CML

Computer Mechanics Laboratory

Technical Report No. 99-024

# The CML Air Bearing Design Program (CMLAir32), Version 5 User Manual

By

Ryan Grisso and D.B. Bogy

Department of Mechanical Engineering
University of California
Berkeley, CA 94720

November, 1999

# Table of Contents

# 1. Introduction

This report is an updated manual for the CML Air Bearing Design Program version 5 for PC including the new 32-bit MS-Windows interface. 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 Solvers (Quick4 and Quick5) and the CML Air Bearing Dynamic Simulator are actually two separate programs, they can all be used in an integrated environment with the Air Bearing Design Program for PC (CMLAir32). Users of the previous versions of the CML Air Bearing Design Program should have little or no trouble adjusting to the new interface, as it is very similar to earlier versions.

Strictly speaking, you do not need the Windows interface in order to use CML simulation software. All functions of the solvers can be accessed using ASCII input and output files. In fact, if you have only a UNIX machine, you will not use the interface. You will find, however, that the Windows interface vastly simplifies the air bearing design and analysis process.

## 1.1. Solvers

The current distribution of the Air Bearing Design Program for PC contains three solvers: quick4, quick5, and dyn4, also known as the CML Air Bearing Rectangular Grid Solver, CML Air Bearing Triangular Grid Solver, and the CML Air Bearing Dynamic Simulator. These solvers are written in Fortran77, C++, and Fortran 77 respectively. They have been compiled as 32-bit Windows console applications using Digital Visual Fortran and Microsoft Visual C++ 5.0. If you plan to run simulations using only a Win95/98/NT PC, you will not need to download any additional files. If you plan to run simulations using a UNIX workstation, you will need to download the solver distribution files and install them on your workstation. UNIX installation is described in Section 2.2. Currently, Quick5 is only available as a compiled PC executable.

## 1.2. Windows Interface

The Windows interface can be run on any Win95/98/NT 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, local and remote solver capability, VMS and NT remote support, and many others. For a full list of changes from previous versions, see the `readme.txt` file included in the distribution.

# 2. Installation

This section describes the installation process for PC and UNIX workstations. PC installation is trivial, while UNIX installation is somewhat more involved. If you are unfamiliar with UNIX commands, you may want to contact someone who is familiar to assist you with UNIX. This applies to the solver installation as well as the PC network setup for remote UNIX runs.

## 2.1. Win95/98/NT Installation

If you have a previous version of CMLAir32, it is recommended that you uninstall it before you install the new version. To uninstall a previous version of CMLAir32, open the "Control Panels" window from the Windows Start Menu and choose "Add/Remove Programs". You can then choose the previous version of CMLAir32 from the list and push the "Add/Remove" button.

Now, download the installation file from the CML web page: `http://cml.me.berkeley.edu`. The file will be called something like "`CMLAir32_v5_install.exe`". Save the file in a temporary directory somewhere on your hard drive (for instance, c:\temp). Now, using Windows Explorer, double click on the "`CMLAir32_v5_install.exe`" icon. This will begin the installation procedure. After the installation is complete, you may double click on the "CMLAir32" icon or choose "CMLAir32" from the Windows Start Menu. If you see a large 3-d image of the CML5 logo, you have successfully installed the CML Air Bearing Design Program version 5 for PC and are ready to design and simulate air-bearing sliders.

## 2.2. UNIX Solver Installation

Currently, there is no UNIX version of Quick5.

There is a separate UNIX source distribution for each available solver: quick419.tar.Z and CML_dyn434.tar.Z. You will need to uncompress and unpack the archive file in order to compile and install the solver on your UNIX workstation. The procedure to uncompress and unpack the archive is identical for each file:

```
uncompress quick419.tar.Z ; tar xvf quick419.tar
uncompress CML_dyn434.tar.Z ; tar xvf CML_dyn434.tar
```

These commands will create two sub-directories: `quick419` and `CML_dyn434`.

## 2.3. Compiling the Solvers

To compile the solvers, you will need a Fotran compiler. Most UNIX workstations have such a compiler, normally `f77`, `f90`, or `g77`. To compile `quick419`, you will also need a make utility, which is standard with most UNIX systems. If you do not have a Fortran compiler or make utility, you can download GNU Fortran (`g77`) or GNU Make (`gmake`) from `www.gnu.org`.

The following is a summary of the instructions to compile each program. If you encounter any problems, read sections 2.3.1 and/or 2.3.2.

**Quick4:**

1. **Change to the top level directory (`quick419`)**
2. **type '`make`'**

**Dyn:**

1. **Change to a source directory (`SRC_DEC` or `SRC_IBM`)**
2. **type '`f77 -O3 -o dyn434 dyn434.f`'**

### 2.3.1. Compiling Quick4

The following is a description of the contents of the `quick419.tar` archive:

Files in the top level directory are `make` utilities: `makefile` and `sys_def`. They are only useful if the user wants to recompile the source code.

Sub directory `src/` contains the source code and includes files: `version.fi`, `size.fi`, `common.fi`, `openout.fi`, `quick.f`, `reynolds.f`, `init.f`, `grid.f`, `mult.f`, `misc.f`, `util.f`, `force.f`, `inv.f`.

Sub directory `ibm/` contains the pre-compiled binary file `quick419` for IBM RS/6000 systems.

Sub directory `dec/` contains the pre-compiled binary file `quick419` for DEC Alpha machines.

Sub directory `pc/` contains the Windows executable `quick419.exe` generated with Digital Visual Fortran.

Sub directory obj/ contains the object codes generated during compilation. It is empty at the time of distribution.

Sub directory sample/ contains a set of sample data files: rail.dat and run.dat. They can be used to check whether the solver is properly installed. The Air Bearing Design Program for PC should also be able to read them once installed.

For platforms where the binary file is available, the user needs only the binary file. After setting the proper path for the binary, simply go to the sample/ directory and try to run it with the sample data files. The user must re-compile the code, however, if the target platform is not directly supported. Re-compilation is also necessary when the user wishes to change some array limits. The process of re-compilation is discussed in the next.

Some make utilities are provided to facilitate user compilation. This distribution is geared toward compilation on UNIX systems. It can, however, be compiled on PCs using many available PC based Fortran compilers. The executable in the distribution was compiled using Visual Digital Fortran. The following instructions describe the compilation procedure for both Visual Fortran and UNIX based command line compilers.

To recompile the source code using Visual Fortan, create a new Windows 32-bit Console project. Add all ".f" and ".fi" files in the src/ directory to the project. Then "Build". This will generate a new Windows Console application.

For command line compilers and UNIX platforms, if nothing in the source code needs to be altered, **simply type 'make'** in the top distribution directory on all platforms. This should create a file called quick419 in the top-level directory. You can now move the executable to a convenient location such as /usr/local/bin or /usr/local/CML.

You may wish to check the "sys_def" file in the top-level directory to ensure the settings are correct. This file also contains the settings for the Watcom PC compiler. You should comment out the UNIX lines and uncomment the Watcom lines if you wish to use the Watcom compiler.

The binary file generated by make is in the top-level directory. On UNIX machines, it is named quick419, while on PC it is quick419.exe.

In most cases, the only files the user wants to change are size.fi and openout.fi. The size.fi file sets the grid size, rail and wall profile array size limits. These variables are self-descriptive and can be easily modified. The second file deals with the problem of writing and reading many numbers in one long line. If there is an input-output problem apart from the initial reading of rail.dat and run.dat files, check this file and make modifications. It is self-explanatory.

If the user does not often modify the code and re-compile, the object files in obj/ generated by the compiler can be deleted to save some disk space. Just type "make clean" after compilation.

## 2.3.2. Compiling Dyn

The following is a description of the contents of the dyn434.tar archive:

The top-level directory contains only a README.1st file. You may want to look at this file before compiling or using the program.

Sub directories SRC_DEC/, SRC_IBM/, and SRC_BOTH/ contains the source code and includes files: common.fi, dyn434.f.

Sub directory BIN/ contains the Windows executable dyn434w.exe generated with Digital Visual Fortran.

Sub directories BUMP/, CSS/, FEM/, IMPULSE/, SEEK/, and SPACING/ contain sets of sample data files. The Air Bearing Design Program for PC should also be able to read them once installed.

Sub directory FEM/ contains FEM input files for a suspension.

Sub directory `POST/` contains Matlab post processing utilities.

To compile `dyn434`, type the following:

```
cd SRC_DEC
f77 -o dyn434 dyn434.f
```

This will create a file called `dyn434` in the current directory. You should, of course, use the appropriate source directory to compile. The above instructions assume the use of the DEC Alpha source directory. You can now copy the executable to a convenient location such as `/usr/local/bin` or `/usr/local/CML`.

## 3. The CML Air Bearing Design Program for PC

The CML Air Bearing Design Program for PC can be invoked by double clicking the program icon for CMLAir32 in Windows. When the interface starts, you will see a window with the CML5 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 types of solver: quick and dyn. Each mode has different pre-processing and post-processing functions. If you are familiar with previous versions of CMLAir, you will recognize the static mode functions, but will probably be unfamiliar with the dynamic mode functions.

First, a few words about file management: In previous versions of CMLAir32, each case was composed of several input files. The input files for each case would be stored in their own directory. In version 5, all input and output files are bundled in a single "`.cml`" file. This file is a compressed binary format particular to CMLAir32. Many files may be stored in the same directory without conflicts. When simulations are run, the ASCII result files generated by the solvers are imported automatically into the "`.cml`" format.

To import a set of ASCII input and output files from a previous version of CML software, choose the "Import ASCII" function from the File menu. To save a case in the older ASCII format, 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. Simulations will be run using either Quick4 or Quick5, depending on the type of grid selected in the Grid Window. 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 the "Static/Dyn" button 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*, *Post 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 read and converted to binary .cml format.

*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`, `trigrid.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).

*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.

*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 Optins Menu are as follows: *Solution*, *Normalized*, *Static Mode*, *Dynamic Mode*, *Solver Interaction*, *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*: Does nothing in 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. The type of run depends on the parameters set in the *Solver Interaction* and the Grid parameters dialog box.

*Toolbar*: Toggles the visibility of the toolbar.

*Small Toolbar*: Toggles between large buttons and small buttons for 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. The user may choose either a *Local Solver* (run the solver under MS Windows on the local computer) or a *Remote Solver* (run the solver on a remote UNIX machine).

Most users will use the former option, however if you have access to a fast UNIX workstation, you may wish to use the latter.

The solver interaction in version 5 has had some major modifications. Because of the new file structure, simulations are run in a temporary directory. When a run begins, a temporary directory is created in the same directory as the current document. The ASCII input files are exported to this temporary directory, and the simulation begins using these ASCII files. For local runs, the solver is launched in the temporary directory. For remote runs, the ASCII files are FTP'd to a remote host, the solver is launched remotely, and the result files are FTPd back to the local temporary directory. When the simulation is complete, the ASCII result files are imported into the interface, and the temporary directory is removed. Sometimes, errors can cause the temporary directory to remain after the job is finished. In this case, a warning message may 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 MS-Windows 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 MS-Windows Quick5 executable. This solver is used for a local static simulation using a triangular 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 Dyn4 Loc.*: Location of the local MS-Windows 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.

## 3.1.2.1.2. Remote Solver

In order to utilize the remote solver feature, you must have the following:

1. A TCP/IP network connection on your PC
2. A networked remote machine with a compiled version of the CML solver, an FTPd and an REXECd.
3. An account with FTP and REXEC access on the remote machine.

When a case is run remotely, the following procedure is followed:

a. The current case configuration is saved in .cml binary format.
b. A temporary directory is created in the same directory as the case file
c. ASCII input files are exported into this directory
d. An FTP connection is established with the remote machine, the appropriate directory is chosen based on current active remote jobs, and input files are transferred from the temporary local directory to the remote directory.
e. The remote solver is started using the specified script or a preset UNIX command:

   `SCRIPT DIR SOLVERPATH` (script syntax)
   or
   `cd DIR; SOLVERPATH` (present UNIX command)

   (Note, if you are using a script, you should make sure your script accepts a directory name as the first argument and the remote solver name as the second argument. See the `README.TXT` file included with the distribution for more details and an example.)

f. When the remote executable completes, ASCII result files are transferred back to the local temporary directory.
g. ASCII result files are imported into binary .cml format.

For more information about using the Remote Solver feature, please check the Appendix of this manual or the `README.TXT` file in the distribution.

*Server Name*: The domain name of the computer on which the solver resides.

*Remote Quick4*: The full pathname of the remote Quick4 executable file.

*Remote Quick5*: The full pathname of the remote Quick5 executable file.

*Remote Dyn4*: The full pathname of the remote Dyn4 executable file.

*Username*: The username of the account that will be used on the remote machine.

*Password*: The password of the account that will be used on the remote machine. The password is stored encrypted on the local hard disk. If you choose to leave this box blank, you will be asked to supply your password each time you run a remote job.

#### 3.1.2.1.3. Remote Options

*Remote Directory*: The relative or absolute pathname of the root directory to be used for running remote simulations. If the directory does not exist on the remote machine, it will be automatically created during the first remote run.

*Erase When . . .:* When checked, all input and output files will be erased on the remote machine when the simulation is complete. You will, of course, still have access to the files locally. If not checked, the files will remain until another case is run in this directory.

*Save Output*: When checked, the remote output will be saved to the specified text file for later viewing. This is not necessary in version 5 since all output is always piped to the interface and saved for future viewing. Some users, however, may still want this feature for debugging purposes.

*Use Script*: When checked, runs the specified script instead of the default action of "cd DIR; SOLVER". The script receives as arguments the remote directory and the remote solver, in that order. This was implemented to allow compatibility with VMS, Windows NT, and other non-UNIX systems. For a description of remote solvers under Windows NT, please see the Adobe Acrobat document included with the distribution package.

*Convert to UNIX*: Converts ASCII linefeeds to from DOS to UNIX before sending input files to remote machine. Some machines are picky about this, so if you get strange errors, you should try this option. This is a workaround for FTPd's that do not support ASCII mode file transfer.

*Use ASCII FTP*: If your FTPd supports ASCII mode, you may wish to use this option. It will prevent some read errors and allow you to view your input and output files more easily. You should use an FTP client such as WS_FTP to check whether your FTPd supports ASCII mode file transfer.

### 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 $y =$

0.5*W*, at a recess of zero, including pitch and roll but not crown, camber, or twist. This may not be a physical point on the slider, in the case of a TPC slider for instance. To track 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 frame are needed for the Fly Height solution. The solution will balance air and contact pressure against the *Load* in the z-direction and *PTorque* and *RTorque* in the pitch and roll directions, respectively, 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.

*PTorque*: Static pitch torque in µN-m.

*RTorque*: Static roll torque in µN-m.

### 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 four points may be specified in mm or normalized to slider length. The origin of the coordinate system is at the inner leading edge.

### 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. A smaller number may be needed in some cases.

*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).

### 3.1.3.1.6. Comments

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

## 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 current 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. Currently, a preview cannot be generated on a triangular mesh.

### 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. You will be warned if you have entered any invalid values.

### 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 area 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.

### 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, Symm, Mirror, Draw

*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.

*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.

### 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. After a rail is selected as the current rail, you can right click on a vertex to select it. The coordinates of the vertex will be highlighted in the *Rail Points* box. Once a vertex is selected, you can move it by double right clicking and dragging it 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 now color coded based on recess height. While this is not as pretty as the rainbow scheme in pervious versions, it makes visual analysis of the 2-d rail diagram easier.

#### 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.

### 3.1.3.3. 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.

### 3.1.3.4. Static Mode - Partial Contact Window

A partial contact model is implemented in the Quick4 (and Dyn4) solver. The model is statistical, and does not model actual impacts of the slider with asperities. Instead, it adds 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.

Partial Contact is not supported in Quick5 (triangular static solver). Partial contact parameters will be ignored by the Quick5 solver.

*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.

## 3.1.3.5. Grid Window

The CML Air Bearing Design Program now supports two different types of mesh: rectangular and triangular. These correspond to the Quick4 and Quick5 solvers, respectively. Both solvers support a pressure gradient based adaptive mesh to optimize mesh density distribution. Several grid control options exists in each case. The triangular grid option is not available for dynamic simulations.

## 3.1.3.5.1. Rectangular Computation Grid

*Initial Grid*: The user may select *Geometric Series* or *Existing Grid*. *Geometric Series* indicates that the user will specify an initial grid using the *Geometric Series* input parameters. *Existing Grid* indicates that the user will use the `x.dat` and `y.dat` files in the current directory, possible generated in a previous simulation.

*Symm. in Width*: Toggles the symmetry of the Geometric Series input parameters. If this option is on, the user need only 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.

*Adaptive Grid*: Indicates the grid will adapt based on the pressure 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.

*Fixed Grid*: Indicates the initial grid, as defined by *Geometric Series* or *Existing Grid*, will be used throughout the entire simulation.

## 3.1.3.5.2. Rectangular Adaptive Grid

*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:

$$\text{MaxWeight}(X) = \text{MAX}(\text{Gradient}(X,Y)), \text{ for all } Y$$
$$\text{AvgWeight}(X) = \Sigma\, \text{Gradient}(X,Y)/n, \text{ for all } Y$$

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 minumum 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.

*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).

#### 3.1.3.5.4. Triangular Grid Window

The Triangular Grid Window allows for modifications of the triangular grid input parameters in the Quick5 solver. You may preview the mesh by pressing the *Calculate* button. This mesh may differ slightly from the mesh generated by the solver due to floating point error. This will not affect the simulation results.

**First Level Mesh**:

*Maximum Triangles*: Specifies the approximate maximum number of triangles allowed for this mesh level. This parameter is an upper limit. Frequently, the number of triangles used in the mesh will be smaller than the maximum. Do not be alarmed if the number of triangles generated exceeds this value slightly. This is normal.

*Recess Region Size*: Specifies the maximum length of the longest side of triangles in the recess region, normalized by slider length.

*Bearing Surface Size*: Specifies the maximum length of the longest side of triangles on the rail surfaces, normalized by slider length.

*Aspect Ratio*: Specifies the maximum aspect ratio of triangles for this mesh level. Aspect ratio of a triangle is defined as: (2 x longest side length) / shortest side length

**Second Level Mesh**:

*Maximum Triangles*: See *First Level Mesh*.

*Recess Region Size*: See *First Level Mesh*.

*Bearing Surface Size*: See *First Level Mesh*.

*Aspect Ratio*: See *First Level Mesh*.

*Allowable Recess Diff*: Maximum allowable difference in recess height between two vertices of a single triangle (normalized by base recess). For instance, an *Allowable Recess Diff* of 0.2 requires that rails at the nominal recess plane with wall profiles that extend to the base recess will have at least 5 triangles in the wall region. Often, your geometry will contain walls with different total recesses. It is recommended that you specify an *Allowable Recess Diff* value to accomodate the steepest wall in order to accurately capture the geometry.

*Smallest Size*: Specifies the minimum length of the longest side of triangles for this mesh level, normalized by slider length.

**Third Level Mesh**:

*Maximum Triangles*: See *First Level Mesh*.

*Recess Region Size*: See *First Level Mesh*.

*Bearing Surface Size*: See *First Level Mesh*.

*Aspect Ratio*: See *First Level Mesh*.

*Allowable Recess Diff*: See *First Level Mesh*.

*Smallest Size*: See *Second Level Mesh*.

**Adaptive Mesh**:

*Maximum Triangles*: See *First Level Mesh*.

*Allowable Pressure Diff*: Maximum allowable pressure difference between two vertices of a single triangle, normalized by average pressure difference over the entire grid.

*Smallest Size*: See *Second Level Mesh*.

#### 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 trailing edge to the inner leading edge. This conforms with the IDEMA standard. Quick5 does not support multiple radius/skew pairs.

*RPMS*: The user must specify at least one *RPM* value. For Quick4, multiple values are specified, the simulation will be run at each value. Quick5 does not support multiple RPM values.

*Altitudes*: The user may specify one or more *Altitude* values. If no value is given, the simulation will use the *Air* parameters instead. If *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*, *PTorque*, *RTorque*, 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.

Quick5 does not support sensitivy cases.

*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 raio 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)

Quick5 does not support the stiffness calculation.

*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. Even then, only the pressure data for basic cases across the disk will be saved. For Quick4, three files will be saved for each radial position: `press`*`xx`*`.dat`, `cprss`*`xx`*`.dat`, and `mflow`*`xx`*`.dat`. In addition to these files, one `geom.dat` file will be generated. For Quick5, one file will be saved containing grid, pressure and geometry information: `xyph.dat`.

## 3.1.4. Static Mode - Postprocessor Menu

The static post-processor supports visualization of all output data from the Quick4 and Quick5 solvers. New features in version 5 of the interface include *Zoom* and *Save As Bitmap*. 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.

*Copy All Results*: Copies the results to the Windows clipboard in a tab delimited format. This may be 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` or `xyph.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.

*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 `press`*`xx`*`.dat` or `xyph.dat`, over the final grid, given by `x.dat` and `y.dat` or `xyph.dat`. This plot is available only if the *Save Pressure* option is enabled. The scale displayed is normalized to ambient pressure: $(p - p_a)/p_a$. 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 `cprss`*`xx`*`.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: $(p - p_a)/p_a$. Double click and drag to rotate the diagram. Right click to display a menu of options. Contact pressure is not calculated in Quick5.

*Mass Flow*: Displays the final mass flow for a particular case, given by `mflow`*`xx`*`.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. Mass flow is not calculated in Quick5.

### 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

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 catergories 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.

*Crown*: See Static Mode Slider Geometry.

*Camber*: See Static Mode Slider Geometry.

*Twist*: See Static Mode Slider Geometry.

*Taper Length*: See Static Mode Slider Geometry.

*Taper Angle*: See Static Mode Slider Geometry.

*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).

*Mass*: Mass of the slider in grams.

#### 3.2.3.1.2. 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.

*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.

*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.

## 3.2.3.1.3. 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).

## 3.2.3.1.4. Convergence

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

## 3.2.3.1.5. Time

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

*Step*: The time in ms alloted for each base step of the simulation. A smaller value will increase the computation time and accuracy of a simulation.

*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 *Current* 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). Triangular grid is not available in Dynamic Mode.

## 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 catergories of input parameters: Suspension Dynamics, Actuator, Suspension Loading, Track Seeking Motion, Disk Velocity Profile, and Disk Flutter.

### 3.2.3.6.1. Suspension Dynamics

The user has the option of integrating suspension dynamics by modal analysis or using stiffness and damping coefficients. If the former is chosen, the user must supply additional files (see below).

*Modal Analysis*: As mentioned above, the user may integrate suspension dynamics by modal analysis. When this option is chosen, the user must supply the files `fmodes.ext` and `fmesh.inp`. `fmodes.ext` contains the natural frequencies and modal shapes. `fmesh.inp` is the ABAQUS input file. Examples of these files are included with the distribution. The user must also supply additional parameters (*# of Mode Shapes*, *Center Node #*, *Alpha*, and *Beta*) as described below. These files should be placed in the document directory with the other input files. If they are not present at run time, the user will be prompted to import them.

*# of Mode Shapes*: Number of truncated suspension mode shapes.

*Center Node #*: Node number for the slider's gravity center defined in the ABAQUS input file.

*Alpha*: Mass proportional suspension damping.

*Beta*: Stiffness proportional suspension damping.

*fmodes*: Specifies the `fmodes.ext` file to be used for the simulation.

*fmesh*: Specifies the `fmesh.inp` file to be used for the simulation.

*Stiff. and Damp.*: The user may opt to integrate suspension dynamics using stiffness and damping coefficients as opposed to modal analysis. 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.

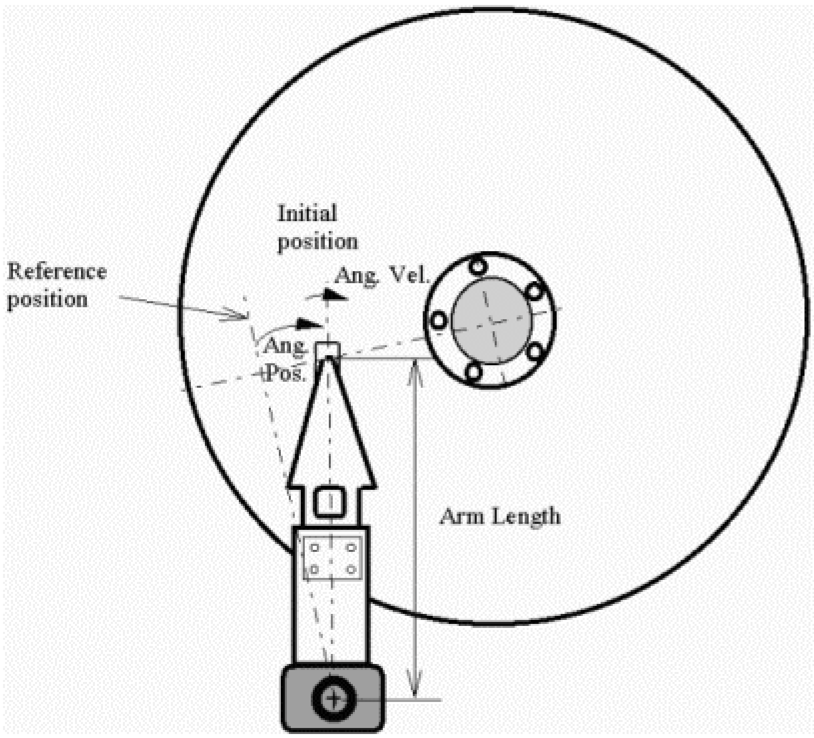
### 3.2.3.6.2. Actuator

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

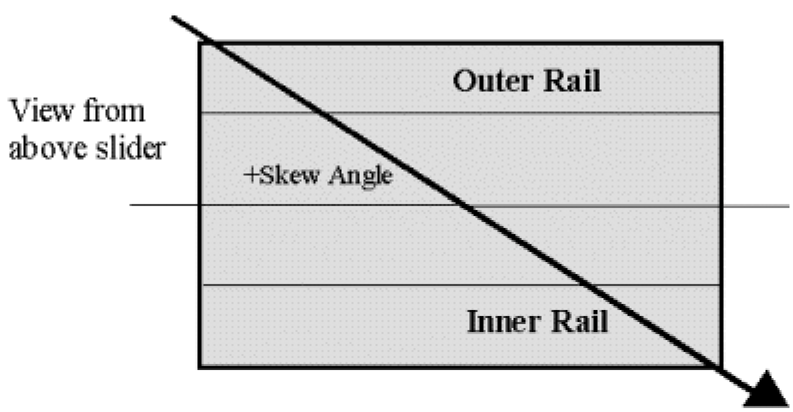
*Arm Length*: Length of the actuator arm in mm.

*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.




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




## 3.2.3.6.3. Suspension Loading

Note the change in *Pitch Offset*. This specification conforms to the standard set by quick413, in which a *Pitch* and *Roll Offset* of zero means that the load point is at the geometric center.

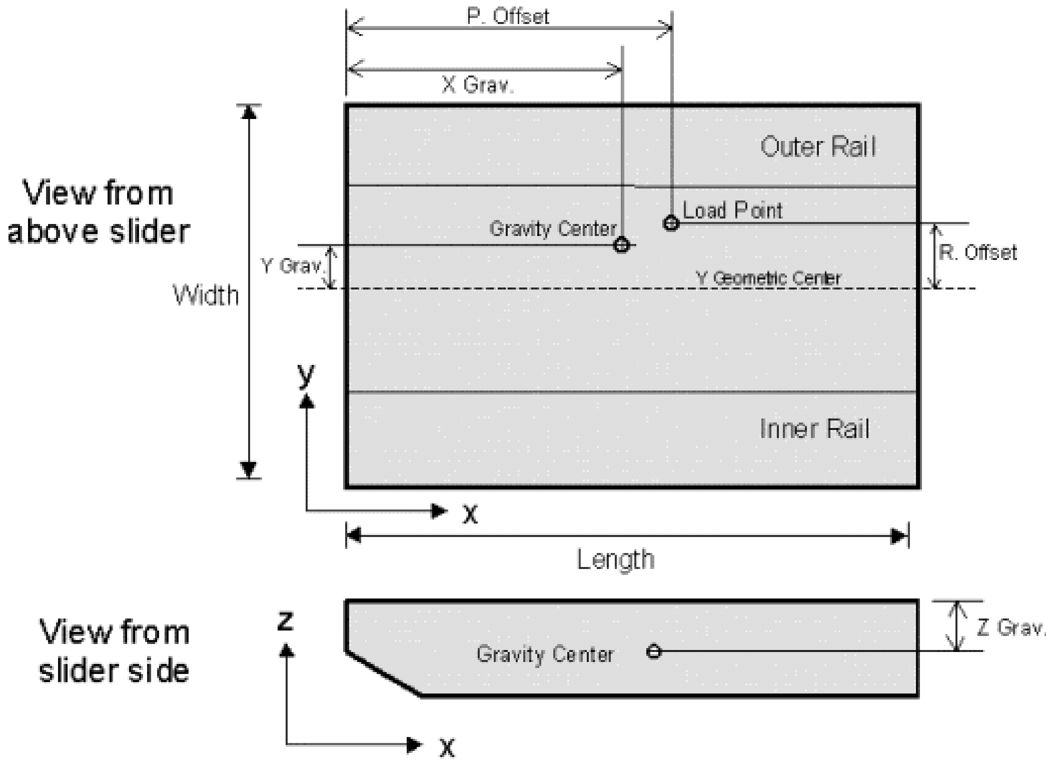
*Load*: Suspension load in grams.

*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.

*Pitch MOI*: Slider pitch moment of inertia in kg-m².

*Roll MOI*: Slider roll moment of intertia in kg-m².




## 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/sec². A linear functional relationship is assumed to exist between adjacent data points. Up to 40 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.

## 3.2.3.7. 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:

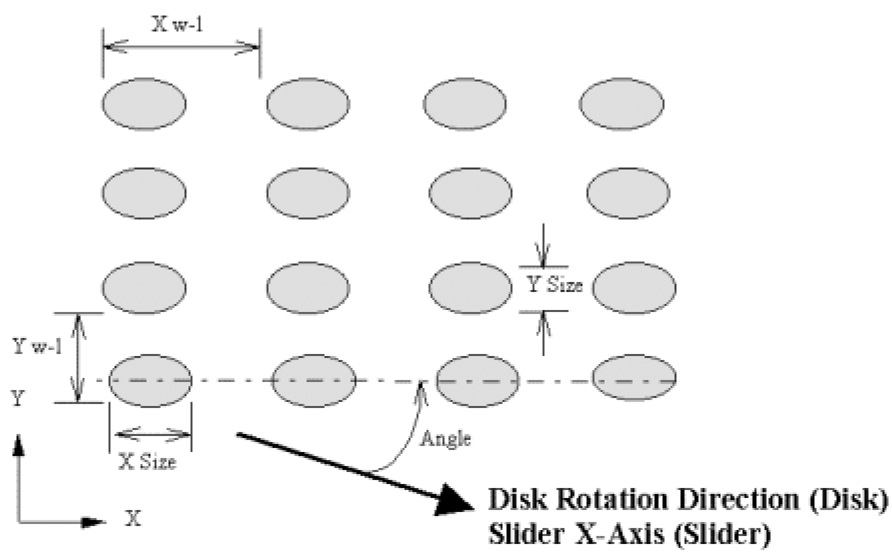
Radius: 0 0.000001
Height: 0.0000005 0

*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 wavelenth 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).

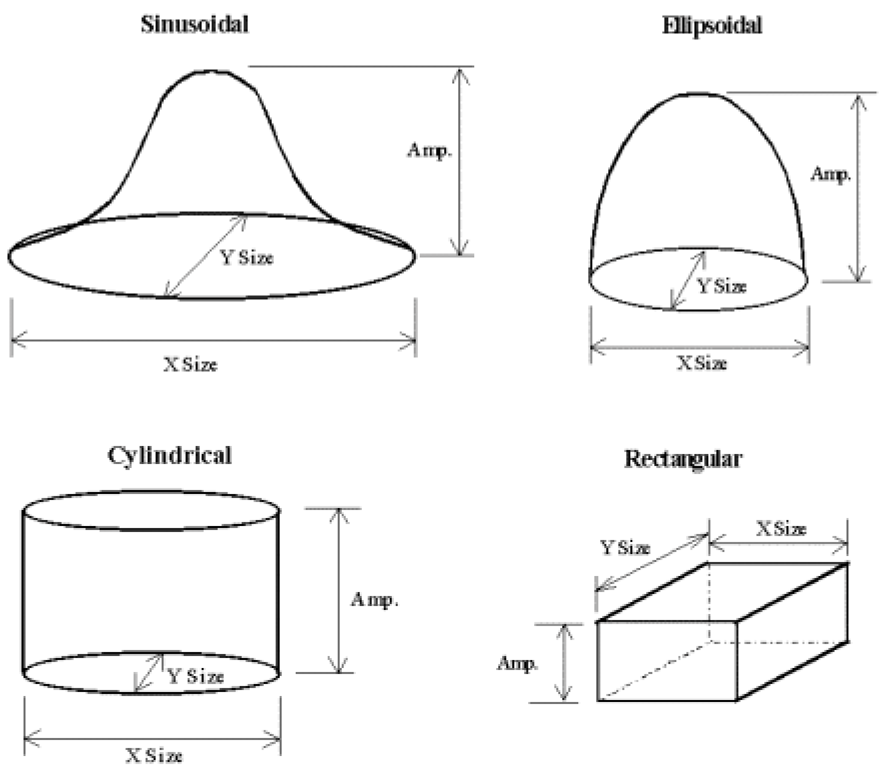



## 3.2.3.7.1.2. Disk and Slider Asperities

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*.




*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 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

The user may specify a point by point disk track profile by entering the exact coordinates into the interface. Points should be entered as x,y coordinates in the input box, with one pair per line, separated by spaces. The user should push the *Update* button after points have been entered or modified.

For convenience, the user may also place a file named `wave.def` in the case directory. This file should be whitespace delimited with two entries per line, corresponding to the disk velocity direction and height of the disk surface.

*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.

*View/Edit Points*: Toggles between a view of the profile and the edit box for profile points.

## 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. New features in version 5 include *Zoom* and *Save As Bimap*.

*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 under DOS or UNIX 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. These files are described in this section.

The triangular static solver (Quick5) uses three main input files: rail.dat, run.dat, and trigrid.dat.

### 4.1.1. rail.dat

Sample `rail.dat:`

```
CML VERSION 4.018  RAIL.DAT
REPORT BUGS TO: info@cml.me.berkeley.edu
2.050E-03       1.600E-03        4.300E-04
 4               2
 4               1
0.000E00        5.000E-05        1
2.050E-03       5.000E-05        1
2.050E-03       4.000E-04        1
0.000E00        4.000E-04        1
1.000E-6
 4               1
0.000E00        1.550E-03        1
2.050E-03       1.550E-03        1
2.050E-03       1.200E-03        1
0.000E00        1.200E-03        1
1.000E-6
 4               1
0.000E00        1.500E-04        2
2.050E-03       1.500E-04        2
```

```
2.050E-03        3.500E-04        2
0.000E00         3.500E-04        2
0.000E00
 4                1
0.000E00         1.500E-03        2
2.050E-03        1.500E-03        2
2.050E-03        1.300E-03        2
0.000E00         1.300E-03        2
0.000E00
 4 5
 0.00E+00  6.00E-06  1.20E-05  2.00E-05
 1.00E-06  3.00E-06  3.60E-06  4.00E-06
 -1.00E-05  -5.00E-06  0.00E+00   5.00E-06  1.00E-05
 0.00E+00  1.00E-07  3.00E-07  8.50E-07  1.00E-06
2.000E-04        1.000E-02        4.000E-06
2.00E-08         1.00E-08         0.00E+00
 2.025E-03  1.950E-03  2.025E-03  1.950E-03
 2.500E-04  2.500E-04  1.350E-03  1.350E-03
```

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 now in meters.

## 4.1.2. run.dat

Sample `run.dat:`

```
CML VERSION 4.019  RUN.DAT
REPORT BUGS TO: info@cml.me.berkeley.edu
***************Solution Control***************
istiff isolv   ioldg   iadpt   isave
 1      1       0       1       1
***************Initial Attitude***************
hm(m)          pitch(rad)      roll(rad)
5.0000E-08     1.0000E-04      0.0000E+00
***************Runs***************************
irad           irpm            ialt
 2              2               2
radii(m)
```

```
   1.5000E-02   2.3000E-02
skews(deg)
   -3.0000E+00   8.0000E+00
RPMs
   3.6000E+03   5.4000E+03
altitudes(m)
   0.0000E+00   2.0000E+03
***************Air Parameters*****************
p0(pa)         al(m)            vis(nsm-2)
1.0135E+05     6.3500E-08      1.8060E-05
***************Load Parameters****************
f0(kg)         xf0(m)           yf0(m)
3.500E-03      2.5000E-05      2.5000E-05
xfs(μNM)       yfs(μNM)         emax
5.0000E-01     5.0000E-01      1.0000E-03
***************Grid Control*******************
nx      ny
 67     67
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         decay           ipmax
 40             40              0
***************Reynolds Equation**************
ischeme        imodel          akmax
 2              3              1.0000E-07
***************Partial Contact****************
icmodel        stdasp(m)       dnsasp(m-2)
 1             6.0000E-09      1.0000E+12
rdsasp(m)      eyoung(pa)      yldstr(pa)
1.0000E-08     1.0000E+10      1.0000E+12
frcoe          pratio
 0.3            0.3
***************Sensitivities******************
crowninc(m)    camberinc(m)    twistinc(m)
1.0000E-08     1.0000E-08      1.0000E-08
tlnginc(m)     tanginc(rad)    loadinc(kg)
1.0000E-05     1.0000E-03      3.0000E-04
ptrquinc(uNM)  rtrquinc(uNM)   recessinc(m)
5.0000E-01     5.0000E-01      5.0000E-01
iwscale
 1
***************Comments***********************
" This is a test case"
```

The following are the variables in the `run.dat` file along with their *counterparts* in the Air Bearing Design Program for PC. 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.

`run.dat` *Air Bearing Design Program for PC* [`run.dat` 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/m$^2$].
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/m$^2$].
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.*

### 4.1.3. trigrid.dat

The values in this input file correspond with the input boxes in the *Triangular Grid Parameters* dialog box.

Sample `trigrid.dat`:

```
CML Version 5.000  trigrid.dat
REPORT BUGS TO INFO@CML.ME.BERKELEY.EDU
****** Grid Control Parameters of First Level Mesh  ******
Maximum Number of Triangles:
2000
Recess Region Size     Bearing Surface Size   Triangle Aspect Ratio
0.100000               0.050000               2.500000
****** Grid Control Parameters of Second Level Mesh ******
Maximum Number of Triangles:
12000
Recess Region Size     Bearing Surface Size   Triangle Aspect Ratio
0.100000               0.020000               2.500000
Allowable Recess Diff.    Smallest size
0.330000                  0.015000
****** Grid Control Parameters of Third Level Mesh  ******
Maximum Number of Triangles:
25000
Recess Region Size     Bearing Surface Size   Triangle Aspect Ratio
0.050000               0.013000               4.500000
Allowable Recess Diff.    Smallest size
0.330000                  0.001000
******  Grid Control Parameters of Mesh Adaptation  ******
Maximum Number of Triangles:
36000
Allowable Recess Diff.    Smallest size
1.500000                  0.003000
*** Rails to be Exempt from the Conforming Requirment ***
Number of Exempt Rails:
2
Exempt Rail Indices:
1 3
```

## 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.

If the *Save Pressure* (`isave`) option is on, pressure files are generated. The files `pressxx.dat`, `cprssxx.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` 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 on or the *Geometry Only* option is selected. This file contains the slider geometry over the mesh defined by `x.dat` and `y.dat`.

The triangular grid static solver produces two main output files: `result.dat` and `xyph.dat`. `xyph.dat` contains a complete map of the grid structure, final air bearing pressure, and slider geometry. `result.dat` is identical in structure to the rectangular static solver.

## 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`:

```
***********************Problem Definition*************************
xl            yl            xg            yg            zg            halt
2.05e-3       0.7805        0.5           0.0           0.105         0.0
f0            xf0           yf0           amz           aip           air
3.5e-3        0.5           0.0           6.0e-6        2.18e-12      1.36e-12
rpm           dt            tf            ra            rif           rof
5400.0        1.0e-6        0.001         23.0e-3       10.0e-3       30.0e-3
***************************Suspension***************************
iact          xact          dact          vact          ske
1             38.0e-3       0.0           0.0           -9.1
isusp         nmodes ncg           alfa          beta
0             10            2149          60.0          1.0e-5
skz           skp           skr           scz           scp           scr
18.0          1.146e-2      1.432e-2       0.002        1.58e-6       1.4e-6
**********************Initial Flying Condition*********************
hm            hp            hr            vz            vp            vr
2.9e-7        165.0e-6      158.0e-9      0.0           0.0           0.0
*************************Solution Control************************
iqpo          akmax         emax          idisc
5             1.0e-7        1.0e-4        1
**************************Grid Control***************************
iadpt         isymmetry     ioldgrid      nx            ny            nsx           nsy
1             0             0             98            98            1             1
xnt(i), i=2,nsx
0.0
nxt(i), i=2,nsx
0
dxr(i), i=1,nsx
1.0
ynt(i), i=2, nsy
0.0
nyt(i), i=2,nsy
0
dyr(i), i=1,nsy
1.0
difmax        decay         ipmax
40.0          40.0          0
*******************Point by Point Disk Track Profile*****************
ims           nfx           dinit
0             1009          3.2
************Numerical Generation of Disk Surface Topography**********
nwave         nzone         nasper
0             0             2
iwtype        wamp          wang   wthx   wthy   wpdx  wpdy  wrs  wre
zr1           zh1           zr2         zh2        zr3         zh3
iatype     aamp      aang     alocx       alocy      asizx      asizy
3          10.e-9    0.0      0.500e-3    0.0e-3     0.50e-4    2.0e-3
4          1.0       0.0      1.000e-3    0.0e-3     0.50e-4    0.5e-4
0.08e-03   0.11e-03  0.3e-03
*************Numerical Generation of Slider Surface Topography*********
nswave nsasper
0             0
```

```
iswtype       swamp         swang         swthx         swthy        swpdx   swpdy
isatype       saamp         saang         salocx        salocy       sasizx  sasizy
************************Track Seeking Motion**********************
nap
0
tac           aac
*****************Time-Dependent Disk Velocity Profile****************
nvp
0
tvp           vtd
**********************Sinusoidal Disk Flutter************************
iflut         tsft          teft          fqft          amft
0             0             0.003         100000.0      10.0e-9
*************************Asperity Contact*************************
icmod         ey            ydst          pratio        frcoe
2             9.7135e+10    1.0e+10       0.3           0.3
ncz
1
sikm          ceta          rasper        rcts          rcte          glidh
3.0e-9        2.0e+12       10.0e-9       10.0e-3       30.0e-3       80.0e-9
*************************End of Input Data*************************
```

The following are the variables in the `dynamics.def` file along with their *counterparts* in the Air Bearing Design Program for PC. The units used in the `dynamics.def` file are in [brackets].

| `dynamics.def` | *Air Bearing Design Program for PC* [`dynamics.def` units] |
|---|---|
| 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× m²]. |
| rpm: | *RPM*. It is initial rpm if the disk speed is time-dependent (nvp > 0). |
| dt, tf: | Time *Step* and *Total*, respectively [s]. |
| ra: | *Radial Pos.* [m]. It is initial radial position for the case of the off-track motion. |
| rif, 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*. |
| 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. Maximum nfx is 1200. 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].
zr*n*, zh*n*: *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/s²].
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/m²].
rasper: *Radius of Curv.* [m].
rcts, rcte: *Contact Zone Start Radius* and *End Radius*, respectively [m].
glidh: *Glide Height* [m].

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`, w*n*`.dat`, a*n*`.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 contains two columns of data: position (meters) in column 1 and height (meters) in column 2. To integrate the suspension dynamics by modal analysis, you need to have two files `fmodes.ext` and `fmesh.inp`. The file `fmodes.ext` stores the natural frequencies and modal shapes. The file `fmesh.inp` is the input file of the commercial finite element code ABAQUS. For fatype=4, the input files to define the cross-sectional profiles for the asperities, `a1.dat`, `a2.dat`, etc. are needed. For fwtype=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 ($m/s^2$) 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.

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) ($N\times m$) 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.

# Index

## 1



## 2



## 3



## 4



## A



## B

## U



## V



## W



## X



## Y