CHAPTER 3

Input Parameters

Descriptions of each of the input line types are provided in this chapter where the section numbers denote the line type (LT) number referred to throughout the manual. An example input file is provided below for a flat plate case. Note that not all of the line types of information will be required for every application.

```
I/O FILES
grdflat5.bin
plot3dg.bin
plot3dq.bin
cfl3d.out
cfl3d.res
cfl3d.turres
cfl3d.blomax
cfl3d.out15
cfl3d.prout
cfl3d.out20
ovrlp.bin
patch.bin
restart.bin
    turbulent flat plate (plate from j=17-65, prior to 17 is symmetry)
     XMACH     ALPHA      BETA  REUE,MIL   TINF,DR     IALPH    IHSTRY
    0.2000    00.000       0.0    06.000     460.0         0         0
      SREF      CREF      BREF       XMC       YMC       ZMC
   1.00000   1.00000    1.0000   0.00000      0.00      0.00
        DT     IREST   IFLAGTS      FMAX     IUNST    CFLTAU
    -5.000         0       000   05.0000         0      10.0
     NGRID   NPLOT3D    NPRINT    NWREST      ICHK       I2D    NTSTEP       ITA
         1         1         0      1200         0         1         1         1
       NCG       IEM  IADVANCE    IFORCE  IVISC(I)  IVISC(J)  IVISC(K)
         2         0         0       001         0         0         7
    IDIM    JDIM    KDIM
      02      65      97
    ILAMLO    ILAMHI    JLAMLO    JLAMHI    KLAMLO    KLAMHI
         1         2         1        17         1        97
     INEWG    IGRIDC        IS        JS        KS        IE        JE        KE
         0         0         0         0         0         0         0         0
  IDIAG(I)  IDIAG(J)  IDIAG(K)  IFLIM(I)  IFLIM(J)  IFLIM(K)
         1         1         1         0         0         0
   IFDS(I)   IFDS(J)   IFDS(K)  RKAP0(I)  RKAP0(J)  RKAP0(K)
         1         1         1    0.3333    0.3333    0.3333
      GRID     NBCI0   NBCIDIM     NBCJ0   NBCJDIM     NBCK0   NBCKDIM    IOVRLP
         1         1         1         1         1         2         1         0
I0:   GRID   SEGMENT    BCTYPE      JSTA      JEND      KSTA      KEND     NDATA
         1         1      1001         0         0         0         0         0
IDIM: GRID   SEGMENT    BCTYPE      JSTA      JEND      KSTA      KEND     NDATA
         1         1      1002         0         0         0         0         0
J0:   GRID   SEGMENT    BCTYPE      ISTA      IEND      KSTA      KEND     NDATA
         1         1      1008         0         0         0         0         0
JDIM: GRID   SEGMENT    BCTYPE      ISTA      IEND      KSTA      KEND     NDATA
         1         1      1002         0         0         0         0         0
K0:   GRID   SEGMENT    BCTYPE      ISTA      IEND      JSTA      JEND     NDATA
         1         1      1001         0         0         1        17         0
         1         2      2004         0         0        17        65         2
               TWTYPE        CQ
                  0.        0.
KDIM: GRID   SEGMENT    BCTYPE      ISTA      IEND      JSTA      JEND     NDATA
         1         1      1003         0         0         0         0         0
```

```
     MSEQ    MGFLAG    ICONSF       MTT      NGAM
        1         1         0         0        02
     ISSC EPSSSC(1) EPSSSC(2) EPSSSC(3)      ISSR EPSSSR(1) EPSSSR(2) EPSSSR(3)
        0       0.3       0.3       0.3         0       0.3       0.3       0.3
     NCYC    MGLEVG     NEMGL     NITFO
     0800         03        00       000
     MIT1      MIT2      MIT3      MIT4      MIT5      MIT6      MIT7      MIT8
       01        01        01        01        01         1         1         1
  1-1 BLOCKING DATA:
     NBLI
        0
NUMBER   GRID     :    ISTA   JSTA   KSTA   IEND   JEND   KEND  ISVA1  ISVA2
NUMBER   GRID     :    ISTA   JSTA   KSTA   IEND   JEND   KEND  ISVA1  ISVA2
 PATCH SURFACE DATA:
   NINTER
        0
 PLOT3D OUTPUT:
  GRID IPTYPE ISTART   IEND   IINC JSTART   JEND   JINC KSTART   KEND   KINC
     1      0      0      0      0      0      0      0      0      0      0
IMOVIE
     0
 PRINT OUT:
  GRID IPTYPE ISTART   IEND   IINC JSTART   JEND   JINC KSTART   KEND   KINC
 CONTROL SURFACE:
 NCS
   0
  GRID ISTART   IEND   JSTART   JEND   KSTART   KEND  IWALL  INORM
```

3.1 LT1 - Input/Output File Names

Input/Output file names, up to 60 characters each, starting in column 1. Specify the user file names in the following order (one name per line), using a "dummy" file name if a file is not actually needed for the current problem:

Input/Output File	File Type
grid (unit 1)	binary[1] input
PLOT3D grid (unit 3)	binary[2],[3] output
PLOT3D solution (unit 4)	binary[2],[3] output
primary case information (unit 11)	ascii output
residual and force coefficient history (unit 12)	ascii output
turbulence model residual history (*not* Baldwin-Lomax) (unit 13)	ascii output
Baldwin-Lomax (unit 14)	ascii output

[1] Defaulted in the code to standard binary for the particular machine on which the code is executed. On Cray computers the default can be changed to IEEE Big unformatted by uncommenting "`call asnfile`" one line prior to the open statement for unit 1 in the main routine.

[2] Defaulted in the code to standard binary for the particular machine on which the code is executed. On Cray computers the default can be changed to IEEE Big unformatted by uncommenting the "`call asnfile`" lines one line prior to the open statements for both unit 3 and unit 4 in the main routine.

[3] Can be changed to formatted by changing the open statements for unit 3 and unit 4 to "`formatted`" and changing `ibin` from 1 to 0 in subroutine `qout`.

secondary case information (unit 15) . . . . . ascii output

flow-field variables printout (unit 17) . . . . . ascii output

unsteady pressures from forced oscillations (**iunst** > 0) (unit 20) . . . . ascii output

interpolation coefficients for chimera grids (unit 21) . . . . . binary input

interpolation coefficients for patched grids (unit 22) . . . . . binary input

restart (unit 2) . . . . . binary input/output

3.2 LT2 - Case Title

title describing case

3.3 LT3 - Flow Conditions

xmach – free-stream[1] Mach number

alpha – angle of attack (See Appendix I.)

beta – side-slip angle (See Appendix I.)

reue – free-stream[1] Reynolds number per unit grid length (millions)

How to Set **reue**

Suppose it is desired to simulate a flow in which the Reynolds number based on some characteristic length (e.g. the chord) is Re. If L is the corresponding length in the grid (ignore the dimension, if any, of L), then set **reue** = Re/L. For example, if unit "1" in the grid corresponds to "1 inch" in the experiment, then give **reue** as the Reynolds number per inch. If unit "1" in the grid corresponds to "16.7845 furlongs", then give **reue** as the Reynolds number per 16.7845 furlongs. (Also remember that in the input file, **reue**/1×10^6 is the actual input value, so put a "20" for "20 million", etc.) (See "Reynolds Number Examples" on page 55.)

tinf – free-stream[1] temperature (degrees Rankine)

ialph – indicator for determining how angle of attack is measured in PLOT3D-type grids (**ngrid** < 0)

If	*Then*
ialph = 0	**alpha** is measured in the $x-z$ plane (with z "up"), i.e. **alpha** = 90° would give a free-stream velocity vector in the positive z direction.

[1] See the note on page 3 about the usage of the phrase *free stream*.

ialph > 0 — **alpha** is measured in the $x - y$ plane (with y "up"), i.e. **alpha** = 90° would give a free-stream velocity vector in the positive y direction.

Note

(1) For CFL3D-type grids, **alpha** is always measured in the $x - z$ plane, with z "up". (If it is desired to have y "up" with CFL3D-type grids, then set **ialph** > 0 and comment out the following line in `subroutine global`:

```
if (ngrid.gt.0) ialph=0
```

ihstry – determines which variables are to be tracked for a convergence history (i.e., which variables are output to file `cfl3d.res` (unit 12) and file `cfl3d.subit.res` (unit 23))

If	*Then*
ihstry = 0	standard convergence history: residual, C_l, C_d, C_y or C_z depending on whether z or y is "up", respectively, and C_{my} or C_{mz} (for z or y "up", respectively)
ihstry > 0	control surface history: residual and mass flow, pressure force, viscous force, thrust (momentum) force (forces are resultant forces, i.e. if f_x, f_y, and f_z are the force components in the x, y, and z directions, then the resultant force is $\sqrt{f_x^2 + f_y^2 + f_z^2}$; must have **ncs** > 0 and specify which control surfaces are to be tracked - only those surfaces with **inorm** 0 are included in the sum)

Note

(1) "LT3 - Flow Conditions" of CFL3D Versions 4.1 and earlier had additional parameters **isnd** and **c2spe** to govern wall temperature and heat transfer. These parameters are no longer used, but their functions are implemented in a more general fashion in boundary condition type 2004. (See notes under "LT14 - I0 Boundary Condition Specification" on page 32). The position formerly occupied by **isnd** is now occupied by **ialph**, so that the value of **ialph** for PLOT3D-type grids is no longer hard-wired in subroutine `rp3d` (as in earlier versions of the code).

3.4 LT4 - Reference Quantities

sref – reference area used to compute non-dimensional forces and moments; **sref** is generally taken to be the plan-form area (*not* the wetted area) of a wing, for example. It should be given in grid dimensions. (You must also account for the grid-distance between 2-D planes when **i2d**=1!)

cref – reference length used to compute non-dimensional moments

bref – reference span used to compute non-dimensional moments

xmc – moment center in x direction

ymc – moment center in y direction

zmc – moment center in z direction

3.5 LT5 - Time Step Parameters

dt – time step

If	*Then*
dt < 0	local time stepping, CFL number = |**dt**|; Use **ncyc** to control the number of cycles. Sub-iterations are not used.
dt > 0	constant time step equal to **dt**; Use **ntstep** to control the number of time steps and **ncyc** to control the number of sub-iterations. Two different sub-iteration strategies can be employed: t-TS and τ-TS. The t-TS method can have unacceptable time-step limitations depending on the case. The τ-TS method allows for much larger time steps, although larger time steps require more sub-iterations. Multigrid may be used to enhance convergence of sub-iterations and is recommended in general.

If	*Then*	
t-TS	**ntstep**	= number of time steps
	ita	= +1 or +2 (+2 recommended)
	mgflag	= 0 or 1 (1 recommended)
	mglevel	= number of multigrid levels
	ncyc	= number of sub-iterations + 1 (typically from 2 to 15) (**ncyc** = 1 will yield standard time-accurate method with *no* sub-iterations.)
τ-TS	**ntstep**	= number of time steps
	ita	= –1 or –2 (–2 recommended) (See "LT6 - Options and Specifications" on page 23)
	mgflag	= 0 or 1 (1 recommended)
	mglevel	= number of multigrid levels
	ncyc	= number of sub-iterations + 1 (typically from 5 to 25)

Notes

(1) The number of sub-iterations required depends partly on the size of the physical time step used. More sub-iterations are required for larger time steps. The use of multigrid will generally reduce the number of sub-iterations required, although the cost per sub-iteration will increase.

(2) It is *strongly* recommended that the sub-iteration convergence be monitored in the output file `cfl3d.subit_res` to determine if sufficient sub-iterations are being used for a particular case. The residual should drop and the force coefficients should tend toward constant values during each sub-iteration. However, from an efficiency standpoint, it is not generally desirable to fully converge each time step.

(3) The τ-TS scheme utilizes, in addition to the physical time step, a "pseudo" time step based on local time stepping. Thus, there is a CFL number associated with τ-TS (see **cfl_tau** on page 23).

(4) See "Time and Time Step" on page 56 for the nondimensionalization of the time step.

(5) To obtain second order accuracy in time (|**ita**| = 2), sub-iterations (**ncyc** > 1) must be used.

irest – restart flag

If	*Then*
irest = 0	no restart
irest > 0	restart from previous solution, using restart file (unit 2)
irest < 0	restart from previous solution, but do not save previous history information

iflagts – time step ramping flag

If	*Then*
iflagts = 0	constant **dt**
iflagts > 0	**dt** ramped over **iflagts** steps to $\mathbf{dt} \times \mathbf{fmax}$

fmax – maximum increase in **dt**; $\mathbf{dt}_{\text{final}} = \mathbf{fmax} \times \mathbf{dt}_{\text{initial}}$

Notes

(1) The ramping of the time step/CFL number to $\mathbf{dt}_{\text{final}}$ is non-linear, occurring slowly at first and then increasing in rate.

(2) When the time step has successfully been ramped to $\mathbf{dt}_{\text{final}}$, remember to set the input value of **dt** to $\mathbf{dt}_{\text{final}}$ before restarting with the previous solution. Also, remember to change **iflagts** to zero and/or **fmax** to 1.0.

iunst – unsteady mesh flag

If	*Then*
iunst = 0	stationary grid
iunst = 1	dynamic grid (translation or rotation)

Notes
(1) If **dt** < 0, **iunst** is automatically set to 0.
(2) See "LT33 - Number of Translated Grids (extra comment line needed prior)" on page 43 and beyond for specification of dynamic grid input.

cfl_tau – CFL number for τ-TS scheme; **cfl_tau** is always > 0 and it is not used for the t-TS scheme (**ita** > 0) or non-time-accurate runs (**dt** < 0). A value in the range from 5 to 10 is recommended; however, a smaller value may be required if the sub-iterations do not converge.

3.6 LT6 - Options and Specifications

ngrid – number of grids input = abs(**ngrid**)

If	*Then*
ngrid > 0	CFL3D grid format
ngrid < 0	PLOT3D (or TLNS3D) grid format

(See "Grid File" on page 65.)

nplot3d – number of flow-field data sets to be output in PLOT3D format
Notes
(1) If **nplot3d** < 0, then the PLOT3D files are automatically set to include all solid surfaces (no field points) for 3-d cases or all field points for 2-d cases. Any negative number will give the same result.
(2) If **nplot3d** < 0, then you must include abs(**nplot3d**) PLOT3D data lines in the PLOT3D input section below ("LT28 - PLOT3D Output Specifications (extra comment line needed prior)").

nprint – number of data sets to be sent to an output file
Notes
(1) If **nprint** < 0, then the printout file is automatically set to include all solid surfaces (no field points) for 3-d cases or all field points for 2-d cases. Any negative number will give the same result.
(2) If **nprint** < 0, then you must include abs(**nprint**) data lines in the print out input section below ("LT30 - Print Out Specifications (extra comment line needed prior)").

nwrest – number of iterations between updates of the binary restart file (Note that the binary restart file is also updated at the last iteration of the run regardless of the value of **nwrest**.) Must be > 0. Used to save an earlier restart file in case the full run does not finish. Generally, set **nwrest** > **ncyc** (or > **ntstep** for time-accurate) unless it is suspected that the run may not finish, or for a margin of safety. Restart file is overwritten each time it is updated.

ichk – checks for negative densities and/or pressures in subroutines `gfluxr`, `hfluxr`, `ffluxr`, and `conu` if set equal to 1.

Note

(1) Checking for negative densities and/or pressures requires additional CPU time, so utilize *only* to diagnose problem cases.

i2d – 2-d case flag

If	*Then*
i2d = 0	3-d case
i2d = 1	2-d case
i2d = –1	2-d case plus far-field point-vortex correction for **bctype** = 1003

Notes

(1) For **i2d** 0, *must* set **idim** = 2.

(2) For **i2d** 0, the $i = 1$ and $i = 2$ planes *must* each be planar.

(3) For **i2d** = -1, the grid *must* be the $x - z$ plane; the point vortex is applied at (**xmc**, **zmc**).

ntstep – number of cycles for time-accurate computations (**dt** > 0)

Notes

(1) For time-accurate computations, **ncyc** controls the number of sub-iterations.

(2) For steady-state computations (**dt** < 0), **ntstep** defaults to 1 and **ncyc** controls the number of cycles.

ita – order of time-accuracy/two-time scheme flag (used only for **dt** > 0)

If	*Then*
ita = +1	first order in time; physical time term only (t-TS method)
ita = +2	second order in time; physical time term only (t-TS method)
ita = –1	first order in time; physical *and* pseudo time terms (τ-TS method)
ita = –2	second order in time; physical *and* pseudo time terms (τ-TS method) (recommended)

Notes

(1) The approximate factorization scheme used to advance the solution in time introduces first order errors in time. Furthermore, if the diagonal version is utilized (**idiag** = 1), additional errors of order Δt are introduced. Sub-iterations can be used to drive these factorization/diagonalization errors to zero. Therefore, if a formally second-order (in time) solution is desired, sub-iterations *must* be used; the run will terminate if |**ita**| > 1 and **ncyc** = 1.

(2) The overhead for second order verses first order is relatively small when compared to the gain in accuracy and is therefore recommended.

(3) The inclusion of a pseudo time term increases (often dramatically) the maximum allowable time step one can take for a particular problem. However, sub-iterations (**ncyc** > 1) are therefore mandatory and multigrid is recommended. Also, note that larger time steps imply greater temporal error; so again, second order is recommended.

3.7 LT7 - Grid, Force, and Viscous Options

(Data for Line Type Seven should be repeated **ngrid** times.)

ncg – number of coarser grids to construct for multigrid and/or mesh sequencing (Set **ncg** = 0 for an embedded mesh.)

Note

(1) With the exception of embedded grids, all grids should have the same value of **ncg**.

iem – embedded mesh flag

If	*Then*
iem = 0	global grid
iem = **l**	level number (**l**) of this embedded grid above the global grid level

iadvance – flag to initiate residual/update calculations

If	*Then*
iadvance 0	evaluate the residual and update the solution in the current block
iadvance < 0	skip the residual/update calculations for the current block (rarely used)

iforce – flag to initiate the calculation of forces on block faces with solid surfaces; **iforce** is a 3-digit number of the form IJK:

Where	*Initiates*
I = 0	no force calculations on the i faces
I = 1	calculation of the force contribution from the i = 1 face
I = 2	calculation of the force contribution from the i = **idim** face
I = 3	calculation of the force contributions from both the i = 1 face *and* the i = **idim** face
J = 0	no force calculations on the j faces
J = 1	calculation of the force contribution from the j = 1 face
J = 2	calculation of the force contribution from the j = **jdim** face
J = 3	calculation of the force contributions from both the j = 1 face *and* the j = **jdim** face
K = 0	no force calculations on the k faces
K = 1	calculation of the force contribution from the k = 1 face

K = 2 calculation of the force contribution from the k = **kdim** face

K = 3 calculation of the force contributions from both the k = 1 face *and* the k = **kdim** face

Notes

(1) *Only* solid surfaces contribute to the force computations; i.e. only those boundaries with boundary condition types 1005 and 2004 can contribute to the force totals. Thus, wakes (with boundary condition type 0) are not computed in the force total, regardless of whether or not they are on a boundary that has been flagged with **iforce** > 0. Note that if an overlapped (Chimera) grid is used and there is grid overlap **on** solid surfaces, then forces are double-counted at the overlap. If possible, use **segment** < 0 (see note (2)) to remedy this.

(2) The **iforce** parameter indicates the calculations of force contributions from entire faces; i.e. **iforce** 0 will cause *all* segments on that particular face to contribute to the force total (assuming those segments have solid-surface boundary conditions, as discussed previously). To eliminate from the force computation a segment that would otherwise contribute to the force total, set **segment** < 0 in "LT14 - I0 Boundary Condition Specification" on page 32 through "LT19 - KDIM Boundary Condition Specification" on page 35. This will eliminate that particular segment from the total. This feature is useful, for example, when the fuselage and sting are both on the k = 1 boundary of grid 1. The force contribution from the fuselage is desired, but not the contribution from the sting. In this case, let the k = 1 boundary be defined by two segments (the first for the fuselage and the second for the sting). Set **iforce** = 001 for grid 1 and then set **segment** = –2 for the second segment in "LT18 - K0 Boundary Condition Specification", e.g.:

```
K0: GRID SEGMENT BCTYPE   ISTA   IEND   JSTA   JEND  NDATA
       1       1   2004      1     97      0      0      2
       1      -2   2004     97    129      0      0      2
```

ivisc(m) – viscous/inviscid surface flag with m = 1, 2, 3 $\Rightarrow m = i, j, k$

If	Then
ivisc = 0	inviscid
ivisc = 1	laminar
ivisc = 2	turbulent Baldwin-Lomax model
ivisc = 3	turbulent Baldwin-Lomax with Degani-Schiff modification
ivisc = 4	turbulent Baldwin-Barth model
ivisc = 5	turbulent Spalart-Allmaras model
ivisc = 6	turbulent Wilcox $k - \omega$ model

ivisc = 7 turbulent Menter's $k-\omega$ SST model

ivisc = 8 turbulent $k-\omega$ Explicit Algebraic Stress Model Gatski–Speziale (EASM Gatski–Speziale) in linear formulation

ivisc = 9 turbulent $k-\varepsilon$ EASM Girimaji in linear formulation

ivisc = 10 turbulent Abid $k-\varepsilon$ model

ivisc = 11 turbulent $k-\varepsilon$ EASM Gatski–Speziale *Nonlinear*

ivisc = 12 turbulent $k-\omega$ EASM Gatski–Speziale *Nonlinear*

ivisc = 13 turbulent $k-\varepsilon$ EASM Girimaji *Nonlinear*

Notes

(1) If **ivisc**(m) < 0 on input, a wall function is employed. This option should only (officially) be used for attached flow, when the first grid point off the wall is not in the sub-layer ($y^+ > 10$ or more). The wall function with the Baldwin-Lomax model (**ivisc** = –2 or –3) can be particularly non-robust and, although it may sometimes work well, it is not recommended in general.

(2) The thin-layer viscous terms (laminar or turbulent) can be included in the j, k, or i directions, either separately or combined. Cross-derivative terms are not included. The exception is for turbulent flow using the Baldwin-Lomax model; terms can be included simultaneously in, at most, two directions, either $j-k$ or $i-k$, for any particular grid.

(3) Unlike previous versions of the code, the Baldwin-Lomax model can now be applied at any boundary or boundaries; i.e., the walls may now be at $j/k/i = 1$ and/or $j/k/i =$ **jdim/kdim/idim**.

(4) For multiple-zone applications, be aware that the Baldwin-Lomax model is *not* a field-equation model and it relies on distances to $j/k/i = 1$ and/or $j/k/i =$ **jdim/kdim/idim** walls *in a given zone*. If a particular zone does not contain walls but it is still desired that **ivisc** = 2 or 3 there, the model will default to using a length scale associated with a distance to the nearest $j/k/i = 1$ face *in that zone*. Another option is to set **ivisc** = 0 or 1 in the zones with no walls, but this may not be viable if the zone in question is very near to a wall(s) in other zones.

(5) For the reasons listed above, the field-equation modes 4 and up, which use the very general minimum distance function (which finds the minimum distance to the nearest wall, regardless of what zone it is in) are generally preferred, especially in multi-zone applications. However, the Baldwin-Lomax model (with the Degani-Schiff option) still seems to be the best model for vortical flows.

(6) It is preferable to let k be the primary viscous direction and i be the secondary viscous direction. See discussion in Section 5.1.1. If i is used as a primary viscous direction, it may be necessary to switch the order of inversions in subroutine `af3f` (e.g. from j,k,i to j,i,k). Keep in mind that, unlike i and k, altering the location of the j inversion in the approximate factorization sequence requires substantial recoding.

(7) The minimum distance function `smin` is computed from viscous walls *only*. If a wall is inviscid, then as far as `smin` is concerned, it is invisible. This is important to remember when viscous boundary conditions are turned on after running a case inviscidly for some time since `smin` may never have been computed!

Caution

The EASM models are currently very preliminary and should be considered as "researcher-oriented" as opposed to "production-oriented". Due to the sensitive nature of the variable coefficient `cmu`, the models are generally less robust than other models (particularly for 3-d cases and particularly for the nonlinear versions). It is recommended that EASM solutions be restarted from previously converged $k-\omega$ or $k-\varepsilon$ solutions for this reason. Even then, certain cases may *still* experience trouble! However, preliminary tests indicate that the Girimaji EASM $k-\varepsilon$ model (**ivisc** = 13) tends to be the most robust of the EASM models.

3.8 LT8 - Grid Dimensions

(Data for Line Type Eight should be repeated **ngrid** times.)

idim – number of grid points in the i direction (must be 2 for |**i2d**| =1)

jdim – number of grid points in the j direction

kdim – number of grid points in the k direction

3.9 LT9 - Laminar Region Specification

(Data for Line Type Nine should be repeated **ngrid** times.)

ilamlo – lower i grid point index defining the laminar region of the current block

ilamhi – upper i grid point index defining the laminar region of the current block

jlamlo – lower j grid point index defining the laminar region of the current block

jlamhi – upper j grid point index defining the laminar region of the current block

klamlo – lower k grid point index defining the laminar region of the current block

klamhi – upper k grid point index defining the laminar region of the current block

Notes

(1) These parameters are used for simulating transition only if **ivisc** > 1. Set **ilamlo, jlamlo, klamlo** = 0 for fully turbulent flow. Currently, only one transition region per grid is allowed.

(2) When **ilamlo**, etc. are used, they define a region inside which the production terms in the turbulence model equations (for **ivisc** > 3) are turned "off" (for the Baldwin-Lomax model, the eddy viscosity is merely zeroed out). The flow usually remains laminar inside that region as a result. In contrast, "fully turbulent" (**ilamlo**, etc. = 0) means that the production terms in the turbulence model equations are "on" everywhere. However, in practice, one might see behavior which indicates that the model is still "transitioning" on its own. In other words, one might see a low "laminar" C_f near the leading edge of an airfoil, transitioning fairly rapidly to a higher "turbulent" C_f, for example. Note that turning off the production term does *not* keep turbulence from convecting into a region, if turbulence exists upstream.

(3) Currently for multigrid, the turbulent viscosity is only computed on the finest level grid, then it is restricted to coarser levels. Therefore, the **ilamlo**, etc. values need not be "good" multigrid numbers when used.

3.10 LT10 - Embedded Mesh Specifications

(Data for Line Type Ten should be repeated **ngrid** times.)

inewg – restart flag for grid (not needed if **irest** = 0)

If	*Then*
inewg = 0	flow-field data is read from the restart file
inewg = 1	flow-field data is initialized by linear interpolation from coarser grid solutions

Notes

(1) The purpose of **inewg** is to allow embedded grids to be added *after* the solution process has begun; that is, if a solution has already been computed, but additional resolution in certain parts of the flow field is desired. In such cases, embedded grids may be appended to the grid file and the case may be restarted. Setting **inewg** = 1 initializes the solution on the new embedded level by interpolating from a solution on a coarser level, thereby providing a reasonable starting solution on the new embedded level (the solution for the new embedded grid is not contained in the restart file at this point).

(2) *Important:* **inewg** should be set to 1 *only for the first restart* after a new embedded level has been added. On subsequent restarts, **inewg** = 0 should be used, since the embedded solution has become part of the restart file and should not be re-initialized.

igridc – connection flag for embedded meshes

If	*Then*
igridc = 0	the current grid is a global mesh (**iem** = 0)
igridc > 0	**igridc** is the grid number to which the embedded mesh (**iem** > 0) connects

is, js, ks – starting indices in the connecting grid for placement of an embedded mesh (set **is** = 0, **js** = 0, **ks** = 0 for global meshes)

ie, je, ke – ending indices in the connecting grid for placement of an embedded mesh (set **ie** = 0, **je** = 0, **ke** = 0 for global meshes)

Note

(1) The embedded meshes must be a regular refinement in all directions of the grid to which it connects. The exception is in the i direction for which the embedded mesh *may* have the same grid spacing as the grid to which it connects if desired (this ensures that **idim** = 2 for all grid levels in 2-d cases, including embedded grid levels).

3.11 LT11 - Matrix Inversion and Flux Limiter

(Data for Line Type Eleven should be repeated **ngrid** times.)

idiag(m) – matrix inversion flag with $m = 1, 2, 3 \Rightarrow m = i, j, k$

If	*Then*
idiag = 0	5×5 block tridiagonal inversions
idiag = 1	scalar tridiagonal inversions

Notes

(1) The scalar tridiagonal inversions are recommended for steady-state computations. They may or may not be appropriate for time-accurate computations. For time-accurate computations with scalar tridiagonal inversions, it is suggested that sub-iterations be used to reduce the diagonalization errors.

(2) When **idiag** = 0, viscous terms are included on the left-hand side of the implicit time advancement scheme in the k direction *only*.

iflim(m) – flux limiter flag with $m = 1, 2, 3 \Rightarrow m = i, j, k$

If	*Then*
iflim = 0	unlimited
iflim = 1	smooth limiter
iflim = 2	min-mod limiter (recommended if limiter is needed and **rkap0** 1/3)
iflim = 3	smooth limiter tuned to $\kappa = 1/3$ with a cut-off to eliminate limiting in regions of small gradients (recommended if limiter is needed and **rkap0** = 1/3)

Notes

(1) Many purely subsonic flows do not require a limiter.

(2) Use of **iflim** = 3 will override the **rkap0** value input and force upwind-biased third order.

3.12 LT12 - Spatial Differencing

(Data for Line Type Twelve should be repeated **ngrid** times.)

ifds(m) – spatial differencing parameter for Euler fluxes with $m = 1, 2, 3 \Rightarrow m = i, j, k$

If	*Then*
ifds = 0	flux-vector splitting
ifds = 1	flux-difference splitting (Roe's scheme) (recommended)

rkap0(m) – spatial differencing parameter for Euler fluxes with $m = 1, 2, 3 \Rightarrow m = i, j, k$

If	*Then*
rkap0 = –1	fully upwind
rkap0 = 0	Frommes' scheme
rkap0 = 1	central
rkap0 = 1/3	upwind-biased third order (recommended)

3.13 LT13 - Boundary Condition Segments

(Data for Line Type Thirteen should be repeated **ngrid** times.)

boundary condition segment flags:

grid – current grid number (must be in order)

nbci0 – number of segments on $i = 0$ boundary

nbcidim – number of segments on $i =$ **idim** boundary

nbcj0 – number of segments on $j = 0$ boundary

nbcjdim – number of segments on $j =$ **jdim** boundary

nbck0 – number of segments on $k = 0$ boundary

nbckdim – number of segments on $k =$ **kdim** boundary

iovrlp – grid-overlapping flag

If	*Then*
iovrlp = 0	no overlapping occurs for the current grid
iovrlp = 1	the current grid receives data from an overlapped grid(s)

Note

(1) The minimum number of segments on any boundary is 1.

3.14 LT14 - I0 Boundary Condition Specification

(Data for Line Type Fourteen should be repeated $\sum_{n=1}^{\mathbf{ngrid}} \mathbf{nbci0}(n)$ times.)

I0 Boundary:

grid – current grid number (must be in order)

segment – current segment (must be in order and the total must equal **nbci0**)

bctype – boundary condition type for the current segment

jsta – j starting grid point location for the current segment

jend – j ending grid point location for the current segment

ksta – k starting grid point location for the current segment

kend – k ending grid point location for the current segment

ndata – number of additional data values required for this boundary condition

Notes

(1) Setting **jsta** = **jend** = 0 and/or **ksta** = **kend** = 0 is a shorthand way of specifying the entire range of j and/or k for this segment.

The following notes pertain to "LT14 - I0 Boundary Condition Specification" on page 32 through "LT19 - KDIM Boundary Condition Specification" on page 35:

(2) If **ndata** > 0, then, following the current line, a header line must appear, followed by a single line with the **ndata** values for this boundary condition.

(3) When multigrid is being used, it is best if all *segment* lengths are also multigridable. This is not a requirement. However, if one-to-one connections are involved, it is possible that the use of non-multigridable segment lengths will result in geometric mismatches on coarser grid levels, causing a "mismatch" message to be written to unit 11 and, most likely, eventual disruption of the execution.

(4) If a particular segment should not be counted in the force total, set **segment** to be negative. (See *Note* (2) under **iforce** in Section 3.7.)

(5) Input values for **bctype** (i.e. boundary condition types currently supported) as follows:

bctype	boundary condition
1000	free stream
1001	general symmetry plane
1002	extrapolation
1003	inflow/outflow
1004	(no longer available, use 2004 instead)
1005	inviscid surface
1008	constant enthalpy and entropy inflow
1011	singular axis – half-plane symmetry
1012	singular – full plane
1013	singular axis – partial plane
2002	specified pressure ratio
2003	inflow with specified total conditions
2004	no-slip wall
2005	periodic in space
2006	set pressure to satisfy the radial equilibrium equation
2007	set all primitive variables
2102	pressure ratio specified as a sinusoidal function of time

Descriptions of the boundary conditions can be found in Chapter 6.

3.15 LT15 - IDIM Boundary Condition Specification

(Data for Line Type Fifteen should be repeated $\sum_{n=1}^{\mathbf{ngrid}} \mathbf{nbcidim}(n)$ times.)

IDIM Boundary:

All entries in this line type are the same as in "LT14 - I0 Boundary Condition Specification" except:

segment – current segment (must be in order and the total must equal **nbcidim**)

3.16 LT16 - J0 Boundary Condition Specification

(Data for Line Type Sixteen should be repeated $\sum_{n=1}^{\mathbf{ngrid}} \mathbf{nbcj0}(n)$ times.)

J0 Boundary:

grid – current grid number (must be in order)

segment – current segment (must be in order and the total must equal **nbcj0**)

bctype – boundary condition type for the current segment

ista – i starting grid point location for the current segment

iend – i ending grid point location for the current segment

ksta – k starting grid point location for the current segment

kend – k ending grid point location for the current segment

ndata – number of additional data values required for this boundary condition

Notes

(1) Setting **ista** = **iend** = 0 and/or **ksta** = **kend** = 0 is a shorthand way of specifying the entire range of i and/or k for this segment.

(2) See general notes in "LT14 - I0 Boundary Condition Specification".

3.17 LT17 - JDIM Boundary Condition Specification

(Data for Line Type Seventeen should be repeated $\sum_{n=1}^{\mathbf{ngrid}} \mathbf{nbcjdim}(n)$ times.)

JDIM Boundary:

All entries in this line type are the same as in "LT16 - J0 Boundary Condition Specification" except:

segment – current segment (must be in order and the total must equal **nbcjdim**)

3.18 LT18 - K0 Boundary Condition Specification

(Data for Line Type Eighteen should be repeated $\sum_{n=1}^{\mathbf{ngrid}} \mathbf{nbck0}(n)$ times.)

K0 Boundary:

grid – current grid number (must be in order)

segment – current segment (must be in order and the total must equal **nbck0**)

bctype – boundary condition type for the current segment

ista – i starting grid point location for the current segment

iend – i ending grid point location for the current segment

jsta – j starting grid point location for the current segment

jend – j ending grid point location for the current segment

ndata – number of additional data values required for this boundary condition

Notes

(1) Setting **ista** = **iend** = 0 and/or **jsta** = **jend** = 0 is a shorthand way of specifying the entire range of i and/or j for this segment.

(2) See general notes in "LT14 - I0 Boundary Condition Specification".

3.19 LT19 - KDIM Boundary Condition Specification

(Data for Line Type Nineteen should be repeated $\sum_{n=1}^{\mathbf{ngrid}} \mathbf{nbckdim}(n)$ times.)

KDIM Boundary:

All entries in this line type are the same as in "LT18 - K0 Boundary Condition Specification" except:

segment – current segment (must be in order and the total must equal **nbckdim**)

3.20 LT20 - Mesh Sequencing and Multigrid

mseq – mesh sequencing flag for global grids (maximum number of grid levels for mesh sequencing is 5)

If	*Then*
mseq = 1	single solution on the finest grid level
mseq = 2	solution on the second finest grid advanced **ncyc**(1) cycles, followed by **ncyc**(2) cycles on the finest grid. The initial solution on the finest grid is obtained by interpolation from the coarser grid solution. If **ncyc**(2) = 0, the computations are terminated on the second finest grid after **ncyc**(1) cycles and the restart file is written for the second finest grid.
mseq > 2	sequencing from coarsest to finest mesh as above

mgflag – multigrid flag

If	*Then*
mgflag = 0	no multigrid
mgflag = 1	multigrid on coarser global meshes
mgflag = 2	multigrid on coarser global meshes and on embedded meshes

iconsf – conservation flag

If	*Then*
iconsf = 0	nonconservative flux treatment for embedded grids
iconsf = 1	conservative flux treatment for embedded grids

mtt – flag for additional iterations on the "up" portion of the multigrid cycle

If	*Then*
mtt = 0	no additional iterations (recommended)
mtt > 0	**mtt** additional iterations

ngam – multigrid cycle flag

If	*Then*
ngam = 1	V-cycle
ngam = 2	W-cycle (not recommended for overlapped grids)

Note

(1) If **mglevg** = 2, **ngam** defaults to 1.

3.21 LT21 - Smoothing

issc – correction smoothing flag

If	*Then*
issc = 0	no correction smoothing (usually recommended, but see note 18 in Chapter 10)
issc = 1	correction smoothing

epsssc(m) – correction smoothing coefficient with $m = 1, 2, 3 \Rightarrow m = i, j, k$; typical values: 0.3, 0.3, 0.3

issr – residual smoothing flag

If	*Then*
issr = 0	no residual smoothing (usually recommended, but see note 18 in Chapter 10)
issr = 1	residual smoothing

epsssr(m) – residual smoothing coefficient with $m = 1, 2, 3 \Rightarrow m = i, j, k$; typical values: 0.3, 0.3, 0.3

3.22 LT22 - Iterations and Multigrid Levels

(Data for Line Type Twenty-Two should be repeated for each sequence 1 through **mseq** (from coarsest to finest).)

ncyc – number of cycles for steady-state computations (**dt** < 0); number of sub-iterations + 1 for time-accurate computations (**dt** > 0)

Notes

(1) Set **ncyc** 1, unless using mesh sequencing and terminating execution on a coarser grid level (see description of **mseq** = 2 for "LT20 - Mesh Sequencing and Multigrid" on page 35, as well as "Mesh Sequencing" on page 134).

(2) For time-accurate computations (**dt** > 0), when **ncyc** = 1, the code is iterating on each time step *once*, which means *no* sub-iterations are used. Using **ncyc** = 1 corresponds to the standard non-sub-iterative time-accurate scheme that was in all previous versions of CFL3D. Similarly, **ncyc** = 2 means that the code is iterating on each time step *twice*, which means sub-iterations are used, since it is *one extra* sub-iteration over the standard non-sub-iterative time-accurate scheme.

(3) For time-accurate computations (**dt** > 0) with multigrid (**mgflag** > 0), at least one sub-iteration must be used (**ncyc** 2).

mglevg – number of grids to use in multigrid cycling for the global meshes

If	*Then*
mglevg = 1	single grid level
mglevg = 2	two grid levels
mglevg = m	m grid levels

nemgl – number of embedded grid levels above the finest global grid

If	*Then*
nemgl = 0	no embedded grids
nemgl = 1	one embedded grid
nemgl = m	m embedded grids

Note

(1) Set **nemgl** = 0 for global grids coarser than the finest global grid.

nitfo1 – number of first order iterations (**nitfo1** = 0 recommended)

3.23 LT23 - Coarse Grid Iterations

(Data for Line Type Twenty-Three should be repeated for each sequence 1 through **mseq** (from coarsest to finest).)

mitL – iterations on level L for each level L from coarsest to finest (**mitL** = 1 recommended, in general, although for some cases 3 2 1 or 2 2 1 can yield better multigrid convergence than 1 1 1 when three levels are used, for example)

3.24 LT 24 - Number of 1-1 Interfaces (extra comment line needed prior)

nbli – number of 1-1 grid-point-connecting block interfaces

3.25 LT25 - 1-1 Blocking Connections

(Data for Line Type Twenty-Five should be repeated **nbli** times.)

number – identifying number to aid the user with "bookkeeping" the 1-1 interfaces in the input file (not used by CFL3D)

grid – grid/block identifier indicating which grid in the common interface is being described in this line of input

ista – starting i limit for 1-1 block interface

jsta – starting j limit for 1-1 block interface

ksta – starting k limit for 1-1 block interface

iend – ending i limit for 1-1 block interface

jend – ending j limit for 1-1 block interface

kend – ending k limit for 1-1 block interface

isva1 – varying index on 1-1 block interface (**isva1** in Line Type Twenty-Five varies with **isva1** in Line Type Twenty-Six)

isva2 – varying index on 1-1 block interface (**isva2** in Line Type Twenty-Five varies with **isva2** in Line Type Twenty-Six)

Notes

(1) The code will check the input connection data by computing the geometric mismatch between both sides of the interface, based on the specified ranges. A true 1-1 interface will have zero (or machine zero) mismatch. Any mismatches larger than ε (where ε is the larger of 10^{-9} and 10 times machine zero) will cause a warning message to be printed out in the main output file (look for the words "`geometric mismatch`". *Always* check the main output file (unit 11) for these words, since the code will only print out the message; it will not stop. A good practice is to make one run with only one iteration, then check the output for "`geometric mismatch`". Small mismatches may be acceptable, but large mismatches [O(1)] most likely indicate that one of the segments on the block boundary has been specified incorrectly.

(2) Any block segments that have 1-1 connectivity must also be input in the boundary condition section ("LT13 - Boundary Condition Segments" on page 31 through "LT19 - KDIM Boundary Condition Specification" on page 35), setting **bctype** = 0.

3.26 LT26 - 1-1 Blocking Connections

(Data for Line Type Twenty-Six should be repeated **nbli** times.)

Line Type Twenty-Six has the same parameter types as Line Type Twenty-Five. It gives the grid/block 1-1 connection information on the opposite side of the interface.

3.27 LT27 - Number of Patched-Grid Interfaces (extra comment line needed prior)

ninter – patched-grid flag (zonal interface along a common surface where points need not match)

If	*Then*
ninter = 0	no patched-grid data is needed (no patched interfaces)
ninter < 0	patched-grid data supplied in a separate file

Notes

(1) In Version 5.0 of CFL3D, patched-grid interpolation data for static patched interfaces (interfaces that do not change with time) must be obtained as a preprocessing step (as for chimera overlapped grids). The code ronnie is designed for this task. Dynamic patched interfaces (such as occur when blocks slide past one another) are computed internally to CFL3D. The input for dynamic patched interfaces is described in "LT41 - Number of Dynamic Patched-Grid Interfaces (extra comment line needed prior)" on page 49 through "LT45 - Displacement and Rotation Amounts" on page 52.

(2) Any block segments that are patched must also be input in the boundary condition section ("LT13 - Boundary Condition Segments" on page 31 through "LT19 - KDIM Boundary Condition Specification" on page 35), setting **bctype** = 0.

3.28 LT28 - PLOT3D Output Specifications (extra comment line needed prior)

(Data for Line Type Twenty-Eight should be repeated **nplot3d** times.)

grid – designated grid number for output

iptype – type of PLOT3D file output

If	*Then*
iptype = 0	grid point type – grid file and **Q** file output
iptype = 1	cell center type – grid file and **Q** file output

iptype = 2 cell center type – grid file and turbulence file output (**ivisc** > 1 only) (The current defaults for output are the production term $\tilde{P} \cdot \tilde{L}_R/(\tilde{u}_\infty)^3$ and the Reynolds stress components $\overline{u'w'}/(\tilde{u}_\infty)^2$, $\overline{u'u'}/(\tilde{u}_\infty)^2$, and $\overline{w'w'}/(\tilde{u}_\infty)^2$ for 2-d; and the same plus Sk/ε for 3-d. (See Ristorcelli[30] for a description of these and other available variables.) These may be modified to output other variables if the user desires. If no turbulence model is being used or if the turbulence model cannot obtain a specific parameter being asked for, a value of zero is output. The "hard wire" occurs in subroutine `plot3t`. Note that the u', v', and w' quantities are aligned with the x, y, and z axes of the grid, respectively.

iptype > 2 cell center type – grid file and PLOT3D function file output

Specific functions currently available:

iptype = 3 minimum distance to nearest viscous wall or directed distance (**ivisc** > 1 only)

iptype = 4 eddy viscosity (**ivisc** > 1 only)

istart – starting location in i direction

iend – ending location in i direction

iinc – increment factor in i direction

jstart – starting location in j direction

jend – ending location in j direction

jinc – increment factor in j direction

kstart – starting location in k direction

kend – ending location in k direction

kinc – increment factor in k direction

Notes

(1) Setting **istart** = **iend** = **iinc** = 0, **jstart** = **jend** = **jinc** = 0, and/or **kstart** = **kend** = **kinc** = 0 is a shorthand way of specifying the entire range of i, j and/or k, respectively.

(2) Grid files are written with a blank array.

(3) All files are in multi-block format (even if there is a single zone).

(4) Files are written as 2-d if **i2d** 0.

(5) Function files contain only one variable.

(6) Files are written out in single precision.

(7) On Cray computers, "uncommenting" the lines

```
c     call asnfile(plt3dg, '-F f77 -N ieee', IER)
```

and

```
c     call asnfile(plt3dq, '-F f77 -N ieee', IER)
```

in module `cbsem.f` will cause the files to be written in IEEE binary. The files may then be transferred directly to a Silicon Graphics workstation and read into FAST or PLOT3D using the unformatted option. If those lines are left as comments, then files are written in Cray native, and must first be "itransed" before being transferred to a Silicon Graphics workstation and read into FAST or PLOT3D using the default (binary) option.

(8) This output will appear in the PLOT3D files (units 3 and 4) named near the top of the input deck.

3.29 LT29 - Movie Option

movie – flag to append periodically to the PLOT3D output files (grid and solution) as the solution progresses. Only one grid file and one solution file are generated. Meant for use with time-accurate runs only (**dt** > 0 and number of total time steps dictated by **ntstep**).

If	*Then*
movie = 0	no output of intermediate solutions (if |**nplot3d**| > 0, then a single solution is written at the end of the run)
movie > 0	output of additional solutions every **movie** iterations (the contents of the augmented PLOT3D files are governed by the parameters under "LT28 - PLOT3D Output Specifications (extra comment line needed prior)" on page 39)
movie < 0	output of the initial flow field at the beginning of the run and output of additional solutions every |**movie**| iterations

Caution

Use with care: PLOT3D files will get very large very quickly! If the complete flow field is output, this is really a viable option only in 2-d. In 3-d, it is generally a viable option only if subsets of the flow field are output (i.e. the surface data). Also, set **nprint** = 0 when **movie** > 0 or large print files will result as well.

3.30 LT30 - Print Out Specifications (extra comment line needed prior)

(Data for Line Type Thirty should be repeated **nprint** times.)

grid – designated grid number for output

iptype – type of flow-field variables to print out

If	*Then*

iptype = 0 grid point type
iptype = 1 cell center type

istart – starting location in i direction

iend – ending location in i direction

iinc – increment factor in i direction

jstart – starting location in j direction

jend – ending location in j direction

jinc – increment factor in j direction

kstart – starting location in k direction

kend – ending location in k direction

kinc – increment factor in k direction

Notes
(1) Setting **istart** = **iend** = **iinc** = 0, **jstart** = **jend** = **jinc** = 0, and/or **kstart** = **kend** = **kinc** = 0 is a shorthand way of specifying the entire range of i, j and/or k.
(2) This output will appear in the printout file (unit 17) named near the top of the input deck.

Section 5.2.2 gives a detailed sample of the contents of the printout file.

3.31 LT31 - Number of Control Surfaces (extra comment line needed prior)

ncs – the number of control surfaces; control surfaces are user-specified grid surfaces through which mass flow, thrust (momentum) force, pressure force, and viscous force may be monitored. Control surfaces *must* be specified if **ihstry** > 0. (See "LT3 - Flow Conditions" on page 19.)

3.32 LT32 - Control Surface Specifications

block – block number of the control surface

istart – starting location in i direction

iend – ending location in i direction

jstart – starting location in j direction

jend – ending location in j direction

kstart – starting location in k direction

kend – ending location in k direction

iwall – control surface type

If	*Then*
iwall = 0	for a flow surface
iwall = 1	for a solid wall

inorm – defines the direction of the control surface outward normal relative to the grid surface normal. This is only required in the calculation of the total forces for all control surfaces to obtain the correct signs.

If	*Then*
inorm = 1	the control volume outward normal is in the same direction as the grid surface normal
inorm = –1	the control volume outward normal is in the opposite direction to the grid surface normal
inorm = 0	do not add this surface into the summation of the total forces

Notes

(1) The starting and ending indices in one of the directions must be identical to define the surface. If they are not, no calculation is performed. In the remaining two directions, setting both the starting and ending indices to zero is a shorthand way of specifying the entire index range.

(2) If all the surfaces have **inorm** = 0, the totals are not calculated.

(3) This output will appear in the printout file named near the top of the input deck; the control surface data appears after the data generated by setting **nprint** > 0.

THE INPUT DECK TERMINATES HERE FOR STEADY-STATE CASES OR TIME-DEPENDENT CASES IN WHICH THE GRID IS STATIONARY (**iunst** = 0)

Line Types Thirty-Three through Forty-Five are required only if **iunst** > 0 (dynamic grid).

3.33 LT33 - Number of Translated Grids (extra comment line needed prior)

ntrans – number of grids undergoing translation motion

Notes

(1) If **ntrans** = 0, only the headers for "LT34 - Translational Reference Length" on page 44 through "LT36 - Maximum Translational Displacements" on page 45 are required (i.e. no numerical values required).

(2) If the moment center also needs to be translated, set **ntrans** = "number of grids undergoing translation + 1". If the moment center is not translated, moments are computed relative to a fixed point in space.

3.34 LT34 - Translational Reference Length

lref – the "grid equivalent" of the dimensional reference length used to define either the reduced frequency needed for **itrans** = 2 or the decay rate needed for **itrans** = 3 (See "LT35 - Translational Information and Velocities" on page 44 below.) For **itrans** 1, any value may be input (it is reset to 1.0 internally).

Note

(1) For example, if the input value of reduced frequency for a sinusoidally plunging wing was defined based on the dimensional chord of the wing and, in the grid, the chord of the wing is 2.0, then set **lref** = 2.0.

3.35 LT35 - Translational Information and Velocities

(Data for Line Type Thirty-Five should be repeated **ntrans** times.)

grid – designated number of grid to be translated

Note

(1) If **grid** = 0, then the following input governs the motion of the moment center; on output, the value of 0 will be replaced by the letters "MC" as a reminder.

itrans – type of translation

If	*Then*
itrans = 0	no translation
itrans = 1	translation with constant speed
itrans = 2	sinusoidal variation of displacement
itrans = 3	smooth increase in displacement, asymptotically reaching a maximum displacement

rfreq – reduced frequency when **itrans** = 2; growth rate to maximum displacement when **itrans** = 3 (set **rfreq** = 0 for **itrans** = 0 or **itrans** = 1)

utrans – translation velocity in x direction when **itrans** = 1; maximum displacement in x direction when **itrans** > 1 (set **utrans** = 0 if there is no x displacement)

vtrans – translation velocity in y direction when **itrans** = 1; maximum displacement in y direction when **itrans** > 1 (set **vtrans** = 0 if there is no y displacement)

wtrans – translation velocity in z direction when **itrans** = 1; maximum displacement in z direction when **itrans** > 1 (set **wtrans** = 0 if there is no z displacement)

Notes

(1) Depending on the type of motion, the nondimensional input parameters listed above have different definitions and it is important to understand the distinction. These multiple definitions allow a range of grid motions with a minimal number of input parameters.

(2) The sign of **utrans**/**vtrans**/**wtrans** governs the direction of the motion: "+" for motion in the increasing coordinate direction, "–" for motion in the decreasing coordinate direction. For **itrans** = 2 (sinusoidal), the sign dictates the "initial" direction of motion.

Caution

If PLOT3D-type grids, where y is "up", are used, it is important to note that the grid is internally treated as if z is "up" and that **utrans**, **vtrans**, and **wtrans** refer to the internal velocities. Thus, if y is "up" in the input grid and translation in the "up" direction is desired, then **wtrans** must be used to set the speed. It is advised that a visualization package such as FAST or PLOT3D be used in the early stages of the computation to verify that the grid is moving in the desired direction.

3.36 LT36 - Maximum Translational Displacements

(Data for Line Type Thirty-Six should be repeated **ntrans** times.)

grid – designated number of grid to be translated

Note

(1) If **grid** = 0, then the following input governs the motion of the moment center; on output, the value of 0 will be replaced by the letters "MC" as a reminder.

dxmax – maximum (absolute) translational displacement in the x direction to be allowed for this grid, measured from the $t = 0$ position; set **dxmax** = 0 if no restriction is required

dymax – maximum (absolute) translational displacement in the y direction to be allowed for this grid, measured from the $t = 0$ position; set **dymax** = 0 if no restriction is required

dzmax – maximum (absolute) translational displacement in the z direction to be allowed for this grid, measured from the $t = 0$ position; set **dzmax** = 0 if no restriction is required

Note

(1) Setting **dxmax**, **dymax**, and/or **dzmax** > 0 will have the following effect: if the grid moves further than |**dxmax**|, |**dymax**|, and/or |**dzmax**| from its position at $t = 0$, the grid will be reset backward the appropriate amount, from where the specified grid motion will start again. The grid will be reset every time it reaches the specified limit. Though each block may have an independent limit, care should be taken to insure that any blocks that must be reset concurrently have the *same* values of **dxmax**, **dymax**, and/or **dzmax**. This resetting feature is used primarily in 2-d turbomachinery applications. For example, in 2-d rotor-stator interaction problems where only a few of an infinite number of rotor and stator blades are actually modeled in the grid system, resetting the grids periodically allows the simulation to be continued for an arbitrarily long period of time with only a limited number of zones. This resetting option is only applicable for constant translational speed (**itrans** = 1).

3.37 LT37 - Number of Rotational Grids (extra comment line needed prior)

nrotat – number of grids undergoing rotational motion

Notes

(1) If **nrotat** = 0, only the headers for "LT38 - Rotational Reference Length" on page 46 through "LT40 - Maximum Rotational Displacements" on page 48 are required (i.e. no numerical values required).

(2) If the moment center also needs to be rotated, set **nrotat** = "number of grids undergoing rotation" + 1.

3.38 LT38 - Rotational Reference Length

lref – the "grid equivalent" of the dimensional reference length used to define either the nondimensional rotation rate needed for **irotat** = 1, the reduced frequency needed for **irotat** = 2, or the growth rate needed for **irotat** = 3 (See "LT39 - Rotational Information and Velocities" below.) For **itrans** 1, any value may be input (it is reset to 1.0 internally).

Note

(1) For example, if the input value of reduced frequency for a sinusoidally pitching wing was defined based on the dimensional chord of the wing and, in the grid, the chord of the wing = 2.0, then set **lref** = 2.0.

3.39 LT39 - Rotational Information and Velocities

(Data for Line Type Thirty-Nine should be repeated **nrotat** times.)

grid – designated number of grid to be rotated

Note

(1) If **grid** = 0, then the following input governs the motion of the moment center; on output, the value of 0 will be replaced by the letters "MC" as a reminder.

irotat – type of rotation

If	*Then*
irotat = 0	no rotation
irotat = 1	rotation with constant angular speed
irotat = 2	sinusoidal variation of angular displacement
irotat = 3	smooth increase in displacement, asymptotically reaching a maximum angular displacement

rfreq – reduced frequency when **irotat** = 2; growth rate to maximum angular displacement when **irotat** = 3 (set **rfreq** = 0 for **irotat** = 0 or **irotat** = 1)

omegax – x component of rotational velocity when **irotat** = 1; maximum angular displacement about the x axis when **irotat** > 1; set **omegax** = 0 if there is no x rotation

omegay – y component of rotational velocity when **irotat** = 1; maximum angular displacement about the y axis when **irotat** > 1; set **omegay** = 0 if there is no y rotation

omegaz – z component of rotational velocity when **irotat** = 1; maximum angular displacement about the z axis when **irotat** > 1; set **omegaz** = 0 if there is no z rotation

xorig – x coordinate of origin of the rotation axis

yorig – y coordinate of origin of the rotation axis

zorig – z coordinate of origin of the rotation axis

Notes

(1) Depending on the type of motion, the nondimensional input parameters listed above have different definitions and it is important to understand the distinction. These multiple definitions allow a range of grid motions with a minimal number of input parameters. (See Section 4.5.2 in the Nondimensionalization chapter for details.)

(2) The sign of **omegax**, **omegay**, and **omegaz** governs the direction of the rotation: "+" for motion in the positive direction, "–" for motion in the negative direction where "positive" and "negative" are dictated by the right-hand rule in which the thumb points in the positive direction of the axis of rotation and the fingers curl in the positive rotational direction. (See "The Right-Hand Rule" on page 67.)

See *Caution* in Section 3.35. It applies here as well.

3.40 LT40 - Maximum Rotational Displacements

(Data for Line Type Forty should be repeated **nrotat** times.)

grid – designated number of grid to be rotated

Note

(1) If **grid** = 0, then the following input governs the motion of the moment center.

dthxmx – maximum (absolute) rotational displacement about the x axis to be allowed for this grid (set **dthxmx** = 0 if no restriction is required)

dthymx – maximum (absolute) rotational displacement about the y axis to be allowed for this grid (set **dthymx** = 0 if no restriction is required)

dthzmx – maximum (absolute) rotational displacement about the z axis to be allowed for this grid (set **dthzmx** = 0 if no restriction is required)

Notes

(1) Setting **dthxmx**, **dthymx**, and/or **dthzmx** > 0 will have the following effect: if the grid rotates more than |**dthxmx**|, |**dthymx**|, and/or |**dthzmx**| from it's position at $t = 0$, the grid will be reset backward the appropriate amount, from where the specified grid motion will start again. The grid will be reset every time it reaches the specified limit. Though each block may have an independent limit, care should be taken to insure that any blocks that must be reset concurrently have the *same* values of **dthxmx**, **dthymx**, and/or **dthzmx**. This resetting feature is used primarily in 3-d turbomachinery applications. For example, in 3-d rotor-stator interaction problems where only a few of the large number of rotor and stator blades are actually modeled in the grid system, resetting the grids periodically allows the simulation to be continued for an arbitrarily long period of time with only a limited number of zones. This resetting option is only applicable for constant rotational speed (**irotat** = 1).

(2) **dthxmx**, **dthymx**, and **dthzmx** are output in degrees.

3.41 LT41 - Number of Dynamic Patched-Grid Interfaces (extra comment line needed prior)

ninter2 – dynamic patched-grid flag (zonal interface along a common interface, with grids in relative motion on either side of the interface)

If	*Then*
ninter2 = 0	no patched-grid data is needed (no patched interfaces)
ninter2 > 0	patched-grid data calculated at every time step

Notes

(1) This type of patched-grid data cannot be read in from a file since it must be recalculated for every time step. If there are interfaces for which static patch data is required, use the standard patch input method (precalculate with ronnie and set **ninter** < 0 in "LT27 - Number of Patched-Grid Interfaces (extra comment line needed prior)" on page 39).

(2) Any block segments that are dynamically patched must also be input in the boundary condition section ("LT13 - Boundary Condition Segments" on page 31 through "LT19 - KDIM Boundary Condition Specification" on page 35), using **bctype** = 0.

(3) Further information on patching can be found in the `input.doc` file for the ronnie preprocessor. It is recommended that some experience be gained with static patching (via ronnie) before attempting dynamic patching with moving zones.

(4) While most of the input for static and dynamic patching is the same, dynamic patching requires a few extra pieces of information, described below.

3.42 LT42 - Dynamic Patched-Grid Specifications

(Data for Line Type Forty-Two should be repeated **ninter2** times.)

int – interpolation number

ifit – type of fit

If	*Then*
ifit = 1	bi-linear fit
ifit = 2	serendipity (degenerate) bi-quadratic fit
ifit = 3	quadratic fit in ξ ; linear fit in η
ifit = 4	linear fit in ξ ; quadratic fit in η

limit – maximum step size (number of cells) to use in the search routine (**limit** = 1 recommended)

itmax – maximum number of iterations allowed to find each "to" cell (**itmax** = maximum grid dimension recommended)

mcxie – flag to indicate whether the $\xi = 0$ boundaries on either side of the patch interface are to be rendered coincident (generally required if $\xi = 0$ corresponds to a viscous surface)

If	*Then*
mcxie = 0	do not render the $\xi = 0$ boundaries coincident
mcxie > 100	render the $\xi = 0$ boundaries coincident
mcxie = 1 to 100	The code will try to decide whether or not the boundaries should be rendered coincident. The larger the value, the greater the number differences between the two boundaries can be before deciding they should not be rendered coincident.

Note

(1) For multiple "from" blocks, in which multiple ξ, η coordinate systems are defined, the proper $\xi = 0$ boundary to consider is the one closest to the $\xi = 0$ boundary on the "to" side.

mceta – flag to indicate whether the $\eta = 0$ boundaries on either side of the patch interface are to be rendered coincident (generally required if $\eta = 0$ corresponds to a viscous surface)

ic0 –flag for C-0 continuous boundaries (*not applicable to dynamically patched interfaces; set* **ic0** = 0)

iorph – flag to allow points which cannot be located by the search routine to be tagged as "orphan" points and to be marked as being "interpolated" from block 0

If	*Then*
iorph = 0	do not allow orphan points
iorph = 1	allow orphan points (not recommended for dynamic patch interfaces at this time)

itoss – flag to allow faster patching for cases in which the interface lies along a constant (or very nearly so) coordinate surface

If	*Then*
itoss = 0	interface is non-planar
itoss = 1	interface is an x = constant plane
itoss = 2	interface is an y = constant plane
itoss = 3	interface is an z = constant plane

Notes

(1) While **itoss** = 0 can always be used, it will slow down the dynamic patching considerably. Thus, if the interface is on a constant coordinate surface, the appropriate value of **itoss** is strongly recommended.

(2) All of the above parameters except **itoss** are also used as input to ronnie.

3.43 LT43 - Dynamic Patching "To" Grid Specifications

(Data for Line Type Forty-Three should be repeated **ninter2** times.)

int – interpolation number

to – a three or four digit number indicating the block number and face of the "to" surface ("to" refers to the block being interpolated)
to = Nmn
where
N is the block number (N may be up to two digits)
m is the coordinate direction; 1 = i, 2 = j, 3 = k
n is the minimum or maximum face; 1 = minimum, 2 = maximum

xie1 – starting index in ξ for which interpolation coefficients will be found on the "to" side of the interface

xie2 – ending index in ξ for which interpolation coefficients will be found on the "to" side of the interface

eta1 – starting index in η for which interpolation coefficients will be found on the "to" side of the interface

eta2 – ending index in η for which interpolation coefficients will be found on the "to" side of the interface

nfb – number of block boundaries which make up the "from" side of the patch surface ("from" refers to the block(s) from which the interpolations are made)

Notes

(1) If the entire range is desired, a shortcut is to set **xie1** = **xie2** = **eta1** = **eta2** = 0.

(2) *Important*: For dynamic patching, additional "ghost" block boundaries are often required. The baseline value of **nfb** is set from the zonal setup at $t = 0$ (as in static patching). However, as soon as the zones start to move past one another, portions of some block faces may not have an opposing block to interpolate from. To provide the patching algorithm with a block to interpolate from, one or more of the "from" blocks that match to the "to" side at $t = 0$ are duplicated and then translated or rotated by an appropriate amount to provide coverage of the "to" side as the zones move past one another. The baseline value of **ntb** must then be increased accordingly.

3.44 LT44 - Dynamic Patching "From" Grid Specifications

(Data for Line Type Forty-Four should be repeated **nfb** times for *each* Line Type Forty-Three.)

from – a three or four digit number indicating the block number and face of the "from" surface ("from" refers to the block(s) from which the interpolations are made)
from = Nmn
where
N is the block number (N may be up to two digits)
m is the coordinate direction; 1 = i, 2 = j, 3 = k
n is the minimum or maximum face; 1 = minimum, 2 = maximum

xie1 – starting index in ξ for which interpolation coefficients will be found on "from" side of interface

xie2 – ending index in ξ for which interpolation coefficients will be found on "from" side of interface

eta1 – starting index in η for which interpolation coefficients will be found on "from" side of interface

eta2 – ending index in η for which interpolation coefficients will be found on "from" side of interface

factj – expansion factor in the ξ direction (not implemented at this time (set **factj** = 0))

factk – expansion factor in the η direction (not implemented at this time (set **factk** = 0))

Note
(1) If the entire range is desired, a shortcut is to input zeros for the beginning and ending indices.

3.45 LT45 - Displacement and Rotation Amounts

(Data for Line Type Forty-Five should be repeated **nfb** times for *each* Line Type Forty-Three.)

dx – amount (in grid units) to displace the "from" block in the x direction

dy – amount (in grid units) to displace the "from" block in the y direction

dz – amount (in grid units) to displace the "from" block in the z direction

dthetx – amount (in degrees) to rotate the "from" block in the x direction

dthety – amount (in degrees) to rotate the "from" block in the y direction

dthetz – amount (in degrees) to rotate the "from" block in the z direction