Domain decomposition methods for non linear problems in fluid dynamics

Roland Glowinski, Jacques Périaux, Q.V. Dinh

CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél: 954 90 20

Rapports de Recherche

N° 147

DOMAIN DECOMPOSITION METHODS FOR NONLINEAR PROBLEMS IN FLUID DYNAMICS

Roland GLOWINSKI
Jacques PERIAUX
Quang Vinh DINH

Juillet 1982

[27] POWELL M-J-D., Restart procedure for the conjugate gradient method, *Math. Programming*, 12, (1977), pp. 148-162.

[28] LANDAU L., LIFCHITZ E., *Mécanique des Fluides*, Mir, Moscou, 1953.

[29] POIRIER G., *Traitements Numériques en éléments finis de la condition d'entropie des équations transoniques*, Thèse de 3ème cycle, Université Pierre et Marie Curie, Déc. 1981.

[30] WILDE D.J., BEIGHTLER C.S., *Foundations of Optimization*, Prentice Hall, Englewood Cliffs, N.J., 1967.

[31] PERIAUX J., *Résolution de quelques problèmes non linéaires en aérodynamique par des méthodes d'éléments finis et de moindres carrés.* Thèse de 3e cycle, Université Pierre et Marie Curie, June 1979.

[32] PERRONNET A., The Club MODULEF : A Library of Subroutines for Finite Element Analysis, in *Computing Methods in Applied Sciences and Engineering, 1977, 1, Proceedings, IRIA Paris*, R. Glowinski, J.L. Lions eds., Lecture notes in Mathematics, Vol. 704, Springer-Verlag , Berlin, 1979, pp. 127-153.

[33] DINH Q.V., *Simulation numérique en éléments finis d'écoulements de fluides visqueux incompressibles par une méthode de décomposition de domaines sur processeurs vectoriels*, Thèse de 3e cycle, Université Pierre et Marie Curie, March 1982.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique

[11] GLOWINSKI R., PIRONNEAU O., On numerical methods for the Stokes problem, Chapter 13 of *Energy Methods in Finite Element Analysis*, R. Glowinski, E.Y. Rodin, O.C. Zienkiewicz eds., J. Wiley and Sons, Chichester, 1979, pp. 243-264.

[12] NECAS J., *Les méthodes directes en théorie des équations elliptiques*, Masson, Paris, 1967.

[13] ADAMS R.A., *Sobolev spaces*, Acad. Press, New York, 1975.

[14] ODEN J.R., REDDY J.N., *An introduction to the mathematical theory of finite elements*, John Wiley and Sons, New York, 1976.

[15] GLOWINSKI R., LIONS J.L., TREMOLIERES R., *Analyse Numérique des Inequations Variationnelles*, Vol. 1 and 2, Dunod-Bordas, Paris, 1976.

[16] GLOWINSKI R., LIONS J.L., TREMOLIERES R., *Numerical Analysis of Variational Inequalities*, North-Holland, Amsterdam, 1981.

[17] BENSOUSSAN A., LIONS J.L., TEMAM R., Sur les méthodes de décomposition de décentralisation de coordination et applications, in *Sur les Méthodes Numériques en Sciences Physiques et Economiques*, J.L. Lions, G.I. Marchouk eds., Dunod-Bordas, Paris, 1974, pp. 133-257.

[18] LEMONNIER P., Résolution numérique d'équation aux dérivées partielles par décomposition et coordination, in *Methodes Numériques d'Analyse de Systèmes, Vol. 2 : Méthodes de décomposition*, Cahier de l'INRIA n° 11, Juin 1972, pp. 191-239.

[19] LIONS J.L., *Controle optimal des systèmes gouvernés par des équations aux dérivées partielles*, Dunod-Gauthier Villars, Paris, 1968.

[20] GLOWINSKI R., LIONS P.L., PERIAUX J., Solution of boundary value problems by domain decomposition methods (to appear).

[21] DANIEL J., *The approximate minimization of functionals*, Prentice Hall, Englewood Cliffs, N.J., 1970.

[22] BRISTEAU M.O., GLOWINSKI R., PERIAUX J., PERRIER P., PIRONNEAU O., On the numerical solution of nonlinear problems in fluid dynamics by least squares and finite element methods (I) Least squares formulations and conjugate gradient solution of the continuous problems. *Comp. Meth. Appl. Mech. Eng.*, 17/18, (1979), pp. 619-657.

[23] GLOWINSKI R., *Numerical methods for nonlinear variational problems*, 2nd edition, Springer-Verlag (in preparation).

[24] CEA J., GEYMONAT G., Une méthode de linéarisation via l'optimisation *Istituto Nazionale di Alta Mat. Symp. Math.*, 10, Bologna, (1972), pp. 431-451.

[25] LOZI R., *Analyse Numérique de certains problèmes de bifurcation*, 3rd cycle Thesis, Université de Nice, 1975.

[26] POLAK E., *Computational methods in Optimization*, Acad. Press, New York, 1971.

REFERENCES

[1] BRISTEAU M.O., GLOWINSKI R., PERIAUX J., PERRIER P., PIRONNEAU O., POIRIER G., Application of Optimal Control and Finite Element Methods to the calculation of Transonic Flows and Incompressible Viscous Flows, in *Numerical Methods in Applied Fluid Dynamics*, B. Hunt Edit., Academic Press, London, 1980, pp. 203-312.

[2] GLOWINSKI R., MANTEL B., PERIAUX J., PIRONNEAU O., POIRIER G., An efficient preconditioned conjugate gradient method applied to nonlinear problems in Fluid Dynamics, in *Computing Methods in Applied Sciences and Engineering*, R. Glowinski, J.L. Lions eds., North Holland, Amsterdam, 1980, pp. 445-487.

[3] BRISTEAU M.O., GLOWINSKI R., PERIAUX J., PERRIER P., PIRONNEAU O., POIRIER G., Transonic flow simulations by finite elements and least squares methods, in *Proceedings of the Third International Conference on Finite Elements in Flow Problems, Banff, Alberta, Canada, 10-13 June 1980*, D.H. Norvie ed., Vol. 1, pp. 11-29.

[4] GLOWINSKI R., MANTEL B., PERIAUX J., PERRIER P., PIRONNEAU O., On an efficient new preconditioned conjugate gradient method. Application to the in cove solution of the Navier-Stokes equations via nonlinear least squarees and finite element methods, in *Proceedings of the Thrird International Conference on Finite Elements in Flow Problems, Banff, Alberta, Canada, 10-13 June 1980*, D.H. Norrie ed., Vol. 2, pp. 229-255.

[5] GLOWINSKI R., MANTEL B., PERIAUX J., Numerical solution of the time dependent Navier-Stokes equations for incompressible viscous fluids by finite elements and alternating direction methods, in *Proceedings of the Conference on Numerical Methods in Aeronautical Fluid Dynamics*, Reading, March 30th. - April 1st., 1981.

[6] HELLER D., A survey of parallel algorithms in numerical linear algebra, *SIAM Review*, 20, (1978), 4, pp. 740-777.

[7] MORICE P., Calcul parallèle et décomposition dans la résolution d'équations aux dérivées partielles de type elliptique, *IRIA Report* INF. 7214/72017, 1972.

[8] FRAILONG J.M., PAKLEZA J., Resolution of general partial differential equations on a fixed size SIMD/MIMD large cellular processor, in *Proceedings of the IMACS International Congress*, Sorente, Italie, Sept. 1979.

[9] LIONS P.L., Interpretation stochastique de la méthode alternée de Schwarz, *Comptes Rendus Acad. Sc., Paris*, T. 286 A, (1978), pp. 325-328.

[10] GLOWINSKI R., PIRONNEAU O., Numerical methods for the first biharmonic equation and for the two-dimensional Stokes problem, *SIAM Review*, 21, (1979), 2, pp. 167-212.

6. - CONCLUSION.

We have introduced in this paper new methods for solving partial differential problems founded on subdomain decomposition. The numerical results discussed in this paper show the efficiency and the robustness of these methods, applied to the solution of very large Poisson problems. The fundamental idea was the minimization of a well chosen distance (of least squares type) between the local solutions on the overlapping regions.

This minimization can be done by either a conjugate gradient algorithm (whose convergence can be accelerated using a suitable preconditioning) or by a quasi-direct method founded on decomposition properties of the least square formulation ; in the second case it is necessary to factorize once and for all a discrete boundary integral operator, which is symmetric, positive definite and block sparse. The two techniques, mentioned above, require the solution of local Poisson problems which can be solved independently one to each other, making these methods particularly well-suited to vector machines and MIMD processors.

The numerical simulations presented in this paper (transonic potential flows for inviscid fluids and incompressible viscous flows) are just a first step ; they show however the important role that can play domain decomposition methods for solving very large realistic problems (three-dimensional problems in Aerodynamics in particular).

We have not discussed here some other applications like :

- Solving problems involving several mathematical modellings according to the region under consideration (a typical example in that direction is the matching of viscous flows and inviscid flows).

- Coupling different types of approximation (finite elements - finite differences, spectral - finite differences, spectral - spectral, etc...).

ACKNOWLEDGEMENT : This work was partly supported by DRET, under contract 80/493.

INCIDENCE 30.00 NB REYNOLDS 100.0
CYCLE DE TEMPS 60 NB D ITERA 3
PAS DE TEMPS 0.10
P1/P1 ISO P2 LIGNES ISO-BARES

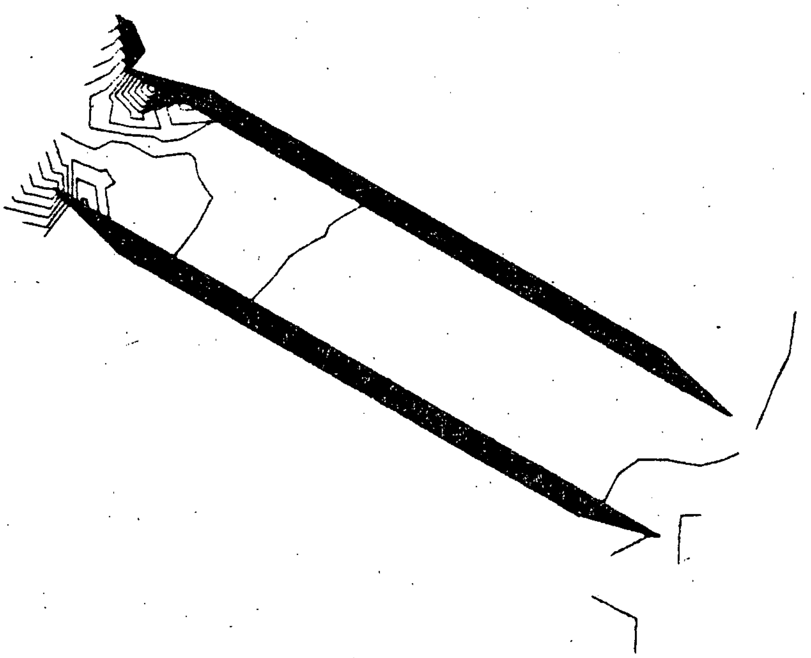


Figure 5.36
Pressure distribution
at t = 6.

INCIDENCE 30.00 NB REYNOLDS 100.0
CYCLE DE TEMPS 60 NB D ITERA 3
PAS DE TEMPS 0.10
P1/P1 ISO P2 LIGNES DE COURANT

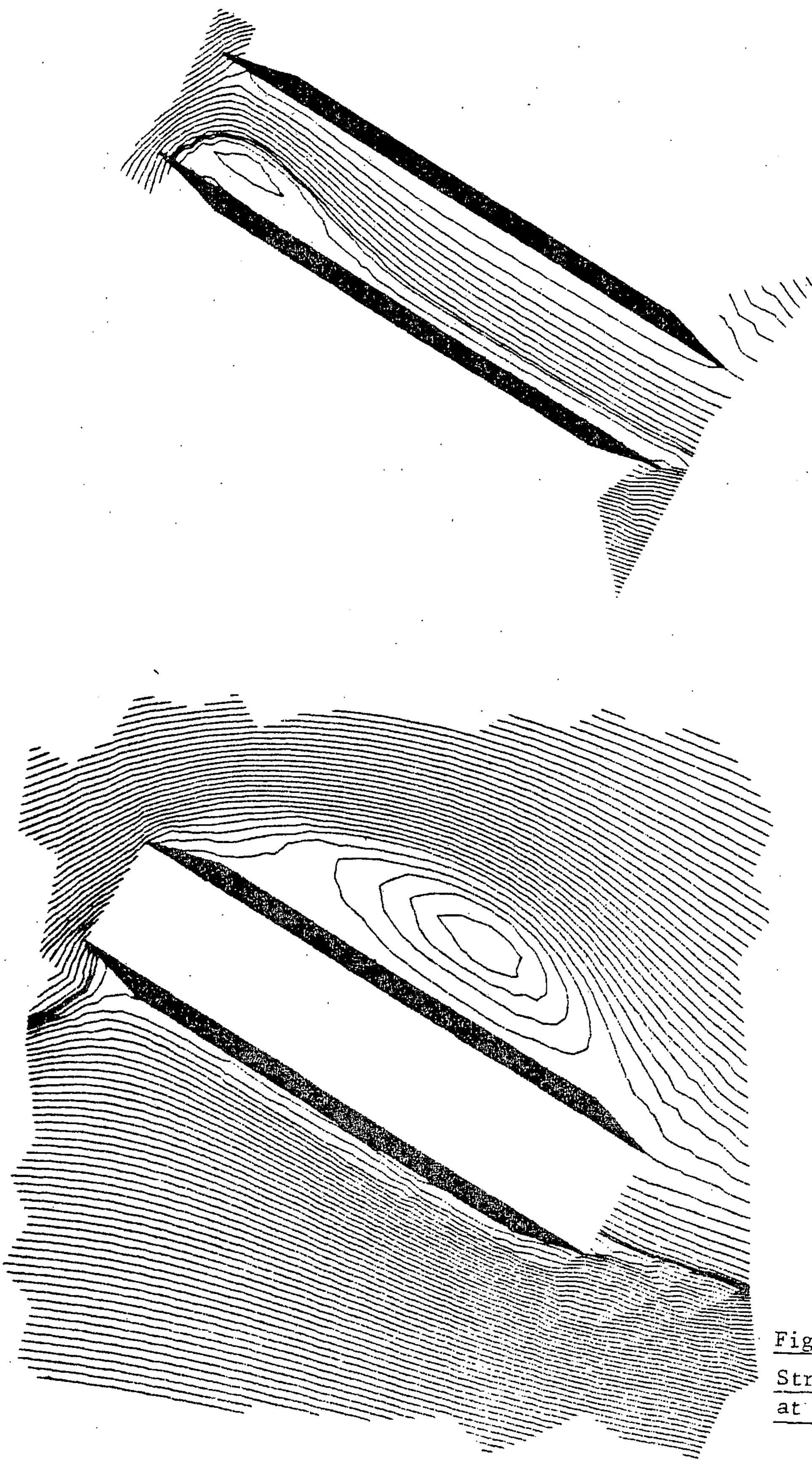


Figure 5.35

Stream lines at t = 6.

INCIDENCE 30.00 NB REYNOLDS 100.0
CYCLE DE TEMPS 40 NB D ITERA 3
PAS DE TEMPS 0.10
P1/P1 ISO P2 LIGNES ISO-BARES

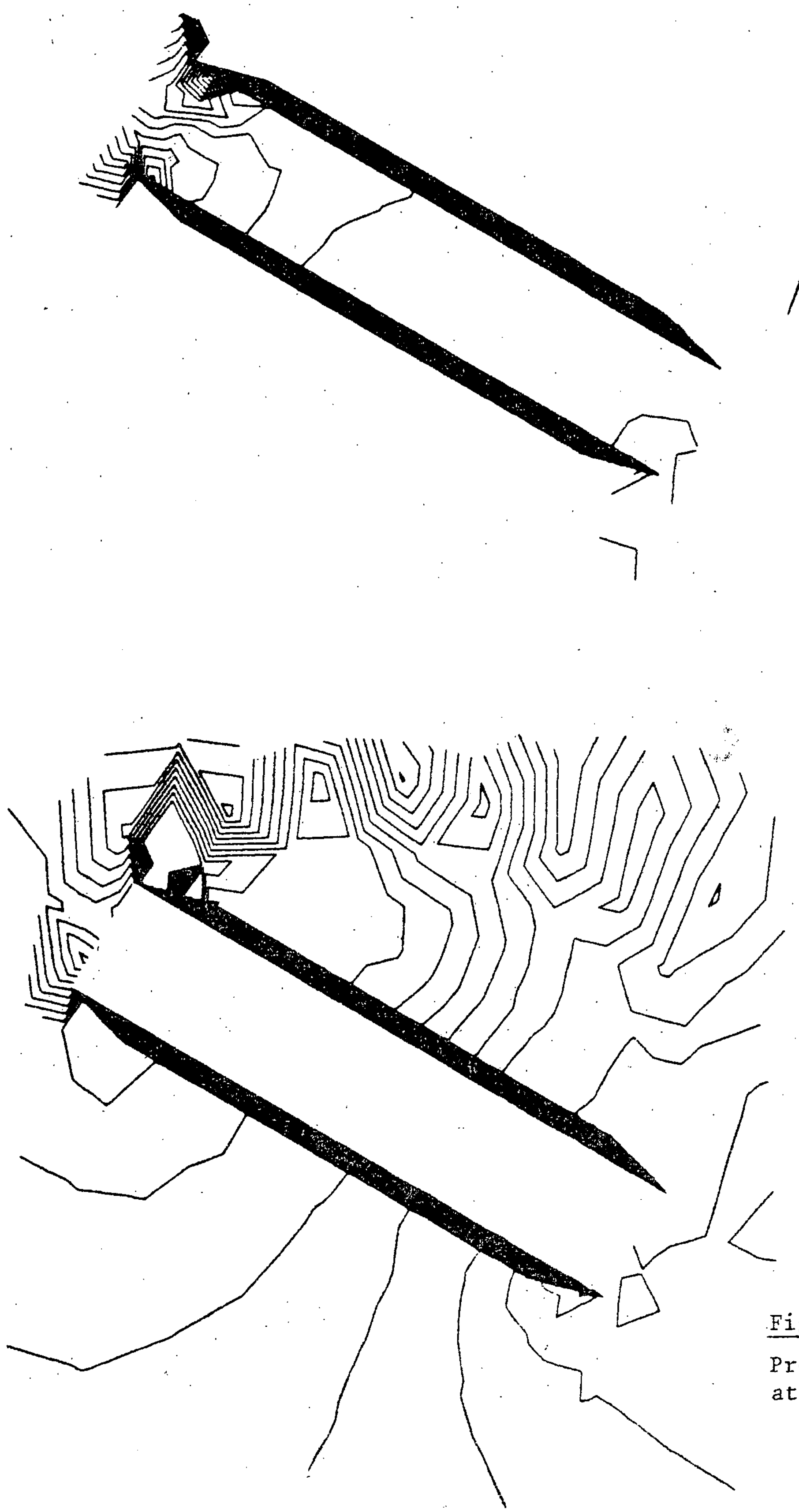


Figure 5.34

Pressure distribution at t = 4.

Figure 5.33

Stream lines at t = 4

INCIDENCE 30.00 NB REYNOLDS 100.0
CYCLE DE TEMPS 10 NB D ITERA 3
PAS DE TEMPS 0.10
P1/P1 ISO P2 LIGNES ISO-BARES

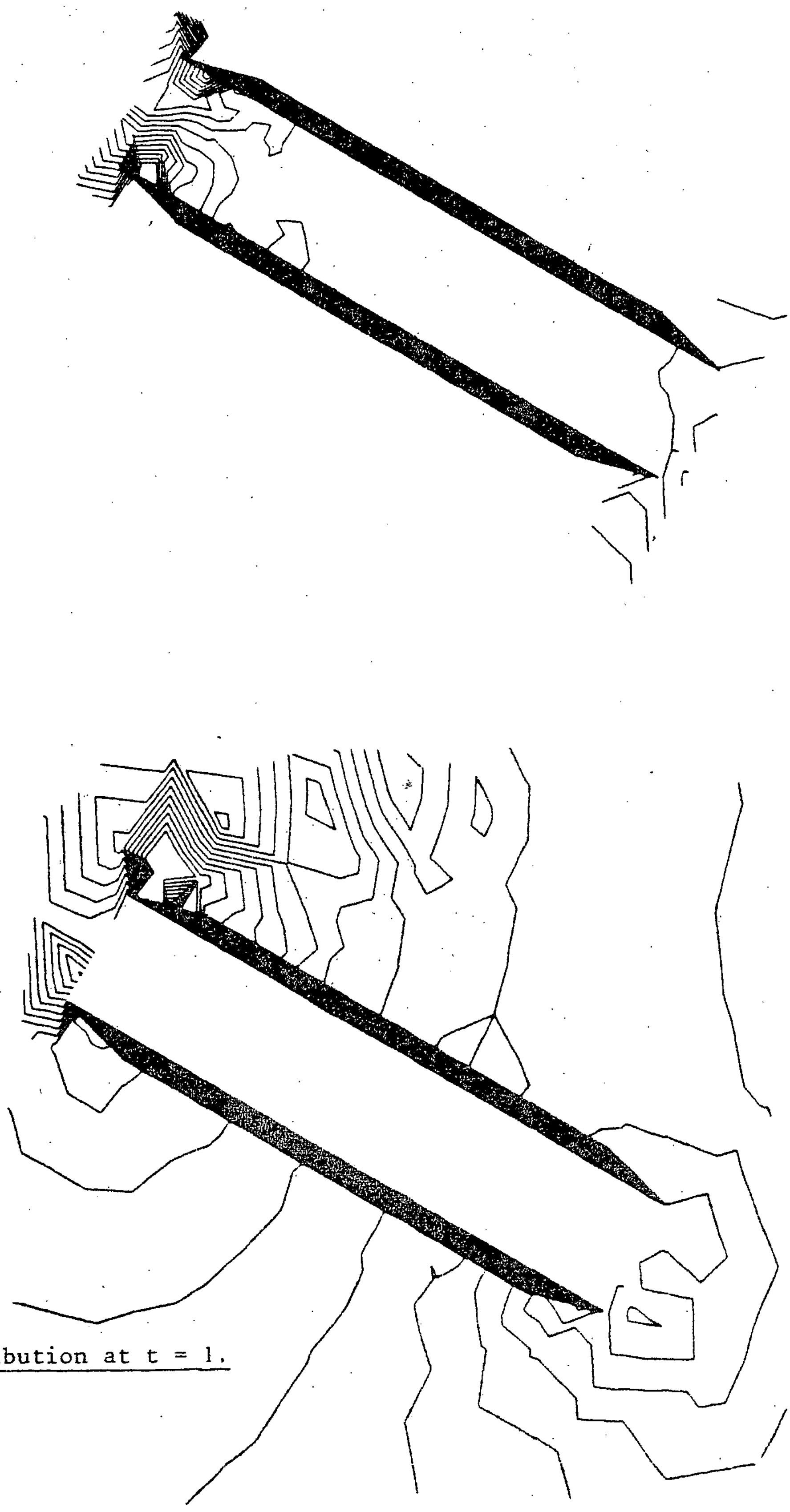


Figure 5.32

Pressure distribution at t = 1.

INCIDENCE 30.00 NB REYNOLDS 100.0
CYCLE DE TEMPS 10 NB D ITERA 3
PAS DE TEMPS 0.10
P1/P1 ISO P2 LIGNES DE COURANT

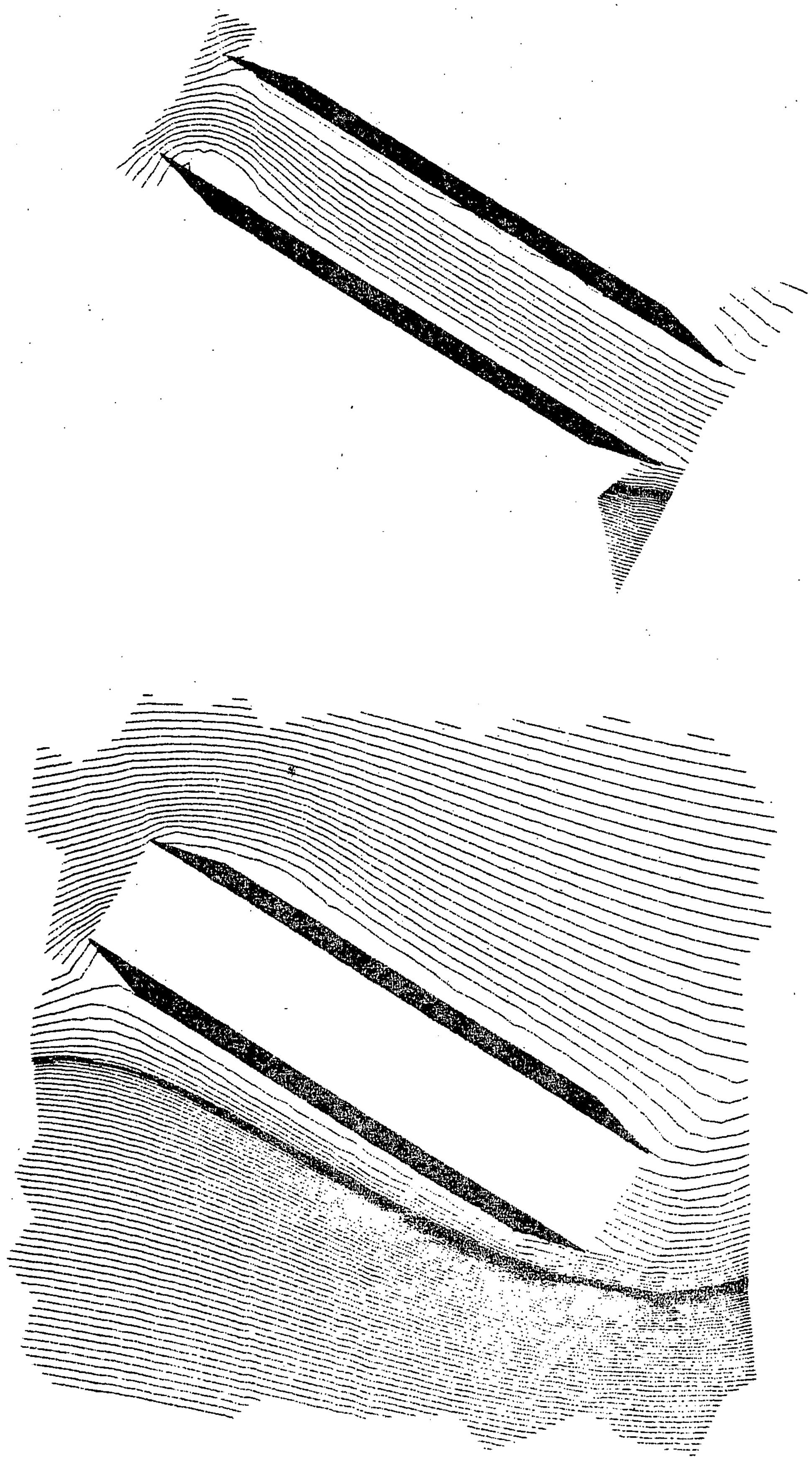


Figure 5.31
Stream lines
at t = 1.

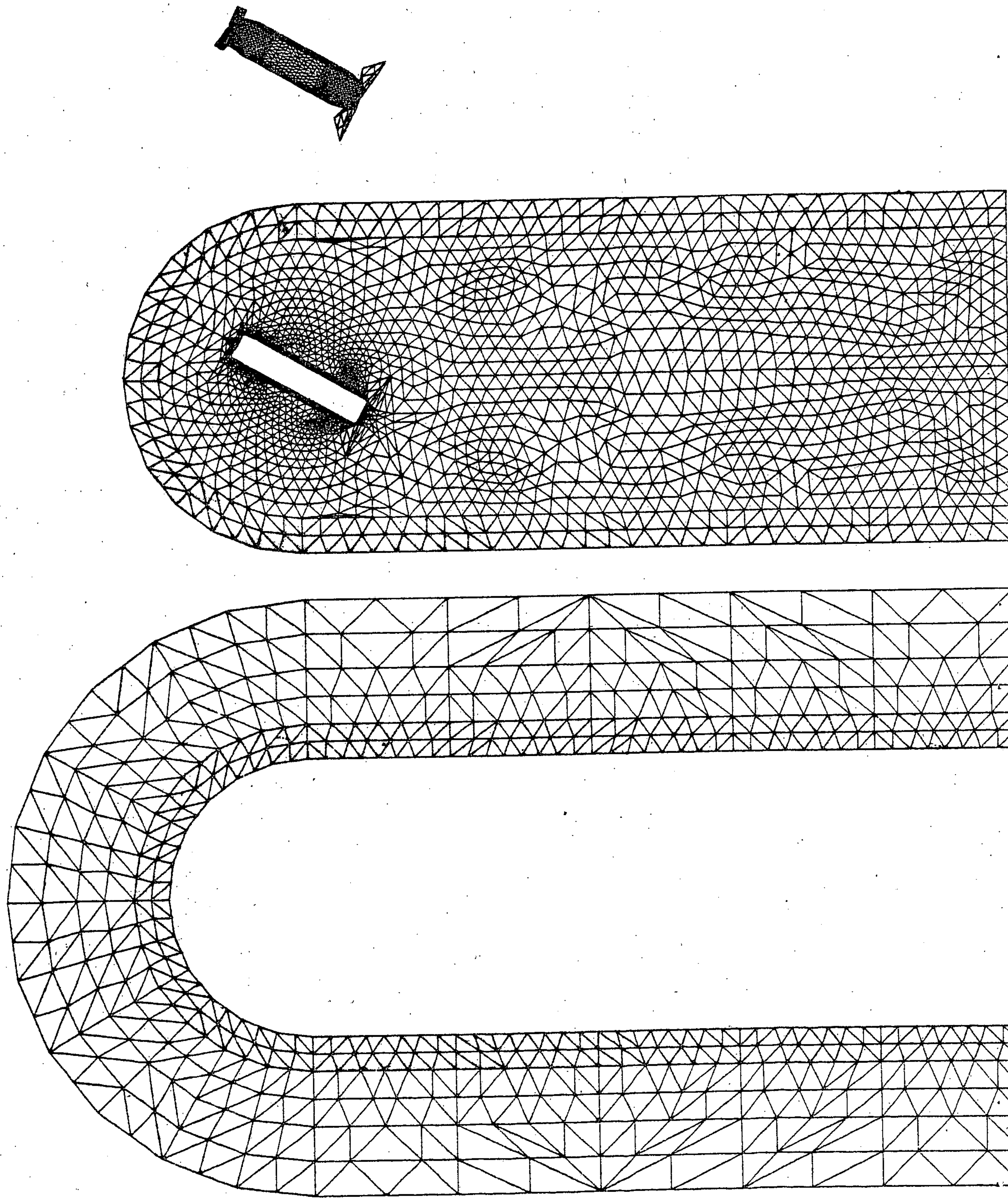


Figure 5.28

Velocity grids

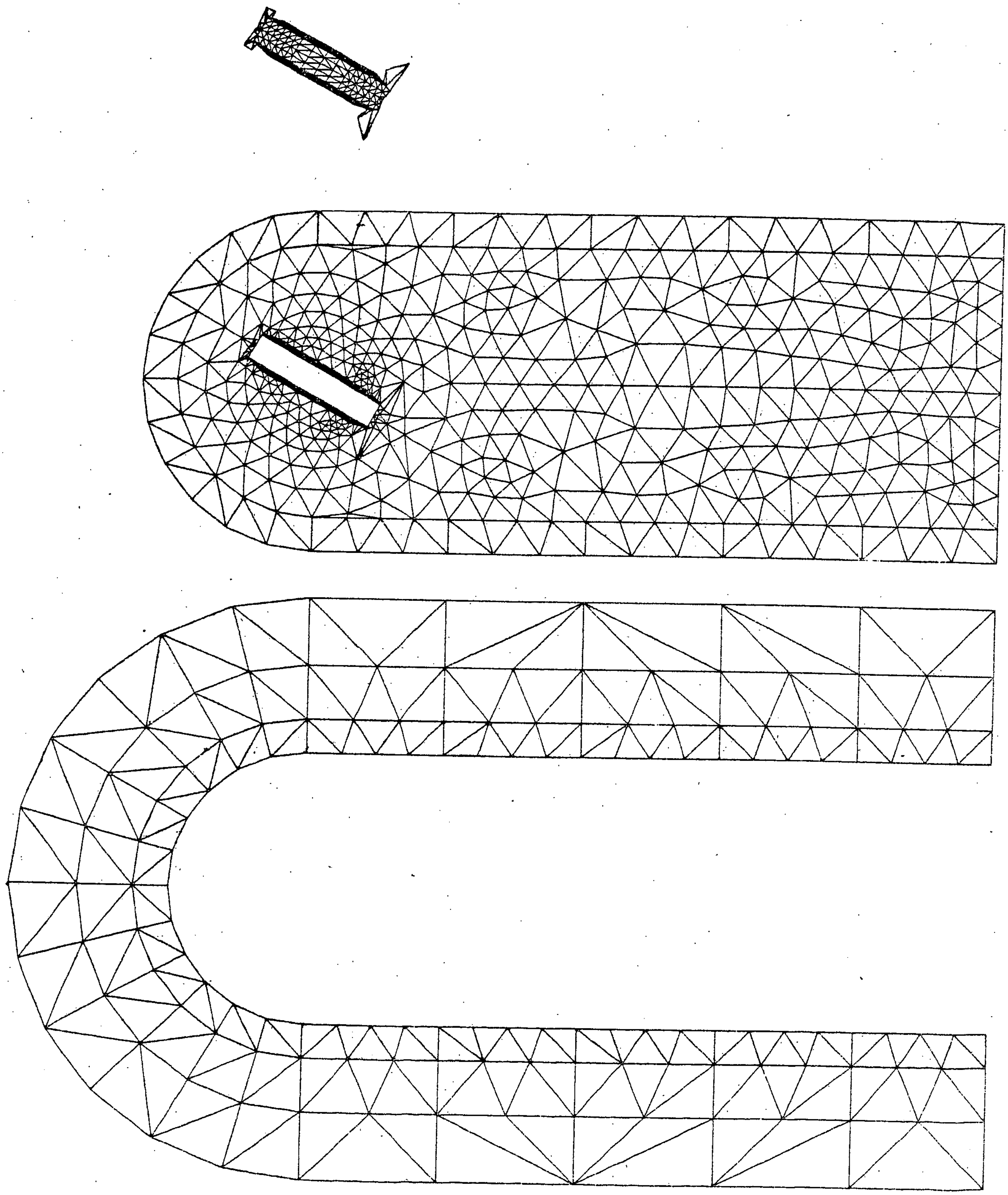


Figure 5.27

Pressure grids

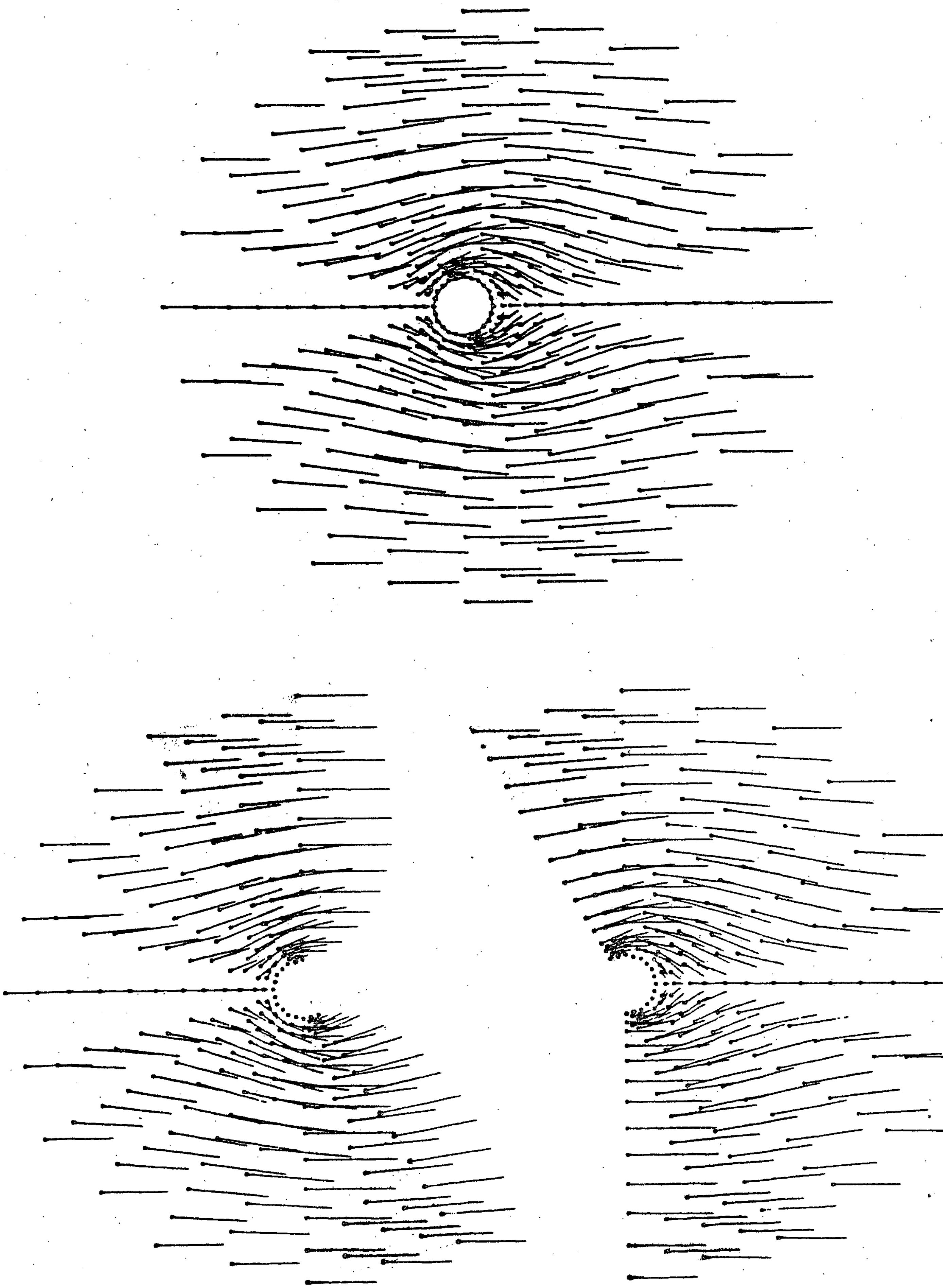


Figure 5.26

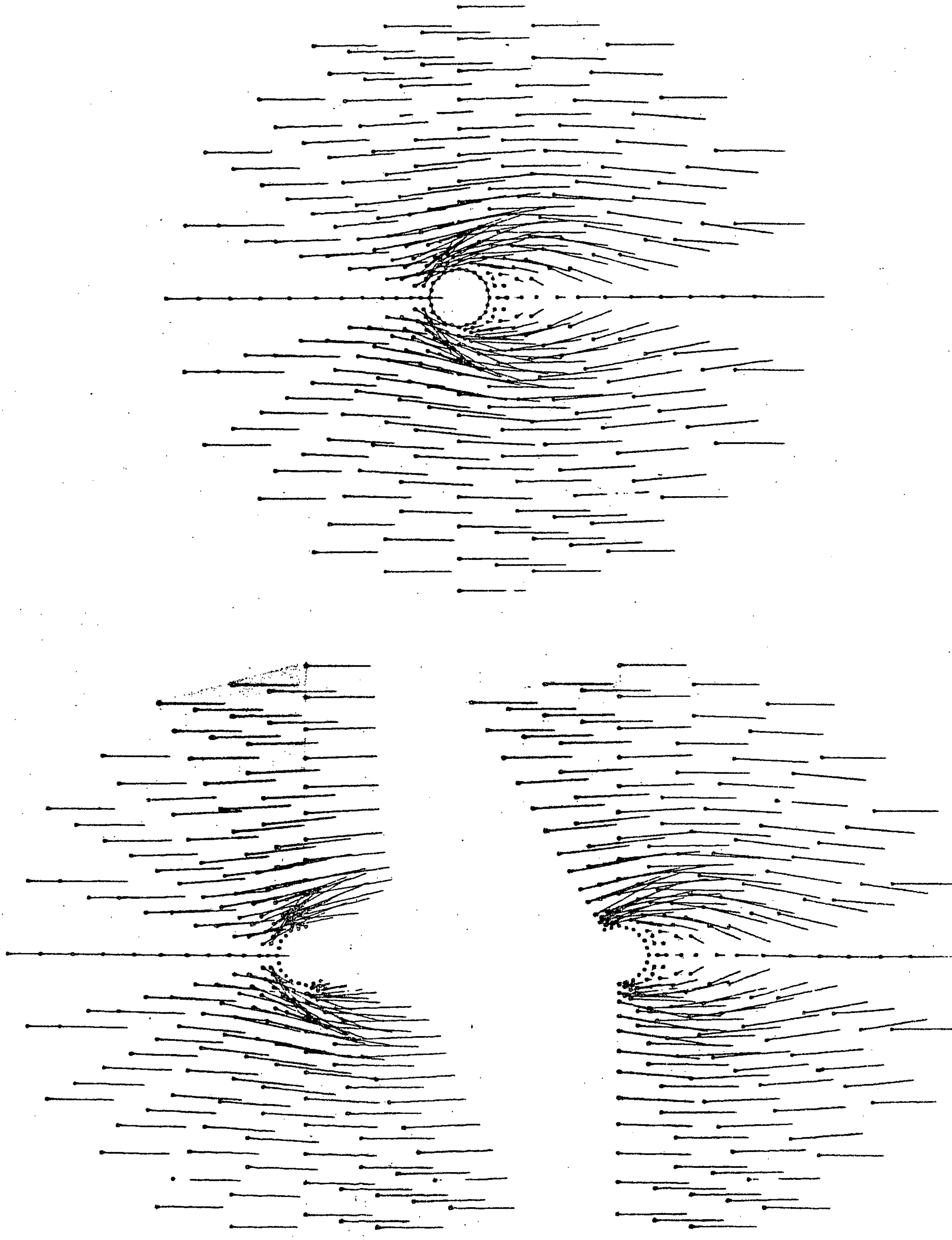
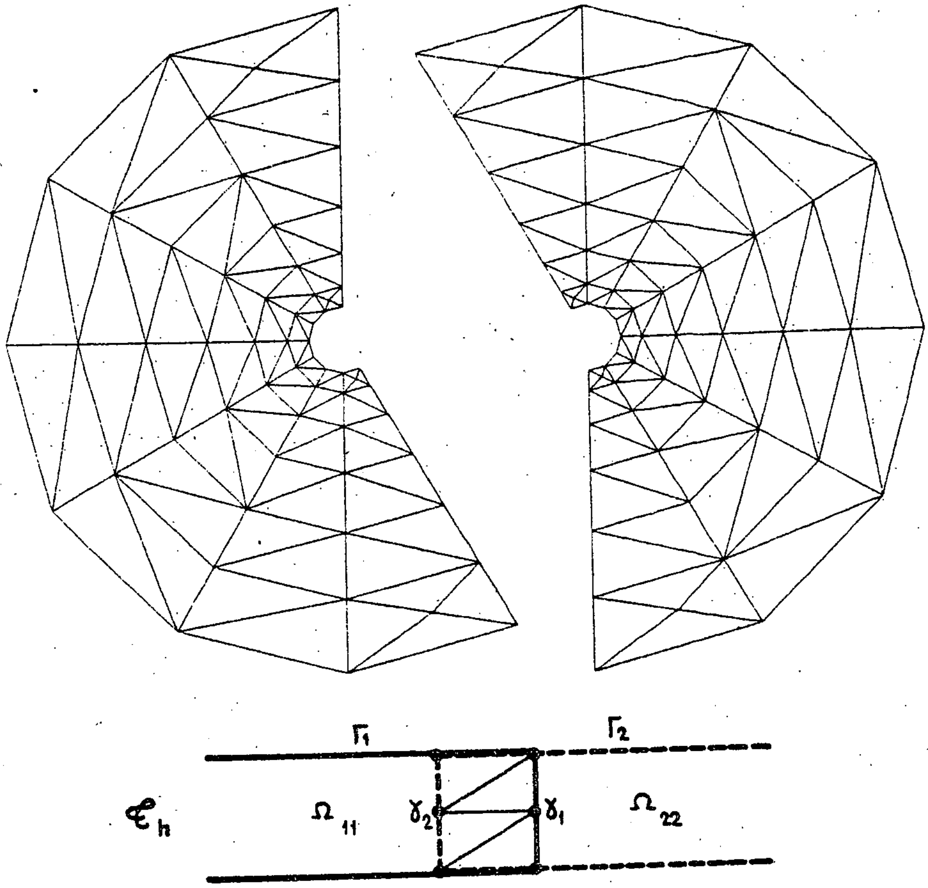
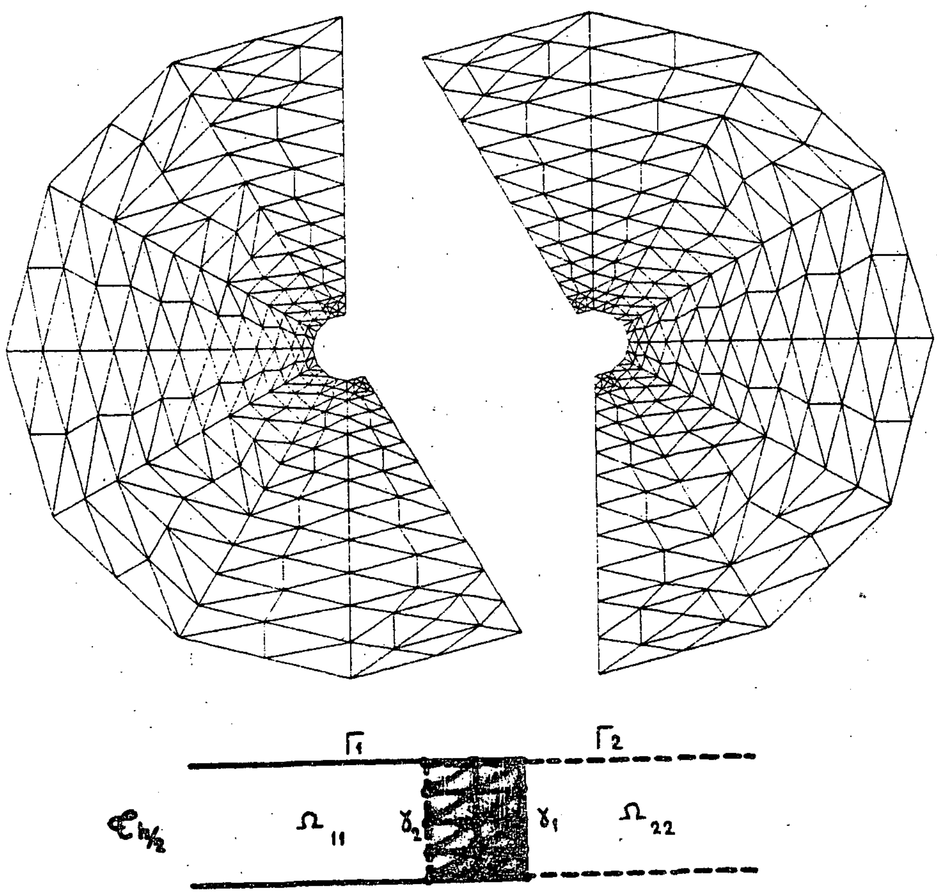


Figure 5.25



P1 _ PRESSURE GRID WITH OVERLAPPING



P1 _ iso P2 VELOCITY GRID WITH OVERLAPPING

Figure 5.24

are those of Sec. 4.2, combined with the domain decomposition methods with overlapping discussed in Sec. 2.3.

5.4.2.1. Flows around a circular cylinder.

We consider here steady flows around a circular cylinder, in fact we suppose that the flow is unvariant by translation parallel to the cylinder axis. From that simplification we consider only 2-D flows around a cross section of the cylinder, i.e. around a disk. Using the solution methods discussed in [1], [23],[31], we use a piecewise linear approximation for the pressure on a given finite element grid $\mathcal{T}_h$, and a piecewise linear approximation, also, for the velocity, but on a twice finer grid as shown on Fig. 5.24, which shows also the two overlappings subdomains Ω_{11},Ω_{22}. Fig. 5.25 (resp. 5.26) shows the velocity distribution at Re = 0 (Stokes flow) (resp. Re = 25). In fact Figs. 5.25, 5.26 show the flow in Ω_{11}, Ω_{22} and also the global flow. We observe the very good quality of the matching of the two local solutions ; the same observation holds for the local pressures and vorticities.

The results in this section have been obtained by sequential computing on IBM 3033 without using the AP's.

5.4.2.2. Unsteady flow around and inside a two-dimensional air intake.

We consider the unsteady flow of an incompressible viscous fluid around and inside an idealized two-dimensional air intake ; the angle of attack is 30 degrees. The Reynold number is 100, taking the distance between the two walls as reference. Velocity and pressure are approximated by the same finite element method than in the above Sec. 5.4.2.1. The domain of computation is bounded and splitted into 3 subdomains as shown on Fig. 5.27, 5.28. The velocity grids (resp. pressure grids) are shown on Figs. 5.27 (resp. 5.28) ; these grids have been obtained using the mesh generator of MODULEF library (cf. [32]).

The following results have been obtained using the computer system with AP's discussed in Sec. 5.3 ; actually 90% of the computing time is run in the AP's, saving thus computer time of the host machine. With $\Delta t = .1$ and using a Stokes solution as initial value we have solved the unsteady Navier-Stokes equations and represented several significant flow characteristics at various time steps.

Figure 5.29 (resp. 5.30) shows the streamlines (resp. pressure lines) at t=0, close to and inside the air intake, according to the domain decomposition. Figure 5.31-5.32, 5.33-5.34, 5.35-5.36, show the same quantities at t=10,40 and 60 respectively. We observe the creation and propagation of vortices, and again the good matching of the local solutions. For more details, see Q.V. DINH [33].

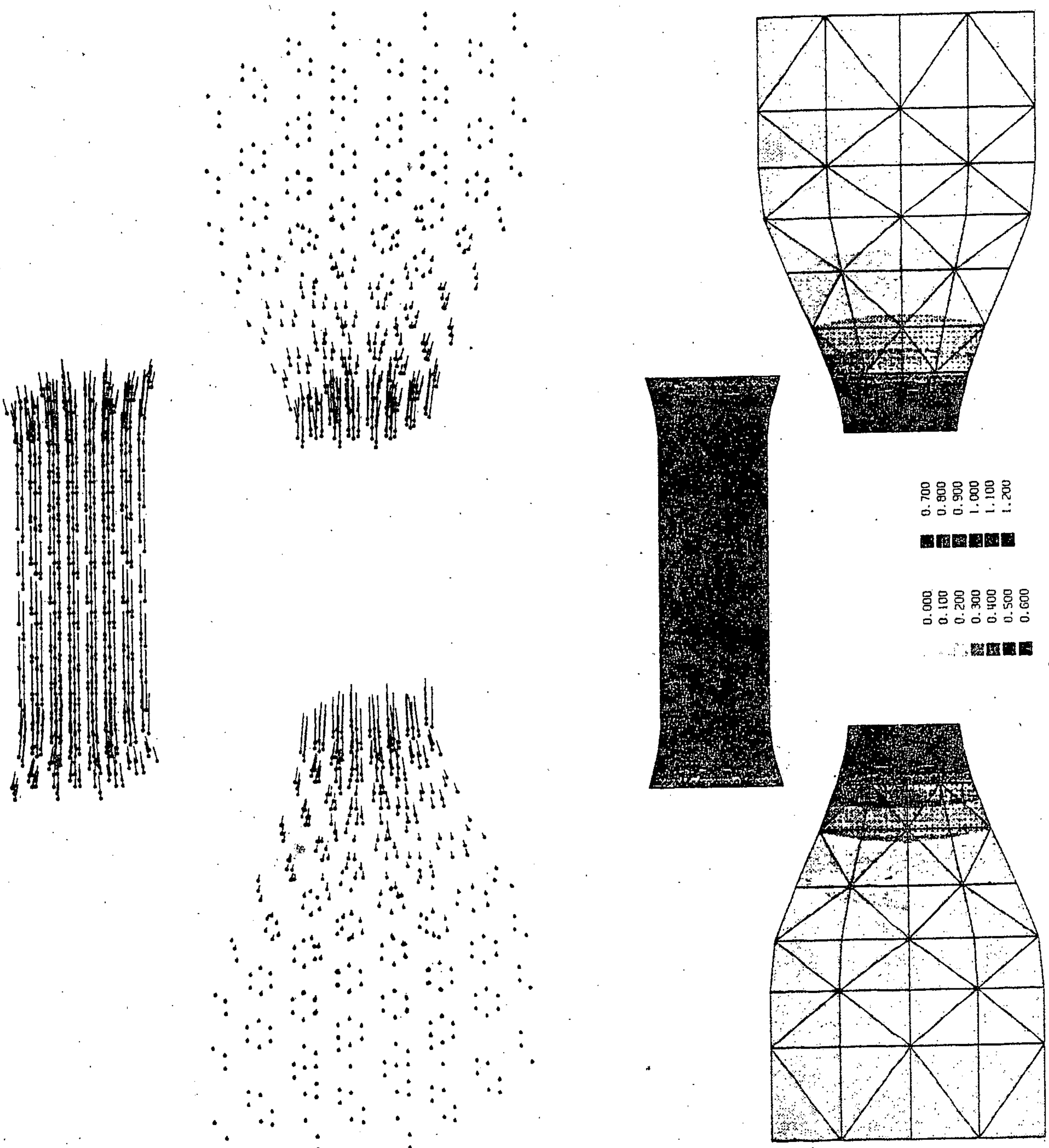

0.000
0.100
0.200
0.300
0.400
0.500
0.600
0.700
0.800
0.900
1.000
1.100
1.200

Figure 5.23

DOMAIN DECOMPOSITION

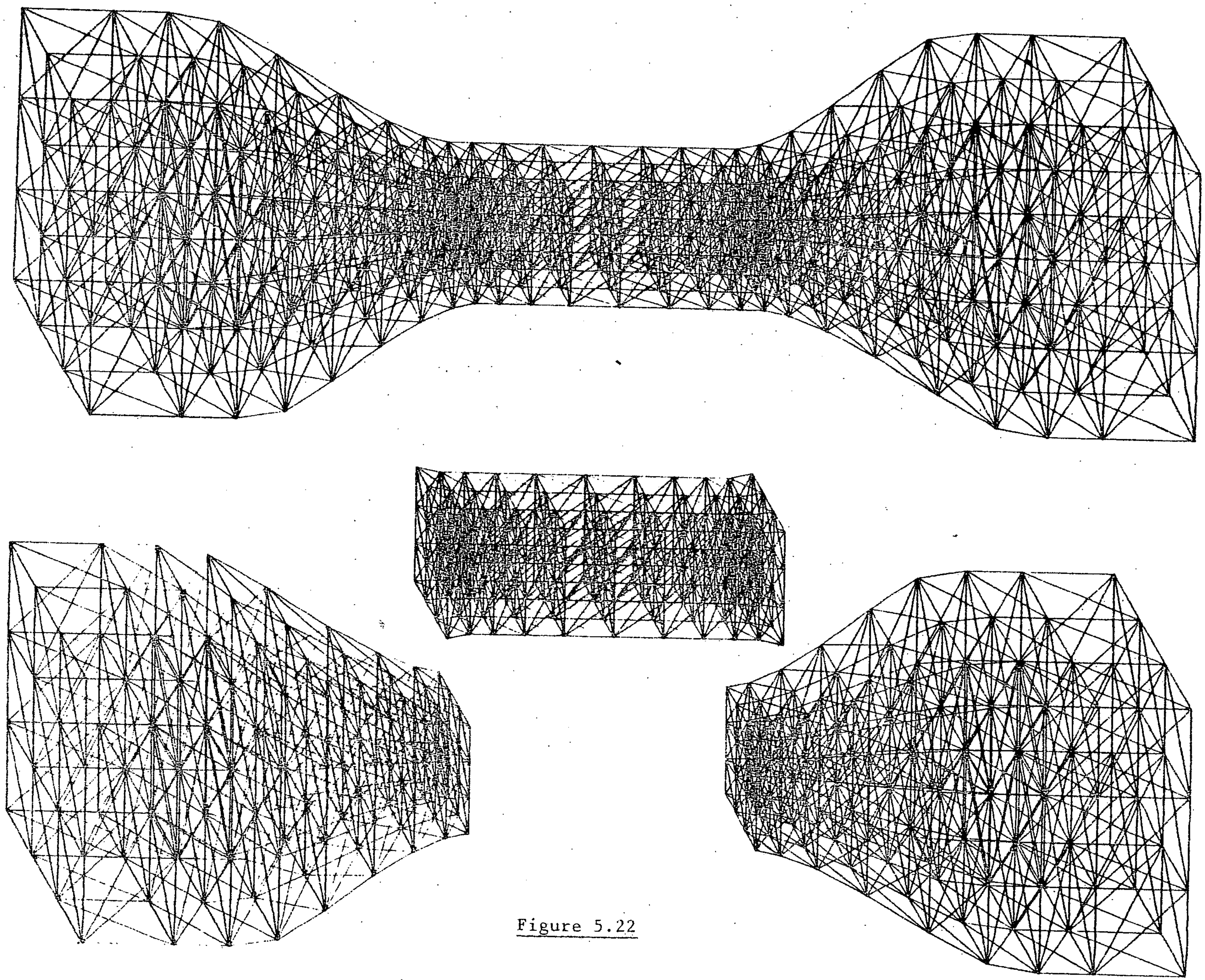
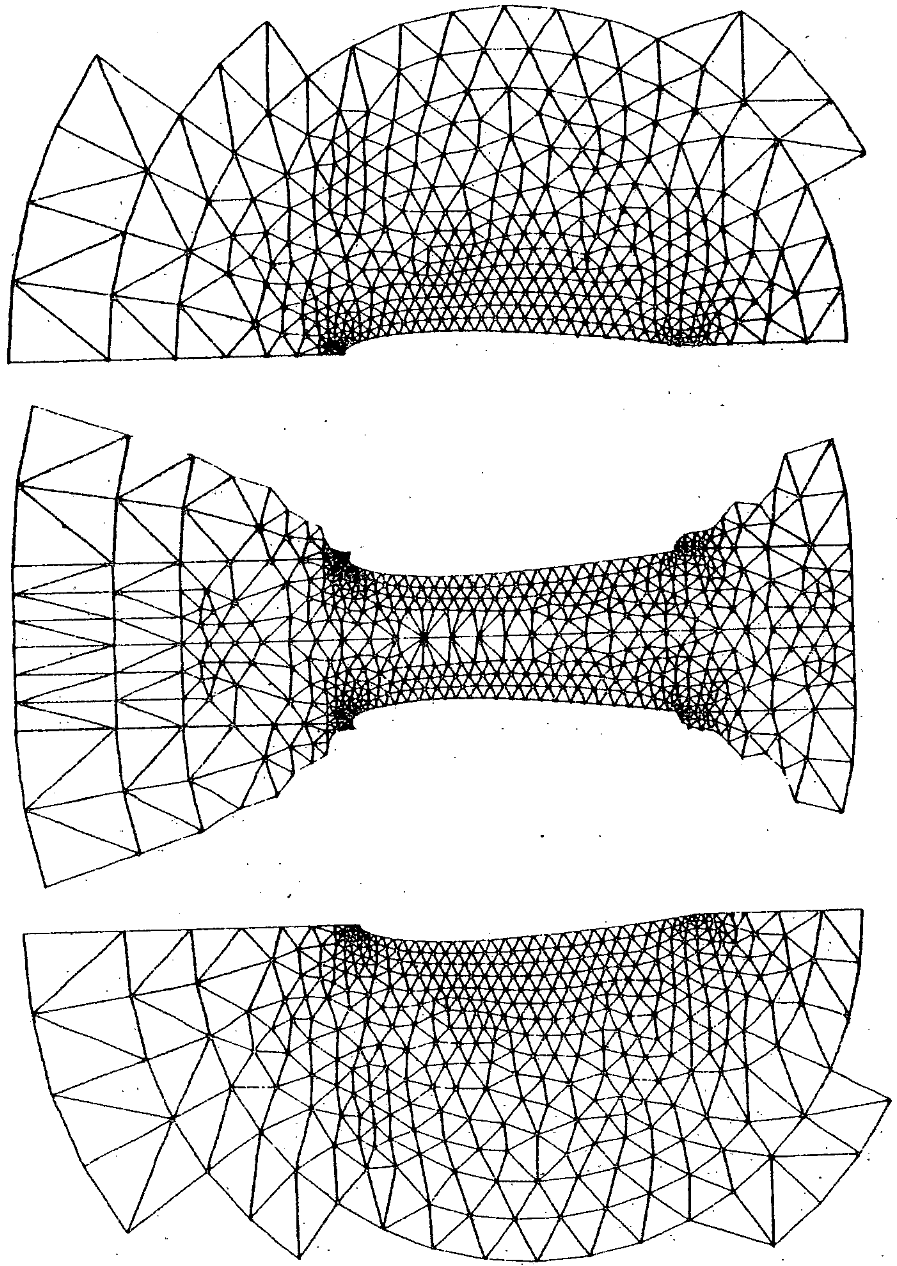


Figure 5.22



Domain decomposition around a two NACA 0012 airfoil

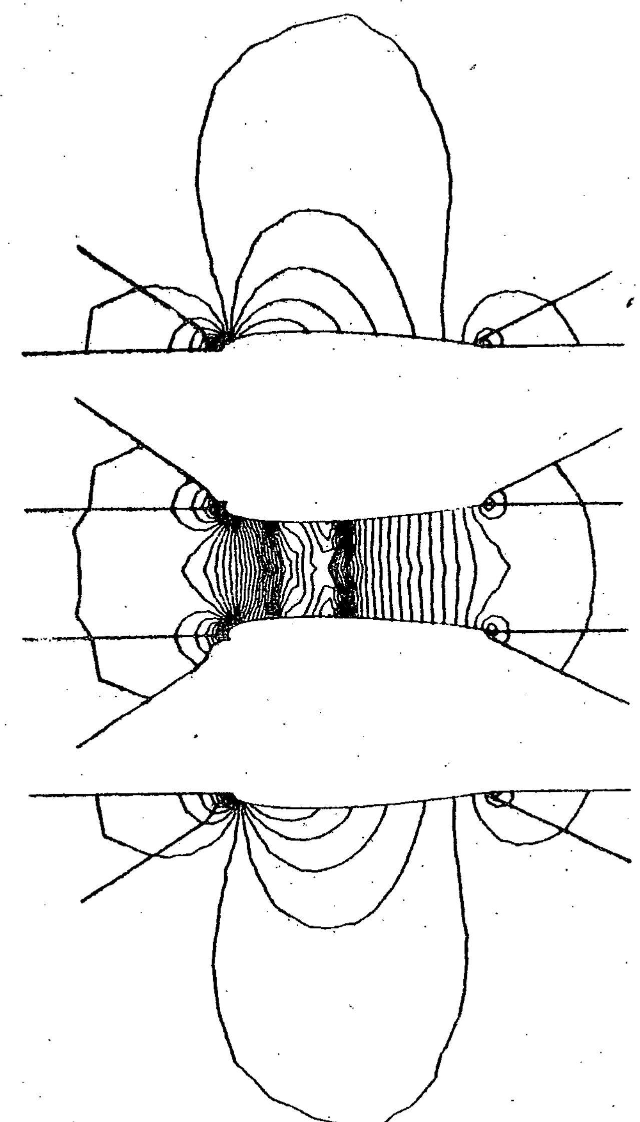
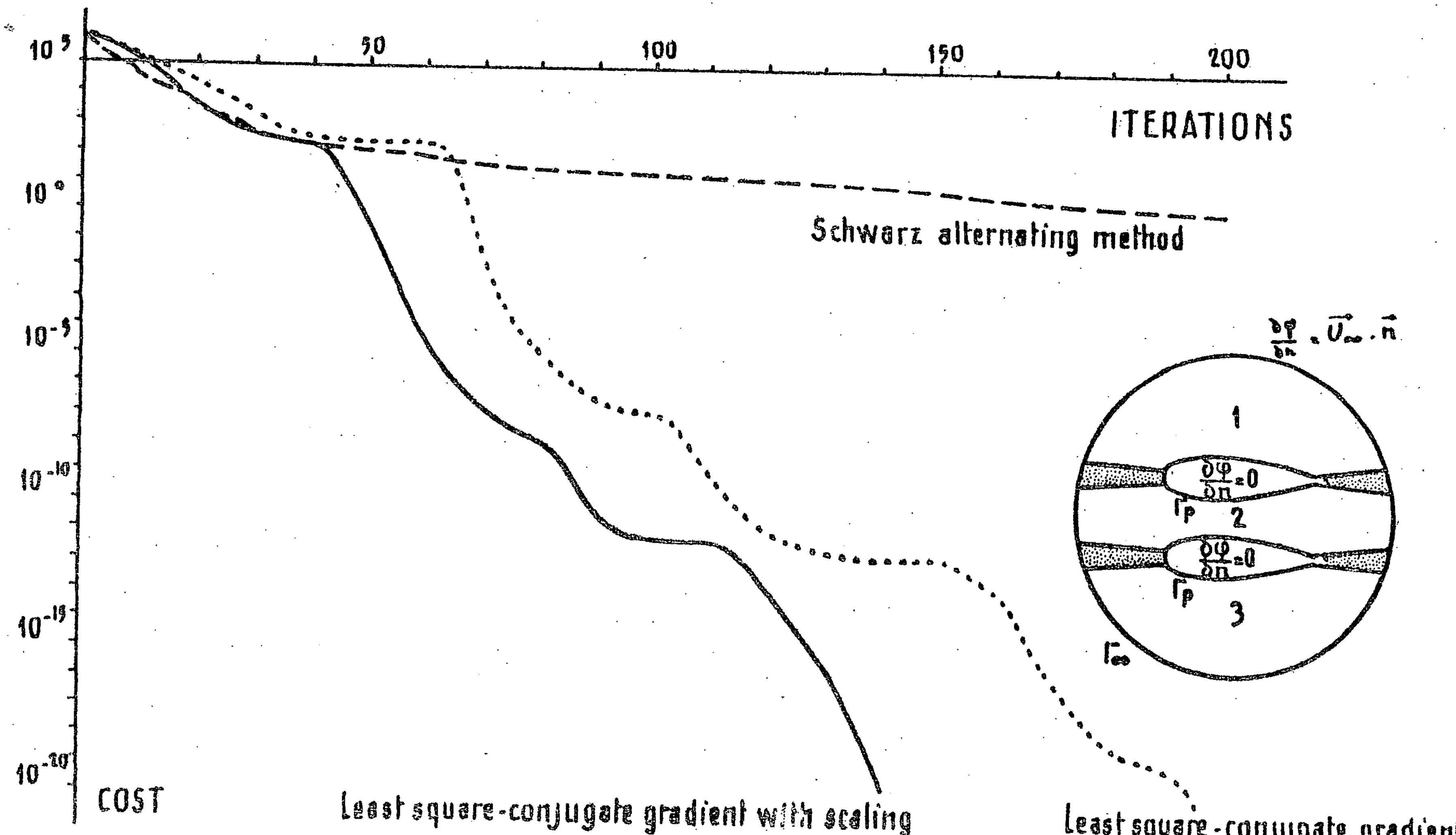


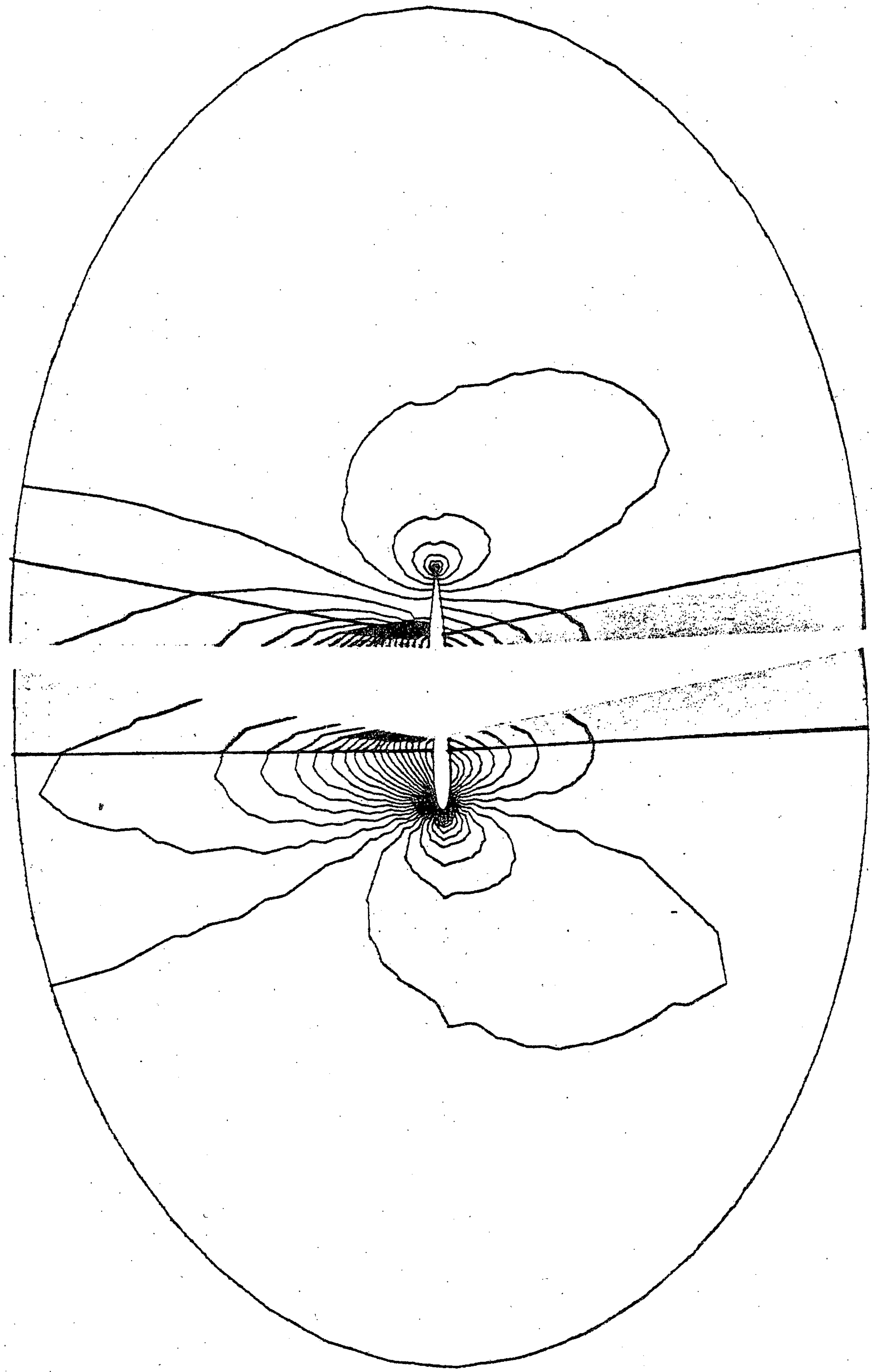
Fig. 5.21

Mach distribution around a two NACA 0012 airfoil



INFLUENCE OF THE NEUMAN BOUNDARY CONDITIONS

Figure 5.20



$M_\infty = .78 \quad \alpha = 1°$

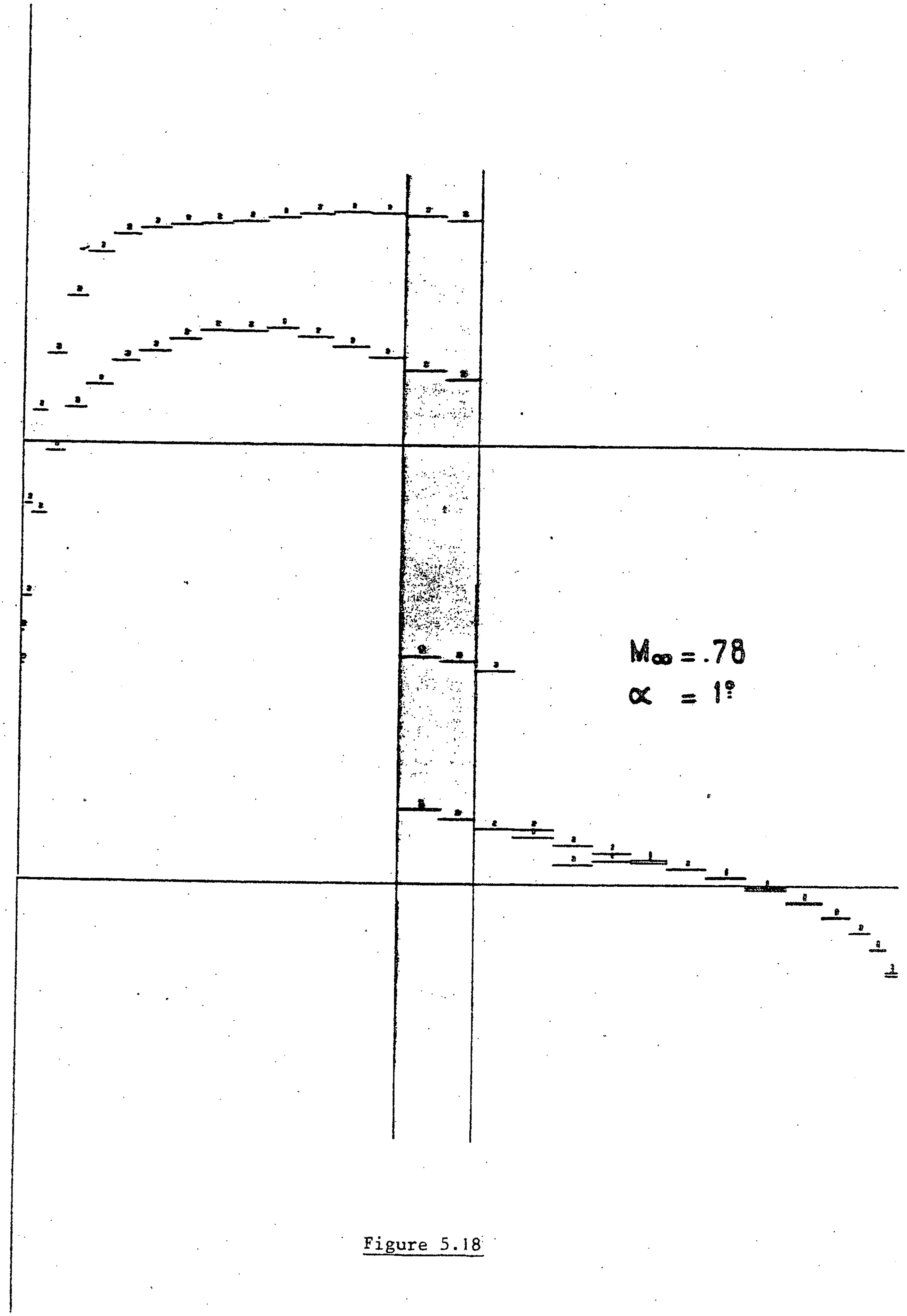
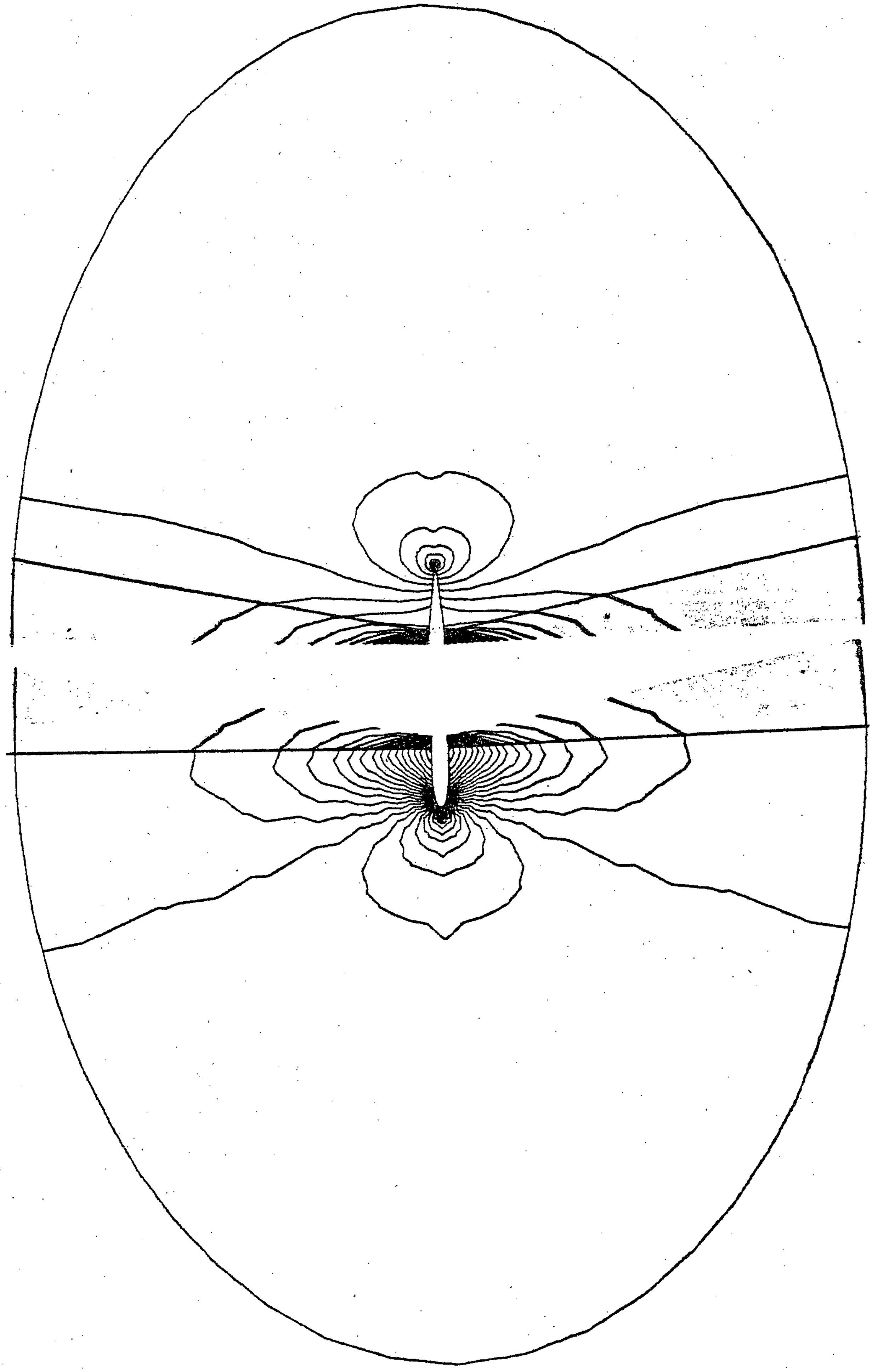


Figure 5.18



$M_\infty = .79 \quad \alpha = 0°$

Figure 5.17

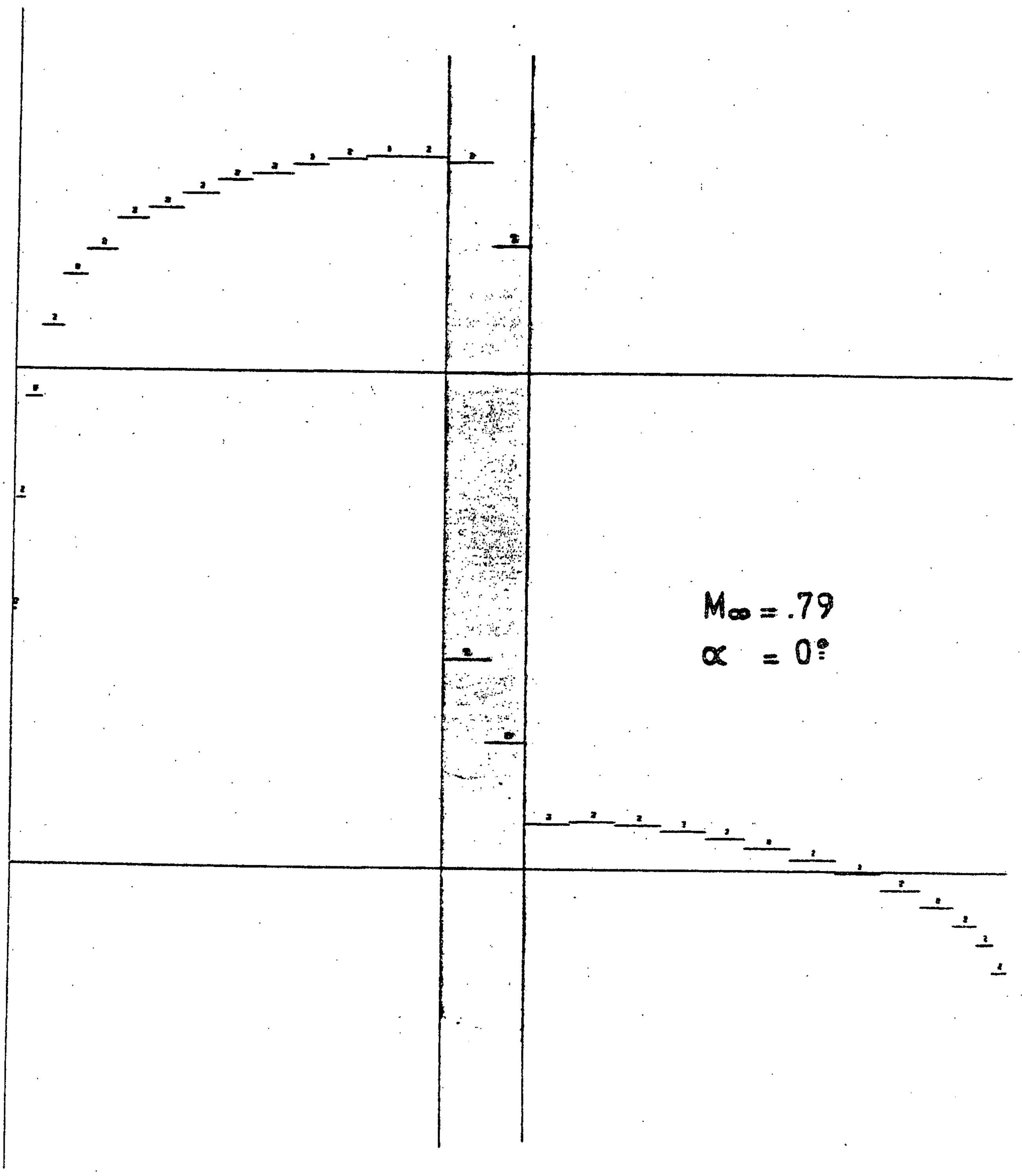


Figure 5.16

hence the third case corresponds to a lifting situation in which a Kutta-Joukowsky condition has to be satisfied. The domain decomposition and triangulation are still those of Figure 5.14. Actually M has been chosen in such a way that the shock occurs in the overlapping region ; a perfect location of the shock is obtained however. For these two test problems we have used the quasi-direct decomposition method of Sec. 2.3.8 as Neumann solver to precondition the nonlinear least squares conjugate gradient method solving the discrete transonic flow equation.

The pressure distribution on the airfoil and the isomach lines are shown on Figures 5.16, 5.17 (second test problem) and 5.18, 5.19 (third test problem).

5.4.1.2. Transonic flow around a two-piece airfoil.

As a second example we consider the simulation of a transonic flow around the two piece airfoil shown on Fig. 5.20 and consisting of two NACA 0012 airfoils whose axis are parallel. Fig. 5.20 shows also a comparison between the performances of several iterative solvers for the corresponding linear Neumann problems ; the domain decomposition uses three subdomains and is indicated on Fig. 5.20, also. The corresponding triangulations and overlapping triangles are shown on Fig. 5.21. The numerical tests prove again the superiority of the new method over the Schwarz method.

Using these decomposition methods as preconditioner we have solved a transonic flow problem around the above airfoil, taking $M_\infty = .54$ and a zero angle of attack ; the Kutta-Joukowsky condition has to be satisfied for both pieces. The Mach lines distribution in each subdomains is shown on Fig. 5.21, and the fitting of the local solutions on the overlapping regions is quasi perfect. We note a shock in subdomain 2.

5.4.1.3. Transonic flow in a three dimensional nozzle.

The same domain decomposition methodology has been used to compute a transonic flow inside the three dimensional convergent-divergent nozzle described on Fig. 5.22. Again the nonlinear least squares method of Sec. 4.1 has been used ; for simplicity we have not included any control of the entropy condition and this explain that an unphysical shock can be observed on Fig. 5.23 . However these preliminary results show the ability of our domain decomposition methods to handle complicated nonlinear three dimensional problems, and also that the existence of shocks in the overlapping regions is not troublesome for the computing process. The results of experiments, under investigation at the moment, in which the entropy condition is satisfied, will be presented elsewhere.

5.4.2. Navier-Stokes flow simulations.

We present in this section the results of numerical experiments concerning the solution of incompressible viscous flows modelled by the Navier-Stokes equations. The solution methods

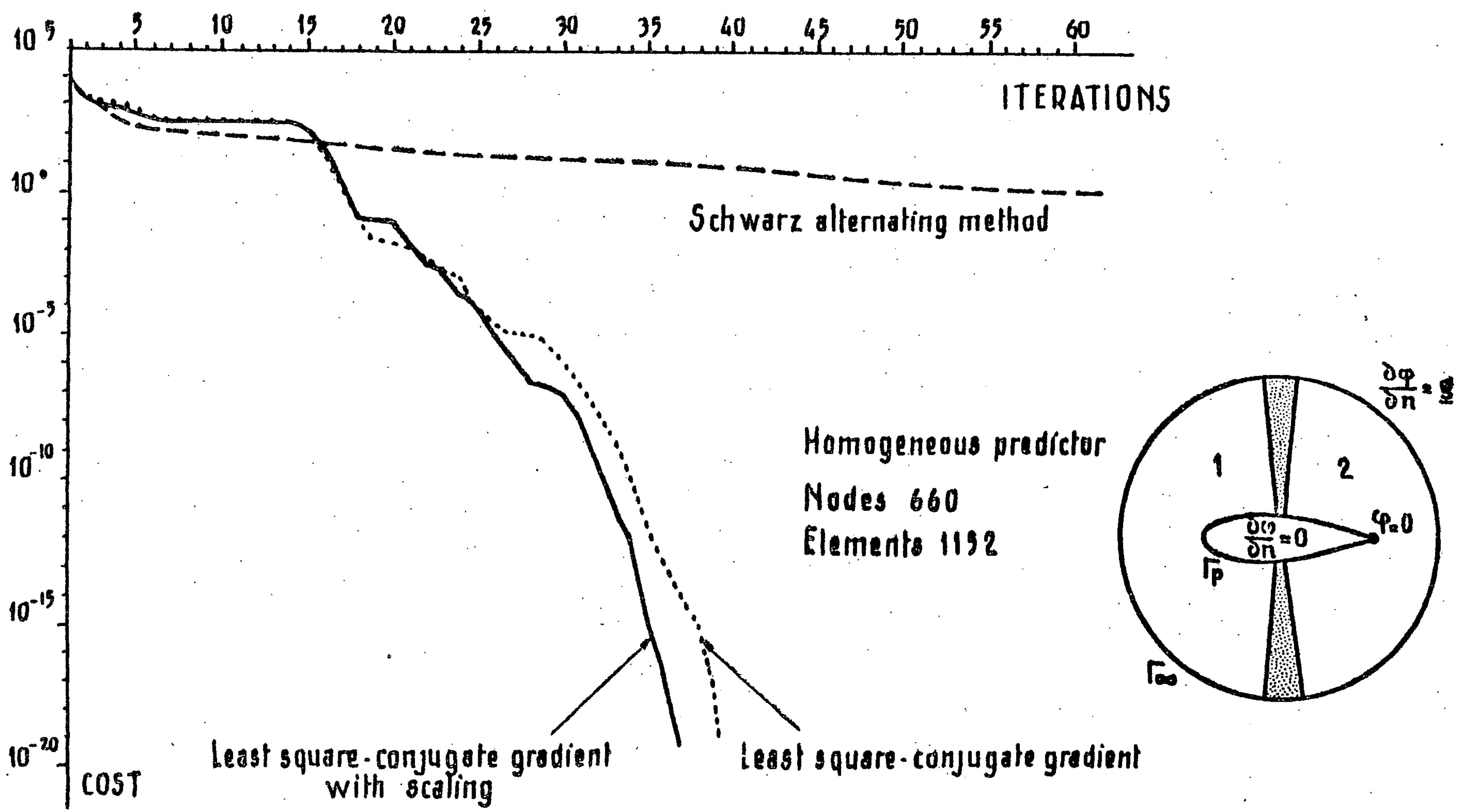


Figure 5.15

Comparison of iterative methods using domain decomposition applied to the solution of a linear Neumann problem around an airfoil NACA 0012.

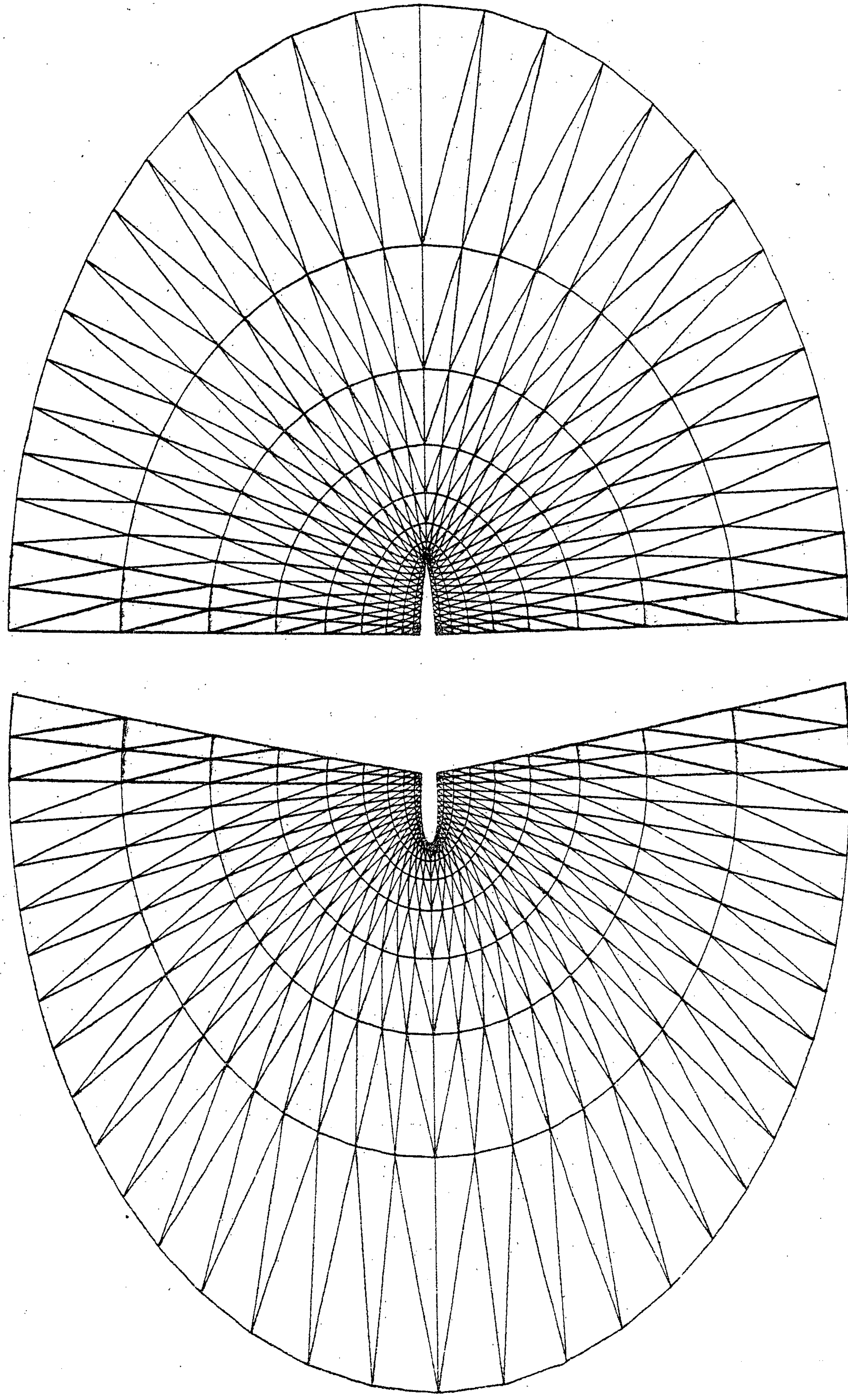


Figure 5.14
Domain decomposition around a NACA 0012 airfoil.

SUB-DOMAIN LAPLACIAN SOLVER BY A QUASI-DIRECT METHOD FOR DIFFERENT NUMBERS OF RIGHT HAND SIDES

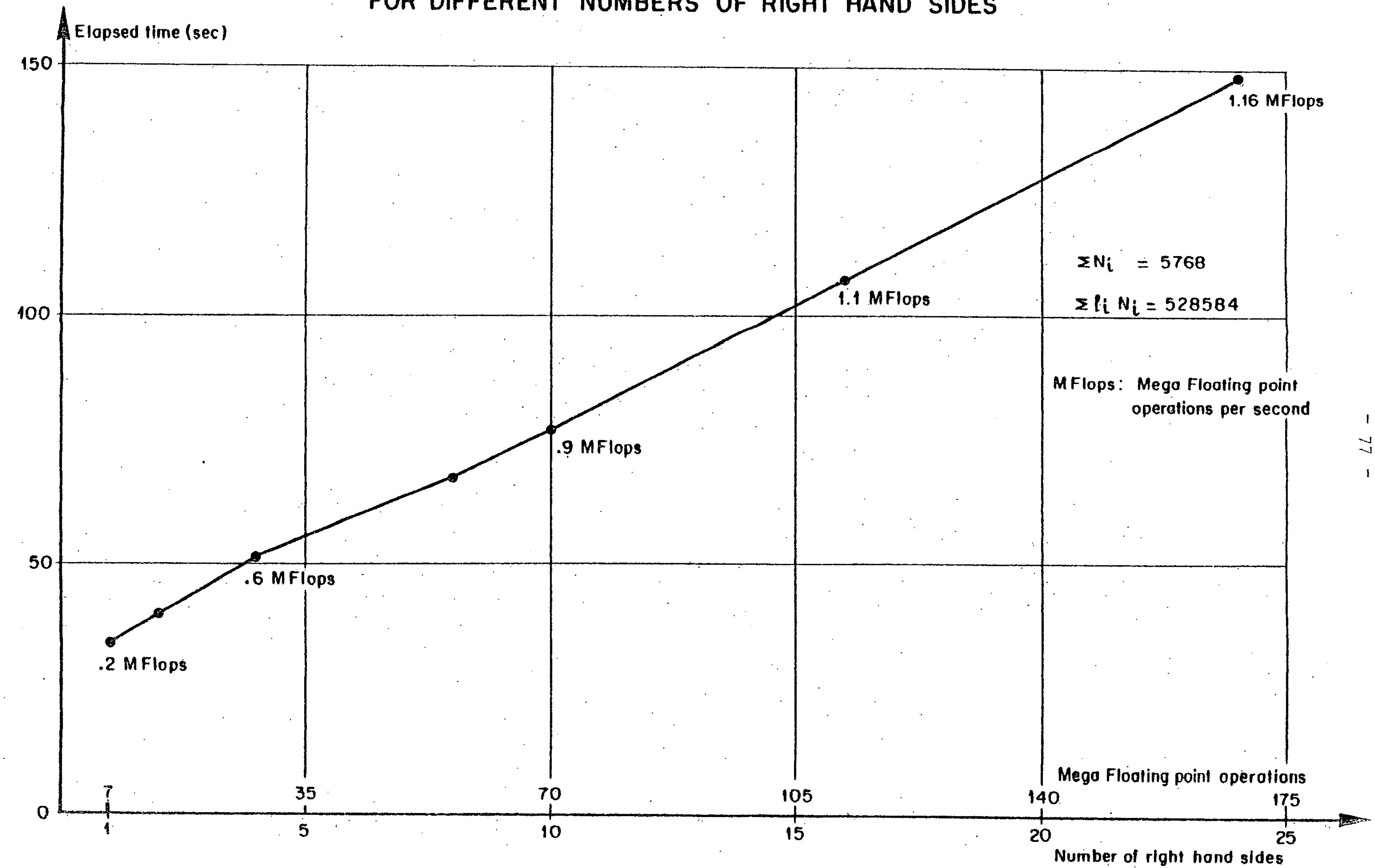


Figure 5.13

local state and adjoint state equations (resp. only local adjoint state equations whose right hand sides contain information about the traces of the solution of the local state equations).

2) In both cases the CPU time of the host system is small compared to the elapsed time run in the AP's with or without data flow.

Method 1 and also the method described in Sec. 5.2.2 have been used to compute potential flows in the subdomains shown in Fig. 5.9. The performances of Method 1 implemented on the parallel architecture of Figure 5.9 is shown on Fig. 5.13 ; we have indicated the elapsed time (measured in Megaflops) versus the number of right hand sides. We observe from Fig. 5.13 that data flow via channels deteriorates the performances of the AP's , however a gain of efficiency can be obtained if pipelining is increased.

It follows from the above observations that in more general cases it will be very important to prefer algorithms whose parallelism is founded on high ratio of computational task per input or output instruction, if one uses a system like the one of Fig. 5.9.

5.4. Solution of nonlinear problems of Fluid Dynamics using domain decomposition techniques.

5.4.1. Transonic flow simulations.

5.4.1.1. Transonic flow around a NACA 0012 airfoil.

As a first example we consider the numerical simulation of a transonic flow like those discussed in Sec. 4.1 around a NACA 0012 airfoil. The triangulation of the computational domain and the corresponding triangulation are shown on Figure 5.14 with the region of overlapping. Since the least squares conjugate gradient solution of the transonic flow problem by the methods of Sec. 4.1 uses a discrete Neumann solver as preconditionner we have tested several such Neumann solvers using the above domain decomposition ; we have solved hence the (linear) Neumann problem

$$\begin{cases} \Delta\phi = 0 \ \underline{\text{in}}\ \Omega, \\ \dfrac{\partial\phi}{\partial n} = g \ \underline{\text{on}}\ \partial\Omega \ (\underline{\text{with}} \displaystyle\int_{\partial\Omega} g \, d\Gamma = 0) \end{cases}$$

by the Schwarz alternating method and the conjugate gradient algorithms of Secs. 2.3.6 (without preconditioning) and 2.3.7 (with preconditioning). The Schwarz method is definitely less efficient when applied to the solution of this Neumann problem than the methods introduced in this paper and which behave very similarly as shown on Figure 5.15.

The second and third test problems are genuine transonic flow problems. In the second (resp. third) test problem we consider a transonic flow around the NACA 0012 airfoil at $M_\infty = .79$ (resp. $M_\infty = .78$) with an angle of attack $\alpha=0°$ (resp. $\alpha=1°$) ;

BLOCK CONSTRUCTION VIA PARALLELISM ON IBM 3033 + 2 AP

- **IN CORE (AP)**

  LOCAL INCOMPLETE CHOLESKI METHOD USED

- $A_i = L_i L_i^T$ and $\tilde{A}_i = \tilde{L}_i \tilde{L}_i^t$

  ONCE FOR ALL IN 3033

- $J' = A_i X_i^h - b_i$ and $\tilde{A}_i^{-1} J'$

  COMPUTED IN LANGUAGE MACHINE APAL

103	44	51	0	0	0
0	99	0	43	0	0
0	0	87	0	73	0
0	0	0	77	0	51
0	0	0	0	117	52
0	0	0	0	0	102

$\Sigma = 990$ s

30 mn for 3 mn CPU

1-column pipelined in each processor with parallelism

ELAPSED TIME VIA HOST SYSTEM

Figure 5.12

BLOCK CONSTRUCTION VIA PARALLELISM AND PIPELINE ON IBM 3033 + 2 AP + DISKS

- OUT OF CORE (AP) LOCAL CHOLESKI METHOD USED

- $A_i = L_i L_i^T$ ONCE FOR ALL IN 3033

$\Sigma = 967$ s

116	65	65	O	O	O
O	89	O	54	O	O
O	O	105	O	60	O
O	O	O	92	O	57
O	O	O	O	95	56
O	O	O	O	O	113

10-columns pipelined in each processor with parallelism

$\Sigma = 772$ s

20 mn for 1 mn30 CPU

93	53	50	O	O	O
O	74	O	43	O	O
O	O	85	O	46	O
O	O	O	75	O	42
O	O	O	O	77	41
O	O	O	O	O	93

24-columns pipelined in each processor with parallelism

ELAPSED TIME VIA HOST SYSTEM

Figure 5.11

3.D DOMAIN DECOMPOSITION AROUND A WING

A-INTEGRAL COORDINATION
SYMMETRIC SPARSE MATRIX

11	12	13	0	0	0
	22	0	24	0	0
		33	0	35	0
			44	0	46
				55	56
					66

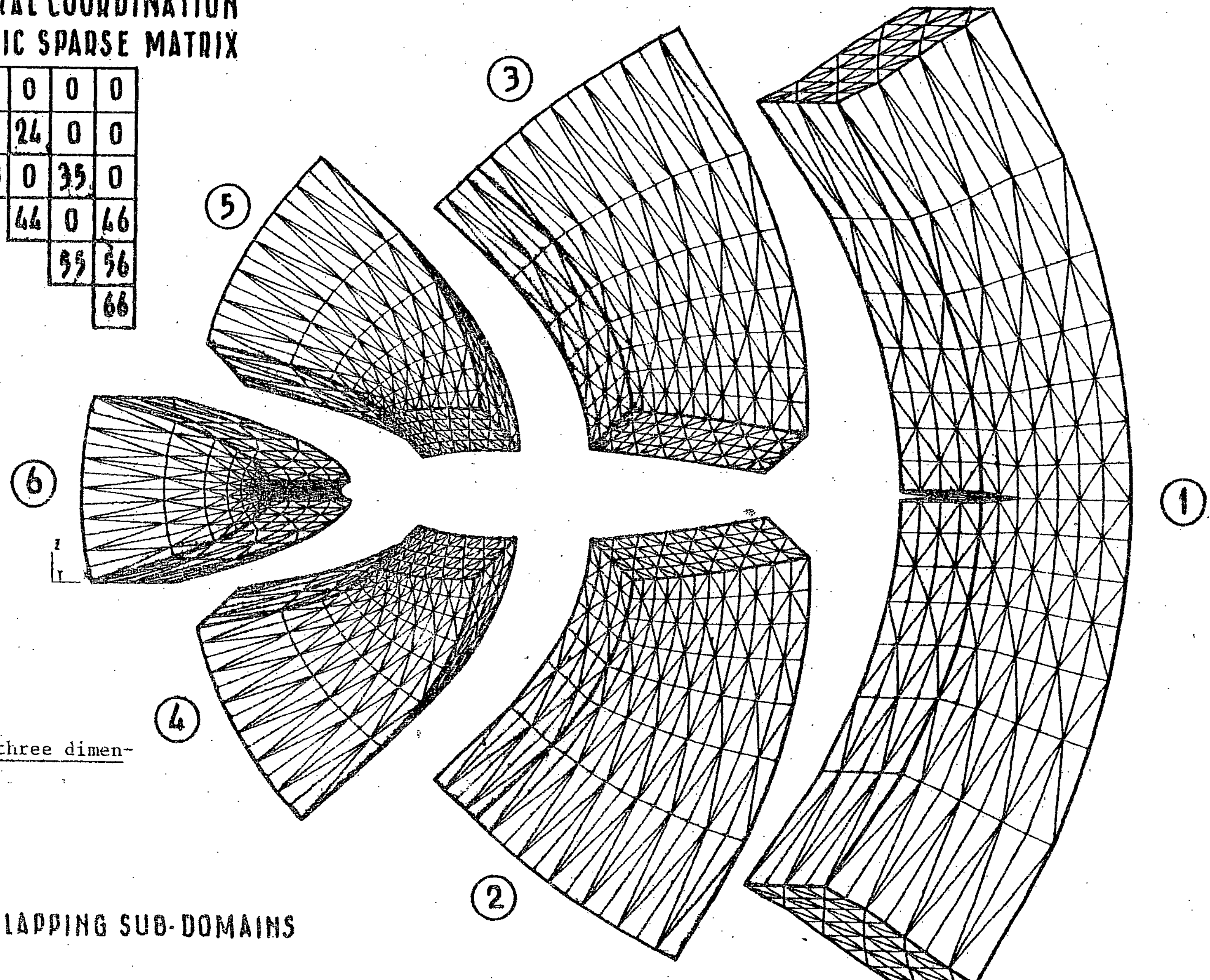


Figure 5.10

Splitting of a three dimensional domain.

OVERLAPPING SUB-DOMAINS

5.3.3. Application to the solution of a three dimensional test problem : Incompressible potential flow around a portion of wing.

The test problem that we consider here corresponds to the potential flow of an inviscid, incompressible fluid around a portion of wing ; such a flow is modelled by equation (5.1) with mixed boundary conditions. The computational domain has been splitted into 6 overlapping subdomains , as shown in Figure 5.10 which shows also the structure of the symmetric boundary operator A_h. The finite element approximation uses about 6000 degrees of freedom.

From the large size of the blocks of A_h, parallelism and pipelining are also used inside the blocks processed by the AP's. Two methods have been used for solving the local Poisson problems in each Ω_{ii} :

(i) With Method 1 (which is a direct method) one solves the Poisson problems in each Ω_{ii}, using a Cholesky factorization done once and for all in the host machine ; the Cholesky factors are stored in the disks of the AP's at the beginning of the computation, and then called via the PIOP's each time it is necessary. The backward/forward substitution is programmed using the VFC traductor. With this method we can also use pipelining to construct several columns of a given blocks of A_h simultaneously.

Two examples,related to the construction of A_h by the above techniques,using parallelism with the two AP's and pipelining, are shown on Fig. 5.11 ; the gain obtained by increasing pipelining is obvious from the numbers indicated on Fig. 5.11,on which we have also indicated the elapsed time (time for computation and data flow in the AP's and the disks) corresponding to each case.

(ii) With Method 2 (which is iterative) one solves the Poisson problems on each Ω_{ii} by a conjugate gradient algorithm using incomplete Cholesky factors as preconditioners (see [ ] for more details) ; the above incomplete Cholesky factors can be stored in the AP's without using disks.

Selected repetitive tasks such as computation of the residuals and backward/forward eliminations are coded using APAL while the remaining instructions of the algorithm are programmed using the VFC traductor. Figure 5.12 illustrates the construction of A_h by Method 2 using parallelism inside the blocks, only. The elapsed time indicated in each block look very similar to those in Fig. 5.11.

Two common features of the above methods 1, 2 can be pointed out from Figures 5.11, 5.12 :

1) It is more time consuming to construct the diagonal blocks than the off-diagonal ones. This follows from the fact that diagonal blocks (resp. off-diagonal blocks) require the solution of the

these tasks are <u>completely</u> processed by the host machine.

(ii) <u>Construction</u> - by the AP's - and <u>factorization</u> - by the host machine - of the discrete boundary operator A_h occuring in (2.210) ; the factorization is done by the host machine since double precision 64 Bits words are available in it, compared to the 38 Bits words of the AP's (38 Bits is unsufficient for the factorization step in such large and badly conditioned problems). Concentrating on the construction of A_h in the AP's we can do the following comments :

From the block sparsity and symmetry of A_h (see Fig. 2.6) we shall use a block partition process to construct this last matrix. This construction process uses the distinction between the diagonal and off-diagonal blocks of A_h ; acting so, these blocks can be constructed independently one to each other, and therefore <u>in parallel</u>. Furthermore since the columns of each block can also be constructed independently one to each other, we can take advantage of the <u>vectorization</u> or <u>pipelining</u> internal possibilities of each AP. According to the size of each block, compared to the storage possibilities of the AP's, several levels of parallelism can be considered ; for example, a first level in which the elementary objects to be processed in parallel (by the AP's) are the entire block of A_h ; on the other hand a second level - corresponding to the internal parallelism and pipelining of the AP's - in which the elementary objects are <u>parts</u> of the above blocks of A_h.

(iii) Construction of the right hand side b_h of (2.210) ; the various comments about the parallelism and the pipelining in the computation of A_h still apply for the computation of b_h.

(iv) From Step (ii) we have constructed $\mathcal{L}_h$ such that

(5.2) $$A_h = \mathcal{L}_h \mathcal{L}_h^T .$$

From (2.210) we have to solve the two triangular systems

(5.3) $$\mathcal{L}_h Z_h = b_h ,$$

(5.4) $$\mathcal{L}_h^T \Lambda_h = Z_h .$$

This can be done either in the AP's or in the host machine.

(v) From Λ_h we obtain the restriction y_{ih} on each Ω_{ii}, of the global solution y_h, by solving a <u>local</u> discrete Poisson problem on each Ω_{ii} ; these local tasks are done in the AP's. Again one can take advantage of parallelism <u>and</u> pipelining for these calculations.

The above methodology has been used to solve the test problem discussed in the following Sec. 5.3.3.

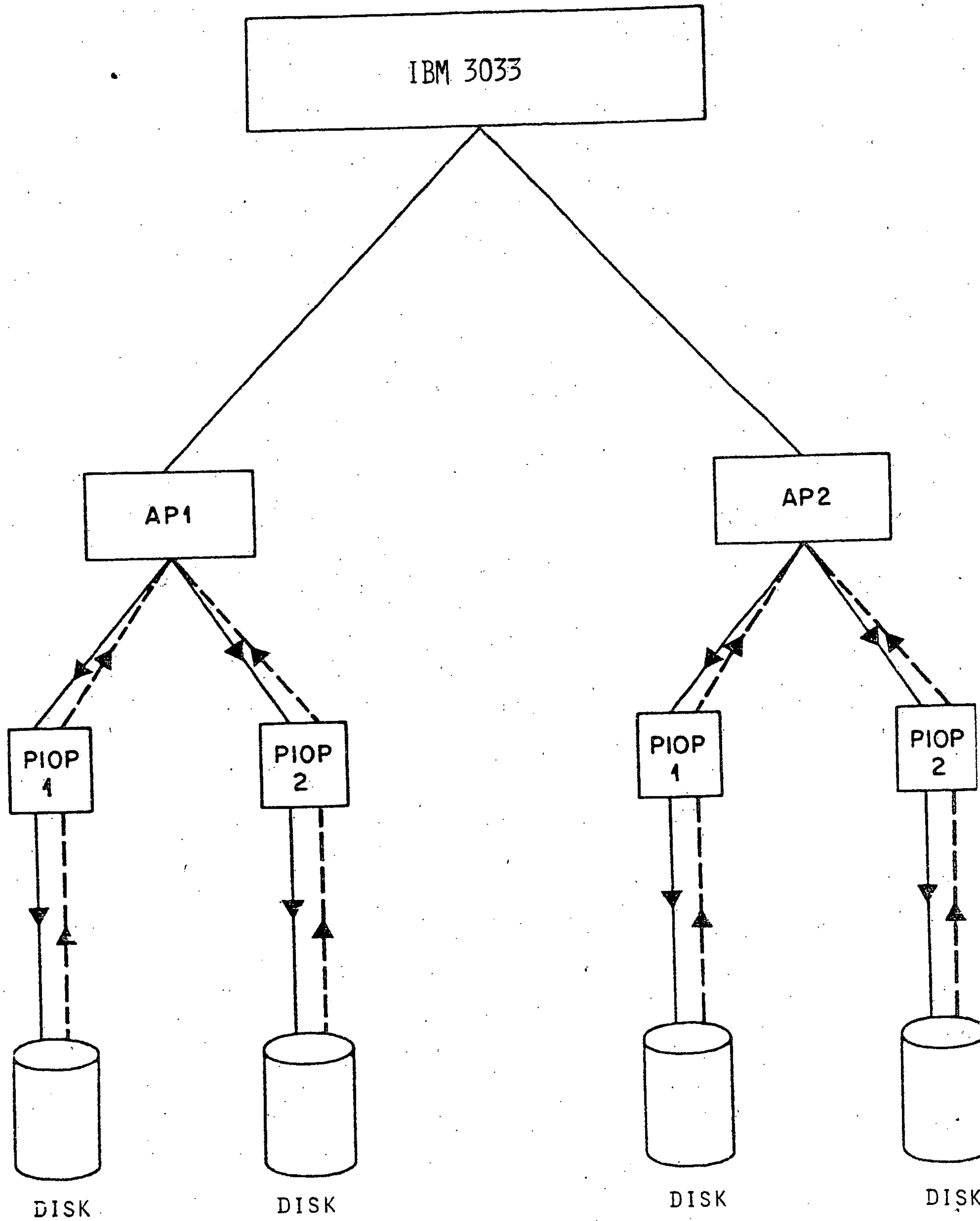


Figure 5.9

Description of the parallel architecture

Fig. 5.7 shows a comparison between the performances of the saddle-point algorithm (2.47)-(2.55) and the least square conjugate gradient with preconditioning algorithm (2.115)-(2.123) ; the second algorithm appears to be much more efficient. In our opinion this is due to the fact that for algorithm (2.47)-(2.55) the sub-domain problems to solve at each iteration on domain 2 are fully Neumann problems.

Fig. 5.8 shows a comparison between the Schwarz alternating algorithm (2.16)-(2.18), the least square conjugate gradient algorithm without scaling (2.65)-(2.73) and the least square conjugate gradient algorithm with scaling (2.115)-(2.123). Both least-squares conjugate gradient algorithms appear to be again much more performant.

5.3. Implementation of the quasi-direct method on array processors.

5.3.1. Description of the parallel architecture.

We have described on Figure 5.9 a computer system consisting of two array processors Floating Point FPS 190L, connected via channels to a host machine IBM 3033 ; this system is able to perform repetitive large vectorized floating point operations in parallel. Each array processor (AP) has a limited core memory of 128K-words of 38 Bits length ; it can also use auxiliary disks, monitored by Programmable Input Output Processors (PIOP), as shown on Fig. 5.9. Data flow between disks and processors is managed by the PIOP's independently of the host system.

The limited core memory of each AP makes impossible the storage of the full computing program in these processors ; in fact the main part of this program is stored in the host system memory, and only repetitive routines such as, for example, solution of local Dirichlet problems, construction of the right hand sides,etc., are implemented on the AP's, in parallel.

In most cases the program is coded in standard FORTRAN language and a traductor (called here VFC : Vector Function Chainer) from FORTRAN to APAL (Machine language of the AP's) has to be used to translate the program sequences to be executed in the AP's ; however a good "vectorization" of these sequences is mandatory in order that translation does not deteriorate the performances of the computing algorithms. To keep the efficiency of the algorithms it is necessary in some cases to shortcut VFC by coding some instructions directly in APAL. These two ways of programming have been used in the following to solve by finite element, a very large Poisson problem using the architecture shown on Fig. 5.9.

5.3.2. Implementation of the quasi-direct method.

From Sec. 2.3.8, the quasi direct solution of the least squares problem (2.146)-(2.148) require the following steps :

(i) Construction and factorization (possibly incomplete) of the local discrete Poisson operators for each subdomain Ω_{ii} ;

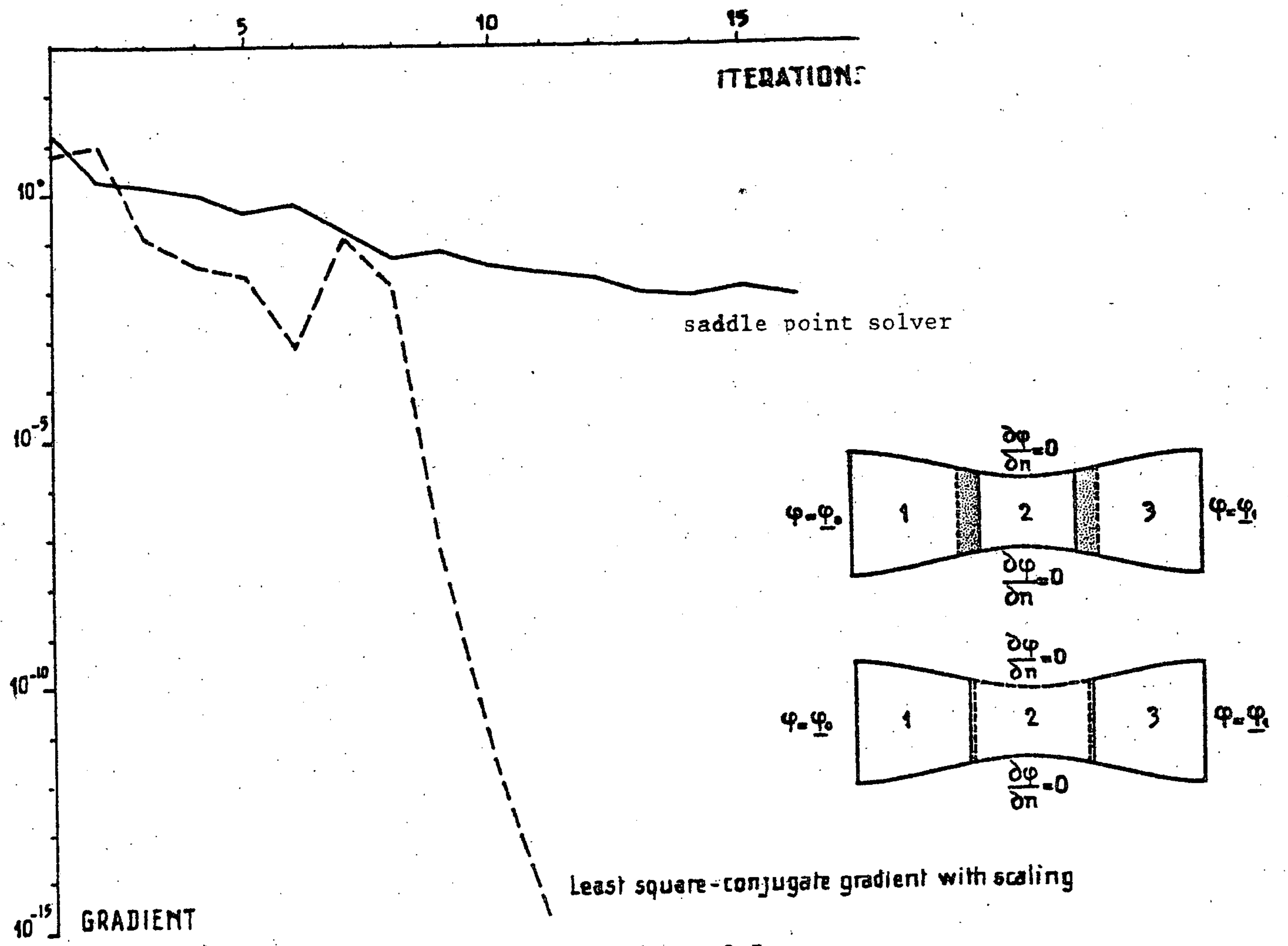


Figure 5.7

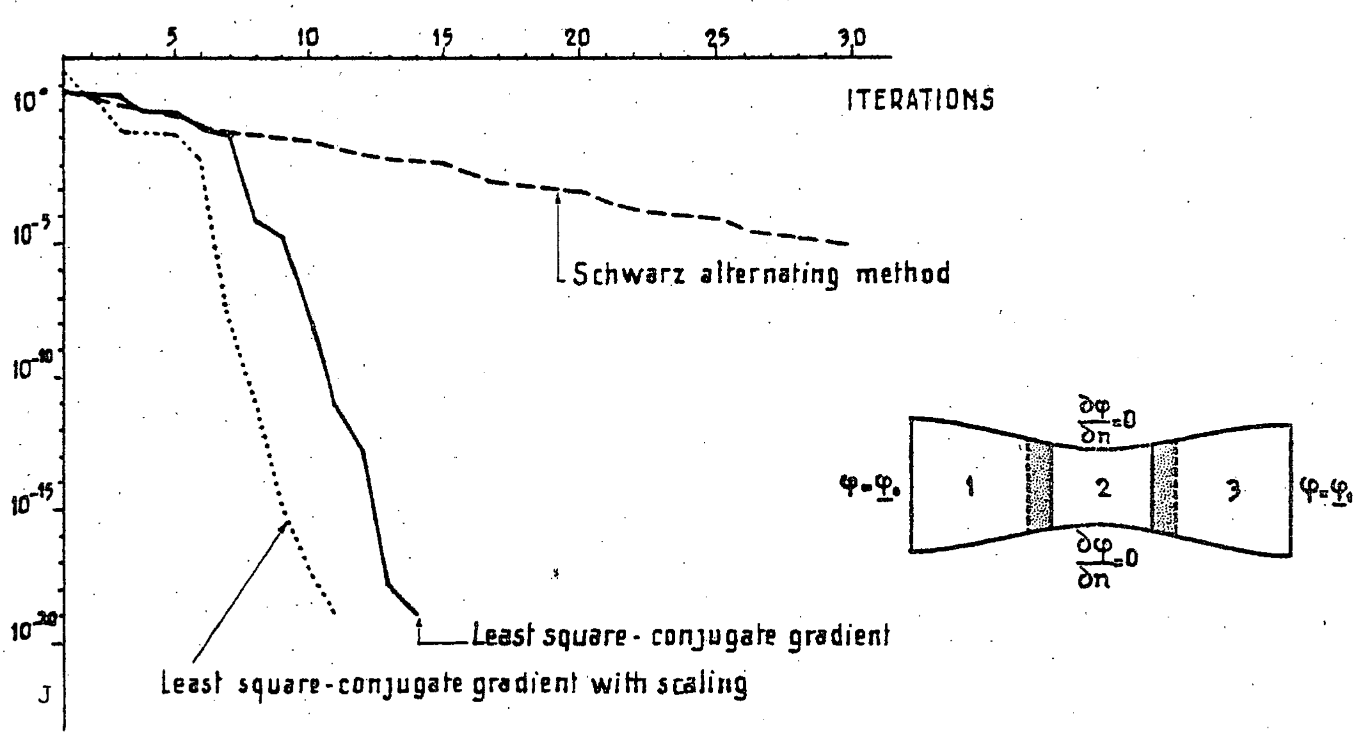


Figure 5.8

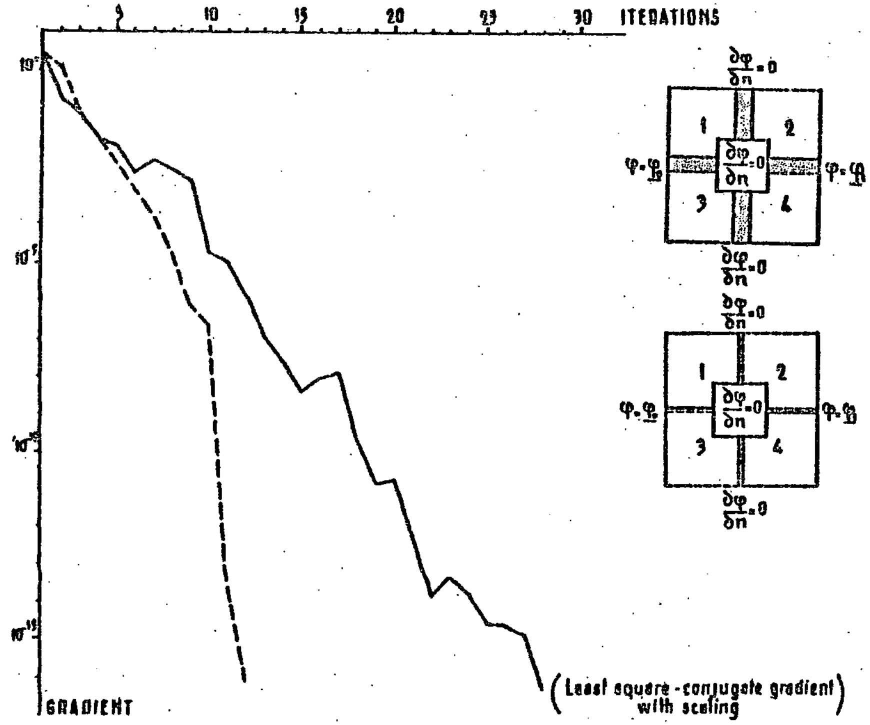


Figure 5.3

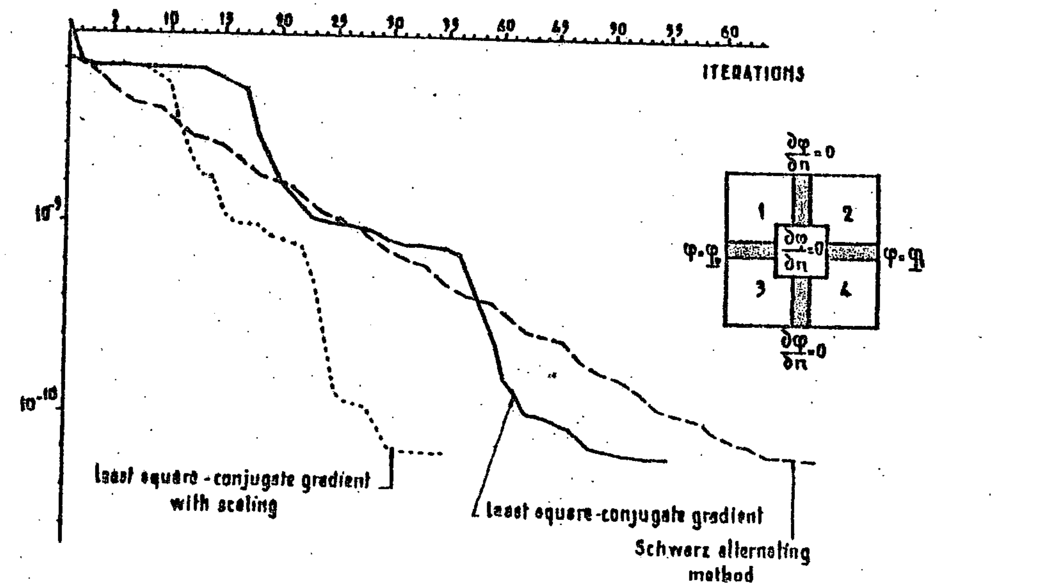


Figure 5.5

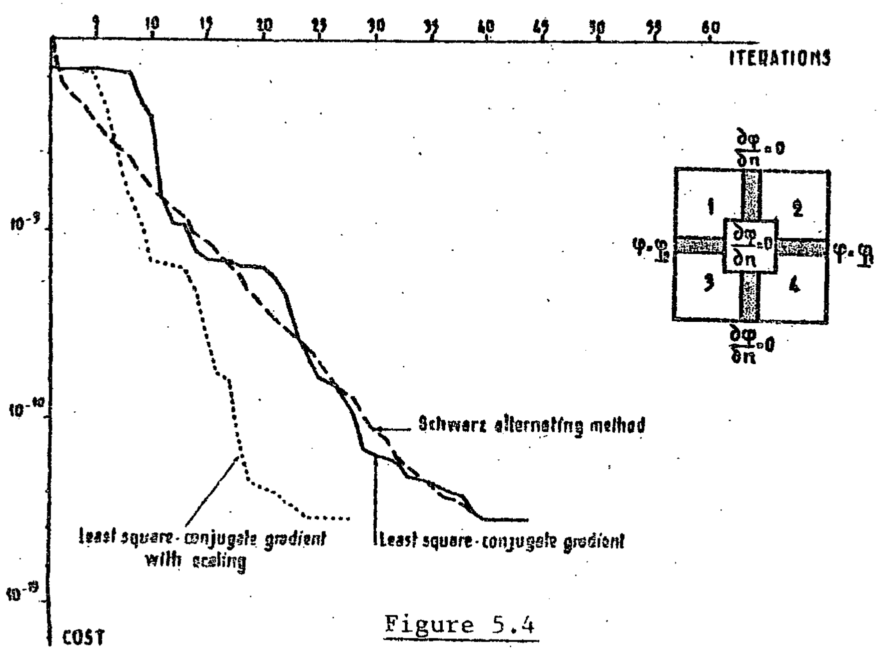


Figure 5.4

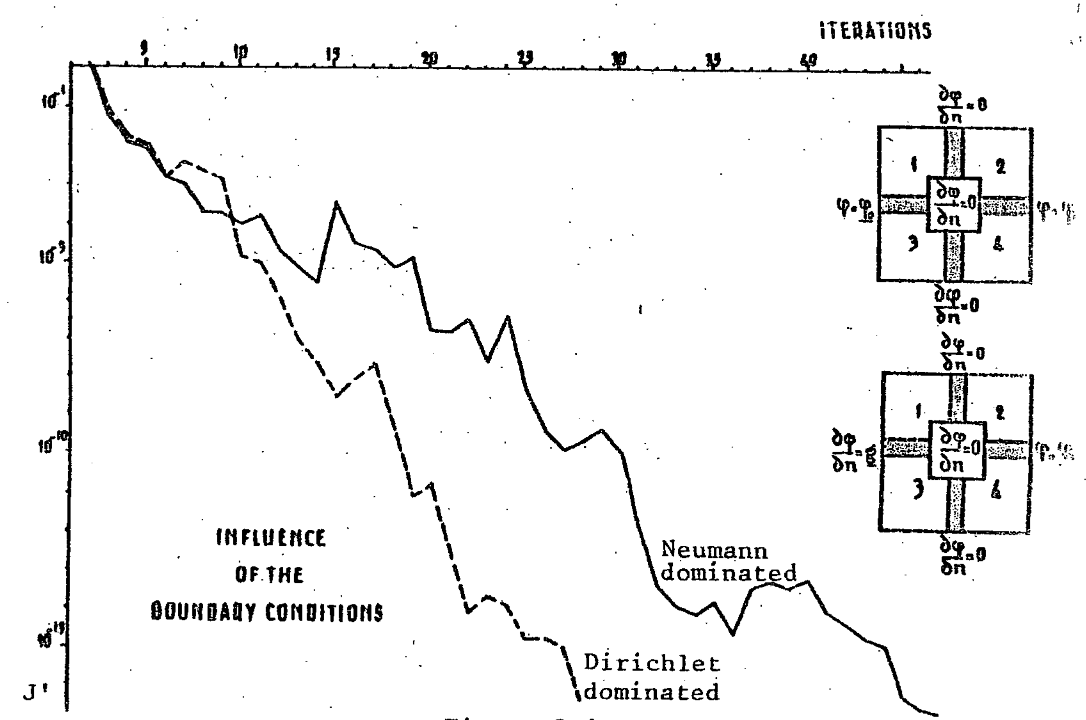


Figure 5.6

As a general remark we would like to point out that Cholesky factorizations have been used to solve the local discrete Poisson problems (i.e. on the Ω_{ii}) when applying the iterative methods described in Sec. 2.1, 2.2, 2.3 to solve (5.1) by domain decomposition techniques.

5.2.2. Description of the numerical results. Comments.

Test 5.1 : Figure 5.3 shows a comparison between the performances of the saddle-point conjugate gradient algorithm (2.47)-(2.55) (which uses a decomposition without overlapping) and the least squares conjugate gradient algorithm (2.65)-(2.73) (which uses a decomposition with overlapping). Both algorithms have been applied to the solution of a discrete version of the Poisson problem (5.1) using the coarse finite element grid of Fig. 5.1 (a) and piecewise linear finite elements ; the boundary conditions are those indicated on Fig. 5.3. The partition and overlapping regions are indicated on Figs. 5.1 and 5.3. For this example for which each subdomain solution has to satisfy Dirichlet boundary conditions on a part of each sub-boundary $\partial\Omega_{ii}$, the saddle-point conjugate gradient algorithm (2.47)-(2.55) appears to be faster than algorithm (2.65)-(2.73).

Test 5.2 : On Figure 5.4, we have compared the performances of three algorithms using domain decompositions with overlapping ; we have compared more precisely the Schwarz alternating algorithm (2.16)-(2.18), the least squares conjugate gradient algorithm (2.65)-(2.73) and the preconditioned least squares conjugate gradient algorithm (2.115)-(2.123). This example shows that algorithms (2.16)-(2.18) and (2.65)-(2.73) behave similarly and that the preconditioned algorithm (2.115)-(2.123) converges much faster. The mesh and the finite element method used in this test are those of Test 5.1.

Test 5.3 : It is the same test than Test 5.2, except that we use the fine grid shown on Figure 5.1 (b) to define the approximate problem. From Fig. 5.5, this test confirms the greater efficiency of the preconditioned algorithm (2.115)-(2.123) ; it shows also that to some extent the global performances of our algorithms are not deteriorated by the refinement of the mesh ; this is not surprising since these algorithms converge also for the continuous problems.

Test 5.4 : The purpose of this test is to compare the influence of the boundary conditions on the speed of convergence. With the same approximation than in Tests 5.1, 5.2 we have used the preconditioned least squares conjugate gradient algorithm (2.115)-(2.123). We observe (from Figure 5.6) that the convergence is slowed down if the boundary conditions are Neumann dominated ; in fact this feature has been observed quite often when solving elliptic boundary value problems by iterative methods.

Test 5.5 : This test is concerned with a problem,which is close to realistic aerodynamic simulation of potential inviscid flows modelled by Laplace equation (i.e. $f=0$ in (5.1)), in the convergent-divergent nozzle of Fig. 5.2 ; the boundary conditions are indicated in Fig. 5.7, 5.8.

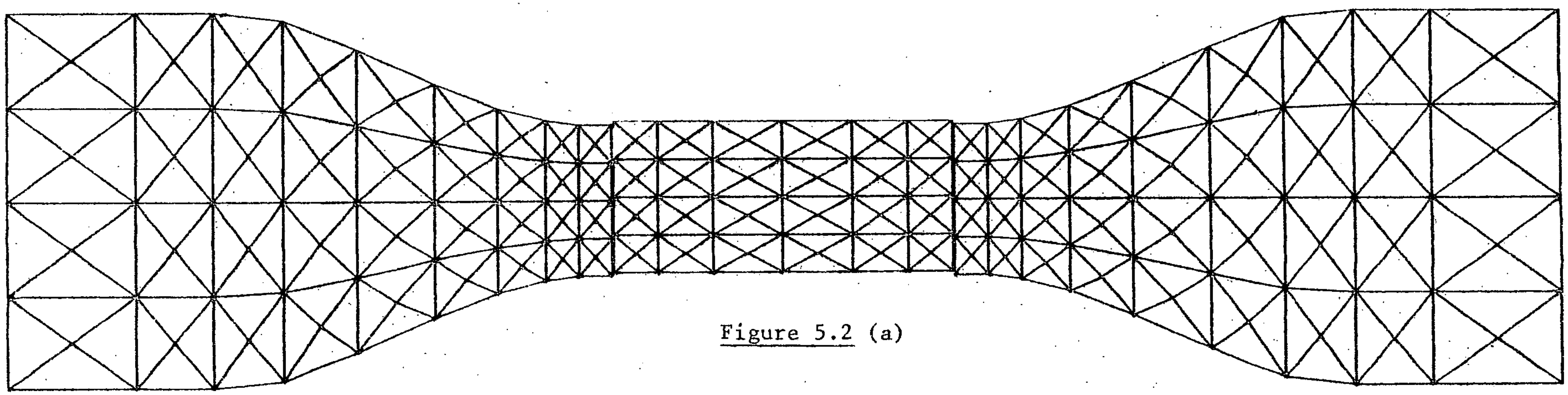


Figure 5.2 (a)

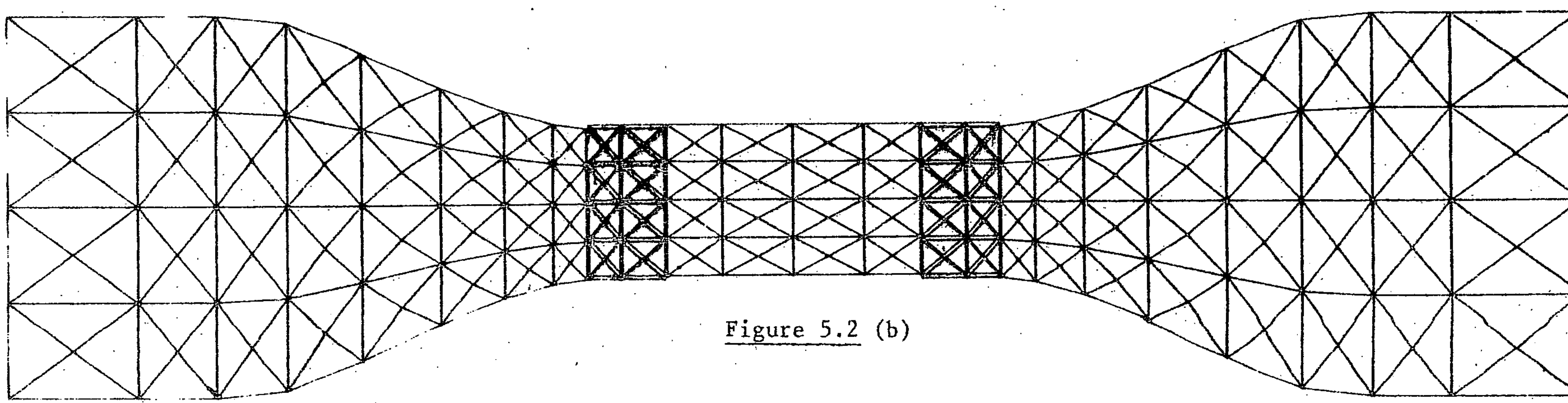


Figure 5.2 (b)

Figure 5.2

A convergent-divergent nozzle and its finite element triangulation

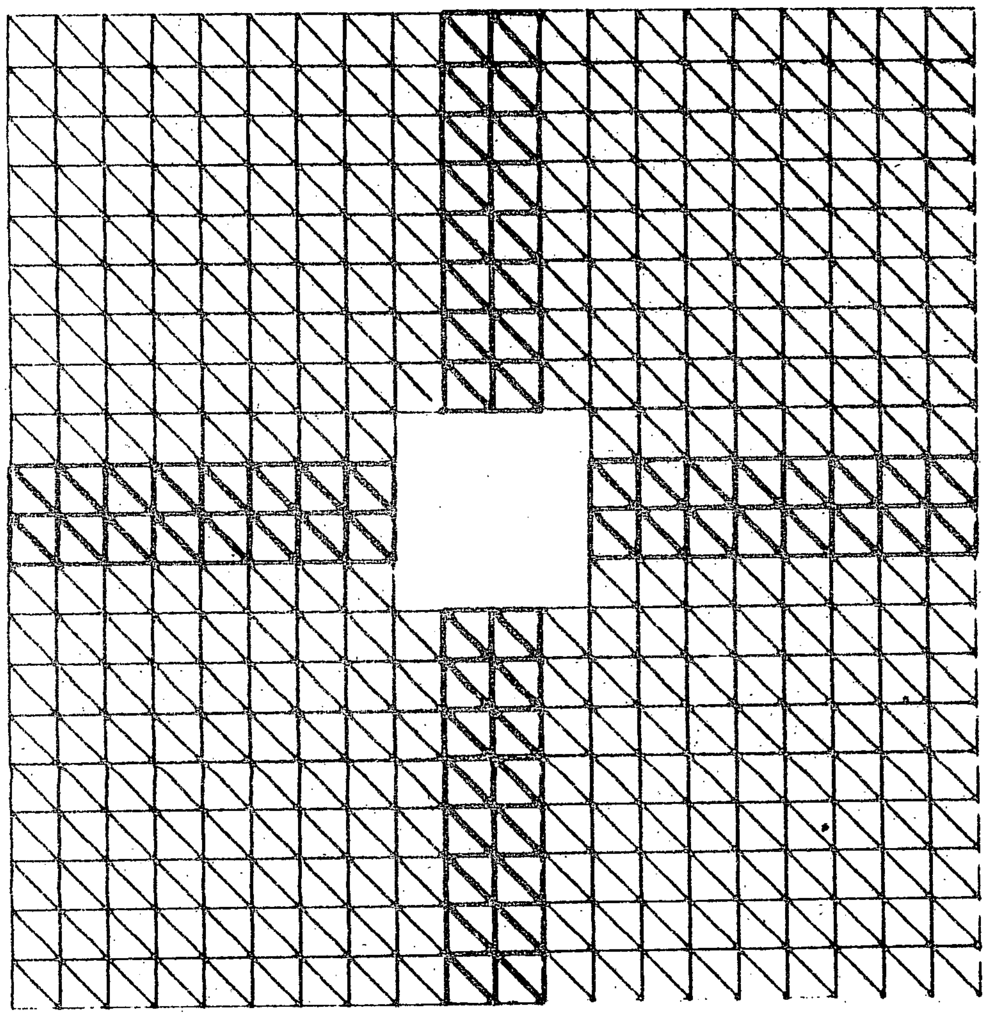


Figure 5.1 (a)
Coarse triangulation

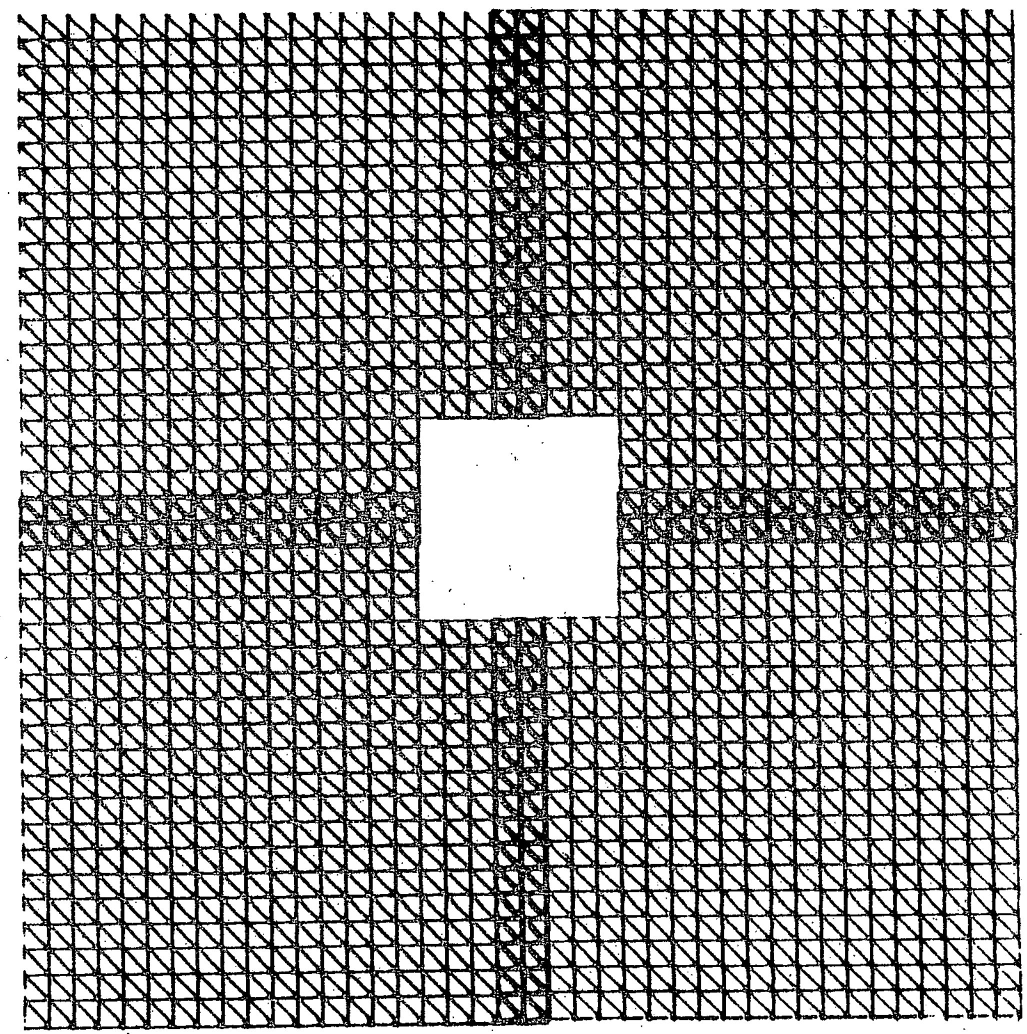


Figure 5.1 (b)
Fine triangulation

Figure 5.1

A square cavity with hole and two finite element triangulations (the overlapping triangles have been reinforced)

Here again the solution of (4.77) can be reduced to the solution of Poisson problems for which the decomposition techniques of Sec. 2.3 still apply. We refer to [1], [11], [23], [31], for a detailed discussion of the theoretical and numerical solution of (4.77) via the solution of Poisson problems like those discussed in the earlier sections of this paper.

5. - NUMERICAL EXPERIMENTS.

5.1. Synopsis.

We shall consider in this Sec. 5 the solution of various test problems either linear or nonlinear by the various iterative and direct decomposition techniques discussed in the previous sections.

In Sec. 5.2 we compare the performances of the various iterative methods described in Secs. 2.1, 2.2, 2.3, taking as test problem the Poisson equation $-\Delta\phi = f$ on various domains Ω, for various boundary conditions. In Sec. 5.3 we discuss the implementation of the quasi-direct method of Sec. 2.3.8, on a computing system consisting of the combination of an host machine IBM 3033 monitoring two array processors FPS 190, working in parallel ; the test problem is the numerical simulation of the three-dimensional flow of an incompressible, inviscid fluid around a portion of wing. In Sec. 5.4 (resp. 5.5) we combine the decomposition techniques of Sec. 2.3 with the least squares formulations of Sec. 4.1 (resp. 4.2) to solve the full potential equation for transonic flows of compressible inviscid fluids (resp. the Navier-Stokes equations for incompressible viscous fluids).

5.2. Compared performances of several iterative domain decomposition methods.

5.2.1. Formulation of the test problems.

We consider hereafter the numerical solution of the following mixed linear elliptic boundary value problem

$$(5.1)\qquad \begin{cases} -\Delta\phi = f \ \underline{\text{in}}\ \Omega, \\ \phi = g_o \ \underline{\text{on}}\ \Gamma_o, \\ \dfrac{\partial\phi}{\partial n} = g_1 \ \underline{\text{on}}\ \Gamma_1 \end{cases}$$

where $\Gamma_o \cup \Gamma_1 = \Gamma\ (= \partial\Omega)$, $\int_{\Gamma_o \cap \Gamma_1} d\Gamma = 0$ and where f, g_o, g_1 are given functions. In our test problems we have chosen for Ω the two following domains :

(i) A square cavity with a hole, as shown on Figure 5.1 with two finite element triangulations and also domain decompositions with overlapping.

(ii) A convergent-divergent nozzle, as shown on Figure 5.2 with a finite element triangulation on which a partition boundary (resp. an overlapping region) has been indicated on Fig. 5.2 (a) (resp. 5.2 (b)).

4.2.3.3.3. Further comments on algorithm (4.64)-(4.71).

Each step of algorithm (4.64)-(4.71) requires the solution of several Dirichlet systems for the operator $\alpha I-\nu\Delta$; more precisely we have to solve the following such systems :

(i) System (4.76) to obtain y^{n+1} from u^{n+1},

(ii) System (4.69) to obtain $\tilde{g}^{n+1}$ from $\tilde{u}^{n+1}$, y^{n+1},

(iii) Two systems to obtain the coefficients of the quartic polynomial

$$\lambda \to J(u^n-\lambda w^n).$$

Thus we have to solve 4 Dirichlet systems for $\alpha I-\nu\Delta$ at each iteration (or equivalently 4N scalar Dirichlet problems for $\nu I-\nu\Delta$ at each iteration).

From the above observations it appears clearly that the practical implementation of algorithm (4.64)-(4.71) will require an efficient (direct or iterative) elliptic solver.

As mentioned in Secs. 3.4 and 4.1.4.3 (for other nonlinear problems) an efficient discrete Poisson solver is basic to solve the nonlinear problems (4.55) using the conjugate gradient algorithm (4.64)-(4.71) ; such a solver can be obtained via the domain decomposition techniques discussed in Sec. 2.3.

The solution of the one-dimensional problem (4.67) can be done very efficiently since it is equivalent to finding the roots of a single variable cubic polynomial whose coefficients are known.

As a last comment we would like to mention that algorithm (4.64)-(4.71) (in fact its finite element variants) is quite efficient ; when used in combination with the alternating direction methods of Sec. 4.2 three iterations suffice to reduce the value of the cost function J by a factor of 10^4.

4.2.4. Solution of the "quasi" Stokes linear subproblems.

At each full step of the alternating direction methods (4.48)-(4.50) and (4.51)-(4.54) we have to solve a linear problem of the following type

$$(4.77)\quad \begin{cases} \alpha u - \nu\Delta u + \nabla p = f \text{ in } \Omega, \\ \nabla\cdot u = 0 \text{ in } \Omega, \\ u = g \text{ on } \Gamma \text{ (with } \int_\Gamma g\cdot n\, d\Gamma = 0), \end{cases}$$

where α and ν are two positive parameters and where f and g are two given functions defined on Ω and Γ, respectively.

We recall that if f and g are sufficiently smooth, then problem (4.77) has a unique solution in $V_g \times (L^2(\Omega)/\mathbb{R})$ (with V_g still defined by (4.57) ; $p \in L^2(\Omega)/\mathbb{R}$ means that p is defined only to within an arbitrary constant).

calculation we shall give it in detail.

Let $\underset{\sim}{v} \in V_g$ and let $\delta\underset{\sim}{v}$ be a perturbation of $\underset{\sim}{v}$ such that $\delta\underset{\sim}{v} \in V_0$ (i.e. $\delta\underset{\sim}{v}=\underset{\sim}{0}$ on Γ) ; we have for the corresponding variation of $J(\underset{\sim}{v})$

$$(4.72) \qquad \delta J(\underset{\sim}{v}) = \langle J'(\underset{\sim}{v}), \delta\underset{\sim}{v}\rangle \; .$$

Using (4.61), (4.63) we also have that

$$(4.73) \qquad J(\underset{\sim}{v}) = \int_\Omega \{\alpha \underset{\sim}{y}\cdot\delta\underset{\sim}{y} + \nu\nabla\underset{\sim}{y}\cdot\nabla\delta\underset{\sim}{y}\}\; dx,$$

where $\delta\underset{\sim}{y}$ is the solution of the linear problem

$$(4.74) \qquad \begin{cases} \delta\underset{\sim}{y} \in V_0, \\ \alpha\int_\Omega \delta\underset{\sim}{y}\cdot\underset{\sim}{z}\; dx + \nu\int_\Omega \nabla\delta\underset{\sim}{y}\cdot\nabla\underset{\sim}{z}\; dx = \alpha\int_\Omega \delta\underset{\sim}{v}\cdot\underset{\sim}{z}\; dx + \nu\int_\Omega \nabla\delta\underset{\sim}{v}\cdot\nabla\underset{\sim}{z}\; dx + \\ + \int_\Omega ((\delta\underset{\sim}{v}\cdot\nabla)\underset{\sim}{v})\cdot\underset{\sim}{z}\; dx + \int_\Omega ((\underset{\sim}{v}\cdot\nabla)\cdot\delta\underset{\sim}{v})\cdot\underset{\sim}{z}\; dx \quad \forall \underset{\sim}{z}\in V_0 . \end{cases}$$

Taking $\underset{\sim}{z}=\underset{\sim}{y}$ in (4.74) we obtain from (4.72), (4.73) that

$$\begin{cases} \langle J'(\underset{\sim}{v}),\delta\underset{\sim}{v}\rangle = \alpha\int_\Omega \underset{\sim}{y}\cdot\delta\underset{\sim}{v}\; dx + \nu\int_\Omega \nabla\underset{\sim}{y}\cdot\nabla\delta\underset{\sim}{v}\; dx + \int_\Omega ((\delta\underset{\sim}{v}\cdot\nabla)\underset{\sim}{v})\cdot\underset{\sim}{y}\; dx + \\ + \int_\Omega ((\underset{\sim}{v}\cdot\nabla)\cdot\delta\underset{\sim}{v})\cdot\underset{\sim}{y}\; dx. \end{cases}$$

Thus $J'(\underset{\sim}{v})$ can be identified with the linear functional from V_0 to $\mathbb{R}$ defined by

$$(4.75) \qquad \begin{cases} \langle J'(\underset{\sim}{v}),\underset{\sim}{z}\rangle = \alpha\int_\Omega \underset{\sim}{y}\cdot\underset{\sim}{z}\; dx + \nu\int_\Omega \nabla\underset{\sim}{y}\cdot\nabla\underset{\sim}{z}\; dx + \int_\Omega \underset{\sim}{y}\cdot(\underset{\sim}{z}\cdot\nabla)\underset{\sim}{v}\; dx + \\ \qquad + \int_\Omega \underset{\sim}{y}\cdot(\underset{\sim}{v}\cdot\nabla)\underset{\sim}{z}\; dx \quad \forall \underset{\sim}{z}\in V_0 \; ; \end{cases}$$

it has therefore a purely integral representation, which is of major importance in view of finite element implementations of algorithm (4.64)-(4.71).

From the above results, to obtain $\langle J'(\underset{\sim}{u}^{n+1}),\underset{\sim}{z}\rangle$ we proceed as follows :

(i) We compute $\underset{\sim}{y}^{n+1}$ from $\underset{\sim}{u}^{n+1}$ through the solution of (4.60) with $\underset{\sim}{v} = \underset{\sim}{u}^{n+1}$, i.e. we solve the Dirichlet system

$$(4.76) \qquad \begin{cases} \alpha\underset{\sim}{y}^{n+1} - \nu\Delta\underset{\sim}{y}^{n+1} = \alpha\underset{\sim}{u}^{n+1} - \nu\Delta\underset{\sim}{u}^{n+1} + (\underset{\sim}{u}^{n+1}\cdot\underset{\sim}{\nabla})\underset{\sim}{u}^{n+1} - \underset{\sim}{f} \;\underline{\text{in}}\; \Omega, \\ \underset{\sim}{y}^{n+1} = \underset{\sim}{0} \;\underline{\text{on}}\; \Gamma. \end{cases}$$

(ii) We finally obtain $\langle J'(\underset{\sim}{u}^{n+1}),\underset{\sim}{z}\rangle$ by taking in (4.75) $\underset{\sim}{v} = \underset{\sim}{u}^{n+1}$ and $\underset{\sim}{y} = \underset{\sim}{y}^{n+1}$.

Step 0 : Initialization

(4.64) $\quad u^o \in V_g$, given,

we define then $g^o, w^o \in V_o$ by

$$(4.65)\quad \begin{cases} \alpha\int_\Omega g^o \cdot z\, dx + \nu\int_\Omega \nabla g^o \cdot \nabla z\, dx = \langle J'(u^o), z\rangle \quad \forall z \in V_o, \\ g^o \in V_o, \end{cases}$$

$$(4.66)\quad w^o = g^o,$$

respectively.

Then for $n \geq 0$, assuming that u^n, g^n, w^n are known we obtain u^{n+1}, g^{n+1}, u^{n+1} by

Step 1 : Descent

$$(4.67)\quad \begin{cases} \text{Find } \lambda^n \in \mathbb{R} \text{ such that} \\ J(u^n - \lambda^n w^n) \leq J(u^n - \lambda w^n) \quad \forall \lambda \in \mathbb{R}, \end{cases}$$

$$(4.68)\quad u^{n+1} = u^n - \lambda^n w^n .$$

Step 2 : Calculation of the new descent direction

$$(4.69)\quad \begin{cases} \text{Find } g^{n+1} \in V_o \text{ such that} \\ \alpha\int_\Omega g^{n+1} \cdot z\, dx + \nu\int_\Omega \nabla g^{n+1} \cdot \nabla z\, dx = \langle J'(u^{n+1}), z\rangle \quad \forall z \in V_o, \end{cases}$$

$$(4.70)\quad \gamma_n = \frac{\int_\Omega g^{n+1} \cdot (g^{n+1} - g^n)\, dx + \nu\int_\Omega \nabla g^{n+1} \cdot \nabla(g^{n+1} - g^n)\, dx}{\alpha\int_\Omega |g^n|^2\, dx + \nu\int_\Omega |\nabla g^n|^2\, dx},$$

$$(4.71)\quad w^{n+1} = g^{n+1} + \gamma_n w^n ,$$

$n=n+1$, go to (4.67).

As we shall see in Secs. 4.2.3.3.2, 4.2.3.3.3, applying algorithm (4.64)-(4.71) to solve the least square problem (4.62) requires the solution at each iteration of several Dirichlet problems associated to the elliptic operator $\alpha I - \nu\Delta$.

4.2.3.3.2. Calculation of J'.

A most important step, when making use of algorithm (4.64)-(4.71) to solve the least square problem (4.62), is the calculation of $\langle J'(u^{n+1}), z\rangle$ at each iteration ; owing to the importance of this

4.2.3.2. Least squares formulation of (4.55),(4.59).

Let $\underset{\sim}{v} \in V_g$; from $\underset{\sim}{v}$ we define $\underset{\sim}{y}$ $(= \underset{\sim}{y}(\underset{\sim}{v})) \in V_o$ as the solution of

(4.60) $$\begin{cases} \alpha \underset{\sim}{y} - \nu \Delta \underset{\sim}{y} = \alpha \underset{\sim}{v} - \nu \Delta \underset{\sim}{v} + (\underset{\sim}{v} \cdot \underset{\sim}{\nabla}) \underset{\sim}{v} - \underset{\sim}{f} \ \underline{\text{in}}\ \Omega, \\ \underset{\sim}{y} = 0 \ \underline{\text{on}}\ \Gamma. \end{cases}$$

We observe that y is obtained from v via the solution of N uncoupled linear Poisson problems (one for each component of y) ; using (4.58) it can be shown that problem (4.60) is actually equivalent to the linear variational problem

(4.61) $$\begin{cases} \underline{\text{Find}}\ \underset{\sim}{y} \in V_o\ \underline{\text{such that}}\ \forall \underset{\sim}{z} \in V_o\ \underline{\text{we have}} \\ \alpha \int_\Omega \underset{\sim}{y} \cdot \underset{\sim}{z}\, dx + \nu \int_\Omega \underset{\sim\sim}{\nabla y} \cdot \underset{\sim\sim}{\nabla z}\, dx = \alpha \int_\Omega \underset{\sim}{v} \cdot \underset{\sim}{z}\, dx + \nu \int_\Omega \underset{\sim\sim}{\nabla v} \cdot \underset{\sim\sim}{\nabla z}\, dx + \\ + \int_\Omega ((\underset{\sim}{v} \cdot \underset{\sim}{\nabla}) \cdot \underset{\sim}{v}) \cdot \underset{\sim}{z}\, dx - \int_\Omega \underset{\sim}{f} \cdot \underset{\sim}{z}\, dx, \end{cases}$$

<u>which has a unique solution.</u>

Suppose now that $\underset{\sim}{v}$ is a solution of the nonlinear problem (4.55), (4.59) ; the corresponding $\underset{\sim}{y}$ (obtained through the solution of (4.60), (4.61) is clearly $\underset{\sim}{y} = 0$. From these observations it is quite natural to introduce the following (nonlinear) least squares formulation of problem (4.55), (4.59) :

(4.62) $$\begin{cases} \underline{\text{Find}}\ \underset{\sim}{u} \in V_g\ \underline{\text{such that}} \\ J(\underset{\sim}{u}) \le J(\underset{\sim}{v}) \quad \forall \underset{\sim}{v} \in V_g, \end{cases}$$

where $J : (H^1(\Omega))^N \to \mathbb{R}$ is that function of v defined by

(4.63) $$J(\underset{\sim}{v}) = \frac{1}{2} \int_\Omega \{\alpha |\underset{\sim}{y}|^2 + \nu |\underset{\sim\sim}{\nabla y}|^2\}\, dx,$$

with $\underset{\sim}{y}$ defined from $\underset{\sim}{v}$ by solving the linear problem (4.60),(4.61).

We observe that if u is solution of (4.55), (4.59) it is also a solution of (4.62) such that $J(\underset{\sim}{u}) = 0$; conversely if u is a solution of (4.62) such that $J(\underset{\sim}{u}) = 0$ it is also a solution of (4.55), (4.59).

4.2.3.3. Conjugate gradient solution of the least squares problem (4.62).

4.2.3.3.1. Description of the algorithm.

We use the Polak-Ribière version of the conjugate gradient method to solve the minimization problem (4.62) ; we have then (with $J'(\underset{\sim}{v})$ the differential of J at $\underset{\sim}{v}$)

where α and ν are two positive parameters and where f and $\underset{\sim}{g}$ are two given functions defined on Ω and Γ, respectively.

We shall not discuss here the existence and uniqueness of solutions for problem (4.55).

We introduce now the following functional spaces of Sobolev's type

(4.56) $\quad V_o = (H^1_o(\Omega))^N,$

(4.57) $\quad V_g = \{\underset{\sim}{v} \mid \underset{\sim}{v} \in (H^1(\Omega))^N,\ \underset{\sim}{v} = \underset{\sim}{g} \text{ on } \Gamma\}$;

if $\underset{\sim}{g}$ is sufficiently smooth then V_g is nonempty.

We shall use in the sequel the following notation

$$dx = dx_1 \dots dx_N,$$

and, if $\underset{\sim}{u} = \{\underset{\sim}{u}_i\}_{i=1}^N$, $\underset{\sim}{v} = \{\underset{\sim}{v}_i\}_{i=1}^N$,

$$\underset{\sim}{u} \cdot \underset{\sim}{v} = \sum_{i=1}^N u_i v_i,$$

$$\underset{\sim\sim}{\nabla u} \cdot \underset{\sim\sim}{\nabla v} = \sum_{i=1}^N \underset{\sim\sim}{\nabla u_i} \cdot \underset{\sim\sim}{\nabla v_i} = \sum_{i=1}^N \sum_{j=1}^N \frac{\partial u_i}{\partial x_j} \frac{\partial v_i}{\partial x_j}.$$

Using Green's formula we can prove that for sufficiently smooth functions $\underset{\sim}{u}$ and $\underset{\sim}{v}$, belonging to $(H^1(\Omega))^N$ and V_o, respectively, we have

(4.58) $\quad -\int_\Omega \Delta \underset{\sim}{u} \cdot \underset{\sim}{v}\, dx = \int_\Omega \underset{\sim\sim}{\nabla u} \cdot \underset{\sim\sim}{\nabla v}\, dx.$

It can also be proved that if $\underset{\sim}{u} \in V_g$ is a solution of (4.55) it is also a solution of the nonlinear variational problem

(4.59)
$$\begin{cases} \text{Find } \underset{\sim}{u} \in V_g \text{ such that} \\ \alpha \int_\Omega \underset{\sim}{u} \cdot \underset{\sim}{v}\, dx + \nu \int_\Omega \underset{\sim\sim}{\nabla u} \cdot \underset{\sim\sim}{\nabla v}\, dx + \int_\Omega ((\underset{\sim}{u} \cdot \underset{\sim}{\nabla}) \underset{\sim}{u}) \cdot \underset{\sim}{v}\, dx = \\ = \int_\Omega \underset{\sim}{f} \cdot \underset{\sim}{v}\, dx \quad \forall \underset{\sim}{v} \in V_o, \end{cases}$$

and conversely.

We observe that (4.55), (4.59) is not equivalent to a problem of the Calculus of Variations since there is no functional of $\underset{\sim}{v}$ with $(\underset{\sim}{v} \cdot \underset{\sim}{\nabla})\underset{\sim}{v}$ as differential ; however using a convenient least square formulation we shall be able to solve (4.55), (4.59) by efficient methods from Nonlinear Programming, like conjugate gradient, for example.

The finite element approximation of problem (4.55),(4.59) is discussed in [23].

$$
(4.54)\quad \begin{cases} \dfrac{\underset{\sim}{u}^{n+1}-\underset{\sim}{u}^{n+3/4}}{(\frac{\Delta t}{4})} - \theta\nu\Delta\underset{\sim}{u}^{n+1} + \underset{\sim}{\nabla} p^{n+1} = \underset{\sim}{f}^{n+1}+(1-\theta)\nu\Delta\underset{\sim}{u}^{n+3/4} \\ = (\underset{\sim}{u}^{n+3/4}\cdot\underset{\sim}{\nabla})\underset{\sim}{u}^{n+3/4} \ \underline{\text{in}}\ \Omega, \\ \underset{\sim}{\nabla}\cdot\underset{\sim}{u}^{n+1} = 0 \ \underline{\text{in}}\ \Omega, \\ \underset{\sim}{u}^{n+1} = \underset{\sim}{g}^{n+1} \ \underline{\text{on}}\ \Gamma. \end{cases}
$$

4.2.2.3. Some comments and remarks concerning the alternating direction schemes (4.48)-(4.50) and (4.51)-(4.54).

Using the two alternating direction schemes described in Secs. 4.2.2.1, 4.2.2.2 we have been able to decouple nonlinearity and incompressibility in the Navier-Stokes equations (4.41),(4.42). We shall describe in the following sections the specific treatment of the subproblems encountered at each step of (4.48)-(4.50) and (4.51)-(4.54) ; we shall consider only the case where the subproblems are still continuous in space (since the formalism of the continuous problems is much simpler) ; for the discrete case see GLOWINSKI-MANTEL-PERIAUX [5] and also GLOWINSKI [23].

Scheme (4.48)-(4.50) has a truncation error in $O(\Delta t)$; due to the symmetrization process involved in it, scheme (4.51)-(4.54) has a truncation error in $O(|\Delta t|^2)$.

We observe that $\underset{\sim}{u}^{n+1/2}$ and $\underset{\sim}{u}^{n+1/4}$, $\underset{\sim}{u}^{n+1}$ are obtained from the solution of the linear problems (4.49) and (4.52),(4.54),respectively, very close to the steady Stokes problem. Despite of its greater complexity scheme (4.51)-(4.54) is almost as economical to use as scheme (4.48)-(4.50) ; this is mainly due to the fact that the "quasi" steady Stokes problems (4.49) and (4.52), (4.54) (in fact convenient finite element approximations of them) can be solved by quite efficient solvers resulting in that most of the computer time used to solve a full alternating direction step (4.49),(4.50) or (4.52)-(4.54) is in fact used to solve the nonlinear subproblem (4.50) or (4.53).

The good choice for θ is $\theta = 1/2$ (resp. $\theta = 1/3$) if one uses scheme (4.48)-(4.50) (resp. (4.51)-(4.54)) ; this follows from the fact that with the above choices for θ, many computer subprograms are the same for both the linear and nonlinear subproblems, resulting therefore in quite substantial computer core memory savings.

4.2.3. Least squares-conjugate gradient solution of the nonlinear subproblems.

4.2.3.1. Classical and Variational Formulations. Synopsis.

At each full step of the alternating direction methods (4.48)-(4.50) and (4.51)-(4.54) we have to solve a nonlinear elliptic system of the following type

$$
(4.55)\quad \begin{cases} \alpha\underset{\sim}{u} - \nu\Delta\underset{\sim}{u} + (\underset{\sim}{u}\cdot\underset{\sim}{\nabla})\underset{\sim}{u} = \underset{\sim}{f} \ \underline{\text{in}}\ \Omega, \\ \underset{\sim}{u} = \underset{\sim}{g} \ \underline{\text{on}}\ \Gamma, \end{cases}
$$

4.2.2.1. A first alternating direction method.

We consider first the following alternating direction method (of Peaceman-Rachford type) :

(4.48) $\underset{\sim}{u}^{o} = \underset{\sim}{u}_{o}$,

then for $n \geq 0$ compute $\{\underset{\sim}{u}^{n+1/2}, p^{n+1/2}\}$ and $\underset{\sim}{u}^{n+1}$, from $\underset{\sim}{u}^{n}$, by solving

$$(4.49)\quad \begin{cases} \dfrac{\underset{\sim}{u}^{n+1}-\underset{\sim}{u}^{n}}{(\Delta t/2)} - \theta\nu\Delta\underset{\sim}{u}^{n+1/2} + \underset{\sim}{\nabla} p^{n+1/2} = \underset{\sim}{f}^{n+1/2} + (1-\theta)\nu\Delta\underset{\sim}{u}^{n} - (\underset{\sim}{u}^{n}\cdot\underset{\sim}{\nabla})\underset{\sim}{u}^{n} \text{ in } \Omega, \\ \underset{\sim}{\nabla}\cdot\underset{\sim}{u}^{n+1/2} = 0 \text{ in } \Omega, \\ \underset{\sim}{u}^{n+1/2} = \underset{\sim}{g}^{n+1/2} \text{ on } \Gamma, \end{cases}$$

and

$$(4.50)\quad \begin{cases} \dfrac{\underset{\sim}{u}^{n+1}-\underset{\sim}{u}^{n+1/2}}{(\Delta t/2)} - (1-\theta)\nu\Delta\underset{\sim}{u}^{n+1} + (\underset{\sim}{u}^{n+1}\cdot\underset{\sim}{\nabla})\underset{\sim}{u}^{n+1} = \\ = \underset{\sim}{f}^{n+1} + \theta\nu\Delta\underset{\sim}{u}^{n+1/2} - \underset{\sim}{\nabla} p^{n+1/2} \text{ in } \Omega, \\ \underset{\sim}{u}^{n+1} = \underset{\sim}{g}^{n+1} \text{ on } \Gamma, \end{cases}$$

respectively.

We use the notation $\underset{\sim}{f}^{j}(x) = \underset{\sim}{f}(x,j\Delta t), \underset{\sim}{g}^{j}(x) = \underset{\sim}{g}(x,j\Delta t)$, and $\underset{\sim}{u}^{j}(x)$ is an approximation of $\underset{\sim}{u}(x,j\Delta t)$.

4.2.2.2. A second alternating direction method.

We consider now the following alternating direction method (of Strang type) :

(4.51) $\underset{\sim}{u}^{o} = \underset{\sim}{u}_{o}$,

then for $n \geq 0$ and starting from $\underset{\sim}{u}^{n}$ we solve

$$(4.52)\quad \begin{cases} \dfrac{\underset{\sim}{u}^{n+1/4}-\underset{\sim}{u}^{n}}{(\frac{\Delta t}{4})} - \theta\nu\Delta\underset{\sim}{u}^{n+1/4} + \underset{\sim}{\nabla} p^{n+1/4} = \underset{\sim}{f}^{n+1/4} + (1-\theta)\nu\Delta\underset{\sim}{u}^{n} - (\underset{\sim}{u}^{n}\cdot\underset{\sim}{\nabla})\underset{\sim}{u}^{n} \text{ in } \Omega, \\ \underset{\sim}{\nabla}\cdot\underset{\sim}{u}^{n+1/4} = 0 \text{ in } \Omega, \\ \underset{\sim}{u}^{n+1/4} = \underset{\sim}{g}^{n+1/4} \text{ on } \Gamma, \end{cases}$$

$$(4.53)\quad \begin{cases} \dfrac{\underset{\sim}{u}^{n+3/4}-\underset{\sim}{u}^{n+1/4}}{(\Delta t/2)} - (1-\theta)\nu\Delta\underset{\sim}{u}^{n+3/4} + (\underset{\sim}{u}^{n+3/4}\cdot\underset{\sim}{\nabla})\underset{\sim}{u}^{n+3/4} = \\ \underset{\sim}{f}^{n+3/4} + \theta\nu\Delta\underset{\sim}{u}^{n+1/4} - \underset{\sim}{\nabla} p^{n+1/4} \text{ in } \Omega, \\ \underset{\sim}{u}^{n+3/4} = \underset{\sim}{g}^{n+3/4} \text{ on } \Gamma, \end{cases}$$

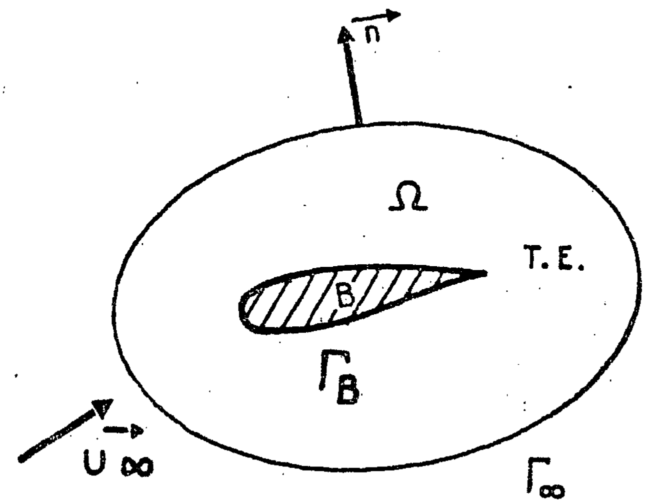


Figure 4.3

where (from the incompressibility of the fluid) the given function $\underset{\sim}{g}$ has to satisfy

(4.46) $$\int_\Gamma \underset{\sim}{g} \cdot \underset{\sim}{n}\, d\Gamma = 0,$$

where $\underset{\sim}{n}$ is the outward unit vector normal at Γ.

Finally for the time dependent problem (4.41),(4.42) an initial condition such as

(4.47) $$\underset{\sim}{u}(x,0) = \underset{\sim}{u}_o(x) \text{ a.e. on } \Omega,$$

with $\underset{\sim}{u}_o$ given, is usually prescribed.

From the above equations we observe three difficulties (even for flows at low Reynold's numbers in bounded regions Ω) which are

(i) the nonlinear term $(\underset{\sim}{u}\cdot\underset{\sim}{\nabla})\underset{\sim}{u}$ in (4.41),
(ii) the incompressibility condition (4.42),
(iii) the fact that the solutions of the Navier-Stokes equations are vector-valued functions of x,t, whose components are coupled by the nonlinear term $(\underset{\sim}{u}\cdot\underset{\sim}{\nabla})\underset{\sim}{u}$ and by the incompressibility condition $\underset{\sim}{\nabla}\cdot\underset{\sim}{u} = 0$.

Using convenient alternating direction methods for the time discretization of the Navier-Stokes equations, we shall be able to decouple the difficulties due to the nonlinearity and to the incompressibility, respectively.

For simplicity we suppose from now on that Ω is bounded and that we have (4.45) as boundary condition (with $\underset{\sim}{g}$ satisfying (4.46) and possibly depending upon t).

4.2.2. Time discretization by alternating direction methods.

Let Δt (> 0) be a time discretization step and θ a parameter such that $0 < \theta < 1$.

position techniques still hold for the present transonic flow problem.

4.2. Solution of the Navier-Stokes equations for incompressible viscous fluids.

4.2.1. Formulation of the time dependent Navier-Stokes equations for incompressible fluids.

Let us consider a newtonian incompressible viscous fluid. If Ω and Γ denote the region of the flow ($\Omega \subset \mathbb{R}^N$, $N=2,3$, in practice) and its boundary, respectively, then this flow is governed by the following Navier-Stokes equations

(4.41) $\quad \frac{\partial \underset{\sim}{u}}{\partial t} - \nu\Delta \underset{\sim}{u} + (\underset{\sim}{u}\cdot\underset{\sim}{\nabla})\underset{\sim}{u} + \underset{\sim}{\nabla} p = \underset{\sim}{f}$ in Ω,

(4.42) $\quad \underset{\sim}{\nabla}\cdot\underset{\sim}{u} = 0$ in Ω (incompressibility condition).

In (4.41), (4.42)

(a) $\underset{\sim}{\nabla} = \{\frac{\partial}{\partial x_i}\}_{i=1}^{N}$, $\Delta = \underset{\sim}{\nabla}^2 = \sum_{i=1}^{N} \frac{\partial^2}{\partial x_i^2}$,

(b) $\underset{\sim}{u} = \{u_i\}_{i=1}^{N}$ is the flow velocity,

(c) p is the pressure,

(d) ν is the viscosity of the fluid ($\nu=1/Re$, Re : Reynold's number),

(e) $\underset{\sim}{f}$ is a density of external forces.

In (4.41), $(\underset{\sim}{u}\cdot\underset{\sim}{\nabla})\underset{\sim}{u}$ is a symbolic notation for the nonlinear (vector) term

$$\left\{ \sum_{j=1}^{N} u_j \frac{\partial u_i}{\partial x_j} \right\}_{i=1}^{N}$$

Boundary conditions have to be added ; for example in the case of the airfoil B of Figure 4.3, we have (since the fluid is viscous) the following adherence condition

(4.43) $\quad \underset{\sim}{u} = \underset{\sim}{0}$ on $\partial B = \Gamma_B$;

typical conditions at infinity are

(4.44) $\quad \underset{\sim}{u} = \underset{\sim}{u}_\infty$

where $\underset{\sim}{u}_\infty$ is a constant vector (with regard to the space variables at least).

If Ω is a bounded region of $\mathbb{R}^N$ we may prescribe as boundary condition

(4.45) $\quad \underset{\sim}{u} = \underset{\sim}{g}$ on Γ

(4.35) $z_h^{n+1} = g_h^{n+1} + \gamma_{n+1} z_h^n$,

$n = n+1$, go to (4.31).

The two non trivial steps of algorithms (4.28)-(4.35) are :

(i) the solution of the single variable minimization problem (4.31) ; the corresponding line search can be achieved by dichotomy of Fibonacci methods (see for example [26], [30]). We have to observe that each evaluation of $J_h(\xi_h)$, for a given argument ξ_h, requires the solution of the linear approximate Neumann problem (4.27) to obtain the corresponding y_h.

(ii) the calculation of g_h^{n+1} from ϕ_h^{n+1} which requires the solution of two linear approximate Neumann's problems (namely (4.27) with $\xi_h = \phi_h^{n+1}$ and (4.33)).

Calculation of $J_h'(\phi_h^n)$ and g_h^n : Owing to the importance of step (ii), let us describe in detail the calculation of $J_h'(\phi_h^n)$ and g_h^n (we suppose for simplicity that $\rho_o = 1$) :

We have by differentiation

(4.36) $$\langle J_h'(\xi_h), \delta\xi_h \rangle = \int_\Omega \nabla y_h \cdot \nabla \delta y_h \, dx,$$

where δy_h is from (4.27) the solution of

(4.37) $$\begin{cases} \delta y_h \in V_h \text{ and } \forall v_h \in V_h \text{ we have} \\ \int_\Omega \nabla \delta y_h \cdot \nabla v_h \, dx = \int_\Omega \rho(\xi_h) \nabla \delta\xi_h \cdot \nabla v_h \, dx + \int_\Omega \delta\rho(\xi_h) \nabla \xi_h \cdot \nabla v_h \, dx. \end{cases}$$

Since $\rho(\xi) = \rho_o (1 - K|\nabla\xi_h|^2)^\alpha$ with $K = \frac{1}{C_*^2} \frac{\gamma-1}{\gamma+1}$ and $\alpha = 1/(\gamma-1)$ we have (with $\rho_o = 1$ for simplicity) :

(4.38) $$\delta\rho(\xi_h) = -2K\alpha(1 - K|\nabla\xi_h|^2)^{\alpha-1} \nabla\xi_h \cdot \nabla\delta\xi_h .$$

It follows from (4.36)-(4.38) that

(4.39) $$\begin{cases} \int_\Omega \nabla y_h \cdot \nabla \delta y_h \, dx = \int_\Omega \rho(\xi_h) \nabla y_h \cdot \nabla \delta\xi_h \, dx \\ \qquad - 2K\alpha \int_\Omega (\rho(\xi_h))^{2-\gamma} \nabla\xi_h \cdot \nabla y_h \nabla\xi_h \cdot \nabla\delta\xi_h \, dx. \end{cases}$$

Eventuelly $\langle J_h'(\xi_h), \eta_h \rangle$ can be identified with the linear functional

(4.40) $$\eta_h \to \int_\Omega \rho(\xi_h) \nabla y_h \cdot \nabla \eta_h \, dx - 2K\alpha \int_\Omega (\rho(\xi_h))^{2-\gamma} \nabla\xi_h \cdot \nabla y_h \nabla\xi_h \cdot \nabla\eta_h \, dx.$$

It is quite easy to obtain g_h^{n+1} from ϕ_h^{n+1}, using (4.33),(4.40).

As mentioned in Sec. 3.4, for a simpler nonlinear problem, an efficient discrete Poisson solver will be a basic tool if one uses the above algorithm (4.28)-(4.35), and the various comments about the possibility of using a Poisson solver based on domain decomwhere, in (4.25), X_h is the set of the discrete feasible solutions, and where

(4.26) $$J_h(\xi_h) = \frac{1}{2}\int_\Omega |\underset{\sim}{\nabla} y_h(\xi_h)|^2\, dx,$$

with $y_h(\xi_h)$ (= y_h) the solution of the discrete variational state equation

(4.27) $$\begin{cases} \underline{\text{Find}}\ y_h \in V_h\ \underline{\text{such that}} \\ \int_\Omega \underset{\sim}{\nabla} y_h \cdot \underset{\sim}{\nabla} v_h\, dx = \int_\Omega \rho(\xi_h) \underset{\sim}{\nabla}\xi_h \cdot \underset{\sim}{\nabla} v_h\, dx - \int_\Gamma g_h v_h\, d\Gamma \quad \forall v_h \in V_h. \end{cases}$$

4.1.4.3. <u>Conjugate gradient solution of the least squares problem (4.25)-(4.27).</u>

We follow [1], [3], [29] ; a <u>preconditioned conjugate gradient algorithm</u> for solving (4.25)-(4.27) - with $X_h = V_h$ - is

<u>Step 0 : Initialization</u>

(4.28) $\phi_h^o \in V_h$ <u>given</u>,

<u>then compute</u> g_h^o <u>from</u>

(4.29) $$\begin{cases} g_h^o \in V_h \\ \int_\Omega \underset{\sim}{\nabla} g_h^o \cdot \underset{\sim}{\nabla} v_h = \langle J'(\phi_h^o), v_h\rangle \quad \forall v_h \in V_h, \end{cases}$$

<u>and set</u>

(4.30) $z_h^o = g_h^o$.

<u>Then for</u> $n \geq 0$, <u>assuming</u> ϕ_h^n, g_h^n, z_h^n <u>known</u>, <u>compute</u> $\phi_h^{n+1}, g_h^{n+1}, z_h^{n+1}$ <u>by</u>

<u>Step 1 : Descent</u>

(4.31) <u>Compute</u> $\lambda^n = \operatorname{Arg}\min_{\lambda\in\mathbb{R}} J_h(\phi_h^n - \lambda z_h^n)$,

(4.32) $\phi_h^{n+1} = \phi_h^n - \lambda^n z_h^n$.

<u>Step 2 : Construction of the new descent direction</u>

<u>Define</u> g_h^{n+1} <u>by</u>

(4.33) $$\begin{cases} g_h^{n+1} \in V_h , \\ \int_\Omega \underset{\sim}{\nabla} g_h^{n+1} \cdot \underset{\sim}{\nabla} v_h\, dx = \langle J_h'(\phi_h^{n+1}), v_h\rangle \quad \forall v_h \in V_h , \end{cases}$$

<u>then</u>

(4.34) $$\gamma_{n+1} = \frac{\int_\Omega \underset{\sim}{\nabla} g_h^{n+1} \cdot \underset{\sim}{\nabla}(g_h^{n+1} - g_h^n)\, dx}{\int_\Omega |\underset{\sim}{\nabla} g_h^n|^2\, dx} ,$$

$\mathcal{C}_h$ a standard triangulation of Ω_h we approximate $H^1(\Omega)$ (and in fact $W^{1,p}(\Omega)$, $\forall p \geq 1$) by

$$(4.20) \qquad H_h^1 = \{v_h | v_h \in C^0(\bar{\Omega}_h),\ v_{h|T} \in P_1 \quad \forall T \in \mathcal{C}_h\}$$

where, in (4.20), P_1 = space of polynomials in two variables of degree 1.

We prescribe the value zero for the potential at T.E., this leads to

$$(4.21) \qquad V_h = \{v_h \in H_h^1,\ v_h(\text{T.E.}) = 0\} ;$$

we clearly have

$$(4.22) \qquad \dim H_h^1 = 1 + \dim V_h = \underline{\text{number of vertices of } \mathcal{C}_h}.$$

We approximate then the variational equation (4.15) by

$$(4.23) \qquad \begin{cases} \underline{\text{Find}}\ \phi_h \in V_h\ \underline{\text{such that}} \\ \int_\Omega \rho(\phi_h) \underset{\sim}{\nabla}\phi_h \cdot \underset{\sim}{\nabla} v_h \, dx = \int_\Gamma g_h v_h \, d\Gamma \quad \forall v_h \in V_h \end{cases}$$

where, in (4.23), g_h is an approximation of the function g of (4.13).

The above discrete variational formulation implies that $\rho \frac{\partial\phi}{\partial n}|_\Gamma = g$ is approximately satisfied, automatically.

Let $\mathcal{B}_h = \{w_i\}_{i=1}^{N_h}$ be a vector basis of V_h, then (4.23) is equivalent to the nonlinear finite dimensional system

$$(4.24) \qquad \begin{cases} \phi_h = \sum_{j=1}^{N_h} \phi_j w_j , \\ \int_\Omega \rho(\phi_h) \underset{\sim}{\nabla}\phi_h \cdot \underset{\sim}{\nabla} w_i \, dx = \int_\Gamma g_h w_i \, d\Gamma \quad \forall\ i=1,\dots N_h, \end{cases}$$

where $\phi_j = \phi(P_j)$ $\forall$ $j=1,\dots N_h$, assuming that $\{P_j\}_{j=1}^{N_h}$ is the set of the vertices of $\mathcal{C}_h$ different from T.E.

From the above choice for H_h^1 and V_h, there is no problem of numerical integration since in (4.23), (4.24) $\underset{\sim}{\nabla}\phi_h, \underset{\sim}{\nabla} v_h$ (and therefore $\rho(\phi_h)$) are piecewise constant.

4.1.4.2. Least squares formulation of the discrete problem (4.23).

Combining the results of Secs. 4.1.3.2. and 4.1.4.1, we introduce the following least squares formulation of the approximate problem (4.23) :

$$(4.25) \qquad \underset{\xi_h \in X_h}{\text{Min}}\ J_h(\xi_h),$$

where (cf. ADAMS [13], NECAS [12]) $W^{1,p}(\Omega)$ is for $p \geq 1$, the Sobolev functional space defined by

$$W^{1,p}(\Omega) = \{v | v \in L^p(\Omega) \ , \ \frac{\partial v}{\partial x_i} \in L^p(\Omega) \ \ \forall i\}$$

(with $H^1(\Omega) = W^{1,2}(\Omega)$) ; the function ϕ is determinated only to within an arbitrary constant.

Remark 4.1 : The space $W^{1,\infty}(\Omega)$ is a natural choice for ϕ since physical flows require (among other properties) a positive density ρ ; therefore from (4.11) ϕ has to satisfy

(4.16) $$|\nabla \phi| \leq \delta < \left(\frac{\gamma+1}{\gamma-1}\right)^{1/2} C_* \ \text{a.e. on } \Omega.$$

4.1.3.2. A least squares formulation of (4.15).

For a genuine transonic flow, problem (4.15) is not equivalent to a standard problem of the Calculus of Variations (as it is the case for purely subsonic flows) ; to remedy to this situation and - in some sense - convexify the problem under consideration, we introduce a nonlinear least squares formulation of the transonic flow problem (4.15) as follows :

Let X be a set of feasible transonic flow solutions ; the least squares problem is then

(4.17) $$\min_{\xi \in X} J(\xi),$$

with

(4.18) $$J(\xi) = \frac{1}{2} \int_\Omega |\nabla y(\xi)|^2 \, dx,$$

where, in (4.18), $y(\xi)$ (= y) is solution of the state equation

(4.19) $$\begin{cases} \text{Find } y \in H^1(\Omega)/\mathbb{R} \text{ such that} \\ \int_\Omega \nabla y \cdot \nabla v \, dx = \int_\Omega \rho(\xi) \nabla \xi \cdot \nabla v \, dx - \int_\Gamma g \, v \, d\Gamma \quad \forall v \in H^1(\Omega). \end{cases}$$

If the transonic flow problem has solutions, these solutions solve the least square problem and give the value zero to the objective function J.

4.1.4. Finite element approximation and least squares-conjugate gradient solution of the approximate problems.

We consider only two-dimensional problems but the following methods can be (and have been) applied to three-dimensional problems.

4.1.4.1. Finite element approximation of the nonlinear variational equation (4.15).

We still consider the non-lifting situation of Sec. 4.1.3.1 ; once the flow region has been imbedded in a large bounded domain Ω, we approximate this latter domain by a polygonal domain Ω_h ; with

4.1.3.1. A variational formulation of the continuity equation.

We consider for simplicity the situation of Fig. 4.2 which shows a symmetric flow, subsonic at infinity, around a symmetric airfoil ; thus the Kutta-Joukowsky condition is automatically satisfied.

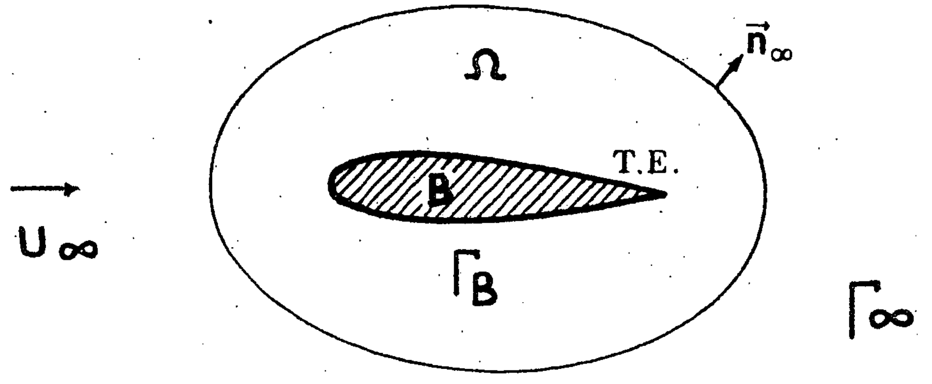


Figure 4.2

For practicality (but other approaches are possible) we imbed the airfoil in a "large" domain ; using the notation of Sec.4.1.2 the continuity equation and the boundary conditions are

(4.10) $$\nabla \cdot \rho(\phi)\nabla\phi = 0 \ \underline{\text{in}}\ \Omega,$$

with

(4.11) $$\rho(\phi) = \rho_o\left(1 - \frac{|\nabla\phi|^2}{\frac{\gamma+1}{\gamma-1}\, c_*^2}\right)^{\frac{1}{\gamma-1}},$$

and

(4.12) $$\rho\,\frac{\partial\phi}{\partial n} = 0 \ \underline{\text{on}}\ \Gamma_B,\quad \rho\,\frac{\partial\phi}{\partial n} = \rho_\infty \mathbf{u}_\infty\cdot\mathbf{n}_\infty \ \underline{\text{on}}\ \Gamma_\infty\ ;$$

on Γ ($= \Gamma_B \cup \Gamma_\infty$) we define g by

(4.13) $$g=0 \ \underline{\text{on}}\ \Gamma_B,\quad g = \rho_\infty \mathbf{u}_\infty\cdot\mathbf{n}_\infty \ \underline{\text{on}}\ \Gamma_\infty\ .$$

We clearly have

(4.14) $$\rho\,\frac{\partial\phi}{\partial n} = g \ \underline{\text{on}}\ \Gamma \ \underline{\text{and}} \int_\Gamma g\, d\Gamma = 0.$$

An equivalent variational formulation of (4.10),(4.14) is

(4.15) $$\begin{cases} \int_\Omega \rho(\phi)\nabla\phi\cdot\nabla v\, dx = \int_\Gamma gv\, d\Gamma \quad \forall v \in H^1(\Omega), \\ \phi \in W^{1,\infty}(\Omega)/\mathbb{R}\ , \end{cases}$$

We have then

(4.4) $\frac{\partial\phi}{\partial n} = \underset{\sim}{u}_\infty \cdot \underset{\sim}{n}$ on Γ_∞,

(4.5) $\frac{\partial\phi}{\partial n} = 0$ on Γ_B.

Since Neumann boundary conditions are involved, the potential is determinated only to within an arbitrary constant. To remedy this we can prescribe the value of ϕ at some point within $\Omega \cup \Gamma$ and, for example, we may conveniently use

(4.6) $\phi = 0$ at the trailing edge T.E. of B.

4.1.2.3. Lifting airfoils and the Kutta-Joukowsky condition.

In fact this condition is not specific of transonic flows since it occurs also for incompressible inviscid flows and compressible inviscid subsonic flows ; we refer to [1],[22] for more information on the numerical treatment of the Kutta-Joukowsky condition for two-dimensional and three-dimensional flows.

4.1.2.4. Shock conditions.

Across a shock the flow has to satisfy the Rankine-Hugoniot conditions

(4.7) $(\rho \underset{\sim}{u} \cdot \underset{\sim}{n})_+ = (\rho \underset{\sim}{u} \cdot \underset{\sim}{n})_-$ (where n is normal at the shock line or surface),

(4.8) the tangential component of the velocity is continuous.

A suitable weak formulation of (4.1)-(4.3) will take (4.7),(4.8) into account.

4.1.2.5. Entropy condition.

It can be formulated as follows (see [28] for more details) :

(4.9) Following the flow we cannot have a positive variation of velocity through a shock, since this would imply a negative variation of Entropy which is an unphysical phenomenon.

The numerical implementation of (4.9) is discussed in [3], [29].

4.1.3. Least squares formulation of the continuous problem.

We do not consider in this paper the practical implementation of (4.9) ; we only discuss the variational formulation of (4.1)-(4.5), (4.8) and of an associate nonlinear least squares formulation.

In the case of flows past bodies we shall suppose that these bodies are sufficiently thin and parallel to the main flow, to not create a wake in the outflow.

4.1.2. Mathematical formulation.

4.1.2.1. Governing Equations.

If Ω is the region of the flow and Γ its boundary, it follows from LANDAU-LIFCHITZ [28] that the flow is governed by

(4.1) $\nabla \cdot \rho \underset{\sim}{u} = 0$ in Ω,

where

(4.2) $$\rho = \rho_o \left(1 - \frac{|\underset{\sim}{u}|^2}{\frac{\gamma+1}{\gamma-1} C_*^2}\right)^{\frac{1}{\gamma-1}},$$

(4.3) $\underset{\sim}{u} = \underset{\sim}{\nabla}\phi$,

where

a) ϕ is the velocity potential ,
b) ρ is the density of the fluid,
c) γ is the ratio of specific heats ($\gamma = 1.4$ in air),
d) C_* is the critical velocity.

4.1.2.2. Boundary conditions.

For an airfoil B (see Fig. 4.1) the flow is assumed to be uniform on Γ_∞ and tangential at Γ_B.

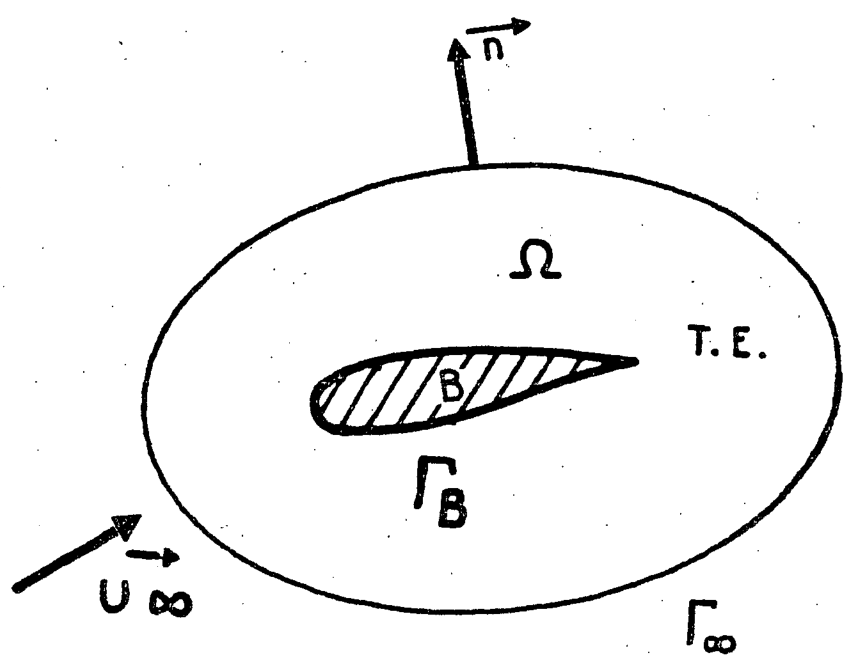


Figure 4.1

$$(3.17) \qquad \langle J'(v),w\rangle = \int_\Omega \nabla\xi\cdot\nabla w \, dx - \langle T'(v)\cdot w,\xi\rangle$$

where, in (2.46), T' is the differential of T, and ξ the solution of (3.3) corresponding to v. Therefore $J'(v) \in H^{-1}(\Omega)$ may be identified with the linear functional defined on $H^1_0(\Omega)$ by

$$(3.18) \qquad w \to \int_\Omega \nabla\xi\cdot\nabla w \, dx - \langle T'(v)\cdot w,\xi\rangle \ .$$

It follows then from (3.14),(3.17),(3.18) that g^n is the solution of the following linear variational problem

$$(3.19) \qquad \begin{cases} \underline{\text{Find}}\ g^n \in H^1_0(\Omega)\ \underline{\text{such that}}\ \forall w \in H^1_0(\Omega) \\ \int_\Omega \nabla g^n\cdot\nabla w \, dx = \int_\Omega \nabla\xi^n\cdot\nabla w - \langle T'(\phi^n)\cdot w,\xi^n\rangle \ , \end{cases}$$

where ξ^n is the solution of (3.3) corresponding to $v = \phi^n$.

We conclude this subsection 3.4 by observing that, from the points (i) and (ii) of the above discussion, an <u>efficient Poisson solver</u> will be a basic tool in the solution of (3.1)(in fact of <u>an approximation</u> of it) by the conjugate gradient algorithm (3.9)-(3.16) ; for such a solver we can use one of those domain decomposition techniques discussed in Sec. 2. Actually, since the conjugate gradient algorithm (3.9)-(3.16) implies the repetitive solution at each iteration of several linear Poisson-problems, it can be worthwhile to use the quasi-direct method of Sec. 2.3.8, with the matrix A_h of (2.210) constructed using (2.177)-(2.179),and then factorized once and for all by a Cholesky method for example.

4. - <u>APPLICATION TO THE LEAST SQUARES SOLUTION OF NONLINEAR PROBLEMS IN FLUID DYNAMICS</u>.

4.1. <u>Numerical simulation of transonic potential flows for compressible inviscid fluids</u>.

4.1.1. <u>Generalities</u>. <u>The physical problem</u>.

The numerical simulation of <u>transonic</u>, <u>potential flows</u> of <u>compressible</u>, <u>inviscid fluids</u> is a non trivial problem since

a) The equations governing these flows are <u>nonlinear</u> and of <u>changing type</u> (<u>elliptic</u> in the <u>subsonic</u> part of the flow, <u>hyperbolic</u> in the <u>supersonic</u> part).

b) <u>Shocks</u> may exist in these flows corresponding to <u>discontinuities of velocity</u>, <u>pressure and density</u>.

c) An <u>Entropy Condition</u> has to be included in order to eliminate <u>rarefaction shocks</u> since they correspond to <u>unphysical</u> situations.

Concerning the fluids and flows under consideration we suppose that : the fluids are <u>compressible</u> and <u>inviscid</u> and their flows are <u>potential</u> and therefore <u>quasi-isentropic</u>, with <u>weak shocks</u> only ; in fact this is only an approximation since usually the flow is no longer potential after a schock(cf. LANDAU-LIFCHITZ [28]).

Step 0 :

(3.9) $\phi^o \in H^1_o(\Omega)$ given,

then compute $g^o \in H^1_o(\Omega)$ from

(3.10) $$\begin{cases} -\Delta g^o = J'(\phi^o) \text{ in } \Omega, \\ g^o = 0 \text{ on } \Gamma, \end{cases}$$

and set

(3.11) $w^o = g^o$.

Then for $n \geq 0$, assuming ϕ^n, g^n, w^n known, compute $\phi^{n+1}, g^{n+1}, w^{n+1}$ by

Step 1 : Descent

(3.12) $$\lambda^n = \underset{\lambda \in \mathbb{R}}{\text{Arg min}}\ J(\phi^n - \lambda w^n),$$

(3.13) $$\phi^{n+1} = \phi^n - \lambda^n w^n.$$

Step 2 : Construction of the new descent direction

Define $g^{n+1} \in H^1_o(\Omega)$ by

(3.14) $$-\Delta g^{n+1} = J'(\phi^{n+1}) \text{ in } \Omega,\ g^{n+1} = 0 \text{ on } \Gamma,$$

then

(3.15) $$\gamma^{n+1} = \frac{\int_\Omega \nabla g^{n+1} \cdot \nabla (g^{n+1} - g^n)\, dx}{\int_\Omega |\nabla g^n|^2\, dx},$$

(3.16) $$g^{n+1} = g^n + \gamma^{n+1} w^{n+1},$$

$n = n+1$, go to (3.12).

The two non trivial steps of algorithm (3.9)-(3.16) are :

(i) The solution of the single variable minimization problem (3.12) ; the corresponding line search can be achieved by dichotomy or Fibonacci methods. We observe that each evaluation of $J(v)$, for a given argument v, requires the solution of the linear Dirichlet problem (3.3) to obtain the corresponding ξ.

(ii) The calculation of g^{n+1} from ϕ^{n+1} which requires the solution of two linear Dirichlet problems (namely (3.3) with $v=\phi^{n+1}$ and (3.14)).

Calculation of $J'(\phi^n)$ and g^n : We refer to [2, Sec. 2.4] where we prove that $\forall v, w \in H^1_o(\Omega)$ we have

(3.2) $$\min_{v \in H_0^1(\Omega)} \|\Delta v + T(v)\|_{-1}$$

(where the H^{-1}-norm $\|\cdot\|_{-1}$ has been defined in Sec. 2.1). It is clear that if (3.1) has a solution, then this solution is also a solution of (3.2) for which the cost function will vanish. Let us introduce $\xi \in H_0^1(\Omega)$ by

(3.3) $$\begin{cases} \Delta\xi = \Delta v + T(v) \ \underline{\text{in}}\ \Omega, \\ \xi = 0 \ \underline{\text{on}}\ \Gamma, \end{cases}$$

then (3.2) reduces to

(3.4) $$\min_{v \in H_0^1(\Omega)} \|\Delta\xi\|_{-1}$$

where, in (3.4), ξ is a function of v via (3.3) ; actually it can be proved that if $\|\cdot\|_{-1}$ is defined by (2.13) with $<\cdot,\cdot>$ obeying (2.12), then

(3.5) $$\|\Delta v\|_{-1} = \|v\|_{H_0^1(\Omega)} \quad \forall v \in H_0^1(\Omega).$$

It follows then from (3.5) that (3.4) may be formulated also by

(3.6) $$\min_{v \in H_0^1(\Omega)} \int_\Omega |\nabla\xi|^2 \, dx$$

where ξ is still a function of v through (3.3).

Remark 3.1 : Nonlinear boundary value problems have been treated by CEA-GEYMONAT [24] and LOZI [25] using formulations close to (3.3),(3.6).

3.4. Conjugate gradient solution of the least squares problem (3.3), (3.6).

Let us define $J : H_0^1(\Omega) \to \mathbb{R}$ by

(3.7) $$J(v) = \frac{1}{2} \int_\Omega |\nabla\xi|^2 \, dx,$$

where ξ is a function of v in accordance with (3.3) ; then (3.6) may also be written as

(3.8) $$\min_{v \in H_0^1(\Omega)} J(v).$$

To solve (3.8) we shall use a conjugate gradient algorithm. Among the possible conjugate gradient algorithms we have selected the Polak-Ribière version (cf. [26]), since this algorithm produced the best performances (compared to other variants) in the numerical tests we made (the good performances of the Polak-Ribière algorithm are discussed in [27]). Let us denote by $J'(\cdot)$ the differential of $J(\cdot)$; then the Polak-Ribière version of the conjugate gradient method, applied to the solution of (3.8) is

(2.219) $$a_{22}^{k\ell} = \tilde{a}_h(\{0,w_{2k}\},\{0,w_{2\ell}\}) \;;$$

A_{ij} is clearly a $N_{ih} \times N_{jh}$ matrix.

The linear system (2.210) can be solved using a Cholesky decomposition of A_h taking advantage of its special structure ; a "good" subdomain numbering is shown on Figure 2.6, resulting in a "nice" sparse block structure (the submatrices A_{ij} can be full matrices).

As a final remark we would like to mention that we can take advantage of the special structure of matrix A_h if one uses parallel processors for the construction and the Cholesky factorization of this matrix. The same observation holds for the construction of the right hand side b_h, also ; this is of fundamental importance since in the applications to the solution of nonlinear problems, discussed in Sec. 3, we shall have to update these right hand sides very often.

3. - APPLICATION OF DOMAIN DECOMPOSITION TO THE LEAST SOUARES CONJUGATE GRADIENT SOLUTION OF NONLINEAR BOUNDARY VALUE PROBLEMS.

3.1. Synopsis.

In this Sec. 3 we shall see that the solution of nonlinear boundary value problems can be reduced - via a least squares formulation coupled to a preconditioned conjugate gradient algorithm - to the solution of a sequence of Poisson problems, solvable themselves by the domain decomposition methods discussed in the previous sections.

In the following subsections we shall discuss the application of the above principles to the solution of a quite simple nonlinear model Dirichlet problem ; actually,as we shall see in Secs.4 and 5,the same ideas apply also to more complicated nonlinear problems.

We follow partly in this Section [22, Sec. 2],[2, Sec. 2.3] (see also GLOWINSKI[23, Chap. 7]).

3.2. Formulation of the nonlinear test problem.

Let $\mathbb{R}^N$ be a bounded domain with a smooth boundary. We consider the nonlinear Dirichlet problem

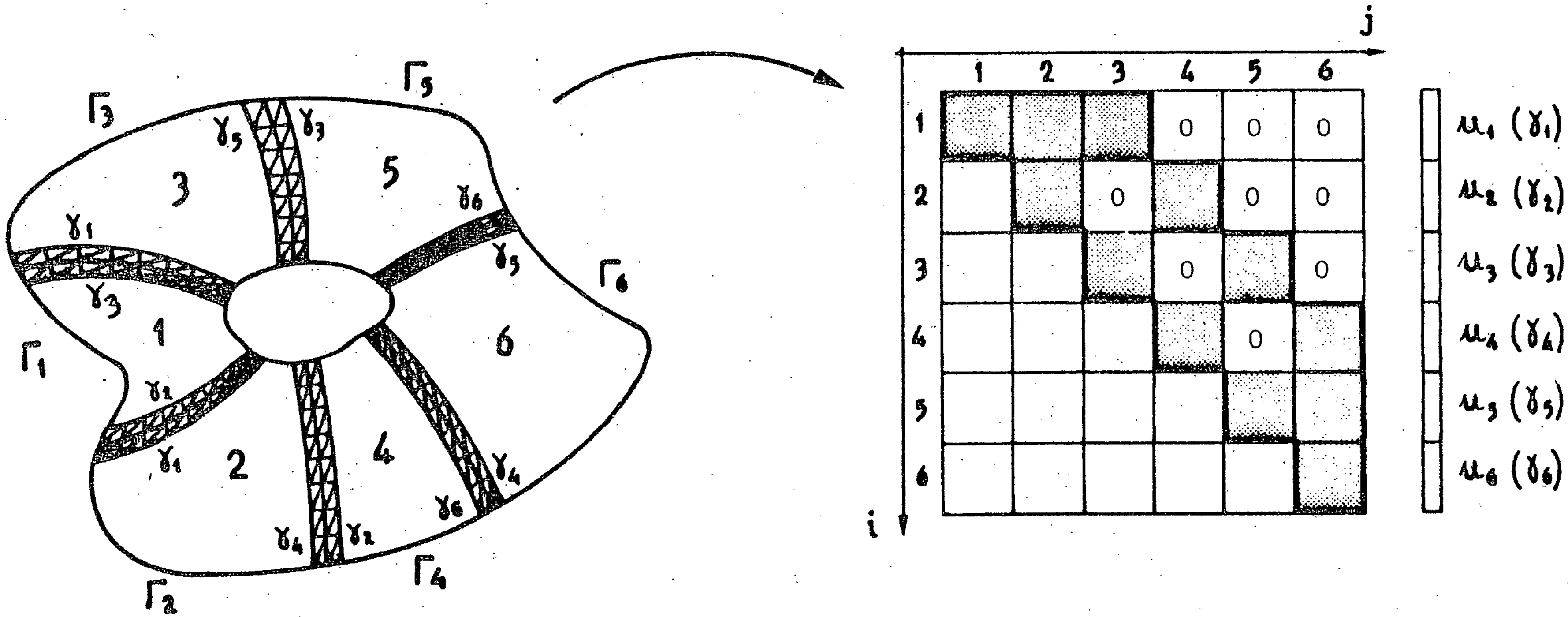
(3.1) $$\begin{cases} -\Delta\phi - T(\phi) = 0 \text{ in } \Omega, \\ \phi = 0 \text{ on } \Gamma. \end{cases}$$

We do not discuss here the existence and uniqueness properties of the solution of (3.1), since we do not want to be too specific about operator T.

We shall concentrate on conjugate gradient methods with preconditioning via convenient least square formulations.

3.3. An H^{-1} least squares method for solving (3.1).

A natural least squares formulation for solving the model problem (3.1) is



A "GOOD" SUBDOMAIN NUMBERING

BLOCK STRUCTURE OF A_h SHOWING ITS SPARSITY

(we have represented the upper part of A_h only)

FIGURE 2.6

$$(2.211)\qquad \Lambda_h = \begin{bmatrix} \lambda_{1h}(P_{11}) \\ \vdots \\ \lambda_{1h}(P_{1N_{1h}}) \\ \lambda_{2h}(P_{21}) \\ \vdots \\ \lambda_{2h}(P_{2N_{2h}}) \end{bmatrix}$$

(with $\lambda_h = \{\lambda_{1h},\lambda_{2h}\} \in M_h$ the solution of (2.175)) ; then

$$(2.212)\qquad b_h = \begin{bmatrix} b_{11} \\ \vdots \\ b_{1N_{1h}} \\ b_{21} \\ \vdots \\ b_{2N_{2h}} \end{bmatrix}$$

with b_{1k} defined, from (2.175), by (2.180) with $\mu_h = \{w_{1k},0\}$, i.e.

$$(2.213)\qquad b_{1k} = -(\tilde{J}_h'(0),\{w_{1k},0\}) \ ;$$

we should have similarly

$$(2.214)\qquad b_{2k} = -(\tilde{J}_h'(0),\{0,w_{2k}\}).$$

Finally, the matrix A_h is <u>symmetric</u>, <u>positive definite</u>. It is also <u>block sparse</u> if the number of subdomains is ≥3 (as shown on Figure 2.6) ; in the simpler case of a two subdomain decomposition we have

$$(2.215)\qquad A_h = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right)$$

with $A_{21} = A_{12}^t$; $a_{ij}^{k\ell}$ is the generic element of A_{ij} and we should use (2.179) to compute it ; we have indeed

$$(2.216)\qquad a_{11}^{k\ell} = \tilde{a}_h(\{w_{1k},0\},\{w_{1\ell},0\}),$$

$$(2.217)\qquad a_{12}^{k\ell} = \tilde{a}_h(\{0,w_{2\ell}\},\{w_{1k},0\}),$$

$$(2.218)\qquad a_{21}^{k\ell} = \tilde{a}_h(\{0,w_{2k}\},\{w_{1\ell},0\}),$$

(2.206) $$\lambda_n = \frac{\sum_{i=1}^{2} \int_{\Omega_{ii}} |\nabla g_{ih}^{n+1}|^2 \, dx}{\sum_{i=1}^{2} \int_{\Omega_{ii}} |\nabla g_{ih}^{n}|^2 \, dx} ,$$

(2.207) $$w_h^{n+1} = g_h^{n+1} + \lambda_n w_h^n ,$$

(2.208) do n=n+1, go to (2.201).

To compute $\tilde{a}_h(w_h^n, w_h)$ we should use (2.193), (2.194) and, instead of (2.195),

(2.209) $$\begin{cases} \tilde{a}_h(w_h^n,w_h) = \int_{\Omega_{12}} \nabla(\eta_{2h}^n - \eta_{1h}^n)\cdot\nabla w_{2h} \, dx + \int_{\Omega_{12}} (\eta_{2h}^n - \eta_{1h}^n) w_{2h} \, dx + \\ \quad + \int_{\Omega_{12}} \nabla(\eta_{1h}^n - \eta_{2h}^n)\cdot\nabla w_{1h} \, dx + \int_{\Omega_{12}} (\eta_{1h}^n - \eta_{2h}^n) w_{1h} \, dx \\ \quad - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla\pi_h^n \cdot \nabla w_{ih} \, dx \quad \forall w_h \in M_h . \end{cases}$$

We observe that each iteration requires the solution of 4 discrete Dirichlet problems, plus the solution in M_h of the "small" linear variational problem (2.205), equivalent to a linear system whose matrix is symmetric, positive definite and sparse.

2.3.8. Quasi direct solution of the least-squares problem (2.146)-(2.148).

It follows from Sec. 2.3.6.E that the solution of the least square problem (2.146)-(2.148) can be reduced to

(i) The solution of two discrete Dirichlet problems on each subdomain Ω_{ii} in order to compute the right side of (2.175),

(ii) The solution of the linear system equivalent to (2.175),

(iii) One discrete Dirichlet problem on each Ω_{ii} to compute the solution of (2.146)-(2.148), once the solution of (2.175) is known.

Let us concentrate on (ii) ; actually (2.175) is equivalent to a linear system, say

(2.210) $$A_h \Lambda_h = b_h .$$

With the notation of Sec. 2.3.6.D we have in (2.210) (if i=1,2)

$$(2.200)\quad \begin{cases} p^o_{ih} \in H^1_{oih}, \\ \int_{\Omega_{ii}} \nabla p^o_{ih}\cdot\nabla\phi_h\, dx = \int_{\Omega_{12}} \nabla(y^o_{ih}-y^o_{jh})\cdot\nabla\phi_h\, dx + \int_{\Omega_{12}} (y^o_{ih}-y^o_{jh})\phi_h\, dx \\ \forall\phi_h \in H^1_{oih} \end{cases}$$

where $j=i+(-1)^{i-1}$.

Define now $g^o_h = \{g^o_{1h}, g^o_{2h}\}$ as the solution in M_h of

$$(2.201)\quad \begin{cases} \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla g^o_{ih}\cdot\nabla w_{ih}\, dx = \int_{\Omega_{12}} \nabla(y^o_{2h}-y^o_{1h})\cdot\nabla w_{2h}\, dx + \\ + \int_{\Omega_{12}} (y^o_{2h}-y^o_{1h})w_{2h}\, dx + \int_{\Omega_{12}} \nabla(y^o_{1h}-y^o_{2h})\cdot\nabla w_{1h}\, dx + \\ + \int_{\Omega_{12}} (y^o_{1h}-y^o_{2h})w_{1h} dx - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla p^o_{ih}\cdot\nabla w_{ih}\, dx \\ \forall w_h = \{w_{1h}, w_{2h}\} \in M_h. \end{cases}$$

Set then

$$(2.202)\qquad w^o_h = g^o_h.$$

Assuming that $u^n_h = \{u^n_{ih}\}^2_{i=1}$, $g^n_h = \{g^n_{ih}\}^2_{i=1}$, $w^n_h = \{w^n_{ih}\}^2_{i=1}$ are known, we obtain u^{n+1}_h, g^{n+1}_h, w^{n+1}_h by

Step 1 : Steepest descent

$$(2.203)\qquad \rho_n = \frac{\sum_{i=1}^{2} \int_{\Omega_{ii}} |\nabla g_{ih}|^2\, dx}{\tilde{a}_h(w^n_h, w^n_h)},$$

$$(2.204)\qquad u^{n+1}_{ih} = u^n_{ih} - \rho_n(w^n_{ih})|_{\gamma_i}, \quad \forall i=1,2.$$

Step 2 : Calculation of the new descent direction

Define $g^{n+1}_h = \{g^{n+1}_{1h}, g^{n+1}_{2h}\}$ as the solution of

$$(2.205)\quad \begin{cases} \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla g^{n+1}_{ih}\cdot\nabla w_{ih}\, dx = \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla g^n_{ih}\cdot\nabla w_{ih}\, dx - \rho_n\tilde{a}_h(w^n_h, w_h) \\ \forall w_h = \{w_{1h}, w_{2h}\} \in M_h, \end{cases}$$

then

$$(2.195)\quad \begin{cases} \tilde{a}_h(\tilde{w}_h^n,\tilde{w}_h) = \int_{\Omega_{12}} \nabla(\eta_{2h}^n-\eta_{1h}^n)\cdot\nabla\tilde{w}_{2h}\,dx + \int_{\Omega_{12}} (\eta_{2h}^n-\eta_{1h}^n)\tilde{w}_{2h}\,dx + \\ \quad + \int_{\Omega_{12}} \nabla(\eta_{1h}^n-\eta_{2h}^n)\cdot\nabla\tilde{w}_{1h}\,dx + \int_{\Omega_{12}} (\eta_{1h}^n-\eta_{2h}^n)\tilde{w}_{1h}\,dx \\ \quad - \sum_{i=1}^{2}\int_{\Omega_{ii}} \nabla\pi_{ih}^n\cdot\nabla\tilde{w}_{ih}\,dx \quad \forall w_h \in V_{oh}, \end{cases}$$

with $\tilde{w}_h$ as above.

We observe that each iteration requires the solution of the 4 discrete Dirichlet problems (2.193),(2.194).

2.3.7. Conjugate gradient solution of the least squares problem (2.146)-(2.148). A second algorithm.

We consider now the discrete analogous of Sec. 2.3.3 ; the key idea is to replace, in the conjugate gradient algorithm discussed in Sec. 2.3.6, the M_h-scalar product

$$(2.196)\quad \{\mu_h,\mu_h'\} \to \sum_{i=1}^{2}\int_{\gamma_i} \mu_{ih}\mu_{ih}'\,d\gamma \quad \forall \mu_h,\mu_h' \in M_h$$

by the following one

$$(2.197)\quad \{\mu_h,\mu_h'\} \to \sum_{i=1}^{2}\int_{\Omega_{ii}} \nabla\mu_{ih}\cdot\nabla\mu_{ih}'\,dx \quad \forall \mu_h,\mu_h' \in M_h.$$

Since the functions of M_h vanish outside a small neighborhood of the γ_i's (see Fig. 2.5), using (2.197) is no more costly, per iteration, than using (2.196) ; moreover from the numerical tests to follow, it seems that (2.197) has, in general, better preconditioning (scaling) properties than (2.196), making the corresponding algorithms faster than those corresponding to (2.196) (which are described in Sec. 2.3.6).

Using the scalar product (2.197) we obtain the following variant of (2.182)-(2.192) (it is still a conjugate gradient algorithm) :

Step 0 : Initialization

$$(2.198)\quad u_h^o = \{u_{1h}^o,u_{2h}^o\} \in V_{1h}\times V_{2h}, \text{ arbitrarily given,}$$

then, for i=1,2, solve

$$(2.199)\quad \begin{cases} y_{ih}^o \in H_{ih}^1,\ y_{ih}^o = g_{ih} \text{ on } \partial\Omega_{ii}\cap\Gamma,\ y_{ih}^o = u_{ih}^o \text{ on } \gamma_i, \\ \int_{\Omega_{ii}} \nabla y_{ih}^o\cdot\nabla\phi_h\,dx = \int_{\Omega_{ii}} f_{ih}\phi_h\,dx \quad \forall \phi_h \in H_{oih}^1 \end{cases}$$

(which has a unique solution in H_{ih}^1) and where, in (2.187), $\tilde{w}_h^n$ is the unique extension of w_h^n in M_h.

Step 2 : Calculation of the new descent direction

Define $g_h^{n+1} = \{g_{1h}^{n+1}, g_{2h}^{n+1}\}$ by solving

$$(2.189)\quad \begin{cases} \sum_{i=1}^{2} \int_{\gamma_i} g_{ih}^{n+1} w_{ih}\, d\gamma = \sum_{i=1}^{2} \int_{\gamma_i} g_{ih}^{n} w_{ih} - \rho_n \tilde{a}_h(\tilde{w}_h^n, \tilde{w}_h) \\ \forall w_h = \{w_{1h}, w_{2h}\} \in V_{01h} \times V_{02h}, \end{cases}$$

then

$$(2.190)\quad \lambda_n = \frac{\sum_{i=1}^{2} \int_{\gamma_i} |g_{ih}^{n+1}|^2\, d\gamma}{\sum_{i=1}^{2} \int_{\gamma_i} |g_{ih}^{n}|^2\, d\gamma},$$

$$(2.191)\quad w_h^{n+1} = g_h^{n+1} + \lambda_n w_h^n ,$$

(2.192) do n=n+1, go to (2.187) .

We detail now the calculation of $\tilde{a}(\tilde{w}_h^n, \tilde{w}_h)$:

Compute $\eta_h^n = \{\eta_{1h}^n, \eta_{2h}^n\} \in H_{1h}^1 \times H_{2h}^1$ via the solution for i=1,2, of

$$(2.193)\quad \begin{cases} \int_{\Omega_{ii}} \nabla\eta_{ih}^n \cdot \nabla\phi_h\, dx = 0 \quad \forall \phi_h \in H_{oih}^1 , \\ \eta_{ih}^n = 0 \text{ on } \partial\Omega_{ii} \cap \Gamma,\ \eta_{ih}^n = w_{ih}^n \text{ on } \gamma_i ; \end{cases}$$

Compute then $\pi_h^n = \{\pi_{1h}^n, \pi_{2h}^n\} \in H_{01h}^1 \times H_{02h}^1$ via the solution, for i=1,2, of

$$(2.194)\quad \begin{cases} \pi_{ih}^n \in H_{oih}^1, \\ \int_{\Omega_{ii}} \nabla\pi_{ih}^n \cdot \nabla\phi_h\, dx = \int_{\Omega_{12}} \nabla(\eta_{ih}^n - \eta_{jh}^n) \cdot \nabla\phi_h\, dx + \int_{\Omega_{12}} (\eta_{ih}^n - \eta_{jh}^n)\phi_h\, dx \\ \forall \phi_h \in H_{oih}^1 , \end{cases}$$

where $j = i + (-1)^{i-1}$; we finally have (from (2.179)) that then, for i=1,2, solve

$$(2.183)\quad \begin{cases} y^o_{ih} \in H^1_{ih},\ y^o_{ih} = g_{ih} \text{ on } \partial\Omega_{ii} \cap \Gamma,\ y^o_{ih} = u^o_{ih} \text{ on } \gamma_i, \\ \int_{\Omega_{ii}} \nabla y^o_{ih} \cdot \nabla \phi_h \, dx = \int_{\Omega_{ii}} f_{ih}\phi_h \, dx \quad \forall \phi_h \in H^1_{oih} \end{cases}$$

(which has a unique solution in H^1_{ih}) and

$$(2.184)\quad \begin{cases} p^o_{ih} \in H^1_{oih}, \\ \int_{\Omega_{ii}} \nabla p^o_{ih} \cdot \nabla\phi_h \, dx = \int_{\Omega_{12}} \nabla(y^o_{ih} - y^o_{jh}) \cdot \nabla\phi_h dx + \\ + \int_{\Omega_{12}} (y^o_{ih} - y^o_{ih})\phi_h \, dx \quad \forall\phi_h \in H^1_{oih}, \end{cases}$$

where $j=i+(-1)^{i-1}$.

Define now $g^o_h = \{g^o_{1h}, g^o_{2h}\}$ as the solution in $V_{01h} \times V_{02h}$ of

$$(2.185)\quad \begin{cases} \sum_{i=1}^{2} \int_{\gamma_i} g^o_{ih} w_{ih} d\gamma = \int_{\Omega_{12}} \nabla(y^o_{2h} - y^o_{1h}) \cdot \nabla\tilde{w}_{2h} \, dx + \\ + \int_{\Omega_{12}} (y^o_{2h} - y^o_{1h})\tilde{w}_{2h} \, dx + \int_{\Omega_{12}} \nabla(y^o_{1h} - y^o_{2h}) \cdot \nabla\tilde{w}_{1h} \, dx \\ + \int_{\Omega_{12}} (y^o_{1h} - y^o_{2h})\tilde{w}_{1h} \, dx - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla p^o_{ih} \cdot \nabla\tilde{w}_{ih} dx \\ \forall w_h = \{w_{1h}, w_{2h}\} \in V_{oh}, \end{cases}$$

where $\tilde{w}_h = \{\tilde{w}_{1h}, \tilde{w}_{2h}\}$ is (and will be in the sequel) the unique extension of w_h in M_h.

Set then

$$(2.186)\quad w^o_h = g^o_h .$$

Assuming that $u^n_h = \{u^n_{ih}\}^2_{i=1}$, $g^n_h = \{g^n_{ih}\}^2_{i=1}$, $w^n_h = \{w^n_{ih}\}^2_{i=1}$ are known, we obtain u^{n+1}_h, g^{n+1}_h, w^{n+1}_h by

Step 1 : Steepest descent

$$(2.187)\quad \rho_n = \frac{\sum_{i=1}^{2} \int_{\gamma_i} |g^n_{ih}|^2 \, d\gamma}{\tilde{a}_h(\tilde{w}^n_h, \tilde{w}^n_h)} ,$$

$$(2.188)\quad u^{n+1}_h = u^n_h - \rho_n w^n_h ,$$

We also have

$$(2.179)\quad\begin{cases}\tilde{a}_h(\mu_h',\mu_h) = \tilde{a}_h(\mu_h,\mu_h') = \int_{\Omega_{12}} \nabla(y_{2h}-y_{1h})\cdot\nabla\mu_{2h}'\,dx + \\ + \int_{\Omega_{12}} (y_{2h}-y_{1h})\mu_{2h}'\,dx + \int_{\Omega_{12}} \nabla(y_{1h}-y_{2h})\cdot\nabla\mu_{1h}'\,dx \\ + \int_{\Omega_{12}} (y_{1h}-y_{2h})\mu_{1h}'\,dx - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla p_{ih}\cdot\nabla\mu_{ih}'\,dx = \\ = \int_{\Omega_{12}} \nabla(y_{2h}'-y_{1h}')\cdot\nabla\mu_{2h}\,dx + \int_{\Omega_{12}} (y_{2h}'-y_{1h}')\mu_{2h}\,dx + \\ \int_{\Omega_{12}} \nabla(y_{1h}'-y_{2h}')\cdot\nabla\mu_{1h}\,dx + \int_{\Omega_{12}} (y_{1h}'-y_{2h}')\mu_{1h}\,dx - \\ - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla p_{ih}'\cdot\nabla\mu_{ih}\,dx,\end{cases}$$

where p_{ih}, p_{ih}' are the solution of (2.161), corresponding to y_{ih}, y_{ih}' given by (2.177),(2.178) respectively.

The right hand side of (2.175) can be computed explicitely since

$$(2.180)\quad\begin{cases}(\tilde{J}_h'(0),\mu_h) = \int_{\Omega_{12}} \nabla(y_{2h}^o-y_{1h}^o)\cdot\nabla\mu_{2h}\,dx + \int_{\Omega_{12}} (y_{2h}^o-y_{1h}^o)\mu_{2h}\,dx + \\ + \int_{\Omega_{12}} \nabla(y_{1h}^o-y_{2h}^o)\cdot\nabla\mu_{1h}\,dx + \int_{\Omega_{12}} (y_{1h}^o-y_{2h}^o)\mu_{1h}\,dx - \\ - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla p_{ih}^o\cdot\nabla\mu_{ih}\,dx \quad \forall\mu_h \in M_h\,,\end{cases}$$

where $y_h^o = \{y_{1h}^o, y_{2h}^o\}$ is obtained from (2.173) with $\tilde{v}_{ih}$ obeying the following hypotheses on $\partial\Omega_{ii}$:

(2.181) $\tilde{v}_{ih} = g_{ih}$ <u>on</u> $\partial\Omega_{ii} \cap \Gamma$, $P_i^M(\tilde{v}_{ih}) = 0$;

then p_{ih}^o is obtained by taking $y_{ih} = y_{ih}^o$ in (2.161).

F) - <u>Application to algorithm</u> (2.150)-(2.158).

Using the isomorphism between V_{oh} and M_h the conjugate gradient algorithm (2.150)-(2.158) can be written as follows (it is a discrete variant of algorithm (2.95)-(2.108) of Sec. 2.3.2 F) :

<u>Step 0</u> : <u>Initialization</u>

(2.182) $u_h^o = \{u_{1h}^o, u_{2h}^o\}$ <u>is arbitrarily given in</u> $V_{1h} \times V_{2h}$,

$$(2.173)\quad \begin{cases} \int_{\Omega_{ii}} \nabla y_{ih}\cdot\nabla\phi_h \, dx = \int_{\Omega_{ii}} f_{ih}\phi_h \, dx \quad \forall\phi_h \in H^1_{oih}, \\ y_{ih} = \tilde{v}_{ih} \ \underline{\text{on}}\ \partial\Omega_{ii}, \ \underline{\text{with}}\ \tilde{v}_{ih} \in \tilde{V}_{ih} \ \underline{\text{such that}} \\ \tilde{v}_{ih} = g_{ih} \ \underline{\text{on}}\ \partial\Omega_{ii}\cap\Gamma, \ P^M_i(\tilde{v}_{ih}) = \mu_{ih}. \end{cases}$$

The unique solution of the least squares problem is also the solution of the following <u>linear variational problem in</u> M_h.

$$(2.174)\quad \begin{cases} \underline{\text{Find}}\ \lambda_h = \{\lambda_{1h},\lambda_{2h}\} \in M_h \ \underline{\text{such that}} \\ (\tilde{J}'_h(\lambda_h),\mu_h) = 0 \quad \forall\mu_h \in M_h. \end{cases}$$

Actually, since $\mu_h \to \tilde{J}_h(\mu_h)$ is <u>quadratic</u>, (2.174) is equivalent to a <u>linear system</u> whose variational formulation is given by

$$(2.175)\quad \begin{cases} \lambda_h \in M_h, \\ \tilde{a}_h(\lambda_h,\mu_h) = -(\tilde{J}'_h(0),\mu_h) \quad \forall\mu_h \in M_h, \end{cases}$$

where $\tilde{a}_h(\cdot,\cdot)$ is a bilinear form <u>symmetric</u> and <u>positive definite</u> over $M_h \times M_h$; it is not known explicitely but however it is quite easy to compute $\tilde{a}_h(\mu_h,\mu'_h)$ for every pair $\{\mu_h,\mu'_h\} \in M_h \times M_h$, since we have

$$(2.176)\quad \tilde{a}_h(\mu_h,\mu'_h) = \int_{\Omega_{12}} \{\nabla(y'_{2h}-y'_{1h})\cdot\nabla(y_{2h}-y_{1h})+(y'_{2h}-y'_{1h})(y_{2h}-y_{1h})\}dx$$

where y_{ih},y'_{ih} are, for $i=1,2$, the solutions of the following discrete Dirichlet problems in Ω_{ii}

$$(2.177)\quad \begin{cases} y_{ih} \in H^1_{ih}, \ y_{ih}=0 \ \underline{\text{on}}\ \partial\Omega_{ii}\cap\Gamma, \ y_{ih} = \mu_{ih} \ \underline{\text{on}}\ \gamma_i, \\ \int_{\Omega_{ii}} \nabla y_{ih}\cdot\nabla\phi_h \, dx = 0 \quad \forall\phi_h \in H^1_{oih}, \end{cases}$$

$$(2.178)\quad \begin{cases} y'_{ih} \in H^1_{ih}, \ y'_{ih}=0 \ \underline{\text{on}}\ \partial\Omega_{ii}\cap\Gamma, \ y'_{ih}=\mu'_{ih} \ \underline{\text{on}}\ \gamma_i, \\ \int_{\Omega_{ii}} \nabla y'_{ih}\cdot\nabla\phi_h \, dx = 0 \quad \forall\phi_h \in H^1_{oih}, \end{cases}$$

respectively.

Suppose that those vertices of $\mathcal{T}_h$ located on γ_i, considered in (2.165), have been numbered from 1 to N_{ih} as follows :

$$\{P_{ij}\}_{j=1}^{N_{ih}} \quad \underline{\text{(where}}\ P_{ij} \in \gamma_i,\ P_{ij} \notin \partial\Omega_{ii} \cap \Gamma) \ ;$$

then the following basis $\mathcal{B}_{ih}$ of M_{ih} plays an important role in the sequel

$$(2.167) \quad \mathcal{B}_{ih} = \{w_{ij}\}_{j=1}^{N_{ih}} \ ,$$

where the w_{ij} are (uniquely) defined by

$$(2.168) \quad \begin{cases} w_{ij} \in M_{ih} \quad \forall j=1,\dots N_{ih}, \\ w_{ij}(P_{ij}) = 1,\ w_{ij}(Q) = 0 \quad \forall Q \text{ vertex of } \mathcal{T}_{ih},\ Q \neq P_{ij} \ ; \end{cases}$$

the support of some w_{ij} are shown on Figure 2.5.

E) - <u>Formulation of the least squares problem(2.146)-(2.148) as a variational problem in M_h.</u>

Let $\phi_{ih} \in \tilde{V}_{ih}$; from (2.164) we have

$$(2.169) \quad \phi_{ih} = \phi_{ih}^M + \phi_{ih}^S \ ,\ \phi_{ih}^M \in M_{ih},\ \phi_{ih}^S \in S_{ih},$$

where ϕ_{ih}^M, ϕ_{ih}^S are uniquely defined ; using (2.169) we introduce then the projector $P_i^M : \tilde{V}_{ih} \to M_{ih}$, defined by

$$(2.170) \quad P_i^M(\phi_{ih}) = \phi_{ih}^M \quad \forall \phi_{ih} \in \tilde{V}_{ih}.$$

By analogy with GLOWINSKI-PIRONNEAU [10] and [11] (for the first biharmonic problem and the Stokes problem, respectively) it is convenient, from a practical point of view to reduce the least squares problem (2.146)-(2.148) to the following problem

$$(2.171) \quad \begin{cases} \text{Find } \{\lambda_{1h},\lambda_{2h}\} \in M_{1h} \times M_{2h} \text{ such that} \\ \tilde{J}_h(\lambda_{1h},\lambda_{2h}) \le \tilde{J}_h(\mu_{1h},\mu_{2h}) \quad \forall \{\mu_{1h},\mu_{2h}\} \in M_{1h} \times M_{2h} \ , \end{cases}$$

where

$$(2.172) \quad \begin{cases} \tilde{J}_h(\mu_{1h},\mu_{2h}) = \frac{1}{2} \int_{\Omega_{12}} \{ |\nabla(y_{2h}(\mu_{2h}) - y_{1h}(\mu_{1h})|^2 + \\ \qquad + |y_{2h}(\mu_{2h}) - y_{1h}(\mu_{1h})|^2 \} \, dx \end{cases}$$

with $y_{ih}(\mu_{ih})$ solutions, for i=1,2, of the discrete Dirichlet problems on Ω_{ii}

(2.162) $\tilde{V}_{ih} = \{\phi_h | \phi_h \in H^1_{ih},\ \phi_{h|T}=0 \quad \forall T \in \mathcal{T}_{ih},\ \partial T \cap \partial\Omega_{ii} = \emptyset\}$.

The above $\tilde{V}_{ih}$ is clearly isomorphic to γH^1_{ih} and we have

$$\dim(\tilde{V}_{ih}) = \text{number of vertices of } \mathcal{T}_h \text{ located on } \partial\Omega_{ii} .$$

We consider now the subspaces M_{ih} and S_{ih} of $\tilde{V}_{ih}$ defined by

(2.163) $M_{ih} = \{\phi_h | \phi_h \in \tilde{V}_{ih},\ \phi_h=0 \text{ on } \partial\Omega_{ii} \cap \Gamma\}$,

(2.164) $S_{ih} \subset \tilde{V}_{ih},\ M_{ih} \oplus S_{ih} = \tilde{V}_{ih}$,

respectively. The above M_{ih} is isomorphic to V_{oih} and

(2.165) $$\begin{cases} N_{ih} = \dim(V_{oih}) = \dim(M_{ih}) = \\ = \text{number of vertices of } \mathcal{T}_h \text{ located on } \gamma_i \text{ but not on } \Gamma. \end{cases}$$

We define then M_h by

(2.166) $M_h = M_{1h} \times M_{2h}$;

M_h is isomorphic to $V_{oh} = V_{01h} \times V_{02h}$.

We have shown on Figure 2.5 an example of the above situation

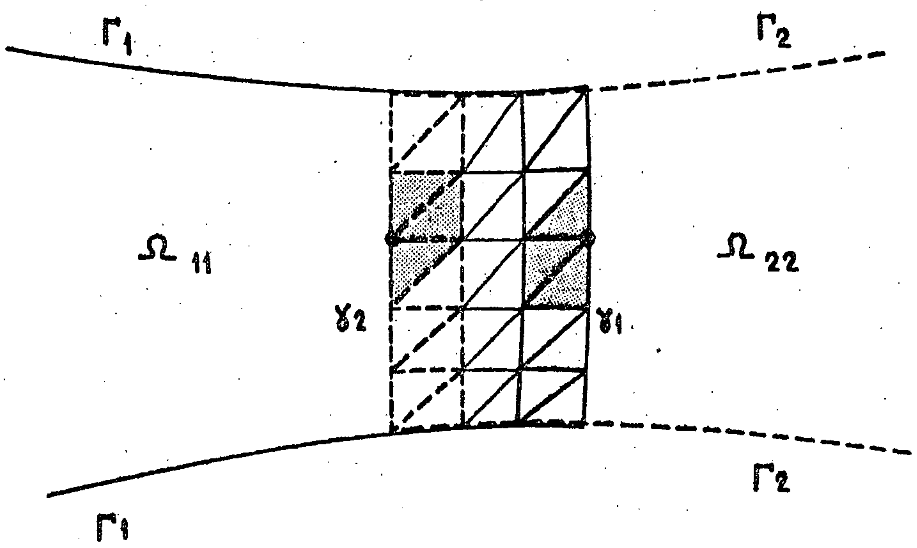


Figure 2.5

Nodes of $\mathcal{T}_h$ on γ_1, γ_2 corresponding to M_{1h}, M_{2h}.

C) – Some remarks concerning algorithm (2.150)–(2.158).

The various comments of Sec. 2.3.2.C still hold for algorithm (2.150)–(2.158). We observe moreover that obtaining g_h^{n+1} from $J_h'(u_h^{n+1})$, via (2.155), requires the solution of a linear system whose matrix is nondiagonal (it is sparse however); we should use the trapezoidal rule to approximate the bilinear form

$$\{z_h, z_h'\} \to \sum_{i=1}^{2} \int_{\gamma_i} z_{ih} z_{ih}' \, d\gamma$$

by another one, whose matrix is diagonal ; the same observation holds for the norm used in (2.156).

D) – Calculation of $J_h'(\cdot)$.

Calculations comparable to those done in Sec. 2.3.2.D yield to a discrete analogue of (2.84), namely

$$(2.159) \quad \begin{cases} \forall w_h \in V_{01h} \times V_{02h} \ \underline{\text{we have}} \\ (J_h'(v_h), w_h) = \int_{\Omega_{12}} \nabla(y_{2h}-y_{1h}) \cdot \nabla \tilde{w}_{2h} \, dx + \int_{\Omega_{12}} (y_{2h}-y_{1h}) \tilde{w}_{2h} dx + \\ + \int_{\Omega_{12}} \nabla(y_{1h}-y_{2h}) \cdot \nabla \tilde{w}_{1h} \, dx + \int_{\Omega_{12}} (y_{1h}-y_{2h}) \tilde{w}_{1h} \, dx \\ - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla p_{ih} \cdot \nabla \tilde{w}_{ih} \, dx, \end{cases}$$

where $\tilde{w}_h = \{\tilde{w}_{1h}, \tilde{w}_{2h}\}$ is any extension of w_h such that

$$(2.160) \quad \tilde{w}_{ih} \in H_{ih}^1, \ \tilde{w}_{ih}|_{\gamma_i} = w_{ih}, \ \tilde{w}_{ih}|_{\partial\Omega_{ii} \cap \Gamma} = 0 \ ;$$

in (2.159), $y_h = \{y_{1h}, y_{2h}\}$ and $p_h = \{p_{1h}, p_{2h}\}$ are obtained from v_h and $\{v_h, y_h\}$, respectively, by (2.148) and (for i=1,2) by

$$(2.161) \quad \begin{cases} p_{ih} \in H_{oih}^1 \ \underline{\text{and}} \\ \int_{\Omega_{ii}} \nabla p_{ih} \cdot \nabla z_{ih} \, dx = \int_{\Omega_{12}} \nabla(y_{ih}-y_{jh}) \cdot \nabla z_{ih} dx + \int_{\Omega_{12}} (y_{ih}-y_{jh}) z_{ih} dx \\ \forall z_{ih} \in H_{oih}^1 \end{cases}$$

(where $j=i+(-1)^{i-1}$), respectively.

From a practical (and industrial) point of view we should simplify the calculations involved in (2.159) if we can have supp $(\tilde{w}_h)$ as small as possible (we recall that supp (ϕ) is the closure of the set $\{x | \phi(x) \neq 0\}$); for that purpose we introduce the subspace $\tilde{V}_{ih}$ of H_{ih}^1, defined by

B) - Description of the conjugate gradient algorithm : For i=1,2 we define V_{oih} by

(2.149) $$\begin{cases} V_{oih} = \{v_{ih} | v_{ih} \in L^2(\gamma_i),\ v_{ih} = \tilde{v}_{ih}|_{\gamma_i} \text{ where } \tilde{v}_{ih} \in H^1_{ih}, \\ \tilde{v}_{ih} = 0 \text{ on } \partial\Omega_{ii} \cap \Gamma\} \ ; \end{cases}$$

the conjugate gradient algorithm is then defined as follows :

Step 0 : Initialization .

(2.150) $u^o_h = \{u^o_{1h}, u^o_{2h}\} \in V_{1h} \times V_{2h}$ is arbitrarily given,

define now

(2.151) $$\begin{cases} g^o_h = \{g^o_{1h}, g^o_{2h}\} \in V_{01h} \times V_{02h} \text{ such that} \\ \sum_{i=1}^{2} \int_{\gamma_i} g^o_{ih} z_{ih}\, d\gamma = (J'_h(u^o_h), z_h)\ \forall z_h = \{z_{ih}\}_{i=1}^2 \in V_{01h} \times V_{02h}, \end{cases}$$

where J'_h denotes the differential of J ; set then

(2.152) $w^o_h = g^o_h$.

Assuming that $u^n_h = \{u^n_{ih}\}_{i=1}^2$, $g^n_h = \{g^n_{ih}\}_{i=1}^2$, $w^n_h = \{w_{ih}\}_{i=1}^2$ are known we obtain u^{n+1}_h, g^{n+1}_h, w^{n+1}_h by

Step 1 : Steepest descent.

(2.153) $\rho_n = \operatorname{Arg\,Min}_{\rho \in \mathbb{R}} J_h(u^n_h - \rho w^n_h)$,

(2.154) $u^{n+1}_h = u^n_h - \rho_n w^n_h$.

Step 2 : Calculation of the new descent direction.

Compute

(2.155) $$\begin{cases} g^{n+1}_h = \{g^{n+1}_{1h}, g^{n+1}_{2h}\} \in V_{01h} \times V_{02h} \text{ such that} \\ \sum_{i=1}^{2} \int_{\gamma_i} g^{n+1}_{ih} z_{ih}\, d\gamma = (J'_h(u^{n+1}_h), z_h) \quad \forall z_h = \{z_{ih}\}_{i=1}^2 \in V_{01h} \times V_{02h}, \end{cases}$$

then

(2.156) $$\lambda_n = \frac{\sum_{i=1}^{2} \int_{\gamma_i} |g^{n+1}_{ih}|^2 d\gamma}{\sum_{i=1}^{2} \int_{\gamma_i} |g^n_i|^2\, d\gamma} ,$$

(2.157) $w^{n+1}_h = g^{n+1}_h + \lambda_n w^n_h$,

(2.158) do n=n+1, go to (2.153).

$$H^1_{oih} = \{\phi_h | \phi_h \in H^1_{ih} \ , \ \phi_h = 0 \ \underline{\text{on}} \ \partial\Omega_{ii}\} \ ,$$

$$V_{ih} = \{v_{ih} | v_{ih} \in L^2(\gamma_i), \ v_{ih} = \tilde{v}_{ih}|_{\gamma_i} \ \underline{\text{where}} \ \tilde{v}_{ih} \in H^1_{ih},$$
$$\tilde{v}_{ih} = g_{ih} \ \underline{\text{on}} \ \partial\Omega_{ii} \cap \Gamma\} \ ,$$

respectively. As in Sec.2.1 we still define γ_i by

$$\gamma_i = \partial\Omega_{ii} - (\partial\Omega_{ii} \cap \Gamma).$$

We have enough tools now to approximate the least squares problem (2.61)-(2.63) of Sec.2.3.1 (equivalent to (E) ; we suppose for simplicity that as in the continuous case we only have two sub-domains Ω_{11} and Ω_{22}) by

$$(2.146) \quad \begin{cases} \underline{\text{Find}} \ \{u_{1h}, u_{2h}\} \in V_{1h} \times V_{2h} \ \underline{\text{such that}} \\ J_h(u_{1h}, u_{2h}) \le J_h(v_{1h}, v_{2h}) \quad \forall \ \{v_{1h}, v_{2h}\} \in V_{1h} \times V_{2h} \end{cases}$$

where

$$(2.147) \quad \begin{cases} J_h(v_{1h}, v_{2h}) = \frac{1}{2} \int_{\Omega_{12}} \{|\nabla(y_{2h}(v_{2h}) - y_{1h}(v_{1h}))|^2 + \\ \qquad\qquad + |y_{2h}(v_{2h}) - y_{1h}(v_{1h})|^2\} \, dx \end{cases}$$

with $y_{ih}(v_{ih})$ solutions, for i=1,2, of the discrete Dirichlet problems on Ω_{ii}

$$(2.148) \quad \begin{cases} \int_{\Omega_{ii}} \nabla y_{ih} \cdot \nabla \phi_h \, dx = \int_{\Omega_{ii}} f_{ih} \phi_h \, dx \quad \forall \phi_h \in H^1_{oih}, \\ y_{ih} \in H^1_{ih}, \ y_{ih} = g_{ih} \ \underline{\text{on}} \ \partial\Omega_{ii} \cap \Gamma \ , \ y_{ih} = v_{ih} \ \underline{\text{on}} \ \gamma_i \ , \end{cases}$$

where $f_{ih} = f_h|_{\Omega_{ii}}$.

We can easily prove that the following

<u>Proposition 2.6</u> : <u>The least squares problem</u> (2.146)-(2.148) <u>has a unique solution such that</u>

$$u_{ih} = y_h|_{\gamma_i} \quad \forall i=1,2,$$

<u>where</u> y_h <u>is the solution of</u> (2.143).

2.3.6. <u>Conjugate gradient solution of the least squares problem (2.146)-(2.148). A first algorithm.</u>

A) - <u>Synopsis</u> : In this subsection we consider the solution of the discrete least squares problem (2.146)-(2.148) by a conjugate gradient algorithm derived from the conjugate gradient algorithm (2.65)-(2.73) of Sec. 2.3.2.B ; as in Sec. 2.3.2.B we suppose that the V_{ih}'s are equipped with $L^2(\gamma_i)$-norms.

Step 3 : Once u is known (from Step 2) we finally obtain the solution $y = \{y_1,y_2\}$ of problem (2.61) (and therefore the solution of problem (E) (cf. (2.1))) by solving (2.63) with $y_i=u_i$ on γ_i.

We observe from Steps 1-3 that solving (2.61) is equivalent to the solution of 3 Poisson problems on each Ω_{ii}, plus the solution of the variational problem (2.88) ; in the following sections we shall discuss the finite element implementation of these methods founded on the above decomposition properties.

2.3.5. Finite element approximation of (E) and of the least square problem (2.61).

We suppose for simplicity that Ω is a bounded polygonal domain of $\mathbb{R}^2$; with $\mathcal{C}_h$ a triangulation of Ω -such that $\bigcup_{T\in\mathcal{C}_h} T=\bar{\Omega}$ - we define

(2.141) $\quad H_h^1 = \{\phi_h | \phi_h \in C^0(\bar{\Omega}),\ \phi_h|_T \in P_1 \quad \forall\, T\in\mathcal{C}_h\}$,

where P_1 is the space of the polynomials in two variables of degree ≤ 1, and then

(2.142) $\quad H_{oh}^1 = H_h^1 \cap H_o^1(\Omega) = \{\phi_h|\phi_h \in H_h^1,\ \phi_h=0 \text{ on } \Gamma\}$.

We denote by γH_h^1 the vector space generated by the traces on Γ of H_h^1 ; assuming for simplicity that g is continuous over Γ we define g_h as the unique element of γH_h^1 such that $g_h(P) = g(P)$ $\forall P$ vertex of $\mathcal{C}_h$ belonging to Γ. We approximate, then, the Poisson problem (E) (see (2.1)) by

(2.143) $$\begin{cases} \text{Find } y_h\in H_h^1,\ y_h|_\Gamma = g_h \text{ such that} \\ \int_\Omega \nabla y_h\cdot\nabla\phi_h\,dx = \int_\Omega f_h\phi_h\,dx \quad \forall\phi_h\in H_{oh}^1, \end{cases}$$

where f_h is a convenient approximation of f.

Problem (2.143) has a unique solution.

We consider now a decomposition of Ω by subdomains Ω_{ii},Ω_{ij} $(i\neq j)$ satisfying (2.5),(2.6) (cf. Sec. 2.1), and also the following properties

(2.144) $\quad \forall i,\ \bar{\Omega}_{ii} = \bigcup_{T\in\mathcal{C}_{ih}} T,$

(2.145) $\quad \forall i,j,\ i\neq j\ \ \bar{\Omega}_{ij} = \bigcup_{T\in\mathcal{C}_{ijh}} T,$

where $\mathcal{C}_{ih},\mathcal{C}_{ijh}$ are subsets of $\mathcal{C}_h$ (thus the Ω_{ii},Ω_{ij} are also polygonal domains) ; to each Ω_{ii} we associate H_{ih}^1, H_{oih}^1, V_{ih} defined by

$$H_{ih}^1 = \{\phi_h|\phi_h \in C^0(\bar{\Omega}_{ii}),\ \phi_h|_T\in P_1 \quad \forall T\in\mathcal{C}_{ih}\}\ ,$$

Step 1 : Calculation of the right-hand side of (2.88).

Let $\bar{u}$ be an arbitrary element of $V_1 \times V_2$; we have from (2.60) that $\bar{u}_i = g_i$ on $\partial\Omega_{ii} \cap \Gamma$.

From (2.84) we obtain that

$$
(2.138) \quad \begin{cases} (J'(\bar{u}),w) = \int_{\Omega_{12}} \nabla(\bar{y}_2-\bar{y}_1)\cdot\nabla\tilde{w}_2 \, dx + \int_{\Omega_{12}} (\bar{y}_2-\bar{y}_1)\tilde{w}_2 \, dx + \\ + \int_{\Omega_{12}} \nabla(\bar{y}_1-\bar{y}_2)\cdot\nabla\tilde{w}_1 \, dx + \int_{\Omega_{12}} (\bar{y}_1-\bar{y}_2)\tilde{w}_2 \, dx - \\ - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla\bar{p}_i\cdot\nabla\tilde{w}_i \, dx, \end{cases}
$$

where, in (2.138), $\tilde{w} = \{\tilde{w}_1,\tilde{w}_2\}$ is an arbitrary extension of w such that

$$\tilde{w}_i \in H^1(\Omega_{ii}), \ \tilde{w}_i|_{\gamma_i} = w_i, \ \tilde{w}_i|_{\partial\Omega_{ii}\cap\Gamma} = 0 ;$$

in (2.138), $\bar{y} = \{\bar{y}_1,\bar{y}_2\}$ and $\bar{p} = \{\bar{p}_1,\bar{p}_2\}$ are obtained from $\bar{u}$ as follows (from (2.63) and (2.80), respectively) :

$$
(2.139) \quad \begin{cases} -\Delta\bar{y}_i = f_i \ \underline{\text{in}} \ \Omega_{ii}, \\ \bar{y}_i = g_i \ \underline{\text{on}} \ \partial\Omega_{ii}\cap\Gamma, \\ \bar{y}_i = \bar{u}_i \ \underline{\text{on}} \ \gamma_i \end{cases}
$$

and

$$
(2.140) \quad \begin{cases} \bar{p}_i \in H^1_o(\Omega_{ii}) \ \underline{\text{and}} \\ \int_{\Omega_{ii}} \nabla\bar{p}_i\cdot\nabla z_i \, dx = \int_{\Omega_{12}} \nabla(\bar{y}_i-\bar{y}_j)\cdot\nabla z_i \, dx + \int_{\Omega_{12}} (\bar{y}_i-\bar{y}_j) z_i \, dx \\ \forall z_i \in H^1_o(\Omega_{ii}) , \end{cases}
$$

where $j=i+(-1)^{i-1}$.

Step 2 : Solution of the variational problem (2.88).

We obtain $u-\bar{u}$ via the solution of the linear variational problem (2.88), and then u, since $u = \bar{u}+(u-\bar{u})$.

The bilinear form $a(\cdot,\cdot)$ occuring in (2.88) is not known explicitely, but using a convenient finite element approximation of (2.61) we shall be able to obtain a finite dimensional approximation of it, modulo the solution of a finite number (a priori known) of discrete Dirichlet problems on each Ω_{ii} ; this construction will be discussed in detail in Sec. 2.3.8.

then

(2.132) $$\lambda_n = \frac{\sum_{i=1}^{2} \int_{\Omega_{ii}} |\nabla \tilde{g}_i^{n+1}|^2 dx}{\sum_{i=1}^{2} \int_{\Omega_{ii}} |\nabla \tilde{g}_i^{n}|^2 dx} ,$$

(2.133) $$\tilde{w}^{n+1} = \tilde{g}^{n+1} + \lambda_n \tilde{w}^n ,$$

(2.134) do $n=n+1$, go to (2.129).

Steps (2.129),(2.131) involve a bilinear form $\tilde{a} : (\tilde{V}_{01} \times \tilde{V}_{02})^2 \to \mathbb{R}$ whose definition is the following :

Let $\tilde{v} = \{\tilde{v}_1, \tilde{v}_2\} \in \tilde{V}_{01} \times \tilde{V}_{02}$; compute $\eta = \{\eta_1, \eta_2\} \in H^1(\Omega_{11}) \times H^1(\Omega_{22})$ as the solution, for $i=1,2$, of

(2.135) $$\begin{cases} \Delta \eta_i = 0 \text{ in } \Omega_{ii}, \\ \eta_i = \tilde{v}_i \text{ on } \partial\Omega_{ii} ; \end{cases}$$

compute then $\pi = \{\pi_1, \pi_2\} \in H_o^1(\Omega_{11}) \times H_o^1(\Omega_{22})$ via the solution, for $i=1,2$, of

(2.136) $$\begin{cases} \pi_i \in H_o^1(\Omega_{ii}), \\ \int_{\Omega_{ii}} \nabla \pi_i \cdot \nabla z_i dx = \int_{\Omega_{12}} \nabla(\eta_i - \eta_j) \cdot \nabla z_i dx + \int_{\Omega_{12}} (\eta_i - \eta_j) z_i dx \\ \forall z_i \in H_o^1(\Omega_{ii}) \end{cases}$$

where $j = i + (-1)^{i-1}$; we have then

(2.137) $$\begin{cases} \tilde{a}(\tilde{v}, \tilde{w}) = \int_{\Omega_{12}} \nabla(\eta_2 - \eta_1) \cdot \nabla(\tilde{w}_2 - \tilde{w}_1) dx + \int_{\Omega_{12}} (\eta_2 - \eta_1)(\tilde{w}_2 - \tilde{w}_1) dx \\ - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla \pi_i \cdot \nabla \tilde{w}_i dx \quad \forall \, \tilde{w} = \{\tilde{w}_1, \tilde{w}_2\} \in \tilde{V}_{01} \times \tilde{V}_{02} . \end{cases}$$

We observe, from (2.124)-(2.137), that each iteration of (2.124)-(2.134) requires the solution of two Dirichlet problems on each Ω_{ii}, similar to (2.135),(2.136), respectively, plus the solution of the variational problem (2.131).

2.3.4. Decomposition properties of the least squares problem (2.61).

In view of the quasi-direct solution (discussed in Sec. 2.3.8) of the finite dimensional problems approximating (2.61), it is of fundamental practical importance to analyse the solution process of problem (2.61), via the variational problem (2.88) ; in that direction three steps appear quite naturally

(2.125) $$\begin{cases} -\Delta y_i^o = f_i \text{ in } \Omega_{ii}, \\ y_i^o - \tilde{u}_i^o \in H_o^1(\Omega_{ii}) \ (\Longrightarrow y_i^o = \tilde{u}_i^o \text{ on } \partial\Omega_{ii}) \end{cases}$$

which has a unique solution in $H^1(\Omega_{ii})$, and then

(2.126) $$\begin{cases} p_i^o \in H_o^1(\Omega_{ii}) \ , \\ \int_{\Omega_{ii}} \nabla p_i^o \cdot \nabla z_i dx = \int_{\Omega_{12}} \nabla(y_i^o - y_j^o) \cdot \nabla z_i dx + \int_{\Omega_{12}} (y_i^o - y_j^o) z_i dx \\ \forall z_i \in H_o^1(\Omega_{ii}), \end{cases}$$

where $j = i + (-1)^{i-1}$.

Define now $\tilde{g}^o = \{\tilde{g}_1^o, \tilde{g}_2^o\}$ as the solution of

(2.127) $$\begin{cases} \tilde{g}^o \in \tilde{V}_{01} \times \tilde{V}_{02} \text{ and } \forall\ \tilde{z} = \{\tilde{z}_1, \tilde{z}_2\} \in \tilde{V}_{01} \times \tilde{V}_{02} \text{ we have} \\ \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla \tilde{g}_i^o \cdot \nabla \tilde{z}_i dx = \int_{\Omega_{12}} \nabla(y_2^o - y_1^o) \cdot \nabla(\tilde{z}_2 - \tilde{z}_1) dx + \\ + \int_{\Omega_{12}} (y_2^o - y_1^o)(\tilde{z}_2 - \tilde{z}_1) dx - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla p_i^o \cdot \nabla \tilde{z}_i dx \end{cases}$$

and set

(2.128) $$\tilde{w}^o = \tilde{g}^o.$$

Assuming that $\tilde{u}^n, \tilde{g}^n, \tilde{w}^n$ are known we obtain $\tilde{u}^{n+1}, \tilde{g}^{n+1}, \tilde{w}^{n+1}$ by

Step 1 : Steepest descent

(2.129) $$\rho_n = \frac{\sum_{i=1}^{2} \int_{\Omega_{ii}} |\nabla \tilde{g}_i^n|^2 dx}{\tilde{a}(\tilde{w}^n, \tilde{w}^n)} ,$$

(2.130) $$\tilde{u}^{n+1} = \tilde{u}^n - \rho_n \tilde{w}^n.$$

Step 2 : Calculation of the new descent direction

Define $\tilde{g}^{n+1} = \{\tilde{g}_1^{n+1}, \tilde{g}_2^{n+1}\}$ as the solution of

(2.131) $$\begin{cases} \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla \tilde{g}_i^{n+1} \cdot \nabla \tilde{z}_i dx = \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla \tilde{g}_i^n \cdot \nabla \tilde{z}_i \ dx - \rho_n \tilde{a}(\tilde{w}^n, \tilde{z}_i) \\ \forall \tilde{z} = \{\tilde{z}_1, \tilde{z}_2\} \in \tilde{V}_{01} \times \tilde{V}_{02} \ , \\ \tilde{g}^{n+1} \in \tilde{V}_{01} \times \tilde{V}_{02}, \end{cases}$$

$$(2.116)\quad \begin{cases} \tilde{g}^o = \{\tilde{g}_1^o, \tilde{g}_2^o\} \in \tilde{V}_{01} \times \tilde{V}_{02} \ \underline{\text{such that}} \\ \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla \tilde{g}_i^o \cdot \nabla \tilde{z}_i \, dx = (\tilde{J}'(\tilde{u}^o), \tilde{z}) \quad \forall\ \tilde{z} = \{\tilde{z}_i\}_{i=1}^2 \in \tilde{V}_{01} \times \tilde{V}_{02}, \end{cases}$$

where $\tilde{J}'$ denotes the differential of $\tilde{J}$, then set

$$(2.117)\quad \tilde{w}^o = \tilde{g}^o .$$

Assuming that $\tilde{u}^n = \{\tilde{u}_i^n\}_{i=1}^2$, $\tilde{g}^n = \{\tilde{g}_i^n\}_{i=1}^2$, $\tilde{w}^n = \{\tilde{w}_i^n\}_{i=1}^2$ are known, we obtain $\tilde{u}^{n+1}, \tilde{g}^{n+1}, \tilde{w}^{n+1}$ by

Step 1 : Steepest descent

$$(2.118)\quad \rho_n = \operatorname*{Arg\,Min}_{\rho \in \mathbb{R}} \tilde{J}(\tilde{u}^n - \rho \tilde{w}^n),$$

$$(2.119)\quad \tilde{u}^{n+1} = \tilde{u}^n - \rho_n \tilde{w}^n .$$

Step 2 : Calculation of the new descent direction

Compute

$$(2.120)\quad \begin{cases} \tilde{g}^{n+1} = \{\tilde{g}_1^{n+1}, \tilde{g}_2^{n+1}\} \in \tilde{V}_{01} \times \tilde{V}_{02} \ \underline{\text{such that}} \\ \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla \tilde{g}_i^{n+1} \cdot \nabla \tilde{z}_i \, dx = (\tilde{J}'(\tilde{u}^{n+1}), \tilde{z}) \\ \forall\ \tilde{z} = \{\tilde{z}_i\}_{i=1}^2 \in \tilde{V}_{01} \times \tilde{V}_{02}, \end{cases}$$

then

$$(2.121)\quad \lambda_n = \frac{\sum_{i=1}^{2} \int_{\Omega_{ii}} |\nabla \tilde{g}_i^{n+1}|^2 dx}{\sum_{i=1}^{2} \int_{\Omega_{ii}} |\nabla \tilde{g}_i^{n}|^2 \, dx},$$

$$(2.122)\quad \tilde{w}^{n+1} = \tilde{g}^{n+1} + \lambda_n \tilde{w}^n ,$$

(2.123) do n=n+1, go to (2.118).

D. On the practical implementation of algorithm (2.115)-(2.123).

Most considerations of Sec. 2.3.2.C,D,E concerning algorithm (2.65)-(2.73) hold also in fact for algorithm (2.115)-(2.123) ; it follows then from these considerations that a more explicit description of the basic algorithm (2.115)-(2.123) is given by the following variant of (2.95)-(2.108) :

Step 0 : Initialization

$$(2.124)\quad \tilde{u}^o = \{\tilde{u}_1^o, \tilde{u}_2^o\} \ \underline{\text{is arbitrarily given in}}\ \tilde{V}_1 \times \tilde{V}_2 .$$

Solve then, for $i=1,2$,

$$(2.109)\quad \begin{cases} \forall\ v \in V_1 \times V_2 \ \underline{\text{and if}}\ \tilde{v} = Tv,\ \underline{\text{then}} \\ \tilde{v} = \{\tilde{v}_1, \tilde{v}_2\} \in H^1(\Omega_{11}) \times H^1(\Omega_{22}),\ \tilde{v}_i|_{\gamma_i} = v_i, \\ \tilde{v}_i|_{\partial\Omega_{ii} \cap \Gamma} = g_i \quad \forall\ i=1,2\ ; \end{cases}$$

there is an infinity of such lifting operators T ; we associate to T the spaces $\tilde{V}_i$, $\tilde{V}_i \subset H^1(\Omega_{ii})$, by

$$(2.110)\quad \tilde{V}_1 \times \tilde{V}_2 = T(V_1 \times V_2).$$

Using the notation in (2.109),(2.110) let us consider the control problem

$$(2.111)\quad \begin{cases} \underline{\text{Find}}\ \tilde{u} \in \tilde{V}_1 \times \tilde{V}_2\ \underline{\text{such that}} \\ \tilde{J}(\tilde{u}) \le \tilde{J}(\tilde{v}) \quad \forall \tilde{v} \in \tilde{V}_1 \times \tilde{V}_2 \end{cases}$$

where

$$(2.112)\quad \tilde{J}(\tilde{v}) = \frac{1}{2} \int_{\Omega_{12}} \{ |\nabla(y_2 - y_2)|^2 + |y_2 - y_1|^2 \} dx$$

with, $\forall$ i=1,2, $y_i = y_i(\tilde{v})$ the solution of

$$(2.113)\quad \begin{cases} -\Delta y_i = f_i\ \underline{\text{in}}\ \Omega_{ii}, \\ y_i - \tilde{v}_i \in H_0^1(\Omega_{ii}) . \end{cases}$$

The solution u of (2.61) and $\tilde{u}$ of (2.111) are clearly related by

$$(2.114)\quad \tilde{u} = Tu.$$

C. <u>A conjugate gradient algorithm for solving</u> (2.111).

Let us define, for i=1,2, $\tilde{V}_{0i}$ by

$$\tilde{V}_{0i} = \{ \tilde{v}_i | \tilde{v}_i \in H^1(\Omega_{ii}),\ \tilde{v}_i + \tilde{z}_i \in \tilde{V}_i \quad \forall \tilde{z}_i \in \tilde{V}_i \} ;$$

we have then the following variant of algorithm (2.65)-(2.73) of Sec. 2.3.2.B.

<u>Step 0 : Initialization</u>

$$(2.115)\quad \tilde{u}^o = \{\tilde{u}_1^o, \tilde{u}_2^o\} \in \tilde{V}_1 \times \tilde{V}_2 \quad \underline{\text{is arbitrarily given}},$$

<u>define now</u>

compute then $\pi^n = \{\pi_1^n, \pi_2^n\} \in H_0^1(\Omega_{11}) \times H_0^1(\Omega_{22})$ *via the solution, for* i=1,2, *of*

$$(2.107)\quad \begin{cases} \pi_i^n \in H_0^1(\Omega_{ii}) , \\ \int_{\Omega_{ii}} \nabla\pi_i^n \cdot \nabla z_i dx = \int_{\Omega_{12}} \nabla(\eta_i^n - \eta_j^n)\cdot\nabla z_i dx + \int_{\Omega_{12}} (\eta_i^n - \eta_j^n) z_i dx \\ \forall z_i \in H_0^1(\Omega_{ii}) \end{cases}$$

where $j=i+(-1)^{i-1}$; *we finally have* (*from* (2.92))

$$(2.108)\quad \begin{cases} a(w^n,w) = \int_{\Omega_{12}} \nabla(\eta_2^n-\eta_1^n)\cdot\nabla\tilde{w}_2 dx + \int_{\Omega_{12}} (\eta_2^n-\eta_1^n)\tilde{w}_2 dx + \\ \qquad + \int_{\Omega_{12}} \nabla(\eta_1^n-\eta_2^n)\cdot\nabla\tilde{w}_1 dx + \int_{\Omega_{12}} (\eta_1^n-\eta_2^n)\tilde{w}_1 dx - \\ \qquad - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla\pi_i^n\cdot\nabla\tilde{w}_i dx \quad \forall\, w \in V_{01}\times V_{02} , \end{cases}$$

with $\tilde{w}$ *as in* (2.92).

We observe that each iteration requires the solution of the 4 Dirichlet problems (2.106),(2.107).

2.3.3. *Conjugate gradient solution of* (2.61). *A second algorithm.*

A. *Synopsis.*

The numerical solution of problem (E) via the discrete variants of algorithm (2.65)-(2.73) has produced quite good results ; we can observe however that the norm we have chosen, i.e.

$$w \to \left(\sum_i \int_{\gamma_i} |w_i|^2 d\gamma\right)^{1/2}$$

is not optimally suited to the ellipticity properties, of the bilinear form $a(\cdot,\cdot)$, that we have mentioned in Sec. 2.3.2.D ; it is therefore reasonable to suppose that using a norm induced by the $H^1(\Omega_{ii})$'s may improve the convergence of our conjugate gradient algorithms. The practical implementation of such norm is discussed below.

B. *An equivalent formulation of* (2.61).

The spaces V_i being defined by (2.60) we introduce an extension operator[1] $T : V_1 \times V_2 \to H^1(\Omega_{11}) \times H^1(\Omega_{22})$, such that

---

[1] It is precisely a *lifting* operator

where $\tilde{w} = \{\tilde{w}_1, \tilde{w}_2\}$ is an arbitrary extension of w such that $\tilde{w}_i \in H^1(\Omega_{ii})$, $\tilde{w}_i|_{\gamma_i} = w_i$, $\tilde{w}_i|_{\partial\Omega_{ii}\cap\Gamma} = 0$, and set

(2.99) $\quad w^o = g^o.$

Assuming that $u^n = \{u_i^n\}_{i=1}^2$, $g^n = \{g_i^n\}_{i=1}^2$, $w^n = \{w_i^n\}_{i=1}^2$ are known we obtain $u^{n+1}, g^{n+1}, w^{n+1}$ by

Step 1 : Steepest descent

(2.100) $$\rho_n = \frac{\sum_{i=1}^{2}\int_{\gamma_i} |g_i^n|^2 \, d\gamma}{a(w^n,w^n)},$$

(2.101) $$u^{n+1} = u^n - \rho_n w^n.$$

Step 2 : Calculation of the new descent direction

Define $g^{n+1} = \{g_1^{n+1}, g_2^{n+1}\}$ by solving

(2.102) $$\begin{cases} \sum_{i=1}^{2}\int_{\gamma_i} g_i^{n+1} w_i \, d\gamma = \sum_{i=1}^{2}\int_{\gamma_i} g_i^n w_i \, d\gamma - \rho_n \, a(w^n,w) \\ \forall \; w = \{w_1,w_2\} \in V_{01}\times V_{02} , \end{cases}$$

then

(2.103) $$\lambda_n = \frac{\sum_{i=1}^{2}\int_{\gamma_i} |g_i^{n+1}|^2 \, d\gamma}{\sum_{i=1}^{2}\int_{\gamma_i} |g^n|^2 \, d\gamma},$$

(2.104) $$w^{n+1} = g^{n+1} + \lambda_n w^n ,$$

(2.105) do $n=n+1$, go to (2.100).

We detail the calculation of $a(w^n,w), \forall w \in V_{01}\times V_{02}$:

Compute $\eta^n = \{\eta_1^n, \eta_2^n\} \in H^1(\Omega_{11})\times H^1(\Omega_{22})$ via the solution, for $i=1,2$, of

(2.106) $$\begin{cases} \Delta\eta_i^n = 0 \text{ in } \Omega_{ii} , \\ \eta_i^n = 0 \text{ on } \partial\Omega_{ii} \cap \Gamma, \\ \eta_i^n = w_i^n \text{ on } \gamma_i ; \end{cases}$$

We also have, from (2.69),(2.70),(2.87), that $g^{n+1} = \{g_1^{n+1}, g_2^{n+1}\}$ can be defined as the solution of the following linear variational problem in $V_{01}\times V_{02}$

$$(2.94)\quad \begin{cases} \sum_{i=1}^{2} \int_{\gamma_i} g_i^{n+1} w_i \, d\gamma = \sum_{i=1}^{2} \int_{\gamma_i} g_i^n w_i \, d\gamma - \rho_n \, a(w^n, w) \\ \forall \; w = \{w_1, w_2\} \in V_{01}\times V_{02}. \end{cases}$$

It appears clearly from (2.93),(2.94) that a crucial point when using algorithm (2.65)-(2.73) to solve (2.61) (and therefore (E)) is the calculation of $a(w^n,w)$ for w given in $V_{01}\times V_{02}$; this can be done using (2.89) or (2.92).

Collecting the above results the final form of algorithm (2.65)-(2.73) is then (we apologize for the redundancy) :

Step 0 : Initialization

$$(2.95)\quad u^o = \{u_1^o, u_2^o\} \text{ is arbitrarily given in } V_1\times V_2.$$

Solve then, for $i=1,2$,

$$(2.96)\quad \begin{cases} -\Delta y_i^o = f_i \text{ in } \Omega_{ii}, \\ y_i^o = g_i \text{ on } \partial\Omega_{ii}\cap\Gamma, \\ y_i^o = u_i^o \text{ on } \gamma_i \end{cases}$$

which has a unique solution in $H^1(\Omega_{ii})$, and then

$$(2.97)\quad \begin{cases} p_i^o \in H_o^1(\Omega_{ii}), \\ \int_{\Omega_{ii}} \nabla p_i^o \cdot \nabla z_i \, dx = \int_{\Omega_{12}} \nabla(y_i^o - y_j^o)\cdot\nabla z_i \, dx + \int_{\Omega_{12}} (y_i^o - y_j^o) z_i \, dx \\ \forall \; z_i \in H_o^1(\Omega_{ii}) \end{cases}$$

where $j=i+(-1)^{i-1}$.

Define now $g^o = \{g_1^o, g_2^o\}$ as the solution of

$$(2.98)\quad \begin{cases} \sum_{i=1}^{2} \int_{\gamma_i} g_i^o w_i \, d\gamma = \int_{\Omega_{12}} \nabla(y_2^o - y_1^o)\cdot\nabla\tilde{w}_2 \, dx + \int_{\Omega_{12}} (y_2^o - y_1^o)\tilde{w}_2 \, dx + \\ + \int_{\Omega_{12}} \nabla(y_1^o - y_2^o)\cdot\nabla\tilde{w}_1 \, dx + \int_{\Omega_{12}} (y_1^o - y_2^o)\tilde{w}_1 \, dx - \\ - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla p_i^o \cdot \nabla\tilde{w}_i \, dx \quad \forall \; w = \{w_1, w_2\} \in V_{01}\times V_{02}, \end{cases}$$

$$(2.91)\quad \begin{cases} \Delta y_i = 0 \ \underline{\text{in}} \ \Omega_{ii}, \\ y_i = 0 \ \underline{\text{on}} \ \partial\Omega_{ii} \cap \Gamma, \\ y_i = w_i \ \underline{\text{on}} \ \gamma_i, \end{cases}$$

respectively.

We also have from (2.84),(2.89) that

$$(2.92)\quad \begin{cases} a(w',w) = a(w,w') = \int_{\Omega_{12}} \nabla(y_2-y_1)\cdot\nabla\tilde{w}_2' dx + \\ + \int_{\Omega_{12}} (y_2-y_1)\tilde{w}_2' \, dx + \int_{\Omega_{12}} \nabla(y_1-y_2)\cdot\nabla\tilde{w}_1' \, dx + \\ + \int_{\Omega_{12}} (y_1-y_2)\tilde{w}_1' \, dx - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla p_i\cdot\nabla\tilde{w}_i' \, dx = \\ = \int_{\Omega_{12}} \nabla(y_2'-y_1')\cdot\nabla\tilde{w}_2 \, dx + \int_{\Omega_{12}} (y_2'-y_1')\tilde{w}_2 \, dx + \\ + \int_{\Omega_{12}} \nabla(y_1'-y_2')\cdot\nabla\tilde{w}_1 \, dx + \int_{\Omega_{12}} (y_1'-y_2')\tilde{w}_1 \, dx - \\ - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla p_i'\cdot\nabla\tilde{w}_i \, dx \end{cases}$$

where p_i',p_i are the solutions of (2.80) corresponding to y_i',y_i given by (2.90),(2.91), and where $\tilde{w}_i$ (resp. $\tilde{w}_i'$) is any extension of w_i (resp. w_i') such that $\tilde{w}_i$ (resp. $\tilde{w}_i'$) $\in H^1(\hat{\Omega}_{ii})$, $\tilde{w}_i|_{\gamma_i} = w_i$, $\tilde{w}_i|_{\partial\Omega_{ii}\cap\Gamma} = 0$ (resp. $\tilde{w}_i'|_{\gamma_i} = w_i'$, $\tilde{w}_i'|_{\partial\Omega_{ii}\cap\Gamma} = 0$) ; if one takes $\tilde{w}_i = y_i$, $\tilde{w}_i' = y_i'$ in (2.92), it follows from (2.90),(2.91) that we recover (2.89). In practice it may be more convenient to use (2.92) than (2.89) despite the fact that (2.92) is, apparently more complicated than (2.89) ; the justification of such a choice will appear clearly in Sec. 2.3.5 where we shall discuss the <u>finite element</u> approximation of problems (E) and (2.61).

F. <u>Application to algorithm</u> (2.65)-(2.73).

It follows from (2.68),(2.85)[1] that we have

$$(2.93)\quad \rho_n = \frac{(J'(u^n),w^n)}{a(w^n,w^n)} = \frac{\sum_{i=1}^{2} \int_{\gamma_i} |g_i^n|^2 \, d\gamma}{a(w^n,w^n)} .$$

---

(1) And also from the basic properties of conjugate gradient methods in the case of linear problems (cf. DANIEL [21]for more details).

where the bilinear form $a(\cdot,\cdot)$ is continuous, symmetric, positive definite over $(V_{01}\times V_{02})\times(V_{01}\times V_{02})$, and independent of f and g (it is in fact strongly elliptic if these spaces are equipped with the norms induced by $H^1(\Omega_{11})$, $H^1(\Omega_{22})$, i.e. $\exists\alpha>0$ such that

$$(2.86)\qquad a(w,w)\geq\alpha\|w\|^2_{V_{01}\times V_{02}}$$

where $\|w\|_{V_{01}\times V_{02}}=\left(\sum_{i=1}^{2}\|w_i\|^2_{V_{0i}}\right)^{1/2}$, with $(\forall i=1,2)$,
$\|w_i\|_{V_{0i}}=\inf_{\tilde{w}_i}\|\tilde{w}_i\|_{H^1(\Omega_{ii})}$ with $\tilde{w}_i\in H^1(\Omega_{ii})$, $\tilde{w}_i=0$ on $\partial\Omega_{ii}\cap\Gamma$, $\tilde{w}_i=w_i$ on γ_i ; the proof of (2.86) is given in GLOWINSKI-LIONS-PERIAUX [20]).

It follows from (2.85) that we have

$$(2.87)\qquad (J'(v')-J'(v),w)=a(v'-v,w)\ \forall v,v'\in V_1\times V_2,\ \forall w\in V_{01}\times V_{02}.$$

It follows also from (2.87) that if u is the solution of (2.61) then u is solution of the linear variational problem over $V_{01}\times V_{02}$, below

$$(2.88)\qquad \begin{cases}\text{Find } u\in V_1\times V_2 \text{ such that}\\ a(u-\bar{u},w)=-(J'(\bar{u}),w)\quad\forall w\in V_{01}\times V_{02},\end{cases}$$

where $\bar{u}$ is an arbitrary element of $V_1\times V_2$ ($\Longrightarrow u-\bar{u}\in V_{01}\times V_{02}$).

E. Calculation of $a(\cdot,\cdot)$.

The bilinear form $a(\cdot,\cdot)$ is not known explicitely ; it is however quite easy to compute $a(w',w)$ for every pair $\{w',w\}\in(V_{01}\times V_{02})^2$, since

$$(2.89)\qquad a(w',w)=\int_{\Omega_{12}}\{\nabla(y_2'-y_1')\cdot\nabla(y_2-y_1)+(y_2'-y_1')(y_2-y_1)\}\,dx,$$

where, in (2.89), y_i',y_i are, for $i=1,2$, the solutions in $H^1(\Omega_{ii})$ of

$$(2.90)\qquad \begin{cases}\Delta y_i'=0 \text{ in } \Omega_{ii},\\ y_i'=0 \text{ on } \partial\Omega_{ii}\cap\Gamma,\\ y_i'=w_i' \text{ on } \gamma_i,\end{cases}$$

Taking $z_i = \delta(y_i - \tilde{v}_i)$ in (2.80) we obtain

$$(2.81)\quad \begin{cases} \int_{\Omega_{12}} \nabla(y_2-y_1)\cdot\nabla\delta(y_2-\tilde{v}_2)dx + \int_{\Omega_{12}} (y_2-y_1)\ \delta(y_2-\tilde{v}_2)dx + \\ + \int_{\Omega_{12}} \nabla(y_1-y_2)\cdot\nabla\delta(y_1-\tilde{v}_1)dx + \int_{\Omega_{12}} (y_1-y_2)\ \delta(y_1-\tilde{v}_1)dx = \\ = \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla p_i\cdot\nabla\delta(y_i-\tilde{v}_i)dx. \end{cases}$$

Since $p_i \in H_o^1(\Omega_{ii})$ we have from (2.75) that

$$(2.82)\quad \int_{\Omega_{ii}} \nabla\delta y_i\cdot\nabla p_i dx = 0\ ;$$

combining (2.79),(2.81),(2.82) we obtain

$$(2.83)\quad \begin{cases} \delta J(v) = (J'(v),\delta v) = \int_{\Omega_{12}} \nabla(y_2-y_1)\cdot\nabla\delta\tilde{v}_2 dx + \\ + \int_{\Omega_{12}} (y_2-y_1)\delta\tilde{v}_2 dx + \int_{\Omega_{12}} \nabla(y_1-y_2)\cdot\nabla\delta\tilde{v}_1 dx + \\ + \int_{\Omega_{12}} (y_1-y_2)\delta\tilde{v}_1 dx - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla p_i\cdot\nabla\delta\tilde{v}_i dx. \end{cases}$$

Thus we have proved that $\forall v \in V_1 \times V_2$, $\forall w \in V_{01}\times V_{02}$, we have

$$(2.84)\quad \begin{cases} (J'(v),w) = \int_{\Omega_{12}} \nabla(y_2-y_1)\cdot\nabla\tilde{w}_2 dx + \int_{\Omega_{12}} (y_2-y_1)\tilde{w}_2 dx + \\ + \int_{\Omega_{12}} \nabla(y_1-y_2)\cdot\nabla\tilde{w}_1 dx + \int_{\Omega_{12}} (y_1-y_2)\tilde{w}_1 dx \\ - \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla p_i\cdot\nabla\tilde{w}_i dx\ , \end{cases}$$

where $\tilde{w} = \{\tilde{w}_1,\tilde{w}_2\}$ is any extension of w such that

$$\tilde{w}_i \in H^1(\Omega_{ii}),\ \tilde{w}_i|_{\gamma_i} = w_i,\ \tilde{w}_i|_{\partial\Omega_{ii}\cap\Gamma} = 0\ ;$$

in (2.84), $y = \{y_1,y_2\}$ (resp. $p = \{p_1,p_2\}$) is obtained from v (resp. v and y) via (2.63) (resp. (2.80)).

On the other hand it follows from (2.62),(2.63) that we have

$$(2.85)\quad \begin{cases} J(\bar{u}+w) = J(\bar{u})+(J'(\bar{u}),w) + \frac{1}{2}a(w,w) \\ \forall \bar{u} \in V_1\times V_2,\quad \forall\ w \in V_{01}\times V_{02}\ , \end{cases}$$

where, in (2.74), $\delta v = \{\delta v_1, \delta v_2\} \in V_{01} \times V_{02}$ is a variation of v, compatible with the boundary conditions on the $\partial\Omega_{ii} \cap \Gamma$; the δy_i's are the corresponding variations of y_i, i.e.

$$(2.75) \quad \begin{cases} -\Delta \delta y_i = 0 \text{ in } \Omega_{ii}, \\ \delta y_i = 0 \text{ on } \partial\Omega_{ii} \cap \Gamma , \\ \delta y_i = \delta v_i \text{ on } \gamma_i . \end{cases}$$

Let $\tilde{v}_i$ be an extension of v_i over Ω_{ii}, such that

$$(2.76) \quad \tilde{v}_i \in H^1(\Omega_{ii}), \ \tilde{v}_i|_{\gamma_i} = v_i, \ \tilde{v}_i|_{\partial\Omega_{ii} \cap \Gamma} = g_i ;$$

we have $y_i - \tilde{v}_i \in H_o^1(\Omega_{ii})$, and also

$$(2.77) \quad \delta\tilde{v}_i \in H^1(\Omega_{ii}), \ \delta\tilde{v}_i|_{\gamma_i} = \delta v_i, \ \delta\tilde{v}_i|_{\partial\Omega_{ii} \cap \Gamma} = 0 ,$$

$$(2.78) \quad \delta(y_i - \tilde{v}_i) \in H_o^1(\Omega_{ii}).$$

We clearly have from (2.74) that

$$(2.79) \quad \begin{cases} \delta J(v) = (J'(v), \delta v) = \int_{\Omega_{12}} \nabla(y_2 - y_1) \cdot \nabla \delta\tilde{v}_2 dx + \\ + \int_{\Omega_{12}} (y_2 - y_1) \delta\tilde{v}_2 dx + \int_{\Omega_{12}} \nabla(y_1 - y_2) \cdot \nabla\delta\tilde{v}_1 dx + \\ + \int_{\Omega_{12}} (y_1 - y_2) \delta\tilde{v}_1 dx + \int_{\Omega_{12}} \nabla(y_2 - y_1) \cdot \nabla\delta(y_2 - \tilde{v}_2) dx + \\ + \int_{\Omega_{12}} (y_2 - y_1) \delta(y_2 - \tilde{v}_2) dx + \int_{\Omega_{12}} \nabla(y_1 - y_2) \cdot \nabla\delta(y_1 - \tilde{v}_1) dx + \\ + \int_{\Omega_{12}} (y_1 - y_2) \delta(y_1 - \tilde{v}_1) dx . \end{cases}$$

Let us introduce now (for i=1,2) the following adjoint state equations

$$(2.80) \quad \begin{cases} p_i \in H_o^1(\Omega_{ii}) \text{ and} \\ \int_{\Omega_{ii}} \nabla p_i \cdot \nabla z_i dx = \int_{\Omega_{12}} \nabla(y_i - y_j) \cdot \nabla z_i dx + \int_{\Omega_{12}} (y_i - y_j) z_i dx \\ \forall z_i \in H_o^1(\Omega_{ii}) , \end{cases}$$

where $j = i + (-1)^{i-1}$; the Dirichlet problem (2.80) has a unique solution.

(2.69) $u^{n+1} = u^n - \rho_n w^n.$

Step 2 : Calculation of the new descent direction.

Compute

$$(2.70) \quad \begin{cases} g^{n+1} = \{g_1^{n+1}, g_2^{n+1}\} \in V_{01} \times V_{02} \text{ such that} \\ \sum_{i=1}^{2} \int_{\gamma_i} g_i^{n+1} z_i \, d\gamma = (J'(u^{n+1}), z) \quad \forall z = \{z_i\}_{i=1}^{2} \in V_{01} \times V_{02}, \end{cases}$$

then

$$(2.71) \quad \lambda_n = \frac{\sum_{i=1}^{2} \int_{\gamma_i} |g_i^{n+1}|^2 d\gamma}{\sum_{i=1}^{2} \int_{\gamma_i} |g_i^n|^2 d\gamma},$$

$$(2.72) \quad w^{n+1} = g^{n+1} + \lambda_n w^n,$$

(2.73) do $n=n+1$, go to (2.68).

C. Some remarks concerning algorithm (2.65)-(2.73).

It appears clearly from the above description that the two non trivial steps in algorithm (2.65)-(2.73) are (2.68) and the calculation of $J'(u^{n+1})$ in (2.70). It seems that solving (2.68) requires a line search, and therefore the solution of several Poisson's problems like (2.63) ; similarly the calculation of g^{n+1} from u^{n+1} seems to require the solution of several Poisson's problems like (2.63). Actually (and fortunately), using the fact that the cost functional $J(\cdot,\cdot)$ is a quadratic functional of $\{v_1,v_2\}$ we shall be able, via an adequate recurrent relation satisfied by the g^n (very close to (2.52)), to reduce to two the number of Poisson's problems like (2.63), to solve on each Ω_{ii} at each iteration.

D. Calculation of $J'(\cdot)$. Formulation of (2.61) as a linear variational problem in $V_{01} \times V_{02}$.

The following considerations may seem rather theoretical, they are however of fundamental importance in view of the practical implementation of algorithm (2.65)-(2.73) ; let begin by the calculation of $J'(v)$:

Using the notation $y_i = y_i(v_i)$ $(i=1,2)$, we have from (2.62)

$$(2.74) \quad \begin{cases} \delta J(v) = (J'(v), \delta v) = \int_{\Omega_{12}} \nabla(y_2 - y_1) \cdot \nabla \delta(y_2 - y_1) \, dx + \\ \int_{\Omega_{12}} (y_2 - y_1) \delta(y_2 - y_1) \, dx \end{cases}$$

(iii) (2.63) the state equations,
(iv) $J(\cdot,\cdot)$ the cost function.

Remark 2.4 : Other cost functions than (2.62) may be used ; let us mention for example

$$\int_{\Omega_{12}} |y_2(v_2)-y_1(v_1)|^2 dx \;;$$

it seems however than $J(\cdot,\cdot)$ defined by (2.62) is optimally suited for solving (E), via (2.61).

2.3.2. Conjugate gradient solution of (2.61). A first algorithm.

A. Synopsis : Any conjugate gradient algorithm for solving (2.61) will be very close to (2.47)-(2.55) ; it is quite clear that the natural metrics to use on the V_i's are those induced by the $H^1(\Omega_{ii})$'s ; using these metrics is rather technical, therefore, and for simplicity, we shall consider first the case where the V_i's are equipped with the $L^2(\gamma_i)$-norms.

B. Description of the conjugate gradient algorithm : Let us define, for i=1,2, V_{0i} by

$$\begin{cases} V_{0i} = \{v_i \in L^2(\gamma_i),\ v_i = \tilde{v}_i|_{\gamma_i} \text{ where } \tilde{v}_i \in H^1(\Omega_{ii}), \\ \qquad \tilde{v}_i = 0 \text{ on } \partial\Omega_{ii} \cap \Gamma\} \;; \end{cases}$$

the conjugate gradient algorithm is then defined as follows :

Step 0 : Initialization.

(2.65) $\quad u^o = \{u_1^o, u_2^o\} \in V_1 \times V_2$ is arbitrarily given,

define now

$$(2.66)\quad \begin{cases} g^o = \{g_1^o ; g_2^o\} \in V_{01} \times V_{02} \text{ such that} \\ \sum_{i=1}^{2} \int_{\gamma_i} g_i^o z_i \, d\gamma = (J'(u^o), z) \quad \forall z = \{z_i\}_{i=1}^2 \in V_{01} \times V_{02} \end{cases}$$

where J' denotes the differential of J, then set

(2.67) $\quad w^o = g^o.$

Assuming that $u^n = \{u_i^n\}_{i=1}^2$, $g^n = \{g_i^n\}_{i=1}^2$, $w^n = \{w_i^n\}_{i=1}^2$ are known we obtain $u^{n+1}, g^{n+1}, w^{n+1}$ by

Step 1 : Steepest descent.

$$(2.68)\quad \rho_n = \operatorname*{Arg\,Min}_{\rho \in \mathbb{R}} J(u^n - \rho w^n),$$

2.3. A new domain splitting method.

2.3.1. Motivation. An equivalence result.

Our main motivation to design the method which follows was to find a conjugate gradient variant of the Schwarz's algorithm (2.16)-(2.18).

We suppose that Ω has been splitted according to situation (ii) of Sec. 2.1 and we use the notation of Sec. 2.1 and 2.2.1 ; for simplicity we suppose that i,j=1,2 in (2.5),(2.6). We achieve the goal defined above by using a convenient least squares formulation equivalent to (E) (see (2.1)).

Let us introduce first some convenient functional spaces (of traces) :

$$(2.60)\quad \begin{cases} V_i = \{v_i \in L^2(\gamma_i),\ v_i = \tilde{v}_i|_{\gamma_i} \text{ where } \tilde{v}_i \in H^1(\Omega_{ii}), \\ \tilde{v}_i = g_i \text{ on } \partial\Omega_{ii} \cap \Gamma\} \end{cases}$$

for i=1,2. We have then (the quite obvious)

Proposition 2.5 : The minimization problem

$$(2.61)\quad \begin{cases} \text{Find } \{u_1,u_2\} \in V_1 \times V_2 \text{ such that} \\ J(u_1,u_2) \le J(v_1,v_2) \quad \forall\ \{v_1,v_2\} \in V_1 \times V_2, \end{cases}$$

where

$$(2.62)\quad J(v_1,v_2) = \frac{1}{2}\int_{\Omega_{12}} \{|\nabla(y_2(v_2)-y_1(v_1))|^2 + |y_2(v_2)-y_1(v_1)|^2\}dx,$$

with $y_i(v_i)$ solutions (for i=1,2) of

$$(2.63)\quad \begin{cases} -\Delta y_i = f_i \text{ in } \Omega_{ii}, \\ y_i = g_i \text{ on } \partial\Omega_{ii} \cap \Gamma, \\ y_i = v_i \text{ on } \gamma_i, \end{cases}$$

has a unique solution such that

$$(2.64)\quad u_i = y|_{\gamma_i} \quad \forall\ i=1,2,$$

where y is the solution of (E).

Remark 2.3 : Problem (2.61) has clearly the structure of an optimal control problem (see LIONS [19]) with

(i) v_1, v_2 the control variables,

(ii) y_1, y_2 the state variables,

$$(2.53)\qquad \gamma_n = \frac{\|g^{n+1}\|^2_{L^2(\gamma_{12})}}{\|g^n\|^2_{L^2(\gamma_{12})}},$$

$$(2.54)\qquad w^{n+1} = g^{n+1}+\gamma_n w^n ,$$

(2.55) <u>do</u> n=n+1, <u>go to</u> (2.50).

The formalism of algorithm (2.47)-(2.55) may seem quite abstract ; in fact algorithm (2.47)-(2.55) is no more complicated to implement than algorithm (2.30)-(2.33), and like this latter requires only the solution at each iteration of a Poisson problem on each Ω_{ii} ; indeed g^o is obtained from λ^o by

$$(2.56)\qquad \begin{cases} -\Delta y_i^o = f_i \ \underline{\text{in}}\ \Omega_{ii} , \\ y_i^o = g_i \ \underline{\text{on}}\ \partial\Omega_{ii}\cap\Gamma, \\ \dfrac{\partial y_i^o}{\partial n_i} = \lambda^o \ \underline{\text{on}}\ \gamma_{12} \end{cases}$$

for i=1,2, and then

$$(2.57)\qquad g^o = (y_1^o-y_2^o)|_{\gamma_{12}}.$$

Concerning Steps 1 and 2, the key point is the calculation of $\mathcal{A}w^n$; this is done as follows :
We define $\{\chi_1^n,\chi_2^n\}\in H^1(\Omega_{11})\times H^1(\Omega_{22})$ by

$$(2.58)\qquad \begin{cases} \Delta\chi_i^n = 0 \ \underline{\text{in}}\ \Omega_{ii}, \\ \chi_i^n = 0 \ \underline{\text{on}}\ \partial\Omega_{ii}\cap\Gamma , \\ \dfrac{\partial\chi_i^n}{\partial n_i} = (-1)^{i-1}w^n \ \underline{\text{on}}\ \gamma_{12} \end{cases}$$

for i=1,2, then

$$(2.59)\qquad \mathcal{A}w^n = (\chi_1^n-\chi_2^n)|_{\gamma_{12}} .$$

Thus one iteration of (2.47)-(2.55) is no more costly than one iteration of algorithm (2.30)-(2.33).

<u>Remark 2.2</u> : Another conjugate gradient variant of algorithm (2.30)-(2.33) is given in LEMONNIER [18] ; it is however <u>twice</u> more costly than (2.47)-(2.55).

The main result is the following

Proposition 2.4 : Let λ be the solution of the dual problem (2.37) ; λ is also the solution of the following linear problem in $L^2(\gamma_{12})$:

$$(2.45) \qquad \mathcal{A}\lambda = (y_{02}-y_{01})|_{\gamma_{12}}$$

where $\{y_{01},y_{02}\} \in H^1(\Omega_{11})\times H^1(\Omega_{22})$ is defined by

$$(2.46) \qquad \begin{cases} -\Delta y_{0i} = f_i \text{ in } \Omega_{ii} , \\ y_{0i} = g_i \text{ on } \partial\Omega_{ii} \cap \Gamma , \\ \dfrac{\partial y_{0i}}{\partial n_i} = 0 \text{ on } \gamma_{12} , \end{cases}$$

for $i=1,2$.

Proof : Very similar to the proof of Theorem 2.2 in GLOWINSKI-PIRONNEAU [10, Sec. 2.5] . ■

Using the above results we can describe now the following conjugate gradient variant of algorithm (2.30)-(2.33) (with $L = L^2(\gamma_{12})$) :

Step 0 : Initialization.

$$(2.47) \qquad \lambda^o \in L, \text{ arbitrarily given,}$$

$$(2.48) \qquad g^o = \mathcal{A}\lambda^o - (y_{02}-y_{01})|_{\gamma_{12}} ,$$

$$(2.49) \qquad w^o = g^o ;$$

then for $n \geq 0$, assuming that $\{\lambda^n,w^n,g^n\} \in L\times L\times L$ are known we compute $\lambda^{n+1},w^{n+1},g^{n+1}$ by

Step 1 : Steepest descent.

$$(2.50) \qquad \lambda^{n+1} = \lambda^n-\rho_n w^n$$

with

$$(2.51) \qquad \rho_n = \frac{\int_{\gamma_{12}} g^n w^n d\gamma}{\int_{\gamma_{12}} (\mathcal{A}w^n)w^n d\gamma} = \frac{\int_{\gamma_{12}} |g^n|^2 d\gamma}{\int_{\gamma_{12}} (\mathcal{A}w^n)w^n d\gamma} .$$

Step 2 : Calculation of the new descent direction.

$$(2.52) \qquad g^{n+1} = g^n-\rho_n\mathcal{A}w^n ,$$

$$\mathcal{A} : L^2(\gamma_{12}) \to L^2(\gamma_{12})$$

as follows :

Let $\lambda \in L^2(\gamma_{12})$; to λ we associate $y_1(\lambda) = y_1$, $y_2(\lambda) = y_2$ as the solutions in $H^1(\Omega_{11})$ and $H^1(\Omega_{22})$, respectively, of

(2.40)
$$\begin{cases} \Delta y_1 = 0 \text{ in } \Omega_{11}, \\ y_1 = 0 \text{ on } \partial\Omega_{11} \cap \Gamma, \\ \dfrac{\partial y_1}{\partial n_1} = \lambda \text{ on } \gamma_{12}, \end{cases}$$

(2.41)
$$\begin{cases} \Delta y_2 = 0 \text{ in } \Omega_{22}, \\ y_2 = 0 \text{ on } \partial\Omega_{22} \cap \Gamma, \\ \dfrac{\partial y_2}{\partial n_2} = -\lambda \text{ on } \gamma_{12}, \end{cases}$$

and we finally define $\mathcal{A}$ by

(2.42)
$$\mathcal{A}\lambda = (y_1(\lambda) - y_2(\lambda))|_{\gamma_{12}}.$$

We have then

Proposition 2.3 : The operator $\mathcal{A} : L^2(\gamma_{12}) \to L^2(\gamma_{12})$ defined by (2.40)-(2.42) is continuous, symmetric and positive definite ; it is not, however, strongly elliptic.

Proof : We just prove that $\mathcal{A}$ is symmetric and positive definite : Let $\mu \in L^2(\gamma_{12})$ and let $y_1(\mu)$, $y_2(\mu)$ be the corresponding solutions of (2.40),(2.41) ; integrating by parts we clearly have

(2.43)
$$\begin{cases} \displaystyle\int_{\gamma_{12}} \mathcal{A}\lambda\, \mu\, d\gamma = \int_{\gamma_{12}} y_1(\lambda) \frac{\partial y_1}{\partial n_1}(\mu) d\gamma + \int_{\gamma_{12}} y_2(\lambda) \frac{\partial y_2}{\partial n_2}(\mu) d\gamma \\ \displaystyle = \sum_{i=1}^{2} \int_{\Omega_{ii}} \nabla y_i(\lambda) \cdot \nabla y_i(\mu)\, dx. \end{cases}$$

The symmetry of $\mathcal{A}$ is obvious from (2.43) ; moreover taking $\lambda = \mu$ in (2.43) shows that $\mathcal{A}$ is positive definite.

Remark 2.1 : Let $a(\cdot,\cdot)$ be the bilinear form in (2.39) ; it follows from Proposition 2.3 that we have

(2.44)
$$a(\lambda,\mu) = \int_{\gamma_{12}} \mathcal{A}\lambda\, \mu\, d\gamma \quad \forall\, \lambda,\mu \in L^2(\gamma_{12}). \blacksquare$$

<u>Proposition 2.2</u> : <u>Suppose that the saddle point problem</u> (2.29) <u>has a solution</u> ; <u>then,</u> $\forall\ \lambda^o \in L^2(\gamma_{12})$ <u>we have,</u> $\forall\ i=1,2,$

(2.34) $$\lim_{n\to+\infty} y_i^n = y_i = y|_{\Omega_{ii}} \quad \text{strongly in } H^1(\Omega_{ii}) ,$$

<u>where</u> y <u>is the solution of problem</u> (E), <u>if</u>

(2.35) $$0 < \rho < \frac{2}{C^2} ,$$

<u>where,</u> <u>in</u> (2.35),

(2.36) $$C = \sup_{\{z_1,z_2\}\in Z-\{0\}} \frac{\|z_2-z_1\|_{L^2(\gamma_{12})}}{\|\{z_1,z_2\}\|_Z} ,$$

<u>and where</u>

$$Z = \{\{z_1,z_2\} \in H^1(\Omega_{11})\times H^1(\Omega_{22}),\ z_i|_{\partial\Omega_{ii}\cap\Gamma} = 0\ \forall i=1,2\},$$

$$\|\{z_1,z_2\}\| = \left(\sum_{i=1}^{2} \int_{\Omega_{ii}} |\nabla z_i|^2 dx\right)^{1/2} . \blacksquare$$

The above algorithm is in fact a <u>gradient algorithm</u>, with the <u>constant step</u> ρ, applied to the solution of the <u>dual</u> problem associated to (2.29), i.e.

(2.37) $$\begin{cases} \underline{\text{Find}}\ \lambda \in L\ \underline{\text{such that}}, \\ J^*(\lambda) \le J^*(\mu) \quad \forall \mu \in L, \end{cases}$$

where

(2.38) $$J^*(\mu) = - \inf_{\{z_1,z_2\}\in V} \mathcal{L}(z_1,z_2,\mu) ;$$

we have then

$$\lambda = \frac{\partial y_1}{\partial n_1} = - \frac{\partial y_2}{\partial n_2} \quad \underline{\text{on}}\ \gamma_{12} ,$$

where, $\forall\ i=1,2$, $\frac{\partial y_i}{\partial n_i}$ denotes the normal derivative at γ_{12}, exterior to Ω_{ii}.

It is fairly easy to show that

(2.39) $$J^*(\mu) = \frac{1}{2}\, a(\mu,\mu) - \ell(\mu) ,$$

where $a(\cdot,\cdot)$ is <u>bilinear</u>, <u>continuous</u>, <u>symmetric</u>, <u>positive definite</u> over L×L (but <u>not</u> strongly-elliptic) and where $\ell(\cdot)$ is <u>linear continuous</u> over L.

The above properties clearly suggest a conjugate gradient algorithm for solving (2.37), and therefore (E) ; in order to have a better understanding of the dual problem (2.37) we shall give an <u>equivalent operator formulation</u>. Let us define then a linear operator

The fact that $\mathcal{L}$ can be used to solve (E) (via (2.23)) lies on the following

Proposition 2.1 : Suppose that $\{y_1,y_2,\lambda\}$ is a saddle-point of $\mathcal{L}$ over $V\times L$, i.e.

$$(2.29)\quad \begin{cases} \{y_1,y_2,\lambda\} \in V\times L, \\ \mathcal{L}(y_1,y_2,\mu) \le \mathcal{L}(y_1,y_2,\lambda) \le \mathcal{L}(z_1,z_2,\lambda) \quad \forall\{z_1,z_2,\mu\}\in V\times L. \end{cases}$$

We have then $\forall$ $i=1,2$

$$y_i = y|_{\Omega_{ii}} ,$$

where y is the solution of (E). The reciprocal property holds if y is sufficiently smooth.

From Proposition 2.1 we can replace the solution of (E) by the solution of the saddle-point problem (2.29).
It follows from GLOWINSKI-LIONS-TREMOLIERES [15, Ch. 2], [16, Ch. 2 and App. 2], BENSOUSSAN-LIONS-TEMAM [17], that (2.29) can be solved by a saddle-point solver like Uzawa's algorithm (and its conjugate gradient variants) ; let us describe the basic Uzawa's algorithm corresponding to (2.29), it is defined by :

$$(2.30)\quad \lambda^o \in L, \text{ arbitrarily given,}$$

then for $n \ge 0$, assuming λ^n known, we compute y_1^n, y_2^n and λ^{n+1} by[1]

$$(2.31)\quad \begin{cases} -\Delta y_1^n = f_1 \text{ in } \Omega_{11}, \\ y_1^n = g_1 \text{ on } \partial\Omega_{11} \cap \Gamma , \\ \dfrac{\partial y_1^n}{\partial n_1} = \lambda^n \text{ on } \gamma_{12}, \end{cases}$$

$$(2.32)\quad \begin{cases} -\Delta y_2^n = f_2 \text{ in } \Omega_{22}, \\ y_2^n = g_2 \text{ on } \partial\Omega_{22} \cap \Gamma , \\ \dfrac{\partial y_2^n}{\partial n_2} = -\lambda^n \text{ on } \gamma_{12} , \end{cases}$$

$$(2.33)\quad \lambda^{n+1} = \lambda^n + \rho(y_2^n - y_1^n)|_{\gamma_{12}} , \rho > 0.$$

Concerning the convergence of algorithm (2.30)-(2.33) we can prove the following

---

(1) n_i : outward unitary normal vector at $\partial\Omega_{ii}$

2.2.2. A domain splitting method with coordination by Lagrange multipliers.

Problem (E) is equivalent to the following minimization problem

(2.19) $$\begin{cases} \text{Find } y \in V_g \text{ such that} \\ J(y) \leq J(z) \quad \forall\, z \in V_g \end{cases}$$

where

(2.20) $$V_g = \{z \in H^1(\Omega),\quad z=g \text{ on } \Gamma\},$$

(2.21) $$J(z) = \frac{1}{2}\int_\Omega |\nabla z|^2 dx - \int_\Omega fz\, dx.$$

We consider now a decomposition of Ω as done in Sec. 2.1, case (i) (we just consider here, as in Figure 2.*, two subdomains Ω_{11}, Ω_{22}) ; we associate to this decomposition the functionals $J_i : H^1(\Omega_{ii}) \to \mathbb{R}$ defined by

(2.22) $$J_i(z_i) = \frac{1}{2}\int_{\Omega_{ii}} |\nabla z_i|^2\, dx - \int_{\Omega_{ii}} f_i z_i dx,\quad i=1,2.$$

It is then quite easy to show that (2.19) (and therefore (E)) is equivalent(1) to the following minimization problem

(2.23) $$\begin{cases} \text{Find } \{y_1,y_2\} \in W \text{ such that} \\ J_1(y_1)+J_2(y_2) \leq J_1(z_1)+J_2(z_2) \quad \forall\, \{z_1,z_2\} \in W, \end{cases}$$

where

(2.24) $$\begin{cases} W = \{\{z_1,z_2\} \in H^1(\Omega_{11}) \times H^1(\Omega_{22}),\ z_1|_{\gamma_{12}} = z_2|_{\gamma_{12}}, \\ z_i|_{\partial\Omega_{ii} \cap \Gamma} = g_i\ (= g|_{\partial\Omega_{ii} \cap\Gamma})\ \forall\, i=1,2\}. \end{cases}$$

The main difficulty in (2.23) is that it involves the linear constraint

(2.25) $$z_1|_{\gamma_{12}} = z_2|_{\gamma_{12}} \quad ;$$

a standard method to overcome (2.25) is to associate to it a Lagrange multiplier, via a convenient lagrangian functional ; in that direction a quite natural choice is $\mathcal{L}$ defined by

(2.26) $$\mathcal{L}(z_1,z_2,\mu) = J_1(z_1)+J_2(z_2) + \int_{\gamma_{12}} \mu(z_2-z_1)\, d\gamma\ .$$

Let us introduce now the following spaces

(2.27) $$V = \{\{z_1,z_2\} \mid \forall\, i=1,2\ ,\ z_i \in H^1(\Omega_{ii}), z_i|_{\partial\Omega_{ii}\cap\Gamma} = g_i\},$$

(2.28) $$L = L^2(\gamma_{12})\ .$$

---

(1) In that $y_i = y|_{\Omega_{ii}}$, $\forall i=1,2$.

We split Ω as in Sec. 2.1, case (ii), using Ω_{11}, Ω_{22} (see Figure 2.4 for other notations) ; the problem under consideration being (E) (see (2.1)) we introduce the following functions

(2.14) $f_i = f|_{\Omega_{ii}}$, i=1,2,

(2.15) $g_i = g|_{\partial\Omega_{ii} \cap \Gamma}$, i=1,2 .

The algorithm is defined by

(2.16) Let a function η be defined on γ_1 then for $n \geq 0$ we construct a sequence $\{y_1^n, y_2^n\}$ as follows

$$(2.17) \quad \begin{cases} -\Delta y_1^n = f_1 \text{ in } \Omega_{11}, \\ y_1^n = g_1 \text{ on } \partial\Omega_{11} \cap \Gamma , \\ y_1^n|_{\gamma_1} = y_2^{n-1}|_{\gamma_1} \text{ if } n \geq 1, \\ y_1^o|_{\gamma_1} = \eta , \end{cases}$$

$$(2.18) \quad \begin{cases} -\Delta y_2^n = f_2 \text{ in } \Omega_{22} , \\ y_2^n = g_2 \text{ on } \partial\Omega_{22} \cap \Gamma , \\ y_2^n|_{\gamma_2} = y_1^n|_{\gamma_2} . \end{cases}$$

The mechanism of the above method is shown on Figure 2.4.

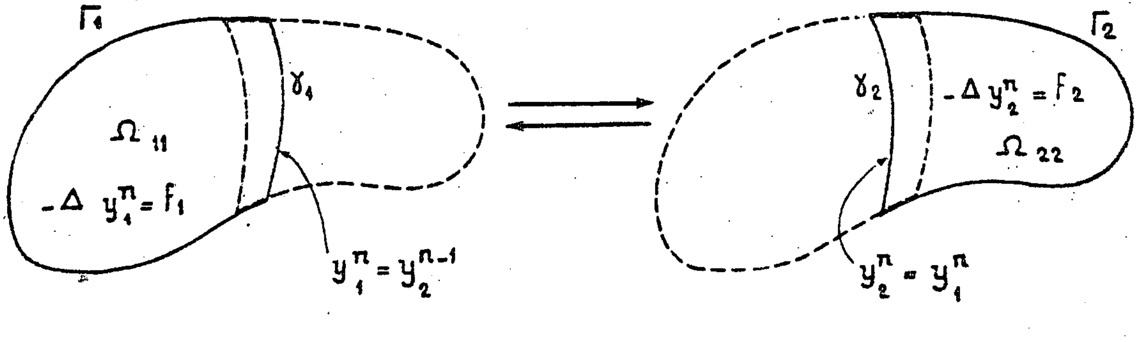


Figure 2.4

Mechanism of the Swhwarz Alternating Method

To conclude this section we shall introduce some functional spaces, useful in the sequel :

(2.7) $H^1(\Omega) = \{v \in L^2(\Omega), \frac{\partial v}{\partial x_i} \in L^2(\Omega) \quad \forall\ i=1,\ldots N\}$,

(2.8) $H_o^1(\Omega) = \{v \in H^1(\Omega),\ v=0 \text{ on } \Gamma\}$;

the space $H^1(\Omega)$ is an Hilbert space for the inner product

(2.9) $(u,v)_{H^1(\Omega)} = \int_\Omega uv\, dx + \int_\Omega \nabla u \cdot \nabla v\, dx$;

moreover $H_o^1(\Omega)$ is a closed subspace of $H^1(\Omega)$, and if Ω is bounded (at least in one direction of $\mathbb{R}^N$), then $H_o^1(\Omega)$ is a Hilbert space if equipped with the inner product

(2.10) $(u,v)_{H_o^1(\Omega)} = \int_\Omega \nabla u \cdot \nabla v\, dx$,

and the corresponding norm $\|v\|_{H_o^1(\Omega)} = (\int_\Omega |\nabla v|^2\, dx)^{1/2}$ is equivalent to the norm induced by $H^1(\Omega)$. Suppose that $L^2(\Omega)$ has been identified to its dual space then if $H^{-1}(\Omega)$ is the dual space of $H_o^1(\Omega)$ we have

(2.11) $H_o^1(\Omega) \subset L^2(\Omega) \subset H^{-1}(\Omega)$.

Furthermore the operator $-\Delta$ is an isomorphism from $H_o^1(\Omega)$ onto $H^{-1}(\Omega)$; we shall denote by $<\cdot,\cdot>$ the duality pairing between $H^{-1}(\Omega)$ and $H_o^1(\Omega)$ such that

(2.12) $<f,v> = \int_\Omega fv\, dx \quad \forall\ f \in L^2(\Omega),\ \forall\ v \in H_o^1(\Omega)$;

if $f \in H^{-1}(\Omega)$ we have

(2.13) $\|f\|_{-1} = \sup_{v \in H_o^1(\Omega)-\{0\}} \frac{|<f,v>|}{\|v\|_{H_o^1(\Omega)}}$.

For more properties and details on Sobolev spaces we refer to, e.g., NECAS [12], ADAMS [13], ODEN-REDDY [14].
In the next sections the above definitions hold also for the sub-domains Ω_{ii}.

2.2. A review of two methods for solving problem (E), based on domain splitting.

2.2.1. The Schwarz alternating method.

This method was introduced by Schwarz around 1860 to solve Dirichlet problems and its convergence was proved using the Maximum Principle. Let us describe briefly the method :

Figure 2.2 illustrates situation (ii) (with i=1,2,3 in (2.3)) ; a more general situation of this type is shown on Fig. 2.3.

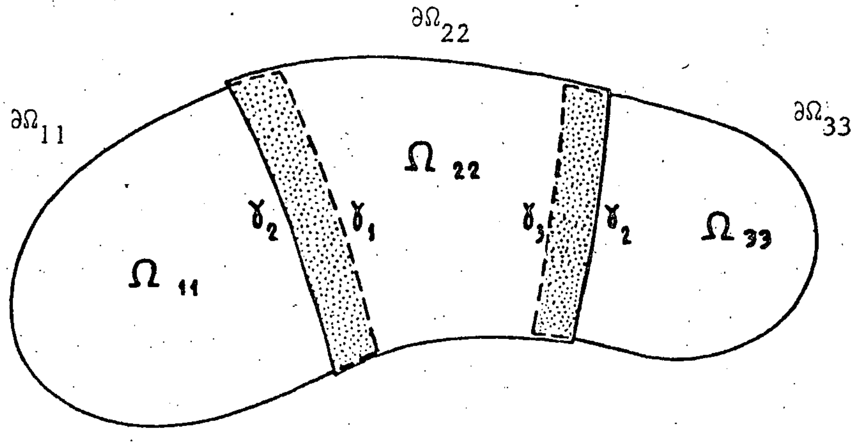


Figure 2.2

Decomposition of type (ii)

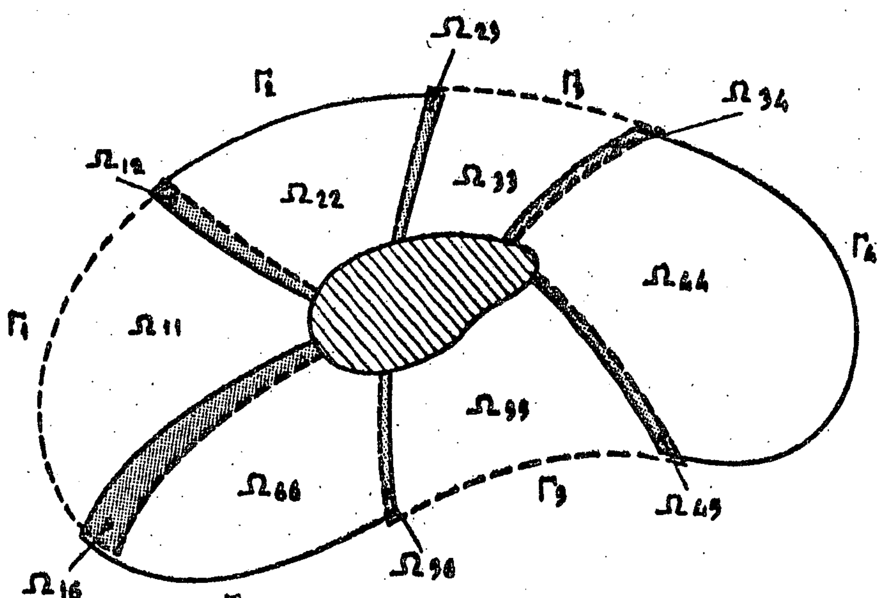


Figure 2.3

In the sequel we shall face the two following types of situations :

(i) We have

(2.3) $$\Omega = \overline{\overset{\circ}{\bigcup_i \Omega_{ii}}}$$

where $\overline{}$ (resp. $^\circ$) denotes the <u>closure</u> (resp. the <u>interior</u>) of a set and also

(2.4) $$\Omega_{ii} \cap \Omega_{jj} = \emptyset \quad \forall\, i,j\ ,\ i \neq j\ .$$

We introduce also γ_{ij} defined by

$$\gamma_{ij} = \gamma_{ji} = \partial\Omega_{ii} \cap \partial\Omega_{jj} \text{ if } i \neq j\ ;$$

Figure 2.1 illustrates this situation (i) (with i=1,2 in (2.3)).

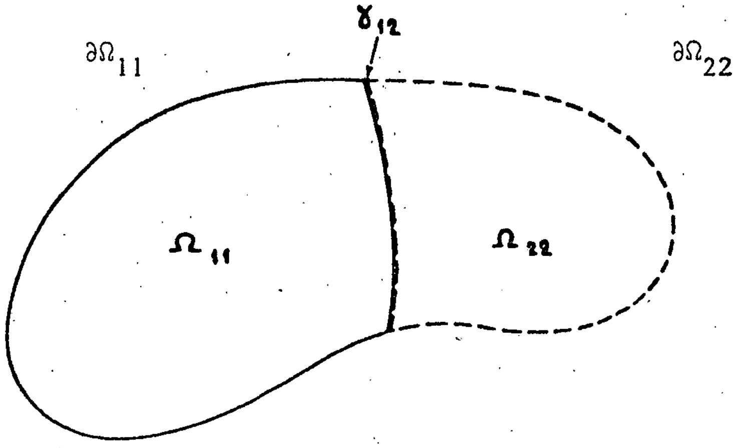


Figure 2.1
Decomposition of type (i)

(ii) We still have (2.3) but (2.4) does not hold since we have, either

(2.5) $$\overline{\Omega}_{ii} \cap \overline{\Omega}_{jj} = \emptyset \quad \forall\, i,j\ ,\ i \neq j,$$

or

(2.6) $$\begin{cases} \Omega_{ii} \cap \Omega_{jj} = \Omega_{ij} \quad (= \Omega_{ji})\ , \\ \displaystyle\int_{\Omega_{ij}} dx > 0\ . \end{cases}$$

We introduce also

$$\gamma_i = \partial\Omega_{ii} - (\partial\Omega_{ii} \cap \Gamma).$$

The main goal of this paper is to describe a new least-squares conjugate gradient variant of the above Schwarz method which provides efficient tools for solving complicated non-linear problems on complicated 2-D and 3-D geometries. This new method has moreover very good properties in view of its implementation on vector machines.

The content of the paper is as follows : in Sec. 2, taking Laplace equation as a model problem we do first a short review of domain decomposition methods ; then we introduce this new least-squares domain splitting technique, and discuss its application via a conjugate gradient algorithm and a quasi-direct method (in the sense of [10],[11]).

In Sec. 3, we show how the above methods can be used as preconditioners for solving by nonlinear least squares and conjugate gradient a nonlinear Poisson model problem ; this method is then extended in Sec. 4 to the solution of much more complicated nonlinear problems, namely the numerical simulation of transonic potential flows for compressible inviscid fluids and to the Navier-Stokes equations of incompressible viscous fluids.

In Sec. 5 various numerical experiments illustrate the possibility of these new methods and show also some comparisons with pre-existing methods using that concept of domain splitting.

2. NUMERICAL SOLUTION OF LINEAR DIRICHLET PROBLEMS USING DOMAIN SPLITTING METHODS.

In view of introducing the decomposition techniques to be applied in Sec. 4 to the solution of nonlinear problems in Fluid Dynamics, we consider in this section the solution of simple linear Dirichlet problems by some of these techniques, and particularly by our least squares-conjugate gradient method.

2.1. Formulation of the model problem.

Let Ω be a bounded domain of $\mathbb{R}^N$ (N=2,3 in practice) with a smooth boundary $\Gamma = \partial\Omega$. We consider in Ω the following linear Dirichlet problem (E)

$$(2.1)\quad (E)\begin{cases} -\Delta y = f \ \underline{\text{in}}\ \Omega, \\ y = g \ \underline{\text{on}}\ \Gamma \end{cases}$$

(where f and g are given functions) ; we associate to (E) the family $((E)_i)_i$ of linear problems defined by

$$(2.2)\quad (E)_i\begin{cases} -\Delta y_i = f_i \ \underline{\text{on}}\ \Omega_{ii}, \\ y_i = g_i \ \underline{\text{on}}\ \partial\Omega_{ii} \end{cases}$$

where $\Omega_{ii} \subset \Omega$.

ABSTRACT

The numerical solution of finite element approximations of complicated two and three dimensional nonlinear problems can be a most formidable task. In order to overcome this difficulty related to dimensionality, domain splitting methods can be very effective, particularly in view of obtaining a fast and economical conjugate gradient solver, which can be used to precondition the solution of nonlinear problems by optimization methods via nonlinear least squares or weighted residual formulations. A new technique of this type will be introduced and analysed and its efficiency will be discussed from numerical experiments concerning the numerical simulation of transonic flows for compressible inviscid fluids and incompressible viscous flows modelled by the Navier-Stokes equations.

1. INTRODUCTION.

Pressure calculation is an important step of the design of new aircrafts, using the methods of Computational Fluid Dynamics. To take into account the actual complicated three-dimensional geometries associated to an aircraft, it is quite natural to use finite element methods for solving the nonlinear partial differential equations governing the flows around these geometries. This requires usually the use of quite powerful computers, able to solve very large nonlinear systems (10^4 unknowns, and even more).

Using conjugate gradient methods with scaling (one says also preconditioning), one has been able to solve on sequential computers these very large nonlinear systems via convenient least squares formulations (see [1]-[5] for more details).

It is clear however that the numerical simulation - by any method - of much more complicated situations, like three dimensional compressible viscous, turbulent separated flows requires much more computing power ; in that direction computers having parallel computation abilities seem to be a very promising approach in a near future. This new generation of computers, quite specialized in their applications leads to the study of new classes of numerical algorithms, particularly in Numerical Linear Algebra (cf. e.g. [6],[7]).

A possible approach to the numerical solution of large linear (and nonlinear) boundary value problems, via task decomposition is provided by the methods using the concept of domain splitting.

Concerning these methods based on domain splitting we can say that they originate from the wellknown Schwarz method (around 1860) for solving elliptic problems ; this method which is essentially sequential has been recently considered from a more numerical point of view by various authors (cf. e.g. [8],[9]) ; we may find in particular in [8] variants of the Schwarz method with good parallelization properties.

CONTENTS

RESUME

La résolution numérique des problèmes approchés par des méthodes d'éléments finis peut être une opération extrêmement coûteuse. Dans le but de surmonter cette difficulté les méthodes de décomposition de domaines peuvent se révéler comme extrêmement efficaces. En particulier pour obtenir des algorithmes de gradient conjugué préconditionné, applicables à la résolution de problèmes non linéaires, via l'utilisation de méthodes de moindres carrés. Une nouvelle méthode de décomposition de domaines est introduite et analysée dans ce rapport et ces possibilités évaluées par l'intermédiaire d'expériences numériques concernant la simulation sur ordinateur d'écoulements transoniques de fluides parfaits compressibles, et de fluides visqueux incompressibles, modélisés par les équations de Navier-Stokes.

ABSTRACT

The numerical solution of finite element approximation of complicated two and three dimensional nonlinear problems can be a most formidable task. In order to overcome this difficulty related to dimensionality, domain splitting methods can be very effective, particularly in view of obtaining a fast and economical conjugate gradient solver, which can be used to precondition the solution of nonlinear problems by optimization methods via nonlinear least squares or weighted residual formulations. A new technique of this type will be introduced and analysed and its effciency will be discussed from numerical experiments concerning the numerical simulation of transonic flows for compressible inviscid fluids and incompressible viscous flows modelled by the Navier-Stokes equations.

DOMAIN DECOMPOSITION METHODS FOR NONLINEAR PROBLEMS IN FLUID DYNAMICS

R. GLOWINSKI

Université Paris VI, L.A. 189, Tour 55-65, 5è étage
4 Place Jussieu, 75230 Paris Cedex 05, and INRIA
Domaine de Voluceau, Rocquencourt, 78153 Le Chesnay, France

Q.V. DINH and J. PERIAUX

A.M.D./BA, 78 Quai Carnot, B.P. 300
92214 St Cloud, France