JOURNAL OF COMPUTATIONAL PHYSICS **142,** 123–128 (1998)
ARTICLE NO. CP985907

Modification of the Carrier, Greengard, and Rokhlin FMM for Independent Source and Target Fields[1]

James H. Strickland* and Roy S. Baty†

Engineering Sciences Center, Sandia National Laboratories, Albuquerque, New Mexico 87185-0836
E-mail: *jhstric@sandia.gov and †rsbaty@sandia.gov

Received July 2, 1997; revised November 24, 1997

The Greengard–Rokhlin fast multipole method (FMM) provides an efficient numerical algorithm to calculate the two-dimensional stream function and velocity field at a number of target points associated with a large system of vortices (sources). In this paper we discuss an extension to their adaptive scheme. The added feature allows the specification of target points that do not have to coincide with the location of the sources. This is useful when specifying separate source and target fields for calculating boundary conditions, trajectories of passive scalar quantities, data for stream-function plots, etc. A simple algorithm has been developed to optimize the method for cases where the number of sources differs significantly from the number of target points.

Key Words: fast multipole; vortex methods.

INTRODUCTION

The Greengard–Rokhlin fast multipole method (FMM) provides an efficient numerical algorithm to calculate the two-dimensional stream function and velocity field associated with a large system of vortices. The general method comes from work developed by Rokhlin [1], Greengard and Rokhlin [2], and Carrier *et al.* [3]. These fast methods take advantage of multipole and local series expansions, which enables calculations to be made for interactions between groups of particles which are in well-separated spatial domains rather than having to consider interactions between every pair of particles.

One of the features of these algorithms is that the locations of sources (vortices) are the same as the locations of the targets (evaluation points). If the goal of the computation is to

[1] This work was performed in part at Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the U.S. Department of Energy under Contract DE-AC04-94AL85000.

0021-9991/98 $25.00

determine the velocity and stream function at the vortex centers this is sufficient. In some cases, however, it is useful to specify separate source and target fields so as to provide an efficient means for calculating boundary conditions, trajectories of passive scalar quantities, data for stream-function plots, etc.

In this paper we discuss an extension to the adaptive scheme of Carrier *et al.* [3]. The added feature allows specification of target point locations that may be different from those of the sources. A simple algorithm based on numerical experiments has been developed to optimize the method for cases where the number of vortices N_V differs significantly from the number of target points N_T.

GENERAL METHODOLOGY

In order to efficiently apply the multipole and local series expansions to regions for which they converge, the flow domain must be decomposed into a number of smaller domains. In the work of Carrier *et al.* [3] the domain decomposition begins by placing a square around all of the source and target points of interest. The square is then subdivided into four smaller squares. The smaller squares are further subdivided if more than a specified number of points reside within them. This process continues until the number of points in any childless box (undivided box) is less than some specified number.

In the present work, a square box is first constructed which encloses all of the N_V sources in the flow as well as all of the N_T target points. This box represents the zeroth level source box as well as the zeroth level target box. We first subdivide the source box into four boxes if it contains more than N_S vortices. Here, N_S is some specified number which will eventually be determined so as to minimize computing time. Smaller boxes which contain more than N_S vortices are further subdivided. The mesh for the target points is generated in a similar manner. Subdivision occurs until there are less than N_F target points in any given childless target box.

A somewhat involved procedure is used to define the separation condition between a particular target box and each of the source boxes in the flow. This procedure determines the way in which the influences of vortices in a particular source box on target points in a given target box are computed. In general, there are five possibilities. These possibilities are specified by placing all of the source boxes in one of five lists for a given target box. Since the definitions of these "box lists" have to be modified for separate source and target meshes it is essential to present them here in some detail.

In order to formalize the box list definitions, let NBOXI refer to the target box for which a box list is being developed. Parent boxes are boxes which have been subdivided into four smaller boxes, while childless boxes have not. A colleague box of NBOXI is a box which is adjacent to NBOXI and which has the same size (level) as NBOXI. Definitions for the five box lists are as follows:

- Box List 1: In order for NBOXI to have any list 1 boxes, NBOXI must itself be childless. If NBOXI is a childless box, then list 1 boxes consist of all childless source boxes at all levels which are adjacent to NBOXI and all childless source boxes which are contained within, congruent with, or contain NBOXI. This list defines childless source boxes which are not sufficiently separated from NBOXI to allow any of the series expansions to be used. Direct calculations must be made for this list.
- Box List 2: List 2 boxes of NBOXI are source boxes which occupy the same positions that would be occupied by children of the colleagues of NBOXI's parent that are well

separated from NBOXI. Note that the colleagues of NBOXI's parent do not actually have to exist or have children. NBOXI and its list 2 source boxes can be either parent or childless boxes. NBOXI and its list 2 boxes will be the same size (level) and will be separated by at least the dimension of one of their sides. For this case, both source domain (multipole) and target domain (local) series expansions can be used.

• Box List 3: In order for NBOXI to have any list 3 boxes, NBOXI must itself be childless. List 3 boxes can be either parent or childless source boxes. List 3 boxes occupy the positions that descendants of the colleagues of NBOXI would occupy. Note that the colleagues of NBOXI do not actually have to exist or to have descendants. The parent of the list 3 box must be adjacent to NBOXI but the list 3 box must not itself be adjacent to NBOXI. List 3 boxes will always be smaller than NBOXI. NBOXI will be separated from the list 3 box by one box which is the same size as the list 3 box. Source domain series expansions (multipole) can be used, target domain (local) expansions cannot.

• Box List 4: NBOXI can be either a parent or a childless box. List 4 boxes must be childless source boxes. NBOXI is in the position that a descendant of the colleagues of any list 4 box would occupy. Note that the colleagues of the list 4 box do not have to exist nor do they have to have descendants. The parent of NBOXI must be adjacent to the list 4 box but NBOXI must not itself be adjacent to the list 4 box. List 4 boxes will always be larger than NBOXI. NBOXI will be separated from the list 4 box by one box which is the same size as NBOXI. Target domain (local) series expansions can be used, source domain (multipole) expansions cannot.

• Box List 5: List 5 boxes consist of all source boxes that are well separated from NBOXI's parent. No calculations are necessary nor is any list ever made for these boxes. Contributions from these distant boxes reside in the parent of NBOXI.

The primary modification to the Carrier–Greengard–Rokhlin set of box lists found in Ref. [3] is in the specification of the List 1 source boxes to include those source boxes that either are contained within or contain the target box in question. Other modifications are more cosmetic than substantive but provide additional help in the interpretation of the rules.

Aside from the generation of separate meshes for the sources and targets along with the indicated modifications to the box lists, the computational algorithm is the same as that used in Ref. [3]. It should be noted, however, that we now have two parameters, N_S and N_F, which represent the maximum number of points in a childless box. These parameters must be chosen to minimize computational time.

BENCHMARK TESTS

The important dependent variables in the benchmark tests are the CPU run times and the truncation errors associated with the calculation of field variables. The independent variables are the number of vortex sources N_V and target points N_T in the field, the distribution of the sources and target points in the field, the number of terms used in the multipole and local series expansions, and the maximum number of source points N_S and target points N_F allowed in any childless box in the source and target meshes.

Four general configurations of vortex and target point placements have been studied. These configurations are indicated in Fig. 1. In the first set of cases, the source and target points are uniformly placed. The total number of source points may be different from the total number of target points, which holds true for any of the configurations studied. This is labeled as the uniform–uniform (U-U) case. For the second set of cases, all of the source

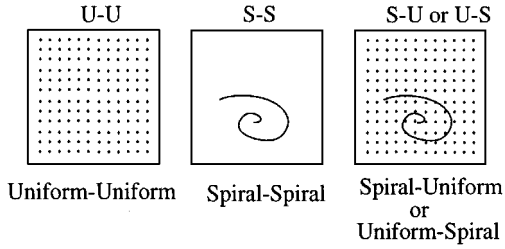

FIG. 1. Benchmark test cases.

and target points are arranged on a spiral. This case is designated as the spiral–spiral (S-S) case. For the third set of cases, the sources are on the spiral while the targets are uniformly spaced (S-U). Finally, for the fourth set, the sources are uniformly spaced while the targets are on the spiral (U-S).

The metric which we have used to assess the error incurred by truncation of the multipole and local series for the stream function and the magnitude of the velocity vector field are given by the following equations:

$$\varepsilon_\psi = \left[\frac{\sum_{i=1}^{N_V}(|\psi|_i - |\psi|_{ei})^2}{\sum_{i=1}^{N_V}|\psi|_{ei}^2}\right]^{1/2}, \quad \varepsilon_U = \left[\frac{\sum_{i=1}^{N_V}(|U|_i - |U|_{ei})^2}{\sum_{i=1}^{N_V}|U|_{ei}^2}\right]^{1/2}. \tag{1}$$

Here, $|\psi|_i$ and $|U|_i$ are the magnitudes of the stream function and velocity vector at point i as calculated from the truncated series. The quantities $|\psi|_{ei}$ and $|U|_{ei}$ are the magnitudes of the stream function and velocity vector at point i as calculated exactly using the direct method.

RESULTS

In order to investigate the relationship between accuracy, CPU time, and optimal values for the maximum number of sources and target points in a box, we first began by making a series of runs for the U-U configuration with 10,202 unit strength vortices. In this series, the maximum number of sources N_S and target points N_F in a box were set equal to each other and varied between 10 and 60. The number of terms used in the series expansions N_{max} was varied between 4 and 20. The error in the stream function ε_ψ was found to be on the order of 1×10^{-6} when the number of terms in the series expansion $N_{max} \geq 6$. The error in the velocity ε_U for $N_{max} \geq 6$ was found to be less than 1×10^{-4}, which is probably acceptable for most engineering calculations. The error appeared to be insensitive to the values of N_S and N_F over the range 10 to 60. The CPU time was minimized for a range of values with N_S and N_F between 20 and 60.

In order to examine the effect of varying N_S, N_F, N_V, and N_T on CPU time, 150 cases were run with $N_{max} = 6$. Three of the set of six contour plots generated are shown in Fig. 2a. The contours represent lines of constant CPU time which are normalized by dividing actual CPU time by the minimum CPU time associated with a given plot. These plots suggest that in order to make optimal choices for N_S and N_F an algorithm such as the following might be used:

$$N_S = 30 \max(1, \sqrt{N_V/N_T}), \quad N_F = 30 \max(1, \sqrt{N_T/N_V}). \tag{2}$$

The symbol ✠ in Fig. 2 represents application of Eq. (2) to the various cases.

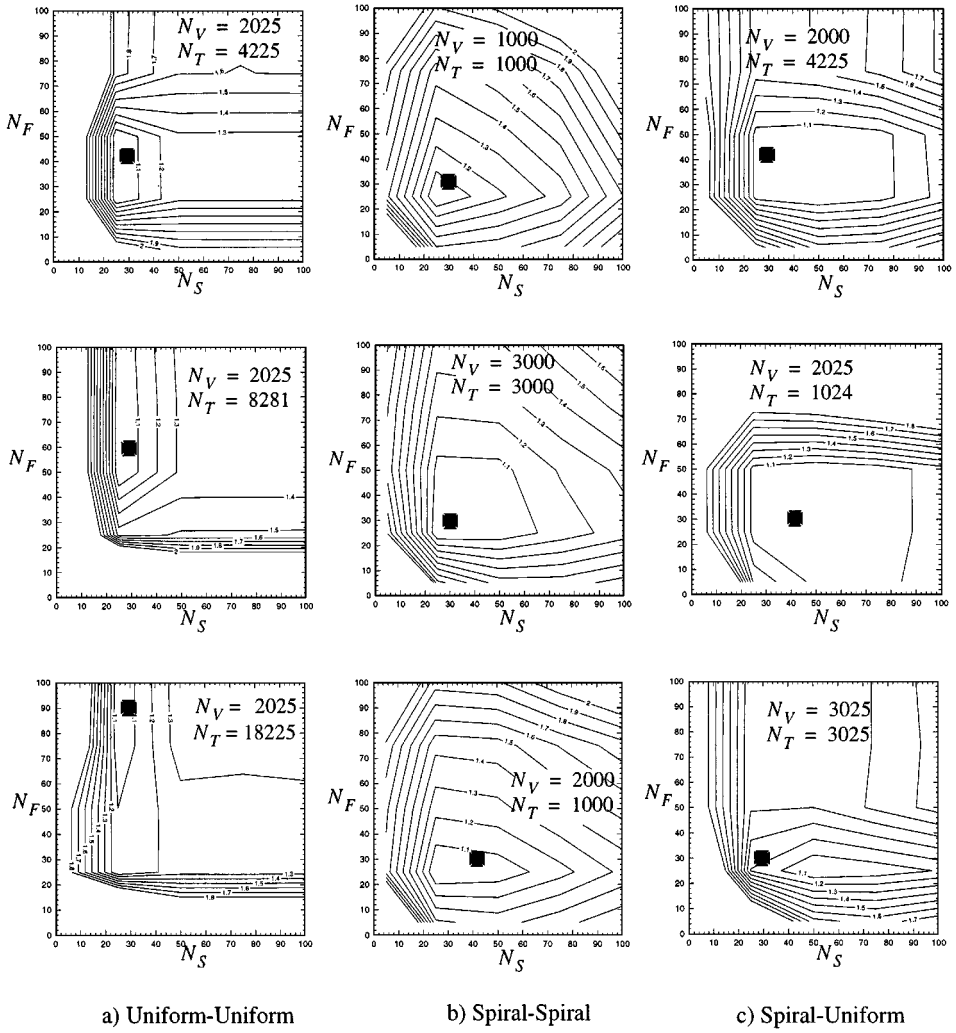

FIG. 2. Normalized contour plots of CPU time ($N_{\max} = 6$).

A detailed analytical development for calculating the optimal value for N_S and N_F is not too useful since it leads to two nonlinear algebraic equations which contain constants that are themselves functions of the computer system, algorithmic implementation, and geometry of the problem. However, if one minimizes the operation count based on the ratio of N_F/N_S it is found that the optimal value for $N_F/N_S \approx \sqrt{N_T/N_V}$. The three plots in Fig. 2a where $N_T > N_V$ suggest that the optimal choice for N_S is a constant equal to about 30. For the three cases (not shown) in which N_V and N_T are interchanged, one observes that the optimal choice for N_F is now a constant equal to about 30. These observations ($N_S \approx 30$, $N_T > N_V$ and $N_F \approx 30$, $N_V > N_T$) along with the minimization result ($N_F/N_S \approx \sqrt{N_T/N_V}$) are sufficient to construct Eq. (2).

Based on Fig. 2 and 13 other contour plots for the U-U, S-S, S-U, and U-S configurations, the use of Eq. (2) yields CPU values which are typically less than 10% above the optimal and in one case (Fig. 2c, bottom plot) 20% above the optimal. The 13 plots not shown here for the most part represent symmetries in the ratio N_V to N_T or in the configuration itself (S-U versus U-S). In general, the results also reflect these symmetries.

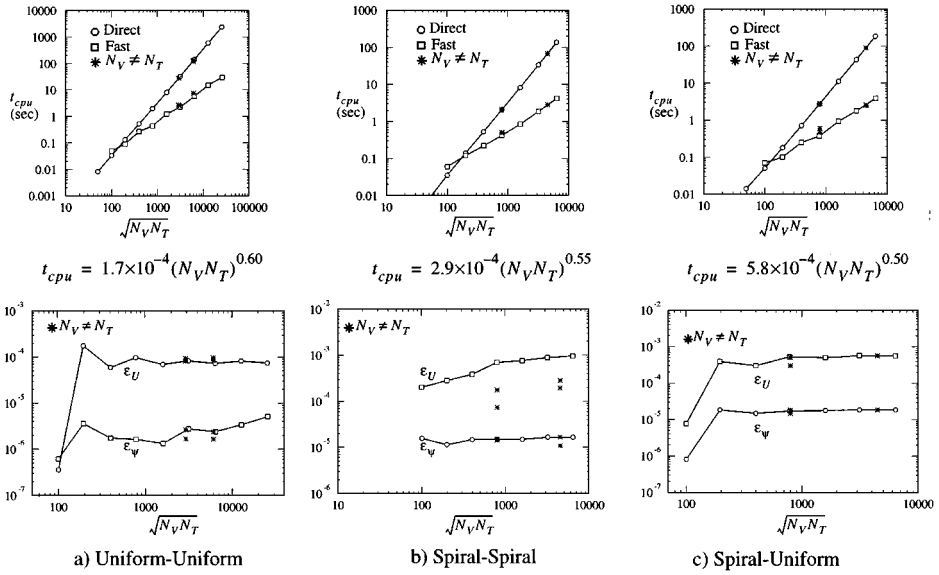

FIG. 3. CPU time and error ($N_{\max} = 6$).

We now examine the CPU times and errors when $N_{\max} = 6$ and N_S and N_F are governed by Eq. (2). For the U-U configuration, a set of runs were made with $N_V = N_T$ where the number of vortices and target points varied from about 100 to 30,000. In addition, four cases were run where $N_V/N_T = (1/9, 1/2, 2, 9)$. A similar set of runs were made for the S-S, S-U, and U-S configurations, although we do not present the U-S results herein since they are very similar to the S-U results. Four cases where $N_V \neq N_T$ were run for the S-S, S-U, and U-S configurations with $N_V/N_T = (1/4, 1/2, 2, 4)$. CPU time and error results for the U-U, S-S, and S-U, configurations are plotted in Fig. 3 versus $\sqrt{N_V N_T}$.

From Fig. 3 it can be seen that the CPU time data are correlated very well with the parameter $\sqrt{N_V N_T}$ for each of the configurations. In general, the error data for cases where $N_V = N_T$ and $N_V \neq N_T$ correlated reasonably well. An exception to this is the velocity error associated with the S-S configuration.

Additional tabulated and graphical data including detailed timing studies for individual operations in the fast solver may be found in Ref. [4].

REFERENCES

1. V. Rokhlin, Rapid solution of integral equations of classical potential theory, *J. Comput. Phys.* **60**, 187 (1983).
2. L. Greengard and V. Rokhlin, A fast algorithm for particle simulations, *J. Comput. Phys.* **73**, 325 (1987).
3. J. Carrier, L. Greengard, and V. Rokhlin, A fast adaptive multipole algorithm for particle simulations, *SIAM J. Sci. Statist. Comput.* **9**, 669 (1988).
4. J. H. Strickland and R. S. Baty, *A Two-Dimensional Fast Solver for Arbitrary Vortex Distributions*, Sandia National Laboratory Report SAND97-0880, April 1997.