INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 14, 495–501 (1992)

SHORT COMMUNICATION

AN IMPROVED FINITE DIFFERENCE SCHEME FOR A NEWTONIAN JET SWELL PROBLEM

TSAI-AN YU AND TA-JO LIU*

Department of Chemical Engineering, National Tsing Hua University, Hsinchu, Taiwan 30043, R.O.C.

SUMMARY

The finite difference scheme developed by Liu *et al.* for the Newtonian jet swell problem has been improved: an algebraic approach has been adopted for the numerical mapping; a new formulation for free surface iteration has been proposed; the discrete flow equations have been solved by a combination of the successive line underrelaxation method and the Picard method. With these modifications we are capable of achieving more accurate numerical solutions and a substantial saving in computing time.

KEY WORDS Jet swell Finite difference method

INTRODUCTION

Recently Liu *et al.*[1] solved the Newtonian jet swell problem with a numerical mapping technique called the boundary-fitted co-ordinate transformation method (BFCTM).[2,3] In this paper we shall discuss an improved finite difference technique for the method previously developed by Liu *et al.* Instead of solving the partial differential equations for mapping, we have taken an algebraic approach to complete the mapping procedure. We have also modified the mathematical formulation for locating the free surface. After the flow equations have been discretized by standard central difference formulae, the resulting finite difference equations can be solved iteratively by a combination of the successive line underrelaxation method and the Picard method. With these modifications we can obtain more accurate numerical solutions and a substantial amount of computing time can be saved.

The flow geometry of the Newtonian jet is shown in Figure 1(a). The mathematical formulation of the jet swell problem is the same as in the work of Liu *et al.*, except that we reformulate the tangential stress boundary condition (TSBC) on the free surface as

$$\left[1-\left(\frac{\mathrm{d}H}{\mathrm{d}x}\right)^2\right]\omega=-4\frac{\mathrm{d}H}{\mathrm{d}x}\frac{\partial^2\varphi}{\partial x\partial y}-2\left[1-\left(\frac{\mathrm{d}H}{\mathrm{d}x}\right)^2\right]\frac{\partial^2\varphi}{\partial x^2}. \tag{1}$$

This equation is used to compute the vorticity on the free surface.

* Author to whom correspondence should be addressed.

0271–2091/92/040495–07$05.00

Received May 1991

CO-ORDINATE TRANSFORMATION

We use the BFCTM to transform the flow geometry in Figure 1(a) onto the regular domain in Figure 1(b). Liu *et al.*[1] solved the mapping equations numerically, but since the flow geometry is not complex, we propose to use an algebraic approach for mapping here. The mapping equations can be rearranged as

$$P(\xi,\eta)=\left(\frac{\partial x}{\partial \eta}f_{\mathrm{B}}-\frac{\partial y}{\partial \eta}f_{\mathrm{A}}\right)\Big/J^3, \tag{2}$$

$$Q(\xi,\eta)=\left(\frac{\partial y}{\partial \xi}f_{\mathrm{A}}-\frac{\partial x}{\partial \xi}f_{\mathrm{B}}\right)\Big/J^3, \tag{3}$$

in which

$$f_{\mathrm{A}}=\alpha\frac{\partial^2 x}{\partial \xi^2}-2\beta\frac{\partial^2 x}{\partial \xi\partial \eta}+\gamma\frac{\partial^2 x}{\partial \eta^2}, \tag{4}$$

$$f_{\mathrm{B}}=\alpha\frac{\partial^2 y}{\partial \xi^2}-2\beta\frac{\partial^2 y}{\partial \xi\partial \eta}+\gamma\frac{\partial^2 y}{\partial \eta^2}, \tag{5}$$

$$\begin{aligned}\alpha&=\left(\frac{\partial x}{\partial \eta}\right)^2+\left(\frac{\partial y}{\partial \eta}\right)^2, &\quad \beta&=\frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta}+\frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta},\\ \gamma&=\left(\frac{\partial x}{\partial \xi}\right)^2+\left(\frac{\partial y}{\partial \xi}\right)^2, &\quad J&=\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta}-\frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta},\end{aligned} \tag{6}$$

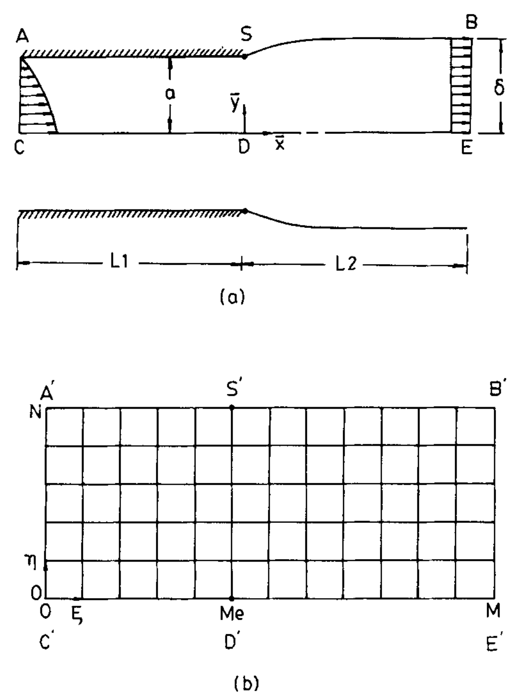

Figure 1. Flow geometry in (a) physical plane and (b) transformed plane

J being the Jacobian of the transformation. $P(\xi, \eta)$ and $Q(\xi, \eta)$ are forcing functions used to regulate the mesh intervals.

After imposing the Dirichlet boundary conditions, we can set up an internal grid distribution in the physical plane by hand, by a digitizer or by a simple algebraic function such as a polynomial, and the correspondence between the (x, y)-plane and the (ξ, η)-plane can be easily found. The first and second derivatives on the right-hand sides of (2) and (3) can also be easily approximated by standard finite difference schemes.[4] By doing so, the forcing functions P and Q which can be assigned arbitrarily now serve as the 'residuals' of the manually set mapping. It is obvious that since the mapping equations need not be solved numerically, a significant amount of computing time can be saved.

The three conditions for the jet free surface should also be transformed and the kinematic condition becomes[1]

$$\frac{\mathrm{d}H}{\mathrm{d}x} = \frac{\partial y/\partial \xi}{\partial x/\partial \xi} = \frac{v}{u}. \tag{7}$$

The tangential stress boundary condition (1) is used to generate the vorticity on the free surface. The terms $\partial^2 \varphi/\partial x \partial y$ and $\partial^2 \varphi/\partial x^2$ in (1) can be expressed as functions of ξ and η.[5] The stress boundary conditions[1] are rearranged and used to update u and v on the jet free surface:

(i) the tangential stress boundary condition (TSBC)

$$\frac{\partial u}{\partial \eta} = \frac{1}{\gamma}\left(\beta + J\frac{\partial y/\partial \xi}{\partial x/\partial \xi}\right)\frac{\partial u}{\partial \xi} - \frac{1}{2}\frac{J}{\gamma \partial x/\partial \xi}\left[\left(\frac{\partial x}{\partial \xi}\right)^2 - \left(\frac{\partial y}{\partial \xi}\right)^2\right]\omega, \tag{8}$$

(ii) the normal stress boundary condition (NSBC)

$$\frac{\partial v}{\partial \eta} = \left(\frac{\beta}{\gamma} + \frac{J}{\gamma}\frac{\partial y/\partial \xi}{\partial x/\partial \xi}\right)\frac{\partial v}{\partial \xi} - \frac{J}{\gamma}\frac{\partial y}{\partial \xi}\omega + \frac{J}{2\partial x/\partial \xi}\left(p + \frac{R}{Ca}\right). \tag{9}$$

NUMERICAL PROCEDURE

It takes three steps to solve the jet swell problem.

(I) Mapping

In the present analysis we do not solve the mapping equations numerically as in the previous work;[1] instead we select an interpolation function to represent the distribution of the internal grid points in the physical plane. It is important to select an interpolation function such that the numerical errors which come from the skewness of grid lines are avoided. We apply an even mesh in the y-direction. For the x-direction we apply an even mesh if $x \leqslant 0$, while a quadratic function is chosen for $x > 0$:

$$x = c_1 + c_2\xi + c_3\xi^2. \tag{10}$$

Three conditions are needed to determine the parameters in (10):

$$\begin{aligned} &\text{(i)}\quad \xi = M_e, \qquad && x = 0 \quad \text{(slot exit)},\\ &\text{(ii)}\quad \xi = M, && x = L_2,\\ &\text{(iii)}\quad \xi = M_e, && \partial x/\partial \xi = f_m, \end{aligned} \tag{11}$$

where f_m is the even mesh size for $x \leqslant 0$. The third condition is used to keep a smooth variation from upstream to downstream. The three parameters c_1, c_2 and c_3 can be found from (11).

After determining the derivatives of x and y by second-order difference approximations,[4] the coefficients α, β, γ and J and the forcing functions P and Q can be estimated by using (2)–(6); these values together with the first derivatives of x and y will be needed later.

(II) Solution of the flow equations

The procedure for solving the flow equations is the same as in the previous study of Liu *et al.*[1] The successive line underrelaxation (SLUR) method was applied to solve the finite difference equations previously; however, as one of the reviewers suggested, the SLUR method can be combined with the Picard method to reduce the computing time. We found this combination could be quite effective if we started the iteration with the SLUR method and switched to the Picard method if the maximum error of vorticity was smaller than a preset tolerance.

The vorticities on the wall and along the jet free surface are unknown and need to be determined iteratively. Several finite difference approximations on vorticity were discussed by Roache.[6] We tested one first-order and two second-order schemes suggested by Roache and also the scheme proposed by Dorodnitsyn and Meller,[7] but found that all these formulae could generate convergent solutions and the values of vorticity on the wall computed by these schemes were almost identical. Therefore using different schemes to compute the vorticity on the wall has little effect on the convergence and numerical accuracy of the solution.

The vorticity ω_s on the free surface can be approximated immediately from (1). It is important to note that ω_s is estimated directly from the TSBC in the present approach, whereas ω_s was approximated using u and v that were computed based on values of φ in the previous study.[1] This modification turns out to be quite efficient in free surface iteration.

We took the same approach as Liu *et al.* to determine the pressure; however, we have found that the pressure term is critical for generating convergent solutions as Re increases. If the newly generated pressure replaces the current one in the NSBC (9), it will cause serious numerical oscillations on free surface iteration for high Re. The oscillations can be eliminated if we update the pressure as

$$p^{\text{new}} = p^{\text{old}} + k(p^1 - p^{\text{old}}), \tag{12}$$

where p^{old} is the current value, p^1 is obtained by integrating the pressure gradient equation and p^{new} is the new value to replace p^{old}; k is an adjustable parameter varying between zero and unity and has to be smaller as Re increases.

(III) Updating the free surface

We still adopt the method of Liu *et al.* for the free surface iteration. However, the TSBC and NSBC are modified as shown in (8) and (9). Comparing with the formulae used by Liu *et al.* previously, we have eliminated $\partial v/\partial \eta$ from (8) and $\partial u/\partial \eta$ from (9). Since estimating $\partial v/\partial \eta$ and $\partial u/\partial \eta$ may require many terms far down the jet free surface, eliminating these two terms can generate more accurate solutions. We take the same approach as Liu *et al.* to update the free surface, i.e. the new values of u and v on the free surface are generated and the kinematic condition (7) is used to determine a new position of the free surface. An adjustable parameter k is needed for updating the free surface; we take the same adjustable parameter k for pressure and free surface position for convenience.

RESULTS AND DISCUSSION

We selected the same test case as Liu *et al.*, i.e. the swelling of a creeping Newtonian jet with $Ca = 1000$, and compare the computational efficiency of the present scheme with the work of Liu *et al.* The computations were performed on a CDC Cyber 180/840 machine and the tolerance of convergence was chosen to be $1{\cdot}0 \times 10^{-3}$ for all cases.

If the SLUR method was applied to solve the flow equations, 69·1 CPU seconds were required to generate the convergent solution; however, it took 113·2 CPU seconds for Liu *et al.* to obtain the same solution.

Since the initial guesses on φ, ω and $H(x)$ were given arbitrarily, starting the iteration with the Picard method could not lead to a convergent solution. Instead, the Picard method could be used after the SLUR method was applied and the error of ω was reduced to a preset tolerance.

The SLUR method was first applied until the maximum difference between the previous and current values of ω was less than 0·1; then H was updated and the maximum difference between the previous and current H was computed. The Picard method was now introduced to compute ω and φ and these were iterated several times until the maximum difference of ω was smaller than the maximum difference of H; then H was updated again. This procedure was repeated until H converged. With this new approach the computing time required was reduced to 39·0 CPU seconds.

The jet free surface of the present simulation for the creeping Newtonian jet is much closer to the predictions based on the finite element method[8–10] than the previous finite difference studies. The jet swell ratio based on the present simulation is 1·182, whereas this value was found to be 1·186 by Georgiou *et al.*[9]

The present study, just as the previous case,[1] fails to predict the variations of free surface position for the case $Re > 100$. With $Re > 100$ the NSBC (9) can only generate a v that is almost identical to the current value; therefore the position of the free surface freezes for the case $Re > 100$.

A comparison of the jet swell ratio with previous works is given in Figure 2. The jet swell ratio C_0 predicted by the present method for $Re = 100$ is 0·836, which is close to the value of 0·833 for the potential jet. It is clear that modifying ω_s can generate numerical results that are closer to the predictions based on the finite element methods than previous finite difference studies.

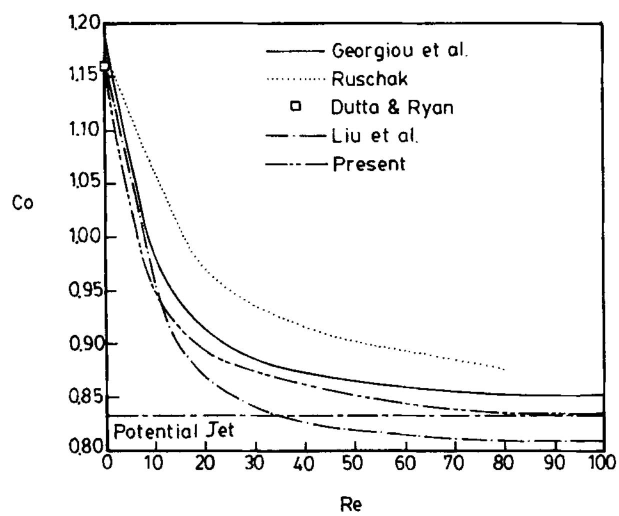

Figure 2. Comparison of the jet swell ratio C_0 with previous studies

CONCLUSIONS

We have improved the finite difference scheme developed by Liu *et al.* previously for the Newtonian jet swell problem. Three modifications have been made, i.e. an algebraic mapping has been adopted, a new formulation for free surface iteration has been proposed and the discrete flow equations can be solved iteratively by a combination of the successive line underrelaxation method and the Picard method.

Since the flow geometry is not very complicated, a simple algebraic approach for numerical mapping is appropriate. There is no need to solve the mapping equations and consequently a substantial amount of computing time can be saved.

We have reformulated the stress conditions on the jet free surface, and the vorticity on the free surface is generated from the tangential stress boundary condition. With these significant modifications we have found that more accurate numerical solutions can be obtained. The combination of the SLUR method and the Picard method can reduce the CPU time effectively.

ACKNOWLEDGEMENTS

This research was supported by the National Science Council of the Republic of China under Grant NSC 79-0405-E007-04. The combination of the SLUR method and the Picard method was suggested by one of the reviewers.

APPENDIX: NOTATION

a	one-half of the exit slot gap
Ca	capillary number, $\mu\langle u\rangle/\sigma$
C_0	jet swell ratio, δ/a
c_1, c_2, c_3	constants, equation (10)
f_A, f_B	constants, equations (4) and (5)
f_m	mesh size upstream, equation (11)
H	location of the free surface, dimensionless
J	Jacobian of mapping equation, equation (6)
k	adjustable parameter, equation (12)
L_1	upstream length, dimensionless
L_2	downstream length, dimensionless
M, N	numbers of grid points in the transformed plane
M_e	grid point at the slot exit in the transformed plane
P, Q	forcing functions for mapping, equations (2) and (3)
p	fluid pressure, dimensionless
R	dimensionless curvature of the jet surface
Re	Reynolds number, $\rho\langle u\rangle a/\mu$
u	velocity component in the x-direction, dimensionless
$\langle u\rangle$	average fluid speed upstream in the slot
v	velocity component in the y-direction, dimensionless
x, y	Cartesian co-ordinates, dimensionless

Greek letters

α, β, γ	coefficients of mapping equations, equation (13)
δ	one-half of the final jet thickness

μ	fluid viscosity
ξ, η	co-ordinates in the transformed plane
ρ	fluid density
σ	surface tension coefficient
φ	streamfunction, dimensionless
ω	vorticity, dimensionless
ω_s	vorticity on the free surface

REFERENCES

1. T. J. Liu, T. A. Yu and S. H. Cheng, 'Finite difference solution of a Newtonian jet swell problem', *Int. j. numer. methods fluids*, **12**, 125–141 (1991).
2. J. F. Thompson, F. C. Thomas and C. W. Mastin, 'Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies', *J. Comput. Phys.*, **15**, 299–319 (1974).
3. J. F. Thompson, Z. U. A. Warsi and C. W. Mastin, 'Boundary-fitted coordinate systems for numerical solution of partial differential equations — a review', *J. Comput. Phys.*, **47**, 1–108 (1982).
4. D. A. Anderson, J. C. Tannehill and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, Hemisphere, New York, 1984.
5. J. F. Thompson, F. C. Thomas and C. W. Mastin, *Numerical Grid Generation: Foundations and Applications*, Elsevier, New York, 1985.
6. P. J. Roache, *Computational Fluid Dynamics*, Hermosa, Albuquerque, NM, 1972.
7. A. A. Dorodnitsyn and N. A. Meller, 'Approaches to the solution of stationary Navier–Stokes equations', *U.S.S.R. Comput. Math. Math. Phys.*, **8**, 205–217 (1968).
8. K. J. Ruschak, 'A method for incorporating free boundary with surface tension in finite-element fluid-flow simulators', *Int. j. numer. methods eng.*, **15**, 639–648 (1980).
9. G. C. Georgiou, T. C. Papanastasiou and J. O. Wilkes, 'Laminar Newtonian jets at high Reynolds number and high surface tension', *AIChE J.*, **34**, 1559–1562 (1988).
10. B. J. Omodei, 'Computer solutions of a plane Newtonian jet with surface tension', *Comput. Fluids*, **7**, 79–96 (1979).