

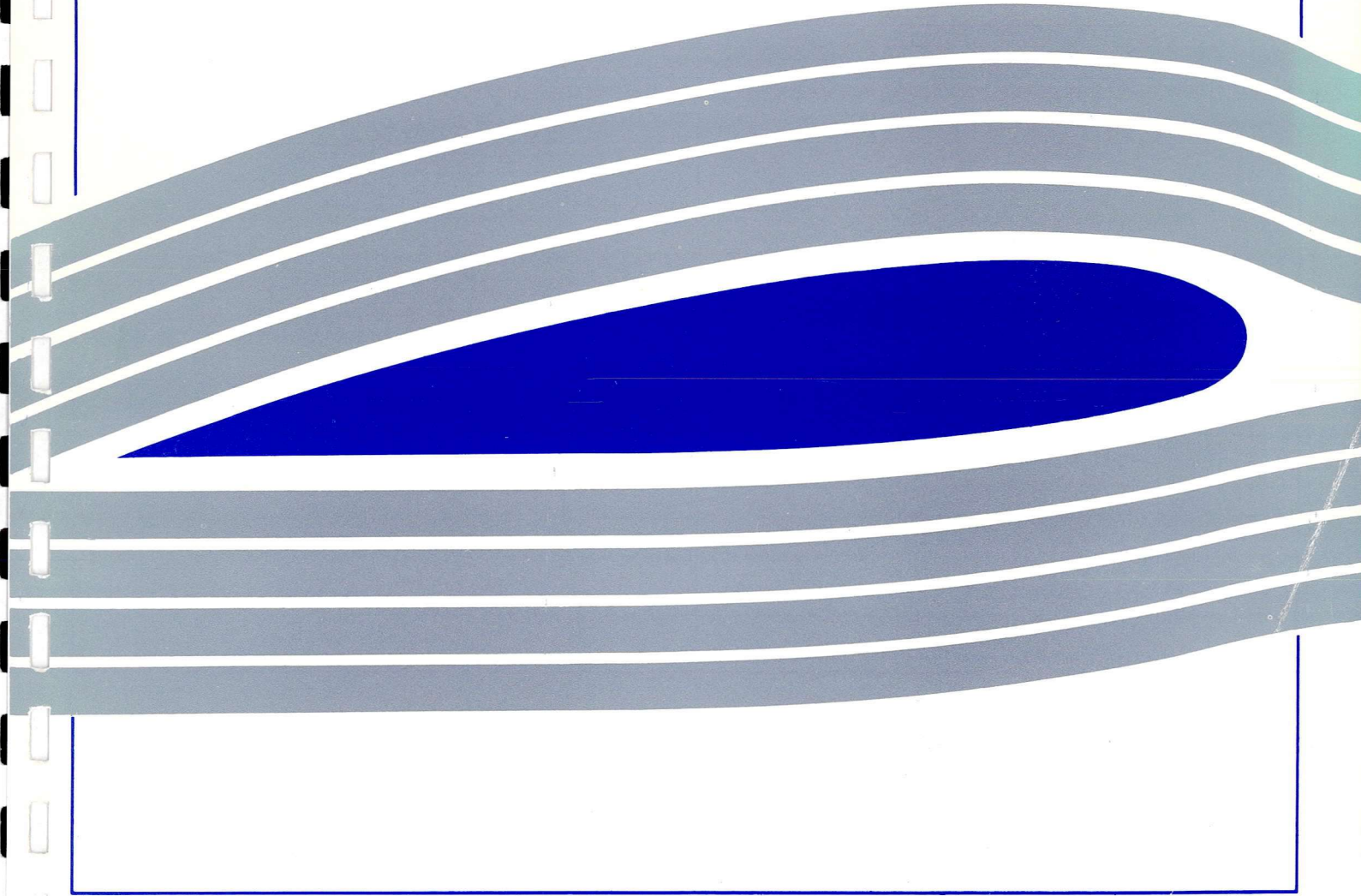
Engineering  
PERIODICALS  
U5000



University of Glasgow  
DEPARTMENT OF  
**AEROSPACE  
ENGINEERING**

Application of a Zonal Decomposition Algorithm,  
to Improve the Computational Operation Count  
of the Discrete Vortex Method Calculation.

I. J. Taylor<sup>(1)</sup> and M. Vezza<sup>(2)</sup>



Application of a Zonal Decomposition Algorithm,  
to Improve the Computational Operation Count  
of the Discrete Vortex Method Calculation.

I. J. Taylor<sup>(1)</sup> and M. Vezza<sup>(2)</sup>

Department of Aerospace Engineering  
James Watt Building South  
University of Glasgow  
Glasgow G12 8QQ

June 1997

## **SUMMARY.**

The vortex method has proved a very useful tool for analysing separated, incompressible flow around two dimensional bodies. The method utilises a grid free, Lagrangian approach, to discretise the vorticity field into a series of vortex particles. These particles are then tracked in time, using the Biot-Savart law to calculate the velocity field. This calculation requires the velocity of each vortex to be found as a sum over all other particles in the flow field. A Discrete Vortex Method (DVM) has been developed at the Department of Aerospace Engineering, University of Glasgow. Currently, this vortex method uses a direct summation technique, which although relatively simple, leads to a computational operation count proportional to the square of the number of particles. In calculations that use a large number of particles, such as bluff body models, the direct summation technique becomes prohibitively expensive.

A new algorithm for the velocity calculation has now been included in the DVM and is presented in this report. The procedure uses a zonal decomposition algorithm for the velocity summation. This allows the effect of groups of particles on the velocity to be calculated using a single series expansion, thus significantly reducing the operation count of the calculation. The algorithm utilises a hierarchical technique, so that the largest possible group of particles is used for each series expansion. The resulting operation count is  $O(N+N\log N)$ , and therefore offers a significant improvement over the direct summation method.

The support and advice of Dr. Marco Vezza, whilst developing this algorithm, is gratefully acknowledged.

## CONTENTS.

- 1.0 Introduction.
  - 1.1 Overview.
  - 1.2 Basis of Discrete Vortex Model.
  
- 2.0 Techniques for fast summation in N-body / particle simulation methods.
  - 2.1 Introduction.
  - 2.2 Vortex in Cell Method.
  - 2.3 Method of Local Corrections.
  - 2.4 Tree Codes, Multipole Expansions and the Fast Multipole Method.
    - 2.4.1 The Fast Multipole Method Algorithm.
  
- 3.0 Implementation of a Zonal Decomposition Algorithm in the Vortex Method.
  - 3.1 Decomposition of Flow Field into a Hierarchical Structure of Square Zones.
  - 3.2 Velocity Calculation using Zonal Decomposition and Series Expansion.
  - 3.3 Comparison of Algorithm with Fast Multipole Method.
  
- 4.0 Results.
  - 4.1 Performance of Algorithm
  - 4.2 Error Introduced by the Zonal Decomposition Algorithm.
  - 4.3 Future Work.
  
- 5.0 Conclusions.

## References

## Appendices.

- A : Derivation of Series Expansion for Zonal Decomposition.
- B : Operation Count of Zonal Decomposition Algorithm.
- C : Optimisation of Parameters in the Zonal Decomposition Algorithm

## Nomenclature.

## Figures.

## 1.0 INTRODUCTION.

### 1.1 Overview

The discrete vortex method has proved to be a very useful analysis tool for studying unsteady, incompressible viscous flows. The method is based on a discretisation of the vorticity field into a series of vortex particles, which are free to move in the flow field that they collectively induce. This Lagrangian approach eliminates the necessity for a calculation grid and hence removes some of the problems associated with grid based methods; these include numerical diffusion and difficulties in achieving good resolution of small scale vortical structures. The vortex method has an advantage in this area as the particles are concentrated in areas of vorticity, and are more likely to capture these small scale structures.

However, one of the main drawbacks of vortex methods is that the computational cost becomes prohibitively more expensive as the number of particles used to model the flow field increases. The velocity of an individual vortex particle is calculated by taking the sum of the influence of all other particles. It is clear that the time taken for the velocity calculation of  $N$  particles, will be proportional to  $N^2$  (Fig. 1.1). It is clear that as the number of particles increases, the computation becomes prohibitively more expensive. This is termed the  $N$ -Body problem and is common to any model that involves  $N$  bodies, where each body is interacting with all the remaining bodies. Examples of similar problems occur in Astrophysics where the bodies are stars, and Plasma physics where the bodies are ions and electrons.

Techniques have been developed to reduce the cost of this calculation, reducing the operation count to  $O(N \log N)$  or even  $O(N)$ . A number of these techniques are discussed in this report, along with the advantages and disadvantages of each of these methods.

Also presented in the report, is the method that has been used to reduce the operation count of the Discrete Vortex Method (DVM) that has been developed at the Department of Aerospace Engineering, University of Glasgow [1-3]. Comparison of the speed of the new algorithm with the original direct summation method of calculation is made, as well as comparing the accuracy of the new method. Further improvements in the efficiency and operation count of the vortex method are possible and indications of how these may be achieved are also presented in the report.

## 1.2 Basis of Discrete Vortex Model.

The vortex method is based on the Navier-Stokes equations for two dimensional, incompressible flow. In velocity and pressure form :

$$\text{Continuity} \quad : \quad \nabla \cdot \bar{\mathbf{U}} = 0 \quad (1.1)$$

$$\text{Momentum} \quad : \quad \frac{D\bar{\mathbf{U}}}{Dt} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \bar{\mathbf{U}} \quad (1.2)$$

$$\text{Boundary Condition} \quad : \quad \bar{\mathbf{U}} = \bar{\mathbf{U}}_i \text{ on } S_i \text{ and } \bar{\mathbf{U}} = \bar{\mathbf{U}}_\infty \text{ on } S_\infty \quad (1.3)$$

Using the definition of vorticity,  $\bar{\omega} = \nabla \times \bar{\mathbf{U}}$  with  $\bar{\omega} = \bar{\mathbf{k}}\omega$ , and defining a vector potential,  $\bar{\Psi}$  such that,  $\bar{\mathbf{U}} = \nabla \times \bar{\Psi}$ ,  $\bar{\Psi} = \bar{\mathbf{k}}\Psi$ ,  $\nabla \cdot \bar{\Psi} = 0$ , then the governing equations (1.1 - 1.3) can be rewritten in vorticity and stream function form :

$$\text{Continuity} \quad : \quad \nabla^2 \Psi = -\omega \quad (1.4)$$

$$\text{Momentum} \quad : \quad \frac{D\omega}{Dt} = \nu \nabla^2 \omega \quad (1.5)$$

$$\text{Boundary Condition} \quad : \quad \nabla \Psi = \nabla \Psi_i \text{ on } S_i \text{ and } \nabla \Psi = \nabla \Psi_\infty \text{ on } S_\infty \quad (1.6)$$

The discretisation of the flow field into a series of vortex particles, provide a solution to these equations. By employing Green's second identity and implementing the boundary conditions, the velocity can be found from (1.7). This is analogous to the Biot-Savart law in electromagnetism, for the magnetic field induced at a point.

$$\bar{\mathbf{U}}_p = \bar{\mathbf{U}}_\infty + \frac{1}{2\pi} \int_F \omega \frac{\bar{\mathbf{k}} \times (\bar{\mathbf{r}}_p - \bar{\mathbf{r}})}{\|\bar{\mathbf{r}}_p - \bar{\mathbf{r}}\|^2} dF \quad (1.7)$$

The body is defined by a series of nodes, that are connected to form panels. The boundary conditions are implemented such that the vortex strength required at each node is calculated, to give zero mass flow across each panel. Vortices are then released into the wake, where the convection and diffusion of the vortex particles are calculated, using (1.7)

and a random walk respectively. A much more detailed description of the method and the discretisation of the vorticity field is given in [1-3].

## **2.0 TECHNIQUES FOR FAST SUMMATION IN N-BODY / PARTICLE SIMULATION METHODS.**

### **2.1 Introduction.**

When calculating the velocity in the vortex method at a point in the flow field, the influence of all the vortex particles must be taken into account. This involves a calculation of  $O(N^2)$  interactions for a flow field made up of  $N$  particles, which can lead to a large computational cost. A number ways of reducing the number of interactions in this calculation have been reported by various researchers. In this section, techniques aimed at improving the efficiency of the algorithm are presented. A brief review of recent research is given, highlighting some of the various methods that may be used to give a "fast" velocity calculation.

### **2.2 Vortex in Cell Method.**

The Vortex in Cell (VIC) method (or sometimes Cloud in Cell method) aims to reduce the operation count of the velocity calculation, by combining elements of both the Lagrangian and Eulerian approaches. A fixed mesh is fitted over the domain of the flow field, with the vortex particles representing the discretised vorticity field flowing through the mesh in a Lagrangian manner. The key to the reduction in the operation count, lies with the use of the mesh to calculate the velocity field. The vorticity is initially interpolated onto the mesh nodes and the velocity at each node can be calculated from the stream function found by solving the Poisson equation. The velocity of each vortex is then found by interpolating the nodal values of the velocity to each vortex location.

The basic method in its simplest form is implemented as follows. The vorticity of the  $n$ th vortex in any given cell is interpolated to the four nearest nodes, using an area weighting scheme,

$$\omega_i = \Gamma_n A_i / A^2 \quad i = 1, 2, 3, 4 \quad (2.1)$$

where  $\Gamma_n$  is the strength of the vortex and  $A$  is the area of the cell (Fig. 2.1). When this interpolation has been completed for all vortices in all cells, Poisson's equation (2.2) is solved to obtain the stream function,  $\Psi_{i,j}$  at all mesh nodes  $(i,j)$ . The velocity components are then calculated from a simple differencing scheme (2.3).

$$\nabla^2 \Psi = -\omega \quad (2.2)$$

$$u_{i,j} = (\Psi_{i,j+1} - \Psi_{i,j-1}) / 2M \quad (2.3a)$$

$$v_{i,j} = -(\Psi_{i+1,j} - \Psi_{i-1,j}) / 2M \quad (2.3b)$$

where  $M$  is the cell dimension. The velocity of the  $n$ th vortex in the cell is then found from,

$$u_n = \sum u_i A_i / A \quad , \quad v_n = \sum v_i A_i / A \quad (2.4)$$

For a flow field containing  $N$  particles and a mesh containing  $M$  nodes, the VIC method reduces the operation count to approximately  $O(N+M \log M)$ . The VIC method is discussed in more detail in [4-7].

Clearly, the accuracy of the method is directly related to the level of refinement of the mesh, the specification of which is a significant problem for the method. The easiest method would be to create a fixed, uniform grid at the start of the calculation, with the required definition. However, in areas where the vortex particles are sparse, unnecessary calculations are being performed to interpolate the velocity at the large number of nodes in regions where the velocity gradients are small. The alternative, is to use a fine definition grid only in regions where the vortex particles are densely clustered. In this case, due to the convection of vorticity as the calculation progresses, a new mesh will have to be created for each iteration to ensure the required definition is obtained. The overhead that this Eulerian part of the calculation carries needs careful consideration. A poor approach in overcoming these problems could mean that the reduction in operation count could be disappointingly small.

The simplest version of the VIC method is termed the Particle-Mesh (PM) method. In this case, both the near and far field contributions to the velocity of any particle are calculated from the velocity field calculated at the mesh nodes. Although this approach is very simple, it is unacceptable for modelling the interaction of particles that are clustered closely



together. The reason for this is that only velocity variations greater than the mesh spacing can be obtained and any small scale vortical structures smaller than this spacing will not be successfully modelled.

The Particle-Particle/Particle-Mesh (P<sup>3</sup>M) method is a simple way of reducing this problem. In this case, the far field contribution to the velocity are found from the nodal values as in the PM method. However, the influence of vortices close to the particle of interest are found by direct summation. Typically, if the distance between any two particles is less than  $3\Delta x$ , where  $\Delta x$  is the grid spacing, then direct summation is used (Fig. 2.2). Hence, the smaller scale vortex interactions and structures can be modelled, but still gaining the benefits of the VIC method. The disadvantage of P<sup>3</sup>M is that it can very easily be dominated by the direct summation terms and so the improvement in the operation count is compromised as accuracy is improved.

The main drawback of the VIC method is that some of advantages of the vortex method are lost with the need to fit a Eulerian mesh to the flow field. The interpolation to and from the mesh will include areas of the flow where there is little or no vorticity. Also, the interpolations will introduce some numerical diffusion into the solution. LEONARD [4] reports that researchers have found that "numerical experiments ... indicate that although these fine scale errors are present they may not seriously affect the large-scale features." SMITH et al. [5] report good results using the VIC method on a circular cylinder, although claim that for an accurate calculation, a large number of particles is required.

### **2.3 Method of Local Corrections.**

The method of local corrections, was proposed by Anderson [8], and is a variation on the P<sup>3</sup>M discussed above. The method is based on the observation that the difference between the velocity induced by a point vortex and a vortex blob is very small at large distances from the vortex centre. From this observation, a velocity field is calculated from a distribution of point vortices, and is then "locally corrected" about the centre of each vortex. A more detailed description of the method is given in [7-9].

The first stage of the method is to calculate the velocity field due to a distribution of point vortices and then to interpolate the velocity obtained to the centre of each vortex. The velocity calculation is based on the observation that the velocity field due to a point vortex

(2.5) is harmonic away from the centre of the vortex, where (2.6) is the velocity field due to a point vortex.

$$\nabla^2 \bar{\mathbf{u}}_j = 0 \quad \text{for } \bar{\mathbf{x}} \neq \bar{\mathbf{x}}_j \quad (2.5)$$

$$\bar{\mathbf{u}}_j = \frac{\Gamma_j (\bar{\mathbf{x}} - \bar{\mathbf{x}}_j)}{2\pi \|\bar{\mathbf{x}} - \bar{\mathbf{x}}_j\|} \quad (2.6)$$

Then for a given grid with a mesh spacing  $h$ , an  $r$ th order approximation to the Laplacian on this grid can be denoted by  $\nabla^{2h}$ . Rather than solving for the stream function,  $\Psi$ , at the grid nodes (2.2), Anderson solves a Poisson equation for the velocity field, using (2.7)

$$\nabla^2 \bar{\mathbf{u}}^h = \sum_j \mathbf{g}_{D_j} \quad (2.7)$$

where the right hand side is the sum of the approximations to  $\nabla^2 \mathbf{u}$  on the grid,

$$\mathbf{g}_{D_j}(\bar{\mathbf{x}}) = \begin{cases} \nabla^{2h} \bar{\mathbf{u}}_j & |\bar{\mathbf{x}} - \bar{\mathbf{x}}_j| < D \\ 0 & |\bar{\mathbf{x}} - \bar{\mathbf{x}}_j| \geq D \end{cases} \quad (2.8)$$

Since  $\nabla^2 \bar{\mathbf{u}}$  is harmonic (2.5), a value of  $D$  can be selected so that  $\mathbf{g}_D$  is an  $r$ th order accurate approximation to  $\nabla^2 \bar{\mathbf{u}}_j$  on the grid. The constant  $D$  is termed the spreading radius. The reason for using the approximation  $\mathbf{g}_D$  to  $\nabla^2 \bar{\mathbf{u}}$ , is that for a small value of  $D$ , the right hand side of (2.7) can be evaluated in  $O(N)$  operations rather than the  $O(NM)$  operations required to calculate  $\nabla^2 \bar{\mathbf{u}}$  at each of the  $M$  grid points. By now solving the Poisson equation (2.7), the velocity field at the grid nodes can be found.

The interpolation of the velocity back to the vortex centres also differs from the normal P<sup>3</sup>M method. Anderson exploits the fact that in two dimensions, away from the centre of the vortex, the two velocity components form the real and imaginary parts of a complex analytic function, given by

$$F(z) = u_1(z) - iu_2(z) \quad (2.9)$$

The usefulness of using the complex representation of the velocity field is that an interpolation formula for the velocity can be found by taking the real and imaginary parts

of any interpolation formula for complex functions. Thus, one can use Lagrange interpolation in the complex plane.

$$F(z) \cong \sum_{j=1}^n \left( \prod_{i \neq j} \frac{(z - z_i)}{(z_j - z_i)} \right) F(z_j) \quad (2.10)$$

for  $n$  points in the complex plane,  $z_i, i=1, n$ .

The advantage of using this interpolation scheme is that for a mesh cell with 4 nodes, a fourth order accurate interpolation can be obtained for the velocity in that cell. This should be contrasted to bilinear interpolation applied to each velocity component separately, which only achieves second order accuracy. It should also be noted that this interpolation scheme does not generalise very easily into three dimensions.

The last stage of the calculation is the local correction to the velocity of each vortex. The interpolation formula described above is not very accurate near the centre of the vortex due to the singular nature of the point vortices. However, this error can be removed by applying a local correction to the velocity of each vortex. This correction consists of subtracting the velocity component which is influenced by the nearby point vortices, and adding the correct (or exact) velocity influenced by the nearby vortex blobs (2.11)

$$\begin{array}{l} \text{velocity} \\ \text{correction for } j\text{th} \\ \text{vortex} \end{array} = \begin{array}{l} \text{exact velocity} \\ \text{contribution due} \\ \text{to vortices such} \\ \text{that} \\ |x_i(t) - x_j(t)| \leq C \end{array} - \begin{array}{l} \text{interpolated velocity} \\ \text{contribution due to vortices} \\ \text{such that} \\ |x_i(t) - x_j(t)| \leq C \end{array} \quad (2.11)$$

This correction step only requires  $O(N)$  operations for a field containing  $N$  vortices. As the discrete Laplacian step in (2.7) can be solved in  $O(M \log M)$  operations for a grid of  $M$  nodes, then the whole velocity calculation is clearly  $O(M \log M + N)$  operations.

Anderson reports that very accurate velocity calculations can be obtained with reasonably small values of  $D$  (approximately  $3h$ , where  $h$  is the grid spacing). Also, it is claimed that the method gives accurate results and preserves the effects of using higher order accurate vortex blobs. The method has the same operation count as the VIC methods discussed above and gives considerably more accurate results, but is considerably more complex to program.

## 2.4 Tree Codes, Multipole Expansions and the Fast Multipole Method.

Tree codes and multipole expansions utilise very different techniques to the VIC methods described above. The basic philosophy of these methods is to decompose the flow field into a series of zones, each containing a cluster of vortex particles. Provided that a zone is sufficiently far from the point  $z$ , at which the velocity is being calculated, then the contribution of the particles in the zone to the velocity, can be found by a Laurent series expansion (2.12) rather than from direct summation of all the particles.

$$U(z) - iV(z) = \sum_k \frac{a_k}{2\pi i(z - Z_c)} \quad (2.12)$$

where  $a_k$  are the coefficients of the series expansion and  $Z_c$  is the centre of the zone.

The advantages of this technique are that firstly, the Lagrangian nature of the vortex method is retained and secondly, accurate results can be obtained with a significant reduction in the operation count. The main problems with VIC methods are avoided, by the fact that no interpolation of the vorticity or velocity field to or from a mesh is required. However, as the method is based on an infinite series expansion, some error is introduced due to the truncation of this series. This error can be minimised to an almost insignificant level, by the ensuring that the series expansion is used for points sufficiently far away, to ensure that the series converges. Judicious choice of the order of the series can also help to minimise the error.

The implementation of the method varies from application to application and full descriptions of some of these methods are given in [7,10-14]. Typically, multipole expansions give an operation count of  $O(N \log_4 N)$  for a flow field of  $N$  particles. The Fast Multipole Method (FMM) of GREENGARD and ROKHLIN [11] is claimed to give an operation count of  $O(N)$ . It is a tree code method, utilising a zonal decomposition and multipole expansions for the velocity influence of a zone, that has been incorporated into the DVM. This algorithm will be presented in more detail later. However, it does have distinct differences to the FMM, a brief description of which will now be given for comparison purposes.

### 2.4.1 The Fast Multipole Method Algorithm.

The first stage of the method is to divide the flow field into a hierarchical structure of zones. The initial zone is a square that contains all of the vortex particles and is called level 0. This initial box is subdivided into four smaller square zones of equal size, and these new zones are at level 1. In general, level  $l$  is obtained by subdividing each zone in level  $l-1$  into four smaller zones, and at each level  $l$  there will be  $4^l$  zones (Fig. 2.3). Further subdivision of each zone takes place until the required level of refinement, level  $k$ , is reached. Given a zone  $A$  at level  $l-1$ , the zones at level  $l$  that are obtained by subdividing  $A$  are termed the *children* of  $A$ , and a zone is termed the *parent* of its children. This zonal decomposition of the flow field is best suited to a uniform flow field. An "Adaptive" decomposition, where subdivision of a zone only takes place if there are greater than some minimum number of particles in a zone, is presented in [13] and will not be discussed here.

The FMM uses an interaction list for each zone. This is defined as follows and is demonstrated in Fig. 2.4 :

- 1) Two zones are *neighbours* if they are of the same size and share a boundary point. (Note that a zone is classed as one of its own neighbours).
- 2) Two zones are *well separated* if they are of the same size and are not near neighbours.
- 3) Each zone,  $A$ , has an associated *interaction list*, which consists of all the children of the neighbours of  $A$ 's parent, that are well separated from  $A$ .

Other terms used in the description of the algorithm are :

*Multipole expansion* - the series expansion (with  $p$  terms) about the zone centre, that gives the velocity field created by the particles contained within the zone.

*Local expansion* - the series expansion (with  $p$  terms) about the centre of zone  $i$  at level  $l$ , describing the velocity field due to all particles outside zone  $i$  and its nearest neighbours.

Consider a zone  $A$  with centre  $z_A$  and radius  $R$  (usually half of the zone width) containing  $n_p$  particles, located at points  $z_j$ , such that  $|z_j - z_A| < R$  for all  $j$  (Fig. 2.5). Then for a point  $z$  where  $|z_j - z_A| > 2R$ , the velocity influence of zone  $A$  at point  $z$  can be expressed by a multipole expansion about zone centre  $z_A$  (2.13)

$$U(z) - i V(z) = \sum_{k=0}^{\infty} \frac{a_k}{(z - z_A)^{k+1}} \quad (2.13)$$

where the coefficients  $a_k$ , of the expansion are given by

$$a_k = \frac{1}{2\pi i} \sum_{j=1}^{n_p} \Gamma_j (z_j - z_A)^k \quad k = 0, 1, 2, \dots, \infty \quad (2.14)$$

The coefficients (2.14) can be calculated for each zone as the zonal decomposition is performed, or calculated for the zones at the lowest level only, and these coefficients used to calculate the coefficients of each parent zone.

The FMM method then performs a downward pass through the zonal hierarchy (largest zones to smallest). For zone  $A$  the multipole expansion for each zone in  $A$ 's interaction list is converted to a local expansion about the centre of  $A$ . The coefficients of the local expansion is shifted to the centres of  $A$ 's children. This process is repeated until the lowest level is reached. In general, each zone  $A$  at level  $l$ , will have coefficients of a local expansion that represents the velocity field due to all vortices in zones that are well separated from  $A$ 's parent. These are added to the coefficients of the local expansions from the zones in  $A$ 's interaction list. At the lowest level, each zone will have a local expansion the represents the velocity field due to all zones except its nearest neighbours. The velocity influence of the neighbouring zones will be evaluated from direct summation.

A more detailed discussion of the FMM along with the mathematical proofs and techniques for creating and shifting the centres of the local expansions is given in [11]. The total cost of the calculation is  $O(N)$ , due to the zone-zone interactions rather than the particle-particle interactions of the direct summation. As mentioned above, an adaptive FMM algorithm is presented in [13] and a parallelised version, giving further improvements in the operation count, is given in [12].

### **3.0 IMPLEMENTATION OF A ZONAL DECOMPOSITION ALGORITHM IN THE VORTEX METHOD.**

The zonal decomposition technique was chosen as the method with which to improve the efficiency of the DVM. The choice of this technique over a VIC method was primarily based on the fact that the Lagrangian nature of the method, and the benefits this brings, is retained. The technique utilised in the DVM uses an adaptive zonal decomposition using square zones. The series expansion for the velocity influence from a zone is similar in form to the FMM, although the interactions between zones and particles are handled very differently. This leads to a much simpler algorithm for the velocity calculation, without any great loss in accuracy when compared to direct summation. The algorithm will now be presented in more detail.

#### **3.1 Decomposition of Flow Field into a Hierarchical Structure of Square Zones.**

As with many other applications, it was decided to use square zones in the decomposition of the flow field. CLARKE et al. [10] use rectangular zones in the zonal decomposition and divide each zone into two, along the longest side, such that each sub-zone contains half the particles of the parent zone. This method was investigated, as it leads to less "dormant" zones that contain no particles. However, it was found that some zones with high aspect ratio were created, which can lead to a large radius around the zone. Consequently there is less chance of using the zonal influence in the velocity calculation (Fig. 3.1). This extra use of the slower direct summation negates the benefit of eliminating the "dormant" zones in the algorithm.

The DVM discretises the vorticity field into two sets of vortex particles, nascent (contained in a small control zone close to the body) and wake [1-3]. These two sets are joined together into a single set of particles, with the zonal decomposition being performed on this combined set of particles. The initial zone is the smallest square that contains all the particles in the flow field. If there are greater than some predetermined number of particles in the wake region,  $NP_{min}$  say, then this initial zone can be subdivided into four smaller zones in the order shown in Fig. 3.2. Each of the children of the initial zone are subdivided if they contain greater than  $NP_{min}$  particles. The children of these new zones are then subdivided, where appropriate, until no further subdivision can take place. This procedure

is illustrated in Fig. 3.3, along with the resulting hierarchical zonal structure for a typical flow field.

This procedure is similar to that described by VAN DOMMELEN et al. [14], although a much simpler method of numbering the zones is used here. The initial zone containing the flow field is numbered zone 0. The four subzones of zone 0 are zones 1 to 4 respectively. If zone 1 can be subdivided, then its children will be zones 5 to 8. In general, the children of a zone  $k$  will be numbered  $Nz+1$  to  $Nz+2$ , where  $Nz$  zones have previously been created. As the children of each zone are always created in the same order (Fig. 3.2), only the *first* child is recorded when a zone is subdivided. Also, when each zone is created, its parent zone will be recorded. Using this numbering scheme, for any zone, its parent and all of its children can easily be traced. The resulting zonal decomposition, of all the vortex particles in the flow field, provides an hierarchical structure of zones, that can be used as part of a fast algorithm for the velocity calculation in the Vortex Method.

### 3.2 Velocity Calculation using Zonal Decomposition and Series Expansion.

As discussed previously (section 2.4), the velocity influence of a group of particles contained within a zone can be used, if the velocity is being calculated at a point sufficiently far from the centre of the zone. It is usual to define "sufficiently far" as some specified multiple of the zone radius, where the radius is half the side length of the zone (Fig. 3.4). The velocity influence of a zone, at a point  $z$  can be calculated from a truncated series expansion (3.1 - 3.2).

$$U(z) - iV(z) = \sum_{k=1}^{Nt} \frac{a_k}{(z - z_A)^k} \quad (3.1)$$

$$a_k = \frac{1}{2\pi i} \sum_{j=1}^{Np} \Gamma_j (z_j - z_A)^{k-1} \quad k = 1, 2, 3, \dots, Nt \quad (3.2)$$

where  $Nt$  is the number of terms in the series expansion,  $z_A$  is the zone centre,  $Np$  is the number of particles in the zone, and  $z_j$  and  $\Gamma_j$  are the position and circulation respectively, of vortex  $j$  in the zone. The derivation of these formulae is given in Appendix A. The coefficients  $a_k$  of the series expansion for each zone, are calculated as the zonal decomposition is being performed.



It should be noted that the flow field is only decomposed into the zonal structure if greater than some predetermined number of wake particles are present in the calculation. If there are less than this limit, then the velocity calculation is performed using direct summation.

If a zone contains  $N_p$  particles, then the operation for the velocity influence at a point is  $O(N_p^2)$  using direct summation, compared to  $O(Nt)$  using zonal decomposition. It is clear that the larger  $N_p$  is, then the greater will be the reduction in the operation count for the velocity calculation. For this reason, the largest possible zone is always used in the calculation, to give the maximum improvement in the speed of the algorithm.

The implementation of the algorithm will now be demonstrated by considering the calculation of the velocity for one vortex particle. First consider the distance of the particle from the first zone (zone 1). If the particle is greater than twice the zone radius from the zone centre, then the particle is sufficiently far from the zone, and the influence of the zone can be used in the velocity calculation. However, if the particle is less than the required distance and the zone has no children, then the velocity influence of the zone is calculated from direct summation. If the zone however does have children, then each of the children are considered in the same manner as described above. The procedure is repeated until either a zonal influence can be used, or the lowest level of decomposition is reached, and direct summation is used. If the particle is contained within the zone, and the lowest level has been reached, then the zone's contribution is calculated using direct summation later in the algorithm. Otherwise the zone's children are considered as usual. This continues until all the children and "grandchildren" of zone 1 have been considered. The procedure is then repeated on the remaining zones at the top level (zones 2-4). This algorithm is summarised in the flowchart shown in Fig. 3.5. This procedure is then repeated for the next vortex particle.

The procedure described above, gives the velocity influence on a particle, of all the particles outside of the zone at the lowest level, which contains the particle (Fig. 3.6). Also, as discussed above, the largest possible zones are used to give the velocity influence, and are found by considering the least number of possible zones. The final part of the algorithm is a single pass through each of the childless zones, so that the velocity influence of the particles contained in the zone on each other, can be calculated using direct summation. This provides the final contribution to the velocity of each particle.

### **3.3 Comparison of Algorithm with Fast Multipole Method.**

The FMM algorithm described in section 2.4.1 uses a zone vs. zone interaction, and gives an operation count of  $O(N)$  [11]. The algorithm implemented in the Vortex Method and described above uses a particle vs. zone interaction, and gives an operation count of  $O(N+N\log N)$  (Appendix B). The reason for the use of an apparently slower algorithm, was primarily due to the simpler implementation of the method. Also, the shifting of the multipole expansions to local expansions in the FMM, can introduce an extra source of error into the calculation, as well as that incurred by the truncation of the series expansion.

As discussed later in section 4, the algorithm gives very satisfactory results in terms of the calculation efficiency. Typical calculations of the Vortex Method use around 6000 particles, and the extra efficiency of the  $O(N)$  FMM only becomes more significant when  $N$  is much greater than this. For this reason, even though the FMM algorithm was investigated, it was decided to implement the simpler, if slightly slower, algorithm. Future work on further improvements to the operation count may include a re-evaluation of the FMM method.

## 4.0 RESULTS.

### 4.1 Performance of Algorithm.

Three different test cases were used to assess the performance of the zonal decomposition algorithm. Each test case involved a very different body geometry and resulting flow field. The test cases were as follows :

1. Static aerofoil at 40 degrees incidence.
2. Circular cylinder.
3. Square cylinder at 0 degrees incidence.

In each cases, the calculations were performed first using the original velocity calculation via direct summation and then using the zonal decomposition algorithm. The calculations were performed on a Silicon Graphics workstation with a 150MHz IP22 R4400 processor, 16Kb cache size and 64Mb main memory size (Dept. of Aerospace label : ATLAS). The performance of each calculation was obtained by outputting the CPU time taken for the whole timestep as well as the CPU for the velocity calculation. The results are shown in Figs. 4.1 to 4.9 and show the CPU plotted against the total number of vortices (wake and nascent combined) for the timestep and velocity calculation respectively. Typical decomposition of the flow field into the hierarchical zonal structure are shown for both the square and circle models in Figs. 4.10 and 4.11.

Comparing first the velocity calculation only for both the direct summation and zonal decomposition algorithms, it is clear for each method that the relationship between the CPU and the number of vortices is very similar in each of the three test cases, especially when direct summation is used (Fig. 4.4). As expected, this shows that the calculation is dominated by the number of vortices contained in the flow field rather than their physical location within the flow field. There is slightly more variation in the zonal decomposition case, as the location of particles will determine whether the series expansion can be used for the velocity calculation. However, a strong relationship between the CPU and number of vortices is clear, irrespective of which model is used.

The operation count for the direct summation method is  $O(N^2)$  as discussed above. For the zonal decomposition, analysis of the operation count is a little more complex and is shown to be  $O(N+N\log N)$  in Appendix B. A least squares curve fit has been fitted to the CPU timings for both direct summation and zonal decomposition, using the operation counts given above. The curve fits, and the derived constants are shown in Figs. 4.5 and 4.6.

Table 4.1 shows the improvement obtained by the zonal decomposition algorithm for the CPU required for one timestep, based on the curve fits to the data.

Number of Vortices.	CPU from Direct Summation. (secs.)	CPU from Zonal Decomposition. (secs.)	Factor Improvement.
1000	2.041	1.381	1.478
2000	8.164	3.638	2.244
3000	18.370	6.225	2.951
4000	32.658	9.027	3.618
5000	51.028	11.988	4.257
7500	114.812	19.904	5.768
10000	204.110	28.356	7.198
20000	816.440	65.470	12.470
30000	1836.990	105.890	17.348

**Table 4.1**

The factor improvement for the whole timestep is also shown in Fig. 4.7, and is compared with the factor improvement in the velocity calculation only. An extra saving was made in the timestep by using the zonal structure as part of the vortex merging calculation [2]. Instead of checking against any vortex in the flow field to find if the merging criteria is satisfied, which can also lead to an  $O(N^2)$  calculation, only vortices within the same zone at the lowest level are checked. The CPU taken for the merging calculation using the new and old methods is shown in Fig. 4.8. The breakdown of the timestep into various elements of the calculation is shown in Fig. 4.9. It is clear that the main element of each timestep is the velocity calculation. As discussed above, the merging calculation now uses little CPU. Importantly, the zonal decomposition and sorting of the vortices also takes a relatively small amount of CPU.

The speed and accuracy of the algorithm can be optimised by judicious selection of a number of parameters that define the zonal decomposition. The optimisation of these parameters is discussed in Appendix C.

#### **4.2 Error Introduced by the Zonal Decomposition Algorithm.**

The infinite series expansion that is used in the velocity calculation for the zonal decomposition algorithm gives the correct velocity as shown in Appendix A. However, for practical implementation of the method, the series needs to be truncated after a reasonable number of terms have been evaluated. This truncation introduces an error into the calculation when the contribution of a zone is used. As discussed in Appendix C, the algorithm has been optimised to attempt to reduce this error, whilst still retaining the calculation efficiency.

By comparing the velocity calculation of a sample flow field using the zonal decomposition and direct summation, estimates of this error can be obtained. The test case was on a circular cylinder, using a developed flow field similar to that shown in Fig. 4.11, and comparing the results over a single timestep. The velocity magnitudes of all the vortices (nascent and wake) was compared. The maximum error was  $9.8791e-4$ , and the rms. error was  $1.1546e-4$  compared to the direct summation results. The percentage error is shown in Fig 4.12. In general, the error is very small, and it is clear that the larger percentage errors occur for vortices where the velocity is very small, and arise due to some ill-conditioning when dividing by the velocity from direct summation. Fig. 4.13 shows a comparison of the flow field around a circular cylinder after the first 200 timesteps of the calculation, using both the direct summation and zonal decomposition algorithms. Although there are differences between the two flow fields, in general they are very similar, showing that despite the errors now introduced into the calculation, the new algorithm gives comparable results to the original method.

It should be noted that the calculation of the velocity using the zonal influence via a series expansion, is derived from assuming that the vortices in the zone are point vortices (Appendix A). However, the vortex method uses vortex blobs, where a core function is implemented, to avoid singularities arising from point vortices [2]. Clearly, there is an inconsistency here that may be a further source of error. Long distances from the vortex locations, the velocity influence of a vortex blob and a point vortex is very similar. This is also discussed in section 2.3, for the method of Local Corrections. This method is "... based upon the observation that the difference between the velocity field due to a point vortex and a vortex blob located at the same point in space becomes very small as one moves away from the centres of the vortices" [8]. As the zonal influence is only used at

"long" distances from the zone centre, it is clear that the error arising from assuming that the vortices contained in the zone are point vortices, is small.

No reference has been found in the literature as to how the zonal contribution could include the effects of modelling vortex blobs rather than point vortices. Solving this problem could be an area of future research, to give a more accurate, but efficient velocity calculation for a distribution of vortex blobs. The error value given above is comprised of both sources of error that have been described in this section.

### **4.3 Future Work.**

The saving that has been obtained using this algorithm is of the order that was being sought. Typical calculations can now be performed in a matter of hours rather than days. The benefits of this are clear, better turn around of calculations, each calculation can utilise more timesteps and more particles may be used without the calculation becoming prohibitive. However, further improvements in the operation count may be achievable. Parallelisation of the code is an obvious method to be considered for further savings. Also, a more detailed investigation of the FMM may yet yield future benefits, with the possibility of an operation count of  $O(N)$  rather than the  $O(N+N\log N)$  which has been achieved. Any future research on improving the calculation efficiency of the vortex method, should be focused primarily on these areas.

## **5.0 Conclusions.**

A zonal decomposition algorithm has successfully been incorporated into the Discrete Vortex Method. The algorithm decomposes all of the particles in the flow field into a hierarchical structure of square zones. The velocity influence of a zone may then be used in the velocity calculation using a truncated series expansion.

A zone vs. particle interaction has been used with the largest possible zone always being used for maximum improvement in the operation count. Direct summation is used for zones that are close to the particle. The algorithm has been shown to have an operation count of  $O(N+N\log N)$ .

Typically, an improvement of a factor of 4 to 5 has been obtained compared to the direct summation. Added savings have been made with modifications to the vortex merging calculation. The error due to the truncated series expansion has been minimised.

Future improvements could be made by parallelisation of the code. A more detailed investigation of an  $O(N)$  algorithm should be made if a more efficient algorithm is required.

**REFERENCES.**

- 1) VEZZA, M., 1992, "A New Vortex Method for Modelling Two-Dimensional, Unsteady Incompressible, Viscous Flows." G.U. Aero Report No. 9245.
- 2) LIN, H and VEZZA M, 1994 "Implementation of a Vortex Method for the Prediction of Separated Incompressible Flows." G.U. Aero Report No. 9425
- 3) TAYLOR, I.J., 1996, "Development and Validation of a Discrete Vortex Method for the Prediction of Separated Incompressible Flows Around Bluff Bodies." G.U. Aero Report No. 9622.
- 4) LEONARD, A., 1980, "Vortex Methods for Flow Simulation." Journal of Computational Physics, Vol. 37, pp. 289-335.
- 5) SMITH, P.A. and STANSBY, P.K., 1988, "Impulsively Started Flow Around a Circular Cylinder by the Vortex Method." Journal of Fluid Mechanics, Vol. 194, pp. 45-77.
- 6) SARPKEYA, T., 1989, "Computational Methods with Vortices - The 1988 Freeman Scholar Lecture." Journal of Fluids Engineering, Vol. 111, pp. 5-52.
- 7) PUCKETT, E.G., 1993, "Vortex Methods : An Introduction and Survey of Selected Research Topics." Incompressible Computational Fluid Dynamics, ed. M.D. Gunzb and R.A. Nicolaides, Cambridge University Press.
- 8) ANDERSON, C.R., 1986, "A Method of Local Corrections for Computing the Velocity Field Due to a Distribution of Vortex Blobs." Journal of Computational Physics, Vol. 62, pp. 111-123.
- 9) BADEN, S.B, 1987, "Very Large Vortex Calculations in Two Dimensions." Lecture Notes in Mathematics 1360 - Vortex Methods, (ed. C. Anderson and C. Greengard), pp. 96-120



- 10) CLARKE, N.R. and TUTTY, O.R., 1994, "Construction and Validation of a Discrete Vortex Method for the Two-Dimensional Incompressible Navier-Stokes Equations." *Computers Fluids*, Vol. 23, No. 6, pp. 751-783.
- 11) GREENGARD, L. and ROKHLIN, V., 1987, "A Fast Algorithm for Particle Simulations." *Journal of Computational Physics*, Vol. 73, pp. 325-348.
- 12) GREENGARD, L. and GROPP, W.D., 1990, "A Parallel Version of the Fast Multipole Method." *Computers Math. Applic.*, Vol. 20, No. 7, pp. 63-71.
- 13) CARRIER, J., GREENGARD, L. and ROKHLIN, V., 1988, "A Fast Adaptive Multipole Algorithm for Particle Simulations." *SIAM JI Sci. Stat Comput*, Vol. 9, pp. 669-686.
- 14) VAN DOMMELEN, L. and RUNDENSTEINER, E.A., 1989, "Fast, Adaptive Summation of Point Forces in the Two-Dimensional Poisson Equation." *Journal of Computational Physics*, Vol. 83, pp. 126-147.
- 15) PRINGLE, G.J., 1994, "Numerical Study of Three Dimensional Flow using Fast Parallel Particle Algorithms." PhD. Thesis, Department of Mathematics, Napier University, Edinburgh.

### Appendix A : Derivation of Series Expansion for Zonal Decomposition.

Consider a zone  $D$ , containing  $N_p$  point vortices, with strengths/circulations  $\Gamma_k$ , at positions given by the complex co-ordinates  $z_k = x_k + iy_k$ ,  $k=1, 2 \dots N_p$ . The velocity  $\mathbf{U}(z)$  induced by the particles contained in  $D$ , at an arbitrary point,  $z = x + iy$ , outside zone  $D$ , is obtained from the Biot-Savart law using,

$$\mathbf{U}(z) = \frac{i}{2\pi} \sum_{k=1}^{N_p} \frac{(z - z_k) \Gamma_k}{|z - z_k|^2}$$

Now  $\mathbf{U}(z) = u + iv$ , where

$$u = -\frac{1}{2\pi} \sum_{k=1}^{N_p} \frac{(y - y_k) \Gamma_k}{|z - z_k|^2}, \quad v = \frac{1}{2\pi} \sum_{k=1}^{N_p} \frac{(x - x_k) \Gamma_k}{|z - z_k|^2} \quad (\text{A.1})$$

Taking the conjugate of the velocity and substituting from A.1, we can write

$$\begin{aligned} \mathbf{U}^*(z) &= u - iv \\ &= -\frac{1}{2\pi} \sum_{k=1}^{N_p} \frac{\Gamma_k}{|z - z_k|^2} [(y - y_k) + i(x - x_k)] \\ &= -\frac{1}{2\pi i} \sum_{k=1}^{N_p} \frac{\Gamma_k}{|z - z_k|^2} [i(y - y_k) - (x - x_k)] \\ \Rightarrow \mathbf{U}^*(z) &= \frac{1}{2\pi i} \sum_{k=1}^{N_p} \frac{\Gamma_k}{|z - z_k|^2} (z - z_k)^* \end{aligned} \quad (\text{A.2})$$

Using  $|z|^2 = z \cdot z^*$  for any complex  $z$ , then (A.2) can be rewritten

$$\mathbf{U}^*(z) = \frac{1}{2\pi i} \sum_{k=1}^{N_p} \frac{\Gamma_k}{(z - z_k)} \quad (\text{A.3})$$

Taking some arbitrary point,  $Z_m$ , within  $D$  (centre of  $D$  say), and rewrite the positions of all the vortex particles in  $D$  relative to  $Z_m$ .

$$\text{i.e. } z_k' = z_k - Z_m$$

then substituting in (A.3)

$$\begin{aligned} \Rightarrow U^*(z) &= \frac{1}{2\pi i} \sum_{k=1}^{Np} \frac{\Gamma_k}{(z - z_k' - Z_m)} \\ &= \frac{1}{2\pi i} \sum_{k=1}^{Np} \frac{\Gamma_k}{(z - Z_m)} \left( 1 - \frac{z_k'}{(z - Z_m)} \right)^{-1} \end{aligned} \quad (\text{A.4})$$

(\*) Provided  $z_k' < (z - Z_m)$  for all  $k$ , then (A.4) can be expanded as a power series of the form  $(1 - x)^{-1} = 1 + x + x^2 + x^3 + \dots + (-1)^n x^n + \dots$

$$\begin{aligned} U^*(z) &= \frac{1}{2\pi i} \sum_{k=1}^{Np} \frac{\Gamma_k}{(z - Z_m)} \left[ 1 + \frac{z_k'}{(z - Z_m)} + \left( \frac{z_k'}{(z - Z_m)} \right)^2 + \dots + \left( \frac{z_k'}{(z - Z_m)} \right)^{Nt} + R_{Nt} \right] \\ &= \frac{1}{2\pi i (z - Z_m)} \left[ \sum_{k=1}^{Np} \Gamma_k + \frac{1}{(z - Z_m)} \sum_{k=1}^{Np} \Gamma_k z_k' + \frac{1}{(z - Z_m)^2} \sum_{k=1}^{Np} \Gamma_k (z_k')^2 + \dots \right. \\ &\quad \left. \dots + \frac{1}{(z - Z_m)^{Nt}} \sum_{k=1}^{Np} \Gamma_k (z_k')^{Nt} + R_{Nt} \right] \\ \Rightarrow U^*(z) &= \frac{1}{2\pi i (z - Z_m)} \sum_{j=1}^{Nt} \frac{a_j}{(z - Z_m)^{j-1}} + R_{Nt} \end{aligned} \quad (\text{A.5})$$

where  $a_j = \sum_{k=1}^{Np} \Gamma_k (z_k - Z_m)^{j-1}$

and  $R_{Nt}$  is the error due to truncation of the infinite geometric series after  $Nt$  terms.

The above shows the derivation of the series coefficients and the zonal expansion formula, with (A.5) being equivalent to equations (3.1) and (3.2). Zonal decomposition is only used if a zone is sufficiently far from the point at which the velocity is being calculated. This ensures that the condition (\*) for the series expansion is satisfied.

## **Appendix B : Operation Count of Zonal Decomposition Algorithm.**

The analysis of the operation count of the zonal decomposition algorithm is very complex. Hence, to simplify the analysis, the case of  $N$  vortices homogeneously distributed over a square domain will be considered. The flow domain is continuously subdivided (as shown in Fig. 2.3) until the lowest level comprises of zones that contain  $n$  particles. The number of zones at the lowest level is given by  $N/n$ . The number of zones at any level of subdivision,  $l$ , for a uniform distribution of vortices and uniform subdivisions, is given by  $4^l$ . Therefore the number of levels of subdivision is given by

$$l = \log_4 \left( \frac{N}{n} \right) \quad (\text{B.1})$$

The total computational time required for one timestep in the calculation is dominated by the velocity calculation. The dominant two factors in the velocity calculation for a domain containing  $N$  particles, is from the contributions from direct summation and zonal decomposition. Only the operation count of these two factors will be considered to simplify the analysis further.

Consider the contribution first from direct summation. For any single vortex, the contribution from direct summation, will be due to at most, the 9 neighbouring zones (including the zone that contains the particle). As each zone contains  $n$  particles and there are  $N$  particles, then the total CPU required for the direct summation contribution is  $t_{ds}$ , where

$$\begin{aligned} t_{ds} &\propto 9nN \\ \text{i.e. } t_{ds} &= aN \end{aligned} \quad (\text{B.2})$$

where  $a$  is a constant that is derived from the number of zones contributing to the direct summation, the number of particles in each zone and the floating point operation speed of the processor.

For the contribution from series expansion due to the influence of distant zones, the analysis is more complex. Consider the worst case, where a series expansion is required to be evaluated at every zone at the lowest level, except the 9 neighbouring zones, i.e.  $(N/n-9)$  zones. The number of terms in the series expansion is  $Nt$ . To evaluate the series expansion at the lowest level, each level and each branch of the quadtree must be

evaluated. The number of operations to perform this calculation will be proportional to the number of levels,  $l$  as shown in (B.1). These are the main factors affecting the operation count of the zonal contribution to the velocity. Hence, the CPU required to evaluate the influence of distant zones on a single vortex using series expansion,  $t_{zd}$ , is approximately given by

$$t_{zd} \propto Nt \left( \frac{N}{n} - 9 \right) . cl$$

where  $c$  is a constant that is derived from the number of "traverses" around the quadtree. This expression is the worst case, where the series expansion is used for every zone at the lowest level, except the neighbours. In general, only  $p$  zones will be required, where  $p < (N/n-9)$ . The total CPU required for the zonal contribution is then given by

$$t_{zd} \propto N.Nt. p.c \log_4 \left( \frac{N}{n} \right)$$

$$t_{zd} \propto N.Nt. p.c \frac{\log \left( \frac{N}{n} \right)}{\log(4)} , \quad \text{using } \log_4(a) = \frac{\log(a)}{\log(4)} \quad (\text{B.3})$$

$$t_{zd} = Nb \log N \quad (\text{B.4})$$

where  $b$  is a constant that is derived from the number of zones whose series expansion is used, the number of terms in the series expansion and the floating point operation speed of the processor. Note that the dominant factor in the logarithm in (B.3) is the number of particles,  $N$  and so  $t_{zd}$  can be approximated using an operation count  $O(N \log N)$ .

Although this is a very simplified analysis of the zonal decomposition algorithm, it is clear that the total CPU required for the velocity calculation is approximately given by

$$t_{tot} = aN + bN \log N \quad (\text{B.5})$$

giving an algorithm that is of  $O(N+N \log N)$ . A curve of the form shown in (B.5), has been fitted to the CPU timings from the zonal decomposition algorithm, and is shown in Fig. 4.6, along with the derived constants. The curve is shown to give a reasonable fit to the data, giving some validation to the somewhat simplified analysis discussed above.

### **Appendix C : Optimisation of Parameters in the Zonal Decomposition Algorithm.**

The main parameters that affect the speed and accuracy of the algorithm are :

1.  $N_p$  - Minimum number of vortices in a zone, such that the zone can be further subdivided.
2.  $N_t$  - Number of terms in series expansion.
3.  $H$  - Series expansion can be used only when the distance of the vortex from the zone centre is greater than  $H$  multiplied by the zones radius.

These parameters have been optimised to obtain the best combination of calculation efficiency and accuracy. The results of this analysis are shown in Figs C.1-C.2. The results are very much as expected. Increasing  $H$  means that only distant zones can be used in the calculation, leading to a larger contribution to the velocity via direct summation and hence an increase in the CPU. However, as the zones used are distant, the series is more likely to converge giving a reduction in the error as  $H$  increases (Fig. C.1).

The CPU also increases as  $N_t$  increases, as more calculations are clearly required to evaluate more terms in the series expansion (Fig. C.2). As  $N_t$  increases, the series is more likely to converge and so the error is reduced. The series is also likely to converge much more quickly if the vortex is further from the zone, and this observation can be used to obtain extra efficiency in the calculation (PRINGLE [15]). For example, if a vortex is 4 times the zone radius from the zone centre, the series may converge in around 10 terms say, whereas a vortex that is only 3 times the zone radius from the zone centre, may require 15 terms. As a result, the algorithm uses a varying number of terms in the series expansion, depending on the distance of the vortex from the zone centre.

The variation of CPU and error with  $N_p$  is a little more complex. If  $N_p$  is small, there are a large number of small zones, each containing a relatively small number of vortices. Each velocity calculation will therefore require a contribution from a large number of zones, with the series expansion giving a small saving in operation count over direct summation as each zone contains a small number of particles. Hence, for small  $N_p$ , the CPU will be high. Also, the small zones mean that vortices will satisfy the distance from zone centre criteria, but will still be a small distance from the zone, and may affect the series convergence leading to an increase in the error. For large  $N_p$ , although when a zone is used in the velocity calculation a large saving in operation count is made, each vortex

will have to be further from the zone before the zonal contribution can be used. Hence, there is likely to be a larger contribution from direct summation and a resulting increase in CPU. However, this does mean that when series expansion is used, the series is more likely to be converged, and so the error is reduced. These effects can be observed in Fig. C.3.

The optimised code requires a balance to be struck between all of these parameters to obtain maximum efficiency for the desired calculation accuracy. The final optimised parameters are:

$$N_p = 200$$

$$H = 2.0$$

$N_t$  varying from 8 to 13 depending on the distance to the zone centre.

These parameters were used in the calculation results that are presented in Section 4.1

## NOMENCLATURE.

$A$	Area of mesh cell.
$a$	Coefficients of series expansion.
$D$	Spreading radius
$F$	Flow field.
$g_d$	Approximation to Laplacian.
$H$	Factor on zone radius used in zonal decomposition.
$h$	Mesh spacing.
$l$	Level of subdivision of zonal structure.
$M$	Number of nodes in mesh.
$M$	Cell dimension.
$N$	Number of vortex particles in flow field.
$N_p, n_p$	Number of particles in a zone.
$N_t$	Number of terms in the series expansion.
$P$	Pressure.
$R$	Zone radius.
$\vec{r}$	Position vector.
$S$	Body surface.
$t$	Time.
$U, V$	Velocity.

$u, v$	Velocity.
$\bar{\mathbf{U}}, \bar{\mathbf{u}}$	Velocity vector.
$\Delta x, \Delta y$	Grid spacing.
$\bar{\mathbf{x}}$	Position of vortex.
$x, y$	Components of position vector.
$Z, z$	Complex position of vortex particle.
$Z_c, Z_a,$	
$Z_m$	Centre of zone.
$\Gamma$	Circulation.
$\nu$	Kinematic Viscosity.
$\rho$	Fluid density.
$\bar{\Psi}$	Vector potential.
$\Psi$	Stream function.
$\omega$	Vorticity.
$\frac{D}{Dt}$	Lagrangian / material derivative.
$\nabla$	Gradient operator.
$\nabla^2$	Laplace operator.
$\nabla \times$	Curl.
$\nabla \cdot$	Divergence.

Subscripts.

$i$	Index number for solid body.
$i$	Index number for cell node.
$j$	Index number for vortex.
$k$	Index number for terms in series expansion.
$n$	Index number for vortex.
$p$	Point in flow field.
$\infty$	Far field.





Velocity of Vortex A is influenced by all other particles.

Similarly, velocity of Vortex B is calculated from all other particles.

In general, velocity of a particle will be influenced by the remaining  $N-1$  particles. Therefore, to calculate the velocity of all  $N$  particles will require  $N^2-N$  calculations. i.e Operation Count is  $O(N^2)$

Fig. 1.1 - Demonstration of  $O(N^2)$  Operation count for Velocity Calculation.

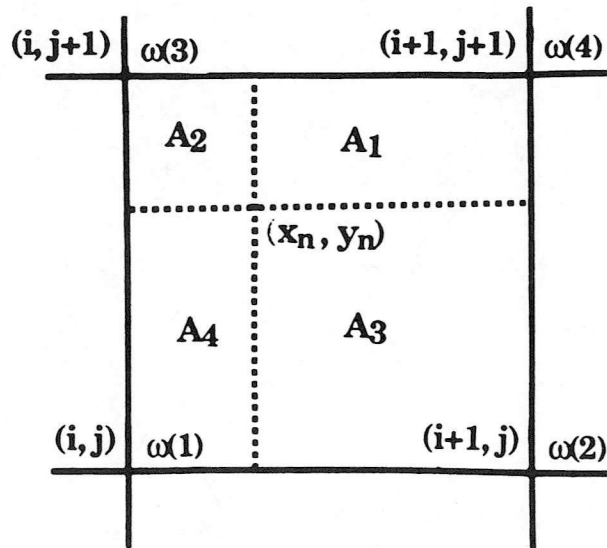
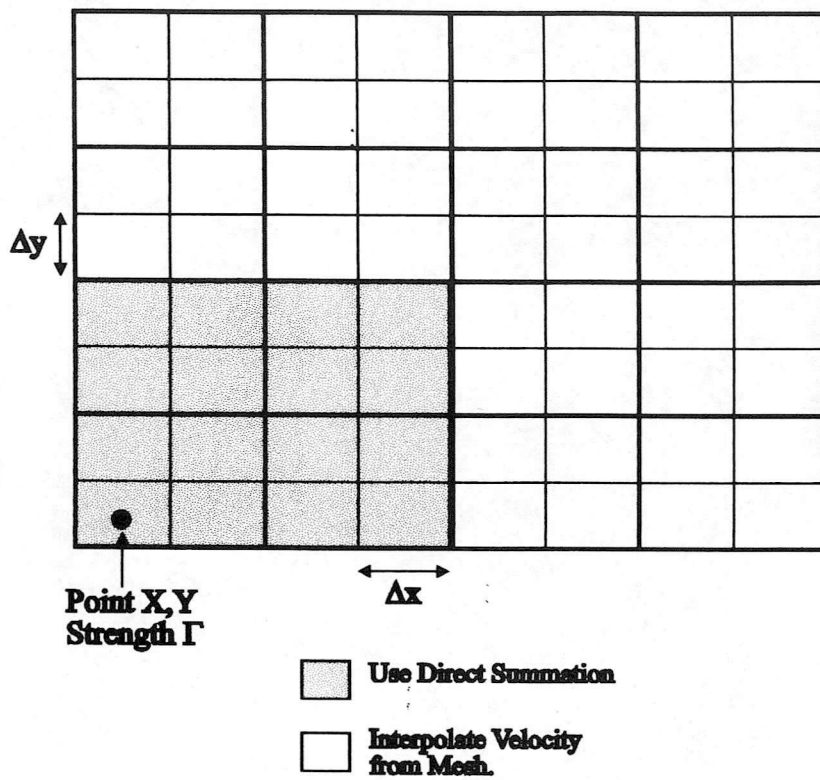


Fig. 2.1 - Area Weighting Scheme used in Vortex-in-Cell Method.  
( from SARPKEYA 1989 [3] )



Interpolation from Mesh used at distances  $>3\Delta x$  or  $3\Delta y$  from vortex

Fig. 2.2 - Use of Velocity Interpolation from Mesh in Vortex-in-Cell Method.

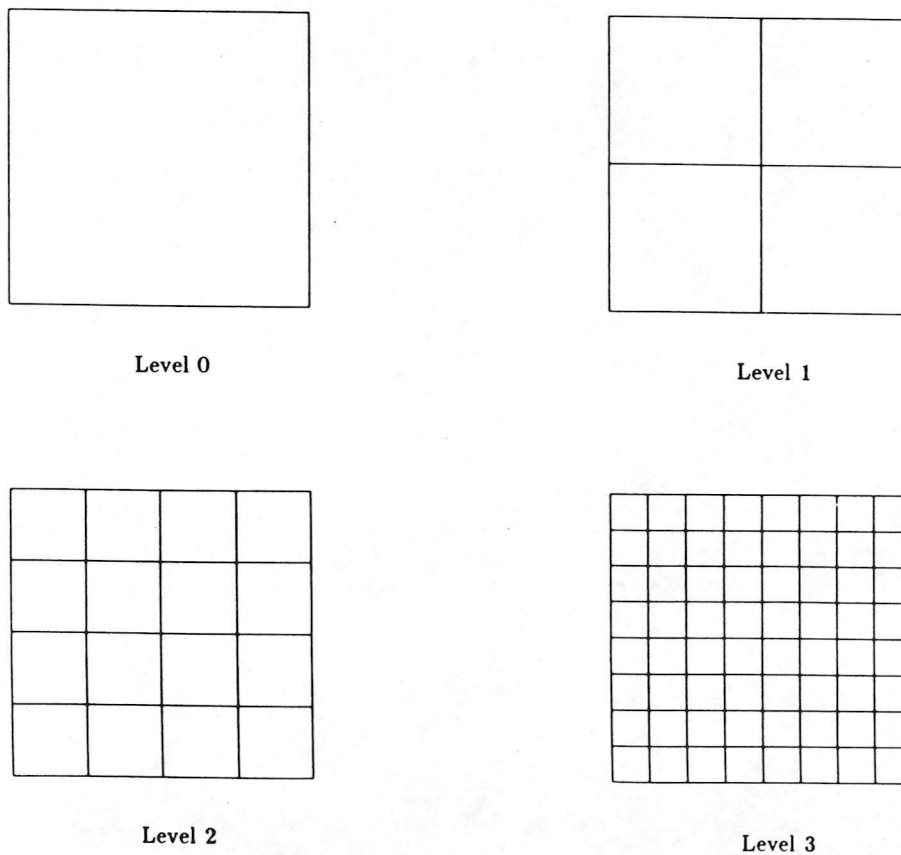


Fig. 2.3 - Division of Flow Field into Uniform Square Zones.  
 ( from GREENGARD 1987 [11] )

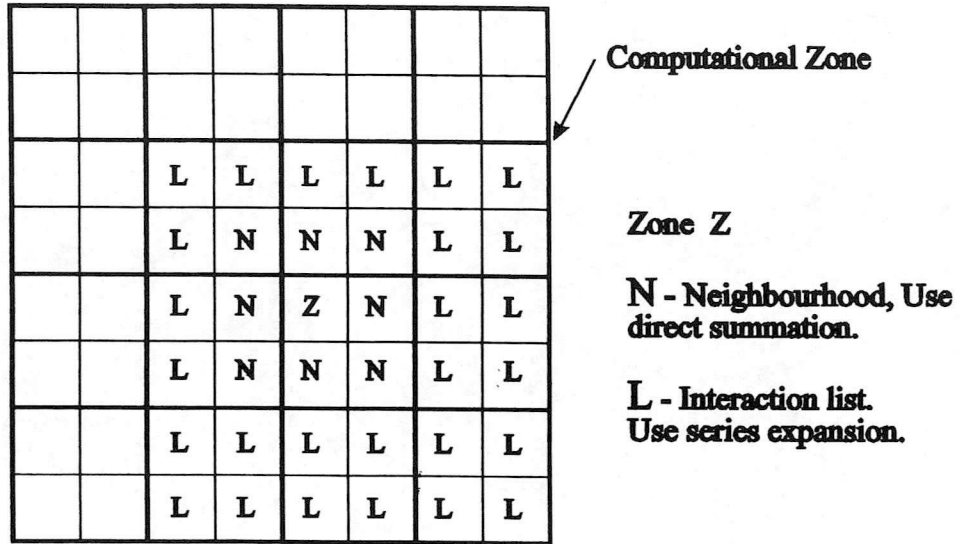
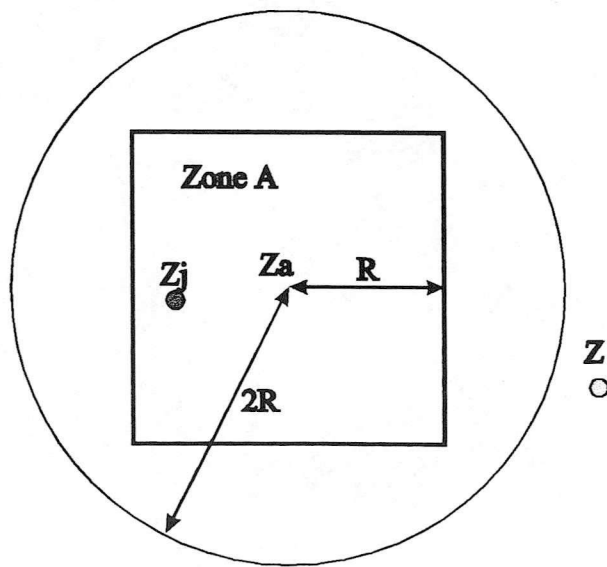


Fig. 2.4 - Fast Multipole Method : Neighbourhood and Interaction List.



**Zone A, centre  $Z_a$ , radius  $R$**   
**Vortex at  $Z$  - use series expansion.**

Fig. 2.5 - Criteria for using Series Expansion or Direct Summation in Velocity Calculation.

Zonal expansion used on all vortices outside the "Direct sum zone".

This region covers a large area of flow field due to aspect ratio of zone, and hence, influence of zone on vortices a large distance away, are still calculated using Direct Summation.

Improvement in Operation Count is very small.

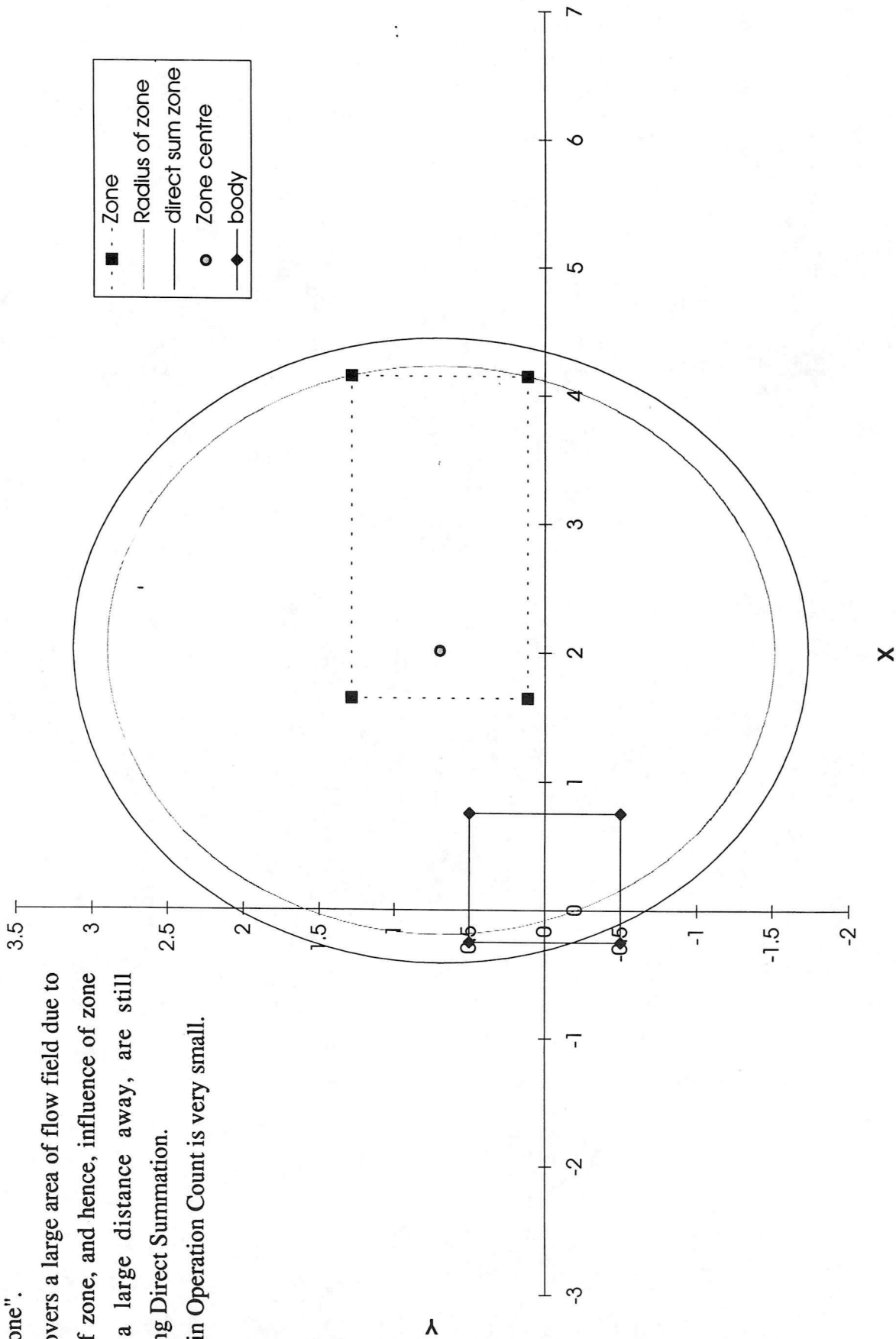
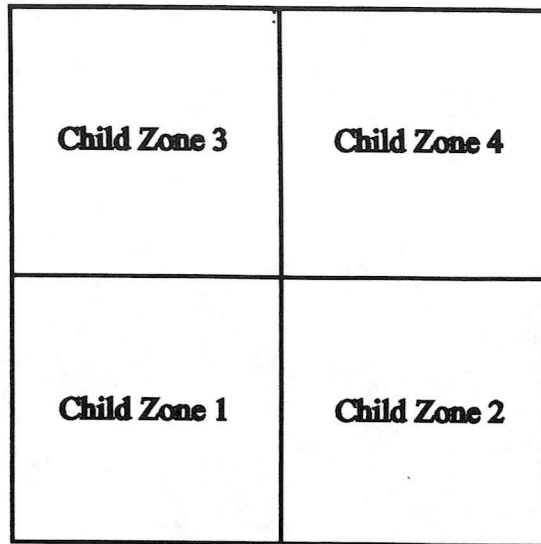
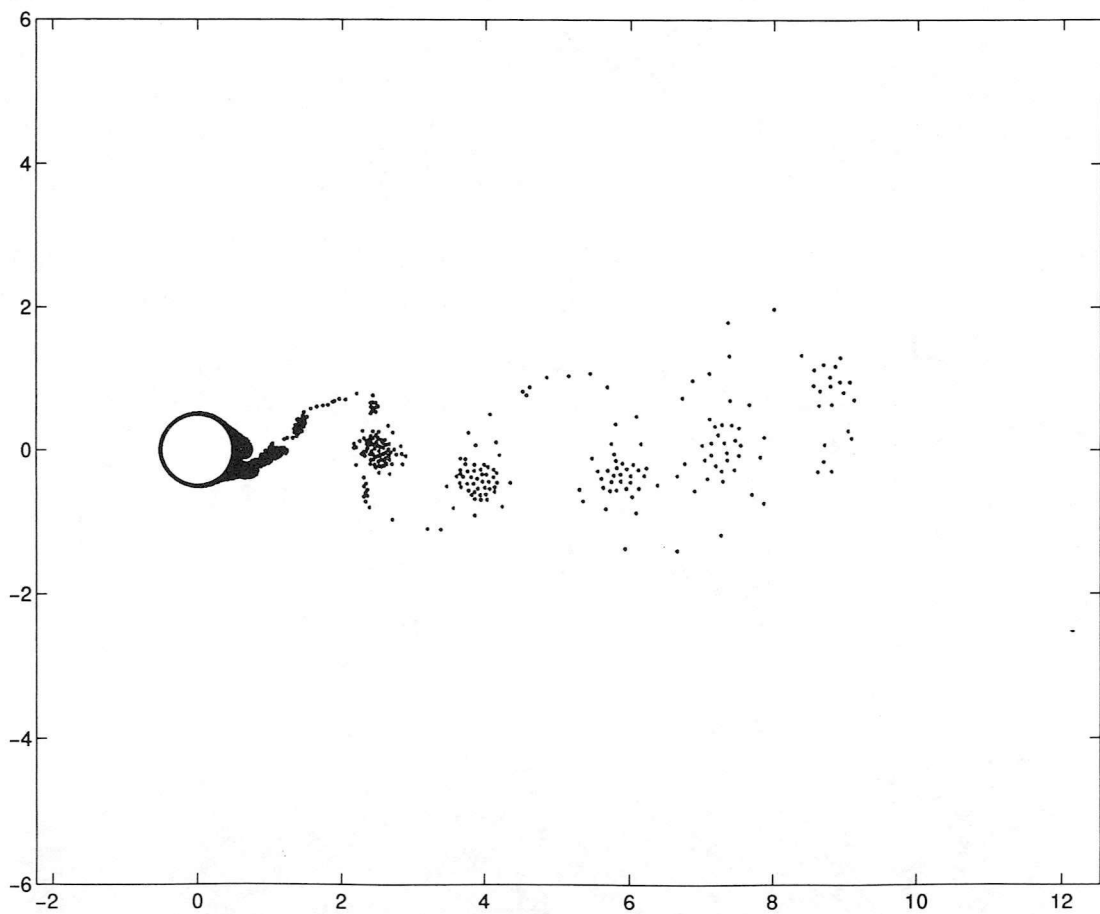


Fig. 3.1 - Demonstration of Problem of Using Rectangular Zones.



**Division of Parent Zone into  
Four Child Zones in Order Shown.**

Fig. 3.2 - Order of Child Zones in Zonal Decomposition Algorithm.



a) Initial Flow Field.

Fig. 3.3 - Decomposition of Flow Field into Hierarchical Zonal Structure.

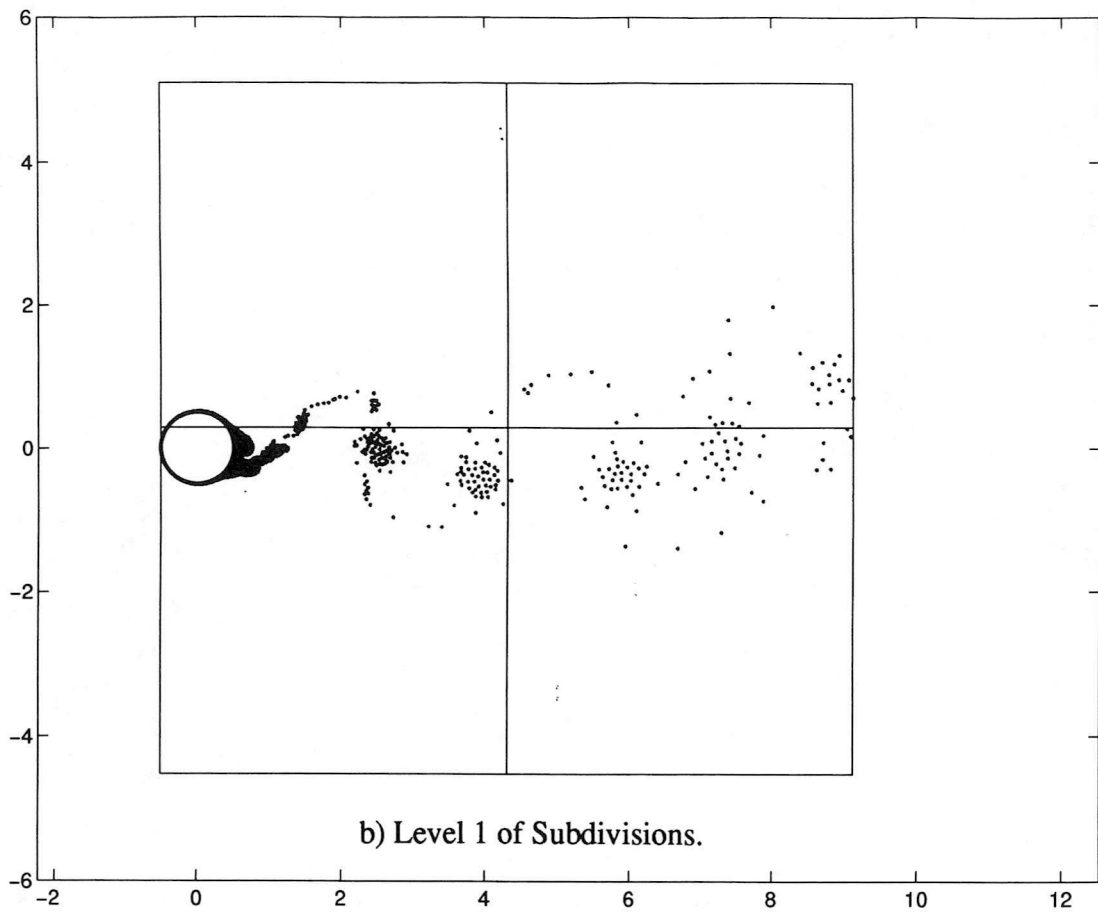
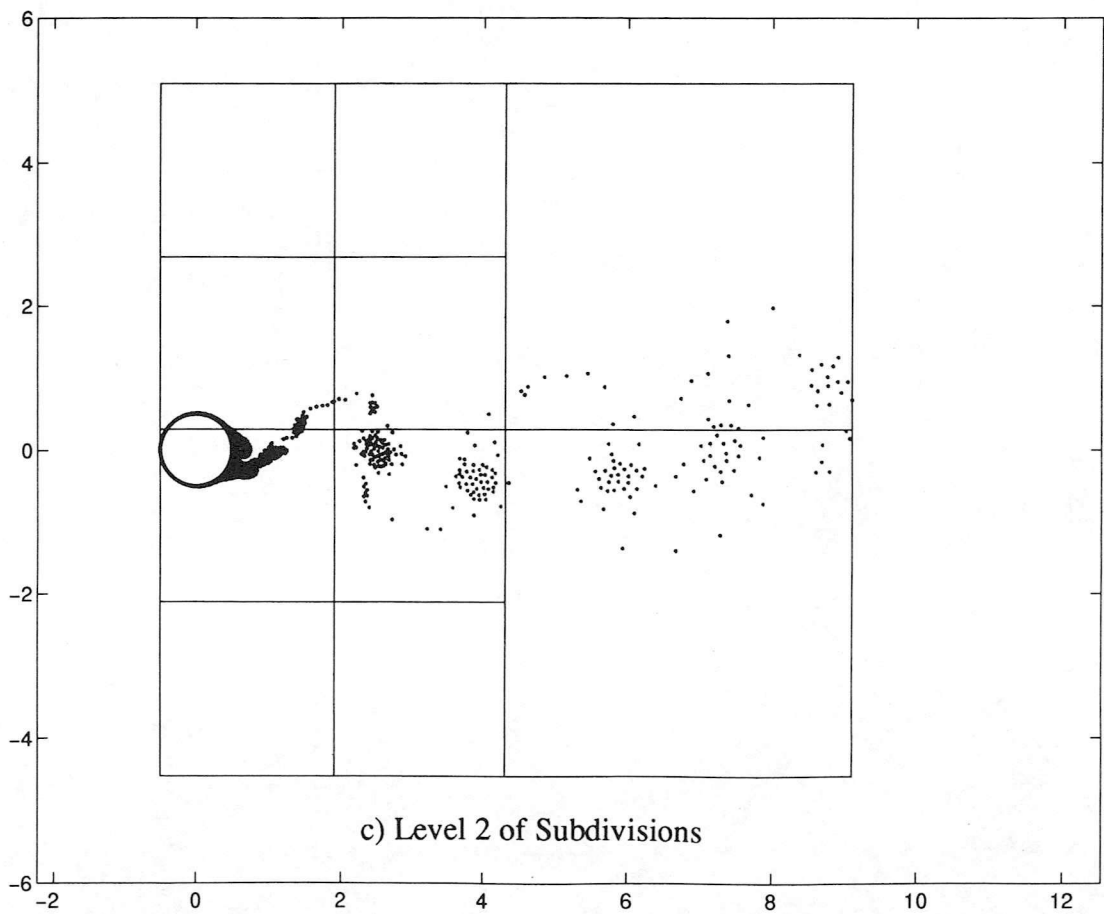
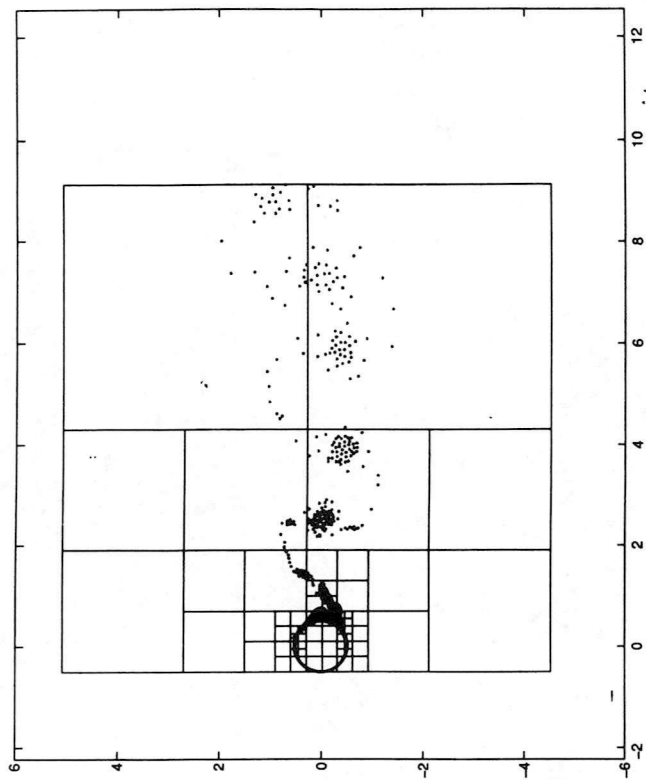
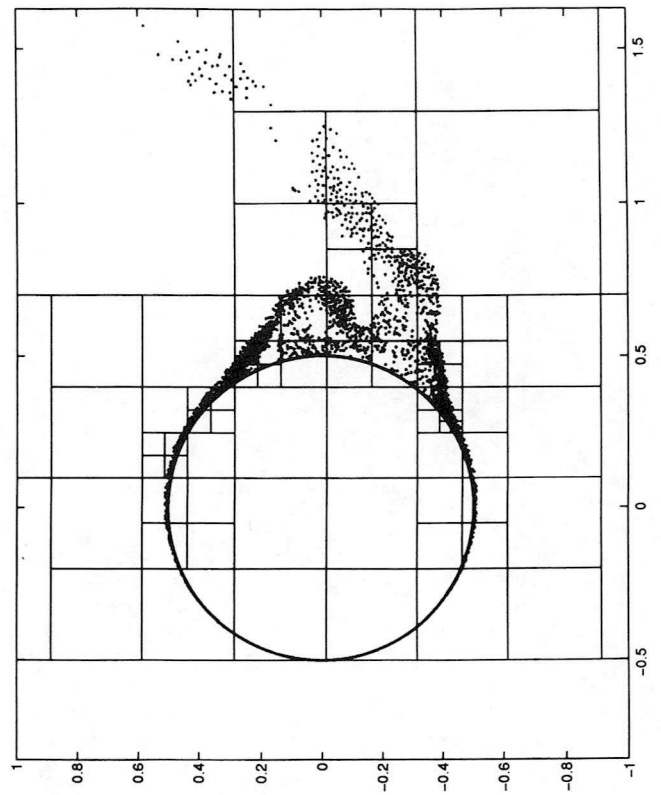


Fig. 3.3 - Decomposition of Flow Field into Hierarchical Zonal Structure.



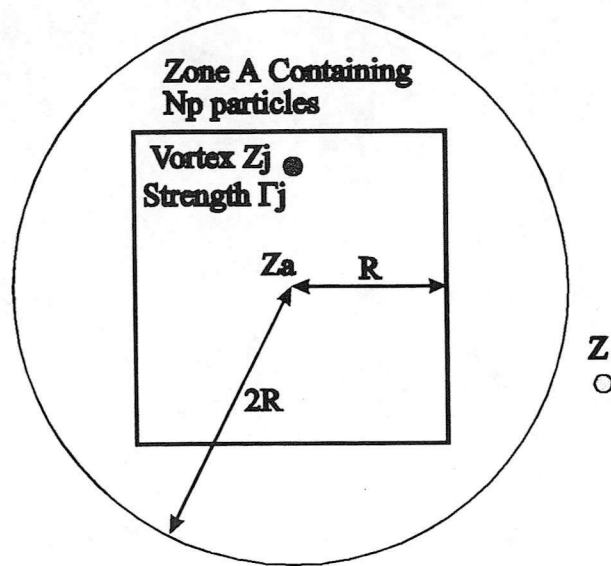


d) Resultant Zonal Decomposition of Flow Field.



e) Close up View of Resultant Zonal Decomposition.

Fig. 3.3 - Decomposition of Flow Field into Hierarchical Zonal Structure.

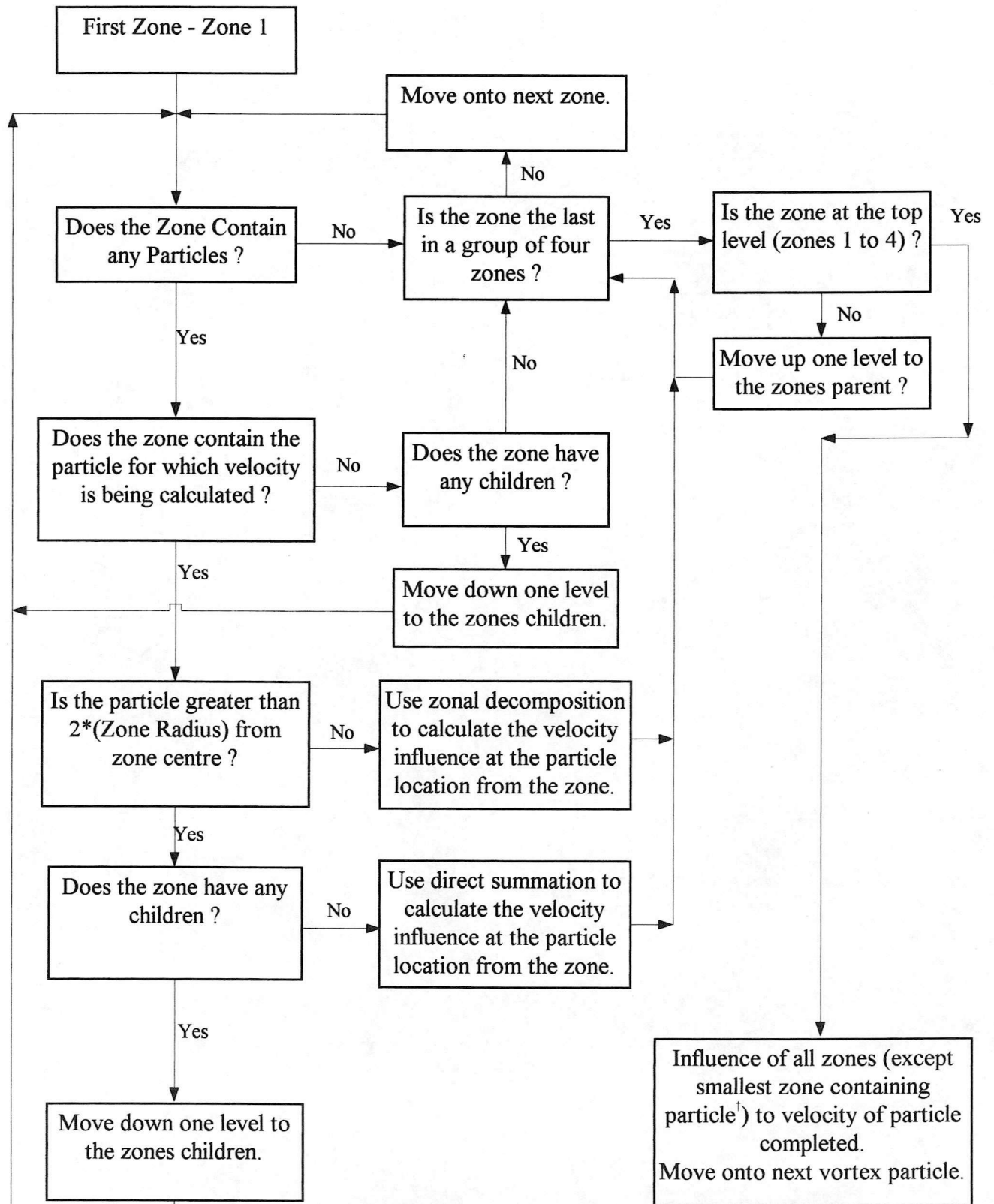


Zone A, centre  $Z_a$ , radius  $R$   
 Vortex at  $Z$  - use series expansion.

Fig. 3.4 - Nomenclature for Zone.

## Algorithm For Velocity Calculation using Zonal Decomposition.

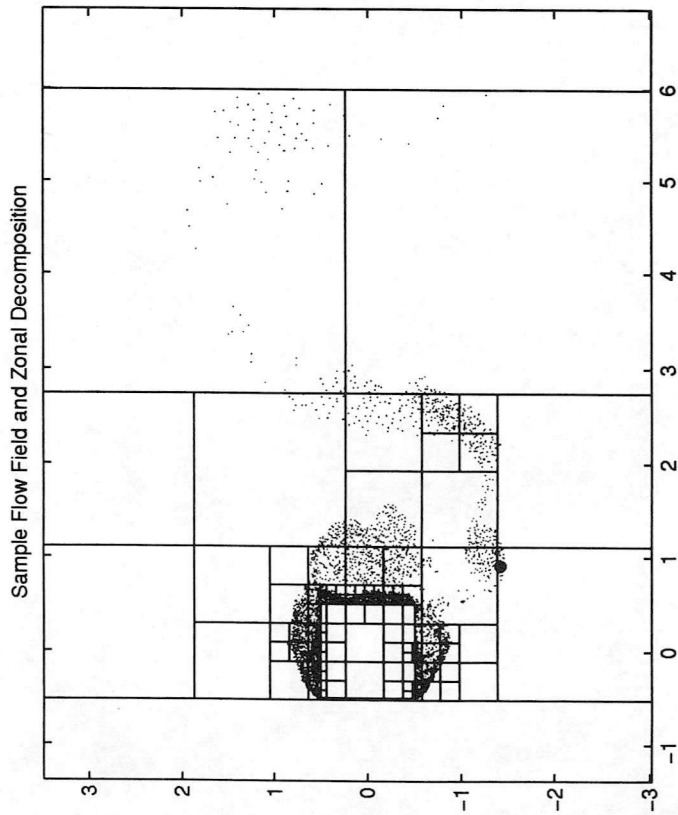
Algorithm performed in a single pass over all vortex particles.






† At lowest level, contribution of other particles in zone calculated from direct summation later in algorithm.

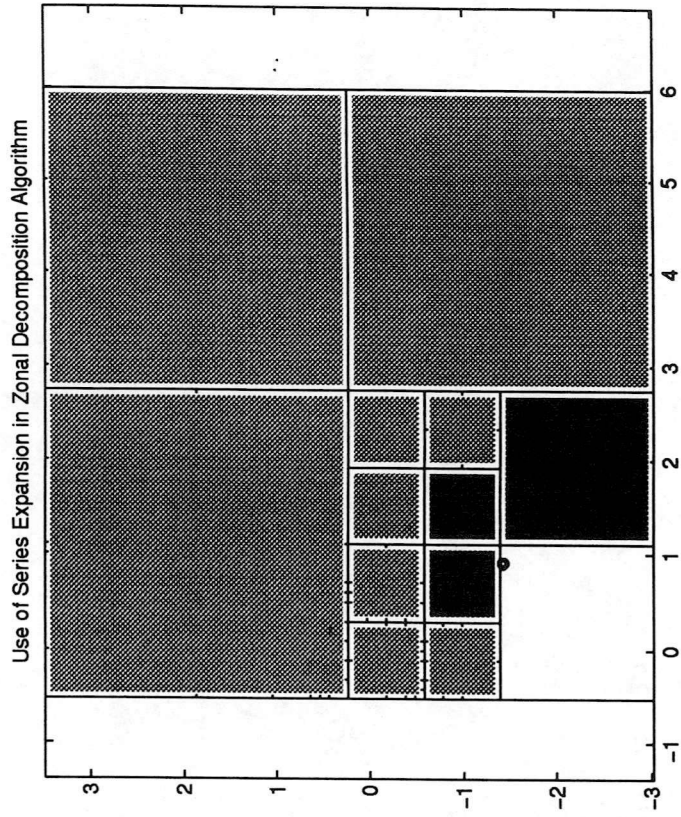
Fig. 3.5 - Flowchart for Velocity Calculation using Series Expansion and Zonal Decomposition.





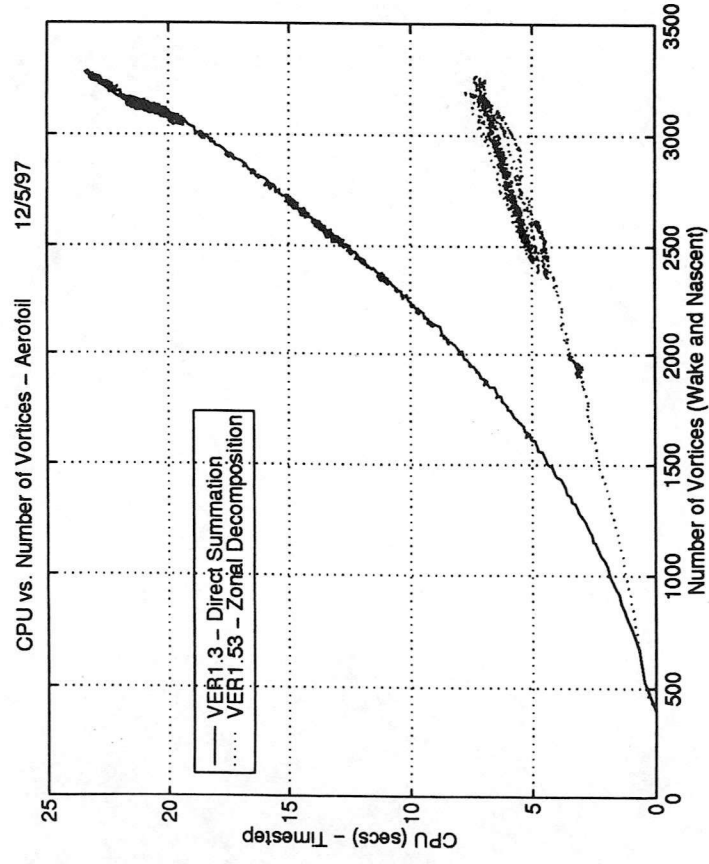
a) Sample Flow Field, Showing Zonal Structure and Point at which Velocity Being Calculated.

-  = Use Series Expansion in Velocity Calculation
-  = Use Direct Summation in Velocity Calculation
-  = Point at Which Velocity is being Calculated

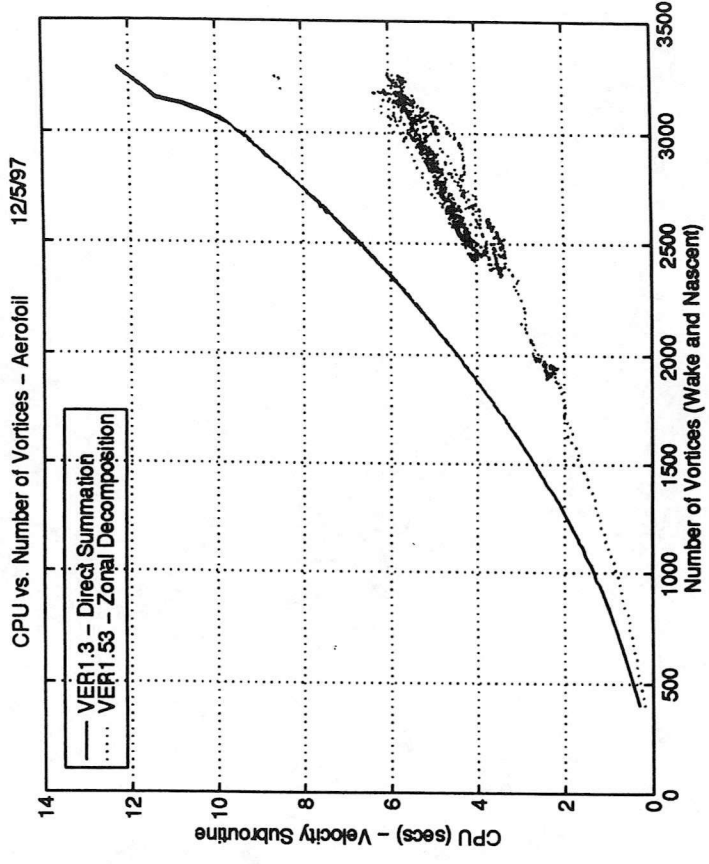


b) Demonstration of Which Zones used in Series Expansion.

Fig. 3.6 - Example of Use of Series Expansion and Zonal Decomposition in Velocity Calculation.

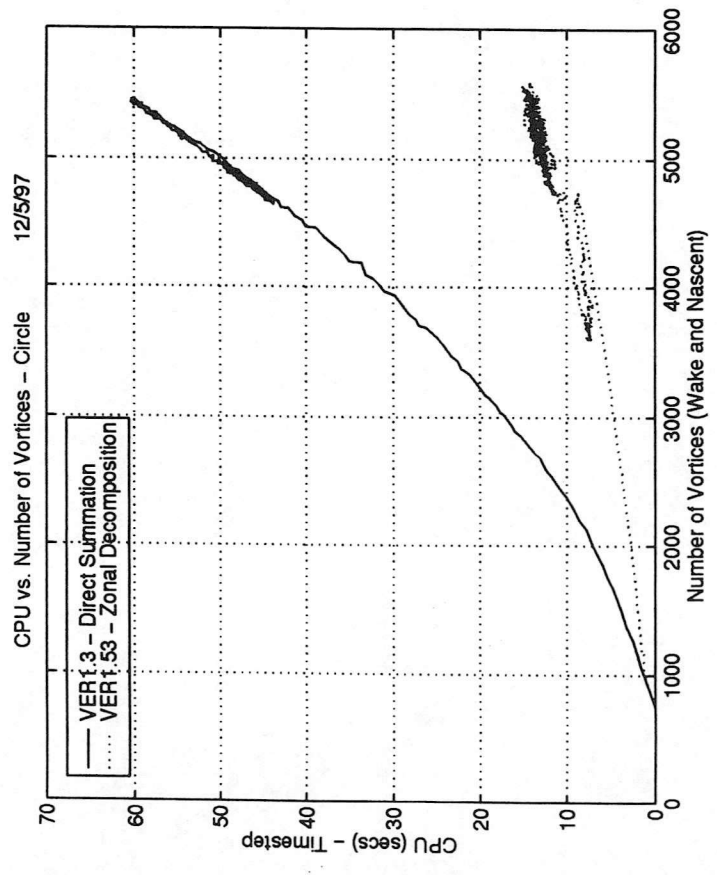


a) Full Timestep

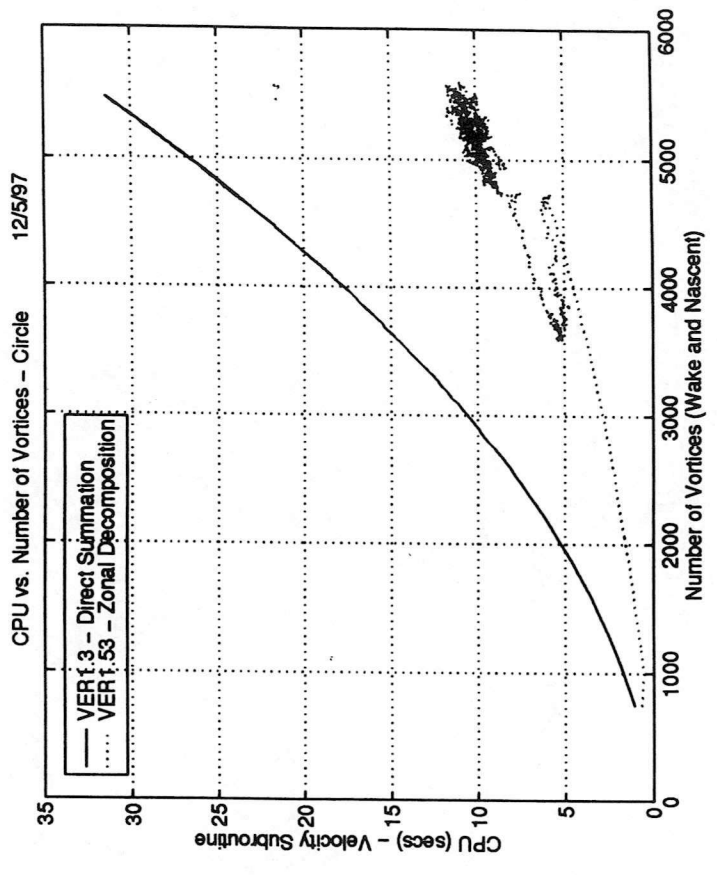


b) Velocity Calculation Only.

Fig. 4.1 - Comparison of CPU Required for Direct Summation and Zonal Decomposition : AEROFOIL Test Case.

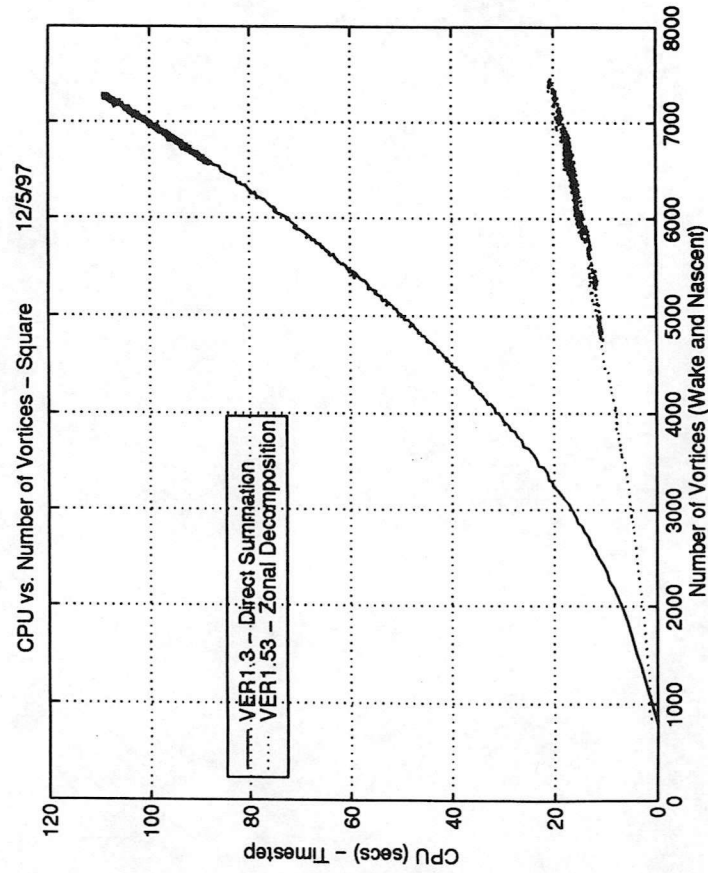


a) Full Timestep

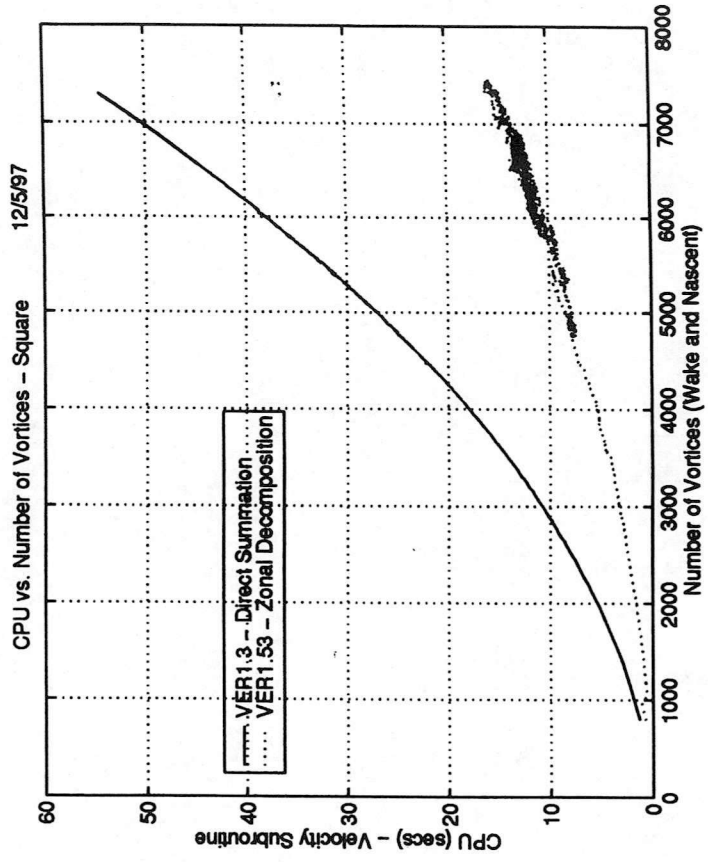


b) Velocity Calculation Only.

Fig. 4.2 - Comparison of CPU Required for Direct Summation and Zonal Decomposition : CIRCLE Test Case.

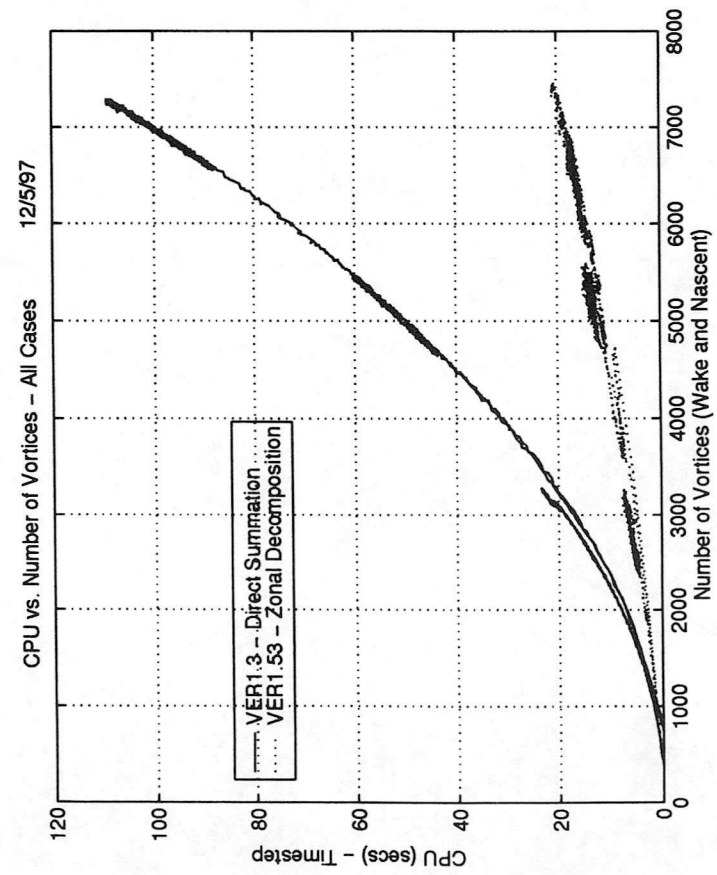


a) Full Timestep

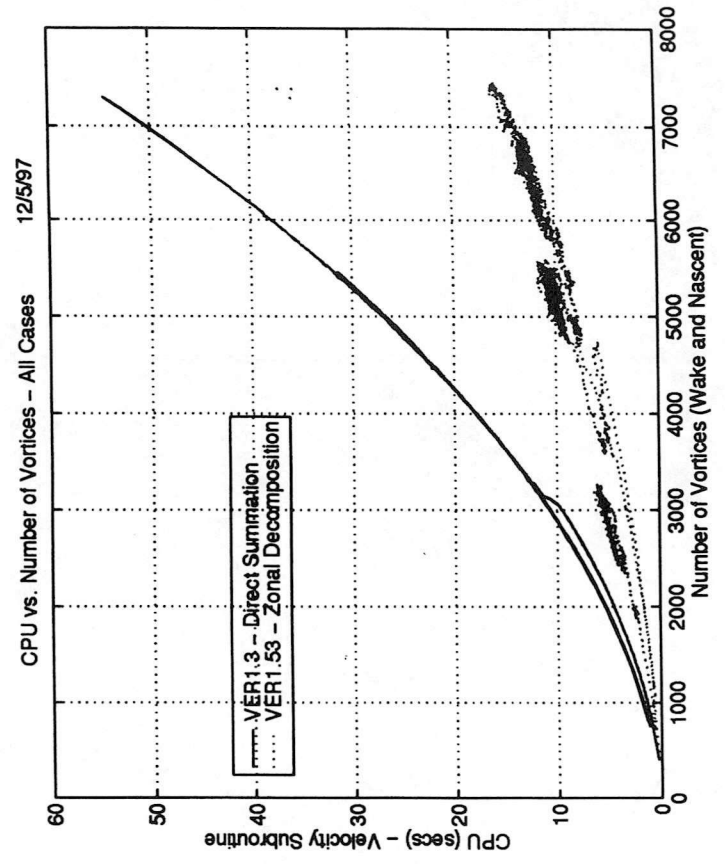


b) Velocity Calculation Only.

Fig. 4.3 - Comparison of CPU Required for Direct Summation and Zonal Decomposition : SQUARE Test Case.

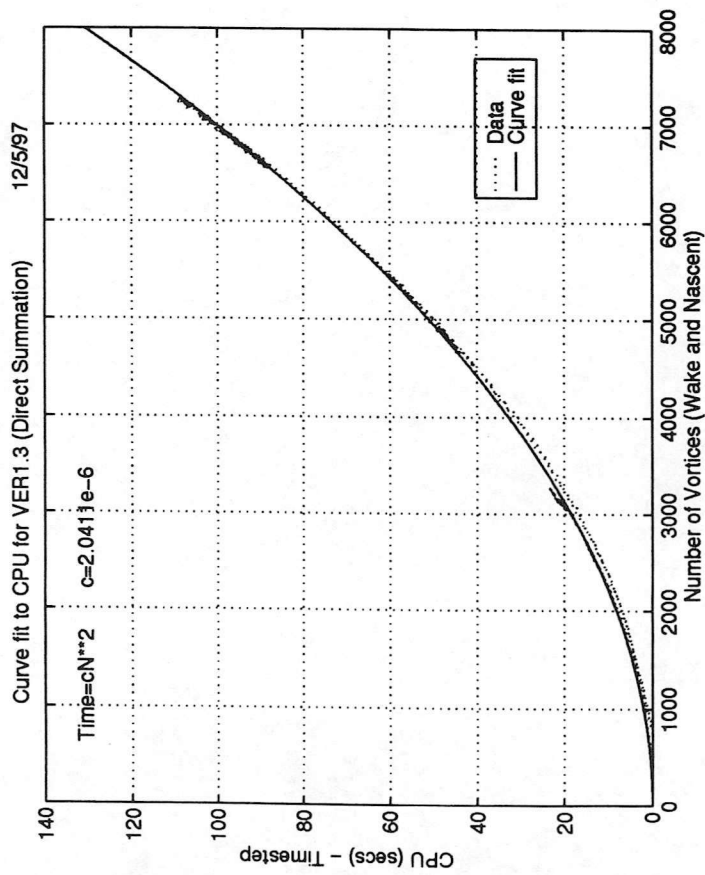


a) Full Timestep

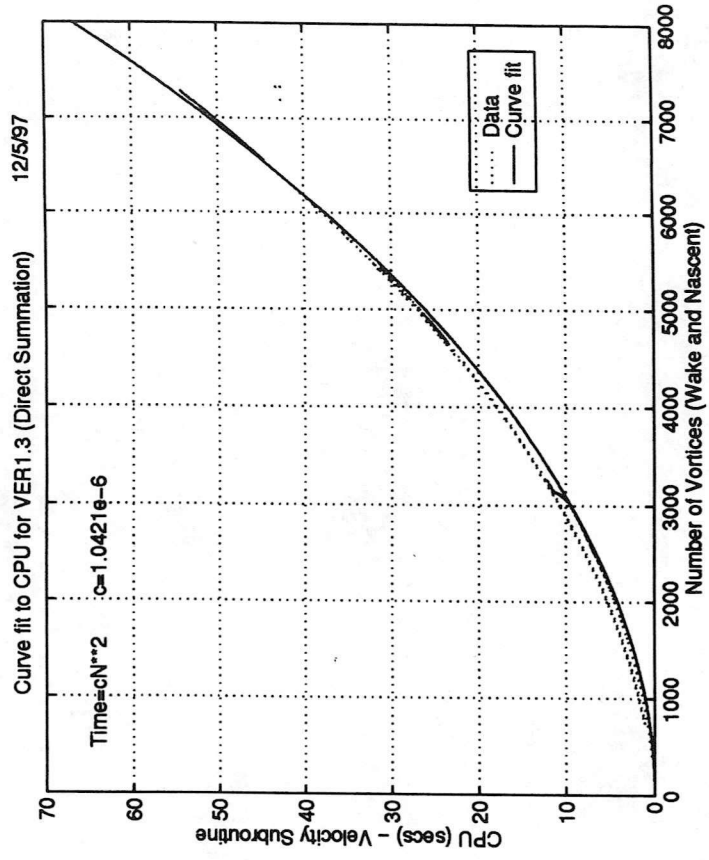


b) Velocity Calculation Only.

Fig. 4.4 - Comparison of CPU Required for Direct Summation and Zonal Decomposition : All Test Cases.



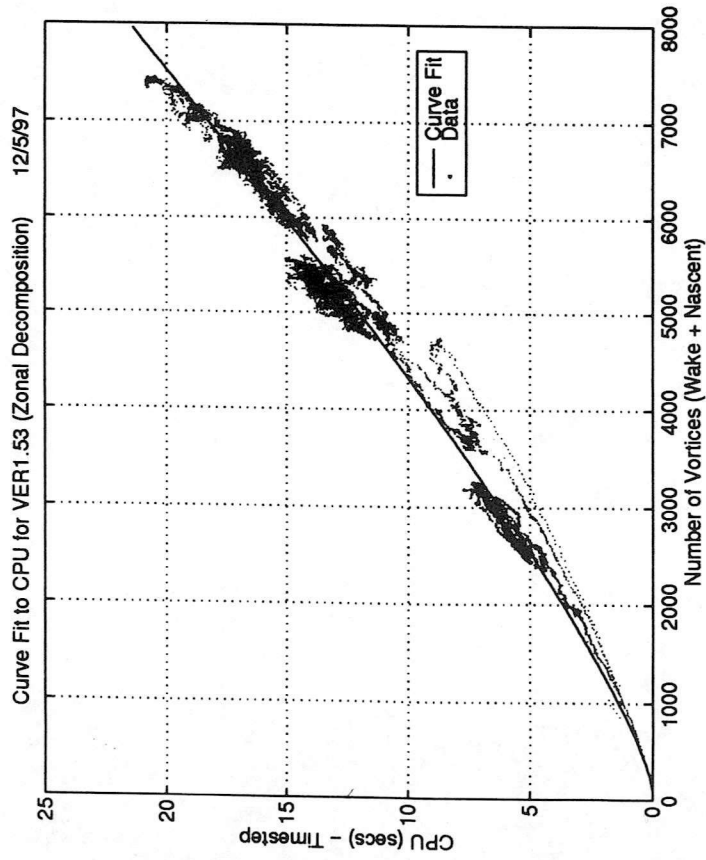
a) Full Timestep



b) Velocity Calculation Only.

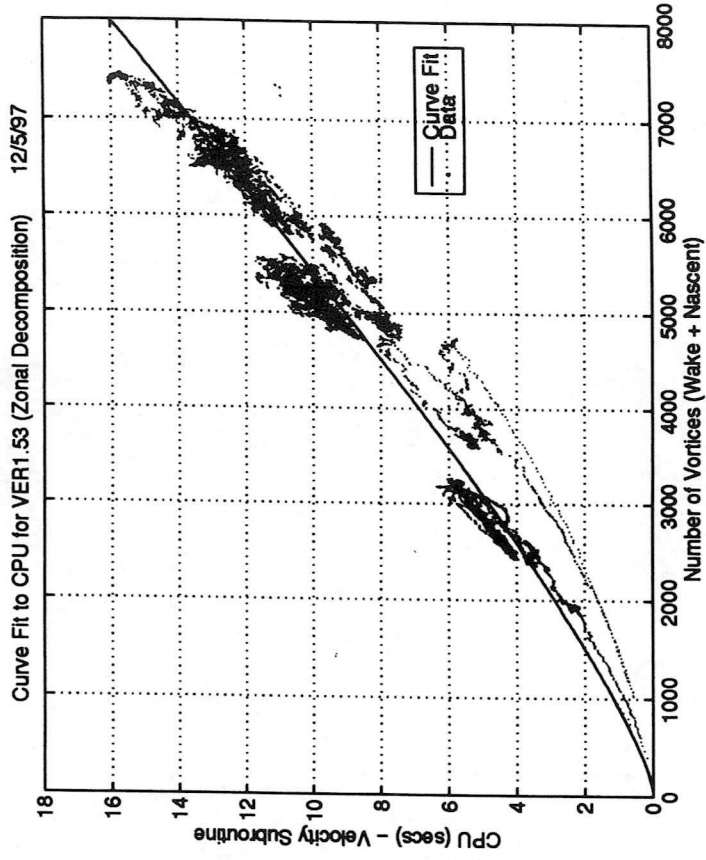
Fig. 4.5 - Curve Fit to Direct Summation Method.

$$\text{Time} = aN + bN \log N \quad a = -2.9832e-3, \quad b = 1.4547e-3$$



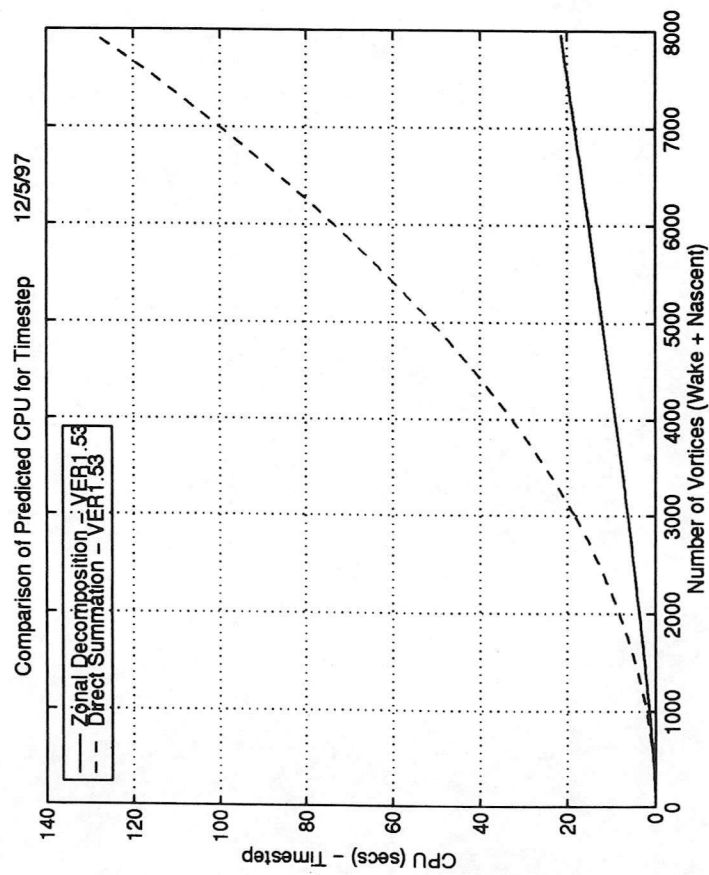
a) Full Timestep

$$\text{Time} = aN + bN \log N \quad a = -1.6905e-3, \quad b = 9.4648e-4$$

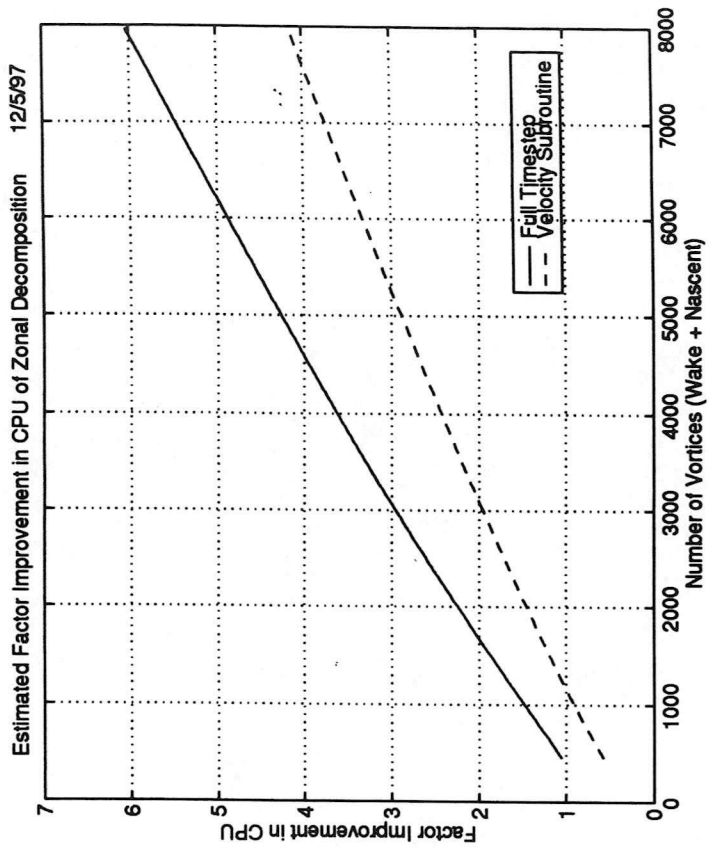


b) Velocity Calculation Only.

Fig. 4.6 - Curve Fit to Zonal Decomposition Method.



a) Comparison of Predicted CPU



b) Factor Improvement Obtained from Zonal Decomposition Algorithm.

Fig. 4.7 - Comparison of Curve Fits.



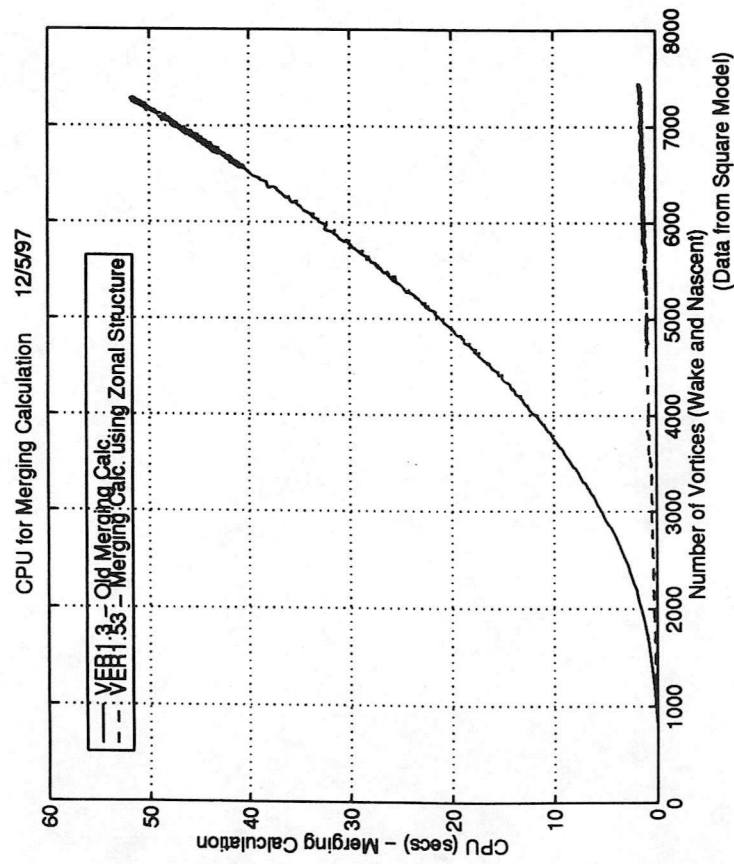


Fig. 4.8 - Comparison of Old and New Merging Calculation.

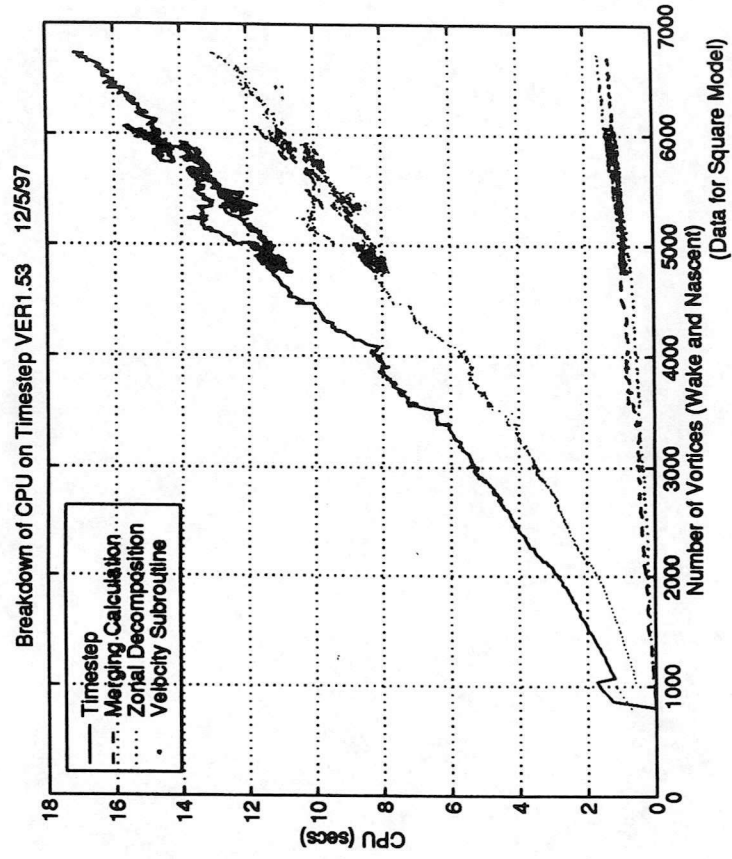
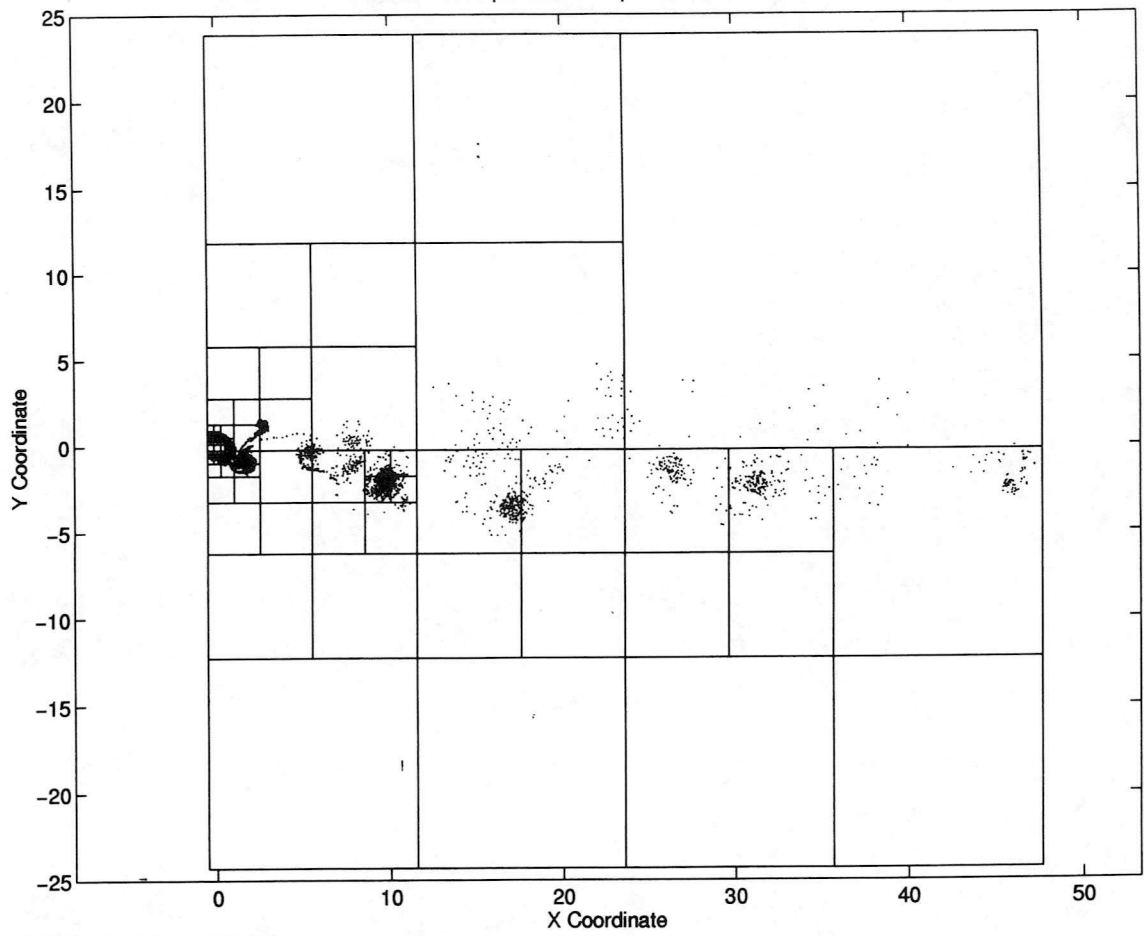


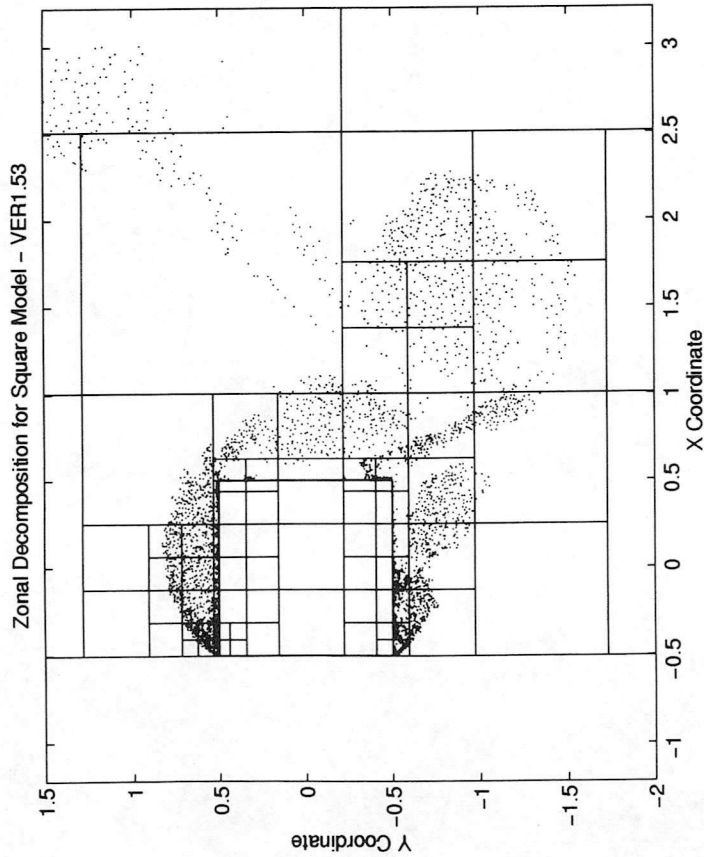
Fig. 4.9 - Breakdown of CPU for a Timestep in the Zonal Decomposition Algorithm

Zonal Decomposition for Square Model - VER1.53

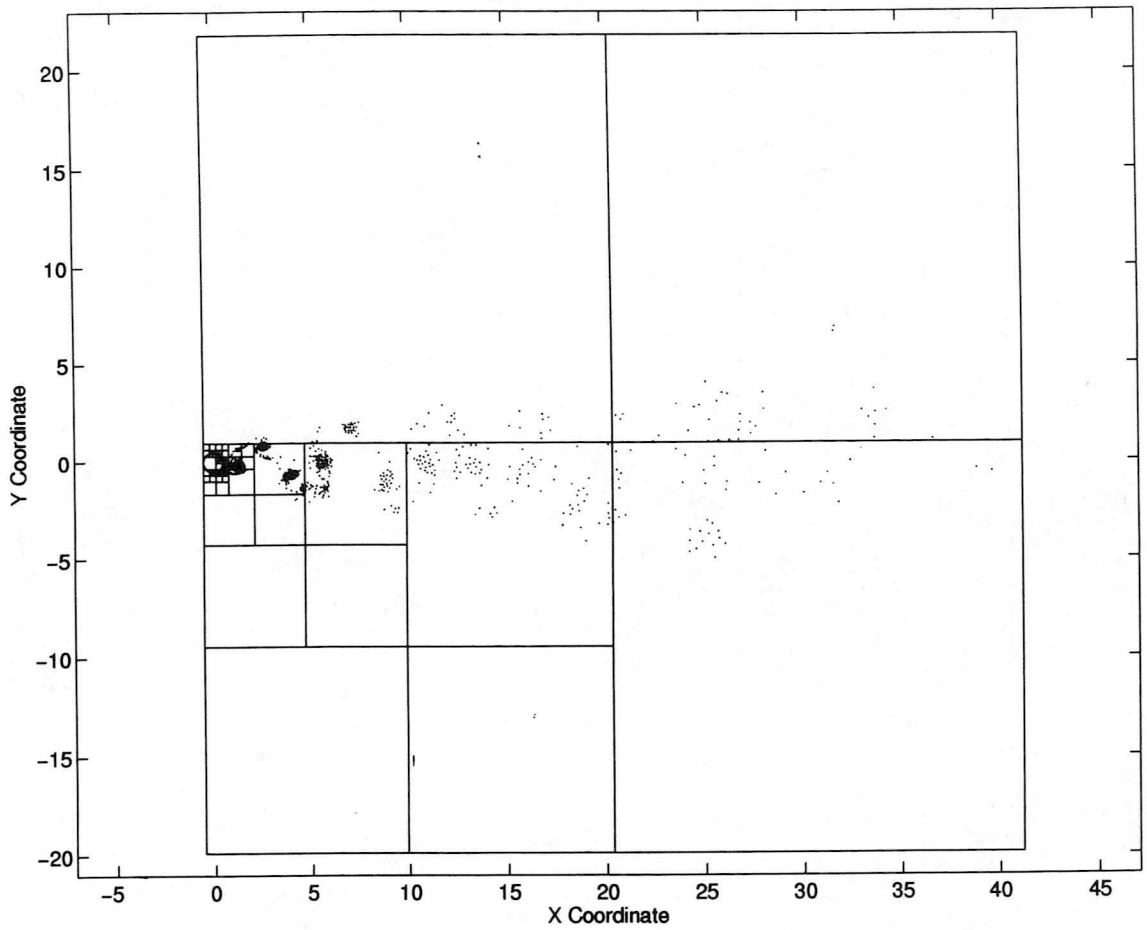


a) Whole Flow Field

Fig. 4.10 - Sample Decomposed Flow Field for Square Model.

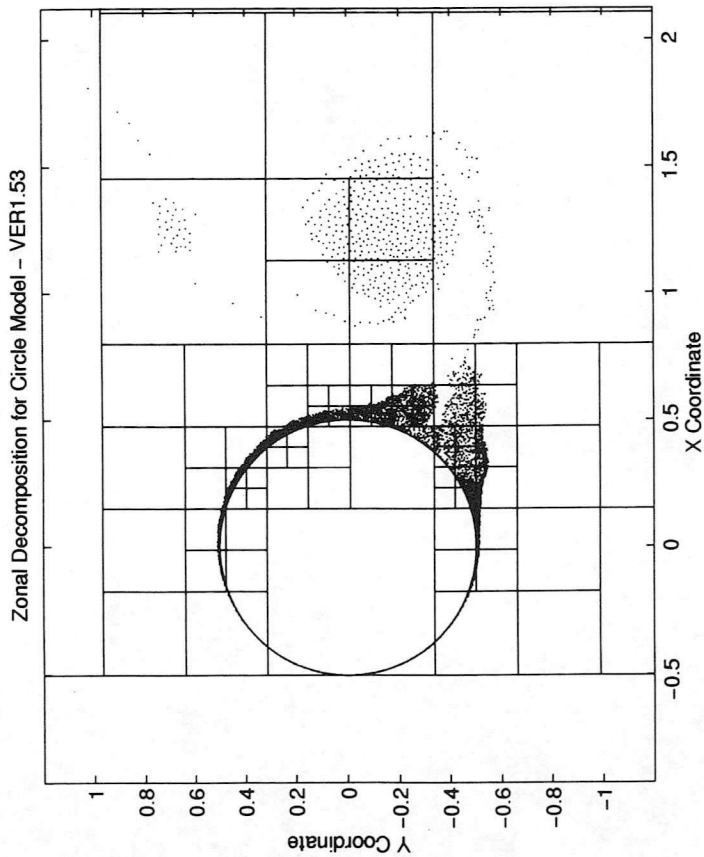


b) Close up View.

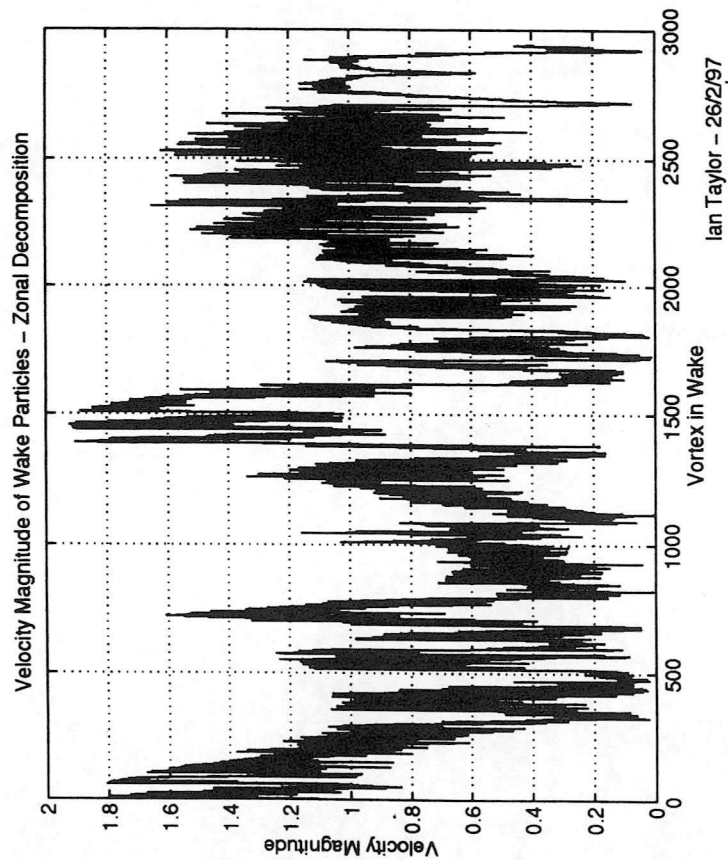


a) Whole Flow Field

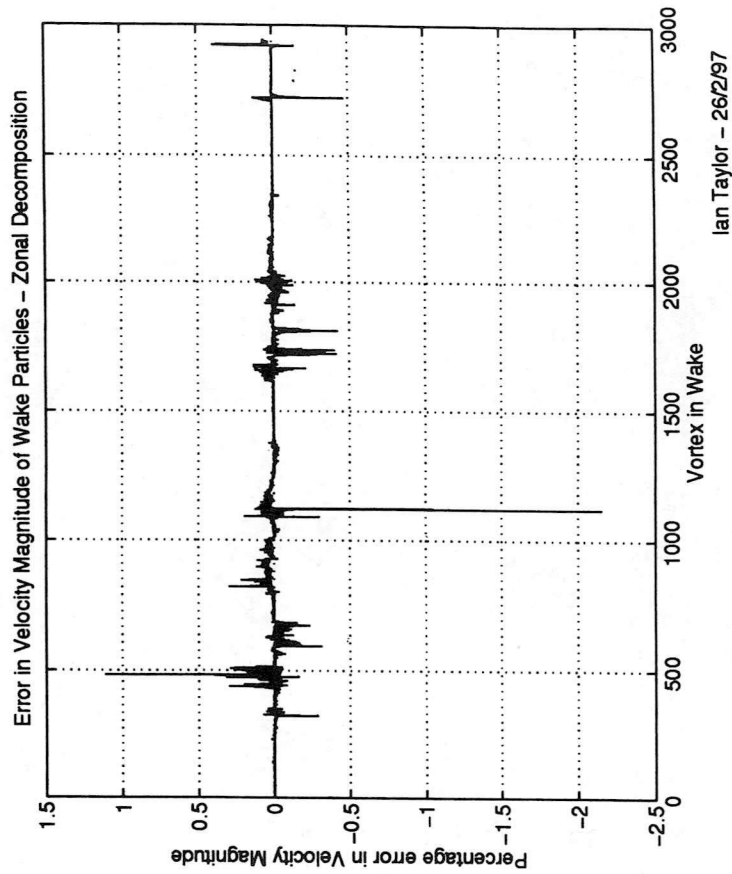
Fig. 4.11 - Sample Decomposed Flow Field for Circle Model.



b) Close up View.



a) Velocity Magnitude



b) Percentage Error.

Fig. 4.12 - Error in Velocity Magnitude for Wake Vortex Particles.

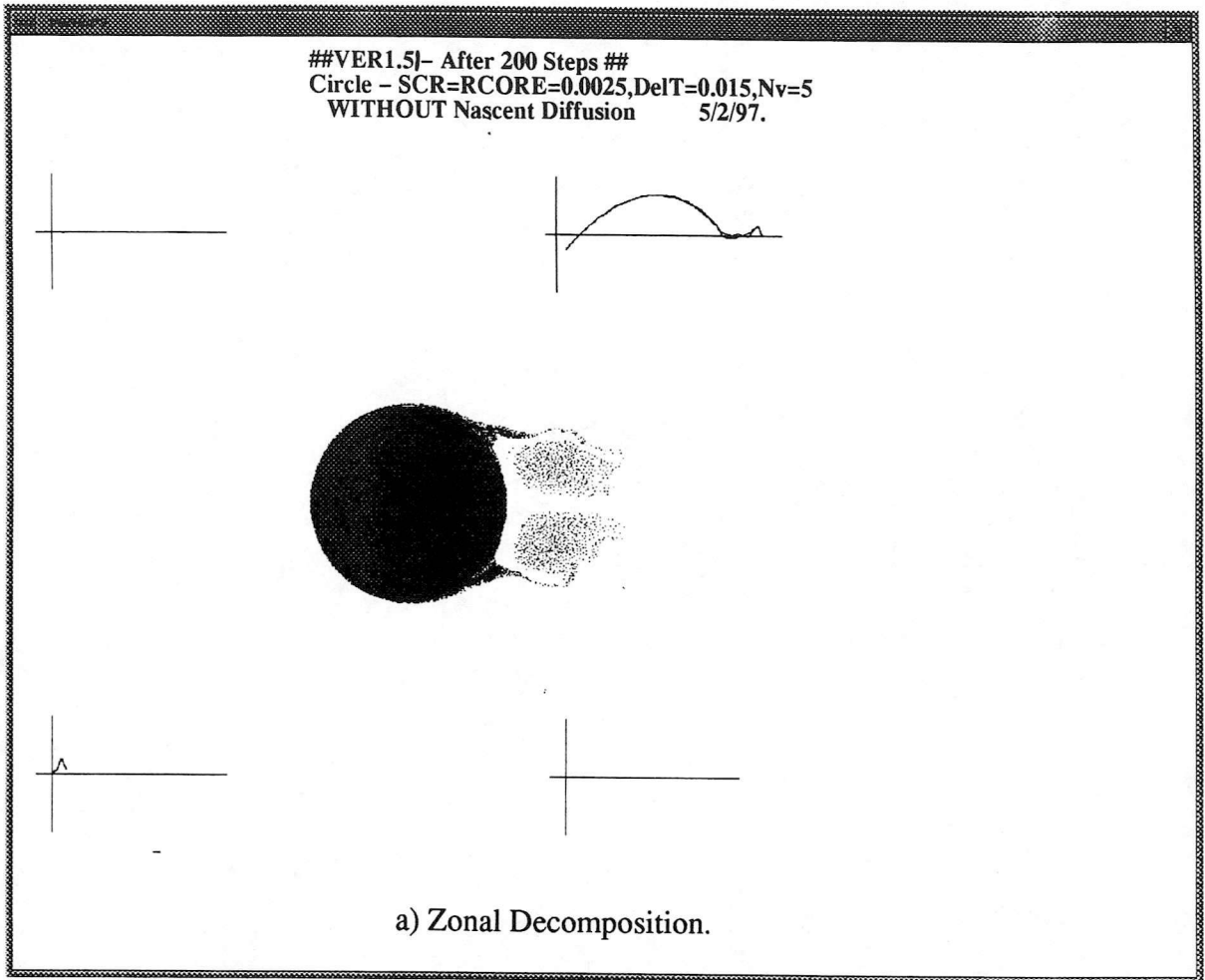
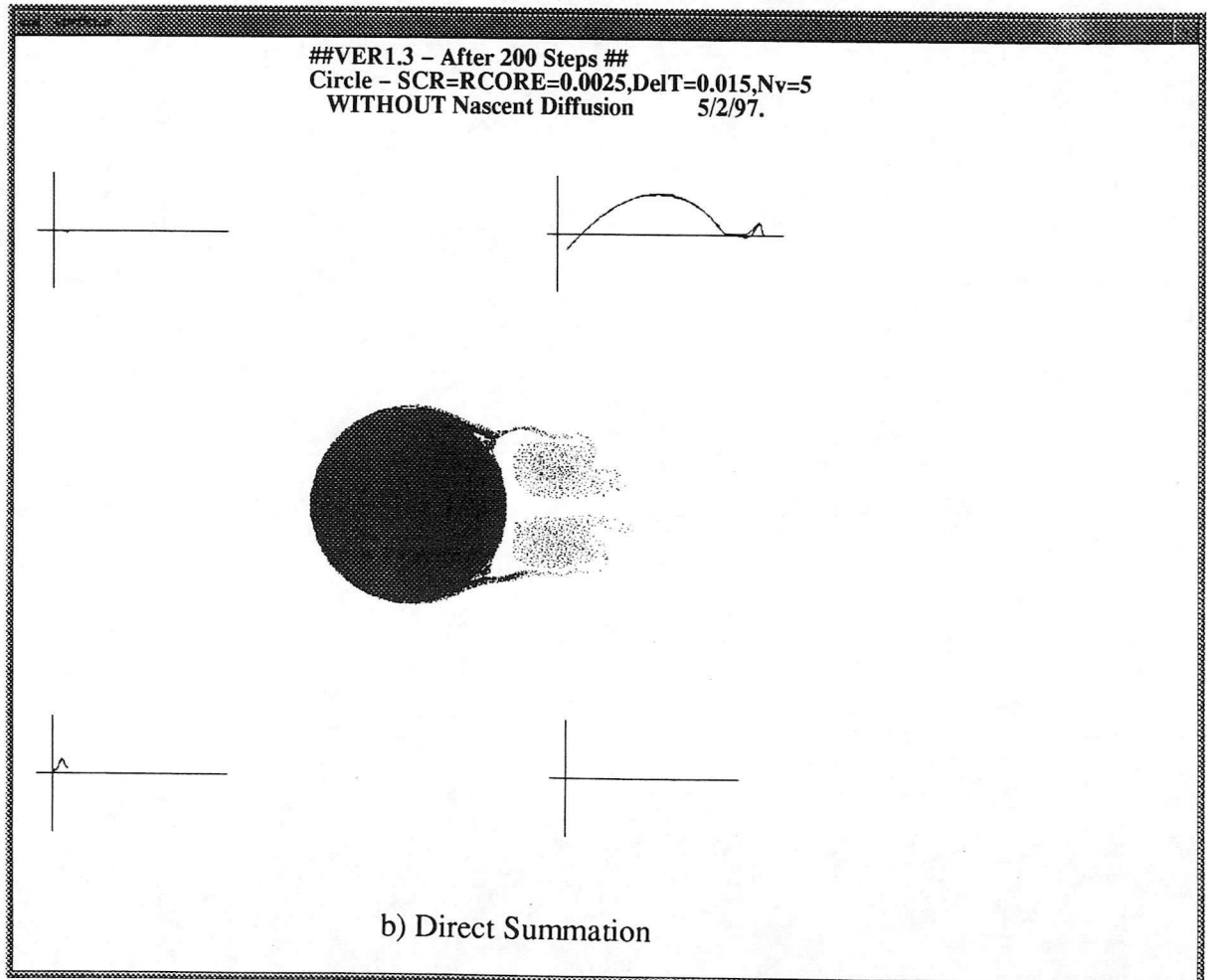


Fig. 4.13 - Comparison of Flow Field for Circle Model after 200 Timesteps.



### Variation in CPU and Error with H

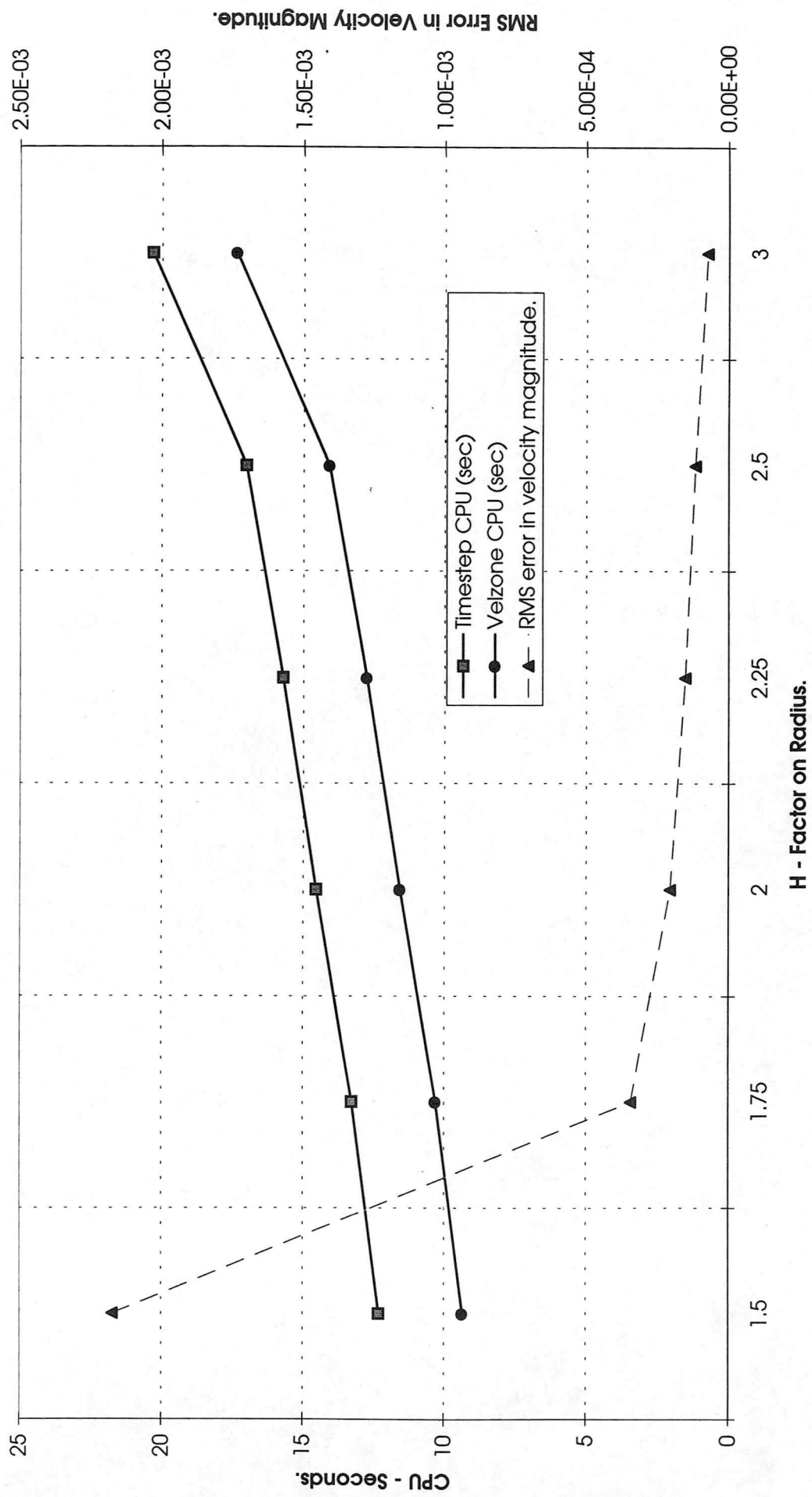


Fig. C.1 - Variation of CPU and RMS Error with Factor on Zone Radius - H.

Variation in CPU and Error with Nt

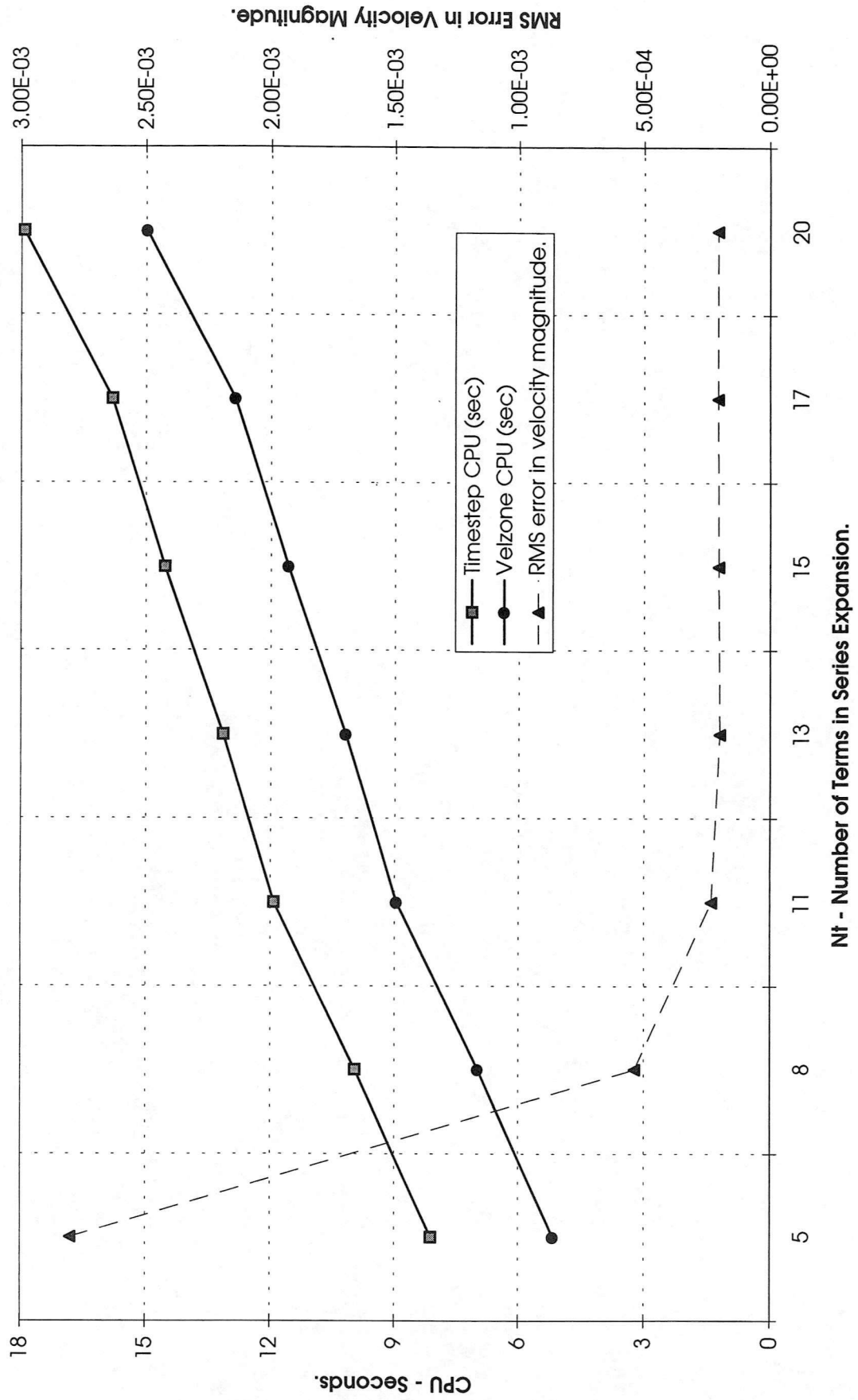


Fig. C.2 - Variation of CPU and RMS Error with Number of Terms in Series Expansion - Nt.

Variation in CPU and Error with Np

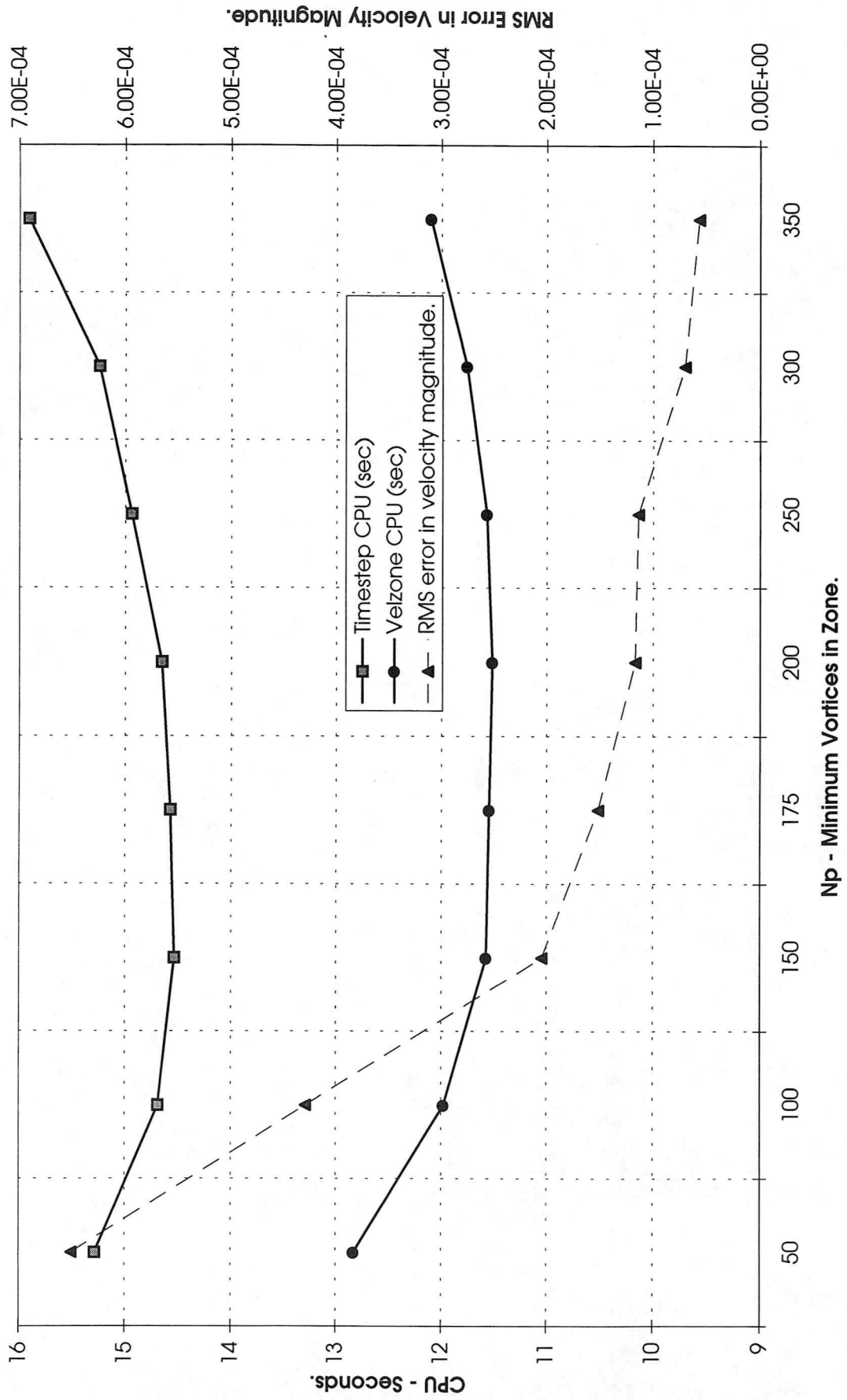


Fig. C.3 - Variation of CPU and RMS Error with Minimum Number of Vortices in Zone - Np.