

## APPENDIX B: Summary of 3-D File Formats

This Appendix provides a summary of the input/output file formats used by the three-dimensional CFD solver developed for this research.

The following *input* files are defined:

- `case.g3d` (required) contains the geometry data structures representing the computational volume mesh as required by the flow solver. (Binary)
- `case.con` (required) contains values for the solver control parameters and flow conditions. (ASCII)
- `case.unk` (optional) contains the nodal values of the primitive flow variables (density, velocity, and pressure) for each node of the computational mesh to be used as the initial conditions for the flow solution. (Binary)
- `case.dyn` (optional) contains the non-inertial matrices and initial conditions as required for a dynamic solution. (ASCII)
- `case.vec` (optional) contains the elastic mode matrices, initial conditions, and vectors for the boundary surfaces as required for an aeroelastic solution. (ASCII)
- `case.frc` (optional) contains external forces to be applied to each solution step in a dynamic or aeroelastic solution. (ASCII)

The following *output* files are defined:

- `case.un1` contains the nodal values of the primitive flow variables (density, velocity, and pressure) for each node of the computational mesh. (Binary)
- `case.rsd` contains a history of the solution residuals for the conservation variables (density, momentum, and total energy). (ASCII)
- `case.lds` contains a history of the non-dimensional aerodynamic forces acting on the solid walls of the CFD geometry. (ASCII)
- `xd.dat` contains a history of the non-inertial displacements, velocities, and accelerations for a dynamic solution. (ASCII)
- `xn.dat` contains a history of the generalized displacements and velocities for an unsteady, aeroelastic solution. (ASCII)

## Geometry Input File (case.g3d)

### Basic File Format

```
nnd nel nsg nbe nbp nwl nsd nsf  
( LBE(i), i = 1,6 )  
( ( COOR(i,j), i=1,nnd ), j=1,3 )  
( ( IELM(i,j), i=1,nel ), j=1,4 )  
( ( ISEG(i,j), i=1,nsg ), j=1,2 )  
( ( IBEL(i,j), i=1,nbe ), j=1,3 )
```

### Definition of Terms

nnd: (int) number of nodes  
nel: (int) number of elements  
nsg: (int) number of segments  
nbe: (int) number of boundary elements  
nbp: (int) number of boundary points  
nwl: (int) number of wall nodes  
nsd: (int) number of singular nodes  
nsf: (int) number of boundary surfaces

LBE(*i*): (int) boundary element  
starting/stopping indexes for three BC types

COOR(*i*, 1): (real) x-coordinate for node *i*  
COOR(*i*, 2): (real) y-coordinate for node *i*  
COOR(*i*, 3): (real) z-coordinate for node *i*

IELM(*i*, 1): (int) node 1 for element *i*  
IELM(*i*, 2): (int) node 2 for element *i*  
IELM(*i*, 3): (int) node 3 for element *i*  
IELM(*i*, 4): (int) node 4 for element *i*

ISEG(*i*, 1): (int) node 1 for segment *i*  
ISEG(*i*, 2): (int) node 2 for segment *i*

IBEL(*i*, 1): (int) node 1 for boundary elem. *i*  
IBEL(*i*, 2): (int) node 2 for boundary elem. *i*  
IBEL(*i*, 3): (int) node 3 for boundary elem. *i*  
IBEL(*i*, 4): (int) boundary surface containing  
boundary element *i*

### Comments

- This is an unformatted (binary) file.
- Nodal data is sorted such that the first nwl nodes are defined as solid wall nodes. Out of the first nwl nodes, the last nsd nodes are defined as singular nodes.
- The nodal coordinates in this file are treated as dimensional values and are non-dimensionalized using the reference dimension specified in the solver control file.
- Boundary element data is sorted based on the starting/stopping indexes for the three BC types, i.e. boundary elements LBE(1) through LBE(2) are solid wall elements, LBE(3) through LBE(4) are symmetry elements, and LBE(5) through LBE(6) are far-field elements.

- The program `makeg3d` is used to convert a standard STARS surface triangulation file, tetrahedral volume file, and modified boundary conditions file into an appropriately sorted three-dimensional geometry file.

## Solver Control File (case .con)

### Basic File Format

```
&control
  dt          = 0.1d0,
  gamma       = 1.4d0,
  diss        = 1.0d0,
  cfl         = 0.5d0,

  mach        = 0.6d0,
  alpha       = 0.0d0,
  beta        = 0.0d0,
  refdim      = 1.0d0,

  nstp        = 100,
  nout        = 50,
  ncyc        = 3,
  isol        = 0,
  idsol       = 2,
  idiss       = 0,
  ipnt        = 1,

  istrtr      = .false.,
  iaero       = .false.,
  idynm       = .false.,
  ielast      = .false.,
  ifree       = .true.,
  iforce      = .true.,

  nr          = 0,
  ainf        = 1.0d0,
  rhoinf     = 1.0d0,
/
```

### Definition of Terms

dt: (real) non-dimensional global time step  
gamma: (real) ratio of specific heats  
diss: (real) dissipation constant  
cfl: (real) local time step stability factor

mach: (real) free-stream mach number  
alpha: (real) 1<sup>st</sup> free-stream orientation angle  
beta: (real) 2<sup>nd</sup> free-stream orientation angle  
refdim: (real) reference dimension

nstp: (int) total solution steps  
nout: (int) output frequency, steps/output  
ncyc: (int) iterative cycles per solution step  
isol: (int) CFD solution type  
idsol: (int) dynamics solution type  
idiss: (int) dissipation type  
ipnt: (int) number of points for numerical integration of flux/source vectors

istrtr: (logical) restart flag  
iaero: (logical) aerodynamic forces flag  
idynm: (logical) dynamic/non-inertial flag  
ielast: (logical) elastic flag  
ifree: (logical) free-stream velocity flag  
iforce: (logical) external force flag

nr: (int) number of elastic modes  
ainf: (real) dimensional free-stream sonic speed  
rhoinf: (real) dimensional free-stream density

### Comments

- This is a plain text (ASCII) file formatted as a Fortran namelist.
- The default values for each parameter are given in the basic file format above.
- The global time step is only used for unsteady solutions.
- Appropriate values for the dissipation factor are in the range  $0.0 < \text{diss} \leq 2.0$ . Some dissipation is required to stabilize the solution, but too much dissipation will corrupt the solution and possibly be a destabilizing influence.
- The local time step stability factor is a safety factor used to compute local time steps for each solution step. For steady solutions, a stability factor of 0.8 is typically acceptable for most problems. For unsteady solutions, the stability factor is typically in the range  $0.3 \leq \text{cfl} \leq 0.8$ .

- The values of `refdim`, `mach`, `ainf`, and `rhoinf` are used to non-dimensionalize all values read in by the flow solver.
- The free-stream orientation angles are ignored for dynamic (non-inertial) problems.
- The number of iterative cycles should be set to 3 for steady solutions. For unsteady solutions, use a sufficient number of cycles to allow for an appropriate level of convergence at each step.
- There are four available CFD solution types defined as follows:
  - `isol = 0` is a steady solution (not time accurate)
  - `isol = 1` is a first-order unsteady solution
  - `isol = 2` is a second-order unsteady solution
  - `isol = 3` is a supersonic piston perturbation solution
- There are three available dynamics solution types defined as follows:
  - `idsol = 0` uses a zero-order integrator for the applied forces
  - `idsol = 1` uses a first-order integrator for the applied forces
  - `idsol = 2` uses a second-order integrator for the applied forces
- There are two available dissipation types defined as follows:
  - `idiss = 0` is a low order dissipation
  - `idiss = 1` is a high order dissipation with gradient limiters
- The low-order dissipation is typically overly diffuse and should be used in conjunction with low values of the dissipation factor. Low-order dissipation works best for problems without strong vortices and for supersonic/hypersonic flows.
- The high-order dissipation is more CPU intensive than the low-order dissipation and less stable. Larger values for the dissipation factor are typically required for stabilization. The high-order dissipation works best for subsonic to transonic flows with strong gradients or vortices. Rotating domains will typically require high-order dissipation to resolve the circulating pattern of the relative flow velocities.
- There are two types of numerical integration defined as follows:
  - `ipnt = 1` uses a one-point gauss quadrature
  - `ipnt = 4` uses a three-point symmetric gauss quadrature
- When the restart flag is set to `.true.`, the solver will read one set of solution unknowns from the `case.unk` file and apply this set of unknowns as the initial conditions for the new iterative solution.
- A restarted solution assumes that the time gradient of the initial state is zero, i.e. the solution stored in the `case.unk` file is a converged, steady state solution. This has a significant impact on the second-order unsteady solution since it relies on two sets of solution unknowns for advancement to the next time step, i.e. a second-order unsteady solution should not be restarted from the last time step of a similar unsteady solution that was stopped because both sets of unsteady data from the last solution step are not available for accurate evaluation of the time gradients in the flow.

- If the free-stream velocity flag is set to `.false.`, the free-stream velocity is set to zero, and relative flow velocities must be generated through dynamic rotation or translation of the non-inertial coordinate system.
- If the external force flag is set to `.true.`, the solver will read the user defined external force vector for each global time step from the input file `case.frc`. If the solver reaches the end of the input file before completing the solution, the last force vector in the file carries over to each of the remaining global time steps if it was non-zero.

## Solution Unknowns Input/Output File (case.un\*)

### Basic File Format

```
nnd gam xmi alp bet ref t
( ( UN(i,j), i=1,nnd ), j=1,6 )
```

### Definition of Terms

nnd: (int) number of nodes  
gam: (real) ratio of specific heats  
xmi: (real) free stream mach number  
alp: (real) 1<sup>st</sup> free-stream orientation angle  
bet: (real) 2<sup>nd</sup> free-stream orientation angle  
ref: (real) reference dimension  
t: (real) non-dimensional time

UN( i , 1 ) : (real) density for node *i*  
UN( i , 2 ) : (real) x-velocity for node *i*  
UN( i , 3 ) : (real) y-velocity for node *i*  
UN( i , 4 ) : (real) z-velocity for node *i*  
UN( i , 5 ) : (real) pressure for node *i*  
UN( i , 6 ) : (real) enthalpy for node *i*

### Comments

- This is an unformatted (binary) file.
- The solution unknowns stored in this file are non-dimensional quantities.
- For dynamic (non-inertial) problems, the solution unknowns stored in this file are relative quantities referenced to the body-fixed coordinate system.

## Dynamic Mesh Input File (case.dyn)

### Basic File Format

Line of Text

```
( R0(i), i=1,3 )
```

Line of Text

```
( ( RM1(i,j), j=1,6 ), i=1,6 )
```

Line of Text

```
( ( RC1(i,j), j=1,6 ), i=1,6 )
```

Line of Text

```
( ( RK1(i,j), j=1,6 ), i=1,6 )
```

Line of Text

```
x, y, z, p, q, r,  
vx, vy, vz, vp, vq, vr,  
ax, ay, az, ap, aq, ar
```

Line of Text

```
( IBXD(i), i=1,6 )
```

### Definition of Terms

R0(1): (real) x-coordinate for origin of rotation

R0(2): (real) y-coordinate for origin of rotation

R0(3): (real) z-coordinate for origin of rotation

RM1(i,j): (real) dimensional mass matrix

RC1(i,j): (real) dimensional damping matrix

RK1(i,j): (real) dimensional stiffness matrix

x: (real) initial x-position of coord. system

y: (real) initial y-position of coord. system

z: (real) initial z-position of coord. system

p: (real) initial roll angle of coord. system

q: (real) initial pitch angle of coord. system

r: (real) initial yaw angle of coord. system

vx: (real) initial x-velocity of coord. system

vy: (real) initial y-velocity of coord. system

vz: (real) initial z-velocity of coord. system

vp: (real) initial roll rate of coord. system

vq: (real) initial pitch rate of coord. system

vr: (real) initial yaw rate of coord. system

ax: (real) initial x-acceleration of coord. system

ay: (real) initial y-acceleration of coord. system

az: (real) initial z-acceleration of coord. system

ap: (real) initial roll accel. of coord. system

aq: (real) initial pitch accel. of coord. system

ar: (real) initial yaw accel. of coord. system

IBXD(i): (int) dynamics flag for  $i^{\text{th}}$  DOF

### Comments

- This is a plain text (ASCII) file.
- All values entered into this file should be dimensional. The solver will automatically non-dimensionalize the values using the reference conditions specified in the solver control file.
- The vector defining the origin of rotation is subtracted directly from the nodal coordinates defined in the geometry input file after it is non-dimensionalized by the reference dimension.
- The mass matrix defined in this file cannot be singular.
- Initial conditions for the three translational degrees of freedom are specified relative to the inertial coordinate system, i.e. as seen by a stationary observer on the ground.

- Initial conditions for the three rotational degrees of freedom should have units of degrees, degrees/sec, etc., and are true Euler angles and rates expressed relative to the body-fixed coordinate system.
- The dynamics of each degree of freedom is controlled separately using the following values for IBXD:
  - IBXD = 0 is a free/forced response calculation, i.e. uses mass, stiffness, and damping to compute position, velocity, and acceleration of system.
  - IBXD = 1 is a clamped condition, i.e. hold at initial position with zero velocity and acceleration.
  - IBXD = 2 is a constant acceleration, uncoupled response, i.e. integrates acceleration and velocity to compute new position.

## Sample File

```
$ Position vector to origin of non-inertial frame (rx, ry, rz)
0.0d0 0.0d0 0.0d0
$ Mass matrix for non-inertial frame (6 x 6)
1.0d0 0.0d0 0.0d0 0.0d0 0.0d0 0.0d0
0.0d0 1.0d0 0.0d0 0.0d0 0.0d0 0.0d0
0.0d0 0.0d0 1.0d0 0.0d0 0.0d0 0.0d0
0.0d0 0.0d0 0.0d0 1.0d0 0.0d0 0.0d0
0.0d0 0.0d0 0.0d0 0.0d0 1.0d0 0.0d0
0.0d0 0.0d0 0.0d0 0.0d0 0.0d0 1.0d0
$ Damping matrix for non-inertial frame (6 x 6)
0.0d0 0.0d0 0.0d0 0.0d0 0.0d0 0.0d0
0.0d0 0.0d0 0.0d0 0.0d0 0.0d0 0.0d0
0.0d0 0.0d0 0.0d0 0.0d0 0.0d0 0.0d0
0.0d0 0.0d0 0.0d0 0.0d0 0.0d0 0.0d0
0.0d0 0.0d0 0.0d0 0.0d0 0.0d0 0.0d0
0.0d0 0.0d0 0.0d0 0.0d0 0.0d0 0.0d0
$ Stiffness matrix for non-inertial frame (6 x 6)
1.0d0 0.0d0 0.0d0 0.0d0 0.0d0 0.0d0
0.0d0 1.0d0 0.0d0 0.0d0 0.0d0 0.0d0
0.0d0 0.0d0 1.0d0 0.0d0 0.0d0 0.0d0
0.0d0 0.0d0 0.0d0 1.0d0 0.0d0 0.0d0
0.0d0 0.0d0 0.0d0 0.0d0 1.0d0 0.0d0
0.0d0 0.0d0 0.0d0 0.0d0 0.0d0 1.0d0
$ IC's for non-inertial frame (6 positions, 6 rates, 6 accels )
0.0d0 0.0d0 0.0d0 0.0d0 0.0d0 0.0d0
0.0d0 0.0d0 0.0d0 0.0d0 0.0d0 0.0d0
0.0d0 0.0d0 0.0d0 0.0d0 0.0d0 0.0d0
$ IBXD for non-inertial frame (6)
1 1 1 1 1 1
```

## Elastic Vectors Input File (case .vec)

### Basic File Format

Line of Text

nr

Line of Text

( ( RM(i,j), j=1,nr ), i=1,nr )

Line of Text

( ( RC(i,j), j=1,nr ), i=1,nr )

Line of Text

( ( RK(i,j), j=1,nr ), i=1,nr )

Line of Text

( XN(i), i=1,nr\*2 )

Line of Text

( IBXN(i), i=1,nr )

Line of Text

( ( PHIA(i,j), i=1,nwl\*3 ), j=1,nr )

### Definition of Terms

nr : (int) number of elastic modes

RM(i,j) : (real) dimensional mass matrix

RC(i,j) : (real) dimensional damping matrix

RK(i,j) : (real) dimensional stiffness matrix

XN(i) : (real) initial gen. displ. for mode  $i$

XN(i+nr) : (real) initial gen. vel. for mode  $i$

IBXN(i) : (int) dynamics flag for  $i^{\text{th}}$  mode

PHIA(i\*3-2,j) :  $x$ -component of displacement vector for mode  $j$  at node  $i$

PHIA(i\*3-1,j) :  $y$ -component of displacement vector for mode  $j$  at node  $i$

PHIA(i\*3 ,j) :  $z$ -component of displacement vector for mode  $j$  at node  $i$

### Comments

- This is a plain text (ASCII) file.
- All values entered into this file should be dimensional. The solver will automatically non-dimensionalize the values using the reference conditions specified in the solver control file.
- The mass matrix defined in this file cannot be singular.
- The dynamics of each degree of freedom is controlled separately using the following values for IBXN:
  - IBXN = 0 is a free/forced response calculation, i.e. uses mass, stiffness, and damping to compute generalized displacement and velocity.
  - IBXN = 1 is a clamped condition, i.e. hold at initial generalized displacement with zero velocity.
  - IBXN = 2 is a constant velocity, uncoupled response, i.e. integrates generalized velocity to compute new displacement.

- $IBXN = 3$  is a forced multistep response used for system identification purposes.
- Do not combine  $IBXN = 0$  with  $IBXN \neq 0$  for different modes if there are coupling or off-diagonal terms in the mass, damping or stiffness matrices.
- A limited set of simple modal vectors representing standard rigid-body degrees of freedom can be generated using the program `makevec3d`.

## Sample File

```
$ Number of elastic modes (nr)
3
$ Mass matrix for elastic modes (nr x nr)
1.0d0      0.0d0      0.0d0
0.0d0      1.0d0      0.0d0
0.0d0      0.0d0      1.0d0
$ Damping matrix for elastic modes (nr x nr)
0.0d0      0.0d0      0.0d0
0.0d0      0.0d0      0.0d0
0.0d0      0.0d0      0.0d0
$ Stiffness matrix for elastic modes (nr x nr)
1.0d0      0.0d0      0.0d0
0.0d0      1.0d0      0.0d0
0.0d0      0.0d0      1.0d0
$ IC's for elastic modes (x1....xn, vx1...vxn)
0.0d0 0.0d0 0.0d0 0.0d0 0.0d0 0.0d0
$ IBXN for elastic modes (nr)
      1      1      1
$ Elastic modes vectors (nwl*3) x nr
0.0d0  1.0d0  1.0d0
0.0d0  1.0d0  1.0d0
0.0d0  1.0d0  1.0d0
0.0d0  1.0d0  1.0d0
1.0d0  0.0d0  1.0d0
0.0d0  1.0d0  1.0d0
0.0d0  1.0d0  1.0d0
  ⋮      ⋮      ⋮
```

## External Force Input File (`case.frc`)

### Basic File Format

```
      1  (FD(i), i=1,6) (FE(j), j=1,nr)
      ⋮      ⋮      ⋮
  istp (FD(i), i=1,6) (FE(j), j=1,nr)
      ⋮      ⋮      ⋮
  nstp (FD(i), i=1,6) (FE(j), j=1,nr)
```

### Definition of Terms

`istp`: (int) current solution step  
`nstp`: (int) total or last solution step  
`nr`: (int) number of elastic modes

`FD(1)`: (real)  $x$  force applied at `istp`  
`FD(2)`: (real)  $y$  force applied at `istp`  
`FD(3)`: (real)  $z$  force applied at `istp`  
`FD(4)`: (real) roll moment applied at `istp`  
`FD(5)`: (real) pitch moment applied at `istp`  
`FD(6)`: (real) yaw moment applied at `istp`  
`FE(j)`: (real) force applied to elastic mode  $j$

### Comments

- This is a plain text (ASCII) file.
- All values entered into this file should be dimensional. The solver will automatically non-dimensionalize the values using the reference conditions specified in the solver control file.
- Forces applied to the three translational degrees of freedom are specified relative to the inertial coordinate system.
- The specified forces are read one line at a time following each solution step.
- Up to `nstp` forces may be specified, but are not required. The last force read in by the solver will be applied for all remaining solution steps.

### Sample File

1	0.0d0	0.0d0	0.0d0	0.0d0	0.0d0	0.0d0
2	0.0d0	0.0d0	0.0d0	0.0d0	0.0d0	0.0d0
3	0.0d0	0.0d0	1.0d0	0.0d0	1.0d0	0.0d0
4	0.0d0	0.0d0	1.0d0	0.0d0	2.0d0	0.0d0
5	0.0d0	0.0d0	1.0d0	0.0d0	1.0d0	0.0d0
6	0.0d0	0.0d0	1.0d0	0.0d0	0.0d0	0.0d0
7	0.0d0	0.0d0	0.0d0	0.0d0	0.0d0	0.0d0

## Solution Residuals Output File (case.rsd)

### Basic File Format

```
1 ( RSD(i), i=1,5 )  
:  
:  
istp ( RSD(i), i=1,5 )  
:  
:  
nstp ( RSD(i), i=1,5 )
```

### Definition of Terms

istp: (int) current solution step  
nstp: (int) total or last solution step  
RSD(1): (real) density solution residual  
RSD(2): (real) x-momentum solution residual  
RSD(3): (real) y-momentum solution residual  
RSD(4): (real) z-momentum solution residual  
RSD(5): (real) energy solution residual

### Comments

- This is a plain text (ASCII) file.
- For steady problems, the solution residuals indicate the degree of convergence to the final steady state solution. All four solution residuals should converge to approximately the same order of magnitude.
- For unsteady problems, the solution residuals indicate the degree of convergence for each global step of the solution, or the degree of convergence for the steady solution that is solved at each step.

## Sample File

1	0.12222E-07	0.10626E-07	0.16687E-07	0.12295E-07	0.22126E-07
2	0.11570E-07	0.10062E-07	0.15729E-07	0.11602E-07	0.20890E-07
3	0.10921E-07	0.95097E-08	0.14773E-07	0.10908E-07	0.19662E-07
4	0.10275E-07	0.89463E-08	0.13814E-07	0.10215E-07	0.18436E-07
5	0.96242E-08	0.83818E-08	0.12854E-07	0.95216E-08	0.17203E-07
6	0.89786E-08	0.78260E-08	0.11892E-07	0.88276E-08	0.15977E-07
7	0.83298E-08	0.72699E-08	0.10929E-07	0.81336E-08	0.14748E-07
8	0.76831E-08	0.67176E-08	0.99662E-08	0.74405E-08	0.13527E-07
9	0.70441E-08	0.61708E-08	0.90087E-08	0.67538E-08	0.12320E-07
10	0.64162E-08	0.56360E-08	0.80584E-08	0.60769E-08	0.11139E-07
11	0.57970E-08	0.51153E-08	0.71208E-08	0.54130E-08	0.99765E-08
12	0.51854E-08	0.46001E-08	0.61957E-08	0.47627E-08	0.88260E-08
13	0.45869E-08	0.40968E-08	0.52837E-08	0.41320E-08	0.76967E-08
14	0.40076E-08	0.36189E-08	0.43870E-08	0.35263E-08	0.66145E-08
15	0.34579E-08	0.31776E-08	0.35119E-08	0.29571E-08	0.56028E-08
16	0.29562E-08	0.27873E-08	0.26754E-08	0.24518E-08	0.47029E-08
17	0.25236E-08	0.24650E-08	0.19320E-08	0.20618E-08	0.39779E-08
18	0.21945E-08	0.22406E-08	0.14353E-08	0.18541E-08	0.35322E-08
19	0.20227E-08	0.21516E-08	0.14507E-08	0.18860E-08	0.34857E-08
20	0.20517E-08	0.22121E-08	0.19558E-08	0.21498E-08	0.38474E-08
⋮	⋮	⋮	⋮	⋮	⋮

## Aerodynamic Loads Output File (case.lds)

### Basic File Format

```
0    0.0    ( FD(i), i=1,6 )
:    :      :
:    :      :
istp t_istp ( FD(i), i=1,6 )
:    :      :
:    :      :
nstp t_nstp ( FD(i), i=1,6 )
```

### Definition of Terms

*istp*: (int) current solution step  
*nstp*: (int) total or last solution step  
*t<sub>i</sub>*: (real) non-dimensional time at step *i*

*FD(1)*: (real) *x*-force coefficient  
*FD(2)*: (real) *y*-force coefficient  
*FD(3)*: (real) *z*-force coefficient  
*FD(4)*: (real) *x*-moment coefficient  
*FD(5)*: (real) *y*-moment coefficient  
*FD(6)*: (real) *z*-moment coefficient

### Comments

- This is a plain text (ASCII) file.
- The force coefficients in this output file are non-dimensional values based on the reference conditions specified in the solver control file.
- For dynamic (non-inertial) problems, the force coefficients stored in this file are referenced to the body-fixed coordinate system.

## Sample File

```
0 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
1 0.10000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
2 0.20000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
3 0.30000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
4 0.40000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
5 0.50000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
6 0.60000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
7 0.70000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
8 0.80000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
9 0.90000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
10 0.10000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
11 0.11000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
12 0.12000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
13 0.13000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
14 0.14000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
15 0.15000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
16 0.16000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
17 0.17000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
18 0.18000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
19 0.19000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
20 0.20000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
⋮      ⋮      ⋮      ⋮      ⋮      ⋮      ⋮
```

## Dynamic Output File (xd.dat)

### Basic File Format

```
0 0.0 (XD(i), i=1,12) (A0(i), i=1,6)
:   :   :   :
istp t_istp (XD(i), i=1,12) (A0(i), i=1,6)
:   :   :   :
nstp t_nstp (XD(i), i=1,12) (A0(i), i=1,6)
```

### Definition of Terms

istp: (int) current solution step  
nstp: (int) total or last solution step  
 $t_i$ : (real) non-dimensional time at step  $i$

XD(1): (real)  $x$ -position  
XD(2): (real)  $y$ - position  
XD(3): (real)  $z$ - position  
XD(4): (real) roll angle  
XD(5): (real) pitch angle  
XD(6): (real) yaw angle  
XD(7): (real)  $x$ -velocity  
XD(8): (real)  $y$ - velocity  
XD(9): (real)  $z$ - velocity  
XD(10): (real) roll rate  
XD(11): (real) pitch rate  
XD(12): (real) yaw rate

A0(1): (real)  $x$ -acceleration  
A0(2): (real)  $y$ - acceleration  
A0(3): (real)  $z$ - acceleration  
A0(4): (real) roll acceleration  
A0(5): (real) pitch acceleration  
A0(6): (real) yaw acceleration

### Comments

- This is a plain text (ASCII) file.
- The dynamic data in this output file are non-dimensional values based on the reference conditions specified in the solver control file.
- The position, velocity, and acceleration vectors in this file are defined relative to the global coordinate system, while the rotational vectors are defined as rotations about the local or body-fixed coordinate system.



## Elastic Output File (xn.dat)

### Basic File Format

```
0 0.0 (XN(i), i=1, nr*2) (FA(i), i=1, nr)
:   :   :
:   :   :
istp t_istp (XN(i), i=1, nr*2) (FA(i), i=1, nr)
:   :   :
:   :   :
nstp t_nstp (XN(i), i=1, nr*2) (FA(i), i=1, nr)
```

### Definition of Terms

istp: (int) current solution step  
nstp: (int) total or last solution step  
 $t_i$ : (real) non-dimensional time at step  $i$

XN( $i$ ): (real) gen. displ. for mode  $i$   
XN( $i+nr$ ): (real) gen. vel. for mode  $i$

FA( $i$ ): (real) gen. force for mode  $i$

## **Comments**

- This is a plain text (ASCII) file.
- The elastic data in this output file are non-dimensional values based on the reference conditions specified in the solver control file.
- The sample file on the following page is for a two mode solution.

## Sample File

```
0 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
1 0.10000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
2 0.20000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
3 0.30000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
4 0.40000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
5 0.50000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
6 0.60000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
7 0.70000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
8 0.80000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
9 0.90000E-01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
10 0.10000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
11 0.11000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
12 0.12000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
13 0.13000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
14 0.14000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
15 0.15000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
16 0.16000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
17 0.17000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
18 0.18000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
19 0.19000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
20 0.20000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
⋮      ⋮      ⋮      ⋮      ⋮      ⋮      ⋮
```