Accommodating Linear and Nonlinear Boundary Conditions in Wave Digital Simulations of PDE Systems

Cathy Qun Xu *Steven C. Bass* *Xiaoming Wang*
Dept. of Computer Science & Engineering, University of Notre Dame
Notre Dame, IN 46556

The Wave Digital multidimensional discretization technique, recently proposed by A. Fettweis *et al*, is a potentially important new method for simulating systems of partial differential equations (PDE's), especially those that model processes appearing in nature. To date, no general method has appeared in the literature indicating how to accommodate boundary conditions in Wave Digital simulations. Since the incorporation of a consistent set of boundary conditions within a given PDE system is a necessary condition for that system even to possess a unique solution, it is clear that accounting for boundary conditions within numeric simulations is just as necessary. We present here a method for accommodating lumped, linear or nonlinear boundary conditions into the Wave Digital simulation of either linear or nonlinear PDE systems. Graphic results from the Wave Digital simulation of a simple acoustics problem are also given.

1 Introduction

Systems of partial differential equations (PDE's) are used throughout engineering and science to describe many phenomena that occur in nature, including acoustics and speech production, electromagnetic wave propagation, fluid flow, electronic device operation and reliability, weather analysis, and others. It is therefore desirable to simulate the behavior of such PDE systems as cheaply and rapidly as possible.

A remarkable new approach to PDE system simulation, discovered by A. Fettweis, has appeared in the recent literature [2]-[10]. It exhibits second-order accuracy, and because it has its origin in the field of digital signal processing, this approach has been termed "multidimensional wave digital filtering," though its action is to simulate, not to filter.

To help describe an important advantage of this fundamental new approach to PDE system simulation, consider the conceptually simplest situation in which a single processor (a very small computer) – which might well be one of millions of such processors – is devoted solely to the computations associated with one (and only one) discrete-space point in a grid spanning the overall simulation space. At each instant of simulation time, it is the job of such a processor to predict the values of the dependent PDE variables at its point in space at the *next* simulation time instant.

It turns out that a benefit of the multidimensional wave digital filter approach of Fettweis is that each such processor calculates its future values using only current values of these dependent variables at its own spatial location, *plus* current values of these quantities at its nearest-neighbor processors; that is, those processors dedicated to computing these same dependent-variables at nearest-neighbor points in multidimensional space [2, 3, 4].

Other second-order finite-difference methods are not known to provide this advantage: That is, unless one restricts oneself to first-order techniques, inter-processor communications beyond nearest-neighbors are required. For example, the widely-used Adams-Bashforth algorithm provides second-order accuracy (as does the wave formulation), but is known to be a "two step" method [13]. Thus, each processor would require nearest-neighbor communications *plus* connections with processors that are two grid distance units away. This not only increases the design complexity of the corresponding processor arrays, it could also increase communication times by as much as a factor of three to four since signal interconnections are known to be the single most limiting factor with respect to performance and circuit density in VLSI circuits [26].

Implied in what we have stated above is the fact that PDE system simulations based on wave digital formulations allow full computational parallelism; that is, except for the exchange of a limited amount of known nearest-neighbor state information early on, the wave digital calculations necessary to advance a grid point's dependent variable values by one time step do not interact in any way with the corresponding calculations taking place at other grid points.

Both these attributes – nearest-neighbor connections and full parallelism – make the wave digital method an attractive candidate for implementation in massively parallel VLSI architectures.

The wave digital method is also known to yield A-stable simulation algorithms. Furthermore, wave digital filter simulation methods can provide these stability/passivity benefits (termed "robustness" by Fettweis [7]) in a fully multidimensional sense: In particular, recursion stability can be provided by this new method in all *spatial* dimensions, as well as the temporal direction [2], [6]-[9]. Especially appealing is the fact that wave simulations can inherit these stability properties from the physical passivity that the corresponding continuous space-time systems naturally possess.

The WD method can be applied directly to *hyperbolic* PDE's only, although it can be extended to *parabolic* and *elliptic* cases, as well.

But, of course, no PDE problem is well-defined until an appropriate set of initial and boundary conditions have also been specified. Incorporating initial conditions into wave digital PDE simulations was described in [11] and [14]. In this article, we will detail a general method for accommodating lumped boundary conditions into WD simulations of PDE systems. (By

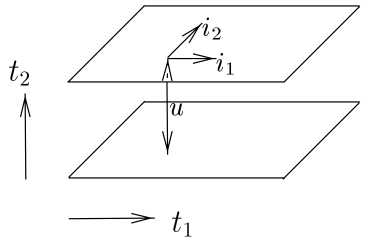

Figure 1: Parallel conducting plates

"lumped," we mean boundary conditions in which derivatives with respect to only time may appear.) These boundary conditions may be either linear or nonlinear.

At the heart of this new method, we will propose a simple "equation counting" technique to determine in any specific case if a given boundary condition may be accommodated at a given grid boundary point. For those situations in which a condition cannot be incorporated into a PDE simulation, we will show what corrective actions may be taken.

Both of these examples involve the simulation of a simple three-dimensional electrical parallel plate problem; they differ in their choices of the underlying sampling grid used. The first example makes use of a pair of mutually-offset rectangular grids (termed a "checkerboard" sampling pattern in [3]), while the second example carries out its calculations on a "densest-ball-packing" grid. (Which, in two spatial dimensions, has a hexagonal or "honeycomb" appearance. See [5] for discussions of this latter grid type in higher or lower dimensions.) In both examples, only piecewise-linear boundaries having segments parallel or orthogonal to the coordinate axes are considered.

These two examples will bring out some of the difficulties that can arise at certain grid boundary points when attempting to satisfy simultaneously both the given PDE's *and* a boundary condition. And it will be in the context of these examples that we will indicate ways of overcoming these difficulties.

After studying these examples, it will be a straightforward matter to summarize a general procedure for dealing with lumped boundary conditions that may be either linear or nonlinear.

Graphical results of simulations of several simple acoustics problems will also be given.

2 Previous Work

We consider the example, as in [2] and [3], of two parallel conducting plates (possibly lossy) separated by a dielectric (possibly also lossy), shown in Figure 1. In this section, we will review the "checkerboard" and "densest-ball-packing" method of wave digital (WD) simulation of this 3-dimensional parallel plate problem given by [3] and [2], respectively.

The partial differential equations representing this parallel plate problem can be written in the form

$$D_3(li_1) + ri_1 + D_1(r_3 i_3) = e_1(\mathbf{t}) \tag{1}$$

$$D_3(li_2) + ri_2 + D_2(r_3 i_3) = e_2(\mathbf{t}) \tag{2}$$

$$D_1 i_1 + D_2 i_2 + D_3(r_3 c i_3) + r_3 g i_3 = e_3(\mathbf{t}) \tag{3}$$

Where t_3 corresponds to time, t_1 and t_2 are the two spatial coordinates, and

$$\mathbf{t} = (t_1, t_2, t_3)^T \tag{4}$$

$$\mathbf{D} = (D_1, D_2, D_3)^T = (\frac{\delta}{\delta t_1}, \frac{\delta}{\delta t_2}, \frac{\delta}{\delta t_3})^T \tag{5}$$

l, c, r and g are given parameters that satisfy $l > 0, c > 0, r \geq 0, g \geq 0$, and that r and g may be functions of $\mathbf{t}$. Finally, i_1 and i_2 are current densities in the direction of t_1 and t_2, respectively, while u is the voltage between the two plates and

$$u = r_3 i_3, \tag{6}$$

r_3 being an arbitrary auxiliary positive quantity, and i_3 is thus defined by this equation.

2.1 "Checkerboard" Sampling

This WD simulation method of the 3-dimensional parallel plate problem is given in [3]. Since the sampling pattern used here resembles a checkerboard, this WD method (algorithm) is referred as "checkerboard" method.(This sampling pattern will be described more fully below.)

As described in [2] and elsewhere, one of the first steps in applying the WD discretization method to a system of PDE's is the derivation of a corresponding "reference circuit". This circuit is constructed in such a way that its equations of motion (i.e., its loop and/or node equations) are precisely the original partial differential equations. In this circuit, the counterparts of the original dependent variables are the circuit loop currents and/or node voltages. Thus, these currents and voltages are functions not only of time, but of the various spatial dimensions also. The elements appearing in reference circuits are typically resistors, inductors, capacitors, ideal transformers, and gyrators. Mathematically, a linear resistor in such a circuit would multiply the multidimensional current passing through it by a resistance value, to arrive at the (multidimensional) voltage appearing across it. A nonlinear resistor would impose a more general relation between its current and voltage, but these “electrical” variables would still be multidimensional. Multidimensional “reactive” elements in reference circuits are especially interesting. The voltage across a linear inductor, for example, is still proportional to the derivative of its current; however, the differentiation is a ”directional derivative” taken in multidimensional space. The derivative direction must be specified for each inductor, just as one would also specify each inductor’s inductance value. Multidimensioanl capacitors would behave in a dual manner, and nonlinear, multidimensional reactive circuit elements can be expected to arise in the reference circuits corresponding to nonlinear systems of PDE’s.

In [1], the well-known methods for transcribing one-dimensional continuous-time reference circuits into wave digital networks are summarized. These networks may be viewed as numeric algorithms that simulate the behavior of the original continuous-time circuit in discrete-time. The underlying discretization process is based on the trapezoidal rule, and – unique to the WD methods – the quantities being processed by the simulating network are incident and reflected “waves”, rather than the original dependent variables appearing in the reference circuit. The remarkable results published by Fettweis, *et al* in [2, 3] (and elsewhere) demonstrate that these same transcription methods may be applied to multidimensional reference circuits, as well. Thus, one may not only construct wave digital “filters” that mimic the behavior of continuous-time signal processors (in discrete-time), it is now also possible to construct wave digital algorithms that simulate the behavior of multidimensional processes, processes modelled by systems of partial differential equations.

A reference for our parallel plate PDE system, $(1-3)$, was developed in [3], and a modified version of this is shown in Figure 2. The loop currents i_1, i_2, and i_3 represent the corresponding dependent variables in $(1-3)$, while the voltages across the inductors will match the directional derivatives of these dependent variables. The reference circuit of Figure 2 may be transcribed into the wave digital network of Figure 3. Again, see [1] (or [11]) for details of the WD transformation method.

In Figure 2, δ_1 and δ_2 are arbitrarily chosen constants such that $\delta_1 > 0$, $\delta_2 > 0$. In this figure, a notation such as $D_3((l-\delta)\cdot)$ etc., indicates that the voltage across the corresponding inductor is equal to $\frac{\delta}{\delta t_3}((l-\delta_1)i)$, where i is the multi-dimensional current flowing through that inductor.

As is the case in classical (one-dimensional) WD theory, every inductor in the reference circuit is modeled as a delay element, together with a sign reversal, in the WD network, and every resistor (possibly together with a series voltage source) is modeled as a sink. For example, the inductor labeled by $D_3((l-\delta_2)\cdot)$ in the reference circuit of Figure 2 corresponds to the highlighted register in the WD network of Figure 3. The delay is of step size $2T$ in the t_3 direction, and its corresponding reference resistance is R_9.

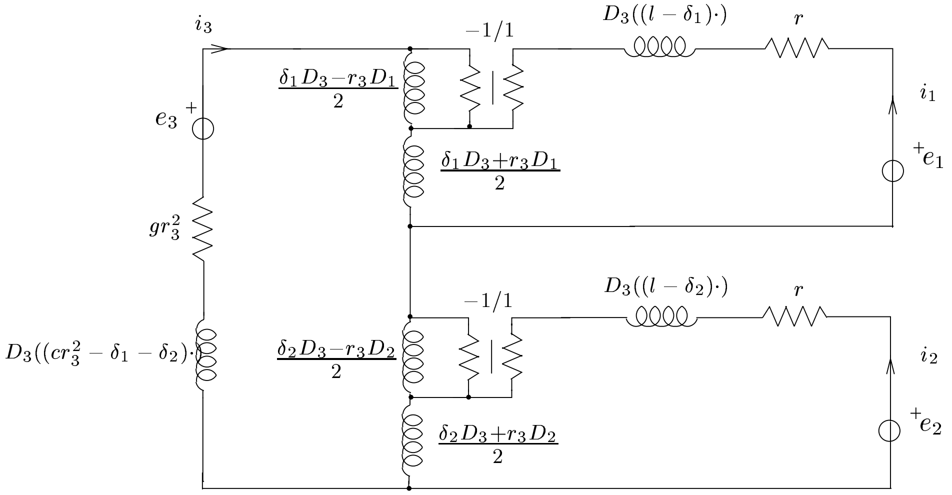
i_3
$-1/1$
$D_3((l-\delta_1)\cdot)$
r
$\frac{\delta_1 D_3 - r_3 D_1}{2}$
e_3 $+$
i_1
$\frac{\delta_1 D_3 + r_3 D_1}{2}$
$+e_1$
gr_3^2
$-1/1$
$D_3((l-\delta_2)\cdot)$
r
$D_3((cr_3^2-\delta_1-\delta_2)\cdot)$
$\frac{\delta_2 D_3 - r_3 D_2}{2}$
i_2
$+e_2$
$\frac{\delta_2 D_3 + r_3 D_2}{2}$

Figure 2: Equivalent reference circuit for the parallel plate PDEs

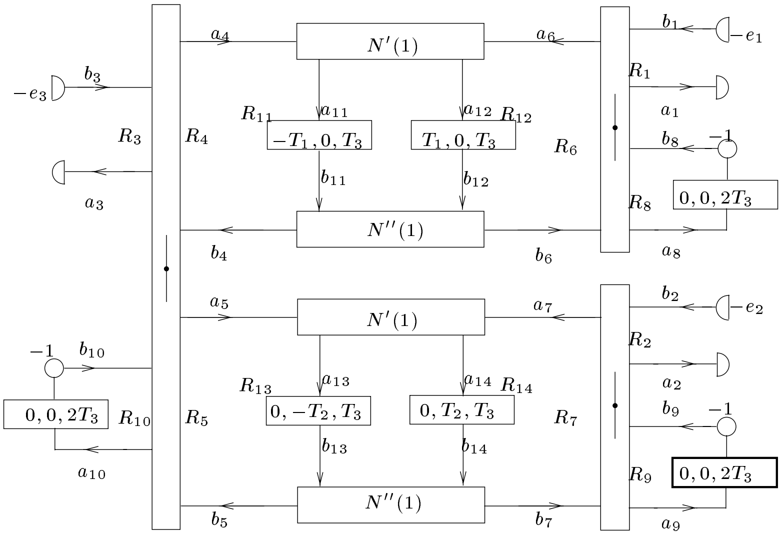

Figure 3: Wave digital network for the parallel plate PDEs assuming checkerboard sampling

The port resistances (R_i) used in Figure 3 were given in [3], where we have assumed that

$$\delta_1 = \delta_2$$

and

$$r_0 = 2\delta_k/T_3 = 2r_3/T_k, \quad r_0' = (l - \delta_k)/T_3, \quad k = 1, 2. \tag{7}$$

$$r_3' = (cr_3^2 - \delta_1 - \delta_2)/T_3. \tag{8}$$

This leads us to the port resistances used in [3]:

$$R_1 = R_2 = r, \quad R_3 = gr_3^2 \tag{9}$$

$$R_4 = R_5 = R_6 = R_7 = r_0 \tag{10}$$

$$R_8 = R_9 = r_0', \quad R_{10} = r_3' \tag{11}$$

$$R_{11} = R_{12} = \frac{1}{2}R_4 \tag{12}$$

$$R_{13} = R_{14} = \frac{1}{2}R_5 \tag{13}$$

The three-port and four-port serial adaptors are given by [1]. The $N'(1)$ and the $N''(1)$ subnetworks are given in [2]. All these network blocks are simple interconnections of adders and multipliers.

As we have seen from Figure 3, the calculation at a grid point requires information from two previous discretization steps in the t_3 direction (see the highlighted box in Figure 3.) and also single steps in either the t_1 and t_3 or t_2 and t_3 directions, as well. We now draw the sampling grid for the parallel plate problem and show its data dependencies in the t_1 and t_2 plane in Figure 4. Calculations are performed alternately on unshaded grid points and then on shaded grid points. Figure 4 also shows how these calculations require previously calculated information from their neighboring grid points. As mentioned above, a grid point calculation also makes use of information from its own previous calculation, carried out two time steps earlier.

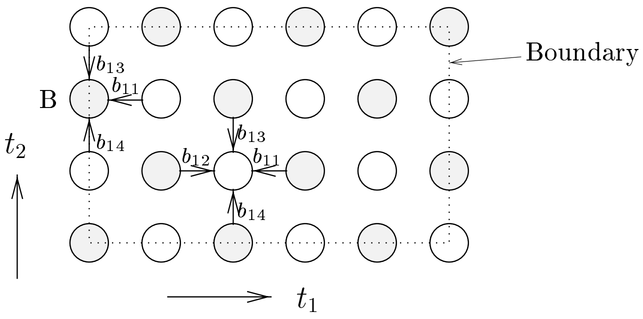

Figure 4: A checkerboard sampling grids for the parallel plate PDE's

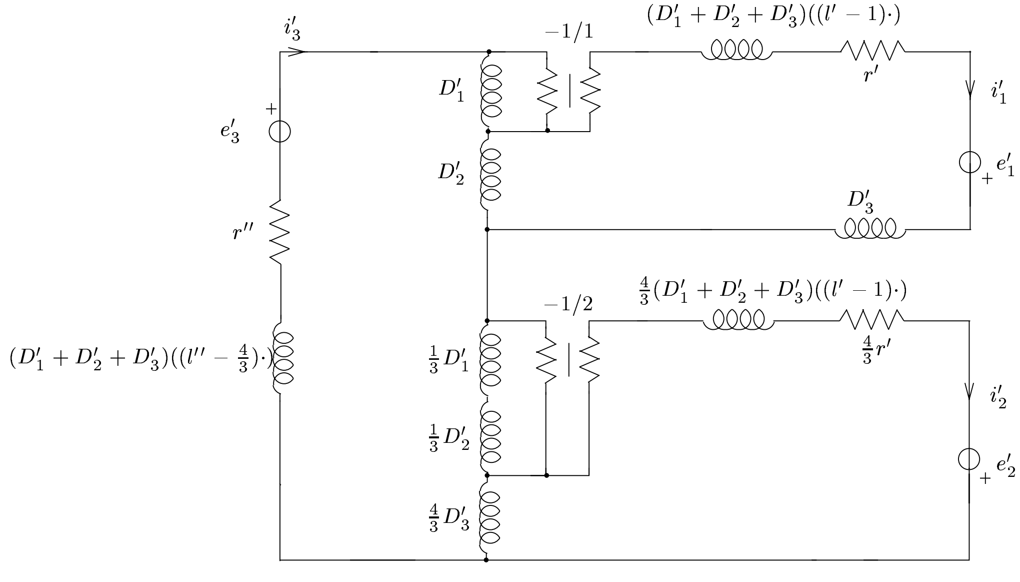

Figure 5: Reference circuit for the parallel plate PDEs when using densest-ball-packing sampling.

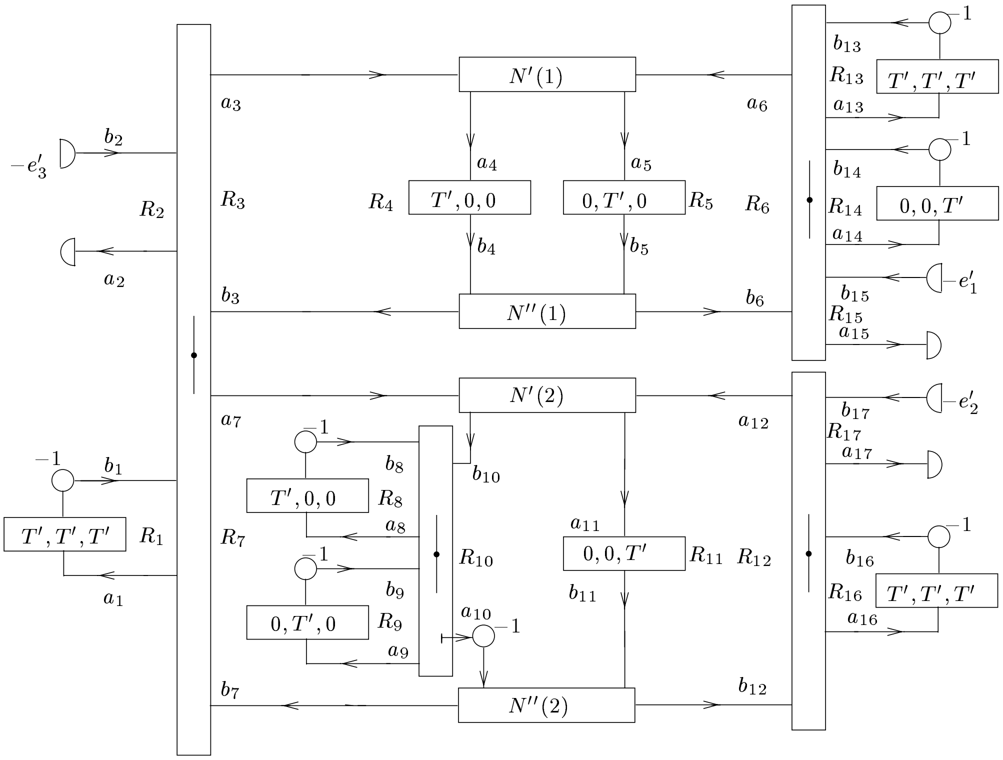

Figure 6: WD network derived from the parallel plate reference circuit of Figure 5.

2.2 "Densest Ball Packing" Sampling

The WD simulation of the 3-dimensional parallel plate problem on a densest-ball-packing grid is given in [2]. A coordinate transformation is first applied to the parallel plate PDEs. Then a different reference circuit and wave digital network are introduced, [2].

The reference circuit derived in [2], is shown in Figure 5. The corresponding WD network appears in Figure 6. Note that the leftmost inductor labeled by $(D_1'+D_2'+D_3')((l''-\frac{4}{3})\cdot)$ in the reference circuit of Figure 5 corresponds to the leftmost delay element in the WD network of Figure 6. The delay is of step size T' in all three D' directions, and its corresponding reference resistance is R_1. The 3-port and 4-port serial adaptors are again given by [1], while $N'(i), N''(i)$, $i = 1, 2$ are shown in [2]. As before, these subnetworks contain only adders and multipliers. Formula for the reference (port) resistances called out in Figure 6 are given in [2].

As we can see from Figure 6, the calculations at an interior grid point need previously-calculated information from grid points in the D_1',D_2',D_3' and $(D_1' + D_2' + D_3')$ directions. A multidimensional inductor that differentiates its current in the D_1' direction will (after discretization) result in one step of delay in the D_1' direction. An inductor that differentiates its current in the D_2' or D_3' direction will similarly result in one step of delay in the D_2' direction or D_3' direction. The inductor of $(D_1' + D_2' + D_3')$ will result in one step of delay in all three D'

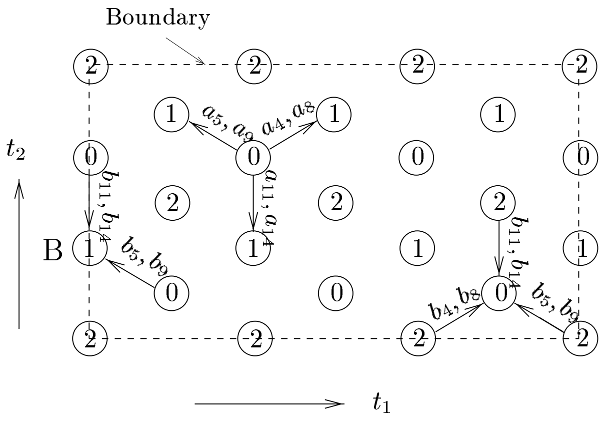

Figure 7: Densest-ball-packing sampling grid for the parallel plate problem and the corresponding data dependencies, depicted in the t_1 and t_2 plane.

directions, which turns out to be equivalent to three steps back in the t_3 direction [2].

Figure 7 shows the sampling grids in the t_1 and t_2 plane. Here, one first carries out the calculations at the grid points marked 0, then the grid points marked 1, and then 2. Then calculations at the points marked 0 are again performed. In Figure 7, we see the patterns of information exchange between neighbors. In all cases, this is information that has been calculated one time step before. A grid point calculation also requires information from its own previous calculation, which was carried out three simulation time steps before.

3 Boundary Conditions

Let us assume, as an example, that the boundaries form a rectangular region (see Figures 4 and 7), and that the termination at a given point on the boundary is a lumped linear one, of the form of Figure 8. Thus, for the western and eastern boundaries, the boundary condition may be written as:

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{f}u + \mathbf{P}\mathbf{q} \tag{14}$$

$$i_1 = \mathbf{d}^T\mathbf{x} + eu + \mathbf{s}^T\mathbf{q} \tag{15}$$

where $\mathbf{A}$ and $\mathbf{P}$ are n×n matrix, $\mathbf{d}$, $\mathbf{f}$, $\mathbf{s}$ and $\mathbf{q}$ are column n-vectors, and e is a scalar. $\mathbf{A}$, $\mathbf{P}$, $\mathbf{d}$, $\mathbf{f}$, $\mathbf{s}$ and e may be functions of the spatial variables t_1 and t_2, but in what follows, we will take these coefficients to be time-invariant. Only the vector of sources, $\mathbf{q}$, will be allowed to be time-varying. $\mathbf{x}$ is a vector of n state variables. i_1 and u are the same dependent variables appearing in the parallel plate PDE's, and i_3 is defined by equation (6).

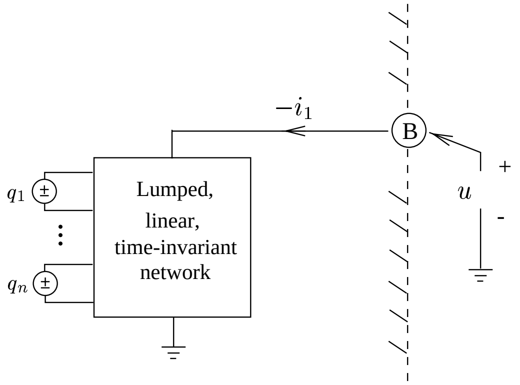

Figure 8: Conceptual view of the termination at a point "B" on the western boundary of a two-dimensional rectangular region. The PDE dependent variables i_1 and u become (at such a boundary point) the port current and voltage of the lumped boundary termination.

Since, $\frac{d}{dt} = D_3$, the boundary equations (14) and (15) will become:

$$D_3\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{f}r_3 i_3 + \mathbf{P}\mathbf{q} \tag{16}$$

$$i_1 = \mathbf{d}^T\mathbf{x} + e r_3 i_3 + \mathbf{s}^T\mathbf{q} \tag{17}$$

We can write the boundary condition equations for the northern and southern boundaries by substituting i_2 for i_1 in $(16-17)$.

3.1 Incorporating Boundary Conditions with "Checkerboard" Sampling

In this section, we will demonstrate how to accommodate lumped linear boundary conditions in the WD simulation of parallel plate PDEs when using the "checkerboard sampling pattern". We will assume a point on the western boundary; however the same procedure can be applied to eastern, southern and northern boundaries as well.

3.1.1 Basic Procedure

As shown in Figure 4, each interior sampling point (node) receives waves (b_{11}, b_{12}) from its eastern and western neighbors, respectively, b_{13} and b_{14} from its northern and southern neighbors, respectively, and b_8, b_9 and b_{10} from its own previous calculations. Note that all these waves have been calculated either one time step or two time steps earlier. However, for grid points lying on the western boundary, such as B in Figure 4, there is no western neighbor. So an incoming b_{12} is not available at these points. We will make use of the boundary condition equations to infer a value for this unknown wave, so that both the original PDE's *and* the boundary condition will be simultaneously satisfied.

In this simulation, these are seven waves entering (or leaving) each grid point, one for each of the seven registers appearing in Figure 2. From [1], we know that incident and reflected (voltage) waves associated with each port are defined, respectively, as

$$a = u + Ri, \qquad b = u - Ri \tag{18}$$

From this, we can conclude that wave quantities for reactive reference circuit elements may be expressed as linear combinations of the dependent variables, i_1, i_2 and i_3, and their derivatives with respect to t_1, t_2 and t_3. And for the same reason, the dependent variables may be expressed as linear combinations of the waves. Let's first define a vector $\mathbf{b}$ as the set of wave values held in the registers of Figure 3, and the vector $\mathbf{w}'$ as the set of dependent variables and their first derivatives with respect to t_1, t_2 and t_3.

$$\mathbf{b} = [b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13}, b_{14}]^T \tag{19}$$

$$\mathbf{w}' = [i_1, i_2, i_3, D_1 i_1, D_2 i_1, D_3 i_1, D_1 i_2, D_2 i_2, D_3 i_2, D_1 i_3, D_2 i_3, D_3 i_3]^T \tag{20}$$

The linear relation between $\mathbf{b}$ and $\mathbf{w}'$ can be expressed by a (7×12) matrix $\mathbf{M}'$. Applying equations (18), it can be shown that in this parallel plate simulation

$\mathbf{M}' =$

$$\begin{pmatrix} -R_8 & 0 & 0 & 0 & 0 & l-\delta_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -R_9 & 0 & 0 & 0 & 0 & 0 & 0 & l-\delta_2 & 0 & 0 & 0 \\ 0 & 0 & -R_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (cr_3^2-\delta_1-\delta_2) \\ R_{11} & 0 & -R_{11} & \frac{1}{2}r_3 & 0 & -\frac{1}{2}\delta_1 & 0 & 0 & 0 & -\frac{1}{2}r_3 & 0 & \frac{1}{2}\delta_1 \\ -R_{12} & 0 & -R_{12} & \frac{1}{2}r_3 & 0 & \frac{1}{2}\delta_1 & 0 & 0 & 0 & \frac{1}{2}r_3 & 0 & \frac{1}{2}\delta_1 \\ 0 & R_{13} & -R_{13} & 0 & 0 & 0 & 0 & \frac{1}{2}r_3 & -\frac{1}{2}\delta_2 & 0 & -\frac{1}{2}r_3 & \frac{1}{2}\delta_2 \\ 0 & -R_{14} & -R_{14} & 0 & 0 & 0 & 0 & \frac{1}{2}r_3 & \frac{1}{2}\delta_2 & 0 & \frac{1}{2}r_3 & \frac{1}{2}\delta_2 \end{pmatrix} \tag{21}$$

so that

$$\mathbf{b} = \mathbf{M}'\mathbf{w}' \tag{22}$$

We notice that matrix $\mathbf{M}'$ has two columns, the fifth and the seventh, with all 0 elements. So $\mathbf{M}'$ can be thought of as a 7×10 matrix since $D_2 i_1$ and $D_1 i_2$ have no effect on (22). Now it is clear that there should be a bijective mapping between the waves and the dependent variable quantities in terms of which they are defined; that is, in terms of i_1, i_2, i_3 and their various first derivatives. That is, an inverse relation to (22) should exist. And, indeed, this is the case here since the original PDE's provide us with three additional relations among the dependent variables and their first derivatives. We rewrite these below as $(23-25)$, assuming (without loss of generality) that the sources e_1, e_2 and e_3 are zero.

$$D_1 i_3 = -\frac{r}{r_3} i_1 - \frac{l}{r_3} D_3 i_1 \tag{23}$$

$$D_2 i_3 = -\frac{r}{r_3} i_2 - \frac{l}{r_3} D_3 i_2 \tag{24}$$

$$D_1 i_1 = -D_2 i_2 - r_3 c D_3 i_3 - r_3 g i_3 \tag{25}$$

From these three equations, we see that three of the first derivative quantities can be expressed in terms of a "minimal" set of seven (excluding, again, $D_2 i_1$ and $D_1 i_2$). This minimal set is listed in the vector $\mathbf{w}$.

$$\mathbf{w} = [i_1, i_2, i_3, D_3 i_1, D_2 i_2, D_3 i_2, D_3 i_3]^T \tag{26}$$

Therefore, we may collapse five of the columns of $\mathbf{M}'$ to form $\mathbf{M}$.

$\mathbf{M} =$

$$\begin{pmatrix} -R_8 & 0 & 0 & l-\delta_1 & 0 & 0 & 0 \\ 0 & -R_9 & 0 & 0 & 0 & l-\delta_2 & 0 \\ 0 & 0 & -R_{10} & 0 & 0 & 0 & (cr_3^2-\delta_1-\delta_2) \\ R_{11}+\frac{r}{2} & 0 & -R_{11}-\frac{r_3^2 g}{2} & -\frac{1}{2}\delta_1+\frac{l}{2} & -\frac{r_3}{2} & 0 & \frac{1}{2}\delta_1-\frac{r_3^2 c}{2} \\ -R_{12}-\frac{r}{2} & 0 & -R_{12}-\frac{r_3^2 g}{2} & \frac{1}{2}\delta_1-\frac{l}{2} & -\frac{r_3}{2} & 0 & \frac{1}{2}\delta_1-\frac{r_3^2 c}{2} \\ 0 & R_{13}+\frac{r}{2} & -R_{13} & 0 & \frac{1}{2}r_3 & -\frac{1}{2}\delta_2+\frac{l}{2} & \frac{1}{2}\delta_2 \\ 0 & -R_{14}-\frac{r}{2} & -R_{14} & 0 & \frac{1}{2}r_3 & \frac{1}{2}\delta_2-\frac{l}{2} & \frac{1}{2}\delta_2 \end{pmatrix} \tag{27}$$

In practice, we find $\mathbf{M}$ to be of full rank, though we have no proof of this that would guarantee this for all possible systems of PDE's. Equation (22) can thus be written:

$$\mathbf{b} = \mathbf{M}\mathbf{w} \tag{28}$$

This implies that given $\mathbf{w}$, we can calculate the wave quantities $\mathbf{b}$, and vice versa (by $\mathbf{w} = \mathbf{M}^{-1}\mathbf{b}$).

Referring again to Figure 4, we recall that the boundary grid point B has no western neighbor that can supply a value for the incoming wave b_{12}. However, in the next subsection, we will show that when the boundary condition (17) is introduced into (28), two of the elements of the vector $\mathbf{w}$ become dependent; that is, one element of $\mathbf{w}$ may be simply written in terms of one of the others. This will enable us to eliminate one element from $\mathbf{w}$ and also merge two columns of $\mathbf{M}$. $\mathbf{M}$ will then have seven rows and six columns, and will be of rank no greater than six. We will then find it to be the case that the unknown element of the $\mathbf{b}$ vector, b_{12}, may be written as a linear combination of the remaining six elements in the $\mathbf{b}$ vector (plus the vector $\mathbf{x}$, since $\mathbf{x}$ also appears in (17)). Therefore, we can express i_3 in vector $\mathbf{w}$ as a function of these six $\mathbf{b}$ elements (without b_{12}). At this point, equation (16) will contain only the state variables $\mathbf{x}$ and (at most) the six *known* $\mathbf{b}$ wave values. Thus we may advance the simulation of the boundary equation (16) by one time step, producing updated estimates for $\mathbf{x}$. We will then find that, with the updated $\mathbf{x}$, a value for the unknown b_{12} may be inferred. With b_{12} known, the simulation of the WD network at grid point B may then advance to the next time step, as well.

3.1.2 Detailed Discussion

We first define the 7-vector $\mathbf{i}$(j) such that $\mathbf{i}^T(j)\mathbf{w}$ will extract the jth element of $\mathbf{w}$. Obviously $\mathbf{i}$(j) is a column vector, all of whose elements are 0 except for a lone 1 in position j. For example,

$$\mathbf{i}(1) = [1, 0, 0, 0, 0, 0, 0]^T \tag{29}$$

Next, we define

$$\mathbf{V} = \begin{pmatrix} 0 & 0 & er_3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{30}$$

Notice that matrix $\mathbf{V}$ has one zero column and is of rank six.

Substituting boundary equation (17) into equation (28), it can be shown that:

$$\mathbf{w} = \mathbf{V}\mathbf{w} + \mathbf{i}(1)[\mathbf{d}^T\mathbf{x} + \mathbf{s}^T\mathbf{q}] \tag{31}$$

and

$$\mathbf{b} = \mathbf{M}\mathbf{V}\mathbf{w} + \mathbf{M}\mathbf{i}(1)[\mathbf{d}^T\mathbf{x} + \mathbf{s}^T\mathbf{q}] \tag{32}$$

Since $\mathbf{V}$ is a matrix of rank six, $\mathbf{MV}$ will be a 7×7 matrix with a rank of no more than six. We will therefore assume that b_{12} may be expressed in terms of the six known b values and $\mathbf{x}$. That is to say, that there exists a vector $\mathbf{h}$

$$\mathbf{h} = [h_1, h_2, h_3, h_4, 0, h_6, h_7]^T \tag{33}$$

such that

$$\mathbf{i}^T(5)\mathbf{M}\mathbf{V} = \mathbf{h}^T\mathbf{M}\mathbf{V} \tag{34}$$

where $h_k, k = 1, ..., 7$ are real numbers which can be derived from equation (34). (Clearly, this assumption must be verified in any given application, but, to date, we have not encountered any realistic combination of parameters in equations (1 - 3) for which such an $\mathbf{h}$ does not exist.)

So, beginning with (32),

$$b_{12} = \mathbf{i}^T(5)\mathbf{b} = \mathbf{i}^T(5)\mathbf{M}\mathbf{V}\mathbf{w} + \mathbf{i}^T(5)\mathbf{M}\mathbf{i}(1)[\mathbf{d}^T\mathbf{x} + \mathbf{s}^T\mathbf{q}] \tag{35}$$

We then have:

$$b_{12} = \mathbf{h}^T\mathbf{M}\mathbf{V}\mathbf{w} + \mathbf{i}^T(5)\mathbf{M}\mathbf{i}(1)[\mathbf{d}^T\mathbf{x} + \mathbf{s}^T\mathbf{q}]$$

$$= \mathbf{h}^T\mathbf{b} + [\mathbf{i}^T(5) - \mathbf{h}^T]\mathbf{M}\mathbf{i}(1)[\mathbf{d}^T\mathbf{x} + \mathbf{s}^T\mathbf{q}] \tag{36}$$

Now, if $\mathbf{M}$ is of full rank, equation (28) can also be written as:

$$\mathbf{w} = \mathbf{M}^{-1}\mathbf{b} \tag{37}$$

So i_3 can be expressed as a linear combination of the elements of $\mathbf{b}$. But given (36), we now know that i_3 may be expressed in terms of $\mathbf{h}^T\mathbf{b}$, $\mathbf{x}$ and $\mathbf{q}$; that is, i_3 may be evaluated without knowing b_{12}. Therefore, our boundary condition (16)

$$D_3\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{f}r_3i_3 + \mathbf{P}\mathbf{q}$$

need not involve b_{12}, either.

Since the WD method for PDE simulation makes use of the trapezoidal rule, it is natural to suggest that the trapezoidal rule be used to discretize (16). Then, with $\mathbf{x}$ known at the current simulation time point, we may advance the discrete-time simulation of the lumped boundary equation forward by one time step to arrive at an updated value for $\mathbf{x}$. Next, with an updated value for $\mathbf{x}$ known, we may use (36) to calculate b_{12}. Finally, using this inferred value for b_{12}, the WD PDE simulation at the boundary grid point may also be advanced by one time step.

3.1.3 Summary

The procedure for accommodating boundary conditions in this specific parallel plate problem (discretized onto a checkerboard grid) may be summarized as follows:

- Express the incoming b waves in terms of the PDE system dependent variables and their first derivatives, as in (22).
- Assuming n_{dv} original PDEs, use the PDE relationships to eliminate n_{dv} columns in $\mathbf{M}'$ and n_{dv} elements in $\mathbf{w}'$. Thus (22) becomes (28).
- With the boundary condition equation (17), eliminate one more column in $\mathbf{M}$ and thus write the unknown wave, b_{12}, in terms of the remaining (known) waves and $\mathbf{x}$, (36).
- Express i_3 in terms of the known waves and the state vector $\mathbf{x}$. Advance the boundary equation simulation of (16) one time step forward to obtain an updated $\mathbf{x}$.
- With the updated value of $\mathbf{x}$, calculate a value for the unknown wave, b_{12}, using (36).
- Perform the standard WD simulation at the boundary point with the incoming waves either known or inferred.

Obvious variations of the above method can clearly be used to handle grid points lying on northern, southern, or eastern boundaries. For a southern boundary, for example, the entering wave b_{14} will be missing. (See Figure 4.) However, this wave may be similarly inferred by making use of the given boundary condition relation between i_2 and u.

3.2 Incorporating Boundary Conditions with "Densest Ball Packing" Sampling

In this section, we will show how to apply the boundary simulation procedure developed in the previous section to accommodate boundary condition in the WD simulation of the parallel plate PDEs with "densest ball packing" sampling pattern. The procedure we have just seen can also be applied at the western and eastern boundaries in this case as well. The southern and northern boundaries require a special approach, however, which we will discuss later.

3.2.1 Vertical Boundaries

As shown in Figure 7, each node receives incoming waves b_4, b_8 from its southwestern neighbor, b_5 and b_9 from its southeastern neighbor, and b_{11} and b_{14} from its northern neighbor. All these b's have been calculated one time step earlier. For grid points lying on the vertical boundary, such as B in Figure 7, there are no southwestern neighbors. So b_4 and b_8 are not available at these points. We will make use of the boundary equations to infer both b_4 and b_8.

As we have shown in the previous section, there are nine waves associated with nine registers. By equation (18), we can again write (22), where vector $\mathbf{b}$ is now defined as the set of nine wave values held in registers, and vector $\mathbf{w}'$ contains the set of dependent variables and their first derivatives with respect to t_1, t_2 and t_3, (20).

$$\mathbf{b} = [b_1, b_4, b_5, b_8, b_9, b_{11}, b_{13}, b_{14}, b_{16}]^T \tag{38}$$

The matrix $\mathbf{M}'$ which expresses the linear relation between $\mathbf{b}$ and $\mathbf{w}'$ now has nine rows corresponding to the nine elements in vector $\mathbf{b}$ and twelve columns corresponding to the twelve-vector $\mathbf{w}'$, see (39).

$\mathbf{M}' =$

$$\begin{pmatrix}
0 & 0 & -R_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{3}}{v}(l'' - \frac{4}{3}) \\
-\frac{R_4}{\sqrt{2}} & 0 & -R_4 & \frac{1}{2} & \frac{\sqrt{3}}{6} & \frac{1}{\sqrt{6}v} & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{6}}{6} & \frac{\sqrt{3}}{3v} \\
\frac{R_5}{\sqrt{2}} & 0 & -R_5 & \frac{1}{2} & -\frac{\sqrt{3}}{6} & -\frac{1}{\sqrt{6}v} & 0 & 0 & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{6}}{6} & \frac{\sqrt{3}}{3v} \\
0 & -\sqrt{\frac{3}{2}}R_8 & -R_8 & 0 & 0 & 0 & \frac{\sqrt{3}}{6} & \frac{1}{6} & \frac{\sqrt{2}}{6v} & \frac{\sqrt{2}}{6} & \frac{\sqrt{6}}{18} & \frac{\sqrt{3}}{9v} \\
0 & -\sqrt{\frac{3}{2}}R_9 & -R_9 & 0 & 0 & 0 & -\frac{\sqrt{3}}{6} & \frac{1}{6} & \frac{\sqrt{2}}{6v} & -\frac{\sqrt{2}}{6} & \frac{\sqrt{6}}{18} & \frac{\sqrt{3}}{9v} \\
0 & \sqrt{\frac{3}{8}}R_{11} & -R_{11} & 0 & 0 & 0 & 0 & \frac{2}{3} & -\frac{\sqrt{2}}{3v} & 0 & -\frac{4\sqrt{6}}{9} & \frac{4\sqrt{3}}{9v} \\
-\frac{R_{13}}{\sqrt{2}} & 0 & 0 & 0 & 0 & \sqrt{\frac{3}{2}}\frac{l'-1}{v} & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{R_{14}}{\sqrt{2}} & 0 & 0 & 0 & -\frac{\sqrt{3}}{3} & \frac{1}{\sqrt{6}v} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\sqrt{\frac{3}{8}}R_{16} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{v}(l' - 1) & 0 & 0 & 0
\end{pmatrix}$$

Again from the transformed parallel plate equations $(23-25)$, we can eliminate the three dependent variables D_1i_3, D_2i_3 and D_1i_1 in $\mathbf{w}'$. Correspondingly, three columns in matrix $\mathbf{M}'$ can be combined with the remaining nine columns. Vector $\mathbf{w}'$ and matrix $\mathbf{M}'$ will become vector $\mathbf{w}$ and matrix $\mathbf{M}$ where

$$\mathbf{w} = [i_1, i_2, i_3, D_2i_1, D_3i_1, D_1i_2, D_2i_2, D_3i_2, D_3i_3]^T \tag{39}$$

$\mathbf{M} =$

$$\begin{pmatrix}
0 & 0 & -R_1 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{3}}{v}(l" - \frac{4}{3}) \\
-\frac{1}{\sqrt{2}}R_4 - \frac{r}{\sqrt{2}r_3} & -\frac{\sqrt{6}r}{6r_3} & -R_4 - \frac{1}{2}r_3g & \frac{\sqrt{3}}{6} & \frac{1}{\sqrt{6}v} - \frac{l}{\sqrt{2}r_3} & 0 & -\frac{1}{2} & -\frac{\sqrt{6}l}{6r_3} & \frac{\sqrt{3}}{3v} - \frac{1}{2}r_3c \\
\frac{1}{\sqrt{2}}R_5 + \frac{r}{\sqrt{2}r_3} & -\frac{\sqrt{6}r}{6r_3} & -R_5 - \frac{1}{2}r_3g & -\frac{\sqrt{3}}{6} & -\frac{1}{\sqrt{6}v} + \frac{l}{\sqrt{2}r_3} & 0 & -\frac{1}{2} & -\frac{\sqrt{6}l}{6r_3} & \frac{\sqrt{3}}{3v} - \frac{1}{2}r_3c \\
-\frac{\sqrt{2}r}{6r_3} & -\sqrt{\frac{3}{2}}R_8 - \frac{\sqrt{6}r}{18r_3} & -R_8 & 0 & -\frac{\sqrt{2}l}{6r_3} & \frac{1}{2\sqrt{3}} & \frac{1}{6} & \frac{\sqrt{2}}{6v} - \frac{\sqrt{6}l}{18r_3} & \frac{\sqrt{3}}{9v} \\
\frac{\sqrt{2}r}{6r_3} & -\sqrt{\frac{3}{2}}R_9 - \frac{\sqrt{6}r}{18r_3} & -R_9 & 0 & \frac{\sqrt{2}l}{6r_3} & -\frac{1}{2\sqrt{3}} & \frac{1}{6} & \frac{\sqrt{2}}{6v} - \frac{\sqrt{6}l}{18r_3} & \frac{\sqrt{3}}{9v} \\
0 & \sqrt{\frac{3}{8}}R_{11} + \frac{4\sqrt{6}r}{9r_3} & -R_{11} & 0 & 0 & 0 & \frac{2}{3} & -\frac{\sqrt{2}}{3v} + \frac{4\sqrt{6}l}{9r_3} & \frac{4\sqrt{3}}{9v} \\
-\frac{R_{13}}{\sqrt{2}} & 0 & 0 & 0 & \sqrt{\frac{3}{2}}\frac{l'-1}{v} & 0 & 0 & 0 & 0 \\
-\frac{R_{14}}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}v} & 0 & 0 & 0 & 0 \\
0 & -\sqrt{\frac{3}{8}}R_{16} & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{2}(l'-1)}{v} & 0
\end{pmatrix} \tag{40}$$

$\mathbf{b}$ and $\mathbf{w}$ are related by equation (28). In what follows, we will assume $\mathbf{M}$ to be a 9×9 full-rank matrix.

The boundary condition we will impose on the western boundary is again $(16-17)$. Also, equation (17) implies that:

$$D_3i_1 = \mathbf{d}^T D_3\mathbf{x} + eD_3u + \mathbf{s}^T D_3\mathbf{q} \tag{41}$$

By substituting equations (16) and (6) into (41), we have:

$$D_3i_1 = \mathbf{d}^T\mathbf{A}\mathbf{x} + \mathbf{d}^T\mathbf{f}r_3i_3 + \mathbf{d}^T\mathbf{P}\mathbf{q} + \mathbf{s}^T D_3\mathbf{q} + er_3D_3i_3 \tag{42}$$

With the boundary equations (17) and (42), we may follow our earlier procedure and combine the columns of $\mathbf{M}$ corresponding to i_1 and D_3i_1 with the remaining seven columns. In fact, $\mathbf{M}$ turns out to have a rank of seven, and, further, the rows of $\mathbf{M}$ corresponding to b_4 and b_8 are dependent upon the remaining seven rows. We may thus express the unknown waves, b_4 and b_8, as a linear combination of the other b's and the boundary state variable $\mathbf{x}$, [32]. Therefore, we are able to express i_3 in vector $\mathbf{w}$ as a function of the b values excluding b_4 and b_8, as that equation (16) need not contain b_4 and b_8, either. Thus we may advance the simulation of the boundary equation (16) by one step in discrete time to obtain updated $\mathbf{x}$ values. With the updated $\mathbf{x}$, we may then calculate the b_4 and b_8, and then perform the normal parallel plate simulation at this same boundary point.

This method can also be used to accommodate boundary conditions when the vertical boundary forms the eastern edge of a rectangular region. For points lying on such an eastern

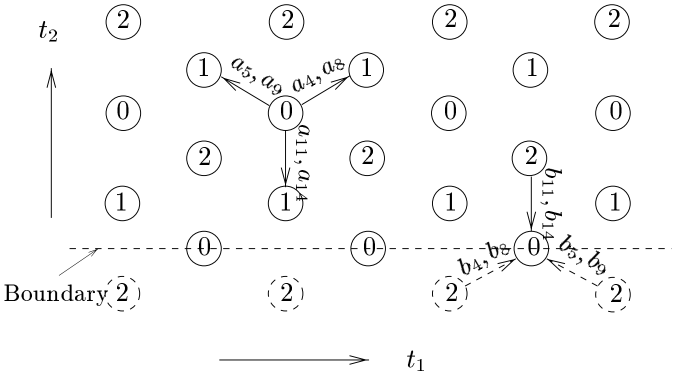

Figure 9: Densest-ball-packing sampling (for the parallel plate problem) at a southern boundary.

boundary, waves b_5 and b_9 are missing. But we still may use the method developed above to infer the two unknown waves.

For a southern horizontal boundary, the situation is different. There are four waves, b_4, b_5, b_8, and b_9, which are unknown, since points lying on the southern boundary have no southeastern *or* southwestern neighbors. A solution to this southern boundary case will be given in the next chapter. For grid points lying on the northern boundary, waves b_{11} and b_{14} are not available. We can use the same kind of approach used for the western boundary to infer these two unknown waves. For the grid points lying immediately below the northern boundary, there are also two waves missing, b_{11} and b_{14}. However, since these points are not precisely on the boundary, we cannot simply apply the boundary procedure given above. Thus, we will also take the northern boundary as a special case and treat it in a later section.

3.2.2 Horizontal Boundaries

Southern Boundary Recall that in the previous section when the boundary edge was parallel to the t_2-axis and lying to the left, there were two unknown waves entering the points on the boundary, b_4 and b_8, because there was no southwestern neighbor. With the boundary equations, (17) and (42), these two unknown waves were inferred from a linear combination of the other b's and the state variables $\mathbf{x}$. Then we advanced the boundary state equation simulation one step to obtain the updated $\mathbf{x}$ and then b_4 and b_8. After that the normal parallel plate simulation was performed at the boundary point: That is, the a waves were calculated for export to neighboring grid points at the next simulation time step.

Since as shown in Figure 9, grid points lying on the southern boundary have no southwestern or southeastern neighbors, there are four waves missing, b_4 and b_8 from the southwest, and b_5 and b_9 from the southeast. The boundary equations, (17) and (42), do not provide sufficient information to infer all four unknown waves. We will add more sampling points to this boundary

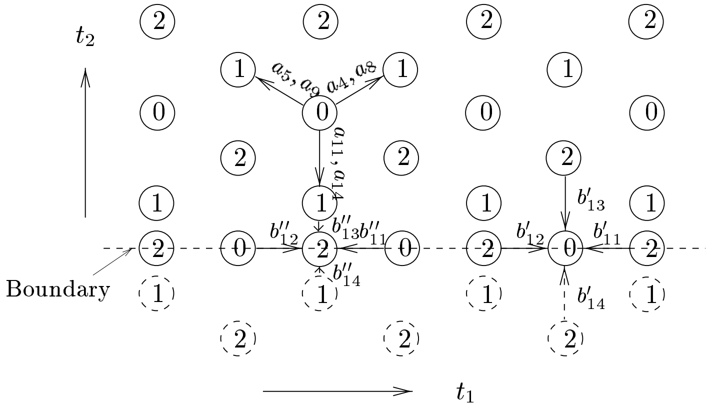

Figure 10: Added sampling points along the southern boundary when using the densest-ball-packing grid.

as a means of dealing with this deficiency. See Figure 10.

As shown in Figure 10, we propose that the parallel plate simulation be done first at points 0, then points 1 and then 2. The points labeled 2 on the boundary receive information from their northern neighbors labeled 1 which have been calculated one time step before, and receive information from their western and eastern neighbors, labeled as 0, which have been calculated two time steps before. These points should also receive information from their southern neighbors, labeled as 1, which are missing. The boundary points labeled as 0 will receive information from their northern, western and eastern neighbors labeled as 2, which have been calculated one time step before; and their southern neighbors are also missing. In addition, boundary points 0 or 2 also make use of three waves from their previous calculation which was carried out three time steps earlier. All boundary points now have only one missing neighbor as a result of our doubling the sampling grid points at the southern boundary in this way. So we can infer the unknown waves associated with these missing neighbors by applying the boundary conditions as in section 3.1.

Clearly the wave digital algorithm performed at these southern boundary points must be different from the one performed at the interior grid points. The reference circuit and the WD network for these southern boundary points (0 and 2) are given in [32].

Notice that boundary grid points labeled 0 and 2 are now performing different WD simulation algorithms, and that information exchange between these grid points is necessary. Also notice that interior grid points of types 1 and 2 which lie adjacent to the southern boundary must pass the correct wave information to boundary points of types 2 and 0. Also, boundary points 0 must send the requested wave information to interior grid points 1. Details of these new information transfers may be found in [32].

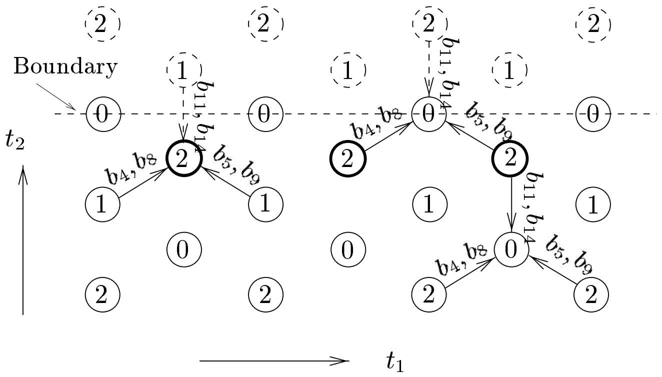

Figure 11: Northern boundary situation (for the parallel plate problem) when using a densest-ball-packing grid.

Northern Boundary For grid points marked 0 lying on the northern boundary shown in Figure 11, there is only one missing neighbor associated with two unknown waves, b_{11} and b_{14}. It is therefore the case that we may use the same procedure applied on the western boundary in section 3.2 to infer these two unknown waves. But for grid points 2 which lie immediately below the northern boundary, (shown highlighted in Figure 11), there is no northern neighbor either. But, since these grid points do not lie directly on the boundary, we cannot apply the same procedure as was used for the western and eastern boundaries.

A way to treat this northern boundary problem is very similar to what was done at the southern boundary. First, we introduce new grid points on the northern boundary between the existing grid points 0, as shown in Figure 12. These new introduced points are marked 1 because they perform their WD simulations at the same simulation time steps as the interior plate points 1 do their WD calculations. Second, the northern boundary points 0 or 1 now perform different WD simulation algorithms compared to inner plate points 0 or 1. This is evident from the observation that a northern boundary point of type 0 now receives information from its four neighbors to the north, south, west and east. Each neighbor contributes only one wave, which is calculated two time steps before. A type 0 point also needs three waves from its previous calculation, which was done three time steps earlier. For a northern boundary point of type 1, it receives one wave from both its north and south neighbors, respectively. These two waves have been calculated two time steps before. It also needs two waves from its west and east neighbors, which have been calculated one time step before. Finally, it needs three waves from its previous calculation which has been carried out three time steps before. So as in the southern boundary, the WD simulation algorithms performed on northern boundary points 0 and 1 are very similar to checkerboard algorithm. We can derive the WD networks for these northern boundary points of types 0 and 1, and these may be found in [32].

The WD simulation algorithms performed at the northern boundary points 0 and 1 enable

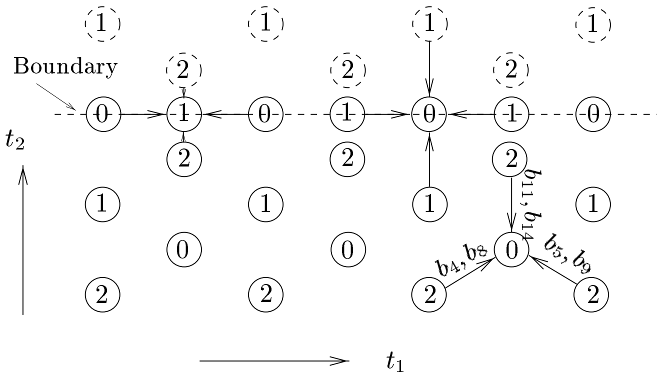

Figure 12: Added sampling points at the northern boundary when using a densest-ball-packing grid.

us to apply the procedure developed in chapter 3.1 to accommodate the boundary condition in the WD simulation, since there is now only one missing north neighbor, associated with only one incoming wave, for northern boundary points of type 0 or 1.

Since the north boundary points 0 or 1 implement WD algorithms that are different from that applied at the interior grid points, a modified information exchange between boundary points and inner plate points is necessary. For details refer to [32].

4 A General Procedure

We will now summarize the boundary condition procedure used in the preceding sections; we will do this in more general terms than before since it is our intention to show:

- that the procedure may be generalized and immediately applied to many hyperbolic systems of PDE's, including nonlinear ones;
- there is a simple "equation counting" method to determine in any specific case if the procedure is or is not applicable; and
- what corrective action may be taken in those cases in which the counting method indicates that the procedure is not immediately applicable.

We will repeatedly refer back to the parallel plate cases that we have developed in our earlier sections, since – as we shall see – these provide us several examples of the various steps involved in the more general procedure.

Suppose that, for a given system of PDE's and corresponding sets of initial and boundary conditions, a reference circuit has been developed, and (from this reference circuit) a WD network has been constructed. At a given sampling point on the boundary, we wish to determine whether the boundary condition at that point may be incorporated into the calculations of the WD network assigned to this sampling point. The result of this determination may be negative; that is, the nature of the reference circuit and the direction of the boundary may be such that insufficient information may be available to the WD network for it to carry out all its calculations. The procedure we enunciate below will allow one to make this determination. Then, if a lack of sufficient information indeed turns out to be the case, we will give suggestions for how this situation might be overcome.

We will use the abbreviation "dv" in place of the term "dependent variable". This would refer to the voltage and current functions in the reference circuits for the parallel plate problem dealt with in the earlier sections for example. Also appearing in the reference circuit will be some number of first derivatives of these dv's taken with respect to the various independent variables. When considering the collection of the dv's *and* these first derivative terms, we will usually abbreviate this larger set of functions by "dv&d" – the dependent variables *and* those of their first derivatives that appear in the reference circuit.

At a boundary sampling point, one or more (say n_{un}) of the n_w incoming wave values will be unknown. In order for the WD network associated with this sampling point to carry out all of its calculations, we obviously need to infer these values from other considerations. To see how this might be done, we first suppose that the number of distinct dv&d functions that appear in the reference circuit is $n_{dv\&d}$. We then recall that every incoming wave is defined as a certain (known) linear combination of these $n_{dv\&d}$ functions. (We will assume that these wave-defining equations are linearly independent.) Therefore, this must be true of the $n_w - n_{un}$ equations defining the waves whose incoming values *are* known. Thus, these equations constitute $n_w - n_{un}$ independent, linear relationships among the dv&d's. It will be the case that

$$n_{dv\&d} > n_w - n_{un}$$

Therefore, if we can find $(n_{dv\&d} - (n_w - n_{un}))$ additional independent relationships solely among the $n_{dv\&d}$ dv&d's, then it will be the case that given the $n_w - n_{un}$ known incoming waves, one may calculate corresponding values for all the dv&d's. Finally, "inferred" values for the n_{un} unknown incoming waves may be calculated since these waves also have known definitions in terms of the dv&d's.

But where can we find these $(n_{dv\&d} - (n_w - n_{un}))$ additional independent relationships among the $n_{dv\&d}$ dv&d's? To begin with, the n_{dv} PDE's from which the reference circuit was constructed are certainly (possibly nonlinear) relationships among the dv&d's. Second, the state variable boundary conditions can provide us with another independent relationship, namely

$$i = \mathbf{d}^T \mathbf{x} + eu + \mathbf{s}^T \mathbf{q} \tag{43}$$

where

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{f}u + \mathbf{P}\mathbf{q} \tag{44}$$

for the linear boundary condition and

$$i = g(\mathbf{x}, u, t) \tag{45}$$

where

$$\frac{d}{dt}\mathbf{x} = \mathbf{h}(\mathbf{x}, u, t) \tag{46}$$

for a large class of nonlinear boundary conditions.

Also, by differentiating (43) (possibly (45)) with respect to time, and then substituting for $\frac{d}{dt}\mathbf{x}$ with the right member of equation (44)(or (46)), we may arrive at another boundary equation that may also be independent of the other relationships in our set. In the linear case, this equation is

$$\frac{di}{dt} = \mathbf{d}^T(\mathbf{A}\mathbf{x} + \mathbf{f}u + \mathbf{P}\mathbf{q}) + e\frac{du}{dt} + \mathbf{s}^T\frac{d\mathbf{q}}{dt} \tag{47}$$

and

$$\frac{di}{dt} = \frac{\delta g}{\delta t}\frac{dx}{dt} + \frac{\delta g}{\delta u}\frac{du}{dt} + \frac{\delta g}{\delta t} \tag{48}$$

for the nonlinear one. Refer to section 3.2 for an example in which this second type of boundary equation is used.

If this overall set of equations (containing the known wave definitions, the PDE's, and the boundary condition) is large enough, it could be solved (numerically, in the case of most nonlinear systems of PDE's) for the $n_{dv\&d}$ dv&d's. These, in turn, could be used to arrive at values for the n_{un} unknown incoming waves. Then, with values in hand for all n_w incoming waves, the WD network calculations assigned to this sampling point could proceed.

But, we must also note that when (43) or (45) (and possibly (47)) are included in the overall equation set, it introduces the n_{st} boundary state variables into the set, and so it would have to be possible for us to arrive at current values for these variables, as well. In simpler PDE systems, one may be able to write a closed-form expression for u in equation (44) that involves only *known* wave quantities. We have shown an example of doing this in section 3.1.1 for a linear PDE system (and see [11, 14]), and this may also be possible to do in simpler nonlinear systems. If this is the case, this value for u may be used to advance the ODE system (44) to the next (current) time step. This would yield us updated values for the state variables $\mathbf{x}$, which could then be used to eliminate all appearances of these variables in the overall set of equations referred to above.

But in more complex PDE systems, and especially in most nonlinear systems, (46), the current values of the $\mathbf{x}$ state variables must simply be solved for simultaneously with the current values of the dv&d's. We may do this by appending the n_{st} equations $(45-46)$ to the overall equation set. But in order to avoid introducing still more unknown quantities – namely the n_{st} $\frac{d}{dt}\mathbf{x}$ values – we first transcribe (44) or (46) into discrete-time form using the same numerical approximation that underlies the WD method, the trapezoidal rule. Starting with the well-known trapezoidal rule approximation,

$$\frac{T}{2}[\frac{d\mathbf{x}(t)}{dt}+\frac{d\mathbf{x}(t-T)}{dt}]=\mathbf{x}(t)-\mathbf{x}(t-T) \tag{49}$$

we see that the current estimate of $\frac{d}{dt}\mathbf{x}$ may be expressed as

$$\frac{d\mathbf{x}(t)}{dt}=\frac{2}{T}\mathbf{x}(t)-\frac{2}{T}\mathbf{x}(t-T)-\frac{d\mathbf{x}(t-T)}{dt} \tag{50}$$

When this is applied in (44), and terms are algebraically rearranged, we arrive at

$$[\mathbf{A}-\frac{2}{T}\mathbf{I}]\mathbf{x}(t)+\mathbf{f}u(t)+\mathbf{P}\mathbf{q}(t)=-[(\mathbf{A}+\frac{2}{T}\mathbf{I})\mathbf{x}(t-T)+\mathbf{f}u(t-T)+\mathbf{P}\mathbf{q}(t-T)] \tag{51}$$

or

$$\mathbf{x}(t)-\frac{T}{2}\mathbf{h}(\mathbf{x},u,t)=\mathbf{x}(t-T)+\frac{T}{2}\mathbf{h}(\mathbf{x},u,t-T) \tag{52}$$

when (50) is applied in (46) and terms are also algebraically rearranged.

These are the algebraic equations that may be appended to the overall set, and, if desired, solved for all the elements of $\mathbf{x}$ as well as the dv&d's. With current values for these known, we may calculate the unknown wave inputs to the WD network which, in turn, will compute the wave quantities to be exported to neighboring WD networks at the next time step. Also, knowing the current $\mathbf{x}$ values, we may calculate any other dependent variables within the state variable boundary system that may be of interest.

One should also remember that when a reference circuit is constructed from a set of PDE's, it can be the case that "new" derivative terms will appear in the reference circuit (and therefore in the wave defining equations) that do not seem to appear in the PDE system. In this situation, it will always turn out to be the case that any "new" derivative direction being applied to a given dependent variable in the reference circuit may be written as a linear combination of the directions used by the derivative terms being applied to the same dependent variable in the PDE's. In these situations, one could simply append these extra linear equations to the overall set of equations described above so that the "new" derivative values may be related to the known incoming waves, also.

Now, if it turns out that not all of the $(n_{dv\&d}-(n_w-n_{un}))$ needed independent relationships exist, what can be done? First of all, a different transformation of the independent variables may be applied. This will give rise to a different reference circuit that may have fewer waves entering the boundary sampling point from outside the boundary, for example. An instance of this has already been seen: recall the two parallel plate reference circuits developed above, one for the case in which the independent variables were transformed versus the case in which they were not. Along the southern boundary of a rectangular region, a WD network stemming from transformed variables (and resulting in a "densest-ball-packing" sampling grid) required four waves entering from outside the boundary, while the WD network constructed from the original independent variables (which made use of the so-called "checkerboard" sampling pattern) required only one.

Alternatively, one may apply a different reference circuit – and thus a different WD network – along the boundary only. An example of this was given in section 3.2, above, in which the southern and northern boundary grid point simulations were carried out differently from those used at interior grid points because distinct reference circuits were applied.

To summarize, in order to determine if a given reference circuit and its corresponding WD network are able to accommodate a boundary condition imposed on a PDE system, one may

- Count the number of dependent variables and their first derivatives that appear in the PDE system. call this $\widehat{n}_{dv\&d}$.
- From the WD network, determine the number of waves that enter the boundary sampling point in question from *outside* the boundary. This will be n_{un}.
- Let the number of PDE's be called n_{eq} and the number of waves in the WD network be n_w.
- Then, if $(n_{eq}+1+(n_w-n_{un})) = \widehat{n}_{dv\&d}$, the boundary condition (43) may be accommodated while making use of the given reference circuit. If the number of equations in the overall set is only one short of $\widehat{n}_{dv\&d}$, then (47) (or its nonlinear counterpart) should be examined to determine if it is independent of the other equations. If so, then we may be sure that the boundary condition can still be modeled simultaneously with the PDE system. However, if the use of (43) and (47) still leaves us with fewer $\widehat{n}_{dv\&d}$ independent relationships, then a different reference circuit will need to be enunciated, at least at the boundary grid points in question.

Once a sufficient number of equations has been found, the simulation at the boundary may then proceed, either sequentially (wherein the lumped boundary condition simulation is time-advanced first, after which the WD network is advanced), or simultaneously (with the lumped and PDE systems being solved for during the same numerical steps.) See above for details.

5 Simulation of a Simple Acoustics Problem

As an example, we simulate an acoustics problem taken from [25], which is similar to the three-dimensional parallel plate problem.

We assume the medium (air) is a continuous, isotropic one, of uniform density and having the property of perfect (lossless) elasticity. The behavior of sound waves traveling through a two-dimensional crossection of this space can be characterized by the following equations, [25].

$$\frac{\delta(\rho v)}{\delta t_1} + \frac{\delta(\rho u)}{\delta t_2} = -\frac{\delta \rho}{\delta t_3} \tag{53}$$

$$-\frac{\delta p}{\delta t_1} = \frac{\delta(\rho v)}{\delta t_3} \tag{54}$$

$$-\frac{\delta p}{\delta t_2} = \frac{\delta(\rho u)}{\delta t_3} \tag{55}$$

where, t_1 and t_2 are two spatial coordinates and t_3 corresponds to time. ρ is the air density; p is the variation in the total pressure P due to the presence of the sound wave; v and u are the velocities along the t_1 and t_2 directions, respectively. Thus, ρv and ρu represent the mass flow per second per unit area along the t_1 and t_2 directions, respectively.

It can be shown that p and ρ are related, however, through the equations [25]

$$p = ßs, \qquad \rho = \rho_0(1+s) \tag{56}$$

where ρ_0 is the absolute density of the air with no wave disturbance present; s is called "condensation" which describes the instantaneous fractional change in density at a point in a field of sound; ß is the bulk modulus and is assumed to be a constant with little error for sound propagated in ordinary open air since any variations due to changes in atmospheric conditions or the presence of the sound wave itself are quite small.

As a result of (56), equations (54) and (55) can be written

$$\frac{\delta(\rho v)}{\delta t_3} + \frac{ß}{\rho_0}\frac{\delta \rho}{\delta t_1} = 0 \tag{57}$$

$$\frac{\delta(\rho u)}{\delta t_3} + \frac{ß}{\rho_0}\frac{\delta \rho}{\delta t_2} = 0 \tag{58}$$

Equation (53), (57) and (58) are very similar to equation (3), (1) and (2). In particular, if we choose in $(1-3)$

$$e_1(\mathbf{t}) = e_2(\mathbf{t}) = e_3(\mathbf{t}) = 0 \tag{59}$$

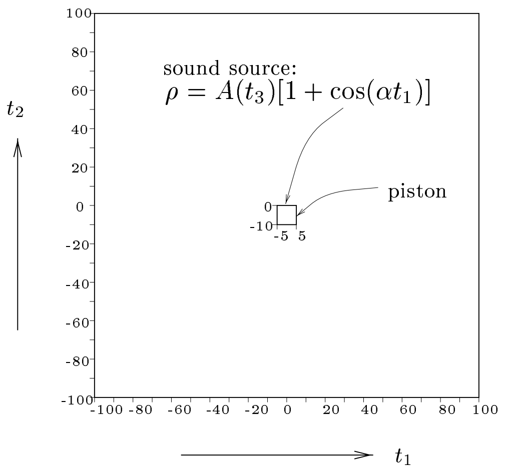

Figure 13: Crossectional drawing of a rigid acoustic piston centered within a rectangular boundary.

$$r = g = 0 \qquad c = 1 \tag{60}$$

$$l = \frac{\rho_0}{ß} \tag{61}$$

and define the dependent variables, i_1, i_2 and i_3, in $(1-3)$ as the ρv, ρu and $\frac{\rho}{r_3}$ in (53) and $(57-58)$, then the two sets of equations are identical.

We simulate the set of parallel plate equations, $(1-3)$, with the parameter choices of equations (59), (60) and (61), using the checkerboard sampling grid and its corresponding technique of accommodating boundary conditions.

Suppose that a rigid acoustic piston, which is 10 by 10 units in crossectional size, is enclosed by a set of rectangular walls whose plane coincide with that of the piston face, as shown in Figure 13. The rectangular region is 200 by 200 in crossectional size. The piston has a small sound source along its upper edge. It imposes a variation of pressure p at this upper edge that is defined by

$$p(t_1, t_3) = A(t_3)[1 + \cos(\alpha t_1)] \tag{62}$$

where

$$A(t_3) = \sin(2\pi \frac{v}{\lambda} t) \tag{63}$$

$A(t_3)$ is a function of time which specifies the amplitude of the sound source, v is the the speed of sound in air which is 330 m/s and λ is the wave length. $1 + \cos(\alpha t_1)$ is a function of the t_1 spatial coordinate where $\alpha = \frac{\pi}{10}$.

5.1 “Hard” Boundary Condition

Suppose that the other three sides of the piston and the four sides of the outer rectangular boundary have no other sound source. We first assume that these boundaries are all “hard” so that the flow per unit area perpendicular to these sides is zero. For example, the flow per unit area perpendicular to the bottom face of the piston, which is ρu, is set equal to zero. The flow per unit area perpendicular to the left or right faces of the piston, which is ρv, are also zero. Similar conditions are defined for the four sides of the outer rectangular boundary.

5.2 “Soft” Boundary Condition

Suppose the right side of the boundary is not a “hard” boundary while the other sides of the boundary and the three sides of the piston remain so. This implies that the right boundary will exhibit some elastic deformation when incoming waves strike it. Thus it will store some energy from the coming waves. This phenomenon can be expressed by

$$\rho v = K\frac{d}{dt}(p - p_{EX}) \tag{64}$$

where p_{EX} is a given exterior pressure and K is a parameter which specifies how “hard” or how “soft” the boundary is.

The boundary may also be porous and may therefore absorb some energy from the waves when they pass through boundary wall. This phenomenon can be expressed by

$$\rho v = G(p - p_{EX}) \tag{65}$$

where G is the boundary parameter which specifies the porosity of the boundary wall.

Due to these two effects, the “soft” boundary condition can be written as

$$\rho v = K\frac{d}{dt}(p - p_{EX}) + G(p - p_{EX}) \tag{66}$$

We will perform the simulations for “hard” boundary and “soft” boundary.

5.3 Simulation Results

The simulation results are shown in the following figures.

5.3.1 Wave Defraction

Figure 14 shows the defraction property of the sound wave. We can clearly see the hemispherical distribution of the sound waves traveling through space with alternating high air pressure and low air pressure. The simulation of Figure 14 is done with the ratio of the length of the upper face of the piston to the wavelength chosen to be approximately 1:4.

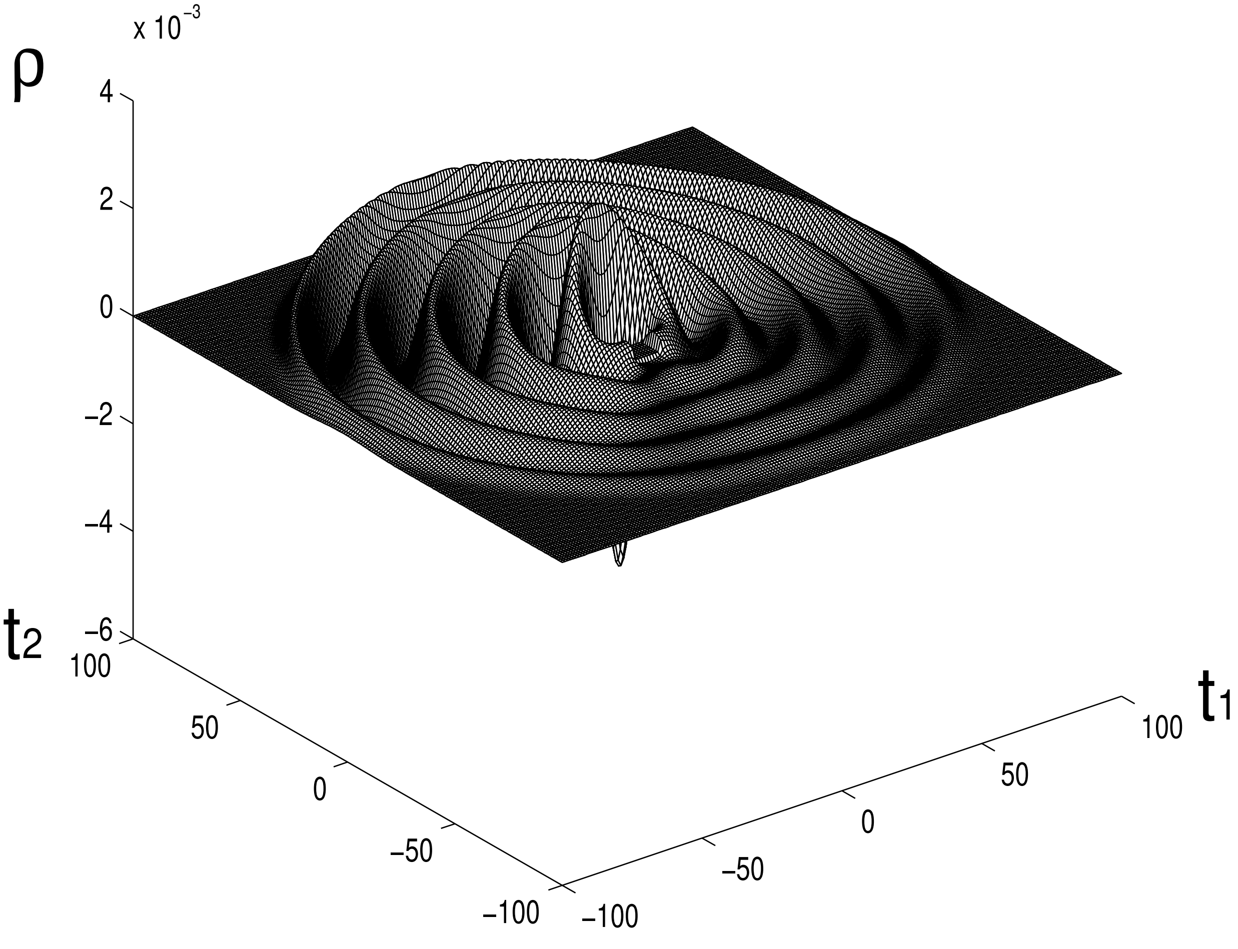

Figure 14: Defraction of acoustic waves

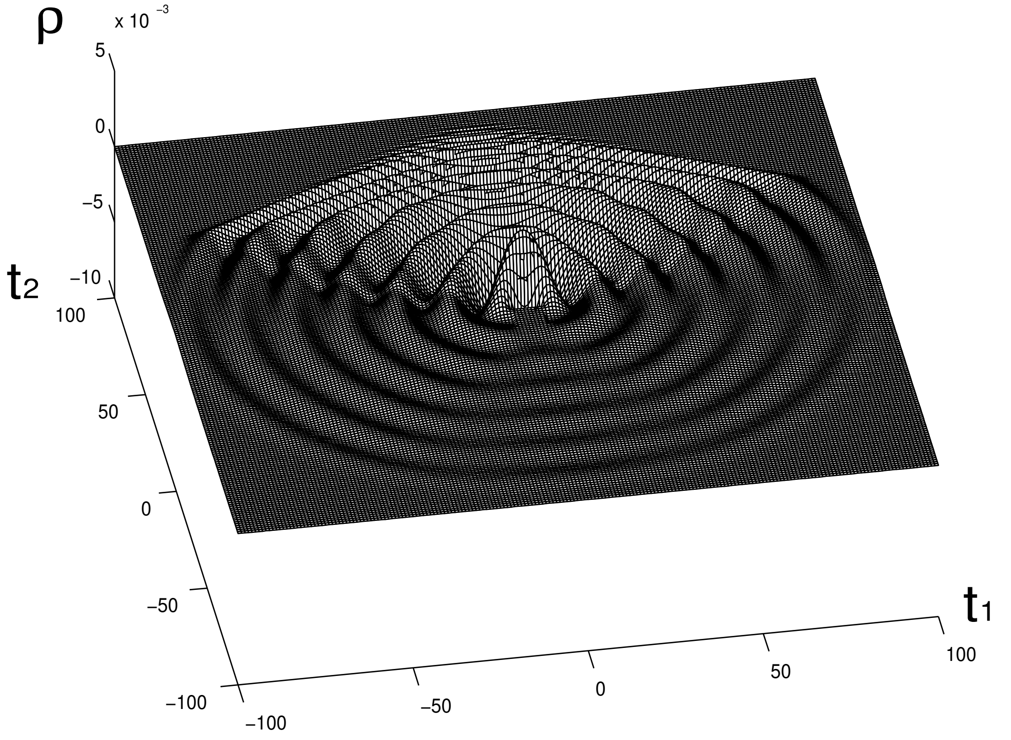

Figure 15: Refraction of acoustic waves

5.3.2 Wave Refraction

When the sound wave reaches a boundary where the nature of the medium changes, the wave will be partly reflected and partly transmitted. If the two media differ only slightly in their properties, most of the energy will be transmitted in the refracted waves into the second medium and but little will be reflected back from the surface. The transmitted wave will travel with a different velocity. This phenomenon is shown in Figure 15. The air density in the upper region ($t_2 \geq 50$) is defined to be four times greater than in the region $t_2 < 50$.

5.3.3 Wave Reflection

When the sound wave reaches a rigid boundary which absorbs no energy, the sound wave will be fully reflected.

If the boundary is a "soft" one, it will absorb energy from the incoming wave. Then the reflected wave from that boundary will have lower peak values compared to the wave reflected from a non-energy-absorbing boundary. In Figure 16, the boundary at $t_1 = 100$ is a purely

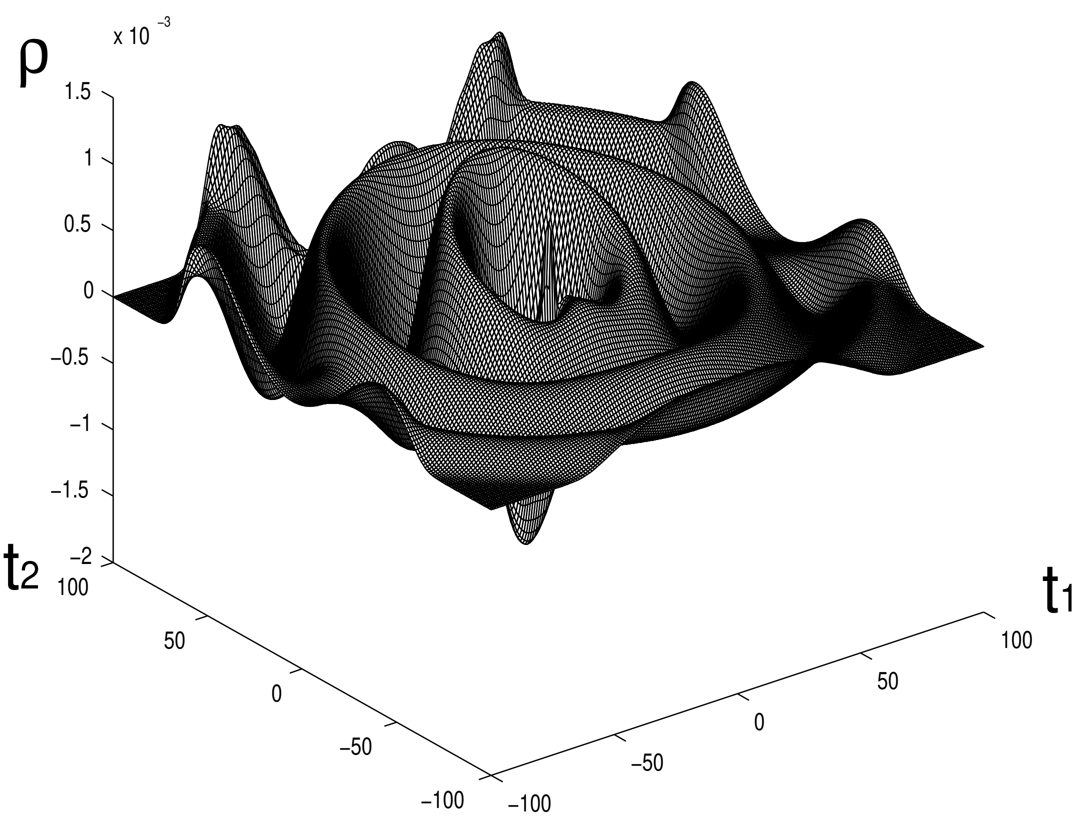

Figure 16: Reflection of acoustic waves

porous boundary which absorbs energy, while the boundary at $t_1 = -100$ is a fully reflective boundary which absorbs no energy. In order to see more clearly the wave behavior at these two boundaries, we plot these two boundary edges at the same time in Figure 18 and Figure 17, respectively.

In addition to Figure 16 - 18, Figure 19 shows the time functions of ρ and ρv at one point, $t_1 = 100$ and $t_2 = 0$, on the resistive boundary. The boundary condition is specified by (65). Clearly, we can see from Figure 19 that ρ and ρv behave exactly as in (65) – no phase difference between ρv and ρ.

If the boundary is a purely energy-storing (elastic) boundary, such as that described by (64), then we can expect a 90 degree phase difference between ρv and ρ. Figure 20 shows the time functions of ρ and ρv at one point, $t_1 = 100$ and $t_2 = 0$, on such a boundary where this 90

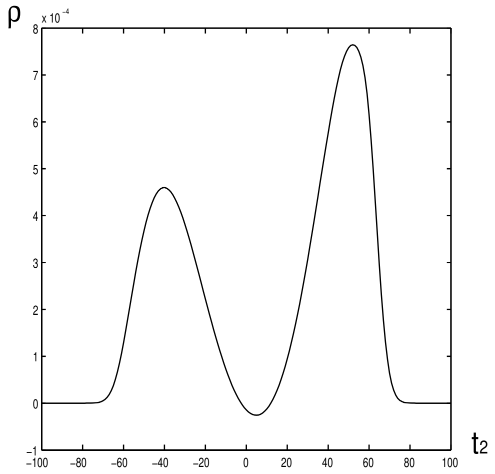

Figure 17: Wave behavior at a non-energy-absorbing boundary when $t_1 = -100$ and $t_3 = 120$

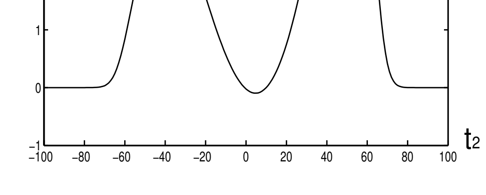

Figure 18: Wave behavior along a purely porous boundary when $t_1 = 100$ and $t_3 = 120$

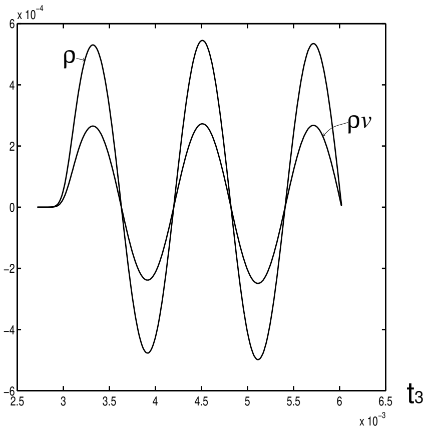

Figure 19: Time functions of ρ and ρv at $t_1 = 100$ and $t_2 = 0$ when the boundary is a purely porous one.

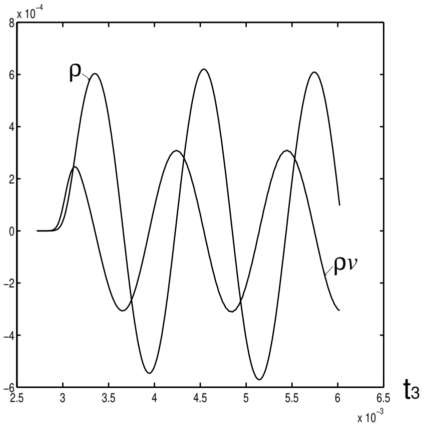

Figure 20: Time functions of ρ and ρv at $t_1 = 100$ and $t_2 = 0$ when the boundary is a purely elastic one.

degree phrase difference between ρv and ρ is evident. If the boundary not only stores energy but also absorbs energy from the incoming wave due to porosity, then we may set the boundary condition at $t_1 = 100$ according to (66). The time functions of ρ and ρv at one point, $t_1 = 100$ and $t_2 = 0$, on this boundaries is shown in Figure 21. We can see from Figure 21 that the phase difference between ρv and ρ is not as great as in the purely elastic boundary case.

These results help confirm the correctness of the wave digital simulation technique discovered by Fettweis [1, 2, 3] together with the method for accommodating boundary conditions developed here.

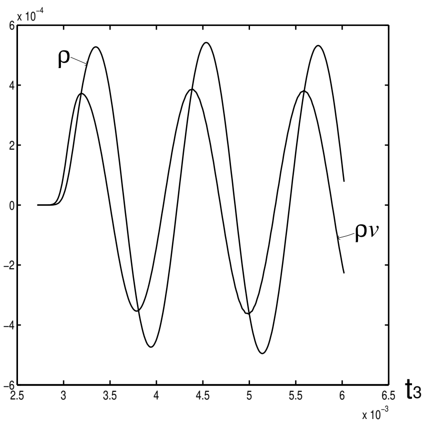

Figure 21: Time functions of ρ and ρv at $t_1 = 100$ and $t_2 = 0$ when the boundary is both porous and elastic one.

6 Conclusions

We have developed a method for incorporating lumped boundary conditions – either linear or nonlinear – into wave digital simulations of higher-dimensional PDE systems. This procedure may be generalized and immediately applied to many hyperbolic system of PDE's, including nonlinear ones.

We have also proposed a simple "equation counting" method to determine in any specific case if the procedure is or is not applicable, and provided examples of corrective actions that may be taken in those cases in which the counting method indicates that the procedure is not immediately applicable.

A three-dimensional parallel plate problem has been used as a running example to illustrate the detailed procedure.

Results of several simulations of an acoustics problem (which is very similar to the three-dimensional parallel plate problem) are shown.

Finally, the authors wish to express their sincere appreciation to Dr. Alfred L. Fettweis for the many fruitful discussions and insights he provided us during his 1994 - 96 appointment as Visiting Distinguished Professor at the University of Notre Dame.

References

[1] A. Fettweis, "Wave Digital Filters: Theory and Practice," *Proc. IEEE*, Vol. 74, No. 2, pp. 270-327, February 1986.

[2] A. Fettweis and G. Nitsche, "Transformation Approach to Numerically Integrating PDE's by Means of WDF Principles," *Multidimensional Systems and Signal Processing*, Vol. 2, pp. 127-159, 1991.

[3] A. Fettweis and G. Nitsche, "Numerical Integration of Partial Differential Equations Using Principles of Multidimensional Wave Digital Filters," *J. of VLSI Signal Processing*, Vol. 3, pp. 7-24, 1991.

[4] A. Fettweis, "Discrete Passive Modelling of Physical Systems Descibed by PDEs," *Proc. EUSIPCO-92, Sixth European Signal Processing Conf.,* Brussels, August 25-28, 1992.

[5] X. Liu and A. Fettweis, "Multidimensional Digital Filtering by Using Parallel Algorithms Based on Diagonal Processing," *Multidimensional Systems and Signal Processing*, Vol. 1, pp. 51-66, 1990.

[6] A. Fettweis, "Discrete Modelling of Lossless Fluid Dynamic Systems," *Internat. J. of Electronics and Communications*, Vol. 46, No. 4, pp. 209-218, 1992.

[7] A. Fettweis, "The Role of Passivity and Losslessness in Multidimensional Digital Signal Processing – New Challenges," *Proc. 1991 IEEE Internat. Symp. on Circuits and Systems*, Singapore, June 11-14, 1991, Vol. 1, pp. 112-115.

[8] A. Fettweis, "Discrete Passive Modelling of Viscous Fluids," *Proc. 1992 IEEE Internat. Symp. on Circuits and Systems*, San Diego, May 10-13, 1992, pp. 640-643.

[9] A. Fettweis, "Multidimensional Wave Digital Filters for Discrete-Time Modelling of Maxwell's Equations," *Internat. J. of Numerical Modelling: Electronic Networks, Devices and Fields*, Vol. 5, pp. 183-201, 1992.

[10] A. Fettweis, "On Assessing Robustness of Recursive Digital Filters," *ETT*, Vol. 1, No. 2, March-April 1990, pp. 103-109.

[11] K. Balemarthy, "Solutions of Systems of Partial Differential Equations via Wave Digital Filters," M.S. Thesis, Department of Comp. Sci. and Engineering, Univ. of Notre Dame, Notre Dame, IN, USA, November 1994.

[12] S. Bass, "Sampling Grid Properties in Wave Digital PDE Simulations," Technical Report, Univ. of Notre Dame, USA, Department of Computer Science and Engineering, CSE-TR-26-94, October 19, 1994.

[13] D. Kincaid and W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing,* Brooks/Cole Publishing Co., Pacific Grove, Calif., 1991.

[14] K. Balemarthy and S.C. Bass, "General, Linear Boundary Conditions in MD Wave Digital Simulations," Proceeding 1995 Internat. Symp. on Circuits and Systems, Seattle, Wash., April 29 - May 3, 1995, pp. I73-I76

[15] Louis Weinberg, *Network Analysis and Synthesis*, McGraw-Hill, 1962.

[16] A. Fettweis, "Multidimensional Wave Digital Filters - Problems and Progress," Proceeding 1986 IEEE Int. Symp. on Circuits and Systems, Ssn Jose, CA., May 5-7, 1986, pp. 506-509

[17] A. Fettweis, "Multidimensional Wave-Digital Principles: From Filtering to Numerical Integration," Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing, Adelaide, Queensland, Australia, April 19-22, 1994, Vol. 6, pp. 173-181

[18] A. Fettweis, "Discrete Passive Modelling of Physical Systems Described by Partial Differential Equations," *Multivariate Analysis: Future Directions*, Chapter 7, pp. 115-130, 1993.

[19] T. Leichel and A. Fettweis, "Implementation of Modified Wave Digital Filters by Floating-point Digital Signal Processors," *Proceedings, 6th European Signal Processing Conference*, Brussels, Belgium, Aug. 24-27, 1992, pp. 265-268

[20] A. Fettweis and T. Leichel, "On Floating-point Implementation of Modified Wave Digital Filters," *Proceedings IEEE International Symposium on Circuits and Systems*, San Diego, CA., May 10-13, 1992, pp. 1812-1815

[21] A. Fettweis, "On Improved Representation Theorems for Multidimensional Lossless Bounded Matrices," *International J. of Circuit Theory and Applications*, Vol. 19, pp. 453-457, 1991.

[22] A. Fettweis, "New Results in Wave Digital Filtering," *Proceedings 1989 URSI Int. Symposium on Signals, Systems, and Electronics*, Erlangen, W. Germany, Sept. 18-20, 1989, pp. 17-23

[23] A. Fettweis, "Passivity and Losslessness in Digital Filtering," *(AEU), Electronics and Communication*, pp. 1-8, 1988.

[24] A. Fettweis and T. Leichel and M. Bolle and U. Sauvagerd, "Realization of Filter Banks by Means of Wave Digital Filters," *Proceedings, 1990 IEEE International Symposium on Circuits and Systems*, New Orleans, Louisiana, May 1-3, 1990, pp. 2013-2016

[25] R. H. Randall, *An Introduction to Acoustics,* Addison-Wesley Press, Inc., Cambridge, Mass, 1951.

[26] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A systems Perspective, Second Edition*, Addison-Wesley Pub. Co., Reading, Mass., 1993.

[28] P.M. Kogge, "EXECUBE – A New Architecture for Scaleable MPP's," *Proc. 1994 Internat. Conf. on Parallel Processing*, St. Charles, IL, August 15-18, 1994.

[29] H.W. Emmons, "Critique of Numerical Modeling of Fluid Mechanics Phenomena," Ann. Rev. Fluid Mech, Vol. 2, pp. 15-36, 1970.

[30] H. Van Dyke, *An Album of Fluid Motion*, Parabolic Press, Stanford, CA, 1982.

[31] A. Jameson, "Universities Foster CFD Growth," *Aerospace America*, pp. 42-47, Feb. 1992

[32] Q. Xu and S.C. Bass, "Accomodating Lumped, Linear Boundary Conditions In The Wave Digital Simulations of PDE Systems", Technical Report, University of Notre Dame, USA, Department of Computer Science and Engineering, CSE-TR96-29, October, 1996.