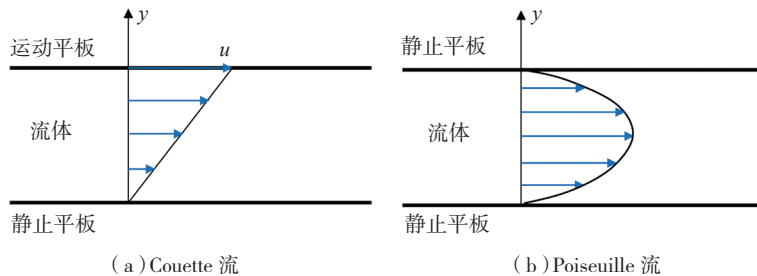


图3 Couette流和Poiseuille流示意图

流与盘腔二次流的耦合仿真。本节从分析耦合计算的执行流程入手,概括了耦合平台的2大主要工作,而后分析界面数据交换的实现方法以及Fluent程序的自动化执行,在此基础上完成了耦合平台的流程框图并介绍了平台的Python实现。

1 耦合计算的执行流程

设计分区耦合平台的重要前提之一是要搞清楚分区耦合过程的执行流程,现在设想人工执行这一流程的步骤如下(假设计算网格已准备好):

(1)运行Fluent程序A,读入燃气主流的计算网格,进行相关设置(湍流模型、边界条件等等),完成主流流场初始化;再运行一个Fluent程序B,读入盘腔二次流的计算网格,进行相关设置(湍流模型、边界条件等等),完成二次流流场初始化。

(2)设定计算步数,程序A(主流计算)进行指定步数的迭代求解;设定计算步数,程序B(二次流计算)也同样进行指定步数的迭代求解。

(3)程序A(主流计算)输出分区界面压力 p ,总温 T_i ,湍流粘性 μ_t 等数据;程序B(二次流计算)输出分区界面速度分量 u, v, w ,温度 T 以及 \bar{v} 和 μ_t 等数据。

(4)处理程序B的输出数据,更新程序A(主流计算)分区界面的边界条件;处理程序A的输出数据,更新程序B(二次流计算)分区界面的边界条件。

(5)跳转到第(2)步,(2)、(3)、(4)、(5)步顺序执行。

(6)若流场满足收敛条件,程序A、程序B分别输出计算结果,而后分别结束。

从上述步骤容易看出,要通过程序代码将耦合计算的执行流程自动化,耦合平台的主要工作包括2大部分:

第一是流程的控制,即控制2个Fluent程序的启动,初始化(读入网

格、进行相关的设置),迭代求解,收敛判定,以及计算结果的输出,程序结束等等,这一部分的重要问题是保证两个软件的同步,即是说迭代求解快的程序等待迭代求解慢的程序,实现界面数据传递的同步。

第二是完成分区界面的数据交换,两个Fluent程序分别迭代计算一定步数后,沿着分区界面分别输出对方程序需要的数据,对输出的数据进行恰当的处理,2个Fluent程序再分别读入处理后的数据,更新分区界面的边界条件。

2 数据交换的实现

通过上一小节搞清楚了分区耦合计算的执行流程后,我们首先考虑如何实现数据的交换,因为相对于整体的流程控制而言,它属于“局部”的问题,对流程依赖不多,可以“独立”的解决。

分析发现,无论对于燃气主流或者盘腔二次流动,耦合计算过程中,除分区界面处边界条件的数值需要不断更新外,其他的边界条件均无需做任何改动;而界面处边条的类型并不发生变化,只是需要更新具体的数值,并且这些数值不是单个的数据,而是界面上的一组数据分布。

利用Fluent的用户自定义函数UDF,可以轻松实现上述更新。图4给出了耦合边界数据交换的示意图,燃气主流Fluent程序A以及盘腔二次流Fluent

程序B完成一定步数的迭代后,开始交换数据,利用UDF,程序A将界面上的相关信息输出到aout.dat文件,程序B则将界面信

息输出到bout.dat文件,基于Python的主控程序读入aout.dat、bout.dat,按照要求对其中的数据进行处理,而后分别生成程序B和程序A的输入文件bin.dat、ain.dat,接下来同样利用UDF,程序A和B分别读入ain.dat和bin.dat,完成界面边界条件数据的更新。这样就实现了一次分区界面的数据交换。

3 Fluent执行的自动化

一般而言,我们在使用Fluent进行仿真工作时,通常会启动Fluent的图形用户界面(简称GUI),以便实现人与程序的实时交互。但是,在某些情况下,我们可能对交互性要求不高,只是希望程序能自动完成一系列的计算工作,Fluent提供了命令行的运行方式,合理利用它很容易实现程序执行的自动化。

该过程主要是通过Fluent的一个命令集合即进程文件(Journal File)实现的,Journal文件可以重播用户曾经进行的所有操作,其创建途径有2个:一是在用户进入图形用户界面后,系统自动记录用户的操作和命令输入,自动生成进程文件;另一个则是用户使用文本编辑器直接用Scheme语言创建进程文件。

对于本项目而言,主控程序需要“同时”反复调用2个Fluent程序,分别进行燃气主流和盘腔二次流的迭代计算,这一过程中,需要修改的

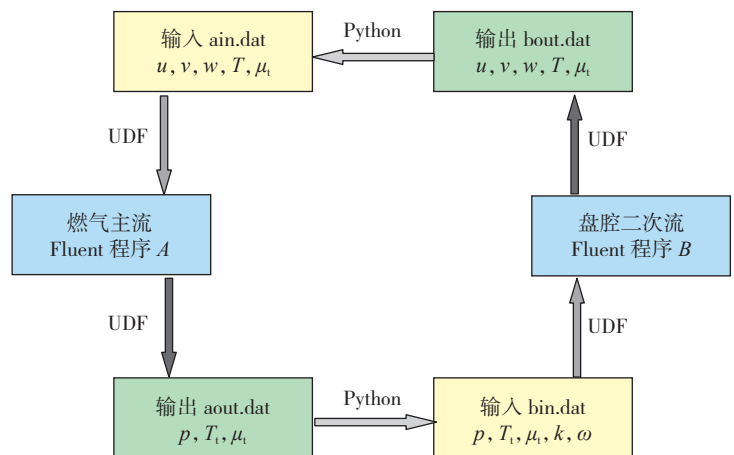


图4 耦合边界的数据交换

参数很少,没有必要也基本上无法调用 Fluent 的 GUI 进行相关设置。在主控程序执行前,我们可以利用 Scheme 语言分别创建好主流计算和二次流计算的 Journal 文件,然后主控程序分别调用这 2 个文件,进行迭代求解,如果有相关计算参数需要修改,则主控程序直接读入 Journal 文件进行修改再保存即可,这样就完全实现了 Fluent 迭代过程的自动化。

4 整体计算流程的控制及实现

通过上述 3 个小节的描述将相关细节讨论的比较清楚之后,接下来考虑整体计算流程的控制,设想我们用某种工具或语言编写耦合计算程序,参考图 5,按照计算流程该平台需要完成的主要工作如下:

(1) 计算启动,平台初始化,这一阶段需要考虑多种情况,是进行全新的计算,还是读入已有的结果进行续算? 如有需要,主控程序分别读入主流和二次流计算的 Journal 文件,进行相关部分的修改然后存盘。

(2) 主控程序调用 Fluent 分别执行两个 Journal,进行耦合计算第 n 次的迭代求解。

(3) 主控程序判断主流和二次流的第 n 次迭代求解是否均已完成,若都已完成,则分别读入两个程序的输出文件 aout.dat 和 bout.dat,按照要求对其中的数据进行处理,再分别

输出两个程序的输入文件 bin.dat 和 ain.dat。

(4) 主控程序判断计算是否收敛,若已收敛,跳到第(5)步执行,否则对两个 Journal 文件进行适当的修改,返回第(2)步执行。

(5) 耦合计算已收敛,将相关结果存盘,主控程序结束。

本文的耦合平台选用开源的面向对象的 Python 语言编写,由于该语言具有良好的可移植性,在避免使用依赖于系统特性的前提下,耦合平台几乎无需修改就可以在任何系统操作系统上运行。作为一种近似解释型语言,Python 相对于编译型语言运行速度较慢,但耦合平台要实现的只是接口和主控功能,在耦合计算过程中,绝大部分时间是“被耦合”的 Fluent 软件在进行计算,耦合程序真正执行的时间是很短的,因而它执行的快慢几乎不会影响整体的耦合计算效率。

结 论

为实现涡轮燃气主流和盘腔二次流的分区耦合仿真,本文从分析该耦合系统所具有的主要流动特征和耦合特性入手,探讨了分区耦合算法的设计及分区耦合平台的构建,主要结论如下:

(1) 燃气主流与盘腔二次流分

区界面的定义应遵循 3 个原则: 盘腔的网格与静子或转子网格固结在一起,二者的界面不与掺混面相交; 分区界面应位于轮缘间隙区域,通过该界面的流动方向尽量保持一致; 分区界面两侧的计算网格应保证完全匹配。

(2) 分区界面

处的流动方向设定为从盘腔指向主流,界面参数的传递结合 UDF 通过设定边界条件实现,既应满足界面两侧流场的质量、动量和能量守恒,也应保证湍流粘性系数的连续分布。

(3) 耦合平台的 2 大主要功能包括实现对计算流程的控制以及完成分区界面的数据交换,用 Python 语言搭建平台,结合 Fluent 的 UDF 功能和命令行执行,可以实现燃气主流和盘腔二次流分区耦合仿真提出的全部要求。

参 考 文 献

- [1] Bogard D G, Thole K A. Gas turbine film cooling. *Journal of Propulsion and Power*, 2006, 22: 249-270.
- [2] 刘大响, 金捷. 21 世纪世界航空动力技术发展趋势与展望. *中国工程科学*, 2004, 6(9): 1-8.
- [3] Valencia A G, Dixon J A, Soghe R D, et al. An investigation into numerical analysis alternatives for predicting re-Ingestion in turbine disc rim cavities. *ASME Paper GT2012-68592*, 2012.
- [4] Mahoney O, Hills T S D, Chew N J, et al. Large-Eddy Simulation of Rim Seal Ingestion. *ASME GT2010-22962*, 2010.
- [5] Chew J W, Hills N J. Computational fluid dynamics for turbomachinery internal air systems. *Phil. Trans. R. Soc. A*, 2007, 365: 2587-2611.
- [6] Wolf K. MpCCI-The General Code Coupling Interface, LS-DYNA Forum, Frankenthal, 2007.
- [7] ANSYS Inc, Workbench User's Guide, Canonsburg: AHSYS Inc, 2013.
- [8] Piacentini A, Morel T, The'venin A, et al. Open-PALM: An open source dynamic parallel coupler. *Coupled problems*, Greece, 2011.
- [9] Pawlowski R, Bartlett R, Belcourt N, et al. A Theory Manual for Multi-Physics Code Coupling in LIME//Sandia Technical Report SAND2011-2195. Albuquerque: Sandia National Laboratories, 2011.
- [10] Schluter J U, Wu X, Weide E, et al. A python approach to multi-code simulations CHIMPS//Center for Turbulence Research, Annual Research Briefs, 2005:97-110..
- [11] Wang P, Zheng Y, Zou Z, et al. A novel multi-fidelity coupled simulation method for flow systems. *Chinese Journal of Aeronautics*, 2013, 26(4): 868-875. (责编 春早)

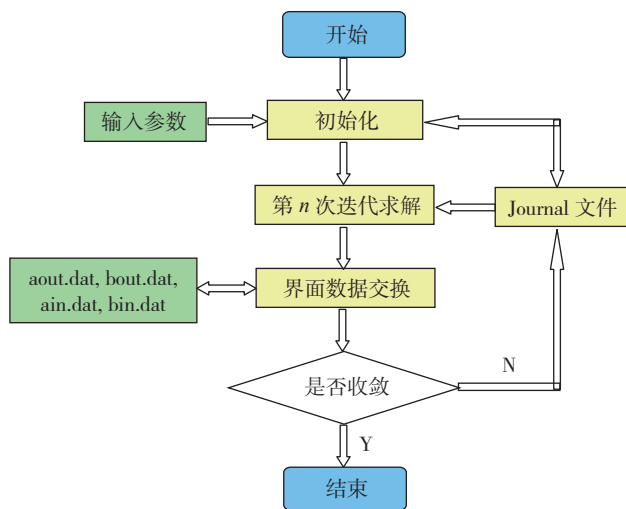


图5 耦合平台的流程框图