JOURNAL OF COMPUTATIONAL PHYSICS **128,** 409–426 (1996)
ARTICLE NO. 0220

Fast Numerical Computation of 2D Free Surface Jet Flow with Surface Tension

N. ANDERS PETERSSON

Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico

Received November 1, 1993; revised January 16, 1996

A system of partial differential equations that approximate the governing equations for inviscid free surface flow subject to surface tension is presented. The approximation is based on linearizing the velocity together with a small scale approximation of the perturbation of the velocity. Two Dirichlet problems must be solved to form the approximate system, after which it can be evolved without solving Dirichlet problems. The accuracy of the solution is determined by how often the velocity term is linearized. This time-interval is called ΔT. We show that the error in the solution of the approximate system at a fixed time T is of the order $\mathcal{O}(\Delta T^2)$. We demonstrate numerically that the error is closely correlated to the size of the normal velocity and that there is a stability limit of the form $\Delta T \leq C/(|u_n|_\infty)^\gamma$, where u_n denotes the normal velocity of the free surface and $\gamma \approx 2.6$. Importantly, C is independent of the resolution, so the time-step ΔT can be chosen independently of the number of grid points, N. This is in contrast to the original system, where the stability limit of the time-step is $\Delta t \leq \mathcal{O}(N^{-3/2})$ and a fixed number of Dirichlet problems have to be solved per time-step. By numerical experiments, we demonstrate that the approximate system requires less than 10% of the CPU time used by the original system to solve the problem very accurately. © 1996 Academic Press, Inc.

1. INTRODUCTION

In this paper we consider the two-dimensional approximation of a slender nonaxisymmetric three-dimensional jet subject to surface tension, where the evolution of the flow in the cross section of the jet is governed by the two-dimensional incompressible Euler equations inside of the free surface; cf. [7]. The velocity field is assumed to be irrotational, so there exists a velocity potential which satisfies Laplace's equation in the interior of the jet subject to Bernoulli's equation and the kinematic condition on the free surface. This type of potential flow problem with surface tension has been simulated numerically by many investigators; see, for example [3, 14]. The focus of the present work is to develop an alternative approach to [11] for reducing the great computational cost of integrating the governing equations numerically, which is caused by the stiffness of the discrete system due to surface tension.

We scale the physical problem by the length scale L such that the area of the scaled initial domain equals π, and by a time-scale $T = \sqrt{\rho L^3/\tau}$. Here, ρ is the density of the liquid and τ is the surface tension. This choice of time-scale makes the scaled surface tension equal to 1; i.e., the Weber number is set to one. Henceforth, only the scaled problem will be considered. Let x and y be the Cartesian coordinates in the cross-sectional plane of the jet, and let t be the time. We assume that the liquid occupies the simply connected time-dependent domain $\Omega(t)$ with a smooth boundary $\Gamma(t)$. It is convenient to describe the motion in the positively oriented Lagrangian coordinate $0 \leq \alpha \leq 2\pi$, such that the boundary at time $t \geq 0$ is given by $x = X(\alpha, t)$, $y = Y(\alpha, t)$ and the velocity potential on the boundary is $\phi = \phi(\alpha, t)$. The governing equations for ϕ, X, Y on $\Gamma(t)$ are

$$\phi_t = \tfrac{1}{2}(u^2 + v^2) - \kappa, \tag{1}$$

$$X_t = u, \tag{2}$$

$$Y_t = v, \tag{3}$$

for $t \geq 0$ subject to the 2π-periodic initial conditions $\phi(\alpha, 0) = \phi_0(\alpha)$, $X(\alpha, 0) = X_0(\alpha)$, and $Y(\alpha, 0) = Y_0(\alpha)$. The curvature of the boundary is

$$\kappa = \sigma^3(X_\alpha Y_{\alpha\alpha} - Y_\alpha X_{\alpha\alpha}), \quad \sigma = 1/\sqrt{X_\alpha^2 + Y_\alpha^2}, \tag{4}$$

and the boundary velocity components satisfy

$$u(\alpha, t) = \psi_x(X(\alpha, t), Y(\alpha, t)), \tag{5}$$

$$v(\alpha, t) = \psi_y(X(\alpha, t), Y(\alpha, t)), \tag{6}$$

where $\psi = \psi(x, y)$ is the solution of the Dirichlet problem

$$\Delta\psi = 0, \quad \text{in } \Omega(t), \tag{7}$$

$$\psi(X(\alpha, t), Y(\alpha, t)) = \phi(\alpha, t), \quad 0 \leq \alpha \leq 2\pi. \tag{8}$$

Since the shape of the domain depends on time, it is rather expensive to compute the boundary velocity by numerically solving (7), (8), followed by evaluating (5), (6). Instead we will apply an equivalent boundary integral formulation of the Dirichlet problem. For a derivation we

0021-9991/96 $18.00

refer to [2, 6]. In this method, we first compute the (unnormalized) vortex sheet strength $\gamma(\alpha, t)$ by solving the Fredholm integral equation of the second kind,

$$\phi_\alpha(\alpha, t) = \frac{\gamma(\alpha, t)}{2} + \mathrm{Re}\left\{\frac{Z_\alpha(\alpha, t)}{2\pi i} P.V. \int_0^{2\pi} \frac{\gamma(\alpha', t)}{Z(\alpha', t) - Z(\alpha, t)} d\alpha'\right\}, \tag{9}$$

where $Z(\alpha, t) = X(\alpha, t) + iY(\alpha, t)$. The velocity on the boundary is then computed by evaluating the integral

$$u(\alpha, t) - iv(\alpha, t) = \frac{\gamma(\alpha, t)}{2Z_\alpha(\alpha, t)} + \frac{1}{2\pi i} P.V. \int_0^{2\pi} \frac{\gamma(\alpha', t)}{Z(\alpha', t) - Z(\alpha, t)} d\alpha'. \tag{10}$$

Henceforth, we will denote Eqs. (1)–(4), (9), and (10) the full system.

The linearized motion of the related problem governing inviscid water waves subject to surface tension was analyzed by [6]. By projecting the surface coordinates onto the local normal and tangential components, it was found that the linearized equations have a tractable form which enabled the authors to prove well-posedness of the linearized system in a Sobolev space of finite order. This work was extended by [5] to the spatially discrete case, were stability and convergence was proven for a pseudo-spectral discretization of the nonlinear problem. It was found that the particular type of spatial discretization is very important in order for certain unbounded operators to cancel to highest order. A similar result was found by [4] who considered the linearized stability properties of different numerical schemes for a vortex sheet subject to surface tension close to equilibrium.

When the full system is discretized in space by the pseudo-spectral method, the highest spatial frequency on the grid is $\omega = N/2$, where N is the number of grid points in the discretization of the free surface. In Section 2, we will carry out a simplified version of the general linearization performed in [6, 13] to show that because of surface tension, the linearized equations have eigenvalues with imaginary part of the order $\mathcal{O}((\max_\alpha \sigma(\alpha, t)\omega)^{3/2})$. Hence, when the discretized equations are integrated in time, the time step is restricted by the stability constraint $\Delta t \leq \mathcal{O}((\max_\alpha \sigma(\alpha, t)N)^{-3/2})$. The stability limit of the time-step can become increasingly restrictive with time if σ becomes large, i.e., if the Lagrangian grid points cluster. The time-step restriction together with the clustering of grid points make the time-integration expensive, because the integral equation (9) must be solved and the velocity integral (10) must be calculated every time the right-hand side of (1)–(3) is evaluated. The cost of performing these tasks is of the order $\mathcal{O}(N^2)$ operations if direct summation is used or $\mathcal{O}(CN)$ operations, where C is a large constant, if the fast multipole method [9] is applied.

In the recent paper [11], a method was proposed to alleviate the stiffness of the more general equations governing the two-dimensional motion of two immiscible ideal fluids subject to surface tension. The governing equations were reformulated using the arc-length and tangent-angle variables, instead of the Cartesian coordinates X and Y to describe the position of the interface. It is shown that the curvature term becomes linear in terms of these variables. This property, together with a small-scale approximation of the velocity, enabled the authors to devise a fast implicit time-integration method to solve the reformulated equations in the case of zero density stratification between the fluids. The method has more recently also been extended to handle the presence of a finite density stratification across the interface.

As is noted in [11], the tangential velocity of the grid points is only dictated by the parameterization of the solution. It would, therefore, be possible to partially reduce the stiffness of the equations by using a non-Lagrangian parameterization of the solution and prescribing a tangential velocity that, for example, distributes the grid points uniformly with respect to arc-length. This possibility will not be investivated further here.

In the present paper, we propose an alternative approach to [11] to handle the stiffness imposed by surface tension. Here, we will take advantage of the leading order structure of the linearized equations proved by [6, 13] to derive an approximate system which can be integrated much faster than the full system. The approximate system is derived in the following way. Let $\overline{T}$ and $\overline{N}$ denote the unit tangent and normal vectors of the boundary. The velocity can be decomposed as

$$\overline{u} \equiv \begin{pmatrix} u \\ v \end{pmatrix} = (\overline{u} \cdot \overline{T})\overline{T} + (\overline{u} \cdot \overline{N})\overline{N}.$$

The tangential component of the velocity equals the tangential derivative of the velocity potential, i.e., $\overline{u} \cdot \overline{T} = \sigma\phi_\alpha$. However, the computation of the normal component $\overline{u} \cdot \overline{N}$ requires the solution of (9), (10), which we want to avoid. The velocity $\overline{u}$ depends on ϕ, X, and Y. By linearizing $\overline{u}$ around $\phi^{(0)}$, $X^{(0)}$, and $Y^{(0)}$ and letting $\phi = \phi^{(0)} + \varepsilon\phi'$, $X = X^{(0)} + \varepsilon X'$, $Y = Y^{(0)} + \varepsilon Y'$, we get

$$\overline{u} = \overline{u}^{(0)} + \varepsilon\overline{u}' + \mathcal{O}(\varepsilon^2), \tag{11}$$

$$\overline{u}' = \frac{\partial\overline{u}}{\partial\phi}\phi' + \frac{\partial\overline{u}}{\partial X}X' + \frac{\partial\overline{u}}{\partial Y}Y'. \tag{12}$$

After some analysis, which we defer to Section 3, we show that the tangential component of the perturbation of the velocity is

$$\overline{u}' \cdot \overline{T}^{(0)} = \sigma^{(0)}(\phi'_\alpha - u^{(0)}X'_\alpha - v^{(0)}Y'_\alpha). \tag{13}$$

We use the theory in [6, 13] to motivate the approximation of the normal component of the perturbation of the velocity,

$$\overline{u}' \cdot \overline{N}^{(0)} \approx \sigma^{(0)}H(\phi'_\alpha - u^{(0)}X'_\alpha - v^{(0)}Y'_\alpha), \tag{14}$$

where the operator H has Fourier symbol $i\,\mathrm{sgn}(\omega)$. In those papers, it is shown that the error in $\overline{u}' \cdot \overline{N}^{(0)}$ is a smoothing operator of the perturbations ϕ', X', and Y'. This means that the contribution to the error is dominated by the low frequency components of ϕ', X', and Y' and that the contribution from the high frequency components tend to zero as the frequency of the perturbation tends to infinity. The velocity is therefore approximately

$$\overline{u} \approx \overline{u}^{(0)} + \varepsilon\overline{T}^{(0)}(\overline{u}' \cdot \overline{T}^{(0)}) + \varepsilon\overline{N}^{(0)}(\overline{u}' \cdot \overline{N}^{(0)}).$$

Because the tangential component of the velocity is known exactly, we will only use the above expression for the normal component of the velocity. We have $\overline{T}^{(0)} \cdot \overline{N} = \mathcal{O}(\varepsilon)$, $\overline{N}^{(0)} \cdot \overline{N} = 1 + \mathcal{O}(\varepsilon^2)$, and $\sigma = \sigma^{(0)} + \mathcal{O}(\varepsilon)$, so it is consistent with the linearization to take the approximate velocity $\tilde{\overline{u}} \approx \overline{u}$ to be

$$\tilde{\overline{u}} \equiv \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} = \overline{T}(\sigma\phi_\alpha) + \overline{N}(\overline{u}^{(0)} \cdot \overline{N} + \varepsilon\sigma H(\phi'_\alpha - u^{(0)}X'_\alpha - v^{(0)}Y'_\alpha)). \tag{15}$$

In the approximate system we replace the exact velocity by the approximation (15). To compensate for the smooth error in (15), we also introduce a time-dependent forcing which will be derived in Section 3. We arrive at

$$\tilde{\phi}_t = \tfrac{1}{2}(\tilde{u}^2 + \tilde{v}^2) - \kappa + (t - t_0)G_1, \tag{16}$$

$$\tilde{X}_t = \tilde{u} + (t - t_0)G_2, \tag{17}$$

$$\tilde{Y}_t = \tilde{v} + (t - t_0)G_3, \tag{18}$$

for $t \geq t_0$ subject to the 2π-periodic initial conditions $\tilde{\phi}(\alpha, t_0) = \phi^{(0)}(\alpha)$, $\tilde{X}(\alpha, t_0) = X^{(0)}(\alpha)$, and $\tilde{Y}(\alpha, t_0) = Y^{(0)}(\alpha)$.

Computing the velocity $\overline{u}^{(0)}$ requires one solution of (9), (10), and calculating the forcing terms G_i, $i = 1, 2, 3$, requires another solution of those equations. Hence, after (9), (10) have been solved twice, the approximate system can be evolved without solving those equations. It should be noted that the stability restriction on the approximative system is similar to the full system, so the explicit time-step limit for the discretized version of (16), (17), and (18) is also of the order $\Delta t \leq \mathcal{O}((\max_\alpha \sigma(\alpha, t)N)^{-3/2})$. The significant advantage of the approximate system is that the right-hand side of (16)–(18) can be evaluated by computing spatial derivatives followed by applying H. Both of these tasks can be done very quickly by the pseudo spectral method which requires $\mathcal{O}(N \log(N))$ operations.

We start integrating the approximate system at $t_0 = 0$, where we take $\phi^{(0)} = \phi_0$, $X^{(0)} = X_0$, $Y^{(0)} = Y_0$. First we compute $\overline{u}^{(0)}$ and G_i, $i = 1, 2, 3$, to set up the approximate system. The system is then integrated in time until $t = t_0 + \Delta T$, after which the linearization of the velocity is redone around $\phi^{(0)}(\alpha) = \tilde{\phi}(\alpha, t_0 + \Delta T)$, $X^{(0)}(\alpha) = \tilde{X}(\alpha, t_0 + \Delta T)$, $Y^{(0)}(\alpha) = \tilde{Y}(\alpha, t_0 + \Delta T)$, and a new approximate system is formed. This procedure is repeated for as long as the equations need to be integrated. An error is committed by approximating the normal component of the velocity. We show in Section 3 that this error is of the order $\mathcal{O}(\Delta T^3)$, which implies that the error at a fixed time T is of the order $\mathcal{O}(\Delta T^2)$. We remark that the time-step between linearizing the velocity, ΔT, is different from the explicit time-step Δt, which is used when the approximate system is integrated in time from t_0 to $t_0 + \Delta T$.

The remainder of the paper is organized as follows. In Section 4 we discretize both the full and the approximate systems by the pseudo-spectral method in space. The details of the discretization are essential for the temporal stability of the spatially discrete problem. For the full system, which will be used to validate the approximate equations, we will use the discretization developed by [5]. In this method, the ‘‘alternating-point-trapezoidal’’ quadrature rule, together with a Fourier filter, is used to discrete the integral equation for the vortex sheet strength and the velocity integral. The curvature is discretized by Fourier-filtered spectral derivatives. A similar technique is used to discretize the approximate system. We close the section with a discussion of the stability and efficiency of different time-integration methods. In Section 5, we perform numerical experiments to investigate how the resolution affects the solution of the full system and to study the stability and accuracy of the approximate method. Some concluding remarks are made in Section 6.

2. ANALYSIS

The purpose of the analysis presented here is to estimate the eigenvalues of the linearized operator with frozen coefficients. This information is necessary for selecting an appropriate time-integration method and for estimating the stability constraint of the time step. For simplicity, we only consider a special case. We refer to [6, 13] for a derivation of the linearized operator in the general case.

To perform the analysis, we find it convenient to use the original formulation of the boundary velocity (5)–(8).

We begin by linearizing (1)–(8) around a solution $\phi^{(0)}(\alpha, t)$, $X^{(0)}(\alpha, t)$, and $Y^{(0)}(\alpha, t)$. Denote the velocity components by

$$u^{(0)} = u[\phi^{(0)}, X^{(0)}, Y^{(0)}], \tag{19}$$

$$v^{(0)} = v[\phi^{(0)}, X^{(0)}, Y^{(0)}]. \tag{20}$$

Let $\phi^{(\varepsilon)} = \phi^{(0)} + \varepsilon\phi'$, $X^{(\varepsilon)} = X^{(0)} + \varepsilon X'$, and $Y^{(\varepsilon)} = Y^{(0)} + \varepsilon Y'$, where $0 < \varepsilon \ll 1$. We have

$$u[\phi^{(\varepsilon)}, X^{(\varepsilon)}, Y^{(\varepsilon)}] = u^{(0)} + \varepsilon u' + \mathcal{O}(\varepsilon^2), \tag{21}$$

$$v[\phi^{(\varepsilon)}, X^{(\varepsilon)}, Y^{(\varepsilon)}] = v^{(0)} + \varepsilon v' + \mathcal{O}(\varepsilon^2), \tag{22}$$

with

$$u' = \lim_{\varepsilon\to 0} \frac{1}{\varepsilon}(u[\phi^{(\varepsilon)}, X^{(\varepsilon)}, Y^{(\varepsilon)}] - u^{(0)}), \tag{23}$$

$$v' = \lim_{\varepsilon\to 0} \frac{1}{\varepsilon}(v[\phi^{(\varepsilon)}, X^{(\varepsilon)}, Y^{(\varepsilon)}] - v^{(0)}). \tag{24}$$

Neglecting the $\mathcal{O}(\varepsilon^2)$ terms yields the following linear problem for the perturbations:

$$\phi'_t = u^{(0)}u' + v^{(0)}v' - \kappa', \tag{25}$$

$$X'_t = u', \tag{26}$$

$$Y'_t = v'. \tag{27}$$

The perturbation of the curvature is

$$\begin{aligned}\kappa' = (\sigma^{(0)})^3(Y'_{\alpha\alpha}X^{(0)}_\alpha - X'_{\alpha\alpha}Y^{(0)}_\alpha + X'_\alpha Y^{(0)}_{\alpha\alpha} - Y'_\alpha X^{(0)}_{\alpha\alpha})\\ - 3\kappa^{(0)}(\sigma^{(0)})^2(X'_\alpha X^{(0)}_\alpha + Y'_\alpha Y^{(0)}_\alpha).\end{aligned} \tag{28}$$

In this expression $\kappa^{(0)}$ is the curvature of the unperturbed boundary and the normalization factor is

$$\sigma^{(0)} = 1/\sqrt{(X^{(0)}_\alpha)^2 + (Y^{(0)}_\alpha)^2}. \tag{29}$$

Because u and v are linear in ϕ, it is clear that the perturbations of u and v can be split according to

$$\begin{aligned}u[\phi^{(\varepsilon)}, X^{(\varepsilon)}, Y^{(\varepsilon)}] = \varepsilon u[\phi', X^{(0)}, Y^{(0)}] + u[\phi^{(0)}, X^{(\varepsilon)}, Y^{(\varepsilon)}]\\ + \mathcal{O}(\varepsilon^2),\end{aligned} \tag{30}$$

$$\begin{aligned}v[\phi^{(\varepsilon)}, X^{(\varepsilon)}, Y^{(\varepsilon)}] = \varepsilon v[\phi', X^{(0)}, Y^{(0)}] + v[\phi^{(0)}, X^{(\varepsilon)}, Y^{(\varepsilon)}]\\ + \mathcal{O}(\varepsilon^2).\end{aligned} \tag{31}$$

The first term on the right-hand sides of (30) and (31) corresponds to a perturbation of the boundary value only and the second term is the contribution from perturbing the shape of the domain only. For the first term, we have $u[\phi', X^{(0)}, Y^{(0)}] = \psi^I_x(X^{(0)}, Y^{(0)})$ and $v[\phi', X^{(0)}, Y^{(0)}] = \psi^I_y(X^{(0)}, Y^{(0)})$, where ψ^I is the solution of

$$\Delta\psi^I = 0, \quad \text{in } \Omega^{(0)}, \tag{32}$$

$$\psi^I(X^{(0)}(\alpha, t), Y^{(0)}(\alpha, t)) = \phi'(\alpha, t), \quad 0 \le \alpha \le 2\pi, \tag{33}$$

Here, $\Omega^{(0)}$ corresponds to the domain interior to the boundary $x = X^{(0)}(\alpha, t)$, $y = Y^{(0)}(\alpha, t)$.

The second term of the right-hand sides of (30) and (31) can be written as

$$u[\phi^{(0)}, X^{(\varepsilon)}, Y^{(\varepsilon)}] = \psi^{II}_x(X^{(\varepsilon)}, Y^{(\varepsilon)}),$$

and a corresponding expression for v, where ψ^{II} is the solution of

$$\Delta\psi^{II} = 0, \quad \text{in } \Omega^{(\varepsilon)}, \tag{34}$$

$$\psi^{II}(X^{(\varepsilon)}(\alpha, t), Y^{(\varepsilon)}(\alpha, t)) = \phi^{(0)}(\alpha, t), \quad 0 \le \alpha \le 2\pi. \tag{35}$$

Here $\Omega^{(\varepsilon)}$ corresponds to the domain interior to the perturbed boundary $x = X^{(\varepsilon)}(\alpha, t)$, $y = Y^{(\varepsilon)}(\alpha, t)$.

In the general case, the term ψ^{II} is difficult to analyze, because it is not trivial to estimate how the normal derivative of ψ^{II} depends on the perturbation of the shape. To circumvent this problem, we will restrict the analysis to the special case when

$$\phi^{(0)}(\alpha, t) = C - \frac{t}{R}, \quad C = \text{const}, \tag{36}$$

$$X^{(0)}(\alpha, t) = R\cos\alpha, \tag{37}$$

$$Y^{(0)}(\alpha, t) = R\sin\alpha, \tag{38}$$

which is a solution of (1)–(3). Now, $\nabla\psi^{II} = 0$ and it is sufficient to study the contribution from ψ^I.

Let (r, α) be polar coordinates, i.e., $x = r\cos\alpha$, $y = r\sin\alpha$. The Dirichlet problem on a circular domain with radius R,

$$\Delta\hat\psi = 0, \qquad 0 \le r \le R, 0 \le \alpha \le 2\pi, \tag{39}$$

$$\hat\psi = e^{i\omega\alpha}, \quad 0 \le r \le R, 0 \le \alpha \le 2\pi, \tag{40}$$

where $\omega = 0, \pm1, \pm2, \ldots$, is solved by

$$\hat\psi(r, \alpha) = \left(\frac{r}{R}\right)^{|\omega|} e^{i\omega\alpha}. \tag{41}$$

On the boundary $r = R$, we have $\hat\psi_r = |\omega|e^{i\omega\alpha}/R$ and $\hat\psi_\alpha = i\omega e^{i\omega\alpha}$. Therefore, the relation between the inward normal and tangential derivatives becomes

$$\hat{\psi}_n = i\,\mathrm{sgn}(\omega)\hat{\psi}_s, \tag{42}$$

where s is the arc-length. This formula defines the Fourier symbol for the relation between the inward normal and tangential derivative for general boundary data,

$$\psi^I_n(\alpha) = H(\psi^I_s)(\alpha). \tag{43}$$

Hence, by transforming the tangential and normal derivatives of ψ to the x and y directions, we arrive at

$$\psi^I_x = \sigma^{(0)}(X^{(0)}_\alpha - Y^{(0)}_\alpha H)\,\sigma^{(0)}\phi'_\alpha, \quad r = R, \tag{44}$$

$$\psi^I_y = \sigma^{(0)}(Y^{(0)}_\alpha + X^{(0)}_\alpha H)\,\sigma^{(0)}\phi'_\alpha, \quad r = R. \tag{45}$$

The system (25)–(27) now takes the form

$$\begin{aligned}\phi'_t &= -(\sigma^{(0)})^3(Y'_{\alpha\alpha}X^{(0)}_\alpha - X'_{\alpha\alpha}Y^{(0)}_\alpha + X'_\alpha Y^{(0)}_{\alpha\alpha} - Y'_\alpha X^{(0)}_{\alpha\alpha})\\ &\quad + 3\kappa^{(0)}(\sigma^{(0)})^2(X'_\alpha X^{(0)}_\alpha + Y'_\alpha Y^{(0)}_\alpha), \qquad (46)\\ X'_t &= \sigma^{(0)}(X^{(0)}_\alpha - Y^{(0)}_\alpha H)\,\sigma^{(0)}\phi'_\alpha, \qquad (47)\\ Y'_t &= \sigma^{(0)}(Y^{(0)}_\alpha + X^{(0)}_\alpha H)\,\sigma^{(0)}\phi'_\alpha, \qquad (48)\end{aligned}$$

Freezing the coefficients and Fourier-transforming the dependent variables yields

$$\frac{\partial}{\partial t}\begin{pmatrix}\hat{\phi}'\\ \hat{X}'\\ \hat{Y}'\end{pmatrix} = A\begin{pmatrix}\hat{\phi}'\\ \hat{X}'\\ \hat{Y}'\end{pmatrix}, \tag{49}$$

where

$$A = (\sigma^{(0)})^2\begin{pmatrix}0 & -a_{12}\omega^2 + b_{12}i\omega & -a_{13}\omega^2 + b_{13}i\omega\\ b_{21}i\omega & 0 & 0\\ b_{31}i\omega & 0 & 0\end{pmatrix}. \tag{50}$$

The coefficients are

$$\begin{aligned}a_{12} &= \sigma^{(0)}Y^{(0)}_\alpha,\\ a_{13} &= -\sigma^{(0)}X^{(0)}_\alpha,\\ b_{12} &= 3\kappa^{(0)}X^{(0)}_\alpha - \sigma^{(0)}Y^{(0)}_{\alpha\alpha},\\ b_{13} &= 3\kappa^{(0)}Y^{(0)}_\alpha + \sigma^{(0)}X^{(0)}_{\alpha\alpha},\\ b_{21} &= X^{(0)}_\alpha - Y^{(0)}_\alpha\, i\,\mathrm{sgn}(\omega),\\ b_{31} &= Y^{(0)}_\alpha + X^{(0)}_\alpha\, i\,\mathrm{sgn}(\omega),\end{aligned}$$

The eigenvalues of A are the roots of $\det(A - \lambda I) = 0$, where

$$\begin{aligned}\det(A - \lambda I) = -\lambda^3 + \lambda i\omega(\sigma^{(0)})^4((-a_{12}\omega^2 + b_{12}i\omega)b_{21}\\ + (-a_{13}\omega^2 + b_{13}i\omega)b_{31}).\end{aligned} \tag{51}$$

One root is $\lambda^{(1)} = 0$ and after some algebra we find that the other two roots are the solutions of

$$\lambda^2 = -(\sigma^{(0)})^3|\omega|^3 - 2\kappa^{(0)}(\sigma^{(0)})^2\omega^2 + (\sigma^{(0)})^2\sigma^{(0)}_\alpha i\omega|\omega|. \tag{52}$$

In our case, $\kappa^{(0)} = 1/R$, $\sigma^{(0)} = 1/R$, so $\sigma^{(0)}_\alpha = 0$, and the solutions of (52) become purely imaginary:

$$\lambda^{(2,3)} = \pm i(\sigma^{(0)}|\omega|)^{3/2}\sqrt{1 + 2/|\omega|}. \tag{53}$$

3. THE APPROXIMATE SYSTEM

When the discretized version of the full system is integrated in time, the boundary velocity (u, v) must be evaluated a constant number of times per time step, where the constant depends on the time-integration method. Each evaluation requires the solution of the integral equation (9) for the vortex sheet strength followed by calculating the velocity integral (10). Even though the vortex sheet strength changes very little between each time step and iterative methods can be constructed that only require a few iterations to find the solution, we have found that the bulk of the computation consists of solving (9) and (10). In this section we derive the approximate system which can be integrated much faster than the full system because (9), (10) can be solved less frequently. The approximation is based on linearizing the velocity terms in the governing equations followed by approximating the solution of Dirichlet's problem.

We will estimate the error in the solution of the approximate system in the L_2-norm, which we define for 2π-periodic vector functions $\overline{F} = (F_1, F_2, F_3)^{\mathrm{T}}$ and $\overline{G} = (G_1, G_2, G_3)^{\mathrm{T}}$, according to

$$\|\overline{G}\|_2 = \langle\overline{G}, \overline{G}\rangle_2^{1/2}, \quad \langle\overline{F}, \overline{G}\rangle_2 = \int_0^{2\pi}\sum_{i=0}^{3} F^*_i(\alpha)G_i(\alpha)\,d\alpha. \tag{54}$$

We proceed by improving the analysis of the linearized velocity (12) to allow for a nonconstant $\phi^{(0)}(\alpha)$ and include effects from the term $\nabla\psi^{II}$, defined by (34), (35). As in [6, 13], we find it convenient to decompose $\overline{u}'$ into its normal and tangential components. Let the unperturbed inward unit normal vectors be $\overline{N}^{(0)}$ and the unperturbed unit tangent vector be $\overline{T}^{(0)}$. In terms of $X^{(0)}$ and $Y^{(0)}$,

$$\overline{T}^{(0)} = \sigma^{(0)}\begin{pmatrix} X_\alpha^{(0)} \\ Y_\alpha^{(0)} \end{pmatrix}, \quad \overline{N}^{(0)} = \sigma^{(0)}\begin{pmatrix} -Y_\alpha^{(0)} \\ X_\alpha^{(0)} \end{pmatrix}. \tag{55}$$

We easily get from (23) and (24),

$$\overline{u}' \cdot \overline{T}^{(0)} = \nabla\psi^I(X^{(0)}, Y^{(0)}) \cdot \overline{T}^{(0)} + \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}(\nabla\psi^{II}(X, Y)\cdot \overline{T}^{(0)} - \overline{u}^{(0)} \cdot \overline{T}^{(0)}) \tag{56}$$

$$\overline{u}' \cdot \overline{N}^{(0)} = \nabla\psi^I(X^{(0)}, Y^{(0)}) \cdot \overline{N}^{(0)} + \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}(\nabla\psi^{II}(X, Y)\cdot \overline{N}^{(0)} - \overline{u}^{(0)} \cdot \overline{N}^{(0)}). \tag{57}$$

The gradient of ψ^{II} can be written as

$$\nabla\psi^{II} = \frac{\partial\psi^{II}}{\partial s}\overline{T} + \frac{\partial\psi^{II}}{\partial n}\overline{N}, \tag{58}$$

with $\partial/\partial s$ being the derivative with respect to arc-length and $\partial/\partial n$ being the inward normal derivative. Straightforward algebra yields

$$\sigma = \sigma^{(0)} - \varepsilon(\sigma^{(0)})^2\overline{T}^{(0)} \cdot \begin{pmatrix} X'_\alpha \\ Y'_\alpha \end{pmatrix} + \mathcal{O}(\varepsilon^2) \tag{59}$$

and

$$\overline{T} \cdot \overline{T}^{(0)} = 1 + \mathcal{O}(\varepsilon^2), \tag{60}$$

$$\overline{N} \cdot \overline{T}^{(0)} = -\varepsilon\sigma^{(0)}\,\overline{N}^{(0)} \cdot \begin{pmatrix} X'_\alpha \\ Y'_\alpha \end{pmatrix} + \mathcal{O}(\varepsilon^2). \tag{61}$$

We now study the tangential component $\overline{u}' \cdot \overline{T}^{(0)}$. We have

$$\overline{u}^{(0)} \cdot \overline{T}^{(0)} = \sigma^{(0)}\phi_\alpha^{(0)}, \frac{\partial\psi^{II}}{\partial n} = \overline{u}^{(0)} \cdot \overline{N}^{(0)} + \mathcal{O}(\varepsilon) \tag{62}$$

and, because ψ^{II} satisfies (35),

$$\frac{\partial\psi^{II}}{\partial s} = \sigma\phi_\alpha^{(0)}. \tag{63}$$

Hence, (56) and (58) yield

$$\begin{aligned}\overline{u} \cdot \overline{T}^{(0)} &= \nabla\psi^I \cdot \overline{T}^{(0)} + \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}(\sigma\phi_\alpha^{(0)}\overline{T}\cdot\overline{T}^{(0)} \\ &\quad + (\overline{u}^{(0)} \cdot \overline{N}^{(0)} + \mathcal{O}(\varepsilon))\overline{N}\cdot\overline{T}^{(0)} - \sigma^{(0)}\phi_\alpha^{(0)}) \\ &= \sigma^{(0)}\left(\phi'_\alpha - ((\overline{u}^{(0)}\cdot\overline{T}^{(0)})\overline{T}^{(0)} + (\overline{u}^{(0)}\cdot\overline{N}^{(0)})\overline{N}^{(0)}) \cdot \begin{pmatrix} X'_\alpha \\ Y'_\alpha \end{pmatrix}\right) \\ &= \sigma^{(0)}(\phi'_\alpha - u^{(0)}X'_\alpha - v^{(0)}Y'_\alpha).\end{aligned} \tag{64}$$

We remark that only geometrical arguments were used in the derivation of $\overline{u}' \cdot \overline{T}^{(0)}$. Hence, ψ^I and ψ^{II} being solutions of Laplace's equation has no bearing on the form of $\overline{u}' \cdot \overline{T}^{(0)}$.

The normal component $\overline{u}' \cdot \overline{N}^{(0)}$ is more difficult to analyze, because it requires knowledge of how the normal derivative of the solution of Dirichlet's problem depends on the shape of the domain Ω. Without analysis, but recalling from (43) the exact relation between the inward normal derivative and the tangential derivative when the domain is circular, we will approximate $\overline{u}' \cdot \overline{N}^{(0)}$ by

$$\overline{u}' \cdot \overline{N}^{(0)} \approx \sigma^{(0)}H(\phi'_\alpha - u^{(0)}X'_\alpha - v^{(0)}Y'_\alpha). \tag{65}$$

It can be shown that the error in the approximate normal velocity is a smoothing operator of ϕ', X', and Y'; cf. [6, 13]. The linearized velocity (12) satisfies

$$\overline{u}' = \overline{T}^{(0)}(\overline{u}' \cdot \overline{T}^{(0)}) + \overline{N}^{(0)}(\overline{u}' \cdot \overline{N}^{(0)}).$$

Because the tangential component of the velocity is known exactly, we will only use the above expression for the normal component of the velocity. We have $\overline{T}^{(0)} \cdot \overline{N} = \mathcal{O}(\varepsilon)$, $\overline{N}^{(0)} \cdot \overline{N} = 1 + \mathcal{O}(\varepsilon^2)$, and $\sigma = \sigma^{(0)} + \mathcal{O}(\varepsilon)$, so it is consistent with the linearization to take the approximate velocity $\tilde{\overline{u}} \approx \overline{u}$ to be

$$\tilde{\overline{u}} \equiv \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} = \overline{T}(\sigma\phi_\alpha) + \overline{N}(\overline{u}^{(0)} \cdot \overline{N} + \varepsilon\sigma H(\phi'_\alpha - u^{(0)}X'_\alpha - v^{(0)}Y'_\alpha)). \tag{66}$$

In the approximate system we replace the exact velocity by the approximation (66). To compensate for the smooth error in (66), we also introduce a time-dependent forcing which we will present below. This results in the system (16)–(18).

We will derive the optimal form of the forcing G_i, $i = 1, 2, 3$, and estimate the error in the solution of the approximate system by solving both the full and approximate systems by asymptotic expansions in time. We start by considering the full system for $t \geq t_0$ subject to the initial conditions,

$$\phi(\alpha, t_0) = \phi^{(0)}(\alpha), \tag{67}$$

$$X(\alpha, t_0) = X^{(0)}(\alpha), \tag{68}$$

$$Y(\alpha, t_0) = Y^{(0)}(\alpha). \tag{69}$$

For t: $0 \leq t - t_0 \ll 1$, we make the ansatz

$$\begin{aligned} \phi(\alpha, t) &= \phi^{(0)}(\alpha) + \sum_{k=1}^{p} \frac{1}{k}(t-t_0)^k \phi^{(k)}(\alpha) + \mathcal{O}((t-t_0)^{p+1}), \\ X(\alpha, t) &= X^{(0)}(\alpha) + \sum_{k=1}^{p} \frac{1}{k}(t-t_0)^k X^{(k)}(\alpha) + \mathcal{O}((t-t_0)^{p+1}), \\ Y(\alpha, t) &= Y^{(0)}(\alpha) + \sum_{k=1}^{p} \frac{1}{k}(t-t_0)^k Y^{(k)}(\alpha) + \mathcal{O}((t-t_0)^{p+1}). \end{aligned} \tag{70}$$

The velocity satisfies (11) and by linearizing the curvature, we get

$$\kappa = \kappa^{(0)} + \varepsilon\kappa' + \mathcal{O}(\varepsilon^2), \tag{71}$$

$$\kappa' = \frac{\partial\kappa}{\partial X} X' + \frac{\partial\kappa}{\partial Y} Y'. \tag{72}$$

Hence, inserting (70) into both (11) and (71) yields

$$\overline{u} = \overline{u}^{(0)} + (t-t_0)\left\{\frac{\partial\overline{u}}{\partial\phi}\phi^{(1)} + \frac{\partial\overline{u}}{\partial X}X^{(1)} + \frac{\partial\overline{u}}{\partial Y}Y^{(1)}\right\} + \mathcal{O}((t-t_0)^2), \tag{73}$$

$$\kappa = \kappa^{(0)} + (t-t_0)\left\{\frac{\partial\kappa}{\partial X}X^{(1)} + \frac{\partial\kappa}{\partial Y}Y^{(1)}\right\} + \mathcal{O}((t-t_0)^2). \tag{74}$$

We determine the functions $\phi^{(k)}$, $X^{(k)}$, $Y^{(k)}$ by inserting (70), (73), and (74) into the system (1)–(3) and identifying the terms with the same power in $t - t_0$. This results in

$$\phi^{(1)} = \tfrac{1}{2}((u^{(0)})^2 + (v^{(0)})^2) - \kappa^{(0)}, \tag{75}$$

$$X^{(1)} = u^{(0)}, \tag{76}$$

$$Y^{(1)} = v^{(0)}. \tag{77}$$

The $\mathcal{O}((t - t_0)^2)$ terms are

$$\begin{aligned} \phi^{(2)} = u^{(0)}&\left\{\frac{\partial u}{\partial\phi}\phi^{(1)} + \frac{\partial u}{\partial X}X^{(1)} + \frac{\partial u}{\partial Y}Y^{(1)}\right\} \\ &+ v^{(0)}\left\{\frac{\partial v}{\partial\phi}\phi^{(1)} + \frac{\partial v}{\partial X}X^{(1)} + \frac{\partial v}{\partial Y}Y^{(1)}\right\} \\ &- \frac{\partial\kappa}{\partial X}X^{(1)} - \frac{\partial\kappa}{\partial Y}Y^{(1)}, \end{aligned} \tag{78}$$

$$X^{(2)} = \frac{\partial u}{\partial\phi}\phi^{(1)} + \frac{\partial u}{\partial X}X^{(1)} + \frac{\partial u}{\partial Y}Y^{(1)}, \tag{79}$$

$$Y^{(2)} = \frac{\partial v}{\partial\phi}\phi^{(1)} + \frac{\partial v}{\partial X}X^{(1)} + \frac{\partial v}{\partial Y}Y^{(1)}. \tag{80}$$

For the approximate system, we make the corresponding ansatz,

$$\begin{aligned} \tilde{\phi}(\alpha, t) &= \phi^{(0)}(\alpha) + \sum_{k=1}^{p} \frac{1}{k}(t-t_0)^k \tilde{\phi}^{(k)}(\alpha) + \mathcal{O}((t-t_0)^{p+1}), \\ \tilde{X}(\alpha, t) &= X^{(0)}(\alpha) + \sum_{k=1}^{p} \frac{1}{k}(t-t_0)^k \tilde{X}^{(k)}(\alpha) + \mathcal{O}((t-t_0)^{p+1}), \\ \tilde{Y}(\alpha, t) &= Y^{(0)}(\alpha) + \sum_{k=1}^{p} \frac{1}{k}(t-t_0)^k \tilde{Y}^{(k)}(\alpha) + \mathcal{O}((t-t_0)^{p+1}). \end{aligned} \tag{81}$$

Proceeding in the same way as for the full system, the terms in the asymptotic expansion for the solution of the approximate system become

$$\tilde{\phi}^{(1)} = \tfrac{1}{2}((u^{(0)})^2 + (v^{(0)})^2) - \kappa^{(0)}, \tag{82}$$

$$\tilde{X}^{(1)} = u^{(0)}, \tag{83}$$

$$\tilde{Y}^{(1)} = v^{(0)}. \tag{84}$$

The $\mathcal{O}((t - t_0)^2)$ terms are

$$\begin{aligned} \tilde{\phi} = u^{(0)}&\left\{\frac{\partial\tilde{u}}{\partial\phi}\phi^{(1)} + \frac{\partial\tilde{u}}{\partial X}X^{(1)} + \frac{\partial\tilde{u}}{\partial Y}Y^{(1)}\right\} \\ &+ v^{(0)}\left\{\frac{\partial\tilde{v}}{\partial\phi}\phi^{(1)} + \frac{\partial\tilde{v}}{\partial X}X^{(1)} + \frac{\partial\tilde{v}}{\partial Y}Y^{(1)}\right\} \\ &- \frac{\partial\kappa}{\partial X}X^{(1)} - \frac{\partial\kappa}{\partial Y}Y^{(1)} + G_1, \end{aligned} \tag{85}$$

$$\tilde{X}^{(2)} = \frac{\partial\tilde{u}}{\partial\phi}\phi^{(1)} + \frac{\partial\tilde{u}}{\partial X}X^{(1)} + \frac{\partial\tilde{u}}{\partial Y}Y^{(1)} + G_2, \tag{86}$$

$$\tilde{Y}^{(2)} = \frac{\partial\tilde{v}}{\partial\phi}\phi^{(1)} + \frac{\partial\tilde{v}}{\partial X}X^{(1)} + \frac{\partial\tilde{v}}{\partial Y}Y^{(1)} + G_3. \tag{87}$$

By comparing the terms in the asymptotic expansions, we see that $\phi^{(1)} = \tilde{\phi}^{(1)}$, $X^{(1)} = \tilde{X}^{(1)}$, and $Y^{(1)} = \tilde{Y}^{(1)}$. Furthermore, the $\mathcal{O}((t - t_0)^2)$ terms will be identical if we take the forcing to be

$$\begin{aligned}
G_1 = u^{(0)} \bigg\{ &\frac{\partial u}{\partial \phi} \phi^{(1)} + \frac{\partial u}{\partial X} X^{(1)} + \frac{\partial u}{\partial Y} Y^{(1)} - \frac{\partial \tilde{u}}{\partial \phi} \phi^{(1)} \\
&- \frac{\partial \tilde{u}}{\partial X} X^{(1)} - \frac{\partial \tilde{u}}{\partial Y} Y^{(1)} \bigg\} \\
+ v^{(0)} \bigg\{ &\frac{\partial v}{\partial \phi} \phi^{(1)} + \frac{\partial v}{\partial X} X^{(1)} + \frac{\partial v}{\partial Y} Y^{(1)} - \frac{\partial \tilde{v}}{\partial \phi} \phi^{(1)} \\
&- \frac{\partial \tilde{v}}{\partial X} X^{(1)} - \frac{\partial \tilde{v}}{\partial Y} Y^{(1)} \bigg\}, \qquad (88)
\end{aligned}$$

$$\begin{aligned}
G_2 = &\frac{\partial u}{\partial \phi} \phi^{(1)} + \frac{\partial u}{\partial X} X^{(1)} + \frac{\partial u}{\partial Y} Y^{(1)} \\
&- \frac{\partial \tilde{u}}{\partial \phi} \phi^{(1)} - \frac{\partial \tilde{u}}{\partial X} X^{(1)} - \frac{\partial \tilde{u}}{\partial Y} Y^{(1)}, \qquad (89)
\end{aligned}$$

$$\begin{aligned}
G_3 = &\frac{\partial v}{\partial \phi} \phi^{(1)} + \frac{\partial v}{\partial X} X^{(1)} + \frac{\partial v}{\partial Y} Y^{(1)} \\
&- \frac{\partial \tilde{v}}{\partial \phi} \phi^{(1)} - \frac{\partial \tilde{v}}{\partial X} X^{(1)} - \frac{\partial \tilde{v}}{\partial Y} Y^{(1)}. \qquad (90)
\end{aligned}$$

By this choice of forcing, the difference will satisfy $(\phi - \tilde{\phi}, X - \tilde{X}, Y - \tilde{Y})^{\mathrm{T}} = \mathcal{O}((t - t_0)^3)$. We remark that the forcing terms are easily computed by numerical differentiation, i.e.,

$$\begin{aligned}
&\frac{\partial u}{\partial \phi} \phi^{(1)} + \frac{\partial u}{\partial X} X^{(1)} + \frac{\partial u}{\partial Y} Y^{(1)} \\
&\quad = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon} (u[\phi^{(0)} + \varepsilon \phi^{(1)}, X^{(0)} + \varepsilon X^{(1)}, Y^{(0)} + \varepsilon Y^{(1)}] - u^{(0)}),
\end{aligned}$$

and a corresponding expression for the v component of the velocity. In the practical computation, ε is taken to be a small positive number.

The approximate system will be integrated up to time $t = t_0 + \Delta T$, after which the linearization of the velocity is redone and a new approximate system is formed. The error in the approximate solution at time $t_0 + \Delta T$ can be estimated in the following way. Lets denote the solution of the approximate system at time $t_0 + \Delta T$ by $\tilde{\phi}(t_0 + \Delta T)$, $\tilde{X}(t_0 + \Delta T)$, and $\tilde{Y}(t_0 + \Delta T)$. From the previous analysis, we have that the error in that solution is of the order $\mathcal{O}(\Delta T^3)$. Hence, the exact velocity at that time can be approximated by

$$\begin{aligned}
\bar{u}(t_0 + \Delta T) &\equiv \bar{u}[\tilde{\phi}(t_0 + \Delta T), \tilde{X}(t_0 + \Delta T), \tilde{Y}(t_0 + \Delta T)] \\
&= \bar{u}[\phi(t_0 + \Delta T), X(t_0 + \Delta T), Y(t_0 + \Delta T)] \qquad (91) \\
&\quad + \mathcal{O}(\Delta T^3).
\end{aligned}$$

In the same way, the curvature satisfies

$$\begin{aligned}
\kappa[X(t_0 + \Delta T), Y(t_0 + \Delta T)] = \kappa[\tilde{X}(t_0 + \Delta T), \tilde{Y}(t_0 + \Delta T)] \\
+ \mathcal{O}(\Delta T^3).
\end{aligned}$$

We also evaluate the approximate velocity

$$\tilde{\bar{u}}(t_0 + \Delta T) \equiv \tilde{\bar{u}}[\tilde{\phi}(t_0 + \Delta T), \tilde{X}(t_0 + \Delta T), \tilde{Y}(t_0 + \Delta T)].$$

At time $t_0 + \Delta T$, the time-derivative of the error therefore satisfies

$$\begin{aligned}
(\phi - \tilde{\phi})_t &= \tfrac{1}{2} |\bar{u}(t_0 + \Delta T)|^2 - \tfrac{1}{2} |\tilde{\bar{u}}(t_0 + \Delta T)|^2 - \Delta T G_1 \\
&\quad + \mathcal{O}(\Delta T^3), \qquad (92) \\
(X - \tilde{X})_t &= u(t_0 + \Delta T) - \tilde{u}(t_0 + \Delta T) - \Delta T G_2 \\
&\quad + \mathcal{O}(\Delta T^3), \qquad (93) \\
(Y - \tilde{Y})_t &= v(t_0 + \Delta T) - \tilde{v}(t_0 + \Delta T) - \Delta T G_3 \\
&\quad + \mathcal{O}(\Delta T^3), \qquad (94)
\end{aligned}$$

Hence, the difference between the third-order terms in the asymptotic expansions fulfill

$$\begin{aligned}
\phi^{(3)} - \tilde{\phi}^{(3)} &= \frac{1}{\Delta T^2} \bigg(\frac{1}{2} |\bar{u}(t_0 + \Delta T)|^2 - \frac{1}{2} |\tilde{\bar{u}}(t_0 + \Delta T)|^2 \\
&\quad - \Delta T G_1 \bigg) := D_1, \qquad (95) \\
X^{(3)} - \tilde{X}^{(3)} &= \frac{1}{\Delta T^2} (u(t_0 + \Delta T) - \tilde{u}(t_0 + \Delta T) \\
&\quad - \Delta T G_2) := D_2, \qquad (96) \\
Y^{(3)} - \tilde{Y}^{(3)} &= \frac{1}{\Delta T^2} (v(t_0 + \Delta T) - \tilde{v}(t_0 + \Delta T) \\
&\quad - \Delta T G_3 := D_3, \qquad (97)
\end{aligned}$$

as $\Delta T \to 0$. By neglecting the fourth-order terms in the asymptotic expansions, the error at time $t_0 + \Delta T$ approximately satisfies

$$\|\bar{e}\|_2 \approx \frac{\Delta T^3 \|\bar{D}\|_2}{3}, \quad \bar{e} = \bar{\Phi} - \tilde{\bar{\Phi}}, \quad \bar{\Phi} = \begin{pmatrix} \phi \\ X \\ Y \end{pmatrix},$$

$$\tilde{\bar{\Phi}} = \begin{pmatrix} \tilde{\phi} \\ \tilde{X} \\ \tilde{Y} \end{pmatrix}, \quad \bar{D} = \begin{pmatrix} D_1 \\ D_2 \\ D_3 \end{pmatrix}.$$

If we assume $\|\bar{D}\|_2$ to vary on a time-scale much slower than ΔT, we can use this estimate to adaptively adjust the next ΔT to keep the error in the approximate system on a constant level. We enforce the relative error $\|\bar{e}\|_2/\|\tilde{\bar{\Phi}}\|_2 \leq \delta$ by taking

$$\Delta T = \delta^{1/3} \left(\frac{3\|\tilde{\bar{\Phi}}\|_2}{\|\bar{D}\|_2} \right)^{1/3}. \qquad (98)$$

To integrate the solution from time 0 to T, the approximate system must be formed $N_T = T/\Delta T$ times. The local errors in each time interval will accumulate to a global relative error at time T, $E(T) \equiv \|\bar{\Phi}(T) - \tilde{\bar{\Phi}}(T)\|_2/\|\tilde{\bar{\Phi}}(T)\|_2$, which can be expected to be of the order

$$E(T) = \mathscr{O}(N_T \delta) = \mathscr{O}\left(T\delta^{2/3} \left(\frac{\|\bar{D}\|_2}{3\|\tilde{\bar{\Phi}}\|_2} \right)^{1/3} \right). \qquad (99)$$

During the integration up to time T, the integral equation (9) and the velocity integral (10) must be solved $2N_T$ times. The time-step in the explicit integration of the approximate system is independent of δ so the effort in solving the approximate system is of the order

$$2TC_D(N)\delta^{-1/3} \left(\frac{\|\bar{D}\|_2}{3\|\tilde{\bar{\Phi}}\|_2} \right)^{1/3} + TC_A(N), \qquad (100)$$

where $C_D(N)$ denotes the cost of solving (9), (10) and $C_A(N)$ is the cost of integrating the approximate system per unit-time. Note that both C_D and C_A depend on the number of grid points N. It is our experience that the first term of (100) dominates the second term. In order to halve the error $E(T)$, δ must decrease by a factor $1/\sqrt{8}$, which increases the δ-dependent cost of computing the solution by a factor $\sqrt{2}$. These estimates will be verified by the numerical examples in Section 5.

4. DISCRETIZING THE SYSTEMS

4.1. *Discretization in Space*

The details of the discretization are essential for the temporal stability of the spatially discrete problem and will therefore be described at some length. For the full system, which will be used to validate the approximate equations, we will apply the discretization suggested by [5]. That scheme was rigorously proven to be stable and convergent when it is applied to the equations governing inviscid water waves with or without surface tension. For the present problem, the stability and convergence will be investigated numerically in Section 5.

We define a grid on the boundary by $\alpha_j = (j-1)h, j = 1, 2, ..., N, h = 2\pi/N$, and let $f_j = f(\alpha_j)$ denote a grid function. Assume N to be even and define the discrete Fourier transform of f by

$$f(\alpha_j) = \hat{f}_0 + \sum_{\omega=1}^{N/2-1} \hat{f}_s(\omega)\sin(\omega\alpha_j) + \sum_{\omega=1}^{N/2} \hat{f}_c(\omega)\cos(\omega\alpha_j), \qquad (101)$$

where

$$\hat{f}_0(\omega) = \frac{1}{N}\sum_{j=1}^{N} f(\alpha_j), \qquad (102)$$

$$\hat{f}_s(\omega) = -\frac{2}{N}\sum_{j=1}^{N} f(\alpha_j)\sin(\omega\alpha_j), \qquad (103)$$

$$\hat{f}_c(\omega) = \frac{2}{N}\sum_{j=1}^{N} f(\alpha_j)\cos(\omega\alpha_j). \qquad (104)$$

Also define the operator D_h to be the filtered spectral approximation of $\partial/\partial\alpha$:

$$D_h f_j = \sum_{\omega=1}^{N/2-1} \omega p(\omega h) \hat{f}_s(\omega)\cos(\omega\alpha_j) - \sum_{\omega=1}^{N/2-1} \omega p(\omega h) \hat{f}_c(\omega)\sin(\omega\alpha_j). \qquad (105)$$

The requirements on the filter function p will be described below.

The discrete version of the full system can be written

$$\frac{d\phi_j}{dt} = \frac{1}{2}(u_j^2 + v_j^2) - \kappa_j, \qquad (106)$$

$$\frac{dX_j}{dt} = u_j, \qquad (107)$$

$$\frac{dY_j}{dt} = v_j, \qquad (108)$$

for $1 \leq j \leq N$. The discretization suggested by [5] is

$$\kappa_j = \frac{(D_h X_j^q)(D_h^2 Y_j) - (D_h Y_j^q)(D_h^2 X_j)}{((D_h X_j^q)^2 + (D_h Y_j^q)^{2)^{3/2}}}, \qquad (109)$$

$$u_j - iv_j = \frac{\gamma_j}{2D_h Z_j} + \frac{h}{\pi i} \sum_{\substack{k=1\\(j-k)\text{odd}}}^{N} \frac{\gamma_k}{Z_k^p - Z_j^p}, \tag{110}$$

where $Z_k = X_k + iY_k$. The integral equation for the vortex sheet strength (9) is discretized by the alternating point trapezoidal rule:

$$D_h\phi_j = \frac{\gamma_j}{2} + \mathrm{Re}\left\{\frac{hD_hZ_j}{\pi i} \sum_{\substack{k=1\\(j-k)\text{odd}}}^{N} \frac{\gamma_k}{Z_k^p - Z_j^p}\right\}. \tag{111}$$

Each variable with a superscript p in (110) and (111) is filtered by the Fourier filter

$$\begin{aligned} X_j^p = p(0)\hat{X}_0 &+ \sum_{\omega=1}^{N/2-1} p(\omega h)\hat{X}_s(\omega)\sin(\omega\alpha_j) \\ &+ \sum_{\omega=1}^{N/2} p(\omega h)\hat{X}_c(\omega)\cos(\omega\alpha_j) \end{aligned} \tag{112}$$

and the superscript q in (109) indicates that the variable is Fourier-filtered by a corresponding expression with the function $p(x)$ replaced by $q(x) = d/dx(xp(x))$. The theory in [5] requires the filter function p to have at least two continuous derivatives and to satisfy $p(x) \geq 0$ for $0 \leq x \leq \pi$, $p(0) = 1$, $p(\pi) = 0$, and, due to surface tension, $p'(\pi) = 0$. Furthermore, the discretization will have spectral accuracy if $p(x) \equiv 1$ for $0 \leq x \leq \lambda\pi$, $0 < \lambda < 1$. In the present work, we will use the C^∞-smooth filter:

$$p(x) = \begin{cases} 1, & 0 \leq x \leq \dfrac{2\pi}{3}, \\ 1 - \left(1 + \exp\left(\dfrac{0.75}{x-\pi} + \dfrac{1.8}{x - 2\pi/3}\right)\right)^{-1}, & \dfrac{2\pi}{3} < x < \pi, \\ 0, & x = \pi. \end{cases} \tag{113}$$

This filter function is similar to $\tilde{p}(x) = \exp(-10(x/\pi)^{25})$, which was used in [5]. However, we found by numerical experiments that the high frequency components of the solution became slightly smaller when (113) was used rather than $\tilde{p}(x)$. One theoretical reason for this might be that $\tilde{p}(\pi) \approx 4.5 \times 10^{-5}$ and $\tilde{p}'(\pi) \approx 3.6 \times 10^{-3}$ are nonzero.

Note that the dependent variables $(\phi_j, X_j, Y_j)^{\mathrm{T}}$, $1 \leq j \leq N$, are not filtered explicitly in this method. The filter is only applied when the right-hand side of (106)–(108) is calculated, i.e., during the evaluation of the curvature, the solution of the integral equation, and the computation of the boundary velocity.

We remark that the merits of the alternating point discretization in (110) was also demonstrated by [4], who performed a stability analysis of different spatial discretizations of a vortex sheet, subject to surface tension close to equilibrium. We refer to [1, 16] for a motivation of the quadrature rule (111) for the integral equation (9). Furthermore, it should be noted that filtered pseudo-spectral discretizations have previously been used and analyzed in the context of hyperbolic systems [12, 17]. An alternative approach to the filtering used in the present paper is described in [11]. In that method an unfiltered spectral derivative ($p(x) \equiv 1$) is successfully used, together with an explicit Fourier filtering of the dependent variables after every time-step. That technique was also attempted for the present problem, but it did not work. One reason might be that the discretization of the curvature induces aliasing instabilities in the present formulation, since the curvature is nonlinear in terms of X and Y. In the formulation used by [11], the curvature is expressed as the arc-length derivative of the tangent angle which can be discretized without aliasing errors.

The dense nonsymmetric matrix in (111) depends on the shape of the domain and will hence change with time. It is therefore not economical to LU-decompose the matrix every time (111) needs to be solved. Instead, we solve the integral equation by the iterative method GMRES [15]. By using the solution from the previous time level as an initial guess, the iteration converges to roundoff level in a few iterations. We have used direct summation to evaluate the matrix–vector products in the GMRES iterations and for computing the velocity. The operational count for these operations is of the order $\mathcal{O}(N^2)$. We note that for large N, the cost of performing these tasks could be reduced to $\mathcal{O}(CN)$, where C is a large constant, by using the fast multipole method [9].

The approximate system (16)–(18) is discretized by

$$\frac{d\tilde{\phi}_j}{dt} = \frac{1}{2}(\tilde{u}_j^2 + \tilde{v}_j^2) - \kappa_j + (t - t_0)G_{1j}^p, \tag{114}$$

$$\frac{d\tilde{X}_j}{dt} = \tilde{u}_j + (t - t_0)G_{2j}^p, \tag{115}$$

$$\frac{d\tilde{Y}_j}{dt} = \tilde{v}_j + (t - t_0)G_{3j}^p, \tag{116}$$

As for the full system, (109) is used to discretize the curvature. Furthermore, we discretize the approximate velocity (66) according to

$$\begin{aligned}\tilde{\overline{u}}_j = \overline{T}_j\tilde{\sigma}_jD_h\tilde{\phi}_j + \overline{N}_j(\overline{u}_j^{(0)} \cdot \overline{N}_j + \tilde{\sigma}_jH \\ [D_h\phi_j' - u_j^{(0)}D_hX_j' - v_j^{(0)}D_hY_j']),\end{aligned}$$

where $\phi_j' = \tilde{\phi}_j - \phi_j^{(0)}$, etc. The discrete unit tangent and normal vectors are defined by $\overline{T}_j = \tilde{\sigma}_j(D_h\tilde{X}_j, D_h\tilde{Y}_j)^{\mathrm{T}}$ and $\overline{N}_j = \tilde{\sigma}_j(-D_h\tilde{Y}_j, D_h\tilde{X}_j)^{\mathrm{T}}$, respectively, with

$$\tilde{\sigma}_j = 1/\sqrt{(D_h\tilde{X}_j)^2 + (D_h\tilde{Y}_j)^2}.$$

Since the operator H has Fourier symbol $i\ \mathrm{sgn}(\omega)$, it satisfies

$$H[f](\alpha_j) = \sum_{\omega=1}^{N/2=1} \hat{f}_s(\omega)\cos(\omega\alpha_j) - \sum_{\omega=1}^{N/2-1} \hat{f}_c(\omega)\sin(\omega\alpha_j). \tag{117}$$

The discrete forcing $\overline{G}_j$ is computed by a formula corresponding to (88)–(90) with $\varepsilon = 10^{-4}$ in the numerical differentiation of $\overline{u}$ and $\tilde{\overline{u}}$. As is indicated in (114)–(116), we filter the forcing by (112). The filtering was found to be necessary to achieve a stable and convergent solution.

4.2. *Time-Integration*

Both the discretized full and approximate problems can be written as large nonlinear systems of ordinary differential equations,

$$\overline{w}_t = A(\overline{w})\overline{w} + \overline{b}(t),$$

where $\overline{w}$ and $\overline{b}$ are $3N$-vectors and $A(\overline{w})$ is a $3N \times 3N$-matrix. The forcing $\overline{b}$ is only present for the approximate system. When a system of ordinary differential equations is integrated in time with an explicit time-integration method, the time-step Δt must satisfy a stability constraint of the type

$$\Delta t\lambda_k \in \mathscr{S} \subset \mathbf{C} \tag{118}$$

for all eigenvalues λ_k of $A(\overline{w})$, where $\mathscr{S}$ is the stability region in the complex plane, cf. [8]. Figure 1 shows the stability region for some popular time-integration methods when A is independent of $\overline{w}$. The stability region is generally smaller for nonlinear systems.

To estimate the eigenvalues λ_k of the operator A, we linearized the discretized full system around a circular cross-section and a constant velocity potential. A straightforward constant coefficient analysis similar to that for the continuous problem in Section 2 showed that

$$\lambda_k^{(1)} = 0, \quad \lambda_k^{(2,3)} = \pm i(\sigma^{(0)}p(h|k|)|k|)^{3/2}, \tag{119}$$
$$-N/2 + 1 \le k \le N/2,$$

if lower order terms in k are neglected.

There are many time-integration methods that could be used for the present problem. Here we will limit the discussion to the methods shown in Fig. 1. To avoid unphysical growth of the eigenmodes which correspond to purely imaginary eigenvalues, it is important that the time-integrator is stable for such eigenvalues. The stability region includes an interval of the imaginary axis for the second-order Adams predictor–corrector method, Hyman's third-order scheme and the classical fourth-order Runge–Kutta method. However, neither the second-order Adams–Bashforth method, nor the second-order Runge–Kutta scheme are stable for purely imaginary eigenvalues; cf. [10].

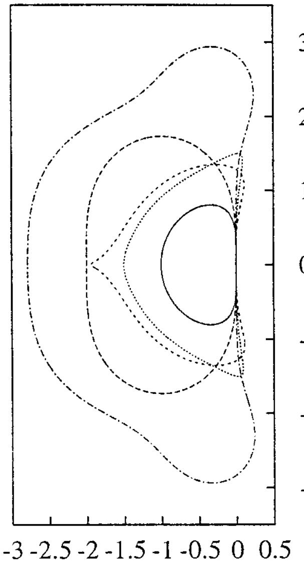

FIG. 1. The neutral stability curve in the complex plane for some explicit time-integration methods: second-order Adams–Bashforth (solid); second-order Runge–Kutta (long-dashed); second-order Adams predictor–corrector (short-dashed); Hyman's third-order method (dotted); and fourth-order Runge–Kutta (dot-dashed). The stability region $\mathscr{S}$ for each method is the bounded domain inside of the corresponding neutral stability curve.

We demonstrate the importance of the stability region by integrating the discrete full system (106)–(108) numerically with the initial data

$$\begin{aligned} \phi_0(\alpha) &= 0, \\ X_0(\alpha) &= 1.25\cos(\alpha), \\ Y_0(\alpha) &= 0.8\sin(\alpha). \end{aligned} \tag{120}$$

The solution was integrated up to time $t = 0.085$ with three of the time-integrators described above: fourth-order Runge-Kutta, second-order Adams–Bashforth, and second-order Runge–Kutta. The resolution in the calculations was $N = 256$. For all time-integrators, we used a time-step restriction of the type $\Delta t = C_{\Delta t}/\max|\lambda_k|$, where $|\lambda_k|$ was estimated with (119). For the fourth-order Runge–Kutta scheme we used $C_{\Delta t} = 2.5$, which is inside of the stability region. Because both the second-order Runge–Kutta

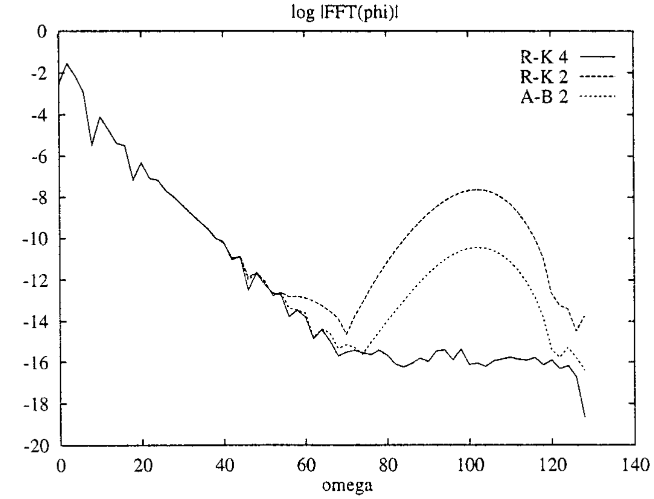

FIG. 2. The magnitude of the Fourier coefficients $|\hat{\phi}(\omega, t)|$ at time $t = 0.085$ calculated by solving the discretized full system starting from the initial data (120). The solid line was computed with the fourth-order Runge–Kutta scheme, the dashed line with the second-order Runge–Kutta method, and the dotted line with the second-order Adams–Bashforth scheme.

scheme and the second-order Adams–Bashforth method are unstable, according to the above theory, it is hard to specify a meaningful time-step restriction. Here we used a stability limit that corresponds to the same number of function evaluations per unit time as the fourth-order Runge–Kutta scheme. This leads to $C_{\Delta t} = 0.625$ for the second-order Adams–Bashforth method and $C_{\Delta t} = 1.25$ for the second-order Runge–Kutta scheme.

To study the stability, it is instructive to monitor the time-evolution of the magnitude of the Fourier coefficients $|\hat{\phi}(\omega, t)| =: \max(|\hat{\phi}_s(\omega, t)|, |\hat{\phi}_c(\omega, t)|)$. Because of symmetries in the initial data, $|\hat{\phi}(\omega, t)| = 0$, for $t \geq 0$, $\omega = 1, 3, 5, \ldots$. Therefore, only even frequencies will be presented in the graphs of $\hat{\phi}$. The results are shown in Fig. 2. Clearly, the fourth-order Runge–Kutta scheme is stable, but both the second-order Adams–Bashforth and the second-order Runge–Kutta schemes are unstable and we can see that the high frequency Fourier coefficients have grown substantially already at $t = 0.085$. The fourth-order Runge–Kutta calculation was continued up to time $t = 3.0$ and showed no sign of instability. However, the second-order Runge–Kutta computation blew up at time $t \approx 0.14$ and the second-order Adams–Bashforth exploded at $t \approx 0.22$. We observed that the growth could be reduced, but not eliminated, by taking a smaller time-step. Since a smaller time-step makes the methods less efficient than the fourth-order Runge–Kutta scheme, they will not be considered further.

We proceed by comparing the efficiency of the stable time-integration methods shown in Fig. 1. Let $C_{\Delta t}$ be the intersection of the positive imaginary axis and the stability region $\mathscr{S}$; see Fig. 1. The largest stable time-step is proportional to $C_{\Delta t}$, but the efficiency of the different methods also depends on the number of function evaluations necessary to complete one time-step. The second-order Adams predictor–corrector needs 2, Hyman's third-order method uses 2, and the fourth-order Runge–Kutta requires 4 function evaluations. By comparing the ratio between $C_{\Delta t}$ and the number of function evaluations we find that the methods are comparable in terms of efficiency. We will use the fourth-order Runge–Kutta method in the present work. We refer to [10] for a more thorough analysis of different time-integration methods.

A similar analysis for the discretized approximate system shows also that these equations must be integrated in time with a method that is stable for purely imaginary eigenvalues. For the same reasons as above, the fourth-order Runge–Kutta scheme will be used also for that problem.

5. NUMERICAL EXPERIMENTS

To study the evolution of a slender three-dimensional jet, we consider the family of initial data given by

$$\phi_0(\alpha) = 0, \tag{121}$$

$$X_0(\alpha) = a\cos\alpha, \tag{122}$$

$$Y_0(\alpha) = \frac{1}{a}\sin\alpha. \tag{123}$$

Hence, the initial cross-sectional velocity is zero and the initial cross section is elliptical with area π and aspect ratio $AR = a^2$. If $a = 1$, the solution of the full system is trivially given by $\phi(\alpha, t) = -t$, $X(\alpha, t) = \cos\alpha$, and $Y(\alpha, t) = \sin\alpha$. The symmetry between the x and y directions makes it sufficient to study $a > 1$.

Unless otherwise mentioned, the computations presented below were performed in 64-bit precision on a DEC-α machine with 134.8 SPECfp92 and 64 Mb RAM. The Fourier transforms were computed by the FFT-package for real-valued functions in the SLATEC library and the time-step in the explicit time-integration was taken to be $\Delta t = 2.5/\max_k |\lambda_k|$, with λ_k estimated by (119).

5.1. *The Full System*

To make sure that the discretization scheme for the full system is stable and convergent we studied the Fourier coefficients $|\hat{\phi}(\omega, t)| =: \max(|\hat{\phi}_s(\omega, t), |\hat{\phi}_c(\omega, t|)$ at fixed time levels for different AR and different resolutions. A representative example is given in Fig. 3, where the case $AR = 2.0$ at time $t = 3.0$ is shown. Note that roundoff errors prevent the Fourier modes from decaying below approximately 10^{-16}. It can be seen that the highest modes

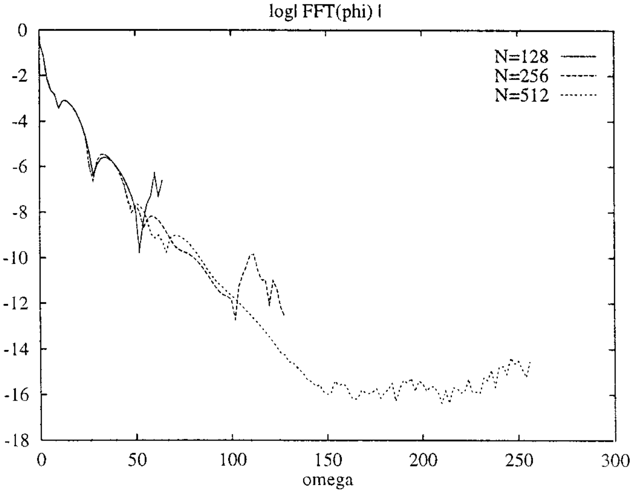

FIG. 3. The spectrum of ϕ as function of ω at time $t = 3.0$ computed by solving the discretized full system for the case $AR = 2.0$.

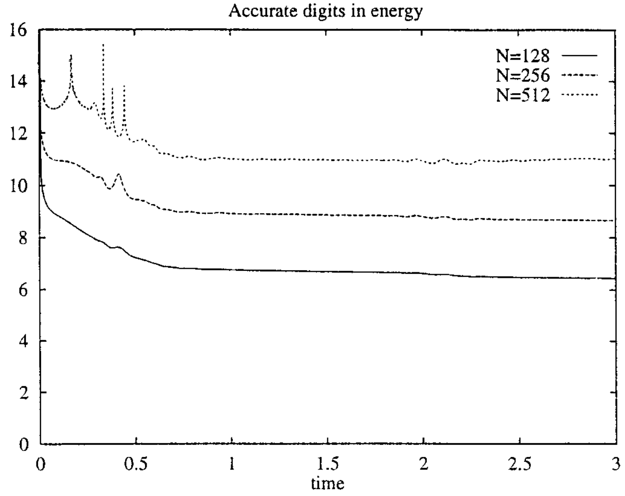

FIG. 4. The number of accurate digits in the perturbation energy $E(t)$ in the solution of the discretized full system as function of time for the case $AR = 2.0$.

remain small for long times and that the lower half of the modes converge well as the resolution is increased. This is numerical evidence for the stability and convergence of the discretized full system.

To check the accuracy of the numerical solution, we will monitor the perturbation energy $E(t)$, which is a conserved quantity not conserved by the discretization [11, 14]. The energy consists of two components,

$$E(t) = K(t) + S(t), \tag{124}$$

where $K(t)$ is the perturbed kinetic energy,

$$K(t) = \frac{1}{2}\int_{\Omega} \sqrt{u^2 + v^2}\, d\Omega = \frac{1}{2}\int_0^{2\pi} (Y_\alpha u - X_\alpha v)\, \phi\, d\alpha,$$

and $S(t)$ is the interfacial energy,

$$S(t) = \int_0^{2\pi} \sqrt{X_\alpha^2 + Y_\alpha^2}\, d\alpha - 2\pi.$$

The integrals are evaluated numerically by the trapezoidal rule, which is spectrally accurate since the integrands are periodic in α.

In Fig. 4, we present $-\log_{10} |E(t) - E(0)|$ as function of t, which is an estimate of the number of accurate digits in the perturbation energy. $E(t) - E(0)$ is also a measure of the error in the numerical solution, and it can be seen from Fig. 4 that it is approximately proportional to 10^{-CN}, for some constant $C > 0$. This indicates that the numerical solution is spectrally accurate.

The shape of the free surface as function of time for the case with $AR = 3.0$ and the resolution $N = 256$ is presented in Fig. 5. This simulation of the full system required 5832 s of CPU time to reach time $t = 5.0$. From this example it is clear that simulating only the first one or two oscillations of a jet with high initial aspect ratio, which calls for a large N to resolve the solution, requires a substantial amount of CPU time. In these cases we are interested in speeding up the computation by replacing the full system by the approximate system. One oscillation of the jet yields important information for engineering applications [7], and we will henceforth restrict our study to the time interval

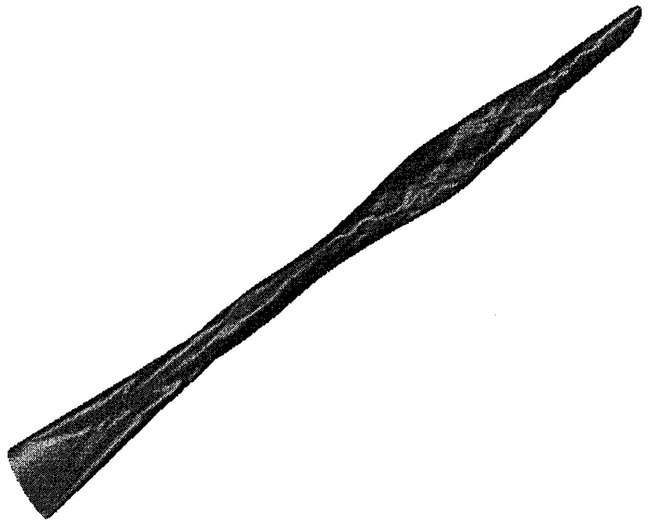

FIG. 5. A rendered image of the free surface as function of time. The initial cross section with $AR = 3.0$ is in the lower left corner. Time increases diagonally upwards and reaches $t = 5.0$ in the upper right corner. Note the chain-like appearance of the free surface which is observed in slender three-dimensional jets [7].

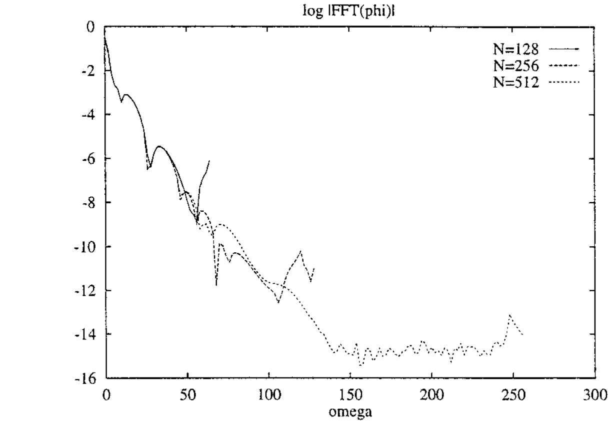

FIG. 6. The spectrum of $\tilde{\phi}$ as function of ω at time $t = 3.0$ computed by solving the discretized approximate system with $\Delta T = 0.01$ for the case $AR = 2.0$.

$0 \leq t \leq 3$, which approximately corresponds to the first oscillation of the cross section.

5.2. *The Approximate System*

Similar to the full system, we studied the stability and convergence of the discretized approximate system by computing the spectrum of $\hat{\phi}$ for different initial conditions and resolutions. In addition, the time interval ΔT between linearizing the velocity was varied. A typical example is shown in Fig. 6, where the case $AR = 2.0$ with $\Delta T = 0.01$ at time $t = 3.0$ is presented. Since the lower half of the Fourier modes converge well, and the highest modes remain small for long times, we conclude that the discretization of the approximate system is stable and convergent.

We proceed by studying how the error in the discretized approximate system depends on the time-interval between linearizing the velocity, ΔT. In Table I, we present the relative error at time $t = 3.0$ for the case $AR = 3.0$ with $N = 256$. The solution of the full system with the same resolution was taken as the reference solution. From this example we see that the error is approximately proportional to ΔT^2. Also note that the estimated relative error,

$$E_{\text{est}} = \sum_{k=1}^{N_T} \frac{\Delta T^3 \|\overline{D}(t_k)\|_2}{3\|\tilde{\Phi}(t_k)\|_2}, \tag{125}$$

consistently overpredicts the actual relative error by a factor ≈ 3. To indicate how much CPU time is saved by using the approximate system, we also present the average number of explicit time-steps per ΔT, defined by

$$\frac{\Delta T}{\Delta t} = \frac{1}{N_T} \sum_{k=1}^{N_T} \frac{\Delta T}{\Delta t(t_k)}.$$

Twice the $\Delta T/\Delta t$ ratio gives a rather good estimate of how much faster it is to integrate the approximate system compared to the full system, because two Dirichlet problems are solved every time the velocity term is linearized, and four Dirichlet problem are solved per explicit time-step when the full system is integrated. Note that the efficiency of the full system would have been similar if, for instance, the second-order Adams predictor–corrector method had been used as time-integrator. In that case, only two Dirichlet problems need to be solved per explicit time step, but the largest stable time-step is less than half of that of the fourth-order Runge–Kutta method.

TABLE I

Results from Integrating the Approximate System with a Fixed Time Interval ΔT between Linearizing the Velocity.

ΔT	Est. error	Rel. error	CPU (s)	N_T	$\Delta T/\Delta T$
1.0×10^{-2}	4.64×10^{-3}	1.35×10^{-3}	336	300	6.25
5.0×10^{-3}	1.15×10^{-3}	3.24×10^{-4}	638	600	3.13
2.5×10^{-3}	2.86×10^{-4}	8.02×10^{-5}	1218	1200	1.56

Note. The error is approximately proportional to ΔT^2. In this example, $AR = 3.0$, $N = 256$, and the solutions were computed at time $t = 3.0$. The full system required 3308 s of CPU time.

According to (95)–(97), the error in the approximation satisfies $\overline{D}\Delta T^3/3 + \mathcal{O}(\Delta T^4)$. To show that $\overline{D}$ is essentially independent of ΔT we present in Fig. 7 the L_2-norm of $\overline{D}$ as a function of time for $\Delta T = 0.01$ and $\Delta T = 0.0025$. Again, the initial data had $AR = 3.0$ and the resolution was $N = 256$. Because the approximation is based on simplifying the normal component of the velocity, the error in the approximation can be suspected to depend on the properties of that quantity. To enable a close comparison, we also present the max-norm of the normal velocity in Fig. 7. We conclude that there is a clear correlation in time between the normal velocity and the error term $\overline{D}$. From this conclusion, we are led to investigate how the size of the normal velocity affects the stability and the accuracy of the time integration. In particular, we are interested in how large ΔT can be before the time integration goes unstable and how small ΔT must be to maintain a constant error level when the normal velocity increases. We therefore took initial data with increasing AR, which correspond to an increasing normal velocity and curvature, and increased ΔT until the solution would blow up. The largest ΔT, where the solution did not blow up as a function of the maximum norm of the normal velocity, can be found in Fig. 8. In these computations, AR was in the range 1.5–3.5 and the resolution was in the range $N = 128$–512. We found that $N = 128$ only provided adequate resolution

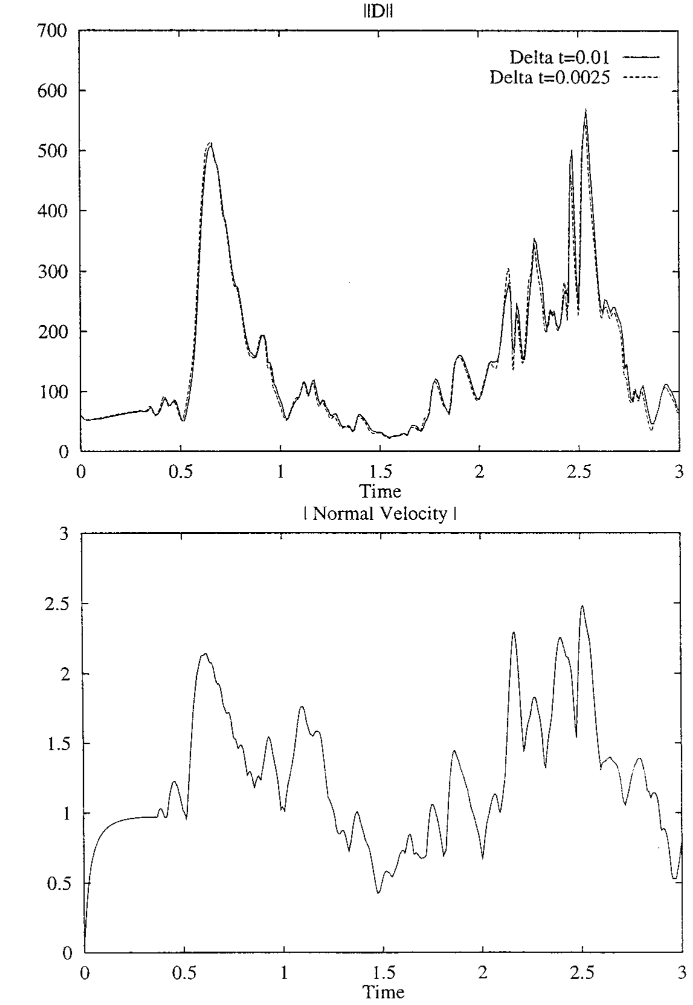

FIG. 7. Top graph: The L_2-norm of the error term $\overline{D}$ as function of time for $\Delta T = 0.01$ (solid) and $\Delta T = 0.0025$ (dotted). Bottom graph: The max-norm of the normal component of the velocity as functions of time. In both graphs the aspect ratio was $AR = 3.0$ and $N = 256$.

for $AR \le 2.5$ and $N = 256$ was only good for $AR \le 3.0$. The resolution $N = 384$ was adequate to resolve all AR and the $N = 512$ resolution was used to verify the $N = 384$ computations. The data points fall on an almost straight line with slope ≈ -2.6 in the log–log graph, so we are led to the estimate

$$\Delta T \le \frac{C}{(|\overline{u} \cdot \overline{N}|_\infty)^\gamma}, \quad \gamma \approx 2.6, \tag{126}$$

where C is independent of N. Therefore, as long as ΔT satisfies (126), it can be chosen to meet accuracy requirements, instead of stability restrictions.

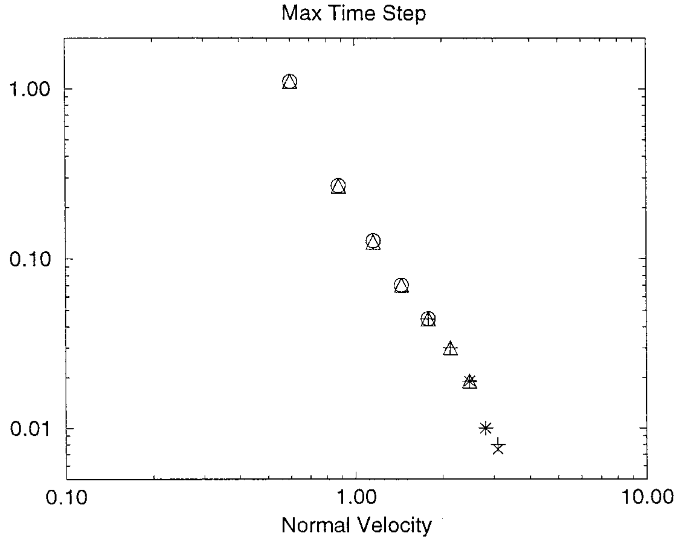

FIG. 8. The largest stable ΔT as function of the maximum norm of the normal velocity on a log–log scale. The initial data had $AR = 1.5 - 3.5$. The resolution was in the range $N = 128 - 512$, where $N = 128$ are circles, $N = 256$ are triangles, $N = 384$ are pluses, and $N = 512$ are denoted by X's. Note that ΔT is essentially independent of N.

To assess the accuracy of the approximate system, we computed the relative error in the solution at time $t = 3.0$ as function of ΔT for initial data with aspect ratio in the range 1.5–3.5. The solution of the full system was taken as reference solution. The results are presented in Fig. 9. The time-step ΔT must be decreased to maintain a fixed error level when the aspect ratio and, therefore, the curvature are increased. This is because the normal velocity increases as the aspect ratio of the initial cross section increases (cf. Table II), and a larger normal velocity leads to a larger error term $\overline{D}$.

We note that $\|\overline{D}\|_2$ varies by an order of magnitude in Fig. 7. From the error estimate (125), we see that a large portion of the error is committed when $\|\overline{D}\|_2$ is large. To better optimize the computational resources, we will employ the adaptive time-step control (98). In Table III, we compare the solution of the approximate system with the solution of the full system at time $t = 3.0$. In this case, $AR = 3.0$ and $N = 256$. In this table, the estimated relative error (125) is compared to the actual relative error, where the solution of the full system with the same resolution was taken as the reference solution. Note that (99) predicts that the relative error should halve when δ is decreased by a factor of $1/\sqrt{8}$. Our numerical results confirm this estimate. Also note that the estimated error is consistently overpredicting the actual error by a factor $\approx$3.

By comparing the number of linearizations, N_T, to the CPU time in Table III, we see that the CPU time/N_T ratio decreases as N_T increases. This indicates that the CPU time is not only spent setting up the approximate system, but also it is used for integrating the approximate system. Therefore, the CPU time/N_T ratio is larger when N_T is

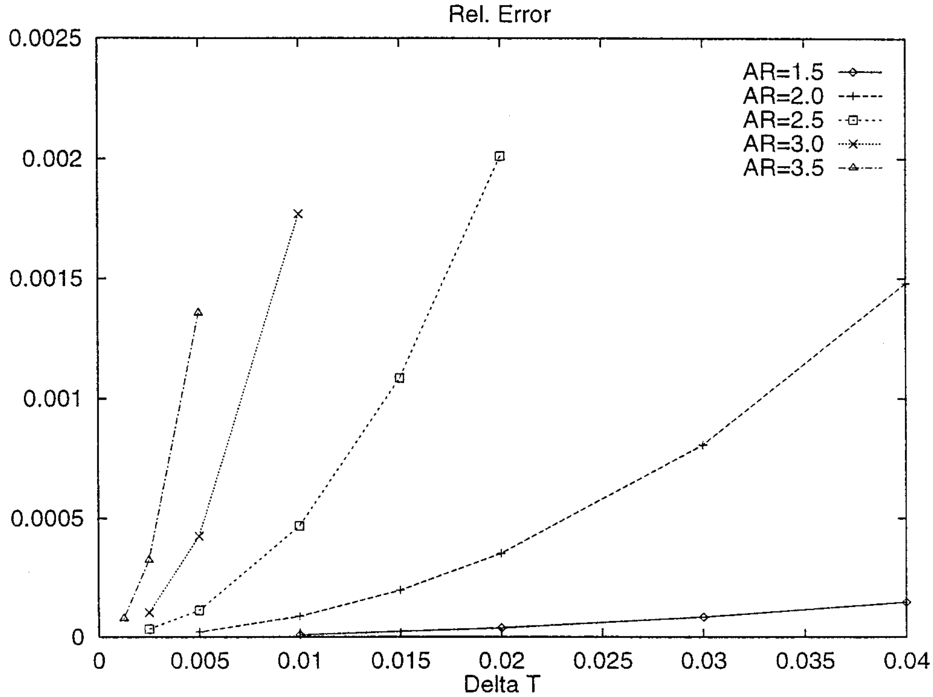

FIG. 9. The relative error as function of ΔT in the solution of the discretized approximate system at time $T = 3.0$. The initial data had aspect ratio in the range 1.5–3.5.

smaller. The estimate (100) predicts that N_T would increase by a factor $\sqrt{2}$ when δ is decreased by a factor $1/\sqrt{8}$. In the present example, N_T increases slightly slower than that.

To study how the resolution affects the solution of the approximate system, we doubled the number of grid points to $N = 512$. The result is presented in Table IV. In this case the reference solution was taken to be the solution of the full system with $N = 512$. It can be seen that the relative error is essentially the same as for $N = 256$. This is explained by the fact that the relative difference between the solutions of the full systems with $N = 256$ and $N = 512$, respectively, is only 5.87×10^{-5}. The relative errors reported in Table III and IV are therefore dominated by errors committed by simplifying the velocity in the approximate system.

The estimate (100) indicates that the cost of solving the approximate system would increase by a factor between 4 and $4\sqrt{2}$ when N is doubled, because C_D increases by a factor 4 and C_A increases by a factor $4\sqrt{2}$, since the number of time-steps for integrating the approximate system increases by a factor $2\sqrt{2}$ and the effort in evaluating the right-hand side of the system doubles. This estimate is verified by comparing the CPU-timings of Tables III and IV. This should be compared to the cost of solving the full system, which increases by a factor $8\sqrt{2}$ when N is doubled,

TABLE II

The Relation between the Aspect Ratio of the Initial Data, the Maximum Norm of the Normal Velocity, and the Maximum Norm of the Curvature.

AR	$\lvert\overline{u}\cdot\overline{N}\rvert_\infty$	$\lvert\kappa\rvert_\infty$
1.5	0.59	1.84
2.0	1.16	2.83
2.5	1.82	3.95
3.0	2.63	5.20
3.5	3.12	7.48

Note. The maximum norm refers to the largest value during the time interval $0 \leq t \leq 3.0$.

TABLE III

Results from Integrating the Approximate System with a Fixed δ.

δ	Est. error	Rel. error	CPU (s)	N_T	$\Delta T/\Delta T$
1.25×10^{-4}	1.90×10^{-2}	8.72×10^{-3}	193	148	12.8
4.42×10^{-5}	8.94×10^{-3}	3.32×10^{-3}	242	200	9.43
1.56×10^{-5}	4.38×10^{-3}	1.56×10^{-3}	322	280	6.73
5.52×10^{-6}	2.18×10^{-3}	7.53×10^{-4}	435	394	4.78
1.95×10^{-6}	1.08×10^{-3}	3.67×10^{-4}	592	556	3.39

Note. In this case, $AR = 3.0$, $N = 256$, and the solutions were compared at time $t = 3.0$. The full system required 3308 s of CPU time.

TABLE IV

Results from Integrating the Approximate System with a Fixed δ.

δ	Est. error	Rel. error	CPU (s)	N_T	$\Delta T/\Delta T$
1.25×10^{-4}	1.90×10^{-2}	7.72×10^{-3}	815	148	36.1
4.42×10^{-5}	8.96×10^{-3}	3.13×10^{-3}	1017	200	26.6
1.56×10^{-5}	4.39×10^{-3}	1.55×10^{-3}	1290	280	19.0
5.52×10^{-6}	2.18×10^{-3}	7.53×10^{-4}	1729	394	13.5
1.95×10^{-6}	1.08×10^{-3}	3.67×10^{-4}	2280	556	9.57

Note. In this case, $R = 3.0$, $N = 512$, and the solutions were compared at time $t = 3.0$. The full system required 30,997 s of CPU time.

because the cost of solving a Dirichlet problem increases by a factor 4 and the number of time-steps increases by a factor $2\sqrt{2}$. The CPU timings for the full system also confirm this estimate. For this reason the relative saving in CPU time by using the approximate system increases when the resolution increases. For example, at this resolution, we only have to spend 3.3% of the CPU time required by the full system to achieve a solution with the relative error 0.33%.

We conclude by using the approximate system to simulate the complicated dynamics emanating from initial data with $AR = 5.0$. In Fig. 10, we show the time-evolution of the cross-sections of the jet. This simulation was done with the resolution $N = 1536$ and $\Delta T = 1.0 \times 10^{-3}$. It required 7322 s of CPU time on one processor on a CRAY YMP. The average $\Delta T/\Delta T$ ratio was 7.68, whichi indicates that it would have taken approximately 15 times longer to compute this solution if the full system had been used. The error estimate (125) predicted that the relative error at time $t = 1.25$ was 9.0×10^{-4}.

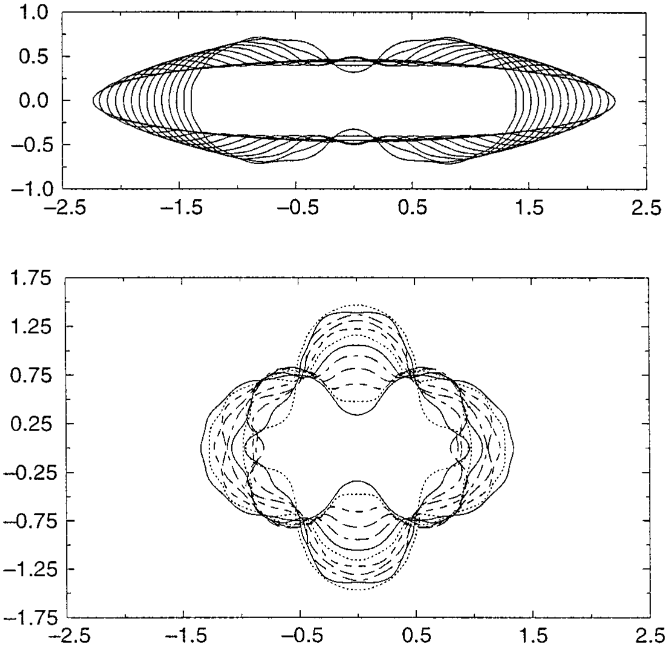

FIG. 10. Time evolution of the cross section of the jet with initial aspect ratio $AR = 5.0$. Top graph: $t = 0.00$ to $t = 0.65$ with spacing 0.05. Bottom graph: $t = 0.70$ to $t = 1.25$ with spacing 0.05. Here, $t = 0.70$ is solid, $t = 0.75$ is dotted, $t = 0.80$ is dashed, $t = 0.85$ is long-dashed, $t = 0.90$ is dot-dashed, $t = 0.95$ is solid, $t = 1.00$ is dotted, $t = 1.05$ is dashed, $t = 1.10$ is long-dashed, $t = 1.15$ is dot-dashed, $t = 1.20$ is solid, and $t = 1.25$ is dotted.

6. CONCLUSIONS

We have presented a system of partial differential equations that approximate the governing equations for inviscid free surface flows subject to surface tension. The approximation is based on repeated linearization of the normal component of the boundary velocity, together with a small scale approximation of the perturbation of the velocity. Two Dirichlet problems must be solved to form the approximate system, after which it can be evolved without solving Dirichlet problems. The accuracy of the solution of the approximate system is determined by the magnitude of the normal velocity and by how often the velocity term is linearized. This time interval is denoted ΔT. We have shown that the error in the solution of the approximate system at a fixed time T is of the order $\mathcal{O}(\Delta T^2)$. We have exemplified the use of the approximate system by integrating the equations governing a slender nonaxisymmetric three-dimensional jet subject to surface tension, where the evolution of the cross section of the jet is governed by the two-dimensional Euler equations inside of the free surface. It has been demonstrated numerically that the error increases with increasing normal velocity and curvature and that there is a stability limit of the form $\Delta T \leq C/(|\overline{u} \cdot \overline{N}|_\infty)^\gamma$, where $\overline{u} \cdot \overline{N}$ denotes the normal velocity and $\gamma \approx 2.6$. Importantly, C is independent of the resolution. Hence, the time-step ΔT can be chosen independently of the number of grid points, N, as long as it satisfies the stability constraint governed by the normal velocity. This is in contrast to the time-step when the original system is integrated, where the stability limit is $\Delta t \leq \mathcal{O}(N^{-3/2})$ and a constant number of Dirichlet problems have to be solved per Δt. Therefore, the relative saving of CPU time by using the approximate system increases when the resolution increases.

ACKNOWLEDGMENT

This work was supported by the U.S. Department of Energy through Los Alamos National Laboratory under Contract W-7405-ENG-36.

REFERENCES

1. C. T. H. Baker, *The Numerical Treatment of Integral Equations* (Clarendon Press, Oxford, 1977).

2. G. R. Baker, ''Generalized Vortex Methods for Free-Surface Flows,'' in *Waves on Fluid Interfaces*, edited by R. Meyer (University of Wisconsin Press, Madison, 1983).
3. G. R. Baker and D. W. Moore, *Phys. Fluids A* **1,** 1451 (1989).
4. G. R. Baker and A. Nachbin, Technical Report 92-23, Dept. of Mathematics, Ohio State University, 1992 (unpublished).
5. J. T. Beale, T. Y. Hou, and J. S. Lowengrub, *SIAM J. Numer. Anal.* **33**(5), (1996).
6. J. T. Beale, T. Y. Hou, and J. S. Lowengrub, *Commun. Pure Appl. Math.* **46,** 1269 (1993).
7. S. E. Bechtel, *J. Appl. Mech.* **56,** 968 (1989).
8. G. Dahlquist and Å. Bjӧrk, *Nunerical Methods* (Prentice–Hall, Princeton, NJ, 1974).
9. L. Greengard and V. Roklin, *J. Comput. Phys.* **105,** 267 (1987).
10. B. Gustafsson, H.-O. Kreiss, and J. Oliger, *Time Dependent Problems and Difference Methods* (Wiley-Interscience, New York, 1995).
11. T. Y. Hou, J. Lowengrub, and M. J. Shelley, *J. Comput. Phys.* **114,** 312 (1994).
12. H.-O. Kreiss and J. Oliger, *SIAM J. Numer. Anal.* **16**(3), 421 (1979).
13. N. A. Petersson and H.-O. Kreiss, LA-UR 94-2978, Los Alamos National Laboratory, NM, 1994 (unpublished).
14. D. I. Pullin, *J. Fluid Mech.* **119,** 507 (1982).
15. Y. Saad and M. H. Schultz, *SIAM J. Sci. Stat. Comput.* **7,** 856 (1986).
16. A. Sidi and M. Israeli, *J. Sci. Comput.* **3,** 323 (1988).
17. E. Tadmor, *SIAM Rev.* **29**(4), 525 (1987).