AIAA JOURNAL
Vol. 30, No. 3, March 1992

# Fast Solver for Systems of Axisymmetric Ring Vortices

James H. Strickland* and Donald E. Amos†
*Sandia National Laboratories, Albuquerque, New Mexico 87185*

**A method that is capable of efficient calculation of the axisymmetric flowfield produced by a large system of ring vortices is presented in this paper. The system of ring vortices can, in turn, be used to model body surfaces and wakes in incompressible unsteady axisymmetric flowfields. This method takes advantage of source-point and field-point series expansions, which enables one to make calculations for interactions between groups of vortices that are in well-separated spatial domains rather than having to consider interactions between every pair of vortices. A Fortran computer code RSOLV has been written to execute the fast solution technique. Test cases have been run to optimize the code and to benchmark the truncation errors and CPU time savings associated with the method. For 100 vortices in the field, there is virtually no CPU time savings with the fast solver. For 10,000 vortices in the flow, the fast solver obtains solutions in about 1–3% of the time required for the direct solution technique. Formulas for the self-induced velocity of discretized regions of the flowfield have been developed. Use of these formulas allows one to correctly convect discretized patches of vorticity in the flowfield.**

## Nomenclature

$A_c$ = vortex core area
$A_{n,m}$ = source coefficient
$A'_{n,m}$ = translated source coefficient
$a$ = circular core radius
$B_{n,m}$ = field coefficient
$B'_{n,m}$ = translated field coefficient
$b$ = vortex ring or source box radius
$b_i$ = ring radius of $i$th vortex
$C_a$ = circular core axial velocity function
$C_s$ = square core axial velocity function
$D_F$ = field domain
$D_S$ = source domain
$D_1$ = source domain 1
$D_2$ = source domain 2
$D_3$ = field domain 3
$D_4$ = field domain 4
$E_V$ = total velocity error
$E_\psi$ = stream-function error
$F$ = ring vortex function
$F_i$ = ring vortex function for $i$th vortex
$I$ = index
$i$ = index
$J_0$ = zeroth-order Bessel function
$J_1$ = first-order Bessel function
$k$ = dummy variable
$L$ = index
$l$ = index or level
$m$ = index
$N$ = maximum order of the partials of $F$
$N_c$ = maximum number of vortices in any childless box
$N_p$ = number of charged particles
$N_v$ = number of vortices
$n$ = index
$P$ = field point
$p$ = index
$q$ = index
$R_F$ = radius of field domain
$R_S$ = radius of source domain
$r$ = radial coordinate
$r_c$ = radial spatial center
$r_f$ = radial distance from center of field domain 3 to field point $P$
$r_o$ = radial center of impulse
$r_p$ = radius at point $P$
$S_l$ = box dimension at level 1
$S_r$ = spiral radius
$s$ = square core size
$t_{cpu}$ = computer processing time
$U_o$ = axial core self-induced velocity
$u$ = axial velocity
$u_i$ = axial velocity at $i$th vortex due to other vortices
$u_{oi}$ = self-induced axial velocity at $i$th vortex
$u_p$ = axial velocity at point $P$
$V_{ei}$ = total velocity at point $i$ obtained using direct method
$V_i$ = total velocity at point $i$ obtained using fast solver
$V_o$ = radial core self-induced velocity
$v$ = radial velocity
$v_i$ = radial velocity at $i$th vortex due to other vortices
$v_p$ = radial velocity at point $P$
$x$ = axial coordinate
$x_c$ = axial spiral center
$x_f$ = axial distance from center of field domain 3 to field point $P$
$x_i$ = axial coordinate of $i$th vortex
$x_o$ = axial center of impulse
$x_p$ = axial distance to point $P$
$\Gamma$ = strength of vortex
$\Gamma_i$ = strength of $i$th vortex
$\Delta b_i$ = radial distance from center of source domain to $i$th vortex
$\Delta b_s$ = radial distance from center of source domain 1 to source domain 2
$\Delta r_f$ = radial distance from center of field domain 3 to field domain 4
$\Delta x_f$ = axial distance from center of field domain 3 to field domain 4
$\Delta x_i$ = axial distance from center of source domain to $i$th vortex
$\Delta x_s$ = axial distance from center of source domain 1 to source domain 2
$\rho$ = fluid density
$\psi$ = stream function

Received Oct. 16, 1990; revision received Jan. 28, 1991; accepted for publication Feb. 6, 1991. 

*Senior Member Technical Staff, Parachute Systems Division. Associate Fellow AIAA.
†Distinguished Member Technical Staff, Numerical Mathematics Division.

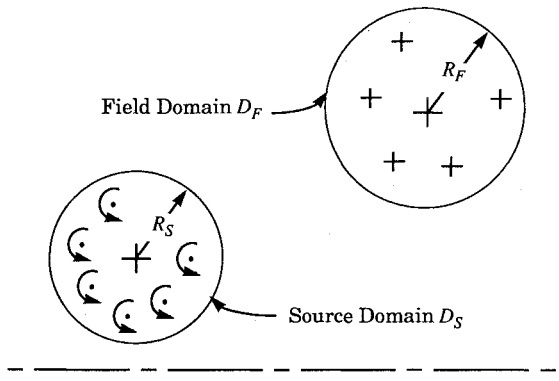



**Fig. 1 Source- and field-point domains.**

$\psi_{ei}$ = stream function at point $i$ obtained using direct method
$\psi_i$ = stream function of $i$th vortex
$\psi_p$ = stream function at point $P$
$\omega$ = vorticity

## Introduction

MOTIVATION for the present work results from a need to be able to perform fast yet accurate calculations of the axisymmetric flowfield produced by large numbers of ring vortices. Such systems of ring vortices can be used, in turn, to model body surfaces and wakes in incompressible unsteady axisymmetric flowfields. Solutions for the flow over streamlined as well as bluff bodies can be simulated. Surface porosity and flexible geometries are also easily modeled using such methods. Bluff-body simulations present the greatest challenge since the wake structure and, thus, the positions of the ring vortices are not known a priori. The simulations for bluff bodies require that the wake structure evolve as a function of time, even for steady flow calculations. Large numbers of vortices ($100 < N_v < 10{,}000$) are required for such simulations. Since the wake must be convected as part of the evolution process, the velocity perturbations from all vortices on all other vortices must be calculated at each time step. This requires an amount of work that is proportional to $N_v^2$ for each time step. If the amount of work can be reduced to being proportional to $N_v$ or even $N_v \log(N_v)$, then considerable CPU time will be saved in large simulations.

The genesis of this method comes from work done by Carrier et al.[1] in which an algorithm to calculate potential and force fields for a large system of charged particles was developed. In that work, it was shown that the asymptotic CPU time was of order $N_p$, where $N_p$ was the number of particles. This was contrasted to CPU times of order $N_p^2$ when traditional methods were used. This method takes advantage of source-point and field-point series expansions, which enables one to make calculations for interactions between groups of particles that are in well-separated spatial domains rather than having to consider interactions between every pair of particles. In the present work, the methodology concerning the use of the spatial domains in which vortex rings are located is essentially the same as that used in Ref. 1 for charged particles. In the present work, the series expansion for the stream function of the vortex is used in place of the expansion for the potential of the charged particles of Ref. 1. Likewise, the two velocity components produced at a point by the vortex rings are obtained explicitly from the stream-function expansion just as the two components of force produced by the charged particles are obtained from their potential function.

In order to gain introductory insight into the method, consider the two domains $D_S$ and $D_F$, which are depicted in Fig. 1. Domain $D_S$ contains a set of vortices that are acting upon (producing perturbation velocities at) a set of points contained in domain $D_F$. In most cases, the set of points in domain $D_F$ would represent the centers of a second set of vortices. First, a Taylor series expansion for the stream function about the center of the source domain $D_S$ is obtained. This expansion includes all of the vortices that are inside $D_S$. The series converges for field points that are outside of $D_S$. In order to be somewhat conservative, the expansion about the center of the source domain is used only for field points that are separated by at least $R_S$ from the boundary of the source domain. Next, a Taylor series expansion for the stream function is obtained about the center of the field domain $D_F$. The series converges for field points that are inside $D_F$. The expansion about the center of the field domain is used only for source points that are separated by at least $R_F$ from the boundary of the field domain. In summary, the method allows one to calculate efficiently the influence of all of the vortices in $D_S$ at all of the points in $D_F$ when the two domains are well separated, as defined previously. As will be shown later, there is additional efficiency inherent in the method if one exploits the possibility of moving the centers of expansion of the Taylor series.

## Source- and Field-Point Expansions

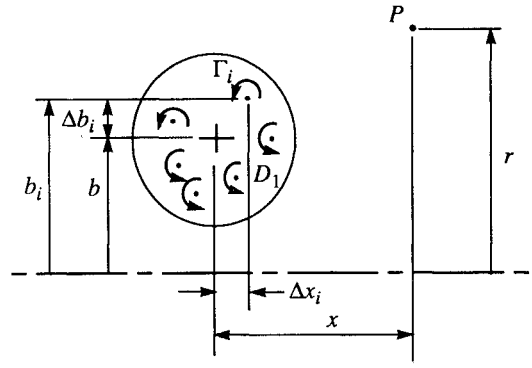
Consider a single vortex ring located at an axial location $x = 0$ with a ring radius equal to $b$. The stream function at a field point $(x,r)$ can be expressed according to Lamb[2] as

$$\psi_i = \frac{1}{2}\Gamma_i br \int_0^\infty e^{-kx} J_1(kr) J_1(kb)\, dk \tag{1}$$

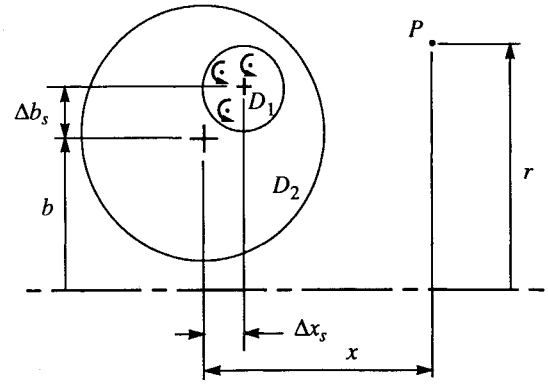
The axial and radial velocity components are given by

$$u = \frac{1}{r}\frac{\partial \psi}{\partial r} \tag{2a}$$

$$v = -\frac{1}{r}\frac{\partial \psi}{\partial x} \tag{2b}$$




**a) Expansion about center of source domain**




**b) Translation of source domain center**

**Fig. 2 Source-domain expansion and translation.**

For the present purposes, let $\psi_i$ be represented by

$$\psi_i = \tfrac{1}{2}\Gamma_i F(x,r,b) \tag{3a}$$

$$F \equiv br \int_0^\infty e^{-kx} J_1(kr) J_1(kb)\, dk \tag{3b}$$

**Expansion Within the Source Domain**

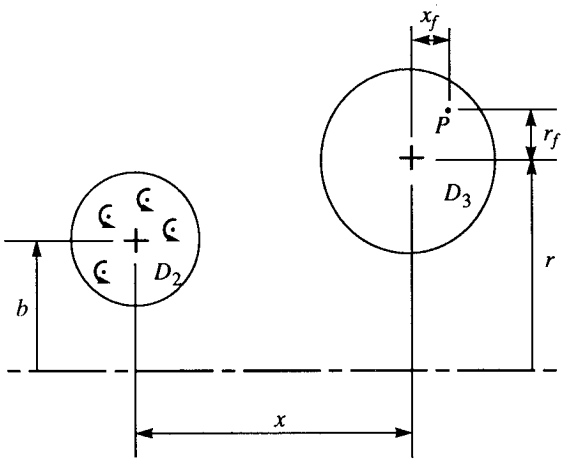
For a cluster of vortex rings, as shown in Fig. 2a, a Taylor series expansion can be obtained that is valid at a field point $P$ outside of a domain $D_1$ enclosing the cluster. The Taylor series for the stream function at point $P$ resulting from this cluster can first be written symbolically as

$$\psi = \sum_i \frac{1}{2}\Gamma_i F_i(x - \Delta x_i, r, b + \Delta b_i) \tag{4}$$

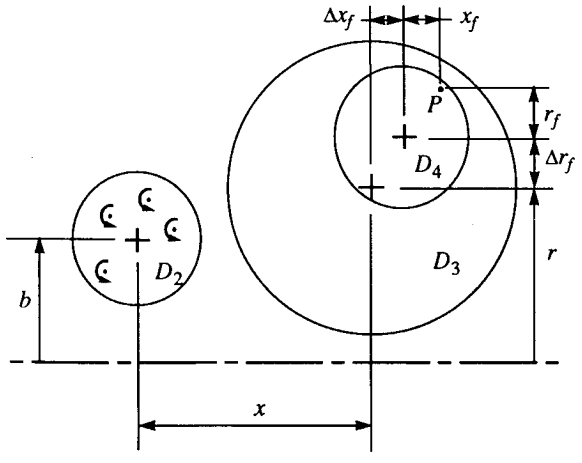
Formalizing the expansion yields a series in the partial derivatives of the function $F$. The coefficients $A$ of this series are a function of the individual vortex strengths and their distance from the center of the expansion:

$$\psi = \sum_{n=0}^{\infty} \sum_{m=0}^{n} A_{n-m,m} \frac{\partial^n F(x,r,b)}{\partial x^{n-m} \partial b^m} \tag{5a}$$

$$A_{n-m,m} = \sum_i \frac{1}{2}\Gamma_i \frac{(-\Delta x_i)^{n-m} \Delta b_i^m}{(n-m)!m!} \tag{5b}$$




**a) Expansion about center of field domain**




**b) Translation of field-domain center**

**Fig. 3 Field-domain expansion and translation.**

The first summation in Eq. (5a) is, of course, calculated as a finite sum. Denoting the maximum value of $n$ used in the calculation by $N$, later results show that accurate stream-function values are obtained for values of $N$ on the order of 5 or 6.

**Translation of the Source Domain**

As indicated previously, it is advantageous to be able to move the center of the expansion given by Eqs. (5) such that the coefficients of the Taylor series in domain $D_1$ can be added to the coefficients of other domains contained in a larger domain $D_2$. This situation is depicted in Fig. 2b. The Taylor series for the stream function at point $P$ resulting from this translation can first be written symbolically as

$$\psi = \sum_{n=0}^{\infty} \sum_{m=0}^{n} A_{n-m,m} \frac{\partial^n F(x - \Delta x_s, r, b + \Delta b_s)}{\partial x^{n-m} \partial b^m} \tag{6}$$

Formalizing the expansion again yields a series in the partial derivatives of the function $F$. The coefficients $A'$ of this series are a function of the old coefficients $A$ and the coordinates between the two centers of expansion:

$$\psi = \sum_{n=0}^{\infty} \sum_{m=0}^{n} A'_{n-m,m} \frac{\partial^n F(x,r,b)}{\partial x^{n-m} \partial b^m} \tag{7a}$$

$$A'_{n-m,m} = \sum_{q=0}^{m} \sum_{p=q}^{q+n-m} A_{n-m-p+q,m-q} \frac{(-\Delta x_s)^{p-q} \Delta b_s^q}{(p-q)!q!} \tag{7b}$$

**Expansion Within the Field Domain**

A Taylor series expansion will now be obtained about the center of a field-point domain $D_3$ due to the influence of vortices in $D_2$. Referring to Fig. 3a, it can be seen that this expansion will be valid at field points inside $D_3$ if $D_2$ and $D_3$ are well separated. The Taylor series for the stream function at point $P$ resulting from the expansion about the center of $D_3$ can be written symbolically as

$$\psi = \sum_{n=0}^{\infty} \sum_{m=0}^{n} A'_{n-m,m} \frac{\partial^n F(x + x_f, r + r_f, b)}{\partial x^{n-m} \partial b^m} \tag{8}$$

Formalizing the expansion yields a power series in $r_f$ and $x_f$. The coefficients $B$ of this series are a function of the partial derivatives of the function $F$ and the old coefficients $A'$:

$$\psi = \sum_{n=0}^{\infty} \sum_{m=0}^{n} B_{n-m,m} x_f^{n-m} r_f^m \tag{9a}$$

$$B_{n-m,m} = \sum_{p=0}^{L} \sum_{q=0}^{p} \frac{A'_{p-q,q}}{(n-m)!m!} \frac{\partial^{n+p} F(x,r,b)}{\partial x^{n-m+p-q} \partial r^m \partial b^q} \tag{9b}$$

In Eq. (9b), the summation limit $L$ will be equal to the maximum value of $n$ less $n$ itself (i.e., $L = N - n$).

**Translation of the Field Domain**

The last Taylor series expansion involves the translation of the center of the field-point domain. As Fig. 3b depicts, the expansion for the domain $D_3$ can be used for any domain $D_4$ that is inside of $D_3$ by performing a simple translation of the expansion center. The symbolic Taylor series expansion for the stream function at point $P$ about the center of $D_4$ is

$$\psi = \sum_{n=0}^{\infty} \sum_{m=0}^{n} B_{n-m,m} (x_f + \Delta x_f)^{n-m} (r_f + \Delta r_f)^m \tag{10}$$

Formalizing the expansion again yields a power series in $r_f$ and $x_f$. The coefficients $B'$ of this series are a function of $\Delta r_f$ and $\Delta x_f$ and the old coefficients $B$:

$$\psi = \sum_{n=0}^{\infty} \sum_{m=0}^{n} B'_{n-m,m} x_f^{n-m} r_f^m \tag{11a}$$

$$B'_{n-m,m} = \sum_{p=0}^{L} \sum_{q=0}^{p} B_{n-m+p-q,m+q} \frac{\Delta x_f^{p-q} \Delta r_f^q (n-m+p-q)!(m+q)!}{(p-q)!q!(n-m)!m!} \tag{11b}$$

As before, the summation limit $L$ in Eq. (11b) will be equal to $N - n$.

**Evaluation of the Partial Derivatives of $F$**

There are several methods that one could use to obtain the various partial derivatives of the function $F$. One method would be to simply differentiate Eqs. (3) with respect to $x$, $r$, and/or $b$ the required number of times. The method used in the present work makes use of the partial differential equation for the stream function, which is satisfied by Eqs. (3), to develop relationships between various partial derivatives of $F$. This eliminates the need to directly calculate many of the partials since the equation for the stream function can be used to obtain recursion relationships.

The axisymmetric stream-function equation for irrotational flow is given by Lamb[2] as

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial r^2} = \frac{1}{r} \frac{\partial \psi}{\partial r} \tag{12}$$

Equation (12) used in conjunction with Eqs. (3) yields two equations in $F$ if one takes advantage of the symmetry between $r$ and $b$:

$$\frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial r^2} = \frac{1}{r} \frac{\partial F}{\partial r} \tag{13a}$$

$$\frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial b^2} = \frac{1}{b} \frac{\partial F}{\partial b} \tag{13b}$$

Equations represented by Eqs. (13) can be differentiated to obtain the following relationships for the $(l + m + n)$th order partials in $F$:

$$\frac{\partial^{l+m+n} F}{\partial x^l \, \partial r^m \, \partial b^n} = -\frac{\partial^{l+m+n} F}{\partial x^{l+2} \, \partial r^{m-2} \, \partial b^n} + \sum_{j=0}^{m-2} \frac{(-1)^j}{r^{j+1}} \frac{(m-2)!}{(m-2-j)!} \frac{\partial^{l+m+n-j-1} F}{\partial x^l \, \partial r^{m-j-1} \, \partial b^n} \tag{14a}$$

$$\frac{\partial^{l+m+n} F}{\partial x^l \, \partial r^m \, \partial b^n} = -\frac{\partial^{l+m+n} F}{\partial x^{l+2} \, \partial r^m \, \partial b^{n-2}} + \sum_{j=0}^{n-2} \frac{(-1)^j}{b^{j+1}} \frac{(n-2)!}{(n-2-j)!} \frac{\partial^{l+m+n-j-1} F}{\partial x^l \, \partial r^m \, \partial b^{n-j-1}} \tag{14b}$$

In Eq. (14a), $m = 0, 1$ is forbidden, as is $n = 0, 1$ in Eq. (14b). Therefore, for each value of $p = l + m + n$, the following four partial derivatives cannot be obtained from Eqs. (14):

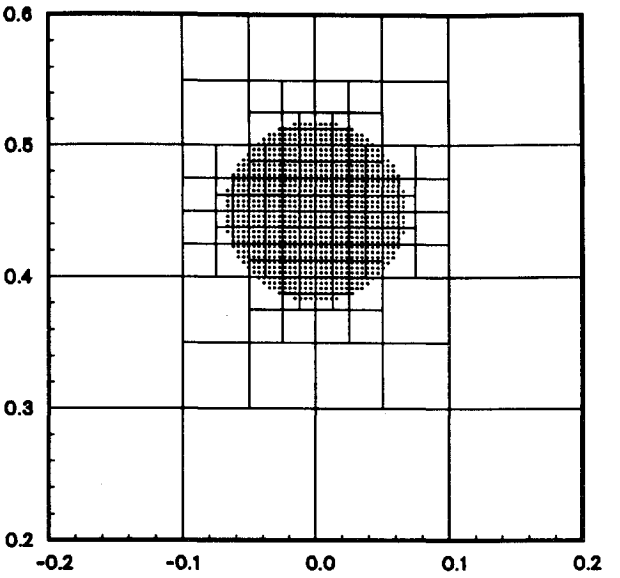
$$\frac{\partial^p F}{\partial x^p}, \quad \frac{\partial^p F}{\partial x^{p-1} \, \partial r}, \quad \frac{\partial^p F}{\partial x^{p-1} \, \partial b}, \quad \frac{\partial^p F}{\partial x^{p-2} \, \partial r \, \partial b} \tag{15}$$

These four basic partial derivatives must be calculated directly. For each value of $p$ there are $\frac{1}{2}(p + 1)(p + 2)$ partial derivatives of order $p$. The equations indicated by Eqs. (14) represent $\frac{1}{2}(p + 1)(p + 2) - 4$ independent equations for each value of $p$. Equation (14a) can be used to obtain $\frac{1}{2}p(p - 1)$ partials. Equation (14b) can be used to obtain another set of $\frac{1}{2}p(p - 1)$ partials. The two sets are not entirely independent. A recursive solution can be obtained for each value of $p$ by first assuming that the partials in Eqs. (15) are already calculated. The calculation starts by using Eq. (14a) with the highest derivative of $x$. The order of $r$ is systematically reduced, and the order of $b$ is increased. If Eq. (14a) becomes invalid ($m = 0, 1$), then Eq. (14b) is used. Additional details explaining the use of the recursive relationships are given in Ref. 3.

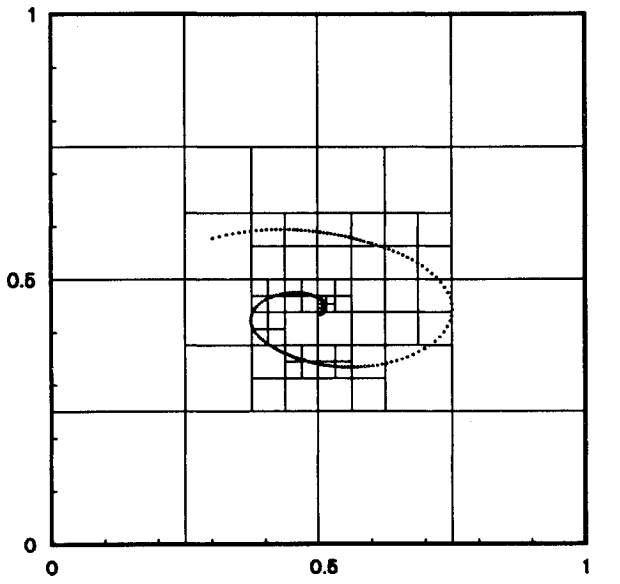
The four basic partial derivatives of $F$ of Eqs. (15) can be obtained by taking the indicated partial derivatives of $F$. Using the definition of $F$ from Eqs. (3), one obtains

$$\frac{\partial^n F}{\partial x^n} = (-1)^n br \int_0^{\infty} e^{-kx} k^n J_1(kb) J_1(kr) \, dk \tag{16a}$$

$$\frac{\partial^n F}{\partial x^{n-1} \, \partial b} = (-1)^{n+1} br \int_0^{\infty} e^{-kx} k^n J_0(kb) J_1(kr) \, dk \tag{16b}$$




a) $N_v = 973$, $N_c = 20$




b) $N_v = 200$, $N_c = 10$

**Fig. 4 Mesh generation examples.**

$$\frac{\partial^n F}{\partial x^{n-1}\,\partial r} = (-1)^{n+1} br \int_0^\infty e^{-kx} k^n J_1(kb) J_0(kr)\,dk \quad (16c)$$

$$\frac{\partial^n F}{\partial x^{n-2}\,\partial r\,\partial b} = (-1)^n br \int_0^\infty e^{-kx} k^n J_0(kb) J_0(kr)\,dk \quad (16d)$$

The four integrals in Eqs. (16) are evaluated in a very efficient manner using associated Legendre functions. The Legendre functions are calculated using a set of recursive relationships given in Ref. 4 and have been checked against the tables of Ref. 5. Additional details concerning evaluation of these integrals are given in Ref. 3.

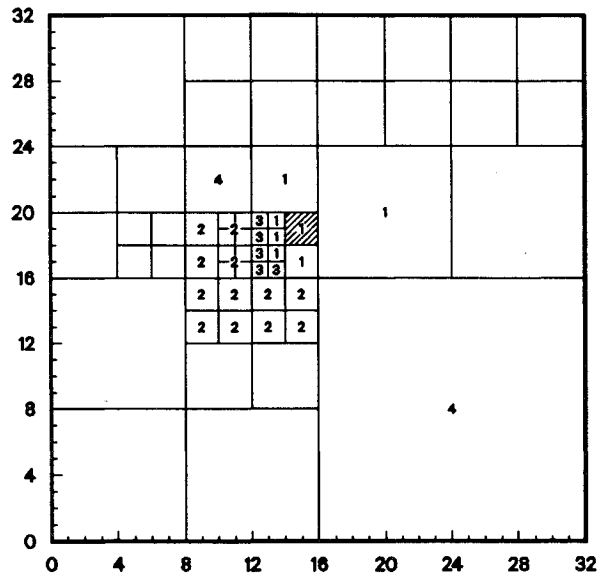
## Method of Solution

The present work is implemented by the computer code Ring SOLVer (RSOLV) developed at Sandia National Laboratories. This code requires that one specify a distribution of vortex rings by their axial locations, ring diameters, and strengths and place such information in an input file. The code also requires one to specify the domain of interest, the maximum number of vortices in a box $N_c$, and the order $N$ of the Taylor series to be used. RSOLV output consists of an output file with values of the stream function, axial velocity, and radial velocity at all of the vortex locations. The following describes the methodology used in the RSOLV code.

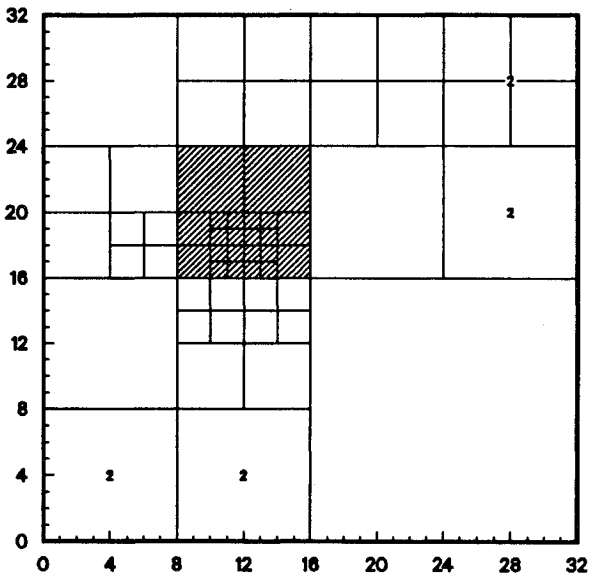
### Mesh Generation

The method requires that an adaptive mesh be generated that encloses all the $N_v$ vortices of interest in the flow. A square box with dimensions $S_0$ is first constructed that encloses all of the chosen vortex rings. This box is subdivided into four equal boxes. If any of these four boxes contains more than a specified number of vortices $N_c$, that particular box is subdivided into four more boxes. Vortices residing on mesh lines are associated with the box whose center has a more positive $x$ or $r$ value than the $x$ or $r$ value defining the mesh line. This process is continued until all of the boxes contain less than the specified number of vortices. Each box size is associated with a box level $l$ with the original box level being equal to 0. Therefore, at a particular level $l$, the box dimension $S_l$ is given by

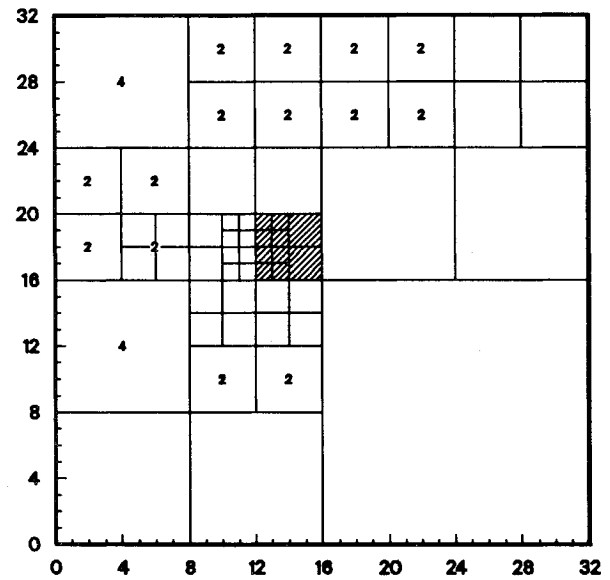
$$S_l = S_0/2^l \quad (17)$$



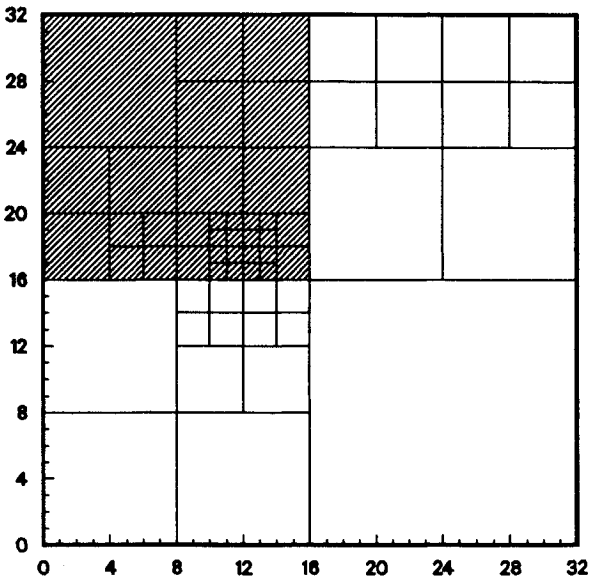
a) Contribution from NBOXI's own box list



c) Contribution from NBOXI's grandparent's box list



b) Contribution from NBOXI's parent's box list



d) Contribution from NBOXI's great grandparent's box list

Fig. 5 Box list example.

At each level, empty boxes are ignored. Boxes that can be subdivided are parent boxes and those that cannot are childless boxes. A list of vortices residing in each parent or childless box at each level is maintained. Each vortex is identified by a unique label (number). Figures 4 provide examples of the adaptive mesh generation capability of RSOLV.

### Box Lists

The purpose of defining a box list for each box at each level is to define the separation condition between the selected box and all other boxes in the mesh. This, in turn, allows one to efficiently calculate the influence of vortices in one box on the stream function in the other. In general, each box at each level has five possible relationships with other boxes in the mesh. The five lists define restrictions on the use of the Taylor series expansions between domains defined by the two boxes.

In order to formalize the box list definitions, let NBOXI refer to the mesh box for which a box list is being developed. As mentioned previously, parent boxes are boxes that can be subdivided into four smaller boxes, whereas childless boxes cannot. Thus, parent boxes contain more than $N_c$ vortices. Empty boxes contain no vortices, whereas childless boxes contain from 1 to $N_c$ vortices. A colleague box of NBOXI is a box that is adjacent to NBOXI and has the same size (level) as NBOXI. There are at most eight colleague boxes associated with NBOXI. Definitions for the five boxlists 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 consists of NBOXI and all childless boxes at all levels that are adjacent to NBOXI. This list defines boxes that are not separated from each other and cannot be subdivided. Direct calculations must be made. Taylor series expansions cannot be used.

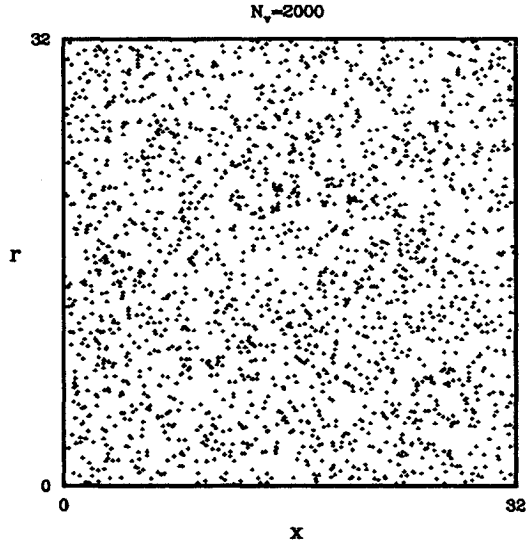
*Box list 2.* List 2 boxes of NBOXI are boxes that are children of the colleagues of NBOXI's parent that are well separated from NBOXI. NBOXI and its list 2 boxes can be either parent or childless boxes. NBOXI and its list 2 boxes will be the same size (level) and will be separated by at least the dimension of one of their sides. Both source-domain and field-domain Taylor 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 boxes. List 3 boxes are descendants of the colleagues of NBOXI. 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 that is the same size as the list 3 box. Source-domain Taylor series expansions can be used; field-domain expansions cannot.

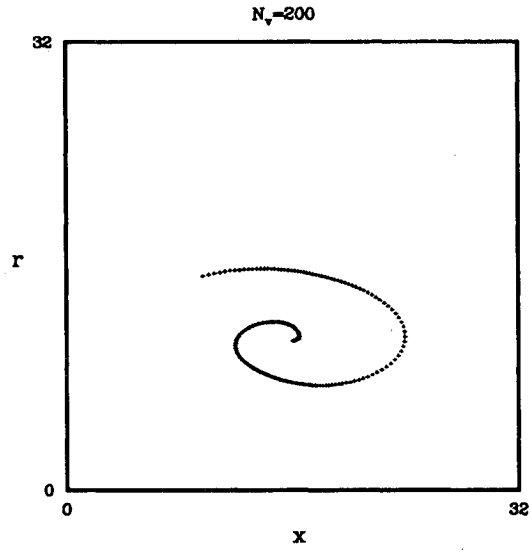
*Box list 4.* NBOXI can be either a parent or childless box. List 4 boxes must be childless. NBOXI is a descendant of the colleagues of any list 4 box. The parent of NBOXI must be adjacent to the list 4 box but NBOXI must not itself be adjacent to the list 4 box. List 4 boxes will always be larger than NBOXI. NBOXI will be separated from the list 4 box by one box that is the same size as NBOXI. Field-domain Taylor series expansions can be used; source-domain expansions cannot.

*Box list 5.* List 5 boxes consist of all boxes that are well separated from NBOXI's parent. No calculations are necessary. Contributions from these distant boxes reside in the parent of NBOXI.

To demonstrate how the set of box lists brings information into the field box NBOXI, consider the example given in Figs. 5. NBOXI is the cross-hatched box shown in Fig. 5a. Assume that the indicated mesh has been generated for a set of vortices and that there are no empty boxes. Therefore, one would expect every box to contribute to the stream function in NBOXI. From Fig. 5a, it can be seen that there are contributions from NBOXI's own box list and that this list contains box lists 1–4. As indicated previously, box list 5 does not contribute directly and is, thus, not even indicated in Figs. 5. In Fig. 5b, there are several list 2 and 4 boxes that contribute to NBOXI's parent. Since this is a parent box, it does not have any list 1 or 3 boxes associated with it. The information from NBOXI's parent is fed into NBOXI by translation of the field-domain center of the parent of NBOXI to the center of NBOXI. Information from NBOXI's grandparent's list boxes (see Fig. 5c) were previously fed into NBOX's parent in the same fashion. As indicated in Fig. 5d, NBOXI's great grandparent has a null box list and, therefore, does not contribute in this case. It can be noted that every box in the mesh has contributed to NBOXI. In general, it can be seen that the box list concept allows information to be passed down through successive levels in an orderly and efficient fashion.




a) Example of random distribution




b) Example of spiral distribution

**Fig. 6 Typical distributions used in optimization of $N_c$.**

### Generation of Taylor Series Coefficients

As mentioned previously, the Taylor series can be used only in cases where the source list box associated with a field box NBOXI is a list 2, 3, or 4 box. Information from the $A$ coefficients is used in cases where NBOXI has list 2 or 3 boxes, whereas information from the $B$ coefficients is used in cases where NBOXI has list 2 or 4 boxes.

### *A* Coefficients

The $A_{n-m,m}$ or $A'_{n-m,m}$ coefficients defined in Eqs. (5) and (7) are used to calculate the $B_{n-m,m}$ coefficient when the source box is a list 2 box. Also, for list 3 boxes, the $A$ coefficients are used directly in the calculation of field variables. The $A$ coefficients are calculated by first using Eqs. (5) to generate values for $A_{n-m,m}$ for all childless boxes. The centers of expansion for the smallest (highest level) child boxes are then shifted to their parent's centers using Eqs. (7). The contributions from the child boxes to the parent box are added together, producing a set of $A$ coefficients for the parent box. This process is repeated for decreasing levels (larger boxes) until the $A$ coefficients are calculated for all parent and childless boxes at all levels. This process may be thought of as the process by which information concerning source domain expansions is promulgated from smaller domains into larger domains.

### *B* Coefficients

The $B$ coefficients are only used for list 2 and 4 boxes. For list 2 boxes, the $B_{n-m,m}$ coefficients can be calculated from Eqs. (9). For list 4 boxes, the contribution from each vortex must be considered separately since the source domain is not well separated from the center of the field box NBOXI. The following equation is used to calculate the contribution from a group of $I$ vortices in a list 4 box to the $B$ coefficients in NBOXI:

$$B_{n-m,m} = \sum_{i=1}^{I} \frac{1}{2} \frac{\Gamma_i}{(n-m)!m!} \frac{\partial^n F(x - x_i, r, b_i)}{\partial x^{n-m}\, \partial r^m} \tag{18}$$

The center of the field box is at $x,r$, whereas the $i$th vortex center is at $x_i$, $b_i$. Therefore, the $B$ coefficients are calculated by first using Eqs. (9) for list 2 boxes and Eq. (18) for list 4 boxes to generate values for $B_{n-m,m}$ for all boxes. Next, the centers of expansion for the largest (lowest level) parent boxes that have list 2 or 4 boxes associated with them are shifted to their children's centers using Eqs. (11). The contribution from the parent box to the child box is added together with the list 2 and 4 contributions associated with the child box itself. This produces a set of $B$ coefficients for the child box. This process is repeated for increasing levels (smaller boxes) until the $B$ coefficients are calculated for all parent and childless boxes at all levels. This process may be thought of as the process by which information concerning field domain expansions is promulgated from larger domains into smaller domains.

### Calculation of Field Variables

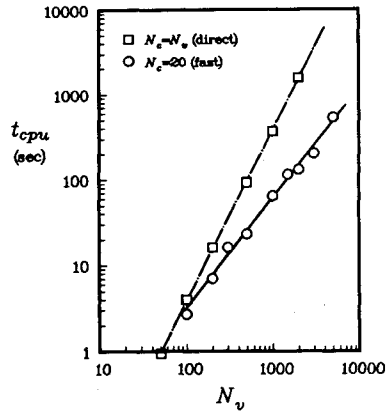
The field variables $\psi_p$, $u_p$, and $v_p$ at a point $P$ are calculated according to the type of list box containing the source information. For list 1 boxes, the field variable at point $P$ must be calculated by considering the influence of each individual vortex in the source box on the point $P$. For list 2 and 4 source boxes, a Taylor series about the center of the field box is used. This series uses the $B$ coefficients. For list 3 source boxes, a series containing the $A$ coefficients is used. Although the source box information is contained in the $A$ coefficients, the partial derivatives of $F$ must be recalculated for each field point.

The field variables $\psi_p$, $u_p$, and $v_p$ at a point $P$ for list 1 source boxes are calculated from

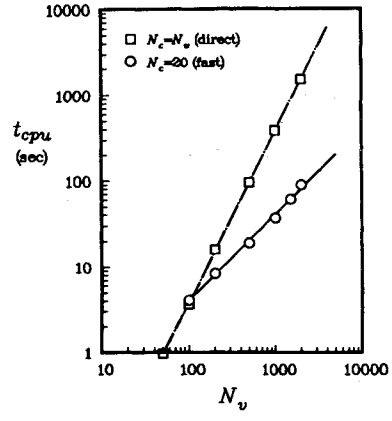
$$\psi_p = \sum_{i=1}^{I} \frac{1}{2} \Gamma_i F(x_p - x_i, r_p, b_i) \tag{19a}$$

$$u_p = \frac{1}{r_p} \sum_{i=1}^{I} \frac{1}{2} \Gamma_i \frac{\partial F(x_p - x_i, r_p, b_i)}{\partial r} \tag{19b}$$

$$v_p = -\frac{1}{r_p} \sum_{i=1}^{I} \frac{1}{2} \Gamma_i \frac{\partial F(x_p - x_i, r_p, b_i)}{\partial x} \tag{19c}$$




a) Random distributions




b) Spiral distributions

**Fig. 7 CPU time comparisons between direct and fast solvers.**

Here, the field-point location $P$ is given by $x_p$ and $r_p$, and the location of the $i$th vortex in the list 1 box is given by $x_i$ and $b_i$. The strength of the $i$th vortex is $\Gamma_i$, and the total number of vortices in the list 1 box is $I$.

The field variables at a point $P$ for list 2 and 4 source boxes are calculated by

$$\psi_p = \sum_{n=0}^{N} \sum_{m=0}^{n} B_{n-m,m} x_f^{n-m} r_f^m \tag{20a}$$

$$u_p = \frac{1}{r_p} \sum_{n=1}^{N} \sum_{m=1}^{n} m B_{n-m,m} x_f^{n-m} r_f^{m-1} \tag{20b}$$

$$v_p = -\frac{1}{r_p} \sum_{n=1}^{N} \sum_{m=0}^{n-1} (n-m) B_{n-m,m} x_f^{n-m-1} r_f^m \tag{20c}$$

Here, the distance from the field box center to the field point $P$ is given by the axial and radial coordinates $x_f$ and $r_f$. It should also be noted that the $B_{n-m,m}$ terms are the $B$ coefficients associated with the field box containing point $P$, which have been obtained according to the method of the previous section.

The field variables at a point $P$ for list 3 source boxes are calculated by

$$\psi_p = \sum_{n=0}^{N} \sum_{m=0}^{n} A_{n-m,m} \frac{\partial^n F(x_p - x_i, r_p, b_i)}{\partial x^{n-m}\, \partial b^m} \tag{21a}$$

$$u_p = \frac{1}{r_p} \sum_{n=0}^{N} \sum_{m=0}^{n} A_{n-m,m} \frac{\partial^{n+1} F(x_p - x_i, r_p, b_i)}{\partial x^{n-m}\, \partial r\, \partial b^m} \tag{21b}$$

$$v_p = -\frac{1}{r_p} \sum_{n=0}^{N} \sum_{m=0}^{n} A_{n-m,m} \frac{\partial^{n+1} F(x_p - x_i, r_p, b_i)}{\partial x^{n-m+1}\, \partial b^m} \tag{21c}$$

Here, the field-point location $P$ is given by $x_p$ and $r_p$, and the location of the $i$th list 3 source box is given by $x_i$ and $b_i$. The $A_{n-m,m}$ terms are the $A$ coefficients associated with the $i$th list 3 source box.

The contribution to a field variable at point $P$ from list 1 and 3 boxes is obtained by adding the contributions from all list 1 and 3 boxes together. The contribution from all list 2 and 4 boxes is inherent in the $B$ coefficients, which are associated with the field box itself. This contribution is added to the contributions from all list 1 and 3 source boxes.

## 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 vortices in the field, the distribution of vortices in the field, the number of terms used in the Taylor series, and the maximum number of vortices allowed in any childless box in the mesh. A limited parametric study was conducted and is described in some detail in Ref. 3. Based on the parametric study, it was found that reasonable accuracy could be obtained if one includes all of the terms through the fifth order in the Taylor series expansions ($N = 5$). Furthermore, if one limits the maximum number of vortices in any childless box to 20 ($N_c = 20$), then $t_{\text{cpu}}$ will be near the minimum in all cases. It should be noted that all of the benchmark tests reported in the following sections were run on a VAX 8820. In order to compare CPU times on different machines, runs were made on a VAX 8600 and on two different VAX 8650s. A uniform distribution of 1024 vortices provided the test case. Values of $N_c = 20$ and $N = 5$ were used. The CPU times for the VAX 8820 and the two VAX 8650s were virtually identical ($t_{\text{cpu}} = 50 \pm 0.9$ s), whereas the CPU time for the VAX 8600 was about 37% longer.

### Error Associated with Truncation

A measure of the error incurred by truncation of the Taylor series after the $N$th-order terms can be obtained for the velocity potential and total velocity fields by means of the following equations:

$$E_\psi = \left[ \frac{\sum_{i=1}^{N_v} (\psi_i - \psi_{ei})^2}{\sum_{i=1}^{N_v} \psi_{ei}^2} \right]^{1/2} \tag{22a}$$

$$E_V = \left[ \frac{\sum_{i=1}^{N_v} (V_i - V_{ei})^2}{\sum_{i=1}^{N_v} V_{ei}^2} \right]^{1/2} \tag{22b}$$

Here, $\psi_i$ and $V_i$ are the velocity potential and total velocity at point $i$, respectively, as calculated from the truncated series. The quantities $\psi_{ei}$ and $V_{ei}$ are the velocity potential and total velocity at point $i$, respectively, as calculated exactly using the direct method. As indicated, these error calculations are obtained from summations over all the $N_v$ vortices in the flow. Assuming that single precision accuracy means that any number stored in the computer will be accurate through six significant digits, then single precision roundoff errors will produce values of $E_\psi$ and $E_V$ up to about $1 \times 10^{-5}$. Therefore, errors calculated from Eqs. (22) that are equal to or less than $1 \times 10^{-5}$ imply that the machine roundoff errors are possibly as large as the truncation errors. For such cases, one should not continue to add terms to the Taylor series expansions for added accuracy unless the machine precision is also increased. Initial work using a uniform distribution of 1024 vortices indicated that a value of $N = 5$ will yield values of $E_\psi$ that are of the same order of magnitude as single precision roundoff errors.

### Optimum Number of Vortices in a Box

In order to optimize the maximum number of vortices $N_c$ allowed in any childless box with respect to $t_{\text{cpu}}$, random and spiral type distributions were used. Examples of these distributions are shown in Figs. 6. The cases in Figs. 6 represent two extreme distributions with vortices being highly localized in the spiral distribution, especially near its center. Optimization runs were made for random distributions with 50, 100, 200, 300, 500, 1000, and 2000 vortices. All of the vortices were chosen to be of equal strength. Vortices in the random distributions were not allowed to be any closer to the box border than 0.003 times the length of a side of the box or no closer than 0.1. Optimization runs were made for spiral distributions with 100, 200, 500, 1000, and 2000 vortices. Vortices of equal strength were placed on the spiral curve at equal angular intervals. Plots of $t_{\text{cpu}}$ vs $N_c$ were made for the 12 random and spiral distributions. Examination of these plots revealed that, if one chooses $N_c = 20$, a near-optimum (minimum) value of $t_{\text{cpu}}$ will be obtained for all cases. It is presently assumed that this optimization will be valid for most other distributions.

Comparisons of $t_{\text{cpu}}$ using the direct simulation vs using the fast solver with $N_c = 20$ and $N = 5$ for the random and spiral distributions are shown in Figs. 7. As can be seen from these figures, there is no advantage in using the fast solver for distributions that have less than about 100 vortices. For situations where there are more than 100 vortices, one can write equations for the straight lines plotted through the data. For the random distribution shown in Fig. 7a, the CPU time for the direct simulation can be expressed by

$$t_{\text{cpu}} = 3.85 \times 10^{-4} N_v^2 \tag{23}$$

On the other hand, the CPU time for the fast solver used on the random distribution can expressed by

$$t_{\text{cpu}} = 8.90 \times 10^{-3} N_v^{1.28} \tag{24}$$

Thus, the fast solver will run in about 3% of the time required for the direct simulation for the case where there are 10,000 vortices in the flow. For the spiral distribution whose CPU times are shown in Fig. 7b, the CPU time for the direct simulation is given by

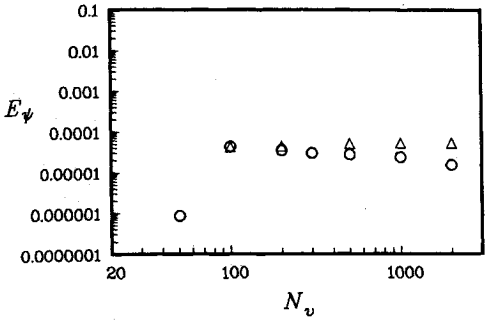
$$t_{\text{cpu}} = 3.85 \times 10^{-4} N_v^2 \tag{25}$$

As would be expected, the CPU times required for direct simulations of either the random or spiral distributions are the same. The CPU time for the fast solver used on the spiral distributions can be expressed by

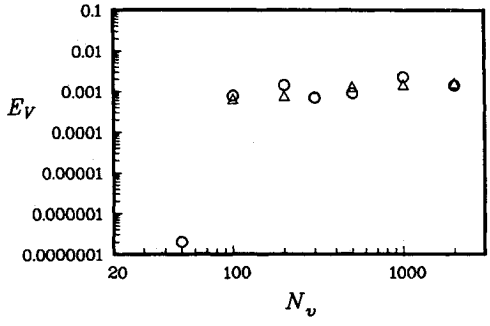
$$t_{\text{cpu}} = 4.08 \times 10^{-2} N_v \tag{26}$$

In this case, the fast solver will run in about 1% of the time required for the direct simulation when there are 10,000 vortices in the flow. The tendency for the exponent on $N_v$ to be greater for uniform distributions can also be observed in the data of Carrier et al.[1] Their data yield an exponent of 1.16 for their uniform distribution and 1.08 for their curve distribution.

At this point, it is appropriate to check the truncation errors produced using the fast solver on the random and spiral distributions to see if they are about as one might expect from preliminary work. Truncation errors are plotted in Figs. 8 for the random and spiral distributions vs the total number of vortices $N_v$. As can be seen from these figures, the orders of magnitude of these errors are relatively independent of the number of total vortices $N_v$ in the distributions. One exception is the error associated with $N_v = 50$. For $N_v = 50$, the errors are very small due to the fact that the simulation is essentially a direct simulation since there will be at most four boxes




**a) Stream-function errors: ○ random, △ spiral**




**b) Total velocity errors: ○ random, △ spiral**

**Fig. 8 Truncation errors for random and spiral distributions.**

produced by the mesh generator for a random distribution with $N_c = 20$. Excluding the data for $N_c = 50$, the truncation error for the stream function $E_\psi$ is on the order of $5 \times 10^{-5}$, whereas the error in total velocity $E_V$ is on the order of $1 \times 10^{-3}$. These errors are roughly of the same order of magnitude as those found for the uniform distribution used in the preliminary work.

## Self-Induction

All of the velocity calculations that have been made thus far have ignored the self-induced velocity associated with each vortex ring. Some very interesting discussions concerning the motion and subsequent distortion of vortex cores may be found in a recent work by Shariff et al.[6] The present work will be limited to obtaining the self-induced velocity at the center of impulse for a circular- or square-core ring vortex. A general discussion concerning the motion of vortex rings with cores is given by Lamb.[2] The cores themselves can be simulated using a large number of discrete vortices. For a collection of $i$ vortices, a centroid based on the impulse of the vortices is first obtained. The impulse for the system of vortices is given by

$$I = \pi\rho \sum_i \Gamma_i b_i^2 \tag{27}$$

The time rate of change of impulse represents a force that is zero for a collection of free vortices such as would be encountered in a wake flow. Assuming that the vortices are free, the impulse given by Eq. (27) is a constant. The $x$ and $r$ coordinates of the center of impulse $(x_o,r_o)$ are given by

$$x_o = \sum_i \Gamma_i b_i^2 x_i \Big/ \sum_i \Gamma_i b_i^2 \tag{28a}$$

$$r_o^2 = \sum_i \Gamma_i b_i^2 \Big/ \sum_i \Gamma_i \tag{28b}$$

A single vortex (with the strength of the sum of the $i$ vortices) has the same impulse as the collection of $i$ vortices when placed at $r_o$. The value of $x_o$ represents the axial centroid of the impulse for the collection of $i$ vortices. The motion of the center of impulse $(x_o,r_o)$ can be obtained by differentiating the expressions in Eqs. (28) with respect to time $t$. Since the impulse is constant, $r_o$ is a constant and the radial velocity $V_o$ of the center of impulse is equal to zero. On the other hand, the axial velocity $U_o$ of the center of impulse is not, in general, zero and is given by

$$U_o = \sum_i \Gamma_i b_i (b_i u_i + 2x_i v_i) \Big/ \sum_i \Gamma_i b_i^2 \tag{29}$$

It can be shown that, if the core is represented by a sufficient number of vortices, the contribution of the $i$th vortex to $u_i$ is negligible.

### Square Cores

Simulations of vortex rings with square cores were first run in order to develop an understanding of the number of discrete vortices required to properly represent the core region. It was assumed that the vorticity in the core was uniform. Cases were run for eight values of $N_v$ and three values of $b/s$, where $s$ is equal to one half of the distance along a side of the square core and $b$ is the radius to the center of the core. Results from this series of simulations are shown in Fig. 9. As can be seen from this figure, the self-induced velocity becomes essentially constant for $N_v > 100$. For example, the induction velocity increases by only 1–2% between $N_v = 100$ and 1000.

Next, a set of simulations were run in order to obtain an equation that can be used to calculate the self-induced velocity for a vortex ring with various ring radius to core size ratios $b/s$. It was first assumed that the self-induced velocity for a vortex ring with a square core can be written in the following form:

$$U_o = (\Gamma/4\pi b)[\mathcal{M}(8b/s) - C_s] \tag{30}$$

where $C_s$ is some function of $b/s$. Simulations were run for 10 different values of $b/s$ by placing 1089 vortices uniformly in the square cores. Equation (30) was solved for $C_s$ using values of $U_o$ obtained from Eq. (29). A very good fit to the data for $C_s$ is given by

$$C_s = 0.423 + 1.064 \exp[-0.593(b/s)] \tag{31}$$

In Fig. 10, the simulated values of $U_o b/\Gamma$ are plotted along with a curve (shown as a chain dot line) obtained by using Eq. (31) in Eq. (30). The fit between the correlation and simulation data is seen to be quite good.

The center of impulse $(x_o,r_o)$ of a square core was obtained from Eqs. (28) applied to a 1089 vortex simulation. Since the vorticity distribution about the center of the core is symmetrical, $x_o$ is equal to the axial location of the core center. The radial location of the center of impulse is shown in Fig. 11. The following equation, which is shown as the chain dot curve, fits the data reasonably well:

$$r_o/b = 1.0 + 0.586 \exp[-1.33(b/s)] \tag{32}$$

### Circular Cores

Relationships for the circular-core vortex rings were obtained in a similar manner by simulating the core with a set of 973 uniformly spaced vortices. An example of the vortex placement can be seen in Fig. 4a. Simulations were first run in order to obtain an equation that can be used to calculate the self-induced velocity for a vortex ring with various ring radius to core radius ratios $b/a$. Here, $a$ is the core radius.

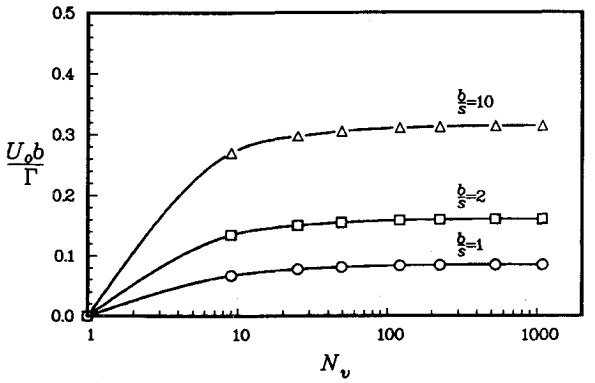



**Fig. 9 Self-induced velocity simulations of square core vortex rings.**

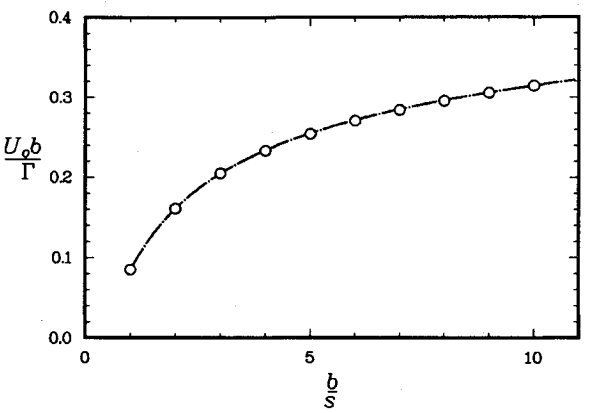



**Fig. 10 Self-induced velocities of square core vortex rings.**

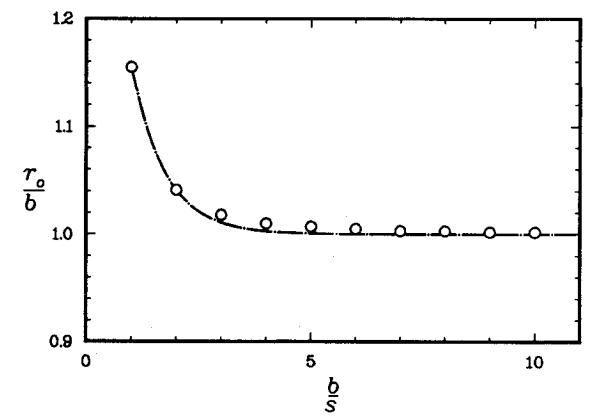



**Fig. 11 Radial center of impulse of square core vortex rings.**

The self-induced velocity for a vortex ring with a circular core was assumed to be of the following form:

$$U_o = (\Gamma/4\pi b)[\ln(8b/a) - C_a] \tag{33}$$

where $C_a$ is some function of $b/a$. Simulations were run for 10 different values of $b/a$. Equation (33) was solved for $C_a$ using values of $U_o$ obtained from Eq. (29). A good fit to the data for $C_a$ is given by

$$C_a = 0.250 + 1.064 \exp[-0.669(b/a)] \tag{34}$$

It is interesting to note that an estimate for $C_a$ could have been obtained from Eqs. (31) and (30) by assuming an equivalence between the core areas $A_c$ for a square and circular core (i.e., $A_c = 4s^2 = \pi a^2$). This assumption yields an equation identical to Eq. (34), except the constant term is equal to 0.302 instead of 0.250. Therefore, if $U_o b/\Gamma$ for the square and circular cores were plotted on the same plot vs $b/\sqrt{A_c}$, the data for the square core would be above that of the circular core data by only 0.004 (i.e., $0.052/4\pi$).

The center of impulse $(x_o, r_o)$ of a circular core was obtained from Eqs. (28) applied to the 973 vortex simulation. Since the vorticity distribution about the center of the core is symmetrical, $x_o$ is equal to the axial location of the core center. The following equation, which fits the simulation data for the center of the impulse very well, was obtained by making the substitution $4s^2 = \pi a^2$ in Eq. (32):

$$r_o/b = 1.0 + 0.586 \exp[-1.50(b/a)] \tag{35}$$

## Summary

Source- and field-point Taylor series expansions have been developed for axisymmetric ring vortices. These series are calculated using associated Legendre functions in conjunction with recursion relationships derived from stream-function considerations. Relationships that allow the source and or field expansion points to be translated within their domains of convergence were also developed.

A Fortran computer code RSOLV has been written to generate the adaptive mesh, box lists, and Taylor series expansions associated with the fast solution technique. The code also executes the fast solution technique to calculate the stream function and the axial and radial velocity components at points in the flowfield.

Test cases have been run to optimize the RSOLV code and to benchmark the truncation errors and CPU time savings associated with the method. The Taylor series were truncated after fifth-order terms, and the maximum number of vortices allowed in a childless box was set at 20. Truncation errors for the stream function and total velocity field were on the order of $5 \times 10^{-5}$ and $1 \times 10^{-3}$, respectively. For 100 vortices in the field, there was virtually no CPU time savings with the fast solver. For 10,000 vortices in the flow, the fast solver obtained solutions in about 1–3% of the time required for the direct solution technique.

Simulations for vortices with square and circular cores were run in order to obtain expressions for the self-induced velocities of such vortices. In each case, the vorticity was assumed to be uniform in the core. Correlations for the nondimensional self-induced velocity as a function of the ring radius nondimensionalized by the square root of the core area were found to be almost identical for the square and circular core vortices. Use of these formulas allows one to correctly convect discretized patches of vorticity in the flowfield.

## Acknowledgment

This work was performed at Sandia National Laboratories and supported by the U.S. Department of Energy under Contract DE-AC04-76DP00789.

## References

[1]Carrier, J., Greengard, L., and Rokhlin, V., "A Fast Adaptive Multipole Algorithm For Particle Simulations," *SIAM Journal on Scientific and Statistical Computing*, Vol. 9, No. 4, 1988, pp. 669–686.

[2]Lamb, H., *Hydrodynamics*, 6th ed., Dover, New York, 1932, p. 239.

[3]Strickland, J. H., and Amos, D. E., "A Fast Solver for Systems of Axisymmetric Ring Vortices," Sandia National Lab., Rept. SAND90-1925, Albuquerque, NM, Sept. 1990.

[4]Gradshteyn, I. S., and Ryzhik, I. M., *Table of Integrals, Series, and Products*, Academic, London, 1980.

[5]Lowan, A. N., *Tables of Associated Legendre Functions*, Columbia Univ. Press, New York, 1945.

[6]Shariff, K., Leonard, A., and Ferziger, J. H., "Dynamics of a Class of Vortex Rings," NASA TM-102257, Dec. 1989.