A method is presented for calculating potential flows in infinite channels. Given a collection of N sources in the channel and a zero normal flow boundary condition, the method requires an amount of work proportional to N to evaluate the induced velocity field at each source position. Previous schemes have been based either on conformal mapping, which experiences numerical difficulties with the domain boundary, or direct evaluation of the Green's function. Both require $O(N^2)$ work.

Potential Flow in Channels

L. Greengard

Research Report YALEU/DCS/RR-635
July 1988

The author was supported in part by the Office of Naval Research under Grant N00014-86-K-0310 and in part by a NSF Mathematical Sciences Postdoctoral Fellowship. Approved for public release: distribution is unlimited.

Keywords: *Fluid Dynamics, Potential Theory, N-body Problem, Fast Multipole Method*

1 Introduction

The evaluation of potential fields in infinite channels arises as a numerical problem in several areas, most notably electrostatics and fluid dynamics. The governing equation is the Poisson equation,

$$\Delta\Psi = -\xi \tag{1}$$

subject to an appropriate boundary condition. In this paper, we will restrict our attention to two-dimensional models and will consistently use the terminology of fluid dynamics. In viscous incompressible flow, the left-hand side is the stream function, the right-hand side is the vorticity, and the condition imposed on the boundary is that of zero normal flow

$$\mathbf{u} \cdot \mathbf{n} = 0 \ , \tag{2}$$

where the velocity field $\mathbf{u}$ is given by

$$\mathbf{u} = \left(\frac{\partial\Psi}{\partial y}, -\frac{\partial\Psi}{\partial x} \right) \ . \tag{3}$$

In terms of the stream function, this is equivalent to specifying homogeneous Dirichlet boundary conditions

$$\Psi = 0 \ . \tag{4}$$

We will concentrate on particle models, where the vorticity field is discretized, not by a mesh, but by N point vortices,

$$\xi = \sum_{i=1}^{N} \xi_i \cdot \delta(x - x_i, y - y_i) \ . \tag{5}$$

Here, δ is the Dirac δ-function and ξ_i is the strength of the ith point vortex located at (x_i, y_i). In vortex methods, what we would like to compute is the stream function and/or velocity field at each particle position. In the absence of boundary effects, the desired results can be obtained from the free-space Green's function for the Poisson equation $(-\frac{1}{2\pi}\ln r)$ as

$$\Psi(x_i, y_i) = \sum_{j \neq i} -\frac{\xi_j}{\pi} \cdot \ln((x_i - x_j)^2 + (y_i - y_j)^2) \quad \text{for} \quad i = 1, ..., N \ , \tag{6}$$

$$\mathbf{u}(x_i, y_i) = \sum_{j \neq i} -\frac{\xi_j}{2\pi} \cdot \frac{(y_i - y_j, x_j - x_i)}{(x_i - x_j)^2 + (y_i - y_j)^2} \quad \text{for} \quad i = 1, ..., N \ . \tag{7}$$

Note that, using direct summation, the calculations (6) and (7) require an amount of work proportional to N^2. To overcome this obstacle, a variety of fast "N-body" methods have been proposed in the last few years, which reduce the computational complexity to $O(N \log N)$ or $O(N)$. These include particle-in-cell methods [1,13], astrophysical tree codes [2,3], series expansion methods [15,16], and the fast multipole method [5,9,10].

Remark 1.1: It is clear from (6) and (7) that the stream function and velocity field are unbounded in the neighborhood of a point vortex. In [7], Chorin introduced the idea of replacing the point vortices by "vortex blobs" whose induced field is held constant within a small neighborhood of the source. More recent work by Hald [11], Beale and Majda [4] and others has shown

that higher order accuracy can be obtained by using different approximations for the local field. Outside a finite-size core, however, the velocity field due to a vortex blob in all of these methods is simply that of a point vortex. Since we are interested in reducing the computational cost of vortex methods, which is generally dominated by *far field* interactions, we will ignore the precise nature of the local interactions and will continue to use the point vortex model.

For a straight channel, the fluid velocity cannot be obtained as in (6) and (7). The main difficulty is that the zero normal flow condition can only be satisfied by an infinite image system (Section 2), making direct summation over a collection of point sources impossible. The most commonly used technique for overcoming this problem in constrained flows is that of conformal mapping. By converting the calculation to one in the upper half plane, the boundary condition can be imposed with one image per particle, and the potential flow computed as in (6) and (7) with only double the number of point vortices (Fig. 1). An attractive feature of this approach is that the fast N-body algorithms for free-space calculations may directly be applied.

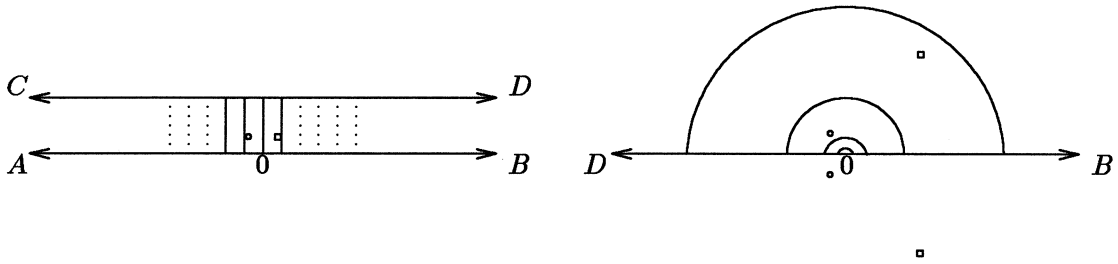

Figure 1

Conformal mapping of the channel to the upper half plane. The left-hand limit points A and C are mapped to the origin and the four solid vertical line segments in the channel are mapped to the four semicircles in the upper half plane. Two representative particles are marked by the small circle and square. The zero normal flow boundary condition is easy to apply with the method of images (each source is simply reflected across the x-axis and given opposite strength). Unfortunately, there is much stretching and contraction of the physical domain which can cause practical difficulties.

There are two objections to this mathematically reasonable procedure. In a channel with zero normal flow boundary conditions, the velocity field induced by a point source decays exponentially along the length of the channel. But the free-space Green's function used in the upper half plane decays as $\frac{1}{r}$, losing much of the physical behavior of the solution. In fact, the physical behavior is expressed by the mapping itself, which for the strip $\{0 \leq y \leq \pi\}$ is simply e^z. The second objection is that a discretization of the boundary is often required (e.g. for vorticity generation). Conformal mapping, however, is well known to experience numerical difficulty when the derivative of the map has a great dynamic range [14,17]. This is clearly observed in Fig. 1, where the images of equispaced points along the top and bottom of the channel are points whose separation is growing (or contracting) exponentially. It would, on both counts, be much preferable to remain in the channel itself. To do this we will first need to replace the infinite image system by an analytic expression for the Green's function. This can be obtained through elliptic function theory. In [6], Choi and Humphrey derive expressions for both the infinite channel and a closed rectangular

domain. With this expression, the velocity field can be obtained in a manner analogous to the N-body calculation of equation (7). Direct summation, of course, will require $O(N^2)$ work.

In this paper, we propose a new algorithm for two-dimensional potential flow in infinite channels. It is based on the analytically derived Green's function, and requires an amount of work proportional to N to evaluate all pairwise interactions.

2 Green's function for an infinite channel

We begin by developing an explicit expression for the velocity field induced by a point vortex in an infinite channel. The domain is defined to be the strip $\{0 \leq y \leq H\}$. We refer to the direction x increasing as *downstream* and to the direction x decreasing as *upstream.* We will use complex notation, equating the points (x_i, y_i) with the complex numbers z_i. If we define $\tilde{\mathbf{u}}$ by

$$\tilde{\mathbf{u}}(z_i) = \sum_{j \neq i} \frac{\xi_j}{2\pi} \cdot \frac{1}{z_i - z_j} \,, \tag{8}$$

then

$$\mathbf{u}(x_i, y_i) = (-\mathrm{Im}(\tilde{\mathbf{u}}(z_i)), \mathrm{Re}(\tilde{\mathbf{u}}(z_i))) \tag{9}$$

is the velocity field induced by a collection of point sources with strength ξ_j located at the points $z_j = (x_j, y_j)$. In the remainder of this paper, we consider the calculation of $\tilde{\mathbf{u}}$ rather than $\mathbf{u}$ and will abuse notation by referring to $\tilde{\mathbf{u}}$ as the velocity field.

Let us now suppose that a source of unit strength is located inside the channel at z_0 and that z is a second point inside the channel with $z \neq z_0$. In order to satisfy the zero normal flow condition along the top and bottom of the channel, we introduce the infinite image system shown in Fig. 2.

Let us first add up the contributions from the images with positive strength, located at $z_0 + 2jHi$ $(j = -\infty, ..., \infty)$. The velocity field $\tilde{\mathbf{u}}_1(z)$ induced by these images is given by the expression

$$\tilde{\mathbf{u}}_1(z) = \sum_{j=-\infty}^{\infty} \frac{1}{z - z_0 + 2jHi} \tag{10}$$

$$= \frac{1}{z - z_0} + \sum_{j=1}^{\infty} \frac{1}{z - z_0 + 2jHi} + \frac{1}{z - z_0 - 2jHi} \tag{11}$$

$$= \frac{1}{z - z_0} + \sum_{j=1}^{\infty} \frac{2(z - z_0)}{(z - z_0)^2 + 4H^2 j^2} \,. \tag{12}$$

But from ([8], p. 36) we have

$$\coth(\pi z) = \frac{1}{\pi z} + \frac{2z}{\pi} \sum_{k=1}^{\infty} \frac{1}{z^2 + k^2} \,, \tag{13}$$

so that

$$\tilde{\mathbf{u}}_1(z) = \sigma \cdot \coth(\sigma(z - z_0)) \,, \tag{14}$$

where

$$\sigma = \frac{\pi}{2H} \,. \tag{15}$$

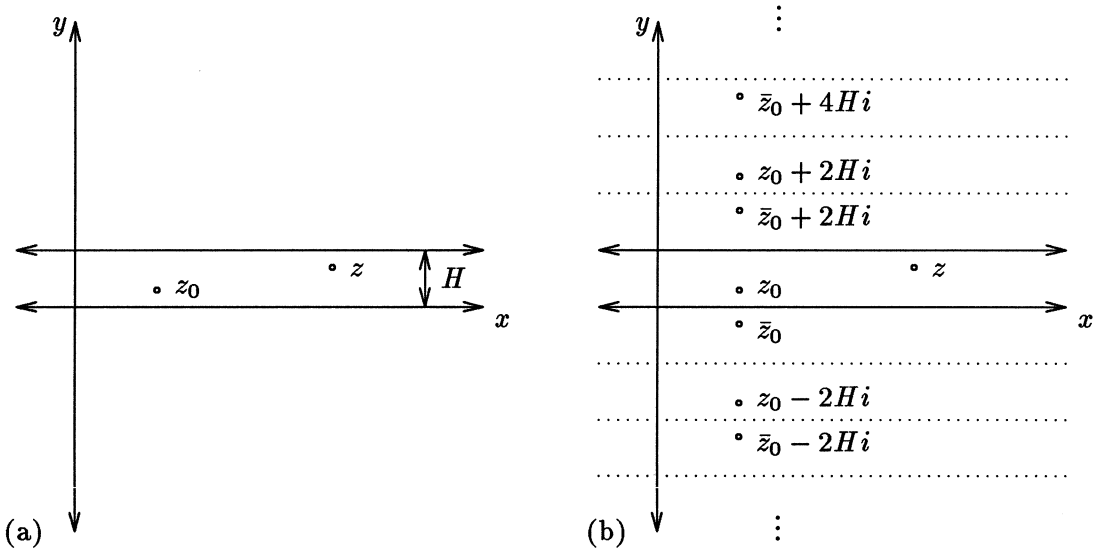

Figure 2

Enforcing boundary conditions by the method of images. Successive reflection across the top and bottom boundaries creates the image system shown. The images at positions $\bar{z}_0 + 2jHi, j = -\infty, ..., \infty$ have strengths of opposite sign from those at positions $z_0 + 2jHi$.

For the images with negative strength, located at $\bar{z}_0 + 2jHi$ ($j = -\infty, ..., \infty$), we obtain an induced velocity field $\tilde{\mathbf{u}}_2$, given by

$$\tilde{\mathbf{u}}_2(z) = -\sigma \cdot \coth(\sigma(z - \bar{z}_0)) \; . \tag{16}$$

The net velocity field is, therefore,

$$\tilde{\mathbf{u}}(z) = \sigma \cdot (\coth(\sigma(z - z_0)) \; - \; \coth(\sigma(z - \bar{z}_0))) \; . \tag{17}$$

A simple integration yields the stream function Ψ induced by a point vortex,

$$\Psi(z) = \mathrm{Re}\left(\log\left(\frac{\sinh(\sigma(z - z_0))}{\sinh(\sigma(z - \bar{z}_0))}\right)\right) \; . \tag{18}$$

A different derivation of Ψ is given by Choi and Humphrey in [6]. As mentioned previously, with this analytic expression for the pairwise interaction, the evaluation of the velocity field at each of the N source positions can be carried out in $O(N^2)$ operations. In order to develop a fast algorithm tailored to the channel itself, we need to examine the properties of the Green's function in more detail.

2.1 Upstream and Downstream Expansions

Let us suppose that z is downstream of z_0 ($\mathrm{Re}(z - z_0) > 0$). Then

$$\coth(\sigma(z - z_0)) \;\; = \;\; \frac{e^{\sigma\cdot(z-z_0)} + e^{-\sigma\cdot(z-z_0)}}{e^{\sigma\cdot(z-z_0)} - e^{-\sigma\cdot(z-z_0)}} \tag{19}$$

$$= -1 + \frac{2}{1 - e^{-2\sigma\cdot(z-z_0)}} \tag{20}$$

$$= -1 + 2\cdot\sum_{k=0}^{\infty} e^{2\sigma z_0 k}\cdot e^{-2\sigma z k} \tag{21}$$

$$= 1 + 2\cdot\sum_{k=1}^{\infty} e^{2\sigma z_0 k}\cdot e^{-2\sigma z k}\ . \tag{22}$$

Note that (21) can be obtained from (20) only if $e^{-2\sigma\cdot(z-z_0)} < 1$ which is ensured by the condition that z be downstream of the source. From (17), then, the velocity field downstream of a unit source at z_0 has the expansion (about the origin)

$$\tilde{\mathbf{u}}(z) = \sigma\cdot(\coth(\sigma(z-z_0)) - \coth(\sigma(z-\bar{z}_0))) \tag{23}$$

$$= 2\sigma\cdot\sum_{k=1}^{\infty}(e^{2\sigma z_0 k} - e^{2\sigma\bar{z}_0 k})\cdot e^{-2\sigma z k}\ . \tag{24}$$

From this, it is immediately obvious that the decay in the field is exponential along the length of the channel. The main reason for developing an expansion of this form, however, is that it allows us to effectively use the superposition principle. By this we mean the following:

Theorem 2.1 *Suppose that m sources with strengths $\{q_j,\ j = 1,...,m\}$ are located at points $\{z_j,\ j = 1,...,m\}$, with $\mathrm{Re}(z_j) < r$. Then for any point z further downstream $(\mathrm{Re}(z) > r)$, the velocity $\tilde{\mathbf{u}}(z)$ induced by the sources is given by*

$$\tilde{\mathbf{u}}(z) = \sum_{k=1}^{\infty} a_k\cdot e^{-2\sigma z k} \tag{25}$$

where

$$a_k = 2\sigma\cdot\sum_{j=1}^{m} q_j\cdot(e^{2\sigma z_j k} - e^{2\sigma\bar{z}_j k})\ . \tag{26}$$

The error in truncating the expansion (25) after p terms has the bound

$$\left|\tilde{\mathbf{u}}(z) - \sum_{k=1}^{p} a_k\cdot e^{-2\sigma z k}\right| \le \frac{A\cdot x^{p+1}}{1-x} \tag{27}$$

where

$$A = 4\sigma\cdot\sum_{j=1}^{m}|q_j| \quad \textit{and} \quad x = e^{-2\sigma\cdot(\mathrm{Re}(z)-r)}\ . \tag{28}$$

Proof: The coefficients a_k are obtained directly from equation (24). The error bound is a consequence of the triangle inequality and the fact that (24) expands the field due to a single source as the sum of two geometric series. □

The upstream direction is treated in an analogous fashion. If $\mathrm{Re}(z - z_0) < 0$, then the velocity due to a source at z_0 can be expressed as

$$\tilde{\mathbf{u}}(z) = 2\sigma\cdot\sum_{k=1}^{\infty}(e^{-2\sigma z_0 k} - e^{-2\sigma\bar{z}_0 k})\cdot e^{2\sigma z k}\ . \tag{29}$$

Theorem 2.2 *Suppose that m sources with strengths $\{q_j,\ j = 1,...,m\}$ are located at points $\{z_j,\ j = 1,...,m\}$, with $\mathrm{Re}(z_j) > r$. Then for any point z further upstream $(\mathrm{Re}(z) < r)$, the velocity $\tilde{\mathbf{u}}(z)$ induced by the sources is given by*

$$\tilde{\mathbf{u}}(z) = \sum_{k=1}^{\infty} b_k \cdot e^{2\sigma z k} \tag{30}$$

where

$$b_k = 2\sigma \cdot \sum_{j=1}^{m} q_j \cdot (e^{-2\sigma z_j k} - e^{-2\sigma \bar{z}_j k}) \ . \tag{31}$$

The error in truncating the expansion (30) after p terms has the bound

$$\left| \tilde{\mathbf{u}}(z) - \sum_{k=1}^{p} b_k \cdot e^{2\sigma z k} \right| \leq \frac{A \cdot x^{p+1}}{1-x} \tag{32}$$

where

$$A = 4\sigma \cdot \sum_{j=1}^{m} |q_j| \quad \textit{and} \quad x = e^{2\sigma \cdot (\mathrm{Re}(z) - r)} \ . \tag{33}$$

Definition 2.1 *The expansions given by (30) and (25) will be referred to as* upstream *and* downstream *expansions, respectively. For a given collection of sources, the pair will be referred to as* S-expansions.

The representation of the velocity field by means of these expansions may be viewed as an analog of the multipole decomposition of the field due to a collection of sources in free space. It is important to keep in mind, however, that both their rate of decay and region of convergence are quite different.

Before examining the properties of S-expansions any further, we demonstrate their usefulness in computing far field interactions with a simple example. For this, suppose that U_1 and U_2 are two sets, each containing N point vortices, located inside a channel of width H, and separated by a distance d (Fig. 3). To compute the velocity at each position in U_1 due to the sources in U_2 (or the

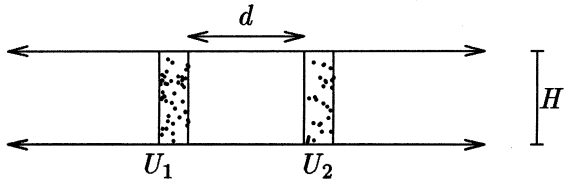

Figure 3

Two clusters of point vortices located inside a channel of width H. The distance between the two clusters is denoted by d.

velocity at each position in U_2 due to the sources in U_1) by means of the Green's function would require $O(N^2)$ operations. Let us instead form a downstream expansion due to the sources in U_1

and an upstream expansion due to the sources in U_2. From (27) and (32), it is easy to determine *a priori* how many terms are needed to achieve a relative precision of ϵ. We simply require that

$$x^{p+1} \approx \epsilon \qquad \text{or} \qquad p \approx \frac{H \cdot \ln \frac{1}{\epsilon}}{\pi \cdot d} , \tag{34}$$

which is independent of N. The cost of formation of the two expansions is clearly proportional to Np. Evaluating the two expansions at all points in the relevant cluster also requires an amount of work proportional to Np, so that the total computation scales linearly with N, assuming that the relative precision ϵ and separation distance d are fixed.

3 The Shifting Lemma for S-expansions

The fast algorithm to be described depends not only on the formation and evaluation of S-expansions, but on their analytic manipulation. The following obvious lemma describes the necessary tools.

Lemma 3.1 *Suppose that*

$$\tilde{\mathbf{u}}(z_u) = \sum_{k=1}^{\infty} b_k \cdot e^{2\sigma z_u k} \tag{35}$$

and

$$\tilde{\mathbf{u}}(z_d) = \sum_{k=1}^{\infty} a_k \cdot e^{-2\sigma z_d k} \tag{36}$$

are the up and downstream expansions about the origin due to m sources with strengths $\{q_j,\ j = 1, ..., m\}$ which are located at points $\{z_j,\ j = 1, ..., m\}$, with $-r < \mathrm{Re}(z_j) < r$ for some $r > 0$. Then

$$\tilde{\mathbf{u}}(z_u) = \sum_{k=1}^{\infty} \beta_k \cdot e^{2\sigma (z_u - z_0) k} \tag{37}$$

and

$$\tilde{\mathbf{u}}(z_d) = \sum_{k=1}^{\infty} \alpha_k \cdot e^{-2\sigma (z_d - z_0) k} \tag{38}$$

are the corresponding up and downstream expansions about z_0, where

$$\beta_k = b_k \cdot e^{2\sigma z_0 k} \tag{39}$$

and

$$\alpha_k = a_k \cdot e^{-2\sigma z_0 k} \ . \tag{40}$$

Furthermore, the error bounds for the shifted S-expansions are exactly the same as those for the original S-expansions.

Note that the behavior of shifted S-expansions contrasts sharply with that of multipole expansions in free-space (see [9,10]). In the latter situation, the validity and accuracy of an expansion depends not only on the source positions but on the location of the expansion center. Note also

from (26) and (31) that the coefficients of S-expansions about the origin are pure imaginary. If the centers of the shifted expansions are chosen to lie along the x-axis, then the coefficients in (39) and (40) are also pure imaginary, yielding a savings in both computational cost and storage.

Remark 3.1: To this point, we have been viewing S-expansions as representations of the *far field* due to a distribution of sources. It is possible, however, to view them in a different light. The expansions (35) and (36) of the preceding lemma are valid outside the strip $-r < \mathrm{Re}(z_j) < r$. By choosing a point z_0 upstream of the strip boundary ($\mathrm{Re}(z_0) < -r$), the shifted expansion (37) yields a representation of the induced field in a neighborhood of z_0. The same obviously holds for shifting a downstream expansion in the downstream direction (38). These are *local* representations of the field, the analogs of Taylor series in free-space, just as the far field S-expansions are the analogs of multipole expansions.

4 The Channel Decomposition Algorithm

In this section, we describe the first part of the fast algorithm. The basic idea is to subdivide the channel into vertical strips and to use S-expansions to compute far field interactions.

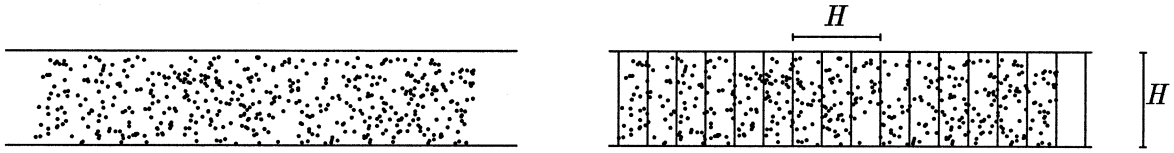

Figure 4

Decomposition of the channel into "elementary strips." The original distribution of particles is shown in (a). In (b), a finite domain containing all particles has been subdivided into rectangular regions, each of which has an aspect ratio of one third. S-expansions can be used to compute the interactions between particles contained in non-neighboring strips.

The "elementary" strips into which the channel is refined have an aspect ratio of one third. The reason for not subdividing too much further is clear from equation (34). As d approaches 0, the number of terms required to achieve a fixed precision grows arbitrarily large. If we stop using expansions when $d = H/3$, however, then the number of terms required is given by

$$p \approx \frac{3 \cdot \ln \frac{1}{\epsilon}}{\pi} < \ln \frac{1}{\epsilon} \,. \tag{41}$$

We, somewhat arbitrarily, choose to stop subdividing the channel at this point. We will, of course, need to compute the interactions within an elementary strip and between nearest neighbor strips. This part of the calculation will be described in section 5. It relies on some additional analysis and the Fast Multipole Method for free-space problems.

The remainder of this section is devoted to a description of the channel decomposition algorithm. The main strategy used is that of clustering particles at a variety of spatial length scales and computing distant interactions by means of S-expansions. We begin by determining the locations of the extreme upstream and downstream particles. The corresponding section of the channel is

considered to be the computational domain, and a sufficient number of elementary strips are created to cover the region (Fig. 4).

We proceed by introducing a binary tree structure which groups particles at coarser and coarser levels (Figure 5). Level 0 corresponds to the finest discretization of space (the elementary strips), while level $l+1$ is obtained from level l by the merger of two strips. The resulting strip at the higher level is referred to as the parent, while the two strips being merged are referred to as the children. Two strips at the same refinement level are said to be nearest neighbors if they share a boundary, otherwise they are said to be well-separated. By construction, then, the minimum distance between well-separated strips is $H/3$, and in order to achieve a precision of ϵ in computing interactions via S-expansions we need only choose the number of terms to be $p = \lceil \ln(1/\epsilon) \rceil$. At coarser levels, the number of terms can obviously be decreased.

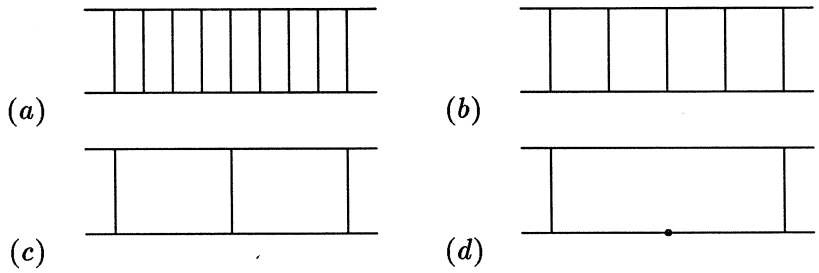

Figure 5

In (a), eight elementary strips are shown which cover the computational domain. This level of spatial refinement is referred to as level 0. In (b), (c) and (d), pairs of strips are successively merged to form coarser and coarser subdivisions of the channel. The "center" of a strip is defined to be the midpoint of the segment of the x-axis bounded by that strip, as indicated in (d).

Definition 4.1 *The* center *of a strip is defined to be the midpoint of the segment of the x-axis bounded by that strip* (Fig. 5 (d)).

Other notation used in the description of the algorithm includes

$F^u_{l,i}$ a p-term upstream expansion about the center of strip i at level l, describing the far field due to the particles contained inside the strip.

$F^d_{l,i}$ a p-term downstream expansion about the center of strip i at level l, describing the far field due to the particles contained inside the strip.

$L^u_{l,i}$ a p-term local S-expansion (see Remark 3.1) about the center of strip i at level l, describing the field due to all particles upstream of strip i's nearest neighbors.

$L^d_{l,i}$ a p-term local S-expansion (see Remark 3.1) about the center of strip i at level l, describing the field due to all particles downstream of strip i's nearest neighbors.

Interaction list for strip i at level l, it is the set of strips which are children of the nearest neighbors of i's parent and which are well-separated from strip i (Figure 6).

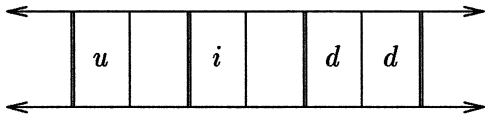

Figure 6

The interaction list for strip i at level l. Strips marked with a "u" are upstream members of the list, while those marked with a "d" are downstream members of the list. Note that thick lines correspond to mesh level l+1.

The channel decomposition algorithm is a two-pass procedure. In the first (upward) pass, we form the far field S-expansions $F^u_{l,i}$ and $F^d_{l,i}$ for all strips at all levels, beginning at the level of elementary strips. In the second (downward) pass, we form the local S-expansions $L^u_{l,i}$ and $L^d_{l,i}$ for all strips at all levels, beginning at the coarsest level.

To see how the latter part is accomplished, suppose that at level $l+1$, the local expansions L^u and L^d have been obtained for each strip i. Then, by using lemma 3.1 to shift these expansions to the centers of strip i's children, we obtain up and downstream expansions for each child strip at level l, describing the velocity field due to all particles up and downstream of strip i's nearest neighbors. For each strip j at level l, then, the interaction list is precisely that set of strips whose contribution to the potential must be added in order to create $L^u_{l,j}$ and $L^d_{l,j}$ (Fig. 6). For each upstream member of the list, we use Lemma 3.1 to shift the center of the corresponding far field expansion F^u to the center of strip j and add the result to the upstream expansion obtained from the parent. Similarly, for each downstream member of the list, we use Lemma 3.1 to shift the center of the corresponding far field expansion F^d to the center of strip j and add the result to the downstream expansion obtained from the parent. Note that at the coarsest level, L^d and L^u are equal to zero, since there are no well-separated strips to consider.

Finally, for each strip j at the finest level, we evaluate the local expansions $L^d_{0,j}$ and $L^u_{0,j}$ at the position of each particle contained in the strip.

Algorithm 1

Comment [Set number of terms to be used in expansions.]

Choose the precision ϵ to be achieved. Set the number of terms in all expansions to $p = \lceil \ln(1/\epsilon) \rceil$.

Upward Pass

Step 1.

Comment [Decompose the channel into elementary strips.]

Define elementary strip width to be $S_{wid} = H/3$.
Compute x_{min} = x-coordinate of extreme upstream particle position.
Compute x_{max} = x-coordinate of extreme downstream particle position.
Compute number of elementary strips $K = \lceil (x_{max} - x_{min})/S_{wid} \rceil$.
Compute height of binary tree $nlev = \lceil \log_2 K \rceil$.

Step 2.

Comment [Form far field S-expansions at finest level.]

do $i = 1, ..., K$
Form p-term up and downstream expansions $F^u_{0,i}$ and $F^d_{0,i}$
by using Theorems 2.1 and 2.2.
end do

Step 3.

Comment [Form far field S-expansions at all coarser refinement levels.]

do $l = 1, ..., nlev$
Form p-term up and downstream expansions $F^u_{l,i}$ and $F^d_{l,i}$
for each strip i at level l by using Lemma 3.1
to shift the center of each child strip's expansions to the current
strip center and adding them together.
end do

Downward Pass

Step 4.

Comment [Form local S-expansions at all refinement levels. Recall that L^u and L^d are zero at level $nlev$ since there are no well separated strips to consider.]

do $l = nlev - 1, ..., 0$
For each strip i at level l, initialize $L^u_{l,i}$ and $L^d_{l,i}$
by shifting the L^u and L^d expansions of strip i's parent to the
center of strip i. For each strip in i's interaction list, determine
whether it is up or downstream of strip i. If upstream, shift the
center of the corresponding F^u expansion to i's center and add to
$L^u_{l,i}$. If downstream, shift the center of the corresponding
F^d expansion to i's center and add to $L^d_{l,i}$ (Fig. 6).
end do

Step 5.

Comment [Local S-expansions are now available at the finest mesh level. They can be used to compute the velocity field due to all particles outside the nearest neighbor elementary strips.]

do $i = 1, ..., K$
For each particle located in elementary strip i, evaluate
$L^u_{0,i}$ and $L^d_{0,i}$. Add results together.
end do

A brief operation count of the channel decomposition algorithm follows.

Step Number	Operation Count	Explanation
Step 1	order N	examine each particle position to determine extreme up and downstream coordinates.
Step 2	order $2Np$	Each particle contributes to an upstream and a downstream expansion.
Step 3	order $K \cdot p$	The number of nodes in a binary tree is less than twice the number of leaves, so that the total number of nodes is of the order K. For each node, an amount of work of the order p is performed.
Step 4	order $5p \cdot K$	For each strip at each level, there are at most three entries in the interaction list. For each entry, the amount of work is proportional to p. In addition, two p-term expansions must be obtained from the parent.
Step 5	order $2Np$	Two p-term S-expansions are evaluated for each particle.

The estimate for the running time is therefore

$$N \cdot (4p + 1) \quad + \quad K \cdot 6p. \tag{42}$$

5 The Evaluation of Nearest Neighbor Interactions

The channel decomposition algorithm has left us with a sequence of uncoupled problems to consider. For each elementary strip, we must compute the internal interactions as well as the effects of the sources contained in that strip on the particles in the nearest neighbors.

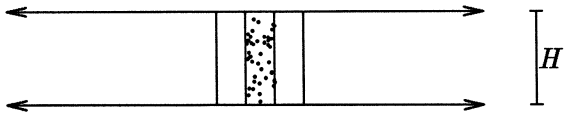

Figure 7

In the second part of the algorithm, interactions are computed within each elementary strip and between nearest neighbors. This is accomplished by marching along the channel, considering one strip at a time, and accounting for its influence on all relevant particles.

Because of their poor convergence rates in this regime, S-expansions are of limited use. We could proceed by direct evaluation of the remaining interactions through the use of the Green's function, but the asymptotic complexity of such an algorithm would be $O(N^2)$. Let us instead examine one of the subproblems in more detail.

We begin by reconsidering the method of images used to impose the zero normal flow condition in Figure 2. Successive reflection across the top and bottom of the channel yields a one-dimensional array of squares (Fig. 8). These are either copies of the channel section itself or of its reflection across the bottom boundary, offset by $2jHi$ for some integer j. Note that we are only acting on the sources contained within the central elementary strip, but that we will compute the velocity field at particle positions within all three elementary strips of which the square is composed. In this manner, all interactions will have been accounted for exactly once.

The problem, again, is how to account for the sources in all image squares. We present a solution based on multipole expansions.

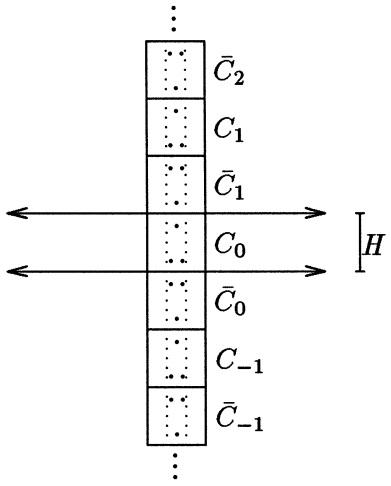

Figure 8

The channel section and its translated images are represented by boxes labelled C. The square obtained by reflection across the bottom boundary and its translates are labelled $\bar{C}$.

5.1 Multipole Expansions

We will require two results. For the first, suppose that m point vortices with strengths q_i and positions z_i are located within a disk of radius r centered at the origin. Then, for a point z with $|z| > r$, the velocity field $\mathbf{v}(z)$ induced by the sources is given by a multipole expansion of the form

$$\mathbf{v}(z) = \sum_{k=1}^{\infty} \frac{a_k}{z^k} \ , \tag{43}$$

where

$$a_k = \sum_{i=1}^{m} q_i \cdot z_i^{k-1} \ . \tag{44}$$

The error in truncating the sum after s terms is

$$\left| \mathbf{v}(z) - \sum_{k=1}^{s} \frac{a_k}{z^k} \right| \leq \left(\frac{A}{c-1} \right) \left(\frac{1}{c} \right)^s \ , \tag{45}$$

where

$$A = \sum_{i=1}^{m} |q_i| \qquad \text{and} \qquad c = \left|\frac{z}{r}\right| . \tag{46}$$

For a proof, see [9].

Note that in order to obtain a relative precision of ϵ (with respect to the total charge), the number of terms required in the series representation of $\mathbf{v}$ is approximately $-\log_c(\epsilon)$, independent of m, the number of source charges.

The second result we need is contained in the following lemma, which describes the conversion of a multipole expansion into a local (Taylor) expansion inside a circular region of analyticity.

Lemma 5.1 (Conversion of a Multipole Expansion into a Local Expansion) *Suppose that m sources of strengths $q_1, q_2, ..., q_m$ are located inside the circle D_1 with radius R and center at z_0, and that $|z_0| > (c+1)R$ with $c > 1$. Then the corresponding multipole expansion*

$$\mathbf{v}(z) = \sum_{k=1}^{\infty} \frac{a_k}{(z - z_0)^k} , \tag{47}$$

converges inside the circle D_2 of radius R centered about the origin. Inside D_2, the potential due to the charges is described by a power series:

$$\mathbf{v}(z) = \sum_{l=0}^{\infty} b_l \cdot z^l, \tag{48}$$

where

$$b_l = \sum_{k=1}^{\infty} \frac{a_k}{z_0^{l+k}} \binom{l+k-1}{k-1} (-1)^k . \tag{49}$$

Furthermore, for any $s \geq max\left(2, \frac{2c}{c-1}\right)$, an error bound for the truncated series is given by

$$\left| \mathbf{v}(z) - \sum_{l=0}^{s} b_l \cdot z^l \right| < \frac{A(4e(s+c)(c+1)+c^2)}{c(c-1)} \left(\frac{1}{c}\right)^{s+1} , \tag{50}$$

where A is defined in (46) and e is the base of natural logarithms.

Proof: See [9]. □

Definition 5.1 *Two squares with sides of length $2d$ are said to be* well-separated *if they are separated by a distance $2d$.*

Remark 5.1: Let A and B be well-separated squares with sides of length $2d$, and let D_A and D_B be the smallest disks containing the boxes A and B, respectively. Then the disks have radii $\sqrt{2} \cdot d$, and the distance from the center of one disk to the closest point in the other disk is at least $(4 - \sqrt{2}) \cdot d$. Letting $c = (4 - \sqrt{2})/\sqrt{2} \approx 1.828$, the error bound (50) applies with a truncation error using s-term expansions of the order c^{-s}.

Remark 5.2: In this section, the *center* of a square refers to its geometric center and not to its strip center (Definition 4.1).

5.2 Reduction to a Free Space Problem

We will use Lemma 5.1 to account for all image sources outside the nearest neighbor squares. The remaining calculation can then be carried using the free space Green's function (see page 1). We begin by choosing a coordinate system with the origin lying at the center of C_0. For each square C_j, the multipole expansion induced by the contained sources is of the form

$$\mathbf{v}(z) = \sum_{k=1}^{s} \frac{a_k}{(z - z_j)^k} \ . \tag{51}$$

where

$$z_j = 2jHi \tag{52}$$

is the square's center. Note that the coefficients a_k of such a multipole expansion are translation invariant; i.e. they are identical for all integer j. Moreover, for $j \neq 0$, C_j is well-separated from C_0, and the field induced inside the channel is accurately representable by an s-term local expansion, where $s = \lceil -\log_c(\epsilon) \rceil$ is the number of terms needed to achieve a relative precision ϵ (see Remark 5.1). This local representation is given by Lemma 5.1 as

$$\Phi_j(z) = \sum_{m=0}^{p} b_m \cdot z^m \tag{53}$$

with

$$b_m = \sum_{k=1}^{p} \frac{a_k}{z_j^{m+k}} \binom{m+k-1}{k-1} (-1)^k \ . \tag{54}$$

Let S be the set of non-zero integers. To account for the field due to all well-separated images C_j, we compute the coefficients of a local representation by adding together the shifted expansions of the form (54) for all z_j with $j \in S$ to obtain

$$\Phi(z) = \sum_{m=0}^{p} b_m^{total} \cdot z^m \tag{55}$$

where

$$b_m^{total} = \sum_{k=1}^{p} a_k \binom{m+k-1}{k-1} (-1)^k \left(\sum_{S} \frac{1}{z_j^{m+k}} \right) . \tag{56}$$

The summation over S for each inverse power of z_j can be precomputed and stored. For powers greater than one, the series is absolutely convergent. For $(m + k) = 1$, however, the series is not absolutely convergent, and the computed value depends on the order of addition. Choosing a reasonable value for the sum of the series requires consideration of the physical model. For this, suppose that the only particle in the simulation is a source of unit strength located at the origin. Then the image system corresponds to a uniform one-dimensional lattice, and by symmetry considerations, the induced velocity at any lattice point must be zero. But the net velocity of the particle at the origin corresponds to the summation over S of $1/z_j$, so that we set

$$\sum_{S} \frac{1}{z_j} = 0. \tag{57}$$

Lemma 5.2 *For $k > 1$,*

$$\sum_S \frac{1}{z_j^k} = \begin{cases} 0 & \text{if } k \text{ is odd} \\ \frac{(-1)^{\frac{k}{2}}}{2^{k-1}H^k}\zeta(k) & \text{if } k \text{ is even} \end{cases} \tag{58}$$

where $\zeta(z)$ is the Riemann zeta function.

Proof: For k odd, we have

$$\sum_S \frac{1}{z_j^k} = \sum_{j=1}^{\infty} \frac{1}{(2jHi)^k} + \sum_{j=1}^{\infty} \frac{1}{(-2jHi)^k} \tag{59}$$

$$= 0\ . \tag{60}$$

For k even, we have

$$\sum_S \frac{1}{z_j^k} = \sum_{j=1}^{\infty} \frac{1}{(2jHi)^k} + \sum_{j=1}^{\infty} \frac{1}{(-2jHi)^k} \tag{61}$$

$$= \frac{1}{2^{k-1}H^k i^k} \sum_{j=1}^{\infty} \frac{1}{j^k} \tag{62}$$

$$= \frac{(-1)^{\frac{k}{2}}}{2^{k-1}H^k} \sum_{j=1}^{\infty} \frac{1}{j^k}\ . \tag{63}$$

□

To account for the well-separated images of $\bar{C}_0$, we will require the corresponding multipole expansion. It is easy to verify, however, that for such squares, centered at a point w_j, the expansion is of the form

$$\mathbf{v}(z) = \sum_{k=1}^{s} \frac{\gamma_k}{(z - w_j)^k}\ . \tag{64}$$

where

$$\gamma_k = -\bar{a}_k\ . \tag{65}$$

Except for $\bar{C}_0$ and $\bar{C}_1$, all of these images are separated from C_0, and as above, the fields they induce inside the channel section are accurately representable by a local expansion,

$$\Psi_j(z) = \sum_{m=0}^{p} \delta_m \cdot z^m \tag{66}$$

with

$$\delta_m = \sum_{k=1}^{p} \frac{\gamma_k}{w_j^{m+k}} \binom{m+k-1}{k-1} (-1)^k\ . \tag{67}$$

The well-separated images $\bar{C}_j$ clearly have centers

$$..., -5Hi, -3Hi, 3Hi, 5Hi, ... \tag{68}$$

Let T be the set of integers of the form

$$\{\pm(2j+1),\ j = 1, 2, ..., \infty\} \tag{69}$$

We again account for the field due to all well-separated images by forming the coefficients of a local representation

$$\Psi(z) = \sum_{m=0}^{p} \delta_m^{total} \cdot z^m \ , \tag{70}$$

where

$$\delta_m^{total} = \sum_{k=1}^{p} \gamma_k \binom{m+k-1}{k-1} (-1)^k \left(\sum_T \frac{1}{w_j^{m+k}} \right) . \tag{71}$$

The summation over T for $(m+k) > 1$ is absolutely convergent. For $(m+k) = 1$, the series is not absolutely convergent, but symmetry considerations again dictate the choice

$$\sum_T \frac{1}{w_j} = 0. \tag{72}$$

Lemma 5.3 *For $k > 1$,*

$$\sum_T \frac{1}{w_j^k} = \begin{cases} 0 & \text{if } k \text{ is odd} \\ \frac{(2^k-1)\pi^k |B_k| (-1)^{\frac{k}{2}-2k!(-1)^{\frac{k}{2}}}}{H^k k!} & \text{if } k \text{ is even} \end{cases} \tag{73}$$

where B_k is the kth Bernoulli number.

Proof: For k odd, we have

$$\sum_T \frac{1}{w_j^k} = \sum_{j=1}^{\infty} \frac{1}{((2j+1)Hi)^k} + \sum_{j=1}^{\infty} \frac{1}{(-(2j+1)Hi)^k} \tag{74}$$

$$= 0 \ . \tag{75}$$

For k even, we have

$$\sum_T \frac{1}{w_j^k} = \sum_{j=1}^{\infty} \frac{1}{((2j+1)Hi)^k} + \sum_{j=1}^{\infty} \frac{1}{(-(2j+1)Hi)^k} \tag{76}$$

$$= \frac{2}{H^k i^k} \sum_{j=1}^{\infty} \frac{1}{(2j+1)^k} \tag{77}$$

$$= \frac{2 \cdot (-1)^{k/2}}{H^k} \sum_{j=1}^{\infty} \frac{1}{(2j+1)^k} \ . \tag{78}$$

The result now follows from the equality ([8], p. 7)

$$\sum_{j=0}^{\infty} \frac{1}{(2j+1)^{2k}} = \frac{(2^{2k}-1)\pi^{2k}}{2 \cdot (2k)!} |B_{2k}| \tag{79}$$

□

If we add the computed coefficients δ_k^{total} from (71) to the coefficients b_k^{total} from (56), we obtain a single local expansion which describes the field due to *all* sources outside the nearest neighbor squares of C_0. This local expansion can then be evaluated at all particle positions in C_0.

The final step in the algorithm is to compute the velocity field due to the *free-space* sources within C_0, $\bar{C}_0$ and $\bar{C}_1$. This problem is handled by the Fast Multipole Method (FMM), which requires an amount of work proportional to $n + m$ to evaluate the field induced by n sources at m points.

Algorithm 2

Comment [Set number of terms to be used in expansions.]

Choose the precision ϵ to be achieved. Set the number of terms
in all expansions to $s = \lceil \log_c(1/\epsilon) \rceil$.

Comment [From Algorithm 1, we are given that the number of elementary strips is K.]

Define n_i to be the number of particles in the *ith* strip.
Clearly, $n_1 + n_2 + ... + n_K = N$, the total number of particles.

Comment [Process each elementary strip .]

do $i = 1, ..., K$
Define C_0 to be the square whose central third is strip i.
Step 1
Form coefficients a_k of s-term multipole expansion about center
of C_0 induced by sources in strip i.
Form coefficients γ_k of s-term multipole expansion for square $\bar{C}_0$
via equation (65).
Step 2
Form coefficients $b_k + \delta_k$ of s-term local expansion about the
center of C_0 which describes the field induced by all reflected
sources outside the nearest neighbor squares.
Step 3
Evaluate local expansion at all particle positions in strips $i - 1$,i and $i + 1$.
Step 4
Compute velocity field induced by sources in C_0, $\bar{C}_0$ and $\bar{C}_1$
at all particle positions in strips $i - 1$, i and $i + 1$ via the FMM.
end do

A brief operation count of the Algorithm 2 follows.

Step Number	**Operation Count**	**Explanation**
Step 1	order Ns	each particle contributes to an s-term multipole expansion when its elementary strip is being processed.
Step 2	order Ks^2	The creation of a local expansion requires order s^2 work and is carried out once for each elementary strip.
Step 3	order $3Ns$	Three local expansions are evaluated for each particle.

Step 4	order $3\alpha N$	K free-space problems are solved, each of dimension n_i, with a factor of three included to account for the extra image sources and evaluation locations. The factor α represents the constant for the linear time FMM.

The estimate for the running time is therefore

$$N \cdot (4s + 3\alpha) \quad + \quad K \cdot s^2. \tag{80}$$

To summarize, then, the full algorithm consists of

1. Decomposition of the channel into elementary strips,
2. **Algorithm 1** to compute distant interactions, leaving a sequence of uncoupled nearest neighbor problems to consider,
3. **Algorithm 2** to compute nearest neighbor interactions.

6 Numerical Results

A computer program has been implemented using the channel decomposition and nearest neighbor algorithms of this paper. For testing purposes, we randomly assigned particles to positions within a channel section of length $5H$, where H was the channel width (Fig. 4), with source strengths between 0 and 1. Five digit accuracy was requested from the expansions. In the first part of the algorithm, S-expansions were computed to 10 terms, while in the second part of the algorithm, multipole and Taylor expansions were computed to about 20 terms. We performed the calculations in four ways: (1) through the algorithm of this paper in single precision; (2) directly from the Green's function in single precision; (3) directly from the Green's function in double precision; (4) via conformal mapping in single precision. The direct evaluation from the Green's function in double precision was used as a standard for comparing the relative accuracies of the other three methods in a least squares sense. Calculations were carried out on a SUN 3/50 workstation using the 68881 co-processor.

The following observations can be made from Table 1.

1. The accuracies of the results obtained by the fast algorithm are in agreement with the error bounds given in this paper. In fact, the results are consistently more accurate than either of the direct calculations.
2. The CPU time requirements of the fast algorithm appear to grow somewhat superlinearly. The reason for this is that there are two constants asssociated with the algorithm, a small one for the channel decomposition and a larger one for the FMM. The observed timings are dominated by the first constant for 100 and 400 particles, and by the second constant for the larger tests. When there are a small number of particles per strip, the FMM with its associated overhead is simply not invoked.

N	T_{alg}	T_{dir}	T_{cm}	E_{alg}	E_{dir}	E_{cm}
100	8.38	34.8	14.0	$4.5\ 10^{-7}$	$7.2\ 10^{-7}$	$1.1\ 10^{-6}$
400	53.1	551	223	$2.7\ 10^{-7}$	$4.1\ 10^{-7}$	$1.2\ 10^{-6}$
1600	398	(8820)	(3550)	$4.3\ 10^{-7}$	$1.3\ 10^{-6}$	$1.1\ 10^{-6}$
6400	1890	(141000)	(56800)	$6.9\ 10^{-7}$	$5.2\ 10^{-6}$	$3.4\ 10^{-6}$

Table 1

Table of CPU times in seconds required by the fast algorithm (alg), the direct Green's function method (dir), and conformal mapping with direct evaluation of the resulting N-body problem (cm). The least squares errors for the three methods are shown in the last three columns. Timings in parentheses are estimated by computing the results for only a subset of 100 of the particles. The corresponding errors are computed from that smaller data set.

3. By the time the number of particles reaches 6400, the fast algorithm is about 75 times more efficient than the direct Green's function method.

4. Even for as few as 100 particles, the fast algorithm is about four times faster than the direct calculation.

7 Conclusions

A fast algorithm for potential flow in channels has been developed. It is based on asymptotic expansions which we refer to as S-expansions, some analytic observations concerning classical multipole expansions and Taylor series, and the Fast Multipole Method. The asymptotic CPU time requirements for the algorithm grow linearly with the number of sources and, despite its complex structure, numerical experiments demonstrate that dramatic speedups can be obtained for even moderate size particle systems.

In its current form, the algorithm requires that the channel boundaries be straight. A method applicable to channels with perturbed boundaries will be described in a subsequent paper.

The author would like to thank V. Rokhlin for several useful conversations.

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. 111-123.

[2] A. W. Appel, *An Efficient Program for Many-body Simulation*, Siam. J. Sci. Stat. Comput., 6 (1985), pp. 85-103.

[3] J. Barnes and P. Hut, *A Hierarchical $O(N \log N)$ Force-Calculation Algorithm*, Nature, 324 (1986), pp. 446-449.

[4] J. T. Beale and A. Majda, *Vortex Methods II: Higher Order Accuracy in Two and Three Dimensions*, Math. Comp., 39 (1982), pp. 28-52.

[5] J. Carrier, L. Greengard, and V. Rokhlin, *A Fast Adaptive Multipole Algorithm for Particle Simulations*, Siam J. Sci. Stat. Comput., to appear (July, 1988).

[6] Y. Choi and J. A. C. Humphrey, *Analytical Prediction of Two-Dimensional Potential Flow Due to Fixed Vortices in a Rectangular Domain*, J. Comput. Phys., 56 (1984), pp. 15-27.

[7] A. J. Chorin, *Numerical Study of Slightly Viscous Flow*, J. Fluid. Mech., 57 (1973), pp. 785-796.

[8] I. S. Gradshteyn and I. M. Ryzhik, *Tables of Integrals, Series, and Products*, Academic Press, New York, 1980 .

[9] L. Greengard and V. Rokhlin, *A Fast Algorithm for Particle Simulations*, J. Comput. Phys., 73 (1987), pp. 325-348.

[10] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, MIT Press, Cambridge, 1988.

[11] O. Hald, *Convergence of Vortex Methods for Euler's Equations*, Siam J. Sci. Stat. Comput., 16 (1979), pp. 726-755.

[12] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*, McGraw-Hill, New York, 1981 .

[13] A. Leonard, *Vortex Methods for Flow Simulation*, J. Comput. Phys., 37 (1980), pp. 289-335.

[14] S. T. O'Donnell and V. Rokhlin, *A Fast Algorithm for the Numerical Evaluation of Conformal Mappings*, Technical Report 554, Yale Computer Science Department, 1986.

[15] A. M. Odlyzko and A. Schönhage, *Fast Algorithms for Multiple Evaluations of the Riemann Zeta Function*, Trans. Am. Math. Soc., to appear (July, 1988).

[16] L. van Dommelen and E. A. Rundensteiner, *Fast, Adaptive Summation of Point Forces in the Two-Dimensional Poisson Equation*, submitted, J. Comput. Phys.

[17] C. Zemach, *A Conformal Map Formula for Difficult Cases*, J. Comp. Appl. Math., 14 (1984), pp. 207-215.