目 录

[**第一章 软件简介** 2](#_Toc531161679)

[**1.1 简介** 2](#_Toc531161680)

[**1.2 软件文档** 2](#_Toc531161681)

[**1.3 安装编译及运行** 2](#_Toc531161682)

[**第二章 输入** 5](#_Toc531161683)

[**2.1 参数文件** 5](#_Toc531161684)

[**2.2 控制参数** 7](#_Toc531161685)

[**第三章 输出** 16](#_Toc531161686)

[**第四章 计算案例** 25](#_Toc531161687)

[**4.1 案例1：XXXX** 25](#_Toc531161688)

[**4.2 案例2：XXXX** 25](#_Toc531161689)

**第一章 软件简介**

**1.1 简介**

CCFD V3.0是针对亚跨音速的飞行器绕流等外流流场，基于多块结构网格求解雷诺平均Navier-Stokes方程或Euler方程进行数值模拟，采用多块结构网格，包含重叠与搭接技术，可以针对复杂的真实飞行器外型进行精细模拟。

CCFDV3.0 支持cgns格式的网格文件，包含格点坐标信息以及多块连接关系和边界条件，用户可通过网格生成软件ICEM-CFD或Gridgen生成cgns网格，并在导出网格前设定边界条件。程序输出Plot3d和cgns格式的全流场解文件,以及物面信息文件和迭代残差文件，用户可使用后处理软件Tecplot等进行流场显示。

**1.2 软件文档**

CCFDV3.0 可在线获取，下载地址是。。。。。。。。。

软件包括以下部分：

1. source：包括主程序，主控制程序、主网格输入程序；
2. module：包括控制模块、网格模块、多重网格算法模块、输出模块、并行模块、求解模块、时间推进模块、湍流模块；
3. tool：包括网格类型读取程序，排序算法程序、其他自定义程序；
4. build: 包括CCFD\_MPI和CCFD\_SEQ并行和串行两个版本，用户根据自己的需求，进入不同的文件下进行编译。
5. license文件，CCFDV3.0是支持GPLV3.0版本的开源程序。

**1.3 安装编译及运行**

CCFD V3.0编译的环境包括Fortran及支持mpi编译器，可以使用GNU或者Intel的编译器，mpi版本使用的是openmpi编译器。默认使用intel编译器，可在build下各版本的makefile文件中修改，如下表所示。

|  |
| --- |
| OBJS\_DIR **=** obj/  EXE\_DIR **=** bin/  EXE **=** CCFD\_SEQ  FC **=** ifort  IDIR **=** -I/home/ccfd/cgns/Include -I/home/ccfd/metis/include  CFLAGS **=** -r8 -Wall -O2 -w -DDEBUG  LFLAGS **=**  LIBS **=** -L/home/ccfd/cgns/Lib -lcgns -L/home/ccfd/metis/lib -lmetis |

CCFD V3.0支持串行和并行两种版本，用户只需要在build文件夹下编译自己需要的运行版本即可。

CCFDV3.0使用CGNS网格类型，所以在编译CCFDV3.0之前，需要确保CGNS版本已经安装成功，同时将CGNS安装路径加入到Makefile中，保证网格的正确使用。目前使用的CGNS是串行版本，为了更大规模的并行计算，后续我们将parallel CGNS添加到CCFDV3.0后续开源版本中。

CCFDV3.0采用多层分级负载平衡策略，在编译CCFDV3.0之前，需要保证METIS（5.1版本）安装正确，将安装路径加入到Makefile中，保证在进行负载平衡分配时，METIS的API调用正常。

·编译方法为:

以串行版本为例，编译生成CCFD\_SEQ,文件在build/CCFD\_SEQ/bin/ 下可见。

#！将编译中间文件及可执行程序都清除；

cd build/CCFD\_SEQ

make clean

#！编译可执行程序；

make（或者make all）。

以并行版本为例，编译生成CCFD\_MPI,文件在build/CCFD\_MPI/bin/ 下可见。

#！将编译中间文件及可执行程序都清除；

cd build/CCFD\_MPI

make clean

#！编译可执行程序；

make（或者make all）。

在安装成功后，可执行程序在相应的版本文件夹下的/bin下。可以直接copy可执行程度到计算目录下，也可以使用链接方式。

串行执行CCFDV3.0程序：

./CCFD\_SEQ

并行支持CCFDV3.0程序：(输入文件名为input.dat)

mpirun –np <noprocessors> ./CCFD\_MPI

noprocessors表示并行核数。

**第二章 输入**

**2.1 参数文件**

CCFD V3.0输入控制文件名为input.dat，范例如下：

表2.1 文件input.dat示例

|  |
| --- |
| ###########################################################################  # CCFDV3.0 Versions  # Supercomputing Center, CNIC, Chinese Academy of Sciences  # Have any questions you can contact with us!  # support\_ccfd@sccas.cn  # Date: 2017.7.22  # File Version 3.0.1 July 22th, 2017  ###########################################################################  \*Grid File\*(file name)  f6.cgns  \*mesh type\*(grids type)  CGNS  \*grid dim\*(=3:three dimension；=2:two dimension)  3  \*steady/unsteady\* (=0:steady;=1:unsteady)  0  \*ialpha\*( =0: x-z plane with z-up;=1: x-y plane with y-up)  1  \*\*\*\*\*\*\*\*Inflow condition \*\*\*\*\*\*\*\*  \*Mach number\* (non-dimensional, based on the free-stream values)  0.7510  \*Angle of attack\*(degrees, only for compressible flows)  1.0030  \*Side slip angle\*(degrees, only for compressible flows)  0.0000  \* Reynolds number\*( free-stream Reynolds number per unit grid length (millions) re/L)  0.0212465  \*Height\*(KM)  -1.0  \*Free-stream temperature\*(288.15 K by default)  460.0008  \*\*\*\*\*\*\*\*Geometric\*\*\*\*\*\*\*\*  \*Sref \*( Reference area for force coefficients)  72700  \*Cref \*( Reference length)  141.2  \*Bref\*( reference span used to compute non-dimensional moments)  587.7  \*Moment center \*( Reference origin for moment computation)  157.9 0.0000 0.0000  \*\*\*\*\*\*\*\*Time\*\*\*\*\*\*\*\*  \*Time advancing\*( <0:steady computing；>0，dual-time schemes unsteady methods, this version just can used steady computing ,the values keep <0.0)  -0.50  \*CFL\*(CFL number)  0.50  \*\*\*\*\*\*\*\*Numerical\*\*\*\*\*\*\*\*  \*Limiter\*(=0:no limiter;=1: van albada;=2:min-mod limiter;=3: SPEKREIJSE VANKAT)  1  \*xkap\*(value<-1.0:no limiter; when xkap≠1/3,used the min-mod, xkap=1/3 can used van albada or SPEKREIJSE VANKAT)  0.3333  \*epsa\_r\*( correction the eigenvalue,0.01<epsa\_r<0.4,when <0.0 no corrections)  -0.01  \*turbulent\*(=0:euler; =1: n-s; =2:B-L model; =3:SA model)  3  \*\*\*\*\*\*\*\*Restart\*\*\*\*\*\*\*\*  \*restart flags\*(=0:no restart;=1: output the restart files for restart computing)  0  \*nrestart\*(iterations interval for each output the restart files)  9999  \*\*\*\*\*\*\*\*Multi-grid methods\*\*\*\*\*\*\*\*  \*Multi-grid Flag\*(=0:not used the multi-gird methods; ≠0 used the multi-grid method)  1  \*multi-grid level\*(level of multi-grids method ,when the value equal 1,mean not used the MG methods)  3  \*VCYCLE\*(=0:used the w-cycle schemes; =1:used the v-cycle)  1  \*WCYCLE\*(=0:used the v-cycle schemes;=1;used the w-cycle)  0  \*implicit residual smoother\*(=0:not used the IRS methods;=1:used the IRS methods)  0  \*SMOOTH\_R\_COE\*(IRS schemes coefficient)  0.30  \* implicit correction smoother\* (=0:not used the ICS methods;=1:used the ICS methods)  1  \*SMOOTH\_C\_COE\*(ICS schemes coefficient)  0.30  \*\*\*\*\*\*\*\*Mesh Sequencing methods\*\*\*\*\*\*\*\*  \*Mesh Sequencing Flag\*(=0:not used the mesh Sequencing methods;≠0:used the mesh Sequencing and must set the iterations for each level)  1  \*Mesh Sequencing Level\*(=1:just used the multi-grid methods;>1:the level of mesh Sequencing methods)  3  \*iterations\*(iterations for each level)  1000  2000  3000  ########################################################################### |

下面分别对参数控制文件中的设置进行说明。

**2.2 控制参数**

|  |
| --- |
| \*Grid File\*(file name)  f6.cgns  \*mesh type\*(grids type)  CGNS  \*grid dim\*（=3:three dimension；=2:two dimension）  3  \*steady/unsteady\* (=0:steady;=1:unsteady)  0  \*ialpha\*( =0: x-z plane with z-up;=1: x-y plane with y-up)  1 |

1. Grid File：输入的网格文件名字；
2. mesh type：网格类型说明，支持串行CGNS(HDF2.4版本)网格类型，该开源版本只支持cgns网格类型，可选参数仅为“cgns”；
3. grid dimension：网格维度，支持二、三维计算；可选参数2/3，分别对应二维三维计算；
4. steady/unsteady：可选参数为0/1，0表示定常计算，1表示非定常计算，目前版本仅支持定常计算，可选参数仅为“0”；
5. ialpha：坐标方向，可选参数为0/1，0表示攻角在XY平面定义，1表示攻角在XZ平面定义；

|  |
| --- |
| \*\*\*\*\*\*\*\*Inflow condition \*\*\*\*\*\*\*\*  \*Mach number\* (non-dimensional, based on the free-stream values)  0.7510  \*Angle of attack\*(degrees, only for compressible flows)  1.0030  \*Side slip angle\*(degrees, only for compressible flows)  0.0000  \* Reynolds number\*( free-stream Reynolds number per unit grid length (millions) re/L)  0.0212465  \*Height\*(KM)  -1.0  \*Free-stream temperature\*(288.15 K by default)  256.00 |

1. Mach number：马赫数；
2. Angle of attack：攻角,单位°；
3. Side slip angle：侧滑角,单位°；
4. Reynolds number：单位雷诺数，输入值为实际雷诺数除以10^6；
5. Height：高度(km)，取值为“-1.0”，表示不采用高度表，来流参数在下文直接指定,该值不能修改；
6. Free-stream temperature:自由来流温度；

|  |
| --- |
| \*\*\*\*\*\*\*\*Geometric\*\*\*\*\*\*\*\*  \*Sref \*( Reference area for force coefficients)  72700  \*Cref \*( Reference length)  141.2  \*Bref\*( reference span used to compute non-dimensional moments)  587.7  \*Moment center \*( Reference origin for moment computation)  157.9 0.0000 0.0000 |

以下设置，均要求模型采用标准单位。

1. Sref：参考面积，单位m^2；
2. Cref：参考长度，单位m^2；
3. Bref：展长，单位m^2；
4. Moment center：力矩作用点，包括xyz三个坐标方向上的取值，单位m^2；

|  |
| --- |
| \*\*\*\*\*\*\*\*Intergration\*\*\*\*\*\*\*\*  \*Time advancing\*( <0:steady computing；>0，dual-time schemes unsteady methods, this version just can used steady computing ,the values keep <0.0)  -0.50  \*CFL\*(CFL number)  0.50 |

1. Time advancing：时间推进参数；

dt<0为单时间步计算，不使用子迭代计算，定常计算；>0，双时间步计算，使用双时间步计算，非定常计算；该开源版本只支持定常计算，该值仅为“-0.5”；

1. CFL：CFL number。

|  |
| --- |
| \*\*\*\*\*\*\*\*Numerical\*\*\*\*\*\*\*\*  \*Limiter\*(=0:no limiter;=1: van albada;=2:min-mod limiter;=3: SPEKREIJSE VANKAT)  1  \*xkap\*(value<-1.0:no limiter; when xkap≠1/3,used the min-mod, xkap=1/3 can used van albada or SPEKREIJSE VANKAT)  0.3333  \*epsa\_r\*( correction the eigenvalue,0.01<epsa\_r<0.4,when <0.0 no corrections)  -0.01  \*turbulent\*(=0:euler; =1: n-s; =2:B-L model; =3:SA model)  3 |

1. Limiter：限制器。

limiter= 0 无限制器；

limiter= 1 van-albada限制器；

limiter= 2 min-mod限制器，一般设置xkap≠1/3；

limiter= 3 spekreijse vankat限制器，一般配合xkap=1/3使用。

如果不使用限制器，需要将xkap系数设置为<-1.0，>-1.0时，加上limiter配合使用才生效。

1. xkap：限制器参数。如果xkap<-1.0则不使用限制器；
2. epsa\_r：熵修正参数：范围0.01 < epsa\_r < 0.4；

熵修正方法一般都是在马赫数>0.7情况下使用，该方法主要是解决通量jacobi矩阵的特征值很小产生非物理解的情况，但是熵修正方法额外引入耗散，影响了计算精度和稳定性，所以使用与否需要用户根据具体求解的问题来衡量。使用熵修正方法需要用户在input.dat文件中编辑，取值范围为0.01 < epsa\_r < 0.4，当epsa\_r<0.0则不调用熵修正方法。

1. Turbulent：湍流模型；

=0 欧拉模型；

=1 层流模型；

=2 baldwin lomax湍流模型；

=3 SA湍流模型；

=4 k-w SST湍流模型，目前开源版本不支持该功能。

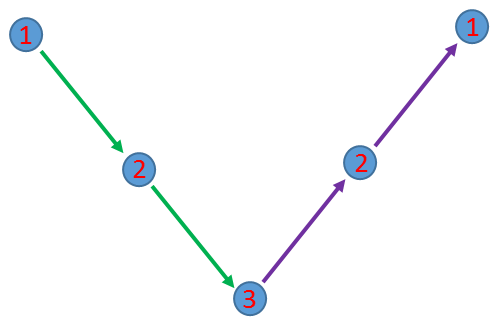
详细的各方程在附录中可见。B-L模型具有很大的局限性，仅在j方向使用湍流计算，所以不建议用户使用该模型。

|  |
| --- |
| \*\*\*\*\*\*\*\*Restart\*\*\*\*\*\*\*\*  \*restart flags\*(=0:no restart;=1: output the restart files for restart computing)  0  \*nrestart\*(iterations interval for each output the restart files)  9999 |

1. Restart：续算开关；
2. nrestart：更新续算文件的输出间隔；

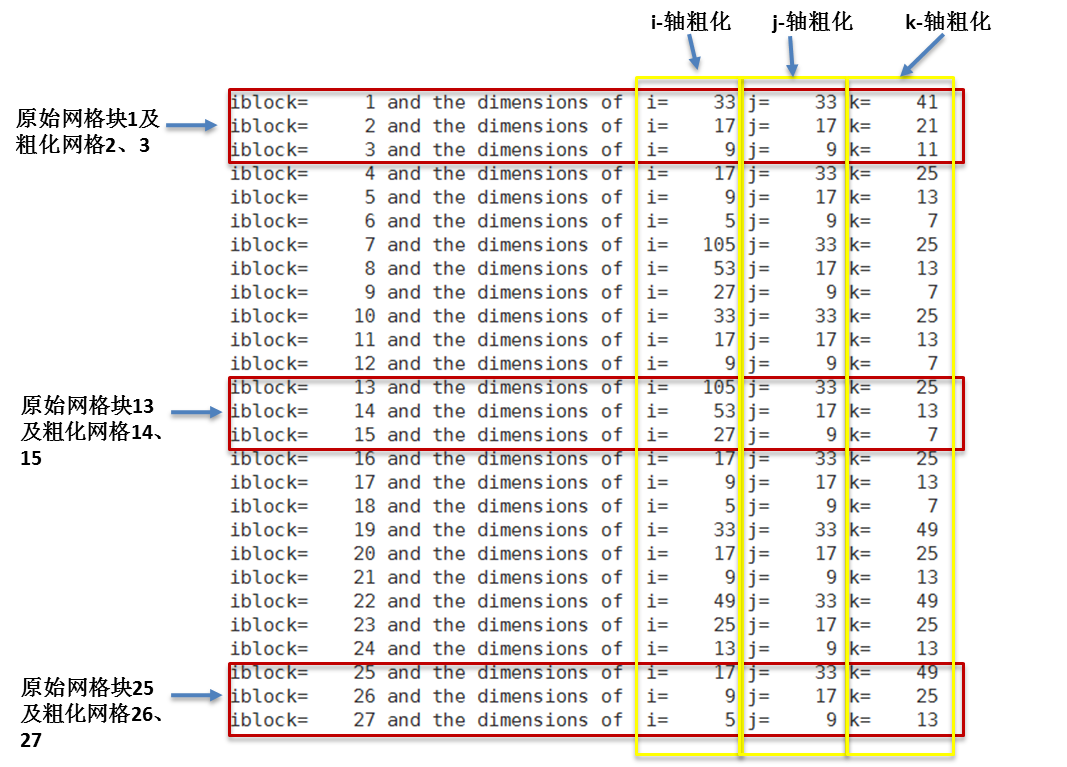
|  |
| --- |
| \*\*\*\*\*\*\*\*Multi-grid methods\*\*\*\*\*\*\*\*  \*Multi-grid Flag\*(=0:not used the multi-gird methods; ≠0 used the multi-grid method)  1  \*Multi-grid level\*(level of multi-grids method ,when the value equal 1,mean not used the MG methods)  3  \*VCYCLE\*(=0:used the w-cycle schemes; =1:used the v-cycle)  1  \*WCYCLE\*(=0:used the v-cycle schemes;=1;used the w-cycle)  0  \*implicit residual smoother\*(=0:not used the IRS methods;=1:used the IRS methods)  0  \*SMOOTH\_R\_COE\*(IRS schemes coefficient)  0.30  \* implicit correction smoother\* (=0:not used the ICS methods;=1:used the ICS methods)  1  \*SMOOTH\_C\_COE\*(ICS schemes coefficient)  0.30 |

1. Multi-grid Flag：多重网格使用开关=1使用，=0不使用。是否使用multi-grid methods需要根据各网格块和网格边界面的大小来决定，因为多重网格方法需要对原始网格进行粗化操作，比如网格块三维大小为(10,20,30), 如果使用两层多重网格，则算法错误，无法形成粗网格，后面会详细解释网格大小和多重网格层数的关系。
2. Multi-grid level：多重网格方法的计算层数。一般都是>1,如果等于1则弱化成使用非多重网格方法。



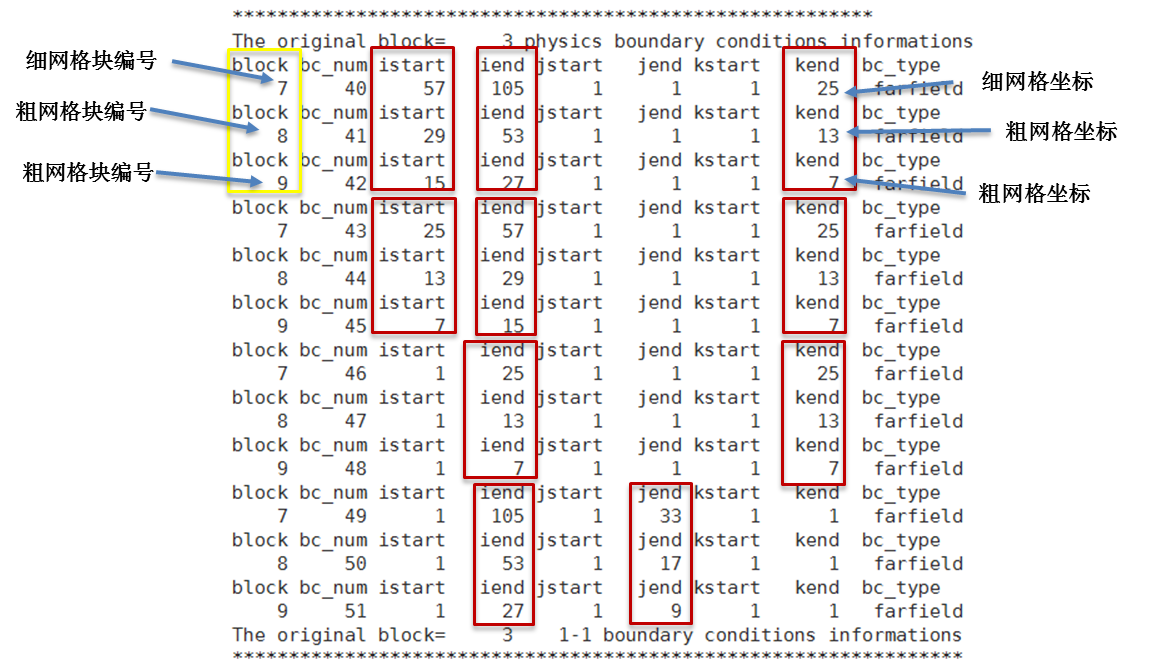
三层多重网格

我们使用给一个具体的算例的细网格和粗网格坐标索引值来说明，3层的多重网格方法中，为了计算的方便性和正确性，我们将原始网格块编号为n，则它的粗化网格块则为n+1，再粗化网格则为n+2编号。比如下图中，第1块原始块和它的粗化网格系列为2、3网格块，i-轴为33->17->9粗化，j-轴为33->17->9粗化，41->21->11粗化过程。同时可以知道，使用多重网格方法后的网格块数目变化为blocks=原始网格块\*多重网格层数



3-层多重网格

上述为网格块从原始网格到粗网格块的粗化过程，为了保证整个计算可以进行，需要将边界也同理进行粗化，下面就是一个3-层多重网格方法的边界的粗化后坐标索引值变化情况。需要注意的是我们将边界粗化，是不改变网格边界的类型和所在网格块的位置的，我们只将其进行坐标维度上的粗化操作。

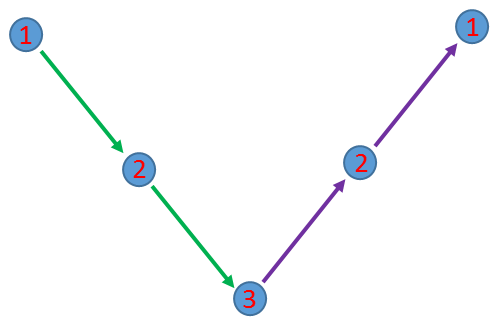


3-层细网格边界到粗网格边界

多重网格方法的使用，需要网格配合，即网格大小和边界segement的大小限制了多重网格可以使用的层数，一般规律为坐标大小应该为2\*level+1，其中level为多重网格方法的层数。

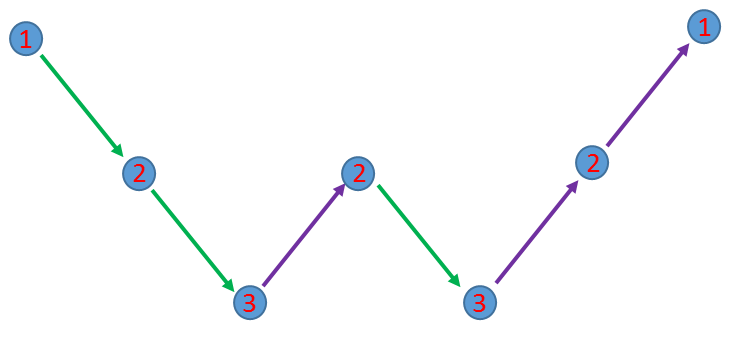
1. VCYCLE：Multigrid方法的V-cycle迭代方法开关，=1使用“V”-cycle，=0则不使用。

多重网格的向下限制和向上提升插值有多种方式，CCFDV3.0只采用了“V”和“W”两种方法，两者方法一定要选择一种，如果设置错误，则默认使用“V”-cycle模式。



“V”-cycle

1. WCYCLE：multi-grid方法的W-迭代方法开关，=1使用“W”-cycle，=0则不使用。



“W”-cycle

1. implicit residual smoother：隐式残差光顺方法使用开关=1使用，=0不使用。
2. implicit residual smoother schemes coefficient：对残差使用残差光顺方法的系数值。
3. implicit correction smoother：隐式修正解光顺方法使用开关=1，=0不使用。
4. implicit residual smoother schemes coefficient修正解使用残差光顺方法的系数值。

一般计算情况下，不建议开启残差光顺和修正光顺功能，请用户根据实际情况来决定使用该功能。

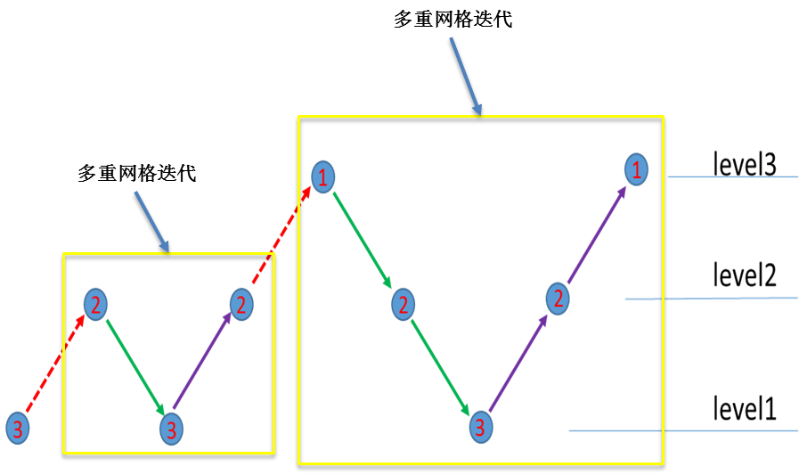
|  |
| --- |
| \*\*\*\*\*\*\*\*Mesh Sequencing methods\*\*\*\*\*\*\*\*  \*Mesh Sequencing Flag\*(=0:not used the mesh Sequencing methods;≠0:used the mesh Sequencing and must set the iterations for each level)  1  \*Mesh Sequencing Level\*(=1:just used the multi-grid methods;>1:the level of mesh Sequencing methods)  3  \*iterations\*(iterations for each level)  1000  2000  3000 |

1. Mesh Sequencing Flag： =1使用mesh Sequencing方法，=0不使用mesh Sequencing方法；

目前我们已经提到了多重网格和Mesh Sequencing网格，这两者的区别仅仅是Mesh Sequencing方法是基于多重网格方法，Mesh Sequencing进一步的发挥了多重网格的优化，降低了计算量，加速收敛。

1. Mesh Sequencing Level：Mesh Sequencing网格算法层数；

Mesh Sequencing是基于多重网格方法的，Mesh Sequencing从最粗网格开始进行迭代求解，完成该层的迭代后，向上插值提升，得到的值作为更细层的初始值进行迭代求解，完成此层的迭代后，也一样的操作向上提升。其中1、2、3是网格块的编号，即1为原始细网格，2是1网格块的粗化网格块，3是2网格块的粗化网格块。

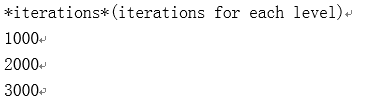


三层Mesh Sequence示意图

如果mesh Sequencing的层数为1，则mesh Sequencing方法弱化成单纯的Multi-grid方法。在使用mesh Sequencing经验上来说，定常计算建议使用，非定常计算则不建议使用。多重网格和mesh Sequencing方法是配合使用的，mesh Sequencing是基于多重网格方法的。

1. iterations：迭代次数。

迭代次数大于0的整数，在使用mesh Sequencing时，还需要为每层的迭代设置迭代次数，按照我们使用的经验来说，粗网格层收敛比较快，所以一般越粗层的迭代次数也越少，细层迭代需要更多，如下图，三层Mesh Sequencing网格方法每层迭代次数设置。需要说明的是，Mesh Sequencing使用情况下，多重网格也必须使用。

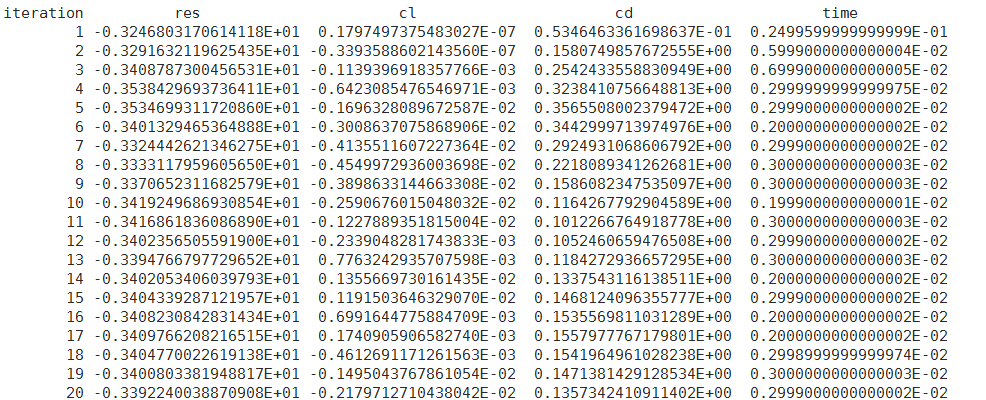


3-层Mesh Sequence网格迭代次数设置

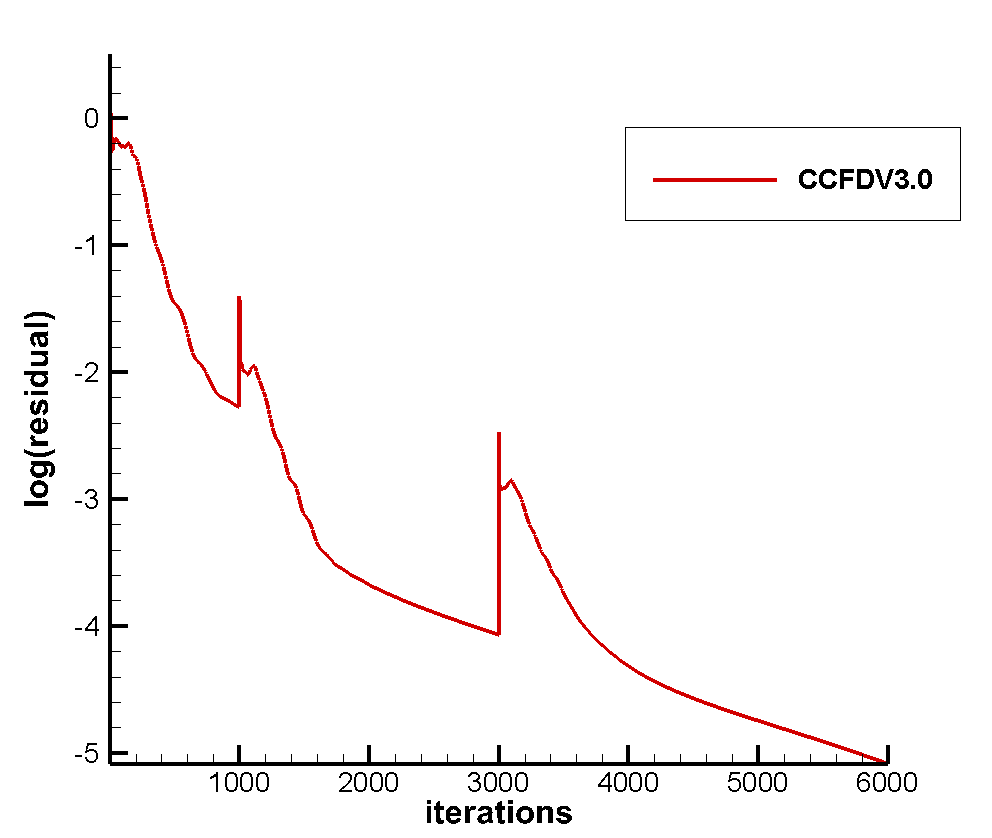
**第三章 输出**

各类输出文件解释如下：

a) 残差文件：ccfd.resid ,实时输出计算过程中残差、升阻力系数的收敛情况，并记录每一迭代步所用时间。



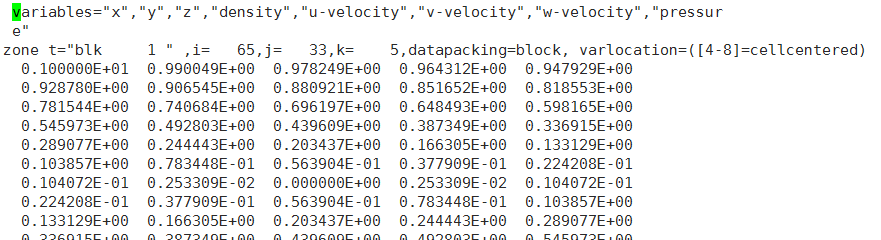
在ccfd.resid文件中，输出信息为迭代次数、残差、升力系数、阻力系数、单步计算时间。该文件可以使用Tecplot直接打开，显示残差、升阻力系数在整个迭代过程的收敛情况。



残差收敛

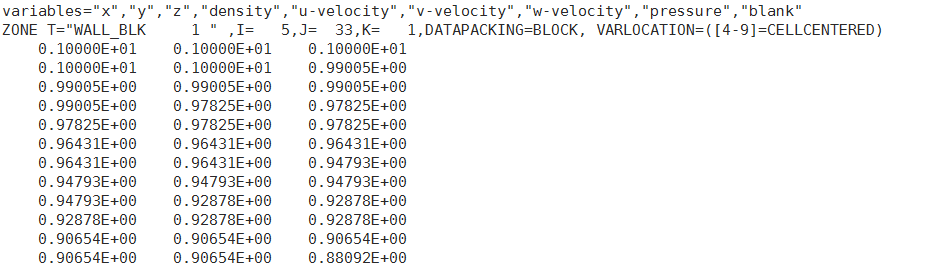
b) CGNS格式流场解文件：\*.cgns,将计算的流场解信息写入初始的cgns网格文件。该网格文件为cgns标准文件。

c) 全局流场信息文件ccfd\_global.plt格式流场解文件：记录全流场网格每个格点的坐标以及流场解信息。



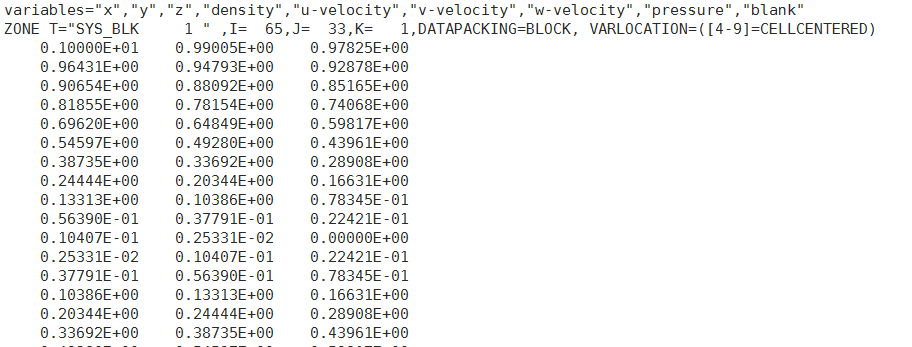
该文件保存了计算结束后，整个流场的信息，包括三个方向的速度矢量、压力、密度五个变量。文件是按照Tecplot格式输出，坐标以网格格点输出，五个变量以网格格心输出，可以直接使用Tecplot打开，查看全局流场的信息。

d) 固壁面信息文件:ccfd\_wall.plt,记录固壁面边界第一层和第二层网格的流场解信息。



该文件只保存了固壁边界信息，包括密度、三个方向的速度矢量、压力、blank值六个变量。文件是按照Tecplot格式输出，坐标以网格格点输出，六个变量以网格格心输出，可以直接使用Tecplot打开，查看固壁边界的信息。

e) 对称面信息文件:ccfd\_sys.plt,记录对称边界第一层和第二层网格的流场解信息。



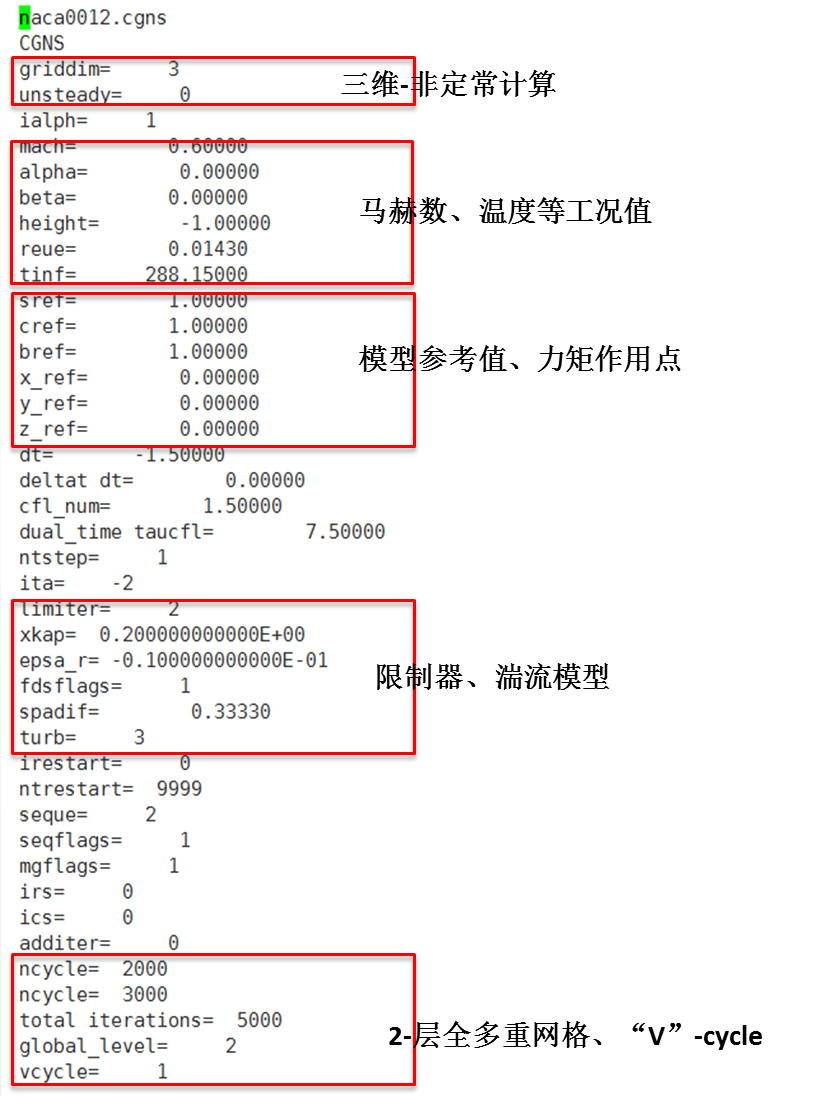
为了后处理画图方便，我们将固壁和对称物理边界分开输出，该文件只保存了对称边界信息，包括密度、三个方向的速度矢量、压力、blank值六个变量。文件是按照Tecplot格式输出，坐标以网格格点输出，六个变量以网格格心输出，可以直接使用Tecplot打开，查看对称边界的信息。

f) 续算日志文件：ccfd.restart,CCFDV3.0中断重启的保存文件，方便下次直接重启程序继续计算。续算文件信息和global文件信息输出格式和内容一致，都是保存了续点计算出的流场全局信息。

g) 求解器输出信息：ccfd.out,在计算过程中实时输出，该文件记录了CCFDV3.0迭代求解过程的详尽信息。

下面分别介绍ccfd.out文件中输出信息的含义。

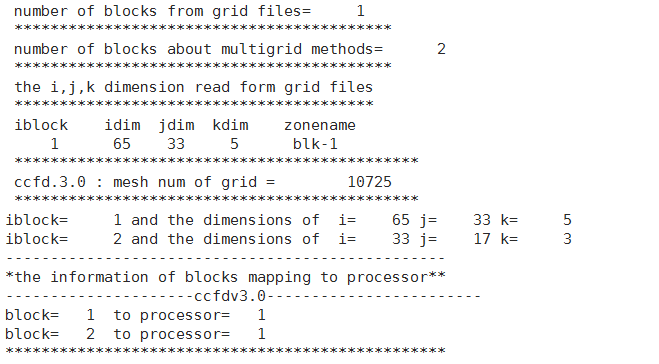
**控制参数信息：**



上述信息为控制文件信息输出，基本来自于输入控制文件input.dat。用户可以检查上述信息，确认求解器、限制器、迭代次数、cfl数、Multi-grid层数等信息的正确性。

**网格块及分配信息**

网格块的大小由三个方向坐标轴索引值idim、jdim、kdim决定的，i、j、k三个在坐标轴上都有方向，有起始索引值（i=1、j=1、k=1）和结束索引值（i=idim、j=jdim、k=kdim），每个值都是一个网格格点坐标。我们将一个网格块和六个面分别i0(i=1) 和 idim、j0(j=1) 和 jdim、k0(k=1) 和 kdim来表示。



上述信息分别表示：

number of blocks from grid files=1，cgns网格文件输入的网格块数，即原始网格块数。

number of blocks about multigrid methods=2，在使用2-层Multigrid方法下的网格块数，这个值应该等于Multigrid层数\*原始网格块数，注意，此处的网格块已经包含了原始网格块和重写生成的粗网格块。

|  |
| --- |
| iblock idim jdim kdim zonename  1 65 33 5 blk-1 |

iblock表示的是网格块编号，比如1为第一个网格块，idim、jdim、kdim分别表示三个方向I,j,k的维度大小，zonename为cgns网格文件中的编号，这个信息可以保证在cgns专门的显示软件中查找到相应的网格块。

ccfd.3.0 : mesh num of grid = 10725

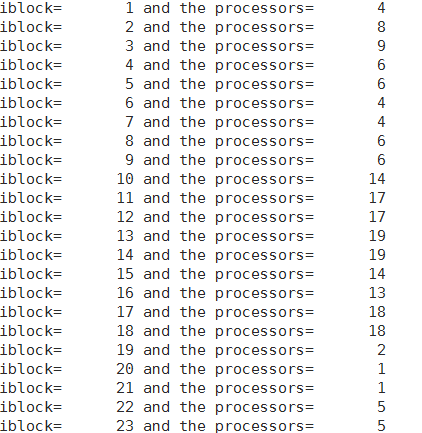
表示读入的网格块的大小，一般为idim\*jdim\*kdim的值，此处不记录多重网格的粗网格大小，因为粗网格是计算时生成的网格。

|  |
| --- |
| iblock= 1 and the dimensions of i= 65 j= 33 k= 5  iblock= 2 and the dimensions of i= 33 j= 17 k= 3 |

表示原始细网格和粗网格的三维大小，比如iblock=2即为粗网格，它的维度大小应该为原始网格iblock=1的idim/2+1，jdim/2+1，kdim/2+1，这是2-层多重网格情况下的，通量，如果使用3-层多重网格，则进一步将iblock=2进行粗化操作，当然每次粗化都要保证粗化后的网格块为三维网格，所以此处的算例只进行了2-层多重网格计算。

|  |
| --- |
| \*the information of blocks mapping to processor\*\*  ---------------------ccfdv3.0------------------------  block= 1 to processor= 1  block= 2 to processor= 1 |

表示每一个网格块对应的处理器号，CCFDV3.0使用主从模式，所以主处理器上只进行通信和其他的逻辑操作，不进行计算处理，主处理上不分配网格块。多重网格算法增加了通信，会削弱并行计算能力，为了将这种影响降至最低，除了从算法层面进行优化，在网格块分配上，我们保证每一个原始网格和它的粗化网格都分配到一个处理上进行计算。

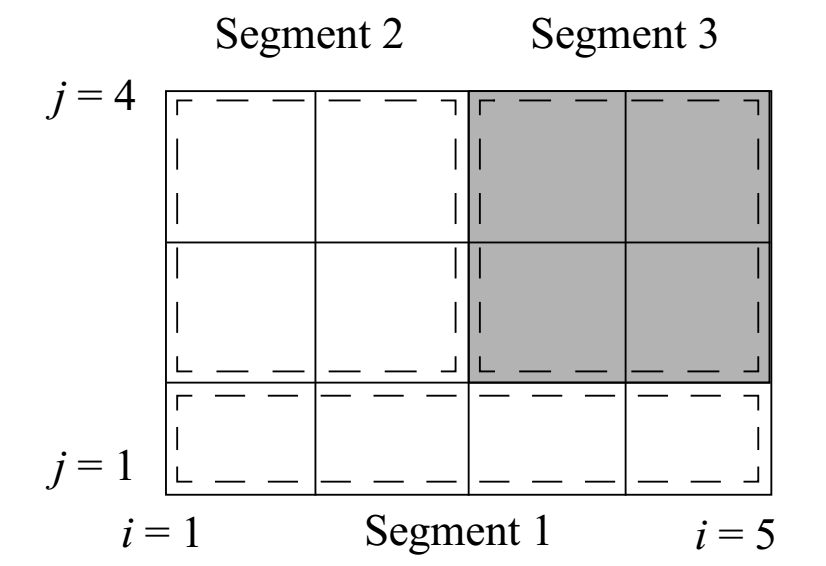


如上图所示，多块结构网格的网格块分配情况，此处使用的负载平衡算法为k-way图剖分方法负载平衡策略。

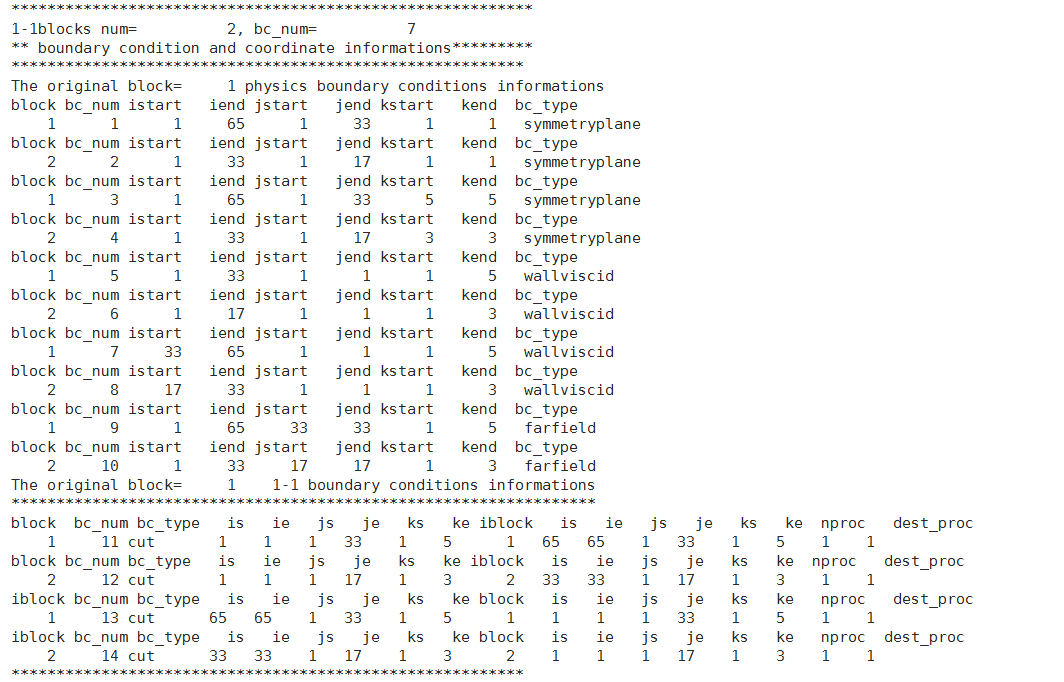
**边界信息**

CCFDV3.0开源版本支持的边界类型包括远场、对称、粘性壁面、无粘壁面、1-1邻接面，在程序中分别使用farfield 、symmetryplane、wallviscid、wallinviscid、cut边界类型名来表示。

每个网格块有6个面，每个面又可以分成多个segement段，每个segement就是一个边界，包括了所在面的坐标号，另外两个方向的起始和结束号。



ccfd.out输出的边界条件信息，包括这些segement大小，位置以及所在的网格块信息完整，保证计算的正确性。



上图所示信息分别表示为，

|  |
| --- |
| 1-1blocks num = 2, bc\_num = 7  \*\* boundary condition and coordinate informations\*\*\*\*\*\*\*\*\* |

表示两种边界个数，此处的边界数只为原始网格的边界数，不包括多重网格算法粗化后的边界数，其中1-1blocks num=2表示该网格块的cut邻接面为2个segement，bc\_num为总的网格边界处，也就是说物理边界（对称面、固面、远场）为5个。在多重网格算法中，总网格边界数应该为14.

**物理边界信息：**

|  |
| --- |
| block bc\_num istart iend jstart jend kstart kend bc\_type  1 1 1 65 1 33 1 1 symmetryplane  block bc\_num istart iend jstart jend kstart kend bc\_type  2 2 1 33 1 17 1 1 symmetryplane |

上述信息中，block表示为网格块编号，bc\_num为网格边界编号，istart、iend为i方向的起始和结束网格点编号，jstart、jend为j方向的起始和结束编号，kstart、kend为k方向的起始和结束编号，bc\_type为网格类型说明。比如上述信息中，属于第1个网格块的对称边界，该边界为k0面，大小为1\*65和1\*33的对称边界面。网格块2位网格1的粗化网格块，同理它的面粗化后为k0面的，大小为1\*33和1\*17的对称面。

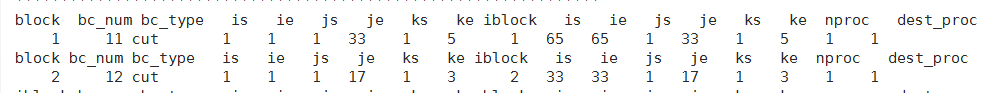
|  |
| --- |
| block bc\_num istart iend jstart jend kstart kend bc\_type  1 5 1 33 1 1 1 5 wallviscid  block bc\_num istart iend jstart jend kstart kend bc\_type  2 6 1 17 1 1 1 3 wallviscid |

上述信息中，block表示为网格块编号，bc\_num为网格边界编号，istart、iend为i方向的起始和结束网格点编号，jstart、jend为j方向的起始和结束编号，kstart、kend为k方向的起始和结束编号，bc\_type为网格类型说明。比如上述信息中，属于第1个网格块的对称边界，边界号为5，该边界为j0面，大小为1\*33和1\*5的对称边界面。网格块2位网格1的粗化网格块，边界号为6，同理它的面粗化后为j0面的，大小为1\*17和1\*3的粘性壁面。

|  |
| --- |
| block bc\_num istart iend jstart jend kstart kend bc\_type  1 9 1 65 33 33 1 5 farfield  block bc\_num istart iend jstart jend kstart kend bc\_type  2 10 1 33 17 17 1 3 farfield |

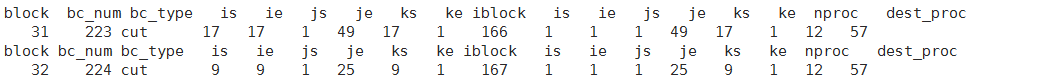
上述信息中，block表示为网格块编号，bc\_num为网格边界编号，istart、iend为i方向的起始和结束网格点编号，jstart、jend为j方向的起始和结束编号，kstart、kend为k方向的起始和结束编号，bc\_type为网格类型说明。比如上述信息中，属于第1个网格块的对称边界，边界号为9，该边界为jdim面，大小为1\*65和1\*5的对称边界面。网格块2位网格1的粗化网格块，边界号为10，同理它的面粗化后为jdim面的，大小为1\*33和1\*3的粘远场边界。

**cut边界（邻接面）信息：**

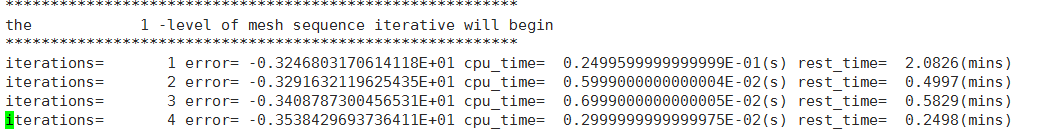


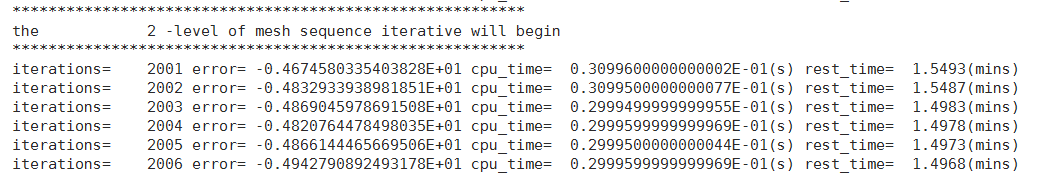
上述信息中，block为网格块编号，bc\_num为网格边界编号，is、ie、js、je、ks、ke是属于该块上的segement邻接面的三个方向的起始和结束编号。iblock表示对应邻接关系的网格块编好，这个邻接关系可以是对应不同的网格块，也可以是对应到自身的邻接面，is、ie、js、je、ks、ke表示对应邻接面三个方向上的起始和结束编号，nproc和dest\_proc为两个有邻接关系的网格块所在的处理器编号，邻接关系的网格块存在不同的处理器上有通信的需要。比如block=1，边界编号为11，cut面在i0面上，大小为1\*33和1\*5，邻接块为网格块1，cut面在idim面上，大小为1\*33和1\*5，处理器所在位置都为1。block=2，边界编号为12，为11边界的粗化边界，cut面在i0面上，大小为1\*17和1\*3，邻接块为网格块1，cut面在idim面上，大小为1\*17和1\*3，处理器所在位置都为1。cut面需要保证两边的面大小一致。

另外一个示例如下图所示。



**迭代求解信息**





上图所示是程序进入迭代后的迭代信息，包括迭代步数，误差、单步计算时间、剩余计算时间。如果使用Mesh Sequence网格，则每层的输入开始都有层数说明信息，在2001步开始，如进入第二层Mesh Sequence网格迭代。

**第四章 计算案例**

**4.1 案例1：XXXX**

。

**4.2 案例2：XXXX**

。