Final Report

AFOSR Grant F49620-95-1-0297

Global Two-Scalar Velocimetry and Development of Low-Order Models for Use in Optical-Phase Correction in a Plane Mixing Layer

Arne J. Pearlstein
Department of Mechanical and Industrial Engineering
University of Illinois at Urbana-Champaign
1206 West Green Street
Urbana, IL 61801

9 4 MAR 2000

REPORTDOCUMENTATIONPAGE

AFRL-SR-BL-TR-00-

0096

1.REPORTDATE (DDMMYYYY)	2.REPORTDATE	3.DATESCOVERED (FromTo)
March 4, 2000		04/01/1995-09/30/1998

4.TITLEANDSUBTITLE
Global Two-Scalar Velocimetry and Development of Low-Order Models for Use in Optical-Phase Correction in a Plane Mixing Layer

5a.CONTRACTNUMBER

5b.GRANTNUMBER
F49620-95-1-0297

5c.PROGRAMELEMENTNUMBER
2307/BS, 61102F

5d.PROJECTNUMBER

5e.TASKNUMBER

5f.WORKUNITNUMBER

6.AUTHOR(S)
Arne J. Pearlstein

7.PERFORMINGORGANIZATIONNAME(S)ANDADDRESS(ES)
The Board of Trustees of the University of Illinois
801 South Wright Street
Champaign, IL 61820

8.PERFORMINGORGANIZATION REPORTNUMBER

9.SPONSORING/MONITORINGAGENCYNAME(S)ANDADDRESS(ES)
Air Force Office of Scientific Research
801 North Randolph Street
Arlington, VA 22203

10.SPONSOR/MONITOR'SACRONYM(S)
AFOSR/NA

11.SPONSORING/MONITORING AGENCYREPORTNUMBER

12.DISTRIBUTIONAVAILABILITYSTATEMENT
Unrestricted

13.SUPPLEMENTARYNOTES

14.ABSTRACT
Work was conducted on two projects related to control of shear flows. In the first, two-dimensional simulations were performed to study the effects of resolution on the extraction of velocity data from measurements of scalar fields in incompressible flows. That work demonstrates "proof-of concept" for the methodology described in earlier work supported by AFOSR. In the second, we developed a general-purpose algorithm for computing all solutions of an arbitrary system of multivariate polynomial equations. That work constitutes the basis of a technique for determining the coefficients in low-order models of scalar transport in turbulent shear flows. The polynomial root-finder has been applied to construction of low-order models of scalar transport in a turbulent plane mixing layer.

15.SUBJECTTERMS
Fluid mechanics, scalar velocimetry, low-order models, proper orthogonal decomposition, fixed point methods, multivariate polynomial equations

16.SECURITYCLASSIFICATIONOF: a.REPORT	b.ABSTRACT	c.THISPAGE	17.LIMITATIONOF ABSTRACT	18.NUMBER OFPAGES	19a.NAMEOFRESPONSIBLEPERSON / 19b.TELEPONENUMBER(*Includeareacode)*
				46	

StandardForm298(Rev.898)
PrescribedbyANSIStdZ3918

03-14-00A09:28 REF:

Technical Report on AFOSR Grant F49620-95-1-0297

Global Two-Scalar Velocimetry and Development of Low-Order Models for Use in Optical-Phase Correction in a Plane Mixing Layer

Summary

Work was conducted on two projects related to control of shear flows.

In the first, two-dimensional simulations were performed to study the effects of resolution on the extraction of velocity data from measurements of scalar fields in incompressible flows. That work, with Bonnie Carpenter, has been published in *Physics of Fluids*, **8**, 2447-2459 (1996) and demonstrates "proof-of concept" for the methodology described in an earlier *Physics of Fluids* paper, also supported by AFOSR. A reprint is included in this report.

In the second, David Cotrell and I have developed a general-purpose algorithm for computing all solutions of an arbitrary system of multivariate polynomial equations. That work constitutes the basis of a technique for determining the coefficients in low-order models of scalar transport in turbulent shear flows. The polynomial root-finder work (preprint included in this report) will be submitted to the *SIAM Journal on Scientific Computing*. The application to low-order model construction is described in a paper (with Ari Glezer, Martin Brooke, and their students) to be submitted to *Journal of Fluid Mechanics).*

1. Global Scalar Velocimetry

This work constituted the bulk of Bonnie Carpenter's MS Thesis. A summary is provided here.

Under earlier AFOSR support, we developed a technique for determining n-dimensional (n=2,3) velocity fields from measurement of n-1 passive or reactive scalars has recently been developed (Pearlstein and Carpenter, *Phys. Fluids*, **7**, 754-763, 1995). That method utilizes n linear, first-order, uncoupled hyperbolic equations for n velocity components, derived from transport equations for n-1 scalars and conservation of mass. The velocity is determined by integration along the characteristic curves defined by the hyperbolic equations. In a second paper (*Phys. Fluids*, **8**, 2447-2459, 1996), supported by Grant F49620-95-1-0297, we presented the results of a computational proof-of-concept study for a steady, two-dimensional, diverging channel flow. We considered the effects of grid resolution (and the shape of its elements) on which the scalar is known and its derivatives are approximated, the integration step size used to calculate the velocity components along characteristic curves, and the effects of multiplicative noise introduced into the scalar field. The results show that extraction of the velocity by integration of the equations along characteristics is stable, and that the techniques proposed (*ibid.*) for removal of singularities are effective. For steady flows, noise in the scalar field measurements can be dealt with by extracting the velocity from the mean of several noisy scalar fields.

2. Low-Order Models of Scalar Transport in Turbulent Flows

2.1. Fixed-Point Algorithm for Solution of Multivariate Polynomial Equation System

We have developed a globalization of the Krawczyk algorithm to compute all real isolated solutions of systems of N real polynomial equations. This is accomplished by transforming the original system to an augmented system in R^{N+1}, in which each of the first N variables lies in the interval $[-1,1]$, and the (N+1)-th variable lies in $[0,\infty)$. The domain of this latter variable can be divided into two intervals $[0,1]$ and $[1,\infty)$, the second of which is mapped to $[0,1]$. Thus, the entire domain R^{N+1} can be examined by considering the finite domain $[-1,1]^N \times [0,1]$ for each of two systems. One of the augmented systems can have one or more solutions not corresponding to finite solutions of the original system. We use techniques from algebraic geometry to transform the (N+1)-th dimensional system so that spurious solutions are excluded, thus restricting the solutions to those corresponding to finite solutions of the original system. The algorithm requires bounds for multivariate polynomials over a finite domain. The best of three bounding methods considered uses an interval extension of the system, which is stored as a rooted, ordered tree, and is equivalent to an N-dimensional Horner scheme that takes advantage of polynomial sparsity.

We have extended the approach to deal with systems having nonsimple solutions (i.e., with multiplicity greater than unity), at which the Jacobian vanishes. This is accomplished by constructing a new system that reduces by one the multiplicity of nonsimple solutions. Except in degenerate cases, this approach can be applied sequentially until no roots have multiplicity exceeding one.

The algorithm discussed above (and in more detail in a preprint included in this report) has been used in determining the coefficients of low-dimensional ordinary differential equations systems describing scalar transport in a turbulent plane mixing layer (see §2.2).

2.2. Low-Order Models of Scalar Transport in a Turbulent Plane Mixing Layer

In joint work with Ari Glezer and Martin Brooke at Georgia Tech, we have demonstrated the utility of the proper orthogonal decomposition (POD) as a data compression technique for efficient representation of massive amounts of spatio-temporal passive scalar data acquired in a turbulent plane mixing layer. We show, by comparison of computation to experiment, that such compressed-data representations are useful in "multi-step-ahead" prediction of scalar transport in a turbulent plane mixing layer. Such capability will allow for drastic reductions in the dimensionality of the control scheme.

In our case, the application of ultimate interest is phase correction in aero-optic flows in shear flows, in which large coherent structures significantly influence temperature and density, and hence index of refraction distributions. A question arises as to whether evolution of the coherent structures can be manipulated in such a way that the optical phase distortion can be predicted or controlled. While entrainment of irrotational fluid in turbulent shear flows is affected by large-scale motions, molecular mixing (and hence reduction of index of refraction gradients) ultimately takes place at the smallest scale. The traditional approach to control of mixing through manipulation of global instability modes of the base flow depends on the classical cascading mechanism to transfer energy from the large coherent structures, whose evolution is being manipulated, to the scales at which molecular mixing occurs. Although mixing at the smallest scales is coupled to control of large coherent structures, more efficient control of mixing might be achieved by direct control of both large- and small-scale mixing processes.

A paper, to be submitted to the *Journal of Fluid Mechanics*, describes our implementation of the POD technique in representing massive amounts of spatio-temporal temperature data in this system. In that paper, we also show how several neural network-based prediction methods are able to predict evolution of the temperature field based on past data. We also show how another approach, based on a low-order nonlinear ODE system constructed from the data, attacks the same problem. The implications of the results for flow control are discussed in a final section.

Publications

One paper has been published that acknowledges support from this grant:

Carpenter, B. N. and A. J. Pearlstein, "Simulation of Extraction of Velocity from Passive Scalar Data in a Two-Dimensional Diverging Channel Flow," *Physics of Fluids,* **8**, 2447-2459, 1996.

One paper acknowledging this grant is essentially ready to be submitted for publication:

Cotrell, D. L. and A. J. Pearlstein, "Globalization of the Krawczyk Algorithm to Compute All Simple and Nonsimple Isolated Real Solutions of Systems of Polynomial Equations," to be submitted to *SIAM Journal on Scientific Computing.*

Another paper acknowledging this grant is in preparation, and will be submitted later this Spring:

Oljaca, M., A. Glezer, W. Zhao, M. Brooke, D. L. Cotrell, and A. J. Pearlstein, Efficient 'Multiple-Step-Ahead' Prediction of Mesoscale Passive Scalar Transport in a Turbulent Plane Mixing Layer Using the Proper Orthogonal Decomposition for Data Compression," in preparation.

Human Resource Development

The work of two graduate students was supported by this Grant.

Bonnie Carpenter completed her MS degree, and was well into her PhD thesis (having passed the departmental PhD qualifying exam) when she decided to marry and go to work at the Aerospace Corporation in southern California. She is the co-author of the paper described in §1.

David Cotrell completed his MS degree, and is continuing as a PhD student in mechanical engineering. He is the co-author of the papers described in §2.

Simulation of extraction of velocity from passive scalar data in a two-dimensional diverging channel flow

Bonnie N. Carpenter[a] and Arne J. Pearlstein[b]
Department of Mechanical and Industrial Engineering, University of Illinois at Urbana–Champaign, 1206 West Green Street, Urbana, Illinois 61801

(Received 6 June 1995; accepted 30 May 1996)

A technique for determining n-dimensional ($n=2,3$) velocity fields from measurement of $n-1$ passive or reactive scalars has recently been developed [Phys. Fluids **7**, 754 (1995)]. The method utilizes n linear, first-order, uncoupled hyperbolic equations for n velocity components, derived from transport equations for $n-1$ scalars and conservation of mass. The velocity is determined by integration along the characteristic curves defined by the hyperbolic equations. Here we present the results of a computational proof-of-concept study for a steady, two-dimensional, diverging channel flow. We consider the effects of the resolution of the grid (and the shape of its elements) on which the scalar is known and its derivatives are approximated, the integration step size used to calculate the velocity components along characteristic curves, and the effects of multiplicative noise introduced into the scalar field. The results show that extraction of the velocity by integration of the equations along characteristics is stable, and that the techniques proposed for removal of singularities are effective. For steady flows, we show that noise in the scalar field measurements can be dealt with by extracting the velocity from the mean of several noisy scalar fields. © *1996 American Institute of Physics.* [S1070-6631(96)03109-1]

I. INTRODUCTION

Several techniques for determining velocity fields from measurements of a single scalar have been proposed.[1–4] In a recent paper,[5] we have discussed the attributes of these approaches and presented a different method, requiring measurement of $n-1$ passive or reactive scalars to uniquely determine an n-dimensional ($n=2$ or 3) solenoidal (divergence-free) velocity field.

As discussed in Ref. 5, our method differs from previous approaches, in that it recovers the exact velocity field in the limit of complete data (i.e., perfect spatial and temporal resolution of noise-free scalar fields). Since experimental data is always acquired with finite spatial and temporal resolution, and is always corrupted to some degree by noise, it is important to understand how our method is affected by these properties of real data, and to develop techniques for minimizing error in the extracted velocity field.

In this paper, we address the issues of spatial resolution and noise in the context of a steady two-dimensional diverging channel flow. Flow in a diverging channel is an excellent testbed for developing an understanding of noise and resolution effects, since the geometry admits an exact solution of the Navier–Stokes equations.[6] Moreover, the divergent nature of the flow is reminiscent of a plane mixing layer, a generic flow of interest in a number of applications. Finally, for simple boundary conditions, this flow gives rise to a singularity in the hyperbolic equations along a curve, thereby allowing evaluation of the technique proposed for integration through such a singularity.[5]

In Sec. II, the analysis and working equations derived earlier[5] for extraction of a two-dimensional, solenoidal velocity field from a single scalar field are briefly reviewed. In Sec. III, we discuss the diverging channel flow and suitable scalar boundary conditions. In Sec. IV, the procedure for extracting the velocity field from the resulting computed scalar field is described. The results of the extraction are presented in Sec. V; discussion and conclusions follow in Sec. VI.

II. DETERMINATION OF TWO-DIMENSIONAL SOLENOIDAL VELOCITY FIELDS FROM MEASUREMENTS OF ONE SCALAR

In this section, we present the working equations derived earlier[5] for extraction of a two-dimensional, solenoidal velocity field from measurements of a single scalar. Our technique is based on recognizing that the transport equation,

$$\frac{\partial S}{\partial t}+\mathbf{u}\cdot\nabla S=\alpha\,\nabla^2 S+R, \tag{1}$$

for a scalar S with diffusivity α is linear in the velocity components, and that it and the solenoidal continuity equation,

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

constitute two linear equations in two velocity components.

The formulas will be displayed in a Cartesian coordinate system. (Results for general curvilinear systems are shown in Ref. 5.) The second term on the right-hand side of (1) allows for the scalar to react, with kinetics that are essentially arbitrary as long as they are known and the rate of reaction depends solely on S, x, y, and t. If temperature is the scalar for which full-field measurements are available, then (1) will be replaced by an energy equation with α replaced by the

[a] Present address: Aerospace Corporation, El Segundo, California 90245.
[b] Corresponding author: Telephone: (217) 333-3658; fax: (217) 244-6534; Electronic mail: arne@ajpiris.me.uiuc.edu

thermal diffusivity κ, and the reaction rate set to zero. In the nonreactive case, in two dimensions, (1) can be written as

$$\frac{\partial S}{\partial t}+u\,\frac{\partial S}{\partial x}+v\,\frac{\partial S}{\partial y}=\alpha\,\nabla^2 S, \tag{3}$$

where ∇^2 is the two-dimensional Cartesian Laplacian, and u and v are the x and y components of $\mathbf{u}$. Equation (3), with the two-dimensional form

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0, \tag{4}$$

of (2) constitutes a pair of linear equations in $u(x,y,t)$ and $v(x,y,t)$. Solving (3) for v and substituting into (4) yields

$$\left[\nabla S\times\nabla\left(\frac{u}{\partial S/\partial y}\right)\right]\cdot\mathbf{e}_z=\frac{\partial}{\partial y}\left(\frac{G}{\partial S/\partial y}\right), \tag{5a}$$

while solution of (3) for u and substitution into (4) yields

$$\left[\nabla S\times\nabla\left(\frac{v}{\partial S/\partial x}\right)\right]\cdot\mathbf{e}_z=-\frac{\partial}{\partial x}\left(\frac{G}{\partial S/\partial x}\right), \tag{5b}$$

where

$$G=\alpha\,\nabla^2 S-\frac{\partial S}{\partial t}. \tag{6}$$

Equations (5a) and (5b) constitute a pair of linear, first-order, uncoupled hyperbolic equations in the velocity components u and v. Given the scalar field data, the solution of (5a) and (5b) is made unique by specification of appropriate boundary conditions on u and v. Equations (5a) and (5b) can then be integrated along characteristic curves, which are the same for both equations. No time derivatives of the velocity components appear in (5a) and (5b), so that no initial data on the velocity field are required. As this is not an initial value problem, errors in the determination of the velocity field will not grow temporally.

Equations (5a) and (5b) have an apparent singularity when either component of the gradient vanishes. As shown in Ref. 5, these apparent singularities are "removable" in all cases except when ∇S vanishes identically over an area. The singularities are removable in the sense that the velocity field can be determined at points or along curves where one or both components of ∇S vanish, and that $\mathbf{u}\cdot\nabla S=0$ over an area poses no difficulty unless $\nabla S=\mathbf{0}$ identically. If only one component of ∇S vanishes, then the integration can be performed for one velocity component using the nonsingular equation of (5a) and (5b), and the other velocity component can be determined from the scalar transport equation (3). If both components of ∇S vanish at a point or on a curve, then $\mathbf{u}$ can be determined as follows. Using index notation, we rewrite (3) as

$$u_j\,\frac{\partial S}{\partial x_j}=G. \tag{7}$$

Taking the gradient of both sides of (7), we obtain

$$u_j\,\frac{\partial^2 S}{\partial x_i\,\partial x_j}=\frac{\partial G}{\partial x_i},\quad i=1,2, \tag{8}$$

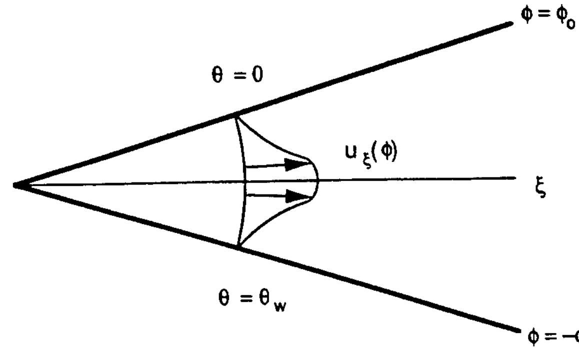

FIG. 1. Flow in a diverging channel. Velocity field shown schematically.

when ∇S vanishes. At each point, (8) is a system of two linearly independent, nonhomogeneous algebraic equations for the velocity components.

III. THE SCALAR FIELD

To better understand the properties of the proposed process for determining the velocity field from passive scalar measurements, a computational proof-of-concept study has been performed using advection of a passive scalar in steady two-dimensional diverging channel flow. We can compare the velocity extracted from the computed scalar field to the Jeffery–Hamel solution,[6] an exact solution of the Navier–Stokes equations. This involves three steps: calculation of the Jeffery–Hamel flow, calculation of the scalar field using the Jeffery–Hamel solution, a scalar transport equation, and appropriate boundary conditions, and extraction of the velocity field from the computed scalar field by integration along characteristics.

One goal of this work is to investigate the effects of spatial resolution and noise on the accuracy of the extracted velocity components. In this steady two-dimensional case, the sources of error in the extracted velocity field include the spatial resolution of the scalar field used to approximate derivatives of the scalar in (5a) and (5b), and the integration step size along the characteristics used in extracting the velocity components. We also consider the effect of multiplicative noise (e.g., shot noise in full-field optical measurement of temperature by laser-induced fluorescence, etc.) corrupting measurements of the scalar field.

A. Exact velocity field

We consider steady two-dimensional flow in a diverging channel, as shown in Fig. 1. In this case, there is an exact solution of the Navier–Stokes equations, referred to as Jeffery–Hamel flow,[6] with the velocity field depending on the half angle ϕ_o and the Reynolds number, $\text{Re}=u_0 r/\nu$, where $u_0(r)$ is the velocity along the central streamline ($\phi=0$). In cylindrical coordinates, this purely radial flow can be written as

$$\mathbf{u}=\frac{\nu F(\phi)}{r}\,\mathbf{e}_r, \tag{9}$$

where F is an even function of ϕ given by

$$F(\phi)=\mathrm{Re}-6\left(\frac{1+\mathrm{Re}/2}{1+k^2}\right)k^2\,\mathrm{sn}^2\left[\left(\frac{1+\mathrm{Re}/2}{1+k^2}\right)^{1/2}\phi,k\right], \tag{10a}$$

with k found from

$$\frac{1+k^2}{3k^2(1+2/\mathrm{Re})}=\mathrm{sn}^2\left[\left(\frac{1+\mathrm{Re}/2}{1+k^2}\right)^{1/2}\phi_o,k\right], \tag{10b}$$

for $\mathrm{Re}<\mathrm{Re}^*(\phi_o)$, or by

$$F(\phi)=\frac{\mathrm{Re}-q+(\mathrm{Re}+q)\mathrm{cn}[(2q/3)^{1/2}\phi,k]}{1+\mathrm{cn}[(2q/3)^{1/2}\phi,k]}, \tag{11a}$$

with k and q defined by

$$\frac{6+\mathrm{Re}(5-4k^2)}{6+\mathrm{Re}(1+4k^2)}=\mathrm{cn}\left[\left(\frac{2+\mathrm{Re}}{2k^2-1}\right)^{1/2}\phi_o,k\right] \tag{11b}$$

and

$$q=\frac{3(1+\mathrm{Re}/2)}{2k^2-1}, \tag{11c}$$

for $\mathrm{Re}>\mathrm{Re}^*(\phi_o)$, where sn and cn are Jacobian elliptic functions, and $\mathrm{Re}^*(5°)=684$.[6] Bisection is used to determine k in (10b) or (11b). Once k is known, the velocity field can be found using Eq. (10a) for $\mathrm{Re}>\mathrm{Re}^*(\phi_o)$ or Eqs. (11a)–(11c) for $\mathrm{Re}<\mathrm{Re}^*(\phi_o)$.

B. Computed scalar field

We use the exact velocity field to calculate the scalar field. Before choosing boundary conditions for the scalar, it is useful to consider the nature of the characteristic equations for steady two-dimensional solenoidal flow. In recognition of the fact that experimental data will frequently be available on domains whose boundaries do not coincide with constant coordinate curves or surfaces of simple coordinate systems, we will extract the velocity components using a Cartesian system, rather than the radial coordinates that are natural for the geometry at hand. Thus, (5a) and (5b) reduce to

$$-S_y u_x+S_x u_y+\left(S_{xy}-\frac{S_x S_{yy}}{S_y}\right)u =\alpha\left((\nabla^2 S)_y-\frac{(\nabla^2 S)S_{yy}}{S_y}\right), \tag{12a}$$

$$-S_y v_x+S_x v_y-\left(S_{xy}-\frac{S_y S_{xx}}{S_x}\right)v =-\alpha\left((\nabla^2 S)_x-\frac{(\nabla^2 S)S_{xx}}{S_x}\right), \tag{12b}$$

where subscripts denote partial derivatives. From (12a) and (12b) one can see that the characteristic curves depend solely on the x and y derivatives of the scalar. Therefore, in order for the characteristics to propagate away from a wall (at $\phi=\pm\phi_o$, on which $u=v=0$), the scalar on at least one wall must be nonuniform. Note that if S_x and S_y are nonzero, characteristics originating on the wall emanate into the flow or in the opposite direction out of the channel. If a characteristic originating on the wall does not emanate into the flow, then the direction of integration along the characteristic must be reversed. This can be accomplished easily by a change of variables. Letting $u'=-u$ and $v'=-v$ and substituting into (12a) and (12b) yields

$$-S_y u'_x+S_x u'_y+\left(S_{xy}-\frac{S_x S_{yy}}{S_y}\right)u' =-\alpha\left((\nabla^2 S)_y-\frac{(\nabla^2 S)S_{yy}}{S_y}\right), \tag{13a}$$

$$-S_y v'_x+S_x v'_y-\left(S_{xy}-\frac{S_y S_{xx}}{S_x}\right)v' =\alpha\left((\nabla^2 S)_x-\frac{(\nabla^2 S)S_{xx}}{S_x}\right). \tag{13b}$$

The net result is to change the signs of the right-hand sides, so that characteristics that did not previously emanate into the flow will now do so.

A simple choice for the boundary conditions on the scalar is

$$S(r,-\phi_o)=\begin{cases} S_1, & r\leq r_1,\\ S_1+\left(\dfrac{r-r_1}{r_2-r_1}\right)(S_2-S_1), & r_1\leq r\leq r_2,\\ S_2, & r_2\leq r,\end{cases} \tag{14a}$$

$$S(r,\phi_o)=S_1. \tag{14b}$$

However, due to the discontinuity in the radial derivative of S on the $\phi=-\phi_o$ wall, the corresponding scalar field is not sufficiently differentiable to allow computation of the third derivatives needed in (12a) and (12b). To obtain a sufficiently smooth scalar field, we will require that Dirichlet conditions on S be continuously differentiable. Such a set of boundary conditions is

$$S(r,-\phi_o)=S_1+\Delta S\,\frac{\tanh b[(r-r_1)/r_1]+\tanh b}{1+\tanh b} \equiv S_w(r), \tag{15a}$$

$$S(r,\phi_o)=S_1, \tag{15b}$$

for $r_1>0$, $b>0$.

At this point, we nondimensionalize by defining $\xi=r/r_1$, $\theta=(S-S_1)/\Delta S$, and $\tilde{\phi}=(\phi+\phi_o)/2\phi_o$. The dimensionless scalar transport equation and boundary conditions are

$$\sigma\,\frac{\tilde{F}(\tilde{\phi})}{\xi}\frac{\partial\theta}{\partial\xi}=\frac{1}{\xi}\frac{\partial}{\partial\xi}\left(\xi\frac{\partial\theta}{\partial\xi}\right)+\frac{1}{4\phi_o^2\xi^2}\frac{\partial^2\theta}{\partial\tilde{\phi}^2}, \tag{16}$$

$$\theta(\xi,0)=\frac{\tanh b(\xi-1)+\tanh b}{1+\tanh b}\equiv\theta_w(\xi), \tag{17a}$$

$$\theta(\xi,1)=0, \tag{17b}$$

where $\tilde{F}(\tilde{\phi})=F(2\phi_o\tilde{\phi}-\phi_o)$, and $\sigma=\nu/\alpha$ is the ratio of momentum and scalar diffusivities. We see that $\theta_w(\xi)$ vanishes at $\xi=0$, and approaches unity as $\xi\to\infty$. The parameter b characterizes the sharpness of the transition of the scalar distribution on the wall at $\xi=1$. We represent the solution of $\theta(\xi,\tilde{\phi})$ as

$$\theta(\xi,\tilde{\phi})=\sum_{n=1}^{\infty} a_n(\xi)\sin n\pi\tilde{\phi}+g(\xi,\tilde{\phi}), \tag{18}$$

where g will be chosen to satisfy the boundary conditions (17a) and (17b), and each term in the summation will satisfy homogeneous boundary conditions on the walls.

If $g(\xi,\tilde{\phi})$ and the Fourier expansion each satisfy the scalar transport equation at the walls, the scalar distribution θ will be sufficiently differentiable there. Since $\tilde{F}=0$ at the walls, we require

$$\left[\frac{1}{\xi}\frac{\partial}{\partial\xi}\left(\xi\frac{\partial g}{\partial\xi}\right)+\frac{1}{4\phi_o^2\xi^2}\frac{\partial^2 g}{\partial\tilde{\phi}^2}\right]\Bigg|_{\tilde{\phi}=0,1}=0. \tag{19}$$

Setting

$$g(\xi,\tilde{\phi})=g_0(\xi)+\tilde{\phi}g_1(\xi)+\tilde{\phi}^2 g_2(\xi)+\tilde{\phi}^3 g_3(\xi), \tag{20}$$

one can show that

$$g_0(\xi)=\theta_w(\xi), \tag{21a}$$

$$g_2(\xi)=\frac{2\phi_o^2 b}{1+\tanh b}\,\xi\,\mathrm{sech}^2\, b(\xi-1)\times[2b\xi\tanh b(\xi-1)-1], \tag{21b}$$

$$g_3(\xi)=\tfrac{1}{3}g_2(\xi), \tag{21c}$$

$$g_1(\xi)=-\theta_w(\xi)-\tfrac{4}{3}g_2(\xi). \tag{21d}$$

In light of (18), (16) becomes

$$\sum_{n=1}^{\infty}\left[a_n''+\left(\frac{1-\tilde{F}(\tilde{\phi})\sigma}{\xi}\right)a_n'-\left(\frac{n\pi}{2\phi_o}\right)^2\frac{1}{\xi^2}a_n\right]\sin n\pi\tilde{\phi}$$
$$=\sigma\frac{\tilde{F}(\tilde{\phi})}{\xi}\frac{\partial g}{\partial\xi}-\frac{1}{\xi}\frac{\partial}{\partial\xi}\left(\xi\frac{\partial g}{\partial\xi}\right)-\frac{1}{4\phi_o^2\xi^2}\frac{\partial^2 g}{\partial\tilde{\phi}^2}, \tag{22}$$

where primes denote differentiation with respect to ξ. Multiplying by $\sin m\pi\tilde{\phi}$, integrating from $\tilde{\phi}=0$ to $\tilde{\phi}=1$, rearranging terms, and approximating the radial derivatives of each function a_n by second-order central differences, we obtain

$$\left(1+\frac{\Delta\xi}{2\xi_j}\right)a_n^{j+1}-\left[2+\left(\frac{n\pi}{2\phi_o}\right)^2\frac{(\Delta\xi)^2}{\xi_j^2}\right]a_n^j+\left(1-\frac{\Delta\xi}{2\xi_j}\right)a_n^{j-1}$$
$$-\frac{\sigma\,\Delta\xi}{\xi_j}\sum_{m=1}^{N}(a_m^{j+1}-a_m^{j-1})A_{m,n}$$
$$=2(\Delta\xi)^2\left(-B_n(\xi_j)+\frac{\sigma}{\xi_j}C_n(\xi_j)\right),\quad 1\le n\le N, \tag{23}$$

where $\Delta\xi=\xi_{j+1}-\xi_j$, $a_n(\xi_j)$ is approximated by a_n^j, and

$$A_{m,n}=\int_0^1\tilde{F}(\tilde{\phi})\sin n\pi\tilde{\phi}\,\sin m\pi\tilde{\phi}\,d\tilde{\phi}, \tag{24a}$$

$$B_n(\xi_j)=\int_0^1\left[\frac{1}{\xi}\frac{\partial}{\partial\xi}\left(\xi\frac{\partial g}{\partial\xi}\right)+\frac{1}{4\phi_o^2\xi^2}\frac{\partial^2 g}{\partial\tilde{\phi}^2}\right]\Bigg|_{\xi_j}\sin n\pi\tilde{\phi}\,d\tilde{\phi}=\frac{(-1)^{n+1}+1}{n\pi}\left[\frac{1}{\xi}\frac{d}{d\xi}\left(\xi\frac{dg_0}{d\xi}\right)\right]\Bigg|_{\xi_j}$$
$$+\frac{(-1)^{n+1}}{n\pi}\left[\frac{1}{\xi}\frac{d}{d\xi}\left(\xi\frac{dg_1}{d\xi}\right)\right]\Bigg|_{\xi_j}+\left(\frac{2[(-1)^n-1]}{n^3\pi^3}+\frac{(-1)^{n+1}}{n\pi}\right)\left[\frac{1}{\xi}\frac{d}{d\xi}\left(\xi\frac{dg_2}{d\xi}\right)\right]\Bigg|_{\xi_j}$$
$$+\frac{1}{2\phi_o^2\xi_j^2}\left(\frac{(-1)^{n+1}+1}{n\pi}\right)g_2(\xi_j)+\left(\frac{6(-1)^n}{n^3\pi^3}+\frac{(-1)^{n+1}}{n\pi}\right)\left[\frac{1}{\xi}\frac{d}{d\xi}\left(\xi\frac{dg_3}{d\xi}\right)\right]\Bigg|_{\xi_j}+\frac{3}{2\phi_o^2\xi_j^2}\frac{(-1)^{n+1}}{n\pi}g_3(\xi_j), \tag{24b}$$

and

$$C_n(\xi)=\int_0^1\frac{\partial g}{\partial\xi}\bigg|_{\xi_j}\tilde{F}(\tilde{\phi})\sin n\pi\tilde{\phi}\,d\tilde{\phi}$$
$$=\frac{dg_0}{d\xi}\bigg|_{\xi_j}\int_0^1\tilde{F}(\tilde{\phi})\sin n\pi\tilde{\phi}\,d\tilde{\phi}$$
$$+\frac{dg_1}{d\xi}\bigg|_{\xi_j}\int_0^1\tilde{\phi}\tilde{F}(\tilde{\phi})\sin n\pi\tilde{\phi}\,d\tilde{\phi}$$
$$+\frac{dg_2}{d\xi}\bigg|_{\xi_j}\int_0^1\tilde{\phi}^2\tilde{F}(\tilde{\phi})\sin n\pi\tilde{\phi}\,d\tilde{\phi}$$
$$+\frac{dg_3}{d\xi}\bigg|_{\xi_j}\int_0^1\tilde{\phi}^3\tilde{F}(\tilde{\phi})\sin n\pi\tilde{\phi}\,d\tilde{\phi}, \tag{24c}$$

with $a_n^0=a_n^J=0$. The infinite series in (23) is then truncated to N terms. We choose $\xi_J=\xi_\infty$ to be large enough so that the scalar field at ξ_∞ is approximately linear in $\tilde{\phi}$.

This system of linear algebraic equations for the coefficients a_n^j was solved using standard numerical linear algebra software. The scalar distribution θ at a given point is determined from (18) using a finite number of terms and interpolation between the discrete values of ξ at which the coefficient functions a_n have been approximated.

IV. EXTRACTION OF VELOCITY FIELD FROM THE COMPUTED SCALAR FIELD

The final step is to calculate the velocity by the method of characteristics. The nondimensional forms of (12a) and (12b) are

$$-\theta_y u_x + \theta_x u_y + \left(\theta_{xy} - \frac{\theta_x \theta_{yy}}{\theta_y}\right) u = \frac{1}{\mathrm{Pe}}\left((\nabla^2\theta)_y - \frac{(\nabla^2\theta)\theta_{yy}}{\theta_y}\right), \tag{25a}$$

$$-\theta_y v_x + \theta_x v_y - \left(\theta_{xy} - \frac{\theta_y \theta_{xx}}{\theta_x}\right) v = -\frac{1}{\mathrm{Pe}}\left[(\nabla^2\theta)_x - \frac{(\nabla^2\theta)\theta_{xx}}{\theta_x}\right], \tag{25b}$$

where $\mathrm{Pe}=\sigma \mathrm{Re}$ is the Péclet number. To integrate these along a characteristic we rewrite (25a) and (25b) as

$$\frac{dx}{ds} = -\theta_y, \tag{26a}$$

$$\frac{dy}{ds} = +\theta_x, \tag{26b}$$

$$\frac{du}{ds} = -u\left(\theta_{xy} - \frac{\theta_x\theta_{yy}}{\theta_y}\right) + \frac{1}{\mathrm{Pe}}\left((\nabla^2\theta)_y - \frac{(\nabla^2\theta)\theta_{yy}}{\theta_y}\right), \tag{26c}$$

$$\frac{dv}{ds} = v\left(\theta_{xy} - \frac{\theta_y\theta_{xx}}{\theta_x}\right) - \frac{1}{\mathrm{Pe}}\left((\nabla^2\theta)_x - \frac{(\nabla^2\theta)\theta_{xx}}{\theta_x}\right). \tag{26d}$$

Equations (26a)–(26d) are integrated along characteristics by a Runge–Kutta method with constant step size. The x and y derivatives can be obtained from ξ and ϕ derivatives by the chain rule.

Recalling that

$$\theta_N(\xi,\tilde{\phi}) = \sum_{n=1}^{N} a_n(\xi)\sin n\pi\tilde{\phi} + g(\xi,\tilde{\phi}), \tag{27}$$

we locally approximate the coefficient functions $a_n(\xi)$ and their first three radial derivatives by fourth-degree Lagrange polynomials fitting the values a_n^j. To illustrate this, consider integration along a characteristic through a point (x_o,y_o), as shown in Fig. 2. The radial grid line closest to (x_o,y_o) is j_3, and a Lagrange polynomial is fitted through the five radial values ξ_j $(j=j_1,j_2,j_3,j_4,j_5)$ to approximate $a_n(\xi)$ at (x_o,y_o). Radial derivatives of $a_n(\xi)$ are approximated by derivatives of the polynomial. Computation of derivatives by this method provides a good test of the efficacy of "singularity removal" by the procedure discussed at the end of Sec. II. However, this method for obtaining the necessary derivatives of the scalar provides more accurate information than one would expect in experiment, since the azimuthal derivatives are computed from an analytical expression rather than by numerical differentiation of spatially discrete data. For this reason, the scalar was also evaluated at locations on a rectangular grid using (27), as discussed in Sec. V A. The derivatives of θ found from the chain rule were then computed by second-order accurate central difference approximations, or by second-order accurate lopsided or one-sided difference approximations where necessary.

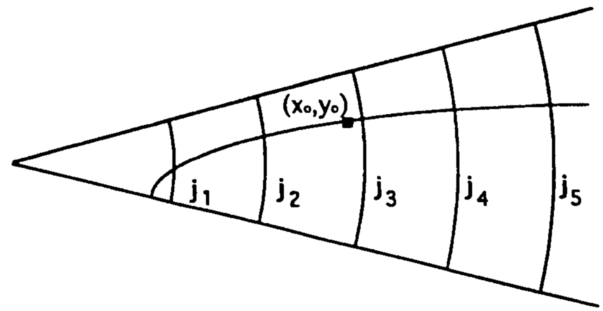

FIG. 2. Relationship of radial grid lines to a characteristic curve.

The integration is initiated at a number of points along the $\phi=-\phi_o$ wall and continued until the characteristic curves leave the computational domain at ξ_∞. This provides a velocity field at points on the characteristics (cf. Fig. 3). Velocity profiles at a given x position can be obtained from the value on each characteristic crossing a constant-x line.

As discussed in Sec. II and in more detail in Ref. 5, the vanishing of either component of $\nabla\theta$ leads to an apparent singularity. In the present case, this corresponds to the vanishing of $\partial\theta/\partial x$ ($\partial\theta/\partial y$ is nonzero throughout the domain). We have examined four ways to extract u and v.

(1) Extract u and v from (25a) and (25b), except when use of a dimensionless scalar transport equation analogous to (3) is necessary to pass over an apparent singularity in (25b), as discussed in Sec. II.

(2) Extract u from (25a) and compute v from the analog of (3).

(3) Extract v from (25b) and compute u from the analog of (3), except at points where $\partial\theta/\partial x$ vanishes, where the extraction proceeds according to method (2).

(4) Extract one velocity component using the hyperbolic equation (25a) or (25b) in which the denominator of the second term in the second factor on the right-hand side has

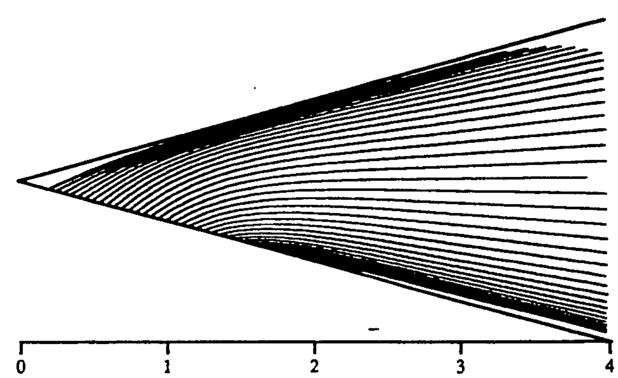

FIG. 3. Characteristics for Eqs. (25a) and (25b). Here Re=2, σ=7, b=2, J=40, N=12, $\Delta s=10^{-2}$.

the largest magnitude. The other velocity component is obtained from the analog of (3). As discussed in Sec. V A, we expect the fourth approach to extract the most accurate velocity field.

Several noise processes can corrupt optical measurements of the scalar, and can be expected to lead to errors in the extracted velocity field. Here, we consider multiplicative noise corrupting M measurements of a steady scalar field on a rectangular grid, so that each measured field is related to the actual field by

$$\theta^{(m)}_{\text{detected}}(x_i,y_j)=\theta_{\text{actual}}(x_i,y_j)[1+\eta D^{(m)}(x_i,y_j)] \quad (1\leq m\leq M), \tag{28}$$

where η is the maximum amplitude of the noise and D is a random number with $|D|\leq 1$ at each point on the grid. If the temperature field were computed M times and then averaged, we would obtain

$$\langle \theta^{(m)}_{\text{detected}}(x_i,y_j)\rangle=\langle \theta_{\text{actual}}(x_i,y_j)\rangle + \eta\langle D^{(m)}(x_i,y_j)\,\theta_{\text{actual}}(x_i,y_j)\rangle, \tag{29}$$

where $\langle\ \rangle$ indicates an ensemble average. For large M, (29) reduces to

$$\langle \theta^{(m)}_{\text{detected}}(x_i,y_j)\rangle\approx\theta_{\text{actual}}(x_i,y_j), \tag{30}$$

since $\langle D\rangle\rightarrow 0$ for a sufficiently large number of realizations. An alternative approach would be to extract the velocity from each of the M computed scalar fields and then average the extracted velocities. Besides being more cumbersome, this method typically yields the wrong result (even in the limit $M\rightarrow\infty$) for nonzero η, since the noise appears nonlinearly in the hyperbolic equations for u and v.

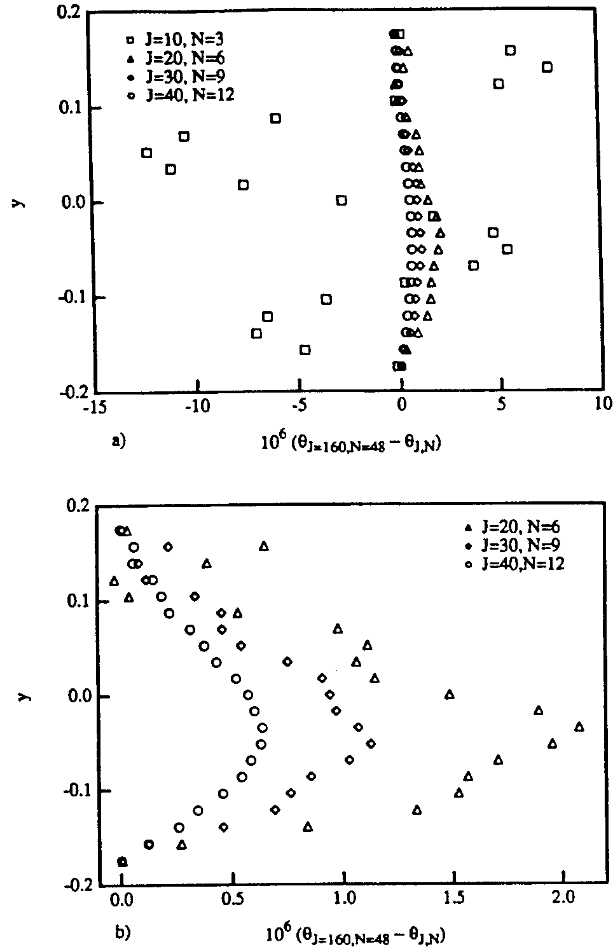

FIG. 4. Absolute differences at $x=2.0$ between the scalar field computed with $J=160$, $N=48$, and those computed with less accuracy. (a) $(J,N)=(10,3)$, $(20,6)$, $(30,9)$, $(40,12)$. (b) $(J,N)=(20,6)$, $(30,9)$, $(40,12)$.

V. RESULTS

Before considering the extracted velocity fields, we first show that the scalar field computation converges as J and N increase. For all results shown here, Re=2, $\sigma=7$ (corresponding to the Prandtl number for water), and $b=2$. Figure 4 shows the absolute difference on the line $x=2$ between θ calculated for $J=160$ and $N=48$ and θ calculated at lower resolutions. The calculation of the scalar (hereinafter referred to as temperature) distribution converges rapidly, and the absolute difference between $\theta_{J=160,N=48}$ and $\theta_{J=40,N=12}$ is less than 10^{-6}. Therefore, we will consider the "exact" temperature field to be converged when computed with 40 radial grid points and 12 azimuthal modes.

Most present optical detectors utilize identical rectangular (usually square) elements in a regular array, with implicit signal averaging over the area of each element. Thus, the temperature data will normally present themselves on a rectangular grid, uniformly spaced in both Cartesian directions. Since in our simulation computation of θ is not performed on a rectangular grid, interpolation is required to effect a rectangular presentation of the data. This constitutes an additional source of error in our simulated extraction of **u**.

Extraction of **u** from data on a rectangular grid requires approximation (e.g., by finite differences) of seven x and y derivatives of θ. The magnitude of the error introduced into **u** will depend on the noise level, grid size, the difference schemes used to approximate the derivatives, and the algorithm and step size used to integrate the hyperbolic equations along the characteristics.

A. Noise-free results

We first illustrate the extraction of **u** and the nature of the apparent singularity and its removal, in the absence of noise in θ ($\eta=0$).

We begin by extracting **u** using azimuthal and radial derivatives of θ and the chain rule to approximate x and y derivatives. We use (27) to evaluate the azimuthal (ϕ) derivatives of θ, and fourth-degree Lagrange polynomials to evaluate radial derivatives at points *on the characteristics*. This allows us to extract **u** with less error in approximating the derivatives of θ than would be incurred if interpolated values of θ were numerically differentiated on a rectangular grid.

Integrating the hyperbolic equations for u and v without removing the singularity by the methods discussed in Sec. II and Sec. IV, we obtain the results shown in Fig. 5. The integration step size is $\Delta s=10^{-2}$. Points in Figs. 5(a) and 5(b) represent values of the x- and y-velocity components, respectively, on characteristics crossing the line $x=1.8$. Fig-

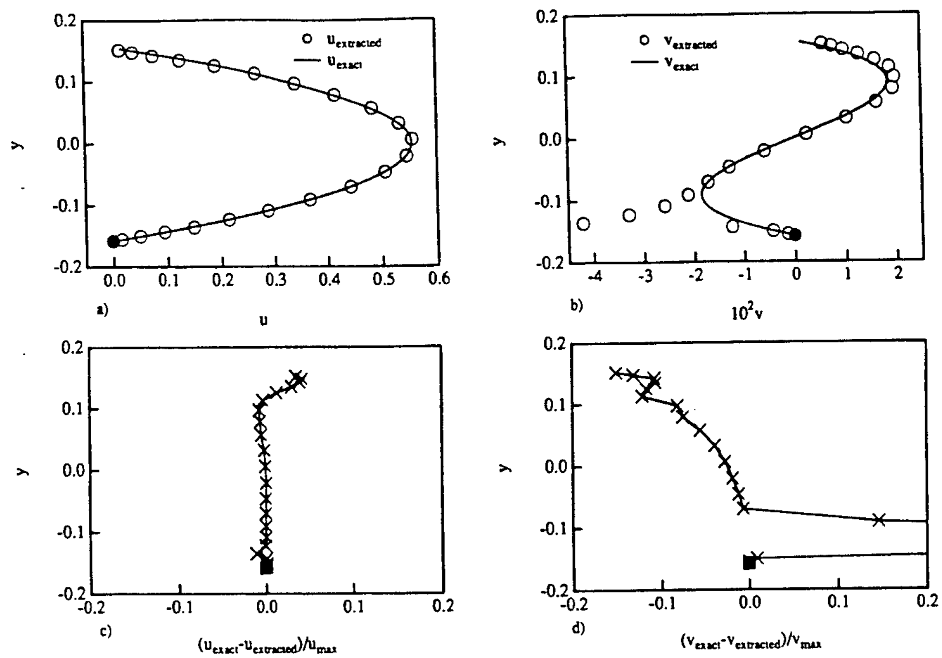

FIG. 5. Here $x=1.8$, Re=2, $\sigma=7$, $b=2$, $J=40$, $N=12$, and $\Delta s=10^{-2}$. Here u and v extracted directly from hyperbolic equations. Filled symbols denote the boundary data. (a) Extracted (○) and exact (—) x component; (b) y component of velocity; (c) relative error in the x component; (d) y component of velocity.

ures 5(c) and 5(d) the relative errors in u and v, respectively. At $x=1.8$, the relative error in v is much larger near the nonisothermal wall than near the isothermal one. The characteristic curves on which the error in v is large are those that have crossed the curve $\partial\theta/\partial x=0$ shown in Fig. 6. (Note that $\partial\theta/\partial y<0$ everywhere in the domain.) This is what is expected for extraction without removal of the singularity in (25b). By integrating the hyperbolic equation for u and computing v from the energy equation, one obtains the results shown in Fig. 7, in which the relative errors in v are much lower than those shown in Fig. 5.

The relative error for u in Fig. 7(c) increases to approximately 5% on the characteristics closest to the isothermal wall. The values of v on these characteristics, computed from u and the energy equation, are in error by as much as 10%. To understand the source of this error, consider again the characteristic curves shown in Fig. 3. The characteristics lying close to the isothermal wall must traverse a greater distance to reach $x=1.8$, but the integration is stable and errors do not grow rapidly along the integration path. We believe that the larger errors are due to errors in approximating radial derivatives of temperature in the upstream region. Finally, we note that the relative error in v is approximately twice that in u. This is not surprising since v depends on a previous computation of u.

To more faithfully mimic extraction of **u** from experimental data, the temperature was computed at locations on a rectangular grid using (27), and its derivatives were approximated by finite differences. The resolution of the grid is denoted by (I_g, J_g), where I_g and J_g are the number of x and y grid points, respectively. With $I_g=215$ and $J_g=210$, we first investigate the effect of integration step size. Figures 8 and 9 show results for $\Delta s=10^{-1}$ and 10^{-2}, respectively. These results were obtained by integrating the hyperbolic equation for u and computing v from the energy equation. The relative errors in both extracted velocity components are much smaller for $\Delta s=10^{-2}$ than for $\Delta s=10^{-1}$. For $\Delta s=10^{-3}$ (not shown), the relative error decreases slightly at a few points near the isothermal wall but is otherwise unchanged. Decreasing Δs to 10^{-4} gives no further reduction in error. Therefore the remaining error is due to approximating the derivatives of θ by finite differences using θ values interpolated onto the points of a rectangular grid. Figures 7 and 9 show that the maximum error in u doubles from 5% to 10% and that for v doubles from 10% to 20% as a result of finite-difference approximation of the derivatives. Thus, the errors incurred by the integration (for $\Delta s=10^{-2}$) are much smaller than those due to approximation of temperature derivatives on a grid. Since the relative errors in the velocity compo-

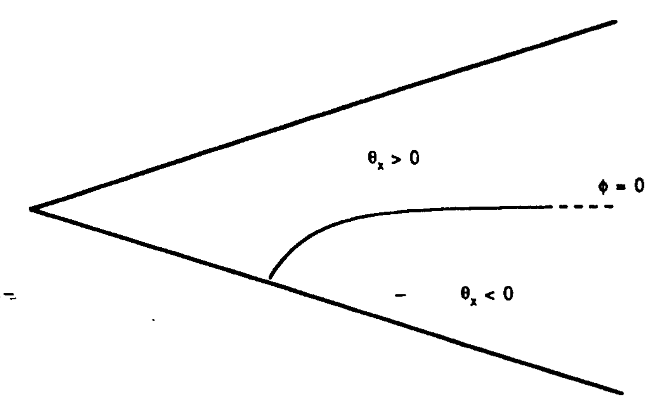

FIG. 6. Sign of θ_x for Jeffery–Hamel flow (Re=2, $\sigma=7$, $b=2$).

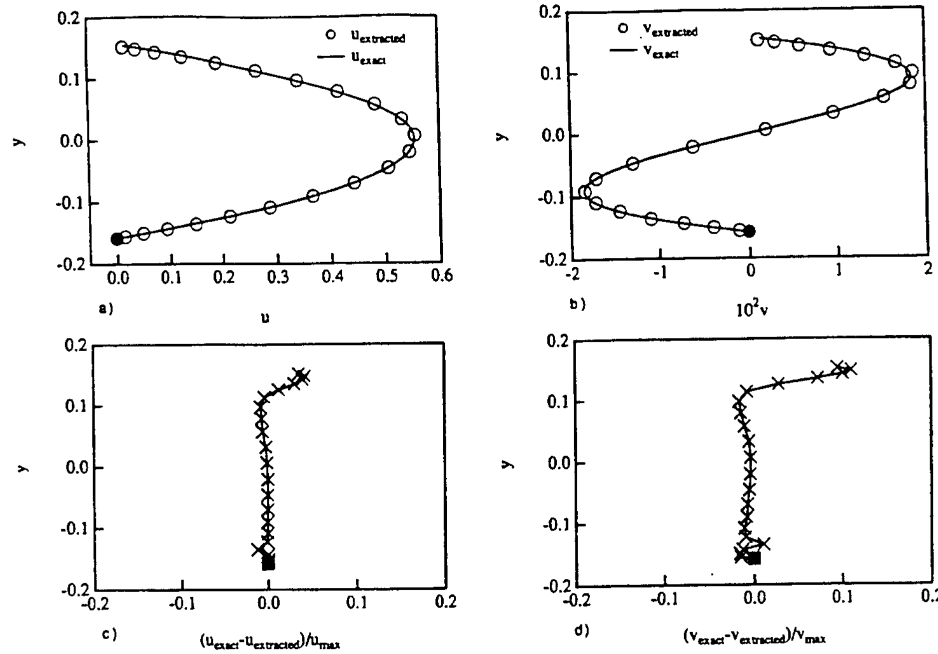

FIG. 7. Here $x=1.8$, Re=2, $\sigma=7$, $b=2$, $J=40$, $N=12$, and $\Delta s=10^{-2}$. Here, u is extracted directly from the hyperbolic equation; v is computed from u and the scalar transport equation. Filled symbols denote the boundary data. (a) Extracted (O) and exact (—) x component; (b) y component of velocity; (c) relative error in the x component; (d) y component of velocity.

nents are virtually the same for $\Delta s=10^{-2}$, 10^{-3}, and 10^{-4}, the remainder of the calculations will be performed with $\Delta s=10^{-2}$.

We next investigate the effects of spatial resolution and element shape on extraction of the velocity components. Fixing $\Delta s=10^{-2}$, $J=40$, and $N=12$ as discussed above, and considering grids with $(I_g, J_g)=(200,34)$, (300,52), (400,68), and (500,86), we obtain the results shown in Fig. 10. Again, these results are obtained by integrating the hyperbolic equation for u and computing v from the energy equation. Figures 10(a)(i,ii) show that the errors, especially near the isothermal wall, are very large for this coarse grid. When the resolution is increased to (300,52), the error decreases significantly [Figs. 10(b)(i,ii)] but is still quite large (>20%) at

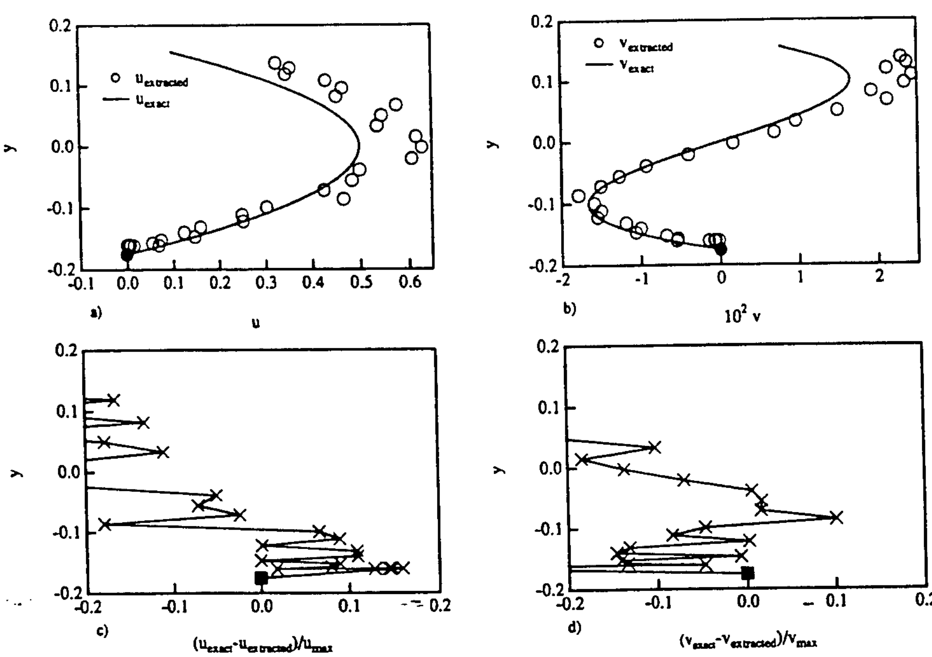

FIG. 8. Here $x=1.8$, Re=2, $\sigma=7$, $b=2$, $J=40$, $N=12$, $\Delta s=10^{-1}$, $I_g=215$, and $J_g=210$. Here, u is extracted directly from the hyperbolic equation; v is computed from u and scalar transport equation. Filled symbols denote the boundary data. (a) Extracted (O) and exact (—) x component; (b) y component of velocity; (c) relative error in the x component; (d) y component of velocity.

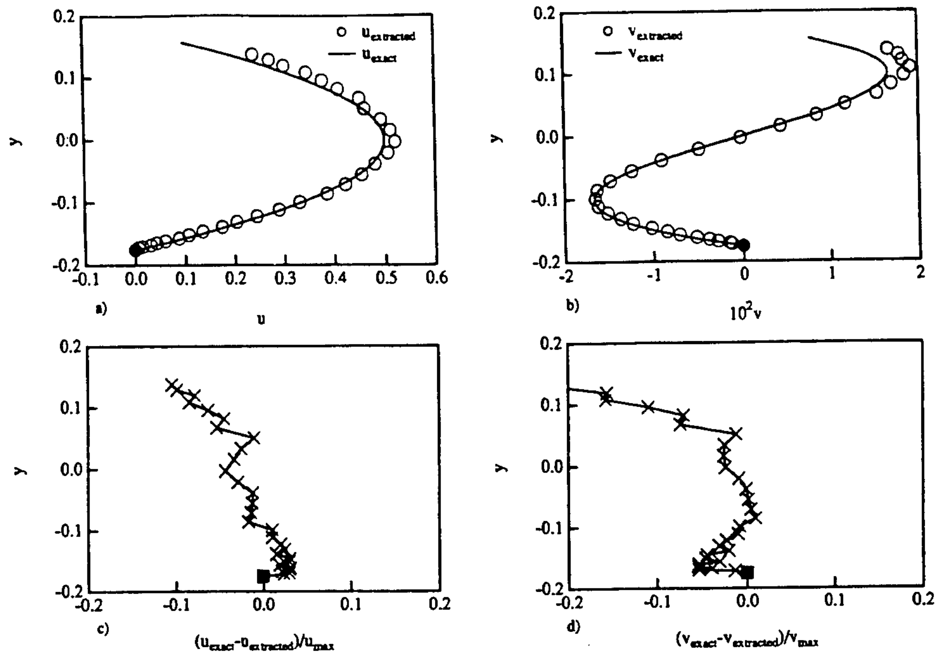

FIG. 9. Here $x=1.8$, $\mathrm{Re}=2$, $\sigma=7$, $b=2$, $J=40$, $N=12$, $\Delta s=10^{-2}$, $I_g=215$, and $J_g=210$. Here, u is extracted directly from the hyperbolic equation; v is computed from u and the scalar transport equation. Filled symbols denote the boundary data. (a) Extracted (○) and exact (—) x component; (b) y component of velocity; (c) relative error in the x component; (d) y component of velocity.

some points. When the resolution is increased further to (400,68) [Figs. 10(c)(i,ii)], the error is further reduced and exceeds 20% only near the isothermal wall. For (500,86) [Figs. 10(d)(i,ii)], the relative error in u is always less than 12%, while the error in v exceeds 10% only on the two characteristics closest to the isothermal wall. This is due to errors incurred in approximating temperature derivatives at and near the upstream origin of these characteristics. A locally refined grid could be used to reduce this source of error.

Comparison of Fig. 9 to Figs. 10(c) and 10(d) shows that the grid of nonsquare elements (Fig. 9) gives results with accuracy comparable to that of a grid of square elements [Fig. 10(c)] having about 40% fewer points (45 150 versus 27 200), and that the accuracy is inferior to that of a grid with a comparable number (43 000) of square elements [Fig. 10(d)].

Four approaches for extracting the velocity components from the hyperbolic equations, and possibly the energy equation, were discussed in Sec. IV. The numerical results described above were found with Approach 2. Setting $J=40$, $N=12$, and $\Delta s=10^{-2}$ as before and fixing $(I_g,J_g)=(400,68)$, we compare the accuracy of the four approaches. Approach 1 extracts u and v from the hyperbolic equations except when the energy equation is necessary to bypass a singularity. The x component is identical to that shown in Fig. 10(c)(i), and the y component is shown in Fig. 11(b). As discussed above, the results from Approach 2 are shown in Figs. 10(c)(i,ii). Those from Approach 3 (v extracted from the hyperbolic equation and u computed from the energy equation) are shown in Figs. 11(a) and 11(b). Results from Approach 4 [one velocity component extracted from the hyperbolic equation (12a) or (12b) having the larger denominator in the second term of the second factor on its right-hand side, with the other component computed from the energy equation] are identical to those shown for Approach 2 in Figs. 10(c)(i,ii). This is because $|\partial\theta/\partial y| > |\partial\theta/\theta x|$ throughout the domain, so that Approaches 2 and 4 are identical. Figure 11(b) shows that extraction of v from its hyperbolic equation results in large errors, since $|\partial\theta/\partial x|$ is small $[O(10^{-3})]$ throughout the domain, especially when close to (but not on) the curve $|\partial\theta/\partial x|=0$ shown in Fig. 6. Figure 11(b) shows that extraction of u from the energy equation using values of v computed from its hyperbolic equation results in large errors in u also. Therefore the most accurate method for extracting the velocity is Approach 4 (identical to Approach 2 for this particular flow). All subsequent results are obtained using Approach 2.

B. Results with noise

To investigate the effects of noise on the error in the velocity extraction, the temperature field was corrupted by multiplicative noise at the grid points. The most accurate temperature field available, $J=160$, $N=12$, with an integration step size $\Delta s=10^{-2}$, was used for all the following results, unless otherwise specified. For noise amplitudes $\eta=10^{-4}$, 10^{-6}, and 10^{-8}, Figs. 12(a)(i,ii), 12(b)(i,ii), and 12(c)(i,ii), respectively, show the extracted velocity for a single realization ($M=1$), with a grid composed of nonsquare elements ($I_g=215$, $J_g=210$). For $\eta=10^{-4}$, the errors in both velocity components are large (>100%). When η is reduced to 10^{-6}, the errors in both velocity components are still large, but significantly smaller than for $\eta=10^{-4}$. When η is reduced to 10^{-8}, the errors incurred due to noise are insignificant, and the results are similar to the noise-free case.

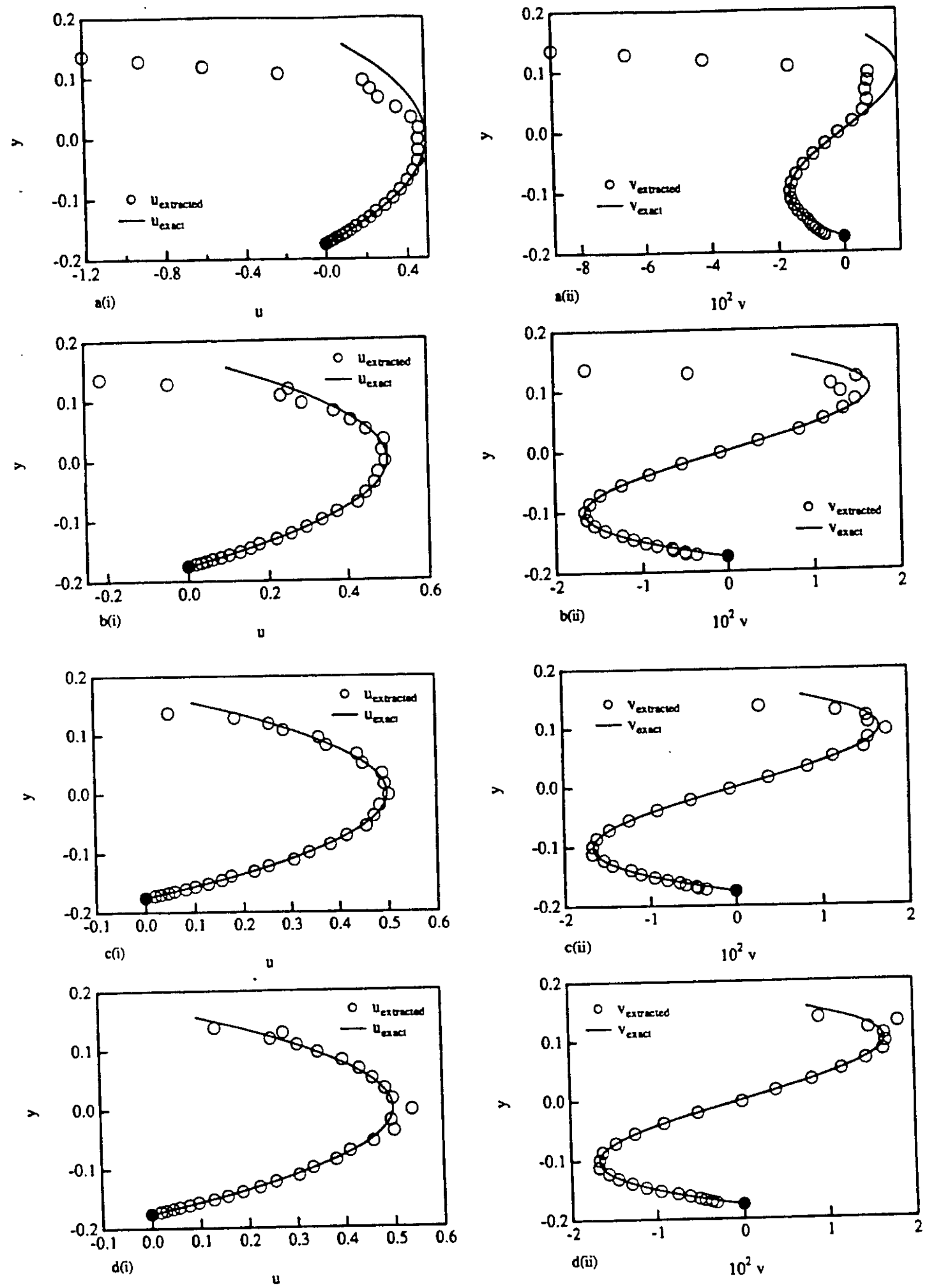

FIG. 10. Here $x=1.8$, Re$=2$, $\sigma=7$, $b=2$, $J=40$, $N=12$, and $\Delta s=10^{-2}$; extracted (O) and exact (—) components. Here, u is extracted directly from the hyperbolic equation; v is computed from u and the scalar transport equation. Filled symbols denote the boundary data. (a)(i) x component; (a)(ii) y component, for $I_g=200$, $J_g=34$; (b)(i) x component; (b)(ii) y component, for $I_g=300$, $J_g=52$; (c)(i) x component; (c)(ii) y component, for $I_g=400$, $J_g=68$; (d)(i) x component; (d)(ii) y component, for $I_g=500$, $J_g=86$.

If a grid with square elements ($I_g=400$, $J_g=68$) is used instead, the results shown in Figs. 13(a) and 13(b) are obtained for $\eta=10^{-4}$ and 10^{-6}, respectively. (In what follows, both u and v were computed by Approach 2, as discussed above, and reference is made to results for both components, although results are shown only for u.) For $\eta=10^{-4}$, the errors in both velocity components are large, but smaller than those shown in Figs. 12(a)(i,ii) for nonsquare elements. When η is reduced to 10^{-6}, the errors in both u and v are similar to those with no noise. For $\eta=10^{-8}$, the results are indistinguishable from those with no noise, and nearly indistinguishable from those with $\eta=10^{-6}$. Comparison of the noise-free results for nonsquare and square elements [Figs. 9(a) and 10(c)(i), respectively] to the corresponding results for $\eta=10^{-6}$ [Figs. 12(b)(i) and 13(b)] indicates that extraction of the velocity components is less susceptible to noise-related errors when the elements of the grid are square. It was noted earlier that $|\partial\theta/\partial y|>|\partial\theta/\partial x|$ throughout the domain, so that we extract u from its hyperbolic equation and compute v from the energy equation. Since $\partial\theta/\partial y$ appears in the denominators of two terms in the hyperbolic equation for u, errors in approximating $\partial\theta/\partial y$ will be exacerbated relative

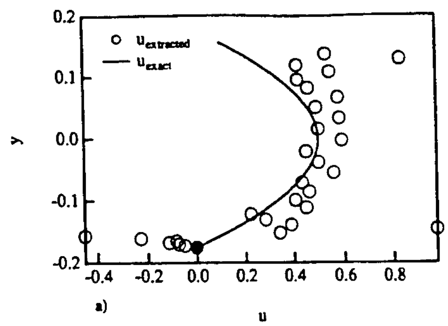

FIG. 11. Here $x=1.8$, $\mathrm{Re}=2$, $\sigma=7$, $b=2$, $J=40$, $N=12$, $\Delta s=10^{-2}$, $I_g=400$, and $J_g=68$. Here, v is extracted directly from the hyperbolic equation; u is computed from v and the scalar transport equation. Filled symbols denote the boundary data. (a) Extracted (O) and exact (—) x component; (b) y component of velocity.

to those for $\partial\theta/\partial x$ as $\Delta y/\Delta x$ decreases, since θ is corrupted by noise at each grid point.

We can reduce the effect of noise on the extracted velocity by averaging, as discussed in Sec. IV. First, we consider extraction of u and v from each of M noisy temperature fields, followed by averaging the velocity fields. By fixing $I_g=400$, $J_g=68$, $\Delta s=10^{-2}$, and the noise amplitude $\eta=10^{-6}$, and taking $M=1$, 100, we obtain the results shown in Figs. 14(a) and 14(b), respectively. (For these results, $J=40$ and $N=12$.) The noise has little effect on the error in u and v for $M=1$, or for $M=10$ or 30 (not shown, but indistinguishable from the $M=1$ case). However, for $M=100$, the effect is dramatic. Both extracted velocity components are significantly less accurate, and it is expected that

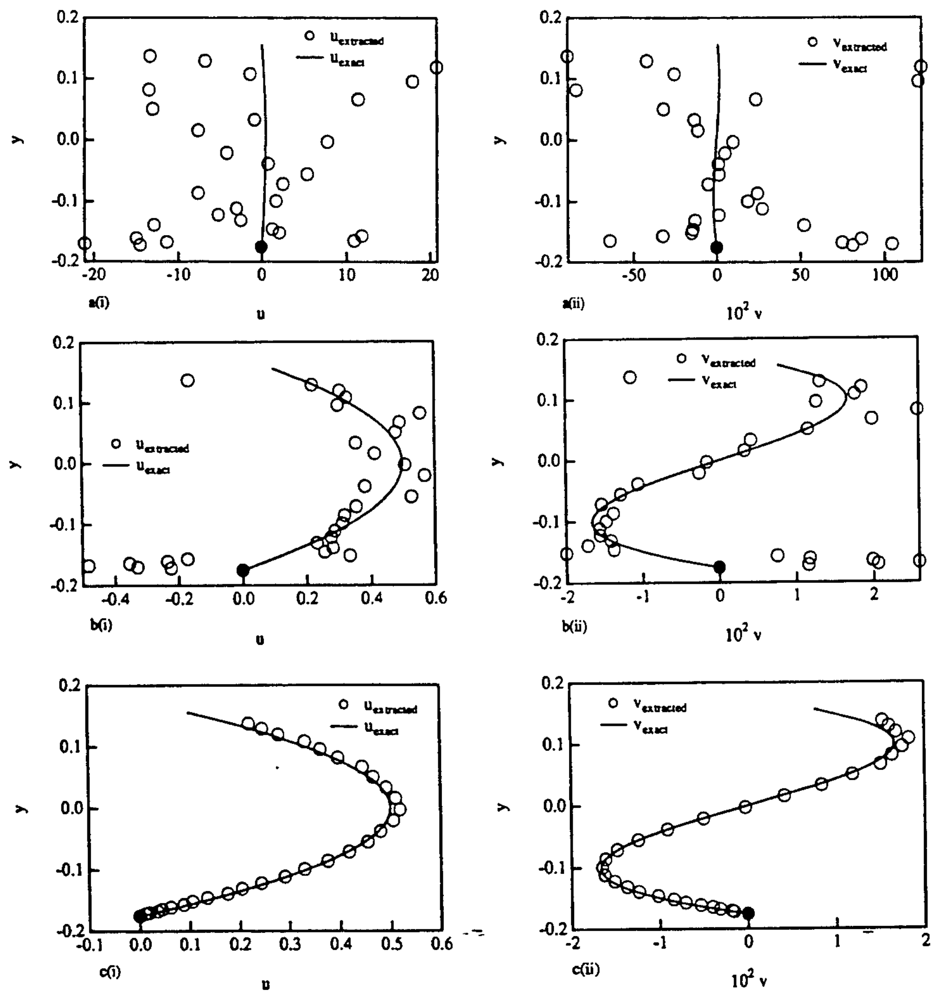

FIG. 12. $x=1.8$, $\mathrm{Re}=2$, $\sigma=7$, $b=2$, $J=160$, $N=48$, $\Delta s=10^{-2}$, $I_g=215$, $J_g=210$, $M=1$. (a)(i) x component; (a)(ii) y component for $\eta=10^{-4}$; (b)(i) x component; (b)(ii) y component for $\eta=10^{-6}$; (c)(i) x component; (c)(ii) y component for $\eta=10^{-8}$.

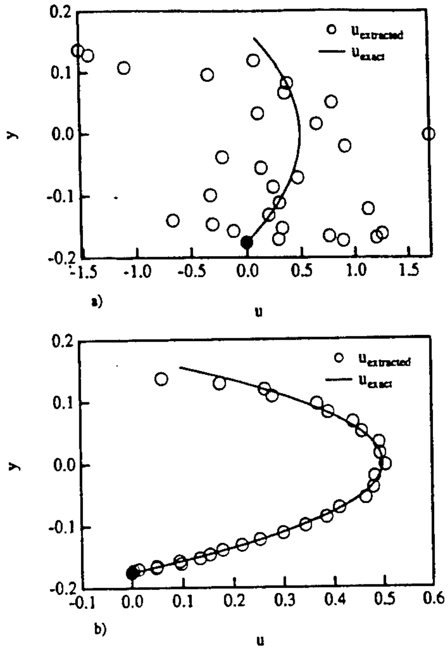

FIG. 13. Here $x=1.8$, $\mathrm{Re}=2$, $\sigma=7$, $b=2$, $J=160$, $N=48$, $\Delta s=10^{-2}$, $I_g=400$, $J_g=68$, and $M=1$. (a) x component of velocity for $\eta=10^{-4}$; (b) x component of velocity for $\eta=10^{-6}$.

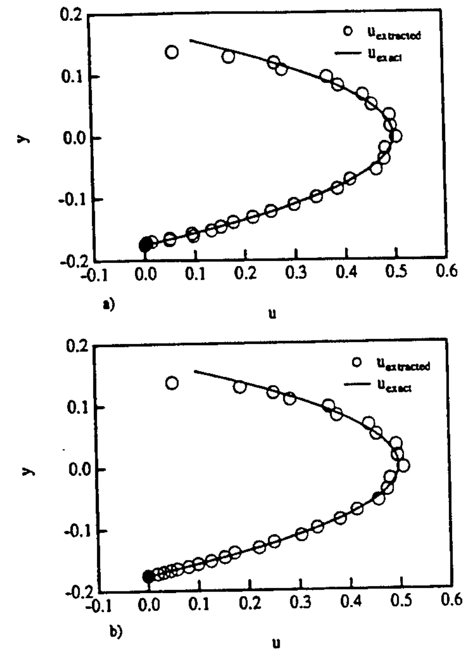

FIG. 15. Here $x=1.8$, $\mathrm{Re}=2$, $\sigma=7$, $b=2$, $J=160$, $N=48$, $\Delta s=10^{-2}$, $I_g=400$, $J_g=68$, and $\eta=10^{-6}$. Averaging takes place on the M temperature fields, then u and v are extracted. (a) $M=1$; (b) $M=100$.

for larger M, the results will be further degraded. This is because noise that corrupts quantities appearing nonlinearly in the hyperbolic equations can lead to large errors in the velocity extracted from individual corrupted scalar fields. Thus, the effects on the means of the extracted velocity components can be profound.

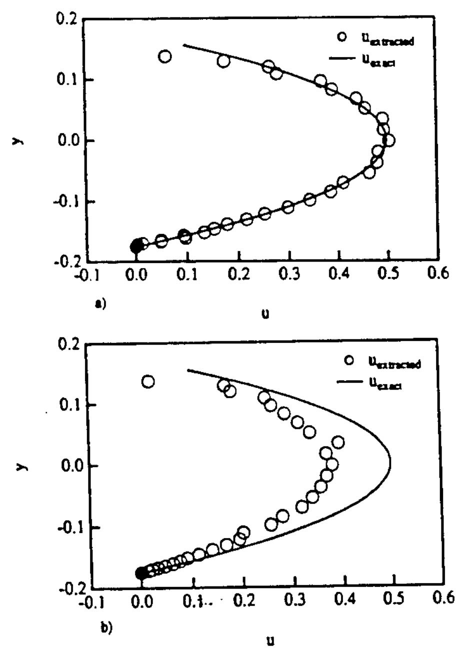

FIG. 14. $\mathrm{Re}=2$, $\sigma=7$, $b=2$, $J=160$, $N=48$, $\Delta s=10^{-2}$, $I_g=400$, $J_g=68$, and $\eta=10^{-6}$. Averaging takes place on the M velocity fields extracted from each of M noisy temperature fields. (a) $M=1$; (b) $M=100$.

A better method for ameliorating the effects of noise is to average M noisy temperature fields and to extract u and v from the mean temperature field. By fixing $I_g=400$, $J_g=68$, $\Delta s=10^{-2}$, and $\eta=10^{-6}$, and setting $M=1$ and 100, we obtain the results shown in Figs. 15(a) and 15(b), respectively. For $M=1$, the errors in both velocity components are similar to the noise-free case. When M is 10, 100, 1000, or 10 000, the results are indistinguishable from the noise-free case. Therefore, the noise in the temperature field has no effect on the extracted velocity for $M \geq 10$. Fixing $I_g=400$, $J_g=68$, $\Delta s=10^{-2}$, increasing η to 10^{-4}, and setting M to 10, 100, 1000, and 10 000, we obtain the results shown in Figs. 16(a), 16(b), 16(c), and 16(d), respectively. For $M=1$ [Fig. 13(a)], the errors in both velocity components are large. As M is increased, the errors decrease until for $M=10\,000$, the error due to noise is insignificant compared to those associated with discretization onto a grid. Therefore, the error due to noise can be reduced by averaging a sufficient number of temperature fields before extracting the velocity.

VI. DISCUSSION AND CONCLUSIONS

We have shown that it is possible to extract both velocity components from advected scalar data for a two-dimensional steady, solenoidal flow with appropriate scalar boundary conditions. The proof-of-concept study reported here shows that we can deal with singularity in the equations from which the velocity is extracted, as indicated in Ref. 5.

From a numerical standpoint, the integration along the characteristic curves is stable, and a relatively large integration step size (100–1000 steps per characteristic) is sufficient. For the flow and boundary conditions considered,

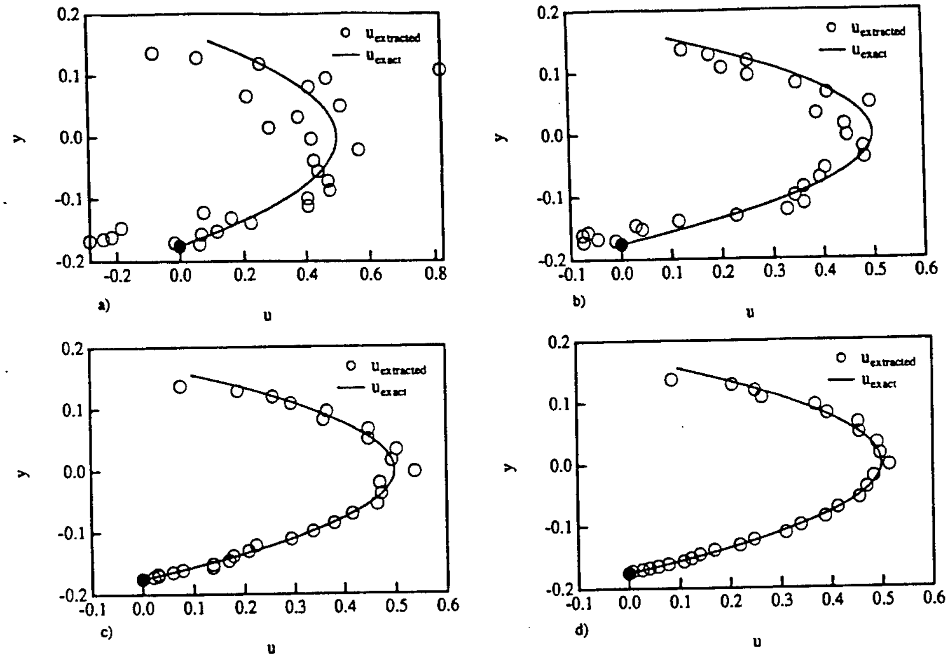

FIG. 16. Here $x=1.8$, $\mathrm{Re}=2$, $\sigma=7$, $b=2$, $J=160$, $N=48$, $\Delta s=10^{-2}$, $I_g=400$, $J_g=68$, and $\eta=10^{-4}$. Averaging takes place on the M temperature fields, then u and v are extracted. (a) $M=10$; (b) $M=100$; (c) $M=1000$; (d) $M=10\,000$.

knowledge of the scalar on a 400×68 grid is sufficient. Available CCD arrays can be used to obtain scalar data with this resolution.

We have introduced multiplicative noise and have shown how to deal with it by averaging noisy scalar fields. More sophisticated filtering and smoothing techniques may be more attractive at higher noise levels. The trade-off between the number of images that need to be processed (with simple averaging), the computational complexity (of concern as the sophistication of the image processing increases), and the cost of special purpose hardware (to reduce the CPU time associated with image processing) will determine the best approach in a given situation. We conjecture that additive noise will give qualitatively similar results at low noise levels.

The ultimate goal of this work is to develop a real-time version of this velocity extraction technique for use in flow control and state estimation. In unsteady three-dimensional flows, the issue of spatial resolution will be critical in determining temporal resolution when simultaneous volumetric data cannot be acquired, in which case two-dimensional images must be "scanned" in the third direction. The spatial and temporal resolution requirements will thus be important in determining framing and processing rate requirements for use of this technique in real-time applications.

ACKNOWLEDGMENTS

The authors acknowledge useful discussions with Professors K.-Y. Fung and N. M. Jokerst. Effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under Grants No. 90-0156 and No. F49620-95-1-0297. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes, notwithstanding any copyright notation thereon. This work was also supported in part by a U.S. Department of Energy Computational Science Graduate Fellowship and a Zonta International Amelia Earhart Fellowship Award. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

[1]W. J. A. Dahm, L. K. Su, and K. B. Southerland, "A scalar imaging velocimetry technique for fully resolved four-dimensional vector velocity field measurements in turbulent flows," Phys. Fluids A **4**, 2191 (1992).
[2]W. J. A. Dahm, "Micromeasurements of fluid turbulence," Bull. Am. Phys. Soc. **38**, 2255 (1993).
[3]L. K. Su and W. J. A. Dahm, "Scalar imaging velocimetry measurements of the dissipative scales of turbulent flows," Bull. Am. Phys. Soc. **38**, 2286 (1993).
[4]P. T. Tokumaru and P. E. Dimotakis, "Image correlation velocimetry," Exp. Fluids **19**, 1 (1995).
[5]A. J. Pearlstein and B. N. Carpenter, "On the determination of solenoidal or compressible velocity fields from measurements of passive or reactive scalars," Phys. Fluids **7**, 754 (1995).
[6]K. Millsaps and K. Pohlhausen, "Thermal distributions in Jeffery–Hamel flows between nonparallel plane walls," J. Aeronaut. Sci. **20**, 187 (1953).

Globalization of the Krawczyk Algorithm to Compute All Simple and Nonsimple Isolated Real Solutions of Systems of Polynomial Equations

by

David L. Cotrell

and

Arne J. Pearlstein

Department of Mechanical and Industrial Engineering

University of Illinois at Urbana-Champaign

1206 West Green Street

Urbana, IL 61801

Running Title: Globalization of the Krawczyk Algorithm

Address for Correspondence: Arne J. Pearlstein
Department of Mechanical and Industrial Engineering
University of Illinois at Urbana-Champaign
1206 West Green Street
Urbana, IL 61801
e-mail: ajp@uiuc.edu
(217) 244-6534 (fax)
(217) 333-3658 (voice)

AMS Subject Classifications: 65H10, 65H20

Keywords: polynomial equations systems, Krawczyk operator, global solution techniques

Abstract. We present a globalization of the Krawczyk algorithm to compute all real isolated solutions of systems of N real polynomial equations. This is accomplished by transforming the original system to an augmented system in R^{N+1}, in which each of the first N variables lies in the interval $[-1,1]$, and the $(N+1)$-th variable lies in $[0,\infty)$. The domain of this latter variable can be divided into two intervals $[0,1]$ and $[1,\infty)$, the second of which is mapped to $[0,1]$. Thus, the entire domain R^{N+1} can be examined by considering the finite domain $[-1,1]^N \times [0,1]$ for each of two systems. One of the augmented systems can have one or more solutions not corresponding to finite solutions of the original system. We use techniques from algebraic geometry to transform the $(N+1)$-th dimensional system so that spurious solutions are excluded, thus restricting the solutions to those corresponding to finite solutions of the original system. The algorithm requires bounds for multivariate polynomials over a finite domain. The best of three bounding methods considered uses an interval extension of the system, which is stored as a rooted, ordered tree, and is equivalent to an N-dimensional Horner scheme that takes advantage of polynomial sparsity.

We extend the approach to deal with systems having nonsimple solutions (i.e., with multiplicity greater than unity), at which the Jacobian vanishes. This is accomplished by constructing a new system that reduces by one the multiplicity of nonsimple solutions. Except in degenerate cases, this approach can be applied sequentially until no roots have multiplicity exceeding one.

1. Introduction. We present here a new computational technique for finding all real solutions in R^N of systems of polynomial equations

$$f_i(x_1, x_2, \cdots, x_N) = 0 \qquad 1 \le i \le N \quad . \tag{1}$$

To put the present work in context, we begin with some background.

Nonlinear equation systems such as (1) arise in many applications. For example, a fundamental problem in computer-aided geometric design is the efficient computation of all the real solutions of a system of multivariate polynomial equations within a given domain. Such equation systems can arise from the need to find all points of intersection between pairs of curves, and curves and surfaces, as well as points lying on curves corresponding to the intersection of two surfaces [1]. Nonlinear equation systems also arise in computer-aided process design where large equation systems are used in, for example, steady-state process flowsheet modeling [2-3]. Another important application is the solution of nonlinear systems arising from spatial discretization of boundary value problems having polynomial nonlinearities, such as the Navier-Stokes equations.

Solution techniques for nonlinear equation systems divide into two broad classes. The first is the class of local methods, where one seeks a solution by starting with an initial iterate. These methods include single- and multi-point iteration techniques (e.g., Newton-Raphson, Muller's method). The second class of methods systematically attempts to find all solutions in a domain. These methods include homotopy and subdivision (e.g., bisection) techniques, as well as techniques that reduce the dimension N to unity (e.g., methods using resultants, Gröbner bases). If enough information is available to select a good initial iterate, single point iteration schemes can find single solutions well. If the initial iterate is within the region of convergence of (1), the method will converge. If, on the other hand the initial iterate is outside the region of convergence, the method might converge to another solution or not converge at all.

Algebraic schemes can be used to reduce solution of (1) to finding zeros of a univariate polynomial. The main approaches use elimination [4-5] or the Gröbner transformation [6-7]. The advantage of these methods is that computation of the solutions of a univariate polynomial equation is a standard procedure, so that these techniques are capable, in principle, of finding all solutions in C^N. The manipulative burden of reducing an N-dimensional system to a lower-dimensional system (typically one dimension) can be significantly diminished by symbolic mathematical programs. The disadvantages are that these methods suffer from numerical instability and undesirable cost growth (e.g., memory and time) as N increases.

Homotopy methods [8] embed (1) in a family of systems $\underline{H}(\underline{x},t)$. A frequently used embedding is the simple one-parameter linear homotopy $\underline{H}(\underline{x},t)=(1-t)\underline{G}(\underline{x})+t\underline{f}(\underline{x})$. The continuation parameter t connects the target system (1) to a system $\underline{G}(\underline{x})=0$, for which all solutions in C^N are known. An advantage of homotopy approaches is that they allow all solutions (real and complex) of (1) to be found. However, since the starting points of real solutions cannot be identified in advance, all solution paths must be followed in order to compute all the real solutions. For polynomial systems, the number of paths is typically equal to the total degree.

The class of global subdivision techniques includes interval methods. These methods find all solutions of (1) in a finite domain. As discussed by Moore and Jones ([9], p. 1056), three possibilities arise when examining such a finite domain. One is sometimes able to show that no solution exists, or that at least one solution exists. The third possibility is that no such conclusion is possible. In the latter case, bisection is used to subdivide the given domain until each subdomain is shown to either contain or not contain a solution. A key advantage of this technique is that it finds all real solutions in a finite domain without computing the complex solutions. Interval methods that take advantage of large-grained parallelism now provide an efficient approach to compute all solutions in a finite domain for application problems of small and intermediate dimension (cf. [10-13]).

In this paper, we show how a technique that uses interval arithmetic and a contraction mapping algorithm [9,14-15] to find all solutions of (1) in a finite domain can be generalized to find all solutions of (1) in R^N. The generalization to unbounded domains is significant in that it lifts the requirement that bounds be established for the solution(s), frequently by techniques that are *ad hoc* or problem-dependent.

2. Development of a global method. In this section, we introduce notation, review a few fundamentals of interval analysis, and introduce the Krawczyk operator. We then proceed to globalize the latter, present the basic algorithm, and discuss how it deals with spurious solutions, solutions on or near the boundaries of intervals, and nonsimple solutions.

2.1. Background. Here, we establish notation and briefly review the basics of interval analysis and computation [16-17], the Krawczyk algorithm [17-18], and Moore's approach to computing real solutions of multivariate equations by bisection of interval vectors [9,17].

An interval X is a closed and bounded set of real numbers $[a,b]$. An interval vector $\underline{X}$ is an ordered n-tuple of intervals. (In this paper, scalars and vectors are denoted by Roman variables with no underscoring and single underscoring, respectively. Real matrices are represented by double-underscored Greek variables, with the exception of the N x N identity matrix, denoted by $\underline{\underline{I}}_N$. All other double-underscored Roman variables denote interval matrices.) Arithmetic operations on intervals are defined straightforwardly. For example, the product $Z = XY$ defines an interval containing the product xy for all $x \in X$ and all $y \in Y$. An interval extension F of f is an interval-valued function whose range $F(\underline{X})$ includes $f(\underline{x})$ for all $\underline{x} \in \underline{X}$. One way to obtain F is to replace the real variables in f by their corresponding interval variables. Such an F might also contain points that do not correspond to $f(\underline{x})$ for any $\underline{x} \in \underline{X}$. We denote the width of an interval X by $w(X) = b - a$. The midpoint of an interval is defined as $m(X) = (b+a)/2$.

The algorithm used by Moore and Jones [9] is based on the work of Krawczyk [18], who used the Brouwer fixed point theorem to develop an interval-based Newton-Kantorovich mapping. The Krawczyk operator is defined by

(2) $$\underline{K}(\underline{X}) = \underline{h} - \underline{\underline{\Phi}}\,\underline{f}(\underline{h}) + \left\{\underline{\underline{I}}_N - \underline{\underline{\Phi}}\,\underline{\underline{J}}(\underline{X})\right\}\left(\underline{X} - \underline{h}\right) \quad ,$$

where $\underline{\underline{\Phi}}$ is an arbitrary nonsingular real matrix and $\underline{h}$ is the real vector $m(\underline{X})$, while $\underline{\underline{I}}_N$ is the identity matrix and $\underline{\underline{J}}(\underline{X})$ is an interval extension of the Jacobian of the vector-valued function

$\underline{f}$. (Midpoints of interval vectors and matrices are defined as the midpoints of their elements.) We define $v(\underline{X}) = \prod_{i=1}^{N} w_i$ to be the volume of a rectangular portion of R^N.

2.2. Globalization. In order to globalize the Moore-Krawczyk algorithm [9], one needs to be able to examine the infinite domain R^N. With this in mind the polynomial system (1) is mapped from the original system (1) in R^N to an augmented system in R^{N+1} by the transformation

(3a)
$$y_{N+1} = \left[\sum_{i=1}^{N} x_i^2\right]^{-1/2}$$

(3b)
$$y_i = x_i y_{N+1} \qquad 1 \le i \le N \quad .$$

After multiplying each transformed equation by $y_{N+1}^{d_l}$, where d_l is the degree of the l-th equation, we get

(4a)
$$g_j(y_1, \cdots, y_{N+1}) = 0 \qquad 1 \le j \le N \quad ,$$

in which the term or terms of highest degree in each equation of system (1) are invariant. The (N+1)-th equation is

(4b)
$$g_{N+1}(y_1, \cdots, y_N) = \sum_{i=1}^{N} y_i^2 = 1 \quad .$$

Thus, the solutions of (4a,b) lie on a semi-infinite right circular cylinder of unit radius for $N = 2$, and on a semi-infinite unit hypercylinder for $N \ge 3$. The rectangular domain to be examined in order to find all solutions of (4a,b) is

(5)
$$y_i \in [-1,1] \quad , \qquad y_{N+1} \in [0,\infty) \quad , \qquad 1 \le i \le N \quad .$$

Thus, the transformation reduces the problem to one for which only one variable is unbounded. We note that if a null solution of (1) exists, it will not be a solution of (4). The existence of such a solution is easily verified.

The unbounded domain $[0,\infty)$ can be broken up into the intervals $[0,1]$ and $[1,\infty)$. Using the transformation

(6)
$$w_i = y_i \qquad 1 \le i \le N$$
$$w_{N+1} = 1/y_{N+1} \quad ,$$

the interval $[1,\infty)$ is mapped to $[0,1]$. This mapping gives rise to the system

(7) $$g_j^{*}(w_1,\cdots,w_{N+1})=0 \qquad 1\le j\le N+1 \quad .$$

Thus the entire domain R^{N+1} can be examined by considering a finite domain for each of the systems (4a,b) and (7).

2.3. Basic Algorithm. An algorithm implementing these ideas is represented by pseudocode in Figure 1. The algorithm examines finite domains $\underline{Y}$, and eliminates those that can be shown to not contain a solution. If the mapping (2) gives a contraction ($\underline{K}(\underline{Y})\subset\underline{Y}$), then $\underline{Y}$ contains a solution. In the event that no existence or nonexistence result is obtained, $\underline{Y}$ is bisected, as described in detail below, and added to a "stack" of finite domains to be examined. This process proceeds until the initial domain has been examined and all solutions have been found.

Examination of each finite domain begins with a nonexistence test [9], checking to see if

(8) $$0\notin G_j(\underline{Y})$$

is true for some j, $1\le j\le N+1$. If (8) is satisfied for some j, then $\underline{Y}$ contains no solutions of (4a,b), and $\underline{Y}$ is discarded. If (8) is false for each j, the possibility remains that $\underline{Y}$ contains a solution of (4a,b). In that case, a second, new nonexistence test (see Appendix) checks

(9) $$\left|g_j(m(\underline{Y}))\right| > \sum_{k=1}^{N+1}\left(\tfrac{1}{2}w(J_{jk})+\left|m(J_{jk})\right|\right)\tfrac{1}{2}w(Y_k)$$

for each j. If (9) is false for each j, we compute the mapping (2), using the choice

(10) $$\underline{\underline{\Phi}}^k=\begin{cases}\left[m(\underline{\underline{J}}(\underline{Y}^k))\right]^{-1} & \text{if } \left\|\underline{\underline{I}}_N-\underline{\underline{\Phi}}^k\,\underline{\underline{J}}(\underline{Y}^k)\right\|\le\left\|\underline{\underline{I}}_N-\underline{\underline{\Phi}}^{k-1}\,\underline{\underline{J}}(\underline{Y}^{k-1})\right\| \\ \underline{\underline{\Phi}}^{k-1} & \text{otherwise}\end{cases} \quad .$$

Otherwise, $\underline{Y}$ contains no solutions and is discarded. If $m(\underline{\underline{J}})$ is singular, we bisect the corresponding $\underline{Y}$, and place the resulting interval vectors on the stack. Otherwise, (2) is computed and a third nonexistence test is applied, which checks for emptiness of

(11) $$\underline{Z}=\underline{K}(\underline{Y})\cap\underline{Y} \quad .$$

If (11) is empty, then $\underline{Y}$ contains no solutions of (4a,b) and is discarded. If (11) is nonempty, we check to see if

(12) $$Y_j \subseteq K_j(\underline{Y})$$

is true for $1 \le j \le N+1$. If so, $\underline{Y}$ is bisected. Otherwise, an existence test which checks

(13) $$K_j(\underline{Y}) \subset Y_j$$

is applied. If (13) is true for $1 \le j \le N+1$, a contraction has taken place and $\underline{Y}$ is guaranteed to contain a solution of (4a,b). In that event, any point in $\underline{Y}$ will be a safe starting point (with guaranteed convergence) for use with interval Newton-Raphson [9]. Details of the convergence and solution acceptance are shown in Figure 1.

At this point, either a) a solution has been shown to exist in the current interval vector $\underline{Y}$, b) nonexistence tests have excluded a solution in $\underline{Y}$, or c) nonexistence tests have failed to exclude all or part of $\underline{Y}$ from further consideration. The final step is deciding whether to place (11) on the stack, or bisect (11) and place the resulting interval vectors on the stack. We decide this by computing $v(\underline{Z})/v(\underline{Y})$. If this volume ratio is less than γ (chosen arbitrarily), (11) will be placed on the stack, and bisected otherwise. This ends a typical pass through the main algorithm. (Trial-and-error experimentation shows that the threshold $\gamma = 1/2$ works well.)

In order to bisect $\underline{Y}$, the direction of bisection must be chosen. This choice can greatly affect the efficiency of the overall scheme. We consider two schemes for making this decision. Scheme 1 bisects in the direction with index j for which $w_j(Y_j)$ is maximum. This scheme tends to minimize the ratio $\max\limits_j w_j(Y_j) \Big/ \min\limits_j w_j(Y_j)$. Scheme 2, described in [19], defines

(14) $$s_j = \max\left\{ \left|J_{l,jk}\right| \; , \; \left|J_{u,jk}\right| \right\} w(Y_j) \quad ,$$

and bisects in the direction with index j for which s_j is maximum, where $J_{l,jk}$ and $J_{u,jk}$ are the lower and upper bounds of the jk-th element of the interval Jacobian, respectively. A comparison of results for these two bisection schemes will be presented in section 4.4.

2.4. "Spurious" Solutions. In some cases, some solutions of the augmented system (4a,b) will not correspond to finite solutions of the original system (1). These spurious solutions, with $y_{N+1} = 0$, are introduced by the transformation and multiplication described in section 2.2. For example, the system

(15a) $$5x_1^9 - 6x_1^5x_2^2 + x_1x_2^4 + 2x_1x_3 = 0$$

(15b) $$-2x_1^6x_2 + 2x_1^2x_2^3 + 2x_2x_3 = 0$$

(15c) $$x_1^2 + x_2^2 - 0.265 = 0$$

when transformed gives the augmented system

(16a) $$5y_1^9 - 6y_1^5y_2^2y_4^2 + y_1y_2^4y_4^4 + 2y_1y_3y_4^7 = 0$$

(16b) $$-2y_1^6y_2 + 2y_1^2y_2^3y_4^2 + 2y_2y_3y_4^5 = 0$$

(16c) $$y_1^2 + y_2^2 - 0.265y_4^2 = 0$$

(16d) $$y_1^2 + y_2^2 + y_3^2 - 1 = 0 \quad ,$$

which has two spurious solutions $(0,0,\pm 1,0)$, corresponding to unbounded solutions of the original system (15a-c). On the other hand, the system

(17a) $$1 + 2x_1 - 3x_1^2 + 4x_2 + 5x_2^2 - 6x_1x_2 = 0$$

(17b) $$-7 - 8x_1 + 9x_1^2 - 10x_2 - 11x_2^2 + 12x_1x_2 = 0$$

when transformed gives the augmented system

(18a) $$y_3^2 + 2y_1y_3 - 3y_1^2 + 4y_2y_3 + 5y_2^2 - 6y_1y_2 = 0$$

(18b) $$-7y_3^2 - 8y_1y_3 + 9y_1^2 - 10y_2y_3 - 11y_2^2 + 12y_1y_2 = 0$$

(18c) $$y_1^2 + y_2^2 - 1 = 0 \quad .$$

The transformed system (18a-c) has no real solutions with $y_3 = 0$, so globalization introduces no spurious solutions.

It is apparent from the properties of the mapping that every real finite solution of (1) is also a real finite solution of (4a,b). Thus, the real solutions of (4a,b) consist of the real solutions of (1) and any spurious solutions introduced by the transformation and multiplication described in section 2.2. No such solutions occur if the equations formed by omitting all but the leading-order terms (i.e., those of highest total degree [7]) in (1) have only isolated solutions, which when transformed do not satisfy $\sum_{i=1}^{N} y_i^2 = 1$. Spurious solutions will occur if at least one original variable $(x_1, \cdots, x_N)$ is absent from the leading-order terms in each equation in (1), or appears in all of the leading-order terms of more than one equation. In the example (15a-c), x_3 is absent from the leading-order terms in each equation, whereas each variable appears in at least one leading-order term in at least one equation of (17a,b), in which system no variable is present in all leading-order terms in more than one equation.

It is possible to transform (4a,b) so that solutions with $y_{N+1} = 0$ are excluded, thus restricting the solutions to those corresponding to finite solutions of (1). This is accomplished using techniques from algebraic geometry. To remove all solutions with $y_{N+1} = 0$ from (4a,b), the ideal quotient [7] of the ideals generated by $y_{N+1} = 0$ and the N equations of (4a) is found. The ideal quotient is itself an ideal defining a new system of N equations. That system, combined with (4b), has a solution set identical to the solution set of (4a,b), less the spurious solution(s) of the latter. To compute the ideal quotient one determines the intersection of two ideals, using Gröbner bases [7]. For example, the system (15a-c) has twelve real solutions in R^3. The solution set of the augmented system (16a-d) consists of fourteen real solutions, including the two spurious solutions noted above. These spurious solutions can be excluded using the above technique. The new system is

(19a) $$68719476736 \cdot 10^4 y_3^3 - 20722660655104 \cdot 10^4 y_3^2 y_4 - 588558700544 \cdot 10^2 y_3 y_4^2 - 3982698885 y_4^3 = 0$$

(19b) $$2048000 y_3^2 - 643072 y_3 y_4 - 9537 y_4^2 = 0$$

(19c) $$64 y_1^2 + 64 y_2^2 - 17 y_4^2 = 0$$

(19d) $$y_1^2 + y_2^2 + y_3^2 - 1 = 0 \quad .$$

The solution set of (19a-d) consists of all nonspurious solutions of (16a-d), i.e., those with $y_4 \neq 0$. The main effort in computing an ideal quotient is computing a Gröbner basis. As noted in section 1, the Gröbner transformation can be used to find all real solutions of (1). An important point to note is that to use the Gröbner transformation a specific term ordering (e.g., $x_1^2 x_2 > x_1 x_2$ using total degree ordering) must be chosen for the monomials [7] in each polynomial equation of (4a,b). The choice of a term ordering (cf. [6]) affects the computational complexity of constructing a Gröbner basis. Our experience also shows that the ordering of the variables (corresponding to permutations such as (x_1, x_2, x_3), (x_1, x_3, x_2), etc.) can also significantly influence the operation count. In most cases, the lexicographical term ordering needed to compute a reduced Gröbner basis (most useful in computing multivariate polynomial zeros) requires many more operations than to compute a generic Gröbner basis using some other term ordering. Thus, in our removal of spurious solutions from (4a,b), we will use a nonlexicographical term ordering, and avoid computing *any* reduced Gröbner basis.

In closing this section we note that ours is a "black box" algorithm for general use. On the other hand, one can use our approach as a "tool", without removing spurious solutions from (4a,b). In that event, there can occur cases in which a large number of very small interval vectors (with width less than a tolerance set by the user) contain spurious solutions. It has been our experience that these interval vectors can be safely discarded without loss of finite real simple solutions (i.e., those with unit multiplicity).

2.5. Multiply-Computed and Boundary Solutions. This contraction-mapping algorithm will not find a solution lying on the boundary of the initial domain or on a boundary created by bisection. To ensure that all solutions of (1) are found, we take two precautions to include the boundaries of each domain. First, when the interval vector $\underline{Y}$ with components $Y_i = [y_{i,l}, y_{i,u}]$ is bisected in the j-th direction we do so according to

(20a) $$Y_{1,i} = [y_{i,l}, y_{i,u}] \quad \text{and} \quad Y_{2,i} = [y_{i,l}, y_{i,u}] \qquad 1 \le i \le N+1 \ , \qquad i \ne j$$

(20b) $$Y_{1,j} = [y_{j,l},\ (1/2-\varepsilon_1)\cdot y_{j,l} + (1/2+\varepsilon_1)\cdot y_{j,u}]$$

(20c) $$Y_{2,j} = [(1/2+\varepsilon_1)\cdot y_{j,l} + (1/2-\varepsilon_1)\cdot y_{j,u},\ y_{j,u}] \ ,$$

so that no point originally in the interior of $\underline{Y}$ lies on the boundary of either new subdomain. Here, $\varepsilon_1 > 0$ is typically 0.005. The second step involves expansion of the interval vectors $\underline{Y}_1$ and $\underline{Y}_2$ before they are placed on the stack. The new interval vectors are

(21a) $$Y^{\bullet}_{1,i} = [y_{1,i,l} - \varepsilon_2 w(Y_{1,i}),\ y_{1,i,u} + \varepsilon_2 w(Y_{1,i})]$$

$$1 \le i \le N+1$$

(21b) $$Y^{\bullet}_{2,i} = [y_{2,i,l} - \varepsilon_2 w(Y_{2,i}),\ y_{2,i,u} + \varepsilon_2 w(Y_{2,i})]$$

where $\varepsilon_2 > 0$ is typically 0.0025. Thus, any solution lying on a boundary of $\underline{Y}_1$ or $\underline{Y}_2$ will be included in $\underline{Y}^{\bullet}_1$ or $\underline{Y}^{\bullet}_2$.

These "precautions" can, however, lead to computing a simple solution more than once. This problem is addressed after all solutions of (1) have been found. We first compute the maximum distance d_{max} between all pairs of solutions. Then if the distance d_i between a given pair of solutions satisfies

(22) $$d_i/d_{\max} < \varepsilon_3 \ ,$$

we consider the possibility that they are the same solution, where a typical value of ε_3 is 10^{-6}. This list can contain simple solutions computed more than once. We place a box around each cluster of such solutions and compute the solution(s) in this box using the algorithm discussed in sections 2.1-2.3. Solutions which are unique will be thus identified. The final solution set contains the original unclustered solutions and those solutions found by this "declustering" process.

2.6. Nonsimple Solutions. Like every Jacobian-based procedure, the method described above encounters difficulties for nonsimple solutions (i.e., those with multiplicity greater than one), at which points the Jacobian is singular. Since the determinant of the Jacobian is a single function of the variables and vanishes at every nonsimple solution, it is easy to see that at each such point the Jacobian either has an isolated singularity or is singular on one or more curves or surfaces passing through the point. Thus, unless the Jacobian has only isolated singularities, Newton iteration will encounter curves (in the bivariate case) or surfaces or hypersurfaces (more generally) on which the Jacobian is singular. This typically leads to poor convergence behavior [20-21].

Here we show how to construct a new system whose solutions include all of the nonsimple solutions of the original system. The multiplicity of these solutions is typically reduced by one. If no solution of the original system has multiplicity greater than two, then the problem is reduced to one that can be dealt with using the techniques described in sections 2.1-2.5. Otherwise, the reduction described here can be applied sequentially until the multiplicity of the solution with highest original multiplicity has been reduced to one, allowing reliable numerical solution. Our approach is a direct extension of a technique for the univariate case [22].

We construct a new system of equations consisting of some $(N-1)$-element subset of the original equations, augmented by the determinant of the original Jacobian

(23) $$f_{N+1} = \left|\underline{\underline{J}}(f_1, f_2, \cdots, f_N)\right| = 0 \ .$$

As indicated in examples below (section 4.4), this process can be repeated to handle nonsimple solutions with multiplicity greater than two. We note here that the new system can have one or more solutions not satisfying (1). Also, there are N ways to construct the new equation set. As discussed in example 5 below, some are more useful than others.

The method performs well for almost all systems with nonsimple solutions, as indicated by the examples below. As discussed at the end of this subsection, it does not work for Powell's degenerate system (cf. [23]), which arises from applying derivative-based techniques to minimization of "Powell's singular function" ([24], p. 150), and another example we have developed.

1. A bivariate system with a solution of multiplicity two:

(24a) $$f_1 = x_1^2 + x_2^2 - 2 = 0$$

(24b) $$f_2 = x_1^2 + 2x_2^2 + 3x_1 + x_2 - 7 = 0 \ .$$

This system has one distinct real solution (1,1), of multiplicity two.

The reduced-multiplicity system (f_2, f_3) is

(25a) $$x_1^2 + 2x_2^2 + 3x_1 + x_2 - 7 = 0$$

(25b) $$4x_1x_2 - 6x_2 + 2x_1 = 0 \ ,$$

with a real simple solution (1,1). It is easily shown that the other real solution of (25a,b) does not satisfy (24a,b). In this example and the next three, any choice of $N-1$ equations, augmented by the Jacobian of the N-dimensional system, reduces the multiplicity.

2. A bivariate system with a solution of multiplicity three:

(26a) $$f_1 = x_1^2 + x_2^2 - 2 = 0$$

(26b) $$f_2 = x_1^2 + 2x_2^2 + x_1 - x_2 - 3 = 0 \ .$$

This system has two distinct real solutions (1,1) and (-1,-1), the latter having multiplicity three.

We construct the new system in two steps. In the first, we compute

(27a) $$f_3 = |\underline{\underline{J}}(f_1, f_2)| = 4x_1x_2 - 2x_1 - 2x_2 = 0 \ ,$$

and in the second step, for the pair (f_1, f_3), compute

(27b) $$f_4 = |\underline{\underline{J}}(f_1, f_3)| = 8x_1^2 - 8x_2^2 - 4x_1 + 4x_2 = 0 \ .$$

Then the new system (f_1, f_4) is

(28a) $$x_1^2 + x_2^2 - 2 = 0$$

(28b) $$8x_1^2 - 8x_2^2 - 4x_1 + 4x_2 = 0 \ .$$

The new system (28a,b) has the same solutions as (26a,b), with each being simple.

3. A trivariate system with a solution of multiplicity two:

(29a) $$f_1 = x_1^2 + x_2^2 - 2 = 0$$

(29b) $$f_2 = x_1^2 + 2x_2^2 + 3x_1 + x_2 - 7 = 0$$

(29c) $$f_3 = x_1^2 - x_2^2 + x_3 + 2 = 0 \ .$$

The new system is

(30a) $$x_1^2 + 2x_2^2 + 3x_1 + x_2 - 7 = 0$$

(30b) $$x_1^2 - x_2^2 + x_3 + 2 = 0$$

(30c) $$2x_1 - 6x_2 + 4x_1x_2 = 0 \ .$$

Here, the original system has one distinct real solution of multiplicity two at (1,1,-2). We find this as a simple solution of the new system (30a-c).

4. A bivariate system with two solutions of multiplicity two:

(31a) $$f_1 = x_1^2 + x_2^2 + x_1 - 2x_2 - 5 = 0$$

(31b) $$f_2 = 52x_1^2 + 73x_2^2 + 72x_1x_2 - 20x_1 - 110x_2 - 575 = 0 \ .$$

The new system is

(32a) $$x_1^2 + x_2^2 + x_1 - 2x_2 - 5 = 0$$

(32b) $$-144x_2^2 + 144x_1^2 + 84x_1x_2 + 60x_1 + 330x_2 - 150 = 0 \ .$$

Here, the original system has two distinct real solutions at (-2,-1) and (1,3), each of multiplicity two. We find these as simple solutions of the new system (32a,b).

5. A bivariate system with a solution of multiplicity two [21]:

(33a) $$f_1 = x + y^2 = 0$$

(33b) $$f_2 = \tfrac{3}{2}xy + y^2 + y^3 = 0 \ .$$

The new system (f_1, f_3) is

(34a) $$x + y^2 = 0$$

(34b) $$\tfrac{3}{2}x + 2y = 0 \ .$$

The original system has two distinct real solutions at (-4,2) and (0,0), with the latter having multiplicity two. The simple solution at (-4,2) is found directly from (33a,b), while the solution of multiplicity two at (0,0) is found from (34a,b). In this example, the choice (f_1, f_3) leads to the desired multiplicity reduction, while (f_2, f_3) does not.

6. A bivariate system with a solution of multiplicity three [20]:

(35a) $$f_1 = x_1^3 + x_1 x_2 = 0$$

(35b) $$f_2 = x_2 + x_2^2 = 0 \ .$$

This system has four distinct real solutions at (0,-1), (-1,-1), (-1,1), and (0,0), with the latter having multiplicity three. As in example 2, one stage of multiplicity reduction (using f_2 and f_3) results in a system where the solution of (35a,b) with multiplicity three is now a solution of multiplicity two. The second stage (using f_2 and $f_4 = |\underline{\underline{J}}(f_2, f_3)|$) gives

(36a) $$x_2 + x_2^2 = 0$$

(36b) $$x_1 + 4x_1 x_2 + 4x_1 x_2^2 = 0 \ .$$

The simple solutions of (35a,b) are found directly from that system, while (36a,b) provide the solution of multiplicity three at (0,0) and the simple solution at (0,-1). In this example, (f_1, f_3) gives no multiplicity reduction in the first stage, and $(f_3,\ f_4)$ gives none in the second stage.

We have found two test cases, each with nonsimple isolated solutions, that do not yield to our approach. One is Powell's four-dimensional system [23-24], for which the Jacobian vanishes on two hyperplanes, on each of which one of the original equations also vanishes. Kearfott's algorithm [25] also experienced difficulty with this case. A second example is

(37a) $$x_1^2 - x_2^2 = 0$$

(37b) $$x_1 x_2 = 0 \ ,$$

for which the Jacobian vanishes only at the origin. In each case, application of our approach without prior multiplicity reduction yields small regions (containing the solution) for which no definitive statement is possible. Use of multiplicity reduction for Powell's example fails to define a new system for which solutions are isolated. For (37a,b), the solution of the new system does not have lower multiplicity.

3. Bounding multivariate polynomials. Our approach requires bounding each equation of (4a,b) over a finite domain $\underline{Y}$. Three methods are discussed and implemented for bounding multivariate polynomials with real coefficients over a finite domain $\underline{Y}$.

In the first method [26], one transforms each polynomial of (4a,b) or (7) into generalized Bernstein form, computes its Bernstein coefficients, and takes the minimum and maximum as the bounds over $\underline{Y}$. This method gives exact bounds if and only if the original polynomial is monotonically increasing or decreasing on the domain over which a bound is required.

The second method [26-27] is based on the mean value theorem and uses function evaluations and a Taylor-like correction term to bound each polynomial of (4a,b) or (7) over $\underline{Y}$. This method relies in part on polynomial evaluations. The evaluations are computed using the tree traversal method [28]. To use this method the coefficients of (4a,b) or (7) are stored as a rooted, ordered tree and then traversed by pre-order traversal. This technique is equivalent to an N-dimensional Horner scheme and takes advantage of polynomial sparsity.

The third method [17] is based on the interval extension $G_j(\underline{Y})$ of either (4a,b) or (7). An interval extension is found by evaluating the polynomial (real variables replaced with interval variables) using the tree traversal technique used in the second method.

The polynomial bounding method was chosen based on extensive tests for univariate and bivariate cases. It was found that the first method gives the tightest bounds for a given computational cost. The second is comparable to the first in time but easier to implement in higher dimensions. The third method is much faster, but provides less tight bounds. The first method is the hardest to implement using the tree traversal representation of polynomials employed here. The second and third methods are easily implemented using some form of tree traversal, while also being easily extended to higher dimensions. After running examples utilizing each method in the "solver," we conclude that the third method gives the best overall performance (i.e., lowest total time).

These techniques give bounds that *include* the true range of g_j. Computational burden is associated with the fact that these bounds are not "tight." The extent to which this is acceptable depends on the relationship between bounding costs and other computational costs. For this reason, it is desirable to find a means of increasing the tightness of any bounding scheme. This is accomplished by bisecting the initial domain and computing bounds for each resulting subdomain. The final bounds over the initial domain are taken to be the minimum and maximum of the bounds found for the subdomains. This is continued until the bounds attain the desired tightness over the initial domain.

4. Computational results. In this section, we present results for four groups of test problems, along with a comparison of bisection techniques. The first group of test problems consists of what have become standard tests for nonlinear equation solvers. The second and third sets consist of N-dimensional quadratic systems with random coefficients, and bivariate M-th degree systems with random coefficients.

4.1. Standard Test Cases. The first test suite is a subset of the test suite from [25], and includes problems with a physical basis (e.g., a combustion problem) and made-up test cases. These problems have been used as test cases for homotopy methods, as well as for simplicial and interval-Newton methods. The "Remarks" are generally taken or paraphrased from [25].

1. Brown's almost linear system (N=5):

(38a) $$f_i = x_i + \sum_{j=1}^{N} x_j - (N+1) = 0 \qquad 1 \le i \le N-1$$

(38b) $$f_N = \prod_{j=1}^{N} x_j - 1 = 0$$

Remarks: The Jacobian matrix is said to be ill-conditioned at the two solutions of smallest magnitude [25]. This function [29] is also found in [23].

It is shown in [23] that (38a,b) has two solutions for even N and three for odd N. We find three.

2. Combustion chemistry problem:

(39a) $$f_1 = -1.697 \cdot 10^7 x_2 x_4 + 2.177 \cdot 10^7 x_2 + 0.55 x_1 x_4 + 0.45 x_1 - x_4 = 0$$

(39b) $$f_2 = 1.585 \cdot 10^{14} x_2 x_4 + 4.126 \cdot 10^7 x_1 x_3 - 8.285 \cdot 10^6 x_1 x_4 + 2.284 \cdot 10^7 x_3 x_4 - 1.918 \cdot 10^7 x_3 + 48.4 x_4 - 27.73 = 0$$

(39c) $$f_3 = x_1^2 - x_2 = 0$$

(39d) $$f_4 = x_4^2 - x_3 = 0$$

Remarks: These equations arise in a chemical equilibrium problem [30]. The problem has been solved via continuation methods. It has a unique solution within the nonnegative unit box, but has other, nonphysical solutions in larger domains.

We find two real solutions, one physical and one nonphysical. (Note that the coefficient of the x_3 term in f_2 is given with different signs in [25] and [30].)

3. A high-degree polynomial system:

(40a) $$5x_1^9 - 6x_1^5 x_2^2 + x_1 x_2^4 + 2x_1 x_3 = 0$$

(40b) $$-2x_1^6 x_2 + 2x_1^2 x_2^3 + 2x_2 x_3 = 0$$

(40c) $$x_1^2 + x_2^2 - 0.265625 = 0$$

Remarks: This problem has 12 real solutions, which are all within the box [-0.6,0.6] × [-0.6,0.6] × [-5,5], and eight other finite solutions. The total degree is 126. Thus, this system causes trouble for most homotopy continuation methods.

We find all 12 real solutions.

4. Rosenbrock's function:

(41a) $$f_1 = 1 - x_1 = 0$$

(41b) $$f_2 = 10(x_2 - x_1^2) = 0$$

Remarks: This problem is a standard test case for minimization algorithms [31] and has also been considered from the standpoint of a nonlinear equation system [32].

We find the sole solution (1,1).

5. A variable-dimension system of quadratics:

(42a) $$f_i = (x_i - 0.1)^2 + x_{i+1} - 0.1 = 0 \qquad 1 \le i \le N-1$$

(42b) $$f_N = (x_N - 0.1)^2 + x_1 - 0.1 = 0$$

Remarks: This simple system of quadratics can be used to test the effects of increasing dimension on bisection methods. Since its Jacobian matrix is sparse, it can also be used to debug techniques for handling structured problems.

We find the two real solutions for $N = 4$.

4.2. N-Dimensional Systems of Quadratics. This test suite consists of systems of quadratics

(43) $$f_i = a_i + \sum_{j=1}^{N} b_{ij} x_j + \sum_{j=1}^{N} \sum_{k=1}^{N} c_{ijk} x_j x_k \qquad 1 \le i \le N$$

with varying dimension N. The coefficients are chosen randomly between -10 and 10, so that unlike example 5 of section 4.1, the Jacobian is typically not sparse. This test suite can also be used to test the effects (e.g., on cost growth) of increasing dimension. Table 1 shows running-time statistics on a 166MHz Pentium machine for 10 test systems for each N. Running time depends strongly on dimension. The large standard deviations indicate a considerable sensitivity with respect to the coefficients.

4.3. Pairs of M-th Degree Equations. This test suite consists of bivariate systems of various degrees. The coefficients are again chosen randomly to be between -10 and 10. Table 2 shows running-time statistics (same machine as used in section 4.2) with an ensemble of 100 bivariate systems. One can see by comparing tables 1 and 2 that cost grows less rapidly with degree than with dimension, as expected. We also note that the ratio of mean times $<T>_M / <T>_{M+1}$ increases slightly with the degree M, and that although the ratio of the standard deviation to the mean time decreases slowly (and not quite monotonically) with degree, the ratio of maximum to minimum run times decreases consistently and significantly with degree.

4.4. Bisection Scheme Comparison. In this section we present timing results for the two bisection schemes discussed in section 2.3. Table 3 shows running-time statistics (same machine degree. Scheme 2 is slightly faster. For small degree, some variation in running time is expected for test suites having identical coefficients.

5. Discussion. For broad classes of multivariate polynomial equation systems, the numerical solution technique discussed in this paper has several advantages over other global methods. First, our technique seeks and finds only real solutions of (1), unlike homotopy methods which must track all solutions (i.e., complex and real solutions). Second, our method can be used to find all solutions in a finite subdomain of R^N (e.g., when the solutions must all be nonnegative, as when computing concentrations in chemically reacting systems, allocating resources, etc.) without dealing with the rest of R^N, thus making it possible to omit the globalization transformation section 2.2. Third, the technique can be tailored to quadratic systems (for which the Jacobian is linear and can be bounded exactly) to significantly reduce run time. Finally, by assigning subdomains to all available processors, the code can easily be parallelized [12].

Our approach to computing nonsimple solutions also has advantages over existing methods. First, unlike approaches [20-21] that carefully select the initial iterate to "coax" Newton methods to converge at singular points (with generally sublinear convergence), we consider a sequence of new systems, each having a lower multiplicity at these singular points until the multiplicity is reduced to one. This allows our interval extension of Newton's method to converge quadratically. We also note that the multiplicity reduction technique can be used as a pre-transformation for Newton-Raphson and other single-point iteration techniques.

Acknowledgments. This work has been supported in part by NASA Grant NGT-51392 and AFOSR Grant F49620-95-1-0297, for which the authors are grateful.

REFERENCES

[1] A. P. MORGAN, *A transform to avoid solutions at infinity for polynomial systems*, Appl. Math. Comput., 18 (1986), pp. 77-86.

[2] K. A. ABBOTT, B. A. ALLAN, and A. W. WESTERBERg, *Global preordering for Newton equations using model hierarchy*, AIChE J., 43 (1997), pp. 3193-3204.

[3] A. J. O' NEILL, D. J. KAISER, and M. A. STADTHERR, *Strategies for equilibrium-stage separation calculations on parallel computers*, AIChE J. 40 (1994), pp. 65-72.

[4] E. L. ALLGOWER, K. GEORG, and R. MIRANDA, *The method of resultants for computing real solutions of polynomial systems*, SIAM J. Numer. Anal. 29 (1992), pp. 831-844.

[5] J. CANNY, *Generalised characteristic polynomials*, J. Symbolic Comput. 9 (1990), pp. 241-250.

[6] B. BUCHBERGER, *Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory*, in Multidimensional Systems Theory: Progress, Directions, and Open Problems in Multidimensional Systems, N. K. Bose, ed., Reidel, Dordrecht, 1985, pp. 184-232.

[7] D. COX, J. LITTLE, and D. O'SHEA, *Ideals, Varieties, and Algorithms: an introduction to computational algebraic geometry and commutative algebra,* 2nd edition, Springer, New York, 1997.

[8] W. I. ZANGWILL and C. B. GARCIA, *Pathways to Solutions, Fixed Points, and Equilibria*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[9] R. E. MOORE and S. T. JONES, *Safe starting regions for iterative methods*, SIAM J. Numer. Anal. 14 (1977), pp. 1051-1065.

[10] J. Z. HUA, J. F. BRENNECKE, and M. A. STADTHERR, *Enhanced interval analysis for phase stability: cubic equation of state models*, Ind. Eng. Chem. Res. 37 (1998), pp. 1519-1527.

[11] R. W. MAIER, J. F. BRENNECKE, and M. A. STADTHERR, *Reliable computation of homogeneous azeotropes*, AIChE J. 44 (1998), pp. 1745-1755.

[12] C. A. SCHNEPPER and M. A. STADTHERR, *Robust process simulation using interval methods*, Computers Chem. Eng. 20 (1996), pp. 187-199.

[13] J. Z. HUA, J. F. BRENNECKE, and M. A. STADTHERR, *Reliable computation of phase stability using interval analysis: cubic equation of state models*, Computers Chem. Eng. 22 (1998), pp. 1207-1214.

[14] R. E. MOORE, *A test for existence of solutions to nonlinear systems*, SIAM J. Numer. Anal. 14 (1977), pp. 611-615.

[15] E. R. HANSEN and G. W. WALSTER, *Nonlinear equations and optimization*, Computers Math. Applic. 25 (1993), pp. 125-145.

[16] G. ALEFELD and J. HERZBERGER, *Introduction to Interval Computations* (Academic Press, New York, 1983).

[17] R. E. MOORE, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.

[18] R. KRAWCZYK, *Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken*, Computing 4 (1969), pp. 187-201.

[19] R. B. KEARFOTT and M. NOVOA, *Algorithm 681. INTBIS, a portable interval Newton/ bisection package*, ACM Trans. Math. Software 16 (1990), pp. 152-157.

[20] D. W. DECKER and C. T. KELLEY, *Newton's method at singular points. II*, SIAM J. Numer. Anal. 17 (1980), pp. 465-471.

[21] D. W. DECKER and C. T. KELLEY, *Broyden's method for a class of problems having singular Jacobian at the root*, SIAM J. Numer. Anal. 22 (1985), pp. 566-574.

[22] J. V. USPENSKY, *Theory of Equation*, McGraw-Hill, 1948.

[23] A. P. MORGAN, *A method for computing all solutions to systems of polynomial equations*, ACM Trans. Math. Software 9 (1983), pp. 1-17.

[24] M. J. D. POWELL, *An iterative method for finding stationary values of a function of several variables*, Computer J. 5 (1962), pp. 147-151.

[25] R. B. KEARFOTT, *Some tests of generalized bisection*, ACM Trans. Math. Software 13 (1987), pp. 197-220.

[26] J. GARLOFF, *Convergent Bounds for the Range of Multivariate Polynomials*, in Interval Mathematics 1985, Lecture Notes in Computer Science 212, K. Nickel, ed., Springer-Verlag, New York, 1985, pp. 37-56.

[27] T. J. RIVLIN, *Bounds on a polynomial*, J. Res. Natl. Bur. Standards 74B (1970), pp. 47-54.

[28] W. C. RHEINBOLDT, C. K. MESZTENYI, and J. M. FITZGERALD, *On the evaluation of multivariate polynomials and their derivatives*, BIT 17 (1977), pp. 437-450.

[29] K. M. BROWN, *A quadratically convergent Newton-like method based upon Gaussian elimination*, SIAM J. Numer. Anal. 6 (1969), pp. 560-569.

[30] A. MORGAN and V. SHAPIRO, *Box-bisection for solving second-degree systems and the problem of clustering*, ACM Trans. Math. Software 13 (1987), pp. 152-167.

[31] H. H. ROSENBROCK, *An automatic method for finding the greatest or least value of a function*, Computer J. 3 (1960), pp. 175-184.

[32] J. J. MORÉ, B. S. GARBOW, and K. E. HILLSTROM, *Testing unconstrained optimization software*, ACM Trans. Math. Software 7 (1981), pp. 17-41.

APPENDIX

A computationally verifiable sufficient condition is given for nonexistence (in a rectangular domain) of a solution to a system of polynomial equations. Using this test, which is based on the Krawczyk operator, the number of bisections required by the Moore-Krawczyk algorithm can be significantly reduced.

We call the reader's attention to the explicit dependence of (2) on the real matrix $\underline{\underline{\Phi}}$, which can be chosen arbitrarily, subject only to $|\underline{\underline{\Phi}}| \neq 0$. We note here that the theorems [9,17] concerning application of (2) are true for arbitrary nonsingular $\underline{\underline{\Phi}}$. In what follows, we exploit this flexibility to develop a nonexistence criterion.

We begin by recognizing that every solution of (1) in $\underline{X}$ must lie in $\underline{K}(\underline{X},\underline{h},\underline{\underline{\Phi}})$ for every $\underline{h}$ in $\underline{X}$ and every nonsingular $\underline{\underline{\Phi}}$. As we have fixed $\underline{h}$, we will concentrate on choosing $\underline{\underline{\Phi}}$. We rewrite (2) in component form (using index notation) as

(A1) $$K_i = m(X_i) - \Phi_{ij} f_j(\underline{h}) + \left\{\delta_{ik} - \Phi_{ij}\left(m(J_{jk}) + [-1,1]\tfrac{1}{2}w(J_{jk})\right)\right\}\left(X_k - m(X_k)\right)$$

where $\underline{\underline{J}}$ is in midpoint-halfwidth form [17]. We denote K_i and X_i by $[\underline{K_i}, \overline{K_i}]$ and $[\underline{X_i}, \overline{X_i}]$, respectively, and note that $\underline{K}(\underline{X},\underline{h},\underline{\underline{\Phi}}) \cap \underline{X}$ will be empty (and hence (1) will have no solutions in $\underline{X}$) if

(A2) $$\underline{K_i} > \overline{X_i} \quad \text{or} \quad \overline{K_i} < \underline{X_i}$$

for some combination of i and (nonsingular) $\underline{\underline{\Phi}}$. We are thus led to rewrite (A1) as

(A3) $$K_i = m(X_i) - \Phi_{ij} f_j(\underline{h}) + \left\{\delta_{ik} - \Phi_{ij} m(J_{jk}) - \Phi_{ij}\tfrac{1}{2}w(J_{jk})[-1,1]\right\}\tfrac{1}{2}w(X_k)[-1,1]$$

from which it follows that

(A4a) $$\underline{K_i} = m(X_i) - \Phi_{ij} f_j(\underline{h}) - \left\{\left|\delta_{ik} - \Phi_{ij} m(J_{jk})\right| + \left|\Phi_{ij}\right|\tfrac{1}{2}w(J_{jk})\right\}\tfrac{1}{2}w(X_k)$$

and

(A4b) $$\overline{K_i} = m(X_i) - \Phi_{ij} f_j(\underline{h}) + \left\{\left|\delta_{ik} - \Phi_{ij} m(J_{jk})\right| + \left|\Phi_{ij}\right|\tfrac{1}{2}w(J_{jk})\right\}\tfrac{1}{2}w(X_k)$$

From (A4a,b) we have

(A5a) $$\underline{K_i} \geq m(X_i) - \Phi_{ij} f_j(\underline{h}) - \left\{\delta_{ik} + \left|\Phi_{ij}\right|\left|m(J_{jk})\right| + \left|\Phi_{ij}\right|\tfrac{1}{2}w(J_{jk})\right\}\tfrac{1}{2}w(X_k)$$

and

(A5b) $$\underline{K}_i \geq m(X_i) - \Phi_{ij} f_j(\underline{h}) - \tfrac{1}{2} w(X_i) - \left|\Phi_{ij}\right| \left\{ \left| m(J_{jk}) \right| + \tfrac{1}{2} w(J_{jk}) \right\} \tfrac{1}{2} w(X_k) \quad ,$$

from which follows an important result.

Theorem. If

(A6) $$\left| f_j(\underline{h}) \right| > \left(\tfrac{1}{2} w(J_{jk}) + \left| m(J_{jk}) \right| \right) \tfrac{1}{2} w(X_k)$$

is true for some j, then (1) has no solutions in $\underline{X}$.

Proof.

From (A2) and (A5b), it follows that if

(A7) $$m(X_i) - \Phi_{ij} f_j(\underline{h}) - \tfrac{1}{2} w(X_i) - \left|\Phi_{ij}\right| \left\{ \left| m(J_{jk}) \right| + \tfrac{1}{2} w(J_{jk}) \right\} \tfrac{1}{2} w(X_k) > m(X_i) + \tfrac{1}{2} w(X_i)$$

is true for some i and nonsingular $\underline{\underline{\Phi}}$, then $\underline{K}(\underline{X}, \underline{h}, \underline{\underline{\Phi}}) \cap \underline{X} = \phi$. Therefore, from (A7),

(A8) $$-\Phi_{ij} f_j(\underline{h}) - \left|\Phi_{ij}\right| \left\{ \left| m(J_{jk}) \right| + \tfrac{1}{2} w(J_{jk}) \right\} \tfrac{1}{2} w(X_k) > w(X_i)$$

will be sufficient to ensure emptiness. Assume (A6) holds for some j. There are two cases. First, consider $f_j > 0$. Then choose

(A9) $$-\Phi_{mj} > \frac{w(X_m)}{f_j(\underline{h}) - \left\{ \left| m(J_{jk}) \right| + \tfrac{1}{2} w(J_{jk}) \right\} \tfrac{1}{2} w(X_k)} > 0$$

for one m (we take $m = j$ for simplicity), $\Phi_{nj} = 0$ for $n \neq j$, and $\Phi_{rs} = \delta_{rs}$ for $s \neq j$ Thus, (A8) is satisfied and $|\underline{\underline{\Phi}}| \neq 0$. For $f_j < 0$, we need only choose

(A10) $$\Phi_{mj} > \frac{w(X_m)}{\left| f_j(\underline{h}) \right| - \left\{ \left| m(J_{jk}) \right| + \tfrac{1}{2} w(J_{jk}) \right\} \tfrac{1}{2} w(X_k)} > 0$$

for one m, with the other elements of $\underline{\underline{\Phi}}$ chosen as before. Thus, if (A6) is satisfied for some j, we can construct a nonsingular $\underline{\underline{\Phi}}$ such that (A8) is satisfied, and $\underline{K}(\underline{X}, \underline{h}, \underline{\underline{\Phi}}) \cap \underline{X}$ will be empty, proving the theorem.

A simple interpretation of this theorem is possible. For $N = 1$ (a one-dimensional system), (1) will have no zeros on an interval [a,b] if

(A11) $$\left| f\left(\frac{a+b}{2} \right) \right| > \frac{b-a}{2} \max_{x \in [a,b]} \left| f'(x) \right| \quad ,$$

the meaning of which is obvious. In N dimensions, satisfaction of (A6) means that the j-th component of the function value $\underline{f}(\underline{h})$ is too large for f_j to vanish in $\underline{X}$.

TABLE 1

Running-time statistics for quadratic systems as a function of dimension N.

dimension, N	maximum time (sec)	minimum time (sec)	mean time (sec)	standard deviation
2	0.187	0.032	0.113	0.047
3	2.875	0.562	1.728	0.731
4	133.8	31.44	52.54	29.74
5	1267	420.5	949.4	289.8

TABLE 2

Running-time statistics for pairs of higher-order equations as a function of degree M.

degree, M	T_{max} maximum time (sec)	T_{min} minimum time (sec)	$<T>$ mean time (sec)	standard deviation
2	0.281	0.031	0.079	0.039
3	0.938	0.125	0.295	0.135
4	1.781	0.344	0.823	0.289
5	3.844	0.875	1.916	0.715

TABLE 3

Mean running-times for pairs of higher-order equations using two bisection schemes.

degree, M	Scheme 1 mean time (sec)	Scheme 2 mean time (sec)
2	0.087	0.081
3	0.295	0.275
4	0.823	0.773
5	1.916	1.709

```
while stack not empty
    get interval vector $\underline{Y}$ from stack
    compute $G_j(\underline{Y})$
    if (8) false
        no solution in $\underline{Y}$
    else if (9) true for some $j$
        no solution in $\underline{Y}$
    else if $\underline{\underline{\Phi}}$ singular
        bisect $\underline{Y}$ and place its halves on stack
    else if $\underline{K}(\underline{Y}) \cap \underline{Y} = \varnothing$
        no solution in $\underline{Y}$
    else if $\underline{Y} \subseteq \underline{K}(\underline{Y})$
        bisect $\underline{Y}$ and place its halves on stack
        else if $\underline{K}(\underline{Y}) \subset \underline{Y}$
            solution lies in $\underline{K}(\underline{Y})$
            if $\left\| \underline{\underline{R}}^0 = \underline{\underline{I}}_N - \underline{\underline{\Phi}}\,\underline{\underline{J}}(\underline{Y}) \right\| < 1$
                $\underline{\underline{\Phi}}^0 = \underline{\underline{\Phi}}$
                $\underline{Y}^1 = \underline{Y}$
                for $k$=1:#iterations
                    do
                        compute $\underline{\underline{\Phi}}^k = \underline{\underline{\Phi}}(\underline{Y}^k)$
                        compute $\underline{\underline{R}}^k = \underline{\underline{I}}_N - \underline{\underline{\Phi}}^k\,\underline{\underline{J}}(\underline{Y}^k)$
                        if $\left\|\underline{\underline{R}}^k\right\| / \left\|\underline{\underline{R}}^{k-1}\right\| > 1$
                            $\underline{\underline{\Phi}}^k = \underline{\underline{\Phi}}^{k-1}$
                        end if
                        compute $\underline{Y}^{k+1} = \underline{Y}^k \cap \underline{K}(\underline{Y}^k)$
                    while $\left\|\underline{Y}^{k+1} - \underline{Y}^k\right\| > \varepsilon$
                end for
            end if
        else
            compute $\underline{Z} = \underline{K}(\underline{Y}) \cap \underline{Y}$
            if $v(\underline{Z}) / v(\underline{Y}) < \gamma$
                place $\underline{Z}$ on stack
            else
                bisect $\underline{Z}$ and place its halves on stack
            end if
        end if
    end if
end while
```

Figure 1. Pseudocode of the basic algorithm.