CFL3D version 6.0 User's Manual for Aeroelastic and Deforming Mesh Simulation

By
R. E. Bartels and R. T. Biedron
Aeroelasticity Branch
and
Computational Simulation and Modeling Branch
NASA Langley Research Center

POC: R.E. Bartels
r.e.bartels@larc.nasa.gov
(757)864-2813

Contents

I. MODIFIED EULER/NAVIER-STOKES EQUATIONS

The finite volume thin layer Navier-Stokes code that forms the basis for this new development is CFL3D version 5.0. The equations are written in a general deforming coordinate system that includes the grid speed flux terms. See R. E. Bartels, "Mesh Strategies for Accurate Computation of Unsteady Spoiler and Aeroelastic Problems," *Journal of Aircraft*, Vol. 37, No. 3, pp. 521-525 for more details. The residual on the right hand side of of the equation is linearized about the most recent subiteration m and the resulting equations are approximately factored. This has required the addition of 20-40 coding lines to each of the inviscid flux subroutines 'gfluxr', 'hfluxr', 'ffluxr' and 'ctime'. Furthermore, the desired first or second order accuracy in time is automatically achieved by making use of the metric time derivatives already being used in the spatial fluxes of conserved flow quantities. Since the volumes at several previous time steps are not stored, very little additional memory and no change in the overall code data structure is required. Using this approach does require that the metric fluxes be recomputed at each subiteration and added to the complete flux equations. This has appeared to result in a modest added computational effort.

The metric fluxes in the new terms are evaluated at cell face centers, and are also differenced in time in a way that is consistent with the temporal differencing of conserved flow variables. This results in consistency between new metric fluxes, the metric time derivatives in the physical fluxes and the integral of $J^{-1}Q_\tau$ over the control volume, and finally in a temporally and spatially consistent solution of the GCL.

I. SUBGRIDS

The use of a "subgrid" is the key to a relatively painless multiblock mesh deformation capability. The method is that due to Hartwich, P. M., and Agrawal, S., "Method for Perturbing Multiblock Patched Grids in Aeroelastic and Design Optimization Applications," AIAA-97-2038-CP, 1997. The subgrid (consisting of "slave verticies" in the references' terminology) is constructed by taking out every n-th point in the input grid. For multiple-block grids, each block has its own subgrid. The value of n may be different for each block and for each of the 3 grid indicies. In the input file these values are called iskip, jskip, and kskip in the i, j, and k directions respectively. Generally, these values should be chosen to be as large as possible
while insuring:

1 The starting and ending points of solid surfaces are preserved in the subgrid.

2. "Breakpoints" (e.g. LE/TE angle changes) on solid surfaces are preserved in the subgrid. (how strongly this rule needs to be enforced is very problem dependent)

3. Blocks that are point-matched remain so at the subgrid level.

3. The values of *iskip*, *jskip* and *kskip* divide evenly into *(idim-1), (jdim-1),* and *(kdim-1),* respectively. This is tantamount to requiring that the outer boundaries be preserved (at some appropriately coarse level) in the subgrid

The default skip settings obtained by setting *nskip=0* attempt to follow these general rules by looking at block dimensions, boundary conditions, and 1-1 interface connections. However, if block boundary conditions are not segmented at "breakpoints" (and in general, this may not be the case), then the default skip values MAY need to be altered.

It is not possible to make very general statements about the required value of the skip parameters, owing to the many possible configurations and layouts of grid blocks. However, several simple examples are given below (under "Skip-value examples") to help illustrate the concepts.

The motion of the points in the subgrid are "slaved" to the motion of the closest solid surface points. Let (xc_i, yc_i, zc_i) be the coordinates of the i-th subgrid point, and let (xs_j, ys_j, zs_j) be the j-th surface point. As a preprocessing step, the distance $|\vec{r}_{ij}|$ from each surface point to the i-th subgrid point is determined according to:

$$|\vec{r}_{ij}| = \sqrt{(xc_i - xs_j)^2 + (yc_i - ys_j)^2 + (zc_i - zs_j)^2}$$

The distance $|\vec{r}_{ij}|$ for the nearest *Nmaster* surface points are stored for each point in the subgrid. Each point in the subgrid is moved to its new position at $tn+1$ from its old position at tn according to

$$xc_i^{n+1} - xc_i^n = \sum W_j D_{ij} (xs_j^{n+1} - xs_j^n) \qquad (1)$$

where the summation runs from $j = 1$ to $j = Nmaster$. Similar expressions are used for for yc_i^{n+1} and zc_i^{n+1}. The damping function $D_{i\,j}$ is given by:

$$D_{ij} = \exp\{-[\min(d_{\max}\ ,\ \beta_1\, dv_{ij}\ /(\varepsilon + dm_j)]\}$$

with

$$dv_{ij} = (xc_i^0 - xs_j^0)^2 + (yc_i^0 - ys_j^0)^2 + (zc_i^0 - zs_j^0)^2$$

$$dm_j = (xs_j^{n+1} - xs_j^n)^2 + (ys_j^{n+1} - ys_j^n)^2 + (zs_j^{n+1} - zs_j^n)^2$$

where the superscript *0* denotes the original, undeformed mesh, ε is a small parameter on the order of machine zero to prevent division by zero for cases where $dm = 0$, and *dmax* is chosen such that *e-dmax* is on the order of machine zero, to prevent numerical underflow for very large values of dv that may occur far from the surface.

β_1 is a user-specified parameter, typically on the order of *0.0001*. Smaller values lead to an increase in the distance away from the surface over which the mesh deforms in response to surface motion; larger values decrease the distance over which the mesh deforms. The weight function W_{ij} is given by

$$W_{ij} = \left|\vec{r}_{ij}\right|_{\min} / \vec{r}_{ij}$$

where $\left|\vec{r}_{ij}\right|_{\min}$ is the minimum value over $j=1,Nmaster$

Once the delta displacements for the subgrid points are determined from Eq. 1, arclength based TFI is used to interpolate these delta displacements along lines connecting the corners of the subgrids. The faces of the subgrids are then filled in via arclength based TFI, and finally the delta displacements interior to the subgrids are determined using arclength based TFI between subgrid faces.

Skip-value examples:

A 2D C-grid around an airfoil, with dimensions *idim x jdim x kdim = 2 x 193 x 41*. The surface of the airfoil is on the $k = 1$ boundary, between $j = 33$ and $j = 161$.

The skip value for the i-direction is a non-issue, since the only value that will divide evenly into *idim - 1* is *iskip = 1*. In the k-direction, the "delta" between the inner and outer boundaries is 40, and is the appropriate value for *kskip*. Note that any value that divides evenly into 40 would also have been a candidate for kskip, but it is always best to choose the maximum.

In the *j*-direction, there are 4 "deltas" to note (although 2 are identical):

1) *32* points lie between the downstream boundary at $j = 1$ and the lower trailing edge point at $j = 33$
2) *128* points lie between the lower TE and the upper TE point at $j = 161$
3) *32* points lie between the upper TE point and the downstream boundary at $j = 192$
4) *192* points (i.e. *jdim - 1*) lie between the boundaries at $j = 1$ and $j = 193$.

The value of *jskip* is chosen as the largest common divisor of *32, 128,* and *192*. Thus, *jskip = 32*. Note that starting at *j = 1* and adding *jskip = 32* gets to the lower TE point, from there, adding *4 x jskip* gets to the upper TE point, and finally adding another jskip points gets to the downstream boundary at *j = 193*, thus statisfying requirements 1) and 3) above.

A 3D C-H grid around a wing with straight LE and TE, with dimensions *idim x jdim x kdim = 65 x 193 x 41.* The surface of the airfoil is on the *k = 1* boundary, between *j = 33* and *j = 161* (the *j* and *k* dimensions and layout are exactly the same as the 2D example above). The symmetry plane lies at *i=1*, and the wing tip at *i=41*.

Only the i-direction differs from the example above, so the jskip and kskip values are the same. In the i-direction, 64 points lie between the symmetry plane and the outer boundary at i = 65, and 40 points lie between the symmetry plane and the wing tip. The largest common divisor of 64 and 40 is 8, so iskip = 8. There are no spanwise breakpoints to take into account owing to the straight LE and TE

A 3D C-H grid around a wing with straight LE and TE, consisting of 8 blocks each of dimension idim x jdim x kdim = 33 x 97 x 21.

This is the same case as above, only split for parallel processing. The k0 boundary conditions for the split-grid input file are as follows:

k0:	grid	segment	bctype	ista	iend	jsta	jend	ndata
	1	1	0	1	33	1	97	0
	2	1	0	1	33	1	97	0
	3	1	0	1	33	1	97	0
	4	1	0	1	33	1	97	0
	5	1	0	1	33	65	97	0
	5	2	1005	1	33	1	65	0
	6	1	1005	1	9	1	65	0
	6	2	0	1	9	65	97	0
	6	3	0	9	33	1	97	0
	7	1	0	9	33	1	97	0
	7	2	0	1	9	1	33	0
	7	3	1005	1	9	33	97	0
	8	1	1005	1	33	33	97	0
	8	2	0	1	33	1	33	0

Focusing on the solid surface segments, with BC type 1005 for this inviscid case, we find the deltas in the *i*-direction of *32* and *8,* of with the largest common divisor is *8.* This also divides evenly into *idim - 1 = 32.* Thus, *iskip = 8.* Again focusing on the solid surface segments, we see that all the deltas in the *j*-direction are *64*. However, this does not divide evenly into *jdim – 1 = 96*; thus, we take *jskip* as the largest common divisor of *64* and *96*, or *jskip = 32.* There are no spanwise breakpoints to take into account owing to the straight LE and TE. In the *k*-direction, the maximum value of *kdim-1 = 20* may be used.

II. SPRING ANALOGY

a. Unique Features of Current Formulation.

This discussion is based on the papers R. E. Bartels, "Mesh Strategies for Accurate Computation of Unsteady Spoiler and Aeroelastic Problems," *Journal of Aircraft*, Vol. 37, No. 3, pp. 521-525 and R. E. Bartels and D. M. Schuster, "A Comparison of Two Navier-Stokes Aeroelastic Methods With BACT Benchmark Experimental Data," *Journal of Guidance and Control*, Vol. 23, No. 5, pp. 1094-1099. *However*, there are *three important* differences between the formulation presented in those papers and the scheme as it is implemented in CFL3D v6.0. *First,* the spring analogy and the near grid boundary motion are now delta formulation rather than based on absolute grid position. This alters the way the grid moves, but was done so that the relative orientation of the original grid is retained. This means that grids at the k=1 boundary do not maintain orthogonality throughout motion. Rather, they roughly maintain the orientation of the original grid. Grids originally orthogonal (near the k=1 boundary) remain more or less orthogonal; grids that are skewed at that boundary remain so. *Secondly,* the near boundary preservation of grid orientation is now performed at *both* the *k = 1* and *k = kdim* boundaries. This means that if the *k=1* boundary is a surface boundary, extrapolation to the surface is done consistently throughout the mesh deformation. This is now also true at the *k=kdim* boundary. If the *k=kdim* boundary is a zonal boundary in the middle of the flow field, orientation of grids near this boundary is retained for consistent interpolation between flow field grid blocks. Note, however, that the spring analogy smoothing as currently implemented does not smooth block or zone boundaries that are within the flow domain. Figure 1 illustrates this scheme for a multiblock grid in which the airfoil surface is at the *k=1* boundary. The *third* difference between the present formulation and that in the second reference above is that the motion of the wake cut no longer bisects the upper and lower trailing edge angles. The wake cut now uses an exponential decay function to return the wake cut location to that at the down stream boundary. This can impact the robustness of the grid scheme due to retention of grid orientation at this boundary. Since the wake cut now returns exponentially to the down stream boundary, there can, for large grid motion such as that for a large pitching airfoil motion, develop a highly acute angle at the trailing edge. If the grid upstream of the trailing edge and the grid down stream of the trailing edge attempt to retain the original orientation of the mesh, as the angle becomes very acute, eventually grids will cross.

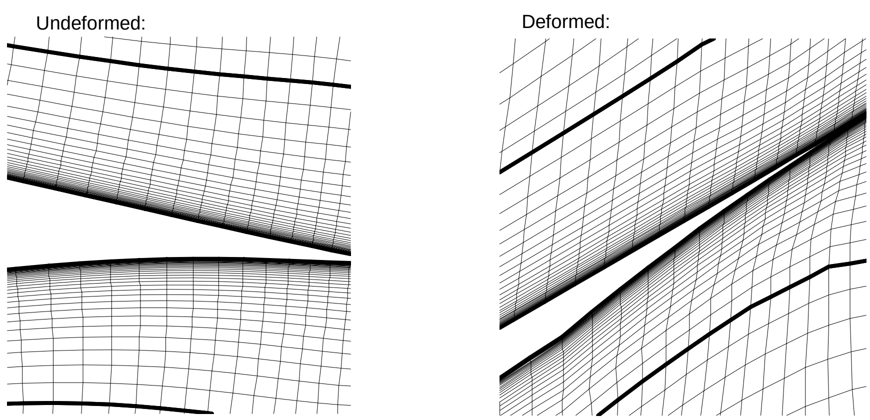

Figure 1. Effect or boundary treatment at *k = 1* and *k = kdim* for a multiblock grid. Darker lines represent block boundaries.

The mesh scheme is a modification of the spring analogy using axial spring stiffness. Spring stiffness in the mesh interior can be controlled by the spacing of the appropriate boundary grid points. For instance, if the nodes at (i,j,k) and (i,j,k+1) are considered and if boundary spacing at boundaries i = 1 and $i = i_{max}$ are used,

$$k_m = 1 / l_{k+1,k} \tag{7}$$

where m=k+1 designates in this case the volume edge between the k and k+1 grid points, and

$$\begin{aligned} l_{k+1,k} &= f(\|\vec{r}_{k+1} - \vec{r}_k\|_2)_{i=1}{}^{-1/2} + \\ &\quad (1-f)(\|\vec{r}_{k+1} - \vec{r}_k\|_2)_{i=i\max}{}^{-1/2} \end{aligned} \tag{8}$$

$$f = (i_{\max} - i) / (i_{\max} - 1) \quad .$$

Since these stiffness values are set at the start of the computations they do not vary in time. They also require storage only for each of the six computational boundaries.

The problem of grid collapse around convex surfaces is handled by selectively increasing/decreasing stiffness based on surface curvature. Stiffness values in two coordinate planes normal to a surface are varied based on surface curvature in the coordinate projection of those planes onto the surface. The final mesh is the weighted combination of the two planar solutions. Take for example a $\xi\eta$-plane solid surface at k=1 from which curvature information is required to construct the grid. Mathematically the expression for interior grid point location can be written

$$\begin{aligned} \vec{\delta}_{ijk} &= (1-\varepsilon)\left[f_1 \sum \bar{k}_{m1} \vec{\delta}_{m1} + f_2 \sum \bar{k}_{m2} \vec{\delta}_{m2} \right] \\ &\quad / \left[f_1 \sum \bar{k}_{m1} + f_2 \sum \bar{k}_{m2} \right] \quad + \quad \varepsilon \Delta P \end{aligned} \tag{9}$$

where

$$f_1 = e^{C_1 \Lambda_\xi} \qquad , \qquad f_2 = e^{C_1 \Lambda_\eta} \quad .$$

Note that equation 9 is now in delta form. In the paper from which this development is taken it was in terms of absolute position. In equation (9) $\bar{k}_{m1} = \frac{1}{2} k_m e^{-C_1 \Lambda_\xi}$ and $\bar{k}_{m2} = \frac{1}{2} k_m e^{-C_1 \Lambda_\eta}$ for the volume edges including the endpoints k+1 and k-1, and $\bar{k}_{m1} = \bar{k}_{m2} = k_m$ otherwise. The index m1 in the summation ranges over the grid points adjacent to ijk in the $\xi\zeta$-plane (indices j,k) and index m2 in the $\eta\zeta$-plane (indices i,k). The constant C_1 is a gain factor that adjusts sensitivity to surface curvature. A value of C_1=20 seems to work well for many geometries. The surface curvature parameters Λ_ξ and Λ_η can be arrived at by considering, for example, the surface grids of Figure 2 in the ξ (j index) direction. A measure that accounts, at grid i,j,k, for convex or concave curvature at the surface point i,j and k=1 is

$$\Lambda_\xi = \frac{\sin\theta}{1-\cos\theta} \quad . \tag{10}$$

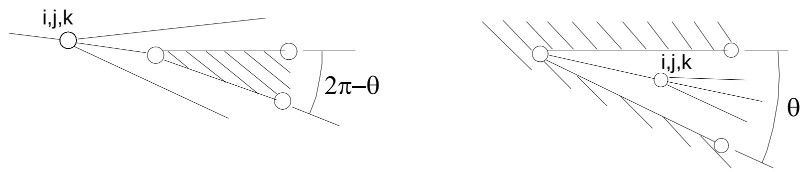

Figure 2. Orientation of surface grids

In terms of grid geometry this is

$$\Lambda_\xi = \frac{\Delta\vec{r}_i \cdot (\Delta\vec{r}_{j+} \times \Delta\vec{r}_{j-})}{\left|\Delta\vec{r}_i\right| \left(\left|\Delta\vec{r}_{j+}\right| \left|\Delta\vec{r}_{j-}\right| - \Delta\vec{r}_{j+} \cdot \Delta\vec{r}_{j-} \right)} \tag{11}$$

where

$$\Delta\vec{r}_i = \vec{r}_{i+1} - \vec{r}_{i-1} \quad , \quad \Delta\vec{r}_j = \vec{r}_{j+1} - \vec{r}_{j-1}, \; \Delta\vec{r}_{j+} = \vec{r}_{j+1} - \vec{r}_j \quad , \quad \Delta\vec{r}_{i+} = \vec{r}_{i+1} - \vec{r}_i \, ,$$
$$\Delta\vec{r}_{i-} = \vec{r}_{i-1} - \vec{r}_i \quad , \text{ Etc.}$$

In practice the values of Λ_ξ and Λ_η are limited in upper and lower bound, currently $\Lambda_\xi=\min(C_1\Lambda_\xi,0)/C_1$ or $\Lambda_\xi=\max(C_1\Lambda_\xi,-12)/C_1$ and similarly for Λ_η. An update of $f_1(\Lambda_\xi)$ and $f_2(\Lambda_\eta)$ are required once per time step for each surface or grid edge plane. This leaves only the computation of equation (9) to solve for the interior of the mesh. This is a linear equation typically requiring 1-2 iterations of a Gauss-Seidel procedure for moderate motions of the grid. Finally, the function P defines an outward normal to the surface. Setting $\varepsilon = \exp(-C_2 k)$ for the *k=1* boundary or $\varepsilon = \exp(-C_2(k\,\mathrm{dim} - k + 1))$ for the *k=kdim* boundary ensures that at least several grid points near those boundaries remain in their original orientation relative to the respective boundaries. The complete mesh solution per time step involves an update using initialization of the mesh via transfinite interpolation (TFI) and a second step involving *ismooth* iterations of equation (9) to arrive at the final mesh.

Example 1: Single-zone, 2D grid with one moving surface, mesh deformation only (iunst=2):

```
moving grid data - deforming surface (forced motion):
 ndefrm
      1
 lref
 1.0000
 grid   idefrm    rfreq   u/omegax    v/omegay    w/omegaz    xorig     yorig     zorig
```

```
  1        2    0.0500   0.0000        10.0000       0.0000     0.2500   0.0000   0.0000
grid    icsi    icsf    jcsi    jcsf    kcsi    kcsf
  1       1      49      25      49      1       1
moving grid data - aeroelastic surface (aeroelastic motion):
naesrf
    0
iaesrf   ngrid   grefl    uinf    qinf   nmodes iskyhook
freq   gmass    damp x0(2n-1)   x0(2n)  gf0(2n)
moddfl     amp    freq     t0
grid    iaei    iaef    jaei    jaef    kaei    kaef
moving grid data - data for field/multiblock mesh movement
nskip   beta1  nmaster  ismooth
    1   0.0001        1          4
grid   iskip   jskip   kskip
   1      1      4      48
moving grid data - multi-motion coupling
ncoupl
   0
slave   master    xorig  yorig zorig
```

The important parameter in this input is *ismooth*. This parameter controls the number of iterations of the spring analogy smoother. If *ismooth = 0*, only TFI motion is performed, with no smoothing. If *ismooth =4* there are four iterations of the spring analogy smoother.

c. Tips on Use

Here are some practical tips on the use of this scheme. First, if the TFI scheme badly mangles the grid, this smoothing step may or may not be able to untangle the grids. For instance, if you have used in improper *jskip* value in the deforming grid input, resulting in very bad grid boundary motion at an airfoil trailing edge, the smoothing step will not smooth the bad boundary motion. In fact the spring analogy smoothing step does not alter any of the grid or zone boundaries. In cases where grids near the surface have moved in a way that results in negative volumes due to the TFI step, the smoothing step may smooth out the grid. If the original grid was of high quality it will have a much better chance of smoothing the grid, since this scheme tends to retain the original orientation of grids near the *k=1* and *k=kdim* boundaries. *The converse is also true!* If the original grid had wild or even moderate oscillations in grids near a viscous boundary layer surface, such that the original grid very nearly had negative volumes, this scheme will very likely aid you in discovering that fact. So, what may happen, if the grid was successfully used in steady state computations, and now it is used for a deforming surface, any imperfections in the grid will more likely become manifest. This fact is also somewhat true, but somewhat less so, for the TFI scheme. In such cases the TFI scheme alone may work better than the TFI + smoothing steps. If a very large (a relative term) time step is used resulting in large motion per time step, more smoothing steps may be required. Experience has suggested that 10-15 iterations (ismooth=10-15) will be usually all that are required for very large time steps, although this is not an ironclad rule. Often 1-3 iterations will noticeably smooth a grid. There are also instances in which the retention of grid orientation near the k=1 or k=kdim boundaries will cause negative volumes. One example was discussed in the first paragraph of this section. Another example is modeling spoiler motion as a backward ramp and step rotating upward. At some point, for most choices of grid spacing, negative volumes will result.

III. DEFORMING MESH PARAMETER DEFINITIONS AND SAMPLE INPUTS

Note that in general there are four sections to the deforming mesh input:

1) deforming surface (forced motion),
2) aeroelastic surface (aeroelastic motion),
3) data for field/multiblock mesh movement, and
4) multi-motion coupling.

Input to the first two sections is optional (except for setting an input control parameter) and can be combined in a variety of ways. It is possible to combine aeroelastic and prescribed deforming surface, aeroelastic and rigid grid motion, prescribed deformation and rigid grid motion, or any of these motions separately. Section 1 is analogous to the forced translation and rotation sections for rigid meshes, except that here one section handles both rotation and translation. The way in which rigid and deforming forced rotation/translation are accomplished are much different. In the rigid motion, the entire grid moves. In the case of deforming motion the moving surface moves and the flow field grid deforms in response. All other boundaries remain fixed. Sections 3 and 4 are required (when *iunst = 2, 3*) whether the mesh deformation occurs via forced motion or aeroelastic motion.

It is not possible to restart a time accurate computation in which the type of deformation is changed. For instance, if a static aeroelastic solution is required proceeding to an aeroelastic solution with deforming or rigid motion superimposed, the static aeroelastic solution must have the deforming/rigid motion invoked but with motion magnitudes set to zero.

For any problem in which the aeroelastic option is not used (which is the focus of this section), the aeroelastic input section should appear as in Example 1 below with *naesrf,* set to zero. That is, after *naesrf* is set to *0*, only header cards are included in the aeroelastic section.

Notes:

1. Deforming grids can only be utilized in time-accurate mode, i.e. $dt > 0$
2. The mesh movement flag, *iunst*, must be set greater than or equal to 2 if deforming meshes are used.
3. Rigid mesh rotation/translation may be used in combination with deforming meshes, at least for simple combinations (e.g all meshes rotate rigidly and all translate with deformation). Set *iunst = 3* to allow rigid and deforming meshes simultaneously. In addition, the parameters in the new "moving grid data - multi-motion coupling" section must be set to provide proper coupling between the rigid and deforming motions.
4. It is *not* possible to restart a deforming/dynamic mesh computation from a previous deforming mesh computation of a different type.
5. As of April 2000, a new input section, "moving grid data - multi-motion coupling", is required. Existing input files for cases with mesh deformation will need to have the following lines appended to the end:

```
moving grid data - multi-motion coupling
ncoupl
    0
slave   master    xorig  yorig zorig
```

This is the default for no coupling of rigid and deforming mesh motions

6. The lines of input shown in the following examples must appear after the control surface data section (line types LT31 through LT32 in the terminology of the Version 5 User Manual)

The following definitions apply:

1) DEFORMING SURFACE SECTION

ndefrm
number of deforming surfaces; if ndefrm = -1, then the code will automatically take all solid surfaces in the mesh to be deforming surfaces. A solid surface is one with one of the following bc types: 1005,1006,2004.

lref
the "grid equivalent" of the dimensional reference length used to define the reduced frequency in the following line. For example, if the input value of the reduced frequency of surface motion was based on the (dimensional) chord of the wing, and in the grid, the chord of the wing is 2, then lref = 2.0.

abs(ndefrm) sets of the parameters grid, idefrm, freq, u/omegax, ...zorig must appear; if ndefrm = -1, then the input value of grid serves merely as a placeholder; and any value may be used, including zero, while the values input for the remaining parameters are applied to all solid surfaces:

grid
grid block containing the moving surface.

idefrm
type of surface oscillatory motion:
= 1 translation
= 2 rotation

freq
reduced frequency of the surface motion

u/omegax
x-component of maximum translational displacement if idefrm = 1
x-component of maximum rotational displacement (Degrees) if idefrm = 2

v/omegay
y-component of maximum translational displacement if idefrm = 1
y-component of maximum rotational displacement (Degrees) if idefrm = 2

w/omegaz
z-component of maximum translational displacement if idefrm = 1
z-component of maximum rotational displacement (Degrees) if idefrm = 2

xorig
x-coordinate of origin of the rotation axis; a value must always be input, even for translation (idefrm = 1)

yorig
y-coordinate of origin of the rotation axis (same comments apply as for xorig)

zorig
z-coordinate of origin of the rotation axis (same comments apply as for xorig)

abs(ndefrm) sets of the parameters grid, icsi, icsf, jcsi, jscf, kcsi, kcsf must appear; if ndefrm = -1, then the input values serve merely as placeholders, and any value may be used, including zero:

grid
grid block containing the moving surface.

icsi
starting index of the deforming surface segment in the i-direction

icsf
ending index of the deforming surface segment in the i-direction

jcsi
starting index of the deforming surface segment in the j-direction

jcsf
ending index of the deforming surface segment in the j-direction

kcsi
starting index of the deforming surface segment in the k-direction

kcsf
ending index of the deforming surface segment in the k-direction

Example 1: Single-zone, 2D grid with one moving surface, mesh deformation only (iunst=2):

```
 moving grid data - deforming surface (forced motion):
  ndefrm
       1
  lref
  1.0000
  grid   idefrm    rfreq   u/omegax    v/omegay    w/omegaz    xorig    yorig    zorig
    1          2     0.0500   0.0000        10.0000        0.0000      0.2500   0.0000   0.0000
  grid    icsi    icsf    jcsi    jcsf    kcsi    kcsf
    1       1      49      25      49       1       1
  moving grid data - aeroelastic surface (aeroelastic motion):
  naesrf
      0
  iaesrf    ngrid    grefl     uinf     qinf   nmodes iskyhook
  freq    gmass     damp x0(2n-1)   x0(2n)  gf0(2n)
  moddfl      amp     freq      t0
  grid    iaei    iaef    jaei    jaef    kaei    kaef
  moving grid data - data for field/multiblock mesh movement
  nskip    beta1  nmaster  ismooth
      1   0.0001        1           0
  grid    iskip    jskip    kskip
```

```
    1       1       4      48
moving grid data - multi-motion coupling
ncoupl
    0
slave   master    xorig  yorig zorig
```

2) FIELD/MULTIBLOCK MESH MOVEMENT SECTION

nskip

number of grid blocks for which you wish to override the default subgrid dimensions. If *nskip = 0,* the code will set the maximum skip values for the particular case at hand. This is the recommended value for a first attempt. Note however, that some situations will require that skip values be lowered from the maximum in order to prevent negative volumes or improve grid quality. If *nskip = -1*, then the skip values for all zones in the grid will be given the user-specified skip values.

beta1

decay parameter for off-body slave point movement. Smaller values cause the surface movement to propagate further into the field; larger values lead to less movement of field points. Values for beta1 are case dependent; try beta1 = +0.0001 as a starting point. *Caution:* Negative values of beta1 trigger the use of transfinite interpolation for points on the body surface, and can lead to a significant alteration of the surface; beta1 < 0 should only be used in desperation to obtain non-negative cell volumes. beta1 > 0 preserves surface fidelity and is therefore recommended.

nmaster

number of master points for each slave points. The motion of slave points are tied to the nmaster nearest surface points, with a weighting factor that varies inversely on the distance from the slave to each master point. Generally, nmaster = 1 is sufficient.

ismooth

number of relaxation sweeps using the spring analogy mesh scheme; *ismooth = 0* results in the TFI mesh movement scheme. The use of *ismooth > 0* tends to preserve any near-surface orthogonality better than the TFI scheme, but may be less robust for poor-quality initial grids and complex geometries.

grid

grid block for which the default subgrid spacing is to be overwritten. If *nskip = -1*, any value for grid is acceptable, since in that case it is merely a placeholder.

iskip

number of points to skip in the i-direction when creating the subgrid. Zero is a shortcut for *idim-1*. For 2D cases the input value in this direction is always ignored and a value of 1 (i.e. *idim-1*) is used.

jskip

number of points to skip in the j-direction when creating the subgrid. Zero is a shortcut for *jdim-1*.

kskip
number of points to skip in the k-direction when creating the subgrid. Zero is a shortcut for *kdim-1*.

3) MULTI-MOTION COUPLING SECTION

ncoupl
number of grid blocks for which you wish to couple mesh deformation to rigid mesh motion, or, to couple two modes of mesh deformation (i.e deforming rotation plus deforming rotation). For cases without coupled motion (the overwhelming majority of cases), set *ncoupl = 0*.

slave
grid number of the slave grid; the slave grid's mesh motion is coupled to the master grid's mesh motion. The slave and master may be the same grid

master
grid number of the master grid; the master grid's mesh motion is influences the slave grid's mesh motion. The slave and master may be the same grid

yorig
y-coordinate of the rotation center of the slave mesh.

yorig
y-coordinate of the rotation center of the slave mesh.

zorig
z-coordinate of the rotation center of the slave mesh.

Example 2: Single-zone, 2D grid with one moving surface, compound mesh deformation (rotation PLUS translation, iunst=2):

Because two modes of deforming mesh motion are used simultaneously, the data under "multi-motion coupling" must be set. In this case the slave grid and master grid are identical. note: examples 2-4 result in identical motion of the airfoil, but the resulting off-surface mesh motions are quite different

```
      .
      .
 control surfaces:
 ncs
   0
 grid  ista  iend  jsta  jend  ksta  kend iwall inorm
moving grid data - deforming surface (forced motion)
 ndefrm
     2
  lref
   1.0
  grid idefrm      rfreqi  omegax  omegay  omegaz  xorig  yorig zorig
     1      2        0.05     0.0  10.000    0.00   1.00    0.0   0.0
     1      1        0.05     0.0   0.000    0.10   1.00    0.0   0.0
  grid   icsi   icsf     jcsi    jcsf     kcsi    kcsf
```

```
     1      1      2       41     217       1      1
     1      1      2       41     217       1      1
moving grid data - aeroelastic surface (aeroelastic motion)
 naesrf
     0
 iaesrf    ngrid    grefl      uinf      qinf    nmodes
    freq   gmass     damp x0(2*n-1)   x0(2*n)   gf0(2*n)
  moddfl     amp     freq       t0
   grid     iaei     iaef     jaei    jaef      kaei      kaef
moving grid data - skip data for field/multiblock mesh movement
  nskip    beta1  nmaster   ismooth
     0   0.0001       1        0
   grid    iskip    jskip    kskip
moving grid data - multi-motion coupling
 ncoupl
     1
  slave   master    xorig  yorig zorig
     1       1          1.      0.      0.
```

Example 3: Single-zone, 2D grid with one moving surface, mesh deformation (rotation about x=1.0) PLUS rigid mesh translation (iunst=3):

Because rigid and deforming mesh motion are used simultaneously, the data under "multi-motion coupling" must be set. In this case the slave grid and master grid are identical. note: examples 2-4 result in identical motion of the airfoil, but the resulting off-surface mesh motions are quite different.

```
        .
       .
  control surfaces:
  ncs
   0
 grid  ista  iend  jsta  jend  ksta  kend iwall inorm
 moving grid data - translation
  ntrans
      1
    lref
  1.0000
 grid itrans     rfreq      xmag      ymag      zmag
    1      2   0.05000   0.00000     0.000   0.10000
 grid     dxmax     dymax     dzmax
    1      0.0000    0.0000    0.0000
 moving grid data - rotation
  nrotat
      0
    lref
 grid irotat    rfreq    thxmag    thymag    thzmag    xorig    yorig    zorig
 grid    thxmax    thymax    thzmax
moving grid data - deforming surface (forced motion)
 ndefrm
     1
   lref
    1.0
   grid idefrm      rfreqi   omegax  omegay  omegaz  xorig  yorig zorig
     1      2           0.05     0.0       10.000       0.00      1.00    0.0    0.0
```

```
  grid   icsi   icsf     jcsi    jcsf     kcsi    kcsf
     1      1       2        41     217       1        1
moving grid data - aeroelastic surface (aeroelastic motion)
 naesrf
     0
 iaesrf    ngrid    grefl      uinf      qinf    nmodes
   freq   gmass     damp x0(2*n-1)   x0(2*n)   gf0(2*n)
  moddfl     amp      freq        t0
  grid     iaei     iaef     jaei    jaef      kaei      kaef
moving grid data - skip data for field/multiblock mesh movement
  nskip    beta1  nmaster   ismooth
     0     0.0001        1            0
  grid    iskip    jskip    kskip
moving grid data - multi-motion coupling
 ncoupl
     1
  slave   master    xorig  yorig zorig
     1             1      1.       0.       0.
```

Example 4: Single-zone, 2D grid with one moving surface, mesh deformation (translation) PLUS rigid mesh rotation about x=1.0 (iunst=3):

Because rigid and deforming mesh motion are used simultaneously, the data under "multi-motion coupling" must be set. In this case the slave grid and master grid are identical. note: examples 2-4 result in identical motion of the airfoil, but the resulting off-surface mesh motions are quite different.

```
        .
       .
  control surfaces:
  ncs
   0
 grid  ista  iend  jsta  jend  ksta  kend iwall inorm
 moving grid data - translation
  ntrans
      0
    lref
 grid itrans     rfreq      xmag      ymag      zmag
 grid     dxmax     dymax     dzmax
 moving grid data - rotation
  nrotat
      1
    lref
  1.0000
 grid irotat     rfreq    thxmag    thymag    thzmag    xorig     yorig     zorig
    1      2   0.05000   0.00000  10.00000   0.00000  1.0000  0.0000  0.0000
 grid    thxmax    thymax    thzmax
    1      0.0000     0.0000     0.0000
moving grid data - deforming surface (forced motion)
 ndefrm
     1
   lref
    1.0
```

```
  grid idefrm   rfreqi  omegax  omegay  omegaz  xorig  yorig zorig
    1     1       0.05     0.0       0.000      0.10      1.00    0.0     0.0
  grid   icsi   icsf     jcsi    jcsf     kcsi    kcsf
    1     1       2       41     217       1       1
moving grid data - aeroelastic surface (aeroelastic motion)
 naesrf
    0
 iaesrf    ngrid    grefl      uinf      qinf    nmodes
   freq   gmass     damp x0(2*n-1)   x0(2*n)   gf0(2*n)
  moddfl     amp     freq       t0
  grid     iaei     iaef     jaei    jaef      kaei      kaef
moving grid data - skip data for field/multiblock mesh movement
  nskip    beta1  nmaster   ismooth
    0     0.0001       1           0
  grid    iskip    jskip    kskip
moving grid data - multi-motion coupling
 ncoupl
    1
  slave   master    xorig  yorig zorig
    1       1           1.       0.      0.
```

Example 5: 16-zone, 2D grid; only zones 10-15 have rotating solid surfaces; no input shortcuts are used; mesh deformation only (iunst=2):

```
       .
       .
 grid  ista  iend  jsta  jend  ksta  kend iwall inorm
 moving grid data - deforming surface (forced motion):
 ndefrm
    6
  lref
   1.0
   grid   idefrm    rfreq u/omegax v/omegay w/omegaz    xorig    yorig    zorig
    10        2        0.05      0.0        10.000        0.0        0.25      0.0      0.0
    11        2        0.05      0.0        10.000        0.0        0.25      0.0      0.0
    12        2        0.05      0.0        10.000        0.0        0.25      0.0      0.0
    13        2        0.05      0.0        10.000        0.0        0.25      0.0      0.0
    14        2        0.05      0.0        10.000        0.0        0. 25      0.0      0.0
    15        2        0.05      0.0        10.000        0.0        0. 25      0.0      0.0
  grid     icsi     icsf     jcsi     jcsf     kcsi     kcsf
    10        1       2       1       25       1       1
    11        1       2       1       33       1       1
    12        1       2       1       33       1       1
    13        1       2       1       33       1       1
    14        1       2       1       33       1       1
    15        1       2       9       33       1       1
 moving grid data - aeroelastic surface (aeroelastic motion):
  naesrf
     0
  iaesrf    ngrid    grefl     uinf     qinf   nmodes iskyhook
    freq    gmass     damp x0(2n-1)   x0(2n)  gf0(2n)
  moddfl      amp     freq       t0
   grid     iaei     iaef     jaei     jaef     kaei     kaef
 moving grid data - data for field/multiblock mesh movement
   nskip    beta1  nmaster  ismooth
     16   0.0001       1       0
```

```
   grid    iskip    jskip    kskip
     1        1       32       48
     2        1        8       48
     3        1       32       48
     4        1       32       48
     5        1       32       48
     6        1       32       48
     7        1        8       48
     8        1       32       48
     9        1       32       48
    10        1        8       48
    11        1       32       48
    12        1       32       48
    13        1       32       48
    14        1       32       48
    15        1        8       48
    16        1       32       48
moving grid data - multi-motion coupling
 ncoupl
     0
  slave   master    xorig  yorig zorig
```

Example 6: The same 16-zone, 2D grid as example 5, but using input shortcuts:

```
      .
      .
      .
  control surfaces:
  ncs
    0
 grid  ista  iend  jsta  jend  ksta  kend iwall inorm
 moving grid data - deforming surface (forced motion):
 ndefrm
    -1
   lref
    1.0
   grid    idefrm  rfreq u/omegax v/omegay w/omegaz    xorig    yorig    zorig
     0        2      0.05       0.0       10.000        0.0        0.25      0.0       0.0
   grid     icsi     icsf     jcsi     jcsf     kcsi     kcsf
     0        0        0        0        0        0        0
 moving grid data - aeroelastic surface (aeroelastic motion):
  naesrf
     0
  iaesrf    ngrid    grefl     uinf     qinf   nmodes iskyhook
    freq    gmass     damp x0(2n-1)   x0(2n)  gf0(2n)
  moddfl      amp     freq       t0
    grid     iaei     iaef     jaei     jaef     kaei     kaef
 moving grid data - data for field/multiblock mesh movement
   nskip    beta1  nmaster  ismooth
     0   0.0001       1        0
    grid    iskip    jskip    kskip
moving grid data - multi-motion coupling
 ncoupl
     0
  slave   master    xorig  yorig zorig
```

Example 7: 16-zone, 2D grid; zones 10-15 have rotating PLUS translating solid surfaces (iunst=2): In this case of dual-mode mesh deformation, it is sufficient to couple the motion of all blocks to the motion of the first block with a deforming surface segment.

```
       .
       .
  control surfaces:
  ncs
    0
 grid  ista  iend  jsta  jend  ksta  kend iwall inorm
moving grid data - deforming surface (forced motion)
 ndefrm
    12
   lref
   1.0
   grid idefrm    rfreqi  omegax  omegay  omegaz  xorig  yorig zorig
    10    1        0.05    0.0     0.000    0.10   1.00   0.0   0.0
    10    2        0.05    0.0    10.000    0.00   1.00   0.0   0.0
    11    1        0.05    0.0     0.000    0.10   1.00   0.0   0.0
    11    2        0.05    0.0    10.000    0.00   1.00   0.0   0.0
    12    1        0.05    0.0     0.000    0.10   1.00   0.0   0.0
    12    2        0.05    0.0    10.000    0.00   1.00   0.0   0.0
    13    1        0.05    0.0     0.000    0.10   1.00   0.0   0.0
    13    2        0.05    0.0    10.000    0.00   1.00   0.0   0.0
    14    1        0.05    0.0     0.000    0.10   1.00   0.0   0.0
    14    2        0.05    0.0    10.000    0.00   1.00   0.0   0.0
    15    1        0.05    0.0     0.000    0.10   1.00   0.0   0.0
    15    2        0.05    0.0    10.000    0.00   1.00   0.0   0.0
   grid  icsi  icsf   jcsi   jcsf   kcsi   kcsf
    10    1     2      1     25      1      1
    10    1     2      1     25      1      1
    11    1     2      1     33      1      1
    11    1     2      1     33      1      1
    12    1     2      1     33      1      1
    12    1     2      1     33      1      1
    13    1     2      1     33      1      1
    13    1     2      1     33      1      1
    14    1     2      1     33      1      1
    14    1     2      1     33      1      1
    15    1     2      9     33      1      1
    15    1     2      9     33      1      1
 moving grid data - aeroelastic surface (aeroelastic motion)
  naesrf
     0
  iaesrf   ngrid   grefl    uinf    qinf   nmodes   iskyhk
    freq  gmass    damp x0(2*n-1)  x0(2*n)  gf0(2*n)
   moddfl    amp    freq     t0
   grid    iaei    iaef    jaei   jaef    kaei    kaef
 moving grid data - skip data for field/multiblock mesh movement
   nskip   beta1  nmaster  ismooth
     0  0.0001      1       0
   grid   iskip   jskip   kskip
 moving grid data - multi-motion coupling
  ncoupl
    16
```

slave	master	xorig	yorig	zorig
1	10	1.	0.	0.
2	10	1.	0.	0.
3	10	1.	0.	0.
4	10	1.	0.	0.
5	10	1.	0.	0.
6	10	1.	0.	0.
7	10	1.	0.	0.
8	10	1.	0.	0.
9	10	1.	0.	0.
10	10	1.	0.	0.
11	10	1.	0.	0.
12	10	1.	0.	0.
13	10	1.	0.	0.
14	10	1.	0.	0.
15	10	1.	0.	0.
16	10	1.	0.	0.

Example 8: 16-zone, 2D grid; zones 10-15 have rotating solid surfaces; all zones subject to translation (iunst=3):

```
      .
      .
      .
 control surfaces:
 ncs
   0
grid  ista  iend  jsta  jend  ksta  kend iwall inorm
moving grid data - translation
 ntrans
    16
  lref
 1.0000
grid itrans    rfreq      xmag      ymag      zmag
   1      2   0.05000   0.00000   0.00000   0.10000
   2      2   0.05000   0.00000   0.00000   0.10000
   3      2   0.05000   0.00000   0.00000   0.10000
   4      2   0.05000   0.00000   0.00000   0.10000
   5      2   0.05000   0.00000   0.00000   0.10000
   6      2   0.05000   0.00000   0.00000   0.10000
   7      2   0.05000   0.00000   0.00000   0.10000
   8      2   0.05000   0.00000   0.00000   0.10000
   9      2   0.05000   0.00000   0.00000   0.10000
  10      2   0.05000   0.00000   0.00000   0.10000
  11      2   0.05000   0.00000   0.00000   0.10000
  12      2   0.05000   0.00000   0.00000   0.10000
  13      2   0.05000   0.00000   0.00000   0.10000
  14      2   0.05000   0.00000   0.00000   0.10000
  15      2   0.05000   0.00000   0.00000   0.10000
  16      2   0.05000   0.00000   0.00000   0.10000
grid     dxmax     dymax     dzmax
   1     0.0000    0.0000    0.0000
   2     0.0000    0.0000    0.0000
   3     0.0000    0.0000    0.0000
```

```
   4      0.0000    0.0000     0.0000
   5      0.0000    0.0000     0.0000
   6      0.0000    0.0000     0.0000
   7      0.0000    0.0000     0.0000
   8      0.0000    0.0000     0.0000
   9      0.0000    0.0000     0.0000
  10      0.0000    0.0000     0.0000
  11      0.0000    0.0000     0.0000
  12      0.0000    0.0000     0.0000
  13      0.0000    0.0000     0.0000
  14      0.0000    0.0000     0.0000
  15      0.0000    0.0000     0.0000
  16      0.0000    0.0000     0.0000
 moving grid data - rotation
  nrotat
      0
    lref
 grid irotat    rfreq   thxmag   thymag   thzmag   xorig   yorig   zorig
 grid   thxmax   thymax   thzmax
moving grid data - deforming surface (forced motion)
 ndefrm
     6
   lref
    1.0
   grid idefrm  rfreqi  omegax  omegay  omegaz  xorig  yorig zorig
     10     2      0.05    0.0     10.000    0.00    1.00    0.0    0.0
     11     2      0.05    0.0     10.000    0.00    1.00    0.0    0.0
     12     2      0.05    0.0     10.000    0.00    1.00    0.0    0.0
     13     2      0.05    0.0     10.000    0.00    1.00    0.0    0.0
     14     2      0.05    0.0     10.000    0.00    1.00    0.0    0.0
     15     2      0.05    0.0     10.000    0.00    1.00    0.0    0.0
   grid   icsi   icsf    jcsi   jcsf    kcsi   kcsf
     10     1     2      1     25      1      1
     11     1     2      1     33      1      1
     12     1     2      1     33      1      1
     13     1     2      1     33      1      1
     14     1     2      1     33      1      1
     15     1     2      9     33      1      1
moving grid data - aeroelastic surface (aeroelastic motion)
 naesrf
     0
 iaesrf   ngrid   grefl    uinf    qinf   nmodes   iskyhk
    freq   gmass    damp x0(2*n-1)   x0(2*n)   gf0(2*n)
  moddfl    amp    freq     t0
   grid    iaei    iaef    jaei   jaef    kaei    kaef
moving grid data - skip data for field/multiblock mesh movement
  nskip   beta1   nmaster  ismooth
     0     0.0001      1          0
   grid   iskip   jskip   kskip
moving grid data - multi-motion coupling
 ncoupl
    16
  slave   master    xorig  yorig zorig
     1       10          1.    0.    0.
     2       10          1.    0.    0.
     3       10          1.    0.    0.
```

4	10	1.	0.	0.
5	10	1.	0.	0.
6	10	1.	0.	0.
7	10	1.	0.	0.
8	10	1.	0.	0.
9	10	1.	0.	0.
10	10	1.	0.	0.
11	10	1.	0.	0.
12	10	1.	0.	0.
13	10	1.	0.	0.
14	10	1.	0.	0.
15	10	1.	0.	0.
16	10	1.	0.	0.

IV. AEROELASTIC PARAMETER DEFINITIONS, INPUTS AND OUTPUT

In general the deforming mesh input can be broken up into four sections:

1) deforming surface (forced motion),
2) aeroelastic surface (aeroelastic motion),
3) data for field/multiblock mesh movement, and
4) multi-motion coupling.

Inputs to the first two sections are optional (except for setting an input control parameter). Sections 3 and 4 are required (when *iunst = 2, 3*) whether the mesh deformation occurs via forced motion or aeroelastic motion. It is possible to combine aeroelastic and prescribed deforming surface, aeroelastic and rigid grid motion, prescribed deformation and rigid grid motion, or any of these motions separately. For any problem in which the forced motion is not used, but the aeroelastic response of one or more surfaces is desired, the forced-motion input section should appear as shown in Examples 1-3 to follow. For these cases, *iunst = 2*, and the only numerical value which should appear in the forced motion data section is the value of *ndefrm* which must be set to zero. Only header cards for the remainder of the prescribed deformed motion should then be included. If both prescribed deforming mesh motion (e.g. oscillating spoiler) plus aeroelastic response are desired, set *iunst = 2* and include both data for deforming prescribed motion ($ndefrm \neq 0$) and aeroelastic response ($naesrf \neq 0$). Example 4 presents a case in which both rigid motion of the grid(s) and aeroelastic deformation are involved. In this case *iunst = 3*, and rigid forced and aeroelastic data are required.

When the aeroelastic option is invoked ($naesrf \neq 0$), the aeroelastic mode file *aesurf.dat* must be present in the same directory as the executable. Unlike the input CFD grid and output files specified in the input file, the location of the *aesurf.dat* file cannot be redirected to a parent directory. The format and data ordering of this file will be discussed later in this section. Note that:

1. Aeroelastic analysis can only be utilized in time-accurate mode, i.e. dt > 0 .
2. It is *not* possible to restart a deforming/dynamic mesh computation from a previous deforming mesh computation of a different type.
3. The mesh movement flag is iunst. These values are defined:
   - iunst = 0 (undeforming, fixed grid)
   - iunst = 1 (rigid moving grid, prescribed motion, e.g. pitching, plunging sinusoidal)
   - iunst = 2 (deforming grid, either prescribed *and/or* aeroelastic motion)
   - iunst = 3 (aeroelastic motion + prescribed rigid motion, e.g. dynamic, flexible aircraft flight, still under construction)

Static aeroelastic computations can be performed by:

1. Start either from scratch (irest = 0), or restart, after a steady state computation in which $dt < 0$, $iunst = 0$. Starting from scratch is not recommended.
2. Set $iunst = 2$, $dt > 0$ and $damp = .99999…$ and perform the computation in a time marching manner to convergence.

Flutter onset computations can be performed by:

1. Converging a static solution as outlined above.
2. Setting *damp* to the correct value for the elastic system being modeled.
3. Setting an initial perturbation *x0(2*n)* or *x0(2*n-1)* in the desired mode.

If a restart in the middle of a flutter computation is performed, the initial perturbation values must be reset to zero at the restart. Per the item 2 above under general notes, it is not possible to restart a time accurate computation in which the type of deformation is changed. For instance, if a static aeroelastic solution is required proceeding to an aeroelastic solution with deforming or rigid motion superimposed, the static aeroelastic solution must have the deforming/rigid motion invoked but with motion magnitudes set to zero.

If *iunst=2*, the following lines in the input file must appear after the control surface data section (line types LT31 through LT32 in the terminology of the Version 5 User Manual) in order to enable the aeroelastic option:

naesrf
number of aeroelastic surfaces; each aeroelastic surface is represented as a distinct set of modal shapes. Most typically, *naesrf = 1,* but for example, *naesrf = 2* would allow the wing to be represented with one set of mode shapes, while the tail could be represented with a different set of mode shapes. All of the remaining data is repeated as a block for each of the aeroelastic surfaces from 1 to *naesrf.*

Repeat the card (iaesrf … iskyhk) *naesrf* times:

iaesrf
the aeroelastic surface for which the data is being set.

ngrid
the number of grids which contain the aeroelastic surface iaesrf. If ngrid = -1, then the aeroelastic surface is assumed to be comprised of all solid surfaces (i.e. bc types 1005, 1006 or 2004) within the entire grid system. Thus, *ngrid = -1* is appropriate only for *naesrf = 1*.

grefl
the conversion factor required to relate the length units of the aeroelastic equations to the CFD grid. Defined as $grefl = \sqrt{S_{AE} / S_{CFD}}$ where S_{AE} is the wing plan form area dimensionalized for the aeroelastic equations, and S_{CFD} is the wing plan form area in the CFD grid. See the example setup discussed in the Appendix A. The following examples should also help clarify the meaning.

Example 1: The aeroelastic equations are in dimensional units with an aeroelastic surface chord length of *1.3* feet and the span of 1 foot. The same chord length in the CFD grid has been nondimensionalized to 1 and a span of 1. In this example *grefl = 1.3*. (This corresponds to the example 1 input shown below) *NOTE:* If *only* one dimension changes between the aeroelastic and CFD models (e.g. chord length), one can define $grefl = l_{AE} / l_{CFD}$ where l_{AE} and l_{CFD} are the differing reference lengths for the aeroelastic model and the CFD grid.

Example 2: The aeroelastic equations are in dimensional units with an aeroelastic surface chord length of .3 meters and span of 1 meter. The same length in the CFD grid is also .3 (meters) with a CFD grid span of 1. In this example, therefore, *grefl = 1.*. (This corresponds to the example 2 input shown below)

Example 3: The aeroelastic equations are in dimensional units with an aeroelastic surface chord length of .18 meters and span of 2 meters. These lengths in the CFD grid are 1 for chord and 1 for span. In this example *grefl ~ .6.*

Example 4: Suppose the aeroelastic response of a wing supported at the root is desired. If the CFD grid is dimensioned identical to the aeroelastic equations then *grefl = 1.* If the CFD grid is nondimensionalized or dimensionalized in some way other than the aeroelastic equations, then $grefl = \sqrt{S_{AE} / S_{CFD}}$.

uinf
reference flow speed consistent with units used in the aeroelastic input.

qinf
reference dynamic pressure used to dimensionalize the generalized forces.

nmodes
number of modes used to represent the aeroelastic surface.

iskyhk
flag to indicate the use of the "skyhook" terms. NOT YET IMPLEMENTED.
However, a value must still be input as a placeholder (any value will do).

Repeat the following cards (freq … gf0, *nmode* cards) *naesrf* times:

The card (freq ... gf0) is repeated *nmode* times, with data arranged from 1 to *nmode*:

freq
the wind-off natural frequency of the mode.

gmass
the generalized mass of the mode.

damp
the damping factor of the mode

x0(2*n-1)
the initial generalized displacement of the mode; will override the value in the restart file (if restarting). This allows the mode to be perturbed for excitation of aeroelastic instabilities after a static aeroelastic starting solution has been performed.

x0(2*n)
the initial generalized velocity of the mode; will override the value in the restart file (if restarting). This allows the mode to be perturbed for excitation of aeroelastic instabilities after a static aeroelastic starting solution has been performed.

gf0(2*n)

the generalized force offset to include for the mode. The generalized force offset is included in the following way in the subroutine 'ae_corr':

Total generalized force = qinf*grefl*grefl*(cx+cy+cz) - gf0(n,iaes)

Repeat the following cards (moddfl … t0, *nmode* cards) *naesrf* times:

The card (moddfl ... t0) is repeated *nmode* times, with data arranged from 1 to *nmode*:

moddfl

type of time-varying modal perturbation desired:
= 0, no perturbation
= 1, harmonic (sinusoidal) perturbation
= 2, Gaussian pulse
= 3, step pulse

amp

amplitude of modal perturbation.

freq

reduced frequency of modal perturbation if moddfl = 1
half-width of Gaussian pulse if moddfl = 2
use any value as a placeholder for moddfl = 0,3

t0

time about which Gaussian pulse is centered if moddfl = 2
time at which step pulse starts is centered if moddfl = 3
use any value as a placeholder for moddfl = 0,3

The types of modal pulsed or perturbation motion are coded as follows:

Harmonic perturbation:

```
modal displacement = amp*sin(freq*ainf*time/grefl)
modal velocity     = amp*ainf*freq*cos(freq*ainf*time/grefl)/grefl
```

(Note: ainf = uinf/Mach where uinf is the velocity defined in the aeroelastic input)

Gaussian pulse:

```
dtime              = const*ainf*ainf*(time-t0)**2/grefl/grefl
expterm            = exp(-dtime)
modal displacement = amp*expterm
modal velocity     = -2.*const*(time-t0)*ainf*xs(2*nm-1,iaes)/grefl
```

Step pulse:

```
if (real(time).lt.real(t0-dt/2.)) then
  modal displacement = 0.
  modal velocity     = 0.
else if (real(time).gt.real(t0-dt/2.).and.real(time).lt.real(t0+dt/2.)) then
  modal displacement = amp
  modal velocity     = amp*ainf/grefl/dt
```

```
else
  modal displacement = amp
  modal velocity        = 0.
end if
```

The card (grid ... kaef) is repeated |ngrid| times:

grid
grid containing all or part of the current aeroelastic surface iaesrf.
(use any value as a placeholder if ngrid = -1)

iaei
starting i-index of the current aeroelastic surface in zone grid.
(use any value as a placeholder if ngrid = -1)

iaef
ending i-index of the current aeroelastic surface in zone grid.
(use any value as a placeholder if ngrid = -1)

jaei
starting j-index of the current aeroelastic surface in zone grid.
(use any value as a placeholder if ngrid = -1)

jaef
ending j-index of the current aeroelastic surface in zone grid.
(use any value as a placeholder if ngrid = -1)

kaei
starting k-index of the current aeroelastic surface in zone grid.
(use any value as a placeholder if ngrid = -1)

kaef
ending k-index of the current aeroelastic surface in zone grid.
(use any value as a placeholder if ngrid = -1)

Example 1: Single-zone, 2D grid with one aeroelastic surface:

```
        .
        .
        .
 control surfaces:
 ncs
   0
grid  ista  iend  jsta  jend  ksta  kend iwall inorm
moving grid data - deforming surface (forced motion)
ndefrm
    0
  lref
  grid idefrm      rfreqi  omegax  omegay  omegaz  xorig  yorig zorig
  grid   icsi   icsf     jcsi    jcsf     kcsi    kcsf
```

```
moving grid data - aeroelastic surface (aeroelastic motion)
 naesrf
     1
 iaesrf    ngrid       grefl           uinf       qinf       nmodes    iskyhk
     1       -1          1.3              393.      155.        2              0
   freq   gmass     damp x0(2*n-1)   x0(2*n)   gf0(2*n)
  21.363      1.         0.         0.            0.           0.
  32.421      1.         0.         0.            0.           0.
  moddfl    amp     freq      t0
     0           0.         0.      0.
     0           0.         0.      0.
   grid    iaei     iaef     jaei     jaef      kaei      kaef
     1       0        0        0        0         0          0
moving grid data - skip data for field/multiblock mesh movement
  nskip    beta1    nmaster    ismooth
     1     0.0001        5            0
   grid    iskip    jskip    kskip
     1       0         8          0
 moving grid data - multi-motion coupling
  ncoupl
     0
  slave   master   xorig   yorig   zorig
```

Example 2: 8 zone, 2D grid with one aeroelastic surface. See Appendix A for the derivation of this input based on the data from the paper, "Investigations of an Oscillating Supercritical 2D Wing Section in a Transonic Flow," by A. Knipfer and G. Schewe, AIAA 99-0653.

```
        .
  control surfaces:
  ncs
   0
 grid  ista  iend  jsta  jend  ksta  kend iwall inorm
 moving grid data - deforming surface (forced motion):
  ndefrm
     0
   lref
   grid  idefrm    rfreq u/omegax v/omegay w/omegaz    xorig    yorig    zorig
   grid    icsi     icsf     jcsi     jcsf     kcsi     kcsf
 moving grid data - aeroelastic surface (aeroelastic motion):
  naesrf
     1
  iaesrf    ngrid       grefl       uinf       qinf      nmodes    iskyhook
     1        4        1.0000     254.70    9600.0       2              0
   freq        gmass     damp    x0(2n-1)    x0(2n)     gf0(2n)
 205.4000   1.0000   0.0065     0.0000     0.0000     0.0000
 299.3000   1.0000   0.0052    -0.1000     0.0000     0.0000
  moddfl       amp       freq        t0
     0       0.0000     0.0000    0.0000
     0       0.0000     0.0000    0.0000
 grid     iaei     iaef     jaei     jaef     kaei     kaef
    5       1        2        1       33        1        1
    6       1        2        1       69        1        1
    7       1        2        1       69        1        1
```

```
    8       1       2      37      69       1       1
 moving grid data - data for field/multiblock mesh movement
  nskip     beta1    nmaster    ismooth
    8    0.000100       5            3
   grid   iskip     jskip      kskip
     1       1          4         46
     2       1         68         46
     3       1         68         46
     4       1          4         46
     5       1          4         46
     6       1         68         46
     7       1         68         46
     8       1          4         46
  moving grid data - multi-motion coupling
  ncoupl
     0
  slave   master   xorig   yorig   zorig
```

Example 3: 11 zone, 3D grid with two aeroelastic surfaces. Zones 5-8 correspond to aeroelastic surface 1 and zones 9-10 correspond to aeroelastic surface 2.

```
  control surfaces:
  ncs
   0
 grid  ista  iend  jsta  jend  ksta  kend iwall inorm
 moving grid data - deforming surface (forced motion):
  ndefrm
     0
   lref
   grid  idefrm    rfreq u/omegax v/omegay w/omegaz    xorig    yorig    zorig
   grid    icsi     icsf     jcsi     jcsf     kcsi     kcsf
 moving grid data - aeroelastic surface (aeroelastic motion):
  naesrf
     2
  iaesrf    ngrid      grefl       uinf       qinf      nmodes    iskyhook
     1        4        1.0000     254.70    9600.0       2            0
     2        2        1.0000     254.70    9600.0       2            0
   freq        gmass     damp    x0(2n-1)    x0(2n)     gf0(2n)
 205.4000   1.0000   0.0065     0.0000     0.0000     0.0000
 299.3000   1.0000   0.0052    -0.1000     0.0000     0.0000
 155.8000   1.0000   0.0000     0.0000     0.0000     0.0000
 320.1000   1.0000   0.0000     0.0000     0.0000     0.0000
  moddfl      amp        freq        t0
     0       0.0000     0.0000    0.0000
     0       0.0000     0.0000    0.0000
     0       0.0000     0.0000    0.0000
     0       0.0000     0.0000    0.0000
 grid     iaei      iaef     jaei     jaef    kaei     kaef
    5       1       133       1       33       1        1
    6       1       133       1       69       1        1
    7       1       133       1       69       1        1
    8       1       133      37       69       1        1
```

```
     9      1      43      1      33      1      1
    10      1      43      1      69      1      1
  moving grid data - data for field/multiblock mesh movement
   nskip     beta1    nmaster    ismooth
    11     0.000100      5           3
     grid   iskip    jskip    kskip
       1      1        4        46
       2      1       68        46
       3      1       68        46
       4      1        4        46
       5      1        4        46
       6      1       68        46
       7      1       68        46
       8      1        4        46
       9      1        4        46
      10      1       68        46
      11      1       68        46
  moving grid data - multi-motion coupling
   ncoupl
       0
   slave   master   xorig   yorig   zorig
```

Example 4: 4 zone, 3D grid with one aeroelastic surface coupled with prescribed motion (iunst=3). This input has been set up to do a flexible aircraft trim solution. The center of mass of the aircraft is at x = 320 inches and y = 10 inches. The rigid moving grid (rotation) input is required, and the center of mass must be specified as the center of mass of the unflexed aircraft. This same point must be specified in the multi-motion coupling cards at the end of the input. The mode for which *moddfl = 4* is the mode defining the control surface used in trimming the aircraft. The trim code is activated by inputting the keyword input line '*irbtrim 1'*. The default value for *irbtrim* is *0*.

```
        .
        .
control surfaces:
  ncs
   0
 grid  ista  iend  jsta  jend  ksta  kend iwall inorm
 moving grid data - translation
 nrotat
    0
  lref
  grid   itrans   rfreq    xmag    ymag    zmag
  grid   dxmax   dymax   dzmax
 moving grid data - rotation
 nrotat
   4
    lref
  1.0000
  grid   irotat   rfreq   thxmag   thymag   thzmag   xorig   yorig   zorig
    1       2    0.050    0.000    0.000    0.000    320.0    0.000   10.000
    2       2    0.050    0.000    0.000    0.000    320.0    0.000   10.000
    3       2    0.050    0.000    0.000    0.000    320.0    0.000   10.000
    4       2    0.050    0.000    0.000    0.000    320.0    0.000   10.000
  grid   thxmax   thymax   thzmax
    1    0.000    0.000    0.000
    2    0.000    0.000    0.000
    3    0.000    0.000    0.000
```

```
      4    0.000    0.000    0.000
moving grid data - deforming surface (forced motion)
 ndefrm
    0
   lref
  grid  idefrm    rfreqi omegax omegay omegaz xorig yorig zorig
  grid  icsi icsf jcsi jcsf kcsi kcsf
moving grid data - aeroelastic surface
  naesurf
       1
   iaesrf  ngrid  grefl      uinf      qinf     nmodes  ISKYHK
    1        -1      1.0      15000.     12.        2          0
   freq              gmass        damp   x0(2*n-1)  x0(2*n)  gf0(2*n)
   13.000000     1.000000    .99999         0.           0.          0.
     1.000000     1.000000     .00000        0.           0.          0.
  moddfl    amp     freq      t0
      0          0.        0.        0.
      4          0.        0.        0.
  grid    iaei    iaef    jaei    jaef    kaei    kaef
   -1       0       0        0       0       0        0
moving grid data - skip data for field/multiblock mesh movement
    nskip      beta1   nmaster    ismooth
       0      0.00010        1           0
     grid      iskip    jskip     kskip
moving grid data - multi-motion coupling
 ncoupl
    4
  slave   master    xorig  yorig zorig
    1       1       320.    0.   10.
    2       2       320.    0.   10.
    3       3       320.    0.   10.
    4       4       320.    0.   10.
```

The ordering and format of the modal input file *aesurf.dat* in general, as it is read in by the subroutine 'modread' is as follows:

```
c
c       Loop over the CFD grid zones
c
          do nbl = 1,nblocks
c
c      Loop over the aeroelastic surfaces in each zone
c
                    do iaes = 1,naesrf
c
c      Loop over each of the modes
c
                              do n = 1,nmodes
c
c      Read modal data for j=0 surface
c
                                do ns = 1,nseg
```

```
            do i = ista,iend
              do k = ksta,kend
                read( …,*) xmdj(k,i,1),xmdj(k,i,2),xmdj(k,i,3)
              enddo
            enddo
          enddo
c
c     Read modal data for jdim surface
c
          do ns = 1,nseg
            do i = ista,iend
              do k = ksta,kend
                read( …,*) xmdj(k,i,4),xmdj(k,i,5),xmdj(k,i,6)
              enddo
            enddo
          enddo
c
c     Read modal data for k=0 surface
c
          do ns = 1,nseg
            do i = ista,iend
              do j = jsta,jend
                read( …,*) xmdk(i,j,1),xmdk(i,j,2),xmdk(i,j,3)
              enddo
            enddo
          enddo
c
c     Read modal data for kdim surface
c
          do ns = 1,nseg
            do i = ista,iend
              do j = jsta,jend
                read( …,*) xmdk(i,j,4),xmdk(i,j,5),xmdk(i,j,6)
              enddo
            enddo
          enddo
c
c     Read modal data for i=0 surface
c
          do ns = 1,nseg
            do k = ksta,kend
              do j = jsta,jend
                read( …,*) xmdi(j,k,1),xmdi(j,k,2),xmdi(j,k,3)
              enddo
            enddo
          enddo
c
c     Read modal data for idim surface
c
          do ns = 1,nseg
            do k = ksta,kend
```

```
                do j = jsta,jend
                    read( …,*) xmdi(j,k,4),xmdi(j,k,5),xmdi(j,k,6)
                enddo
              enddo
            enddo
          enddo

      enddo

    enddo
```

The ordering of the grid points *must* exactly correspond to the order of these grid points in the CFD grid file read by CFL3D. If a split CFD grid is used, the order must still correspond to the order of the grids in the CFD grid file. Note that if a multi zonal grid has been created from a single block grid, the ordering of data in this file depends on the ordering of the final split grid not the ordering of the original grid. If the utility 'splitter' is used usually the final ordering will generally *not* correspond to the ordering of the unsplit grid. Ordering of the split grid zones can be found in the *splitter.out* file. From this file the order of the surface grid points for the *aesurf.dat* file can be determined.

Aeroelastic time history output is in the file *genforce.dat*. This file is generated if *iunst = 2* and aeroelastic surfaces are defined in the input file *(naesrf≠0)*. After header information, modal response data for each mode is written. Unlike output data in the cfl3d.subit_res file, a complete time history of this data for the entire simulation is retained and written/read to/from restart files and subsequently output to the *genforce.dat* file. The items contained in each column are:

it
Iteration number (if steady computation) or time step (if time accurate unsteady)

time
Nondimensional time

xs(2*n-1)
Modal displacement

xs(2*n)
Modal displacement velocity

gforcn(2*n)
Generalized force

Time is the CFL3D nondimensional time. Units on the last three items are consistent with the aeroelastic problem formulation. If the aeroelastic surface is in the *xy*-plane, the generalized force *gforcn(2*n)* can be written mathematically as

$$\tilde{G}_{iaesurf,n} = q_\infty \cdot (grefl)^2 \iint_{iaesurf} \phi_n(x,y)\; p(x,y)\,dx\,dy$$

where x and y are the CFD grid coordinates. The generalized force represents the total force summed over all boundaries of all blocks per mode per aeroelastic surface.

APPENDIX A. DERIVATION AND SETUP OF AN EXAMPLE AEROELASTIC INPUT

The following development of the equations of motion may help clarify the input definition. It is the test set up used in the experiment discussed in the paper, "Investigations of an Oscillating Supercritical 2D Wing Section in a Transonic Flow," by A. Knipfer and G. Schewe, AIAA 99-0653. This example is the basis for the input defined in example 2 of Section IV above. The aeroelastic equations are maintained in dimensional form. The CFD grid has also been generated based on the airfoil dimensions. That is, in the CFD grid the airfoil chord is .3 and the grid span is 1.

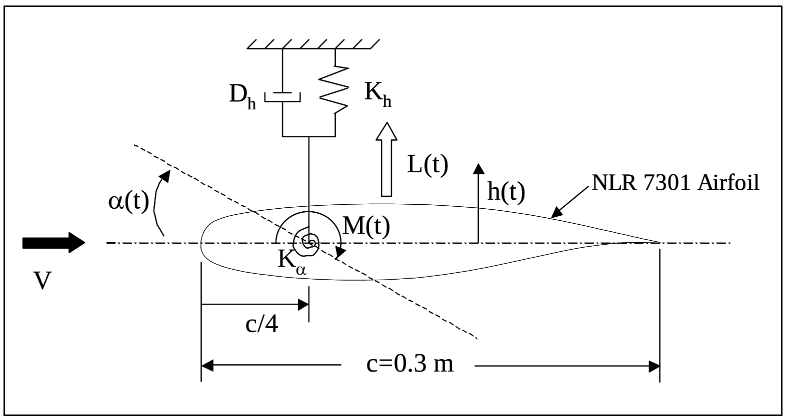

Figure 1. Two-Degree-of-Freedom Dynamic Model

Figure 1 depicts a simplified model of the 2-degree-of-freedom test set-up. The 2-D wing has a chord length of 0.3 m (c = 0.3 m) and a span of 1 m (b = 1 m). The pitching spring and heaving spring are attached to the same c/4 position. The corresponding 2-degree-of-freedom equation of motion of the set-up reads,

$$\begin{bmatrix} m_h & -s_\alpha \\ -s_\alpha & I_{c/4} \end{bmatrix} \begin{Bmatrix} \ddot{h} \\ \ddot{\alpha} \end{Bmatrix} + \begin{bmatrix} D_h & 0 \\ 0 & D_\alpha \end{bmatrix} \begin{Bmatrix} \dot{h} \\ \dot{\alpha} \end{Bmatrix} + \begin{bmatrix} K_h & 0 \\ 0 & K_\alpha \end{bmatrix} \begin{Bmatrix} h \\ \alpha \end{Bmatrix} = \begin{Bmatrix} L(t) \\ M(t) \end{Bmatrix} \tag{1}$$

where

m_h	is the total mass (m_h = 26.64 kg)
$I_{c/4}$	is the mass moment of inertia about c/4 ($I_{c/4}$ = 0.086 kg-m^2)
s_α	is the static unbalance (s_α = 0.378 kg-m)
D_h and D_α	are the damping factors of the heave motion (h) and the pitch motion (α), respectively (D_h = 82.9 kg/s and D_α = 0.197 kg-m^2/(rad-s))
K_h and K_α	are the stiffness of the heaving spring and pitching spring, respectively (K_h = 1.21 x 10^6 N/m and K_α = 6.68 x 10^3 N-m/rad), and

$L(t)$ and $M(t)$ are the aerodynamic lift and moment, respectively in Newtons.

To perform the time-marching CFD computation in CFL3D v6.0, it is necessary to convert Eq (1) into modal coordinates, i.e.:

$$\begin{Bmatrix} h \\ \alpha \end{Bmatrix} = [\,\phi\,]\,\{\,q\,\} \tag{2}$$

where q is modal coordinate and ϕ is the modal matrix of the undamped structure. For this numerical example we have

$$\phi = \begin{bmatrix} -0.1735 & 0.1004 \\ 0.9277 & 3.403 \end{bmatrix}.$$

Substituting Eq (2) into Eq (1) and pre-multiplying the resulting equation by ϕ^{T} yields:

$$[I]\,\{\ddot{q}\} + \begin{bmatrix} 2\,\omega_h\,\zeta_h & 0 \\ 0 & 2\,\omega_\alpha\,\zeta_\alpha \end{bmatrix}\{\dot{q}\} + \begin{bmatrix} \omega_h{}^2 & 0 \\ 0 & \omega_\alpha{}^2 \end{bmatrix}\{q\} = q_\infty\;\phi^T \begin{Bmatrix} \iint c_p(x^*,y^*)\,dx^*\,dy^* \\ \iint c_p(x^*,y^*)(x^*{}_{ea}-x^*)\,dx^*\,dy^* \end{Bmatrix}$$

(3)

where

ω_h and ω_α are the undamped natural frequencies of the heaving and pitching motions, respectively (ω_h = 205.4 rad/s and ω_α = 299.5 rad/s)

ζ_h and ζ_α are the heaving and pitching damping ratios, respectively (ζ_h = 0.00648 and ζ_α = 0.00474). Note that the off-diagonal terms in the damping matrix are assumed to be zero for simplicity.

q_∞ is the dynamic pressure.

Note that CFL3D does ***not*** *use the C_l and C_m, computed in the aerodynamic portion of the code, in calculating aeroelastic loads*. Rather it integrates pressures over the aeroelastic surface portion of the CFD grid, using the factor *grefl* to convert to the proper dimensions used in the aeroelastic calculations. That is, in subroutine 'genforce' the integration, in general, will be

$$G_{iaesurf,n} = \iint_{iaesurf} \phi_n(x,y)\;c_p(x,y)\,dx\,dy$$

for the n^{th} mode, where x and y are as defined by the CFD grid. In the subroutine 'ae_corr' the resulting force is multipled by *qinf*grefl*grefl* to properly dimension the force within the elastic structural integration. (See the discussion under the input parameter *grefl* for the meaning of this term and for discussion of the non dimensionalization of the output generalized force) In the present example *grefl = 1* since the CFD grid has been set up such that the grid chord length is .3 and the grid span is 1. (See the input in example 2 of Section V and the discussion on the input parameter.)

The modal input for CFL3D in the *present* example is calculated by

$$\Phi_{j=1,jaegrids} = \phi^T \begin{Bmatrix} 1 \\ (x^*_{ea} - x^*_n) \end{Bmatrix}$$

where Φ is a (*jaegrids,3,2*) array where *jaegrids* is the number of aeroelastic surface grids in the surface segment. The dimension *2* corresponds to the two modes in this example.

…. If generalized aerodynamic forces per dynamic pressure per area are desired, …

APPENDIX B. KEYWORD INPUT

APPENDIX C. FLOW CHART OF DEFORMING MESH, AEROELASTICITY