Numerical Simulation of Slider Air Bearings of Hard Disk Drives with Complex Shapes and Steep Walls: Part 1 Unstructured Adaptive Triangular Mesh Generation Techniques and Finite Volume Schemes

L. Wu and D. B. Bogy

Computer Mechanics Laboratory,

Department of Mechanical Engineering,

University of California, Berkeley, CA 94720

Abstract

Unstructured adaptive triangular mesh generation techniques and vertex based finite volume schemes that suit slider air bearing simulation of hard disk drives are constructed and implemented. Different refinement and adaptation techniques are used to generate several levels of good quality mesh over sliders with complex rail shapes. At each level, either one geometrical or one physical property of the problem is captured. The overall mesh generation procedure offers great flexibility and control over the quality and distribution of the generated mesh, which makes it superior to its much simpler structured counterpart. An explicit vertex based finite volume scheme is first constructed. The modified Reynolds equation is locally integrated over the control volumes that are taken to be the Voronoi polygons. Backward-Euler time differencing is used to discretize the unsteady term. Patankar's strategy is used to evaluate the numerical flux across each edge of the control volumes. A local time stepping technique is adopted to speed up the convergence rate. To further improve the convergence rate, the explicit scheme is extended to a fully implicit one, and the resulting simultaneous linear algebraic equations are solved iteratively by the Gauss-Seidel method. Unconditional stability of the scheme is achieved, and preliminary calculation shows rapid convergence compared to the explicit scheme.

Introduction

In today's hard disk drives, the read-write components attached to the sliders are separated from the disk surfaces by the air-bearing force generated by a thin air layer squeezed into the narrow space between the sliders and disk surfaces due to the high rotation speed of the disks. The slider's flying height and other geometric parameters like pitch, skew and roll angles, have a profound influence on the performance of the hard disk drives. These geometric parameters are determined by the balance of the forces and torque generated by the spring suspension and that of the air bearing. Simulation of the air bearing force accurately and efficiently becomes the key issue in the design of sliders.

To increase the storage capacity of the hard disk drives, the sliders must fly lower with rail shapes that are becoming increasingly complicated, imposing a great challenge to the numerical simulation. To meet an increasing need, unstructured triangular mesh generation techniques and numerical schemes that are suitable for the air bearing simulation of complicated shapes have been recently developed in CML.

Current sliders in hard disk drives are characterized by complicated rail shapes and highly recessed air bearing regions between the rails. The rails are joined to the fully recessed regions by abruptly changing wall profiles determined by the etch process. According to previous investigations, in addition to having enough mesh in high pressure gradient regions, an accurately specified wall profile is also necessary for getting accurate numerical results, which requires the clustering of very fine meshes in the narrow recess wall regions. When the rail boundary is curved or not aligned with the coordinate axes and a Cartesian rectangular mesh system is used, it is extremely difficult to achieve the above goals without refining the mesh in

other regions where a fine mesh is not needed. This normally causes the mesh number to become much too large for efficient numerical simulation, since the CPU time needed to get converged results increases very rapidly with the total grid number.

A properly chosen grid system and its corresponding generation technique are as important in the numerical simulations as the numerical scheme itself. A good grid generator must be efficient and convenient to use. At the same time, it must give the user enough flexibility and control such that reasonably distributed meshes with the desired quality can be generated for each specific problem.

Because of its geometric flexibility in constructing a quality mesh around complex configurations and relative convenience in incorporating an adaptive methodology, the triangular unstructured mesh has become popular in recent years in many applications. Two major techniques have been used to generate unstructured triangular meshes in the CFD field. One is the advancing front method (ref.1). In this method, new triangles are formed from the front advancing away from domain boundaries. The advantage of this approach lies in its robustness and relative ease of application. Compared with other triangulation procedures like Delaunay triangulation, this method suffers the shortcoming that it is less efficient and permits limited control over the quality of the generated triangles. The second one is the Delaunay method, as will be fully described later. Due to its efficiency and its ability to generate optimal connections of the existing node points, say maximizing the minimum angle of each triangle, the Delaunay method is used as a building block in our approach. Here we use two different techniques to generate the Delaunay triangulation. One is the Bowyer-Watson algorithm (see refs. 2,3), the other is Sloan's algorithm (see ref. 4). Traditionally, all node points are generated by a certain

means before they are connected into triangles using the above algorithms. Then one by one, they are inserted into the existing triangulation. But before the insertion, a search is needed to locate the triangle that encloses the new node. When the number of nodes is large, the efficiency of the above algorithms is greatly reduced due to the searching process. Rebay (ref. 5) proposed an incremental approach, in which the locations of new nodes are simultaneously decided by a local property of the previous triangulation as new connections are formed, instead of being decided before the triangulation. This eliminates the need to do the search. We adopt Rebay's incremental ideas in our approach.

In our simulation process a multi-grid technique is used to achieve fast convergence. As a result, we need several levels of meshes ranging from very fine to very coarse. We use three different grid refinement and adaptation techniques to generate these levels of grids. To ensure accuracy and efficiency of the following simulations, each of them will capture either one geometrical or one physical characteristic of the problem. At the coarsest level it's important that the boundaries of the rail be fully defined by triangle sides in the triangulation. Otherwise, the represented geometry can be easily distorted far away from the actual one, which may cause detrimental effects on the following multi-gird iteration process. If the geometry represented by the coarsest grids is quite different from that represented by the finer grids, the iteration on the coarsest mesh may converge to a quite different solution from that of the finer meshes. A conforming Delaunay refinement technique in Ruppert (ref. 6) is adopted to generate the conformed coarsest mesh. At the second level of grids, with more grids available, it's possible to specify the wall profile to some extent. The longest-side-bisection Delaunay refinement technique in Rivara *et.al.* (ref.7) is modified so that a high quality fine mesh can be clustered

toward the recess wall region based on geometrical considerations. In the air bearing field, the majority region with smoothly changing pressure is surrounded by small regions with extremely high-pressure gradients. Global grid refinement is definitely not a good choice, since enormous amounts of computer resources will be wasted in regions with smoothly changing pressure profiles where high grid resolution is not needed. In such a situation, the grid adaptation technique has proved to be an efficient way to improve the accuracy of simulation at a much lower cost than that of the global grid refinement strategy. An adaptive mesh refinement technique, which uses the undivided pressure difference as the sensor, is developed to refine a good quality mesh in the high-pressure gradient regions during the simulation process, which gives the finest grid.

In recent years extensive effort has been devoted to the simulation of slider air bearings of hard disk drives with complex rail shapes. Most numerical methods fall into one of three categories: finite difference methods, finite volume methods (also called control volume methods) and finite element methods. All of them have been used in air bearing simulations (ref. 8, 9, 10). The finite difference method is known for its efficiency if smooth and adequate meshes can be generated. When the geometry is complicated, generally a body-fitted mesh system that can smoothly follow the boundaries is generated by either solving a PDE (ref. 11) or simply using algebraic interpolation. A coordinate transformation is carried out to transform the curvilinear meshes in the physical domain into the equally spaced rectangular mesh in the computational domain. But in the air-bearing field, in most cases, only Cartesian rectangular mesh systems have been used, so the potential of the finite difference method has not been fully explored. When the geometry is very complicated, generating good structured meshes becomes

extremely difficult, and the finite volume and finite element methods are preferred. Both the finite volume and finite element method are integral methods that can be implemented in the physical domain without the need of coordinate transformation, and unstructured meshes can be used, which makes the mesh generation much easier and more flexible. Due to their local conservative quality, the finite volume methods are better at capturing sharp gradients, which, in the air bearing simulation, are unavoidable. Therefore the finite volume method is used in our approach.

Both explicit and implicit finite volume schemes have been constructed to locally integrate the general Reynolds equation over the control volumes that are taken to be the dual Voronoi polygons of the triangles (see fig. 1). The pressure is stored at the nodes. The property that each edge of the Voronoi polygon is a perpendicular bisector of the triangle edge that connects its generation point and the generating point of the adjacent Voronoi polygon makes it possible to use Patankar's strategy (ref.12) to evaluate the numerical flux across the edges of the control volume. Backward-Euler time differencing is used to discretize the unsteady term resulting in first order accuracy in time. The solution is evolved by marching in time until the steady state is reached. For steady state problems, the unsteady term is not needed physically, but it is kept here to serve as an under-relaxation term. When a relatively large time step is used, the unsteady term can be ignored, and the technique is more like a direct iteration than time marching. The Gauss-Seidel iteration technique is used to solve the linear algebraic equations. The implicit scheme is unconditionally stable, so an arbitrarily large CFL number (dimensionless time step) such as 1.0E12 can be used.

This approach also has its disadvantages. Due to its unstructured nature, additional information about connectivity between meshes must be stored, and a complicated date structure must be set up and used to locate the needed information. This indirect addressing results in increased computational time and memory. Most mature and well-tested numerical algorithms developed for structured mesh systems can no longer be used or at least require modification. Solvers over unstructured meshes are more complicated than their structured counterparts. Historically, solvers based on unstructured meshes were less efficient. However, after several years of development, implicit time stepping and multi-grid techniques have been successfully developed and implemented in unstructured solvers to the point where both their accuracy and efficiency can compete with their structured counterparts. Most importantly, compared with a rectangular Cartesian mesh, much fewer meshes are needed, and it is well known that the computational time increases quickly with increased mesh number.

Delaunay triangulation and its dual Voronoi regions

In 1850, Dirichlet proposed a way to divide a domain into a series of non-overlapping convex polygons. These polygons are called Voronoi regions. Each Voronoi region is associated with a generating point. A Voronoi region is defined as the set of points that are closer to their own generating point than to the generating point of any other Voronoi region (see Fig.1). The Delaunay triangulation is formed by connecting the generating points of neighboring Voronoi regions. From the definition of the Voronoi regions and their dual Delaunay triangulation, it can be seen, in two dimensions, that each vertex of a Voronoi polygon is the circumcircle center of one of the triangles. Each edge of the triangle is perpendicularly bisected by an edge of two

Voronoi polygons, with the two end points of the triangle edge being the generating points of the two neighboring Voronoi regions. The circumcircle of each triangle does not include any other triangle nodes. This last local quality of the Delaunay triangulation forms the basis for the different Delaunay triangulation algorithms.

Bowyer-Watson algorithm for two dimensional problems

In this algorithm (ref 2,3), new nodes are sequentially introduced into the previous triangulation. The triangle that encloses a new node is first located, then starting from this triangle, a local search is conducted to find all the triangles of the previous triangulation that violate the Delaunay circle test, which is used to check if the circumcircle of a certain triangle encloses the new node. Then all triangles violating the test are agglomerated into a cavity. These triangles are deleted from the list and new triangles are formed by connecting the new node with each edge of the cavity. The procedure continues until all the nodes have been inserted into the domain. This algorithm is known for its simplicity and readiness to be extended to three dimensional problems.

Sloan algorithm

Sloan's algorithm (ref.4) is a swapping process that works fast for two dimensional problems. In this scheme, when a new node is introduced into the domain, three new triangles are formed first by connecting the new node to the three vertices of the triangle that enclose it, then the old triangle is deleted. The net increase of the number of triangles is two. All adjacent triangles that share the edges opposite to the new node are placed in a "last-in-first-out" stack (initially with

three triangles in it). The algorithm is: pop up the last triangle in the stack, check if the new node is outside the circumcircle of the triangle, if not, Lawson's swapping scheme (ref.13) is executed. The diagonal of the quadrilateral formed by the popped up triangle and the adjacent one that contains the new node is replaced by the other diagonal that includes the new node, or in other words, the two old triangles are replace by two new triangles with the minimum angle being maximized . After this, place all the triangles that are opposite the new node into the stack. The process continues until the stack runs out of triangles.

Conforming background mesh generation

It is desirable for the boundary lines of the rails to coincide with triangle sides in the triangulation of the coarsest mesh. Our approach is based on Ruppert's scheme (ref. 6). First, a series of binary search trees (see Fig. 2) are constructed with the first elements being lines that form the boundary of the rail, or the line of the taper, or the boundary of the slider. To avoid certain confusing situations in the following encroachment test, one must take care to insure there is no overlapping between these 'mother' lines. Overlapping will happen when the rails touch the boundary of the slider. In that case, the two partly or completely overlapping lines are replaced by subsidiary lines that do not overlap. In most cases, the slider's shape is rectangular, so the whole domain can be initially divided into two triangles. If the slider's shape is not rectangular, the domain can be divided into more triangles. The list of triangles (initially with two triangles) is first expanded by inserting the vertices of the rail polygons into the domain using either the Bowyer-Watson algorithm or the Sloan algorithm. After that, starting from the beginning of the list, we check if the longest side of the triangle is smaller than a prescribed

tolerance or if the aspect ratio of the triangle is larger than a prescribed value. The aspect ratio is defined as the ratio between the longest and the shortest sides of the triangle. If the answer is no, we move to the next triangle in the list. Otherwise, we calculate the location of a new node at the circumcenter of the triangle. A search is conducted to see if any 'mother' or 'child' of the search tree list, which does not have any further 'children', is encroached by the node at the circumcenter. A line is said to be encroached when the new node lies within the diametral circle of the line. If the answer is no, the location of the new node is final. Otherwise, we divide the encroached line at the middle to form two 'children' and put them under the new 'mother' in the search tree. Then we modify the location of the new node from the circumcircle center of the triangle to the middle of the new 'mother'. Finally, we insert the new node into the domain using the above algorithms. Newly formed triangles are put at the end of the list, while triangles not in the triangulation any more are deleted from the list. The process is continued until the end of the triangle list is reached or the number of triangles has reached a prescribed value.

The above procedure gives the users great flexibility in controlling the quality and the distribution of the grids by choosing the following three parameters. The first one is the allowable minimum of the triangle's longest side, which controls the coarseness of the grids. It is beneficial to specify different minimum tolerances for the rail region and recessed region, because normally, the pressure profile in the rail region changes more rapidly than that in the recessed region. Consequently, a finer mesh is needed in the rail region. The second one is the aspect ratio of the triangles, which controls the quality of the grids. The last one is the maximum allowable number of triangles.

Clustering of meshes to the recess wall regions

As stated at the beginning, accurate simulation requires a fine mesh in the recess wall region. At the same time, the quality of the mesh must be maintained. Although, the Delaunay triangulation gives optimal connections, if the nodes are not properly positioned, the generated mesh still can be highly distorted. To make things worse, bad quality triangles tend to form local clusters. A highly skewed and distorted mesh may cause severe problems in the simulation, such as slowing down the convergence, destabilizing the code or deteriorating the result. The mixed longest-side-bisection and Delaunay technique of Rivara (ref. 7) is essentially a method that combines the node placement strategy of the longest-side-bisection technique (ref. 14) with the Delaunay node insertion technique. The propagating node placement strategy of the longest-side-bisection technique gives the advantage of smoothly distributed nodes with a linear cost. The Delaunay technique gives the optimal connections of these nodes. When they are coupled together, satisfactory results can be obtained. Although the pure longest-side-bisection technique is faster than the mixed technique, it is worthwhile to sacrifice a certain amount of computer resource to the quality of the mesh.

To cluster a fine mesh in the recess wall region, starting from the first triangle in the triangle list, we conduct a test to see if the maximum recess height difference between any pair of nodes is greater than a prescribed value. If the answer is no, we move to the next triangle in the list, otherwise we find the longest side of the triangle and put its middle point into a last-in-first-out stack. Then we check to see if the longest side is also the longest side of the neighboring triangle. If the answer is no, then we set the neighboring triangle as the current triangle. We find the longest side of the current triangle and insert its middle point into the stack, and then another

check is conducted. The check-insert process continues until the longest side of the current triangle is also the longest side of its neighboring triangle. When this is done, we check each of the triangles that share one of the vertices of the longest side of the original triangle from the list to see if its longest side is longer than the longest side of the triangle from the list, multiplied by a prescribed constant. If that is true, we insert the middle point of the longest side of this triangle into the stack. After this, all the points in the stack are inserted into the domain using either the Bowyer-Watson algorithm or the Sloan algorithm. New triangles are again put at the end of the list, and triangles no longer in the triangulation are deleted from the list. The refinement process stops when either the end of the list is reached (in this case, we say the process has converged) or the total number of triangles has reached a prescribed value.

By changing the allowable maximum value of the recess height difference in each triangle, we can easily control, on the average, how many node points are needed across the narrow recess wall region. For example, when the dimensionless difference between the recess height of the rail and the fully recessed region is 1, if we choose the allowable value to be 0.33, then, when the refinement process converges, at least three nodes appear across the recess wall.

Grid adaptation

Except for the regions with dramatic geometric changes (which have been captured by the above refinement algorithms), for sliders with a complex rail system it is difficult to decide in advance the location of the regions with high-pressure gradients, where high grid resolution is needed. A grid adaptation technique proves to be an economical way to improve the accuracy and efficiency of numerical simulation in such situations. Grid adaptation techniques generally

fall into two categories; the r-refinement and the h-refinement methods. In the r-refinement strategy, the total number of nodes is fixed, while the locations of the nodes are gradually shifted in favor of the regions where fine meshes are needed. This is the method used in the current rectangular grid CML code. In the h-refinement strategy, the positions of the nodes do not move, instead, nodes are locally added to or deleted from the region according to the need. The method of incremental Delaunay triangulation makes the h-refinement a natural choice.

A good sensor that can detect the place where fine meshes are needed is important in grid adaptation. If the sensor is too sensitive, wide spread regions will be refined, resulting in too many nodes, which makes the adaptation meaningless. The other extreme is also not acceptable. In the air-bearing simulation, it is desirable to add fine meshes to the high-pressure gradient regions. Pressure gradient seems to be a good choice for the sensor. But adaptation is basically an iteration process, in which meshes are added or moved gradually until a certain standard is met by the sensor. Pressure gradient is a physical value, which does not change with the size of the grids when the solution is grid converged. In most cases, the gradient will increase with the refinement of the mesh. This makes the pressure gradient a poor choice as the sensor. Instead, the absolute value of the pressure difference along the edges of the triangle can be used as the sensor. To have the same absolute pressure difference, a smaller mesh size is needed in the high-pressure gradient region. When the mesh is refined, the absolute pressure difference will drop, which makes convergence possible.

Another important thing is to choose an adequate value for the sensor below which no refinement is carried out. The average value of the absolute pressure difference of all the lines, multiplied by a prescribed constant, is used in our approach.

Only one adaptation is carried out in our simulation. After enough iterations are done on the relative coarse mesh, the average value of the absolute pressure difference of all the lines is calculated. Starting from the first triangle in the triangle list, the largest absolute pressure difference of the triangle is calculated using the current pressure distribution. If the value is greater than the average value multiplied by a constant, the same mixed longest-side-bisection and Delaunay refinement process is carried out as above. The pressure at the new nodes is taken as the mean value of the two nodes of the longest edge. The process continues until the end of the list is reached.

Laplacian smoothing

After the generation of the mesh, a simple Laplacian type smoother is widely used to slightly reposition the location of the nodes to distribute them more smoothly throughout the domain (see ref. 15). It is an iteration process. At each step, the locations of the nodes are updated as

$$x = x_{old} + \frac{\nu}{n}\sum_{k=1}^{n}\left(x_k - x_{old}\right), \qquad y = y_{old} + \frac{\nu}{n}\sum_{k=1}^{n}\left(y_k - y_{old}\right), \tag{1 a,b}$$

where ϖ serves as a relaxation factor, and n is the number of the neighboring nodes. Either a fixed number iteration can be carried out or it can continue until convergence is reached. Our experience shows that a fixed number of 20 to 30 iterations will give a satisfactory result.

Governing equations and boundary conditions

In air bearing simulations, the generalized Reynolds equation is solved to get the pressure field. In the derivation of the traditional Reynolds equation, many severe assumptions, based on dimensional analysis, have been made. The slider air bearing brings up additional challenging issues. Because of the extremely narrow spacing between the slider and the disk (in the order of 10 nm, which is only a fraction of the mean free path of the gas molecules), the gas in the spacing is extremely rarefied and the gas molecules near the solid surfaces no long simply adhere to them (although the non-slip condition has been under attack even in normal situations, it is much more severe here), instead they are slipping. The continuity and non-slip condition assumptions are no longer an acceptable approach to the actual physics. Until now, the modified versions of the Reynolds equation that take the rarefaction and slipping effect into account give the best results. The modification makes the equation appear more complicated, but from the numerical point of view, it also gives the equation a better numerical quality than that of the traditional Reynolds equation, since the pressure fields given by modified versions have less steep pressure profiles than those predicted by the traditional equation. The different versions of the Reynolds equation can be written in a unified dimensionless from as

$$s\,\frac{\partial}{\partial T}(PH)=\frac{\partial}{\partial X}\left(QPH^{3}\frac{\partial P}{\partial X}-\Lambda_{x}PH\right)+\frac{\partial}{\partial Y}\left(QPH^{3}\frac{\partial P}{\partial Y}-\Lambda_{y}PH\right), \tag{2}$$

where $s=\dfrac{12\mu wL^{2}}{P_{a}h_{m}^{2}}$ is the squeeze number, which represents the relative importance between the unsteady effect and the diffusion effect, with μ being the dynamic viscosity of the gas, ω

being the angular velocity of the disk, L being the length scale of the slider, p_a being the ambient pressure, h_m being the flying height. $T = wt$ is the dimensionless time, $X = \frac{x}{L}$ is the dimensionless x coordinate, $Y = \frac{y}{L}$ is the dimensionless y coordinate, $P = \frac{p}{p_a}$ is the dimensionless pressure, $H = \frac{h}{h_m}$ is the dimensionless normal distance from the disk to the slider, $\Lambda_x = \frac{6mUL}{p_a h_m^2}$ and $\Lambda_y = \frac{6mVL}{p_a h_m^2}$ are the bearing numbers in the x and y direction, respectively, which represent the relative importance between the convection effect and the diffusion effect. Q is the flow factor, which marks the difference between different rarefaction models of the equation. Different Q for different models are briefly listed below. The details can be found in references 16,17 and 18.

$Q = 1,$ continuum model.

$Q = 1 + 6a\frac{K_n}{PH},$ first order slip model.

$Q = 1 + 6\frac{K_n}{PH} + 6\left(\frac{K_n}{PH}\right)^2,$ second order slip model.

$Q = f\left(\frac{K_n}{PH}\right),$ Fukui-Kaneko model.

Where $a = \frac{2-a}{a}$,α is the accomodation factor, $K_n = \frac{l}{h_m}$ is the Knudsen number, λ is the mean free molecular path.

Along the outside boundary of the slider, the pressure is simply taken as the ambient pressure.

Explicit finite volume discretization of the governing equation

After the computational domain has been decomposed into a series of non-overlapping conforming triangles, the Reynolds equation is locally integrated over each control volume. Generally, there are two choices for storing the dependent variables. One is to store the variables at the centroid of the triangles, where the corresponding finite volume scheme is called the cell centered scheme. The other is to store the variables at the vertices of the triangles, where the corresponding scheme is called the nodal scheme or vertex based scheme. Here we store our pressure at the triangle vertices. This is based on the consideration that the number of triangles is roughly two times the number of nodes. This can be seen from the Delaunay triangulation process. In Sloan's algorithm (ref. 4), when a new node is introduced into the domain, the old triangle enclosing the new node is first replaced by three new triangles with a net increase of two. In the following operation, the number of triangles is not changed. Using the same mesh, since the number of unknowns is smaller, the nodal scheme normally runs faster than the cell centered scheme. In addition, the information stored at the vertices can be more conveniently and fully used in the nodal scheme approach. We chose the Voronoi polygons as our control volumes (see fig. 3). The Voronoi polygons and the triangles form dual meshes to each other.

The Reynolds equation in integral form can be written as

$$\iint_{\Omega}\left[s\,\frac{\partial}{\partial T}(PH)-\frac{\partial}{\partial X}\left(QPH^{3}\frac{\partial P}{\partial X}-\Lambda_{x}PH \right)-\frac{\partial}{\partial Y}\left(QPH^{3}\frac{\partial P}{\partial Y}-\Lambda_{y}PH \right)\right]d\Omega = 0. \tag{3}$$

After using the divergence theorem, we have

$$\iint_{\Omega} s\,\frac{\partial(PH)}{\partial T}d\Omega-\oint_{s}\left[\left(QPH^{3}\frac{\partial P}{\partial X}-\Lambda_{x}PH \right)n_{x}+\left(QPH^{3}\frac{\partial P}{\partial Y}-\Lambda_{y}PH \right)n_{y}\right]ds = 0, \tag{4}$$

where **s** is the boundary around the control volume, ds is the length of the line element, and n_x and n_y are the unit outward normal vector components in the x and y direction respectively. Define the local normal bearing number as

$$\Lambda_{n}=\Lambda_{x}n_{x}+\Lambda_{y}n_{y}=\frac{6mL\left(Un_{x}+Vn_{y}\right)}{p_{a}h_{m}{}^{2}}=\frac{6mLU_{n}}{p_{a}h_{m}{}^{2}}, \tag{5}$$

with $U_{n}=Un_{x}+Vn_{y}$ being the disk velosity component normal to the control volume boundary, and the normal outward pressure derivative along the boundary as

$$\frac{\partial P}{\partial n}=\frac{\partial P}{\partial X}n_{x}+\frac{\partial P}{\partial Y}n_{y}. \tag{6}$$

Then the integral equation becomes

$$\iint_{\Omega} s\,\frac{\partial(PH)}{\partial T}d\Omega - \oint_{s}\left(QPH^{3}\frac{\partial P}{\partial n} - \Lambda_{n}PH\right)ds = 0. \tag{7}$$

Over the Voronoi polygon surrounding vertex i (see Fig. 3), the above equation can be approximated as

$$\left(As\,\frac{\partial(PH)}{\partial T}\right)_{i} - \sum_{j=1}^{M}\left(QPH^{3}\frac{\partial P}{\partial n} - \Lambda_{n}\right)_{ij} ds_{j} = 0, \tag{8}$$

where the first term is the unsteady term, A_i is the area of the control volume around vertex i and the lumped assuption (the value of the integrand at vertex i is taken to be the value all over the control volume) has been used. M is the number of sides of the Voronoi polygon and ds_j is the length of its jth side. The expression in the bracket of the second term represents the numerical flux across the sides, which can be evaluated using the information of vertices i and j. Backward-Euler time differencing is used to discretize the time derivative of the first term, resulting in first order accuracy in time,

$$\left(As\,\frac{\partial(PH)}{\partial T}\right)_{i} = A_{i}s\,\frac{\left[(PH)_{i}^{\,n+1} - (PH)_{i}^{\,n}\right]}{\Delta T}, \tag{9}$$

Here subscript n+1 represents the new value at time level n+1, n means the old value at time level n. The Delaunay trangulation and its dual Voronoi polygon have the quality that each side of the Voronoi polygon is a perpendicular bisector of the triangle side that connects the two adjacent generating points of the Voronoi polygon where the pressure is stored (in Fig.3, the line connecting vertices i and j is perpendicularly bisected by the jth side of the Voronoi polygon around i). Patankar's strategy (ref. 12) is employed to calculate the numerical flux across each side of the control volume. The governing equation is locally taken to be steady and one dimensional with the space coordinate in the direction of the local outward unit normal vector, and the equation is linearized by freezing the coefficients with the value of the last time step. The resulting ordinary differential equation is solved to give the needed numerical flux. The final explicit discretization form of the equation can be written as

$$\frac{A_i H_i}{\Delta T} P_i^{n+1} = \sum_{j=1,\, j\neq i}^{M} C_{ij} P_j^{\,n} + \left(\frac{A_i H_i}{\Delta T} - C_i \right) P_i^{\,n}, \tag{10 a}$$

with

$$C_{ij} = D_{ij} A\left(\left|P_{ij}\right|\right) + \left[\left[-F_{ij}, 0\right]\right], \tag{10 b}$$

$$C_i = \sum_{j=1,\, j\neq i}^{M} \left(C_{ij} + F_{ij}\right), \tag{10 c}$$

where the operator $[[a,b]]$ yields the larger of a and b. Function $A(|P|)$ can be written for the different schemes as (ref. 12)

$1-0.5|P|$, Central difference

1, Upwind

$[[0,1-0.5|P|]]$, Hybrid

$[[0,(1-0.1|P|)^5]]$, Power law

$\frac{|P|}{[\exp(|P|)-1]}$, Exponential

and

$$D_{ij} = \frac{(QPH^3)_{ij}\, ds_j}{s\, l_{ij}},$$

$$F_{ij} = \frac{\Lambda_n H_{ij} ds_j}{s},$$

$$P_{ij} = \frac{F_{ij}}{D_{ij}}.$$

l_{ij} is the distance between vertices i and j. All other values with the double subscripts ij are related to the vertices i and j.

Equation 10 is solved by marching in time, where a maximum time step is enforced by the stability requirement. The CFL number (Courant-Friedrichs-Lewy number) can be defined as

$\frac{\Lambda \Delta T}{s \Delta l}$, which is the local dimensionless time step that governs the stability of the wave equation. Λ is taken as the local value of $\sqrt{\Lambda_x{}^2 + \Lambda_y{}^2}$, and Δl is taken to be the diameter of the control volume. If the ultimate goal is the steady state solution, then the time step does not need to be uniform throughout the domain. Instead, it can be taken as the local maximum allowable value. Larger time steps can be used for larger control volumes. We chose a uniform CFL number. The local time stepping technique allows the information to propagate more quickly throughout the domain to increase the convergence rate.

Extension of the explicit scheme to a fully implicit scheme

The fully implicit scheme can be written at each time step as

$$\left(\frac{A_i H_i}{\Delta T} + C_i \right) P_i{}^{n+1} - \sum_{j=1, i \neq j}^{M} C_{ij} P_j{}^{n+1} = \frac{A_i H_i}{\Delta T} P_i{}^{n}, \tag{11}$$

in which all of the coefficents are the same as in the explicit scheme. For the steady state problem, the unsteady term only serves as an under-relaxation term. The simultanous linear algebraic equations can be written in matrix form as,

$$[C]^n \{P\}^{n+1} = \{d\}^n, \tag{12}$$

where $[C]^n$ is the coefficient matrix evaluated using the most recent values available , $\{P\}^{n+1}$ is the unknown vector at the new time level, and $\{d\}$is the known source term. Because of the

unstructured nature of the mesh, $[C]^n$ is a sparce matrix. At each row, only the elements that are associated with the directly neighboring vertices of the diagonal element vertex are not zero. Matrix $[C]^n$ can be broken into three parts,

$$[C]^n = [M]^n + [D]^n + [N]^n, \tag{13}$$

where $[M]^n$ is the lower triangular matrix, $[D]^n$ is the diagonal matrix and $[N]^n$ is the upper triangular matrix. The solution of the algebraic equation can be obtained by classical point iterations. For the point Jacobi method, the equations can be rearranged into

$$[D]^n \{P\}^{n+1} = \{d\} - [M]^n \{P\}^n - [N]^n \{P\}^n. \tag{14}$$

This eqauation can be readily solved by direct inversion of $[D]^n$. To speed up the covergence, the newly updated variables can be used, which result in the point Gauss-Seidel iteration

$$[D]^n \{P\}^{i+1} = \{d\} - [M]^n \{P\}^{i+1} - [N]^n \{P\}^i. \tag{15}$$

Here the superscript i means the level of the inner iteration. Each iteration consists of two sub-iterations. The outer iteration is used to update the matrices, the inner iteration is used to solve the resulting linear equations. Since the final value of the matrices depends on the converged pressure, it is not necessary to get a converged result in the inner iteration; instead, a fixed

number of iterations are carried out. The first sweep of the iteration starts from the beginning of the list, then the following sweep begins with the end of the list. This takes into account the effect that the diffusion terms of the Reynolds equation are elliptic in nature, and the disturbance is propagating in all directions at the same time. In this way, unconditional stability is achieved, and the time step can be chosen as arbitrarily large. When it is taken to be infinite, the above process is equivalent to direct interation.

Results and Discussion

The "LU" slider of Zeng and Bogy (ref.19) was chosen as an example for the mesh generation techniques as well as the explicit and implicit finite volume schemes. The simulation was done for a disk velocity of 13 m/s with a flying attitude of 25 nm fly height, 155 mrad pitch, 0 roll, and 0 skew. Figure 4 shows the coarsest conforming mesh and its dual Voronoi polygons with 605 nodes. It serves also as the coarsest level of mesh in the multi-grid technique that will be discussed in another paper. The figure clearly shows that the boundaries of the rail, the line that defines the taper and the boundaries of the slider are all well represented in the triangulation. This is a desirable quality for the multi-grid iterations. The meshes on the rail are also made finer than in the fully recessed region, and the mesh size and distribution can be easily controlled as explained above. Figure 5 shows a finer mesh with 7614 nodes. It can be seen that extremely fine meshes have been clustered toward the recess wall regions which makes the specification of the wall profile possible. Normally these regions with rapid geometrical changes are also regions with high-pressure gradients. The meshes on the rail were also refined. A high mesh size gradient was achieved while at the same time preserving the quality of the mesh. Figure 6 shows

the adaptively refined mesh with 18150 nodes. From this figure we can see that several regions have been refined with a very fine mesh. They are the regions around the boundary and along the taper line of the two forward pads, the downwind recess wall region and the trailing part of the rail. From Figs. 7 and 8, which present the pressure contours and the three dimensional pressure profile, it can be seen that the high-pressure gradient regions have been accurately and efficiently captured by the adaptation process. Figure 9 shows the convergence history of the explicit and the implicit schemes. The explicit time marching was done on the mesh shown in Fig. 5 with a CFL of 1. It is clear that the explicit scheme converges very slowly. Even with 4000 time steps, the result is far from converged. The sudden jump of the error for the implicit scheme at the 200 time step is due to the mesh adaptation. We first did time marching using the mesh shown in Fig. 5. After the solution converged to some extent, the adaptation occurred. The following time marching was done on the adapted mesh. The CFL number was 1.0E12. The convergence history shows that the convergence rate of the implicit technique has been greatly improved, with orders of simulation time being saved. Its rapid high frequency error damping quality also makes it a good candidate for multi-grid techniques. The converged result gives a positive force of 6.88 grams and a negative force of 3.75 grams. The maximum pressure is 4.85, and the minimum pressure is -0.74.

Summary and conclusions

In this paper, we present unstructured triangular mesh generation and adaptation techniques together with explicit and implicit finite volume schemes that are suitable for slider air bearing simulations for hard disk drives. Three different refinement and adaptation techniques based on

two different Delaunay triangulation algorithms have been utilized to cluster fine meshes to the rail top, the recess wall and the high pressure gradient regions. At the coarsest level, the boundaries of the rail and the slider are well represented in the triangulation, which is important for the multi-grid iteration. At the second level, very fine meshes are clustered in the recess wall region where the geometry and the pressure profile change rapidly. In the simulation process, a mesh adaptation technique is used to refine the meshes in the high-pressure gradient regions. The undivided absolute pressure difference is used as the sensor in this mesh adaptation. The overall mesh generation process gives the user great flexibility and control over the distribution and quality of the meshes. This is demonstrated to be an efficient and convenient way of generating quality meshes over the complex geometries in the air bearing simulations. An explicit finite volume scheme is first constructed, with which the steady state solution can be obtained by marching in time. But due to the stability limitation, only very small time steps can be used, which makes the convergence very slow. The explicit scheme is successfully extended to an implicit scheme in which the Gauss-Seidel point iteration technique is used to solve the simultaneous algebraic equations. The resulting scheme is unconditionally stable, and an arbitrarily large time step can be used. The convergence rate is greatly improved over the explicit scheme.

In a following paper, Part 2, we will introduce a multi-grid solution method and extend the solution to the inverse problem for which the bearing load is specified and the flying attitude is obtained.

Acknowledgments

This study was supported by the Computer Mechanics Laboratory at the University of California at Berkeley.

Reference

1 Lo, S. H., *Int. J. Numer. Methods. Eng*., **21**,1403 (1985).

2 Bowyer, A., *Comput. J*., **24**, No.2, 62 (1981).

3 Watson, D. F., *Comput. J.* , **24**, No.2, 167 (1981).

4 Sloan, S. W., *Advances in Engng Software*. **9**, 34 (1987).

5 Rebay, S., *J. Comput. Phys*. **106,** 125 (1993).

6 Ruppert, J., *J. of Algorithms*, **18**, 548 (1995).

7 Rivara, M., and Inostroza, P., *Int. J. Numer. Methods. Eng*., **40**, 581 (1997).

8 White, J. W. and Nigam, A., *ASME J. Lubrication Technology*, **105**, 485 (1980).

9 Lu, S. and Bogy, D. B., *CML Technical Report No. 94-016* (1994)

10 Garcia-Suarez, C., Bogy, D. B., and Talke, F. E., *ASLE SP-16*, 90 (1984).

11 Thompson, J. F., Thames, F. C., and Mastin, C. W., *NASA CR-2729* (1977).

12 Patankar, S. V., McGraw-Hill, New York (1980).

13 Lawson, C. L., *Mathematical Software III*, pp.161-194. Academic Press, New York (1977).

14 Rivara, M., *Int. J. Numer. Methods. Eng*., **40**, 3313 (1997).

15 Mavriplis, D. J., *AIAA J*., **26**, 824 (1988)

16 Burgdorfer, A., *ASME J. of Basic Engineering*, **81**, 94 (1959).

17 Hsia, Y. T., and Domoto, G. A., *ASME J. of Lubrication Technology*, **105**, 120 (1983).

18 Fukui, S., and Kaneko, R., *ASME J. of Tribology*, **110**, 335 (1988).

19 Zeng, Q. H. and Bogy D. B., to appear at *IEEE Trans. of Magnatics*.

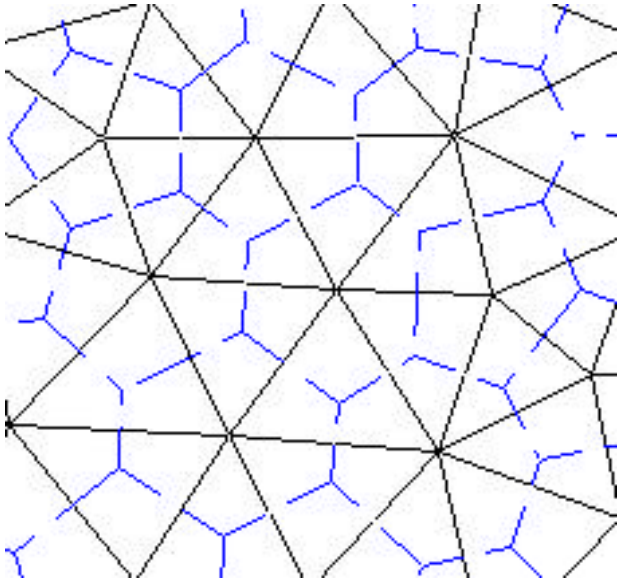

Fig. 1: Delaunay triangulation and its dual Voronoi diagram.

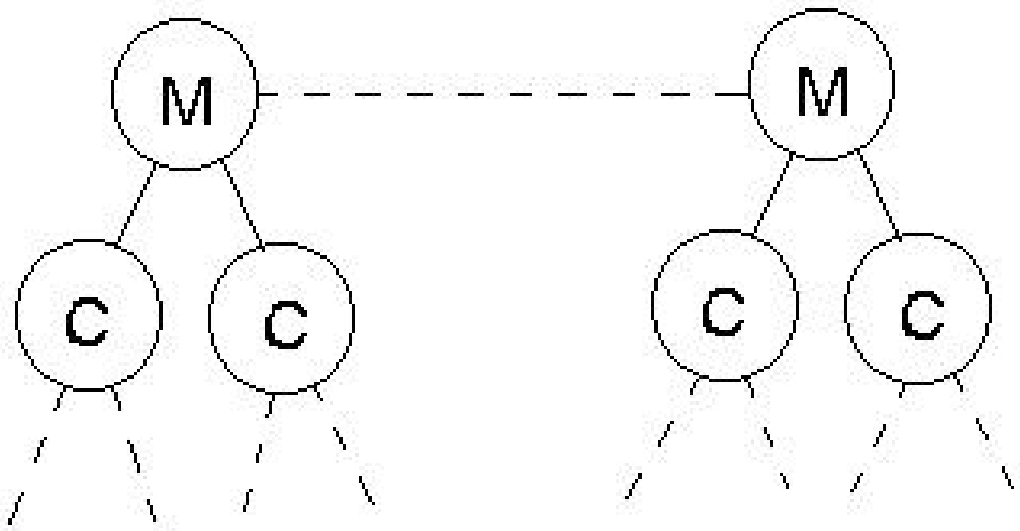

Fig. 2: Binary search tree for encroachment test..

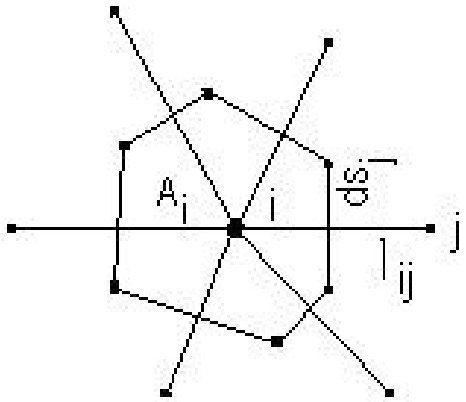

Fig. 3: Control volume for vertex based finite volume schemes.

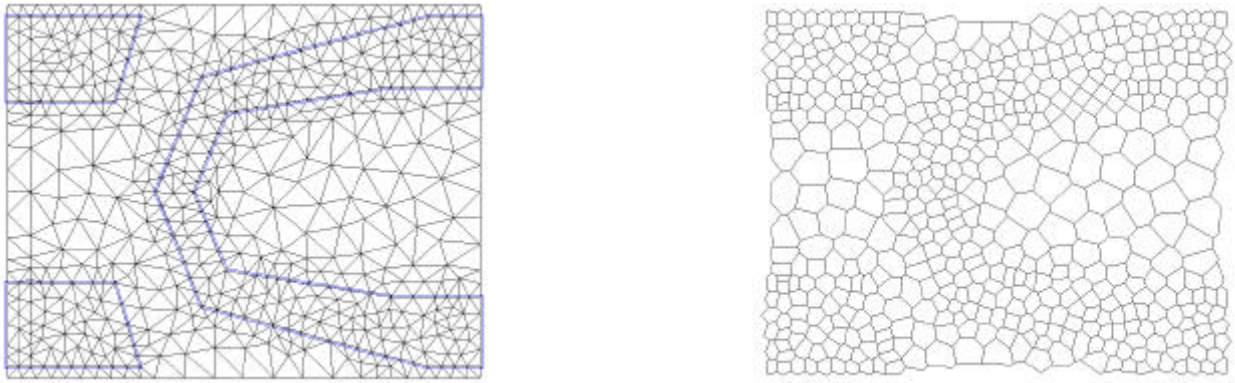

Fig. 4: Conforming background mesh and its dual Voronoi polygons (605 nodes).

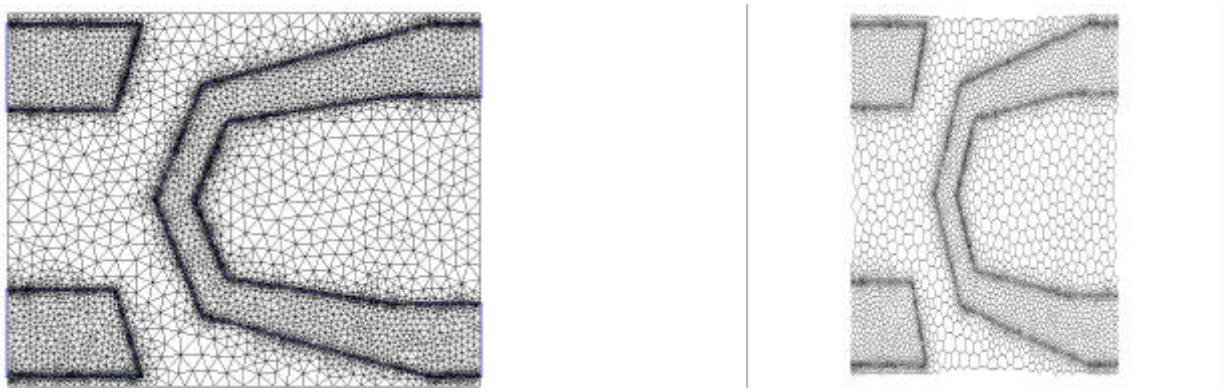

Fig. 5: Finer mesh with recess wall region refined and its Voronoi polygons (7614 nodes).

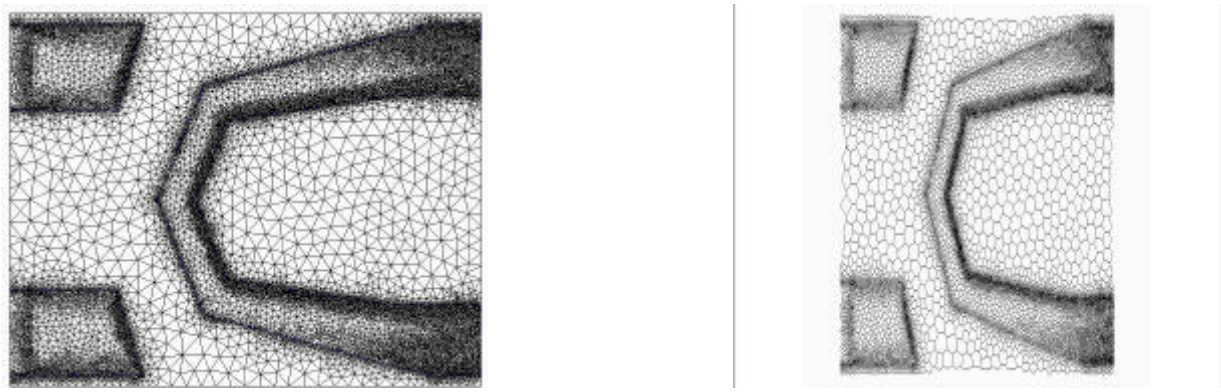

Fig. 6: Adapted finest mesh and its dual Voronoi polygons (18150 nodes).

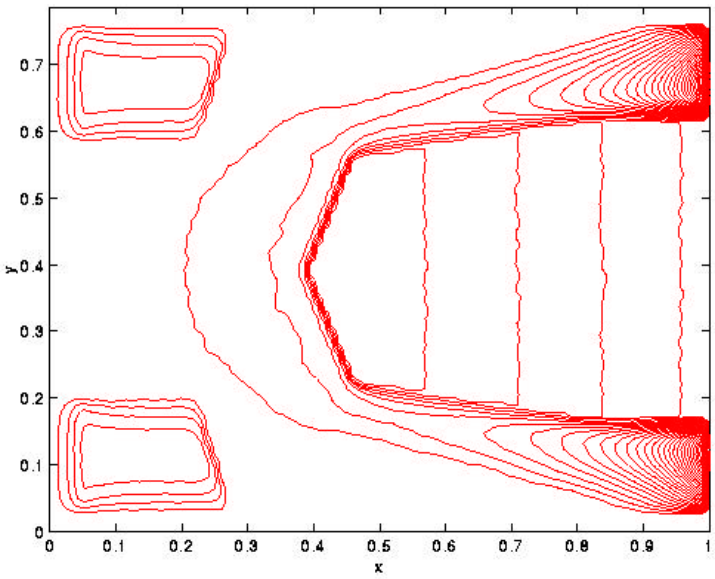

Fig. 7: Pressure contours.

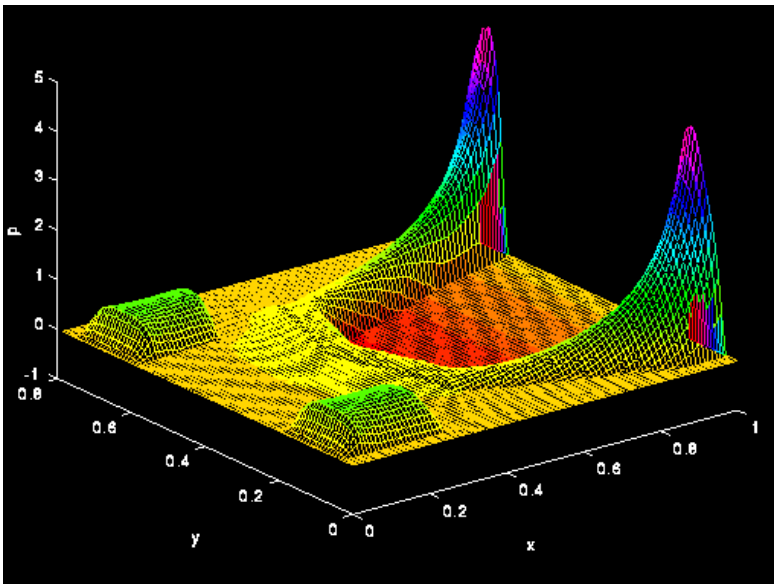

Fig. 8: Three dimensional pressure profile.

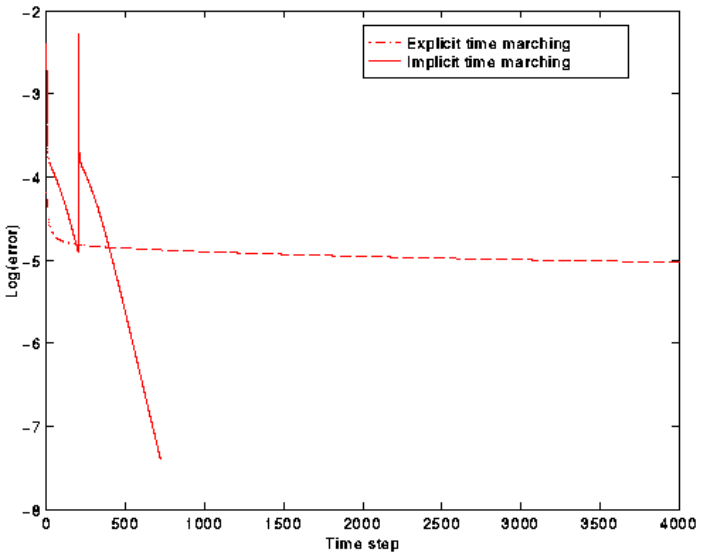

Fig. 9: Comparision of the convergence history for explicit and implicit schemes.