

## NASCART-GT Input File Formats

---

### Description: Main Input Parameter File

File Name: input.dat

File Type: ascii

Namelist format. See code web site for complete list of Input Variables.

---

### Description: Surface Geometry Input File

File Name: bodyin.dat

File Type: ascii

Read Statements: Eleven formats are possible...

If igrd=1, then:

```
read(1,*) npts
```

```
nbpanel = 0
```

```
do 110 npant = 1,npandim
```

```
do 109 nptt = 1,npts
```

```
read(1,*) npdum,xbody(npant,nptt),ybody(npant,nptt),
```

```
&      zbody(npant,nptt)
```

```
if(npdum.eq.0) go to 111
```

```
109 continue
```

```
nbpanel = nbpanel + 1
```

```
if(npts.eq.3) then
```

```
    xbody(npant,4) = xbody(npant,3)
```

```
ybody(npant,4) = ybody(npant,3)
```

```
zbody(npant,4) = zbody(npant,3)
```

```
endif
```

```
110 continue
```

```
111 continue
```

If igrd=2, then:

```
read(1,*) iptskpts
```

```
read(1,*) ((xp3d(i,k),i=1,iptsk),k=1,kpts),
```

```
& ((yp3d(i,k),i=1,iptsk),k=1,kpts),
```

```
& ((zp3d(i,k),i=1,iptsk),k=1,kpts)
```

If igrd=3, then:

CATIATM v5 meshed output file is read. The meshed outer surface may be performed using quadrilaterals, triangles, or a mixture of each. See the CATIA File Compatibility page for instructions on geometry creation with CATIATMv5

If igrd=4, then:

```
read(1,*) nsegmax
```

```
nline = 0
```

```
do nseg = 1,nsegmax
```

```
read(1,*) nptss(nseg)
```

```
do n1 = 1,nptss(nseg)
```

```
nline = nline + 1
```

```
read(1,*) xline(nline),yline(nline)
```

enddo

enddo

If igrd=5, then:

read(1,\*) nzone

do nz = 1,nzone

read(1,\*) iptsz(nz),kptsz(nz)

enddo

do nz = 1,nzone

read(1,\*) ((xp3d(i,k),i=1,iptsz(nz)),k=1,kptsz(nz)),

& ((yp3d(i,k),i=1,iptsz(nz)),k=1,kptsz(nz)),

& ((zp3d(i,k),i=1,iptsz(nz)),k=1,kptsz(nz))

enddo

If igrd=6, then:

File is from VSP (Formerly RAM).

If igrd=7, then:

File is in UCD format.

If igrd=8, then:

No solid body is included in the calculation so no geometry file required.

If igrd=9, then:

File is in ascii STL format.

If igrid=10, then:

File is in ascii VRMLv1.0 format.

If igrid=11, then:

File is in ascii VRMLv2.0 format.

Definitions:

If igrid=1, then:

npts                      If = 3, then 3 corner points describe each panel (i.e. triangular panels)

                            If = 4, then 4 corner points describe each panel (i.e. quadrilateral panels)

npdum                     If = 0, then this indicates the last line in the file

                            Otherwise, then npdum is any integer

xbody, ybody, zbody    x, y, and z coordinates, respectively, of a corner point

                            Each panel has npts lines of corner coordinates

If igrid=2, then:

ipts                      Number of i points in PLOT3D surface grid

kpts                      Number of k points in PLOT3D surface grid

xp3d, yp3d, zp3d        x, y, and z coordinates, respectively, of PLOT3D surface grid points

If igrid=4, then:

nsegmax                 Number of line segments

nptss(nseg)             Number of points in segment number nseg

xline, yline             Ordered (x,y) (for 2-D), or (x,r) (for axisymmetric) coordinates of  
the geometry [m]

Note:

For 2-D cases ( $n2d=1$ ) the upper and lower surfaces may be read in as two separate segments OR the entire geometry may be read in as one continuous segment. However, the body must be fully enclosed (i.e. airtight). Thus if the geometry is read in as one continuous segment, the first and last points in the segment must be the same.

For axisymmetric cases ( $n2d=-1$ ) only the upper surface (x,r) coordinates should be supplied. Thus, the geometry read in is not fully enclosed (i.e. airtight). The geometry is computed to correspond to a circular cross section surface with the user supplied radial values.

If  $igrid=5$ , then:

nzone	Number of PLOT3D grid zones
iptsz(nz)	Number of i points in PLOT3D surface grid for zone number nz
kptsz(nz)	Number of k points in PLOT3D surface grid for zone number nz
xp3d, yp3d, zp3d	x, y, and z coordinates, respectively, of PLOT3D surface grid points

### Description: Multi-Surface Input File (Required only if $igrid=0$ )

File Name: bodyinall.dat

File Type: ascii

Read Statements:

```

read(25,*) ncolor
read(25,*) do ncol = 1,ncolor
  read(25,*)
ncol_,ncnames(ncol),igrid(ncol),ibsym(ncol),jbsym(ncol),kbsym(ncol),nbminc(ncol),ibccolor(ncol),nbcnames(ncol)
enddo
read(25,*) ngaps
do n1 = 1,ngaps
  read(25,*) ncolgap(n1,1),ncolgap(n1,2)
enddo

```

Definitions:

ncolor                    Number of separate body input files to be read

ncol                      Input file number (Start with 1 and increase by one on each line)

ncnames(ncol)          Name of geometry input file for geometry number ncol

igrd(ncol)              igrd value (i.e. file type) for geometry number ncol (see above for possible igrd values)

                            If igrd > 0 then calculate and store neighbor information (bodynext.dat) for input geometry (req'd for first run of geometry)

                            If igrd < 0, then read in neighbor information from bodynext.dat (faster for subsequent runs of the same geometry).

ibsym(ncol)             If = 1 then duplicate the input geometry number ncol about the x=0 plane

                            If = 0, then do NOT duplicate the body about the x=0 plane

jbsym(ncol)             If = 1 then duplicate the input geometry number ncol about the y=0 plane

                            If = 0, then do NOT duplicate the body about the y=0 plane

kbsym(ncol)             If = 1 then duplicate the input geometry number ncol about the z=0 plane

                            If = 0, then do NOT duplicate the body about the z=0 plane

nbminc(ncol)            Minimum number of grid points along the longest body dimension for input geometry number ncol.

                            Grid is refined until nbminc is met and body is successfully resolved.

                            Note that the value of nbminc(ncol) overrides the global value of nbmin specified in input.dat.

                            Note also that nbminc(ncol) for a given body may be set to zero and in this case, the global value of nbmin specified in input.dat is used for that body.

ibccolor(ncol)          If = 1, then geometry number ncol is a stationary solid surface

                            If = 2, then geometry number ncol is a moving solid surface (with rigid-body translation and/or rotation)

If = 3, then geometry number ncol is a flow-through boundary. The fluid flowing through this boundary either has with specified flow properties (e.g. rocket nozzle exit) or has properties which are the same as the surrounding cells (e.g. inlet to an engine)

**nbcnames(ncol)** Name of input file containing boundary condition data for geometry number ncol. If `ibccolor(ncol)=1`, then use name 'dummy' since no additional boundary condition data is needed

**ngaps** Specifies gaps are to be "enforced" between some of the input geometries. When a gap between two bodies is enforced, then the grid cells will be automatically refined enough such that at least one grid cell fits between those two bodies.

If > 0, then ngaps is the number of gaps to enforce. If `ngaps>0` then ngaps more lines are read from `bodyinall.dat`

If = 0, then do not force the geometries to be separated by at least one grid cell. Use `ngaps=0` if the bodies are allowed to be in contact with each other. If `ngaps=0` is used, no subsequent lines are read in from `bodyinall.dat`

**ncolgap(n1,1),ncolgap(n1,2)** Pair of body numbers that should be keep separate (i.e. a finite gap exists between the two bodies). Body numbers `ncolgap(n1,1)` and `ncolgap(n1,2)` will be separated by at least one grid cell. The number of lines of geometry pairs to be read in is ngaps.

---

### **Description: Boundary Condition Data Input File**

File Name: Specified using the variable `nbcname` in `bodyinall.dat`. One file needed for each input geometry, ncol, used.

File Type: ascii

If `ibccolor(ncol) = 1`, then:

This file is not needed

If  $ibccolor(ncol) = 2$ , then:

Read Statements:

read(26,\*) utr(ncol), vtr(ncol), wtr(ncol)

read(26,\*) irotmode(ncol), omegarx(ncol), omegary(ncol), omegarz(ncol)

read(26,\*) thetamax(ncol), fm(ncol)

read(26,\*) xrotm(ncol), yrotm(ncol), zrotm(ncol)

Definitions:

utr(ncol)                    x-component of translational velocity of geometry number ncol [m/sec].

vtr(ncol)                    y-component of translational velocity of geometry number ncol [m/sec].

wtr(ncol)                    z-component of translational velocity of geometry number ncol [m/sec].

irotmode(ncol)if = 0, then no rigid body rotation

If = 1, then constant angular velocity rotation

If = 2, then sinusoidal body rotation

These variables are only used if  $irotmode(ncol) = 1$ :

omegarx(ncol) Angular velocity about x axis for geometry number ncol [deg/sec]

omegary(ncol) Angular velocity about y axis for geometry number ncol [deg/sec]

omegarz(ncol) Angular velocity about z axis for geometry number ncol [deg/sec]

These variables are only used if  $irotmode(ncol) = 2$ :

thetamax(ncol)            Maximum sinusoidal deflection angle for geometry number ncol [ deg]

fm(ncol)                    Frequency of sinusoidal motion for geometry number ncol [1/sec]

These variables are only used if  $irotmode(ncol) > 0$ :

xrotm(ncol) x location of center of rotation for geometry number ncol [m].

yrotm(ncol) y location of center of rotation for geometry number ncol [m].

zrotm(ncol) z location of center of rotation for geometry number ncol [m].

If ibccolor(ncol) = 3, then:

Read Statements:

```
read(26,*) bcconst(1,ncol), bcconst(2,ncol),bcconst(3,ncol),bcconst(4,ncol),bcconst(5,ncol)
```

Definitions:

bcconst(1,ncol) If >0, then all flow properties on this boundary are to be specified by the user (e.g. rocket exit boundary condition). In this case bcconst(1,ncol) is the density on geometry number ncol [kg/m<sup>3</sup>].

If = 0, then the flow properties on this boundary are taken to be the same as the surrounding fluid (e.g. inlet of an engine). In this case, values of the following variables must still be specified but they are never used:

bcconst(2,ncol), bcconst(3,ncol), bcconst(4,ncol), bcconst(5,ncol).

bcconst(2,ncol) x-velocity on geometry number ncol [m/sec].

bcconst(3,ncol) y-velocity on geometry number ncol [m/sec].

bcconst(4,ncol) z-velocity on geometry number ncol [m/sec].

bcconst(5,ncol) Static pressure on geometry number ncol [N/m<sup>2</sup>].

---

### **Description: Turbulent Case Input File (Required only if ivisc = -1)**

File Name: turbin.dat

File Type: ascii

Read Statements:

```
read(1,*) nturb
```

```
read(1,*) turbint
```

read(1,\*) turbintmax  
read(1,\*) xmutr  
read(1,\*) xkk  
read(1,\*) xkeps  
read(1,\*) vk  
read(1,\*) capb  
read(1,\*) prandtl  
read(1,\*) amintk  
read(1,\*) amaxtk  
read(1,\*) aminteps  
read(1,\*) amaxteps  
read(1,\*) yplim

Definitions:

nturb            Iteration number after which turbulent calculations start  
turbint          Freestream turbulence intensity  
turbintmax      Maximum turbulence intensity permissible in the flow  
xmutr            Ratio of the freestream turbulent viscosity to the freestream molecular viscosity  
xkk              Limit for  $(\rho * TKE)$  {Typical: 0.0001}  
xkeps            Limit for  $(\rho * \varepsilon)$  {Typical range: 0.0001 - 0.01}  
vk                von Karman constant (For boundary layers ~0.41, page 305 \*Pope)  
                  \*Pope, S.B., Turbulent Flows, Cambridge University Press, 2000.  
capb             Log-law constant (For boundary layers ~5.2, page 305 Pope)  
prandtl          Turbulent Prandtl number (~1.0, page 371 Pope)  
amintk           Refinement parameter for turbulent kinetic energy. Coarsen grid if variable

	is smaller than amintk
amaxtk	Refinement parameter for turbulent kinetic energy. Refine grid if variable is larger than amaxtk
aminteps	Refinement parameter for TKE dissipation rate. Coarsen grid if variable is smaller than aminteps
amaxteps	Refinement parameter for TKE dissipation rate. Refine grid if variable is larger than aminteps
yplim	Determines boundary layer cell size (Limiting value for $y^+$ )

---

**Description: Surface Plane Location Input File (Required only if iplanes=1)**

File Name: xyzplanes.dat

File Type: ascii

Read Statements:

```

read(1,*) nxplanes, nyplanes, nzplanes
  do n1 = 1,nxplanes
    read(1,*) xplanes(n1)
  enddo
  do n1 = 1,nyplanes
    read(1,*) yplanes(n1)
  enddo
  do n1 = 1,nzplanes
    read(1,*) zplanes(n1)
  enddo

```

Definitions:

nxplanes                    Number of x=constant planes upon which to write out surface data

nyplanes	Number of y=constant planes upon which to write out surface data
nzplanes	Number of z=constant planes upon which to write out surface data
xplanes(n1)	x location of plane upon which to write out surface data
yplanes(n1)	y location of plane upon which to write out surface data
zplanes(n1)	z location of plane upon which to write out surface data

---

### Description: Actuator Disk Input File (Required only if idisk=1)

File Name: disk.in.dat

File Type: ascii

Read Statements:

```

      read(1,*) Disk_system_casename

      read(1,*) ndisk

      do nd = 1,ndisk

      read (1,*)
      read (1,*) ixyztype(nd)
      read (1,*) rdiskd(nd),thdiskd(nd)
      read (1,*) rcutoutd(nd)
      read (1,*) xcdiskd(nd),ycdiskd(nd),zcdiskd(nd)
      read (1,*) ibet(nd)

      if (ibet(nd).eq.0) then
c.....Read in Pressure Jump Distribution for NON-blade element theory cases
          read(1,*)
          read(1,*) nsegdisk(nd)
          read(1,*)
          do ng = 1,nsegdisk(nd)
              read(1,*) rdiskpd(nd,ng),delppd(nd,ng)
          enddo
          endif
      if (ibet(nd).eq.1) then
c.....Read in Blade Element Theory(BET) Parameters
c.....See sample disk.in.dat file for description of these variables
          read(1,*)
          read(1,*) xmuadv
          read(1,*) nbl(nd)

```

```

read(1,*) chord(nd)
read(1,*) clslope(nd)
read(1,*) cl0(nd)
read(1,*) cd2(nd)
read(1,*) cd1(nd)
  read(1,*) cd0(nd)
read(1,*) clmin(nd)
read(1,*) clmax(nd)
read(1,*) a0bl(nd)
  read(1,*) a1bl(nd)
read(1,*) b1bl(nd)
read(1,*) alphas(nd)
  read(1,*) itrim(nd)
read(1,*) iwake(nd)
read(1,*) ctides(nd)

```

endif

Definitions:

ndisk	Number of actuator disks to model
ixyztype(nd)	Orientation of axis of actuator disk number nd 1=x constant plane, 2=y constant plane, 3=z constant plane
rdiskd(nd)	Radius of actuator disk number nd
thdiskd(nd)	Coning angle of rotor disk [deg]
rcutoutd(nd)	Radius of cutout (i.e. hole) in center of actuator disk number nd
xcdiskd(nd),ycdiskd(nd),zcdiskd(nd)	x, y, and z coordinates of center of actuator disk
ibet(nd)	0 = Use Specified Pressure jump for Disk Model 1 = Use Blade Element Theory for Disk Model 3 = Adapt grid to actuator disk but do not apply any flow boundary conditions there
nsegdisk(nd)	Number of radial locations along disk number nd at which to model pressure jump
rdiskpd(nd,ng)	Radial location along disk number nd
delppd(nd,ng)	Static pressure jump [N/m <sup>2</sup> ] across disk at radial location rdiskpd

---

**Description: Chemistry Case Input File (Required only if ichem and/or itherm = 1)**

File Name: data/mixtures/chem\_model.mix

File Type: ascii

Read Statements:

```
read(201,*)
read(201,*)
read(201,*)
read(201,*) nspec
read(201,*)
read(201,*)
do i=1,nspec
    read(201,*) spec_names(i)
enddo
read(201,*)
read(201,*)
do i=1,nspec
    read(201,*) xconc_0(i)
enddo
read(201,*)
read(201,*)
read(201,*)
read(201,*)
read(201,*)
read(201,*)
```

```

read(201,*) ncnr
read(201,*)
do i=1,ncnr
    read(201,*)
    read(201,*) icrate(i)
    read(201,*) ( frc(i,j), j=1,3 )
    read(201,*) ( stm_r(i,j), j=1,nspec )
    read(201,*) ( stm_p(i,j), j=1,nspec )
enddo

```

#### Definitions:

nspec	Number of species
spec_name(nspec)	Name of species input files
xconc_0(nspec)	Initial composition (mole fractions)
ncnr	Number of reactions
icrate(ncnr)	Temperature coupling flag, for the Reaction Rates
	If = 1, $T = \sqrt{T_{\text{translation}} * T_{\text{vibration}}}$
	If = 2, $T = T_{\text{translation}}$
	If = 3, $T = T_{\text{vibration}}$
frc(ncnr,1),frc(ncnr,2),frc(ncnr,3)	Constants in the Forward Reaction Rate expressions
stm_r(ncnr,nspec)	Stoichiometric coefficients of the reactants
stm_p(ncnr,nspec)	Stoichiometric coefficients of the products

**File Name: data/species/spec\_names (Required only if ichem = 1)**

File Type: ascii

Read Statements:

```
read(202,*)
read(202,*)
read(202,*)
read(202,*) vmmol(i)
read(202,*)
read(202,*) hform(i)
read(202,*)
read(202,*) hform_298(i)
read(202,*)
read(202,*) chrg(i)
read(202,*)
read(202,*) ntseg(i)
read(202,*)
read(202,*) ncoeff(i)
read(202,*)
read(202,*)
do j=1,ntseg(i)
    read(202,*) ( poly(i,j,k), k=1,ncoeff(i) )
enddo
read(202,*)
if (itherm.eq.1) then
    read(202,*)
```

```

read(202,*) ipoly(i)
read(202,*)
read(202,*) rdof(i)
read(202,*)
read(202,*) svct(i)

```

endif

#### Definitions:

vmmol	Molecular weight [kg/mole]
hform	Heat of formation at 0K [J/mole]
hform_298	Heat of formation at 298.15K [J/mole]
chrg	Charge on the radical
ntseg	Number of temperature-segments in polynomial curve fit
ncoeff	Number of coefficients in polynomial curve fit
poly(nspec,ntseg,ncoeff)	Coefficients in polynomial curve fits for thermodynamics properties (NASA Ref. Pub. 1232, August 1990)
ipoly	Flag for vibrational internal energy mode (ivib)  If ipoly = 1, vibrational energy included amongst the internal energy modes  If ipoly = 0, no vibrational internal energy mode for this species
rdof	Rotational degrees of freedom
svct	Vibrational characteristic temperature [K]