Flexible Discretization Algorithm for Fixed-Grid MOC in Pipelines

By Bryan W. Karney[1] and Mohamed S. Ghidaoui[2]

Abstract: The problem of selecting a time step that exactly satisfies the Courant condition arises most commonly when the fixed-grid method of characteristics (MOC) is applied to multipipe systems. Traditionally, this problem has been solved by employing one of a variety of interpolation techniques or by allowing small adjustments in the value of wave speed. In this paper, the number of available strategies is increased substantially by allowing several new kinds of interpolation as well as blended combinations of these interpolation approaches with the wave-speed–adjustment technique. The new hybrid approaches include interpolation along a secondary characteristic line and minimum-point interpolation, which minimizes the distance from the interpolated point to the primary characteristic. An intuitively appealing technique involving direct adjustment of the wave path is rejected because it is potentially unstable. The entire flexible algorithm is implemented as a preprocessor step, uses memory efficiently, executes quickly, and provides a flexible tool for investigating the importance of discretization errors in pipeline systems. The properties of the new algorithm are analyzed theoretically and illustrated by example.

INTRODUCTION

For many years the fixed-grid method of characteristics (MOC) has been used with great success to calculate transient conditions in pipe systems and networks. In fact, for pipeline transients this numerical procedure is noted generally for its numerical efficiency, computational accuracy, and programming simplicity (Wylie and Streeter 1993; Almeida and Koelle 1992; Chaudhry 1987). However, one difficulty that commonly arises relates to the selection of an appropriate level of discretization (or time step) to use for the analysis. The obvious trade-off is between computational speed and accuracy: in general, the smaller the time step, the longer the run time but the greater the numerical accuracy.

The challenge of selecting a time step is made difficult in pipeline systems by two conflicting constraints. First, to calculate many boundary conditions, such as obtaining the head and discharge at the junction of two or more pipes, it is necessary that the time step be common to all pipes. The second constraint arises from the nature of the MOC. If the advective terms in the governing equations are neglected (as is almost always justified), the MOC requires that ratio of the distance step Δx to the time step Δt be equal to the wave speed a in each pipe. In other words, the Courant number $C_r = a\Delta t/\Delta x$ should ideally be equal to one and must not exceed one for stability reasons. For most pipeline systems, having as they do a variety of different pipes with a range of wave speeds and lengths, it is impossible to satisfy exactly the Courant requirement in all pipes with a reasonable (and common) value of Δt.

Faced with this challenge, researchers have sought for ways of relaxing the numerical constraints. Two contrasting strategies present themselves. The "method of wave-speed adjustment" changes one of the pipeline properties (usually the wave speed, though more rarely the pipe length is altered) so as to satisfy exactly the Courant condition. Despite the obvious liberties this kind of adjustment takes with the physical problem, this procedure is widely recommended in the pipeline literature (Wylie and Streeter 1993; Chaudhry 1987; Karney and McInnis 1992). The second alternative is to allow Courant numbers less than one and to interpolate between known grid points. The most common methods include linear interpolation at a fixed time level, including both space line interpolation and reach-out in space interpolation (Wiggert and Sundquist 1977), as well as interpolation at a fixed location, such as time line interpolation or reach-back in time interpolation (Goldberg and Wylie 1983). Lai (1989) combined these options to form what he calls the multimode scheme. A number of nonlinear interpolation techniques also have been proposed including the Holly-Preissmann scheme (Holly and Preissmann 1977), Holly-Preissmann with time line interpolation (Liggett and Chen 1994), and a method that uses cubic splines (Sibetheros et al. 1991).

Ghidaoui and Karney (1994) have developed the concept of an equivalent hyperbolic partial differential equation (EHDE) to analyze theoretically the numerical properties of a variety of interpolation schemes. They found that all common interpolation procedures considerably distort the original governing equations and that even interpolation procedures effectively change the wave speed. Thus, Ghidaoui and Karney concluded that it is unlikely that any discretization approach would be ideal for all pipeline systems and for all kinds of disturbances, making it improbable that the ongoing controversy about which discretization method is "best" will be resolved in the near future. However, despite the ubiquitous nature of discretization errors in complex pipeline systems, the authors contend that discretization errors are sometimes the dominant numerical error and may outweigh errors arising from nonlinear terms such as friction or minor uncertainties in the boundary conditions.

Thus, no attempt is made here to end debate about which discretization approach is best for a given problem. Nor does this paper presume to dismiss complex or higher-order methods. However, at least for the constant wave-speed cases considered here, it is logical to explore more fully simple algorithms that may either replace or be used along with more complex strategies. To this end, a flexible and efficient discretization algorithm is presented that can readily be applied to existing fixed-grid MOC codes and allows a variety of discretization choices to be selected conveniently and investigated.

More specifically, this paper uses a generalized algebraic formulation to derive a flexible class of discretization approaches for interpolating between grid points to obtain values of the dependent variables at the foot of characteristic curves.

[1]Prof., Dept. of Civ. Engrg., Univ. of Toronto, Toronto, M5S 1A4, Canada.

[2]Assist. Prof., Dept. of Civ. and Struct. Engrg., The Hong Kong Univ. of Science and Technology, Kowloon, Hong Kong.

Note. Discussion open until April 1, 1998. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on June 19, 1995. This paper is part of the ***Journal of Hydraulic Engineering***, Vol. 123, No. 11, November, 1997. ©ASCE, ISSN 0733-9429/97/0011-1004–1011/$4.00 + $.50 per page. Paper No. 10938.

The new approaches include interpolation along a secondary or intersecting characteristic line, minimum point interpolation that uses the shortest distance from the interpolated point to the primary characteristic, and a method of wave-path adjustment that, like wave-speed adjustment, distorts the path of propagation, but does not directly change the wave speed. In addition, any of these techniques can be blended with the method of wave-speed adjustment. The performance of the new algorithm is studied theoretically, using the equivalent EHDE approach, and practically, using a numerical example of a transient disturbance in a realistic pipeline system. Thus, the flexible interpolation algorithm is shown to be a convenient tool for investigating interpolation errors.

NATURE OF THE INTERPOLATION PROBLEM

Before developing a flexible interpolation algorithm, it is useful to consider the specific nature of the discretization problem when using the fixed-grid MOC to analyze a general pipeline system. First, it is emphasized that practical interpolation methods, at least for the constant wave-speed case, must be computationally efficient, because the interpolation errors can be made progressively smaller by successive reductions in the time step. Moreover, a small time step may be preferable because it reduces other numerical errors, such as those arising from fluid friction or the representation of boundary conditions. However, small time steps may be computationally impractical, particularly in complex systems, because of either memory limitations or prohibitive execution times.

As an aside, it should be pointed out that short pipes may sometimes be more conveniently modeled by ignoring their compressibility effects entirely, treating these links as either incompressible or quasi-steady conduits within the larger MOC simulation (Wylie and Streeter 1993; Karney and McInnis 1992). Because the choice of time step is now independent of these links, this approach is clearly expedient from a computational point of view. In essence, the modeler ignores the wave travel time in these conduits, effectively adjusting their wave speed to infinity. However, even if the shortest possible conduits are eliminated by this approach, the discretization problem remains for all those pipes still included in the MOC simulation. It is to those pipes that remain within the MOC computational framework that the remainder of this paper is addressed.

The relationship between discretization errors in the MOC and the time step—or, equivalently, the number of reaches—is made clear in Fig. 1. This figure indicates the Courant number C_r as a function of the ideal number of reaches $R = L/(a\Delta t)$ in a pipeline segment of length L where R is a real number. But this simple relation highlights the key discretization problem, for the ideal number of reaches R is almost invariably noninteger, whereas the number of reaches N in each pipe must be integer. To avoid unstable Courant numbers C_r with values greater than one, while simultaneously limiting interpolation distances, it is logical to select $C_r = [R]/R$ in which $[R] = N$ is the floor of R. However, the shift from R to N can represent a large proportional change, particularly when R is small. For example, if $R = 1.99$, and thus $N = 1$, then the number of reaches is almost 50% smaller than that required to satisfy exactly the wave travel time.

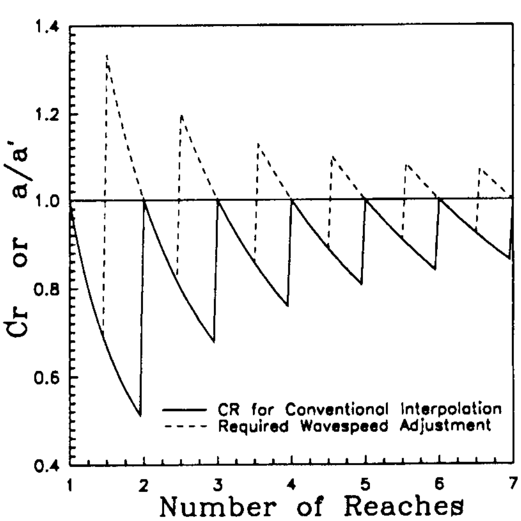

FIG. 1. Courant Number and Wave-Speed Adjustment as a Function of the Ideal Number of Reaches

Ghidaoui and Karney (1994) have shown that discretization errors generally grow with $|1 - C_r|$; thus it is not surprising that short pipes with small R are often problematic. Yet the diminishing saw-tooth form of Fig. 1 also illustrates that a small time step with an associated large number of reaches R can always be found, which reduces interpolation errors to negligible values. More precisely, the departure of the Courant number from one is bounded by $1/R$. Clearly, then, any reasonable interpolation technique using a small time step will likely perform well. However, in complex systems the computational price for such a large number of reaches can be high because not only do execution times for a given duration of simulation increase roughly as the square of R, but the convergence to $C_r = 1.0$ is so irregular that modest reductions in the time step may actually temporarily increase the discretization error. Moreover, discretization approaches must generally be viable with single-reach pipes. The rather abrupt and discontinuous convergence to $C_r = 1.0$ shown in Fig. 1 underlies all discretization problems in realistic pipeline systems, greatly complicating numerical investigation.

The second (dashed) curve in Fig. 1 indicates the relative distortion in the wave speed as a function of the ideal number of reaches. Because the wave-speed–adjustment approach exactly satisfies the Courant condition, it is permissible to round off the number of reaches during the calculation for the adjusted wave speed: thus, $a/a' = [R + 0.5]/R$, where a' is the adjusted wave speed. As a result, the relative adjustment in wave speed is bounded by $1/2R$, half of the value associated with pure interpolation. Moreover, the fact that this technique may either increase or decrease the wave speeds means that there is a greater chance of offsetting errors between different parts of a pipeline system, thereby possibly decreasing the overall error. For particular systems, the effect of rounding may be important because it can greatly reduce the required adjustment. For example, suppose again that $R = 1.99$; the required wave-speed adjustment for $N = 2$ is minimal, equal to an adjustment of about 0.5% of the original wave speed when two reaches are used, whereas as we have shown the required interpolation $N = 1$ is quite large. One of the advantages of the hybrid approach presented next is that it can preserve the desirable flexibility of the rounding approach when using alternative interpolation strategies.

FLEXIBLE DISCRETIZATION ALGORITHM

The basic approach suggested here is to extend the characteristic back for an entire spatial step, as illustrated in Fig. 2 for the C^+ characteristic. The point at which this characteristic is exactly Δt before the current point (i.e., point 1 in Fig. 2) determines the usual space line point, whereas the point exactly Δx away (point 2) determines the usual time line point. However, because errors are associated with moving information from the grid points to the characteristic (i.e., interpolation), intermediate points such as point 3 in Fig. 2 may sometimes represent a better compromise.

Once this basic observation is made, the details of a more general approach quickly follow. If it is desired to interpolate

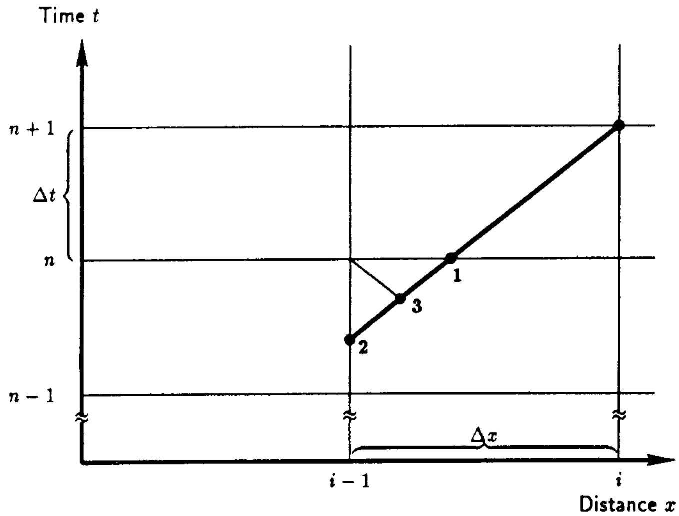

FIG. 2. Generalized Single-Reach Interpolation Approaches

in both space and time, one need only determine the contribution or weight of the four nearest grid points surrounding the interpolated point in question. Borrowing the form of a linear basis from finite elements, a powerful interpolation function can be written as

$$\phi = \sum_{i=0}^{3} W_i \phi_i \tag{1}$$

where ϕ = general interpolated variable (either head H or discharge Q); ϕ_i = same variable at the adjacent grid points; and W_i = associated weight. If the nondimensional interpolation variable in the distance direction is ξ and that in the time direction is ζ, then the equation can be written more explicitly for the C^+ characteristic leading to grid point (n + 1, i) as

$$\phi = (1 - \xi)(1 - \zeta)\phi_{i-1}^n + \xi(1 - \zeta)\phi_i^n + (1 - \xi)\zeta\phi_{i-1}^{n-1} + \xi\zeta\phi_i^{n-1} \tag{2}$$

where the superscript n = temporal location; and the subscript i = spatial location of the associated grid points. Note that if no interpolation is required, $\xi = \zeta = 0$, and thus $\phi = \phi_{i-1}^n$. Similar limiting values can be obtained at the other corners of the interpolated grid. Thus, for space line interpolation $\zeta = 0$ while $\xi = 0$ produces the appropriate weighting for time line interpolation. It is a simple matter to write the expressions for ζ and ξ in terms of Courant number for any selected interpolation procedure. Once the interpolated values of head and discharge have been obtained, the MOC solution can proceed as shown in many references (Wylie and Streeter 1993; Chaudhry 1987; Karney and McInnis 1992).

Other than the traditional space line and time line interpolation, a number of other interpolation procedures suggest themselves. Two logically appealing alternative interpolation approaches include minimum-point interpolation, which selects the point on the characteristic that lies closest to the primary MOC grid point ϕ_{i-1}^n, and characteristic line interpolation, which interpolates in the direction of an intersecting characteristic. This final interpolation is particularly attractive because it joins the grid point to the required characteristic with a line segment that is itself a characteristic. Both of these approaches can be selected by simply choosing appropriate values of the corner weights. Moreover, once the basic procedure has been determined, further refinements are easily made. For example, wave-speed adjustments can be viewed as an offset of the Courant number before performing any interpolation, thus effectively rotating the characteristic curve to its adjusted slope prior to interpolation. After this, interpolation can take place using any reasonable strategy along the shifted characteristic. Such a hybrid approach allows a smooth and gradual transition between interpolation procedures and the wave-speed–adjustment approach.

Before presenting the details of the algorithm, one other approach to the discretization problem is worth mentioning. Rather than adjusting the wave speed, why not simply set the two interpolated parameters to zero, thus taking conditions at the foot of the characteristic to be equal to those at the nearby grid point? Thus, we may be tempted to simply set $\phi = \phi_{i-1}^n$ and not to calculate an adjusted wave speed. This wave-path–adjustment approach preserves the magnitude of the pressure wave caused by sudden flow stoppage but distorts the speed at which this information is transmitted to the remainder of the pipe system. However, numerical testing has indicated that in some systems instability can arise with this method. For this reason, the wave-path–adjustment approach is not recommended nor is it considered further here.

The general algorithm that combines all of these options into a single preprocessor step is written for compactness in standard C code and as shown in Fig. 3. This code requires specification of the pipe length (L), wave speed (A), and number of reaches in the shortest pipe (NRSP) where the shortest pipe has the minimum wave travel time L/a. The time step is set by the wave travel time for one reach in this shortest pipe, thus ensuring that all the pipes will have at least one reach. The algorithm is controlled using two primary parameters: INTERP, which selects the interpolation approach, and PSIMAX, a user-specified limit on the maximum allowable wave speed adjustment. As written, the parameter PSIMAX is used preferentially to avoid interpolation if possible and to preserve the desirable near-miss property of the round-off approach. If the nearest grid point is determined to be within striking distance of the allowable wave-speed adjustment, interpolation is avoided. If PSIMAX is set to zero, no wave-speed adjustment is allowed and the discretization weights are calculated using one of the pure interpolation approaches.

Fig. 4 illustrates the basic behavior of this hybrid approach. This graph displays the same data as Fig. 1, but now the pair of lines apply simultaneously to a single system. The solid line indicates the interpolation that remains after first using a maximum allowable wave-speed adjustment of 10%; the dashed line shows the required adjustment in wave speed. Note that with the 10% limit, no interpolation is required as long as the required number of reaches R exceeds about 4.5. Moreover, the bound of either interpolation or wave-speed adjustment is considerably less than if either strategy is used exclusively.

The secondary parameter CRTL in the algorithm is used to set a time-line threshold for the courant number, thus illustrating some of the additional flexibility that can be incorporated into the hybrid approach. If the calculated Courant number in any pipe is close to 0.5, it is clear that the minimum interpolation approach should be based on time-line interpolation, because the foot of the (say) C^+ characteristic will pass near to the point (i − 1, n − 1). Thus, by appropriately choosing CRTL, it is possible to force certain pipes to use time line interpolation, even if some other interpolation approach is being used for other pipes. In this case, it is logical to have the wave speeds of these pipes adjusted toward the previous time step (i.e., toward 0.5), with the switch criterion based on the Courant number. A value such as CRTL = 0.55 appears to work well in most circumstances. The extreme range of CRTL values is 0.50–1.0.

The attributes of the algorithm presented in Fig. 3 are summarized as follows:

- It can be easily applied to existing code
- The new algorithm is applicable to all pipes in a system, including those having only one reach

```
/* INTERP   Interpolation Method    | INTERP   Interpolation Method
      0  =  wavespeed adjustment    |   3   =  'minimum-point'
      1  =  space-line              |   4   =  'characteristic line'
      2  =  time-line               |   5   =  'wave path' adjustment
   Step 1: Find minimum wave travel time */
    R[1] = TTMIN = L[1]/A[1];         /* travel time = pipe length/wavespeed */
    for (I=2; I<=NP; ++I)
       if ( (R[I] = L[I]/A[I]) < TTMIN)  TTMIN = R[I];
/* Step 2: Find time step as wave travel time in 'stortest' pipe */
redisc:
     DT = TTMIN/NRSP;          /* NRSP = number of reaches in 'shortest' pipe */
/* Step 3: Find weights for each grid point for each pipe */
    for (I=1; I<=NP; ++I) {
       R[I] /= DT;                          /* Exact (real) number of reaches */
       N[I] = R[I]  + 0.500;          /* Rounded (integer) number of reaches */
       if ( (CR = N[I]/R[I]) > 1.001/(1. - PSIMAX) ) CR = --N[I]/R[I];
           /* Shift CR (maximum allowable change in A controlled by PSIMAX) */
       PSI = PSIMAX * CR;       /* PSI = allowable shift in Courant Number CR */
       if (fabs(1-CR) <= PSI) CR = 1.0;  /* If possible, avoid interpolation */
       else if (INTERP == 0) {              /* Wavespeed change is too large */
          NRSP++;             /* Increase number of reaches in shortest pipe */
          goto redisc;}           /* Hence, try again with shorter time step */
       else if (CR <= CRTL + PSI) CR -= PSI;         /* Closer to earlier step */
       else  CR += PSI;                    /* Increase CR as much as possible */
       A[I] = CR*L[I] / (N[I]*DT);              /* Obtain 'adjusted' wavespeed */
       /* Obtain corner point weights based on CR and INTERP */
       ZETA[I] = XI = 0.0;                  /* Default values for interpolation */
       if (INTERP == 0 || INTERP == 5) ;                    /* Accept defaults */
       else if (INTERP == 1 && CR > CRTL)  ZETA[I] = 1 - CR;   /* Space Line */
       else if (INTERP == 3 && CR > CRTL) {                 /* 'Minimum Point' */
          ZETA[I] = (1 - CR)/(1+pow(CR,2)); XI = CR*ZETA[I]; }
       else if (INTERP == 4 && CR > CRTL) {          /* Characteristics Line */
          ZETA[I] = 0.5*(1 - CR); XI = (1.0-CR)/(2.0*CR);}
       else  XI = (1.0-CR)/CR;           /* Hence, use Time Line interpolation */
       W[I][0] = (1 - ZETA[I]) * (1 - XI);
       W[I][1] = (1 - ZETA[I]) *    XI;      /* Thus, knowing distances, ...  */
       W[I][2] =     ZETA[I]    * (1 - XI);  /* ... find required weights.    */
       W[I][3] =     ZETA[I]    *    XI;
       }
```

FIG. 3. C Code for Generalized Interpolation and Wave-Speed–Adjustment Algorithm

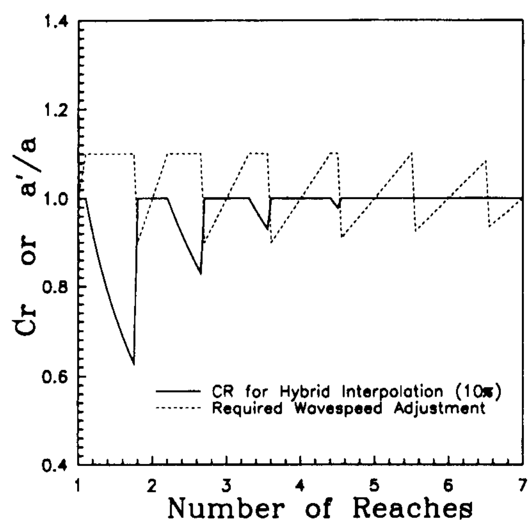

FIG. 4. Hybrid Discretization Combining Wave-Speed Adjustment and Interpolation

- It couples two traditional interpolation strategies and three new ones, with a controllable blend of the wave-speed–adjustment technique
- The procedure is a preprocessing step, so there is no run-time cost to the flexibility during the simulation of transient events; all weights and adjustments are calculated only once for each pipeline in the system
- The term ZETA relating to the spatial interpolation is obtained during the discretization loop and can be stored to scale properly the integration of the friction term
- The use of wave-speed adjustment allows rounding, thus possibly reducing error in individual pipes and decreasing the overall error in wave travel time

Note that the demonstrated flexibility of the algorithm is not exhaustive. For example, by adding an additional subscript it is simple to index the interpolation approach to each pipe, allowing different approaches to be used effortlessly in different parts of the system. Again it is emphasized that there is essentially no run-time cost to this flexibility. Furthermore, one could easily improve the automatic rediscretization test in step 3 to include a number of other criteria, such as testing whether the degree of interpolation is too large, even allowing for wave-speed adjustments. For example, one could test whether $|\text{PSIMAX} + (1 - C_r)| \le \text{TOL}$ where TOL is some user-specified tolerance; if the test fails, the number of reaches could be increased, exactly as the displayed algorithm does if the required wave-speed adjustment is too large.

EHDE FOR FLEXIBLE ALGORITHM

An additional advantage of the flexible algorithm approach developed here is that it allows simplification not only to the numerical calculations, but also to its associated theoretical analysis. Thus, the EHDE derivation shown in this section combines several sets of equations presented in Ghidaoui and Karney (1994) into a single collection of compact expressions. These composite forms reduce to the previously presented