OpenCFD-SCU控制文件与参数说明

> Edited by Dang Guanlin, 2022-01, dangguanlin@imech.ac.cn

OpenCFD-SCU参数定义模块

- OpenCFD-SCU中所有可输入的参数都定义在变量类型为configItem的结构体数组configList中。
- 每个控制参数即为configList数组中的一项。

> configItem类型定义如下：

```
typedef struct configItem_
{
    char name[1000];   //变量名
    char value[1000];  //变量值
} configItem;
```

> OpenCFD-SCU目前主计算部分可输入的控制参数为28个：

```
configItem configList[27] = {
    {"GRID_3D", 0},                 //x、y、z三个方向的网格数
    {"PARALLEL_3D", 0},             //x、y、z三个方向并行分割数目
    {"LAP", 0},                     //重叠区宽度
    {"MSG_BLOCK_SIZE", 0},          //通讯方式
    {"STREAM", 0},                  //计算通讯重叠
    {"TEST", 0},                    //是否开启节点测试
    {"IPERIODIC", 0},               //周期性边界条件
    {"JAC_BOUND", 0},               //计算Jacobian时的边界格式
    {"DIF_BOUND", 0},               //差分计算时的边界格式
    {"NON_REFLETION", 0},           //无反射边界条件
    {"SCHEME_INVIS", 0},            //无粘项计算格式
    {"SCHEME_VIS", 0},              //粘性项计算格式
    {"RE", 0},                      //雷诺数
    {"AMA", 0},                     //马赫数
    {"GAMMA", 0},                   //气体绝热指数
    {"PR", 0},                      //普朗特数
    {"T_REF", 0},                   //无量纲参考温度
    {"EPSL_SW", 0},                 //S-w分裂中的稳定系数
    {"DT", 0},                      //时间步长
    {"END_TIME", 0},                //计算结束时间
    {"KSTEP_SHOW", 0},              //显示步数
    {"KSTEP_SAVE", 0},              //流场保存步数
    {"INIT_STAT", 0},               //流场初始化
    {"IBC", 0},                     //边界类型
    {"BC_NPARA", 0},                //边界参数（整型变量）
    {"BC_RPARA", 0},                //边界参数（浮点型变量）
    {"CHARTERIC", 0}                //特征重构
};
```

> 滤波模块部分可输入的控制参数为2个：

```
    FILTER_NPARA&                   //滤波参数（整型变量）
    FILTER_RPARA&                   //滤波参数（浮点型变量）
    //此处&代指从0开始的自然数，若程序调用一个滤波，控制参数名只有FILTER_NPARA0、FILTER_RPARA0，若程序同时调用多个滤波，则参数名应为FILTER_NPARA0、FILTER_RPARA0，FILTER_NPARA1、FILTER_RPARA1，FILTER_NPARA2、FILTER_RPARA2 ...
```

> 流场分析与后处理部分可输入的控制参数为3个：

```
    ANA_EVENT&                      //分析事件
    ANA_NPARA&                      //事件参数（整型变量）
    ANA_RPARA&                      //事件参数（浮点型变量）
    //此处&代指从0开始的自然数，若程序调用一个分析，控制参数名只有ANA_EVENT0、ANA_NPARA0、ANA_RPARA0，若程序同时调用多个分析，则参数名应为ANA_EVENT0、ANA_NPARA0、ANA_RPARA0，ANA_EVENT1、ANA_NPARA1、ANA_RPARA1，ANA_EVENT2、ANA_NPARA2、ANA_RPARA2 ...
    //若分析的事件没有控制参数，则不填写对应的ANA_NPARA和ANA_RPARA
```

> 混合格式相关参数为5个：

```
    HY_STYLE                    //混合格式类型
    HY_DP_INTV                  //阈值
    HY_SMOOTH_DP                //光滑性处理
    HY_PATCH_ZONE               //指定判别值的范围个数
    HY_ZONE&                    //指定判别值的范围与指定的判别值
```

OpenCFD-SCU参数读取模块

- OpenCFD-SCU通过遍历文本 opencfd-scu.in 获取以上定义的参数值。
- 在并行计算时，只有my_id为0的进程读取文本 opencfd-scu.in 获取参数值，后广播到其他进程。

> 参数读取模块代码如下：

```
    FILE * file = fopen("opencfd-scu.in","r");//打开文件opencfd-scu.in
    int nk,nr;

    if(file == NULL){
        printf("\033[31mopencfd-scu.in is not find!\033[0m\n");
        exit(-1);
    }//如果没有发现文件opencfd-scu.in则退出

    SearchItem(file, configList, configNum);//遍历文件opencfd-scu.in，找到控制文件中参数名对应的参数值

    sscanf(configList[0].value,"%d%d%d",&NX_GLOBAL,&NY_GLOBAL,&NZ_GLOBAL);//将网格数参数GRID_3D的值写入
NX_GLOBAL,NY_GLOBAL,NZ_GLOBAL
    sscanf(configList[1].value,"%d%d%d",&NPX0,&NPY0,&NPZ0);//将并行分割块数对应的参数PARALLEL_3D的值写入
NPX0,NPY0,NPZ0
    sscanf(configList[2].value,"%d",&dummy_i);//目前版本程序LAP参数程序默认值为4，即使读取LAP值也不生效，并将值写入到变量
dummy_i中
    sscanf(configList[3].value,"%d",&MSG_BLOCK_SIZE);//读取通讯方式，目前程序只有阻塞式MPI通讯一种，即使读取
MSG_BLOCK_SIZE值也不生效，并将值写入到无效变量MSG_BLOCK_SIZE中
    sscanf(configList[4].value,"%d",&Stream_MODE);//读取控制计算通讯重叠的参数STREAM，并将值写入到变量Stream_MODE中
    sscanf(configList[5].value,"%d",&TEST);//读取控制节点测试的参数TEST，并将值写入到变量TEST中

    sscanf(configList[6].value,"%d%d%d",&Iperiodic[0],&Iperiodic[1],&Iperiodic[2]);//读取控制周期性边条的参数
IPERIODIC，并将值写入到数组变量Iperiodic[3]中
    sscanf(configList[7].value,"%d%d%d",&Jacbound[0],&Jacbound[1],&Jacbound[2]);//读取控制Jacob计算边界格式的参数
JAC_BOUND，并将值写入到数组变量Jacbound[3]中

 sscanf(configList[8].value,"%d%d%d%d%d%d",&D0_bound[0],&D0_bound[1],&D0_bound[2],&D0_bound[3],&D0_bound[4],&D
0_bound[5]);//读取控制差分计算边界格式的参数DIF_BOUND，并将值写入到数组变量D0_bound[6]中

 sscanf(configList[9].value,"%d%d%d%d%d%d",&Non_ref[0],&Non_ref[1],&Non_ref[2],&Non_ref[3],&Non_ref[4],&Non_re
f[5]);//读取控制无反射边界条件的参数NON_REFLETION，并将值写入到数组变量Non_ref[6]中

    sscanf(configList[10].value,"%s", Scheme_invis);//读取控制无粘项计算格式的参数SCHEME_INVIS，并将值写入到变量
Scheme_invis中
    sscanf(configList[11].value,"%s", Scheme_vis);//读取控制粘性项计算格式的参数SCHEME_VIS，并将值写入到变量Scheme_vis
中
    SCHEME_CHOOSE scheme = {Scheme_invis, Scheme_vis};//将格式信息传递到结构体变量scheme中
    Schemes_Choose_ID(&scheme);//判断选择的格式与对应的编号

    if(strcmp(Scheme_invis, "SCHEME_HYBRIDAUTO") == 0) IFLAG_HybridAuto = 1;//判断是否是混合格式

    HybridAuto.Num_Patch_zones = 0;
    HybridAuto.IF_Smooth_dp = 0;

    HybridAuto.P_intvs = (REAL *)malloc((HybridA_Stage - 1)*sizeof(REAL));
    HybridAuto.zones = (int *)malloc(6*Patch_max*sizeof(int));
    HybridAuto.Pa_zones = (REAL *)malloc(Patch_max*sizeof(REAL));

    if(IFLAG_HybridAuto == 1){//如何启用混合格式则读取以下对应参数

        int (*HybridAuto_zones)[6] = (int(*)[6])HybridAuto.zones;

        configItem Hybridbuff = {"HY_DP_INTV", 0};
        SearchItem(file, &Hybridbuff, 1);

        tmp = PartItem(Hybridbuff.value, Part_buff);
        for(int i=0;i<(HybridA_Stage-1);i++) sscanf(Part_buff[i],"%lf",&HybridAuto.P_intvs[i]);//读取控制混合格式
阈值的参数HY_STYLE，对应变量写入到HybridAuto.Style中，目前默认HybridA_Stage为3，即只有2个阈值

        sprintf(Hybridbuff.name, "HY_STYLE");
        SearchItem(file, &Hybridbuff, 1);
```

```
        sscanf(Hybridbuff.value,"%d",&HybridAuto.Style);//读取控制混合类型的参数HY_STYLE，对应值写入到
HybridAuto.Style中

        if(HybridAuto.Style != 1 && HybridAuto.Style != 2){
            printf("\033[31mHYBRID SCHEMES CHOOSE IS WRONG! ! ! \033[0m\n");
            exit(0);
        }

        sprintf(Hybridbuff.name, "HY_SMOOTH_DP");
        SearchItem(file, &Hybridbuff, 1);
        sscanf(Hybridbuff.value,"%d",&HybridAuto.IF_Smooth_dp);//读取是否做光滑性处理的参数HY_SMOOTH_DP，对应值写入到
HybridAuto.IF_Smooth_dp中

        sprintf(Hybridbuff.name, "HY_PATCH_ZONE");
        SearchItem(file, &Hybridbuff, 1);
        sscanf(Hybridbuff.value,"%d",&HybridAuto.Num_Patch_zones);//读取patch区数量的变量

        for(int i=0; i<HybridAuto.Num_Patch_zones; i++){
            sprintf(Hybridbuff.name, "HY_ZONE%d", i);
            SearchItem(file, &Hybridbuff, 1);

            sscanf(Hybridbuff.value,"%d%d%d%d%d%d%lf",&HybridAuto_zones[i][0],&HybridAuto_zones[i]
[1],&HybridAuto_zones[i][2],
            &HybridAuto_zones[i][3],&HybridAuto_zones[i][4],&HybridAuto_zones[i]
[5],&HybridAuto.Pa_zones[i]);//读取patch区对应的区域，将参数HY_ZONE&的值写入到数组HybridAuto_zones对应的行和列中，行表示
patch区域起止的范围，列表示第几个patch区。
        }
    }

    sscanf(configList[12].value,"%lf",&Re);//读取控制雷诺数的参数RE，并将值写入到变量Re中
    sscanf(configList[13].value,"%lf",&Ama);//读取控制马赫数的参数AMA，并将值写入到变量Ama中
    sscanf(configList[14].value,"%lf",&Gamma);//读取控制气体绝热指数的参数GAMMA，并将值写入到变量Gamma中
    sscanf(configList[15].value,"%lf",&Pr);//读取控制普朗特数的参数PR，并将值写入到变量Pr中
    sscanf(configList[16].value,"%lf",&Ref_T);//读取控制无量纲参考温度的参数T_REF，并将值写入到变量Ref_T中
    sscanf(configList[17].value,"%lf",&epsl_SW);//读取控制S-W分裂中的稳定系数的参数EPSL_SW，并将值写入到变量epsl_SW中

    sscanf(configList[18].value,"%lf",&dt);//读取控制时间步长的参数DT，并将值写入到变量dt中
    sscanf(configList[19].value,"%lf",&end_time)//读取控制计算结束时间的参数END_TIME，并将值写入到变量end_time中;
    sscanf(configList[20].value,"%d",&Kstep_show);//读取控制显示步数的参数KSTEP_SHOW，并将值写入到变量Kstep_show中
    sscanf(configList[21].value,"%d",&Kstep_save);//读取控制流场保存步数的参数KSTEP_SHOW，并将值写入到变量Kstep_save中
    sscanf(configList[22].value,"%d",&Init_stat);//读取控制流场初始化的参数INIT_STAT，并将值写入到变量Init_stat中

    sscanf(configList[23].value,"%d",&IBC_USER);//读取控制边界类型的参数IBC，并将值写入到变量IBC_USER中
    BC_npara = (int*)malloc(sizeof(int)*100);
    BC_rpara = (REAL*)malloc(sizeof(REAL)*100);

    nk = PartItem(configList[24].value, Part_buff);//统计边界参数（整形变量）BC_NPARA具有几个变量值，并写入到数组
Part_buff中
    for(int i=0;i<nk;i++) sscanf(Part_buff[i],"%d",BC_npara+i);//将数组Part_buff的值赋于数组BC_npara

    nr = PartItem(configList[25].value, Part_buff);//统计边界参数（浮点型变量）BC_RPARA具有几个变量值，并写入到数组
Part_buff中
    for(int i=0;i<nr;i++) sscanf(Part_buff[i],"%lf",BC_rpara+i);//将数组Part_buff的值赋于数组BC_rpara

    sscanf(configList[26].value,"%d", &IF_CHARTERIC);//读取控制特征重构的参数CHARTERIC，并将值写入到变量IF_CHARTERIC
中

    int NameNUM[1000];
    //滤波功能的参数读取
    NFiltering = ItemNUM(file, "FILTER_NPARA", &NameNUM[0]);//判断调用滤波的次数

    Filter_para = (int(*)[11])malloc(sizeof(int)*(NFiltering+1)*11);
    Filter_rpara = (REAL(*)[3])malloc(sizeof(REAL)*(NFiltering+1)*3);

    for(int i=0;i<NFiltering;i++){
        configItem Hybridbuff;
        //ntime, Filter_X, Filter_Y, Filter_Z, ib, ie, jb, je, kb, ke, Filter_scheme
        sprintf(Hybridbuff.name, "FILTER_NPARA%d", NameNUM[i]);
        SearchItem(file, &Hybridbuff, 1);

        tmp = PartItem(Hybridbuff.value, Part_buff);
        for(int n=0;n<11;n++) sscanf(Part_buff[n],"%d",&Filter_para[i][n]);//读取FILTER_NPARA&（滤波整形参数）对应
的值写入数组Filter_para[&][11]中

        sprintf(Hybridbuff.name, "FILTER_RPARA%d", NameNUM[i]);
        SearchItem(file, &Hybridbuff, 1);
        tmp = PartItem(Hybridbuff.value, Part_buff);
```

```
        for(int n=0;n<3;n++) sscanf(Part_buff[n],"%lf",&Filter_rpara[i][n]);//读取FILTER_RPARA&（滤波浮点型参数）对应的值写入数组Filter_rpara[&][3]中
    }

    //后处理和分析功能的参数读取
    N_ana = ItemNUM(file, "ANA_EVENT", &NameNUM[0]);//判断开启了几个分析功能

    ANA_npara = (int(*)[100])malloc(sizeof(int)*100*N_ana);
    ANA_rpara = (REAL(*)[100])malloc(sizeof(REAL)*100*N_ana);
    K_ana = (int*)malloc(sizeof(int)*N_ana);
    Kstep_ana = (int*)malloc(sizeof(int)*N_ana);

    for(int i=0;i<N_ana;i++){
        configItem Hybridbuff;
        sprintf(Hybridbuff.name, "ANA_EVENT%d", NameNUM[i]);
        SearchItem(file, &Hybridbuff, 1);//按顺序查找ANA_EVENT开头的变量，并获取值

        sscanf(Hybridbuff.value,"%d%d",K_ana+i,Kstep_ana+i);//将分析类型与分析步数写入数组K_ana和Kstep_ana

        sprintf(Hybridbuff.name, "ANA_NPARA%d", NameNUM[i]);
        SearchItem(file, &Hybridbuff, 1);//查找分析所需的整形参数ANA_NPARA&

        nk = PartItem(Hybridbuff.value, Part_buff);//分析参数个数
        for(int n=0;n<nk;n++) sscanf(Part_buff[n],"%d",&ANA_npara[i][n]);//将参数写入到数组ANA_npara中

        sprintf(Hybridbuff.name, "ANA_RPARA%d", NameNUM[i]);
        SearchItem(file, &Hybridbuff, 1);//查找分析所需的浮点型参数ANA_RPARA

        nr = PartItem(Hybridbuff.value, Part_buff);
        for(int n=0;n<nr;n++) sscanf(Part_buff[n],"%lf",&ANA_rpara[i][n]);//将参数写入到数组ANA_rpara中
    }

    fclose(file);
```

控制文件参数填写注意事项与说明

- 为防止由于大小写容易混淆造成的错误，控制文件中所有参数均为**大写**，具体参数见参数定义模块说明。
- 参数名前后可加空格，但不可以加其他特殊字符（添加特殊字符会使此行参数无效）。
- 参数填写没有顺序要求，只要在控制文件任意行中填写参数名与对应参数值，程序可自动查找识别。
- 一行只能填一个参数名与对应值，其后加注释等其他文字不会造成错误。
- 控制文件中参数赋值时，参数名与参数值以**等号**隔开，参数具有多个输入值时，输入值以**空格**隔开，空格个数无限制。
- 对于滤波模块，程序支持计算时启用多个滤波。若程序只调用一个滤波，滤波控制参数名应为FILTER_NPARA0、FILTER_RPARA0，若程序同时调用多个滤波，则参数名末尾以自然数排列，如FILTER_NPARA0、FILTER_RPARA0，FILTER_NPARA1、FILTER_RPARA1，FILTER_NPARA2、FILTER_RPARA2 ...，程序可根据参数名自动判断调用了几个滤波。
- 对于流场分析与后处理模块，程序支持计算时启用多个分析与后处理。与滤波类似，参数名末尾以自然数排列，程序可自动识别调用了几个分析与后处理。如调用三个分析与后处理时，参数名应为ANA_EVENT0、ANA_NPARA0、ANA_RPARA0，ANA_EVENT1、ANA_NPARA1、ANA_RPARA1，ANA_EVENT2、ANA_NPARA2、ANA_RPARA2。
- 混合格式相关参数与分析只有在混合格式启用时才生效。
- 控制文件目前可读入但无用的参数有（存入废变量中，以便后续扩展功能使用）：

> MSG_BLOCK_SIZE

> LAP

参数说明			参数值个数
GRID_3D	三个方向的网格数		3
PARALLEL_3D	三个方向的并行分割数目（三维剖分时，应保证每个进程的数据块三个维度网格数目接近）		3
	1.00.28版本程序当单进程网格规模设置为280*280*280时，DCU显存（16G）占比70%		
LAP	重叠区宽度（此参数目前版本不生效,重叠区默认为4）		1
MSG_BLOCK_SIZE	MPI通讯类型（此参数目前版本不生效,只包含一种MPI通讯方式---阻塞式）		1
STREAM	是否开启计算通讯重叠（建议开启，开启后每个进程的每个方向网格数应大于24，否则会造成程序卡死）		1
	0	- 不开启计算通讯重叠	
	1	- 开启计算通讯重叠	
CHARTERIC	是否开启特征重构（特征重构可使计算鲁棒性大幅提升，但计算速度有0.5倍到1倍的减慢）		1
	0	- 不启用特征重构	
	1	- 启用特征重构	
TEST	是否进行节点测试（每个MPI进程各自执行单独的任务，打印各进程计算机名与计算时间，可用来筛选慢节点）		1
	0	- 不进行测试	
	1	- 进行测试	
IPERIODIC	是否是周期边界条件		3
	0	- 不是周期边界条件	
	1	- 周期边界条件	
JAC_BOUND	计算Jacobian系数时是否调用边界格式（只有当网格的两侧物理坐标相连接时，如顿锥展向，可以不调用边界格式计算Jacobian系数）		6
	0	- 不调用边界格式	
	1	- 调用边界格式	
DIF_BOUND	进行差分计算时是否调用边界格式（只有周期性边界条件时可以不调用边界格式）		6
	0	- 不调用边界格式	
	1	- 调用边界格式	
NON_REFLETION	是否是无反射边界条件		6
	0	- 不是无反射边界条件	
	1	- 无反射边界条件	
SCHEME_INVIS	无粘项差分格式		1
	UP7	- 7阶迎风	
	WENO5	- 5阶WENO	
	WENO7	- 7阶WENO	
	WENO7_SYMBO	- WENO SYMBO格式	
	WENO7_SYMBO_LIM	- 加了限制器的WENO SYMBO格式	
	NND2	- 2阶NND格式	
	OMP6_HR	- 6阶高鲁棒优化保单调格式	
	OMP6_LD	- 6阶低耗散优化保单调格式	
	OMP6_CD8	- 6阶中心型优化保单调格式	
	SCHEME_HYBRIDAUTO	- 混合格式	
SCHEME_INVIS	粘性项差分格式		1
	CD6	- 6阶中心格式	

	CD8	- 8阶中心格式		
RE	雷诺数			1
AMA	马赫数			1
GAMMA	气体绝热指数			1
PR	普朗特数			1
T_REF	无量纲参考温度			1
EPSL_SW	S-W分裂中的稳定系数			1
DT	时间步长			1
END_TIME	计算结束时间			1
KSTEP_SHOW	显示步数			1
KSTEP_SAVE	流场保存步数			1
INIT_STAT	如何进行流场初始化			1
	0	- 程序内部生成原始变量全为1的流场，不读入外部网格和初始场		
	1	- 需要读入外部网格和初始场		
边界条件相关参数				
IBC	边界条件类型（目前程序中集成了2种边界条件）			1
	108	针对平板、压缩折角等算例的边界条件		
	124	针对顿锥、升力体等算例的边界条件		
BC_NPARA	整型的边界条件参数			
	108边界条件对应的整型参数	BC_NPARA[0]	空间上叠加的扰动波数量 MZMAX	4
		BC_NPARA[1]	时间上叠加的扰动波数量 MTMAX	
		BC_NPARA[2]	是否读取入口边界条件 INLET_BOUNDARY	
			0 - 不读取	
			1 - 读取	
		BC_NPARA[3]	是否读壁面条件（当前版本程序此参数不生效） IFLAG_WALL_NOT_NORMAL	
	124边界条件对应的整型参数	BC_NPARA[0]	展向是否是对称边界条件 IF_SYMMETRY	5
			0 - 不采取对称边界条件	
			1 - 采取对称边界条件	
		BC_NPARA[1]	是否包含头部（是否读入口边条文件） INLET_BOUNDARY	
			0 - 包含头部	
			1 - 不包含头部（读文件）	
		BC_NPARA[2]	是否包含激波（是否读外边条文件） IFLAG_UPPERBOUNDARY	
			0 - 包含激波	
			1 - 不包含激波（读文件）	
		BC_NPARA[3]	空间上叠加的扰动波数量 MZMAX	
		BC_NPARA[4]	时间上叠加的扰动波数量 MTMAX	
BC_RPARA	浮点型的边界条件参数			
	108边界条件对应的浮点型参数	BC_RPARA[0]	壁温 TW	8
		BC_RPARA[1]	扰动强度（幅值） EPSL	
		BC_RPARA[2]	扰动起始位置 X_DIST_BEGIN	

		BC_RPARA[3]	扰动结束位置 X_DIST_END		
		BC_RPARA[4]	扰动频率 BETA		
		BC_RPARA[5]	壁面开始位置（应包含全部计算域） X_WALL_BEGIN		
		BC_RPARA[6]	上边界来流条件起始位置 X_UP_BOUNDARY_BEGIN		
		BC_RPARA[7]	展向计算域长度 SLZ		
	124边界条件对应的浮点型参数	BC_RPARA[0]	攻角 AOA		7
		BC_RPARA[1]	壁温 TW		
		BC_RPARA[2]	壁面扰动强度（幅值） EPSL_WALL		
		BC_RPARA[3]	上边界扰动强度 EPSL_UPPER		
		BC_RPARA[4]	扰动频率 BETA		
		BC_RPARA[5]	扰动起始位置 WALL_DIS_BEGIN		
		BC_RPARA[6]	扰动结束位置 WALL_DIS_END		
滤波相关参数					
FILTER_NPARA&	整型滤波相关参数				11
	FILTER_NPARA&[0]	滤波步数间隔 Filter_step			
	FILTER_NPARA&[1]	x方向是否滤波 fiter_judge_X			
		0	- 不开启滤波		
		1	- 开启滤波		
	FILTER_NPARA&[2]	y方向是否滤波 fiter_judge_Y			
		0	- 不开启滤波		
		1	- 开启滤波		
	FILTER_NPARA&[3]	z方向是否滤波 fiter_judge_Z			
		0	- 不开启滤波		
		1	- 开启滤波		
	FILTER_NPARA&[4]	x方向滤波区域起始网格点 ib			
	FILTER_NPARA&[5]	x方向滤波区域结束网格点 ie			
	FILTER_NPARA&[6]	y方向滤波区域起始网格点 jb			
	FILTER_NPARA&[7]	y方向滤波区域结束网格点 je			
	FILTER_NPARA&[8]	z方向滤波区域起始网格点 kb			
	FILTER_NPARA&[9]	z方向滤波区域结束网格点 ke			
	FILTER_NPARA&[10]	滤波类型（目前版本不生效，程序只集成了一种滤波，守恒形式） Filter_scheme			
FILTER_RPARA&	浮点型滤波相关参数				3
	FILTER_RPARA&[0]	滤波强度（滤波强度越大对流场的改变程度越大，默认值一般为1.0） s0			
	FILTER_RPARA&[1]	滤波的阀值（此值越小接受滤波的范围越大，对流场改变越大，默认值一般为e-5） rth			
	FILTER_RPARA&[2]	滤波结束的时刻，当流场时间超过此值后，滤波关闭 Filter_end			
后处理相关参数					
ANA_EVENT&	ANA_EVENT&[0]	分析或后处理功能模块选择	100	对流场进行过滤，分析出流场中的发散点，即nan或出现负温度的点 ana_NAN_and_NT	2
			101	流场时间平均 ana_time_average	

			102	当无粘项格式为混合格式时方可生效，输出x、y、z各个中截面混合格式的分布情况 HybridAuto_scheme_IO	
			103	输出流场的Q判据文件 get_Q	
			104	保存XY块流场时间序列 ana_saveplaneXY	
			105	保存YZ块流场时间序列 ana_saveplaneYZ	
			106	保存XZ块流场时间序列 ana_saveplaneXZ	
			107	当无粘项格式为混合格式时方可生效，屏幕打印混合格式的格式占比 HybridAuto_scheme_Proportion	
	ANA_EVENT&[1]	进行流场分析或后处理的步数间隔 Kstep_ana			
ANA_NPARA&	104	ANA_NPARA[0]	需要保存的XY块流场时间序列个数 point		2+ANA_NPARA[0]
		ANA_NPARA[1]	需要保存的XY块流场宽度 bandwidth		
		ANA_NPARA[2]	需要保存的第一个XY块流场起始网格点		
		ANA_NPARA[2+...]	需要保存的第n个XY块流场起始网格点		
		ANA_NPARA[2+point-1]	需要保存的第point个XY块流场起始网格点		
	105	ANA_NPARA[0]	需要保存的YZ块流场时间序列个数 point		2+ANA_NPARA[0]
		ANA_NPARA[1]	需要保存的YZ块流场宽度 bandwidth		
		ANA_NPARA[2]	需要保存的第一个YZ块流场起始网格点		
		ANA_NPARA[2+...]	需要保存的第n个YZ块流场起始网格点		
		ANA_NPARA[2+point-1]	需要保存的第point个YZ块流场起始网格点		
	106	ANA_NPARA[0]	需要保存的XZ块流场时间序列个数 point		2+ANA_NPARA[0]
		ANA_NPARA[1]	需要保存的XZ块流场宽度 bandwidth		
		ANA_NPARA[2]	需要保存的第一个XZ块流场起始网格点		
		ANA_NPARA[2+...]	需要保存的第n个XZ块流场起始网格点		
		ANA_NPARA[2+point-1]	需要保存的第point个XZ块流场起始网格点		
混合格式相关参数					
HY_STYLE	混合格式类型选择				1
	1	利用压力梯度的混合格式（鲁棒性较强，耗散偏大，只适合存在激波的特定问题）			
	2	使用Jameson推荐的激波识别器进行格式选择的混合格式（推荐）			
HY_DP_INTV	混合格式阈值，1型混合格式推荐1 8，2型混合格式推荐0.01 0.2				2
HY_SMOOTH_DP	是否启用光滑性处理（目前只有1型混合格式可用此功能）				1
HY_PATCH_ZONE	指定判别值的范围个数（目前只有1型混合格式可用此功能）				1
HY_ZONE&	指定判别值的范围与指定的判别值				7
	HY_ZONE&[0]	x方向起始网格点			
	HY_ZONE&[1]	x方向结束网格点			
	HY_ZONE&[2]	y方向起始网格点			
	HY_ZONE&[3]	y方向结束网格点			

	HY_ZONE&[4]	z方向起始网格点
	HY_ZONE&[5]	z方向结束网格点
	HY_ZONE&[6]	指定的判别值（处于哪个阈值区间内，此区域将全部使用阈值对应的格式）

OpenCFD-SCU控制文件示例

```
GRID_3D = 8800   640   800              //此算例的网格规模为8800*640*800
PARALLEL_3D = 24   4   4                //并行分割方式为24*4*4，共用了384个进程
STREAM = 1                              //启用了计算通讯重叠
CHARTERIC = 1                           //启用了特征重构

IPERIODIC = 0   0   1                   //z方向启用了周期性边界条件
JAC_BOUND = 1   1   1                   //计算Jacobian系数时三个方向都使用了边界格式
DIF_BOUND = 1   1   1   1   0   0       //x，y的正负方向都使用了边界格式
NON_REFLETION = 0   1   0   1   0   0   //x+，y+方向使用了无反射边界条件

#SCHEME_INVIS = WENO7_SYMBO
SCHEME_VIS = CD8                        //粘性项计算采用了8阶中心
SCHEME_INVIS = SCHEME_HYBRIDAUTO        //无粘性计算采用了混合格式
HY_STYLE = 2                            //混合格式类型为2型
HY_DP_INTV = 0.02 0.1                   //阈值为0.02和0.1
#HY_SMOOTH_DP = 1
#HY_PATCH_ZONE = 0
#HY_ZONE0 = 10   25   100   240   5   20   20.0


RE = 100000.0                           //雷诺数为100000
AMA = 10.0                              //马赫数为10.0
GAMMA = 1.40                            //气体绝热指数为1.4
PR = 0.70                               //普朗特数为0.7
T_REF = 79.0                            //参考温度为79.0
EPSL_SW = 0.0                           //sw分裂时没有添加增强稳定性的系数

DT = 0.002                              //时间步长为0.002
END_TIME = 3000                         //计算结束的无量纲时间为3000
KSTEP_SHOW = 1                          //每一步进行一次流场显示，打印计算时间，平均温度、能量等信息
KSTEP_SAVE = 10000                      //每10000步保存一次流场
INIT_STAT = 1                           //读外部初始场和网格

IBC = 108                               //边界条件选择为108
#mzmax, mtmax, Inlet_boundary, If_wall_not_normal
BC_NPARA = -10   1   1   0              //108边界条件对应整型参数

#Tw, epsl, x_dis_begin, x_dis_end, beta, x_wall_begin, x_up_bound_begin, SLZ
BC_RPARA = 3.72   0.1   -320.   -300   0.08   -1000.   1000.   24.                    //108边界条件对应浮点型参数

#nstep_filter, Filter_X, Filter_Y, Filter_Z, ib, ie, jb, je, kb, ke, Filter_scheme  // s0, rth,
Filter_end_time
#FILTER_NPARA0 = 10   1   0   0   700   900   300   500   0   800   2
#FILTER_RPARA0 = 1.0   1.e-6   310000

ANA_EVENT0 = 107   100
ANA_EVENT1 = 101   1
ANA_EVENT2 = 105   100
ANA_NPARA2 = 2   5   3000   5000        //启用了三个流场分析和后处理事件，第三个事件有需要输入的整型参数
```