A Reproduced Copy

OF

NASA CONTRACTOR REPORT 166565 - Pt.1

Solution of Three-Dimensional Time-Dependent Viscous Flows
Part 1: Investigation of Candidate Algorithms

Bernard C. Weinberg
Henry McDonald

CONTRACT NAS2-10016
April 1979

NASA

NASA CONTRACTOR REPORT 166565

Solution of Three-Dimensional Time-Dependent Viscous Flows
Part 1: Investigation of Candidate Algorithms

Bernard C. Weinberg
Henry McDonald
Scientific Research Associates, Inc.
Glastonbury, Connecticut

Prepared for
Ames Research Center
under Contract NAS2-10016

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

TABLE OF CONTENTS

	Page
SUMMARY	1
INTRODUCTION	2
LIST OF SYMBOLS	4
ANALYSIS	7
Background	7
Temporal Schemes	13
Consistently Split Block Implicit Scheme	14
Predictor-Corrector Method	17
The Hopscotch Algorithm	19
Spatial Difference Approximations	21
Implicit Tridiagonal Finite Differences	21
Basis Functions	32
RESULTS	46
Numerical Experiments	46
CONCLUSIONS	53
APPENDICES	54
Appendix A - Linearization Technique	54
Appendix B - Discretization of Model Equations	59
REFERENCES	64
TABLES	68
FIGURES	78

SUMMARY

The flow over a helicopter rotor blade in forward flight is an important example of three-dimensional time-dependent flow. The boundary layers on the rotor blade set loss levels and control retreating blade stall. As a consequence there is considerable interest in developing a numerical scheme for solving the time-dependent viscous compressible three-dimensional flow to aid in the design of helicopter rotors. In the present report candidate numerical algorithms are examined to determine their overall suitability for the efficient and routine solution of an appropriate system of partial differential equations. It is concluded that a consistently split time-linearized block implicit scheme using either quintic B-spline collocation or the generalized operator compact implicit approach to generate a fourth order accurate algorithm is particularly well suited for use on the present problem. High cell Reynolds number behavior leads to favoring the generalized operator compact implicit approach over the quintic B-spline collocation method.

INTRODUCTION

The behavior of boundary layers on wings and bodies has long been of interest to aerodynamicists. In both steady and unsteady flows the boundary layers are known to govern a major portion of the losses and to significantly influence the vehicle lift and moment coefficients. When the flow is steady, boundary layer prediction schemes based on numerical solution to the governing partial differential equations of motion have reached a high level of sophistication and predictive accuracy, even in three space dimensions. In unsteady flows, such as are commonly encountered in rotary winged aircraft, some progress has been made in two space dimensions but little to date has appeared on unsteady three-dimensional boundary layers.

Two particular problems arise with time-dependent three-dimensional boundary layers relative to the steady case. The first of these is the rather obvious one of time integration with its added requirements of transient accuracy coupled with an increase in the computational labor. The second of these is the so-called negative cross flow problem, which to some extent has troubled the steady boundary layer prediction schemes. Kendall et al., (Ref. 1) discuss the negative cross flow problem for steady three-dimensional boundary layers in a very illuminating fashion. This particular problem arises when the spanwise component of velocity changes sign and will be discussed in detail subsequently. Because of the interest by external aerodynamicists in swept wing boundary layers where the negative cross flow problem (in this case flow from tip to root) is not usually encountered, the negative cross flow problem has not received a great deal of attention to date. However in transient flows, particularly those encountered on rotor blades in forward flight, negative cross flows are frequently encountered. For instance, the advancing rotor blade has cross flows of one sign during the first ninety degrees of rotation and these can change sign over part of the blade during the second ninety degrees.

Thus to be of practical value, time-dependent three-dimensional boundary layer prediction schemes require high computational efficiency and transient accuracy coupled to the ability to treat arbitrary cross flow profiles. These attributes are not available in any existing available computer code and hence in view of the potential use for a code of this type its development

is appropriate and timely. In this report the initial phase in the development of an efficient time-dependent three-dimensional boundary layer prediction procedure is investigated, namely, the choice of the computational algorithm and spatial differencing technique.

LIST OF SYMBOLS

a	Coefficient in Eq. (6)
$A_1 - A_6$	Coefficients in Eq. (54)
b	Coefficient in Eq. (6)
$B_1 - B_6$	Coefficients in Eq. (55)
B_i	i^{th} B-spline
B_{ki}	i^{th} B-spline of order k
c	Coefficient in Eq. (6)
d	Coefficient in Eq. (6)
D_y	Spatial operator - $au_{yy} + cu_y$
D_z	Spatial operator - $bu_{zz} + du_z$
D_+, D_-, D_0	First order difference operators
$\mathcal{D}_x$	Spatial operator in x direction
$\mathcal{D}_y$	Spatial operator in y direction
$\mathcal{D}_z$	Spatial operator in z direction
F	Function defined in Appendix A
F_j	First derivative approximation at point j (Eq. 12)
G	Function defined in Appendix A
h	Spatial step size
H	Function defined in Appendix A
I	Number of intervals in x direction
J	Number of intervals in y direction
J_{MAX}	Maximum number of intervals
K	Number of intervals in z direction
L	Spatial operator
$L(u)$	Spatial operator
$\mathcal{L}$	Operator defined in Eq. (A7c)
n	n^{th} time level
N	Number of equations
N_i	Nonlinear spatial operator
P	Pressure
q_j^-, q_j^c, q_j^+	Difference weights in Eq. (15)
Q	Operator defined in Eq. (17)
r_j^-, r_j^c, r_j^+	Difference weights in Eq. (15)
R	Operator defined in Eq. (17)

LIST OF SYMBOLS (CONT'D)

Rc_j	Cell Reynolds number
s	Spline approximation to u
S_j	Second derivative approximation at point j (Eq. 13)
S^n	Source term
t	Time
$T_j^0 - T_j^6$	Truncation error terms, Eq. (23)
u	Streamwise velocity component
u_i, u_j	Velocity components
v	Normal velocity component
w	Spanwise velocity component
W_m	Weighting function
x	Streamwise direction
y	Surface normal direction
z	Spanwise direction
$\alpha,\beta,\gamma,\delta,\varepsilon$	Coefficients in model equations (Eq. 56 and Eq. 57)
α_i	Coefficient of B-spline
α_0, α_1	Boundary condition values
β	Parameter to allow centering of time step
β_i	Coefficient of B-spline
Δt	Time step
λ_m	Parameter in Eq. (25)
$\lambda_z = \Delta t/\Delta z^2$	Viscous stability parameter
μ	Roots of characteristic equation (cf. Eq. (26))
ν	Exponent defined in Eq. (23)
ν,α	Coefficients in Burgers equation
ρ	Density
τ	Truncation error
ϕ	Vector of unknowns with components ϕ_1, ϕ_2, ϕ_3
ψ	$\phi - \phi^n$ - increment value

Superscripts

(ℓ)	Order of derivative
m	Defined in Eq. (25)
n	n^{th} time step, $t = n\Delta t$
n+1	$(n+1)^{st}$ time step, $t = (n+1)\Delta t$
-,c,+	Indicates where r and q coefficients are evaluated
$( )'$	Primes indicate derivatives
$(\tilde{})$	Terms in Eq. (9) not divided by a(x)

Subscripts

j	Index - designates grid point
m	Defined in Eq. (39)

ANALYSIS

Background

In this section we discuss the requirements of a three-dimensional unsteady boundary layer code and demonstrate how the physics of the flow influences the choice of a computational technique.

Three-dimensional boundary layers occur on the wings and fuselages of both conventional and rotary wing aircraft. In both types of vehicles, the boundary layers are important in setting loss levels and determining useful operating ranges. As is well known, boundary layers are sensitive to pressure gradients. In time-dependent flow the temporal acceleration terms appear in the momentum equation in a form very similar to the conventional imposed pressure gradient and so for qualitative evaluation purposes can be regarded as 'pseudo' or 'auxiliary' pressure gradients. Viewed in this manner the temporal acceleration terms are likely to be able to influence quantities of practical importance such as skin friction, displacement thickness and the onset of separation. At the range of frequencies typically encountered in rotary wing aircraft aerodynamic problems, it is clear, for instance, from the very thorough review of McCroskey (Ref. 2), that very significant transient boundary layer effects can be observed.

In examining the flow problems of practical interest such as loss levels or the onset of separation it is evident that all three space dimensions must be considered. In conventional aircraft the sweep effect is of interest and inherently three-dimensional. In rotary wing aircraft in forward flight clearly very substantial transient changes occur in what might be termed the local sweep angle. However generally speaking, the boundary layers remain thin unless catastrophic flow separation occurs or the flow at the wing or rotor tip is considered. As a consequence it might be supposed that the usual three-dimensional thin boundary sheet approximations (Nash and Patel, Ref. 3) could be used to produce a valid set of governing equations. Fortunately some improvements in thin boundary sheet approximations are possible as a result of having to treat the negative cross flow problem mentioned earlier.

The negative cross flow problem is best explained in a somewhat intuitive manner, and for steady boundary layers a very good physical description of the problem is given by Kendall et al., (Ref. 1). Looking at the suction surface of a conventional swept back wing the boundary layer cross flow, w, is usually

outward in the z positive direction along the span from root to tip. Thus conventional steady boundary layer integration schemes have developed by forward marching the streamwise velocity u in the streamwise x direction and simultaneously marching out along the span in the z positive direction. In view of the physics of the problem, the spanwise marching scheme does not normally encounter negative w, i.e., spanwise inflow. This is very fortunate because it is difficult, indeed it could be argued impossible, to structure a physically satisfactory unconditionally stable scheme which permits forward marching in the spanwise direction with a negative w cross flow. At least intuitively the problem of negative cross flow implies information being transferred upstream against the spanwise marching direction. Conventional stability analyses confirm the inability to forward march into regions of significant negative w. From experience with attempts to march the two-dimensional boundary layer equations into a region of separated flow and its obvious relationship to the negative cross flow problem, it is not surprising that spanwise marching into a negative cross flow region is not accomplished without special treatment, for instance the Krause "zig-zag" scheme (Ref. 4). Recently conventional boundary layer developers have been turning to a spanwise as well as normal implicit formulation to remove the restriction of only positive cross flows (Kendall et al., Ref. 1). With a spanwise implicit formulation spanwise diffusion is allowed, and the resulting implicit system of equations can be treated by direct elimination (Ref. 1), by a predictor-corrector iterative approach (Ref. 5), or by the process of matrix splitting which reduces the matrix elimination labor (Refs. 6 and 7). Lin and Rubin (Ref. 5) in their predictor-corrector boundary region solutions for flow over a yawed cone at moderate incidence showed that allowing diffusion in the spanwise direction not only eliminates the problems associated with negative cross flow, but improves upon the solutions obtained by three-dimensional boundary layer techniques. Again intuitively a spanwise implicit construction permits information transfer in either direction. Boundary conditions applied at the tip can influence the flow inboard, if required by the physics of the flow. For these reasons the implicit spanwise construction has been a feature of the three-dimensional duct flow analysis of Briley (Ref. 6) and McDonald and Briley (Ref. 7). Based on the experience in Refs. 6 and 7, the additional computational effort resulting from a spanwise implicit formulation could be

as little as a 20% increase relative to the explicit spanwise marching approach. The extension of the steady three-dimensional boundary layer equations to allow spanwise diffusion is easily accomplished, and in view of the improved physical representation which thus follows, it is recommended. As a matter of course it has been assumed that normal to the wall an implicit formulation would be structured. In recent years for boundary layer type problems there has been little dispute as to the efficiency gains to be had from an implicit formulation normal to the wall (Ref. 8). However in the streamwise direction for steady two-dimensional flow, the equations are normally forward marched and the implicit stability obtained entirely from being implicit in the normal to the wall direction.

For unsteady boundary layers the problem can again be formulated in either an explicit or implicit manner. As with spatial marching of steady boundary layers for usual aerodynamic applications, the locally refined spatial mesh required to define a (turbulent) boundary layer influenced by a (transiently) varying pressure distribution, when an explicit (stability restricted) scheme is employed, results in a maximum time step that is much less than the time scale of the physical processes of interest. Thus for solving unsteady boundary layers of the type usually encountered in rotary winged aircraft an implicit formulation is desirable. Since in time-dependent flow diffusion in the streamwise direction is normally negligible due to the usual boundary layer approximations, it is possible to formulate an implicit time-dependent scheme that retains the implicit structure in the spanwise and normal directions (which was found desirable for the steady boundary layer) and march the solution in the streamwise direction.

As mentioned earlier the streamwise marching sweep would probably require less computational effort by about 20% than a fully implicit formulation and of course less storage. However since the solution is being time marched, the opportunity to use a streamwise implicit formulation at roughly the same cost as the streamwise marching sweep does arise. If one does perform a streamwise marching sweep, then the linearization of nonlinear terms is performed about the known spatial marching level. If a fully implicit structure is adopted, then full time linearization can be utilized. That is the linearization of the nonlinear terms is performed about the known time level. As is pointed out in Ref. 7, it is easier to obtain a consistent spatial-temporal order accurate

linearization by marching in time than in space (in time the nonlinear marching derivatives have the form ρu_i whereas in space marching they have the form $\rho u_i u_j$). Further by structuring implicitly in the space marching direction, (small) regions of axial reverse flow would be permitted. As a result of these combined benefits of linearization and separation, a fully implicit structure is advocated.

Transient calculations mean that, in essence, a full three-dimensional spatial integration is carried out at each time step. Thus, spatial accuracy is very important to minimize the spatial grid point density for efficiency since many time steps are contemplated in a given cycle. In order to get the most out of a given spatial difference formula, the errors from representing nonlinear terms by linear combinations of terms should be less than or equal to the spatial discretization errors. If the linearization introduces a greater error than the spatial differencing, then either a coarser spatial mesh could be used, or iteration, or some form of linearization improvement is called for. Iteration across a time step is <u>not</u> recommended since this only reduces the linearization error and computationally costs as much as a complete time step. Cutting back the time step would be preferable to iterating to preserve the linearization error at some acceptable level, since cutting back on the time step would improve <u>both</u> the transient error <u>and</u> the linearization error. To obtain a linearization, which introduces errors of at most the same as the spatial difference formulae, a Taylor series expansion about the known time level can be performed. This process clearly demands a formal block, i.e., coupled, treatment of the system of equations. For instance in the streamwise momentum equation a typical term is linearized:

$$(uw)^{n+1} = u^{n+1}w^{n} + u^{n}w^{n+1} - u^{n}w^{n} + O(\Delta t^2)$$

and clearly one cannot lag w^{n+1} at the old time level n without introducing a first order time error in order to get an uncoupled system, i.e., w^{n+1} <u>not</u> appearing in the streamwise momentum equation. Thus formal linearization and consideration of the resulting errors indicate the coupled system <u>ought</u> to be treated from the accuracy point of view. This is further reinforced when it is realized that direct elimination of block, i.e., coupled, banded systems are not computationally expensive compared to the iterative solution of an

uncoupled system. The linearization technique is described in detail in Appendix A.

Additionally a second type of approximation arises unconnected with linearization but arising from basic coupling terms in the original equations and if indeed some terms in an equation are time lagged in order to uncouple the equation system and these terms are of equal importance to the terms retained, then again an iterative updating is called for in order to achieve stability, accuracy and consistency. (This could be termed ad hoc equation uncoupling). Blottner (Ref. 8) has shown that many iterations around the ad hoc uncoupled set (>10) is sometimes required in order to achieve an overall solution accuracy commensurate with the local difference molecule accuracy. Thompson and MacDonald (Ref. 9) found in a three-dimensional momentum integral procedure that a lagged sequential iterative type of calculation would not, in a number of instances, even converge. The weight of opinion definitely favors the block coupled approach.

As a general observation, care is required to obtain acceptable transient accuracy for long time integration with conventional finite difference schemes. A Crank-Nicolson centered time implicit scheme for instance, although second order in time, shows quite a dispersion problem (relative to other schemes) on the simple pure convection problem. However the problem of transient accuracy is significantly reduced in the typical boundary layer problem since the time dependency is continuously input through initial and boundary conditions and relatively the concern is with 'short' time integrations. The computational problem is more of what the phase lag of the wall shear is, relative to the prescribed free stream disturbance, than concern over the convection velocity of a wave in a shear after a long propagation time. The interest is in forced oscillations with a minimum scale of the boundary layer thickness over a few cycles of the motion, just enough to obtain repetition cyclically. It is therefore expected that a significant dispersion problem will not arise with a conventional implicit scheme.

The equation system which will be considered is formally of block size four, consisting of the continuity and two momentum equations and an 'energy' equation for ρ, u, v, w (p is specified everywhere). If constant stagnation temperature is assumed, ρ, u and w are related by an algebraic equation and the problem can be reduced to a block-three system rather than block-four upon

option, with significant reduction in computer time. Matrix splitting techniques such as those described in Refs. 6 and 7 have considerable potential to reduce the computational labor of solving the block implicit system of (linear) algebraic equations which result from discretizing the governing equations. Schemes of this general type are termed split linearized block implicit or for brevity split LBI schemes and are reviewed in detail by Briley and McDonald (Ref. 10). With a careful ordering of the sweeps with a split LBI scheme it is possible to use a block-two with the third equation uncoupled on two of the sweeps so that this would be a potential major advantage of a split LBI approach.

The ultimate goal of the "optimum" scheme is to diminish both storage requirements and running times in order to achieve a desired accuracy level. Although schemes can be constructed to satisfy either one or both of these goals, the robustness of a method can only be verified by considering its applicability to a general class of problems. Our concern here is with an approximate form of the three-dimensional unsteady compressible Navier-Stokes equations, so that the method chosen will by necessity be required to treat a coupled system of nonlinear partial differential equations. At the outset the following observations can be made concerning the characteristics of the method.

1. Implicit methods (preferably in all three spatial directions) are desired in order to eliminate stability restrictions and permit solutions with both positive and negative streamwise and spanwise velocities.
2. The nature of the nonlinear coupling of the variables in the governing equations require that the equations be solved coupled.
3. Iteration should be avoided and "time linearization" procedures employed (a discussion of this point is given in the following section).
4. The method should allow for general boundary conditions.
5. The method should allow the flexibility of incorporating higher order spatial differencing methods.

Although Item 5 pertains to spatial differencing, in the next section we will demonstrate that it also plays a crucial role in choosing the type of temporal scheme. It would thus appear that first an overall temporal discretization procedure must be chosen and only then an efficient spatial scheme that could be incorporated with it.

The arguments above, linearization, stability considerations and physics lead us to suggest that the current state of the art dictates that the recommended scheme should be in the framework of a time linearized block implicit method.

With the foregoing as background indicating the recommended overall approach to the problem of developing an efficient accurate computer code to predict time-dependent viscous flows, attention is turned to the specific problem dealt with in this report. This report concerns itself with first which overall numerical algorithm to adopt and secondly which spatial differencing scheme to employ or what basis of approximating solution functions to use.

In the next section we will discuss three methods for solving general systems of multidimensional parabolic equations, the consistently split block implicit scheme, Rubin's Predictor-Corrector technique and the Hopscotch algorithm. The following two sections will describe two spatial differencing methods, which we classify as Q-R operator schemes and basis function schemes (which deal mainly with B-splines). Applications of these methods to model two point boundary value problems and to a coupled system of two nonlinear one-dimensional parabolic equations possessing a "three-dimensional boundary layer-like behavior" are then given.

Temporal Schemes

Consider a system of three-dimensional nonlinear parabolic differential equations

$$\phi_t = \mathcal{D}(\phi, \phi_x, \phi_{xx}, \phi_y, \phi_{yy}, \phi_z, \phi_{zz}, x, y, z, t) \qquad (1)$$

where ϕ is a vector of unknowns (ϕ_1, ϕ_2, ϕ_3). With the equations appropriately linearized, at each time step, a system of $N = 3(I-1)(J-1)(K-1)$ linear equations result, where I, J, K are the number of intervals in the x, y, z directions, respectively. Direct inversion of the system is not practical in three space dimensions except for extremely coarse meshes since the operation count is proportional to $N^3/3$ *. Although higher order spatial schemes allows one to reduce the number of grid points, results of model problems indicate that for the range of accuracy desired, one can expect at best a reduction of a factor of four in grid points in each coordinate direction. As significant as this may seem, it may not appreciably affect the overall computation unless the original problem is reduced to a more tractable form.

---

*For $I = J = K = 10$ $N = 2187$ and the matrix inversion operation count is proportional to 10^{10}.

For $I = J = K = 50$ $N = 352947$ and the matrix inversion operation count is proportional to 10^{16}.

The usual procedure is to transform the original problem into a sequence of simpler problems or into a sequence of one-dimensional problems. We consider the linearized version of Eq. (1) (see Appendix A for a description of the time linearization procedure) with the differential operators identified with their coordinate direction, namely

$$(\phi^{n+1} - \phi^n) / \Delta t = (\mathcal{D}_x^n + \mathcal{D}_y^n + \mathcal{D}_z^n)(\beta\phi^{n+1} + (1-\beta)\phi^n) + S^n \qquad (2)$$

where superscripts indicate the time level, i.e., $t^n = n\Delta t$, S^n is a source term and $0 \leq \beta \leq 1$ is a parameter allowing one to center the time step, i.e., $\beta = 0$ corresponds to a forward difference, $\beta = 1/2$ to Crank-Nicolson and $\beta = 1$ to a backward difference.

Consistently Split Block Implicit Scheme

We will first consider the consistently split block implicit scheme. Solution of Eq. (2) is accomplished by application of a generalization to systems of PDE's of an alternating-direction implicit (ADI) technique for parabolic-hyperbolic equations. The original ADI method was introduced by Peaceman and Rachford (Ref. 11) and Douglas (Ref. 12); however, the alternating-direction concept has since been expanded and generalized. A discussion of various alternating-direction techniques is given by Mitchell (Ref. 13), Yanenko (Ref. 14) and more recently by Briley and McDonald (Ref. 10).

The present technique is simply an application of a generalization of the procedure developed by Douglas and Gunn (Ref. 15) for generating consistently split ADI schemes as perturbations of fundamental implicit difference schemes such as the backward-difference or Crank-Nicolson schemes in its natural extension to systems of partial differential equations. In this context a consistent scheme is one where the intermediate levels represent a discrete approximation to the governing equations whose truncation error can be made to vanish as the time or spatial mesh is arbitrarily reduced. Consistency in this sense is a very valuable property as it can greatly simplify the accurate implementation of boundary conditions (Ref. 10).

For the present, it will be assumed that $\mathcal{D}(\phi)$ contains derivatives of first and second order with respect to the coordinate direction, but no mixed derivatives. Mixed derivatives are allowable within the formal framework but unless they are important they are best treated explicitly (lagging) or by extrapolation.

The Douglas-Gunn representation of Eq. (2) can be written as the following three-step solution procedure:

$$(\phi^{*}-\phi^{n})/\Delta t = \beta\mathcal{D}_x\phi^{*} + \left[(1-\beta)\mathcal{D}_x+\mathcal{D}_y+\mathcal{D}_z\right]\phi^{n}+S^{n}$$

$$(\phi^{**}-\phi^{n})/\Delta t = \beta\mathcal{D}_x\phi^{*}+\beta\mathcal{D}_y\phi^{**} + \left[(1-\beta)(\mathcal{D}_x+\mathcal{D}_y)+\mathcal{D}_z\right]\phi^{n}+S^{n}$$

$$(\phi^{***}-\phi^{n})/\Delta t = \beta\mathcal{D}_x\phi^{*}+\beta\mathcal{D}_y\phi^{**}+\beta\mathcal{D}_z\phi^{***} + \left[(1-\beta)(\mathcal{D}_x+\mathcal{D}_y+\mathcal{D}_z)\right]\phi^{n}+S^{n}$$

$$\phi^{n+1} = \phi^{***} + O(\Delta t^{3}) \tag{3}$$

In increment form Eq. (3) reduces to the algorithm given by Briley and McDonald (Refs. 10 and 17) for solving the compressible time-dependent three-dimensional Navier-Stokes equations, viz.,

$$(I-\beta\Delta t\mathcal{D}_x)(\phi^{*}-\phi^{n}) = \Delta t(\mathcal{D}_x+\mathcal{D}_y+\mathcal{D}_z)\phi^{n}+\Delta t S^{n}$$

$$(I-\beta\Delta t\mathcal{D}_y)(\phi^{**}-\phi^{n}) = \phi^{*}-\phi^{n} \tag{4}$$

$$(I-\beta\Delta t\mathcal{D}_z)(\phi^{***}-\phi^{n}) = \phi^{**}-\phi^{n}$$

$$\phi^{n+1} = \phi^{***} + O(\Delta t^{3})$$

where ϕ^{*} and ϕ^{**} are intermediate solutions. Each of Eqs. (4) can be written in narrow block-banded matrix form and solved by efficient block-elimination methods. If ϕ^{*} and ϕ^{**} are eliminated, Eqs. (4) become

$$(I-\beta\Delta t\mathcal{D}_x)(I-\beta\Delta t\mathcal{D}_y)(I-\beta\Delta t\mathcal{D}_z)(\phi^{n+1}-\phi^{n}) = \Delta t\left[(\mathcal{D}_x+\mathcal{D}_y+\mathcal{D}_z)\phi^{n}+S^{n}\right] \tag{5}$$

If the multiplication on the left-hand side of Eq. (5) is performed, it becomes apparent that Eq. (5) approximates Eq. (2) to order $(\Delta t)^2$. Although the stability of Eqs. (4) has not been established in circumstances sufficiently general to encompass the Navier-Stokes equations, it is often suggested (e.g., Richtmyer and Morton, Ref. 16, p. 215) that the scheme is stable and accurate under conditions more general than those for which rigorous proofs are available. This latter notion was adopted here as a working hypothesis supported by favorable results obtained in actual computations (e.g., Refs. 17, 18 and 19).

Several observations can be made concerning Eqs. (3) and (4).

1. The system of three-dimensional equations has been reduced to three systems of one-dimensional equations.
2. The inversion of the total system is now at most approximately 108 (IJ + JK + IK) operations compared to approximately $(3IJK)^3$ operations for the direct inversion problem.
3. The first step involves at least 40% - 50% of the operations (Ref. 7).
4. The method does not have a CFL stability condition and for $\beta \geq 1/2$ is von Neumann stable.
5. The method is applicable to rectangular domains. Although it does not necessarily preserve symmetry along diagonals of rectangular domains, the procedure can be corrected if so desired (Ref. 20).

A major attraction of the Douglas-Gunn scheme is that the intermediate solutions ϕ^* and ϕ^{**} are consistent approximations to ϕ^{n+1}. Furthermore, for steady solutions, $\phi^n = \phi^* = \phi^{**} = \phi^{n+1}$ independent of Δt. Thus, physical boundary conditions for ϕ^{n+1} can be used in the intermediate steps without a serious loss in accuracy and with no loss for steady solutions. In this respect, the Douglas-Gunn scheme appears to have an advantage over locally one-dimensional (LOD) or "splitting" schemes, and other schemes whose intermediate steps do not satisfy the consistency condition. The lack of consistency in the intermediate steps complicates the treatment of boundary conditions and, according to Yanenko (Ref. 14, p. 33), does not permit the use of asymptotically large time steps.

It is worth noting that the operator $\mathcal{D}$ can be split into any number of components which need not be associated with a particular coordinate direction. As pointed out by Douglas and Gunn (Ref. 15), the criterion for identifying

sub-operators is that the associated matrices be "easily solved" (i.e., narrow-banded). Thus, mixed derivatives can be treated implicitly within such a framework, although this would increase the number of intermediate steps and thereby complicate the solution procedure. Finally, only minor changes are introduced if, in the foregoing development of the numerical method, $\mathcal{D}$, and S are functions of the spatial coordinates and time, as well as ϕ.

Alternative techniques to the consistently split linearized block implicit scheme presented here have been proposed as general algorithms. We discuss two such methods, a predictor-corrector scheme due to Rubin (which we will refer to as P/C) and the Hopscotch algorithm of Gourlay and his co-workers. Although these authors have had success with these methods, it will be shown that in order to meet the requirements of the problem under consideration, they are not as versatile or as efficient as the consistently split linearized block implicit scheme previously discussed.

Predictor-Corrector Method

A predictor-corrector method has been successfully employed by Rubin and Lin for three-dimensional viscous flows in which diffusion is important in two directions (Refs. 5 and 21). Their objective in developing a compromise between explicit techniques and implicit methods (ADI) was to eliminate viscous stability restrictions and to minimize CFL stability limitations common to explicit methods and to reduce the total work per time step by eliminating one of the block tridiagonal inversions required in the usual ADI procedure. In addition they desired a "symmetric" method that would easily treat symmetry conditions and other derivative boundary conditions.

A comparison of ADI with Rubin's predictor-corrector method can be obtained by considering the model two-dimensional linear parabolic equation.

$$u_t = (au_{yy} + cu_y) + (bu_{zz} + du_z) \tag{6}$$

The P/C method reduces to the following expression

$$\begin{aligned}\left\{I - \beta\Delta t D_y + \underline{2\lambda_z \beta b}\right\}\left[u^{**} - u^n\right] &= \Delta t\left\{D_y + D_z\right\}u^n \\ &+ \beta\Delta t D_z\left[u^* - u^n\right] + \underline{2\lambda_z \beta b(u^* - u^n)}\end{aligned} \tag{7}$$

where u^* denotes u at a previous iteration, u^{**} denotes u at the latest iteration, $D_y = au_{yy} + cu_y$, $D_z = bu_{zz} + du_z$ and $\lambda_z = \Delta t/\Delta z^2$. As a first guess one of the following extrapolations is used;

linear replacement	$u^* = u^n$
$0(\Delta t^2)$ Taylor series	$u^* = 2u^n - u^{n-1}$

The use of linear replacement achieves consistency only with many iterations or small time steps.

The sequence of steps for solving Eq. (7) by the P/C method are:

1. In Eq. (7) compute $\Delta t(D_y + D_z)u^n$.
2. Compute $\beta\Delta tD_z(u^* - u^n)$ and $2\lambda_z\beta b(u^* - u^n)$ using u^* from a previous iteration.
3. Compute the coefficient matrix of $(u^{**} - u^n)$.
4. Obtain $(u^{**} - u^n)$ (requires one tridiagonal sweep).
5. Repeat steps 2-4 until convergence, usually two to three iterations.

Note that even for linear problems, iteration is necessary to obtain the desired accuracy. Thus at a minimum two to three tridiagonal sweeps and two or three explicit evaluations of the terms in step two are required.

By comparison the Douglas-Gunn ADI procedure gives for the combined two steps

$$\left[I - \beta\Delta t D_y \right]\left[u^{**} - u^n \right] = \Delta t\left[D_y + D_z \right]u^n + \underline{\beta\Delta t D_z\left[u^* - u^n \right]} \qquad (8)$$

where u^* corresponds to an intermediate solution and u^{**} to the solution at the $(n+1)^{st}$ time step. Note that Eq. (8) differs from Eq. (7) in appearance only in the underlined terms in the former equation.

The sequence of steps in the Douglas-Gunn procedure for the two-dimensional problem is as follows:

1. Compute the right-hand side to be used in the first sweep $\Delta t[D_y + D_z]u^n$.
2. Compute $(u^* - u^n)$, which requires one tridiagonal inversion, i.e., $(I - \beta\Delta tD_z)(u^* - u^n) = \Delta t(D_y + D_z)u^n$.
3. Compute $(u^{**} - u^n)$ which requires one tridiagonal inversion, i.e., $(I - \beta\Delta tD_y)(u^{**} - u^n) = (u^* - u^n)$.
4. Evaluate $u^{n+1} = (u^{**} - u^n) + u^n$.

The major effort is expended by both methods in evaluating the term

$\Delta t(D_y + D_z)u^n$. Neglecting the evaluation of the coefficient matrix of $(u^{**} - u^n)$, two passes of P/C is equivalent to one double sweep of ADI. Hence by comparing the sequence of steps for both P/C and ADI we see there is no apparent advantage to the predictor-corrector method. Furthermore, due to the "explicit" nature of the P/C method in the "z" direction it possesses a CFL stability condition while ADI does not.

If we were to consider higher order spatial approximations in both coordinate directions, each operator evaluation could involve a matrix inversion, (see section on spatial approximations). Hence the P/C method could be more costly than an ADI procedure. Finally the P/C method, to the authors' knowledge, has not been formulated for a full three-dimensional problem (with diffusion in all three directions, such as is required for the present problem) so that its applicability under such conditions is unknown. We see no advantage to the P/C method and therefore do not recommend it for the present problem.

The Hopscotch Algorithm

The so-called family of Hopscotch algorithms have been advocated by Gourlay and his coworkers (Refs. 22, 23 and 24) for the solution of multi-dimensional parabolic equations. Several variants of Hopscotch exist; for two-dimensional problems there is the fully explicit odd-even Hopscotch which resembles the DuFort-Frankel algorithm and the partially implicit line and ADI Hopscotch procedures which resemble the Peaceman-Rachford ADI method. These methods have lower operation counts than ADI schemes due to their partial explicit nature that, depending on the scheme, eliminates some or all of the matrix inversions. For instance, line Hopscotch requires only half the number of matrix inversions of a comparable ADI computation. However, the Hopscotch methods have stability restrictions and are only first order accurate in time.

Numerical results of model linear scalar parabolic equations (Ref. 23) when the stability conditions are not violated, confirm the above conclusions, i.e., Hopscotch is more efficient than ADI. However, recent results for the driven cavity problem (Ref. 25), which requires the solution of a coupled system of nonlinear equations, lead to contrary conclusions. The Poisson equation for the stream function was solved separately by a direct method

(Ref. 26), while the vorticity transport equation was solved by both ADI and odd-even Hopscotch. The ADI solutions were 20% faster than Hopscotch to obtain a converged steady state solution. Although Hopscotch requires fewer operations per time step than ADI, the CFL stability condition necessitates the use of a smaller time step which results in the noted increased running times.

More recently, Greenberg (Ref. 27) has added a number of new members to the Hopscotch family for three-dimensional parabolic problems. As with the two-dimensional procedures, these new members also have restrictive stability conditions. To date there has not been widespread use of Hopscotch-type schemes, in particular for coupled nonlinear parabolic equations, and their viability under such circumstances is still an open question.

In general, Hopscotch owes its favorable characteristic to what Gourlay terms E-operators, of which the standard tridiagonal finite differences is a member. For other higher order spatial differencing, where the operators are tridiagonal but not E-operators, e.g., evaluations of spatial operators of the form $au_{xx} + bu_x$ which even for an explicit temporal scheme involves a matrix inversion, the method loses most of its desirable features (see section on spatial approximations). In addition the ability to handle coupled implicit boundary conditions is also not as flexible with Hopscotch. Finally, there have been claims that one of the attributes of Hopscotch is its ease of programming. While this may be true for model problems, for more complex problems, i.e., coupled systems of three-dimensional nonlinear parabolic equations, the basic logic for setting up the block inversions, whether in one direction as in Hopscotch, or in three directions as in split LBI, is comparable. Hence, in this case, programming considerations should not greatly influence one's choice of method.

The lack of versatility of the method, i.e., stability restriction, inability to incorporate higher order spatial methods, and results of more realistic problems (Ref. 25) lead us not to recommend the Hopscotch algorithm.

Spatial Difference Approximations

Implicit Tridiagonal Finite Differences

Q-R Operator Notation

In this section implicit tridiagonal finite difference approximations to the first and second derivatives and to the spatial differential operator will be considered. The very versatile Q-R operator notation will be introduced, which allows as special cases a variety of schemes such as standard second order finite differences, first order upwind differences, fourth order operator compact implicit (OCI), fourth order generalized OCI and exponential type methods. Since all these schemes are of the same form, a single subroutine which defines the difference weights is all that is required to identify the method, while leaving the basic structure of the program unaltered. Subsequently, the results of numerical experiments for a number of these schemes will be presented.

The Q-R formulation allows for ADI methods and permits the treatment of systems of coupled equations, i.e., LBI methods. Although variable mesh schemes can be employed within the Q-R framework, it is believed preferable to use analytic transformations to obtain a uniform computational mesh, hence attention is restricted to uniform mesh formulations.

The general concepts and notation for two point boundary value problems will be introduced and then the methodology entended to more general linear and nonlinear parabolic partial differential equations in one dimension. The extension to multidimensional problems will also be indicated.

Consider the two point boundary value problem

$$\tilde{L}(u) = \tilde{a}(x)u_{xx} + \tilde{b}(x)u_x + \tilde{c}(x)u = \tilde{f}(x) \tag{9}$$

with u(0) and u(1) prescribed. Derivative boundary conditions, although not treated here, can easily be incorporated into the framework of the Q-R operator notation. Let the domain be discretized so that $x_j = (j-1)h$, $j = 1, 2, \ldots, J+1$, and $U_j \sim u(x_j)$, $F_j \sim u_x(x_j)$, $S_j \sim u_{xx}(x_j)$ and $h = 1/J$ is the mesh width. The numbering convention was chosen here to be compatible with FORTRAN coding.

Without loss in generality for $a(x) \neq 0$, Eq. (9) can be divided by $a(x)$ so that we may treat instead the following equation

$$\bar{L}(u) = u_{xx} + b(x)u_x + c(x)u = f(x) \tag{10}$$

where

$$b(x) = \tilde{b}(x)/\tilde{a}(x),\ c(x) = \tilde{c}(x)/\tilde{a}(x) \text{ and } f(x) = \tilde{f}(x)/\tilde{a}(x)$$

The spatial differential operator is identified as

$$L(u) = u_{xx} + b(x)u_x + c(x)u \tag{11}$$

Substituting the finite difference approximations to the first and second derivatives

$$\frac{D_o}{2h}U_j = \frac{U_{j+1} - U_{j-1}}{2h} = F_j = u_x(x_j) + O(h^2) \tag{12}$$

$$\frac{D_+D_-}{h^2}U_j = \frac{U_{j-1} - 2U_j + U_{j+1}}{h^2} = S_j = u_{xx}(x_j) + O(h^2) \tag{13}$$

into Eq. (10) and rearranging, we obtain

$$L(u) \sim S_j + b_jF_j = \left[\frac{1}{h^2} - \frac{b_j}{2h}\right]U_{j-1} + \left[c_j - \frac{2}{h^2}\right]U_j + \left[\frac{1}{h^2} + \frac{b_j}{2h}\right]U_{j+1} = f_j$$

or

$$\left[1 - \frac{Rc_j}{2}\right]U_{j-1} + \left[h^2c_j - 2\right]U_j + \left[1 + \frac{Rc_j}{2}\right]U_{j+1} = h^2f_j \tag{14}$$

where $Rc_j = hb_j$ is the cell Reynolds number.

Equation (14) can be generalized by introducing operator format, i.e.,

$$r_j^- U_{j-1} + r_j^c U_j + r_j^+ U_{j+1} = h^2(q_j^- f_{j-1} + q_j^c f_j + q_j^+ f_{j+1}) \tag{15}$$

where the superscripts (-) minus, (c) center, and (+) plus indicate the difference weight that multiplies the variable evaluated at the (j-1), (j) and (j+1) grid points, respectively, and where the r_j's and q_j's for grid point j are functions of h, b_{j-1}, b_j, b_{j+1}, c_{j-1}, c_j and c_{j+1}. Comparing Eqs. (14) and (15) we can identify the r_j's and q_j's, viz.,

$$\begin{aligned} r_j^- &= 1 - Rc_j/2 & q_j^- &= 0 \\ r_j^c &= h^2 c_j - 2 & q_j^c &= 1 \\ r_j^+ &= 1 + Rc_j/2 & q_j^+ &= 0 \end{aligned} \tag{16}$$

We now define the tridiagonal difference operators Q and R

$$\begin{aligned} R[U_j] &= r_j^- U_{j-1} + r_j^c U_j + r_j^+ U_{j+1} \\ Q[f_j] &= q_j^- f_{j-1} + q_j^c f_j + q_j^+ f_{j+1} \end{aligned} \tag{17}$$

Noting that L(u) = f and substituting Eq. (17) into Eq. (15) we obtain

$$R[U_j] = h^2 Q[L(u)_j] = h^2 Q[f_j] \tag{18}$$

Alternatively by employing the inverse operator Q^{-1} an expression for $L(u)_j$ can be obtained

$$L(u)_j = \frac{1}{h^2} Q^{-1} R U_j \tag{19}$$

For standard central finite differences $Q = Q^{-1} = I$, the identity matrix, (the spatial operator is given explicitly in terms of U_{j-1}, U_j and U_{j+1}) so that nothing was gained in obtaining Eq. (19). However, in general, for higher order methods Q is tridiagonal and Q^{-1} is a full matrix. Hence Eq. (19) gives us a means of expressing the spatial operator for a wider class of difference approximations. The formalism in Eq. (19) is also applicable for first and second derivatives appearing alone (cf. Ref. 28). It must be pointed out however that Eq. (19) is not the most general formulation since the compact implicit formulas cannot be combined to yield a single scalar equation relating the spatial operator to the function values (Ref. 28).

In the next section a method due to Berger et al., (Ref. 29) is described that enables one to construct fourth order tridiagonal methods with certain desirable properties, i.e., evaluate the q_j and r_j coefficients.

Generalized Operator Compact Implicit Schemes

Given

$$L(u) = u_{xx} + b(x)u_x + c(x)u$$

an expression relating L(u) and u is sought in the form

$$\frac{1}{h^2} RU_j = Q(Lu)_j + \tau_j \tag{20}$$

where τ_j is the truncation error and Q and R are tridiagonal displacement operators. The maximum accuracy attainable is fourth order, i.e., $\tau_j \sim O(h^4)$.

Expanding Eq. (20) in terms of $q_j^{-,c,+}$ and $r_j^{-,c,+}$ we obtain,

$$\begin{aligned}\tau_j = \frac{1}{h^2} RU_j - Q(Lu)_j = \frac{1}{h^2}\left[r_j^- U_{j-1} + r_j^c U_j + r_j^+ U_{j+1}\right] \\ - \left[q_j^-(Lu)_{j-1} + q_j^c(Lu)_j + q_j^+(Lu)_{j+1}\right]\end{aligned} \tag{21}$$

A Taylor series expansion yields for τ_j

$$\tau_j = T_j^0 u(x_j) + T_j^1 u^{(1)}(x_j) + T_j^2 u^{(2)}(x_j) + T_j^3 u^{(3)}(x_j) + T_j^4 u^{(4)}(x_j) + T_j^5 u^{(5)}(x_j) + T_j^6 u^{(6)}(x_j) + O(h^5) \tag{22}$$

where superscripts in parentheses denote derivatives with respect to x, and where

$$T_j^0 = \frac{1}{h^2}\left[(r_j^- + r_j^c + r_j^+) + h^2(q_j^- c_{j-1} + q_j^c c_j + q_j^+ c_{j+1})\right]$$

$$T_j^1 = \frac{1}{h}\left[(r_j^+ - r_j^-) - h(q_j^- b_{j-1} + q_j^c b_j + q_j^+ b_{j+1}) - h^2(q_j^+ c_{j+1} - q_j^- c_{j-1})\right]$$

$$T_j^2 = \frac{1}{2}(r_j^+ + r_j^-) - (q_j^- + q_j^c + q_j^+) - h(q_j^+ b_{j+1} - q_j^- b_{j-1}) - \frac{h^2}{2}(q_j^+ c_{j+1} + q_j^- c_{j-1}) \tag{23}$$

$$T_j^\nu = h^{\nu-2}\left\{\frac{1}{\nu!}(r_j^+ + (-1)^\nu r_j^-) - \frac{h}{(\nu-1)!}(q_j^+ b_{j+1} + (-1)^{\nu-1} q_j^- b_{j-1}) - \frac{1}{(\nu-2)!}(q_j^+ + (-1)^\nu q_j^-) + h^2(q_j^+ c_{j+1} + (-1)^\nu q_j^- c_{j-1})\right\} \quad \nu = 3,4,5,6$$

For second order central finite differences we set $T^0 = T^1 = T^2 = 0$. This yields, when $q_j^c = 1$ and $q_j^- = q_j^+ = 0$, the following relations

$$r_j^c = -(r_j^- + r_j^+) + h^2(q_j^- c_{j-1} + q_j^c c_j + q_j^+ c_{j+1})$$

$$r_j^+ - r_j^- = hb_j = Rc_j$$

$$r_j^+ + r_j^- = 2$$

which recovers Eq. (16), i.e.,

$$r_j^c = -2 + h^2 c_j$$

$$r_j^- = 1 - Rc_j/2$$

$$r_j^+ = 1 + Rc_j/2$$

To obtain the fourth order operator compact implicit scheme we again set $T^0 = T^1 = T^2 = 0$ to obtain three expressions for $r_j^{-,c,+}$ in terms of the $q_j^{-,c,+}$, (note that q^-, $q^+ \neq 0$ and q^c is not necessarily unity), i.e.,

$$r_j^c = -(r_j^+ + r_j^-) + h^2(q_j^- c_{j-1} + q_j^c c_j + q_j^+ c_{j+1}) \tag{24a}$$

$$r_j^+ - r_j^- = Rc_{j-1}q_j^- + Rc_j q_j^c + Rc_{j+1}q_j^+ + h^2(q_j^+ c_{j+1} + q_j^- c_{j-1}) \tag{24b}$$

$$r_j^+ + r_j^- = 2(q_j^- + q_j^c + q_j^+) + 2\left[Rc_{j+1}q_j^+ - Rc_{j-1}q_j^-\right] + h^2(q_j^+ c_{j+1} + q_j^- c_{j-1}) \tag{24c}$$

Now T^3 and T^4 must be set to evaluate $q_j^{-,c,+}$. The standard Swartz OCI method requires $T^3 = T^4 = 0$,

$$\frac{1}{6}(r^+ - r^-) - \frac{1}{2}\left[Rc_{j+1}q_j^+ + Rc_{j-1}q_j^-\right] - \left[q_j^+ - q_j^-\right] = 0 \\ - h^2(q_j^+ c_{j+1} - q_j^- c_{j-1}) \tag{24d}$$

$$\frac{1}{24}(r^+ + r^-) - \frac{1}{6}\left[Rc_{j+1}q_j^+ - Rc_{j-1}q_j^-\right] - \frac{1}{2}\left[q_j^+ + q_j^-\right] = 0 \\ - h^2(q_j^+ c_{j+1} + q_j^- c_{j-1}) \tag{24e}$$

and results in a leading truncation error term of $(\tilde{A}u^{(5)} + \tilde{B}u^{(6)})h^4$. Substituting (24b) and (24c) into (24d) and (24e) r_j^+ and r_j^- can be eliminated and a system of two equations in q_j^- and q_j^c with q_j^+ as a parameter is obtained. The parameter q_j^c is proportional to the determinant of the system. The values of $q_j^{-,c,+}$ and $r_j^{-,c,+}$ are presented in Table I.

As shown in Ref. 28, a cell Reynolds number stability condition exists for the Swartz OCI scheme, i.e., for $Rc \geq \sqrt{12}$ nonrealistic or oscillatory solutions will be obtained. In order to eliminate this restriction one can relax the conditions $T^3 = T^4 = 0$, and allow the coefficients of $u^{(3)}$ and $u^{(4)}$ to be of $O(h^4)$.

By expanding $q_j^{-,c,+}$ in a series in Rc

$$q^{-,c,+} = \sum_{m=0}^{3} \lambda_m^{-,c,+} Rc_j^m \tag{25}$$

12 parameters, $\lambda_m^{-,c,+}$ $m = 0, 1, 2, 3$, are introduced. The equations for $T^3 = O(h^4)$ and $T^4 = O(h^4)$ yield 5 linear relations, leaving for disposal 6 "free" parameters plus a factor.

These parameters can be set according to some criteria that would yield certain desirable properties for the difference equations. The following constraints are prescribed

$$\begin{gathered} q_j^+ \geq 0,\ q_j^- > 0,\ q_j^c > 0 \\ q_j^c b_j \geq q_j^- b_{j-1} + q_j^+ b_{j+1} \\ r_j^+ > r_j^- \geq 0 \\ -r_j^c \geq r_j^+ + r_j^- \end{gathered} \tag{26}$$

and h is sufficiently small so that

$$\begin{gathered} 10b_j - b_{j-1} - b_{j+1} > 0 \text{ and } hc_{j+1}/b_{j+1} < 2 \\ \text{for } j = 2, \cdots, J \text{ and } c_j \leq 0 \end{gathered}$$

These conditions assure that R is diagonally dominant and Q is invertible for all R_c. Further details are given in Ref. 29. The significance of this approach is that one can construct a scheme possessing certain desired properties employing a set of preassigned rules. This is contrary to usual practice, in which a scheme is chosen, and then its properties are determined. It is important to note that the derivation of the q and r coefficients is not a trivial task. In computational effort it is also not cheap. This point will be discussed in greater detail in a later section.

The q and r coefficients for the generalized OCI scheme described in Ref. 29 are given in Table II. In this report comparisons are made with another generalized OCI scheme, whose coefficients are given in Table III. Numerical experiments indicate that these two schemes are comparable, differing only in the magnitude of the truncation error.

Exponential Type Schemes

Another family of schemes that can be expressed in Q-R operator notation are the so-called exponential methods. The idea, originally due to Allen (Ref. 30) (independently derived by Il'in (Ref. 31) and McDonald (Ref. 32)) and employed by Dennis (Ref. 33), is to set the difference weights so that the numerical solution is equated to the analytic solution for the locally frozen constant coefficient equation. Allen (Ref. 30) and Il'in (Ref. 31) considered the homogeneous constant coefficient equation,

$$L(u) = u_{xx} + bu_x = 0 \tag{27}$$

so that the difference approximation was set identically equal to the analytic solution.

The analytic solution of Eq. (27) has the form

$$u = A + Be^{-bx}$$

where A and B are determined from the boundary conditions.

The Q-R operator formulation of Eq. (27),

$$r_j^- U_{j-1} + r_j^c U_j + r_j^+ U_{j+1} = 0 \tag{28}$$

possesses a fundamental solution of the form μ^j. Solving for μ and employing Eq. (24a) we obtain

$$\mu = 1,\ r_j^+ / r_j^-$$

With the aid of Eq. (24b) and setting μ^j to the analytic solution $e^{-b_j x}$ we obtain

$$\mu^j = e^{-b_j x} = e^{-b_j h j} = e^{-Rc_j} = r_j^+ / r_j^-$$

With the condition $r_j^+ - r_j^- = Rc_j$ ($T_j^1 = 0$ in Eq. 23), we can define $r_j^-, c, +$ as

$$r_j^- = Rc\, e^{-Rc_j} / (1 - e^{-Rc_j})$$

$$r_j^+ = Rc_j / (1 - e^{-Rc_j}) \qquad (29)$$

$$r_j^c = -Rc_j$$

and $q^- = q^+ = 0$ and $q^c = 1$ where we have allowed b_j to vary. Equation (29) can be rearranged to yield an alternate form

$$L(u)_j = \frac{Rc_j}{2}\left[\coth\frac{Rc_j}{2}\right] D_+ D_- u_j + b_j D_o u_j \qquad (30)$$

where D_+, D_-, D_o are the forward, backward and central first difference operators, respectively.

This method is second order accurate for $Rc \sim O(1)$ and becomes first order accurate as $Rc \to \infty$ where the scheme reverts to first order upwind differencing (in Eq. (28) $r_j^- \to 0$, $r_j^+ \to Rc_j$, $r_j^c \to -Rc_j$ for $b_j > 0$ to give $Rc_j(U_{j+1} - U_j) = 0$).

Another exponential scheme which is uniformly second order accurate was developed by El-Mistikawy and Werle (Refs. 34 and 35). The "exponential box scheme" which is incorporated in their solution of the boundary layer equations with strong blowing, is based on a spatial operator of the form given in Eq. (10). Berger et al., (Ref. 36) derived the counterpart for an operator of

the form given in Eq. (27). The q and r coefficients are presented in Table IV. Although this scheme reverts to second order upwind differences as $Rc \to \infty$, it does not possess a maximum principle analogous to the ordinary differential equation it is approximating as does the exponential scheme of Allen (Ref. 30).

The Allen exponential scheme can be applied to differential equations of the type given in Eq. (10). Substituting the Q-R representation for $\bar{L}(u)$, we obtain

$$\frac{1}{h^2} Q^{-1} R\left[U_j\right] + c_j U_j = f_j \tag{31}$$

Multiplying through by Q, and combining terms, the difference approximation to Eq. (10) becomes

$$R\left[U_j\right] + h^2 Q\left[c_j U_j\right] = h^2 Q f_j$$

or

$$\left(R + h^2 Q c_j\right)\left[U_j\right] = h^2 Q f_j \tag{32}$$

Note that the only difference between Eq. (32) and Eq. (18) (where $c_j \equiv 0$) is the coefficient matrix multiplying U_j. Hence the methodology is unaltered.

Application of Q-R Operator Schemes to 1-D Parabolic Equations

Consider the one-dimensional linear parabolic equation*

$$u_t = a u_{xx} + b u_x + c u + d \tag{33}$$

with appropriate boundary conditions and initial conditions, where a, b, c, d could be functions of x and t. Dividing by $a \neq 0$, identifying $L(u) = u_{xx} + b u_x$,

*If Eq. (33) were nonlinear, assume it was linearized by the method described in Appendix B.

and employing a two point temporal difference scheme, we obtain

$$\left(\frac{1}{a_j}\right)^{n+\beta}\left(\frac{u_j^{n+1}-u_j^n}{\Delta t}\right) = L_j^{n+\beta}\left(\beta u_j^{n+1} + (1-\beta)u_j^n\right) + d_j^{n+\beta}$$

where $\beta = 1$ reduces to a backward difference and $\beta = 1/2$ to the Crank-Nicolson approximation.

Letting

$$L^{n+\beta} = \frac{1}{h^2}\left[Q^{n+\beta}\right]^{-1}\left[R^{n+\beta}\right]$$

and with some algebra we obtain,

$$\left\{Q^{n+\beta}\left(\frac{1}{a_j^{n+\beta}}\right) - \lambda\beta R^{n+\beta}\right\} u_j^{n+1} \tag{34}$$

$$= \left\{Q^{n+\beta}\left(\frac{1}{a_j^{n+\beta}}\right) + \lambda(1-\beta)R^{n+\beta}\right\} u_j^n + \Delta t\, Q^{n+\beta}\left[d_j^{n+\beta}\right]$$

where $\lambda = \Delta t/h^2$.

Again, Eq. (34) is general and permits second order finite differences, upwind differencing, exponential type schemes, and OCI schemes. Aside from the evaluation of the Q and R operators, the problem is no more complicated than standard second order finite differences. Results of model problems can be found in Ref. 28.

Application to Coupled Nonlinear Parabolic Equations

Given a system of m nonlinear parabolic equations in m unknowns,

$$\sum_{i=1}^{m}\left\{\frac{1}{a_{ij}^{n+\beta}}\,\frac{(u_{ij}^{n+1}-u_{ij}^n)}{\Delta t} - N_i^{n+\beta}(u_1, u_2, \ldots u_m, x_1, x_2, x_3, t)\right\} = 0$$

$$j = 1, 2, \ldots, J+1$$

where $N_i^{n+\beta}$ is a nonlinear nonconservative spatial operator, the Q-R formalism carries directly over provided that for any equation only one independent variable is operated upon by the differential operator. For example,

$$\frac{1}{a(u,w,v)} u_t = u_{xx} + b(u,v,w)u_x + c(u,v,w)$$

is allowed since x derivatives of u only appear, while

$$\frac{1}{a(u,w,v)} u_t = u_{xx} + b(u,v,w)u_x + c(u,v,w) + d(u,v,w)w_x$$

is not allowed since x derivatives of both u and w appear. The modified unsteady Navier-Stokes equations for the three-dimensional time-dependent boundary layer, when written in quasilinear form, fall within the class of allowable differential operators. Thus for the problem being addressed in the present study the OCI schemes are applicable.

Multidimensional problems and/or more general equation forms can usually be accommodated by a splitting procedure, which reduces the differential operator to a sequence of one-dimensional problems which have the appropriate allowable form. However, as with standard finite differences, special procedures must be applied to cross derivative terms, e.g., extrapolation or lagging at the previous time step or increasing the number of intermediate steps in the splitting.

In Appendix B an example of the above procedure is presented for a coupled system of one-dimensional nonlinear parabolic equations possessing "boundary layer-like reverse flow" behavior.

Basis Functions

A convenient approach for developing numerical procedures for the solution of partial differential equations is founded on a basis function representation of the dependent variable, i.e., in one dimension,

$$u(x) = \sum_i a_i B_i(x)$$

where u(x) is represented by a linear combination of suitably chosen basis functions, $B_i(x)$, and the α_i are determined by some set of constraint relationships. The advantage of the basis function representation is that once the α_i are determined, one can obtain at very little additional cost the function and its derivatives anywhere in the domain.

The major concern here will be with the B-spline basis functions. However by way of comparison some of the characteristics of piecewise Lagrange polynomials and piecewise Hermite polynomials will also be given. In the following basis functions are sought, polynomials of order k (degree k-1), that possess certain smoothness properties, and the computational efficiency of these functions for the solution of differential equations are investigated.

The simplest functions are the piecewise Lagrange polynomials, which can be computed by the cardinal basis functions. Consider a grid numbered from j = 1 to J + 1 (to be consistent with FORTRAN coding) so that one is considering J + 1 grid points and J intervals. Over each interval consisting of k knots the basis functions are polynomials of degree k-1 which are equal to unity at one particular knot and zero at the other knots. The dimension of the piecewise Lagrange polynomials (the number of basis functions) is J + 1 (independent of the degree of the polynomial) and thus J + 1 constraints must be satisfied to determine all the α_i. However, in order to evaluate a function and its derivatives at a particular location, only k basis functions distributed over k adjacent grid points are required. In addition Lagrange polynomials are only C^o at the end points of each sub-interval, again independent of the degree of the polynomial and thus allows for jumps in the first derivative there. Due to the purely interpolatory character of piecewise Lagrange polynomials the α_i are nothing more than the functions evaluated at the appropriate knots.

The interpolation problem reduces to an explicit procedure as does the evaluation of the derivatives at the internal knots. When incorporated with the method of collocation (the analog of interpolation for the solution of ordinary differential equations), the piecewise Lagrange basis functions used locally on a uniform mesh (with k odd) recovers the standard centered finite difference approximations. Thus a quadratic polynomial leads to a three point formula and the inversion of a tridiagonal matrix of order J + 1 and a quartic

polynomial leads to a five point formula and the inversion of a pentadiagonal matrix of order J + 1.

In order to obtain smoother representations additional constraints must be applied on the derivatives of the approximating functions. This leads to more complicated basis functions. For example, the family of Hermite polynomials requires that for a (2m - 1) degree polynomial ($m \geq 2$) on two adjacent knots, m interpolatory constraints must be satisfied for the function and its (m-1) derivatives. The approximation is still local, i.e., given a function and an appropriate number of derivatives on two adjacent knots, the function and its derivatives can be computed implicitly in the interior of that domain. However, the solution of a differential equation by the method of collocation involves the inversion of a matrix of order m(J + 1) of bandwidth (3m - 1) (= n + m where n = degree of polynomial). Thus in the process of obtaining a C^{n-2} representation a substantial increase in labor in comparison to Lagrange polynomials has been incurred. Further details can be found in Ref. 37.

B-Splines

Another family of functions are the polynomial splines, i.e., polynomials of degree k-1 that are C^{k-2}. One would expect that the additional smoothness of the spline approximation would translate into a better behaved solution - perhaps even more accurate - for "smooth" functions, than a Lagrange or Hermite polynomial of the same degree. Numerical experiments have shown this to be the case in many instances. However, the additional smoothness constraints that must be specified not only change the nature of the approximation, but may add considerably to the computational effort, depending on the manner in which they are applied.

There are several approaches for representing polynomial splines. The first method involves specifying the polynomial and the smoothness conditions separately, and then solving the imposed constraint equations* simultaneously for the undetermined coefficients. This approach has successfully been

*The constraint equations may either be interpolatory conditions or a differential equation.

employed by Kendall and Bartlett (Ref. 38) in their solution of the chemically reacting boundary layer equations, and Murphy et al., (Ref. 39) for the solution of the boundary layer equations. With the aid of a skillful partitioning of the resulting banded block matrix, the above authors were able to solve for the function values directly. However, by so doing they lost ready access to the interpolatory polynomial representation, which is one of the acknowledged benefits of a spline technique. An alternate approach described in Ahlberg, Nilson and Walsh (Ref. 40) and adapted by Rubin and Khosla (Refs. 41 and 42) in their solution of the boundary layer equations also employs a polynomial formulation. However, the additional continuity and smoothness conditions are used to transform their equations into expressions relating the function values and their derivatives. Their "spline equations" are solved simultaneously with the appropriate constraint equations, which entails for a single scalar differential equation the inversion of 2 x 2 block tridiagonal matrix to obtain fourth order accuracy and a 3 x 3 block tridiagonal matrix to obtain sixth order accuracy.

In order to avoid inverting block banded matrices that arise even in the solution of scalar differential equations a third alternative is considered, the B-spline representation, which has the appropriate smoothness built into the functions themselves. For the solution of the coupled Navier-Stokes equations this procedure has a definite advantage in that the order of the block submatrices, typically three or four, can be reduced by one. A detailed discussion of B-splines, their construction and their mathematical properties are given in Refs. 37 and 43 and will not be repeated here. Instead, the properties of B-splines that make them attractive for the solution of partial differential equations will be emphasized. In the following discussion, without loss in generality, a uniform mesh will be considered.

The normalized B-spline (the sum of the basis functions at any knot is unity) of order k (degree k-1) is bell-shaped and spans k + 1 knots. Within the interval the B-spline is positive while outside the interval, including the end points, it is zero. For knots of multiplicity unity the first k - 2 derivatives of the B-spline are zero with a jump possible in the (k - 1)st derivative. In general, for a knot of multiplicity k - ν the νth derivative of the B-spline is discontinuous at the point with all lower order derivatives continuous. Due to the compact support property of the B-splines at most k

B-splines are nonzero over any interval, while at a knot k - 1 B-splines are nonzero. This property enables one to reduce the bandwidth of the matrices that are obtained in the solution of differential equations, while retaining the scalar structure.

DeBoor (Ref. 44) has shown that for B-splines of order k the dimension of the basis is J + 1 + k - 2, where it is assumed that the knots are of multiplicity unity and there are J + 1 nodal points (J intervals) in the domain. Thus for cubics (k = 4) there are J + 3 independent basis functions and for quintics (k = 6) there are J + 5 basis functions. Hence J + 1 + k - 2 conditions are required to fully specify the B-spline representation. In view of the above, for a B-spline of order k, in order to determine all the α_i, the resulting matrix is scalar, is of order J + 1 + k - 2 and has a bandwidth of k. After the α_i are determined, the evaluation of the function and its k - 2 derivatives each require at most k additional multiplications.

For the pure interpolation problem

$$u^{(\ell)}(x_1) = f_1^{(\ell)} \qquad \ell = 1, \ldots, K-3$$

$$u(x_j) = f_j \qquad j = 1, \ldots, J+1$$

$$u^{(\ell)}(x_{J+1}) = f_{J+1}^{(\ell)} \qquad \ell = 1, \ldots, K-3$$

where superscript ℓ designates a derivative of order ℓ, and k even, Prenter (Ref. 37) shows that a unique function exists, the polynomial spline of order k. What is apparent from the statement of this problem is that the B-spline interpolating polynomial is global in character, being "tied together" by the derivative constraints at the end points. In contrast to piecewise Lagrange and Hermite polynomials where coefficients were determined for each subinterval by the appropriate interpolatory constraints, the evaluation of B-splines requires the inversion of a matrix. In addition the α_i's that are determined are not equal to the values of the functions and its derivatives at the knots as was the case for the Lagrange and Hermite representations. It is in part due to this global character that B-splines derive their smoothness properties.

DeBoor (Ref. 44) has presented a computer program to determine the normalized B-splines of order $k < 20$, which allows for nonuniform meshes and for discontinuities in lower order derivatives at the knots (knots of multiplicity > 1). He has also conditioned the B-splines near the boundaries to simplify the setting of boundary conditions. Alternatively the procedure suggested by Prenter (Ref. 34) could be employed (for a uniform mesh with simple knots), but that would require additional algebraic manipulation to modify the coefficient matrix so that one can obtain the appropriate band structure. Since both methods are equivalent (the uniqueness of the B-spline representation) the choice is a matter of convenience. Prenter's approach employs the same basis functions throughout the domain, which reduces the memory and computer logic requirements but increases the algebraic manipulation, in particular for higher order splines. The more general DeBoor approach requires more computer memory, but reduces the preprocessing required of the user. In the subsequent work DeBoor's approach was used.

The accuracy of the B-spline representation is now considered. For a function u that is sufficiently smooth, with derivatives of order k that are continuous, $u \in C^k(a,b)$, the B-spline representation of u, s (of order k, k even), Prenter gives the following error estimates

$$\|u-s\|_\infty = O(h^k)$$

$$\|u'-s'\|_\infty = O(h^{k-1})$$

$$\|u''-s''\|_\infty = O(h^{k-2})$$

Hence, to solve a second order ODE by collocation using cubic splines ($k = 4$) we can expect $O(h^2)$ accuracy but require only a simple tridiagonal matrix elimination. However, to obtain greater accuracy larger bandwidth matrices are required. Consideration of this point is given in the following section.

Application of B-Splines to the Solution of Differential Equations

Since the B-spline representation will be employed as a spatial approximation, its properties, order of accuracy, efficiency and spatial stability

behavior can be investigated by considering its application to the solution of two point boundary value problems.

In this section B-spline basis functions of order k are applied to the solution of ordinary differential equations of the form

$$L(u) = u_{xx} + bu_x + cu = f \qquad x\epsilon[0,1] \tag{35}$$

with boundary conditions

$$u(0) = \alpha_0 \ , \ u(1) = \alpha_1$$

Several different approaches can be employed for the solution of Eq. (35), depending on the degree of accuracy desired and the level of complexity one is willing to accept. The techniques of collocation, Galerkin, subdomain, least squares, etc., can be viewed as special cases of the method of weighted residuals (of Crandall (Ref. 45) for instance). More recently Murphy (Ref. 46) has shown the relationship of these methods to orthogonalization processes and has characterized them as generalized Galerkin techniques. Here we will review the relationship of the different approaches to the method of weighted residuals and indicate how they translate into computational effort.

The basic idea of the method of weighted residuals is to choose a trial function for the independent variable

$$u(x) = \sum_i \alpha_i B_{ki}(x) \tag{36}$$

where B_{ki} are the basis functions (here B-splines of order k) and α_i are coefficients to be determined. The derivatives that are given by

$$\begin{aligned} u'(x) &= \sum_i \alpha_i B'_{ki}(x) \\ u''(x) &= \sum_i \alpha_i B''_{ki}(x) \end{aligned} \tag{37}$$

are substituted into Eq. (35) to yield

$$L(u) = L\sum_i a_i B_{ki}(x) = \sum_i a_i L B_{ki}(x)$$

$$L(u) = \sum_i a_i \left\{ B''_{ki}(x) + b(x) B'_{ki}(x) + c(x) B_{ki}(x) \right\} = f(x) \tag{38}$$

We now require that the residual error in some sense vanishes over the domain, i.e.,

$$\sum_m \int_0^1 \left[Lu(x) - f(x) \right] W_m(x) dx = 0 \tag{39}$$

where W_m is a suitably chosen weighting function. The simplest method is to force the residual to vanish at each of the nodal points in the domain. This method of collocation is analogous to the interpolation problem which requires the approximating function and/or its derivatives to be equal to the true function at the nodal points.

In this case W_m is the Dirac delta function and we recover the system of equations

$$Lu(x_m) - f(x_m) = 0 \quad m = 1, \ldots, J+1$$

or

$$\sum_i a_i \left[B''_{ki}(x_j) + b(x_j) B'_{ki}(x_j) + c(x_j) B_{ki}(x_j) \right] = f_j \qquad j = 1, J+1 \tag{40}$$

For B-splines of order k there are J + 1 + k - 2 basis functions (dimension of the B-spline basis) so that in addition to the collocation relationships at each of the nodal points, including the boundary points, k - 2 supplemental conditions are required. For cubic B-spline (k = 4) the two boundary conditions are sufficient to close the system. Hence a linear system of equations of order J + 3 and bandwidth k - 1 = 3 (tridiagonal) is obtained. Note that the problem is no more difficult than the standard second order finite difference case, except that J + 3 equations are solved instead of at

most J + 1 equations. The overall accuracy of the method is $O(h^{k-2})$ which corresponds to the accuracy of spline approximation of the second derivative.

Once the α_i's are determined, u, u' and u'' can be obtained from Eq. (37). For higher order splines, e.g., k = 6 ($O(h^4)$ accuracy) an additional k - 2 = 4 conditions are required to close the system, which reduces to specifying two more conditions than are available, i.e., the two boundary conditions. If one has information concerning the behavior of the differential equation at the boundaries (values of derivatives there), then that can be used. However, in general this is not always the case, so it was decided to collocate at two points that were noncoincident with the knots. These locations were set at x = h/2 and at x = 1 - h/2 solely for ease of computation (modifying the matrix), but insight into the behavior of the differential equation could be used to choose the collocation points, i.e., regions of rapid change. It is important to note that these collocation points are located in a subregion between two knots. There is no need to add grid points (knots).

The bandwidth of the resulting matrix has now been increased to 5 in order to obtain an $O(h^4)$ solution. If the coefficients, b, c and f are not known at x = h/2 and x = h/2 then these values must be obtained by interpolation.

It can be shown that more accurate results can be obtained (one order better) if derivatives are introduced as unknowns, i.e.,

$$u = \sum_i \alpha_i B_{ki}$$
$$u' = \sum_i \beta_i B_{ki} \tag{41}$$

A system of two equations in α_i and β_i are thus obtained

$$L(u)_j = \sum_i \left\{ \beta_i \left[B'_{ki}(x_j) + b_j B_{ki}(x_j) \right] + \alpha_i c_j B_{ki}(x_j) \right\} = f_j \tag{42}$$

and

$$\sum_i \left\{ \beta_i B_{ki}(x_j) - \alpha_i B'_{ki}(x_j) \right\} = 0 \tag{43}$$

However, a 2 x 2 block matrix is now required to be inverted and boundary conditions on u' must also be supplied.

Galerkin Method

If we choose the test function W_m to be from the same family as the trial function, then the weighted residual becomes analogous to an orthogonality condition. This leads to the following relationship,

$$\sum_m \int_0^1 \left[\sum_i \alpha_i (B''_{ki} + b(x_j) B'_{ki} + c(x_j) B_{ki}) - f_j \right] B_{km} \, dx = 0 \qquad (44)$$

and we obtain (J + 1 + k - 2) equations which is exactly equal to the dimension of the basis. However, the bandwidth of the coefficient matrix for the α_i's is increased to 2k - 1 (7 for cubics and 11 for quintics). This result can be obtained by focusing our attention at the B-spline test function centered at knot j = m. It will contribute to the integral in the interval (m - k/2, m + k/2) where it is nonzero. However, the B-splines centered in the interval [m - (k - 1), m + (k - 1)] will also contribute to the integral. Hence the B_{ki}'s will span (k - 1) + (k - 1) + 1 = 2k - 1 knots and will contain 2k - 1 entries.

The Galerkin scheme can be shown to be $O(h^4)$ accurate for cubic B-splines. However, as compared to the method of collocation the bandwidth has increased from k - 1 to 2k - 1 so that the procedures would be much more costly. Furthermore, the Galerkin scheme requires four nontrivial integrals which necessitates an integration scheme consistent with the order of the method, e.g., for cubics a fourth order Simpson's rule. Since the B-splines are not orthogonal in the sense that integrals of products of basis functions are not zero, no simplifications exist.

Since in general one is not solving a problem that stems from a variational formulation, one would be justified in searching for test functions from a different family which could simplify the integrations and reduce the bandwidth of the resulting matrix. One such technique, the subdomain method (Ref. 45) or what Murphy (Ref. 46) terms the generalized Galerkin procedure, employs a unit step function as the test function. This formulation yields the following system of equations

$$\int_{x_j}^{x_{j+1}} \left[\sum_i a_i \left\{ B_{ki}'' + b(x_j) B_{ki}' + c(x_j) B_{ki} \right\} - f_j \right] dx = 0 \qquad j = 1,2,\ldots,J \quad (45)$$

Since only J equations are derived for the $(J + 1 + k - 2)a_i$s, k - 3 additional relationships are required. For cubics, this leaves, aside from the two boundary conditions, one as yet unspecified condition. Murphy (Ref. 37) and Bartlett and Kendall (Ref. 38) in their viscous flow solutions specified an additional derivative boundary condition at the edge of the viscous layer. Although their condition was exact mathematically for the problem they considered, for more general equations such conditions could overconstrain the solution. Hence it is felt that collocating at some point, i.e., one of the boundaries, would be helpful and would not deteriorate the order of accuracy of the solution (this point is further discussed in the section on numerical results).

The bandwidth of the resulting matrix has now been reduced to k as compared to 2k - 1 for the standard Galerkin scheme while still retaining $O(h^4)$ accuracy. However, the integration scheme (now over two adjacent grid points) needs to be appropriately handled in order to achieve the desired accuracy of the method. In order to employ Simpson's rule for cubic B-splines, values of the coefficients b, c and f are required at the intermediate points $x_{j + 1/2}$. If they are not known analytically, interpolations would be required which would increase the total computation time.

The advantages of the cubic B-spline generalized Galerkin procedure are its decreased bandwidth of four and its applicability to treat equations in conservative form. However, it has the disadvantage of being sensitive to the type of boundary condition used to satisfy the extraneous condition (cf., section on numerical results).

The advantages of the quintic B-spline collocation procedure are that boundary conditions are easy to apply and it possesses the lowest truncation error for a given grid spacing. However, it is less efficient than the generalized Galerkin method due to its larger bandwidth of five. At the expense of increasing computer memory requirements, quintic B-spline collocation can be made more efficient.

Both methods have the advantage over nonbasis function schemes of allowing

one to easily obtain derivatives of the solution vector as well as the solution at any point in the domain. Their main shortcoming is in the cell Reynolds number stability condition (cf., section on numerical results) which manifests itself as oscillations in the solution. In view of the above a blanket recommendation cannot be given. Only after an analysis of the problem under consideration is made and the goals of the solution (accuracy and efficiency) have been stated can one recommend either of the schemes.

Time was not available to program either Rubin's method or the Murphy-Kendall-Bartlett technique and thus a rigorous comparison with the B-spline formulation presented here cannot be given. However, it would appear that both B-spline procedures are competitive with these other spline variants, and they do have the advantage of reducing the order of the blocks for coupled systems of equations.

A comparison of the B-spline algorithms with the Q-R operator techniques is given in the section on numerical experiments.

Application of B-splines to 1-D Parabolic Equations

Consider the one-dimensional linear parabolic equation*

$$u_t = au_{xx} + bu_x + cu + d \tag{46}$$

with appropriate boundary conditions and initial conditions, where a, b, c, d could be functions of x and t. Employing a two point temporal difference scheme, we obtain,

$$\left(\frac{u_j^{n+1} - u_j^n}{\Delta t}\right) = a_j^{n+\beta}(\beta u_j''^{n+1} + (1-\beta)u_j''^{n}) + b_j^{n+\beta}(\beta u_j'^{n+1} + (1-\beta)u_j'^{n}) + c_j^{n+\beta}(\beta u_j^{n+1} + (1-\beta)u_j^{n}) + d_j^{n+\beta} \tag{47}$$

*If Eq. (46) were nonlinear, assume it was linearized by the method described in the appendix.

where $\beta = 1$ reduces to a backward difference and $\beta = 1/2$ to the Crank-Nicolson approximation.

The B-spline spatial approximation can be combined directly with a temporal discretization scheme to solve Eq. (46). Here collocation and the generalized Galerkin procedures are considered. Although no numerical experiments were run with these methods, they are presented here to indicate how they can be applied.

1. Collocation

Substituting Eqs. (36) and (37) into (47) and rearranging one obtains

$$\sum_i \alpha_i^{n+1} \left\{ B_{ki}(x_j) - \Delta t \beta \left[a_j^{n+\beta} B_{ki}''(x_j) + b_j^{n+\beta} B_{ki}'(x_j) + c_j^{n+\beta} B_{ki}(x_j) \right] \right\} - \Delta t\, d_j^{n+\beta} = RHS^n \qquad j = 1, 2, \ldots, J+1 \tag{48}$$

where

$$RHS^n = \Delta t (1-\beta) \left[a_j^{n+\beta} u_j''^n + b_j^{n+\beta} u_j'^n + c_j^{n+\beta} u_j^n \right] + u_j^n$$

is known from the previous time step and

$$a_j^{n+\beta} = \beta a_j^{n+1} + (1-\beta) a_j^n$$

and similarly for

$$b_j^{n+\beta},\ c_j^{n+\beta} \text{ and } d_j^{n+\beta}$$

2. Generalized Galerkin (cubic B-splines)

Integrating Eq. (48) over $(j, j+1)$ one obtains

$$\int_{x_j}^{x_{j+1}} \Big[\sum_i \alpha_i^{n+1} \Big\{ B_{ki}(x_j) - \Delta t \beta \big(a_j^{n+\beta} B_{ki}''(x_j) + b_j^{n+\beta} B_{ki}'(x_j) \big)$$

$$+ c_j^{n+\beta} B_{ki}(x_j) \Big\} - \Delta t\, d_j^{n+\beta} \Big] dx = \int_{x_j}^{x_{j+1}} (RHS^n)_j \, dx$$

By the use of Simpson's rule the integrals can be evaluated in terms of the function values at x_j, $x_{j+1/2}$ and x_{j+1}. If the values at the "half points", $a_{j+1/2}^{n+\beta}$, $b_{j+1/2}^{n+\beta}$, $c_{j+1/2}^{n+\beta}$ and $d_{j+1/2}^{n+\beta}$ are not known analytically they must be obtained by an interpolation routine.

RESULTS

Numerical Experiments

Spatial schemes can be analyzed by considering two point boundary value ordinary differential equations. The properties of these schemes, accuracy versus grid spacing, running time for a given accuracy, ease of programming, storage requirements and cell Reynolds number effects can easily be obtained without the encumbrance of temporal discretizations.

Here the following differential equation is considered

$$u_{xx} + bu_x = f \tag{49}$$

with appropriate boundary conditions specified.

The first type of problem investigated was the homogeneous constant coefficient case

$$\begin{aligned} u_{xx} + bu_x &= 0 \\ u(0) = 0 \,,\; u(1) &= 1 \end{aligned} \tag{50}$$

which has as its exact solution

$$u_{exact} = \frac{1 - e^{-bx}}{1 - e^{-b}} \tag{51}$$

The aim of these numerical experiments was to verify the convergence rates of the various methods and the effect of cell Reynolds number on obtaining physically meaningful solutions. The convergence rates were verified by considering the case of $b = 1$ for 10, 20 and 40 intervals, so that effects of cell Reynolds number, which ranged from .025 to .100 would be insignificant. In Table V these results are presented for the generalized OCI schemes (OCI-G1, (cf., Table III), OCI-G2, (cf., Table II)), cubic B-spline collocation, (CBS-COL), quintic B-spline collocation, (QBS-COL), and cubic B-spline generalized Galerkin, (CBS-GAL). All the schemes except for CBS-COL have fourth order convergence rates, with QBS-COL possessing the lowest truncation error.

The behavior of QBS-COL is due to the different orders of approximation of the derivatives, i.e., $O(h^4)$ for the second derivative and $O(h^5)$ for the first derivative, which reduces to $O(h^4)$ for the overall approximation to the spatial operator. It is also interesting to note that CBS-COL gives identical results as the standard second order centered finite difference method, (CFD), since the resulting matrices of the two methods are linearly related.

It has been observed (for instance in Ref. 47) that for the solution of the boundary layer equations, near the outer edge (at large values of the normal coordinate), where the flow is nearly uniform, oscillations and/or overshoots in the velocity profile may occur. This nonphysical behavior, which can be traced to the violation of a cell Reynolds number stability condition could deteriorate the entire solution. Therefore an understanding of the behavior of the spatial approximations with respect to cell Reynolds number is desirable. Considered here is the constant coefficient equation once more with b set equal to 80 for 10, 20 and 40 intervals which correspond to cell Reynolds numbers of 8, 4 and 2, respectively. In Table VI these results are presented. Solution profiles for Rc = 10 are shown in Table VII. The generalized OCI schemes give uniformly monotonic solutions (they are constructed to do just that), while the B-spline schemes give oscillatory solutions for cell Reynolds numbers approximately greater than 2.

The behavior of the OCI schemes in the range of Reynolds number 2 to 4 is a property of such an unrestricted (with Rc) scheme (cf., Ref. 29). As is true for all schemes, only in the limit as $h \to 0$ does one obtain the convergence rates predicted by the theory. Error estimates obtained by Berger et al., (Ref. 29) for the generalized OCI schemes indicate there are several overlapping regions that are dependent on the magnitude of Rc which varies as h^{-p} with h fixed. For p positive and large, $Rc \to 0$ and one recovers an $O(h^4)$ convergence rate, while for p negative and large, $Rc \to \infty$ and one obtains an $O(h^2)$ convergence rate. The transition, which is automatic, results in a second order upwind differencing formula when Rc is large. A more detailed discussion for the entire range of p values is presented in Ref. 29.

By way of comparison, the exponential scheme of Il'in (Ref. 31) has been shown to be uniformly first order accurate (Ref. 48) while the exponential box

scheme due to El-Mistikawy and Werle (Ref. 35) is uniformly second order accurate (Ref. 36). For the problem under consideration (constant coefficient) both exponential schemes give the exact solution as expected. The above discussion points out one of the major advantages of the Q-R operator format. Once a program has been written within such a framework, various schemes can be implemented easily. Even hybrid type schemes that are evaluated pointwise are allowed. For example, schemes can be chosen by considerations of accuracy, cell Reynolds number behavior, or running time.

For actual viscous flow problems, where moderate to high cell Reynolds numbers appear in uniform regions (near the outer edge of the viscous layer), the generalized OCI schemes work well in practice (Ref. 49).

It is important to note that the cubic B-spline generalized Galerkin procedure is sensitive to the type of boundary condition set at $x = 0$. When collocation at $x = 0$ was employed, the results were nonphysical while setting the first derivative to its exact value at that point gave results comparable to QBS-COL. The boundary condition was applied in a region of steep gradients so that the second order collocation approximation is not sufficient to prevent the erratic behavior. However, at the low cell Reynolds number range, the collocation boundary condition worked well as the results in Table V indicate.

The second case considered is the linearized Burgers equation

$$u_{xx} + \left[\frac{\alpha}{\nu} \tanh\left(\frac{\alpha x}{2\nu}\right)\right] u_x = 0 \tag{52}$$

which has the exact solution

$$u_{exact} = \alpha\left[1 - \tanh\left(\frac{\alpha x}{2\nu}\right)\right] \tag{53}$$

and

$$u(-\infty) \to 1\,,\ u(\infty) \to 0$$

The coefficient α was set to 1/2 while ν was varied; small ν corresponds to a shock near $x = 0$. Calculations were carried out in the regions

$-5 \leq x \leq 0$ and/or $0 \leq x \leq 5$, with the boundary conditions set to their exact function values. The Burgers equation mimics a true boundary layer in that regions of sharp gradients (near $x = 0$) corresponds to low cell Reynolds numbers while in the "uniform flow" region ($|x| \to \infty$) the cell Reynolds number reaches a maximum.

Results are presented in Table VIII for $\nu = 1/16$ and for mesh intervals 50, 100 and 150 for the following schemes; OCI-G1, OCI-G2, OCI-Swartz, CBS-GAL, QBS-COL and Allen's exponential scheme. The QBS-COL method again has the lowest truncation error for a given mesh distribution. However, the computation times (on a CDC 7600 machine) as presented in Table VIII indicate that in order to attain a given accuracy, OCI-G2 is the most efficient scheme for this problem while there is little difference between OCI-G1 and CBS-GAL. Although QBS-COL does not fare as well, its performance can be improved by storing the values of the basis functions, B_i, at the nodal points and not computing them as needed. In Table IX OCI-G1, OCI-G2, CBS-GAL, QBS-COL, the Allen exponential and the El-Mistikawy-Werle exponential schemes are compared for the case $\nu = 1/24$ as the cell Reynolds number is increased beyond 2. Again, the OCI schemes and the exponential schemes have monotonic behavior while the B-Spline techniques lead to oscillatory solutions.

The results indicate that with respect to accuracy there is little to choose from among the generalized OCI schemes and the higher order B-spline procedures. However, the B-spline procedures lead to larger banded matrices and have cell Reynolds number stability restrictions. Note that it is certainly not ruled out here that at some point in the future the B-splines could be modified to eliminate the oscillatory behavior at high cell Reynolds number. Although the B-spline schemes do have the advantage of being able to evaluate derivatives and treat derivative boundary conditions more easily than the OCI methods, at present they do not seem to be as versatile as OCI at least on equation systems that OCI can be applied to. It is therefore recommended that for the problem under consideration here for the spatial scheme, the family of Q-R operator schemes, in particular the generalized OCI schemes be adopted.

As a model problem for investigating the chosen spatial and temporal schemes, a coupled system of one-dimensional, nonlinear, parabolic equations

was constructed which exhibit typical boundary layer-like behavior, and have the form

$$u_t = u_{yy} + A_2 u_y + A_3 u + A_4 uw + A_5 w + A_6 \tag{54}$$

$$w_t = w_{yy} + B_2 w_y + B_3 w + B_4 uw + B_5 u + B_6 \tag{55}$$

where

$$A_2 = w, \quad A_3 = -\alpha^2, \quad A_4 = \alpha, \quad A_5 = -\alpha(\epsilon + \sin\omega t),$$

$$A_6 = \omega\cos\omega t(1 - e^{-\alpha y}) + \alpha^2(\epsilon + \sin\omega t)$$

and

$$B_2 = u, \quad B_3 = -\alpha^2, \quad B_4 = \alpha,$$

$$B_5 = -\alpha\delta\sin\omega t - \gamma e^{-\alpha y}(2y + \beta\sin\omega t),$$

$$B_6 = \omega\cos\omega t\left\{\gamma\beta y e^{-\alpha y} + \delta(1 - e^{-\alpha y})\right\} + \alpha^2\delta\sin\omega t$$
$$- 2\gamma e^{-\alpha y}\left\{1 - \alpha(2y + \beta\sin\omega t)\right\}$$

with boundary conditions

$$u(0,t) = w(0,t) = 0$$

$$u(\infty,t) = \epsilon + \sin\omega t \qquad \epsilon > 1 \qquad\qquad w(\infty,t) = \delta\sin\omega t \qquad \delta > 0$$

The initial conditions were chosen to be the exact solution values at the initial time.

The exact solutions are

$$u = (\epsilon + \sin\omega t)(1 - e^{-\alpha y}) \tag{56}$$

$$w = \gamma y(y + \beta\sin\omega t)e^{-\alpha y} + \delta\sin\omega t(1 - e^{-\alpha y}) \tag{57}$$

The exact solutions possess the following properties:

1. periodicity in time,
2. exponential decay in space,
3. nonlinearities and equation coupling,

and w exhibits a form of "flow reversal".

The Q-R operator spatial schemes were examined, including second order central finite differences, second order Il'in (exponential) and the fourth order generalized operator compact implicit method. As noted previously, all the above schemes require the inversion of a scalar tridiagonal matrix for a single equation in one unknown.

In the computer code, Eqs. (54) and (55) were solved simultaneously, resulting in a 2 x 2 block tridiagonal system. Both first order fully implicit and second order Crank-Nicolson discretizations were included in the program. Appendix B presents the appropriate linearizations and describes the implementation of the method.

The functions u and w were chosen, as previously mentioned, to mimic the velocity components of a time-dependent boundary layer flow. The "u velocity component" is always positive, with its maximum value (which is reached at the outer edge) varying between $\epsilon + 1$ and $\epsilon - 1$. The "w cross flow velocity component", as shown in Fig. 1, has a maximum value in the interior, and exhibits a cross flow reversal in that w changes sign through a time cycle. The numerical results to be discussed are for the case shown, i.e., $\alpha = 1$, $\beta = 4.0$, $\gamma = -2.0$, $\delta = 0.40$, and $\epsilon = 1.5$.

A comparison of Newton iteration with the second order noniterative time linearization procedure was made. The results shown in Figs. 2 and 3 verify the contention that the effort spent at iterating can be more effectively used in decreasing the step size, thus reducing both the temporal truncation and linearization errors. The spatial step size for the calculation shown in Figs. 2 and 3 was chosen (at h = .2), and the method employed was OCI-G1, so that the predominant error would be due to temporal effects, e.g., linearization and discretization. As an auxiliary benefit, time linearization requires one less time level of storage, and is easier to program (cf., Appendix B).

Figures 4 and 5 present a comparison of the generalized OCI scheme with second order central differences. The benefits in accuracy of the higher order OCI scheme is evident in these figures. In fact, for the J = 40 case, the

temporal truncation error, which can be shown to be of order 10^{-4}, dominated the spatial truncation error so that the OCI scheme did not attain its theoretical convergence rate.

CONCLUSIONS

In this report several temporal schemes have been investigated, consistently split block implicit, Rubin's predictor-corrector and Hopscotch. It has been shown that, in order to meet the requirements of three-dimensional unsteady boundary layer flows, the consistently split linearized block implicit scheme is the most versatile and efficient of the three, and was thus recommended.

Two approaches to spatial approximation were described, the Q-R operator formulation and the B-spline basis function technique. Results of numerical experiments indicate that both quintic B-spline collocation and generalized OCI schemes performed well and that the Q-R operator formulation, in particular the generalized OCI schemes, are at present particularly well suited for the problem of time-dependent boundary layers in regard to efficiency, cell Reynolds number stability restrictions, and flexibility.

The linearized block implicit temporal scheme, in conjunction with a generalized OCI spatial scheme, was employed to solve a system of two coupled nonlinear parabolic equations that exhibit "three-dimensional unsteady boundary layer" behavior. The results of numerical experiments indicate that the generalized OCI approach is viable and can be applied to viscous flow problems.

APPENDIX A

Linearization Technique

A number of techniques have been used for implicit solution of the following first-order nonlinear scalar equation in one dependent variable $\phi(x,t)$:

$$\partial\phi/\partial t = F(\phi)\,\partial G(\phi)/\partial x \tag{A1}$$

Special cases of Eq. (A1) include the conservation form if $F(\phi) = 1$, and quasi-linear flow if $G(\phi) = \phi$. Previous implicit methods for Eq. (A1) which employ nonlinear difference equations and also methods based on two-step predictor-corrector schemes are discussed by Ames (Ref. 50, p. 82) and von Rosenburg (Ref. 51), p. 56). One such method is to difference nonlinear terms directly at the implicit time level to obtain nonlinear implicit difference equations; these are then solved iteratively by a procedure such as Newton's method. Although otherwise attractive, there may be difficulty with convergence in the iterative solution of the nonlinear difference equations, and some efficiency is sacrificed by the need for iteration. An implicit predictor-corrector technique has been devised by Douglas and Jones (Ref. 52) which is applicable to the quasilinear case ($G = \phi$) of Eq. (A1). The first step of their procedure is to linearize the equation by evaluating the nonlinear coefficient as $F(\phi^n)$ and to predict values of $\phi^{n+1/2}$ using either the backward difference or the Crank-Nicolson scheme. Values for ϕ^{n+1} are then computed in a similar manner using $F(\phi^{n+1/2})$ and the Crank-Nicolson scheme. Gourlay and Morris (Ref. 53) have also proposed implicit predictor-corrector techniques which can be applied to Eq. (A1). In the conservative case ($F = 1$), their technique is to define $\hat{G}(\phi)$ by the relation $G(\phi) = \phi\hat{G}(\phi)$ when such a definition exists, and to evaluate $\hat{G}(\phi^{n+1})$ using values for ϕ^{n+1} computed by an explicit predictor scheme. With $\hat{G}$ thereby known at the implicit time level, the equation can be treated as linear and corrected values of ϕ^{n+1} are computed by the Crank-Nicolson scheme.

A technique is described here for deriving linear implicit difference approximations for nonlinear differential equations. The technique is based

on an expansion of nonlinear implicit terms about the solution at the known time level, t^n, and leads to a one-step, two-level scheme which, being linear in unknown (implicit) quantities, can be solved efficiently without iteration. This idea was applied by Richtmyer and Morton (Ref. 16, p. 203) to a scalar nonlinear diffusion equation. Here, the technique is developed for problems governed by ℓ nonlinear equations in ℓ dependent variables which are functions of time and space coordinates. The technique will be described for the three-dimensional, unsteady equations.

The solution domain is discretized by grid points having equal spacings in the computational coordinates, Δy^1, Δy^2 and Δy^3 in the y^1, y^2 and y^3 directions, respectively, and an arbitrary time step, Δt. The subscripts i, j, k and superscript n are grid point indices associated with y^1, y^2, y^3 and t, respectively, and thus $\phi^n_{i,j,k}$ denotes $\phi(y^1_i, y^2_j, y^3_k, t^n)$. It is assumed that the solution is known at the n level, t^n, and is desired at the (n+1) level, t^{n+1}. At the risk of an occasional ambiguity, one or more of the subscripts is frequently omitted, so that ϕ^n is equivalent to $\phi^n_{i,j,k}$.

The numerical method employed is quite general and is formally derived for systems of governing equations which have the following form:

$$\partial H(\phi)/\partial t = \mathcal{D}(\phi) + S(\phi) \tag{A2}$$

where ϕ is a column vector containing ℓ dependent variables, H and S are column vector functions of ϕ, and $\mathcal{D}$ is a column vector whose elements are spatial differential operators which may be multidimensional. The generality of Eq. (A2) allows the method to be developed concisely and permits various extensions and modifications (e.g., noncartesian coordinate systems, turbulence models) to be made more or less routinely. It should be emphasized, however, that the Jacobian $\partial H/\partial\phi$ must usually be nonsingular if the ADI techniques as applied to Eq. (A2) are to be valid. A necessary condition is that each dependent variable appear in one or more of the governing equations as a time derivative. An exception would occur if for instance, a variable having no time derivative also appeared in only one equation, so that this equation could be decoupled from the remaining equations and solved a posteriori by an alternate method.

The linearized difference approximation is derived from the following implicit time-difference replacement of Eq. (A2):

$$(H^{n+1}-H^{n})/\Delta t=\beta\left[\mathcal{D}(\phi^{n+1})+S^{n+1}\right]+(1-\beta)\left[\mathcal{D}(\phi^{n})+S^{n}\right] \tag{A3}$$

where, for example, $H^{n+1} \equiv H(\phi^{n+1})$. The form of $\mathcal{D}$ and the spatial differencing are as yet unspecified. A parameter $\beta(0 \leq \beta \leq 1)$ has been introduced so as to permit a variable centering of the scheme in time. Equation (A3) produces a backward difference formulation for $\beta = 1$ and a Crank-Nicolson formulation for $\beta = 1/2$.

The linearization is performed by a two-step process of expansion about the known time level t^n and subsequent approximation of the quantity $(\partial\phi/\partial t)^n\Delta t$, which arises from chain rule differentiation, by $(\phi^{n+1} - \phi^n)$. The result is

$$H^{n+1}=H^{n}+(\partial H/\partial\phi)^{n}(\phi^{n+1}-\phi^{n})+O(\Delta t)^{2} \tag{A4a}$$

$$S^{n+1}=S^{n}+(\partial S/\partial\phi)^{n}(\phi^{n+1}-\phi^{n})+O(\Delta t)^{2} \tag{A4b}$$

$$\mathcal{D}(\phi^{n+1})=\mathcal{D}(\phi^{n})+(\partial\mathcal{D}/\partial\phi)^{n}(\phi^{n+1}-\phi^{n})+O(\Delta t)^{2} \tag{A4c}$$

The matrices $\partial H/\partial\phi$ and $\partial S/\partial\phi$ are standard Jacobians whose elements are defined, for example, by $(\partial H/\partial\phi)_{qr} \equiv \partial H_q/\partial\phi_r$. The operator elements of the matrix $\partial\mathcal{D}/\partial\phi$ are similarly ordered, i.e., $(\partial\mathcal{D}/\partial\phi)_{qr} \equiv \partial\mathcal{D}_q/\partial\phi_r$; however, the intended meaning of the operator elements requires some clarification. For the q^{th} row, the operation $(\partial\mathcal{D}_q/\partial\phi)^n(\phi^{n+1} - \phi^n)$ is understood to mean that $\{\partial/\partial t\mathcal{D}_q[\phi(x,y,z,t)]\}^n\Delta t$ is computed and that all occurrences of $(\partial\phi_r/\partial t)^n$ arising from chain rule differentiation are replaced by $(\phi_r^{n+1} - \phi_r^n)/\Delta t$.

After linearization as in Eqs. (A4), Eq. (A3) becomes the following linear implicit time-differenced scheme:

$$(\partial H^n/\partial\phi)(\phi^{n+1}-\phi^n)/\Delta t = \mathcal{D}(\phi^n)+S^n + \beta\,(\partial\mathcal{D}/\partial\phi + \partial S^n/\partial\phi)(\phi^{n+1}-\phi^n) \quad \text{(A5)}$$

Although H^{n+1} is linearized to second order in Eq. (A4), the division by Δt in Eq. (A3) introduces an error term of order Δt. A technique for maintaining formal second-order accuracy in the presence of nonlinear time derivatives is discussed by McDonald and Briley (Ref. 7), however, a three-level scheme results. Second-order temporal accuracy can also be obtained (for $\beta = 1/2$) by a change in dependent variable to $\hat{\phi} \equiv H(\phi)$, provided this is convenient, since the nonlinear time derivative is then eliminated. The temporal accuracy is independent of the spatial accuracy.

On examination, it can be seen that Eq. (A5) is linear in the quantity $(\phi^{n+1} - \phi^n)$ and that all other quantities are either known or evaluated at the n level. Computationally, it is convenient to solve Eq. (A5) for $(\phi^{n+1} - \phi^n)$ rather than ϕ^{n+1}. This both simplifies Eq. (A5) and reduces roundoff errors, since it is presumably better to compute a small $O(\Delta t)$ change in an $O(1)$ quantity than the quantity itself. To simplify the notation, a new dependent variable ψ defined by

$$\psi \equiv \phi - \phi^n \quad \text{(A6)}$$

is introduced, and thus $\psi^{n+1} = \phi^{n+1} - \phi^n$, and $\psi^n = 0$. It is also convenient to rewrite Eq. (A5) in the following simplified form:

$$(A + \Delta t\,\mathcal{L})\psi^{n+1} = \Delta t\left[\mathcal{D}(\phi^n)+S^n\right] \quad \text{(A7a)}$$

where the following symbols have been introduced to simplify the notation:

$$A \equiv \partial H^n/\partial\phi - \beta\Delta t(\partial S^n/\partial\phi) \quad \text{(A7b)}$$

$$\mathcal{L} \equiv -\beta(\partial\mathcal{D}/\partial\phi) \quad \text{(A7c)}$$

It is noted that $\mathcal{L}(\psi)$ is a linear transformation and thus $\mathcal{L}(0) = 0$. Furthermore if $\mathcal{L}(\phi)$ is linear, then $\mathcal{L}(\psi) = -\beta\mathcal{D}(\psi)$.

Spatial differencing of Eq. (A7a) is accomplished simply by replacing derivative operators such as $\partial/\partial y^i$, $\partial^2/\partial y^i \partial y^i$ by corresponding finite difference operators, D_i, D_i^2. Henceforth, it is assumed that $\mathcal{D}$ and $\mathcal{L}$ have been discretized in this manner, unless otherwise noted.

Before proceeding, some general observations seem appropriate. The foregoing linearization technique assumes only Taylor expandability, an assumption already implicit in the use of a finite difference method. The governing equations and boundary conditions are addressed directly as a system of coupled nonlinear equations which collectively determine the solution. The approach thus seems more natural than that of making ad hoc linearization and decoupling approximations, as is often done in applying implicit schemes to coupled and/or nonlinear partial differential equations. With the present approach, it is not necessary to associate each governing equation and boundary condition with a particular dependent variable and then to identify various "nonlinear coefficients" and "coupling terms" which must then be treated by lagging, predictor-corrector techniques, or iteration. The Taylor expansion procedure is analogous to that used in the generalized Newton-Raphson or quasi-linearization methods for iterative solution of nonlinear systems by expansion about a known current guess at the solution (e.g., Bellman & Kalaba, Ref. 54). However, the concept of expanding about the previous time level apparently had not been employed to produce a noniterative implicit time-dependent scheme for coupled equations, wherein nonlinear terms are approximated to a level of accuracy commensurate with that of the time differencing. The linearization technique also permits the implicit treatment of coupled nonlinear boundary conditions, such as stagnation pressure and enthalpy at subsonic inlet boundaries, and in practice, this latter feature was found to be crucial to the stability of the overall method (Ref. 17).

APPENDIX B

Discretization of Model Equations

Herein we shall consider the system of model equations discussed in the section on numerical experiments namely,

$$u_t = u_{yy} + A_2 u_y + A_3 u + A_4 uw + A_5 w + A_6 \tag{B1}$$

where

$$A_2 = w, \qquad A_3 = -\alpha^2, \qquad A_4 = \alpha, \qquad A_5 = -\alpha(\epsilon + \sin\omega t),$$

$$A_6 = \omega\cos\omega t(1 - e^{-\alpha y}) + \alpha^2(\epsilon + \sin\omega t),$$

and

$$w_t = w_{yy} + B_2 w_y + B_3 w + B_4 uw + B_5 u + B_6 \tag{B2}$$

where

$$B_2 = u, \qquad B_3 = -\alpha^2, \qquad B_4 = \alpha,$$

$$B_5 = -\alpha\delta\sin\omega t - \gamma e^{-\alpha y}(2y + \beta\sin\omega t),$$

$$B_6 = \omega\cos\omega t\left\{\gamma\beta y e^{-\alpha y} + \delta(1 - e^{-\alpha y})\right\} + \alpha^2\delta\sin\omega t$$
$$- 2\gamma e^{-\alpha y}\left\{1 - \alpha(2y + \beta\sin\omega t)\right\}$$

Since nonlinear terms wu_y, uw_y and uw appear in Eqs. (B1) and (B2), linearization is required. Two types of linearization procedures are described, time linearization and Newton iteration (quasilinearization). The application of these methods within the Q-R operator framework is also demonstrated.

Time Linearization

The nonlinear terms uw and wu_y are linearized by the method described in Appendix A to yield

$$(wu_y)^{n+\beta} = w^n(\beta u_y^{n+1} + (1-\beta)u_y^n) + \beta u_y^n w^{n+1} - \beta w^n u_y^n \tag{B3}$$

$$(uw)^{n+\beta} = \beta w^n u^{n+1} + \beta u^n w^{n+1} + (1-2\beta)w^n u^n \tag{B4}$$

where $\beta = 1/2$ corresponds to a Crank-Nicolson scheme and $\beta = 1$ to a fully implicit method.

Substituting Eqs. (B3) and (B4) into (B1) we obtain

$$\begin{aligned}(u^{n+1}-u^n)/\Delta t &= \frac{Q^{-1}R}{\Delta y^2}\left[\beta u^{n+1} + (1-\beta)u^n\right] \\ &+ A_5\left[\beta w^n u^{n+1} + \beta u^n w^{n+1} + (1-2\beta)u^n w^n\right] \\ &+ A_6\left[\beta w^{n+1} + (1-\beta)w^n\right] + A_7 + \beta u_y^n w^{n+1} - \beta w^n u_y^n\end{aligned}$$

which, with some rearrangement, reduces to

$$\begin{aligned}&\left\{Q\left[\frac{1}{\lambda\beta} - \Delta y^2(A_5 w^n)\right] - R\right\}u^{n+1} - \Delta y^2 Q\left\{u_y^n + A_5 u^n + A_6\right\}w^{n+1} \\ &= \left\{Q\left[\frac{1}{\lambda\beta} + \Delta y^2\left(A_3 + \frac{(1-2\beta)}{\beta}A_5 w^n\right)\right] + \frac{(1-\beta)}{\beta}R\right\}u^n \\ &+ \Delta y^2 Q\left\{\frac{(1-\beta)}{\beta}A_6 w^n + \frac{A_7}{\beta} - w^n u_y^n\right\}\end{aligned} \tag{B5}$$

where

$$\lambda = \Delta t/\Delta y^2$$

and Q and R are evaluated at the n^{th} time level.

Similarly, Eq. (B2) reduces to

$$\left\{Q\left[\frac{1}{\lambda\beta} - \Delta y^2(B_5 u^n)\right] - R\right\} w^{n+1} - \Delta y^2 Q \left\{ w_y^n + B_5 w^n + B_6 \right\} u^{n+1}$$
$$= \left\{ Q\left[\frac{1}{\lambda\beta} + \Delta y^2\left(B_3 + \frac{(1-2\beta)}{\beta} B_5 u^n \right)\right] + \frac{(1-\beta)}{\beta} R \right\} w^n \qquad \text{(B6)}$$
$$+ \Delta y^2 Q \left\{ \frac{(1-\beta)}{\beta} B_6 u^n + \frac{B_7}{\beta} - u^n w_y^n \right\}$$

Equations (B5) and (B6) are solved as a coupled (2 x 2 block) system for u^{n+1} and w^{n+1}.

Newton Iteration

Alternatively one can linearize about a previous iteration instead of the previous time level, to yield for the nonlinear term $(wu_y)^{n+\beta}$ for example,

$$(wu_y)^{n+\beta} = \left[\beta^2 w^* + \beta(1-\beta) w^n \right] u_y^{n+1} + \left[\beta^2 u_y^* + \beta(1-\beta) u_y^n \right] w^n$$
$$+ \left[(1-\beta)^2 w^n u_y^n - \beta^2 w^* u_y^* \right] \qquad \text{(B7)}$$

Substituting Eq. (B7) in Eq. (B1) the quasilinear approximation becomes

$$(u^{n+1} - u^n)/\Delta t = \frac{(Q^{-1}R)}{\Delta y^2} \left[\beta u^{n+1} + (1-\beta) u^n \right]$$
$$+ A_5 \left[(\beta w^* + (1-\beta) w^n) \beta u^{n+1} + (\beta u^* + (1-\beta) u^n) \beta w^{n+1} \right.$$
$$\left. + (1-\beta)^2 u^n w^n - \beta^2 u^* w^* + A_6 \left[\beta w^{n+1} + (1-\beta) w^n \right] \right.$$
$$\left. + A_7 + \left[\beta u_y^* + (1-\beta) u_y^n \right] \beta w^{n+1} - \left[\beta u_y^* + (1-\beta) u_y^n \right] \beta w^* \right.$$

which after some manipulation reduces to

$$\begin{aligned}
&\left\{Q\left[\frac{1}{\lambda\beta} - \Delta y^2(A_5(\beta w^* + (1-\beta)w^n))\right] - R\right\} u^{n+1} \\
&- \Delta y^2 Q\left\{A_5(\beta u^* + (1-\beta)u^n) + A_6 + (\beta u_y^* + (1-\beta)u_y^n)\right\} w^{n+1} \\
&= \left\{Q\left[\frac{1}{\lambda\beta} + \Delta y^2 \frac{(1-\beta)}{\beta}(A_5(1-\beta)w^n)\right] + \frac{(1-\beta)}{\beta} R\right\} u^n \\
&+ \Delta y^2 Q\left\{\frac{A_7}{\beta} + \frac{(1-\beta)}{\beta} A_6 w^n - \beta A_5 u^* w^* - (\beta u_y^* + (1-\beta)u_y^n) w^*\right\}
\end{aligned} \tag{B8}$$

Here Q and R are assumed to be evaluated at $(n+\beta)$. A similar expression can be obtained for Eq. (B2), viz.,

$$\begin{aligned}
&\left\{Q\left[\frac{1}{\lambda\beta} - \Delta y^2(B_5(\beta u^* - (1-\beta)u^n))\right] - R\right\} w^{n+1} \\
&- \Delta y^2 Q\left\{B_5(\beta w^* + (1-\beta)w^n) + B_6 + (\beta w_y^* + (1-\beta)w_y^n)\right\} u^{n+1} \\
&= \left\{Q\left[\frac{1}{\lambda\beta} + \Delta y^2 \frac{(1-\beta)}{\beta}(B_5(1-\beta)u^n)\right] + \frac{(1-\beta)}{\beta} R\right\} w^n \\
&+ \Delta y^2 Q\left\{\frac{A_7}{\beta} + \frac{(1-\beta)}{\beta} B_6 u^n - \beta B_5 u^* w^* - (\beta w_y^* + (1-\beta)w_y^n) u^*\right\}
\end{aligned} \tag{B9}$$

Equations (B8) and (B9) are solved as a coupled system for u^* and w^*, the latest iterants. The coefficients and the Q and R operators are updated, and the system of equations is solved again. The process is continued until convergence is attained. The process involves:

1. updating the coefficients $A_2 - A_7$ and $B_2 - B_7$,
2. updating the Q and R operators,
3. solving the resulting 2 x 2 block system,
4. repeating steps 1-3 until convergence is attained.

Note that Eqs. (B5), (B6), (B8) and (B9) can easily be transformed to increment form. However, the results in this report were obtained for the form of the equations given above.

REFERENCES

1. Kendall, R. M., Bonnett, W. S., Nardo, C. T., and Abbett, M. J.: Computational Procedure for Three Dimensional Boundary Layers on Aircraft and Aerospace Vehicles. Proceedings AIAA 2nd Fluid Dynamics Conference, Hartford, 1975, pp. 113-121.

2. McCroskey, W. J.: Some Current Research in Unsteady Fluid Dynamics. J. Fluids Engineering, Vol. 99, March 1977, pp. 8-38.

3. Nash, J. F., and Patel, V. C.: Three Dimensional Turbulent Boundary Layers. Scientific and Business Consultants Inc., 1972.

4. Krause, E.: Comment on Solution of a Three-Dimensional Boundary Layer with Separation, AIAA J., Vol. 7, No. 3, March, 1969.

5. Lin, T. C., and Rubin, S. G.: Viscous Flow over a Cone at Moderate Incidence. Part 2. Supersonic Boundary Layer., J. Fluid Mech., Vol. 59, Part 3, 1973, pp. 593-620.

6. Briley, W. R.: Numerical Method for Predicting Three-Dimensional Steady Viscous Flow in Ducts. J. Comp. Physics, Vol. 14, No. 1, 1974.

7. McDonald, H., and Briley, W. R.: Three-Dimensional Supersonic Flow of a Viscous or Inviscid Gas. J. of Comp. Physics, Vol. 19, No. 2, 1975.

8. Blottner, F. G.: Investigation of Some Finite Difference Techniques for Solving the Boundary Layer Equations. Computer Methods in Applied Mechanics and Engineering. Vol. 6, 1975, pp. 1-30.

9. Thompson, B. G. J., MacDonald, A. G. J.: The Prediction of Boundary Layer Profile Development for Infinite Yawed Wings. Aero Res. Council, CP 1307.

10. Briley, W. R., and McDonald, H.: On the Structure and Use of Linearized Block ADI and Related Schemes. Scientific Research Associates, Inc., Report No. 78-3. To be published in the Journal of Computational Physics.

11. Peaceman, D. W., and Rachford, H. H.: The Numerical Solution of Parabolic and Elliptic Differential Equations. Soc. for Indust. Appl. Math., Vol. 3, 1955, pp. 28-41.

12. Douglas, J.: On the Numerical Integration of $u_{xx} + u_{yy} = u_t$ by Implicit Methods. Soc. for Indust. Appl. Math., Vol. 3, 1955, pp. 42-65.

13. Mitchell, A. R.: Computational Methods in Partial Differential Equations. Wiley, New York, 1969.

14. Yanenko, N. N.: The Method of Fractional Steps. Springer-Verlag, 1971.

15. Douglas, J. and Gunn, J. E.: A General Formulation of Alternating Direction Methods. Numerische Math., Vol. 6, 1964, p. 428.

16. Richtmyer, R. D., and Morton, K. W.: Difference Methods for Initial Value Problems. Second Edition. Interscience Publishers, New York, New York, 1967.

17. Briley, W. R., and McDonald, H.: Solution of the Multidimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method. J. Comp. Physics, Vol. 24, No. 4, August, 1977, p. 372.

18. Gibeling, H. J., McDonald, H., and Briley, W. R.: Development of a Three-Dimensional Combustor Flow Analysis, Volume I: Theoretical Studies. Technical Report AFAPL-TR-75-59, Vol. I, July 1975.

19. Gibeling, H. J., McDonald, H., and Briley, W. R.: Development of a Three-Dimensional Combustor Flow Analysis, Volume II: Theoretical Studies. Technical Report AFAPL-TR-75-59, Vol. II, October, 1976.

20. Briley, W. R.: Private communication.

21. Rubin, S. G., Lin, T. C.: A Numerical Method for Three Dimensional Viscous Flow: Application to the Hypersonic Leading Edge. J. of Comp. Physics, Vol. 9, 1972, p. 339.

22. Gourlay, A. R.: Hopscotch: A Fast Second-Order Partial Differential Equation Solver. J. Inst. Maths. Applics. No. 6, 1970, pp. 375-390.

23. Gourlay, A. R., and McGuire, G. R.: General Hopscotch Algorithm for the Numerical Solution of Partial Differential Equations. J. Inst. Maths. Applics. No. 7, 1971, pp. 216-227.

24. Gourlay, A. R.: Some Recent Methods for the Numerical Solution of Time-Dependent Partial Differential Equations. Proc. Royal Soc. Lond. A 323, 1971, pp. 219-233.

25. Smith, R. E., Jr., and Kidd, A.: Comparative Study of Two Numerical Techniques for the Solution of Viscous Flow in a Driven Cavity. NASA SP-378, 1975.

26. Buzbee, B. L., Golub, G. H., and Nielson, C. W.: On Direct Methods for Solving Poisson's Equations. SIAM J. Numerical Anal., Vol. 7, No. 4, Dec., 1970, pp. 627-656.

27. Greenberg, J. B.: Hopscotch Algorithms - A Study of Some Hybrid Finite-Difference Techniques. TAE Report No. 270, Technion-Israel Institute of Technology, Haifa, Israel, 1976.

28. Ciment, M., Leventhal, S. H., and Weinberg, B. C.: The Operator Compact Implicit Method for Parabolic Equations. J. Comp. Physics, 28 (1978), pp. 135-166.

29. Berger, A. E., Solomon, J. M., Ciment, M., Leventhal, S. H., and Weinberg, B. C.: Generalized Operator Compact Implicit Schemes for Boundary Layer Problems. Submitted for publication.

30. Allen, D. N., and Southwell, R. V.: Relaxation Methods Applied to Determine the Motion, in Two Dimensions, of a Viscous Fluid Past a Fixed Cylinder, Quart. J. Mech. Appl. Math., 8, 1955, pp. 129-145.

31. Il'in, A. M.: Differencing Scheme for a Differential Equation with a Small Parameter Affecting the Highest Derivative. Mat. Zametki, 6, 1969, pp. 237-248 - Math. Notes, 6, 1969, pp. 596-602.

32. McDonald, H.: A Novel Finite Difference Scheme Particularly Suited for Fluid Flow Problems Containing Discontinuities. Scientific Research Associates Report P76-1, August 1976.

33. Dennis, S. C. R.: Finite Differences Associated with Second-Order Differential Equations, Quart. J. Mech. Appl. Math., Vol. 23, 1960, pp. 487-507.

34. El-Mistikawy, T. M., and Werle, M. J.: Numerical Methods for Inviscid, Viscous Fluid Flows, Report No. AFL 77-9-34, University of Cincinnati, March 1977.

35. El-Mistikawy, T. M., and Werle, M. J.: Numerical Method for Boundary Layers with Blowing - the Exponential Box Scheme. AIAA J., 16, 1978, pp. 749-751.

36. Berger, A. E., Solomon, J. M., and Ciment, M.: An Analysis of a Uniformly Accurate Finite Difference Method for a Singular Perturbation Problem. Paper presented at Second International Conference on Computational Methods in Nonlinear Mechanics, March 26-30, Austin, Texas.

37. Prenter, P. M.: Splines and Variational Methods. John Wiley & Sons, New York, 1975.

38. Kendall, R. M., and Bartlett, E. P.: Nonsimilar Solution of the Multi-Component Laminar Boundary Layer by an Integral-Matrix Method. AIAA J. 6, 1968, p. 1089.

39. Murphy, J. D., Presley, L. L., and Rose, W. C.: On the Calculation of Supersonic Separating and Attaching Flows, NASA SP-347, 1975, pp. 151-175.

40. Ahlberg, J. H., Nilson, E. N., and Walsh, J. L.: The Theory of Splines and Their Applications. Academic Press, New York, 1967.

41. Rubin, S. G., and Khosla, P. K.: Polynomial Interpolation Methods for Viscous Flow Calculations, J. Comp. Physics, Vol. 14, No. 3, 1977, pp. 217-244.

42. Rubin, S. G., and Khosla, P. K.: Turbulent Boundary Layer With and Without Mass Injection. Computers and Fluids, Vol. 5, 1977, pp. 241-259.

43. DeBoor, C.: On Calculating with B-Splines. J. Approximation Theory, 6, 1972, pp. 50-62.

44. DeBoor, C.: Package for Calculating with B-Splines. SIAM J. Numer. Anal., Vol. 14, No. 3, June 1977.

45. Crandall, S. H.: Engineering Analysis. McGraw-Hill Book Company, 1956.

46. Murphy, J. D.: Application of the Generalized Galerkin Method to the Computation of Fluid Flows. Proc. AIAA Computational Fluid Dynamics Conference, 1973.

47. Weinberg, B. C., Leventhal, S. H., Ciment, M.: The Operator Compact Implicit Scheme for Viscous Flow Problems. AIAA Paper No. 77-638. Presented at 3rd AIAA Comp. Fluid Dynamics Conference, Albuquerque NM, 1977.

48. Kellogg, R. B., and Tsan, A.: Analysis of Some Difference Approximations for a Singular Perturbation Problem without Turning Points. To appear in Math. Comp.

49. Weinberg, B. C., and Glowacki, W.: Higher Order Methods for Three-Dimensional Boundary Layers. Paper in preparation.

50. Ames, W. F.: Numerical Methods for Partial Differential Equations. Barnes & Noble, Inc., New York, New York, 1969.

51. von Rosenberg, D. A.: Methods for the Numerical Solution of Partial Differential Equations. American Elsevier Publishing Co., Inc., New York, New York, 1969.

52. Douglas, J., and Jones, B. F.: On Predictor-Corrector Methods for Nonlinear Parabolic Differential Equations. Soc. for Indust. Appl. Math., Vol. 11, 1963, pp. 195-204.

53. Gourlay, A. R., and Morris, J. L.: Finite-Difference Methods for Nonlinear Hyperbolic Systems. Math. Comp., Vol. 22, 1968, pp. 28-39.

54. Bellman, R. E., and Kalaba, R. E.: Quasilinearization and Nonlinear Boundary-Value Problems. American Elsevier Publ. Co., Inc., New York, 1965.

TABLE I. - OPERATOR COEFFICIENTS FOR STANDARD OPERATOR COMPACT IMPLICIT SCHEME

$$q_j^- = 6 - 5\rho_j + 2\rho_{j+1} - \rho_j\rho_{j+1}$$

$$q_j^c = 60 + 16\rho_{j+1} - 16\rho_{j-1} - 4\rho_{j-1}\rho_{j+1}$$

$$q_j^+ = 6 + 5\rho_j - 2\rho_{j-1} - \rho_j\rho_{j-1}$$

$$r_j^- = q_j^-\left(1 - \frac{3}{2}\rho_{j-1}\right) + q_j^c\left(1 - \frac{1}{2}\rho_j\right) + q_j^+\left(1 + \frac{1}{2}\rho_{j+1}\right) + h^2 q_j^- c_{j-1}$$

$$r_j^c = -(r_j^+ + r_j^-) + h^2(q_j^- c_{j-1} + q_j^c c_j + q_j^+ c_{j+1})$$

$$r_j^+ = q_j^-\left(1 - \frac{1}{2}\rho_{j-1}\right) + q_j^c\left(1 + \frac{1}{2}\rho_j\right) + q_j^+\left(1 + \frac{3}{2}\rho_{j+1}\right) + h^2 q_j^+ c_{j+1}$$

where

$$\rho_j = hb_j$$

TABLE II. - OPERATOR COEFFICIENTS FOR GENERALIZED OPERATOR COMPACT IMPLICIT SCHEME OCI-G2

$$q_j^- = 6 + [p_1 - 3]Rc_j + [p_2]Rc_j^2$$

$$q_j^c = 60 + [10p_1]Rc_j + [p_3]Rc_j^2 + [\tau_{j+1}p_4]Rc_j^3$$

$$q_j^+ = 6 + [p_1 + 3]Rc_j + [p_1 + p_2]Rc_j^2 + [p_4]Rc_j^3$$

where

$$p_1 = 3 \ , \ p_2 = 0 \ , \ p_3 = \max[\pi_1, \pi_2]$$

$$\pi_1 = (\tau_{j+1} + \tau_{j-1})p_2 + \tau_{j+1}p_1 + \bar{\pi}_1 \qquad \pi_2 = 15 - 2p_2 + (\sigma_2 - 1)p_2 - 3(\tau_{j+1} + \sigma_2) + \bar{\pi}_2$$

$$\bar{\pi}_1 = \begin{cases} 0 & \sigma_1 \geq 0 \\ \frac{3}{8}\sigma_1^2(10 - \tau_{j+1} - \tau_{j-1}) & \sigma_1 \leq 0 \end{cases} \qquad \bar{\pi}_2 = \begin{cases} 0 & 2p_1 - \sigma_2 \geq 0 \\ (2p_1 - \sigma_2)^2/8 & 2p_1 - \sigma_2 < 0 \end{cases}$$

$$\sigma_1 = p_1 + 3 + \frac{(\tau_{j-1} - \tau_{j+1})}{10 - \tau_{j+1} - \tau_{j-1}} \qquad 2\sigma_2 = 3\tau_{j+1} - \tau_{j-1} + 10 + 2h\tau_{j-1}\left(\frac{c_{j-1}}{b_{j-1}}\right)$$

$$p_4 = \frac{1}{2}\left[1 + \tau_{j+1}\right]^{-1}\pi_3 \qquad \pi_3 = p_3 - \pi_1 + \bar{\pi}_1 + 2\tau_{j-1}\left(2 + h\frac{c_{j-1}}{b_{j-1}}\right)p_2$$

with h sufficiently small so that

$$10b_j - b_{j-1} - b_{j+1} > 0 \text{ and } hc_{j+1}/b_{j+1} < 2 \text{ for } j = 2, \cdots, J \text{ and } c_j \leq 0$$

where

$$\tau_{j-1} = b_{j-1}/b_j \ , \ \tau_{j+1} = b_{j+1}/b_j \text{ and } Rc_j = hb_j$$

r_j^-, r_j^c, r_j^+ given in TABLE I

TABLE III. - OPERATOR COEFFICIENTS FOR GENERALIZED OPERATOR COMPACT IMPLICIT SCHEME OCI-G1

$$q_j^{-,c,+} = Q_0^{-,c,+} + Q_1^{-,c,+} + Q_2^{-,c,+} + Q_3^{-,c,+}$$

$$Q_0^c = 60 \ , \ Q_0^- = Q_0^+ = 6$$

$$Q_1^c = 16(\rho^+ + \rho^-)$$

$$Q_1^- = \frac{Q_1^c}{10} - Q_0^c\left(\frac{\rho^c}{12} - \frac{\rho^+}{150} - \frac{\rho^-}{75}\right)$$

$$Q_1^+ = \frac{Q_1^c}{10} + Q_0^c\left(\frac{\rho^c}{12} - \frac{\rho^-}{150} - \frac{2\rho^+}{75}\right)$$

$$Q_2^- = \min(\mu_1, \mu_2)$$

$$Q_2^c = \omega_2 - 2Q_2^- - \frac{1}{6}(-2Q_1^c\rho^c + Q_1^+\rho^+ - 11Q_1^-\rho^-)$$

$$Q_2^+ = Q_2^- + \frac{1}{6}(Q_1^c\rho^c - 2Q_1^-\rho^- - 2Q_1^+\rho^+)$$

$$Q_3^- = 0$$

$$Q_3^c = \left\{\omega_3 + \frac{1}{2}\left[3Q_2^-\rho^- - Q_2^+\rho^+ + Q_2^c\rho^c\right]\right\} / (1 + \rho^+/\rho^c)$$

$$Q_3^+ = Q_3^c\rho^c/\rho^+$$

TABLE III. - CONTINUED

where

$$\omega_2 = \min(\mu_3, \mu_4)$$

$$\omega_3 = 0$$

$$\mu_1 = (Q_1^-)^2$$

$$\mu_2 = (Q_1^+)^2 - \frac{Q_0^c}{15}(Q_1^c\rho^c - 2Q_1^-\rho^- - 2Q_1^+\rho^+)$$

$$\mu_3 = \frac{5}{24Q_0^c}\left[\frac{6}{5}Q_1^c - Q_0^c\left(\frac{\rho^c}{2} + \frac{13}{100}\rho^- - \frac{3}{100}\rho^+\right)\right]^2$$

$$\mu_4 = \frac{1}{24Q_0^c\rho^c}\left|(Q_1^c\rho^c - Q_1^-\rho^- - Q_1^+\rho^+)\Big/\left[\rho^c - \frac{1}{10}(\rho^+ + \rho^-)\right] + 4(\rho^+ + \rho^- + 2\rho^c)Q_2 - Q_0^c\right|$$

$$+ \frac{1}{6}\left[(-20_1^c\rho^c + Q_1^+\rho^+ - 11Q_1^-\rho^-) + \frac{\rho^+}{\rho^c}(Q_1^c\rho^c - 2Q_1^-\rho^- - 2Q_1^+\rho^+)\right]$$

with h sufficiently small so that

$$10b_j - b_{j-1} - b_{j+1} > 0$$

r_j^-, r_j^c, r_j^+ coefficients are given in TABLE I

TABLE IV. - OPERATOR COEFFICIENTS FOR EL-MISTIKAWY WERLE EXPONENTIAL BOX SCHEME

$$r_j^- = \rho^- \exp(-\rho^-) \,/\, \left[1 - \exp(-\rho^-)\right]$$

$$r_j^+ = \rho^+ \,/\, \left[1 - \exp(-\rho^+)\right]$$

$$r_j^c = -(r_j^+ + r_j^-)$$

$$q_j^- = (1 - r_j^-)\,/\,(2\rho^-)$$

$$q_j^+ = (r_j^+ - 1)\,/\,(2\rho^+)$$

$$q_j^c = q_j^- + q_j^+$$

where

$$\rho^- = \frac{1}{2}(\rho_{j-1} + \rho_j) \quad , \quad \rho^+ = \frac{1}{2}(\rho_j + \rho_{j+1})$$

and

$$\rho_j = hb_j$$

TABLE V. - COMPARISON OF CONVERGENCE RATES FOR CONSTANT COEFFICIENT PROBLEM

Scheme	Jmax	R_c	Max error	Rate
OCI-G1	10	.100	-.576 E-07	
	20	.050	-.363 E-08	3.99
	40	.025	-.229 E-09	3.99
OCI-G2	10	.100	-.105 E-06	
	20	.050	-.687 E-08	3.93
	40	.025	-.438 E-09	3.91
CBS-COL	10	.100	.101 E-03	
	20	.050	.251 E-04	2.01
	40	.025	.629 E-05	2.00
QBS-COL	10	.100	-.163 E-07	
	20	.050	-.104 E-08	3.97
	40	.025	-.656 E-10	3.99
CBS-GAL	10	.100	-.296 E-05	
	20	.050	-.189 E-06	3.97
	40	.025	-.119 E-07	3.99

TABLE VI. - COMPARISON OF SPATIAL ERRORS FOR CONSTANT COEFFICIENT PROBLEM

Scheme	Jmax	Rc	Max error	Comment
CBS-GAL[1]	40	2	-.158 E+01	Oscillatory
QBS-COL	40	2	-.484 E-03	Oscillatory
OCI-G1	40	2	-.162 E-01	Monotone
OCI-G2	40	2	-.863 E-02	Monotone
CBS-GAL[1]	20	4	-.145 E+01	Oscillatory
QBS-COL	20	4	.612 E-01	Oscillatory
OCI-G1	20	4	-.586 E-01	Monotone
OCI-G2	20	4	-.127 E-01	Monotone
CBS-GAL	10	8	-.136 E+01	Oscillatory
QBS-COL	10	8	.288 E+02	Oscillatory
OCI-G1	10	8	-.561 E-01	Monotone
OCI-G2	10	8	-.425 E-02	Monotone

[1]Collocation at x=0 used.

TABLE VII. - SOLUTION PROFILES FOR CONSTANT COEFFICIENT PROBLEM - Rc = 10

X	Exact Solution	Computed Solution		
		(QBS-COL)	(CBS-GAL)[1]	(OCI-G2)
0.0	0.	0.	0.	0.
0.1	.999955 E+00	.374802 E+00	.118628 E+01	.997532 E+00
0.2	.999999 E+00	.304662 E+01	.965301 E+00	.999994 E+00
0.3	.100000 E+01	.311375 E-01	.100646 E+01	.999999 E+00
0.4	.100000 E+01	.251958 E+01	.998796 E+00	.100000 E+01
0.5	.100000 E+01	.476597 E+00	.100022 E+01	.100000 E+01
0.6	.100000 E+01	.215299 E+01	.999958 E+00	.100000 E+01
0.7	.100000 E+01	.778851 E+00	.100001 E+01	.100000 E+01
0.8	.100000 E+01	.189431 E+01	.999999 E+00	.100000 E+01
0.9	.100000 E+01	.107440 E+01	.100000 E+01	.100000 E+01
1.0	.100000 E+01	.100000 E+01	.100000 E+01	.100000 E+01

[1]Exact derivative at x = 0 set.

TABLE VIII. - COMPARISON OF RUNNING TIMES FOR BURGERS EQUATION - $\nu = 1/16$

Scheme	Jmax	Max error	Running time[1] (sec)
Allen Exponential	50	.336 E-02	.016
	100	.828 E-03	.028
	150	.367 E-03	.043
CBS-GAL	50	.260 E-03	.032
	100	.155 E-04	.042
	150	.297 E-05	.053
QBS-COL	50	.404 E-04	.078
	100	-.346 E-05	.118
	150	-.679 E-06	.157
OCI-G1	50	.242 E-03	.025
	100	.150 E-04	.044
	150	.298 E-05	.064
OCI-G2	50	-.696 E-04	.016
	100	-.656 E-05	.031
	150	-.148 E-05	.044

[1]CPU time on CDC 7600 - excludes input/output.

TABLE IX. - COMPARISON OF SPATIAL ERRORS FOR BURGERS EQUATION - $\nu = 1/24$

Scheme	Jmax	Rc max	Max error	Comment
Allen	25	2.4	.182 E-01	Monotone
El-Mistikawy Werle	25	2.4	-.213 E-01	Monotone
CBS-GAL	25	2.4	.242 E-02	Oscillations
QBS-COL	25	2.4	.243 E-02	Oscillations
OCI-G1	25	2.4	.438 E-02	Monotone
OCI-G2	25	2.4	.872 E-03	Monotone
Allen	50	1.2	-.814 E-02	Monotone
El-Mistikawy Werle	50	1.2	-.758 E-02	Monotone
CBS-GAL	50	1.2	-.800 E-03	Monotone
QBS-COL	50	1.2	.392 E-03	Monotone
OCI-G1	50	1.2	.123 E-02	Monotone
OCI-G2	50	1.2	-.279 E-03	Monotone

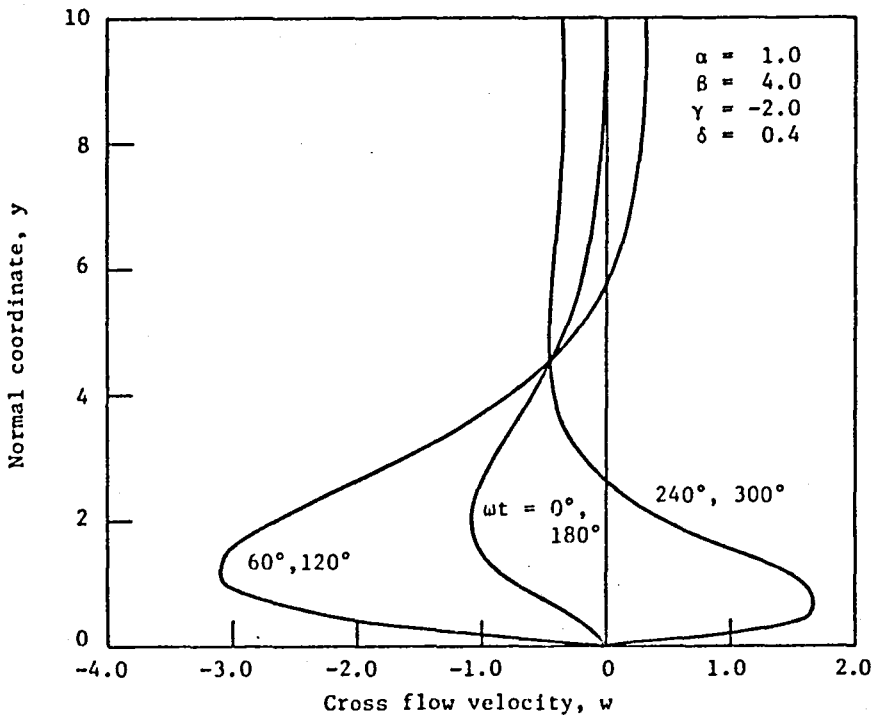

Figure 1. - Cross flow velocity profiles as a function of time.

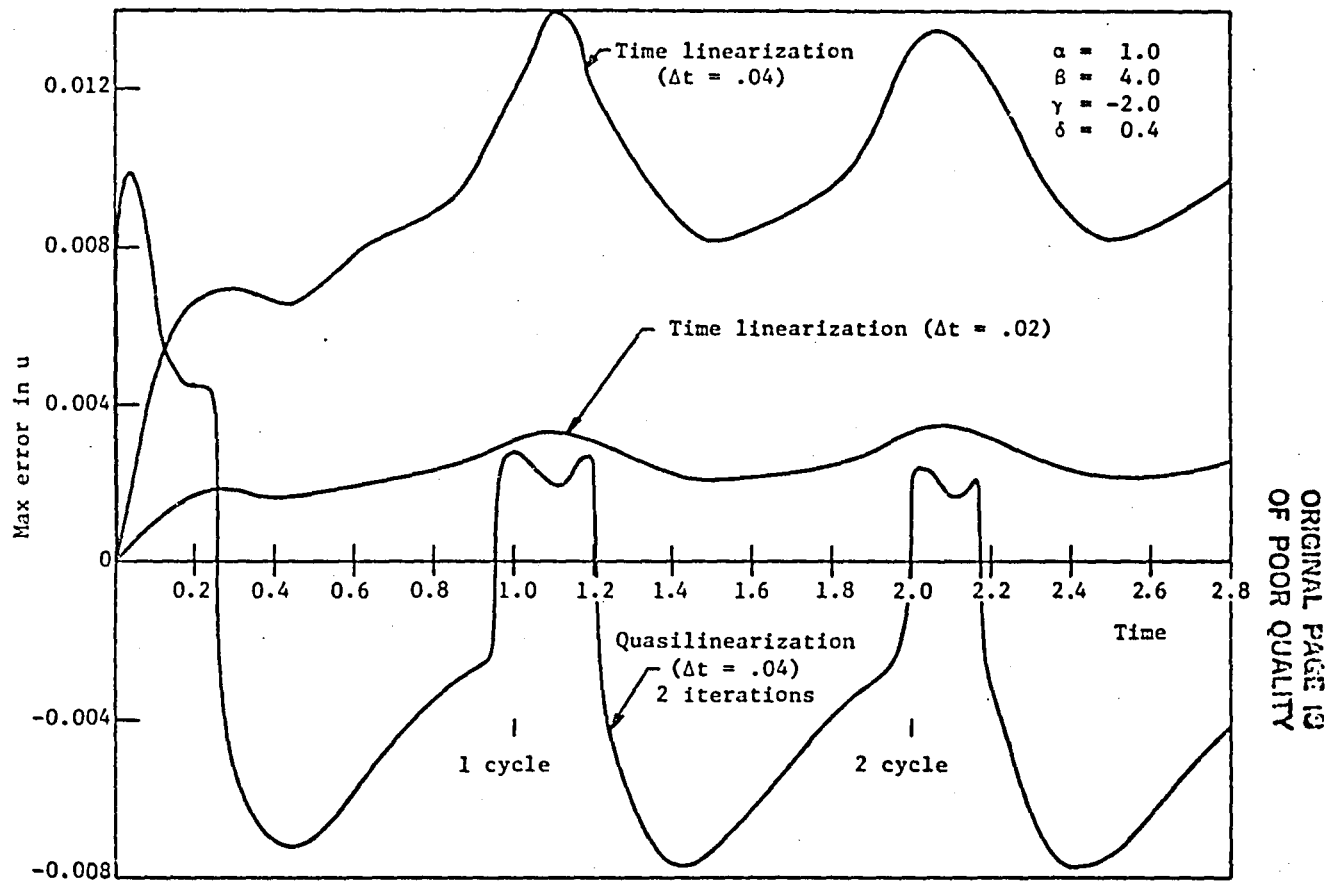

Figure 2. - Maximum error of u as a function of time.

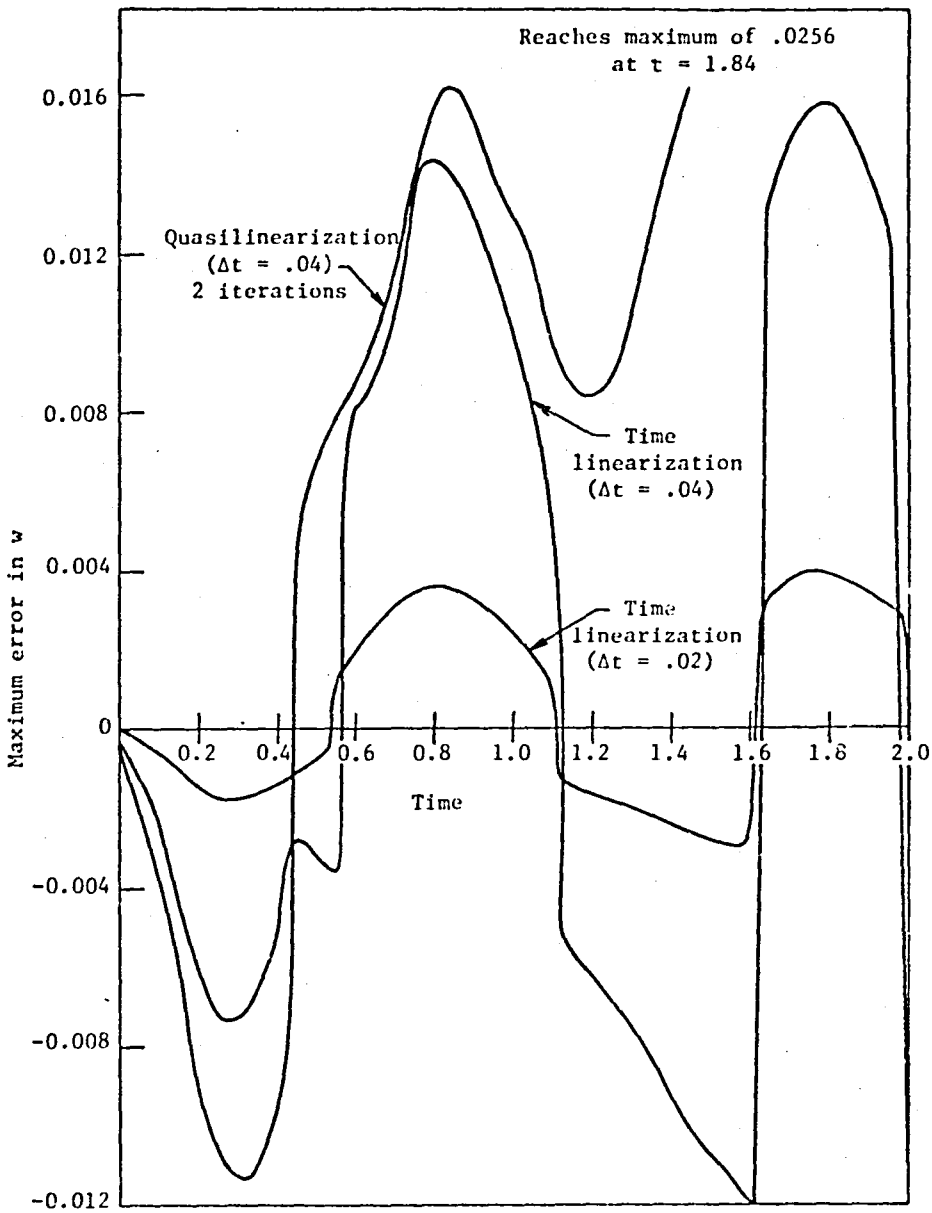

Figure 3. - Maximum error of w versus time.

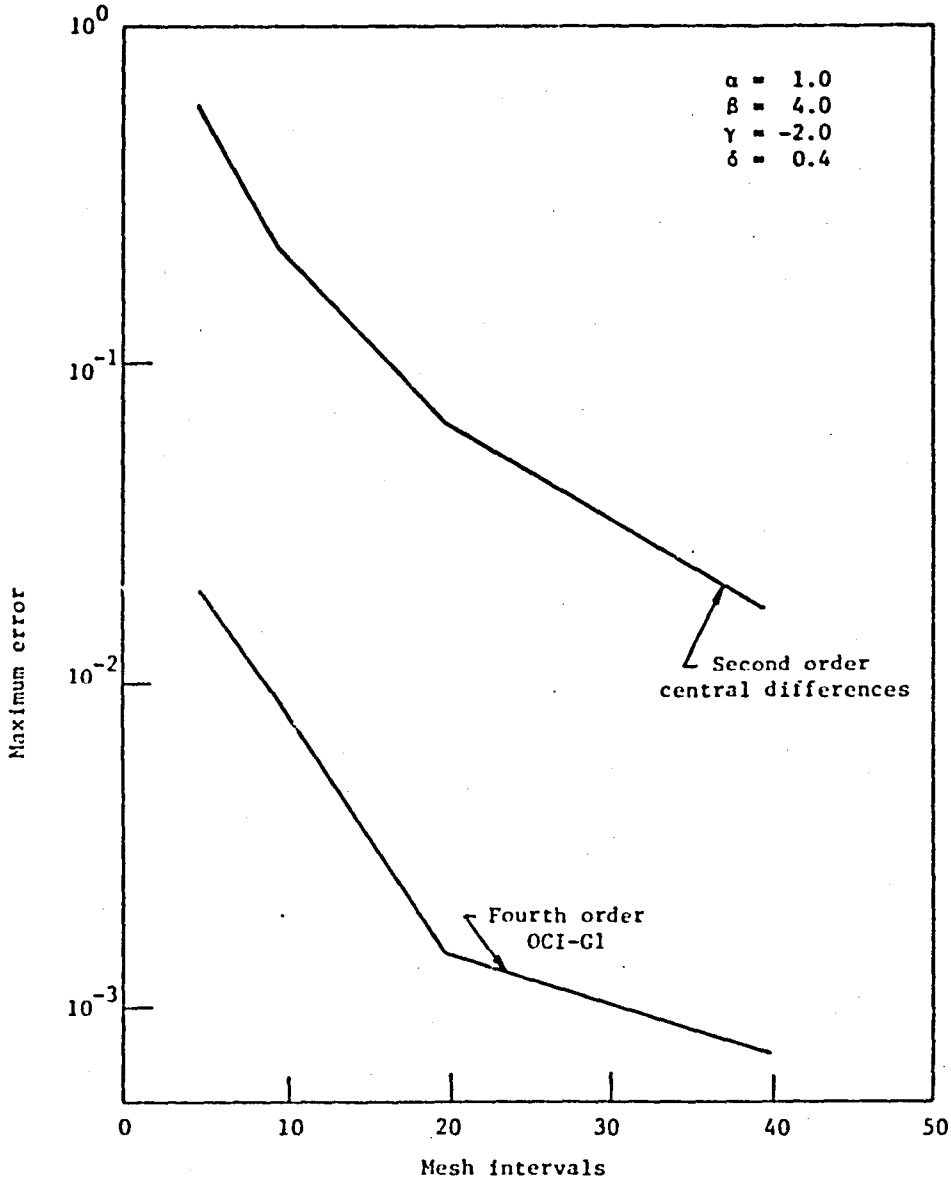

Figure 4. - Solution of w equation, maximum error vs. mesh intervals.

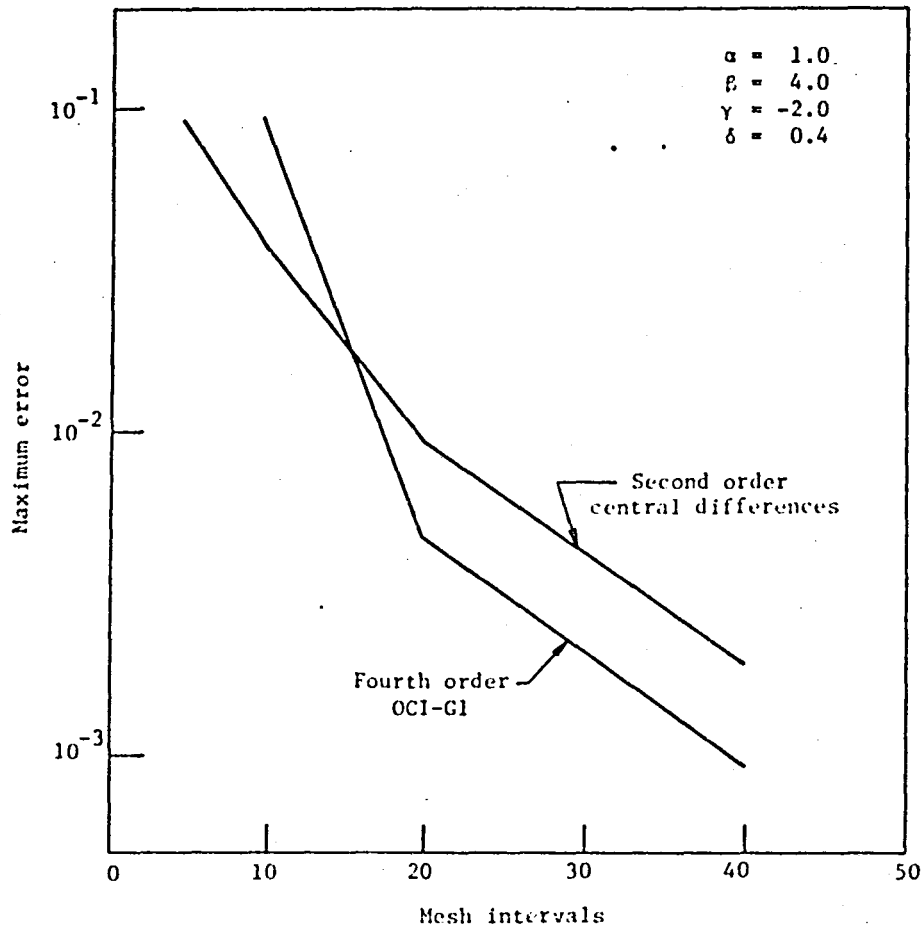

Figure 5. - Solution of u equation, maximum error vs. mesh intervals.

END

DATE

FILMED

SEP 10 1984

End of Document