UNIVERSITY OF READING

DEPARTMENT OF MATHEMATICS

COST-EFFICIENCY STUDY OF VARIOUS METHODS FOR SOLVING THE SEEPAGE THROUGH DAM PROBLEM USING VARIATIONAL INEQUALITIES

A.W. Craig and W.L. Wood

Numerical Analysis Report 3/80

June 1980

Summary

The problem of seepage flow through a dam is solved using variational inequalities and finite element methods. The finite elements used are (i) linear triangles, (ii) bilinear rectangles (iii) quadratic triangles. The discretised problem is then solved by (1) Cryer's algorithm [1], (2) Rusin's algorithm [2]. These methods are compared for cost-efficiency by observing the computer time necessary for (a) converging the value of the functional to be minimised to four significant figures, (b) converging the position of the free surface to three significant figures. The conclusion is that it is better to use Cryer's algorithm but whether linear triangles or bilinear rectangles are better depends on how the cost-efficiency is measured.

1. Introduction

The problem studied here is that of seepage through a homogeneous, isotropic dam separating two reservoirs at different levels. This gives rise to a problem where there are two boundary conditions to satisfy on the free boundary. The standard method of solution has previously been to begin by guessing the position of the free boundary, and then after solving the differential equation using one of the boundary conditions, to adjust the position of the free boundary to make the other boundary condition hold and then re-solve the equation, adjust again and iterate towards a solution.

However in 1973 Baiocchi ([3], [4]) introduced a method in which the domain is fixed and the problem formulated as the solution of a variational inequality. This has two main advantages, firstly it leads to existence and uniqueness theorems for the solution, and secondly the formulation suggests numerical methods for solution which compete very favourably both in simplicity of programming and time of execution with the previously existing methods.

In this paper we consider several different methods of solving the problem using variational inequalities and attempt to find the most efficient.

Statement of the problem (fig. 1)

(The notation used here is the same as in Baiocchi [1]. In particular $[a, b]$ is a closed interval and $]a, b[$ is open).

Let a, y_1, y_2 be (real) numbers such that $a > 0, y_1 > y_2 \geq 0$, find a function $y = \phi(x)$, such that

$$\left.\begin{array}{l}\phi \text{ is defined and 'smooth' in } [0, a] \\ \phi(0) = y_1; \quad \phi(a) \geq y_2\end{array}\right\} \qquad (1.1)$$

such that setting

$$\Omega = \{(x, y) : 0 < x < a, 0 < y < \phi(x)\} \tag{1.2}$$

there exists a function $u(x, y)$ defined and 'smooth' in $\overline{\Omega}$, such that

$$\nabla^2 u = 0 \text{ in } \Omega \tag{1.3}$$

$$\left.\begin{array}{l} u = y_1 \text{ on } [AF] : u = y_2 \text{ on } [BC] \\ u = y \text{ on } [CC_\phi] \end{array}\right\} \tag{1.4}$$

$$u = y \text{ on } \widehat{FC}_\phi \tag{1.5}$$

$$\frac{\partial u}{\partial y} = 0 \text{ on } [AB] \tag{1.6}$$

$$\frac{\partial u}{\partial n} = 0 \text{ on } \widehat{FC}_\phi \ . \tag{1.7}$$

Now let us set $D =]0, a[\times]0, b[$, and extend the function u continuously in the following way

$$\tilde{u}(x, y) = \begin{cases} u(x, y), & (x, y) \in \overline{\Omega} \\ y \quad , & (x, y) \in \overline{D}\backslash\overline{\Omega} \end{cases} \tag{1.8}$$

and define the following transformation

$$w(x, y) = \int_y^b (\tilde{u}(x, t) - t)dt \ . \tag{1.9}$$

It can be shown (see [3]) that the following relations hold:

$$\left.\begin{array}{lll} w = \dfrac{y_1^2}{2} - \dfrac{y_1^2 - y_2^2}{2a} x & \text{on} &]AB[\\ w = \frac{1}{2}(y_1^2 - y^2) & \text{on} &]AF[\\ w = \frac{1}{2}(y_2^2 - y^2) & \text{on} & [BC] \\ w = 0 \text{ elsewhere on } \partial D & & \end{array}\right\} \tag{1.10}$$

$$\nabla^2 w = \chi_\Omega, \text{ the characteristic function of } \Omega \tag{1.11}$$

Remark (1). If we calculate $\left.\frac{\partial w}{\partial x}\right|_{y=0}$ from eq. (1.9) we see that

$$\left.\frac{\partial w}{\partial x}\right|_{y=0} = \int_0^{\phi(x)} u_x(x, t) = -\frac{q}{k}$$

where q is the discharge of the dam and k the permeability of the medium. However we also have for a rectangular dam

$$q = k\,\frac{y_1^2 - y_2^2}{2a} \quad ;$$

that is we could write the boundary condition on $]AB[$ as

$$w = \frac{y_1^2}{2} + \frac{q}{k}x.$$

In other problems associated with this one the value of q is an unknown, and we would have to find some compatability condition which the solution must satisfy in order to find the true value of q. (See e.g. Bruch [5]).

Remark (2). It is obvious from transformation (1.9) that the following relation holds

$$w(x, y) = 0 \quad \text{on} \quad \overline{D}\backslash\overline{\Omega} \tag{1.12}$$

and it is also possible to show that (see [3])

$$w(x, y) > 0 \quad \text{on} \quad \Omega \quad . \tag{1.13}$$

Thus the free boundary is uniquely determined from the values of w.

The function w also has the following important property: if we set

$$K^+ = \{v : v \in H'(D);\ v \text{ satisfies conditions (1.10)};\ v \geq 0 \text{ a.e. on } D\} \tag{1.14}$$

then w is the solution of the variational problem:

Find $w \in K^+$ such that

$$\left.\begin{aligned} &\int_D \nabla w.\nabla(v - w)dxdy + \int_D (v - w)dxdy \geq 0 \\ &\forall\ v \in K^+ \end{aligned}\right\} \tag{1.15}$$

and because of the symmetry of the bilinear form in the inequality (1.15), w is also the solution of the following minimisation problem:

Find $w \in K^+$ such that

$$\left.\begin{aligned} &J^{+}(w) \le J^{+}(v) \quad \forall\, v \in K^{+} \\ \text{where} \quad &J^{+}(v) = \tfrac{1}{2}\int_D (\nabla v)^2 + \int_D v \end{aligned}\right\} \qquad (1.16)$$

This is the problem which we shall be considering.

2.

We now want to compare the following methods of solution of the minimisation problem:

the region is discretised into finite elements on a regular mesh of rectangles such that the larger side of each is h. The three different types of elements used are

(i) linear triangles

(ii) bilinear rectangles

(iii) quadratic triangles.

When the equations have been set up using these three different discretisations we solve them using two different algorithms, the first due to Cryer [1] and the second to Rusin [2].

Cryer's Algorithm

If A is an $n \times n$, positive definite, symmetric matrix and $\underline{f}$ is a real n-dimensional column vector, then Cryer's algorithm is a method of solving the following problem:

Problem 1. Find an n-dimensional column vector $\underline{v}$ which minimises the functional

$$g(\underline{v}) = \tfrac{1}{2}\underline{v}^T A \underline{v} - \underline{f}^T \underline{v} \qquad (2.1)$$

subject to $\underline{v} \ge 0$. (2.2)

The algorithm is a modification of the well known successive overrelaxation (SOR) technique, and is as follows:

choose an n-dimensional column vector $\underline{v}^{(0)} = \{v_i^{(0)}\}$ where $\underline{v}^{(0)} \geq 0$ and a parameter ω (the relaxation parameter), such that $0 \leq \omega \leq 2$, and generate a sequence of vectors $\underline{v}^{(k)}$, $k = 1, 2, \ldots,$ by the following scheme

$$\left.\begin{aligned} v_i^{(k+\frac{1}{2})} &= \frac{f_i}{a_{ii}} - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} v_j^{(k+1)} - \sum_{j=i}^{n} \frac{a_{ij}}{a_{ii}} v_j^{(k)} \\ v_i^{(k+1)} &= \max\{0,\ v_i^{(k)} + \omega(v_i^{(k+\frac{1}{2})} - v_i^{(k)})\} \end{aligned}\right\} \qquad (2.3)$$

It can be proved (see [1]) that $\underline{v}^{(k)} \to \underline{v}$ as $k \to \infty$ where $\underline{v}$ is the solution of problem 1.

Estimation of optimum relaxation parameter

For normal SOR the optimum relaxation parameter can be worked out exactly (see [6]). We shall denote this by ω_A. In this problem however the optimum parameter ω_{opt} cannot be found exactly, but Cryer does show that

$$\omega_{opt} \leq \omega_A \qquad (2.4)$$

and we can approximate ω_{opt} by ω_A.

Rusin's Algorithm [2]

This algorithm can be applied to problem 1 together with a system of constraints of the form

$$B\underline{v} = \underline{b} \qquad (2.5)$$

where B is an $m \times n$ matrix, as well as the constraint $\underline{v} \geq 0$.

Here the algorithm is simplified to suit the present problem.

Define the vector $\underline{\delta}$ as

$$\underline{\delta} = \underline{v}^T A - \underline{f}^T \qquad (2.6)$$

(then $\delta_i = \frac{\partial}{\partial v_i} g(\underline{v})$ where $g(\underline{v})$ is defined in relation (2.1)).

With relation (2.6) we can now define a solution of problem 1 in which the following conditions hold

$$\left.\begin{aligned} \underline{v} &\geq 0 \\ \underline{\delta} &\geq 0 \\ \underline{\delta}\cdot\underline{v} &= 0 \end{aligned}\right\} \qquad (2.7)$$

These are just the Kuhn-Tucker conditions for the quadratic minimisation problem [7].

The simplified version of the algorithm is as follows:

at the beginning of the solution we start with a certain number, say k, variables which we know are strictly greater than zero in the final solution; we call this the basic set. We then partition the problem data as in fig. 2.

Then we actually find the values of $\underline{v}_B$ (the basic set) by solving the system

$$A_B\underline{v}_B = \underline{f}_B \qquad (2.8)$$

Next for all j not belonging to the basic set we calculate

$$\delta_j = \underline{a}_j^T\underline{v}_B - f_j \qquad (2.9)$$

and then find

$$\delta_s = \min_j \delta_j \; . \qquad (2.10)$$

Now, if $\delta_s \geq 0$ the present solution is optimal, that is it satisfies conditions (2.7) and is therefore a solution of problem 1. If however $\delta_s < 0$ then we include v_s in the basic set, rearrange and repartition the problem data and then start again from solving the system (2.8).

Programming the algorithms

We note here that we are searching for a solution in the convex set K^+ as defined in (1.12), that is $w(x, y)$ must be greater than or equal to zero everywhere, not just at the element nodes. For the linear and bilinear elements it is sufficient to insist that the solution is non-negative at the nodes, but this is not, in general, sufficient for the quadratic element.

Glowinski [8] remarks that he has used quadratic triangle elements for the obstacle problem [which is in effect a generalised version of the problem of seepage flow through a dam]. He considers the two conditions which would here be equivalent to

(1) $w(x, y) \geq 0$ at all interior nodes

(2) $w(x, y) \geq 0$ at interior midside nodes

and proves [9] that for either of these there is convergence to the exact solution as $h \to 0$ provided the angles of the triangular elements are bounded below by $\theta_0 > 0$ independent of h. This is true in the investigation reported in this paper because the triangular elements are obtained from a regular mesh. In this investigation Glowinski conditions (1) and (2) have been used and it is interesting to note that in both cases, even for a very coarse mesh of 8 triangles the solution obtained is non-negative within each element as well as at the nodes.

Cryer's algorithm presents no problems when programming. If we keep to a regular mesh, then the matrix has a regular pattern and so the storage requirements can be cut down considerably, in fact on the Reading ICL 1904 it was possible to run programs with meshes containing up to 10^4 nodes.

Rusin's algorithm presents more difficulty, the main problem being that in order to rearrange the problem data at the end of each iteration it is necessary to store the whole matrix.

In order to increase the number of available mesh points, the whole matrix equation is solved each time by SOR on a regular mesh, but keeping the non-basic variables equal to zero; hence we use the algorithm

$$v_i^{(k+1)} = v_i^{(k)} \quad \text{if} \quad v_i^{(k)} = 0$$

$$\left.\begin{aligned} v_i^{(k+\frac{1}{2})} &= \frac{f_i}{a_{ii}} - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} v_i^{(k+1)} - \sum_{j=1}^{n} \frac{a_{ij}}{a_{ii}} v_i^{(k)} \\ v_i^{(k+1)} &= v_i^{(k)} + \omega(v_i^{(k+\frac{1}{2})} - v_i^{(k)}) \end{aligned}\right\} \text{if} \quad v_i^{(k)} \neq 0 \tag{2.11}$$

This means that we can preserve the regular pattern of the matrix.

In going from one iteration to the next the values of the variables which are already basic do not change very much, so the old basic variables are used as a starting vector for the next solution, with the new basic variable initially set equal to that of the nearest basic variable on the mesh.

The starting nodes used as the first basic set are shown in fig. 3.

3. Results.

Two different methods (a) and (b) were used to estimate the efficiency of the numerical procedures.

(a) The value of the functional $J^{+}(v)$ (eq. (1.16)) was calculated for decreasing mesh sizes, and the method was considered to have converged if, on halving the mesh, the values of the functional agreed to four significant figures.

(b) From a more practical point of view the position of the free boundary was estimated in the following way: if, on looking up a mesh line in the y-direction the last non-zero node was the (i, j)th node, with value $w_{i,j}$, then a quadratic curve was fitted through the points $(i, j - 1)$, (i, j) and $(i, j + \theta)$, where $(i, j + \theta)$ is the point to be found where the function becomes zero (i.e. the free boundary). Using the values $w_{i,j-1}$, $w_{i,j}$ and the fact that $w_{i,j+\theta} = 0$ and $\left.\frac{\partial}{\partial y} w \right|_{i,j+\theta} = 0$, we can then find the value of θ.

Thus the position of the free boundary was calculated, the mesh lengths halved and the position of the free boundary was then recalculated. The method was considered to have converged if the successive positions of the free boundary agreed to 3 significant figures.

In slightly different problems from this one, e.g. a dam with a sloping base, another unknown, q, (the discharge) is introduced and Baiocchi [3] shows that

$$q = - \left.\frac{\partial w}{\partial x}\right|_{y=0}$$

Thus the boundary condition in equation (1.10) on]AB[(fig. 1) depends on the value of q. The discharge through any vertical section of the dam must be constant and it might be thought to be a good practical test of the method to see whether it modelled this correctly. However, for this problem, as the discharge is included explicitly in the boundary condition on]AB[it is always modelled exactly.

The particular values of geometrical parameters for which we present results are those suggested by Baiocchi [4] as typical for the problem and are as follows

$$y_1 = 3.22, \quad y_2 = 0.84, \quad a = 1.62 \quad . \qquad (3.1)$$

However experiments show that the results hold true for a wide range of parameters.

For this problem Rusin's algorithm did not compare favourably with Cryer's algorithm. Even the especially adapted form of Rusin's algorithm takes approximately 10 times longer to solve the same set of matrix equations.

It is likely though that Rusin would be much more competitive if the matrix was either full or irregular, rather than a sparse regular matrix. The reasons for this are two-fold:

(a) Cryer's algorithm would become less effective as we would now have to store the whole matrix, therefore the number of nodes which we could use would be reduced.

(b) We would no longer have to worry about preserving the structure of the matrix, so we could perform the row and column interchanges which would mean that at each step of the Rusin algorithm we need only solve the reduced matrix equation.

It also should be noted that Rusin's algorithm (in its original form) can deal with a more complicated problem than Cryer's algorithm.

The method of discretisation by quadratic elements turned out to be less efficient than either of the other two discretisations. Although the number of elements needed to make the method converge by criteria (a) and (b) above was reduced, the number of nodes needed was increased and consequently the size of the matrix system to be solved was increased.

In order for Glowinski's conditions (1) and (2) to converge to the same value of the functional it was necessary to use a more refined mesh with condition (2) than that used with condition (1), although the solutions for both were always non-negative everywhere in the region. The refinement necessary however was only slight and in practice meant a difference of approximately 1 second in computer time.

Bilinear elements were found to converge much more quickly when using the value of the functional as a measure of convergence. However for the position of the free surface both bilinear and linear element discretisations took about the same time to converge to the solution, and over a wide range of problem data it was not possible to conclude that one was better than the other.

This work was suggested by a result in a paper by Bruch, [10] where he implies that for the problem of seepage from a channel it is more efficient to use bilinear rectangles.

Table 1 shows the mesh and times of execution needed in order to obtain accuracy of four decimal places in the functional.

Table 2 shows the mesh and times of execution needed in order to obtain accuracy of three decimal places in the position of the free surface.

Table 1. Comparison of linear and bilinear elements by method (a) (minimisation of functional)

Type of elt	Mesh	Time of Execution	Value of $J^+(v)$
Linear	24 × 36	31 secs	12.9365
Bilinear	14 × 21	13 secs	12.9372

Table 2. Comparison of linear and bilinear elements by method (b) (position of free surface)

Type of elt	Mesh	Time of Execution
Linear	20 × 30	22 secs
Bilinear	18 × 27	24 secs

References

[1] C.W. Cryer, 'The solution of a quadratic programming problem using systematic overrelaxation'. SIAM J. Control 9 N3, 397-411 (1970).

[2] M.H. Rusin, 'A revised simplex method for quadratic programming'. SIAM J. Appl. Math. 20 N2, 143-160 (1971).

[3] C. Baiocchi, V. Comincioli, E. Magenes and G.A. Pozzi, 'Free boundary problems in the theory of fluid flow through porous media: Existence and Uniqueness Theorems'. Ann. Di Mat. Pur. et App. 47, 1-82 (1973).

[4] C. Baiocchi, V. Comincioli, L. Guerri and C. Volpi, 'Free boundary problems in the theory of fluid flow through porous media: Numerical Approach'. Calcolo 10, 1-85 (1973).

[5] J.C. Bruch, 'Seepage streams out of canals and ditches overlying shallow water tables'. J. Hydrol. 41, 31-41 (1979).

[6] R. Varga, Matrix Iterative Analysis. Prentice-Hall, New Jersey 1962.

[7] H.W. Kuhn and A.W. Tucker, 'Nonlinear Programming'. Proc. 2nd Berkeley Symposium on Mathematical Statistics and Probability, J. Neyman ed., Univ. of California Press, Berkeley and Los Angeles, 481-492 (1951).

[8] R. Glowinski, 'Finite Elements and Variational Inequalities'. I.R.I.A. (1978).

[9] R. Glowinski, 'Introduction to the approximation of elliptic variational inequalities'. Report 76006, Laboratoire d'Analyse Numérique, Université Pierre et Marie Curie (1976).

[10] J.C. Bruch, F.C. Sayle and J.M. Sloss, 'Seepage from a trapezoidal and a rectangular channel using variational inequalities'. J. Hydrol. 36, 247-260 (1978).

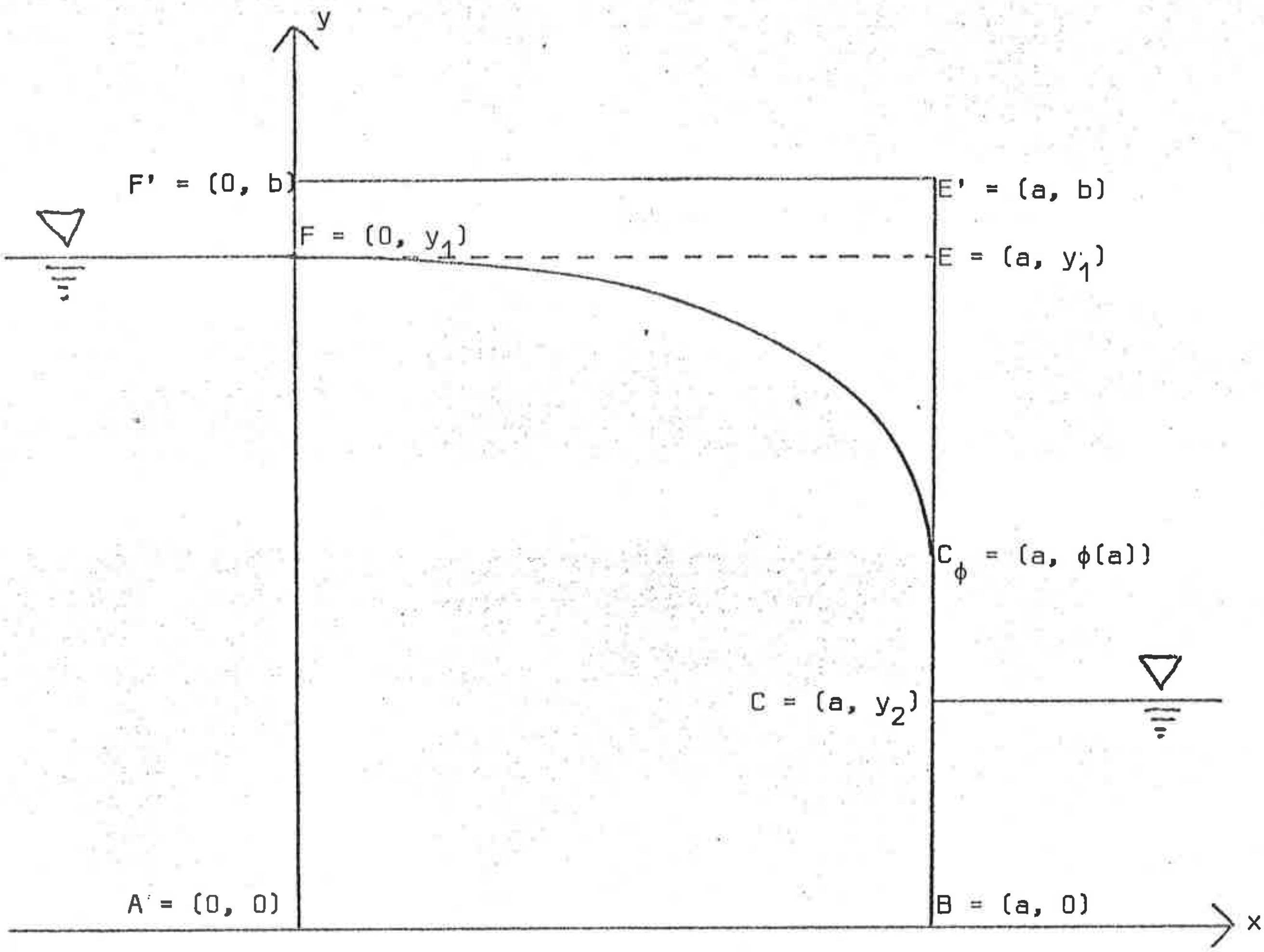

Figure 1. Rectangular Dam

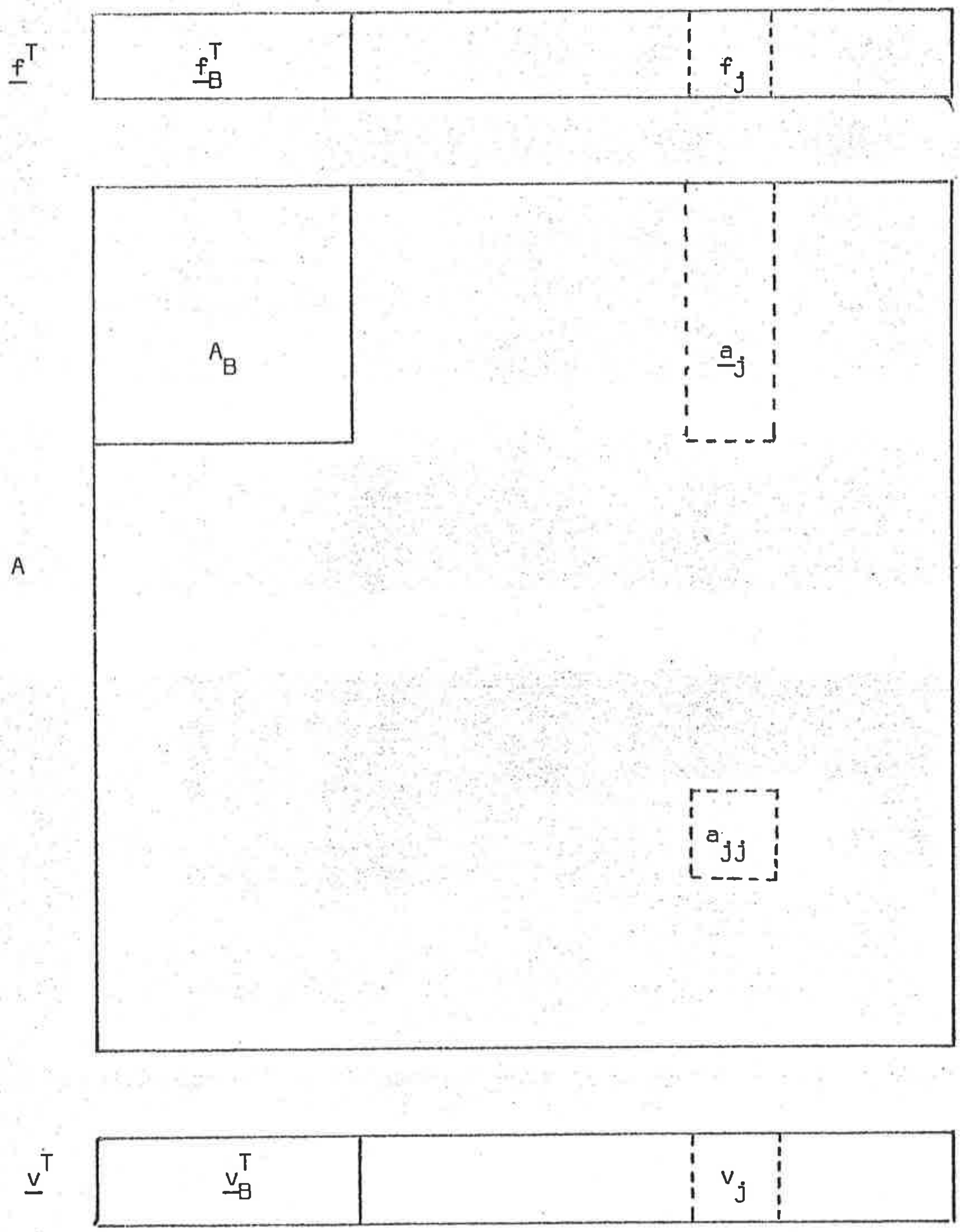

Figure 2. Partitioning for Rusin's Algorithm

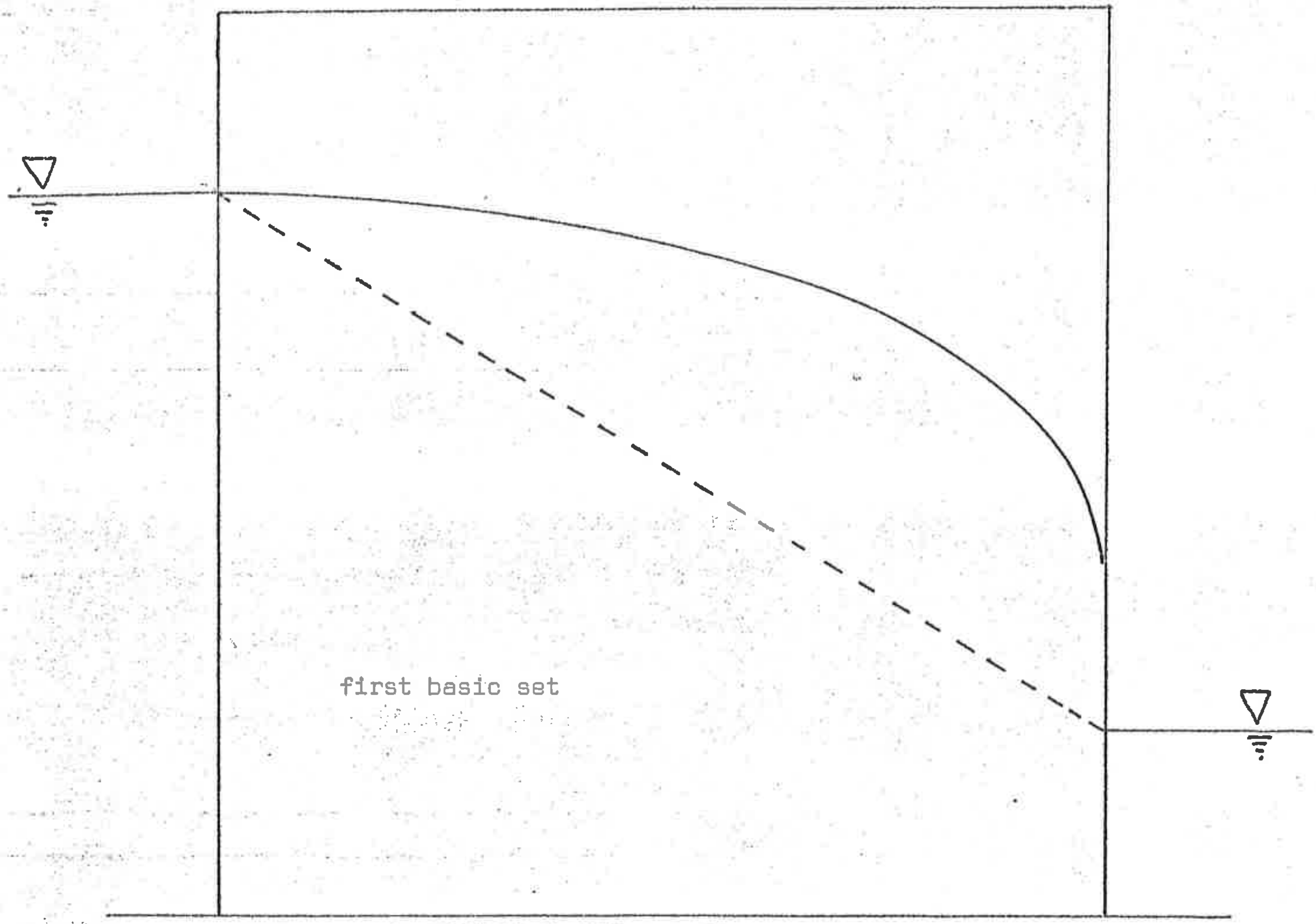

Figure 3. Starting nodes for Rusin's Algorithm