NASA/CR-1999-209350
ICASE Report No. 99-23

Long-time Numerical Integration of the Three-dimensional Wave Equation in the Vicinity of a Moving Source

V.S. Ryaben'kii and V.I. Turchaninov
Keldysh Institute for Applied Mathematics, Moscow, Russia

S.V. Tsynkov
Tel Aviv University, Tel Aviv, Israel
and
ICASE, Hampton, Virginia

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA
Operated by Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NAS1-97046

June 1999

LONG-TIME NUMERICAL INTEGRATION OF THE THREE-DIMENSIONAL WAVE EQUATION IN THE VICINITY OF A MOVING SOURCE*

V. S. RYABEN'KII[†], V. I. TURCHANINOV[‡], AND S. V. TSYNKOV[§]

Abstract. We propose a family of algorithms for solving numerically a Cauchy problem for the three-dimensional wave equation. The sources that drive the equation (i.e., the right-hand side) are compactly supported in space for any given time; they, however, may actually move in space with a subsonic speed. The solution is calculated inside a finite domain (e.g., sphere) that also moves with a subsonic speed and always contains the support of the right-hand side.

The algorithms employ a standard consistent and stable explicit finite-difference scheme for the wave equation. They allow one to calculate the solution for arbitrarily long time intervals without error accumulation and with the fixed non-growing amount of the CPU time and memory required for advancing one time step. The algorithms are inherently three-dimensional; they rely on the presence of lacunae in the solutions of the wave equation in oddly dimensional spaces.

The methodology presented in the paper is, in fact, a building block for constructing the nonlocal highly accurate unsteady artificial boundary conditions to be used for the numerical simulation of waves propagating with finite speed over unbounded domains.

Key words. wave equation, lacunae, finite-difference approximation, explicit numerical integration, arbitrarily long time intervals, non-accumulation of error, uniform error bounds, fixed expenses per time step

Subject classification. Applied and Numerical Mathematics

1. Introduction.

We will be solving numerically a Cauchy problem for the three-dimensional wave equation:

$$\frac{\partial^2 u}{\partial t^2} - c^2\left(\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \frac{\partial^2 u}{\partial x_3^2}\right) = f(\bar{x}, t), \quad t \geq 0, \tag{1.1}$$

$$u\Big|_{t=0} = \frac{\partial u}{\partial t}\Big|_{t=0} = 0. \tag{1.2}$$

[*] This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1–97046 while the first and third authors were in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681–2199. The work of the first and third authors was supported by Director's Discretionary Fund, NASA Langley Research Center, Hampton, VA 23681–2199.

[†] Keldysh Institute for Applied Mathematics, Russian Ac. Sci., 4 Miusskaya Sq., Moscow 125047, Russia; and ICASE, MS 132C, NASA Langley Research Center, Hampton, VA 23681–2199, USA.

[‡] Keldysh Institute for Applied Mathematics, Russian Ac. Sci., 4 Miusskaya Sq., Moscow 125047, Russia.

[§] Corresponding author. School of Mathematical Sciences, Tel Aviv University, Ramat Aviv, Tel Aviv 69978, Israel; and ICASE, MS 132C, NASA Langley Research Center, Hampton, VA 23681–2199, USA. E-mail: tsynkov@math.tau.ac.il, URL: http://www.math.tau.ac.il/∼tsynkov/

We will be interested in calculating the solution $u = u(\bar{x},t)$ only for those values of the argument $\bar{x} = (x_1,x_2,x_3)$ that belong to the ball $S = S(t)$ defined by the following inequality

$$(x_1 - x_1^0)^2 + (x_2 - x_2^0)^2 + (x_3 - x_3^0)^2 \le d^2/4. \tag{1.3}$$

In inequality (1.3), d is the diameter of the sphere $S(t)$, and the functions $x_1^0 = x_1^0(t)$, $x_2^0 = x_2^0(t)$, and $x_3^0 = x_3^0(t)$ are given smooth functions of their argument t — time; these functions define the motion of the sphere's center so that

$$\left(\frac{dx_1^0}{dt}\right)^2 + \left(\frac{dx_2^0}{dt}\right)^2 + \left(\frac{dx_3^0}{dt}\right)^2 \le k^2, \tag{1.4}$$

where $k < c$. In other words, we consider only subsonic motions of the sphere $S(t)$, the maximum speed k of its center may never exceed the speed of sound c (see equation (1.1)). Regarding the right-hand side $f(\bar{x},t)$ of equation (1.1), we always assume that it is a sufficiently smooth function with respect to all its four arguments and also

$$\operatorname{supp} f(\bar{x},t) = \{\, (x_1,x_2,x_3,t) |\, \bar{x} \in S(t),\ t > 0 \} \,. \tag{1.5}$$

In this paper, we construct efficient numerical algorithms for solving the foregoing problem. These algorithms employ the simplest central finite-difference scheme that approximates problem (1.1), (1.2) on smooth solutions with the second order of accuracy. This scheme has been chosen primarily for the reason of simplicity so that we can demonstrate the concept using the least cumbersome approach. Generally, any scheme that possesses the properties of stability and consistency on smooth solutions, including high-order schemes, can be used for building the algorithms of the type described hereafter.

The second order central-difference scheme is constructed on the uniform Cartesian grid with the size h in all spatial directions and time step τ:

$$\Big\{(x_{m_1},y_{m_2},z_{m_3},t_{m_4}) = (m_1h, m_2h, m_3h, m_4\tau)\Big| m_1,m_2,m_3 = 0,\pm1,\pm2,\dots, m_4 = 0,1,2,\dots\Big\}.$$

In every grid node $m \equiv (m_1,m_2,m_3,m_4)$, equation (1.1) is replaced by the finite-difference equation

$$\sum_{n\in N_m} a_{mn}u_n = f_m, \qquad m_4 = 2,3,4,\dots\,. \tag{1.6}$$

The discrete solution $u_n \equiv u_{n_1,n_2,n_3,n_4}$ is defined on the same grid: $n_1,n_2,n_3 = 0,\pm1,\pm2,\dots$, $n_4 = 0,1,2,3,\dots$; equation (1.6) connects the values of u_n in the following nine nodes of the grid that form the stencil N_m:

$$N_m = \begin{cases} (m_1h, m_2h, m_3h, (m_4-j)\tau), \quad j = 0,1,2, \\ ((m_1\pm1)h, m_2h, m_3h, (m_4-1)\tau), \\ (m_1h, (m_2\pm2)h, m_3h, (m_4-1)\tau), \\ (m_1h, m_2h, (m_3\pm1)h, (m_4-1)\tau). \end{cases} \tag{1.7}$$

The coefficients a_{mn} of the discrete operator and the values f_m of the discrete right-hand side in (1.6) are defined as follows

$$a_{mn} = \begin{cases} 1, & \text{if } \ n = (m_1, m_2, m_3, m_4), \ n = (m_1, m_2, m_3, m_4 - 2) \\ -r^2, & \text{if } \ n = (m_1 \pm 1, m_2, m_3, m_4 - 1), \\ -r^2, & \text{if } \ n = (m_1, m_2 \pm 1, m_3, m_4 - 1), \\ -r^2, & \text{if } \ n = (m_1, m_2, m_3 \pm 1, m_4 - 1), \\ -2 + 6r^2, & \text{if } \ n = (m_1, m_2, m_3, m_4 - 1), \end{cases} \tag{1.8}$$

$$f_m = \tau^2 f(m_1 h, m_2 h, m_3 h, (m_4 - 1)\tau h), \tag{1.9}$$

where r is the Courant number

$$r = \frac{\tau}{h} \leq \frac{1}{c\sqrt{3}}; \tag{1.10}$$

estimate (1.10) immediately follows from the standard stability considerations of von Neumann type. Initial conditions (1.2) are replaced by the conditions

$$u^p_{n_1, n_2, n_3} = 0, \qquad p \equiv n_4 = 0, 1. \tag{1.11}$$

It is known that if the right-hand side $f(\bar{x}, t)$ is sufficiently smooth with respect to all its arguments, then the finite-difference solution

$$u_n \equiv u_{n_1, n_2, n_3, n_4} \equiv u^p_{n1, n_2, n_3},$$

where $p = n_4$, converges to the continuous solution $u(\bar{x}, t)$ of problem (1.1), (1.2) as the grid size decreases, and the following estimate holds

$$\max_{n_1, n_2, n_3, p} \left| u(n_1 h, n_2 h, n_3 h, p\tau) - u^p_{n_1, n_2, n_3} \right| \leq C|f|_{K,T} h^2, \quad p\tau \leq T_{\text{final}}, \tag{1.12}$$

here $C = C(T_{\text{final}})$ is a constant, K is a sufficiently large positive integer number, and $|f|_{K,T}$ is the sum of maximal absolute values of all derivatives of the function $f(\bar{x}, t)$ up to the order K for $t \leq T_{\text{final}}$. Therefore, speaking formally, estimate (1.12) allows one to use the finite-difference scheme (1.6), (1.11) for calculating the solution $u(\bar{x}, t)$ of problem (1.1), (1.2) on arbitrarily long time intervals $0 \leq t \leq T_{\text{final}}$.

There are, however, two most substantial obstacles. First, when calculating the solution using the scheme (1.6), (1.11), the number of grid nodes involved in the computation on each time level increases approximately as $(d/h + p)^3$ with the number of level p. Consequently, when $p \approx T_{\text{final}}/\tau$ and $\tau = rh$, $r = 1/c\sqrt{3}$, this number is of the same order of magnitude as $\left(\frac{T_{\text{final}}}{h}\right)^3$. Therefore, the associated storage and CPU time requirements grow rapidly as the final time T_{final} increases. Second, estimate (1.12) basically does not guarantee the algorithm from the possible error accumulation as the final time T_{final} and the corresponding number of steps increase. Indeed, although the constant C in inequality (1.12) does not depend on the grid size h, it, generally speaking, depends on the final time T_{final} (may grow with T_{final}).

In the paper, we propose three approaches to improve the simplest scheme (1.6), (1.11); these approaches take into account some specific properties of the solution to problem (1.1), (1.2). Similar approaches can be developed for other converging schemes, besides the scheme (1.6), (1.11). Each of the three algorithms proposed hereafter guarantees that the error will not accumulate as the number p of the time level increases. Moreover, both the memory and CPU time required for advancing each time step remain bounded independently of p (and T_{final}) for the fixed grid sizes h and τ.

The number of arithmetic operations required for advancing one time step using either of the three algorithms that we are discussing is $\mathcal{O}(N)$, where N is the number of grid nodes in space (i.e., on one time level) inside the sphere of the fixed diameter d; clearly, $N = \mathcal{O}(h^{-3})$. This number does not depend on t, i.e., does not increase with p because unlike in the original scheme (1.6), (1.11), the computational domain in the new algorithm will not need to expand in space as the time goes by. Obviously, the number $\mathcal{O}(N)$ is optimal (linear with respect to the grid dimension) and cannot be improved by choosing any other algorithm. The required memory (number of words) in all three versions of the new algorithm is of the order $\mathcal{O}(N)$ as well, i.e., does not exceed some $C \cdot N$, where C is a constant that depends on the version. For the third algorithm this constant is smaller than for the others.

All three aforementioned methodologies for improving the original scheme (1.6), (1.11) so that one can calculate the solution $u(\bar{x}, t)$ of problem (1.1), (1.2) on the domain $\bar{x} \in S(t)$ and arbitrarily large time intervals rely on a particular property of solutions to the three-dimensional wave equation (1.1), namely *the property of having lacunae.*

The lacunae-based technique proposed in this paper for the long-time numerical integration of the three-dimensional wave equation in the neighborhood of a moving source is a generalization and extension of the technique developed previously in [1] for the case of stationary sources. In the future, these lacunae-based techniques will be used as a building block for constructing global artificial boundary conditions (ABCs) for the numerical simulation of waves propagating with finite speed over infinite domains. The latter framework includes, in particular, the problems of electromagnetic diffraction and scattering, as well as the problems in both ambient and advective acoustics. The unsteady ABCs' methodology that we have mentioned will be described in a forthcoming publication. A general review of different ABCs' methodologies available in the literature can be found in [2].

2. Lacunae of the Wave Equation. Consider the nonhomogeneous wave equation with zero initial conditions:

$$\frac{\partial^2 v}{\partial t^2} - c^2 \left(\frac{\partial^2 v}{\partial x_1^2} + \frac{\partial^2 v}{\partial x_2^2} + \frac{\partial^2 v}{\partial x_3^2} \right) = \varphi(\bar{x}, t), \quad t \geq 0, \tag{2.1}$$

$$v\Big|_{t=0} = \frac{\partial v}{\partial t}\Big|_{t=0} = 0. \tag{2.2}$$

Assume that $\varphi(\bar{x}, t)$ is a sufficiently smooth function with respect to all its arguments and that $\varphi(\bar{x}, t) = 0$ for $t \leq 0$.

For every $(\bar{x}, t)$, the solution $v = v(\bar{x}, t)$ of problem (2.1), (2.2) can be written in the form of the Kirchhoff integral:

$$v(\bar{x}, t) = \frac{1}{4\pi c^2} \iiint\limits_{\rho \leq ct} \frac{\varphi\left(\xi, t - \rho/c\right)}{\rho} d\bar{\xi}, \tag{2.3}$$

where $\rho = \sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2 + (x_3 - \xi_3)^2}$ and $d\bar{\xi} = d\xi_1 d\xi_2 d\xi_3$. The integration in (2.3) is performed over the ball of radius ct centered at $\bar{x}$ in the space $\bar{\xi} = (\xi_1, \xi_2, \xi_3)$. Formula (2.3), in fact, implies that the solution $v(\bar{x}, t)$ at the point $(\bar{x}, t)$ depends only on the values of $\varphi(\bar{\xi}, \theta)$ on the surface of the characteristic cone (its lower portion) with the vertex $(\bar{x}, t)$:

$$(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2 + (x_3 - \xi_3)^2 = c^2(t - \theta)^2, \quad \theta < t \tag{2.4a}$$

and does not depend on $\varphi(\bar{\xi}, \theta)$ when $(\bar{\xi}, \theta)$ belongs to the interior of the cone (2.4a). In other words, changing the values of $\varphi(\bar{\xi}, \theta)$ in the interior of the cone (2.4a) will not affect the solution $v(\bar{x}, t)$ at the point $(\bar{x}, t)$. To emphasize this circumstance, we will call the domain

$$(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2 + (x_3 - \xi_3)^2 < c^2(t - \theta)^2, \quad \theta < t, \tag{2.4b}$$

i.e., the interior of the characteristic cone (2.4a) *the lacuna of the right-hand side* of equation (2.1) with respect to the point $(\bar{x}, t)$. The presence of the lacuna (2.4b) of the right-hand side implies that *the solution $v(\bar{x}, t)$ of (2.1), (2.2) will also have a lacuna* $D^+(Q)$ with respect to the domain Q of the right-hand side. Indeed, consider a δ-source for equation (2.1) concentrated at the point $(\bar{\xi}, \theta)$ of the space-time: $\delta(\bar{\xi}, \theta)$. At any moment of time $t > \theta$, the solution of problem (2.1), (2.2) driven by this source will be concentrated on the surface of the sphere of radius $c(t - \theta)$ centered at $\bar{\xi}$ in the space $\bar{x} = (x_1, x_2, x_3)$. Inside this sphere, the solution will be identically zero: $v(\bar{x}, t) \equiv 0$ for $\rho \equiv |\bar{x} - \bar{\xi}| < c(t - \theta)$. Therefore, let us now interpret the surface (2.4a) as the upper portion of the characteristic cone of equation (2.1) in the space-time $(\bar{x}, t)$ with the vertex $(\bar{\xi}, \theta)$. Then, the solution of (2.1), (2.2) driven by $\delta(\bar{\xi}, \theta)$ is zero in the interior of the cone (2.4a), i.e., on the domain (2.4b) that we now denote by $D^+(\bar{\xi}, \theta)$,

$$D^+(\bar{\xi}, \theta) = \left\{ (\bar{x}, t) \middle| \, |\bar{x} - \bar{\xi}| < c(t - \theta), \; t > \theta \right\}, \tag{2.4c}$$

and call *the lacuna of the fundamental solution of the wave equation.* (Note, this fundamental solution is actually a single layer on the spherical surface $|\bar{x} - \bar{\xi}| = c(t - \theta)$, $t > \theta$.) If we consider a general source $\varphi(\bar{\xi}, \theta)$ rather than the δ-source $\delta(\bar{\xi}, \theta)$, then for every particular $(\bar{\xi}, \theta)$ the solution of (2.1), (2.2) inside the lacuna $D^+(\bar{\xi}, \theta)$ given by (2.4c) does not depend on the value of $\varphi(\bar{\xi}, \theta)$ at this point $(\bar{\xi}, \theta)$. By the superposition principle, the solution of (2.1), (2.2) with a general source φ will be concentrated on the set given by the union of all spheres $|\bar{x} - \bar{\xi}| = c(t - \theta)$, $t > \theta$, when the vertex $(\bar{\xi}, \theta)$ of the cone (2.4a) sweeps the support of the right hand side $\operatorname{supp} \varphi(\bar{\xi}, \theta)$. Accordingly, the intersection of all $D^+(\bar{\xi}, \theta)$ of (2.4c) for all $(\bar{\xi}, \theta) \in Q$ will be called *the lacuna of the solution $v(\bar{x}, t)$ with respect to the domain Q*:

$$D^+(Q) = \bigcap_{(\bar{\xi},\theta)\in Q} D^+(\bar{\xi},\theta). \tag{2.4d}$$

Clearly, the solution $v(\bar{x},t)$ of (2.1), (2.2) is zero on $D^+(Q)$ of (2.4d),

$$v(\bar{x},t) \;\text{ for }\; (\bar{x},t) \in D^+(Q) \tag{2.5}$$

if

$$\operatorname{supp}\varphi \subseteq Q. \tag{2.6}$$

Alternatively, one can say that changing the values of $\varphi(\bar{\xi},\theta)$ in the domain Q is not going to affect the solution $v(\bar{x},t)$ of (2.1), (2.2) in the points of the lacuna $D^+(Q)$ given by (2.4d). In other words, we see that the waves governed by the three-dimensional wave equation (2.1) have *trailing fronts.* If the source is compactly supported in both space and time, then at any given location $\bar{x}$ in space the solution $v(\bar{x},t)$ becomes identically zero after a finite interval of time. This finite time interval is the time from the moment of source activation till the moment when the point $\bar{x}$ falls into the lacuna $D^+(Q)$ given by (2.4d), or in other words, till all the waves generated by the source have pass through $\bar{x}$ and accordingly, the solution there become zero again.

If the domain Q is defined as follows

$$Q = \left\{ (\bar{x},t) \,|\, \bar{x} \in S(t),\; t_0 < t < t_1 \right\}, \tag{2.7}$$

then condition (2.6) implies that that the solution $v(\bar{x},t)$ of (2.1), (2.2) satisfies the identities

$$v(\bar{x},t) \equiv 0, \quad \text{for} \;\; t \le t_0 \tag{2.8a}$$

and

$$v(\bar{x},t) \equiv 0, \quad \text{for} \;\; \bar{x} \in S(t), \quad t \ge t_0 + \frac{d + (t_1 - t_0)(c+k)}{c-k}. \tag{2.8b}$$

The first identity, (2.8a), is obvious, it takes place because the initial data of the Cauchy problem are homogeneous, see (2.2). The second identity, (2.8b) holds in virtue of (2.6) because the region of the space-time $(\bar{x},t)$ defined as $\left\{ \bar{x} \in S(t),\; t \ge t_0 + \frac{d+(t_1-t_0)(c+k)}{c-k} \right\}$ is completely contained inside the lacuna $D^+(Q)$ of (2.4d). *In other words, as long as (1.4) holds the time interval $\frac{d+(t_1-t_0)(c+k)}{c-k}$ is sufficient for all the waves generated by the sources inside $S(t)$ during $t_0 \le t \le t_1$ to completely leave the moving domain $S(t)$.* For the case of stationary sources, $k = 0$, the inequality of (2.8b) reduces to the obvious estimate $t \ge t_1 + d/c$, see [1]. Let us also note that the estimate for t given in (2.8b) is, in fact, conservative, it does not make any assumptions regarding the character of the source motion except that its maximal speed is $k < c$. To obtain (2.8b), one only assumes that at any $t > t_0$ the source, i.e., $S(t)$, can be anywhere inside the sphere of diameter $d + 2k(t - t_0)$ centered at the center of $S(t_0)$. If, however, we make an additional assumption regarding the motion of the sphere, e.g., that it moves with a constant speed k is some prescribed direction, then the estimate of (2.8b) can be alleviated and instead we obtain

$$v(\bar{x}, t) \equiv 0, \quad \text{for} \quad \bar{x} \in S(t), \quad t \geq t_1 + \frac{d}{c - k}. \tag{2.8c}$$

For a stationary source, $k = 0$, (2.8c) again reduces to $t \geq t_1 + d/c$ of [1].

Let us now introduce the following partition of unity. Define the function

$$\Psi(t) = \begin{cases} 1, & 0 \leq t \leq 1/2 \\ P(t), & 1/2 \leq t \leq 1 \\ 0, & t > 1 \\ \Psi(-t), & t \leq 0 \end{cases} \tag{2.9}$$

where $P(t)$ is a polynomial of the type

$$P(t) = \frac{1}{2} + a\left(t - \frac{3}{4}\right) + b\left(t - \frac{3}{4}\right)^3 + c\left(t - \frac{3}{4}\right)^5 + d\left(t - \frac{3}{4}\right)^7 + e\left(t - \frac{3}{4}\right)^9$$

with the coefficients a, b, c, d, and e such that the following equalities hold

$$P(1) = P'(1) = P''(1) = P'''(1) = P^{(IV)}(1) = 0.$$

Obviously, $P(1/2) = 1$ and the derivatives of $P(t)$ up to the fourth order are equal to zero at $t = 1/2$. Therefore, $\Psi(t)$ is an even function with four continuous derivatives for all $t \in \mathbb{R}$ and also $\Psi(t)$ is compactly supported, $\Psi(t) \equiv 0$ for $|t| \geq 1$, i.e.,

$$\text{supp}\Psi(t) = [-1,\, 1].$$

Specify now a parameter T and introduce the functions

$$\Psi_j(t, T) = \Psi\left(\frac{t - jT}{T}\right), \quad j = 0, 1, 2, \ldots$$

Clearly,

$$\text{supp}\Psi_j(t, T) = [(j-1)T,\, (j+1)T], \quad j = 0, 1, 2, \ldots$$

Moreover, for any $T > 0$ we have

$$\sum_{j=0}^{\infty} \Psi_j(t,T) \equiv 1, \quad t \geq 0. \tag{2.10}$$

The representation of a function which is identically equal to 1 in the form (2.10) is a partition of unity. Notice that for every given t no more than two terms on the left-hand side of the identity (2.10) may differ from zero.

We now represent the right-hand side $f(\bar{x},t)$ of equation (1.1) in the form

$$f(\bar{x},t) = f(\bar{x},t)\sum_{j=0}^{\infty} \Psi_j(t,T) = \sum_{j=0}^{\infty} \Psi_j(t,T) f(\bar{x},t) = \sum_{j=0}^{\infty} f_j(\bar{x},t,T), \tag{2.11}$$

where $f_j(\bar{x},t,T) = \Psi_j(t,T)f(\bar{x},t)$. Clearly,

$$\text{supp} f_j(\bar{x},t) = Q_j(T), \tag{2.12}$$

$$Q_j(T) = \{\, (\bar{x},t) |\, \bar{x} \in S(t),\ (j-1)T \leq t \leq (j+1)T \} \,. \tag{2.13}$$

Consider the following sequence of problems

$$\begin{gathered} \frac{\partial^2 u_j}{\partial t^2} - c^2 \left(\frac{\partial^2 u_j}{\partial x_1^2} + \frac{\partial^2 u_j}{\partial x_2^2} + \frac{\partial^2 u_j}{\partial x_3^2} \right) = f_j(\bar{x},t,T), \\ u_j\Big|_{t=0} = \frac{\partial u_j}{\partial t}\Big|_{t=0} = 0, \quad j = 0,1,2,\ldots \end{gathered} \tag{2.14}$$

Because of the linearity of problem (1.1), (1.2) and representation of $f(\bar{x},t)$ in the form (2.11), the solution $u(\bar{x},t)$ of problem (1.1), (1.2) can also be represented as a similar sum

$$u(\bar{x},t) = \sum_{j=0}^{\infty} u_j(x,t,T), \tag{2.15}$$

where $u_j(x,t,T)$ is the solution of problem (2.14). Let us show that for $\bar{x} \in S(t)$ and any fixed $t > 0$ there are only a few values of j for which $u_j(\bar{x},t,T) \neq 0$. First, we apply identities (2.8a) and (2.8b) which hold under conditions (2.6), (2.7) to the solution $u_j(\bar{x},t,T)$ of problem (2.14). In so doing, instead of (2.6), (2.7) we use (2.12), (2.13). Then, instead of (2.8a) and (2.8b) we obtain the following two identities

$$u_j(\bar{x},t,T) \equiv 0, \ \ \text{for} \ \ t \leq (j-1)T \tag{2.16a}$$

and

$$u_j(\bar{x},t,T) \equiv 0, \ \ \text{for} \ \ \bar{x} \in S(t), \ \ t \geq (j-1)T + \frac{d + 2T(c+k)}{c-k} \tag{2.16b}$$

Identities (2.16) imply that for any given t and T the solution $u_j(\bar{x},t,T)$ may differ from zero for $\bar{x} \in S(t)$ only if the following two inequalities hold simultaneously

$$(j-1)T < t, \tag{2.17a}$$

$$t < (j-1)T + \frac{d+2T(c+k)}{c-k}. \tag{2.17b}$$

A fixed prescribed $t = t^0$ can meet both conditions (2.17) if and only if the index j satisfies the inequalities

$$1 + \frac{t^0}{T} - \frac{d+2T(c+k)}{(c-k)T} < j < 1 + \frac{t^0}{T}. \tag{2.18}$$

Therefore, there is only a finite number of values j for which $u_j(\bar{x},t,T)$ differs from zero for $\bar{x} \in S(t)$ and given t. If $k=0$ and $T > d/c$ (and also t^0 is sufficiently large) then there is no more than three such values of j. If $T \longrightarrow +0$ or $k \longrightarrow c$ then the number of indexes j that satisfy (2.18) increases with no bound.

Henceforth, we will be using representation (2.15) for the solution $u(\bar{x},t)$ of problem (1.1), (1.2). We note that the term $u_j(\bar{x},t,T)$ in formula (2.15) is of interest for us only till the moment

$$t = (j-1)T + \frac{d+2T(c+k)}{c-k}, \tag{2.19}$$

as starting from this moment the component $u_j(\bar{x},t,T)$ turns into zero inside the computational domain $S(t)$ because of (2.16b) and therefore no longer contributes into the sum (2.15). The sphere $S(t)$ of diameter d centered at $(x_1^0(t), x_2^0(t), x_3^0(t))$ represents at the time t of (2.19) *the trailing front* of the propagation of $u_j(\bar{x},t,T)$ over the unperturbed zero background. (In fact, in many cases the spherical surface $S(t)$ may be a conservative estimate for the actual location of the trailing front; but at any rate, $S(t)$ is always inside the trailing front.)

Numerical algorithms proposed hereafter are based on the concept that when calculating the solution $u(\bar{x},t)$ of (1.1), (1.2), for every t we actually need to calculate only a few terms $u_j(\bar{x},t,T)$ in the sum (2.15) that differ from zero for $\bar{x} \in S(t)$.

Let us make the following important remark. Specify some $z > 0$ and consider the following problem periodic with the period z in every coordinate direction x_l, $l = 1, 2, 3$.

$$\begin{gathered}\frac{\partial^2 v_j}{\partial t^2} - c^2\left(\frac{\partial^2 v_j}{\partial x_1^2} + \frac{\partial^2 v_j}{\partial x_2^2} + \frac{\partial^2 v_j}{\partial x_3^2}\right) = f_j(\bar{x},t,T,z), \\ v_j(\bar{x},t,T,z) = 0, \quad t \le (j-1)T,\end{gathered} \tag{2.20a}$$

$$\begin{gathered}v_j(x_1+k_1z, x_2+k_2z, x_3+k_3z, t, T, z) = v_j(\bar{x},t,T,z) \\ f_j(x_1+k_1z, x_2+k_2z, x_3+k_3z, t, T, z) = f_j(\bar{x},t,T,z) \\ k_1,\, k_2,\, k_3 = 0, \pm 1, \pm 2, \ldots \\ f_j(\bar{x},t,T,z) \equiv f_j(\bar{x},t,T) \;\text{ if }\; |x_l| < z/2,\; l = 1,2,3.\end{gathered} \tag{2.20b}$$

Theorem 2.1. *The solution $v_j(\bar{x},t,T,z)$ of problem (2.20) coincides with the solution $u_j(\bar{x},t,T)$ of problem (2.14)*

$$u_j(\bar{x},t,T) = v_j(\bar{x},t,T,z) \tag{2.21}$$

on the domain

$$\bar{x} \in S(t), \quad (j-1)T \leq t < (j-1)T + \frac{z-d}{c+k}. \tag{2.22}$$

Proof. Let us first note that as long as $\left|\dfrac{d\bar{x}^0(t)}{dt}\right| \leq k < 1$ (see (1.4)), where $\bar{x}^0 = \bar{x}^0(t)$ may be any prescribed law of motion for the center of the sphere $S(t)$, the right-hand side $f_j(\bar{x},t,T,z)$ of (2.20a), which is periodic in all three coordinate directions x_1, x_2, and x_3, may differ from zero only on the collection of balls

$$\begin{gathered}(x_1 - k_1 z)^2 + (x_2 - k_2 z)^2 + (x_3 - k_3 z)^2 \leq \left[\frac{d}{2} + k\,(t - (j-1)T)\right]^2, \\ t \geq (j-1)T, \quad k_1,\, k_2,\, k_3 = 0, \pm 1, \pm 2, \ldots\end{gathered} \tag{2.23}$$

This actually follows from the fact that the sphere $S(t)$ for $t \geq (j-1)T$ completely belongs to the ball (2.23) for $k_1 = k_2 = k_3 = 0$. Moreover, it is easy to see that the lower portion of the characteristic cone (2.4a):

$$(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2 + (x_3 - \xi_3)^2 = c^2(t-\theta)^2, \quad \theta < t,$$

with the vertex in an arbitrary point $(\bar{x},t)$ that belongs to the ball (2.23) for $k_1 = k_2 = k_3 = 0$ intersects none of the spherical domains (2.23) for $k_1^2 + k_2^2 + k_3^2 \neq 0$ (i.e., none of the other balls (2.23)) on the time interval $(j-1)T \leq \theta \leq t$ if only

$$t < (j-1)T + \frac{z-d}{c+k}. \tag{2.24}$$

Then, the Kirchhoff formula (2.3) implies that the value of the solution $v_j(\bar{x},t,T,z)$ in the vertex $(\bar{x},t)$ of the characteristic cone (2.4a) will not depend on the values of the right-hand side $f_j(\bar{\xi},\theta,T,z)$ of equation (2.20a) on the domains (2.23) for $k_1^2+k_2^2+k_3^2 \neq 0$ ($\bar{\xi}$ is substituted for $\bar{x}$ and θ is substituted for t in formula (2.23)). In particular, the value $v_j(\bar{x},t,T,z)$ will not change if the right-hand side $f_j(\bar{\xi},\theta,T,z)$ on all domains (2.23) except the "central" one $k_1 = k_2 = k_3 = 0$ was replace by the identical zero for all $\theta \leq t$, t of (2.24). On the other hand, this replacement makes the right-hand side of (2.20a) coincide with the non-periodic right-hand side of equation (2.15), which has the solution $u_j(\bar{x},t,T)$. Thus, $v_j(\bar{x},t,T,z) = u_j(\bar{x},t,T)$ for all those $(\bar{x},t)$, for which t satisfies (2.24) and $\bar{x}$ belongs to the ball (2.23) for $k_1 = k_2 = k_3 = 0$. At the same time, it has been mentioned that the sphere $S(t)$ completely belongs to the ball (2.23), $k_1 = k_2 = k_3 = 0$, for any $t \geq (j-1)T$. This proves the theorem. □

3. Finite-Difference Algorithms. We will construct three algorithms for the approximate calculation of the solution to problem (1.1), (1.2) on arbitrarily long time intervals using finite-difference scheme (1.6). All three algorithms will guarantee that the error will not accumulate and computer expenses per time step (both CPU time-wise and storage-wise) will not increase, i.e., will remain fixed and bounded throughout the entire computation. All three algorithms will also have the same non-improvable computational complexity, i.e., asymptotic order of the number of required arithmetic operations and amount of memory with respect to the grid size. However, the algorithms will differ from one another by the actual number of required arithmetic operations and amount of memory (while still having the same asymptotic order with respect to the grid size h), as well as by certain conveniency features from the standpoints of their justification on one hand and practical computing on the other hand.

For all three algorithms we assume that $T > 0$, $h > 0$, $\tau > 0$, and $z > 0$ are chosen so that $T = q\tau$, $z = rh$, where q and r are positive integers, and $\tau/h \leq 1/c\sqrt{3}$, i.e., the von Neumann stability criterion is met.

The First Algorithm. This algorithm is based on the representation (2.15) of the solution $u(\bar{x}, t)$ to problem (1.1), (1.2) in the sphere $S(t)$. Let us fix some arbitrary integer l and consider t from the interval

$$(l-1)T \leq t \leq lT. \tag{3.1}$$

For these t, formula (2.15) can be rewritten as follows

$$u(\bar{x}, t) = u_{l-s}(\bar{x}, t, T) + u_{l-s+1}(\bar{x}, t, T) + \ldots + u_l(\bar{x}, t, T), \tag{3.2}$$

where $u_j(\bar{x}, t, T)$, $j = l-s, l-s+1, \ldots, l$, are solutions of the corresponding problems (2.14). The positive integer number s is chosen from the inequalities (3.1) and (2.18). If, instead of the smallest possible s that satisfies the foregoing constraints, one takes, e.g., $s+1$, then an additional term $u_{l-s-1}(\bar{x}, t, T)$ will simply appear in the sum (3.2). This term, however, will turn into zero for $\bar{x} \in S(t)$ and t satisfying (3.1) and consequently, the work required for computing this term will be superfluous.

Assume, for definiteness, that s is chosen according to the formula:

$$s = \left[\frac{d + 2T(c+k)}{(c-k)T}\right] + 1, \tag{3.3}$$

where $[\,\cdot\,]$ denotes the integer part. We will be calculating the solution $u(\bar{x}, t)$ of problem (1.1), (1.2) on the grid $n = (n_1 h,\, n_2 h,\, n_3 h,\, n_4 \tau)$ for

$$(n_1 h,\, n_2 h,\, n_3 h) \in S(n_4 \tau), \quad (l-1)T \leq n_4 \tau \leq lT,$$

i.e., inside the sphere $S(t)$ for t satisfying (3.1). According to (3.2) and (3.3), *the exact values* of this solution on the grid are given by

$$\begin{gathered} u(n_1 h,\, n_2 h,\, n_3 h,\, n_4 \tau) = u_{l-s}(n_1 h,\, n_2 h,\, n_3 h,\, n_4 \tau,\, T) + \\ + u_{l-s+1}(n_1 h,\, n_2 h,\, n_3 h,\, n_4 \tau,\, T) + \ldots + u_l(n_1 h,\, n_2 h,\, n_3 h,\, n_4 \tau,\, T) \end{gathered} \tag{3.4}$$

Instead of the exact values (3.4) the first algorithm that we are discussing yields *approximate values* of the solution $u(\bar{x},t)$ on the grid according to the formula:

$$u(n_1h,\, n_2h,\, n_3h,\, n_4\tau) \approx u_n^{(I)} \equiv u_n(l-s,\, T) + u_n(l-s+1,\, T) + \ldots + u_n(l,\, T). \tag{3.5}$$

Each term $u_n(j,\, T)$, $j = l-s, l-s+1, \ldots, l$, on the right-hand side of relation (3.5) solves the following finite-difference counterpart to problem (2.14):

$$\begin{gathered}\sum_{n \in N_m} a_{mn} u_n(j,\, T) = f_m(j,\, T), \quad j = l-s, l-s+1, \ldots, l, \\ u_n(j,\, T) = 0, \;\text{ if }\; n_4\tau \le (j-1)T + \tau.\end{gathered} \tag{3.6}$$

The right-hand side $f_m(j,\, T)$ of (3.6) is given by the expression

$$\begin{aligned}f_m(j,\, T) = f_j(\bar{x},t,T)\Big|_{\bar{x}=(m_1h,m_2h,m_3h),\, t=m_4\tau} &= \\ = \Psi_j(t,T) f(\bar{x},t)\Big|_{\bar{x}=(m_1h,m_2h,m_3h),\, t=m_4\tau}&.\end{aligned}$$

(see (2.11), (2.12), (2.13)). Obviously,

$$f_m(j,\, T) = 0, \;\text{ if }\; m_4\tau \le (j-1)T. \tag{3.7}$$

Actually, it turns out that to obtain *a discrete approximation* (see (3.5)) to the solution $u_j(\bar{x},t)$ of problem (2.14) for $\bar{x} \in S(t)$, $(l-1)T \le t \le lT$, one needs to solve the finite-difference problem (3.6) for every j, $j = l-s, l-s+1, \ldots, l$, only on the time interval $(j-1)T \le t \le lT$ (rather than $0 \le t \le lT$). The length of the largest of these intervals does not exceed $d/(c-k) + 4cT/(c-k)$ (see (3.3)) and does not depend on l. Therefore, the following theorem holds.

THEOREM 3.1. *The error due to the replacement of the true solution $u_j(\bar{x},t)$ in the exact formula (3.4) by the difference solution $u_n^{(I)} \equiv \sum u_n(j,\, T)$ of the first algorithm given by the approximate formula (3.5) does not exceed $(s+1)Ch^2$,*

$$\max_{n_1,n_2,n_3,n_4} \left| u(n_1h,\, n_2h,\, n_3h,\, n_4\tau) - u_n^{(I)} \right| \le (s+1)Ch^2, \tag{3.8}$$

where the constant C depends only on the properties of the function $f(\bar{x},t)$ for $(l-s-1)T \le t \le lT$ and $s+1$ is the number of terms in the sum (3.5). Theorem 3.1 is, in fact, an obvious consequence of the construction of the first algorithm. Indeed, the estimate of type (1.12) will hold for each term on the right-hand side of formula (3.5). None of these terms is calculated on the time interval that exceeds $\mathcal{T} = d/(c-k) + 4cT/(c-k)$ in length and therefore, all constants will be bounded. Consequently, the overall error estimate can be written in the form (3.8). Note, unlike in estimate (1.12), the constant C of (3.8) includes the factor of type $|f|_{K,\mathcal{T}}$ that reflects the smoothness properties of the right-hand side. The function $\Psi(t)$ of (2.9) that helps us build the partition of unity (2.10) was chosen sufficiently smooth for the particular reason of guaranteeing that this partition that was introduced primarily for the algorithmic purposes will not "interfere" with the error estimates for the algorithm. Obviously, the error given by (3.8) will not increase, i.e., will remain uniformly bounded, when the number n_4 of the time step increases as long as the smoothness properties of $f(\bar{x}, t)$ remain uniformly "good" with respect to t. Note, for higher-order schemes one may need smoother functions $\Psi(t)$.

Let us now estimate the computer resources required by the proposed algorithm. Clearly, both the operations count and the amount of memory (i.e., number of words) needed for advancing one time step when calculating each term $u_n(j, T)$ of (3.5) are of the same asymptotic order $\mathcal{O}(h^{-3})$ with respect to the grid size h. The number of terms $s+1$ does not change (i.e., does not grow) when the grid is refined as long as T is fixed. Therefore, neither does the overall number of arithmetic operations and words of memory required when calculating the solution by means of formula (3.5) — both quantities remain of the order $\mathcal{O}(h^{-3})$. The number of grid nodes that belong to the sphere $S(t)$ for a fixed $t = n_4\tau$ is also of the order $\mathcal{O}(h^{-3})$; therefore, the foregoing algorithm appears asymptotically non-improvable — *linear with respect to the grid dimension* — as long as one uses the scheme (1.6).

We also note that out of the several terms that need to be computed according to (3.5), the first one, $u_n(l - s, T)$, is the most expensive numerically, its calculation by an explicit scheme up to the time level $t = n_4\tau = lT$ requires the widest grid domain of the size approximately

$$d + 2(s+1)Tc = d + 2\left(\frac{d}{c-k} + \frac{4cT}{c-k}\right)c = \frac{1}{c-k}\left((3c-k)d + 8c^2T\right) \tag{3.9}$$

Dividing (3.9) by h and taking the third power of the result, we will obtain a quantity of the order $\mathcal{O}(h^{-3})$ (as long as T is fixed while the grid is refined). This quantity gives an estimate of what will be the actual amount of memory needed for advancing one time step using the first algorithm.

Finally, let us mention that when discussing the long-time integration we can consider a formulation that slightly differs from (3.1). Considering t from the interval (3.1) means, in fact, that l can be arbitrarily large and we calculate the solution on the interval of the fixed length T, which can be placed arbitrarily far away from the initial data. Alternatively, one may be interested in knowing the overall temporal evolution of the solution, i.e., in calculating the solution on an arbitrarily long time interval, say from 0 to some large T_{final}. From the standpoint of building a lacunae-based algorithm that provides for non-accumulation of error and non-increasing expenses, this formulation is not much different from the one analyzed previously. For every time interval (3.1), $T \ll T_{\text{final}}$, i.e., every l, the solution will still be computed using formula (3.5). To advance further in time, we then need to replace l by $l+1$ in formula (3.1). This will simply imply dropping the first term $u_n(l-s, T)$ on the right-hand side of formula (3.5) and adding the new last term $u_n(l+1, T)$. In so doing, each term $u(l-q, T)$, $q = 0, 1, \dots, s-1$, for the previous interval l becomes $u(l+1-(q+1), T)$ for the new interval $l+1$. Of course, for the new interval there is no need to calculate this term from the very beginning by solving the corresponding problem (3.6) starting from $j = l - q$; the calculation of each term of (3.5) that is not dropped when going from l to $l+1$ (i.e., every term except the first one) is rather continued from the previous interval using the same explicit scheme.

The Second Algorithm. In this algorithm, instead of formula (3.2) we use the following representation of the solution $u(\bar{x}, t)$ for $\bar{x} \in S(t)$, $(l-1)T \leq t \leq lT$:

$$u(\bar{x},t) = v_{l-s}(\bar{x},t,T,z) + v_{l-s+1}(\bar{x},t,T,z) + \ldots + v_l(\bar{x},t,T,z), \tag{3.10}$$

here $v_j(\bar{x},t,T,z)$, $j = l-s, l-s+1, \ldots, l$, are solutions of the corresponding problems (2.20). In so doing, the period z has to be chosen so that for every function $u_j(\bar{x},t,T)$, $j = l-s, l-s+1, \ldots, l$, the equality

$$u_j(\bar{x},t,T) = v_j(\bar{x},t,T,z)$$

hold on the entire time interval

$$(j-1)T < t < (j-1)T + \frac{d + 2T(c+k)}{c-k}, \tag{3.11}$$

on which according to formulae (2.17) the function $u_j(\bar{x},t,T)$ may differ from zero for $\bar{x} \in S(t)$. In other words, we require that the time interval (3.11) be contained inside the time interval (2.22) (see Theorem 2.1) or

$$(j-1)T + \frac{d + 2T(c+k)}{c-k} \leq (j-1)T + \frac{z-d}{c+k},$$

which yields the following condition:

$$z \geq \frac{1}{c-k}\left(2cd + 2(c+k)^2 T\right). \tag{3.12}$$

Condition (3.12) essentially guarantees that all the waves generated by the sources $\bar{x} \in S(t)$ that operate on the time interval $2T$ will leave the domain of interest $S(t)$ before the waves generated by the other sources from the periodic structure can enter this domain.

To actually build the second algorithm, we replace the differential equation and initial condition of (2.20a) by the finite-difference equation and discrete initial condition, respectively:

$$\begin{gathered}\sum_{n \in N_m} a_{mn} v_n(j,\, T,\, z) = f_m(j,\, T,\, z), \quad j = l-s, l-s+1, \ldots, l \\ v_n(j,\, T,\, z) = 0, \text{ if } \; n_4\tau \leq (j-1)T + \tau,\end{gathered} \tag{3.13a}$$

where the right-hand side $f_m(j,\, T,\, z)$ is a z-periodic grid function with node values

$$f_m(j,\, T,\, z) = f_j(\bar{x},t,T,z)\Big|_{\bar{x}=(m_1h, m_2h, m_3h),\; t=m_4\tau}.$$

The periodic boundary conditions (2.20b) are replaced by their discrete counterparts in every coordinate direction x_1, x_2, and x_3 (the period is $z = rh$):

$$
\begin{gathered}
v_n(j,\,T,\,z) = v_{n'}(j,\,T,\,z)\\
n = (n_1, n_2, n_3, n_4), \quad n' = (n_1 + k_1 r, n_2 + k_2 r, n_3 + k_3 r, n_4)\\
k_1,\, k_2,\, k_3 = 0, \pm 1, \pm 2, \dots
\end{gathered}
\tag{3.13b}
$$

The approximation to the solution $u(\bar{x}, t)$ for $\bar{x} \in S(t)$, $(l-1)T \leq t \leq lT$, in the second algorithm is obtained as an approximation to the sum (3.10) (compare to (3.5) and (3.2), respectively):

$$
\begin{gathered}
u(n_1 h,\, n_2 h,\, n_3 h,\, n_4 \tau) \approx u_n^{(II)} \equiv\\
v_n(l-s,\,T,\,z) + v_n(l-s+1,\,T,\,z) + \ldots + v_n(l,\,T,\,z).
\end{gathered}
\tag{3.14}
$$

Each term $v_n(j,\,T,\,z)$, $j = l-s, l-s+1, \dots, l$, on the right-hand side of relation (3.14) solves the corresponding problem (3.13). Clearly, the error estimate of type (3.8) (with $u_n^{(I)}$ of (3.5) replaced by $u_n^{(II)}$ of (3.14)) provided by Theorem 3.1 for the first algorithm, will hold for the second algorithm as well.

As we did previously for the first algorithm, let us now consider the transition from l to $l+1$ in formula (3.1) in the framework of the second algorithm. Assume we are interested in calculating the overall temporal evolution of the solution from $t = 0$ till an arbitrarily large $t = T_{\text{final}}$. Over this period of time, the domain $S(t)$ that was centered at $\bar{x}^0(0) = (x_1^0(0), x_2^0(0), x_3^0(0))$ at the moment $t = 0$ can travel arbitrarily far in space from its initial location, in fact, as far as kT_{final}:

$$
\bar{x}^0(t) = (x_1^0(t), x_2^0(t), x_3^0(t)) = \bar{x}^0(0) + \int_0^t \frac{d\bar{x}^0}{dt} dt, \tag{3.15a}
$$

$$
\left|\bar{x}^0(t)\right| \leq kT_{\text{final}}, \quad 0 \leq t \leq T_{\text{final}}. \tag{3.15b}
$$

In the z-periodic setting, all functions are defined for $|x_i| \leq z/2$, $i = 1,2,3$, and the edges $x_i = \pm z/2$, $i = 1,2,3$, are identified with one another. Accordingly, instead of the motion described by relation (3.15a), we consider the motion of $S(t)$ on a three-dimensional toroidal surface. Instead of (3.15a) we will then have

$$
\bar{x}^0(t) = (x_1^0(t), x_2^0(t), x_3^0(t)) = \left\{ \frac{1}{z} \left(\bar{x}^0(0) + \int_0^t \frac{d\bar{x}^0}{dt} dt \right) \right\} z - \frac{z}{2}, \tag{3.16a}
$$

where $\{\,\cdot\,\}$ denotes the fractional part. Also, conforming to the periodicity conditions, inequality (3.15b) transforms into

$$
\left|\bar{x}^0(t)\right| \leq z/2, \quad 0 \leq t \leq T_{\text{final}}. \tag{3.16b}
$$

The computational procedure does not change much. We calculate separately every term on the right-hand side of (3.14). When we go from l to $l+1$, we stop calculating $v_n(l-s, T, z)$ and add the new term $v_n(l+1, T, z)$. This allows us to run the computation arbitrarily long with no error accumulation and no growth of computer expenses per time step. In so doing, of course, the center $\bar{x}^0(t)$ of $S(t)$, as well as the entire domain of interest $S(t)$ itself, can be located anywhere within the period, i.e., in the cube $\{|x_i| \leq z/2,\ i = 1, 2, 3\}$, and not necessarily close to its middle. This, however, does not affect the solution calculated inside $S(t)$ because according to condition (3.12), no waves from outside can enter the domain $S(t)$ before the waves generated inside $S(t)$ by the sources that operate during the interval $2T$ leave it. Then, as soon as these waves have left, this entire portion of the solution, both the waves generated inside $S(t)$ that have already left it and the waves generated outside $S(t)$ by the other sources of the periodic structure that operate during the same time interval it taken out by eliminating the term $v_n(l-s, T, z)$ in the sum (3.14). As has been pointed out, this does not essentially change anything inside $S(t)$ as the waves have already left, but it prevents the waves generated outside from propagating further in.

The second algorithm appears somewhat more efficient than the first one as the actual domain size and consequently, the number of grid nodes involved, are smaller for the second algorithm, compare (3.12) against (3.9). The number of terms $s+1$ in formulae (3.5) and (3.10) does not depend on the grid size h if T does not depend on h. If, however, T decreases as h decreases (when the grid is refined) then the number s and also constants involved in the estimates $|\mathcal{O}(h^{-3})| \leq Kh^{-3}$ grow. If $T \sim \tau$, then the number s becomes $s = \mathcal{O}(h^{-1})$ and the operations count, accordingly, $\mathcal{O}(h^{-4})$ instead of the non-improvable quantity $\mathcal{O}(h^{-3})$.

The third algorithm keeps the non-improvable asymptotic order $\mathcal{O}(h^{-3})$ for both the number of arithmetic operations and amount of memory required for advancing one time step even if T decreases when the grid is refined as long as $T \geq \tau|\ln\tau|$.

The Third Algorithm. The third algorithm uses exactly the same approximation on each time step as the second algorithm does (see (3.10)). However, the computations in the third algorithm are organized in a substantially different way. In this algorithm, instead of calculating separately each term on the right-hand side of (3.14) we rather use a "one-sweep" time-marching approach and when it comes to the transition from l to $l+1$ in (3.1), the term $v_n(l-s, T, z)$ on the right-hand side of (3.14) is taken out by the explicit subtraction.

Introduce a new grid function

$$V_n(l, s, T, z) = \sum_{j=l-s}^{\infty} v_n(j, T, z), \tag{3.17}$$

where $v_n(j, T, z)$ is the solution of problem (3.13). Here s is the same integer number as in formula (3.10). Notice that for any $n = (n_1h, n_2h, n_3h, n_4\tau)$ only several terms of the series (3.17) may differ from zero. For those nodes n that belong to the grid time levels

$$t = n_4\tau, \quad n_4 = \frac{(l-1)T}{\tau},\ \frac{(l-1)T}{\tau}+1,\ \ldots,\ \frac{lT}{\tau}-1 \tag{3.18}$$

the sum (3.17) coincides with the approximation of the solution $u(\bar{x}, t)$ given by formula (3.14) for $(n_1h, n_2h, n_3h) \in S(t)$, $t = n_4\tau$. Thus, the computation of $V_n(l, s, T, z)$ by formula (3.17) can be interpreted as the computation of the approximate solution by formula (3.14).

Substituting expression (3.17) into the left-hand side of the finite-difference equation (1.6), we obtain

$$\sum_{n\in N_m} a_{mn}V_n(l,s,T,z) = f_m(l-s,T,z)+ \\ +f_m(l-s+1,T,z)+\ldots+f_m(l,T,z)+\ldots \tag{3.19}$$

According to the definition of $\Psi(t)$, see (2.9), and formula (2.11), the following equality holds

$$f_m \equiv f(m_1h, m_2h, m_3h, m_4\tau) = \sum_{j=l-s}^{\infty} f_m(j,T,z) \tag{3.20}$$

for all those m, for which

$$m_4\tau \geq (l-s)T. \tag{3.21}$$

Equalities (3.19) and (3.20) imply that the function $V_n(l,s,T,z)$ satisfies the finite-difference equation

$$\sum_{n\in N_m} a_{mn}V_n(l,s,T,z) = f_m \tag{3.22}$$

for those m, for which the inequality (3.21) holds. Clearly, for $l=0$ and $n_4=0$, $n_4=1$, we have $V_n(l,s,T,z)=0$.

Assume now that for some integer $l\geq 0$ we already know the values of $V_n(l,s,T,z)$ on the first two time levels of the grid (3.18), i.e., for $n_4=(l-1)T/\tau$ and $n_4=(l-1)T/\tau+1$. We will describe the computation of $V_n(l,s,T,z)$ on all other levels of the grid (3.18), as well as computation of $V_n(l+1,s,T,z)$ for $n_4=lT/\tau$ and $n_4=lT/\tau+1$. Thus, we will have completed the cycle of going from the use of $V_n(l,s,T,z)$ for calculating the approximate solution on the grid levels (3.18) to the use of $V_n(l+1,s,T,z)$ for calculating the approximate solution on the subsequent levels

$$t=n_4\tau, \quad n_4=\frac{lT}{\tau},\ \frac{lT}{\tau}+1,\ \ldots,\frac{(l+1)T}{\tau}-1.$$

So, having specified $V_n(l,s,T,z)$ for

$$t=n_4\tau, \quad n_4=\frac{(l-1)T}{\tau} \ \text{ and } \ \frac{(l-1)T}{\tau}+1,$$

we calculate the values of $V_n(l,s,T,z)$ on all other levels of the grid (3.18) using the explicit finite-difference scheme (3.22). Then, the values of $V_n(l+1,s,T,z)$ for $n_4=lT/\tau$ and $n_4=lT/\tau+1$ can be obtained with the help of the following obvious recurrence formula:

$$V_n(l+1,s,T,z) = V_n(l,s,T,z) - v_n(l-s,T,z). \tag{3.23}$$

The first term on the right-hand side of formula (3.23) is calculated with the explicit scheme (3.22) using the data $V_n(l,s,T,z)$ that have already been obtained on the last two levels $n_4 = lT/\tau - 2$ and $n_4 = lT/\tau - 1$ of the grid (3.18). The second term $v_n(l-s,T,z)$ on the right-hand side of formula (3.23) is the solution of problem (3.13) for $j = l - s$. The right-hand side $f_m(j,T,z)$ of the latter problem may differ from zero only for those $m = (m_1h, m_2h, m_3h, m_4\tau)$ that satisfy

$$(l-s-1)T < m_4\tau < (l-s+1)T,$$

and the initial data for calculating $v_n(l-s,T,z)$ are homogeneous.

The actual computation of $v_n(l-s,T,z)$ will be split into two stages. First, we will calculate the solution $v_n(l-s,T,z)$ with the explicit finite-difference scheme step after step in time for the levels

$$t = n_4\tau, \quad n_4 = \frac{(l-s-1)T}{\tau}, \dots, \frac{(l-s+1)T}{\tau}. \tag{3.24}$$

The computation of $v_n(l-s,T,z)$ on the levels (3.24) takes

$$\mu \frac{2T}{\tau} \left(\frac{z}{h}\right)^3 \tag{3.25}$$

arithmetic operations, where μ is the number of operations required for calculating the solution in one grid node on the next time level while the solution on the two previous levels is already known. (In other words, μ is the number of arithmetic operations "on the stencil" of the scheme.)

The values of the solution $v_n(l-s,T,z)$ on the last two levels of (3.24), i.e., $n_4 = (l-s+1)T/\tau - 1$ and $n_4 = (l-s+1)T/\tau$, will be used as the initial data for calculating this solution $v_n(l-s,T,z)$ for $n_4 = lT/\tau$ and $n_4 = lT/\tau + 1$. As the right-hand side $f_m(l-s,T,z)$ is equal to zero, $f_m(l-s,T,z) = 0$, for $m_4 \geq (l-s+1)T/\tau$, an efficient way to calculate the solution $v_n(l-s,T,z)$ for $n_4 = lT/\tau$ and $n_4 = lT/\tau + 1$ will be through representing it in the form of a discrete Fourier series while the initial data for $v_n(l-s,T,z)$ on the levels $n_4 = (l-s+1)T/\tau - 1$ and $n_4 = (l-s+1)T/\tau$ are known. The aforementioned finite Fourier expansion is built with respect to the following system of grid basis functions e_n^j:

$$\begin{gathered} e_n^j = \exp(i\langle n,j\rangle), \quad i = \sqrt{-1}, \\ n = (n_1,n_2,n_3), \quad j = (j_1,j_2,j_3), \\ j_1,\, j_2,\, j_3 = 0,1,2,\dots,\frac{z}{h} - 1, \\ \langle n,j\rangle = n_1j_1 + n_2j_2 + n_3j_3. \end{gathered}$$

Hereafter we assume that z/h is a power of 2 so that the fast Fourier transform (FFT) can be used for for calculating the coefficients of the discrete Fourier series of a given grid function, as well as for restoring the point-wise values of the grid function from its Fourier representation. Each of the foregoing transformations requires

$$\mathcal{O}\left(\left(\frac{z}{h}\right)^3 \ln\frac{z}{h}\right) \tag{3.26}$$

arithmetic operations.

Thus, the second stage of the computation of $v_n(l-s,T,z)$ for $n_4 = lT/\tau$ and $n_4 = lT/\tau + 1$ will first consist of Fourier transforming the data on the last two levels of grid (3.24), which takes (3.26) operations (twice the actual amount of (3.26) for two levels). Then, we advance each Fourier component independently to the levels $n_4 = lT/\tau$ and $n_4 = lT/\tau + 1$ using the explicit formulae that are easily obtained from the Fourier representation of the finite-difference operator of (3.22); essentially, this reduces to multiplication by the appropriate powers of the corresponding amplification factors and obviously takes $\mathcal{O}\left(\left(\frac{z}{h}\right)^3\right)$ operations. Finally, we need to restore the point-wise values of $v_n(l-s,T,z)$ at $n_4 = lT/\tau$ and $n_4 = lT/\tau + 1$ using the inverse FFT, which again takes (3.26) operations. The overall computational cost of this second stage will then be

$$\mathcal{O}\left(\left(\frac{z}{h}\right)^3 \ln\frac{z}{h}\right) + \mathcal{O}\left(\left(\frac{z}{h}\right)^3\right) + \mathcal{O}\left(\left(\frac{z}{h}\right)^3 \ln\frac{z}{h}\right) = \mathcal{O}\left(\left(\frac{z}{h}\right)^3 \ln\frac{z}{h}\right) \tag{3.27}$$

arithmetic operations. Consequently, the total operations count for calculating $v_n(l-s,T,z)$ for $n_4 = lT/\tau$ and $n_4 = lT/\tau + 1$, i.e., calculating the second term in the recurrence formula (3.23), consists of the expenses for the first (3.25) and second (3.27) stages and adds up to

$$\mu\frac{2T}{\tau}\left(\frac{z}{h}\right)^3 + \mathcal{O}\left(\left(\frac{z}{h}\right)^3 \ln\frac{z}{h}\right) \tag{3.28}$$

operations. Recurrence formula (3.23) is used for the entire "chunk" of $2T/\tau$ time levels, $(l-s-1)T \le t \le (l-s+1)T$. Therefore, if one recalculates the associated expense (3.28) proportionally per time step, it obviously reduces to

$$\mu\left(\frac{z}{h}\right)^3 + \mathcal{O}\left(\frac{\tau}{2T}\left(\frac{z}{h}\right)^3 \ln\frac{z}{h}\right) = \mathcal{O}\left(\left(\frac{z}{h}\right)^3\right).$$

The calculation of the first term in the recurrence formula (3.23) also requires $\mathcal{O}\left(\left(\frac{z}{h}\right)^3\right)$ arithmetic operations per time step as this is done simply using the explicit scheme (3.22).

Summarizing, we conclude that the overall computational cost of the third algorithm is $\mathcal{O}\left(\left(\frac{z}{h}\right)^3\right)$ arithmetic operations per one time step. It is also easy to see that the required amount of memory (i.e., number of words) is of the order $\mathcal{O}\left(\left(\frac{z}{h}\right)^3\right)$ as well.

As the third algorithm essentially reproduces the calculation according to formula (3.14) with the difference only in the computational procedure, the uniform error estimate of type (3.8) provided by Theorem 3.1 for the first algorithm, will hold for the third algorithm as it does for the second algorithm. Moreover, the interpretation of spatial periodicity given by formulae (3.16) and subsequent comments for the second algorithm, applies to the third algorithm with no changes.

4. Possible Generalizations. First, let us note that the assumption of homogeneity of the initial data (1.2) can be alleviated and replaced by a weaker requirement

$$u(\bar{x},t)\Big|_{t=0} = \varphi_0(\bar{x}), \qquad \frac{\partial u(\bar{x},t)}{\partial t}\Big|_{t=0} = \varphi_1(\bar{x}),$$

where $\varphi_0(\bar{x})$ and $\varphi_1(\bar{x})$ are sufficiently smooth functions that turn into zero outside the domain $S(t)\big|_{t=0} = S(0)$.

Further, the requirement of smoothness for $f(\bar{x},t)$ throughout the entire space-time $(\bar{x},t)$ along with the consideration of $f(\bar{x},t)$ only for $t > 0$ actually implies that $f(\bar{x},t)$ and its first several derivatives with respect to t have to be smooth as $t \longrightarrow +0$. This condition can also be alleviated by requiring that the function $f(\bar{x},t)$, $\text{supp} f(\bar{x},t) \subset S(t)$, be smooth for $t \geq 0$ rather than on the entire space-time $(\bar{x},t)$. The resulting Cauchy problem, which appears somewhat more complex, can actually be reduced to problem (1.1), (1.2) if one represents the solution to the new problem as a sum of two functions:

$$u(\bar{x},t) = v(\bar{x},t) + w(\bar{x},t).$$

The function $v(\bar{x},t)$ will be a solution to the Cauchy problem with the given non-homogeneous initial data and the right-hand side $F(\bar{x},t) = \Psi(t)f(\bar{x},t)$ that turns into zero for $t \geq 1$. The function $w(\bar{x},t)$ will be a solution to the problem

$$\begin{gathered}\frac{\partial^2 w}{\partial t^2} - c^2\left(\frac{\partial^2 w}{\partial x_1^2} + \frac{\partial^2 w}{\partial x_2^2} + \frac{\partial^2 w}{\partial x_3^2}\right) = f(\bar{x},t) - F(\bar{x},t), \quad t \geq 0, \\ w\Big|_{t=0} = \frac{\partial w}{\partial t}\Big|_{t=0} = 0.\end{gathered} \tag{4.1}$$

Problem (4.1) is obviously of the type (1.1), (1.2). The problem for $v(\bar{x},t)$ needs to be solved only till some $t = t_0$, after which $v(\bar{x},t) \equiv 0$ when $\bar{x} \in S(t)$ because of the presence of lacunae in the solutions of the three-dimensional wave equation.

Another possible generalization includes building similar lacunae-based algorithms for the long-time numerical integration of problems in acoustics (linearized Euler's equations), electromagnetics (Maxwell's equations), and elastodynamics (Lame's equations). These algorithms may then be used for constructing highly accurate global genuinely time-dependent ABCs in the corresponding fields of scientific computing.

REFERENCES

[1] V. S. Ryaben'kii, V. I. Turchaninov, and S. V. Tsynkov, *The Use of Lacunae of the Three-Dimensional Wave Equation for Calculating the Solution on Long Time Intervals*, submitted to Mathematical Modeling. [in Russian]

[2] S. V. Tsynkov, *Numerical Solution of Problems on Unbounded Domains. A Review*, Appl. Numer. Math., 27 (1998) pp. 465–532.