SIAM J. SCI. COMPUT.
Vol. 21, No. 1, pp. 166–206

EXTERNAL BOUNDARY CONDITIONS FOR THREE-DIMENSIONAL PROBLEMS OF COMPUTATIONAL AERODYNAMICS*

SEMYON V. TSYNKOV†

Abstract. We consider an unbounded steady-state flow of viscous fluid over a three-dimensional finite body or configuration of bodies. For the purpose of solving this flow numerically, we discretize the governing equations (Navier–Stokes) on a finite-difference grid. Prior to the discretization, we obviously need to truncate the original unbounded domain by introducing an artificial computational boundary at a finite distance from the body; otherwise, the number of discrete variables will not be finite. This artificial boundary is typically the external boundary of the domain covered by the grid.

The flow problem (both continuous and discretized) formulated on the finite computational domain is clearly subdefinite unless supplemented by some artificial boundary conditions (ABCs) at the external computational boundary. In this paper, we present an innovative approach to constructing highly accurate ABCs for three-dimensional flow computations. The approach extends our previous technique developed for the two-dimensional case; it employs the finite-difference counterparts to Calderón's pseudodifferential boundary projections calculated in the framework of the difference potentials method (DPM) of Ryaben'kii. The resulting ABCs appear spatially nonlocal but are particularly easy to implement along with the existing flow solvers.

The new boundary conditions have been successfully combined with the NASA-developed production code `TLNS3D` and used for the analysis of wing-shaped configurations in subsonic and transonic flow regimes. As demonstrated by the computational experiments and comparison with the standard local methods, the DPM-based ABCs allow one to greatly reduce the size of the computational domain while still maintaining high accuracy of the numerical solution. Moreover, they may provide for a noticeable speedup of multigrid convergence.

Key words. external flows, infinite-domain problems, artificial boundary conditions, far-field linearization, boundary projections, generalized potentials, difference potentials method, auxiliary problem, separation of variables

AMS subject classifications. 65N99, 76M25

PII. S1064827597318757

1. Introduction.

1.1. Preliminaries.

A standard approach to solving computational infinite-domain boundary-value problems involves truncation as a first step, prior to the discretization of the continuous problem and solution of the resulting discrete system. The truncated problem in both continuous and discrete formulations is clearly subdefinite unless supplemented by the proper closing procedure at the outer computational boundary. The latter boundary is often called *artificial*, emphasizing the fact that it originates from the numerical limitations (the discrete system should contain no more than a finite number of variables) rather than from the original formulation. Typically, the artificial boundary is introduced as an external boundary of the finite computational domain (i.e., the domain covered by the grid, on which the original system is discretized). The corresponding closing procedure at the outer boundary is called the *artificial boundary conditions (ABCs)*.

*Received by the editors March 24, 1997; accepted for publication (in revised form) July 30, 1998; published electronically August 26, 1999. This paper was prepared while the author held a National Research Council Resident Research Associateship at the Aerodynamic and Acoustic Methods Branch, NASA Langley Research Center, Hampton, VA.

http://www.siam.org/journals/sisc/21-1/31875.html

†Department of Applied Mathematics, School of Mathematical Sciences, Tel Aviv University, Ramat Aviv, Tel Aviv 69978, Israel (tsynkov@math.tau.ac.il, http://www.math.tau.ac.il/∼tsynkov/).

In the ideal case, the ABCs would be specified so that the solution on the truncated domain coincides with the corresponding fragment of the original infinite-domain solution. However, in spite of the fact that different ABC methodologies have been studied extensively over the past two decades, the construction of such ideal (i.e., exact) ABCs that would at the same time be computationally inexpensive, easy to implement, and geometrically universal still remains a fairly remote possibility. The primary reason for this is that the exact ABCs are typically nonlocal, for steady-state problems in space and for time-dependent problems in both space and time. The exceptions are rare and, as a rule, restricted to one-dimensional model formulations. From the viewpoint of computing, nonlocality may imply cumbersomeness and high cost. Moreover, as the standard apparatus for deriving the exact ABCs involves integral transforms (along the boundary) and pseudodifferential operators, such boundary conditions can be obtained easily only for the boundaries of regular shape.

On the other hand, highly accurate ABCs are demanded in many areas of scientific computing because, as shown by various authors both theoretically and computationally, the overall accuracy and performance of numerical algorithms, as well as interpretation of the numerical results, strongly depend on the proper treatment of external boundaries. This applies not only to external aerodynamics, which is the subject of this paper, but to many other areas of scientific computing as well.

As mentioned above, the other usual requirements of ABCs, besides minimization of the error associated with domain truncation, are low computational cost, geometric universality (i.e., applicability to a variety of irregular boundaries often encountered in real-life settings), and implementation without difficulties, in particular, readiness in combining the ABCs with existing (interior) solvers. The requirements of this group are typically met by many approximate local methods that are considered alternatives to the exact ABCs as the latter are not attainable routinely. However, the basic trend in terms of accuracy remains the following: higher accuracy for the boundary procedure requires more of the nonlocal nature of exact ABCs to be somehow taken into account.

In fact, almost any numerical algorithm for setting the ABCs can be thought of as a compromise between the foregoing two groups of requirements that in a certain sense contradict one another. Shifting the balance toward locality and practical efficacy often implies insufficient accuracy; shifting it to the other side, toward highly accurate nonlocal techniques, may often yield cumbersome and all but impractical algorithms. It is not surprising, therefore, that the treatment of external boundaries in modern production computations typically follows the first, local, path. In *computational fluid dynamics (CFD)*, for example, only a few ABC methodologies out of the wide variety proposed to date can be regarded as commonly used tools. All of them are either based on essential model simplifications, e.g., local quasi-one-dimensional treatment in the vicinity of the artificial boundary, or are obtained as a localization of some nonlocal ABCs. To meet the overall accuracy requirements when using such simple boundary procedures, one often has to choose excessively large computational domains.

A survey of methods for setting the ABCs in different areas of scientific computing can be found in our recent paper [1], as well as in the comprehensive reviews by Givoli [2, 3]. These surveys give a comparative assessment of the advantages and disadvantages of various global and local techniques and also explain the connections between global and local methods.

1.2. Methods and objectives. This paper continues our work on constructing the ABCs that would combine the advantages relevant to both local and nonlocal

approaches. The specific area of application that we are looking at is *steady-state external viscous flows.*

Previously, we have developed and implemented in practice highly accurate ABCs for the two-dimensional case (plane geometry). Our approach is based on use of the *Calderón generalized potentials and pseudodifferential boundary projections* [4] (see also work by Seeley [5]). The potentials and projections are actually employed in the modified form proposed by Ryaben'kii; the numerical technique used for calculating the potentials and projections is known as the *difference potentials method (DPM)*; see work by Ryaben'kii [6, 7, 8] and also a description of the method in the book by Mikhlin, Morozov, and Paukshto [9]. The resulting DPM-based boundary conditions appear global. As will be seen, however, one of the principal advantages that we gain using the DPM is that the method allows us to simultaneously meet the high standards of accuracy and the requirements of geometric universality and ease of implementation.

The two-dimensional DPM-based ABCs have been used along with the multigrid Navier–Stokes code `FLOMG` by Swanson and Turkel [10, 11, 12]. In spite of their nonlocal nature, the new boundary conditions readily apply to the boundaries of irregular shape and appear very easy to incorporate into the existing solver. In our computations, the DPM-based ABCs have clearly outperformed the standard local methods from the standpoints of *accuracy, convergence rate, and robustness.* The investigated regimes range from the very low (incompressible limit) to transonic Mach numbers and encompass both laminar and turbulent flows.

The aforementioned two-dimensional constructions and corresponding numerical results have been reported in a series of papers. In [13], we describe the foundations of the DPM-based approach to setting the ABCs for computation of two-dimensional external viscous flows (Navier–Stokes equations). In [14], we implement this approach along with the code `FLOMG` and present some numerical results for subsonic and transonic laminar flows over single-element airfoils. In [15], we show the results of subsequent numerical experiments and propose an approach to the approximate treatment of turbulence in the far field. Our work [16] delineates the algorithm for solving one-dimensional systems of ordinary difference equations that arise when calculating the generalized difference potentials. In [17], we extend the area of application for the DPM-based ABCs by analyzing two-dimensional flows that oscillate in time; we also provide some solvability results for the linearized thin-layer equations used for constructing the ABCs. In [18], we present a general survey of the DPM-based methodology applied to solving external problems in CFD, including parallel implementation of the algorithm, combined implementation of nonlocal ABCs with multigrid, and entry-wise interpolation of the matrices of boundary operators with respect to the Mach number and angle of attack. Additionally, in [18] one can find some new theoretical results on the computation of generalized potentials, the construction of ABCs based on the direct implementation of boundary projections (thin-layer equations), and some numerical results for various airfoil flows: laminar and turbulent, transonic and subsonic, including low speeds.

The next natural objective after constructing the two-dimensional algorithm is the analysis of three-dimensional steady-state flows. This case is undoubtedly the one most demanded by the current practice in CFD. In [19, 20], we outline the basic elements of the DPM-based ABCs for steady-state viscous flows around wing-shaped configurations and show some preliminary numerical results for the subsonic regime. The numerical results of [20] have been obtained with the NASA-developed production

code TLNS3D of Vatsa et al. [21, 22]. In [23] we further develop the three-dimensional DPM-based algorithm and present some computational results for transonic flows. In all cases (see [20, 23]), the DPM-based ABCs allow one to greatly reduce the size of the computational domain (compared to the standard local boundary conditions) while still maintaining high accuracy of the numerical solution. This actually means the overall increase of accuracy due to the improved treatment of the artificial boundary; it also implies a substantial economy of the computer resources. Moreover, the DPM-based ABCs may provide for a noticeable speedup (up to a factor of 3) of the convergence of multigrid iterations.

In this paper we, for the first time, systematically describe the three-dimensional DPM-based ABCs for calculating viscous flows around the wings. We address theoretical foundations of the approach, describe numerical algorithm in fairly extensive detail, and demonstrate computational results for different flow regimes, including low speed flow and flow with the shock-induced separation. Numerical results for the DPM-based ABCs are compared with those obtained by the standard local method.

The material in the paper is prepared as follows. In section 2, we formulate the problem, describe the specific geometric setting for three space dimensions, provide foundations for the DPM-based ABCs on the continuous level, and then implement the new algorithm in the finite-difference framework. In section 3, we first briefly summarize the results of our previous numerical experiments in two space dimensions and then report on the recent three-dimensional computations for various flow regimes. Section 4 contains our conclusions.

2. External flow.

2.1. Formulation of the problem. We consider an unbounded steady-state flow of viscous fluid past a three-dimensional wing. The flow is uniform at infinity. We consider both incompressible and compressible formulations; in the latter case we assume that the fluid (gas) is thermodynamically perfect and that the free stream is subsonic. Moreover, as the fluid is viscous and the size of the immersed body (wing) is finite, the flow limit at infinity is free stream.

Generally, the near-field flow is governed by the full Navier–Stokes equations. However, in many cases (including those studied in this paper; see section 2.2) the full system can be simplified and reduced to the so-called thin-layer equations [24], which do not contain streamwise viscous derivatives. In particular, this simplification is done in the code TLNS3D that we use for our numerical tests (section 3). Moreover, for the most interesting case of turbulent flows the near-field numerical algorithm should also incorporate some turbulence model; we comment on this issue in section 3, which is devoted to numerics.

2.1.1. Linearization. In the far field (i.e., far enough from the finite immersed body) the perturbations of the flow induced by the immersed body are small and we therefore linearize the governing thin-layer equations against a constant free-stream background. Introducing the Cartesian coordinates (x, y, z) and assuming (without loss of generality) that the free stream is aligned with the positive x direction, we can write the dimensionless linearized equations as

$$\boldsymbol{L}\,\boldsymbol{u} \equiv \boldsymbol{C}\frac{\partial \boldsymbol{u}}{\partial x} + \boldsymbol{D}\frac{\partial \boldsymbol{u}}{\partial y} + \boldsymbol{E}\frac{\partial \boldsymbol{u}}{\partial z} + \boldsymbol{F}\frac{\partial^2 \boldsymbol{u}}{\partial y^2} + \boldsymbol{H}\frac{\partial^2 \boldsymbol{u}}{\partial z^2} + \boldsymbol{J}\frac{\partial^2 \boldsymbol{u}}{\partial y \partial z} = \mathbf{o}, \tag{2.1a}$$

where for the incompressible case

$$\boldsymbol{u}=\begin{bmatrix} p\\ u\\ v\\ w\end{bmatrix},\ \boldsymbol{C}=\begin{bmatrix} 0&1&0&0\\ 1&1&0&0\\ 0&0&1&0\\ 0&0&0&1\end{bmatrix},\ \boldsymbol{D}=\begin{bmatrix} 0&0&1&0\\ 0&0&0&0\\ 1&0&0&0\\ 0&0&0&0\end{bmatrix},\ \boldsymbol{E}=\begin{bmatrix} 0&0&0&1\\ 0&0&0&0\\ 0&0&0&0\\ 1&0&0&0\end{bmatrix},$$

(2.1b)

$$\boldsymbol{F}=\boldsymbol{H}=-\frac{1}{\mathrm{Re}}\begin{bmatrix} 0&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&0&1\end{bmatrix},\quad \boldsymbol{J}=\boldsymbol{0},$$

and for the compressible case

$$\boldsymbol{u}=\begin{bmatrix} \rho\\ u\\ v\\ w\\ p\end{bmatrix},\ \boldsymbol{C}=\begin{bmatrix} 1&1&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&{M_0}^{-2}&0&0&1\end{bmatrix},\ \boldsymbol{D}=\begin{bmatrix} 0&0&1&0&0\\ 0&0&0&0&0\\ 0&0&0&0&1\\ 0&0&0&0&0\\ 0&0&{M_0}^{-2}&0&0\end{bmatrix},$$

(2.1c)

$$\boldsymbol{E}=\begin{bmatrix} 0&0&0&1&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&1\\ 0&0&0&{M_0}^{-2}&0\end{bmatrix},\quad \boldsymbol{F}=-\frac{1}{\mathrm{Re}}\begin{bmatrix} 0&0&0&0&0\\ 0&1&0&0&0\\ 0&0&4/3&0&0\\ 0&0&0&1&1\\ Pr^{-1}{M_0}^{-2}&0&0&0&\gamma\,Pr^{-1}\end{bmatrix},$$

$$\boldsymbol{H}=-\frac{1}{\mathrm{Re}}\begin{bmatrix} 0&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&4/3&1\\ Pr^{-1}{M_0}^{-2}&0&0&0&\gamma\,Pr^{-1}\end{bmatrix},$$

$$\boldsymbol{J}=-\frac{1}{\mathrm{Re}}\begin{bmatrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&1/3&0\\ 0&0&1/3&0&0\\ 0&0&0&0&0\end{bmatrix}.$$

Here, u, v, w, and p are the perturbations of the Cartesian velocity components and pressure, respectively, and ρ is the perturbation of density for the compressible case.

The derivation of (2.1) involves two consecutive steps. First, we introduce the dimensionless quantities for the original thin-layer system. To do that, in both incompressible and compressible cases we scale the velocity projections by the dimensional free-stream value of the x-component u_0. (As mentioned above, the free-stream y and z velocity components are zero: v_0=0, w_0=0.) Moreover, for the incompressible flow we scale the original pressure by ${u_0}^2$ and for the compressible flow the pressure is scaled by $\rho_0{u_0}^2$; the internal energy ϵ is scaled by ${u_0}^2$ and the viscosity μ is scaled by μ_0. (Everywhere above, the subscript "0" denotes the corresponding "full," i.e.,

"nonlinear," dimensional value.) Finally, we scale the coordinates x, y, and z by the characteristic length L; for example, it may be the root chord or semispan of the wing (see section 2.2). For the compressible flow, we also have to use the equation of state (perfect gas) to eliminate internal energy from the original system.

After the nondimensionalization, we represent each quantity (velocities and pressure for the incompressible case and velocities, pressure, and density for the compressible case) as a sum of the constant background value (free stream) and small perturbation and then modify the equations by retaining only the first-order terms with respect to the perturbations and dropping all others. (Note that in the incompressible case the original system contains only the gradient of the pressure, therefore the actual value of the background constant for the pressure does not matter.) In so doing, we arrive at (2.1a), (2.1b) for the incompressible flow and (2.1a), (2.1c) for the compressible flow. In both (2.1b) and (2.1c) Re is the Reynolds number (in the turbulent case, it is an effective far-field value; see [15]); in addition, for the compressible flow (see (2.1c)), $M_0 = u_0/\left(\gamma p_0/\rho_0\right)^{1/2}$ is the free-stream Mach number (always $M_0 < 1$), Pr is the Prandtl number, and γ is the ratio of specific heats.

For the incompressible case it is clear that the differential equations in small perturbations are linear. For the compressible case, however, this fact may require some additional justification; see section 2.1.2.

System (2.1) describes the flow in the far field. In both incompressible and compressible cases, it is supplemented by the boundary condition

$$\boldsymbol{u} \longrightarrow \mathbf{o}, \quad \text{as} \quad r \equiv (x^2 + y^2 + z^2)^{1/2} \longrightarrow +\infty, \tag{2.2}$$

which means that all the perturbations vanish at infinity, or equivalently, the flow approaches the free-stream limit.

Let us mention, that the matrices $\boldsymbol{C}, \boldsymbol{D}, \boldsymbol{E}, \boldsymbol{F}, \boldsymbol{H}$ of (2.1b) are symmetric, whereas the matrices $\boldsymbol{C}, \boldsymbol{D}, \boldsymbol{E}, \boldsymbol{F}, \boldsymbol{H}$ of (2.1c) are not. As the symmetric form of the matrices may sometimes be more convenient for the analysis and also more suitable for the numerical calculations (especially when the Mach number M_0 is low), one can use the transformation proposed by Abarbanel and Gottlieb in [25] to simultaneously reduce $\boldsymbol{C}, \boldsymbol{D}, \boldsymbol{E}, \boldsymbol{F}, \boldsymbol{H}$ of (2.1c) to the symmetric form (and some of the matrices to diagonal form). The symmetrized form [25] of the system matrices of (2.1c) appears useful when analyzing the incompressible limit $M_0 \longrightarrow +0$. A detailed study of the compressible Euler and Navier–Stokes equations as the Mach number approaches zero can be found in [26, 27, 28].

2.1.2. Asymptotic methods—linear vs. nonlinear. In connection with the linearized model proposed above for the far field, especially as it regards the compressible case, we will discuss here one group of the ABC methods that are widely used in computations. These methods employ some asymptotic form of the far-field solution for closing the system of equations to be solved inside the computational domain. Typically, this approximate asymptotic form can be obtained as a few leading terms of the series (or asymptotic series) that represents the solution in the far field. The corresponding ABCs most often appear local. The ideas of this type were employed by various authors; see, e.g., [29, 30, 31, 32, 33, 34, 35, 36, 37].

As a rule, asymptotic ABC methods are derived on the basis of the linear (or linearized) equations. In certain cases, however, one takes into account the nonlinear corrections as well. For example, when analyzing the transonic limit $M_0 \longrightarrow 1$ for small perturbations of the velocity potential (compressible flow), some second-order

terms should formally be retained in the differential equation along with the first-order terms (see, e.g., Cole and Cook [38]). This leads to the nonlinear Kármán–Guderley equation rather than the linear Prandtl–Glauert equation (the latter is valid for smaller M_0). For two-dimensional external flows (e.g., flows around airfoils) described by the Kármán–Guderley equation, it turns out that the nonlinear corrections to the leading linear lift-based term $\sim -\Gamma\theta/2\pi$ in the far-field expansion of the potential (Γ is the flow circulation, θ is the polar angle) contain terms proportional to $\log r/r$ (r is the polar radius), which formally decay more slowly than the next linear term $\sim 1/r$ as $r \longrightarrow +\infty$. This circumstance, in particular, gave Drela in [39] and Giles and Drela in [40] reason to include the nonlinear correction terms in their simplified far-field potential model for the compressible airfoil calculations. (Note that the entire series that represents the behavior at infinity of the potential function of a two-dimensional subsonic compressible flow has been accurately constructed by Ludford in [41] on the basis of the hodograph plane techniques.)

Our two-dimensional DPM-based approach of [13], however, uses the full flow system; we never introduce the velocity potential and always consider only the linearized far-field flow. The accuracy and performance of the corresponding nonlocal ABCs are demonstrated by the numerical experiments; see [14, 15, 18]. This accuracy and performance is typically better than those of the standard methods. However, we should say that the investigated Mach numbers have never come close to the transonic limit; we have always run our calculations in the range $M_0 \lesssim 0.8$. Generally, retaining the second-order nonlinear terms in the compressible far-field model for two-dimensional flows is most relevant to the case of Mach numbers close to one, $M_0 \longrightarrow 1$, whereas the linear theory works best for $\delta \ll \left(1 - {M_0}^2\right)^{3/2}$ (see [38]); here δ can be regarded as, e.g., the airfoil thickness.

The case of the compressible far-field expansion for three space dimensions is entirely different. Let us consider here the Kármán–Guderley equation (see [38])

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial \hat{y}^2} + \frac{\partial^2\phi}{\partial \hat{z}^2} = \frac{\gamma+1}{K}\frac{\partial\phi}{\partial x}\frac{\partial^2\phi}{\partial x^2}. \tag{2.3}$$

In (2.3), ϕ is the perturbation of the full potential Φ of the flow around a thin three-dimensional wing so that

$$\frac{1}{u_0}\frac{\partial\Phi}{\partial x} = 1 + \delta^{2/3}\frac{\partial\phi}{\partial x}, \quad \frac{1}{u_0}\frac{\partial\Phi}{\partial y} = \delta\frac{\partial\phi}{\partial\tilde{y}}, \quad \frac{1}{u_0}\frac{\partial\Phi}{\partial z} = \delta\frac{\partial\phi}{\partial\tilde{z}}, \tag{2.4a}$$

$$\tilde{y} = \delta^{1/3}y, \quad \tilde{z} = \delta^{1/3}z,$$

δ is the wing thickness ($\delta \longrightarrow +0$ along with $M_0 \longrightarrow 1$ in the transonic limit),

$$K = \frac{1 - {M_0}^2}{\delta^{2/3}} \tag{2.4b}$$

is the parameter of transonic similarity (the true linear theory corresponds to large values of K; see, e.g., [42]), the additional coordinate transformation is given by

$$\hat{y} = \sqrt{K}\tilde{y}, \quad \hat{z} = \sqrt{K}\tilde{z}, \tag{2.4c}$$

γ is still the ratio of specific heats, and the free stream is again aligned with the positive x direction.

The far-field expansion for ϕ in the linear theory, i.e., when the right-hand side (RHS) of (2.3) is omitted, starts with the horseshoe vortex (see, e.g., [43])

$$\phi_1 = \frac{\hat{y}}{\hat{y}^2+\hat{z}^2}\left(1+\frac{x}{\hat{r}}\right) = \frac{(1+\cos\theta)\cos\varphi}{\hat{r}\sin\theta}, \tag{2.5a}$$

where the spherical coordinates are introduced as

$$x = \hat{r}\cos\theta, \quad \hat{y} = \hat{r}\sin\theta\cos\varphi, \quad \hat{z} = \hat{r}\sin\theta\sin\varphi.$$

Expression (2.5a) obviously has a singularity in the wake, i.e., along the line $\theta = 0$. Clearly, the source term of order $1/\hat{r}$ is not present in the far-field expansion because the surface of the wing is assumed closed. Therefore, the next term in the linear expansion should be proportional to $1/\hat{r}^2$. We consider its general form

$$\phi_2 = \sum_{l,m} \hat{r}^{-l-1} Y_l^m(\theta,\varphi), \quad l = 1, \quad m = -1,\ 0,\ 1,$$

where the spherical functions $Y_l^m(\theta,\varphi)$ are given by

$$Y_l^m(\theta,\varphi) = \mathcal{P}_l^m(\cos\theta)e^{im\varphi}$$

and

$$\mathcal{P}_l^m(\mu) = (1-\mu^2)^{m/2}\mathcal{P}_l^{(m)}(\mu), \quad m \le l;$$

here $\mathcal{P}_l(\mu) = \dfrac{1}{2^l l!}\dfrac{d^l}{d\mu^l}(\mu^2-1)^l$ are the Legendre polynomials. Using real representation, we obtain a general form of the second term as

$$\phi_2 \;=\; a\frac{x}{\hat{r}^3} + b\frac{\hat{y}}{\hat{r}^3} + c\frac{\hat{z}}{\hat{r}^3} \;=\; a\frac{\cos\theta}{\hat{r}^2} + b\frac{\sin\theta\cos\varphi}{\hat{r}^2} + c\frac{\sin\theta\sin\varphi}{\hat{r}^2}, \tag{2.5b}$$

where a, b, and c are some arbitrary constants.

To obtain the nonlinear corrections due to the RHS of (2.3), one can substitute linear terms (2.5) into this RHS and solve the resulting Poisson equation. For the purpose of simple demonstration, we will do that separately for ϕ_1 of (2.5a) and ϕ_2 of (2.5b), although a similar procedure can be carried out for any weighted sum of ϕ_1 and ϕ_2.

Substitution of (2.5a) into the RHS of (2.3) yields

$$\frac{1}{\hat{r}^2}\frac{\partial}{\partial\hat{r}}\left(\hat{r}^2\frac{\partial\phi}{\partial\hat{r}}\right) + \frac{1}{\hat{r}^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial\phi}{\partial\theta}\right) + \frac{1}{\hat{r}^2\sin^2\theta}\frac{\partial^2\phi}{\partial\varphi^2} = -\frac{\gamma+1}{K}\frac{3\cos\theta\sin^2\theta\cos^2\varphi}{\hat{r}^5}. \tag{2.6}$$

Note that the singularity of potential (2.5a) in the wake (along $\theta = 0$) vanishes with differentiation. The RHS of (2.6) can now be expanded with respect to the spherical functions; the corresponding finite Fourier series is, in fact, given by

$$\cos\theta\sin^2\theta\cos^2\varphi = \frac{1}{5}Y_1^0(\theta,\varphi) - \frac{1}{5}Y_3^0(\theta,\varphi) + \frac{1}{30}Y_3^2(\theta,\varphi). \tag{2.7}$$

Taking into account that spherical functions Y_l^m are actually the eigenfunctions of Beltrami's operator on the sphere (the last two terms on the left-hand side (LHS) of (2.6)), we can separate the variables in (2.6) and reduce it to a finite family of one-dimensional equations with respect to the Fourier transformation $\hat{\phi}_{l,m} \equiv \hat{\phi}$:

$$\frac{d^2\hat{\phi}}{d\hat{r}^2} + \frac{2}{\hat{r}}\frac{d\hat{\phi}}{d\hat{r}} - \frac{l(l+1)}{\hat{r}^2}\hat{\phi} = \frac{A_l}{\hat{r}^{k+2}}, \qquad l = 1,\, 3, \quad k = 3. \tag{2.8}$$

The constants A_l in (2.8) are, of course, inversely proportional to the transonic similarity parameter K of (2.4b); they also involve the coefficients of expansion (2.7). The homogeneous counterpart to (2.8) has two linearly independent solutions $\hat{\phi}_I(\hat{r}) = \hat{r}^l$ and $\hat{\phi}_{II}(\hat{r}) = \hat{r}^{-l-1}$. Therefore, solution $\hat{\phi}(\hat{r})$ to the nonhomogeneous equation (2.8) can be found in the form

$$\hat{\phi}(\hat{r}) = c_I(\hat{r})\hat{\phi}_I(\hat{r}) + c_{II}(\hat{r})\hat{\phi}_{II}(\hat{r}) \equiv c_I(\hat{r})\hat{r}^l + c_{II}(\hat{r})\hat{r}^{-l-1}, \tag{2.9a}$$

where $c_I(\hat{r})$ and $c_{II}(\hat{r})$ satisfy the system

$$\begin{bmatrix} \hat{\phi}_I(\hat{r}) & \hat{\phi}_{II}(\hat{r}) \\ \dfrac{d\hat{\phi}_I(\hat{r})}{d\hat{r}} & \dfrac{d\hat{\phi}_{II}(\hat{r})}{d\hat{r}} \end{bmatrix} \frac{d}{d\hat{r}} \begin{bmatrix} c_I(\hat{r}) \\ c_{II}(\hat{r}) \end{bmatrix} = \begin{bmatrix} 0 \\ \dfrac{A_l}{\hat{r}^{k+2}} \end{bmatrix}. \tag{2.9b}$$

Solving (2.9b) with respect to $c_I(\hat{r})$ and $c_{II}(\hat{r})$, we have

$$c_I(\hat{r}) \sim \frac{1}{\hat{r}^{l+k}}, \qquad c_{II}(\hat{r}) \sim \frac{1}{\hat{r}^{-l+k-1}}, \tag{2.10}$$

provided that $l + k \neq 0$ and $-l + k \neq 1$, respectively. As the latter conditions are met for l and k from (2.8), we substitute expressions (2.10) into equality (2.9a) and finally obtain the nonlinear correction due to the horseshoe potential of (2.5a) as

$$\phi_{1_{NL}} \sim \frac{1}{\hat{r}^3}. \tag{2.11a}$$

The same type of derivation can be performed for the doublet potential of (2.5b). Substituting ϕ_2 of (2.5b) into the RHS of (2.3), one obtains the expression proportional to $\hat{r}^{-7}$ instead of $\hat{r}^{-5}$ in the RHS of (2.6). The expansion analogous to (2.7) will now contain Y_l^m for $l = 1,\, 3,\, 5$, and after the separation of variables, (2.8) will also change accordingly: we will have $k = 5$ instead of $k = 3$ and add $l = 5$ to the set of wavenumbers. As a result, the nonlinear correction due to the potential ϕ_2 of (2.5b) can be shown to have the form

$$\phi_{2_{NL}} \sim \frac{1}{\hat{r}^5}. \tag{2.11b}$$

We see that the nonlinear correction $\phi_{1_{NL}}$ of (2.11a) decays at infinity two orders of magnitude faster than the term ϕ_1 of (2.5a) that it originates from. Analogously, the nonlinear correction $\phi_{2_{NL}}$ of (2.11b) decays at infinity three orders of magnitude faster than the corresponding term ϕ_2 of (2.5b). We therefore conclude that, unlike the two-dimensional case, *the transonic nonlinear corrections are not required* when analyzing the far field for three space dimensions. This conclusion basically coincides with the results of [38] which state that the far field around a thin three-dimensional finite-span wing is essentially linear. In other words, we have shown that the small perturbations of velocity potential of a three-dimensional compressible flow are described by linear

formulas in the far field even when $M_0 \longrightarrow 1$. This justifies the far-field linearization of section 2.1.1.

2.1.3. Outline of the algorithm. Having introduced the linearized far-field model (2.1), (2.2) (see section 2.1.1) we now obtain the combined problem: nonlinear inside the finite computational domain and linear on its infinite exterior. The nonlinear and linear parts of the problem are, of course, not independent. The interior and exterior solutions should match at the artificial boundary; consequently, the combined problem must be solved as a whole. Therefore, at first glance the new problem is no easier than the original one from the standpoint of solving it numerically because it is still formulated on an unbounded domain. However, using the methodology of Calderón's projections and the DPM [6, 7, 8], the exterior linear problem can be effectively reduced to a certain nonlocal relation formulated on the artificial boundary. The latter relation can serve as the desired ABCs.

More precisely, we introduce a special space of (vector-)functions at the artificial boundary; it will be called the *space of clear traces* [6, 7, 8]. We also define a (pseudodifferential) projection operator that maps the space of clear traces onto itself. This operator will be analogous to Calderón's boundary projections. Under certain conditions, one can show that the element of the space of clear traces is actually a trace of some solution to the problem (2.1), (2.2) on the exterior domain if and only if this element belongs to the image of the aforementioned projection operator. The latter condition can, in fact, be written in the form of Calderón's boundary equation or the so-called *boundary equation with projection (BEP)* [6, 7, 8]. Its solutions will provide us with *complete boundary classification* (in terms of the appropriate traces) of all those and only those $\boldsymbol{u}$'s (see (2.1b, c)) that solve (2.1), (2.2) outside the computational domain.

As we intend to set the ABCs for the discretized flow problem, the foregoing boundary classification of the exterior solutions will also be obtained in the discrete framework using the concept of finite-difference clear traces and finite-difference counterparts to Calderón's projections and generalized potentials [6, 7, 8]. Once we are able to calculate the image of Calderón's projection (i.e., the result of application of this operator to a given input) we can actually set the ABCs in a few different ways. Earlier (see [13, 14, 15]) we solved the corresponding BEP using some variational approach. Below, we follow a different path, namely, we implement the boundary projections directly as proposed first in [18]. In fact, applying the Calderón operator we update the missing boundary values on every cycle of the iteration procedure that is employed inside the computational domain.

2.2. Geometric issues and basics of the discrete algorithm. The specific configuration of the domains that we will be dealing with hereafter is shown in Figure 2.1.

The actual structure displayed in this figure is the well-known test wing ONERA M6 (the wingtip is blunted and is in the "hidden" area on the figure). The wing stretches span-wise along the Cartesian axis z and is assumed symmetric with respect to the plane $z = 0$. The fluid flow is uniform at infinity and aligned with the positive x direction; together with the symmetry of the wing this implies the symmetry of the entire flow pattern with respect to $z = 0$.

The latter symmetry, in turn, means that the components u^i of $\boldsymbol{u}$, $i = 1, \ldots, 4$ for (2.1b) and $i = 1, \ldots, 5$ for (2.1c) (note that the x projection of the velocity vector $u \equiv u^i$ for $i = 2$) satisfy the following set of equalities for $i \neq 4$,

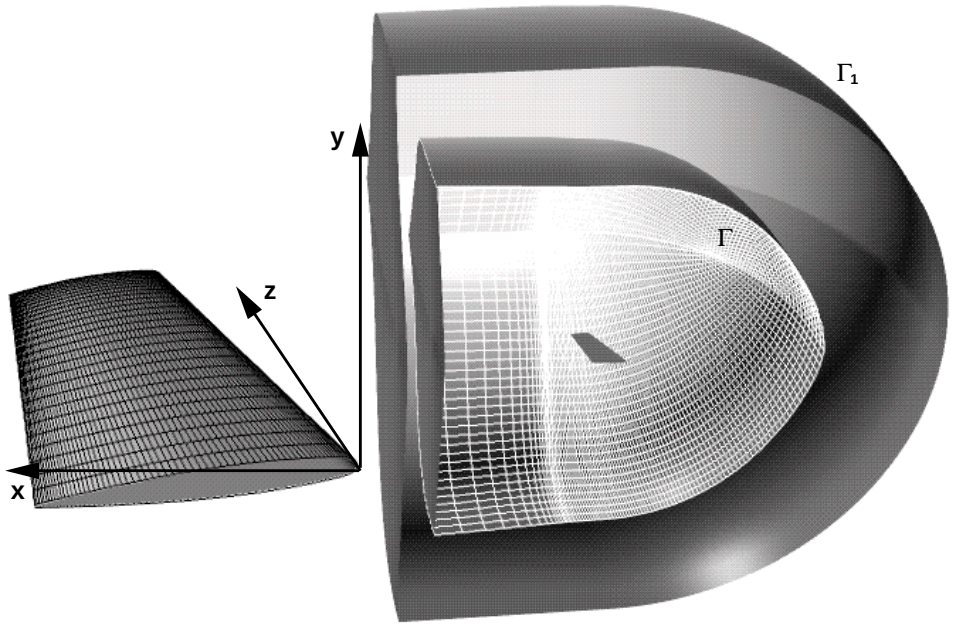

FIG. 2.1. *Schematic geometric setup; the wing on the left is enlarged.*

$$u^i(\cdot,|z|) = u^i(\cdot,-|z|),$$

$$\frac{\partial u^i}{\partial x}(\cdot,|z|) = \frac{\partial u^i}{\partial x}(\cdot,-|z|), \quad \frac{\partial u^i}{\partial y}(\cdot,|z|) = \frac{\partial u^i}{\partial y}(\cdot,-|z|), \quad \frac{\partial u^i}{\partial z}(\cdot,|z|) = -\frac{\partial u^i}{\partial z}(\cdot,-|z|),$$

$$\frac{\partial^2 u^i}{\partial y^2}(\cdot,|z|) = \frac{\partial^2 u^i}{\partial y^2}(\cdot,-|z|), \quad \frac{\partial^2 u^i}{\partial z^2}(\cdot,|z|) = \frac{\partial^2 u^i}{\partial z^2}(\cdot,-|z|), \quad \frac{\partial^2 u^i}{\partial y \partial z}(\cdot,|z|) = -\frac{\partial^2 u^i}{\partial y \partial z}(\cdot,-|z|),$$

(2.12a)

which, in particular, implies

$$\frac{\partial u^i}{\partial z}(\cdot,0) = 0, \quad \frac{\partial^2 u^i}{\partial y \partial z}(\cdot,0) = 0 \quad \text{for } i \neq 4, \tag{2.12b}$$

and satisfy the following set of equalities for $i = 4$,

$$u^i(\cdot,|z|) = -u^i(\cdot,-|z|),$$

$$\frac{\partial u^i}{\partial x}(\cdot,|z|) = -\frac{\partial u^i}{\partial x}(\cdot,-|z|), \quad \frac{\partial u^i}{\partial y}(\cdot,|z|) = -\frac{\partial u^i}{\partial y}(\cdot,-|z|), \quad \frac{\partial u^i}{\partial z}(\cdot,|z|) = \frac{\partial u^i}{\partial z}(\cdot,-|z|),$$

$$\frac{\partial^2 u^i}{\partial y^2}(\cdot,|z|) = -\frac{\partial^2 u^i}{\partial y^2}(\cdot,-|z|), \quad \frac{\partial^2 u^i}{\partial z^2}(\cdot,|z|) = -\frac{\partial^2 u^i}{\partial z^2}(\cdot,-|z|), \quad \frac{\partial^2 u^i}{\partial y \partial z}(\cdot,|z|) = \frac{\partial^2 u^i}{\partial y \partial z}(\cdot,-|z|),$$

(2.13a)

which yields

$$u^i(\cdot,0)=0,\quad \frac{\partial u^i}{\partial x}(\cdot,0)=0,\quad \frac{\partial u^i}{\partial y}(\cdot,0)=0, \tag{2.13b}$$

$$\frac{\partial^2 u^i}{\partial y^2}(\cdot,0)=0,\quad \frac{\partial^2 u^i}{\partial z^2}(\cdot,0)=0 \qquad \text{for } i=4.$$

Relations (2.12), (2.13) will be used in section 2.4 when constructing discretization for the linearized system.

The flow equations are integrated numerically on a curvilinear grid generated around the wing. The grid shown in Figure 2.1 is a one-block C-O type grid; in this paper we will use the grids of this type only. The surface designated Γ in Figure 2.1 is actually an external set of nodes of the C-O grid, i.e., the artificial boundary. Henceforth, we will also need the notation D_{in} for the interior of Γ, i.e., for the finite computational domain, and the notation D_{ex} for the infinite exterior of Γ.

The curve $\Gamma_1 \subset D_{ex}$ (see Figure 2.1) actually represents the set of ghost nodes (or centers of the ghost cells for the case of finite-volume discretization); it can also be thought of as the outermost set of nodes of the original C-O grid; the surface Γ then becomes the penultimate set of nodes. We will further assume that the linearization (2.1) is valid in D_{ex}, i.e., outside Γ, so that Γ_1 already belongs to the linear zone. The actual admissible size of D_{in} such that the perturbations can be considered sufficiently small and therefore that the assumption of linearity in D_{ex} holds, is, of course, unknown ahead of time. We verify the validity of linearization in D_{ex} a posteriori through the series of numerical experiments; see section 3.

Clearly, when the stencil of the scheme used inside D_{in} is applied to any node from Γ, it generally requires some ghost cell data. Note that for the second-order central difference schemes (like those employed in the code `FLOMG` (see [10, 11, 12]) and `TLNS3D` (see [21, 22])) the consideration of only one layer of ghost cells Γ_1 is sufficient, but the case when the stencil is more extensive, and, consequently, more ghost cells are required, can be treated similarly. Unless the missing ghost cell data are provided, i.e., obtained with the help of the ABCs, the discrete system solved inside the computational domain is subdefinite; in other words, it has less equations than it has unknowns. As mentioned in section 2.1.3, the apparatus of DPM [6, 7, 8] gives us a complete characterization of boundary traces of all exterior linear solutions. Since the linear zone D_{ex} extends from Γ to infinity and contains Γ_1, the following approach appears most natural for setting the ABCs. First, the data provided from inside D_{in} are subjected to the projection operation. The resulting projection will, by definition, admit a complement on D_{ex} that solves (2.1), (2.2). The latter complement can be calculated in the form of a generalized potential and considered on Γ_1. Altogether, this procedure yields the missing relations between the values of the solution on Γ and Γ_1. In other words, it provides for a desired closure to the discrete system solved inside D_{in}, or the ABCs. Typically, the solution algorithm inside D_{in} involves some pseudotime iterations (see section 3); then, the foregoing closing procedure is applied on every iteration cycle, more precisely, every time the ghost cells need to be updated in order to advance to the next time step.

We now describe the generalized potentials and boundary projections, as well as their finite-difference counterparts, that are required for setting the DPM-based ABCs. Note that if the potentials and projections are calculated for the true operator $\boldsymbol{L}$ of (2.1a) (this operator acts on functions $\boldsymbol{u}$ defined on the entire infinite domain D_{ex} and satisfying boundary conditions (2.2)), then the corresponding BEP appears

equivalent to the original linear problem (2.1), (2.2) (see [6, 7, 8]). We, however, will have to introduce some simplifications and carry out the DPM-based procedure for a certain approximation of problem (2.1), (2.2) (see sections 2.3 and 2.4) rather than for this problem itself. Nevertheless, the corresponding approximate solution can be made as close to the original one as initially prescribed (see [18] and below for more details). Therefore, within the accuracy of far-field linearization the resulting ABCs can be made as close to the exact ones as desired.

2.3. Foundations of the DPM-based ABCs. Here, we will first formulate and solve the so-called *auxiliary problem (AP)* for the inhomogeneous version of system (2.1) with boundary conditions (2.2). This AP will be a central element of our construction of Calderón's generalized potentials and boundary projections. In fact, the solution of the AP can be thought of as a substitute for the convolution with the fundamental solution in classical potential theory.

2.3.1. Infinite-domain AP. Let us consider a compactly supported vector-function $\{f^i\} \equiv \boldsymbol{f} \equiv \boldsymbol{f}(x,y,z)$, $\operatorname{supp}\boldsymbol{f} \subseteq D_{in}$, of dimension 4 ($i = 1,\ldots,4$ for (2.1b)) or 5 ($i = 1,\ldots,5$ for (2.1c)); in the meantime, we do not specify the concrete form of $\boldsymbol{f}$. This function $\boldsymbol{f}$ will be the RHS for the AP. The AP is initially formulated on the entire space R^3; namely, we will be looking for a solution $\boldsymbol{u}$ of system

$$\boldsymbol{L}\boldsymbol{u} = \boldsymbol{f} \tag{2.14}$$

that meets boundary condition (2.2) at infinity.

Note that when discussing regular solutions below, we assume, if necessary, that the functions involved can be represented in the form of their Fourier integrals.

PROPOSITION 2.1. *Let $\boldsymbol{f}$ be a compactly supported distribution, $\boldsymbol{f} \in \mathcal{D}'(R^3)$, $\operatorname{supp}\boldsymbol{f} \subset D_{in}$. Then, system* (2.14), (2.1b) *is solvable in the Schwartz space $\mathcal{D}'(R^3)$.*

To justify Proposition 2.1, we use the standard methodology based on application of the Fourier transform over the entire R^3 (see, e.g., [44]). Clearly, to demonstrate that (2.14) is solvable in $\mathcal{D}'(R^3)$ it is sufficient to make sure that the inverse symbol of the operator $\boldsymbol{L}$ (see (2.1)) belongs locally to $L_1(R^3)$. Denoting the dual (Fourier) variables to $(x,\,y,\,z)$ by $(\xi,\,\eta,\,\zeta)$, we can write the symbol $\boldsymbol{Q}$ of $\boldsymbol{L}$ as

$$\boldsymbol{Q} = i\xi\boldsymbol{C} + i\eta\boldsymbol{D} + i\zeta\boldsymbol{E} - \eta^2\boldsymbol{F} - \zeta^2\boldsymbol{H} - \eta\zeta\boldsymbol{J}. \tag{2.15}$$

Then, the entrees $\tilde{q}_{jk} = \tilde{q}_{kj}$, $j = 1,\ldots,4$, $k = 1,\ldots,4$, of the inverse symbol $\boldsymbol{Q}^{-1}$ for the incompressible case (2.1b) are given by

$$\begin{aligned}
&\tilde{q}_{11} = \frac{i\xi + \frac{\eta^2+\zeta^2}{\mathrm{Re}}}{\varrho^2}, && \tilde{q}_{12} = \frac{-i\xi}{\varrho^2}, && \tilde{q}_{13} = \frac{-i\eta}{\varrho^2}, && \tilde{q}_{14} = \frac{-i\zeta}{\varrho^2},\\
&\tilde{q}_{22} = \frac{\eta^2+\zeta^2}{\left(i\xi + \frac{\eta^2+\zeta^2}{\mathrm{Re}}\right)\varrho^2}, && \tilde{q}_{23} = \frac{-\xi\eta}{\left(i\xi + \frac{\eta^2+\zeta^2}{\mathrm{Re}}\right)\varrho^2}, && \tilde{q}_{24} = \frac{-\xi\zeta}{\left(i\xi + \frac{\eta^2+\zeta^2}{\mathrm{Re}}\right)\varrho^2},\\
&&& \tilde{q}_{33} = \frac{\xi^2+\zeta^2}{\left(i\xi + \frac{\eta^2+\zeta^2}{\mathrm{Re}}\right)\varrho^2}, && \tilde{q}_{34} = \frac{-\eta\zeta}{\left(i\xi + \frac{\eta^2+\zeta^2}{\mathrm{Re}}\right)\varrho^2},\\
&&&&& \tilde{q}_{44} = \frac{\xi^2+\eta^2}{\left(i\xi + \frac{\eta^2+\zeta^2}{\mathrm{Re}}\right)\varrho^2},
\end{aligned} \tag{2.16}$$

where $\varrho^2 = \xi^2+\eta^2+\zeta^2$. From (2.16) one can see that $\boldsymbol{Q}^{-1}$ has only one real singularity, located in the origin, $(\xi, \eta, \zeta) = (0, 0, 0)$. Clearly, all $\tilde{q}_{1k}$, $k = 1, \ldots, 4$, are absolutely integrable near the origin and consequently, $\tilde{q}_{1k} \in L_1^{loc}(R^3)$ for $k = 1, \ldots, 4$. As for the other $\tilde{q}_{jk}$, we introduce the spherical coordinates $\xi = \varrho\cos\theta$, $\eta = \varrho\sin\theta\cos\varphi$, $\zeta = \varrho\sin\theta\sin\varphi$, $0 \le \theta \le \pi$, $0 \le \varphi < 2\pi$, and notice that for sufficiently small ϱ's

$$\frac{\varrho^2}{\mathrm{Re}^2} \le \cos^2\theta + \frac{\varrho^2}{\mathrm{Re}^2}\sin^4\theta, \qquad 0 \le \theta \le \pi. \tag{2.17}$$

For $j = 2, \ldots, 4$ and $k = j, \ldots, 4$, estimate (2.17) immediately yields

$$|\tilde{q}_{jk}| \;\le\; \mathrm{const}\frac{\varrho^2}{\left|i\xi + \frac{\eta^2+\zeta^2}{\mathrm{Re}}\right|\varrho^2} \;=\; \frac{\mathrm{const}}{\varrho\left(\cos^2\theta + \frac{\varrho^2}{\mathrm{Re}^2}\sin^4\theta\right)^{1/2}} \;\le\; \mathrm{const}\frac{\mathrm{Re}}{\varrho^2},$$

and we therefore conclude that $\tilde{q}_{jk} \in L_1^{loc}(R^3)$ for all $j = 1, \ldots, 4$, $k = 1, \ldots, 4$. Thus, we have shown that Proposition 2.1 does hold for the incompressible case (see (2.1b)). Note that a similar proof for the two-dimensional compressible case can be found in [17].

PROPOSITION 2.2. *Let $\boldsymbol{f} \in L_1(R^3)$. Then,* (2.14), (2.1b) *may have no more than one regular solution $\boldsymbol{u} \in \mathcal{D}'(R^3)$ that vanishes at infinity, i.e., satisfies* (2.2).

Indeed, the Fourier transformation $\hat{\boldsymbol{f}}$ of the RHS is continuous on R^3 in this case. The regular solution $\boldsymbol{u}$ to (2.14), (2.2) is given by the inverse Fourier transform $\boldsymbol{u} = (\boldsymbol{Q}^{-1}\hat{\boldsymbol{f}})\check{}$. Since $\boldsymbol{Q}^{-1}\hat{\boldsymbol{f}}$ has only one real singularity (in the origin), then any other solution can differ from $\boldsymbol{u}$ by no more than an inverse Fourier transformation of a distribution concentrated in the origin. The latter can be nothing but a sum of δ-functions and their derivatives, which correspond to polynomials after the Fourier transform. Therefore, Proposition 2.2, which is actually a statement of conditional uniqueness, has been justified. Note that we have been able to establish uniqueness so easily because the inverse symbol $\boldsymbol{Q}^{-1}$ has only one isolated real singular point. This, for example, would not be the case for the Euler equations, which can be obtained by formally letting $\mathrm{Re}^{-1} = 0$.

PROPOSITION 2.3. *Let $\boldsymbol{f}$ be compactly supported and $\boldsymbol{f} \in L_2(R^3)$. Then,* (2.14), (2.1b) *has a solution $\boldsymbol{u} = \boldsymbol{u}^I + \boldsymbol{u}^{II}$, where $\boldsymbol{u}^I$ is infinitely smooth on R^3, i.e., $\boldsymbol{u}^I \in C^\infty(R^3)$, and satisfies boundary condition* (2.2), *and $\boldsymbol{u}^{II} \in L_2(R^3)$. Moreover, for any $\epsilon > 0$, one can always choose the representation $\boldsymbol{u} = \boldsymbol{u}^I+\boldsymbol{u}^{II}$ so that $\|\boldsymbol{u}^{II}\|_{L_2(R^3)} < \epsilon$.*

Consider a partition of unity $1 = g_0 + g_{\bar{0}}$, where both (scalar) functions g_0 and $g_{\bar{0}}$ are infinitely smooth on R^3, $g_0 \equiv 1$ on a ball U_{R_0} centered in the origin with the fixed radius R_0, and $g_0 \equiv 0$ outside a bigger ball $U_{R_0+\mu}$ with the radius $R_0 + \mu$, $\mu > 0$. Clearly, as $\boldsymbol{f} \in L_2(R^3)$ and $\boldsymbol{f}$ is compactly supported, then $\boldsymbol{f} \in L_1(R^3)$. The solution $\boldsymbol{u}$ is given by the inverse Fourier transform $\boldsymbol{u} = (\boldsymbol{Q}^{-1}\hat{\boldsymbol{f}})\check{} = (\boldsymbol{Q}^{-1}g_0\hat{\boldsymbol{f}})\check{}+(\boldsymbol{Q}^{-1}g_{\bar{0}}\hat{\boldsymbol{f}})\check{}$. The first term of the foregoing sum, $\boldsymbol{u}^I \equiv (\boldsymbol{Q}^{-1}g_0\hat{\boldsymbol{f}})\check{}$, is obviously an inverse Fourier transformation of a function from $L_1(R^3)$. Moreover, for any polynomial $P \equiv P(\xi, \eta, \zeta)$: $P\boldsymbol{Q}^{-1}g_0\hat{\boldsymbol{f}} \in L_1(R^3)$. Therefore, $\boldsymbol{u}^I \in C^\infty(R^3)$ and $\boldsymbol{u}^I$ meets boundary condition (2.2). The second term, $\boldsymbol{u}^{II} \equiv (\boldsymbol{Q}^{-1}g_{\bar{0}}\hat{\boldsymbol{f}})\check{}$, is an inverse Fourier transformation of a function from $L_2(R^3)$ because $\hat{\boldsymbol{f}} \in L_2(R^3)$ and $\boldsymbol{Q}^{-1}$ is bounded when $\varrho \longrightarrow +\infty$ as can be clearly seen from expressions (2.16). (The Fourier transform in this case can be regarded in the sense of Plancherel.) Therefore, $\boldsymbol{u}^{II} \in L_2(R^3)$. Clearly, both $\boldsymbol{u}^I$ and $\boldsymbol{u}^{II}$ depend on the choice of the partition of unity $1 = g_0 + g_{\bar{0}}$, i.e., on the choice of R_0. Since $\hat{\boldsymbol{u}}^{II} \equiv \boldsymbol{Q}^{-1}g_{\bar{0}}\hat{\boldsymbol{f}} \in L_2(R^3)$ for any R_0, then $\|\boldsymbol{Q}^{-1}g_{\bar{0}}\hat{\boldsymbol{f}}\|_{L_2(R^3)} \longrightarrow 0$ as

$R_0 \longrightarrow +\infty$ and, consequently, $\|\boldsymbol{u}^{II}\|_{L_2(R^3)} \longrightarrow 0$ as $R_0 \longrightarrow +\infty$. This concludes the proof of Proposition 2.3.

PROPOSITION 2.4. *Let $\hat{\boldsymbol{f}}$ be the Fourier transformation of $\boldsymbol{f}$ on R^3 and $\hat{\boldsymbol{f}} \in L_1(R^3)$. Then, system* (2.14), (2.1b) *has a continuous solution $\boldsymbol{u}$ on R^3 that meets boundary condition* (2.2).

Proposition 2.4 is obvious as in this case $\boldsymbol{Q}^{-1}\hat{\boldsymbol{f}} \in L_1(R^3)$.

In our further constructions, however, we will not always be able to guarantee the inclusion $\hat{\boldsymbol{f}} \in L_1(R^3)$ as required in Proposition 2.4. Typically, $\mathrm{supp}\boldsymbol{f} \subseteq D_{in}$ and we may also assume that $\boldsymbol{f}$ is sufficiently smooth on D_{in}. On the other hand, we do not generally require the differentiability of $\boldsymbol{f}$ on the entire R^3; $\boldsymbol{f}$ and its derivatives may have discontinuities (of the first kind) on the surface $\Gamma \equiv \partial D_{in}$. For any RHS $\boldsymbol{f}$ of this type, we will make sure that when we approximate $\boldsymbol{f}$ by a sequence of smooth functions $\boldsymbol{f}_n$, the corresponding smooth solutions $\boldsymbol{u}_n$ in a certain sense converge to the solution $\boldsymbol{u}$ guaranteed by Proposition 2.3.

Consider a sequence $\boldsymbol{f}_n$, $n = 1, 2, \ldots$, of infinitely smooth compactly supported on D_{in} functions that converges to $\boldsymbol{f}$ in the sense of $L_2(D_{in})$: $\|\boldsymbol{f} - \boldsymbol{f}_n\|_{L_2(D_{in})} \equiv \|\boldsymbol{f} - \boldsymbol{f}_n\|_{L_2(R^3)} \longrightarrow 0$ as $n \longrightarrow +\infty$. (The sequence $\boldsymbol{f}_n$ always exists because $\boldsymbol{f} \in L_2(D_{in})$ and the space $\mathcal{D}(D_{in})$ of all compactly supported infinitely smooth functions on D_{in} is everywhere dense in $L_2(D_{in})$; see, e.g., [44].) The Fourier transformation $\hat{\boldsymbol{f}}_n$ of any $\boldsymbol{f}_n \in \mathcal{D}(D_{in})$ is infinitely smooth on R^3 and decays at infinity faster than any power of r^{-1} with all its derivatives. (Fourier transform in the sense of Plancherel obviously coincides here with the standard transform in the sense of L_1.) Therefore, for any polynomial $P \equiv P(\xi, \eta, \zeta)$: $P\boldsymbol{Q}^{-1}\hat{\boldsymbol{f}}_n \in L_1(R^3)$ and, consequently, the solution $\boldsymbol{u}_n$ to the system $\boldsymbol{L}\boldsymbol{u}_n = \boldsymbol{f}_n$ is also infinitely smooth on R^3, $\boldsymbol{u}_n \in C^\infty(R^3)$, and satisfies boundary condition (2.2). We now consider the same partition of unity $1 = g_0 + g_{\bar{0}}$ as used when proving Proposition 2.3. $\boldsymbol{u}_n = (\boldsymbol{Q}^{-1}\hat{\boldsymbol{f}}_n)\check{} = \boldsymbol{u}_n^I + \boldsymbol{u}_n^{II}$, where $\boldsymbol{u}_n^I = (\boldsymbol{Q}^{-1}g_0\hat{\boldsymbol{f}}_n)\check{}$ and $\boldsymbol{u}_n^{II} = (\boldsymbol{Q}^{-1}g_{\bar{0}}\hat{\boldsymbol{f}}_n)\check{}$; clearly, both $\boldsymbol{u}_n^I,\ \boldsymbol{u}_n^{II} \in C^\infty(R^3)$ and both $\boldsymbol{u}_n^I,\ \boldsymbol{u}_n^{II} \longrightarrow 0$ when $r \longrightarrow \infty$. As $\boldsymbol{f}_n \overset{L_2(R^3)}{\longrightarrow} \boldsymbol{f}$, then $\hat{\boldsymbol{f}}_n \overset{L_2(R^3)}{\longrightarrow} \hat{\boldsymbol{f}}$ and $g_0\hat{\boldsymbol{f}}_n \overset{L_2(U_{R_0})}{\longrightarrow} g_0\hat{\boldsymbol{f}}$. Consequently, $g_0\hat{\boldsymbol{f}}_n \overset{L_1(U_{R_0})}{\longrightarrow} g_0\hat{\boldsymbol{f}}$ and therefore $g_0\hat{\boldsymbol{f}}_n \overset{L_1(R^3)}{\longrightarrow} g_0\hat{\boldsymbol{f}}$. Since $\boldsymbol{Q}^{-1} \in L_1^{loc}(R^3)$, then $\boldsymbol{Q}^{-1}g_0\hat{\boldsymbol{f}}_n \overset{L_1(R^3)}{\longrightarrow} \boldsymbol{Q}^{-1}g_0\hat{\boldsymbol{f}}$ and also for any polynomial $P \equiv P(\xi, \eta, \zeta)$: $P\boldsymbol{Q}^{-1}g_0\hat{\boldsymbol{f}}_n \overset{L_1(R^3)}{\longrightarrow} P\boldsymbol{Q}^{-1}g_0\hat{\boldsymbol{f}}$. Therefore, $\boldsymbol{u}_n^I$ uniformly converges to $\boldsymbol{u}^I = (\boldsymbol{Q}^{-1}g_0\hat{\boldsymbol{f}})\check{}$ on R^3 with all its derivatives, $\frac{\partial^{\alpha+\beta+\gamma}}{\partial x^\alpha \partial y^\beta \partial z^\gamma}\boldsymbol{u}_n^I \rightrightarrows \frac{\partial^{\alpha+\beta+\gamma}}{\partial x^\alpha \partial y^\beta \partial z^\gamma}\boldsymbol{u}^I$ as $n \longrightarrow +\infty$ (see, e.g., [45]). As for the second term, obviously $g_{\bar{0}}\hat{\boldsymbol{f}}_n \overset{L_2(R^3)}{\longrightarrow} g_{\bar{0}}\hat{\boldsymbol{f}}$ and, consequently, $\boldsymbol{Q}^{-1}g_{\bar{0}}\hat{\boldsymbol{f}}_n \overset{L_2(R^3)}{\longrightarrow} \boldsymbol{Q}^{-1}g_{\bar{0}}\hat{\boldsymbol{f}}$. Therefore, $\boldsymbol{u}_n^{II} \overset{L_2(R^3)}{\longrightarrow} \boldsymbol{u}^{II}$ as $n \longrightarrow +\infty$. Thus, we have justified the following proposition.

PROPOSITION 2.5. *Let* $\mathrm{supp}\boldsymbol{f} \subseteq D_{in}$ *and* $\boldsymbol{f} \in L_2(D_{in})$. *For any $\boldsymbol{f}_n \in \mathcal{D}(D_{in})$, the solution $\boldsymbol{u}_n$ to the system $\boldsymbol{L}\boldsymbol{u}_n = \boldsymbol{f}_n$ (see* (2.1b)*) satisfies boundary conditions* (2.2) *and $\boldsymbol{u}_n \in C^\infty(R^3)$. Moreover, if the sequence $\boldsymbol{f}_n \in \mathcal{D}(D_{in})$, $n = 1, 2, \ldots$, converges to $\boldsymbol{f}$ in the sense of L_2, $\boldsymbol{f}_n \overset{L_2(R^3)}{\longrightarrow} \boldsymbol{f}$, then each solution $\boldsymbol{u}_n$ can be represented as a sum of two terms, $\boldsymbol{u}_n = \boldsymbol{u}_n^I + \boldsymbol{u}_n^{II}$, where $\boldsymbol{u}_n^I,\ \boldsymbol{u}_n^{II} \in C^\infty(R^3)$, $\boldsymbol{u}_n^I,\ \boldsymbol{u}_n^{II} \longrightarrow 0$ as $r \longrightarrow \infty$, $\frac{\partial^{\alpha+\beta+\gamma}}{\partial x^\alpha \partial y^\beta \partial z^\gamma}\boldsymbol{u}_n^I \rightrightarrows \frac{\partial^{\alpha+\beta+\gamma}}{\partial x^\alpha \partial y^\beta \partial z^\gamma}\boldsymbol{u}^I$, and $\boldsymbol{u}_n^{II} \overset{L_2(R^3)}{\longrightarrow} \boldsymbol{u}^{II}$ as $n \longrightarrow +\infty$. Here $\boldsymbol{u}^I$ and $\boldsymbol{u}^{II}$ are the same as in Proposition* 2.3.

2.3.2. Finite-domain AP. To implement the DPM-based ABCs in practice (section 3), we will need to be able to actually calculate the solution to the auxiliary

problem. Since the formulation of the AP from section 2.3.1 still involves infinite domain, we replace it by an approximate finite-domain formulation that allows easy numerical solution.

As any linear system of PDEs with constant coefficients, (2.14) admits the separation of variables in the Cartesian coordinates. Therefore, we implement Fourier's transform in the cross-stream and span-wise directions,

$$\hat{\boldsymbol{u}}(x,\,\eta,\,\zeta) = \frac{1}{2\pi} \iint\limits_{-\infty}^{\infty} \boldsymbol{u}(x,\,y,\,z) e^{-i\eta y - i\zeta z}\, dy\, dz, \tag{2.18a}$$

$$\hat{\boldsymbol{f}}(x,\,\eta,\,\zeta) = \frac{1}{2\pi} \iint\limits_{-\infty}^{\infty} \boldsymbol{f}(x,\,y,\,z) e^{-i\eta y - i\zeta z}\, dy\, dz, \tag{2.18b}$$

and obtain a family of one-dimensional systems

$$\boldsymbol{C}\frac{d\hat{\boldsymbol{u}}}{dx} + \left(i\eta\boldsymbol{D} + i\zeta\boldsymbol{E} - \eta^2\boldsymbol{F} - \zeta^2\boldsymbol{H} - \eta\zeta\boldsymbol{J}\right)\hat{\boldsymbol{u}} = \hat{\boldsymbol{f}} \tag{2.19}$$

that we consider on the entire line $-\infty < x < \infty$ for all $(\eta,\,\zeta) \in R^2$. Each system (2.19) is supplemented by the boundary condition

$$|\hat{\boldsymbol{u}}(x,\,\eta,\,\zeta)| \le \text{const} \qquad \text{for} \quad -\infty < x < \infty, \tag{2.20}$$

which actually implies $|\hat{\boldsymbol{u}}(x,\,\eta,\,\zeta)| \longrightarrow 0$ as $|x| \longrightarrow \infty$ (compare to (2.2)) if the matrix $\boldsymbol{Q} \equiv \boldsymbol{C}^{-1}\left(i\eta\boldsymbol{D} + i\zeta\boldsymbol{E} - \eta^2\boldsymbol{F} - \zeta^2\boldsymbol{H} - \eta\zeta\boldsymbol{J}\right)$ has no zero eigenvalues. The only special case, for which the decay of $\hat{\boldsymbol{u}}(x,\,\eta,\,\zeta)$ when $|x| \longrightarrow \infty$ cannot be guaranteed, is $(\eta,\,\zeta) = (0,\,0)$; therefore, we generally set the boundary conditions in the form (2.20). However, it has been shown in [13] that after the inverse Fourier transform the solution $\boldsymbol{u}$ will still vanish as $|x|$ increases.

Note that, although designated by the same notation, $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{f}}$ in formulas (2.18), as well as $\boldsymbol{Q}$, are not the same here as in the previous section. Indeed, the direction x has been left out when calculating Fourier's transformations (2.18). This has been done because the natural spatial anisotropy prescribed by the free-stream direction exists in our model and therefore the stream-wise coordinate x will be given special treatment in the finite-domain AP.

Generally, the solution $\hat{\boldsymbol{u}}(x,\,\eta,\,\zeta)$ to problem (2.19), (2.20) can be found as a convolution

$$\hat{\boldsymbol{u}}(x,\,\eta,\,\zeta) = \int\limits_{-\infty}^{\infty} \boldsymbol{G}_1(x - x',\,\eta,\,\zeta)\hat{\boldsymbol{f}}(x',\,\eta,\,\zeta)\, dx' \tag{2.21}$$

with the corresponding one-dimensional fundamental solution $\boldsymbol{G}_1(x,\,\eta,\,\zeta)$. Then, the solution $\boldsymbol{u}$ to $\boldsymbol{L}\boldsymbol{u} = \boldsymbol{f}$ can be restored by means of the inverse Fourier transform, which eventually yields

$$\begin{aligned} \boldsymbol{u}(x,\,y,\,z) \;=\; & \frac{1}{(2\pi)^2} \iint\limits_{-\infty}^{\infty} e^{iy\eta + iz\zeta} \int\limits_{-\infty}^{\infty} \boldsymbol{G}_1(x - x',\,\eta,\,\zeta) \\ & \times \iint\limits_{-\infty}^{\infty} \boldsymbol{f}(x',\,y',\,z') e^{-i\eta y' - i\zeta z'}\, dy'\, dz'\, dx'\, d\eta\, d\zeta. \end{aligned} \tag{2.22}$$

Now, let us consider the new formulation of the AP that is *periodic in both cross-stream and span-wise directions*. Specifically, we introduce the periods Y and Z for the coordinates y and z, respectively, replace Fourier's integrals with Fourier's series, and, instead of (2.22), obtain

$$\begin{aligned}\boldsymbol{u}_{YZ}(x,\,y,\,z) \;&=\; \frac{1}{YZ}\sum_{k_z=-\infty}^{k_z=\infty}\sum_{k_y=-\infty}^{k_y=\infty} e^{iy\frac{2\pi k_y}{Y}+iz\frac{2\pi k_z}{Z}}\int\limits_{-\infty}^{\infty}\boldsymbol{G}_1\left(x-x',\,\frac{2\pi k_y}{Y},\,\frac{2\pi k_z}{Z}\right)\\ &\quad\times\iint\limits_{-\infty}^{\infty}\boldsymbol{f}(x',\,y',\,z')e^{-i\frac{2\pi k_y}{Y}y'-i\frac{2\pi k_z}{Z}z'}\,dy'\,dz'\,dx'.\end{aligned} \tag{2.23}$$

In our previous work (see [13, 17, 18]), we analyzed similar periodic formulations for the two-dimensional case. It has been shown that for any fixed-size subdomain the periodic solution will converge to the original nonperiodic solution as the period increases. These results can be transferred to the case of three space dimensions without changes. Namely, let Y_0 and Z_0 be fixed. Then,

$$\begin{gathered}\boldsymbol{u}_{YZ}(x,\,y,\,z)\longrightarrow\boldsymbol{u}(x,\,y,\,z)\quad\text{as}\quad(Y,\,Z)\longrightarrow(+\infty,\,+\infty),\\ \text{when}\quad -\frac{Y_0}{2}<y<\frac{Y_0}{2},\quad -\frac{Z_0}{2}<z<\frac{Z_0}{2}.\end{gathered} \tag{2.24}$$

Typically, the type of convergence considered in (2.24) is uniform. Convergence for the derivatives can also be established under some additional conditions (see [18]). Finally, as we are going to solve the AP by a finite-difference method (section 3), certain relations between the period(s) and grid size(s) should hold; see [13, 17] for more details. We also note that convergence on a fixed-size domain is sufficient for our purposes because for constructing the ABCs we will need to know the solution of the AP only on some neighborhood of the artificial boundary.

Thus, we have replaced the original infinite-domain AP with the new problem formulated on the domain $[-\infty,\,\infty]\times[-Y/2,\,Y/2]\times[-Z/2,\,Z/2]$. This domain is still infinite in the stream-wise direction. To make the entire formulation truly finite, we first introduce some interval $[0,\,X]$ so that $[0,\,X]\times[-Y/2,\,Y/2]\times[-Z/2,\,Z/2]\supset\Gamma_1$. Consequently, systems (2.19) will be homogeneous outside $[0,\,X]$ for all $(\eta_{k_y},\zeta_{k_z})\equiv(\frac{2\pi k_y}{Y},\,\frac{2\pi k_z}{Z})$. Then, boundary condition

$$\left[\prod_{\Re\lambda(\boldsymbol{k})<0}\left(\boldsymbol{Q}(\boldsymbol{k})-\lambda(\boldsymbol{k})\boldsymbol{I}\right)\right]\hat{\boldsymbol{u}}(0,\boldsymbol{k})=\mathbf{o} \tag{2.25a}$$

prohibits the nondecreasing modes in the solution of the corresponding homogeneous system as $x\longrightarrow-\infty$, and boundary condition

$$\left[\prod_{\Re\lambda(\boldsymbol{k})\geq0}\left(\boldsymbol{Q}(\boldsymbol{k})-\lambda(\boldsymbol{k})\boldsymbol{I}\right)\right]\hat{\boldsymbol{u}}(X,\boldsymbol{k})=\mathbf{o} \tag{2.25b}$$

prohibits the modes that increase as $x\longrightarrow+\infty$. Therefore, boundary conditions (2.25) are equivalent to (2.20) in the sense that the solution to (2.19), (2.25) on $[0,\,X]$

will be the same as the corresponding fragment of the solution given by (2.21). In formulas (2.25), $\boldsymbol{k} \equiv (k_y, k_z)$, $\hat{\boldsymbol{u}}(\cdot, \boldsymbol{k}) \equiv \hat{\boldsymbol{u}}(\cdot, \eta_{k_y}, \zeta_{k_z})$, $\boldsymbol{Q}(\boldsymbol{k}) \equiv \boldsymbol{C}^{-1}(i\eta_{k_y}\boldsymbol{D} + i\zeta_{k_z}\boldsymbol{E} - \eta_{k_y}^2\boldsymbol{F} - \zeta_{k_z}^2\boldsymbol{H} - \eta_{k_y}\zeta_{k_z}\boldsymbol{J})$, $\lambda(\boldsymbol{k})$ are the eigenvalues of $\boldsymbol{Q}(\boldsymbol{k})$, and $\boldsymbol{I}$ is the identity matrix of appropriate dimension.

The formulation of the finite-domain AP is therefore complete. For a given compactly supported RHS $\boldsymbol{f}$, $\mathrm{supp}\boldsymbol{f} \subseteq D_{in}$, it consists of solving system (2.14) on the parallelepiped $[0, X] \times [-Y/2, Y/2] \times [-Z/2, Z/2]$ with periodicity boundary conditions in the y and z directions and boundary conditions (2.25) in the x direction. As mentioned above, by increasing the periods Y and Z one can make the solution to this AP arbitrarily close to the original nonperiodic solution on any finite fixed neighborhood of D_{in}.

We will designate the Green's, i.e., inverse, operator of the finite-domain AP by $\boldsymbol{G}$ so that if $\boldsymbol{L}\boldsymbol{u} = \boldsymbol{f}$, then $\boldsymbol{u} = \boldsymbol{G}\boldsymbol{f}$. We also introduce the space $\mathcal{F} \ni \boldsymbol{f}$ of the RHSs for the finite-domain AP (for all $\boldsymbol{f} : \mathrm{supp}\boldsymbol{f} \subseteq D_{in}$) and the space $\mathcal{U} \ni \boldsymbol{u}$ of its solutions so that $\boldsymbol{L} : \mathcal{U} \longmapsto \mathcal{F}$ and $\boldsymbol{G} : \mathcal{F} \longmapsto \mathcal{U}$. Keeping in mind that the functions $\boldsymbol{u} \in \mathcal{U}$ approximate the solutions to the infinite-domain AP of section 2.3.1 in the sense mentioned above, we will henceforth consider those $\boldsymbol{u} \in \mathcal{U}$ as satisfying the appropriate boundary conditions at infinity.

2.3.3. Generalized potentials and boundary projections. Let us now introduce the space of clear traces Ξ. The elements $\xi \in \Xi$ are the vector-functions defined on the artificial boundary Γ; typically, for any $\boldsymbol{u} \in \mathcal{U}$ we may consider $\xi = \left(\boldsymbol{u}, \frac{\partial \boldsymbol{u}}{\partial n}\right)\Big|_\Gamma$, where n is the normal to Γ. The concept of clear trace is delineated in [6, 7]. The operator $\boldsymbol{Tr} : \mathcal{U} \longmapsto \Xi$ that associates the clear trace with each $\boldsymbol{u} \in \mathcal{U}$ is called the clear trace operator.

Now let some $\xi \in \Xi$ be prescribed. One can always find a compactly supported function $\boldsymbol{v}$ such that $\boldsymbol{Tr}\,\boldsymbol{v} = \xi$. Then, the truncated function $\boldsymbol{f} = (\boldsymbol{L}\,\boldsymbol{v})|_{D_{in}} \in \mathcal{F}$ can be an RHS for the finite-domain AP. The corresponding solution of the finite-domain AP considered only on D_{ex} is called the generalized potential with the density ξ: $\boldsymbol{P}\xi \stackrel{def}{=} \left[\boldsymbol{G}\left((\boldsymbol{L}\,\boldsymbol{v})|_{D_{in}}\right)\right]\Big|_{D_{ex}}$. The generalized potential can be shown to depend only on its density ξ and not on the choice of $\boldsymbol{v}$ (see [6, 7]).

The composition of operators $\boldsymbol{P}$ and $\boldsymbol{Tr}$, $\boldsymbol{P}_\Gamma \stackrel{def}{=} \boldsymbol{Tr}\,\boldsymbol{P}$, maps the space of clear traces onto itself, $\boldsymbol{P}_\Gamma : \Xi \longmapsto \Xi$. This new operator is a projection, $\boldsymbol{P}_\Gamma^2 = \boldsymbol{P}_\Gamma$, and is called the generalized boundary projection. Those and only those $\xi \in \Xi$ that belong to the image of the generalized boundary projection, $\xi \in \mathrm{Im}\boldsymbol{P}_\Gamma$, or in other words, that satisfy the boundary equation with projection $\xi = \boldsymbol{P}_\Gamma\xi$, are actually traces of some $\boldsymbol{u} \in \mathcal{U}$.

In the next section, we construct the finite-difference counterparts to the generalized potentials and boundary projections and apply those to setting the ABCs.

2.4. Computation of the DPM-based ABCs.

2.4.1. Formulation of the difference AP. Let us introduce a Cartesian grid on the parallelepiped $[0, X]\times[-Y/2, Y/2]\times[0, Z/2] \supset \Gamma_1$. By virtue of symmetry (see section 2.2 and, in particular, formulas (2.12), (2.13)), we may consider only half the domain along the coordinate z. The x-grid is uniform with the size h_x: $x_m = mh_x$, $m = 0, 1, \ldots, M$, $x_0 = 0$, $x_M = X$. The grids in y and z can also be uniform with the sizes h_y and h_z, respectively: $y_{j_y} = -Y/2 + j_y h_y$, $j_y = 0, 1, \ldots, 2J_y + 1$, $y_0 = -Y/2$, $y_{2J_y+1} = Y/2$; and $z_{j_z} = -Z/2 + j_z h_z$, $j_z = J_z, \ldots, 2J_z + 1$, $z_{J_z} = -h_z/2$, $z_{2J_z+1} = Z/2$. (For the z-grid, we use here the same indexing of nodes as if we

considered the entire interval $[-Z/2,\, Z/2]$ rather than only its half $[0,\, Z/2]$. This is done mostly to keep consistency in the notation.) However, as we expect to have better accuracy for bigger periods Y and Z (see section 2.3.2), it may be convenient for application to keep the y- and z-grids uniform only in the vicinity of D_{in} and then stretch them away from the computational domain. This will allow us to cover bigger periods with the same number of nodes. In so doing, we can retain the same indexing for the nodes y_{i_y} and z_{j_z} but the grid sizes h_y and h_z will no longer be constant. In all our computations (section 3), we have actually used the stretched grids in the y and z directions.

We designate the entire three-dimensional Cartesian grid by N^0,

$$\mathrm{N}^0 \equiv \left\{ (x_m, y_{j_y}, z_{j_z}) \middle|\, m = 0, 1, \ldots, M,\ j_y = 0, 1, \ldots, 2J_y + 1,\ j_z = J_z, \ldots, 2J_z + 1 \right\}.$$

The solutions $\boldsymbol{u}^h \in \mathcal{U}^h$ of the difference AP will be defined on this grid. We also introduce another Cartesian grid M^0, on which we will define the RHSs $\boldsymbol{f}^h \in \mathcal{F}^h$ of the difference AP. Compared to the nodes of the grid N^0, the nodes of the new grid M^0 are shifted by the half-size $h_x/2$ in the x direction:

$$\mathrm{M}^0 \equiv \left\{ (x_{m-1/2}, y_{j_y}, z_{j_z}) \middle|\, m = 1, \ldots, M,\ j_y = 0, 1, \ldots, 2J_y + 1,\ j_z = J_z, \ldots, 2J_z + 1 \right\},$$

where $x_{m-1/2} \equiv (m - 1/2)h_x$.

We discretize the operator $\boldsymbol{L}$ of (2.1) on the grid N^0 with the second order of accuracy. The finite-difference scheme is centered with respect to the nodes $(m - 1/2, j_y, j_z)$. To discretize $\frac{\partial \boldsymbol{u}}{\partial x}$, we use first-order differences in x; this ensures the second order of approximation because the residuals are evaluated on the same semi-integer grid M^0, on which the RHSs are specified. For the first derivatives $\frac{\partial \boldsymbol{u}}{\partial y}$ and $\frac{\partial \boldsymbol{u}}{\partial z}$, we use the three-point second-order discretization and designate the corresponding grid operators by $\boldsymbol{D}_y$ and $\boldsymbol{D}_z$, respectively. The dimension of these operators is the same as the dimension of the grid because they operate on vector-functions $\boldsymbol{u}_{\cdot,j_y,\cdot}$ and $\boldsymbol{u}_{\cdot,\cdot,j_z}$ componentwise. On the uniform grid, this discretization turns into the standard central differencing as the central node drops out, but if the grid is stretched, the discretization contains all three nonzero coefficients. The second derivatives, $\frac{\partial^2 \boldsymbol{u}}{\partial y^2}$, $\frac{\partial^2 \boldsymbol{u}}{\partial z^2}$, and $\frac{\partial^2 \boldsymbol{u}}{\partial y \partial z}$, are discretized by the appropriate compositions of the first difference derivatives $\boldsymbol{D}_y^2$, $\boldsymbol{D}_z^2$, and $\boldsymbol{D}_y \boldsymbol{D}_z$, respectively. We will designate the discrete direct operator by $\boldsymbol{L}^h$.

Now let $\boldsymbol{u}^h \equiv \boldsymbol{u}_{m,j_y,j_z}$ and $\boldsymbol{f}^h \equiv \boldsymbol{f}_{m-1/2,j_y,j_z}$. Because of the periodicity in y,

$$\boldsymbol{u}_{\cdot,0,\cdot} = \boldsymbol{u}_{\cdot,2J_y+1,\cdot}, \qquad \boldsymbol{f}_{\cdot,0,\cdot} = \boldsymbol{f}_{\cdot,2J_y+1,\cdot}. \tag{2.26a}$$

Also, because of the symmetry/antisymmetry with respect to $z = 0$ (see boundary conditions (2.12), (2.13)) and periodicity in z,

(2.26b)

$$\begin{aligned}
&u^i_{\cdot,\cdot,j_z} = u^i_{\cdot,\cdot,2J_z+1-j_z}, \quad f^i_{\cdot,\cdot,j_z} = f^i_{\cdot,\cdot,2J_z+1-j_z}, \quad j_z = 0, 1, \ldots, J_z \qquad \text{for } i \neq 4,\\
&u^i_{\cdot,\cdot,j_z} = -u^i_{\cdot,\cdot,2J_z+1-j_z}, \quad f^i_{\cdot,\cdot,j_z} = -f^i_{\cdot,\cdot,2J_z+1-j_z}, \quad j_z = 0, 1, \ldots, J_z \qquad \text{for } i = 4.
\end{aligned}$$

of although we formally enumerate the z-nodes from 0 to $2J_z + 1$ in (2.26b), these formulas in fact show how one can consider only half these nodes instead.

To replace the continuous transforms (2.18), we introduce the discrete transforms $\boldsymbol{T}^{(y)}$ and $\boldsymbol{T}^{(z)}$ so that for each i

$$\hat{u}^i_{\cdot,k_y,k_z} = \sum_{j_z=1}^{j_z=2J_z+1} \sum_{j_y=1}^{j_y=2J_y+1} T^{(z)}_{k_z,j_z} T^{(y)}_{k_y,j_y} u^i_{\cdot,j_y,j_z}, \tag{2.27a}$$

$$\hat{f}^i_{\cdot,k_y,k_z} = \sum_{j_z=1}^{j_z=2J_z+1} \sum_{j_y=1}^{j_y=2J_y+1} T^{(z)}_{k_z,j_z} T^{(y)}_{k_y,j_y} f^i_{\cdot,j_y,j_z}. \tag{2.27b}$$

The operators $\boldsymbol{T}^{(y)}$ and $\boldsymbol{T}^{(z)}$ have the inverse that we denote by $\tilde{\boldsymbol{T}}^{(y)} \equiv \boldsymbol{T}^{(y)^{-1}}$ and $\tilde{\boldsymbol{T}}^{(z)} \equiv \boldsymbol{T}^{(z)^{-1}}$, respectively, so that

$$u^i_{\cdot,j_y,j_z} = \sum_{k_z=-J_z}^{k_z=J_z} \sum_{k_y=-J_y}^{k_y=J_y} \tilde{T}^{(z)}_{j_z,k_z} \tilde{T}^{(y)}_{j_y,k_y} \hat{u}^i_{\cdot,k_y,k_z}, \tag{2.28a}$$

$$f^i_{\cdot,j_y,j_z} = \sum_{k_z=-J_z}^{k_z=J_z} \sum_{k_y=-J_y}^{k_y=J_y} \tilde{T}^{(z)}_{j_z,k_z} \tilde{T}^{(y)}_{j_y,k_y} \hat{f}^i_{\cdot,k_y,k_z}. \tag{2.28b}$$

We require that the operator $\boldsymbol{T}^{(y)}$ diagonalize the first and, consequently, second difference derivative with respect to y:

$$\tilde{\boldsymbol{T}}^{(y)} \boldsymbol{D}_y \boldsymbol{T}^{(y)} = diag\left\{i\eta_{k_y}\right\}, \qquad \tilde{\boldsymbol{T}}^{(y)} \boldsymbol{D}_y^2 \boldsymbol{T}^{(y)} = diag\left\{-\eta^2_{k_y}\right\}, \tag{2.29a}$$

where η_{k_y}, $k_y = -J_y, \ldots, J_y$, are real. Similarly, we require that the operator $\boldsymbol{T}^{(z)}$ diagonalize the first and second difference derivatives with respect to z:

$$\tilde{\boldsymbol{T}}^{(z)} \boldsymbol{D}_z \boldsymbol{T}^{(z)} = diag\left\{i\zeta_{k_z}\right\}, \qquad \tilde{\boldsymbol{T}}^{(z)} \boldsymbol{D}_z^2 \boldsymbol{T}^{(z)} = diag\left\{-\zeta^2_{k_z}\right\}, \tag{2.29b}$$

where ζ_{k_y}, $k_z = -J_z, \ldots, J_z$, are also real. From (2.29a) and (2.29b) it follows that

$$\tilde{\boldsymbol{T}}^{(z)} \tilde{\boldsymbol{T}}^{(y)} \boldsymbol{D}_y \boldsymbol{D}_z \boldsymbol{T}^{(y)} \boldsymbol{T}^{(z)} = -diag\left\{\eta_{k_y}\right\} \mathrm{diag}\left\{\zeta_{k_z}\right\}. \tag{2.29c}$$

Clearly, the columns of the matrix $\boldsymbol{T}^{(y)}$ should therefore be the eigenvectors of $\boldsymbol{D}_y$ and, analogously, the columns of $\boldsymbol{T}^{(z)}$ should be the eigenvectors of $\boldsymbol{D}_z$.

Note that in practical computations on the stretched grids (section 3) the eigenvectors and eigenvalues of $\boldsymbol{D}_y$ and $\boldsymbol{D}_z$ and the inverse operators $\tilde{\boldsymbol{T}}^{(y)} \equiv \boldsymbol{T}^{(y)^{-1}}$ and $\tilde{\boldsymbol{T}}^{(z)} \equiv \boldsymbol{T}^{(z)^{-1}}$ are calculated using standard IMSL (International Mathematical and Statistical Library) subroutines. Although the resulting systems of eigenvectors are, generally speaking, not orthogonal, the accuracy of this eigenvalue and eigenvector computation, as well as the accuracy of the numerical transforms $\boldsymbol{T}^{(y)}$, $\boldsymbol{T}^{(z)}$, $\tilde{\boldsymbol{T}}^{(y)}$, and $\tilde{\boldsymbol{T}}^{(z)}$ (expansions with respect to skewed systems), has been found to greatly exceed the accuracy of the interior discretization.

If, in particular, the grids in y and z are uniform, then $\boldsymbol{T}^{(y)}$ and $\boldsymbol{T}^{(z)}$ are reduced to the well-known discrete Fourier transforms (from here on, the overbar $\bar{}$ means complex conjugate)

$$T^{(y)}_{k_y,j_y} = \frac{e^{-ik_y j_y h_y \frac{2\pi}{Y}}}{\sqrt{2J_y+1}\sqrt{2J_z+1}}, \qquad T^{(z)}_{k_z,j_z} = \frac{e^{-ik_z j_z h_z \frac{2\pi}{Z}}}{\sqrt{2J_y+1}\sqrt{2J_z+1}}, \tag{2.30a}$$

$$\tilde{T}^{(y)}_{j_y,k_y} = \bar{T}^{(y)}_{k_y,j_y} = \frac{e^{ik_y j_y h_y \frac{2\pi}{Y}}}{\sqrt{2J_y+1}\sqrt{2J_z+1}}, \qquad \tilde{T}^{(z)}_{j_z,k_z} = \bar{T}^{(z)}_{k_z,j_z} = \frac{e^{ik_z j_z h_z \frac{2\pi}{Z}}}{\sqrt{2J_y+1}\sqrt{2J_z+1}}.$$

(2.30b)

Let us now consider a special class of grids, namely, $y_{j_y} = -y_{2J_y+1-j_y}$ for $j_y = 0,\ldots,J_y$ and $z_{j_z} = -z_{2J_z+1-j_z}$ for $j_z = 0,\ldots,J_z$. Obviously, all uniform grids belong to this class; for the stretched grids it means symmetric stretching. Then, one can make sure that

$$\begin{aligned} &T^{(y)}_{-k_y,j_y} = \bar{T}^{(y)}_{k_y,j_y} \quad \text{for} \quad k_y = 0,\ldots,J_y, \\ &T^{(z)}_{-k_z,j_z} = \bar{T}^{(z)}_{k_z,j_z} \quad \text{for} \quad k_z = 0,\ldots,J_z, \end{aligned} \tag{2.31a}$$

$$\begin{aligned} &T^{(y)}_{k_y,2J_y+1-j_y} = \bar{T}^{(y)}_{k_y,j_y} \quad \text{for} \quad j_y = 0,\ldots,J_y, \\ &T^{(z)}_{k_z,2J_z+1-j_z} = \bar{T}^{(z)}_{k_z,j_z} \quad \text{for} \quad j_z = 0,\ldots,J_z, \end{aligned} \tag{2.31b}$$

and also

$$\begin{aligned} &\tilde{T}^{(y)}_{j_y,-k_y} = \bar{\tilde{T}}^{(y)}_{j_y,k_y} \quad \text{for} \quad k_y = 0,\ldots,J_y, \\ &\tilde{T}^{(z)}_{j_z,-k_z} = \bar{\tilde{T}}^{(z)}_{j_z,k_z} \quad \text{for} \quad k_z = 0,\ldots,J_z, \end{aligned} \tag{2.32a}$$

$$\begin{aligned} &\tilde{T}^{(y)}_{2J_y+1-j_y,k_y} = \bar{\tilde{T}}^{(y)}_{j_y,k_y} \quad \text{for} \quad j_y = 0,\ldots,J_y, \\ &\tilde{T}^{(z)}_{2J_z+1-j_z,k_z} = \bar{\tilde{T}}^{(z)}_{j_z,k_z} \quad \text{for} \quad j_z = 0,\ldots,J_z. \end{aligned} \tag{2.32b}$$

For the discrete Fourier transform on uniform grids, relations (2.31) and (2.32) immediately follow from (2.30a) and (2.30b), respectively; for nonuniform grids these relations are verified experimentally.

Substituting (2.31) into (2.27b), taking into account relations (2.26) and also that $\boldsymbol{f}^h$ is real, we obtain for $i \neq 4$ ($\Re$ means the real part)

$$\begin{aligned} \hat{f}^i_{\cdot,|k_y|,|k_z|} \;=\; &\sum_{j_y=1}^{j_y=2J_y+1} \Bigg[\sum_{j_z=J_z+1}^{j_z=2J_z} \Big(2T^{(y)}_{|k_y|,j_y} \Re T^{(z)}_{|k_z|,j_z} f^i_{\cdot,j_y,j_z} \Big) \\ &+ T^{(y)}_{|k_y|,j_y} \Re T^{(z)}_{|k_z|,2J_z+1} f^i_{\cdot,j_y,2J_z+1} \Bigg], \quad \hat{f}^i_{\cdot,|k_y|,-|k_z|} = \hat{f}^i_{\cdot,|k_y|,|k_z|}, \end{aligned} \tag{2.33a}$$

$$\hat{f}^i_{\cdot,-|k_y|,|k_z|} = \hat{f}^i_{\cdot,-|k_y|,-|k_z|} = \bar{\hat{f}}^i_{\cdot,|k_y|,|k_z|},$$

and for $i = 4$ ($\Im$ means the imaginary part)

$$\hat{f}^i_{\cdot,|k_y|,|k_z|} \;=\; i \sum_{j_y=1}^{j_y=2J_y+1} \sum_{j_z=J_z+1}^{j_z=2J_z} 2T^{(y)}_{|k_y|,j_y} \Im T^{(z)}_{|k_z|,j_z} f^i_{\cdot,j_y,j_z}, \tag{2.33b}$$

$$\hat{f}^i_{\cdot,|k_y|,-|k_z|} = -\hat{f}^i_{\cdot,|k_y|,|k_z|}, \quad \hat{f}^i_{\cdot,-|k_y|,|k_z|} = -\bar{\hat{f}}^i_{\cdot,|k_y|,|k_z|},$$

$$\hat{f}^i_{\cdot,-|k_y|,-|k_z|} = \bar{\hat{f}}^i_{\cdot,|k_y|,|k_z|}.$$

The relations similar to (2.33) can also be obtained for $\boldsymbol{u}^h$ on the basis of formulas (2.26), (2.27a), and (2.31). Furthermore, taking into account that $\boldsymbol{u}^h$ is real and using formulas (2.28a) and (2.32), we obtain for the inverse transform, if $i \neq 4$,

$$\begin{aligned} u^i_{\cdot,j_y,j_z} &= 4 \sum_{k_z=1}^{k_z=J_z} \sum_{k_y=1}^{k_y=J_y} \left(\Re \tilde{T}^{(z)}_{j_z,k_z} \Re \tilde{T}^{(y)}_{j_y,k_y} \Re \hat{u}^i_{\cdot,k_y,k_z} - \Re \tilde{T}^{(z)}_{j_z,k_z} \Im \tilde{T}^{(y)}_{j_y,k_y} \Im \hat{u}^i_{\cdot,k_y,k_z} \right) \\ &\quad + 2 \sum_{k_z=1}^{k_z=J_z} \Re \tilde{T}^{(z)}_{j_z,k_z} \Re \tilde{T}^{(y)}_{j_y,0} \Re \hat{u}^i_{\cdot,0,k_z} \\ &\quad + 2 \sum_{k_y=1}^{k_y=J_y} \left(\Re \tilde{T}^{(z)}_{j_z,0} \Re \tilde{T}^{(y)}_{j_y,k_y} \Re \hat{u}^i_{\cdot,k_y,0} - \Re \tilde{T}^{(z)}_{j_z,0} \Im \tilde{T}^{(y)}_{j_y,k_y} \Im \hat{u}^i_{\cdot,k_y,0} \right) \\ &\quad + T^{(z)}_{j_z,0} T^{(y)}_{j_y,0} \hat{u}^i_{\cdot,0,0}\,, \end{aligned} \tag{2.34a}$$

and if $i = 4$,

$$\begin{aligned} u^i_{\cdot,j_y,j_z} &= -4 \sum_{k_z=1}^{k_z=J_z} \sum_{k_y=1}^{k_y=J_y} \left(\Im \tilde{T}^{(z)}_{j_z,k_z} \Re \tilde{T}^{(y)}_{j_y,k_y} \Im \hat{u}^i_{\cdot,k_y,k_z} + \Im \tilde{T}^{(z)}_{j_z,k_z} \Im \tilde{T}^{(y)}_{j_y,k_y} \Re \hat{u}^i_{\cdot,k_y,k_z} \right) \\ &\quad - 2 \sum_{k_z=1}^{k_z=J_z} \Im \tilde{T}^{(z)}_{j_z,k_z} \Re \tilde{T}^{(y)}_{j_y,0} \Im \hat{u}^i_{\cdot,0,k_z}. \end{aligned} \tag{2.34b}$$

Use of the transforms (2.33) and (2.34) instead of (2.27) and (2.28), respectively, allows us to calculate only *one fourth of the total number of coefficients*, namely, those for $k_y = 0, 1, \ldots, J_y$ and $k_z = 0, 1, \ldots, J_z$. This obviously implies a fourfold speedup and fourfold shrinkage of the storage requirements when implementing in practice the separation of variables for the difference AP.

In the transformed space, instead of $\boldsymbol{L}^h \boldsymbol{u}^h = \boldsymbol{f}^h$, we obtain a family of one-dimensional systems

$$\begin{gathered} \boldsymbol{A_k} \hat{\boldsymbol{u}}_{m,\boldsymbol{k}} + \boldsymbol{B_k} \hat{\boldsymbol{u}}_{m-1,\boldsymbol{k}} = \hat{\boldsymbol{f}}_{m-1/2,\boldsymbol{k}}, \\ m = 1, \ldots, M, \ \boldsymbol{k} \equiv (k_y, k_z), \\ k_y = 0, \ldots, J_y, \ k_z = 0, \ldots, J_z, \end{gathered} \tag{2.35}$$

where

$$\boldsymbol{A_k} = \frac{1}{h_x} \boldsymbol{C} + \frac{i\eta_{k_y}}{2} \boldsymbol{D} + \frac{i\zeta_{k_z}}{2} \boldsymbol{E} - \frac{\eta^2_{k_y}}{2} \boldsymbol{F} - \frac{\zeta^2_{k_z}}{2} \boldsymbol{H} - \frac{\eta_{k_y}\zeta_{k_z}}{2} \boldsymbol{J}, \tag{2.36}$$

$$\boldsymbol{B_k} = -\frac{1}{h_x} \boldsymbol{C} + \frac{i\eta_{k_y}}{2} \boldsymbol{D} + \frac{i\zeta_{k_z}}{2} \boldsymbol{E} - \frac{\eta^2_{k_y}}{2} \boldsymbol{F} - \frac{\zeta^2_{k_z}}{2} \boldsymbol{H} - \frac{\eta_{k_y}\zeta_{k_z}}{2} \boldsymbol{J}.$$

For each system (2.35), we have to specify the boundary conditions at $m = 0$ and $m = M$. Analogously to the continuous boundary conditions (2.25), the boundary conditions for the discrete system should explicitly prohibit the corresponding growing

modes of the solution. This can be achieved by setting

$$\left[\prod_{|\lambda(\boldsymbol{k})|>1}(\boldsymbol{Q_k}-\lambda(\boldsymbol{k})\boldsymbol{I})\right]\hat{\boldsymbol{u}}_{0,\boldsymbol{k}}=\mathbf{o} \tag{2.37a}$$

and

$$\left[\prod_{|\lambda(\boldsymbol{k})|\le 1}(\boldsymbol{Q_k}-\lambda(\boldsymbol{k})\boldsymbol{I})\right]\hat{\boldsymbol{u}}_{M,\boldsymbol{k}}=\mathbf{o}, \tag{2.37b}$$

where $\boldsymbol{Q_k}=\boldsymbol{A_k}^{-1}\boldsymbol{B_k}$, $\lambda(\boldsymbol{k})$ are the eigenvalues of $\boldsymbol{Q_k}$, and $\boldsymbol{I}$ is the identity matrix of appropriate dimension.

The finite-difference AP has thus been formulated completely. It consists of solving the discrete system $\boldsymbol{L}^h\boldsymbol{u}^h=\boldsymbol{f}^h$ on the grid N^0 with the RHS specified on the grid M^0. The boundary conditions in the directions y and z are periodicity and symmetry; see (2.26). The boundary conditions in the direction x are specified by formulas (2.37) in the transformed space separately for each component after the original system $\boldsymbol{L}^h\boldsymbol{u}^h=\boldsymbol{f}^h$ has been reduced to (2.35), (2.36) by the separation of variables (2.33), (2.34). The methodology for solving systems (2.35), (2.36) with boundary conditions (2.37), as well as the specific structure of these boundary conditions, is studied in the next section.

2.4.2. Solvability of the difference AP. Let us first concentrate here on the incompressible case, when the 4×4 system matrices are given in (2.1b). For simplicity, we will temporarily omit the indices $\boldsymbol{k}$. If $\eta\equiv\eta_{k_y}\neq0$ or $\zeta\equiv\zeta_{k_z}\neq0$, then the solutions $\lambda_s\equiv\lambda_s(\boldsymbol{k})$ and $\boldsymbol{e}_s\equiv\boldsymbol{e}_s(\boldsymbol{k})$, $s=1,\ldots,4$, of the problem $\boldsymbol{B_k}\boldsymbol{e}-\lambda\boldsymbol{A_k}\boldsymbol{e}=\mathbf{o}$, are given by

$$\begin{aligned}
\lambda_1 &= -\left(\frac{1}{h_x}-\frac{\eta^2+\zeta^2}{2Re}\right)\left(\frac{1}{h_x}+\frac{\eta^2+\zeta^2}{2Re}\right)^{-1}=\lambda_2, \qquad (2.38)\\
\lambda_3 &= \left(\frac{\sqrt{\eta^2+\zeta^2}}{2}-\frac{1}{h_x}\right)\left(\frac{\sqrt{\eta^2+\zeta^2}}{2}+\frac{1}{h_x}\right)^{-1},\\
\lambda_4 &= \left(\frac{\sqrt{\eta^2+\zeta^2}}{2}+\frac{1}{h_x}\right)\left(\frac{\sqrt{\eta^2+\zeta^2}}{2}-\frac{1}{h_x}\right)^{-1},\\
\boldsymbol{e}_1 &= [0,\,0,\,-\zeta,\,\eta]^t,\\
\boldsymbol{e}_2 &= \left[0,\,1,\,\frac{-i\eta}{Re},\,\frac{-i\zeta}{Re}\right]^t,\\
\boldsymbol{e}_3 &= \left[-\sqrt{\eta^2+\zeta^2}+\frac{\eta^2+\zeta^2}{Re},\,\sqrt{\eta^2+\zeta^2},\,-i\eta,\,-i\zeta\right]^t,\\
\boldsymbol{e}_4 &= \left[\sqrt{\eta^2+\zeta^2}+\frac{\eta^2+\zeta^2}{Re},\,-\sqrt{\eta^2+\zeta^2},\,-i\eta,\,-i\zeta\right]^t.
\end{aligned}$$

From (2.38) we see that we have to analyze two different cases. In the regular case, when $\sqrt{\eta^2+\zeta^2}/2-1/h_x\neq0$, none of the eigenvalues λ_s degenerate, the inverse $\boldsymbol{A_k}^{-1}$

exists, and the eigenvalues/eigenvectors (2.38) are also the eigenvalues/eigenvectors of $\boldsymbol{Q_k}$. The determinant of the Gram matrix constructed on the normalized eigenvectors $\boldsymbol{e}_s$ from (2.38) can be shown to be

$$Det_G = \frac{4\left(1-\frac{\eta^2+\zeta^2}{\mathrm{Re}^2}\right)^2}{\left(1+\frac{\eta^2+\zeta^2}{\mathrm{Re}^2}\right)\left(\left(1+\frac{\eta^2+\zeta^2}{\mathrm{Re}^2}\right)^2+8\right)}. \tag{2.39}$$

Therefore, if $(\eta^2+\zeta^2)\neq \mathrm{Re}^2$, then the eigenvectors $\boldsymbol{e}_s$ are linearly independent, and for $\sqrt{\eta^2+\zeta^2}/2-1/h_x\neq 0$ we can diagonalize the system (2.35):

$$\boldsymbol{S_k^{-1}Q_kS_k} = diag\left\{\lambda_s\right\}, \quad \text{where} \quad \boldsymbol{S_k}=\left[\frac{\boldsymbol{e}_1}{|\boldsymbol{e}_1|},\frac{\boldsymbol{e}_2}{|\boldsymbol{e}_2|},\frac{\boldsymbol{e}_3}{|\boldsymbol{e}_3|},\frac{\boldsymbol{e}_4}{|\boldsymbol{e}_4|}\right]. \tag{2.40}$$

Let us note that since $\eta^2 \lesssim 1/h_y^2$ and $\zeta^2 \lesssim 1/h_z^2$ (h_y and h_z are the smallest grid sizes), then the condition $(\eta^2+\zeta^2)\neq \mathrm{Re}^2$ appears to be not too restrictive. For example, the asymptotic width of the plane viscous wake in the far field behind the body is $\sim 1/\sqrt{\mathrm{Re}}$ (see, e.g., [46]). Therefore, to resolve this structure it is sufficient to have the grid sizes of order $1/\sqrt{\mathrm{Re}}$ as well (Re is an effective turbulent Reynolds number), which puts the operator $\boldsymbol{S_k}$ of (2.40) far from the possible singularity. We also note that in the formal inviscid limit $1/\mathrm{Re}\longrightarrow 0$, the determinant Det_G of the Gram matrix (see (2.39)) becomes fully independent of the wavenumbers η and ζ, which essentially means that the "extent of skewness" for the basis $\{\boldsymbol{e}_s\}$ will be constant.

The solution to the diagonalized system (2.35) is easy to find by marching those components, for which $\lambda_s\leq 1$, from left to right and those components, for which $\lambda_s>1$, from right to left. It is also easy to make sure that boundary conditions (2.37a) essentially imply that the components, for which $\lambda_s>1$, are not specified (i.e., can be arbitrary) on the left edge of the interval and the components, for which $\lambda_s\leq 1$ (those that would not decay as $m\longrightarrow -\infty$), are zero at $m=0$. Similarly, boundary conditions (2.37b) mean that the components, for which $\lambda_s\leq 1$, are not specified (i.e., any value is admitted) on the right edge of the interval and the components, for which $\lambda_s>1$ (those that would increase as $m\longrightarrow +\infty$), are zero at $m=M$. Let us also note that $1/\mathrm{Re}$ may be arbitrarily small but as long as it is positive, $|\lambda_s|\neq 1$ for all s. Consequently, we have only growing and decaying modes and no constant or oscillating modes in the solution of the corresponding homogeneous system. Therefore, in accordance with the results of [47] we have arrived at the following.

PROPOSITION 2.6. *Let $\eta_{k_y}\neq 0$ or $\zeta_{k_z}\neq 0$; let also $\sqrt{\eta_{k_y}^2+\zeta_{k_z}^2}/2-1/h_x\neq 0$. Then, system* (2.35), (2.36), (2.1b) *with boundary conditions* (2.37) *is uniquely solvable and well-posed for any compactly supported RHS $\hat{\boldsymbol{f}}_{m-1/2,\boldsymbol{k}}$. The constant in the well-posedness estimate $\|\hat{\boldsymbol{u}}_{\cdot,\boldsymbol{k}}\|\leq const\|\hat{\boldsymbol{f}}_{\cdot,\boldsymbol{k}}\|$ does not depend on M.*

Note that the system (2.35), (2.36), (2.1b), (2.37) can also be solved using the methodology of [16].

The case $\sqrt{\eta^2+\zeta^2}/2-1/h_x=0$ requires special analysis. In this case, $\lambda_3=0$ and also formally $\lambda_4=\infty$. However, it is in fact easy to make sure that both matrices $\boldsymbol{A_k}$ and $\boldsymbol{B_k}$ are singular for $\sqrt{\eta^2+\zeta^2}/2-1/h_x=0$. Let us therefore consider a regular pencil of matrices $\boldsymbol{A_k}+\mu\boldsymbol{B_k}$ (see, e.g., [48]). We can rewrite these pencil matrices as follows: $\boldsymbol{A_k}+\mu\boldsymbol{B_k}=(\boldsymbol{A_k}-\boldsymbol{B_k})+(\mu+1)\boldsymbol{B_k}\equiv\boldsymbol{A'_k}+(\mu+1)\boldsymbol{B_k}$. As $\boldsymbol{A'_k}=\frac{2}{h_x}\boldsymbol{C}$,

this matrix is nonsingular and therefore $\boldsymbol{A_k}+\mu\boldsymbol{B_k}=\boldsymbol{A'_k}(\boldsymbol{I}+(\mu+1)\boldsymbol{A'_k}^{-1}\boldsymbol{B_k})$. The combination of matrices in the brackets can be diagonalized, which yields

$$\boldsymbol{A_k}+\mu\boldsymbol{B_k}=$$

$$\boldsymbol{A'_k}\boldsymbol{S'_k}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \left(\frac{1}{2}+\frac{1}{h_xRe}\right)+\mu\left(-\frac{1}{2}+\frac{1}{h_xRe}\right) & 0 & 0 \\ 0 & 0 & \left(\frac{1}{2}+\frac{1}{h_xRe}\right)+\mu\left(-\frac{1}{2}+\frac{1}{h_xRe}\right) & 0 \\ 0 & 0 & 0 & -\mu \end{bmatrix}\boldsymbol{S'_k}^{-1},$$

(2.41)

where $\boldsymbol{S'_k}$ is the corresponding similarity transform. (It is easy to make sure that all the eigenvectors are linearly independent so that nonsingular $\boldsymbol{S'_k}$ does exist.)

From representation (2.41) we conclude that there are still three components in the solution that should be calculated by marching from left to right and one component that should be calculated by marching from right to left. This obviously matches the structure of boundary conditions (2.37) as both equalities can also be multiplied from the left by a nonsingular matrix $\boldsymbol{A'_k}$. In fact, the pencil $\boldsymbol{A_k}+\mu\boldsymbol{B_k}$ has one zero elementary divisor that corresponds to marching from right to left, at least one "infinite" elementary divisor that corresponds to marching from left to right, and may have either two finite elementary divisors or another two "infinite" elementary divisors that would also correspond to marching from left to right. Clearly, any of these marching procedures will easily lead to an M-independent estimate of the resulting solution via the prescribed RHS. Therefore, we have justified the following.

PROPOSITION 2.7. *Let* $\eta_{k_y}\neq 0$ *or* $\zeta_{k_z}\neq 0$; *let also* $\sqrt{\eta_{k_y}^2+\zeta_{k_z}^2}/2-1/h_x=0$. *Then, system* (2.35), (2.36), (2.1b) *with boundary conditions* (2.37) *is uniquely solvable and well-posed for any compactly supported RHS* $\hat{\boldsymbol{f}}_{m-1/2,\boldsymbol{k}}$. *The constant in the well-posedness estimate* $\left\|\hat{\boldsymbol{u}}_{\cdot,\boldsymbol{k}}\right\|\leq const\|\hat{\boldsymbol{f}}_{\cdot,\boldsymbol{k}}\|$ *does not depend on* M.

Let us now mention that for the discrete Fourier transforms on uniform grids $\eta_{k_y}=\sin(\frac{2\pi k_y h_y}{Y})/h_y$ and $\zeta_{k_z}=\sin\left(\frac{2\pi k_z h_z}{Z}\right)/h_z$. Then, to avoid the considerations that result in Proposition 2.7 and to restrict oneself by the case of Proposition 2.6 only, one can impose the following limitation on the grid sizes: $h_x^{-2}>(h_y^{-2}+h_z^{-2})/4$. We note also that general analysis of constant-coefficient ordinary difference equations based on the canonical forms of the corresponding pencils of matrices can be found in [49].

Analysis of the last remaining case, $\eta_{k_y}=\zeta_{k_y}=0\Longleftrightarrow\boldsymbol{k}=\mathbf{o}$, is straightforward as $\boldsymbol{Q_o}=-\boldsymbol{I}$ and the solution of (2.35), (2.37) can therefore be found by marching all the components from left to right. The well-posedness constant in this case may formally be proportional to M. However, as $\hat{\boldsymbol{f}}_{\cdot,\mathbf{o}}$ is premultiplied by $\boldsymbol{A_o}^{-1}$, which is of order $h_x=1/M$, then the overall estimate will again be M-independent.

For the compressible case (2.1c), similar results hold also. However, analytical expressions of type (2.38) are generally hard to obtain, so the actual eigenvalues and eigenvectors must be calculated numerically (we again use the standard IMSL subroutines). The critical value, for which the eigenbasis becomes singular (see Proposition 2.7), is now $\sqrt{\eta_{k_y}^2+\zeta_{k_z}^2}/2-\sqrt{1-M_0^2}/h_x=0$.

We will designate Green's, i.e., the inverse, operator of the difference AP by $\boldsymbol{G}^h$ so that if $\boldsymbol{f}^h\in\mathcal{F}^h$ and $\boldsymbol{L}^h\boldsymbol{u}^h=\boldsymbol{f}^h$, then $\boldsymbol{u}^h=\boldsymbol{G}^h\boldsymbol{f}^h$ and $\boldsymbol{u}^h\in\mathcal{U}^h$.

2.4.3. Difference potentials and projections. Let $\mathrm{St}_{m-1/2,j_y,j_z}$ be the stencil of the difference operator $\boldsymbol{L}^h$; according to section 2.4.1, we use first-order differences for the coordinate x and central-type differences and their products for the coordinates y and z. Let us also introduce the following grid sets (the overbar $\bar{D}_{in}$ here means the set-theoretical closure):

$$\mathrm{M}_{in} \stackrel{def}{=} \mathrm{M}^0 \bigcap \bar{D}_{in}, \qquad \mathrm{M}_{ex} \stackrel{def}{=} \mathrm{M}^0 \bigcap D_{ex},$$

$$\mathrm{N}_{in} = \bigcup_{(m-1/2,j_y,j_z)\in \mathrm{M}_{in}} \mathrm{St}_{m-1/2,j_y,j_z}, \qquad \mathrm{N}_{ex} = \bigcup_{(m-1/2,j_y,j_z)\in \mathrm{M}_{ex}} \mathrm{St}_{m-1/2,j_y,j_z},$$

$$\gamma = \mathrm{N}_{in} \bigcap \mathrm{N}_{ex}. \tag{2.42}$$

By definition (2.42), M_{in} and M_{ex} do not have common nodes. The sets N_{in} and N_{ex} already have some common nodes because these sets are swept by the stencil $\mathrm{St}_{m-1/2,j_y,j_z}$ as it is applied to every node from M_{in} and M_{ex}, respectively. The intersection of N_{in} and N_{ex} is called the grid boundary γ. It is actually a multilayered fringe of nodes of the auxiliary Cartesian grid concentrated in the vicinity of and straddling the continuous artificial boundary Γ. Similarly to the continuous case (section 2.3.3), the density of the generalized difference potential will be defined on the grid boundary γ. An example of the grid boundary (actually, a few planar cross sections of this set) for a typical configuration studied in this paper is shown in Figure 2.2.

The difference clear traces $\xi_\gamma \in \Xi_\gamma$ of functions $\boldsymbol{u}^h \in \mathcal{U}^h$ are now defined as merely restrictions to the grid boundary, i.e., $\boldsymbol{Tr}^h \boldsymbol{u}^h \stackrel{def}{=} \boldsymbol{u}^h\big|_\gamma = \xi_\gamma$, $\boldsymbol{Tr}^h : \mathcal{U}^h \longmapsto \Xi_\gamma$.

Now let some $\xi_\gamma \in \Xi_\gamma$ be prescribed and $\boldsymbol{v}^h$ be a grid function defined on N^0 such that $\boldsymbol{Tr}^h \boldsymbol{v}^h = \xi_\gamma$. Clearly, there are many functions $\boldsymbol{v}^h$ that would meet this condition, for example,

$$\boldsymbol{v}^h = \begin{cases} \xi_\gamma & \text{on } \gamma, \\ \boldsymbol{o} & \text{on } \mathrm{N}^0 \backslash \gamma. \end{cases} \tag{2.43}$$

Then, consider the function

$$\mathcal{F}^h \ni \boldsymbol{f}^h = \begin{cases} \boldsymbol{L}^h \boldsymbol{v}^h & \text{on } \mathrm{M}_{in}, \\ \boldsymbol{o} & \text{on } \mathrm{M}_{ex}, \end{cases} \tag{2.44}$$

where $\boldsymbol{v}^h$ is defined by (2.43), and solve the difference AP with this RHS $\boldsymbol{f}^h$ of (2.44). The resulting solution considered only on N_{ex} is called the generalized difference potential with the density ξ_γ:

$$\boldsymbol{P}^h \xi_\gamma \stackrel{def}{=} \left(\boldsymbol{G}^h \boldsymbol{f}^h\right)\big|_{\mathrm{N}_{ex}} \tag{2.45}$$

($\boldsymbol{f}^h$ in (2.45) is defined by (2.44)). Analogously to the continuous case (section 2.3.3), the generalized potential $\boldsymbol{P}^h \xi_\gamma$ of (2.45) can be shown to depend only on its density ξ_γ and not on the choice of $\boldsymbol{v}^h$ (in other words, the actual $\boldsymbol{v}^h$ in (2.44) may differ from the one presecribed by (2.43) as long as $\boldsymbol{Tr}^h \boldsymbol{v}^h = \xi_\gamma$ holds); see [7].

The composition of operators $\boldsymbol{P}^h$ and $\boldsymbol{Tr}^h$, $\boldsymbol{P}^h_\gamma \stackrel{def}{=} \boldsymbol{Tr}^h \boldsymbol{P}^h$ maps the space of the difference clear traces onto itself, $\boldsymbol{P}^h_\gamma : \Xi_\gamma \longmapsto \Xi_\gamma$. This new operator is a projection,

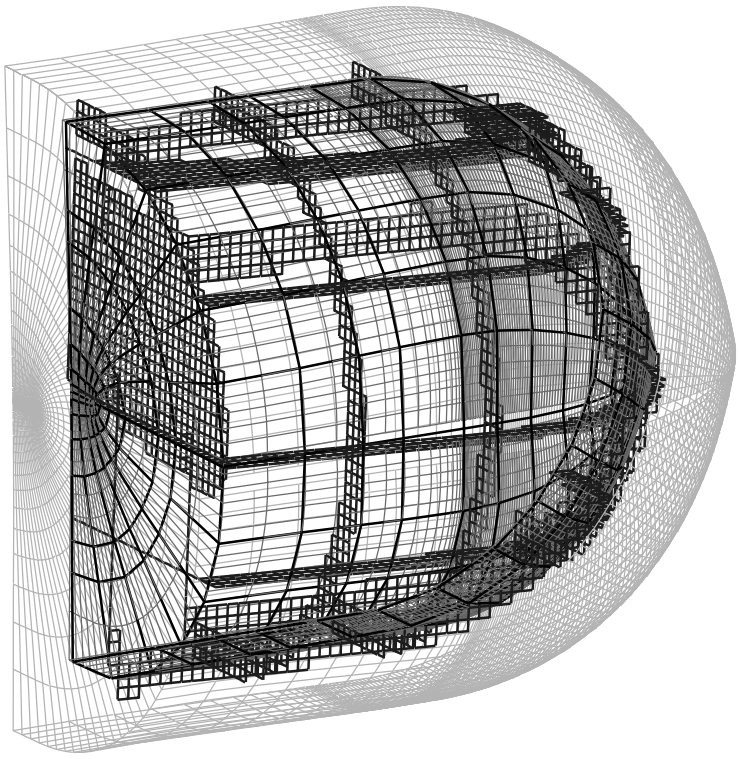

FIG. 2.2. *Continuous artificial boundary* Γ*, grid boundary* γ*, collocation grid* σ *on* Γ*, and ghost nodes* Γ_1 *for a typical three-dimensional configuration.*

$\boldsymbol{P}_\gamma^{h\,2} = \boldsymbol{P}_\Gamma^h$, and is called the generalized difference boundary projection. Those and only those $\xi_\gamma \in \Xi_\gamma$ that belong to the image of the generalized difference boundary projection, $\xi_\gamma \in \mathrm{Im}\boldsymbol{P}_\gamma^h$, or, in other words, that satisfy the difference boundary equation with projection $\xi_\gamma = \boldsymbol{P}_\gamma^h \xi_\gamma$, are actually traces of some $\boldsymbol{u}^h \in \mathcal{U}^h$.

Note that numerical verification of the projection property $\boldsymbol{P}_\gamma^{h\,2} = \boldsymbol{P}_\gamma^h$ is an ideal test for accuracy of the solution of difference AP. In our practical computations for different geometries on different grids, we have always been able to obtain, for arbitrary ξ_γ's, $\|\boldsymbol{P}_\gamma^{h\,2}\xi_\gamma - \boldsymbol{P}_\gamma^h \xi_\gamma\| < 10^{-9}$. This, in particular, justifies the use of stretched grids when solving the difference AP.

As mentioned before (section 2.3.2), we think that the continuous functions $\boldsymbol{u} \in \mathcal{U}$ satisfy the appropriate boundary conditions at infinity because the difference between the nonperiodic solution and its periodic approximation is controlled by Y and Z and can, in fact, be made as small as initially prescribed. In its own turn, the discrete space $\mathcal{U}^h$ approximates the continuous space $\mathcal{U}$; therefore, we consider those grid densities ξ_γ that belong to $\mathrm{Im}\boldsymbol{P}_\gamma^h$, $\xi_\gamma \in \mathrm{Im}\boldsymbol{P}_\gamma^h$ as boundary functions that admit the exterior complement in the right sense. In other words, these and only these functions ξ_γ admit such a complement $\boldsymbol{u}_{ex}^h \stackrel{def}{=} \boldsymbol{P}^h \xi_\gamma$ that satisfies boundary conditions of the difference AP (see (2.26), (2.37)); this complement can be made arbitrarily close (near D_{in}) to

the original linearized exterior solution; in the next section, it is used for setting the difference ABCs.

2.4.4. Global DPM-based artificial boundary conditions. Having constructed the procedure for calculating the generalized difference potentials and projections, we can now provide for a closure to the discretized Navier–Stokes system that is solved inside the computational domain D_{in}, i.e., obtain the ABCs. As mentioned in section 2.2, the interior solvers typically involve some sort of pseudotime iterations. To advance every next step of the iteration procedure, we need to know the previous-step solution everywhere on the grid, including the ghost nodes Γ_1. If these data are available, then on the next step we will know the solution everywhere except on Γ_1. Consequently, to proceed with another iteration we will have to supplement the missing data on Γ_1. This will be done by projecting the available boundary data at Γ onto the "right manifold," i.e., the one that admits the right exterior complement (see the previous section), and then calculating this complement on Γ_1. In so doing, we can obtain the missing relations between the values of the solution on Γ and Γ_1 every time the ghost nodes need to be updated.

First, let us introduce the intermediate collocation grids σ and σ_1 on both surfaces Γ and Γ_1. An example of such $\sigma \subset \Gamma$ is shown in Figure 2.2. These grids are typically a few times coarser than Γ and Γ_1. Use of the collocation grid on Γ is an element of general procedure of the difference potentials method [7]. Moreover, for the specific problem under study use of the collocation grids results in multifold acceleration of the computational procedure and also in memory savings if the boundary conditions are implemented in the matrix form (see below).

Then, let us take $\xi_\Gamma = \left(\boldsymbol{u}, \frac{\partial \boldsymbol{u}}{\partial n}\right)\Big|_\Gamma$, where n is the normal to Γ (these data are available from inside the computational domain D_{in} on every iteration) and, using the clustering $\boldsymbol{R}_\sigma$ on Γ, obtain ξ_σ. The latter procedure (clustering), in fact, implies local averaging or smoothing along Γ. Furthermore, we drop normals from all nodes γ to the surface Γ and interpolate ξ_σ with sufficiently high order to the feet of these normals. The corresponding operation is denoted $\boldsymbol{R}_n$; typically, we use the biquadratic surface interpolation. Having obtained $\boldsymbol{u}$ and $\frac{\partial \boldsymbol{u}}{\partial n}$ at the feet of the normals, we use the first two terms of the Taylor expansion (denoted π_γ) and obtain ξ_γ:

$$\xi_\gamma = \pi_\gamma \boldsymbol{R}_n \boldsymbol{R}_\sigma \left(\boldsymbol{u}, \frac{\partial \boldsymbol{u}}{\partial n}\right)\Big|_\Gamma. \tag{2.46}$$

Then, we calculate the potential $\boldsymbol{P}^h \xi'_\gamma$ for the density $\xi'_\gamma = \boldsymbol{P}^h_\gamma \xi_\gamma$ and interpolate it (operation $\boldsymbol{R}_{\sigma_1}$) from N_{ex} to the nodes $\sigma_1 \subset \Gamma_1$:

$$\boldsymbol{u}|_{\sigma_1} = \boldsymbol{R}_{\sigma_1} \boldsymbol{P}^h \xi'_\gamma \equiv \boldsymbol{R}_{\sigma_1} \boldsymbol{P}^h \xi_\gamma. \tag{2.47}$$

The second equality in (2.47) holds because of the projection property of $\boldsymbol{P}^h_\gamma$. Finally, the missing values of the solution at the nodes Γ_1 are obtained from $\boldsymbol{u}|_{\sigma_1}$ by means of interpolation along the surface Γ_1, which altogether yields the nonlocal DPM-based ABCs in the form

$$\boldsymbol{u}|_{\Gamma_1} = \boldsymbol{T} \left(\boldsymbol{u}, \frac{\partial \boldsymbol{u}}{\partial n}\right)\Big|_\Gamma; \tag{2.48}$$

here the operation $\boldsymbol{T}$ is composed of the operations (2.46), (2.47), and interpolation along Γ_1. As mentioned above, in the course of the iteration procedure boundary

condition (2.48) is applied every time we need to update the values of the solution at the ghost nodes Γ_1. The implementation of ABCs (2.48) can either be direct or involve preliminary calculation of the matrix $\boldsymbol{T}$. In the latter case, the runtime implementation of the ABCs (2.48) is reduced to a matrix-vector multiplication. Moreover, in this case we can do the first clustering $\boldsymbol{R}_\sigma$ and the last interpolation along Γ_1 separately, i.e., leave these operations out of the structure of $\boldsymbol{T}$. Then, instead of (2.48) one can write

$$\boldsymbol{u}\big|_{\sigma_1} = \boldsymbol{T}'\xi_\sigma,$$

where both the dimension of $\boldsymbol{T}'$ and its computational cost are many times smaller than those of $\boldsymbol{T}$ from (2.48).

Let us also note that we need to know the potential only on some neighborhood of the surface Γ_1 (see (2.47)). At the same time, according to (2.43) and (2.44) the density of the potential differs from zero only near γ. Therefore, for both the direct $\boldsymbol{T}^{(y)}$, $\boldsymbol{T}^{(z)}$ (see (2.27)) and inverse $\tilde{\boldsymbol{T}}^{(y)}$, $\tilde{\boldsymbol{T}}^{(z)}$ (see (2.28)) transforms, we actually have to take into account only a few "nonzero" nodes out of the total numbers of $2J_y + 1$ and $J_z + 1$ along y and z, respectively. This effectively makes the computational cost of these transforms grow linearly rather than quadratically, with respect to $2J_y+1$ and $J_z + 1$, and obviously implies a very substantial reduction of the required computer resources.

3. Numerical results.

3.1. Two-dimensional summary. For completeness we first briefly comment on the two-dimensional results from our previous work (see [14, 15, 18]), when we have calculated several subsonic and transonic viscous flows past single-element airfoils (NACA0012 and RAE2822).

The two-dimensional computational domain is formed by the C-type curvilinear grid generated around the airfoil. On this grid, the Navier–Stokes equations are integrated using the code `FLOMG` of Swanson and Turkel [10, 11, 12]. The standard treatment of external boundary in `FLOMG` is based on locally one-dimensional analysis of characteristics, which may or may not be supplemented by the point-vortex correction [33].

Basic conclusions that could be drawn from our two-dimensional numerical experience are the following. The DPM-based ABCs are geometrically universal, algorithmically simple, and easy to implement with the existing solver. For large computational domains (30–50 chords of the airfoil), performance of the standard method and performance of the DPM-based ABCs are very close to one another. However, as the distance between the artificial boundary and airfoil shrinks, discrepancy between the corresponding solutions increases. The lift and drag coefficients obtained on the basis of the two-dimensional version of boundary conditions (2.48) deviate from their asymptotic (50 chords) values much less (within fractions of 1%) than the coefficients obtained with the local ABCs do. In other words, the nonlocal DPM-based ABCs allow one to use much smaller computational domains (as small as 2–3 chords) than standard boundary conditions do and still maintain high accuracy of the numerical solution. Moreover, if we compare three models, DPM-based, point-vortex, and standard local (characteristics-based), then it turns out that the DPM-based ABCs display the best performance for small computational domains, performance of the local characteristic boundary conditions for small domains is very poor, and the point-vortex boundary conditions perform much better for the lift coefficient than they do

for the drag coefficient. This is natural since the point-vortex model is a lift-based treatment.

We also note that for certain variants of computation, the DPM-based ABCs may noticeably increase (by up to a factor of 3) the convergence rate of multigrid iterations; see [13, 14, 15]. Some discussion on the combined implementation of the DPM-based ABCs with multigrid is contained in [18]; see also [50] for more details.

3.2. Three-dimensional computations. The DPM-based boundary conditions (2.48) have been combined with the interior Navier–Stokes solver and used for calculating viscous flows around the ONERA M6 wing for different regimes that range from very low to transonic speeds and include both attached and separated turbulent flows.

We use the NASA-developed code `TLNS3D` of Vatsa et al. [21, 22] to integrate the thin-layer equations on the C-O type curvilinear grid (see Figures 2.1 and 2.2) generated around the wing. The code is based on the central-difference finite-volume discretization in space with first- and third-order artificial dissipation. Pseudotime iterations are used for obtaining the steady-state solution; the integration in time is done by the five-stage Runge–Kutta algorithm (with Courant's number calculated locally) supplemented by the residual smoothing. For the purpose of accelerating the convergence, the multigrid methodology is implemented; in our computations we used three nested grid levels with V-cycles; this three-level V-cycle algorithm is, in fact, a final stage of the full multigrid (FMG) procedure. In addition, we use the preconditioning technique of [51] to improve the convergence to steady state. We implement the DPM-based ABCs (2.48) only on the finest grid for the V-cycle in the final FMG stage; the boundary data for coarser levels are provided by the coarsening procedure. Moreover, even on the finest level we implement the DPM-based ABCs only on the first and last Runge–Kutta stages, which has been shown to make very little difference compared to the implementation on all five stages; the boundary data for the three intermediate stages are provided by the DPM-based ABCs on the first stage. Unlike the two-dimensional case, the standard treatment of external boundary in three dimensions (code `TLNS3D`) is based only on locally one-dimensional analysis of characteristics and extrapolation (as the point-vortex model is not applicable).

All three-dimensional flows that we have analyzed are turbulent. In the near field (i.e., inside D_{in}), the Navier–Stokes solver is supplemented by a special turbulence model to account for the corresponding phenomena. Depending on the specific flow variant, either an algebraic or a differential turbulence model can be employed. In the far field, we use Boussinesq's concept of the effective turbulent viscosity, i.e., effective Reynolds number (see [15]). This simplest approach has been found to produce accurate results when incorporated into the structure of the DPM-based ABCs. The value of the Prandtl number for all the calculations was either $Pr = 0.72$ (air) or $Pr = 1$.

In all the cases below, the auxiliary Cartesian grids are stretched along the coordinates y and z. The stretching typically starts outside Γ_1; the stretching factors (we use geometric progression) vary between 1.07 and 1.1 for different variants. The typical values of Y and Z that we have used vary between 20 and 30 sizes of D_{in} in the cross-stream direction and 4 and 10 sizes of D_{in} in the span-wise direction. The uniform Cartesian grid in the vicinity of D_{in} is always chosen so that the distance between Γ and Γ_1 can be well resolved.

We should also emphasize that in spite of their nonlocal nature the DPM-based ABCs (2.48) are geometrically universal. In other words, these boundary conditions can be obtained for the boundary Γ of any irregular shape by means of the same com-

TABLE 3.1
ONERA M6: $M_0 = 0.01$*;* $\mathrm{Re}_0 = 11.7 \cdot 10^6$*;* $\alpha = 3.06^\circ$.

"Average radius" of D_{in}	1.25 root chords		10 root chords	
Dimension of the grid	$193 \times 49 \times 33$		$193 \times 49 \times 33$	
Type of ABCs	Standard	DPM	Standard	DPM
Full lift, C_L	0.2052	0.1954	0.1940	0.1939
Relative error	5.78%	0.77%	0%	0%
Full drag, $C_D \times 100$	0.695	0.685	0.681	0.681
Relative error	2.1%	0.58%	0%	0%

putational procedure. This conclusion directly follows from the previous considerations and has also been corroborated repeatedly in numerical experiments. Moreover, ABCs (2.48) appear easy to incorporate in the structure of the existing flow solvers. This has been demonstrated in practice as well and is very important from the standpoint of applications. The issue of computational cost of boundary conditions (2.48) and some possible ways of its reduction will be addressed later, in section 3.3.

Finally, we would like to stress that, as has been shown in our previous work and will also be seen from the following numerical tests, one of the main advantages of the new boundary conditions is that they allow smaller computational domains. The associated potential benefit from the standpoint of practical computing is twofold. One can truncate the discretization to fit the smaller computational domain, thus obtaining a computation of comparable accuracy to the large domain computation using less computational resources. One can also use the smaller domain, but employ the same number of points, thus using the DPM-based boundary conditions to obtain a more accurate computation at roughly the same expense of the original computation. In every particular set of numerical tests described below we will follow one of these strategies.

3.2.1. Low Mach number regime. We first consider a very low speed flow, $M_0 = 0.01$, which is, in fact, close to the truly incompressible case. Preconditioning [51] makes the analysis of this flow possible with `TLNS3D`. The flow is turbulent with the molecular Reynolds number based on the root chord of the wing $\mathrm{Re}_0 = 11.7 \cdot 10^6$; the angle of attack is $\alpha = 3.06^\circ$; there is no separation and the turbulence inside D_{in} is simulated using the Baldwin–Lomax algebraic model, which is based on the concept of mixing length.

Since the free-stream Mach number is so small, we have implemented here the incompressible version of the nonlocal DPM-based ABCs (2.48) constructed on the basis of matrices (2.1b). In Table 3.1, we present the results of calculations for two different computational domains of the "average radii" of 10 and 1.25 root chords of the wing, respectively (root chord means the chord length at $z = 0$).

In both cases, we used the C-O type grids of the same dimension $193 \times 49 \times 33$; for the small domain the grid was obtained by scaling down the bigger grid and was obviously finer in the near field. Therefore, for this set of computations one can expect to achieve a better resolution of the flow field near the wing on the small domain. From Table 3.1, one can see that for the large domain the results (force coefficients C_L and C_D) obtained using both methods are very close to each other. However, as the domain shrinks the accuracy obtained with the DPM-based procedure appears to be much better than the accuracy provided by the standard methodology. In other words, the nonlocal DPM-based ABCs (2.48) allow one to substantially reduce the size of the computational domain without compromising the accuracy. This confirms that if the structure of the far-field solution is correctly taken into account by means

TABLE 3.2
ONERA M6: $M_0 = 0.5$; $Re_0 = 11.7 \cdot 10^6$; $\alpha = 3.06^\circ$.

"Average radius" of D_{in}	1.25 root chords		2 root chords		10 root chords	
Dimension of the grid	$193 \times 49 \times 33$		$193 \times 49 \times 33$		$193 \times 49 \times 33$	
Type of ABCs	Standard	DPM	Standard	DPM	Standard	DPM
Full lift, C_L	0.2218	0.2065	0.2185	0.2065	0.2081	0.2072
Relative error	6.58%	0.34%	5.0%	0.34%	0%	0%
Full drag, $C_D \times 100$	0.817	0.791	0.793	0.791	0.787	0.788
Relative error	3.8%	0.38%	0.76%	0.38%	0%	0%

of the ABCs, then within a certain range of domain sizes the computed near-field solution becomes essentially domain-independent. Moreover, as the near-field grid on the small domain is finer than on the large domain, it provides for a mechanism to improve the overall accuracy of numerical solution.

3.2.2. Subsonic regime. The next case is a subcritical (i.e., fully subsonic) compressible flow for $M_0 = 0.5$. Here, the free-stream Mach number is already high enough to make the compressibility effects essential, but on the other hand it is still not too high and therefore the flow remains subsonic throughout the entire domain. The angle of attack and the molecular Reynolds number for this case are the same as for the previous one: $\alpha = 3.06^\circ$, $\mathrm{Re}_0 = 11.7 \cdot 10^6$. The flow is also fully attached and the turbulence model inside D_{in} is algebraic (Baldwin–Lomax).

The DPM-based ABCs (2.48) for this case were constructed on the basis of non-symmetrized matrices (2.1c). For this specific value of Mach number, $M_0 = 0.5$, the "extent of nonsymmetry" in the system matrices (2.1c) still appears quite acceptable. However, for low Mach numbers $M_0 \lesssim 0.1$ treated in the compressible framework (unlike in section 3.2.1), the use of the symmetrizer [25] can be recommended. On the other hand, we should note that in [18] we have been able to obtain accurate two-dimensional results for $M_0 = 0.01$ without symmetrizing the system matrices in boundary conditions.

In Table 3.2, we compare the results of calculations for three different computational domains. Similarly to the previous case (section 3.2.1), we use the same grid dimension for the domains of different sizes; as a consequence, we expect to obtain a higher resolution of the near field on the smaller domains. As one can clearly see from Table 3.2, the DPM-based ABCs produce much more accurate solutions for the domains of (half-)size of 1.25 and 2 root chords than standard boundary conditions do. This essentially creates a vehicle for calculating the solutions unaffected by the size of the computational domain (within a certain range). Along with the grid refinement on smaller domains, it implies (at least for this series of tests) a better overall accuracy while keeping the computational cost at approximately the same level.

3.2.3. Transonic regime. Most of the standard test cases for flows around the ONERA M6 wing are transonic (see, e.g., the experimental work [52]). In such flows the free-stream Mach number is sufficiently high so that the local speed exceeds the speed of sound in some bounded region near the upper surface of the wing. This leads to the formation of a supersonic (i.e., supercritical) "bubble," which typically has a sonic–surface-type upstream boundary and a shock–wave-type downstream boundary.

Attached flow. The first transonic case that we present is $M_0 = 0.84$, $\alpha = 3.06^\circ$, $\mathrm{Re}_0 = 11.7 \cdot 10^6$. In this case, the angle of attack α remains sufficiently small so that the weak shock on the upper surface of the wing does not cause flow separation.

TABLE 3.3
ONERA M6: $M_0 = 0.84$; $\mathrm{Re}_0 = 11.7 \cdot 10^6$; $\alpha = 3.06^\circ$.

"Average radius" of D_{in}	3 root chords		10 root chords	
Dimension of the grid	$193 \times 49 \times 33$		$209 \times 57 \times 33$	
Type of ABCs	Standard	DPM	Standard	DPM
Full lift, C_L	0.298±0.004	0.2798	0.2805	0.2786
Relative error	6.24%±1.43%	0.43%	0%	0%
Full drag, $C_D \times 10$	0.168±0.008	0.1537	0.1542	0.1531
Relative error	8.95%±5.19%	0.39%	0%	0%

Therefore, we still use the Baldwin–Lomax model for simulating the turbulence inside D_{in}. An important difference compared to the previously studied cases is that here we cannot bring the artificial boundary as close to the wing as done in sections 3.2.1 and 3.2.2. The reason is that our far-field treatment is purely subsonic and, therefore, the artificial boundary should not come too close to the boundary of the supercritical bubble. Therefore, we ran our computations for two domains—the "radius" of the large one is still about 10 root chords of the wing and the "radius" of the small one is about 3 root chords of the wing. Moreover, unlike in the previous cases, here we employed a different strategy, namely, used a larger grid (i.e., more nodes) for a larger domain. Thus, provided that the new boundary conditions algorithm will again produce solutions (almost) unaffected by the domain size, we will obtain the reduction of computational cost while preserving the accuracy. Geometrically, the smaller (3 root chords) C-O grid is constructed as an exact subset of the larger (10 root chords) grid. This should completely eliminate any influence that the change of the grid in the near field may possibly exert on the calculated solution.

The nonlocal ABCs (2.48) for this case were again constructed on the basis of matrices (2.1c). In Table 3.3, we compare the computed results (calculated lift C_L and drag C_D coefficients) for two different types of ABCs on two different domains. For the small computational domain, the DPM-based ABCs again clearly outperform the standard method from the standpoint of accuracy. Moreover, the total number of nodes in the larger grid here (see Table 3.3) is about 25% more than in the smaller grid, which obviously implies a proportional difference in the associated cost of computations.

Even more important is that for this transonic case the DPM-based ABCs influence not only the final accuracy of the solution but also convergence rate of the iteration procedure employed inside D_{in}. Namely, multigrid iterations with standard ABCs on the small domain converge noticeably more slowly than they do if supplemented by the DPM-based ABCs. In fact, for the same 500 V-cycles on the finest multigrid level, we simply could not obtain a fully converged solution on the 3 root chords domain with standard boundary conditions. That's why the corresponding data in Table 3.3 are given with the error bands indicated. Convergence history for transonic computation on the 3 chords domain is given in Figure 3.1(a) for the residual of the continuity equation and in Figure 3.1(b) for the number of supersonic points in the domain. (The latter quantity is deemed very sensitive for calculation of the transonic flows.) Note that in our opinion the behavior of the corresponding curves in Figure 3.1 suggests that the standard algorithm on the 3 root chords domain still converges, although extremely slowly.

Quantitatively, from Figures 3.1 one can see that the multigrid convergence rates for different types of ABCs can differ by as much as approximately a factor of 3.

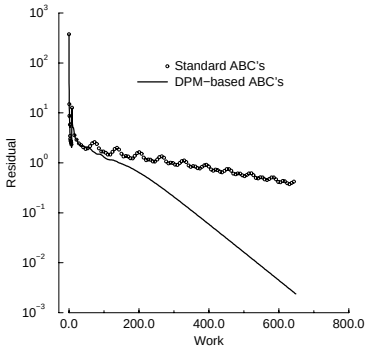

FIG. 3.1(a). *ONERA M6: $M_0 = 0.84$, $\mathrm{Re}_0 = 11.7 \cdot 10^6$, $\alpha = 3.06°$. Convergence history for the residual of the continuity equation. "Average radius" of D_{in} is 3 root chords of the wing; dimensions of the grid are* $193 \times 49 \times 33$.

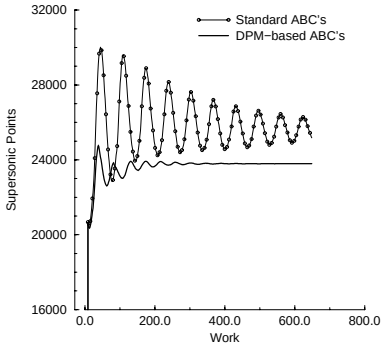

FIG. 3.1(b). *ONERA M6: $M_0 = 0.84$, $\mathrm{Re}_0 = 11.7 \cdot 10^6$, $\alpha = 3.06°$. Convergence history for the number of supersonic nodes in the domain. "Average radius" of D_{in} is 3 root chords of the wing; dimensions of the grid are* $193 \times 49 \times 33$.

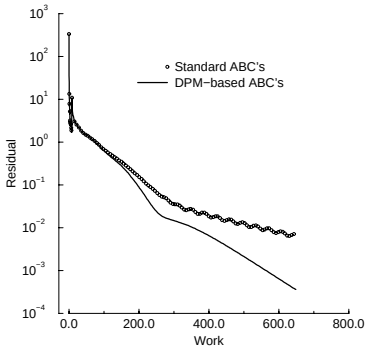

FIG. 3.2(a). *ONERA M6: $M_0 = 0.84$, $\mathrm{Re}_0 = 11.7 \cdot 10^6$, $\alpha = 3.06°$. Convergence history for the residual of the continuity equation. "Average radius" of D_{in} is 10 root chords of the wing; dimensions of the grid are* $209 \times 57 \times 33$.

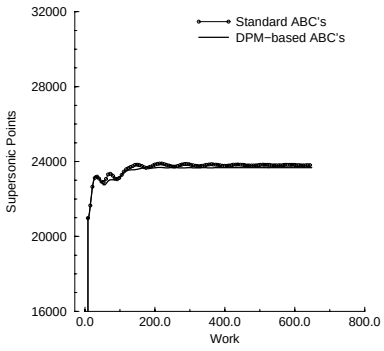

FIG. 3.2(b). *ONERA M6: $M_0 = 0.84$, $\mathrm{Re}_0 = 11.7 \cdot 10^6$, $\alpha = 3.06°$. Convergence history for the number of supersonic nodes in the domain. "Average radius" of D_{in} is 10 root chords of the wing; dimensions of the grid are* $209 \times 57 \times 33$.

The history of convergence of the same two quantities for the large (10 root chords) computational domain is presented in Figure 3.2. We see that in this case the DPM-based ABCs also provide for some convergence speedup, although the difference between the two methodologies appears less dramatic. This seems reasonable because one could generally expect that the larger the computational domain, the smaller the influence that the external boundary conditions exert on the numerical procedure.

Let us also note that on the small (3 root chords) domain the two algorithms apparently converge to quite different solutions (this is most clearly seen in Figure 3.1(b)), whereas Figure 3.2(b) allows one to assume that on the large (10 root chords) domain the final solutions are close to one another. The data from Table 3.3 corroborate these conclusions. This behavior of the solution again fits into the

aforementioned concept that the overall impact of the ABCs on the computational algorithm decreases as the domain enlarges.

Separated flow. When one increases the angle of attack α in the transonic regime, the flow pattern changes. The shock on the upper surface of the wing becomes stronger. Since the chord length of the wing decreases span-wise as z increases (see Figure 2.1), then the stream-wise dimension of the supersonic bubble decreases as well, and eventually the upstream sonic surface and the downstream shock wave meet somewhere in the area close to the wingtip. For sufficiently strong shocks this, in particular, produces flow separation on the upper surface of the wing. We have analyzed the separated flow of this type for $M_0 = 0.84$, $\alpha = 5.06^\circ$, $\mathrm{Re}_0 = 11.7 \cdot 10^6$.

The separation zone on the upper surface of the wing for this case is relatively small; the flow fully re-attaches before it reaches the trailing edge so that no phenomena associated with the separation are present in the wake. However, the simulation of such flows already requires more sophisticated turbulence models inside the computational domain; we have used the the two-equation Menter's model [53]. Moreover, it requires much finer grids in the near field than the simulation of the attached flows does.

As in the previous transonic case, the global ABCs (2.48) are constructed here on the basis of matrices (2.1c). The computations are conducted for two different domains of the "average radii" of 3 and 10 root chords of the wing, respectively. The grids for both domains in this case have dimensions $193 \times 49 \times 33$, i.e., we choose the same gridding strategy as for subsonic numerical experiments of sections 3.2.1 and 3.2.2. Of course, the actual grids here are not the same as those used previously; their normal cell size near the wing surface is an order of magnitude smaller. Similarly to the foregoing subsonic cases (sections 3.2.1 and 3.2.2) we expect for this case that the resolution of the near field for the smaller domain will be better than for the larger domain. This is particularly important because of the complicated flow structure that involves the shock-induced separation.

In Figure 3.3, we present the distribution of the pressure coefficient

$$C_p = \frac{p - p_0}{\frac{1}{2}\rho_0 {u_0}^2}$$

on the upper and lower surfaces of the wing in the cross section $z = \text{const}$ at 90% of semispan. The 90% of semispan station corresponds to the area of developed separation. The three solutions that we have computed in this case are for global DPM-based ABCs on the 3 and 10 root chords domains and standard ABCs on the 10 chords domain. These solutions are compared in Figure 3.3 against the experimental data.

From Figure 3.3 we conclude that all three numerical solutions match one another very well and also match the experimental data to a reasonable degree of accuracy. We also emphasize that, analogously to the previous cases, the DPM-based global ABCs (2.48) are capable of generating an accurate numerical solution on the small domain for this separated flow case. On the other hand, the standard algorithm for $\alpha = 5.06^\circ$ (separated flow around ONERA M6) fails to converge on the 3 root chords computational domain; the corresponding convergence history is presented in Figure 3.4(a). Comparing Figure 3.4(a) with Figure 3.1(a) we see that for the attached case $\alpha = 3.06^\circ$ (which is relatively easy) one could still observe a very slow convergence of the standard algorithm, whereas its convergence for the separated case $\alpha = 5.06^\circ$ is completely destroyed. At the same time, on the large (10 root chords)

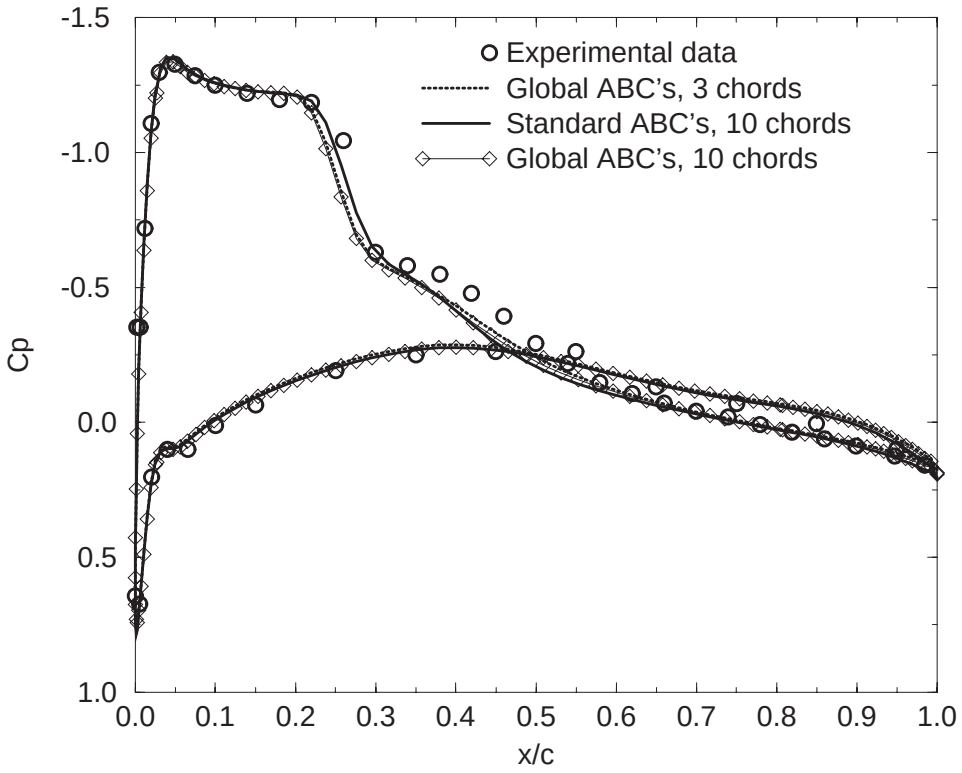

FIG. 3.3. *ONERA M6:* $M_0 = 0.84$, $\mathrm{Re}_0 = 11.7 \cdot 10^6$, $\alpha = 5.06^\circ$. *Surface pressure distribution at the* 90% *of semispan (x/c: x is the coordinate calculated from the leading edge, c is the local chord length). Dimensions of all grids are* $193 \times 49 \times 33$.

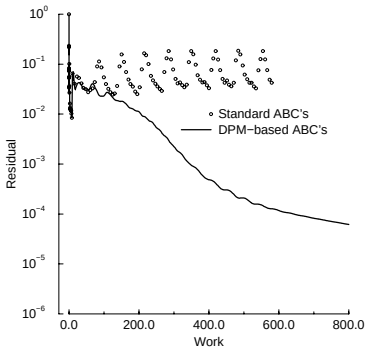

FIG. 3.4(a). *ONERA M6:* $M_0 = 0.84$, $Re_0 = 11.7 \cdot 10^6$, $\alpha = 5.06^\circ$. *Convergence history for the residual of the continuity equation. "Average radius" of* D_{in} *is* 3 *root chords of the wing; dimensions of the grid are* $193 \times 49 \times 33$.

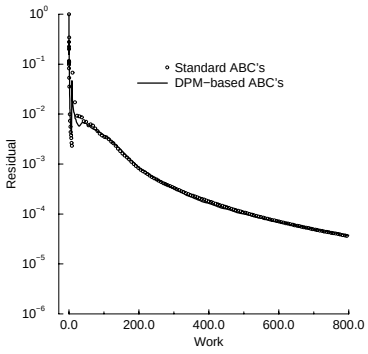

FIG. 3.4(b). *ONERA M6:* $M_0 = 0.84$, $\mathrm{Re}_0 = 11.7 \cdot 10^6$, $\alpha = 5.06^\circ$. *Convergence history for the residual of the continuity equation. "Average radius" of* D_{in} *is* 10 *root chords of the wing; dimensions of the grid are* $193 \times 49 \times 33$.

domain both the DPM-based and standard algorithms for $\alpha = 5.06^\circ$ converge at the same rate. See Figure 3.4(b).

Figures 3.3 and 3.4 allow us to conclude that nonlocal DPM-based ABCs (2.48) not only speed up the convergence of multigrid iterations but are generally capable of increasing robustness of the entire numerical procedure. Note that the issues of combined implementation of global DPM-based ABCs with multigrid and resulting speedup of multigrid convergence have been specifically addressed and emphasized in our recent paper [50].

3.3. Computational cost of the DPM-based ABCs. In all the three-dimensional computations described above, the DPM-based ABCs were implemented directly, without computing the matrix of operator $\boldsymbol{T}$ from (2.48). By applying the new procedure only on the first and last Runge–Kutta stages and only on the finest multigrid level, the total number of calculations of the generalized potential has been brought to a minimum. In so doing, the average cost of applying the DPM-based ABCs (2.48) adds about 20–25% of the CPU time to the cost of the same procedure with the standard (characteristics-based) boundary conditions. This extra expense is not high (taking into account the improvement of accuracy); moreover, it can often be compensated for and even noticeably prevailed over by the convergence acceleration and reduction of the domain size. Furthermore, to explicitly decrease the computational cost associated with the DPM-based ABCs we plan on the future use of the entry-wise interpolation of boundary operators (see [23]) and/or multiresolution-based methodologies (see [20, 23]). We expect that the latter can also be employed when implementing the DPM-based ABCs for multiblock grids.

Having addressed the actual computational cost of the DPM-based ABCs for the specific implementation described above, we would also like to make a few general comments on our choice of the interior solver (`TLNS3D`) as well as on the potential for combining the new ABCs methodology with other available CFD codes. The primary reason for choosing `TLNS3D` is that this code is, in fact, one of the "workhorses" extensively used nowadays for production flow computations in both research and industrial applications. The code is universal; it can calculate fluid flows around over complicated geometries for a wide variety of flow regimes. Although the chimera grid capability is not yet incorporated into `TLNS3D`, experimentally the code has been found quite robust on multiblock grids that consist of point-matched and patched grid block interfaces.

Concerning the performance, it is generally acknowledged that `TLNS3D` is not the fastest code. In particular, it is quite possible that the multigrid strategy employed in `TLNS3D` can be replaced by a much better algorithm (see below). However, other currently available production codes in CFD are not much faster than `TLNS3D`. For example, the NASA-developed code `PAB3D`, which does not use multigrid, is expected to be somewhat slower than `TLNS3D` for the computations that involve the reverse flow regions. Although this code has advanced capabilities for turbulence modeling (which is advantageous, e.g., for the jet mixing problems), it was not the issue for the class of problems studied in this paper. Another widely used NASA-developed solver, `CFL3D` (see [54]), which is based on upwind discretization (as opposed to central differences in `TLNS3D`) and does employ multigrid, may display a better performance compared to `TLNS3D` (the same is true for the code `OVERFLOW`, which is very similar to `CFL3D` in its basic concepts). However, the difference in performance is not that dramatic. In [55], Rumsey and Vatsa compare the results obtained with `TLNS3D` and `CFL3D` for the flow cases that are very close (some are actually the same) to

those investigated in this paper; they have found that the difference in CPU times was only about 11% to the advantage of CFL3D on a Cray Y-MP platform. All the computations presented in this paper were actually conducted on either a Cray Y-MP or another Cray Research computer, the J90 series, which is similar to the Y-MP in its architecture. Therefore, on these platforms, implementation of the DPM-based ABCs, even with a faster production solver, CFL3D, would not result in an overhead much higher than the 20–25% mentioned previously (this is well compensated for by the improvement of accuracy and the speedup of multigrid convergence).

It is also known that the actual performance of the code is, in fact, platform-dependent. For example, on a Cray C90 system, CFL3D may run about 1.5 times faster than TLNS3D, which is most likely due to the better use of vectorization. Our code for calculating the ABCs (2.48) can also take full advantage of the vector capabilities of the machine; therefore we do not think its performance would decrease if run on a C90 platform. At the same time, on the cache-based machines (powered, e.g., by an SGI R10000 processor) the performance of CFL3D and TLNS3D is essentially the same. Therefore, the overhead due to the new ABCs would be the same as well. Other flow solvers, e.g., ARC3D, may be better tuned for the cache-based machines and therefore run almost twice as fast as either TLNS3D or CFL3D, which only proves once again that the issue of program performance cannot be studied abstractly—it has to relate to a specific code implemented on a specific platform.

For the sake of completeness, however, we should say that all the aforementioned solvers (as well as all others that are used extensively for production computations nowadays) are based on what most likely can already be called the previous generation of concepts in CFD. A massive effort is currently underway toward developing much faster CFD codes with the potential of achieving the so-called "textbook multigrid efficiency." The corresponding approaches are based on the innovative ideas of genuinely multidimensional upwind discretizations and factorization of the flow system into the elliptic and advection subsystems with subsequent multigrid solution of the elliptic part and downstream marching of the advection part. In our view, these techniques, if successful, can revolutionize the entire area of CFD. However, at the current stage of their development they are nowhere close to solving the viscous flows of the type analyzed in this paper.

Therefore, the discussion on performance of the DPM-based ABCs, if combined with some solver other than TLNS3D, has to be restricted to the currently available production codes. As has been pointed out, all these codes basically demonstrate similar performance; recently, some of them (TLNS3D and CFL3D) have even become similar in their coding structures. Consequently, both implementation-wise and overhead-wise there are no serious reasons to expect any substantial difference if the DPM-based boundary conditions are combined with some other production solver. Implementation with TLNS3D alone already proves, in our view, the effectiveness of the new methodology for the fluid flow computations.

4. Conclusions. The new global ABCs for calculating steady-state external viscous flows in three space dimensions have been constructed on the basis of the difference potentials method (DPM). The approach generalizes and extends our previous two-dimensional results.

The new ABCs are capable of greatly reducing the size of the computational domain (compared to the standard methods) while still maintaining high accuracy of the numerical solution. This size reduction amounts to either the possibility of refining the grid in the near field, which potentially leads to increasing the accuracy, or usage of

the smaller-dimensional grids while keeping the accuracy at the same level. Moreover, the DPM-based ABCs may noticeably speed up the convergence of multigrid iterations and generally improve robustness of the entire numerical procedure. Finally, the new boundary conditions appear geometrically universal and easy to incorporate in the structure of the existing flow solvers. The properties of the new ABCs have been corroborated experimentally by computing subsonic and transonic flows past the ONERA M6 wing with the NASA-developed code `TLNS3D`, which is widely used for production flow computations in both research and industrial environments.

Acknowledgments. I am most grateful to V. N. Vatsa of NASA Langley Research Center for giving me an opportunity to use the code `TLNS3D` and for his diverse and extremely valuable help on every stage of this work. I am also very thankful to E. B. Parlette of Vigyan, Inc., for generating the grids used in the foregoing computations. Numerous discussions on the subject of this work with R. C. Swanson and J. L. Thomas of NASA Langley Research Center have undoubtedly contributed much to the improvement of the paper. Most helpful comments of the referees of the paper are gratefully acknowledged and appreciated.

REFERENCES

[1] S. V. Tsynkov, *Numerical solution of problems on unbounded domains. A review*, Appl. Numer. Math., 27 (1998), pp. 465–532.
[2] D. Givoli, *Non-reflecting boundary conditions*, J. Comput. Phys., 94 (1991), pp. 1–29.
[3] D. Givoli, *Numerical Methods for Problems in Infinite Domains*, Elsevier, Amsterdam, 1992.
[4] A. P. Calderón, *Boundary value problems for elliptic equations*, in Proceedings 1963 Joint Symposium Partial Differential Equations at Novosibirsk, Fizmatgiz, Moscow, 1963, Acad. Sci. USSR Siberian Branch, Moscow, 1963, pp. 303–304.
[5] R. T. Seeley, *Singular integrals and boundary value problems*, Amer. J. Math., 88 (1966), pp. 781–809.
[6] V. S. Ryaben'kii, *Boundary equations with projections*, Russian Math. Surveys, 40 (1985), pp. 147–183.
[7] V. S. Ryaben'kii, *Difference Potentials Method for Some Problems of Continuous Media Mechanics*, Nauka, Moscow, 1987 (in Russian).
[8] V. S. Ryaben'kii, *Difference potentials method and its applications,* Math. Nachr., 177 (1996), pp. 251–264.
[9] S. G. Mikhlin, N. F. Morozov, and M. V. Paukshto, *The Integral Equations of the Theory of Elasticity*, B. G. Teubner Verlagsgesellschaft, Stuttgart, 1995.
[10] R. C. Swanson and E. Turkel, *A Multistage Time-Stepping Scheme for the Navier-Stokes Equations*, AIAA paper 85-0035, AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, 1985.
[11] R. C. Swanson and E. Turkel, *Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations*, AIAA paper 87-1107-CP, AIAA 8th Computational Fluid Dynamics Conference, Honolulu, Hawaii, 1987.
[12] R. C. Swanson and E. Turkel, *Multistage Schemes with Multigrid for the Euler and Navier-Stokes Equations. Volume* I: *Components and Analysis,* NASA Technical paper 3631, NASA Langley Research Center, Hampton, VA, August 1997.
[13] V. S. Ryaben'kii and S. V. Tsynkov, *Artificial boundary conditions for the numerical solution of external viscous flow problems*, SIAM J. Numer. Anal., 32 (1995), pp. 1355–1389.
[14] S. V. Tsynkov, *An application of nonlocal external conditions to viscous flow computations*, J. Comput. Phys., 116 (1995), pp. 212–225.
[15] S. V. Tsynkov, E. Turkel, and S. Abarbanel, *External flow computations using global boundary conditions*, AIAA J., 34 (1996), pp. 700–706; also available as AIAA paper 95-0562, 33rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 1995.
[16] V. S. Ryaben'kii and S. V. Tsynkov, *An effective numerical technique for solving a special class of ordinary difference equations*, Appl. Numer. Math., 18 (1995), pp. 489–501.
[17] S. V. Tsynkov, *Artificial boundary conditions for computation of oscillating external flows*, SIAM J. Sci. Comput., 18 (1997), pp. 1612–1656; also available as NASA Technical Memorandum 4714, NASA Langley Research Center, Hampton, VA, August 1996.

[18] V. S. Ryaben'kii and S. V. Tsynkov, *An application of the difference potentials method to solving external problems in CFD*, in Computational Fluid Dynamics Review 1998, Vol. 1, M. Hafez and K. Oshima, eds., World Scientific, Singapore, 1998, pp. 169–205; also available as NASA Technical Memorandum 110338, NASA Langley Research Center, Hampton, VA, March 1997.

[19] S. V. Tsynkov, *Nonlocal artificial boundary conditions for computation of external viscous flows*, in Computational Fluid Dynamics '96, Proceedings of the Third ECCOMAS CFD Conference, September 9–13, 1996, Paris, France, J.-A. Desideri, C. Hirsch, P. Le Tallec, M. Pandolfi, and J. Périaux, eds., John Wiley & Sons, New York, 1996, pp. 512–518.

[20] S. V. Tsynkov, *Artificial boundary conditions for infinite-domain problems*, in Barriers and Challenges in Computational Fluid Dynamics, V. Venkatakrishnan, M. D. Salas, and S. R. Chakravarthy, eds., Kluwer Academic Publishers, Norwell, MA, 1998, pp. 119–138.

[21] V. N. Vatsa and B. Wedan, *Development of a multigrid code for* 3*-D Navier-Stokes equations and its application to a grid-refinement study*, Comput. Fluids, 18 (1990), pp. 391–403.

[22] V. N. Vatsa, M. D. Sanetrik, and E. B. Parlette, *Development of a Flexible and Efficient Multigrid-Based Multiblock Flow Solver,* AIAA Paper 93-0677, 31st AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 1993.

[23] S. V. Tsynkov and V. N. Vatsa, *An improved treatment of external boundary for three-dimensional flow computations*, AIAA J., 36 (1998), pp. 1998–2004; also available as AIAA paper 97–2074, in Proceedings of the 13th AIAA Computational Fluid Dynamics Conference, Part 2, Snowmass Village, Colorado, 1997, pp. 1139–1149.

[24] D. Anderson, J. Tannehill, and R. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corporation, New York, 1984.

[25] S. Abarbanel and D. Gottlieb, *Optimal Time Splitting for Two- and Three-Dimensional Navier-Stokes Equations with Mixed Derivatives*, J. Comput. Phys., 41 (1981), pp. 1–33.

[26] S. Abarbanel, P. Duth, and D. Gottlieb, *Splitting methods for low mach number Euler and Navier–Stokes equations*, Comput. & Fluids, 17 (1989), pp. 1–12.

[27] J. Guerra and B. Gustafsson, *A numerical method for incompressible and compressible flow problems with smooth solutions*, J. Comput. Phys., 63 (1986), pp. 377–397.

[28] B. Gustafsson and H. Stoor, *Navier–Stokes equations for almost incompressible flow*, SIAM J. Numer. Anal., 28 (1991), pp. 1523–1547.

[29] J.-Y. Sa and K. S. Chang, *Far-field stream function condition for two-dimensional incompressible flows*, J. Comput. Phys., 91 (1990), pp. 398–412.

[30] R. H. Burkhart, *Asymptotic expansion of the free-space Green's function for the discrete* 3*-D Poisson equation*, SIAM J. Sci. Comput., 18 (1997), pp.1142–1162.

[31] R. H. Burkhart, J. Bussoletti, F. T. Johnson, S. S. Samant, and D. P. Young, *Solution of the Discrete Free-Space* 3*-D Poisson Equation*, Boeing Computer Services Technical Report BCSTECH–94–015, Seattle, WA, April 1994.

[32] F. W. Wubs, J. W. Boerstoel, and A. J. Van der Wees, *Grid size reduction in flow calculations on infinite domains by higher-order far-field asymptotics in numerical boundary conditions*, J. Engrg. Math., 18 (1984), pp. 157–177.

[33] J. L. Thomas and M. D. Salas, *Far-Field Boundary Conditions for Transonic Lifting Solutions to the Euler Equations*, AIAA paper 85–0020, 23rd AIAA Aerospace Science Meeting and Exhibit, Reno, NV, January 1985.

[34] A. Bayliss and E. Turkel, *Radiation boundary conditions for wave-like equations,*, Comm. Pure Appl. Math., 33 (1980), pp. 707–725.

[35] A. Bayliss and E. Turkel, *Outflow boundary conditions for fluid dynamics*, SIAM J. Sci. Stat. Comput., 3 (1982), pp. 250–259.

[36] A. Bayliss and E. Turkel, *Far-field boundary conditions for compressible flows*, J. Comput. Phys., 48 (1982), pp. 182–199.

[37] A. Bayliss, M. Gunzburger, and E. Turkel, *Boundary conditions for the numerical solution of elliptic equations in exterior regions*, SIAM J. Appl. Math., 42 (1982), pp. 430–451.

[38] J. D. Cole and L. P. Cook, *Transonic Aerodynamics*, Elsevier, Amsterdam, 1986.

[39] M. Drela *Two-Dimensional Transonic Aerodynamic Design and Analysis Using the Euler Equations*, Massachusetts Institute of Technology, Gas Turbine Laboratory Report 187, Boston, MA, February 1986.

[40] M. B. Giles and M. Drela, *Two-dimensional transonic aerodynamic design method*, AIAA J., 25 (1987), pp. 1199–1206.

[41] G. S. S. Ludford, *The behavior at infinity of the potential function of a two-dimensional subsonic compressible flow*, J. Math. Phys., 30 (1951), pp. 117–130.

[42] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics*, Pergamon Press, Oxford, 1986.

[43] H. Ashley and M. T. Landahl, *Aerodynamics of Wings and Bodies*, Addison-Wesley, Read-

ing, MA, 1965.
[44] V. S. Vladimirov, *Equations of Mathematical Physics*, Marcel Dekker, New York, 1971.
[45] A. N. Kolmogorov and S. V. Fomin, *Elements of the Theory of Functions and Functional Analysis*, Nauka, Moscow, 1981 (in Russian).
[46] H. Schlichting, *Boundary Layer Theory*, McGraw-Hill, New York, 1968.
[47] V. S. Ryaben'kii, *Necessary and sufficient conditions for good definition of boundary value problems for systems of ordinary difference equations*, U.S.S.R. Comput. Math. and Math. Phys., 4 (1964), pp. 43–61.
[48] F. R. Gantmakher, *Applications of the Theory of Matrices*, Interscience Publishers, New York, 1959.
[49] S. K. Godunov and V. S. Ryaben'kii, *Canonical forms of systems of ordinary linear difference equations with constant coefficients*, Comput. Math. Math. Phys., 3 (1963), pp. 281–295.
[50] S. V. Tsynkov, *On the combined implementation of global boundary conditions with central-difference multigrid flow solvers*, in IUTAM Symposium on Computational Methods for Unbounded Domains, July 27 – 31, 1997, University of Colorado at Boulder, T. L. Geers, ed., Kluwer Academic Publishers, Dordrecht, 1998, pp. 285–294.
[51] E. Turkel, V. N. Vatsa, and R. Radespiel, *Preconditioning Methods for Low-Speed Flows*, AIAA paper 96–2460–CP, 14th AIAA Applied Aerodynamics Conference, New Orleans, LA, June 1996.
[52] V. Schmitt and F. Charpin, *Pressure distributions on the ONERA M6 wing at transonic mach numbers*, AGARD AR–138, 1979, p. B1.
[53] F. R. Menter, *Performance of Popular Turbulence Models for Attached and Separated Adverse Pressure Gradient Flows*, AIAA paper 91–1784, AIAA 22nd Fluid Dynamics, Plasma Dynamics & Lasers Conference, Honolulu, Hawaii, June 1991.
[54] J. L. Thomas, S. Krist, and W. K. Anderson, *Navier-Stokes computations of vortical flows over low-aspect-ratio wings*, AIAA J., 28 (1990), pp. 205–212.
[55] C. L. Rumsey and V. N. Vatsa, *Comparison of Predictive Capabilities of Several Turbulence Models*, AIAA J. Aircraft, 32 (1995), pp. 510–514.