Fifth International Conference on
Numerical Methods in Fluid Dynamics
Twente University of Technology, Enschede, The Netherlands
June 28 to July 3, 1976

DIRECT POISSON EQUATION SOLVER FOR POTENTIAL AND PRESSURE FIELDS ON A STAGGERED GRID WITH OBSTACLES

by
U.Schumann
Kernforschungszentrum Karlsruhe, Institut für Reaktorentwicklung
7500 Karlsruhe, Postfach 3640, Federal Republic of Germany

and
R.A.Sweet
National Center for Atmospheric Research, Boulder, Colorado 80303, USA

(~~To be~~ published by Springer Verlag as a volume 59 of lecture notes in physics),
p. 398-403

DIRECT POISSON EQUATION SOLVER FOR POTENTIAL AND PRESSURE FIELDS ON A STAGGERED GRID WITH OBSTACLES

by
U. Schumann
Kernforschungszentrum Karlsruhe, Institut für Reaktorentwicklung
7500 Karlsruhe, Postfach 3640, Federal Republic of Germany

and
R.A. Sweet
National Center for Atmospheric Research, Boulder, Colorado 80303, USA

1. GENERALIZED POISSON EQUATION SOLVER

Many two-dimensional fluid dynamics problems require the solution of Poisson's equation

$$a(x)\,\frac{\partial^2 u}{\partial x^2} + b(x)\,\frac{\partial u}{\partial x} + c(x)\,u + \frac{\partial^2 u}{\partial y^2} = f(x,y). \tag{1}$$

If (1) is discretized by central second order finite differences on a rectangular staggered grid $\{\ x_i = (i-\frac{1}{2})\Delta x,\ i=0,1,\ldots,M+1;\ y_j=(j-\frac{1}{2})\Delta y,\ j=0,1,\ldots,N+1\ \}$ we get the block-tridiagonal linear system for $v_{i,j} \approx u(x_i,y_j)$:

$$\begin{bmatrix} (A-\alpha I) & -I & & & \\ -I & A & -I & & \\ & & \cdots & & \\ & & -I & A & -I \\ & & & -I & (A-\beta I) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ v_{N-1} \\ v_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \cdot \\ \cdot \\ f_{N-1} \\ f_N \end{bmatrix} \tag{2}$$

where A is a tridiagonal M x M matrix with non-constant coefficients, I the M x M unit matrix, v_j the vector $\{v_{ij}\}$ and $f_j = -\{f(x_i,y_j)\}\cdot\Delta y^2$ (plus modifications accounting for boundary values). The values of α and β depend upon the boundary conditions in the y-direction:

$$\alpha = \begin{cases} 0 & \text{for } v_{i,o} = 0 \\ 1 & \text{for } v_{i,1}-v_{i,o} = 0 \end{cases} \quad ; \quad \beta = \begin{cases} 0 & \text{for } v_{i,N+1} = 0 \text{ (Dirichlet)} \\ 1 & \text{for } v_{i,N+1}-v_{i,N} = 0 \text{ (Neumann)} \end{cases} \tag{3}$$

The boundary conditions in x-direction are arbitrary and may include periodicity conditions. An efficient non-iterative algorithm for the solution of (2) has been described in /¯1_7 for the case of $\alpha = \beta = 1$ (Neumann boundary conditions). The procedure is based on Buneman's cyclic reduction (CYR) /¯2,3_7 in a version suited for arbitrary values of M and N. The common restriction $N = 2^k$ /¯2_7 as required in older versions is abandoned. Nevertheless the new algorithm requires only an order $M \cdot N \cdot \log N$ operations and slightly more than $M \cdot N$ storage locations. Here we shall describe an extension to Dirichlet conditions.

The idea of the algorithm is as follows. We reduce the number of unknowns in a sequence of reduction steps ($r = 0,1,2,\ldots,k \approx \log_2 N$) in which we eliminate every second unknown v_j by a proper linear combination of adjacent equations. Hereby the original block-tridiagonal matrix is reduced to a new block-tridiagonal matrix with about half as many rows and columns. Its elements become polynomials of degree 2^r in A. These products are not computed explicitely since this would decrease the efficiency and accuracy considerably. Instead we describe the polynomials implicitely in terms of their roots. This can be done for each equation as known from older CYR versions. Special attention is required for the last equation where we have to use different linear combinations depending on whether the remaining number of equations is odd or even. In case of an even number, the elimination of the last unknown requires the inversion of a matrix. The explicit computation of this inversion, which would be very expensive, can be avoided by writing the matrix at the r-th reduction step as

$$\begin{bmatrix} K^{(r)} & -I & & \\ -I & A^{(r)} & -I & \\ & \cdots & & \\ & -I & A^{(r)} & -I \\ & & -I & ([C^{(r)}]^{-1} B^{(r)}) \end{bmatrix} \qquad \begin{aligned} K^{(o)} &= A - \alpha I \\ A^{(o)} &= A \\ \\ B^{(o)} &= A - \beta I \\ C^{(o)} &= I \end{aligned} \qquad (4)$$

where we explicitely keep the denominator $C^{(r)}$. Now it is possible not only to express $A^{(r)}$ as a polynomial of the form

$$A^{(r)} = \prod_{j=1}^{2^r} (A - \lambda_j^{(r)} I) \qquad (5)$$

with analytically determinable roots $\lambda_j^{(r)}$, but rather to give similar expressions for $K^{(r)}$, $B^{(r)}$ and $C^{(r)}$ as well.

For Neumann boundary conditions the details are given in [1]. The appendix of this paper contains the polynomial representations for Neumann and Dirichlet conditions. A FORTRAN-subroutine POISSX has been coded that solves (2) for both types of boundary conditions. It takes typically 1 s to solve (1) for M = N = 100 on an IBM 370/168 or 0.3 s on the NCAR-CDC 7600. A PL/1-version is also available.

2. CAPACITANCE MATRIX APPROACH FOR IRREGULAR REGIONS

The CYR method is most efficient only if applied to rectangular regions. It is not directly applicable for flow regions with obstacles or other geometrical irregularities. It has been shown, however, [4,5] that one can find the solution of Poisson's equation, say $\underline{A}\ \underline{u} = \underline{v}$, on an irregular region by twice solving some slightly modified Poisson's equation, say $\underline{B}\ \underline{\hat{u}} = \underline{v}$ and $\underline{B}\ \underline{u} = \underline{v} + \underline{w}$, for the rectangular region. The matrix $\underline{B}$ is chosen equal to $\underline{A}$ for most except for some p equations which correspond to the irregular grid points. For details we must refer to the references. The perturbation $\underline{w}$ of the right hand side is computed from the first step solution $\underline{\hat{u}}$ by use of the inverse of the so called capacitance matrix C. C^{-1} is a p x p matrix which is independent of $\underline{v}$ and can be precomputed by solving p systems $\underline{B}\ \underline{x} = \underline{y}$. This requires an order $O(p\ M\ N \log N) + O(p^3)$ operations. Once C^{-1} is known, the solution of $\underline{A}\ \underline{u} = \underline{v}$ requires an order $O(2\ M\ N \log N) + O(p^2)$ operations. The flexibility of the method is demonstrated by Fig.1 which shows a potential flow computed using POISSX.

3. APPLICATION IN "REMAC", A SMAC-TYPE FLUID DYNAMICS CODE

The capacitance matrix approach using POISSX becomes efficient if the system $\underline{A}\ \underline{u} = \underline{v}$ has to be solved many times. This is the case in viscous incompressible fluid dynamics. If we use the primitive variables velocity and pressure, then we have to solve at each time step Poisson's equation for the pressure. The pressure must satisfy Neumann boundary conditions at walls (including internal obstacles) and at prescribed velocity in- and outflow boundaries. Dirichlet conditions are given at continuative outflow boundaries and at free surfaces. All these options are possible in the SMAC code [6], a PL/1 version of which (REMAC) has been implemented in the REGENT system [7]. SMAC uses a staggered grid. We have implemented POISSX in REMAC. Due to its flexibility with respect to the values of M and N and by use of a general procedure for treating obstacles, free surfaces, and varying boundary conditions with the capacitance matrix approach, the new version of REMAC keeps the old flexibility of SMAC. Some extensions for even more complicated obstacle and inflow/outflow configurations are in their debugging phase.

Several test cases (A to F) have been run in order to check the new (POISSX-) version of REMAC and to compare the computing times with the older (SOR-) version where Poisson's equation was solved by successive overrelaxation. Fig.2 shows vector plots of typical resulting velocity fields. Cases A to E are run with a Reynolds number of the order one; in case F it is of the order 100. Cases A to E have been run for 30 time steps, case F for 80 steps. The time step is about one half the value allowed by stability.The overrelaxation parameter is 0.8 for cases A to E and 0.92 for case F. The initial velocity is zero except for case E where the initial velocity field corresponds to one large vortex.All cases are in Cartesian coordinates except for case E which is run in cylindrical coordinates. A velocity vector has been plotted for each cell filled with fluid. During the first 30 time steps the expected recirculating flow did not develop in cases B, C and D. The ratio R between the SOR-REMAC and the POISSX-REMAC computing times is given in the figure captions. We see, the gain in efficiency is typically a factor ten. We must note, however, that the SOR iteration scheme which is as described in /¯6_7 can be made more efficient. In general, the POISSX version is faster than the SOR version especially in the following situations:
a) Neumann boundary conditions at all walls, b) no free surfaces,
c) highly transient flow problems (no steady state), d) not too many obstacles (p of the order of M or N), e) large time steps.
In other situations, especially with many and fast moving free surface cells (a change in the surface configuration requires a new capacitance matrix) the SOR version is faster. We conclude that the fast direct Poisson solver POISSX in combination with the capacitance matrix technique is competitive with iterative schemes even in a general purpose production code.

Mr.K.H.Bechler's assistance in coding the PL/1 version is gratefully acknowledged.
The National Center for Atmospheric Research is sponsored by the National Science Foundation.

REFERENCES

/¯1_7 U.Schumann and R.A.Sweet, J.Comp.Phys. 20 (1976) 171-182
/¯2_7 B.L.Buzbee, G.H.Golub and C.W.Nielson, SIAM J.Numer.Anal. 7 (1970) 627-656
/¯3_7 U.Schumann, Über die direkte Lösung der diskretisierten Poisson-Gleichung mittels zyklischer Reduktion. KFK-Ext.8/75-6 (1976)

/‾4_7 B.L.Buzbee, F.W.Dorr, J.A.George and G.H.Golub, SIAM J.Numer.Anal. 8 (1971) 722-736
/‾5_7 E.D.Martin, Comp. and Fluids 2 (1974) 79-97
/‾6_7 A.A.Amsden and F.H.Harlow, The SMAC method: A numerical technique for calculating incompressible fluid flows. LA-4370 (1970)
/‾7_7 G.Enderle, Definition,Übersetzung und Anwendung benutzerorientierter Sprachen als Erweiterung von PL/1 in dem System für das rechnerunterstützte Entwickeln und Konstruieren REGENT. KFK 2204 (1975)

APPENDIX: ROOTS OF $A^{(r)}$, $K^{(r)}$, $B^{(r)}$, $C^{(r)}$ FOR $(\alpha,\beta) \in (0,1)$

Using the substitution $A = 2\cos\theta$ we find $A^{(r)} = 2\cos 2^r\theta$; $K^{(r)} = \sin[(2^r+1)\theta]/\sin\theta$ for $\alpha = 0$ and $K^{(r)} = \cos[(2^r+1/2)\theta]/\cos(\theta/2)$ for $\alpha = 1$; $B^{(r)} = \sin[(k_r+1)\theta]/\sin\theta$ for $\beta = 0$ and $\cos[(k_r+1/2)\theta]/\cos(\theta/2)$ for $\beta = 1$; $C^{(r)} = \sin[(\ell_r+1)\theta]/\sin\theta$ for $\beta = 0$ and $C^{(r)} = \cos[(\ell_r+1/2)\theta]/\cos(\theta/2)$ for $\beta = 1$ (k_r and ℓ_r are as defined in /‾1_7). From this it is easy to show that the required roots $\lambda_j^{(r)}$ as in (5) are of the form $\lambda_j^{(r)} = 2\cos\theta_j$. In case of $A^{(r)}$ these roots are independent of α and β and as given in /‾1_7. For $K^{(r)}$: $\theta_j = (2j-\alpha)\pi/(2^{r+1}+2-\alpha)$, $B^{(r)}$: $\theta_j = (2j-\beta)\pi/(2k_r+2-\beta)$, $C^{(r)}$: $\theta_j = (2j-\beta)\pi/(2\ell_r+2-\beta)$. We see: the change in boundary conditions merely requires some generalized coding of these roots.

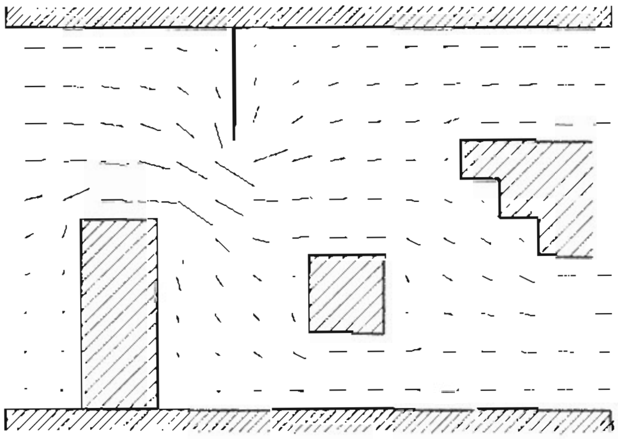

Fig.1: Velocity field in a channel with obstacles for potential flow. The flow is from left to right.

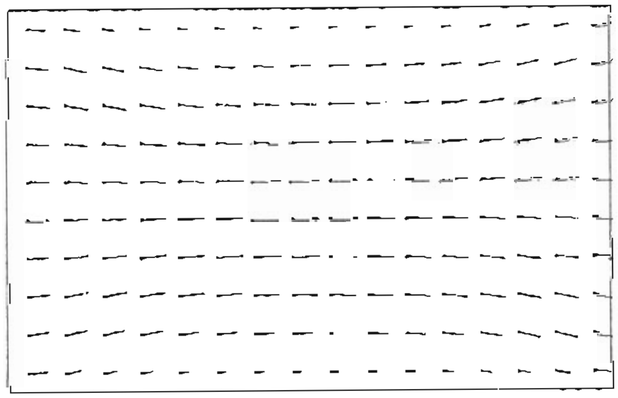

Fig.2A: Prescribed inflow and prescribed outflow. This corresponds to Neumann boundary conditions for the pressure everywhere. The physically unrealistic outflow condition has been used to test this type of boundary condition. R = 10.2

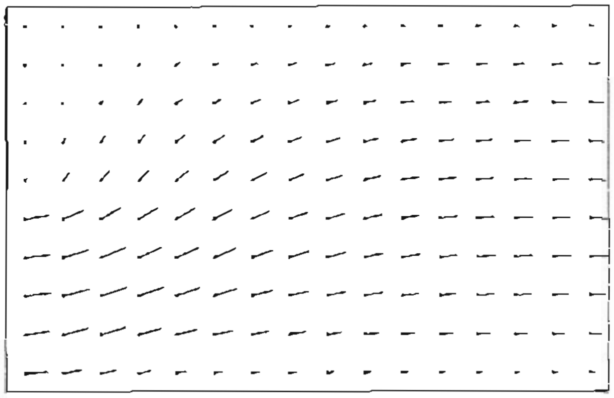

Fig.2B: Prescribed inflow for the lower part of the left boundary, continuative /¯6_7 outflow at the right, corresponding to a Dirichlet condition for the pressure. R = 10.4

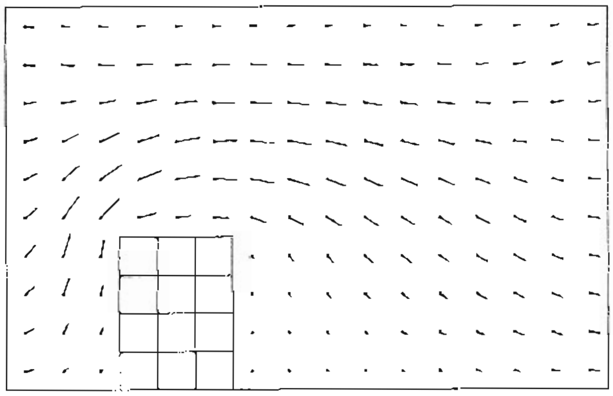

Fig.2C: Flow over an obstacle with prescribed in- and outflow. R = 9.2

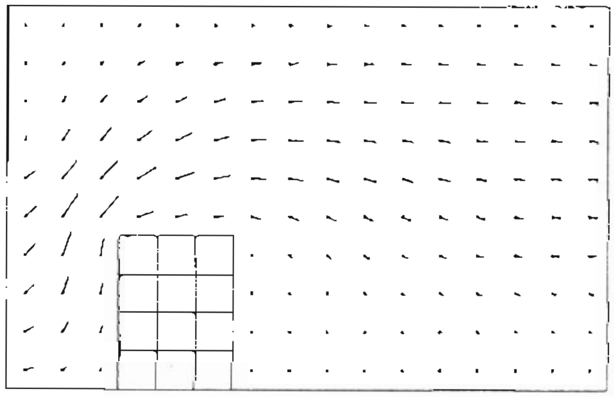

Fig.2D: Flow over an obstacle with prescribed inflow and continuative outflow. R = 35

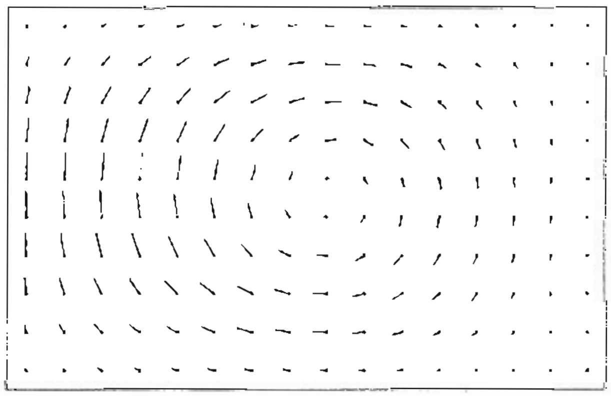

Fig.2E: Decaying vortex in a cylindrical container. The axis of symmetry corresponds to the left boundary. R = 30

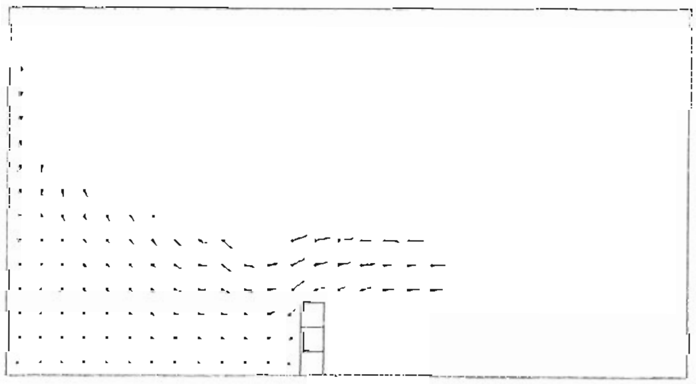

Fig.2F: Broken dam problem of /¯6_7 with an obstacle. R = 0.3