THREE-DIMENSIONAL VORTEX METHOD FOR PARACHUTES

MAYER HUMI

Department of Mathematical Sciences, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609–2280, U.S.A.

SUMMARY

We describe the implementation of a new 3D vortex algorithm for the computation of the drag and flow field around parachutes. Among its novel features, the algorithm couples large eddy simulation methodology with the vortex method, away from the wall region. Furthermore, boundary conditions for a wall (no-slip) and compliant boundaries were implemented. The results of several simulations using this algorithm are analysed and discussed. The spectral contents of the vortex method are also considered.

KEY WORDS Parachutes Vortex methods CFD Spectral analysis

1. INTRODUCTION

The simulation of flow fields around parachute geometries at high Reynolds numbers presents some inherent difficulties.[1–4] At the core of these difficulties is the fact that the problem has three length scales. The first is the parachute span, which is of the order of 10 m; the second is its thickness, which is of the order of 10^{-3} m; the third is related to the wake, whose size can exceed 200 m. The accurate resolution of these different turbulent flow scales in a grid-based model seems to be impossible with the present computer technology. As a result, the aerodynamic characteristic of a parachute has to be computed using models where the ratio between the first two length scales mentioned above is 10 rather than 10^4. Furthermore, parachutes undergo dynamic shape deformations which are not easy to account for.

In view of these circumstances simulation techniques which are grid-independent have a clear advantage in a realistic modelling of the parachute problem. Accordingly, the vortex method[4–12] represents, in principle, an ideal paradigm for these computations. In fact, the vortex blobs used in these simulations do not require the introduction of a fixed grid. Furthermore, if necessary, vorticity generation can be adjusted to account for the physical insights one has about the nature of the problem. Nevertheless, due to the large lateral parachute span and the dimension of the wake domain, a large number of vortices is needed to obtain a satisfactory representation of the flow field. Furthermore, the flows we are interested in are turbulent in nature, and the impact of these effects on the vortex blobs in the wake region is important. It is, therefore, essential for our purposes to couple the vortex method with a proper turbulence model.[5]

In the past vortex methods were used mostly for inviscid flows in 2D[9] although some attempts were made[6,11,12] to extend it to 3D. Viscous effects away from the wall were implemented through random walk on the position of the vortices. The correct method to include these effects directly was found only recently.[3,8] However, an algorithm incorporating a turbulence model[5]

0271-2091/93/100879-11$10.50

Received 10 April 1992
Revised 28 September 1992

with the vortex method had never appeared in the literature. In this paper such an attempt is made. For the problem at hand, a coupling between the paradigms of large eddy simulation[4, 11] and the vortex method seems to be the most appropriate. In this context Smagorinski model[11] was used to model the turbulent eddy viscosity of the flow.

Due to this coupling, the circulation of the vortices in the wake region decreases smoothly. As a result the algorithm became more robust, and a better representation of the wake region became possible. Also, the fluctuations in the value of the computed drag tended to be less drastic than otherwise. The resulting algorithm is computationally more extensive from the (pure) vortex method. However, the effort is not prohibitive as no differential equations have to be solved.

It is our prime objective in this paper to describe this algorithm and apply it to the computation of the flow field and the drag coefficients of parachutes. As an example we present the computation of the drag coefficient for a circular plate with different thickness-to-radius ratios, different Reynolds numbers and different attack angles near 90° (i.e. the free flow is normal to the plate). Also, different boundary conditions (for a wall and compliant surface) are considered.

From another perspective flow simulations with the vortex method depend crucially on the number of vortices used and the choice of the blob function. In the past, various prescriptions appeared in the literature for the choice of this function. The prime concerns were in most cases, smoothness, computational simplicity and the stability of the resulting method. However, to our knowledge, very little (if any) attention was paid to the impact of this choice on the spectral contents of the flow. We shall prove that if the blob function has compact spectral support then the computed flow has exactly the same spectral support for all times. Thus, under these circumstances, the vortex method mimics large-eddy simulation.

The plan of the paper is as follows: In Section 2 we present a brief overview of the vortex method and its coupling to large eddy simulation. In Section 3 we describe the algorithm that was used to implement this method and the effects of compliant boundaries. In Section 4 the simulation results for the circular plate are presented and discussed. In Section 5 the spectral contents of the vortex method is analysed. We end up with some conclusions in Section 6.

2. BASIC EQUATIONS

Incompressible fluid flow is governed by Navier–Stokes equations. In proper units these can be written as

$$\nabla\cdot\mathbf{u}=0, \tag{1}$$

$$\frac{\partial\mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla)\mathbf{u}=-\nabla p+\frac{1}{Re}\nabla^2\mathbf{u}, \tag{2}$$

where $Re=Ul/v_0$ is the Reynolds number. The vorticity of the flow is defined as

$$\omega=\nabla\times\mathbf{u}. \tag{3}$$

By taking the curl of equation (2), we obtain

$$\frac{\partial\omega}{\partial t}+(\mathbf{u}\cdot\nabla)\omega=\omega\cdot\nabla\mathbf{u}+\frac{1}{Re}\nabla^2\omega. \tag{4}$$

The relationship between ω and $\mathbf{u}$ can be inverted using Biot–Savart law. In three dimensions we have

$$\mathbf{u}(\mathbf{x},t)=\frac{1}{4\pi}\int\frac{\omega(\mathbf{x}',t)\times(\mathbf{x}-\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|^3}\,\mathrm{d}\mathbf{x}'. \tag{5}$$

The essence of the vortex method is to discretize the vorticity field in the form

$$\omega(\mathbf{x}, t)=\sum_{i=1}^{n} \boldsymbol{\Gamma}_i(t)\delta_\sigma[\mathbf{x}-\mathbf{x}_i(t)], \tag{6}$$

where δ_σ are 'blob' functions approximating the Dirac δ-function. Usually δ_σ is chosen so that

$$\delta_\sigma(\mathbf{x})=\frac{1}{\sigma^3} f\left(\frac{|\mathbf{x}|}{\sigma}\right) \tag{7}$$

with the 3D normalization

$$4\pi \int_0^\infty f(s)s^2\,\mathrm{d}s=1. \tag{8}$$

Among the various blob functions that are available in the literature,[8,9,12] we chose the one suggested by Leonard *et al.*[6] where

$$f(x)=\frac{15}{8\pi}\frac{1}{(x^2+1)^{7/2}}. \tag{9}$$

With this choice of $f(x)$, equation (5) can be integrated explicitly to obtain

$$\mathbf{u}(\mathbf{x}, t)=\frac{1}{4\pi}\sum \boldsymbol{\Gamma}_j \times \frac{(\mathbf{x}-\mathbf{x}_j)}{|\mathbf{x}-\mathbf{x}_j|^3} F\left(\frac{|\mathbf{x}-\mathbf{x}_j|}{\sigma}\right), \tag{10}$$

where

$$F(r)=4\pi \int_0^r f(s)s^2\,\mathrm{d}s=\frac{1}{8\pi}\frac{2r^2+5}{(r^2+1)^{5/2}}. \tag{11}$$

The evolution equation for the vortex blobs assumes that they follow the characteristic lines

$$\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t}=\mathbf{u}_i(\mathbf{x}_i(t), t), \tag{12}$$

where $\mathbf{u}_i$ is the velocity field produced by all the vortex blobs except the ith one. To derive an equation for the evolution of the circulations $\boldsymbol{\Gamma}_i$, we substitute equation (7) into equation (4). This yields

$$\sum_i \delta_\sigma(\mathbf{x}-\mathbf{x}_i)\left\{\frac{\mathrm{d}\boldsymbol{\Gamma}_i}{\mathrm{d}t}-(\boldsymbol{\Gamma}_i\cdot\nabla)\mathbf{u}-\frac{1}{Re}\boldsymbol{\Gamma}_i h(\mathbf{x}-\mathbf{x}_i)\right\}=0, \tag{13}$$

where

$$\nabla^2\delta_\sigma(\mathbf{x}-\mathbf{x}_i)=\delta_\sigma(\mathbf{x}-\mathbf{x}_i)h(\mathbf{x}-\mathbf{x}_i). \tag{14}$$

Assuming that the blobs do not come close together and $\sigma \ll 1$ (this is justified due to the merging algorithm which is described in the next section) equation (13) implies that

$$\frac{\mathrm{d}\boldsymbol{\Gamma}_i}{\mathrm{d}t}=(\boldsymbol{\Gamma}_i\cdot\nabla)\mathbf{u}_i(\mathbf{x}_i, t)+\frac{1}{Re}\boldsymbol{\Gamma}_i h(0). \tag{15}$$

However, for the solution of these equations to be a weak solution of equations (1) and (2), we must replace[12] equation (15) by

$$\frac{\mathrm{d}\boldsymbol{\Gamma}_i}{\mathrm{d}t}=\left[\boldsymbol{\Gamma}_i\cdot\left(\frac{\nabla+\nabla^{\mathrm{T}}}{2}\right)\right]\mathbf{u}_i(\mathbf{x}_i, t)+\frac{1}{Re}\boldsymbol{\Gamma}_i h(0). \tag{16}$$

To account for turbulence effects in this equation, a proper eddy viscosity model must be used. This model will replace ν_0 by $\nu_0+\nu_T$ where ν_T is the turbulent eddy viscosity. In this paper we used the Smagorinski model[13] to compute this quantity. Thus,

$$\nu_T=C\sqrt{(S_{mn}S_{mn})}, \qquad S_{mn}=\frac{\delta u_m}{\delta x_n}+\frac{\partial u_n}{\delta x_m}, \quad m, n=1, 2, 3,$$

where n, m refer to the components of the velocity field and Einstein summation convention is used. C is a constant that is usually chosen to be proportional to the grid step. In our formulation we choose C to be proportional to the blob radius.

This model was chosen for two reasons. First, it is computationally simple. Second, it is one of the more commonly used in large eddy simulations of non-homogeneous turbulence. In fact, in the wake region the vortex method can be viewed as a large-eddy simulation on an irregular grid whose mesh points are the point vortices.

3. BOUNDARY CONDITIONS

In most of the literature, the parachute surface is considered as a solid wall.[1-4] Accordingly, the no-slip boundary conditions are used

$$u|_{\partial\Omega}=0, \tag{17}$$

where $\partial\Omega$ is the parachute surface. However, in actuality, the parachute should be considered as a membrane and appropriate boundary conditions for compliant surface should be used.[14,15]

We now describe how these two sets of boundary conditions were implemented.

On both sides (and edges) of the parachute surface, we introduce a set of grid points $\mathbf{x}_i$. The spacing Δ between these points was uniform except at the edges where $\Delta/2$ was used. On a staggered grid whose height from the surface is R_0, we define creation points for 'new vortices'. To satisfy the no-slip boundary conditions at each time step, the velocity field $\mathbf{u}_f$ due to the free vortices (away from the wall) is computed at the creation points $\mathbf{x}_i$. The circulation of the new vortices $\mathbf{\Gamma}_i$ at the creation points near the wall must be chosen then so that at each $\mathbf{x}_i$

$$\mathbf{u}_b(\mathbf{x}_i)+\mathbf{u}_f(\mathbf{x}_i)=0, \tag{18}$$

where $\mathbf{u}_b$ is the flow field generated by the new vortices. Using equation (10) this gives rise to a system of linear equations for the circulations $\mathbf{\Gamma}_i$ which must be solved at each time step. The size of this system of linear equations can be reduced by a proper use of the structure of the coefficient matrix. However, once these vortices are created, they are left to propagate with the flow so that in the next time step there are no vortices present at the creation points. Furthermore, vortices that crash into the wall are assumed to be absorbed by it.

When the parachute is considered as a compliant surface, the linearised boundary conditions due to Benjamin[14] and others[15] were used. Assuming that the free flow (far from the parachute) is in the z-direction, we can write these boundary conditions as

$$\begin{aligned} u(\mathbf{x}, t)+\eta\frac{\partial u}{\partial z}&=0,\\ v(\mathbf{x}, t)+\eta\frac{\partial v}{\partial z}&=0,\\ w(\mathbf{x}, t)+\eta\frac{\partial w}{\partial z}&=\frac{\partial \eta}{\partial t}, \end{aligned} \tag{19}$$

where $\mathbf{u}=(u, v, w)$ and η is the wall deflection in the z-direction (deflections in other directions are neglected). Assuming

$$\eta = kp, \qquad \left|\frac{\partial \eta}{\partial t}\right| \ll 1, \tag{20}$$

and using finite differences to approximate, equations (19) can be rewritten as

$$\mathbf{u}_{\mathrm{b}}(\mathbf{x}_i)+\mathbf{u}_{\mathrm{f}}(\mathbf{x}_i)+kp\left[\frac{\mathbf{u}_{\mathrm{cr}}(i)-\mathbf{u}(\mathbf{x}_i)}{R_0}\right]=0, \tag{21}$$

where $\mathbf{u}_{\mathrm{cr}}(i)$ is the velocity at the creation point near $\mathbf{x}_i$. This represents again a system of linear equations for $\mathbf{\Gamma}_i$ which must be solved at each time step.

In this algorithm, as described above, vortices are shed from the wall region at each time step. Hence, a mechanism is needed to control their number through a merger process. However, this merge algorithm must leave the flow field unchanged as far as possible. Following Spalart[9] the merge algorithm sorts the vortices according to their distance and strength (i.e. $\|\mathbf{\Gamma}_i\|$). If the number of vortices exceeds the computational bounds then those at the bottom of the list (and, hence, carrying minimal information about the flow) are merged with their nearest neighbours. The circulation and position of the new vortex are

$$\mathbf{\Gamma}=\mathbf{\Gamma}_i+\mathbf{\Gamma}_j,$$

$$\mathbf{x}=\frac{\mathbf{x}_i\|\mathbf{\Gamma}_i\|+\mathbf{x}_j\|\mathbf{\Gamma}_j\|}{\|\mathbf{\Gamma}_i\|+\|\mathbf{\Gamma}_j\|},$$

where i, j represent the indices of the vortices to be merged. In this way the total circulation of the flow is conserved, and the new vortex is placed at the centroid of the two merged vortices.

4. RESULTS

The algorithm described in the previous sections has been used to compute the flow field and drag coefficients of various parachute geometries. We present here the results for a circular plate of unit area. In all these simulations we used 3000 vortices and integrated the evolution equations for 3000 time steps using $\Delta t = 0{\cdot}005$. The computed drag coefficients represent averages over the value of this variable from iteration 1000 to 3000 (thus avoiding the initial transient state of the flow). However, even between these iteration limits the drag coefficient fluctuates widely due to (periodic) vortex shedding. Also, the pressure gradients were found to be large near the edge of the plate.

Figure 1 illustrates the dependence of the drag coefficient C_{D} on the thickness of the plate (while its radius remains constant). We see that the drag increases as the thickness Δ is reduced. This follows from the fact that as $\Delta \to 0$, the flow around the edges resembles a 'dipole flow' due to the strong gradients there. Figure 2 shows the dependence of C_{D} on the Reynolds number for circular plates with thickness of 0·02. As expected C_{D} remains (almost) constant as the vortex method fails to find computational evidence for the drag crisis. However, this failure is shared by all CFD algorithms in current use. Furthermore, as expected, the vortex method becomes unstable for $Re \simeq 10^4$. For comparison we observe that for circular plate the experimental value of C_{D} is 1·17. Thus, the discrepancy between the value shown in Figure 2 and the experimental value is about 7%. However, from Figure 1 we see that this discrepancy depends also on the plate thickness. This can be used as an indication for the expected accuracy of the method in similar geometries. Figure 3 demonstrates that as the attack angle is changed around 90°, the flow field undergoes

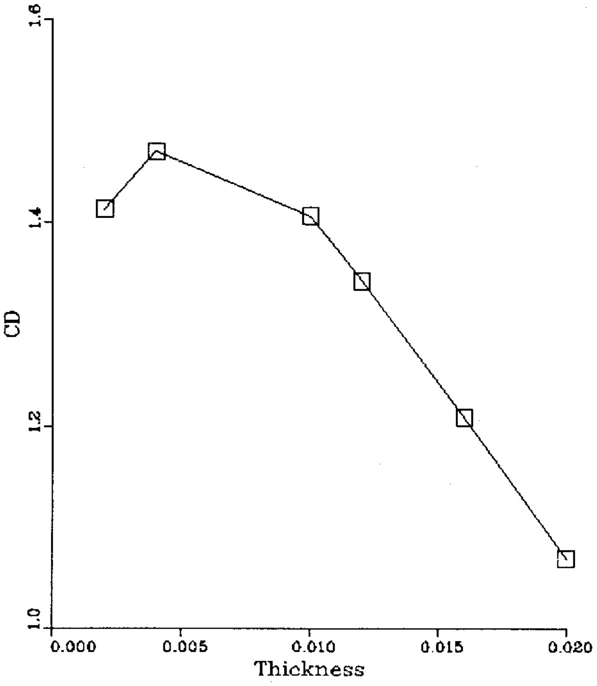

Figure 1. Circular plate *CD* vs. thickness

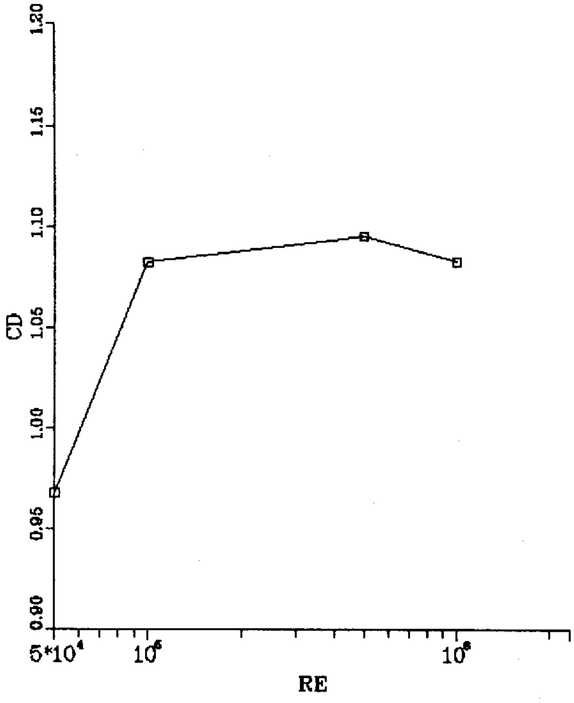

Figure 2. Circular plate *CD* vs. *RE*

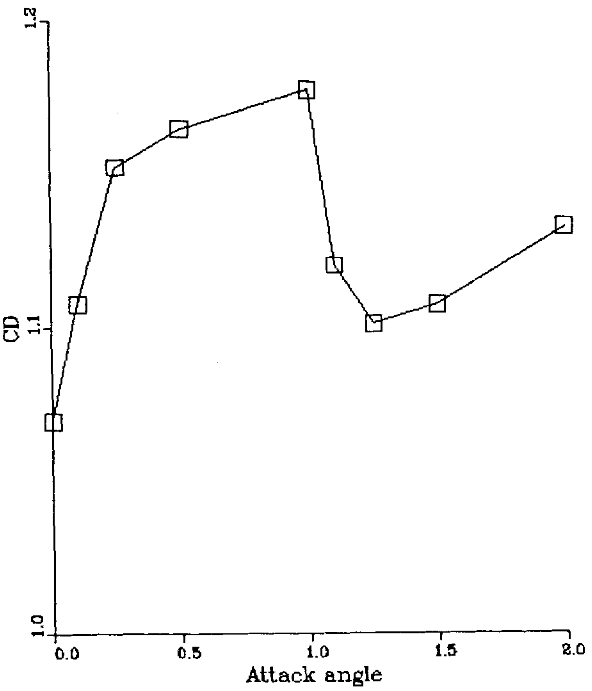

Figure 3. Circular plate *CD* vs. angle

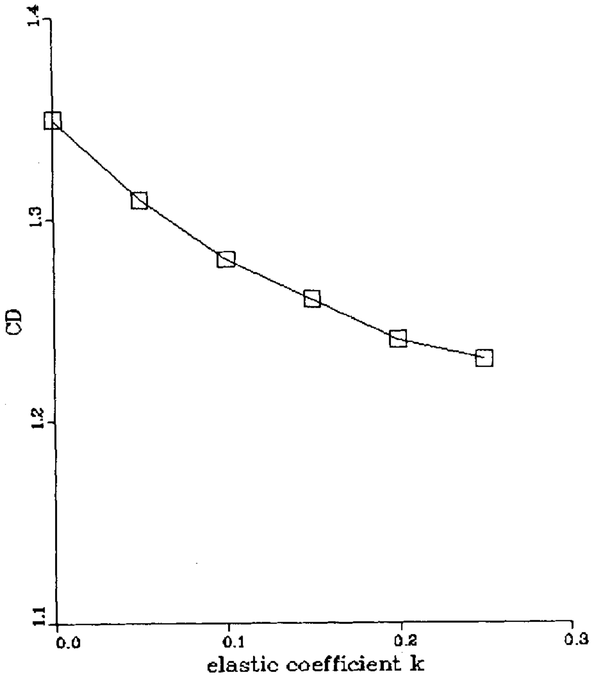

Figure 4. Circular plate *CD* vs. *k*

a bifurcation. This is due to the loss of symmetry in the flow field which in turn leads to oscillations in the value of the drag coefficient.

Figure 4 presents the dependence of the drag on the elastic constant k in equation (20). We see that for parachutes the increase in the drag due to turbulence effects is compensated to some extent by the elasticity of the boundary. This bifurcation in the drag for non-zero angles of attack requires further theoretical investigation.

5. SPECTRAL ANALYSIS OF THE VORTEX METHOD

In this section we examine the spectral dependence of the vortex method on the choice of the blob function. The derivations will be carried out for 2D as the arguments are more transparent in this case. However, the same results apply for the 3D case.

In 2D, equations (5), (7) and (10) are replaced, respectively, by

$$\mathbf{u}(\mathbf{x}, t)=\frac{1}{2\pi}\mathbf{e}_z\times\int\frac{(\mathbf{x}-\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|^2}\omega(\mathbf{x}', t)\,\mathrm{d}\mathbf{x}', \tag{22}$$

$$\delta_\sigma(\mathbf{x})=\frac{1}{\sigma^2}f\left(\frac{|\mathbf{x}|}{\sigma}\right), \tag{23}$$

$$\mathbf{u}(\mathbf{x}, t)=\frac{1}{2\pi}\mathbf{e}_z\times\sum\mathbf{\Gamma}_j(t)\frac{(\mathbf{x}-\mathbf{x}_j)}{|\mathbf{x}-\mathbf{x}_j|^2}F\left(\frac{|\mathbf{x}-\mathbf{x}_j|}{\sigma}\right), \tag{24}$$

where

$$F(r)=2\pi\int_0^r sf(s)\,\mathrm{d}s.$$

In the literature various forms of $f(x)$ were used. In KPD12 code[7] two such functions are mentioned

$$f_1(x)=\frac{1}{\pi(1+x^2)}, \tag{25}$$

$$f_2(x)=\begin{cases}\dfrac{1-x^2}{\pi} & x\leq 1,\\ 0 & x>1.\end{cases} \tag{26}$$

Beale and Majda (inspired by large eddy simulation methodology) suggested[16]

$$f_3(x)=\frac{1}{\pi}\mathrm{e}^{-x^2}. \tag{27}$$

Hald[17] used

$$f_4(x)=\frac{1}{3\pi x^2}[4J_2(2x)-J_2(x)], \tag{28}$$

where J_2 is the second-order Bessel function. Fishlov[8] showed that a sufficient condition for the stability of the vortex method is that the Fourier transform of $f(x)$ must be non-negative and used[8]

$$f_5(x)=\frac{1}{2\pi}[4\mathrm{e}^{-x^2}-\mathrm{e}^{-x^2/2}] \tag{29}$$

in the numerical computations.

From a spectral point of view, however, it is clear that $\hat{\omega}(\mathbf{k}, t)$—the spectral transform of $\omega(\mathbf{x}, t)$—is given by

$$\omega(\mathbf{k}, t)=\int\int e^{i\mathbf{k}\cdot\mathbf{x}}\,\omega(\mathbf{x}, t)\,d\mathbf{x}=\frac{1}{\sigma^2}\sum\Gamma_j(t)\,e^{i\mathbf{k}\cdot\mathbf{x}_j}\int_0^\infty\int_0^{2\pi} f\left(\frac{r_j}{\sigma}\right)e^{ikr_j\cos(\theta-\varphi)}r_j\,d\theta\,dr_j, \tag{30}$$

where $k=|\mathbf{k}|$ and $r_j=|\mathbf{x}-\mathbf{x}_j|$. But

$$J_0(kr)=\frac{1}{2\pi}\int_0^{2\pi} e^{ikr\cos(\theta-\varphi)}\,d\theta, \tag{31}$$

hence

$$\hat{\omega}(\mathbf{k}, t)=2\pi\sum\Gamma_j(t)\,e^{i\mathbf{k}\cdot\mathbf{x}_j}\int_0^\infty f(r)\,J_0(k\sigma r)\,r\,dr. \tag{32}$$

Thus, the spectral decomposition of $\omega(x, t)$ for all times is determined by the Bessel transform of the blob function $f(x)$.

To see the implications that this has for the flow field, we first rewrite equation (24) in the form

$$\mathbf{u}(\mathbf{x}, t)=-\frac{1}{2\pi}\sum\Gamma_j\frac{1}{r_j}F\left(\frac{r_j}{\sigma}\right)\mathbf{e}_{\theta_j}, \tag{33}$$

where $\mathbf{e}_{r_j}=\mathbf{r}_j/r_j$, $\mathbf{e}_{\theta_j}$ and $\mathbf{e}_z$ form a right-handed frame at $\mathbf{x}_j$. For the spectral decomposition of $\mathbf{u}$, we, therefore, have

$$\hat{\mathbf{u}}(\mathbf{k}, t)=-\frac{1}{(2\pi)}\sum\Gamma_j\,e^{i\mathbf{k}\cdot\mathbf{x}_j}\mathbf{e}_{\theta_j}\int_0^\infty\int_0^{2\pi} F\left(\frac{r_j}{\sigma}\right)e^{ikr_j\cos(\theta-\varphi)}\,d\theta\,dr_j, \tag{34}$$

and, therefore,

$$\hat{\mathbf{u}}(\mathbf{k}, t)=-\sum\Gamma_j\,e^{i\mathbf{k}\cdot\mathbf{x}_j}\mathbf{e}_{\theta_j}\int_0^\infty F\left(\frac{r}{\sigma}\right)J_0(kr)\,dr. \tag{35}$$

Thus, we proved that the spectral decomposition of the flow is determined by the Bessel transform of $F(r)/r$. In particular, if this function has a compact spectral support then the flow will have exactly the same support for all times.

For the blob functions given by equations (28) and (29), the Bessel transform $\tilde{f}_4(k)$ of f_4 is zero for $k>2$. Similarly, all other blob functions mentioned above have an effectively finite spectral range. That is $\tilde{f}_i(k)$ is practically zero for large k. Furthermore, as $f_i(k)$ are not constant, they introduce without any physical justification an *a priori* bias in the simulation. To neutralize it partially, one must use simulations with large number of vortices for long periods of time. (As a result, the phase factors in equation (35) will, hopefully, cancel the initial spectral bias.) One possible way to overcome this difficulty is to perform an unbiased vortex simulations using a blob function whose Fourier spectral decomposition is of the form ('top hat' functions)

$$\hat{f}_6(\mathbf{k})=\begin{cases}1, & |\mathbf{k}|<k_0,\\ 0, & \text{otherwise.}\end{cases} \tag{36}$$

The use of such a blob function with large enough k_0 and a large number of vortices ensures that the simulation is unbiased and has enough degrees of freedom to obtain a true spectral representation of the flow.

In 2D, the inverse of $\hat{f}_6(\mathbf{k})$ is

$$f_6(\mathbf{x})=\frac{k_0}{2\pi x}J_1(k_0|x|), \tag{37}$$

and, therefore,

$$F_6(\mathbf{x}) = 1 - J_0(k_0|\mathbf{x}|). \tag{38}$$

In 3D, the inverse of $\hat{f}(k)$ is given by

$$f_7(\mathbf{x}) = \frac{4\pi}{r^3}\int_0^{k_0 r} j_3(s)s^2\,\mathrm{d}s = \frac{4\pi}{r^3}\left\{k_0 r \sin k_0 r - 15\frac{\sin k_0 r}{k_0 r} + 7\cos k_0 r + 8\right\}, \quad r = |\mathbf{x}|, \tag{39}$$

where $j_3(x)$ is the third-order spherical Bessel function.

Using $f_6(\mathbf{x})$ with different k_0, we simulated the flow normal to a 2D plate at $Re = 5 \times 10^5$ with the vortex method. In all these simulations, we used 2000 vortices and integrated the evolution equations for 3000 time steps using $\Delta t = 0{\cdot}005$. The results for the computed drag coefficients are shown in Figure 5. These coefficients represent averages over the value of this variable from iteration 1000 to 3000. From this figure we see that as k_0 increases and the blob function include wave numbers which are critical to the correct simulation of the flow, there is sudden change in the value of the drag on the plate.

In view of these results, it is natural to inquire as to the existence of the optimal blob function. However, it seems improbable that universal optimal blob function exists. Nevertheless, from a practical point of view one may tailor the choice of $f(x)$ to the 'prototype' problem at hand. This can be done by simulating accurately one of the typical flows under consideration and computing the actual spectral decomposition of ω in the desired range of Reynolds numbers. The derived blob function can be used then in the simulation of similar flows.

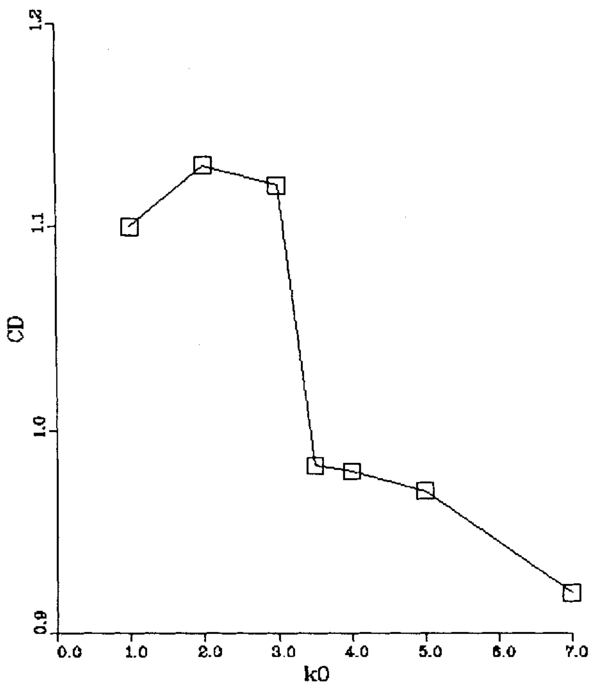

Figure 5. 2D Plate CD vs. k_0

6. CONCLUSIONS

This paper presented a coupling between the methodologies of large-eddy simulation and the vortex method. This coupling led to a stable algorithm which allowed us to explore various dynamical aspects of the flow around very thin structures such as parachutes. This does not seem to be possible in grid-based algorithms at the present time. The algorithm is computationally intensive but, with advances in computer hardware, it is hoped that a substantial increase in the number of vortices used in these simulations will be possible. This will enable us to model and resolve other aspects of turbulent flows such as flow separation and the drag crisis. Furthermore, the optimal choice of the blob function for the simulation of flows with high Reynolds numbers merits further investigation.

We also showed in this paper that the spectral contents of a vortex simulation is strongly coupled to the choice of the blob function. A universal optimal blob function may not exist. However, for each class of flows it must be possible to determine such a function. In particular, for turbulent flows, Kolmogorov 5/3 law can be used to determine this function in the inertial range. Failing this, the functions $f_6(x)$ in 2D and $f_7(x)$ in 3D with large enough k_0 seem the most appropriate. These functions contain unbiased range of wave numbers and the amplitude of each $\mathbf{k}$ will adjust to the true spectral decomposition of the flow if enough vortices are used in the simulation.

ACKNOWLEDGEMENTS

Part of this research was carried out at the U.S. Army Natick RD&E Center in Natick, MA. The author would like to thank Dr. E. C. Steeves for his kind help and encouragement. Some of the simulations were carried out at the Cornell Supercomputer Facility.

REFERENCES

1. D. J. Cockrell, 'The aerodynamics of parachutes', *Agardograph No. 295*, Agard, Essex, U.K. 1987.
2. E. C. Steeves, 'Analysis of decelerators in motion using computational fluid dynamics',, *AIAA 10th Aerodynamic Decelerator System Technology Conference*, Cocoa Beach, FL, 1989, pp. 269–278, No. 89–0931.
3. M. Humi, 'Drag computation by vortex methods', *AIAA J. Aircraft*, **29**, 819–822 (1992).
4. S. K. Ibrahim, 'Potential flow and added mass for idealized hemispherical parachutes', *AIAA J. Aircraft*, **4**, 96–100 (1967).
5. N. C. Markatos, 'The mathematical modeling of turbulent flows', *Appl. Math. Modelling*, **10**, 190–236 (1986).
6. A. Leonard, 'Computing three dimensional incompressible flow with vortex elements', *Ann. Rev. Fluid Mech.*, **17**, 253–564 (1985).
7. A. J. Chorin, 'Vortex sheet approximation of boundary layers', *J. Comput. Phys.*, **27**, 428–436 (1978).
8. D. Fishelov, 'A new vortex scheme for viscous flows', *J. Comput. Phys.*, **86**, 211–224 (1990).
9. P. R. Spalart, 'Vortex methods for separated flows', *NASA TM No. 1000068*.
10. *Proc. AMS Conference Vortex Dynamics and Vortex Methods*, University of Washington, Seattle, 1990, American Math. Soc., Providence, R.I., U.S.A., 1991.
11. G. Winkelmans and A. Leonard, 'Improved vortex methods for three dimensional flows' in R. E. Caflisch (ed). *Mathematical Aspects of Vortex Dynamics*, SIAM, 1989.
12. C. Anderson and C. Greengrad, 'On vortex methods', *SIAM J. Numer. Anal.*, **22**, 413–440 (1985).
13. J. Smagorinski, 'General circulation experiments with the primitive equations', *Mon. Weather Rev.*, **91**, 99–164 (1983).
14. T. B. Benjamin, 'Fluid flow with flexible boundaries', *Proc. 11th Int. Cong. in Appl. Math*, Munich, Germany, 1964, pp. 109–126.
15. M. T. Lendhahl and R. E. Kaplan, 'Effects of compliant wall on boundary layer stability', *Agardograph No. 91–353*, 1965, pp. 363–394.
16. T. Beale and A. Majda, 'High order accurate vortex methods with explicit velocity kernels', *J. Comput. Phys.*, **58**, 188–208 (1985).
17. O. H. Hald, 'Convergence of vortex methods for Euler's equations', *SIAM J. Numer. Anal.*, **24**, 538–582 (1987).