# Continuous-Field Image-Correlation Velocimetry and its Application to Unsteady Flow Over an Airfoil

Galen Gerald Gornowicz

*Graduate Aeronautical Laboratories*
*California Institute of Technology*
*Pasadena, CA 91125*

16 April 1997

Thesis submitted in partial fulfillment of the degree requirements
of Aeronautical Engineer

## Acknowledgements

Many people contributed their special talents and effort to the work contained in this thesis. Without their devoted help, none of this work would have been possible. Below is a list of people I am forever indebted to for this reason:

- Paul Dimotakis, Professor of Aeronautics and Applied Physics. Even when the "going got tough," Paul always had words of encouragement, his understanding and faith went way beyond the bounds of a normal advisor. Paul never lost confidence in the science, or in me, even when I had.

- Pavel Svitek, GALCIT technical staff. Almost all of the construction work for the models was completed by Pavel. In addition, he was always ready to lend a hand in anything that needed to be done, as well as helping to bail me out of a few of the predicaments I managed to get into.

- Daniel Lang, Staff Engineer. Dan's absolute genius with anything electronic, from complex circuits to computer networks, is truly awe inspiring. The experiments performed for this thesis would have suffered greatly were it not for the superior equipment designed and fabricated by Dan, and his many late night technical support sessions. Words simply can't express Dan's success in building an entire, extremely sophisticated, data acquisition system that never failed me.

- David Laidlaw, Postdoctoral Scholar, Computer Science. David's breadth of algorithmic knowledge, combined with his experience and willingness to sit down and understand the problems with my code, proved an invaluable resource. In particular he had almost a clairvoyant perception of the nature, and location, of subtle bugs.

- Phillip Tokumaru, Senior Scientist, AeroVironment, Inc. Phillip's preceeding work in Image Correlation Velocimetry was the foundation for the current version of the code, including preliminary use of B-splines. His recommendation of pitfalls to avoid in this research were accurate, if not well heeded.

- Dominique Fourguette, Research Scientist, Rice Systems. As the in-house laser expert, Dominique aided in the resurrection of many finicky beam profiles. Her practical experience with optics was much appreciated when designing experiments.

This work was made possible by the Air Force Office of Scientific Research Grant Nos. F49620–93–1–0338 and F49620–94–1–0353, whose generous support I would gratefully like to acknowledge.

## Abstract

Continuous-field Image Correlation Velocimetry (ICV) is an extension to the ICV technique of Tokumaru & Dimotakis (1995). The method determines the optical flow in sequences of images, and relies on a convected Lagrangian marker, *e.g.*, a conserved scalar field, or particles, *etc.*. The method has been applied to several simulated-flow test cases and results are presented for the error of the method, with and without noise added to the correlated test-images. The results of further tests are reported, for two laboratory flows, a NACA–0012 airfoil at high angle of attack, and a transverse jet in a coflowing stream.

# Table of Contents

## 1. Introduction

The computational analysis of motion from a sequence of images has been a continuing focus of researchers for the better part of two decades. Contributions have been made from a wide range of disciplines, resulting in a variety of methods. In the context of fluid mechanics, the measurement is especially valuable, potentially providing velocity *field* information, over the imaged domain. A recent overview by Dracos & Gruen (1997) of various two- and three-dimensional implementations, as applied to fluid mechanics, dubbed "videogrammetric methods in velocimetry" by these authors, provides a comprehensive discussion and bibliographical reference list.

In their review of the various so-called optical-flow methods, Barron *et al.* (1994) classify the techniques into four categories, two of which are supersets of techniques commonly used in experimental flow velocimetry. One category identified by Barron is region-based matching, which rely on matching of sub-regions between images. Velocimetry methods that fall into this general category have approached the problem from a variety of bases. Various incarnations of PIV/DPIV, for example, effectively solve the matching problem using spatial cross-correlations of discrete windows of image pairs (*e.g.*, Adrian 1991; Willert & Gharib 1991, Sholl & Savas 1997), or by actually calculating the spatial correlation function (Huang 1994). Other techniques from photogrammetry define systems of equations using a least-squares formulation of the matching criterion for small regions of pixels (Ackermann 1983, Gruen 1985, Maas 1993). Anandan (1989) uses a similar approach, but implementing a coarse-to-fine procedural hierarchy by decomposing the optical features by length scales.

A very similar implementation to the previous ICV method that relied on a variational approach, described by Tokumaru & Dimotakis (1995), is by Szeliski & Shum (1996). These authors employed a multiresolution-spline representation of the displacement field between image pairs, as also adopted in the implementation to be described below. The development by us of this idea is surprisingly similar to the Szelinski & Shum approach, that was developed in a different context. Variational methods have been used to infer displacement fields in a variety

of contexts, as done, for example, by Zhou *et al.* (1995), who employed a multiresolution representation of the three-dimensional displacement field in the interior of a cylindrical asphalt/aggregate core, assuming a volume-preserving (divergence-free) displacement field.

Another general category identified by Barron *et al.* (1994) are the so-called differential techniques, pioneered by Horn & Schunck (1981). These methods calculate the components of the scalar transport equation and use additional constraints to remove ambiguities. Recent work has investigated and compared the required additional constraint(s) to the scalar transport equation proposed by various researchers after Horn & Schunck (*e.g.*, Willick & Yang 1991). Strong proponents of the application of this technique for fluid velocimetry have been Dahm *et al.* (1991, 1992). More recently, a variational approach was offered by Su & Dahm (1995) and Dahm *et al.* (1996). Pearlstein & Carpenter (1995), however, noted that the method of Dahm and collaborators suffers from a local ambiguity problem in that the local velocity field is only defined in the direction of the imaged-scalar gradient. Pearlstein & Carpenter proposed to mitigate this ambiguity problem by simultaneously tracking of multiple scalar fields. Additionally, these methods must differentiate the image data to deduce the convecting velocity field, rendering them rather susceptible to the inevitable image noise.

While it is appreciated that no one method is best for every situation, the general methodology for ICV can be shown to reduce, to leading order, to the scalar-transport equation, for the case of negligible diffusion. The continuous-field ICV methodology to be described below draws from proven techniques for fluid velocimetry and utilizes approaches found useful in machine-vision contexts. It also has the ability to impose known boundary conditions and a hierarchy of spatial resolutions, as by Szeliski & Shum (1996). This results in a robust procedure for the recovery of motion from images of scalar markers transported by a fluid, subject to certain assumptions, as will be discussed below.

## 2. Continuous-field ICV method

The ICV procedure seeks the displacement field, $\boldsymbol{\xi}(\mathbf{x})$, such that the region in the neighborhood of $\mathbf{x}$, in the image $I_1(\mathbf{x})$, at time $t_1$, is best mapped into the region $\mathbf{x}+\boldsymbol{\xi}$ in the next image, $I_2(\mathbf{x})$, at $t_2 = t_1 + \tau$, *i.e.*, such that,

$$I_1(\mathbf{x}) \;\mapsto\; I_2(\mathbf{x}+\boldsymbol{\xi}) \;. \tag{1}$$

Quantitatively, the procedure seeks the displacement field, $\boldsymbol{\xi}(\mathbf{x})$, that minimizes the square of the difference of the two images, integrated over the correlation domain, $\Omega$, *i.e.*, a cost function given by,

$$\mathcal{J}\{\boldsymbol{\xi}\} \;=\; \int_{\Omega} [\, I_2(\mathbf{x}+\boldsymbol{\xi}) - I_1(\mathbf{x})\,]^2 \; \mathrm{d}\Omega(\mathbf{x}) \;\rightarrow\; \min \;. \tag{2}$$

This cost function itself does not guarantee either a unique or smooth solution. Such attributes depend on the functional representation of $\boldsymbol{\xi}$ and are addressed in Sec. 2.2. Furthermore, in the implementation to be described below, the correlation domain may, but need not, extend to the full imaged domain (less a small boundary region that would allow both $\mathbf{x}$ and $\mathbf{x}+\boldsymbol{\xi}$ to remain within the two images).

If the time differences, $\tau = t_2 - t_1$, between the two images is small, in some appropriate sense, one can Taylor-expand the displaced-image field at $t_2$, *i.e.*,

$$I_2(\mathbf{x}+\boldsymbol{\xi}, t_1+\tau) \;=\; I_1(\mathbf{x},t_1) \;+\; \tau\,\frac{\partial}{\partial t} I_1(\mathbf{x},t_1) \;+\; \boldsymbol{\xi}\cdot\frac{\partial}{\partial \mathbf{x}} I_1(\mathbf{x},t_1) \;+\; \text{H.O.T.'s} \;.$$

where the higher-order terms would be $\mathcal{O}(\tau^2)$, $\mathcal{O}(\boldsymbol{\xi}^2)$, or $\mathcal{O}(\tau\boldsymbol{\xi})$. A mapping (displacement) field (Eq. 1), *i.e.*, one that produces,

$$I_2\,[\mathbf{x}+\boldsymbol{\xi}(\mathbf{x}), t_1+\tau] \;\simeq\; I_1(\mathbf{x},t_1) \;,$$

is seen to be equivalent to the requirement that,

$$\tau\,\frac{\partial}{\partial t} I_1(\mathbf{x},t_1) \;+\; \boldsymbol{\xi}\cdot\frac{\partial}{\partial \mathbf{x}} I_1(\mathbf{x},t_1) \;\simeq\; 0 \;,$$

again, to leading order in the space/time displacements, *i.e.*,

$$\frac{\partial}{\partial t} I_1 \;+\; \mathbf{u}\cdot\frac{\partial}{\partial \mathbf{x}} I_1 \;\simeq\; 0 \;, \qquad \text{for,} \quad \mathbf{u} \;=\; \frac{1}{\tau}\,\boldsymbol{\xi} \;. \tag{3a,b}$$

Equation 3 is the standard scalar-transport equation, provided diffusive effects are negligible, which typically translates to an upper limit on the time interval, $\tau$, between the image pair. Since scalar diffusivity is essentially fixed by the choice of the fluid, the time interval must be chosen such that diffusion is negligible (*cf.* Tokumaru & Dimotakis 1995).

If the time interval, $\tau$, between the image pair is not small, such that the actual convection velocity varies (temporally) within this interval, the ICV algorithm will still produce a mapping displacement field, $\boldsymbol{\xi}(\mathbf{x})$, that may be regarded as a time integration of some effective Lagrangian velocity, $\mathbf{u}[\mathbf{x}+\boldsymbol{\xi}(\mathbf{x};t),t]$, at intermediate times, $t$. That is,

$$\boldsymbol{\xi}(\mathbf{x};t_1+\tau) \;=\; \int_{t_1}^{t_1+\tau} \mathbf{u}\left[\mathbf{x}+\boldsymbol{\xi}(\mathbf{x};t),t\right]\,\mathrm{d}t \;. \tag{4}$$

The assignment of the inferred velocity to a midpoint in space and time is then seen to be correct to second order in the image-pair time interval, $\tau$, *i.e.*,

$$\mathbf{u} \;=\; \frac{1}{\tau}\,\boldsymbol{\xi}(\mathbf{x})\,\left[\mathbf{x}+\boldsymbol{\xi}(\mathbf{x})/2,\,t_1+\tau/2\right] \;+\; \mathcal{O}\left(\tau^2\right) \;. \tag{5}$$

The ICV method does not actually require images closely-spaced in time to produce a successful mapping (displacement) field.

To compute an optimal mapping field, the ICV method relies on a parametric representation of the displacement field, $\boldsymbol{\xi}(\mathbf{x})$. In several refined DPIV implementations (*e.g.*, Huang 1994, Sholl & Savas 1997), as well as in the previous ICV implementation (Tokumaru & Dimotakis 1995), local Taylor expansions of the displacement field were employed, to various orders, *i.e.*,

$$\begin{aligned}\boldsymbol{\xi}(\mathbf{x}) \;=\; \boldsymbol{\xi}(\mathbf{x}_\mathrm{c}) \;&+\; (\mathbf{x}-\mathbf{x}_\mathrm{c})_i\,\alpha_{\mathrm{c},i} \\ &+\; \frac{1}{2}\,(\mathbf{x}-\mathbf{x}_\mathrm{c})_i\,(\mathbf{x}-\mathbf{x}_\mathrm{c})_j\,\beta_{\mathrm{c},ij} \\ &+\; \frac{1}{3!}\,(\mathbf{x}-\mathbf{x}_\mathrm{c})_i\,(\mathbf{x}-\mathbf{x}_\mathrm{c})_j\,(\mathbf{x}-\mathbf{x}_\mathrm{c})_k\,\gamma_{\mathrm{c},ijk} \\ &+\; etc.\;,\end{aligned} \tag{6}$$

where $\mathbf{x}_\mathrm{c}$ is any expansion point with the parameters, $\alpha_{\mathrm{c},i}$, $\beta_{\mathrm{c},ij}$, $\gamma_{\mathrm{c},ijk}$, *etc.* determined by the DPIV or ICV solution.

In the ICV implementation of Tokumaru & Dimotakis (1995), the cost function that was minimized included terms that increased the cost function with the (square of the) amplitude of any discontinuities of the displacement field and its derivatives at the boundaries of the array of Taylor-expansion regions around the selected control points, $\mathbf{x}_c$. As a consequence, much of the built-in flexibility in describing spatial variations of the displacement (velocity) field was lost, with degrees of freedom gained from the Taylor-expansion coefficients in Eq. 6, in effect, expended to minimize discontinuities of the velocity field and of its derivatives at the Taylor-expansion region boundaries.

To mitigate this difficulty, the present ICV implementation relies on a displacement field that possesses the required, $\mathcal{C}^n$, continuity properties by construction (where the order of continuity, $n$, is chosen appropriately as described below). The remaining (true) degrees of freedom are utilized to minimize the cost function, $\mathcal{J}\{\boldsymbol{\xi}\}$, with no added (smoothing) terms in the integrand. Velocity- and vorticity-field solutions of the Navier-Stokes equations are continuous, with continuous derivatives to all orders, *i.e.*, are $\mathcal{C}^\infty$. In the present implementation, which was limited to two-dimensional fields, a $\mathcal{C}^2$ displacement (velocity) field was employed, *i.e.*, possessing continuous second derivatives, corresponding to inferred vorticity fields that possessed continuous first derivatives. This was achieved by representing the displacement field in terms of B-splines with appropriate basis functions, whose control parameters, $\mathbf{q}_{ij}^{(r,R)} \in \Re^2$, then provided the parametric description of the displacement field, *i.e.*,

$$\boldsymbol{\xi}(\mathbf{x}) \;=\; \boldsymbol{\xi}\left[\mathbf{x};\, \mathbf{q}_{ij}^{(r,R)}\right] \;, \tag{7}$$

as will be described below.

With the solution space of the minimization problem (Eq. 2) restricted in this fashion, the cost *functional*, $\mathcal{J}\{\boldsymbol{\xi}\}$, becomes a *function* of the control parameters, *i.e.*,

$$\mathcal{J}\{\boldsymbol{\xi}\} \;\rightarrow\; \mathcal{J}\left[\mathbf{q}_{ij}^{(r,R)}\right] \;, \tag{8}$$

possessing a minimum where derivatives of $\mathcal{J}$, with respect to each $\mathbf{q}_{ij}^{(r,R)}$, vanish. This allows a global minimization over the (selected) image-correlation domain to be sought, using an iterative, multi-dimensional, conjugate-gradient method over

all parameter values, with a suitable initial guess, as will be described below.

## 2.1 Multi-resolution B-splines

Almost any interesting fluid flow will entail a wide range of spatial scales in its velocity field. Flows near a body will possess relatively-thin boundary layers, where the velocity will increase from the body velocity at the solid wall, to near freestream velocity values, in a relatively short distance as compared to length of the entire flow field. In regions outside the boundary layer, velocity-field length scales might be large, with the field itself relatively featureless. Flows which can generally be classified as turbulent, however, are likely to possess the entire range of scales throughout the turbulent-flow regions.

In representing the velocity field, it is desirable to employ a representation that has sufficient degrees of freedom, but no more. Determining how many degrees of freedom are required is itself a challenging problem and, in the current implementation, is a user-defined parameter. As shown in the test-case section, the accuracy of the method decreases when the solution space is allowed more degrees of freedom than the local flow field warrants and attempts to fit the (high wavenumber) noise, chasing image and other noise in the data.

In the ICV implementation described here, a multi-resolution B-spline representation of the velocity field was employed to address these considerations. With a multi-resolution construction, high-resolution basis functions can be used near the boundary of an object, for example, or in any region of the flow that warrants their use. In other regions of the flow, only the lower-resolution basis functions need to be activated, as appropriate.

The complete B-spline representation is, effectively, a summation of sets of different-resolution basis functions. The parameters of the lowest-resolution basis functions carry the information for the overall offsets of the velocity field, while the parameters of higher-resolution basis functions contribute refinements, as suggested by Forsey & Bartels (1988, 1995). The actual implementation of this concept, however, has been modified for its use in ICV, as will be described below. In

general, the Forsey & Bartels (1988) method deals with surface construction in the context of $3-D$ modelling. The concept of an induced frame of reference for the higher resolution surfaces is useful for intuitive interactive modelling, but is not important in an automated method such as that employed in ICV. Futhermore, the concept of individual patches of higher resolution is foregone in favor of a unified, multi-resolution basis set.

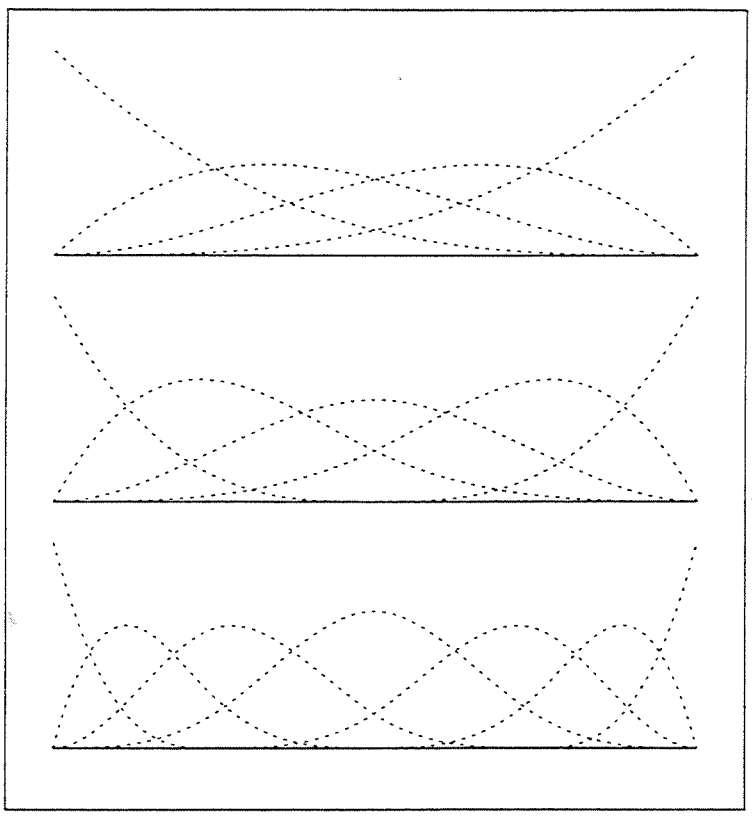

FIG. 1 One-dimensional, three-level, multi-resolution cubic B-spline basis function set. Top: lowest spatial resolution, one knot interval; middle: two-knot intervals; bottom: four-knot intervals.

The multi-resolution B-splines were implemented as follows. A complete set of basis functions is generated for the entire flow domain, starting with the coarsest resolution, up through the highest (finer) resolution. The number of knot intervals in

a given dimension always doubles for each increase in resolution. A one-dimensional example employing cubic B-spline basis functions is depicted in Fig. 1, showing three levels of the multi-resolution hierarchy.

The final spline, $\mathbf{f}(\mathbf{x})$, is (conceptually) a summation of the different resolution level splines, *i.e.*,

$$\mathbf{f}(\mathbf{x}) \;=\; \sum_{r=1}^{R} \mathbf{f}^{(r)}(\mathbf{x}) \;, \tag{9a}$$

where the superscript denotes resolution level and each individual level, $\mathbf{f}^{(r)}$, is evaluated as,

$$\mathbf{f}^{(r)}(\mathbf{x}) \;=\; \sum_{i,j} \mathbf{q}_{ij}^{(r)}\, B_i^{(r)}(x)\, B_j^{(r)}(y) \;. \tag{9b}$$

All parameters of lower-resolution splines are first transformed to those corresponding to the highest-resolution level. The general procedure has been referred to as "knot refinement" by Piegl & Tiller (1995). Let $r$ denote a particular resolution level and parameters $\mathbf{q}_{ij}^{(r,r)}$ define a spline at that resolution. The spline can also be represented with a higher number of parameters, in particular, corresponding to the highest-resolution level, $R$. Parameters $\mathbf{q}_{ij}^{(r,R)}$ result in an $R$-resolution, B-spline representation that matches the $r$-resolution representation, *i.e.*,

$$\mathbf{f}^{(r)}(\mathbf{x}) \;=\; \sum_{i,j} \mathbf{q}_{ij}^{(r)}\, B_i^{(r)}(x)\, B_j^{(r)}(y) \;=\; \sum_{i,j} \mathbf{q}_{ij}^{(r,R)}\, B_i^{(R)}(x)\, B_j^{(R)}(y) \;, \tag{10}$$

where the intermediate equation is employed only once in computing the mapping of the coefficients and included here as a conceptual aid. The coefficients $\mathbf{q}_{ij}^{(R,R)}$ can be seen to correspond to the highest-resolution contribution to the total representation.

As implemented here, where knot grids of lower-resolution representations are derived as spatial binary subdivisions of higher-resolution knot grids, lower-level knots are subsets of higher-level knots and Eq. 10 can be solved exactly. This scheme permits the resulting spline evaluation to employ a single (the highest-resolution) basis-function set throughout and can be written as,

$$\mathbf{f}(\mathbf{x}) \;=\; \sum_{i,j} \left[ \sum_{r=1}^{R} \mathbf{q}_{ij}^{(r,R)} \right] B_i^{(R)}(x)\, B_j^{(R)}(y) \;=\; \sum_{i,j} \mathbf{q}_{ij}\, B_i^{(R)}(x)\, B_j^{(R)}(y) \;. \tag{11}$$

The transformation of all lower-resolution parameters that map the B-spline representation to the highest-resolution representation is computed ahead of time, allowing fast repetitive evaluations of the solution vector field, as required for the efficient, iterative solution of the optimization algorithm. The desired variable resolution across the solution domain is then implemented setting $\mathbf{q}_{ij}^{(r,R)} = 0$, for $r_{\max} < r \leq R$, with $r_{\max}$ selected, as appropriate, in each region.

In the case of the unsteady-flow around an accelerating airfoil, for example, after the irrotational regions (that, generally, lack high velocity gradients) have been captured by the lower-resolution parameters, the higher-resolution parameters, near the boundary layer, wake, and in any shed structures, are enabled by increasing $r_{\max}$ locally, as necessary. The final effective knot grid for this example is depicted in Fig. 19. Plotted white lines connect knot points at each resolution where full support of the basis functions has been enabled, at that resolution.

## 2.2 ICV algorithm implementation

The ICV implementation described here is comprised of a sequence of iterative, algorithmic steps: image-data preparation, image-correlation domain definition, cross-correlation displacement-field initialization, and conjugate-gradient displacement-field optimization.

The procedure starts by further processing individual data images, after background removal, illumination normalization, *etc.*, for shot-to-shot intensity variations of the illuminating laser sheet. A geometry file is generated next, which locates the correlation domain, $\Omega$, within the image domain. An initial hierarchy of the B-spline resolution knot grid is specified and any excluded regions from the correlation domain (*e.g.*, laser shadows, imaging occlusions, *etc.*) are also identified. The outer boundary of the correlation domain, $\Omega$, is specified as a polyline ($n$-sided polygon). Inner boundaries can also be accommodated, allowing correlation domains to be defined that are not simply-connected, as necessary. These boundaries are defined on the first of the two images (for each pair). The algorithm permits the extension of the domain outside $\Omega(t_1)$, as required by the mapping field, $\boldsymbol{\xi}(\mathbf{x})$, provided it isn't

trying to map from a point within an excluded region. For example, if only a small portion of an image pair is being correlated, the algorithm is free to look anywhere in the second image for a match to the correlation region of the first image, except in excluded regions that may have been identified, as described above.

The next step is to initialize the solution at the coarsest resolution level; usually, one spline patch. The initialization is performed by cross-correlating spatially-local windows, using Fourier techniques, as in DPIV analyses (*e.g.*, Adrian 1991, Willert & Gharib 1991). The results of these correlations initialize the mapping vector field, $\boldsymbol{\xi}(\mathbf{x})$. No equations are solved to improve the displacement vector field, at this stage, with results from each cross-corelation window representing an average of the displacement of the two imaged fields in each window. Windows are then centered at the peak of each B-spline basis function and the cross-correlation results are used to determine the corresponding B-spline control parameters, $\mathbf{q}_{ij}^{(r)}$, at the resolution level $r$. Near edges, or where the window will not fit within the image, the window is placed as close as possible to the desired location. Velocities returned by the cross-correlation procedure that exceed a maximum threshold are discarded and replaced by the average of the values determined for neighboring regions.

The initialization, $\boldsymbol{\xi}^{(0)}$, of the B-spline representation for $\boldsymbol{\xi}$, allows Eq. 1 to be invoked, producing an initial mapped version of the second image, *i.e.*, $I_2(\mathbf{x}+\boldsymbol{\xi}^{(0)})$, that is "closer" to the first image, $I_1(\mathbf{x})$. Further cross-correlations are run between $I_1(\mathbf{x})$ and $I_2(\mathbf{x}+\boldsymbol{\xi})$ to produce subsequent estimates, $\boldsymbol{\xi}^{(n)}$. Had the best possible mapping been found in the first pass, the result of later correlations would be a null vector field. This, however, is seldom the case.

A similar process for determining the displacement field in DPIV, but without FFT's, is outlined in Huang (1994) and termed, "Particle Image Distortion". A fast version, termed, "Lagrangian Particle Tracking", was introduced by Sholl & Savas (1997). In these implementations, the deduced displacement field was specified in terms of local Taylor expansions, to first and second order, respectively.

The ICV initialization step starts with large cross-correlation windows (up to $256 \times 265$ pixels, or $1/4 \times 1/4$ the image domain, for example) to avoid spurious

correlations and to pick up any large displacements. This initialization step is particularly important, if there are displacements greater than 1/2 times the characteristic length of a continuous scalar used to mark the flow (equivalent to a Nyquist criterion). While large correlation windows tend to average out small features of the velocity field, they produce robust estimates of large, near-uniform displacements. Small-scale features of the velocity field are then determined in subsequent stages. This aspect is particularly important, in as much as the subsequent minimization stages may not correct for errors introduced at this stage and a local minimum of $\mathcal{J}(\boldsymbol{\xi})$ might be found instead.

Once large-scale displacements have been found with such windows, the size of the window is successively reduced by a factor of 2, cross-correlations are performed, and the corresponding B-spline parameters are computed to yield the next window-size estimates of the displacement (mapping) field. These successive halvings continue until a user-determined minimum window size is reached.

The cross-correlation initialization sequence does not attempt to solve the minimization equation (Eq. 2), although it does typically reduce the cost function, $\mathcal{J}$. The displacement field, $\boldsymbol{\xi}$, produced by the cross-correlation sequence is used to initialize an iterative minimization procedure. This procedure solves Eq. 2, within the solution sub-space spanned by the parametric B-spline representation of the displacement field, as described above, driving $\mathcal{J}$ to a minimum via a (multi-dimensional) conjugate-gradient scheme (Press *et al.* 1992).

Numerical evaluation of the $\mathcal{J}$ integral is fairly straight-forward. The continuous integral expressed in Eq. 2 is converted to a pixel-by-pixel summation, with $I_2$ reconstructed as $I_2(\mathbf{x} + \boldsymbol{\xi})$, using a 2-D Mitchell filter (Mitchell & Arun 1988).

The projection of the displacement field on the set of B-spline basis functions (*cf.* Eq. 7 and discussion in Sec. 2.1) converts the integral $\mathcal{J}$ to be minimized from a functional of $\boldsymbol{\xi}$ to a function of the finite number of B-spline control parameters (*cf.* Eq. 8), as noted above. The required minimization of $\mathcal{J}$ is now performed in a finite-dimensional space.

While no one minimization method is clearly better than others, good success

was realized with a multi-dimensional, gradient-based approach. The specific algorithm implemented here is the Polak-Ribiere variant of the conjugate-gradient algorithm (Press *et al.* 1992). Our implementation of this algorithm has modified termination criteria based not only on a decrease of the cost function but also on the magnitude of the gradient and parameter-space "distance" traveled by the optimization iteration step.

The conjugate-gradient algorithm requires gradient information (with respect to the B-spline parameters). The components of grad$\{\mathcal{J}\}$ were estimated using a centered, two-point, finite-difference scheme. Symbolically, for each component of $\mathbf{q}_{ij}$,

$$(\text{grad}\{\mathcal{J}\})_{q_{ij}} = \frac{1}{h}\left[\mathcal{J}(\ldots, q_{ij}+h/2, \ldots) - \mathcal{J}(\ldots, q_{ij}-h/2, \ldots)\right] , \qquad (12)$$

where $h$ is small in an appropriate sense.

As a result of the compact support of the basis functions associated with an individual $q_{ij}$, significant benefits, including mitigation of roundoff-errors, were realized. In particular, the entire integral (whole correlation domain) need not be calculated in Eq. 12, since a change in a particular parameter will only influence the local region where the associated basis function is non-zero (compact support).

The conjugate-gradient minimization is "local", *i.e.*, it cannot guarantee global minimization and will converge to the first minimum encountered. It was found that the coarse-to-fine cross-correlation initialization sequence was generally able to position the solution "close" enough to the global minimum. This allowed the subsequent conjugate-gradient minimization sequence to complete the multi-dimensional-space path and converge to a plausible global minimum, at least as ascertained by visual inspection, for the cases presented in this paper.

The algorithmic sequence benefits, at all stages, from visual inspection by the user, since the human eye is extremely adept at detecting motions. In particular, viewing an animation of $I_1(\mathbf{x})$, compared to $I_2(\mathbf{x}+\boldsymbol{\xi})$, one can typically assess whether it is not some local minimum that has been found. This assesment is very effective, in that a successful mapping will produce no apparent motion between the first and mapped second images. The only residual visual difference should

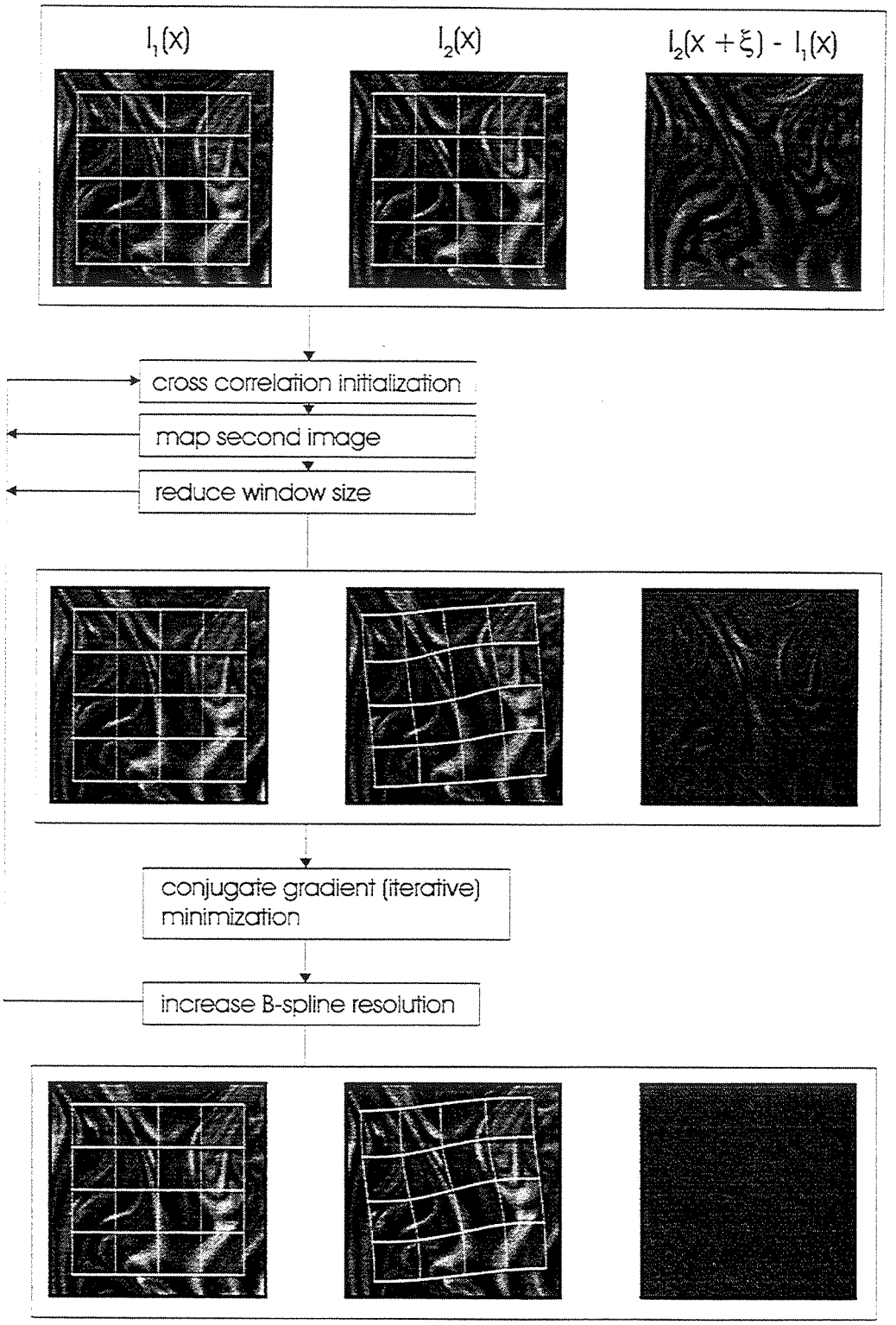


FIG. 2 ICV algorithmic sequence. Windows on the right depict gray-scale images of the (algebraic) difference between $I_1(\mathbf{x})$ and $I_2(\mathbf{x}+\boldsymbol{\xi})$, at each step. Uniform (half-scale) gray level denotes zero.

be from image-acquisition noise, consequences of representiation/resolution inadequacies, and in the present implementation, spurious consequencs of out-of-plane motion, for example.

Following convergence at the spatial-resolution hierarchy specified in the domain-definition initialization, the user has the option of revising the spatial resolution. This is done interactively by scripting a new geometry file. The process is then repeated, either anew, or retaining the initialization, or last-iteration solution, and confining the iterations to the optimization steps of the algorithm.

This implementation using multi-resolution B-spline representation is akin to multigrid methods generally employed for solving elliptic equations. A coarse approximation of the solution is calculated and then refined as the spatial resolution of the solution is allowed to increase.

The combined algorithmic sequence of cross-correlation steps, followed by the conjugate-gradient minimization ($\xi$-optimization) steps, is schematically depicted in Fig. 2.

The ellipticity of subsonic-flow equations results in a potentially strong dependence on boundary conditions. As a consequence, it is desirable to incorporate knowledge of the boundary conditions, as is feasible. The local representation (compact support) of the displacement field adopted in the present implementation localizes effects of errors at the boundary and they do not tend to propagate, as strongly, throughout the entire domain.

Boundary conditions present significant challenges due in part to irregular geometries and a lack of image data to correlate on one side of the boundary. In the case of a physical correlation domain with a straight boundary (along a coordinate direction), there is an elegant solution using the B-spline formulation implemented here. One can simply "turn off" control points which correspond to the basis functions yielding full support along the boundary, *i.e.*, the outer-most basis functions. An example is presented in Sec. 3.3.

This idea could be carried into the general case, where a curvilinear domain

is mapped into a Cartesian computational domain, where the boundary conditions can be applied, in a similar manner along a straight edge in the computational domain. This has not been implemented as yet, however, and in our accelerating-airfoil experimental test case, where such a curvilinear boundary was encountered, the no-slip boundary condition on the foil surface, for example, was imposed at discrete points along the airfoil.

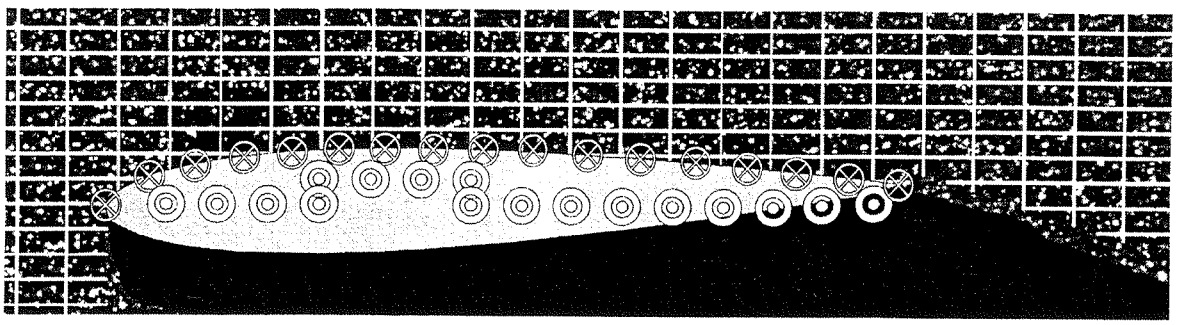

FIG. 3 Discrete constraint points and the set of parameters devoted to satisfying the chosen boundary condition.

The Cartesian grid upon which the correlation was performed presented a challenging problem in imposing a boundary condition on a curve, within the B-spline representation. The chosen method relied on user-identified, multiple, discrete constraint points (circles with x's in Fig. 3) within the correlation domain, on which the specified boundary condition was enforced. For the accelerating, NACA–0012 airfoil flow experiment described below (Sec. 4.1), the chosen constraint points were evenly spaced (highest-resolution knot-grid spacing) along the airfoil chord. At each chosen constraint point, the displacement field was prescribed to satisfy the no-slip condition, *i.e.*,

$$\xi_c(\mathbf{x}) \;=\; 0 \;.$$

The velocity field within the airfoil has no physical significance, even though the functional B-spline representation is still defined there. The parameters which define

the B-spline in the airfoil interior can then be used to enforce the no-slip boundary condition at the selected points on the airfoil surface. The position of the peaks of the basis functions used to enforce the boundary condition are denoted by concentric circles in Fig. 3. This is termed "constraint-based surface modification" in Piegl & Tiller (1995), although applied here to a vector field. The solution to the problem is solved once and stored for multiple applications of the constraints in the ICV algorithm.

A set of equations can be written for the dependance of the chosen B-spline parameters used to satisfy boundary conditions, as a function of the parameters that remain to be determined by the optimization equation. For each boundary constraint point, $(x_b, y_b)$, the displacement field is set to zero (*cf.* Eq. 11),

$$\mathbf{0} = \sum_{i,j} \mathbf{q}_{ij} B_i^{(R)}(x_b) B_j^{(R)}(y_b) \ , \tag{13a}$$

and the no-slip boundary condition, in this case, is imposed by the implicit equality (cancellation) on the boundary control points, *i.e.*,

$$\sum_{i,j:\,\text{constrained}} \mathbf{q}_{ij} B_i^{(R)}(x_b) B_j^{(R)}(y_b) = - \sum_{i,j:\,\text{optimized}} \mathbf{q}_{ij} B_i^{(R)}(x_b) B_j^{(R)}(y_b) \ . \tag{13b}$$

These equations can be written in matrix form, and solved once (influence matrix) during initialization of the program by singular-value decomposition, allowing the resulting dependence relations to be efficiently applied. The B-spline parameters for which this calculation was performed were those at the highest resolution, $R$, of the B-spline hierarchy. These constrained parameters are effectively taken out of the optimization equation, reducing the degrees of freedom for the problem. While this changes the dynamics of the multi-resolution implementation, the change only affects a small region (compact-support extent from the influenced points) around the imposed boundary.

Alternative boundary conditions can also be implemented in this fashion, corresponding to an *a priori* knowledge of irrotational in-flow on a boundary, for example, *etc.*

## 3. Simulated-flow test cases

The algorithmic sequence described above was tested on a set of simulated flows (displacement fields). Simulated flows corresponding to a Lamb-Oseen vortex and a parallel boundary-layer were employed. The robustness of the ICV-inferred displacement field with respect to image noise was also assessed by comparing the results as a function of additive noise, independently superimposed on each one of the two images that were processed in each case. The results of these test cases will be described below.

### 3.1 Lamb-Oseen vortex

The continuous-field ICV algorithm has been tested on a simulated model-flow field of a Lamb-Oseen vortex, with an added (vertical) freestream component, $\mathbf{u}_\infty = \widehat{\mathbf{y}}\, V$. The Lamb-Oseen vortex flow is an analytical solution for the temporal decay of a vortex filament (*e.g.*, Batchelor 1967, Saffman 1992),

$$\mathbf{u}(\mathbf{x},t) \;=\; \widehat{\mathbf{r}}\;\; V\,\cos\theta \;+\; \widehat{\boldsymbol{\theta}}\,\left\{\,\frac{\kappa}{2\pi r}\left[1-e^{-r^2/(4\nu t)}\right] \;+\; V\,\sin\theta\,\right\}\,. \qquad (14)$$

This field was used to convect the two-dimensional image in Fig. 4 into a simulated scalar image at two times, $t_1$ and $t_2 = t_1 + \tau$. The $(800 \times 800)$-pixel test image in Fig. 4 was formed using a fluorescent dye (kriegrocine) and a $(1 - 2\,\mathrm{mm})$-thick Nd:YAG laser sheet in water, recorded on a $1134 \times 486$ (physical) pixel, TI CCD-camera (Model TI MC–1134P), digitized at 12 bits/pixel, at 10 frames/s as dictated by the laser pulse-repetition frequency, and acquired on an in-house data-acquisition system.* A single image from that sequence was mapped (bilinear interpolation) from the (rectangular) physical-pixel grid onto a square-pixel grid for subsequent processing. The resulting square-pixel image is the one displayed in Fig. 4. The pair of ICV input images were produced by numerically-convecting this single image into the simulated scalar images at two distinct times as described above. The ICV algorithm was applied to a $(650 \times 650)$ correlation-domain, $\Omega$ in Eq. 2, subregion of the full images.

---

* Designed and fabricated by Dan Lang.

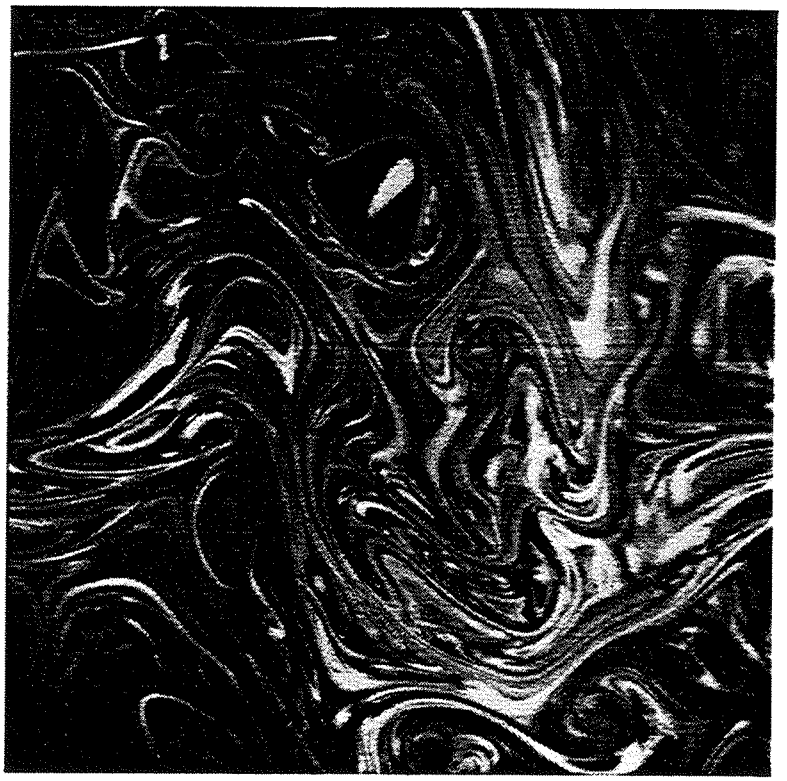

FIG. 4 Laser-induced fluorescence image, scaled to $800 \times 800$ pixels, use as source-image data for Lamb-Oseen vortex simulated test case.

The simulated flow field, defined in Eq. 14, has the Lamb-Oseen vortex centered in the image, with parameter values: $\kappa = 7000\,\text{pixels}^2/\text{frame}$ ("frame" here denotes "frame-time interval"), $\nu t = 1000\,\text{pixels}^2$, and $V = 1\,\text{pixel/frame}$. Figure 5 reproduces the surface plots of the analytical vertical component of the velocity field, $v(x, y)$, and out-of-plane component of the vorticity, $\omega_z = \omega(x, y)$.

Iterative refinements of the ICV solution for the Lamb-Oseen vortex test case are plotted in Figs. 6 through 9. For the test case presented here, one resolution level of the B-spline representation was enabled. The solution was represented with bicubic B-splines defined on a $(16 \times 16)$-grid of evenly-spaced knots, with collapsed knots at the boundaries. Figure 25 (Appendix A) depicts the set of basis functions used for both dimensions.

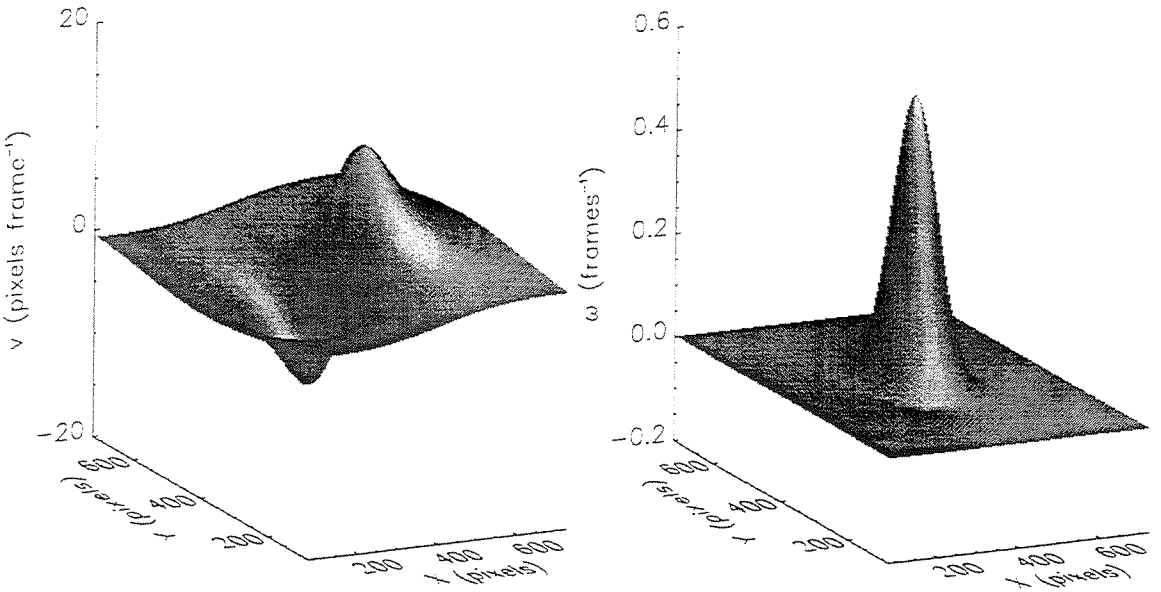


FIG. 5 Surface plot of the analytical vertical component of velocity, $v(x, y)$, and out-of-plane component of vorticity, $\omega(x, y)$.

The first three figures show the ICV solution after successive cross-correlation initializations. The surface plots of Fig. 6 show the solution after two iterations using $(128 \times 128)$-pixel cross-correlation windows. Figures 7 and 8 depict the result of successive refinements to $(64 \times 64)$-pixel and $(32 \times 32)$-pixel correlation windows, respectively, as described above. Considerable errors are evident, in both velocity and vorticity, when visually compared to the analytical fields in Fig. 5.

Starting with the initialized solution and following to the next step in the ICV algorithm, the mapping-error integral, $\mathcal{J}\{\boldsymbol{\xi}(\mathbf{x}; \mathbf{q}_{ij})\}$ of Eq. 2, was iteratively minimized, using the conjugate-gradient minimization algorithm, as discussed above. The final velocity- and vorticity-field results are presented in Fig. 9. A preliminary visual comparison with the analytical expression in Fig. 5 helps assess the level of

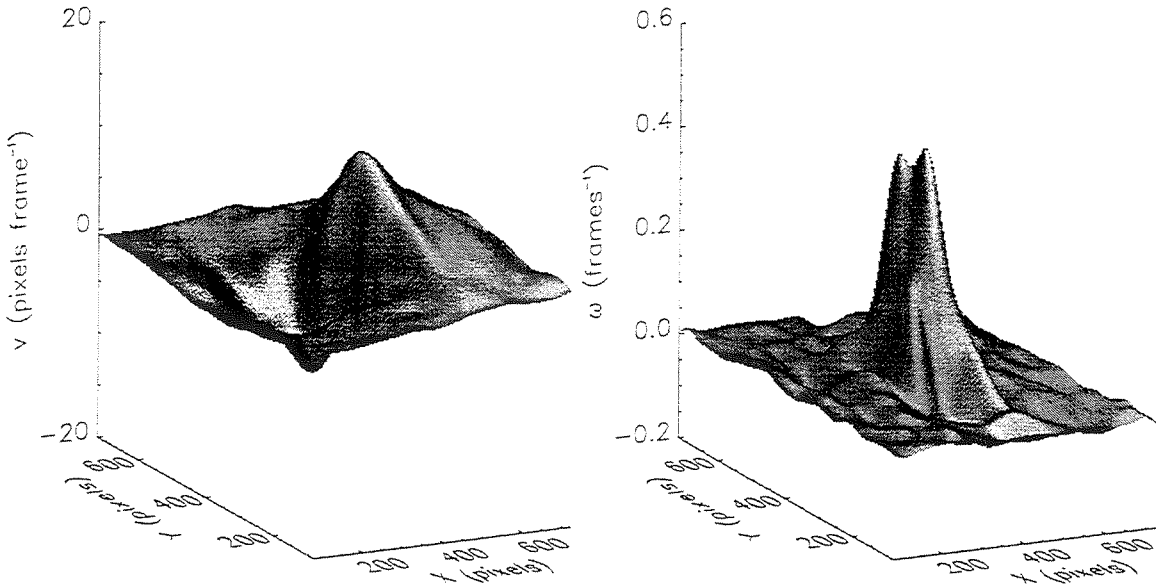


FIG. 6 Surface plot of the ICV solution for $v(x, y)$, and, $\omega(x, y)$, after initialization and refinement using 128 × 128-pixel cross-correlation windows.

accuracy attained at this step.

To assess the inferred velocity-field errors, the error surface, $\epsilon(x, y)$, for the experimentally-determined $v_{\rm exp}(x, y)$-velocity and $\omega_{\rm exp}(x, y)$-vorticity fields, *i.e.*,

$$\epsilon_v(x, y) \equiv \frac{v_{\rm exp}(x, y) - v_{\rm th}(x, y)}{\max |v_{\rm th}(x, y)|} \tag{15}$$

and

$$\epsilon_\omega(x, y) \equiv \frac{\omega_{\rm exp}(x, y) - \omega_{\rm th}(x, y)}{\max |\omega_{\rm th}(x, y)|} \ , \tag{16}$$

are plotted in Fig. 10, expressed as percentages, *i.e.*, $z_v = 100\,\epsilon_v(x, y)$, on the left, and $z_\omega = 100\,\epsilon_\omega(x, y)$, on the right. The error in the "experimental", ICV-deduced $v$-velocity is 0.4% (rms) over the whole field, with a maximum error of 7.7% on one

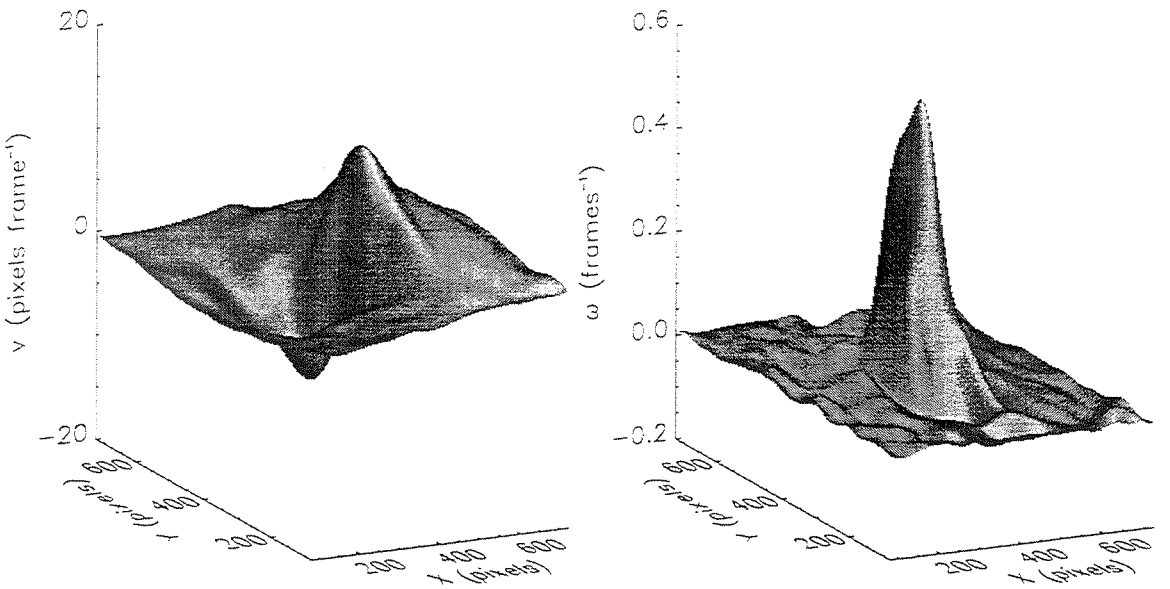


FIG. 7 Surface plot of the ICV solution for $v(x, y)$, and, $\omega(x, y)$, after refinements using 64 × 64-pixel cross correlation windows.

corner. The error in the deduced vorticity over the whole field is 0.6% (rms) with a maximum error of 13.5%, for this test case, at the same corner. Overall, the high-error regions are in the neighborhood of the image correlation-domain boundaries.

### 3.2 Effects of noise

Tests were also performed to assess the effects of image noise on the robustness of the deduced velocity fields. The same LIF-image data shown in Fig. 4 were again numerically convected with the same velocity field (Eq. 14). Each image in the pair was then independently degraded with additive noise, in a simple way, with amplitudes up to 15% rms. Specifically, a random-number generator was used to

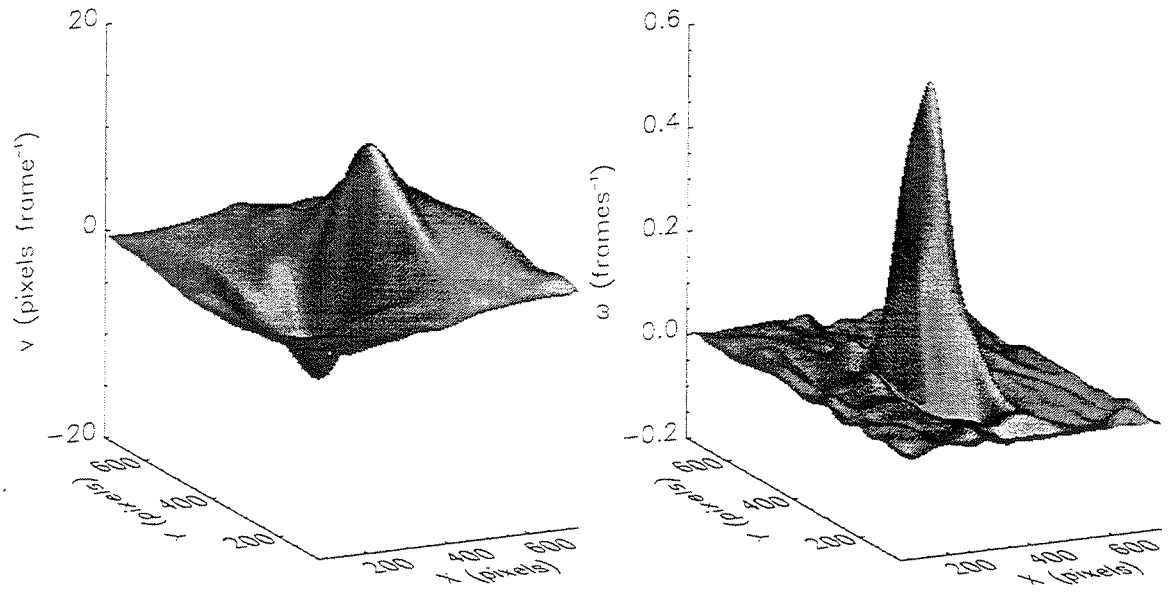


FIG. 8 Surface plot of the ICV solution for $v(x, y)$, and, $\omega(x, y)$, after refinements using $32 \times 32$-pixel cross correlation windows.

produce (a top-hat pdf of) noise, that was added pixel by pixel and was quantified by its rms magnitude. The magnitude of the noise was normalized by the rms of the image data, before the noise was added, and expressed as a percentage. Sample sub-regions, with and without additive noise, are shown in Fig. 11.

ICV results with the flow field of Sec. 3.1 were attained using bicubic B-splines with a $(16 \times 16)$ knot grid (patches). The discussion in this section considers two different-sized correlation areas, as a single level and as the highest multi-resolution level of a B-spline hierarchy, as well as the effect of noise degradation of the images. Knot grids of $16 \times 16$ and $32 \times 32$ were used in the B-spline representation of the displacement (velocity) fields. In test cases for which multi-resolution was enabled, all basis functions were allowed to contribute at all levels, in other words, every

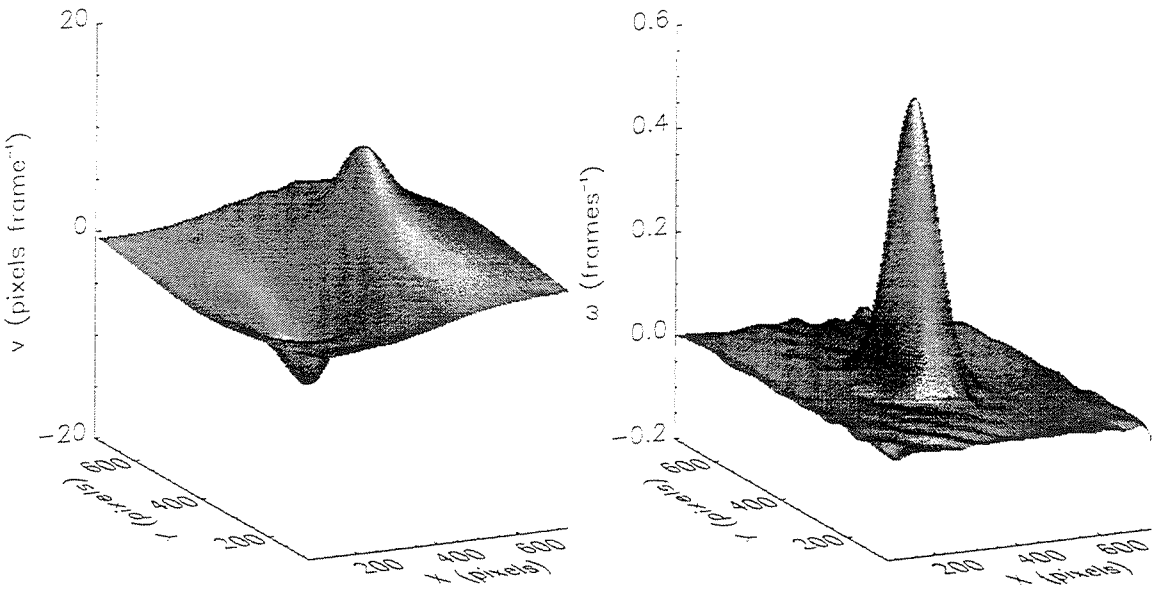


FIG. 9 Surface plot of the ICV solution for $v(x, y)$, and, $\omega(x, y)$, after conjugate-gradient optimization.

parameter, $\mathbf{q}_{ij}^{\tau}$, participated in the ICV correlation.

The number of knot regions is inversely proportional to the effective size of the individual correlation regions. The higher knot-number, *i.e.*, number of piecewise regions, the denser the spacing of basis functions for the same image area. The rms error of the ICV solutions for the vorticity, defined as in Eq. 16, for the various cases, is plotted as a function of added noise level, in Fig. 12.

As can be seen the ICV implementation has an improved ability to correlate noisy images when using the (spatially) wider basis functions of the $16 \times 16$ knot grid representation. The wider basis functions are less prone to erroneous correlations of low spatial-frequency noise components, acting as low-pass filters as a consequence

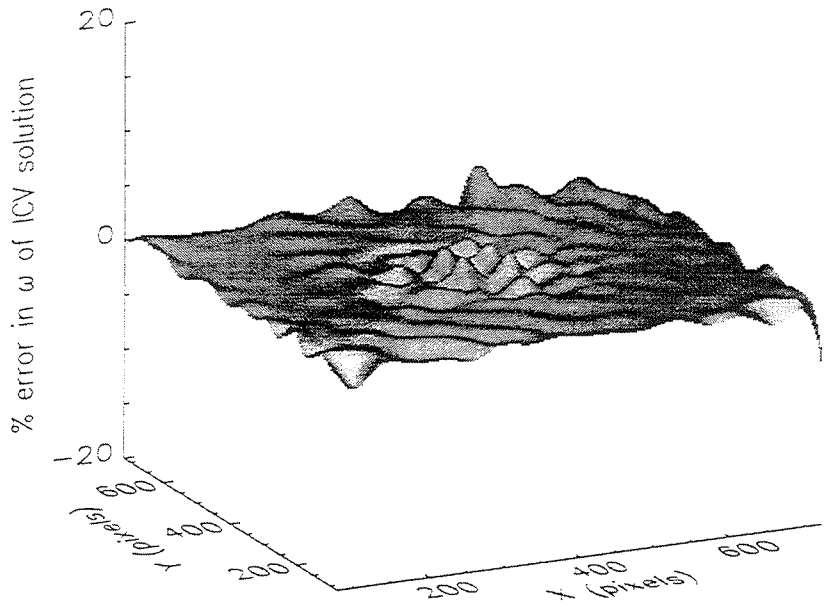


FIG. 10 Surface plot of percent error in ICV solution for the vertical velocity, $v(x, y)$, left, and out-of-plane vorticity $\omega(x, y)$, right, for the simulated Lamb-Oseen vortex flow (*cf.* Eq. 15 and related discussion).

of their effectively-larger image-correlation regions.

### 3.3 Boundary-layer flow

The ICV algorithm was also tested on the simulated flow field of a two-dimensional boundary layer, with a velocity profile approximated by a quarter-sine function, *i.e.*,

$$\mathbf{u}(\mathbf{x}, t) = \widehat{\mathbf{x}}\, U \begin{cases} \sin\left(\dfrac{\pi y}{2\delta}\right), & \text{for } y \leq \delta\,; \\ 1, & \text{otherwise}\,. \end{cases} \tag{17}$$

The image pairs were generated from (a portion of) the same scalar image (Fig. 4) in a similar fashion (with and without added noise) to the Lamb-Oseen vortex test

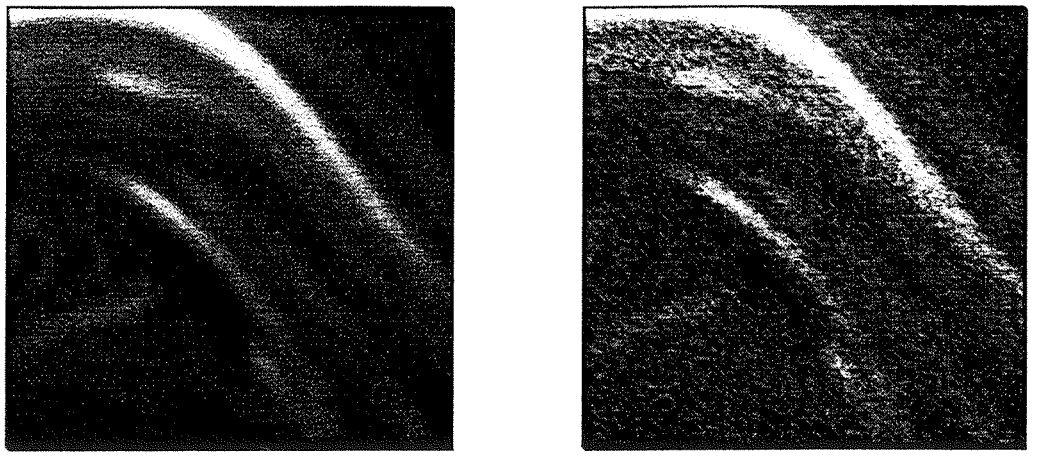

FIG. 11 Sub-region of LIF image data showing original data, left, and with 15% rms noise added, right (see text).

case. The results with a hierarchical, full-resolution B-spline representation were compared to those from a multi-resolution representation. A $(32 \times 32)$ reference grid is superimposed on the scalar-field image in Fig. 13 (left). The ICV solution of the displaced second image (per Eq. 17) was then used to derive the mapped grid depicted in Fig. 13 (right).

The ICV solution, using the hierarchical, full-resolution representation, is depicted in Fig. 14. As can be seen, ICV was able to capture the steep gradient near the (simulated) wall, as well as properly accommodate the no-slip boundary condition. In this case, because the flow boundary (wall) is along a B-spline coordinate direction, it was possible to impose it exactly. The rms error of this solution, computed over the whole image domain, is 0.3%, as normalized by the freestream value, $U$. The full-resolution representation performed better, with no noise added to the image pair, than the multi-resolution displacement-field representation (Fig. 15). However, with added noise, the multi-resolution representation was slightly better, employing larger freestream correlation regions, while still possessing sufficient degrees of freedom to accurately represent the velocity field. The resulting rms-error is plotted in Fig. 16, as a function of the added (rms) random noise.

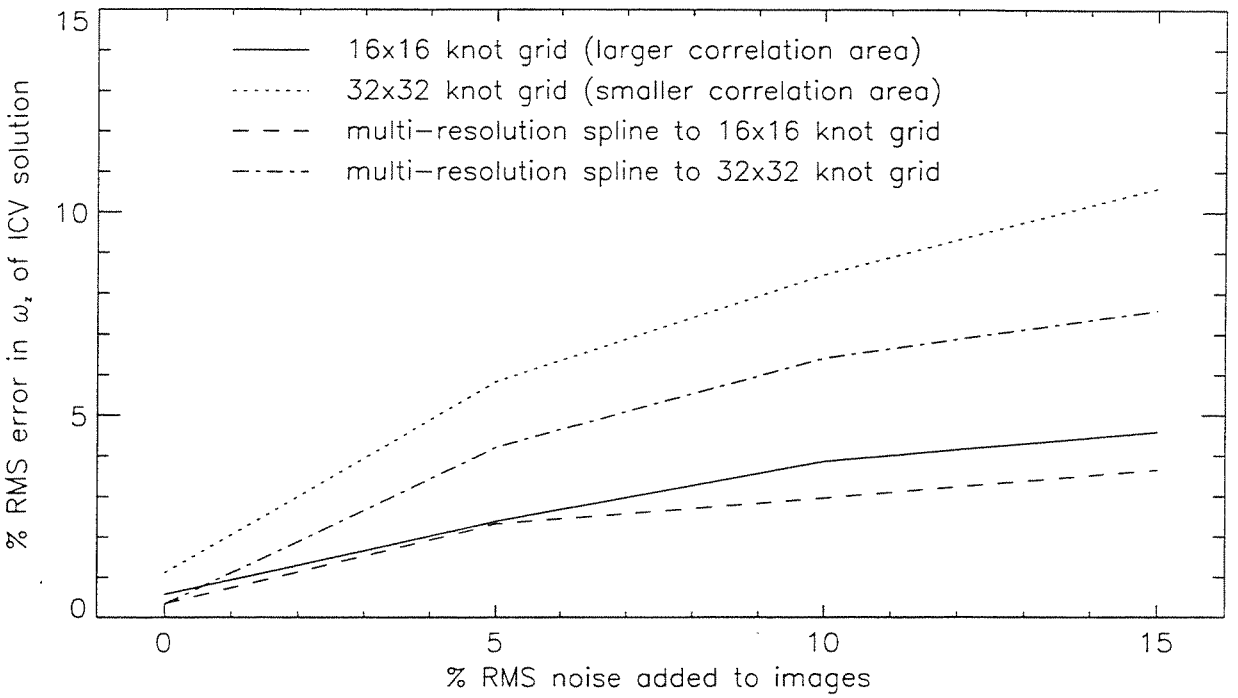


FIG. 12 Percent-rms error of vorticity, *vs.* added noise, for the simulated Lamb-Oseen vortex image pair, for two sizes of image-correlation regions (see text).

## 4. Laboratory-flow test cases

The ICV method was applied to two laboratory flows. The first was a two-dimensional flow over an accelerating airfoil at an angle of attack and utilized both particles and scalars as Lagrangian markers. The second was a three-dimensional flow generated by a transverse jet in a coflowing stream, utilizing the jet-fluid concentration field, as labeled with a fluorescent dye, as a Lagrangian marker. These two test and illustrate different issues in the ICV methodology and will be discussed below.

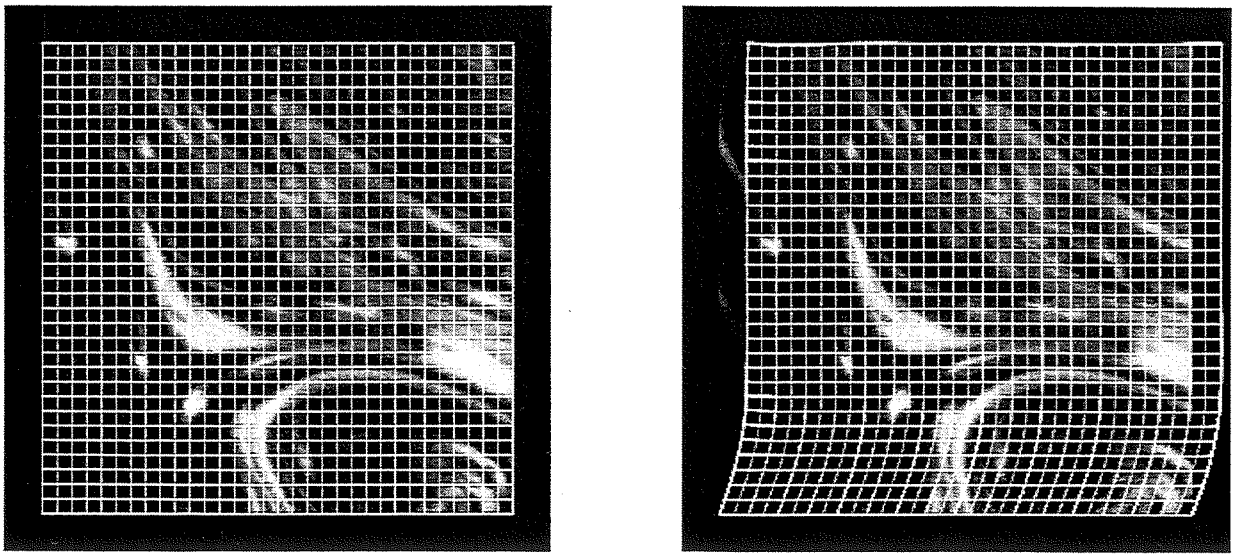

FIG. 13 Simulated boundary-layer flow. Hierarchical, full-resolution representation. Left: Reference grid on first image. Right: ICV solution superimposed on displaced scalar image.

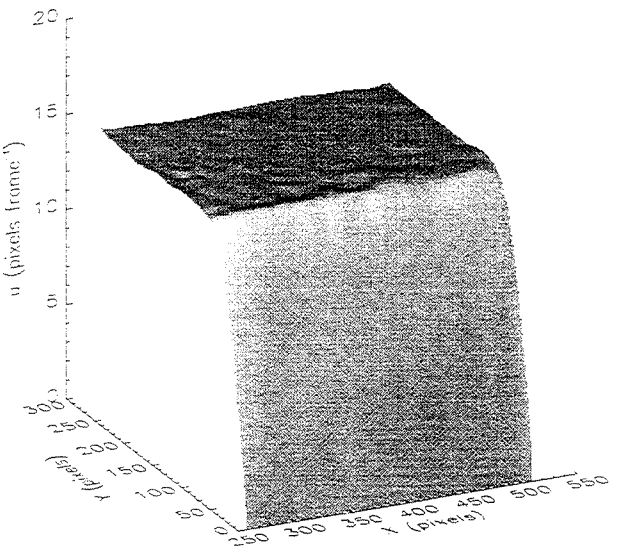


FIG. 14 Simulated boundary-layer flow. Hierarchical, full-resolution representation. ICV solution surface for velocity field.

### 4.1 Accelerating NACA–0012 airfoil

The experiment described here focused on the investigation of the unsteady, low-Reynolds number flow over a uniformly-accelerated, $a = 0.108\,\mathrm{cm/s^2}$, NACA–0012 airfoil in a water tow-tank. The airfoil had a chord of $c = 9.05\,\mathrm{cm}$, and was

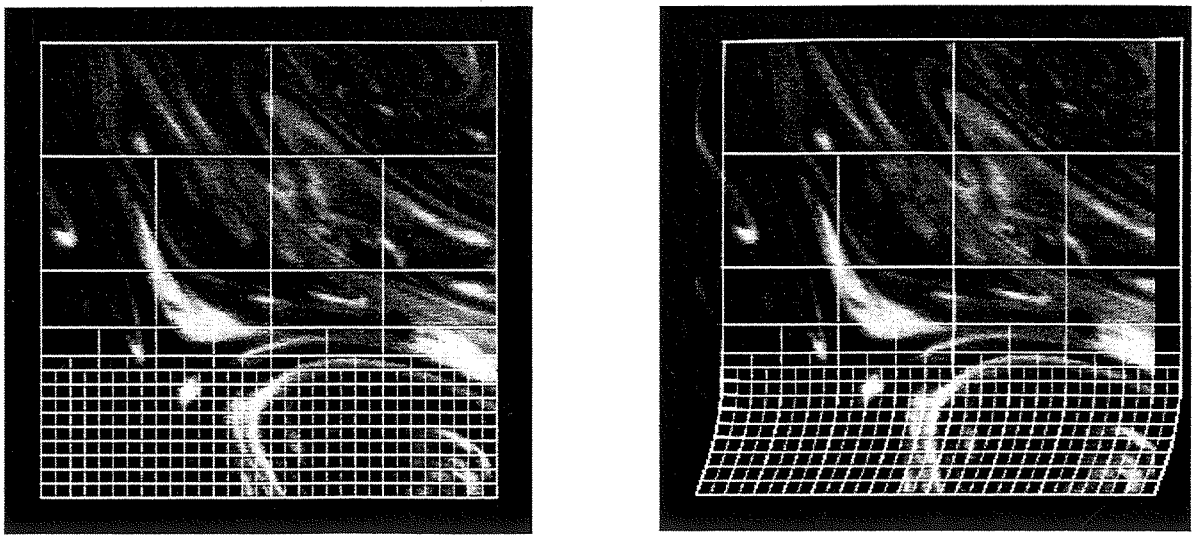

FIG. 15 Simulated boundary-layer flow. Hierarchical, multi-resolution representation. Left: Reference grid on first image. Right: ICV solution superimposed on displaced scalar image.

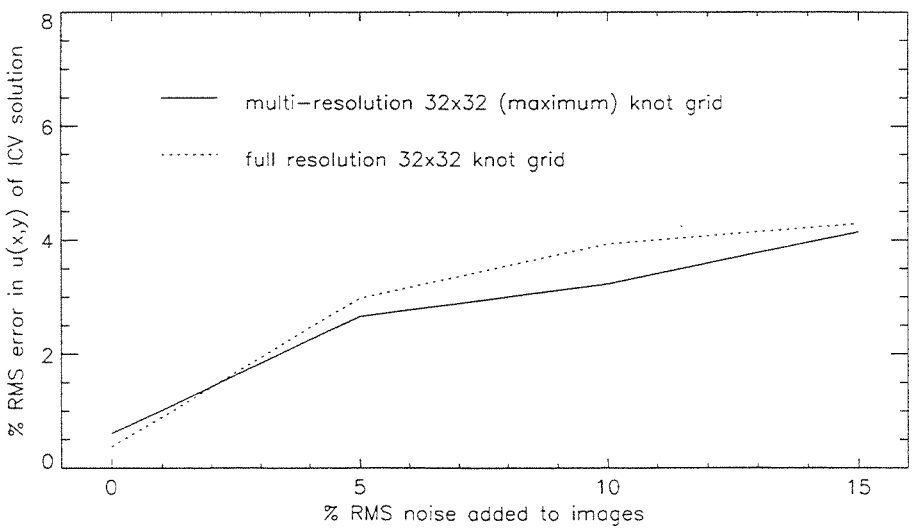


FIG. 16 ICV solution error (rms) for the simulated boundary-layer flow, as a function of added (rms) random noise.

mounted at a fixed angle of attack, $\alpha = 22.5°$. The water temperature of $T = 21.9°\mathrm{C}$ yielded a kinematic viscosity of $\nu = 1.05 \times 10^{-2}\,\mathrm{cm}^2/\mathrm{s}$.

The airfoil was attached to a computer-controlled, linear-traverse carriage system that allowed it to follow a prescribed motion profile in a water-filled tow

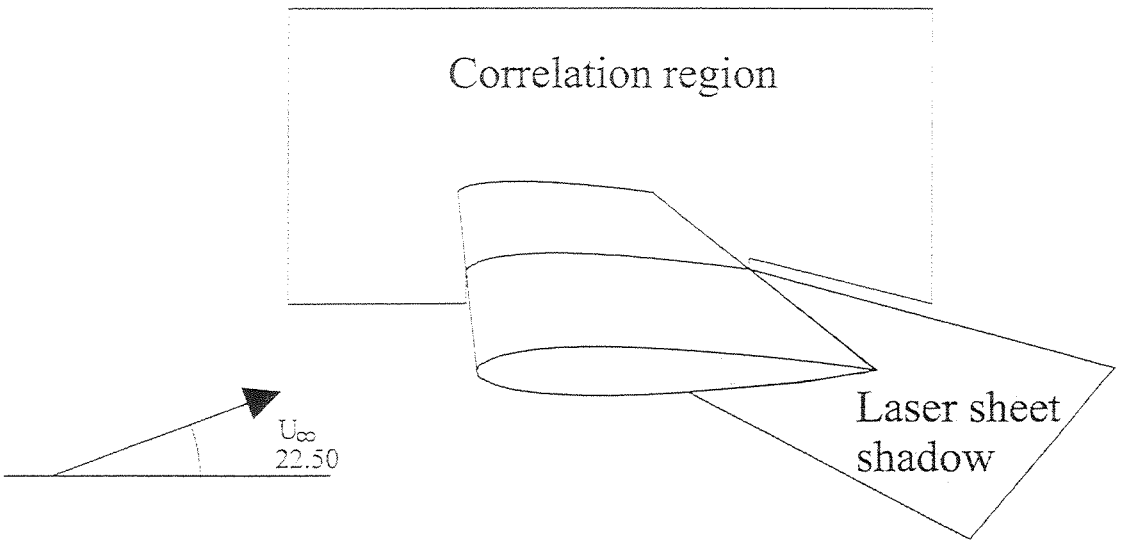


FIG. 17 Accelerating NACA–0012 airfoil and image geometry. CCD camera was positioned to image the lifting surface, with the lower surface occluded, as a consequence.

tank. The final velocity in the acceleration profile was 2.41 cm/s, *i.e.*, below surface capillary-wave speed. A CCD camera was attached to the same carriage that supported the airfoil and was oriented with one pixel axis (approximately) parallel to the airfoil chord, so that the images were recorded in airfoil-fixed coordinates.

The camera recorded instantaneous images of a Lagrangian flow tracer. Both fluorescent dye and particles were tried as markers in the fluid. The results presented are from a run seeded with particles only. These yielded similar, but slightly-better, results than the continuous-scalar-field images. A 2-D slice of the flow at the mid-span of the airfoil was illuminated with a frequency-doubled (532 nm), pulsed (10 pulses/s) Nd:YAG laser, synchronized with the camera. The CCD camera, timing-control electronics, and data acquisition system were built in-house. The system was capable of recording up to 42 images of 1024 × 1024 pixels each, digitized to 12 bits. The timing controllers were programmed to record a three-image sequence, of images spaced by $\tau = 0.1$ s, pause, record the next three-image sequence, *etc.* The period of each cycle was $17\,\tau = 1.7$ s.

As opposed to continuous-scalar Lagrangian tracers/markers that can provide misleading information when the velocity component perpendicular to the imaged

plane is large (Tokumaru & Dimotakis 1995, Sec. 4), particles can produce correct information for the in-plane velocity components, even in the presence of an out-of-plane velocity component. Out-of-plane motion may cause particles captured in one image to disappear, or change intensity, in the next image. This can lead to errors in the inferred local convection and a higher minimum value of the cost function, $\mathcal{J}\{\xi\}$ (Eq. 2). The requirement for a correct measurement of the in-plane velocity components is that the probability that a particle will leave the illumination sheet in the time interval, $\tau$, between images must be small, *i.e.*, the product of the local image-plane-normal velocity component and $\tau$ must be small compared with the laser-sheet thickness (*e.g.*, Dimotakis *et al.* 1981).

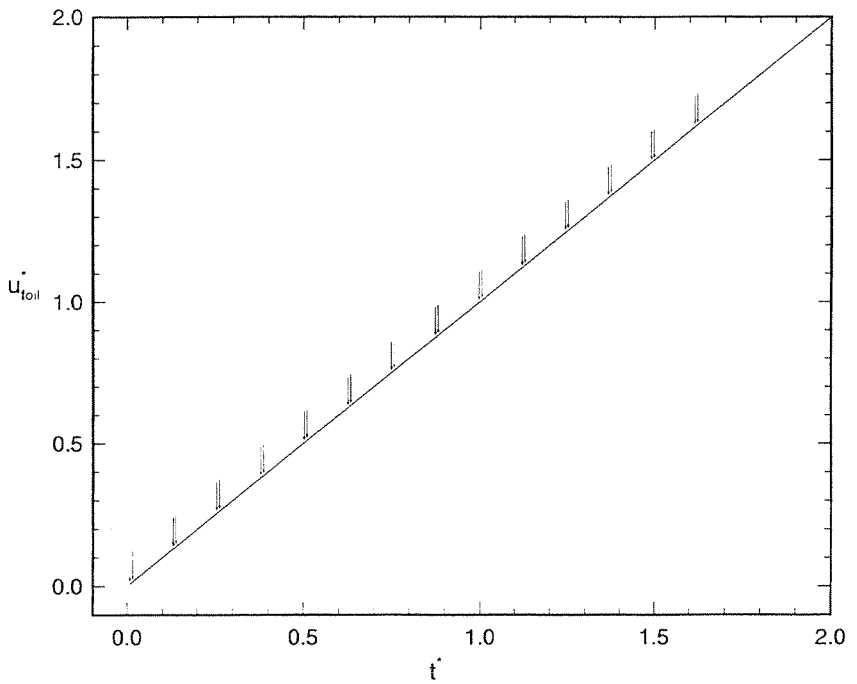


FIG. 18 Image data-acquisition sequence indicated on velocity s time airfoil history. Velocity and time scaled with $\sqrt{2ac}$ and $\sqrt{2c/a}$, respectively, (see text).

The exposure times for the second and third images in each triplet are indicated in Fig. 18. The scaled velocities and times are given by,

$$\mathbf{u}^*(\mathbf{x}^*,t^*) \;=\; \frac{\mathbf{u}(\mathbf{x},t)}{u_0}\;, \qquad t^* \;=\; \frac{t}{t_0}\;, \tag{18a}$$

with,

$$u_0 \;=\; \sqrt{2\,a\,c} \;\simeq\; 1.40\,\mathrm{cm/s}\;, \quad t_0 \;=\; \sqrt{\frac{2\,c}{a}} \;\simeq\; 12.9\,\mathrm{s}\;, \quad \text{and} \quad \mathbf{x}^* = \frac{1}{c}\,\mathbf{x}\;. \tag{18b}$$

For the constant acceleration employed here, the airfoil velocity (in the lab frame) plotted in Fig. 18 was given by $u^*_{\text{foil}} = t^*$. The middle image of each triplet sequence was recorded at times (recall that $\tau = 0.1\,\text{s} \simeq 7.75 \times 10^{-3}\, t_0$),

$$t_n \;=\; (17n + 1)\,\tau \;, \quad \text{for} \quad n \;=\; 1, 2, \ldots 13 \;. \tag{19}$$

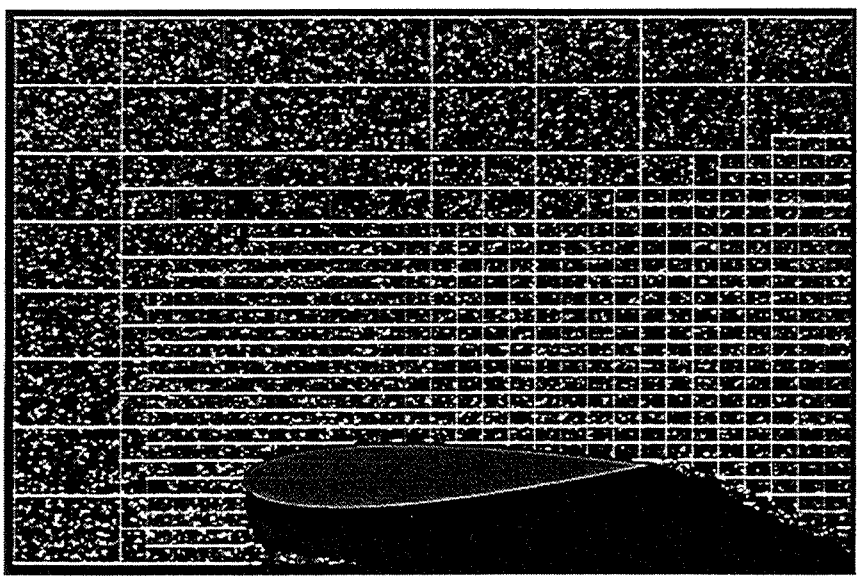

FIG. 19 NACA–0012 accelerated airfoil test case image data with knot grid overlay depicting multi-resolution hierarchy.

Figure 19 depicts the multi-resolution correlation grid overlayed on one of the (particle) images. This grid was used throughout the sequence and was chosen iteratively to possess sufficient spatial resolution to capture the velocity field and its derivatives. The no-slip boundary condition was enforced on the foil surface as discussed in Sec. 2.2. No boundary condition was enforced on the (outer) domain boundary in the flow (free ends) with associated control parameters in the solution as determined by the cross-correlation/optimization algorithm.

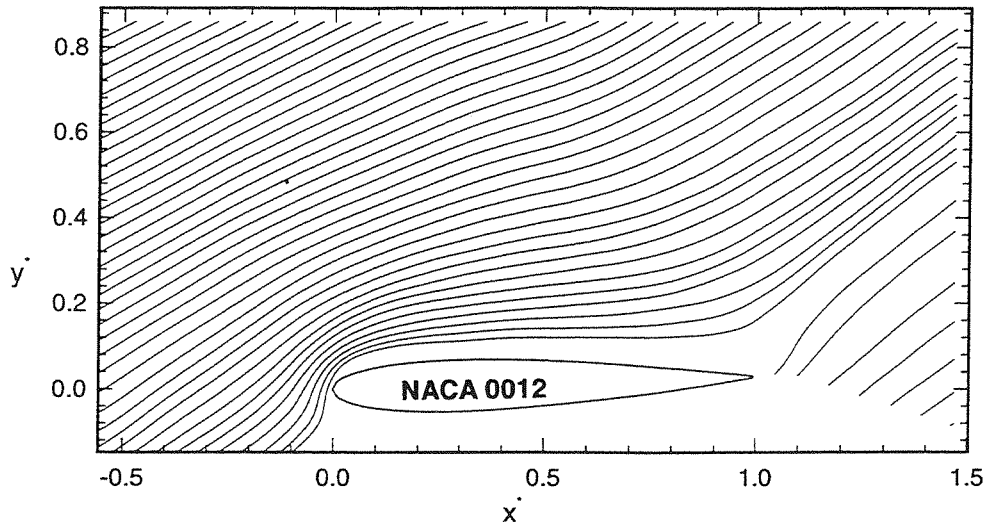


FIG. 20a Streamline field for accelerating airfoil during attached-flow initial-phase. Flow derived from image recorded at $t = t_2$.

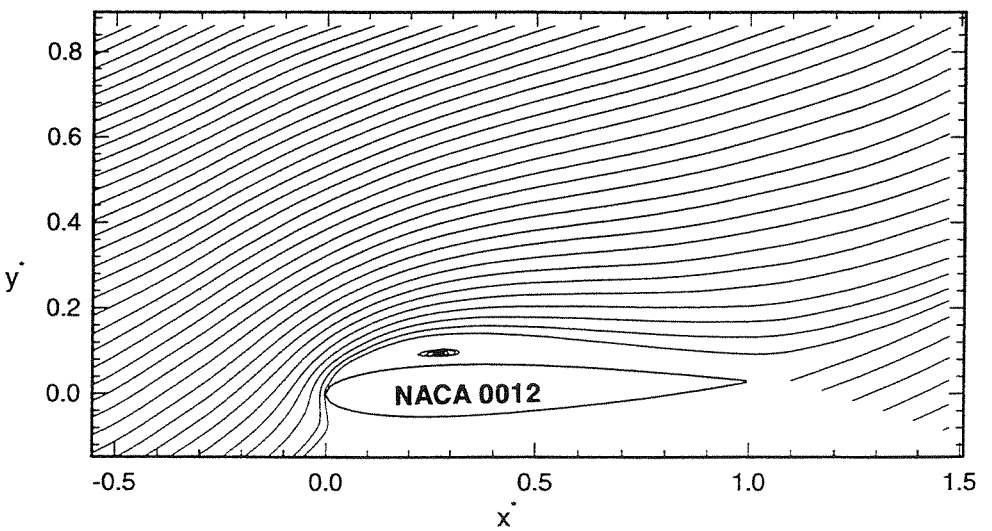


FIG. 20b First appearance of separation bubble. Streamline field derived from image recorded at $t = t_7$.

The resulting flow is found to remain attached to the airfoil, with little change in the topology of the streamline field until a small separation bubble appears on the lifting surface. Figure 20a depicts the velocity-tangent (streamline) field during the attached-flow phase, recorded at $t = t_2$ (Eq. 19). The inclination of the streamlines above and away from the airfoil reflect the freestream at an inclination near the angle

of attack ($\alpha = 22.5°$C). Figure Fig. 20b depicts the streamline pattern recorded at a later time, $t = t_7$. It was the first to capture the separation, which must have occurred during the previous time interval, *i.e.*, $t_6 < t_{sep} < t_7$.

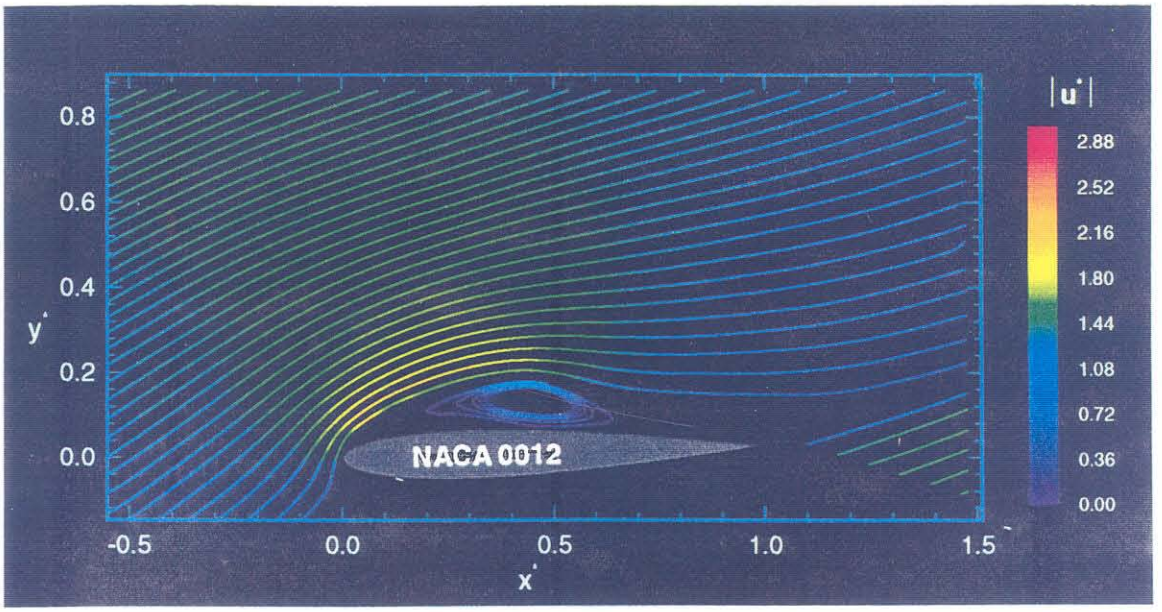


FIG. 20c Color streamline pattern at $t_9$. Color codes (scaled) velocity magnitude (Eq. 18).

The streamline field is useful in displaying the topology of the instantaneous flow but does not convey velocity-magnitude information. The two can be combined in a streamline plot, however, in which color denotes velocity magnitude. Such a plot is reproduced in Fig. 20c, recorded at $t_9$. As can be seen, the low-velocity (blue-purple) separation bubble has grown in the interim, with a high-velocity region (green-yellow) projected some distance above the airfoil. Some distance away from the airfoil, viscous effects are small, the (unsteady) Bernoulli equation applies, and color (velocity magnitude) may be used as an indicator of the low-pressure regions in the flow.

Figure 20d reproduces the flow field captured in the next interval, $t_{10}$. The separation bubble has grown further and moved aft. The ICV solution has captured the front- and rear-stagnation points on the lifting surface. The scaling employed (Eq. 18) results in dimensionless velocities that continuously (linearly) increase in magnitude with time.

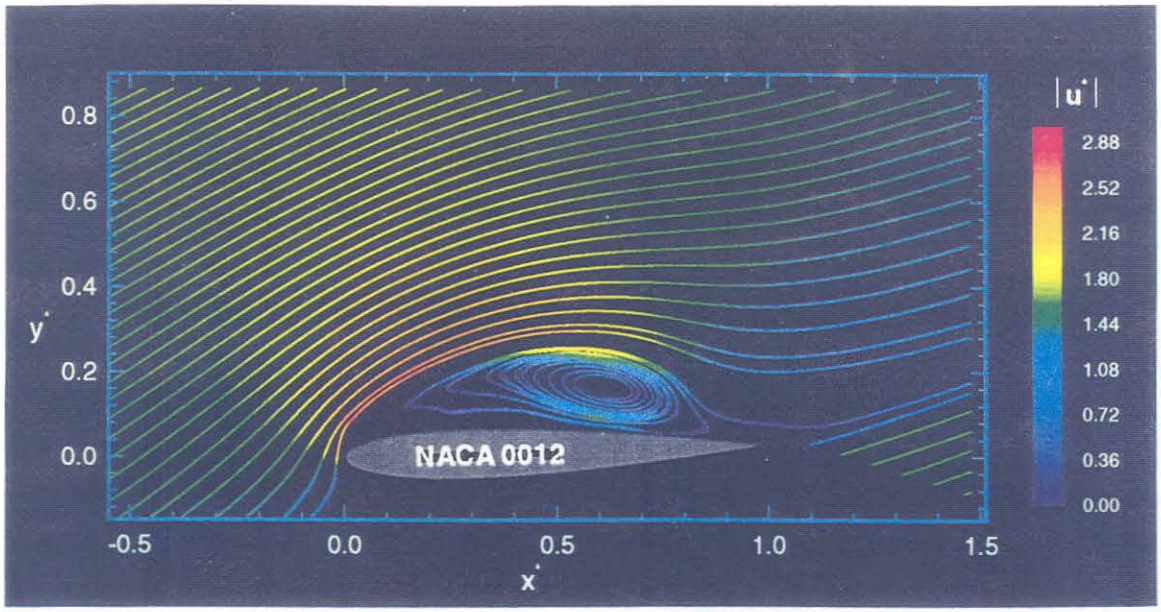


FIG. 20d Color streamline pattern at $t_{10}$, indicating front- and rear-stagnation points on airfoil lifting surface.

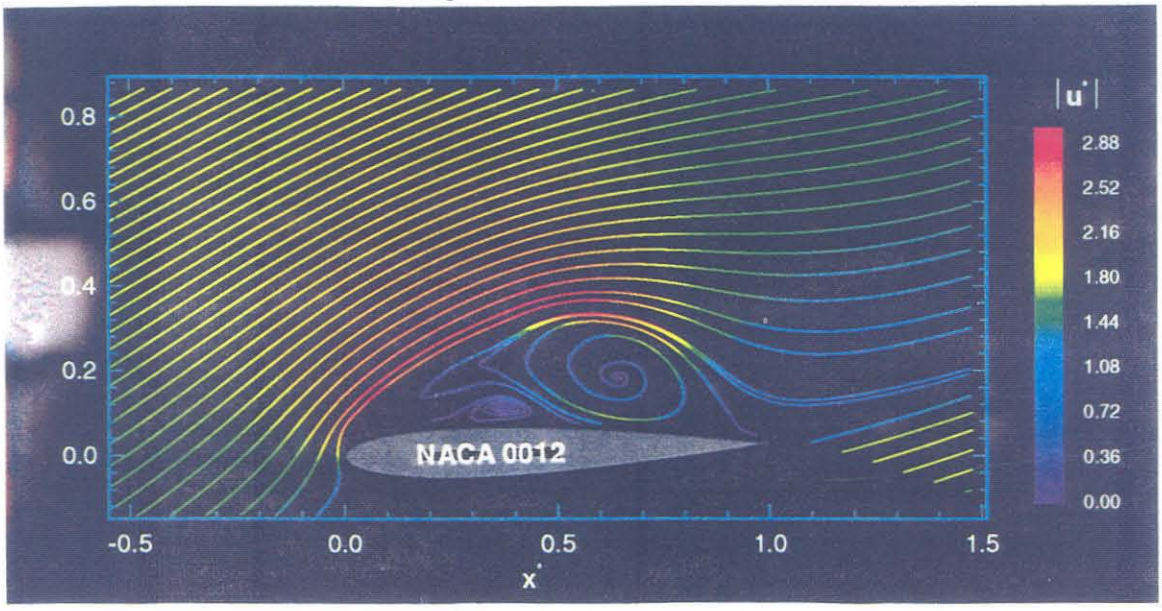


FIG. 20e Color streamline pattern at $t_{12}$, indicating secondary separation bubble.

Figure 20e reproduces the flow field at $t_{12}$. The rear stagnation point of the primary separation bubble is now (approximately) coincident with the airfoil trailing edge, while a secondary separation bubble has appeared. Evidence of high-shear regions are evident on the periphery of the separated-flow region, as well as in the wake.

The final realization in the time sequence is depicted in Fig. 20f. The primary-

vortex rear-stagnation point is now off the airfoil, the secondary vortex has been substantially lifted, and a tertiary separation vortex has been ejected in the low-velocity region above the airfoil. The shear-layers on the periphery of the separated region and in the wake are stronger yet.

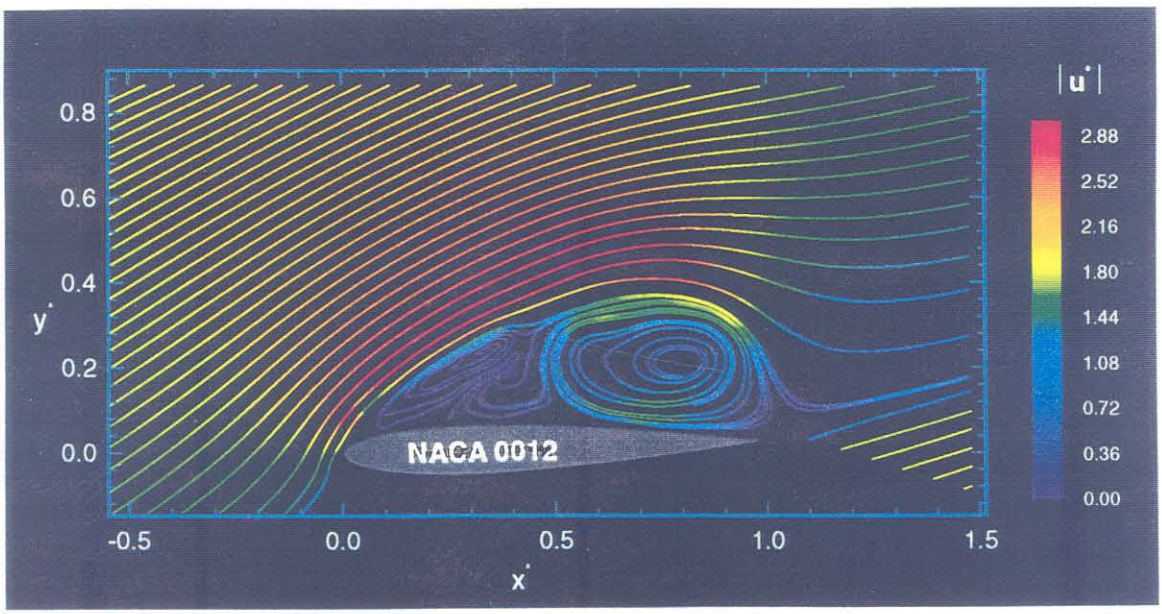


FIG. 20f Color streamline pattern at the final time, $t_{13}$, indicating tertiary separation vortex.

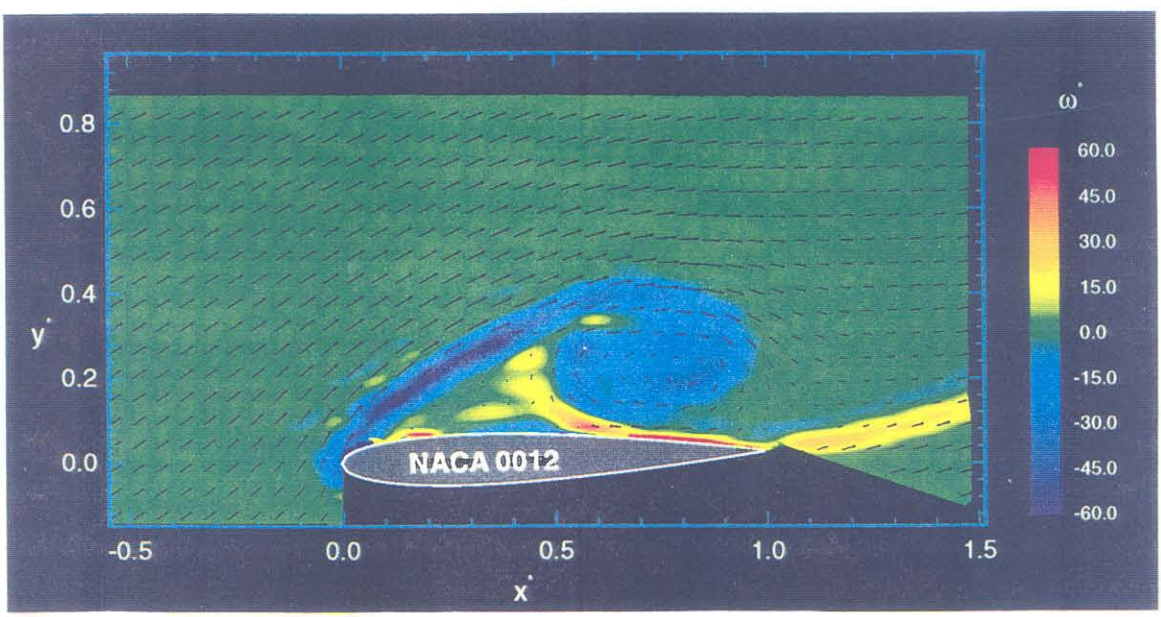


FIG. 21 Velocity vectors and vorticity field (color) plot at $t_{13}$ (same time as in Fig. 20f).

A traditional velocity-field plot, superimposed on the computed vorticity field, is depicted in Fig. 21. The primary vortex can be seen to be accumulating the flux of vorticity emanating from the leading-edge region. The wake is seen to possess an asymmetric vorticity field, indicating a strong net positive (counterclockwise) vorticity and a net shear in the wake. Also noteworthy is the strong positive-vorticity region on the rear portion of the lifting surface that is induced by the primary vortex and sustained by the no-slip boundary condition. The weak, alternating positive/negative vorticity layers that parallel the main vorticity trough, as well as the indicated, small, positive-vorticity bubble near the leading edge are, most likely, artifacts.

Comparing the visualization in Fig. 21 to that in Fig. 20f, the value of both can be appreciated. The topology in the separated-flow region is not discernible in the latter, while the high-shear (vorticity) regions are difficult to identify in the former. Both are required to capture the important features of this unsteady-flow realization.

As can be seen in the streamline plots, the computed streamlines in the separated-flow regions are very nearly closed, for most images. While a failure to close exactly could indicate (weak) out-of-plane motion, only small changes in the computed velocity field would be required to close the streamlines, in most realizations. Streamline spiraling is conspicuous, however, in the $t_{12}$ flow, indicating a reduction in radius as high as, roughly, 1/3 to 1/2 to local radius per turn. Interestingly, the spiraling at the next (and final) time, $t_{13}$, is negligible (actually, slightly outward). The robustness of the in-spiraling in the $t_{12}$ realization was checked by processing the second and third images in the $t_{12}$ triplet sequence, as an independent pair. The deduced velocity-streamline pattern was virtually the same. We are led to conclude that a transient, three-dimensional phenomenon has been captured, possibly akin to the axial flow in the vortex cores reported by Koochesfahani (1989), in his study of the shed vortices in the wake of an oscillating NACA–0012 airfoil.

## 4.2 Transverse jet

The ICV method was also used to analyze scalar images derived from a transverse jet in a co-flow in the GALCIT Free Surface Water Tunnel. The jet fluid was marked with kriegrocine dye and illuminated by a pulsed, frequency-doubled (532 nm) Nd:YAG laser sheet. The laser was formed into a sheet with a cylindrical lens, aligned with the freestream and centered on the jet axis. The width of the sheet was narrowed by a long focal length spherical lens. Image data were acquired at the laser PRF of 10 Hz, on the same $1024 \times 1024$ CCD camera used for the accelerating NACA–0012 flow test case.

In the image domain in the far field correlated by ICV ($100 \lesssim x/d \lesssim 200$), inter-frame displacement of the scalar field, *i.e.*, labeled jet fluid, was predominantly in-plane. Such images do not provide a correlatable scalar field throughout the field of view (*e.g.*, there is no scalar gradient beyond the boundary of the jet fluid) and require an adaptive definition of the correlation domain, $\Omega$. This was defined for these images as a non-convex domain with a polygonal boundary, as described in Sec. 2.2.

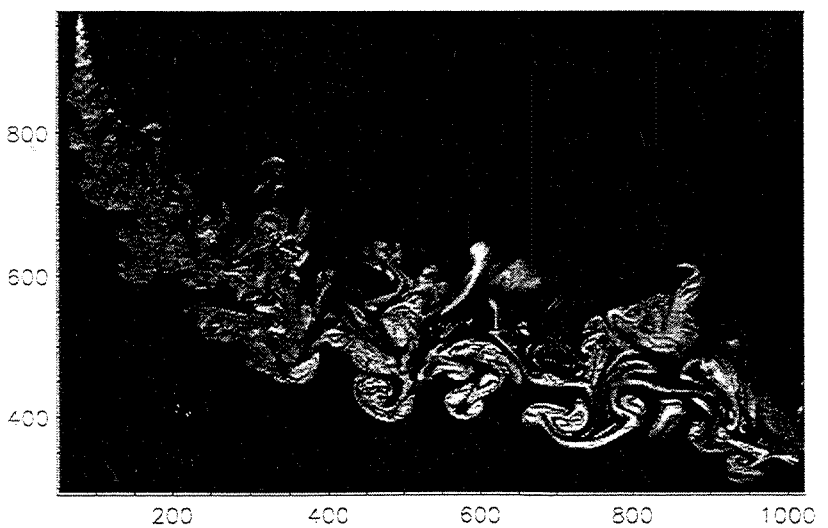


FIG. 22 Transverse-jet in coflow. Image of jet-fluid concentration in the plane of symmetry of the jet. $Re_{\text{jet}} \approx 3000$.

As mentioned, the velocity is predominantly in plane within the correlation

region, however, there is a small amount of out-of-plane (spanwise) velocity. In this case of a 2-D slice of a 3-D flow labeled by a scalar with 3-D structure, the out-of-plane velocity component manifests as an apparent in-plane divergence, and also as error in the inferred in-plane velocity. For more discussion see Tokumaru & Dimotakis (1995).

An example of the resulting images is depicted in Fig. 22. Successive pairs of such images were correlated only within a specified domain with the displacement field represented on a hierarchical, 32 × 32-knot grid, using a bicubic B-spline representation for the displacement field. The results are presented in Fig. 23, that also indicates the correlation domain. Color labels the inferred (out-of-plane) $z$-component of the vorticity, as identified in the color table, in units of reciprocal frame intervals $(1/\tau)$, with overlaid arrows representing the inferred velocity field. The primary components of vorticity for this flow are in-plane, corresponding to a counter-rotating vortex pair, asymptotically tending towards the streamwise direction, with substantial amplitudes for the instantaneous out-of-plane component, however, for this turbulent flow.

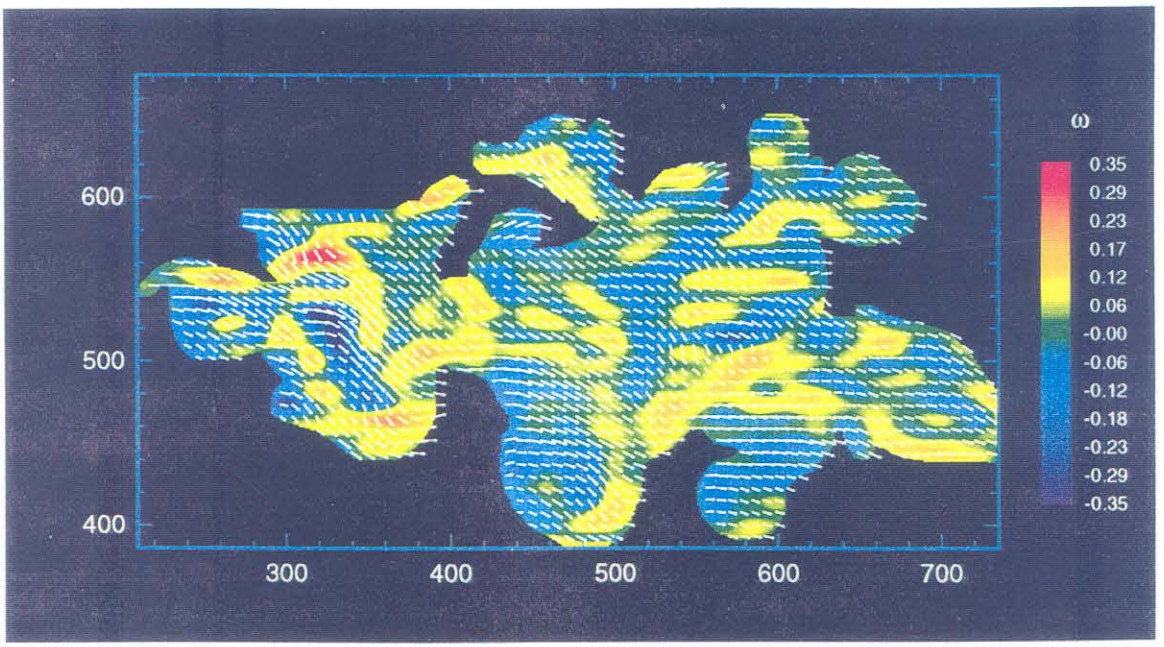


FIG. 23 Continuous-field ICV results for correlation of transverse jet in co-flow test case. Black regions are outside of the correlation domain $\Omega$, while color indicates out of plane vorticity, velocity arrows are overlayed in black.

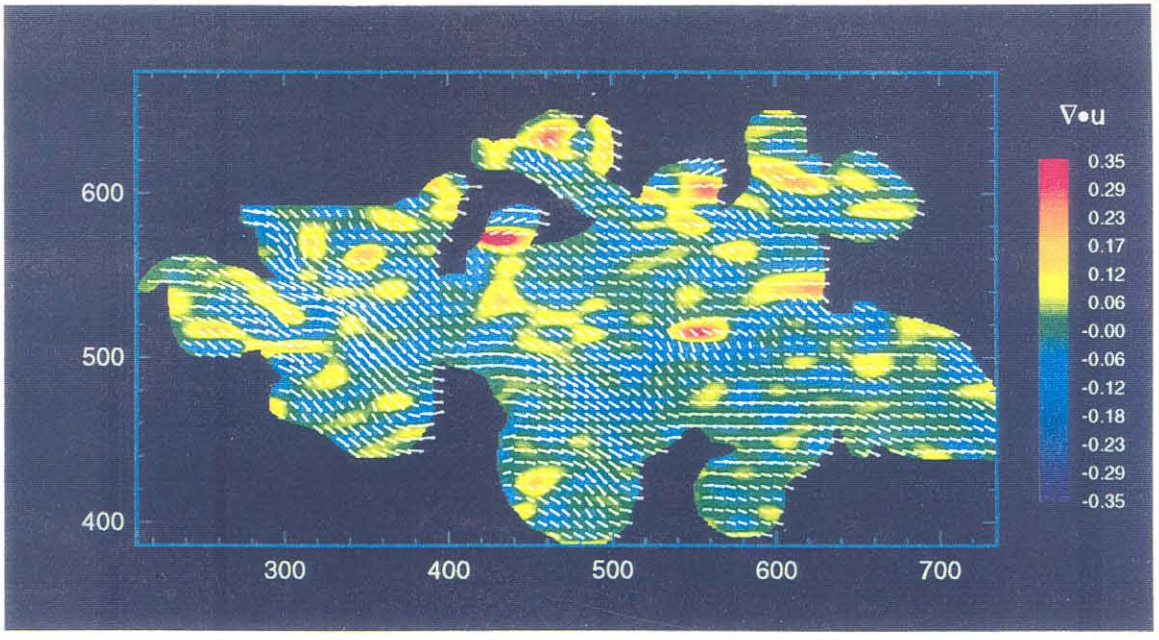


FIG. 24 Divergence field for results as described in Fig. 23 and text.

## 5. Conclusions

Continuous-field Image Correlation Velocimetry methodology is a versatile means of recovering optical flow from a variety of image sequences of convected Lagrangian markers. This implementation uses B-splines of arbitrary (user-selected) order to represent the displacement field between a pair of images. The user can also select a multi-resolution hierarchical B-spline representation, where appropriate.

The method was tested on simulated flow fields. Numerically-generated images with various levels of added noise were used as input to the ICV algorithm. In the case of the simulated Lamb-Oseen vortex, the algorithm deduced the vorticity field with a 0.6% rms error over the entire correlation domain. For the test cases with noise added to the images, the advantage of using larger correlation regions, that still had sufficient spatial resolution to capture the velocity field, was shown.

A further test case investigated the simulated boundary layer flow. Due to simple geometry, the boundary condition was enforced exactly at the (simulated) wall. The advantage of using a multi-resolution hierarchical vs. a full-resolution hierarchical displacement field representation was investigated. The full-resolution representation performed best with no added noise to the images, and was able to

deduce the wall-parallel component of the velocity field with 0.3% rms error over the entire correlation domain. For images with added noise, the multi-resolution representation performed slightly better, but it is questionable whether the improvement was significant enough to warrant the added complexity.

Two laboratory-flow test cases were also reported. The accelerating NACA–0012 airfoil at high angle of attack shows the utility of the method to capture high dynamic-range (in the velocity field) flows. From one set of full-field images, the method captured the onset of separation, development of the primary shed vortex, in addition to secondary and tertiary vortices. The latter of the three vortices was found in a very low energy region of the flow and can only be identified when data are displayed as an instantaneous streamline plot. A shift of the B-spline representation knot grid (somewhat analogous to a CFD grid refinement study) revealed the same structures, thus it is believed that the tertiary vortex was not an artifact of the method.

The second laboratory-flow test case was a transverse jet in a coflowing stream, where the jet concentration imaged using fluorescent dye. The boundary of the correlation domain considered in the minimization sequence of the code was along a prescribed, complex-curve. Significant regions of vorticity were found, with out-of-plane motion producing as in-plane divergence.

APPENDIX A

# B-spline representation

B-splines are an efficient means of representing curves and surfaces for computational evaluation. The NURBS (non-uniform rational B-spline) is a popular version of B-splines used extensively in the computer-graphics and CAD/CAM/CAE industries for its ability to analytically represent conic sections using rational polynomials (Farin 1992). In the representation of fluid flows, there is of course no need to model conic sections. Accordingly, the non-rational, non-uniform, B-splines were chosen since they allow sufficient degrees of freedom without adding additional complexity for unnecessary features. A brief introduction to B-splines will be given below, demonstrating the construction of a curve, followed by a generalization to higher dimensions (surfaces, volumes, *etc.*).

B-splines are piecewise polynomials of a chosen degree, $p$. The splines are continuous to the $(p-1)^{\text{th}}$ derivative, *e.g.*, a cubic spline ($p = 3$) is $\mathcal{C}^2$ (continuous second derivatives). The advantage of B-splines, over other potential representations, is that they are evaluated as a weighted sum of easily-computed basis functions that by construction satisfy the specified continuity requirement, $\mathcal{C}^{(p-1)}$, over a minimal number of piecewise intervals. In one dimension, the sum,

$$f(s) \;=\; \sum_i q_i \, B_{i,p}(s) \;, \tag{20}$$

evaluates a spline, given the basis functions, $B_{i,p}(s)$, and a set of scaling or weighting parameters, $q_i$. The basis functions can be computed ahead of time and a different spline, with different parameters $q_i$, can be evaluated efficiently, over the same piecewise domain.

The independent variable, $s$, can be parametric, *e.g.*, $0 \leq s \leq 1$, so that a curve embedded in a higher-dimensional space is parameterized along $s$, *i.e.*, $\mathbf{f}(s) = [f_1(s), f_2(s), \ldots]$. Alternatively, one could let the range of $s$ coincide with a chosen domain so that the B-spline representation is an explicit function of one variable, *i.e.*, $f = f(s)$.

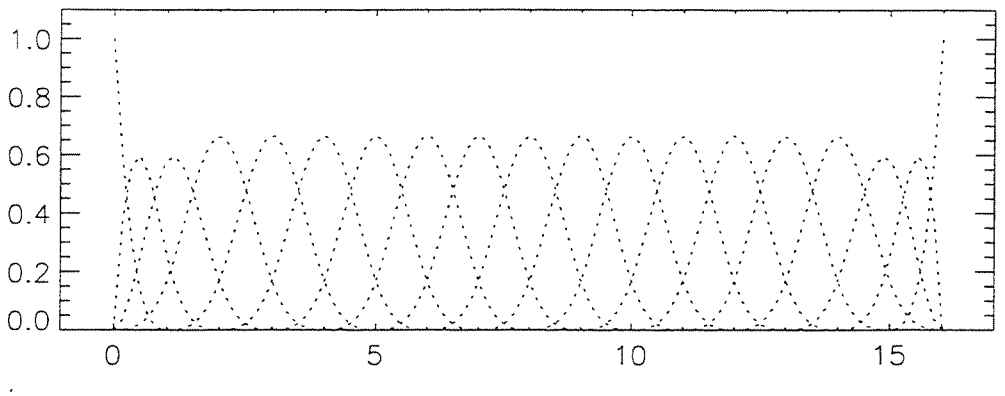


FIG. 25 Cubic non-uniform 1-D B-spline basis functions.

There are many methods to generate basis functions, including blossoming and divided differences (Bartels *et al.*1987). One that lends itself particularly well to numerical evaluation is the recurrence method attributed to deBoor (1978), or Cox, deBoor, & Mannsfield (Piegl & Tiller 1995). A set of cubic basis functions, $B_{i,3}$, generated in this fashion, is plotted in Fig. 25.

To evaluate the basis functions, piecewise intervals must first be defined over the domain of $s$. The resulting sequence of (non-decreasing) numbers is referred to as a *knot vector*, where the joint between each interval is a knot. The knot vector used in the generation of the basis functions in Fig. 25 was,

$$\mathbf{s} = \{0, 0, 0, 0, 1, 2, 3, \ldots, 13, 14, 15, 16, 16, 16, 16\} \ . \tag{21}$$

There are 16 non-zero piecewise segments (knot intervals). The collapsed (zero-length) knot intervals at the ends of the knot vector allow the generation of useful basis functions for boundary conditions, as will be discussed below.

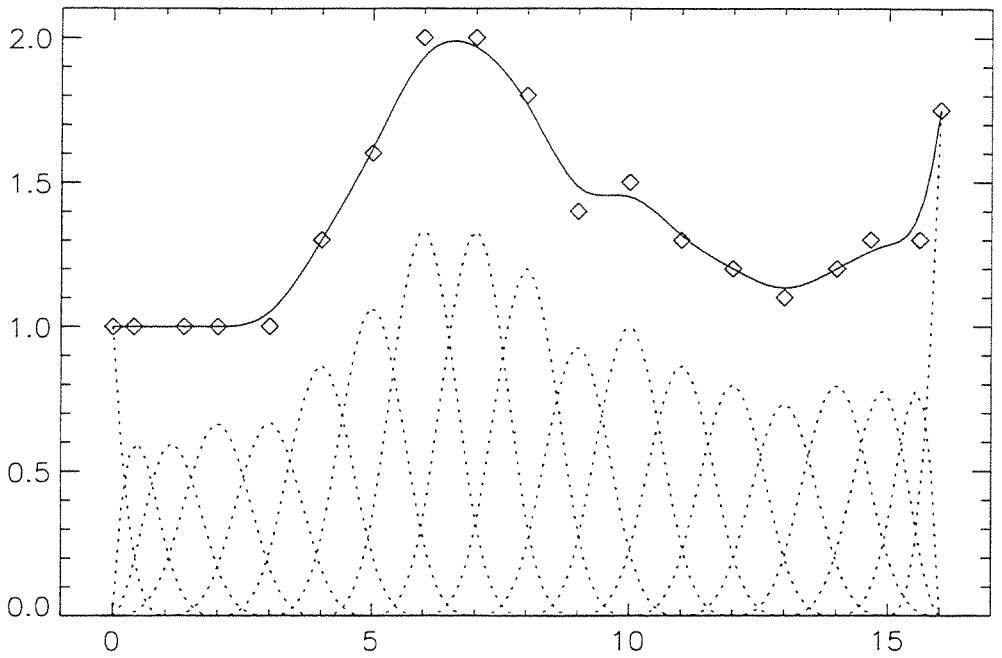


FIG. 26 Cubic non-uniform B-spline curve, (solid line). Basis functions (broken lines) have been weighted by their corresponding parameters (diamonds).

Given the knot vector, $\mathbf{s} = \{ s_i \}$, one can recursively evaluate the basis functions from,

$$\begin{aligned} B_{i,0}(s) &= \begin{cases} 1, & \text{for } s_i \leq s \leq s_{i+1}\,; \\ 0, & \text{otherwise;} \end{cases} \\ B_{i,p}(s) &= \frac{s - s_i}{s_{i+p} - s_i} B_{i,p-1}(s) + \frac{s_{i+p+1} - s}{s_{i+j+1} - s_{i+1}} B_{i+1,j-1}\ . \end{aligned} \tag{22}$$

Each recursive evaluation of Eq. 22 increases the basis-function degree by one. Thus, it is easy to select the order of continuity of a spline, for a given knot vector. Each additional increase of basis-function degree will increase the number of knot intervals over which the basis function is non-zero (excepting the boundaries), also resulting in an increased computational burden when evaluating Eq. 20.

There are a total of 19 cubic basis functions for the example knot vector, with

each $B_{i,3}(s)$ basis function requiring a control parameter to define the spline. The example curve of Fig. 26 is constructed from the basis functions shown in Fig. 25. Each basis function has been scaled by its control parameter, $q_i$, whose value is plotted above the peak of the corresponding basis function.

At the boundaries of the spline there are typically fewer continuity requirements and one has more latitude in the shape of the basis functions. The common choice (and the one made here) is to construct basis functions for the boundary regions that give full support to the resulting spline, *i.e.*, the spline interpolates the bounding (first and last) $q_i$'s. These basis functions are generated by Eq. 22, with the collapsed outer-knot intervals in the knot vector, **s**.

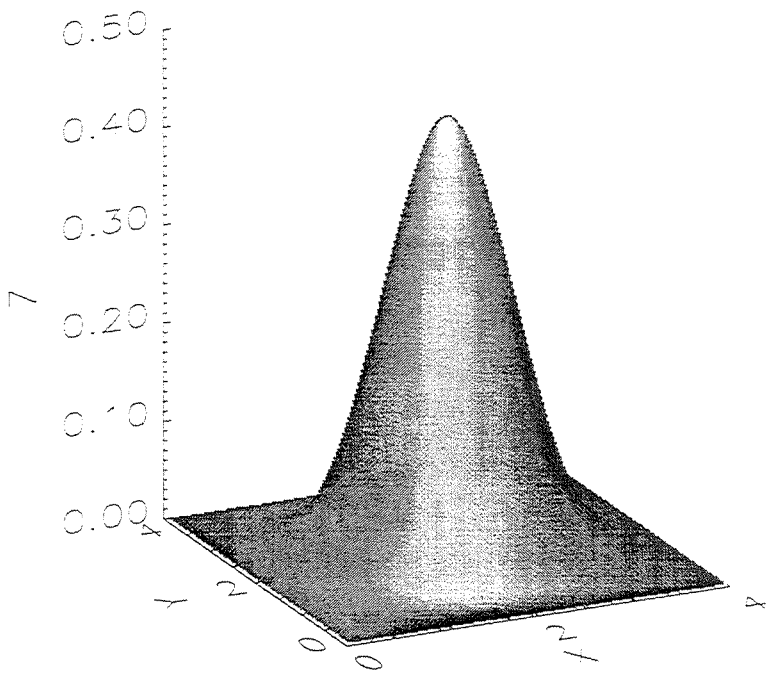


FIG. 27 Two-dimensional, cubic, uniform B-spline basis function.

Generalization to higher dimensions is straightforward. A surface, for example, may be regarded as a summation of a grid of weighted 2-D basis functions, such as the one shown in Fig. 27. The 2-D basis functions are generated as the tensor

product of 1-D sets of basis functions, with each 1-D set corresponding to a different direction. A vector field can employ the same 2-D basis functions, with vector parameters $\mathbf{q}_{ij}$, with the two-dimensional extension of Eq. 20 given by,

$$\mathbf{f}(\mathbf{x}) \;=\; \sum_{i,j} \mathbf{q}_{ij}\, B_{i,p}(x)\, B_{j,p}(y) \;, \tag{23}$$

for a $p^{\text{th}}$-degree vector-field representation.

## References

ACKERMANN, F. 1983 "High Precision Digital Image Correlation," *Proceedings of the* 39th *Photogrammetric Week* (Stuttgart University), 231–243.

ADRIAN, R. J. 1991 "Particle-Imaging Techniques for Experimental Fluid Mechanics," *Ann. Rev. Fluid Mech.* **23**, 261–304.

ANANDAN, P. 1989 "A Computational Framework and an Algorithm for the Measurement of Visual Motion," *Int. J. Comp. Vision* **2**, 283–310.

BARRON, J. L., FLEET, D. J. & BEAUCHEMIN, S. S. 1994 "Performance of Optical Flow Techniques," *Int. J. Comp. Vision* **12**, 43–77.

BARTELS, R.H., BEATTY, J.C. & BARSKY B.A. 1987 *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling* (Morgan Kaufmann, Los Altos, CA).

BATCHELOR, G. K. 1967 *An Introduction to Fluid Dynamics* (Cambridge U. P.).

DAHM, W. J. A., SOUTHERLAND, K. B. & BUCH, K. A. 1991 "Direct, High Resolution, Four-Dimensional Measurements of the Fine Scale Structure of $Sc \gg 1$ Molecular Mixing in Turbulent Flows," *Phys. Fluids A* **3**(5, Pt. 2), 1115–1127.

DAHM, W. J. A., SU, L. K. & SOUTHERLAND, K. B. 1992 "A Scalar Imaging Velocimetry Technique for Fully Resolved Four-Dimensinoal Vector Velocity Measurements in Turbulent Flows," *Phys. Fluids A* 4(10), 2191–2206.

DAHM, J. A., SU, L. K. & TACINA K. M. 1996 "Four-Dimensional Measurements of Vector Fields in Turbulent Flows," (invited paper) *AIAA* 27th *Fluid Dynamics Conference*, Paper 96–1987.

DE BOOR, C. 1978 *A Practical Guide to Splines* (*Applied Mathematical Sciences* **27**, Springer-Verlag).

DIMOTAKIS, P. E., DEBUSSY, F. D. & KOOCHESFAHANI, M. M. 1981 "Particle Streak Velocity Field Measurements in a Two-Dimensional Mixing Layer," *Phys. Fluids* **24**(6), 995–999.

DRACOS, T. & GRUEN, A. 1997 "Videogrammetric Methods in Velocimetry," *Appl. Mech. Rev. (ASME)* (to appear).

FARIN, G. 1992 "From Conics to NURBS: A Tutorial and Survey," *IEEE Computer Graphics & Applications* (September), 78–86.

FORSEY, D.R. & BARTELS R.H. 1988 "Hierarchical B-Spline Refinement," *Comp. Graphics* **22**(4), 205–212.

FORSEY, D.R. & BARTELS R.H. 1995 "Surface Fitting with Hierarchical Splines," *ACM Transactions on Graphics* **14**(2), 134–161.

GRUEN, A. W. 1985 "Adaptive Least Squares Correlation, a Powerful Image Matching Technique," *South African Journal of Photogrammetry, Remote Sensing and Cartography* **14**(3), 175–187.

HORN, B.K.P. & SCHUNCK, B.G. 1981 "Determining Optical Flow," *Artificial Intelligence* **17**, 185–203.

HUANG, M.-J. 1994 "Theoretical and Computational Studies of Isotropic Homogeneous Turbulence," California Institute of Technology, Ph.D. thesis.

KOOCHSESFAHANI, M. M. 1989 "Vortical Patterns in the Wake of an Oscillating Airfoil," *AIAA J.* **27**(9), 1200–1205.

MAAS, H.-G. 1993 "Determination of Velocity Fields in Flow Tomography Sequences by 3-D Least Square Matching," Conference proceedings, *Optical 3-D Measurement Techniques* **II** (4-7 October 1993, Zurich, Switzerland), 366–376.

MITCHELL, D.P. & ARUN, N.M. 1988 "Reconstruction Filters in Computer Graphics," *Comp. Graphics* **22**(4), 221-226.

PEARLSTEIN, A. J. & CARPENTER, B. N. 1995 "On the Determination of Solenoidal or Compressible Velocity Fields from Measurements of Passive or Reactive Scalars," *Phys. Fluids A* **7**(4), 754–763.

PIEGL, L. & TILLER, W. 1995 *The NURBS Book* (Springer-Verlag, Berlin).

PRESS, W.H., TEUKOLSKY, S.A., VETTERLING, W.T. & FLANNERY, B.P. 1992 *Numerical Recipies in C. The Art of Scientific Computing.* (Cambridge University Press).

SAFFMAN, P. G. 1992 *Vortex Dynamics* (Cambridge University Press).

SHOLL, M. & SAVAS, Ö. 1997 "A Fast Lagrangian PIV Method for Study of General High-Gradient Flows," *AIAA* 35$^{th}$ *Aerospace Sciences Meeting*, Paper 97–0493.

SU, L. K. & DAHM, W. J. A. 1995 *Scalar Imaging Velocimetry and Its Application in Measurements of the Structure and Dynamics of the Complete Velocity Gradient Tensor in Turbulent Flows*, Ph.D. thesis, University of Michigan.

SZELISKI, R. & SHUM, H. 1996 "Motion Estimation with Quadtree Splines," *IEEE Trans. Pattern Matching and Machine Intelligence* **18**(12), 1199–1210.

TOKUMARU, P. T. & DIMOTAKIS, P. E. 1995 "Image Correlation Velocimetry," *Exps. in Fluids* **19**(1), 1–15.

WILLERT, C.E. & GHARIB, M. 1991 "Digital Partical Image Velocimetry," *Exps. in Fluids* **10**, 181–193.

WILLICK, D. & YANG, Y.-H. 1991 "Experimental Evaluation of Motion Constraint Equations," CVGIP: Image Understanding 54(2), 206–214.

ZHOU, Z., SYNOLAKIS, C. E., LEAHY, R. M. & SONG, S. M. 1995 "Calculation of 3D Internal Displacement Fields from 3D X-ray Computer Tomographic Images," *Proc. Roy. Soc. London* A **449**, 537–554.