

**NASA Contractor Report 172591**

**ICASE REPORT NO. 85-21**

# ICASE

**INTERMEDIATE BOUNDARY CONDITIONS FOR LOD, ADI,  
AND APPROXIMATE FACTORIZATION METHODS**

**Randall J. LeVeque**

**(NASA-CR-172591) INTERMEDIATE BOUNDARY  
CONDITIONS FOR LOD, ADI AND APPROXIMATE  
FACTORIZATION METHODS (NASA) 26 p  
HC A02/MF A01**

**NE5-29626**

**CSSL 12A**

**Unclas**

**G3/64 21555**

**Contract No. NAS1-17070**

**March 1985**

**INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING  
NASA Langley Research Center, Hampton, Virginia 23665**

**Operated by the Universities Space Research Association**

**NASA**

**National Aeronautics and  
Space Administration**

**Langley Research Center  
Hampton, Virginia 23665**



## Intermediate Boundary Conditions for LOD, ADI and Approximate Factorization Methods

Randall J. LeVeque  
Mathematics Department  
University of California, Los Angeles  
Los Angeles, CA 90024

*Abstract.* A general approach to determining the correct intermediate boundary conditions for dimensional splitting methods is presented and illustrated. The intermediate solution  $U^*$  is viewed as a second order accurate approximation to a modified equation. Deriving the modified equation and using the relationship between this equation and the original equation allows us to determine the correct boundary conditions for  $U^*$ . To illustrate this technique, we apply it to LOD and ADI methods for the heat equation in two and three space dimensions. The approximate factorization method is considered in slightly more generality.

*1980 Mathematics Subject Classification:* 65N05, 65N15

*Key words and phrases:* boundary conditions, ADI, approximate factorization, fractional step, splitting methods, modified equations

Research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-17070 while the author was in residence at ICASE, NASA Langley Research Center, Hampton, VA 23665.

March 20, 1985

## 1. Introduction

When solving time-dependent partial differential equations numerically, it is often desirable to use implicit methods for reasons of numerical stability. This is particularly true for parabolic equations where the time step restriction is severe for explicit methods. In more than one space dimension, however, implicit methods lead to linear systems of equations with complicated structure which cannot be solved efficiently by direct methods.

Consequently, a wide variety of dimensional splitting or fractional step methods have been used. In these methods, a single multidimensional implicit time step is replaced by a sequence of steps, each of which is implicit in only one coordinate direction. In that direction, the equations can be solved along one line of gridpoints at a time, giving a banded system of equations which can be easily solved. The reader is referred to [2], [4], [10], [13] for an introduction to many of the methods used in practice.

One difficulty which has sometimes plagued these methods and frequently caused confusion among users is the proper specification of boundary conditions for the intermediate solutions which arise between the individual steps of this sequence. The purpose of this paper is to show how the proper boundary conditions can frequently be determined quite simply and logically by considering the "modified equation" (or "model equation") corresponding to each finite difference approximation in the sequence.

We will illustrate this technique by considering several fractional step methods for the heat equation in two space dimensions: the locally one dimensional (LOD) method, the alternating-direction implicit (ADI) methods of Peaceman-Rachford[11] and Douglas-Gunn[3], and the approximate factorization (AF) method of Beam and Warming[1]. General boundary conditions involving normal derivatives are handled.

We have chosen a simple problem and well-known fractional step methods in order to illustrate the technique as clearly as possible. The same ideas used here can be applied to other methods and more complicated equations. The extension to methods involving more than two fractional steps is illustrated by considering the LOD method in three space dimensions in Section 6.

Although here we only discuss dimensional splitting methods, the same approach can be used to determine the correct intermediate boundary conditions for any additive splitting method. Other contexts

where these ideas have already been applied include the splitting of hyperbolic systems into subproblems with disparate wave speeds[8], [9], the splitting of convection-diffusion equations into hyperbolic and parabolic subproblems[8], and splittings of the incompressible Navier-Stokes equations[7].

For most of this paper we will consider the two-dimensional heat equation,

$$u_t = u_{xx} + u_{yy} = Lu \quad \text{in } \Omega = [0,1] \times [0,1] \quad (1.1)$$

with boundary conditions on  $\partial\Omega$ . Along the boundary  $x = x_b$  ( $x_b = 0$  or  $1$ ), for example, we will consider either the Dirichlet condition

$$u(x_b, y, t) = g(y, t) \quad (1.2a)$$

or the more general Robbins condition

$$\alpha(y, t)u(x_b, y, t) + \beta(y, t)u_x(x_b, y, t) = \gamma(y, t). \quad (1.2b)$$

A fractional step method for this problem typically has the form

$$Q_x U_{ij}^n = P_1 U_{ij}^n \quad (1.3a)$$

$$Q_y U_{ij}^{n+1/2} = P_2 U_{ij}^n. \quad (1.3b)$$

Here the  $P$ 's and  $Q$ 's are spatial difference operators, with  $Q_x$  and  $Q_y$  involving differences in only the  $x$ - and  $y$ -directions respectively.  $U_{ij}^n$  represents the numerical approximation to the solution  $u(x_i, y_j, t_n)$ . For simplicity, we take equal mesh widths  $h$  in the  $x$  and  $y$  directions, so  $x_i = ih$  and  $y_j = jh$  for  $i, j = 0, 1, \dots, N$ , where  $h = 1/N$ . The time step is denoted by  $k$ , so  $t_n = nk$ ,  $n = 0, 1, \dots$

The problem is to determine appropriate boundary conditions for the "intermediate solution"  $U_{ij}^n$  which arises in (1.3). We typically require boundary data  $U_{bj}^n$ ,  $b = 0, N$  for (1.3a). We may also require  $U_{ib}^n$ ,  $b = 0, N$ , for (1.3b) if  $P_2$  involves  $y$ -differences.

Typically, (1.3) gives a second order accurate approximation  $U^{n+1/2}$ , while the individual steps (1.3a) and (1.3b) are not second order accurate methods for the original equation. Hence the intermediate solution  $U^n$  is nonphysical and does not correspond to the true solution at any intermediate time. This makes it difficult to determine a priori the proper boundary conditions.

The approach we will take consists of determining the modified equation for (1.3a), the differential equation which *would* be solved to second order accuracy if we iterated with the scheme

$$Q_x U_{ij}^{m+1} = P_1 U_{ij}^m.$$

The modified equation is derived by replacing the difference operators by expansions of differential operators. This expansion can be truncated at an appropriate point to give a differential equation.

We denote the modified equation by

$$u_i^* = L^* u^* \quad (1.4a)$$

where  $L^*$  is a differential operator involving only spatial derivatives. Similarly, we can derive the modified equation for (1.3b):

$$u_i^{**} = L^{**} u^{**}. \quad (1.4b)$$

By truncating the expansions in each modified equation appropriately, we will have

$$L = L^* + L^{**}, \quad (1.5)$$

an additive splitting of the original operator.

There is an obvious splitting of  $L$  into one dimensional operators, namely

$$L^* = \partial_x^2, \quad L^{**} = \partial_y^2. \quad (1.6)$$

This is precisely the splitting used by the LOD method discussed in Section 2. The other methods, however, use more complicated splittings of the operator  $L$ .

Once the modified equation (1.4a) has been determined, we can view (1.3a) as taking a single time step on the equation (1.4a) with initial conditions

$$u^*(x, y, t_n) = u(x, y, t_n) \quad \forall x, y \in \Omega. \quad (1.7)$$

Here we assume  $U_{ij}^n = u(x_i, y_j, t_n)$ . If the solution to (1.4a), (1.7) is denoted by  $u^*(x, y, t)$  for  $t \geq t_n$ , then

$$U_{ij}^n = u^*(x_i, y_j, t_n + k) + O(k^3) \quad (1.8a)$$

since (1.3a) is second order accurate on (1.4a). One approach to specifying boundary conditions for  $U^*$  is to simply employ a one-sided, second order accurate, finite difference approximation to (1.4a). Then we are solving the same equation at the boundary as in the interior, but with a one-sided scheme instead of

the (presumably centered) scheme (1.3a).

The problem with this approach is that it does not make any use of the specified boundary conditions (1.2). We can often do better by determining the boundary behavior of  $u^*(x,y,t)$  in terms of the (given) boundary behavior of  $u$ . If we succeed in determining  $u^*(x,y,t_n+k)$  to  $O(k^3)$  for points along the boundary, this gives us the proper specification of  $U^*$ .

Actually, it usually suffices to have boundary data which are one order of accuracy lower than the interior scheme, i.e., we only need to insure that

$$U_{ij}^* = u^*(x_i, y_j, t_n + k) + O(k^2) \quad (1.8b)$$

at boundary points to maintain second order accuracy. For hyperbolic equations this is a result of Gustafsson[6]. For the heat equation it is a simple consequence of linearity and the maximum principle. However, in practice it is found that such boundary conditions can lead to a large increase in the error constant. Although the results are second order accurate as the mesh is refined, the errors on any particular mesh may be an order of magnitude larger than necessary. This can make a significant difference in the effort required to solve a multidimensional problem to a specified accuracy, and so we recommend using boundary conditions with  $O(k^3)$  accuracy whenever possible. Some examples of the resulting increase in accuracy will be seen in later sections.

To determine the behavior of  $u^*$  along the boundary, consider a typical point along one of the boundaries  $x = x_b$ . Expanding  $u^*(x_b, y, t_n + k)$  about time  $t_n$  gives

$$\begin{aligned} u^*(x_b, y, t_n + k) &= u^*(x_b, y, t_n) + k u_t^*(x_b, y, t_n) + \frac{1}{2} k^2 u_{tt}^*(x_b, y, t_n) + \dots \\ &= u^* + k L^* u^* + \frac{1}{2} k^2 M^* u^* + \dots \end{aligned} \quad (1.9)$$

We obtain  $M^*$  by differentiating (1.4a) with respect to  $t$  and replacing time derivatives of  $u^*$  on the right hand side by  $L^* u^*$ , so that  $M^*$  involves only *spatial* derivatives. For the linear equation considered here we have  $M^* = (L^*)^2$ , but the same process can be applied even in nonlinear problems.

It is important that the final expression in (1.9) involves only spatial derivatives of  $u^*$  at time  $t_n$ . (We make the convention throughout this paper that if a function occurs with no arguments on the right hand side of such an expression, the point  $(x_b, y, t_n)$  is assumed.) According to the initial conditions (1.7),  $u^* = u$  at time  $t_n$  and so  $L^* u^* = L^* u$  and similarly for any spatial operator (N.B. this is valid only at time  $t_n$ !). So (1.9) becomes

$$u^*(x_b, y, t_n + k) = u + kL^* u + \frac{1}{2}k^2 M^* u + \dots \quad (1.10)$$

This can frequently be manipulated to yield an expression in terms of the original boundary data for  $u$  by using the original equation (1.1) to replace  $L^*$  and  $M^*$  by "tangential" operators involving only  $t$ - and  $y$ -derivatives. The form of the final expression obtained depends on the particular case in question. Rather than continuing in such generality, we will demonstrate the process on particular examples in the following sections.

## 2. The locally one dimensional (LOD) method

The LOD method for the heat equation (1.1) takes the form

$$(I - \frac{1}{2}kD_x^2)U_{ij}^* = (I + \frac{1}{2}kD_y^2)U_{ij}^m \quad (2.1a)$$

$$(I - \frac{1}{2}kD_y^2)U_{ij}^{m+1} = (I + \frac{1}{2}kD_x^2)U_{ij}^*. \quad (2.1b)$$

Here we use

$$D_x^2 U_{ij} = h^{-2}(U_{i+1,j} - 2U_{ij} + u_{i-1,j}), \quad D_y^2 U_{ij} = h^{-2}(U_{i,j+1} - 2U_{ij} + u_{i,j-1}).$$

Since (2.1a) is the second order accurate Crank-Nicolson method applied to

$$u_t^* = u_{xx}^*, \quad (2.2)$$

we can take this as the modified equation for (2.1a). The second step (2.1b) is the same in the  $y$ -direction, and so the LOD method corresponds to the splitting (1.6).

Note that (2.1a) can be applied at  $j = b = 0, N$  in order to obtain the values  $U_{ib}^*$  needed in (2.1b). However, we still need to specify boundary conditions for  $U_{kj}^*, j = 0, N$ . Consider a typical point  $(x_b, y)$

and expand  $u^*(x_b, y, t_n + k)$  about  $(x_b, y, t_n)$  to obtain

$$\begin{aligned} u^*(x_b, y, t_n + k) &= u^*(x_b, y, t_n) + ku_t^*(x_b, y, t_n) + \frac{1}{2}k^2u_{tt}^*(x_b, y, t_n) + \cdots \\ &= u^* + ku_{xt}^* + \frac{1}{2}k^2u_{xxxx}^* + \cdots \\ &= u + ku_{xt} + \frac{1}{2}k^2u_{xxxx} + \cdots \end{aligned} \quad (2.3)$$

which corresponds to (1.10). We now use the original equation (1.1) to solve for

$$u_{xx} = u_t - u_{yy}$$

and

$$u_{xxxx} = u_{tt} - 2u_{yyt} + u_{yyyy}.$$

This allows us to rewrite the boundary data (2.3) in terms of  $t$ - and  $y$ -derivatives of  $u$  along the boundary

$x = x_b$ :

$$\begin{aligned} u^*(x_b, y, t_n + k) &= u + k(u_t - u_{yy}) + \frac{1}{2}k^2(u_{tt} - 2u_{yyt} + u_{yyyy}) + \cdots \\ &= u(x_b, y, t_n + k) - ku_{yy} + \frac{1}{2}k^2(u_{yyyy} - 2u_{yyt}) + \cdots \\ &= [I - k\partial_y^2 + \frac{1}{2}k^2\partial_y^4 + \cdots] u(x_b, y, t_n + k). \end{aligned} \quad (2.4)$$

This can also be obtained by integrating (1.4b) backwards in time from  $U^{n+1}$ ; see Section 6 for details.

If the original problem specified Dirichlet boundary conditions,  $u(x_b, y, t) = g(y, t)$ , (2.4) immediately provides the appropriate boundary conditions for  $U^*$ . Based on (1.8a), we can take

$$U_{bj}^* = g(y_j, t_n + k) - kg_{yy}(y_j, t_n + k) + \frac{1}{2}k^2g_{yyyy}(y_j, t_n + k). \quad (2.5)$$

With the Robbins condition (2.2b), appropriate boundary conditions are still easy to derive if  $\alpha$  and  $\beta$  are constant, so (2.2b) is

$$\alpha u(x_b, y, t) + \beta u_x(x_b, y, t) = \gamma(y, t). \quad (2.6)$$

Differentiating (2.4) with respect to  $x$  and taking the linear combination  $\alpha u^* + \beta u_x^*$  gives the boundary condition

$$\alpha u_x^*(x_b, y, t_n + k) + \beta u_x^*(x_b, y, t_n + k) = \gamma(y, t_n + k) - k\gamma_{yy}(y, t_n + k) + \frac{1}{2}k^2\gamma_{yyy}(y, t_n + k). \quad (2.7)$$

If  $\alpha$  and  $\beta$  are not constant then the situation is more complicated since, for example,  $\alpha u_{yy} + \beta u_{yy} \neq \gamma_{yy}$ . Based on the second line of (2.4), we can still obtain a condition of the form

$$\alpha u^* + \beta u_x^* = \gamma(y, t_n + k) - k(\alpha u_{yy} + \beta u_{yy}) + \frac{1}{2}k^2(\alpha u_{yyy} + \beta u_{yyy} - 2\alpha u_{yy} - 2\beta u_{yy}), \quad (2.8)$$

but now the derivatives of  $u(x_b, y, t_n)$  on the right hand side must be replaced by finite difference approximations based on  $U_{ij}^n$ . Unless this is done carefully, numerical instabilities can arise. This is discussed in more detail in Section 4.

To demonstrate the validity of the boundary conditions derived here, we present the results of two numerical experiments with Dirichlet conditions. We first consider the problem (1.1) with exact solution

$$u(x, y, t) = \sin(x/2 + y)e^{-1.25t} \quad (2.9)$$

and specify boundary conditions obtained by evaluating (2.9) along the boundaries, e.g., along  $x = 0$ ,

$$g(y, t) = \sin(y)e^{-1.25t}. \quad (2.10)$$

We compare the results obtained with three different choices of boundary conditions  $U_{bj}^*$ . The first of these is simply

$$U_{bj}^* = g(y_j, t_n + k/2). \quad (2.11)$$

This has been recommended by some authors, but comparing this to (2.5) shows that it differs by  $O(k)$ . We thus expect this boundary condition to destroy the second order accuracy. This is confirmed by the results of Table 2.1. Here we have displayed the errors at time  $t = 0.75$  measured in both the  $L_x$ - and  $L_2$ -norms. We have also computed the  $L_2$ -rate of convergence, obtained by comparing the errors with different values of  $k$  (we always take  $k/h$  fixed as the mesh is refined and  $h = 1/N$ ). The  $L_2$ -rate is defined by  $\log(e_2/e_1) / \log(k_2/k_1)$ , where  $e_1$  and  $e_2$  are the  $L_2$ -errors obtained with time steps  $k_1$  and  $k_2$  respectively.

The second boundary condition used is

$$U_{bj}^* = g(y_j, t_n + k) - kg_{yy}(y_j, t_n + k), \quad (2.12)$$

which was derived by Dwoyer and Thames[4] using the "method of undetermined functions." In this approach, one inserts undetermined constants in place of the unknown boundary values and combines the

stages into a one-step method. Requiring that the resulting method be second order accurate yields the boundary condition (2.12). Since (2.12) corresponds to the first two terms in (2.5), according to our comments concerning equation (1.8b) this should be sufficient to restore second order accuracy. This is also confirmed by Table 2.1. Finally, we have used (2.5) itself. Again the method is second order accurate, but the errors are reduced by an order of magnitude over those seen with (2.12).

In our second experiment, we investigate the effect of various boundary conditions on a steady state solution obtained with the LOD method. Presumably the choice of boundary conditions will have little effect on the rate of convergence to steady state, but may have a significant impact on the accuracy of the resulting solution. To investigate this latter effect, we have taken initial data equal to an exact steady state solution to (1.1), namely,

$$u(x,y) = \cosh(y-.5) \sin x + \cosh(x-.5) \sin y. \quad (2.13)$$

Again, Dirichlet boundary conditions were obtained by evaluating (2.13) along the boundaries. Within a few iterations, the LOD method converges to a numerical steady state. Table 2.2 shows the error in the numerical solution for each of the boundary conditions considered previously. Again, (2.11) gives only first order accuracy while (2.12) restores second order accuracy. In this case, the use of (2.5) improves the error constant by a factor of at least 20 over (2.12).

### 3. The Peaceman-Rachford and Douglas-Gunn ADI Methods

The ADI method introduced by Peaceman and Rachford[11] has the form

$$(I - \frac{1}{2}kD_x^2)U_{ij}^* = (I + \frac{1}{2}kD_y^2)U_{ij}^n \quad (3.1a)$$

$$(I - \frac{1}{2}kD_y^2)U_{ij}^{n+1} = (I + \frac{1}{2}kD_x^2)U_{ij}^* \quad (3.1b)$$

Here the splitting is no longer simply (1.6). The ADI method is usually viewed in the following way: Each step (3.1a) and (3.1b) is a *first order* accurate method for the original equation (1.1) on a time step of length  $k/2$  which combine to give second order accuracy over a step of length  $k$ . Because (3.1a) is con-

sistent with the original equation, it is tempting to specify

$$U_{bj}^* = g(y_j, t_n + k/2), \quad (3.2)$$

for example, in the case of Dirichlet conditions. However, it has long been known that this causes a loss of accuracy since this boundary condition does not contain the  $O(k^2)$  error present in the interior which is required to cancel out the  $O(k^2)$  error in the second step. Fairweather and Mitchell[5] determined the correct boundary conditions by the method of undetermined functions, obtaining

$$U_{bj}^* = \frac{1}{2}(I - \frac{1}{2}kD_y^2)g_j^{n+1} + \frac{1}{2}(I + \frac{1}{2}kD_y^2)g_j^n \quad (3.3)$$

where  $g_j^n = g(y_j, t_n)$ . The same result can be obtained by the approach advocated here.

Rather than viewing (3.1a) as a first order scheme for (1.1) with time step  $k/2$ , we derive the modified equation for which (3.1a) is a second order approximation with time step  $k$ . We have

$$(I - \frac{1}{2}kD_y^2)u^*(x, y, t_n + k) = (I + \frac{1}{2}kD_y^2)u^*(x, y, t_n).$$

Expanding  $u^*(x, y, t_n + k)$  about  $u^*(x, y, t_n)$  and using, for example,

$$D_y^2 u^* = \partial_y^2 u^* + O(k^2)$$

we obtain

$$u^* + k(u_t^* - \frac{1}{2}u_{xx}^*) + \frac{1}{2}k^2(u_{tt}^* - u_{txx}^*) + O(k^3) = u^* + \frac{1}{2}ku_{yy}^* + O(k^3)$$

or, solving for  $u_t^*$ :

$$u_t^* = \frac{1}{2}(u_{xx}^* + u_{yy}^*) - \frac{1}{2}k(u_{tt}^* - u_{txx}^*) + O(k^2). \quad (3.4)$$

We can obtain expressions for  $u_{tt}^*$  and  $u_{txx}^*$  by differentiating this. Plugging the resulting expressions back into (3.4) gives

$$u_t^* = \frac{1}{2}(u_{xx}^* + u_{yy}^*) + \frac{1}{8}k(u_{xxxx}^* - u_{yyyy}^*) \quad (3.5)$$

plus terms which are  $O(k^2)$ . Hence (3.1a) gives a second order accurate approximation to (3.5) on a time step of length  $k$  and we can take

$$L^* = \frac{1}{2}(\partial_x^2 + \partial_y^2) + \frac{1}{8}k(\partial_x^4 - \partial_y^4). \quad (3.6a)$$

Since (3.1b) is the same as (3.1a) but with  $x$  and  $y$  interchanged, deriving the modified equation for (3.1b) gives

$$L^{**} = \frac{1}{2}(\partial_x^2 + \partial_y^2) + \frac{1}{8}k(\partial_y^4 - \partial_x^4). \quad (3.6b)$$

Note that (1.5) is satisfied.

Using the equation (3.5), we can proceed as before to determine the boundary data. Since

$$L^* = \frac{1}{2}L + \frac{1}{8}k(\partial_x^4 - \partial_y^4), \quad M^* = (L^*)^2 = \frac{1}{4}L^2 + O(k),$$

we now obtain, by (1.10),

$$\begin{aligned} u^*(x_b, y, t_n + k) &= u(x_b, y, t_n) + kL^*u + \frac{1}{2}k^2M^*u + \dots \\ &= u + \frac{1}{2}kLu + \frac{1}{8}k(L^* - \frac{1}{2}L)u + \frac{1}{8}k^2L^2u + \dots \\ &= u(x_b, y, t_n + k/2) + \frac{1}{8}k^2(u_{xxxx} - u_{yyyy}) + \dots \end{aligned} \quad (3.7)$$

Moreover, we can manipulate (1.1) to obtain

$$u_{xxxx} - u_{yyyy} = u_{xx} - 2u_{yy}$$

so that this can be reexpressed in terms of  $t$ - and  $y$ - derivatives along the boundary. Hence, with Dirichlet boundary conditions, we can take

$$U_{bj}^* = g(y_j, t_n + k/2) + \frac{1}{8}k^2(g_{yy}(y_j, t_n) - 2g_{yy}(y_j, t_n)). \quad (3.8)$$

This gives the required  $O(k^2)$  correction to (3.2). Instead of using derivatives of the function  $g$ , one could approximate (3.8) by any finite difference expression correct to  $O(k^3)$ . It is easy to verify that Fairweather and Mitchell's boundary condition (3.3) is one such approximation.

The approach taken here easily extends to more general boundary conditions. For example, corresponding to the boundary conditions (2.6) with  $\alpha$  and  $\beta$  constant, we obtain

$$\alpha u^*(x_b, y, t_n + k) + \beta u_x^*(x_b, y, t_n + k) = \gamma(y, t_n + k/2) + \frac{1}{8}k^2(\gamma_{yy}(y, t_n) - 2\gamma_{yy}(y, t_n)).$$

Based on the formulas used above, it is also easy to determine the intermediate boundary conditions for the ADI method proposed by Douglas and Gunn[3]:

$$(I - kD_x^2)U_{ij}^* = (I + kD_y^2)U_{ij}^n \quad (3.9a)$$

$$(I - kD_y^2)U_{ij}^{n+1} = U_{ij}^n + kD_x^2 U_{ij}^*. \quad (3.9b)$$

Except for the factor 1/2, (3.9a) is identical to (3.1a) and the modified operator is easily determined to be

$$L^* = (\partial_x^2 + \partial_y^2) + \frac{1}{2}k(\partial_x^4 - \partial_y^4). \quad (3.10)$$

The intermediate boundary conditions in the Dirichlet case are found to be

$$U_{ij}^* = g(x_j, t_n + k) + \frac{1}{2}k^2(g_{xx}(x_j, t_n) - 2g_{xy}(x_j, t_n)). \quad (3.11)$$

Although it is not necessary to compute the operator  $L^{**}$ , it is interesting to do so since the equation (3.9b) has a feature not seen before; it involves  $U^n$  as well as  $U^*$  and  $U^{n+1}$ . Because of this it appears senseless to discuss the modified equation for (3.9b) since we cannot apply it in isolation. However, if we multiply (3.9b) by  $(I + kD_y^2)$  and use (3.9a), we can eliminate  $U^n$  and obtain

$$(I + kD_y^2)(I - kD_x^2)U_{ij}^{n+1} - [(I - kD_x^2) + k(I + kD_y^2)D_x^2] U_{ij}^*. \quad (3.12)$$

Note that this is a bit of a twist on the usual analysis in which one eliminates  $U^*$  from (3.9) to obtain a second order accurate method for (1.1). We have eliminated  $U^n$  to obtain a second order accurate method for the (as yet unknown) modified equation (1.4b). In this equation we use initial data

$$u^{**}(t_n + k) = u^*(t_n + k) = U^*$$

and view  $U^{n+1}$  as an  $O(k^3)$  approximation to  $u^{**}(t_n + 2k)$ . Replacing  $U^{n+1}$  by  $u^{**}(t_n + k) + ku_i^{**}(t_n + k) + \dots$  in (3.12) and proceeding as usual gives

$$L^{**} = \frac{1}{2}k(\partial_y^4 - \partial_x^4)$$

after a tedious calculation. Of course the same result can be obtained much more easily by using (3.10) and (1.5). However, this sort of technique is sometimes necessary in dealing with multi-step methods involving more than one intermediate solution.

#### 4. The Approximate Factorization (AF) Method

Approximate factorization methods are typically written in a form different from (1.3), namely

$$Q_x \delta_{ij}^* = P U_{ij}^n \quad (4.1a)$$

$$Q_y \delta_{ij}^* = \delta_{ij}^* \quad (4.1b)$$

where  $\delta^n = U^{n+1} - U^n$  so that the new approximation  $U^{n+1}$  is obtained by setting

$$U_{ij}^{n+1} = U_{ij}^n + \delta_{ij}^* \quad (4.2)$$

after solving (4.1). Clearly (4.1) is a two-step procedure for solving

$$Q_x Q_y U_{ij}^{n+1} = (P + Q_x Q_y) U_{ij}^n$$

and  $Q_x Q_y$  is an approximate factorization of some two-dimensional spatial operator.

For the AF method, we need only specify boundary conditions for  $\delta^*$  along  $x = x_b$ . If Dirichlet conditions are imposed, then the proper boundary conditions for  $\delta_{bj}^*$  in (4.1a) can be easily determined using (4.1b):

$$\begin{aligned} \delta_{bj}^* &= Q_y \delta_{bj}^* \\ &= Q_y (g(y_j, t_{n+1}) - g(y_j, t_n)). \end{aligned} \quad (4.3)$$

For more general boundary conditions, however, it is useful to pursue the modified equation approach.

Beam and Warming[1] consider AF methods for a more general problem with mixed derivatives:

$$u_t = au_{xx} + bu_{xy} + cu_{yy} = Lu. \quad (4.4)$$

They show that stable second order accurate methods can be obtained while handling the mixed derivative term explicitly. One such method is

$$(I - \omega ka D_x^2) \delta_{ij}^* = \frac{1}{1+\xi} \left[ k(a D_x^2 + c D_y^2) (I + (\xi - \theta + \frac{1}{2}) \nabla) + kb D_x D_y (I + (\xi + \frac{1}{2}) \nabla) + \xi \nabla \right] U_{ij}^n \quad (4.5a)$$

$$(I - \omega ka D_y^2) \delta_{ij}^* = \delta_{ij}^*. \quad (4.5b)$$

Here  $\xi$ ,  $\omega$ , and  $\theta$  are parameters satisfying

$$\omega = \theta / (1 + \xi)$$

and

$$\nabla U^n = U^n - U^{n-1} = \delta^{n-1},$$

$$D_x D_y U_{ij} = \frac{1}{4h^2} (U_{i+1,j+1} - U_{i+1,j-1} - U_{i-1,j+1} + U_{i-1,j-1}).$$

For the heat equation  $a = c = 1$ ,  $b = 0$ , and the simplest such method (corresponding to Crank-Nicolson) is obtained by taking  $\xi = 0$ ,  $\omega = \theta = 1/2$ . Then the operator  $P$  on the right hand side of (4.5a) reduces to

$$P = k(D_x^2 + D_y^2).$$

For  $b \neq 0$ , other parameter choices give more stable methods, e.g.,  $\theta = 1$ ,  $\xi = 1/2$ ,  $\omega = 2/3$  (see [1]). We have also found that with Robbins boundary conditions, the latter choice of parameters gives a more stable method even for the heat equation.

In this section, we will consider the more general method (4.5) since the additional complications only affect the operator  $P$  in (4.5a). We will see that this has essentially no effect on the derivation of boundary conditions.

Thus the method has the form (4.1) with

$$Q_x = (I - \omega k a D_x^2) \quad (4.6a)$$

$$Q_y = (I - \omega k a D_y^2) \quad (4.6b)$$

and  $P$  given by (4.5a) ( $P$  could be even more complicated – e.g., for the incompressible Navier-Stokes equations we could obtain a method of this form in which  $P$  contains nonlinear terms [7]). In order to derive the modified equation for (4.1a) without having to expand all the terms in  $P$ , we instead base our expansion on (4.1b):

$$Q_y(U_{ij}^{n+1} - U_{ij}^n) = U_{ij}^n - U_{ij}^n. \quad (4.7)$$

Assuming as usual that  $U_{ij}^n = u^*(x_i, y_j, t_n + k)$  with initial conditions (1.6), we expand (4.7) to obtain

$$(I - \omega k c \partial_y^2)(k u_x + \frac{1}{2} k^2 u_{xx} + \dots) = k u_x^n + \frac{1}{2} k^2 u_{xx}^n + \dots$$

all evaluated at time  $t_n$ . Using  $u_t = Lu = Lu^n$  at time  $t_n$ , solving for  $u_x^n$  gives

$$u_i^* = Lu^* + \frac{1}{2}k(L^2 - 2\omega c\partial_y^2 L - \partial_t^2)u^* + \dots \quad (4.8)$$

Differentiating this with respect to  $t$  gives  $u_{tt}^* = L^2u^* + O(k)$ . Inserting this in the final term of (4.8) yields

$$u_i^* = Lu^* - \omega kc\partial_y^2 Lu^* \quad (4.9)$$

plus terms which are  $O(k^2)$ . Thus (4.5a) gives a second order accurate approximation to (4.9) if we set  $U^* = U^n + \delta^*$ . We thus have

$$L^* = (I - \omega kc\partial_y^2)L. \quad (4.10)$$

We first consider the case of Dirichlet boundary conditions. Then (1.10) gives

$$u^*(x_b, y, t_n + k) = u + k(I - \omega kc\partial_y^2)Lu + \frac{1}{2}k^2L^2u + O(k^3)$$

since  $(L^*)^2 = L^2 + O(k)$ . Using the original equation  $u_t = Lu$ , we can simplify this to obtain

$$u^*(x_b, y, t_n + k) = u(x_b, y, t_n + k) - \omega k^2 c u_{yy}(x_b, y, t_n) + O(k^3). \quad (4.11)$$

We obtain the boundary condition for  $\delta^*$  by subtracting  $u(x_b, y, t_n) = g(y, t_n)$  from both sides, and dropping the  $O(k^3)$  terms,

$$\delta_{bj}^* = (g(y_j, t_{n+1}) - g(y_j, t_n)) - \omega k^2 c g_{yy}(y_j, t_n). \quad (4.12)$$

Notice that using simply

$$\delta_{bj}^* = g(y_j, t_{n+1}) - g(y_j, t_n) \quad (4.13)$$

should give second order accuracy, but again including the final term in (4.12) can lead to a significant improvement in the error constant. It is interesting to compare (4.12) to the boundary conditions (4.3) derived directly from (4.1). Using (4.6b), it is easy to check that they agree to  $O(k^3)$ .

Table 4.1 shows some numerical results with Dirichlet boundary conditions for the same problem used in Section 2. We see that (4.10) does give second order accuracy, but that the results are improved by using (4.9). Note that for a steady state problem with time-independent boundary conditions,  $g_{yy} = 0$  so that (4.10) and (4.9) agree.

For the Robbins boundary conditions (1.2b), we can differentiate (4.11) with respect to  $x$  and take the appropriate linear combination to obtain

$$\alpha^{n+1}u^*(x_b, y, t_n+k) + \beta^{n+1}u_t^*(x_b, y, t_n+k) = \gamma^{n+1} - \omega k^2 c [\alpha^{n+1}u_{yy}(x_b, y, t_n) + \beta^{n+1}u_{yy}(x_b, y, t_n)] \quad (4.14)$$

where we use the shorthand  $\alpha^{n+1} = \alpha(y, t_{n+1})$ , for example. Subtracting  $\alpha^{n+1}u(x_b, y, t_n) + \beta^{n+1}u_t(x_b, y, t_n)$  from (4.14) gives boundary conditions for  $\delta^*$ :

$$\begin{aligned} \alpha^{n+1}\delta^* + \beta^{n+1}\delta_t^* &= \gamma^{n+1} - (\alpha^{n+1}u + \beta^{n+1}u_t) - \omega k^2 c (\alpha^{n+1}u_{yy} + \beta^{n+1}u_{yy}) \\ &= (\gamma^{n+1} - \gamma^n) - [(\alpha^{n+1} - \alpha^n)u + (\beta^{n+1} - \beta^n)u_t] - \omega k^2 c (\alpha^{n+1}u_{yy} + \beta^{n+1}u_{yy}). \end{aligned} \quad (4.15)$$

Here we are abusing notation slightly and letting  $\delta^*$  also represent the function  $u^*(x, y, t) - u(x, y, t_n)$  evaluated at  $t = t_{n+1}$ . Note that if  $\alpha$  and  $\beta$  are constant, (4.15) becomes simply

$$\alpha\delta^* + \beta\delta_t^* = (\gamma^{n+1} - \gamma^n) - \omega k^2 c \gamma_{yy}^n. \quad (4.16)$$

When  $\alpha$  and  $\beta$  are not constant, it is necessary to discretize the derivatives of  $u(x_b, y, t_n)$  occurring on the right hand side of (4.15). This can easily be done to the required accuracy, but in practice it appears that great care must be taken in order to avoid numerical instabilities caused by high order differences in the boundary conditions. One possibility is to transfer as many derivatives as possible onto the functions  $\alpha$ ,  $\beta$  and  $\gamma$ , as we now discuss. First note that if we assume the nondegeneracy condition  $\beta(y, t) \neq 0$  for all  $y, t$ , we can divide (1.2b) by  $\beta(y, t)$ . Hence, by modifying  $\alpha$  and  $\gamma$  we can assume without loss of generality that  $\beta = 1$ .

With this assumption, computing  $\gamma_{yy}$  gives

$$\begin{aligned} \gamma_{yy} &= \partial_t \partial_y^2 (\alpha u + u_t) \\ &= \alpha u_{yy} + u_{yyt} + \alpha_{yy} u + \alpha_{yy} u_t + 2\alpha_y u_y + 2\alpha_y u_{ty} + \alpha_t u_{yy}. \end{aligned}$$

Solving for  $\alpha u_{yy} + u_{yyt}$  gives an expression which can be used on the right hand side of (4.15) (note that  $\alpha^{n+1}u_{yy} = \alpha^n u_{yy} + O(k)$ ). Then (4.15) becomes

$$\begin{aligned} \alpha^{n+1}\delta^* + \delta_t^* &= (\gamma^{n+1} - \gamma^n) - (\alpha^{n+1} - \alpha^n)u \\ &\quad - \omega k^2 c [\gamma_{yy}^n - \alpha_{yy}^n u - \alpha_{yy}^n u_t - 2\alpha_y^n u_y - 2\alpha_y^n u_{ty} - \alpha_t^n u_{yy}]. \end{aligned} \quad (4.17)$$

This can be discretized in a straightforward manner and appears to give much more stable boundary

conditions than directly discretizing (4.15). For completeness we present the details of our AF implementation in Section 5.

As a numerical experiment, we compare (4.17) to the more naive conditions

$$\alpha^{n+1}\delta_x^n + \beta^{n+1}\delta_x^n = (\gamma^{n+1} - \gamma^n) - (\alpha^{n+1} - \alpha^n)u \quad (4.18)$$

which do not contain the  $O(k^2)$  correction. We again use the heat equation (1.1) with solution

$$u(x,y,t) = \sin(x+y-1) e^{-2t}. \quad (4.19)$$

For simplicity, we specified Dirichlet boundary conditions along three boundaries and the Robbins condition

$$\alpha(y,t)u(1,y,t) + u_x(1,y,t) = \gamma(y,t)$$

only along the boundary  $x = 1$ . We take

$$\alpha(y,t) = \cos(y) e^t, \quad \gamma(y,t) = \frac{1}{2}\sin(2y) e^{-t} + \cos(y) e^{-2t}.$$

Table 4.2 gives a comparison of results obtained at  $t = 0.3$  with three different combinations of boundary conditions:

- a) (4.13) at  $x = 0$  and (4.18) at  $x = 1$ ,
- b) (4.12) at  $x = 0$  and (4.18) at  $x = 1$ , (4.20)
- c) (4.12) at  $x = 0$  and (4.17) at  $x = 1$ .

As expected, all three combinations give second order accuracy, but including the correct  $O(k^2)$  terms improves the error constant. The combination (4.20c) gives errors about 10 times smaller than (4.20a).

In this experiment we have used the parameters  $\theta = 1$ ,  $\xi = 1/2$ ,  $\omega = 2/3$  for improved stability.

## 5. Implementation of the AF Method

We include a detailed discussion of our implementation of (4.5) since in performing the experiments presented above it was found that both the accuracy and stability of the scheme were greatly affected by the manner in which the boundary conditions were imposed.

For each fixed  $j = 1, 2, \dots, N-1$ , (4.5a) gives rise to a tridiagonal system of equations of the form

$$-\omega r a \delta_{i-1,j}^n - (1 + 2\omega r a) \delta_{ij}^n - \omega r a \delta_{i+1,j}^n = p_{ij} \quad (5.1)$$

for  $i = 1, 2, \dots, N-1$ , where  $r = k/h^2$  and  $p_{ij}$  is given by the right hand side of (4.5a). Note that  $\nabla U_{ij}^n = \delta_{ij}^{n-1}$  which can be saved from the previous time step. Since in general this is a three level scheme, two time levels are required as initial data. In our numerical experiments we used the exact solution at time 0 and  $k$ , but in practice a two-level scheme must be used for the first step.

When Dirichlet conditions are imposed, the system (5.1) is completed by specifying  $\delta_{0j}^n$  and  $\delta_{Nj}^n$  as discussed in Section 4. For the Robbins condition we discretize the boundary conditions obtained for  $\delta^n$  in Section 4. For example, when  $\alpha$  and  $\beta$  are constant we discretize (4.16) as

$$\alpha \delta_{bj}^n + \frac{\beta}{2h} (\delta_{b+1,j}^n - \delta_{b-1,j}^n) = \hat{p}_{bj} \quad (5.2)$$

for  $b = 0$  or  $N$ , with  $\hat{p}_{bj}$  representing the right hand side of (4.16). This can be combined with equation (5.1) for  $i = b$  to eliminate  $\delta_{-1,j}^n$  (when  $b = 0$ ) or  $\delta_{N+1,j}^n$  (when  $b = N$ ). For example, at the right boundary ( $b = N$ ) we obtain

$$\delta_{N+1,j}^n = \delta_{N-1,j}^n + \frac{2h}{\beta} (\hat{p}_{Nj} - \alpha \delta_{Nj}^n) \quad (5.3)$$

from (5.2). Inserting this in (5.1) gives the equation

$$-2\omega r a \delta_{N-1,j}^n + [1 + 2\omega r a(1 + h\alpha/\beta)] \delta_{Nj}^n = p_{Nj} + \frac{2\omega r a h}{\beta} \hat{p}_{Nj}$$

with a similar equation obtained for  $i = 0$ . We thus have a tridiagonal system of  $N + 1$  equations for  $\delta_{ij}^n$ ,  $i = 0, 1, \dots, N$ .

Note that the expression for  $p_{ij}$  (the right hand side of (4.5a)) involves second differences of  $U_{ij}^n$  and  $\delta_{ij}^{n-1}$  in the  $x$ -direction and hence evaluating  $p_{Nj}$  requires values  $U_{N+1,j}^n$  and  $\delta_{N+1,j}^n$ . These can be obtained in the same way we derived (5.3) by discretizing (1.2b). We obtain

$$U_{N+1,j}^n = U_{N-1,j}^n + \frac{2h}{\beta} (\gamma_j^n - \alpha U_{Nj}^n) \quad (5.4)$$

and

$$\delta_{N+1,J}^{n-1} = U_{N+1,J}^n - U_{N+1,J}^{n-1}.$$

Actually, in practice we have found that the use of (5.4) leads to a considerable loss of accuracy near the boundary which can be avoided by using a higher order approximation to the derivative. If (1.2b) is instead discretized by

$$\alpha U_{Nj}^n + \frac{1}{6h} \beta (2U_{N+1,J}^n + 3U_{Nj}^n - 6U_{N-1,J}^n + U_{N-2,J}^n) = \gamma_j^n$$

we obtain

$$U_{N+1,J}^n = \frac{1}{2}(-3U_{Nj}^n + 6U_{N-1,J}^n - U_{N-2,J}^n) + \frac{6h}{\beta}(\gamma_j^n - \alpha U_{Nj}^n) \quad (5.5)$$

in place of (5.4). This did not seem to adversely affect the numerical stability but improved the errors considerably.

When  $\alpha$  depends of  $y$  and  $t$  (and  $\beta = 1$  as discussed in Section 4), the boundary condition (4.17) is discretized to again give (5.3) with  $\alpha$  replaced by  $\alpha_j^{n+1}$  and

$$\begin{aligned} \hat{p}_{Nj} = & (\gamma_j^{n+1} - \gamma_j^n) - (\alpha_j^{n+1} - \alpha_j^n)U_{Nj}^n - \omega k^2 c [\gamma_{yy}^n - \alpha_{yy}^n U_{Nj}^n - \alpha_{yy}^n \delta_{Nj}^{n-1}/k - \alpha_y^n (U_{Nj+1}^n - U_{Nj-1}^n)/h \\ & - \alpha_y^n (\delta_{Nj+1}^{n-1} - \delta_{Nj-1}^{n-1})/kh - \alpha_t^n (U_{Nj+1}^n - 2U_{Nj}^n + U_{Nj-1}^n)/h^2]. \end{aligned}$$

## 6. Multi-step methods

In many situations it is necessary to use fractional step methods involving more than two steps. Consequently, additional intermediate solutions arise for which boundary conditions must also be specified.

For example, the LOD and LF methods are easily extended to three space dimensions by adding a third step to the process. We consider the LOD method, which becomes

$$(I - \frac{1}{2}kD_x^2)U^n = (I + \frac{1}{2}kD_x^2)U^{n-1} \quad (6.1a)$$

$$(I - \frac{1}{2}kD_y^2)U^{n-1} = (I + \frac{1}{2}kD_y^2)U^{n-2} \quad (6.1b)$$

$$(I - \frac{1}{2}kD_z^2)U^{n-2} = (I + \frac{1}{2}kD_z^2)U^{n-3}. \quad (6.1c)$$

For clarity we have dropped the (three) subscripts on these variables and will also suppress the spatial

arguments of functions where convenient. The method (6.1) corresponds to the splitting

$$L = L^* + L^{**} + L^{***}$$

with  $L^*$  and  $L^{**}$  given by (1.6) and  $L^{***} = \partial_z^2$ . The boundary conditions for  $u^*$  along  $x = x_b$  are again given by (2.3), but now we have

$$\begin{aligned} u_{xx} &= (L - L^{**} - L^{***})u = u_t - u_{yy} - u_{zz} \\ u_{xxxx} &= (L - L^{**} - L^{***})^2 u. \end{aligned}$$

The boundary conditions for  $U^*$  along  $x = x_b$  become

$$U_{bij}^* = [I - k(\partial_y^2 + \partial_z^2) + \frac{1}{2}k^2(\partial_y^2 + \partial_z^2)^2] u(x_b, y_i, z_j, t_n + k). \quad (6.2)$$

Note that this involves only tangential derivatives along the boundary  $x = x_b$ .

In the second step, we need boundary data for  $U^{**}$  along  $y = y_b$ . Recall that we view  $U^{**}$  as an approximation to  $u^{**}(t_n + 2k)$ , where  $u^{**}$  satisfies (1.4b) with initial conditions  $u^{**} = u^*$  at  $t = t_n + k$ .

Hence we can expand

$$u^{**}(t_n + 2k) = [I + kL^{**} + \frac{1}{2}k^2(L^{**})^2 + \dots] u^*(t_n + k). \quad (6.3)$$

Using the previously determined expansion (1.10) in (6.3) gives boundary data for  $U^{**}$  in terms of  $u(t_n)$ .

Actually, for three-step methods such as this, there is a much easier approach. By (6.1c), we can view  $U^{**}$  as an approximation to the function obtained by solving (1.4b) *backwards* in time from  $U^{n+1} = u(t_{n+1})$ . With time step  $-k$  we thus obtain

$$U^{**} = [I - kL^{***} + \frac{1}{2}k^2(L^{***})^2 - \dots] u(t_{n+1})$$

so that the boundary conditions are

$$U_{ibj}^{**} = u - ku_{zz} + \frac{1}{2}k^2 u_{zzzz}$$

all evaluated at  $(x_i, y_b, z_j, t_{n+1})$ .

Finally, we note that even in two space dimensions it may be necessary to use a three-step method if the operator  $L$  is split as the sum of two noncommuting operators, as is typically the case in variable coef-

efficient or nonlinear problems. Then second order accuracy is retained by using a three-step Strang splitting[12].

Table 2.1

BC	N	k	$L_\infty$ -error	$L_2$ -error	$L_2$ -rate
(2.11)	11	.15	8.14E-3	3.69E-3	
	21	.075	4.50E-3	1.97E-3	.91
	31	.05	3.17E-3	1.35E-3	.93
(2.12)	11	.15	1.54E-3	7.05E-4	
	21	.075	4.38E-4	1.94E-4	1.86
	31	.05	2.04E-4	8.93E-5	1.91
(2.5)	11	.15	1.07E-4	5.49E-5	
	21	.075	1.51E-5	8.66E-6	2.66
	31	.05	4.73E-6	3.04E-6	2.58

Table 2.2

BC	N	k	$L_\infty$ -error	$L_2$ -error	$L_2$ -rate
(2.11)	11	.15	1.55E-2	5.28E-3	
	21	.075	9.87E-3	3.02E-3	.81
	31	.05	7.23E-3	2.11E-3	.88
(2.12)	11	.15	2.64E-3	1.06E-3	
	21	.075	7.94E-4	2.93E-4	1.86
	31	.05	3.76E-4	1.34E-4	1.93
(2.5)	11	.15	1.02E-4	4.45E-5	
	21	.075	1.59E-5	8.59E-6	2.37
	31	.05	6.09E-6	3.41E-6	2.28

Table 4.1

BC	N	k	$L_\infty$ -error	$L_2$ -error	$L_2$ -rate
(4.10)	11	.15	2.12E-3	9.55E-4	
	21	.075	5.75E-4	2.50E-4	1.93
	31	.05	2.63E-4	1.13E-4	1.96
(4.9)	11	.15	8.65E-4	2.71E-4	
	21	.075	2.57E-4	7.80E-5	1.80
	31	.05	1.20E-4	3.64E-5	1.88

Table 4.2

BC	N	k	$L_\infty$ -error	$L_2$ -error	$L_2$ -rate
(4.20a)	11	.06	1.35E-3	4.38E-4	
	21	.03	4.11E-4	1.15E-4	1.93
	31	.02	1.94E-3	5.20E-5	1.96
(4.20b)	11	.06	6.55E-4	2.41E-4	
	21	.03	1.64E-4	5.57E-5	2.11
	31	.02	7.32E-5	2.41E-5	2.07
(4.20c)	11	.06	1.40E-4	4.96E-5	
	21	.03	3.46E-4	9.80E-6	2.34
	31	.02	1.54E-5	4.00E-6	2.21

## References

- [1] R.M. Beam and R.F. Warming, Alternating direction implicit methods for parabolic equations with a mixed derivative, *SIAM J. Sci. Stat. Comp.* 1(1980), pp. 131-159.
- [2] W.R. Briley and H. McDonald, On the structure and use of linearized block implicit schemes, *J. Comput. Phys.* 34(1980), pp. 54-73.
- [3] J. Douglas Jr. and J.E. Gunn, A general formulation of alternating direction methods Part I. parabolic and hyperbolic systems, *Numer. Math.* 6(1964), pp. 428-453.
- [4] D.L. Dwoyer and F.C. Thames, Accuracy and stability of time-split finite-difference schemes, 5th AIAA Computational Fluid Dynamics Conference, Palo Alto, CA, 1981.
- [5] G. Fairweather and A.R. Mitchell, A new computational procedure for ADI methods, *SIAM J. Numer. Anal.* 4(1967), pp. 163-170.
- [6] B. Gustafsson, The convergence rate for difference approximations to mixed initial boundary value problems, *Math. Comp.* 29(1975), pp. 396-406.
- [7] J. Kim and P. Moin, Applications of a fractional-step method to incompressible Navier-Stokes equations, to appear in *J. Comp. Phys.*, 1985.
- [8] R.J. LeVeque, Time-split methods for partial differential equations, Ph.D. Thesis, Stanford Computer Science Department, Stanford University, 1982.
- [9] R.J. LeVeque and J. Olinger, Numerical methods based on additive splittings for hyperbolic partial differential equations, *Math. Comp.* 40(1983), pp. 469-497.
- [10] A.R. Mitchell, *Computational Methods in Partial Differential Equations*, John Wiley, London, 1969.
- [11] D.W. Peaceman and H.H. Rachford, Jr., The numerical solution of parabolic and elliptic differential equations, *J. Soc. Indust. Appl. Math.* 3(1955), pp. 28-41.
- [12] G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* 5(1968), pp. 506-517.
- [13] N.N. Yanenko, *The Method of Fractional Steps*, Springer-Verlag, 1971.

1. Report No. NASA CR-172591 ICASE Report No. 85-21		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Intermediate Boundary Conditions for LOD, ADI, and Approximate Factorization Methods				5. Report Date March 1985	
				6. Performing Organization Code	
7. Author(s) Randall J. LeVeque				8. Performing Organization Report No. 85-21	
9. Performing Organization Name and Address Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665				10. Work Unit No.	
				11. Contract or Grant No. NAS1-17070	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546				13. Type of Report and Period Covered Contractor Report	
				14. Sponsoring Agency Code 505-31-83-01	
15. Supplementary Notes Langley Technical Monitor: J. C. South, Jr. Final Report				Submitted to Journal of Computational Physics	
16. Abstract  A general approach to determining the correct intermediate boundary conditions for dimensional splitting methods is presented and illustrated. The intermediate solution $U^*$ is viewed as a second-order accurate approximation to a modified equation. Deriving the modified equation and using the relationship between this equation and the original equation allows us to determine the correct boundary conditions for $U^*$ . To illustrate this technique, we apply it to LOD and ADI methods for the heat equation in two and three space dimensions. The approximate factorization method is considered in slightly more generality.					
17. Key Words (Suggested by Author(s)) boundary conditions, ADI approximate factorization fractional step, splitting methods modified equations			18. Distribution Statement 64 - Numerical Analysis  Unclassified - Unlimited		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 25	22. Price A02