CML Air Bearing Design Program User's Manual

Sha Lu and David B. Bogy

Computer Mechanics Laboratory
Department of Mechanical Engineering
University of California at Berkeley

March 1995

1. INTRODUCTION

This manual describes the air bearing computer design code developed in the Computer Mechanics Laboratory at the University of California at Berkeley.

The current code is fully integrated with MATLAB, which performs the pre and post processing for the simulation. The code solves the slider air bearing pressure distributions for given flying attitude or suspension preload. The latter case is called the inverse problem, and there the steady state flying attitude is found through a Quasi-Newton search procedure.

With the rapid decrease of slider flying height in magnetic disk drive technology in the effort to increase the recording density, air bearing surface designs are becoming more complicated. Sub-ambient pressure air bearing designs are gaining popularity, and they often contain rails with complex shapes. This calls for a robust design tool.

The program described here is intended to facilitate the design of shaped rail sliders. The rail shapes are defined by piece-wise linear boundaries. A rail can be a step with a given recess depth, or it can be defined as a 'ramp', i.e., a flat plane having an arbitrary orientation. Each rail may have a different recess depth. Normally, a zero recess is assigned to the main air bearing surface.

A multi grid method is implemented, dramatically shortening the run time. This method is optimally efficient in the sense that the time for convergence increases only linearly with the number of unknowns, so its superiority is strongly realized when the number of unknowns is large. An adaptive grid method is also implemented in the program, which can adjust the grid distribution according to the pressure gradient. This usually results in better usage of the available grid points.

2. MATLAB INTERFACE

The pre and post processing is performed through the Matlab interface. To initiate a session, Matlab must be started by entering the command 'Matlab'. Then type 'steady' in the Matlab command window. A greeting will appear on the screen along with a menu bar at the top (Fig. 1). The menu items are labeled *PreProcessor*, *Run*, *PostProcessor*, and *GraphicsOptions*. In the MS-Windows environment, these items are preceded by a few other default menu items(File, Edit, Windows, Help). The user often needs to supply or modify the design parameters through editable texts. To change the editable texts, click the left mouse button once to select and modify, or double click(the box turns black) to overwrite the old values.

2.1 Preprocessor

A sub-menu will appear after clicking on *PreProcessor*, with the choices *SteadyDefinition*, *RailCreation*, *Loadcase*, *Savecase* and *Exit*.

2.1.1 Steady definition

The prompt for a set of parameters will appear in the graphics window after the *SteadyDefinition* is chosen from the *PreProcessor* sub-menu. The parameters used to define a static air bearing problem are entered through this window (Fig. 2).

Title: enter a name for the case here. It should not contain any dots (.).

Units: choose from *SI*, *British* or *custom*. This will affect the units of most of the parameters. If custom unit is selected, you are prompted to enter the unit conversion factor to millimeters.

Slider length and width: slider dimensions in units specified by *units* above. Default values are 2 mm x 1.6 mm.

Recess: recess depth for all areas not covered by any of the rails (defined in the *RailCreation* window chosen from the *PreProcessor* sub-menu).

Crown, Camber and Twist: enter here the values of global crown , camber and twist in nanometers. These are the second order surface topography components superimposed on each other over the whole slider. Positive crown and camber values represent convex parabolae in the length and width directions, respectively. With a positive twist, the inner leading edge and outer trailing edge are recessed (larger spacing) while the outer leading edge and inner trailing edge are raised (smaller spacing).

Radial position: the distance from the geometrical center of the slider to the disk center.

Revolutions per minute: disk rotation speed.

Skew: skew angle in degrees at the geometrical center of the slider. For positive skew, the flow comes from the inner leading edge towards the outer trailing edge.

Ambient pressure: ambient pressure in Pascals, used as the boundary condition around the entire slider.

Mean free path: mean free path of the air molecules in meters.

Viscosity: viscosity of air in NS/M^2.

Maximum residual: residual of the discretized Reynolds equation normalized by the main term in the equation, used as the convergence criterion. The default value is small enough for most cases.

Scheme: different schemes have been implemented to treat the convective term in the Reynolds equation; _Up-wind_, _Hybrid_, _Power-law_ and _QUICK_. Considering accuracy, stability and convergence characteristics, the Hybrid or Power-law scheme is recommended.

Model: three different correction models to the Reynolds equation have been implemented to account for the rarefaction effects. Among the _First-order Slip Model_, the _Second-order Slip Model_, and the _Fukui-Kaneko's Linearized Boltzmann Equation Model_, the last one seems to render the best results for low flying, high bearing number cases and it is the default choice.

Stiffness calculation/No stiffness calculation: when this switch is turned on, the program calculates and outputs the 3x3 stiffness matrix. It represents the ratio of the change in bearing force components (bearing load, pitch moment and roll moment) over the change in displacement components (height, pitch and roll). In the current setup, the results are stored in the file 'result.dat'.

Taper specification/No taper: there are two ways to define a taper. In the first way, the *taper length* and *taper angle* should be given when this switch is turn on. Accordingly, a wedge with the given taper angle starting at a taper length from the leading edge will be removed from the slider. If the taper can not be properly defined in this manner, it may be defined in the 'RailCreation' window as *ramp*s(see 2.1.2). In this case, the taper switch should still be on, with correct taper length and zero taper angle, so that when the adaptive grid option is selected, the program will resolve the taper end with enough grid points.

Initial flying attitude: the flying attitude used to start the calculation, including nominal trailing edge center height *Hm*, *pitch* and *roll*. For positive pitch, the spacing at the leading edge is larger than at the trailing edge, and the spacing at the outer rail is larger than at the inner rail for positive roll. For the inverse problem, the specified values are only initial guesses. But the convergence to the steady state flying attitude can be accelerated by using a better initial guess.

Solution for given attitude/Steady steady solution: when solution for given attitude is chosen, the program only calculates the pressure distribution for the given flying attitude. Otherwise, it computes the inverse solution. The code will search for the steady state flying attitude using the Quasi-Newton method. The user has to supply the *load* and the *load position offsets* in the pitch (x) and roll (y) directions with the origin at the slider center. The suspension static *pitch moment* and *roll moment* can also be supplied. Currently, a non-standard unit of gram-mm is used. Positive static pitch tends to increase the pitch angle and positive static roll tends to lift the outer rail. *Maximum error* is the normalized difference between the computed bearing load including its moments and the suspension preload as well as its static moments. It is used as a criterion for convergence in the inverse solution. *Points of interest* allows the user to specify up to four points for which the program outputs the fly heights. The *x and y coordinates* need to be specified.

Quasi-static take-off/No take-off simulation: when this option is turned on, the user can enter a *RPM vector*, with the numbers separated by blanks. The code then solves for different RPMs and stores the results in 'result.dat'.

Fixed position only/A group of radial positions/A matrix of radii and skews: the code only solves for one radius when the first option is chosen. The user needs to supply a vector of *radii* and a *skew vector* for both the second and third options. In the second option, the length of the radii vector and of the skew vector have to be the same, since there is one-to-one correspondence.

Initial grid: the initial grid can either be *created* or *imported*. When the latter is selected, the program reads the grid data from 'x.dat' and 'y.dat' files. These files are created when the program finishes a run. If the initial grid needs to be created, more parameters have to be entered. *x grid* and *y grid* are the number of grid points in the x (length) and y (width) directions. They are rounded by the program to the nearest number having the form (16n+1), dictated by the multi grid method in the solver. For example, 100 will be rounded to 97. The program uses piece-wise geometric progression to generate the grids. A uniform grid is generated by default. *x controls* are a set of points (separated by blanks) by which the slider length is cut into segments. For example, two points cut the length into three segments. In each segment, the successive grid size changes at a fixed ratio. *x indices* are the grid indices for the control points. The ratios of successive grid sizes in each segment are specified by *x ratios*. It is similar in the y direction, except that if the *symmetry in width* is chosen instead of *specify entire width*, the grid needs only to be specified on half the width.

Adaptive grid control: no matter how the initial grid is obtained, the user has the option to use *adaptive grid* or *fixed grid only*. When adaptive grid is chosen, the program will generate an adaptive grid according to various measures of the pressure gradient after an initial calculation. The user has some control over the method by which the adaptive grid is generated. In order to compute the grid density in one direction, e.g., x direction, the user can choose to use either the *maximum* or the *averaged pressure gradient* along all the y locations. The pressure gradient in some area may be very small(e.g., a fully relieved region), but some minimum grid concentration may be needed. The user can specify the ratio *max/min pressure gradient*(grid density). A smaller ratio generates a more uniform mesh. Also, the pressure gradient may change abruptly, but 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. The larger the *decay factor*, the more abruptly the grid density changes.

2.1.2 Rail creation

Choosing this option from the *'PreProcessor'* sub-menu will create a new window for drawing the rail shapes by use of the mouse (Fig. 3). If some old rail data already exist in Matlab, the user will first be prompted by the choices: *Erase old rails*, *Edit old rails* and *Add new rails*. The first option erases the old rail data, while the second option enters the editing mode. The last option keeps the old data and allows the user to add new rails.

The rectangular box in the window with a dashed square grid represents the slider. The grid lines in the box are reference lines. The units of the axes are given by *units* discussed in 2.1.1. In the lower left corner of the window, the current position of the mouse pointer is indicated.

Drawing the rails: the rail shapes are generated by piece-wise linear segments. The user can input a rail point at the mouse pointer by clicking the left mouse button. To drag the line only horizontally, type 'h' on the keyboard while holding the mouse still. Type 'v' to drag it only vertically. The right mouse button should be used to fix the last point of the rail. The rail boundaries are marked by lines of different colors, with small circles around the rail points. The rails and boundary points are labeled by letters in the same color as the lines. For some rail shapes, the rail label may fall outside its boundary. When the last rail is finished, click the right mouse button again. Two small editing windows then appear on the right side of the window (Fig. 3).

Editing the rail as a whole: the upper editing window is used to edit each rail. The user can either click on the letters 'Rail No.x' on the rail to make that rail current, or the user can click and change the *rail number* on the first line in the editing window. The next two lines display the *x* and *y* limits of the smallest rectangle containing the current rail to help the user identify the current rail. The push buttons that follow perform the editing functions. *Mirror* creates another rail which is a mirror image of the current rail with respect to the slider's center line. Clicking on *Delete* will delete the rail. To move the current rail to another location, click on *Move*, then click on any place in the slider and move the mouse, click again when the rail arrives at the desired location. The amount of displacement is shown in the lower left corner. A similar procedure can be used to *Copy* the current rail and put it in a new location. If one single rail is symmetrical with respect to the slider's center line, the user needs only to draw half of the rail, and use *Symmetry* to create the other half. Note that the first and last points in the half rail will be connected to their corresponding mirror images. The user should arrange these two points properly in the half rail to create the correct rail. *Rotate* can be used to change the orientation of the rail. The procedure is similar to that of moving the rail. Sometimes the display may be incorrect after editing, possibly due to Matlab graphics bugs. The problem can be corrected by going back to the top menu and returning to the *RailCreation* window again.

Next, the user can specify whether the rail is a *step* or a *ramp*. A step has a uniform *recess* depth. The number is positive for recess under the ABS. Arbitrary wall profiles can be specified around the entire rail. Here, the wall profile can include both the edge blend and the etch slope. The profile is approximated piece-wise linearly. The user can enter a series of *points* by specifying their distance to the nominal edge, separated by blanks. The coordinates may start from negative numbers (edge blend) and end with positive numbers (etch slope). Corresponding to each point, a *recess* value should be specified. The recess values should be separated by blanks. When the same point is repeated and given different recess values, a discontinuity can be included. If the number of points is less than 2, the wall profile is ignored by the program and a nominal discontinuity is assumed. A *ramp* is a plane having an arbitrary orientation. It can be used to define a taper region(another way to define a taper is described in 2.2.1). It is defined by specifying the *recess depths at the first three rail points*(they should not be on the same line). Again, positive recess is below the ABS. Wall profiles can not be specified for a ramp.

Editing the rail points: the lower window provides functions for editing the rail boundary points. Clicking on the point index letter on the rail will make that point current, or the user can click and change the *rail* and *point* indices on the first line of the editing window. The next two lines show the x and y coordinates of the current point. The user may enter different coordinates to change the current point. This may be useful when exact coordinates are desired, since the mouse input has limited resolution, or the user can choose *Change* and click at a new point on the slider to change the current rail point graphically to the new point. To insert points between the current rail point and the next one, select *Insert* and click on new points on the slider. Use the left button to insert points and the right button for the last point.. *Snap* moves the current rail point to its closest neighbor. This function is used when two points belonging to different rails are meant to be identical. Select *Delete* to delete the current rail point. *Pivot* works similarly to *Insert*, except that all the points inserted are projected to the arc beginning from the current rail point and pivoted at the user specified pivot point, whose coordinates are entered from the box appearing on the right. This provides a convenient way to define an arc in the rail boundary.

2.1.3 Load case

Existing cases can be loaded into Matlab by choosing *LoadCase* in the menu. The case files are in .mat format. The user is provided with a list of cases to choose from. The input data files STEADY.DEF, RAIL.DAT, MULTCASE.DAT and TOL.DAT are updated to the current case.

2.1.4 Save case

The *SaveCase* option saves the current case in .mat format. It also generates the input files STEADY.DEF, RAIL.DAT, MULTCASE.DAT and TOL.dat used by the simulator. The user will be prompted to enter the case name, which can be modified here.

2.2 Run

After the case is saved, the user can start the simulation by choosing this item and clicking on *Steady*. The letters 'Simulator Running ...' will appear on the screen. The user can monitor the progress through the output window(in X Windows, bring up the Matlab command window). When the program finishes, the results are loaded into Matlab and saved in the .mat format.

When running on a PC, the program is loaded into memory much faster in the DOS environment than in Windows, since other Windows programs take away part of the RAM. Here is the procedure to do this:

> After saving the case, quit Matlab and MS-Windows, go to the appropriate directory in DOS, start the simulator by the command QUICK301(executable file of the simulator). When the case is finished, start MS-Windows and Matlab, go to the appropriate directory, start the interface STEADY, choose *LoadCase* in the *Preprocessor*, load the saved case, go to the *PostProcessor* menu, choose *Loaddata*. The case will now be saved along with the result data and will be ready for post processing.

2.3 PostProcessor

The functions provided in this menu can be used for post processing of the data generated by the simulator. It has limited capability so far.

2.3.1 Load data

This is only used when the simulator is run outside the interface(see 2.2).

2.3.2 View grid

The final grid used in the computation is displayed.

2.3.3 View rail

The rail areas are filled with color.

2.3.4 3-D pressure

The 3-D pressure contour is displayed with a color scale.

2.3.5 Settling history

The Quasi-Newton iteration history of various variables is displayed. The eight figures are the iteration histories of :

load err(normalized difference between bearing load and moments and the suspension load and static torques);

nominal trailing edge center flying height(on the flat reference plane with pitch and roll)(nm);

pitch angle(micro radian);

roll angle(larger outer rail gap for positive roll)(micro radian);

flying heights for four points of interest(coordinates start from the inner leading edge, normalized with slider's length)(nm);

2.3.6 Print

This option will invoke the Matlab print command to print out the contents of the current figure window, equivalent to typing ‘print’ in the Matlab command window.

2.4 Graphics options

This menu provides some graphics options mainly for the _3D pressure contour_.

2.4.1 Color control

Color map: the user can choose different colormaps to plot the pressure. By default, Matlab colormap 'jet' is used, which ranges from blue for the low pressure region to red in the high pressure region.

Shading: _flat shading_ fills the surface with piece-wise uniform colors, while the _interpolated shading_ smoothes the color between grid lines by linear interpolation. _Faceted shading_ adds black grid lines to the flat shading. The _no shading_ option draws only the color coded grid lines, which is the default choice.

Brighten: A slider will appear at the bottom of the figure window. The user can adjust the brightness of the picture using the slider.

2.4.2 View

When this option is selected, sliders will appear at the lower left corner of the figure window. The viewing angle can be adjusted through the sliders or by just typing in the numbers. Two angles can be adjusted. The viewer moves around the object counter-clockwise in the horizontal plane with the increase of <u>*azimuthal angle*</u>. The viewer looks straight downward with 90 degrees <u>*elevation angle*</u>, horizontally at 0 degrees, and straight upward with negative 90 degrees.

3. TUTORIAL

In this tutorial, the usage of the interface is illustrated through an example.

3.1 Some rules for creating the rails

The current implementation enables the inclusion of very complex rails and geometric features. In order to simplify the rail creation process and ensure consistency, it is important to follow the basic rules outlined below.

1. A 'rail' is classified into two types: <u>*step*</u> and <u>*ramp*</u>. A <u>*step*</u> is parallel to the reference surface, while a <u>*ramp*</u> is a plane with arbitrary spatial orientation. In fact, a <u>*ramp*</u> can also be used to define a <u>*step*</u>.
2. Any area that is not defined as a 'rail' assumes the recess depth defined in the <u>*recess*</u> box in the <u>*SteadyDefinition*</u> window.
3. Any given area on the slider can be occupied by more than one rail. The only rail that is effective in the given area is the one with the highest rail index (the last one created). This rule can be utilized to simplify the creation of certain rails, e.g., the TPC rails.
4. In the current version, each rail of the <u>*step*</u> type can have an arbitrary wall profile. A <u>*ramp*</u> does not have a wall profile. Also, for any given point, if it is covered by some <u>*ramp*</u>, then it does not belong to any wall profile region. If a wall profile extends beneath some other surface or another wall profile, that part of the wall profile is ignored.

3.2 An example

To start Matlab, type 'matlab' in the command shell.

Type 'steady' in the Matlab command window. Figure 1 shows the interface window as it appears in MS-Windows environment.

Choose the *PreProcessor* on the top menu bar, and select *Steady Definition*. A table of parameters appears(see Fig. 2). The user may change the default values by double clicking and typing new numbers. Some parameters shown in Fig. 2 have already been modified from the default values.

Next, choose *Rail Creation* in the *PreProcessor* menu to start drawing the rails. The rectangular box represents the slider. Click the left button of the mouse to enter rail boundary points. Rail shapes are approximated by piece wise linear segments. Click the right mouse button when entering the last point in a rail. When all the rails are drawn, click the right mouse button again to finish the drawing mode and enter the editing mode. Figure 3 shows two rails. Note that Rail No. 2 has only two points at this stage. Don't be concerned if mistakes are made or the locations of the points are not accurate. Everything can be corrected in the editing mode.

In the editing mode, two small editing windows appear on the right side of the slider box (see Fig. 3). The top one is for editing the rail as a whole, and the bottom one is for editing individual points on the rail. To select a particular rail to edit, click on 'Rail No.x' in the middle of the rail(the rail index may be outside the rail for some concave rails), or simply type the rail index number on the top of the rail editing window. Similarly, to choose a point to edit, click on the point indices on rail boundaries or enter the indices on the top of the point editing window.

Now, the point coordinates for Rail No. 1 can be modified by using the point editing functions if necessary. In fact, the exact coordinates can be entered directly in the point editing window. The two points in Rail No. 2 are meant to be coincident with Points 1 and 2 of Rail No. 1. However, when they are entered from the mouse, this can not be done exactly. The way to correct it is to either enter the coordinates directly or use the *snap* editing function.

After all the coordinates are made exact, select Rail No. 2 and click on the *Symmetry* function of the rail editing window. The other half of Rail No. 2 is created (see Fig. 4). Then select Rail No. 1 and click on the *Mirror* function in the rail editing window. A mirrored rail is now generated (also see Fig. 4). The height information for the rails can be entered from the lower part of the rail editing window. The rails here are defined as *steps*. The first *recess* describes the global rail height relative to the reference surface. Usually, the nominal ABS is used as the reference surface. Therefore, Rail No. 1 is given zero recess. An arbitrary piece-wise linear wall profile can be entered using no more than ten points. If fewer than two points are given, a vertical wall is assumed. Here, a linear etch slope is given for Rail No. 1 (see Fig. 4) and Rail No. 3, extending from the ABS to the etched pocket (3 um deep as prescribed in *SteadyDefinition* window) with a width of 15 um. On the other hand, Rail No. 2 near the leading edge has 1 um recess relative to the ABS, thus its wall profile also starts from 1 um recess(see rail editing window in Fig. 3). The rails are now fully defined.

Now go back to *PreProcessor* and select *SaveCase*. The title is prompted on the screen and can be modified. Click on *OK* to save the case.

Choose *Run* on the menu bar and click on *New*. The program now starts to run. When the case finishes, the data are loaded into Matlab automatically and are ready for post processing. An alternative way to run the program on a PC is described in 2.2.

Go to *Post Processor* on the menu bar. Figure 5 shows the three dimensional pressure contour generated by the *3-D Pressure* option. The effect of the recess of Rail No. 2 on the leading edge pressure is evident in this plot. The final grid generated by the adaptive grid algorithm is shown in Fig. 6.

4. INPUT AND OUTPUT DATA FILES

When the case is saved in the Matlab interface, a few data files are generated which are ready to be used by the FORTRAN program.

4.1 Input file 'steady.def

The file 'steady.def' contains most of the parameters in the *SteadyDefinition* menu in the interface. Not all the parameters in this file are currently used by the program.

The following is a list of currently active parameters:

hm(m) : nominal initial trailing edge center flying height. Note that the reference point is on the nominal plane with pitch and roll, not including crown, twist or camber.

h0 : nominal leading edge center height, normalized with hm.

hs(rad) : roll, positive roll widens the outer rail gap.

xl(m) : slider length.

yl : slider width normalized with xl.

ske(deg) : skew angle, positive skew implies that the air flows from the inner to the outer rail.

ra(m) : radial position of the slider center.

rpm : revolutions per minute.

isolv: 0 = solve for fixed flying attitude, 1 = find steady state flying height.

f0 (kg) : suspension load

xf0 : load x-position, normalized with xl, starting from leading edge.

yf0 : load y-position, normalized with xl, starting from center towards outer edge.

xfs : static pitch moment in g-mm, positive value tends to increase the leading edge spacing.

yfs : static roll moment, positive value tends to increase the outer rail spacing.

istiff : 0 = no stiffness calculation; 1 = calculate stiffness matrix.

akmax : normalized residual of Reynolds equation, criterion of convergence of the solver.

emax : normalized difference between the current bearing load and the target suspension load, the criterion of convergence for the inverse solution.

p0(pa) : ambient pressure.

al(m) : mean free path of air.

iadpt : 1 = use adaptive grid; 0 = disable adaptive grid;

isymmetry: 0 = manually generate the grid over the whole slider width; 1 = generate only half of the grid, which is symmetrical in the slider width direction. This has no effect when the adaptive grid option is used.

ioldgrid: 0 = either use adaptive grid or manually generated grid; 1 = use the old grid locations in the files x.dat and y.dat.

nx, ny : grid size, must be in the form of (16n+1) because of multi grid method.

nsx, nsy : number of grid sections in x and y directions, respectively, for manually generated grids.

nest : multi grid level. nest = 4 is the highest level and should be used.

xnt(i) : i from 2 to nsx, coordinates for the end of each section in the x direction, normalized with xl.

nxt(i) : i from 2 to nsx, grid indices for the end of each section in the x direction.

dxr(i) : i from 1 to nsx, ratio of grid size over previous one for each section in the x direction.

ynt(i) : i from 2 to nsy, coordinates for the end of each section in the y direction, normalized with xl.

nyt(i) : i from 2 to nsy, grid indices for the end of each section in the y direction.

dyr(i) : i from 1 to nsy, ratio of grid size over previous one for each section in the y direction.

vis1 : viscosity of air.

idisc: different schemes for treating the convective term. 1=power-law; 2=central difference; 3=upwind; 4=hybrid; 5=central difference in the hybrid form; 6= QUICK; idisc=1 is recommended.

iqpo : slip flow models, 0=continuum model; 1=first order slip model; 2=second order slip model; 5=Fukui-Kaneko linearized Boltzman equation model. 5 is the recommended choice.

difmax : used in the adaptive grid, a larger number allows a larger grid density difference.

decay : used in the adaptive grid, a larger number has less smoothing effect, and the grid density depends more on the local pressure gradient and may change more abruptly.

ipmax : used in the adaptive grid, 0 = use averaged pressure gradient in each direction along the cross section grid locations; 1 = use the maximum gradient in each direction along the cross section grid locations.

4.2 Input file 'rail.dat'

The first line in 'rail.dat' indicates how many rails are defined and how many different recess heights they possess.

The data for each rail follow. First, the number of boundary points, the recess height index of the current rail and the number of points in the wall profile are given. The last number should be less than two if the rail is of the ramp type (the recess height index is 0). Next, the coordinates of the boundary points normalized with slider length are shown, with the origin at the inner leading edge. If the recess height index is 0, there is an additional line consisting of recess depths for the first three boundary points of the rail. Finally, if the wall profile has at least two points (otherwise a vertical wall is assumed), two more lines are used to describe it. The first line gives the coordinates of the points in terms of the normal distance to the nominal wall normalized by the slider length. Points with negative coordinates are inside the nominal rail boundary, and those with positive coordinates are outside the boundary. The second line contains the recess values (in meters) for the wall profile points.

After all the rails are defined, there is a line which gives the base recess depth (areas not defined as rails), and rail recess depths in index order. All recess values are in meters. The next line shows taper length (normalized with slider length) and taper depth in meters. If a virtual taper exists, but is defined using ‘ramps’ instead of by the 'taper specification' section in the 'SteadyDefinition' of the interface, the taper length should be set to the actual value, with zero taper depth at the front. The code will then try to resolve the taper end automatically when the adaptive grid is used.

The next line contains information on crown, twist and camber in meters. The crown is a longitudinal parabolic surface superimposed on the whole slider. A positive crown decreases the spacing between the slider and the disk. The camber is the same as the crown except that it is in the transverse direction. The twist is given in terms of the relative height of four corners to the center. A positive twist increases the separation between the slider and the disk at the inner leading edge and the out trailing edge, and decreases the separation at the outer leading and inner trailing edge.

The last two lines give the x and y normalized coordinates of four points on the slider, respectively. The program outputs the fly height at these four points.

4.3 Input file 'tol.dat'

This file contains the information about sensitivity parameters. It is currently inactive. The first parameter should be set to zero.

4.4 Input file 'multcase.dat'

This file can be used to run multiple cases for different skew, radius, RPM.

The first line contains five integers: itake, nrpms, imultcase, nrads, and nskew.

itake: 0, no quasi-static take-off simulation; 1, yes.

nrpms: number of different RPMs for the take-off (maximum 10).

imultcase: 0, fixed radius; 1, a group of cases with a different skew corresponding to each radius; 2, a matrix for a set of skews and a set of radii.

nrads: number of different radii(maximum 10).

nskew: number of different skews (maximum 10, = nrads if imultcase = 1).

Lines 2 - 4 contains a group of RPMs, radii and skews, respectively.

4.5 Output files

The pressure matrix is stored in the file 'p2.dat'. The files 'x.dat' and 'y.dat' contain the normalized x and y coordinates, respectively. The iteration history of the flying attitudes is stored in the file 'invs.dat'. The file 'result.dat' contains the final flying attitudes and the stiffness matrix.

APPENDIX

Installation Guide

There are two main directories on the diskette, which are for PCs and UNIX workstations respectively. The .m files and FORTRAN source codes are in the PC directory only, since they are the same for workstations. The FORTRAN source programs should be renamed '*.f' from '*.for' for the workstation version. Only the executable files are provided in the workstation directory. They may be either in DEC Alpha or IBM RS/6000 format depending on the user's specification. For other computer models, the user needs to re-compile the source programs. The user interface requires Matlab 4.1 or later version.

a. PC version

1. There should be at least 8 MB RAM available. The system should be running MS-Windows 3.0 or later.
2. Put .m files in a separate directory. Add its path to 'matlab\matlabrc.m'.
3. Put 'new4g.vmc' in the c:\ directory. Add the contents in 'autoexec.cml' to 'autoexec.bat'.
4. Put the other files in a separate directory and add the path in 'autoexec.bat'.
5. Set the MS-Windows swap space to 30 MB. Please refer to MS-Windows User's Manual.
6. Create or edit 'matlab\startup.m' if necessary. For example, adding the line 'cd c:\case' enables Matlab to go to directory 'c:\case' when it is started.

b. UNIX workstation version

1. Put .m files in the '~/matlab' sub-directory of the home directory and Matlab will search for these files automatically. '~ ' in this instance refers to the user's home directory.
2. Put the other files in a separate directory and set the path in '~/.login' to include this directory.