

# Morph Mesher User Manual

SAND2026-19968W  
Unclassified Unlimited Release

April 17, 2026

FAST Team  
Sandia National Labs <sup>1</sup>

---

<sup>1</sup>Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

## 1 Introduction

Morph options can be specified either on the command line or in an input file. Only a limited number of options can be specified on the command line. All but a few rarely used options can be specified in the input file.

## 2 Input File Requirements

The morph input file must be formatted using the standard yaml format.

The minimum input to run morph is the name of the geometry file morph will read, a global element size, and the name of an output file that morph will create containing the resulting mesh.

## 3 Geometry File

The geometry can be either an ACIS (\*.sat) file, a STEP (\*.stp, or \*.step) file, an STL (\*.stl) file, or an SGM (\*.sgm) file. The full path to the geometry file must be specified. The geometry file is specified in the input file as follows:

```
geometry_file: "path/MyModel.sat"
```

## 4 Output File

The output file is written by morph and contains the output mesh in exodus format. If a file of the specified name already exists it will be overwritten. The file will be spread across the number of mpi procs that morph was run on. The output file is specified in the input file as follows:

```
output_mesh_file: "path/MyMeshedModel.exo"
```

## 5 Global Mesh Size

The global element size is the approximate size of the edges in the resulting mesh. The global element size must be a floating point number greater than 0.0. the global element size is specified in the input file as follows:

```
global_elem_size: 10.2
```

## 6 Referencing Geometry

Any morph input file option that references a CAD entity (i.e. volume, surface, curve or vertex) can refer to the CAD entity by name or by id. Names are strongly recommended since ids can change unexpectedly. Use of wildcards is supported for specifying CAD entities by name.

CAD volumes can be referenced by name in the morph input file with any or all of the following options:

```
volume_names: ["Housing1", "Housing2", "Housing3"] # 3 volumes
volume_names: ["Housing*"] # all names starting with Housing
volume_names: [Housing*] # all names starting with Housing
volume_names: [1Bolt,2Bolt] # most names do not need quotes
volume_names: ["*Bolt"] # needs quotes due to leading wildcard
```

CAD surfaces, curves, and vertices can be referenced by name in the morph input file with the `surface_names`, `curve_names`, and `vertex_names` keywords using the same syntax as above for volumes.

CAD volumes can be referenced by id in the morph input file with any or all of the following options:

```
volumes: [17,24,8]
volumes: [34]
volume_ranges: [[2,4], [120,300]]
```

CAD surfaces, curves, and vertices can be referenced by id in the morph input file with the `surfaces`, `surface_ranges`, `curves`, `curve_ranges`, `vertices`, and `vertex_names` keywords using the same syntax as above for volumes.

## 7 Geometry Engines

By default, morph will use the ACIS geometry engine when given a SAT or STEP file. Optionally, morph will use the SGM geometry engine for STEP or SGM files if the following is specified in the input deck:

```
sgm: true
```

## 8 Faceted Geometry

This option controls how morph does geometric evaluations with the CAD engine. If OFF, morph asks the CAD engine directly for all geometric evaluations. This results in direct analytic and spline evaluations of the CAD BREP surfaces, which can be slow, especially if the model has spline surfaces. If ON, morph asks the CAD engine for the graphics facets of the model and then does the geometric evaluations on the facets avoiding the expensive geometry evaluation calls on the CAD engine.

The default is ON for ACIS geometry engine and OFF for the SGM geometry engine. You can also manually control the behaviour like this:

```
use_faceted_geometry: off
```

## 9 Tet 10 higher order elements

By default, morph outputs tet4 (4-noded tetrahedral) elements. The 'tet10' option specifies that the resulting mesh should have 10-noded tetrahedral elements. The midnodes are projected to the geometry unless the resulting neighbor elements have a normalized inradius quality value less than the value specified with 'tet10\_normalized\_inradius\_straightening\_threshold'. Here is an example of using both of these options:

```
tet10: on
tet10_normalized_inradius_straightening_threshold: 0.15
```

## 10 Sidesets

Sidesets can be added to a morph mesh through the morph input file only. There are no command line options to add sidesets.

Morph will add sidesets to CAD surfaces by adding a "sidesets" section in the morph input file. Any number of sidesets can be specified. Each sideset must have its own sub-section under the "sidesets" section. Each sideset is specified with a collection of CAD surfaces, an id, and a name:

```
id: <integer> # required
name: <string> # optional
surface_names/surfaces/surface_ranges: # required
wrt: <OWNING|opposing_blocks|all_touching_blocks> # optional
```

The wrt keyword is optional and indicates the sense of the sideset with respect to the adjacent blocks. OWNING will be used if wrt is not specified and will create the sideset with respect to the block of the volume the specified surfaces are topologically connected to. If wrt is specified as opposing\_blocks, the sideset will be created with respect to the block opposing the topologically connected volume. If wrt is specified as all\_touching\_blocks, the sideset will be created with respect to all adjacent blocks.

Below is an example of a sidesets section in a morph input file that will create 2 sidesets:

```
sidesets:
- id: 13
  name: pressureSurface
  surface_names: ["PressureLoad*"]
- id: 17
  name: temperatureBC
  surface_names: ["Heat*Source"]
  wrt: opposing_blocks
```

This will create 2 sidesets. Sideset 13 will be named pressureSurface and contain all surfaces with the naming pattern of "PressureLoad\*". Sideset 17 will be

named temperatureBC and contain all surfaces with the naming pattern of "Heat\*Source"

## 11 Local Refinement

The mesh built by morph will be refined through the morph input file only. There are no command line options to refine the mesh.

Morph refines the resulting mesh if a "refine\_at" section is added in the morph input file. Any number of refinements can be specified. Each refinement must have its own sub-section under the "refine\_at" section. With the exception of 2 custom refinements (i.e. surface\_curvature and sizes\_from\_file), each refine\_at sub-section must have a location indicator (i.e. where to refine), and a target size desired at that location.

The target size is specified with the "size" keyword as follows:

```
- size: <double>
```

supported location indicators include:

```
- location: [<double-x>, <double-y>, <double-z>]
- bounding_box:
  min: [<double-x>, <double-y>, <double-z>] # min corner
  max: [<double-x>, <double-y>, <double-z>] # max corner
- line_segment:
  start: [<double-x>, <double-y>, <double-z>]
  end: [<double-x>, <double-y>, <double-z>]
- bounding_sphere:
  center: [<double-x>, <double-y>, <double-z>]
  radius: <double>
- volume_names
- volumes
- volume_ranges
- surface_names
- surfaces
- surface_ranges
- curve_names
- curves
- curve_ranges
- vertex_names
- vertices
- vertex_ranges
```

Below is an example of a refine\_at section with 3 refinements specified:

```
refine_at:
- surface_names: ["BCSurface"]
  size: 4.5
```

```

- location: [-3.4, 5.1, -0.02]
  size: 1.25
- vertex_names: ["PointLoadLocation"]
  size: 5.3e-2

```

## 11.1 Local Refinement Using Sizes From File

Morph supports an advanced refinement feature to honor sizes specified in an input exodus file. The syntax of this option is:

```

refine_at:
- sizes_from_file: "/path/MeshWithNodalSizeField.exo"
  nodal_size_field_name: MyNodalSizeField

```

In this example, the specified exodus file must contain a nodal size field named `MyNodalSizeField`. Morph will refine the resulting mesh according to the sizes stored in the nodal field.

The `sizes_from_file` option can be combined with other types of local refinement specifications.

## 12 Interfaces

By default, morph meshes everything contiguously, meaning adjacent elements associated to different volumes will share the same nodes. The "interfaces" section of the input file allows you to specify sets of volumes that should be disconnected from other sets of volumes. Disconnecting volumes can be useful when two volumes are physically close to each other but should not actually be touching or when you want the simulation code to compute contact between two volumes.

In the following example, any element in a volume that has a name starting with 'bolt' or 'washer' will be disconnected from the volume named 'frame'. In addition, elements in the volume 'fred' will be disconnected from elements in the volume 'bob':

```

interfaces:
- disconnect:
  volume_names: [bolt*, washer*]
  from:
  volume_names: [frame]
- disconnect:
  volume_names: [fred]
  from:
  volume_names: [bob]

```

If you don't want any volumes in the mesh to be meshed contiguously with other volumes, this can be specified with the keyword "disconnect\_all" as shown here:

```
interfaces:  
  - disconnect_all
```