SealMesh User's Manual
Release 5.0

Turbomachinery Laboratory
Texas A&M University

June 5, 2024

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This manual describes how to install and run the `SealMesh` software package. This software package consists of the following codes: (i) `SealMesh`, the main code that generates the grid in .ugrid format, (ii) `prep`, a code that, using the .ugrid file, generates the .mesh and .c2n files needed to run the `UNS3D` CFD solver, (iii) `splitmesh`, a code that splits the .mesh and .c2n files for parallel processing, and (iv) `splitugrid`, a code that splits the .ugrid file for parallel processing, and (v) `parallel_prep`, a code that generates .mesh and .c2n files from the splitted .ugrid files for parallel processing. The sequence of codes and file types used for grid generation is shown in Table 1.1.

Table 1.1: Sequence of codes and file types used for grid generation. xxx denotes multiple files used for parallel processing.

(i) `SealMesh` | .ugrid			
"Small" grids		"Large" grids	
(ii) `prep`	.mesh, .c2n	(iv) `splitugrid`	.ugrid_xxx
(iii) `splitmesh`	.mesh_xxx, .c2n_xxx	(v) `parallel_prep`	.mesh_xxx, .c2n_xxx

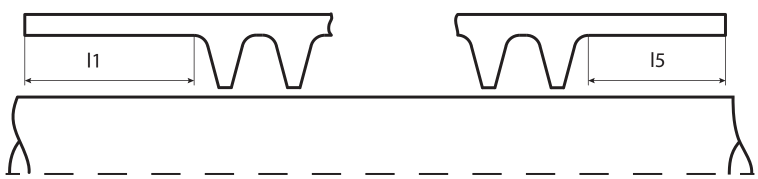
The sequence of codes (ii) and (iii) is limited to smaller size grids, while the sequence of codes (iv) and (v) can be applied to grids of any size [1].

[1] To date, the largest grid for which the sequence of codes (iv) and (v) was applied to had more than 300 million cells

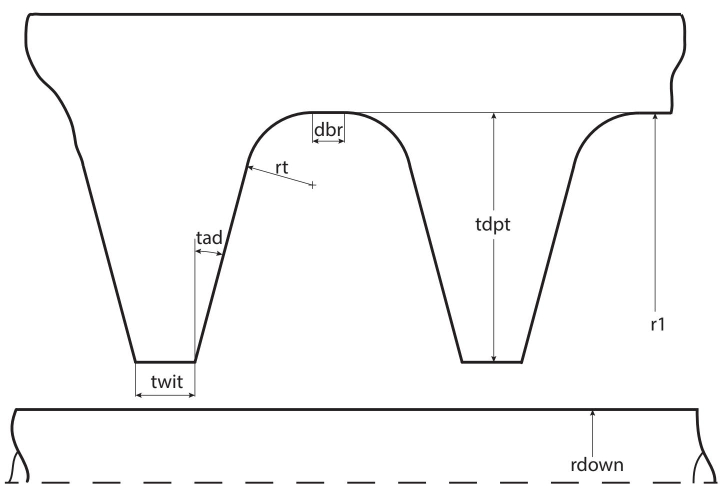
Chapter 2

SealMesh

The `SealMesh` code generates the grid for a labyrinth or pocket damper seal. The seal grid can span the full annulus or only a sector of the annulus. The typical geometry of the seal along the axial direction is shown in Fig. 2.1.



(a) Inlet and outlet dimensions.



(b) Seal detail.

Figure 2.1: Seal geometry.

2.1 SealMesh input file

2.1.1 Labyrinth seal

`SealMesh` accepts a single `.dat` input file to generate grids with geometries and node distributions defined by the user. A formatted example is shown below:

Listing 2.1: inputlaby.dat

```
$input_data           ! all dimensions in m

  !-------------- SEAL GEOMETRY --------------
  angle= 360.d0          ! sector angle
  rdown = 57.15d-3       ! shaft radius
  r1    = 61.646d-3      ! radius stator at inlet
  ntee  = 4              ! number of teeth
  l1    = 30.0736d-3     ! length from inlet to first tooth
  rt    = 1.016d-3       ! radius fillet
  tad   = 15.d0          ! tooth angle (degrees)
  tdpt  = 4.293d-3       ! tooth depth (height)
  twit  = 0.254d-3       ! tooth width at tip
  dbr   = 0.18d-3        ! length between fillet radia
  l5    = 30.0736d-3     ! length from last tooth to exit


  !------------ NODE DISTRIBUTION ------------
  i1   = 26           ! # elements from inlet to tooth filet
  i21  = 12           ! # elements over filet
  i22  = 18           ! # elements from filet to tip
  i3   = 8            ! # elements over tooth tip
  i4   = 2            ! # elements over transition between teeth
  i5   = 26           ! # elements from last tooth filet to outlet
  jmax = 63           ! # nodes in j direction (radial)
  equal=.false.       ! set .true. to space evenly in radial direction
  y1lower = 1.d-6     ! y1 for lower boundary
  y1upper = 1.d-6     ! y1 for upper boundary


  !---------------- SEAL TYPE ----------------
  is_damper =.false. ! set .true. to generate damper

  kmax  = 145         ! # nodes in k direction (tangential)


  !----------- BOUNDARY CONDITIONS -----------
  idbcstator = -41    ! idbcs for stator
```

```
  idbcrotor   = -4     ! idbcs for rotor (shaft)


  !-------------- DISPLACEMENT ---------------
  displ = 0            ! 0-1 percent of the clearance to displace
  yc = 0               ! unit vector pointing in displacement direction
  zc = 1               ! unit vector pointing in displacement direction
  ! clearance = 0      ! Code will calculate this if not provided

$END !
```

The basic geometry of the seal is defined along the axial direction according to the parameters in the “SEAL GEOMETRY” section of the input file shown in Listing 2.1. Figure 2.1 illustrates these geometric parameters. If angle=360 the grid for the full annulus is generated, otherwise only a sector of angle is generated.

Parameters defined in the ”NODE DISTRIBUTION” section determine the node distribution in the axial cross sections. i1 specifies the number of elements the seal inlet of length l1 is divided into. i21 specifies the number of elements the filet of radius rt is divided into. i22 specifies the number of elements the side of the tooth with slope tad is divided into. i3 specifies the number of elements the top of the tooth of dimension twit is divided into. i4 specifies the number of elements the gap between teeth dbr is divided into. i5 specifies the number of elements the seal outlet of length l5 is divided into. jmax specifies the number of nodes in radial direction. If equal=.true. the grid in radial direction is equally spaced. If equal=.false. the radial displacements of nodes are defined by y1lower and y1upper (alongside jmax), which define the distance between the innermost and outermost nodes, respectively.

To define the geometry and nodal distribution in the tangential direction, this input file defines the parameter is_damper as .false., such that an axisymmetric labyrinth seal is represented in the grid geometry with kmax equally-distributed nodes spanning angle degrees along the tangential direction.

The parameters idbcstator and idbcrotor define the boundary conditions along the stator and rotor, respectively, with boundary condition identification values which are needed for UNS3D, the CFD code available to TRC members.

The eccentricity of the rotor is specified by displ, which is the ratio of the clearance between the tips of the seal’s teeth and the rotor to displace the rotor. yc and zc are the components of the unit vector that defines the direction the rotor is displaced in the x-plane with respect to the stator.

Figure 2.2 includes images of the grid generated from the input file shown in Listing 2.1.

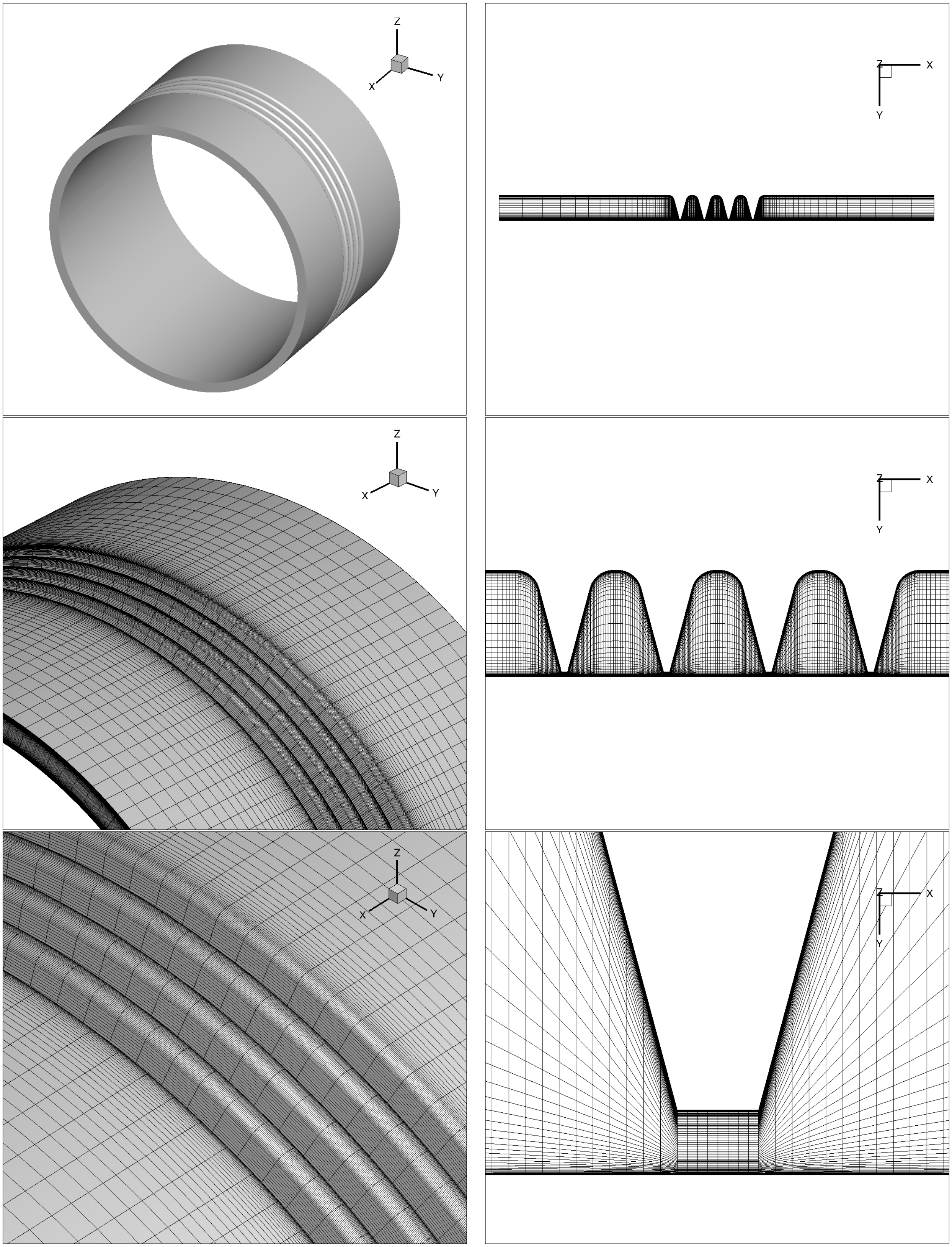


Figure 2.2: `SealMesh` labyrinth seal

2.1.2 Pocket damper seal

The pocket damper seal has several input parameters that are identical to those of the labyrinth seal. In addition, the pocket damper seals has some specific input parameters that define the number and geometry of the bridges. The following example of an input file shows these parameters.

Listing 2.2: inputpock.dat

```
$input_data          ! all dimensions in m

  !-------------- SEAL GEOMETRY --------------
  angle= 360.d0         ! sector angle
  rdown = 57.15d-3      ! shaft radius
  r1    = 61.646d-3     ! radius stator at inlet
  ntee  = 4             ! # teeth
  l1    = 30.0736d-3    ! length from inlet to first tooth
  rt    = 1.016d-3      ! radius fillet
  tad   = 15.d0         ! tooth angle (degrees)
  tdpt  = 4.293d-3      ! tooth depth (height)
  twit  = 0.254d-3      ! tooth width at tip
  dbr   = 0.18d-3       ! length between fillet radia
  l5    = 30.0736d-3    ! length from last tooth to exit


  !------------ NODE DISTRIBUTION ------------
  i1   = 26          ! # elements from inlet to tooth filet
  i21  = 12          ! # elements over filet
  i22  = 18          ! # elements from filet to tip
  i3   = 8           ! # elements over tooth tip
  i4   = 2           ! # elements over transition between teeth
  i5   = 26          ! # elements from last tooth filet to outlet
  jmax = 63          ! # nodes in j direction (radial)
  equal=.false.      ! set .true. to space evenly in radial direction
  y1lower = 1.d-6    ! y1 for lower boundary
  y1upper = 1.d-6    ! y1 for upper boundary


  !---------------- SEAL TYPE ----------------
  is_damper = .true. ! set .true. to generate damper

  num_pockets   = 4     ! number of pocket-separating bridges
  bridge_angle  = 3.d0  ! bridge (between pockets) size in degrees
  btrans_angle  = 1.d0  ! size of transition to and from bridge

  k3 = 20            ! # elements tangential on bridge
  k1 = 80            ! # elements tangential on pocket
```

```
  !----------- BOUNDARY CONDITIONS -----------
  idbcstator = -41   ! idbcs for stator
  idbcrotor  = -4    ! idbcs for rotor (shaft)

  !------------- DISPLACEMENT ---------------
  displ = 0          ! 0-1 percent of the clearance to displace
  yc = 0             ! unit vector pointing in displacement direction
  zc = 1             ! unit vector pointing in displacement direction
  ! clearance = 0    ! Code will calculate this if not provided
$END !
```

Listing 2.1 shows an input file for generating a grid corresponding to a labyrinth seal, along with more detailed documentation of most of the parameters shown above. For the case of a pocket damper seal, when `is_damper=.true.`, different parameters are shown in the "`SEAL TYPE`" portion of the input file.

The input files of the labyrinth and pocket damper seals are identical except for the `SEAL TYPE` section, as are their axial cross sections everywhere except the region containing and transitioning into the "bridges" between the pockets of the pocket damper seal.

`num_pockets` defines the number of pockets to equally space along the circumference of the seal. The "bridges" between these pockets have sizes defined by `bridge_angle` in degrees, with a smooth transition spanning `btrans_angle` degrees on each side. `k3` defines the number of elements spanning the tangential length of the bridge (`bridge_angle` degrees), while `k1` defines the number of elements spanning the tangential length of each pocket. `kmax` is not necessary in this case.

Figure 2.3 includes images of the grid generated from the input file shown in Listing 2.2.

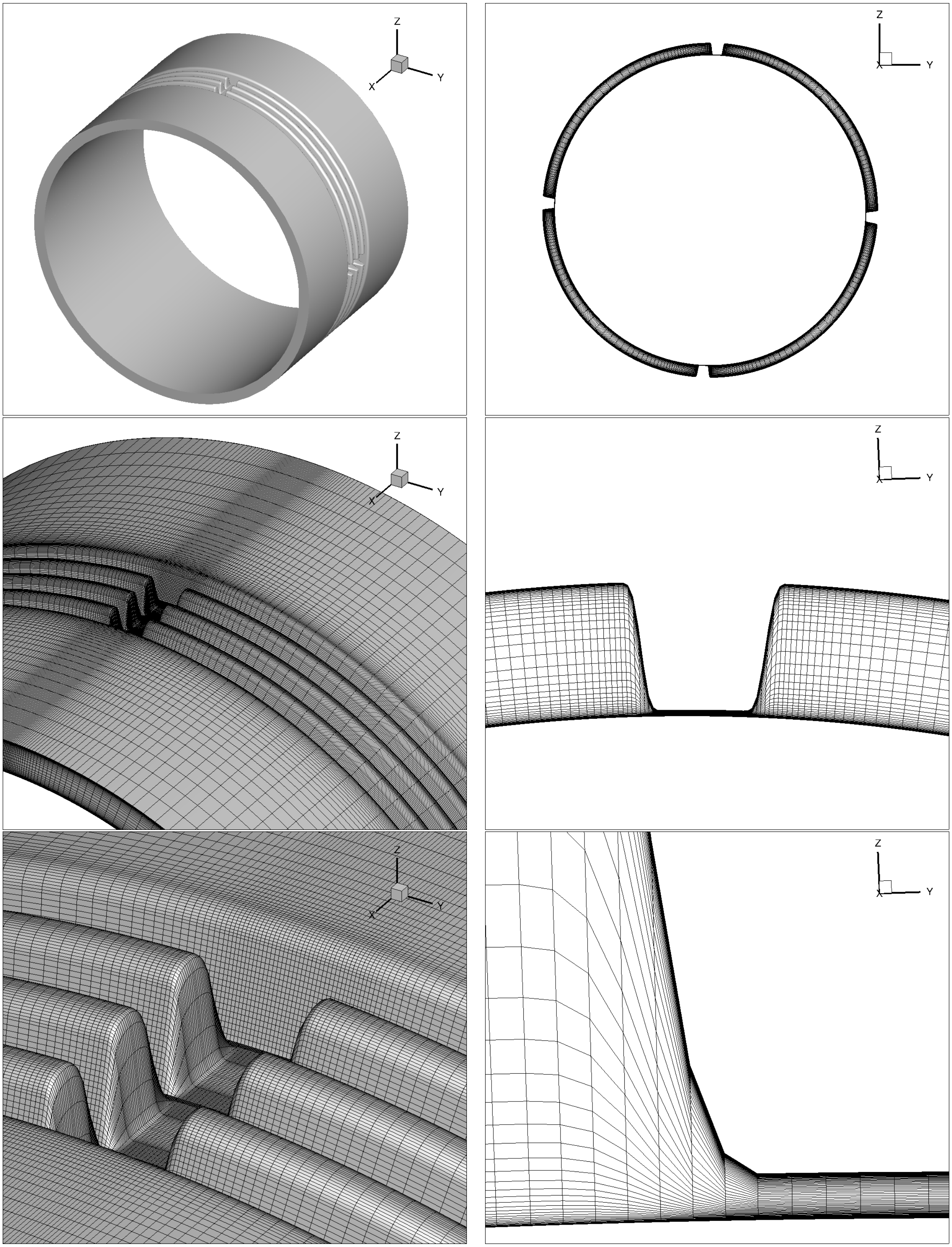


Figure 2.3: `SealMesh` pocket damper seal