SIAM J. SCI. COMPUT.
Vol. 24, No. 6, pp. 1864–1878


# A FAST, TWO-DIMENSIONAL PANEL METHOD*

PRABHU RAMACHANDRAN†, S. C. RAJAN†, AND M. RAMAKRISHNA†

**Abstract.** A new two-dimensional panel technique has been developed to solve Laplacian flows, which eliminates the edge effect present in traditional panel methods. Such a method is very useful for applications where the velocity induced by the panels is required at arbitrary locations. Particle based flow solvers are a prime example. The method, however, requires considerably more computational effort. In this paper the method is modified to improve computational efficiency by adapting the fast multipole algorithm for the panel method. Significant improvement in computational efficiency is obtained while ensuring that the edge effects are eliminated.

**Key words.** panel method, edge effect, fast multipole method

**AMS subject classifications.** 31A99, 34B60, 35J05, 65E05, 65Y99, 76M15

**PII.** S1064827500374662

## 1. Introduction.

Panel methods provide an elegant methodology for solving a class of flows past arbitrarily shaped bodies in both two and three dimensions. The basic idea is to discretize the body in terms of a singularity distribution on the body surface, satisfy the necessary boundary conditions, find the resulting distribution of singularity on the surface, and thereby obtain fluid dynamic properties of the flow. The body geometry is represented in terms of smaller subunits called panels, hence the name "panel" method. In two dimensions the panels are usually straight lines, and in three dimensions planar elements are used. The singularities used can be either sources, doublets, or vortices. Each panel is constructed to have some kind of singularity distribution. Depending on the accuracy, computational speed and other factors one can use constant, linear, parabolic, or even higher orders of distribution of the singularity on each panel. The number of panels that represent the body can also be varied. The actual singularity distribution is initially unknown, but by enforcing the boundary conditions on the body, it is possible to solve for them. The boundary conditions can be represented in terms of the velocity field, called the Neumann condition, or in terms of the potential inside the body, called the Dirichlet condition. Hess and Smith [8] laid the foundation for the source panel method. The idea of the vortex panel method is due to Martensen [14] and is extended by Lewis [10]. Katz [9] gives an excellent, comprehensive overview of panel methods in general (in both two and three dimensions).

The main advantage of the panel method is that it is relatively easy to formulate and compute. The panel methods are also "grid free" and represent the geometry as accurately as possible. Rajan [16] also shows that the vortex based panel methods are capable of explaining both the kinematic motion of the rigid body and the fluid flow. Unlike finite difference methods panel methods are, however, a little restrictive in their applicability. Typically panel methods are used to solve Laplacian flows, which they solve very accurately and efficiently. There are techniques to handle compressible flows past thin bodies by using the Prandtl–Glauert transformation and possible extensions to supersonic flows by taking the domain of dependence and range of

*Received by the editors June 26, 2000; accepted for publication (in revised form) February 22, 2002; published electronically May 2, 2003.

http://www.siam.org/journals/sisc/24-6/37466.html

†Department of Aerospace Engineering, IIT-Madras, Chennai 600 036 India (prabhu@aero.iitm.ernet.in, rajan@aero.iitm.ernet.in, krishna@aero.iitm.ernet.in).

influence into consideration. Viscous effects can be included, but only approximately, as the boundary layer. However, for low speed aerodynamics panel methods are an excellent choice. There are also an important class of numerical methods where the accurate solution of Laplace's equations is of great importance, namely the particle based vortex blob methods. These methods are capable of accurately simulating the viscous incompressible flow past arbitrary bodies. They require a fast and accurate velocity evaluation due to the body at a large number of arbitrarily positioned points. In [5, 11, 18, 19, 20] different panel methods are used to compute the velocity field due to the body. [5] and [18] use a parabolic panel method where the geometry of the body is discretized into parabolic segments. Others use linear panel methods. The present work is concerned with panel methods in two-dimensional, incompressible flows.

Traditionally, panel methods have been used to compute aerodynamic loads on bodies moving in a fluid. Finding the velocity at an arbitrary point in the flow is not usually the focus of such a computation. In certain applications such a velocity field is important. Commonly used panel methods have severe problems in determining the velocity field in the vicinity of the panel edges. This problem is called the "edge effect." Ramachandran, Rajan, and Ramakrishna [17] demonstrate for the case of flat panels using any kind of singularity distribution, in two dimensions, that this error increases indefinitely as one approaches the edge of the panel. They also show that this error is due to the discretization of the body geometry in terms of linear panels and provide a means to circumvent this problem by using cubic panels to approximate the body geometry. The equations for such a panel method are also derived. The problem with this methodology is that it becomes computationally more expensive. It can be seen in [5] that the computation of the velocity due to the parabolic panels on the $N$ vortex blobs is almost half as expensive as the computation of the velocity of the blobs on themselves (which is an $O(N \log(N))$ process). This is so, in spite of the fact that the number of panels is much smaller than the number of blobs. In this work the panel method is accelerated by modifying the adaptive fast multipole method (AFMM) due to Carrier, Greengard, and Rokhlin [4]. The equations for the application of the FMM to this problem are derived. It is possible to apply Anderson's technique [2] directly to the cubic panel method. However, in this study the AFMM has been used. This is because it is relatively easy to adapt any existing AFMM when panels are involved, as in the present case. The work also enables for future comparisons between Anderson's technique [2] and the AFMM [4] as applied to panels.

In the vicinity of the panel the cubic panel method is used and thus the edge effect is dealt with. By doing this it is possible to evaluate the velocity very efficiently without sacrificing accuracy.

In what follows the equations derived in [17] are quickly recapitulated and the accuracy of the method is demonstrated. The AFMM is modified to suit this problem and the equations governing the multipole and local expansions and their corresponding transfer functions are derived. Some useful and simple modifications to the original AFMM that are beneficial to the current problem are proposed. Finally results comparing the original cubic and flat panel methods and the currently developed accelerated panel technique are presented.

**2. The cubic panel method.** In [17] the authors choose the singularity distribution to be a linear distribution of vorticity. For simplicity, the complex number notation is used. For a flat panel (illustrated in Figure 1), $\gamma$ is the vorticity at any point on a flat panel and $\gamma_1$ and $\gamma_2$ are the values of $\gamma$ at the ends of the panel. The velocity at an arbitrary point $z$ in the fluid due to a single panel oriented at an

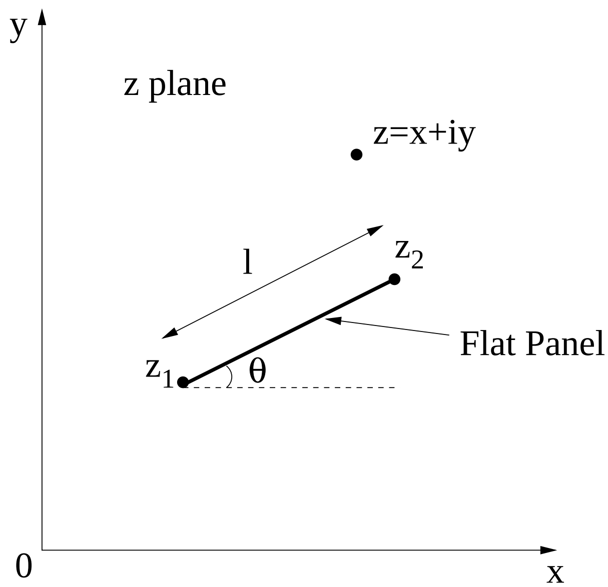


FIG. 1. *Sketch of a single flat panel in the complex plane.*

arbitrary angle to the $x$-axis can be easily found as

$$\begin{aligned} V(z) &= u - iv \\ &= \frac{-i}{2\pi}\left\{\gamma_1\left[\left(\frac{z'}{l}-1\right)\ln\left(\frac{z'-l}{z'}\right)+1\right]\right. \\ &\qquad \left.-\gamma_2\left[\frac{z'}{l}\ln\left(\frac{z'-l}{z'}\right)+1\right]\right\}e^{-i\theta}, \end{aligned} \tag{2.1}$$

where $z' = (z - z_1)e^{-i\theta}$, where $i = \sqrt{-1}$.

It is clear from the above equation that the velocity diverges at the end points of the panel. The effect of this singularity is not nullified by any of the adjacent panels due to the linear discretization of the body geometry and the consequent mismatch of slopes of each adjacent panel. The authors in [17] show that this error can be nullified if the body geometry is discretized in terms of cubic panels by matching the slopes of the adjacent panels. The equation for the panel with chord oriented along the $x$-axis is assumed to be $\eta = a_1\zeta + a_2\zeta^2 + a_3\zeta^3$, where $\zeta$ and $\eta$, respectively, are the $x$ and $y$ coordinates of the panel surface, as shown in Figure 2. The distribution of vorticity is assumed to be linear with respect to the chord of the cubic, namely $\zeta$. The vorticity is distributed on the chord in order to keep the integral tractable. The equation for the velocity due to such a panel at a point $z$ is given as

$$V(z') = \frac{k}{2\pi a_3}\int_0^l \frac{(\zeta + \frac{\gamma_1}{k})}{\left(\zeta^3 + \frac{a_2}{a_3}\zeta^2 + \frac{(a_1-i)}{a_3}\zeta + \frac{iz'}{a_3}\right)}d\zeta, \tag{2.2}$$

where $k = (\gamma_2 - \gamma_1)/l$. The integral in (2.2) is not straightforward to evaluate. In order to solve it the cubic in the denominator is reduced as follows:

$$\zeta^3 + \frac{a_2}{a_3}\zeta^2 + \frac{(a_1-i)}{a_3}\zeta + \frac{iz'}{a_3} = (\zeta-a)(\zeta-b)(\zeta-c), \tag{2.3}$$

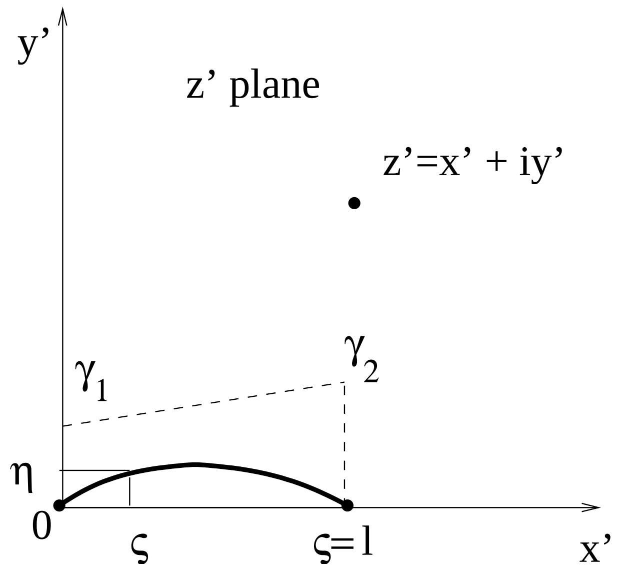


FIG. 2. *Sketch of a single cubic panel having a chord length $l$ in the $z'$ plane.*

where $a$, $b$, and $c$ are the complex cube roots of the cubic, and these roots can be computed numerically. The numerical procedure is standard and is given in [15]. Given the roots, one can find the velocity field due to the panels by performing the integration using the method of partial fractions. After integration and simplification the velocity due to the cubic panel is obtained as

$$
\begin{aligned}
V(z) = \frac{-\gamma_2}{2\pi a_3 l} &\left[ \frac{a \log\left(\frac{a-l}{a}\right)}{(a-c)(a-b)} + \frac{b \log\left(\frac{b-l}{b}\right)}{(b-c)(b-a)} + \frac{c \log\left(\frac{c-l}{c}\right)}{(c-a)(c-b)} \right] e^{-i\theta} \\
- \frac{\gamma_1}{2\pi a_3 l} &\left[ \frac{(l-a) \log\left(\frac{a-l}{a}\right)}{(a-c)(a-b)} + \frac{(l-b) \log\left(\frac{b-l}{b}\right)}{(b-c)(b-a)} + \frac{(l-c) \log\left(\frac{c-l}{c}\right)}{(c-a)(c-b)} \right] e^{-i\theta},
\end{aligned}
\tag{2.4}
$$

$a_1$, $a_2$, and $a_3$ are the coefficients of the cubic panel representation that depend on the slopes of the panel at each end point and the chord length, $l$. It must be noted that the cube roots involve $z$ implicitly. $\theta$ is the angle between the chord of the cubic panel and the $x$-axis and, as for the linear panel case, $z' = (z - z_1)e^{-i\theta}$.

The method produces very accurate results [17] when compared to the linear panel technique and eliminates the edge effect. The error in the computed solution is defined as follows:

$$
E = \left( \frac{\sum_{i=1}^{N} |v_i - \tilde{v}_i|^2}{\sum_{i=1}^{N} |v_i|^2} \right)^{1/2}, \tag{2.5}
$$

where $v_i$ is the exact velocity, $\tilde{v}_i$ is the computed velocity, and $N$ is the number of points over which the error is averaged. The flow past a circular cylinder of unit radius is considered. The exact solution is compared with that obtained by using 200 flat and cubic panels. A ring of about 1000 points at a distance $r$ from the surface of the cylinder is considered and the value of $E$ computed. In Figure 3 plots of the error $E$ versus $r$ for both the flat and cubic panel methods are given. As is evident, the

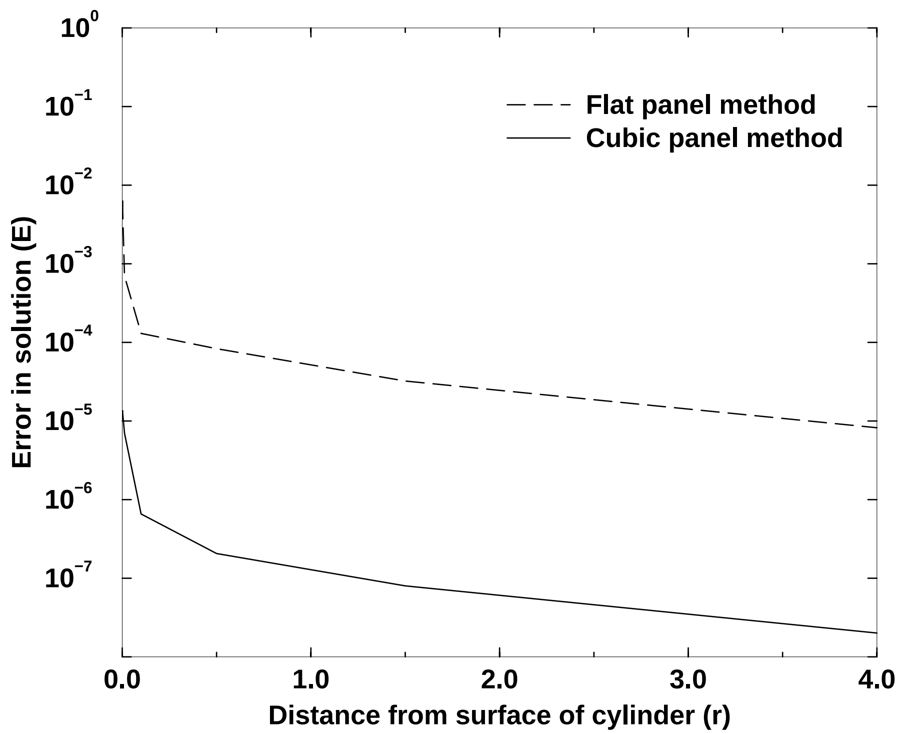


FIG. 3. *Comparison of the error $E$ versus distance from the surface of a circular cylinder obtained by using flat and cubic panels. $E$ is computed using* (2.5).

cubic panel is at least two orders of magnitude more accurate. Even extremely close to the surface, the error is relatively small. In the case of the flat panels the error is much larger and grows without bound as one approaches the edge of the panels.

The problem with the cubic panel method is that for the evaluation of each velocity, the three roots of a complex cubic equation must be computed. This makes the method computationally expensive. As mentioned earlier, certain computations require the velocity field due to the body at a large number of arbitrarily positions. In a sample computation a set of 10000 points is considered and the velocity due to a 400 paneled circle is computed at each of these points by the flat panel technique and also by the cubic panel method. The cubic panel method is found to be about 4 times slower. There are two possible ways to make this computationally less intensive. The first is to use the cubic panel technique in the vicinity of each panel and then use the linear panel method when one is sufficiently far away from the panel. The notion of "sufficient" clearly depends on the accuracy required. This can at best achieve a speed equal to that of the linear panel method, which is a factor of around four. This is not acceptable. The second is to use an advanced method, like the fast summation techniques, which are used for $N$ body computations.

**3. Fast summation techniques.** $N$ body simulations require one to evaluate the effect of $N$ particles on each other. Since this is an $O(N^2)$ process, as $N$ becomes very large, the computation becomes prohibitively expensive. There are a large number of methods developed that reduce the computation to either an $O(N \log(N))$ or an $O(N)$ process. The basic idea behind all the schemes is to approximate a cluster of particles that is far away into a single computational entity. Some of the relevant articles are mentioned here. The hierarchical solver [3] for force computations, the particle in cell methods, and the method of local corrections [1] are relatively older techniques. The methods based on multipole expansions due to Greengard and Rokhlin [7] and the adaptive FMM due to Carrier, Greengard, and Rokhlin [4] are probably the most efficient in two dimensions. The adaptation of the FMM using Poisson's formula due to Anderson [2] and the FMM based on the Laurent series due to van Dommelen and Rundensteiner [21] are other FMM based schemes. Lustig, Rastogi, and Wagner [12] detail a technique that modifies and speeds up the FMM by telescoping the multipole method by using Chebyshev economization. The method due to Draghicescu and Draghicescu [6] is based on Taylor expansions. There are recent advances on the method due to Anderson [2] by Makino [13]. In the following work, the FMM due to Carrier, Greengard, and Rokhlin [4] is considered due to its adaptive nature and high efficiency. This method is $O(N)$ and is highly efficient for arbitrary particle distributions.

The panel method is quite different from the $N$ body problem in that for two-dimensional flows there usually aren't a large number of panels as compared to the typical $N$ body problems. In the case of vortex based flow solvers, the largest number of particles are the vortex blobs and one has to compute the velocity of the panels on these particles efficiently. If there are $M$ panels and $N$ particles, the number of computational operations scale as $O(M * N)$, which is different from the $O(N^2)$ operations for the particles themselves, since $M << N$. However, an AFMM-like technique [4] reduces the $O(N^2)$ interaction of the vortex blobs among themselves to an $O(N)$ process. Then the computational time of the panel method can be much larger than the blob velocity computation. In a particular implementation for the vortex blobs it is seen that the time for computation of the velocity of 10000 vortex blobs using the AFMM is anywhere between 4.5 to 16 times (depending on the type of the blob used) less than the time taken to evaluate the velocity of 400 linear panels on the 10000 particles. Hence, the computational efficiency of the panel method needs to be improved. The AFMM appears very attractive for this purpose. There are five important quantities which are to be computable for a cluster of panels, if one is to perform a fast multipole summation. They are

- direct computation of the velocity field;
- evaluation of the multipole expansion about a given center;
- transferring the multipole to a different center;
- conversion of the multipole expansion to a local expansion about a center that converges in a radius around the center;
- transfer the local expansion to another center.

If these operations can be performed, one can use an FMM to greatly speed up the computation. The above operations are performed depending on the distance of the point of evaluation to the location of the cluster of particles. This requires a hierarchical mesh that divides the various regions based on proximity. If the particles are very close to the point of evaluation, then the computation is performed directly. If the particles are sufficiently far away, the multipole representation of those particles are converted in the region of interest to a local expansion and this is evaluated to give a velocity/force/potential field due to the far away particles. Similar operations are performed for intermediate particles. For the actual methodology of the FMM with all the theoretical details the reader is referred to [7] and for the adaptive implementation to [4].

**4. The fast panel method.** The cubic panel equation (2.4) for the velocity due to a single panel is complicated, involving the point $z$ at which the velocity is evaluated implicitly in the form of complex cube roots of (2.3). On the other hand the equation (2.1) due to the linear panels is much easier to handle. This equation will be considered for the AFMM applied to the panel method. The problem with this method is that its accuracy is very poor especially in the vicinity of the edge of the panel. In the present method the edge effect is eliminated by performing the direct velocity evaluation due to a panel by using the cubic panel equation, (2.4).

For the multipole expansion and other expressions, the equation for the velocity due to a linear panel, (2.1), is considered. Upon simplification this reduces to

$$V(z) = u - iv = \frac{i}{2\pi}A, \tag{4.1}$$

where $A$ is given as

$$\begin{aligned} A &= (\gamma_2' - \gamma_1')\left[\frac{(z-z_2)}{l'}\ln(z-z_2) - \frac{(z-z_1)}{l'}\ln(z-z_1)\right] \\ &\quad + \gamma_2'\ln(z-z_2) - \gamma_1'\ln(z-z_1) + (\gamma_2' - \gamma_1'), \end{aligned} \tag{4.2}$$

where $\gamma_2' = \gamma_2 e^{-i\theta}$, $\gamma_1' = \gamma_1 e^{-i\theta}$, and $l' = le^{i\theta}$. The terms $z_1$, $z_2$, and $\theta$ are as in Figure 1. This form is very elegant from the perspective of the FMM. The multipole expansion and other expressions for the terms of the form $q_i \ln(z - z_i)$ are already derived by Greengard and Rokhlin [7]. All that is required is to derive similar expressions for the terms of the form $q_i(z - z_i)\ln(z - z_i)$. With this, the FMM can be applied to the linear vortex panel method. In the following the required expressions are derived.

A particle of singularity strength $q$ located at $z_1$ is considered. The "velocity" that it induces at a point $z$ is given as

$$V(z) = q(z - z_1)\ln(z - z_1). \tag{4.3}$$

This can be expressed as a multipole expansion about a circle centered at $z_0$ as follows:

$$\begin{aligned} B(z) &= q(z - z_1)\ln(z - z_1) \\ &= q(z - z_0 - (z_1 - z_0))\left[\ln(z - z_0) - \sum_{k=1}^{\infty}\frac{1}{k}\left(\frac{z_1 - z_0}{z - z_0}\right)^k\right] \\ &= q(z - z_0)\left[\ln(z - z_0) - \sum_{k=1}^{\infty}\frac{1}{k}\left(\frac{z_1 - z_0}{z - z_0}\right)^k\right] \\ &\quad - q\left[(z_1 - z_0)\ln(z - z_0) - \sum_{k=1}^{\infty}\frac{(z_1 - z_0)^{k+1}}{k(z - z_0)^k}\right]. \end{aligned}$$

If there are $m$ singularities of strength $q_i$ located at positions $z_i$ inside a circle of radius $R$ with center $z_0$, then the total velocity is found from the above equation as a simple sum of the various singularities, which after some simplification reduces to

$$\begin{aligned} B(z) &= \sum_{i=1}^{m} q_i(z - z_i)\ln(z - z_i) \\ &= a_1 + (a_0(z - z_0) - d_0)\ln(z - z_0) + \sum_{k=1}^{\infty}\frac{a_{k+1} - d_k}{(z - z_0)^k}, \end{aligned} \tag{4.4}$$

where

$$a_0 = \sum_{i=1}^{m} q_i, \qquad a_k = \sum_{i=1}^{m} \frac{-q_i}{k}(z_i - z_0)^k \quad \text{for} \quad k > 0$$

and

$$d_0 = \sum_{i=1}^{m} q_i(z_i - z_0), \qquad d_k = \sum_{i=1}^{m} \frac{-q_i}{k}(z_i - z_0)^{k+1} \quad \text{for} \quad k > 0.$$

Equation (4.4) is the multipole expansion for a set of $m$ singularities that have a "velocity" field given by (4.3) and are located in a circle centered at $z_0$.

In order to transfer the multipole expansion about the center $z_0$ to any other center, (4.4) is expanded to obtain a multipole expansion about the origin, as performed in [4] and [7]. Using this it is easy to generalize the transfer to any center. Hence, expanding (4.4) about the origin and carrying out simplifications results in the following:

$$\begin{aligned} B(z) &= a_1 - a_0 z_0 + (a_0 z - a_0 z_0 - d_0)\ln z \\ &\quad + \sum_{l=1}^{\infty} \frac{1}{z^l}\left[\left(\sum_{k=1}^{\infty} z_0^{l-k} \begin{pmatrix} l-1 \\ k-1 \end{pmatrix} (a_{k+1} - d_k)\right) + \frac{z_0^l}{l}\left(\frac{a_0 z_0}{l+1} + d_0\right)\right]. \end{aligned} \tag{4.5}$$

This expression governs the transfer of the multipole expansion given by (4.4) and is the equivalent to the Lemma 2.3 in [7].

Given a multipole expansion about a center one has to be able to convert it to a local (Taylor) expansion in a circular region of analyticity. Therefore, from (4.4) a power series representation has to be obtained. (4.4) can be split into two parts. The first part given below can be reduced as

$$\begin{aligned} a_1 + (a_0(z - z_0) - d_0)\ln(z - z_0) &= a_1 - (a_0 z_0 + d_0)\ln(-z_0) \\ &\quad + z\left(a_0 \ln(-z_0) + \frac{1}{z_0}(a_0 z_0 + d_0)\right) \\ &\quad - \sum_{k=2}^{\infty} \frac{1}{k}\left(\frac{z}{z_0}\right)^k \left(\frac{a_0 z_0}{k-1} - d_0\right), \end{aligned} \tag{4.6}$$

and the second part can be reduced as

$$\begin{aligned} \sum_{k=1}^{\infty} \frac{a_{k+1} - d_k}{(z - z_0)^k} &= \sum_{k=1}^{\infty} \frac{a_{k+1} - d_k}{z_0^k}(-1)^k + \frac{z}{z_0}\left(\sum_{k=1}^{\infty} \frac{a_{k+1} - d_k}{z_0^k} k(-1)^k\right) \\ &\quad + \sum_{l=2}^{\infty}\left(\frac{z}{z_0}\right)^l \left(\sum_{k=1}^{\infty} \frac{a_{k+1} - d_k}{z_0^k} \begin{pmatrix} l+k-1 \\ k-1 \end{pmatrix} (-1)^k\right). \end{aligned} \tag{4.7}$$

The right-hand sides of (4.6) and (4.7) are obtained using the result from Lemma 2.4 in [7]. Combining (4.6) and (4.7) and arranging the terms based on powers of $z$, we obtain the following power series:

$$
\begin{aligned}
B(z) &= a_1 - (a_0 z_0 + d_0)\ln(-z_0) + \sum_{k=1}^{\infty} \frac{a_{k+1} - d_k}{z_0^k}(-1)^k \\
&\quad + z\left[a_0 \ln(-z_0) + \frac{1}{z_0}\left(a_0 z_0 + d_0 + \sum_{k=1}^{\infty} \frac{a_{k+1} - d_k}{z_0^k} k(-1)^k\right)\right] \\
&\quad + \sum_{l=2}^{\infty}\left(\frac{z}{z_0}\right)^l \left[\sum_{k=1}^{\infty} \frac{a_{k+1} - d_k}{z_0^k}\binom{l+k-1}{k-1}(-1)^k - \frac{1}{l}\left(\frac{a_0 z_0}{l-1} - d_0\right)\right].
\end{aligned} \tag{4.8}
$$

This equation corresponds to the final equation in Lemma 2.4 in [7]. The first term is a constant, the second is linear in $z$, and the third term includes all the higher powers of $z$.

The last equation needed for the FMM is the expression that transfers the Taylor's series about a center $z_0$ to that about any other center. Since this is nothing but a transfer of a generic Taylor's series, Lemma 2.5 in [7] is applicable here and is reproduced below:

$$
\sum_{k=0}^{n} a_k (z - z_0)^k = \sum_{l=0}^{n} z^l \sum_{k=l}^{n} a_k \binom{k}{l} (-z_0)^{k-l} \tag{4.9}
$$

With (4.4), (4.5), (4.8), and (4.9) the FMM can be used for a singularity that has a velocity field expressed by (4.3). The reference [7] contains the corresponding expressions for a singularity that behaves as $q_i \ln(z - z_i)$. Combining their results with the ones derived above, it is possible to use the FMM for the velocity due to a panel that has a linear geometry.

It can be seen from (4.1) and (4.2) that the velocity field due to the panels can be represented as the sum of a set of singularities that are of the form $z \ln z$ and $\ln z$ that are placed at the end points, $z_1$ and $z_2$, of the panels. There are two singularities of strength $(\gamma_2' - \gamma_1')/l'$ and $(\gamma_1' - \gamma_2')/l'$ of the form $z \ln z$ , and two more singularities of the form $\ln z$ with strengths $\gamma_2'$ and $-\gamma_1'$ placed, respectively, at the ends of the panel, $z_2$ and $z_1$. Hence for a single panel, it can be easily shown that

$$
a_0 = 0, \tag{4.10a}
$$

$$
a_k = \frac{-(\gamma_2' - \gamma_1')}{k l'}\left((z_2 - z_0)^k - (z_1 - z_0)^k\right), \tag{4.10b}
$$

$$
d_0 = 0, \tag{4.10c}
$$

$$
\begin{aligned}
d_k = \frac{1}{k}\Bigg[&\frac{(\gamma_1' - \gamma_2')}{l'}\left((z_2 - z_0)^{k+1} - (z_1 - z_0)^{k+1}\right) \\
&+ \gamma_2'(z_2 - z_0)^k - \gamma_1'(z_1 - z_0)^k\Bigg],
\end{aligned} \tag{4.10d}
$$

$$
\begin{aligned}
a_{k+1} - d_k = \frac{1}{k}\Bigg[&\frac{(\gamma_2' - \gamma_1')}{l'(k+1)}\left((z_2 - z_0)^{k+1} - (z_1 - z_0)^{k+1}\right) \\
&- \gamma_2'(z_2 - z_0)^k + \gamma_1'(z_1 - z_0)^k\Bigg],
\end{aligned} \tag{4.10e}
$$

where $k > 0$. Since $a_0$ and $d_0$ are zero for a single panel, (4.4), (4.5), and (4.8) simplify greatly. It is now possible to use the FMM for any number of panels by adding to the above coefficients, (4.10), the effect of each panel and substituting the result in (4.4), (4.5), (4.8), and (4.9). In order to numerically solve this, the series expansions

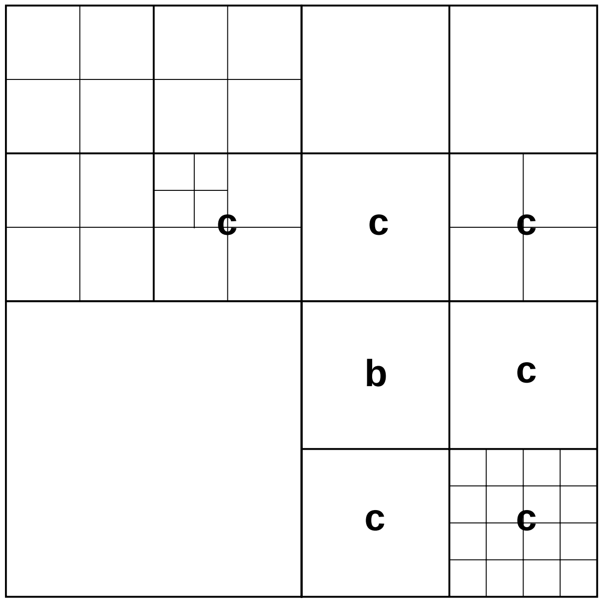


Fig. 4. *Sketch of the colleagues of cell* b *for an arbitrary distribution of particles. All cells marked* c *are the colleagues.*

for the multipole method need to be truncated up to some order $p$ depending on the accuracy desired. In [4], given an accuracy $\epsilon$, the number of terms to be considered in the multipole expansion is given as $p = \ln_2 \epsilon$. Since there is a multiplicative factor of $z$ in the case of a panel, for the same accuracy to be obtained, $p$ must be chosen as $p = 1 + \ln_2 \epsilon$.

**5. Numerical implementation.** Carrier, Greengard, and Rokhlin [4] provide most of the important details in order to implement the AFMM. They prove the accuracy of the method and also show that the method is $O(N)$. In the numerical implementation of the AFMM, the first and most important step is to create a hierarchical mesh that contains all the panels and particles. This is done by using a quad tree structure. The domain is split up into cells that contain the particles and panels. Depending on the number of particles and panels in a particular cell, it may be split into four daughter cells. An unsplit cell is called a *childless* or *leaf* cell. A split cell is either called a *parent* or a *branch*. The cell splitting can also be controlled by the size of the cell. It is useful to define two kinds of particles in a FMM. Particles that are responsible for a force, velocity, or potential are called *causes*. Particles that are influenced by these "causes" are called *effects*. Ordinarily, $N$ body problems involve the causes alone. In the case of the panel method applied to certain problems like vortex based flow solvers the number of causes is less (small number of panels) and the number of effects is very large. This must be carefully handled.

A simple method is proposed to handle different number of cause and effect elements efficiently. The first cell, which contains all the particles and panels, is defined to be at the level 0. If a cell is at level $l$ cells of the next level $l + 1$ are obtained by splitting it into four daughter cells. Consider a cell $b$ at a level $l$. The colleagues of cell $b$ are defined as all the cells at the same level $l$ that share either a side or corner with the cell $b$. This is illustrated in Figure 4.

In order to handle both causes and effects efficiently, the decision to split the cell is handled as per the following pseudocode. Here `n_cause`, `n_effect` are the number of cause or effect particles in the particular cell and `max_cause`, `max_effect` are the maximum allowed number of the particle type per cell. The lines with a `#` are comments.

```
if (cell.n_cause > max_cause) and (cell.n_effect > max_effect):
        Split the cell into daughter cells.

else if (cell.n_cause <= max_cause) and (cell.n_effect <= max_effect):
        # No need to split the cell: it has too few particles
        Store the cell as a leaf/childless cell.

else if (cell.n_cause > max_cause):
        for each colleague of cell:
                if (colleague.n_effect > max_effect):
                        Split the cell into daughter cells
                        break
        if cell is not split:
                Store the cell as a leaf/childless cell.

else:
        for each collegue of cell:
                if (collegue.n_cause > max_cause):
                        Split the cell into daughter cells.
                        break.
        if cell is not split:
                Store the cell as a leaf/childless cell.
```

The reasoning behind the algorithm is as follows.

1. If there are a large number of causes and effects in a cell, it would be inefficient to perform a direct computation. Therefore, the cell needs to be split into daughter cells.
2. If there are very small number causes and effects in the cell, then it is faster to perform the computation on the cell directly. Therefore, the cell should be left alone and stored as a leaf/childless cell.
3. If there are a large number of cause elements but small number of effects in the cell, $b$, then the decision to split the cell depends on the nature of its colleagues. If any of the colleagues, $c$, of cell $b$ have a large number of effects, then $b$ should be split. This will reduce the number of direct computations that have to be performed between the cell $b$ and $c$. If there are no such colleague cells, then the cell $b$ can be stored as a leaf/childless cell.
4. Similarly, if there are a large number of effect elements but small number of causes in the cell $b$, then it should be split if any of its colleagues $c$, have a large number of causes. If there are no such colleague cells, then the cell $b$ can be stored as a leaf/childless cell.

By splitting the cells based on the above it is possible to efficiently handle all the different cases where there are varying numbers of cause and effect particles. Apart from the cell splitting, the rest of the algorithm closely resembles the original AFMM [4] due to Carrier, Greengard, and Rokhlin. The only difference is that small changes are made in the algorithm to accommodate the passive "effect" particles.

**6. Results.** The theoretical and algorithmic details have been presented above and it remains to be seen if the resulting method does indeed improve the computational efficiency. The case of flow past a circular cylinder, centered at the origin

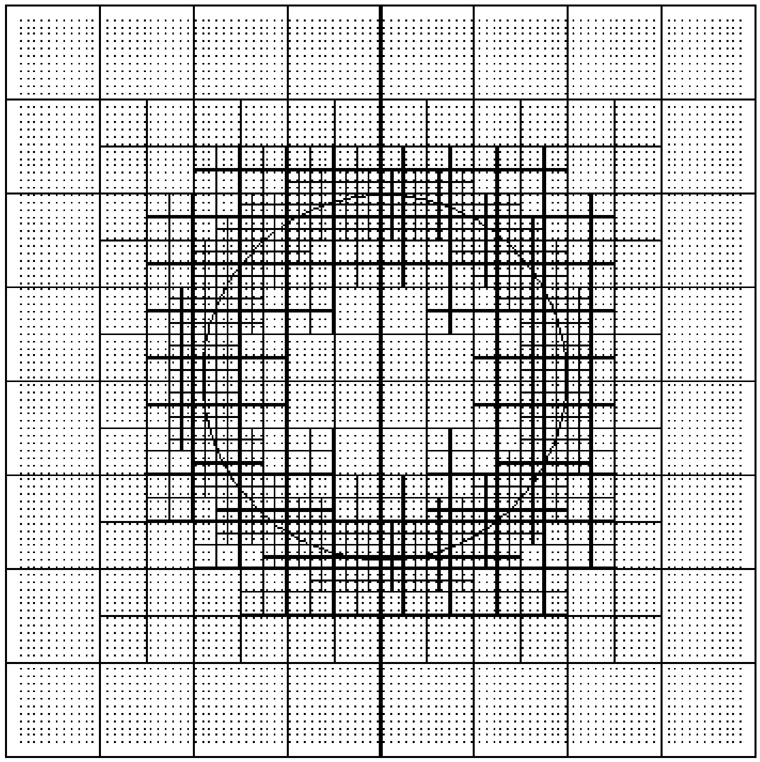

FIG. 5. *Hierarchical mesh for the flow past a circular cylinder with* 400 *panels and* 10000 *passive particles for the case* 1.

TABLE 6.1
*Comparison of times for cubic, flat, and fast panel methods.*

| Time taken (seconds) | Cubic panel | Flat panel | Fast panel method |
|---|---|---|---|
| Case 1 | 43.88 | 11.18 | 0.42 |
| Case 2 (wake region) | 43.53 | 10.95 | 0.12 |

with a radius of 1 unit, with 400 panels is considered. A set of 10000 uniformly spaced particles in a square region is considered. The coordinates of two diagonally opposite corners of the region are $(-2.0, -2.0)$ and $(2.0, 2.0)$. This is called Case 1. The velocity at each of these 10000 points is computed using the cubic panel method, the linear panel method, and finally the presently developed fast summation technique. The velocity of the panels on themselves is not computed. In the above computation the accuracy for the FMM is chosen as $\epsilon = 10^{-6}$ and the `max_cause` and `max_effect` are chosen as 7 panels and particles, respectively. The corresponding hierarchical mesh is shown in Figure 5. As is evident from Table 6.1 the fast panel method increases the speed by a significant amount. There is a 100 fold increase as compared to the cubic method and a 26 fold increase as compared to the traditional flat panel method. It is significant to note that if the domain of interest (the region where the particles are distributed) is further away from the body, a much greater increase is seen. To demonstrate this a square region between

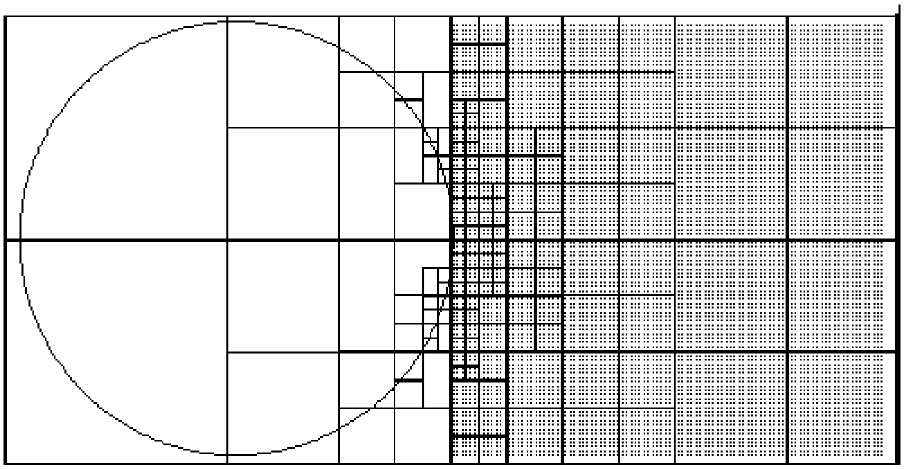

FIG. 6. *Hierarchical mesh for the flow past a circular cylinder with* 400 *panels and* 10000 *passive particles for case* 2.

$(1.0, -1.0)$ and $(3.0, 1.0)$ with 10000 uniformly spaced points (case 2), is considered. This region is in the wake of the cylinder. For case 2 the results are again shown in Table 6.1. The corresponding mesh is shown in Figure 6. It is clear from Figure 6 that the refinement of the mesh becomes finer as the particles get closer to the body and coarser as the particles move away from the body. From the results it can be observed that as one moves away from the body the computation becomes more efficient (in this case by a factor of 3.5). There is certain to be an upper limit of efficiency gained for a given computation but as is evident from the above results, more than two orders of magnitude in speed increase are achievable. The parameters `max_cause` and `max_effect` have to be chosen carefully. The minimum cell size must also be chosen carefully and in this case the minimum length of a cell is two panel lengths.

The fast panel technique has very significant advantages from the perspective of computational efficiency. In the following, the accuracy of the method is compared to that of the cubic panel technique. The flow past a cylinder is used as a benchmark. (2.5) is used to compute the error on a ring away from the cylinder. The ring is made up of around 1000 particles. 200 panels are used for the cylinder. Figure 7 plots the variation of the error, $E$, versus the distance of the ring from the surface the cylinder, $r$. The computation was performed using a maximum of 7 panels or particles per cell. The fast panel method is more accurate than the flat panel technique because (a) the cubic panel method is used to compute the strengths on the panel and (b) in the vicinity of the panels the cubic panel method is used to find the velocity. Figure 8 plots the error $E$ for different values of the `max_cause` parameter, which determines the maximum number of panels allowed per cell. The larger this parameter the larger the number of panels that are used to perform the direct computation. From the figure it is evident that there is a lower bound for the error in the case of the fast panel technique. Though increasing `max_cause` reduces the error in the vicinity of the cylinder there is no significant gain away from the body. On the other hand, increasing `max_cause` increases the computational time. Therefore the parameter must be chosen carefully depending on the accuracy required.

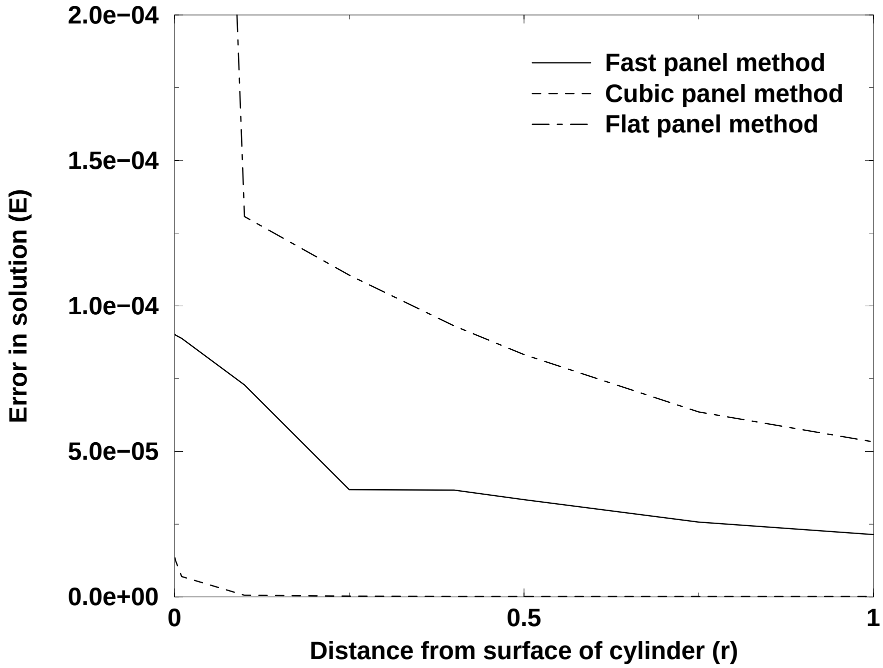


FIG. 7. *Comparison of the error $E$ versus distance from the surface of a circular body obtained by using the fast panel method, the flat panel method, and the cubic panel method. $E$ is computed using* (2.5).

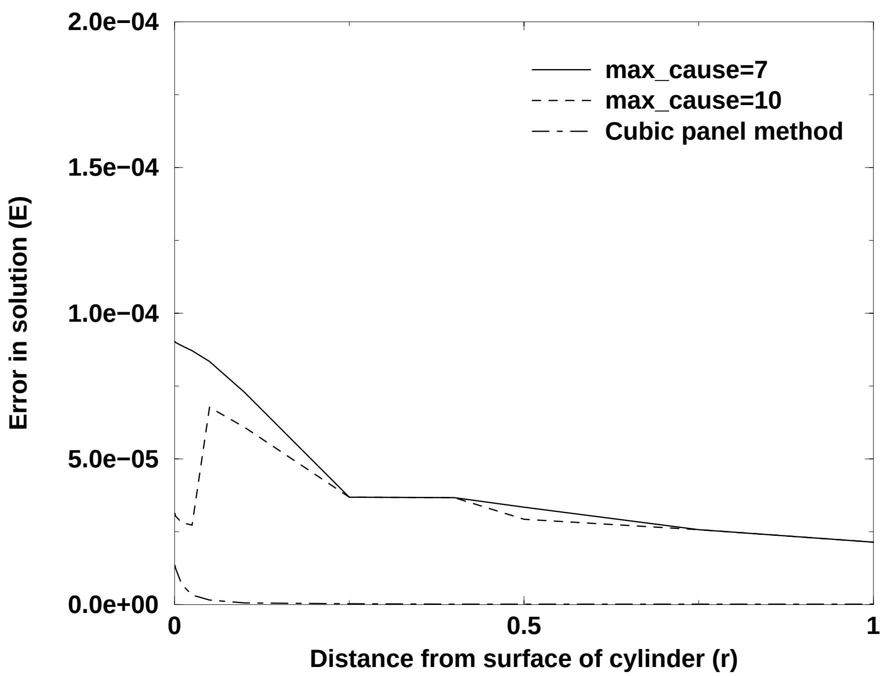


FIG. 8. *Comparison of the error $E$ versus distance from the surface of a circular body obtained by using different* max_cause *parameters as compared to using only cubic panels.*

**7. Conclusions.** The AFMM applied to the linear panels greatly improves the efficiency of the velocity computation. The mesh generation criterion that has been developed here is simple and produces the expected efficiency while retaining generality. The expressions derived for the fast panel technique can be readily used in an existing fast multipole program. The methodology has been shown to eliminate the edge effect while improving computational efficiency considerably. From the results it is clear that the error of the fast method is larger than the cubic panel method. This is due to the fact that the flat panel method has been used for the AFMM. The error can be improved by using the cubic panel technique for the AFMM. This is currently being pursued.

## REFERENCES

[1] C. R. Anderson, *A method of local corrections for computing the velocity field due to a distribution of vortex blobs*, J. Comput. Phys., 62 (1986), pp. 11–123.

[2] C. R. Anderson, *An implementation of the fast multipole method without multipoles*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 923–947.

[3] J. Barnes and P. Hut, *A hierarchical o(nlogn) force-calculation algorithm*, Nature, 324 (1986), pp. 446–449.

[4] J. Carrier, L. Greengard, and V. Rokhlin, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 669–686.

[5] N. R. Clarke and O. R. Tutty, *Construction and validation of a discrete vortex methods for the two-dimensional incompressible Navier Stokes equations*, Comput. & Fluids, 23 (1994), pp. 751–783.

[6] C. I. Draghicescu and M. Draghicescu, *A fast algorithm for vortex blob interactions*, J. Comput. Phys., 116 (1995), pp. 69–78.

[7] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.

[8] J. L. Hess and A. M. O. Smith, *Calculation of nonlifting potential flow about arbitrary three dimensional bodies*, J. Ship Research, 8 (1964), pp. 22–44.

[9] J. Katz and A. Plotkin, *Low-Speed Aerodynamics: From Wing Theory to Panel Methods*, McGraw–Hill, New York, 1992.

[10] R. I. Lewis, *Vortex Element Methods for Fluid Dynamical Analysis of Engineering Systems*, Cambridge University Press, Cambridge, UK, 1991.

[11] H. Lin, M. Vezza, and R. A. M. Galbraith, *Discrete vortex method for simulating unsteady flow around pitching aerofoils*, AIAA J., 35 (1997), pp. 494–499.

[12] S. R. Lustig, S. Rastogi, and N. Wagner, *Telescoping fast multipole methods using Chebyshev economisation*, J. Comput. Phys., 122 (1995), pp. 317–322.

[13] J. Makino, *Yet another fast multipole method without multipoles — pseudoparticle multipole method*, J. Comput. Phys., 151 (1999), pp. 910–920.

[14] E. Martensen, *The calculation of the pressure distribution on a cascade of thick airfoils by means of Fredholm integral equation of the second kind*, NASA TTF, 702 (1971).

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in 'C': The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 1992.

[16] S. C. Rajan, *Unification of ideal fluid and rigid body kinematics using vorticity*, Mech. Res. Comm., 21 (1994), pp. 443–448.

[17] P. Ramachandran, S. C. Rajan, and M. Ramakrishna, *An accurate two-dimensional panel method*, in Seminar on Advances in Aerospace Technologies, SAAT-2000, Aeronautical Society of India, New Delhi, 2000.

[18] K. Takeda, O. R. Tutty, and D. A. Nicole, *Parallel discrete vortex methods on commodity supercomputers; an investigation into bluff body far wake behaviour*, ESAIM Proc., 7 (1999), pp. 418–428.

[19] I. Taylor and M. Vezza, *Calculation of the flow around a square section cylinder undergoing forced transverse oscillations using a discrete vortex method*, J. Wind Engineering and Industrial Aerodynamics, 82 (1999), pp. 271–291.

[20] I. Taylor and M. Vezza, *Prediction of unsteady flow around square and rectangular cylinders using a discrete vortex method*, J. Wind Engineering and Industrial Aerodynamics, 82 (1999), pp. 247–269.

[21] L. van Dommelen and E. A. Rundensteiner, *Fast, adaptive summation of point forces in the two-dimensional Poisson equation*, J. Comput. Phys., 83 (1989), pp. 126–147.