The CML Air Bearing Design Program (CMLAir), Version 7 User Manual

By

Brendan Cox and David B. Bogy

Revision of Earlier Manual by Ryan Grisso and D.B. Bogy

Department of Mechanical Engineering University of
California Berkeley, CA 94720

December, 2007

Table of Contents

1. Introduction 3
1.1. Solvers 3
1.2. Windows Interface 3
2. Installation 3
3. The CML Air Bearing Design Program for PC 3
3.1. Static Mode 4
3.1.1. Static Mode - File Menu 4
3.1.2. Static Mode – Options Menu 5
3.1.2.1. Solver Interaction 5
3.1.2.1.1. Local Solver 5
3.1.2.2. Solver Status 6
3.1.3. Static Mode - Preprocessor Menu 6
3.1.3.1. Static Mode - General Window 6
3.1.3.1.1. Slider Geometry 6
3.1.3.1.2. Initial Flying Attitude 7
3.1.3.1.3. Suspension 7
3.1.3.1.4. Points of Interest 8
3.1.3.1.5. Convergence 8
3.1.3.1.6. Comments 9
3.1.3.1.7. Geometry Modifications 9
3.1.3.2. Rails Window 10
3.1.3.2.1. Rail Points 11
3.1.3.2.2. Rail Type 11
3.1.3.2.3. Recess Height 11
3.1.3.2.4. Base Recess 12
3.1.3.2.5. Rail Index 13
3.1.3.2.6. Add, Delete, Update, and Rail Tools 13
3.1.3.2.7. Rail Diagram 14
3.1.3.2.8. Overlapping Rails 14
3.1.3.2.9. Rail Manipulation 14
3.1.3.3. Wall Profiles Window 15
3.1.3.4. Static Mode - Partial Contact Window 15
3.1.3.5. Grid Window 16
3.1.3.5.1. Rectangular Computation Grid 16
3.1.3.5.2. Rectangular Adaptive Grid 16
3.1.3.5.3. Rectangular Grid Geometric Series 17
3.1.3.6. Static Mode - Run Setup Window 17
3.1.3.7. MultiModule 19
3.1.3.7.1. Definitions 19
3.1.3.7.2. Sample and Search Space 20
3.1.3.7.3. MultiModule Window 21
3.1.3.8. Static Track Seek Simulation 22
3.1.3.9. Python Interpreter 23
3.1.3.9.1. Python Interpreter Example 23
3.1.4 Static Mode – Postprocessor Menu 25
3.1.5. Windows Menu 26
3.1.6. Help Menu 26
3.2. Dynamic Mode 27
3.2.1. File Menu 27
3.2.2. Options Menu 27
3.2.3. Dynamic Mode - Preprocessor Menu 27
3.2.3.1. Dynamic Mode - General Window 27
3.2.3.1.1. Slider Geometry 27

3.2.3.1.2 Slider Mass 27
3.2.3.1.3. Initial Flying Condition 28
3.2.3.1.4. Points of Interest 29
3.2.3.1.5. Convergence 29
3.2.3.1.6. Time 29
3.2.3.2. Dynamic Mode - Rails Window 29
3.2.3.3. Dynamic Mode - Wall Profiles Window 29
3.2.3.4. Dynamic Mode - Partial Contact Window 29
3.2.3.5. Dynamic Mode - Grid Window 30
3.2.3.6. Dynamic Mode - Run Setup Window 30
3.2.3.6.2. Actuator 31
3.2.3.6.3. Suspension Loading 32
3.2.3.6.4. Track Seeking Motion 33
3.2.3.6.5. Disk Velocity Profile 33
3.2.3.6.6. Disk Flutter 33
3.2.3.6.7. Shock 33
3.2.3.6.8. Parameter Identification Program (PIP) 34
3.2.3.7. Disk and Slider Surface Topography Window 35
3.2.3.7.1. Numerical Generation of Topography 35
3.2.3.7.1.1. Disk and Slider Waviness/Texture 35
3.2.3.7.1.2. Disk and Slider Asperities 36
3.2.3.7.1.3. Disk Zone Profiles 38
3.2.3.7.1.4. Output Disk Topography 38
3.2.3.7.1.5. Point by Point Disk Track Profile 38
3.2.4. Dynamic Mode - Postprocessor Menu 39
4. Input and Output Files 39
4.1. Quick Input Files 39
4.1.1. rail.dat 40
4.1.2. run.dat 41
4.2. Quick Output Files 43
4.3. Dyn Input Files 44
4.4. Dynamic Output Files 49

1. Introduction

This report is an updated manual for the CML Air Bearing Design Program version 7.0. This manual is also available as an online help file from the "Help" menu of the Windows interface and is included with the distribution. While the Air Bearing Design Solver Quick4 and the CML Air Bearing Dynamic Simulator are actually separate programs, they can all be used in an integrated environment with the Air Bearing Design Program CMLAir.

Strictly speaking, you do not need the graphical interface in order to use CML simulation software. All functions of the solvers can be accessed using ASCII input and output files, however, users will find that the graphical interface vastly simplifies the air bearing design and analysis process.

1.1. Solvers

The current distribution of the Air Bearing Design Program for PC is packaged with two solvers: Quick4, and Dyn4, also known as the CML Air Bearing Rectangular Grid Solver, and the CML Air Bearing Dynamic Simulator. These solvers are written in Fortran 95 and have been compiled as 32-bit Windows console applications using the Intel Fortran Compiler. These solvers are included in the CMLAir installer, if you plan to run simulations using only a Windows 2000/XP/Vista computer, you will not need to download any additional files.

1.2. Windows Interface

The Windows interface can be run on any Windows PC system. The interface allows for visual pre- and post-processing of air bearing designs. Features include point and click rail design, extensive error checking, visual grid construction, 2-d and 3-d post-processing, plotting and analysis tools as well as tools for automating the study of air bearing sliders.

2. Installation

The current version of the CMLAir installer can be found online at http://cml.me.berkeley.edu/quick.html. To install CMLAir 7.0 simply download and run the installer. Most of the default options should be suitable for a typical installation. Previous versions of CMLAir suggested that the user uninstall all existing versions of CMLAir, however this is not necessary when working on the more recent versions of Microsoft Windows.

3. The CML Air Bearing Design Program for PC

The CML Air Bearing Design Program can be invoked by double clicking the program icon for CMLAir in Windows. When the interface starts, you will see a window with the CML logo, a menu bar, and a tool bar. The Air Bearing Design Program for PC has two modes: Static and Dynamic. These modes correspond to the two different solvers: Quick 4 and Dyn 4. Each mode has different pre-processing and post-processing functions. First, a few words about file management: In early versions of CMLAir (versions 3 and 4), each case was composed of several input files. The input files for each case would be stored in their own directory. Starting in version 5, all input and output files are bundled in a single ".cml" file. In CMLAir versions 5 and 6, the file format of .cml files was a compressed binary format particular to CMLAir and could not be modified by users. Starting in CMLAir version 7.0, all input files are bundled together as a PKZIP archive. This compressed format allows for .cml files to be small while allowing a simple, published, compressed file format that other programs can use to interoperate with CMLAir files. When a simulation terminates, the ASCII result files generated by the solvers are imported automatically into the ".cml" file. There is no need to save the case after a simulation terminates since all data is automatically saved by the interface. To import a set of ASCII input and output files into a .cml file, choose the "Import ASCII" function from the File menu. To save a case as plain text ASCII files, choose the "Export ASCII" from the File menu.

3.1. Static Mode

Static mode is used for the design and steady state simulation of air bearing sliders. The static mode menu bar selections are as follows: *File*, *Options*, *Preprocessor*, *Postprocessor*, *Windows*, and *Help*. The static mode toolbar selections are as follows: *General*, *Rails*, *Walls*, *Partial Contact*, *Grid*, *Run Setup*, *Dynamic Mode*, *Open*, *Save*, *Solver*, and *Run*. When in Dynamic Mode, Static Mode is enabled by pressing either the *Static or Dynamic* buttons or choosing *Static Mode* from the *Options Menu*.

3.1.1. Static Mode - File Menu

The selections in the static mode file menu are as follows: *New*, *Open Case*, *Save Case*, *Save Case As*, *Import ASCII*, *Export ASCII*, *Preferences*, and *Exit*.

New: Creates a new case with the default parameters.

Open Case: Opens a previously saved case. This can also be accessed from the toolbar.

Save Case: Saves the current case. This can also be accessed from the toolbar.

Save Case As: Saves the current case under a different filename. Results will not be saved in the new file. To copy an entire case, including results, use the Windows Explorer to Copy and Paste the file to a new location.

Import ASCII: Imports a directory with ASCII input and output files. To import a static case, choose "Import ASCII/Static Case". To import a dynamic case, choose "Import ASCII/Dynamic Case". All input and output ASCII files are imported into a a .cml file.

Export ASCII: Exports a case, including all output files, to the current directory. For instance, if you choose to export a Quick4 case after running a fixed-attitude simulation, the following files will be written to the directory where the case file is located: `rail.dat`, `run.dat`, `x.dat`, `y.dat`, `press01.dat`, `geom.dat`. In general, you will only need to use this function if you need to do processing outside the Windows interface (in Matlab, for instance).

Application Preferences: Allows modification of general preferences related to the running of CMLAir. Currently there is only one Application Preference option. *Undo Depth* specifies the number of successive operations in the rails or walls window that can be undone.

Post Preferences: Allows modification of post-processing preferences. These preferences include *Normalize 3-d Plots* and *Background Color*. The Text Viewer is obsolete and input to this box will be ignored. *Normalize 3-d Plots* toggles the normalization of pressure plots. *Background Color* modifies the default background color of plots.

File Preferences: *Warn when loading files containing unknown parameters:* When loading files created by newer versions of CMLAir (i.e. opening a case created by CMLAir 7.01 in CMLAir 7.0), it is possible that the file will contain parameters that the older software does not recognize. Enabling this option allows the program to warn the user when they try to open a file containing unknown parameters.

Load default parameters from file: If the user specifies a .cml file in this edit box, CMLAir will load all input values from the specified file (instead of using the standard CMLAir defaults) when CMLAir starts up or the user selects *New* from the File menu. Standard CMLAir default values will be used if the file cannot be found or if no file is specified.

Export Spacing Map: Will export a map of the spacing between the slider and disk (see section 3.1.4) when exporting ASCII files (*File->Export ASCII)*

Exit: Exits the program. User will be prompted to save the current case if changes have been made.

3.1.2. Static Mode – Options Menu

The selections in the the Static Mode Options Menu are as follows: *Solution*, *Normalized*, *Static Mode*, *Dynamic Mode*, *Solver Interaction*, *Run*, *Batch Run*, *Toolbar*, *Small Toolbar*, and *Error Checking*.

Solution: Allows the user to choose the solution type: *Rail Geometry*, *Fixed Attitude*, or *Fly Height*. *Rail Geometry* calculates only the defined rail geometry and does not solve for pressure or fly height. Users may wish to use this option in the design process to ensure the geometry is as intended. *Fixed Attitude* calculates the pressure for a given attitude as entered in the *Initial Flying Attitude* section of the pre-processor. *Fly Height* determines the steady state fly height for the design.

Normalized: Toggles normalized pre-processor coordinates.

Static Mode: Switches to static mode. A check next to this selection indicates the program is in static mode.

Dynamic Mode: Switches to dynamic mode. This can also be accessed from the toolbar.

Solver Interaction: Activates the Solver Interaction dialog box (see section 3.1.2.1). This can also be accessed from the toolbar.

Run: Runs the case. This can also be accessed from the toolbar.

Batch Run: Allows for multiple cases to be selected and run sequentially. In the *Batch Run* dialog, select files to be run by pressing *Add File* or dragging Files from Windows Explorer and dropping them over the open dialog. Press *Run Batch* to queue the selected files for simulation.

Toolbar: Toggles the visibility of the toolbar.

Error Checking: Toggles pre-processor limit checking. If this feature is activated, the pre-processor will warn you if your input parameters are beyond the recommended values. You may wish to deactivate this feature if you wish to specify a design with non-standard values (such as extremely large or small sliders, very high RPMs, or very high or low loads).

3.1.2.1. Solver Interaction

This window allows the user to control how the interface interacts with the solver. As of CMLAir 7.0, the remote solver capabilities of CMLAir have been removed. With the decreasing use of UNIX computers for numerical simulation and air bearing analysis, it was found that few (if any) users were actively using the remote solver capabilities in CMLAir.

For the user, it is important to be aware of the process through which simulations are run. When a run begins, a temporary directory is created in the same directory as the current .cml document. The ASCII input files are exported to this temporary directory, the solver begins running and reads in the contents of these ASCII files. When the simulation is complete, the ASCII result files output by the solver are imported into the interface, stored in the .cml file, and the temporary directory is removed. Sometimes, the temporary directory cannot be removed after a job is finished. Typically this is caused by open files in the temporary directory that cannot be deleted. In this case, a warning message will be displayed to the user and a warning will also appear when the case is run again. This is normal and should not cause any problems.

3.1.2.1.1. Local Solver

Local Quick4 Loc.: Location of the local Quick4 executable. This solver is used for a local static simulation using a rectangular grid. The default will be the solver that was distributed with the interface. You may choose another solver by typing in the new location or pushing the Browse button to choose the new solver location.

Local Quick5 Loc.: Location of the local Quick5 executable. This solver has been discontinued by CMLAir however remnants of it remain in the interface as we prepare to release a new triangular mesh based static solver.

Local Dyn4 Loc.: Location of the local Dyn4 executable. The default will be the solver that was distributed with the interface. You may choose another solver by typing in the new location or pushing the Browse button to choose the new solver location.

Max Simultaneous Local Solvers: CMLAir is a multithreaded application and allows multiple simulations to be run at the same time on the host machine. This allows users to make the most use of multi-processor and multi-core machines. The *Max Simultaneous Local Solvers* option limits the number of simulations running on the host PC. If more jobs are submitted, the excess simulations are queued and run when running simulations finish and solvers become available.

Solver Priority: Sets the scheduling priority the operating system gives to the solvers. Setting this value to *Low* allows the user to run other applications while a simulation is running the same machine. This usually results in longer simulation times. Setting this value to *High* can improve simulation times but may cause a computer to become unresponsive.

3.1.2.2. Solver Status

This window displays the currently running jobs. Each job window may be shown or hidden by choosing the corresponding job entry and pressing the *Show* or *Hide* buttons. To stop a job, choose the corresponding entry and press the *Stop* button. For MultiModule and other multi-job runs (i.e. track seek or PIP simulations), this will stop the entire multi-job run.

3.1.3. Static Mode - Preprocessor Menu

The preprocessor menu has the following selections: *General*, *Rails*, *Wall Profiles*, *Partial Contact*, *Grid*, and *Run Setup*. All of these selections are also available from the toolbar.

3.1.3.1. Static Mode - General Window

The *General Dialog* box allows for the input of the following categories of input parameters: *Slider Geometry*, *Initial Flying Attitude*, *Suspension*, *Convergence*, *Points of Interest*, and *Comments*.

3.1.3.1.1. Slider Geometry

Length: Slider length in mm.

Width: Slider width in mm or normalized to length.

Height: Slider height in mm or normalized to length.

Crown: Parabolic deformation of the slider in the length direction. A positive crown indicates a convex deformation (smaller spacing at slider center), while negative crown indicates a concave deformation (larger spacing at slider center). Crown is calculated as follows:

Displacement at $(x, y) = 4\,R\,(x/L - (x/L)^2)$

where L is the slider length and R is crown.

Camber: Parabolic deformation of the slider in the width direction. A positive camber indicates a convex deformation (smaller spacing at slider center), while negative camber indicates a concave deformation (larger spacing at slider center). Camber is defined as follows:

Displacement at $(x, y) = 4\,A\,(y/W - (y/W)^2)$

where W is the slider width and A is camber.

Twist: Parabolic deformation of the slider causing equal displacement of diagonal corners. A positive twist indicates the inner leading edge and outer trailing edge are recessed (larger spacing), while the outer leading and inner trailing edges are raised (smaller spacing). Twist is defined as follows:

Displacement at $(x, y) = 4\,T\,(x/L - 0.5)(y/W - 0.5)$

where W is the slider width, L is the slider length, and T is twist.

Taper Length: The X coordinate (in mm or normalized to length) at which the taper begins. This creates a wedge with the given taper angle that will be removed from the front of the slider. In some cases, the user may wish to model a taper that is machined before the etching process, resulting in a recessed ramp in the front of the slider. This area can be modeled by defining a ramp in the Rails window (see section 3.1.3.2.2).

Taper Angle: The angle in µrad at which the tapered region cuts through the front of the slider.

3.1.3.1.2. Initial Flying Attitude

For the Fixed Attitude solution, these values define the attitude at which the solution is sought. For the Fly Height solution (inverse solution), these values define the initial values at which the fly height solution will be sought. The closer these values are to the actual solution, the shorter the calculation should be.

TEC: The height in nm of the nominal trailing edge center. The nominal trailing edge center is the point at y = 0.5W, at a recess of zero. This does not take into account crown, camber, or twist and the point may not be a physical point on the slider, in the case for many modern slider designs. To instruct the solver to track and output fly heights at physical points on the slider, define Points of Interest (see section 3.1.3.1.4). Note that this value does not have to be positive. It can be negative without contact between the slider and the disk. In some cases, it is convenient to define a negative TEC value to obtain faster convergence.

Pitch: The angular displacement in µrad about the y-axis. For positive pitch, the leading edge spacing is larger than the trailing edge spacing.

Roll: The angular displacement in µrad about the x-axis. For positive roll, the inner edge spacing is larger than the outer edge spacing.

3.1.3.1.3. Suspension

The parameters in this section are needed for the Fly Height solution. The solution will balance air and contact pressure against the load in the z-direction and pitch torque and roll torque about the load point.

Load: Suspension load in grams.

POffset: Load point offset in length direction measured in µm from the center of the slider. Positive value moves load point toward the trailing edge.

ROffset: Load point offset in width direction measured in µm from the center of the slider. Positive value moves load point toward the outer edge.

Use Static Torques: Instructs the solver to apply constant static torques on the slider. These values do not change with a change in flying attitude of the slider.

PTorque: Static pitch torque in µN-m, this will only be used by the solver if *Use Static Torques* is selected.

RTorque: Static roll torque in µN-m, this will only be used by the solver if *Use Static Torques* is selected.

Use Stiffnesses: Instructs the solver to automatically calculate the static torques on the slider at each iteration using the specified stiffnesses, static offsets and flying attitude of the slider.

Pitch Stiffness: Suspension stiffness in the pitch direction, this will only be used by the solver if *Use Stiffnesses* is selected.

Roll Stiffness: Suspension stiffness in the roll direction, this will only be used by the solver if *Use Stiffnesses* is selected.

PSA: Pitch Static Attitude. The pitch attitude of the slider when unloaded from the disk. This will only be used by the solver if *Use Stiffnesses* is selected.

RSA: Roll Static Attitude. The roll attitude of the slider when unloaded from the disk. This will only be used by the solver if *Use Stiffnesses* is selected.

3.1.3.1.4. Points of Interest

These values define the points at which steady state fly height will be calculated and written to the output files. Up to one hundred points may be specified in mm or normalized to slider length. The origin of the coordinate system is at the inner leading edge of the slider

3.1.3.1.5. Convergence

These are the two convergence criteria for the solution of the Reynolds equation and the fly height iteration.

Reynolds Equation: Normalized residual for Reynolds equation. The default value of 10^{-7} is usually sufficient, however for higher stiffness sliders, a value of 10^{-8} can help ensure convergence of the load error. Even smaller values may be needed for certain slider designs or requirements.

Load Error: The maximum normalized difference between suspension load and the calculated bearing load, including torque balance. This is used in finding the steady state fly height (and as the convergence criterion for each time step in the dynamic case).

Num. Multigrid

Levels: Quick 4 uses a multigrid method to speed up the solution of the Reynolds equation. For a full discussion of the multigrid method used in Quick 4, please refer to CML Gold Report G9704

The multigrid nesting parameter controls how deeply the multigrid V-Cycle is nested. Briefly, the multigrid method uses a series of 5 grids ranging from coarse to fine. When solving the Reynolds equation on the finest grid (level 1 in Quick 4), the solution is passed up through the successively coarser grids and solved at each level. The solution is then passed back down through the successively finer grids and solved at each level until it reaches the finest grid. This process is called a V-Cycle and is illustrated below in Figure 1. The Num. Multigrid Levels parameter controls the maximum number of coarser levels through which the solution is passed before turning around and being passed back through the finer and finer meshes. The effect of changing the number of multigrid levels is shown in figure 2.

For sliders with steep pressure gradients, Quick 4 sometimes has trouble converging to a solution of the Reynolds equation (i.e. the residual is not decreasing at every iteration). To ensure convergence, Quick 4 can automatically adjust the number of multigrid levels in the simulation. However, for some cases, faster convergence can be achieved by manually reducing the number of multigrid levels.

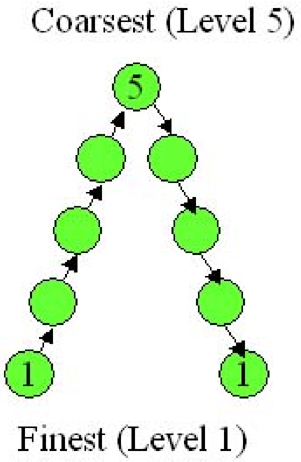


Figure 1

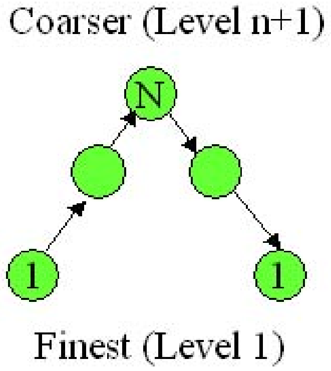


Figure 2

3.1.3.1.6. Comments

Users may enter comments about the current case in this box.

3.1.3.1.7. Geometry Modifications

User Defined Geometry: CMLAir offers air bearing designers the ability to modify the recess height of multiple regions of their slider. This is done by inputting an arbitrary sized matrix describing the deformation (in nanometers) of a region of the slider at evenly spaced points. Values between specified points are quadratically interpolated to give a smooth surface. Multiple user defined geometry regions can be specified. Overlapping regions are allowed as well. The file used to specify the geometry deformation is formatted as follows:

1. The first line contains 4 floating point values that specify the region of the slider (in mm) where the height data will be applied:

```
Lower_X_Value   Upper_X_Value    Lower_Y_Value   Upper_Y_Value
```

For Example, to modify the geometry of a rectangular region stretching from 1.1mm to 1.25mm in the X direction and from 0.45mm to 0.55mm in the Y direction, the first line should be:

```
1.1   1.25   0.45   0.55
```

2. The second line contains the dimensions of the geometry matrix in the X and Y directions respectively.

3. The subsequent lines contain the geometry matrix. The first value in the matrix describes the deformation at the corner closest to the inner leading edge of the slider and the entire first row describes the deformation along the leading edge of the region. Thus, rows describe the deformation of points along the Y direction while columns describe the deformation of points along the X direction.

Example 1: A non-standard crown and camber applied to the entire area of a pico slider
The following user defined geometry file describes a slider's geometry at 3 locations in the X direction and 5 locations in the Y direction. At location (0, 0) the slider has a deformation of -1nm. The center of the slider has a positive deformation of 6nm and the trailing edge has a deformation of 0 0 0 0 -2. The deformation at (1.25, 1.0) is -2nm.

```
0.0  1.25  0  1.0
3 5
-1  0  1  0  0
1  3  6  3   1
0  0  0  0  -2
```

Example 2: a simple method of adding a bump to the rear pad of a pico slider:
The following file describes a small region about 0.0016 square mm near the center of the trailing edge of the pico slider. There is a small spike at the center of this region with an apex at (1.22, 0.5).

```
1.2  1.24  0.48  0.52
3  3
0  0  0
0  5  0
0  0  0
```

Matrix Dimensions displays the X x Y dimensions of the User Geometry matrix. *Area* displays the region of the slider that the current User Defined Geometry region will modify. *Delete* removes the currently selected geometry file.

Regional Height Adjustments:

Regional Height Adjustments allow the user the ability to easily modify the geometry over a region of the slider. The geometry is modified by specifying a planar deformation of the geometry defined by slopes in the X and Y directions and an offset for the center of the plane. Slopes are defined as (Z change in nm / change in mm)

3.1.3.2. Rails Window

This window is used to create and modify rails. Rail creation and modification can be accomplished using several different methods, described below. If changes are made in the Rail Points, Rail Type, or Recess Height areas, the *Add* or *Update* button must be pushed for the changes to take effect. The *Add* button creates a new rail with the specified values. The *Update* button modifies the currently selected rail.

Preview Geometry: Generates a 3-d preview of current rail geometry based on the current rectangular grid parameters. For a higher resolution geometry preview, increase the grid size in the *Grid Window*.

3.1.3.2.1. Rail Points

This text box is used to enter the X and Y coordinates of rails in mm or normalized to length. Each line should contain three numbers: X position, Y position, and wall profile index, separated by spaces. The wall profile index is an integer that defines the index of the wall profile to be used for the edge starting at this point. For example, the third number of the second line in the *Rail Points* box defines the wall for the edge between the second and third points in a rail. The third number of the last line in the *Rail Points* box defines the wall for the edge between the last and first points in the rail. For a vertical wall profile, use 0 as the third number in the line. See the section on *Wall Profiles* for more information.

After the points and wall profile indices have been entered, click on either *Add* or *Update*, located above the rail shape graphic. *Add* will create a new rail. *Update* will modify the current rail. The interface will warn the user if invalid values have been entered.

3.1.3.2.2. Rail Type

The rail type can be either Step or Ramp. Step indicates that the rail has a top surface that is parallel to the disk (assuming zero Pitch, Roll, Crown, Camber, and Twist). Ramp indicates an arbitrary angular orientation of the rail, as defined by the values entered for Recess Height (see Recess Height description). The Update button must be pushed to make the modification after this value is changed.

3.1.3.2.3. Recess Height

This value(s) defines the Z position of the rail in µm. This position is defined relative to a zero recess plane. A zero recess height corresponds to an area that is not etched, such as a rail with *Recess Height* of 0. A recessed region, such as a rail with *Recess Height* greater than 0, corresponds to an etched area. The zero recess plane is normally defined as the area closest to the disk (neglecting Pitch, Roll, Crown, Camber, and Twist), but this is not a requirement (for example, rails can be defined with a negative *Recess Height*).

For a *Step* type rail, only one value is needed for *Recess Height*. This value defines the distance in µm from the top of the rail to the zero recess plane. Thus, a value of 1 defines a rail that is recessed 1µm from the zero recess plane.

For a *Ramp* type rail, three values are needed for *Recess Height*. These values define the recesses of the first three points defined in the *Rail Points* box. Thus, the positions of the first three points explicitly define a plane to which the top surface of the rail is parallel (disregarding Crown, Camber, and Twist). The recess values of the remaining points in the rail are calculated to be coplanar with these three points. You may wish to use the *Preview Geometry* button to ensure you have correctly defined your ramp.

Changing the recess height of a rail or the base recess will bring up the *Update Walls Dialog*. This dialog allows the user to specify how the walls will be modified to match the change in recess height. The dialog has two sets of options. The first set of options specifies the manner in which the walls will be modified, the second set of options specifies which walls will be modified.

Wall Modification Options

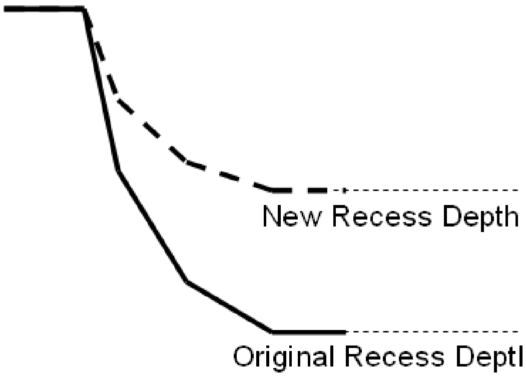
No, do not modify walls: Allows the recess of the rail to change but will not modify the walls in any way.

Yes, interpolate all points on walls: Modifies all the walls that match the wall selection criteria. Walls are modified by interpolating all wall point recess heights either up or down depending on the direction of the recess change.

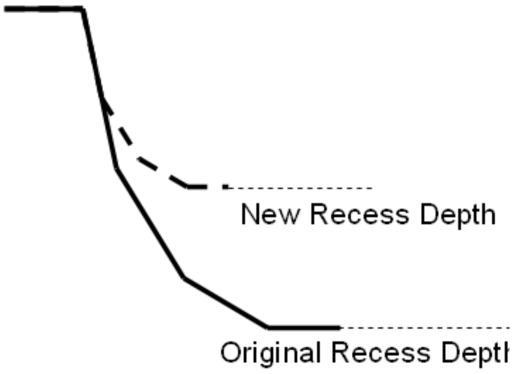
There are two options for handling how the nominal distances of the walls are changed: fixed and proportional.

Fixed: Does not modify the nominal distances of the wall points. Keeping the nominal distance of the wall fixed will result in walls with different wall angles after the wall modification.

Proportional: Modifies the nominal distance of the walls to keep the wall angles the same after the modification.



Fixed wall modification



Proportional wall modification

Yes, but only change the point on the wall at this recess: This option does not change all the points on the wall. For all walls matching the wall selection criteria, only the point at the original recess height will be changed to the new recess. All other points remain unchanged.

Wall Selection Criteria

Walls with a topmost point at the changing recess: By selecting this option, all walls that have a topmost point at the height of the changing recess will be modified.

Walls with a bottommost point at the changing recess: By selecting this option, all walls that have their bottommost point at the height of the changing recess will be modified.

Walls with a central point at the changing recess: By selecting this option, all walls with an internal point (neither bottommost or topmost point) at the height of the changing recess will be modified. When interpolating the wall points, all internal points will be moved while the topmost and bottommost points will remain fixed.

3.1.3.2.4. Base Recess

Base Recess defines the recess depth in µm of areas on which there are no defined rails. In other words, it is the distance between the zero recess plane and the areas in which no rails are defined. This normally corresponds to the areas of maximum etch depth, but this is not a requirement (for instance, rails can be defined with a recess height greater than the *Base Recess* value). The default *Base Recess* value of 3µm means that rails with recess height of 1µm are 2µm closer to the disk than areas with no rails. Rails with recess height of 0µm are 3µm closer to the disk than areas with no rails. Once again, this is neglecting Pitch, Roll, Crown, Camber, and Twist.

Base Recess is defined globally (over all rails), whereas *Recess Height* and *Rail Type* are defined for each individual rail. You do not need to push *Update* to change the *Base Recess* value.

3.1.3.2.5. Rail Index

This displays the index number of the current rail. If no rails are defined, 0 will be displayed. You may scroll through the rails by pushing the arrow buttons next to the index number. You may also choose rails by double left clicking them in the *Rail Diagram*. The total number of defined rails is displayed to the right of the Base Recess input box.

3.1.3.2.6. Add, Delete, Update, and Rail Tools

Add: Creates a new rail with the parameters defined in *Rail Points*, *Rail Type*, and *Recess Height*.

Delete: Removes the current rail.

Update: Modifies the current rail with the parameters defined in *Rail Points*, *Rail Type*, and *Recess Height*.

Symm: Makes rail symmetric about the y-center line by adding a mirror image of the defined points. The new points will be added at the end of the point list with the same wall profiles as their counterparts.

Mirror: Creates a new rail that is a mirror image of the current rail across the y-center line.

Guide: Adds a vertical or horizontal guide to the rails diagram. When the button is pushed, the user is prompted for the location and direction of the guide. When manipulating the rail points with the mouse, the currently selected point the user is manipulating will snap to a guide if it is within a few pixels of the guide. To manipulate a guide that has been placed on the rail diagram, position the mouse over the guide and either click and drag the guide to another location on the slider or right click to bring up a menu of available options for manipulation.

Zoom: Selecting this button puts the rails window in Zoom mode. A left click will zoom in on the rails at the point clicked while a left click will zoom out. Pressing and holding the left mouse button and dragging the mouse over a region of the slider then releasing the mouse button the will cause the rails diagram to zoom in on the selected region.

Circle: Brings up a dialog that allows the user to specify where a circle shaped rail will be placed on the rails diagram. These rails are often used to create landing pads for contact start stop sliders.

Spline: Allows the user to create a quadratic spline shaped rail on the slider. When the *Spline* button is pushed, the user is presented with a dialog that allows them to specify the three control points that define the spline. The first two control points define the end points of the spline; the third point defines the curvature. The user also has the option to draw the spline with the mouse by pressing the draw button present at the bottom of the Draw Spline dialog. If this option is selected, the mouse is used to define the spline's two end points and finally the curvature point.

If a point on a rail is selected when the user originally hits the *Spline* button, that point is inserted as the spline's first control point. If two points are selected, they are inserted as the first and second control points. Finally, if two neighboring points on a rail are selected, when the user hits the *Spline* button, the spline will be inserted into the selected rail instead of creating a new rail.

Draw: Allows the user to draw a rail using the mouse. When the button is pushed, the user is prompted for the wall profile index, type, and recess of the rail to be defined. Then, the user can left click on the rail diagram to define points. When the last point of the desired rail is defined, the user can right click anywhere in rail diagram area to close the rail.

Undo: Allows the user to undo recent rail or wall manipulation operations. The number of successive operations that can be undone is set in Application Preferences dialog (*File->Preferences->Application Preferences*

Redo: Allows the user to undo a previous undo operation.

3.1.3.2.7. Rail Diagram

The *Rail Diagram* is a graphical display of the rails you have defined. You can set a rail as the current rail by double left clicking on it in the diagram. The current rail is highlighted in yellow. If you have defined several rails in one area, double left clicking on that area will cycle through all the rails defined in the area. As noted before, you can also set the current rail using the *Rail Index* arrow buttons. Left clicking on individual rail points will select those points for manipulation. Multiple points can be simultaneously selected by holding the Ctrl key while clicking rail points. It is also possible to select rail points by pressing the left mouse button and dragging the mouse over a region of the slider. The points that fall within the highlighted region will be selected when the mouse button is released. Selected points can be moved by double right clicking on a selected point and dragging the point to the desired location.

The Rails window can be resized to create a larger *Rail Diagram*. Position the mouse over one of the corners of the whole Rails window, click, and drag the corner until the desired size is reached. The *Rail Diagram* will be resized accordingly. You may also maximize or minimize the window. Rails are color coded based on recess height.

3.1.3.2.8. Overlapping Rails

The precedence of rails is set in the opposite order in which they were defined. In other words, when an area contains two overlapping rails, the rail with the highest index number dictates the geometry over that area. So, if you want to define an area with overlapping rails, make sure to define the rails with the lowest *Recess Height* with the highest rail index value.

3.1.3.2.9. Rail Manipulation

The *Rail Manipulation* section of the rails dialog allows the user the ability to easily modify existing rails.

Step Size: This is used to specify the amount of movement when expanding or shrinking a rail, and moving a rail. It is also used to specify the size of the background grid.

Expand Rail: Expands the selected rail by the amount specified in the *Step Size* edit box.

Shrink Rail: Shrinks the selected rail by the amount specified in the *Step Size* edit box.

Up, Down,
Lef,t Right: Moves the currently selected points along the rail a distance specified by the *Step Size* edit box.

Background Grid: When manipulating rail features with the mouse, the *Background Grid* snaps the features to the closest grid point. The size of the background grid is specified in the *Step Size* edit box.

Insert Node: Inserts a node into the rail halfway between the currently selected point and the next point on the same rail. Offsets the node by a small amount so as to avoid colinear points.

Round: Rounds a corner in a rail. When the Round button is pushed, the user is presented with a dialog prompting for the radius of curvature of the rounded corner and the number of points that will be inserted in order to round the corner.

3.1.3.3. Wall Profiles Window

This window is used to define wall profiles for the rails defined in the *Rails Window*. As described in the *Rails Window* section, three numbers define each vertex of each rail: X position, Y position, and wall profile index. The wall profile index is an integer corresponding to one of the wall profiles defined in the *Wall Profiles* Window. A wall profile index of 0 indicates a vertical wall.

Wall profiles are entered in coordinate pairs in the *Profile* box, with each line corresponding to a new point. The X coordinate defines the normal distance from the nominal edge in µm, while the Y coordinate defines the vertical distance in µm from the zero recess plane. The nominal edge is the plane that is perpendicular to the disk (assuming zero Pitch and Roll) and passes through the two points that define the edge. Note that to define a wall that begins at the same vertical position as the top surface of the rail, the first Y coordinate of the wall and the *Recess Height* of the rail should be equal.

The X and Y values of the wall profile are not required to be positive. A negative X coordinate (normal distance) indicates the wall begins inside of the nominal edge, allowing for a smooth transition from rail to wall. A negative Y coordinate (recess depth) indicates a wall that extends beyond (closer to the disk) the zero recess plane.

After the desired coordinate pairs are entered in the *Profile* box, push *Add* to create the new profile. To modify an existing profile, scroll through the profiles using the *Profile* Index arrow buttons, modify the coordinates in the *Profile* box, and push *Update*. The *Delete* button deletes the currently selected profile. Remember, if you delete a profile, all profiles with index numbers higher than the deleted profile will be pushed down by one. Take care to ensure the index numbers of these profiles agree with the values defined in the Rails Window.

The *Undo* and *Redo* buttons in the upper right hand corner of the rails window can be used to correct any mistakes one might make while modifying the wall profiles. The number of operations that can be undone is set in the *Application Preferences* dialog available from the *File* menu.

3.1.3.4. Static Mode - Partial Contact Window

A partial contact model is implemented in the Quick4 solver. The model allows the user to simulate physical contact between the slider and the disk. The Elastic Plastic and Greenwood-Williamson contact models are statistical, and do not model actual impacts of the slider with asperities. Instead, these models add a contact force to the air bearing force based on the statistical amount of contact. The contact force is calculated based on the disk surface parameters and the flying characteristics of the slider. While actual impact effects cannot be predicted, the model provides a useful qualitative analysis of partial contact. In addition to physical contact, intermolecular and electrostatic forces between the slider and the disk can also be simulated.

Model: There are two models available: *Greenwood-Williamson* and *Elastic Plastic*. Benchmarks show that these two models produce similar results. If no contact simulation is desired, choose *No Contact Model*.

Asperity Density: Arial density of asperities per square meter.

STD of Asp.Height: Standard deviation of asperity heights in m.

Radius of Curv.: Mean radius of curvature for asperities in m.

Young's Modulus: Young's Modulus of the disk material in pa.

Poisson's Ratio: Poisson's ratio of the disk material.

Yield Strength: Yield strength of the disk material.

Friction Coeff.: Friction coefficient of the disk surface.

Molecular Forces: To include intermolecular forces in the simulation, the *Use Intermolecular Forces* checkbox must be selected. The Hamacker constants A and B specify constants in the intermolecular forces calculation.

Electrostatic Pot.: Specifies the electrostatic potential between the slider and disk.

3.1.3.5. Grid Window

A high quality computational mesh that captures the geometry of the slider is important for generating accurate results from a simulation. The CML solvers feature adaptive grid methods that adapt to pressure gradients or slider geometry. These adaptive grid techniques allow the solvers to cluster grid lines in areas where the pressure or geometry is rapidly changing without significantly impacting the time required for a simulation. The solvers will also snap gird lines to nearby rail nodes. This allows the solvers to better capture steep clearance changes along rail boundaries.

3.1.3.5.1. Rectangular Computation Grid

Initial Grid: The user may select *Geometric Series* or *Existing Grid*. *Geometric Serie*s indicates that the user will specify an initial grid using the *Geometric Series* input parameters. *Existing Grid* indicates that the interface will use the `x.dat` and `y.dat` files available from a previous simulation. If the grid files are not available, the grid diagram will display a warning for the user and simulation cannot be run.

Grid Type: *Fixed Grid* Indicates the initial grid, as defined by Geometric Series or Existing Grid, will be used throughout the entire simulation. *Adapt Once* indicates the grid will adapt based on the pressure or geometry gradient in the initial case. Note that if multiple cases are run in the same job (multiple RPMs, multiple radii, sensitivity analysis, etc.), the grid will adapt on the first case and stay fixed for the remainder of the cases. *Adapt Each Run* specifies that a new grid will be generated for each case in the job.

Symm. in Width: Toggles the symmetry of the *Geometric Series* input parameters. If this option is on, the user only needs to enter the first half of the *Geometric Series* parameters to specify an initial grid. The remaining half will be mirrored across the y-center line.

3.1.3.5.2. Rectangular Adaptive Grid

Adaptive Mode: Typically it is best to allow the grid to adapt to the pressure gradient. Users can instruct the solver to adapt to the slider geometry gradient along the X direction. In certain instances this can lead to better girds for the simulation, resulting in more accurate results with lower grid densities.

Pressure Gradient: The grid density function used to adjust the grid is based on the pressure gradient obtained from the initial calculation. For a particular X value, a certain weight is given based on the pressure distribution. This weight can be given based on the maximum pressure gradient across all Y values for this X value, or the average pressure gradient across all Y values:

MaxWeight(X) = MAX(Gradient(X,Y)), for all Y
AvgWeight(X) = ΣGradient(X,Y)/n, for all Y

where n is the Y grid number

Max/Min: The pressure gradient in some areas may be very small (e.g. a fully recessed region), but some minimum grid concentration must be maintained for an accurate calculation. *Max/Min* defines the ratio of maximum grid density to minimum grid density. A large number indicates a less uniform grid, while a small number indicates a more uniform grid. A value of 1 would indicate a completely uniform grid.

Decay Factor: The pressure gradient may change abruptly in some regions. Such abrupt changes in the grid distribution should be avoided to reduce discretization error. A smoothing method has been implemented so that the pressure gradient at one point not only affects the grid density at that point, but also has an exponentially decaying influence over neighboring locations. A larger *Decay Factor* causes more abrupt grid density changes.

3.1.3.5.3. Rectangular Grid Geometric Series

The primary method of initial grid generation is by piecewise geometric series.   This feature is rarely used since the adaptive grid can typically generates high quality meshes for complicated slider designs.

Total Grid Size: The total number of mesh points in the X and Y directions. The program uses a multi-grid method to achieve solver efficiency and requires that the grid numbers have the format 16k + 1 or 16k + 2 in Static and Dynamic Modes, respectively.

Control Points: This set of points marks where the slider length and width are cut into segments. For example, two points generate three segments. Within each segment, the successive grid size changes at a fixed ratio (geometric series).

Grid Indices: Each *Grid Index* corresponds to a Control Point. The *Grid Index* specifies the grid number associated with the corresponding Control Point.

Grid Ratios: The *Grid Ratio* specifies the successive change from one grid point to another within a certain segment. The number of *Grid Ratio* entries should be equal to the number of *Control Point* and *Grid Index* entries plus one (the total number of geometric series segments).

Mesh Refinement: Allows the user to specify a region in the x or y direction bounded by *Lower Bound* and *Upper Bound* (in mm), in which the weighting for mesh refinement increased by a factor specified by *Refinement Amount*. *Mesh Refinement* is helpful for accurately capturing high pressure gradients. For more information on mesh refinement, please see CML Technical report #03008

3.1.3.6. Static Mode - Run Setup Window

This window is used to set up the operating conditions for a static simulation.

Radial Pos./Skew: The user must specify at least one *Radius/Skew* pair. For Quick4, if multiple pairs are specified, the simulation will be run at each of these pairs. Positive skew indicates that air flows from the outer leading edge to the inner trailing edge. This conforms to the IDEMA standard.

Sort Runs: Determines the order in which individual cases are run for a simulation with multiple radial positions. By choosing *Absolute Value of Skew*, cases with smaller skews will be run first. When *Adapt Once* is selected as the Grid type (see section 3.1.3.5.1) this choice can result in higher quality meshes for all cases. *Radial Position (Descending)* will run cases at the OD first and then move inwards to the ID.

RPMS: The user must specify at least one *RPM* value. For Quick4, if multiple values are specified, the simulation will be run at each value.

Altitudes: The user may specify one or more *Altitude* values. If no value is given or the altitude is zero, the simulation will use the *Air Parameters* instead. If non-zero *Altitude* values are specified, the *Air Parameters* are ignored.

Press: Specifies ambient pressure.

MFP: Specifies the mean free path.

Viscosity: Specifies the viscosity.

Sensitivities: The user may perform a sensitivity analysis. To examine the sensitivity to a particular parameter, the user should enter a value in the box next to that parameter. The program will run the base case (no increment), a positive increment, and a negative increment using this value. For example, if the user enters "2" in the box next to "Crown", and the base crown, as defined in the General window, is 10nm, the program will run cases at 10nm, 12nm, and 8nm.

The following parameters are available for sensitivity analysis: *Crown, Camber, Twist, Taper Length, Taper Angle, Load, Pitch Torque, Roll Torque*, and *Base Recess*. When *Base Recess* is changed, the wall profiles that end with the same recess height must be adjusted accordingly. There are two methods to adjust the wall profile: *Proportion* and *Fixed Width*. *Proportion* will scale width linearly with recess to keep the same aspect ratio, while *Fixed Width* only allows points to move vertically.

Calculate Stiffness: If *Calculate Stiffness* is checked, the program calculates and outputs a 3x3 stiffness matrix for each basic case across the disk radius. This matrix represents the ratio of change in bearing force components (bearing load, pitch moment, and roll moment) over the change in displacement components (spacing, pitch, and roll). Note that the sign convention is such that the main diagonal elements should normally be positive. The matrix is as follows:

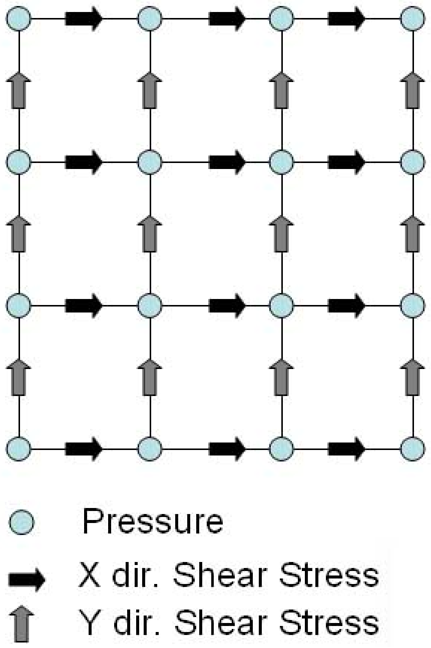
dL / dZ	dL / dP	dL / dR
dPT / dZ	dPT / dP	dPT / dR
dRT / dZ	dRT/ dP	dRT / dR

dZ:	Change in Vertical Displacement (nm)
dP:	Change in Pitch (μrad)
dR:	Change in Roll (μrad)
dL:	Change in Load (g)
dPT:	Change in PTorque (μN-m)
dRT:	Change in RTorque (μN-m)

Save Pressure: Since the pressure data files are rather large, especially when there are many runs with a large grid, they are only saved when *Save Pressure* is checked. If *First traversal only* is selected, only the pressure files for the first traversal of the radial positions will be saved. To save the pressure files for all the runs in a simulation job, select *All runs*. For Quick4, four files will be saved for each radial position: `pressxx.dat`, `cprssxx.dat`, `mflowxx.dat` and `imfxx.dat`.

Save Shear: As of Quick 4.27, the solver can output shear stress files as well. If *Save Shear* is selected, four files detailing the shear stress on the slider will be generated. The files are `CShearXxx.dat`, `CShearYxx.dat`, `PShearXxx.dat` and `PShearYxx.dat`. These four files contain the Couette and Poiseuille components of the shear stress (in atmospheres) acting in the X and Y directions. For most of the output files, the solver outputs the data at each grid point however, for the shear stress calculation, the data calculated and output a bit differently. Shear stress is calculated between the nodes. While this might seem to be a strange break in consistency, this allows the solver to accurately approximate the derivative of pressure in the shear stress calculation. In the picture below it is easy to see the locations of the shear stress output. It is important to note that the sizes of the shear stress output files do not match the computational grid size. The output file dimensions are as follows:

For a grid size of N x M
CShearXxx.dat and PShearXxx.dat: N-1 x M
CShearYxx.dat and PShearYxx.dat: N x M-1



3.1.3.7. MultiModule

The MultiModule adds extensive multiple run capabilities to the CML Air Bearing design program. Its uses are many, including manufacturing tolerance study and optimization. What follows is a general overview of the multimodule options and parameters. For a more complete tutorial, please refer to CML Report #05006

3.1.3.7.1. Definitions

Sample Space and Search Space

Each MultiModule run is separated into sample space and search space. Sample space is the space that is spanned by all cases that are actually sent to the solver and calculated. Search space is the space that is spanned by all cases for which interpolated results are sought. Denser sample spaces result in more accurate search spaces.

The choice of sample space is of utmost importance in determining the efficiency of a MultiModule run. The time required for a run is approximately directly proportional to the number of nodes in sample space.

Solo and Cross

Solo variation of a parameter means that the parameter is varied independently of other parameters. All other parameters are set to the base amount when the solo parameter is varied. This should be used when the effect of this parameter is assumed to be independent of all other parameters.

Cross variation of a parameter means that the parameter is varied through all possible permutations of the other cross parameters. Thus, a full matrix of input variations is generated for cross parameters. This should be used when the effect of this parameter is assumed to be dependent on all other cross parameters.

The number of nodes in sample space varies linearly with the number of defined solo parameters and varies exponentially with the number of defined cross parameters. You may construct sample and search spaces using any combination of solo or cross parameters.

Frequency

Search and sample frequency define the number of possible values that a particular parameter will take.

Global Sample/Search Frequency and Global Sample/Search Type

For parameters that are designated as Global sample types or Global search types, the global frequency and global types are used. Global sample and search types are always uniform.

MC (Monte Carlo)

A MC search type indicates that a particular parameter will receive a randomly generated value for each node in search space. The value is normally distributed with the user-defined standard deviation. For search spaces in which all parameters are MC, the user must set the size of the search space manually by modifying the total search cases value at the bottom right of the MultiModule window. Note that the generated value is only statistically random. A hash of the input parameters determines the seed. This means that successive identical runs will produce identical results.

3.1.3.7.2. Sample and Search Space

Sample Space and Search Space, revisited

It is important to have a stong understanding of the relationship between sample and search space in order to maximize the efficiency of the MultiModule. As described in the previous section, sample space defines the nodes for which results are generated using an air bearing solver (quick4 or quick5). Search space results are generated through linear interpolation of sampled results. As you might expect, a good sample set is one which properly characterizes the result space. For a result space that is highly linear, a sparse sample distribution is sufficient. For non-linear result spaces, a dense sample distribution may be necessary.

The MultiModule offers two basic sample types: Solo and Cross.

Solo indicates that individual input parameters are varied independently. The sample space size varies linearly with the number of parameters and the sample frequency of each parameter. For instance, if there are two solo parameters with a sample frequencies of 4 and 5, the sample space will contain 9 nodes (4 + 5).

Cross indicates that input parameters will be varied simultaneously in every possible permutation. The sample space size varies exponentially with the number of parameters and the sample frequency of each parameter. For instance, if there are two cross parameters with sample frequencies of 4 and 5, the sample space will contain 20 nodes (4 x 5).

Proper selection of frequency and sample type will optimize the efficiency of the MultiModule. In some cases, solo samples with a sample frequency of 2 are sufficient. This will minimize the amount of solver time spent generating results. For parameters such as crown, camber, and twist, sensitivity independence can normally be assumed. For parameters such as rail and vertex locations, parameter sensitivies are almost always dependent. With experience, users should be able to decide the appropriate sample type and frequency for each case.

Interpolation

Search space results are generated through linear interpolation of sampled results. The case in which sampled results are generated by varying input parameters independently (solo) is trivial. Search results can be easily generated by assuming that the sensitivity of one parameter does not affect the sensitivity of another parameter.

Example: To study the effect of crown and camber on a particular design, crown and camber are each varied independently (solo sample type). The sampled results show that pitch sensitivity to crown is 5µrad/nm and pitch sensitivity to camber is 3µrad/nm. The pitch effect if both crown and camber are varied simultaneously can be calculated by summing the individual effects. So, for a design where crown is increased by 5nm and camber is decreased by 2nm, the pitch effect can be calculated as follows:

Pitch Effect = (5µrad/nm * 5nm) + (3µrad/nm * -2nm)

= 19µrad

Resulting Pitch = Base Pitch + 19µrad

Of course, for many cases, independence of parameter sensitivities cannot be assumed. For these cases, the sample space must be generated by varying each relevant parameter simultaneously with other parameters (cross sample type). The result is a much denser sample space which more accurately characterizes the result space when interpolated.

Sample space can be seen as an n-dimensional mesh, where n is the number of relevant parameters varied. For instance, if crown, camber, and twist are varied, the sample space will be a 3-dimensional mesh. The n-dimensional sample space can be viewed as a set of adjacent, non-overlapping n-dimensional cubes, each having 2-to-the-n corners. Each corner represents a node in sample space. For each node in search space that lies within one of these n-dimensional cubes, results can be interpolated from the sampled results. For search nodes which lie outside all cubes, results can be extrapolated from the nearest cube.

Note that if the base case is not included explicitly in the sample space, the results will be interpolated in the search space. Although the base case is always calculated through the air bearing solver, it will not necessarily be a node in the sample space. If the base case is not a sample node, you may notice differences in the base sample results and the base search results.

3.1.3.7.3. MultiModule Window

Parameter Type: The input parameter that will be varied.

Range: The upper and lower search space boundaries for this parameter. The users enters the amount of variation relative to the base. Absolute values are displayed to the right of the value entered by the user. These input boxes are ignored for parameters for which the Relative box is checked in the Rail/Point Motion section of this dialog box.

Sample Type: The sample space parameters. Users may select from Global, Cross, and Solo. Global indicates that the Global Freq/Type settings will be used. For a discussion of Cross and Solo sampling, see the manual sections on MultiModule Definitions and Sample and Search Space.

Sample Frequency: Specifies the number of possible sample values the parameter will assume.

Sample Sub-Type: The distribution of sample nodes. Uniform indicates k sample values will be distributed uniformly over the interval defined in the Range section, where k is the Sample Frequency. Median and Limit indicate 2 or 3 parameter values will be distributed over the Range. If Unidirectional is checked, the base and one other value (up or down) will be sampled. If Unidirectional is not checked, the base and two other values will be sampled (up and down).

Search Type: The search space parameters. Users may select from Global, Cross, Solo, and MC. MC indicates that the search value will be determined randomly for each node in the search space. The random value is chosen from a nomal distribution with the user specified standard deviation.

Search Frequency: The number of possible search values the parameter will assume.

Global Freq/Type: These values are used if either the Sample Type or Search Type are set to Global.

Rail/Point Motion: If the parameter is of type Rail or Rail Point, the rail and/or point numbers must be specified. As the user enters rail and point numbers, the locations are plotted in the chart to the right. The rail to be moved is outlined in yellow. The point to be moved is outlined with a circle. If the Relative box is checked, the parameter will follow the motion of the relative point. If the Relative and Opposite boxes are checked, the parameter's motion will be exactly opposite that of the relative point.

Note that there can be multiple levels of relativity and motion. For instance, Rail A can be relative to Point B which can be relative to Point C. Also, if a rail and a point within the rail move, the rail will always move first, then the point within the rail.

You may select rails and points with the mouse in same manner as your select rails and point in the Rails Window. To select a rail, double click on the desired rail. To select a point within the selected rail, right click on the point. Note that only the actual rail and point index values for the parameter are affected. The relative rail and point index values must be set manually by editing the appropriate text boxes.

Active Parameters: Displays a list of the active parameters. Clicking on a parameter name will display the parameter values in the rest of the dialog.

Add: To add a new parameter, select the desired settings in the dialog box and press the Add button. Note, you must press this button in order for changes to take effect.

Update: To update the selected parameter, make changes to the settings in the dialog box and press the Update button. You must press this button in order for changes to take effect.

Delete: To delete the selected parameter, press the Delete button.

New: To start creating a new blank parameter, press the New button.

Total Cases: Displays the total size of the sample and search spaces. If all search parameters are of type MC, the user will need to manually enter the size of the search space. Otherwise, the sample and search space sizes are automatically calculated.

Yield: Displays the Yield Limit dialog. In this dialog, the user may enter absolute limits for the result values. If a result value of a particular case falls outside these limits, it will be considered a yield failure. To specify limits for a particular result value, select the check box for that value and enter limit values.

Results: Displays a list of all search and sample nodes, including generated or intepolated results, if any. This window will also allow users to copy results to the clipboard for pasting into spreadsheets such as Excel, export cases to ASCII files, or view histograms of the results. Yield percentage, if any, is displayed in the upper right of the window. The histogram window dispays histograms of search space results. Neither the histograms nor the yield percentage include case 0 (base case).

Run: Clears all previous results and calculates new results for each node in sample space. The Solver Status window will be displayed showing the status of each cases being run. The base case is always run first, even if it is not explicitly included in the sample space. If subsequent sample nodes are equivalent to the base case, the base case results are used.

3.1.3.8. Static Track Seek Simulation

The static track seeking dialog allows the user to perform quasi-static track seek simulations. By taking into account airflow (effective skew/RPM) and inertial effects of track seeking, a quasi-static track seeking simulation can be an effective alternative to performing a full dynamic simulation. To perform a static track seeking simulation, the user must enter the inner and outer radial positions for the track seek as well as acceleration, deceleration and maximum velocity for the seek. These parameters must be specified such that the slider can reach the maximum seek velocity during the seek. Most parameters from the track seek dialog box should be self explanatory. The arm length is the length (mm) from the pivot to the geometric center of the slider. The pivot-motor distance is the distance (mm) from the center of the actuator pivot to the center of the spindle motor. More info on quasi-static track seeking simulations can be found in CML Blue Report 05021.

3.1.3.9. Python Interpreter

A python interpreter has been embedded in the CMLAir user interface. This allows the user to write python scripts that automate the study of air bearing sliders and extend the interface to work in different environments. In the most basic of situations, the python interpreter can be used as a macro language to automate tedious or repetitive tasks. In more ambitious situations, the interpreter can be used to submit jobs to remote machines or tie the interface in with other software. In its current incarnation, the user is not required to have python installed on the host machine, all the major functionality is contained within the interface and a python library. However to get the most out of python's extensive functionality, a working python installation is still encouraged for the user and can be freely downloaded from www.python.org. What follows is a brief introduction to the CMLAir python API and a description of the functions you can call from within a CMLAir python script. Only a passing familiarity is necessary to use the python API. Users unfamiliar with the language should spend an hour or two familiarizing themselves with the basics of the language by using the tutorials available at www.python.org.

3.1.3.9.1. Python Interpreter Example

Probably the best way to get introduced to the general flow and feel of using python to control CMLAir is through a quick example. The script listed below is breif and simple but does a good job of highlighting the necessary points of automating the interface with python. To run this script in CMLAir, simply copy it into a text editor (ie. notepad) and save it as a `gridConvergence.py`. After creating and saving the python script, select "*Run Script...*" from the Preprocessor window in CMLAir. In the dialog box that pops up, select the python script and click "*Open*". This will run the python script from within the CMLAir interface.

```
import cml, results

def main():
    baseCase = "C:\\Quick4_Example.cml"
    outFile = "C:\\output.txt"
    f = open(outFile, "w")

    #open our .cml case
    if cml.OpenCase(baseCase) == 0:
        return

    #set initial mesh size
    meshSize = 177
    cml.SetGridSizeX(meshSize)
    cml.SetGridSizeY(meshSize)
    cml.SaveCaseNoWarn()
    while meshSize < 400:
        #run it
        cml.RunCaseNoWarn()
        #get results
        res = results.Results()
        res.GetResults()
        if res.HasResults() == 0:
            cml.MsgBox("error: No results, aborting")
            return
        #pull out minimum fly height
```

```
            singleResult = res[0]
            fh = singleResult.minFH
            #output mesh size and minimum fly height
            dataStr = str(meshSize) + ' ' + str(fh)
            f.write(dataStr)
            #increment mesh size
            meshSize += 16
            cml.SetGridSizeX(meshSize)
            cml.SetGridSizeY(meshSize)
            cml.SaveCaseNoWarn()

    main()
```

In case the comments in the above script aren't detailed enough, the sole purpose of the script is to study the effect of mesh density on minimum flyheight. At each step in the loop, the mesh size is increased, the case is run and the resulting minimum flyheight is recorded. In order to document the significant aspects of a python script, the important lines in the script will be detailed below.

```
import cml, results
```

In order to communicate with the interface and access results, the script must import the `cml` and `results` modules. Unlike many non-standard modules, these modules do not exist as separate files in the script directory or python search path, instead the modules are compiled into the CMLAir executable.

```
if cml.OpenCase(baseCase) == 0:
    return
```

This is the first call to a cml module function. Notice the call to `OpenCase()` is proceeded by `cml`. The python interpreter sees CMLAir as an object with associated member functions. Thus, preceding all calls to the interface by the module name `cml` (and later `results`) will be the standard convention.

```
cml.SetGridSizeX(meshSize)
cml.SetGridSizeY(meshSize)
```

The script starts by setting the initial mesh size for the simulations. If the values passed to the `SetGridSize` functions are not the correct multiples of 16, the functions will correct the input values and round the value to the nearest multiple of 16 + 1.

```
cml.SaveCaseNoWarn()
```

It's always a good idea to save the case before running it. If results exist for the current case, a call to `SaveCase()` will pause the script and bring up a dialog box warning the user that they are about to overwrite the existing results. The `SaveCaseNoWarn()` function prevents the results warning from being issued. Typically this is the best option for saving cases when running batch jobs.

```
cml.RunCaseNoWarn()
```

This simply runs the CML case. Just like the `SaveCaseNoWarn()` function used above, in order to not to have to worry about dismissing warning message boxes about overwriting results we use the `NoWarn` variant of `RunCase()`. The script will wait until the case is finished running before continuing execution.

```
res = results.Results()
res.GetResults()
if res.HasResults() == 0:
    cml.MsgBox("error: No results, aborting")
    return
```

At this point the case has finished running and has results. In order to pull the results out of the interface, the user must create a `results` object which will hold the simulation results. A call to the results constructor `results.Results()` does the trick. Even though a new results object has been created, it the current results in the interface must be imported into the results object by making a call to `res.GetResults()`. Before proceeding in the script, a call to `res.HasResults()` is made to ensure the results object was actually able to extract results from the interface. If this fails (possibly due to a solver crash or other unknown error), appropriate action is taken (in this case, a warning message is displayed before stopping execution of the script).

```
singleResult = res[0]
```

A single CMLAir simulation run can generate results for multiple radial positions, altitudes or sensitivities. Thus, the result object behaves as a list with each index containing the result of a single run in the simulation. Even with only one result, `res` still behaves as a list of results. We use the sequence subscript operator to access individual results in the list. Doing so returns a singleResult object.

```
min = singleResult.minFH
```

The `singleResult` structure has a number of parameters that can be queried in order to access the relevant result info. In this case, the user gets the minimum flying height of the slider.

While this was a simple script, as stated earlier it is possible to leverage python's powerful libraries, visualization software and scientific packages to write complex scripts for air bearing analysis. The full description of the CMLAir Python API functions and modules is included in the help file distributed with CMLAir

3.1.4 Static Mode – Postprocessor Menu

The static post-processor supports visualization of all output data from the Quick4 and Quick5 solvers. Right click on the post-processing window to display the options available for the window.

Result File: Displays the `result.dat` file.

Result Table: Summarizes the results from `result.dat` in a table. Users can select which results will displayed by clicking on *Select Columns…* For jobs with more than one simulation run (i.e. multiple radii, rpms or altitudes), it is possible to display a plot of input values vs results. Simply click on *Plot Results...* In the dialog that appears, select the input values that will be displayed on the X axis and the results that will be displayed on the Y Axis. Finally use the mouse and the Ctrl and Shift keys to select the simulation runs that will be plotted.

Copy All Results: Copies the results to the Windows clipboard in a tab delimited format. This may be easily pasted into a spreadsheet application such as MS Excel.

Output File: Displays the solver output.

Rails: Displays a 2-d diagram of the rails.

3-d Rail Geometry: Displays a plot of the 3-d rail geometry, given by `geom.dat`, over the final grid, given by `x.dat` and `y.dat`. This plot is available only if the *Save Pressure* or *Geometry Only* options are enabled. To view the plot from a different angle, double left click and drag the diagram. Right click to display a menu of options.

Pressure Analysis: Allows the user to calculate the pressure and center of force over a specific region of the slider. Region boundaries can be input by hand or by using the mouse. To select a rail as the region boundary, double click on the desired rail. When analyzing a case with multiple runs, select the desired run from the result drop down list at the top-left of the dialog.

Massflow Analysis: The *Massflow Analysis* dialog allows the user to plot the strealines at a particular spacing between the slider and the disk. By clicking on the "*Options…*" button, the user can select how the streamlines are plotted as well as change how the streamline depth is calculated. For percentage of spacing, the streamline is traced along a path that is a given percentage of the spacing between the slider and the disk. For the recess depth, the streamlines are traced out along a plane parallel to the plane of the slider.

Spacing: Allows the user to find the spacing between the disk and specific points on the slider. To use this feature, simply input the x and y coordinates of points on the slider in the edit box on the left hand side of the window, or use the mouse to click the desired points in the picture. Once all desired points are selected, click *Calculate Height*. The slider-disk spacing at each point is shown in the edit box. When analyzing a case with multiple runs, select the desired run from the Result drop down list at the top-left of the dialog.

Lube Depletion Dialog: The lube depletion dialog allows the user to numerically determine the lube depletion due to air bearing pressure. It is necessary for the user to have experimental data on the surface energy of the lubricant. The A coefficients in the dialog come from an exponential curve fit of the disjoining surface energy of the lubricant. The B coefficients come from a polynomial curve fit of the polar surface energy of the lubricant. For more information, please see CML Blue Report 06007 “Slider-Lubricant Interactions at the Head-Disk Interface.”

Grid: Displays a diagram of the final grid, given by `x.dat` and `y.dat` or `xyph.dat`. Right click to display a menu of options.

Air Pressure: Displays the final pressure for a particular case, given by `pressxx.dat` or `xyphxx..dat`, over the final grid, given by `x.dat` and `y.dat`. This plot is available only if the *Save Pressure* option is enabled. The scale displayed is normalized to ambient pressure: (p - pa)/pa. Double click and drag to rotate the diagram. Right click to display a menu of options.

Contact Pressure: Displays the final contact pressure for a particular case, given by `cprssxx.dat`, over the final grid, given by `x.dat` and `y.dat`. This plot is available only if the *Save Pressure* option is enabled. The scale displayed is in kg / (m*m). Double click and drag to rotate the diagram. Right click to display a menu of options.

Mass Flow: Displays the final averaged mass flow for a particular case, given by `mflowxx.dat`. `xm.dat` and `ym.dat` define the grid which is shifted by a half grid cell from `x.dat` and `y.dat`. This plot is available only if the *Save Pressure* option is enabled. Right click to display a menu of options.

Intermolecular Forces: If *Use Intermolecular Forces* is selected in the *Partial Contact* window, this will display the intermolecular forces over the final grid given by `x.dat` and `y.dat`

3.1.5. Windows Menu

This menu contains a list of open windows. To close all open windows, choose Close All Windows. This will not close any run windows or halt any simulations.

3.1.6. Help Menu

Help: Displays the Windows Help file corresponding to this manual.

About: Displays a dialog box with information about this program.

3.2. Dynamic Mode

Dynamic mode is used for the design and dynamic simulation of air bearing sliders. The Dynamic Mode menu bar selections are as follows: *File*, *Options*, *Preprocessor*, *Postprocessor*, *Windows*, and *Help*. The Dynamic Mode toolbar selections are as follows: *General*, *Rails*, *Walls*, *Partial Contact*, *Grid*, *Run Setup*, *Disk Topography*, *Static Mode*, *Open*, *Save*, *Solver*, and *Run*. When in Static Mode, Dynamic Mode is enabled by pressing the "*Static/Dyn*" button or choosing *Dynamic Mode* from the *Options Menu*.

3.2.1. File Menu

This menu is identical to the *File Menu* in Static Mode (see Static Mode File Menu).

3.2.2. Options Menu

This menu is identical to the *Options Menu* in Static Mode (see Static Mode Options Menu) except for the absence of the *Solution* option.

3.2.3. Dynamic Mode - Preprocessor Menu

The preprocessor menu has the following selections: *General*, *Rails*, *Wall Profiles*, *Partial Contact*, *Grid*, *Run Setup*, and *Disk Topography*. All of these selections are also available from the toolbar.

3.2.3.1. Dynamic Mode - General Window

The General Dialog box allows for the input of the following categories of input parameters: *Slider Geometry*, *Initial Flying Condition*, *Points of Interest*, *Convergence*, and *Time*.

3.2.3.1.1. Slider Geometry

Length: See Static Mode Slider Geometry.

Width: See Static Mode Slider Geometry.

Taper Length: See Static Mode Slider Geometry.

Taper Angle: See Static Mode Slider Geometry.

Crown: See Static Mode Slider Geometry.

Camber: See Static Mode Slider Geometry.

Twist: See Static Mode Slider Geometry.

3.2.3.1.2 Slider Mass

Mass: Mass of the slider in grams.

X Grav.: Slider gravity X-position in mm, starting from leading edge (see Suspension Loading).

Y Grav.: Slider gravity Y-position in mm, starting from geometric center (see Suspension Loading).

Z Grav.: Distance in mm from the top of the slider to slider's gravity center in Z direction (see Suspension Loading).

Pitch MOI: Slider pitch moment of inertia in kg-m^2.

Roll MOI: Slider roll moment of intertia in kg-m^2.

Yaw MOI: Slider yaw moment of intertia (MOI about z axis) in kg-m^2.

3.2.3.1.3. Initial Flying Condition

These values define the initial flying conditions for the slider (t=0).

Altitude: The altitude of the simulation in meters. The altitude is fixed throughout the simulation.

TEC: Initial trailing edge center fly height. See Static Mode Initial Flying Attitude.

Pitch: Initial pitch. See Static Mode Initial Flying Attitude.

Roll: Initial roll. See Static Mode Initial Flying Attitude.

Initial X Pos: Initial displacement of the slider in the X direction

Initial Y Pos: Initial displacement of the slider in the Y direction

Initial Yaw Pos: Initial yaw of the slider

Vert. Vel.: Initial vertical impulse velocity at the slider gravity center in m/s.

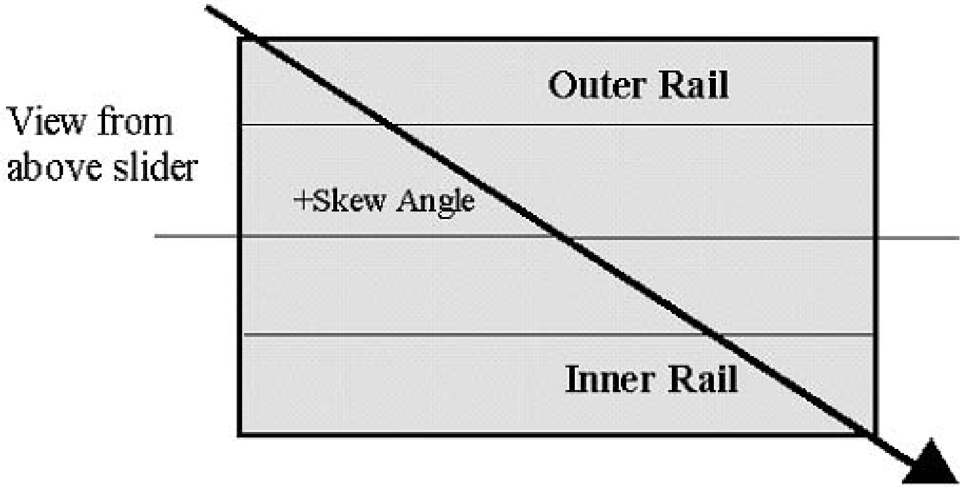
Pitch Vel.: Initial pitch angular velocity in µrad/s.

Roll Vel.: Initial roll angular velocity in µrad/s.

RPM: Initial disk speed. Disk speed can be varied using the Disk Velocity Profile in the Run Setup Window.

Radial Pos.: Initial radial position of the slider's gravity center. Radial position can be varied using Track Seeking Motion in the Run Setup Window.

Skew: Initial skew angle in degrees. Positive skew indicates that air flows from the outer leading edge to the inner trailing edge. This has been changed from previous versions of the dynamic simulator to conform to the IDEMA standard.



3.2.3.1.4. Points of Interest

These are the four points of interest at which various spacings are calculated. They are defined similarly to the *Static Mode Points of Interest* (see Static Mode POI).

3.2.3.1.5. Convergence

These values are the same as those in Static Mode (see *Static Mode Convergence*).

3.2.3.1.6. Time

These values determine the manner in which time is handled in the dynamic simulation.

Step: The time in ms allotted for each base step of the simulation. A smaller value will increase the computation time and accuracy of a simulation. For simulations using a finite element suspension it is suggested that a time step of 10^{-7} seconds or less is used.

Total: The total duration of the simulation in ms.

3.2.3.2. Dynamic Mode - Rails Window

This is identical to the *Rails Window* in Static Mode (see *Static Mode Rails Window*).

3.2.3.3. Dynamic Mode - Wall Profiles Window

This is identical to the *Wall Profiles Window* in Static Mode (see *Static Mode Wall Profiles Window*).

3.2.3.4. Dynamic Mode - Partial Contact Window

This window is similar to its Static Mode counterpart (see *Static Mode Partial Contact Window*), with a few small changes. In Dynamic Mode, the user may enter several different contact zones, each with different surface characteristics. The parameters are summarized here, including those already described in the *Static Mode Partial Contact Window*.

Model: There are two models available: *Greenwood-Williamson* and *Elastic Plastic*. Benchmarks show that these two models produce similar results.

Young's Modulus: Young's Elastic Modulus of the disk material in Pa.

Poisson's Ratio: Poisson's ratio of the disk material.

Yield Strength: Yield strength of the disk material.

Friction Coeff.: Friction coefficient of the disk surface.

Asperity Density: Arial desity of asperities per square meter.

STD of Asp.Height: Composite standard deviation of asperity heights in m.

Radius of Curv.: Mean radius of curvature for asperities in m.

Glide Height: Height in nm above which there is no contact. This parameter is only available in Dynamic Mode.

Start Radius: Radial position in mm at which the contact zone begins.

End Radius: Radial position in mm at which the contact zone ends.

To add a contact zone, first choose a model and enter the material parameters (*Young's Modulus*, *Poisson's Ratio*, *Yield Strength*, and *Friction Coefficient*). These parameters are will not vary between different contact zones. Now, enter values for the surface parameters (*Asperity Density*, *STD of Asp.Height*, *Radius of Curvature*, *Glide Height*, *Start Radius*, and *End Radius*) and push the *Add* button. Up to five contact zones with different surface parameters may be added. To delete a contact zone, scroll to the desired zone using the arrow buttons and push the *Delete* button. To modify a contact zone, scroll to the desired zone, make the necessary modification, and push the *Update* button.

3.2.3.5. Dynamic Mode - Grid Window

This is almost identical to the *Grid Window* in Static Mode (see *Static Mode Grid Window*). The only difference is in the *Total Grid Size*, which must be of the form 16k + 2, where k is a positive integer (in Static Mode, *Total Grid Size* must be of the form 16k + 1). Certain grid options available in Static Mode are not available in Dynamic Mode. Specifically the ability to adapt to geometry gradient and refinement regions are not implemented in the dynamic simulator.

3.2.3.6. Dynamic Mode - Run Setup Window

This window is used to set up the operating conditions for a dynamic simulation. The Dynamic Mode Run Setup Dialog box allows for the input of the following categories of input parameters: *Suspension Dynamics*, *Actuator*, *Suspension Loading*, *Track Seeking Motion*, *Disk Velocity Profile*,*Disk Flutter* and *PIP*.

The user has the option of integrating suspension dynamics using a reduced order finite element suspension or using stiffness and damping coefficients. If the former is chosen, the user must supply additional files (see below).

Finite Element Suspension: The current CML dynamic simulator is different from its predecessor in a few ways. The most obvious and far reaching change is the addition of an accurate reduced order finite element suspension. Instead of using an ABAQUS suspension or simple stiffness and damping parameters to model the suspension, the user must export, from ANSYS, a finite element suspension model and use CML's model order reduction code to create the proper finite element input files for the dynamic simulator.

While the entire process of specifying a suspension is more involved than in previous versions of CMLAir, the new suspension model allows for fast and accurate dynamic simulations.

The CML reduction code and instructions on obtaining accurate suspension input files can be downloaded from CML website.

By default, the simulator will use a simple 6x6 suspension matrix and will autogenerate the mass and nodal coordinate files. The autogenerated mass file is a diagonal matrix with the first three values equal to the mass of the slider. The following three values are the roll, pitch and yaw moments of inertia respectively. The autogenerated degree of freedom file consists of one node located at the slider dimple with six degrees of freedom. The coordinates are calculated using the arm length specified in the *Actuator/Track Seeking* dialog.

To use custom suspension, mass and coordinate files, use Ansys and the CML Reduction code to generate the reduced order suspension files then import each of the files output from the reduction code into CMLAir. It is also possible to edit the suspension and mass matrices after they have been imported into CMLAir. Simply click the Edit button and make the desired changes to the matrices in the dialog that pops up.

Damping Parameters: The system damping matrix is computed as a linear combination of the stiffness and mass matrices. [C] = a1*[M] + a2[k]. To easily compute a1 and a2, the user specifies the percentage of damping desired at two given frequencies. The coeficients are then calculated by solving the following linear system for a1 and a2.

*percent_a = a1/(2*frequency_a) + a2*frequency_a/2*
*percent_b = a1/(2*frequency_b) + a2*frequency_b/2*

As a simple example, for 5% damping at 10Hz and 50kHz we would have
a1 = 0.9998
a2 = 2.0e-6

Stiffness and Damping Coefs.: The user may opt to integrate suspension dynamics using stiffness and damping coefficients as opposed to using a finite element suspension. No additional files are needed in this case. The user must supply additional parameters (*Vertical Stiffness*, *Pitch Stiffness*, *Roll Stiffness*, *Vertical Damping*, *Pitch Damping*, and *Roll Damping*) as described below.

Vetical Stiffness: Vertical suspension stiffness coefficient.

Pitch Stiffness: Pitch suspension stiffness coefficient.

Roll Stiffness: Roll suspension stiffness coefficient.

Vertical Damping: Vertical suspension damping coefficient.

Pitch Damping: Pitch suspension damping coefficient.

Roll Damping: Roll suspension damping coefficient.

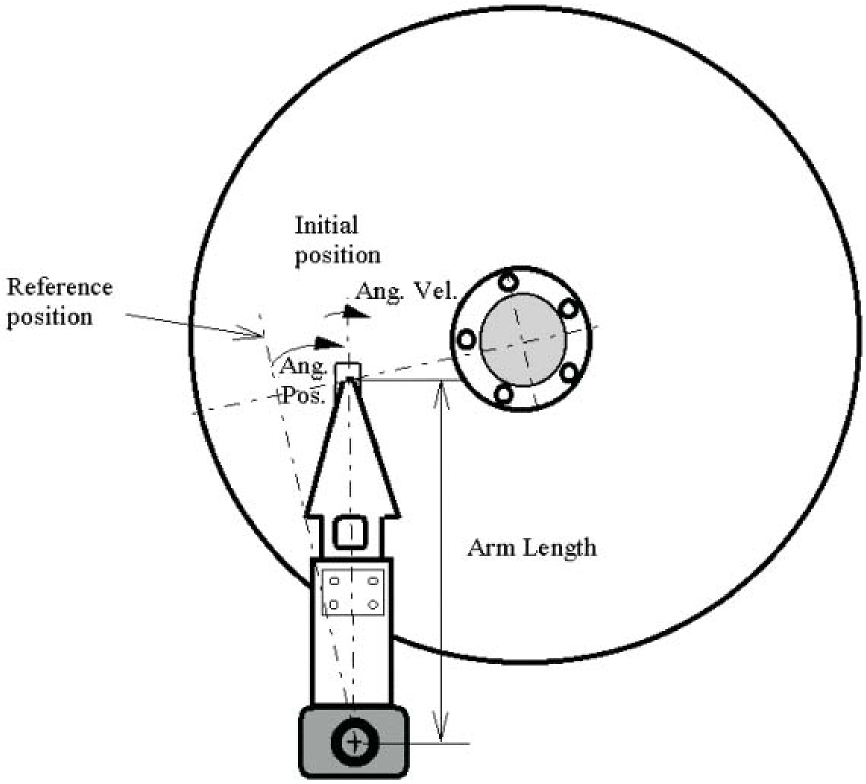
3.2.3.6.2. Actuator

No/Inline Actuator: Toggles between no actuator and inline actuator.

Arm Length: Length of the actuator arm in mm.

Angular Position: Angular position in radians of the actuator with respect to the reference position.

Angular Velocity: Angular velocity of actuator in rad/s. Positive velocity indicates that the slider is moving from OD to ID.



Min. Inner Rad.: The minimum flyable radial position of the slider's gravity center.

Max. Outer Rad.: The maximum flyable radial position of the slider's gravity center.

3.2.3.6.3. Suspension Loading

Load: Suspension load in grams.

Pitch Offset: Load x-position in µm, starting from geometric center towards trailing edge.

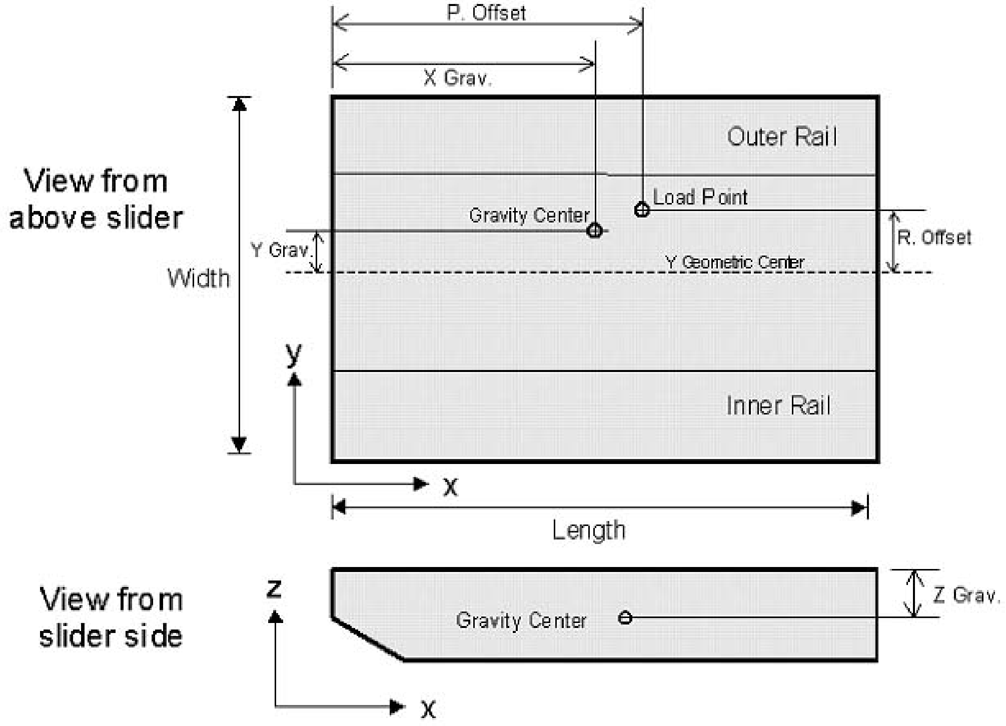
Roll Offset: Load y-position in µm, starting from geometric center towards outer edge.

PSA: Pitch Static Attitude. The pitch attitude of the slider when unloaded from the disk.

RSA: Roll Static Attitude. The roll attitude of the slider when unloaded from the disk.

Pitch Stiffness: Pitch suspension stiffness coefficient. It is necessary to specify this parameter if using a finite element suspension and non-zero PSA

Roll Stiffness: Roll suspension stiffness coefficient. It is necessary to specify this parameter if using a finite element suspension and non-zero RSA



3.2.3.6.4. Track Seeking Motion

The user may control the radial position of the slider by entering a track seeking acceleration profile. The format of the *Track Seeking Motion* edit box is two numbers per line, separated by spaces or tabs, with the first representing time in seconds and the second representing acceleration in rad/sec^2. A linear functional relationship is assumed to exist between adjacent data points. Up to 100,000 lines may be entered.

3.2.3.6.5. Disk Velocity Profile

The user may control the speed of the disk by entering a disk velocity profile. The format of the Disk Velocity Profile edit box is two numbers per line, separated by spaces or tabs, with the first representing time and the second disk RPMs. Up to 40 lines may be entered.

3.2.3.6.6. Disk Flutter

Enable: When checked, disk flutter is enabled. Otherwise, it is disabled.

Start Time: Time in seconds at which disk flutter begins.

End Time: Time in seconds at which disk flutter ends.

Frequency: Frequency in Hz of disk flutter.

Amplitude: Amplitude in nm of disk flutter.

3.2.3.6.7. Shock

The shock window allows the user to specify parameters needed in order to run a simple shock simulation. Shock is simulated by applying a normal force (N), a pitch moment (Nm) and a roll moment (Nm) to the slider at the load point. The responses of the disk and other components are neglected. The user can specify a simple half sinusoidal shock or a custom user defined shock. The signs of the forces acting on the slider are consistent with the conventions established by other parameters in the interface. Briefly, a positive normal force acts to decrease the flyheight, a positive pitch moment will cause the slider's pitch to increase and a positive roll will increase the slider's roll in the positive direction.

User Defined Shock: The correct user defined shock file should contain 4 columns and N rows (N should be large enough to accurately describe the shock applied to the slider during the simulation). The columns of the file, from left to right are: Time (seconds), Normal force (Newtons), Pitch Torque (Newton meters), and Roll Torque (Newton meters)

A sample user defined shock file should look like:

```
0.000       0         0          0.000
1.00E-07    0         0          0.000
2.00E-07    0         0          0.000
3.00E-07    0         0          0.000
4.00E-07    0         0          0.000
5.00E-07    0         0          0.000
6.00E-07    6.18E-03  -3.09E-06  0.000
7.00E-07    1.18E-02  -5.88E-06  0.000
8.00E-07    1.62E-02  -8.09E-06  0.000
9.00E-07    1.90E-02  -9.51E-06  0.000
1.00E-06    2.00E-02  -1.00E-05  0.000
1.10E-06    1.90E-02  -9.51E-06  0.000
1.20E-06    1.62E-02  -8.09E-06  0.000
1.30E-06    1.18E-02  -5.88E-06  0.000
1.40E-06    6.18E-03  -3.09E-06  0.000
1.50E-06    0         0          0.000
1.60E-06    0         0          0.000
1.70E-06    0         0          0.000
1.80E-06    0         0          0.000
```

The (oversimplified) shock file shown above simulates a half sinusoidal shock from 0.5 microsecond to 1.5 microseconds. The maximum force applied in the normal direction is 0.02 N, the maximum moment applied in the pitch direction is -0.00001 NM and no moment is applied in the roll direction. The user should be aware of two key points when creating a user defined shock file: (1) the file does not contain a header and (2) in this release, the shock file length is limited to 10,000 lines.

3.2.3.6.8. Parameter Identification Program (PIP)

The PIP window allows users to generate the proper input files for using the CML Parameter Identification Program.

The steps for running a PIP simulation and using the PIP software are as follows:

1. Open the desired .cml dynamic case in CMLAir
2. Set the dynamic properties:
   - Initial guess for fly height, pitch and roll and disk rpm (General Window)
   - Grid size
   - Important: For good results, the convergence parameters in the general window should be set for high accuracy.

- Error for Reynolds Equation solver < 10-9
- Load error < 10-4

3. In CMLAir, select the Dynamic mode, open the Setup window and select the PIP tab. Enter impulse velocities in the z (height), p(pitch) and r(roll) directions and the time for each simulation. The default values are given and form a good guideline for choosing these values but, depending on the slider design, your mileage may vary.
4. Click Run PIP Simulations
5. When the 4 simulations have finished running, make sure that the correct path to pip.exe is entered in the PIP analysis section of the dialog and click Launch PIP Software. This will export the simulation data to a folder in the same directory as pip.exe. It is important that all PIP simulation data be within subfolders of pip.exe
6. From within pip.exe, select Getdata-> Get Data from the Simulator
7. Select Getdata-> Calculate IRFs and FRFs and select Frequency range.
8. After obtaining the IRFs and FRFs, select Estimate-> Automatic Estimate for automated estimation of modes, or Estimate->Interactive Estimate for more flexibility in mode identification. After this step, the required modes have been identified and the mode identification is complete. Results can be viewed by selecting DisplayCurves or Results.
9. Display Curves displays the impulse response of the slider to the impulse in z/p/r directions along with their IRFs and FRFs.
10. Results tab has options for displaying stiffness matrices, nodal lines and mode animations.
11. The results obtained can be stored by selecting File->Save Current Job for easy retrieval.

For more information on the parameter identification program please see CML blue report 97006

3.2.3.7. Disk and Slider Surface Topography Window

The user may specify arbitrary disk and slider surface topography using this window. Options include *disk asperities*, *slider asperities*, *disk waviness*, and *slider waviness* generation and *track profile* specification.

3.2.3.7.1. Numerical Generation of Topography

The user may specify topography of the disk or slider in a number of ways. The selection box at the top of the window offers the following choices: *Disk Waviness/Texture*, *Disk Asperities*, *Disk Zone Profiles*, *Slider Waviness/Texture*, and *Slider Asperities*. You may specify up to 40 waves, 40 asperities, and 40 zones on the disk and 40 waves and 40 asperities on the slider. Each topography option requires different parameters. Consequently, the input boxes will change when you select a new type of topography. You may scroll through the current topography settings by selecting the desired topography type and pushing the up and down arrows next to the Current indicator. To add an entry, enter the required parameters and push the *Add* button. To delete an entry, scroll to the desired entry and push the *Delete* button. To modify an entry, scroll to the desired entry, make the modification, and push the *Update* button.

3.2.3.7.1.1. Disk and Slider Waviness/Texture

The user may define a waviness or textured zone on a radial zone of the disk. By defining waviness, the user places on the disk or slider a series of bumps or waves with size defined by X Size and Y Size, separated by a distance of X Wavelength and Y Wavelength. An infinite size value corresponds to disk waviness in the specified zone, while a finite size corresponds to a series of evenly spaced asperities. Disk and slider wavinesses are defined in similar ways. The main differences are as follows:

- *Arbitrary* type waviness is not allowed on sliders.
- *Start* and *End Radii* are not used for sliders. Waviness is defined over the entire slider.
- *Angle* is defined with respect to the disk rotation direction for disks. *Angle* is defined with respect to the slider x-axis for sliders.

Type: The user has several options for the waviness type: *Sinusoidal*, *Ellipsoidal*, *Cylindrical*, *Rectangular*, and *Arbitrary* (Disk only). The *Arbitrary* option specifies a user defined cross-sectional profile as entered in the *Radius* and *Height* input boxes. With an *Arbitrary* type waviness, *X Size* and *Y Size* are ignored, and the *Amplitude* and *Angle* input boxes are changed to *Radius* and *Height*.

Amplitude: The waviness amplitude in meters (not used for *Arbitrary* type waviness).

Angle: The waviness orientation angle in degrees (not used for *Arbitrary* type waviness). For disk waviness, this is relative to the disk rotation direction. For slider waviness, this is relative to the slider x-axis.

Radius: (used only for *Arbitrary* type waviness) The user should enter a series of values, separated by spaces, corresponding to the radial distance in meters from the wave center. These will be used in conjunction with the series *Height* values to define a cross-sectional waviness profile. The number of entries in this box should be equal to the number of entries in the *Height* box.

Example: To specify an area with conical bumps, each with radius 1µm and height 0.5µm, the user would enter the following values:

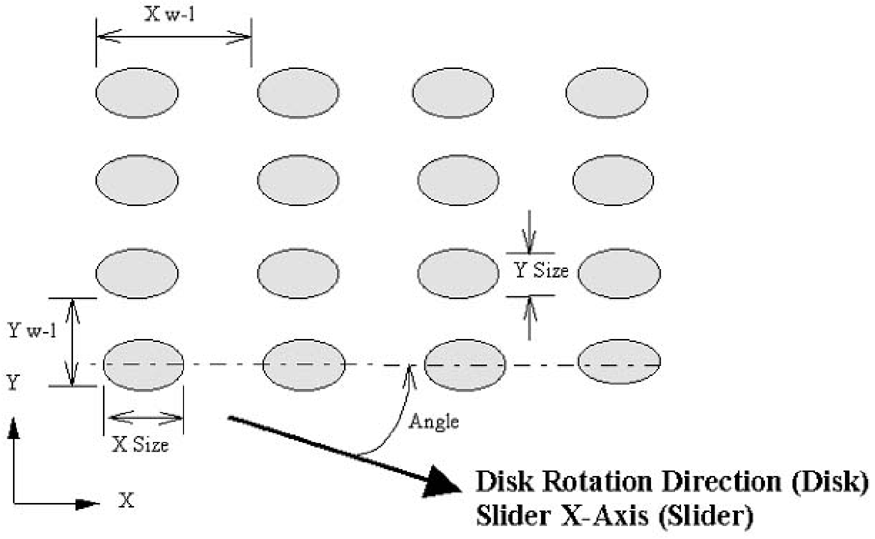
Radius: 0 0.000001
Height: 0.0000005 0

Height: (used only for *Arbitrary* type waviness) The user should enter a series of values, separated by spaces, corresponding to the height in meters of the point above the disk or slider surface. The number of entries in this box should be equal to the number of entries in the *Radius* box. See the above description for further details.

X, Y w-l: The wavelength or repetition distances in meters in the X and Y directions. A zero value indicates an infinite wavelength in that direction.

X, Y Size: (not used for *Sinusoidal* or *Arbitrary* type waviness) The pulse widths in meters in the X and Y directions. A zero value indicates an infinite pulse width in that direction.

Start, End Rad: Starting and ending radii in meters (*Disk Waviness* only).



3.2.3.7.1.2. Disk and Slider Asperities

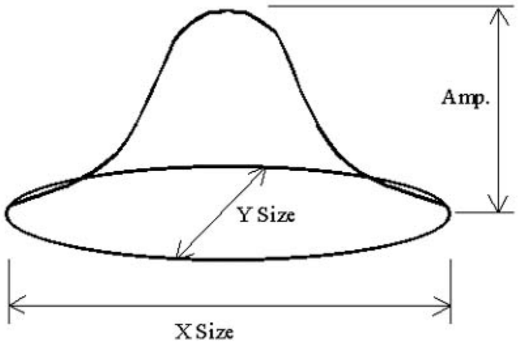
Asperities are defined in a manner similar to waviness. By defining an asperity, the user places a single bump on the disk or slider surface. The size of the bump is defined by *X Size* and *Y Size*, while the location of the bump is defines by

X Position and *Y Position*. Diagrams of the various asperity types follow the input parameter descriptions. The main differences between disk and slider asperities are as follows:

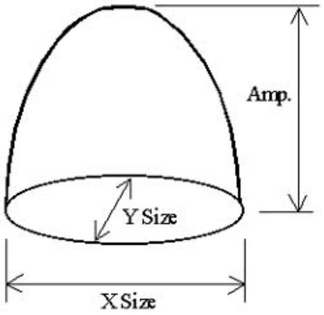
- *Arbitrary* type asperities are not allowed on sliders.
- For disk asperities, the X and Y axes are defined with respect to the slider gravity center, with positive X in the direction of slider motion and positive Y in the radial direction toward the OD. For slider asperities, the X and Y axes are defined with respect to the slider coordinate system, with the inner-leading corner at the origin. Diagrams of the above definitions follow.
- *Angle* is defined with respect to the disk rotation direction for disks. *Angle* is defined with respect to the slider x-axis for sliders.

Type: The user has several options for the asperity type: *Sinusoidal*, *Ellipsoidal*, *Cylindrical*, *Rectangular*, and *Arbitrary* (Disk Asperities only). The *Arbitrary* option specifies a user defined cross-sectional profile as entered in the *Radius* and *Height* input boxes. With an Arbitrary type asperities, *X Size* and *Y Size* are ignored, and the *Amplitude* and *Angle* input boxes are changed to *Radius* and *Height*.

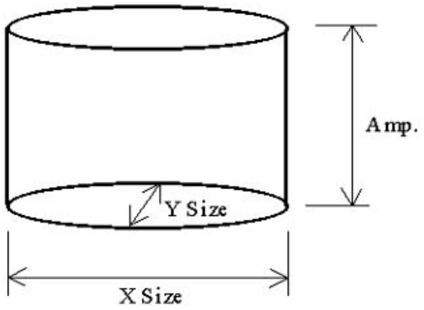
Sinusoidal



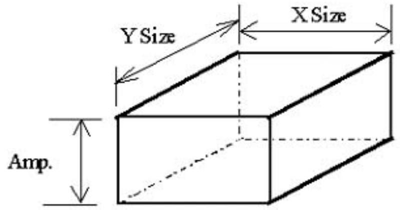
Ellipsoidal



Cylindrical



Rectangular



Amplitude: The asperity amplitude in meters (not used for *Arbitrary* type asperities).

Angle: The asperity orientation angle in degrees (not used for *Arbitrary* type asperities). For disk asperities, this is relative to the disk rotation direction. For slider asperities, this is relative to the slider x-axis.

Radius: (used only for *Arbitrary* type asperities) The user should enter a series of values, separated by spaces, corresponding to the radial distance in meters from the asperity center. These will be used in conjunction with the series *Height* values to define a cross-sectional asperity profile. The number of entries in this box should be equal to the number of entries in the *Height* box.

Height: (used only for *Arbitrary* type asperities) The user should enter a series of values, separated by spaces, corresponding to the height in meters of the point above the disk or slider surface. The number of entries in this box should be equal to the number of entries in the *Radius* box. See the above description for further details.

X, Y Pos.: Asperity center location at initial time. For disk asperities, it is measured from the slider's gravity center in the slider motion and radial directions, respectively. For slider asperities, it is measured from the inner-leading corner.

X, Y Size: Asperity footprint lengths. For disk asperities, measured in the slider motion and radial directions, respectively. For slider asperities, measured in the slider coordinate system, with the origin at the inner-leading corner. A zero value indicates an infinite size in that direction.

3.2.3.7.1.3. Disk Zone Profiles

The user may define up to 40 radial zones on which a zone profile is specified. The zone profile specifies a parabolic height function from the start radius to the end radius.

Rad1, Height1: The radial position and height in meters of the first boundary point.

Rad2, Height2: The radial position and height in meters of the middle profile point.

Rad3, Height3: The radial position and height in meters of the end boundary point.

3.2.3.7.1.4. Output Disk Topography

If a disk topography is specified, the user may enter three times (in ms) at which the solver will output a 3-d profile of the disk surface underneath the slider. These may be viewed in the postprocessor.

3.2.3.7.1.5. Point by Point Disk Track Profile

In order to accurately simulate a rough disk surface, the dynamic simulator allows the user to specify a 2-D point by point disk track profile. At each step of the simulation, the spacing at each point under the slider is obtained by a linear interpolation of the disk track profile data. The disk track profile input file, `wave.def` has the following format

```
x(1)  h(1,1)  h(1,2)  h(1,3) … h(1,nfy-1)  h(1,nfy)
x(2)  h(2,1)  h(2,2)  h(2,3) … h(2,nfy-1)  h(2,nfy)
x(3)  h(3,1)  h(3,2)  h(3,3) … h(3,nfy-1)  h(3,nfy)
…
…
x(nfx)  h(nfx,1) … h(nfx,nfy-1)  h(nfx,nfy)
y(1)  y(2)  y(3) …  y(nfy)
```

`x`: distance [m] along the disk in the x direction
`y`: radial position [m] along the disk
`h`: height of the disk [nm] at that point.
`nfx`: total number of points in the x direction
`nfy`: total number of points in the y direction

Enable PBP Profile: Enables/Disables Point by Point Disk Track Profile.

Init. Slider Location: Specifies the initial distance in mm of the slider's center of gravity from the zero track profile point.

Remove: Removes the current point by point disk track profile from the .cml file

Import: Import a point by point disk track profile into the interface. The file will be saved in the currently open .cml file.

3.2.4. Dynamic Mode - Postprocessor Menu

For the any of the 2-d plotting options (Flying Dynamics, Displacements, Disk Floor Heights, Track Access, Air Bearing Force, Contact Force), the user may right click and select Copy Data To Clipboard. This will copy the data in tab-delimited format to the clipboard for use in spreadsheet and data-plotting software such as MS Excel.

Output File: Displays the solver output if the simulation was run remotely and the output was saved.

Rails: Displays a diagram of the rails.

Grid: Displays a diagram of the final grid.

Flying Dynamics: Plots Nominal Fly Height, Pitch, Roll, and Off-Track Displacement versus time.

Displacements: Plots displacements measured from the mean disk surface at the four points of interest specified in the General Window.

Disk Floor Heights: Plots corresponding disk floor heights at four points of interest.

Track Access: Plots Radial Position, accessing Velocity, accessing Acceleration, Geometric Skew, Effective Skew, and Disk Speed versus time. The Effective Skew is the angle between the slider's length axis and the relative disk velocity vector.

Air Bearing Force: Plots Air Bearing Force, Normalized Force Center XF, Normalized Force Center YF, Positive Force, and Negative Force versus time.

Contact Force: Plots Normal Contact Force, Pitch Contact Moment, Roll Contact Moment, and Ratio of Real to Apparent Contact Area versus time.

Disk Topography: Plots disk surface profile at times specified in the Disk and Slider Surface Topography Window.

Air Pressure: Plots final air bearing pressure profile.

4. Input and Output Files

As mentioned previously, the CML static and dynamic solvers do not require the Windows interface. They can be run solely from the Windows or UNIX command line using ASCII input files. This section describes the input and output file formats used with the solvers. Generally, you do not need to know anything about the input and output file formats if you use the Windows interface, but it is sometimes helpful to look at the input files when debugging a problem. ASCII input and output files can be generated from the Windows interface by choosing *Export ASCII* from the *File* menu.

4.1. Quick Input Files

The rectangular static solver (Quick4) has two main input files: `rail.dat` and `run.dat`. It also accepts initial grid files: `x.dat` and `y.dat` as well as an initial pressure file `press_init.dat`. These files are described in this section.

4.1.1. rail.dat

Sample `rail.dat`:

```
CML Version 7.0  RAIL.DAT  4.26
REPORT BUGS TO INFO@CML.ME.BERKELEY.EDU
2.0500E-003      1.6000E-003      4.3000E-004
 4                2
 4                1
0.0000E+000      5.0000E-005      1
2.0500E-003      5.0000E-005      1
2.0500E-003      4.0000E-004      1
0.0000E+000      4.0000E-004      1
1.0000E-006
 5                1
0.0000E+000      1.5500E-003      1
1.2000E-003      1.5501E-003      1
2.0500E-003      1.5500E-003      1
2.0500E-003      1.2000E-003      1
0.0000E+000      1.2000E-003      1
1.0000E-006
 4                1
0.0000E+000      1.5000E-004      2
2.0500E-003      1.5000E-004      2
2.0500E-003      3.5000E-004      2
0.0000E+000      3.5000E-004      2
0.0000E+000
 4                1
0.0000E+000      1.5000E-003      2
2.0500E-003      1.5000E-003      2
2.0500E-003      1.3000E-003      2
0.0000E+000      1.3000E-003      2
0.0000E+000
 4 5
 0.0000E+000  6.0000E-006  1.2000E-005  2.0000E-005
 1.0000E-006  3.0000E-006  3.6000E-006  4.0000E-006
 -1.0000E-005 -5.0000E-006 0.0000E+000  5.0000E-006  1.0000E-005
 0.0000E+000  1.0000E-007  3.0000E-007  8.5000E-007  1.0000E-006
2.0000E-004      1.0000E-002      4.0000E-006
2.0000E-008      2.0000E-008      0.0000E+000
 2.0250E-003  1.9500E-003  2.0250E-003  1.9500E-003
 2.5000E-004  2.5000E-004  1.3500E-003  1.3500E-003
```

The first two lines are the header. These lines are required. Modification of these lines may cause read errors in both the solver and the interface. The third line contains the slider dimensions in meters: length, width, thickness. The fourth line indicates the number of rails and the number of (different) wall profiles used.

Next, each rail is defined in succession. The starting line for each rail describes the number of points and style. A step has a style value of 1, while ramp has a style value of 0. The lines that follow contain x and y coordinates of rail point and the wall profile index for the rail edge starting at that point. Note that x and y coordinates are now in meters instead of normalized, and the wall profile indices should be zero for a ramp. The final line in a rail description contains the recess height (or three heights for ramp) in meters for the rail.

The line that follows the rail definitions contains the number of wall profile points for all wall profiles. Next, each wall profile definition occupies two lines. The first line contains the normal distances to the nominal edge for the profile points. Note the unit is now in meters. A negative value indicates that the point is inside the nominal rail boundary, while a positive value otherwise. The second line contains recess depths in meters for each point. Note that the zero recess reference plane is the same as for all the rail recess depths, rather than the nominal recess for each rail.

The next line contains the taper length in meters, the taper angle in radians, and the base recess in meters. Note the change of units for the taper.

The following line gives the crown, camber and twist in meters. Note the change of order for camber of twist.

The final two lines are the x and y coordinates for the points of interest, respectively. Note the units are in meters. Up to 100 points of interest can be specified.

4.1.2. run.dat

Sample `run.dat`:

```
CML Version 7.0  RUN.DAT  4.24
REPORT BUGS TO INFO@CML.ME.BERKELEY.EDU
***************Solution Control***************
istiff isolv   ioldg   iadpt   isave
 1      1       0       1       1
***************Intial Attitude***************
hm(m)          pitch(rad)     roll(rad)
5.0000E-008    1.0000E-004    0.0000E+000
***************Runs***************
irad           irpm           ialt         sort
 1             1              0              0
radii(m)
   2.3000E-002
skews(deg)
   8.0000E+000
RPMs
   5.4000E+003
altitudes(m)

***************Air Parameters***************
p0(pa)         al(m)          vis(nsm^-2)
1.0135E+005    6.3500E-008    1.8060E-005
***************Load Parameters***************
f0(kg)         xf0(m)         yf0(m)        psa           rsa          usestiff
3.5000E-003    2.5000E-005    2.5000E-005   0.0000E+000   0.0000E+000  0
xfs(µNM)       yfs(µNM)       emax          skp           skr
5.0000E-001    5.0000E-001    1.0000E-003   1.7450E-005   1.7450E-005
***************Grid Control***************
nx      ny
 145     145
nsx      nsy     isymm
 1       1       0
xnt(i), i = 2, nsx

nxt(i), i = 2, nsx

dxr(i), i = 1, nsx
 1
ynt(i), i = 2, nsy

nyt(i), i = 2, nsy

dyr(i), i = 1, nsy
 1
```

```
***************Adaptive Grid****************
difmax_x        decay_x        ipmax_x      iadaptmode_x
 100            50             0            1
difmax_y        decay_y        ipmax_y
 40             50             0
********Mesh Refinement in X Direction*******
numXRegions
0
Lower         Upper          Amount
*******Mesh Refinement in Y Direction********
numYRegions
0
Lower         Upper          Amount
***************Reynolds Equation***************
ischeme      imdoel       akmax       beta         gamma     accommodation   mgnest
2            3            1.000E-007  6.000E+000   6.000E+000  1.000E+000    4
***************Partial Contact***************
icmodel        stdasp(m)      dnsasp(m^-2)   ConstantA      ConstantB
 1             0.0000E+000    1.0000E+012    1.0000E-019    1.0000E-076
rdsasp(m)      eyoung(pa)     yldstr(pa)     elecpot(v)
0.0000E+000    1.0000E+010    1.0000E+012    0.0000E+000
frcoe          pratio
 0.3            0.3
***************Sensitivities***************
crowninc(m)    camberinc(m)   twistinc(m)
0.0000E+000    0.0000E+000    0.0000E+000
tlnginc(m)     tanginc(rad)   loadinc(kg)
0.0000E+000    0.0000E+000    0.0000E+000
ptrqinc(µNM)   rtrquinc(µNM)  recessinc(m)
0.0000E+000    0.0000E+000    0.0000E+000
iwscale
 1
***********************Additional Parameters*******************
output_shear    use_imf
 0               0
***************Comments***************
"This is a test case."
```

The following are the variables in the `run.dat` file along with their counterparts in CMLAir. The units used in the `run.dat` file are in [brackets], which may differ from those listed in section 3.1.3. The interface will automatically convert to the proper units.

istiff:	0 = don't Calculate Stiffness; 1 = Calculate Stiffness.
isolv:	0 = Fixed Attitude; 1 = Fly Height; 2 = Geometry Only.
ioldg:	0 = Geometric Series; 1 = Existing Grid.
iadpt:	0 = Fixed Grid; 1 = Adaptive Grid.
isave:	0 = don't Save Pressure; 1 = Save Pressure.
hm:	TEC [m].
pitch:	Pitch [rad].
roll:	Roll [rad].
irad:	number of radii, generated automatically by Windows interface.
irpm:	number of RPMs, generated automatically by Windows interface.
ialt:	number of altitudes, generated automatically by Windows interface.
radii:	Radii [m].
skews:	Skews [deg].
rpms:	RPMs.

altitudes:	Altitudes [m].
p0:	Press [pa].
al:	MFP [m].
vis:	Viscosity [Ns/m²].
f0:	Load [kg].
xf0:	POffset [m].
yf0:	ROffset [m].
xfs:	PTorque [μNm].
yfs:	RTorque [μNm].
emax:	Load Error.
nx, ny:	Total Grid Sizes in X and Y directions, respectively.
nsx, nsy:	number of grid sections in x and y directions, respectively, automatically calculated in Windows interface.
xnt(i):	X Control Points, i from 2 to nsx [m].
nxt(i):	X Grid Indices, i from 2 to nsx.
dxr(i):	X Grid Ratios, i from 1 to nsx.
ynt(i):	Y Control Points, i from 2 to nsy [m].
nyt(i):	Y Grid Indices, i from 2 to nsy.
dyr(i):	X Grid Ratios, i from 1 to nsy.
difmax:	Max/Min.
decay:	Decay Factor.
ipmax:	0 = Maximum Pressure Gradient; 1 = Averaged Pressure Gradient.
ischeme:	convective term scheme. 0 = upwinds; 1 = hybrid; 2 = power-law, default. Not available in interface.
imodel:	Slip Model, 1 = 1st Order Slip model; 2 = 2nd Order Slip model; 3 = F-K Boltzman model.
akmax:	Reynolds Equation.
icmodel: icmod:	Partial Contact Model; 0 = No Conact Model; 1 = Greenwood-Williamson; 2 = Elastic Plastic.
stdasp:	STD of Asperity Height [m].
dnsasp:	Asperity Density [1/m²].
rdsasp:	Radius of Curv. [m].
eyoung:	Young's Modulus [Pa].
yldstr:	Yield Strength [Pa].
frcoe:	Friction Coefficient.
pratio:	Poisson's Ratio.
crowninc:	Crown Sensitivity [m].
camberinc:	Camber Sensitivity [m].
twistinc:	Twist Sensitivity [m].
tlnginc:	Taper Length Sensitivity [m].
tanginc:	Taper Angle Sensitivity [rad].
loadinc:	Load Sensitivity [kg].
ptrqinc:	Pitch Torque Sensitivity [μNm].
rtrqinc:	Roll Torque Sensitivity [μNm].
recessinc:	Base Recess Sensitivity [m].
iwscale:	0 = Fixed Width; 1 = Proportional.
comments:	Comments.

4.2. Quick Output Files

The rectangular grid static solver produces three main output files: `result.dat`, `x.dat`, and `y.dat`. These three output files are generated on every successful solver run. `result.dat` is a summary of the results of the simulation. It contains information on the flying attitude, air pressure force, contact force, and stiffness for each case of the simulation. `x.dat` and `y.dat` are the final grid files generated by the solver. They are each one line with space delimited values corresponding to the final mesh in the respective directions. These files can be used as initial grid files in subsequent simulations. As of CMLAir 7.0, Quick 4 can also output grid files for each radial position in a multi run simulation. The grid files are named x*xx*.dat and y*xx*.dat where *xx* corresponds to the run number.

If the Save Pressure (isave) option is on, four pressure files are generated. The files `pressxx.dat`, `cprssxx.dat`, `imfxx.dat`, and `mflowxx.dat` are generated for each radial position of the simulation, where *`xx`* corresponds to the radius number. These contain the air pressure matrix, contact pressure matrix, and mass flow matrix, respectively. These files have the same size and format. They are space-delimited matrices, with one entry for each point in the grid. `pressxx.dat`, `imfxx.dat` and `cprssxx.dat` use the files `x.dat` and `y.dat` as the grid data, while `mflowxx.dat` uses `xm.dat` and `ym.dat`. The files `xm.dat` and `ym.dat` will also be generated by the solver. The file `geom.dat` will also be generated when either *Save Pressure* is selected on or the *Geometry Only* option is selected. This file contains the slider geometry over the mesh defined by `x.dat` and `y.dat`.

4.3. Dyn Input Files

The CML Dynamic Simulator uses two main input files: `rail.dat` and `dynamics.def`. The format of `rail.dat` is described in the previous section. This section focuses on `dynamics.def` and other dynamic input files. Sample `dynamics.def`:

```
CML Version 4.039  DYNAMICS.DEF
  **********************Problem Definition*******************
  xl(m)      yl(/xl)    xg(/xl)    yg(/xl)    zg(/xl)    halt
  0.00085    0.823529   0.5        0          0.135294   0
  f0(kg)     xf0(/xl)    yf0(/yl)   amz        aip        air
  0.0008     0.5        0          1.0722E-006 1.75841E-013 1.28664E-013
  rpm        dt(s)      tf(s)      ra         ra_if      ra_of
  10000      1E-007     0.003      0.0171     0.015      0.045
  ************************Suspension*************************
  iact       xact(m)    dact       vact       ske(deg)
  1          0.038      0          0          1.1
  isusp      nmodes     ncg        alfa       beta
  0          0          0          0          1E-005
  skz        skp        skr        scz        scp        scr
  18         1.745E-005 1.745E-005 1E-010     1E-017     1E-017
  psa        rsa
  0          0
  ******************Initial Flying Condition*****************
  hm(m)      hp(rad)    hr(rad)    vz(m/s)    vp(rad/s)  vr(rad/s)
  4.4380E-009  1.6310E-004  1.8330E-006  0.0000E+000  0.0000E+000  0.0000E+000
  *******************Solution Control************************
  iqpo       akmax      emax       idisc
  5          1E-007     0.0001     1
  ********************Grid Control***************************
  iadapt     isymmetry  ioldgrid   nx   ny   nsx   nsy
  1          0          0          306  306  1    1
  xnt(i), i = 2, nsx

  nxt(i), i = 2, nsx

  dxr(i), i = 1, nsx
   1
  ynt(i), i = 2, nsy

  nyt(i), i = 2, nsy

  dyr(i), i = 1, nsy
   1
  difmax     decay      ipmax
  40.000000  50.000000  0
```

```
***********Point by Point Disk Track Profile***************
ims        nfx        dinit      nfy
0          0          0.000000   0
******Numerical Generation of Disk Surface Topography******
nwave  nzone  nasper
0      0      0
iwtype wamp(m)  wang(dg) wthx(m)  wthy(m)  wpdx(m)  wpdy(m)  wrs(m)   wre(m)
zr1(m)    zh1(m)   zr2(m)   zh2(m)   zr3(m)  zh3(m)
iatype aamp(m)   aang(dg)  alocx(m)  alocy(m)  asizx(m)  asizy(m)
****Numerical Generation of Slider Surface Topography******
nswave  nsasper
0       0
istype  swamp(m)   swng(dg)   swthx(m)   swthy(m)   swpdx(m)   swpdy(m)
isatype saamp(m)   saang(dg)  salocx(m)  salocy(m)  sasizx(m)  sasizy(m)
******************Track Accessing Motion*******************
nap
0
tac(s)     aac(rad/s^2)
*************Time-Dependent Disk Velocity*****************
nvp
0
tvp(s)     vtd(RPM)
****************Sinusoidal Disk Flutter*******************
iflut   tsft      teft      fqft      amft
0       0.000000  0.003000  100.000000 1E-008
********************Intermolecular Forces**********************
imfmode    ConstantA    ConstantB    elecpot
0          1.0000E-019  1.0000E-076  0.000000
********************Asperity Contact**********************
icmod   ey          ydst     pratio   frcoe
2       1E+010      1E+012   0.3      0.3
ncz
0
sikm(m)    ceta(/m/m) rasper(m)  rcts(m)    rcte(m)    glidh(m)
***************************Shock**********************************
useshock  normal    pitch     roll      tstart    tend
0         0         0         0         0         0
***********************Additional Parameters*******************
p0(pa)          al(m)           vis(nsm-2)
1.0135E+005     6.3500E-008     1.8060E-005
init_disp_x     init_disp_y     init_disp_yaw
0.0000E+000     0.0000E+000     0.0000E+000
init_vel_x      init_vel_y      init_vel_yaw
0.0000E+000     0.0000E+000     0.0000E+000
yaw_inertia     num_susp_nodes   slider_node   nairflow
2.3400E-012     6               3               0
damping_a       frequency_a     damping_b      frequency_b
10.000000        10.000000       10.000000       50000.000000
***********************End of Input Data*******************
```

The following are the variables in the `dynamics.def` file along with their *counterparts* in CMLAir. The units used in the `dynamics.def` file are in [brackets].

xl: Length [m].
yl: Width [normalized with xl].
xg: X Grav [normalized with xl, starting from leading edge].
yg: Y Grav [normalized with xl, starting from center towards outer edge].

zg:: Z Grav [normalized with xl].
halt: Altitude [m].
f0: Load [kg].
xf0: Pitch Offset [normalized with xl, starting from leading edge].
yf0: Roll Offset [normalized with xl, starting from center towards outer edge].
amz: Mass [kg].
aip, air: Pitch and Roll MOI, respectively [kg× m^2].
rpm: RPM. It is initial rpm if the disk speed is time-dependent (nvp > 0).
dt, tf: Time Step and Total, respectively [s].
ra: rof: Min. Inner Rad. and Max Outer Rad., respectively [m].
iact: 0 = No Actuator; 1 = Inline Actuator
xact: Arm Length [m].
dact: Angular Position [rad].
vact: Angular Velocity [rad/s].
skew: Skew[deg].
isusp: 0 = Stiff. and Damp; 1 = Modal Analysis (depreciated); 2 = Finite Element Suspension
nmodes: # of Mode Shapes.
ncg: Center Node #.
alfa: Alpha [1/s].
beta: Beta [s].
skz: Vetical Stiffness [N/m].
skp, skr: Pitch and Roll Stiffness, respectively [N×m/rad].
scz: Vertical Damping [N×s/m].
scp, scr: Pitch and Roll Damping [N×m×s/rad].
hm: TEC [m].
hp, hr: Pitch and Roll, respectively [rad].
vz: Vert. Vel. [m/s].
vp, vr: Pitch and Roll Vel., respectively [rad/s].
iqpo: Slip Model, 0 = continuum; 1 = 1st Order Slip model; 2 = 2nd Order Slip model; 5 = F-K Boltzman
model. Only 1, 2, and 5 are available using the Windows interface.
akmax: Reynolds Equation.
emax: Load Error.
idisc: different schemes for treating the convective term. 1 = power-law; 2 = central difference; 3 = upwind; 4 = hybrid. idisc = 1 is recommended. This is not available in the Windows interface.
iadpt: 0 = Fixed Grid; 1 = Adaptive Grid
isymmetry: 0 = no Symmetry in Width; 1 = Symmetry in Width
ioldgrid: 0 = Geometric Series; 1 = Existing Grid
nx, ny: Total Grid Sizes in X and Y directions, respectively.
nsx, nsy: number of grid sections in x and y directions, respectively, automatically calculated in Windows interface.
xnt(i): X Control Points, i from 2 to nsx [normalized with xl].
nxt(i): X Grid Indices, i from 2 to nsx.
dxr(i): X Grid Ratios, i from 1 to nsx.
ynt(i): Y Control Points, i from 2 to nsy [normalized with xl].
nyt(i): Y Grid Indices, i from 2 to nsy.
dyr(i): X Grid Ratios, i from 1 to nsy.
difmax: Max/Min.
decay: Decay Factor.
ipmax: 0 = Maximum Pressure Gradient; 1 = Averaged Pressure Gradient.
ims: 0 = disable Point by Point Track Profile; 1 = directly read the Point by Point Disk Track Profile from the file wave.def.
nfx: number of track profile points in the X direction. Calculated automatically in Windows interface.
nfy: number of track profile points in the Y direction. Calculated automatically in Windows interface.
dinit: Init. Slider Location [normalized with xl].

nwave: total number of waves. Correspondingly, there should be the same number of lines to define these waves following the 'iwtype' line. Maximum nwave is 40. This is calculated automatically by the Windows interface.

nzone: number of the radial zone profiles. Correspondingly, there should be the same number of lines to define the zone profiles following the 'zr1' line. Each zone profile is defined by three radial points. The program can input up to 40 radial zone profiles. This is calculated automatically by the Windows interface.

nasper: number of asperities. Correspondingly, there should be the same number of lines to define the asperities following the 'iatype' line. Up to 40 asperities can be distributed over the disk surface. This is calculated automatically by the Windows interface.

iwtype: Disk Waviness Type. 0 = Sinusoidal; 1 = Ellipsoidal; 2 = Cylindrical; 3 = Rectangular; 4 = Arbitrary profile defined in files: w1.dat, w2.dat, etc.

wamp: Disk Waviness Amplitude [m]. Note that for iwtype=4 this parameter is used to indicate the input profile number, e.g., 1.0 corresponds to w1.dat.

wang: Disk Waviness Angle with respect to the disk rotation direction [deg].

wthx,wthy: Disk Waviness X w-l and Y w-l, respectively [m].

wpdx,wpdy: Disk Waviness X Size and Y Size, respectively [m].

wrs, wre: Disk Waviness Start Radius and End Radius, respectively [m].

zrn, zhn: Radn and Heightn (n=1, 2, 3), respectively [m].

iatype: Disk Asperity Type. 0 = Sinusoidal; 1 = Ellipsoidal; 2 = Cylindrical; 3 = Rectangular; 4=Arbitrary profile defined by a1.dat, a2.dat, etc.

aamp: Disk Asperity Amplitude [m]. Note that this parameter is used to indicate the input profile number for iatype=4, e.g., 1.0 corresponds to a1.dat.

aang: Disk Asperity Angle with respect to the disk rotation direction [deg].

alocx,alocy: Disk Asperity X and Y Pos. measured from the slider's gravity center at initial time in disk rotation and radial directions, respectively [m].

asizx,asizy: Disk Asperity X and Y Size, respectively [m].

nswave: total number of waves on slider surface. Correspondingly, there should be the same number of lines to define these waves following the 'iswtype' line. Maximum nswave is 40. This is calculated automatically by the Windows interface.

nsasper: number of asperities on slider surface. Correspondingly, there should be the same number of lines to define the asperities following the 'isatype' line. The program can generate up to 40 asperities over the slider surface. This is calculated automatically by the Windows interface.

iswtype: Slider Waviness Type. 0 = Sinusoidal; 1 = Ellipsoidal; 2 = Cylindrical; 3 = Rectangular.

swamp: Slider Waviness Amplitude [m].

swang: Slider Waviness Angle with respect to x direction [deg].

swthx,swthy: Slider Waviness X w-l and Y w-l, respectively [m].

swpdx,swpdy: Slider Waviness X Size and Y Size, respectively [m].

isatype: Slider Asperity Type. 0 = Sinusoidal; 1 = Ellipsoidal; 2 = Cylindrical; 3 = Rectangular.

saamp: Slider Asperity Amplitude [m].

saang: Slider Asperity Angle with respect to x direction [deg].

salocx, salocy: Slider Asperity X Pos and Y Pos, respectively [m].

sasizx,sasizy: Slider Asperity X Size and Y Size, respectively [m].

nap: number of points to follow which describe the acceleration/deceleration profile. A linear functional relationship is assumed to exist between adjacent data points. The program can input up to 40 data points. This is calculated automatically by the Windows interface.

tac: Track Seek Time [s].

aac: Track Seek Angular Acceleration [rad/s^2].

nvp: number of points used in describing the time-dependent disk velocity profile. Correspondingly, there should be the same number of data lines to define the speed profile following the 'tvp' line. The program can take up to 40 data points. If the simulation proceeds to times greater than that of data input, the value of velocity at the largest time will be used. This will be calculated automatically by the Windows interface.

tvp: Disk Velocity Time [s].

vtd: Disk Speed [rpm].

iflut: 0 = disable Disk Flutter; 1 = enable Disk Flutter.

tsft, teft: Disk Flutter Start Time and End Time, respectively [s].

fqft: Disk Flutter Frequency [HZ].

amft: Disk Flutter Amplitude [m].

icmod: Partial Contact Model; 1 = Greenwood-Williamson; 2 = Elastic Plastic.

ey: Young's Modulus [Pa].

ydst: Yield Strength [Pa].

pratio: Poisson's Ratio.

frcoe: Friction Coefficient.

ncz: number of radial contact zones. There should be the same number of line data to follow which describe the roughness of each zone. Maximum ncz is 5. This is calculated automatically by the Windows interface.

sikm: STD of Asperity Height [m].

ceta: Asperity Density $[1/m^2]$.

rasper: Radius of Curv. [m].

rcts, rcte: Contact Zone Start Radius and End Radius, respectively [m].

glidh: Glide Height [m].

useshock:: 0 = No Shock Simulation; 1 = half sinusoid shock; 2 = user defined shock

normal: aplitude of shock in z direction (N) (only used for half sinusoid shock)

pitch: aplitude of shock moment in pitch direction(NM) (only used for half sinusoid shock)

roll: amplitude of shock moment in roll direction (NM) (only used for half sinusoid shock)

tstart: shock start time (seconds) (only used for half sinusoid shock)

tend: shock end time (seconds) (only used for half sinusoid shock)

p0: Pressure [pa].

al: Mean Free Path [m].

vis: Viscosity $[Ns/m^2]$.

init_disp_x: initial displacement of the slider in the X direction [m]

init_disp_y: initial displacement of the slider in the Y direction [m]

init_disp_yaw: initial rotation of the slider about the load point[rad]

init_vel_x: initial velocity of slider in the X direction [m/s]. This parameter cannot be changed from within CMLAir

init_vel_y: initial velocity of slider in the Y direction [m/s]. This parameter cannot be changed from within CMLAir

init_vel_yaw: initial yaw velocity of the slider [rad/s]. This parameter cannot be changed from within CMLAir

yaw_inertia: yaw moment of inertia (MOI) [kg*m2]

num_susp_nodes: number of nodes in the reduced order suspension (should be equal to the number of lines in suspmass.dat)

slider_node: specifies which node in the reduced order suspension connects the slider to the suspension. If the reduced order suspension arrays were generated by CML's Reduction.exe software, this will always be 3.

nairflow: number of lines listed in the airflow file

damping_a: percentage damping desired at frequency_a

frequency_a: first frequency for specifying Raleigh damping

damping_b: percentage damping desired at frequency_b

frequency_b: second frequency for specifying Raleigh damping

Also including in the `dynamics.def` file are three numbers following the Disk Surface Topography section (one line above Numerical Generation of Slider Surface Topography). These three values are required and correspond to the three Output Disk Topography times in seconds.

Other input files for the dynamics simulator are `wave.def`, `wn.dat`, `an.dat`, `fmodes.ext` and `fmesh.ext`. The file `wave.def` is needed for the simulation using the point by point disk track profile (ims = 1). The file format is explained in section 3.2.3.7.1.5. For iatype=4, the input files to define the cross-sectional profiles for the asperities, `a1.dat`, `a2.dat`, etc. are needed. For iwtype=4, the input files to define the cross-sectional profiles for the waviness, `w1.dat`, `w2.dat`, etc. are needed. The format for `a1.dat` or `w1.dat` is as follows: the first line specify the number of points used to specify the profile (less than 50 in current version), then the coordinates (radius and height) of those points (in meters) ensue.

4.4. Dynamic Output Files

The output file `fhhist.dat` contains slider settling history. There are five columns of data: time (s) in column 1, nominal fly height (m) in column 2, pitch (rad) in column 3, roll (rad) in column 4 and off-track displacement (m) in column 5. The file `hpoint.dat` stores the displacements (measured from the mean disk surface) at the four points defined in `rail.dat`. The corresponding disk floor heights at these four points are contained in the file `hfloor.dat`. The file `track.dat` stores the following data: time (s) in column 1, radial position (m) in column 2, accessing velocity (m/s) in column 3, accessing acceleration (m/s2) in column 4, geometrical skew angle (deg) in column 5, effective skew angle (deg) in column 6 and disk speed (rpm) in column 7. The effective skew angle is the angle between the slider's length axis and the relative disk velocity vector. Minimum flyheights are stored in `fhmin.dat` the first column contains time data, the second is the minimum flyheight of the slider without taking the disk topography, roughness, waviness, and disk flutter into account (flying over a perfectly smooth disk). The third and fourth columns contain the X and Y location of this minimum on the slider. The fifth column contains the minimum flying height when taking all disk roughness, waviness, topgraphy and flutter into account. The sixth and seventh columns contain the X and Y location of this minimum point on the slider. As of Dyn4.4, two new files are output from the simulator: `displacement.dat` and `loaderrors.dat`. The columns of `displacement.dat` contain the time [s], nominal flyheight [m], pitch [rad], roll [rad], x displacement [m], y displacement [m], yaw [rad] and skew [deg] of the slider. The file `loaderrors.dat` contains the load errors along the three translational and three rotational degrees of freedom corresponding to the slider node.

Air bearing force history is stored in the file `aload.dat` which has four columns of data: time (s) in column 1, air bearing force (kg) in column 2, normalized air bearing force center XF in column 3, normalized air bearing force center YF in column 4, positive air bearing force (kg) in column 5, and negative air bearing force (kg) in column 6. The contact force history is saved in the file `contact.dat`. The file contains five columns of data: time(s) in column 1, the normal contact force (kg) in column 2, the contact moments in pitch direction (column 3) and roll direction (column 4) (N×m) as well as the ratio of the real to apparent contact area in column 5.

The files `x.dat` and `y.dat` contain the normalized x and y coordinates, respectively. The pressure matrix is stored in the file `p2.dat`. The files `disktop1.dat`, `disktop2.dat` and `disktop3.dat` store the disk topography at the specified time.