JOURNAL OF COMPUTATIONAL PHYSICS 46, 1–14 (1982)

Direct Solution of the Discretized Poisson–Neumann Problem on a Domain Composed of Rectangles

U. SCHUMANN AND J. BENNER

Kernforschungszentrum Karlsruhe,
Institut für Reaktorentwicklung, Projekt Nukleare Sicherheit,
Postfach 3640, D-7500 Karlsruhe, Federal Republic of Germany

Received September 11, 1981

Finite-difference approximations for the two-dimensional Poisson–Neumann problem

$$(1/\lambda)\,\mathrm{div}\,\lambda\,\mathrm{grad}\,p = q, \qquad \lambda > 0 \quad \text{on} \quad R$$

$$\lambda \mathbf{n} \cdot \mathrm{grad}\,p = g_R \qquad \text{on} \quad \partial R$$

result in a large system of linear equations $\mathbf{Lp} = \mathbf{q}$. The $n \times n$ matrix $\mathbf{L}$ is singular with $\mathrm{rank}(\mathbf{L}) = n - 1$. A solution exists if $\mathbf{q}$ satisfies the consistency condition $\mathbf{v}^T\mathbf{q} = 0$, where $\mathbf{L}^T\mathbf{v} = \mathbf{0}$, $\mathbf{v} \neq 0$. A direct-solution scheme is described for the case where R can be decomposed into a set of rectangular domains each having a possibly different but constant material coefficient λ. We assume that this problem has to be solved repeatedly for many vectors $\mathbf{q}$. The method is efficient in that the number of operations is of order $n \log n$. This efficiency is gained by using fast elliptic solvers for each single rectangle and a proper variant of the so-called *influence-* or *capacitance*-matrix technique. Here, on each rectangle a separate Poisson–Neumann problem has to be solved. The essential point is the manner in which the consistency condition is enforced for each rectangle. The new method has been successfully applied in a code FLUX to analyze fluid-structure interactions.

1. INTRODUCTION

1.1 *The Poisson–Neumann Problem*

A considerable number of problems in mathematical physics are of the "Poisson–Neumann" type, i.e., they require the solution of Poisson's equation

$$(1/\lambda)\,\mathrm{div}\,\lambda\,\mathrm{grad}\,p = q, \qquad \lambda > 0 \quad \text{on} \quad R \tag{1a}$$

with Neumann boundary conditions

$$\lambda \mathbf{n} \cdot \mathrm{grad}\,p = g_R \qquad \text{on} \quad \partial R. \tag{1b}$$

Here, λ is a given space-dependent material coefficient and $\mathbf{n}$ is the unit-outward-normal vector on ∂R. Examples are diffusion problems with prescribed flux at the boundaries and flow problems in closed domains where p represents the velocity

0021-9991/82/040001-14$02.00/0

potential or the pressure field. Often such situations require the repeated solution of (1) for a large set of different "sources" q and g_R so that efficient solution schemes are necessary.

As can be seen from Gauss' theorem, q and g_R have to satisfy the consistency condition

$$\iint_R \lambda q \, dV - \oint_{\partial R} g_R \, dS = 0 \tag{2}$$

as a prerequisite for existence of a solution. The solution is nonunique, i.e., if p' is a solution of (1) and a an arbitrary constant, then

$$p = p' + a \tag{3}$$

is a solution as well. It is this lack of uniqueness which makes a numerical solution difficult.

1.2 *The Discretized Problem*

A discrete approximation to (1), using finite differences, e.g., results in a large linear system of equations

$$\mathbf{Lp} = \mathbf{q}. \tag{4}$$

The $n \times n$ matrix $\mathbf{L}$ is singular ($\det \mathbf{L} = 0$) with defect one, i.e., $\operatorname{rank}(\mathbf{L}) = n - 1$. This means that there exists one zero eigenvalue and corresponding eigenvectors $\mathbf{u} \neq 0$ and $\mathbf{v} \neq 0$ such that

$$\mathbf{Lu} = 0, \qquad \mathbf{L}^T\mathbf{v} = \mathbf{0}. \tag{5}$$

The discrete consistency condition corresponding to (2) is [1]

$$\mathbf{v}^T \cdot \mathbf{q} = 0. \tag{6}$$

If (6) is satisfied, then nonunique solutions exist, i.e., if $\mathbf{Lp}' = \mathbf{q}$, then

$$\mathbf{p} = \mathbf{p}' + a\mathbf{u} \tag{7}$$

is the general solution were a is an arbitrary scalar value.

1.3 *Fast Elliptic Solvers for Poisson–Neumann Problems on a Rectangle*

If the domain R is a simple rectangle, if $\lambda = \text{const}$, and if special regular finite-difference approximations are employed, then (4) can be solved directly in an order n log n operations by one of several fast elliptic solvers developed in recent years [2–5]. Finite-difference approximations which are based on a staggered grid are particularly suited to Poisson–Neumann problems (Fig. 1). On such grids the discrete solution values are solely specified on the interior of the domain. The resultant coefficient

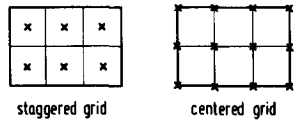

FIG. 1. Staggered and nonstaggered grids.

matrix can be written in a symmetric form. Staggered grids are very often used for the pressure in fluid-flow problems. There also exist fast elliptic solvers for nonstaggered grids [4, 5] but these will not be considered in this paper.

As **L** is singular, any solution scheme instead of (4) solves a system

$$\mathbf{D}\mathbf{p}' = \mathbf{q}, \qquad \det(\mathbf{D}) \neq 0, \tag{8}$$

where at least one row of **D** differs from the corresponding row in **L**. The differences between **L** and **D** are such that $\det(\mathbf{D}) \neq 0$ and such that for all **q** with $\mathbf{v}^T\mathbf{q} = 0$, one has $\mathbf{q} = \mathbf{L}\mathbf{D}^{-1}\mathbf{q}$, i.e., a solution of (8) is also a special solution of (4).

In the fast elliptic solvers, the change from **L** to **D** is made inside the solution algorithm. Typically, this change is made at a stage of the solution procedure where it comes to the final equation which, for the matrix **L**, would imply division by a zero-matrix element. Then this element is altered into a nonzero value. So, only one element is changed explicitly. Because of the transformations involved in the solution process, however, this usually implies that **D** differs from **L** in more than one row. Therefore, without studying the details of the fast elliptic solver, one does not know how many and which rows are altered. For the application to be described below, it is essential that we can allow **D** to differ from **L** in more than one row and do not require the explicit knowledge of the differences.

1.4 *Problem Formulation for a Domain Composed of Rectangles*

We shall describe a solution scheme which solves a discrete approximation of (1) for a domain R which is composed of several rectangular domains R_i, $i = 1, 2, \ldots, d$, with $\lambda = \lambda_i = \text{const}$ within each rectangle (Fig. 2). At the interfaces between the domains, λ varies in a stepwise manner but the normal gradients (physically, the fluxes) $g = \lambda \mathbf{n} \cdot \operatorname{grad} p$ and the solutions p are continuous (here, the sign convention

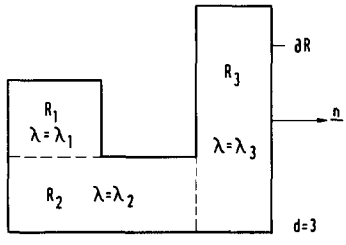

FIG. 2. Example of the domain $R = \sum_{i=1}^{d} R_i$.

for $\mathbf{n}$ is fixed for each interface). By introduction of g as a new unknown we can separate (1) into a number of Poisson–Neumann problems for each single domain R_i.

After finite-difference approximation the set of Poisson–Neumann problems can be written in the form

$$\mathbf{L}_i\mathbf{p}_i + \mathbf{G}_i\mathbf{g} = \mathbf{q}_i, \qquad i = 1, 2,..., d, \tag{9a}$$

$$\mathbf{g} + \sum_{j=1}^{d} \mathbf{H}_j\mathbf{p}_j = \mathbf{0}. \tag{9b}$$

For a staggered grid, $\mathbf{p}_i$ and $\mathbf{q}_i$ are the subvectors of $\mathbf{p}$ and $\mathbf{q}$ used in (4) corresponding to the d domains R_i with a total of n components in $\mathbf{p}$. The vector $\mathbf{g}$ of length k, say, includes the discrete values of the interface gradients g. Strictly, we have $d - 1 \leqslant k$, but we assume $d \ll k = O(n^{1/2})$.

If $\mathbf{L} = [\mathbf{L}_{i,j}]$ is partitioned like $\mathbf{p}$, then a comparison of (4) with (9) gives the relation

$$\mathbf{L}_{i,j} = \delta_{i,j}\mathbf{L}_i - \mathbf{G}_i\mathbf{H}_j; \qquad i,j = 1, 2,..., d. \tag{10}$$

Due to the origin from a set of Poisson–Neumann problems, each matrix $\mathbf{L}_i$ in (9a) is singular with defect one. Thus,

$$\mathbf{L}_i\mathbf{u}_i = 0, \qquad \mathbf{L}_i^{\mathrm{T}}v_i = \mathbf{0}, \qquad \mathbf{u}_i \neq 0, \quad \mathbf{v}_i \neq 0. \tag{11}$$

From (10) we see that $\mathbf{u}_i, \mathbf{v}_i$ are the subvectors of $\mathbf{u}$ and $\mathbf{v}$ if

$$\mathbf{H}_i\mathbf{u}_i = \mathbf{0}, \qquad i = 1, 2,..., d, \tag{12a}$$

$$\sum_{j=1}^{d} \mathbf{v}_j^{\mathrm{T}}\mathbf{G}_j\mathbf{g}' = 0 \tag{12b}$$

for an arbitrary k-vector $\mathbf{g}'$. In Section 4 we shall describe an example for which $\mathbf{G}_i$, $\mathbf{H}_i$, and $\mathbf{L}_i$ can be defined explicitly satisfying (9)–(12). Because of (11) we have a set of d consistency conditions

$$\mathbf{v}_i^{\mathrm{T}}(\mathbf{q}_i - \mathbf{G}_i\mathbf{g}) = 0, \qquad i = 1, 2,..., d. \tag{13}$$

In view of (6) and (12b) these are only $d - 1$ independent conditions. Because of the assumption of constant λ_i, fast elliptic solvers are applicable to solve (9a) if $\mathbf{q}_i$ and $\mathbf{g}$ satisfying (13) are given. As explained in Section 1.3 such solvers, in fact, solve

$$\mathbf{D}_i\mathbf{p}_i' = \mathbf{q}_i - \mathbf{G}_i\mathbf{g}, \qquad i = 1, 2,..., d, \tag{14}$$

and the general solutions are

$$\mathbf{p}_i = \mathbf{p}_i' + a_i\mathbf{u}_i. \tag{15}$$

For $\mathbf{p}_i$, $\mathbf{g}$ being solutions of (9a) and (9b), only one of the d scalars a_i can be taken

arbitrarily. The other $d-1$ coefficients have to be determined such that (9) and (13) are satisfied. By fixing one scalar, e.g., $a_d = 0$, system (9b), (13)–(15) provides a unique solution, i.e., it represents a nonsingular system.

1.5 *Overview of the Solution Algorithm*

We shall now give a simplified outline of the algorithm which should guide the reader to understand the formal description which is given in the subsequent sections.

The solution is obtained in a two-sweep algorithm. In a first sweep we start with an arbitrary estimate $\bar{\mathbf{g}}$ for the unknown $\mathbf{g}$ and solve (14) for $\mathbf{p}_i'$ using a fast elliptic solver for each rectangular domain R_i, $i = 1, 2, \ldots, d$. Further, we assume arbitrary values for the a_i. After this sweep we have preliminary solutions which satisfy most equations of system (9) which we wish to solve. We have, however, a nonzero residuum in the k equations (9b) and in at least one row of each of the d equations (9a), because our estimate $\bar{\mathbf{g}}$ will generally not satisfy consistency conditions (13). By means of the known *capacitance* [6] or influence-matrix technique [7] (summarized in Section 2), we can then find correct values for $\mathbf{g}$ and the a_i such that all Eqs. (9) are satisfied after this sweep. This procedure, however, would require that we know in advance which rows of (9a) result in a nonzero residuum and there must be at most d such rows. This is not the case and, therefore, we have to use an algorithm which is a bit more complicated.

Some additional unknowns b_i and equations are introduced which guarantee that after the first sweep the residuum will appear in a well-defined set of rows. In the first sweep the $d-1$ coefficients b_i are chosen such that the vector $\mathbf{g}'$ which results from the mapping

$$\mathbf{g}' = \bar{\mathbf{g}} + \sum_{i=1}^{d-1} b_i \mathbf{e}_i$$

(with linearly independent vectors $\mathbf{e}_i$ still to be defined) satisfies consistency conditions (13). Using this estimate for $\mathbf{g}$, the solutions of (14) and (15) will satisfy all rows of (9a). A nonzero residuum will still appear from (9b) and from the additional equations $b_i = 0$, $i = 1, 2, \ldots, d-1$. Now, the influence-matrix technique can be used to get correct values of $\mathbf{g}$, $b_i = 0$, and $\mathbf{p}_i$ after the second sweep. The reader will note that such a two-sweep procedure is a direct and not an iterative solution scheme.

An alternative to this method would be to formulate (9) such that the $\mathbf{L}_i$ correspond to the Dirichlet problems inside the rectangles R_i while taking the solution values on *one* side of the interface boundaries as the additional unknowns. This was the procedure used in [7]. It requires, however, $\lambda = \text{const}$ for *all* domains; otherwise the change in λ at the interfaces causes a structure of the coefficient matrix $\mathbf{L}_i$ for which the fast elliptic solvers cannot be applied. The need to get a solution scheme which allows for different values of λ_i initiated the present study. Dirichlet solvers could be applied if the solution values on *both* sides of the interface are treated by means of the influence-matrix technique. But this would require an

influence matrix for $2k$ components instead of $k + d - 1$ as in the present proposal and $d \ll k$ usually. The size m of the influence matrix should be small for efficiency ($m^2 \lesssim n$).

Another alternative would be to follow the variant of the influence-matrix technique described by Buzbee *et al.* [6] for singular matrices. This proposal, however, is not directly applicable because in our case the rank of (9a) is $n - d$ instead of $n - 1$ as assumed by Buzbee *et al.* Further, one can show that the algorithm proposed in [6] for such singular case requires more computations or more storage than the solution scheme described below.

We shall first repeat the essentials of the influence-matrix technique as required for our purpose. Then, we shall formulate the mapping according to the consistency conditions and the practical solution sequence. Finally, an application of this new algorithm will be described which should illustrate the usefulness of the procedure.

2. The Influence-Matrix Technique

The influence-matrix technique [6] provides a solution to any "A problem"

$$\mathbf{A}\mathbf{x} = \mathbf{y}, \qquad \det(\mathbf{A}) \neq 0 \tag{16}$$

by two solutions of a "B problem" $\mathbf{B}\bar{\mathbf{x}} = \bar{\mathbf{y}}$. The $n \times n$ matrix $\mathbf{B}$ is obtained by altering m rows of $\mathbf{A}$ such that $\mathbf{B}$ can be solved with existing fast solvers. The method is effective if $m^2 \lesssim n$ and if (16) has to be solved repeatedly for many different vectors $\mathbf{y}$. Formally, the influence-matrix technique can be described as follows:

Partition $\mathbf{A}$ in the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}, \tag{17}$$

where $\mathbf{A}_1$ is an $m \times n$ matrix and $\mathbf{A}_2$ an $(n - m) \times n$ matrix. The matrix $\mathbf{A}_1$ corresponds to the "irregular" equations which prohibit application of a fast solver directly to the A problem. The matrix $\mathbf{B}$ is

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{A}_2 \end{bmatrix}, \tag{18}$$

where $\mathbf{B}_1$ is taken as appropriate for the fast solver. We require $\det(\mathbf{B}) \neq 0$. In the same way as $\mathbf{A}$, we partition

$$\mathbf{y} = \begin{Bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{Bmatrix} \tag{19}$$

and define $\bar{\mathbf{y}}$ to be any vector of the form

$$\bar{\mathbf{y}} = \begin{Bmatrix} \bar{\mathbf{y}}_1 \\ \mathbf{y}_2 \end{Bmatrix}. \tag{20}$$

Let $\mathbf{W}_1$ be any nonsingular $m \times m$ matrix (usually the unit matrix) and define the $n \times m$ matrix $\mathbf{W}$ by

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 \\ 0 \end{bmatrix}. \tag{21}$$

Then the $m \times m$ matrix

$$\mathbf{C} = \mathbf{A}_1 \mathbf{B}^{-1} \mathbf{W} \tag{22}$$

is the so-called influence or *capacitance* matrix. It can be precomputed and decomposed in lower and upper triangular matrices in an order $O(mn \log n) + O(m^3)$ operations during a preparational step of the solution procedure. As shown in [6]

$$\det \mathbf{C} = (\det \mathbf{A})(\det \mathbf{W}_1)/\det(\mathbf{B}) \neq 0. \tag{23}$$

Once $\mathbf{C}$ is known we obtain the solution of the A problem (16) for each vector $\mathbf{y}$ by solving

$$\mathbf{B}\bar{\mathbf{x}} = \bar{\mathbf{y}}, \tag{24a}$$

$$\mathbf{C}\mathbf{w} = \mathbf{y}_1 - \mathbf{A}_1 \bar{\mathbf{x}}, \tag{24b}$$

$$\mathbf{B}\mathbf{x}^* = \bar{\mathbf{y}} + \mathbf{W}\mathbf{w}. \tag{24c}$$

Because of (18), (20), (21), and (24c) the result $\mathbf{x}^*$ satisfies $\mathbf{A}_2 \mathbf{x}^* = \mathbf{y}_2$. Further, one can easily prove that $\mathbf{A}_1 \mathbf{x}^* = \mathbf{y}_1$ by back substituting the given generic equations. Thus, $\mathbf{A}\mathbf{x}^* = \mathbf{y}$ and $\mathbf{x}^* = \mathbf{x}$ is the required solution.

3. The Solution Procedure

3.1 *Consistency Mapping*

For any given "source" vector $\mathbf{q}_i$, $i = 1, 2, \ldots, d$, and any "gradient" vector $\bar{\mathbf{g}}$ one can construct a vector $\mathbf{g}'$ which satisfies consistency conditions (13). For this purpose we use

$$\mathbf{g}' = \mathbf{g} + \sum_{j=1}^{d-1} b_j \mathbf{e}_j \tag{25}$$

and determine the $(d-1)$ coefficients b_j such that according to (13)

$$\mathbf{v}_i^{\mathrm{T}}(\mathbf{G}_i \mathbf{g}' - \mathbf{q}_i) = 0, \qquad i = 1, 2, \ldots, d. \tag{26}$$

Because of (6) and (12b) these are only $d-1$ independent conditions and this is the

reason why $d-1$ coefficients b_j are introduced. Because of (25) and (26), the b_j have to satisfy

$$\sum_{j=1}^{d-1} M_{i,j} b_j = \mathbf{v}_i^{\mathrm{T}}(\mathbf{q}_i - \mathbf{G}_i \mathbf{g}), \qquad i = 1, 2, \ldots, d-1, \tag{27a}$$

where the $(d-1) \times (d-1)$ matrix $[M_{i,j}]$ is defined by

$$M_{i,j} \equiv \mathbf{v}_i^{\mathrm{T}} \mathbf{G}_i \mathbf{e}_j.$$

The vectors $\mathbf{e}_j$ can be chosen arbitrarily except for the condition $\det(M_{i,j}) \neq 0$. As $\mathbf{e}_j$ is introduced in (25) and (26) in order to shift contributions to the consistency sums from one domain to the other, the vectors $\mathbf{e}_j$ should have many nonzero elements. Of course, they have to be linearly independent. In practice it is simplest to construct these vectors from random numbers. If the first set of such generated vectors should result in a singular or quasi-singular matrix $[M_{i,j}]$, then the next set of random numbers will usually be satisfactory. The amount of work required to solve (27a) will be negligible in comparison to the overall work because $d \ll n$.

3.2 *Formulation of the A Problem*

We now reformulate our original problem (9) in a form to which one can directly apply the influence-matrix technique. For this purpose the quantities b_j are artificially introduced as unknowns although we know that $b_j = 0$ in the final solution

$$\mathbf{A}_1 \mathbf{x} = \mathbf{y}_1 \quad \leftrightarrow \quad \mathbf{g} + \sum_{j=1}^{d} \mathbf{H}_j \mathbf{p}_j = \mathbf{0}, \tag{28a}$$

$$\leftrightarrow \quad b_i = 0, \qquad i = 1, 2, \ldots, d-1. \tag{28b}$$

$$\mathbf{A}_2 \mathbf{x} = \mathbf{y}_2 \quad \leftrightarrow \quad \sum_{j=1}^{d-1} M_{i,j} b_j + \mathbf{v}_i^{\mathrm{T}} \mathbf{G}_i \mathbf{g} = \mathbf{v}_i^{\mathrm{T}} \mathbf{q}_i, \quad i = 1, 2, \ldots, d-1, \tag{29a}$$

$$\leftrightarrow \quad \mathbf{D}_i \mathbf{p}_i' + \mathbf{G}\left[\mathbf{g} + \sum_{j=1}^{d-1} \mathbf{e}_j b_j\right] = \mathbf{q}_i, \quad i = 1, 2, \ldots, d, \tag{29b}$$

$$\leftrightarrow \quad \mathbf{p}_i - (\mathbf{p}_i' + a_i \mathbf{u}_i) = 0, \qquad i = 1, 2, \ldots, d-1, \tag{29c}$$

$$\leftrightarrow \quad \mathbf{p}_d - \mathbf{p}_d' = 0. \tag{29d}$$

One easily verifies that the number of equations is $k + 2(d-1) + 2n$ and equals the number of unknowns $\mathbf{g}$, $\{b_i\}$, $\{a_i\}$, $\{\mathbf{p}_i\}$, and $\{\mathbf{p}_i'\}$. Because of the trivial solution $b_i = 0$, this system is equivalent to that in (9), (13)–(15). The matrix $\mathbf{A}$ composed as in (17) by $\mathbf{A}_1$ and $\mathbf{A}_2$ defined in (28) and (29) is nonsingular. It delivers a special solution of (9)–(15) with $a_d = 0$.

3.3 *Formulation of B Problem and Solution*

The *B* Problem is composed of the submatrices $\mathbf{B}_1$ and $\mathbf{A}_2$, see (29) and (18). It is easy to define a suitable $\mathbf{B}_1$ which guarantees $\det(\mathbf{B}) \neq 0$ and a fast solvable set of equations

$$\mathbf{B}_1 \bar{\mathbf{x}} = \bar{\mathbf{y}}_1, \qquad \bar{\mathbf{y}}_1^{\mathrm{T}} = (\boldsymbol{\gamma}^{\mathrm{T}}, \alpha_1, \ldots, \alpha_{d-1}) \quad \leftrightarrow \quad \bar{\mathbf{g}} = \boldsymbol{\gamma}, \tag{30a}$$

$$\leftrightarrow \quad \bar{\mathbf{a}}_i = \alpha_i, \qquad i = 1, 2, \ldots, d-1. \tag{30b}$$

Thus, the number m of irregular equations to be treated by the influence-matrix technique is

$$m = k + d - 1.$$

A solution $\bar{\mathbf{x}}$ of the *B* Problem is obtained by successively solving (30) and (29) for any $\bar{\mathbf{y}}_1$ and the given $\mathbf{y}_2$. It is convenient to set $\bar{\mathbf{y}}_1 = \mathbf{y}_1 = \mathbf{0}$. By construction, systems (29b) satisfy consistency condition (13). Therefore, after the first sweep the solution $\bar{\mathbf{x}}$ satisfies $\mathbf{A}_2 \bar{\mathbf{x}} = \mathbf{y}_2$ and (9a), as required. The m equations $\mathbf{A}_1 \bar{\mathbf{x}} = \mathbf{y}_1$ are not satisfied. This we correct in the second sweep with the influence matrix as described in Section 2. As $d \ll n$, we have $m \approx k = O(n^{1/2})$. Thus, $m^2 \lesssim n$ as required for efficiency.

4. Application

4.1 *The Code* FLUX

The above method has been implemented in a new version of the code FLUX described earlier [7, 8]. This code simulates coupled fluid-structure motions in a three-dimensional model of a pressurized water reactor after breaking one of the coolant inlet pipes. In time, an implicit finite-difference scheme is used so that for every time step a three-dimensional elliptic equation has to be solved. For the axisymmetric vessel, the three-dimensional problem is separated into a set of two-dimensional ones by means of an azimuthal Fourier transform. For the zeroth Fourier mode the singular Poisson–Neumann problem arises if the fluid is modeled by incompressible potential flow and if the vessel has closed walls or prescribed outflow everywhere. In this case p corresponds to the fluid pressure, λ to the inverse fluid density ρ, and g_R to the pressure gradients at the vessel walls and outflow boundaries or at internal structure walls (like the core barrel, see Fig. 3). These wall gradients are nonzero because of nonzero wall accelerations. In cylindrical coordinates (r, z) the differential equation to be solved for $p\ (r, z)$ is

$$\frac{\rho}{r}\frac{\partial}{\partial r}\left[\frac{r}{\rho}\frac{\partial p}{\partial r}\right] + \rho\frac{\partial}{\partial z}\left[\frac{1}{\rho}\frac{\partial p}{\partial z}\right] = q(r, z). \tag{31}$$

Subsequently, we shall give the finite-difference approximations which define the

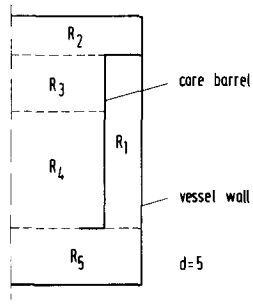

FIG. 3. Domain partitioning as used in the reactor pressure vessel model FLUX.

matrix equation $\mathbf{Lp} = \mathbf{q}$ considered in this paper. Thereafter, we shall identify the eigenvectors $\mathbf{u}$ and $\mathbf{v}$ for this example as required for the solution algorithm.

4.2 *The Finite-Difference Equations*

The finite-difference grid shown in Fig. 4 is characterized by the coordinates of the cell boundaries $r_{i+1/2}$, $z_{j+1/2}$, $i = 0, 1,..., I$, $j = 0, 1,..., J$. From these coordinates the mesh cell midpoints (r_i, z_j) and mesh spacings are determined,

$$r_0 = 0,$$

$$r_i = (r_{i+1/2} + r_{i-1/2})/2, \qquad z_j = (z_{j+1/2} + z_{j-1/2})/2,$$

$$\Delta r_i = r_{i+1/2} - r_{i-1/2}, \qquad \Delta z_j = z_{j+1/2} - z_{j-1/2},$$

$$i = 1, 2,..., I, \qquad j = 1, 2,..., J,$$

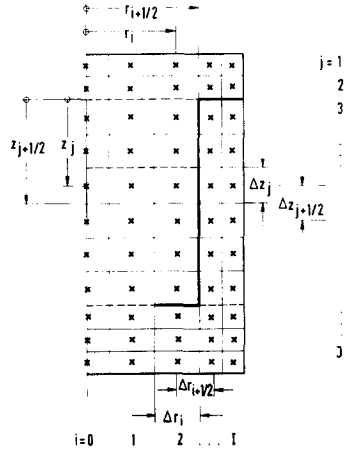

FIG. 4. The finite-difference grid of the FLUX model.

and

$$\Delta r_{i+1/2} = r_{i+1} - r_i, \qquad \Delta z_{j+1/2} = z_{j+1} - z_j,$$
$$i = 0, 1, \ldots, I-1, \quad j = 1, 2, \ldots, J-1.$$

The mesh-cell midvalues or cell averages

$$p_{ij} \approx p(r_i, z_j), \qquad q_{ij} = q(r_i, z_j)$$

form the components of the vector $\mathbf{p} = \{p_{ij}\}$ and an intermediate vector $\mathbf{q}' = \{q_{ij}\}$. Let

$$g^r_{(i+1/2)j} \approx \frac{1}{\rho}\frac{\partial p}{\partial r}(r_{i+1/2}, z_j)$$

$$g^z_{i(j+1/2)} \approx \frac{1}{\rho}\frac{\partial p}{\partial z}(r_i, z_{j+1/2})$$

represent the average gradients at the cell boundaries and

$$\rho_{ij} = \rho(r_i, z_j).$$

Then by Gauss' theorem one obtains from (31), for $i > 0$,

$$\frac{\rho_{ij}}{r_i \Delta r_i}\{r_{i+1/2}\, g^r_{(i+1/2)j} - r_{i-1/2}\, g^r_{(i-1/2)j}\}$$
$$+ (\rho_{ij}/\Delta z_j)\{g^z_{i(j+1/2)} - g^z_{i(j-1/2)}\} = q_{ij}; \tag{32a}$$

for $i = 0$,

$$\frac{2\rho_{0j}}{r^2_{1/2}}\{r_{1/2}\, g_{(1/2)j}\} + (\rho_{0j}/\Delta z_j)\{g^z_{i(j+1/2)} - g^z_{i(j-1/2)}\} = q_{0j}. \tag{32b}$$

At walls, i.e., at the domain ∂R, either g^r or g^z is given by the boundary value g_R. For mesh cells adjacent to such walls these gradients are subtracted from (32) and result in a modification of the vector $\mathbf{q}'$ which then defines $\mathbf{q}$. Thus, from now on we assume that the gradients g^r, g^z are zero at mesh boundaries coinciding with ∂R. For the internal-mesh boundaries, finite-difference approximations are introduced,

$$g^r_{(i+1/2)j} = \frac{2}{(\rho_{ij} + \rho_{(i+1)j})\Delta r_{i+1/2}}(p_{(i+1)j} - p_{ij}), \tag{33a}$$

$$g^z_{i(j+1/2)} = \frac{2}{(\rho_{ij} + \rho_{i(j+1)})\Delta z_{j+1/2}}(p_{i(j+1)} - p_{ij}). \tag{33b}$$

Equations (32) and (33) together for all mesh cells (ij) define the matrix problem (4) or (9). Equation (9) contains the matrices $\mathbf{L}_m$, $\mathbf{G}_m$, $\mathbf{H}_m$, $m = 1, 2,..., d$. None of these (sparse) matrices is stored explicitly; rather, they are implemented algorithmically. In fact, the matrices $\mathbf{L}_m$ (or $\mathbf{D}_m$, see (14)) are defined intrinsically in the fast elliptic solver, the only input being the grid parameters and densities. In order to specify $\mathbf{G}_m$ and $\mathbf{H}_m$, a list of indices

$$\{i_l, j_l^+, j_l^-;\ l = 1, 2,..., k\}$$

is set up, where (i_l, j_l^-) and (i_l, j_l^+) are the mesh cells below and above (with respect to the z axis) a domain interface (Fig. 5). (In this applications all interfaces are parallel to the radial coordinate so that one radial index suffices.) Then

$$\mathbf{g} = -\sum_{m=1}^{d} \mathbf{H}_m \mathbf{p}_m, \qquad \mathbf{g} = \{g_l\}$$

is equivalent to

$$g_l = (p_{\alpha\beta^+} - p_{\alpha\beta^-})2/[(\rho_{\alpha\beta^-} + \rho_{\alpha\beta^+})\Delta z_{\beta^- + 1/2}] \tag{34}$$

and

$$\mathbf{q}_m = \mathbf{q}'_m + \mathbf{G}_m \mathbf{g}, \qquad m = 1, 2,..., d,$$

is equivalent to

$$q_{\alpha\beta^\pm} = q'_{\alpha\beta^\pm} \pm g_l \rho_{\alpha\beta^\pm}/\Delta z_{\beta^\pm}, \tag{35}$$

with

$$\alpha \equiv i_l, \qquad \beta^- \equiv j_l^-, \qquad \beta^+ \equiv j_l^+, \qquad l = 1, 2,..., k.$$

4.3 *Eigenvectors* **u** *and* **v**

We note that **L** is a nonsymmetric matrix. However, **L** can be made symmetric. Let

$$\begin{aligned} v_{ij} &= \Delta z_j r_{1/2}^2/(2\rho_{0j}) \qquad \text{for} \quad i = 0, \\ &= \Delta z_j r_i \Delta r_i/\rho_{ij} \qquad \text{for} \quad i > 0, \end{aligned} \tag{36}$$

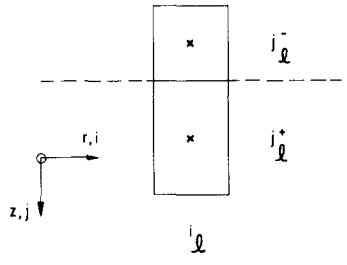

FIG. 5. Cells at a domain interface.

and

$$\mathbf{V} = \mathrm{diag}(v_{ij}), \qquad \mathbf{V}^{\mathrm{T}} = \mathbf{V}, \tag{37}$$

then $\mathbf{VL}$ is symmetric, i.e.,

$$\mathbf{VL} = (\mathbf{VL})^{\mathrm{T}} = \mathbf{L}^{\mathrm{T}}\mathbf{V}. \tag{38}$$

One easily verifies that

$$\mathbf{u} = \{u_{ij}\}, \qquad u_{ij} = 1 \tag{39}$$

is the eigenvector which satisfies $\mathbf{Lu} = \mathbf{0}$ because the gradients g^r, g^z are zero for $p_{ij} = u_{ij}$, see (33). From (38) and (39) $\mathbf{L}^{\mathrm{T}}\mathbf{v} = \mathbf{0}$ follows for

$$\mathbf{v} = \mathbf{Vu} = \{v_{ij}\}. \tag{40}$$

We further verify that for the eigenvectors given in (39) and (40) the matrices $\mathbf{H}_m$, $\mathbf{G}_m$ defined by (34) and (35) satisfy conditions (12) as required.

It is interesting to note that v_{ij} is a discrete representation of $\rho\, dV$ so that $\sum_i \sum_j q_{ij} v_{ij}$ is a natural finite-difference approximation to the integral in (2) (we recall that the contributions of g_R are already included in q_{ij}). If q_{ij} satisfies (32), then one sees that this sum over all mesh cells is zero. Thus, $\mathbf{v}^{\mathrm{T}}\mathbf{q} = 0$ has the same meaning as consistency condition (2).

4.4 *Application with Fast Elliptic Solvers*

In the code FLUX we assume that the densities ρ_{ij} and the mesh spacings Δz_j are constant within each of the five domains shown in Fig. 3. For this case the finite-difference equations have a structure such that a fast elliptic solver can be applied for each domain. Here, we use the subroutine POISTP described in [8] which is based on the algorithm given in [3]. This special algorithm is particularly suited for our purpose because it does not restrict the number of mesh cells on each domain to a power of two like most of the other fast elliptic solvers. FLUX can also be applied to nonsingular problems as they appear for compressible flow. In this case the coefficients b_i are set to zero everywhere. Further we can treat cases where on some part of ∂R Dirichlet boundary conditions are applied. In this case some of the matrices $\mathbf{L}_i$ are nonsingular and no consistency condition has to be enforced for these parts.

As the new algorithm is a small part of the overall solution scheme, we cannot report on the performance of this part in detail. It has been found, however, that for $I = 10$ ($n = 660$), the number m of "irregular" equations which defines the size of the influence matrix, is $m = 31$. For such values the amount of work spent in solving for $\mathbf{w}$ in (24b) (after L–U decomposition of $\mathbf{C}$) is smaller than the amount of work required to solve one B problem. Thus, the latter, which is of order $n \log n$, controls the overall computing time. The residuum of the numerical solution using 16 digits arithmetic is typically less than $10^{-10}\|\mathbf{q}\|$. In summary, the algorithm works completely satisfactorily.

REFERENCES

1. R. ZURMÜHL, "Matrizen," p. 98ff, Springer-Verlag, Berlin, 1964.
2. H. LOMAX AND E. D. MARTIN, *J. Comput. Phys.* **15** (1974), 55.
3. U. SCHUMANN AND R. A. SWEET, *J. Comput. Phys.* **20** (1976), 171.
4. P. N. SWARZTRAUBER AND R. A. SWEET, *ACM Trans. Math. Software* **5** (1979), 352.
5. R. W. HOCKNEY, *in* "Numerical Methods in Applied Fluid Dynamics" (B. Hunt, Ed.), pp. 1–48, Academic Press, London 1980.
6. B. L. BUZBEE, F. W. DORR, J. A. GEORGE, AND G. H. GOLUB, *SIAM J. Numer. Anal.* **8** (1971), 722.
7. U. SCHUMANN, *J. Comput. Phys.* **36** (1980), 93.
8. U. SCHUMANN, Kernforschungszentrum Karlsruhe Rep. KfK 2645, 1979.