Pergamon

0045-7930(94)00039-5

Computers & Fluids Vol. 24, No. 4, pp. 447–458, 1995

0045-7930/95 $9.50 + 0.00

INTERMEDIATE DIRICHLET BOUNDARY CONDITIONS FOR OPERATOR SPLITTING ALGORITHMS FOR THE ADVECTION–DIFFUSION EQUATION

LIAQAT ALI KHAN† and PHILIP L.-F. LIU

School of Civil and Environmental Engineering, Cornell University, Ithaca, NY 14853, U.S.A.

(Received 15 December 1993; in revised form 24 October 1994)

Abstract—When operator splitting algorithms are used to solve the advection–diffusion equation, it is necessary to derive boundary conditions applicable to the split advection and diffusion equations. In this paper intermediate Dirichlet boundary conditions are formulated for Strang type splitting algorithms for the one-dimensional advection–diffusion equation. The derived boundary conditions are applicable to advection dominated problems and are $\mathcal{O}(\min(k\epsilon^3, k^2\epsilon^2))$ accurate, where k is the computational time step and $\epsilon \ll 1$ is the reciprocal of Peclet number.

1. INTRODUCTION

Operator splitting algorithms are frequently employed to solve the advection–diffusion equation [1]. For advection dominated transport problems, the algorithm generally consists of splitting up the governing equation into an advection equation and a diffusion equation. The split equations are then solved sequentially to approximate the solution of the governing equation. The intermediate solutions, resulting from the use of splitting algorithms, are numerical artifacts and have no physical equivalence. In the physical problem all the transport processes—advection and diffusion—take place simultaneously. However, in splitting algorithms these processes are imposed on the initial condition sequentially. Therefore, operator splitting algorithms can only approximate the physical problem after all the transport processes have been allowed to take place in a computational cycle. As the boundary conditions are defined for the unsplit governing equation, it is necessary to determine appropriate boundary conditions for the split equations.

A large number of research works [2, 3] has been reported on the derivation of intermediate Dirichlet boundary conditions for locally one-dimensional methods. In locally one-dimensional methods, multi-dimensional governing equations are split into one-dimensional equations. A review of different methods of deriving intermediate Dirichlet boundary conditions for the two-dimensional diffusion equation can be found in Yanenko [2]. Sommeijer *et al.* [3] have analyzed the accuracy of different numerical approximations to these boundary conditions. An approach similar to that reported for parabolic equations has been utilized by Gourlay and Morris [4], McGuire and Morris [5] and Gourlay and Mitchell [6] for two-dimensional hyperbolic equations. The boundary conditions derived in these studies can be viewed as applications of the method of undetermined coefficients [2] to the discretized equations. Consequently, algebraic expressions of the derived boundary conditions are functions of the numerical schemes used to discretize the split equations.

Unlike locally one-dimensional methods dealing with either parabolic or hyperbolic equations, splitting of an advection–diffusion equation into an advection equation and a diffusion equation results in different classes of equations. As a consequence, a direct adaptation of the discretized boundary conditions presented in the context of a locally one-dimensional method is not feasible when dealing with the advection–diffusion equation. This is especially true when different numerical procedures, like method of characteristics and finite-difference or finite-element method, are combined in developing the composite numerical procedure. The presence of spatial gradients of the dependent variables on the boundary further hinder their adaptation. It is necessary to design spatial discretization of the intermediate boundary conditions compatible with the numerical

†Present address: HydroQual, Inc., 1 Lethbridge Plaza, Mahwah, NJ 07430, U.S.A.

scheme for the differential equation. Otherwise, the resulting composite algorithm can be numerically unstable [5].

An algorithm to determine intermediate boundary conditions for a one-dimensional system of hyperbolic equations has been proposed by LeVeque [7]. The system of equations can be expressed as $c_t = Ac_x$, where $c(t,x)$ is a vector of dependent variables, and A is a matrix such that $A = A_1 + A_2$. The split equations are $c_t^* = A_1 c_x^*$ and $c_t^{**} = A_2 c_x^{**}$. The derivation of intermediate boundary conditions is based on the assumption that A, A_1 and A_2 are invertible. Therefore, $c_x = A^{-1}c_t$, $c_x^* = A_1^{-1}c_t^*$ and $c_x^{**} = A_2^{-1}c_t^{**}$. These relations allow the change of variables from c_x^* and c_x^{**} to c_t^* and c_t^{**} respectively, and switching from the latter variables to c_t. The resulting intermediate boundary conditions are independent of the numerical scheme used to solve the split equations. For the advection–diffusion equation, LeVeque's [7] method requires determination of the inverse of advection–diffusion operator, rather than the inverse of matrices. Consequently, the approach is not applicable when dealing with the advection–diffusion equation.

In the context of the advection–diffusion equation, the only reported study is by Aiyesimoju and Sobey [8] for a first-order accurate splitting algorithm. The theory of constructing first- and second-order accurate splitting algorithms can be found in Marchuk [1] and Yanenko [2]. Compared with a second-order accurate splitting algorithm, also known as Strang [9, 10] type splitting, a first-order accurate algorithm requires the determination of fewer and less accurate intermediate boundary conditions. In deriving intermediate boundary conditions, Aiyesimoju and Sobey [8] do not distinguish between the dependent variables of the governing equation, $c(t,x)$, the split advection equation, $c^*(t,x)$, and the split diffusion equation, $c^{**}(t,x)$. The analysis ignores the fact that different differential equations and physical processes are responsible for the time evolution of the dependent variables $c(t,x)$, $c^*(t,x)$ and $c^{**}(t,x)$. Additional discussions on the intermediate boundary conditions by Aiyesimoju and Sobey [8] are presented in Section 4.2.

Some of the recently reported operator splitting algorithms for the one-dimensional advection–diffusion equation are by Li [11, 12], Muralidhar *et al.* [13], Noye [14], Szymkiewicz [15], Wheeler *et al.* [16], and Xin and Wong [17]. Due to splitting errors [10], these splitting algorithms are first-order accurate in time. No attempt has been made to derive time dependent intermediate boundary conditions for the split equations. The boundary conditions defined by the physical problem for the governing equation have been directly applied to the split equations. As the simulation characteristics of the algorithms are demonstrated for boundary conditions that are independent of time, results presented in these studies appear acceptable. However, such an approach is not applicable if the boundary conditions are time dependent.

Considering the number of operator splitting algorithms reported for the advection–diffusion equation and the Navier–Stokes equations (see Demkowicz *et al.* [18] and Perot [19] for discussions and references), very "little work has been done to study the effects of boundary conditions on the performance and accuracy of the splitting methods" [18]. As a result, "serious confusion and/or disagreement concerning boundary conditions and the details of the methods implementation exists" [19]. Unless appropriate boundary conditions are developed for the split advection and diffusion equations, practical utility of operator splitting algorithms for the advection–diffusion equation will remain limited.

In this paper intermediate Dirichlet boundary conditions for a Strang [9] type operator splitting algorithm for the one-dimensional advection–diffusion equation are presented. The derived boundary conditions are applicable to advection dominated transport problems. They are independent of the numerical procedures for solving the split equations, and can be directly adopted by any operator splitting algorithm for the one-dimensional advection–diffusion equation. The rest of the paper is organized as follows: the physical problem is defined in Section 2. An operator spliting algorithm for the advection–diffusion equation is described in Section 3. The derivation of intermediate boundary conditions is presented in Section 4. Numerical examples are presented in Section 5. Summary and conclusions of the study are listed in Section 6.

2. GOVERNING EQUATION

The one-dimensional advection–diffusion equation can be expressed as

$$c_t = -\hat{u}c_{\hat{x}} + \hat{D}c_{\hat{x}\hat{x}}, \tag{1}$$

where $c(\hat{t}, \hat{x})$ is concentration, $\hat{u}(\hat{t}, \hat{x})$ is velocity, $\hat{D}(\hat{t}, \hat{x})$ is diffusion coefficient, and $\hat{x}$ and $\hat{t}$ are the spatial and time coordinates, respectively. The solution of equation (1) is sought in a domain Ω, subjected to the following initial condition

$$c(0, \hat{x}) = f^0(\hat{x}), \quad \hat{x} \in \Omega, \tag{2}$$

and boundary conditions

$$c(\hat{t}, \hat{x}) = f^1(\hat{t}), \quad \hat{t} > 0, \quad \hat{x} \in \Gamma_1, \tag{3}$$

$$\hat{u}c - \hat{D}c_{\hat{x}} = f^2(\hat{t}), \quad \hat{t} > 0, \quad \hat{x} \in \Gamma_2, \tag{4}$$

where Γ_1 and Γ_2 are the boundaries of Ω, and f^0, f^1 and f^2 are known functions.

Introducing non-dimensional variables

$$x = \frac{\hat{x}}{L}, \quad t = \frac{\hat{t}}{T}, \quad u = \frac{\hat{u}}{T/L}, \quad D = \frac{\hat{D}}{D_0}, \tag{5}$$

where L, T and D_0 are the representative values of $\hat{x}$, $\hat{t}$ and $\hat{D}$ respectively, equation (1) can be expressed as

$$c_t = -uc_x + DP_e^{-1}c_{xx}, \tag{6}$$

where $P_e = L^2/D_0T$ is the Peclet number. For convenience, equation (6) is expressed as

$$c_t = -uc_x + \epsilon c_{xx}, \tag{7}$$

where $\epsilon = D/P_e$. In this study the transport problem is assumed advection dominated such that $P_e \gg 1$ and $\epsilon \ll 1$.

3. AN OPERATOR SPLITTING ALGORITHM

A second-order accurate Strang type splitting algorithm [9, 10] for equation (7) can be formulated by solving the following split equations over the indicated time steps and subjected to the accompanying initial conditions

$$c_t^* = -uc_x^*, \quad c^*(t_n, x) = c(t_n, x), \quad t \in [t_n, t_{n+1/2}], \tag{8}$$

$$c_t^{**} = \epsilon c_{xx}^{**}, \quad c^{**}(t_n, x) = c^*(t_{n+1/2}, x), \quad t \in [t_n, t_{n+1}], \tag{9}$$

$$c_t^* = -uc_x^*, \quad c^*(t_{n+1/2}, x) = c^{**}(t_{n+1}, x), \quad t \in [t_{n+1/2}, t_{n+1}], \tag{10}$$

and set $c(t_{n+1}, x) = c^*(t_{n+1}, x)$. In the above equations $c^*(t, x)$ and $c^{**}(t, x)$ are the dependent variables of the split advection and diffusion equations, respectively, n is the number of time steps, $t_n = nk$, and k is the time step. The dependent variable $c(t, x)$ as approximated by equations (8)–(10) is defined only at the beginning and the end of a computational cycle: $c^*(t_n, x) = c(t_n, x)$ by the initial condition, $c^*(t_{n+1}, x) \approx c(t_{n+1}, x)$ by the convergence [10] of the splitting algorithm to the solution of equation (7), and $c^*(t_{n+1/2}, x) \neq c(t_{n+1/2}, x)$, $c^{**}(t_n, x) \neq c(t_n, x)$, $c^{**}(t_{n+1}, x) \neq c(t_{n+1}, x)$.

For a Strang type splitting algorithm, the composite solution will be second-order accurate [9, 10] provided the numerical procedures for the split equations are second-order accurate in time and space. As the objective of this paper is to illustrate the derivation of intermediate boundary conditions, the split equations (8)–(10) are solved by two well known numerical procedures. The advection equations (8) and (10) are solved by a backward method of characteristics using Holly and Preissmann [20] scheme. The diffusion equation (9) is solved by the Crank–Nicolson finite-difference method [21]. The composite algorithm is therefore, unconditionally stable and second-order accurate.

4. INTERMEDIATE BOUNDARY CONDITIONS

Intermediate boundary conditions for the split equations (8)–(10) are presented in this section. These boundary conditions are derived for advection dominated transport problem such that $P_e \gg 1$ and $\epsilon \ll 1$. In addition, u and ϵ are assumed locally constant. The latter assumption is not essential, but simplifies the presentation.

4.1. Derivation

First step. Equation (8) updates $c^*(t, x)$ over a period $k/2$ subjected to initial condition $c^*(t_n, x) = c(t_n, x)$. Therefore, the equation can be integrated to

$$c^*(t_{n+1/2}, x) = \exp\left(-\frac{ku}{2}\partial_x\right)c(t_n, x). \tag{11}$$

An approximation to $c^*(t_{n+1/2}, x)$ can be obtained by expanding the exponential operator in a Taylor series as

$$c^*(t_{n+1/2}, x) = c(t_n, x) - \frac{k}{2}uc_x(t_n, x) + \frac{k^2}{8}u^2c_{xx}(t_n, x) + \mathcal{O}(k^3). \tag{12}$$

To express boundary conditions in terms of $f^1(t)$, it is necessary to replace $c_x(t, x)$ and $c_{xx}(t, x)$ by temporal variations of $c(t, x)$. The following approximations to these spatial gradients

$$uc_x(t, x) = -c_t + \frac{\epsilon}{u^2}c_{tt} - \frac{2\epsilon^2}{u^4}c_{ttt} + \frac{5\epsilon^3}{u^6}c_{tttt} + \mathcal{O}(\epsilon^4), \tag{13}$$

$$u^2c_{xx}(t, x) = c_{tt} - \frac{2\epsilon}{u^2}c_{ttt} + \frac{5\epsilon^2}{u^4}c_{tttt} + \mathcal{O}(\epsilon^3), \tag{14}$$

can be obtained as shown in the Appendix. Using the above relations, equation (12) can be expressed as

$$\begin{aligned} c^*(t_{n+1/2}, x) = c(t_n, x) + \frac{k}{2}c_t + \frac{k}{2}\left(\frac{k}{4} - \frac{\epsilon}{u^2}\right)c_{tt} + \frac{k\epsilon}{u^2}\left(\frac{\epsilon}{u^2} - \frac{k}{4}\right)c_{ttt} \\ + \frac{5k\epsilon^2}{2u^4}\left(\frac{k}{4} - \frac{\epsilon}{u^2}\right)c_{tttt} + \mathcal{O}(\min(k\epsilon^4, k^2\epsilon^3)). \end{aligned} \tag{15}$$

By the compatibility of boundary and initial conditions

$$c(t_n, x_0) = f^1(t_n), \tag{16}$$

where $x_0 \in \Gamma_1$. Therefore, boundary condition for equation (8) can be expressed as

$$\begin{aligned} c^*(t_{n+1/2}, x_0) = f^1(t_n) + \frac{k}{2}f^1_t + \frac{k}{2}\left(\frac{k}{4} - \frac{\epsilon}{u^2}\right)f^1_{tt} + \frac{k\epsilon}{u^2}\left(\frac{\epsilon}{u^2} - \frac{k}{4}\right)f^1_{ttt} \\ + \frac{5k\epsilon^2}{2u^4}\left(\frac{k}{4} - \frac{\epsilon}{u^2}\right)f^1_{tttt} + \mathcal{O}(\min(k\epsilon^4, k^2\epsilon^3)), \end{aligned} \tag{17}$$

where all the time derivatives of $f^1(t)$ are evaluated at $t = t_n$.

Second step. Considering equation (10) as an initial value problem, the equation can be integrated with respect to t to obtain

$$c^*(t_{n+1}, x) = \exp\left(-\frac{ku}{2}\partial_x\right)c^{**}(t_{n+1}, x). \tag{18}$$

Multiplication of equation (18) by $\exp(ku\,\partial_x/2)$ results in

$$c^{**}(t_{n+1}, x) = \exp\left(\frac{ku}{2}\partial_x\right)c^*(t_{n+1}, x). \tag{19}$$

By the convergence [7, 10] of splitting algorithms to the solution of the governing equation

$$c^*(t_{n+1}, x) \approx c(t_{n+1}, x), \tag{20}$$

and matching condition on the boundary is

$$c(t_{n+1}, x_0) = f^1(t_{n+1}). \tag{21}$$

Therefore, equation (19) can be written as

$$c^{**}(t_{n+1}, x) \approx \exp\left(\frac{ku}{2}\partial_x\right)c(t_{n+1}, x). \tag{22}$$

A second-order accurate approximation to the equation is

$$c^{**}(t_{n+1}, x) = c(t_{n+1}, x) + \frac{k}{2} u c_x(t_{n+1}, x) + \frac{k^2}{8} u^2 c_{xx}(t_{n+1}, x) + \mathcal{O}(k^3). \tag{23}$$

By equations (13), (14) and (21), boundary condition for equation (9) is

$$c^{**}(t_{n+1}, x_0) = f^1(t_{n+1}) - \frac{k}{2} f^1_t + \frac{k}{2}\left(\frac{k}{4} + \frac{\epsilon}{u^2}\right) f^1_{tt} - \frac{k\epsilon}{u^2}\left(\frac{\epsilon}{u^2} + \frac{k}{4}\right) f^1_{ttt} + \frac{5k\epsilon^2}{2u^4}\left(\frac{k}{4} + \frac{\epsilon}{u^2}\right) f^1_{tttt} + \mathcal{O}(\min(k\epsilon^4, k^2\epsilon^3)), \tag{24}$$

where all the time derivatives of $f^1(t)$ are evaluated at $t = t_{n+1}$.

Third step. At the end of a computational cycle, the solution obtained by a splitting algorithm converges to the solution of equation (7). The resulting boundary condition for equation (10) follows from equations (20) and (21) as

$$c^*(t_{n+1}, x_0) = f^1(t_{n+1}). \tag{25}$$

Equations (17), (24) and (25) are the necessary intermediate Dirichlet boundary conditions for a second-order accurate operator splitting algorithm for the advection–diffusion equation. If $f^1(t)$ is assumed constant, then $c^*(t_{n+1/2}, x_0) = c^{**}(t_{n+1}, x_0) = c^*(t_{n+1}, x_0) = f^1$, and boundary condition (3) for the governing equation can be directly applied to the split equations (8)–(10).

4.2. Discussions

Note that $\epsilon < 1$ is an essential condition for expressing equations (17) and (24) in convergent sequences in ϵ, and to exclude the possibility of $u = 0$. If $\epsilon = \mathcal{O}(k)$, then these intermediate boundary conditions are $\mathcal{O}(k^4)$ accurate. Since Strang type operator splitting algorithms are second-order accurate, it is sufficient to keep the first three terms in equations (17) and (24). As $\epsilon \to 1$, it will be necessary to include additional terms to maintain a given accuracy.

Note that if $u = 0$ on the boundary, then characteristics do not originate from the boundary and no boundary condition is needed for the advection equation (8) or (10). From equation (11), $c^*(t_{n+1/2}, x_0) = c(t_n, x_0)$, which can be combined with the initial condition for equation (9) to obtain $c^{**}(t_n, x_0) = c^*(t_{n+1/2}, x_0) = c(t_n, x_0) = f^1(t_n)$. Similarly, from equations (18), (20) and (21) $c^{**}(t_{n+1}, x_0) = c^*(t_{n+1}, x_0) = c(t_{n+1}, x_0) = f^1(t_{n+1})$. These relationships indicate that total change in the boundary value of concentration, $f^1(t)$, takes place in the diffusion step of a computational cycle. Consequently, boundary condition (3) is directly applicable to the split diffusion equation (9), and there is no need to compute intermediate values of boundary concentration.

Operator splitting algorithms are generally used for advection dominated transport problems [8, 11–17]. As $\epsilon \to 0$ ($P_e \to \infty$), equation (7) can be considered as a perturbation problem. In each computational time step the solution is essentially determined by the advection equation. The contribution of diffusion equation can be viewed as correction term—mainly smoothing out segments of the concentration profile approaching discontinuity in $c_x(t, x)$. Even if $\epsilon \ll 1$ and the solution is smooth, the influence of diffusion will be significant as $t \to \infty$. Consequently, the split diffusion equation can not be completely ignored in the computations. Such advection dominated problems are commonly encountered in environmental contaminant transport and fluid flow analyses. Therefore, boundary conditions (17), (24) and (25) are applicable to an important class of transport problems.

The boundary conditions (17) and (25) can be used if the split equations are solved in the sequence indicated by equations (8)–(10), i.e. solving the split advection equation at the beginning and end of a computational cycle. If the split diffusion equation is solved at the beginning and the end of a computational cycle, then the present formulation should be appropriately modified. The resulting intermediate boundary conditions will contain higher order derivatives of $f^1(t)$, compared with equations (17) and (24), for the same order of accuracy. However, splitting algorithms solving split advection and diffusion equations in the sequence as indicated by equations (8)–(10) are preferable for advection dominated problems.

In view of the theory of operator splitting algorithms [10], briefly described in Section 3, the intermediate boundary conditions presented by Aiyesimoju and Sobey [8] (for examples Methods A and E) can only be obtained by assuming $c_t = c_t^* = c_t^{**}$, $c_{tt} = c_{tt}^* = c_{tt}^{**}$, $c_x = c_x^* = c_x^{**}$, $c_{xx} = c_{xx}^* = c_{xx}^{**}$ etc. for $t_n \leqslant t \leqslant t_{n+1}$. Obviously, these relations are not true. For example, equations (7), (8) and (9) clearly illustrate that $c_t \neq c_t^* \neq c_t^{**}$, which also implies that $c_{tt} \neq c_{tt}^* \neq c_{tt}^{**}$. In a similar manner, it can be easily shown that the spatial gradients of the three dependent variables are not equal in a computational cycle. Therefore, the intermediate boundary conditions presented by Aiyesimoju and Sobey [8] are not consistent with the basic theory of operator splitting algorithms. The erroneous formulation of intermediate boundary conditions, not recognizing the fact that equations for $c(t, x)$, $c^*(t, x)$ and $c^{**}(t, x)$ do not represent identical physical problems and physical processes, is the result of using the same symbol, $c(t, x)$, to represent dependent variables of all the equations involved in an operator splitting algorithm.

The first-order accurate operator splitting algorithms presented by Aiyesimoju and Sobey [8], Li [11, 12], Muralidhar *et al.* [13], Noye [14], Szymkiewicz [15], Wheeler *et al.* [16] and Xin and Wong [17] are based on split equations (8) and (9) in a computational cycle. Both the split equations are solved with time step k. For such algorithms, two intermediate boundary conditions are necessary. The boundary condition for the first equation is given by equation (17) with k replaced by $2k$. The second boundary condition is given by equation (25).

5. NUMERICAL EXAMPLES

Three numerical examples are described in this section. The first example is for constant Dirichlet boundary condition. The objective of this example is to illustrate the characteristics of the operator splitting algorithm when boundary conditions do not introduce additional computational errors. The second and third examples demonstrate applications of time dependent Dirichlet boundary conditions.

5.1. Example 1

An analytical solution [22] of equation (1) for constant $\hat{u}$ and $\hat{D}$, and the following initial and boundary conditions

$$\begin{aligned} &f^0(\hat{x}) = 0, \quad \hat{x} \in [0, \infty), \\ &f^1(\hat{t}) = 1.0, \quad \hat{x} = 0, \quad f^1(\hat{t}) = 0, \quad \hat{x} \to \infty, \end{aligned} \tag{26}$$

is given by

$$c(\hat{t}, \hat{x}) = \frac{1}{2}\left[\operatorname{erfc}\left(\frac{\hat{x} - \hat{u}\hat{t}}{2\sqrt{\hat{D}\hat{t}}}\right) + \exp\left(\frac{\hat{u}\hat{x}}{\hat{D}}\right)\operatorname{erfc}\left(\frac{\hat{x} + \hat{u}\hat{t}}{2\sqrt{\hat{D}\hat{t}}}\right)\right]. \tag{27}$$

For $\hat{u} = 0.125\ \mathrm{m\,s^{-1}}$, $\hat{D} = 0.0025\ \mathrm{m^2\,s^{-1}}$, time step $\hat{k} = 1.0$ s and a nodal spacing $\hat{h} = 0.25$ m, Fig. 1 compares the numerical and analytical solutions after 10, 20 and 40 time steps. The Courant number, $C_r = \hat{u}\hat{k}/\hat{h}$, and Peclet number, $P_e = \hat{h}^2/\hat{k}\hat{D}$, are 0.5 and 25.0 respectively. A Courant number of 0.5 is used because both the numerical dispersion and dissipation of the Holly and Preissmann [20] scheme are maximim. Figure 1 indicates that the operator splitting algorithm simulates advection dominated problem satisfactorily.

5.2. Example 2

The domain and the initial condition are same as in the previous example. The boundary conditions are

$$f^1(\hat{t}) = \exp(-\lambda\hat{t}), \quad \hat{x} = 0, \quad c_{\hat{x}} = 0, \quad \hat{x} \to \infty, \tag{28}$$

where λ is a constant. The analytical solution [22] of equation (1) is

$$c(\hat{t}, \hat{x}) = \frac{1}{2}\exp(-\lambda\hat{t})\left[\exp\left(\frac{(\hat{u} - \alpha)\hat{x}}{2\hat{D}}\right)\operatorname{erfc}\left(\frac{\hat{x} - \alpha\hat{t}}{2\sqrt{\hat{D}\hat{t}}}\right) + \exp\left(\frac{(\hat{u} + \alpha)\hat{x}}{2\hat{D}}\right)\operatorname{erfc}\left(\frac{\hat{x} + \alpha\hat{t}}{2\sqrt{\hat{D}\hat{t}}}\right)\right], \tag{29}$$

with $\alpha = (\hat{u}^2 - 4\lambda\hat{D})^{1/2}$. Figure 2(a) compares the numerical and analytical solutions for $C_r = 0.25$ and $P_e = 25$. The computational parameters are: $\hat{u} = 0.125\ \mathrm{m\,s^{-1}}$, $\hat{D} = 0.0025\ \mathrm{m^2\,s^{-1}}$, $\hat{k} = 0.5$ s,

$\hat{h} = 0.25$ m and $\lambda = 0.1\ s^{-1}$. For $L = \hat{h}$, $T = \hat{k}$, the non-dimensional parameters defined in equation (5) are $u = C_r$, $\epsilon^{-1} = P_e$ and $k = \hat{k}$. Compared with the previous example, a smaller time step is used to resolve the temporal variation of boundary condition (28).

Figure 2(a) indicates that the operator splitting algorithm significantly underestimates the peak concentration and propagates the leading edge faster for $\hat{t} < 40\hat{k}$. The latter tendency can also be observed in Fig. 1. The algorithm simulates the rising limb of the concentration profile, specially the region close to the inflow boundary, quite well. The discrepancy between the analytical and numerical solutions increases as inflection point of a concentration profile is approached. Numerical analysis [20] of the backward method of characteristics indicates that 10 or more grid points per wave length are necessary for an adequate spatial resolution of a problem. In the initial stage of computations, approximately up to $\hat{t} = 20\hat{k}$, this criteria is not satisfied. As a result, damping of the peak concentration and leading edge phase error are considerable.

The results of numerical computations using a simple modification of the present example, so that the initial condition is well resolved, are shown in Fig. 2(b). The initial condition is obtained from equation (29) with $\hat{t} = 20\hat{k}$, when $c(\hat{t}, \hat{x})$ is approximately distributed over eight elements. The numerical and analytical solutions are compared after 20, 40 and 60 time steps. All computational parameters, except the initial conditions, are the same in both the cases. Therefore, a comparison of the results presented in Fig. 2 indicates that errors in Fig. 2(a) are mainly associated with the numerical procedures for solving the split equations. Equations (17), (24) and (25) provide sufficiently accurate boundary conditions for the operator splitting algorithm, which is further illustrated by the following example.

5.3. Example 3

The advection of a Gaussian hump is a frequently used test problem for demonstrating the simulation characteristics of numerical procedures [23]. The analytical solution of equation (1), in an infinite domain, is given by

$$c(\hat{t}, \hat{x}) = \frac{\sigma_0}{\sigma} \exp\left[-\frac{(\hat{x} - \hat{x}_0 - \hat{u}\hat{t})^2}{2\sigma^2} \right], \tag{30}$$

$$\sigma^2 = \sigma_0^2 + 2\hat{D}\hat{t}, \tag{31}$$

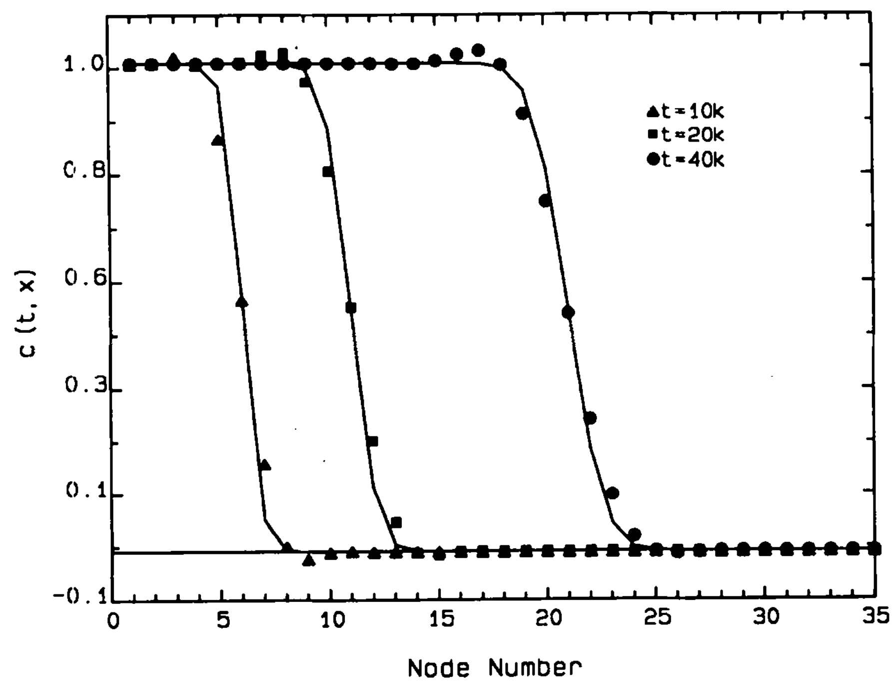

Fig. 1. Comparison of analytical and numerical solutions for example 1. The symbols are numerical solutions and lines are analytical solutions. The results are compared after 10, 20 and 40 time steps.

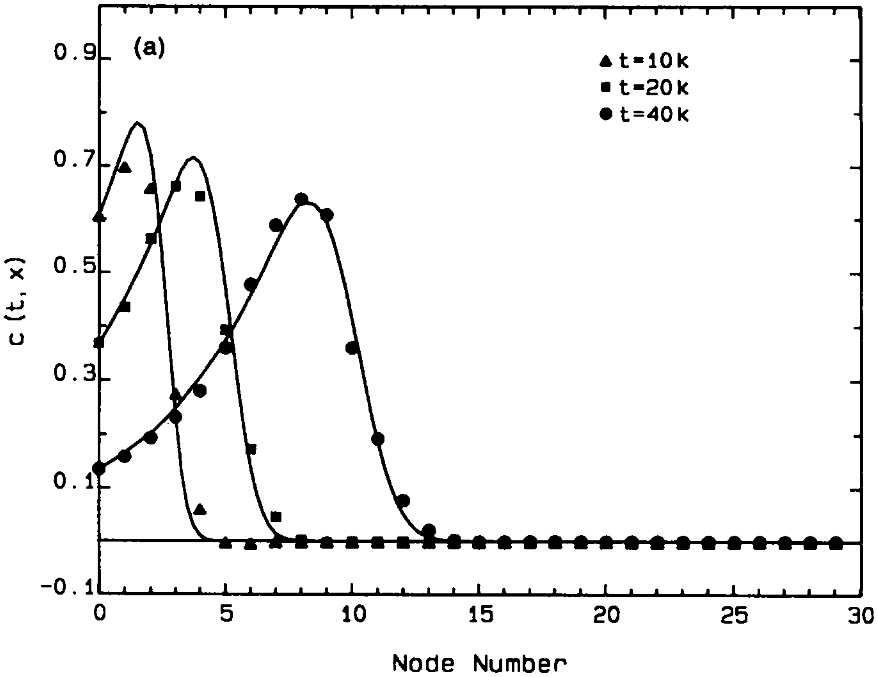

Fig. 2(a). Comparison of analytical and numerical solutions for example 2, with zero initial concentration inside the computational domain. The symbols are numerical solutions and lines are analytical solutions. The results are compared after 10, 20 and 40 time steps.

where σ_0 and $\hat{x}_0$ are the initial standard deviation and the location of the Gaussian hump, respectively. In the present example, following Aiyesimoju and Sobey [8], advection and diffusion of the hump which is initially outside the computational domain is considered. For $\hat{u} > 0$, the hump

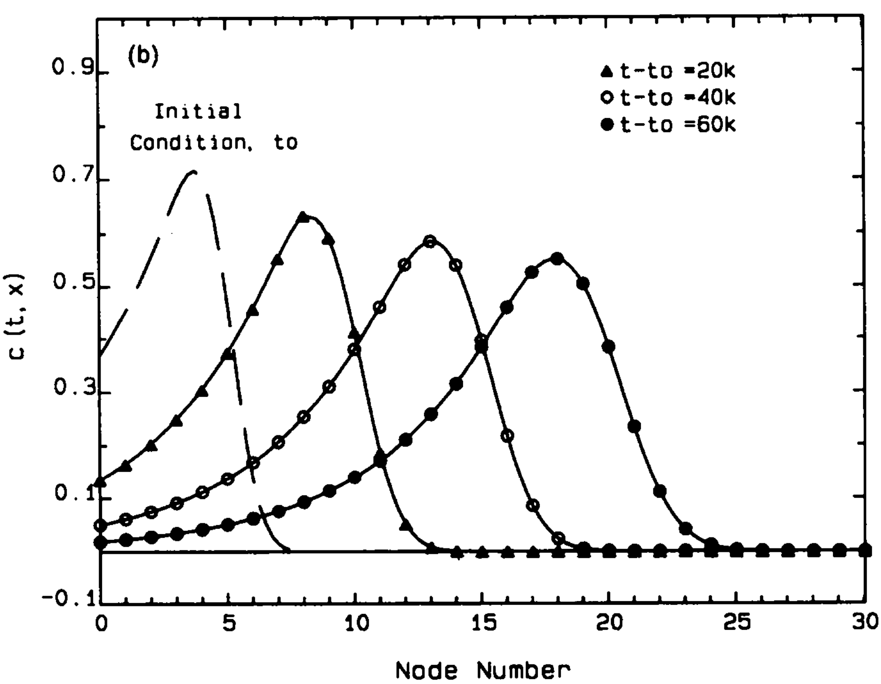

Fig. 2(b). Comparison of analytical and numerical solutions for example 2, with initial condition as indicated in the figure. The symbols are numerical solutions and lines are analytical solutions. The results are compared after 20, 40 and 60 time steps.

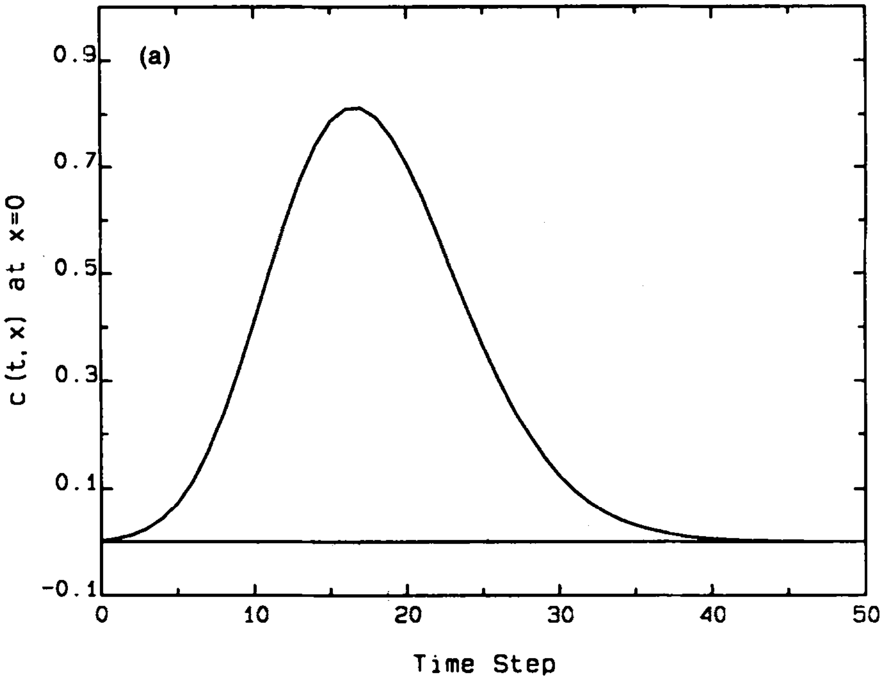

Fig. 3(a). Time-dependent inflow boundary condition for example 3.

is advected into the domain as computations advance in time. The boundary condition at $\hat{x} = 0$ can be expressed as

$$f^1(\hat{t}) = \frac{\sigma_0}{\sigma} \exp\left[-\frac{(-\hat{x}_0 - \hat{u}\hat{t})^2}{2\sigma^2} \right], \tag{32}$$

where $\hat{x}_0 < 0$. The outflow boundary is taken sufficiently far so that the boundary condition in the previous example can be applied. The Gaussian hump (30) with $\sigma_0 = 2.5\hat{h}$ is placed at $\hat{x}_0 = -8\hat{h}$. The value of σ_0 has been selected, based on the analysis of Holly and Preissmann [20], so that the problem is well resolved and the associated numerical dispersion and dissipation are negligible. For computational parameters such that $C_r = 0.5$ and $P_e = 5$, Fig. 3(a) shows the time variation of the inflow boundary condition, and the numerical and analytical solutions are compared in Fig. 3(b). The agreement between the two solutions is quite good.

5.4. Analysis of accuracy

The intermediate boundary conditions presented in Section 4 are at least second-order accurate if $\epsilon = \mathcal{O}(k) \ll 1$. As indicated in Section 3, the operator splitting algorithm used in this study is also $\mathcal{O}(k^2, h^2)$ accurate. Consequently, solutions of initial-boundary value problems by the splitting algorithm and subjected to intermediate boundary conditions (17), (24) and (25) should be second-order accurate. Example 3 is used to verify the accuracy of solution. The computations are performed by refining (k, h) in the following sequence: $(k/2^m, h/2^m)$, $m = 0, 1, \ldots, 3$. At the end of a fixed simulation period, maximum discrepancy (absolute value) between the analytical and numerical solutions is taken as error, $E(k, h)$. Figure 4 shows the plot of $E(k, h)$ vs (k, h) and the best fit line through the points. The slope of the best fit line, indicating the accuracy of solution, is 1.88, which is approximately consistent with the theoretical accuracy of solution.

Computations in example 3 were also performed with different values of P_e varying from 1 to 100. Keeping all other parameters same, $\hat{D}$ was decreased to increase P_e. The computations indicate that results are not very sensitive to the approximations made in deriving boundary conditions (17) and (24) provided $P_e \geqslant 5$. This observation follows from the fact that exact boundary condition is applied at the end of each computational cycle by equation (25). In addition, the first two terms in equations (17) and (24), which are independent of Peclet number, are the dominant terms as P_e increases.

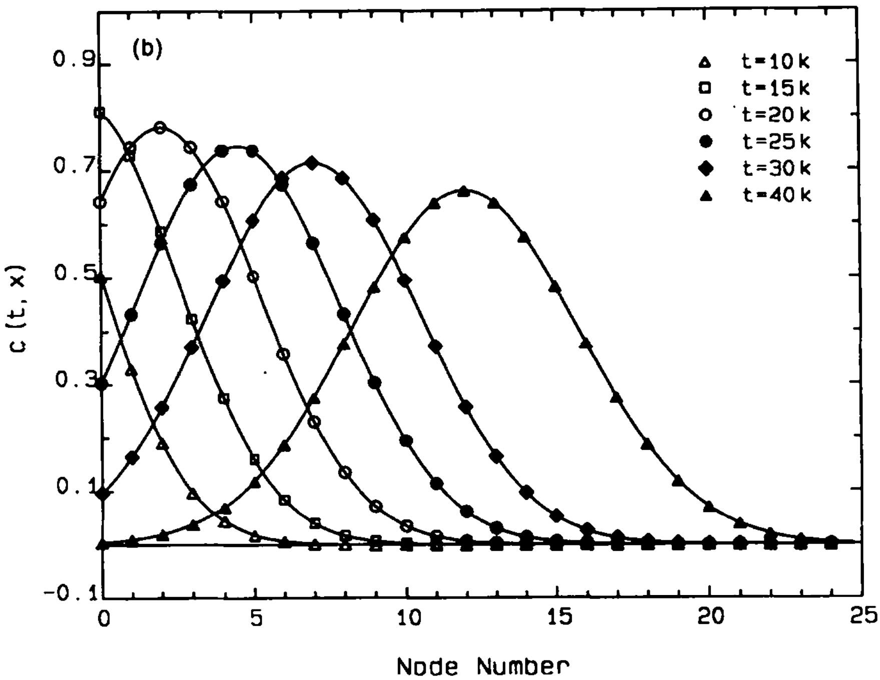

Fig. 3(b). Comparison of analytical and numerical solutions for example 3. The symbols are numerical solutions and lines are analytical solutions. The results are compared after 10, 15, 20, 25, 30 and 40 time steps.

6. SUMMARY AND CONCLUSIONS

A disadvantage of operator splitting algorithms results from the fact that boundary conditions defined by the physical problem are not directly applicable to the split equations. Thus, it is necessary to derive appropriate boundary conditions for the split equations. In the previous studies related to locally one-dimensional methods, boundary conditions for either the advection equation [5, 7] or the diffusion equation [2, 3] have been derived. Therefore, in this paper intermediate

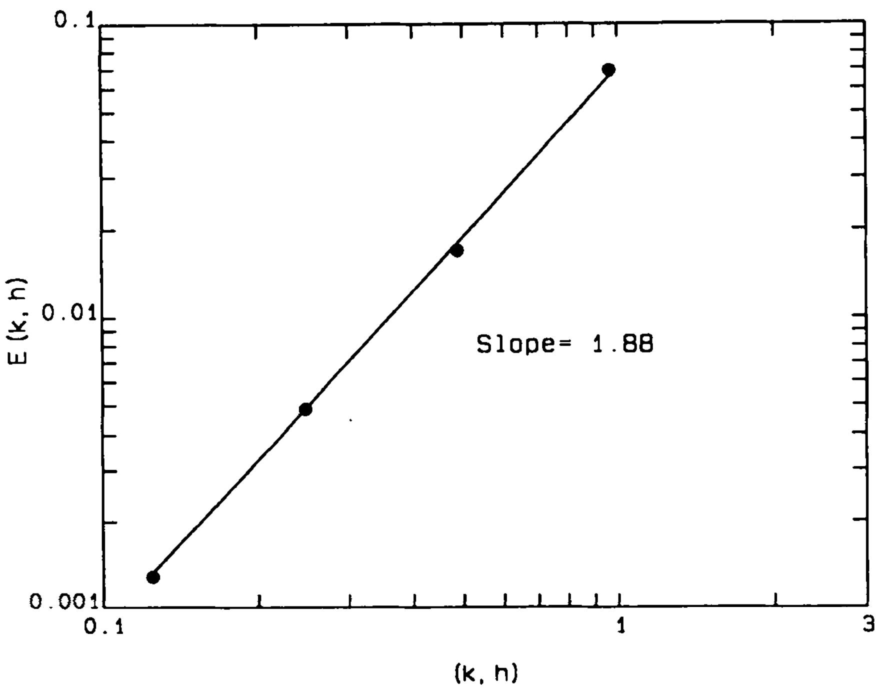

Fig. 4. Accuracy of solution for the initial-boundary value problem defined in example 3.

Dirichlet boundary conditions for Strang type operator splitting algorithms for the one-dimensional advection–diffusion equation have been presented. The derived intermediate boundary conditions are applicable to advection dominated transport problems, and independent of the numerical schemes used to solve the split equations. Therefore, these boundary conditions can be utilized by operator splitting algorithms based on different numerical procedures [11–17].

Acknowledgements—The research reported in this paper was in part supported by a research grant from the Xerox Corporation to Cornell University. During the course of study, LAK has also been supported by a Fellowship from the DeFrees Foundation. This research was conducted using the Cornell National Supercomputer Facility, which receives major funding from the National Science Foundation and IBM Corporation, and with additional support from New York State and members of the Corporate Research Institute.

REFERENCES

1. G. I. Marchuk, *Mathematical Models in Environmental Problems*. North-Holland, Amsterdam (1986).
2. N. N. Yanenko, *The Method of Fractional Steps*. Springer, Berlin (1971).
3. B. P. Sommeijer, P. J. van der Houwen and J. G. Verwer, On the treatment of time-dependent boundary conditions in splitting methods for parabolic differential equations. *Int. J. Numer. Methods Eng.* **17**, 335–46 (1981).
4. A. R. Gourlay and J. Ll. Morris, A multi-step formulation of the optimized Lax–Wendroff method for non-linear hyperbolic systems in two space variables. *Math. Comp.* **22**, 715–20 (1968).
5. G. R. McGuire and J. Ll. Morris, Boundary techniques for the multistep formulation of the optimized Lax–Wendroff method for non-linear hyperbolic systems in two space dimensions. *J. Inst. Maths. Applics.* **10**, 150–65 (1972).
6. A. R. Gourlay and A. R. Mitchell, On the structure of alternating direction implicit (A.D.I.) and locally one dimensional (L.O.D.) difference methods. *J. Inst. Maths. Applics.* **9**, 80–90 (1972).
7. R. J. Le Veque, Intermediate boundary conditions for time-split methods applied to hyperbolic partial differential equations. *Math. Comp.* **47**(175), 37–54 (1986).
8. K. O. Aiyesimoju and R. J. Sobey, Process splitting of boundary conditions for the advection–diffusion equation. *Int. J. Numer. Methods Fluids* **9**, 235–44 (1989).
9. G. Strang, On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.* **5**, 506–17 (1968).
10. R. J. LeVeque and J. Oliger, Numerical methods based on additive splittings for hyperbolic partial differential equations. *Math. Comp.* **40**(162), 469–97 (1983).
11. C. W. Li, Least-square characteristics and finite elements for advection–dispersion simulations. *Int. J. Numer. Methods Engr.* **29**, 1343–58 (1990).
12. C. W. Li, Fractional step solution to slack tide oxygen balance equation. *Comm. Appl. Numer. Methods Engr.* **8**, 93-98 (1992).
13. K. Muralidhar, M. Verghese and K. M. Pillai, Application of an operator-splitting algorithm for advection–diffusion problems. *Numer. Heat Transfer A* **23**, 99–113 (1993).
14. J. Noye, Time-splitting the one-dimensional transport equation. In *Numerical Modeling: Applications to Marine Systems* (Edited by J. Noye), pp. 271–295. North-Holland, Amsterdam (1987).
15. R. Szymkiewicz, Solution of the advection–diffusion equation using the spline function and finite elements. *Comm. Appl. Numer. Methods Engr.* **9**, 197–206 (1993).
16. M. F. Wheeler, W. A. Kinton and C. N. Dawson, Time-splitting for advection-dominated parabolic problems in one space variable. *Comm. Appl. Numer. Methods Engr.* **4**, 413–423 (1988).
17. X. K. Xin and Y. S. Wong, Eulerian–Lagrangian splitting methods for convection dominated equations. *Fluid Dynamic Res.* **5**, 13–27 (1989).
18. L. Demkowicz, T. J. Oden and W. Rachowicz, A new finite element method for solving compressible Navier–Stokes equations based on an operator splitting and h-p adaptivity. *Comput. Methods Appl. Mech. Eng.* **84**, 275–326 (1990).
19. J. B. Perot, An analysis of the fractional step method. *J. Comput. Phy.* **108**, 51–58 (1993).
20. F. M. Holly Jr. and A. Preissmann, Accurate calculation of transport in two-dimensions. *Proc. ASCE J. Hyd. Div.* **103**, 1259–78 (1977).
21. P. J. Roache, *Computational Fluid Dynamics*. Hermosa Publishers, Albuquerque, NM (1972).
22. M. Th. van Genuchten, Analytical solutions for chemical transport with simultaneous adsorption, zero-order production and first-order decay. *J. Hydrology* **49**, 213–33 (1981).
23. P. M. Gresho and R. L. Lee, Don't suppress the wiggles—they're telling you something!, *Computers Fluids* **9**, 223–53, 1981.

APPENDIX

Successive differentiations of equation (7) with respect to t and using the same equation to replace c_t result in the following equations

$$c_{tt} = u^2 c_{xx} - 2\epsilon u c_{xxx} + \epsilon^2 c_{xxxx}, \tag{A1}$$

$$c_{ttt} = -u^3 c_{xxx} + 3\epsilon u^2 c_{xxxx} - 3\epsilon^2 u c_{xxxxx} + \epsilon^3 c_{xxxxxx}, \tag{A2}$$

$$c_{tttt} = u^4 c_{xxxx} - 4\epsilon u^3 c_{xxxxx} + 6\epsilon^2 u^2 c_{xxxxxx} - 4\epsilon^3 u c_{xxxxxxx} + \epsilon^4 c_{xxxxxxxx}, \tag{A3}$$

where u and ϵ are assumed locally constant. From equation (7)

$$u c_x = -c_t + \epsilon c_{xx}, \tag{A4}$$

which can be expressed as follows by using equations (A1)

$$uc_x = -c_t + \frac{\epsilon}{u^2}(c_{tt} + 2\epsilon u c_{xxx} - \epsilon^2 c_{xxxx}). \tag{A5}$$

Replacing c_{xxx} by using equation (A2)

$$uc_x = -c_t + \frac{\epsilon}{u^2} c_{tt} + \frac{2\epsilon^2}{u^4}(-c_{ttt} + 3\epsilon u^2 c_{xxxx} - 3\epsilon^2 u c_{xxxxx} + \epsilon^3 c_{xxxxxx}) - \frac{\epsilon^3}{u^2} c_{xxxx}. \tag{A6}$$

Substituting expression for c_{xxxx} from equation (A3) and keeping terms up to $\mathcal{O}(\epsilon^3)$ gives equation (13) in Section 4. The next relation (14) can be obtained by expressing equation (A1) as

$$u^2 c_{xx} = c_{tt} + 2\epsilon u c_{xxx} - \epsilon^2 c_{xxxx}, \tag{A7}$$

and using equation (A2) and (A3) to replace c_{xxx} and c_{xxxx} respectively.